Team Sonic: Andrew Fryer • Jamie Won • Kyle Singer • Tristan Lawson

Waste Map-Tracker

OVERVIEW

Every year, 8 million tons of plastic end up in the oceans. Some of this loss occurs during the transportation of waste from sites to sorting and recycling facilities. We designed an algorithm that selects the optimal path for plastics at the end of their life cycle. The optimal path minimizes the risk of plastics ending in the ocean, and the distance traveled to move the plastics.

STAKEHOLDERS

This problem affects everyone in different ways. Here are three important stakeholder groups:

Residents	Fishing Industry	Waste Management
Residents near oceans can be harmed by plastics. The waste can damage their land and negatively impact their health.	Plastics harm fish, reducing the quality of catch. They also interfere with or damage fishing equipment.	These organizations have a strong influence on the issue. It also affects their reputation and key performance indicators.

SOLUTION

We designed a web app that provides an efficient path for waste management. We broke the problem into two parts: first, we determine the waste path that minimizes the QoR for each waste site. Next we determined the optimal transportation path for a single vehicle to move the waste between facilities. This breakdown makes the algorithm scalable and adaptable. The transportation algorithm can be updated for a new fleet without changing the waste path for each waste site.

Target Users

This solution is designed to help plan waste dispersion and transportation routes. As such, it is intended for organizations that perform waste management. They will be able to optimize their practices using this algorithm, and easily operate our product with the GUI.

Waste Path

We designed a greedy algorithm to find the optimal waste path. For each waste source the algorithm selects the local sorting facility that minimizes the QoR. This optimizes the waste path based on variables a and b. The same is done to select the path from local sorting facilities to regional sorting facilities, and then regional recycling facilities.

Transportation Path

The transportation path begins from the starting node of the supplied data sheet (per the requirements) which is a waste location and precedes to retrieve waste from all other waste locations that have a corresponding optimal local sorting facility. Once all of the waste has been placed into their respective local sorting facilities, the algorithm begins at an arbitrary facility and collects the sorted waste from the different facilities that correspond to a regional sorting facility (similar to the previous step). Once all of the waste is in their respective regional sorting facilities, the wasted can then be moved on to recycling stations where each of the sorted stations that correspond to a recycled station are collected from before dropping off at the recycled station.

Graphical User Interface

The GUI has three main components: an interactive interface, a map displaying sites and routes, and a results page. A user can select a file and enter algorithm parameters, and view the results on a colour-coded interactive map. Finally, the results of the algorithm can be found at the bottom of the page.

Tools Used

We used Flask to design the connections between the algorithm and the user interface. We used pandas to construct our algorithm. We ran the backend on a flask server and proxied requests made from a React app to it. Material-ui and and react-globe was used for styling and the map respectively.

NEXT STEPS

Improvements to the Algorithm

We currently construct our transportation path by visiting each child node (upstream) before visiting the parent node, visiting the child closest to the parent last. In the future we would like to implement Djikstras algorithm to further optimize the path.

Improvements to the Graphical User Interface

The following improvements could be made to the graphical user interface: error handling, consistent design and colouring, a feature that allows users to upload csv files for the algorithm to run on, and loader components for better usability.

HOW TO USE

Please follow the instructions found here