ARTICLES

Interactive Multi-Objective Path Planning Algorithm for Accessible Sidewalks

Alisa Todorova

Aug 29, 2023 • 8 min read

3D City Model | Accessibility | +3 others

Suppose you want to go to the Rijksmuseum on foot. Navigation apps determine the shortest path from your location to the Rijksmuseum based on one objective (i.e., criterion), which is most often the length of the path. However, considering only the length of the path is often not sufficient for people with mobility issues and those who use assistive devices, such as wheelchairs, strollers, and canes. They have different and many needs with respect to the accessibility of sidewalks. For example, they might prefer to take a longer path which has fewer crossings than a shorter path with more crossings. They might even prefer to take only those sidewalks which have a certain width. Hence, this accessibility, and especially the lack thereof, of the paths results in additional and different objectives. This research is part of the broader project *Amsterdam For All* [1], which aims to help Amsterdam citizens with mobility issues to find the most accessible sidewalks, with respect to their specific needs. To cater to different people, this will be done through a highly user-dependent app. The focus of this research is on creating an interactive algorithm which can output only those paths the users are likely to prefer to take, given their specific accessibility needs, from and to their desired destinations.

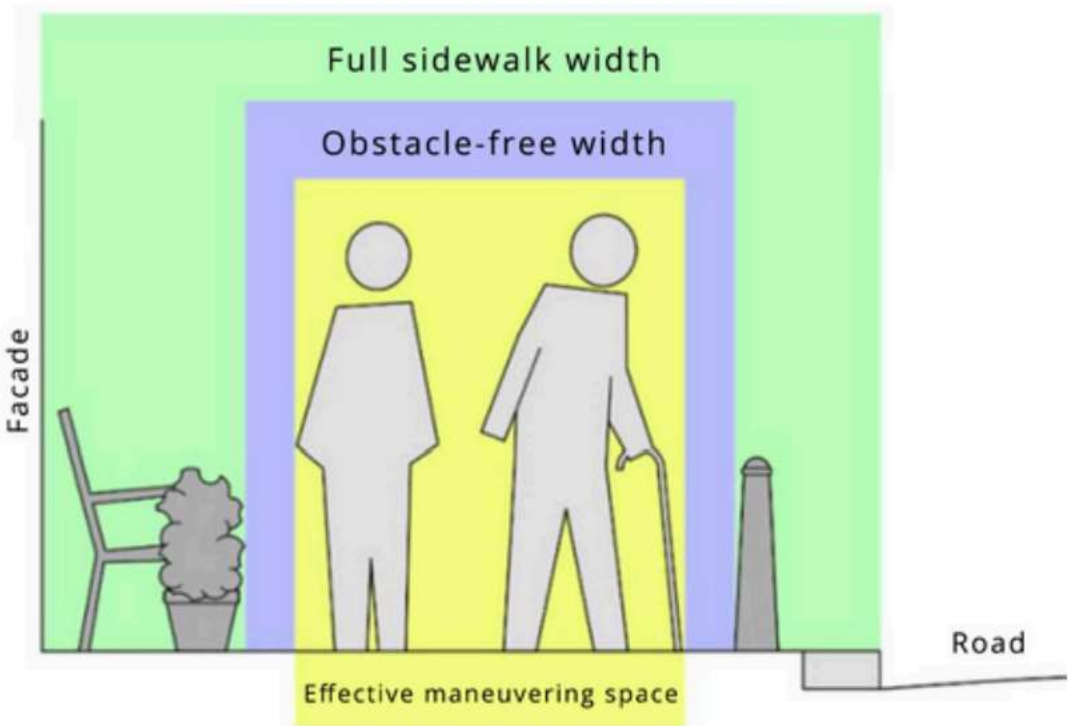

Fig. 1: Image with full sidewalk width in green and obstacle-free width in purple [5].

Our Approach

When we consider the optimization of multiple objectives (or criterions) at the same time, we work with a multi-objective shortest path problem (MOSPP). For this research, we limit the problem to two objectives: the length of the path in meters and the number of crossings. In practice, the objectives might be potentially conflicting, as one path might perform well in one objective, but worse off in another. Hence, there is not a single, feasible path that can optimize all objectives at the same time [8]. Instead, the goal of MOSPP is to find the Pareto set. This is a set of all possible solutions that have different optimal trade-offs (i.e., best compromises) between the objectives. These solutions are called Pareto-optimal paths. Pareto-optimality is achieved when it is no longer possible to improve one objective without making some of the others worse [7].

Furthermore, for this research, we did not work directly with real users, but instead we used an interactive approach method to obtain the (simulated) user preferences. Our work builds on “Ordered preference elicitation strategies for supporting multi-objective decision making” [4], as their work has proven effective for preference elicitation and minimizing the number of queries that need to be posted to the user. That is why we used their Gaussian process (GP) as a blackbox in our approach. The GP is a utility function model that models the user's preference. It compares the paths in terms of their values, and picks the one which has the maximum user utility, because the most preferred path in an MOSPP is the path which maximizes the user utility. In that sense, the GP acts as a surrogate method for user ranking of paths. This allows us to construct an algorithm which finds a Pareto-optimal path (in terms of trade-offs between the objectives) for each individual user, without having to compute the entire Pareto set.

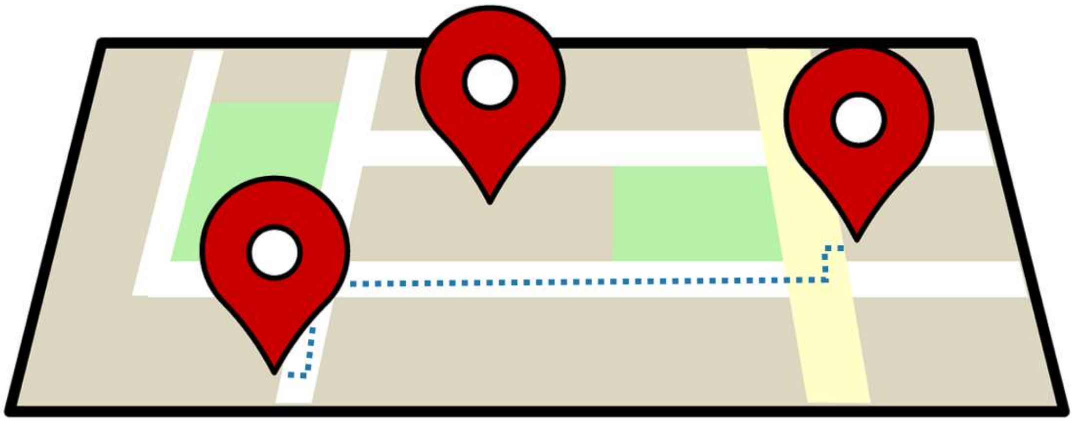

Fig. 2: A map illustrating a path (blue dots) from one location (red pin) to another.

Methods

In our scenario, we can represent the locations on the map as nodes in a graph, and the sidewalks and the crossings as edges connecting these nodes.

Our proposed algorithm for finding a Pareto-optimal path for each individual user without having to compute the entire Pareto set is split into two sub-algorithms: an outer-loop, which helps us select a target region (i.e., a part of value space that is likely to contain possible value vectors (and their associated paths) preferred by the user), and an inner-loop, which finds a path from that target region.

A key observation that we make in our research is that finding the closest path to a given target (i.e., a desired value) in a **MOSPP is NP-hard**. The way we attempt to heuristically solve our NP-hard problem is by using a depth-first search algorithm (DFS) in our inner-loop. DFS begins its exploration from the starting node *S*, goes as deep as possible along all branches, and then backtracks [6]. Our DFS algorithm does not guarantee to output a Pareto-optimal path let alone the Pareto-optimal path which is closest to the given target, but if ran until convergence, it can return a Pareto-optimal path which lies in the target region.

Results

For the experiments, we worked with a map of Amsterdam, which is a square with a radius of 800m, centered around the Rijksmuseum. We refer to it in this paper as the *full map*. When we transform the *full map* into a graph, we have 11401 nodes. For some of the experiments, we scaled down this map to an area of 250m by 250m, still centered around the Rijksmuseum. We refer to this as the *small map*. When we transform this *small map* into a graph, we have 1006 nodes.

We did 100 experiments on the *full map*, and 50 experiments on the *small map*, with different *S* (i.e., starting node or location) and *T* (i.e., terminating/ending node or location). In order to evaluate the accuracy of our proposed algorithm, we implemented a baseline which computes the full Pareto set. The implementation of our baseline is multi-objective value iteration, for which we follow Theorem 2 from [3] and Pareto value iteration from [2].

As expected, our proposed algorithm is computationally faster than the baseline. The baseline takes approximately 6 hours and

39 minutes to compute the full Pareto set of the *full map*. On the other hand, our proposed algorithm takes 18 seconds on average to compute a path. On the *small map*, the baseline takes 43.63 seconds, while our proposed algorithm takes 0.25 seconds on average. We also note that the regret is 0.07. Therefore, we lose approximately 7% of the utility in order to gain orders of magnitude in the computation time.

As we anticipated, our algorithm misses quite a lot of Pareto-optimal paths. It is important to note that our proposed algorithm does not guarantee to output a Pareto-optimal path nor the Pareto-optimal path, which is closest to the given target. Nevertheless, from all experiments, both on the full and the *small map*, we observed that our proposed algorithm always returns one Pareto-optimal path. In our visualizations, we call this path p^* and color it in green. We observed that path p^* is always more optimal in length rather than in the number of crossings, i.e., path p^* minimizes the length while compromising the number of crossings. For examples of such paths, refer to Figures 2 and 3 below.

Fig. 3: *Full map*. Distance between *S* and *T* : 2.84 km.; Cost of p*: (2841.71, 37.); Cost of alternative path: (3451.74, 21.)

Fig. 4: *Small map*. Distance between *S* and *T* : 0.76 km.; Cost of p*: (757.28, 4.); Cost of alternative path: (760.05, 4.)

To see whether our algorithm can find a Pareto-optimal path that can improve upon the number of crossings, our proposed algorithm also tries to output an alternative path. We color this path in orange in our visualizations. From almost all experiments, both on the full and the *small map*, we observed that the alternative path is a non-Pareto optimal path. In some cases, however, the alternative path is also a Pareto-optimal path. For example of such cases, refer to Figures 5 and 6 below. We assume that perhaps the location of *S* and *T* has some influence on whether the alternative path is also a Pareto-optimal path. The sidewalks of the area around the Rijksmuseum have a grid-like pattern. As seen in Figure 6, for example, if we want fewer crossings, then the path becomes longer, and vice versa.

Fig. 5: *Full map*. Distance between S and T : 1.41 km.; Cost of p*: (1414.67, 15.); Cost of alternative path: (1437.68, 13.)

Fig. 6: *Small map*. Distance between *S* and *T* : 0.12 km.; Cost of p*: (119.52, 3.); Cost of alternative path: (148.14, 1.)

To see when our algorithm starts finding non-Pareto optimal paths, we tested empirically by decreasing the runtime of our DFS algorithm. We examined that, on average, half of the runtime of the DFS algorithm is used to find a path, while the other half is used to check whether this path is indeed a Pareto-optimal path.

Conclusion and Future Work

We proposed an algorithm which finds a Pareto-optimal path for each individual user without having to compute the entire Pareto set. Our algorithm does not guarantee to output a Pareto-optimal path nor the Pareto-optimal path, which is closest to the given target. Nevertheless, from all experiments, we observed that our algorithm always returned one Pareto-optimal path. Furthermore, compared to computing the full Pareto set, our algorithm is very computationally quick and does not lose much utility. We conclude that our algorithm is a good heuristic solution to the NP-hard problem of finding the closest path to a given target for our given MOSPP.

We anticipate future work to entail extending this research and algorithm to work with all objectives of the broader project [1], as well as with real users. We also note that the *full map* we worked with is still a relatively small area of Amsterdam. That is why we propose for further investigation in future work how our algorithm will perform on a bigger map - for example, on the whole city of Amsterdam.

Note: This research was done as part of a **Bachelor thesis internship project.**

References

[1] Amsterdam for all. https://www.amsterdamintelligence.com/projects/amsterdam-for-all. Accessed: 2023-04-28

[2] Ann Nowé, Roxana Radulescu Diederik M Roijers, Willem Röpke. On following Pareto-optimal policies in multi-objective planning and reinforcement learning. *Proceedings of the Multi-Objective Decision Making (MODeM) Workshop*, 2021.

[3] D.J White. Multi-objective infinite-horizon discounted Markov decision processes. *Journal of Mathematical Analysis and Applications*, 89(2):639–647, 1982

[4] Luisa M. Zintgraf, Diederik M. Roijers, Sjoerd Linders, Catholijn M. Jonker, and Ann Nowé. Ordered preference elicitation strategies for supporting multi-objective decision making. AAMAS '18, page 1477–1485, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

[5] Pinhão, Cláudia & Eijgenstein, Chris & Gornishka, Iva & Jansen, Shayla & Roijers, Diederik & Bloembergen, Daan. (2022). Determining Accessible Sidewalk Width by Extracting Obstacle Information from Point Clouds. 10.48550/arXiv.2211.04108.

[6] S.J. Russell, P. Norvig, and E. Davis. *Artificial Intelligence: A Modern Approach*. Prentice Hall series in artificial intelligence. Prentice Hall, 2010.

[7] V. Pareto, A. Montesano, A. Zanni, L. Bruni, J.S. Chipman, and M. McLure. *Manual of Political Economy: A Critical and Variorum Edition*. OUP Oxford, 2014.

[8] Zhongqiang Ren, Richard Zhan, Sivakumar Rathinam, Maxim Likhachev, and Howie Choset. Enhanced multi-objective A* using balanced binary search trees. *ArXiv*, 2022.

**Royalty-free header from Pixabay*

3D City Model | Accessibility | Interactive Algorithm | Multi-Objective Path Planning | Sidewalk Accessibility

Alisa Todorova

I'm a Data Science and AI Bachelor student at Maastricht University and an AI intern at the municipality. I'm creating an interactive multi-objective path planning algorithm for accessible sidewalks.

Blog Posts | Code | Datasets | Publications

ARTICLES

From Obstacles to Opportunities:...

Aditi Mishra • Oct 2, 2023

ARTICLES

Computer Vision for European Urban...

Kasper Verhavert • Sep 19, 2023

ARTICLES

Automatically Extracting Venue...

Iva Gornishka • Jan 24, 2023

INTERVIEWS

An Interview with Jorrit Ypenga on...

Jason Fiawo • Jun 9, 2022

Subscribe to our newsletter

Artificial Intelligence related articles, project stories, interviews and videos delivered to your inbox. Unsubscribe anytime.

Email address *

e.g. myemail@amsterdam.nl

Subscribe

By providing us with your email, you agree to the terms of our our **Privacy Policy**

intelligence

Useful Links

AI not only stands for Artificial Intelligence but also for “Amsterdam Intelligence”. With this website we wish to share our knowledge, process and results. Check our blog posts and projects to keep up to date. Feel free to get in touch.

Resources
Privacy Policy
Projects
About
Contact

Latest Posts

Part 2: RAG for open.amsterdam - Results and ethical considerations

Part 1: RAG for open.amsterdam - Technical Implementation

Illumination Modeling of Street Lights in Urban Point Clouds