



LBS Research Online

D den Hertog, [J Pauphilet](#), Y Pham, B Sainte-Rose and B Song
Optimizing the Path Towards Plastic-Free Oceans
Article

This version is available in the LBS Research Online repository: <https://lbsresearch.london.edu/id/eprint/3815/>

den Hertog, D, [Pauphilet, J](#), Pham, Y, Sainte-Rose, B and Song, B
(2024)

Optimizing the Path Towards Plastic-Free Oceans.

Operations Research.

ISSN 0030-364X

(In Press)

DOI: <https://doi.org/10.1287/opre.2023.0515>

INFORMS (Institute for Operations Research and Management Sciences)
<https://pubsonline.informs.org/doi/10.1287/opre.20...>

Users may download and/or print one copy of any article(s) in LBS Research Online for purposes of research and/or private study. Further distribution of the material, or use for any commercial gain, is not permitted.



Operations Research

Publication details, including instructions for authors and subscription information:
<http://pubsonline.informs.org>

Optimizing the Path Towards Plastic-Free Oceans

Dick den Hertog; , Jean Pauphilet; , Yannick Pham, Bruno Sainte-Rose, Baizhi Song;

To cite this article:

Dick den Hertog; , Jean Pauphilet; , Yannick Pham, Bruno Sainte-Rose, Baizhi Song; (2024) Optimizing the Path Towards Plastic-Free Oceans. Operations Research

Published online in Articles in Advance 11 Dec 2024

. <https://doi.org/10.1287/opre.2023.0515>

Full terms and conditions of use: <https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2024 The Author(s)

Please scroll down for article—it is on subsequent pages



With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes.

For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

Crosscutting Areas

Optimizing the Path Towards Plastic-Free Oceans

 Dick den Hertog,^a Jean Pauphilet,^{b,*} Yannick Pham,^c Bruno Sainte-Rose,^c Baizhi Song^b
^a Amsterdam Business School, University of Amsterdam, 1018 TV Amsterdam, Netherlands; ^b Management Science and Operations, London Business School, London NW1 4SA, United Kingdom; ^c The Ocean Cleanup, 3011 AD Rotterdam, Netherlands

*Corresponding author

Contact: d.denhertog@uva.nl,  <https://orcid.org/0000-0002-1829-855X> (DdH); jpauphilet@london.edu,

 <https://orcid.org/0000-0001-6352-0984> (JP); y.pham@theoceancleanup.com (YP); bruno.sainte-rose@theoceancleanup.com,

 <https://orcid.org/0000-0002-8067-8860> (BS-R); bsong@london.edu,  <https://orcid.org/0009-0009-4141-0595> (BS)

Received: September 23, 2023

Revised: April 12, 2024; July 6, 2024


Accepted: July 19, 2024

Published Online in Articles in Advance:
December 11, 2024

Area of Review: Environment, Energy, and Sustainability

<https://doi.org/10.1287/opre.2023.0515>
Copyright: © 2024 The Author(s)

Abstract. Increasing ocean plastic pollution is irreversibly harming ecosystems and human economic activities. We partner with a nonprofit organization and use optimization to help clean up oceans from plastic faster. Specifically, we optimize the route of their plastic collection system in the ocean to maximize the quantity of plastic collected over time. We formulate the problem as a longest path problem in a well-structured graph. However, because collection directly impacts future plastic density, the corresponding edge lengths are nonlinear polynomials. After analyzing the structural properties of the edge lengths, we propose a search-and-bound method, which leverages a relaxation of the problem solvable via dynamic programming and clustering, to efficiently find high-quality solutions (within 6% optimal in practice) and develop a tailored branch-and-bound strategy to solve it to provable optimality. On one year of ocean data, our optimization-based routing approach increases the quantity of plastic collected by more than 60% compared with the current routing strategy, hence speeding up the progress toward plastic-free oceans.

 **Open Access Statement:** This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. You are free to download this work and share with others for any purpose, except commercially, if you distribute your contributions under the same license as the original, and you must attribute this work as “*Operations Research*. Copyright © 2024 The Author(s). <https://doi.org/10.1287/opre.2023.0515>, used under a Creative Commons Attribution License: <https://creativecommons.org/licenses/by-nc-sa/4.0/>.”

Funding: J. Pauphilet and B. Song wish to acknowledge the financial support of the Research and Materials Development Fund (RAMD_Pauphilet_J_22/23_8789) at London Business School.

Supplemental Material: All supplemental materials, including the code, data, and files required to reproduce the results were reviewed and are available at <https://doi.org/10.1287/opre.2023.0515>.

Keywords: ocean cleaning • sustainable operations • longest path problem • dynamic programming • branch and bound • polynomial optimization

1. Introduction

Oceans are vital to life on earth: they home a vast array of plant and animal species and play a critical role in regulating the climate. In addition, they provide important economic benefits, for example, by supporting industries such as fishing, aquaculture, tourism, and the extraction of minerals. However, oceans are being threatened by growing and severe plastic pollution. As of 2015, 80% of the 6.3 billion tons of plastic waste ever generated ended up in landfills or the natural environment (Geyer et al. 2017). As of 2020, there were around three million tons of plastic waste floating in the oceans (Kaandorp et al. 2023). Furthermore, the amount of plastic emissions in the ocean increases by 4% every year (Kaandorp et al. 2023) with 0.5–2.7 million tons emitted via rivers every year (Lebreton et al. 2017, Schmidt et al. 2017, Meijer et al. 2021). Plastic pollution

is posing a threat to the marine ecosystem and the species that rely on it (Gall and Thompson 2015, Wilcox et al. 2015). It also has a detrimental impact on human activities. We refer to Li et al. (2016) for a comprehensive review of marine plastic pollution and its sources and effects. Because of its environmental and economic relevance, the reduction of ocean pollution is listed as an explicit target in the United Nations’ Sustainable Development Goal 14: Life Below Water.

The reduction of marine plastic pollution requires two concurrent actions: reducing yearly emissions and removing persistent legacy plastic pollution. Regarding the first effort, many legislative and nonlegislative actions have been taken to ban (or discourage) the use of single-use plastic (e.g., plastic bags or straws) with varying degrees of efficiency (see Schnurr et al. 2018 for a review). Given the importance of land-based pollution

and the role of rivers in transporting land-based pollution into the oceans, solutions also include improved in-land plastic waste management, recycling, and plastic interception in rivers (see, e.g., Dijkstra et al. 2021, Winterstetter et al. 2021). On the other hand, the active removal of plastic already emitted in the oceans has received lower attention and may be regarded as less efficient than preventing emissions because of the low average concentration of floating plastic in the oceans.

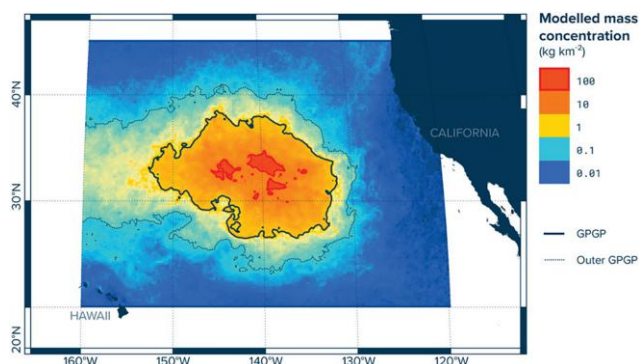
Fortunately, floating plastic debris get trapped in large circulating currents, called gyres, and tends to accumulate in specific areas called garbage patches. The largest of these five patches, the Great Pacific garbage patch (GPGP), is located halfway between California and Hawaii. Nearly 80,000 tons of plastic float inside this area of 1.6 million km² or three times the size of France (Lebreton et al. 2018). Figure 1 displays yearly average plastic density estimates in the GPGP. In short, plastic density in the GPGP is about 20 times higher than in the rest of the ocean.

The Ocean Cleanup is a Dutch nongovernmental organization whose mission is to clean up oceans from plastic. In addition to interception activities in rivers, it has developed a technology to collect plastic debris in the oceans. It has been trialing solutions in the GPGP since 2018 and operating its newest system since 2021. The system consists of a large (600-meter-wide and 4-meter-deep at the beginning of our collaboration) U-shaped screen, slowly dragged by two ships, which can capture floating plastics without capturing any marine animals. In this collaboration, we investigate the potential for improving the efficiency of the plastic collection system by optimizing its route in the GPGP. In particular, we use data and models about weather conditions and plastic density in the GPGP to construct an optimization-based routing algorithm that directly maximizes the quantity of plastic collected, hence speeding up the progress toward cleaner and healthier oceans.

1.1. Problem Description

The Ocean Cleanup's plastic collection system (which we later refer to as the system) is composed of two ships

Figure 1. (Color online) Yearly Plastic Density Map in the GPGP



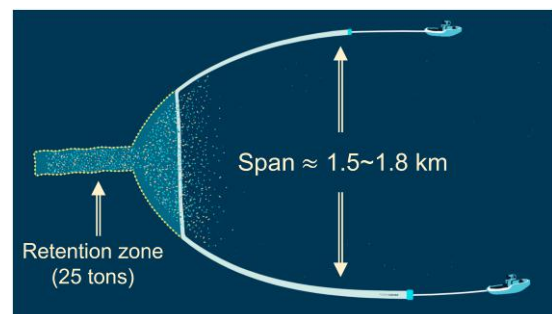
and a U-shaped screen (or net) as shown in Figure 2. In its original configuration (system 002/B), the screen had a span of 600 meters, but it has been increased to 1.4 km in the latest version of the system (system 03). It acts as an artificial coastline that intercepts floating debris. These floating plastic particles then gradually accumulate in a partially closed contraption at the apex of the screen called the retention zone.

Routing the system in the GPGP needs to satisfy some navigation requirements. To preserve the physical integrity of the screen, for example, the system can only slowly change course and cannot make any sharp turns. In addition, in order not to catch any fish or other marine life, the system moves at a fixed and low speed of around 1.5 knots (2.78 km/h). Finally, as any sea vessel, it is sensitive to weather and navigation conditions such as waves and wind. For example, when the wave height exceeds 4.5 meters, the system has to head against the waves to protect the screen. Above six-meter waves, the screen no longer intercepts any plastic.

The retention zone has a limited capacity of 25 metric tons and needs to be emptied regularly. The process of emptying the retention zone is called an extraction and is a complex operation: the screen is closed, the retention zone is hauled onto the ship deck and lifted by a crane, and the collected plastic is discharged on deck before being sorted and sent ashore for recycling. In particular, the crane cannot be operated when the wave height exceeds 2.5 meters. Overall, an extraction takes around 24 hours, during which the collection is stopped. Hence, extractions play an important role in the overall collection efficiency, and extraction scheduling should be incorporated in our search for a better routing system.

Our primary objective is to maximize the quantity of plastic collected. The Ocean Cleanup has developed a suite of models to estimate the dispersion and density of plastic in the GPGP (Klink et al. 2022) by using hind-cast and forecast models of ocean currents, waves, and wind. Assimilation methods (Peytavin et al. 2021) and plastic-specific transport models have also been investigated (Sainte-Rose et al. 2022). On the sensing front, satellite imaging (Park et al. 2021, 2022) and remote sensing techniques (de Vries et al. 2021) are being

Figure 2. (Color online) The Ocean Cleanup's System



developed to acquire field data. These models provide a picture of where plastics are located and how they move within this region, hence creating a dynamic view of present and future plastic density (similar to Figure 1 yet evolving over time). Our objective is to integrate these predictions directly into an optimal routing problem, so the system naturally accounts for plastic movements, which is crucial because both the system and the plastics are moving at comparable speeds. A central challenge is to account for the fact that the collection process removes plastic from the oceans and, as such, should directly impact the (estimate of) future plastic density. Hence, plastic density cannot be seen as an exogenous input to our model only; it is also impacted by our routing decisions.

To summarize, our objective is to find a route for the system and to schedule extractions of the retention zone in order to maximize the total quantity of plastic collected by the system. In particular, we need to account for weather and operational constraints, plastic dynamics, and the direct impact of our decision on future plastic density.

1.2. Contributions and Structure

In this work, we develop and validate an optimization approach to jointly optimize the routing and the extractions of the plastic collection system. After reviewing the relevant literature in Section 2, we make the following contributions:

- In Section 3, by discretizing space and time, we model the routing and scheduling decisions as paths in a directed acyclic graph (DAG). Among others, this model can account for relevant operational and weather constraints and provides efficient dynamic programming (DP) algorithms for the longest path type of optimization problems.

- Under this lens, the quantity of plastic collected can be seen as edge length in this graph and our problem as a longest path optimization problem. However, because of the direct impact of our routing decisions on future plastic density, our resulting optimization problem is a nonlinear and nondecomposable longest path problem. In Section 4, we formally analyze the structure of our path-dependent edge lengths, which resembles structure arising in covering problems. We derive lower and upper bounds on the estimation error obtained when ignoring the path dependency, which serves as the basis for our algorithmic strategy.

- We propose a search-and-bound strategy to efficiently find a high-quality solution for this class of problems with certificates of near optimality (Section 4). Our algorithm leverages a linear relaxation of the problem solvable via dynamic programming to efficiently search through the space of trajectories by combining geographical clustering with terminal values of the DP approach. We also propose a tailored branch-and-bound (B&B)

scheme to solve this class of problems exactly, using our search-and-bound algorithm as the root node analysis. On small instances (up to three-day planning), our search-and-bound strategy finds an optimal solution, scaling better with respect to the problem size than exact approaches.

- Finally, we evaluate the benefit of our search-and-bound algorithm on a one-year data set of ocean weather conditions and plastic density in Section 5. We find that our optimization approach yields at least a 60% improvement in terms of average collection efficiency compared with the current routing strategy. In particular, we observe greater benefits (+100%) during winter months because weather conditions (and wave height in particular) are limiting the ability to extract, hence exacerbating the benefit of jointly optimizing the route and the extraction schedule. In addition, our algorithm allows The Ocean Cleanup to explore the nonlinear impact of strategic system dimensioning decisions (e.g., span of the system and size of the retention zone).

2. Literature Review

Our problem can be summarized as a ship joint routing and scheduling problem, in which the objective is to steer the system in the GPGP and schedule extractions (i.e., emptying of the retention zone) in order to maximize the quantity of plastic collected. In Section 2.1, we review the optimization literature related to marine operations and ship routing. We then focus on methods for fishing optimization, which is similar to our plastic collection problem. Eventually, we model our problem as that of a longest path in an appropriately defined graph, so we review the literature on longest path optimization in Section 2.3.

2.1. Optimization for Ship Routing Problems

Following Granado et al. (2021), we divide the literature into weather and tactical routing.

In weather routing, the objective is to find a route that connects a given origin with a given destination and minimizes travel time or fuel consumption, which depend on weather and navigation conditions. The typical planning horizon in weather routing is a few weeks. The great circle passing through these two locations provides the shortest route in terms of travel distance. So an optimal route is often to be found in the vicinity of the shortest route. Most approaches create a discrete grid of potential locations around the great circle using isochrone lines (James 1957, Hagiwara and Spaans 1987) or a fixed grid (Zoppoli 1972, de Wit 1990). By representing a trajectory as a sequence of locations, the weather routing problem can, thus, be formulated as a shortest path problem, which can be solved efficiently by DP (Zoppoli 1972, de Wit 1990, Ting and Tzeng 2003, Meng and Wang 2011, Aydin et al. 2017) or Dijkstra's algorithm (Takashima et al.

2009, Skoglund 2012, Sen and Padhy 2015). Additional decision variables, such as engine power in Shao et al. (2012), can be modeled within a shortest path formulation by extending the description of the state of the ship. Heuristic methods are also used to deal with more complex objectives or constraints, such as simulated annealing (Kosmas and Vlachos 2012), the A* algorithm (Langbein et al. 2011, Yoon et al. 2018), or particle swarm optimization methods (Zheng et al. 2019). Recently, Cheng and Zhang (2018) and Chen et al. (2019) use a reinforcement learning approach to optimize the route, learning the complex dynamics between waves and speed or fuel consumption. We refer to Zis et al. (2020) for a comprehensive review on ship weather routing.

For our problem, we adopt a similar modeling paradigm by discretizing the location of the system in the GPGP. However, we adopt a more fine-grained discretization of time (three-hour time steps) and space (8 km) and extend the state variable to account for extraction decisions as well, so our resulting graph is of much larger scale, that is, in the order of 10^6 nodes. In addition, the destination in our problem is not fixed, which can lead to more complex trajectories, such as circling or crossing. In terms of objective, we assume that the fuel efficiency does not depend on the routing decision because of the limited propelling speed, so our primary objective is to maximize the amount of plastic collected in a given amount of time. After appropriately defining edge weights, we formulate our problem as a longest path optimization problem and solve it using DP strategies similar to the ones used in weather ship routing.

Tactical ship routing consists in finding the lowest cost route for a ship that needs to visit different locations (e.g., a cargo ship visiting different ports). Because the time horizon is long (several weeks or months) and the ports are fixed isolated locations, the problem can be formulated as a traveling salesperson problem (TSP) solved by branch-and-bound (Appelgren 1971, Stalhane et al. 2015), branch-cut-and-price (Battarra et al. 2014), or heuristic methods (Malaguti et al. 2018) or DP (Fagerholt and Christiansen 2000). We refer to Christiansen et al. (2004) for a comprehensive review of the literature and its connection to supply chain management.

2.2. Fish Routing

Among all maritime activities, fishing is the most comparable to our plastic collection problem because the objective is to capture floating elements in the oceans.

Before solving any route optimization problem, one needs to first predict the density of fish at different locations. However, unlike plastic, fish are actively moving, which makes their precise location highly unpredictable. Instead, most works describe fish density with coarse granular distributions (see Robinson

et al. 2017 for a review). Unfortunately, estimating the accuracy of these different approaches remains an open challenge. Indeed, 94% of the studies reviewed by Robinson et al. (2017) failed to report the uncertainty of their model.

In the weather routing literature, algorithms that consider wave and wind forecast to design safe and efficient routes have been applied to fishing (e.g., Vetter et al. 2016). In these use cases, the objective of maximizing the quantity of fish collected is captured in the choice of the target destination and is typically left to the end user. This implementation bypasses the issue of inaccurate predictions by letting the human user identify (based on quantitative models and intuition) the destination. To the best of our knowledge, no weather ship routing approach uses quantitative fish density predictions directly as an input to optimize the short-term (within the next days) route of fishing ships. Instead, predictions on the presence and movements of fish banks or clusters are mostly used as locations in a tactical ship routing problem. For tuna fishing, for example, floating devices are dispersed in the ocean to attract fish. Groba et al. (2015, 2018, 2020) model the problem of visiting all devices as a dynamic TSP, in which locations can drift because of sea current.

2.3. Optimization for Longest Path

Given weights on the edges of a graph, the length of a path is defined as the sum of the weights of the edges composing the path. The problem of finding the longest path in a graph is shown to be \mathcal{NP} -complete as a generalization of the Hamiltonian path problem (Karp 2010). Actually, the longest path problem cannot be approximated in polynomial time unless $\mathcal{P} = \mathcal{NP}$ as proved by Karger et al. (1997) for undirected and Björklund et al. (2004) for directed graph.

In contrast, finding the shortest path in a graph can be solved in polynomial time using algorithms such as the greedy-type Dijkstra's algorithm (Dijkstra 1959, Dantzig 1960) or the Bellman–Ford algorithm (Shimbel 1955, Ford 1956, Bellman 1958, Moore 1959). We refer to Pollack and Wiebenson (1960) and Schrijver (2012) for comprehensive reviews. Understanding the structural differences between the longest and shortest path problems and their implications for problem complexity has been a vivid research topic (see, e.g., Cormen et al. 2022), unraveling conceptual connections between shortest path algorithms and DP (Sniedovich 2006).

Nonetheless, polynomial time algorithms for longest path problems exist for particular classes of graphs, such as trees (Bulterman et al. 2002, Uehara and Uno 2007), block graphs, cactus graphs (Uehara and Uno 2007), and cocomparability graphs (Ioannidou and Nikolopoulos 2013). The graph we propose in Section 3.1 is a DAG. The longest path problem in a DAG can be solved in linear time by transforming it into the

shortest path problem (Pandit 1962, Cormen et al. 2022) or using DP on the topological sort of the DAG (Madraki and Judd 2019). The DAG in our project has a natural topological sort, and we use DP in Section 3.2 to solve linear longest path problems.

3. Graph-Based Routing Model

In this section, we propose a graph-based formulation for our problem. By discretizing time and space, we show in Section 3.1 how the routing decision can be modeled as a path in a sparse directed acyclic graph. Accordingly, longest path optimization problems can be solved efficiently over this graph using DP as presented in Section 3.2. We conclude this section by discussing how extraction scheduling decisions can be incorporated as well—details are deferred to Online Section A.3 of the electronic companion—and identifying a set of tractable optimization problems we can solve in Section 3.3.

3.1. Discretization and Graph Representation

To describe the system’s trajectory and model the key decisions and constraints of our problem, we discretize space onto a finite grid as represented in Figure 3. Similarly, we divide our planning horizon using a fixed time step. In our implementation, we use an 8-km step to discretize space and a three-hour time step. Denoting \mathcal{L} as the set of all possible locations and $\mathcal{T} := \{0, 1, \dots, T\}$ as the set of time periods, we can represent a system trajectory as a sequence of locations, $\{\ell_t\}_{t \in \mathcal{T}}$ with $\ell_t \in \mathcal{L}$.

However, as explained in Section 1.1, the steering direction also plays an important role in our problem because of operational (e.g., no sharp turn) and weather constraints (e.g., if the wave height exceeds 4.5 meters, need to navigate against the waves). Hence, at a given time t , knowledge of the current location ℓ_t is not sufficient to determine whether the constraints are satisfied and what the accessible next locations are. Accordingly,

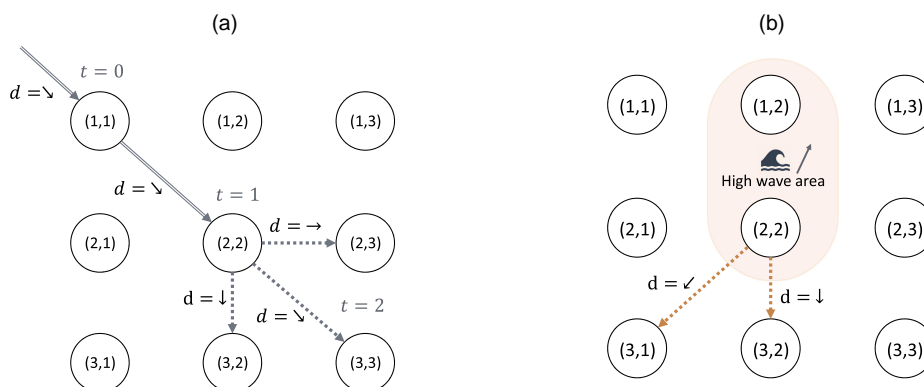
we describe a trajectory by a sequence $\{(\ell_t, d_t)\}_{t \in \mathcal{T}}$, where ℓ_t is the location at time t and $d_t \in \mathcal{D}$ is the steering direction at time t . Here, \mathcal{D} denotes the (finite) set of allowable steering directions in our grid, $\mathcal{D} := \{\uparrow, \nearrow, \rightarrow, \searrow, \downarrow, \swarrow, \leftarrow, \nwarrow\}$. For example, we can easily enforce the no-sharp-turn constraint by limiting the change in direction between t and $t + 1$, that is, the angle between d_t and d_{t+1} (no more than 45° in our case).

Figure 3(a) shows an example of a discretized trajectory. For simplicity, we associate \mathcal{L} with a set of discrete coordinates in \mathbb{R}^2 , which are in a one-to-one correspondence with the latitude and longitude of the system. In this example, at $t = 0$, the system is at location $(0,0)$, moving southeast (\searrow). It keeps the same steering direction at $t = 1$ and reaches $(1,1)$. At $t = 2$, it can reach three different locations depending on whether it continues southeast ($d_2 = \searrow$) or decides to change course and move \rightarrow or \downarrow . Because the propelling speed of our system is limited (in order not to catch any marine life), we use discretization steps for time (three hours) and space (8 km) that are consistent with this low propelling speed (1–1.5 knots) and assume that the system in one location can only reach the neighboring locations at the next time period. We could relax this assumption and adopt a finer discretization strategy to account for travel time differences between diagonal and horizontal/vertical moves or allow for different propelling speed depending on the steering direction (e.g., to maintain a constant speed relative to water).

Using terminology from DP, we refer to the triplet $s := (\ell, d, t) \in \mathcal{L} \times \mathcal{D} \times \mathcal{T}$ as the state of the system. For each state $s = (\ell, d, t)$, we can then define its set of successors, that is, the set of admissible next states $s' = (\ell', d', t + 1)$ that satisfy all operational and weather constraints, such as

- Consistency between locations and directions: The next location ℓ' needs to correspond to the location reached from ℓ after following the direction d' , which

Figure 3. (Color online) Example: Routing in a Small Grid



Notes. (a) Discretization of path. (b) High wave region and direction at $t = 1$.

we can express algebraically as $d' = \ell' - \ell''$ after appropriately mapping \mathcal{L} and \mathcal{D} to vectors in \mathbb{R}^2 .

- No sharp angles: In our problem, the angle between the steering directions d and d' is at most 45° (or $\pi/4$). For example, for $s = ((2,2), \searrow, 1)$ in Figure 3(a), we must have $d' \in \{\rightarrow, \searrow, \downarrow\}$.

- High-wave regions: When the wave height exceeds 4.5 meters, the system has to navigate against the waves. For example, in Figure 3, (a) and (b), assume that the location at $t = 1$ is in a high-wave region with waves going north–northeast. Then, according to this constraint, the next direction d' can only be \swarrow or \downarrow .

Our model can, thus, account for any constraint defined on the location or direction of the system; we provide a list of the operational constraints of our problem in Online Section A.1. In Online Section A.3, we describe how to extend the state space further to incorporate the decision of extraction scheduling and impose the associated constraints (in particular, extraction can only be performed when the wave height is below 2.5 meters). Altogether, these constraints define the successors of a state s , which we concisely denote $\text{succ}(s)$. For the example, in Figure 3, we have $\text{succ}((2,2), \searrow, 1) = \{((3,2), \downarrow, 2)\}$. In other words, the system only has one feasible next state given the current state and weather conditions.

With these notations, we can represent admissible trajectories as paths on a graph $\mathcal{G} = (\mathcal{S}, \mathcal{E})$. The set of nodes \mathcal{S} can be naturally partitioned by the time period $t \in \mathcal{T}$, that is, $\mathcal{S} = \cup_{t \in \mathcal{T}} \mathcal{S}_t$, where \mathcal{S}_t is the set of feasible states at time t and is defined recursively. At time $t = 0$, if we are given an initial location ℓ_0 only, then the set of all possible initial states is $\mathcal{S}_0 = \{(\ell_0, d, 0) : d \in \mathcal{D}\}$. We then apply the recursion $\mathcal{S}_{t+1} = \cup_{s \in \mathcal{S}_t} \text{succ}(s)$. This offers the flexibility to fix (or not) the starting/ending point of the trajectory. Similarly, the set of edges can be decomposed into $\mathcal{E} = \cup_{t=1}^T \mathcal{E}_t$, with $\mathcal{E}_{t+1} = \{(s, s') \in \mathcal{S}_t \times \mathcal{S}_{t+1} : s' \in \text{succ}(s)\}$. In particular, observe that the graph \mathcal{G} is a DAG. Furthermore, it is relatively sparse. Because of the no-sharp-turn constraint, the number of edges satisfies $|\mathcal{E}_t| = \mathcal{O}(|\mathcal{S}_t|)$. Online Figure A.1 shows the graph corresponding to the example of Figure 3. In our implementation, we plan for seven days with three-hour time steps, so $T = 7 \times 8 = 56$ and the grid of all reachable locations is of size $|\mathcal{L}| = (56 + 1 + 56) \times (56 + 1 + 56) = 12,769$. Hence, for each time t , the number of possible states for time t is bounded as follows: $|\mathcal{S}_t| \leq |\mathcal{S}| \times |\mathcal{D}| \approx 10^5$.

3.2. Efficient Search for Longest Path

Thanks to the convenient structure of the graph \mathcal{G} , given fixed weights on the edges, w_e for $e \in \mathcal{E}$, we can efficiently find the longest path, that is, the reward-maximizing trajectory. Assuming that w_e is associated with the quantity of plastic collected when the system moves along edge e at time t , our plastic collection

problem is equivalent to finding the longest path in \mathcal{G} with edges weighted by w . Because \mathcal{G} is a DAG, the longest path can be found using a DP algorithm. We present the key ingredients and intuition in this section for the sake of completeness and defer a complete description of the longest path algorithm (Algorithm A.1) in the appendix.

For any state $s \in \mathcal{S}_{t+1}$, let $V^{t+1}(s)$ denote the length of the longest path connecting s to \mathcal{S}_0 :

$$V^{t+1}(s) := \max_{s_0 \in \mathcal{S}_0, \dots, s_t \in \mathcal{S}_t} \sum_{\tau=0}^t w_{s_\tau, s_{\tau+1}} \text{ with } s_{t+1} = s.$$

The key idea in the DP algorithm is that a solution of the optimization problem above can be computed recursively by connecting the longest path between \mathcal{S}_0 and s' and the edge (s', s) for some $s' \in \mathcal{S}_t$, that is, $V^{t+1}(s) = \max_{s' \in \mathcal{S}_t: s \in \text{succ}(s')} \{w_{s', s} + V^t(s')\}$. The latter maximization problem is solved by exhaustively searching through \mathcal{S}_t .

Algorithm A.1 proceeds by recursively computing the values $V^t(s)$ for $t = 0, \dots, T$. At the end, Algorithm A.1 returns, for every possible terminal state $s \in \mathcal{S}_T$, the value of the longest path problem across all possible paths terminating at s , $V^T(s)$, alongside a candidate path achieving this value. Figure 4 shows an example of a seven-day collection route obtained by applying Algorithm A.1.

3.3. Summary: Power of the Graph-Based Modeling

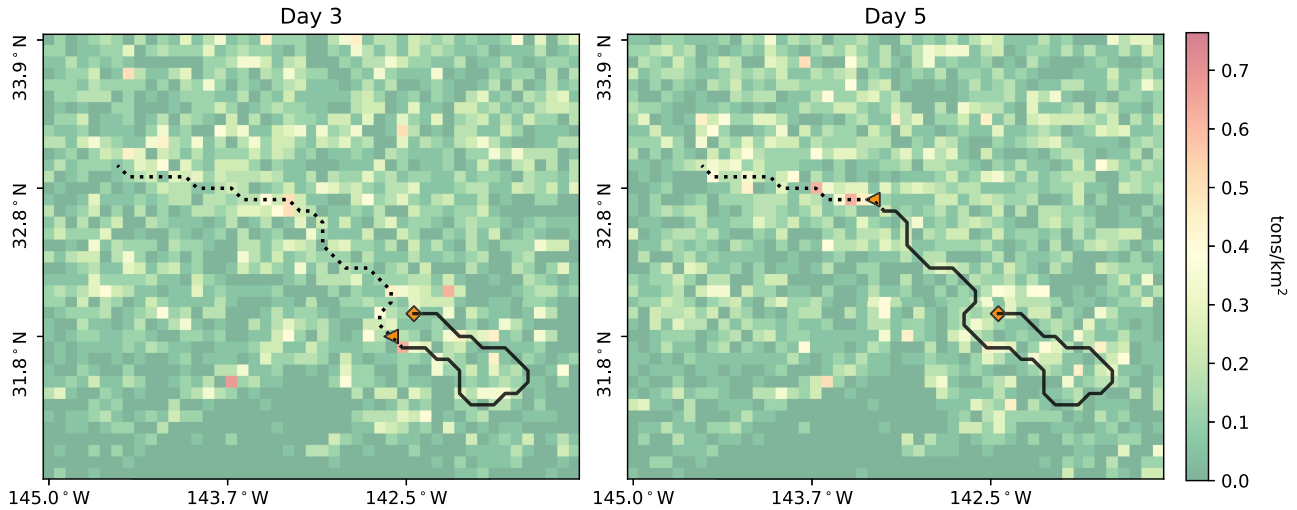
In this section, we propose a graph-based model to represent the routing decision as a path in a sparse DAG, \mathcal{G} . In Online Section A.3, we show how we can extend the state space \mathcal{S} of our system and the graph \mathcal{G} to also account for extraction scheduling. Hence, we obtain a similar DAG in which each path now corresponds to a sequence of routing and extraction scheduling decisions.

The DP algorithm described in Section 3.2 (Algorithm A.1) is an efficient approach for solving longest path problems over this graph. In this work, we are particularly interested in problems in which rewards are associated with states instead of edges, that is, longest path problems of the form

$$\max_{x \in \mathcal{X}} \sum_{t \in \mathcal{T}} \sum_{s \in \mathcal{S}_t} r_s^t x_s^t \quad (1)$$

where $x_s^t \in \{0, 1\}$ indicates whether the system is in state s at time t and \mathcal{X} denotes the set of admissible such binary variables. A formal definition of the feasible set \mathcal{X} is provided in G.4. Of course, problems of the form (1) can be solved by Algorithm A.1 as longest path problems with weights $w_{s, s'} = r_{s'}^t$ for $(s, s') \in \mathcal{E}_t$. Conceptually, one can interpret the DP approach as an efficient partitioning of the set \mathcal{X} . At the end of Algorithm A.1, the set of trajectories \mathcal{X} is partitioned

Figure 4. (Color online) Example of an Optimal Seven-Day Route (Starting at the Diamond Location on Day 1), Represented on Days 3 and 5 (Triangle Locations) on Top of the Plastic Density Map for Each Day



according to the terminal state they reach. The quantity $V^T(s)$ corresponds to the value of the longest path problem across all paths terminating at s , and we can get one path achieving this value, denoted x^s .

4. Path-Dependent Reward Structure and Algorithms

Because our graph-based formulation presented in Section 3.1 allows us to efficiently find trajectories that maximize a given reward, a natural approach is to cast our plastic collection problem as a longest path problem. Because plastics are freely floating in the oceans, however, plastic particles are constantly moving, and the plastic collecting system directly impacts these dynamics. We present the dynamics of plastic movements in the absence of any collection in Section 4.1 and then derive the resulting dynamics for our objective function in Section 4.2. In particular, the rewards (or edge length) we need to consider are path-dependent; that is, they depend on the entire past trajectory of the system. We analyze the structure of such path-dependent rewards in Section 4.3. Based on this analysis, we propose an efficient search-and-bound algorithm to find near-optimal solutions to the resulting nonlinear optimization problem efficiently (Section 4.4), and a tailored branch-and-bound scheme that solves it to optimality (Section 4.5). We evaluate our algorithms numerically in Section 4.6.

4.1. Fluid Mechanics Model of Free-Floating Plastic Dispersal

Plastic particles move passively in the oceans, and as such, their movements can be modeled and predicted using fluid dispersal models. Among others, the

engineering team at The Ocean Cleanup models the velocity of plastic particles in the oceans using data on sea currents and waves and taking into account the Stokes drift (Stokes 1847) and Eddy diffusivity (Taylor 1915) phenomena.

Denoting r_ℓ^t as the quantity of plastic present at time $t \in \mathcal{T}$ and at location $\ell \in \mathcal{L}$, these fluid dispersal models provide us with estimates on the quantity of plastic present in the region at times (r^0, \dots, r^T) as well as structural relationships connecting the vectors. Formally, from the models developed by The Ocean Cleanup, we also obtain matrices $Q^t \in \mathbb{R}_+^{\mathcal{L} \times \mathcal{L}}$ such that

$$r^{t+1} = Q^t r^t. \quad (2)$$

Each entry $Q_{\ell', \ell}^t$ of the matrix Q^t indicates the fraction of plastic present at location ℓ' at time t that moves to location ℓ at time $t + 1$. If the total quantity of plastic is constant (which is a reasonable assumption given our relatively short planning horizon), then the matrix Q^t should be left stochastic (i.e., $\sum_{\ell \in \mathcal{L}} Q_{\ell', \ell}^t = 1$ for all $\ell' \in \mathcal{L}$). In this case, we could interpret Q^t as the transition matrix of a Markov process. In our approach, we only require that Q^t has nonnegative entries.

Remark 1. In practice, the matrices Q^t are large, $125,000 \times 125,000$ in our implementation, so matrix vector products involving Q^t s can be computationally challenging. Actually, the construction of the matrices Q^t from the particle-level fluid dispersal model is the most time-consuming step. We discuss computational aspects of the plastic dynamics model in Online Section C.

4.2. Path Dependency

Given this information, we now define a relevant objective for our longest path problem. Whereas the plastic

density vectors (or maps) r^t , $t \in \mathcal{T}$, defined in the previous section describe the plastic dynamics in absence of any collection process, our system actively removes plastic from the ocean, and the collected plastic no longer evolves according to (2). In other words, the quantity of plastic collected by our system (and the locations where this plastic has been collected) directly impacts the future spatial distribution of plastic. It is relevant to our optimization problem because our system moves at a speed comparable to that of the plastic. We refer to this phenomenon as path dependency and now appropriately define a reward (or length) vector for our optimization problem that takes this phenomenon into account.

Let us denote the location of the system at time t through a one-hot vector $x^t \in \{0,1\}^{\mathcal{L}}$, where $x_\ell^t = 1$ if and only if the system is in ℓ at time t . We denote by $r_{|x^{0:t-1}}^t \in \mathbb{R}_+^{\mathcal{L}}$ the quantity of plastic present (or reward) associated with each location at time t , where $x^{0:t-1}$ concisely denotes the sequence $\{x^0, \dots, x^{t-1}\}$ and emphasizes the dependency on the past trajectory. If the system is in location ℓ at time t , it collects a fraction $\alpha \in [0,1]$ of the plastic present. Hence, it collects $\alpha r_{\ell|x^{0:t-1}}^t$ and the remaining $(1-\alpha)r_{\ell|x^{0:t-1}}^t$ continues to float in the ocean, together with the plastic present in other locations, $r_{\ell'|x^{0:t-1}}^t$ for $\ell' \neq \ell$. Altogether, the spatial density of plastic at time $t+1$, $r_{|x^{0:t}}^{t+1}$, should depend on $r_{|x^{0:t-1}}^t$ and x^t through the following recursion:

$$r_{|x^{0:t}}^{t+1} = Q^t(r_{|x^{0:t-1}}^t - \alpha r_{|x^{0:t-1}}^t \circ x^t), \quad (3)$$

where \circ denotes the Hadamard or element-wise product between two vectors. Hence, $r_{|x^{0:t-1}}^t - \alpha r_{|x^{0:t-1}}^t \circ x^t$ corresponds to the density map on which we remove a fraction α of the plastic in the location of the cleaning system.¹ Note that, because Q^t , x^t , and r^0 have nonnegative entries, one can show by induction that $r_{|x^{0:t}}^{t+1} \geq 0$.

With these dynamics in mind, the problem of jointly routing the system and scheduling the extractions in order to collect the maximum amount of plastic possible can be formulated as the following longest path optimization problem:

$$\max_{x \in \mathcal{X}} \sum_{t \in \mathcal{T}} \sum_{s \in \mathcal{S}_t} r_{s|x^{0:t-1}}^t x_s^t \quad \text{s.t.} \quad r_{|x^{0:t}}^{t+1} = Q^t(r_{|x^{0:t-1}}^t, x^t), \quad (4)$$

which is analogous to the longest path problem (1) except that the rewards are no longer fixed, but also depend on the past decisions, $x^{0:t}$. Unfortunately, Problem (4) is much more challenging to solve than (1) because the objective is nonlinear. To better understand the dynamics and complexities of the problem, we study analytically the path-dependent reward vectors defined by Recursion (3) in the following section.

Remark 2. Note that, with a slight abuse of notation, we use the variable x_s^t in Problem (4) to encode for the

state of the system at time t , whereas plastic dynamics (3) are described using binary variables x_ℓ^t encoding for the location of the system (location being one component of the state only) and similarly for the associated rewards. However, it should be clear that we can recover the location from the system's state via a simple affine mapping and that the reward dynamics described at a location level in (3) imply similar dynamics for the state rewards. We formally define this mapping in Online Section B.1 and introduce a generic operator Q^t in the optimization problem (4) to concisely capture the resulting dynamics on the state variables/rewards. In the remainder of this section, for ease of notation, we implicitly work with location-based x variables when analyzing the structure of the rewards generated by the recursive Equation (3) but refer to the state-level variables when describing optimization problems and algorithms. This simplification is valid because the mapping between the two descriptions is monotonous.

4.3. Reward Decomposition

We analyze the structure of the path-dependent reward $r_{|x^{0:t}}^{t+1}$ to inform our algorithmic strategy.

To build intuition, we start by the special case in which plastic does not move, that is, when the matrices Q^t are the identity matrices. In this case, in the absence of any plastic collection, the plastic density maps r^t defined by Equation (2) are constant over time, $r^0 = \dots = r^T =: r$. Accordingly, we drop the time superscript although the path-dependent reward, $r_{|x^{0:t-1}}$ still depends on time t through the past trajectory $x^{0:t-1}$. In this case, we have the following expansion.

Lemma 1. *When the matrices Q^t are all equal to the identity matrix, we have*

$$r_{|x^{0:t}} = r + \sum_{1 \leq k \leq t} (-\alpha)^k \sum_{0 \leq t_1 < \dots < t_k \leq t} r \circ x^{t_1} \circ \dots \circ x^{t_k}.$$

Proof of Lemma 1. In this case, the plastic dynamics (3) simplify as

$$\begin{aligned} r_{|x^{0:t}} &= r_{|x^{0:t-1}} - \alpha r_{|x^{0:t-1}} \circ x^t = r_{|x^{0:t-1}} \circ (\mathbf{1} - \alpha x^t) \\ &= r \circ (\mathbf{1} - \alpha x^0) \circ \dots \circ (\mathbf{1} - \alpha x^t). \end{aligned}$$

The Hadamard product being commutative and associative, we can use the classical polynomial expansion technique to obtain

$$\begin{aligned} r_{|x^{0:t}} &= r \circ \left[\sum_{0 \leq k \leq t} (-\alpha)^k \sum_{0 \leq t_1 < \dots < t_k \leq t} x^{t_1} \circ \dots \circ x^{t_k} \right] \\ &= \sum_{0 \leq k \leq t} (-\alpha)^k \sum_{0 \leq t_1 < \dots < t_k \leq t} r \circ x^{t_1} \circ \dots \circ x^{t_k}. \quad \square \end{aligned}$$

Lemma 1 shows that the path-dependent reward $r_{|x^{0:t}}$ can be computed from r by applying successive

corrections. The k th order correction in this expansion involves Hadamard products of the form $x^{t_1} \circ \dots \circ x^{t_k}$, each of them being different from the $\mathbf{0}$ vector if and only if there exists a location ℓ such that $x_\ell^{t_1} = \dots = x_\ell^{t_k} = 1$. In other words, the first order correction consists in removing a fraction α of the plastic in locations visited at least once by the system, the second order correction adds a fraction α^2 of the plastic in locations visited at least twice by the system, and so on. In the general case, the plastic particles move according to the matrix Q^t , so they are not assigned to a fixed location. Still, the intuition of Lemma 1 holds: the path-dependent rewards $r_{|x^{0:t-1}}^t$ can be obtained from the original rewards r^t by removing a fraction α of the plastic particles encountered once, adding a fraction α^2 of the plastic particles encountered twice, etc. We derive analytically the order two expansion (in α) of the path-dependent reward in the general case in Online Section B.2.

Remark 3. The terms in the expansion in Lemma 1 decay exponentially in k because of the α^k term and because the number of locations being updated decreases:

$$\sum_{0 \leq t_1 < \dots < t_k < t_{k+1} \leq t} x^{t_1} \circ \dots \circ x^{t_k} \circ x^{t_{k+1}} \leq \sum_{0 \leq t_1 < \dots < t_k \leq t} x^{t_1} \circ \dots \circ x^{t_k}, \quad \forall k \geq 0.$$

The structure of these expansions is analogous to the inclusion–exclusion principle from probability. From this analogy, one can expect that truncating the expansion at a fixed order k with k even (respectively, odd) leads to an upper (lower) bound on the path-dependent reward. Indeed, Proposition 1 shows that zeroth and first order expansion provides valid upper and lower bounds, respectively, on the path-dependent reward.

Proposition 1. *The path-dependent reward $r_{|x^{0:t}}^{t+1}$ satisfies the following bounds:*

$$r^{t+1} - \alpha \sum_{0 \leq t_1 \leq t} (Q^{t_1} \times \dots \times Q^{t_1})(r^{t_1} \circ x^{t_1}) \leq r_{|x^{0:t}}^{t+1} \leq r^{t+1}. \quad (5)$$

Proof of Proposition 1. We prove the result by induction. For $t = 0$, $r_{|x^{0:0}}^1 = r^1 - \alpha Q^0(r^0 \circ x^0)$. Hence, in this case, $r_{|x^{0:0}}^1$ is exactly equal to the lower bound in (5). Because Q^0 , r^0 , and x^0 have nonnegative entries, $Q^0(r^0 \circ x^0) \geq 0$, and the upper bound in (5) holds as well.

Let us now assume that Bounds (5) hold for some $t \geq 0$, and let us show that they also hold for $t + 1$.

For the upper bound,

$$\begin{aligned} r_{|x^{0:t+1}}^{t+2} &= Q^{t+1} r_{|x^{0:t}}^{t+1} - \alpha Q^{t+1}(r_{|x^{0:t}}^{t+1} \circ x^{t+1}) \leq Q^{t+1} r_{|x^{0:t}}^{t+1} \\ &\leq Q^{t+1} r^{t+1} = r^{t+2}. \end{aligned}$$

For the lower bound,

$$\begin{aligned} r_{|x^{0:t+1}}^{t+2} &= Q^{t+1} [r_{|x^{0:t}}^{t+1} \circ (\mathbf{1} - \alpha x^{t+1})] \\ &\geq Q^{t+1} \left[\left(r^{t+1} - \alpha \sum_{0 \leq t_1 \leq t} (Q^{t_1} \times \dots \times Q^{t_1})(r^{t_1} \circ x^{t_1}) \right) \right. \\ &\quad \left. \circ (\mathbf{1} - \alpha x^{t+1}) \right] \\ &= Q^{t+1} r^{t+1} - \alpha Q^{t+1} \sum_{0 \leq t_1 \leq t} (Q^{t_1} \times \dots \times Q^{t_1})(r^{t_1} \circ x^{t_1}) \\ &\quad - \alpha Q^{t+1}(r^{t+1} \circ x^{t+1}) + \alpha^2 \underbrace{\dots}_{\geq 0} \\ &\geq r^{t+2} - \alpha \sum_{0 \leq t_1 \leq t+1} Q^{t+1} \times (Q^{t_1} \times \dots \times Q^{t_1})(r^{t_1} \circ x^{t_1}), \end{aligned}$$

which concludes the proof. \square

Our analysis of the structure of the path-dependent rewards highlights the fact that path dependency is essentially a double-counting issue: the path-dependent rewards count the quantity of plastic that can be collected by the system. If a plastic particle has already been captured by the system in the past, it should not be counted again to estimate future rewards. This structure is also present in optimization problems with coverage objectives. In the maximal covering location problem (Church and ReVelle 1974), for example, the total coverage is not the sum of the regions covered by each facility separately because regions covered by more than one facility should not be counted multiple times. Our problem offers a novel real-life example of this classical structure with the following twists: a large number of regions to cover (plastic particles in the GPGP in our language), dispersion dynamics, and the presence of an efficiency ratio ($\alpha \in (0, 1]$).

Conceptually, the reward vectors $r_{|x^{0:t-1}}^t$ are order- t polynomials in x , leading to the hard nonlinear optimization problem (4). However, by Section 3.2, we know how to efficiently maximize linear functions over the feasible set \mathcal{X} . Hence, in the rest of this section, we develop and validate the following algorithmic strategy: a linear relaxation of the path-dependent reward and an efficient feasible solution search to find high-quality solutions quickly combined with a tailored branching scheme to refine this approximation and ultimately converge to an optimal solution.

Remark 4. Proposition 1 motivates us to relax (or upper bound) $r_{|x^{0:t-1}}^t$ by a reward vector that does not depend on x and to refine this approximation by branching. Alternatively, we could construct tighter relaxations by using the last k locations ($k \geq 0$) and converge to an optimal solution by taking $k \rightarrow T$. Although less scalable than our tailored branch and bound, we present and evaluate this alternative strategy in Online Section D.5.

4.4. Search-and-Bound Heuristic

In this section, we propose an algorithm for finding a provable near-optimal solution to Problem (4) quickly. Our algorithm (Algorithm 1) relies on a linear relaxation solvable by DP (Algorithm A.1), followed by a search through the feasible space, which is informed by a clustering of the trajectories based on their last location and the terminal values obtained from the DP algorithm, $V^T(s)$. Algorithms A.1 and 1 can be interpreted as relaxation-induced searches (Danna et al. 2005) that constitute the root node analysis in our branch-and-bound scheme.

Instead of solving (4), we solve the linear longest-path problem (1) with the path-independent rewards r^t as the objective. According to Proposition 1, this linear problem provides an upper bound on (4), that is, constitutes a valid relaxation. This result is intuitive: ignoring the effect of our collection on future plastic collection leads to an optimistic (i.e., over) estimate of the plastic we can actually collect. Furthermore, for any t , Proposition 1 states that $\|r^t - r^t_{|x^{0:t-1}}\| = \mathcal{O}(\alpha t)$; hence, the total relaxation error scales at most like αT^2 .

For concision, let us omit the time superscript in this section and concisely denote r as the plastic density maps obtained by applying the fluid advection Equations (2) without any active collection and $r_{|x}$ its path-dependent version. Both Problem (4) and its linear relaxation optimize over the same feasible space, $x \in \mathcal{X}$, but differ in their objective function, $\langle r_{|x}, x \rangle$ and $\langle r, x \rangle$, respectively, where we use $\langle \cdot, \cdot \rangle$ to denote the inner product between two vectors indexed by time $t \in \mathcal{T}$ and state $s \in \mathcal{S}$. Let us denote $x^*(r_{|x})$ and $x^*(r)$ as their respective solutions.

With these notations, $x^*(r)$ is the solution returned by Algorithm A.1. Because $r_{|x} \leq r$ (Proposition 1) and because flows are nonnegative,

$$\langle r_{|x^*(r_{|x})}, x^*(r_{|x}) \rangle \leq \langle r, x^*(r_{|x}) \rangle \leq \langle r, x^*(r) \rangle =: UB,$$

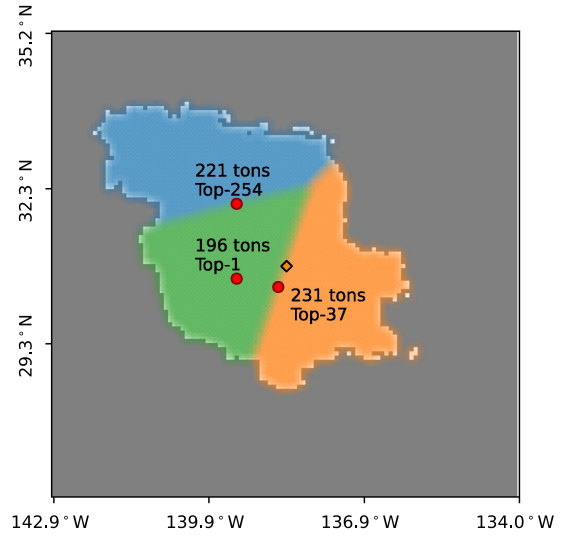
where the last inequality follows from the fact that $x^*(r_{|x}) \in \mathcal{X}$ is feasible for (1) and the optimality of $x^*(r)$. In addition, $x^*(r) \in \mathcal{X}$ is feasible for (4), so

$$LB := \langle r_{|x^*(r)}, x^*(r) \rangle \leq \langle r_{|x^*(r_{|x})}, x^*(r_{|x}) \rangle.$$

Note that the reward vectors in the two sides of this inequality are different because they correspond to different paths $x^*(r)$ and $x^*(r_{|x})$. Hence, the path-dependent reward achieved by $x^*(r)$ also provides a valid lower bound on the value of (4). Altogether, Algorithm A.1 can be used to return a feasible solution to Problem (4) alongside an optimality gap.

Furthermore, any other feasible solution $x \in \mathcal{X}$ provides a valid and potentially better solution. In Algorithm 1, we propose a procedure to search for a better feasible solution $x \in \mathcal{X}$, which we illustrate in Figure 5. Remember that, for any terminal state s , Algorithm A.1 returns the length (according to r) of the longest path

Figure 5. (Color online) Illustration of Our Cluster-Based Search Strategy on One Seven-Day Planning Instance (January 15, 2002)



Notes. We first exclude locations that are provably suboptimal (gray region), cluster the remaining area into K clusters (here, $K = 3$), and then evaluate one trajectory per cluster (circle dots). The starting location is indicated by a diamond at the center of the map.

reaching s , $V^T(s)$, as well as one path of that length, x^s . First, we exclude the terminal states $s \in \mathcal{S}_T$ such that $V^T(s) < LB$ because they cannot contain any solution better than $x^*(r)$ (the outer gray zone in Figure 5). Then, we cluster the space of remaining terminal states into K regions $\mathcal{S}_T^{(k)}$ based on geographical coordinates (we have $K = 3$ in Figure 5). For each region, we consider one candidate trajectory, corresponding to the state within $\mathcal{S}_T^{(k)}$ with the largest terminal value $V^T(s)$ and evaluate its path-dependent reward. This step, (\star) , is the most computationally expensive part of Algorithm 1. Alternatively, one could have searched through \mathcal{S}_T by decreasing terminal value $V^T(s)$. As illustrated in Figure 5, however, searching by geographical cluster enforces diversity among the candidate trajectories and leads to better performance. We compare these two search strategies extensively in Online Section D.2.

Algorithm 1 (Search-and-Bound Algorithm for Problem (4))

Data: Weighted graph \mathcal{G} with dynamic plastic density estimates $r^0, \{Q^t\}_{t \in \mathcal{T}}$;

- 1 **Initialize:** Compute r^t for all $t \in \mathcal{T}$ according to (2);
- 2 **Stage 1:** Run Algorithm A.1, obtain values $V^T(s)$ and solutions x^s for $s \in \mathcal{S}_T$;
- 3 **Stage 2:** Search-and-bound;
- 4 Initialize upper bound $UB = \max_{s \in \mathcal{S}_T} V^T(s)$;
- 5 Find $s^* \in \arg \max_{s \in \mathcal{S}_T} V^T(s)$ and define $x^* = x^{s^*}$;
- 6 Initialize lower bound $LB = \langle r_{|x^*}, x^* \rangle$, best solution $\hat{x} = x^*$;
- 7 Apply K -means clustering to construct the partition $\{s : V^T(s) \geq LB\} = \bigcup_{k=1}^K \mathcal{S}_T^{(k)}$;

```

8 for each region  $\mathcal{S}_T^{(k)}, k = 1, \dots, K$  do
9   Find  $s^{(k)} \in \arg \max_{s \in \mathcal{S}_T^{(k)}} V^T(s)$  and define  $x = x^{s^{(k)}}$ ;
10  (*) Compute the path-adjusted reward  $r_{|x}$ ;
11  if  $\langle r_{|x^s}, x^s \rangle > LB$  then
12    Update lower bound  $LB = \langle r_{|x^s}, x \rangle$ ;
13    Update best solution  $\hat{x} = x$ ;
14  end
15 end
16 return the solution  $\hat{x}$  and optimality gap  $(UB - LB)/UB$ .
    
```

Overall, Algorithm 1 uses the value of the relaxed problem (1) as an upper bound on the final value and efficiently searches for high-quality feasible solutions to obtain a lower bound, hence the term search and bound.

4.5. Tailored Branch and Bound

In this section, we propose a tailored branch-and-bound algorithm to iteratively refine our upper bound on the value of (4) and converge toward an optimal solution. Recall that, in the relaxation, we use the raw density maps r^t (path-independent) as edge lengths. This relaxation is valid because, for any admissible path x , $r_{|x^{0:t}}^{t+1} \leq r^{t+1}$ (Proposition 1). We now show how to refine this approximation when fixing the specific locations visited by the system and use this principle as the basis for our branching scheme.

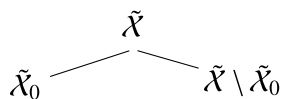
Let us consider a subset of trajectories $\tilde{\mathcal{X}} \subseteq \mathcal{X}$ such that trajectories within $\tilde{\mathcal{X}}$ must pass through k locations at k particular time points. Formally, we consider an integer k , times $t_1, \dots, t_k \in \mathcal{T}$, and locations $\ell_1, \dots, \ell_k \in \mathcal{L}$ and assume that $\tilde{\mathcal{X}} = \{x \in \mathcal{X} : x_{\ell_i}^{t_i} = 1, \forall i = 1, \dots, k\}$. We construct an upper bound on the path-dependent reward, \tilde{r} , as follows:

$$\tilde{r}^0 = r^0, \quad \tilde{r}^{t+1} = \begin{cases} Q^t(\tilde{r}^t - \alpha \tilde{r}^t \circ x^t) & \text{if } t \in \{t_1, \dots, t_k\}, \\ Q^t \tilde{r}^t & \text{otherwise.} \end{cases} \quad (6)$$

These new reward vectors are indeed tighter upper bounds on $r_{|x^{0:t}}^{t+1}$:

$$r_{|x^{0:t}}^{t+1} \leq \tilde{r}^{t+1} \leq r^{t+1}, \quad \forall t \geq 0, \quad \forall x \in \tilde{\mathcal{X}}. \quad (7)$$

We defer a formal proof of (7) to Online Section D.3. Because of this property, we can relax the path-dependent Problem (4) over the restricted feasible space $\tilde{\mathcal{X}}$ into a linear longest path optimization problem with edge lengths \tilde{r} , which is tighter than the original relaxation using r . We can apply Algorithm A.1 or 1 on this restricted problem to obtain an upper and lower bound. To refine this approximation, we pick a time $t_0 \in \mathcal{T}$ and a location $\ell_0 \in \mathcal{L}$ and partition $\tilde{\mathcal{X}}$ into



where $\tilde{\mathcal{X}}_0 := \{x \in \tilde{\mathcal{X}} : x_{\ell_0}^{t_0} = 1\}$. In other words, $\tilde{\mathcal{X}}_0$ fixes a new time/location for the admissible trajectories. Hence, we can construct a tighter upper approximation of the path-dependent rewards over $\tilde{\mathcal{X}}$ (the left child node) by applying (6) again. The subproblem $\tilde{\mathcal{X}}_0$ benefits from both a tighter approximation and a reduced search space, so we should expect to effectively reduce the optimality gap on this child node. For the right child node, however, the benefit only comes from reducing the search space from $\tilde{\mathcal{X}}$ to $\tilde{\mathcal{X}} \setminus \tilde{\mathcal{X}}_0$. Because of this imbalance, we expect this branching scheme to experience slow convergence toward an optimal solution, especially for large instances. Nonetheless, theoretically, this branching strategy eventually enumerates all possible trajectories and so is guaranteed to converge to an optimal solution after a finite yet exponential number of iterations. We evaluate its numerical behavior alongside Algorithm 1 in the following section. Details about the implementation of our branch and bound are provided in Online Section D.3.

4.6. Numerical Validation

In this section, we evaluate the numerical performance of Algorithm A.1, Algorithm 1, and tailored B&B. All numerical experiments are conducted on a desktop computer with an 11th-generation Intel i7 3.60 GHz CPU and 64 GB of memory, implemented in Python 3.9.16.

For Algorithm 1, we use $K = 12$ clusters. For the tailored branch and bound, we impose a limit on the maximum number of branches (50 or 150). We impose no gap or time limit.

First, we consider 250 instances of our seven-day routing problem (see Section 5.1 for a description of the weather and plastic ocean data), so $T = 8 \times 7 = 56$. We generate the instances by considering 50 different starting times (the beginning of each week except for the first and last weeks of the year) and five different initial locations. Table 1 reports the average lower bound, upper bound, optimality gap, and computational time achieved by each method on these instances.

Comparing the performance of Algorithms A.1 and 1, we observe that Algorithm 1 improves the quality of the best solution from 51.378 up to 52.763 (+2.7%), the returned upper bound being the same (as expected). This improvement translates into a reduction in the optimality gap returned by the algorithms from 9.2% (Algorithm A.1) down to 6.3% for Algorithm 1—a 2.9 percentage point or 32% improvement. The distribution of these gaps (see boxplots in Online Figure D.4) also shows that Algorithm 1 displays a lower variability in performance across instances, which is desirable. The additional computational time required (45 seconds instead of 14) is noticeable, yet affordable for seven-day planning problems.

Table 1. Average Performance of Algorithm A.1, Algorithm 1, and Our Tailored Branch and Bound on 250 Seven-Day Planning Instances

Metric	Algorithm A.1	Algorithm 1	Tailored B&B
Best solution (LB)	51.378	52.763	52.830
Best bound (UB)	56.236	56.236	54.475
Gap (with respective UB), %	9.172	6.323	2.918
Gap (with B&B UB), %	5.758	3.016	2.918
Time, s	14.15	45.60	951.52

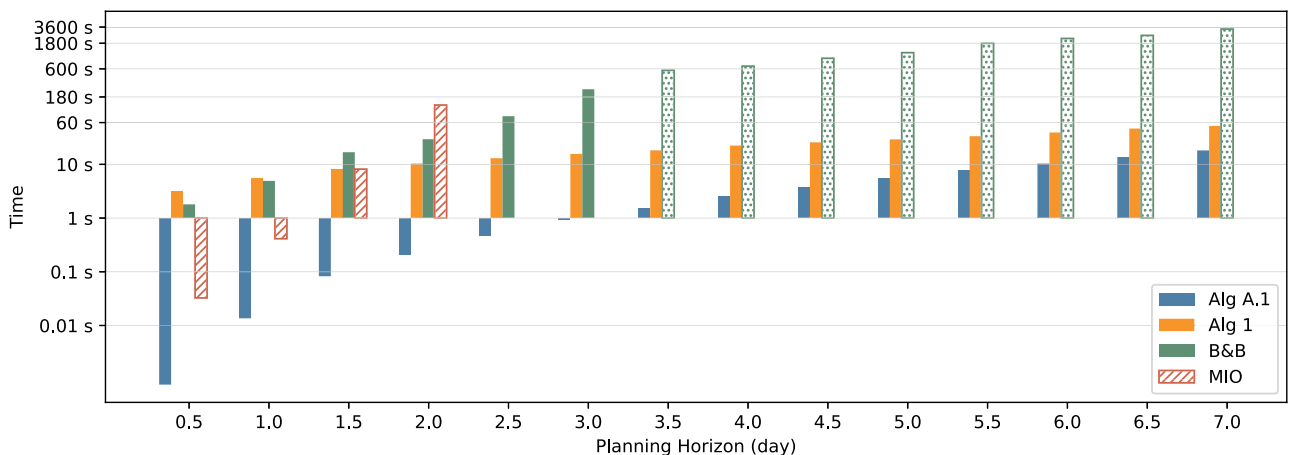
Note. We impose a limit on the maximum numbers of branches of 50 for tailored B&B.

Our tailored branch-and-bound algorithm, on the other hand, provides a much narrower improvement in terms of solution quality (+0.1%) and requires 15 minutes on average. Note that, at this scale, it terminates because of the limit on the maximum number of branches (50). For this reason, we do not view our tailored branch-and-bound algorithm as a practical alternative to Algorithm 1 for finding high-quality solutions; Algorithm 1 achieves the same quality and is 20 times faster. However, it returns a significantly tighter lower bound (54.475 on average, -3%). This suggests that the primary benefit of branching is to certify optimality. In particular, we can use this refined upper bound to tighten (and, in our case, halve) the suboptimality gap of the solutions obtained by Algorithms A.1 and 1, which suggests that the solution returned by Algorithm 1 could be much closer to optimality than indicated by the linear relaxation.

We investigate the validity of these observations as the length of the planning horizon, T , varies. Figure 6 reports the average computational time of each method for six instances and 14 different values of T . Among others, we observe that the relative difference in computational time between Algorithm A.1 and Algorithm 1

decreases with T , illustrating that the additional search stage does come at an exponential (in T) cost. Second, our tailored branch-and-bound algorithm solves to optimality instances with up to three-day horizons (beyond that, it terminates because of the limit on the number of branches). Nonetheless, it constitutes a significant improvement over solving a mixed-integer optimization (MIO) formulation of Problem (4) (see Online Section D.1) with commercial solvers (in our implementation: Gurobi 9.5.2 with Julia 1.9.3/JuMP for the interfacing). For two-day instances, MIO requires around four hours to create all the matrices Q^t (excluded from the time reported in Figure 6) and a couple of minutes for solving it, whereas our B&B does not need such preprocessing (whose time increases exponentially with T) and terminates in less than a minute, a two-order-of-magnitude improvement.

Similarly, Online Figure D.6 reports the average solution quality and gap for each method as the planning horizon increases. Among others, we observe that the solution returned by Algorithm 1 is actually optimal for instances with a planning horizon of less than three days (although the returned gap is strictly positive) and that the returned gap overestimates the true

Figure 6. (Color online) Average Computational Time (Log Scale) Returned by Algorithm A.1, Algorithm 1, and Our Tailored Branch-and-Bound Algorithm

Notes. We also report the time required by an off-the-shelf commercial MIO solver, excluding precomputation time required for creating the matrix Q^t 's (hashed bars). For B&B, the dotted bars indicate instances in which the maximum number of branches (150) triggered algorithm termination.

suboptimality of the solution by at least a factor of two on the larger instances.

5. Numerical Experiment

In this section, we evaluate our method on one-year weather and plastic density data. After presenting our experimental setting in Section 5.1, we compare the performance of different implementations of our algorithm in Section 5.2. In Section 5.3, we delve deeper into the differences in plastic collection efficiency across seasons. Finally, we use our algorithm to investigate the nonlinear relationships between some system design decisions (such as total span or size of the retention zone) on the overall efficiency in Section 5.4.

5.1. Experimental Setting and Implementation Choices

We work with one year of weather and plastic density data (year 2002 in our data set). The weather data provides the height and direction of the waves and the wind. The plastic density data are provided as trajectories of a particle-based dispersal model as described in Online Section C. Consistent with a span of 1.8 km, we consider a plastic collection rate $\alpha = 20\%$. The capacity of the retention zone is fixed to 25 metric tons. We assume that an extraction takes one day (eight time periods).

We divide the year into 13 nonoverlapping 28-day periods,² which we later refer to as simulations. For each simulation, we assume the system starts in the center of the GPGP, whose coordinates are (31.92°N, 142.4°W). We do not impose any restriction on the system's location at the end of each simulation (no depot location).

As a benchmark, we use The Ocean Cleanup's current heuristic, which we formally describe in Online Section E.1. In short, the benchmark picks the steering direction d leading to the highest distance-weighted reward over the next T time periods. The benchmark does not include any rule for extraction scheduling, so we start an extraction as soon as the system reaches capacity.

We evaluate the performance of four different implementations of Algorithm 1 with different optimization and implementation horizons (results for Algorithm A.1 are presented in Online Section E.2): first, we solve our longest path problem (4) for $T = 8$ time periods only (one day), solving and implementing the resulting solution every day. We refer to this implementation as "Myopic." To be more forward-looking, we consider using $T = 56$ time periods instead (one week): at the beginning of each week, we run Algorithm 1, obtain a solution, and implement it for the following seven days ("Week"). In these two variants, the planning horizon (used in the definition of the optimization problem)

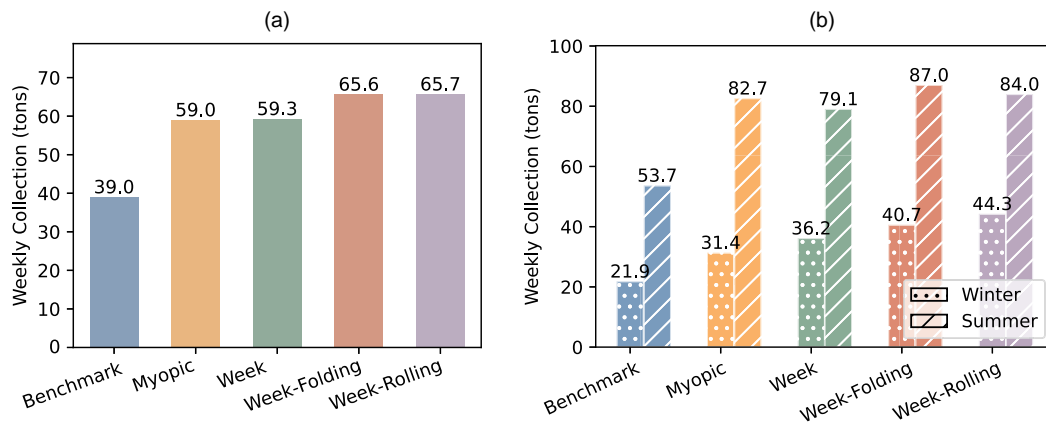
and the implementation period are the same. However, they do not have to be. For example, we can use a rolling horizon by solving each day a seven-day longest path problem (4) and implementing its solution for the first day ("Week-Rolling"). Alternatively, we can use a folding (or shrinking) horizon by finding the longest path until the end of the week (i.e., over seven days in the beginning, six days after the first day, etc.) and implementing its solution for the first day ("Week-Folding").

We should acknowledge that we choose a seven-day planning horizon as an illustration because it is simple, tractable, and captures most of the long-term dynamics of our system. However, experiments with varying planning horizon lengths (see Online Section E.4) suggest that it could be reduced to five days without much performance loss. Generally, we recommend a planning horizon long enough to include the next extraction, especially in winter (see discussion in Section 5.3). A longer planning horizon, on the other hand, would be unrealistic in our view because weather forecasts are not reliable beyond four to five days. Unfortunately, our data does not allow us to quantify more precisely the impact of forecast accuracy on performance at this stage. Resolving—every day for the Myopic, Week-Rolling, and Week-Folding implementations and every week for Week—allows us to update our plastic density estimates based on the past-day trajectory and potentially based on new weather forecasts. Hence, it can help mitigate the path-dependency issues described in Section 4 and improve robustness to forecasting errors in practice.

5.2. Overall Improvement in Plastic Collection Rate

Figure 7(a) represents the weekly quantity of plastic collected by each method, averaged over our 13 four-week simulations. First, Figure 7(a) illustrates the edge of optimization with all methods significantly improving over the benchmark—as identified formally by a paired t -test with $\leq 10^{-5}$ p -values in Online Table E.1. Among all methods, Week-Rolling collects the most plastic (65.7 tons/week), which is around 1.68 times more than the benchmark (39.0 tons/week). Among all optimization-based approaches, Myopic and Week perform the worst and comparably. This suggests that the benefit of being forward-looking of the Week implementation is outweighed by the path dependency issue (to which Week is more sensitive because it reoptimizes every week only). Accordingly, methods that reoptimize every day and consider a longer planning horizon, namely, Week-Folding and Week-Rolling, have a clear edge.

Figure 7(b) breaks down these yearly averages by season. We observe that the quantity of plastic collected (by any method) is higher in summer than in winter (53.3–87.0 range in summer versus 21.9–44.3 in

Figure 7. (Color online) Weekly Quantity of Plastic Collected for the Benchmark and Each Optimization-Based Approach

Notes. For the right panel, Winter = January–March (three first simulations) + October–December (three last simulations); Summer = April–September (seven simulations). (a) Results averaged over the 13 simulations. (b) Results averaged by seasons.

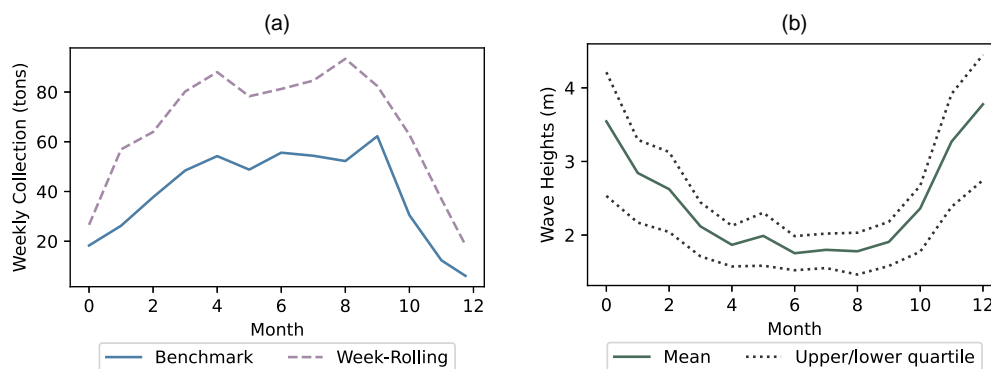
winter), and we observe that the relative benefit from our optimization methods (e.g., Week-Rolling) compared with the benchmark is higher during these winter months ($\times 1.6$ in summer versus $\times 2.0$ in winter). Similarly, the gain from using a seven-day (Week-Rolling) instead of one-day (Myopic) planning horizon is negligible in summer (84.0 versus 82.7: +1.6%) but most acute in winter (44.3 versus 31.4: +41%). We investigate the mechanisms driving this pattern in the coming section.

5.3. Heterogeneity Across Seasons and Impact of Extraction Scheduling

Figure 7(b) raises the question of the impact of season on the plastic collection efficiency. We should emphasize from the start that the behavior we observe is not driven by differences in the overall plastic density during the year. Values of plastic density continuously increase over the year (roughly by 10% in our 2002 data and by around 2% nowadays) but do not exhibit this inverted U-shape (see Online Figure E.4(a)).

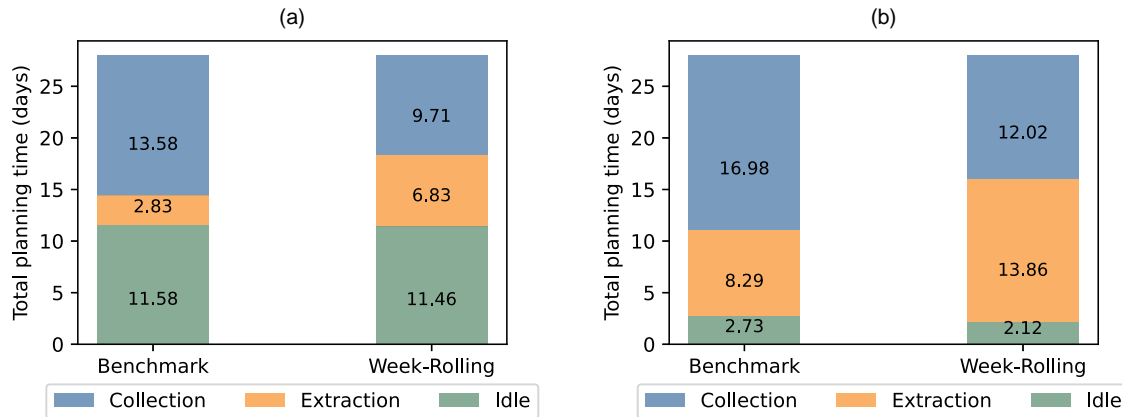
If plastic density (i.e., the objective of our optimization problem) cannot explain this behavior, it is natural to consider the impact of weather (which drives most of the operational constraints) on the heterogeneous performance across seasons. We observe in Figure 8 that wave height (right panel) follows the same pattern as the quantity of plastic collected (left panel) with lower waves experienced in the middle of the year (April–August) and higher waves during November–February. High waves affect the collection process in two ways. First, the system cannot operate when the wave height exceeds six meters. Hence, the average collectable plastic density is much lower in winter than in summer (see Online Figure E.4(b)). Furthermore, extractions require the waves to be below 2.5 meters for the first six hours and below 3.5 meters for 12 hours. Hence, weather and its impact on the feasibility of extractions most likely drive the behavior we observe.

To confirm this intuition, we quantify the time spent by the system when waiting to extract. At any point in time, the system can be in one of three phases: it can be

Figure 8. (Color online) Plastic Collected (Left Panel) and Average Wave Height in the GPGP (Right Panel) Across the Year 2002

Notes. (a) Plastic collected. (b) Wave height.

Figure 9. (Color online) Number of Days Spent on Collection, Extraction, and Idle per Four-Week Simulation for the Benchmark and Week-Rolling Methods



Notes. Results are aggregated over the winter (left panel) and summer (right panel) months. (a) Winter. (b) Summer.

actively collecting plastic, it can be undergoing an extraction, or it can be idle (i.e., unable to collect plastic because it has reached capacity but unable to start an extraction either because of weather). For each four-week simulation, we compute the number of days the system spent in collecting, extracting (which is equivalent to the number of extractions performed), and staying idle. For each simulation, the above three numbers should add up to 28 days. We averaged these numbers per season (winter/summer), for which we define winter as the first three and last three simulations and summer as the remaining seven ones. Figure 9 reports these metrics for the benchmark and Week-Rolling methods in both winter months (Figure 9(a)) and summer months (Figure 9(b)).

First, we observe that, in both winter and summer, Week-Rolling spends less time collecting than the benchmark. Given that Week-Rolling collects more plastic (67% more on average), this indicates that our approach is more efficient: it collects more in less time. Comparing Figure 9, (a) and (b), we observe that idle time is significantly higher in winter, confirming the fact that weather conditions limit the ability to extract (hence, to collect further) during winter. Surprisingly, Week-Rolling does not materially reduce total idle time in winter (around 12 days out of 28 for both methods). However, Week-Rolling performs twice as many extractions, around 6.8 times on average compared with 2.8 times for the benchmark (which aligns with the increase in quantity of plastic collected), so Week-Rolling experiences a much lower idle time per extraction than the benchmark.

The above observations highlight the importance of jointly finding a collection route and a schedule for the extractions for overall efficiency and the importance of considering a sufficiently long planning horizon. On this matter, our seven-day optimization approach that can explicitly account for weather-related constraints

experiences greater benefits in the winter when the ability to extract constitutes the main bottleneck.

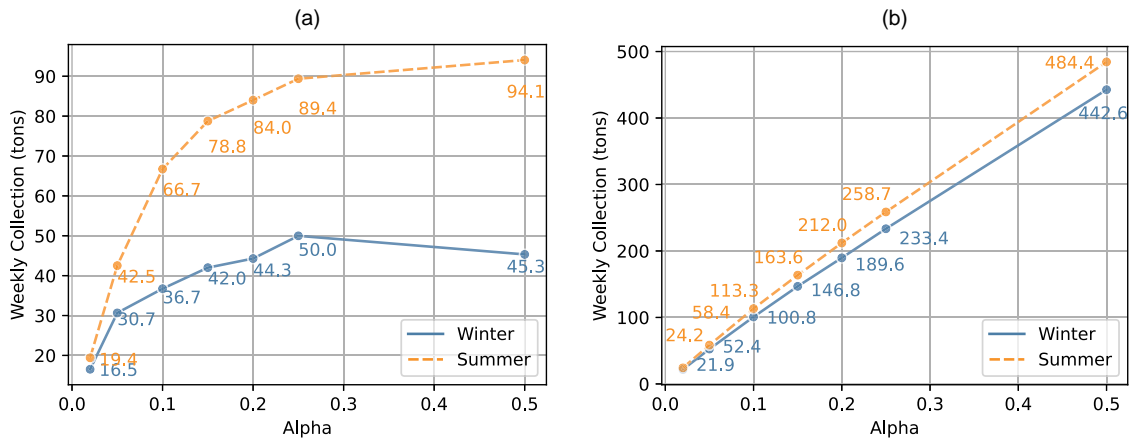
5.4. Designing a New System: Is Bigger Better?

In Section 5.2, we show that, with optimization, we can improve the collection speed from 40 tons/week to 60+ tons/week, using the current system. In this section, we use our optimization model to help answer strategic dimensioning decision for the next-generation system. The Ocean Cleanup was first considering increasing the span of the system from 0.6–0.8 km ($\alpha \approx 0.1$) to 1.6–1.8 km ($\alpha \approx 0.2$) without increasing the size of the retention zone (25 tons). Indeed, the size of the retention zone is partially constrained in practice by the size of the ship used to store and sort the plastic collected.

Figure 10 represents the weekly quantity of plastic collected in winter (solid blue lines) and summer (dashed orange lines) for increasing values of α in the case of a 25-ton capacity (left panel, Figure 10(a)) and an infinite capacity (right panel, Figure 10(b)). Without capacity constraints from the retention zone, one expects the total quantity of plastic collected to depend linearly in the span of the system α as displayed in Figure 10(b). However, with a finite capacity (Figure 10(a)), we observe (i) an overall lower quantity of plastic collected (which is because of the need to extract and the fact that we stop collection during extraction) and (ii) a strong concave dependency of the plastic collected on α . Indeed, by doubling the span size from $\alpha = 0.1$ to $\alpha = 0.2$, the weekly collection increases by 26% in summer (from 66.7 to 84.0 tons/week) and by 20% only in winter (from 36.7 to 44.3 tons/week). Moreover, increasing the span beyond $\alpha = 0.25$ (i.e., 2 km span) provides barely any improvement.

Intuitively, this different behavior across seasons is because a larger span requires more frequent extractions, which are very sensitive to weather conditions. The comparison of Figure 10, (a) and (b), highlights

Figure 10. (Color online) Weekly Quantity of Plastic Collected by Week-Rolling Under Different α



Notes. Results are aggregated over the winter (blue solid line) and summer (orange dashed line) months and with (left panel) or without (right panel) extraction. (a) System capacity 25 tons. (b) System infinite capacity (no extraction).

the impact of having a finite-capacity retention zone on the overall performance. In the future, the difficulty to extract could largely erode the benefit of having a larger system. This leads us to the next question: how to design a new system with better extraction?

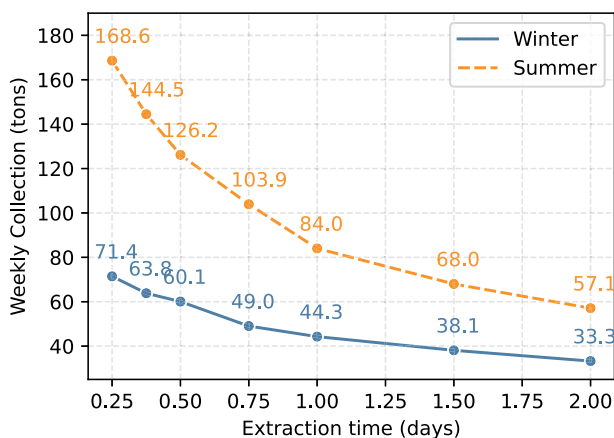
There are several ways to improve the current extraction process. One solution could be to reduce the unit extraction time, namely, the time spent per extraction. In practice, this could be achieved via more efficient extraction operations. For example, one could first empty the plastic from the retention zone on the deck (taking approximately six hours), put the system back into the water, and sort the plastic while resuming the plastic collection. Another solution could be to increase the total capacity of the retention zone, which we do not discuss in this paper.

Figure 11 represents the weekly quantity of plastic collected in winter (solid blue lines) and summer (dashed

orange lines) for increasing values of unit extraction time in the case of a 25-ton capacity. By reducing the time per extraction from one day (current practice) to 0.25 days (or six hours), the weekly collection increases by 101% in summer (84.0 to 168.6 tons/week) and by 61% in winter (44.3 to 71.4 tons/week). Observe, in comparison, that the potential improvement by further increasing the span size beyond $\alpha = 0.2$ (Figure 10(a)) is less than 12%. Reducing the unit extraction time demonstrates a greater potential for impact in both winter and summer.

We emphasize that the above improvement solely comes from a shorter extraction time, not from a lower impact of weather constraints because we kept, in our implementation, the same weather constraints for extraction (described in Online Section A.1) irrespective of the extraction time. In practice, a shorter extraction time might also translate into less stringent weather constraints, which could, in turn, provide additional benefits.

Figure 11. (Color online) Weekly Collections of Week-Rolling with Different Unit Extraction Times



Note. Results are aggregated over the winter (blue solid line) and summer (orange dashed line) months.

6. Conclusion

Our oceans are being threatened by the millions of tons of plastic that have been emitted over the recent decades. To limit future harm to marine ecosystems and activities, we need to clean up oceans from plastic as quickly as possible. To this end, we develop a graph-based model and formulate the problem of routing a plastic-collecting system in the GPGP to maximize the quantity of plastic encountered as a longest path problem. However, because of the plastic dynamics and the direct impact of collection on these dynamics, our resulting longest path problem (4) is nonconvex and nonseparable over edges. To deal with these computational difficulties, we propose to relax the reward dynamics and solve large-scale instances of this relaxation in linear time using a DP algorithm. Then, we obtain near-optimal solutions to our original problem, together with certificates of near optimality, by building an efficient

search algorithm based on geographical clustering and the terminal values of the DP algorithm (and not only on its optimal solution). We also develop a tailored branch-and-bound algorithm that solves instances with 24 time steps (three days) to optimality in minutes and instances with 56 time steps (seven days) within 4%–5% in an hour.

On one-year weather and plastic density data, we observe that our optimization approaches increase the quantity of plastic collected by 68% compared with the status quo, thus accelerating the path to plastic-free oceans. We also leverage our optimization algorithms to explore the nonlinear relationships between system characteristics and system performance. For example, because of difficulties to extract (i.e., empty the capacity of the system) in winter, we find that increasing the span of the system beyond 1.8 km will have barely any impact on collection efficiency in winter.

For our current application, the main concern and area for future research is to account for uncertainty in weather predictions and plastic dynamics. In particular, we are currently investigating whether collection of real-world data by drones or satellites could help quantify uncertainty and lead to robust versions of our longest path problem. More broadly, we are excited to study whether the class of longest path problems we identify in (4) could find other applications as a model for operations with nature dynamics.

Appendix. Dynamic Programming Algorithm for the Longest Path Problem

In this section, we provide the pseudo-code of an efficient DP algorithm for solving longest path problems given that our graph \mathcal{G} is a DAG. The algorithm is described in Algorithm A.1.

At each iteration, the algorithm finds the longest path ending in $s \in \mathcal{S}_t$ by connecting the longest path between \mathcal{S}_0 and s' and the edge (s', s) for some $s' \in \mathcal{S}_t$. The corresponding maximization problem can be solved by exhaustively searching through \mathcal{S}_t . Fortunately, in our graph \mathcal{G} , because of the no-sharp-angle constraint, we have $|\{s' \in \mathcal{S}_t : s \in \text{succ}(s')\}| \leq 3$, so this maximization problem can be solved in $\mathcal{O}(1)$ operations.

Algorithm A.1 (Dynamic Programming Algorithm for Finding the Longest Path)

Data: Weighted graph \mathcal{G} with weight $\{w_{s,s'}\}_{(s,s') \in \mathcal{E}}$

- 1 **Initialize** (values and optimal paths): $V^0(s) = 0$, $\text{path}[s] = \{s\}$, for all $s \in \mathcal{S}_0$;
- 2 **for** $t = 1:T$ **do**
- 3 **for** $s \in \mathcal{S}_t$ **do**
- 4 Find an optimal previous state,
 $s^* \in \arg \max_{s' \in \mathcal{S}_{t-1} : s \in \text{succ}(s')} \{w_{s',s} + V^{t-1}(s')\}$;
- 5 Update value function: $V^t(s) = w_{s^*,s} + V^{t-1}(s^*)$;
- 6 Update optimal path: $\text{path}[s] \leftarrow \text{path}[\text{path}[s^*], s]$;
- 7 **end**
- 8 **end**
- 9 Find an optimal terminal state $s^* \in \arg \max_{s \in \mathcal{S}_T} V^T(s)$;
- 10 **Return:** value $V^T(s^*)$, longest path $\text{path}[s^*]$.

Endnotes

¹ The dynamics in (3) implicitly assume that we can decompose the time interval $[t, t+1]$ into two distinct steps: a first step in which some of the plastic present at time t is removed and a second step in which the remaining plastics float according to the dynamics captured by Q^t . Of course, this is a simplification of reality in which these two steps occur concurrently. Yet we believe it is an appropriate model of reality, which captures the essence of path dependency.

² Because each four-week simulation requires five weeks of data (because some of our optimization algorithms are forward-looking and take into account information of the next seven days), we would need 53 weeks to conduct 13 nonoverlapping 28-day simulations. Instead, we start the last experiment one week earlier (on day 329 instead of 336).

References

- Appelgren LH (1971) Integer programming methods for a vessel scheduling problem. *Transportation Sci.* 5(1):64–78.
- Aydin N, Lee H, Mansouri SA (2017) Speed optimization and bunkering in liner shipping in the presence of uncertain service times and time windows at ports. *Eur. J. Oper. Res.* 259(1):143–154.
- Battarra M, Pessoa AA, Subramanian A, Uchoa E (2014) Exact algorithms for the traveling salesman problem with draft limits. *Eur. J. Oper. Res.* 235(1):115–128.
- Bellman R (1958) On a routing problem. *Quart. Appl. Math.* 16(1):87–90.
- Björklund A, Husfeldt T, Khanna S (2004) Approximating longest directed paths and cycles. Díaz J, Karhumäki J, Lepistö A, Sannella D, eds. *31st Internat. Colloquium Automata Languages Programming (ICALP 2004)*, vol. 3142 (Springer, Berlin, Heidelberg), 222–233.
- Bulterman RW, van der Sommen FW, Zwaan G, Verhoeff T, van Gasteren AJM, Feijen WHJ (2002) On computing a longest path in a tree. *Inform. Processing Lett.* 81(2):93–96.
- Chen C, Chen XQ, Ma F, Zeng X-J, Zeng XJ, Wang J (2019) A knowledge-free path planning approach for smart ships based on reinforcement learning. *Ocean Engng.* 189:106299.
- Cheng Y, Zhang W (2018) Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels. *Neurocomputing* 272:63–73.
- Christiansen M, Fagerholt K, Ronen D (2004) Ship routing and scheduling: Status and perspectives. *Transportation Sci.* 38(1):1–18.
- Church R, ReVelle C (1974) The maximal covering location problem. *Papers Regional Sci. Assoc.* 32(1):101–118.
- Cormen TH, Leiserson CE, Rivest RL, Stein C (2022) *Introduction to Algorithms* (MIT Press, Cambridge, MA).
- Danna E, Rothberg E, Pape CL (2005) Exploring relaxation induced neighborhoods to improve MIP solutions. *Math. Programming* 102:71–90.
- Dantzig GB (1960) On the shortest route through a network. *Management Sci.* 6(2):187–190.
- 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.
- de Wit C (1990) Proposal for low cost ocean weather routing. *J. Navigation* 43(3):428–439.
- Dijkstra EW (1959) A note on two problems in connexion with graphs. *Numerische Mathematik* 1(1):269–271.
- Dijkstra H, van Beukering P, Brouwer R (2021) In the business of dirty oceans: Overview of startups and entrepreneurs managing marine plastic. *Marine Pollution Bull.* 162:111880.
- Fagerholt K, Christiansen M (2000) A travelling salesman problem with allocation, time window and precedence constraints—An application to ship scheduling. *Internat. Trans. Oper. Res.* 7(3): 231–244.

- Ford LRJ (1956) Network flow theory. Technical report, Rand Corp., Santa Monica, CA.
- Gall SC, Thompson RC (2015) The impact of debris on marine life. *Marine Pollution Bull.* 92(1–2):170–179.
- Geyer R, Jambeck JR, Law KL (2017) Production, use, and fate of all plastics ever made. *Sci. Adv.* 3(7):e1700782.
- Granado I, Hernando L, Galparsoro I, Gabiña G, Groba C, Prellezo R, Fernandes JA (2021) Towards a framework for fishing route optimization decision support systems: Review of the state-of-the-art and challenges. *J. Cleaner Production* 320:128661.
- Groba C, Sartal A, Bergantiño G (2020) Optimization of tuna fishing logistic routes through information sharing policies: A game theory-based approach. *Marine Policy* 113:103795.
- Groba C, Sartal A, Vázquez XH (2015) Solving the dynamic traveling salesman problem using a genetic algorithm with trajectory prediction: An application to fish aggregating devices. *Comput. Oper. Res.* 56:22–32.
- Groba C, Sartal A, Vázquez XH (2018) Integrating forecasting in metaheuristic methods to solve dynamic routing problems: Evidence from the logistic processes of tuna vessels. *Engrg. Appl. Artificial Intelligence* 76:55–66.
- Hagiwara H, Spaans JA (1987) Practical weather routing of sail-assisted motor vessels. *J. Navigation* 40(1):96–119.
- Ioannidou K, Nikolopoulos SD (2013) The longest path problem is polynomial on cocomparability graphs. *Algorithmica* 65(1):177–205.
- James RW (1957) *Application of Wave Forecasts to Marine Navigation* (U.S. Naval Oceanographic Office, Washington, DC).
- Kaandorp ML, Lobelle D, Kehl C, Dijkstra HA, van Sebille E (2023) Global mass of buoyant marine plastics dominated by large long-lived debris. *Nature Geoscience* 16(8):689–694.
- Karger D, Motwani R, Ramkumar GDS (1997) On approximating the longest path in a graph. *Algorithmica* 18(1):82–98.
- Karp RM (2010) *Reducibility Among Combinatorial Problems* (Springer, Berlin, Heidelberg).
- Klink D, Peytavin A, Lebreton L (2022) Size dependent transport of floating plastics modeled in the global ocean. *Frontiers Marine Sci.* 9:903134.
- Kosmas OT, Vlachos D (2012) Simulated annealing for optimal ship routing. *Comput. Oper. Res.* 39(3):576–581.
- Langbein J, Stelzer R, Frühwirth T (2011) A rule-based approach to long-term routing for autonomous sailboats. Schlaefer A, Blaurcock O, eds. *Robotic Sailing Proc. Fourth Internat. Robotic Sailing Conf.* (Springer, Berlin, Heidelberg), 195–204.
- Lebreton L, Van Der Zwet J, Damsteeg JW, Slat B, Andrady AL, Reisser J (2017) River plastic emissions to the world's oceans. *Nature Comm.* 8(1):15611.
- Lebreton L, Slat B, Ferrari FF, Sainte-Rose B, Aitken J, Marthouse R, Hajbane S, et al. (2018) Evidence that the Great Pacific Garbage Patch is rapidly accumulating plastic. *Sci. Rep.* 8(1):4666.
- Li WC, Tse H, Fok L (2016) Plastic waste in the marine environment: A review of sources, occurrence and effects. *Sci. Total Environ.* 566:333–349.
- Madraki G, Judd RP (2019) Recalculating the length of the longest path in perturbed directed acyclic graph. *IFAC-PapersOnLine* 52(13):1560–1565.
- Malaguti E, Martello S, Santini A (2018) The traveling salesman problem with pickups, deliveries, and draft limits. *Omega* 74:50–58.
- Meijer LJ, 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. *Sci. Adv.* 7(18):eaaz5803.
- Meng Q, Wang T (2011) A scenario-based dynamic programming model for multi-period liner ship fleet planning. *Transportation Res. Part E Logist. Transportation Rev.* 47(4):401–413.
- Moore EF (1959) The shortest path through a maze. Aiken HH, ed. *Proc. Internat. Sympos. Theory Switching*, Part II (Harvard University Press, Cambridge, MA), 285–292.
- Pandit SN (1962) Some observations on the routing problem. *Oper. Res.* 10(5):726–727.
- Park YJ, Garaba SP, Sainte-Rose B (2021) Detecting the Great Pacific Garbage Patch floating plastic litter using worldview-3 satellite imagery. *Optics Express* 29(22):35288–35298.
- Park YJ, Garaba S, Sainte-Rose B, Han HJ (2022) Satellite remote sensing of marine litter floating in open ocean and coastal waters. *Living Planet Sympos.* (The European Space Agency, Paris).
- Peytavin A, Sainte-Rose B, Forget G, Campin JM (2021) Ocean plastic assimilator v0.2: Assimilation of plastic concentration data into Lagrangian dispersion models. *Geoscientific Model Development* 14(7):4769–4780.
- Pollack M, Wiebenson W (1960) Solutions of the shortest-route problem—A review. *Oper. Res.* 8(2):224–230.
- Robinson NM, Nelson WA, Costello MJ, Sutherland JE, Lundquist CJ (2017) A systematic review of marine-based species distribution models (SDMs) with recommendations for best practice. *Frontiers Marine Sci.* 4:421.
- Sainte-Rose B, Rakotonirina AD, van den Bremer T, Pham Y (2022) Numerical simulation of the wave-induced drift of floating marine plastic debris modeled as discrete particles. Ehlers S, ed. *Internat. Conf. Offshore Mech. Arctic Engrg.*, vol. 85925 (The American Society of Mechanical Engineers, New York), V007T08A008.
- Schmidt C, Krauth T, Wagner S (2017) Export of plastic debris by rivers into the sea. *Environ. Sci. Tech.* 51(21):12246–12253.
- Schnurr RE, Alboiu V, Chaudhary M, Corbett RA, Quanz ME, Sankar K, Srain HS, Thavarajah V, Xanthos D, Walker TR (2018) Reducing marine pollution from single-use plastics (SUPs): A review. *Marine Pollution Bull.* 137:157–171.
- Schrijver A (2012) On the history of the shortest path problem. *Documenta Mathematica* 17(1):155–167.
- Sen D, Padhy CP (2015) An approach for development of a ship routing algorithm for application in the North Indian Ocean region. *Appl. Ocean Res.* 50:173–191.
- Shao W, Zhou P, Thong SK (2012) Development of a novel forward dynamic programming method for weather routing. *J. Marine Sci. Tech.* 17(2):239–251.
- Shimbel A (1955) Structure in communication nets. Fox J, ed. *Proc. Sympos. Inform. Networks* (Polytechnic Institute of Brooklyn, New York), 119–203.
- Skoglund L (2012) A new method for robust route optimization in ensemble weather forecasts. Unpublished master's thesis, KTH School of Engineering Sciences, Stockholm, Sweden.
- Sniedovich M (2006) Dijkstra's algorithm revisited: The dynamic programming connexion. *Control Cybernetics* 35(3):599–620.
- Stalhane M, Hvattum LM, Skaar V (2015) Optimization of routing and scheduling of vessels to perform maintenance at offshore wind farms. *Energy Procedia* 80:92–99.
- Stokes GG (1847) On the theory of oscillatory waves. *Trans. Cambridge Philos. Soc.* 8:441–455.
- Takashima K, Mezaoui B, Shoji R (2009) On the fuel saving operation for coastal merchant ships using weather routing. *TransNav Internat. J. Marine Navigation Safety Sea Transportation* 3(4):401–406.
- Taylor GI (1915) I. Eddy motion in the atmosphere. *Philos. Trans. Roy. Soc. London Ser. A Containing Papers Math. Phys. Character* 215(523–537):1–26.
- Ting SC, Tzeng GH (2003) Ship scheduling and cost analysis for route planning in liner shipping. *Maritime Econom. Logist.* 5(4):378–392.
- Uehara R, Uno Y (2007) On computing longest paths in small graph classes. *Internat. J. Foundations Comput. Sci.* 18(05):911–930.
- Vettor R, Tadros M, Ventura M, Guedes Soares C (2016) Route planning of a fishing vessel in coastal waters with fuel consumption restraint. *Maritime Tech. Engrg.* 3:167–173.
- Wilcox C, Van Sebille E, Hardesty BD (2015) Threat of plastic pollution to seabirds is global, pervasive, and increasing. *Proc. Natl. Acad. Sci. USA* 112(38):11899–11904.

- Winterstetter A, Grodent M, Kini V, Ragaert K, Vrancken KC (2021) A review of technological solutions to prevent or reduce marine plastic litter in developing countries. *Sustainability* 13(9):4894.
- Yoon H, Nguyen V, Nguyen T (2018) Development of solution for safe ship considering seakeeping performance. *TransNav Internat. J. Marine Navigation Safety Sea Transportation* 12(3): 517–525.
- Zheng J, Zhang H, Yin L, Liang Y, Wang B, Li Z, Song X, Zhang Y (2019) A voyage with minimal fuel consumption for cruise ships. *J. Cleaner Production* 215:144–153.
- Zis T, Psarftis HN, Ding L (2020) Ship weather routing: A taxonomy and survey. *Ocean Engrg.* 213:107697.
- Zoppoli R (1972) Minimum-time routing as an N-stage decision process. *J. Appl. Meteorology (1962-1982)* 11(3):429–435.

Dick den Hertog is a professor of operations research at the University of Amsterdam and an expert in robust optimization. He is also active in applying optimization for a better society. He is the

cofounder and science-to-impact codirector of Analytics for a Better World and received the INFORMS Franz Edelman Award in 2013 and 2021.

Jean Pauphilet is an assistant professor of management science and operations at London Business School. His research sits at the intersection of discrete optimization, robust optimization, and machine learning with applications to healthcare and sustainable operations. His work has been recognized by multiple awards, including the 2020 INFORMS George E. Nicholson and the 2021 INFORMS Pierskalla awards.

Yannick Pham is a hydrodynamic engineer by training and the steering strategy manager at The Ocean Cleanup.

Bruno Sainte-Rose is the lead computational modeler at The Ocean Cleanup. His work combines Lagrangian dispersal modeling, computational fluid dynamics, and oceanography to model the field efficiency of plastic cleanup systems.

Baizhi Song is a PhD candidate in management science and operations at London Business School. His research interests involve data-driven decision making and its applications to sustainable operations.