



MASTERARBEIT | MASTER'S THESIS

Titel | Title

A Computational Experimental Study of the Vehicle Routing
Problem with Zone-Based Pricing

verfasst von | submitted by
Xueyao Zhang BSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of
Master of Science (MSc)

Wien | Vienna, 2026

Studienkennzahl lt. Studienblatt | Degree
programme code as it appears on the
student record sheet:

UA 066 915

Studienrichtung lt. Studienblatt | Degree
programme as it appears on the student
record sheet:

Masterstudium Betriebswirtschaft

Betreut von | Supervisor:

Ass.-Prof. Dr. Christian Tilk

Zusammenfassung:

Den Kern der vorliegenden Arbeit bildet eine computergestützte Experimentalstudie, die auf einer bahnbrechenden Studie von Afsar et al. (2021) basiert. In der ursprünglichen Studie wird unter anderem untersucht, wie es sich auf die Profitabilität auswirkt, wenn Zustellungstarife anhand von vordefinierten geografischen Zonen festgelegt werden, wobei angenommen wird, dass potenzielle Kunden die Dienstleistung nur dann in Anspruch nehmen, wenn die Zustellungskosten einen bestimmten Grenzwert nicht überschreiten.

Das Ziel dieser Arbeit ist es, die Auswirkung der Zonengranularität auf die Qualität der Lösung, auf die Profitabilität und auf die Servicequote genauer zu untersuchen. In der ursprünglichen Studie wurden Kundenzuteilungen auf drei und fünf Zonen verglichen. Zugleich gibt es wenig Forschung darüber, wie sich Veränderungen der Zonenkonfiguration auf die resultierenden Lösungen auswirken. Um das zu testen, verwendet diese Arbeit die originalen Instanzen der ursprünglichen Studie, modifiziert sie jedoch, indem sie die ursprüngliche Zuteilung der Kunden auf 3 und 5 Zonen auf eine Zuteilung auf 4 und 6 Zonen ändert. Die modifizierten Instanzen werden dabei mithilfe eines von Ass.-Prof. Dr. Christian Tilk bereitgestellten Branch-and-Price-Frameworks gelöst. Die Ergebnisse werden mit den Resultaten der ursprünglichen Drei- und Fünf-Zonen-Konfigurationen verglichen.

Die Ergebnisse zeigen, dass die Zonengestaltung einen erheblichen Einfluss darauf hat, wie sich das Vehicle Routing Problem mit Zonentarifen (VRP-ZP) darstellt und welche Lösungen sich ergeben. Insbesondere wirken sich unterschiedliche Zuordnungen der Kunden zu den Zonen auf die erzielten Gewinne aus, was darauf hindeutet, dass die Profitabilität von der gewählten Zonenstruktur abhängt. Die Ergebnisse bestätigen zudem die Bedeutung der Zahlungsbereitschaft der Kunden, da höhere Schwellenwerte im Allgemeinen zu höheren Gewinnen und einem höheren Servicegrad führen.

Abstract (EN):

This thesis presents a computational experimental study based on the seminal study of Afsar et al. (2021). In their study, the authors determined transportation prices according to predefined geographical zones and the prices that customers are willing to pay, referred to as threshold values: a customer accepts the delivery service only when the proposed delivery price does not exceed the corresponding threshold value. The question was how zone-based pricing affected profitability.

The main objective of this thesis is to examine how zone granularity affects solution quality, profitability, and service rate. In the original study, the authors compared configurations with three and five zones. However, little research has been conducted on how changes in zone configuration influence the resulting solutions. To investigate this issue, the original benchmark instances were obtained from the authors and adapted for this thesis. The original customer zone assignments were modified to create alternative four-zone and six-zone configurations. The modified instances were then solved using an existing branch-and-price framework provided by the Ass.-Prof. Dr. Christian Tilk, and the results were compared with those obtained for the original three-zone and five-zone configurations.

The results show that zone design has a significant influence on the performance of the Vehicle Routing Problem with Zone Pricing (VRP-ZP). In particular, variations in customer-to-zone assignments affect the resulting profit levels, indicating that profitability is sensitive to the selected zone structure. The findings also confirm the importance of customers' willingness to pay, as higher threshold values generally lead to greater profits and higher service rates.

Statement on the Usage of Artificial Intelligence

Artificial intelligence (AI) tools, including ChatGPT, Claude by Anthropic, and Microsoft 365 Copilot, were used during the preparation of this thesis. These tools were primarily employed to support document formatting, consistency checks, and some of the literature search conducted for the literature review.

All AI-generated suggestions were critically evaluated, carefully reviewed, and adapted where necessary. The AI tools were used solely as supporting instruments and did not replace the author's independent work, analysis, interpretation, or the generation of the research results.

Acknowledgement

I would like to express my sincere gratitude to Ass.-Prof. Dr. Christian Tilk for his invaluable guidance and continuous support throughout the process of writing this thesis. I greatly appreciated his assistance in clarifying important questions and the constructive feedback he provided throughout the process.

I would also like to thank my family, friends, and colleagues for their support, love, and understanding throughout the time I was working on this thesis.

List of figures

Figure 1: The Vehicle Routing Problem and methods for addressing it	4
Figure 2: Branch and Bound Algorithms (reproduced from Clausen 1999).....	8
Figure 3: Four zones depending on the distances to axis	21
Figure 4: Six zones depending on the distances to axis	21

List of tables

Table 1 Overview of applied methods in absolute and relative numbers (Braekers et al. 2016).....	4
Table 2 Comparison for each instance within 50 customers for 5 zones between the current solver and the result from Afsar et al.	25

Contents

1	Introduction	1
2	Literature review and general concepts	2
2.1	Vehicle Routing Problem (VRP)	2
2.1.1	General overview of the VRP and its background	2
2.1.2	Adding pricing to the VRP.....	3
2.2	General analytic methods for the VRP.....	4
2.2.1	Classical Heuristic Approaches	4
2.2.2	Metaheuristics	5
2.2.3	Exact Algorithms	5
2.2.4	Branch-and-Price	5
3	Experimental study: methods	11
3.1	Problem definition.....	11
3.2	Mathematical models	12
3.2.1	The non-linearized original mathematical model	12
3.2.2	The linearized mathematical model	13
3.3	Branch-and-Price algorithms.....	14
3.3.1	Branch-and-Price	14
4	Experiment study: implementation and results	20
4.1	Data setup	20
4.2	Numeric Result	22
5	Conclusion	27
6	Appendix.....	28
6.1	Four zone average threshold	28
6.2	Four zone high threshold	30
6.3	Four zone low threshold	33
6.4	Six zone average threshold	36

6.5	Six zone high threshold.....	39
6.6	Six zone low threshold.....	42
6.7	Five zone average threshold.....	45
6.8	Five zone high threshold	48
6.9	Five zone low threshold	51
7	References.....	54

1 Introduction

In a globalized world, in which every second vast numbers of goods are being delivered to myriads of different places, logistic routes play a critical role. Every day, countless routes are being taken – not only within towns, cities, and countries, but also globally across the boundaries of countries and continents. Deciding what routes to take exactly, however, is anything but a trivial matter and challenges companies in various ways. In fact, routing decisions are not merely about minimizing operating costs. Instead, they also affect revenues and profit margins through the service levels that companies offer, and through the pricing policies they pursue.

While being very important in the daily business of maximizing profit and minimizing costs, routing problems have also roused interest in the academic world and inspired much research: many studies have been carried out, and various different theories have been proposed and tested that attempt to provide optimal solutions to vehicle routing problems.

This thesis elaborates on an approach to the vehicle routing problem that takes zone-based pricing into account and that was proposed by Afsar et al. (2021). The idea behind that study was to consider the possibility of assigning different specific delivery prices depending on the area, or ‘zone’ in which specific customers are located. Thereby, a crucial real-world factor was considered that is evidently relevant to the maximization of total profit (calculated as revenue minus cost when applying branch-and-price). How Afsar et al.’s model works in detail will be discussed below (see chapter 3). At this point, it is enough to say that their paper opened a new field of research, in which not many studies have yet been carried out. This means that various aspects of the question of how Zone-Based Pricing affects the vehicle routing problem are still to be investigated.

The specific aspect that this thesis explores is the effect of zone granularity on the quality of solutions to the routing problem, i.e. it investigates whether changes in the way in which an area to be serviced is divided into different pricing zones affects the resulting solution. For that purpose, Afsar et al.’s original allocation of customers to particular zones is modified based on predefined criteria and the model is then re-solved under an algorithmic extension, in which bidirectional labelling is added to the solution approach.

2 Literature review and general concepts

This chapter is divided into three sections. The first two sections review the literature on the Vehicle Routing Problem (henceforth VRP) as well as selected extensions that incorporate pricing problems. The last section introduces a brief overview of analytic methods used for VRP with particular emphasis on branch-and-price, which is the major algorithmic framework and which is also applied in this thesis.

2.1 Vehicle Routing Problem (VRP)

2.1.1 General overview of the VRP and its background

One of the most widely studied problems in operations research is the Vehicle Routing Problem (VRP). It originates from the truck dispatching problem introduced by Dantzig & Ramser (1959). That problem considered the practical task of delivering gasoline from a bulk terminal to service stations. In this original version, the VRP is capacitated (CVRP). This means that each vehicle used has a specific capacity, which cannot be exceeded. Dantzig and Ramser's article aimed at finding feasible routes for a certain set of customers in which the demand for each customer is fulfilled when being visited exactly once. Moreover, the routes to be taken are defined as feasible only if they begin and end at a certain depot. Finally, the capacity of the vehicle which executes a route must at least be equal to the sum of the customer demand to be met by the vehicle. The main purpose in addressing this problem is to minimize travel costs, which are typically considered proportional to the total distance traveled (Dantzig & Ramser 1959).

As pointed out, the VRP is one of the most widely investigated problems in operations research. There are 1012 journal articles published between 1959 and 2008, which address the VRP as the main topic (Eksioglu et al. 2009). According to Eksioglu et al. coverage of the topic has been growing at an annual rate of 6%, and over the years, many extensions or variants of approaches to the VRP Problem have been developed. Since routing problems are one of the most important problems to be faced in the real world of business, most of the variants that have been developed have taken additional real-world factors into account, such as pickup and delivery requests, start of the time windows for service at customers, multiple depots, split deliveries, and various others. (Costa et al. 2019)

Many papers that investigated time windows have been inspired by Marius M. Solomon (1983), whose approach was then further developed or analyzed by different researchers, such as Bräysy & Gendreau (2005a, 2005b) or Gendreau & Tarantilis (2010). Delivery and Pick-up Constraints were investigated by Savelsbergh & Sol (1995), and problems deriving from heterogeneous fleets and multiple depots were addressed by Hadjar. et al. (2006). Most commonly, VRP problems have been addressed in terms of general methods such as heuristics or exact algorithms. These will be discussed in more detail in a later section of the thesis, together with the pricing problem and its role in branch-and-price (see 2.2). Let it suffice, at this point, to emphasize that the economic relevance of logistics decisions has been

understood for decades. Already in Christian 1965, it is reported that, depending on the industry, physical distribution costs account for 10% of sales (machinery) to 30% (food and food product).

The specific real-world issue that this thesis investigates is the fact that companies that offer logistics services pursue profit maximization rather than only cost minimization, and that pricing policies (i.e. the prices charged to customers) do not simply reflect in a linear manner the costs of transportation (i.e. the length of the routes to be serviced). (Shi et al. 2023) The next section discusses this issue in greater detail.

2.1.2 Adding pricing to the VRP

According to Dastaki et al. (2020), there exist different pricing strategies, based on whether shipment costs are included in the price or not. In so-called ‘Mill Pricing’ the customer pays the whole delivery cost in addition to the product cost. In ‘(Uniform) Delivered Pricing’ a certain amount of shipping costs are covered by the selling company if these costs exceed the fixed price they set for customers. ‘Zone Pricing’ is a special type of ‘Discriminatory Pricing’, which allows a company to maximize its profit. ‘Zone Pricing’ means that delivery prices are set differently, depending on the location of receiving customers. Areas to be serviced are divided into zones, and for all locations within a zone the delivery price is the same. (Dastaki et al. 2020)

Clearly, the price that customers must pay for goods or services, influences their demand, and this creates pricing problems for selling companies. Taking into account that delivery costs affect the price that customers need to pay, Liu and Chen (2011) investigated pricing problems in inventory routing in the supply chain. Etebari and Dabiri (2016) extended that investigation by taking time windows into account as well, i.e. in scenarios where the price is changed dynamically according to period. Aras et al. (2011) studied a multi station reverse logistics routing problem involving price-based purchasing decisions, i.e. scenarios in which a product is purchased from a dealer when customers consider it to be most profitable.

However, this thesis focuses on the computational experimental extension of the VRP with zone-based pricing, and extant literature on this specific issue is still scarce. In fact, apart from Afsar et al.’s seminal 2021 study, there is hardly any relevant literature on the topic¹. Afsar et al. will therefore serve as the main foundation and starting point of this thesis.

As pointed out, Afsar et al. (2021) attempted to integrate VRP with zone-based pricing issues. In their approach, VRP with zone-based pricing (VRP-ZP) involves vehicles travelling through different zones to serve customers, starting from a depot. In VRP-ZP, the pricing strategies for transportation costs are heterogeneous, which means that customers are paying different transportation costs, depending on their

¹ It needs to be added, that some papers regarding alternative solution methods were published. Shi et al.(2023) proposed an adaptive large neighborhood search algorithm designed to generate high-quality solution in shorter computational times. Gil-Gala et al. (2023) subsequently proposed a genetic-programing-based hyper-heuristic methods which generate priority functions for integrated pricing and route decision automatically. But even though they use different methods, they retain the core problem formulation introduced by Afsar et al.

location. At the same time, each customer has the possibility to choose whether they accept the offered delivery service or not, depending on a threshold value that can be defined for each customer separately. When the offered delivery price is higher than the threshold value, the service offer is rejected by the customer, when lower, it is accepted. (Afsar et al. 2021)

2.2 General analytic methods for the VRP

The VRP and its variants can be addressed using methods of different methodological types. These include mostly (a) heuristics, (b) metaheuristics, and (c) exact algorithms as shown in figure 1.

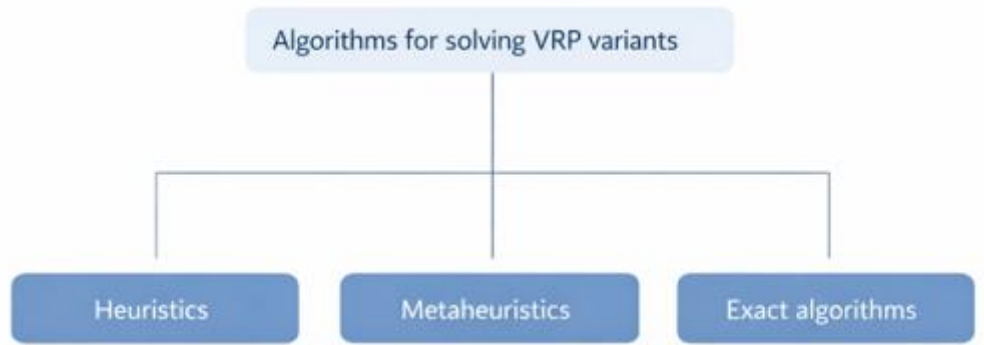


Figure 1: The Vehicle Routing Problem and methods for addressing it

(adopted from Konstantakopoulos et al. 2022: 2036)

Another overview of methods applied to solve logistics problems is in Braekers et al. 2016, summarized in Table 1. As can be seen, the metaheuristic method is the most widely used method used, followed by the exact (-algorithm) method. Since Afsar et al.'s paper employed the latter, and since this thesis is based on their paper, it will also focus on the exact algorithm.

Applied method (1.2)	Number of models	Relative presence							
		Overall (%)	2009 (%)	2010 (%)	2011 (%)	2012 (%)	2013 (%)	2014 (%)	2015 (%)
Metaheuristic (1.2.3)	233	71.25	65%	63	65	77	80	76	65
Exact Method (1.2.1)	56	17.13	17	20	26	10	13	17	19
Classical Heuristic (1.2.2)	32	9.79	11	15	15	17	4	5	10
Real-time solution methods (1.2.5)	11	3.36	6	0	6	4	5	2	0
Simulation (1.2.4)	7	2.14	4	4	6	0	2	0	0

Table 1 Overview of applied methods in absolute and relative numbers (Braekers et al. 2016)

2.2.1 Classical Heuristic Approaches

The Heuristic Approach is a classical method. Based on Shapiro 2001, the heuristic approach is sometimes referred to as a 'rule-of-thumb algorithm'. It involves a sequence of steps taken to solve combinatorial optimization problems. According to Griffis et al. 2012, although heuristics can be used to iteratively refine feasible solutions, the complexity and the size of the problem limits its capability for solution building. Nevertheless, and even though heuristic approaches do not always provide the optimal solution to the problem, the solutions that they produce are frequently near optimal, and the time required to reach a solution is generally shorter than would be required by optimization (Cen et al. 2025). This has been demonstrated in much literature, covering generic algorithms (Katoch et al. 2021; Mirjalili

& Mirjalili 2019), tabu search (Jia et al. 2013; Van Breedam 1995), variable neighborhood search algorithms (Cen et al. 2018; Guo et al. 2022; Rezgui et al. 2019; Sarasola et al. 2016) and simulated annealing (Kuo 2010). – However, over the past two decades, Metaheuristics, an improved type of heuristics, has emerged. (Griffis et al. 2012)

2.2.2 Metaheuristics

Metaheuristics have reduced the quality gap between solutions produced through exact optimization and traditional heuristic approaches, while displaying properties that make them particularly well suited for logistics and supply chain research (Griffis et al. 2012). Unlike classic heuristics, metaheuristics “... perform a much more thorough search of the solution space, allowing inferior and sometimes infeasible moves, as well as recombination of solutions to create new ones” (Cordeau et al. 2002: 516).

2.2.3 Exact Algorithms

Unlike the heuristic approaches, which pursue only near optimal solutions, Exact-Algorithm approaches attempt to find exact optimal solutions, as their name indicates.

The most widely used method among exact algorithms that are used to solve vehicle routing problems is ‘Branch-and-bound’ (henceforth B&B) (Costa et al. 2019). However, the exact algorithm is computationally expensive and therefore not suitable for large scale problems (Braekers et al., 2016). In extant literature, exact algorithms are employed that use B&B (Theurich et al. 2021; Toth & Vigo 2002), column generation (Ceselli et al. 2009; Chabrier 2006), branch-and-price and cut (Desaulniers 2010; Duman et al. 2022), and bender decomposition (Alkaabneh et al. 2020; Fachini & Armentano 2020). When branch-and-bound is combined with column generation, the resulting framework is typically referred to as branch-and-price (Costa et al. 2019). Because Afsar et al. (2021) solve the VRP-ZP by using branch-and-price, the following sections introduce the key ideas of branch-and-bound and column generation and explain how they connect to the pricing subproblem (shortest path problem).

2.2.4 Branch-and-Price

As pointed out, Branch-and-price is a combination of **branch-and-bound** with **column generation**. (Costa et al. 2019)

2.2.4.1 *Branch-and-bound*²

B&B is an algorithm by which the optimal solutions among all possible solutions to a given problem is found. Since evaluating all possible solutions explicitly is computationally infeasible, the algorithm always guides the search toward promising regions in the solution space by using upper and lower bounds. At any stage of branch-and-bound, two elements are maintained in the algorithm: a set of unexplored subproblems and the best feasible solution found so far. Initially, the algorithm starts with

² Unless stated otherwise, the information in this section is derived from (Jens Clausen 1999).

one single subproblem, namely the original problem which is represented by the root node of the search tree. At this point, no best-known solution is available.

Let us assume the problem in which $f(x)$ (a function) of $(x_1 \dots \dots x_n)$ (variables) needs to be minimized over S (region of feasible solutions): $\min_{x \in S} f(x)$.

After the root (the original problem that needs to be solved) is defined, the method which is used in solving B&B algorithms is described as an exploration of the search tree. When node (unexplored subspace) Q is given of a tree, the subproblems are the descendants of Q . They are usually derived through introducing a single additional (new) constraint for each subproblem. The descendants of Q are subproblems that satisfy the same constraints as Q , together with further constraints introduced during branching.

Furthermore, solving B&B involves three main components: (1) calculation of the bound; (2) selection of the node, which needs to be processed; (3) branching. In the following, each of the three components will be briefly described.

(1) Calculation of the bound: The most important component of a B&B algorithm is the bounding function (g). If the quality of the bounding function is low, then this deficiency cannot be compensated, not even through careful selection and branching. Optimally, the bounding function value (to a sub problem) matches the best feasible solution value to the problem. However, it is NP-hard to achieve this. Therefore, it is generally aimed to achieve this optimized bounding function goal as closely as possible with limited computational effort, i.e. within polynomial time. If this goal is achieved, the function is called a strong bounding function. Otherwise, it is called a weak bounding function. There often exists a correlation between time spent finding the best solution and the quality of the bound. A better bound-value will be found if more time is invested in finding the best solution. A stronger bounding function is beneficial, as it results in a smaller search tree.

(2) Selection of the node: The selection for choosing the next node usually involves a trade-off between not **exceeding** a computer's computational capacity and holding the number of explored nodes in search tree low. The best first search strategy (BeFS) is the one in which one of the lower bounds is chosen, when selecting subproblems. For a small search tree, this is illustrated in Figure 2 (a) below. The numbers in each node match the sequence for processing the node.

Sometime, node calculation turns out to be unnecessary only in the end, and this can be referred as critical sub problem. A critical subproblem P exists when the bound function takes a value that is strictly less than the optimal solution of the problem in question. The B&B algorithm has to divide nodes, which are found in the search tree that matches the critical problem, regardless of when the optimized solution is found. Bounding functions can never ignore these nodes, because their lower bounds are still smaller in comparison to the optimal objective value. In a minimization problem, the

lower bound represents the theoretically smallest possible objective value in the search space. Therefore, when an optimal solution exists in a subproblem, its lower bound must equal to or less than the optimal value. Consequently, only nodes that meet this condition need to be explored. Even though an apparently optimal solution has been found, when critical problems exist, they must be processed further in order to prove that no better solution exists.

This argument for BeFS-optimality (in terms of nodes processed) holds under an eager strategy. This strategy chooses the following subproblem according to the bound value of the current subproblem. After a node is selected in an iteration, the first operation is branching, that is, subdividing the node's solution space into two or more subspaces for further exploration in subsequent iterations. For each of these sub-solutions, it is considered whether a single solution exists or not. When there is a single solution in the subgroup, this is compared to the best solution known until now. The whole subgroup will be withdrawn, if no optimal solution exists. Otherwise, the subgroup is kept in a group of live nodes along with its associated bound. BeFS can also be applied to the lazy strategy, which calculates the bound of the chosen node and starts with it. It only branches on the node, when it is necessary. The nodes that are generated are put together with the processed node, which was bounded. This strategy is often employed when the next node is selected as the live node of maximal depth in the search tree. The search terminates once no unexplored regions of the solution space remain. It means that the algorithm stops searching for any other node, once the currently best one is identified as the optimal solution.

Even though this method produces a possible good solution in the end, computational capacity will be exceeded if the subproblems become too large. In such a case, the situation matches a breadth-first search strategy (BFS). In this strategy, all nodes on one level of a search tree are processed before moving to the next level. This is illustrated in Figure 2 (b). The number in each node matches the processing sequence of the Best First Search. However, this is impossible to solve for large problems, because the number of nodes grows exponentially at each level of search tree.

Consequently, depth-first search (DFS) emerges as an alternative strategy. Here, a node which lives in the search tree with the largest level will be chosen for exploration. As shown in Figure 2 (c). The memory needed is bound by the number of levels in the search tree, times the maximum number of children of any node, which is usually manageable.

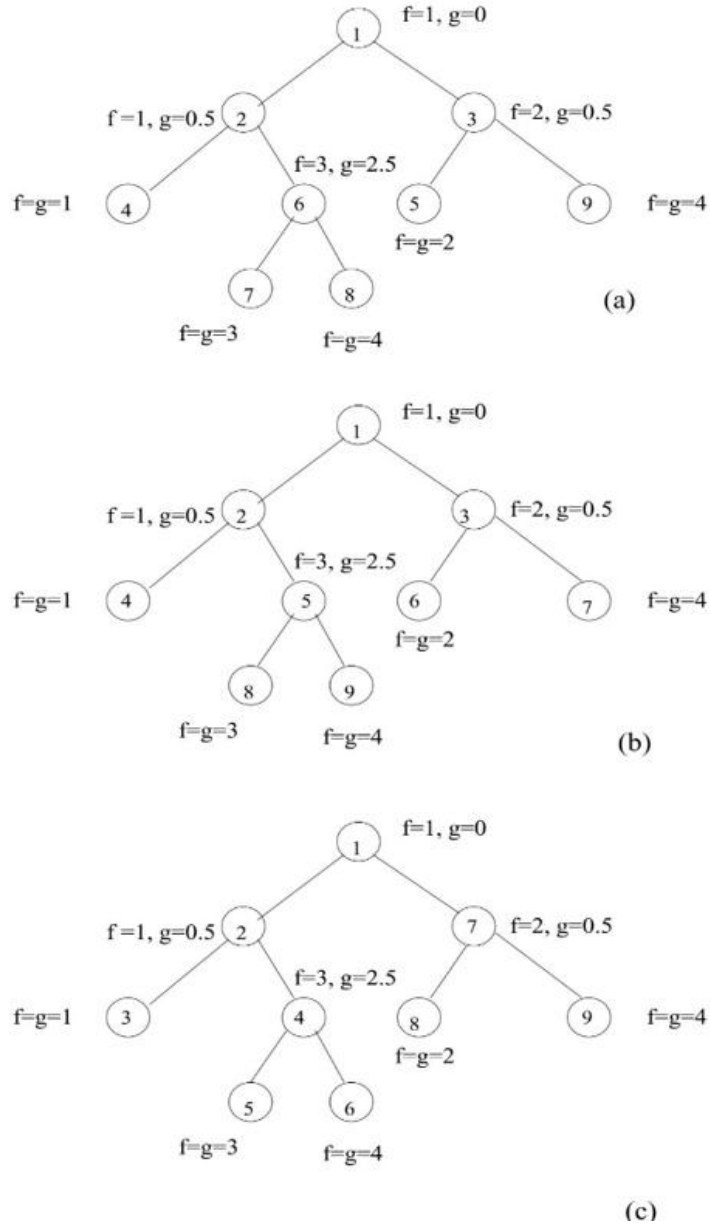


Figure 2: Branch and Bound Algorithms (reproduced from Clausen 1999)

- (3) **The Branching rule:** In the B&B context, all branching rules can be treated as dividing search spaces via additional constraints. This is frequently done by assigning values to variables. When a subspace is divided into two, the process is called dichotomic branching; otherwise it is referred to as polytomic. B&B is guaranteed to arrive at a solution if each generated subproblem is smaller than the original problem and the number of feasible solutions is finite. Typically, the generated subproblems are separated, which prevent the same feasible solution from appearing in multiple subspaces of the search tree.

2.2.4.2 Column generation

The general idea of branch-and-bound has just been sketched. As pointed out, however, branch-and-price is a combination of B&B and column generation. Therefore, in this section, the idea of column generation (henceforth CG) will be shortly introduced as well.

As described in Castaño & Sevaux 2016, CG divides the initial problem into two problems. One is the restricted master problem; the other the pricing subproblem. The restricted master problem (RMP) is the initial optimization problem but is restricted to a reduced solution space defined by a limited subset of columns (variables). The pricing problem facilitates the identification of further columns, which then can be added to the restricted RMP. (Castaño & Sevaux 2016)

According to Castaño & Sevaux (2016), CG adds new variables to the problem gradually – a possibility that had not been considered before. By doing so, the search space is enlarged gradually, which then improves the objective function of RMP. RMP is initialized with a restricted subset of variables that yields a feasible initial solution to the problem. New variables are then introduced sequentially according to reduced cost criteria. At each iteration, these previously unknown columns are generated through the pricing subproblem and incorporated into the RMP to improve the solution progressively (Castaño and Sevaux 2016). Furthermore, the elementary shortest-path problem is often used as a pricing subproblem (as discussed in Irnich & Villeneuve 2006). This is also the pricing subproblem in this thesis, which will be discussed in greater detail in the next section.

Finally, it deserved to be pointed out that column generation (CG) has been found to be a good method to solve the linear programming problem. With CG, the computational effort that is required to solve problems is reduced. (Adelman 2007; Rossi et al.2014; Maher 2015)

2.2.4.3 The Shortest-Path problem as pricing problem

For all living species, finding the shortest path to a goal is often vital, as it saves energy and increases chances of survival. It is the same in the logistic context: the shortest path incurs the lowest cost. Since finding the shortest path is often difficult, however, algorithms for this problem have been studied for a long time (e.g. Bellman 1958; Dial 1969; Dijkstra 1959).

There are two main categories All Pairs Shortest Path (APSP) and Single Source Shortest Path (SSSP). APSP looks for the shortest path from one node to every other node. SSSP searches for shortest path from one single start point to all other vertices. In this section, the SSSP will be discussed in more detail, because it is also used in the algorithm employed in this study. There exist different methods across literature to calculate the shortest path, in this section, only some of the main methods would be generally introduced.

A matrix method for identifying unit-length shortest paths was developed to evaluate the relationship between networks already in 1946. Two points S and T exist in a directed graph, and. T can be reached from S . (See Landahl & Runge 1946; Landahl 1947; Luce & Perry 1949)

After that, in 1955, the shortest path as linear programming was observed by Orden (1955). He conceptualized the problem as an uncapacitated minimum cost flow problem. Based on that, Dantzig (1957) then developed a graphical procedure that is applicable to this problem. It begins with a rooted spanning tree on a set of nodes $(1, \dots, n)$, with node 1 chosen as the root. For each node i , U_i equals length of the path from 1 to i in T . After these distance labels have been defined, the shortest path conditions can be checked: the distance to j via the tree should be no longer than going from 1 to i along the tree, using arc (i, j) . If this holds for all arcs, then the shortest path in the tree from 1 to every other node is already found. However, if an arc from node i to a node j is found, where the current distance to node j is longer than the path obtained by first reaching node i and then travelling along arc (i, j) to j , then the current path is too long. This means that there exists a shorter way to reach the node via i . In this case, the arc in tree, which connects j to its parent with arc (i, j) needs to be replaced. By doing this, a shorter path from the root of j is found. The process is to be constantly repeated, until no violations exist. The algorithm always terminates eventually, because with each correction, the global measures are improved, and there can only exist a finite number of ways in which the tree can change.

Apart from the matrix method for unit-length shortest path, there also exists a shortest path concept that can be called ‘shortest length paths’. According to Schrijver (2012), when a starting node S , is provided, then the shortest-length path (distances) can be calculated in a directed graph $D = (V, A)$, and the real-valued length function is defined as $l: A \rightarrow \mathbb{R}$. To measure the length, two methods were frequently used: the ‘Bell-Ford method’ and ‘Dijkstra’s method’. The “Bell-Ford method” can find the shortest path even when the weighted edge value is negative. The “Dijkstra’s method” is known to be quicker than the “Bell-Ford” method. However, the length function must be positive. (Schrijver 2012)

The labelling method is one of the most widely used method to solve the shortest path problem (Cherkassky et al.1995). It is also applied in the algorithm to be adopted later, and will therefore be discussed at that later point in the thesis (see below 3.3.1.4) rather than here.

Apart from the methods just mentioned, there exist many other possibilities for solving the shortest path problem but discussing them here would exceed the scope of this thesis. For a survey of further methods, see (Schrijver 2012).

3 Experimental study: methods

To examine the sensitivity of the model proposed in Afsar et al. 2021 to zone configuration, I modified the zone-assignment parameters and reclassified customers according to the proposed criteria. For each resulting customer-to-zone allocation, the optimization model was solved again, allowing us to assess how changes in geographical zoning affect pricing decisions, customer selection, vehicle routing, and profitability.

While my thesis is based on a previously developed general framework, it has nevertheless introduced a few changes, so that the model which I used here shows several differences to the framework used by Afsar et al. These differences will be discussed explicitly. Apart from that, I need to mention that I will refrain from referring to Afsar et al. 2021 at each point in the following sections where something has been taken over from their work. Instead, I hope it will suffice to say that most of what follows is indebted to their seminal study.

3.1 Problem definition

Afsar et al. 2021 conceptualized the vehicle routing problem with zone-based pricing in terms of a complete graph $G(V, A)$ that is directed, where V includes the depot node (0) and V^+ as the potential customers to whom delivery services are being considered. Furthermore, arcs (A) are used to connect every pair of nodes. Thus, V^+ is the set of all potential clients, which can be divided into p groups. Each group represents a different geographic zone. These groups are non-overlapping, which means none of the customers can be in more than one zone. Together, though, all zones cover all clients. More importantly, each customer has a demand Q_i , which needs to be satisfied together with a threshold value, which is represented by th_i , which corresponds to the maximum value, which a customer is willing to pay for a delivery.

There are multiple delivery vehicles, all of which are identical. Each of them has the same capacity, which is defined as Q . This capacity is also the maximum capacity, so that it cannot be exceeded. All vehicles not only start but also end every working day at the depot (0). Each time when a vehicle travels from one location to another, regardless of whether from customer or from the depot, it follows an arc $a_{ij} \in A$, on the graph. Following this arc incurs a specific cost C_{ij} .

For each Zone (defined as k), one fixed delivery price (pk) can be defined by the transporter. In order to make sure that all zones are visited, there is at least one customer in each zone that is served. Each client from the set of potential clients (V^+) is visited at most once. The pk (fixed delivery price) is collected from every customer served in zone k .

A function z is defined as customer i belongs to zone k , if and only if i belongs to the set of potential customers:

$$z(i) = k \Leftrightarrow i \in VK, \forall i \in V +$$

For the sake of simplicity, in each zone the customers are sorted ascendingly based on their threshold value. Another function is also defined to count customers in the same zone, when the customer has the threshold (th_j) which is equal to or greater than the th_i .

$$l(i) = |\{j: th_j \geq th_i \text{ and } z(i) = z(j)\}|$$

A home-delivery offer is rejected if the proposed pk (price) from the transporter exceeds the threshold value of the customer.

Profits are calculated as the difference between the delivery cost of the vehicle raised when traveling to the customer, and the sum of the price paid by the customers who are willing to be served.

It has been shown in Afsar et al. 2021, that in the optimal solution, the price pk for a zone k should be equal to the threshold value of one of the customers in that zone, even though the transporter is free to choose any value for pk .

3.2 Mathematical models

Two mathematical models are proposed in the original paper. One is not linearized; the other one is. This thesis employs the linearized model.

3.2.1 The non-linearized original mathematical model

Even though the model used in the thesis is the linearized one, this section briefly describes the non-linearized model proposed by the authors as well.

A binary variable X_{ij} has value 1 when the arc (i, j) is selected by a vehicle. If this is not the case, then the variable X_{ij} has value 0. Another variable f_i is defined with real value, in which the value corresponds to the total capacity of the vehicle, departing from node i . The binary variable (w_i) is set to 1 when the service provided by the company is accepted by customer i . When this scenario holds, the vehicle visits this customer. If not, it has the value of 0. Let P_k denote the price proposed for geographical zone k . The objective function would be the maximized profit:

$$\max \theta = \sum_{i \in V^+} w_i \cdot p_{z(i)} - \sum_{i \in V} \sum_{j \in V} c_{i,j} \cdot x_{i,j}$$

Only when the service price suggested is smaller than the threshold value of the customer, the customer accepts the service offer: $w_i \cdot p_{z(i)} \leq th_i \forall i \in V^+$. Once the service offer is accepted, a vehicle must visit this customer. Furthermore, the service provider must serve at least one customer in a zone, so that it is certain that no zone is abandoned. The number of arcs entering a node must equal the number of arcs

leaving. Other than that, subtours are eliminated because the vehicle load consistency is regulated: $f_i - f_j + q_j + \bar{Q} \cdot X_{i,j} \leq \bar{Q} \quad \forall i \in V, \forall j \in V^+,$ where $\bar{Q}$ is $Q + \max_{v \in V^+} q_v$. The flow will be changed accordingly, when the same vehicle serves customer j right after customer i . The vehicle can only deliver the quantity within its own capacity. For every customer node $i \in V^+$, the load variable f_i is bounded by $w_i \cdot Q$, where Q is the vehicle capacity. This ensures that f_i can only take positive values up to the vehicle capacity when the node is actually served and is bounded by zero otherwise. At the beginning of any route, the total distributed quantity is always 0. The variable p_k is defined as a non-negative real number for each $k \in 1, \dots, p$, representing the price assigned to zone k .

3.2.2 The linearized mathematical model

The original mathematical model proposed in the previous section was not linearized, because two variables are multiplied not only in the objective function, but also in the constraints. It is very difficult to solve, and therefore, this thesis uses the linearized mathematical model, which was also proposed by the authors. It means, a new binary variable y_i equals 1, if and only if $p_z(i) = th_i$. Therefore, the objective function will be reformulated as

$$\max \theta = \sum_{i \in V^+} y_i \cdot th_i \cdot l(i) - \sum_{i \in V} \sum_{j \in V} c_{i,j} \cdot x_{i,j}$$

- y_i is a binary variable that indicates whether the zone price $p_z(i)$ is selected as customer i 's threshold value th_i
- th_i is the price assigned to node i .
- $l(i)$ is defined in the last chapter which represents the number of customers with threshold larger or equal to th_i
- $C_{i,j}$ is the cost of traversing arc i, j , and $X_{i,j}$ is a binary variable equal to 1 if the arc is used, 0 otherwise.

Fixing the price of a zone k to th_i allows the transporter to charge th_i from each of the $l(i)$ clients in that zone. An additional constraint is added, to make sure that each zone has a price:

$$\sum_{i \in V_k} y_i = 1 \quad \forall k \in \{1, \dots, p\}$$

- V_k is the set of nodes (customers) in zone k .
- $y_i \in \{0,1\}$ is a binary variable which was defined in the beginning of this section.
- This constraint ensures that exactly one node in each zone k is selected, i.e., exactly one y_i equals 1 for each zone.

By this constraint, the previous constraint, in which the vehicle cannot abandon a zone can be removed. Finally, the previous constrain $w_i \cdot pz(i) \leq th_i \quad \forall i \in V^+$ can be redefined as for each i in V^+ , w_i is the sum of y_j for all j in $V_z(i)$ such that $th_j \leq th_i$:

$$w_i = \sum_{j \in V_{z(i)}: th_j \leq th_i} y_j \quad \forall i \in V^+$$

- $i \in V^+$: the formula applies for every i in the set V^+ .
- $V_z(i)$: represents the customer set of the zone to which customer i belongs.
- $th_j \leq th_i$: only include terms j in $V_z(i)$ whose threshold (or value) th_j is less than or equal to th_i .
- y_j : this is the quantity being summed over those j which satisfy the condition.

This constraint enforces that if the price of customers i 's zone k where $(k = z(i))$ is set to the threshold th_j of another customer j with $th_j < th_i$, then this customer (i) has to be visited.

In this linearized model, the variable pk is redundant, since the objective function and one of the constraints have been changed. Even though the mathematical model is being linearized, it is still very difficult to solve. Therefore, a branch-and-price algorithm is used by the authors to break the problem down and to solve it.

3.3 Branch-and-Price algorithms

The general idea of how branch-and-price works has already been explained in the literature review. In this section, the notation and difference between the branch-and-price algorithm used in Afsar et al.'s original paper, and the one employed in this thesis will be explained and illustrated by way of example.

3.3.1 Branch-and-Price

3.3.1.1 The Master Problem in the original paper

A linear relaxation of the restricted master problem (LRMP) is defined:

$$\max \theta = \sum_{i \in V^+} y_i th_i l(i) - \sum_{r \in \Omega'} C_r \lambda_r$$

Subject to

$$\sum_{i \in V_k} y_i = 1 \quad \forall k \in \{1, \dots, p\}$$

$$\sum_{r \in \Omega'} \gamma_r^i \lambda_r - \sum_{j \in V_{z(i)}: th_j \leq th_i} y_j = 0 \quad \forall i \in V^+$$

$$\lambda_r \geq 0 \quad \forall r \in \Omega'$$

$$y_i \geq 0 \quad \forall i \in V^+$$

The objective function is the same as the one defined in the last chapter, which is:

$$\max \theta = \sum_{i \in V^+} y_i \cdot th_i \cdot l(i) - \sum_{i \in V} \sum_{j \in V} c_{i,j} \cdot x_{i,j}$$

In this LRMP, there exists a binary variable λ_r . This variable is equal to 1 if route $r \in \Omega$ is selected in the solution. Otherwise, it is 0. Ω represents set of all feasible solutions. Since the number of feasible routes is exponential, a smaller subset Ω' is the starting point and new routes are added iteratively with corresponding variables through column generation. C_r denotes the sum of travel costs of route r . Furthermore, how often the customer i in route r is visited, is represented by γ_r^i .

The first constraint that is imposed specifies that 1 customer can be selected from each group. Another constraint makes sure that the customer is served when the threshold value of the customer is greater than the price set for that zone. Both binary variables y_i and λ_r can take any fractional value between 0 and 1. It is linearly relaxed.

3.3.1.2 The Master Problem in this thesis

The Master Problem formulated in this thesis is mathematically identical to the one proposed by Afsar et al. 2021, which was described in the previous section. However, while the original model from the authors is expressed as profit maximization, the framework which I used was implemented as minimization problem: the revenue contribution of each price columns in the objective has a negative coefficient $th_i \cdot -l(i)$, while route columns keep the standard positive travel cost coefficients. Consequently, the solver minimizes $-\theta = \text{Cost-Revenue}$ rather than maximizing θ directly. As result, all objective values reported in this thesis are negated ($\theta = -UB_{\text{solver}}$) to reflect the true profit in a way that is consistent with the original formulation.

3.3.1.3 The Pricing Problem in the original paper

Two dual variables arise, based on the two constraints just mentioned in the previous section, The dual variables are a_k and Π_i .

A route with reduced costs can be defined as:

$$\overline{C_r} = - \sum_{i \in V^+} \gamma_i^r \cdot \Pi_i - \sum_{(i,j) \in r} c_{i,j}$$

When an existing route r whose reduced costs $\overline{C_r}$ exceed Π_0 , in any given set of dual values Π_i , $i \in V^+$, then this route is added to subset Ω' . Π_0 has value 0 initially. The fleet size constraints will be added during branching phase to the linearized restricted master problem which was introduced in section 3.3.1.1 The value of the dual variables, which is associated with these constraints, will be aggregated

into Π_0 . The column generation stops when no routes with a reduced cost strictly greater than Π_0 can be found. Furthermore, the best solution for the LRMP is found. In order to make sure that no such route exists, the elementary longest path with resource constraints is used by the original author. However, in the framework that this thesis uses, a shortest-path-problem model with resource constraints (SPPRC) is applied.

3.3.1.4 The Pricing Problem in the thesis

As just mentioned, in this thesis, the pricing problem is a shortest path constraint problem solved by labelling algorithms. The idea of shortest path problem has already been introduced in chapter 2.2.4.3. In this section, the general idea of labelling algorithms which are used in the general framework will be introduced.

3.3.1.4.1 Basic labelling algorithms

According to Costa et al. 2019, in a SPPRC, partial routes in G represent labels which start at the original depot (i.e., vertex 0). Labels expand through networks, starting at the original depot, and in the end, reaches the destination depot (Vertex $n + 1$) while visiting some of the customers. In this basic version, a label L is a tuple L , which includes $v, \bar{c}, q, t$. This L represents at the same time path p . In other words, $L = (v, \bar{c}, q, t)$, where vertex v is the final location on the vehicle's route p , and it is one of the vertices in the graph. $\bar{c}$ represents the reduced cost of path p . The load is represented by q , which is cumulated, and t represents the earliest time at which service can start at vertex v , when following route p to reach destination v . A label L also keeps track of its predecessor label, enabling the reconstruction of complete paths after the algorithm finishes. (Costa et al. 2019)

The initial label at vertex 0 is $L_0 = (0, 0, 0, 0)$. If along an arc (i, j) , a label extension is performed, then a new label L' is acquired, by applying following updates:

- Vertex update: the new label L' ends at vertex j . ($v(L') = j$)
- Cost update: The cumulative cost of L' is the cost of the previous label L plus the cost of traveling from the previous vertex i to j . ($\bar{c}(L') = \bar{c}(L) + \bar{c}_{ij}$)
- Load update: The total demand/load of L' is the previous load plus the demand at vertex j . ($q(L') = q(L) + q_j$)
- Time Update: The earliest start time at vertex j is either:
 - the previous label's time plus travel time to j , or
 - the earliest allowed service time a_j at j , whichever is later.

$$(t(L') = \max\{t(L) + t_{ij}, a_j\})$$

The ability to eliminate paths which are not useful determines how efficient the labeling algorithms work. To do this, the concept of a dominance rule is introduced. L_1 and L_2 are two labels in which $v(L_1) = v(L_2)$. L_2 is dominated by L_1 if:

- the cumulated cost of $L1$ is smaller or equal to the ones of $L2$ ($\bar{c}(L1) \leq \bar{c}(L2)$)
- the total load of $L1$ is no greater than that of $L2$ ($q(L1) \leq q(L2)$)
- the earliest service start time of $L1$ is no greater than that of $L2$ ($t(L1) \leq t(L2)$)

This dominance rule holds because of the right-hand side of the relation formulation which was introduced previously (from the cost, load and time update).³

Furthermore, the quality of the bounds, which are generated from Master Problem depends strongly on whether elementarity constraints exist. (Feillet et al. 2004) Beasley & Christofides (1989) made one of the early attempts in this direction, in which they proposed a dynamic programming algorithm that uses customer resource vectors ($\mathcal{E}$) to manage elementarity. This $\mathcal{E}$ stores the customers visited along a path. Here, it will not be discussed in detail. However, this idea is then extended in Feillet et al. 2004, in which they replaced the customer resource vectors with a set of unreachable vertices $\mathcal{U}$. If a vertex j is visited along path $p(L)$, then it is added to $\mathcal{U}(L)$. Alternatively, if it is unreachable, due to the resource limitation (like when the total load, which is a combination from demand already collected in Label L and the demand of customer j , exceeds vehicle capacity Q), then the condition $\mathcal{U}(L1) \subseteq \mathcal{U}(L2)$ is added to the dominance rules, which were just mentioned in the last paragraph. (Feillet et al. 2004)

More importantly, Feillet et al. (2004) have already proved that using routes which are elementary can contribute to much better lower bounds at the starting point of the search tree.

3.3.1.4.2 Ng-Path Relaxation

Currently, the most effective Branch-and-price algorithms in the pricing procedure use the Ng-Path Relaxation. (Costa et al. 2019) In this general framework, the Ng-Path Relaxation is also used to solve the problem. Furthermore, in the original paper, the authors have used Ng-Path Relaxation as well. This relaxation is proposed in Baldacci et al. 2011. Each partial route ending at node i has a forbidden set of nodes ($F(r, i)$), which is not allowed to be visited in the future. A route r can be extended to a node j if and only if $j \notin F(r, i)$. To define the set $F(r, i)$, a variable s is at first defined to represent a position in route r . $r(s)$ is the node at position s following certain sequence. The earlier node $r(u)$ in the route is looked at. If this node appears in all the neighborhoods of nodes, which come after it in the path, then this node $r(u)$ is added to the forbidden set, which means the future path cannot be extended back to $r(u)$. Thereby, the formation of cycles that would revisit nodes considered “close” to all following nodes is prevented. As a formulation, it can be described as:

$$F(r, i) = \{ r(u) : r(u) \in \bigcap_{|r|s=u+1} Nr(s) \} \cup \{ i \}$$

Furthermore, the set $Nr(s)$ is defined as the set of closest “neighbors” to node $r(s)$. Because of this relation, more visits to a node i are possible, if there exists at least one node j , visited between two consecutive visits to i , such that $i \notin N_j$. (Pecin et al. 2017) Since there is the possibility that a node

³ For a proof that this dominance rule holds, see Desaulniers et al. 1998.

can be visited more than once, the resulting route can, therefore, not guarantee elementary. However, these routes are added to Ω' anyway. In the end, there will only be elementary routes, because in the branching process, the routes which are not elementary, are eliminated. Furthermore, any integral solution obtained is feasible.

In order to accelerate the solution process, another labelling algorithm is added in comparison to the original research in this framework. It is bidirectional Labelling.

3.3.1.4.3 Bidirectional Labelling

Bidirectional Labelling was proposed in Righini & Salani 2006. It builds path in both directions, forward and backward (forward from vertex 0 to its successors and backward from vertex $n + 1$ to its predecessors.). Then, complete feasible routes are produced, in which the forward and backward labels are joined. Irnich 2008 has proved that a forward labelling algorithm is applicable to backward algorithm as well. To prevent producing twice as many labels as in the one directional labelling algorithm, the algorithm stops generating new paths, once it is guaranteed that the remaining routes will be generated in the other direction. There are two ways which are generally used to set the stopping point. One is arc bounding; the other one is resource bounding. Here, resource bounding is used. It draws limits based on the capacity of the specific resources (like time, load, distance or cost), that have already been consumed. These specific resources which will be consumed are called critical resources. As long as the accumulated consumption does not exceed the halfway point of the availability along the path, the path is extended. In our case, the critical resource is chosen to be the resource bounding. This means, the Bidirectional Labelling will only extend arcs whose capacity consumption has not been consumed half of the vehicle capacity yet.

As already explained, in this label, the same routes can be obtained multiple times, sometimes by combining more than one pair of backward and forward labels. In order to reduce the combination process as much as possible and to avoid generating the same routes more than once, certain conditions are to be considered. For example, Baldacci et al. (2008) have proposed a condition in which the combination from a pair of forward label and backward label can only happen when the critical-resource consumptions deviate by the smallest amount possible along the resulting path.

Another aspect, which should be considered according to Costa et al. 2019, is the probability of an unbalanced number of labels, which could be generated through the algorithm. This happens when in one direction, a much higher number of labels are generated than in the other direction, regardless of the application of the halfway restriction from the critical source of the extension. There could be different reasons why such a situation could happen. One lies on the asymmetric instance data from the VRPs, the other one lies on the dual information required from the CG iteration (Costa et al. 2019). In order to prevent this problem, Tilk et al. 2017 proposed a halfway point strategy, which can be applied to each iteration dynamically. To explain this more clearly, two more variables are introduced. HF is the resource limit for the forward algorithm and HB is the resource limit for the backward algorithm. The value of

these two variables is changed dynamically during the execution of the algorithm. Furthermore, in order to guarantee optimization, HF is always larger or equal to HB. Other than that, the forward and backward labels are extended one after the other. In order to decide which label should be extended next and in what direction, the algorithm chooses the smallest number of generated labels, processed or unprocessed labels. After the direction is chosen (either forward or backward), the algorithm chooses the next label to extend based on the standard rule (for example least load). It is extended before the HF or HB is updated in the newly generated labels based on the critical resource. The algorithm ends when HF=HB. (Tilk et al. 2017)

3.3.1.4.4 Heuristic Pricing

According to (Costa et al. 2019) it can be very time-consuming to solve an algorithm exactly, even though many speed-up techniques are developed to quicken exact dynamic programming algorithms. Fast and effective heuristics have been developed, since it is not necessary to solve pricing problem exactly, apart from proving the optimality of the current solution in the final CG iteration. This heuristic, known as heuristic pricing, is also used in this paper. Furthermore, it is used to identify variables with negative reduced cost. The purpose of it is to reduce the number of calls to the exact labelling algorithm, which often results in a substantial decrease in total computational time. (Costa et al. 2019) Here, pricing heuristics are constructed by taking into account only parts of the network.

3.3.1.5 Branching strategy

In the branch-and-price tree, the node with the largest upper bound is selected. When the optimal solution for the linear relaxed master problem is not integer, then the following branching strategies come in place:

If the sum of all vehicles for every element r that belongs to the set Ω is not integer: $\alpha = \sum_{r \in \Omega} \lambda_r$. Then two branches are created with extra respective constraints, so that an integer value is required:

The total value of the vehicle should be no greater than the integer part of

$$\sum_{r \in \Omega} \lambda_r \leq \lfloor \alpha \rfloor$$

The other constraint requires the total value of the vehicle to be at least the smallest integer greater than

$$\sum_{r \in \Omega} \lambda_r \geq \lceil \alpha \rceil$$

Furthermore, it is important to note that there could exist several constraints, which limit the size of the vehicle used (fleet size), at any leaf node of the branch-and-price tree. Π_0 adds the dual values obtained from all of them for the pricing problem. Moreover, the constraint just introduced does not affect the pricing problem, it only influences the value of Π_0 . However, in practice, when the vehicle used is bound

by the constraint, the sum of the vehicle should be no greater than the integer part of α . Then “ Π_0 becomes greater and pricing problem takes longer to converge” (Afsar et al. 2021).

The algorithm branches on the most fractional $y_{i'}$, if there are any price-threshold variables that are non-binary. Furthermore, two branches are created according to the following constraints:

$$y_{i'} \geq 1$$

$$y_{i'} \leq 0$$

On the branch that is created when $y_{i'}$ is larger or equal to 1, if a node i' in area $z(i')$ has a higher threshold th_i than another node j in the same area, then node j can be eliminated (and all arcs involving j) from the graph. Since the routing variables λ_r do not appear in the additional constraints, their dual values do not influence the reduced cost calculation. As a result, while the feasible network may be slightly reduced, the overall structure and formulation of the pricing problem remain unchanged. However, since the graph is smaller after the reduction, the pricing problem can be solved more quickly.

When both the price-threshold variable and the vehicle size are integer, but the final solution which comes out is fractional, the arcs that carry fractional flow are identified for further branching or analysis. For each arc (i, j) , the total flow value is calculated. This value is calculated by adding up the values of all variables λ_r corresponding to the routes that pass through arc (i, j) . Afterwards, the arc with the maximum total flow value is selected to be branched on. The arc (i, j) is either forbidden and cannot be part of any route, or it is forced, in which any feasible route must include it. When it is forced, all other arcs outgoing from node i and all arcs incoming to node j are removed from A' , leaving only i, j .

4 Experiment study: implementation and results

4.1 Data setup

The methods which have been described above are run through an existing provided framework and implemented in C++. Furthermore, the CPLEX Optimization Studio Free Edition is used to solve the mathematical model presented in section 3.2. All tests are run on a ThinkPad L15 Gen 4 Laptop.

More importantly, the exact same instances have been acquired as the ones which the Afsar et al. used in their research. They are categorized into three main price-threshold sets across two different zone configurations: a coarser 4-zone allocation and a finer 6-zone allocation.

Since this is a master thesis and running all the instances-files acquired would take too much time and resources, the instances used are the first twenty instances' files with 50 Customers in each price threshold (high, average, low). Therefore, total of 60 instance files are used. Furthermore, customers are first divided into 4 Zones based on the given coordination on X and Y axis. For example, Zone 1 is for customers located on the lower halfway on both X and Y axis. All four zones are shown in Figure 3.

In order to analyze the difference, another zone allocation is also designed: instead of 4 zones, customers are divided into 6 zones. They are shown in Figure 4.

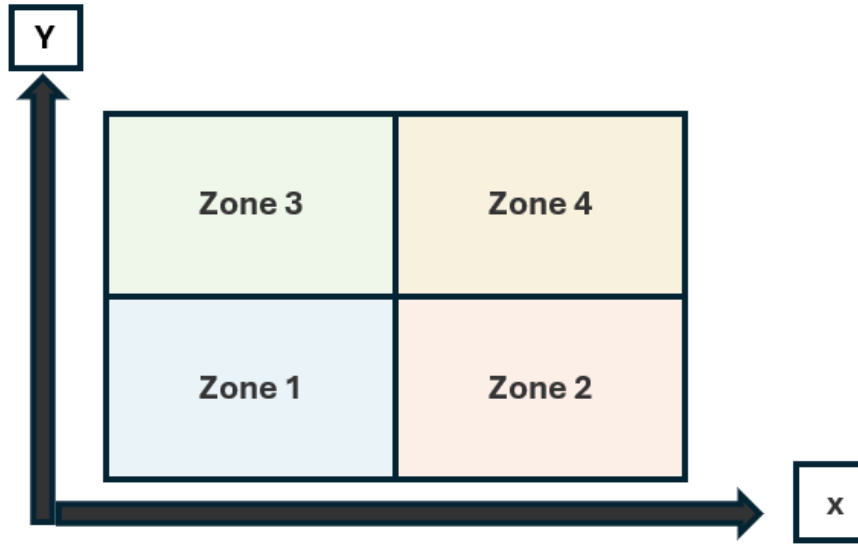


Figure 3: Four zones depending on the distances to axis

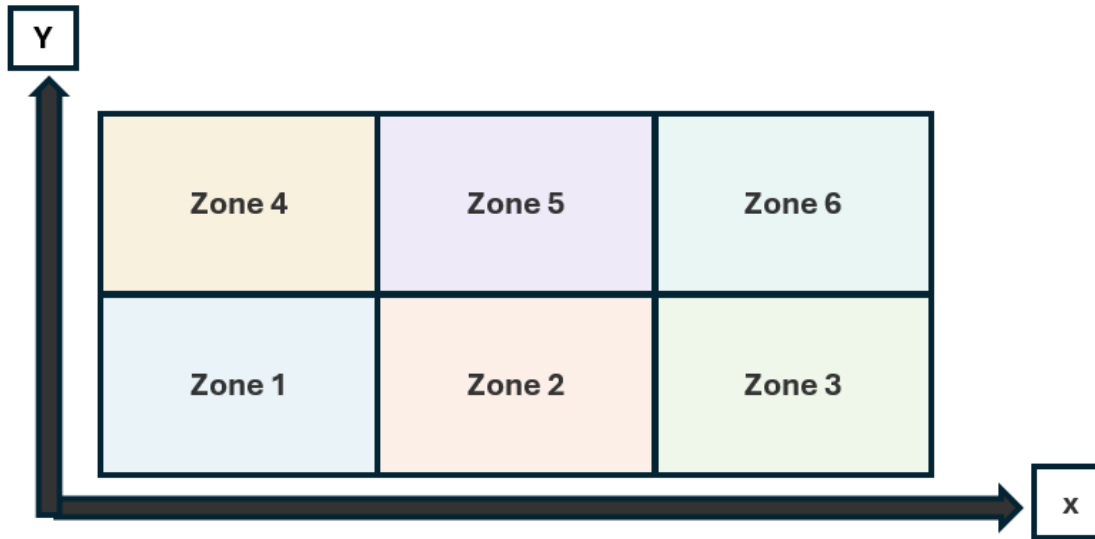


Figure 4: Six zones depending on the distances to axis

Since the instances are adopted from the original authors, the price thresholds which were just mentioned in the last paragraph are the same as the authors (Afsar et al. 2021):

- Low threshold means that the delivery fee, which the customers accept are not very high ($\beta_i \in [0.8, 1.0]$, $\forall i \in V^+$)
- Medium threshold means the price, which the customers accept to pay are a bit higher than the cost of the shortest path ($\beta_i \in [1.0, 1.4]$, $\forall i \in V^+$)

- High threshold means that a significant willingness to pay a higher price than the cost for the service delivery ($\beta_i \in [1.6, 2.0]$, $\forall i \in V^+$)

4.2 Numeric Result

Solver performance is evaluated by comparing the result of the branch-and-price algorithm with that of Afsar et al. (2021) on the same twenty instances selected and scenario settings. Additional experiments as described in the above section are reported separately as scenario analysis. Furthermore, all objective values (UB, LB) presented in this section correspond to the true profit $\Theta = -UB_{solver}$ as defined in section 3.3.1.2 and are reported as positive values.

4.2.1.1 Solver Performance

Before going into the details of different scenarios, the efficiency of the program needs to be checked first. To test it, the exact same first twenty instance files in each threshold which were modified later are first ran for 50 customers with 5 Zones without any changes. These are then compared to the results from the original instances as derived from Afsar et al. 2021. These are first shown as a single instance comparison in table 2, and then the averages are calculated in table 3.

Instances	UB (Afsar et.al)	UB	Time (s) (Afsar et.al)	Time	GAP (%) (Afsar et.al)	GAP (%)
instances/Z5/average/p_50_1_vrpn_c10.txt	708.23	713.773	2623.47	96.8447	2.52%	1.73%
instances/Z5/average/p_50_1_vrpn_c11.txt	1841.48	1 852.560	35346.20	21 600.3	1.29%	0.01%
instances/Z5/average/p_50_1_vrpn_c12.txt	777.85	778.201	1786.09	3 518.41	1.72%	1.66%
instances/Z5/average/p_50_1_vrpn_c13.txt		2 013.120		23 169.2		4.07%
instances/Z5/average/p_50_1_vrpn_c14.txt	747.24	747.241	1291.74	15.2133	1.28%	1.28%
instances/Z5/average/p_50_1_vrpn_c2.txt	629.09	629.866	1507.70	174.873	1.52%	1.40%
instances/Z5/average/p_50_1_vrpn_c3.txt	686.93	686.930	2369.73	5 433.87	2.06%	2.03%
instances/Z5/average/p_50_1_vrpn_c4.txt	699.76	700.166	3446.88	10 378.2	3.46%	3.40%
instances/Z5/average/p_50_1_vrpn_c5.txt	629.04	629.038	1142.46	77.0838	1.46%	1.46%
instances/Z5/average/p_50_1_vrpn_c7.txt	543.04	550.302	1377.34	895.851	3.66%	2.30%

instances/Z5/average/p_50_1_vrpnc8.txt	691.13	691.129	8872.39	2 115.91	1.29%	1.29%
instances/Z5/average/p_50_1_vrpnc9.txt	655.93	655.929	5260.58	2 709.78	1.79%	1.77%
instances/Z5/average/p_50_2_vrpnc10.txt	693.07	694.826	5187.59	3 600.16	1.67%	1.34%
instances/Z5/average/p_50_2_vrpnc2.txt	647.97	652.609	1521.23	806.754	2.95%	2.22%
instances/Z5/average/p_50_2_vrpnc3.txt	726.74	726.737	3673.79	124.788	0.60%	0.60%
instances/Z5/average/p_50_2_vrpnc4.txt	593.07	598.634	2876.03	1 936.5	2.40%	1.45%
instances/Z5/average/p_50_2_vrpnc5.txt	602.64	607.772	2267.04	770.108	2.94%	2.07%
instances/Z5/average/p_50_2_vrpnc7.txt	541.71	552.690	1492.71	2 525.64	4.49%	2.41%
instances/Z5/average/p_50_2_vrpnc8.txt	703.38	703.382	2891.77	308.702	1.57%	1.57%
instances/Z5/average/p_50_2_vrpnc9.txt	778.33	777.882	4370.18	8 421.34	2.21%	2.23%
instances/Z5/High/p_50_1_vrpnc10.txt	1293.17	1 308.360	2820.73	26.3887	1.17%	0
instances/Z5/High/p_50_1_vrpnc11.txt	3294.62	3 300.030	17669.40	396.449	0.16%	0.00%
instances/Z5/High/p_50_1_vrpnc12.txt	1542.05	1 549.380	137.95	22.156	0.00%	0.00%
instances/Z5/High/p_50_1_vrpnc13.txt		3 593.600		21600.8		0.00%
instances/Z5/High/p_50_1_vrpnc14.txt	1539.98	1 551.640	1654.68	14.5517	0.77%	0.00%
instances/Z5/High/p_50_1_vrpnc2.txt	1300.92	1 321.590	1295.73	76.7628	1.59%	0.00%
instances/Z5/High/p_50_1_vrpnc3.txt	1270.64	1 285.410	3311.59	525.496	1.16%	0.00%
instances/Z5/High/p_50_1_vrpnc4.txt	1382.19	1 418.650	3400.37	6087.17	2.64%	0.00%

instances/Z5/High /p_50_1_vrpnc5.txt	1276.21	1 283.540	2158.46	23.2244	0.57%	0.00%
instances/Z5/High /p_50_1_vrpnc7.txt	1146.86	1 165.660	1154.16	2476.83	1.64%	0.00%
instances/Z5/High /p_50_1_vrpnc8.txt	1366.51	1 379.370	5709.34	2232.89	0.94%	0.00%
instances/Z5/High /p_50_1_vrpnc9.txt	1324.09	1 339.210	2970.37	453.511	1.14%	0.00%
instances/Z5/High /p_50_2_vrpnc10.txt	1289.77	1 302.680	8110.22	3600.13	1.02%	0.00%
instances/Z5/High /p_50_2_vrpnc2.txt	1236.23	1 269.350	1412.56	21600.4	2.68%	0.00%
instances/Z5/High /p_50_2_vrpnc3.txt	1414.88	1 434.910	13508.80	7418.75	1.42%	0.00%
instances/Z5/High /p_50_2_vrpnc4.txt	1227.08	1 240.150	6532.18	2789.85	1.07%	0.00%
instances/Z5/High /p_50_2_vrpnc5.txt	1284.48	1 298.000	4622.29	294.664	1.06%	0.00%
instances/Z5/High /p_50_2_vrpnc7.txt	1194.98	1 218.120	1014.64	2689.74	1.94%	0.00%
instances/Z5/High /p_50_2_vrpnc8.txt	1308.61	1 320.030	1641.45	364.237	0.87%	0.00%
instances/Z5/High /p_50_2_vrpnc9.txt	1526.36	1 551.020	4645.72	1276.34	1.63%	0.00%
instances/Z5/Low /p_50_1_vrpnc10.txt	379.13	387.434	2562.41	2 339.17	6.88%	4.58%
instances/Z5/Low /p_50_1_vrpnc11.txt	1372.96	1 389.210	29511.10	566.958	1.74%	0.54%
instances/Z5/Low /p_50_1_vrpnc12.txt	514.12	520.047	4336.87	285.178	2.79%	1.62%
instances/Z5/Low /p_50_1_vrpnc13.txt		1 421.990		21 600.9		4.59%
instances/Z5/Low /p_50_1_vrpnc14.txt	490.72	491.613	613.37	14.6495	1.05%	0.63%
instances/Z5/Low /p_50_1_vrpnc2.txt	334.36	339.530	1747.76	158.35	4.34%	2.75%

instances/Z5/Low /p_50_1_vrpnc3.t xt	380.65	383.023	3114.07	931.491	2.99%	2.34%
instances/Z5/Low /p_50_1_vrpnc4.t xt	423.72	435.836	5066.10	317.684	6.72%	3.75%
instances/Z5/Low /p_50_1_vrpnc5.t xt	420.03	420.034	2819.32	72.4545	2.54%	2.54%
instances/Z5/Low /p_50_1_vrpnc7.t xt	279.41	279.161	1940.11	9.8603	1.98%	2.07%
instances/Z5/Low /p_50_1_vrpnc8.t xt	458.84	457.837	6142.59	907.331	2.19%	2.39%
instances/Z5/Low /p_50_1_vrpnc9.t xt	391.80	391.798	2537.22	66.1017	2.91%	2.91%
instances/Z5/Low /p_50_2_vrpnc10. txt	442.04	445.681	4986.41	3 886.06	4.02%	3.09%
instances/Z5/Low /p_50_2_vrpnc2.t xt	361.69	368.004	1949.33	1 091.01	6.81%	4.98%
instances/Z5/Low /p_50_2_vrpnc3.t xt	457.76	468.062	6089.08	118.968	3.22%	0.92%
instances/Z5/Low /p_50_2_vrpnc4.t xt	367.60	370.900	2506.23	4 690.75	4.70%	3.77%
instances/Z5/Low /p_50_2_vrpnc5.t xt	377.72	379.183	2338.12	232.955	3.45%	3.05%
instances/Z5/Low /p_50_2_vrpnc7.t xt	307.30	308.153	1684.87	30.8278	2.01%	1.72%
instances/Z5/Low /p_50_2_vrpnc8.t xt	408.67	408.665	998.57	14.9774	1.62%	1.62%
instances/Z5/Low /p_50_2_vrpnc9.t xt	535.61	535.608	2392.72	3 236.14	2.95%	2.95%

Table 2 Comparison for each instance within 50 customers for 5 zones between the current solver and the result from Afsar et al.

Threshold	Average UB (Afsar et.al)	Average UB	Average Time (s) (Afsar et.al)	Average Time (s)	Average GAP (Afsar et.al)	Average GAP
Low	408.67	414.3495	2537.22	138.659	2.95%	2.64%
Median	691.13	692.9775	2623.47	174.873	1.79%	1.72%
High	1300.92	1320.81	2970.37	900.918	1.14%	0%

Table 3 Comparison between the average results of instances within 50 customers for 5 zones

From both tables, we can see that the model employed in the present thesis is efficient because it needs much shorter computational time to calculate solutions and the solution quality appears to be better as well. Note that the gap is calculated using the same calculation from the paper, which is upper bound on the root. The values appear to be smaller for most instances across the different thresholds. Nevertheless, none of the UB acquired for each threshold is far apart in comparison with the ones from the authors. On the whole, we can therefore conclude that compared to Afsar et al. 2021, the solver used in the present thesis achieves a better result with smaller GAP in less time.

4.2.1.2 Scenario Analysis

This section analyzes the influence of alternative zone assignment rules and the impact on the outcome.

Threshold	Zones	Average Profit	Average Vehicle Number	Average Service Rate
Low	4	269.6165	3	66
Low (Afsar et al.)	5	408.67	4	88
Low	6	278.6235	3	66
Average	4	492.951	3	70
Average (Afsar et al.)	5	691.13	4	89
Average	6	502.25	3	67
High	4	952.606	3	71

High (Afsar et al.)	5	1300.92	4	92
High	6	933.816	3	73

Table 4 Average results from the distance zone-based pricing across different thresholds

From this table we can see that the original results of 5 zones from the author performed in general better across the thresholds with higher service rate and higher profit. When four zones are used, customers are automatically grouped into larger groups, limiting the ability to differentiate services prices across the customer willingness to pay value. On the other hand, even though increasing the number of zones to six zones introduces greater pricing flexibility, it also leads to excessive market segmentation.

Furthermore, it seems that the threshold value influences the profit and service rate as well. The larger the threshold value, the higher the profits and in the meantime, the higher the service rate as well. This suggests that pricing acceptance of the customer will be a dominant factor for a company to consider.

5 Conclusion

In conclusion, the results confirm that zone design has a relevant impact on the performance of VRP-ZP. In particular, the experimental variations of customer zone design assignments demonstrate that profit outcomes are sensitive to the chosen zone structure. At the same time, the reasons why zone structure affects profits in the way in which it seems to do appear to be anything but obvious and certainly deserve to be investigated in future research. The same applies to its interaction with the maximal price that customers are willing to accept, which – as this study has confirmed – plays an important role as well: generally, higher threshold value tends to produce stronger profit and larger service rate.

6 Appendix

6.1 Four zone average threshold

Table A: Computational Results

instance name	UB	LB tree	LB root	LB root+cuts	time limit [s]	num vehicles
instances/Z4/average/p_50_1_vrpnc10.txt	-496.229	-496.229	-503.324	-503.324	3 600	3
instances/Z4/average/p_50_1_vrpnc11.txt	-1 775,23	-1 798,55	-1 826,68	-1 826,68	21 600	2
instances/Z4/average/p_50_1_vrpnc12.txt	-535.149	-535.511	-545.398	-545.398	3 600	4
instances/Z4/average/p_50_1_vrpnc13.txt	-2 038,19	-2 054,36	-2 065,43	-2 065,43	21 600	2
instances/Z4/average/p_50_1_vrpnc14.txt	-556.999	-556.999	-557.108	-557.108	3 600	3
instances/Z4/average/p_50_1_vrpnc2.txt	-425.796	-425.981	-437.84	-437.84	3 600	4
instances/Z4/average/p_50_1_vrpnc3.txt	-492.951	-492.951	-504.691	-504.691	3 600	3
instances/Z4/average/p_50_1_vrpnc4.txt	-91 727,5	-91 741,6	-91 750	-91 750	3 600	1
instances/Z4/average/p_50_1_vrpnc5.txt	-536.445	-537.407	-548.85	-548.85	3 600	3
instances/Z4/average/p_50_1_vrpnc7.txt	-358.376	-358.983	-373.744	-373.744	21 600	4
instances/Z4/average/p_50_1_vrpnc8.txt	-500.676	-500.965	-514.146	-514.146	21 600	3
instances/Z4/average/p_50_1_vrpnc9.txt	-475.62	-475.62	-480.78	-480.78	3 600	3
instances/Z4/average/p_50_2_vrpnc10.txt	-560.571	-560.761	-568.615	-568.615	3 600	3
instances/Z4/average/p_50_2_vrpnc2.txt	-422.045	-422.558	-437.601	-437.601	3 600	3
instances/Z4/average/p_50_2_vrpnc3.txt	-503.558	-512.834	-523.67	-523.67	3 600	2
instances/Z4/average/p_50_2_vrpnc4.txt	-470.704	-471.009	-484.003	-484.003	3 600	3
instances/Z4/average/p_50_2_vrpnc5.txt	-480.379	-480.379	-481.643	-481.643	3 600	3
instances/Z4/average/p_50_2_vrpnc7.txt	-376.056	-376.74	-381.841	-381.841	3 600	4
instances/Z4/average/p_50_2_vrpnc8.txt	-456.699	-456.986	-465.663	-465.663	3 600	3
instances/Z4/average/p_50_2_vrpnc9.txt	-605.831	-605.914	-614.192	-614.192	3 600	3

Table B: Generated Routes

instance name	routes
instances/Z4/average/p_50_1_vrpnc10.txt	1: (0 10 20 8 18 7 28 0) ,2: (0 12 19 2 49 40 32 46 16 44 37 0) ,3: (0 48 25 39 5 29 38 50 27 42 6 3 9 0) ,
instances/Z4/average/p_50_1_vrpnc11.txt	1: (0 32 20 45 35 4 25 22 17 31 11 36 42 30 0) ,2: (0 29 26 23 37 12 50 28 24 40 6 48 39 21 38 8 19 0) ,
instances/Z4/average/p_50_1_vrpnc12.txt	1: (0 2 6 8 29 40 13 28 12 26 0) ,2: (0 3 5 41 27 7 25 48 16 31 19 0) ,3: (0 33 36 20 9 39 11 34 45 0) ,4: (0 49 17 50 46 43 47 37 22 30 35 10 1 42 0) ,
instances/Z4/average/p_50_1_vrpnc13.txt	1: (0 4 15 2 9 48 18 10 42 47 43 27 41 34 8 0) ,2: (0 36 16 35 44 20 21 33 32 11 29 30 12 49 39 50 7 37 31 6 26 0) ,
instances/Z4/average/p_50_1_vrpnc14.txt	1: (0 11 21 22 38 37 5 33 31 24 4 7 26 14 28 0) ,2: (0 27 8 12 30 13 19 47 17 2 23 0) ,3: (0 39 44 6 50 41 1 43 3 34 20 29 46 49 0) ,
instances/Z4/average/p_50_1_vrpnc2.txt	1: (0 17 18 48 45 36 16 42 30 2 0) ,2: (0 24 50 15 12 35 1 49 0) ,3: (0 25 4 40 7 34 31 0) ,4: (0 39 20 10 21 14 8 38 47 5 0) ,

instances/Z4/average/p_50_1_vrpnc3.txt	1: (0 10 14 34 8 25 37 6 5 1 18 15 42 50 0) ,2: (0 17 35 32 16 3 22 2 4 39 12 31 7 30 11 0) ,3: (0 27 41 38 45 28 48 23 9 40 20 0) ,
instances/Z4/average/p_50_1_vrpnc4.txt	1: (0 41 46 36 29 18 42 24 26 28 33 47 48 27 35 0) ,
instances/Z4/average/p_50_1_vrpnc5.txt	1: (0 2 37 12 30 36 27 4 32 24 3 7 38 45 0) ,2: (0 10 6 20 11 40 34 23 17 21 47 25 0) ,3: (0 31 28 1 14 16 46 41 18 35 44 39 50 22 29 42 0) ,
instances/Z4/average/p_50_1_vrpnc7.txt	1: (0 2 26 3 50 32 0) ,2: (0 20 40 11 47 27 34 24 48 19 0) ,3: (0 31 14 33 29 46 13 35 39 4 0) ,4: (0 43 21 38 10 16 18 22 0) ,
instances/Z4/average/p_50_1_vrpnc8.txt	1: (0 25 30 24 47 45 48 50 33 7 26 13 0) ,2: (0 8 28 43 38 2 34 31 37 36 40 27 14 29 10 0) ,3: (0 39 49 4 9 12 6 11 42 5 23 0) ,
instances/Z4/average/p_50_1_vrpnc9.txt	1: (0 8 42 7 23 39 6 11 18 32 1 0) ,2: (0 14 5 24 9 45 4 22 10 15 17 35 12 50 29 0) ,3: (0 41 36 25 37 3 2 40 47 46 33 26 30 13 0) ,
instances/Z4/average/p_50_2_vrpnc10.txt	1: (0 4 18 42 35 14 47 21 31 30 45 49 0) ,2: (0 20 17 46 9 34 44 25 33 12 10 23 3 19 0) ,3: (0 29 2 24 40 7 36 43 5 27 48 28 0) ,
instances/Z4/average/p_50_2_vrpnc2.txt	1: (0 19 18 43 39 15 6 42 40 10 38 0) ,2: (0 31 2 16 26 41 13 0) ,3: (0 34 5 33 48 23 49 24 37 9 0) ,
instances/Z4/average/p_50_2_vrpnc3.txt	1: (0 7 19 15 6 28 2 42 32 44 5 43 36 10 37 0) ,2: (0 35 9 25 50 14 49 48 29 12 24 8 26 11 4 21 39 3 22 45 0) ,
instances/Z4/average/p_50_2_vrpnc4.txt	1: (0 15 38 2 25 50 29 1 42 28 26 20 8 19 24 37 0) ,2: (0 32 36 27 6 11 17 44 34 30 5 14 0) ,3: (0 43 9 3 47 16 49 40 48 22 4 33 39 0) ,
instances/Z4/average/p_50_2_vrpnc5.txt	1: (0 7 44 20 5 31 3 29 34 13 28 1 33 0) ,2: (0 8 10 9 50 39 37 12 6 23 40 19 43 0) ,3: (0 16 32 25 27 11 49 24 48 15 14 0) ,
instances/Z4/average/p_50_2_vrpnc7.txt	1: (0 17 1 4 13 6 44 14 0) ,2: (0 19 33 2 16 11 36 27 41 50 10 0) ,3: (0 26 49 45 42 48 37 0) ,4: (0 28 5 35 38 7 0) ,
instances/Z4/average/p_50_2_vrpnc8.txt	1: (0 12 7 27 24 25 45 49 44 28 1 0) ,2: (0 23 21 33 35 14 2 26 5 9 36 42 3 40 0) ,3: (0 32 11 46 47 10 22 37 13 15 17 50 43 29 0) ,
instances/Z4/average/p_50_2_vrpnc9.txt	1: (0 25 22 11 14 1 35 33 10 8 42 31 23 9 19 0) ,2: (0 26 4 28 32 29 34 49 20 45 48 39 15 0) ,3: (0 43 24 30 44 50 40 36 18 16 3 2 17 41 0) ,

6.2 Four zone high threshold

Table A: Computational Results

instance name	UB	LB tree	LB root	LB root+cuts	time limit[s]	num vehicles
instances/Z4/High/p_50_1_vrpnc10.txt	-846.473	-846.473	-854.214	-854.214	3 600	3
instances/Z4/High/p_50_1_vrpnc11.txt	-2 951.38	-2 952.46	-3 006.74	-3 006.74	21 600	3
instances/Z4/High/p_50_1_vrpnc12.txt	-1 188.3	-1 188.3	-1 188.99	-1 188.99	3 600	4
instances/Z4/High/p_50_1_vrpnc13.txt	-2 145.13	-3 294.49	-3 295.7	-3 295.7	3 600	1
instances/Z4/High/p_50_1_vrpnc14.txt	-1 128.82	-1 128.82	-1 131.46	-1 131.46	3 600	3
instances/Z4/High/p_50_1_vrpnc2.txt	-866.748	-866.95	-885.653	-885.653	3 600	5
instances/Z4/High/p_50_1_vrpnc3.txt	-931.691	-931.691	-936.035	-936.035	3 600	3
instances/Z4/High/p_50_1_vrpnc4.txt	-999.18	-999.996	-1 011.01	-1 011.01	3 600	3
instances/Z4/High/p_50_1_vrpnc5.txt	-1 053.02	-1 053.23	-1 054.67	-1 054.67	3 600	3
instances/Z4/High/p_50_1_vrpnc7.txt	-809.698	-809.962	-830.435	-830.435	3 600	5
instances/Z4/High/p_50_1_vrpnc8.txt	-955.155	-955.582	-972.82	-972.82	3 600	3
instances/Z4/High/p_50_1_vrpnc9.txt	-958.063	-958.063	-962.187	-962.187	3 600	3
instances/Z4/High/p_50_2_vrpnc10.txt	-985.232	-985.242	-988.19	-988.19	3 600	3
instances/Z4/High/p_50_2_vrpnc2_k.txt	-881.354	-881.767	-894.751	-894.751	21 600	4
instances/Z4/High/p_50_2_vrpnc3.txt	-1 024.06	-1 024.67	-1 046.52	-1 046.52	3 600	3
instances/Z4/High/p_50_2_vrpnc4.txt	-975.837	-977.054	-984.008	-984.008	3 600	3
instances/Z4/High/p_50_2_vrpnc5.txt	-959.698	-959.698	-959.698	-959.698	3 600	3
instances/Z4/High/p_50_2_vrpnc7.txt	-870.826	-871	-888.481	-888.481	3 600	5
instances/Z4/High/p_50_2_vrpnc8.txt	-950.057	-950.82	-967.408	-967.408	3 600	4
instances/Z4/High/p_50_2_vrpnc9.txt	-1 040.31	-1 171.48	-1 183.95	-1 183.95	3 600	2

Table B: Generated Routes

instance name	routes
instances/Z4/High/p_50_1_vrpnc10.txt	1: (0 1 4 2 49 40 32 46 16 37 44 33 0) ,2: (0 28 7 18 8 20 19 12 0) ,3: (0 45 9 3 6 42 27 50 38 29 5 39 25 0) ,
instances/Z4/High/p_50_1_vrpnc11.txt	1: (0 32 14 45 20 35 4 51) ,2: (0 3 22 25 17 31 11 36 42 30 21 38 8 19 0) ,3: (0 12 50 39 48 6 40 24 28 37 23 26 15 33 29 0) ,
instances/Z4/High/p_50_1_vrpnc12.txt	1: (0 13 40 29 8 6 2 3 0) ,2: (0 15 1 42 31 16 19 23 48 25 7 27 41 5 0) ,3: (0 22 10 35 30 37 43 47 4 46 50 17 49 0) ,4: (0 45 33 36 20 9 39 11 34 28 12 26 0) ,
instances/Z4/High/p_50_1_vrpnc13.txt	1: (0 39 30 29 11 32 33 21 44 16 34 41 27 43 47 42 10 18 9 0) ,
instances/Z4/High/p_50_1_vrpnc14.txt	1: (0 11 22 38 37 5 33 31 4 7 26 14 28 0) ,2: (0 27 8 12 30 13 19 47 17 2 23 0) ,3: (0 46 29 20 34 3 43 1 41 50 6 21 44 39 0) ,
instances/Z4/High/p_50_1_vrpnc2.txt	1: (0 2 30 42 16 15 50 24 0) ,2: (0 11 17 36 45 48 18 46 27 4 0) ,3: (0 32 43 20 10 21 14 8 39 26 0) ,4: (0 33 31 34 7 40 0) ,5: (0 47 38 12 35 1 49 0) ,
instances/Z4/High/p_50_1_vrpnc3.txt	1: (0 34 20 40 9 23 48 28 45 38 41 27 0) ,2: (0 47 11 17 35 32 16 3 22 2 4 39 12 31 7 30 43 24 0) ,3: (0 50 44 42 15 18 1 5 6 37 25 8 14 10 29 0) ,
instances/Z4/High/p_50_1_vrpnc4.txt	1: (0 13 4 45 8 6 3 44 31 15 1 22 9 40 18 0) ,2: (0 17 39 37 20 25 2 23 10 19 27 38 0) ,3: (0 41 46 36 29 42 24 26 28 33 47 14 48 43 5 0) ,
instances/Z4/High/p_50_1_vrpnc5.txt	1: (0 5 42 50 22 29 45 38 7 3 24 32 4 0) ,2: (0 27 36 30 12 37 17 23 34 40 11 20 6 10 0) ,3: (0 31 28 1 14 16 46 18 44 39 35 41 49 0) ,
instances/Z4/High/p_50_1_vrpnc7.txt	1: (0 2 26 3 50 32 0) ,2: (0 4 46 29 33 14 5 25 31 30 0) ,3: (0 19 17 43 22 18 0) ,4: (0 20 48 21 38 10 16 42 0) ,5: (0 40 24 34 27 11 47 35 13 39 0) ,
instances/Z4/High/p_50_1_vrpnc8.txt	1: (0 25 10 29 36 40 14 27 47 45 48 50 24 0) ,2: (0 33 7 26 13 23 5 42 6 11 0) ,3: (0 37 31 34 2 38 43 28 8 49 4 9 12 39 0) ,
instances/Z4/High/p_50_1_vrpnc9.txt	1: (0 29 50 12 35 17 15 10 22 4 45 9 24 5 14 0) ,2: (0 30 26 32 1 18 11 6 39 23 7 42 8 0) ,3: (0 41 25 36 33 46 47 40 2 3 37 0) ,
instances/Z4/High/p_50_2_vrpnc10.txt	1: (0 4 18 42 35 14 47 21 31 30 45 49 0) ,2: (0 16 7 36 43 40 5 24 2 29 27 48 28 0) ,3: (0 23 10 3 12 33 25 44 34 9 46 17 0) ,
instances/Z4/High/p_50_2_vrpnc2_k.txt	1: (0 27 23 48 49 24 37 9 47 0) ,2: (0 31 2 16 26 41 13 0) ,3: (0 12 38 10 5 33 34 44 0) ,4: (0 19 18 43 15 6 42 40 39 35 8 0) ,
instances/Z4/High/p_50_2_vrpnc3.txt	1: (0 4 21 11 26 8 12 29 48 49 14 25 50 36 43 5 10 0) ,2: (0 31 30 24 20 17 13 33 51) ,3: (0 38 3 39 22 45 23 28 6 19 2 42 32 44 7 0) ,

instances/Z4/High/p_50_2_vrpnc4.txt	1: (0 17 11 44 6 27 36 3 47 16 49 40 9 0) ,2: (0 20 28 26 37 24 19 39 33 4 22 31 48 43 0) ,3: (0 34 30 14 5 50 29 42 1 25 2 38 15 46 0) ,
instances/Z4/High/p_50_2_vrpnc5.txt	1: (0 16 32 25 27 41 11 49 24 48 15 14 0) ,2: (0 21 7 20 5 31 3 29 34 13 28 1 33 0) ,3: (0 30 8 10 9 50 39 37 12 6 23 40 19 43 0) ,
instances/Z4/High/p_50_2_vrpnc7.txt	1: (0 7 45 42 48 37 9 22 0) ,2: (0 14 44 6 32 50 10 0) ,3: (0 27 41 36 11 47 16 2 33 18 19 0) ,4: (0 28 5 1 13 4 24 17 0) ,5: (0 29 49 26 38 35 0) ,
instances/Z4/High/p_50_2_vrpnc8.txt	1: (0 23 49 45 25 24 27 7 12 0) ,2: (0 40 2 26 5 9 36 42 3 29 16 41 0) ,3: (0 48 14 35 33 21 44 28 1 19 0) ,4: (0 50 17 43 15 13 37 22 32 10 47 46 11 6 0) ,
instances/Z4/High/p_50_2_vrpnc9.txt	1: (0 2 3 16 18 36 40 50 44 22 11 14 1 35 33 0) ,2: (0 39 48 45 20 49 34 29 32 9 23 31 42 10 8 0) ,

6.3 Four zone low threshold

Table A: Computational Results

instance name	UB	LB tree	LB root	LB root+cuts	time [s]	num vehicles
instances/Z4/Low/p_50_1_vrpnc10.txt	-263.486	-263.486	-272.412	-272.412	13.1499	3
instances/Z4/Low/p_50_1_vrpnc11.txt	-1 240,33	-1 272,12	-1 305,43	-1 305,43	72 401,2	3
instances/Z4/Low/p_50_1_vrpnc12.txt	-350.901	-350.901	-350.901	-350.901	6.73141	4
instances/Z4/Low/p_50_1_vrpnc13.txt	-1 334,51	-1 334,51	-1 337,91	-1 337,91	3 583,32	2
instances/Z4/Low/p_50_1_vrpnc14.txt	-307.141	-307.932	-311.27	-311.27	144.054	3
instances/Z4/Low/p_50_1_vrpnc2.txt	-221.406	-221.406	-223.501	-223.501	4.77546	3
instances/Z4/Low/p_50_1_vrpnc3.txt	-277.209	-277.25	-286.957	-286.957	285.993	2
instances/Z4/Low/p_50_1_vrpnc4.txt	-273.553	-273.857	-278.736	-278.736	27.7822	3
instances/Z4/Low/p_50_1_vrpnc5.txt	-292.195	-292.249	-304.982	-304.982	35.0172	3
instances/Z4/Low/p_50_1_vrpnc7.txt	-164.6	-164.852	-174.989	-174.989	36.6866	4
instances/Z4/Low/p_50_1_vrpnc8.txt	-278.928	-278.928	-292.78	-292.78	88.4097	2
instances/Z4/Low/p_50_1_vrpnc9.txt	-265.68	-265.68	-270.021	-270.021	8.59062	3
instances/Z4/Low/p_50_2_vrpnc10.txt	-310.161	-310.938	-312.289	-312.289	38.1387	3
instances/Z4/Low/p_50_2_vrpnc2.txt	-220.333	-220.333	-220.333	-220.333	1.37916	4
instances/Z4/Low/p_50_2_vrpnc3.txt	-295.779	-295.779	-297.751	-297.751	102.195	2
instances/Z4/Low/p_50_2_vrpnc4.txt	-261.315	-261.91	-269.008	-269.008	201.76	3
instances/Z4/Low/p_50_2_vrpnc5.txt	-257.242	-257.983	-263.832	-263.832	18.2144	3
instances/Z4/Low/p_50_2_vrpnc7.txt	-182.42	-183	-199.521	-199.521	218.477	4
instances/Z4/Low/p_50_2_vrpnc8.txt	-247.303	-247.778	-253.974	-253.974	437.304	3
instances/Z4/Low/p_50_2_vrpnc9.txt	-399.843	-399.955	-417.204	-417.204	2 387,37	3

Table B: Generated Routes

instance name	routes
instances/Z4/Low/p_50_1_vrpnc10.txt	1: (0 9 3 6 42 27 50 17 0) ,2: (0 12 19 2 49 40 32 46 16 37 44 33 0) ,3: (0 30 13 38 29 5 39 25 28 7 18 8 20 0) ,
instances/Z4/Low/p_50_1_vrpnc11.txt	1: (0 32 14 20 45 35 51) ,2: (0 29 15 26 23 37 12 50 39 48 6 40 24 28 0) ,3: (0 38 21 30 42 36 19 11 31 17 25 22 3 4 0) ,
instances/Z4/Low/p_50_1_vrpnc12.txt	1: (0 12 28 13 40 29 8 6 2 0) ,2: (0 15 1 42 31 16 19 23 48 25 7 27 41 5 0) ,3: (0 33 36 20 9 39 11 34 45 0) ,4: (0 50 46 17 4 43 47 37 30 35 10 22 0) ,
instances/Z4/Low/p_50_1_vrpnc13.txt	1: (0 8 16 34 41 27 43 47 42 10 18 2 9 15 4 0) ,2: (0 26 6 31 7 37 50 39 49 12 40 30 29 11 32 33 21 20 44 35 0) ,
instances/Z4/Low/p_50_1_vrpnc14.txt	1: (0 11 22 38 37 5 33 31 24 4 7 26 14 28 0) ,2: (0 27 8 12 30 13 19 47 17 16 2 0) ,3: (0 44 21 6 50 41 1 43 3 34 20 29 46 49 0) ,
instances/Z4/Low/p_50_1_vrpnc2.txt	1: (0 17 18 48 45 36 42 16 15 12 0) ,2: (0 33 31 34 7 40 0) ,3: (0 43 20 10 21 14 8 47 38 35 0) ,
instances/Z4/Low/p_50_1_vrpnc3.txt	1: (0 34 8 25 37 1 18 15 3 22 16 2 4 32 35 17 11 0) ,2: (0 41 9 23 48 28 45 38 12 31 39 30 7 43 24 0) ,
instances/Z4/Low/p_50_1_vrpnc4.txt	1: (0 17 39 3 6 21 8 44 31 15 1 45 4 13 0) ,2: (0 22 9 40 46 36 42 24 26 28 33 47 43 48 14 0) ,3: (0 49 23 2 25 20 37 10 19 27 0) ,
instances/Z4/Low/p_50_1_vrpnc5.txt	1: (0 5 42 50 22 29 45 7 38 0) ,2: (0 17 37 12 30 36 27 24 3 32 4 0) ,3: (0 41 35 39 44 18 16 14 1 6 20 11 40 34 0) ,
instances/Z4/Low/p_50_1_vrpnc7.txt	1: (0 4 46 29 33 14 5 25 31 30 0) ,2: (0 20 19 48 21 38 10 16 0) ,3: (0 32 50 3 26 2 0) ,4: (0 40 24 34 27 11 47 35 13 39 0) ,
instances/Z4/Low/p_50_1_vrpnc8.txt	1: (0 13 26 7 33 48 45 47 27 14 40 36 29 10 0) ,2: (0 37 31 34 2 38 43 28 8 49 4 9 12 39 11 42 23 0) ,
instances/Z4/Low/p_50_1_vrpnc9.txt	1: (0 14 5 24 9 45 4 22 10 15 17 35 12 50 29 0) ,2: (0 30 26 32 1 18 11 6 39 23 7 42 8 0) ,3: (0 41 25 36 33 46 47 40 2 3 37 0) ,
instances/Z4/Low/p_50_2_vrpnc10.txt	1: (0 20 17 46 9 34 44 25 33 12 19 3 10 23 8 0) ,2: (0 26 35 14 47 21 31 45 49 30 42 18 4 0) ,3: (0 29 2 24 40 7 36 43 5 27 48 28 0) ,
instances/Z4/Low/p_50_2_vrpnc2.txt	1: (0 20 31 19 18 15 6 42 40 39 43 0) ,2: (0 23 48 33 34 5 10 38 44 0) ,3: (0 41 13 47 9 37 24 49 0) ,4: (0 46 2 16 26 36 29 0) ,
instances/Z4/Low/p_50_2_vrpnc3.txt	1: (0 20 30 24 12 8 26 11 4 21 39 3 22 45 38 0) ,2: (0 33 29 48 49 14 25 50 36 43 5 32 42 2 6 15 28 23 0) ,
instances/Z4/Low/p_50_2_vrpnc4.txt	1: (0 17 11 36 27 6 44 34 30 14 5 50 0) ,2: (0 22 4 33 39 19 24 37 26 28 42 1 29 25 2 38 15 0) ,3: (0 31 48 40 49 16 47 3 9 43 0) ,

instances/Z4/Low/p_50_2_vrpnc5.txt	1: (0 16 32 25 27 41 11 49 24 48 15 14 0) ,2: (0 30 8 10 9 50 39 37 12 6 23 40 19 43 0) ,3: (0 33 1 28 13 34 29 3 31 5 20 44 7 0) ,
instances/Z4/Low/p_50_2_vrpnc7.txt	1: (0 10 50 32 6 44 14 0) ,2: (0 17 24 4 13 1 5 35 0) ,3: (0 45 49 26 38 0) ,4: (0 48 19 33 2 16 47 11 36 41 27 0) ,
instances/Z4/Low/p_50_2_vrpnc8.txt	1: (0 20 29 43 50 17 15 13 37 22 10 47 46 11 32 0) ,2: (0 23 49 19 44 21 45 25 24 7 27 39 0) ,3: (0 48 40 9 42 36 5 26 2 14 33 35 28 1 0) ,
instances/Z4/Low/p_50_2_vrpnc9.txt	1: (0 9 23 31 42 8 10 21 33 35 1 14 11 22 0) ,2: (0 26 4 28 32 29 34 49 20 13 45 48 39 0) ,3: (0 44 50 40 36 18 16 3 2 17 15 0) ,

6.4 .Six zone average threshold

Table A: Computational Results

instance name	UB	LB tree	LB root	LB root+cuts	time limit [s]	num vehicles
instances/Z6/average/p_50_1_vrpnc10.txt	-475.765	-475.908	-489.402	-489.402	3 600	3
instances/Z6/average/p_50_1_vrpnc11.txt	-1 897,55	-1 898	-1 917,32	-1 917,32	21 600	3
instances/Z6/average/p_50_1_vrpnc12.txt	-558.966	-561.299	-576.932	-576.932	3 600	4
instances/Z6/average/p_50_1_vrpnc13.txt	-2 133,31	-2 140,09	-2 156,79	-2 156,79	21 600	2
instances/Z6/average/p_50_1_vrpnc14.txt	-576.777	-576.777	-581.769	-581.769	3 600	3
instances/Z6/average/p_50_1_vrpnc2.txt	-439.811	-439.869	-449.048	-449.048	3 600	3
instances/Z6/average/p_50_1_vrpnc3.txt	-502.25	-502.67	-515.478	-515.478	3 600	3
instances/Z6/average/p_50_1_vrpnc4.txt	-91 886,6	-91 886,7	-91 897,8	-91 897,8	3 600	2
instances/Z6/average/p_50_1_vrpnc5.txt	-543.265	-543.915	-548.216	-548.216	3 600	3
instances/Z6/average/neue_50_02_vrpnc7.txt	-378.014	-378.979	-389.263	-389.263	21 600	3
instances/Z6/average/p_50_1_vrpnc8.txt	-512.42	-512.736	-528.268	-528.268	3 600	3
instances/Z6/average/p_50_1_vrpnc9.txt	-457.37	-457.37	-468.854	-468.854	3 600	3
instances/Z6/average/p_50_2_vrpnc10.txt	-521.046	-521.046	-521.046	-521.046	3 600	3
instances/Z6/average/p_50_2_vrpnc2.txt	-461.851	-462.719	-483.665	-483.665	3 600	4
instances/Z6/average/p_50_2_vrpnc3.txt	-535.22	-535.504	-544.159	-544.159	21 600	2
instances/Z6/average/p_50_2_vrpnc4.txt	-457.874	-457.999	-468.967	-468.967	21 600	3
instances/Z6/average/p_50_2_vrpnc5.txt	-481.695	-481.695	-481.695	-481.695	3 600	3
instances/Z6/average/p_50_2_vrpnc7.txt	-416.336	-416.983	-435.557	-435.557	21 600	4
instances/Z6/average/p_50_2_vrpnc8.txt	-502.371	-502.371	-506.359	-506.359	3 600	3
instances/Z6/average/p_50_2_vrpnc9.txt	-613.66	-613.66	-631.828	-631.828	3 600	3

Table B: Generated Routes

instance name	routes
instances/Z6/average/p_50_1_vrpnc10.txt	1: (0 12 19 2 49 40 32 46 16 44 37 0) ,2: (0 13 38 50 27 42 6 3 9 0) ,3: (0 48 29 5 39 25 28 7 18 8 20 10 0) ,
instances/Z6/average/p_50_1_vrpnc11.txt	1: (0 10 25 22 17 31 11 19 36 42 30 21 38 8 44 0) ,2: (0 18 29 33 15 26 23 37 28 24 40 6 48 39 50 12 9 16 13 0) ,3: (0 43 49 4 35 20 45 14 32 47 41 0) ,
instances/Z6/average/p_50_1_vrpnc12.txt	1: (0 11 39 9 20 36 33 0) ,2: (0 25 41 27 7 48 23 19 16 31 42 0) ,3: (0 40 29 8 6 2 0) ,4: (0 49 17 50 46 43 47 37 32 22 30 35 10 1 0) ,
instances/Z6/average/p_50_1_vrpnc13.txt	1: (0 28 24 3 35 44 20 21 33 32 11 29 30 12 49 39 50 37 7 31 6 26 0) ,2: (0 4 15 2 48 46 18 10 42 47 43 27 41 34 36 16 8 19 0) ,
instances/Z6/average/p_50_1_vrpnc14.txt	1: (0 25 10 48 9 27 8 30 12 28 14 26 7 4 24 0) ,2: (0 36 13 19 47 17 16 20 34 3 43 1 0) ,3: (0 37 5 31 33 38 22 21 41 50 6 0) ,
instances/Z6/average/p_50_1_vrpnc2.txt	1: (0 17 18 48 45 36 42 16 15 12 0) ,2: (0 20 10 21 14 8 47 38 35 24 0) ,3: (0 27 4 7 34 31 33 0) ,
instances/Z6/average/p_50_1_vrpnc3.txt	1: (0 10 14 34 8 25 37 6 5 1 18 15 42 50 0) ,2: (0 11 30 7 31 12 39 4 2 22 3 16 32 35 17 0) ,3: (0 27 41 38 45 28 48 23 9 40 20 0) ,
instances/Z6/average/p_50_1_vrpnc4.txt	1: (0 3 8 45 4 22 1 15 31 44 10 27 0) ,2: (0 12 41 40 9 46 36 29 42 24 26 28 33 47 48 35 0) ,
instances/Z6/average/p_50_1_vrpnc5.txt	1: (0 2 37 12 30 36 27 24 3 32 4 7 38 45 0) ,2: (0 25 47 21 17 23 34 40 11 20 6 10 0) ,3: (0 31 28 1 14 16 46 41 18 35 44 39 50 22 29 42 0) ,
instances/Z6/average/neue_50_02_vrpnc7.txt	1: (0 2 26 3 50 32 0) ,2: (0 16 10 38 21 48 24 34 40 20 0) ,3: (0 27 47 13 46 29 33 14 25 31 0) ,
instances/Z6/average/p_50_1_vrpnc8.txt	1: (0 19 8 28 43 38 2 34 31 37 22 0) ,2: (0 23 5 42 11 6 12 9 4 49 39 0) ,3: (0 29 36 40 14 27 47 45 48 50 33 7 26 13 0) ,
instances/Z6/average/p_50_1_vrpnc9.txt	1: (0 8 38 42 7 23 39 6 11 18 32 1 0) ,2: (0 29 50 12 35 17 15 10 22 4 45 9 24 5 14 0) ,3: (0 30 26 33 36 25 46 47 40 2 3 37 0) ,
instances/Z6/average/p_50_2_vrpnc10.txt	1: (0 4 18 42 35 14 47 21 31 30 45 49 0) ,2: (0 20 41 17 46 9 34 44 25 33 12 10 23 3 19 0) ,3: (0 50 32 7 36 43 40 24 5 27 48 28 0) ,
instances/Z6/average/p_50_2_vrpnc2.txt	1: (0 19 18 43 15 6 42 40 39 12 0) ,2: (0 23 48 33 34 5 10 38 0) ,3: (0 36 26 16 2 20 31 0) ,4: (0 49 24 37 9 45 13 41 29 0) ,
instances/Z6/average/p_50_2_vrpnc3.txt	1: (0 19 2 42 32 44 7 5 43 36 50 25 14 49 48 29 12 0) ,2: (0 24 8 26 11 4 21 39 3 22 45 23 28 6 15 0) ,
instances/Z6/average/p_50_2_vrpnc4.txt	1: (0 33 4 22 31 48 40 49 16 47 3 9 43 0) ,2: (0 32 36 27 6 11 17 44 34 30 5 14 0) ,3: (0 15 38 25 50 29 1 42 28 26 20 8 37 24 19 39 0) ,

instances/Z6/average/p_50_2_vrpnc5.txt	1: (0 7 44 20 5 31 3 29 34 13 28 1 0) ,2: (0 8 10 9 50 39 37 12 6 23 40 19 43 0) ,3: (0 16 32 25 27 11 49 24 48 15 14 0) ,
instances/Z6/average/p_50_2_vrpnc7.txt	1: (0 35 38 26 49 42 0) ,2: (0 10 50 41 27 36 11 16 18 0) ,3: (0 17 28 5 1 13 6 4 0) ,4: (0 7 45 48 37 19 33 2 23 0) ,
instances/Z6/average/p_50_2_vrpnc8.txt	1: (0 12 7 27 24 25 45 21 33 35 14 48 0) ,2: (0 32 11 46 47 10 22 37 13 15 17 50 43 20 0) ,3: (0 40 2 26 5 9 36 42 3 29 16 41 0) ,
instances/Z6/average/p_50_2_vrpnc9.txt	1: (0 19 8 10 42 31 23 9 28 32 29 49 20 0) ,2: (0 33 35 1 14 11 22 51) ,3: (0 45 48 39 15 17 2 3 16 18 36 40 50 0) ,

6.5 Six zone high threshold

Table A: Computational Results

instance name	UB	LB tree	LB root	LB root+cuts	time [s]	num vehicles
instances/Z6/High/p_50_1_vrpnc10.txt	-860.789	-860.789	-861.098	-861.098	6.45744	3
instances/Z6/High/p_50_1_vrpnc11.txt	-3 138,69	-3 146,28	-3 170,37	-3 170,37	21 600,4	3
instances/Z6/High/p_50_1_vrpnc12.txt	-1 212,42	-1 212,61	-1 220,54	-1 220,54	45.1046	4
instances/Z6/High/p_50_1_vrpnc13.txt	-3 497,69	-3 520,31	-3 537,08	-3 537,08	21 600,4	3
instances/Z6/High/p_50_1_vrpnc14.txt	-1 170,67	-1 170,67	-1 173,11	-1 173,11	20.0071	3
instances/Z6/High/p_50_1_vrpnc2.txt	-893.061	-900.319	-920.735	-920.735	3 600,08	4
instances/Z6/High/p_50_1_vrpnc3.txt	-965.272	-965.996	-977.262	-977.262	30.2832	3
instances/Z6/High/p_50_1_vrpnc4.txt	-1 051,38	-1 051,56	-1 057,93	-1 057,93	282.395	3
instances/Z6/High/p_50_1_vrpnc5.txt	-1 014,08	-1 014,89	-1 024,39	-1 024,39	540.781	3
instances/Z6/High/p_50_1_vrpnc7.txt	-826.126	-826.658	-843.723	-843.723	475.453	4
instances/Z6/High/p_50_1_vrpnc8.txt	-974.128	-974.971	-998.861	-998.861	150.701	3
instances/Z6/High/p_50_1_vrpnc9.txt	-953.78	-953.971	-962.443	-962.443	157.423	3
instances/Z6/High/p_50_2_vrpnc10.txt	-926.514	-926.514	-926.514	-926.514	1.81022	3
instances/Z6/High/p_50_2_vrpnc2.txt	-933.816	-933.996	-966.591	-966.591	3 052,73	4
instances/Z6/High/p_50_2_vrpnc3.txt	-1 016,13	-1 017	-1 038,41	-1 038,41	2 761,03	2
instances/Z6/High/p_50_2_vrpnc4.txt	-963.944	-963.965	-970.442	-970.442	899.597	3
instances/Z6/High/p_50_2_vrpnc5.txt	-959.083	-959.083	-959.083	-959.083	2.12899	3
instances/Z6/High/p_50_2_vrpnc7.txt	-893.492	-893.983	-908.056	-908.056	437.167	5
instances/Z6/High/p_50_2_vrpnc8.txt	-1 015,06	-1 015,06	-1 015,06	-1 015,06	1.25585	3
instances/Z6/High/p_50_2_vrpnc9.txt	-1 172,65	-1 172,84	-1 182,9	-1 182,9	109.432	3

Table B: Generated Routes

instance name	routes
instances/Z6/High/p_50_1_vrpnc10.txt	1: (0 10 20 8 18 7 28 48 25 39 5 29 11 0) ,2: (0 12 19 2 49 40 32 46 16 44 37 0) ,3: (0 17 47 9 3 6 42 27 50 38 13 30 0) ,
instances/Z6/High/p_50_1_vrpnc11.txt	1: (0 41 32 14 45 20 10 4 35 49 43 1 13 5 0) ,2: (0 19 8 38 21 30 42 36 11 31 17 25 22 3 47 0) ,3: (0 18 29 33 15 26 23 37 28 24 40 6 48 39 50 12 44 9 16 0) ,
instances/Z6/High/p_50_1_vrpnc12.txt	1: (0 15 1 42 31 16 19 23 48 7 27 41 25 0) ,2: (0 28 40 29 8 6 2 0) ,3: (0 32 22 10 35 30 37 47 43 4 17 46 50 0) ,4: (0 49 33 36 20 9 39 11 34 0) ,
instances/Z6/High/p_50_1_vrpnc13.txt	1: (0 14 24 19 3 28 51) ,2: (0 26 6 31 37 7 50 39 49 40 12 30 29 11 32 33 21 20 44 35 8 0) ,3: (0 36 16 34 41 27 43 47 42 10 18 46 48 2 9 15 0) ,
instances/Z6/High/p_50_1_vrpnc14.txt	1: (0 1 43 3 34 20 29 16 17 47 19 13 0) ,2: (0 11 22 37 5 31 33 38 21 41 50 6 44 0) ,3: (0 36 10 48 9 27 8 30 12 28 14 26 7 4 24 0) ,
instances/Z6/High/p_50_1_vrpnc2.txt	1: (0 4 27 28 46 18 48 45 36 16 42 17 0) ,2: (0 11 30 50 15 12 24 0) ,3: (0 20 10 21 14 8 47 38 35 1 49 0) ,4: (0 39 33 31 34 7 40 0) ,
instances/Z6/High/p_50_1_vrpnc3.txt	1: (0 10 14 34 8 25 37 6 5 1 18 15 42 0) ,2: (0 24 31 12 38 45 28 48 23 9 40 20 0) ,3: (0 47 11 17 35 32 16 3 22 2 4 39 30 7 43 0) ,
instances/Z6/High/p_50_1_vrpnc4.txt	1: (0 5 43 48 14 47 33 28 26 24 42 36 46 29 18 0) ,2: (0 17 39 3 6 8 44 31 15 1 45 4 22 9 40 0) ,3: (0 49 23 2 25 20 37 10 19 27 38 0) ,
instances/Z6/High/p_50_1_vrpnc5.txt	1: (0 5 42 50 22 29 45 38 7 3 24 32 4 0) ,2: (0 27 36 30 12 37 17 23 34 40 11 20 6 10 0) ,3: (0 31 28 1 14 16 46 18 44 39 35 41 49 0) ,
instances/Z6/High/p_50_1_vrpnc7.txt	1: (0 16 26 3 2 32 0) ,2: (0 19 17 48 21 38 10 22 18 0) ,3: (0 35 13 47 11 27 34 24 40 20 0) ,4: (0 46 29 33 14 5 25 31 50 30 0) ,
instances/Z6/High/p_50_1_vrpnc8.txt	1: (0 33 7 26 13 23 5 42 6 11 0) ,2: (0 37 31 34 2 38 43 28 8 49 4 9 12 39 0) ,3: (0 48 45 47 27 14 40 36 29 10 25 0) ,
instances/Z6/High/p_50_1_vrpnc9.txt	1: (0 8 38 42 7 23 39 6 11 18 32 1 0) ,2: (0 29 50 12 35 17 15 10 22 4 49 45 9 24 0) ,3: (0 30 26 33 46 36 25 37 3 2 47 40 5 14 0) ,
instances/Z6/High/p_50_2_vrpnc10.txt	1: (0 19 3 23 10 12 33 25 44 34 9 46 17 41 20 0) ,2: (0 49 45 30 31 21 47 14 35 42 18 0) ,3: (0 50 32 7 36 43 40 5 27 48 28 0) ,
instances/Z6/High/p_50_2_vrpnc2.txt	1: (0 8 35 39 40 42 6 15 43 18 19 0) ,2: (0 23 48 33 34 5 10 12 0) ,3: (0 29 41 13 45 9 37 24 49 0) ,4: (0 36 26 16 2 20 31 0) ,
instances/Z6/High/p_50_2_vrpnc3.txt	1: (0 33 29 48 49 14 25 50 36 43 5 10 7 44 32 42 2 19 0) ,2: (0 30 24 12 8 26 11 4 21 39 3 22 45 23 28 6 46 0) ,

instances/Z6/High/p_50_2_vrpnc4.txt	1: (0 17 11 44 6 27 36 3 47 16 49 40 9 0) ,2: (0 31 48 22 4 33 39 19 24 37 8 26 28 20 0) ,3: (0 46 15 38 2 25 1 42 29 50 5 14 30 34 0) ,
instances/Z6/High/p_50_2_vrpnc5.txt	1: (0 8 10 9 50 39 37 17 12 6 23 40 19 45 0) ,2: (0 21 7 44 20 5 31 3 29 34 13 28 33 0) ,3: (0 32 25 27 41 11 49 24 48 15 14 43 0) ,
instances/Z6/High/p_50_2_vrpnc7.txt	1: (0 10 50 41 27 36 11 47 0) ,2: (0 28 17 24 1 5 35 7 0) ,3: (0 38 26 49 45 0) ,4: (0 42 48 37 19 33 2 16 23 18 0) ,5: (0 44 4 13 6 32 14 0) ,
instances/Z6/High/p_50_2_vrpnc8.txt	1: (0 1 28 44 49 45 25 24 27 7 12 6 0) ,2: (0 11 46 47 10 22 37 13 17 15 43 29 16 41 0) ,3: (0 40 3 42 36 9 5 26 2 14 35 33 21 19 0) ,
instances/Z6/High/p_50_2_vrpnc9.txt	1: (0 15 39 48 45 20 49 34 29 32 28 4 26 0) ,2: (0 22 11 14 1 35 33 21 10 8 42 31 23 9 0) ,3: (0 44 50 40 36 18 16 3 2 17 41 0) ,

6.6 Six zone low threshold

Table A: Computational Results

instance name	UB	LB tree	LB root	LB root+cuts	time limit [s]	num vehicles
instances/Z6/Low/p_50_1_vrpnc10.txt	-251.484	-251.484	-261.58	-261.58	3 600	3
instances/Z6/Low/p_50_1_vrpnc11.txt	-1 341.42	-1 349.44	-1 374.61	-1 374.61	21 600	3
instances/Z6/Low/p_50_1_vrpnc12.txt	-352.094	-352.885	-359.737	-359.737	3 600	4
instances/Z6/Low/p_50_1_vrpnc13.txt	-1 452.11	-1 452.73	-1 454.97	-1 454.97	21 600	2
instances/Z6/Low/p_50_1_vrpnc14.txt	-345.326	-345.326	-345.326	-345.326	3 600	3
instances/Z6/Low/p_50_1_vrpnc2.txt	-222.375	-222.607	-232.651	-232.651	3 600	3
instances/Z6/Low/p_50_1_vrpnc3.txt	-290.213	-290.289	-301.269	-301.269	3 600	2
instances/Z6/Low/p_50_1_vrpnc4.txt	-291.013	-291.374	-293.4	-293.4	3 600	3
instances/Z6/Low/p_50_1_vrpnc5.txt	-283.03	-283.03	-295.993	-295.993	3 600	3
instances/Z6/Low/p_50_1_vrpnc7.txt	-172.999	-172.999	-181.963	-181.963	3 600	3
instances/Z6/Low/p_50_1_vrpnc8.txt	-275.42	-275.94	-289.318	-289.318	3 600	2
instances/Z6/Low/p_50_1_vrpnc9.txt	-263.669	-263.669	-266.952	-266.952	3 600	2
instances/Z6/Low/p_50_2_vrpnc10.txt	-284.733	-284.733	-285.391	-285.391	3 600	3
instances/Z6/Low/p_50_2_vrpnc2.txt	-230.552	-230.971	-243.237	-243.237	3 600	4
instances/Z6/Low/p_50_2_vrpnc3.txt	-295.509	-319.429	-324.856	-324.856	3 600	2
instances/Z6/Low/p_50_2_vrpnc4.txt	-256.787	-256.939	-268.268	-268.268	21 600	3
instances/Z6/Low/p_50_2_vrpnc5.txt	-250.031	-250.238	-257.871	-257.871	3 600	3
instances/Z6/Low/p_50_2_vrpnc7.txt	-187.788	-187.984	-204.958	-204.958	3 600	4
instances/Z6/Low/p_50_2_vrpnc8.txt	-281.827	-281.999	-289.501	-289.501	3 600	3
instances/Z6/Low/p_50_2_vrpnc9.txt	-402.262	-402.575	-414.479	-414.479	3 600	3

Table B: Generated Routes

instance name	routes
instances/Z6/Low/p_50_1_vrpnc10.txt	1: (0 12 19 2 49 40 32 46 16 44 37 0) ,2: (0 17 50 27 42 6 3 9 0) ,3: (0 30 13 38 29 5 39 25 48 28 7 18 8 20 0) ,
instances/Z6/Low/p_50_1_vrpnc11.txt	1: (0 41 3 22 25 17 31 11 36 42 30 21 38 19 44 0) ,2: (0 47 32 14 45 20 10 4 35 49 43 1 0) ,3: (0 18 29 33 15 26 23 37 28 24 40 6 48 39 50 12 9 16 13 0) ,
instances/Z6/Low/p_50_1_vrpnc12.txt	1: (0 25 41 27 7 48 23 19 44 16 31 42 1 15 0) ,2: (0 28 13 40 29 8 6 2 0) ,3: (0 32 22 10 35 30 37 47 43 4 17 46 50 0) ,4: (0 49 33 36 20 9 39 11 34 0) ,
instances/Z6/Low/p_50_1_vrpnc13.txt	1: (0 8 16 34 41 27 43 47 42 10 18 46 48 9 2 15 0) ,2: (0 3 19 35 44 20 21 33 32 11 29 30 40 12 49 39 50 37 7 31 6 26 0) ,
instances/Z6/Low/p_50_1_vrpnc14.txt	1: (0 1 43 3 34 20 29 16 17 47 19 13 0) ,2: (0 11 22 37 5 31 33 38 21 41 50 6 44 0) ,3: (0 27 8 30 12 28 14 26 7 4 24 0) ,
instances/Z6/Low/p_50_1_vrpnc2.txt	1: (0 4 27 17 18 48 45 36 42 16 15 0) ,2: (0 33 31 34 7 40 0) ,3: (0 35 12 38 47 8 14 21 10 0) ,
instances/Z6/Low/p_50_1_vrpnc3.txt	1: (0 34 8 25 37 1 18 15 3 22 16 2 4 32 35 17 11 7 0) ,2: (0 40 9 23 48 28 45 38 12 31 39 30 43 24 0) ,
instances/Z6/Low/p_50_1_vrpnc4.txt	1: (0 5 43 48 14 47 33 28 26 24 42 29 36 46 41 0) ,2: (0 17 39 3 44 31 8 45 13 4 15 1 22 9 40 0) ,3: (0 38 27 19 10 37 20 25 2 23 49 0) ,
instances/Z6/Low/p_50_1_vrpnc5.txt	1: (0 17 37 12 30 36 27 24 3 32 4 0) ,2: (0 34 40 11 20 6 1 14 16 18 44 39 35 41 0) ,3: (0 38 7 45 29 22 50 42 5 0) ,
instances/Z6/Low/p_50_1_vrpnc7.txt	1: (0 2 26 3 50 32 0) ,2: (0 13 46 29 33 14 5 25 31 30 0) ,3: (0 16 10 38 21 48 24 34 27 40 0) ,
instances/Z6/Low/p_50_1_vrpnc8.txt	1: (0 13 26 7 33 48 45 47 27 14 40 36 29 10 0) ,2: (0 23 42 11 39 12 9 4 49 8 28 43 38 2 34 31 37 0) ,
instances/Z6/Low/p_50_1_vrpnc9.txt	1: (0 2 47 40 5 24 9 22 10 15 17 35 12 50 29 0) ,2: (0 3 46 33 26 30 11 6 39 23 7 42 0) ,
instances/Z6/Low/p_50_2_vrpnc10.txt	1: (0 26 35 14 47 21 31 45 49 30 42 18 4 0) ,2: (0 41 17 46 9 34 44 25 33 12 10 23 3 19 0) ,3: (0 50 32 7 36 43 40 5 27 48 28 0) ,
instances/Z6/Low/p_50_2_vrpnc2.txt	1: (0 13 9 37 24 49 48 23 0) ,2: (0 20 31 19 18 15 6 42 40 39 43 0) ,3: (0 41 26 16 2 36 0) ,4: (0 44 34 33 5 10 38 12 0) ,
instances/Z6/Low/p_50_2_vrpnc3.txt	1: (0 13 17 20 41 30 24 8 26 11 4 21 39 3 22 45 0) ,2: (0 23 28 6 2 42 32 44 5 43 36 50 25 14 49 48 29 12 33 0) ,
instances/Z6/Low/p_50_2_vrpnc4.txt	1: (0 8 26 28 42 1 38 15 2 25 29 50 5 14 30 34 0) ,2: (0 9 40 49 16 47 3 36 27 6 44 11 17 0) ,3: (0 31 48 22 4 33 39 19 24 37 0) ,

instances/Z6/Low/p_50_2_vrpnc5.txt	1: (0 32 25 27 41 11 49 24 48 15 14 43 0) ,2: (0 33 1 28 13 34 29 3 31 5 20 44 7 0) ,3: (0 45 19 40 23 6 12 17 37 39 50 9 10 8 0) ,
instances/Z6/Low/p_50_2_vrpnc7.txt	1: (0 16 47 11 36 27 41 50 10 0) ,2: (0 17 28 5 1 13 6 4 0) ,3: (0 18 23 2 33 19 48 45 7 0) ,4: (0 42 49 26 38 35 0) ,
instances/Z6/Low/p_50_2_vrpnc8.txt	1: (0 8 11 46 47 10 32 22 37 13 15 17 50 0) ,2: (0 19 1 28 44 21 45 25 24 27 7 12 6 0) ,3: (0 23 49 33 35 14 2 26 5 9 36 42 3 29 0) ,
instances/Z6/Low/p_50_2_vrpnc9.txt	1: (0 4 28 32 29 34 49 20 13 45 48 39 15 0) ,2: (0 9 23 31 42 8 10 21 33 35 1 14 11 22 0) ,3: (0 44 50 40 36 18 16 3 2 17 41 0) ,

6.7 Five zone average threshold

Table A: Computational Results

instance name	UB	LB tree	LB root	LB root+cuts	time [s]	num vehicles
instances/Z5/average/p_50_1_vrpnc10.txt	-713.773	-713.933	-726.105	-726.105	96.8447	4
instances/Z5/average/p_50_1_vrpnc11.txt	-1 852.56	-1 853.4	-1 865.29	-1 865.29	21 600.3	3
instances/Z5/average/p_50_1_vrpnc12.txt	-778.201	-778.997	-791.123	-791.123	3 518.41	4
instances/Z5/average/p_50_1_vrpnc13.txt	-2 013.12	-2 082.32	-2 095.12	-2 095.12	23 169.2	3
instances/Z5/average/p_50_1_vrpnc14.txt	-747.241	-747.907	-756.785	-756.785	15.2133	4
instances/Z5/average/p_50_1_vrpnc2.txt	-629.866	-629.995	-638.654	-638.654	174.873	6
instances/Z5/average/p_50_1_vrpnc3.txt	-686.93	-686.997	-700.866	-700.866	5 433.87	4
instances/Z5/average/p_50_1_vrpnc4.txt	-700.166	-701	-723.956	-723.956	10 378.2	4
instances/Z5/average/p_50_1_vrpnc5.txt	-629.038	-629.29	-638.242	-638.242	77.0838	4
instances/Z5/average/p_50_1_vrpnc7.txt	-550.302	-550.999	-562.937	-562.937	895.851	7
instances/Z5/average/p_50_1_vrpnc8.txt	-691.129	-691.266	-700.075	-700.075	2 115.91	3
instances/Z5/average/p_50_1_vrpnc9.txt	-655.929	-655.995	-667.553	-667.553	2 709.78	4
instances/Z5/average/p_50_2_vrpnc10.txt	-694.826	-695.304	-704.123	-704.123	3 600.16	4
instances/Z5/average/p_50_2_vrpnc2.txt	-652.609	-652.976	-667.098	-667.098	806.754	6
instances/Z5/average/p_50_2_vrpnc3.txt	-726.737	-726.778	-731.111	-731.111	124.788	3
instances/Z5/average/p_50_2_vrpnc4.txt	-598.634	-598.997	-607.287	-607.287	1 936.5	3
instances/Z5/average/p_50_2_vrpnc5.txt	-607.772	-607.973	-620.377	-620.377	770.108	4
instances/Z5/average/p_50_2_vrpnc7.txt	-552.69	-553	-566.022	-566.022	2 525.64	7
instances/Z5/average/p_50_2_vrpnc8.txt	-703.382	-703.957	-714.441	-714.441	308.702	4
instances/Z5/average/p_50_2_vrpnc9.txt	-777.882	-781.285	-795.237	-795.237	8 421.34	4

Table B: Generated Routes

instance name	routes
instances/Z5/average/p_50_1_vrpnc10.txt	1: (0 12 19 2 49 40 32 4 1 14 36 0) ,2: (0 23 10 20 8 18 7 28 43 0) ,3: (0 31 33 46 16 44 37 27 42 6 3 9 45 26 0) ,4: (0 48 25 39 5 29 38 50 47 41 17 13 30 11 0) ,
instances/Z5/average/p_50_1_vrpnc11.txt	1: (0 5 16 9 44 12 50 39 48 6 40 24 28 37 23 26 15 29 18 0) ,2: (0 13 1 43 49 4 35 45 20 32 47 0) ,3: (0 19 8 38 21 30 42 36 11 31 17 25 22 3 41 0) ,
instances/Z5/average/p_50_1_vrpnc12.txt	1: (0 34 11 39 9 20 36 33 17 50 49 0) ,2: (0 44 19 23 48 25 7 27 41 5 3 21 0) ,3: (0 12 28 13 40 29 8 6 2 0) ,4: (0 22 32 47 43 37 30 35 10 16 31 42 1 15 0) ,
instances/Z5/average/p_50_1_vrpnc13.txt	1: (0 38 22 12 30 29 11 32 33 21 20 44 35 8 16 36 34 41 27 19 3 24 0) ,2: (0 43 47 42 10 18 48 9 2 15 4 0) ,3: (0 49 39 50 7 37 31 6 26 13 25 17 0) ,
instances/Z5/average/p_50_1_vrpnc14.txt	1: (0 18 45 27 8 30 12 28 14 26 7 4 24 0) ,2: (0 40 49 46 29 20 34 3 43 1 41 50 6 0) ,3: (0 23 2 17 47 19 13 36 10 48 9 42 25 0) ,4: (0 11 22 37 31 33 38 21 44 39 0) ,
instances/Z5/average/p_50_1_vrpnc2.txt	1: (0 2 42 36 45 48 18 17 11 41 0) ,2: (0 3 43 20 10 14 8 39 26 0) ,3: (0 5 47 38 12 35 1 49 23 0) ,4: (0 6 37 4 27 46 28 13 44 0) ,5: (0 22 24 50 15 16 30 9 0) ,6: (0 33 31 34 7 40 25 0) ,
instances/Z5/average/p_50_1_vrpnc3.txt	1: (0 13 21 24 31 39 30 7 43 11 47 46 0) ,2: (0 19 35 2 16 22 3 50 44 42 15 18 1 5 6 36 0) ,3: (0 26 20 34 8 25 37 14 10 29 0) ,4: (0 33 27 41 38 45 28 48 23 9 40 49 0) ,
instances/Z5/average/p_50_1_vrpnc4.txt	1: (0 30 14 48 43 47 33 28 26 24 42 18 11 0) ,2: (0 4 45 8 21 51) ,3: (0 17 39 3 44 31 15 1 22 9 40 41 46 36 29 12 0) ,4: (0 7 38 27 19 10 37 20 25 2 49 32 0) ,
instances/Z5/average/p_50_1_vrpnc5.txt	1: (0 5 42 50 22 29 45 7 38 0) ,2: (0 15 43 49 41 35 39 44 18 46 16 14 1 28 31 0) ,3: (0 25 2 47 27 32 3 24 36 30 12 37 17 21 0) ,4: (0 48 34 40 11 20 6 10 0) ,
instances/Z5/average/p_50_1_vrpnc7.txt	1: (0 6 26 3 2 30 37 0) ,2: (0 12 36 15 4 1 0) ,3: (0 18 22 43 8 23 0) ,4: (0 20 40 24 34 27 11 47 41 44 0) ,5: (0 28 31 50 32 45 7 0) ,6: (0 42 16 10 38 21 48 17 19 0) ,7: (0 49 5 14 33 29 46 13 35 39 0) ,
instances/Z5/average/p_50_1_vrpnc8.txt	1: (0 15 35 23 5 42 11 6 12 9 4 49 39 20 21 0) ,2: (0 18 19 8 28 43 38 2 34 31 37 22 17 46 0) ,3: (0 26 7 50 48 45 47 24 30 27 14 40 36 29 25 1 0) ,
instances/Z5/average/p_50_1_vrpnc9.txt	1: (0 8 38 42 7 23 39 6 11 18 1 0) ,2: (0 14 16 49 45 9 24 5 40 47 2 3 37 28 0) ,3: (0 20 34 4 22 10 15 17 35 12 50 29 19 0) ,4: (0 41 25 36 46 33 26 30 13 21 0) ,

instances/Z5/average/p_50_2_vrpnc10.txt	1: (0 1 2 24 40 43 36 7 16 39 32 50 15 22 0) ,2: (0 4 30 49 45 28 48 27 5 29 38 0) ,3: (0 8 35 14 47 21 31 42 18 26 6 11 0) ,4: (0 19 3 23 10 12 33 25 44 34 9 46 17 41 20 0) ,
instances/Z5/average/p_50_2_vrpnc2.txt	1: (0 7 35 39 40 42 6 15 43 19 28 0) ,2: (0 17 38 10 5 34 48 23 27 0) ,3: (0 22 29 2 20 31 3 4 0) ,4: (0 25 11 49 24 37 9 47 1 0) ,5: (0 36 16 26 41 13 45 0) ,6: (0 50 44 12 8 21 14 0) ,
instances/Z5/average/p_50_2_vrpnc3.txt	1: (0 7 44 32 42 2 19 15 6 28 23 45 22 38 47 0) ,2: (0 18 16 3 39 21 4 11 26 8 24 20 31 41 27 0) ,3: (0 40 9 35 37 10 5 43 36 50 25 14 49 48 29 12 33 17 13 34 1 0) ,
instances/Z5/average/p_50_2_vrpnc4.txt	1: (0 43 9 40 49 16 47 3 36 27 6 11 32 12 0) ,2: (0 45 35 15 38 1 42 28 26 20 8 37 19 39 33 4 22 48 31 0) ,3: (0 18 46 25 29 50 5 14 30 34 44 17 0) ,
instances/Z5/average/p_50_2_vrpnc5.txt	1: (0 4 1 28 16 32 25 27 26 0) ,2: (0 18 30 8 10 9 50 39 37 6 23 40 12 22 0) ,3: (0 36 21 7 44 20 5 31 3 29 34 13 35 0) ,4: (0 42 45 19 43 14 15 48 24 49 11 47 38 0) ,
instances/Z5/average/p_50_2_vrpnc7.txt	1: (0 12 19 33 2 16 23 18 39 0) ,2: (0 24 4 13 6 32 50 10 0) ,3: (0 27 41 36 11 47 43 34 21 0) ,4: (0 28 17 1 5 35 7 25 0) ,5: (0 30 29 42 48 37 9 22 0) ,6: (0 40 46 3 44 14 31 0) ,7: (0 45 49 26 38 0) ,
instances/Z5/average/p_50_2_vrpnc8.txt	1: (0 19 1 44 21 33 35 14 48 0) ,2: (0 20 43 50 17 15 13 37 22 10 47 11 32 31 0) ,3: (0 30 29 3 42 36 9 5 26 2 40 0) ,4: (0 38 39 23 49 45 25 24 27 7 12 6 8 0) ,
instances/Z5/average/p_50_2_vrpnc9.txt	1: (0 15 39 48 45 13 7 0) ,2: (0 37 27 6 22 11 14 1 35 33 21 10 42 8 19 0) ,3: (0 43 24 30 44 50 40 36 18 16 3 2 17 41 0) ,4: (0 47 12 26 4 28 31 23 9 32 29 34 49 20 0) ,

6.8 Five zone high threshold

Table A: Computational Results

instance name	UB	LB tree	LB root	LB root+cuts	time limit [s]	num vehicles
instances/Z5/High/p_50_1_vrpnc10.txt	-1308.36	-1 297,94	-1 308,36	-1 308,36	3 600	4
instances/Z5/High/p_50_1_vrpnc11.txt	-3300.03	-3 294,91	-3 300,03	-3 300,03	3 600	3
instances/Z5/High/p_50_1_vrpnc12.txt	-1549.38	-1 542,05	-1 549,38	-1 549,38	3 600	4
instances/Z5/High/p_50_1_vrpnc13.txt	-3593.6	-3 584,81	-3 593,6	-3 593,6	21 600	4
instances/Z5/High/p_50_1_vrpnc14.txt	-1551.64	-1 543,81	-1 551,64	-1 551,64	3 600	4
instances/Z5/High/p_50_1_vrpnc2.txt	-1321.59	-1 312,48	-1 321,59	-1 321,59	3 600	6
instances/Z5/High/p_50_1_vrpnc3.txt	-1285.41	-1 271,99	-1 285,41	-1 285,41	3 600	4
instances/Z5/High/p_50_1_vrpnc4.txt	-1418.65	-1 405	-1 418,65	-1 418,65	21 600	4
instances/Z5/High/p_50_1_vrpnc5.txt	-1283.54	-1 276,53	-1 283,54	-1 283,54	3 600	4
instances/Z5/High/p_50_1_vrpnc7.txt	-1165.66	-1 151,82	-1 165,66	-1 165,66	3 600	6
instances/Z5/High/p_50_1_vrpnc8.txt	-1379.37	-1 366,98	-1 379,37	-1 379,37	3 600	4
instances/Z5/High/p_50_1_vrpnc9.txt	-1339.21	-1 330,93	-1 339,21	-1 339,21	3 600	3
instances/Z5/High/p_50_2_vrpnc10.txt	-1302.68	-1 292,51	-1 302,68	-1 302,68	3 600	4
instances/Z5/High/p_50_2_vrpnc2.txt	-1269.35	-1 245,55	-1 269,35	-1 269,35	21 600	6
instances/Z5/High/p_50_2_vrpnc3.txt	-1434.91	-1 423	-1 434,91	-1 434,91	21 600	3
instances/Z5/High/p_50_2_vrpnc4.txt	-1240.15	-1 228	-1 240,15	-1 240,15	3 600	4
instances/Z5/High/p_50_2_vrpnc5.txt	-1298	-1 286,99	-1 298	-1 298	3 600	4
instances/Z5/High/p_50_2_vrpnc7.txt	-1218.12	-1 203,75	-1 218,12	-1 218,12	3 600	7
instances/Z5/High/p_50_2_vrpnc8.txt	-1320.03	-1 308,99	-1320.03	-1 320,03	3 600	4
instances/Z5/High/p_50_2_vrpnc9.txt	-1551.02	-1 536,6	-1 551,02	-1 551,02	3 600	4

Table B: Generated Routes

instance name	routes
instances/Z5/High/p_50_1_vrpnc10.txt	1: (0 1 4 2 49 40 32 46 33 14 36 31 26 0) ,2: (0 15 48 25 39 5 29 38 13 30 11 21 22 0) ,3: (0 41 45 16 37 27 42 6 3 9 47 50 17 0) ,4: (0 43 28 7 18 8 20 19 12 23 35 34 0) ,
instances/Z5/High/p_50_1_vrpnc11.txt	1: (0 16 9 44 12 50 39 48 6 40 24 28 37 23 26 15 33 29 18 0) ,2: (0 19 8 38 21 30 42 36 11 31 17 25 22 3 41 0) ,3: (0 47 32 14 45 20 10 35 4 49 43 1 13 5 0) ,
instances/Z5/High/p_50_1_vrpnc12.txt	1: (0 18 22 4 47 43 37 30 35 10 16 31 42 1 15 0) ,2: (0 21 3 5 41 27 7 25 48 23 19 44 0) ,3: (0 26 12 28 13 40 29 8 6 2 0) ,4: (0 45 34 11 39 9 20 36 33 17 49 0) ,
instances/Z5/High/p_50_1_vrpnc13.txt	1: (0 8 16 36 34 41 27 43 47 42 10 18 46 48 9 2 0) ,2: (0 25 13 26 6 31 7 37 50 39 49 12 40 30 29 11 32 33 21 20 44 35 0) ,3: (0 15 4 51) ,4: (0 1 23 17 22 38 5 19 3 24 28 51) ,
instances/Z5/High/p_50_1_vrpnc14.txt	1: (0 25 42 9 48 10 36 13 19 47 17 16 2 23 0) ,2: (0 39 44 21 38 33 31 5 37 22 11 0) ,3: (0 40 46 29 20 34 3 43 1 41 50 6 0) ,4: (0 45 15 27 8 30 12 28 14 26 7 4 24 0) ,
instances/Z5/High/p_50_1_vrpnc2.txt	1: (0 3 43 20 10 21 14 26 39 0) ,2: (0 6 37 25 4 27 28 13 44 0) ,3: (0 22 49 1 35 12 38 47 8 0) ,4: (0 23 9 2 30 42 17 11 41 0) ,5: (0 24 50 15 16 36 45 48 18 46 0) ,6: (0 32 31 34 7 40 0) ,
instances/Z5/High/p_50_1_vrpnc3.txt	1: (0 13 33 41 45 38 12 31 39 30 7 43 24 46 21 0) ,2: (0 36 6 37 25 8 34 14 10 29 0) ,3: (0 49 27 28 48 23 9 40 20 26 0) ,4: (0 50 44 42 5 1 18 15 3 22 16 2 4 32 35 17 11 47 19 0) ,
instances/Z5/High/p_50_1_vrpnc4.txt	1: (0 7 38 27 19 10 25 2 23 49 50 0) ,2: (0 16 13 4 45 8 21 6 3 39 17 32 0) ,3: (0 34 11 18 42 24 26 28 33 47 43 5 48 14 30 0) ,4: (0 12 29 36 46 41 40 9 22 1 15 31 44 37 20 0) ,
instances/Z5/High/p_50_1_vrpnc5.txt	1: (0 5 42 50 22 29 45 38 7 3 24 32 4 0) ,2: (0 19 31 1 14 16 46 18 44 39 35 41 49 43 15 0) ,3: (0 25 2 47 27 36 30 12 37 17 23 21 0) ,4: (0 48 34 40 11 20 6 0) ,
instances/Z5/High/p_50_1_vrpnc7.txt	1: (0 18 16 10 38 21 48 24 40 20 0) ,2: (0 23 19 17 43 22 8 0) ,3: (0 37 32 50 31 49 12 0) ,4: (0 41 47 11 27 34 13 35 39 0) ,5: (0 44 1 4 46 29 33 14 5 36 0) ,6: (0 45 30 2 3 26 42 0) ,
instances/Z5/High/p_50_1_vrpnc8.txt	1: (0 13 26 7 33 16 44 0) ,2: (0 21 20 39 49 4 9 12 6 11 42 5 23 35 15 0) ,3: (0 32 18 19 8 28 43 38 2 34 31 37 22 17 3 0) ,4: (0 50 48 45 47 24 30 27 40 14 29 25 1 46 0) ,
instances/Z5/High/p_50_1_vrpnc9.txt	1: (0 19 8 38 42 7 23 39 6 11 18 1 0) ,2: (0 20 34 29 50 12 35 17 15 10 22 4 45 49 16 28 0) ,3: (0 21 30 26 33 46 2 47 40 5 24 9 14 37 25 41 0) ,

instances/Z5/High/p_50_2_vrpnc10.txt	1: (0 8 26 18 42 21 47 14 35 23 10 3 19 0) ,2: (0 20 12 33 25 44 34 9 46 17 41 22 15 50 32 0) ,3: (0 38 29 27 48 28 45 49 30 4 11 0) ,4: (0 39 16 7 36 43 40 5 24 2 1 0) ,
instances/Z5/High/p_50_2_vrpnc2.txt	1: (0 47 9 37 24 49 48 23 11 0) ,2: (0 1 45 13 41 26 36 22 0) ,3: (0 50 27 44 34 33 5 10 38 0) ,4: (0 3 31 20 2 16 29 46 0) ,5: (0 28 19 18 43 15 6 42 40 39 21 0) ,6: (0 32 17 12 8 35 7 14 0) ,
instances/Z5/High/p_50_2_vrpnc3.txt	1: (0 47 46 23 28 6 15 19 2 42 32 44 10 37 35 40 0) ,2: (0 27 41 31 11 26 8 21 39 3 22 45 38 16 18 0) ,3: (0 1 33 13 17 20 24 12 29 48 49 14 25 50 36 43 5 7 0) ,
instances/Z5/High/p_50_2_vrpnc4.txt	1: (0 10 45 35 15 38 1 42 28 26 20 8 37 24 19 39 33 4 22 0) ,2: (0 12 21 17 11 44 6 27 36 32 23 0) ,3: (0 13 46 2 25 29 50 5 14 30 34 18 0) ,4: (0 41 31 48 40 49 16 47 3 9 43 0) ,
instances/Z5/High/p_50_2_vrpnc5.txt	1: (0 4 1 28 16 32 25 27 26 38 22 0) ,2: (0 18 30 8 10 9 50 39 37 6 23 40 12 17 0) ,3: (0 35 33 13 34 29 3 31 5 20 44 7 21 0) ,4: (0 47 41 11 49 24 48 15 14 43 19 45 42 0) ,
instances/Z5/High/p_50_2_vrpnc7.txt	1: (0 3 4 44 14 31 0) ,2: (0 10 50 32 6 13 1 24 17 0) ,3: (0 20 27 41 36 11 47 43 34 0) ,4: (0 21 39 18 23 16 2 33 19 12 0) ,5: (0 25 7 35 5 28 40 0) ,6: (0 30 29 42 48 37 9 22 0) ,7: (0 38 26 49 45 0) ,
instances/Z5/High/p_50_2_vrpnc8.txt	1: (0 18 3 42 36 9 5 26 2 40 0) ,2: (0 31 32 22 37 13 15 17 50 43 29 16 41 20 0) ,3: (0 34 8 6 11 47 10 46 7 12 27 24 25 45 23 39 0) ,4: (0 48 14 35 33 21 44 1 19 4 38 0) ,
instances/Z5/High/p_50_2_vrpnc9.txt	1: (0 15 39 48 45 13 20 12 47 7 0) ,2: (0 37 27 6 21 33 35 1 14 11 22 25 0) ,3: (0 43 24 30 44 50 40 36 18 16 3 2 17 41 0) ,4: (0 49 34 29 32 9 23 31 42 10 8 19 28 4 26 0) ,

6.9 Five zone low threshold

Table A: Computational Results

instance name	UB	LB tree	LB root	LB root+cuts	time [s]	num vehicles
instances/Z5/Low/p_50_1_vrpnc10.txt	-387.434	-387.992	-405.179	-405.179	2 339,17	4
instances/Z5/Low/p_50_1_vrpnc11.txt	-1 389,21	-1 389,92	-1 396,78	-1 396,78	566.958	3
instances/Z5/Low/p_50_1_vrpnc12.txt	-520.047	-520.999	-528.448	-528.448	285.178	4
instances/Z5/Low/p_50_1_vrpnc13.txt	-1 421,99	-1 477,39	-1 487,3	-1 487,3	21 600,9	3
instances/Z5/Low/p_50_1_vrpnc14.txt	-491.613	-491.759	-494.734	-494.734	14.6495	4
instances/Z5/Low/p_50_1_vrpnc2.txt	-339.53	-339.986	-348.865	-348.865	158.35	6
instances/Z5/Low/p_50_1_vrpnc3.txt	-383.023	-383.955	-391.99	-391.99	931.491	3
instances/Z5/Low/p_50_1_vrpnc4.txt	-435.836	-435.999	-452.188	-452.188	317.684	4
instances/Z5/Low/p_50_1_vrpnc5.txt	-420.034	-420.362	-430.712	-430.712	72.4545	4
instances/Z5/Low/p_50_1_vrpnc7.txt	-279.161	-279.656	-284.934	-284.934	9.8603	6
instances/Z5/Low/p_50_1_vrpnc8.txt	-457.837	-458.835	-468.799	-468.799	907.331	4
instances/Z5/Low/p_50_1_vrpnc9.txt	-391.798	-391.955	-403.208	-403.208	66.1017	4
instances/Z5/Low/p_50_2_vrpnc10.txt	-445.681	-451.203	-459.471	-459.471	3 886,06	4
instances/Z5/Low/p_50_2_vrpnc2.txt	-368.004	-368.59	-386.331	-386.331	1 091,01	6
instances/Z5/Low/p_50_2_vrpnc3.txt	-468.062	-468.402	-472.361	-472.361	118.968	3
instances/Z5/Low/p_50_2_vrpnc4.txt	-370.9	-370.997	-384.883	-384.883	4 690,75	3
instances/Z5/Low/p_50_2_vrpnc5.txt	-379.183	-379.71	-390.744	-390.744	232.955	4
instances/Z5/Low/p_50_2_vrpnc7.txt	-308.153	-308.503	-313.468	-313.468	30.8278	6
instances/Z5/Low/p_50_2_vrpnc8.txt	-408.665	-408.665	-415.277	-415.277	14.9774	4
instances/Z5/Low/p_50_2_vrpnc9.txt	-535.608	-535.999	-551.392	-551.392	3 236,14	4

Table B: Generated Routes

instance name	routes
instances/Z5/Low/p_50_1_vrpnc10.txt	1: (0 1 4 2 49 40 32 46 33 14 36 31 26 0) ,2: (0 15 48 25 39 5 29 38 13 30 11 22 21 0) ,3: (0 17 50 47 9 3 6 42 27 37 16 45 41 0) ,4: (0 34 35 23 12 19 20 8 18 7 28 43 0) ,
instances/Z5/Low/p_50_1_vrpnc11.txt	1: (0 9 8 38 21 30 42 36 19 11 31 17 25 22 3 41 0) ,2: (0 18 29 15 26 23 37 28 24 40 6 48 39 50 12 44 16 13 0) ,3: (0 47 32 14 20 45 35 4 49 43 1 0) ,
instances/Z5/Low/p_50_1_vrpnc12.txt	1: (0 26 12 28 13 40 29 8 6 2 0) ,2: (0 42 31 16 44 19 23 48 25 7 27 41 5 21 0) ,3: (0 49 17 33 36 20 9 39 11 34 45 0) ,4: (0 50 46 4 43 47 37 30 35 10 1 15 22 18 0) ,
instances/Z5/Low/p_50_1_vrpnc13.txt	1: (0 38 22 35 44 20 21 33 32 11 29 30 40 12 49 39 37 50 7 6 26 0) ,2: (0 8 16 36 34 41 27 43 47 42 10 18 46 48 2 9 0) ,3: (0 15 4 19 3 24 5 23 25 13 17 1 0) ,
instances/Z5/Low/p_50_1_vrpnc14.txt	1: (0 11 21 22 38 37 5 33 31 24 4 7 26 14 28 0) ,2: (0 32 45 27 8 12 30 42 0) ,3: (0 39 44 6 50 41 1 43 3 34 20 29 46 49 0) ,4: (0 40 23 2 16 17 47 19 13 36 10 48 9 25 0) ,
instances/Z5/Low/p_50_1_vrpnc2.txt	1: (0 3 43 20 10 21 14 26 39 0) ,2: (0 5 49 1 35 12 38 47 8 0) ,3: (0 9 2 11 17 36 45 48 18 46 0) ,4: (0 23 22 24 50 15 16 42 30 0) ,5: (0 32 33 31 34 7 25 0) ,6: (0 41 13 28 27 4 40 37 6 0) ,
instances/Z5/Low/p_50_1_vrpnc3.txt	1: (0 27 41 38 45 28 48 23 9 40 20 0) ,2: (0 36 10 14 34 8 25 37 6 5 1 18 15 42 44 0) ,3: (0 50 3 22 16 2 32 35 17 11 43 7 30 39 31 24 13 0) ,
instances/Z5/Low/p_50_1_vrpnc4.txt	1: (0 7 5 43 48 30 14 47 33 28 26 24 42 18 11 0) ,2: (0 17 39 3 44 31 15 1 22 9 40 41 46 36 29 12 0) ,3: (0 32 21 8 45 4 13 0) ,4: (0 49 23 2 25 20 37 10 19 27 38 0) ,
instances/Z5/Low/p_50_1_vrpnc5.txt	1: (0 4 32 24 3 7 38 45 29 22 50 42 5 0) ,2: (0 15 43 41 35 39 44 18 46 16 14 1 19 0) ,3: (0 21 17 37 12 30 36 27 47 2 25 13 0) ,4: (0 48 34 40 11 20 6 28 31 0) ,
instances/Z5/Low/p_50_1_vrpnc7.txt	1: (0 12 36 15 4 1 0) ,2: (0 18 22 43 8 23 0) ,3: (0 20 40 24 34 27 11 47 41 44 0) ,4: (0 39 35 13 46 29 33 14 5 49 0) ,5: (0 42 16 10 38 21 48 19 17 0) ,6: (0 45 31 50 26 2 30 37 0) ,
instances/Z5/Low/p_50_1_vrpnc8.txt	1: (0 1 25 27 14 40 36 29 10 46 0) ,2: (0 17 22 37 31 34 2 38 43 28 8 19 18 21 0) ,3: (0 20 39 49 4 9 12 6 11 42 5 23 35 15 0) ,4: (0 30 24 47 45 48 50 33 7 26 13 44 0) ,
instances/Z5/Low/p_50_1_vrpnc9.txt	1: (0 8 19 29 50 12 35 17 15 10 22 4 34 20 0) ,2: (0 14 16 49 45 9 24 5 40 47 2 3 37 28 0) ,3: (0 31 1 18 11 6 39 23 7 42 38 0) ,4: (0 44 41 25 36 46 33 26 30 13 21 0) ,

instances/Z5/Low/p_50_2_vrpnc10.txt	1: (0 1 39 16 7 36 43 40 5 24 2 29 38 0) ,2: (0 11 4 30 49 45 28 48 27 0) ,3: (0 19 3 10 23 35 14 47 21 31 42 18 26 8 0) ,4: (0 32 50 15 22 41 17 46 9 34 44 25 33 12 20 0) ,
instances/Z5/Low/p_50_2_vrpnc2.txt	1: (0 3 31 20 2 29 46 22 0) ,2: (0 14 21 8 12 17 32 0) ,3: (0 27 23 48 34 33 10 38 44 0) ,4: (0 28 19 18 43 15 6 42 40 39 35 0) ,5: (0 45 13 41 26 16 36 0) ,6: (0 47 9 37 24 49 11 1 0) ,
instances/Z5/Low/p_50_2_vrpnc3.txt	1: (0 18 16 38 39 3 22 45 23 28 42 2 19 6 15 46 0) ,2: (0 27 41 31 4 21 11 26 8 12 20 17 13 34 1 0) ,3: (0 33 29 48 49 14 25 50 36 43 5 32 44 7 10 37 35 40 0) ,
instances/Z5/Low/p_50_2_vrpnc4.txt	1: (0 43 9 40 49 16 47 3 36 27 6 11 32 0) ,2: (0 18 2 25 29 50 5 14 30 34 44 17 12 0) ,3: (0 35 15 38 1 42 28 26 8 37 24 19 39 33 4 48 31 0) ,
instances/Z5/Low/p_50_2_vrpnc5.txt	1: (0 2 12 40 23 6 37 39 50 10 9 30 18 42 0) ,2: (0 16 32 27 25 28 1 4 35 0) ,3: (0 33 13 34 29 3 31 5 20 44 7 21 36 0) ,4: (0 47 45 19 43 14 15 48 24 49 11 41 26 0) ,
instances/Z5/Low/p_50_2_vrpnc7.txt	1: (0 12 37 48 49 45 42 29 0) ,2: (0 15 10 50 32 44 3 0) ,3: (0 20 27 36 11 47 43 34 21 0) ,4: (0 28 17 24 4 13 6 14 31 0) ,5: (0 35 26 38 7 25 0) ,6: (0 39 18 23 16 2 33 19 9 22 0) ,
instances/Z5/Low/p_50_2_vrpnc8.txt	1: (0 32 11 46 47 10 22 37 13 15 17 50 43 20 0) ,2: (0 38 39 23 49 45 25 24 27 7 12 6 8 0) ,3: (0 40 2 26 5 9 36 42 3 29 16 41 30 0) ,4: (0 48 14 35 33 21 44 1 19 0) ,
instances/Z5/Low/p_50_2_vrpnc9.txt	1: (0 15 39 48 7 51) ,2: (0 12 26 4 28 31 23 9 32 29 34 49 20 13 0) ,3: (0 37 27 22 11 14 1 35 33 21 10 42 8 19 47 0) ,4: (0 5 24 44 50 40 36 18 16 3 2 17 41 43 0) ,

7 References

- Adelman, D. (2007). Dynamic bid prices in revenue management. *Operations Research*, 55(4), 647–661. <https://doi.org/10.1287/opre.1060.0368>
- Afsar, H. M., Afsar, S., & Palacios, J. J. (2021). Vehicle routing problem with zone-based pricing. *Transportation Research Part E: Logistics and Transportation Review*, 152, 102383. <https://doi.org/10.1016/j.tre.2021.102383>
- Alkaabneh, F., Diabat, A., & Gao, H. O. (2020). Benders decomposition for the inventory vehicle routing problem with perishable products and environmental costs. *Computers & Operations Research*, 113, 104751. <https://doi.org/10.1016/j.cor.2019.07.009>
- Aras, N., Aksen, D., & Tekin, M. T. (2011). Selective multi-depot vehicle routing problem with pricing. *Transportation Research Part C: Emerging Technologies*, 19(5), 866–884. <https://doi.org/10.1016/j.trc.2010.08.003>
- Baldacci, R., Christofides, N., & Mingozzi, A. (2008). An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, 115(2), 351–385. <https://doi.org/10.1007/s10107-007-0178-5>
- Baldacci, R., Mingozzi, A., & Roberti, R. (2011). New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5), 1269–1283. <https://doi.org/10.1287/opre.1110.0975>
- Beasley, J. E., & Christofides, N. (1989). An algorithm for the resource constrained shortest path problem. *Networks*, 19(4), 379–394. <https://doi.org/10.1002/net.3230190402>
- Bellman, R. E. (1958). On a routing problem. *Quarterly of Applied Mathematics*, 16, 87–90. <https://doi.org/10.1090/qam/102435>

- Braekers, K., Ramaekers, K., & Van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99, 300–313. <https://doi.org/10.1016/j.cie.2015.12.007>
- Bräysy, O., & Gendreau, M. (2005a). Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Science*, 39(1), 104–118. <https://doi.org/10.1287/trsc.1030.0056>
- Bräysy, O., & Gendreau, M. (2005b). Vehicle routing problem with time windows, Part II: Metaheuristics. *Transportation Science*, 39(1), 119–139. <https://doi.org/10.1287/trsc.1030.0057>
- Castaño, F., Rossi, A., Sevaux, M., & Velasco, N. (2014). A column generation approach to extend lifetime in wireless sensor networks with coverage and connectivity constraints. *Computers & Operations Research*, 52, 220–230. <https://doi.org/10.1016/j.cor.2013.11.001>
- Cen, X., Lo, H. K., Li, L., & Lee, E. (2018). Modelling electric vehicles adoption for urban commute trips. *Transportation Research Part B: Methodological*, 117, 431–454. <https://doi.org/10.1016/j.trb.2018.09.003>
- Cen, X., Yang, X., Ren, K., Liu, W., & Lee, E. (2025). Optimal pricing and vehicle routing of vehicle-to-vehicle charging platform with time windows. *Transportation Research Part C: Emerging Technologies*, 180, 105319. <https://doi.org/10.1016/j.trc.2025.105319>
- Ceselli, A., Righini, G., & Salani, M. (2009). A column generation algorithm for a rich vehicle-routing problem. *Transportation Science*, 43(1), 56–69. <https://doi.org/10.1287/trsc.1080.0256>
- Chabrier, A. (2006). Vehicle routing problem with elementary shortest path based column generation. *Computers & Operations Research*, 33(10), 2972–2990. <https://doi.org/10.1016/j.cor.2005.02.029>

- Cherkassky, B. V., Goldberg, A. V., & Radzik, T. (1996). Shortest paths algorithms: Theory and experimental evaluation. *Mathematical Programming*, 73(2), 129–174.
<https://doi.org/10.1007/BF02592101>
- Christian, R. C. (1965). Physical distribution: Key to improved volume and profits. *Journal of Marketing*, 29, 65–70.
- Clausen, J. (1999). Branch-and-bound algorithms: Principles and examples [Technical report]. Department of Computer Science, University of Copenhagen.
- Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y., & Semet, F. (2002). A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, 53, 512–522.
<https://doi.org/10.1057/palgrave.jors.2601319>
- Costa, L., Contardo, C., & Desaulniers, G. (2019). Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Science*, 53(4), 946–985.
<https://doi.org/10.1287/trsc.2018.0878>
- Dantzig, G. B. (1957). Discrete-variable extremum problems. *Operations Research*, 5(2), 266–277. <https://doi.org/10.1287/opre.5.2.266>
- Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1), 80–91. <https://doi.org/10.1287/mnsc.6.1.80>
- Desaulniers, G. (2010). Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. *Operations Research*, 58(1), 179–192.
<https://doi.org/10.1287/opre.1080.0713>
- Desaulniers, G., Desrosiers, J., Ioachim, I., Solomon, M. M., Soumis, F., & Villeneuve, D. (1998). A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In T. G. Crainic & G. Laporte (Eds.), *Fleet management and logistics* (pp. 57–93). Springer.

- Dial, R. B. (1969). Algorithm 360: Shortest-path forest with topological ordering. *Communications of the ACM*, 12(11), 632–633.
<https://doi.org/10.1145/363269.363610>
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269–271. <https://doi.org/10.1007/BF01386390>
- Duman, E. N., Tas, D., & Çatay, B. (2022). Branch-and-price-and-cut methods for the electric vehicle routing problem with time windows. *International Journal of Production Research*, 60(17), 5332–5353. <https://doi.org/10.1080/00207543.2021.1955995>
- Eksioglu, B., Vural, A. V., & Reisman, A. (2009). The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57(4), 1472–1483.
<https://doi.org/10.1016/j.cie.2009.05.009>
- Etebari, F., & Dabiri, N. (2016). A hybrid heuristic for the inventory routing problem under dynamic regional pricing. *Computers & Chemical Engineering*, 95, 231–239.
<https://doi.org/10.1016/j.compchemeng.2016.09.018>
- Fachini, R. F., & Armentano, V. A. (2020). Logic-based Benders decomposition for the heterogeneous fixed fleet vehicle routing problem with time windows. *Computers & Industrial Engineering*, 148, 106641. <https://doi.org/10.1016/j.cie.2020.106641>
- Feillet, D., Dejax, P., Gendreau, M., & Gueguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44(3), 216–229.
<https://doi.org/10.1002/net.20033>
- Gendreau, M., & Tarantilis, C. D. (2010). Solving large-scale vehicle routing problems with time windows: The state-of-the-art (CIRRELT-2010-04). Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation.

- Griffis, S. E., Bell, J. E., & Closs, D. J. (2012). Metaheuristics in logistics and supply chain management. *Journal of Business Logistics*, 33(2), 90–106.
<https://doi.org/10.1111/j.0000-0000.2012.01042.x>
- Guo, J., Long, J., Xu, X., Yu, M., & Yuan, K. (2022). The vehicle routing problem of intercity ride-sharing between two cities. *Transportation Research Part B: Methodological*, 158, 113–139. <https://doi.org/10.1016/j.trb.2022.02.013>
- Hadjar, A., Marcotte, P., & Soumis, F. (2006). A branch-and-cut algorithm for the multiple depot vehicle scheduling problem. *Operations Research*, 54(1), 130–149.
<https://doi.org/10.1287/opre.1050.0240>
- Irnich, S. (2008). Resource extension functions: Properties, inversion and generalization to segments. *OR Spectrum*, 30(1), 113–148. <https://doi.org/10.1007/s00291-007-0083-6>
- Irnich, S., & Villeneuve, D. (2006). The shortest-path problem with resource constraints and k-cycle elimination for $k \geq 3$. *INFORMS Journal on Computing*, 18(3), 391–406.
<https://doi.org/10.1287/ijoc.1040.0117>
- Jia, H., Li, Y., Dong, B., & Ya, H. (2013). An improved tabu search approach to the vehicle routing problem. *Procedia - Social and Behavioral Sciences*, 96, 1208–1217.
<https://doi.org/10.1016/j.sbspro.2013.08.138>
- Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: Past, present, and future. *Multimedia Tools and Applications*, 80(5), 8091–8126.
<https://doi.org/10.1007/s11042-020-10139-6>
- Konstantakopoulos, G. D., Gayialis, S. P., & Kechagias, E. P. (2022). Vehicle routing problem and related algorithms for logistics distribution: A literature review and classification. *Operational Research*, 22(3), 2033–2062. <https://doi.org/10.1007/s12351-020-00600-7>
- Kuo, Y. (2010). Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. *Computers & Industrial Engineering*, 59(1), 157–165. <https://doi.org/10.1016/j.cie.2010.03.012>

- Landahl, H. D. (1947). A matrix calculus for neural nets: II. *Bulletin of Mathematical Biophysics*, 9, 99–108.
- Landahl, H. D., & Runge, R. (1946). Outline of a matrix algebra for neural nets. *Bulletin of Mathematical Biophysics*, 8, 75–81.
- Liu, S. C., & Chen, J. R. (2011). A heuristic method for the inventory routing and pricing problem in a supply chain. *Expert Systems with Applications*, 38(3), 1447–1456.
<https://doi.org/10.1016/j.eswa.2010.07.051>
- Luce, R. D., & Perry, A. D. (1949). A method of matrix analysis of group structure. *Psychometrika*, 14, 95–116.
- Maher, S. J. (2016). Solving the integrated airline recovery problem using column-and-row generation. *Transportation Science*, 50(1), 216–239.
<https://doi.org/10.1287/trsc.2014.0552>
- Mirjalili, S. (2019). Genetic algorithm. In *Evolutionary algorithms and neural networks: Theory and applications* (pp. 43–55). Springer. https://doi.org/10.1007/978-3-319-93025-1_4
- Orden, A. (1956). The transshipment problem. *Management Science*, 2(3), 276–285.
<https://doi.org/10.1287/mnsc.2.3.276>
- Pecin, D., Pessoa, A., Poggi, M., & Uchoa, E. (2017). Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation*, 9, 61–100.
<https://doi.org/10.1007/s12532-016-0108-8>
- Rezgui, D., Siala, J. C., Aggoune-Mtalaa, W., & Bouziri, H. (2019). Application of a variable neighborhood search algorithm to a fleet size and mix vehicle routing problem with electric modular vehicles. *Computers & Industrial Engineering*, 130, 537–550.
<https://doi.org/10.1016/j.cie.2019.03.001>

- Righini, G., & Salani, M. (2008). New dynamic programming algorithms for the resource-constrained elementary shortest path problem. *Networks*, 51(3), 155–170.
<https://doi.org/10.1002/net.20142>
- Sadeghi Dastaki, M., Setak, M., & Karimi, H. (2021). The multi-zone location–routing problem with pricing: A flow-based formulation and two heuristic approaches. *Soft Computing*, 25(1), 741–769. <https://doi.org/10.1007/s00500-020-05186-y>
- Sarasola, B., Doerner, K. F., Schmid, V., & Alba, E. (2016). Variable neighborhood search for the stochastic and dynamic vehicle routing problem. *Annals of Operations Research*, 236, 425–461. <https://doi.org/10.1007/s10479-014-1689-2>
- Savelsbergh, M. W. P., & Sol, M. (1995). The general pickup and delivery problem. *Transportation Science*, 29(1), 17–29. <https://doi.org/10.1287/trsc.29.1.17>
- Schrijver, A. (2012). On the history of the shortest path problem. *Documenta Mathematica*, Extra Volume ISMP, 155–167.
- Shapiro, J. F. (2001). *Modeling the supply chain*. Thomson Learning.
- Shi, Y., Liu, W., & Zhou, Y. (2023). An adaptive large neighborhood search based approach for the vehicle routing problem with zone-based pricing. *Engineering Applications of Artificial Intelligence*, 124, 106506. <https://doi.org/10.1016/j.engappai.2023.106506>
- Solomon, M. M. (1983). Vehicle routing and scheduling with time window constraints: Models and algorithms [Doctoral dissertation, University of Pennsylvania].
- Theurich, F., Fischer, A., & Scheithauer, G. (2021). A branch-and-bound approach for a vehicle routing problem with customer costs. *EURO Journal on Computational Optimization*, 9, 100003. <https://doi.org/10.1016/j.ejco.2020.100003>
- Tilk, C., Rothenbächer, A.-K., Gschwind, T., & Irnich, S. (2017). Asymmetry matters: Dynamic half-way points in bidirectional labeling for solving shortest path problems with resource constraints faster. *European Journal of Operational Research*, 261(2), 530–539. <https://doi.org/10.1016/j.ejor.2017.03.017>

- Toth, P., & Vigo, D. (2002). Branch-and-bound algorithms for the capacitated VRP. In P. Toth & D. Vigo (Eds.), *The vehicle routing problem* (pp. 29–51). Society for Industrial and Applied Mathematics. <https://doi.org/10.1137/1.9780898718515.ch2>
- Van Breedam, A. (1995). Improvement heuristics for the vehicle routing problem based on simulated annealing. *European Journal of Operational Research*, 86(3), 480–490. [https://doi.org/10.1016/0377-2217\(94\)00064-J](https://doi.org/10.1016/0377-2217(94)00064-J)