# A Branch-Cut-And-Price approach for the Two-Echelon Vehicle Routing Problem with Drones

Sylvain Lichau, Ruslan Sadykov, Julien François, Rémy Dupas

▶ **To cite this version:**

Sylvain Lichau, Ruslan Sadykov, Julien François, Rémy Dupas. A Branch-Cut-And-Price approach for the Two-Echelon Vehicle Routing Problem with Drones. 2024. hal-04584429

**HAL Id: hal-04584429**
**https://hal.science/hal-04584429**

Preprint submitted on 23 May 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - ShareAlike 4.0 International License

# A Branch-Cut-And-Price approach for the Two-Echelon Vehicle Routing Problem with Drones

**Sylvain Lichau**
Univ. Bordeaux, CNRS, Bordeaux INP, IMS, UMR 5218, F-33400 Talence, France,
Email: sylvain.lichau@u-bordeaux.fr
**Ruslan Sadykov**
Univ. Bordeaux, CNRS, INRIA, Bordeaux INP, IMB, UMR 5251, F-33400 Talence, France
**Julien François**
Univ. Bordeaux, CNRS, Bordeaux INP, IMS, UMR 5218, F-33400 Talence, France,
**Rémy Dupas**
Univ. Bordeaux, CNRS, Bordeaux INP, IMS, UMR 5218, F-33400 Talence, France

June 2023

**Abstract**

In this paper, we propose a new set-partitioning model for the two-echelon vehicle routing problem with drones (2E-VRP-D), where partial routes corresponding to drone movements are enumerated using an efficient dynamic program. To solve the model we use an exact branch-cut-and-price algorithm, and a labelling algorithm for the pricing problem both based on state-of-the art literature. We also propose an adaptation of the well-known rounded capacity cuts for this problem, as well as pre-processing methods to reduce the size of the problem. In addition, this paper presents an effective heuristic branch-cut-and-price, based on the exact branch-cut-and-price algorithm. Extensive computational experiments are presented, showing that the exact algorithm can solve all the instances from the literature on the 2E-VRP-D, and almost multiply by four the size of the clustered instances solved optimally. Sensitivity analysis is also conducted for the proposed improvements.

## 1 Introduction

In recent years, the explosion in e-commerce and the ever-growing concentration of populations in cities have led to a drastic increase of demand for home deliveries, overloading last mile logistics. To overcome this challenge, incorporating drones in deliveries is a promising solution. In addition to paralleling deliveries, enabling customers to receive their packages faster, using drones would also decrease costs for the provider (Wang, 2015). It would also decrease the pollution emitted by truck routes, drones being less polluting than trucks. By reducing

the number of customers delivered by trucks in cities, and in particular on small roads, the traffic congestion would decrease significantly.

Numerous drone delivery pilot projects have been successfully completed and many delivery companies are trying to integrate drones to deliveries. One of the first successful implementation of a couple drone/truck delivery, to our knowledge, has been established in France in 2016, to a difficult-to-reach area (DPDgroup, 2016). In urban area, in 2017 a pilot project named Vans & drones was conducted in Zürich (Roca-Riu and Menendez, 2019), which combined drones and vans to test on-demand delivery. In 10 days, 50 packages were successfully delivered, and the delay was always lower than two hours. Swiss Post has permanently established the use of drones to transfer blood samples between hospitals in Lugano, Zürich and Bern (Post, 2023). More recently, Amazon is claiming that they will start using drone deliveries in Italy, the UK and the US by the end of 2024 (Amazon, 2023).

Despite the positive impact of drone-assisted delivery and the existence of some real-word applications, certain difficulties remain in terms of implementation as it might bring this new technology very close to the public. Drone use raises security issues as the drone must be completely foolproof and ready to face any problem, such as extreme weather, technical default, breakdown, signal loss or even attacks. Drones are also subject to prohibited air zone and other legal restrictions (Lin et al., 2018). Finally, public risk beliefs, such as privacy invasions or injuring people due to malfunction, are also an important obstacle to the implementation of drone deliveries on large scales, in particular in urban areas (Zhu, 2019). Garg et al. (2023) provides a literature review over the potential benefits, challenges, and limitations of the usage of drones in last mile delivery.

Due to the potential evolution of public opinion, regulation and technology, it is of first importance to understand how deliveries with drones will impact cities and how to design them in the best possible way. In line with these future developments, the field of operations research is tackling several variants of drone delivery problems, as reviewed in (Macrina et al., 2020) and (Chung et al., 2020). Researchers are considering different truck-drone cooperation. In the classic vehicle routing problem with drones (VRP-D), each truck is paired with a single drone that delivers a single customer leaving and merging only when the truck is parked at a customer. Many different cooperation systems exist, leading to a large variety of problems.

In this paper we study the Two-Echelon Vehicle Routing Problem with Drones (2E-VRP-D). In this problem a set of vehicles and a set of drones have to serve a set of customers from a depot. The aim is to minimize the total duration of the routes. We have chosen to focus on this problem because it is sufficiently generic to correspond to several real-world situations, whereas encompassing the following set of requirements:

- Each truck contains an adjustable number of drones. The useful capacity of the trucks depends of the number of carried drones, each drone and its necessary equipment having a weight. Carrying several drones on a truck rather than just one can be easily implemented, while having an important impact on the delivery.

- Full synchronization exists between the truck and the drone. Indeed, a model where drones can change truck en route will be hard to implement on a real-world application, for security and responsibility reasons. A drone can fly only when its assigned truck is stationed at a customer, so as to achieve a compromise between simplicity and efficiency in the proposed model. Furthermore, the drones can operate multiple round trips from the same customer. Choosing the number of drones carried for each vehicle enables us to balance the total number of drones, the weight taken by the drones in a truck and the efficiency of the route.

- Each drone has a limited capacity and can only carry one package per flight. One package per drone is realistic as it is more secure than allowing a drone to carry several packages at the same time. It also reduces the payload, enabling longer flight. With removable batteries, drones can be instantaneously recharged when reaching the truck. Carrying a sufficient number of batteries for deliveries does not entail any particular transport constraints. Static truck when drones are flying allows close range monitoring in case of problems. A battery limits the maximal flight distance, dependent of the payload. It is assumed that drones are fully recharged upon reaching the truck.

- Each customer is characterized by a demand, a deadline, its availability for drone delivery and a service time dependent on the means of transport. The service at a customer delivered by a vehicle happens in parallel with the deliveries made by drones from this customer.

This is not the usual cooperation system between trucks and drones, but it can be argued that it enables an operator to remain near the drones to monitor them, or in case of an accident. For these reasons we think that the 2E-VRP-D is a highly topical and realistic problem as it is easy to implement in the delivery process. It is related to the two-echelon vehicle routing problem (2E-VRP), see (Sluijk et al., 2023), as we can consider the deliveries of the trucks as a first echelon, and the deliveries of the drones as a second echelon. In this context, Zhou et al. (2023) recently proposed a branch-and-price approach that optimally solves instances of up to 30 customers in less than 3 hours for the 2E-VRP-D. Given that real-world last mile delivery problems commonly contain a higher number of customers to serve it is necessary to be able to solve larger instances in a reasonable time. In this paper we propose a branch-cut-and-price algorithm for the 2E-VRP-D using recent advances from the well-known capacitated vehicle routing problem (CVRP). The main contributions of this paper are as follows:

- A new branch-cut-and-price approach based on the enumeration of the partial routes corresponding to drone movements in the subproblems and modelled as scheduling problems, and solved using an efficient dynamic program and state-of-the-art components. The algorithm is strengthened by an adaptation of the rounded capacity cuts to better suit this problem, by taking into account the drones weight in the truck. We also propose preprocessing methods to eliminate dominated and infeasible the enumerated partial routes to efficiently reduce the subproblems size without excluding optimal solutions.
- A heuristic branch-cut-and-price based on the restriction of the subproblem solution. This heuristic is particularly efficient for instances where customers are close to each other. In this setting, drones flying from a parked customer $c$ can reach many customers. We say $c$ has a large drone-neighborhood. Moreover, in this setting, the drone-neighborhoods are highly interconnected, meaning that a customer can be reached by drone by many different customers.
- Extensive experiments are conducted on the instances and show that our methods can solve clustered instances with up to 120 customers, increasing greatly the size of the solved instances from the literature. We also show the benefit of the improvements proposed. Six new sets of medium-size instances with 40, 100, 120, 130, 140 and 150 customers extending the dataset of small-size instances proposed in (Zhou et al., 2023) and based on (Chen et al., 2021), (Solomon, 1987) and (Gehring and Homberger, 1999).

In section 2, we present a literature review mainly focusing on exact methods to solve vehicle routing problems with drones. In section 3, we define the problem we are addressing. In section 4, we present the pricing problem and how we solve it. In section 5, we present our approach, providing details on the branch-and-cut algorithm and on improvements we use. In section 6, we present the heuristic branch-cut-and-price. Finally, in section 7, we provide computational experiments highlighting the strength of our method.

# 2 Litterature review

The vehicle routing problem with drones (VRPD) class of problem is a variant of the classic vehicle routing problem (VRP) that integrates drones in the deliveries. This section is separated in two parts, the first one will cover the variants of the travelling salesman problem (TSP) (i.e. single tour), and the second the variants of the VRP (i.e. multiple tours).

## 2.1 Drones integrated in TSP

The interest to incorporate drones into routing problem emerged with the work of Murray and Chu (2015), in which a drone and a truck cooperate to solve a

TSP-like problem called the flying sidekick travelling salesman problem (FSTSP). They presented a MIP model and two heuristics to solve the problem with up to 10 customers. Various extensions were proposed, as in (Agatz et al., 2018) that allowed trucks to wait for the drones at the nodes and proposed a MILP and two heuristics based on the route-first, cluster-second method to solve the problem. The authors also analyse experimentally the time saving achieved by the use of a truck paired with a drone versus a truck alone. Bouman et al. (2018) used a dynamic program to solve the TSP-D up to 20 customers. They also showed that decreasing the number of customers visited by a truck while the drone was flying significantly decreased the computation time, without decreasing the quality of the solution significantly. Yang et al. (2023) consider the robust drone-truck delivery problem, where a truck and a drone have to deliver customers to maximize a profit function under truck routes uncertainty. The uncertainty is considered on the duration of the truck arcs, due to network congestion. They propose a branch and price algorithm to solve instances with up to 40 customers.
To improve the efficiency of the cooperation even further, researchers have studied the use of several drones on the truck. In (Karak and Abdelghany, 2019) the authors propose the "Hybrid Vehicle-Drone Routing Problem" (HVDRP), where a single vehicle carries a fleet of drones performing pickup and delivery operations. They propose a MIP formulation along with 3 heuristics to solve the problem. Murray and Raj (2020) introduced the "Multiple Flying Sidekicks travelling Salesman Problem" (mFSTSP), which is an extension with several vehicles of the FSTSP. It considers an arbitrary number of heterogeneous drones that may be deployed from the depot or from the delivery truck. They provide a MIP formulation solving instances with at most 8 customers, as well as a heuristic approach. Kang and Lee (2021) introduce the Heterogeneous Drone-Truck Routing Problem (HDTRP), where drones with different speeds and batteries are setup on a truck, and can fly only when the truck is parked at a customer. They propose a MILP formulation as well as a branch-and-cut algorithm to solve up to 50 customers instances optimally.

## 2.2 Drones integrated in VRP

The use of drones with a set of trucks was first introduced in (Wang et al., 2017), which presents the Vehicle Routing Problem with Drones (VRPD), where multiple trucks are equipped with drones to serve a set of customers. In this problem, the drones carried by a truck can deliver one parcel per flight, and the truck can move between the launch and the retrieval of its drones. The authors analyse different worst-case scenarios and provide upper bounds on the savings induced by the use of drone. The work is extended in (Wang and Sheu, 2019), but the drones can operate multiple deliveries per trip and the drone-truck assignment is open. The authors propose a branch-and-price algorithm to solve up to 13 customers. They also provide a sensitivity analysis over the maximal authorized flying duration of the drones. Poikonen et al. (2017) consider the battery of the drones, a different metric for the drones and the trucks,

as well as a cost objective function. They provide theoretical results regarding the VRPD. Di Puglia Pugliese and Guerriero (2017) present the first MIP model for an extension of the VRP-D called the vehicle-drone routing problem with time windows (VDRPTW), which also considers multiple drones per truck, time windows and truck capacity. They also provide a numerical study comparing VRP to VRP-D. They analyse that drone is only useful when the drone's transportation cost is lower than the truck's transportation cost. This work is extended in (Di Puglia Pugliese et al., 2020) where the authors present a MIP model solving up to 15 customers instances. They also compare VRP, VRP-D and routing problem with drones (RPD), where the drones deliver customers directly from the depot, on their $CO_2$ emission and traffic congestion. Sacramento et al. (2019) address an extension of the VRP-D where the drones can return to the depot directly. They present an adaptative large neighborhood search and with a MIP model. In (Schermer et al., 2019), the authors consider the Vehicle Routing Problem and En Route Operation (VRPDERO), where a drone can be launched and retrieved on a moving truck. They propose a MIP model and a heuristic. In (Kitjacharoenchai et al., 2020) the objective function is to minimize the total completion time. A truck can only launch or retrieve a single drone upon reaching a node. A drone can deliver multiple customers before returning to its assigned truck. The authors propose a mixed integer non-linear program (MINLP) and solve instances up to 9 customers optimally. Tamke and Buscher (2021) proposed a branch-and-cut algorithm for the VRP-D without truck capacity solving up to 30 customers instances. The work is extended in (Tamke and Buscher, 2023), where the speed at which a drone performs a flight must be selected from a discrete set, and influence the energy consumption. The authors propose a MIP model along with valid inequalities and preprocessing methods to accelerate the solving. They solve instances with 20 customers optimally, and find feasible solutions on 50 customers instances with a heuristic version of their MIP.

Although the VRPD and its extensions are the most studied, other works related to the integration of drones in vehicle routing problems are also considered. Ulmer and Thomas (2018) consider a dynamic variant, the Same-Day Delivery Routing Problem with Heterogeneous Fleets of drones and vehicles (SDDPHF), where customers requests arrive while the vehicles are already on the road. Drones make deliveries from the depot, serving a single customer before coming back to the depot. The authors present a policy function approximation, to decide whether a request is fulfilled by a drone or a vehicle. Kitjacharoenchai et al. (2019) introduce the multiple travelling salesman problem with drones (mTSPD), where multiple vehicles and drones deliver a set of customers from a depot. They do not limit the number of drones on a vehicle and the truck-drone assignment is open ( i.e. the launch and retrieve operations can be operated by different trucks). The authors propose a MIP model solving instances with up to 9 customers instances and a heuristic. Dukkanci et al. (2021) proposes the Energy Minimizing and Range Constrained Drone Delivery Problem (ERDDP) where trucks move from the depot to parking spots. Drones can fly from a parking spot or from the depot to deliver a single customer. The objective function

minimizes a cost function over the energy spent, and the operational costs. The drone speed directly impacts its energy consumption, and is considered a decision variable. The authors conduct a parametric analysis over the number of vehicles used. Meng et al. (2023) introduce the multi-visit drone routing problem for pickup and delivery services (MDRP-PD), propose a MIP and a two-stage heuristic based on simulated annealing. Zhou et al. (2023) presented the two-echelon vehicle routing problem with drones (2E-VRP-D). We recall that in this problem, the number of drones on a truck is a decision variable, which is allowing for a better use of the drone fleet. The drones can only fly when their assigned truck is parked at a nearby client. To solve the problem, they presented a MIP, a branch and price and a heuristic. They managed to solve up to 30 clients to optimality. A very similar problem is studied in (Yin et al., 2024), which consider the routing of vehicle-and-drone cooperative to deliver blood products to hospitals. The main difference between the two problems is the objective function, as Yin et al. (2024) penalise the squared weighted arrival time at each hospital. They use a Benders-decomposition column-generation algorithm to solve instances up to 45 customers.

To give an overview of the literature regarding exact approaches for drones integrated in VRP, we present a summary table. Let us first recall notation for the vehicle routing problems with drones introduced in (Tamke and Buscher, 2021). The criteria are the following: number of trucks (#t), number of drones per truck (#d), drone-truck assignment (dta) i.e. whether the drone has an assigned truck or not, launch and retrieve operations (lro), drone range (ran), drone capacity (cap), and objective function (obj). Most criteria are self-explanatory. In the table 1, the keyword 'return' refers to the ability of a truck to return to an already visited customer. The keyword 'loops' refers to the ability of a drone to operate loop operation, in which the start and end location of the flight is the same.

| Criterion | Value | Description | Criterion | Value | Description |
|---|---|---|---|---|---|
| #t | 1 | single truck | ran | t | time |
| | m | multiple trucks | | d | distance |
| #d | 1 | single drone | | n | number of nodes |
| | m | multiple drones | | e | energy consumption |
| | u | unlimited number | | u | unrestricted |
| dta | f | fixed | | m | multiple modes |
| | o | open | cap | 1 | single customer |
| lro | a | return and loops allowed | | m | multiple customers |
| | p | return and loops prohibited | | | |
| | l | only loops allowed | | | |
| obj | min ct | minimize completion time | | | |
| | min tc | minimize total costs | | | |
| | min wt | minimize customer waiting times | | | |
| | bi-obj | minimize total costs and completion time | | | |
| | min max ct | minimize maximum completion time | | | |
| | min total ct | minimize total completion time | | | |

Table 1: Classification criteria for delivery systems with truck-drone cooperation. (Tamke and Buscher, 2021)

Using the notation defined in Table 1, we present Table 2, providing information about the problem studied, the exact resolutions method and the results of different paper considering the integration of drones to vehicle routing problems. The Approach are abbrebivated as follows: BP stands for branch-and-price, BC for branch-and-cut and BCP for branch-and-cut-and-price.

| Reference | Problem | ⟨#t \| #d \| dta \| lro \| ran \| cap \| obj⟩ | Approach | Size | Note |
|---|---|---|---|---|---|
| Sacramento et al. (2019) | VRP-D | ⟨ m \| 1 \| f \| p \| t \| 1 \| min tc ⟩ | MIP | 12 | |
| Sacramento et al. (2019) | VRP-D | ⟨ m \| 1 \| f \| p \| t \| 1 \| min tc ⟩ | MIP | 12 | |
| Di Puglia Pugliese and Guerriero (2017) | VDRPTW | ⟨ m \| m \| f \| p \| d \| 1 \| min tc ⟩ | MIP | 10 | Time Window |
| Meng et al. (2023) | MDRP-PD | ⟨ m \| 1 \| f \| p \| e \| m \| min tc ⟩ | MIP | 11 | Pickup and Delivery |
| Wang and Sheu (2019) | VRPD | ⟨ m \| m \| o \| p \| t \| m \| min tc ⟩ | BP | 13 | Docking Hubs |
| Schermer et al. (2019) | VRPDERO | ⟨ m \| m \| f \| p \| d \| 1 \| min max ct ⟩ | MIP | 10 | En-route Operation |
| Kitjacharoenchai et al. (2019) | mTSPD | ⟨ m \| u \| o \| p \| u \| 1 \| min max ct ⟩ | MIP | 9 | |
| Kitjacharoenchai et al. (2020) | VRPD | ⟨ m \| m \| f \| p \| d \| m \| min total ct ⟩ | MINLP | 9 | |
| Di Puglia Pugliese et al. (2020) | VRP-D | ⟨ m \| m \| f \| p \| d \| 1 \| min tc ⟩ | MIP | 15 | Time Window |
| Dukkanci et al. (2021) | ERDDP | ⟨ m \| m \| f \| l \| e \| 1 \| min tc ⟩ | SOCP | 37 | Drone Speed Variable |
| Tamke and Buscher (2021) | VRPD | ⟨ m \| m \| f \| p \| d \| 1 \| min max ct ⟩ | BC | 20 | |
| Zhou et al. (2023) | 2E-VRP-D | ⟨ m \| m \| f \| l \| e \| 1 \| min total ct ⟩ | BP | 30 | Variable nb Drone per Truck |
| Tamke and Buscher (2023) | VRPD-DSS | ⟨ m \| m \| f \| p \| e \| 1 \| min tc ⟩ | MIP | 20 | Drone Speed Variable |
| Yin et al. (2024) | BPDRP | ⟨ m \| m \| f \| l \| t \| 1 \| min tc ⟩ | BP | 45 | Quadratic objective |
| Our | 2E-VRP-D | ⟨ m \| m \| f \| l \| e \| 1 \| min total ct ⟩ | BCP | 120 | Variable Drone per Truck |

Table 2: Overview of exact approaches for drones integrated in VRP

# 3 Formulation

We now introduce the addressed problem. The 2E-VRP-D was first defined in (Zhou et al., 2023). In this problem a set of vehicles denoted $K_v = \{1, ..., k_v\}$ and a set of drones $K_d = \{1, ..., k_d\}$ have to serve a set of customers $V = \{1, ..., n\}$ from a depot 0. Let $D = \{0, ..., \Gamma\}$ be the set of all possible numbers of drones a vehicle can carry. The following assumptions are made:

- **(a)** Both drones and vehicles deliver packages, but some customers can not be delivered by drones.
- **(b)** A drone is limited by its capacity and its battery but is fully recharged upon reaching the vehicle.
- **(c)** A drone can only carry one package per flight, and is synchronised with a single assigned vehicle.
- **(d)** A drone can fly only when its vehicle is parked at a customer, but can take off multiple times from the same customer.
- **(e)** The vehicle total capacity is noted $Q$. The vehicle payload is impacted by the amount of carried drones, each drone and its necessary equipment having a weight $q_1^d$.
- **(f)** The vehicle service time at a customer and the drone deliveries from this customer happen in parallel.

We define $\mathcal{K}_i$, $i \in V$, the drone-reachable neighborhood of $i$, as the set of customers that can be delivered by drone from $i$. We call a drone schedule an ordered set of round trips of the drones from a customer. The problem is defined by the complete symmetric directed graph $G = (V_+, A)$, where $V_+ = \{0\} \cup V$ is the set of vertices and $A = \{(i, j) \mid i, j \in V_+,\ i \neq j\}$ is the set of arcs. A customer $i \in V$ has a demand $q_i$, a vehicle service time $s_i^v$, a drone service time $s_i^d$ and a deadline $\bar{d}_i$, which is the latest date a vehicle can arrive to serve the customer $i$. Travelling time between two nodes $i, j \in V_+$ is noted $c_{i,j}^v$ for a vehicle and $c_{i,j}^d$ for a drone.

We now present our mathematical formulation of the problem. Let $\mathcal{R}^d$, $d \in D$, be the set of all feasible routes of a vehicle carrying $d$ drones. For every $d \in D$, every $i \in V$, and every subset $K \subseteq \mathcal{K}_i$, let $S_{iK}^d$ be the set of all optimal drone schedule serving the set of customers $K$ with $d$ drones from the customer $i$. Set $S_{iK}^d$ may contain several schedules, because the arrival date of the truck at customer $i$ may vary and the customers have deadlines, hence the optimal solution may not be feasible for a given truck arrival date. Note that set $S_{iK}^d$ contains the Pareto front minimizing the maximal drone schedule duration, and maximizing the maximal truck arrival date. Let $b_{i,j}^r$ be equal to one if and only if the vehicle moves from $i$ to $j$ in route $r \in \mathcal{R}^d$. Let $a_s^r$ be equal to one if and only if the drones follow drone schedule $s \in S_{iK}^d$ when the vehicle is parked at customer $i \in V$ in route $r$. Let variable $\lambda_r$, $r \in \mathcal{R}^d$, $d \in D$ be equal to 1 if the route $r$ is used in the solution and 0 otherwise. We also define the following additional variables :

$x_{ij}^d$ — an integer variable which is equal to the number of times a vehicle

with $d \in D$ drones travels between node $i \in V_+$ and node $j \in V_+$, $i < j$, regardless of the order of visit.

$y^d$ — a continuous variable which is equal to the total truck immobility time in routes of vehicles with $d$ drones.

$z_j^d$ — a binary variable which is equal to one if and only if a vehicle with $d$ drones sends a drone to visit customer $j \in V$.

$u^d$ — the integer variable which is equal to the number of used vehicles with $d$ drones.

Variables with superscript $d$ belong to subproblem $d \in D$.
Using the previous notations, a set partitioning formulation is presented in two parts. The first part (i.e. equations (2) to (4)) uses the additional variables to model the problem. The second part (i.e. equations '(5) to (8)) maps the variable $\lambda_r$ to the additional variables.

$$\min \sum_{d \in D} \left( \sum_{i \in V_+} \sum_{j \in V_+} c_{i,j}^v \times x_{ij}^d + y^d \right) \tag{1}$$

$$\text{s.t.} \sum_{d \in D} \left( \sum_{j=1}^{i-1} \frac{1}{2} x_{ji}^d + \sum_{j=i+1}^{n} \frac{1}{2} x_{ij}^d + z_i^d \right) = 1, \qquad \forall i \in V, \tag{2}$$

$$\sum_{d \in D} u^d \leq k_v, \tag{3}$$

$$\sum_{d \in D} d u^d \leq k_d, \tag{4}$$

$$x_{ij}^d = \sum_{r \in \mathcal{R}^d} (b_{ij}^r + b_{ji}^r) \times \lambda_r, \qquad \forall d \in D,\ i, j \in V_+, i < j, \tag{5}$$

$$y^d = \sum_{r \in \mathcal{R}^d} \sum_{j \in V} \sum_{K \in \mathcal{K}_j} \sum_{s \in S_{jK}^d} (m_s \times a_s^r) \times \lambda_r, \quad \forall d \in D,\ \forall i \in V, \tag{6}$$

$$z_i^d = \sum_{r \in \mathcal{R}^d} \sum_{j \in V} \sum_{\substack{K \in \mathcal{K}_j : \\ i \in K}} \sum_{s \in S_{jK}^d} a_s^r \times \lambda_r, \qquad \forall d \in D,\ \forall i \in V \tag{7}$$

$$u^d = \sum_{r \in \mathcal{R}^d} \sum_{j \in V} b_{0j}^r \times \lambda_r, \qquad \forall d \in D, \tag{8}$$

$$\lambda_r \in \{0, 1\}, \qquad \forall d \in D,\ \forall r \in \mathcal{R}^d. \tag{9}$$

The objective function (1) and constraints (2)–(4) belong to the master. The objective function (1) minimizes the total duration of the routes. The constraints (2) ensure that each customer is visited exactly once. The constraints (3) and (4) verify that the solution uses a feasible number of trucks and drones. The constraints (5)–(8) define the mapping between the additional variables and the route variables.Note that additional variables are used mainly for the clarity

of the formulation and can be eliminated from the formulation using relations (5)–(8).

# 4 Pricing subproblem

We design the pricing subproblems as Resource Constrained Shortest Path Problem (RCSPP), where a feasible solution is a feasible route for the 2E-VRP-D. The objective of this problem is to find the shortest path between a source and a sink in a graph where each arc consumes resources and requires that the remaining resources lie between lower and upper limits.
We solve the subproblems using the bucket graph labelling algorithm presented in (Sadykov et al., 2021). The algorithm tries to extend partial paths called labels to form complete paths. Given their set of accumulated resources, labels are stored and extended according to so-called buckets, helping greatly with dominance checks. This algorithm supports ng-paths relaxation as well as the presence of limited rank-1 cuts.
We model a RCSP problem for each $d \in D$, in a graph defined as follows. Let $\mathcal{G}^d = (\mathcal{V}^d, \mathcal{A}^d)$ be a directed graph, where $\mathcal{V}^d = V_+ \cup V'_+$, with $V'_+$ a copy of $V_+$. Set $\mathcal{A}^d = A^{dT} \cup A^{dD}$ with $A^{dT}$ the set of arcs representing a truck delivery possible with $d$ drones and $A^{dD}$ the set of arcs representing possible drone deliveries with $d$ drones. The source of the graph is node $0'$ and the sink is node 0. Arcs in $A^{dD}$ go from a vertex $i \in V$ to the vertex $i' \in V'$, which is the copy of $i$. Let $S^d_{iK}$ be the set of drone routes with $d$ drone, delivering the set of customers $K \subseteq \mathcal{K}_i$ from $i$. For each $i \in V$, each possible subset $K \subseteq \mathcal{K}_i$, and each drone routes $s \in S^d_{iK}$, we define an arc in set $A^{dD}$, labelled $(i, i')^s$. Let $a^r_s$ be equal to one if and only if route $r \in \mathcal{R}^d$ follows this arc, i.e., the drones follow drone route $s \in S^d_{iK}$ in route $r$ when the vehicle is positioned at customer $i \in V$. Arcs in $A^{dT}$ go from a vertex $i' \in V'_+$ to a vertex $j \in V^+$. Let $b^r_{i,j}$ be equal to one if and only if route $r \in \mathcal{R}^d$ follows arc $(i', j)$. Note that the prime is omitted in the notation without ambiguity, as there is no arc between $i$ and $j$ with $i, j \in V_+$.

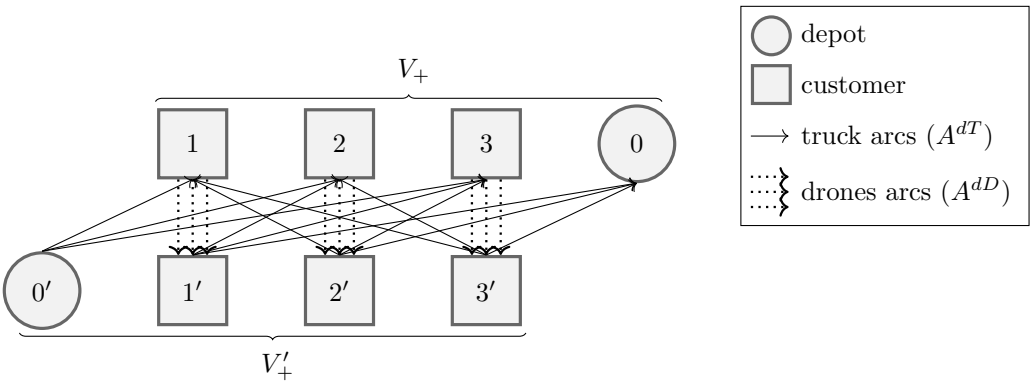


Figure 1: Example of a subproblem graph $\mathcal{G}^d$

Figure (1) shows an example of a subproblem graph. We represented only three arcs from $i \in V$ to $i' \in V'$ for clarity, but there should be an arc for all $s \in S^d_{iK}, \forall K \subseteq \mathcal{K}_i$.

## 4.1 Resources for the pricing subproblem

In our model, we use 2 resources, the duration $\beta_d$ of a route, and the capacity $\beta_c$ of a truck, and we define a pricing subproblem for each $d \in D$. The objective function is the minimization of the reduced cost. The reduced cost of a route $r$ with a cost $c_r$ is given by: $\bar{c}_r = c_r - \sum_{i \in V} \alpha_{ir} \times \pi_i - \lambda_0^v - d_r \times \lambda_0^d$ where $\alpha_{ir}$ is equal to one if and only if customer $i$ is visited in the route $r$, $d_r$ the number of drones used in route $r$, $\pi_i$ are the dual variables associated with constraints (2), $\lambda_0^v$ is the dual variable associated with constraint (3) and $\lambda_0^d$ is dual variable associated with constraint (4).
Let $[l_{a,r}, u_{a,r}]$ be the resource feasibility interval, for each $a \in \mathcal{A}^d$ and $r \in \{\beta_c, \beta_d\}$, i.e. the accumulated resource $r$ has to belong lay in the interval $[l_{a,r}, u_{a,r}]$ to travel through arc $a$. Let $q_{a,r}$ be the consumption of resource $r$ on arc $a$. For the resource capacity we set:

$$l_{a,\beta_c} = 0 \qquad \forall a \in \mathcal{A}^d \tag{10}$$

$$u_{a,\beta_c} = Q - d \times q_1^d \qquad \forall a \in \mathcal{A}^d \tag{11}$$

$$q_{a,\beta_c} = \begin{cases} 0 & \forall a = (i,0), i \in V' \\ q_j & \forall a = (i,j), i \in V'_+, j \in V \\ \sum_{k \in K} q_k & \forall a = (i,i')^s, i \in V, i' \in V', K \in \mathcal{K}_i, s \in S^d_{iK} \end{cases} \tag{12}$$

The truck starts with a remaining capacity equal to $Q - d \times q_1^d$. The remaining capacity of a truck must lie between 0 and $Q - d \times q_1^d$ on every arcs of the graph.

The consumption of remaining capacity is set to the demand of the customer served, or the sum of the demand of the customers delivered if the arc represents a drone route (i.e. $a = (i, i')i \in V, i' \in V'$), or 0 if the destination of the arc is node 0.
Let $\bar{a}_s$ be the maximal starting time of a solution $s$ that will be formally defined in section (4.2.1). For the resource duration we set:

$$l_{a,\beta_d} = 0 \qquad \forall a \in \mathcal{A}^d \tag{13}$$

$$u_{a,\beta_d} = \begin{cases} \bar{d}_j & \forall a = (i,j), i \in V'_+, j \in V_+ \\ \bar{a}_s & \forall a = (i,i')^s, i \in V, i' \in V', K \in \mathcal{K}_i, s \in S^d_{iK} \end{cases} \tag{14}$$

$$q_{a,\beta_d} = \begin{cases} c^v_{i,j} & \forall a = (i,j), i \in V'_+, j \in V_+ \\ \max(s^v_i, m_s) & \forall a = (i,i')^s, i \in V, i' \in V', K \in \mathcal{K}_i, s \in S^d_{iK} \end{cases} \tag{15}$$

The truck starts at time 0. A truck arc delivering a customer can only be used if the accumulated duration lie between 0 and the customer's deadline. A drone arc delivering a set of customers can only be used if the accumulated duration lie between 0 and the drone route's maximum starting time. The duration consumption of a drone arc is the maximum between the service time in truck at customer $i$ and the duration of the drone route, which are operated in parallel.

## 4.2 Drone routes enumeration

We enumerate all the feasible drone routes that might belong to an optimal solution. Therefore we must calculate the optimal drone routes from all customers $i \in V$ such that $\bar{f}^d_i = 1$, for any number of drones on the truck $d \in [1, ..., \Gamma]$, and for all sets of available customers, $K \subseteq \mathcal{K}_i$. Because of customers deadline, the optimal drone route delivering the set $K$ depends on the truck arrival time at $i$. Let $T_i = [c^v_{0,i}, ..., \bar{d}_i]$ be the set of all possible arrival times of a truck at customer $i$.

### 4.2.1 Scheduling problem

Since the drones can only deliver one customer before going back to the truck, and they are automatically fully recharged when reaching the truck, we can model the drone routing problem for a truck with $d$ drones arrived at a customer $i$ at time $a \in T_i$ and delivering set $K \subseteq \mathcal{K}_i$ customers as a scheduling problem with $d$ parallel identical machines, $|K|$ jobs with deadlines, a unique global release date $a$ and minimizing the makespan. The notation of the scheduling problem is $P|\bar{d}_j|C_{\max}$ (Graham et al., 1979). Note that the global release date $a$ does not appear in the notation, since the release date is unique for all the jobs, the machines will start working at time $a$. Hence, with a simple trick on the deadlines, we can get rid of the global release date.
In this scheduling problem, if there exists an optimal solution of the problem, there always exists an optimal solution without waiting time. Hence increasing

$a$ can only increase the makespan of a solution. Therefore, if an optimal solution $s^*$ at time $a$ remains feasible at time $a+\Delta$, with $\Delta > 0$ then $s^*$ is also an optimal solution at time $a + \Delta$. In other word, an optimal schedule can be optimal for a range of elements of $T_i$. We model the problem as follows:
Let $J_K^i$ be the set of jobs where each job represents the delivery of a customer $K \subseteq \mathcal{K}_i$ by drone from $i \in V$. We note $p_t$ the processing time of the job $t \in J_K^i$, $v_t$ the customer it delivers and $\bar{d}_t$ its deadline. We set $p_t = c^d_{i,v_t} + s^d_{v_t} + c^d_{v_t,i}$. To keep the consistency with the definition of deadline in the 2E-VRP-D, we set $\bar{d}_t = \bar{d}_{v_t} + s^d_{v_t} + c^d_{v_t,i} - a$.
As mentioned earlier, the set of solutions for a given $i \in V, d \in D, K \subseteq \mathcal{K}_i$ is noted $S^d_{iK}$. For $s \in S^d_{iK}$, let $m_s$ be its makespan, i.e. its completion time minus its starting time, $C^s_t$ be the completion time of a job $t \in J^i_K$ and $\bar{a}_s$ be its maximal starting time i.e. $\bar{a}_s = \min_{t \in J^i_K}(\bar{d}_t - C^s_t)$. Hence, $s$ remains optimal and feasible for all $a' \in [a, \bar{a}_s]$ and we don't have to calculate other optimal solutions in the interval.

### 4.2.2 Scheduling algorithm

We use a dynamic program to find all the solutions in a $S^d_{iK}$, $i \in V, d \in D, K \subseteq \mathcal{K}_i$. Because we solve a whole set $S^d_{iK}$, we have to find the solution for all $a \in T_i$. Hence we must find the Pareto front minimizing the makespan and maximizing the maximal starting time. The idea of the dynamic program is to check all possible combinations of solution of problems with fewer machines and subsets of jobs.
First we solve $S^1_{iK}$ $\forall i \in V_+, K \in \mathcal{K}_i$ using the Jackson's rule, also known as the Earliest Deadline First (Jackson, 1955), (Stankovic et al., 1998), and checking that the schedule is feasible; otherwise there is no solution in $S^1_{iK}$. Note that $|S^1_{iK}| \leq 1$. Let $K \subseteq \mathcal{K}_i$, $s \in S^d_{iK}$ and $Z$ be the set of jobs done on one of the machines of $s$. $\exists u \in S^1_{iZ}$ and $\exists v \in S^{d-1}_{iK\setminus Z}$ such that by combining $u$ and $v$, we obtain $s$. We write $s = (u, v)$. We have, $m_{(u,v)} = \max(m_u, m_v)$ and $\bar{a}_{(u,v)} = \min(\bar{a}_u, \bar{a}_v)$.

$$S^d_{iK} = \{(u,v) | u \in S^1_{iZ}, v \in S^{d-1}_{iK\setminus Z}, \forall s \in S^d_{iK}, m_s < m_{(u,v)} \Rightarrow \bar{a}_s < \bar{a}_{(u,v)}(1), \forall Z \subseteq K\} \tag{16}$$

The condition (1) ensure that $S^d_{iK}$ contains the Pareto front minimizing $m_s$ and maximizing $\bar{a}_s$. To avoid symmetries, we set the machine running set $Z$ to be the bottleneck machine inducing the makespan. This enable us to only check sets $Z$ such that $\sum_{t \in Z} p_t \geq \frac{1}{d} \times \sum_{t \in K} p_t$, because the makespan of a feasible solution can not be lower than $\frac{1}{d} \times \sum_{t \in K} p_t$.
$|\{Z \subseteq K\}| = 2^{|K|}$, hence the complexity of this dynamic program is exponential in $|K|$. The algorithm is output sensitive, as the output size can be exponential over the input size. The complexity of this algorithm is polynomial, $O(n^3)$ where $n$ is the size of its output. ($n$ schedules to calculate, for each we check

each combination of schedule with 1 machine and with n-1 machine: $O(n^2)$)

#### 4.2.3 Criteria for valid arcs elimination

In this section, we introduce two criteria that allow us to perform preprocessing to eliminate infeasible and dominated arcs in $A^{dD}$, from a subproblem graphs $\mathcal{G}$, $d \in D$.

- Any arc in $S^d_{iK}, d \in D, i \in V, K \subseteq \mathcal{K}_i$ is infeasible if the total payload is higher than the truck capacity, i.e. $q_i + d \times q^d_1 + \sum_{k \in K} q_k > Q$ because the scheduling solution is not feasible in the routing problem, thus the arc representing this schedule is infeasible.
- Let a truck with $d$ drones be parked at customer $i$. If there is a faster route (combining truck and drones) delivering set of customers $K$ than delivering all of them by drone from $i$ (i.e. using an appropriate drone schedule in $S^d_{iK}$), then we know that the corresponding solution in $S^d_{iK}$ will not be used in any optimal solution and we can remove it safely.

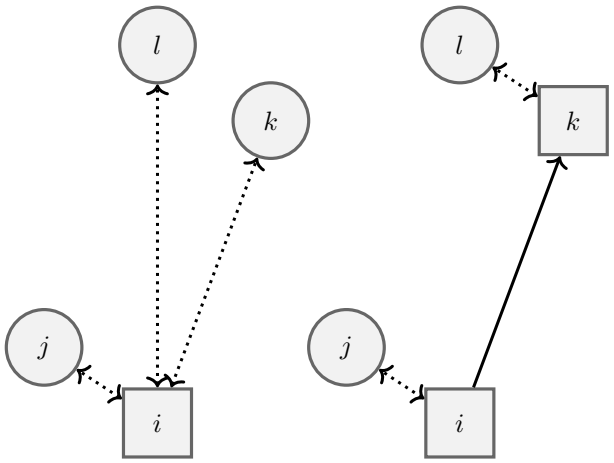


Figure 2: Example of an application of SPR criterion

In the figure 2 we see an example of an application of the SPR criterion. In the left side a single drone delivers customers $j$,$k$,$l$ from customer $l$. In the right side a drone delivers $j$ from $i$ then the truck moves to customer $k$ and then the drone delivers $l$ from $k$. The duration of the left partial solution is longer than the duration of the right side. It is not sufficient to remove $S^1_{i\{jkl\}}$, indeed we also need to verify that the right solution is feasible regardless of where the vehicle moves. This will be explained in detail in the following section. The SPR criterion generalize and tackle such solutions.

More formally, for a given $i \in V_+$, $d \in D$, $K \in \mathcal{K}_i$ and $j \in V \setminus K \cup \{i\}$, let $\text{SPR}(d, i, K, j)$ be the duration of the shortest partial route from $i$ to $j$ using a truck equipped with $d$ drones and delivering either by drone or by truck the customers of set $K$. If for any feasible $j$, $\text{SPR}(d, i, K, j) < m_s + c^v_{i,j}$, $s$ will not be used in an optimal solution.
We consider the following set of feasible $j$:

$$\left\{ j \in V_+ \setminus K \cup \{i\} | \left( c^v_{0,i} + \max(s^v_i, m_s) + c^v_{i,j} < \bar{d}_j \right) \wedge \left( q_i + d \times q^d_1 + q_j + \sum_{k \in K} q_k \leq Q \right) \right\} \tag{17}$$

In our computational experiment and because checking this criterion would be too time consuming, we only consider special cases where the truck is only allowed to visit a single customer of $K$, and the others must be delivered by drone. This criterion let us remove up to 90% of the arcs in $\mathcal{G}^1$, 61% in $\mathcal{G}^2$ and 6% in $\mathcal{G}^3$.

# 5 Branch-cut-and-price algorithm

Because it would be impractical to enumerate the set $\mathcal{R}^d, d \in D$ due to its exponential size over the number of customers we solve this formulation using a column and cut generation approach. We use a similar branch-and-cut-and-price (BCP) as in (Pessoa et al., 2020), with the following features. The BCP algorithm iteratively generates new variables (columns) to improve a linear programming (LP) relaxation of an integer program. Branching decisions are made based on the LP solution and cutting planes are added to strengthen the relaxation. Two primal heuristics are used in branch and bounds nodes. The first heuristic consists in solving the restricted master problem using a MIP solver and a time limit. The second heuristic is called the diving heuristic, where columns with largest fractional values are iteratively set to one and the remaining problem is solved using column generation.

## 5.1 Column generation

We use three-stage column generation. The pricing is solved using heuristic labelling algorithms in the two first stages as in (Sadykov et al., 2021), and using the exact labelling dynamic algorithm as in (Pessoa et al., 2020) and (Sadykov et al., 2021). The first two stages generate at most 30 columns, and the last stage at most 150. Automatic dual price smoothing stabilization is used, as in (Pessoa et al., 2018) to help the convergence of the process. The bucket-arc elimination procedure is also used, removing arcs proven not to belong to any optimal solution. After each call, we use a route enumeration technique similar to (Baldacci et al., 2008), which tries to enumerate all improving routes for a subproblem, and if it succeeds, the future pricing problems are solved by inspection of the enumerated routes. Furthermore, if the number of enumerated routes in all subproblem is less than a certain threshold (we use 5000 as threshold), all

the routes are added to the restricted master problem, and solved by the MIP solver.

## 5.2 Cutting planes

### 5.2.1 Adapted rounded capacity cuts

The rounded capacity cuts were first introduced in (Laporte and Nobert, 1983) for the constrained vehicle routing problem (CVRP). Let $C \subset V$ and let $h_r$ be the number of times $r \in \mathcal{R}^d$ visits $C$ i.e.

$$h_r = \frac{1}{2} \times \sum_{i \in V_+ \setminus C} \sum_{j \in C} b^r_{\min(i,j),\max(i,j)} + \sum_{i \in V_+ \setminus C} \sum_{\substack{K \in \mathcal{K}_i: \\ K \cap C \neq \varnothing}} \sum_{s \in S^d_{iK}} a^r_s \quad \forall d \in D, r \in \mathcal{R}^d \tag{18}$$

These valid inequalities can be formulated as follows:

$$\sum_{d \in D} \sum_{r \in \mathcal{R}^d:} \lambda_r \times h_r \geq \left\lceil \frac{\sum_{i \in C} q_i}{Q} \right\rceil, \quad C \subset V \tag{19}$$

However, because in this problem the drones take a large amount of the truck's capacity (about 20% per drone in the instances), we must take them into account. Let $k = \left\lfloor \frac{Q}{q^d_1} \right\rfloor$, and adapt the rounded capacity cuts as follows:

$$\sum_{d \in D} \sum_{r \in \mathcal{R}^d:} \lambda_r \times h_r \times (k - d) \geq \left\lceil \frac{\sum_{i \in C} q_i}{q^d_1} \right\rceil, \quad C \subset V \tag{20}$$

We will refer to these cuts as adapted rounded capacity cuts (ARCC). To separate the ARCC we use the greedy construction heuristic from (Lysgaard et al., 2004).

**Algorithm 1** Separation for the adapted rounded capacity cuts

**Require:** $t \in \mathbb{R}$
  **for** $i \in V$ **do**
    $C \leftarrow \{i\}$
    **while** $C \neq V$ **do**
      $C \leftarrow C \cup \{\arg\max_{j \in V \setminus C} \big(violation(C \cup \{j\})\big)\}$
      **if** $violation(C) > t$ **then**
        $addCut(C)$

We describe the separation algorithm in Algorithm (1), where $t$ is a user-defined threshold, the function $violation(C)$ calculates the difference between the right-hand side and the left-hand side in constraints (20) for set $C$ and $addCut(C)$ add the valid inequality to the problem. The algorithm can be explained as follows: starting with a single customer in the set, while not all customers are

in the set, we add a customer that is not already in the set and that maximizes the violation. If the violation of the set exceeds the threshold, we add the corresponding cut to the linear program. We generate at most 100 cuts per round.

### 5.2.2 Limited memory rank-one cuts

We use limited memory rank-one cuts (lm-R1C) similar to (Pecin et al., 2017), (Pessoa et al., 2020). The R1C uses a Chvátal-Gomory rounding on a weighted sum of a subset of constraints (2). When the coefficients are all equal, these cuts are called Subset-Row cuts (Jepsen et al., 2008).
The R1C are defined as follows. Let $C \subset V_+$, with multipliers $0 < \rho_i < 1 \quad \forall i \in C$ and let $h_i^r$ be the number of time the vertex $i$ is visited in route $r$:

$$h_i^r = \frac{1}{2} \sum_{j \in C} b^r_{\min(i,j),\max(i,j)} + \sum_{j \in V_+ \setminus C} \sum_{\substack{K \in \mathcal{K}_j: \\ i \in K}} \sum_{s \in S^d_{jK}} a_s^r \quad \forall d \in D, r \in \mathcal{R}^d \tag{21}$$

The following equation is a valid inequalities:

$$\sum_{d \in D} \sum_{r \in \mathcal{R}^d} \left\lfloor \rho_i \sum_{i \in C} h_i^r \right\rfloor \lambda_r \leq \left\lfloor \sum_{i \in C} \rho_i \right\rfloor \tag{22}$$

Because the R1C are non-robust constraints, we use the lm-R1C instead, where the set $C$ is paired with a memory set $M$ such that $C \subseteq M \subseteq V_+$. $\left\lfloor \rho_i \sum\limits_{i \in C} h_i^r \right\rfloor$ in equation (22) is replaced by $\alpha(C, M, \rho, P)$ and is calculated during the pricing computation as in algorithm 2.

**Algorithm 2** $\alpha(C, M, \rho, P = (v_0 = 0, v_1, ..., v_k, v_k + 1 = 0'))$

```
α ← 0, s ← 0
for i = 1 to k do
    if (v_{i-1}, v_i) ∉ M then
        s ← 0
    if v_i ∈ C then
        s ← s + ρ_{v_i}
        if s ≥ 1 then
            s ← s − 1, α ← α + 1
return α
```

If the partial path $P$ leaves the set $M$, all previous visits are forgotten, leading to an easier to compute but smaller coefficient. Note that if $M = V$, the lm-R1C are equivalent to the R1C. To separate those cuts, we find a violated constraint, and we calculate its minimal memory set such that the limited memory R1C has the same violation.

### 5.3 Branching

We obtain branching candidates from: the total number of drones ( $\sum_{d \in D} d \times u^d$), the total number of trucks ( $\sum_{d \in D} u^d$), the number of trucks with $d$ drones (variables $u$) and the truck travel variables (variables $x$).

We use a multi-phase strong branching procedure, similar to (Pecin et al., 2017), to choose the most promising branching candidate. The idea is to spend more time evaluating branching variables in the lowest depth of the branch-and-bound tree where each selection has a greater impact on the overall time, and spend less time as the depth increases, taking advantage of the history of previous evaluations.

In the first phase, half the candidates are chosen from history from previous calls of the strong branching procedure, while the others are chosen in a balanced way from all the branching strategies based on the distance between its value and the closest integer value. The candidates with the largest distances are selected.
In the second phase, selected candidates from the first phase are evaluated by solving the current restricted master linear program modified for each candidate, without column generation. We select the candidates using the product rule (Achterberg, 2007).
In the third phase, selected candidates from the second phase are evaluated by solving the restricted master linear program modified for each candidate, with heuristic column generation but without cut generation. We select the best candidate using the product rule.

## 6 Heuristic branch-cut-and-price

In this section, we propose a heuristic branch-cut-and-price based on the reduction of the size of the subproblems graphs. It has four parameters :

$\Gamma_h$ — The maximum number of drones a truck can carry.

$w_h$ — The maximum number of customers that can be delivered by drone from a customer in a scheduling.

$k_h$ — The maximum number of drone-reachable neighbors kept. Only the $k_h$ closest drone-reachable neighbors are kept.

$p_h$ — The number of schedule with earliest deadline kept for each $S^d_{iK}$.

$\Gamma_h$ decreases the number of subproblems. $w_h$ and $k_h$ reduce the number of drones arcs in each subproblems graphs. Note that the difference between $w_h$ and $k_h$ is that $k_h$ control which customer can be delivered by drone from a parked vehicle, while $w_h$ controls how many customers can be delivered by drone from a parked vehicle. Finally, $p_h$ reduces number of schedule and therefore the number of arcs in subproblems with multiple drones.
Let $\mathcal{K}_i^k$ be the set of the $k$ closest to $i$ customers in $\mathcal{K}_i$. More formally, $\mathcal{K}_i^k \subseteq \mathcal{K}_i$, $|\mathcal{K}_i^k| = k$ and $\forall k \in \mathcal{K}_i^k, k' \in \mathcal{K}_i \setminus \mathcal{K}_i^k, c_{ik}^d \leq c_{ik'}^d$. Let $S_{iK}^{dp}$ be the set of the $p_h$ solutions from $S_{iK}^d$ with earliest deadlines. To apply these rules, we transform each subproblems graphs $\mathcal{G}^d = (\mathcal{V}^d, \mathcal{A}^d), d \in [1, ..., \Gamma]$ with $\mathcal{A}^d = A^T \cup A^D$ from the exact branch-cut-and-price subproblems graphs as follows: we create the graphs $\mathcal{G}_h^d = (\mathcal{V}_h^d, \mathcal{A}_h^d), d \in [1, ..., \Gamma_h]$, where $\mathcal{V}_h^d = \mathcal{V}^d$ and $\mathcal{A}_h^d = A^T \cup \{(i, i')^s \in A^D, \forall K \subset \mathcal{K}_i^{k_h}, s \in S_{iK}^{dp_h} : |K| \leq w_h\}$

# 7 Computational experiments

## 7.1 Configuration

Our algorithms such as the dynamic program for the enumeration, the separation algorithm for the ARCC and the SPR criterion are coded in C++. We also used:

- IBM ILOG CPLEX 20.1 as the LP solver in the relaxed restricted master problem and as the solver for the enumerated MIPs.
- The generic branch-cut-and-price code from (Sadykov and Vanderbeck, 2021), with automatic dual price smoothing stabilization from (Pessoa et al., 2018), primal heuristics from (Sadykov et al., 2019). We also use the labelling algorithm code from (Sadykov et al., 2021).

The experiments presented in this paper were carried out:

- Using the PlaFRIM experimental testbed, supported by Inria, CNRS (LABRI and IMB), Université de Bordeaux, Bordeaux INP and Conseil Régional d'Aquitaine (see https://www.plafrim.fr).
- Running on 2.6 GHz 2x 18-core Cascade Lake Intel Xeon Skylake Gold 6240 CPU using a single thread with 5.3 Go memory.

## 7.2 Instances

To measure the performance of our algorithm, we used and extended the sets of instances from (Zhou et al., 2023), which is composed of modified instances from (Chen et al., 2021), and from (Solomon, 1987). In all these instances, the trucks and the drones follow the same distance metric. We also used clustered instances from (Gehring and Homberger, 1999) to experiment on larger instances, as they

are an extension of the Solomon (1987) instances. The maximal drone capacity is set to 10 kg as in (Chen et al., 2021), the service times with drone are set to $s_i^d = \lfloor 0.5 s_i^v \rfloor, i \in V$.
In the instances proposed in (Chen et al., 2021), an additional $\lfloor 0.2 \times |V| \rfloor$ customers are randomly selected to be unavailable for drone delivery. Table 3 describes the different data sets. For each data set, we give the name of the instances (Instances) where $X$ is the id of a particular instance of the set, the number of instances in the set (Size), the customer positioning (Pos), the distance metric (Dist), the original authors of the instances (Authors), the capacity of the vehicle ($Q$), the maximum number of drones allowed on a vehicle ($\Gamma$), the total number of vehicles ($k_v$) and the total number of drones ($k_d$). In column (Pos), "rand" stands for "random", and "clust" for "clustered". In column (Dist), "eucl" stands for "euclidean".

| Instances | $\|V\|$ | Size | Pos | Dist | Authors | $Q$ | $\Gamma$ | $k_v$ | $k_d$ |
|---|---|---|---|---|---|---|---|---|---|
| Small-Size Instances | | | | | | | | | |
| *Cardiff_10_X* | 10 | 20 | rand | real | Chen et al. (2021)* | 170 | 2 | 4 | 4 |
| *Cardiff_15_X* | 15 | 20 | rand | real | Chen et al. (2021) | 170 | 2 | 5 | 5 |
| *Cardiff_25_X* | 25 | 20 | rand | real | Chen et al. (2021) | 200 | 3 | 5 | 10 |
| *c10X_35* | 35 | 9 | clust | eucl | Solomon (1987) | 250 | 3 | 5 | 10 |
| Medium-Size Instances | | | | | | | | | |
| *Cardiff_40_X* | 40 | 20 | rand | real | Chen et al. (2021)* | 200 | 3 | 5 | 10 |
| *c10X_100* | 100 | 9 | clust | eucl | Solomon (1987) | 250 | 3 | 14 | 28 |
| *120_C1_2_X* | 120 | 10 | clust | eucl | Gehring and Homberger (1999) | 250 | 3 | 40 | 60 |
| *130_C1_2_X* | 130 | 10 | clust | eucl | Gehring and Homberger (1999) | 250 | 3 | 40 | 60 |
| *140_C1_2_X* | 140 | 10 | clust | eucl | Gehring and Homberger (1999) | 250 | 3 | 40 | 60 |
| *150_C1_2_X* | 150 | 10 | clust | eucl | Gehring and Homberger (1999) | 250 | 3 | 40 | 60 |

Table 3: Instances sets and their characteristics

Note that the set with 10 customers and the set with 40 customers are marked with '*' as they are not in (Chen et al., 2021), but are derived from sets with larger number of customers from(Chen et al., 2021), namely 15 and 50, by keeping only the 10 and 40 first customers.
The instances *c10X_N* are the 100-customers clustered instances called *c10X* in (Solomon, 1987), restricted to the $N$ first customers. We categorize instances with 35 or fewer customers as small, while instances with more than 35 customers will be referred to as medium. The instances *N_C1_2_X* are the 200-customer clustered instances called *C1_2_X* in (Gehring and Homberger, 1999), restricted to the first$N$ customers.

## 7.3 Results

In this section we present all the results obtained by setting the computation time limit at three hours per instance. Unless specified otherwise, the ARCC

and the SPR criterion are used. To measure the difficulty of an instance more precisely, we introduce $\hat{\mathcal{K}} = \max_{i \in V} |\mathcal{K}_i|$ the largest drone-reachable neighborhood of an instance, and $|\mathcal{A}^{\max}| = \max_{d \in D} |\mathcal{A}^d|$ the maximum number of arcs in a subproblem graph. $|\mathcal{A}^{\max}|$ is extremely dependent on $\hat{\mathcal{K}}$, as the number of arcs in a subproblem graph is exponential in the size of the drone-reachable neighborhoods. Gap is calculated as follows: $\frac{\text{PB}-\text{DB}}{\text{DB}}$, with PB the primal bound and DP the dual bound.
We present in the first subsection a comparison with the state-of-the-art results on the small-size instances. In the second subsection, we present the performance of our algorithm on medium size instances. In a third section, we present the results of the heuristic branch-cut-and-price on instances with large drone-reachable neighborhoods.

### 7.3.1 Performance of the exact BCP on the small-size instances

Table 4 shows the summary of the performance of our algorithm on the small-size instances and a comparison with the results obtained in (Zhou et al., 2023). The comparison is fair, as we use a computer with similar speed. For each data set, we give the name of the set (Instances), the number of instances solved optimally over the total number instances in the dataset (Solved), the average time in seconds required to solve the instances of the dataset (Time), the average number of nodes processed in the branch-cut-and-price algorithm (Nodes), and the maximum number of arcs in the subproblems graphs ($|\mathcal{A}^{\max}|$).

| Instances | Zhou et al. | | Our | | | |
|---|---|---|---|---|---|---|
| | Solved | Time (s) | Solved | Time (s) | Nodes | $\lvert\mathcal{A}^{\max}\rvert$ |
| *Cardiff_10_X* | 20/20 | 0.4 | 20/20 | 0.57 | 1 | 217 |
| *Cardiff_15_X* | 20/20 | 0.7 | 20/20 | 1.33 | 1 | 993 |
| *Cardiff_25_X* | 15/20 | 3561.5 | 20/20 | 86.25 | 2.5 | 5207 |
| *c10X_35* | 4/9 | 7408 | 9/9 | 556.30 | 8.12 | 2490 |

Table 4: Average exact performance comparison given small-size instances

Our branch-price-and-cut algorithm can solve the small-size instances very efficiently, indeed it solves all instances featured in (Zhou et al., 2023), including 10 solved for the first time. The computation time is improved by a factor 13 for the largest instances of the set. The number of nodes processed is very low. Our algorithm solves any of the small-size instances in less than half an hour. Therefore we increase the difficulty of the instances, by adding customers.

### 7.3.2 Performance of the exact BCP on the medium-size instances

Table 5 shows the summary of the performance of our algorithm on the medium-size instances. For each data set, we give the name of the set (Instances), the

maximum number of arcs in the subproblems graphs ($|\mathcal{A}^{\max}|$) of all instances, the average maximal size of drone-reachable neighborhood ($\hat{\mathcal{K}}$), the number of instances solved optimally over the total number instances (Solved), the number of instances for which no feasible solution has been found (No Sol), the average Gap of the instances for which a feasible solution has been found, the average time in seconds required to solve the instances of the dataset (Time) and the average number of nodes processed in the branch-cut-and-price algorithm (Nodes). The detailed results can be found in table 11, table 12 and table 13 in Appendix A.

| Instances | $\lvert\mathcal{A}^{\max}\rvert$ | $\hat{\mathcal{K}}$ | Solved | No Sol | Gap | Time (s) | Nodes |
|---|---|---|---|---|---|---|---|
| *Cardiff_40_X* | 78422.55 | 13 | 9/20 | 6 | 3.12 | 6241.96 | 12.2 |
| *c10X_100* | 19009.23 | 11 | 8/9 | 0 | 0.14 | 3134.03 | 30.4 |
| *120_C1_2_X* | 14931.50 | 3 | 7/10 | 3 | 0 | 4430.94 | 58.8 |
| *130_C1_2_X* | 17615.80 | 4 | 5/10 | 4 | 0.07 | 7081.01 | 77.4 |
| *140_C1_2_X* | 20367.80 | 4 | 3/10 | 4 | 0.27 | 8229.01 | 84.3 |
| *150_C1_2_X* | 23332.80 | 4 | 2/10 | 4 | 0.45 | 9384.44 | 107.8 |

Table 5: Average exact performance comparison given medium-size instances

The size of the drone-reachable neighborhood has a heavy impact on the number of arcs of the subproblems graphs, which strongly affect calculation time. Therefore, the size of the drone-reachable neighborhood should be considered when measuring the size of an instance. Our algorithm is particularly efficient to solve clustered instances: we solve most clustered instances with 120 customers and some with up to 150 customers. It is worth mentioning that in Gehring and Homberger (1999) instances, due to the weight of the packages, less than 50% of the customers are available for drone delivery, leading to very small drone-reachable neighborhoods. Increasing the number of customers in Gehring and Homberger (1999) instances slightly increases the gap, the number of processed nodes, and decreases the number of instances solved to optimality. However with randomized instances, the performance of our method deteriorates much faster. Already with 40 customers the algorithm can only find optimal solutions for half of the Chen et al. (2021) instances and cannot find any feasible solution for 6 instances. This can be explain by the large and highly interconnected drone-reachable neighborhoods in the randomized instances. Note that instances *Cardiff_40_X* in addition of having a high number of arcs in average, also have great disparity among the instances of the set, as the standard deviation of the maximum number of arc in a subproblem graph is 61705.8. In *Cardiff_40_X* instances, we only solve less than half instances. Due to the extremely high number of arcs, solving a node is slow. To obtain good quality solutions, we propose to use the heuristic defined in section 6.

### 7.3.3 Performance of the heuristic BCP on the random medium-size instances

To evaluate the performance of the proposed heuristic, we tested several different settings, which are reported in Table 6. The last row of the table corresponds to the performance of the exact BCP algorithm. In the absence of results in the literature, we compare all the results obtained by both the exact BCP and the heuristic BCP, and retain the BKS from one of these methods. The average gap between the heuristic solution and the BKS is then calculated as follows: $\frac{\text{PB}-\text{BKS}}{\text{BKS}}$, with PB the primal bound and BKS the best known solution value (Gap BKS).

Table 6 shows the summary of the performance of our heuristic branch-cut-and-price on the Chen et al. (2021) instances, with different sets of parameters. For each set of parameters $\Gamma_h$, $w_h$, $k_h$ and $p_h$ we give the average number of arcs in the largest subproblem of each instance ($|\mathcal{A}^{\max}|$), the average number of nodes processed in the branch-cut-and-price algorithm (Nodes), the average gap between the best known solution and the solution found in percentage for the instances where a solution was found (Gap BKS), the number of instances for which the solver reached the time limit (Time lim), the number of instances for which the solver was not able to find a feasible solution (No sol) and the average time solving time in seconds (Time). When the value of $p_h$ is inf, it means the parameter is left unrestricted. The detailed results can be found in table 14 in Appendix A.

Note that the maximum authorized number of drones per truck in these instances is 3, and the largest drone-neighborhood is 15, hence setting the related parameters to these values is similar to leaving them unrestricted.

| $\Gamma_h$ | $w_h$ | $k_h$ | $p_h$ | $\lvert\mathcal{A}^{\max}\rvert$ | Time lim | No sol | Gap BKS | Time (s) | Nodes |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 2 | 2 | 1 | 1796.30 | 0 | 0 | 4 | 392.72 | 5 |
| 2 | 3 | 3 | 1 | 1937.20 | 0 | 0 | 2.09 | 510.60 | 7.1 |
| 2 | 5 | 5 | 1 | 2641.55 | 0 | 0 | 1.05 | 360.44 | 5.6 |
| 2 | 8 | 5 | 1 | 6250.15 | 6 | 0 | 0.74 | 3738.86 | 19.4 |
| 3 | 5 | 3 | 1 | 2490.50 | 0 | 0 | 0.85 | 681.09 | 9.7 |
| 3 | 5 | 5 | 1 | 2664.95 | 0 | 0 | 0.67 | 381.38 | 5.4 |
| 3 | 5 | 5 | inf | 2773.70 | 0 | 0 | 0.67 | 348.23 | 5.1 |
| 3 | 8 | 5 | 1 | 6465.15 | 4 | 1 | 0.31 | 3496.81 | 25.2 |
| 3 | 8 | 5 | inf | 7767.95 | 6 | 2 | 0.26 | 3878.18 | 20.6 |
| 3 | 8 | 8 | 1 | 7088.75 | 6 | 2 | 0.18 | 4052.90 | 23.2 |
| 3 | 15 | 10 | 1 | 15222.95 | 5 | 0 | 0.04 | 4125.97 | 29.3 |
| 3 | 15 | 10 | inf | 78192.80 | 11 | 8 | 2.49 | 6203.80 | 12.5 |
| 3 | 15 | 15 | 1 | 43094.35 | 9 | 7 | 0.44 | 5625.45 | 13.5 |
| 3 | 15 | 15 | inf | 78422.55 | 11 | 6 | 1.89 | 6241.96 | 13 |

Table 6: Heuristic average performances given parameters on the Chen et al. (2021) instances with 40 customers

We can observe in table 6 that reducing the number of arcs in $\mathcal{A}^{dD}$ with $d \in [1, ..., \Gamma_h]$ by decreasing $w_h$ or $k_h$ can significantly decrease computation time, without decreasing the solutions quality much. For these instances, reducing $\Gamma_h$ has a heavy negative impact on the gap BKS, without having a positive impact on the computation time. It is worth mentioning that the heuristic with parameters $\Gamma_h = 3$, $w_h = 15$, $k_h = 10$, $p_h = 1$ gives a major improvement compared to the exact method, as both the average gap and the average computation time largely decrease. The heuristic with parameters $\Gamma_h = 3$, $w_h = 5$, $k_h = 5$ , $p_h = 1$ despite having a slightly larger gap has a very fast average solving time of less than 10 minutes. Unrestricted parameter $p_h$ has a major negative impact on the number of arcs and on the computation time, while having no positive impact on the gap. That is why we set it to one in most cases.

## 7.4 Sensitivity analysis

In this section we first present the impact of the ARCC and then we present the impact of the SPR criterion. We use the concept of cut off values. Cut off values are primal bound that the user provides to the algorithm, in order to prune the branch and bound tree. We provide the best known solution as cut off value. It can be use to prove the optimality of a solution, but here the goal is to avoid random performance of the primal heuristics, so we can fairly compare the performance with and without the tested improvement. Unless specified otherwise, the ARCC and the SPR criterion are used. We recall that the gap is calculated as follows: $\frac{\text{PB}-\text{DB}}{\text{DB}}$, with PB the primal bound and DP the dual bound.

### 7.4.1 Impact of the adapted rounded capacity cuts

In this section, we present the impact of the adapted rounded capacity cuts. The capacity constraint in the Chen et al. (2021) instances is not tight, as a large proportion of the trucks capacity is not used in optimal solution, therefore the ARCC are not useful on these instances, and the instances with 100 customers are all solved within 10 minutes, most of them solved at root node. For these reasons, we focus on the instances from Gehring and Homberger (1999) in this section. As shown in table 5, within the three-hour limit, the exact branch-cut-and-price does not find feasible solution for several instances of the Gehring and Homberger (1999) datasets. To find accurate cut off values, we run the exact BCP on these instances with a ten-hour time limit. However, we did not find feasible solution for all instances with 150 customers. Therefore, we present the results on the Gehring and Homberger (1999) instances with 120, 130 and 140 customers in table 7. We compare the performances with and without the ARCC. Table 7 shows the summary of the experiments to measure the impact of the ARCC. For each data set, we give the name of the set (Instances), whether the ARCC were used or not (ARCC), the average gap obtained on instances that are not solved optimally within the time limit (Gap), the average time in

seconds required to solve the instances of the dataset (Time) and the average number of nodes processed in the branch-cut-and-price algorithm (Nodes). Note that the number of optimal solutions found is omitted as it is the same for both. The detailed results can be found in the table 15 in Appendix A.

| Instances | ARCC | Gap | Time (s) | Nodes |
|---|---|---|---|---|
| *120_C1_2_X* | yes | 0.12 | 2437.30 | 31.6 |
| *120_C1_2_X* | no | 0.12 | 2451.13 | 40.6 |
| *130_C1_2_X* | yes | 0.28 | 5784.12 | 88.2 |
| *130_C1_2_X* | no | 0.30 | 5744.47 | 67.6 |
| *140_C1_2_X* | yes | 0.56 | 5955.80 | 75.4 |
| *140_C1_2_X* | no | 0.61 | 5925.10 | 72.6 |

Table 7: Average impact of the adapted rounded capacity cuts on the (Gehring and Homberger, 1999) instances

We can observe on table 7 that the ARCC have a positive impact of the gap, growing with the size of the instances. The ARCC have a negligible impact on the computation time, this is due to the fact that the capacity constraints are not very tight for these instances.

#### 7.4.2 Impact of the shorter partial route criterion

In this section we study the impact of the shorter partial route criterion both in terms of efficiency of the exact algorithm, and in terms of graph size. As the instances from (Gehring and Homberger, 1999) have a small drone-reachable neighborhood, this criterion does not have a significant impact on the number of arcs. For this reason, we perform the sensibility analysis only for the medium-size instances from (Chen et al., 2021) and (Solomon, 1987). We compare the performances of the algorithm with and without the SPR criterion. We recall that $\mathcal{G}^d$ is the directed graph representing the delivery possibilities for a truck with $d$ drones. Table 8 shows the summary of the experiments to measure the impact of the SPR criterion on the medium-size instances. For each data set, we give the name of the set (Instances), the number of arcs in subproblem graph and the proportion of arcs compared to without the SPR criterion ($|\mathcal{A}^1|$ ), ($|\mathcal{A}^2|$ ), ($|\mathcal{A}^3|$ ), the average Gap of the instances for which a feasible solution has been found, the average time in seconds required to solve the instances of the dataset (Time) and the average number of nodes processed in the branch-cut-and-price algorithm (Nodes).

| Instances | SPR | $\lvert\mathcal{A}^1\rvert$ | $\lvert\mathcal{A}^2\rvert$ | $\lvert\mathcal{A}^3\rvert$ | Gap | Time (s) | Nodes |
|---|---|---|---|---|---|---|---|
| *Cardiff_40_X* | yes | 4449.15 ( 20%) | 62169.85 ( 54%) | 72709.35 ( 97%) | 6.84 | 6376.47 | 10.7 |
| *Cardiff_40_X* | no | 22597.45 (100%) | 116217.80 (100%) | 75056.20 (100%) | 7.26 | 6318.14 | 8.8 |
| *c10X_100* | yes | 12902 ( 68%) | 18298.67 ( 95%) | 19009.23 (100%) | 0 | 206.81 | 5.23 |
| *c10X_100* | no | 19079 (100%) | 19326.56 (100%) | 19009.23 (100%) | 0 | 274.63 | 5.89 |

Table 8: Sensitivity analysis over SPR criterion

Note that using cut off values, the algorithm solves all *c10X_100* instances to optimality. The criterion removes a very large number of arc in $\mathcal{G}^1$ in both cases, less in $\mathcal{G}^2$ and almost none in $\mathcal{G}^3$. It removes more arcs when instances have larger drone-reachable neighborhood. For both data sets, the criterion accelerates most solved instances. The unsolved instances gaps are decreased of 0.5% in average. Note that although the criterion seems to have a negative impact on computation time in the *Cardiff_40_X* instances, in fact, the time decreases for 7 out of 9 instances solved optimally. Also the criterion decreases the number of nodes in the *c10X_100* instances. It is a positive impact, as it improves convergence speed. The criterion increases the number of nodes in instances *Cardiff_40_X*, which is also a positive result, as it is mainly due to the number of nodes in instances where the optimal solution was not found, demonstrating a faster node processing.

### 7.4.3 Impact of the restricted master heuristic

In this section, we present the impact of the restricted master heuristic. In this experiment, cut off values are not provided to the solver.

| Instances | RMH | Solved | No Sol | Gap | Time | Nodes |
|---|---|---|---|---|---|---|
| *Cardiff_40_X* | yes | 9/20 | 6 | 3.12 | 6241.96 | 12.2 |
| *Cardiff_40_X* | no | 11/20 | 2 | 3.64 | 5621.62 | 39.0 |
| *c10X_100* | yes | 8/9 | 0 | 0.14 | 3134.03 | 30.4 |
| *c10X_100* | no | 8/9 | 1 | 0.00 | 4184.41 | 57.0 |
| *120_C1_2_X* | yes | 7/10 | 3 | 0 | 4430.94 | 58.8 |
| *120_C1_2_X* | no | 6/10 | 2 | 0.19 | 6116.54 | 132.40 |

Table 9: Sensitivity analysis over restricted master heuristic.

We can observe that the number of nodes increases drastically when the restricted master heuristics are deactivated.

# 8 Conclusion

In this paper, we introduced a new branch-cut-and-price algorithm for the two-echelon vehicle routing problem with drones, based on the enumeration of the

drones round trips between customers and the vehicle parked at another customer. We presented ways to improve the method by reducing the number of drone trips enumerated, and by adapting the well-known rounded capacity cuts, supported by sensitivity analysis. We also proposed an efficient heuristic to tackle dense instances.

Extensive numerical experiments demonstrated the high efficiency of our method. It allows for solving all instances of literature containing up to 35 customers improving significantly computation times by a factor higher than 10 and optimally solving 10 of them for the first time. We also proposed 6 new sets of instances to promote research on VRP-D type problems. The experiments on our method shows that we solve all but one clustered instance with 100 customers, and more than half of the clustered instances with 120 customers. In the random dataset with 40 customers and large drone-reachable neighborhood, 11 instances remain open. To tackle these instances we proposed an heuristic branch-cut-and-price based on the elimination of certain arcs in the subproblems, and using the exact approach for solving. This approach shows great results, as the gaps with the best known solutions are very small. We also show the improvement caused by our adaptation of the rounded capacity cuts, and by the SPR criterion.

Testing these new instances highlighted some limitations of our branch-cut-and-price approach. On the largest instances, the size of the sub-problem graph becomes untractable due to the number of parallel arcs representing the drones round trips possibilities which is exponential given the number of customers that can be delivered by drone from a customer. A possible remedy is to model the scheduling problem as a part of the RCSP, to avoid the explicit enumeration of the scheduling solutions. This would require a special care for the resources, as only one drone is responsible for the duration of the makespan. In the same vein, it would be useful to find other rules of dominance for the drones arcs. In order to increase the efficiency of our algorithm, another possible enhancement concerns the primal heuristic to quickly find good quality primal bounds. It would also be very valuable to create instances more suitable for truck/drone deliveries, with different metrics for the drones and for the trucks and realistic features regarding last miles delivery service. One could also consider uncertainty for the travel with the truck. Comparing the gain using the 2E-VRP-D and the VRPD, on different sets of instances and objective functions could also be very interesting. In addition, most public authorities at national or metropolitan level are currently moving towards tighter regulation of deliveries in city centers. In particular, more restrictions are being imposed on access times for vehicles, and therefore also for customer deliveries. We could integrate time window constraints with non-zero ready times into our model. In particular, this extension will require a special care for the drones operations, as drones might need to wait between round trips. Another perspective is to apply this type of enumeration strategy to problems with a similar structure, such as the park-and-loop vehicle routing problem for instance, in which the driver of a vehicle parked at a client can make a walking tour to deliver nearby customers, to avoid network congestion and use of gas.

# Appendix A Detailed computational experiments

In the next tables we present the detailed results of our experiments. We present different information among: the name of the set (Instances), the maximum number of arcs in the subproblem $\mathcal{G}^\Gamma$ ($|\mathcal{A}^\Gamma|$) of all instances, the average maximal size of drone-reachable neighborhood ($\hat{\mathcal{K}}$), the number of instances solved optimally over the total number instances (Solved), the number of instances for which no feasible solution have been found (No Sol), the average Gap of the instances for which a feasible solution have been found, the average time in seconds required to solve the instances of the dataset (Time) and the average number of nodes processed in the branch-cut-and-price algorithm (Nodes).

Table 10: Detailed performance of the exact algorithm on the small-size instances

| Instance | $\lvert\mathcal{A}^{\max}\rvert$ | Root Gap | Root Time | Nodes | DB | PB | Gap | Time (s) |
|---|---|---|---|---|---|---|---|---|
| Cardiff_10_01 | 217 | 0 | 0.72 | 1 | 1223.62 | 1223.62 | 0 | 0.79 |
| Cardiff_10_02 | 134 | 0 | 0.447 | 1 | 1905.08 | 1905.08 | 0 | 0.49 |
| Cardiff_10_03 | 147 | 0 | 0.543 | 1 | 1412.67 | 1412.67 | 0 | 0.6 |
| Cardiff_10_04 | 136 | 0 | 0.572 | 1 | 1633.32 | 1633.32 | 0 | 0.62 |
| Cardiff_10_05 | 136 | 0 | 0.436 | 1 | 1682.93 | 1682.93 | 0 | 0.48 |
| Cardiff_10_06 | 155 | 0 | 0.591 | 1 | 1221.75 | 1221.75 | 0 | 0.65 |
| Cardiff_10_07 | 129 | 0 | 0.473 | 1 | 1931.75 | 1931.75 | 0 | 0.52 |
| Cardiff_10_08 | 143 | 0 | 0.615 | 1 | 1578.15 | 1578.15 | 0 | 0.67 |
| Cardiff_10_09 | 126 | 0 | 0.456 | 1 | 2552.03 | 2552.03 | 0 | 0.49 |
| Cardiff_10_10 | 125 | 0 | 0.46 | 1 | 2045.18 | 2045.18 | 0 | 0.5 |
| Cardiff_10_11 | 137 | 0 | 0.424 | 1 | 2095.25 | 2095.25 | 0 | 0.46 |
| Cardiff_10_12 | 161 | 0 | 0.578 | 1 | 1737.25 | 1737.25 | 0 | 0.63 |
| Cardiff_10_13 | 136 | 0 | 0.5 | 1 | 1485.88 | 1485.88 | 0 | 0.57 |
| Cardiff_10_14 | 125 | 0 | 0.441 | 1 | 1926.62 | 1926.62 | 0 | 0.5 |
| Cardiff_10_15 | 202 | 0 | 0.479 | 1 | 1814.33 | 1814.33 | 0 | 0.53 |
| Cardiff_10_16 | 148 | 0 | 0.602 | 1 | 1140.05 | 1140.05 | 0 | 0.69 |
| Cardiff_10_17 | 129 | 0 | 0.449 | 1 | 1733.83 | 1733.83 | 0 | 0.5 |
| Cardiff_10_18 | 137 | 0 | 0.456 | 1 | 1960.93 | 1960.93 | 0 | 0.51 |
| Cardiff_10_19 | 136 | 0 | 0.478 | 1 | 1533.67 | 1533.67 | 0 | 0.53 |
| Cardiff_10_20 | 135 | 0 | 0.502 | 1 | 1366.42 | 1366.42 | 0 | 0.59 |
| Cardiff_15_01 | 993 | 0 | 3.59 | 1 | 1332 | 1332 | 0 | 3.66 |
| Cardiff_15_02 | 383 | 0 | 0.773 | 1 | 1970.62 | 1970.62 | 0 | 0.83 |
| Cardiff_15_03 | 292 | 0 | 1.099 | 1 | 1971.38 | 1971.38 | 0 | 1.19 |
| Cardiff_15_04 | 320 | 0 | 0.799 | 1 | 2188.32 | 2188.32 | 0 | 0.84 |
| Cardiff_15_05 | 311 | 0 | 0.727 | 1 | 1831.27 | 1831.27 | 0 | 0.79 |
| Cardiff_15_06 | 408 | 0 | 1.395 | 1 | 1695.17 | 1695.17 | 0 | 1.5 |
| Cardiff_15_07 | 318 | 0 | 0.755 | 1 | 2050.67 | 2050.67 | 0 | 0.82 |
| Cardiff_15_08 | 318 | 0 | 1.091 | 1 | 1689.55 | 1689.55 | 0 | 1.17 |
| Cardiff_15_09 | 322 | 0 | 0.687 | 1 | 2765.68 | 2765.68 | 0 | 0.74 |

Continued on next page

Table 10: Detailed performance of the exact algorithm on the small-size instances

| Instance | $\lvert\mathcal{A}^{\max}\rvert$ | Root Gap | Root Time | Nodes | DB | PB | Gap | Time (s) |
|---|---|---|---|---|---|---|---|---|
| Cardiff_15_10 | 271 | 0 | 0.9 | 1 | 2984.98 | 2984.98 | 0 | 0.94 |
| Cardiff_15_11 | 319 | 0 | 0.824 | 1 | 2242.98 | 2242.98 | 0 | 0.89 |
| Cardiff_15_12 | 379 | 0 | 0.991 | 1 | 2155.83 | 2155.83 | 0 | 1.06 |
| Cardiff_15_13 | 369 | 0 | 1.512 | 1 | 1606.55 | 1606.55 | 0 | 1.59 |
| Cardiff_15_14 | 300 | 0 | 4.598 | 1 | 2369.88 | 2369.88 | 0 | 4.76 |
| Cardiff_15_15 | 728 | 0 | 1.228 | 1 | 1883.45 | 1883.45 | 0 | 1.29 |
| Cardiff_15_16 | 522 | 0 | 1.129 | 1 | 1756.85 | 1756.85 | 0 | 1.22 |
| Cardiff_15_17 | 293 | 0 | 0.583 | 1 | 1858.17 | 1858.17 | 0 | 0.64 |
| Cardiff_15_18 | 330 | 0 | 0.893 | 1 | 2288.73 | 2288.73 | 0 | 0.99 |
| Cardiff_15_19 | 303 | 0 | 0.69 | 1 | 1762.17 | 1762.17 | 0 | 0.76 |
| Cardiff_15_20 | 306 | 0 | 0.871 | 1 | 1697.1 | 1697.1 | 0 | 0.95 |
| Cardiff_25_01 | 1622 | 0 | 78.333 | 1 | 2233.87 | 2233.87 | 0 | 78.81 |
| Cardiff_25_02 | 1688 | 0.3 | 47.381 | 3 | 1979.95 | 1979.95 | 0 | 64.68 |
| Cardiff_25_03 | 985 | 0.7 | 28.089 | 3 | 2389.7 | 2389.7 | 0 | 40.9 |
| Cardiff_25_04 | 1280 | 5.8 | 31.931 | 3 | 2797.63 | 2797.63 | 0 | 66.87 |
| Cardiff_25_05 | 1481 | 0 | 27.47 | 1 | 2062.97 | 2062.97 | 0 | 27.72 |
| Cardiff_25_06 | 1140 | 0 | 24.194 | 1 | 2075.48 | 2075.48 | 0 | 24.52 |
| Cardiff_25_07 | 1212 | 1.3 | 31.748 | 3 | 2112.6 | 2112.6 | 0 | 44.93 |
| Cardiff_25_08 | 1119 | 0 | 20.74 | 1 | 2605.48 | 2605.48 | 0 | 20.91 |
| Cardiff_25_09 | 1634 | 1.4 | 33.46 | 3 | 2055.22 | 2055.22 | 0 | 48.93 |
| Cardiff_25_10 | 1605 | 2.7 | 75.233 | 5 | 2217.12 | 2217.12 | 0 | 296.04 |
| Cardiff_25_11 | 5207 | 0.7 | 85.909 | 3 | 2051.12 | 2051.12 | 0 | 99.0 |
| Cardiff_25_12 | 1245 | 7 | 18.103 | 5 | 4775.93 | 4775.93 | 0 | 157.68 |
| Cardiff_25_13 | 2730 | 0 | 57.655 | 1 | 2248.63 | 2248.63 | 0 | 58.04 |
| Cardiff_25_14 | 900 | 4 | 57.477 | 3 | 2530.68 | 2530.68 | 0 | 257.38 |
| Cardiff_25_15 | 1240 | 0.7 | 35.323 | 3 | 2871.6 | 2871.6 | 0 | 51.14 |
| Cardiff_25_16 | 1553 | 1.5 | 29.461 | 3 | 2503.27 | 2503.27 | 0 | 40.64 |
| Cardiff_25_17 | 3780 | 0 | 65.629 | 1 | 2275.83 | 2275.83 | 0 | 65.9 |
| Cardiff_25_18 | 922 | 0.6 | 56.438 | 3 | 2167.9 | 2167.9 | 0 | 125.93 |
| Cardiff_25_19 | 1796 | 1.7 | 55.092 | 3 | 2257.8 | 2257.8 | 0 | 71.74 |
| Cardiff_25_20 | 2449 | 0 | 82.744 | 1 | 2816.63 | 2816.63 | 0 | 83.14 |
| c101_35 | 2460 | 0.7 | 20.556 | 3 | 1896.5 | 1896.5 | 0 | 113.09 |
| c102_35 | 2460 | 0.5 | 27.488 | 5 | 1892.1 | 1892.1 | 0 | 729.18 |
| c103_35 | 2460 | 0.6 | 17.262 | 3 | 1889.2 | 1889.2 | 0 | 32.16 |
| c104_35 | 2472 | 0.8 | 50.63 | 25 | 1888.6 | 1888.6 | 0 | 1406.2 |
| c105_35 | 2464 | 0.5 | 38.436 | 3 | 1896.5 | 1896.5 | 0 | 257.74 |
| c106_35 | 2460 | 0.6 | 34.21 | 5 | 1896.5 | 1896.5 | 0 | 241.3 |
| c107_35 | 2490 | 0.6 | 37.053 | 3 | 1896.5 | 1896.5 | 0 | 162.77 |
| c108_35 | 2460 | 0.7 | 28.446 | 5 | 1893 | 1893 | 0 | 408.04 |
| c109_35 | 2490 | 0.5 | 121.423 | 21 | 1893 | 1893 | 0 | 1656.22 |

Note that the instance *Cardiff_40_11* is not feasible under our assumptions.

Table 11: Detailed performance of the exact algorithm on instances adapted from (Chen et al., 2021)

| Instance | $\lvert\mathcal{A}^{\max}\rvert$ | $\hat{\mathcal{K}}$ | Root Gap | Root Time | Nodes | DB | PB | Gap | Time (s) |
|---|---|---|---|---|---|---|---|---|---|
| Cardiff_40_01 | 13034 | 10 | 0.3 | 214.127 | 3 | 2931.88 | 2931.88 | 0 | 306.84 |
| Cardiff_40_02 | 95883 | 14 | 20.1 | 92.803 | 37 | 3044.72 | 3217.42 | 5.7 | 10800 |
| Cardiff_40_03 | 30318 | 12 | 2.7 | 168.228 | 3 | 2682.5 | 2682.5 | 0 | 351.02 |
| Cardiff_40_04 | 169680 | 15 | - | 401.227 | 9 | 4292.31 | - | - | 10800 |
| Cardiff_40_05 | 25801 | 11 | 1.7 | 329.68 | 11 | 3770.9 | 3770.9 | 0 | 2141.71 |
| Cardiff_40_06 | 26451 | 12 | 12.5 | 203.151 | 19 | 2909 | 2909 | 0 | 1633.04 |
| Cardiff_40_07 | 89523 | 15 | - | 674.039 | 3 | 2093.42 | - | - | 10800 |
| Cardiff_40_08 | 9510 | 9 | 0.6 | 191.295 | 3 | 3834.02 | 3834.02 | 0 | 358.6 |
| Cardiff_40_09 | 81734 | 14 | 35.1 | 381.086 | 11 | 3580.1 | 4484.18 | 25.3 | 10800 |
| Cardiff_40_10 | 55913 | 14 | 1.3 | 171.288 | 3 | 2692.08 | 2692.08 | 0 | 573.32 |
| Cardiff_40_11 | 48327 | 12 | 0 | 29.336 | 1 | unfeasible | unfeasible | 0 | 30.12 |
| Cardiff_40_12 | 38128 | 14 | 0.7 | 200.695 | 3 | 2642.87 | 2642.87 | 0 | 276.22 |
| Cardiff_40_13 | 121649 | 14 | 21.9 | 196.029 | 23 | 2723.1 | 2913.13 | 7 | 10800 |
| Cardiff_40_14 | 247124 | 15 | - | 1716.28 | 1 | 2236.37 | - | - | 10800 |
| Cardiff_40_15 | 33041 | 12 | 20.3 | 181.949 | 39 | 2606.74 | 2707.12 | 3.9 | 10800 |
| Cardiff_40_16 | 153756 | 15 | - | 785.406 | 5 | 2927.97 | - | - | 10800 |
| Cardiff_40_17 | 24463 | 13 | 3.2 | 278.477 | 23 | 2733.69 | 2781.87 | 1.8 | 10800 |
| Cardiff_40_18 | 93744 | 14 | - | 224.516 | 21 | 2707.07 | - | - | 10800 |
| Cardiff_40_19 | 10767 | 11 | 1.9 | 198.707 | 3 | 2784.05 | 2784.05 | 0 | 368.26 |
| Cardiff_40_20 | 85341 | 14 | - | 214.132 | 23 | 2649.34 | - | - | 10800 |

Table 12: Detailed performance of the exact algorithm on 100-customer instances

| Instance | $\lvert\mathcal{A}^{\max}\rvert$ | Root Gap | Root Time | Nodes | DB | PB | Gap | Time (s) |
|---|---|---|---|---|---|---|---|---|
| c101_100 | 18845 | 0.3 | 88.031 | 5 | 5298.76 | 5298.76 | 0 | 643.78 |
| c102_100 | 18845 | 0.4 | 86.379 | 7 | 5298.36 | 5298.36 | 0 | 751.62 |
| c103_100 | 18898 | 0.4 | 104.734 | 43 | 5296.46 | 5296.46 | 0 | 4008.0 |
| c104_100 | 20064 | 1.4 | 141.282 | 121 | 5274.95 | 5339.06 | 1.2 | 10800 |
| c105_100 | 18849 | 0.3 | 90.378 | 19 | 5298.76 | 5298.76 | 0 | 3249.86 |
| c106_100 | 18844 | 0.4 | 90.421 | 9 | 5298.76 | 5298.76 | 0 | 1144.6 |
| c107_100 | 18875 | 0.3 | 80.05 | 5 | 5298.76 | 5298.76 | 0 | 769.38 |
| c108_100 | 18845 | 0.4 | 93.02 | 15 | 5295.36 | 5295.36 | 0 | 1712.0 |
| c109_100 | 19018 | 0.4 | 125.412 | 49 | 5295.16 | 5295.16 | 0 | 5127.0 |

Table 13: Detailed performance of the exact algorithm on instances from (Gehring and Homberger, 1999)

| Instance | $\lvert\mathcal{A}^{\max}\rvert$ | $\hat{\mathcal{K}}$ | Root Gap | Root Time | Nodes | DB | PB | Gap | Time (s) |
|---|---|---|---|---|---|---|---|---|---|
| 120_C1_2_1 | 14931 | 3 | 0.4 | 45.306 | 7 | 9090 | 9090 | 0 | 251.78 |
| 120_C1_2_2 | 14931 | 3 | 0.3 | 87.072 | 29 | 8920.5 | 8920.5 | 0 | 1827.92 |
| 120_C1_2_3 | 14931 | 3 | 1035 | 185.251 | 143 | 8815.06 | - | - | 10800 |
| 120_C1_2_4 | 14931 | 3 | 1046.7 | 163.474 | 153 | 8728.17 | - | - | 10800 |
| 120_C1_2_5 | 14931 | 3 | 0.5 | 58.236 | 19 | 8986.2 | 8986.2 | 0 | 1454.68 |
| 120_C1_2_6 | 14931 | 3 | 0.9 | 91.461 | 17 | 8988.4 | 8988.4 | 0 | 1899.7 |
| 120_C1_2_7 | 14931 | 3 | 0.5 | 67.444 | 29 | 8958.6 | 8958.6 | 0 | 2438.2 |
| 120_C1_2_8 | 14931 | 3 | 0.3 | 166.804 | 9 | 8817.9 | 8817.9 | 0 | 642.36 |
| 120_C1_2_9 | 14931 | 3 | 0.3 | 171.819 | 51 | 8790.62 | 8790.62 | 0 | 3394.73 |
| 120_C1_2_10 | 14931 | 3 | 1044.6 | 193.899 | 131 | 8741.16 | - | - | 10800 |
| 130_C1_2_1 | 17614 | 4 | 0.6 | 61.628 | 11 | 9325.92 | 9325.92 | 0 | 744.67 |
| 130_C1_2_2 | 17614 | 4 | 0.3 | 105.671 | 11 | 9141.2 | 9141.2 | 0 | 787.67 |
| 130_C1_2_3 | 17614 | 4 | 1005.5 | 147.702 | 105 | 9054.61 | - | - | 10800 |
| 130_C1_2_4 | 17617 | 4 | 1015.2 | 162.767 | 125 | 8973.53 | - | - | 10800 |
| 130_C1_2_5 | 17614 | 4 | 0.5 | 87.648 | 29 | 9209.62 | 9209.62 | 0 | 3294.54 |
| 130_C1_2_6 | 17614 | 4 | 0.8 | 119.465 | 109 | 9222.4 | 9222.4 | 0 | 9784.0 |
| 130_C1_2_7 | 17614 | 4 | 0.6 | 122.121 | 21 | 9197.92 | 9197.92 | 0 | 2199.14 |
| 130_C1_2_8 | 17614 | 4 | 0.7 | 148.914 | 153 | 9053.06 | 9091.7 | 0.4 | 10800 |
| 130_C1_2_9 | 17614 | 4 | 1008.9 | 177.902 | 107 | 9029.23 | - | - | 10800 |
| 130_C1_2_10 | 17614 | 4 | 1013.6 | 215.358 | 103 | 8985.44 | - | - | 10800 |
| 140_C1_2_1 | 20366 | 4 | 0.6 | 82.444 | 51 | 9970.42 | 9970.42 | 0 | 2840.77 |
| 140_C1_2_2 | 20366 | 4 | 0.5 | 135.393 | 129 | 9786.12 | 9817.38 | 0.3 | 10800 |
| 140_C1_2_3 | 20366 | 4 | 931.4 | 192.591 | 125 | 9710.4 | - | - | 10800 |
| 140_C1_2_4 | 20369 | 4 | 939 | 181.705 | 87 | 9632.6 | - | - | 10800 |
| 140_C1_2_5 | 20366 | 4 | 0.4 | 94.988 | 25 | 9848.82 | 9848.82 | 0 | 1981.57 |
| 140_C1_2_6 | 20366 | 4 | 1.1 | 121.619 | 89 | 9812.82 | 9891.3 | 0.8 | 10800 |
| 140_C1_2_7 | 20366 | 4 | 0.3 | 152.451 | 21 | 9836.32 | 9836.32 | 0 | 1867.67 |
| 140_C1_2_8 | 20366 | 4 | 0.6 | 193.816 | 103 | 9707.63 | 9753.78 | 0.5 | 10800 |
| 140_C1_2_9 | 20366 | 4 | 933 | 215.832 | 107 | 9683.85 | - | - | 10800 |
| 140_C1_2_10 | 20366 | 4 | 939.2 | 211.671 | 105 | 9636.41 | - | - | 10800 |
| 150_C1_2_1 | 23331 | 4 | 0.4 | 95.102 | 35 | 10699.3 | 10699.3 | 0 | 1539.33 |
| 150_C1_2_2 | 23331 | 4 | 0.6 | 202.898 | 103 | 10551.2 | 10599.8 | 0.5 | 10800 |
| 150_C1_2_3 | 23331 | 4 | 1.6 | 160.827 | 137 | 10453.6 | 10611.5 | 1.5 | 10800 |
| 150_C1_2_4 | 23334 | 4 | 860.4 | 221.244 | 91 | 10425.2 | - | - | 10800 |
| 150_C1_2_5 | 23331 | 4 | 0.6 | 107.043 | 123 | 10582.5 | 10628.2 | 0.4 | 10800 |
| 150_C1_2_6 | 23331 | 4 | 0.5 | 170.548 | 159 | 10561.4 | 10595.1 | 0.3 | 10800 |
| 150_C1_2_7 | 23331 | 4 | 0.5 | 131.694 | 113 | 10602.5 | 10602.5 | 0 | 5905.0 |
| 150_C1_2_8 | 23331 | 4 | 856.6 | 215.009 | 105 | 10468.6 | - | - | 10800 |
| 150_C1_2_9 | 23331 | 4 | 860.5 | 236.564 | 105 | 10416.2 | - | - | 10800 |
| 150_C1_2_10 | 23331 | 4 | 865 | 210.899 | 107 | 10381 | - | - | 10800 |

Continued on next page

Table 13: Detailed performance of the exact algorithm on instances from (Gehring and Homberger, 1999)

| Instance | $\lvert\mathcal{A}^{\max}\rvert$ | $\hat{\mathcal{K}}$ | Root Gap | Root Time | Nodes | DB | PB | Gap | Time (s) |
|---|---|---|---|---|---|---|---|---|---|

Table 14: Detailed performance of the heuristic

| Instance | $\Gamma_h$ | $w_h$ | $k_h$ | $p_h$ | $\lvert\mathcal{A}^{\max}\rvert$ | Root Time | Nodes | PB | BKS | Gap | Time (s) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Cardiff_40_01 | 2 | 2 | 2 | 1 | 1795 | 83.27 | 3 | 3023.92 | 2931.88 | 3.14 | 108.18 |
| Cardiff_40_01 | 2 | 3 | 3 | 1 | 1931 | 89.94 | 3 | 2931.88 | 2931.88 | 0.00 | 115.41 |
| Cardiff_40_01 | 2 | 5 | 5 | 1 | 2491 | 111.19 | 1 | 2931.88 | 2931.88 | 0.00 | 112.07 |
| Cardiff_40_01 | 2 | 5 | 8 | 1 | 5261 | 147.21 | 1 | 2931.88 | 2931.88 | 0.00 | 147.91 |
| Cardiff_40_01 | 3 | 3 | 5 | 1 | 2366 | 138.05 | 1 | 2931.88 | 2931.88 | 0.00 | 138.87 |
| Cardiff_40_01 | 3 | 5 | 5 | 1 | 2501 | 113.29 | 3 | 2931.88 | 2931.88 | 0.00 | 141.00 |
| Cardiff_40_01 | 3 | 5 | 5 | inf | 2636 | 121.99 | 3 | 2931.88 | 2931.88 | 0.00 | 152.10 |
| Cardiff_40_01 | 3 | 5 | 8 | 1 | 5283 | 175.87 | 3 | 2931.88 | 2931.88 | 0.00 | 231.41 |
| Cardiff_40_01 | 3 | 5 | 8 | inf | 6601 | 213.12 | 3 | 2931.88 | 2931.88 | 0.00 | 276.06 |
| Cardiff_40_01 | 3 | 8 | 8 | 1 | 5715 | 180.78 | 1 | 2931.88 | 2931.88 | 0.00 | 181.72 |
| Cardiff_40_01 | 3 | 10 | 15 | inf | 16141 | 208.76 | 3 | 2931.88 | 2931.88 | 0.00 | 282.59 |
| Cardiff_40_01 | 3 | 15 | 10 | 1 | 10714 | 197.94 | 3 | 2931.88 | 2931.88 | 0.00 | 273.71 |
| Cardiff_40_01 | 3 | 15 | 15 | inf | 10714 | 190.00 | 3 | 2931.88 | 2931.88 | 0.00 | 252.57 |
| Cardiff_40_02 | 2 | 2 | 2 | 1 | 1798 | 74.15 | 1 | 3258.43 | 3217.42 | 1.27 | 74.65 |
| Cardiff_40_02 | 2 | 3 | 3 | 1 | 1940 | 76.30 | 1 | 3217.42 | 3217.42 | 0.00 | 76.65 |
| Cardiff_40_02 | 2 | 5 | 5 | 1 | 2691 | 79.61 | 1 | 3217.42 | 3217.42 | 0.00 | 80.37 |
| Cardiff_40_02 | 2 | 5 | 8 | 1 | 6439 | 119.73 | 3 | 3217.42 | 3217.42 | 0.00 | 368.83 |
| Cardiff_40_02 | 3 | 3 | 5 | 1 | 2521 | 94.48 | 3 | 3217.42 | 3217.42 | 0.00 | 193.52 |
| Cardiff_40_02 | 3 | 5 | 5 | 1 | 2707 | 99.99 | 1 | 3217.42 | 3217.42 | 0.00 | 100.56 |
| Cardiff_40_02 | 3 | 5 | 5 | inf | 2831 | 73.55 | 3 | 3217.42 | 3217.42 | 0.00 | 163.43 |
| Cardiff_40_02 | 3 | 5 | 8 | 1 | 6743 | 110.46 | 7 | 3217.42 | 3217.42 | 0.00 | 1222.35 |
| Cardiff_40_02 | 3 | 5 | 8 | inf | 7965 | 117.88 | 13 | 3217.42 | 3217.42 | 0.00 | 1812.41 |
| Cardiff_40_02 | 3 | 8 | 8 | 1 | 7431 | 110.16 | 11 | 3217.42 | 3217.42 | 0.00 | 1928.13 |
| Cardiff_40_02 | 3 | 10 | 15 | inf | 94606 | 104.14 | 37 | - | 3217.42 | - | 10800.00 |
| Cardiff_40_02 | 3 | 15 | 10 | 1 | 15732 | 143.74 | 5 | 3217.42 | 3217.42 | 0.00 | 1813.37 |
| Cardiff_40_02 | 3 | 15 | 15 | inf | 44541 | 71.34 | 43 | 3217.42 | 3217.42 | 0.00 | 10800.00 |
| Cardiff_40_03 | 2 | 2 | 2 | 1 | 1798 | 64.23 | 3 | 2761.12 | 2682.5 | 2.93 | 92.10 |
| Cardiff_40_03 | 2 | 3 | 3 | 1 | 1932 | 83.51 | 1 | 2686.92 | 2682.5 | 0.16 | 84.03 |
| Cardiff_40_03 | 2 | 5 | 5 | 1 | 2573 | 101.58 | 3 | 2682.5 | 2682.5 | 0.00 | 193.17 |
| Cardiff_40_03 | 2 | 5 | 8 | 1 | 6071 | 150.94 | 3 | 2682.5 | 2682.5 | 0.00 | 277.87 |
| Cardiff_40_03 | 3 | 3 | 5 | 1 | 2441 | 83.80 | 3 | 2682.5 | 2682.5 | 0.00 | 138.85 |
| Cardiff_40_03 | 3 | 5 | 5 | 1 | 2605 | 107.42 | 3 | 2682.5 | 2682.5 | 0.00 | 186.08 |
| Cardiff_40_03 | 3 | 5 | 5 | inf | 2676 | 119.35 | 1 | 2682.5 | 2682.5 | 0.00 | 119.95 |
| Cardiff_40_03 | 3 | 5 | 8 | 1 | 6311 | 126.94 | 3 | 2682.5 | 2682.5 | 0.00 | 252.25 |
| Cardiff_40_03 | 3 | 5 | 8 | inf | 7550 | 132.35 | 7 | 2682.5 | 2682.5 | 0.00 | 447.35 |

Continued on next page

Table 14: Detailed performance of the heuristic

| Instance | $\Gamma_h$ | $w_h$ | $k_h$ | $p_h$ | $\lvert\mathcal{A}^{\max}\rvert$ | Root Time | Nodes | PB | BKS | Gap | Time (s) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Cardiff_40_03 | 3 | 8 | 8 | 1 | 6983 | 141.43 | 5 | 2682.5 | 2682.5 | 0.00 | 470.14 |
| Cardiff_40_03 | 3 | 10 | 15 | inf | 32435 | 203.85 | 5 | 2682.5 | 2682.5 | 0.00 | 531.98 |
| Cardiff_40_03 | 3 | 15 | 10 | 1 | 14395 | 184.13 | 21 | 2682.5 | 2682.5 | 0.00 | 2524.46 |
| Cardiff_40_03 | 3 | 15 | 15 | inf | 20100 | 185.76 | 5 | 2682.5 | 2682.5 | 0.00 | 456.20 |
| Cardiff_40_04 | 2 | 2 | 2 | 1 | 1792 | 60.52 | 5 | 5259.18 | 5082.62 | 3.47 | 119.43 |
| Cardiff_40_04 | 2 | 3 | 3 | 1 | 1934 | 60.83 | 5 | 5239.48 | 5082.62 | 3.09 | 164.87 |
| Cardiff_40_04 | 2 | 5 | 5 | 1 | 2720 | 72.19 | 3 | 5239.48 | 5082.62 | 3.09 | 137.02 |
| Cardiff_40_04 | 2 | 5 | 8 | 1 | 7764 | 90.31 | 39 | 5245.75 | 5082.62 | 3.21 | 10800.00 |
| Cardiff_40_04 | 3 | 3 | 5 | 1 | 2573 | 79.86 | 7 | 5236.25 | 5082.62 | 3.02 | 224.70 |
| Cardiff_40_04 | 3 | 5 | 5 | 1 | 2781 | 69.25 | 7 | 5236.25 | 5082.62 | 3.02 | 196.73 |
| Cardiff_40_04 | 3 | 5 | 5 | inf | 2784 | 72.60 | 17 | 5236.25 | 5082.62 | 3.02 | 434.19 |
| Cardiff_40_04 | 3 | 5 | 8 | 1 | 8532 | 162.82 | 73 | 5142.45 | 5082.62 | 1.18 | 10800.00 |
| Cardiff_40_04 | 3 | 5 | 8 | inf | 8641 | 167.72 | 51 | 5178.7 | 5082.62 | 1.89 | 10800.00 |
| Cardiff_40_04 | 3 | 8 | 8 | 1 | 9603 | 115.39 | 49 | - | 5082.62 | - | 10800.00 |
| Cardiff_40_04 | 3 | 10 | 15 | inf | 169680 | 429.19 | 9 | - | 5082.62 | - | 10800.00 |
| Cardiff_40_04 | 3 | 15 | 10 | 1 | 26672 | 195.76 | 45 | 5082.62 | 5082.62 | 0.00 | 10800.00 |
| Cardiff_40_04 | 3 | 15 | 15 | inf | 146716 | 396.58 | 11 | - | 5082.62 | - | 10800.00 |
| Cardiff_40_05 | 2 | 2 | 2 | 1 | 1795 | 166.61 | 3 | 3812.85 | 3770.9 | 1.11 | 332.22 |
| Cardiff_40_05 | 2 | 3 | 3 | 1 | 1947 | 142.14 | 3 | 3812.85 | 3770.9 | 1.11 | 208.14 |
| Cardiff_40_05 | 2 | 5 | 5 | 1 | 2582 | 185.46 | 3 | 3770.9 | 3770.9 | 0.00 | 250.84 |
| Cardiff_40_05 | 2 | 5 | 8 | 1 | 5696 | 266.27 | 15 | 3770.9 | 3770.9 | 0.00 | 1436.28 |
| Cardiff_40_05 | 3 | 3 | 5 | 1 | 2453 | 207.88 | 3 | 3770.9 | 3770.9 | 0.00 | 273.77 |
| Cardiff_40_05 | 3 | 5 | 5 | 1 | 2609 | 213.88 | 3 | 3770.9 | 3770.9 | 0.00 | 269.74 |
| Cardiff_40_05 | 3 | 5 | 5 | inf | 2758 | 223.78 | 3 | 3770.9 | 3770.9 | 0.00 | 294.98 |
| Cardiff_40_05 | 3 | 5 | 8 | 1 | 5955 | 282.85 | 11 | 3770.9 | 3770.9 | 0.00 | 1134.08 |
| Cardiff_40_05 | 3 | 5 | 8 | inf | 7149 | 336.73 | 9 | 3770.9 | 3770.9 | 0.00 | 1313.44 |
| Cardiff_40_05 | 3 | 8 | 8 | 1 | 6492 | 300.18 | 13 | 3770.9 | 3770.9 | 0.00 | 1614.24 |
| Cardiff_40_05 | 3 | 10 | 15 | inf | 25801 | 319.38 | 7 | 3770.9 | 3770.9 | 0.00 | 1614.57 |
| Cardiff_40_05 | 3 | 15 | 10 | 1 | 12799 | 397.57 | 5 | 3770.9 | 3770.9 | 0.00 | 1000.81 |
| Cardiff_40_05 | 3 | 15 | 15 | inf | 16776 | 239.74 | 3 | 3770.9 | 3770.9 | 0.00 | 674.01 |
| Cardiff_40_06 | 2 | 2 | 2 | 1 | 1800 | 108.36 | 1 | 3013.15 | 2909 | 3.58 | 108.80 |
| Cardiff_40_06 | 2 | 3 | 3 | 1 | 1939 | 111.47 | 1 | 2909.0 | 2909 | 0.00 | 112.15 |
| Cardiff_40_06 | 2 | 5 | 5 | 1 | 2671 | 167.08 | 3 | 2909.0 | 2909 | 0.00 | 263.32 |
| Cardiff_40_06 | 2 | 5 | 8 | 1 | 6008 | 193.26 | 3 | 2909.0 | 2909 | 0.00 | 331.11 |
| Cardiff_40_06 | 3 | 3 | 5 | 1 | 2501 | 157.41 | 1 | 2909.0 | 2909 | 0.00 | 158.25 |
| Cardiff_40_06 | 3 | 5 | 5 | 1 | 2680 | 177.21 | 3 | 2909.0 | 2909 | 0.00 | 292.64 |
| Cardiff_40_06 | 3 | 5 | 5 | inf | 2858 | 174.96 | 1 | 2909.0 | 2909 | 0.00 | 175.67 |
| Cardiff_40_06 | 3 | 5 | 8 | 1 | 6041 | 239.07 | 5 | 2909.0 | 2909 | 0.00 | 536.92 |
| Cardiff_40_06 | 3 | 5 | 8 | inf | 7529 | 235.25 | 3 | 2909.0 | 2909 | 0.00 | 387.91 |
| Cardiff_40_06 | 3 | 8 | 8 | 1 | 6590 | 256.48 | 3 | 2909.0 | 2909 | 0.00 | 503.01 |
| Cardiff_40_06 | 3 | 10 | 15 | inf | 26451 | 211.78 | 7 | 2909.0 | 2909 | 0.00 | 865.43 |
| Cardiff_40_06 | 3 | 15 | 10 | 1 | 13386 | 262.61 | 3 | 2909.0 | 2909 | 0.00 | 425.62 |

Continued on next page

Table 14: Detailed performance of the heuristic

| Instance | $\Gamma_h$ | $w_h$ | $k_h$ | $p_h$ | $\lvert\mathcal{A}^{\max}\rvert$ | Root | Time | Nodes | PB | BKS | Gap | Time (s) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cardiff_40_06 | 3 | 15 | 15 | inf | 17176 | 175.53 | 5 | 2909.0 | 2909 | 0.00 | 591.90 | |
| Cardiff_40_07 | 2 | 2 | 2 | 1 | 1796 | 77.83 | 1 | 2510.25 | 2491.27 | 0.76 | 78.45 | |
| Cardiff_40_07 | 2 | 3 | 3 | 1 | 1947 | 81.88 | 1 | 2510.25 | 2491.27 | 0.76 | 82.69 | |
| Cardiff_40_07 | 2 | 5 | 5 | 1 | 2756 | 96.54 | 3 | 2508.6 | 2491.27 | 0.70 | 220.03 | |
| Cardiff_40_07 | 2 | 5 | 8 | 1 | 6981 | 141.71 | 7 | 2498.08 | 2491.27 | 0.27 | 711.42 | |
| Cardiff_40_07 | 3 | 3 | 5 | 1 | 2579 | 115.86 | 3 | 2508.6 | 2491.27 | 0.70 | 239.77 | |
| Cardiff_40_07 | 3 | 5 | 5 | 1 | 2781 | 112.91 | 3 | 2508.6 | 2491.27 | 0.70 | 243.98 | |
| Cardiff_40_07 | 3 | 5 | 5 | inf | 2884 | 116.35 | 3 | 2508.6 | 2491.27 | 0.70 | 205.18 | |
| Cardiff_40_07 | 3 | 5 | 8 | 1 | 7195 | 173.82 | 7 | 2498.08 | 2491.27 | 0.27 | 772.11 | |
| Cardiff_40_07 | 3 | 5 | 8 | inf | 9005 | 154.60 | 3 | 2498.08 | 2491.27 | 0.27 | 362.34 | |
| Cardiff_40_07 | 3 | 8 | 8 | 1 | 7704 | 170.55 | 3 | 2498.08 | 2491.27 | 0.27 | 394.30 | |
| Cardiff_40_07 | 3 | 10 | 15 | inf | 144776 | 701.13 | 3 | - | 2491.27 | - | 10800.00 | |
| Cardiff_40_07 | 3 | 15 | 10 | 1 | 18094 | 224.58 | 3 | 2491.27 | 2491.27 | 0.00 | 414.07 | |
| Cardiff_40_07 | 3 | 15 | 15 | inf | 70597 | 375.00 | 9 | - | 2491.27 | - | 10800.00 | |
| Cardiff_40_08 | 2 | 2 | 2 | 1 | 1798 | 67.13 | 5 | 3964.98 | 3834.02 | 3.42 | 288.31 | |
| Cardiff_40_08 | 2 | 3 | 3 | 1 | 1932 | 70.08 | 7 | 3962.05 | 3834.02 | 3.34 | 263.64 | |
| Cardiff_40_08 | 2 | 5 | 5 | 1 | 2590 | 100.03 | 3 | 3880.87 | 3834.02 | 1.22 | 194.73 | |
| Cardiff_40_08 | 2 | 5 | 8 | 1 | 5238 | 143.00 | 1 | 3834.02 | 3834.02 | 0.00 | 143.70 | |
| Cardiff_40_08 | 3 | 3 | 5 | 1 | 2444 | 138.10 | 3 | 3880.87 | 3834.02 | 1.22 | 223.55 | |
| Cardiff_40_08 | 3 | 5 | 5 | 1 | 2605 | 114.78 | 3 | 3870.2 | 3834.02 | 0.94 | 205.67 | |
| Cardiff_40_08 | 3 | 5 | 5 | inf | 2703 | 115.98 | 3 | 3870.2 | 3834.02 | 0.94 | 191.49 | |
| Cardiff_40_08 | 3 | 5 | 8 | 1 | 5467 | 226.22 | 5 | 3834.02 | 3834.02 | 0.00 | 725.98 | |
| Cardiff_40_08 | 3 | 5 | 8 | inf | 6622 | 235.22 | 5 | 3834.02 | 3834.02 | 0.00 | 683.52 | |
| Cardiff_40_08 | 3 | 8 | 8 | 1 | 5922 | 217.93 | 5 | 3834.02 | 3834.02 | 0.00 | 751.23 | |
| Cardiff_40_08 | 3 | 10 | 15 | inf | 9510 | 229.68 | 5 | 3834.02 | 3834.02 | 0.00 | 799.80 | |
| Cardiff_40_08 | 3 | 15 | 10 | 1 | 6941 | 181.43 | 3 | 3834.02 | 3834.02 | 0.00 | 384.79 | |
| Cardiff_40_08 | 3 | 15 | 15 | inf | 6941 | 201.69 | 5 | 3834.02 | 3834.02 | 0.00 | 631.42 | |
| Cardiff_40_09 | 2 | 2 | 2 | 1 | 1799 | 148.26 | 3 | 4122.2 | 3668.2 | 12.38 | 323.28 | |
| Cardiff_40_09 | 2 | 3 | 3 | 1 | 1950 | 105.21 | 3 | 4114.32 | 3668.2 | 12.16 | 172.51 | |
| Cardiff_40_09 | 2 | 5 | 5 | 1 | 2698 | 186.22 | 5 | 3977.35 | 3668.2 | 8.43 | 469.85 | |
| Cardiff_40_09 | 2 | 5 | 8 | 1 | 6482 | 180.19 | 33 | 3726.48 | 3668.2 | 1.59 | 10800.00 | |
| Cardiff_40_09 | 3 | 3 | 5 | 1 | 2530 | 224.38 | 3 | 3955.0 | 3668.2 | 7.82 | 367.35 | |
| Cardiff_40_09 | 3 | 5 | 5 | 1 | 2717 | 222.18 | 3 | 3924.03 | 3668.2 | 6.97 | 334.68 | |
| Cardiff_40_09 | 3 | 5 | 5 | inf | 2830 | 236.50 | 3 | 3924.03 | 3668.2 | 6.97 | 369.09 | |
| Cardiff_40_09 | 3 | 5 | 8 | 1 | 6656 | 421.09 | 49 | 3668.2 | 3668.2 | 0.00 | 5115.00 | |
| Cardiff_40_09 | 3 | 5 | 8 | inf | 8011 | 446.94 | 49 | 3719.7 | 3668.2 | 1.40 | 10800.00 | |
| Cardiff_40_09 | 3 | 8 | 8 | 1 | 7350 | 323.74 | 31 | 3668.2 | 3668.2 | 0.00 | 10800.00 | |
| Cardiff_40_09 | 3 | 10 | 15 | inf | 80282 | 380.71 | 15 | 4639.5 | 3668.2 | 26.48 | 10800.00 | |
| Cardiff_40_09 | 3 | 15 | 10 | 1 | 15941 | 361.56 | 65 | 3668.2 | 3668.2 | 0.00 | 10800.00 | |
| Cardiff_40_09 | 3 | 15 | 15 | inf | 42829 | 362.02 | 21 | 3876.85 | 3668.2 | 5.69 | 10800.00 | |
| Cardiff_40_10 | 2 | 2 | 2 | 1 | 1800 | 76.35 | 1 | 2769.2 | 2692.08 | 2.86 | 76.72 | |
| Cardiff_40_10 | 2 | 3 | 3 | 1 | 1950 | 76.03 | 3 | 2769.2 | 2692.08 | 2.86 | 114.10 | |

Continued on next page

Table 14: Detailed performance of the heuristic

| Instance | $\Gamma_h$ | $w_h$ | $k_h$ | $p_h$ | $\lvert\mathcal{A}^{\max}\rvert$ | Root Time | Nodes | PB | BKS | Gap | Time (s) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Cardiff_40_10 | 2 | 5 | 5 | 1 | 2669 | 85.22 | 1 | 2711.32 | 2692.08 | 0.71 | 86.05 |
| Cardiff_40_10 | 2 | 5 | 8 | 1 | 5995 | 141.77 | 3 | 2711.32 | 2692.08 | 0.71 | 276.20 |
| Cardiff_40_10 | 3 | 3 | 5 | 1 | 2515 | 99.54 | 1 | 2692.08 | 2692.08 | 0.00 | 100.29 |
| Cardiff_40_10 | 3 | 5 | 5 | 1 | 2690 | 100.58 | 1 | 2692.08 | 2692.08 | 0.00 | 100.99 |
| Cardiff_40_10 | 3 | 5 | 5 | inf | 2799 | 103.25 | 1 | 2692.08 | 2692.08 | 0.00 | 103.87 |
| Cardiff_40_10 | 3 | 5 | 8 | 1 | 6222 | 151.79 | 1 | 2692.08 | 2692.08 | 0.00 | 152.67 |
| Cardiff_40_10 | 3 | 5 | 8 | inf | 7334 | 173.19 | 1 | 2692.08 | 2692.08 | 0.00 | 174.24 |
| Cardiff_40_10 | 3 | 8 | 8 | 1 | 6779 | 136.26 | 1 | 2692.08 | 2692.08 | 0.00 | 136.99 |
| Cardiff_40_10 | 3 | 10 | 15 | inf | 55901 | 167.17 | 5 | 2692.08 | 2692.08 | 0.00 | 521.83 |
| Cardiff_40_10 | 3 | 15 | 10 | 1 | 12351 | 156.87 | 1 | 2692.08 | 2692.08 | 0.00 | 157.60 |
| Cardiff_40_10 | 3 | 15 | 15 | inf | 28248 | 210.33 | 1 | 2692.08 | 2692.08 | 0.00 | 211.15 |
| Cardiff_40_11 | 2 | 2 | 2 | 1 | 1792 | 5.47 | 1 | unfeasible | unfeasible | 0.00 | 5.69 |
| Cardiff_40_11 | 2 | 3 | 3 | 1 | 1921 | 4.57 | 1 | unfeasible | unfeasible | 0.00 | 4.79 |
| Cardiff_40_11 | 2 | 5 | 5 | 1 | 2619 | 9.09 | 1 | unfeasible | unfeasible | 0.00 | 9.75 |
| Cardiff_40_11 | 2 | 5 | 8 | 1 | 6236 | 10.38 | 1 | unfeasible | unfeasible | 0.00 | 10.86 |
| Cardiff_40_11 | 3 | 3 | 5 | 1 | 2467 | 8.26 | 1 | unfeasible | unfeasible | 0.00 | 8.76 |
| Cardiff_40_11 | 3 | 5 | 5 | 1 | 2635 | 8.60 | 1 | unfeasible | unfeasible | 0.00 | 9.02 |
| Cardiff_40_11 | 3 | 5 | 5 | inf | 2774 | 8.18 | 1 | unfeasible | unfeasible | 0.00 | 8.48 |
| Cardiff_40_11 | 3 | 5 | 8 | 1 | 6535 | 10.62 | 1 | unfeasible | unfeasible | 0.00 | 10.95 |
| Cardiff_40_11 | 3 | 5 | 8 | inf | 8065 | 12.39 | 1 | unfeasible | unfeasible | 0.00 | 13.12 |
| Cardiff_40_11 | 3 | 8 | 8 | 1 | 7188 | 13.01 | 1 | unfeasible | unfeasible | 0.00 | 13.37 |
| Cardiff_40_11 | 3 | 10 | 15 | inf | 48327 | 32.05 | 1 | unfeasible | unfeasible | 0.00 | 33.12 |
| Cardiff_40_11 | 3 | 15 | 10 | 1 | 15181 | 16.39 | 1 | unfeasible | unfeasible | 0.00 | 17.03 |
| Cardiff_40_11 | 3 | 15 | 15 | inf | 27035 | 23.10 | 1 | unfeasible | unfeasible | 0.00 | 23.63 |
| Cardiff_40_12 | 2 | 2 | 2 | 1 | 1792 | 54.70 | 3 | 2692.03 | 2642.87 | 1.86 | 72.94 |
| Cardiff_40_12 | 2 | 3 | 3 | 1 | 1929 | 53.43 | 3 | 2642.87 | 2642.87 | 0.00 | 66.04 |
| Cardiff_40_12 | 2 | 5 | 5 | 1 | 2577 | 60.04 | 3 | 2642.87 | 2642.87 | 0.00 | 105.03 |
| Cardiff_40_12 | 2 | 5 | 8 | 1 | 5911 | 99.59 | 3 | 2642.87 | 2642.87 | 0.00 | 145.91 |
| Cardiff_40_12 | 3 | 3 | 5 | 1 | 2435 | 81.01 | 3 | 2642.87 | 2642.87 | 0.00 | 129.47 |
| Cardiff_40_12 | 3 | 5 | 5 | 1 | 2596 | 85.92 | 3 | 2642.87 | 2642.87 | 0.00 | 140.96 |
| Cardiff_40_12 | 3 | 5 | 5 | inf | 2720 | 87.54 | 3 | 2642.87 | 2642.87 | 0.00 | 142.77 |
| Cardiff_40_12 | 3 | 5 | 8 | 1 | 5911 | 98.25 | 3 | 2642.87 | 2642.87 | 0.00 | 138.24 |
| Cardiff_40_12 | 3 | 5 | 8 | inf | 7484 | 109.45 | 3 | 2642.87 | 2642.87 | 0.00 | 168.09 |
| Cardiff_40_12 | 3 | 8 | 8 | 1 | 6505 | 109.39 | 3 | 2642.87 | 2642.87 | 0.00 | 159.03 |
| Cardiff_40_12 | 3 | 10 | 15 | inf | 81967 | 173.31 | 3 | 2642.87 | 2642.87 | 0.00 | 251.61 |
| Cardiff_40_12 | 3 | 15 | 10 | 1 | 14174 | 108.45 | 3 | 2642.87 | 2642.87 | 0.00 | 165.07 |
| Cardiff_40_12 | 3 | 15 | 15 | inf | 35393 | 157.84 | 3 | 2642.87 | 2642.87 | 0.00 | 217.98 |
| Cardiff_40_13 | 2 | 2 | 2 | 1 | 1798 | 76.77 | 1 | 2999.0 | 2827.82 | 6.05 | 77.11 |
| Cardiff_40_13 | 2 | 3 | 3 | 1 | 1935 | 74.68 | 1 | 2999.0 | 2827.82 | 6.05 | 75.12 |
| Cardiff_40_13 | 2 | 5 | 5 | 1 | 2676 | 116.97 | 1 | 2883.55 | 2827.82 | 1.97 | 117.87 |
| Cardiff_40_13 | 2 | 5 | 8 | 1 | 6437 | 141.25 | 57 | 2864.38 | 2827.82 | 1.29 | 4819.00 |
| Cardiff_40_13 | 3 | 3 | 5 | 1 | 2528 | 141.22 | 3 | 2881.38 | 2827.82 | 1.89 | 254.77 |

Continued on next page

Table 14: Detailed performance of the heuristic

| Instance | $\Gamma_h$ | $w_h$ | $k_h$ | $p_h$ | $\lvert\mathcal{A}^{\max}\rvert$ | Root Time | Nodes | PB | BKS | Gap | Time (s) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Cardiff_40_13 | 3 | 5 | 5 | 1 | 2717 | 101.87 | 1 | 2827.82 | 2827.82 | 0.00 | 102.28 |
| Cardiff_40_13 | 3 | 5 | 5 | inf | 2853 | 99.53 | 1 | 2827.82 | 2827.82 | 0.00 | 100.08 |
| Cardiff_40_13 | 3 | 5 | 8 | 1 | 6635 | 142.07 | 79 | 2827.82 | 2827.82 | 0.00 | 6390.00 |
| Cardiff_40_13 | 3 | 5 | 8 | inf | 8202 | 152.43 | 67 | - | 2827.82 | - | 10800.00 |
| Cardiff_40_13 | 3 | 8 | 8 | 1 | 7251 | 151.86 | 73 | 2827.82 | 2827.82 | 0.00 | 8276.00 |
| Cardiff_40_13 | 3 | 10 | 15 | inf | 121449 | 206.87 | 23 | 2913.13 | 2827.82 | 3.02 | 10800.00 |
| Cardiff_40_13 | 3 | 15 | 10 | 1 | 16815 | 208.34 | 113 | 2827.82 | 2827.82 | 0.00 | 8293.00 |
| Cardiff_40_13 | 3 | 15 | 15 | inf | 60077 | 156.43 | 41 | - | 2827.82 | - | 10800.00 |
| Cardiff_40_14 | 2 | 2 | 2 | 1 | 1796 | 75.74 | 15 | 3091.18 | 2910.9 | 6.19 | 960.53 |
| Cardiff_40_14 | 2 | 3 | 3 | 1 | 1929 | 93.03 | 45 | 3053.88 | 2910.9 | 4.91 | 2932.49 |
| Cardiff_40_14 | 2 | 5 | 5 | 1 | 2680 | 78.02 | 63 | 3028.5 | 2910.9 | 4.04 | 3480.14 |
| Cardiff_40_14 | 2 | 5 | 8 | 1 | 6624 | 117.48 | 63 | 3023.08 | 2910.9 | 3.85 | 10800.00 |
| Cardiff_40_14 | 3 | 3 | 5 | 1 | 2515 | 108.57 | 139 | 2957.28 | 2910.9 | 1.59 | 9388.00 |
| Cardiff_40_14 | 3 | 5 | 5 | 1 | 2705 | 112.64 | 55 | 2957.28 | 2910.9 | 1.59 | 3692.00 |
| Cardiff_40_14 | 3 | 5 | 5 | inf | 2819 | 120.63 | 39 | 2957.28 | 2910.9 | 1.59 | 2816.44 |
| Cardiff_40_14 | 3 | 5 | 8 | 1 | 6992 | 170.78 | 53 | - | 2910.9 | - | 10800.00 |
| Cardiff_40_14 | 3 | 5 | 8 | inf | 8269 | 192.80 | 63 | - | 2910.9 | - | 10800.00 |
| Cardiff_40_14 | 3 | 8 | 8 | 1 | 7750 | 187.90 | 81 | - | 2910.9 | - | 10800.00 |
| Cardiff_40_14 | 3 | 10 | 15 | inf | 246924 | 1796.16 | 1 | - | 2910.9 | - | 10800.00 |
| Cardiff_40_14 | 3 | 15 | 10 | 1 | 21082 | 102.81 | 113 | 2910.9 | 2910.9 | 0.00 | 7905.00 |
| Cardiff_40_14 | 3 | 15 | 15 | inf | 116234 | 1094.78 | 3 | - | 2910.9 | - | 10800.00 |
| Cardiff_40_15 | 2 | 2 | 2 | 1 | 1797 | 82.77 | 1 | 2760.08 | 2707.12 | 1.96 | 83.05 |
| Cardiff_40_15 | 2 | 3 | 3 | 1 | 1942 | 106.92 | 1 | 2760.08 | 2707.12 | 1.96 | 107.20 |
| Cardiff_40_15 | 2 | 5 | 5 | 1 | 2643 | 125.59 | 3 | 2707.12 | 2707.12 | 0.00 | 205.99 |
| Cardiff_40_15 | 2 | 5 | 8 | 1 | 6574 | 174.10 | 3 | 2707.12 | 2707.12 | 0.00 | 318.79 |
| Cardiff_40_15 | 3 | 3 | 5 | 1 | 2506 | 118.92 | 3 | 2707.12 | 2707.12 | 0.00 | 186.76 |
| Cardiff_40_15 | 3 | 5 | 5 | 1 | 2680 | 134.02 | 3 | 2707.12 | 2707.12 | 0.00 | 201.04 |
| Cardiff_40_15 | 3 | 5 | 5 | inf | 2730 | 132.68 | 3 | 2707.12 | 2707.12 | 0.00 | 198.58 |
| Cardiff_40_15 | 3 | 5 | 8 | 1 | 6721 | 172.12 | 3 | 2707.12 | 2707.12 | 0.00 | 330.60 |
| Cardiff_40_15 | 3 | 5 | 8 | inf | 8005 | 180.66 | 9 | 2707.12 | 2707.12 | 0.00 | 966.44 |
| Cardiff_40_15 | 3 | 8 | 8 | 1 | 7296 | 186.29 | 5 | 2707.12 | 2707.12 | 0.00 | 597.70 |
| Cardiff_40_15 | 3 | 10 | 15 | inf | 37408 | 185.77 | 41 | 2716.77 | 2707.12 | 0.36 | 10800.00 |
| Cardiff_40_15 | 3 | 15 | 10 | 1 | 15169 | 133.81 | 25 | 2707.12 | 2707.12 | 0.00 | 3442.63 |
| Cardiff_40_15 | 3 | 15 | 15 | inf | 23134 | 200.91 | 35 | 2707.12 | 2707.12 | 0.00 | 6986.00 |
| Cardiff_40_16 | 2 | 2 | 2 | 1 | 1796 | 152.34 | 3 | 4095.77 | 3353.42 | 22.14 | 325.30 |
| Cardiff_40_16 | 2 | 3 | 3 | 1 | 1938 | 139.04 | 3 | 3378.82 | 3353.42 | 0.76 | 246.38 |
| Cardiff_40_16 | 2 | 5 | 5 | 1 | 2634 | 157.76 | 3 | 3378.82 | 3353.42 | 0.76 | 290.33 |
| Cardiff_40_16 | 2 | 5 | 8 | 1 | 6861 | 251.24 | 37 | 3415.72 | 3353.42 | 1.86 | 10800.00 |
| Cardiff_40_16 | 3 | 3 | 5 | 1 | 2483 | 223.02 | 3 | 3378.82 | 3353.42 | 0.76 | 387.75 |
| Cardiff_40_16 | 3 | 5 | 5 | 1 | 2660 | 146.76 | 3 | 3353.42 | 3353.42 | 0.00 | 271.65 |
| Cardiff_40_16 | 3 | 5 | 5 | inf | 2771 | 161.23 | 3 | 3353.42 | 3353.42 | 0.00 | 291.54 |
| Cardiff_40_16 | 3 | 5 | 8 | 1 | 7072 | 278.11 | 49 | 3353.42 | 3353.42 | 0.00 | 10800.00 |

Continued on next page

Table 14: Detailed performance of the heuristic

| Instance | $\Gamma_h$ | $w_h$ | $k_h$ | $p_h$ | $\lvert\mathcal{A}^{\max}\rvert$ | Root | Time | Nodes | PB | BKS | Gap | Time (s) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cardiff_40_16 | 3 | 5 | 8 | inf | 8688 | 351.24 | 27 | 3366.97 | 3353.42 | 0.40 | 10800.00 | |
| Cardiff_40_16 | 3 | 8 | 8 | 1 | 7898 | 296.23 | 61 | 3410.52 | 3353.42 | 1.70 | 10800.00 | |
| Cardiff_40_16 | 3 | 10 | 15 | inf | 152985 | 835.01 | 5 | - | 3353.42 | - | 10800.00 | |
| Cardiff_40_16 | 3 | 15 | 10 | 1 | 19971 | 335.19 | 15 | 3366.97 | 3353.42 | 0.40 | 10800.00 | |
| Cardiff_40_16 | 3 | 15 | 15 | inf | 65872 | 480.78 | 9 | - | 3353.42 | - | 10800.00 | |
| Cardiff_40_17 | 2 | 2 | 2 | 1 | 1798 | 92.36 | 3 | 2831.3 | 2749.83 | 2.96 | 115.20 | |
| Cardiff_40_17 | 2 | 3 | 3 | 1 | 1939 | 87.32 | 1 | 2777.07 | 2749.83 | 0.99 | 87.80 | |
| Cardiff_40_17 | 2 | 5 | 5 | 1 | 2624 | 128.11 | 1 | 2749.83 | 2749.83 | 0.00 | 129.20 | |
| Cardiff_40_17 | 2 | 5 | 8 | 1 | 5241 | 189.10 | 5 | 2749.83 | 2749.83 | 0.00 | 676.60 | |
| Cardiff_40_17 | 3 | 3 | 5 | 1 | 2474 | 162.52 | 3 | 2749.83 | 2749.83 | 0.00 | 235.68 | |
| Cardiff_40_17 | 3 | 5 | 5 | 1 | 2642 | 160.60 | 1 | 2749.83 | 2749.83 | 0.00 | 161.46 | |
| Cardiff_40_17 | 3 | 5 | 5 | inf | 2738 | 164.62 | 1 | 2749.83 | 2749.83 | 0.00 | 165.30 | |
| Cardiff_40_17 | 3 | 5 | 8 | 1 | 5350 | 225.94 | 3 | 2749.83 | 2749.83 | 0.00 | 364.92 | |
| Cardiff_40_17 | 3 | 5 | 8 | inf | 6334 | 204.08 | 7 | 2749.83 | 2749.83 | 0.00 | 928.89 | |
| Cardiff_40_17 | 3 | 8 | 8 | 1 | 5787 | 191.46 | 3 | 2749.83 | 2749.83 | 0.00 | 307.75 | |
| Cardiff_40_17 | 3 | 10 | 15 | inf | 24600 | 280.43 | 33 | - | 2749.83 | - | 10800.00 | |
| Cardiff_40_17 | 3 | 15 | 10 | 1 | 9307 | 233.03 | 7 | 2749.83 | 2749.83 | 0.00 | 1051.50 | |
| Cardiff_40_17 | 3 | 15 | 15 | inf | 16770 | 256.00 | 9 | 2749.83 | 2749.83 | 0.00 | 4918.00 | |
| Cardiff_40_18 | 2 | 2 | 2 | 1 | 1787 | 120.56 | 3 | 2898.75 | 2888.65 | 0.35 | 264.26 | |
| Cardiff_40_18 | 2 | 3 | 3 | 1 | 1915 | 82.22 | 3 | 2888.65 | 2888.65 | 0.00 | 146.25 | |
| Cardiff_40_18 | 2 | 5 | 5 | 1 | 2579 | 88.51 | 5 | 2888.65 | 2888.65 | 0.00 | 268.48 | |
| Cardiff_40_18 | 2 | 5 | 8 | 1 | 6586 | 110.40 | 57 | 2941.17 | 2888.65 | 1.82 | 10800.00 | |
| Cardiff_40_18 | 3 | 3 | 5 | 1 | 2435 | 120.02 | 5 | 2888.65 | 2888.65 | 0.00 | 378.09 | |
| Cardiff_40_18 | 3 | 5 | 5 | 1 | 2593 | 109.83 | 5 | 2888.65 | 2888.65 | 0.00 | 355.63 | |
| Cardiff_40_18 | 3 | 5 | 5 | inf | 2711 | 117.91 | 3 | 2888.65 | 2888.65 | 0.00 | 190.47 | |
| Cardiff_40_18 | 3 | 5 | 8 | 1 | 6763 | 155.29 | 75 | 2888.65 | 2888.65 | 0.00 | 9097.00 | |
| Cardiff_40_18 | 3 | 5 | 8 | inf | 8311 | 165.19 | 37 | 2888.65 | 2888.65 | 0.00 | 4932.00 | |
| Cardiff_40_18 | 3 | 8 | 8 | 1 | 7454 | 153.60 | 57 | 2902.2 | 2888.65 | 0.47 | 10800.00 | |
| Cardiff_40_18 | 3 | 10 | 15 | inf | 93353 | 233.75 | 21 | - | 2888.65 | - | 10800.00 | |
| Cardiff_40_18 | 3 | 15 | 10 | 1 | 17614 | 94.82 | 93 | 2888.65 | 2888.65 | 0.00 | 10800.00 | |
| Cardiff_40_18 | 3 | 15 | 15 | inf | 48958 | 156.16 | 29 | - | 2888.65 | - | 10800.00 | |
| Cardiff_40_19 | 2 | 2 | 2 | 1 | 1800 | 100.68 | 3 | 2807.1 | 2784.05 | 0.83 | 206.02 | |
| Cardiff_40_19 | 2 | 3 | 3 | 1 | 1956 | 125.08 | 3 | 2807.1 | 2784.05 | 0.83 | 272.73 | |
| Cardiff_40_19 | 2 | 5 | 5 | 1 | 2690 | 181.37 | 3 | 2784.05 | 2784.05 | 0.00 | 345.11 | |
| Cardiff_40_19 | 2 | 5 | 8 | 1 | 5057 | 147.83 | 3 | 2784.05 | 2784.05 | 0.00 | 312.69 | |
| Cardiff_40_19 | 3 | 3 | 5 | 1 | 2541 | 174.93 | 3 | 2784.05 | 2784.05 | 0.00 | 298.16 | |
| Cardiff_40_19 | 3 | 5 | 5 | 1 | 2710 | 172.47 | 3 | 2784.05 | 2784.05 | 0.00 | 306.01 | |
| Cardiff_40_19 | 3 | 5 | 5 | inf | 2801 | 156.48 | 7 | 2784.05 | 2784.05 | 0.00 | 557.54 | |
| Cardiff_40_19 | 3 | 5 | 8 | 1 | 5075 | 158.20 | 3 | 2784.05 | 2784.05 | 0.00 | 261.90 | |
| Cardiff_40_19 | 3 | 5 | 8 | inf | 6009 | 167.29 | 3 | 2784.05 | 2784.05 | 0.00 | 297.71 | |
| Cardiff_40_19 | 3 | 8 | 8 | 1 | 5412 | 169.81 | 11 | 2784.05 | 2784.05 | 0.00 | 924.35 | |
| Cardiff_40_19 | 3 | 10 | 15 | inf | 11376 | 209.09 | 3 | 2784.05 | 2784.05 | 0.00 | 374.94 | |

Continued on next page

Table 14: Detailed performance of the heuristic

| Instance | $\Gamma_h$ | $w_h$ | $k_h$ | $p_h$ | $\lvert\mathcal{A}^{\max}\rvert$ | Root Time | Nodes | PB | BKS | Gap | Time (s) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Cardiff_40_19 | 3 | 15 | 10 | 1 | 7577 | 211.69 | 7 | 2784.05 | 2784.05 | 0.00 | 650.65 |
| Cardiff_40_19 | 3 | 15 | 15 | inf | 8484 | 189.95 | 3 | 2784.05 | 2784.05 | 0.00 | 345.95 |
| Cardiff_40_20 | 2 | 2 | 2 | 1 | 1798 | 104.94 | 41 | 3167.3 | 3083.32 | 2.72 | 4142.00 |
| Cardiff_40_20 | 2 | 3 | 3 | 1 | 1937 | 94.93 | 53 | 3167.3 | 3083.32 | 2.72 | 4879.00 |
| Cardiff_40_20 | 2 | 5 | 5 | 1 | 2668 | 104.20 | 3 | 3083.32 | 3083.32 | 0.00 | 249.27 |
| Cardiff_40_20 | 2 | 5 | 8 | 1 | 7541 | 126.64 | 51 | 3083.32 | 3083.32 | 0.00 | 10800.00 |
| Cardiff_40_20 | 3 | 3 | 5 | 1 | 2503 | 152.96 | 3 | 3083.32 | 3083.32 | 0.00 | 295.26 |
| Cardiff_40_20 | 3 | 5 | 5 | 1 | 2685 | 142.13 | 3 | 3083.32 | 3083.32 | 0.00 | 315.32 |
| Cardiff_40_20 | 3 | 5 | 5 | inf | 2798 | 138.74 | 3 | 3083.32 | 3083.32 | 0.00 | 283.32 |
| Cardiff_40_20 | 3 | 5 | 8 | 1 | 7844 | 162.42 | 71 | 3217.35 | 3083.32 | 4.35 | 10800.00 |
| Cardiff_40_20 | 3 | 5 | 8 | inf | 9585 | 183.54 | 51 | 3102.13 | 3083.32 | 0.61 | 10800.00 |
| Cardiff_40_20 | 3 | 8 | 8 | 1 | 8665 | 198.26 | 47 | 3109.15 | 3083.32 | 0.84 | 10800.00 |
| Cardiff_40_20 | 3 | 10 | 15 | inf | 89884 | 223.75 | 23 | - | 3083.32 | - | 10800.00 |
| Cardiff_40_20 | 3 | 15 | 10 | 1 | 20544 | 273.60 | 55 | 3089.5 | 3083.32 | 0.20 | 10800.00 |
| Cardiff_40_20 | 3 | 15 | 15 | inf | 55292 | 162.79 | 31 | - | 3083.32 | - | 10800.00 |

Table 15: Detailed experiments on the impact of the ARCC

| Instance | ARCC | Root Gap | Root Time | Nodes | DB | PB | Gap | Time (s) |
|---|---|---|---|---|---|---|---|---|
| 120_C1_2_01 | yes | 0 | 58.04 | 1 | 9090 | 9090 | 0 | 58.94 |
| 120_C1_2_01 | no | 0 | 43.98 | 1 | 9090 | 9090 | 0 | 44.63 |
| 120_C1_2_02 | yes | 0 | 162.93 | 1 | 8920.5 | 8920.5 | 0 | 163.87 |
| 120_C1_2_02 | no | 0 | 145.72 | 1 | 8920.5 | 8920.5 | 0 | 146.54 |
| 120_C1_2_03 | yes | 0.2 | 250.16 | 5 | 8843.52 | 8843.52 | 0 | 428.45 |
| 120_C1_2_03 | no | 0.3 | 185.19 | 9 | 8843.52 | 8843.52 | 0 | 433.68 |
| 120_C1_2_04 | yes | 0.7 | 240.19 | 187 | 8740.64 | 8784.92 | 0.51 | 10800 |
| 120_C1_2_04 | no | 0.6 | 165.93 | 241 | 8740 | 8784.92 | 0.51 | 10800 |
| 120_C1_2_05 | yes | 0 | 240.02 | 1 | 8986.2 | 8986.2 | 0 | 241.07 |
| 120_C1_2_05 | no | 0 | 237.6 | 1 | 8986.2 | 8986.2 | 0 | 238.7 |
| 120_C1_2_06 | yes | 0.3 | 395.59 | 9 | 8988.4 | 8988.4 | 0 | 847.64 |
| 120_C1_2_06 | no | 0.3 | 341.83 | 9 | 8988.4 | 8988.4 | 0 | 908.45 |
| 120_C1_2_07 | yes | 0.1 | 210.22 | 3 | 8958.6 | 8958.6 | 0 | 375.54 |
| 120_C1_2_07 | no | 0.1 | 242.94 | 3 | 8958.6 | 8958.6 | 0 | 419.64 |
| 120_C1_2_08 | yes | 0.1 | 229.57 | 11 | 8817.9 | 8817.9 | 0 | 375.16 |
| 120_C1_2_08 | no | 0.1 | 175.54 | 15 | 8817.9 | 8817.9 | 0 | 529.05 |
| 120_C1_2_09 | yes | 0 | 281.36 | 1 | 8790.62 | 8790.62 | 0 | 282.26 |
| 120_C1_2_09 | no | 0 | 189.73 | 1 | 8790.62 | 8790.62 | 0 | 190.56 |
| 120_C1_2_10 | yes | 0.7 | 264.75 | 97 | 8740.41 | 8794.92 | 0.62 | 10800 |
| 120_C1_2_10 | no | 0.7 | 155.31 | 125 | 8737.92 | 8794.92 | 0.65 | 10800 |
| 130_C1_2_01 | yes | 0 | 193.32 | 1 | 9325.92 | 9325.92 | 0 | 194.05 |

Continued on next page

Table 15: Detailed experiments on the impact of the ARCC

| Instance | ARCC | Root Gap | Root Time | Nodes | DB | PB | Gap | Time (s) |
|---|---|---|---|---|---|---|---|---|
| 130_C1_2_01 | no | 0 | 188.75 | 1 | 9325.92 | 9325.92 | 0 | 189.49 |
| 130_C1_2_02 | yes | 0 | 201.23 | 1 | 9141.2 | 9141.2 | 0 | 202.12 |
| 130_C1_2_02 | no | 0 | 169.74 | 1 | 9141.2 | 9141.2 | 0 | 170.74 |
| 130_C1_2_03 | yes | 0.7 | 198.25 | 241 | 9056.65 | 9119 | 0.69 | 10800 |
| 130_C1_2_03 | no | 0.7 | 199.17 | 167 | 9059.79 | 9119 | 0.65 | 10800 |
| 130_C1_2_04 | yes | 0.8 | 254.91 | 223 | 8993.56 | 9056.21 | 0.70 | 10800 |
| 130_C1_2_04 | no | 0.8 | 176.29 | 101 | 8989.81 | 9056.21 | 0.74 | 10800 |
| 130_C1_2_05 | yes | 0 | 264.44 | 1 | 9209.62 | 9209.62 | 0 | 265.45 |
| 130_C1_2_05 | no | 0 | 214.59 | 1 | 9209.62 | 9209.62 | 0 | 215.36 |
| 130_C1_2_06 | yes | 0.3 | 336.13 | 33 | 9222.4 | 9222.4 | 0 | 2152.1 |
| 130_C1_2_06 | no | 0.3 | 403.74 | 15 | 9222.4 | 9222.4 | 0 | 1633.03 |
| 130_C1_2_07 | yes | 0 | 687.49 | 3 | 9197.92 | 9197.92 | 0 | 1027.5 |
| 130_C1_2_07 | no | 0 | 771.28 | 3 | 9197.92 | 9197.92 | 0 | 1236.02 |
| 130_C1_2_08 | yes | 0.4 | 375.96 | 141 | 9082.79 | 9091.7 | 0.10 | 10800 |
| 130_C1_2_08 | no | 0.3 | 432.57 | 137 | 9066.88 | 9091.7 | 0.27 | 10800 |
| 130_C1_2_09 | yes | 0.6 | 261.74 | 119 | 9028.84 | 9077.5 | 0.54 | 10800 |
| 130_C1_2_09 | no | 0.6 | 236.97 | 129 | 9029.86 | 9077.5 | 0.53 | 10800 |
| 130_C1_2_10 | yes | 0.8 | 277.84 | 119 | 8992 | 9060.5 | 0.76 | 10800 |
| 130_C1_2_10 | no | 0.8 | 193.69 | 121 | 8990.42 | 9060.5 | 0.78 | 10800 |
| 140_C1_2_01 | yes | 0 | 268.45 | 1 | 9970.42 | 9970.42 | 0 | 269.33 |
| 140_C1_2_01 | no | 0 | 224.69 | 3 | 9970.42 | 9970.42 | 0 | 259.88 |
| 140_C1_2_02 | yes | 0.4 | 323.74 | 53 | 9817.38 | 9817.38 | 0 | 3780.0 |
| 140_C1_2_02 | no | 0.4 | 241.88 | 53 | 9817.38 | 9817.38 | 0 | 3656.0 |
| 140_C1_2_03 | yes | 0.8 | 271.7 | 115 | 9722.55 | 9784.7 | 0.64 | 10800 |
| 140_C1_2_03 | no | 0.8 | 174.26 | 119 | 9715.29 | 9784.7 | 0.71 | 10800 |
| 140_C1_2_04 | yes | 1.4 | 288.78 | 143 | 9637.97 | 9762.48 | 1.29 | 10800 |
| 140_C1_2_04 | no | 1.4 | 220.5 | 119 | 9637.72 | 9762.48 | 1.29 | 10800 |
| 140_C1_2_05 | yes | 0 | 262.62 | 1 | 9848.82 | 9848.82 | 0 | 263.61 |
| 140_C1_2_05 | no | 0 | 219.09 | 1 | 9848.82 | 9848.82 | 0 | 220.18 |
| 140_C1_2_06 | yes | 0.1 | 638.89 | 3 | 9853.5 | 9853.5 | 0 | 881.09 |
| 140_C1_2_06 | no | 0.1 | 598.12 | 3 | 9853.5 | 9853.5 | 0 | 809.91 |
| 140_C1_2_07 | yes | 0 | 325.88 | 3 | 9836.32 | 9836.32 | 0 | 363.95 |
| 140_C1_2_07 | no | 0 | 229.26 | 3 | 9836.32 | 9836.32 | 0 | 304.98 |
| 140_C1_2_08 | yes | 0.4 | 374.68 | 165 | 9726.07 | 9754.28 | 0.29 | 10800 |
| 140_C1_2_08 | no | 0.5 | 274.15 | 129 | 9716.74 | 9754.28 | 0.39 | 10800 |
| 140_C1_2_09 | yes | 0.8 | 271.95 | 157 | 9684.21 | 9754 | 0.72 | 10800 |
| 140_C1_2_09 | no | 0.9 | 258.01 | 181 | 9690.01 | 9754 | 0.66 | 10800 |
| 140_C1_2_10 | yes | 2.9 | 296.33 | 113 | 9639.97 | 9900 | 2.70 | 10800 |
| 140_C1_2_10 | no | 2.9 | 185.31 | 115 | 9638.62 | 9900 | 2.71 | 10800 |

Table 16: Detailed experiments on the impact of the SPR

| Instance | SPR | $\lvert\mathcal{A}^{\max}\rvert$ | Root Gap | Root Time | Nodes | DB | PB | Gap | Time (s) |
|---|---|---|---|---|---|---|---|---|---|
| Cardiff_40_01 | yes | 13034 | 0 | 268.41 | 1 | 2931.88 | 2931.88 | 0 | 269.49 |
| Cardiff_40_01 | no | 13174 | 0 | 297.93 | 1 | 2931.88 | 2931.88 | 0 | 298.89 |
| Cardiff_40_02 | yes | 95883 | 20.1 | 88.12 | 45 | 2827.85 | 3217.43 | 13.8 | 10800 |
| Cardiff_40_02 | no | 97648 | 20.1 | 117.31 | 29 | 2774.49 | 3217.42 | 16 | 10800 |
| Cardiff_40_03 | yes | 30318 | 0 | 400.03 | 1 | 2682.5 | 2682.5 | 0 | 401.32 |
| Cardiff_40_03 | no | 30721 | 9.1 | 264.48 | 3 | 2682.5 | 2682.5 | 0 | 542.77 |
| Cardiff_40_04 | yes | 169680 | 20.2 | 317.5 | 9 | 4282.36 | 5082.63 | 18.7 | 10800 |
| Cardiff_40_04 | no | 178022 | 20.2 | 478.01 | 7 | 4281.08 | 5082.63 | 18.7 | 10800 |
| Cardiff_40_05 | yes | 25801 | 1.6 | 543.24 | 27 | 3770.9 | 3770.9 | 0 | 6574.0 |
| Cardiff_40_05 | no | 26586 | 1.6 | 603.97 | 23 | 3770.9 | 3770.9 | 0 | 4977.0 |
| Cardiff_40_06 | yes | 26451 | 0 | 491.63 | 1 | 2909 | 2909 | 0 | 492.73 |
| Cardiff_40_06 | no | 27449 | 0 | 559.3 | 1 | 2909 | 2909 | 0 | 560.32 |
| Cardiff_40_07 | yes | 89523 | 19.8 | 696.68 | 3 | 2093.42 | 2491.28 | 19 | 10800 |
| Cardiff_40_07 | no | 91461 | 19.8 | 1008.99 | 1 | 2080.22 | 2491.28 | 19.8 | 10800 |
| Cardiff_40_08 | yes | 9510 | 0 | 215.16 | 1 | 3834.02 | 3834.02 | 0 | 215.85 |
| Cardiff_40_08 | no | 9671 | 0 | 258.93 | 1 | 3834.02 | 3834.02 | 0 | 259.65 |
| Cardiff_40_09 | yes | 81734 | 12.7 | 1084.47 | 9 | 3569.6 | 3668.21 | 2.8 | 10800 |
| Cardiff_40_09 | no | 84161 | 13.1 | 1156.82 | 11 | 3541.86 | 3668.21 | 3.6 | 10800 |
| Cardiff_40_10 | yes | 55913 | 0 | 220.62 | 1 | 2692.08 | 2692.08 | 0 | 221.98 |
| Cardiff_40_10 | no | 59456 | 0 | 297.74 | 1 | 2692.08 | 2692.08 | 0 | 299.62 |
| Cardiff_40_11 | yes | 48327 | 0 | 5.94 | 1 | unfeasible | unfeasible | 0 | 6.59 |
| Cardiff_40_11 | no | 51524 | 0 | 6.89 | 1 | unfeasible | unfeasible | 0 | 8.0 |
| Cardiff_40_12 | yes | 38128 | 0 | 280.93 | 1 | 2642.87 | 2642.87 | 0 | 282.13 |
| Cardiff_40_12 | no | 38637 | 0 | 351.3 | 1 | 2642.87 | 2642.87 | 0 | 353.38 |
| Cardiff_40_13 | yes | 121649 | 18.4 | 199.97 | 17 | 2742.72 | 2827.83 | 3.1 | 10800 |
| Cardiff_40_13 | no | 123340 | 18.4 | 274.22 | 15 | 2742.19 | 2827.83 | 3.1 | 10800 |
| Cardiff_40_14 | yes | 247124 | 30.2 | 1794.42 | 1 | 2236.37 | 2910.91 | 30.2 | 10800 |
| Cardiff_40_14 | no | 260625 | 30.2 | 3466.11 | 1 | 2236.37 | 2910.91 | 30.2 | 10800 |
| Cardiff_40_15 | yes | 33041 | 19.9 | 281.4 | 37 | 2549.95 | 2707.12 | 6.2 | 10800 |
| Cardiff_40_15 | no | 33301 | 20 | 351.13 | 35 | 2643.72 | 2707.13 | 2.4 | 10800 |
| Cardiff_40_16 | yes | 153756 | 18 | 844.32 | 3 | 2904.8 | 3353.43 | 15.4 | 10800 |
| Cardiff_40_16 | no | 156619 | 19.1 | 1134.11 | 3 | 2881.14 | 3353.43 | 16.4 | 10800 |
| Cardiff_40_17 | yes | 24463 | 1.8 | 678.22 | 13 | 2735.28 | 2749.84 | 0.5 | 10800 |
| Cardiff_40_17 | no | 24917 | 1.9 | 916.07 | 13 | 2722.56 | 2749.84 | 1 | 10800 |
| Cardiff_40_18 | yes | 93744 | 31 | 213.52 | 21 | 2676.43 | 2888.66 | 7.9 | 10800 |
| Cardiff_40_18 | no | 95960 | 31 | 291.94 | 15 | 2509.3 | 2888.66 | 15.1 | 10800 |
| Cardiff_40_19 | yes | 10767 | 0 | 264.74 | 1 | 2784.05 | 2784.05 | 0 | 265.26 |
| Cardiff_40_19 | no | 10810 | 0 | 262.37 | 1 | 2784.05 | 2784.05 | 0 | 263.17 |
| Cardiff_40_20 | yes | 85341 | 27.2 | 219.7 | 21 | 2586.44 | 3083.33 | 19.2 | 10800 |
| Cardiff_40_20 | no | 87042 | 27.2 | 322.25 | 13 | 2594.19 | 3083.33 | 18.9 | 10800 |
| c100_01 | yes | 18845 | 0 | 55.84 | 1 | 5298.76 | 5298.76 | 0 | 56.75 |

Continued on next page

Table 16: Detailed experiments on the impact of the SPR

| Instance | SPR | $\lvert\mathcal{A}^{\max}\rvert$ | Root Gap | Root Time | Nodes | DB | PB | Gap | Time (s) |
|---|---|---|---|---|---|---|---|---|---|
| c100_01 | no | 18845 | 0 | 57.8 | 1 | 5298.76 | 5298.76 | 0 | 58.64 |
| c100_02 | yes | 18845 | 0 | 88.01 | 1 | 5298.36 | 5298.36 | 0 | 88.81 |
| c100_02 | no | 18845 | 0 | 84.7 | 1 | 5298.36 | 5298.36 | 0 | 85.49 |
| c100_03 | yes | 18898 | 0 | 139.25 | 1 | 5296.46 | 5296.46 | 0 | 140.3 |
| c100_03 | no | 18898 | 0 | 150.06 | 1 | 5296.46 | 5296.46 | 0 | 150.89 |
| c100_04 | yes | 20064 | 0.5 | 111.65 | 39 | 5294.66 | 5294.66 | 0 | 1042.88 |
| c100_04 | no | 20064 | 0.5 | 99.7 | 45 | 5294.66 | 5294.66 | 0 | 1629.27 |
| c100_05 | yes | 18849 | 0 | 56.7 | 1 | 5298.76 | 5298.76 | 0 | 57.6 |
| c100_05 | no | 18849 | 0 | 57.66 | 1 | 5298.76 | 5298.76 | 0 | 58.43 |
| c100_06 | yes | 18844 | 0 | 80.37 | 1 | 5298.76 | 5298.76 | 0 | 81.23 |
| c100_06 | no | 18844 | 0 | 78.6 | 1 | 5298.76 | 5298.76 | 0 | 79.32 |
| c100_07 | yes | 18875 | 0 | 65.13 | 1 | 5298.76 | 5298.76 | 0 | 66.0 |
| c100_07 | no | 18875 | 0 | 66.53 | 1 | 5298.76 | 5298.76 | 0 | 67.33 |
| c100_08 | yes | 18845 | 0 | 107.34 | 1 | 5295.36 | 5295.36 | 0 | 108.15 |
| c100_08 | no | 18845 | 0 | 114.34 | 1 | 5295.36 | 5295.36 | 0 | 115.17 |
| c100_09 | yes | 19018 | 0 | 218.49 | 1 | 5295.16 | 5295.16 | 0 | 219.57 |
| c100_09 | no | 19018 | 0 | 226.06 | 1 | 5295.16 | 5295.16 | 0 | 227.08 |

Table 17: Detailed experiments on the impact of the restricted master heuristics

| Instance | RMH | Root Gap | Root Time | Nodes | DB | PB | Gap | Time (s) |
|---|---|---|---|---|---|---|---|---|
| Cardiff_40_01 | yes | 0.3 | 214.13 | 3 | 2931.88 | 2931.88 | 0.00 | 306.84 |
| Cardiff_40_01 | no | 0.5 | 234.17 | 5 | 2931.88 | 2931.88 | 0.00 | 447.26 |
| Cardiff_40_02 | yes | 21.1 | 99.25 | 77 | 2986.6 | 3242.98 | 8.58 | 10800.00 |
| Cardiff_40_02 | no | 20.1 | 92.80 | 37 | 3044.72 | 3217.42 | 5.67 | 10800.00 |
| Cardiff_40_03 | yes | 2.7 | 168.23 | 3 | 2682.5 | 2682.5 | 0.00 | 351.02 |
| Cardiff_40_03 | no | 14.6 | 114.83 | 21 | 2682.5 | 2682.5 | 0.00 | 1287.88 |
| Cardiff_40_04 | yes | inf | 401.23 | 9 | 4292.31 | - | - | 10800.00 |
| Cardiff_40_04 | no | 9.2 | 453.32 | 57 | 4297.88 | 4620.65 | 7.51 | 10800.00 |
| Cardiff_40_05 | yes | 1.7 | 329.68 | 11 | 3770.9 | 3770.9 | 0.00 | 2141.71 |
| Cardiff_40_05 | no | 1.8 | 378.99 | 15 | 3770.9 | 3770.9 | 0.00 | 1415.53 |
| Cardiff_40_06 | yes | 12.5 | 203.15 | 19 | 2909.0 | 2909.0 | 0.00 | 1633.04 |
| Cardiff_40_06 | no | 14.6 | 214.06 | 17 | 2909.0 | 2909.0 | 0.00 | 1067.80 |
| Cardiff_40_07 | yes | 21.9 | 741.64 | 53 | 2297.3 | 2535.67 | 10.38 | 10800.00 |
| Cardiff_40_07 | no | inf | 674.04 | 3 | 2093.42 | - | - | 10800.00 |
| Cardiff_40_08 | yes | 0.6 | 191.29 | 3 | 3834.02 | 3834.02 | 0.00 | 358.60 |
| Cardiff_40_08 | no | 0.5 | 247.14 | 5 | 3834.02 | 3834.02 | 0.00 | 509.05 |
| Cardiff_40_09 | yes | 35.1 | 381.09 | 11 | 3580.1 | 4484.18 | 25.25 | 10800.00 |

Continued on next page

Table 17: Detailed experiments on the impact of the restricted master heuristics

| Instance | RMH | Root Gap | Root Time | Nodes | DB | PB | Gap | Time (s) |
|---|---|---|---|---|---|---|---|---|
| Cardiff_40_09 | no | 17.5 | 338.15 | 73 | 3580.5 | 3707.23 | 3.54 | 10800.00 |
| Cardiff_40_10 | yes | 1.9 | 184.87 | 5 | 2692.08 | 2692.08 | 0.00 | 441.81 |
| Cardiff_40_10 | no | 1.3 | 171.29 | 3 | 2692.08 | 2692.08 | 0.00 | 573.32 |
| Cardiff_40_11 | yes | 0 | 29.34 | 1 | unfeasible | unfeasible | 0.00 | 30.12 |
| Cardiff_40_11 | no | 0 | 32.62 | 1 | unfeasible | unfeasible | 0.00 | 33.43 |
| Cardiff_40_12 | yes | 0.8 | 200.06 | 3 | 2642.87 | 2642.87 | 0.00 | 307.18 |
| Cardiff_40_12 | no | 0.7 | 200.69 | 3 | 2642.87 | 2642.87 | 0.00 | 276.22 |
| Cardiff_40_13 | yes | inf | 211.40 | 81 | 2722.68 | - | - | 10800.00 |
| Cardiff_40_13 | no | 21.9 | 196.03 | 23 | 2723.1 | 2913.13 | 6.98 | 10800.00 |
| Cardiff_40_14 | yes | inf | 1716.28 | 1 | 2236.37 | - | - | 10800.00 |
| Cardiff_40_14 | no | inf | 1914.77 | 13 | 2435.77 | - | - | 10800.00 |
| Cardiff_40_15 | 22.9 | 157.89 | 53 | 2707.12 | 2707.12 | 0.00 | 3757.00 | |
| Cardiff_40_15 | no | 20.3 | 181.95 | 39 | 2606.74 | 2707.12 | 3.85 | 10800.00 |
| Cardiff_40_16 | yes | 19.8 | 863.33 | 61 | 3188.12 | 3372.57 | 5.79 | 10800.00 |
| Cardiff_40_16 | no | inf | 785.41 | 5 | 2927.97 | - | - | 10800.00 |
| Cardiff_40_17 | yes | 2.1 | 291.80 | 57 | 2749.83 | 2749.83 | 0.00 | 4976.00 |
| Cardiff_40_17 | no | 3.2 | 278.48 | 23 | 2733.69 | 2781.87 | 1.76 | 10800.00 |
| Cardiff_40_18 | yes | 33.0 | 254.40 | 63 | 2731.72 | 2932.45 | 7.35 | 10800.00 |
| Cardiff_40_18 | no | inf | 224.52 | 21 | 2707.07 | - | - | 10800.00 |
| Cardiff_40_19 | yes | 1.9 | 198.71 | 3 | 2784.05 | 2784.05 | 0.00 | 368.26 |
| Cardiff_40_19 | no | 1.7 | 227.32 | 17 | 2784.05 | 2784.05 | 0.00 | 989.55 |
| Cardiff_40_20 | yes | 32.1 | 237.58 | 103 | 2617.56 | 3203.98 | 22.40 | 10800.00 |
| Cardiff_40_20 | no | inf | 214.13 | 23 | 2649.34 | - | - | 10800.00 |
| c100_01 | 0.3 | yes | 88.03 | 5 | 5298.76 | 5298.76 | 0.00 | 643.78 |
| c100_01 | 0.3 | no | 70.29 | 33 | 5298.76 | 5298.76 | 0.00 | 3785.00 |
| c100_02 | 0.4 | yes | 86.38 | 7 | 5298.36 | 5298.36 | 0.00 | 751.62 |
| c100_02 | 0.3 | no | 80.16 | 63 | 5298.36 | 5298.36 | 0.00 | 2802.46 |
| c100_03 | 0.4 | yes | 111.98 | 81 | 5296.46 | 5296.46 | 0.00 | 3877.00 |
| c100_03 | 0.4 | no | 104.73 | 43 | 5296.46 | 5296.46 | 0.00 | 4008.00 |
| c100_04 | inf | yes | 151.20 | 121 | 5274.14 | - | - | 10800.00 |
| c100_04 | 1.4 | no | 141.28 | 121 | 5274.95 | 5339.06 | 1.22 | 10800.00 |
| c100_05 | 0.3 | yes | 93.02 | 61 | 5298.76 | 5298.76 | 0.00 | 5342.00 |
| c100_05 | 0.3 | no | 90.38 | 19 | 5298.76 | 5298.76 | 0.00 | 3249.86 |
| c100_06 | 0.4 | yes | 90.42 | 9 | 5298.76 | 5298.76 | 0.00 | 1144.60 |
| c100_06 | 0.3 | no | 89.64 | 23 | 5298.76 | 5298.76 | 0.00 | 1840.12 |
| c100_07 | 0.3 | yes | 87.01 | 37 | 5298.76 | 5298.76 | 0.00 | 3516.10 |
| c100_07 | 0.3 | no | 80.05 | 5 | 5298.76 | 5298.76 | 0.00 | 769.38 |
| c100_08 | 0.4 | yes | 113.60 | 25 | 5295.36 | 5295.36 | 0.00 | 1823.05 |
| c100_08 | 0.4 | no | 93.02 | 15 | 5295.36 | 5295.36 | 0.00 | 1712.00 |
| c100_09 | 0.4 | yes | 125.41 | 49 | 5295.16 | 5295.16 | 0.00 | 5127.00 |

Continued on next page

Table 17: Detailed experiments on the impact of the restricted master heuristics

| Instance | RMH | Root Gap | Root Time | Nodes | DB | PB | Gap | Time (s) |
|---|---|---|---|---|---|---|---|---|
| c100_09 | 0.6 | no | 126.27 | 69 | 5295.16 | 5295.16 | 0.00 | 3874.00 |
| 120_C1_2_01 | yes | 0.4 | 43.36 | 7 | 9090.0 | 9090.0 | 0.00 | 279.21 |
| 120_C1_2_01 | no | 0.4 | 45.31 | 7 | 9090.0 | 9090.0 | 0.00 | 251.78 |
| 120_C1_2_02 | yes | 0.3 | 93.16 | 83 | 8920.5 | 8920.5 | 0.00 | 2895.06 |
| 120_C1_2_02 | no | 0.3 | 87.07 | 29 | 8920.5 | 8920.5 | 0.00 | 1827.92 |
| 120_C1_2_03 | yes | inf | 185.25 | 143 | 8815.06 | - | - | 10800.00 |
| 120_C1_2_03 | no | 0.7 | 158.94 | 247 | 8812.02 | 8868.12 | 0.64 | 10800.00 |
| 120_C1_2_04 | yes | inf | 163.47 | 153 | 8728.17 | - | - | 10800.00 |
| 120_C1_2_04 | no | inf | 155.55 | 229 | 8718.23 | - | - | 10800.00 |
| 120_C1_2_05 | yes | 0.5 | 65.29 | 21 | 8986.2 | 8986.2 | 0.00 | 1552.51 |
| 120_C1_2_05 | no | 0.5 | 58.24 | 19 | 8986.2 | 8986.2 | 0.00 | 1454.68 |
| 120_C1_2_06 | yes | 0.9 | 97.45 | 99 | 8988.4 | 8988.4 | 0.00 | 4892.00 |
| 120_C1_2_06 | no | 0.9 | 91.46 | 17 | 8988.4 | 8988.4 | 0.00 | 1899.70 |
| 120_C1_2_07 | yes | 0.4 | 84.31 | 29 | 8958.6 | 8958.6 | 0.00 | 1845.62 |
| 120_C1_2_07 | no | 0.5 | 67.44 | 29 | 8958.6 | 8958.6 | 0.00 | 2438.20 |
| 120_C1_2_08 | yes | 0.3 | 153.67 | 135 | 8817.9 | 8817.9 | 0.00 | 6501.00 |
| 120_C1_2_08 | no | 0.3 | 166.80 | 9 | 8817.9 | 8817.9 | 0.00 | 642.36 |
| 120_C1_2_09 | yes | inf | 164.03 | 231 | 8771.15 | - | - | 10800.00 |
| 120_C1_2_09 | no | 0.3 | 171.82 | 51 | 8790.62 | 8790.62 | 0.00 | 3394.73 |
| 120_C1_2_10 | yes | 1.0 | 191.08 | 243 | 8729.87 | 8808.82 | 0.90 | 10800.00 |
| 120_C1_2_10 | yes | inf | 193.90 | 131 | 8741.16 | - | - | 10800.00 |

# Appendix B Calculation of drone-reachable neighborhood

Let $Q^d$ be the drones maximal capacity. For $i \in V$, $f_i^d$ is set to 0 if $q_i > Q^d$ or if $i$ is not available for deliveries by drone. Otherwise we set $f_i^d = 1$. We introduce a parameter $\bar{f}_i^d = 0$ expressing whether client $i \in V$ is available to launch drone from or not. The latest can be explained because no parking spot is available nearby for example. We set $\bar{f}_i^d = 0$ in this case and $\bar{f}_i^d = 1$ otherwise. An energy function limits the maximal flight distance, calculated as follows: $P(q) = (W + m + q)^{\frac{3}{2}} \sqrt{\frac{g^3}{2\rho\zeta h}}$ in Watt, introduced by Dorling et al. (2016) as the hovering power consumption of a h-rotor drone, an upper bound on the general power consumption of a h-rotor drone, and based on Leishman (2006). $W$ is the frame weight ($kg$), $m$ the battery weight ($kg$), q the payload ($kg$), $g$ is the force due to gravity ($N$), $\rho$ the fluid density of air in ($kg/m^3$) and $\zeta$ the area of spinning blade ($m^2$). We note $B_c$ the drone's battery capacity in

watt-hours. Let $g = 9.81$ N/kg and $\rho = 1.204$ kg/m$^3$ as in (Dorling et al., 2016).

$$C_i = \{j \in V | \left(\bar{f}_i^d == 1\right) \wedge \left(f_j^d == 1\right) \wedge \left(P(q_i) * c_{i,j}^d + P(0) * c_{j,i}^d \leq B_c\right) \wedge \left(c_{0,i}^v + c_{i,j}^d \leq \bar{d}_j\right) \quad i \in V\}$$

# References


Tobias Achterberg. Constraint Integer Programming. PhD thesis, Technische Universitat Berlin, 2007. URL https://depositonce.tu-berlin.de/items/9f46a10e-2f7b-4dea-8e27-9cae07de5258.

Niels Agatz, Paul Bouman, and Marie Schmidt. Optimization Approaches for the Traveling Salesman Problem with Drone. Transportation Science, 52(4): 965–981, 2018. ISSN 0041-1655, 1526-5447. doi:10.1287/trsc.2017.0791. URL https://pubsonline.informs.org/doi/10.1287/trsc.2017.0791.

Amazon. Amazon is launching ultra-fast drone deliveries in Italy, the UK, and a third location in the U.S. https://www.aboutamazon.com/news/operations/amazon-prime-air-drone-delivery-updates, 2023. URL https://www.aboutamazon.com/news/operations/amazon-prime-air-drone-delivery-updates. [Accessed 27-10-2023].

Roberto Baldacci, Nicos Christofides, and Aristide Mingozzi. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. Mathematical Programming, 115(2):351–385, October 2008. ISSN 1436-4646. doi:10.1007/s10107-007-0178-5. URL https://doi.org/10.1007/s10107-007-0178-5.

Paul Bouman, Niels Agatz, and Marie Schmidt. Dynamic programming approaches for the traveling salesman problem with drone. Networks, 72(4): 528–542, December 2018. ISSN 00283045. doi:10.1002/net.21864. URL https://onlinelibrary.wiley.com/doi/10.1002/net.21864.

Cheng Chen, Emrah Demir, and Yuan Huang. An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and delivery robots. European Journal of Operational Research, 294(3):1164–1180, November 2021. ISSN 03772217. doi:10.1016/j.ejor.2021.02.027. URL https://linkinghub.elsevier.com/retrieve/pii/S037722172100120X.

Sung Hoon Chung, Bhawesh Sah, and Jinkun Lee. Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions. Computers & Operations Research, 123:105004, November 2020. ISSN 0305-0548. doi:10.1016/j.cor.2020.105004. URL https://www.sciencedirect.com/science/article/pii/S0305054820301210.

Luigi Di Puglia Pugliese and Francesca Guerriero. Last-Mile Deliveries by Using Drones and Classical Vehicles. In Antonio Sforza and Claudio Sterle, editors, Optimization and Decision Science: Methodologies and

Applications, Springer Proceedings in Mathematics & Statistics, pages 557–565, Cham, 2017. Springer International Publishing. ISBN 978-3-319-67308-0. doi:10.1007/978-3-319-67308-0_56. URL https://link.springer.com/chapter/10.1007/978-3-319-67308-0_56.

Luigi Di Puglia Pugliese, Francesca Guerriero, and Giusy Macrina. Using drones for parcels delivery process. Procedia Manufacturing, 42:488–497, 2020. URL https://www.sciencedirect.com/science/article/pii/S2351978920305928.

Kevin Dorling, Jordan Heinrichs, Geoffrey G Messier, and Sebastian Magierowski. Vehicle routing problems for drone delivery. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(1):70–85, 2016. URL https://ieeexplore.ieee.org/abstract/document/7513397.

DPDgroup. Innovation for delivering difficult-to-reach areas, 2016. URL https://www.dpd.com/fr/en/sending-parcels/our-innovative-services/drone-delivery/.

Okan Dukkanci, Bahar Y. Kara, and Tolga Bektaş. Minimizing energy and cost in range-limited drone deliveries with speed optimization. Transportation Research Part C: Emerging Technologies, 125:102985, April 2021. ISSN 0968090X. doi:10.1016/j.trc.2021.102985. URL https://linkinghub.elsevier.com/retrieve/pii/S0968090X21000206.

Vipul Garg, Suman Niranjan, Victor Prybutok, Terrance Pohlen, and David Gligor. Drones in last-mile delivery: A systematic review on Efficiency, Accessibility, and Sustainability. Transportation Research Part D: Transport and Environment, 123:103831, October 2023. ISSN 13619209. doi:10.1016/j.trd.2023.103831. URL https://linkinghub.elsevier.com/retrieve/pii/S1361920923002286.

Hermann Gehring and Jörg Homberger. A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows. In Proceedings of EUROGEN99, volume 2, pages 57–64. Citeseer, 1999. URL https://www.semanticscholar.org/paper/A-Parallel-Hybrid-Evolutionary-Metaheuristic-for-Gehring/e41b80d075429403f569980d278f6d4f91da2594.

Ronald Lewis Graham, Eugene Leighton Lawler, Jan Karel Lenstra, and A.H.G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. In Annals of discrete mathematics, volume 5, pages 287–326. Elsevier, 1979. URL https://www.sciencedirect.com/science/article/abs/pii/S016750600870356X.

James R Jackson. Scheduling a production line to minimize maximum tardiness. Research Report 43, Management Science Research Project, 1955.

Mads Jepsen, Bjørn Petersen, Simon Spoorendonk, and David Pisinger. Subset-row inequalities applied to the vehicle-routing problem with time windows. Operations Research, 56(2):497–511, 2008. doi:10.1287/opre.1070.0449. URL https://doi.org/10.1287/opre.1070.0449.

Munjeong Kang and Chungmok Lee. An Exact Algorithm for Heterogeneous Drone-Truck Routing Problem. Transportation Science, 55(5):1088–1112, September 2021. ISSN 0041-1655, 1526-5447. doi:10.1287/trsc.2021.1055. URL http://pubsonline.informs.org/doi/10.1287/trsc.2021.1055.

Aline Karak and Khaled Abdelghany. The hybrid vehicle-drone routing problem for pick-up and delivery services. Transportation Research Part C: Emerging Technologies, 102:427–449, May 2019. ISSN 0968090X. doi:10.1016/j.trc.2019.03.021. URL https://linkinghub.elsevier.com/retrieve/pii/S0968090X18312932.

Patchara Kitjacharoenchai, Mario Ventresca, Mohammad Moshref-Javadi, Seokcheon Lee, Jose M. A. Tanchoco, and Patrick A. Brunese. Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. Computers & Industrial Engineering, 129:14–30, March 2019. ISSN 0360-8352. doi:10.1016/j.cie.2019.01.020. URL https://www.sciencedirect.com/science/article/pii/S0360835219300245.

Patchara Kitjacharoenchai, Byung-Cheol Min, and Seokcheon Lee. Two echelon vehicle routing problem with drones in last mile delivery. International Journal of Production Economics, 225:107598, July 2020. ISSN 0925-5273. doi:10.1016/j.ijpe.2019.107598. URL https://www.sciencedirect.com/science/article/pii/S0925527319304335.

Gilbert Laporte and Yves Nobert. A branch and bound algorithm for the capacitated vehicle routing problem. Operations-Research-Spektrum, 5(2):77–85, June 1983. ISSN 1436-6304. doi:10.1007/BF01720015. URL https://doi.org/10.1007/BF01720015.

Gordon J Leishman. Principles of helicopter aerodynamics with CD extra. Cambridge university press, 2006.

Connie A. Lin, Karishma Shah, Lt Col Cherie Mauntel, and Sachin A. Shah. Drone delivery of medications: Review of the landscape and legal considerations. The Bulletin of the American Society of Hospital Pharmacists, 75(3):153–158, 2018. URL https://academic.oup.com/ajhp/article-abstract/75/3/153/5102071.

Jens Lysgaard, Adam N. Letchford, and Richard W. Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. Mathematical programming, 100:423–445, 2004. URL https://link.springer.com/article/10.1007/s10107-003-0481-8.

Giusy Macrina, Luigi Di Puglia Pugliese, Francesca Guerriero, and Gilbert Laporte. A literature review: Drone-aided routing:. Transportation Research Part C: Emerging Technologies, 120:102762, November 2020. ISSN 0968-090X. doi:10.1016/j.trc.2020.102762. URL https://www.sciencedirect.com/science/article/pii/S0968090X20306744.

Shanshan Meng, Xiuping Guo, Dong Li, and Guoquan Liu. The multi-visit drone routing problem for pickup and delivery services. Transportation Research Part E: Logistics and Transportation Review, 169:102990, January 2023. ISSN 1366-5545. doi:10.1016/j.tre.2022.102990. URL https://www.sciencedirect.com/science/article/pii/S1366554522003672.

Chase C. Murray and Amanda G. Chu. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. Transportation Research Part C: Emerging Technologies, 54:86–109, May 2015. ISSN 0968090X. doi:10.1016/j.trc.2015.03.005. URL https://linkinghub.elsevier.com/retrieve/pii/S0968090X15000844.

Chase C. Murray and Ritwik Raj. The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. Transportation Research Part C: Emerging Technologies, 110:368–398, January 2020. ISSN 0968-090X. doi:10.1016/j.trc.2019.11.003. URL https://www.sciencedirect.com/science/article/pii/S0968090X19302505.

Diego Pecin, Artur Pessoa, Marcus Poggi, and Eduardo Uchoa. Improved branch-cut-and-price for capacitated vehicle routing. Mathematical Programming Computation, 9:61–100, 2017. URL https://link.springer.com/article/10.1007/s12532-016-0108-8.

Artur Pessoa, Ruslan Sadykov, Eduardo Uchoa, and François Vanderbeck. Automation and combination of linear-programming based stabilization techniques in column generation. INFORMS Journal on Computing, 30(2):339–360, 2018. doi:10.1287/ijoc.2017.0784. URL https://doi.org/10.1287/ijoc.2017.0784.

Artur Pessoa, Ruslan Sadykov, Eduardo Uchoa, and François Vanderbeck. A generic exact solver for vehicle routing and related problems. Mathematical Programming, 183(1-2):483–523, September 2020. ISSN 0025-5610, 1436-4646. doi:10.1007/s10107-020-01523-z. URL https://link.springer.com/10.1007/s10107-020-01523-z.

Stefan Poikonen, Xingyin Wang, and Bruce Golden. The vehicle routing problem with drones: Extended models and connections. Networks, 70 (1):34–43, August 2017. ISSN 00283045. doi:10.1002/net.21746. URL https://onlinelibrary.wiley.com/doi/10.1002/net.21746.

Swiss Post. Swiss post drone transport in the healthcare sector, 2023. URL https://www.post.ch/en/about-us/innovation/innovations-in-development/drones.

Mireia Roca-Riu and Monica Menendez. Logistic deliveries with drones: State of the art of practice and research. In 19th Swiss Transport Research Conference (STRC 2019). STRC, 2019. URL https://www.research-collection.ethz.ch/handle/20.500.11850/342823.

David Sacramento, David Pisinger, and Stefan Ropke. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. Transportation Research Part C: Emerging Technologies, 102:289–315, May 2019. ISSN 0968090X. doi:10.1016/j.trc.2019.02.018. URL https://linkinghub.elsevier.com/retrieve/pii/S0968090X18303218.

Ruslan Sadykov and François Vanderbeck. BaPCod - a generic branch-and-price code. Technical report, Inria Bordeaux Sud-Ouest, November 2021. URL https://inria.hal.science/hal-03340548.

Ruslan Sadykov, François Vanderbeck, Artur Pessoa, Issam Tahiri, and Eduardo Uchoa. Primal heuristics for branch and price: The assets of diving methods. INFORMS Journal on Computing, 31(2):251–267, 2019. doi:10.1287/ijoc.2018.0822. URL https://doi.org/10.1287/ijoc.2018.0822.

Ruslan Sadykov, Eduardo Uchoa, and Artur Pessoa. A Bucket Graph–Based Labeling Algorithm with Application to Vehicle Routing. Transportation Science, 55(1):4–28, January 2021. ISSN 0041-1655, 1526-5447. doi:10.1287/trsc.2020.0985. URL http://pubsonline.informs.org/doi/10.1287/trsc.2020.0985.

Daniel Schermer, Mahdi Moeini, and Oliver Wendt. A hybrid VNS/Tabu search algorithm for solving the vehicle routing problem with drones and en route operations. Computers & Operations Research, 109:134–158, September 2019. ISSN 03050548. doi:10.1016/j.cor.2019.04.021. URL https://linkinghub.elsevier.com/retrieve/pii/S0305054819301042.

Natasja Sluijk, Alexandre M. Florio, Joris Kinable, Nico Dellaert, and Tom Van Woensel. Two-echelon vehicle routing problems: A literature review. European Journal of Operational Research, 304(3):865–886, 2023. URL https://www.sciencedirect.com/science/article/pii/S0377221722001278.

Marius M Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations research, 35(2):254–265, 1987. URL https://pubsonline.informs.org/doi/abs/10.1287/opre.35.2.254.

John A. Stankovic, Marco Spuri, Krithi Ramamritham, and Giorgio C. Buttazzo. Fundamentals of EDF Scheduling. In Deadline Scheduling for Real-Time Systems: EDF and Related Algorithms, pages 27–65. Springer US, Boston, MA, 1998. ISBN 978-1-4615-5535-3. doi:10.1007/978-1-4615-5535-3_3. URL https://doi.org/10.1007/978-1-4615-5535-3_3.

Felix Tamke and Udo Buscher. A branch-and-cut algorithm for the vehicle routing problem with drones. Transportation Research Part B: Methodological, 144:174–203, February 2021. ISSN 01912615. doi:10.1016/j.trb.2020.11.011. URL https://linkinghub.elsevier.com/retrieve/pii/S0191261520304410.

Felix Tamke and Udo Buscher. The vehicle routing problem with drones and drone speed selection. Computers & Operations Research, 152:106112, April 2023. ISSN 0305-0548. doi:10.1016/j.cor.2022.106112. URL https://www.sciencedirect.com/science/article/pii/S0305054822003422.

Marlin W. Ulmer and Barrett W. Thomas. Same-day delivery with heterogeneous fleets of drones and vehicles. Networks, 72(4):475–505, December 2018. ISSN 00283045. doi:10.1002/net.21855. URL https://onlinelibrary.wiley.com/doi/10.1002/net.21855.

Dan Wang. The economics of drone delivery, 2015. URL https://www.flexport.com/blog/drone-delivery-economics/.

Xingyin Wang, Stefan Poikonen, and Bruce Golden. The vehicle routing problem with drones: several worst-case results. Optimization Letters, 11(4):679–697, April 2017. ISSN 1862-4472, 1862-4480. doi:10.1007/s11590-016-1035-3. URL http://link.springer.com/10.1007/s11590-016-1035-3.

Zheng Wang and Jiuh-Biing Sheu. Vehicle routing problem with drones. Transportation Research Part B: Methodological, 122:350–364, April 2019. ISSN 01912615. doi:10.1016/j.trb.2019.03.005. URL https://linkinghub.elsevier.com/retrieve/pii/S0191261518307884.

Yu Yang, Chiwei Yan, Yufeng Cao, and Roberto Roberti. Planning robust drone-truck delivery routes under road traffic uncertainty. European Journal of Operational Research, 309(3):1145–1160, September 2023. ISSN 0377-2217. doi:10.1016/j.ejor.2023.02.031. URL https://www.sciencedirect.com/science/article/pii/S0377221723001649.

Yunqiang Yin, Ling Qing, Dujuan Wang, T. C. E. Cheng, and Joshua Ignatius. Exact solution method for vehicle-and-drone cooperative delivery routing of blood products. Computers & Operations Research, 164:106559, April 2024. ISSN 0305-0548. doi:10.1016/j.cor.2024.106559. URL https://www.sciencedirect.com/science/article/pii/S0305054824000315.

Hang Zhou, Hu Qin, Chun Cheng, and Louis-Martin Rousseau. An exact algorithm for the two-echelon vehicle routing problem with drones. Transportation Research Part B: Methodological, 168:124–150, February 2023. ISSN 0191-2615. doi:10.1016/j.trb.2023.01.002. URL https://www.sciencedirect.com/science/article/pii/S0191261523000024.

Xun Zhu. Segmenting the public's risk beliefs about drone delivery: A belief system approach. Telematics and Informatics, 40:27–40, 2019. URL https://www.sciencedirect.com/science/article/pii/S0736585319302692.