Wirtschaftswissenschaftliche Fakultät

Lehrstuhl für Betriebswirtschaftslehre mit Schwerpunkt

Management Science / Operations and Supply Chain Management

Bachelorarbeit

Operations and Supply Chain Management

Exact models for the Pickup and Delivery Problem with Transfers – a comparative study

Themensteller: Prof. Dr. Alena Otto

Betreuer: Luis Rocha

Bearbeitet von: Kieu Minh Tri

Matrikelnummer: 103093

Studiengang: Bachelor Wirtschaftsinformatik

Fachsemester: 8

Anschrift: Spitalhofstr. 75A, 94032 Passau

Telefon: +4915258239092

E-Mail: trikieu293@gmail.com

Passau, Abgabedatum: 25.08.2023

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Bachelorarbeit / Masterarbeit selbstständig, ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt hat. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Quellen entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Passau, 24.08.2023

Ort, Datum

Unterschrift

Contents

1. Introduction

In recent years, there has been a growing interest in the pickup and delivery problem. The consistent concern within the field of corporate operations has been the optimization of costs in logistics and transportation. The competitive advantage of producers is enhanced through the reduction of transportation time, transportation distance, and fleet costs, resulting in a corresponding decrease in the overall cost of products for end-users. The optimization of third-party delivery businesses, particularly in the context of increased competition in the sector and increased client demand for quantity and quality, holds significant importance. This is especially true for passenger transportation companies, as the optimization directly impacts the user experience.

The increasing emphasis on transportation flexibility is driven by the implementation of stricter criteria by users and enterprises. The implementation of transfer stations, where requests can be transferred between vehicles, is receiving increasing attention. This extension of the transport system has been proven through many studies to enhance performance and effectively address challenging situations that would otherwise remain unresolved. Therefore, this thesis aims at studying the Pickup and Delivery Problem with Transfers (PDPT), which is a generalization of the Pickup and Delivery Problem (PDP).

The inclusion of time limitations in service level agreements (SLAs) is a common practice in various real-world scenarios between logistics businesses and their clients. Businesses, for example, may have specific hours of operation, and customers may only be available at certain times. As a result, in the models that will be discussed, we incorporate time-window constraints to help bring the study closer to real-world instances. In addition, taking time-window constraints into account contributes to unifying the method used to remove sub-tours in these models.

In their study, Cortés, Matamala, and Contardo (2010) present a node-based mixed-integer linear programming model for the PDPT with time windows for requests (PDPTWT). In this study, the authors propose a design strategy that involves dividing each transfer point into two distinct nodes in order to accurately identify the specific operations that take place, such as loading or unloading the request. The authors argue that this design approach offers advantages in terms of optimizing customer-based objective functions. In their study, Rais, Alvelos, and Carvalho (2014) propose a generic mixed-integer linear programming (MILP) model for the Pickup and Delivery Problem with Time Windows (PDPT). They use additional binary variables as a logical counter for transfer operations. This model has been used in many other studies.

This thesis aims to compare and evaluate the differences in decision variables, constraints, and transfer point configurations between the model proposed by Cortés et al. (2010) and the model suggested by Rais et al. (2014). Additionally, this study will assess the performance of these two models. This thesis examines the issues related to two models and proposes improvements based on existing studies (most of which have been outlined in other studies).

In the following section, we will begin with the theoretical foundations, defining PDP and PDP(TW)T and comparing the fundamental distinctions between the two problems. We also investigate the effects of transfer locations and conduct a brief literature review to assess the results within the context of existing information. Following that, we conduct a thorough comparison of two mixed-integer mathematical formulations for these scheduling problems, one proposed by Cortés et al. (2010) and the other by Rais et al. (2014). A table of unified parameters used in both models is shown, as well as an analysis of decision variables and a comparison of objective functions and constraints in both models. We gain insights into the complexity of each formulation by assessing the number of variables and constraints. Next, we examine the revisions made by Lyu and Yu (2022) to improve the model of Rais et al. (2014) and propose new revisions and constraints with the goal of improving the performance of this model. The symmetries breaking constraints (CBS) are taken into account in order to improve the solving process even further. Finally, computational analysis will be performed to visualize the study. Then, we compare the performances of the analyzed models and check the effectiveness of the techniques used.

2. Theoretical Foundations

2.1. Definition of PDP and PDPT

The pickup and delivery problem (PDP) is a generalization of the vehicle routing problem (VRP), a NP-hard problem in which a list of requests must be satisfied by a fleet of vehicles. Information about the pickup and delivery locations of each request, as well as the origin and destination depots of each vehicle, is known in advance. At each point of picking up (or delivering) the request, only one corresponding operation occurs. Therefore, if more than one operation takes place at a location, this location will be considered multiple separate locations with the same coordinates.

The goal of the problem is to find the best set of paths within the constraints so that the requests can be served as efficiently as possible. This objective is usually defined as a combination of vehicle cost (from the service provider's perspective) and customer satisfaction (from the customer's perspective), depending on the actual characteristics of the instances. In the case of customer shipping, it can be related to the dial-a-ride problem. The most important stakeholders are express post, postal couriers, shipping companies, and carrier companies (Shiri, Rahmani and Bafruei, 2019).

In traditional PDP, all requests must be fulfilled. Each vehicle's path must begin at its origin depot and terminate at its destination depot (due to depot constraints). Each pick-up and delivery location must be visited at once (visiting constraints). Vehicles

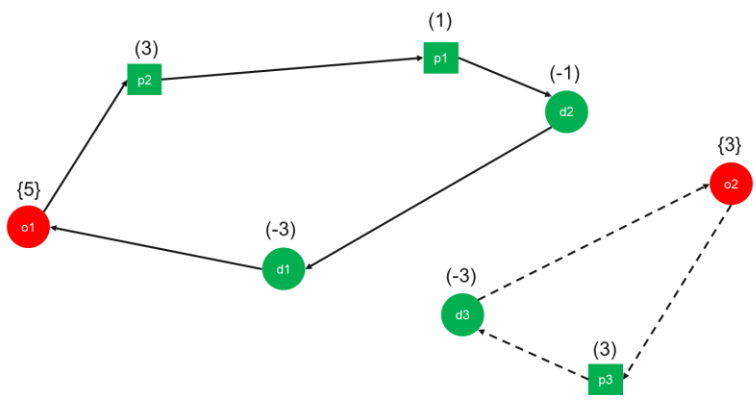

Figure 1: An instance of PDP

can't carry requests that exceed their capacity (vehicle capacity constraints). We must visit the pickup location before the delivery location in order to fulfill a request (precedence constraints). Because of coupling constraints, a request can only be served by one vehicle. Resource constraints refer to limitations on the resources needed for jobs. This could include limitations on the availability of vehicles or employees required to complete tasks.

Figure 1 shows an example of the PDP. Here we have two vehicles. The origin and destination depots of each vehicle are located at the same coordinates. The capacity of vehicle one is 5 and 3 for vehicle two. Then we have three requests and loads of them. *p1* and *d1,* respectively, are the pickup and delivery locations of vehicle one.

In pickup and delivery problems with transfers (PDPT), we relax the coupling constraints. Requests are allowed to transfer between vehicles at the transfer points. By expanding solution space, we can refer to PDPT as a generalization of classical PDP.

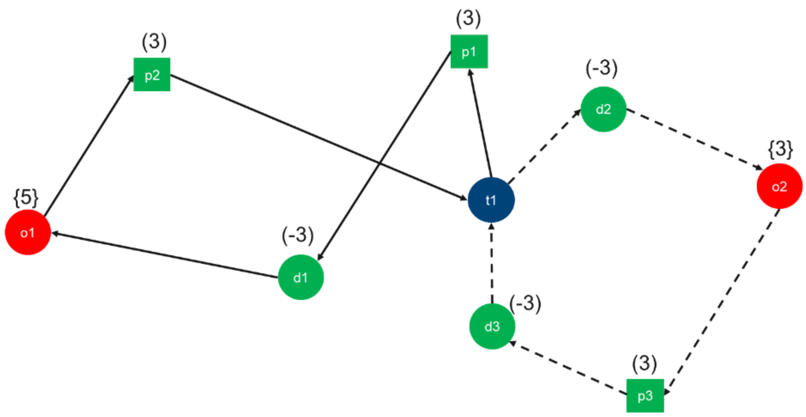

Figure 2: An instance of PDPT

Figure 2 shows an example of a PDPT instance. The instance configuration is identical to the previous PDP example, with the addition of a transfer point and a greater demand for request two. As a result, vehicle one cannot simultaneously transport requests one and two. Therefore, request two will be transferred at the transfer point between vehicles one and two.

2.2. Effects of transfer locations

The trade-off between system flexibility and user satisfaction, which includes factors such as wait times and the need for customer acceptance of transfer operations, is an inevitable consideration. However, the notable benefits of transfer capability should not be overlooked. The role of transfer capability in situations with limited travel distance and time for vehicles, as discussed by Rais et al. (2014), is of considerable importance. Shiri, Rahmani and Bafruei (2019) has also shown that the travel cost saved by transfers can be proportional to the square root of the number of requests.

Shiri, Rahmani and Bafruei (2019) indicates that, although the benefit from the transfer operation depends on the objective functions of the particular problem, it increases the rate of feasible solutions in instances containing the difficulty conditions of the problem. Using the model proposed by Rais et al. (2014), they show that in addition to reducing the total distance, delay time, and vehicles that must be used, there is a significant improvement in the number of solvable instances (feasibility ratio) in the cases where vehicle capacity is reduced, the time windows of requests are short, or especially in the case of transportation with heterogeneous vehicles and vehicles with time windows, where the fleet without transfer is unable to complete long distance requests or requests that do not align entirely with a vehicle route.

2.3. Short literature review

Author	Year	Type	Object	Objective function	Solution method
Cortés et al.	2010	PDPTWT	customers	Minimizing the total ride time spent by vehicles	Branch-and-cut algorithm method with Benders Decomposition
Rais et al.	2014	PDPT;PDPTWT	goods	Minimizing the total vehicle travel distances related to the cost factors of the vehicles.	Simplex method combined with branch-and-cut and branch-and-bound techniques
Afonso Sampaio; Martin Savelsbergh; Lucas P. Veelenturf; Tom Van Woensel	2020	MPDPTWT	freight and passenger delivery in urban environment	Minimizing the travel cost (distance) of vehicles	Large neighborhood search
David Wolfinger; Juan-José Salazar-González	2021	PDPSLT	customer requests	Minimizing the travel costs and transshipment costs	Branch-and-cut
Stefan Voigt and Heinrich Kuhn	2021	PDPTOD	parcels and goods	Minimizing the costs from regular drivers and compensation for occasional drivers	Adaptive large neighborhood search
Zefeng Lyu, Andrew Junfang Yu	2023	PDPT;PDPTWT	goods	Minimizing the detour taken by the vehicles	Commercial optimization solver (Gurobi)
Cansu Agrali, Seokcheon Lee	2023	MPDPCTTW	goods	Minimizing the travel cost (distance) of vehicles	Combination of Simulated Annealing (SA) and Large Neighborhood Search (LNS)

Table 1: Literature review

In Table 1, we see the diversity in the targets and the objective functions of the models studied for the PDP(TW)T with other extensions. This also demonstrates that the PDPT can be applied to a wide range of practical issues, is utilized in numerous fields, and is gaining increasing interest.

Studies also show that problems can be solved using a variety of methods, such as extract methods or hybrid metaheuristics. While metaheuristics can solve large and complex instances, they do not guarantee an optimal result, as in the case of extract methods, the methods used by the two models concentrated in this thesis.

3. Comparison of two mathematical formulations

In this section, we will compare the main differences between the model proposed by Cortés et al. (2010) and the model suggested by Rais et al. (2014). Both are Mixed Integer Programming (MIP) models and can be represented as graphs $\boldsymbol{G} = (\boldsymbol{V},\ \boldsymbol{A})$, which consist of a set of vertices (or nodes) V and set of arcs A. However, there are differences between how the graphs are constructed in the two models, which will be discussed in more detail in Section 3.1.

3.1. Table of unified parameters

To make it easier to compare, we unify the syntax used in both models, which is shown in Table 2. However, most of the syntax is derived from the formulation proposed by Rais et al. (2014).

	Cortés et al. (2010)	**Rais et al. (2014)**
Set of vehicles	$\boldsymbol{K} = \{\boldsymbol{1}, \dots, \lvert\boldsymbol{K}\rvert\}$	
Set of requests	$\boldsymbol{R} = \{\boldsymbol{1}, \dots, \lvert\boldsymbol{R}\rvert\}$	
Set of transfer points	$\boldsymbol{T} = \{\boldsymbol{1}, \dots, \lvert\boldsymbol{T}\rvert\}$	
Set of origin depots for vehicles	$\boldsymbol{O} = \{\boldsymbol{o}(\boldsymbol{k}): \boldsymbol{k} \in \boldsymbol{K}\}$	
Set of destination depots for vehicles	$\boldsymbol{O}' = \{\boldsymbol{o}'(\boldsymbol{k}): \boldsymbol{k} \in \boldsymbol{K}\}$	
Set of pickup nodes for requests	$\boldsymbol{P} = \{\boldsymbol{p}(\boldsymbol{r}): \boldsymbol{r} \in \boldsymbol{R}\}$	
Set of delivery nodes for requests	$\boldsymbol{D} = \{\boldsymbol{d}(\boldsymbol{r}): \boldsymbol{r} \in \boldsymbol{R}\}$	
Set of nodes associated with requests	$\boldsymbol{N} = \boldsymbol{P} \cup \boldsymbol{D}$	
Set of start nodes of transfers	$\boldsymbol{s}(\boldsymbol{T}) = \{\boldsymbol{s}(\boldsymbol{t}): \boldsymbol{t} \in \boldsymbol{T}\}$	-
Set of finish nodes of transfers	$\boldsymbol{f}(\boldsymbol{T}) = \{\boldsymbol{f}(\boldsymbol{t}): \boldsymbol{t} \in \boldsymbol{T}\}$	-
Set of nodes	$\boldsymbol{V} = \boldsymbol{O} \cup \boldsymbol{O}' \cup \boldsymbol{N} \cup \boldsymbol{s}(\boldsymbol{T}) \cup \boldsymbol{f}(\boldsymbol{T})$	$\boldsymbol{V} = \boldsymbol{O} \cup \boldsymbol{O}' \cup \boldsymbol{N} \cup \boldsymbol{T}$
Set of arcs	$\boldsymbol{A} = \boldsymbol{A}_1 \cup \mathbf{A}_2 \cup \boldsymbol{A}_3 \cup \boldsymbol{A}_4 \cup \boldsymbol{A}_5$	$\boldsymbol{A} = \{(\boldsymbol{i}, \boldsymbol{j}): \boldsymbol{i}, \boldsymbol{j} \in \boldsymbol{N}, \boldsymbol{i} \neq \boldsymbol{j}\}$
Size of request $\boldsymbol{r} \in \boldsymbol{R}$	$\boldsymbol{q}_{\boldsymbol{r}}$	
Capacity of vehicle $\boldsymbol{k} \in \boldsymbol{K}$	$\boldsymbol{u}_{\boldsymbol{k}}$	

Minimum ride time from node i to node j	$\boldsymbol{\tau_{ij}}$
Time-window for node $\boldsymbol{i}$	$[\boldsymbol{E_i},\ \boldsymbol{L_i}]$

Table 2: List of unified parameters

While in the model proposed by Rais et al. (2014), each transfer point is represented by only one node in the graph, in the model proposed by Cortés et al. (2010), each transfer point is represented by two separate nodes (a start node and a finish node). In Section 3.4, this difference will be presented and analyzed more thoroughly. The differences in the representation of these transfer points will result in the graph for the model proposed by Cortés et al. (2010) having more vertices.

One important point to note about the set of arcs is that in the model of Rais et al. (2014), they present a complete graph, which means that there is always an arc connecting the two vertices in the graph. However, in the model of Cortés et al. (2010), they exclude the paths that are infeasible from the model:

- $\boldsymbol{A_1} = \boldsymbol{O} \times \left(\boldsymbol{P} \cup \boldsymbol{s}(\boldsymbol{T})\right) \cup \{(\boldsymbol{o}\{\boldsymbol{k}\}, \boldsymbol{o}'\{\boldsymbol{k}\}): \boldsymbol{k} \in \boldsymbol{K}\}$ corresponds to the feasible arcs starting from the origin depots of vehicles.
- $\boldsymbol{A_2} = (\boldsymbol{P} \times \left(\boldsymbol{N} \cup \boldsymbol{s}(\boldsymbol{T})\right))\backslash\{(\boldsymbol{p}\{\boldsymbol{r}\}, \boldsymbol{p}\{\boldsymbol{r}\}): \boldsymbol{r} \in \boldsymbol{R}\}$ corresponds to the feasible arcs starting from the pickup nodes.
- $\boldsymbol{A_3} = \left(\boldsymbol{D} \times (\boldsymbol{N} \cup \boldsymbol{s}(\boldsymbol{T}) \cup \boldsymbol{O}')\right)\backslash \{\left(\boldsymbol{d}(\boldsymbol{r}), \boldsymbol{p}(\boldsymbol{r})\right), (\boldsymbol{d}(\boldsymbol{r}), \boldsymbol{d}(\boldsymbol{r})): \boldsymbol{r} \in \boldsymbol{R}\}$ corresponds to the feasible arcs starting from the delivery nodes.
- $\boldsymbol{A_4} = \{(\boldsymbol{s}\{\boldsymbol{t}\}, \boldsymbol{f}\{\boldsymbol{t}\}): \boldsymbol{t} \in \boldsymbol{T}\}$ corresponds to the feasible arcs between start transfer nodes and finish transfer nodes.
- $\boldsymbol{A_5} = (\boldsymbol{f}(\boldsymbol{T}) \times \left(\boldsymbol{N} \cup \boldsymbol{O}' \cup \boldsymbol{s}(\boldsymbol{T})\right))\backslash\{\boldsymbol{f}\{\boldsymbol{t}\}, \boldsymbol{s}\{\boldsymbol{t}\}: \boldsymbol{t} \in \boldsymbol{T}\}$ corresponds to the feasible arcs ending at destination depots of vehicles.

The removal of the infeasible arcs reduces the actual number of variables and the number of constraints that need to be taken into account when solving the problem, contributing to the difference in performance between the two models.

3.2. Comparison of decision variables

	Cortés et al. (2010)	**Rais et al. (2014)**
Flow of vehicles	$x_{ij}^{k}\ \forall k \in K,\ \forall (i,j) \in A$	
Flow of requests	$z_i^{kr}\ \forall k \in K,\ \forall r \in R,\ \forall i \in V$	$y_{ij}^{kr}\ \forall k \in K,\ \forall r \in R,\ \forall (i,j) \in A$
Synchronization at transfer points	-	$s_{tr}^{k1k2}\ \forall k1,\ k2 \in K,\ \forall t \in T,\ \forall r \in R$
Arrival time at node $\boldsymbol{i}$	$a_i;\ a_{s(t)}^{k};\ a_{f(t)}^{k}\ \forall k \in K,\ \forall t \in T$	$a_i^k\ \forall k \in K,\ \forall i \in V$
Departure time at node $\boldsymbol{i}$	-	$b_i^k\ \forall k \in K,\ \forall i \in V$

Table 3: Decision variables

Both models use variable $\boldsymbol{x_{ij}^k}$, to represent the flow of vehicles. This variable indicates whether vehicle k travels from node i to node j or not. In order to relax the coupling constraints in the PDPT, it becomes necessary to introduce a new variable to track the states of requests. The Cortés et al. (2010) model tracks the position of the request by using variable z_i^{kr} to show whether or not request k is being carried by vehicle k when it arrives at node i. In the Rais et al. (2014) model, the flows of requests are tracked by using the variable y to show whether or not vehicle k is carrying request r as it travels from node i to node j. In my perspective, it is an important difference between the two models, which consequently leads to other differences.

The Rais et al. (2014) model includes further variable s_{tr}^{k1k2} used as a logical counter for transfer operations, which indicates whether vehicle $k1$ will transmit request r to vehicle $k2$ at transfer point t.

There are variables a_i^k and b_i^k indicate the arrival and departure time of vehicle k at node i in model of Rais et al. (2014). In the model of Cortés et al. (2010), we don't have variable for the departure time but have two others for tracking the arrival time at start transfer node and finish transfer node. While the triangle inequality makes it possible to eliminate the k-index for the time variable at the other points, determining the arrival (and departure) of each vehicle in both models at the transfer point is important for synchronization between operations.

3.3. Comparison of Objective Function

In both models, they consider the single objective function that they want to minimize the total cost of the fleet of vehicles.

Cortés et al. (2010)	Rais et al. (2014)
$\boldsymbol{Minimize \sum_{k\in K} \sum_{(i,j)\in A} t_{ij}x_{ij}^k}$	$\boldsymbol{Minimize \sum_{k\in K} \sum_{(i,j)\in A} c_{ij}^k x_{ij}^k}$

Table 4: Objective Functions

The main objective function in the model proposed by Cortés et al. (2010) is minimizing the total ride time spent by the fleet of vehicles, since the request is intended for customers. In Rais et al. (2014), the cost is a combination of the distance traveled and the vehicle cost coefficient. When we assume that all vehicles have the same cost coefficient and that one unit of distance requires one unit of time to travel, the objective values calculated from both formulations are identical, despite the different primary goals.

Note that in the original model from Rais et al. (2014), in some cases, only a portion of the fleet of vehicles may be used, whereas in the proposed model from Cortés et al. (2010), all vehicles must be used, resulting in a difference in the final results of the two models because if a vehicle is not used, the distance between the origin and destination depots will not be considered in the objective function.

As mentioned in Lyu and Yu (2022), the use of all vehicles will help to avoid the situation where vehicles with origin and destination depots distant from each other will be restricted from use. It will have a good effect when the objective function is set to a minimum of detour. In this thesis, we will not consider this objective, however, we will use all vehicles in the model of Rais et al. (2014) to facilitate the comparison of objective values between models without losing generality.

3.4. Comparison of transfer points' setting

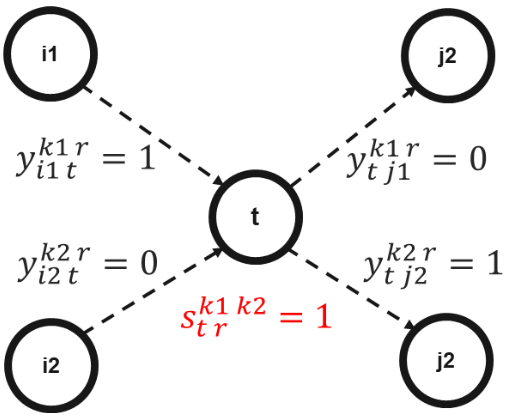

Figure 3: Transfer point in Rais et al. (2014)

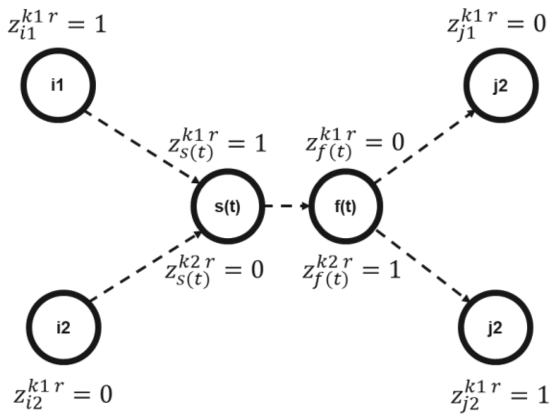

Figure 4: Transfer point in Cortés et al. (2010)

Two models use different set-ups of transfer nodes. In the model of Rais et al. (2014), as s_{tr}^{k1k2} variables are used for tracking the transfer operations between vehicles, it is sufficient to represent each transfer point using a single node.

In the model of Cortés et al. (2010), the authors do not use additional variables for the transfer operation. Consequently, each transfer point t is divided into two separate connected nodes (start node $s(t)$ and finish node $f(t)$). When the vehicle arrives at a transfer point, it can drop down requests at $s(t)$, then move to $f(t)$ to (possibly) pick up requests, this means that only one operation (load or unload) is performed with the request at $s(t)$ or $f(t)$. By examining the presence of the request (in which vehicle) at nodes $s(t)$ and $f(t)$, it is possible to determine the operations that occurred at t. Additionally, this configuration is helpful for the formation of constraints related to transfer points in the formulation.

3.5. Comparison of Constraints

According to Cortés et al. (2010), the following statements must be retained for any feasible solutions of PDPT:

(1) Every vehicle takes a route, starting at its origin depot and ending at its destination depot (without cycles).
(2) Each pickup and delivery location must be visited correctly once, which means that all requests must be completed.
(3) For each request, the pickup location must be visited before the delivery location.
(4) If vehicle $k1$ transfers request r to vehicle $k2$ at transfer point t, it has to reach this transfer point before vehicle $k2$ leaves t.
(5) At a time, the total loads of requests carried by a vehicle must not exceed the capacity of this vehicle.

Because this thesis considers time-window constraints, an additional statement must hold:

(6) All nodes in the model must be visited within their time-window.

To compare the constraints used in both models, they are grouped into different groups for their own purposes (note that some constraints may be repeated in different groups). In this thesis, we will not mention the constraints used to define decision variables.

Cortés et al. (2010)		Rais et al. (2014)	
$\sum_{(o(k),j)\in A} x^k_{o(k)j} = 1 \;\; \forall k \in K$	(C1)	$\sum_{(i,j)\in A} x^k_{ij} \leq 1 \;\; \forall k \in K,\, \forall i = o(k)$	(R1)
$\sum_{(i,o\prime(k))\in A} x^k_{io\prime(k)} = 1 \;\; \forall k \in K$	(C2)	$\sum_{(i,j)\in A} x^k_{ij} = \sum_{(j,l)\in A} x^k_{jl} \;\; \forall k \in K,\, \forall i \in o(k), \forall l \in o'(k)$	(R2)
$\sum_{i\in O} \sum_{(i,j)\in A} x^k_{ij} \leq 1 \;\; \forall k \in K$	(C3)		

The group of constraints above maintains flows that go in and out of the origin and destination depots of vehicles. They restrict that a vehicle must start the route at its origin and finish the route at its destination depot. In the model proposed by Cortés et al. (2010), all vehicles must be used. On the other hand, we don't have to use all of the vehicles in the model proposed by Rais et al. (2014), which may lead to a difference in objective values between the two formulations. The constraint set (C3) is used to ensure that a vehicle can only start from its origin depot and not from others.

$\sum_{(i,j)\in A} x^k_{ij} = \sum_{(j,i)\in A} x^k_{ji} \;\; \forall k \in K,\, \forall i \in N$	(C4)		
$\sum_{(i,\,s(t))\in A} x^k_{is(t)} = x^k_{s(t)f(t)} \;\; \forall k \in K,\, \forall t \in T$	(C5)	$\sum_{(i,j)\in A} x^k_{ij} = \sum_{(j,i)\in A} x^k_{ji} \;\; \forall k \in K,\, \forall i \in V \backslash \{o(k),\, o'(k)\}$	(R3)
$\sum_{(f(t),j)\in A} x^k_{f(t)j} = x^k_{s(t)f(t)} \;\; \forall k \in K,\, \forall t \in T$	(C6)		

These constraints ensure that, except for the origin and destination depots, if a vehicle visits a node, it must go out of this node (flow conservation constraints). In the model proposed by Cortés et al. (2010), as we use two nodes to represent a transfer location and there is only one arc connecting these two nodes, we need two additional constraint sets (C5) and (C6) exclusively for these nodes.

$$\sum_{k\in K}\sum_{(i,d(r))\in A} x_{id(r)}^{k} = 1 \ \forall r \in R \quad \text{(C7)}$$

$$\sum_{k\in K}\sum_{(p(r),j)\in A} x_{p(r)j}^{k} = 1 \ \forall r \in R \quad \text{(C8)}$$

$$\sum_{k\in K}\sum_{(i,j)\in A} y_{ij}^{kr} = 1 \ \forall r \in R, \forall i \in P \quad \text{(R4)}$$

$$\sum_{k\in K}\sum_{(i,j)\in A} y_{ij}^{kr} = 1 \ \forall r \in R, \forall j \in D \quad \text{(R5)}$$

$$y_{ij}^{kr} \le x_{ij}^{k} \quad \forall k \in K,\ \forall r \in R,\ \forall (i,j) \in A \quad \text{(R8)}$$

These sets of constraints ensure that each pickup node or delivery node is visited once and only once by a vehicle. In the model proposed by Rais et al. (2014), the y -variables will be used for this setting since they use the constraint set (R8) to synchronize vehicle flows and request flows. So, if y_{ij}^{kr} takes the value 1, the x_{ij}^{k} will also take the value 1.

$$x_{o(k)p(r)}^{k} = 1 \Rightarrow \tau_{o(k)p(r)} \le a_{p(r)} \quad \forall k \in K,\ \forall r \in R \quad \text{(C9)}$$

$$x_{o(k)s(t)}^{k} = 1 \Rightarrow \tau_{o(k)s(t)} \le a_{s(t)}^{k} \quad \forall k \in K,\ \forall r \in R \quad \text{(C10)}$$

$$x_{ij}^{k} = 1 => a_i + \tau_{ij} \le a_j \quad \forall k \in K,\ \forall (i,j) \in N^2 \cap A \quad \text{(C11)}$$

$$x_{is(t)}^{k} = 1 \Rightarrow a_i + \tau_{is(t)} \le a_{s(t)}^{k} \quad \forall k \in K,\ \forall i \in N,\ \forall t \in T \quad \text{(C12)}$$

$$x_{s(t)f(t)}^{k} = 1 \Rightarrow a_{s(t)}^{k} + \tau_{s(t)f(t)} \le a_{f(t)}^{k} \quad \forall k \in K,\ \forall t \in T \quad \text{(C13)}$$

$$x_{f(t)i}^{k} = 1 \Rightarrow a_{f(t)}^{k} + \tau_{f(t)i} \le a_i \quad \forall k \in K,\ \forall i \in N,\ \forall t \in T \quad \text{(C14)}$$

$$x_{f(t1)s(t2)}^{k} = 1 \Rightarrow a_{f(t1)}^{k} + \tau_{f(t1)s(t2)} \le a_{s(t2)}^{k} \quad \forall k \in K,\ \forall t1, t2 \in T,\ t1 \ne t2 \quad \text{(C15)}$$

$$b_i^k + \tau_{ij} - a_j^k \le M\left(1 - x_{ij}^{k}\right) \quad \forall k \in K,\ \forall (i,j) \in A \quad \text{(R15)}$$

$$a_i^k \le b_i^k \ \ \forall k \in K,\ \forall i \in V \quad \text{(R16)}$$

Here we have constraints associated with arrival (and departure) time at nodes, which are used to remove sub-tours (cycles) from the model, ensuring that a vehicle will not come back to a node after visiting this node. We have more sets of constraints in the model proposed by Cortés et al. (2010) mostly because of the configuration of transfer points. It is mentioned in Cortés et al. (2010) that to remove zero-length cycles from the model in case where different nodes are located in the same coordinate, a small positive number can be added to τ when it takes the value 0.

In my opinion, there is a typo in constraint set (C10), we should replace $\forall r \in R$ with $\forall t \in T$. This set of constraints can be formulated as:

(C10) $x^k_{o(k)s(t)} = 1 \Rightarrow \tau_{o(k)s(t)} \leq a^k_{s(t)}\ \forall k \in K,\ \forall t \in T$

Alternatively, constraint sets (C9) -(C15) can be expressed as linear expressions using the Big-M technique:

(C9) $\tau_{o(k)p(r)} - a_{p(r)} \leq M\big(1 - x^k_{o(k)p(r)}\big)\ \forall k \in K,\ \forall r \in R$
(C10) $\tau_{o(k)s(t)} - a^k_{s(t)} \leq M\big(1 - x^k_{o(k)s(t)}\big)\ \forall k \in K,\ \forall t \in T$
(C11) $a_i + \tau_{ij} - a_j \leq M\big(1 - x^k_{ij}\big)\ \forall k \in K,\ \forall (i,j) \in N^2 \cap A$
(C12) $a_i + \tau_{is(t)} - a^k_{s(t)} \leq M(1 - x^k_{is(t)})\ \forall k \in K,\ \forall i \in N,\ \forall t \in T$
(C13) $a^k_{s(t)} + \tau_{s(t)f(t)} - a^k_{f(t)} \leq M\big(1 - x^k_{s(t)f(t)}\big)\ \forall k \in K,\ \forall t \in T$
(C14) $a^k_{f(t)} + \tau_{f(t)i} - a_i \leq M(1 - x^k_{f(t)i})\ \forall k \in K,\ \forall i \in N,\ \forall t \in T$
(C15) $a^k_{f(t1)} + \tau_{f(t1)s(t2)} - a^k_{s(t2)} \leq M(1 - x^k_{f(t1)s(t2)})\ \forall k \in K,\ \forall t1, t2 \in T,\ t1 \neq t2$

For optimization purposes, we try to keep the value of M as small as possible. In a constraint (or set of constraints), M can take a maximum value between 0 and the maximum value of the left-hand expression. For example, in the set of constraints (C11), we can adjust $M = \max\big(0,\ E_i + \tau_{ij} - L_j\big)$.

Table 5 Constraints for the flows of vehicles

Constraints in Table 5 are used to represent the flows of vehicles. With these constraints, statements (1) and (2) are ensured.

Cortés et al. (2010)	**Rais et al. (2014)**
$x^k_{ij} = 1 \Rightarrow z^{kr}_i = z^{kr}_j\ \ \forall k \in K, \forall r \in R,$ $\forall (i,j) \in E \backslash \{(s(t), f(t)) : t \in T\},$ $i \notin \{p(r), d(r)\}$ (C17)	$\sum_{(i,j)\in A} y^{kr}_{ij} - \sum_{(j,i)\in A} y^{kr}_{ji} = 0$ $\forall k \in K, \forall r \in R, \forall i \in V \backslash T$ (R7)

Lyu and Yu (2022) proposed a revision to the constraints set (R7). They removed the origin and destination depots of the requests from these constraints. Otherwise, the model proposed by Rais et al. (2014) becomes infeasible because the vehicle can neither carry a request to its pickup location nor carry requests out of its delivery location. They edited the set of constraints (R7) to be:

(R7) $\sum_{(i,j)\in A} y^{kr}_{ij} - \sum_{(j,i)\in A} y^{kr}_{ji} = 0\ \ \forall k \in K, \forall r \in R, \forall i \in V \backslash \{T \cup \{p(r), d(r)\}\}$

While the constraint set (C17) establishes that requests will not be dropped off in transit, the constraint set (R7) ensures that requests will not be picked up or left in non-transshipment nodes other than the pickup and delivery locations of It. Both are used for maintaining the continuity of request flows.

As all variables used are binary, constraint set (C17) can be expressed as linear expressions:

(C17) $z^{kr}_i - z^{kr}_j \leq 1 - x^k_{ij}\ \ \forall k \in K,\ \forall r \in R,\ \forall (i,j) \in E \backslash \{(s(t), f(t)) : t \in T\},\ i \notin \{p(r), d(r)\}$

$x_{p(r)i}^{k} = 1 \Rightarrow z_i^{kr} = 1 \quad \forall k \in K, \forall r \in R,\ \forall (p(r), i) \in A$	(C18)	$\sum_{k\in K} \sum_{(i,j)\in A} y_{ij}^{kr} = 1 \ \ \forall r \in R, \forall i \in P$	(R4)
$x_{d(r)i}^{k} = 1 \Rightarrow z_i^{kr} = 0 \quad \forall k \in K,\ \forall r \in R,\ \forall (d(r), i) \in A$	(C19)	$\sum_{k\in K} \sum_{(i,j)\in A} y_{ij}^{kr} = 1 \ \ \forall r \in R, \forall j \in D$	(R5)

These constraints are used to maintain the process of picking up or delivering requests when the vehicle arrives at their pickup or delivery locations, ensuring that a request can only be picked up or delivered by exactly one vehicle at these nodes.

Furthermore, the constraint sets (C18) and (C19) can also be expressed as linear expressions:

(C18) $z_i^{kr} - 1 \leq 1 - x_{p(r)i}^{k} \ \ \forall k \in K,\ \forall r \in R,\ \forall (p(r), i) \in A$

(C19) $z_i^{kr} \leq 1 - x_{d(r)i}^{k} \ \ \forall k \in K,\ \forall r \in R,\ \forall (d(r), i) \in A$

$\sum_{k\in K} z_{s(t)}^{kr} - \sum_{k\in K} z_{f(t)}^{kr} = 0 \ \ \forall t \in T,\ \forall r \in R$	(C20)	$\sum_{k\in K} \sum_{(i,j)\in A} y_{ij}^{kr} - \sum_{k\in K} \sum_{(j,i)\in A} y_{ji}^{kr} = 0 \quad \forall r \in R, \forall t \in T$	(R6)

These constraints (C20) and (R6) represent the relaxation of coupling constraints in PDP. In both models, if a request arrives at a transfer location, it must leave the transfer location in the same or other vehicles.

$z_{o(k)}^{kr} = z_{o'(k)}^{kr} = 0 \ \ \forall k \in K,\ \forall r \in R$	(C16)	$y_{ij}^{kr} \leq x_{ij}^{k} \quad \forall k \in K,\ \forall r \in R,\ \forall (i,j) \in A$	(R8)
$\sum_{(i,j)\in A} x_{ij}^{k} = 0 \Rightarrow \sum_{r\in R} z_j^{kr} \leq 0 \ \forall k \in K,\ \forall j \in V\{o(k), o'(k)\}$	(C21)		

Constraint set (C16) ensures that vehicles do not carry any requests when they are present at their origin or destination depots. Constraint set (C21) enforces that, if vehicle k does not reach node j, then z_j^{kr} will take value of 0 for any $r \in R$ (implicitly indicating that if $z_j^{kr} = 1$ is given, vehicle k has to visit node j). The combination of these two constraint sets (C16) and (C21) represents the synchronization between vehicle flows and request flows, aiming at the same purpose as constraints (R8) as mentioned earlier.

Constraints (C21) can also be expressed as linear expressions:

(C21) $\sum_{r\in R} z_j^{kr} \leq \sum_{(i,j)\in A} x_{ij}^{k} \ \ \forall k \in K,\ \forall j \in V\{o(k), o'(k)\}$

Table 6: Constraints for tracking of requests

Constraints in Table 6 are used to track the location (or flows) of requests and how they relate to flows of vehicles. This is important in PDPT because a request can be carried by many different vehicles during transport. So far, statement (3) still doesn't hold if a request is transported by more than one vehicle, so we need constraints to assure the transfer operations between vehicles at transfer points. Therefore, the constraints in Table 7 are required to ensure that statements (3)-(6) are fulfilled.

Cortés et al. (2010)		Rais et al. (2014)	
$z_{s(t)}^{k1r} + z_{f(t)}^{k2r} = 2 => a_{s(t)}^{k1} + \Delta \leq a_{f(t)}^{k2}$ $\forall t \in T,\ \forall k1, k2 \in K,\ \forall r \in R$	(C22)	$\sum_{(j,t)\in A} y_{jt}^{k1r} + \sum_{(t,j)\in A} y_{tj}^{k2r} \leq 1 + s_{tr}^{k1k2}$ $\forall r \in R,\ \forall t \in T,\ \forall k1,\ k2 \in K,\ k1 \neq k2$	(R19)
		$a_t^{k1} - b_t^{k2} \leq M\left(1 - s_{tr}^{k1k2}\right)$ $\forall r \in R,,\forall t \in T\ k1, k2 \in K,\ k1 \neq k2$	(R20)

Constraints (R19) and (R20), or constraints (C22) are used to assure proper synchronization of transfer operations at transfer points. Constraint sets (C22) and (R19) have similar functionality, ensuring the precedence that if request r is transferred from vehicle $k1$ to vehicle $k2$, vehicle $k1$ must reach the transfer point before vehicle $k2$ leaves it. Parameter Δ in (C22) can be expressed as the amount of time it takes to transfer requests between two vehicles.

Constraints (C22) can be indicated in form of linear expressions:

(C22) $a_{s(t)}^{k1} + \Delta - a_{f(t)}^{k2} \leq 2 - \left(z_{s(t)}^{k1r} + z_{f(t)}^{k2r}\right)\ \forall t \in T,\ \forall k1, k2 \in K,\ \forall r \in R$

Constraint set (R19) uses s -variables as logical counters, indicating when requests are transferred between two vehicles. However, the variable s_{tr}^{k1k2} can take a false positive value, which means that $s_{tr}^{k1k2} = 1$ even if r is not transferred from vehicle $k1$ to vehicle $k2$ at the transfer point t. To solve this problem, I propose an additional constraint set, which in combination with constraints (R19) can ensure the value of s-variables corresponds to the transfer operations:

(T1) $\sum_{(j,t)\in A} y_{jt}^{k1r} + \sum_{(t,j)\in A} y_{tj}^{k2r} \geq 2 s_{tr}^{k1k2}\ \ \forall r \in R,\ \forall t \in T,\ \forall k1,\ k2\ \in K,\ k1 \neq k2$

$\sum_{r\in R} q_r z_i^{kr} \leq u_k\ \ \forall k \in K,\ \forall i \in V$	(C23)	$\sum_{r\in R} q_r y_{ij}^{kr} \leq u_k x_{ij}^k\ \ \forall k \in K,\ \forall (i,j) \in A$	(R9)

Constraint sets (C23) and (R9) keep vehicles from carrying requests exceeding their capacities. In the model proposed by Cortés et al. (2010), the capacity constraint is checked at each node, which is consistent with the use of z -variables. Rais et al. (2010) ensure capacity constraints while vehicles are moving with the use of flow variable y.

	$a_i^k \leq b_i^k \ \ \forall k \in K, \forall i \in V$ (R16)
$E_i \leq a_i \leq L_i \ \ \forall i \in V$ (C24)	$E_{p(r)} \leq \ a_{p(r)}^k, b_{p(r)}^k \leq L_{p(r)} \ \forall r \in R, \forall k \in K$ (R17)
	$E_{d(r)} \leq \ a_{d(r)}^k, b_{d(r)}^k \leq L_{d(r)} \ \forall r \in R, \forall k \in K$ (R18)
These constraints maintain the time-window constraint for requests. In this thesis, we consider the hard time windows, so no violation is accepted.	

Table 7: Constraints for synchronization of transfer, capacity and time-windows

3.6. Number of variables and constraints.

Based on the size of the problem instance, in the model proposed by Cortés et al. (2010), there are a total of $|K||A|$ x-variables, $|K||R||V|$ z -variable, $|V|$ a^i, $|K||T|$ $a_{s(t)}^k$, and $|K||T|$ $a_{f(t)}^k$. $|R|$ is bounded by $|N^2|$. In the worst case, each request is satisfied by a separate vehicle, so $|K|$ can be assumed to be $O(|R|)$. With $|V| = n$, the total number of variables is bounded by $O(n^5)$. Cortés et al. (2010) also pointed out that the number of z -variables can also be reduced to bounded by $O(n^4)$ by relaxing these variables to take real values in the range $[0, 1]$, generally lead to improved computational performance. This may reduce the total number of variables in the model to bounded by $O(n^4)$.

About the total number of constraints in Cortés et al. (2010): The constraint set (24) is bounded by $O(n)$. Constraints (3), (7) and (8) are bounded by $O(n^2)$. Constraints (1), (2), (4)-(6), (13), (21), and (23) are bounded by $O(n^3)$. Constraints (9)-(12), (14)-(16), and (20) are bounded by $O(n^4)$. Constraints (18) and (19) are bounded by $O(n^5)$. The constraint set (17) is bounded by $O(n^6)$ and the constraints set (22) is bounced by $O(n^7)$.

In the model proposed by Rais et al. (2014), there are a total of $|K||A|$ x-variables, $|K||R||A|$ y -variable, $|K^2||T||R|$ s-variables, $|K||V|$ a_i^k , and $|K||V|$ b_i^k. As $|T|$ is bounded by $|V|$, the total number of variables is bounded by $O(n^7)$.

In this model, constraints (1), (3)-(6), and (16) are bounded by $O(n^3)$. Constraints (2), (8), (9) (15), (17), and (18) are bounded by $O(n^4)$. The constraint set (7) is bounded by $O(n^5)$. Constraints (19) and (20) are bounced by $O(n^7)$.

Although there is a difference between the number of variables used in the two formulations, as outlined in Oliver and Yajaira (2022), we cannot directly conclude that the formulation proposed by Cortés et al. (2010) will give better results than the formulations of Rais et al. (2014).

4. Techniques for improving performance of models

In order to improve the computing time of MILP formulations, several techniques can be applied. Lyu and Yu (2023) and Morrison, Jacobson, Sauppe, and Sewell (2016) have shown that redundant constraints (valid inequalities) can be added to improve the performance of the model by tightening the feasible region without removing integer solutions. They are inequalities that hold true for all feasible solutions of the optimization problem. As we tighten the LP relaxation, we may get a better lower bound, which can help to prune solutions whose lower bound is no better than the incumbent's solution value quicker. These redundant constraints are also valid cuts for a branch-and-cut algorithm.

According to Liberti (2012) and Cortes et al. (2010), symmetry breaking constraints (SBCs) can be used to speed up the solution process. Solution symmetries are one type of symmetry covered in this thesis, which means that having some symmetries of variables can result in a solution with similar values. Once these symmetries are identified, we can add constraints to the model to make some of the symmetric solutions infeasible, thereby reducing the solution space and resulting in faster convergence to the optimal solution.

4.1. Valid inequalities proposed by Lyu and Yu (2022)

In order to improve the performance (reducing the computing time) of the formulation proposed by Rais et al. (2014), Lyu and Yu (2022) put forward some redundant constraints, contributing to tightening the relaxation of the MILP models. The main idea of most of these constraints is to remove infeasible arcs from considering graphs. Additionally, some constraint sets share the idea with some of the constraints in the model proposed by Cortés et al. (2010).

(LY1) $\sum_{(j,i)\in A} x_{ji}^k = 0 \;\; \forall k \in K, i = o(k)$

(LY2) $\sum_{(i,j)\in A} x_{ij}^k = 0 \;\; \forall k \in K, \forall i \in O \cup O', i \neq o(k)$

(LY3) $\sum_{(j,i)\in A} x_{ji}^k = 1 \;\; \forall k \in K, i = o'(k)$

(LY4) $\sum_{(i,j)\in A} x_{ij}^k = 0 \;\; \forall k \in K, i = o'(k)$

(LY5) $\sum_{(i,j)\in A} x_{ij}^k \leq 1 \;\; \forall k \in K, i \in T$

(LY6) $\sum_{(i,j)\in A} \sum_{k\in K} x_{ij}^k = 1 \;\; \forall i \in P \cup D$

(LY7) $\sum_{(i,j)\in A} \sum_{k\in K} y_{ij}^{kr} = 0 \;\; \forall r \in R, j = p(r)$

(LY8) $\sum_{(i,j)\in A} y_{ij}^{kr} = 0 \; \forall r \in R, \forall k \in K, \forall i \in O \cup O', i \neq o(k), i \neq o'(k)$

Constraints (LY1) ensure that the vehicle cannot return to its origin depot. Constraint set (LY2) prevents a vehicle from starting a route from a location other than its origin depot, having the same idea as constraints (C3) with the combination of constraints (C1). Constraint set (LY3) can be compared to constraint set (C2), which ensures a vehicle arrives at its destination depot at the end of the route. Constraints (LY4) make sure that the vehicles do not leave the destination depots. Constraints (LY5) require that each vehicle only visit the same transfer station once. Constraints (LY6) require that pickup and delivery locations only be visited once, which is equivalent to constraints (C7) and (C8) in Cortés et al. (2010). Constraints (LY7) indicate that request flows must not include arcs that go to pickup locations, which were again dropped from the set of arcs A3 in Cortés et al. (2010). Constraint set (LY8) ensures that the request flows do not include either the origin or destination depots, the same purpose as constraints (C16).

Lyu and Yu (2022) pointed out that these redundant constraints (valid inequalities), especially constrains (LY1) -(LY4), could significantly improve the computing time for this model. However, they also mention that eliminating unnecessary arcs before the computation does not result in much improvement in performance.

Based on the observations of Lyu and Yu (2022) and the idea in Cortés et al. (2010), in the computational analysis section, we will assess the computing time of a model derived from Rais et al. (2014), in which we remove the infeasible arcs from the graph before the calculation process. By excluding them from the model, the solution space is reduced, making it potentially faster to find optimal solution.

The set of arcs $\boldsymbol{A} = \boldsymbol{A_1} \cup \boldsymbol{A_2} \cup \boldsymbol{A_3} \cup \boldsymbol{A_4}$ in this model are set up as follows:

- $\boldsymbol{A_1} = \boldsymbol{O} \times (\boldsymbol{P} \cup \boldsymbol{T}) \cup \{\boldsymbol{o}\{\boldsymbol{k}\}, \boldsymbol{o'}\{\boldsymbol{k}\}: \boldsymbol{k} \in \boldsymbol{K}\}$
- $\boldsymbol{A_2} = (\boldsymbol{P} \times (\boldsymbol{N} \cup \boldsymbol{T})) \backslash \{(\boldsymbol{p}\{\boldsymbol{r}\}, \boldsymbol{p}\{\boldsymbol{r}\}): \boldsymbol{r} \in \boldsymbol{R}\}$
- $\boldsymbol{A_3} = \big(\boldsymbol{D} \times (\boldsymbol{N} \cup \boldsymbol{T} \cup \boldsymbol{O'})\big) \backslash \{(\boldsymbol{d}(\boldsymbol{r}), \boldsymbol{p}(\boldsymbol{r})), (\boldsymbol{d}(\boldsymbol{r}), \boldsymbol{d}(\boldsymbol{r})): \boldsymbol{r} \in \boldsymbol{R}\}$
- $\boldsymbol{A_4} = \boldsymbol{T} \times (\boldsymbol{N} \cup \boldsymbol{O'} \cup \boldsymbol{T})$

4.2. Symmetries Breaking Constraints

Cortés et al. (2010) proposed a set of symmetry breaking constraints (SBCs) for vehicles when they are considered indistinguishable. Here, two vehicles can be indicated as indistinguishable if they have the same capacity and the same coordinate of origin (as well as destination) depots. In this case, permuting the numbering of the vehicles does not affect the objective values. Thus, defining an exact order, where vehicles are used in ascending numbering, can help eliminate some of the symmetric solutions. This set of constraints can be formulated as:

$$\sum_{k>r} \sum_{(p(r),j)\in A} x_{p(r)j}^{k} = 0, \forall r \in R$$

5. Computational results

In this section, we evaluate and compare the performance of the formulations proposed by Cortés et al. (2010) and Rais et al. (2014). In addition, we also test the effectiveness of the techniques used to improve the models mentioned in the thesis. These formulations are written in Python 3 and tested with a Python extension module called "gurobipy", running on an Intel(R) Xeon(R) Gold 6248R CPU @ 3.00GHz, 48 physical cores, 96 logical processors, using up to 32 threads.

To ensure the generalization of the experiments, we will use 50 instances of PDPT and 54 instances of PDPTWT, which were provided by Lyu and Yu (2022).

Instances for PDPT are considered in the case of shipping products. All nodes are distributed at random on a 100 x 100 Euclidean grid. The value of Euclidean distance is both the distance between two nodes and the time it takes to move between these nodes, which means it takes one unit of time to move one unit of distance. Each pickup and delivery node is associated with a positive load and a negative load, both of which are within the range [1, 100]. The vehicles have varied origin and destination depots, but their capacity is uniform and fixed at 100. These 50 instances are divided into five groups (each group contains 10 instances), increasing according to the size of the problem. The names of groups (and instances) will follow the following conventions: "R" represents the number of requests, "K" shows the number of vehicles, "T" means the number of transfer points, and "Q" represents the capacity of vehicles.

Instances for PDPTWT are considered with the idea for the customer shipping case. Here we have 120 x 120 Euclidean grids. The size of the request is fixed to 1 (q = 1 for pickup location and q = -1 for delivery location). The number of vehicles is also fixed at 4. The origin depot and destination depot of each vehicle are in the same position, and the capacity of all vehicles is fixed at 99. These 54 instances are divided into 18 groups (each group contains 3 instances), arranged by the size of the problem, the shift length of vehicles (240 min or 300 min), and the distance between the pickup and delivery locations of the requests (indicated by "L", "M", or "S"). "L" stands for long-distance requests where the distance between pickup and delivery locations of any request is always greater than 60 units; "S" indicates short-distance requests (less than 60 units); and "M" has both short and long-distance requests. The remaining name conventions are set similarly to the PDPT instances.

To test the effectiveness of symmetrical constraints to eliminate symmetry between vehicles, we edit the instances and place the positions of all the origin and destination depots of all vehicles at the center of the grid. So, all the vehicles can be considered indistinguishable.

The calculation results in this section will be given according to the average of each group of instances. In the appendix section, more detailed information about the computing results of each instance will be given.

5.1. Model of Cortés et al. (2010) vs Fixed model of Rais et al. (2014)

The table shows the performance comparison results of two models proposed by Cortés et al. (2010) and Rais et al. (2014). As noted in the previous sections, the model by Rais et al. (2014) has been adjusted to be feasible. In addition, every vehicle in the fleet will be used. This table construction is referenced from Lyu and Yu (2022), where columns #opt., and #lim., respectively, is the number of instances solved optimally and number of instances solved achieving feasible solutions but not the optimal ones. There is no instance where a feasible solution cannot be found. The column obj. captures the average objective values for each group, and the t(s) column represents the average computation time for each group of instances.

Instances' Group	Cortés et al. (2010)			Rais et al. (2014)		
	#opt.	obj.	t(s)	#opt	obj.	t(s)
PDPT-R5-K2-T1-Q100	10	442.439	1.906	10	442.439	43.107
PDPT-R5-K2-T2-Q100	10	415.540	3.154	10	415.540	47.260
PDPT-R5-K3-T3-Q100	10	421.411	6.238	10	421.411	212.139
PDPT-R7-K2-T1-Q100	10	525.011	24.354	8	526.845	1552.981
PDPT-R7-K3-T3-Q100	10	560.428	1028.742	7	567.954	2341.767
Average		472.966	212.879		474.838	839.451

PDPTWT-3R-4K-4T-240L	3	366.795	5.789	3	367.530	249.204
PDPTWT-3R-4K-4T-240M	3	256.353	1.703	3	256.353	108.718
PDPTWT-3R-4K-4T-240S	3	229.576	0.535	3	229.576	14.066
PDPTWT-3R-4K-4T-300L	3	366.795	5.827	3	366.795	230.241
PDPTWT-3R-4K-4T-300M	3	256.353	1.720	3	256.353	88.284
PDPTWT-3R-4K-4T-300S	3	229.576	0.562	3	229.576	7.731
PDPTWT-3R-4K-5T-240L	3	358.443	6.000	3	358.980	538.344
PDPTWT-3R-4K-5T-240M	3	256.353	2.365	3	256.353	124.784
PDPTWT-3R-4K-5T-240S	3	229.576	1.207	3	229.576	14.899
PDPTWT-3R-4K-5T-300L	3	358.443	6.297	3	358.443	217.253
PDPTWT-3R-4K-5T-300M	3	256.353	2.209	3	256.353	76.308
PDPTWT-3R-4K-5T-300S	3	229.576	1.115	3	229.576	7.367
PDPTWT-4R-4K-4T-240L	3	416.700	27.399	3	416.700	1866.456
PDPTWT-4R-4K-4T-240M	3	342.663	15.811	2	342.663	1264.964
PDPTWT-4R-4K-4T-240S	3	332.713	16.254	3	332.713	1049.480
PDPTWT-4R-4K-4T-300L	3	416.700	26.766	2	416.700	1639.987
PDPTWT-4R-4K-4T-300M	3	342.663	16.127	2	342.663	1257.398
PDPTWT-4R-4K-4T-300S	3	332.713	16.587	3	332.713	355.764
Average		309.908	8.571		309.979	506.180

Table 8: Results solved by models in Cortés et al. (2010) and Rais et al. (2014)

Based on this comparison, we see the superior computational speed of the model proposed by Cortés et al. (2010) over Rais et al. (2014), which is seen from the improvement in time. average calculation from 839.451 s to 212.879 s (a reduction of 74.65%) for PDPT-instances and from 506.180 s to 8.571 s for PDPTWT-instances (a reduction of 98.3%). This superiority also considers the difference in the ability to give optimal results between the two models in the time allowed. While the model in Cortés et al. (2010) can optimally solve all given instances, the model proposed by Rais et al. (2014) cannot guarantee optimality for 5 PDPT-instances and 3 PDPTWT-instances. However, the gaps between the optimal solutions and the feasible solutions found for the PDPTWT instances are very small, so we do not see a difference in the average objective values between the two models for these instances after rounding.

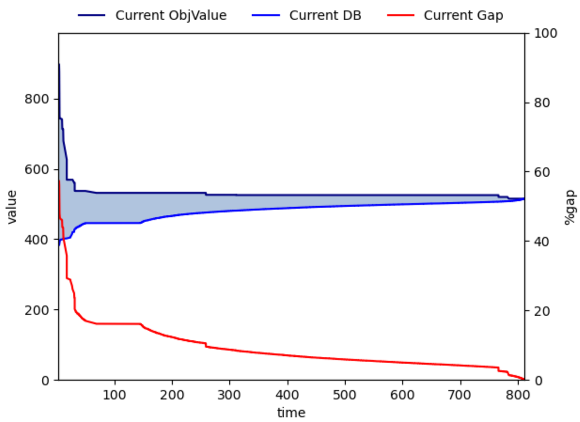

Figure 5: Gap and Time - Cortés et al. (2010) - PDPT

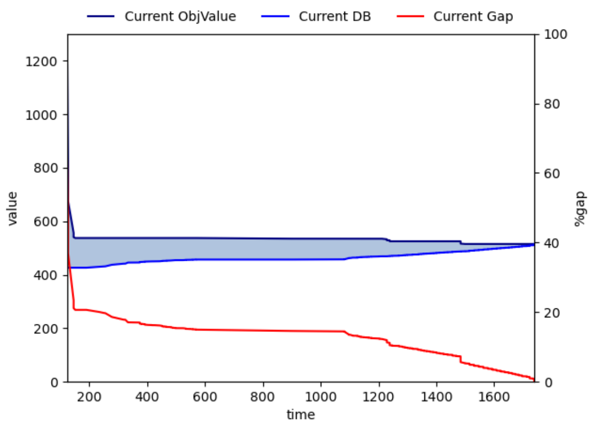

Figure 6: Gap and Time - Rais et al. (2014) - PDPT

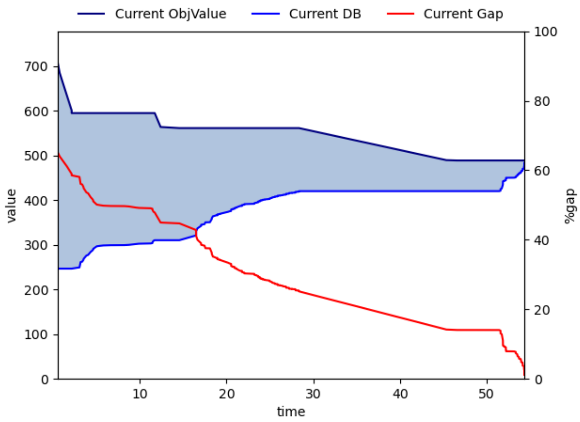

Figure 7: Gap and Time - Cortés et al. (2010) - PDPTWT

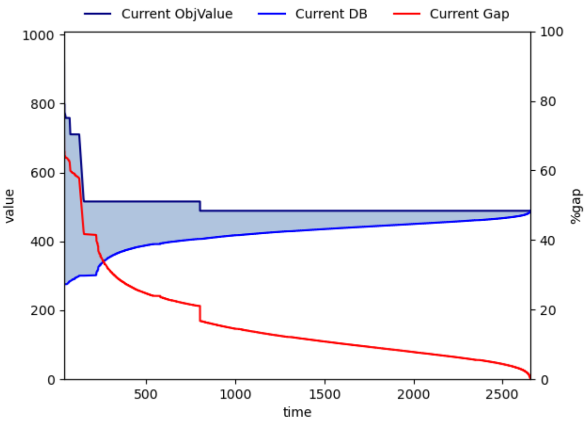

Figure 8: Gap and Time - Rais et al. (2014) - PDPTWT

The gap between current the best objective value (upper bound) and lower bound is expressed as a percentage with respect to runtime. The gap can be calculated using the formula:

$$gap(\%) = \frac{|Current\ Best\ Objective - Current\ Bound|}{|Current\ Best\ Objective|} \times 100$$

The instance representing the PDPT is *PDPT-R7-K3-T3-Q100-6* and the instance selected for the PDPTWT is *PDPTWT-4R-4K-4T-300L-2*. In these cases, the gaps narrow quickly at first (under 50%) and slow down over time. As we found a good feasible solution early, the rest of the time was mostly spent trying to get optimality by pushing the lower bound towards the objective value of the incumbent solution.

5.2. Effectiveness of improvement methods

We review the performance of the model proposed by Lyu and Yu (2022) and compare it with the model from Cortés et al. (2010) due to the many similarities in the changes and redundant constraints they have outlined. in comparison with Cortés et al. (2010)'s model. In addition to testing the effectiveness of removing infeasible arcs from the model, we add the calculation results of model proposed by Rais et al. (2014) without the infeasible arcs.

Instances' Group	Cortés et al. (2010)		Lyu and Yu (2022)		Rais et al. (2014) without infeasible arcs	
	obj.	t(s)	obj.	t(s)	obj.	t(s)
PDPT-R5-K2-T1-Q100	442.439	1.906	442.439	0.128	442.439	0.701
PDPT-R5-K2-T2-Q100	415.540	3.154	415.540	0.177	415.540	0.977
PDPT-R5-K3-T3-Q100	421.411	6.238	421.411	0.437	421.411	3.139
PDPT-R7-K2-T1-Q100	525.011	24.354	525.011	1.133	525.011	32.733
PDPT-R7-K3-T3-Q100	560.428	1028.742	560.428	1.556	560.428	215.187
Average	472.966	212.879	472.966	0.686	472.966	50.547
PDPTWT-3R-4K-4T-240L	366.795	5.789	366.795	2.749	366.795	4.201
PDPTWT-3R-4K-4T-240M	256.353	1.703	256.353	1.150	256.353	1.910
PDPTWT-3R-4K-4T-240S	229.576	0.535	229.576	0.417	229.576	1.025
PDPTWT-3R-4K-4T-300L	366.795	5.827	366.795	3.029	366.795	3.430

PDPTWT-3R-4K-4T-300M	256.353	1.720	256.353	1.144	256.353	1.473
PDPTWT-3R-4K-4T-300S	229.576	0.562	229.576	0.402	229.576	0.567
PDPTWT-3R-4K-5T-240L	358.443	6.000	358.980	4.903	358.443	5.268
PDPTWT-3R-4K-5T-240M	256.353	2.365	256.353	2.103	256.353	2.814
PDPTWT-3R-4K-5T-240S	229.576	1.207	229.576	0.422	229.576	0.880
PDPTWT-3R-4K-5T-300L	358.443	6.297	358.443	3.150	358.443	4.902
PDPTWT-3R-4K-5T-300M	256.353	2.209	256.353	1.205	256.353	2.100
PDPTWT-3R-4K-5T-300S	229.576	1.115	229.576	0.518	229.576	0.943
PDPTWT-4R-4K-4T-240L	416.700	27.399	416.700	11.569	416.700	28.579
PDPTWT-4R-4K-4T-240M	342.663	15.811	342.663	41.538	342.663	20.314
PDPTWT-4R-4K-4T-240S	332.713	16.254	332.713	9.801	332.713	15.297
PDPTWT-4R-4K-4T-300L	416.700	26.766	416.700	11.881	416.700	20.565
PDPTWT-4R-4K-4T-300M	342.663	16.127	342.663	25.727	342.663	25.320
PDPTWT-4R-4K-4T-300S	332.713	16.587	332.713	10.581	332.713	20.384
Average	309.908	8.571	309.908	7.349	309.908	8.887

Table 9: Results solved by models in Lyu and Yu (2022) and Rais et al. (2014) without infeasible arcs

All three models can solve given instances optimally. As for the PDPTW-instances, all three models show relatively close performance. In particular, removing infeasible arcs from the model of Rais et al. (2014) significantly improved computing time from 506.18 s to 8.887 s (a 98.24% improvement). Even for solving PDPT-instances, this model outperforms the model proposed by Cortés et al. (2010) (average computing time 50.547s vs 212.879) and has a 93.98% improvement over the original model, improving the model's ability to find feasible solutions. In addition, the model proposed by Lyu and Yu (2022) gives better results than the other two models for PDPT-instances, demonstrating the effectiveness of the added redundant constraints.

Figures 9-12 show the movement of the gap curve from the model of Lyu and Yu (2022) and model proposed by Rais et al., (2014) without infeasible arcs in solving instances *PDPT-R7-K3-T3-Q100-6* and *PDPTWT- 4R-4K-4T-300L-2* (same as in section 5.1). The gaps in these cases are also reduced quickly at the beginning. Notably, the model proposed by Lyu and Yu achieves better lower bounds at baseline by adding redundant constraints, speeding up the convergence process.

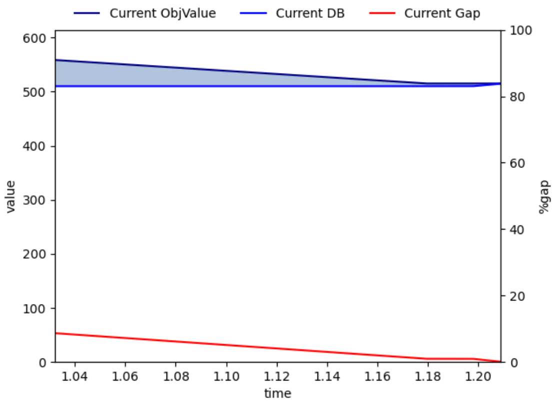

Figure 9: Gap and Time – Lyu and Yu (2023) - PDPT

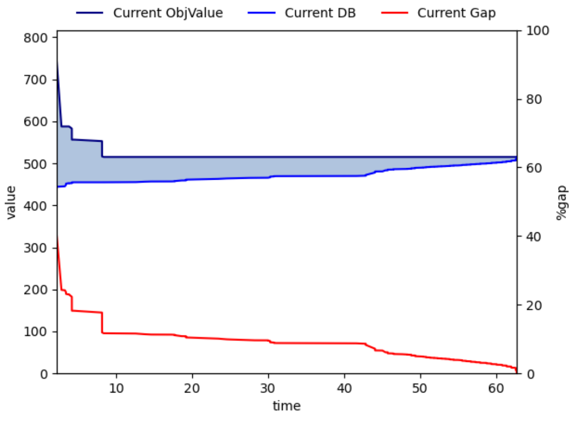

Figure 10: Gap and Time - Rais et al. (2014) without infeasible arcs - PDPT

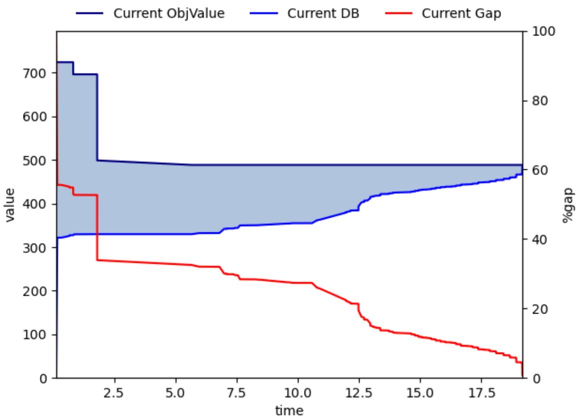

Figure 11: Gap and Time Lyu and Yu (2023) - PDPTWT

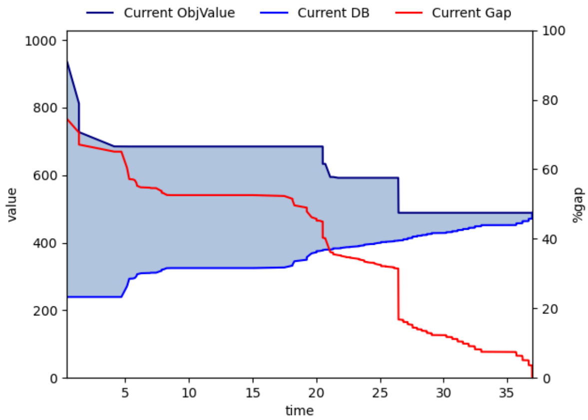

Figure 12: Gap and Time - Rais et al. (2014) without infeasible arcs - PDPTWT

5.3. Impacts of Symmetry Breaking Constraints

In this section, we examine the effectiveness of the symmetry breaking constraints outlined in Section 4.2 in improving the computing time of the model proposed by Cortés et al. (2010). Two versions of this model will be used to resolve instances: one with SBCs and the other without SBCs. Since the locations of the origin and destination depots of the vehicles have been changed relative to the original instances, there will be differences in objective values from the old results. These changes have been tested and found to not affect the feasibility of the problems.

Instances' Group	**Cortés et al., (2010) without SBCs**			**Cortés et al., (2010) with SBCs**		
	#opt.	**obj.**	**t(s)**	**#opt**	**obj.**	**t(s)**
PDPT-R5-K2-T1-Q100	10	390.757	2.319	10	390.757	2.074
PDPT-R5-K2-T2-Q100	10	386.833	7.122	10	386.833	7.099
PDPT-R5-K3-T3-Q100	10	361.104	19.521	10	361.104	19.994
PDPT-R7-K2-T1-Q100	10	464.497	65.094	10	464.497	65.482
PDPT-R7-K3-T3-Q100	7	498.154	1981.051	7	498.154	1979.862
Average		420.269	415.021		420.269	414.902
PDPTWT-3R-4K-4T-240L	3	436.664	21.188	3	436.664	20.865
PDPTWT-3R-4K-4T-240M	3	334.845	6.636	3	334.845	6.369
PDPTWT-3R-4K-4T-240S	3	301.027	3.176	3	301.027	3.353
PDPTWT-3R-4K-4T-300L	3	436.664	21.332	3	436.664	21.868
PDPTWT-3R-4K-4T-300M	3	334.845	6.444	3	334.845	6.656
PDPTWT-3R-4K-4T-300S	3	301.027	3.154	3	301.027	2.875
PDPTWT-3R-4K-5T-240L	3	436.664	392.187	3	436.664	392.326
PDPTWT-3R-4K-5T-240M	3	334.845	39.727	3	334.845	39.942
PDPTWT-3R-4K-5T-240S	3	301.027	6.894	3	301.027	6.517
PDPTWT-3R-4K-5T-300L	3	436.664	389.827	3	436.664	390.980
PDPTWT-3R-4K-5T-300M	3	334.845	39.990	3	334.845	40.094
PDPTWT-3R-4K-5T-300S	3	301.027	6.918	3	301.027	7.319
PDPTWT-4R-4K-4T-240L	1	582.834	3247.306	1	582.834	3246.374
PDPTWT-4R-4K-4T-240M	3	457.325	819.301	3	457.325	823.886
PDPTWT-4R-4K-4T-240S	3	402.896	25.488	3	402.896	24.573
PDPTWT-4R-4K-4T-300L	1	582.834	3246.037	1	582.834	3252.626

PDPTWT-4R-4K-4T-300M	3	457.325	821.544	3	457.325	827.881
PDPTWT-4R-4K-4T-300S	3	402.896	25.467	3	402.896	25.100
Average		398.681	506.812		398.681	507.756

Table 10: Results solved by models in Cortés et al. (2010) with and without Symmetries Breaking Constraints

All instances are solved with feasible solutions. The results show almost no difference between the average computing times of the two models (21,888 s vs. 20,865 s for PDPT-instances and 506,812 s vs 507.756 s for PDPTWT-instances).

6. Conclusion

In this thesis, we indicate the difference between PDP and PDPT and show the effect of the transfer operation. We analyze the core MILP models from Cortés et al. (2010) and Rais et al. (2014) for PDPTWT and show the major differences in terms of core configurations, decision variables, objective function, and setup of transfer locations in the two models, showing the potential effects of these differences.

Based on that, the differences and similarities between the constraints in the two models are outlined based on the function of each group of constraints. In addition, we point out some of the problems in these two models and offer solutions based on proposals from Lyu and Yu (2022) and our own knowledge.

Using the commercial solver Gurobi and the list of instances provided by Lyu and Yu (2022), we evaluate the performance of the two models and the effectiveness of techniques that can be used to improve their performance. The results show that the model of Cortés et al. (2010) is superior to that of Rais et al. (2014). However, it also demonstrates that the computation time of Rais et al. (2014) can be significantly improved through the addition of redundant constraints proposed by Lyu and Yu (2022) and the removal of infeasible arcs from the model's graph. This also indicates that the bounds of the number of decision variables and constraints are not decisive for the model's performance evaluation. In addition, the results do not demonstrate the possible improvement from symmetries breaking constraints proposed by Cortés et al. (2010). One possible reason for this result is that the solver has built-in strategies for dealing with symmetries in the problems.

7. References

Agrali, C. & Lee, S. (2023). The multi-depot pickup and delivery problem with capacitated electric vehicles, transfers, and time windows. *Computers & Industrial Engineering*, *179*, 109207. https://doi.org/10.1016/j.cie.2023.109207

Moeini, A., & Smith-Miles, K. (2019). Strong Valid Inequalities Identification for Mixed Integer Programming Problems. arXiv: Optimization and Control.

Avalos-Rosales and Cardona-Valdés. (2022). Binary variables relaxations in MIP formulations. https://doi.org/10.2139/ssrn.4160695

Cortés, C. E., Matamala, M. & Contardo, C. (2010). The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. *European Journal of Operational Research*, *200*(3), 711–724. https://doi.org/10.1016/j.ejor.2009.01.022

Drexl. Synchronization in Vehicle Routing— A Survey of VRPs with Multiple Synchronization Constraints.

Hiva Shiri, Morteza Rahmani and Morteza Khakzar Bafruei. (2019). Examining the impact of transfers in pickup and delivery systems. https://doi.org/10.5267/j.uscm.2019.7.003

Jacques Desrosiers, Yvan Dumas, Marius M. Solomon and François Soumis. (1995). Time Constrained Routing and Scheduling.

Morrison, D. R., Jacobson, S. H., Sauppe, J. J. & Sewell, E. C. (2016). Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization*, *19*, 79–102. https://doi.org/10.1016/j.disopt.2016.01.005

Rais, A., Alvelos, F. & Carvalho, M. S. (2014). New mixed integer-programming model for the pickup-and-delivery problem with transshipment. *European Journal of Operational Research*, *235*(3), 530–539. https://doi.org/10.1016/j.ejor.2013.10.038

Ropke, S., Cordeau, J.-F. & Laporte, G. (2007). Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *European Journal of Operational Research Networks*, *49*(4), 258–272. https://doi.org/10.1002/net.20177

S. Boyd and L. Vandenberghe. (2018). Localization and Cutting-Plane Methods.

Soares, R., Marques, A., Amorim, P. & Parragh, S. N. (2023). Synchronisation in vehicle routing: Classification schema, modelling framework and literature review. *European Journal of Operational Research*. https://doi.org/10.1016/j.ejor.2023.04.007

Wolfinger, D. & Salazar-González, J.-J. (2021). The Pickup and Delivery Problem with Split Loads and Transshipments: A Branch-and-Cut Solution Approach. *European Journal of Operational Research*, *289*(2), 470–484. https://doi.org/10.1016/j.ejor.2020.07.032

Zefeng Lyu, Andrew Junfang Yu. (2022). The pickup and delivery problem with transshipments: Critical review of two existing models and a new formulation. https://doi.org/10.1016/j.ejor.2022.05.053

Zhexi Fu, Joseph Y. J. Chow.. The pickup and delivery problem with synchronized en-route transfers for microtransit planning. https://doi.org/10.1016/j.tre.2021.102562

Liberti, L. (2012). Symmetry in Mathematical Programming. In J. Lee & S. Leyffer (Hrsg.), *Mixed Integer Nonlinear Programming* (The IMA Volumes in Mathematics and its Applications, Bd. 154, S. 263–283). New York, NY: Springer New York. https://doi.org/10.1007/978-1-4614-1927-3_9

Appendix

Detailed results of computational analysis will be given in this section. The names of the models are distinguished according to the list below:

Model 1. Cortés et al. (2010)
Model 2. Rais et al. (2014)
Model 3. Lyu and Yu (2022)
Model 4. Rais et al. (2014) without infeasible arcs
Model 5. Cortés et al. (2010) without SBCs
Model 6. Cortés et al. (2010) with SBCs

Instace name	Model 1		Model 2		Model 3		Model 4		Model 5		Model6	
	obj.	t(s)	obj.	t(s)	obj.	t(s)	obj.	t(s)	obj.	t(s)	obj.	t(s)
PDPT-R5-K2-T1-Q100-0	512.791	2.193	512.791	9.797	512.791	0.095	512.791	0.141	457.124	2.214	457.124	2.155
PDPT-R5-K2-T1-Q100-1	484.061	1.756	484.061	73.697	484.061	0.222	484.061	1.137	449.783	3.903	449.783	1.291
PDPT-R5-K2-T1-Q100-2	499.241	2.711	499.241	21.06	499.241	0.215	499.241	3.079	449.132	2.121	449.132	2.117
PDPT-R5-K2-T1-Q100-3	515.793	7.034	515.793	267.271	515.793	0.108	515.793	0.271	397.305	2.858	397.305	3.13
PDPT-R5-K2-T1-Q100-4	389.457	0.151	389.457	1.034	389.457	0.089	389.457	0.124	397.39	1.089	397.39	1.147
PDPT-R5-K2-T1-Q100-5	315.749	0.99	315.749	2.051	315.749	0.059	315.749	0.109	301.371	6.751	301.371	6.656
PDPT-R5-K2-T1-Q100-6	405.546	0.669	405.546	19.549	405.546	0.083	405.546	1.123	404.028	1.108	404.028	1.053
PDPT-R5-K2-T1-Q100-7	403.926	0.769	403.926	17.529	403.926	0.11	403.926	0.564	303.108	0.579	303.108	0.572
PDPT-R5-K2-T1-Q100-8	341.607	0.235	341.607	10.725	341.607	0.078	341.607	0.095	300.477	0.702	300.477	0.73
PDPT-R5-K2-T1-Q100-9	556.215	2.55	556.215	8.358	556.215	0.216	556.215	0.371	447.85	1.863	447.85	1.89
PDPT-R5-K2-T2-Q100-0	428.414	2.776	428.414	47.179	428.414	0.183	428.414	0.732	369.26	2.352	369.26	2.453
PDPT-R5-K2-T2-Q100-1	376.575	2.296	376.575	75.035	376.575	0.624	376.575	2.2	357.111	43.197	357.111	42.594
PDPT-R5-K2-T2-Q100-2	346.885	1.017	346.885	10.802	346.885	0.089	346.885	0.762	350.818	3.258	350.818	3.229
PDPT-R5-K2-T2-Q100-3	535.781	10.515	535.781	25.909	535.781	0.12	535.781	1.324	471.833	4.693	471.833	5.076
PDPT-R5-K2-T2-Q100-4	331.558	1.546	331.558	1.464	331.558	0.079	331.558	0.285	296.969	1.751	296.969	1.793
PDPT-R5-K2-T2-Q100-5	556.571	7.399	556.571	277.085	556.571	0.076	556.571	2.03	534.131	4.689	534.131	4.393
PDPT-R5-K2-T2-Q100-6	334.881	0.687	334.881	11.725	334.881	0.324	334.881	0.827	301.189	1.687	301.189	1.707
PDPT-R5-K2-T2-Q100-7	365.109	0.521	365.109	2.274	365.109	0.108	365.109	0.352	333.705	1.428	333.705	1.473
PDPT-R5-K2-T2-Q100-8	479.829	1.714	479.829	18.645	479.829	0.087	479.829	0.782	516.53	6.383	516.53	6.329
PDPT-R5-K2-T2-Q100-9	399.796	3.073	399.796	2.485	399.796	0.081	399.796	0.475	336.779	1.779	336.779	1.941
PDPT-R5-K3-T3-Q100-0	352.842	0.4	352.842	10.374	352.842	0.315	352.842	0.593	275.32	4.675	275.32	5.144
PDPT-R5-K3-T3-Q100-1	430.463	6.275	430.463	1.332	430.463	0.26	430.463	0.217	393.31	16.351	393.31	17.448
PDPT-R5-K3-T3-Q100-2	505.852	10.43	505.852	797.401	505.852	0.158	505.852	3.98	429.639	4.68	429.639	4.674
PDPT-R5-K3-T3-Q100-3	396.785	1.605	396.785	46.673	396.785	0.201	396.785	0.542	329.048	4.863	329.048	4.961
PDPT-R5-K3-T3-Q100-4	370.568	3.017	370.568	0.891	370.568	0.196	370.568	0.194	353.332	21.207	353.332	21.739
PDPT-R5-K3-T3-Q100-5	415.321	2.74	415.321	185.663	415.321	0.169	415.321	1.915	408.859	18.635	408.859	20.497
PDPT-R5-K3-T3-Q100-6	397.504	7.39	397.504	58.194	397.504	0.297	397.504	4.212	307.662	19.858	307.662	20.019
PDPT-R5-K3-T3-Q100-7	436.69	7.221	436.69	96.324	436.69	0.189	436.69	2.604	367.144	15.654	367.144	15.91
PDPT-R5-K3-T3-Q100-8	505.978	19.983	505.978	860.693	505.978	2.367	505.978	10.207	374.809	2.946	374.809	2.954
PDPT-R5-K3-T3-Q100-9	402.109	3.321	402.109	63.849	402.109	0.221	402.109	6.927	371.914	86.34	371.914	86.595
PDPT-R7-K2-T1-Q100-0	403.047	10.362	403.047	59.888	403.047	0.309	403.047	21.027	364.816	9.964	364.816	10.398
PDPT-R7-K2-T1-Q100-1	629.956	23.985	629.956	153.524	629.956	0.869	629.956	12.769	559.461	35.752	559.461	36.326
PDPT-R7-K2-T1-Q100-2	579.831	49.452	579.831	679.551	579.831	0.182	579.831	10.019	540.672	35.372	540.672	35.606
PDPT-R7-K2-T1-Q100-3	483.452	45.529	491.081	3600.058	483.452	7.538	483.452	147.664	425.097	55.449	425.097	61.052
PDPT-R7-K2-T1-Q100-4	451.723	10.254	451.723	111.342	451.723	0.331	451.723	4.76	415.542	95.225	415.542	95.55
PDPT-R7-K2-T1-Q100-5	458.503	21.983	458.503	3600.051	458.503	0.302	458.503	27.781	372.333	103.018	372.333	103.925
PDPT-R7-K2-T1-Q100-6	536.817	15.52	536.817	40.909	536.817	0.251	536.817	5.067	488.692	267.325	488.692	264.409
PDPT-R7-K2-T1-Q100-7	596.933	44.007	596.933	3600.051	596.933	0.471	596.933	11.194	515.306	23.705	515.306	22.855
PDPT-R7-K2-T1-Q100-8	546.647	8.2	546.647	84.399	546.647	0.194	546.647	5.066	522.159	10.07	522.159	10.136
PDPT-R7-K2-T1-Q100-9	563.203	14.247	573.913	3600.032	563.203	0.884	563.203	81.979	440.895	15.063	440.895	14.561
PDPT-R7-K3-T3-Q100-0	555.03	955.192	560.102	3600.049	555.03	1.915	555.03	521.226	484.298	3598.582	484.298	3593.136
PDPT-R7-K3-T3-Q100-1	580.063	261.767	580.063	137.539	580.063	0.55	580.063	27.96	548.379	532.439	548.379	531.982
PDPT-R7-K3-T3-Q100-2	556.786	3600.086	593.368	3600.094	556.786	0.959	556.786	303.135	447.266	3600.167	447.266	3600.186
PDPT-R7-K3-T3-Q100-3	535.187	553.101	535.187	3600.048	535.187	4.135	535.187	145.319	476.633	3600.064	476.633	3600.054
PDPT-R7-K3-T3-Q100-4	586.176	151.668	586.176	2449.792	586.176	0.655	586.176	157.091	545.185	804.597	545.185	806.199
PDPT-R7-K3-T3-Q100-5	508.842	3366.259	508.842	875.369	508.842	4.352	508.842	126.574	445.761	2025.328	445.761	2021.809
PDPT-R7-K3-T3-Q100-6	515.202	796.824	515.202	1715.573	515.202	1.151	515.202	63.575	499.106	3600.041	499.106	3600.08
PDPT-R7-K3-T3-Q100-7	713.191	85.431	713.191	3600.103	713.191	0.526	713.191	473.134	608.748	303.232	608.748	296.708
PDPT-R7-K3-T3-Q100-8	543.132	427.547	576.736	3600.685	543.132	0.722	543.132	317.729	442.002	1364.52	442.002	1365.975
PDPT-R7-K3-T3-Q100-9	510.668	89.541	510.668	238.417	510.668	0.592	510.668	16.126	484.158	381.543	484.158	382.494
PDPTWT-3R-4K-4T-240L-0	365.75	1.923	365.75	164.93	365.75	1.727	365.75	3.981	431.394	9.034	431.394	10.07
PDPTWT-3R-4K-4T-240L-1	306.647	1.44	306.647	70.525	306.647	1.05	306.647	1.304	369	4.987	369	4.199
PDPTWT-3R-4K-4T-240L-2	427.987	14.004	430.194	512.158	430.194	5.471	427.987	7.319	509.598	49.542	509.598	48.325
PDPTWT-3R-4K-4T-240M-0	180.12	0.726	180.12	10.233	180.12	0.214	180.12	0.939	261.578	11.553	261.578	11.282

PDPTWT-3R-4K-4T-240M-1	309.829	3.786	309.829	299.902	309.829	2.735	309.829	4.257	328.622	1.254	328.622	1.239
PDPTWT-3R-4K-4T-240M-2	279.111	0.596	279.111	16.02	279.111	0.502	279.111	0.535	414.335	7.101	414.335	6.585
PDPTWT-3R-4K-4T-240S-0	257.099	0.23	257.099	9.635	257.099	0.61	257.099	0.345	344.663	1.844	344.663	1.907
PDPTWT-3R-4K-4T-240S-1	222.201	0.51	222.201	19.49	222.201	0.349	222.201	0.302	358.401	7.075	358.401	7.462
PDPTWT-3R-4K-4T-240S-2	209.427	0.864	209.427	13.072	209.427	0.292	209.427	2.427	200.017	0.609	200.017	0.689
PDPTWT-3R-4K-4T-300L-0	365.75	1.725	365.75	226.942	365.75	2.541	365.75	2.432	431.394	10.196	431.394	9.661
PDPTWT-3R-4K-4T-300L-1	306.647	1.414	306.647	104.72	306.647	1.233	306.647	1.138	369	4.621	369	4.754
PDPTWT-3R-4K-4T-300L-2	427.987	14.343	427.987	359.06	427.987	5.312	427.987	6.72	509.598	49.18	509.598	51.19
PDPTWT-3R-4K-4T-300M-0	180.12	0.79	180.12	4.121	180.12	1.042	180.12	0.636	261.578	11.357	261.578	10.761
PDPTWT-3R-4K-4T-300M-1	309.829	3.686	309.829	240.848	309.829	2.134	309.829	3.186	328.622	1.345	328.622	1.303
PDPTWT-3R-4K-4T-300M-2	279.111	0.684	279.111	19.884	279.111	0.256	279.111	0.597	414.335	6.63	414.335	7.903
PDPTWT-3R-4K-4T-300S-0	257.099	0.243	257.099	1.751	257.099	0.597	257.099	0.323	344.663	1.734	344.663	1.734
PDPTWT-3R-4K-4T-300S-1	222.201	0.463	222.201	3.712	222.201	0.201	222.201	0.284	358.401	7.128	358.401	6.262
PDPTWT-3R-4K-4T-300S-2	209.427	0.979	209.427	17.731	209.427	0.408	209.427	1.093	200.017	0.6	200.017	0.63
PDPTWT-3R-4K-5T-240L-0	365.75	6.183	365.75	287.649	365.75	3.741	365.75	2.366	431.394	33.53	431.394	34.342
PDPTWT-3R-4K-5T-240L-1	306.647	2.618	306.647	125.224	306.647	2.002	306.647	1.854	369	34.959	369	35.336
PDPTWT-3R-4K-5T-240L-2	402.931	9.199	404.542	1202.158	404.542	8.965	402.931	11.583	509.598	1108.071	509.598	1107.299
PDPTWT-3R-4K-5T-240M-0	180.12	3.969	180.12	18.227	180.12	0.686	180.12	0.927	261.578	82.207	261.578	82.201
PDPTWT-3R-4K-5T-240M-1	309.829	2.397	309.829	341.461	309.829	5.341	309.829	6.85	328.622	11.594	328.622	11.71
PDPTWT-3R-4K-5T-240M-2	279.111	0.729	279.111	14.664	279.111	0.282	279.111	0.664	414.335	25.38	414.335	25.916
PDPTWT-3R-4K-5T-240S-0	257.099	0.35	257.099	11.6	257.099	0.506	257.099	0.268	344.663	7.47	344.663	7.247
PDPTWT-3R-4K-5T-240S-1	222.201	0.578	222.201	9.781	222.201	0.244	222.201	0.584	358.401	12.238	358.401	11.333
PDPTWT-3R-4K-5T-240S-2	209.427	2.694	209.427	23.317	209.427	0.516	209.427	1.787	200.017	0.974	200.017	0.971
PDPTWT-3R-4K-5T-300L-0	365.75	6.218	365.75	174.477	365.75	2.587	365.75	4.251	431.394	31.904	431.394	33.556
PDPTWT-3R-4K-5T-300L-1	306.647	2.45	306.647	111.191	306.647	1.534	306.647	2.28	369	35.363	369	35.691
PDPTWT-3R-4K-5T-300L-2	402.931	10.224	402.931	366.09	402.931	5.329	402.931	8.176	509.598	1102.213	509.598	1103.694
PDPTWT-3R-4K-5T-300M-0	180.12	3.657	180.12	57.738	180.12	0.27	180.12	1.001	261.578	83.578	261.578	83.378
PDPTWT-3R-4K-5T-300M-1	309.829	2.248	309.829	148.399	309.829	3.112	309.829	4.537	328.622	11.308	328.622	11.752
PDPTWT-3R-4K-5T-300M-2	279.111	0.723	279.111	22.787	279.111	0.232	279.111	0.763	414.335	25.083	414.335	25.153
PDPTWT-3R-4K-5T-300S-0	257.099	0.346	257.099	9.13	257.099	0.727	257.099	0.313	344.663	7.142	344.663	8.232
PDPTWT-3R-4K-5T-300S-1	222.201	0.538	222.201	3.444	222.201	0.486	222.201	0.318	358.401	12.568	358.401	12.644
PDPTWT-3R-4K-5T-300S-2	209.427	2.46	209.427	9.527	209.427	0.341	209.427	2.198	200.017	1.045	200.017	1.081
PDPTWT-4R-4K-4T-240L-0	405.405	11.411	405.405	901.388	405.405	4.764	405.405	9.331	579.413	2541.814	579.413	2538.912
PDPTWT-4R-4K-4T-240L-1	356.039	16.289	356.039	1097.909	356.039	15.201	356.039	20.351	480.106	3600.071	480.106	3600.183
PDPTWT-4R-4K-4T-240L-2	488.656	54.498	488.656	3600.07	488.656	14.742	488.656	56.056	688.984	3600.033	688.984	3600.026
PDPTWT-4R-4K-4T-240M-0	310.245	3.175	310.245	66.357	310.245	5.086	310.245	1.751	435.859	479.794	435.859	481.384
PDPTWT-4R-4K-4T-240M-1	420.402	42.556	420.402	3600.079	420.402	113.754	420.402	57.302	418.993	60.687	418.993	59.796
PDPTWT-4R-4K-4T-240M-2	297.343	1.703	297.343	128.455	297.343	5.775	297.343	1.888	517.124	1917.421	517.124	1930.478
PDPTWT-4R-4K-4T-240S-0	377.713	11.008	377.713	295.714	377.713	7.888	377.713	12.483	428.415	14.166	428.415	13.788
PDPTWT-4R-4K-4T-240S-1	338.304	4.368	338.304	55.975	338.304	2.472	338.304	2.186	479.359	25.031	479.359	23.974
PDPTWT-4R-4K-4T-240S-2	282.121	33.386	282.121	2796.75	282.121	19.044	282.121	31.222	300.914	37.267	300.914	35.958
PDPTWT-4R-4K-4T-300L-0	405.405	10.932	405.405	691.97	405.405	5.828	405.405	9.667	579.413	2537.974	579.413	2557.678
PDPTWT-4R-4K-4T-300L-1	356.039	16.488	356.039	627.918	356.039	9.626	356.039	16.086	480.106	3600.097	480.106	3600.167
PDPTWT-4R-4K-4T-300L-2	488.656	52.879	488.656	3600.072	488.656	20.189	488.656	35.942	688.984	3600.04	688.984	3600.032
PDPTWT-4R-4K-4T-300M-0	310.245	3.002	310.245	77.811	310.245	2.306	310.245	2.619	435.859	480.66	435.859	486.193
PDPTWT-4R-4K-4T-300M-1	420.402	43.662	420.402	3600.087	420.402	74.181	420.402	71.114	418.993	59.03	418.993	59.742
PDPTWT-4R-4K-4T-300M-2	297.343	1.716	297.343	94.295	297.343	0.695	297.343	2.228	517.124	1924.943	517.124	1937.709
PDPTWT-4R-4K-4T-300S-0	377.713	12.088	377.713	329.976	377.713	9.615	377.713	10.435	428.415	13.617	428.415	13.406
PDPTWT-4R-4K-4T-300S-1	338.304	4.302	338.304	59.728	338.304	2.049	338.304	2.076	479.359	24.397	479.359	24.713
PDPTWT-4R-4K-4T-300S-2	282.121	33.37	282.121	677.587	282.121	20.078	282.121	48.641	300.914	38.387	300.914	37.18