Proceedings of the Twenty-Seventh Conference on Innovative Applications of Artificial Intelligence

Capturing Human Route Preferences from Track Information: New Results

Johnathan Gohde * **Mark Boddy** **Hazel Shackleton** **Steve Johnston**
Adventium Labs
111 3rd Ave S, Suite 100, Minneapolis MN 55401 USA
{firstname.lastname@adventiumlabs.com}

Abstract

In previous work, we described G2I2, a system that adjusts the cost function used by an off-road route planning system in order to more closely mimic the route choices made by humans. In this paper, we report on an extension to G2I2, called GUIDE, which adds significant new capabilities. GUIDE has the ability to induce a cost function starting with a set of historical tracks used as training input, with no requirement that these tracks be even close to cost-optimal. Given a cost function, either induced as above or provided from elsewhere, GUIDE can then compare planned routes with the actual tracks executed to adjust that cost function as either the environment or human preferences change over time. The features used by GUIDE in both the initial induction of the cost function and subsequent tuning include time-varying meta-data such as the temperature and precipitation at the time a given track was executed. We present results showing that, even when presented with tracks that are very far from cost-optimal, GUIDE can learn a set of preferences that closely mimics terrain choices made by humans.

1 Introduction

For the past several years, we have been working to improve map-based route planners for human route traversal. The applications of interest include off-road terrain, so our maps are grids of pixels, rather than more general graphs.

Our objective is to improve the performance of these route planners in several ways. The planners should be *tolerant* of errors in the maps used in planning, *adaptable* to reduce mismatches between (possibly hand-coded) cost functions and actual human preferences, *evolvable* over time as the map, mission, or human preferences change. The planning decisions made by these systems should be *explainable* in terms that make sense to the user of such a system, not just the implementer. They should be capable of *integrating* multiple data sources, both sources of data regarding the area in which the route is to be planned, and alternative dimensions of preference or cost for a given route. Finally, these systems should perform these marvels largely *independent* of explicit human input. For example, users should not be asked to explain why their choices in executing a route differed from the route planned by the system.

In this paper, we describe our progress to date in addressing the requirements of this application, currently implemented in a system called GUIDE. In Section 2 we present this motivating application in more detail, showing how problem features have guided our design decisions. We also summarize our previous work on a predecessor system called G2I2 (Gohde, Boddy, and Shackleton 2013), and describe new capabilities and improvements captured in GUIDE. Subsequent sections present the two different forms of learning performed by GUIDE, induction of an initial cost function based on a training set of historical tracks (Section 3), and tuning of that cost function based on pairwise differences between planned routes and the tracks as executed, based on those routes (Section 4). We then present some experiments demonstrating the current performance of the system (Section 5), discuss some relevant previous work by others in this area (Section 6), and conclude with a discussion of our results and some future work, in Section 7.

2 Off-Road Route Planning for Human Execution

As discussed above, we are primarily concerned with *improving the behavior* of off-road route-planning systems *from the point of view of the users of those systems*. We are interested in systems that generate novel routes between specified start and end points, rather than systems that work by retrieving and combining previously-executed segments, and so focus on planning systems doing some form of cost-based heuristic search. There are several axes along which these systems might be improved, all related in one way or another to the model of movement cost used by the planner. There may be errors in the underlying map, for example roads that should be present that are not represented. The cost function(s) relating map features to human preferences may not model those preferences sufficiently closely. There may be features not appearing in the maps at all, which affect human route choices.

Further complicating the problem is the fact that preferences, the relevant features, and the map itself may all change over time. Personnel may rotate in and out of a particular organization. Seasonal changes in ground cover

*This work was supported by DARPA, under Contract # D11PC20177

and soil moisture can drastically affect preferences for areas to traverse. And the map may change through the addition or removal of map features. For example, Figure 1 shows a set of GPS tracks in Olathe, KS, gathered over a period of several months. On the left, there is a straight, vertical track through the center of the map, showing the presence of tracks that took that route. The right shows the same area and the same data, with tracks filtered to exclude those before a specified date. In this figure, the vertical feature is missing. The explanation is visible in the satellite image on which the tracks are overlaid: there is a curving road through the area in question, which was only recently completed. Previously, the road ran straight north and south.

Figure 1: Left: Older GPS tracks along a straight road. Right: Tracks filtered by time, removing older tracks.

In (Gohde, Boddy, and Shackleton 2013), we presented G2I2, a system that improves the routes generated by an off-road planner by altering a movement cost function based on map features of terrain type and slope. G2I2 can also alter costs on the map at given coordinates, rather than based on annotated features. These "local features" can be used to make coordinate-based alterations to the cost function, for example based on errors in the map such as a bridge that has been washed out. The signal used by G2I2 for making these adjustments come from comparing planned routes and the tracks resulting from execution based on those plans.

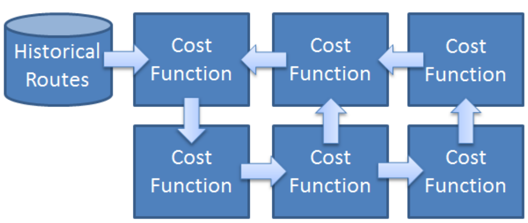

Figure 2: A cost function is induced from historical tracks, then updated by comparing routes generated with it to those taken by human route executors.

GUIDE extends G2I2 in several ways. Most notably, GUIDE includes time-varying features such as temperature and precipitation. In a large database of tracks gathered from real users, there is a very clear indication that these features do affect their routing preferences. GUIDE also adds the ability to *induce* a cost function based on historical track data, rather than just adjusting an existing cost function based on comparing planned routes and the resulting tracks. A tuning capability based on these comparisons has also been included in GUIDE, re-implemented as required by the different form of cost functions used by GUIDE. Figure 2 shows how the cost function is used and altered. In the rest of this paper, we describe in more detail how GUIDE functions and present some results on system performance.

3 Inducing a Cost Function

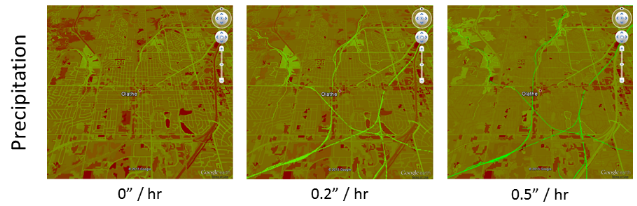

Figure 4: Preferences vary with increasing precipitation

Starting with a set of tracks and a map describing the area traversed, GUIDE induces a movement cost function over the map. This cost function is based on *features* associated with map locations, rather than with the locations themselves. Consequently, the resulting cost functions can be used to plan routes over parts of the map that were not traversed by any of the tracks used in training.

Induction of this cost function is a two-step process. First, GUIDE computes the relative amount of a given terrain type present in each track, compared to the prevalance of that terrain type in the area around the route, resulting in a numeric value between -1 (the terrain is only present in the track, and comprises the entire track), and 1 (the terrain is only present in the surrounding area, and comprises the entire surrounding area). This value is mapped via a sigmoid function to a value between 0 and 255, so as to provide a proxy cost (inverse preference) for that type of terrain under the prevailing conditions (slope, speed, temperature, and precipitation). We chose a sigmoid over a linear mapping for better separation around 0 in the comparison above. Few comparisons result in values close to -1 or 1.

Terrain type is the only discrete feature GUIDE uses. Consequently, the cost function induced is represented as a set of functions over continuous variables, one for each terrain type. Each of these cost functions is computed as follows. For each track in the training set, GUIDE performs a linear regression over a combination of the proxy cost and a quadratic combination of the continuous features listed above. The result is a set of linear coefficients over the features and feature combinations, defining a cost function for the given terrain type.

The use of a quadratic kernel function is motivated by the insight that preferences are not well represented by a cost computed as a linear combination of features. There are dependencies: terrain preferences can flip, based on things like the current temperature, or whether it is raining, or a combination of the two. Figure 3 shows the cost of traversing three

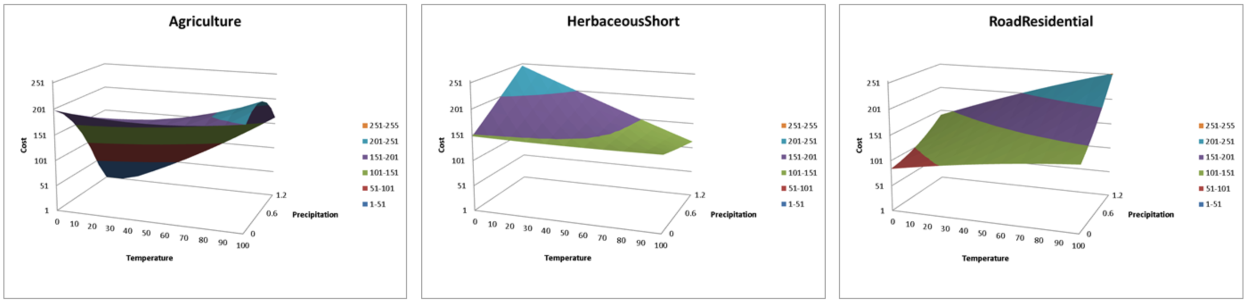

Figure 3: Relative cost to traverse terrains with varying temperature and precipitation

different terrain types with varying temperatures and precipitation with all other feature values in the induced cost function held constant. These values are relative: the decrease in cost for agriculture as precipitation increases at a temperature of 10 degrees indicates that agriculture becomes more preferable *in relation to other terrain types* as precipitation increases.

The resulting effect on relative costs is shown in shown in Figure 4. In those maps, redder is more costly, greener is less costly (i.e., more preferred). It can clearly be seen that time-varying conditions are significant: there is a clearly increased preference for roads once there is any precipitation.

4 Tuning by Pairwise Comparison

In our previous work on G2I2, we implemented a function for learning cost updates from the differences between a planned route and the track as executed, based on that plan. Again, the objective is to use the cost function as a proxy for human preferences, and the assumption is that divergences between the planned route and the executed track represent errors in modeling those preferences. GUIDE includes the same capability, modified to account for the fact that GUIDE uses a different form of cost function than G2I2. Similar to the process of inducing a cost function described in the previous section, terrain types over-represented in the planned route (compared in this case to the executed track, not the surrounding area) have their costs increased in those conditions, while those under-represented have their costs decreased in those conditions. Updates may be local, affecting a specific region on the map, or global, where costs for traversing all instances of a terrain type are altered.

Among the reasons that G2I2's approach to tuning cost functions will not work in GUIDE is the large increase in the number of dimensions in the cost function: in addition to adding new features, we have added quadratic combinations of all features, both new and old. Consequently, tuning updates in GUIDE are localized, adjusting the cost function at and around a specific combination of feature values, rather than making a more global update. For example, because the update is localized, changes based on routes executed in cold weather do not affect the values calculated by the cost function in warm weather. This local adjustment is accomplished by adding or subtracting a bell-shaped function the magnitude of which is scaled by the difference in the proportion of the terrain type, centered on the feature values for those conditions. The width of the bell is scaled along each feature proportional to the span of the data found in that feature.

The function used in GUIDE for local adjustments is

$$e^{-\frac{1}{2}\left(\frac{(C_1-U_1)^2}{(span_1/100)^2}+\ldots+\frac{(C_n-U_n)^2}{(span_n/100)^2}\right)}$$

C is the set of current feature values, U is the set of feature values from the route that produced the update, and *span* is the span of the data for that feature (i.e., along a given dimension of the cost function). Each update (one per route/track comparison) is added to the cost function for each terrain type, scaled as described above by the relative terrain type proportions.

Figure 5 shows a cost function limited to varying temperature, with two cost updates, one increasing cost around 10F, one decreasing cost around 80F. In practice, GUIDE accumulates updates over numerous comparisons, making small local adjustments for each one. Over time, the cost function changes to reflect the route executor's preferences.

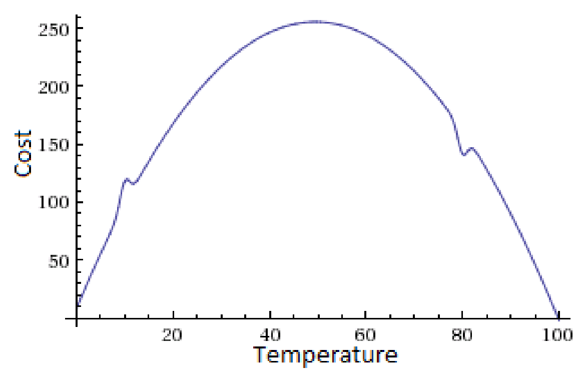

Figure 5: Single feature cost function with learned updates

5 Experiments

For all of the experiments reported in this section, the route planner being used was Primordial's Ground Guidance off-road route planning system[1]. We evaluated GUIDE's ef-

[1]http://www.primordial.com/index.php/products/ground-guidance

Cost Induction	Mean	Quartile		
		1st	2nd	3rd
Terrain Error	4.0%	0.1%	0.5%	4.0%
Absolute Cost Error	1.6%	0.1%	0.2%	0.7%
Cost Error / distance	4.3%	0.1%	0.4%	2.7%
Pairwise Tuning	**Mean**	**Quartile**		
		1st	2nd	3rd
Terrain Error	3.4%	0.0%	0.3%	2.8%
Absolute Cost Error	2.6%	0.1%	0.2%	1.2%
Cost Error / distance	3.9%	0.2%	0.8%	3.1%

Table 1: Learning a cost function from cost-optimal routes

fect on routes generated by Ground Guidance in experiments with two different datasets. The first is a set of 15,000 cost-optimal routes produced according to a known cost function. These routes were generated using random start and end points for each route, up to 1.5 kilometers apart, within an area of roughly 11,000 km^2.

The objective for experimenting with this dataset was to determine the accuracy with which GUIDE can reconstruct a cost function from a set of routes produced using that cost function. The second dataset consisted of 17,000 tracks captured by human route executors using GPS devices. These routes are very far from optimal: the tracks include such things as people running around a school track, or tracing a path through their neighborhood on an evening walk. The objective in this case was to evaluate GUIDE's ability to learn a cost function that captures human preferences, even in presence of serious sub-optimality. For each dataset, 90% of the routes are used as a training set to induce a cost function, with 10% withheld for evaluation. Subsequently, roughly 10% of the training set are used for further tuning of the cost function through pairwise comparison of planned and actual routes for a given source and destination, evaluated using the same evaluation set.

In evaluating the cost function, we use three metrics. The first is *terrain error*, measuring the difference in proportion of terrain types between a route planned by the cost function with the same start and end points as an evaluation route. For example, a planned route traversing 60% trail and 40% residential roads compared to an evaluation route that traversed 40% trail and 60% residential roads has an error of 20%. The second is *cost difference* between the planned route and evaluation route. The third is *cost per unit distance*.

Table 1 shows the results for the first dataset, consisting of cost-optimal routes. For initial induction of the cost function from the tracks in the training set, terrain error is low. Differences in total cost and cost per unit distance between planned and actual routes in the evaluation set are also small. In all three cases, the median error (shown as the 2nd Quartile value) is much lower than the mean error, showing the results are strongly skewed toward lower error, with a few outliers with higher errors. With the addition of tuning using pairwise comparisons, mean terrain error and cost per unit distance improve while absolute cost error worsens. This indicates that the pairwise learning is successfully adapting the cost function to mimic the terrain type choice, but has produced slightly longer routes to do so. The change in cost per unit distance is small, with a minor decrease in mean and increase in median error. Median error was less than 1% across all metrics, indicating GUIDE was able to faithfully reconstruct the output from the cost-optimal cost function.

Cost Induction	Mean	Quartile		
		1st	2nd	3rd
Terrain Error	27.2%	17.3%	25.5%	36.3%
Absolute Cost Error	81.7%	32.0%	53.4%	95.3%
Cost Error / distance	24.1%	9.0%	19.4%	33.6%
Pairwise Tuning	**Mean**	**Quartile**		
		1st	2nd	3rd
Terrain Error	26.6%	16.3%	24.9%	36.4%
Absolute Cost Error	60.7%	22.1%	38.85%	75.54%
Cost Error / distance	8.6%	2.3%	5.9%	10.8%

Table 2: Learning a cost function from historical tracks

In the second set of experiments, we tested GUIDE against routes produced by human route executors, as captured by GPS devices. Table 2 shows the results of inducing a cost function and of tuning using pairwise comparisons against that induced cost function. The results differ considerably from the first set of experiments. One problem was that the track data itself is noisy, with numerous GPS registration errors large enough to displace tracks from one terrain type (e.g., a road) to another (such as a nearby field), and map errors, such as missing roads. Figure 6 shows examples of these sorts of errors. In order to perform a more accurate evaluation, we corrected most of the GPS registration errors in the evaluation set. The training set was left unmodified, as a test of GUIDE's robustness (and because real data is likely to have similar problems).

Figure 6: Left: GPS registration errors between plan (yellow/top) and actual (white/bottom). Top Right: Terrain map missing roads. Bottom Right: Aerial imagery with roads.

Several additional aspects of this dataset make learning a cost function more difficult. First is that many of the routes are not even close to cost optimal. Many routes have start and end points that are nearly identical, such as when somebody goes out jogging. While the historical track may be a 1 kilometer loop, a cost optimal planner will produce a short, direct route of a few meters. To combat this, we created a waypoint at each quarter of the distance along the track, forcing the planner to find paths to points along the loop. Some of the tracks were even worse than loops. For example, the track shown in Figure 7, drawn from this dataset, loops back on itself in such a way that adding waypoints based on distance along the track does not help.

We also found that people appeared to be making route choices based on features that do not appear in the map. For example, someone might take a road away from their starting point, then return to it along a nearby trail, apparently preferring the novelty of not retracing their steps to sticking to which of road or trail would otherwise be preferable. Adding a cost for retracing a route in this way requires changes to the route planning algorithm itself, not just the cost map, and so is outside GUIDE's scope, at least for now. The result of this behavior is that GUIDE will induce very similar costs for road and trail, since they appear to be equi-preferable in some cases. These equivalences mean that the terrain error metric is not indicative of how well GUIDE has performed in this set, because very small differences in cost can lead to large differences in terrain choice, where those differences are not in fact very important to the human user.

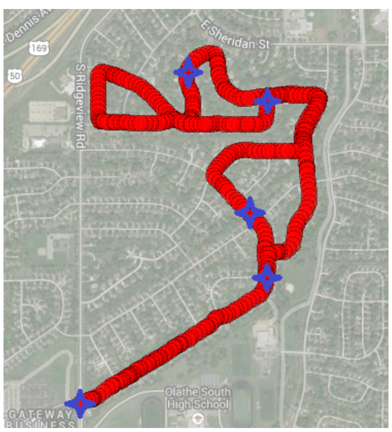

Figure 7: Route with loops that produces large cost error. Stars show automatically added waypoints.

The second metric, absolute cost error, is also not an effective way to measure the performance of GUIDE against this set. As stated, the historical routes are not cost optimal. Instead, many of the routes may be trying to travel a specific distance or for a set amount of time, as is often the case in walking or running for health or pleasure. Again, reflecting these objectives in planned routes will require modifying the behavior of the planner, not just the cost function.

The third metric, cost per unit distance, compensates for both terrain equivalences and lack of route optimality. Thus, even if a historical track consists of walking multiple circuits of a loop along both roads and trails, the error will be low

Terrain Type	Track Presence	Min Cost	Max Cost	Mean	SD
Trail	25%	34.1	49.9	38.4	1.9
Res. Road	18%	32.0	50.2	42.5	2.0
Arterial Road	12%	34.5	42.0	40.2	0.7
Herb, Short	17%	55.4	73.4	59.8	1.8
Dev., Low	15%	54.6	79.8	60.6	3.9

Table 3: Common terrain types with presence and costs.

if the cost function has correctly assigned costs to the terrain types and produces a route covering terrain types in the equivalence class. GUIDE produces errors even with this metric with only the induced cost function, but these errors are greatly reduced after performing pairwise updates.

One potential confound for this metric would be if *all* of the terrain types commonly occuring in tracks had very similar costs (i.e., were all roughly equi-preferable). Table 3 show the costs found for common terrain types and the proportion of historical routes they make up. Each terrain type has a range of costs, because those costs vary depending on conditions. There are still clear dominance relations among them in terms of cost. Clearly, people are making choices to traverse terrain with significantly different costs. Coupled with the information in Table 3, the results in Table 2 indicate that the routes planned with GUIDE's adjusted cost function are making very similar terrain choices to those found in the training dataset of tracks.

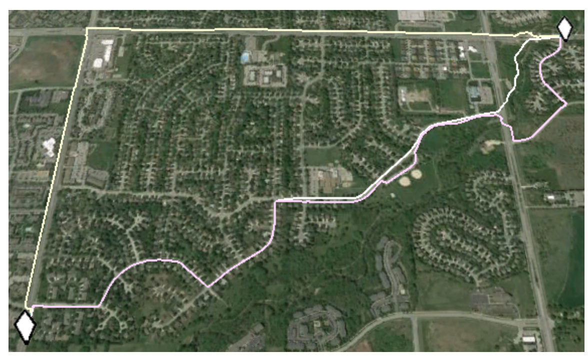

Figure 8: The cost function learned by GUIDE produces different routes in cold, snowy weather (yellow/top) than hot, rainy weather while the route planner's default cost function produces one route in all conditions (pink/bottom).

Finally, we showed in Section 3 that the cost functions induced by GUIDE include a significant influence from time-varying features such as temperature and precipitation. Figure 8 shows that this influence does in fact lead to the generation of significantly different routes by the route planner.

6 Related Work

There is a great deal of previous work on route planning for humans and robots, on-road and off-road, in static and dynamic environments. The work in this area that is relevant to GUIDE is that which seeks to learn or adjust the nature of the routes returned. David Silver and others have developed theory and tools for learning cost models for robots moving through configuration spaces, including but not limited to map traversal, through the application of inverse optimal control (Ratliff, Bagnell, and Zinkevich 2006), (Silver, Bagnell, and Stentz 2008) (Silver 2010). This work is very relevant to GUIDE as a source of techniques for analyzing and improving performance. However, these approaches assume a cost-optimal or near cost-optimal route to learn from and do not take into account time-varying features such as weather. In (Kavraki et al. 1996), *probabilistic roadmaps* are used to address a related problem, but with a different emphasis. In that work, the objective is to find collision-free movement paths in configuration space for holonomic robots with several degrees of freedom. Finally, our work can be differentiated from previous work on map learning such as SLAM[2] in several ways. Notably, we start with a map, albeit one that may contain errors of various kinds, and localization is not part of the problem.

There has also been previous work on systems that attempt to learn some form of model to better reflect human preferences. (Quercia, Schifanella, and Aiello 2014) and (Skoumas et al. 2014) describe methods for modifying generated routes that trade off distance and *emotional* components of the route as reported by previous travelers, such as beauty, quietness, or happiness. Waze is a community oriented application that tracks human route executors via their cell phone, updating traffic conditions by monitoring speed and location, and recalculating routes to avoid congestion.[3] (Rogers, Fiechter, and Langley 1995) describes an on-road navigation system that models the user's preference for different road classes, such as highway, freeway, arterial roads, and local roads, along with other route features such as driving time, distance, and number of turns and intersections. The modeled preferences are adjusted based upon comparison between proposed routes accepted or rejected by the user. In (Letchner, Krumm, and Horvitz 2006), a route planner called TRIP is described that uses previously executed plans in the form of GPS tracks to inform future route generation. This information is used to update speed information along roads for the time at which the trip was recorded. Additionally a user's inefficiencies (deviation from the fastest route) are bundled into a preference factor for non-optimal routes. TRIP then plans over route segments, discounting previously-taken segments by the preference factor.

7 Discussion and Future Work

In this paper, we have described the problem of providing effective route planning for humans, especially in complex and dynamic off-road domains. We presented GUIDE, a system that learns and iteratively tunes movement cost models for route-planners, so as to more closely match human preferences in a variety of applications, using data that is both noisy and very far from cost-optimal.

A GUIDE-augmented route planner is doing a form of "iterative planning," in which planning performance improves over time specifically because of the results of executing previous plans (Smith 2012). There are other ways in which we can usefully view plans as objects subject to manipulation and analysis. In work left out of this paper for reasons of both space and focus, we have implemented a capability for generating multiple plans, either as a set of options roughly following a Pareto frontier in a multi-attribute value space, or in the generation of *interestingly different* plans against the same objective function.

At present, we are working to integrate GUIDE with operational route planning systems. In addition to providing a more detailed validation of where GUIDE will be most useful, this integration will almost certainly impose additional requirements in the form of new features, preferences, or behaviors to be modeled. Finally, having demonstrated the effectiveness of this approach, we are also planning to put it on a somewhat firmer analytical foundation, for example using some of the techniques developed in (Silver 2010).

[2]Simultaneous Localization And Mapping

[3]https://www.waze.com/

References

Gohde, J.; Boddy, M.; and Shackleton, H. 2013. Learning from previous execution to improve route planning. In *ICAPS Workshop on Planning and Learning*.

Kavraki, L.; Svestka, P.; claude Latombe, J.; and Overmars, M. 1996. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. In *IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION*, 566–580.

Letchner, J.; Krumm, J.; and Horvitz, E. 2006. Trip router with individualized preferences (TRIP): incorporating personalization into route planning. In *Proceedings of the 18th conference on Innovative applications of artificial intelligence - Volume 2*, IAAI'06, 1795–1800. AAAI Press.

Quercia, D.; Schifanella, R.; and Aiello, L. M. 2014. The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. *CoRR* abs/1407.1031.

Ratliff, N.; Bagnell, J. A. D.; and Zinkevich, M. 2006. Maximum margin planning. In *International Conference on Machine Learning*.

Rogers, S.; Fiechter, C.-N.; and Langley, P. 1995. An adaptive interactive agent for route advice. In *Proceedings of the Third International Conference on Autonomous Agents (Agents99)*, 198–205. ACM Press.

Silver, D.; Bagnell, J. A. D.; and Stentz, A. T. 2008. High performance outdoor navigation from overhead data using imitation learning. In *Robotics Science and Systems*.

Silver, D. 2010. *Learning Preference Models for Autonomous Mobile Robots in Complex Domains*. Ph.D. Dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Skoumas, G.; Schmid, K. A.; Jossé, G.; Züfle, A.; Nascimento, M. A.; Renz, M.; and Pfoser, D. 2014. Towards knowledge-enriched path computation. *CoRR* abs/1409.2585.

Smith, D. 2012. Planning as an iterative process. In *Proceedings of AAAI*.