MASTERARBEIT | MASTER'S THESIS

Titel | Title

A Heuristic Approach for the Time-Dependent Orienteering Problem with Moving Targets in a Maritime Environment

verfasst von | submitted by

Paulina Heine B.Sc.

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 977
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Business Analytics
Betreut von | Supervisor:	Univ.-Prof. Dr. Jan Fabian Ehmke

Contents

Acknowledgements

I would like to express my gratitude to my supervisor, Prof. Ehmke, and especially to Tom Bormann for the invaluable guidance, constructive feedback, and continuous support throughout the development of this thesis. I am more than thankful for your support throughout this thesis, and I could not have wished for better supervision. Your guidance not only helped me learn and grow academically, but also opened new perspectives and opportunities that have significantly shaped my professional path.

I would also like to thank my former colleagues at the university, especially Dr. Ninja Scherr, whose kindness, encouragement, and inspiring attitude have had a lasting impact on me. My appreciation also goes to the University of Vienna for providing the academic environment and resources that enabled this research, as well as for granting me the privilege to pursue my studies.

Finally, I would like to thank everyone who directly or indirectly contributed to this thesis: the researchers whose work laid the foundation for this study; my best friend, whose motivation and encouragement through countless Zoom sessions have accompanied me since our bachelor's studies; my partner, for his support, endless patience, and for always making me laugh even in the most stressful moments; and my parents, for their unconditional love, encouragement, and belief in me throughout my life.

Zusammenfassung

Diese Masterarbeit untersucht das zeitabhängige Orienteering-Problem mit beweglichen Zielen (TDOP-MT) im maritimen Kontext der Plastiksammlung im offenen Ozean. Angesichts der wachsenden Umweltbelastung durch Meeresmüll entwickelt die Arbeit heuristische Algorithmen zur effizienten Routenplanung von Sammelbooten, die treibende Plastik-Hotspots unter dem Einfluss von Meeresströmungen einsammeln. Das Problem wird als Orienteering-Problem modelliert, bei dem Sammelschiffe innerhalb eines begrenzten Zeithorizonts eine Teilmenge von Hotspots auswählen und ansteuern müssen, um den Gesamtwert der gesammelten Verschmutzung zu maximieren.

Die zentrale Herausforderung liegt in der Bewegung sowohl der Ziele als auch der Schiffe durch Ozeanströmungen, was zu einer zeitabhängigen Problemstruktur führt, bei der Reisezeiten nicht vorhersagbar sind. Der Kern der Arbeit liegt in der Entwicklung und systematischen Evaluation von Online-Routing-Heuristiken, um zu untersuchen, wie die Gewichtung zwischen Nähe und Wert bei der Zielauswahl die Performance beeinflusst. Zwei Algorithmen werden entwickelt: ein grundlegender Greedy-Algorithmus (OG) und eine erweiterte Variante mit Retargeting-Mechanismus (OGR).

Die zentrale Forschungsfrage ist, ob eine ausgewogene Gewichtungsstrategie beiden Extremfällen (reine Wert- oder Distanzfokussierung) überlegen ist und ob sich verallgemeinerbare Muster identifizieren lassen. Die Algorithmen-Parameter werden durch Offline-Kalibrierung mittels Grid-Search optimiert, um die optimalen Parameterwerte und deren Einfluss auf das Sammelergebnis zu untersuchen. Die Online-Ausführung erfolgt in einer zeitschrittbasierten Simulation unter Verwendung realer Strömungsdaten des Copernicus Marine Service.

Die Ergebnisse zeigen, dass ausgewogene Gewichtungsstrategien konsistent bessere Performance erzielen als Extremfälle und dass der Retargeting-Mechanismus die Anpassungsfähigkeit an dynamische Umgebungen signifikant verbessert. Die Arbeit liefert wichtige Erkenntnisse für die Gestaltung adaptiver Routenplanungssysteme in dynamischen maritimen Umgebungen und legt eine methodische Grundlage für zukünftige Forschung zur vollständig autonomen Parameteranpassung.

Schlüsselbegriffe: Orienteering-Problem; Zeitabhängige Optimierung; Online-Heuristik; Greedy-Algorithmus; Retargeting; Müllsammlung; Plastikverschmutzung; Operations Research

Abstract

This master's thesis examines the Time-Dependent Orienteering Problem with moving targets (TDOP-MT) in the maritime context of plastic collection in the open ocean. In view of the growing environmental impact of marine litter, the thesis develops heuristic algorithms for efficient route planning for collection boats that collect drifting plastic hotspots under the influence of ocean currents. The problem is modelled as an Orienteering Problem in which collection vessels must select and navigate to a subset of hotspots within a limited time horizon to maximize the total value of the pollution collected.

The central challenge lies in the movement of both the targets and the ships through ocean currents, resulting in a time-dependent problem structure where travel times are unpredictable. The core of the work lies in the development and systematic evaluation of online routing heuristics to investigate how the weighting between proximity and value in target selection influences performance. Two algorithms are developed: a basic greedy algorithm (OG) and an extended variant with a retargeting mechanism (OGR).

The central research question is whether a balanced weighting strategy is superior to both extreme cases (pure value or distance focus) and whether generalizable patterns can be identified. The algorithm parameters are optimized through offline calibration using grid search to investigate the optimal parameter values and their influence on the collection result. Online execution takes place in a time step based simulation using real current data from the Copernicus Marine Service.

The results show that balanced weighting strategies consistently achieve better performance than extreme cases and that the retargeting mechanism significantly improves adaptability to dynamic environments. The work provides important insights for the design of adaptive route planning systems in dynamic maritime environments and lays a methodological foundation for future research on fully autonomous parameter adaptation.

Keywords: Orienteering problem; time-dependent optimization; online heuristics; greedy algorithm; retargeting; waste collection; plastic pollution; operations research

1 Introduction

1.1 Background and Motivation

The Global Plastic Crisis

Based on the landmark study by Geyer, Jambeck, and Law (2017), the cumulative amount of plastic ever produced increased from roughly 2 million metric tons in 1950 to 8.3 billion metric tons by 2015. This rapid growth is forecasted to continue, with amounts reaching approximately 12 billion metric tons by 2050.

The high rate of plastic production is largely driven by consumer usage patterns, particularly the reliance on plastics for single-use and short-term applications, such as packaging and everyday products (Chen, Awasthi, Wei, Tan, and Li (2021)). Consequently, plastics are used frequently in daily life, but typically for only brief periods. This creates a paradox. The most commonly used plastics, such as polyethylene (PE) and polypropylene (PP), are designed for durability. However, they are used only for a short time, are quickly disposed of, and persist in the environment for a long time (Schnurr et al. (2018)).

Recycling this plastic is expensive and complex. As a result, 79% of all plastic waste has been discarded in landfills or the natural environment, while only 9% has been recycled (Geyer et al. (2017)). This mismatch between the material's properties and its intended use is a core driver of the global plastic crisis (Geyer et al. (2017)).

Upstream and Downstream Measures for Reducing Plastic

Two classes of strategies have been developed to address plastic pollution: upstream measures and downstream measures (Lau et al. (2020)). This distinction clarifies the approaches available to tackle the problem effectively.

Upstream measures aim to address the problem at the beginning, primarily by changing the behaviour of plastic users, who are companies and consumers. They propose a model for a circular economy, moving away from the current linear economy (Meijer, van Emmerik, van der Ent, Schmidt, and Lebreton (2021)). Central strategies include reducing plastic production, incentivising reusable alternatives, and establishing efficient recycling and circular systems. By limiting the inflow of new plastic into the market, these measures seek to cut pollution at its origin (Lau et al. (2020)).

Downstream measures, in turn, focus on the management of plastic that has already made its way into ecosystems. These interventions involve the physical removal of waste, particularly in heavily affected areas, through initiatives such as cleanup operations or river barrier systems. While they do not address the underlying causes of plastic pollution, downstream measures are essential for mitigating immediate ecological and environmental impacts associated with plastic waste (Lau et al. (2020)).

Although many current efforts are directed at upstream measures, it is important to recognize the need for both types of interventions. Lasting improvements depend on changes in consumer plastic usage, yet downstream measures are essential, since plastic will remain in the environment unless it is actively removed (MacLeod, Arp, Tekman, and Jahnke (2021)).

Why Ocean Cleanup Remains Crucial

Considering the diverse and far-reaching harms of plastic pollution, the importance of the interventions becomes even clearer. This is especially true for the focus area of this study, which is the ocean: Once released into the ocean, plastics fragment into smaller particles, including microplastics and nanoplastics, which persist for decades to centuries (Lv et al. (2024)). These particles enter marine food webs through ingestion by plankton, invertebrates, fish, and birds. This disrupts ecological cycles and poses severe risks to the ocean, leading to toxic effects across trophic levels and potential human health risks through seafood consumption and impacts on biodiversity (M. E. Miller, Hamann, and Kroon (2020); Tuuri and Leterme (2023)).

In addition, the physical presence of plastics in the ocean can have direct impacts on animals. Floating debris accumulates in gyres and coastal zones, leading to entanglement in larger debris and causing injury and mortality among marine mammals, seabirds, and turtles (Gregory (2009)).

Ocean plastic pollution is not only an environmental challenge but also a significant socioeconomic and public health issue. The economic damage is substantial; a UN report from 2021 estimates that in 2018 alone, the global tourism, fisheries, and aquaculture industries suffered losses between 6 billion (USD) and 19 billion (USD) due to marine plastic pollution (United Nations Environment Programme (UNEP) (2021)). Taken together, these facts highlight the need for comprehensive strategies that combine preventative, upstream measures with active cleanup efforts. While reducing future plastic production is critical for prevention, downstream mechanisms remain indispensable because they directly address the vast amount of existing pollution that has already accumulated and cannot be prevented through upstream action alone. This study, therefore, focuses specifically on downstream mechanisms for the removal of plastic from the open ocean.

Focusing on Ocean-Based Removal

When examining downstream methods in the ocean, it is important to note that most marine plastic originates from land and enters the sea primarily through rivers (Meijer et al. (2021)). To address this, downstream strategies can be split into two distinct parts: interception and retrieval. These approaches focus on different stages of the plastic's journey. Interception aims to capture plastic debris before it enters extensive marine ecosystems, while retrieval targets the plastic that has already dispersed into the ocean.

The first approach is to intercept plastic before it reaches the ocean, for instance, by installing barrier systems in rivers or at river mouths. This strategy is particularly effective because it prevents plastic leakage at a stage when it is still relatively concentrated and accessible, thereby reducing the load entering marine ecosystems.

The second approach targets plastic that has already reached the ocean. Despite ongoing efforts to reduce plastic inputs with downstream measures, a substantial volume of plastic has already accumulated in the marine environment (Geyer et al. (2017)). This pollution continues to circulate in large-scale ocean currents known as subtropical gyres, settle in sediments, and wash up on coastlines. Gyres are rotating systems of ocean currents driven by winds and the Coriolis effect. They create convergence zones where floating debris becomes trapped. There are five major gyres worldwide: the North Pacific, South Pacific, North Atlantic, South Atlantic, and Indian Ocean gyres. The North Pacific gyre, often called the "Great Pacific Garbage Patch," is the most studied. These systems act as long-term accumulation zones that concentrate plastic, which may persist for decades and pose significant threats to marine ecosystems far from the original sources of pollution (Lebreton et al. (2018)).

Addressing existing plastic in the ocean requires targeted removal strategies, including large-scale cleanup initiatives in ocean gyres. However, these interventions are technically challenging due to the dispersed and degraded nature of marine plastic.

Challenges of Marine Debris Collection

Retrieving plastic from the ocean is not straightforward. It might be assumed that collecting plastics, especially within subtropical gyres, would be easier because they accumulate in these convergence zones.

In reality, gyres do not contain a continuous mass of floating debris. Instead, they are vast accumulation zones covering millions of square kilometres, with plastics of different sizes and types dispersed throughout the water column. While larger items, such as fishing nets, crates, or packaging materials, can sometimes be found floating at the surface, most of the material consists of microplastics and fragmented particles. These are suspended at varying depths or mixed with plankton (Seeley, Nuss, and Smith (2024)).

Plastics in the open ocean also undergo continuous physical and chemical degradation. Solar radiation, wave activity, and mechanical abrasion quickly break larger items into micro- and even nano plastics. As particles become smaller, they mix more easily into deeper layers of the water column. This makes them increasingly difficult to detect or remove (Andrady (2011)).

This heterogeneity poses major challenges for cleanup operations because small fragments are harder to collect than large ones, making fast retrieval crucial. Moreover, ocean dynamics such as winds, waves, and currents constantly redistribute the particles, making accumulation zones highly dynamic. Most of the plastic is not directly detectable: microplastics and small fragments cannot be reliably tracked using satellites or conventional remote sensing technologies, making them effectively invisible to large-scale monitoring (Hu (2021)).

As a result, while gyres are accumulation zones of ocean plastic pollution, their diffuse and dynamic characteristics make large-scale removal complicated. The area of the GPGP spans over 1.6 million square kilometres (Lebreton et al. (2018)). Assuming a vessel travels at a steady speed of 10 knots (18.5 km/h or 11.5 mph), it would take approximately 68 hours, or nearly 3 days of non-stop travel, to cross the distance in a straight line. However, collection vessels often must drive slower, at 1.5 knots (approximately 2.78 km/h or 1.73 mph), to avoid disrupting fish and to ensure debris does not float around the collection net (den Hertog, Pauphilet, Pham, Sainte-Rose, and Song (2024)). At collection speed, it would take approximately 455 hours or 19 days of non-stop travel just to cross the GPGP in a straight line. Therefore, routing needs to be planned carefully.

These challenges show that large-scale removal of plastic from the ocean is complex, while important for the environment. Cleanup missions need to be carried out effectively. Optimization is therefore critical: through careful planning of collection routes, resource allocation, and operational timing, cleanup initiatives can substantially enhance their effectiveness and contribute more meaningfully to reducing ocean plastic pollution.

1.2 Problem

To address the challenge of oceanic plastic retrieval in a tractable and operationally relevant manner, this thesis adopts a simplified representation of marine plastic distribution. Rather than modelling debris as a continuous field, plastic pollution is abstracted into a set of small, discrete, high-density accumulation zones (“hotspots”) composed of a few pieces of debris, which are affected by ocean currents. This aggregation is an abstraction; however, it makes the problem solvable for the OP and transforms the complex challenge of cleaning a continuous debris field into a well-defined combinatorial optimization problem. Where each hotspot is characterized by a fixed value, reflecting its impact on the collection. The retrieval problem itself is formulated as a Time-Dependent Orienteering Problem with Moving Targets (TDOP-MT). A homogeneous fleet of collection vessels must navigate to maximize aggregate value by visiting a subset of hotspots within a finite operational window.

The vessels are also subject to current-induced drift, and both the vessels and the targets are repositioned at each time step based on oceanographic data. Thus, this problem is fundamentally time-dependent, as both the targets (plastic hotspots) and the collection vessels change their positions over time due to steering and ocean currents. At each time step, the spatial configuration of the environment evolves, altering the travel time between vessels and targets. This temporal dependence makes the problem significantly more complex than the classical Orienteering Problem, which assumes static targets and fixed travel costs. In contrast, the spatio-temporal nature of this problem requires routing strategies that adapt continuously to the evolving state of the environment. Importantly, the developed heuristics themselves do not require prior knowledge of future hotspot trajectories. Instead, routing decisions must be made reactively based on recent information, continuously adapting to the current positions of the targets. However, since the algorithms rely on weighting parameters, these have to be tuned prior to the actual optimization of the visiting order of hotspots. This tuning is done via a grid search, which relies on prior knowledge about trajectories, as it involves multiple simulations. However, since parameter tuning is not the focus of this thesis, it is not within the scope of the problem; instead, this thesis examines how routing can be done in the OP framework with moving targets and how it changes under different weighting parameters.

1.3 Research Gap

Despite growing research on (time-dependent) Orienteering Problems, there is still a notable gap regarding moving targets. While several studies have investigated moving targets in the context of the Travelling Salesman Problem (TSP), such formulations for the OP are rare.

Most existing approaches for the TDOP do not treat the movement of targets as the reason for time dependency; they rather build on static targets. Movement of targets is mostly discussed in other routing problems, like the Travelling Salesman Problem. However, the TSP differs from the OP in many ways. Also, trajectories are often assumed to be known in advance, focusing on complete offline rather than on online optimization. While this work includes heuristics that run online with offline-tuned parameters, it aims to uncover patterns that can guide the parametrization of algorithms, with the long-term goal of achieving fully online adaptability.

The lack of research on Orienteering Problems with Moving Targets is a significant gap in current optimization literature. This thesis aims to help close this gap by introducing and analysing an Orienteering Problem with Moving Targets, where both agents and targets evolve over time in a drifting marine environment.

1.4 Research Questions

This thesis addresses a time-dependent Orienteering problem with exogenous target and agent movement using a greedy heuristic with offline calibration and online execution. To address this gap, the following research questions are posed:

Research Question One: In the setting of the TDOP-MT, is it more beneficial to focus completely on value, on proximity or to adopt an intermediate balance between both, to maximise the overall collected value?

Research Question Two: Can generalizable insights about weighting parameters be derived regarding the optimal weighting strategy in order to omit the calibration phase in the future? Are there scenario-specific patterns that suggest how the selection process should be weighted under different conditions?

Research Question Three: How do the proposed online heuristics algorithms perform compared to an offline static baseline solution in terms of total collected value? Does the introduction of online decision-making and retargeting improve overall performance?

1.5 Assumptions

As this work is conducted within the scope of a master's thesis, a set of simplifying abstractions has been applied to enhance the model's manageability and reproducibility. Given the inherent complexity of the underlying problem, these simplifications are necessary to keep the implementation and analysis feasible. The applied assumptions are outlined in the following section.

Plastic Distribution and Hotspots Plastic debris is represented as discrete accumulation zones, referred to as hotspots, rather than as a continuous distribution. The model does not allow the formation of new hotspots during its execution. Plastic is assumed to be present exclusively on the surface, resulting in a two-dimensional environmental model.

Environmental Dynamics and Movement Modelling The model uses only ocean surface current data to simulate movement. It excludes other environmental factors to keep the model simple and reproducible. All trajectories evolve deterministically based on the input current field and object-specific drift factors to ensure reproducibility.

Fleet Characteristics and Vessel Behaviour The fleet is assumed homogeneous; all vessels have identical movement capabilities and behaviour. Operational constraints such as fuel consumption or loading capacity are excluded. This keeps the analysis focused on route-finding under time constraints and maintains a manageable algorithmic complexity for a master's thesis.

Routing and Target Selection Logic The vessels do not predict future hotspot trajectories and make routing decisions only on the location information of the present time step. Minimizing travel distance is not a goal and is not achievable, as vessels move at their assigned speed throughout, and they cannot stand still. Thus, the travel distance is influenced solely by the time horizon and current fields, which are neither avoided nor preferred, as the route between a present vessel location and a target hotspot is not subject to optimization, only the selection of the target.

Parameter Calibration All external optimization parameters are determined via a grid search prior to the actual execution of the heuristic algorithms. The development of a sophisticated parameter optimization method is not within the scope of this thesis. Instead, this work includes an evaluation of the influence of these parameters in order to lay the foundation for future research, in which they can be determined in a more advanced manner.

1.6 Structure of the Thesis

The thesis is structured to provide a clear progression from foundational context to methodological development, empirical analysis, and discussion.

The introduction presents the motivational background, addresses marine plastic pollution and cleanup, states the research gap and questions, and outlines the main modelling assumptions.

The literature review builds directly on the introductory context, establishing the theoretical foundation for this study by providing an overview of the Orienteering Problem (OP). It begins by exploring the classical problem before focusing on the version most relevant to this work: the time-dependent OP together with an analysis of how moving targets have been handled in related routing problems. To address the real-time nature of the problem, the review differentiates between online and offline optimization frameworks. Finally, the chapter concludes with a state-of-the-art summary of current approaches to plastic cleanup missions in the ocean.

Transitioning from theory to methodology, the following Methodology chapter details the analytical framework of this study.
The chapter begins with an overview over the used notation. It then defines the core Model Components: the drifting plastic hotspots, the navigating vessels and how they interact with each other. In the following section, the structure of the underlying graph is explained. Building on this, the formal Mathematical Optimization Model Formulation, which defines the problem's notation, variables, constraints will be presented. It follows the core of the methodology is the section on Algorithms, it introduces the static Baseline solutions (B1 and B2). It then describes the two dynamic heuristics central to this work: the Online Greedy Algorithm (OG) and its enhanced version with a retargeting mechanism (OGR). The chapter concludes with a comparison and an overview over the whole system.

The subsequent sections moves to the experimental procedures in the Testing and Discussion chapter. This builds on the defined methodology and outlines the simulation design for scenario settings. Four main experiments are conducted: Test 0 investigates the influence of different scoring weights on performance; Test 1 analyses the optimal weighting factors across different environmental scenarios; Test 2 compares the proposed heuristic approaches with the static baseline in terms of objective value; Test 3 focuses on computational runtimes; and Test 4 examines the generalizability of averaged parameter settings and their influence on solution quality.

To extend the findings beyond simulation, the real-world relevance chapter presents use-cases perspective and assesses the practical implications of the proposed approach for operational deployment. Finally, drawing together the insights, the summary and outlook summarize the key findings, highlight the methodological contributions of this thesis, and propose directions for future research.

2 Literature

This chapter provides a review of the literature foundational to this thesis. To offer a structural overview, the following mind map illustrates the key research areas and their interconnections, serving as a visual guide for the topics discussed. The specific research gap this thesis aims to address, ocean cleanup missions as an Orienteering Problem (OP) with moving targets solved online, is highlighted in the bottom right corner, because there is no direct connection between these main topics.

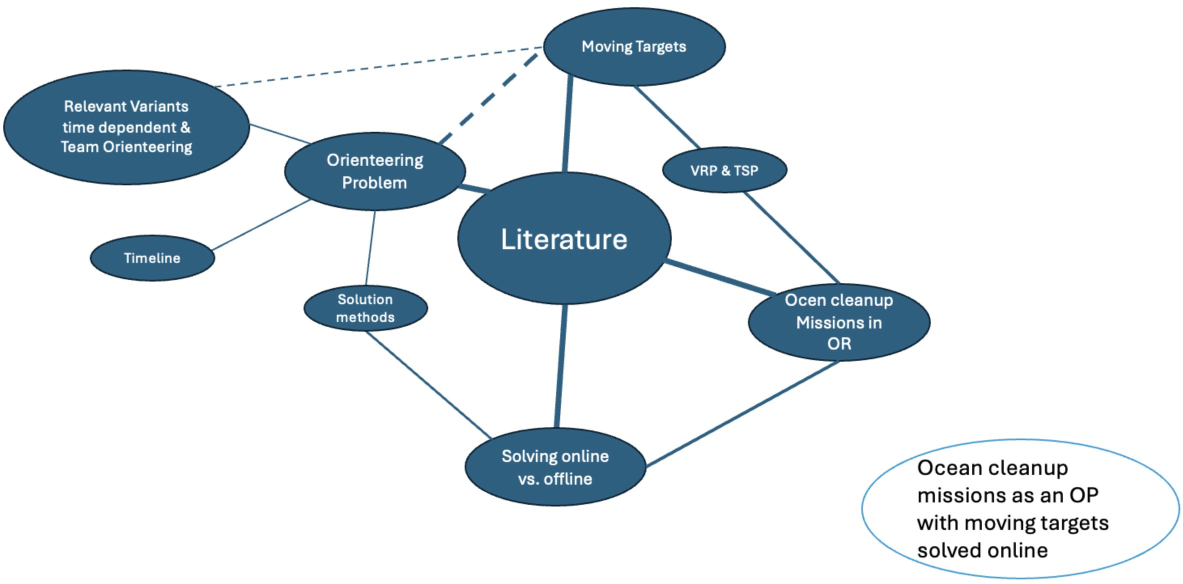

Figure 2.1: Mind map literature overview

First, there will be a review of the OP in general and its timeline, as this forms the most important framework. Building on this foundation, a more specific analysis of the most relevant variant of the OP, the time-dependent one, will follow. This variant will be distinguished from moving targets in general. The following section will then clearly define the distinction between online and offline solution frameworks for the problem, emphasizing how online frameworks adapt to information as it becomes available, whereas offline frameworks work with all information known in advance. After the discussion of frameworks, an overview of the state of the art in marine debris collection will be provided to contextualize the problem within real-world applications and the current research landscape.

2.1 The Orienteering Problem

The Orienteering Problem (OP) is a combinatorial optimization problem. It combines key aspects of the Knapsack Problem (KP) and the Travelling Salesperson Problem (TSP) (Vansteenwegen, Souffriau, and Van Oudheusden (2011)). Figure 2.2 visually summarizes this relationship. It incorporates the routing and tour construction of the TSP (green) and the selection criteria of the KP (blue), and merges these features to model the OP. Where the objective is to maximize collected rewards within resource constraints.

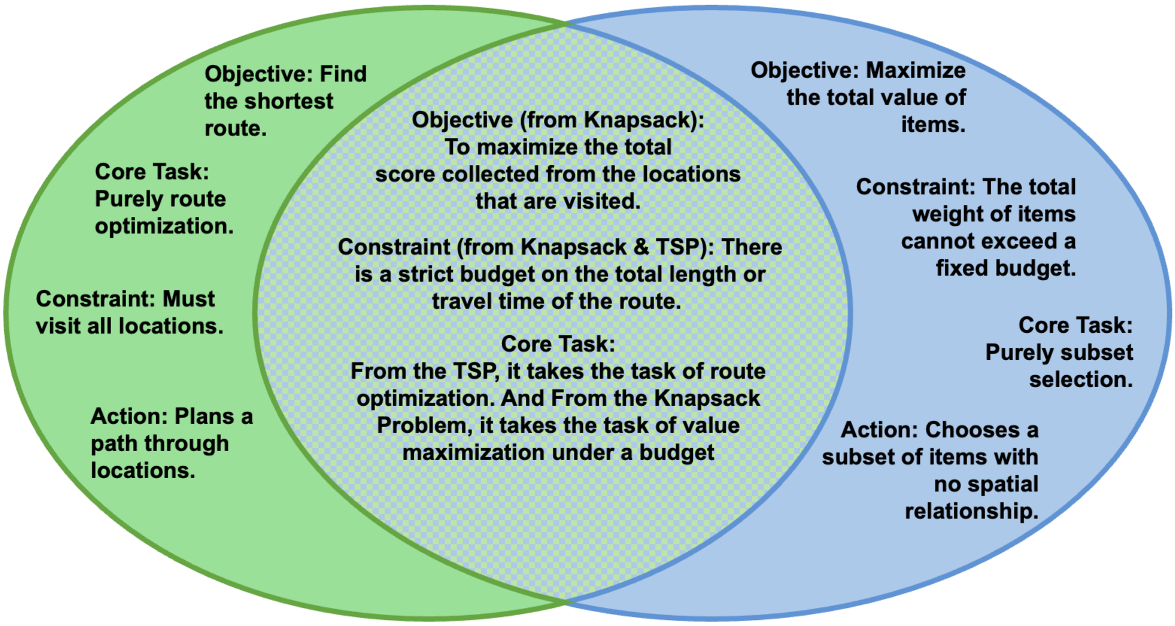

Figure 2.2: Orienteering Problem: A mix of TSP (Gavish and Graves (2004)) and KSP (Böckenhauer et al. (2026))

The Orienteering Problem (OP) originates from the sport of Orienteering. In this sport, participants start at a fixed point and try to visit as many checkpoints as possible; each has an assigned score. The goal is to maximize the total score within given resource limits (Tsiligiridis (1984),Chao, Golden, and Wasil (1996a)). Unlike the Travelling Salesperson Problem (TSP), the OP does not require visiting all nodes. This adds complexity by introducing another selection component (Schmid and Ehmke (2017)). The following section outlines key milestones in the development of the OP.

The OP was introduced by Tsiligiridis (1984) in 1984, who also proposed the first stochastic and deterministic solution heuristics. In 1987, Golden, Levy, and Vohra (1987) proved the OP is NP-hard. Around 1990, exact methods for small cases emerged, including branch-and-bound approaches for fewer than 20 nodes by (Laporte and Martello (1990)).

Building on this progress, the first OP variants appeared, increasing the realism of the problem. In 1992 (Kantor and Rosenwein (1992)), the Orienteering Problem with Time Windows was solved, illustrating how visits could be constrained by specific timing requirements. Further expanding the OP, the concept of using multiple vehicles was introduced as the Team Orienteering Problem by (Chao, Golden, and Wasil (1996b))in 1996. The same year, (Chao et al. (1996a)) published a five-step heuristic, which outperformed all previously proposed heuristics for the OP.

Another quality improvement in solution algorithms occurred in 1998. That year, Fischetti, González, and Toth (1998) introduced Branch-and-Cut algorithms, enabling the optimal solution of OP instances with up to 500 nodes. Building on these advances, Butt and Ryan (1999) developed the first exact algorithm for the TOP in 1999 using column generation. This new algorithm was capable of solving problems with up to 100 vertices, though the path lengths had to be small. Then, Fomin and Lingas (2002) considered a time-dependent version of the OP (TDO), in which travel time depends on the departure time at a specific node.

In the following years, significant effort was directed toward solving the constrained variants and optimizing solution speed for real-world applications like tourist trip design. By the end of the 2000s, the Orienteering Problem had become a well-established topic in the operations research literature. It was widely recognized both as a theoretically rich problem and as a practically relevant model. All the previous research effort was summarized in a comprehensive survey of Vansteenwegen et al. (2011). This survey summarized decades of research and positioned the OP as a foundational problem with many extensions and real-world applications. In 2016, Gunawan, Lau, and Vansteenwegen (2016) extended this survey in the work on "A Survey of Recent Variants, Solution Approaches and Applications". In this work, the authors also emphasize the relevance of the OP: "The Orienteering Problem (OP) has received a lot of attention in the past few decades."(Gunawan et al. (2016)).

Despite this extensive body of work, certain questions remain unanswered, especially those concerning how to integrate movement factors. By addressing these challenges, this thesis takes the next logical step, aiming to provide solutions and offer further insights into the evolving landscape of the Orienteering Problem.

In summary, the OP is a relatively recent but significant research topic due to its complexity and practical relevance. Its formulation is central to this thesis, as it models problems involving value-assigned targets and the selection of a subset to visit. Among its variants, the time-dependent OP is the most relevant version of the OP here. In this version, travel times depend on previous decisions and affect departure times at each node (Fomin and Lingas (2002)). The next section provides a detailed literature review of this variant.

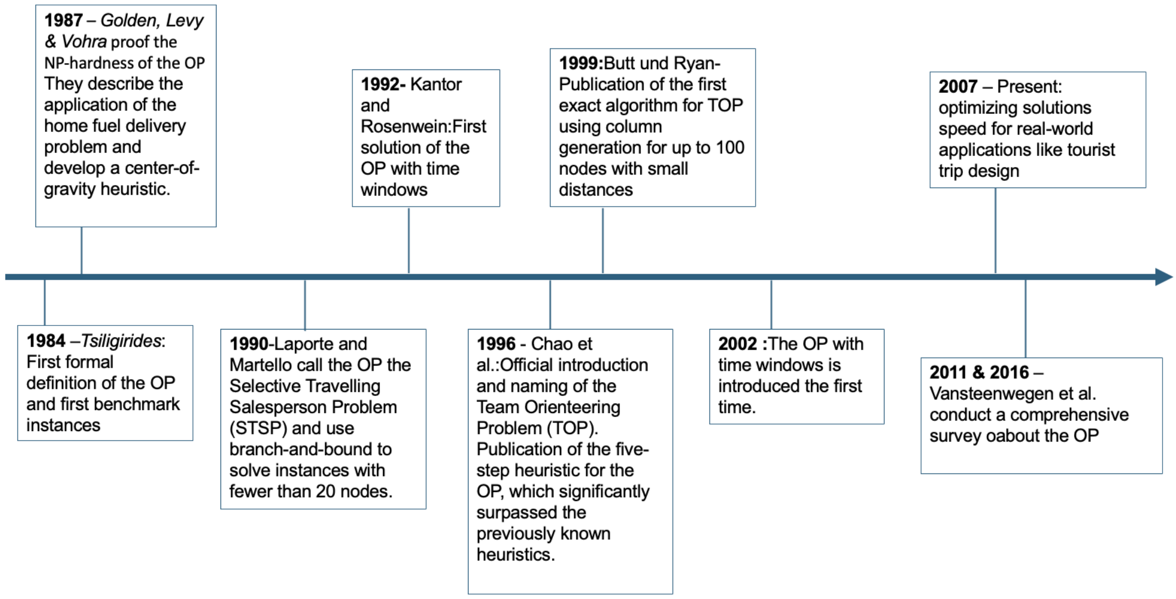

Figure 2.3: Timeline of the Orienteering Problem.

2.2 Key Dimensions of the presented Routing Problem

2.2.1 Time-Dependency and Moving Targets

The time-dependent Orienteering Problem (TDOP) can involve moving targets, directly affecting travel times. However, time dependency in general does not necessarily mean that the targets themselves are moving; it can also result from factors such as traffic congestion. In most studies, time dependency refers to changing traffic conditions rather than moving targets. A paper which addresses time dependency in an OP because of the ocean current is Mansfield, Macharet, and Hsieh (2023). Here the authors study the case where travel times are time-dependent between nodes. However, targets remain static, as typical for most time-dependent OPs.

One of the few papers that explicitly mentions moving targets as a cause of time dependency in the context of the Orienteering Problem is "Approximation Algorithms for time-dependent Orienteering" by Fomin and Lingas (2002). The authors discuss solutions for time-dependent Orienteering and mentions one of the applications could be on a kinetic TSP. This kinetic TSP is described by Hammar and Nilsson (1999), the authors assume targets to be moving. Fomin and Lingas (2002) algorithm divides the total available travel time into small, equal time slots and greedily selects the maximum possible number of new, unvisited locations in each of these slots before carefully connecting all of these short partial trips into a single, continuous route within the total time limit. The authors assume to know the function governing how long each journey will take depending on the start time is known to the algorithm for the entire planned duration, enabling it to calculate and plan the optimal route entirely in advance. This is a difference to this study, as there is no such function present in this work.

Besides Hammar and Nilsson (1999), most research addressing moving targets is found in the context of the Travelling Salesperson Problem (TSP). While both the time-dependent Orienteering Problem (TDOP) and the Moving-Target Travelling Salesperson Problem (MT-TSP) incorporate temporal dynamics, their formulations differ fundamentally. In the TDOP, targets are typically static, and time dependency arises from changing factors like traffic that affect travel times. In contrast, MT-TSP explicitly incorporates moving targets that must all be visited, with their movement directly part of the problem structure. Therefore, while OP time dependency typically refers to variable travel times, MT-TSP centres on the challenge of visiting all moving targets. The explicit study of OPs with moving targets remains absent. This lack of research on OPs with moving targets is understandable, as OP mainly applies to stationary-target contexts. Studies often focus on applications like transportation, theme park navigation, or tourist trip planning, and commonly assume static targets (Ali Abbaspour and Samadzadegan (2011); Garcia, Arbelaitz, Vansteenwegen, Souffriau, and Linaza (2010); Garcia, Vansteenwegen, Arbelaitz, Souffriau, and Linaza (2013); Gavalas, Konstantopoulos, Mastakas, Pantziou, and Vathis (2014); Gunawan et al. (2016); Verbeeck, Sörensen, Aghezzaf, and Vansteenwegen (2014)).

This poses an interesting research gap, since moving targets represent a realistic and challenging aspect of many real-world applications and deserve more attention in the context of the Orienteering Problem. However, there is a lot of research for the moving target TSP in general. Although this research on moving targets focuses on TSPs, the literature on these problems remains relevant for this work and will be discussed in the next paragraph.

The MT-TSP was introduced by Helvig, Robins, and Zelikovsky (2003), who demonstrated its NP-hardness and provided exact algorithms for one-dimensional cases in which targets move at constant velocity. Since then, the problem has evolved under various application-driven scenarios. For example, Choubey (2013) approached the Moving Target TSP with a Genetic algorithm. However, the distances, speeds, and directions of movement are predetermined. In more recent work, researchers have continued to push the boundaries of moving-target modelling. Philip, Ren, Rathinam, and Choset (2024) proposed a mixed-integer conic programming formulation based on space-time graphs of convex sets, while Bhat et al. (2024) introduced the first complete algorithm for the MT-TSP with obstacles. These contributions were further extended to multi-agent scenarios by Philip, Ren, Rathinam, and Choset (2025), who presented the Multi-Agent Moving-Target TSP, and by Bhat et al. (2025), who proposed parallel algorithms that are asymptotically optimal for generic moving-target patterns. Stieber and Fügenschuh (2022) investigated the Multiple Travelling Salesmen Problem with moving targets in order to find out which formulation solves the problem most efficiently when using exact methods.

Despite these advances, all of the above contributions are built for the MT-TSP framework or its related problem classes, such as pursuit-evasion or dynamic vehicle routing, where all targets must be visited. In contrast, the Orienteering Problem (OP) is characterized by the ability to select which targets to visit (Schmid and Ehmke (2017)) and has been formulated with static target locations. The selection component of the OP, combined with the challenge of targets potentially moving and the resulting time dependency, forms a relevant area that deserves more research attention.

Bridging the two fields of research, the moving targets of the TSP with the time-dependent OP, by introducing a selective routing problem with changing target positions, represents a contribution that extends beyond the scope of existing models.

One important factor when it comes to moving targets and time dependency in general is that information will change during the execution. In this thesis, only the recent location of targets is available at each timestep, which makes it impossible to predict future locations and travel times due to the lack of information about future and past locations. This changing information makes it difficult to fully precomputed a route over the entire time horizon. Thus, the route needs to be computed online. The difference between online and offline computed routes will be explained in the following passage.

2.2.2 Online vs. Offline Planning

There are two main ways to compute a route: online or offline. In offline approaches, routes are precomputed before execution based on knowledge of the environment and all relevant data. In online approaches, information is received and processed during optimization, and decisions are continuously updated as new information becomes available throughout execution. Especially in dynamic or moving target environments, where information changes over time, online approaches are important (Ritzinger, Puchinger, and Hartl (2016)). This changing of the environment is also present in this work, which makes online approaches especially important. The following passage will explain online and offline solutions and sort relevant routing literature into online and offline approaches.

The classical Moving-Target Travelling Salesman Problem, introduced by Helvig et al. (2003), assumes full availability of all input data, thus solving the problem offline. Specifically, the initial positions and constant velocity vectors of all targets are known in advance, allowing the pursuer to compute a complete interception tour before execution. This formulation enables exact planning along deterministic trajectories but requires full foresight and centralized computation. In such problems where full availability of the trajectories is known in advance, the problem is to precomputed a route where the agent intercepts with the target at a good time and place.

The previously named time-dependent OP in ocean currents from Mansfield et al. (2023) also assumes full availability of all data, thus solving the route offline. Even in dynamic environments, such as changing ocean currents, these dynamics are modelled predicatively and integrated into the planning phase. The aim is to compute a globally optimal or near-optimal route in advance. Many studies (i.e.: Abdollahzadeh, Javadi, Toragay, Epicoco, and Khodadadi (2024); Duan, Fan, Chen, Chen, and Ma (2021))in marine environments follow this structure: first, debris drift is forecasted, and then time windows for feasible collection are derived. This is followed by solving a routing problem with time constraints using algorithms like Adaptive Large Neighbourhood Search.

Another practical example of offline planning in a maritime environment is the routing framework by den Hertog et al. (2024). The problem in their work is modelled as a longest-path problem in a directed acyclic graph, where edge weights (representing collected plastic) depend on path history. Planning is performed over a seven-day forecast horizon and re-executed daily in a rolling-horizon scheme using updated forecasts.

In contrast, this work uses an online, reactive approach for the implemented heuristics. Although target movement could, in principle, be predicted from ocean currents using past or future data, the algorithm here relies only on present target positions at each time step. No future trajectories are planned, and no global route is computed in advance. Vessel assignments and movements are updated iteratively with the most recent environmental data.

If a problem is dynamic and solved offline, either there is full availability of data or the data is predicted. In contrast, online planning adapts routes in real time as new or unexpected information becomes available, such as emerging debris hotspots or weather changes. These algorithms emphasize adaptability over optimality.

Online versions of routing models have also been explored in other domains than the maritime. Shiri, Akbari, and Hassanzadeh (2024) analyses scenarios where service times or rewards become known only upon arrival at a site, requiring immediate route updates. Bertsimas, Jaillet, and Martin (2019) investigates Online vehicle routing with sequential demands, however targets are not moving here. The same holds for Jaillet and Wagner (2008), here the authors studied generalized online routing, again with stationary targets.

In summary, the choice between offline and online planning depends on the mission context and the availability of predictive information. Offline approaches are suitable for stable, predictable environments, but can be fragile when unexpected changes occur. Online methods, in turn, provide responsiveness but may yield suboptimal long-term results due to their reactive nature. Since the locations in the problem of this thesis are changing, a system that focuses on reactivity might be beneficial and needs to be fast to handle the routing.

In the previous sections, it was explained what the OP is and that this is the modelling framework of this thesis. Then, the most relevant variant of the OP, the time-dependent OP, was presented, and it was noted that the special case of moving targets is primarily addressed by another modelling framework, the TSP. Lastly, the difference between online and offline optimization was described, and why this work will focus on online optimization. To illustrate how no paper so far has concentrated on the overlap of these sections (OP, Moving Targets, Online), see this table:

It can be observed that there is only one paper that references an OP with moving targets. In the paper by Fomin and Lingas (2002), the kinetic TSP is identified as a potential problem

Table 2.1: Literature Overview

Reference	Problem	MT	TK	Online?
Helvig et al. (2003)	Moving-Target TSP	Yes	Yes	No
Stieber and Fügenschuh (2022)	Multiple TSP with Moving Targets	Yes	Yes	No
Bhat et al. (2024)	Moving Target TSP with Obstacles	Yes	Yes	No
Bertsimas et al. (2019)	Online vehicle routing with sequential demands	No	–	Yes
Duan et al. (2021)	Marine Debris Collection Routing (Vessel Routing)	Yes	predicted	No
Choubey (2013)	Moving Target Travelling Salesman Problem	Yes	Yes	No
Hammar and Nilsson (1999)	Kinetic Variants of TSP	Yes	Yes	No
Abdollahzadeh et al. (2024)	VRP	Yes	predicted	No
Jaillet and Wagner (2008)	Generalized Online Routing	No	–	Yes
Mansfield et al. (2023)	Time-Dependent OP	No	–	No
Shiri et al. (2024)	Capacitated Team Orienteering Problem	No	–	Yes
Fomin and Lingas (2002)	Time-Dependent Orienteering	Both	Yes	No
Tsiligiridis (1984)	Orienteering Problem (OP)	No	–	No

MT = Moving Targets explicitly named, TK = Trajectory known

for the developed algorithm. However, in this paper, an approximation for the travel time is provided, which is different from the study here.

This section outlined the key characteristics of the problem, it is a time-dependent Orienteering Problem with moving Targets, the last segment is the setting of the problem, which is ocean cleanup. Ocean cleanup research, however, extends beyond Operations Research and addresses multiple academic disciplines. The following passage identifies these additional fields.

2.3 State of the Art in Marine Debris Collection Routing

In the final section of the literature, the research on maritime plastic collection with a focus on the OR context will be explored shortly, clarifying where to position the research conducted in this thesis. The research on maritime plastic collection combines multiple disciplines and can be grouped into four parts: understanding the nature of the problem, developing theoretical planning models, implementing practical systems, and addressing ecological considerations. Only the second can be grouped in the OR landscape; however, the others will also be explained very shortly.

Understanding the Nature of the Problem Before any collection mission can be planned, it is important to understand where plastic accumulates and how it moves. Several studies have focused on this foundational aspect by analysing ocean currents, identifying accumulation zones, and investigating the composition of marine debris. For instance, Maximenko, Hafner, and Niiler (2012) used satellite-tracked Lagrangian drifters to simulate the drift of marine waste and identified five major accumulation zones located in subtropical gyres. One of the best-known of these zones is the Great Pacific Garbage Patch (GPGP). The existence and location of all five zones were validated through direct surface sampling of microplastics. Lebreton et al. (2018) confirmed elevated plastic concentrations in the GPGP.

Modelling the movement of plastic requires accurate oceanographic models. One such model is provided by the Copernicus Marine Service (CMEMS) Service (2024), delivering high-resolution data on ocean surface currents. This data will also be used in this study.

Regarding the scale and type of waste, Lebreton et al. (2018) found that the GPGP contains at least 79,000 tonnes of plastic spread over an area of 1.6 million km^2, which is four to sixteen times more than previous estimates. Over three-quarters of this mass consists of items larger than 5 cm, with fishing gear accounting for at least 46%. Microplastics make up 94% of the estimated 1.8 trillion particles, but only 8% of the total mass.

Developing theoretical Planning Models Once the plastic hotspots are identified, the next challenge is to plan efficient vessel routes. This is where the research from this study will be important. Several mathematical optimization models adapted from classical logistics problems have been applied to this domain.

The state of the art in routing for marine debris collection is mostly characterized by offline, predictive planning frameworks. A common approach involves a two-step process: first, forecasting debris movement using oceanographic models like the General NOAA Operational Modelling Environment (GNOME), and second, formulating and solving a routing problem, often with time constraints, based on these predictions. Seminal works by Duan et al. (2021) and Abdollahzadeh et al. (2024) use this paradigm, employing sophisticated algorithms such as Adaptive Large Neighbourhood Search to optimize collection routes based on pre-computed drift scenarios.

A prominent real-world application of this offline strategy is the framework developed by den Hertog et al. (2024) for The Ocean Cleanup. Their model treats the mission as a longest-path problem within a directed acyclic graph, where planning is conducted over a multi-day forecast horizon and re-optimized daily in a rolling-horizon scheme. This predictive approach has proven highly effective, significantly increasing collection yields compared to reactive strategies. While these methods are powerful, their reliance on accurate forecasts makes them fundamentally offline.

In contrast, fully online and hybrid approaches, while less common, represent an emerging research direction. Hybrid systems, such as the two-stage framework proposed by Gao, Du, Yu, and Zhao (2023), combine proactive planning based on historical data with a reactive component that adjusts routes in real-time when new debris is detected. Similarly, hierarchical planners like the one developed by Mahmoudzadeh, Powers, Sammut, Yazdani, and Atyabi (2018) for AUVs integrate high-level reactive mission planning with low-level motion control to adapt to dynamic ocean environments. These models offer greater flexibility than purely offline methods but often trade global optimality for real-time adaptability.

Implementing practical Systems The transition from theoretical optimization models to tangible, real-world solutions is driven by non-governmental organizations and research initiatives. Among these, The Ocean Cleanup is one of the most prominent, operating large-scale collection systems directly in the Great Pacific Garbage Patch (GPGP). **?** The existence of such foundations shows that research in this area is requested and important, as efficient missions are crucial in such dynamic and complex environments.

Addressing ecological Considerations A critical factor in the planning of cleanup missions is minimizing ecological impact, particularly on the neuston community, organisms that live at the ocean surface. The GPGP is not only a plastic accumulation zone but also a neuston habitat, raising concerns about possible overlap and unintended harm. Egger et al. (2024) found that neuston abundance in high-plastic-density areas is often similar to or even lower than in surrounding waters, suggesting that focusing cleanup operations on peak-density zones may help avoid negative ecological impacts.

3 Methodology

The previous chapter provided an extensive overview of the literature and theoretical background relevant to this thesis. This chapter applies that theoretical knowledge by detailing the model, its components, and implementation. It introduces the main components: hotspots and vessels, describing how vessels interact with hotspots and how this interaction is influenced by ocean currents. Building on this, the mathematical model is derived and explained. The chapter concludes with a presentation of the algorithms used for implementation.

Notation Overview

Table 3.1: Model notation and symbol reference

Symbol	Type	Description	Domain/Range
H	Set	All hotspots	$i \in H$
B	Set	All vessels	$j \in B$
T	Set	Discrete timesteps	$t \in \{0, \ldots, T_{\max}\}$
$N(t)$	Set	Active hotspots at time t	$N(t) \subseteq H$
π_i	Parameter	Value of hotspot i	$[1, 10000]$
d_i	Parameter	Drift factor of hotspot i	$[0.01, 0.2]$
d_j	Parameter	Drift factor of vessel j	0.1 (constant)
ϵ	Parameter	Collection radius	0.1 km
Δt	Parameter	Timestep duration	1 hour
$p_i(t)$	State	Position of hotspot i	R^2
$q_j(t)$	State	Position of vessel j	R^2
$y_i(t)$	State	Activity status of hotspot i	$\{0, 1\}$
$z_{ij}(t)$	Decision	Assignment: j targets i	$\{0, 1\}$
$x_{ij}(t)$	Outcome	Collection: j collects i	$\{0, 1\}$
$s_j(t)$	Function	Baseline speed of vessel j	R_+ (km/h)
$u_j(t)$	Function	Unit vector from j to target	$R^2, \|u_j\| = 1$
$c_j(t)$	Function	Current drift acting on j	R^2 (km/h)
$ed_{ij}(t)$	Function	Euclidean distance j to i	R_+ (km)

3.1 Model Components

This study considers two main components: plastic accumulation zones (hotspots) and collection vessels, both components were implemented synthetically in this work. Implementing these components and their interactions involved complex and iterative refinement, requiring thorough consideration of environmental dynamics. Modelling hotspot movement and integrating vessel control strategies requires balancing computational efficiency with ecological realism. Since modelling these components was a substantial part of this work, their implementation is presented in the following. The subsequent subsections describe each component's representation, internal attributes, and movement behaviour within the model.

3.1.1 Hotspots

Rather than simulating a continuous plastic field, this study directly models a set of discrete, high-density plastic accumulations called hotspots. Each hotspot constitutes a compact and coherent cluster of plastic which has a defined composition, spatial extent, and drift behaviour. This approach abstracts from the full complexity of plastic dispersal and focuses on identifiable, collectible clusters of debris. Accordingly, these hotspots serve as targets in the optimization model and represent high-priority cleanup areas within a broader polluted region. The following sections outline the characterization of these hotspots and their role within the modelling framework.

Hotspot Modelling and Properties A hotspot is characterized by two main properties: its value and its drift factor. The value quantifies the importance of collecting the hotspot. The drift factor determines the extent to which it is influenced by ocean currents. These properties are derived from the hotspot's composition. The following passage explains the formation of a hotspot:

Initially, a hotspot is generated by selecting a small set of plastic types from a weighted list. This list favours commonly observed items such as fragments, bottles, or bags. The total number of items in the hotspot is drawn from a probability distribution. This typically results in a size between 100 and 1000 pieces. The total is distributed among the selected plastic types in variable proportions, yielding a unique composition for each hotspot. Based on this composition, the hotspot's overall weight, density, and surface area are calculated. Using a fixed random seed for hotspot composition enables regeneration of the hotspots and ensures reproducibility across experiments. For each hotspot, a value is first derived as

$$\pi_{\text{i raw}} = area_i \times density_i \times weight_i$$

Without scaling, the score would differ greatly across different runs, since the random generation may produce either predominantly small or predominantly large hotspots. This would make a meaningful comparison between runs difficult. To address this, a min–max normalization is applied, mapping the lowest score in a run to 1 and the highest to 10 000, with all intermediate values distributed proportionally.

$$\pi_i = 1 + \left(\pi_{\text{raw}} - \pi_{\min}\right) \frac{9999}{\pi_{\max} - \pi_{\min}}.$$

This approach offers two key advantages. First, it keeps values within a consistent and interpretable range. Second, it preserves the relative ordering of hotspots. Hotspots with higher physical magnitude are consistently assigned higher values. As a result, the values are easy to interpret and directly comparable within each seeding run. Accordingly, the value attribute should be interpreted as a relative priority score that distinguishes between more and less important hotspots in the modelled environment. It is important to note that the value of a hotspot remains consistent throughout the entire time frame, as the composition of a hotspot does not change after seeding. The value pi_i is not influenced by the movement of a hotspot.

The movement of a hotspot is only influenced by currents in this study. Each hotspot is assigned an individual drift factor d_i,based on the composition which determines its advection. The drift factor depends on three properties: density, surface area, and weight. Hotspots with lower density, larger surface area, and lower weight are transported more strongly by the surrounding flow. Three quantities are defined to represent these effects: the density factor, which captures buoyancy relative to seawater; the area factor, scaling with exposed surface; and the weight factor. These components are then combined to calculate the drift factor. The result is finally constrained to the interval $[0.01,\ 0.2]$ in order to maintain physically realistic behaviour. The function looks similar to the one for the value; however, there are some key differences: the drift factor is not scaled, it is not constrained to an interval and it uses factors, which are adjusted values to account for factors such as water salinity.

$$d_i = \text{density factor}_i * \text{area factor}_i * \text{weight factor}_i$$

Hotspot Movement After explaining the core properties of a hotspot, based on this information, the actual hotspot movement can now be explained.

Figure 3.1 illustrates the simplified process underlying hotspot movement. Following the initialization (seeding) of a hotspot, current data corresponding to its position are retrieved. Subsequently, the new position is computed as a function of the current and the drift factor. The simulation advances by one time step, updating the position of each hotspot accordingly. This iterative process is repeated for each time step, resulting in a time-resolved trajectory. The simulation terminates upon reaching the specified time horizon. As this constitutes a core feature of the study, the following pseudocode (see algorithm 1) provides a more detailed formalization of the process:

The process begins with an initialization step, where a set of hotspots H is seeded at random locations within the predefined study area. Each hotspot i is assigned the composition-specific drift factor d_i and the value π_i.

This is followed by an iterative advection procedure that updates the position of each hotspot over a defined time horizon T. The simulation proceeds in discrete time steps Δt, which is one

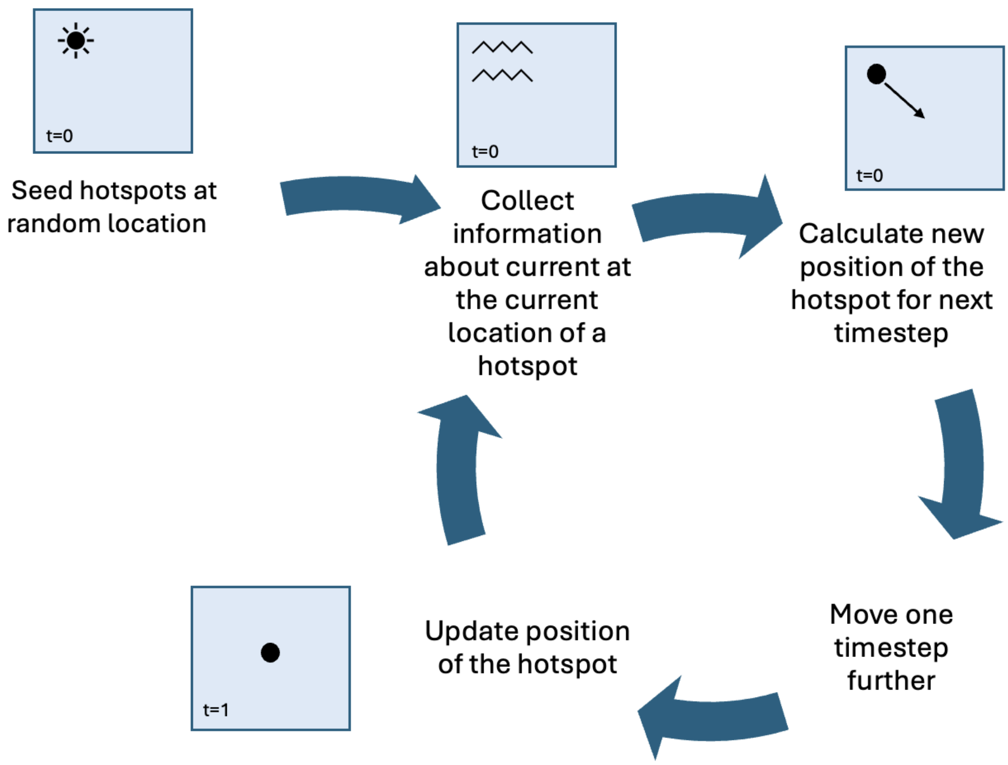

Figure 3.1: Illustration of the cyclic movement behaviour of hotspots within the simulation.

hour. At each time step t, for every hotspot, the velocity components of the local ocean current, $(u_{\text{sea}}(t), v_{\text{sea}}(t))$, are retrieved from the environmental dataset at the hotspot's current position $p_i(t)$. This ocean current velocity is then scaled by the hotspot's specific drift factor d_i, resulting in a relative velocity, $(u_{\text{rel}}(t), v_{\text{rel}}(t))$, which dictates the hotspot's actual displacement.

Finally, the new position of the hotspot, $p_i(t + \Delta t)$, is calculated based on its relative velocity. The position is then updated based on the previous movement calculation. By repeating this procedure, the model generates a complete trajectory for each hotspot over the time horizon T. The advection step itself is fully deterministic, as the displacement of each hotspot is uniquely determined by the input current field and its drift factor. Furthermore, since the initial random seeding of hotspots is controlled by a fixed random seed, the entire simulation process is reproducible: identical seeds and input data will always yield identical hotspot trajectories.

Algorithm 1: Advection of hotspots

Input: Hotspots H with drift factors d_i, ocean current data, time horizon T
Output: Trajectories $p_i(t)$ over T
$t \leftarrow 0$;
while $t < T$ **do**
  foreach *hotspot* i **do**
    Collect local current $(u_{\text{sea}}, v_{\text{sea}})$ at position of $p_i(t)$;
    Compute relative velocity: $u_{\text{rel}}(t) \leftarrow u_{\text{sea}}(t) \cdot d_i$, $v_{\text{rel}}(t) \leftarrow v_{\text{sea}}(t) \cdot d_i$;
    Update position of $p_i(t + \Delta t)$ using $(u_{\text{rel}}(t), v_{\text{rel}}(t))$;
  $t \leftarrow t + \Delta t$;

3.1.2 Collection Boats

Following the examination of the target component, the subsequent elements are the collection vessels. The purpose of the vessels is the collection of hotspots. Hotspots move passively according to ocean currents, whereas vessels also navigate actively toward them. As a result, vessel trajectories are determined by hotspot positions during the simulation, and positioning involves both passive (current influence) and active (navigation) movement, thereby increasing system complexity.

The motion of the vessels is characterized by a baseline speed $s_j(t)$ and is influenced by the surrounding current field. At each time step, vessel j moves in the direction of the current position of its assigned hotspot $p_i(t)$ and advances by its maximum Euclidean step length, unless $p_i(t)$ is reached earlier. The position of vessel j is updated according to

$$q_j(t+1) = q_j(t) + (s_j(t)\, u_j(t)\,) + c_j(t)$$

where $q_j(t)$ denotes the present position, $s_j(t)$ the vessel's baseline speed, and $u_j(t)$ a unit vector pointing from $q_j(t)$ to the current position of the target hotspot $p_i(t)$. Each vessel always moves directly toward its assigned target, following the unit direction vector

$$u_j(t) = \frac{p_i(t) - q_j(t)}{\|p_i(t) - q_j(t)\|}$$

The term $c_j(t)$ accounts for the movement based on the current of the ocean at the vessel's location.

$$c_j(t) = d_j \cdot \begin{pmatrix} u_{\text{sea}}(t) \\ v_{\text{sea}}(t) \end{pmatrix}$$

This movement is illustrated in 3.2. The vessel (square) proceeds toward the hotspot (circle). The direct route, represented by the unit vector, is indicated by the red arrow. The ocean current exerts additional influence on the vessel's trajectory, as represented by $c_j(t)$ in the formula and the dotted red arrow in the graphic. Consequently, the overall movement of a vessel results from the combination of current-induced and self-propelled motion, visualized by the orange arrow.

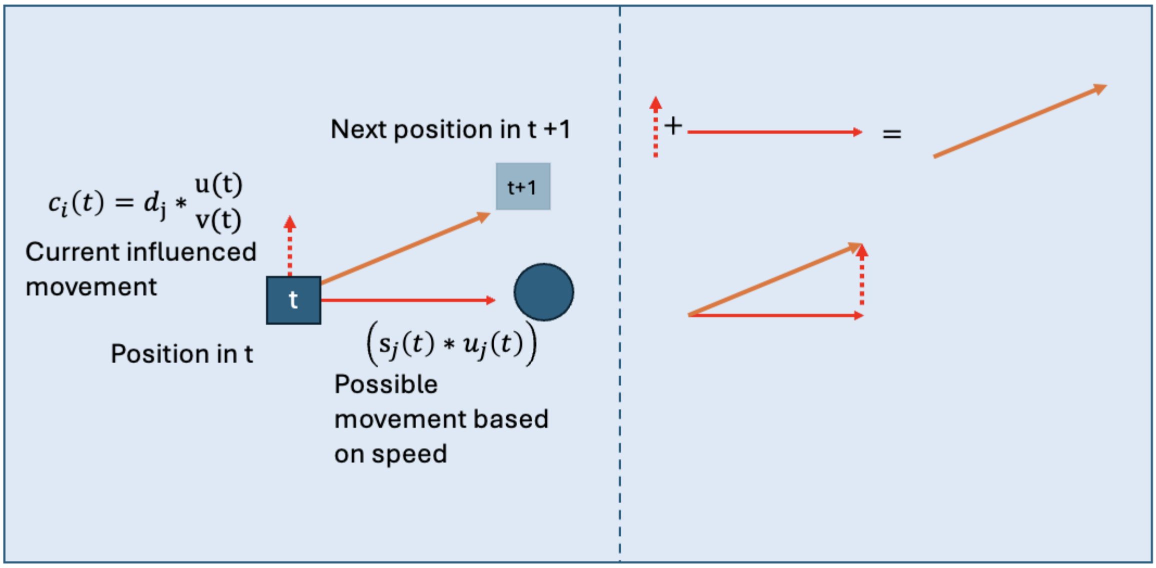

Figure 3.2: Detailed movement behaviour of the vessel.

Vessels experience a slight, uniform current influence. Because the fleet is homogeneous, a fixed drift factor $d_j = 0.1$ is used for all vessels. This means that in practice, currents may speed or slow a vessel, or push it slightly off course. But still, the vessel actively navigates and is not simply carried by the current.

Vessel speed $s_j(t)$ is time-dependent, although the basic vessel speed is constant. This variation is required to ensure that a vessel enters the predefined collection radius ϵ around a hotspot for successful collection. With a fixed speed, there is a risk that the vessel may overshoot the hotspot within a single time step, resulting in delayed collection until subsequent environmental changes bring the hotspot within range. To mitigate this, $s_j(t)$ is reduced whenever the remaining distance to the target hotspot is less than the vessel's nominal step length, ensuring precise arrival at the collection radius. The general movement of vessels over the time horizon is visualized in 3.3. Initially, each vessel is placed at the centre of the map, current information at its position is obtained, and the position of the assigned target hotspot is determined. Subsequently, $(u_j(t))$ and $c_j(t)$ are computed, and $q_j(t+1)$ is updated accordingly. The simulation advances by one time step, and the process is repeated until $t = T$ is reached, ensuring movement at each discrete time step.

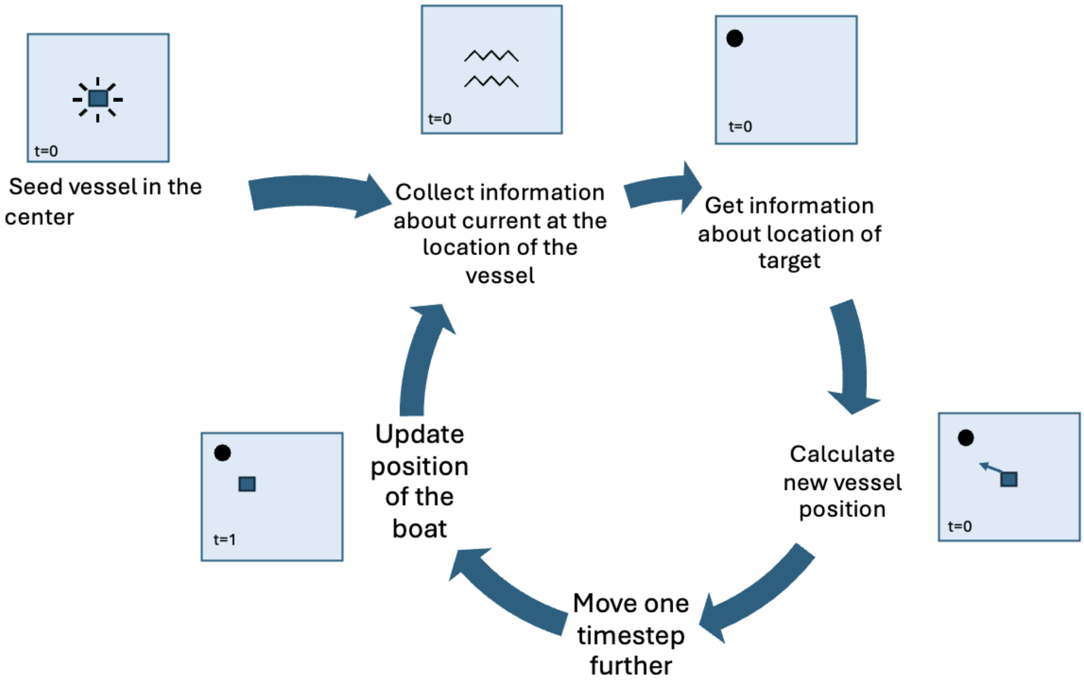

Figure 3.3: Cyclic movement behaviour of vessels within the simulation.

3.2 Illustration of System Interaction

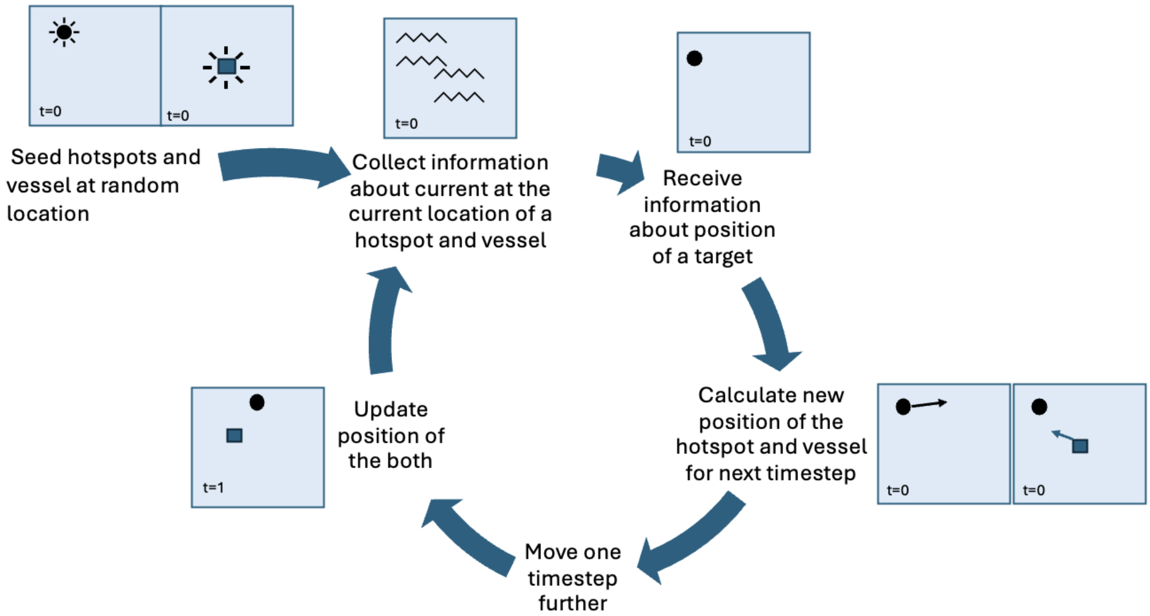

Figure 3.4: Cyclic movement pattern of vessels and hotspots.

Figure 3.4 illustrates the combined movement of both components, focusing on the interaction between drifting plastic hotspots and navigating vessels within the simulation framework. This process is repeated iteratively at each time step, reflecting the online nature of the decision-making and movement model.

The cycle begins with the random seeding of both plastic hotspots (represented as circles) and vessels (represented as squares) within the defined study region at $t = 0$. In the subsequent step, both agents access the environmental dataset to obtain current information at their respective positions, which is essential for computing advection. The vessel then receives updated information regarding the current location of the target hotspots. In the next phase, the new positions of both the hotspot and the vessel are determined for the subsequent time step. The hotspot drifts passively according to its individual drift factor and the current, while the vessel navigates actively toward the assigned target, which is also influenced by the surrounding current field. After the new positions are computed, the simulation advances by one time step. The updated positions are then incorporated into the cycle, serving as input for the next iteration. This sequence continues until $t = T$, when the time horizon is exceeded.

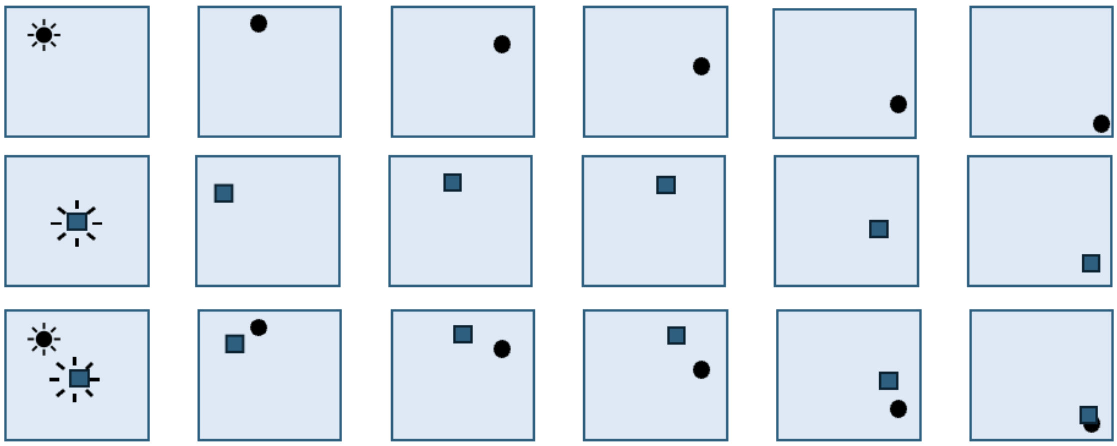

Figure 3.5: Trajectory visualization of hotspots and vessels.

Figure 3.5 presents a visual summary of the simulation logic and the interaction between plastic hotspots and collection vessels over multiple timesteps. The diagram is structured in three rows, each highlighting a distinct component of the system.

The top row sequence illustrates the movement of plastic hotspots. After initial seeding at a single location (leftmost frame), the hotspot drifts over time under the influence of ocean currents. The subsequent sequence depicts vessel movement within the same environment, where positions are updated at each time step based on both active navigation and passive movement due to currents. Vessel trajectories consistently align with the current position of the assigned hotspot. The third row integrates both components, illustrating the coupled interaction between vessel and the drifting hotspot. It shows how the vessel continuously adjusts its heading in response to updated hotspot positions at each time step. Although the vessel and hotspot start in close proximity, the collection process still requires six discrete time steps to complete. This delay arises because the vessel always moves toward the most recently known position of the hotspot. Since the hotspot itself continues to drift during the update interval, the vessel is effectively chasing a moving target, resulting in a longer overall approach time.

3.3 Technical Implementation in OpenDrift with Python

The technical implementation of the model is achieved using two separate classes, both based on the OpenDrift (Dagestad, Röhrs, Breivik, and Ådlandsvik (2018)) framework in Python (Version 3.11).

This work uses functions to read in current data, advection functions and visualisations from Opendrift. There is a predefined class for plastic movement included, however this was not suitable for this work, which is why a custom implementation, the class “OpenDriftPlastCustom” class was implemented. It models the plastic hotspots, where each hotspot is characterized by an initial position, an associated profit value, a drift factor d_i, and a status variable indicating activity. Advection of hotspots based on ocean currents is also managed within this class. The “DriftBoat” class was also implement for this work and models the vessels, which are initialized at the depot located at the centre of the map and assigned speed factors. Vessel instances reference the hotspot model, enabling access to hotspot attributes. Both classes utilize a shared reader and consistent time and extent settings, thereby maintaining a coherent spatio-temporal simulation environment. Implementing the two components as independent OpenDrift classes reflects their distinct roles, hotspots as drifting targets and vessels as controllable collection agents, which supports modularity and functional clarity.

In the developed algorithms, information from both classes is integrated. This information is used to construct the whole interacting cycle, visualized in 3.4.

3.4 Mathematical Model Formulation

3.4.1 Model Components and Notation

For the formal specification of the model, the graph structure, its attributes, and the system evolution, the following notation is introduced:

Sets The mathematical formulation is built upon the following fundamental sets, which represent the vessels, hotspots, and temporal components.

H set of all hotspots, indexed by i.

B set of all vessels, indexed by j.

$T = \{0, 1, \ldots, \mathcal{T}\}$ set of discrete time steps with step size Δt.

$N(t) \subseteq H$ set of active hotspots at time t, defined as

$$N(t) = \{i \in H : y_i(t) = 1\}.$$

Parameters These parameters and variables quantify the physical properties of the hotspots and vessels, the environmental influences they are subject to, and their dynamic states over time.

$\pi_i > 0$ Value of hotspot $i \in H$.

$\epsilon = 0.1$ Collection radius of 100 meters.

$s_j(t) > 0$ Nominal vessel speed of vessel $j \in B$

d_i Drift factor of hotspot $i \in H$ (between 0.01 and 0.2), determined by its composition.

d_j Drift factor of a vessel, set to 0.1.

$u(t),\ v(t)$ Ocean current components at a position R^2 and time t.

Δt Discrete simulation time step (one hour).

T Time horizon of the simulation.

$ed_{ij}(t)$ Euclidean distance from a vessel j to a hotspot i at timestep t

$q_j(t)$ Position of vessel $j \in B$ at time t.

$u_j(t)$ Unit vector pointing from vessel $q_j(t)$ to its assigned hotspot $p_i(t)$.

$c_j(t)$ Current-induced drift vector acting on vessel $j \in B$.

State and Decision Variables The system's dynamics are determined by a set of variables that describe both the state of the environment and the operational choices made by the vessels. These include continuous state variables for the positions of hotspots ($p_i(t)$) and vessels ($q_j(t)$), and a binary state variable ($y_i(t)$) to track whether a hotspot is active. The core of the model lies in the binary decision variable $z_{i,j}(t)$, which represents a vessel's choice to target a specific hotspot at a specific time, and the resulting binary outcome variable $x_{i,j}(t)$, which indicates a successful collection.

$p_i(t) \in R^2$ position of hotspot i at time t.

$q_j(t) \in R^2$ position of vessel j at time t.

$z_{i,j}(t) \in \{0, 1\}$ decision variable: 1 if vessel j targets hotspot i at time t, 0 otherwise.

$x_{i,j}(t) \in \{0, 1\}$ outcome variable: 1 if hotspot i is collected by vessel j at time t, 0 otherwise, directly influenced by $z_{i,j}$.

$y_i(t) \in \{0, 1\}$ state variable: 1 if hotspot i is active at time t, 0 otherwise.

The $z_{i,j}(t)$ is the variable which is optimized, to find the best values for these variables determines the outcome of the whole simulation. The state of the system evolves based on the assignment decisions $z_{i,j}(t)$ made at each step. The goal is to maximize the total value collected over the horizon.

3.4.2 Graph Structure and Time-Dependency

In order to understand the underlying optimization problem, he following section describes the underlying graph structure with time dependency.

Static Representation In classical routing and optimization problems, graph nodes typically represent static locations, and arcs denote traversable paths, often weighted by distance, time, or cost. While this is appropriate for problems with fixed locations, the present context involves changing locations. To illustrate the graph structure, the influence of environmental currents is initially omitted, and the problem is modelled statically. Subsequently, based on this preliminary graph, the movement component is incorporated. The graph 3.6 presents a static example.

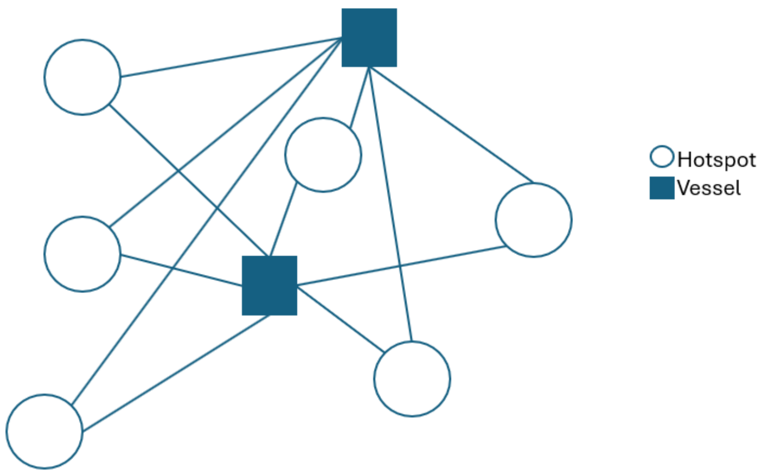

Figure 3.6: Static graph without movement

There are two sets of nodes in 3.6: Vessels B and active hotspots N. In a static graph, this would be:

$$G = (V, A)$$

where V is $V = N \cup B$ are the vessels and hotspots. An arc represents the potential connection of a vessel to a hotspot, so that there is a connection between each vessel to each hotspot, but not between the hotspots or the vessels themselves. The arc weights in this example would be the distances from each vessel to each target hotspot. The problem would no longer be time-dependent, as the current has no influence on the positions.

Graph with Movement incorporated However, in the studied example, there is movement over the multiple discrete time steps. For each discrete timestep in the time horizon, the graph is changing. Thus, a time-layered graph (Michail (2015)) will be constructed, consisting of a snapshot at each t of the time frame. In the graphic below (see: 3.7), these snapshots are depicted in three steps. Each boat (square) can move toward each hotspot (circle) at each timestep.

At each time step t, a snapshot of the respective graph structure is obtained, reflecting the dynamic evolution of the system.

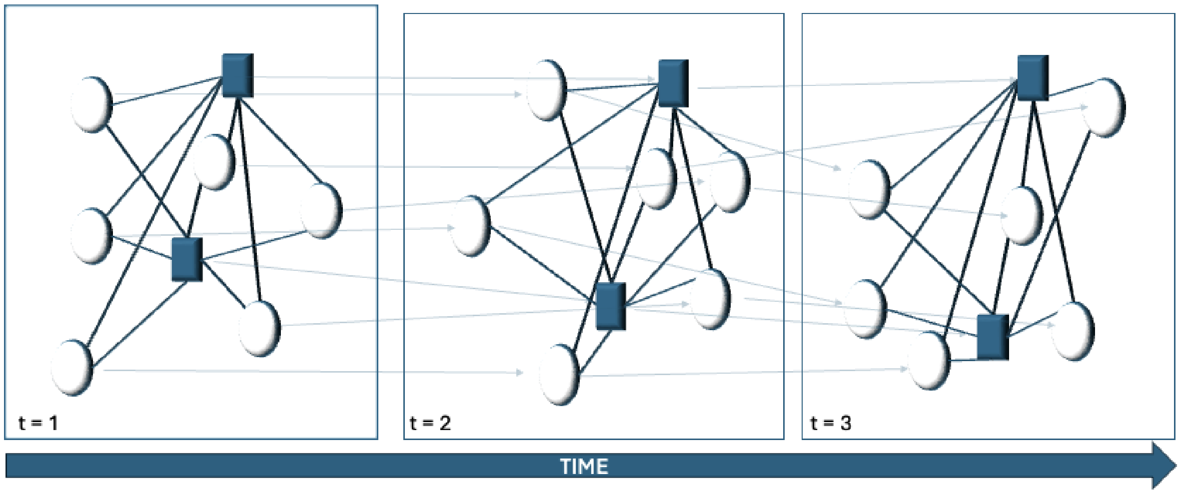

Figure 3.7: Time layered graph for t = {1,2,3}

$$G(t) = (V(t), A(t))$$

this denotes the state of the environment at time t, where $V(t)$ is the set of nodes (vessels and active hotspots)

$$V(t) = B \cup N(t)$$

and $A(t)$ is the set of arcs representing the connections at that time between each boat and hotspots.

The complete time-dependent system is then given by the collection:

$$\mathcal{G} = \{G(t) \mid t = 0, \ldots, T\},$$

Arc Weights As previously described, the nodes of the time-layered graph represent vessels and hotspots, with arcs denoting the connections between these two classes. In this time-dependent context, defining arc weights presents a particular challenge.

The Euclidean distance between a vessel and a hotspot at time $t = 0$ could, in principle, serve as an arc weight for decision-making. However, as shown in Figure 3.8, this value does not represent the actual distance the vessel must travel to reach the hotspot. It only captures the straight-line distance between both positions at the initial time step (blue dotted line), while in reality, both the vessel and the hotspot are continuously affected by ocean currents. The actual trajectory (orange line) is therefore longer and curved, reflecting the cumulative effect of environmental drift.

Distance, in this context, is only an indirect constraint, as it translates into time cost rather than serving as a hard limit. The more relevant measure would be the total travel time required

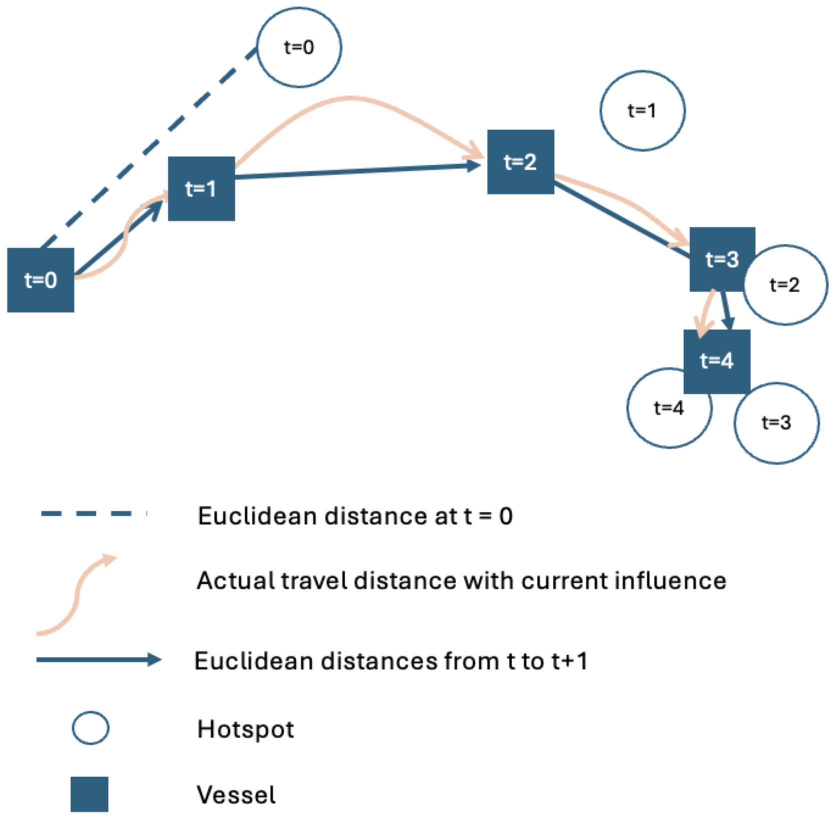

Figure 3.8: Trajectory of a vessel and its corresponding target over five time steps.

for a vessel to reach a hotspot, since this directly limits the operational horizon. However, neither total travel distance nor total travel time can be used as arc weights in the model.

Using total travel distance would require knowing when a vessel actually reaches the hotspot, information that is only available retrospectively. Even if this arrival time were known, simply summing up the Euclidean distances between intermediate time steps (dark blue solid lines) would not yield the true path length. The vessel's movement is continuously altered by ocean currents, meaning the actual travelled distance (orange line) can only be obtained by tracking its trajectory throughout the simulation.

Total travel time would in theory be an ideal cost measure, as it reflects the real constraint of limited mission duration. Yet, it cannot be pre-determined either: at the moment a target is selected, it is unknown when or whether the vessel will reach it. Even with full knowledge of the hotspot's future trajectory, the vessel's arrival time cannot be predicted, since it depends on the interplay between both motion patterns and the ocean current field.

Pre computing all possible travel distances or times would therefore be computationally infeasible. As Figure 3.9 illustrates, the system is time-dependent, the travel time between two nodes varies with the start time and the prior routing sequence. Because of a different routing decision in the first timestep, the travel time from node three to four differs. This travel time can only be found out via a simulation. Thus, for each possible order of visits, both vessel and target trajectories would have to be simulated explicitly, leading to an exponential growth of combinations.

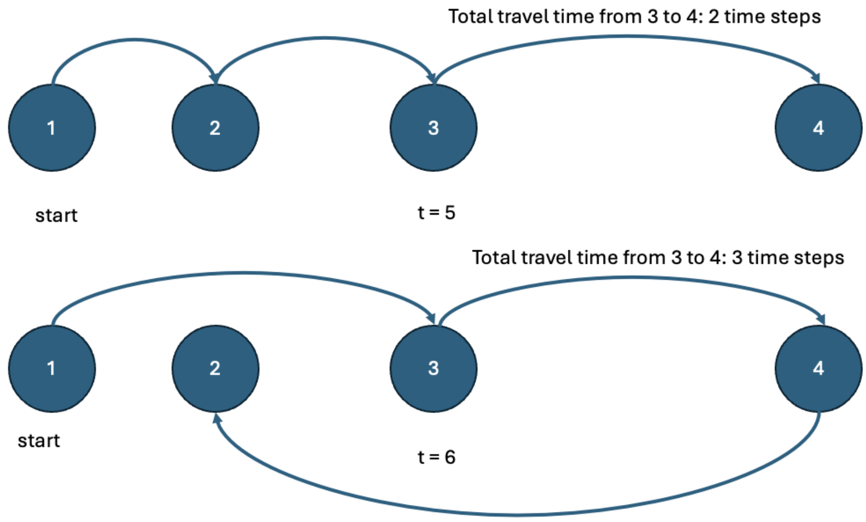

Figure 3.9: Illustration of time dependency caused by movement.

For these reasons, neither total distance nor total travel time can serve as reliable arc weights. Instead, this work employs the current Euclidean distance between a vessel B and an active hotspot $N(t)$ at time t, denoted as $ed_{i,j}(t)$. This measure, while simplified, can be evaluated instantly and provides a consistent local proxy suitable for online decision-making, where future hotspot positions are intentionally not used.

3.4.3 Objective Function and Constraints

The previous section explained the structure of the graph, which will now be used to form the mathematical optimization model.

Objective Function The objective function of the optimization problem is formalized as "Maximize the total collected value":

$$\max \sum_{t \in T} \sum_{j \in B} \sum_{i \in N(t)} \pi_i \, x_{i,j}(t).$$

C1 – Collection Rule A hotspot can only be collected if all three of the following conditions are met. In that case, $x_{i,j}(t) = 1$. This is directly influenced by the decision variable $z_{i,j}(t)$:

(i) Hotspot must be active This ensures that a collection is only counted if the hotspot has not already been collected in a previous step.

$$x_{i,j}(t) \le y_i(t), \quad \forall i, j, t$$

(ii) Hotspot must be targeted This rule prevents accidental collections by requiring that the vessel has an explicit, active assignment to the hotspot.

$$x_{i,j}(t) \le z_{i,j}(t), \quad \forall i, j, t$$

(iii) Vessel must be in range This is the physical constraint, confirming that the vessel is within the predefined collection radius ϵ of the hotspot.

$$x_{i,j}(t) \le 1\Big(\|q_j(t) - p_i(t)\| \le \epsilon\Big), \quad \forall i, j, t$$

This ensures that collection cannot occur "by accident": a hotspot must be active, explicitly chosen as a target, and spatially reachable.

C2 – Non-negative Values

$$\pi_i \ge 0, \quad \forall i \in H.$$

Hotspot values are assumed to be non-negative, so the collected value is always non-decreasing.

C3 – Each Hotspot can be collected only once This rule is modelled using a binary state variable $y_i(t)$ that tracks whether a hotspot is active (1) or has already been collected (0). The process is defined by an initial state and a state update rule.

Initialization At the beginning of the simulation ($t = 0$), every hotspot is marked as active and available for collection. This sets the initial condition for the system.

$$y_i(0) = 1, \quad \forall i \in H$$

State Update For every time step, the state of a hotspot is updated based on collection events. If a hotspot i is collected by any vessel j at time t, the sum $\sum_{j \in B} x_{i,j}(t)$ becomes 1. This transitions the hotspot's state from active ($y_i(t) = 1$) to inactive ($y_i(t+1) = 0$) for all future time steps. If no vessel collects the hotspot, the sum is 0, and its state remains unchanged.

$$y_i(t+1) = y_i(t) - \sum_{j \in B} x_{i,j}(t), \quad \forall i \in H, \forall t \in T$$

This two-part formulation guarantees that once a hotspot's value is collected, it is permanently removed from the set of active targets and cannot be collected again.

C4 – Exclusivity: one Vessel per Hotspot at a Time

$$\sum_{j \in B} z_{i,j}(t) \leq 1, \quad \forall i, t.$$

At most one vessel may target a given hotspot at the same time, avoiding conflicts.

C5 – Exclusivity: one Target per Vessel at a Time

$$\sum_{i \in H} z_{i,j}(t) \leq 1, \quad \forall j, t.$$

Each vessel can pursue at most one hotspot at a time, preventing simultaneous multiple targeting.

C6 – Initial Vessel Position

$$q_k(0) = \textit{centre of the map}, \quad \forall k \in K.$$

All vessels start at the centre of the map at the beginning of the time horizon.

C7 – Vessel Dynamics (Flow Conservation)

$$q_j(t+1) = q_j(t) + (s_j(t)\, u_j(t)\,) + c_j(t) \quad \forall j \in B,\, \forall t \in T,$$

This equation governs the vessel motion: it moves according to its chosen direction and nominal speed, plus a drift caused by currents.

C8 – Time Horizon Restriction Collections are only allowed within the planning horizon, i.e., for $t \in T = \{0, 1, \ldots, \mathcal{T}\}$.

C9 – Target Persistence (Continuity)

$$z_{i,j}(t+1) \geq z_{i,j}(t) - x_{i,j}(t), \quad \forall i, j, t.$$

If a vessel is targeting a hotspot, it will continue to do so until the hotspot is collected or the vessel explicitly switches target. This constraint is only valid for the Online Greedy Algorithm without retargeting.

3.5 Algorithms

This section introduces the algorithms developed to address the optimization problem and to find the optimal decisions for $z_{i,j}(t)$. It first outlines the construction of a baseline solution, followed by a detailed explanation of the main algorithm and an enhanced variant that incorporates additional flexibility. The section concludes with a comparative analysis of both implemented algorithms.

3.5.1 Baseline Problem Formulation and Implementation [B1 and B2]

The baseline solution addresses the problem by removing the current influence, thereby simplifying the dynamic aspects of the model, resulting in static hotspot positions following their initial placement. Vessel movements are likewise assumed to be unaffected by current-induced drift.

Within this stationary framework, the problem can be readily formulated as a graph. The graph structure remains constant throughout the entire time horizon, and the core decision problem reduces to assigning vessels from the set K to active hotspots in the set $N(t)$. This assignment is represented graphically, as previously described in 3.6.

$$G = (V, A),$$

where the vertex set V includes the vessels B and the active hotspots $N(t)$. An arc $(j, i) \in A$ represents a feasible assignment of vessel $j \in B$ to hotspot $i \in N(t)$.

By omitting the movement of targets and environmental dynamics, the problem is reduced to a classical Orienteering Problem. This simplified formulation is subsequently solved using the Gurobi (Gurobi Optimization, LLC (2024)) optimization solver. This formulation is computationally easier to solve, as it corresponds to a standard Orienteering Problem (OP) without dynamic moving target components. However, it remains an NP-hard problem as proved by Golden et al. (1987). A key simplification is that the graph structure is consistent and does not vary with the time step, which reduces the complexity compared to time-dependent formulations. The mathematical formulation of the static baseline problem closely resembles the structure of the dynamic model. The key difference is that several components no longer depend on time or ocean currents. In particular, the positions of hotspots remain fixed throughout the planning horizon, and the movement of vessels is not influenced by environmental drift. Consequently, the model is considerably simplified while retaining the same notation for sets, parameters, and decision variables where possible.

Sets The sets used in the static formulation remain identical to those of the dynamic model. They define the hotspots, vessels, and time steps.

Parameters In contrast to the dynamic case, no drift effects are modeled. Hotspots do not drift, and vessels are unaffected by ocean currents. The parameterization is therefore reduced to reward values, the collection radius, and nominal vessel speed:

State and Decision Variables Since hotspot positions are static, they are modeled as fixed points in the map. Vessels still evolve in discrete time, but their positions change only as a result of their own propulsion. The decision variables remain binary and retain the same interpretation as in the dynamic model: they indicate whether a vessel targets or collects a hotspot at a given time step.

Objective Function Although the static baseline discards current effects and hotspot drift, the structure of the objective function remains unchanged compared to the dynamic model. The goal is still to maximize the total reward collected over the planning horizon.

The baseline scenario is implemented as a static Orienteering Problem, in which a fleet of vessels, denoted by the set B, is tasked with collecting rewards by visiting a subset of fixed targets (plastic hotspots), indexed by $i \in H$. Each vessel starts from the center of the map and operates within a prescribed time budget T. The overarching objective is to maximize the aggregate value accrued from all collected hotspots, while adhering to both routing and temporal constraints.

To address this optimization problem, a Mixed-Integer Linear Programming (MILP) formulation is employed, which is subsequently solved using the commercial solver Gurobi. The practical implementation proceeds as follows:

Constraints

The following constraints are incorporated in the Problem formulation:

Objective Function The main goal is to choose a set of hotspots and assign routes to the vessels in such a way that the total value of all collected hotspots is maximized.

C1: Each hotspot is Visited At Most Once This rule ensures that a hotspot cannot be collected multiple times, either by the same vessel or by different vessels. Once a hotspot's value is counted, it is removed from the list of available targets.

C2: Every Vessel Starts at the Depot This constraint dictates that each vessel used in the mission must begin its collection tour from the central starting point (the depot). No vessel can start its route from a random hotspot.

C3: Open Tours are Allowed This is a rule of convenience. It means that vessels do not have to return to the depot after they have collected their last assigned hotspot. Their tour simply ends at the last collected location.

C4: Flow Conservation This is a fundamental rule for creating valid routes. It states that if a vessel travels to a hotspot, it must also travel away from it (unless it's the last stop on its tour). This prevents dead ends and ensures that the path is continuous.

C5: No Branching from Unvisited Nodes This logical constraint prevents a vessel's route from continuing from a hotspot that it never actually visited. A vessel can only travel *from* hotspot A *to* hotspot B if its route includes a stop *at* hotspot A.

C6: Time Limit per Vessel Each vessel has a fixed time budget. The total time required to travel its assigned route, calculated from the distances between hotspots and the vessel's speed, cannot exceed this maximum allowed time. This limits how far a single vessel can travel.

C7: Subtour Elimination Using Miller-Tucker-Zemlin (MTZ) C. E. Miller, Tucker, and Zemlin (1960) constraints: This is a crucial technical constraint that prevents invalid solutions. It ensures that each vessel follows one single, connected tour. Without this rule, the model might create illogical solutions where a vessel completes several small, disconnected loops (e.g., visiting hotspots 1 and 2, and separately visiting hotspots 8 and 9, without a path connecting them). This constraint forces the route to be a single, continuous path.

Solution Extraction and Evaluation

The optimization problem is solved with a time limit of 600 seconds. Upon completion, the set of active arcs and collected hotspots is extracted to reconstruct the corresponding route for each vessel.

While the static problem described above can be solved to optimality, it constitutes a substantially simplified variant of the original dynamic problem. Because all targets remain stationary, the solutions obtained are not directly comparable to those produced in the dynamic setting. Although the resulting collected reward offers some insight, these results cannot serve as a fully meaningful benchmark for the more advanced algorithms introduced later. Nevertheless, for completeness, the outcomes of this simplified approach are reported as Baseline 1 (B1).

To provide a more relevant benchmark, the visiting order derived from B1 is subsequently applied to the dynamic environment. Although the underlying optimization model was originally designed for a static scenario, transferring its solution sequence to the dynamic context enables an informative comparison. In particular, this approach provides a lower bound for the achievable performance of the dynamic algorithms. The outcomes of this procedure are denoted as Baseline 2 (B2) in the following discussion. B2 serves as an approximate lower bound, as it is notably disadvantaged when compared to the more sophisticated heuristic methods.

3.5.2 Online Greedy Algorithm [OG]

In the time-dependent setting with moving targets, computing exact optimal solutions is impractical. Solving such problems requires significant computation time, and the environment changes continuously due to drifting hotspots and varying conditions. As a result, by the time an optimal plan is computed, it is typically no longer valid. Instead, heuristic online approaches are required that can quickly solve problems and react to new state information.

The Online Greedy Algorithm (from now on, OG) provides a heuristic alternative to exact optimization. OG operates greedily: vessels make short-term decisions about $z_{i,j}(t)$ within the time of one collection, which lasts from assignment to the retrieval of a hotspot. The decision rule for assigning hotspots is based on a weighted trade-off between the value of a hotspot and its current Euclidean distance to the vessel.

The heuristic procedure is formally described in Algorithm 2 in a simplified form:

Algorithm 2: Greedy Heuristic for Plastic Retrieval

Input: Vessels B, hotspots H, time horizon T, weighting factors α, decision rule $f(\cdot)$, current data(t)

Output: Total collected value, route of each vessel

```
t ← 0;
while t < T do
    collect location information of all B and N(t);
    foreach vessel j ∈ B do
        if j has no assigned target then
            assign i ← arg max_{i∈N(t)} f(i, j, t, α);
        if j reaches i then
            collect i; remove i from N(t);
            assign new target based on f(·);
    t ← t + 1;
```

The algorithm begins by initializing the simulation time to $t \leftarrow 0$. During each iteration, which corresponds to a discrete time step of one hour, the system retrieves the location information of all vessels B and active hotspots $N(t)$. This ensures decisions are based on the latest environmental context.

Each vessel $j \in B$ is evaluated individually. If the vessel currently lacks an assigned hotspot, it selects a hotspot from the set $N(t)$ by maximizing a decision rule $f(i, j, t, \alpha)$ over all hotspots.

The scoring function used by each vessel is defined as follows: for every available hotspot, the vessel computes a score by weighing the normalized value and the normalized distance of that hotspots. The vessel then selects the hotspot with the highest computed score, systematically accounting for both the value (π_i) and euclidean distance ($ed_{ij}(t)$) of each hotspot. The hotspot value π_i remains static throughout the simulation. In contrast, the distance $ed_{ij}(t)$ nearly always changes due to ocean currents and movement affecting both vessel and hotspot positions. This method guarantees that the decision is transparent and directly based on the scoring logic. The Scoring function is as follows:

$$S_{ij}(t) = \alpha \cdot \left(1 - \frac{ed_{ij}(t)}{ed_{\text{max}}(t)}\right) + (1 - \alpha) \cdot \left(\frac{\pi_i}{\pi_{\text{max}}}\right) \tag{3.1}$$

The terms ed_{max} and v_{max} are the maximum observed distance and value, respectively, among all candidate hotspots at the current time. These are used to normalize the components to the interval $[0, 1]$, ensuring numerical comparability.

Normalization is necessary because distance and value use different units and have different ranges. Without normalization, the trade-off parameter α would not provide consistent weighting between proximity and value. After normalization, $\alpha \in [0, 1]$ is interpreted clearly: setting $\alpha = 1$ selects by distance only, while $\alpha = 0$ selects the highest-value targets regardless of location. As already explained, note that ed_{ij} is used as a proxy for travel time. The actual time to reach a hotspot is very variable and depends on environmental factors.

This decision rule is the core of the target selection process. It is re-evaluated each time a vessel becomes available for reassignment after receiving updated information about the locations. The outcome of the decision rule directly influences the decision variable $z_{i,j}(t)$ for the optimization problem, which is why a careful selection of the decision parameter α is crucial. While simple, this greedy strategy enables responsive behaviour in a dynamic environment without requiring full trajectory prediction. Formally, this scoring mechanism can also be interpreted as a parametrized decision rule $f(i, j, t, \alpha)$, which assigns a desirability score to each hotspot i for a given vessel j at time t. The vessel then selects the hotspot according to

$$i = \arg\max_{i \in N(t)} f(i, j, t, \alpha) \tag{3.2}$$

After target assignment, the vessel heads toward its assigned hotspot. At every time step, it moves toward the hotspot's updated position. Thus, the vessel constantly adjusts its heading based on the moving hotspot. Upon reaching its assigned hotspot (the predefined collection radius), the vessel collects the hotspot, which is then removed from the active set $N(t)$. Immediately after collection, the vessel executes a new assignment using the same decision rule $f(\cdot)$, ensuring uninterrupted operation and avoiding idle states.

The assignment is performed only when a vessel is unassigned. Once a hotspot is chosen, the vessel remains committed to reaching that specific hotspot and does not reevaluate its decision. Thus, the planning horizon is defined as the time between the assignment of a vessel to a hotspot and the actual collection of this hotspot. Once all vessels have completed their actions for the current time step, the simulation time is incremented by one, i.e., $t \leftarrow t + 1$. This process repeats until the time horizon T is reached.

The algorithm terminates by returning two outputs: the total accumulated value of collected plastic, and the complete trajectory followed by each vessel throughout the simulation. The greedy nature of this heuristic ensures that, at every decision point, unassigned vessels select the most promising hotspot, while assigned vessels persist in pursuing their current objectives. Although the approach does not guarantee global optimality, it remains computationally efficient and is well-suited for real-time plastic retrieval in dynamic maritime environments.

The vessels operate without any coordination or communication. Each vessel independently selects and pursues its assigned hotspot according to the decision rule described above. To prevent overlap, a one-to-one assignment ensures each hotspot is assigned to one vessel, and each vessel pursues only one hotspot at a time. With no coordination, trajectories are neither globally optimized nor mutually aware. As a result, vessels may follow intersecting or even colliding paths while heading to different hotspots. This occurs because movement planning is entirely local and reactive: each vessel advances toward its assigned hotspot's current position, adjusted by local ocean currents, without considering other vessels' paths or locations. This non-cooperative strategy reduces algorithmic complexity and allows decentralized deployment, but may cause inefficiencies or risks in constrained or congested environments. Addressing these issues would require integrating collision avoidance or cooperative path-planning mechanisms, which the presented heuristics do not consider.

Scoring Function detailed: Example Calculations

In the visualization 3.10, the assignment process of the greedy algorithm is illustrated. All four hotspots are considered simultaneously, each characterized by a specific value and distance to the vessel. The algorithm uses these two features to compute individual scores based on the Score Function at the current time step. The score calculation results are shown in the referenced table.

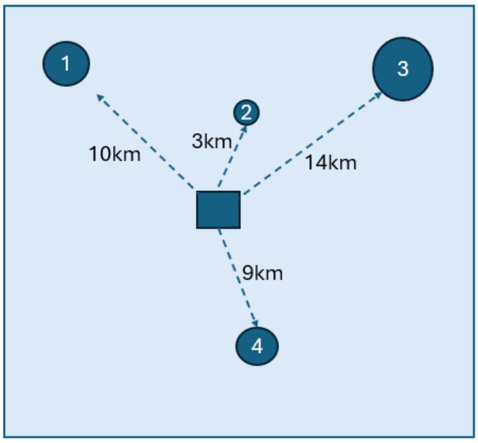

Figure 3.10: Illustration: Scoring-based assignment process for potential targets

Based on the calculated scores, the vessel in this example would first select target 4, as it achieves the highest score among all options (see table 3.2. After collecting this target, the algorithm then receives updated information about the locations of hotspots and evaluates the environment again to recompute the scores for the remaining hotspots, incorporating the updated distances resulting from the vessel's new position. If a second vessel were present, following this, it would select the next-best available target according to the same scoring rule. In the case of a tie, as seen in 3.2 between targets 1 and 2, the algorithm selects the hotspot with the lower target ID. This tie-breaking method ensures reproducibility and prevents ambiguity in multi-vessel coordination while maintaining a fully greedy decision structure.

Table 3.2: Decision-process in the OG: Example calculation for target selection

Target ID	π_i	$ed_{i,j}$ **[km]**	$S_{i,j} = 0.5 \cdot \left(1 - \frac{ed_{i,j}}{14}\right) + (1-0.5) \cdot \left(\frac{\pi_i}{10000}\right)$
1	8000	10	$0.5 \cdot \left(1 - \frac{10}{14}\right) + 0.5 \cdot \left(\frac{8000}{10000}\right) = \mathbf{0.54}$
2	3000	3	$0.5 \cdot \left(1 - \frac{3}{14}\right) + 0.5 \cdot \left(\frac{3000}{10000}\right) = \mathbf{0.54}$
3	10000	14	$0.5 \cdot \left(1 - \frac{14}{14}\right) + 0.5 \cdot \left(\frac{10000}{10000}\right) = \mathbf{0.50}$
4	8000	9	$0.5 \cdot \left(1 - \frac{9}{14}\right) + 0.5 \cdot \left(\frac{8000}{10000}\right) = \mathbf{0.58}$

Offline Calibration

It is important to note that the parameter α remains constant throughout the entire simulation. It is a fixed input to the algorithm, does not adapt over time, and is not optimized during execution. In practice, the best-performing value for α is determined via multiple simulation runs, which are performed with different α values, and the one yielding the highest performance is selected retrospectively. This approach is suboptimal, as it requires knowledge of the best α value in advance, an assumption that does not hold in real-world applications where such information is typically unavailable. Also, this requires multiple test runs, which results in a higher computational execution time.

Limitations of the OG Algorithm: Consequences of Fixed Assignments

The OG algorithm follows an online decision-making framework in which vessels are assigned to targets sequentially based on the current state of the environment. However, a key limitation lies in its treatment of these assignments as irrevocable: once a vessel is assigned to a target, it is committed to pursuing that target until collection, regardless of how the situation evolves. This rigid assignment strategy gives rise to two major inefficiencies, both of which stem directly from this lack of flexibility.

First, if a vessel is assigned to a target that later moves away due to ocean currents, the time to reach it can increase significantly. This is especially problematic if the vessel is slower than the target or if the distance grows over time. The OG algorithm does not re-assess target assignments during execution. As a result, a vessel may remain committed to an unreachable target, wasting time and reducing collection efficiency. This can severely lower performance, as no collection occurs for long periods.

Second, while en route to a target, a vessel may overlook new opportunities that might arise. Changing hotspot positions may cause a hotspot to drift close to another target, making it more favourable. OG does not allow mid-mission reassignment or path deviation; the vessel cannot exploit these emerging opportunities. This further reduces the system's adaptability and leads to missed opportunities for additional rewards at minimal extra cost.

These limitations both result from one design choice: treating initial assignments as final. This approach simplifies coordination and ensures consistency, but it reduces responsiveness to environmental changes. In missions with moving targets and changing conditions, a lack of reactivity can be a major drawback. Adding mechanisms for reassignment or re-planning could greatly improve adaptability and maximize reward as conditions evolve.

Runtime For the OG, a new target assignment is computed only when a vessel reaches its previous target. Each reassignment requires $O(N(t))$ operations over the remaining hotspots, leading to an overall runtime of approximately $O(B \cdot H)$.

3.5.3 Online Greedy Algorithm with Retargeting [OGR]

The OG algorithm assigns a target only if a vessel has none. This happens at the start of the simulation and when the vessel reaches its current target's radius. In the Online Greedy with Retargeting (OGR), this changes: Constraint No. 9 (Target persistence) no longer applies. Now, a vessel's assignment can change at every discrete step because a second scoring function allows it to reevaluate whether a new target is more promising than the old one.

Algorithm Description

The algorithmic structure is best understood through a simplified version of the pseudo code (see algorithm 3). The following outline demonstrates how the retargeting mechanism fits into the main control loop. In particular, it highlights the differences from the original greedy assignment logic, setting the stage for the inclusion of the new retargeting component.

The full retargeting mechanism is integrated into the main algorithm, as shown in algorithm 3. The algorithm operates over a discrete time horizon and iteratively makes routing decisions.

The first lines initialize the simulation at time step $t = 0$ and define the termination condition at the time horizon T. Then, the present positions of all vessels B and all active hotspots $N(t)$ are collected at each time step.

After that, the main control loop is started, where each vessel $j \in B$ If it has no assigned target, it selects the hotspot $i \in N(t)$ that maximizes the score given by the function $f(i, j, t, \alpha)$. This score combines the distance to the target at the present time step and the hotspots value, weighted by the trade-off parameter α, thereby balancing proximity and reward.

Now the core addition of the OGR algorithm follows, the retargeting mechanism. If a vessel already has an assigned target, it uses a second decision rule $g(\cdot)$ to check whether switching to another target would be beneficial. This evaluation happens at every time step and applies a new trade-off parameter β, leading to a distinct weighting of distance and value compared to

Algorithm 3: Greedy Heuristic with Retargeting for Plastic Retrieval

```
Input: Set of vessels B, hotspots H, time horizon T, assignment rule f(·), retargeting
       rule g(·), retarget threshold r, weighting factor α, opportunistic factor β
Output: Total collected value, Route of each j
t ← 0;
while t < T do
    Collect location information of all B and N(t);
    foreach vessel j ∈ B do
        if j has no assigned target then
            assign target i ← arg max_{i∈N(t)} f(i, j, t, α);
        if j has an assigned target i then
            best_candidate ← arg max_{i'∈N(t)} g(i, j, t, β);
            if Z_{i,j}(t') ≥ r · S_{i,j}(t) then
                reassign target i' ← best_candidate;
        move j towards target i;
        if j reaches i then
            collect i; remove i from N(t);
            assign new target based on f(·);
    t ← t + 1;
```

the original function $f(\cdot)$.
The scoring function used in the reassessment is defined as:

$$Z_{ij}(t) = \beta \cdot \left(1 - \frac{ed_{ij}(t)}{ed_{\text{max}}(t)}\right) + (1 - \beta) \cdot \left(\frac{\pi_i}{\pi_{\text{max}}}\right) \tag{3.3}$$

This formulation mirrors the structure of $f(\cdot)$ but applies a different weighting factor. Using a separate parameter for reassessment is essential, as it allows the algorithm to evaluate its environment from a different perspective.
If both $f(\cdot)$ and $g(\cdot)$ used the same weighting factor, the reassessment would likely yield identical results and prevent retargeting. The distinct parameter β, therefore, enables flexibility and responsiveness, particularly when hotspots drift and new opportunities arise nearby.

Moreover, the interpretation of β differs conceptually from α. While α governs the initial assignment and aims for a generally optimal trade-off between value and distance across the entire environment, β is designed for local, short-term decisions that occur along the way. It prioritizes nearby or newly emerged hotspots that can be collected with minimal deviation from the current route. This distinction allows the OGR algorithm to capture valuable intermediate opportunities that appear during movement, something the original OG approach cannot achieve. Although, in rare cases, environmental dynamics alone could still lead to different decisions even with identical weights, the introduction of a dedicated reassessment parameter explicitly strengthens this adaptive behaviour and systematically improves responsiveness to spatial changes.

It is intended to have both α and β as a team, where is to find a good target and β constantly checks if there is a target that is close enough to make a worthwhile detour. To avoid excessive switching, a retargeting threshold $r \in [0, 1]$ is introduced. A vessel changes its target from i to a new hotspot i' only if the new score is at least a factor of r higher than the current score:

$$Z_{i,j}(t') > r * S_{i,j}(t)$$

This rule ensures that only significant improvements trigger a target change. Small fluctuations, such as minor position shifts or temporary scoring noise, do not lead to reassignments. This makes the system stable under small perturbations while preserving the ability to react quickly when the environment changes considerably, for instance, when a new high-value hotspot drifts into proximity.

The main execution phase remains the same as in the OG. Each vessel moves toward its currently assigned target according to its velocity and environmental forces. When a target is reached, the hotspot is collected and removed from the list $N(t)$. Immediately afterward, the vessel assigns a new target using the original rule $f(\cdot)$, ensuring continuous operation throughout the simulation period.

Finally, the simulation time t is incremented, and the process repeats until the time horizon T is reached.

By re-evaluating its environment at each discrete time step, OGR resolves a key limitation of the original OG algorithm, which remains committed to its first assigned target regardless of environmental change. In OG, a vessel may continue pursuing a distant or drifting hotspot even when nearer and more rewarding options become available. OGR, in contrast, introduces a mechanism that enables adaptive retargeting, balancing persistence with flexibility and thereby achieves more efficient collection behaviour in dynamic marine environments.

Simplified OGR explained

The functionality of the OGR algorithm is best explained via a simplified illustration (see 3.11. The basic working mechanism of the OGR compared to the OG is observable, and the trajectory of the vessels (squares) to their targets (circles) is depicted. The first row shows the already known OR algorithm. The second row shows the OGR algorithm. The arrows indicate the target, which is assigned to a vessel.

This example has five discrete timesteps. Red circles show a target was collected in that timestep. A small α value is assumed, high-value hotspots are preferred. Only a nearby low-value hotspot would be targeted instead. Big circles are high-value hotspots (value = 20); small circles are low-value hotspots (value = 5).

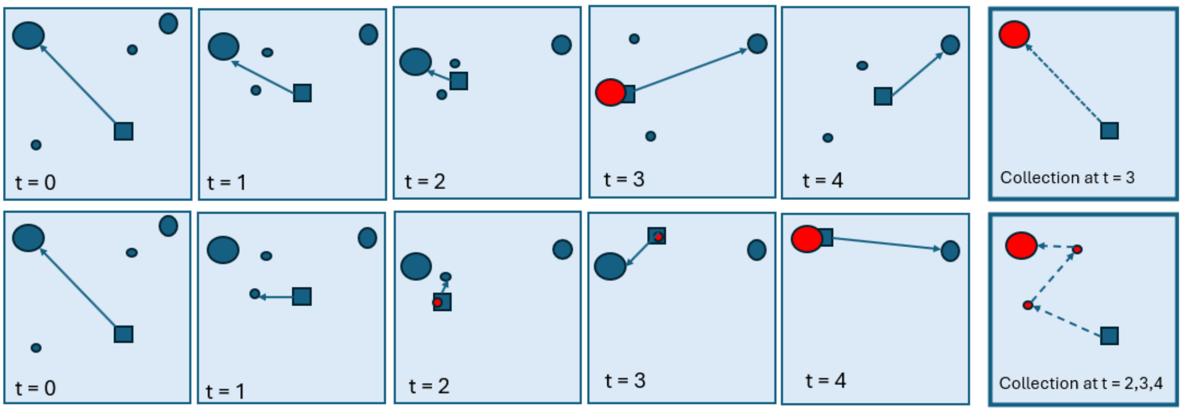

Figure 3.11: Comparison of vessel trajectories towards hotspots: OG vs. OGR.

At timestep $t = 0$, the upper left target is assigned to both algorithms. At this time, both use $f(\cdot)$ to evaluate the environment and pick the first target. At $t = 1$, the algorithms behave differently. OG moves toward its assigned target. OGR, however, shifts to another hotspot because it reconsidered the assignment with the decision rule $g(\cdot)$; the new assigned target hotspot is much closer, making fast collection more likely.

OGR reaches and collects this new target at $t = 2$. It then assigns a new nearby target with $f(\cdot)$, showing the algorithm's ability to react to changes. In the same timestep, OG has not collected any target. It stays committed to its first target and continues toward it. Even if other targets are closer, OG does not reconsider its initial choice.

In the next timestep (t=3), the OG finally reaches its initially assigned target. Only now will it use the scoring function $f(\cdot)$ again to evaluate its environment and commit itself to the upper right target. However, this will never be reached because the time runs out before the vessel reaches it.

By the time OG collects its first target, OGR has already collected its second. Now, OGR also targets its original hotspot again after considering $f(\cdot)$ and reaches it at $t = 4$. OGR will then also aim for the upper right target, but will not reach it in time.

The collection history of both algorithms can also be observed in 3.3

OG ends up closer to the upper right target than OGR. However, this does not matter: the goal is to target and collect hotspots that maximize value. Neither algorithm can reach the upper-right target. OGR, though, fills its remaining time by collecting nearby smaller targets. This example highlights OGR's key feature: it always reconsiders assignments. It shortens the time between collections and adapts to a changing environment. The next sections analyse detailed functionality.

Table 3.3: Stepwise comparison of OG and OGR algorithm behaviour over time

Timestep (t)	OG Algorithm	OGR Algorithm
0	–	–
1	–	–
2	–	Collection of small target $\rightarrow$ Value = 5
3	Collection of big target $\rightarrow$ value = 20	Collection of small target $\rightarrow$ value = 5
4	–	Collection of big target $\rightarrow$ value = 20
Sum	Value = 20	value = 30

Runtime of the OGR For the OGR, retargeting is evaluated at every discrete time step. Each vessel recalculates scores for all H hotspots, resulting in a per-step effort of $O(B \cdot H)$ and a total runtime scaling linearly with the time horizon, $O(T \cdot B \cdot H)$. Thus, OGR is computationally more demanding than OG.

Offline Calibration

As with the parameter α in the OG algorithm, the additional parameters β and r in OGR are not internally optimized. They are a fixed input to the algorithm, do not adapt over time, and are not optimized during execution. Instead, their values are determined through a grid search across multiple simulations, where different combinations are tested and the best-performing configuration is selected retrospectively. This is done because the focus of this work is not on optimizing algorithm parameters but on analysing how to make decisions with moving targets in an online framework.

3.5.4 Comparison and Overall Structure

Table 3.4 gives an overview of the differences and similarities between the OG and the OGR. The general structure of the approach is visualized in 3.12.

Table 3.4: Comparison between the OG and OGR algorithms

Criterion	**OG**	**OGR**
Decision Trigger	Only when a vessel has no active target.	Re-evaluated at every time step.
Decision Rule	Single decision rule $f(\cdot)$ with trade-off parameter α.	$f(\cdot)$ for initial assignment and $g(\cdot)$ for reassessment, using parameters α, β, r.
Retargeting Mechanism	–	Supported; vessel may switch if a new target's score exceeds the current one by a threshold factor r.
Responsiveness to Environment	Reactive only after collection events - Low flexibility	Continuously adaptive at every discrete time step - High flexibility
Coordination Between Vessels	None	None
Runtime Complexity	$O(B \cdot H)$.	$O(T \cdot B \cdot H)$.
Parameter Tuning Effort	α	α, β, r

The overall process is divided into two main components. The first part focuses on identifying the optimal algorithm parameters through an offline grid search, while the second part deals with the actual online execution of the heuristic. It is crucial to understand how these two parts interact. In essence, the approach involves two interrelated optimization problems:

1. **Offline optimization:** determining the algorithmic parameters that define the behaviour of the scoring function through a grid search, and
2. **Online optimization:** maximizing the total collected value during execution by selecting hotspots based on the predefined scoring function.

The decision variable at the core of this thesis is $z_{ij}(t)$, which determines whether a hotspot is targeted at time t (and thus directly affects $x_{ij}(t)$ and the overall collected value). The value of $z_{ij}(t)$ is governed by the outcome of the scoring functions $f(\cdot$ and $Z(\cdot)$, whose structure and weighting are defined by the optimized parameters α, β and r obtained during the offline phase.

It should be noted that the first phase, the offline grid search, technically requires full knowledge of the trajectories of all moving targets. This is because the parameter tuning process repeatedly simulates the OG and OGR algorithms under complete information to determine the best-performing parameter settings. In a real-world online scenario, such information would

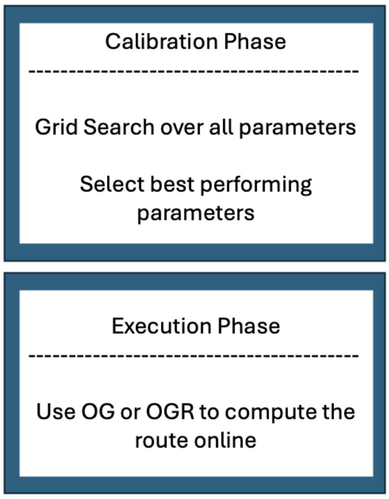

Figure 3.12: Two steps in the general approach: Offline calibration and online decision-process

not be available in advance. Consequently, this creates a conceptual inconsistency between the offline calibration and the online decision-making process.

However, this assumption is deliberately accepted in the context of this thesis. The primary aim is not to model a fully realistic operational pipeline, but to explore how the proposed heuristics behave under both optimal and non-optimal routing strategies. In particular, the study examines what flexibility means in this context (OG vs. OGR) and how simple, interpretable decision rules can be applied to problems involving moving targets. The use of pre-optimized parameters, therefore, serves as a methodological simplification, enabling a clearer analysis of the decision logic and performance characteristics of the OG and OGR heuristics within dynamic maritime environments. In reality, this knowledge could be replaced by data about simulated or observed hotspot behaviour, in order to learn from the data and to select optimal parameters without the grid search approach.

4 Testings and Discussion

4.1 Data

This study uses ocean current data from the Copernicus Marine Environment Monitoring Service (CMEMS)(Service (2024)) to simulate the movement of plastic and vessels on the ocean surface. CMEMS is part of the European Union's Copernicus program. It is operated by Mercator Ocean International. CMEMS provides free marine data that supports environmental monitoring, maritime safety, research, and sustainable ocean use.

CMEMS data, regularly updated on ocean physical, ice, and biogeochemical conditions, supports EU directives, international agreements, and Blue Economy activities. Publicly sharing this information promotes marine awareness and informed maritime decisions.

The thesis uses a global ocean forecast model, which integrates physical simulations with observational data from satellites and ocean sensors. The specific product is the Global Ocean Physics Analysis and Forecast dataset (ID: GLOBAL_ANALYSISFORECAST_PHY_001_024), providing hourly surface current data on a global grid.

For this study, only the surface current components were used. The data provides two values for each time and location: one for east–west flow and one for north–south flow. These values correspond to the horizontal velocity of ocean surface water at about 0.47 meters depth. This level is suitable for modelling transport of floating plastic debris and surface vessels. The dataset does not explicitly store velocity vectors. The two components are combined during simulation to create a movement vector for each location and time step.

The dataset covers the time period from June 1 to June 20, 2024, and the region around the Great Pacific Garbage Patch. Hourly current fields are provided in NetCDF format. These fields are read directly into the simulation framework. For each moving object, current velocities are interpolated from the data to its position at every time step. These values update the object's location step by step.

Besides the current data, no additional datasets were required for this study. There is no publicly available high-resolution dataset with real-time or spatially detailed information about floating ocean plastic. As a result, both plastic hotspots and collection vessels were simulated synthetically within the modelling framework. This approach allows controlled experimentation under defined initial conditions. It enables systematic testing of routing strategies without relying on incomplete or uncertain observational plastic data.

4.2 Simulation Parameters

The solution algorithms are tested under variable input parameters, which directly affect the size and complexity of problem instances.

$timeframe$ Total simulation horizon T in hours.

$plasticradius$ Radius of each simulation setting in [km].

$plasticnumber$ Number of plastic hotspots initialized in the simulation.

$plasticseed$ Random seed for reproducible hotspot initialization.

$boatnumber$ Number of homogeneous vessels in the simulation.

These parameters serve as the input for the algorithms and determine the complexity of the corresponding optimization problems. Increasing the number of vessels or hotspots, or enlarging the time horizon, results in higher computational complexity and greater combinatorial search space.

4.3 Evaluation Setup

To evaluate the performance of the proposed algorithms and answer the research questions, several experimental settings are tested and compared. The main evaluation metric is the obtained objective value, representing the total collected hotspot value within the defined time horizon. The experiments systematically analyse three aspects: how the weighting factor α in the scoring function affects target selection and overall performance (RQ1); whether there are generalizable patterns in optimal parameter settings across scenarios (RQ2); and how the proposed online heuristic algorithms compare to static baselines (RQ3).

The results depend strongly on the simulation setup. Different parameter configurations are tested. The default configuration uses a time horizon of 100 hours, 50 hotspots distributed around the map centre within a 10 km radius, and two vessels. This setup is systematically changed. For example, the time horizon may be increased, or the number and distribution of hotspots modified, to explore scenario-dependent performance patterns. Scenario setups are listed in table 4.1.

ID	Scenario	Time	Hotspots	Radius	Vessels
1	Default	100	50	10.0	2
2	Short Time	50	50	10.0	2
3	Long Time	500	50	10.0	2
4	Few Hotspots	100	20	10.0	2
5	Many Hotspots	100	300	10.0	2
6	Narrow Radius	100	50	2.0	2
7	Wide Radius	100	50	30.0	2
8	Few Vessels	100	50	10.0	1
9	Many Vessels	100	50	10.0	5
10	Vast and Dense	100	300	30.0	1

Table 4.1: Scenario overview

Building on these setups, the scenarios are chosen to examine key aspects of system behaviour and address the research questions. Each scenario changes specific parameters, such as time, number of hotspots, area size, and vessel count. This focused approach helps analyse how different environmental and operational factors impact performance. The considered scenarios are as follows:

- **Default:** Serves as the baseline configuration with moderate time, hotspot density, and vessel count for balanced performance evaluation.
- **Short Time:** Tests the system's efficiency under strict time constraints, emphasizing rapid decision-making and prioritization.
- **Long Time:** Evaluates long-term behaviour and the ability to maintain effective collection strategies over extended horizons.
- **Few hotspots:** Represents a sparse environment to analyse vessel utilization and idle-time effects when targets are limited.
- **Many hotspots:** Models a dense target field to test scalability and coordination among vessels in high-density conditions.
- **Narrow Radius:** Simulates difficult collection conditions where precise navigation is required due to a small capture radius.
- **Wide Radius:** Explores the opposite case with easier target acquisition, emphasizing routing over precision.
- **Few Vessels:** Test system performance under limited resources and potential overload of available vessels.
- **Many Vessels :** Analyses coordination efficiency and potential redundancy when resources are abundant.

- **Vast and Dense:** Combines a large, dense area with few vessels to evaluate performance under challenging coverage and workload imbalance.

Based on the scenarios above, the following tests will be conducted in the subsequent section:

- **Test 0:** Examines how the objective value changes across different α values in the default scenario (RQ1), providing an initial understanding of the balance between value and proximity in the OG algorithm.
- **Test 1:** Determines the optimal α values across scenarios (RQ1, RQ2) and identifies consistent weighting patterns under different environmental conditions.
- **Test 2:** Compares OG, OGR, and baseline algorithms (RQ3) to evaluate the performance impact of online adaptation and retargeting.
- **Test 3:** Evaluates the computational runtime and efficiency of the proposed algorithms.
- **Test 4:** Assesses the generalization of averaged parameter settings (RQ2) by comparing averaged-weighted vs. individually calibrated configurations in the default scenario.

Why travelled distance is not indicated Figure 4.1 illustrates that the total travelled distance of vessels varies only slightly between different seeded runs within the same setting. While one might consider travelled distance a potential performance indicator, it is only marginally influenced by the decision-making process. When a vessel approaches a hotspot, its speed is automatically reduced to prevent overshooting, which introduces minor deviations in total distance.

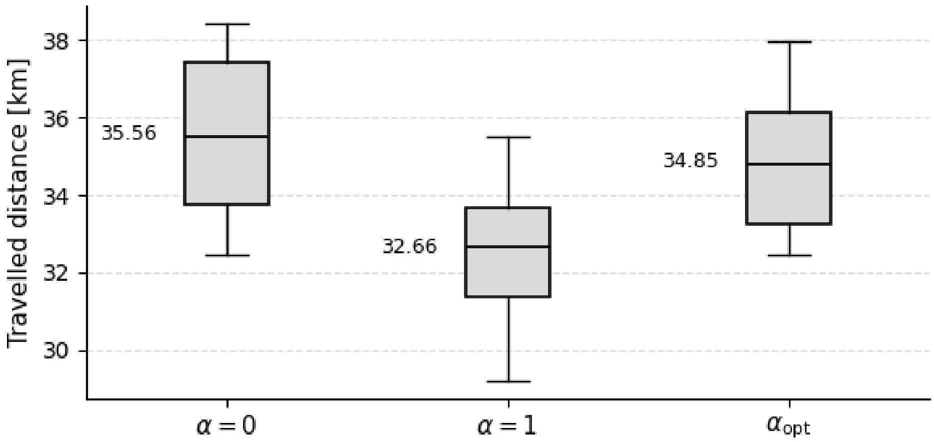

Figure 4.1: Mean of total travelled distances over 10 runs for $\alpha = 0, 1, opt$ values in default scenario

Since the time horizon is fixed and all vessels move at the same base speed, with only small variations caused by ocean currents, the overall distance travelled remains within a relatively narrow range across identical configurations. For instance, the mean travelled distance in the default setting over 10 seeds changes by only about 2.9 km between $\alpha = 0$ and $\alpha = 1$ (see Table 8.1) over multiple seeds. Furthermore, as the routes between targets are not explicitly optimized (vessels simply move in the direction of the currently selected hotspot) this metric does not meaningfully capture algorithmic differences. Consequently, in the following analysis, the focus is on the collected objective value as the primary performance metric.

4.4 Test 0: Performance of different α Values in the default Scenario with OG

Test 0 is conducted to examine how the objective value changes for different α values under the default configuration with the OG Algorithm. This test provides an initial understanding of the relationship between the weighting of value and proximity in the scoring function. The results offer an early look at the performance of α values (RQ1). In this preliminary experiment, the default scenario is used to analyse the impact of varying α values on the overall objective value. In the first step, the alpha distribution will be investigated for one seed in the default scenario; then, the averaged distribution over multiple seeds for the default setting will be examined.

Analysis: Figure 4.2 shows a non-linear and sensitive relationship between α and the achieved objective value. For very small α values ($\alpha \approx 0.0$), performance is moderate but falls to a local minimum around $\alpha = 0.3$–0.4. Beyond $\alpha = 0.4$, performance climbs steeply and peaks near $\alpha \approx 0.6$, suggesting that a balanced trade-off between proximity and value yields the best results. For higher α values (> 0.6), moderate fluctuations occur, followed by a clear decline after $\alpha > 0.9$, Excessive proximity focus leads vessels toward low-value, easily reachable hotspots.

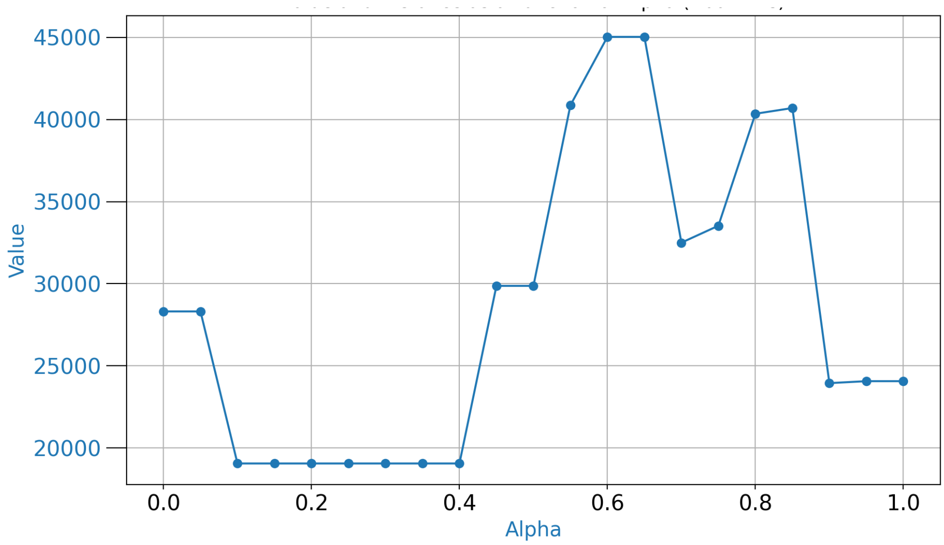

Figure 4.2: Objective value as a function of different α values, one run in default scenario

This preliminary test demonstrates that a balanced weighting (α)around 0.6 achieves the highest objective values, while both extreme and intermediate unbalanced weightings lead to suboptimal strategies. The test offered initial insights into Research Question One, examining whether an extreme focus on one criterion outperforms a balanced strategy. The results indicate that neither selection purely based on value ($\alpha = 0$) nor purely on proximity ($\alpha = 1$) is optimal. Rather, a balanced combination yields the best collection performance, confirming that the greedy decision process benefits from a moderate α, where both spatial efficiency and value potential are considered.

However, Figure 4.2 represents only a single seeded run of the default scenario, which primarily serves to illustrate how crucial a careful selection of the α value is. Even minor adjustments can result in substantial increases or decreases in the objective value, demonstrating the system's high sensitivity to parameter changes. Therefore, it is a reasonable next step to test over multiple seeds to answer Research Question One. For this purpose, all α values from 0 to 1 in steps of 0.05 in the default scenario were tested over 44 seeds. 4.3 represents the results. Detailed results can be found in Appendix 8.2.

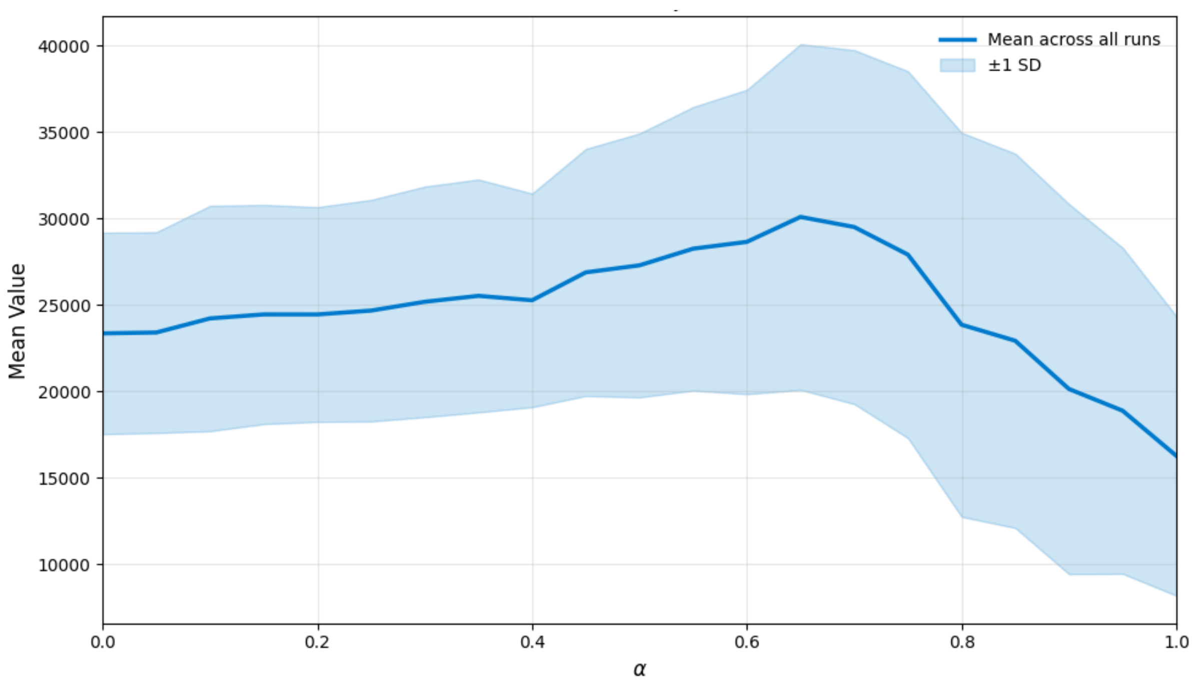

Figure 4.3: Objective value as a function of α values (mean of 44 runs in the default scenario)

The figure illustrates the mean objective values across 44 simulation runs for varying α values between 0 and 1. The solid blue line represents the mean performance. The shaded area indicates the range of one standard deviation (SD) around the mean. The results show a general increase in objective value up to approximately $\alpha = 0.65$, where performance peaks. Beyond this point, the mean value decreases notably, and the variance increases. This suggests moderate α values lead to the most stable and efficient outcomes. Extreme values of α result in less consistent performance across runs. Overall, the results demonstrate that moderate α values achieve both high and reliable objective values.

Based on this test, Research Question One can be preliminarily answered: neither edge solution yields optimal results in the default setting, while a moderate choice of α around 0.65 achieves the best results. This finding applies only to the default scenario. The next test will determine whether the optimal α of 0.65 also provides the best results in different scenarios and will compare the edge solutions with this optimal value in those cases.

Findings from Test 0

- **Finding 1: Balanced weighting achieves the highest performance.** The results clearly show that neither extreme configuration—pure value focus ($\alpha = 0$) nor pure distance focus ($\alpha = 1$)—is optimal. The best performance is achieved with a moderate weighting around $\alpha \approx 0.65$ (for the default scenario), confirming that an equilibrium between proximity and value yields the most effective collection strategy.

4.5 Test 1: Variation of optimal α Values across Scenarios with the OG

The purpose of this test is to address Research Question Two (RQ2) by determining if generalizable insights about the optimal weighting strategy can be identified, and to assess whether scenario-specific characteristics influence the choice of α. Additionally, the test provides a conclusion for Research Question One (RQ1) by examining whether an edge solution can ever be optimal.

The analysis focuses on understanding how α influences the greedy decision-making process. It addresses whether environmental or structural factors cause systematic variations in the optimal weighting. The main emphasis is on the relative improvement of the optimal α configuration over the edge cases ($\alpha = 0$ and $\alpha = 1$). The absolute objective values of static baselines are not considered.

Table 4.2 summarizes the average results across 20 seeds per scenario (200 runs in total). It shows the objective values for each edge case, the corresponding optimal α, and the relative improvement over the best-performing edge solution. Detailed per-seed results are provided in the Appendix (see: Table 8.3, Table 8.4, Table 8.5, Table 8.6, Table 8.6, Table 8.7).

Table 4.2: Comparison of objective values for the edge cases $\alpha = 0$ and $\alpha = 1$ against α_{opt}, averaged over all seeds per scenario, including the relative improvement over the better edge case.

Scenario	$V_{\alpha=0}$	$V_{\alpha=1}$	V_{opt}	α_{opt}	Improvement [%]
1	24 354	21 304	34 421	0.64	34.5
2	12 985	11 473	19 693	0.70	34.5
3	54 922	64 617	65 820	0.90	2.1
4	21 995	18 760	28 867	0.66	18.5
5	26 645	30 930	57 006	0.75	78.9
6	54 116	60 649	64 664	0.84	6.7
7	11 838	12 044	19 474	0.73	39.1
8	15 663	11 093	21 330	0.61	38.2
9	40 553	39 351	53 325	0.77	22.1
10	4 762	5 316	12 344	0.66	104.7
Mean	26 683	27 354	37 294	0.73	37.8

Comparison of Edge Case vs. Weighted Strategy Across Multiple Scenarios Using the OG

The findings from the preliminary test are confirmed by the results of Test 1 for all scenarios. Out of a total of 200 individual test runs, only 21 cases (10.5%) yielded an edge case ($\alpha = 0$ or $\alpha = 1$) as the optimal value. Among these, 12 corresponded to $\alpha = 0$ (pure value-based selection) and 9 to $\alpha = 1$ (pure proximity-based selection).

Comparing objective values across all scenarios (see Table 4.2 and 4.4), the advantage of a weighted strategy is clear. Figure 4.4 visualizes the table 4.2. In every scenario, the weighted approach outperforms the edge cases on average. Using the optimal α_{opt} yields a relative improvement of about 37.7% compared to the best edge case. In extreme scenarios, like Scenario 10, improvement exceeds 100%, demonstrating the substantial effect of the weighting parameter. In Scenario 3, where α_{opt} approaches 1.0, improvement is marginal (around 2%), suggesting some distributions or conditions may favour an extreme.

Overall, these findings provide a clear answer to Research Question 1: in the vast majority of cases, an edge solution is not optimal. A balanced weighting between value and proximity consistently yields better collection performance, confirming that a well-calibrated α offers a more effective decision process.

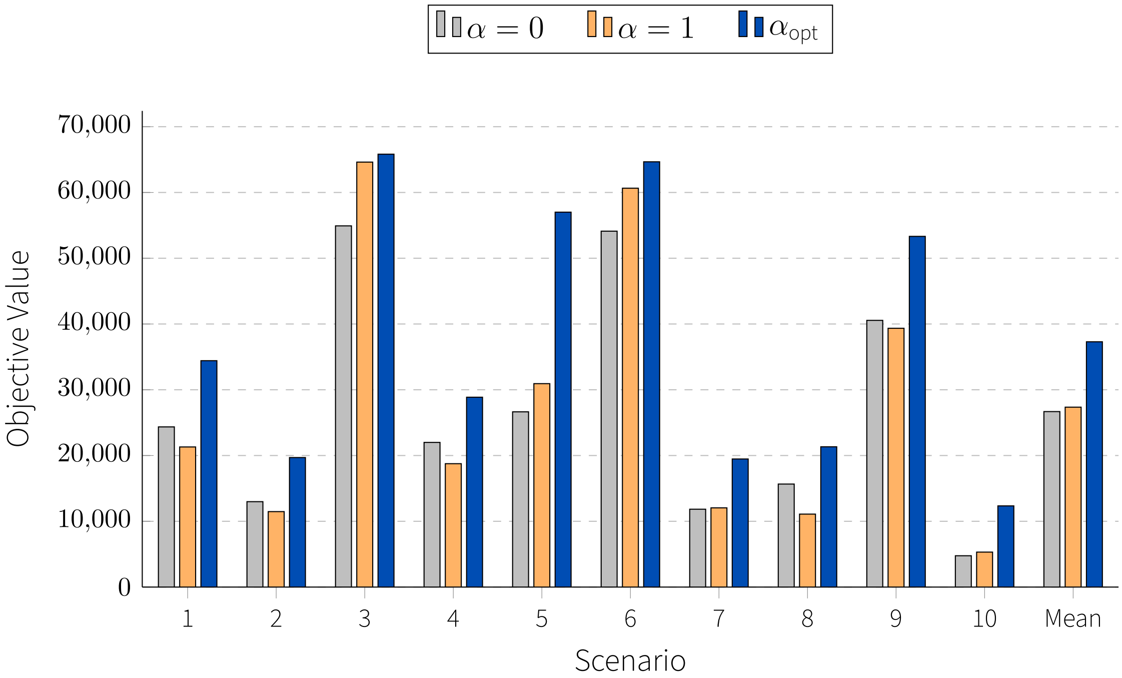

Figure 4.4: Visualization of Table 4.2. Comparison of the mean total collected objective values for $\alpha = 0$, $\alpha = 1$, and α_{opt} per scenario.

Scenario-Specific and Cross-Scenario Patterns in the Distribution of α_{opt} and Objective Values - RQ2

With this confirmation that a weighted approach is generally superior, Research Question Two becomes increasingly relevant. The following section now builds on these results to explore R2: Can general assumptions be made about which weighting strategy is optimal across scenarios, and to what extent do such patterns persist? Additionally, do scenario-specific characteristics systematically influence the optimal choice of α?

To investigate these questions, the analysis proceeds in two steps. First, the overall distribution of optimal α values is examined to assess whether general conclusions can be drawn about the balancing of value and proximity. Subsequently,scenario-specific patterns are examined to identify whether particular environmental or structural factors lead to distinct α_{opt} values or performance outcomes.

Overall Distribution Figure 4.5 illustrates the frequency with which specific α values were found to be optimal across all 200 test runs for each of the scenarios 10, serving as a bridge between the general findings and subsequent scenario-specific insights. The distribution of α_{opt} shows a clear dominance of medium to high α_{opt} values, typically ranging between 0.6 and 0.9. This confirms that, in most environmental settings, a stronger emphasis on proximity improves collection performance while still balancing hotspot value. One interesting finding is that figures 4.5, 4.2, and 4.3 all show a similar trend in which α values perform well. α = 0 yields results that are not optimal. However, they perform better than the other extreme case of α = 1. If α is set low (around 0.2 to 0.4), the objective value increases, but these α values are almost never optimal. From α = 0.6 to α = 0.8, there is a peak in quality. These α values are optimal in most cases. However, there is a decline in solution quality when α approaches 1. In general, the weighted solution of α_{opt} across all scenarios is 0.73, which yields the best results.

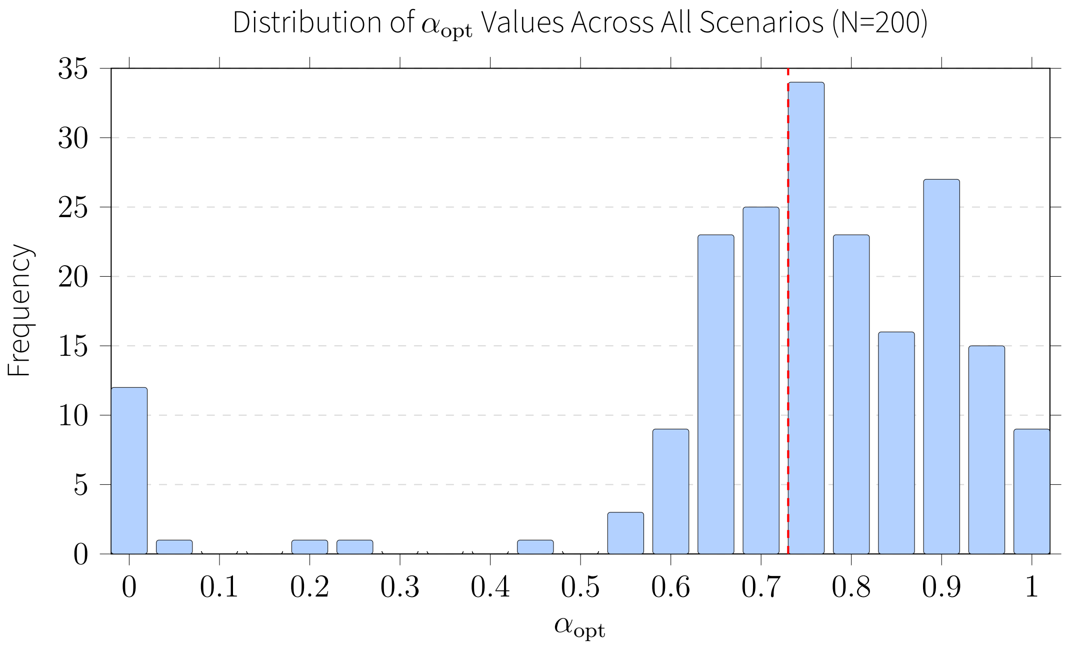

Figure 4.5: Frequency distribution of optimal α_{opt} values across 200 test runs for all scenarios. The red line indicates the overall mean (0.73).

Scenario-specific analysis Despite this general trend, noticeable scenario-specific differences emerge (see 4.6).

This box plot shows which α values have been optimal for each scenario over all seeds. For each scenario, 20 seeds were tested, resulting in 20 runs per scenario. Scenarios 3 (long time) and 6 (narrow radius) exhibit the highest α_{opt} value. This could suggest that in scenario 6, nearby targets yield high returns due to the dense distributions. In scenario 3, however, the time horizon is extended, meaning there could be the potential to collect all or almost all hotspots if there are no unnecessarily long routes. Scenarios 1, 4, and 6 show broad distributions, indicating that the optimal weighting is more sensitive to stochastic variations, likely because hotspot value and spatial dispersion vary more strongly.

Interestingly, low α_{opt} outliers (close to 0) appear sporadically across several scenarios but never dominate the distribution. These rare cases likely occur when a few exceptionally high-value hotspots justify longer detours. Overall, the scenario-specific analysis highlights that while the precise α_{opt} may shift slightly depending on spatial or value heterogeneity, the majority of scenarios favour a balanced-to-proximity-weighted strategy.

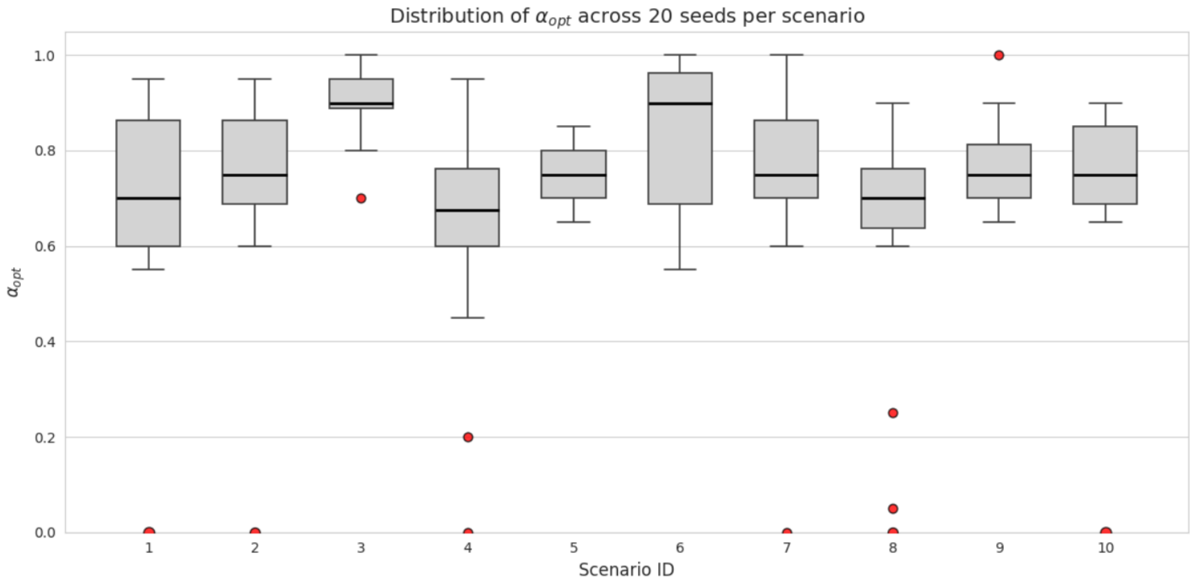

Figure 4.6: Box plot showing the distribution of optimal α values per scenarios.

After comparing the overall distribution of α_{opt} values across scenarios, each scenario is now examined individually. For this purpose, an example route is constructed using the optimal α_{opt} values for each scenario, that was obtained by averaging over all seeded test runs for each scenario(summarized in Table 4.2). In each case, the averaged optimal α is applied to the corresponding scenario configuration.

This approach provides an illustrative understanding of how the greedy algorithm behaves under different weighting strategies and environmental conditions. By visualizing the resulting vessel trajectories, it becomes possible to observe how the balance between value orientation and spatial efficiency translates into concrete routing decisions.

In each trajectory visualization, the small grey dots represent the hotspots , specifically their initial positions at the start of the simulation. The thin grey lines illustrate the drift trajectories of these hotspots over time. The central square marks the starting position of the vessel(s). Coloured lines depict the movement of the vessel paths, and each coloured circle along a trajectory indicates a point at which a hotspot was successfully collected. Together, these visual elements provide a comprehensive view of both the environmental dynamics and the vessels' adaptive collection behaviour under the respective weighting configuration.

Scenario 1: Default In the normal scenario (ID 1), a moderate $\alpha_{opt} = 0.64$ resulted in a clear improvement of around 34%, representing a balanced condition where value and distance are of comparable relevance.

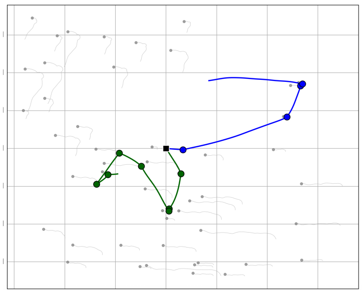

Figure 4.7: Example Trajectory for default Scenario with mean $\alpha_{opt} = 0.64$

Scenario 2 and 3: Long and short time horizon The short time scenario (ID 2) exhibited a slightly higher $\alpha_{opt} = 0.70$ and a similar improvement of roughly 35%. The shorter operational window favors strategies that consider distance, as long-range movements can prevent the vessel from collecting any hotspots within the given time horizon. However, focusing solely on distance is also not optimal, as this may result in missing high-value hotspots that are still reachable. This indicates that even under time pressure, a certain degree of value consideration remains beneficial.

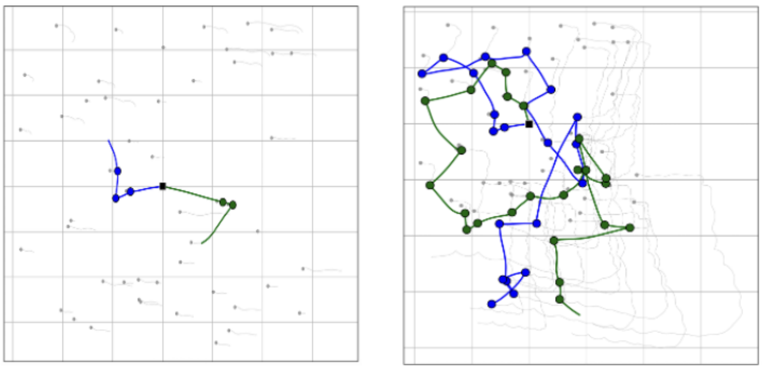

Figure 4.8: Example Trajectory for Scenario 2 (left) with mean $\alpha_{opt} = 0.7$ and for Scenario 3 (right) with mean $\alpha_{opt} = 0.9$

In contrast, the long time scenario (ID 3) with a very high $\alpha_{opt} = 0.90$ achieved the best objective value but with only marginal improvement of about 2%. This might appear counter-intuitive, since a longer time horizon allows vessels to explore freely and reach distant high-value hotspots, meaning a low α would also be reasonable. However, when sufficient time is available, most high-value hotspots will eventually become reachable and collected, even under a primarily distance-based steering strategy. This also becomes clear if it is considered that 46 of the 50 present hotspots are collected at the end of the time horizon. This means that adaptive balancing is less critical when time constraints are relaxed, as the boat can naturally explore over an extended period. As shown in the picture, even though the focus is almost entirely on distance, the travel distances are comparably long. But with a long time horizon, almost all near points are visited, and the vessel has to make longer tours in order to reach the next hotspot.

Scenario 4 and 5: Few and many hotspots The few hotspots configuration (ID 4) showed an $\alpha_{opt} = 0.66$ and a moderate improvement of around 19%. When targets are sparse, the algorithm must balance distance and value carefully to avoid inefficient movement or the selection of suboptimal nearby hotspots. This leads to a slightly distance-oriented but still balanced trade-off that ensures effective coverage.

In contrast, the many hotspots scenario (ID 5) displayed a much stronger effect, with $\alpha_{opt} = 0.75$ and an improvement of nearly 79%. In dense environments, moderate distance orientation allows efficient traversal while still exploiting high-value areas. Over emphasizing value in such cases would lead to unnecessary detours, while focusing solely on distance would neglect valuable regions.

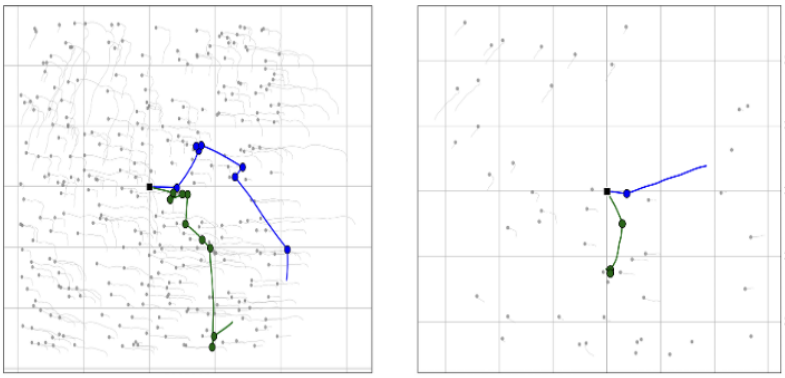

Figure 4.9: Example Trajectory for Scenario 4 (left) with mean $\alpha_{opt} = 0.66$ and for Scenario 5 (right) with mean $\alpha_{opt} = 0.75$

Scenario 6 and 7: Narrow and wide radius The narrow radius scenario (ID 6) resulted in $\alpha_{opt} = 0.84$ with a small improvement of about 7%. Here, movement is spatially constrained, meaning that hotspots can be collected quickly regardless of direction. Similar to the long-time scenario, most hotspots are collected eventually, and a distance-oriented steering approach is almost sufficient on its own. This scenario can therefore be considered relatively easy, as optimization yields only minor improvement.

Conversely, the wide radius scenario (ID 7) with $\alpha_{opt} = 0.73$ achieved an improvement of nearly 39%. When the vessel can collect hotspots from a larger radius, integrating value considerations becomes beneficial, as this allows the selection of high-value hotspots over a broader area. Unlike the narrow radius case, steering decisions here matter more because not all hotspots can be reached.

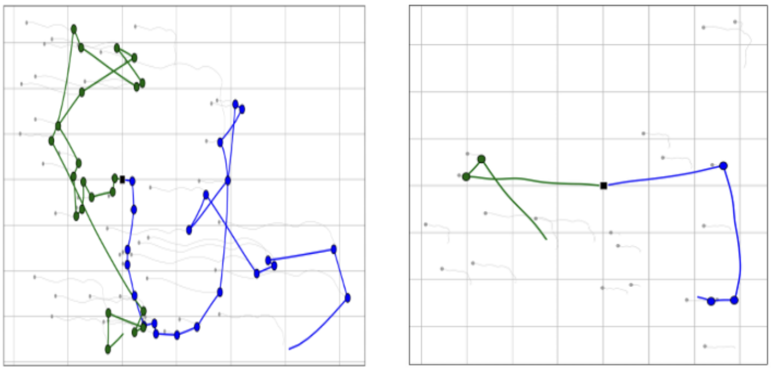

Figure 4.10: Example Trajectory for Scenario 6 (left)with mean $\alpha_{opt} = 0.84$ and for Scenario 7 (right) with mean $\alpha_{opt} = 0.73$

Scenario 8 and 9: Few boats and many boats Fleet size also shows a clear influence on the trade-off behaviour. With few boats (ID 8), $\alpha_{opt} = 0.61$, and an improvement of roughly 38% indicates that a balanced weighting between value and distance yields the highest efficiency. A single vessel must prioritize valuable hotspots while remaining efficient.

When the number of boats increases (ID 9), α_{opt} rises to 0.77 while the improvement drops to 22%. With multiple vessels, the system can cover the area more effectively, which favours a slightly stronger distance orientation to avoid overlapping trajectories. It would be interesting to examine how coordination between vessels might change this outcome, given that in the current setup, the boats act independently, potentially leading to suboptimal and overlapping paths.

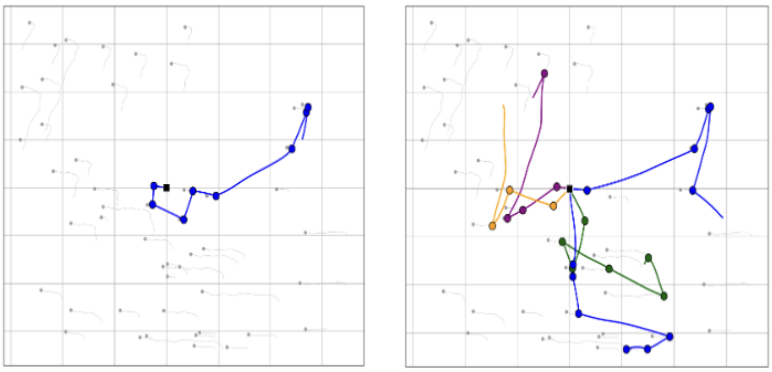

Figure 4.11: Example Trajectory for Scenario 8 (left) with mean $\alpha_{opt} = 0.61$ and for Scenario 9 (right) with mean $\alpha_{opt} = 0.77$

Scenario 10: Vast and dense Finally, the vast and dense scenario (ID 10) demonstrated the strongest effect, with $\alpha_{opt} = 0.66$ and an improvement exceeding 100%. This complex spatial structure demands dynamic weighting. Such highly structured environments particularly benefit from adaptive behaviour, since neither purely value- nor distance-based steering can adequately capture the heterogeneous spatial dynamics. In the trajectory, this can be observed as well. Especially because the scale is vast here (30 km radius), if the boat were to go for the nearest hotspot, this would still take up a lot of time.

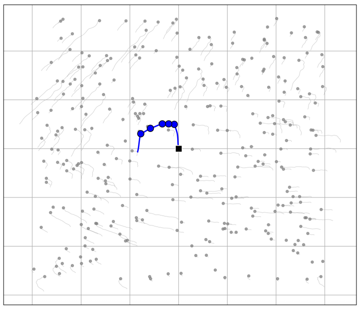

Figure 4.12: Example Trajectory for Scenario 10 with mean $\alpha_{opt} = 0.66$

Findings from Test 1

- **Finding 1: Weighted strategies outperform edge cases in all scenarios.** In almost all tests, an intermediate weighting between value and distance ($0.6 \leq \alpha_{opt} \leq 0.9$) led to superior results compared to pure edge cases. The average improvement across all scenarios was 37.8%, confirming that a balanced trade-off between proximity and value consistently enhances performance.

- **Finding 2: Scenario structure systematically affects α_{opt}.** Scenario-specific characteristics, such as time horizon, hotspot density, collection radius, and fleet size strongly influence the optimal α.

Conclusion of scenario-specific analysis A balanced trade-off, oriented slightly toward the distance side, consistently yields the best performance. Depending on the scenario, this trade-off shifts in magnitude, but in general, it remains within this range, which allows this information. Research Question Two can be answered. It is important to recall that the analysis so far has focused solely on the Online Greedy algorithm. The primary objective up to this point was to evaluate whether a weighted decision strategy based on proximity and value provides a meaningful improvement in target selection. The next step is to examine whether this strategy can be further enhanced by enhancing flexibility through retargeting. In this context, the Online Greedy with Retargeting (OGR) algorithm is evaluated and compared to the OG and the static baseline solutions (B1 and B2). This allows us to assess whether online adaptability yields measurable gains in the overall objective value and how important flexibility is for the TDOP-MT.

4.6 Test 2: Performance Comparison Across Scenarios

This experiment directly addresses Research Question Three. It examines how the proposed online heuristics perform against an offline static baseline in total collected value. It also checks if introducing online decision-making and retargeting improves overall performance. The experiment compares the Online Greedy (OG) and Online Greedy with Retargeting against each other as well as against the two baseline solutions, B1 and B2. The ten scenarios introduced earlier are evaluated under three different random seeds to capture variability. Another goal is to quantify the added value of dynamic decision-making in a changing environment. In particular, the experiment checks if flexible online heuristics, especially dynamic retargeting, offer measurable improvements over static, pre-optimized routes.

Both OG and OGR use multiple parameters, optimized through a grid search over relevant values. For the OG, the parameter α is tested in incremental steps of 0.5 to obtain performance trends.

$$\alpha \in [0.0, 1.0], \text{ tested in steps of } 0.05.$$

OGR undergoes a grid search for r and β parameter combinations, as detailed below.

$$r \in \{0.5, 0.6, 0.65, 0.7, 0.8, 0.9, 0.95, 1.0, 1.1, 1.2\},$$

$$\beta \in \{0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 0.99\}.$$

The retargeting threshold r is initialized at 0.5 to prevent switching to targets with less than half the current value. Similarly, β starts at 0.5 to allow moderate on-the-way diversions without causing major detours. Together with the tested α values in the same way as in the OG, this results in a total of 1848 parameter combinations per run for the OGR.

Aggregated performance comparison

In the following table (4.3, the averaged results over all three seeds are displayed (detailed results in the appendix 8.9,8.8, 8.10). The columns in the table provide a detailed breakdown of the results for each tested algorithm and scenario.

The set of columns dedicated to the OG shows α_{opt}, which is the optimal value for the parameter α. Value is the total score collected using this optimal α, and Δ_{B2} [%] shows the percentage improvement of OG over the B2 baseline solution. Following this is the OGR Algorithm. The table includes α_{opt} for the initial target selection and Value for the total score achieved. It also introduces β_{opt}, the optimal tradeoff parameter for the retargeting decision, and r_{opt}, the optimal threshold that a new target's score must exceed to trigger a switch. The performance is measured by Δ_{B2} [%], which indicates improvement over the B2 baseline, and Δ_{OG} [%], which shows improvement compared to the simpler OG algorithm. Finally, the Static section contains results for the baseline models. B1 represents the score from an idealized static optimization where hotspots do not move. B2 is the score from applying the static B1 route to the realistic, dynamic environment, serving as the primary benchmark for the other algorithms.

Table 4.3: Mean of optimal parameter settings and mean of obtained Objective Value per scenario (B1, B2, OG and OGR)

ID	Avg. Parameter and Results										
	OG			OGR						Static	
	α_{opt}	Value	Δ_{B2} [%]	α_{opt}	Value	β_{opt}	r_{opt}	Δ_{B2} [%]	Δ_{OG} [%]	B1	B2
1	0.78	32 312	+21.9	0.43	33 492	0.77	1.00	+26.4	+3.6	36 361	26 507
2	0.73	21 468	+1.8	0.07	22 783	0.88	0.73	+8.0	+6.1	25 563	21 088
3	0.80	47 819	+39.0	0.47	48 495	0.95	0.93	+41.0	+1.4	50 719	34 385
4	0.38	21 478	+13.4	0.17	22 315	0.78	0.90	+17.8	+3.9	24 647	18 937
5	0.68	54 698	+26.0	0.68	57 338	0.81	1.00	+32.1	+4.8	64 282	43 404
6	0.75	50 040	+40.1	0.53	50 746	0.93	1.00	+42.1	+1.4	51 054	35 718
7	0.70	20 141	-4.6	0.23	21 450	0.93	0.90	-1.2	+6.5	25 563	23 213
8	0.72	18 356	-4.2	0.55	20 526	0.87	1.07	+7.1	+11.8	24 325	19 160
9	0.68	40 011	+6.1	0.28	42 014	0.92	0.93	+11.4	+5.0	46 736	37 709
10	0.50	11 614	+16.4	0.55	12 410	0.68	0.80	+24.4	+6.8	10 696	9 974
M	0.67	31 794	+15.6	0.40	33 157	0.852	0.926	+21.4	+5.1	35 595	26 660

Finding one: B1 > B2 The results clearly indicate that B1, the static solution applied to the static problem, consistently achieves higher scores than B2, where the static plan is applied in a dynamic setting. The average score drops by 25.1% when moving from B1 to B2. This significant decrease highlights the main challenge: static plans struggle to maintain high performance when targets move. The data and figure illustrate how precomputed plans quickly become inefficient in dynamic environments, reinforcing the importance of adaptive strategies. These findings underscore that online heuristics, capable of making real-time adjustments, are crucial for addressing the time-dependent Orienteering Problem with moving targets.

Finding two: OGR and OG outperform B2 Figure 4.14 shows the average percentage performance improvement of the two dynamic algorithms, OG and OGR, compared to the static baseline B2. Results are plotted across the ten scenarios and represent the mean value over the seeds. Each bar shows the percentage by which each algorithm performed better or worse than the static reference solution B2.

A key finding from the results is that OG outperformed B2 in 8 out of 10 scenarios, while OGR outperformed B2 in 9 out of 10 scenarios. This demonstrates that the dynamically constructed routes of OG and especially OGR are better suited to dynamic environments where targets move, resulting in higher collected values compared to the static baseline.

Finding 3: OGR outperforms OG As shown in Table4.3 and visualized in Figure4.13, the Online Greedy with Retargeting algorithm consistently outperforms the basic Online Greedy (OG)

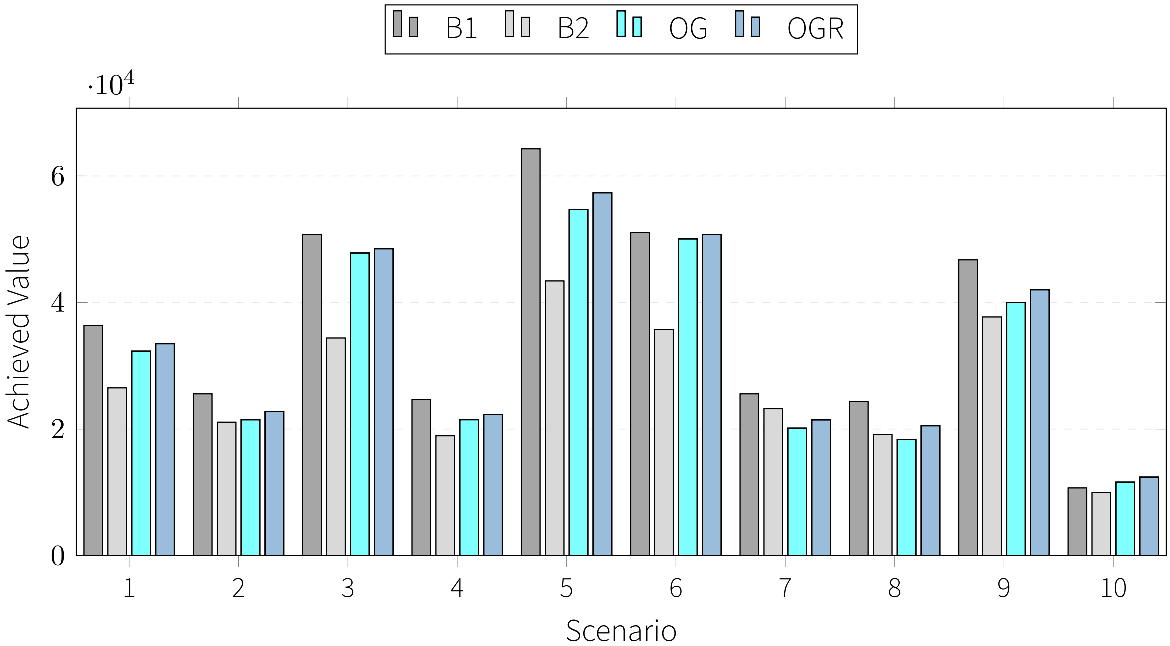

Figure 4.13: Visualisation of table 4.3: different averaged objective values per scenario across algorithms

approach. On average, OGR achieves a 5.1% higher objective value across all scenarios. The improvement is most pronounced in Scenario 8, where there is only one boat. The flexibility to revise target assignments during execution allows vessels to react to changing hotspot positions more efficiently. Even in cases with relatively stable environmental dynamics, OGR performs at least as well as OG, confirming that dynamic retargeting generally enhances collection efficiency without introducing instability into the decision process.

Finding 4: α_{opt} changes from OG to OGR The table 4.3 also reveals that the optimal α value decreases when retargeting is introduced. On average, α_{opt} drops from 0.67 in the OG algorithm to 0.40 in the OGR variant. This shift indicates that when dynamic retargeting is allowed, proximity becomes relatively more important than in the static greedy selection, as vessels can continuously adjust their routes to emerging opportunities. The scenario-wise variation of this effect is illustrated in Figure 4.15, which shows how the inclusion of retargeting systematically lowers the weighting parameter across most scenarios.

The graphical representation in Figure 4.15 confirms a clear strategic shift once retargeting is introduced. In 9 out of 10 scenarios, the optimal α for OGR is lower than for OG, indicating that proximity gains importance when adaptive re-planning is available. This shift results from the additional flexibility provided by the operational parameters β and r. While OG relies solely on α to balance value and distance, typically requiring higher α values (around 0.7), OGR can delegate short-term adjustments to the retargeting mechanism. Consequently, it operates efficiently with lower α values (around 0.35–0.55), focusing on local opportunities while maintaining a strategic orientation toward valuable targets. This effect is further illustrated in Figure

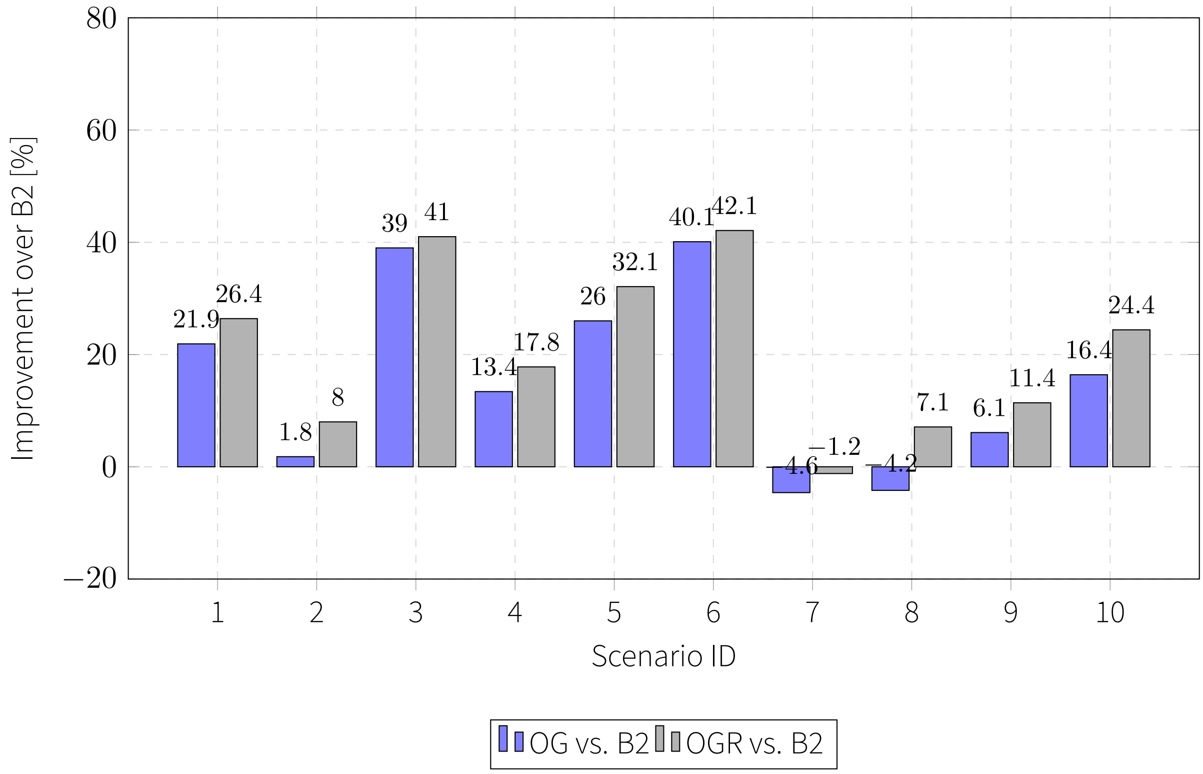

Figure 4.14: Average relative improvement of OG and OGR over B2 across scenarios.

4.16, which compares the objective values across different α settings for both algorithms.

This comparison highlights the substantial impact of the retargeting mechanism on the algorithm's overall behaviour. The two graphs illustrate how the total collected value varies with α, revealing a fundamental shift in strategy once retargeting is introduced. The right plot OGR clearly shows a lower optimal α and a consistently higher objective value than the left plot OG, with the entire performance curve shifted upward. This demonstrates that retargeting not only improves robustness but also increases overall efficiency, confirming the strategic advantage of adaptive online decision-making.

Finding 5: β and r The optimal value for β in the OGR over all scenarios corresponds to 0.852 and r to 0.926(see: 4.3). This is interesting regarding α, as the trend seems to be that α is used for navigating towards high-value hotspots, while β, which is very high, acts as the retargeting mode that checks every time step for a nearby hotspot. The r is also very high, meaning a switch will only be allowed if the potential replacement hotspots score from the retargeting decision rule is almost as good as the initial score from the assignment decision rule. Preventing aggressive switching.

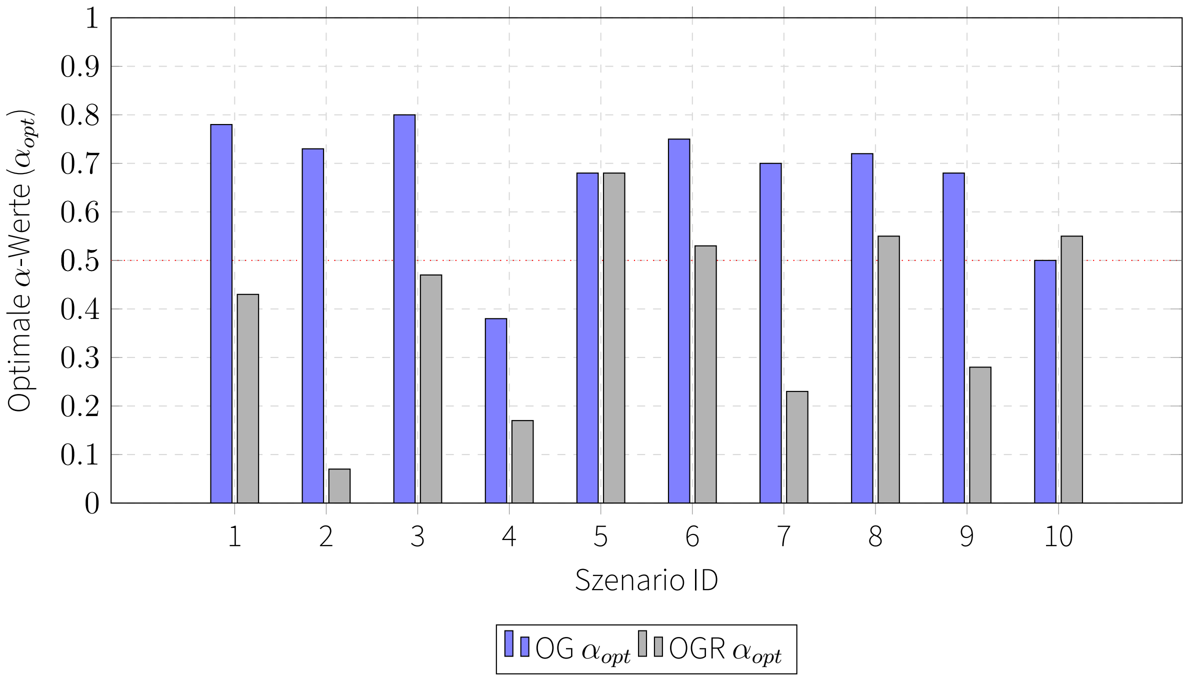

Figure 4.15: Comparison of optimal α-Values (α_{opt}) for OG and OGR over all Scenarios.

Findings from Test 2

- **Finding 1:** Static precomputed routes (B1) lose substantial performance when applied in dynamic environments (B2), with an average decrease of 25.1%, underscoring the need for adaptive, online approaches.
- **Finding 2:** Both online heuristics (OG and OGR) outperform the dynamic baseline (B2) in nearly all scenarios, with average improvements of +15.6% for OG and +21.4% for OGR.
- **Finding 3:** The inclusion of retargeting consistently enhances the performance of OGR, achieving on average 5.1% higher objective values than OG, demonstrating the value of dynamic re-planning.
- **Finding 4:** The optimal α shifts significantly downward (from 0.67 to 0.40) when retargeting is enabled.
- **Finding 5:** The optimal β and r serve to enable collection on the way to a high-value hotspot, but only if the quality of the new hotspot is sufficient.

Overall, these findings confirm that online, adaptive heuristics clearly outperform static baselines in dynamic environments. The ability to re-target in real time improves total collection performance of the decision process. Hence, the results provide strong empirical evidence in support of Research Question Three: introducing online decision-making and retargeting significantly enhances overall performance in the time-dependent Orienteering Problem with

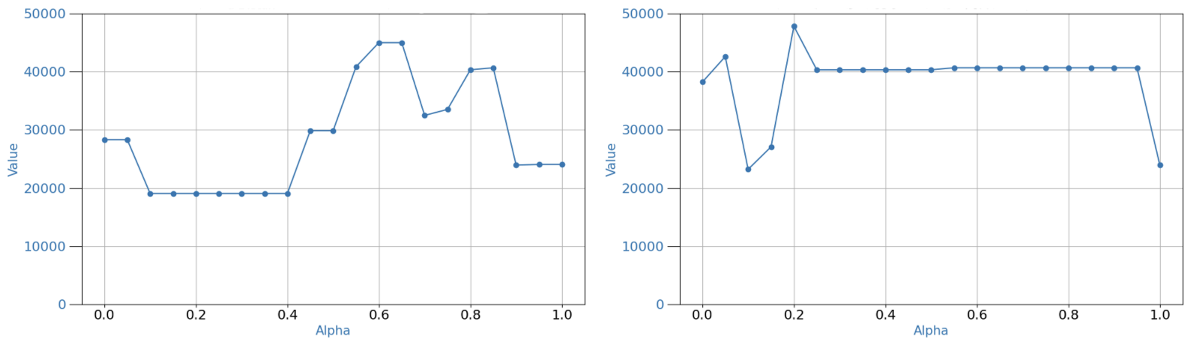

Figure 4.16: Objective value as a function of α values for one run in the default scenario. Left: OG; Right: OGR

moving targets. It also provides more information on Research Question Two, revealing that the optimal alpha will be lower if the ORG is used.

4.7 Test 3: Computation Time

As the previous tests have shown, the OGR algorithm appears to be the most suitable approach for the TDOP-MT. However, it is equally important to evaluate the computational efficiency of this method, specifically by determining how long the algorithm takes to execute at different stages, to properly assess its overall performance quality.

The computational run times of the various planning and execution components are compared in this section. All algorithms were executed on a MacBook with an M4 chip, and the recorded times represent the duration of the computational steps, which may vary depending on the hardware. Detailed results can be found in the appendix. The results are summarized in the table 4.4:

The benchmark solutions first require a significant offline optimization step to find the optimal route B1 for a static environment. As shown in the corresponding table, the time required to find the optimal B1 solution varied considerably across scenarios. For five of the ten scenarios, the process reached the maximum allocated execution time of 600 seconds (10 minutes) without converging to a verified optimum, highlighting the inherent computational complexity of solving the static Orienteering Problem (OP) optimally for large, complex instances.

The execution times for the Online Greedy (OG) heuristic are divided into two phases: parameter calibration and operational runtime. The process of finding the optimal α (α_{opt}) for the OG algorithm requires a search through multiple alpha values for each scenario. As shown in the table, these searches were generally short, ranging from 9 seconds to 40 seconds. This demonstrates that the parameter calibration for the simpler OG heuristic is more efficient. The operational runtime for a single execution of the OG algorithm using the determined α_{opt} is extremely fast. For most scenarios, the execution time was less than one second (< 1s), with the

Table 4.4: Range (Minimum, Maximum) and Mean of runtime for the parameter calibration and optimization execution steps across all 10 scenarios.

Process	Unit	Minimum	Maximum	Mean
Static Baseline Solution (B1 and B2)				
Find Optimal Static Route (B1)	Seconds (s)	20	600 (max)	437.4
Online Greedy (OG)				
Grid Search for α_{opt}	Seconds (s)	9	40	19.6
Single OG Execution	Seconds (s)	<1	4	1.3
Online Greedy with Retargeting (OGR)				
Grid Search for $\alpha_{opt}, \beta_{opt}, r_{opt}$	Seconds (s)	1 400	8 000	3 344
Single OGR Execution	Seconds (s)	<1	4	1.2

longest instance taking only 4 seconds. This confirms that, once a good α is found, the heuristic route can be found quickly, enabling real-time operation.

The OGR algorithm introduces two additional parameters, β and r, resulting in a considerably more complex parameter calibration phase. To find the optimal combination of parameters (α_{opt}, β_{opt}, r_{opt}) for OGR, a comprehensive grid search was performed, checking approximately 1,800 combinations for each scenario. As expected, the runtime for this grid search was substantially longer, ranging from 1,400 seconds to 8,000 seconds. These time values were rounded for presentation. This extended search time reflects the increased dimensionality and complexity of optimizing the parameters for the retargeting heuristic. However, the grid was very detailed in the execution and could be reduced in the future to reduce the runtime.

Similar to the OG algorithm, the operational runtime for a single execution of the OGR algorithm using the optimal parameters is very fast. Most scenarios executed in less than one second (< 1s), indicating that the overhead introduced by the retargeting mechanism during the actual mission execution is minimal. This confirms that the heuristic provides near real-time routing capabilities.

Findings from Test 3

- **Finding 1: Computational Complexity of the Static Baseline.** The static optimization approach (B1) required significantly longer computation times, often reaching the upper limit of 600 seconds without full convergence.
- **Finding 2: Efficiency of the Online Heuristics.** Both online heuristics (OG and OGR)

showed very low operational runtimes. Once the parameters were calibrated, single executions were completed in less than 4 seconds, confirming the suitability of both approaches for real-time applications.

- **Finding 3: Trade-off Between Calibration Effort and Flexibility.** The calibration phase for OG was lightweight (9–40 s), whereas OGR required substantially more time (1 400–8 000 s) due to multi-parameter tuning.

Given the significantly lower runtime of the stand-alone OG and OGR executions compared to the exhaustive grid search, using the averaged parameter configuration instead of individual calibration offers a substantial reduction in computational cost while maintaining a comparably high performance level. For this reason, this approach is further examined in the final experiment, Test 4.

4.8 Test 4: Evaluating the Generalization of Averaged Parameters in the default Scenario

The previous experiments compared the performance of the calibrated parameters in the OG and OGR against the baseline configurations B1 and B2. In this test, the performance of the OG algorithms is evaluated using the fixed, averaged weighting parameter ($\alpha_{mean} = 0.64$) from test one. This is compared to individually optimized α values to determine the importance of parameter calibration and examine the persistence of performance patterns (RQ2). We conduct the same comparison for OGR in the default scenario, using the optimal averaged parameter values from test 2. The goal is to determine how the generalized configuration performs relative to fully calibrated parameters. This comparison tests the robustness and transferability of the weighting strategy across different random seeds.

Online Greedy

Specifically, for the weighting factor $\alpha_{mean} = 0.64$, the average of the individual optimal α_{opt} values determined in Test 1 is used. This helps assess how well this configuration performs compared to the individually optimized settings.

Table 8.12 presents the results for the default scenario averaged across twenty random seeds. For each simulation run, the individually optimized value (V_{opt}) is compared to the performance obtained under the mean weighting configuration ($V_{\alpha=mean}$). The column Δ_{mean} indicates the relative percentage difference between these two values. On average, the mean configuration achieves a total collected value of $V_{\alpha=mean} = 30{,}997.7$, which is approximately 8.7% lower than the individually optimized value of $V_{opt} = 33{,}890.9$. Interestingly, one seed (#3000) even exhibits a slight positive deviation ($+0.91\%$), as the averaged parameter value α_{mean} does not necessarily coincide with any specific grid point used during optimization.

While it naturally falls short of fully calibrated parameters, the average loss of roughly 9% is relatively moderate. This finding is encouraging. It suggests that extensive calibration efforts might be avoided in future deployments, particularly when computational efficiency is a priority.

Online Greedy with Retargeting

To further test the generalization of averaged parameters, we applied the averaged parameters derived from Table 4.3 to the three seeds tested for the default scenario. These configurations of parameters might be even harder since now we have three parameters instead of just one. The resulting performance comparison is summarized below:

- Scenario **1111**: $V_{opt} = 41{,}337 \to V_{\alpha=mean} = 36{,}737$ (-11.1%)
- Scenario **5040**: $V_{opt} = 27{,}881 \to V_{\alpha=mean} = 24{,}592$ (-11.8%)
- Scenario **42**: $V_{opt} = 29{,}245 \to V_{\alpha=mean} = 24{,}720$ (-15.5%)

These results confirm the trend observed in the default setup: The mean weighting configuration consistently performs below the optimized settings. However, it only decreases by 15.5% maximum for this test. It would be a question of time how fast a solution needs to be computed, and if there is enough knowledge about what the trajectories will be like.

Findings from Test 4

- **Finding 1: Robustness of Averaged Parameters.** The results demonstrate that using the averaged weighting factor ($\alpha_{mean} = 0.64$) for the OG yields a relatively good performance across the tested seeds. Although the configuration performs approximately 8–9% below the individually optimized solutions, this deviation remains moderate. This supports the idea that a globally averaged parameter can serve as an effective compromise between accuracy and computational efficiency.
- **Finding 2: Trade-off Between Calibration and Runtime.** For the OGR algorithm, the averaged multi-parameter configuration produces slightly larger deviations (up to 15.5%), yet the results remain within an acceptable range. Considering that applying the averaged parameters entirely omits the computationally intensive grid search, the trade-off between reduced precision and substantially lower runtime is favourable for many practical applications where fast online decision-making is essential.

5 Relevance for the real World: Use Cases

Use Case One: Maritime Environment Simulation-based evaluation of the proposed algorithms provided key insights under controlled conditions, but a key question remains: how can these results help real-world operations? Specifically, how could ocean cleanup organizations use the findings of this thesis?

For example, The Ocean Cleanup (The Ocean Cleanup (2025)) operates a single collection system in the Great Pacific Garbage Patch (GPGP). Given high logistical and financial costs, mission planning relies on efficient routing and hotspot prediction. A core tool is the ADIS platform *Automated Debris Imaging System (ADIS)* (n.d.); de Vries, Egger, Mani, and Lebreton (2021). The Automated Debris Imaging System (ADIS) uses ship-mounted cameras to gather data on debris hotspots.

Analysis uses weeks of satellite data, both historical and recent, to estimate likely plastic hotspot locations. ADIS also provides 72-hour drift forecasts, letting operators anticipate hotspot changes. The algorithmic framework developed in this thesis, particularly the OGR heuristic, offers significant value. With real-time or near-real-time ADIS hotspot data, OGR could assign and reassign vessels to the best collection areas. The retargeting mechanism supports flexible adjustments as new hotspots emerge, drift, or vanish—addressing the operational dynamism that simulations did not capture.

OGR's online operation is beneficial because ADIS data is constantly updated, whereas a preplanned route may fail with new environmental data. Instead of a preplanned simulation, routing would occur on the water, allowing direct response to new data. For operational effectiveness, some adaptations are needed. Notably, instead of relying on simple Euclidean distance for travel cost decisions, ADIS's drift forecasts could be used to estimate travel time. By projecting both vessel and hotspot forward, a more realistic cost that reflects their probable meeting point could be used. Regarding parameter optimization, this process could also be based on past data to find well-suited parameters for the environment.

Figure 5.1 illustrates a potential integration cycle, combining ADIS forecasting with OGR-based route planning. The process begins with the identification of active hotspots based on ADIS data. These are then projected forward in time using ocean drift models. Based on this forecast about the environment, parameter values α, β, and r will be inserted or will be determined via a more sophisticated method. Then the OGR assigns vessels to the most valuable targets, considering not only the current position but also the predicted future states. As new data becomes available, parameters can be dynamically adjusted, and the route updated accordingly, maintaining both flexibility and performance.

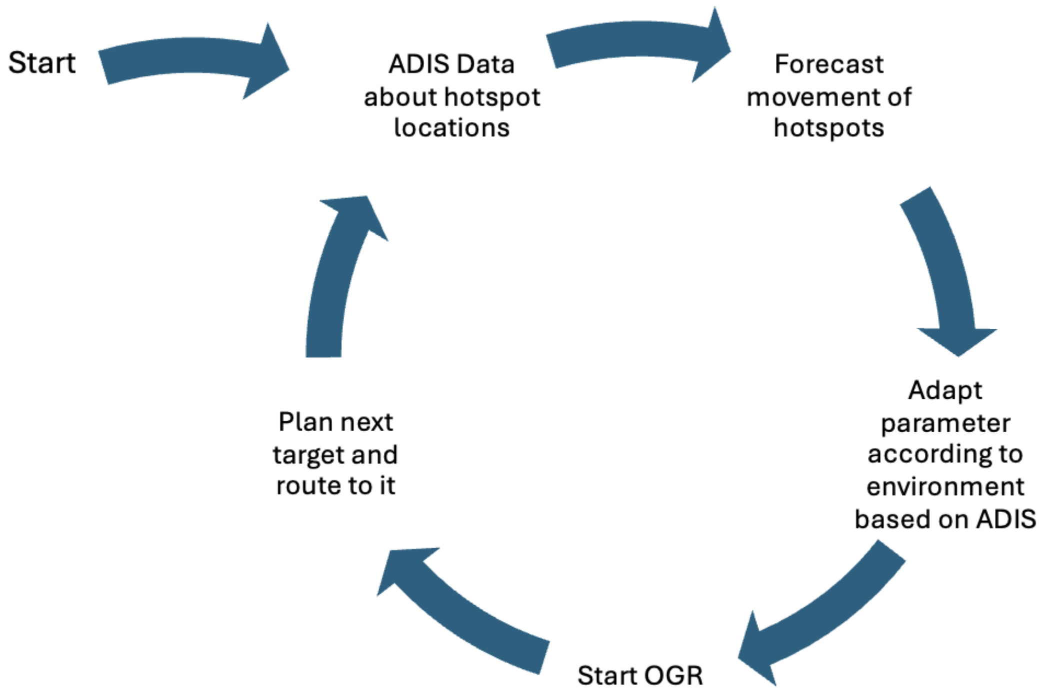

Figure 5.1: Integration cycle illustrating the interaction between ADIS and OGR.

This concept points to a promising step for real-world applications. By enabling adaptive planning based on real-time data, OGR could support more efficient, scalable cleanup, especially as operations expand to more vessels or regions. Such integration moves toward intelligent, data-driven marine resource management.

Alternative Use Case in Disaster Relief Operations: Mobile Search and Rescue Teams after Natural Disasters While the previous use case focused on maritime environments, the time-dependent Orienteering Problem with Moving Targets (TDOP-MT) is also highly applicable to emergency logistics. Following catastrophic events such as earthquakes or floods, mobile Search and Rescue teams must rapidly navigate to prioritized, ever-changing incident sites with limited operational capacity, such as restricted time or fuel. These applications can require online routing, making it an ideal non-maritime use case for the developed heuristics.

The targets would be the last locations of missing persons; these targets are dynamic in two ways. Firstly, their value, representing the survival probability of missed individuals, decays steeply over time. New intelligence from drone footage or distress calls continuously identifies new, potentially more urgent hotspots. Secondly, and critically, the targets themselves are often mobile: missing persons may be moving in search of safety, become displaced by floods, or shift their location within collapsed structures.

The mobile resources are search and rescue teams, constrained by tight operational time. The infrastructure is highly dynamic, with road blockages and debris creating uncertainty in travel times. This reflects the time-dependent aspect: travel costs shift quickly as the environment deteriorates or improves.

The goal is to maximize life-saving value or the highest priority score within the strict operational window. Here, the online decision-making and retargeting mechanisms developed for maritime use fit well. Teams must adjust routes as new information or environmental changes (like road blockages) arise, applying principles from the Online Greedy with Retargeting (OGR) heuristic.

This use case shows the developed algorithms work beyond maritime contexts, making them versatile and robust for complex, time-critical resource allocation under dynamic conditions.

6 Outlook

The results of this thesis demonstrate the potential of dynamic and reactive routing strategies in the context of marine plastic collection, particularly under the influence of drifting hotspots and changing ocean conditions. However, several assumptions had to be made to conduct this study.

Parameter Tuning One of the central findings of this study is the sensitivity of the system to the weighting parameter α, which governs the trade-off between proximity and value in target selection. In both the OG and OGR algorithms, this parameter was treated as static throughout the simulation after the optimal value was found via a grid search. While this design simplifies implementation, it also introduces significant limitations in both realism and flexibility.

First, identifying the optimal parameters via grid search requires access to the full trajectory dataset to simulate all parameter combinations. While the algorithm itself does not use future or past data during execution, the parameter optimization relies on complete information. Additionally, the vessel faces a changing environment and strict time constraints, so static parameters may lead to suboptimal performance as conditions evolve.

An attempt to implement a dynamically adaptive α that adjusts during runtime based on the environment was made during the work on this thesis. This was not successful because at this time, there was not enough knowledge about the effects and influence of the parameters. Although this approach was not successful, the result is informative: it highlights the complexity and non-linearity of the decision landscape. Developing an α schedule, or a learning-based adaptation mechanism, remains challenging yet promising. One approach could be, to start with a well performing α on average, which is then adjusted based on the surrounding environment.

In light of these insights, future research should thus move beyond static parameter tuning (e.g., grid search) toward meta-heuristic or reinforcement learning–based control strategies capable of adjusting parameters in real time.

Other Potential Extensions Beyond parameter adaptation, another limitation of the current framework is the lack of cooperation between vessels. At present, the agents operate independently and select targets as if they were alone in the system. Introducing explicit cooperation could significantly enhance efficiency. Sharing target information and making collaborative routes could improve global performance. Cooperative strategies would also allow for more sophisticated division of labour, such as regional partitioning or role specialization, which could lead to more efficient overall collection. Another component could be to add different types of vessels, making the problem heterogeneous.

The time dependency is a significant part of the problem; it is implicitly present due to the moving targets, as previously stated many times. The result is an inability to predict the arc weights, necessitating the use of the Euclidean distance as a reference value for them. Having a way to predict the arc weights, thus not using the distance for the score, could be beneficial. This could also be achieved with a stochastic approach for the trajectories.

Focusing on operational realism, one limitation is the lack of restrictions on vessels. There are no fuel, capacity, or distance limitations. However, this does not reflect real vessels, as these would need to refuel at some point and unload their collected targets. Also, the amount of debris collected might affect their speed and drift.

Another potential extension could be to to optimize not only which hotspot to target next, but also how to approach it, meaning how could the current field be used to save potential fuel. Or how to predict the future position of hotspots in order to reduce total travelled distance.

Overall, this thesis addresses a problem with direct relevance to real-world maritime logistics. While preliminary in scope, the proposed framework lays the groundwork for advancing dynamic routing strategies and highlights key directions for impactful future research. By building on these insights and addressing the outlined limitations, future work has the potential to deliver significant advancements for sustainable marine plastic collection and operational efficiency.

7 Summary

This thesis addressed the challenge of real-time plastic collection in dynamic marine environments by formulating and analysing a variant of the Orienteering Problem, the time-dependent Orienteering Problem with Moving Targets (TDOP-MT). Motivated by the need for scalable downstream cleanup solutions in the face of growing ocean plastic pollution, the study combined environmental simulation with algorithmic routing to evaluate how drifting plastic hotspots can be effectively collected by vessels.

The central contribution of this work is the development and evaluation of two heuristic algorithms: a basic online greedy approach and an enhanced version with retargeting capabilities. Both algorithms were embedded into a time stepped simulation framework based on OpenDrift, where plastic hotspots drift under the influence of real ocean current data. Each vessel continuously selects and navigates toward targets, aiming to maximize the total collected value within a fixed time horizon.

Simulation results across multiple synthetic scenarios demonstrated that even relatively simple algorithms can yield effective strategies, provided that parameter tuning is done carefully. The system was shown to be highly sensitive to α. This highlights the importance of not only selecting parameters thoughtfully but also of building adaptive systems in response to environmental changes.

The results demonstrated the potential of a well-tuned online heuristic. However, as stated multiple times, the calibration of the parameters was not performed online. The execution of the tests and the answering of the research questions laid the foundation for future work on tuning these parameters online, or alternatively, for using the average best parameter settings identified in this study.

Overall, this thesis demonstrated that integrating dynamic environmental data into real-time decision-making leads to qualitatively different behaviours and improved outcomes. It also highlighted key limitations and future directions. The methods developed here may serve as a foundation for more advanced and adaptive strategies for the TDOP-MT, strategies that are not only computationally efficient but also ecologically responsive.

List of Figures

List of Tables

Bibliography

Abdollahzadeh, B., Javadi, H., Toragay, O., Epicoco, N., & Khodadadi, N. (2024, 11). The green marine waste collector routing optimization with puma selection-based neighborhood search algorithm. *Cluster Computing*, *28*. doi: 10.1007/s10586-024-04812-w

Ali Abbaspour, R., & Samadzadegan, F. (2011, 09). Time-dependent personal tour planning and scheduling in metropolises. *Expert Syst. Appl.*, *38*, 12439-12452. doi: 10.1016/j.eswa.2011.04.025

Andrady, A. L. (2011). Microplastics in the marine environment. *Marine Pollution Bulletin*, *62*(8), 1596–1605. doi: 10.1016/j.marpolbul.2011.05.030

Automated debris imaging system (adis). (n.d.). `https://theoceancleanup.com/adis/`. (Accessed October 2025)

Bertsimas, D., Jaillet, P., & Martin, S. (2019, 01). Online vehicle routing: The edge of optimization in large-scale applications. *Operations Research*, *67*. doi: 10.1287/opre.2018.1763

Bhat, A., Gutow, G., Vundurthy, B., Ren, Z., Rathinam, S., & Choset, H. (2024, 09). *A complete algorithm for a moving target traveling salesman problem with obstacles.* doi: 10.48550/arXiv.2409.09852

Bhat, A., Gutow, G., Vundurthy, B., Ren, Z., Rathinam, S., & Choset, H. (2025). *Parallel, asymptotically optimal algorithms for moving target traveling salesman problems.*

Butt, S. E., & Ryan, D. M. (1999). An optimal solution procedure for the multiple tour maximum collection problem using column generation. *Computers & Operations Research*, *26*(4), 427–441. doi: 10.1016/S0305-0548(98)00071-9

Böckenhauer, H.-J., Hromkovič, J., Komm, D., Rossmanith, P., & Stocker, M. (2026). A survey of online knapsack problems. *Discrete Applied Mathematics*, *378*, 492–507. doi: 10.1016/j.dam.2025.08.011

Chao, I.-M., Golden, B. L., & Wasil, E. A. (1996a). A fast and effective heuristic for the orienteering problem. *European Journal of Operational Research*, *88*, 475-489. doi: 10.1016/0377-2217(95)00035-6

Chao, I.-M., Golden, B. L., & Wasil, E. A. (1996b). The team orienteering problem. *European Journal of Operational Research*, *88*(3), 464–474. doi: 10.1016/0377-2217(94)00289-4

Chen, Y., Awasthi, A. K., Wei, F., Tan, Q., & Li, J. (2021). Single-use plastics: Production, usage, disposal, and adverse impacts. *Science of The Total Environment*, *752*, 141772. doi: 10.1016/j.scitotenv.2020.141772

Choubey, N. (2013, 05). Moving target travelling salesman problem using genetic algorithm. *International Journal of Computer Application*, *70*, 975-8887. doi: 10.5120/11937-7726

Dagestad, K.-F., Röhrs, J., Breivik, , & Ådlandsvik, B. (2018). Opendrift v1.0: a generic framework for trajectory modelling. *Geoscientific Model Development*, *11*, 1405–1420. Retrieved from `https://doi.org/10.5194/gmd-11-1405-2018` doi: 10.5194/gmd-11-1405-2018

den Hertog, D., Pauphilet, J., Pham, Y., Sainte-Rose, B., & Song, B. (2024). Optimizing the path towards plastic-free oceans. *Oper. Res.*, *73*(3), 1165–1183. doi: 10.1287/opre.2023.0515

de Vries, R., Egger, M., Mani, T., & Lebreton, L. (2021). Quantifying floating plastic debris at sea using vessel-based optical data and artificial intelligence. *Remote Sensing*, *13*(17), 3401. doi: 10.3390/rs13173401

Duan, G., Fan, T., Chen, X., Chen, L., & Ma, J. (2021). A hybrid algorithm on the vessel routing optimization for marine debris collection. *Expert Systems with Applications*, *182*, 115198. doi: 10.1016/j.eswa.2021.115198

Egger, M., et al. (2024). Neuston distribution and implications for plastic cleanup in the gpgp. *Nature Communications*. (Forthcoming)

Fischetti, M., González, J. J. S., & Toth, P. (1998). Solving the orienteering problem through branch-and-cut. *INFORMS J. Comput.*, *10*, 133-148. doi: DOI:10.1287/ijoc.10.2.133

Fomin, F. V., & Lingas, A. (2002). Approximation algorithms for time-dependent orienteering. *Information Processing Letters*, *83*(2), 57–62. doi: 10.1016/S0020-0190(01)00313-1

Gao, P., Du, W., Yu, H., & Zhao, X. (2023, 10). A two-stage decision-support system for floating debris collection in reservoir areas. *Computers Industrial Engineering*, *185*. doi: 10.1016/j.cie.2023.109685

Garcia, A., Arbelaitz, O., Vansteenwegen, P., Souffriau, W., & Linaza, M. T. (2010). Hybrid approach for the public transportation time dependent orienteering problem with time windows. In (Vol. 6077). Berlin, Heidelberg: Springer. doi: 10.1007/978-3-642-13803-4_19

Garcia, A., Vansteenwegen, P., Arbelaitz, O., Souffriau, W., & Linaza, M. (2013, 03). Integrating public transportation in personalised electronic tourist guides. *Computers Operations Research*, *40*, -. doi: 10.1016/j.cor.2011.03.020

Gavalas, D., Konstantopoulos, C., Mastakas, K., Pantziou, G., & Vathis, N. (2014). Efficient heuristics for the time dependent team orienteering problem with time windows. In P. Gupta & C. Zaroliagis (Eds.), *Applied algorithms* (pp. 152–163). Springer International Publishing. doi: https://doi.org/10.1007/978-3-319-04126-1_13

Gavish, B., & Graves, S. (2004, 05). The traveling salesman problem and related problems.

Geyer, R., Jambeck, J. R., & Law, K. L. (2017). Production, use, and fate of all plastics ever made. *Science Advances*, *3*(7), e1700782. doi: 10.1126/sciadv.1700782

Golden, B. L., Levy, L., & Vohra, R. (1987). The orienteering problem. *Naval Research Logistics*, *34*(3), 307–318. doi: 10.1002/1520-6750(198706)34:3<307::AID-NAV3220340302>3.0.CO;2-D

Gregory, M. R. (2009). Environmental implications of plastic debris in marine settings–entanglement, ingestion, smothering, hangers-on, hitch-hiking and alien invasions. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *364*(1526), 2013–2025. doi: 10.1098/rstb.2008.0265

Gunawan, A., Lau, H., & Vansteenwegen, P. (2016, 05). Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, *225*. doi: 10.1016/j.ejor.2016.04.059

Gurobi Optimization, LLC. (2024). *Gurobi Optimizer Reference Manual.* Retrieved from `https://www.gurobi.com`

Hammar, M., & Nilsson, B. J. (1999). Approximation results for kinetic variants of tsp. In *Proceedings of the 26th international colloquium on automata, languages and programming* (p. 392–401).

Helvig, C. S., Robins, G., & Zelikovsky, A. (2003). The moving-target traveling salesman problem. *Journal of Algorithms*, *49*(1), 153–174. doi: 10.1016/S0196-6774(03)00075-0

Hu, C. (2021). Remote detection of marine debris using satellite observations in the visible and near infrared spectral range: Challenges and potentials. *Remote Sensing of Environment*, *259*, 112414. doi: 10.1016/j.rse.2021.112414

Jaillet, P., & Wagner, M. R. (2008). Generalized online routing: New competitive ratios, resource augmentation, and asymptotic analyses. *Operations Research*, *56*(3), 745–757. doi: 10.1287/opre.1070.0450

Kantor, M. G., & Rosenwein, M. B. (1992). The orienteering problem with time windows. *Journal of the Operational Research Society*, *43*, 629-635. doi: 10.1057/JORS.1992.88

Laporte, G., & Martello, S. (1990). The selective travelling salesman problem. *Discrete Applied Mathematics*, *26*(2–3), 193–207. doi: 10.1016/0166-218X(90)90100-Q

Lau, W. W. Y., Shiran, Y., Bailey, R. M., Cook, E., Stuchtey, M. R., Koskella, J., … Palardy, J. E. (2020). Evaluating scenarios toward zero plastic pollution. *Science*, *369*(6510), 1455–1461. doi: 10.1126/science.aba9475

Lebreton, L., et al. (2018). Evidence that the great pacific garbage patch is rapidly accumulating plastic. *Scientific Reports*, *8*, 4666. doi: 10.1038/s41598-018-22939-w

Lv, S., Cui, K., Zhao, S., Li, Y., Liu, R., Hu, R., … Shao, Z. (2024). Continuous generation and release of microplastics and nanoplastics from polystyrene by plastic-degrading marine bacteria. *Journal of Hazardous Materials*, *465*, 133339. doi: 10.1016/j.jhazmat.2023.133339

MacLeod, M., Arp, H. P. H., Tekman, M. B., & Jahnke, A. (2021). The global threat from plastic pollution. *Science*, *373*(6550), 61-65. doi: 10.1126/science.abg5433

Mahmoudzadeh, S., Powers, D., Sammut, K., Yazdani, A., & Atyabi, A. (2018, 01). A hierarchal planning framework for auv mission management in a spatio-temporal varying ocean. *Computers Electrical Engineering*, *67*. doi: 10.1016/j.compeleceng.2017.12.035

Mansfield, A., Macharet, D., & Hsieh, M. (2023, 10). Energy-efficient team orienteering problem in the presence of time-varying ocean currents. In (p. 838-843). doi: 10.1109/IROS55552.2023.10341645

Maximenko, N., Hafner, J., & Niiler, P. (2012). Pathways of marine debris derived from trajectories of lagrangian drifters. *Marine Pollution Bulletin*, *65*, 51–62. doi: 10.1016/j.marpolbul.2011.04.016

Meijer, L. J., van Emmerik, T., van der Ent, R., Schmidt, C., & Lebreton, L. (2021). More than 1000 rivers account for 80% of global riverine plastic emissions into the ocean. *Science Advances*, *7*(18), eaaz5803. doi: 10.1126/sciadv.aaz5803

Michail, O. (2015). *An introduction to temporal graphs: An algorithmic perspective.*

Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM*, *7*(4), 326–329. doi: 10.1145/321043.321046

Miller, M. E., Hamann, M., & Kroon, F. J. (2020). Bioaccumulation and biomagnification of microplastics in marine organisms: A review and meta-analysis of current data. *PLOS ONE*, *15*(10), e0240792. doi: 10.1371/journal.pone.0240792

Philip, A. G., Ren, Z., Rathinam, S., & Choset, H. (2024, October). A mixed-integer conic program for the moving-target traveling salesman problem based on a graph of convex sets. In *2024 ieee/rsj international conference on intelligent robots and systems (iros)* (p. 8847–8853). IEEE. doi: 10.1109/iros58592.2024.10802374

Philip, A. G., Ren, Z., Rathinam, S., & Choset, H. (2025). *A mixed-integer conic program for the multi-agent moving-target traveling salesman problem.*

Ritzinger, U., Puchinger, J., & Hartl, R. (2016, 01). A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*, *54*, 215-231. doi: 10.1080/00207543.2015.1043403

Schmid, V., & Ehmke, J. F. (2017). An effective large neighborhood search for the team orienteering problem with time windows. In *Lecture notes in computer science* (Vol. 10572, pp. 1–16). Springer. doi: 10.1007/978-3-319-68496-3_1

Schnurr, R. E. J., Alboiu, V., Chaudhary, M., Corbett, R. A., Quanz, M. E., Sankar, K., … Walker, T. R. (2018). Reducing marine pollution from single-use plastics (sups): A review. *Marine Pollution Bulletin*, *137*, 157–171. doi: 10.1016/j.marpolbul.2018.10.001

Seeley, M., Nuss, S., & Smith, B. (2024). Trawling through the five gyres: A microplastic research study.

Service, C. M. (2024). *Global_analysisforecast_phy_001_024 - global ocean physics analysis and forecast.* `https://data.marine.copernicus.eu/product/GLOBAL_ANALYSISFORECAST_PHY_001_024`. (Accessed: July 2024)

Shiri, D., Akbari, V., & Hassanzadeh, A. (2024). The capacitated team orienteering problem: An online optimization framework with predictions of unknown accuracy. *Transportation Research Part B: Methodological*, *185*, 102984. doi: 10.1016/j.trb.2024.102984

Stieber, A., & Fügenschuh, A. (2022, 09). Dealing with time in the multiple traveling salespersons problem with moving targets. *Central European Journal of Operations Research*, *30*. doi: 10.1007/s10100-020-00712-7

The Ocean Cleanup. (2025). *The ocean cleanup project.* `https://theoceancleanup.com`. (Accessed October 2025)

Tsiligiridis, T. (1984, 09). Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, *35*, 797-809. doi: 10.1057/jors.1984.162

Tuuri, E. M., & Leterme, S. C. (2023). How plastic debris and associated chemicals impact the marine food web: A review. *Environmental Pollution*, *321*, 121156. doi: 10.1016/j.envpol.2023.121156

United Nations Environment Programme (UNEP). (2021). *From pollution to solution: A global assessment of marine litter and plastic pollution.* Retrieved 2025-10-16, from `https://www.unep.org/resources/pollution-solution-global-assessment-marine-litter-and-plastic-pollution`

Vansteenwegen, P., Souffriau, W., & Van Oudheusden, D. (2011). The orienteering problem: A survey. *European Journal of Operational Research*, *209*(1), 1–10. doi: 10.1016/j.ejor.2010.03.045

Verbeeck, C., Sörensen, K., Aghezzaf, E.-H., & Vansteenwegen, P. (2014, 07). A fast solution method for the time-dependent orienteering problem. *European Journal of Operational Research*, *236*, 419–432. doi: 10.1016/j.ejor.2013.11.038

8 Appendix

8.1 Distances in default Scenario:

This table contains the total travelled distance over T from all vessels in the default scenario with different α configurations over multiple seeds. The table was used to construct the box plot (see: 4.1)

Table 8.1: Total travelled distances of vessels in the default Scenario with different across 10 random seeds (all values in km).

	Collected Value		
Seed	$\alpha = 0$ **(Pure Value)**	$\alpha = 1$ **(Pure Distance)**	$\alpha = \alpha_{opt}$ **(Optimal)**
100	34.64	29.22	33.54
200	38.14	31.05	36.61
300	35.12	32.28	32.97
400	37.61	30.90	35.97
500	38.42	35.52	37.98
600	35.91	33.31	36.19
700	36.87	33.02	34.23
800	32.45	35.15	33.15
900	33.44	33.78	32.47
1000	32.98	32.35	35.36
Mean	**35.56**	**32.66**	**34.85**

8.2 Test 0 detailed Results:

The table 8.2 includes detailed results about the objective values obtained with different α values in the default scenario. The results were used in 4.3.

Seed	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
0	22486	22486	22486	22486	22486	22486	22486	22486	22486	26001	28648	28648	31626	31546	31719	33751	27317	27318	31026	21719	21719
50	21727	21727	21727	21727	21727	21727	21727	21727	21727	21727	21727	21727	23088	23773	25018	26969	24722	15605	15605	15605	15605
100	28559	28559	43315	43315	43315	43315	43315	45151	35413	40874	40874	40874	44217	52760	53919	53926	54114	54114	50039	45716	38150
150	25529	25529	25529	25529	25529	25529	25529	25529	25529	27892	27892	27892	31544	36200	36200	37967	32828	25939	22057	34748	33003
200	20088	22276	22276	22276	22276	22276	22063	22063	22063	22063	23580	23580	23580	23761	25359	23109	10493	10493	9154	9154	9154
250	29443	29443	29443	29443	29443	29443	29443	29443	29443	29443	29443	29443	30074	32794	34937	26882	26882	18823	18823	14284	14284
300	25695	25695	29429	29429	29429	29429	29429	29429	29429	29429	29429	28472	28472	26229	24970	27392	29604	31154	31154	20572	11843
350	24520	24520	24520	24520	24520	24520	24520	24520	24520	28640	30279	30279	26718	18654	18084	23381	23381	14897	14897	16109	15297
400	17033	17033	17033	20818	20818	20818	25298	25298	25298	25298	20130	20130	20130	21942	26469	6437	6437	6437	6437	6457	6457
450	18110	18110	19433	19433	19433	19433	19433	19433	19433	19433	19433	19433	21439	22353	19595	19595	18116	18116	18672	18672	14381
500	34829	34829	34829	34829	34829	34829	34829	34829	34829	34829	34829	34829	43293	43293	42387	40830	26702	34256	29054	30075	22830
550	27927	27927	27927	27927	27927	27927	27927	27927	27927	27927	27927	26471	32133	32760	34040	19242	15219	15479	15479	16825	8913
600	23653	23653	23653	23653	23653	23653	23653	27187	27187	27187	27187	28800	21684	22339	22694	22633	18822	19006	19131	19014	19014
650	18409	18409	18409	22102	22102	22102	22102	22102	22102	22102	22102	23482	23482	24080	24312	24949	11541	11562	11562	11562	11562
700	17404	17404	17404	17404	17404	19478	19478	19478	19478	30416	30416	31765	31765	33944	33259	21700	20020	20206	12303	12303	11060
750	29023	29023	29023	29023	29023	29023	32567	32567	32567	32567	34404	34404	34404	34956	33187	33187	33187	30009	26166	30212	25137
800	30661	30661	30661	30661	30661	34557	34557	38137	41795	44193	44193	44193	44193	45845	38409	39217	39241	44128	39631	39631	21444
850	20499	20499	26954	26954	26954	26954	26954	26954	22986	22986	22986	22986	23548	24811	29610	31178	26890	12629	12674	12674	12674
900	27202	27202	27202	27202	22564	22564	22564	22564	22564	25077	28517	28517	29287	32223	29762	34989	25728	20641	21132	24362	21823
950	18927	18927	18927	20909	20909	20909	20909	22274	22274	22274	22274	22274	22274	23138	23406	23406	8926	15596	4087	4087	4087
1000	28312	28312	19053	19053	19053	19053	19053	19053	24152	34967	34967	40865	45029	45029	42411	34948	36166	35665	27734	27442	13504
1050	14957	14957	14957	14957	15936	15936	15936	15936	17183	18770	18770	18770	18770	18770	18770	18945	15646	19108	13915	5625	5625
1100	17443	17443	17443	17443	17443	17443	17443	16112	16706	16706	17373	18209	19567	19885	19154	19154	19625	19695	11317	11317	18090
1150	26340	26340	26340	26340	26340	26340	25636	25636	25636	25636	25636	30122	29319	32282	32885	30248	30095	30845	18780	22359	19750
1200	19966	19966	27937	27937	27937	27937	35451	35451	35451	45172	45172	53758	53758	62103	62103	59938	44067	45656	51806	20850	21004
1250	19490	19490	19490	19490	19490	19490	19490	19490	19490	19490	19490	26999	27388	30461	16924	19703	16630	22502	22503	22593	22602
1300	17765	17765	17765	17765	24497	24497	30711	30711	30711	30711	30711	35401	30393	30582	22131	23511	32593	31894	33733	33733	22795
1350	17923	17923	17923	17923	17923	17923	17923	17923	17923	17923	17923	17923	19865	20390	17796	20638	18406	19031	19031	18922	14179
1400	19845	19845	19845	19845	19845	19845	19845	19845	19845	23085	23085	23085	22741	23601	24065	24065	12123	14296	13380	13380	13380
1450	26900	26900	26900	26900	26900	26900	26900	26900	26900	26900	26900	29469	29608	29527	29225	29028	11320	11395	10936	10936	10936
1500	24074	24074	27200	27200	24074	24074	24074	27200	27200	29595	29595	28183	34268	31257	30146	17005	17451	17451	11427	11427	11427
1550	24206	24206	24206	24206	24206	24206	24206	24206	24206	24206	24206	24206	24206	28384	29523	29598	25998	29184	7329	7329	7329
1600	30301	30301	30301	30301	30301	30301	30301	30301	30301	30301	30301	30301	30301	32285	39643	34746	36448	17964	18754	16157	16157
1650	32514	32514	32514	32514	32514	32514	32514	32514	32514	32514	32514	32796	32796	34954	35336	30961	30961	38445	17181	17181	15887
1700	38369	38369	38369	38369	38369	38369	38369	38369	38369	38369	38369	38369	38369	38394	27267	38246	40336	27170	27170	27170	27170
1750	22092	22092	22092	22092	22092	22092	24845	24845	24006	24006	24006	25669	26140	27286	28692	28692	28692	19907	15921	15921	8275
1800	16140	16140	16140	17111	17111	18028	18028	18028	18028	18028	18028	18513	18513	19636	19636	19009	3579	3471	3471	3471	3471
1850	16686	16686	16686	16686	16686	16686	16686	16686	16686	18944	18944	18944	13576	14114	14216	14027	14198	14198	15032	15816	5136
1900	19566	19566	19566	19566	19566	19566	19566	19566	20287	20287	20287	20287	20287	20287	20287	20733	4545	18576	18576	18576	10016
1950	19888	19888	28480	28480	28480	28480	28480	28480	28480	28480	32737	35786	35786	38791	36072	27690	30786	18267	17498	9980	9980
2000	37350	37350	37350	37350	37350	40423	40423	33529	25219	27913	27913	27913	30742	42487	42436	41252	37313	37313	29742	29742	29920
2050	22607	22607	22607	22607	22607	22607	22023	32248	32248	39724	47186	47186	44424	44424	50288	50825	35126	40680	39154	33064	34229
2100	22208	22208	22208	22208	22208	22208	22208	22208	22208	22208	22208	22208	15495	17277	17277	12127	12183	12183	12183	12717	12717
2150	13917	13917	13917	13917	13917	13917	13917	13917	13917	13917	13917	16742	18370	18370	17523	14943	13848	15201	15201	14892	14892
2200	20082	20082	20082	20082	20082	20082	20082	20082	20082	21256	21256	21256	22269	25892	26475	25040	24865	24927	24957	24843	24843

Table 8.2: Results for different Seeds (rows) for the default scenario over all possible α values (columns)

8.3 Test 1 detailed Results:

These are the detailed results of test 1. The tables include all 20 runs over each scenario. These are the results that were used to compute table 4.2 and illustrations 4.4,4.5 and 4.6. The mean α_{opt} values per scenario are applied for the trajectories: 4.7,4.8,4.9,4.10,4.11,4.12.

Table 8.3: Results for Scenarios 1 and 10 for edge cases and optimal α.

Scenario 1

Seed	$V_{\alpha=0}$	$V_{\alpha=1}$	V_{opt}	α_{opt}	Δ_{opt} [%]
1000	28312	13504	45029	0.60	+59.1
2000	37350	29920	42487	0.65	+13.7
3000	19767	19541	39929	0.60	+101.9
4000	32612	30177	57228	0.85	+75.5
5000	23854	22520	30144	0.80	+26.4
6000	16124	8256	17208	0.75	+6.7
7000	35650	34790	41736	0.90	+17.1
8000	28584	35591	35591	0.95	+0.0
9000	19964	21356	39004	0.65	+82.6
10000	27047	18027	27047	0.00	+0.0
11000	17076	17439	26622	0.65	+52.6
12000	27381	19334	33412	0.70	+22.0
13000	23877	33316	44536	0.95	+33.7
14000	22194	13513	32542	0.70	+46.6
15000	21021	16962	21021	0.00	+0.0
16000	19971	30752	40467	0.95	+31.6
17000	22139	13158	22139	0.00	+0.0
18000	12072	13383	15932	0.90	+19.0
19000	17211	14095	24830	0.75	+44.3
20000	30908	28254	40914	0.55	+32.4
Mean	24,353.62	21,304.38	34,421.29	0.64	**+34.5**

Scenario 10

Seed	$V_{\alpha=0}$	$V_{\alpha=1}$	V_{opt}	α_{opt}	Δ_{opt} [%]
1000	0	3440	12324	0.65	+258.26
2000	0	2170	6716	0.75	+209.50
3000	10000	2735	10000	0.00	+0.00
4000	0	5574	9382	0.70	+68.32
5000	10000	4529	10000	0.00	+0.00
6000	0	1365	7006	0.90	+413.26
7000	10000	6263	15129	0.90	+51.29
8000	0	14829	20783	0.85	+40.14
9000	10000	8903	16077	0.90	+60.77
10000	0	4918	8713	0.75	+77.18
11000	10000	1922	13115	0.80	+31.15
12000	10000	2112	10572	0.75	+5.72
13000	10000	5123	14250	0.75	+42.50
14000	0	8144	17892	0.75	+119.70
15000	10000	3091	10000	0.00	+0.00
16000	10000	5045	12157	0.75	+21.57
17000	0	8050	12140	0.85	+50.81
18000	10000	9370	15238	0.65	+52.38
19000	0	8406	15332	0.70	+82.39
20000	0	2215	10084	0.85	+355.26
Mean	4,761.90	5,316.38	12,344.48	0.66	+104.7

Table 8.4: Results for Scenarios 2 and 3 for edge cases and optimal α.

Scenario 2

Seed	$V_{\alpha=0}$	$V_{\alpha=1}$	V_{opt}	α_{opt}	Δ_{opt} [%]
1000	0	9643	20006	0.60	+107.48
2000	25955	9322	27274	0.75	+5.08
3000	19767	10099	25484	0.60	+28.92
4000	18857	11642	33506	0.75	+77.79
5000	15081	17707	17707	0.95	+0.00
6000	12376	1192	12981	0.75	+4.89
7000	8379	8786	17438	0.85	+98.47
8000	24677	13137	27667	0.90	+12.11
9000	10000	13594	13959	0.90	+2.68
10000	5965	8911	11885	0.65	+33.37
11000	17076	11174	17824	0.90	+4.38
12000	8218	11107	23460	0.75	+111.23
13000	10000	16873	26776	0.80	+58.69
14000	22194	7933	22194	0.00	+0.00
15000	10000	5211	14210	0.90	+42.10
16000	9971	14976	17625	0.70	+17.69
17000	10000	12891	13094	0.75	+1.57
18000	10000	11979	13265	0.80	+10.74
19000	10000	12988	13030	0.75	+0.32
20000	24165	22118	24165	0.00	+0.00
Mean	12,984.81	11,472.67	19,693.14	0.70	+34.5

Scenario 3

Seed	$V_{\alpha=0}$	$V_{\alpha=1}$	V_{opt}	α_{opt}	Δ_{opt} [%]
1000	95658	110157	110546	0.95	+0.35
2000	62908	74459	74459	0.80	+0.00
3000	61914	71333	74457	0.90	+4.38
4000	77914	82605	82605	1.00	+0.00
5000	45232	50638	52697	0.80	+4.07
6000	27132	31831	32401	0.90	+1.80
7000	51204	57647	58970	0.90	+2.30
8000	50114	59121	62554	0.95	+5.81
9000	58347	69555	72092	0.90	+3.65
10000	59921	65626	65626	0.90	+0.00
11000	40084	43200	44961	0.95	+4.08
12000	52909	64999	64999	0.95	+0.00
13000	56626	82090	82090	0.95	+0.00
14000	46735	56094	57568	0.70	+2.63
15000	33069	36465	37365	0.95	+2.47
16000	84066	101203	104975	0.85	+3.73
17000	35658	42336	45869	0.85	+8.35
18000	22366	25395	25401	0.90	+0.02
19000	34013	40166	40166	1.00	+0.00
20000	61834	81881	81881	0.95	+0.00
Mean	54,922.00	64,617.05	65,820.38	0.90	+2.1

Table 8.5: Results for Scenarios 4 and 5 for edge cases and optimal α.

Scenario 4

Seed	$V_{\alpha=0}$	$V_{\alpha=1}$	V_{opt}	α_{opt}	Δ_{opt} [%]
1000	19648	31311	36016	0.90	+15.03
2000	32906	24413	37569	0.65	+14.17
3000	19766	32863	42188	0.75	+28.37
4000	29084	25309	35760	0.70	+22.95
5000	26089	5084	31143	0.80	+19.37
6000	20243	18728	21278	0.65	+5.11
7000	13486	2221	13486	0.00	+0.00
8000	23257	15029	26390	0.60	+13.47
9000	15163	20388	20551	0.95	+0.80
10000	18129	940	23051	0.20	+27.15
11000	14956	8838	17981	0.55	+20.22
12000	32990	32717	41408	0.85	+25.52
13000	19195	14586	29684	0.65	+54.65
14000	16101	14332	19252	0.65	+19.57
15000	14376	3140	16840	0.75	+17.14
16000	26647	7656	35808	0.70	+34.38
17000	16448	14658	17400	0.75	+5.89
18000	30339	31146	33887	0.45	+8.79
19000	17211	15902	22336	0.80	+29.78
20000	36211	43387	48163	0.60	+11.01
Mean	21,994.90	18,759.95	28,867.00	0.66	+18.5

Scenario 5

Seed	$V_{\alpha=0}$	$V_{\alpha=1}$	V_{opt}	α_{opt}	Δ_{opt} [%]
1000	18298	29872	59161	0.80	+98.05
2000	23661	23171	47357	0.80	+100.15
3000	18844	22885	47660	0.70	+108.26
4000	32454	37084	63983	0.65	+72.56
5000	30064	27064	42094	0.75	+39.99
6000	23159	21649	37749	0.75	+62.99
7000	34361	46423	64839	0.80	+39.67
8000	28117	47520	77663	0.70	+63.43
9000	24477	27353	57931	0.75	+111.79
10000	20134	15996	36740	0.75	+82.48
11000	33548	32346	68663	0.80	+104.66
12000	27813	35460	58852	0.70	+66.00
13000	33443	19057	61290	0.80	+83.27
14000	34951	44243	71109	0.65	+60.72
15000	35422	52627	83834	0.80	+59.30
16000	27239	22215	45149	0.75	+65.75
17000	23591	32401	47493	0.70	+46.58
18000	25764	33797	67318	0.85	+99.19
19000	29829	26514	46824	0.80	+57.05
20000	16086	21984	52259	0.75	+137.71
Mean	26,645.38	30,930.14	57,006.14	0.75	+78.9

Table 8.6: Results for Scenarios 6 and 7 for edge cases and optimal α.

Scenario 6

Seed	$V_{\alpha=0}$	$V_{\alpha=1}$	V_{opt}	α_{opt}	Δ_{opt} [%]
1000	92189	106893	109838	0.85	+2.75
2000	62987	74459	74459	1.00	+0.00
3000	60873	71079	72352	0.85	+1.79
4000	76854	74882	81995	0.95	+6.69
5000	44820	43093	53381	0.90	+19.09
6000	26533	31856	32100	0.95	+0.77
7000	51547	47405	57788	0.90	+12.11
8000	51139	49768	61946	0.80	+21.13
9000	62721	62158	71135	0.90	+13.41
10000	57819	65608	65608	1.00	+0.00
11000	40084	43042	43709	0.65	+1.55
12000	51603	51022	62280	0.65	+20.70
13000	59019	81786	81786	1.00	+0.00
14000	46055	53992	55643	0.70	+3.06
15000	33316	36365	37145	0.90	+2.15
16000	73098	99869	99869	1.00	+0.00
17000	34223	34618	44220	0.55	+27.76
18000	23069	25187	25187	1.00	+0.00
19000	34436	38174	38665	0.65	+1.29
20000	61862	75487	79000	0.65	+4.66
Mean	54,116.00	60,649.33	64,664.00	0.84	+6.7

Scenario 7

Seed	$V_{\alpha=0}$	$V_{\alpha=1}$	V_{opt}	α_{opt}	Δ_{opt} [%]
1000	0	9643	20006	0.60	+107.48
2000	17645	9727	27155	0.70	+53.90
3000	19767	18668	25484	0.60	+28.92
4000	10000	11642	33506	0.75	+187.80
5000	15081	17707	17707	0.95	+0.00
6000	12376	1430	12981	0.75	+4.89
7000	8379	8016	17438	0.85	+108.13
8000	24677	13137	29438	0.70	+19.29
9000	10000	14344	14344	1.00	+0.00
10000	5965	8852	11830	0.65	+33.64
11000	17076	11473	17824	0.90	+4.38
12000	8218	13424	20206	0.70	+50.52
13000	10000	16873	24630	0.85	+46.00
14000	22194	7328	22194	0.00	+0.00
15000	10000	5211	14210	0.90	+42.10
16000	9971	15556	17625	0.70	+13.30
17000	10000	12891	13101	0.75	+1.63
18000	10000	12245	13265	0.80	+8.33
19000	10000	12988	13022	0.75	+0.26
20000	17244	22118	22977	0.90	+3.88
Mean	11,837.76	12,043.62	19,473.76	0.73	+39.1

Table 8.7: Results for Scenarios 8 and 9 for edge cases and optimal α.

Scenario 8

Seed	$V_{\alpha=0}$	$V_{\alpha=1}$	V_{opt}	α_{opt}	Δ_{opt} [%]
1000	10000	5205	24523	0.70	+145.23
2000	18310	14145	25219	0.25	+37.74
3000	19767	10918	20525	0.65	+3.83
4000	26751	9156	31407	0.90	+17.39
5000	15081	7883	20944	0.80	+38.88
6000	12376	2454	12376	0.00	+0.00
7000	18379	22559	27445	0.75	+21.65
8000	19899	20399	26202	0.80	+28.45
9000	19964	15850	22865	0.70	+14.53
10000	10000	5025	15965	0.60	+59.65
11000	17076	10361	17676	0.75	+3.52
12000	10000	3106	17906	0.70	+79.06
13000	23877	21974	33218	0.90	+39.12
14000	10000	12089	15735	0.75	+30.16
15000	10000	13964	16857	0.85	+20.72
16000	10000	10366	19971	0.05	+92.65
17000	16076	13104	16076	0.00	+0.00
18000	10000	12580	13078	0.65	+3.96
19000	17211	1405	19658	0.70	+14.22
20000	24165	15196	25768	0.70	+6.64
Mean	15,663.43	11,092.57	21,330.33	0.61	+38.2

Scenario 9

Seed	$V_{\alpha=0}$	$V_{\alpha=1}$	V_{opt}	α_{opt}	Δ_{opt} [%]
1000	48797	38105	76825	0.70	+57.44
2000	47784	57649	66564	0.75	+15.46
3000	51509	55525	61957	0.80	+11.58
4000	63966	61281	78883	0.90	+23.32
5000	33669	32178	44090	0.70	+30.95
6000	23033	10614	27993	0.65	+21.53
7000	41289	33799	52465	0.90	+27.06
8000	37562	49173	54926	0.80	+11.70
9000	43794	34824	58875	0.75	+34.44
10000	44613	27389	53978	0.65	+20.99
11000	32192	32895	38547	0.80	+17.18
12000	39987	44263	58052	0.80	+31.16
13000	38613	58380	64996	0.85	+11.33
14000	40597	40330	46800	0.65	+15.29
15000	28788	24569	32994	0.65	+14.61
16000	52191	54720	66874	0.85	+22.20
17000	28808	23369	34869	0.75	+21.04
18000	19621	20718	22515	0.70	+8.67
19000	29432	20224	32537	0.75	+10.55
20000	56573	68251	68251	1.00	+0.00
Mean	40,553.10	39,350.52	53,324.57	0.77	+22.1

8.4 Test 2 detailed Results:

These are the detailed results of test 2. The tables include the performance of the tested algorithms over three seeds in all scenarios. Due to the excessive runtime only three seeds were used. These are the results that were used to compute table 4.3 and illustrations 4.13,4.14, 4.15.

Table 8.8: Results of simulations for Seed 42 for OG, OGR and Baseline solutions with calibrated parameters

Szenario	OG			OGR						Static	
	α_{opt}	**Value**	Δ_{B2} [%]	α_{opt}	**Value**	β_{opt}	r_{opt}	Δ_{B2} [%]	Δ_{OG} [%]	**B1**	**B2**
1	0.65	27 637	+75.2	0.35	29 245	0.90	1.00	+85.4	+5.8	30 222	15 772
2	0.65	18 227	+1.4	0.00	19 250	0.85	0.80	+7.1	+5.6	19 584	17 973
3	0.75	39 533	+24.1	0.20	39 659	0.95	0.90	+24.6	+0.3	40 548	31 846
4	0.75	18 117	+122.1	0.20	18 506	0.70	0.90	+126.9	+2.1	20 908	8 155
5	0.75	66 960	+7.4	0.90	70 335	0.85	0.90	+12.8	+5.0	78 604	62 327
6	0.75	40 444	+24.2	0.65	40 548	0.95	0.90	+24.5	+0.3	40 548	32 564
7	0.65	18 227	-5.4	0.60	19 385	0.99	1.10	+5.7	+6.4	19 584	19 282
8	0.85	17 885	+106.1	1.00	17 885	0.75	1.20	+106.1	0.0	18 677	8 677
9	0.70	35 702	-4.4	0.45	38 993	0.90	1.00	+4.4	+9.2	40 218	37 367
10	0.70	10 887	+7.5	0.65	12 292	0.75	0.10	+21.3	+12.9	12 589	10 130

(Column group heading: **Parameter and results**)

Table 8.9: Results of simulations for Seed 1111 for OG, OGR and Baseline solutions with calibrated parameters

Szenario	OG			OGR						Static	
	α_{opt}	**Value**	Δ_{B2} [%]	α_{opt}	**Value**	β_{opt}	r_{opt}	Δ_{B2} [%]	Δ_{OG} [%]	**B1**	**B2**
1	0.95	41 337	+8.7	0.95	41 337	0.50	1.20	+8.7	0.0	48 806	38 125
2	0.90	26 407	-2.3	0.00	26 407	0.90	0.50	-2.3	0.0	31 960	27 024
3	0.75	64 554	+71.9	0.75	67 077	0.95	1.00	+78.7	+3.9	71 156	37 538
4	0.00	27 494	+2.6	0.00	28 195	0.95	0.90	+5.2	+2.5	31 046	26 793
5	0.70	55 935	+40.8	0.50	56 582	0.85	0.10	+42.3	+1.2	62 724	39 756
6	0.65	68 835	+39.0	0.60	70 531	0.95	1.00	+42.3	+2.5	71 378	49 543
7	0.85	26 407	-17.3	0.10	26 407	0.90	0.50	-17.3	0.0	31 960	31 960
8	0.60	28 566	+1.3	0.60	30 577	0.95	1.10	+8.8	+7.0	31 736	28 099
9	0.60	57 368	+16.8	0.40	59 167	0.95	1.00	+20.3	+3.1	68 936	49 136
10	0.80	13 954	+73.3	1.00	14 939	0.80	0.90	+85.5	+7.1	18 054	8 054

(Column group heading: **Parameter and results**)

Table 8.10: Results of simulations for Seed 5040 for OG, OGR and Baseline solutions with calibrated parameters

Szenario	Parameter and results										
	OG			OGR						Static	
	α_{opt}	Value	Δ_{B2} [%]	α_{opt}	Value	β_{opt}	r_{opt}	Δ_{B2} [%]	Δ_{OG} [%]	B1	B2
1	0.75	26 962	+1.3	0.00	27 881	0.90	0.80	+4.7	+3.4	31 054	26 625
2	0.65	18 769	-2.6	0.20	20 692	0.90	0.90	+2.2	+10.3	23 145	19 268
3	0.90	38 370	+10.9	0.45	39 750	0.95	0.90	+14.4	+3.6	40 452	34 771
4	0.40	18 824	-13.6	0.30	20 244	0.70	0.90	-7.2	+7.6	21 987	21 862
5	0.60	41 200	+46.5	0.65	45 097	0.75	1.00	+60.3	+9.5	51 517	28 130
6	0.85	40 842	+63.0	0.45	41 160	0.90	1.00	+64.3	+0.8	41 237	25 046
7	0.70	18 769	-2.6	0.00	19 559	0.90	0.80	+0.8	+4.2	23 145	19 397
8	0.70	20 618	+0.5	0.05	21 117	0.90	0.90	+2.0	+2.4	23 561	20 704
9	0.75	26 962	+1.3	0.00	27 881	0.90	0.80	+4.7	+3.4	31 054	26 625
10	0.00	10 000	-14.8	0.00	10 000	0.50	0.50	-14.8	0.0	14 445	11 738

8.5 Test 3 detailed Results:

These are the detailed results of the runtime tests. The table includes one representative run for each scenario.

Table 8.11: Comparison of computational run times across all approaches and scenarios.

Scenario	Static (B1/B2)	OG		OGR	
	Find & Execute (s)	Find α_{opt} (s)	Execute OG (s)	Find Params (s)	Execute OGR (s)
1	59	15	<1	1640	<1
2	20	10	<1	1400	<1
3	600	40	4	8000	4
4	35	10	<1	1900	1
5	600	20	<1	2000	<1
6	460	15	<1	1600	<1
7	600	17	<1	1900	<1
8	200	9	<1	3300	1
9	600	30	<1	7500	1
10	600	30	2	4200	2
Mean	**377.4**	**19.6**	**<1.2**	**3344.0**	**<1.3**

8.6 Test 4 detailed Results:

These are the detailed results of test 4. The table includes the performance of the OGR over 20 seeds in the default scenario. The results of the OGR are directly in the testings section as these were only three runs.

Table 8.12: Mean optimal α form default scenario, applied for multiple seeds and percentage in/decrease

Seed	V_{opt}	Δ_{mean} [%]	$V_{\alpha=mean}$
1000	45029	0.00	45029
2000	42487	-24.89	31913
3000	39929	+0.91	40292
4000	57228	-13.76	49360
5000	30144	-8.57	27563
6000	17208	-6.43	16102
7000	41736	-22.92	32169
8000	35591	-3.93	34195
9000	39004	-0.86	38667
10000	27047	-3.72	26042
11000	26622	-7.96	24508
12000	33412	-4.37	31954
13000	44536	-26.55	32719
14000	32542	-4.40	31105
15000	21021	-8.37	19256
16000	40467	-7.15	37577
17000	22139	-2.47	21591
18000	15932	-10.28	15291
19000	24830	-4.53	23706
20000	40914	-0.00	40914
Mean	**33,890.9**	**-8.0**	**30,997.7**