



**Politecnico
di Torino**

Politecnico di Torino

Master's Degree in Mathematical Engineering

A.Y. 2025/2026

Graduation Session July 2026

A Cost Function Approximation Approach to the Dynamic Stochastic Vehicle Routing Problem

Supervisor:

Paolo Brandimarte

Candidate:

Fabio Lauro

Abstract

The recent boom of online shopping caused a reshaping of the logistics industry: demand skyrocketed and new business models emerged, along with stronger concerns about environmental and energy impact. Therefore, the need for better service, possibly obtained through algorithmic decision-making, is as strong as ever. The Vehicle Routing Problem (VRP) deals with the optimization of service to clients characterized by their location through a fleet of trucks. In Dynamic Stochastic VRPs, the problem is defined in a sequence of time periods in which new information is revealed dynamically (like new clients, for example), and there are stochastic variables representing uncertainty on location or demand, among other things. These problems are usually very hard to solve, due to their dimensionality, the need to account for uncertainty, the variety of constraints and the interaction between decision variables. Although scientific research has focused on representing uncertainty through complicated stochastic programs that are somewhat close to reality but almost never solvable in real-world applications, industry practitioners typically solve deterministic look-ahead models that account for the future outcome of random variables through a series of approximations that make the model manageable, at the cost of reducing its accuracy. The Cost Function Approximation (CFA) approach bridges the gap between the two frameworks by introducing parametric terms that are meant to account for the impact of stochastic variables in the problem, resulting in a solvable and less myopic model. These parametric terms relate to features that are inherent to the problem: their fine-tuning is done by solving the CFA formulation in a stochastic simulator, therefore shifting the complexity of the problem from the model solution to algorithmic, offline testing. The relative importance of these parameters allows to gain knowledge on the most impactful features for the problem at hand. This thesis uses this framework to develop on the paper "The dynamic multiperiod vehicle routing problem with probabilistic information" by Albareda et al. ([1]), where the parametric term takes the form of prizes to be assigned to clients, that are informative on the convenience of service at the current day with respect to later time periods. In particular, the probability of observing future requests from neighboring clients, that allows to group close clients together, is leveraged, resulting in more optimized routing

decisions. Therefore, in each time period we solve a Prize-Collecting VRP with service times, instead of a usual VRP. This approach is developed on synthetic and real data, belonging to Mondo Convenienza, one of Italy's largest furniture company, in the time-horizon of the working days of July 2025. The results are compared with heuristics to demonstrate the effectiveness of this approach. For the full code of this thesis: <https://github.com/fabiolauro/Cost-Function-Approximation-for-a-Dynamic-Stochastic-Vehicle-Routing-Problem>

*

Table of Contents

List of Tables	VIII
List of Figures	X
1 Introduction	1
2 Literature review	5
3 Theoretical background	7
3.1 Stochastic Optimization Problems	7
3.2 Exploiting the Value Function	8
3.3 Cost Function Approximation	9
4 Problem formulation	11
4.1 Instance generation	12
4.1.1 Synthetic data	12
4.1.2 Mondo Convenienza’s data	14
4.2 Policies	15
4.2.1 First In First Out	16
4.2.2 Earliest Due Date	16
4.2.3 Prize-Collecting policies	16
4.3 Daily subproblem formulation	17
4.3.1 FIFO and EDD: Vehicle Routing Problem with service times	17
4.3.2 Cost function approximation: Prize-Collecting VRP	18
5 Implementation details	21
5.1 Simulators	21
5.2 Solvers	22
5.3 Data handlers	22

6	Computational experiments	25
6.1	Synthetic data	25
6.1.1	Fixed time windows' lengths	26
6.1.2	Variable time windows' length	36
6.2	Mondo Convenienza's data	47
6.2.1	Catanzaro depot	47
6.2.2	Lainate depot	54
7	Conclusions	61
	Bibliography	65

List of Tables

6.1	Performance comparison in the circular set, fixed windows, variable score term.	28
6.2	Performance comparison in the globular set, fixed windows, variable score term.	28
6.3	Performance comparison in the rectangular set, fixed windows, variable score term.	29
6.4	Performance comparison in the circular set, fixed windows, variable probability term.	30
6.5	Performance comparison in the globular set, fixed windows, variable probability term.	31
6.6	Performance comparison in the rectangular set, fixed windows, variable probability term.	31
6.7	Performance comparison in the circular set, fixed windows, variable time term.	33
6.8	Performance comparison in the globular set, fixed windows, variable time term.	33
6.9	Performance comparison in the rectangular set, fixed windows, variable time term.	34
6.10	Performance comparison in the circular set, fixed windows, variable big M term.	34
6.11	Performance comparison in the globular set, fixed windows, variable big M term.	36
6.12	Performance comparison in the rectangular set, fixed windows, variable big M term.	36
6.13	Performance comparison in the circular set, variable windows, variable score term.	37
6.14	Performance comparison in the globular set, variable windows, variable score term.	39
6.15	Performance comparison in the rectangular set, variable windows, variable score term.	39

6.16	Performance comparison in the circular set, variable windows, variable probability term.	41
6.17	Performance comparison in the globular set, variable windows, variable probability term.	41
6.18	Performance comparison in the rectangular set, variable windows, variable probability term.	42
6.19	Performance comparison in the circular set, variable windows, variable time term.	42
6.20	Performance comparison in the globular set, variable windows, variable time term.	44
6.21	Performance comparison in the rectangular set, variable windows, variable time term.	44
6.22	Performance comparison in the circular set, variable windows, variable big M term.	46
6.23	Performance comparison in the globular set, variable windows, variable big M term.	46
6.24	Performance comparison in the rectangular set, variable windows, variable big M term.	47
6.25	Performance comparison in the Catanzaro set, fixed windows, variable score term.	48
6.26	Performance comparison in the Catanzaro set, fixed windows, variable probability term.	51
6.27	Performance comparison in the Catanzaro set, fixed windows, variable time term.	51
6.28	Performance comparison in the Catanzaro set, fixed windows, variable big M term.	54
6.29	Performance comparison in the Lainate set, fixed windows, variable score term.	55
6.30	Performance comparison in the Lainate set, fixed windows, variable probability term.	55
6.31	Performance comparison in the Lainate set, fixed windows, variable time term.	59
6.32	Performance comparison in the Lainate set, fixed windows, variable big M term.	59

List of Figures

1.1	Visualization of a vehicle routing problem.	2
4.1	Synthetic datasets.	13
4.2	Catanzaro dataset and geographic area.	15
4.3	Lainate dataset and geographic area.	15
4.4	Visualization of a prize-collecting vehicle routing problem.	19
6.1	Performance analysis of the ADTV policy with variable Score term under fixed windows , on synthetic instances. Dashed lines represent the FIFO baseline, while dotted lines represent the MLTP strategy.	27
6.2	Performance analysis of the ADTV policy with variable Probability term under fixed windows , on synthetic instances. The FIFO baseline is denoted by dashed lines, while the MLTP strategy is represented by dotted lines.	30
6.3	Performance analysis of the ADTV policy with variable Time term under fixed windows , on synthetic instances. Dashed lines represent the FIFO baseline, while dotted lines denote the MLTP standard.	32
6.4	Performance analysis of the ADTV and MLTP policies over variable Big M values under fixed windows , on synthetic instances. Solid lines represent the proposed ADTV policy, dashed lines indicate the MLTP strategy, and dotted lines establish the FIFO baseline.	35
6.5	Performance analysis of the ADTV policy with variable Score term under variable time windows , on synthetic instances. The horizontal baselines represent the FIFO (dashed), EDD (dash-dot), and MLTP (dotted) heuristics.	38
6.6	Performance analysis of the ADTV policy with variable Probability term under variable time windows , on synthetic instances. The horizontal baselines represent the FIFO (dashed), EDD (dash-dot), and MLTP (dotted) strategy.	40

6.7	Performance analysis of the ADTV policy with a variable Time term under variable time windows , tested on synthetic instances. The horizontal baselines correspond to the FIFO (dashed line), EDD (dash-dot line), and standard MLTP (dotted line) reference heuristics.	43
6.8	Performance analysis comparing ADTV and MLTP policies over variable Big M penalties under variable time windows , on synthetic instances. Solid lines represent ADTV, while dashed lines indicate MLTP. Horizontal baselines are provided for FIFO (dotted) and EDD (dash-dot) heuristics.	45
6.9	Performance evaluation on the real-world Catanzaro dataset varying the Score term. The solid blue line represents the ADTV policy. Baselines for FIFO and standard MLTP are reported.	49
6.10	Performance evaluation on the real-world Catanzaro dataset varying the Probability term. The solid blue line represents the ADTV policy. Baselines for FIFO and standard MLTP are reported.	50
6.11	Performance evaluation on the real-world Catanzaro dataset varying the Time term. The solid blue line represents the ADTV policy. Baselines for FIFO and standard MLTP are reported.	52
6.12	Performance comparison between ADTV and MLTP policies over variable Big M penalties on the Catanzaro dataset. Solid dark blue lines represent the proposed ADTV policy, while dashed crimson lines indicate the MLTP standard. The horizontal line establishes the FIFO baseline.	53
6.13	Performance analysis on the real-world Lainate dataset varying the Score term. The deep teal solid line denotes the proposed ADTV policy. Baselines for FIFO (indigo) and standard MLTP (orange) are represented by horizontal dashed and dotted lines, respectively.	56
6.14	Performance analysis on the Lainate dataset varying the Probability term. The ADTV policy is represented by the deep teal line. Horizontal reference lines indicate the performance of the FIFO (indigo) and standard MLTP (orange) heuristics.	57
6.15	Performance analysis on the Lainate dataset varying the Time term. The ADTV policy is represented by the deep teal line. Horizontal reference lines indicate the performance of the FIFO (indigo) and standard MLTP (orange) heuristics.	58
6.16	Performance comparison between ADTV and MLTP policies over variable Big M penalties on the Lainate dataset. Solid deep teal lines represent the proposed ADTV policy, while dashed orange lines indicate the MLTP standard. The horizontal indigo line establishes the FIFO baseline.	60

Chapter 1

Introduction

The Vehicle Routing Problem (VRP) is a historical operations research framework, first tackled by Dantzig and Ramser in 1959 as a generalization of the Traveling Salesmen Problem. A fleet of vehicles needs to visit a series of locations, satisfying various constraints, be it meeting demand without violating capacity, or arriving to locations within given time-windows. Its appeal derives from its ubiquitous and ever-transforming nature: this framework can be adapted and applied to many companies or entities with significantly different needs. Food delivery companies must take into account the food preparation time; if orders are accepted in real time, routes can change to accommodate more customers; other companies may deliver goods that must be stored in specific ways, posing significant limits to capacity usage.

From a historical perspective, the boom of online shopping and the internet have been responsible for a dramatic surge in demand for logistic services, a radical reshaping of the industry, and skyrocketed the importance of logistics in the world economy. In addition, new business models emerged, along with stronger concerns about fuel emissions and environmental impact and a more urgent need for energy efficiency. The optimization of logistic services is therefore a crucial problem in today's economy, and new ways of solving VRPs are needed to tackle different needs.

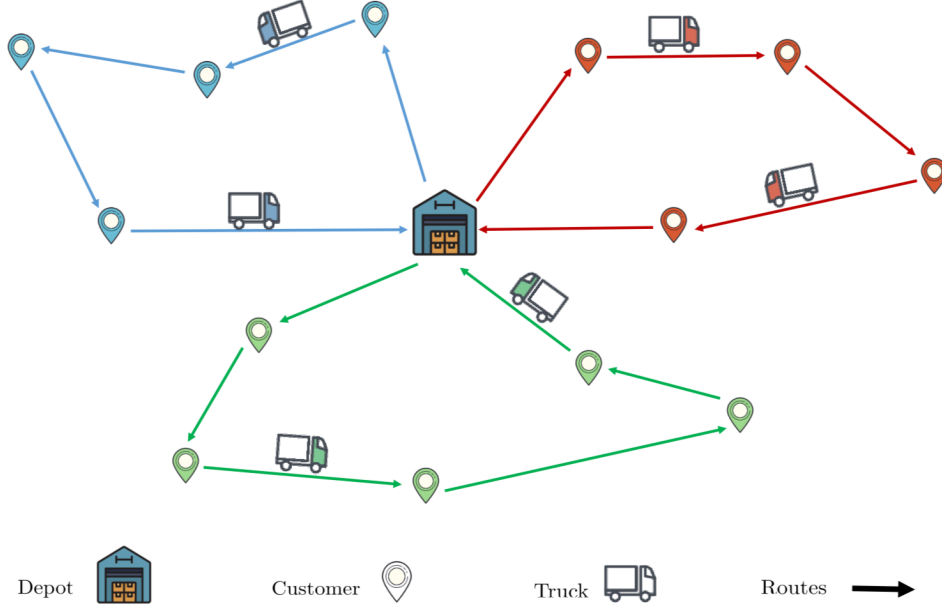


Figure 1.1: Visualization of a vehicle routing problem.

Given the wide variety of constraints that can appear and the dimensionality of the problem, where it is not unusual to have hundreds of customers, VRPs are usually solved through heuristics, which can be simple decision rules or algorithmic approaches to approximate a problem solution. This is even more necessary when multiple time periods and stochasticity come into play: we may have to serve clients over a weekly or monthly time-horizon, with uncertainty on location, time-windows or demand, to name a few possibilities.

This thesis borrows from Mondo Convenienza’s context to tackle a Dynamic Stochastic Multi-period Vehicle Routing Problem. We optimize service over a month, with unknown clients and locations that are revealed as days roll on, stochastic service times, and both fixed and random time-windows. The problem is very high dimensional, which makes it nearly impossible to use Dynamic Programming techniques. To solve the problem, we adopt a Cost Function Approximation approach, framing each daily subproblem as a Prize-Collecting Vehicle Routing Problem (PCVRP), where each client is linked to a prize that signals the convenience of service. The aim is therefore to maximize the collected prize minus the traveled distance. This means that, contrary to standard VRPs, clients may not be served and can be postponed to later time-periods.

This thesis uses this framework to leverage probabilistic information of new client orders in different areas, updated after each day, to improve the efficiency of service. If we expect future orders in a certain area, clients may be postponed to serve them altogether, improving efficiency and costs. This opportunity is reflected in their

daily prize, along with other characteristics such as urgency. In this sense, we test the approach of Albareda et al. [1] in a context closer to real-world operations. In fact, in their case, clients are fixed and probabilistic information on service requests is given. Moreover, we consider service times instead of demand, thus adding a further layer of complexity, since the total available time also interacts with the distances between clients through travel times.

Finally, the functional form of prizes we propose gives us insight on the most important factors to be considered when making routing decisions. We obtain these findings by parameter fine-tuning within the cost function approximation framework. To analyze the behavior of the algorithm in different contexts, we use both Mondo Convenienza’s real data and synthetic data, comparing the results of the PCVRPs to different heuristics.

Chapter 2

Literature review

The problem formulation that we face in this thesis is close in nature to what in the literature is called the dynamic multiperiod vehicle routing problem (DMVRP), in which service over a time-horizon must be optimized under a diverse set of constraints, and orders are revealed dynamically as clients become serviceable.

A key contribution in this area was made by Wen et al. [2], who modeled the case of a large retailer where customers place orders dynamically, specifying a set of consecutive days on which delivery is acceptable. To address this problem, the authors propose a rolling horizon approach supported by a three-stage heuristic. They solve a deterministic look-ahead model by optimizing service between time t and $t + \tau$. Clients are first selected based on a spatio-temporal correlation index: if a client has many close clients with overlapping time-windows, their index is higher. After the clients have been selected, the routes are formed for the time horizon, and in the final step the routes at time t are post-optimized. The objective function minimizes a linear combination of travel time, customer waiting time and trucks' workload. Note that decisions in this model are made using only information on orders already placed, without formally anticipating future demand.

Albareda et al. [1] expand on this work by introducing probabilistic information. Customers are fixed and known, and can order again after an inter-arrival exponential time with mean 3, that starts when their order is completed. They are characterized by uniformly distributed time-windows of length between 1 and 3. To optimize service, a neighborhood is defined based on the relative savings of serving two customers together with respect to serving them in different routes. This index is defined for every pair of customers, and if it surpasses a certain threshold they are set as neighbors. The arrival probabilities of having orders in the neighborhood in subsequent time periods is exploited by inserting them into prizes assigned to each client, defined as a signal on the convenience of serving customers now rather than postponing them. Therefore the daily sub-problem is a Prize-Collecting VRP: the objective function maximizes the total collected

prize minus the travel costs. They used Solomon instances from Solomon [3] to generate the data, considering a horizon of 6 days. They compared the performance with an Early Policy (clients are served in the first time window) and a Delayed Policy (clients are served in the last time window). Moreover, they consider vehicle capacity and customer demand. Their approach beats the heuristics for almost any Solomon instance, with different values depending on the neighborhood threshold value. Finally, they generate several instances from the Wen et al. [2] paper data and prove that their approach improves on the heuristics also in that case. This framework is already a cost function approximation, although it is not explicitly stated as such.

The CFA approach was first formalized by Perkins III and Powell [4] as a framework used in industrial contexts often overlooked by operations research scholars. The real-world, deterministic look-ahead approaches and the stochastic programming ones are compared, examining strengths and weaknesses. Finally, the CFA is introduced along with a gradient-based stochastic search algorithm to optimize the parametric terms, and it is tested in an example of an energy storage application.

Developing this theoretical framework further, Powell and Ghadimi [5] define CFAs as a single stage policy search approach, which can also be exploited in policies based on look-ahead approximations, thus concerning multi-stage decision problems. Examples of such approach are given, and the energy storage application of the previous paper is presented again with emphasis on rolling forecasts modeling. Forecast modeling is the focus of the subsequent paper from Ghadimi and Powell [6], where they propose a stochastic search algorithm for parameters optimization. Instead of relying on exact computation of stochastic sub-gradients as in [4], which can be done under severe and often unrealistic assumptions, they propose the Stochastic Averaging Numerical Gradient algorithm, with numerical derivatives based on stochastic noise.

This work develops Albareda’s idea of segmenting the overall problem in daily Prize-Collecting VRPs while leveraging the CFA approach of parameter fine-tuning. In our example, arrival probabilities are estimated rather than given, and we consider service times rather than vehicle capacity, which causes complex interactions with travel times and customer sequencing. Uncertainty lies in clients’ location, service time and time-window. While in [1] there is a unique parameter used for fine tuning, we introduce an additive functional form for prizes that underlines the effects of terms regarding urgency, cluster concentration and a score term that signals priority to bigger orders.

Chapter 3

Theoretical background

When solving a stochastic optimization problem, one wants to make decisions at each time period that not only minimize the current cost, but also take into consideration the future evolution of the system, avoiding myopic actions. Industry practitioners usually deal with the intrinsic uncertainty of a stochastic optimization problem by defining it as a deterministic look-ahead model, where uncertainty is incorporated as given data: this can be, for example, the weather forecast in an energy planning problem. This kind of approach has not been explored thoroughly in the literature because scholars tend to dismiss it claiming it does not appropriately consider stochasticity.

On the other hand, stochastic programs, which are meant to give a better representation of the problem, are very hard to solve, since they usually involve complicated multidimensional expectations, using large scenario trees to approximate potential future events. Even when solving the model is possible, they require significant assumptions to simplify the problem.

The Cost Function Approximation (CFA) approach introduces a parametric term in a deterministic look-ahead model that accounts for the downstream impact of uncertainty, to be fine-tuned offline using a stochastic simulator. In this way, it is possible to incorporate the problem's structure in the parameters, leveraging domain-dependent knowledge while maintaining the model computationally tractable. This can be, for example, extra-reserves of energy to manage demand peaks in the unit commitment problem, or schedule slack in routing operations.

3.1 Stochastic Optimization Problems

Sequential stochastic decision problems are modeled across five dimensions: the state variable S_t , the decision variable x_t , the exogenous information W_t , the transition function and the objective function.

At any time t , the state S_t of the system encapsulates all the information needed to compute the cost function, the constraints, and the transition function, from time t onward. S_t can contain information on the location of clients, among other things. A decision $x_t \in \mathcal{X}_t$, where $\mathcal{X}_t$ is the set of feasible actions at time t , is made according to a policy function $X^\pi(S_t)$, depending on the state of the system. After making a decision, new exogenous information W_{t+1} (such as new customer requests) is revealed, which may be affected by our actions. The system then evolves to the next state according to the transition function:

$$S_{t+1} = S^M(S_t, x_t, W_{t+1})$$

The ultimate goal of the stochastic optimization problem is to find the policy π that minimizes the expected sum of costs over a finite horizon T :

$$\min_{\pi \in \Pi} \mathbb{E}^\pi \left\{ \sum_{t=0}^T C_t(S_t, X_t^\pi(S_t)) \mid S_0 \right\}$$

. We write $\mathbb{E}^\pi(\cdot)$ since the exogenous information (also known as risk factors) can change based on our policy. This formulation, together with the state transition function and the exogenous information process is known as the base model.

3.2 Exploiting the Value Function

The exact optimal policy $X_t^*(S_t)$ is defined as the choice that minimizes the sum of the immediate cost and the expected downstream impact, assuming that from time $t + 1$ onward optimal decisions are made according to all admissible policies:

$$X_t^*(S_t) = \arg \min_{x_t \in \mathcal{X}_t} \left(C_t(S_t, x_t) + \mathbb{E} \left\{ \min_{\pi \in \Pi} \mathbb{E} \left\{ \sum_{t'=t+1}^T C_{t'}(S_{t'}, X_{t'}^\pi(S_{t'})) \mid S_{t+1} \right\} \mid S_t, x_t \right\} \right)$$

This can be reformulated by defining the value function $V_{t+1}(S_{t+1})$, which identifies the value of being in state S_{t+1} at time $t + 1$ and making optimal decisions down the line, as

$$V_{t+1}(S_{t+1}) = \min_{\pi \in \Pi} \mathbb{E} \left\{ \sum_{t'=t+1}^T C_{t'}(S_{t'}, X_{t'}^\pi(S_{t'})) \mid S_{t+1} \right\}$$

So the previous equation can be re-written in the standard Bellmann's equation form:

$$V_t(S_t) = \min_{x_t \in \mathcal{X}_t} (C_t(S_t, x_t) + \mathbb{E} \{ V_{t+1}(S_{t+1}) \mid S_t, x_t \})$$

By solving this recursive equation, one can find the optimal decision path that minimizes the overall cost over the time horizon. This approach is commonly stated as Dynamic Programming. However, solving this problem exactly is computationally intractable for virtually all real-world applications due to the presence of the multidimensional expected value, the need to optimize over the space of admissible policies, and the dimensionality of the state and action spaces.

Moreover, defining a meaningful value function is challenging for most practical applications, and several strategies have been developed over the years, under the name of Value Function Approximation or Approximate Dynamic Programming. Even these approaches can fail in several contexts, like in challenging VRPs where one may have to optimize routing decision and timing sequences altogether.

As stated before, industry practitioners tend to circumvent these problems by changing the look-ahead model, which can be achieved by using deterministic approximations, limiting the time horizon, aggregating decision stages or discretization, among other strategies.

3.3 Cost Function Approximation

Parametric Cost Function Approximation bridges the gap between stochastic methods and deterministic models by inserting tunable parameters into a deterministic optimization model. Instead of modeling uncertainty explicitly through massive scenario trees, the CFA modifies the objective function and/or the constraint set to account for the stochastic nature of the problem. The parameterized policy can be expressed as

$$X_t^\pi(S_t|\theta) = \arg \min_{x_t \in \mathcal{X}_t^\pi(\theta)} \{C_t(S_t, x_t) + \sum_{f \in \mathcal{F}} \theta_f^c \phi_f(S_t, x_t) : A_t x_t = b_t^\pi(\theta_f^b)\}$$

where θ^c is the vector of cost-function parameters, θ^b is the vector of constraint parameters, and ϕ_f are the basis functions corresponding to the features $f \in \mathcal{F}$ relevant to the problem. This parameterization allows the model to directly incorporate domain-specific intuition regarding how uncertainty should be handled. In a dynamic VRP, a practitioner might inherently know that waiting to serve clients in order to group them together usually leads to better routing. Using CFA one can signal this knowledge by fine-tuning the parameters, which is done offline via the simulator: their relative weight should resemble their strategic relevance.

Therefore, the critical paradigm shift in the CFA approach is that the complexity of modeling stochasticity is entirely moved from the field (the look-ahead model) to the laboratory (the stochastic base model). The parameterized deterministic look-ahead is kept computationally simple and fast to solve online, while the heavy

algorithmic lifting —finding the optimal parameter vector θ via stochastic search— is performed offline within a realistic simulator.

Chapter 4

Problem formulation

Our goal is to optimize service to a pool of clients over a month: we take the working days of July 2025 for both data. This means that we want to serve the maximum number of clients with the minimum travel distance while controlling the number of clients that are served after their time window expires.

Clients are revealed progressively as the days unfold. A client i is seen for the first time by the solver once their time-window, i.e. the interval of days in which they can be served on time, starts at time b_i . When the time-window ends at time e_i , clients are still available for service if they have not been visited.

Each client is characterized by their location (x_i and y_i coordinates, or Latitude Lat_i and Longitude Lon_i in real-world's data), their time-window, and their service time s_i , which is the time that a truck needs to spend at the client location before moving on. Borrowing from Mondo Convenienza's context, this can be the time needed to assemble furniture, for example. Service times consume the available daily time of 8 working hours, which also needs to be considered when traveling from a location to another, at a fixed speed of 30 km/h. This value was chosen to get close to the average daily travel time of real data. Trucks have the ability to use some slack, which is penalized, to accommodate some flexibility.

It can happen, especially in real-world's data, that a client is too far and has too high service time to be served in a day. In this case, their order is split into two or more parts: first we take the service time that can fit within the available time considering round-trip travel times, and leave the remainder as another order with same time-window and location. We perform this operation iteratively until all service time is allocated to feasible orders.

Clients are divided in clusters $k = \{1, 2, \dots, K\}$ to take advantage of probabilistic information on upcoming orders. The size of the clusters depends on the instance and is selected for them to be neither too big nor too small, making the information relevant: if too large, arrival probabilities will often be equal to 1, and close to zero in the other case. At the start of the simulation, a mean daily order arrival

rate θ_k is computed based on historical available data up to the first day. This estimate is updated every day through an exponential smoothing mechanism: given the new present arrival date $r_{k,t}$, the estimate for each cluster becomes $q_{k,t} = \alpha r_{k,t} + (1 - \alpha)q_{k,t-1}$, with $q_{k,0} = \theta_k$ and $\alpha \in (0,1)$. The arrival probabilities of each cluster are approximated as exponential random variables and they are computed over the remaining time windows for each client: for each available client we compute $p_{i,t} = 1 - \exp[-q_{k,t}(e_i - t)]$, if $t < e_i$ and client i belongs to cluster k . Finally, each client is characterized by a score that depends on their service time. This choice has been made to incentivize service of big orders, that usually correspond to remunerative sales and can be left behind by the solver if their demanding service time is not compensated by higher prizes. Scores are defined as follows:

$$score_i = \begin{cases} 1 & \text{if } s_i < 30 \\ 2 & \text{if } 30 \leq s_i < 60 \\ 5 & \text{if } 60 \leq s_i < 120 \\ 10 & \text{otherwise} \end{cases}$$

4.1 Instance generation

4.1.1 Synthetic data

We consider three types of synthetic instances, which have different geographical characteristics: clients can be sampled according to rectangular, circular, or globular clusters. For each geographic type, two instances are generated, one with fixed time windows of 7 days and the other with variable time-windows, sampled from a uniform distribution between 4 and 7 days. In each case, we sample 320 clients over two months (June and July), even though we consider only clients that can be served in the latter (that is, whose time window was not yet expired at the 1st of July). The clients falling into the month of June are used to create initial estimates of clusters and arrival probabilities. Service times are sampled uniformly between 15 and 300 minutes, resembling Mondo Convenienza's data. Arrival days are again sampled uniformly among the working days of June and July 2025. If time-windows are fixed, 6 working days are added to define due dates; otherwise we sample a number between 3 and 6 at random and add it to the starting date.

Regarding the "Circular" clusters (Fig. 4.1a), we chose two radii equal to 20 and 50, with clients roughly distributing with proportion 5:3 for the two clusters respectively.

In the "Rectangular" instance (Fig. 4.1b), three rectangles of width 20, 70 and 110 and heights 10, 50 and 100 respectively are generated. The number of clients

are 80, 160 and 80 for the clusters.

In the rectangular and circular cases a random noise with distribution $\mathcal{N}(0,1)$ is added to the sampled points to allow some variation in the geography.

Finally, the "Globular" (Fig. 4.1c) instance has been created by generating three clusters with centers in coordinates (10, 15), (20, -30) and (-30, 10). The random distributions are multivariate isotropic Gaussians with standard deviations equal to 5, 7, and 9, respectively. For each cluster there are, in order, 80, 160 and 80 clients.

For every synthetic dataset the depot, i.e., the location from which trucks depart to serve clients and where they have to get back at the end of the day, has coordinates (0,0).

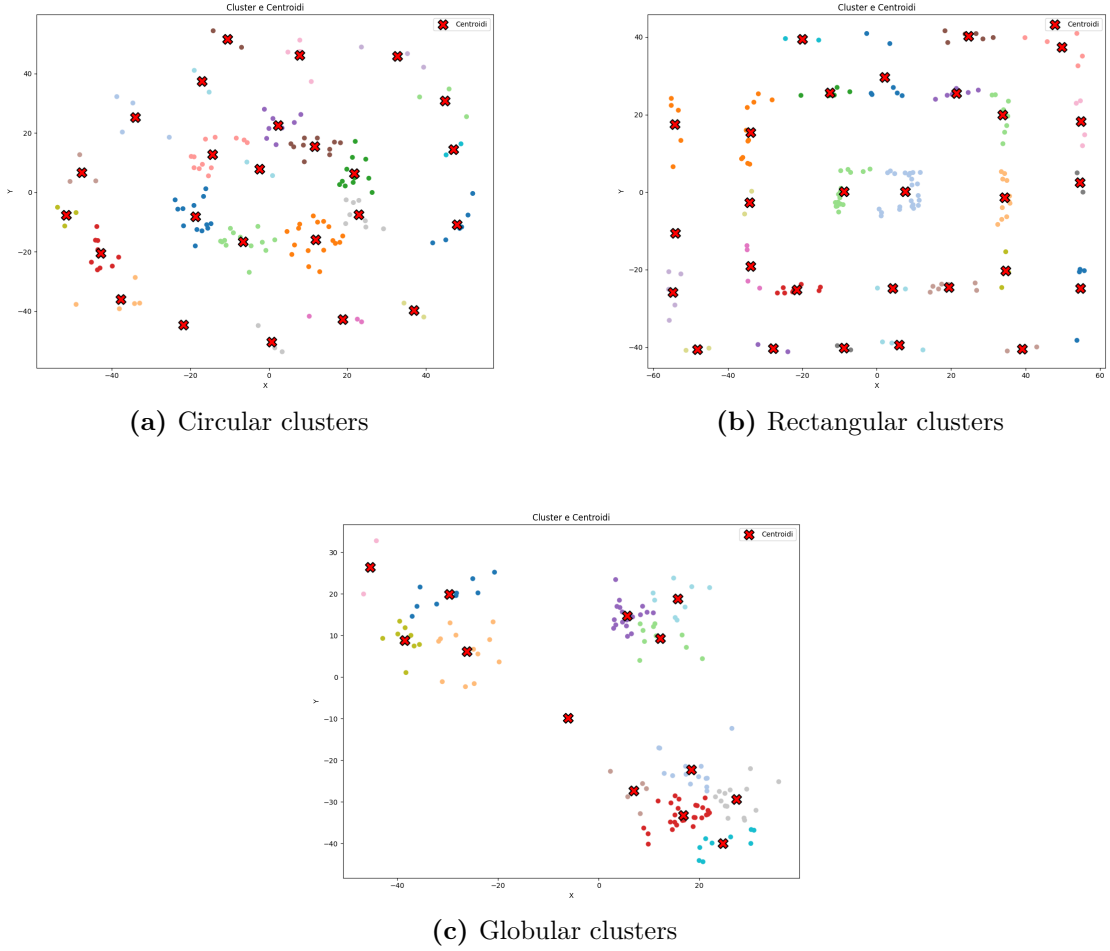


Figure 4.1: Synthetic datasets.

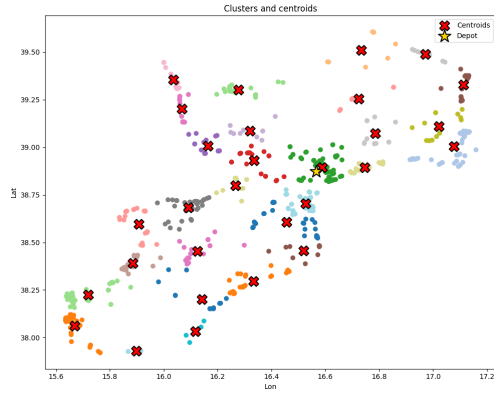
4.1.2 Mondo Convenienza's data

These records are the deliveries made by Mondo Convenienza from the depots of Catanzaro (Fig. 4.2) and Lainate (Fig. 4.3) in the month of July 2025. The data consist, for each client, of the day of service, the due date, the arrival date, the service time, the client's ID, and its Latitude and Longitude. In our case, to simulate dynamic orders, we discarded the day of service and computed the arrival day as the due date minus 6 working days.

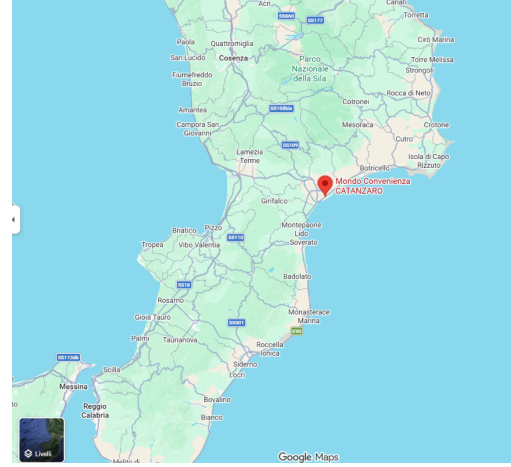
Originally, the Catanzaro set had 2776 records, while the Lainate set had 7157. The number of records in our simulation has increased compared to the original data, since unfeasible orders are split into smaller ones to allocate the whole service time in a series of feasible orders. This means that we take the service time for that client and create two orders: the first has the service time consisting of the available working hours minus the travel time from the depot to the client and vice-versa; the remaining time is allocated to a new order with the same location, due date, and arrival date, which can be split again if unfeasible, according to the same procedure.

The travel speed between clients is constant, equal to 30 km/h. This choice was made to obtain in the simulation the same average daily travel time for each truck as in the real data, which is around 180 minutes. Since some orders are too far from the depot and require a long service time, we made the assumption that the travel speed between the clusters and the depot depends on the distance, which resembles the presence of high-speed highways. This choice was necessary to make all orders feasible. Clusters were defined using complete-linkage and a maximum diameter of 30 km for Catanzaro and 10 km for Lainate, due to the smaller geographic region covered by this depot. These choices were made to avoid uninformative probabilistic information or clusters that did not represent the actual geography, as can happen with k-means.

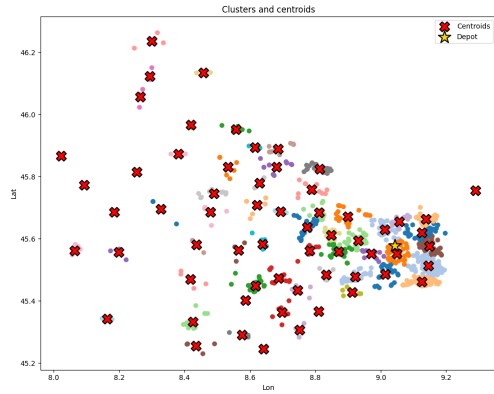
A relevant point in these sets is the presence of many late orders, particularly in Catanzaro. This makes the prizes in the cost function approximation approach similar between many clients in the first days of simulation.



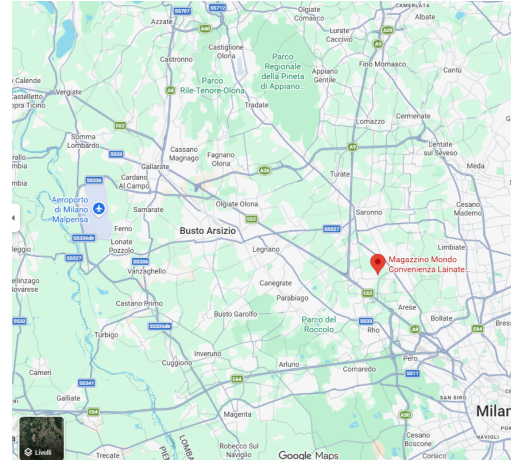
(a) Orders distribution - star is depot



(b) Real map

Figure 4.2: Catanzaro dataset and geographic area.

(a) Orders distribution - star is depot



(b) Real map

Figure 4.3: Lainate dataset and geographic area.

4.2 Policies

To test the performance of our cost function approximation approach, we compare this to two heuristics: First In First Out (FIFO) and Earliest Due Date (EDD).

4.2.1 First In First Out

Clients with earlier starting date have higher priority: when there is a tie, precedence is given to clients with highest service time, which usually correspond to biggest orders that we don't want to leave behind.

To implement this policy, clients are ordered and each truck is filled according to the priority rules. Only clients that can be served from the current point (client or depot) and from which we can go back to the depot, are considered at each iteration. This procedure continues until the available time is up or no other clients can be inserted in the route.

Once each truck is filled, we have a feasible list of clients that can be served during the day. This list is passed to the optimization solver, which can then further improve the routes.

4.2.2 Earliest Due Date

EDD works in a similar fashion as FIFO, but in this case clients whose time window ends earlier have higher priority. Clients with longer service time are preferred in the ties; the truck fulfillment and route optimization procedures are implemented the same way as above.

Notice that for deterministic time-windows, FIFO and EDD are the same priority rule, thus leading to the same results.

4.2.3 Prize-Collecting policies

The prizes are a proxy for the priority given to each client. There are three factors that we want to consider: a probabilistic term such that clients belonging to clusters in which we expect many future orders can be delayed to optimize service in the following days; an urgency term that increases progressively as the end of the time-window approaches, in order to avoid serving clients late; and a score depending on the service time as a signal of the importance of the order.

We consider two forms for prizes, one additive and one multiplicative. The following equations are used when $t < e_i$. The **Additive** (ADTV) one considers three parameters that can be fine-tuned, one for each term. The prize for client i at time t is thus defined as

$$prize_{i,t} = (1 - p_{i,t}) * prob_const + time_const / (e_i - t) + score_i * score_const$$

The **Multiplicative** prize (MLTP) instead considers only one parameter, called Big_M , which is multiplied by the other terms:

$$prize_{i,t} = Big_M * \frac{2 - p_{i,t}}{2} * score_i / (e_i - t)$$

The term $\frac{2-p_{i,t}}{2}$, ranging from 0.5 to 1, has been chosen to avoid the prizes from coming too close to zero in crowded clusters. In both frameworks, when service to a client is late, that is, if $t \geq e_i$, their prize is set equal to

$$prize_{i,t} = Big_M * score_i * 2$$

in this way, all late clients have very high prizes, which incentivize service as soon as possible.

4.3 Daily subproblem formulation

4.3.1 FIFO and EDD: Vehicle Routing Problem with service times

While evaluating the heuristics performance, we solve a Vehicle Routing Problem with service times each day. The list of clients found through the heuristic rule is passed to the solver, which then optimizes the routes. Thus, the fundamental difference with respect to the CFA policies is that every client must be served - the list we built is always feasible - and we minimize over the total travel cost and the slack. Assuming linearly proportional travel costs, the problem can then be formulated as:

$$\min \quad \sum_{i,j \in \mathcal{I}, k \in \mathcal{K}} \delta_{i,j,k} c_{i,j} d_{i,j} + \sum_{k \in \mathcal{K}} \beta \sigma_k$$

$$\text{s.t.} \quad \sum_{k \in \mathcal{K}} \sum_{j=1}^N \delta_{0,j,k} = \sum_{k \in \mathcal{K}} \sum_{i=1}^N \delta_{i,0,k} = |\mathcal{K}| \quad (1)$$

$$\sum_{i:(i,h) \in \mathcal{A}} \delta_{i,h,k} = \sum_{j:(h,j) \in \mathcal{A}} \delta_{h,j,k}, \quad h \in \mathcal{I}, k \in \mathcal{K} \quad (2)$$

$$\sum_{k \in \mathcal{K}} \sum_{j:(i,j) \in \mathcal{A}} \delta_{i,j,k} = 1, \quad i \in \mathcal{I} \quad (3)$$

$$\sum_{i=0}^N \sum_{j:(i,j) \in \mathcal{A}} (t_{i,j} + s_j) \delta_{i,j,k} \leq T_{\max} + \sigma_k, \quad k \in \mathcal{K} \quad (4)$$

$$u_{i,k} + t_{i,j} + s_i - M(1 - \delta_{i,j,k}) \leq u_{j,k}, \quad i, j \in \mathcal{I}, k \in \mathcal{K} \quad (5)$$

$$\delta_{i,j,k} \in \{0,1\}, \quad i, j \in \mathcal{I}, k \in \mathcal{K} \quad (6)$$

$$u_{i,k} \geq 0, \sigma_k \geq 0, \quad i \in \mathcal{I}, k \in \mathcal{K} \quad (7)$$

$\mathcal{I}$ is the set of clients, while $\mathcal{K}$ is the set of vehicles. $\delta_{i,j,k}$ is 1 if vehicle k goes from i to j directly. $t_{i,j}$ is the travel time between i and j , while s_i is the service

time at node i . T_{max} is the available daily working time. $u_{i,k}$ is the time in the day when vehicle k visits i .

The objective function minimizes travel costs minus the slack, multiplied by a penalty term β . Constraint (1) forces all routes to start and end at the depot, (2) ensures the connectivity of routes, (3) makes sure that each client is visited exactly once. Constraint (4) is related to the daily time budget, spent on travel and service times, to which the slack is added. (5) is the Miller-Tucker-Zemlin constraint to avoid sub-tours, i.e. routes between clients that do not start and end at the depot: it imposes that if we go from i to j ($\delta_{i,j,k} = 1$) we visit j after the travel time and the service time, thus avoiding going back to i afterwards.

4.3.2 Cost function approximation: Prize-Collecting VRP

In the cost function approximation approach, the daily subproblem is a Prize-Collecting VRP, in which clients can be delayed and we maximize the total prize minus travel costs and slack. It is formulated as:

$$\begin{aligned}
 \max \quad & \sum_{i \in \mathcal{I}} prize_i \left(\sum_{k \in \mathcal{K}} \sum_{j: (i,j) \in \mathcal{A}} \delta_{i,j,k} \right) - \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} \delta_{i,j,k} c_{i,j} d_{i,j} - \sum_{k \in \mathcal{K}} \beta \sigma_k \\
 \text{s.t.} \quad & \sum_{k \in \mathcal{K}} \sum_{j=1}^N \delta_{0,j,k} \leq |\mathcal{K}| \tag{1} \\
 & \sum_{i: (i,h) \in \mathcal{A}} \delta_{i,h,k} = \sum_{j: (h,j) \in \mathcal{A}} \delta_{h,j,k}, \quad h \in \mathcal{I}, k \in \mathcal{K} \tag{2} \\
 & \sum_{k \in \mathcal{K}} \sum_{j: (i,j) \in \mathcal{A}} \delta_{i,j,k} \leq 1, \quad i \in \mathcal{I} \tag{3} \\
 & \sum_{i=0}^N \sum_{j: (i,j) \in \mathcal{A}} (t_{i,j} + s_j) \delta_{i,j,k} \leq T_{max} + \sigma_k, \quad k \in \mathcal{K} \tag{4} \\
 & u_{i,k} + t_{i,j} + s_i - M(1 - \delta_{i,j,k}) \leq u_{j,k}, \quad i, j \in \mathcal{I}, k \in \mathcal{K} \tag{5} \\
 & \delta_{i,j,k} \in \{0,1\}, \quad i, j \in \mathcal{I}, k \in \mathcal{K} \tag{6} \\
 & u_{i,k} \geq 0, \sigma_k \geq 0 \quad i \in \mathcal{I}, k \in \mathcal{K} \tag{7}
 \end{aligned}$$

The model is very similar to the previous one, but in this case constraint (1) and (3) have a $\leq$ sign rather than $=$ because we are not forced to use all trucks nor visit all clients.

4.3 – Daily subproblem formulation

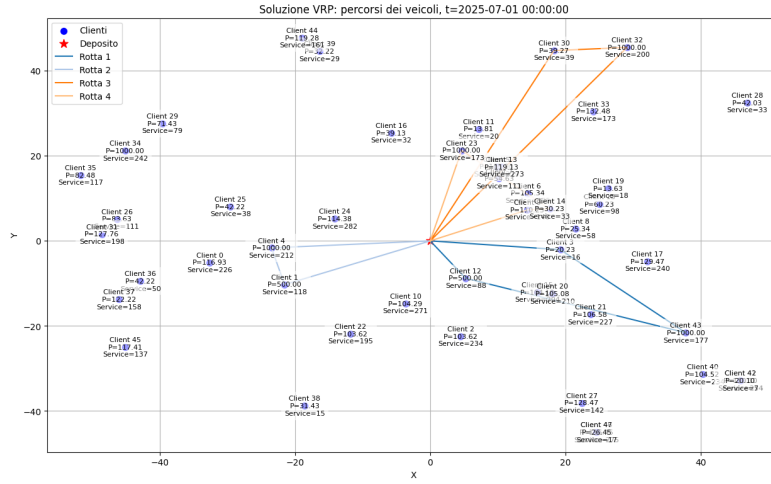


Figure 4.4: Visualization of a prize-collecting vehicle routing problem.

Chapter 5

Implementation details

There are three main entities that constitute the simulation code: simulators, solvers, and data handlers.

5.1 Simulators

There are two simulators that deal either with real data or synthetic data: `simulator_mc` and `simulator_synthetic`. Their functioning is identical, apart from some minor differences due to the different data structures: for example, real data clients have location expressed in geographical coordinates, so all distances have to be computed using the haversine formula.

Two main methods manage the simulation and are implemented also by the abstract superclass `AbstractSimulator`: `simulate_one_step` and

`simulate_the_horizon`. The first simulates one single day, while the latter carries on the whole simulation.

Once `simulate_the_horizon` is launched, before the simulation starts, a range of methods perform all necessary pre-processing:

- `add_score` computes the score for each client based on the service time.
- `get_starting_dates` computes starting dates in case of Mondo Convenienza's data; this step is carried when the data is generated in the case of the synthetic instances.
- `label_df` identifies the clusters based on historical data (i.e. orders that have appeared before the first of July), and assigns the remaining clients to these clusters.
- `compute_depot_to_cluster_speed` identifies the speed from the depot to the clusters and viceversa, based on the distance between the depot and the

clusters' medoid. This is done only in the case of Mondo Convenienza's data: in particular, the Catanzaro depot covers a vast geographical area, which makes this step necessary. The speed for distances equal to or greater than 100 km is 70 km/h, while between 100 and 50 km included is 40 km/h. In all other cases is equal to 30 km/h.

- If we use the cost function approximation, arrival rates are computed at the beginning of the simulation and before each day by the method `compute_rates`.

In the actual simulation loop, `simulate_one_step` takes all remaining clients and filters the ones that can be served, i.e. those whose starting date is lower or equal to the current day and have not been served yet. If FIFO or EDD are used, clients are further filtered by selecting the ones that we want to serve before passing the list to the solver.

Once the solver has solved the optimization problem, the results are collected and saved in a results dataframe and other data structures that are passed to the outer simulation loop to carry on the simulation and to be saved for the final results summary. The findings are written in an Excel file.

5.2 Solvers

Solvers take the input data and solve the optimization problem, giving the results to the simulator. There are two solvers, that deal with the VRP and the Prize-Collecting VRP, respectively: `hexaly_VRP_solver` and `hexaly_PC_solver`. The Hexaly optimization library is used since its great flexibility allows us to solve our formulation of the problem efficiently.

The two methods are:

`solve_and_get_results`: takes the input data, solves the problem, organizes the results in a dataframe, gives them in output, and prints them.

`plot_the_solution`: plots the served client on the map, along with the different routes taken by the trucks.

5.3 Data handlers

Data handlers are responsible for generating synthetic instances, reading the data, and other lateral operations.

The module `generate_instance` defines the synthetic datasets: it samples clients' location, their service time and their starting date and due date.

`read_mc_data` reads Mondo Convenienza's data to feed it into the simulator.

`set_data` performs operations needed for the solver to operate, such as computing the clients' prize, the arrival probability in each cluster, and the distance between points.

Chapter 6

Computational experiments

This chapter compares the CFA approach to the two heuristics and investigates the impact of the different parameters in the performance of the algorithm. We are able to draw meaningful conclusions on the strategies that business operators should use when scheduling vehicle routing, depending on the instance characteristics and on the type of clients that we are serving, particularly regarding the real-data tests.

The sensitivity analysis that we have performed is a one-at-a-time method in which all parameters are kept fixed at a standard value and one is able to vary across predefined points.

Tests concerning real data were run only with fixed windows' lengths, to adapt to the true business case.

To better interpret the results, we underline the role of the parameters regarding the score, probability and time terms. The score term makes demanding clients more attractive, in order to compensate for their high service time which can cause the algorithm to neglect them and visit them only when they are late as their prize increases, making poor decisions. The probability term incentivizes service to clients in clusters where we don't expect many new orders in the future, in order to visit them as soon as they become easy to get to. The time term goal is to avoid serving clients too late. In the multiplicative prizes, the big M term functions as an order of magnitude for the prizes, in order for long-distance clients to be attractive.

6.1 Synthetic data

The datasets have different characteristics. In the Globular instances, clients are closer together, and the results variations between parameter values are minimal. The Circular case has clients at a greater distance, so different approaches have different outcomes and we can appreciate the impact of changes in parameters. Finally, in the Rectangular case clients are furthest apart, which makes these tests

more challenging, causing different behaviors in the algorithm.

Generally, the additive prize form performs better than other strategies. FIFO and EDD are always the worst choice, with significantly less clients visited, much higher Km/clients, number of late clients, and tardiness metrics. This shows that the CFA approach is definitely superior and worth implementing. Moreover, with this strategy we can gain relevant information on the most important factors that we must pay attention to when scheduling clients in a dynamic multi-period setting with uncertainty. We discuss this topic in more detail in the following sections.

Standard values for the parameters are $prob_const = 25$, $time_const = 25$, $score_const = 10$ and $big_M = 50$. These are selected as the ones generating the best results in the majority of instances, considering, in particular, the Kilometers/clients metric as the most meaningful to define efficiency.

In addition to kilometers and number of clients served, other important metrics are tardiness and lateness. Mathematically, we can write the tardiness for client i as $T_i = \max(0, C_i - d_i)$, where C_i is their completion time (i.e. the day they have been served) and d_i is their due date. Similarly, lateness is defined as $L_i = C_i - d_i$. Therefore, tardiness measures the number of days past the deadline at which a client is served. In contrast, lateness takes a negative value if the service occurs ahead of due date (representing the days remaining until the deadline) and equals tardiness otherwise. Although lateness is of minor interest—since early deliveries are not critical in this context—tardiness plays a more important role, as we want to avoid serving customers too late. Consequently, both this metric and the number of customers served after their due date represent key performance indicators.

6.1.1 Fixed time windows' lengths

score_const variations

In the Circular case, higher score levels result in greater prizes, that make on-time clients comparable to late clients. In this way, we tend to serve customers much later, making the tardiness and number of late clients explode for $score_const = 100$, as in (Fig. 6.1d), (Fig 6.1c) reaching a whopping 8 days average delay. Another effect is that for higher levels, we serve more clients for a similar distance (Fig. 6.1a), thus with lower km/client score (Fig. 6.1b). Even though service may be more efficient, such high levels of tardiness and late clients make this configuration non-viable. Low levels of the score factor significantly decrease the incentive for service, resulting in 20 less clients on average and an explosion of clients served after their window expires: they become attractive only when the big M parameter comes into play.

In the Globular case too high levels of the score term result in worse decisions: since clients are very close to each other, if prizes are uneven, the algorithm makes

bad choices. Moreover, we have the same dynamic with late clients observed in the Circular instance.

In the Rectangular instance the analysis is closer to the Circular case: at higher scores we have better km performances, but much poorer tardiness and late clients values.

Generally speaking, *score_const* is shown to be a fundamental parameter for efficient service: if it is not present, some clients are not attractive enough, while if too high, we tend to postpone some others aggressively.

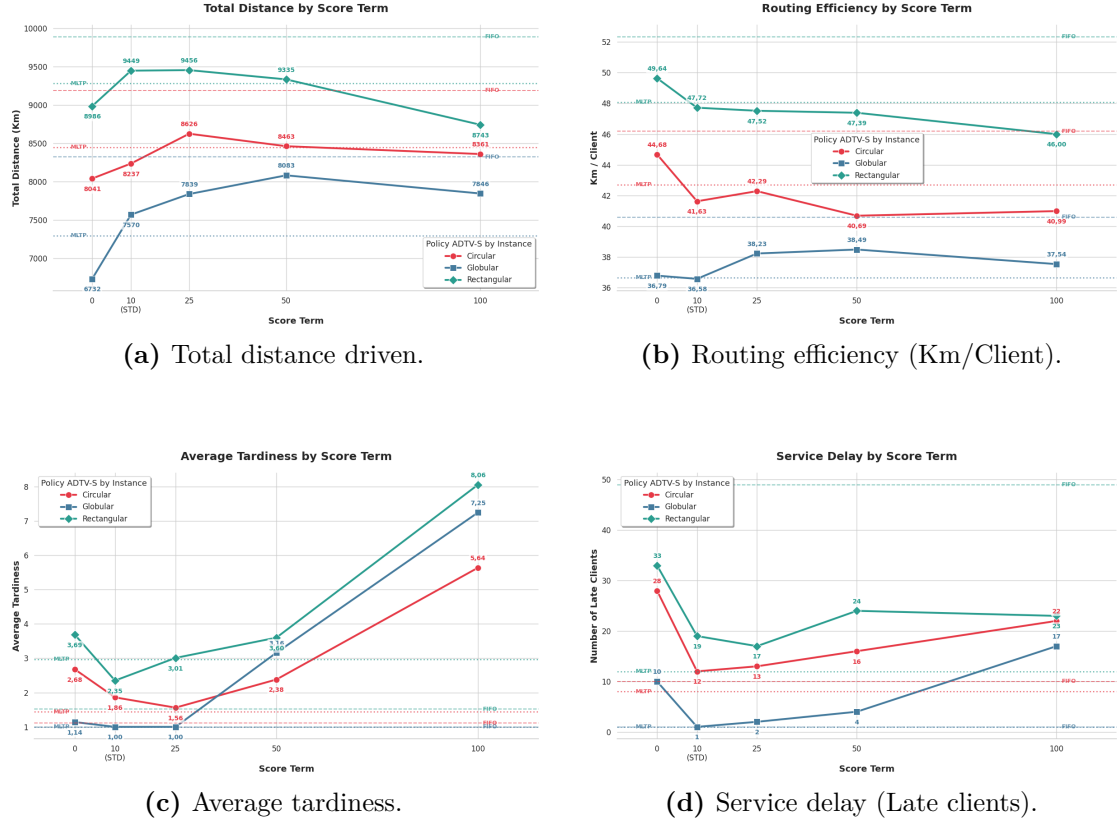


Figure 6.1: Performance analysis of the ADTV policy with variable **Score** term under **fixed windows**, on synthetic instances. Dashed lines represent the FIFO baseline, while dotted lines represent the MLTP strategy.

Table 6.1: Performance comparison in the circular set, fixed windows, variable score term.

Policy	Circular - Score - Fixed windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	9192	199	46,19	-2,02	1,11	10
MLTP - STD	8450	199	42,71	-3,56	1,44	8
ADTV - STD	8237	199	41,63	-3,34	1,86	12
ADTV - S 0	8041	180	44,68	-2,79	2,68	28
ADTV - S 25	8626	204	42,29	-3,39	1,56	13
ADTV - S 50	8463	208	40,69	-2,80	2,38	16
ADTV - S 100	8361	204	40,99	-0,31	5,64	22

Table 6.2: Performance comparison in the globular set, fixed windows, variable score term.

Policy	Globular - Score - Fixed windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	8324	205	40,60	-3,63	1	1
MLTP - STD	7295	199	36,66	-4,12	1	1
ADTV - STD	7570	207	36,58	-4,02	1	1
ADTV - S 0	6732	183	36,79	-3,29	1,14	10
ADTV - S 25	7839	205	38,23	-3,94	1	2
ADTV - S 50	8083	210	38,49	-4,08	3,16	4
ADTV - S 100	7846	209	37,54	-1,13	7,25	17

Table 6.3: Performance comparison in the rectangular set, fixed windows, variable score term.

Policy	Rectangular - Score - Fixed windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	9892	189	52,34	-0,78	1,52	49
MLTP - STD	9281	193	48,08	-3,2	2,97	12
ADTV - STD	9449	198	47,72	-3,00	2,35	19
ADTV - S 0	8986	181	49,64	-2,18	3,69	33
ADTV - S 25	9456	199	47,52	-2,90	3,01	17
ADTV - S 50	9335	197	47,39	-1,81	3,60	24
ADTV - S 100	8743	190	46,00	-2,00	8,06	23

prob_const variations

In every instance, higher probability scores increase the number of clients served at the cost of worsening the Km (Fig. 6.2b) and Km/clients (Fig. 6.2a) metrics by significant margins, decreasing the number of late clients (Fig. 6.2c) and the tardiness, apart from the Rectangular one, where these two actually increase. In the Globular and Circular instances, the standard value of 25 is the best, while in the rectangular case a value of 50 works better (Tab. 6.6). This can be explained due to the longer distances that characterize this instance. Therefore, the probability term works best as a fine-tuner to catch clients at medium distance that may not be served at the time otherwise. Too high values can make the algorithm visit clients that are too far from the best routes.

A value of zero worsens both KM and tardiness, stressing the relevance of the term.

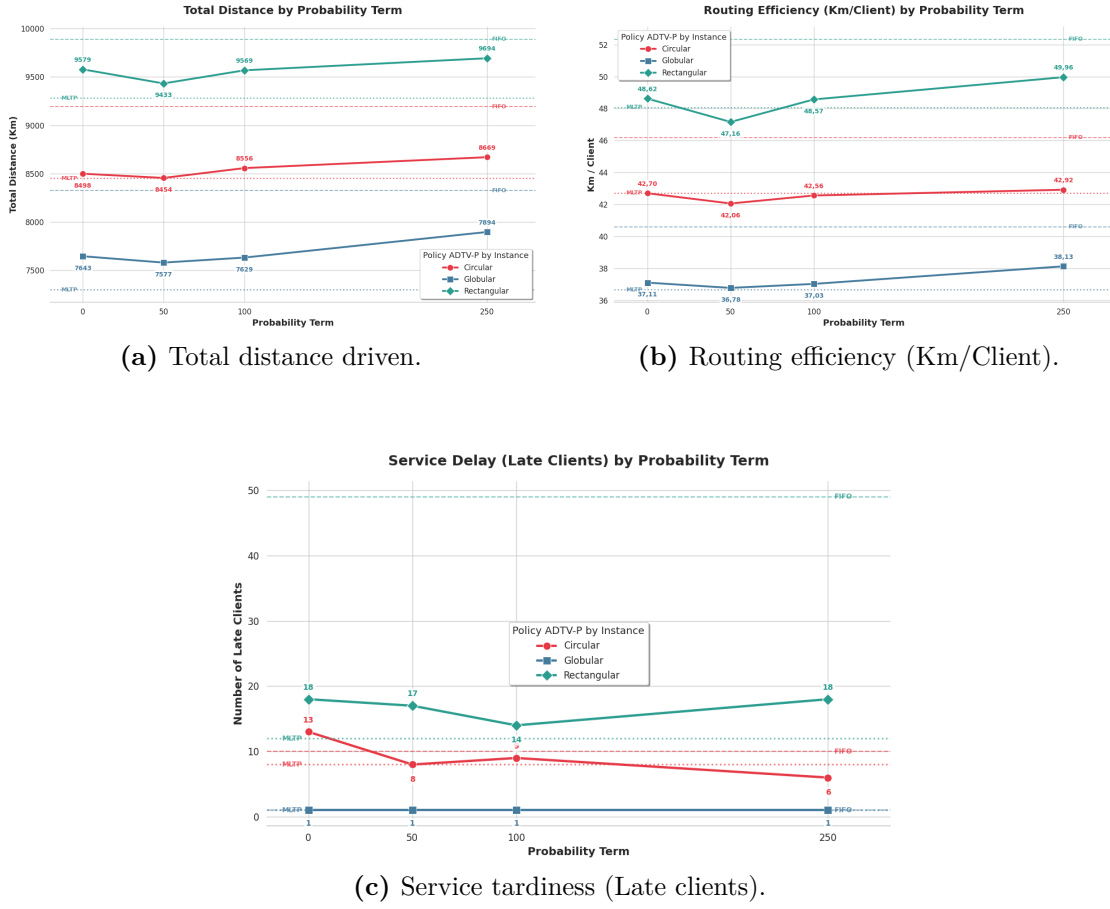


Figure 6.2: Performance analysis of the ADTV policy with variable **Probability** term under **fixed windows**, on synthetic instances. The FIFO baseline is denoted by dashed lines, while the MLTP strategy is represented by dotted lines.

Table 6.4: Performance comparison in the circular set, fixed windows, variable probability term.

Policy	Circular - Probabilities - Fixed windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	9192	199	46,19	-2,02	1,11	10
MLTP - STD	8450	199	42,71	-3,56	1,44	8
ADTV - STD	8327	200	41,63	-3,34	1,86	12
ADTV - P 0	8498	199	42,70	-3,23	2,19	13
ADTV - P 50	8454	201	42,06	-3,19	4,53	8
ADTV - P 100	8556	201	42,56	-3,49	2,33	9
ADTV - P 250	8669	202	42,92	-3,66	1,58	6

Table 6.5: Performance comparison in the globular set, fixed windows, variable probability term.

Policy	Globular - Probabilities - Fixed windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	8324	205	40,60	-3,63	1	1
MLTP - STD	7295	199	36,66	-4,12	1	1
ADTV - STD	7570	207	36,58	-4,02	1	1
ADTV - P 0	7643	206	37,11	-3,93	1	1
ADTV - P 50	7577	206	36,78	-4,09	1	1
ADTV - P 100	7629	206	37,03	-4,11	1	1
ADTV - P 250	7894	207	38,13	-4,21	1	1

Table 6.6: Performance comparison in the rectangular set, fixed windows, variable probability term.

Policy	Rectangular - Probabilities - Fixed windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	9892	189	52,34	-0,78	1,52	49
MLTP - STD	9281	193	48,08	-3,20	2,97	12
ADTV - STD	9449	198	47,72	-3,00	2,35	19
ADTV - P 0	9579	197	48,62	-2,90	2,81	18
ADTV - P 50	9433	200	47,16	-3,06	2,75	17
ADTV - P 100	9569	197	48,57	-3,10	3,06	14
ADTV - P 250	9694	194	49,96	-2,85	3,73	18

time_const variations

time_const seems to play a smaller role than other terms (Fig. 6.3a, Fig. 6.3b). For higher values, it improves the total number of served clients and, in the Circular case, the number of clients served on time, at the cost of worsening significantly the tardiness, which implies that late clients have similar prizes to the ones close to the end of their time-windows, therefore losing some appeal.

In general, its impact is not relevant in routing decisions and can be damaging for high values.

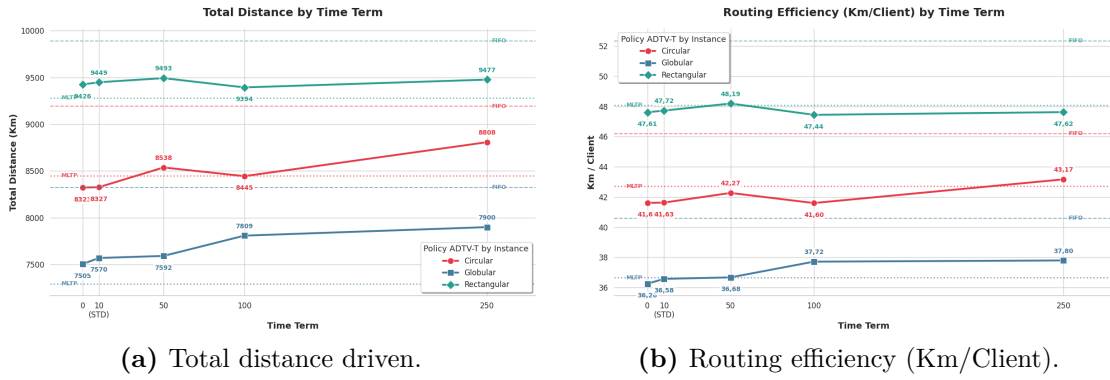


Figure 6.3: Performance analysis of the ADTV policy with variable **Time** term under **fixed windows**, on synthetic instances. Dashed lines represent the FIFO baseline, while dotted lines denote the MLTP standard.

Table 6.7: Performance comparison in the circular set, fixed windows, variable time term.

Policy	Circular - Time - Fixed windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	9192	199	56,19	-2,02	1,11	10
MLTP - STD	8450	199	42,71	-3,56	1,44	8
ADTV - STD	8327	200	41,63	-3,34	1,86	12
ADTV - T 0	8323	200	41,61	-3,33	2,00	10
ADTV - T 50	8538	202	42,27	-3,18	4,53	9
ADTV - T 100	8445	203	41,60	-3,30	4,63	8
ADTV - T 250	8808	204	43,17	-3,26	5,58	7

Table 6.8: Performance comparison in the globular set, fixed windows, variable time term.

Policy	Globular - Time - Variable windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	8324	205	40,60	-3,63	1	1
MLTP - STD	7295	199	36,66	-4,12	1	1
ADTV - STD	7570	207	36,58	-4,02	1	1
ADTV - T 0	7505	207	36,26	-3,96	1	1
ADTV - T 50	7592	207	36,68	-4,06	1	1
ADTV - T 100	7809	207	37,72	-4,10	1	1
ADTV - T 250	7900	209	37,80	-4,20	1	1

Table 6.9: Performance comparison in the rectangular set, fixed windows, variable time term.

Policy	Rectangular - Time - Fixed windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	9892	189	52,34	-0,78	1,52	49
MLTP - STD	9281	193	48,08	-3,20	2,97	12
ADTV - STD	9449	198	47,72	-3,00	2,35	19
ADTV - T 0	9426	198	47,61	-3,00	3,13	15
ADTV - T 50	9493	197	48,19	-3,02	2,70	15
ADTV - T 100	9394	198	47,44	-3,00	3,39	16
ADTV - T 250	9477	199	47,62	-2,68	4,66	19

Big_M variations

Higher values of *big_M* increase the magnitude of prizes in the multiplicative strategy, and the urgency associated to clients whose due date has expired in both policies.

Too high values are detrimental in all cases, suggesting that moderate urgency works best if we want to improve on Km and Km/client (Fig. 6.4a, Fig. 6.4c), with the tradeoff of worsening the number of clients served late (Fig. 6.4d).

On the other hand, higher prizes significantly improve the performance in the Rectangular additive instance, with around 350 km less than the standard case (Tab. 6.12).

Table 6.10: Performance comparison in the circular set, fixed windows, variable big M term.

Policy	Circular - Big M - Fixed windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	9192	199	46,19	-2,02	1,11	10
MLTP - STD	8450	199	42,71	-3,56	1,44	8
MLTP - M 100	8719	202	43,17	-3,66	2,08	6
MLTP - M 500	9006	204	44,14	-3,73	1,1	6
MLTP - M 1000	8990	207	43,43	-3,93	1	2
ADTV - STD	8327	200	41,63	-3,34	1,86	12
ADTV - M 100	8395	202	41,56	-3,38	1,67	10
ADTV - M 500	8596	202	42,56	-3,34	1,58	9
ADTV - M 1000	8483	202	42	-3,4	1,3	9

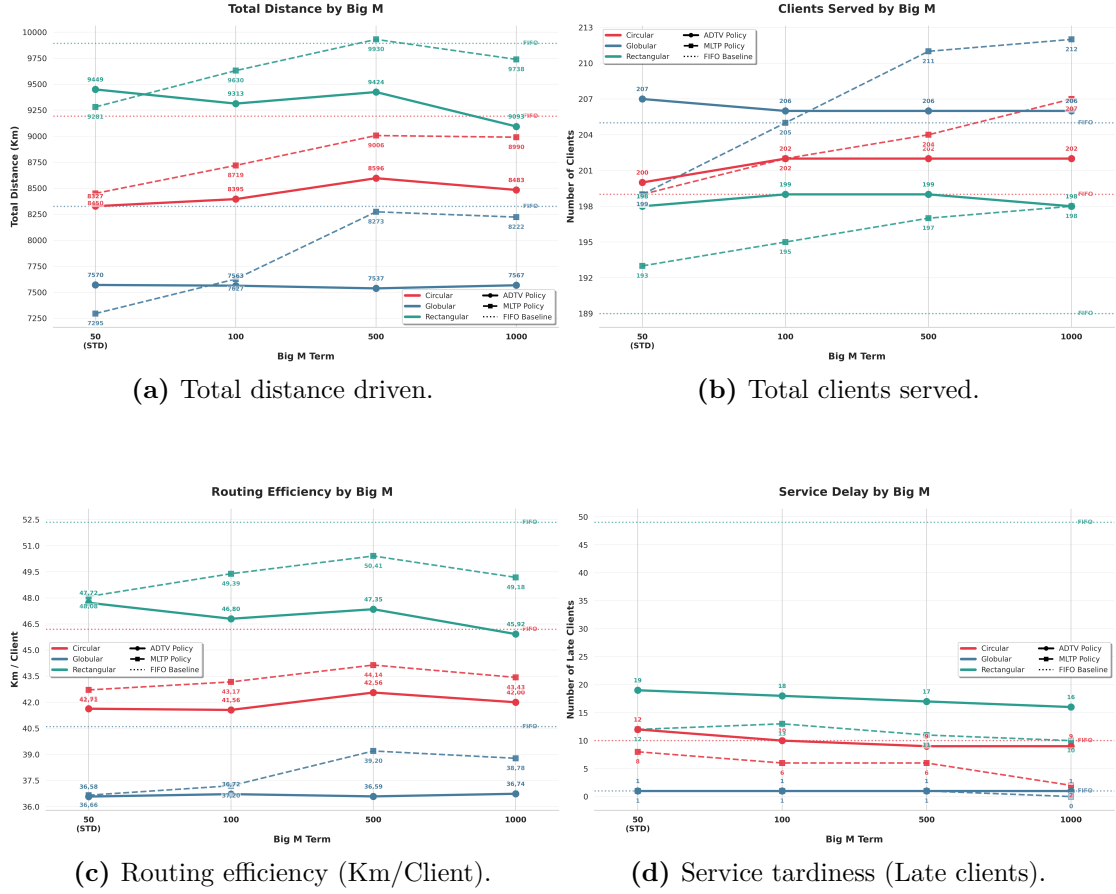


Figure 6.4: Performance analysis of the ADTV and MLTP policies over variable **Big M** values under **fixed windows**, on synthetic instances. Solid lines represent the proposed ADTV policy, dashed lines indicate the MLTP strategy, and dotted lines establish the FIFO baseline.

Table 6.11: Performance comparison in the globular set, fixed windows, variable big M term.

Policy	Globular - Big M - Fixed windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	8324	205	40,6	-3,63	1	1
MLTP - STD	7295	199	36,66	-4,12	1	1
MLTP - M 100	7627	205	37,2	-4,26	1	1
MLTP - M 500	8273	211	39,2	-4,5	1	1
MLTP - M 1000	8222	212	38,78	-4,6	0	0
ADTV - STD	7570	207	36,58	-4,02	1	1
ADTV - M 100	7563	206	36,72	-3,93	1	1
ADTV - M 500	7537	206	36,59	-3,95	1	1
ADTV - M 1000	7567	206	36,74	-3,96	1	1

Table 6.12: Performance comparison in the rectangular set, fixed windows, variable big M term.

Policy	Rectangular - Big M - Fixed windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	9892	189	52,34	-0,78	1,52	49
MLTP - STD	9281	193	48,08	-3,2	2,97	12
MLTP - M 100	9630	195	49,39	-3,12	3,33	13
MLTP - M 500	9930	197	50,41	-3,25	3,79	11
MLTP - M 1000	9738	198	49,18	-3,36	2,3	10
ADTV - STD	9449	198	47,72	-3	2,35	19
ADTV - M 100	9313	199	46,8	-3,11	2,04	18
ADTV - M 500	9424	199	47,35	-3,13	2,3	17
ADTV - M 1000	9093	198	45,92	-3,2	1,89	16

6.1.2 Variable time windows' length

We now perform a sensitivity analysis in instances where the time windows are sampled from a uniform distribution between 4 and 7 days.

We find that the relative importance of constants does not vary with respect to the previous case. In general, the algorithm travels less kilometers to serve a lower number of clients, resulting in similar Km/client values. Since here the driving

factor is usually the Big M parameters, as we have tighter deadlines, some terms play a smaller role with respect to the context with fixed windows.

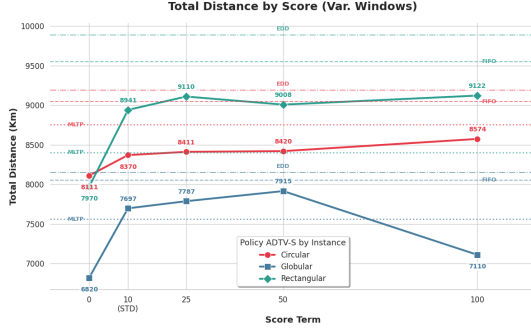
CFA still significantly outperforms the heuristics: EDD and FIFO generate similar results, with the first usually traveling a further distance and serving less clients late, depending on the case.

score__const variations

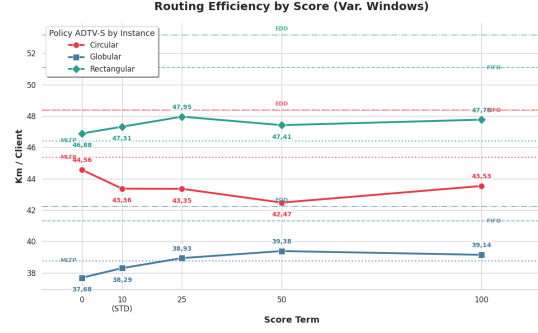
In the case of variable windows, the score term has a smaller effect on distance and efficiency, although moderate values still give the best performance (Fig 6.5a, Fig. 6.5b). As before, higher values make tardiness and late clients increase significantly (Fig. 6.5c, Fig. 6.5d). The difference in performance with respect to the fixed windows case can be related to the presence of tighter windows: the score is less fundamental in making difficult clients more attractive, as the *big_M* term comes into play sooner. A zero value worsens the performance, neglecting some clients, and serving others too late.

Table 6.13: Performance comparison in the circular set, variable windows, variable score term.

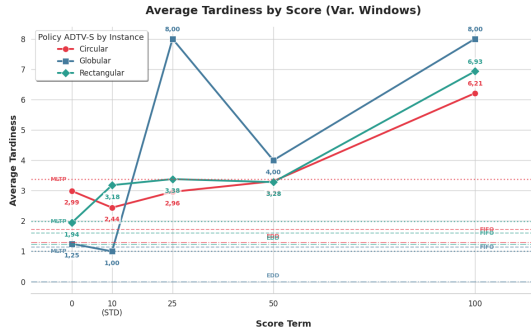
Policy	Circular - Score - Variable windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	9046	187	48,37	-0,6	1,72	56
EDD	9189	190	48,36	-0,59	1,29	58
MLTP - STD	8755	193	45,36	-2,34	3,37	20
ADTV - STD	8370	193	43,36	-2,19	2,44	27
ADTV - S 0	8111	182	44,56	-2,07	2,99	30
ADTV - S 25	8411	194	43,35	-2	2,96	32
ADTV - S 50	8420	197	42,47	-1,83	3,31	32
ADTV - S 100	8574	197	43,53	-2	6,21	21



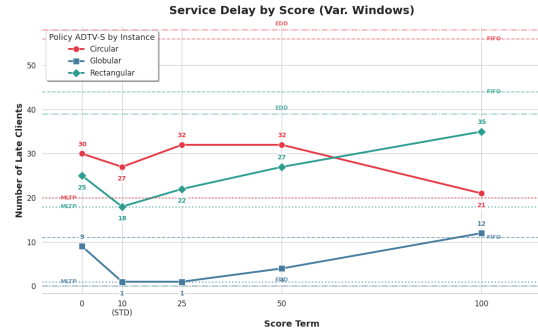
(a) Total distance driven.



(b) Routing efficiency (Km/Client).



(c) Average Tardiness.



(d) Number of Late Clients.

Figure 6.5: Performance analysis of the ADTV policy with variable **Score** term under **variable time windows**, on synthetic instances. The horizontal baselines represent the FIFO (dashed), EDD (dash-dot), and MLTP (dotted) heuristics.

Table 6.14: Performance comparison in the globular set, variable windows, variable score term.

Policy	Globular - Score - Variable windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	8053	195	41,3	-2,2	1,14	11
EDD	8150	193	42,23	-2,37	0	0
MLTP - STD	7560	195	38,77	-3,2	1	1
ADTV - STD	7697	201	38,29	-3,15	1	1
ADTV - S 0	6820	181	37,68	-2,56	1,25	9
ADTV - S 25	7787	200	38,93	-3	8	1
ADTV - S 50	7915	201	39,38	-3,1	4	4
ADTV - S 100	7110	197	39,14	-0,31	8	12

Table 6.15: Performance comparison in the rectangular set, variable windows, variable score term.

Policy	Rectangular - Score - Variable windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	9551	187	51,08	-1,44	1,6	44
EDD	9890	186	53,17	-1,32	1,23	39
MLTP - STD	8404	181	46,42	-2,71	1,99	18
ADTV - STD	8941	189	47,31	-2,36	3,18	18
ADTV - S 0	7970	170	46,88	-2,25	1,94	25
ADTV - S 25	9110	190	47,95	-2,36	3,38	22
ADTV - S 50	9008	190	47,41	-1,81	3,28	27
ADTV - S 100	9122	191	47,76	0,75	6,93	35

prob_const variations

The probabilities term generates the same behavior as in the fixed windows contest. It works as a fine-tuner for moderate values, while it is detrimental if too high. In this context, a higher value performs better: when we have closer deadlines, actually anticipating service more aggressively to moderately difficult clients helps prevent hard situations (Tab. 6.18).

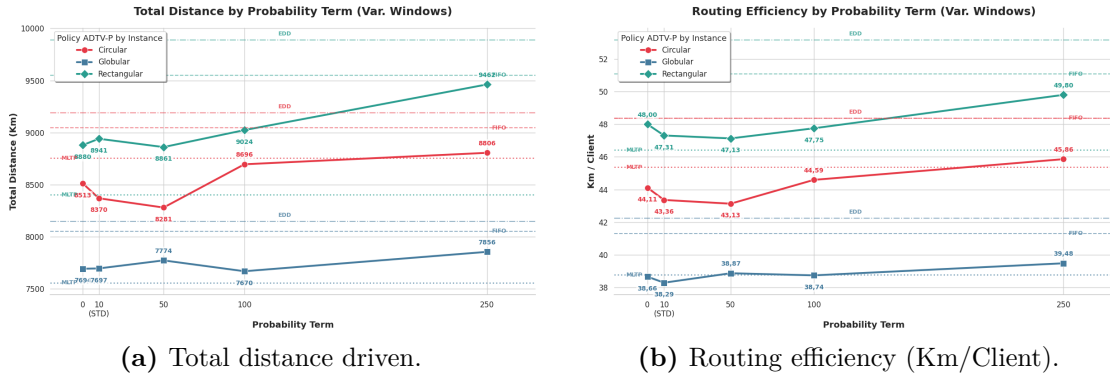


Figure 6.6: Performance analysis of the ADTV policy with variable **Probability** term under **variable time windows**, on synthetic instances. The horizontal baselines represent the FIFO (dashed), EDD (dash-dot), and MLTP (dotted) strategy.

Table 6.16: Performance comparison in the circular set, variable windows, variable probability term.

Policy	Circular - Probabilities - Variable windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	9046	187	48,37	-0,61	1,72	56
EDD	9189	190	48,36	-0,59	1,29	58
MLTP - STD	8755	193	45,36	-2,34	3,37	20
ADTV - STD	8370	193	43,36	-2,19	2,44	27
ADTV - P 0	8513	193	44,11	-2,15	2,1	29
ADTV - P 50	8281	192	43,13	-2,17	2,38	32
ADTV - P 100	8696	195	44,59	-2,24	2,2	25
ADTV - P 250	8806	192	45,86	-2,3	2,29	24

Table 6.17: Performance comparison in the globular set, variable windows, variable probability term.

Policy	Globular - Probabilities - Variable windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	8053	195	41,3	-2,2	1,14	11
EDD	8150	193	42,23	-2,37	0	0
MLTP - STD	7560	195	38,77	-3,2	1	1
ADTV - STD	7697	201	38,29	-3,15	1	1
ADTV - P 0	7694	199	38,66	-3	1	1
ADTV - P 50	7774	200	38,87	-3,13	1	1
ADTV - P 100	7670	198	38,74	-3,16	1	1
ADTV - P 250	7856	199	39,48	-3,19	3	1

Table 6.18: Performance comparison in the rectangular set, variable windows, variable probability term.

Policy	Rectangular - Probabilities - Variable windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	9551	187	51,08	-1,44	1,6	44
EDD	9890	186	53,17	-1,32	1,23	39
MLTP - STD	8404	181	46,42	-2,71	1,99	18
ADTV - STD	8941	189	47,31	-2,36	3,18	18
ADTV - P 0	8880	185	48	-2,48	2,14	25
ADTV - P 50	8861	188	47,13	-2,72	2,31	17
ADTV - P 100	9024	189	47,75	-2,81	2,28	15
ADTV - P 250	9462	190	49,8	-2,46	3,89	14

time_const variations

Even though one could expect that, when windows are tighter, a stronger urgency term can be more relevant, once again we find that its impact is relative: a small value can slightly improve metrics like Km/client (Fig. 6.7b), but not in a significant manner.

Table 6.19: Performance comparison in the circular set, variable windows, variable time term.

Policy	Circular - Time - Variable windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	9046	187	48,37	-0,61	1,72	56
EDD	9189	190	48,36	-0,59	1,29	58
MLTP - STD	8755	193	45,36	-2,34	3,37	20
ADTV - STD	8370	193	43,36	-2,19	2,44	27
ADTV - T 0	8470	192	44,11	-2,00	2,41	32
ADTV - T 50	8506	192	44,30	-2,16	2,38	28
ADTV - T 100	8472	193	43,90	-2,23	2,57	25
ADTV - T 250	8572	195	43,96	-2,16	3,17	27

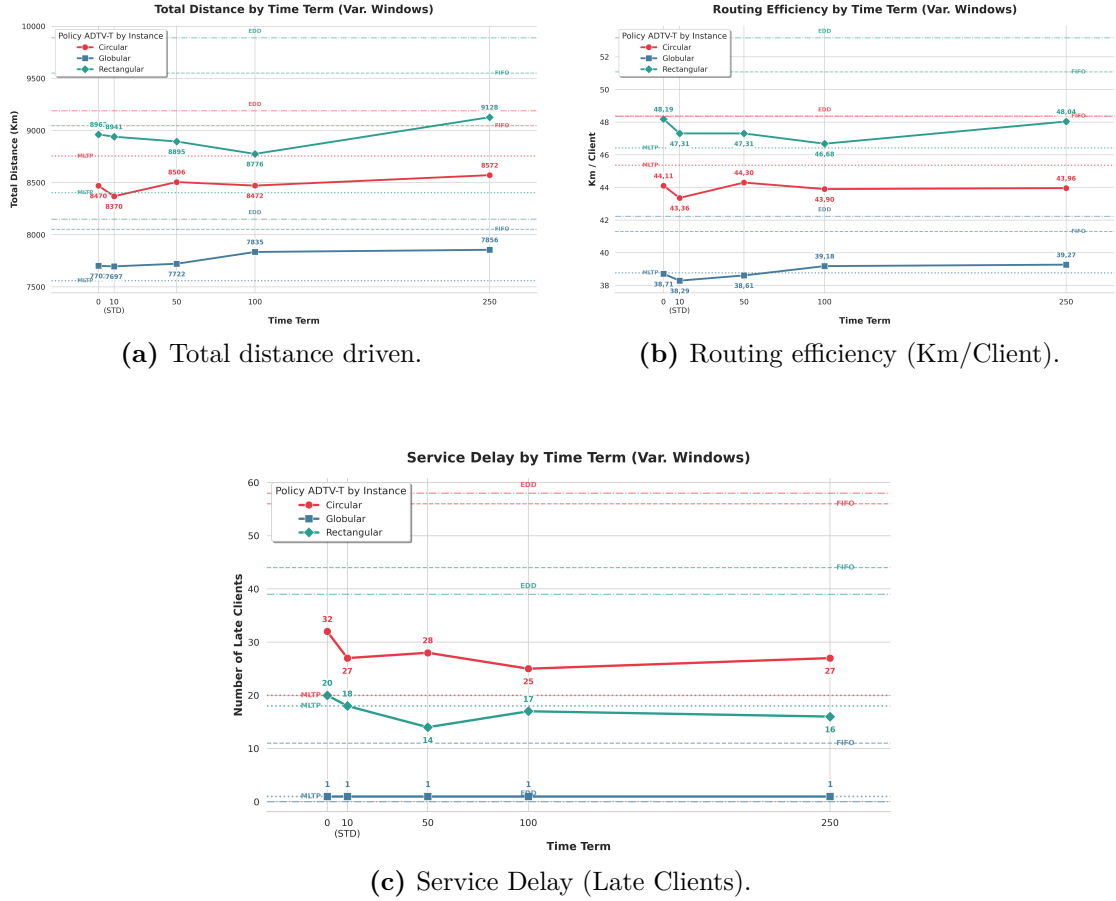


Figure 6.7: Performance analysis of the ADTV policy with a variable **Time** term under **variable time windows**, tested on synthetic instances. The horizontal baselines correspond to the FIFO (dashed line), EDD (dash-dot line), and standard MLTP (dotted line) reference heuristics.

Table 6.20: Performance comparison in the globular set, variable windows, variable time term.

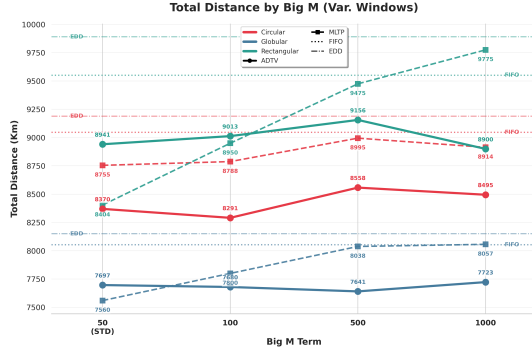
Policy	Globular - Time - Variable windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	8053	195	41,30	-2,20	1,14	11
EDD	8150	193	42,23	-2,37	0	0
MLTP - STd	7560	195	38,77	-3,20	1	1
ADTV - STD	7697	201	38,29	-3,15	1	1
ADTV - T 0	7702	199	38,71	-3,00	1	1
ADTV - T 50	7722	200	38,61	-3,17	1	1
ADTV - T 100	7835	200	39,18	-3,20	1	1
ADTV - T 250	7856	200	39,27	-3,20	1	1

Table 6.21: Performance comparison in the rectangular set, variable windows, variable time term.

Policy	Rectangular - Time - Variable windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	9551	187	51,08	-1,44	1,6	44
EDD	9890	186	53,17	-1,32	1,23	39
MLTP - STD	8404	181	46,42	-2,71	1,99	18
ADTV - STD	8941	189	47,31	-2,36	3,18	18
ADTV - T 0	8963	186	48,19	-2,6	2,35	20
ADTV - T 50	8895	188	47,31	-2,62	2,93	14
ADTV - T 100	8776	188	46,68	-2,7	2,32	17
ADTV - T 250	9128	190	48,04	-2,65	3,15	16

Big_M variations

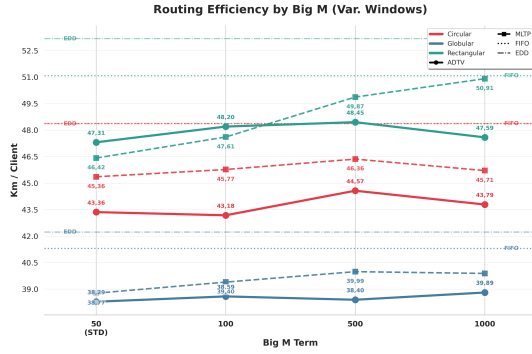
In the Circular instances additive prizes perform better, with higher values leading to worst performances. The Globular case does not show significant differences, while in the Rectangular one we have high values in the additive setting that give strong results. In this latter case, the best results come from the multiplicative standard setting, where lower prizes tend to avoid difficult clients, resulting in lower Km, number of clients and Km/client. Overall, changes in Big M do not generate variations in performance.



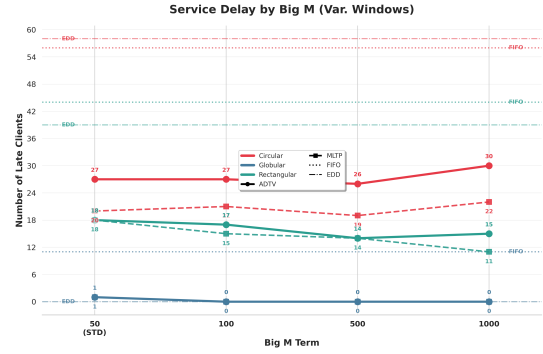
(a) Total distance driven.



(b) Total clients served.



(c) Routing efficiency (Km/Client).



(d) Service Delay (Late clients).

Figure 6.8: Performance analysis comparing ADTV and MLTP policies over variable **Big M** penalties under **variable time windows**, on synthetic instances. Solid lines represent ADTV, while dashed lines indicate MLTP. Horizontal baselines are provided for FIFO (dotted) and EDD (dash-dot) heuristics.

Table 6.22: Performance comparison in the circular set, variable windows, variable big M term.

Policy	Circular - Big M - Variable windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	9046	187	48,37	-0,6	1,72	56
EDD	9189	190	48,36	-0,59	1,29	58
MLTP - STD	8755	193	45,36	-2,34	3,37	20
MLTP - M 100	8788	192	45,77	-2,36	2,97	21
MLTP - M 500	8995	194	46,36	-2,55	2,38	19
MLTP - M 1000	8914	195	45,71	-2,58	2,24	22
ADTV - STD	8370	193	43,36	-2,19	2,44	27
ADTV - M 100	8291	192	43,18	-2,2	2,8	27
ADTV - M 500	8558	192	44,57	-2,31	1,81	26
ADTV - M 1000	8495	194	43,79	-2,37	1,73	30

Table 6.23: Performance comparison in the globular set, variable windows, variable big M term.

Policy	Globular - Big M - Variable windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	8053	195	41,3	-2,2	1,14	11
EDD	8150	193	42,23	-2,37	0	0
MLTP - STD	7560	195	38,77	-3,2	1	1
MLTP - M 100	7800	198	39,4	-3,29	0	0
MLTP - M 500	8038	201	39,99	-3,47	0	0
MLTP - M 1000	8057	202	39,89	-3,52	0	0
ADTV - STD	7697	201	38,29	-3,15	1	1
ADTV - M 100	7680	199	38,59	-3	0	0
ADTV - M 500	7641	199	38,4	-3,1	0	0
ADTV - M 1000	7723	199	38,81	-3,1	0	0

Table 6.24: Performance comparison in the rectangular set, variable windows, variable big M term.

Policy	Rectangular - Big M - Variable windows					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	9551	187	51,08	-1,44	1,6	44
EDD	9890	186	53,17	-1,32	1,23	39
MLTP - STD	8404	181	46,42	-2,71	1,99	18
MLTP - M 100	8950	188	47,61	-2,75	2,36	15
MLTP - M 500	9475	190	49,87	-3,03	1,71	14
MLTP - M 1000	9775	192	50,91	-3	2,45	11
ADTV - STD	8941	189	47,31	-2,36	3,18	18
ADTV - M 100	9013	187	48,2	-2,66	1,72	17
ADTV - M 500	9156	189	48,45	-2,72	1,92	14
ADTV - M 1000	8900	187	47,59	-2,62	1,96	15

6.2 Mondo Convenienza’s data

In this section, we test the cost function approximation approach on real data, taken from the deliveries Mondo Convenienza, one of Italy’s largest furniture companies, made in the month of July 2025 from the depots of Catanzaro and Lainate. We only consider fixed windows’ lengths of 7 working days to adhere to the business case. Thus, FIFO and EDD are the same policy, which we beat in all instances, proving that also in real word data our strategy is worth implementing. In addition, improvements are more evident as the number of clients served plummets, while the number of clients served late increases dramatically. We also have significant results regarding efficiency, that is, we observe lower km/client ratios. The geographical areas covered by the Catanzaro and Lainate depots are very different, as well as the type of clients. We explore this topic in more detail in the following sections.

6.2.1 Catanzaro depot

The Catanzaro depot covers a vast geographical area, approximately from Crotona to Reggio Calabria, and up to Cosenza, with a maximum diameter of more than 200 kilometers, and a distance from the depot of around 150 km, which can be covered in 1h 30min approximately. Therefore, we have many clients that are hard to serve, in particular if they also have long service times, and the order splitting mechanisms is needed to have a feasible solution, which means that we can go to the client, return and serve them in the available working time. We have 35 vehicles

at the start of the simulation, that increase up to 40 after some days, following the business dynamics.

Standard values for the terms are $score_const = 50$, $prob_const = 500$, $time_const = 500$, $big_M = 500$ for the additive strategy, while $big_M = 1000$ in the multiplicative case. Higher values are present due to longer distances.

In addition, 445 out of 815 clients have an expired due date at the start of the simulation (before challenging orders are split), which accounts for around 54,5% of orders serviceable in the first day (the total number of orders present in the simulation is 2778, before splitting). This creates a snowball effect for which we serve a great number of clients late: therefore, prizes are usually determined by the big_M constant, leaving less room for maneuver and fine-tuning of parameters in the additive framework.

This is reflected in the small effects on results in the changes of parameters. While always outperforming the heuristic, changing values of the different constants do not affect performance significantly: lines are flat in most graphs (as in Fig. 6.9, or Fig. 6.10). The effect that we can appreciate is that higher values of the parameters diminish the number of late clients (Fig. 6.10c).

Table 6.25: Performance comparison in the Catanzaro set, fixed windows, variable score term.

Policy	Catanzaro - Score					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	107514	2578	41,7	4,71	5,27	2329
MLTP - STD	113644	2937	38,69	-2,72	4,26	990
ADTV - STD	113422	2936	38,63	-2,69	4,38	980
ADTV - S 0	113191	2930	38,63	-2,51	4,19	1004
ADTV - S 25	113375	2934	38,64	-2,71	4,42	954
ADTV - S 100	113742	2941	38,67	-2,63	4,39	970

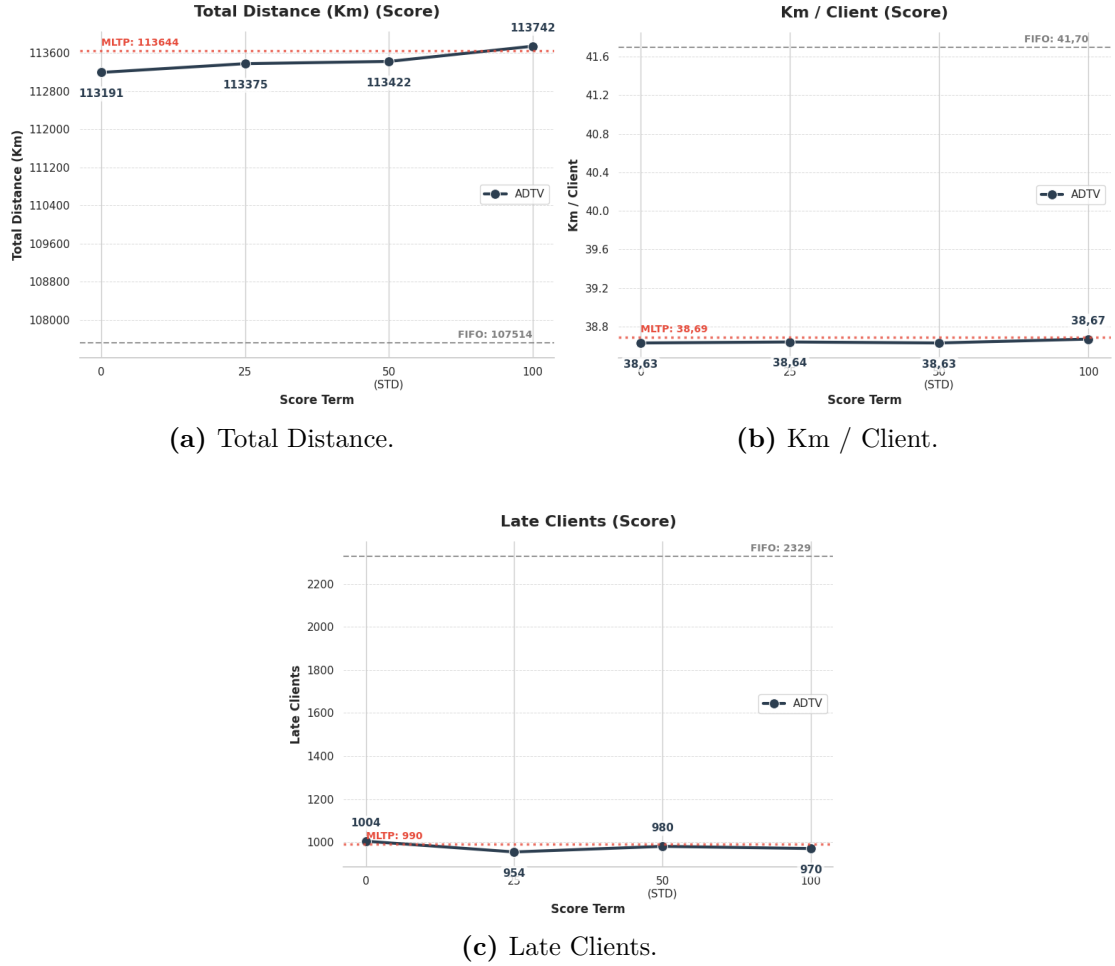


Figure 6.9: Performance evaluation on the real-world **Catanzaro** dataset varying the **Score** term. The solid blue line represents the ADTV policy. Baselines for FIFO and standard MLTP are reported.

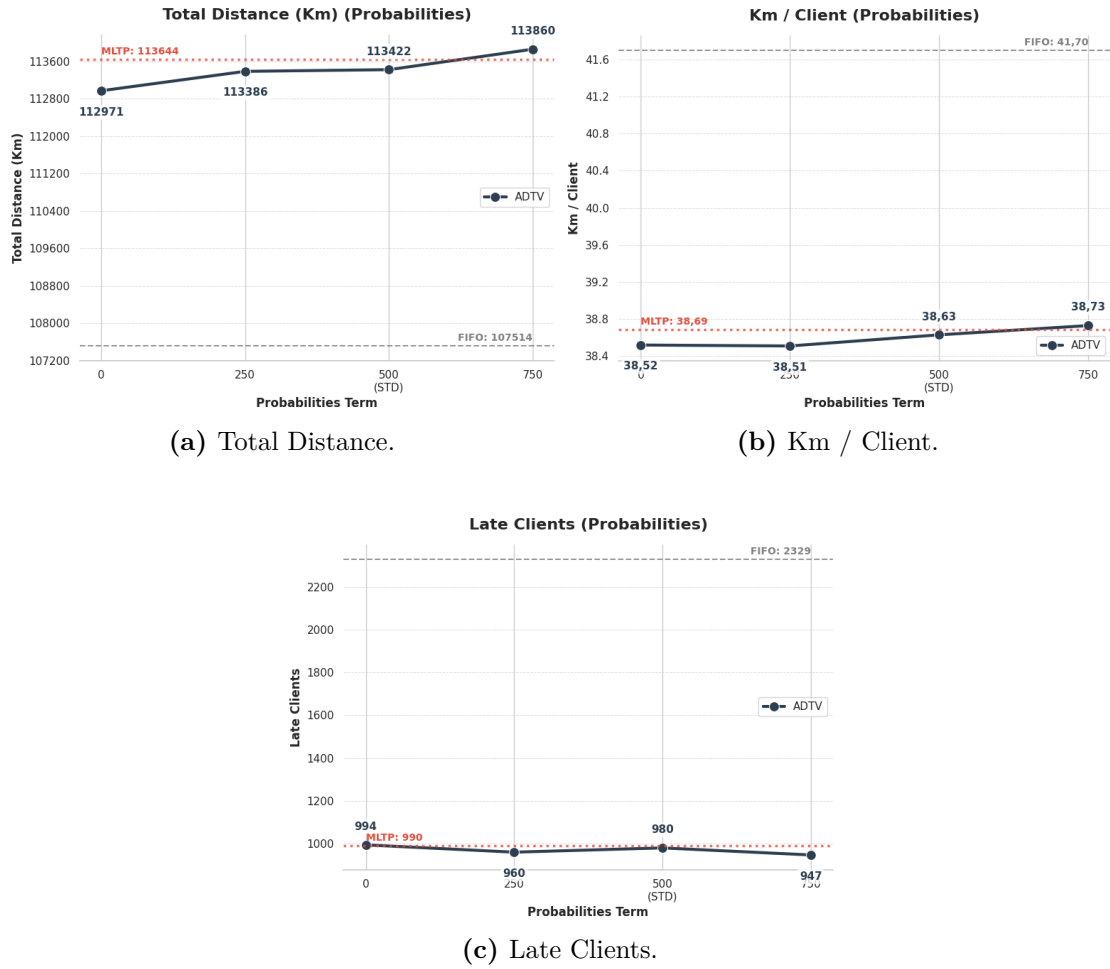


Figure 6.10: Performance evaluation on the real-world **Catanzaro** dataset varying the **Probability** term. The solid blue line represents the ADTV policy. Baselines for FIFO and standard MLTP are reported.

Table 6.26: Performance comparison in the Catanzaro set, fixed windows, variable probability term.

Policy	Catanzaro - Probabilities					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	107514	2578	41,7	4,71	5,27	2329
MLTP - STD	113644	2937	38,69	-2,72	4,26	990
ADTV - STD	113422	2936	38,63	-2,69	4,38	980
ADTV - P 0	112971	2933	38,52	-2,63	4,27	994
ADTV - P 250	113386	2944	38,51	-2,7	4,13	960
ADTV - P 750	113860	2940	38,73	-2,69	4,33	947

Table 6.27: Performance comparison in the Catanzaro set, fixed windows, variable time term.

Policy	Catanzaro - Time					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	107514	2578	41,7	4,71	5,27	2329
MLTP - STD	113644	2937	38,69	-2,72	4,26	990
ADTV - STD	113422	2936	38,63	-2,69	4,38	980
ADTV - T 0	113191	2930	38,63	-2,51	4,19	1004
ADTV - T 250	113179	2933	38,58	-2,62	4,25	983
ADTV - T 750	113245	2940	38,52	-2,67	4,28	967

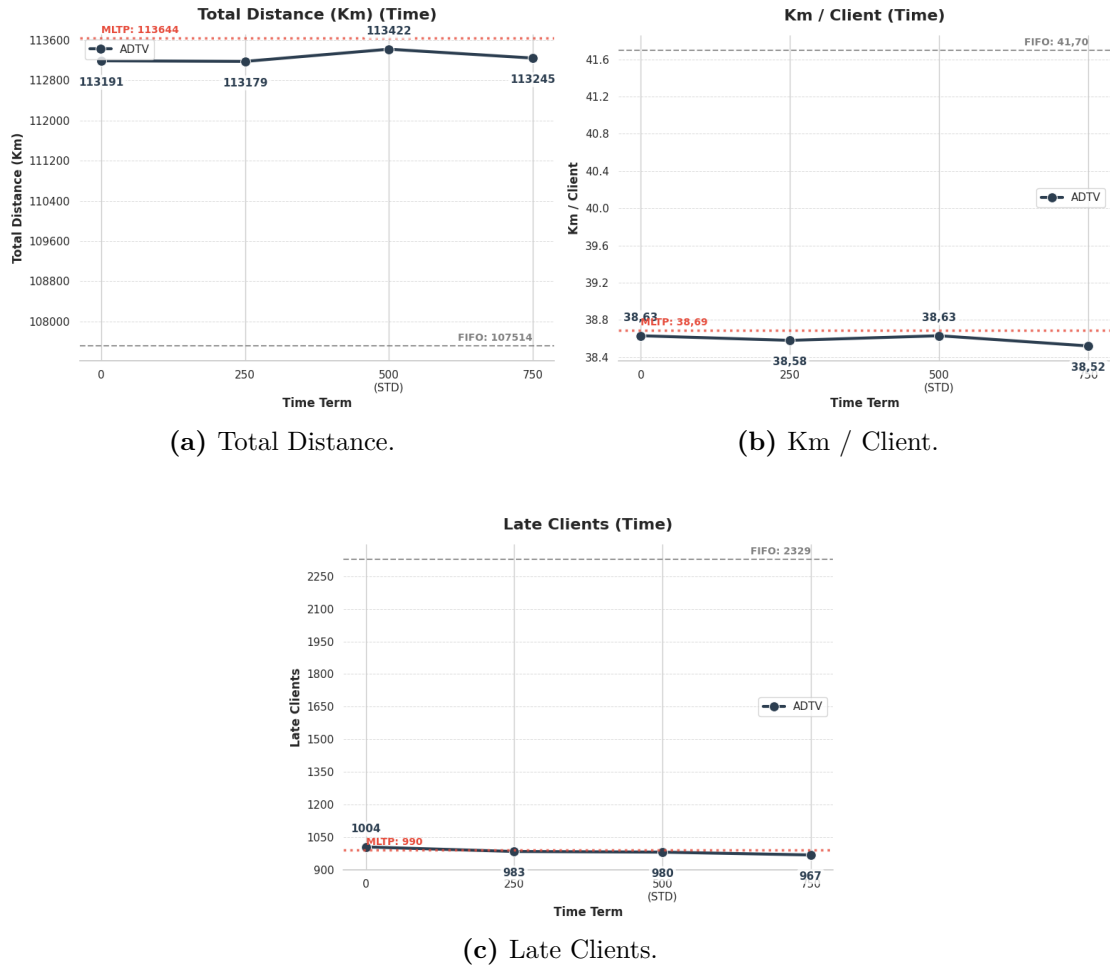
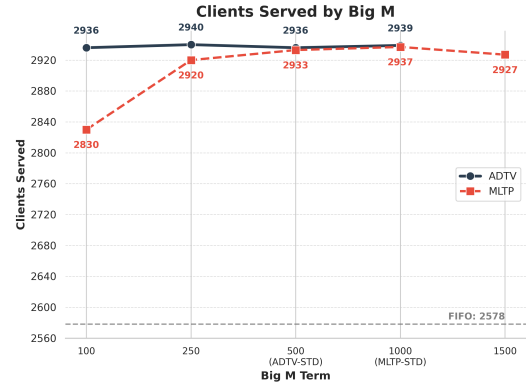


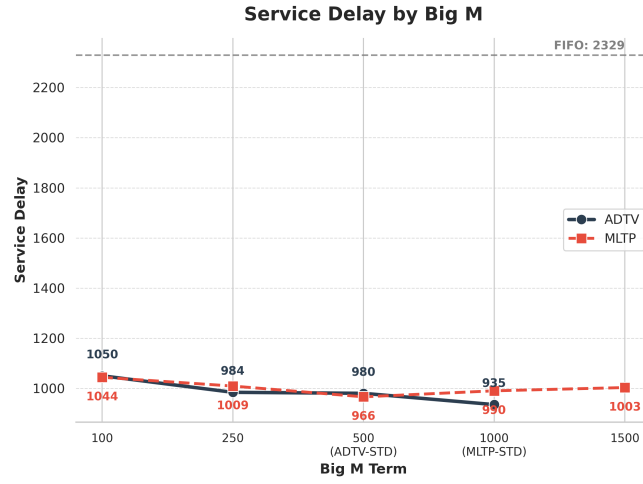
Figure 6.11: Performance evaluation on the real-world **Catanzaro** dataset varying the **Time** term. The solid blue line represents the ADTV policy. Baselines for FIFO and standard MLTP are reported.



(a) Routing efficiency (Km/Client).



(b) Total clients served.



(c) Service Delay (Late Clients).

Figure 6.12: Performance comparison between ADTV and MLTP policies over variable **Big M** penalties on the **Catanzaro** dataset. Solid dark blue lines represent the proposed ADTV policy, while dashed crimson lines indicate the MLTP standard. The horizontal line establishes the FIFO baseline.

Table 6.28: Performance comparison in the Catanzaro set, fixed windows, variable big M term.

Policy	Catanzaro - Big M					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	107514	2578	41,7	4,71	5,27	2329
MLTP - STD	113644	2937	38,69	-2,72	4,26	990
MLTP - M 100	107108	2830	37,85	-2,54	4,06	1044
MLTP - M 250	113069	2920	38,72	-2,65	4,19	1009
MLTP - M 500	114042	2933	38,88	-2,72	4,42	966
MLTP - M 1500	113336	2927	38,72	-2,66	4,22	1003
ADTV - STD	113422	2936	38,63	-2,69	4,38	980
ADTV - M 100	113190	2936	38,55	-2,34	4,75	1050
ADTV - M 250	113452	2940	38,59	-2,62	4,36	984
ADTV - M 1000	114060	2939	38,81	-2,71	4,5	935

6.2.2 Lainate depot

The Lainate data has different characteristics: the great majority of orders are very close to the depot, with most of the further at around 30 min, and very few at 1h 30min travel time. Moreover, the percentage of orders with expired due date is lower, equal to 33,7%, that is 406 out of 1205 serviceable orders in the first day of the simulation (before splitting). The total number of orders is around 7157. Since we have much more vehicles in this context (80 at the start, and then 72 after a week to simulate the business dynamics), we get rid of the backlog in a few days.

Standard values are $score_const = 10$, $time_const = 50$, $prob_const = 50$ and $big_M = 250$ for the additive prizes, while $big_M = 100$ in the multiplicative ones. These values are lower than in the Catanzaro case, as distances here are much shorter.

In this context, the results suggest a behavior closer to the synthetic instances. The score term is necessary to balance the attractiveness of clients: a value of zero decreases the number of customers while increasing delay, and higher values increase delay and worsen efficiency (Fig. 6.13): this last result is due to the fact that we have proportionally fewer clients late than in the synthetic instances, where we could optimize distances when prizes were leveled. There is a little spike in tardiness for the standard value: in the other cases we serve the same clients late, plus other but with lower tardiness, so the average decreases.

The probability term has little impact here, as clusters are very dense and distances are much smaller: it has better efficiency performance and fewer late served clients, although not by a very significant margin (Fig. 6.14).

Contrarily, urgency plays a more important role: it improves significantly efficiency, total distance and delay, at the cost of poorer tardiness performance (Fig. 6.15), as prizes of late clients become relatively less attractive.

Finally, changing big M values does not produce significant changes for the additive policy, while higher values worsen distance and efficiency for the multiplicative strategy. ADTV beats MLTP in these metrics, with similar tardiness but a lower number of clients served late.

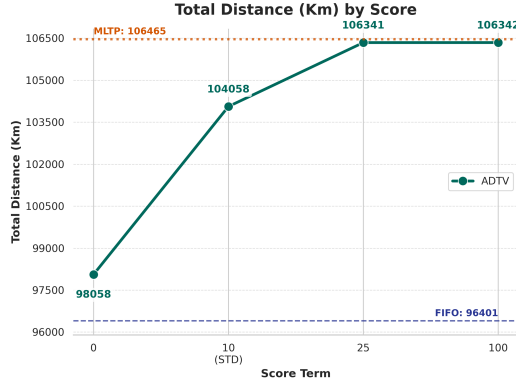
Lainate’s characteristics, therefore, produce different insights than Catanzaro and synthetic instances. When a large number of clients are packed in a small area, forecasting new client arrival does not lead to better routing decisions, and the important drivers are the score term and urgency.

Table 6.29: Performance comparison in the Lainate set, fixed windows, variable score term.

Policy	Lainate - Score					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	96401	6097	15,81	-0,81	3	2635
MLTP - STD	106465	7234	14,72	-4,29	2,63	482
ADTV - STD	104058	7242	14,37	-4,13	3,41	518
ADTV - S 0	98058	6895	14,35	-3,65	2,59	683
ADTV - S 25	106341	7267	14,63	-4,09	3,28	539
AADTV - S 100	106342	7285	14,6	-4,04	3,52	630

Table 6.30: Performance comparison in the Lainate set, fixed windows, variable probability term.

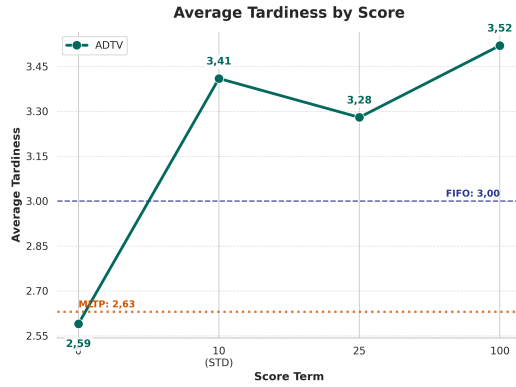
Policy	Lainate - Probabilities					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	96401	6097	15,81	-0,81	3	2635
MLTP - STD	106465	7234	14,72	-4,29	2,63	482
ADTV - STD	104058	7242	14,37	-4,13	3,41	518
ADTV - P 0	104034	7238	14,37	-4,07	3,38	534
ADTV - P 25	104158	7239	14,39	-4,1	3,16	526
ADTV - P 100	105810	7245	14,6	-4,13	3,27	537



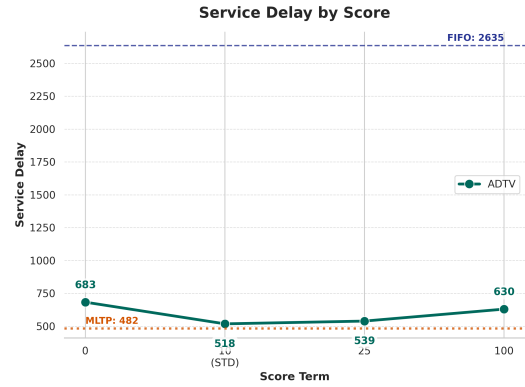
(a) Total distance driven.



(b) Total clients served.

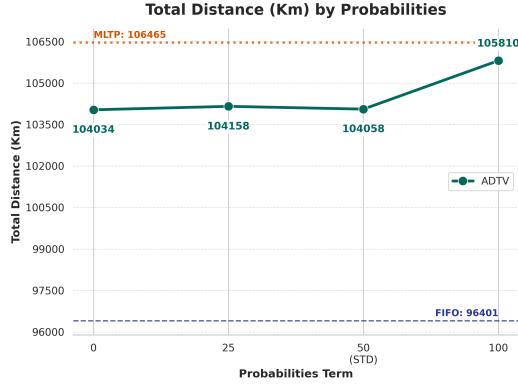


(c) Average Tardiness.

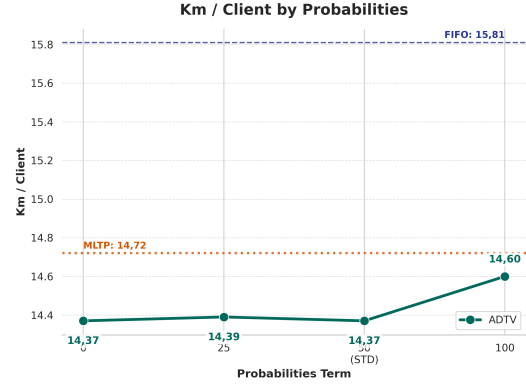


(d) Service Delay (Late Clients).

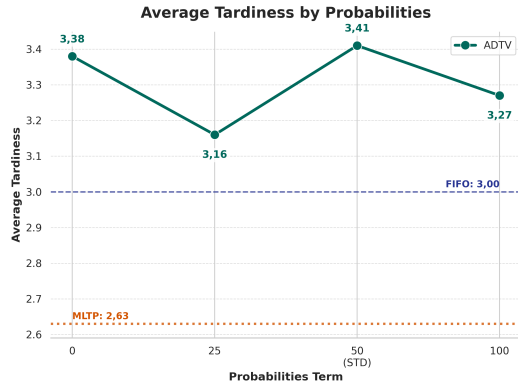
Figure 6.13: Performance analysis on the real-world **Lainate** dataset varying the **Score** term. The deep teal solid line denotes the proposed ADTV policy. Baselines for FIFO (indigo) and standard MLTP (orange) are represented by horizontal dashed and dotted lines, respectively.



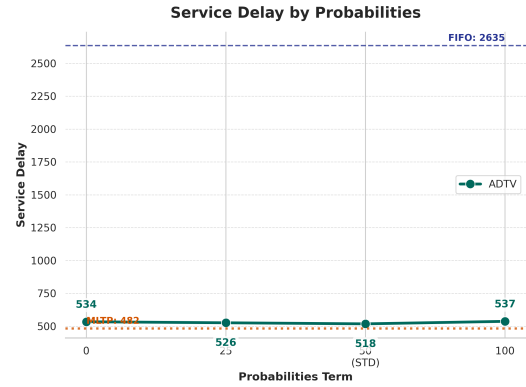
(a) Total distance driven.



(b) Routing efficiency (Km/Client).

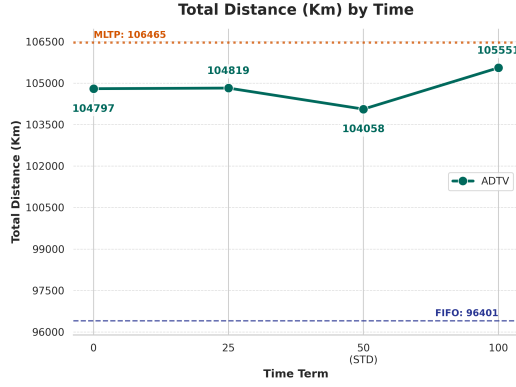


(c) Average Tardiness.

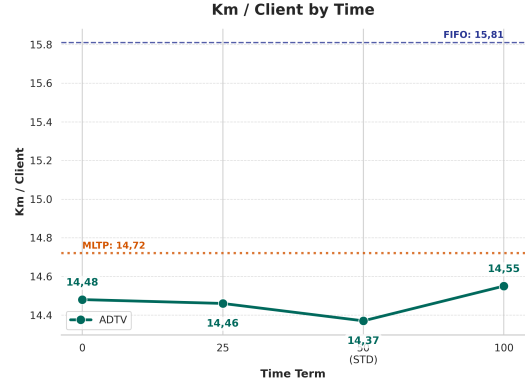


(d) Service Delay (Late Clients).

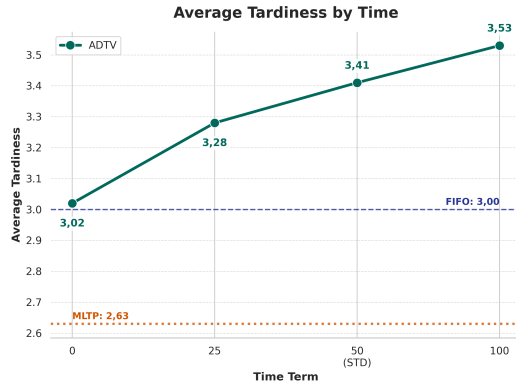
Figure 6.14: Performance analysis on the **Lainate** dataset varying the **Probability** term. The ADTV policy is represented by the deep teal line. Horizontal reference lines indicate the performance of the FIFO (indigo) and standard MLTP (orange) heuristics.



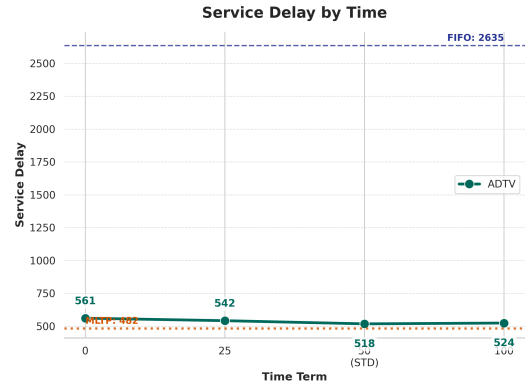
(a) Total distance driven.



(b) Routing efficiency (Km/Client).



(c) Average Tardiness.



(d) Service Delay (Late Clients).

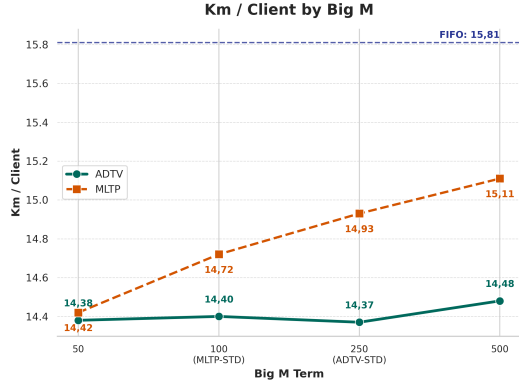
Figure 6.15: Performance analysis on the **Lainate** dataset varying the **Time** term. The ADTV policy is represented by the deep teal line. Horizontal reference lines indicate the performance of the FIFO (indigo) and standard MLTP (orange) heuristics.

Table 6.31: Performance comparison in the Lainate set, fixed windows, variable time term.

Policy	Lainate - Time					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	96401	6097	15,81	-0,81	3	2635
MLTP - STD	106465	7234	14,72	-4,29	2,63	482
ADTV - STD	104058	7242	14,37	-4,13	3,41	518
ADTV - T 0	104797	7237	14,48	-3,97	3,02	561
ADTV - T 25	104819	7249	14,46	-4,08	3,28	542
ADTV - T 100	105551	7252	14,55	-4,15	3,53	524

Table 6.32: Performance comparison in the Lainate set, fixed windows, variable big M term.

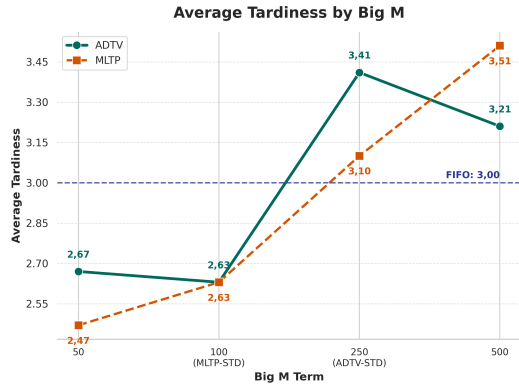
Policy	Lainate - Big M					
	Km	Clients	Km/clients	Lateness	Tardiness	Late clients
FIFO	96401	6097	15,81	-0,81	3	2635
MLTP - STD	106465	7234	14,72	-4,29	2,63	482
MLTP - M 50	103343	7166	14,42	-4,13	2,47	540
MLTP - M 250	108199	7246	14,93	-4,37	3,1	461
MLTP - M 500	109347	7238	15,11	-4,39	3,51	463
ADTV - STD	104058	7242	14,37	-4,13	3,41	518
ADTV - M 50	104221	7247	14,38	-4,03	2,67	605
ADTV - M 100	104346	7247	14,4	-4,1	2,63	541
ADTV - M 500	104840	7242	14,48	-4,09	3,21	530



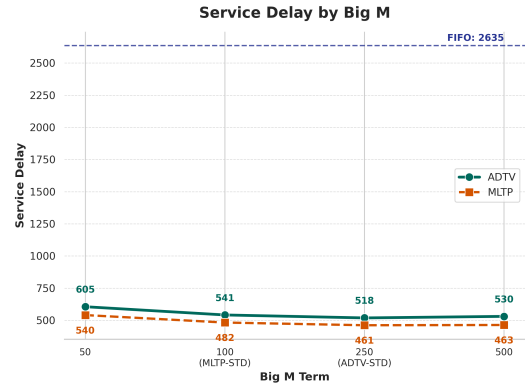
(a) Routing efficiency (Km/Client).



(b) Total clients served.



(c) Average Tardiness.



(d) Service Delay (Late Clients).

Figure 6.16: Performance comparison between ADTV and MLTP policies over variable **Big M** penalties on the **Lainate** dataset. Solid deep teal lines represent the proposed ADTV policy, while dashed orange lines indicate the MLTP standard. The horizontal indigo line establishes the FIFO baseline.

Chapter 7

Conclusions

This thesis tackled the Dynamic Stochastic Vehicle Routing Problem, where clients located on the map must be served starting from a depot by a fleet of vehicle, optimizing routing decisions over a time horizon. The problem is dynamic because information is gradually revealed with time, and stochastic due to the presence of uncertainty on service times, due dates, and clients' locations. We borrow the problem definition from a real-word business application, the context of Mondo Convenienza, one of Italy's largest furniture companies: we optimize service over 30 days, clients are characterized by a service time and can be visited in a range of days with fixed length, equal to 7 working days, and vehicles have 8 available working hours each day.

This formulation of the problem is inherently hard to solve: particularly, stochastic programming models and the dynamic programming approach are unable to tackle the dimensionality and computational requirements of the problem needed to describe stochasticity, even when approximate strategies are implemented. To circumvent the problem, industry practitioners solve deterministic look-ahead models that approximate the effect of future events in several ways, yielding less accurate but manageable models.

The parametric Cost Function Approximation approach, meant to bridge the gap between these worlds, is explored in this thesis. It implements parametric terms in a deterministic model to account for uncertainty, which are to be fine-tuned in a stochastic simulator of the real world, shifting the complexity of the problem from model solution to algorithmic optimization. In our case, the parametric terms took the form of prizes to be assigned to each client, borrowing from Albareda et al. ([1]), which solved a multiperiod VRP by defining the daily subproblems as Prize-Collecting Vehicle Routing Problems. Prizes signal the convenience of serving a client on that particular day: we expand on this idea by introducing new terms depending on relevant problem features, such as urgency, the probability of new orders from neighboring clients and a score term depending on the relevance

(service time) of the order.

We tested this framework on Mondo Convenienza’s data with fixed time-windows and on synthetic data, where also variable time-windows were considered. Our approach always improved on the benchmark heuristics on most metrics, proving to be effective and worth implementing.

Moreover, by fine-tuning the parameters, we were able to gain strong information on the relevant features of the problem. When we have orders well separated and distributed in the plane, as in the synthetic instances, the time factor is not relevant for high-quality solution. Considering possible future orders can improve routing decisions, particularly when we have tight due dates as in the variable time windows context. The most relevant factor is the score term, which levels the attractiveness of the orders by assigning higher prizes to more demanding orders that can be overlooked otherwise.

Additionally, if we have a long backlog, this approach is shadowed by the cascade effect of clients with surpassed due date, such that routing decisions are governed by the high prizes of these clients. In these cases, the effect of the parameters is negligible.

Finally, in a context with many orders packed in a small area, considering the probability of future orders is less relevant, while taking into account urgency can optimize routing decisions.

This strategy has proven to be promising and further research directions can be taken. CFA can be extended to other operations research contexts; new formulations of the Dynamic Stochastic VRP can be considered, for example with physical capacity constraints or online orders; and new prize formulation may be studied, where new features enter and give us insights on the nature of complex logistics problems.

Bibliography

- [1] Maria Albareda-Sambola, Elena Fernández, and Gilbert Laporte. «The dynamic multiperiod vehicle routing problem with probabilistic information». In: *Computers & Operations Research* 48 (2014), pp. 31–39. ISSN: 0305-0548. DOI: <https://doi.org/10.1016/j.cor.2014.02.010>. URL: <https://www.sciencedirect.com/science/article/pii/S0305054814000458> (cit. on pp. iii, 3, 5, 6, 61).
- [2] Min Wen, Jean-François Cordeau, Gilbert Laporte, and Jesper Larsen. «The dynamic multi-period vehicle routing problem». In: *Computers & Operations Research* 37.9 (2010), pp. 1615–1623. ISSN: 0305-0548. DOI: <https://doi.org/10.1016/j.cor.2009.12.002>. URL: <https://www.sciencedirect.com/science/article/pii/S0305054809003165> (cit. on pp. 5, 6).
- [3] Marius M. Solomon. «Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints». In: *Oper. Res.* 35 (1987), pp. 254–265. URL: <https://api.semanticscholar.org/CorpusID:15346313> (cit. on p. 6).
- [4] Raymond T. Perkins III and Warren B. Powell. *Stochastic Optimization with Parametric Cost Function Approximations*. 2017. arXiv: 1703.04644 [math.OC]. URL: <https://arxiv.org/abs/1703.04644> (cit. on p. 6).
- [5] Warren B Powell and Saeed Ghadimi. *The Parametric Cost Function Approximation: A new approach for multistage stochastic programming*. 2022. arXiv: 2201.00258 [math.OC]. URL: <https://arxiv.org/abs/2201.00258> (cit. on p. 6).
- [6] Saeed Ghadimi and Warren B. Powell. *Stochastic Search for a Parametric Cost Function Approximation: Energy storage with rolling forecasts*. 2022. arXiv: 2204.07317 [math.OC]. URL: <https://arxiv.org/abs/2204.07317> (cit. on p. 6).