

ROSE – AN INTELLIGENT MOBILE ASSISTANT

Discovering Preferred Events and Finding Comfortable Transportation Links

Bjørn Zenker and Bernd Ludwig
*Friedrich-Alexander-University Erlangen-Nrnberg
Haberstrasse 2, 91058 Erlangen, Germany*

Keywords: Mobile navigation system, Route planning.

Abstract: In this paper, we describe ROSE (Routing Service), an application for mobile phones, which suggests the user events and locations and guides him to them, using the public transport system. It determines the best possible transport link and accompanies passengers throughout their entire journey. Further, it reacts in real time to delays in the public transport system and calculates alternative routes when necessary. For route planning in this context, we will propose a $h_{\epsilon u}$ -optimal algorithm for incorporating non-monotone multi dimensional user preferences. The algorithm is based on an analysis of theoretical foundations for real world route planning problems and leads to a new approach of how recommendation and route generation subsystems should be coupled to increase user satisfaction.

1 INTRODUCTION

State-of-the-art navigation systems provide point-to-point navigation for their users: given a starting point and an ordered list of destinations, the navigation system computes a shortest path between subsequent points and guides the users from one point to the next in the list. A major drawback of this type of assistance is that the user has to know where he wants to go to. Often however, this is not the case. For example, tourists in unfamiliar cities cannot apply a navigation system successfully until they have looked up the places available to visit, i.e. a famous castle. Even city residents often do not know where to go in order to spend their leisure time or to settle a private or professional matter.

In on-the-move situations without having the usual sources of information at hand, this is not practical. To ease the whole process of planning, we are developing ROSE: it combines recommendation of events and locations with navigation. While the provided services can be easily accessed from nearly every mobile phone, the ROSE server incorporates and preprocesses data from different web sources, like live public transport data, event and location directories and map services.

In many situations, users need to get to more than a single location. From a theoretical point of view, this results in a traveling salesman problem. In con-

trast to transport companies, pedestrians are much more flexible in planning their tour: They do not have fixed time tables, so the problem of satisfying many constraints can be relaxed: the ordering of visiting the locations is arbitrary, so tour plans may be changed at travel time. Often, even the location is not fixed because users want to satisfy “tasks” (like buying printer paper or getting something to eat) without having a certain shop in mind. This allows tour plans to be rearranged interactively: at any time, the system can propose a near-by shop for printer paper and settle one item in the total list of matters.

In this paper, we present an iterative algorithm that approximates graph theoretical optimization problems of tour planning by applying the relaxations just described. Before discussing the algorithm, the paper presents the ROSE system and its current state of implementation comparing it to other state-of-the-art systems. Next, we discuss differences between optimization criteria that are user adaptive and admissible heuristics that are used regularly for finding optimal solutions in graphs. We give an account of the underlying hierarchy of graph-theoretical problems and outline how the idea of relaxing the optimization might help in developing tractable algorithms. We explain our implementation of optimizing user adaptive criteria as a multi-attribute decision problem and present examples encoded in our programming language MADL for multi-attribute optimizations.

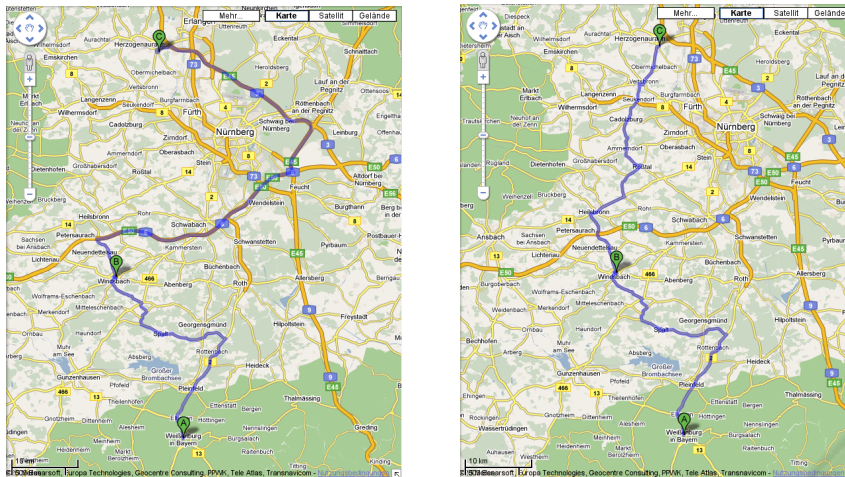


Figure 1: Two routes computed by Google Maps: On the left Google's best recommendation; on the right Google's second best recommendation.

2 RELATED WORK

In this section, we argue that the issues of user adaptive heuristics for planning in assistance systems and combination of services from a user-oriented perspective have not been considered extensively enough in research on intelligent assistance systems.

2.1 The Need for User Preferences in Search

Multi-attribute optimization is a well established area of research. However, as efficient search algorithms need admissible optimization function for heuristic evaluation of partial solutions in order to efficiently prune the search space, weighted sums are the preferred type of heuristics. They provide a clear advantage: as they are a mapping from a set of influence factors to the real numbers, a set of options the search algorithm has to decide between can always be ordered totally whereas multi-attribute decisions just compute a pareto-optimal solution. In general, it contains more than a single element. Therefore it remains unclear which decision is the best for optimal pruning.

On the other hand however, weighted sums suffer from an important disadvantage: they are bad at explaining their decisions and – even worse – at allowing the user to take influence on a decision. To illustrate this point, let us look at an example taken from the Google map server (<http://maps.google.de>): we are asking for a route (using a car) from between the three German cities Weidenburg, Windsbach, and Herzogenaurach. The route considered best by Google can be seen on the left map in Fig. 1. It is

123 km long and takes one hour and 42 minutes for a normal car. The second best recommendation is the map on the right side in Fig. 1. It is 90 km long and takes one hour and 45 minutes: the advantage of 3 minutes is reached at the cost of 33 km! This example is a case for multi-attribute decisions which allow the user to state explicitly what his preferences are. Depending on the driver, highways may be preferred (taking long distances into account) or not (preferring to save fuel or just disliking the high speed on German highways). Generalizing this example, in Sect. 4 we present an algorithm that incorporates user-adaptive preferences into heuristics for pruning search spaces.

2.2 Comparison to Similiar Systems

ROSE's application domain is also treated by other researchers. Similiar approaches for recommendation and navigation systems have been implemented in COMPASS (van Setten et al., 2004) and Magitti (Bellotti, 2008). They employ different prediction strategies in recommendation and rate the yielded results based on user preferences. P-TOUR (Maruyama et al., 2004) uses a genetic algorithm to find different near-best solutions and presents them to the user in a k-means clustered overview, from which he can select his preferred route. RouteCheckr (Voelkel and Weber, 2008) employs a multi-criteria Dijkstra based algorithm, which remains limited to the usage of the weighted sum. As it does not use an estimation function, it does not have to cope with nonmonotone and optimal criteria, but is presumably slower than an algorithm using such a heuristic. Hochmair (Hochmair, 2004) studies, which decision rules bicy-

clists utilize in route planning and discusses different decision rules. He concludes that a compensatory decision rule should be used, but he does not implement this concept in a new algorithm. These systems do not support public transport.

In contrast, PECITAS (Tumas and Ricci, 2009) is a mobile personalisable navigation system, which advises routes using means of public transportation. However, it does not include recommendation of events or locations and routes are restricted to one starting point and one destination point only. For user adaptation, PECITAS generates multiple routes by using different heuristics (e.g. fastest route, or not taking any bus) and ranks them according to user preferences (walking preferences, number of bus changes, arrival at destination, sightseeing).

Compared to these systems, the unique features of ROSE are:

- routing to multiple destinations,
- integration of recommendation, route generation with live public transport
- support and navigation
- usage of various compensatory decision rules.

3 OVERVIEW OF THE ROSE SYSTEM

In this section, we present a short overview of the current implementation of ROSE and its client-server architecture. In a sample session we demonstrate how ROSE is used in practice.

3.1 Pedestrian Navigation, Event Recommendation, and Live Public Transport Routing – All in One

To get a recommendation, the user enters a query, like 'modern opera', into his mobile phone (see 2, left). The recommender then generates a list of suggestions based on the user input and the user's preferences. In this example it would likely be a list of events which feature music similar to the users preference (see 2, right). After the user has chosen one of the presented options, the system calculates a route from the current location to the selected goal. Figure (see 3, right) shows a route overview from the users current point to his chosen destination. To consider user preferences in route generation, we propose a h_{eu} -optimal algorithm in section 4.

To ease the travelling, public transportation is also considered. The system calculates a route from the

user's current position to the nearest public transport option, which means of transportation to take, where to change transportation and how to walk from the last stop to the goal location. Departure times are displayed to the user and he is informed, i.e. if he has to hurry to catch a bus.

3.2 System Architecture

The ROSE system consists of a server, which calculates recommendations and routes and a client which is used for user input, display of results and navigation. More details can be found in (Ludwig et al., 2009).

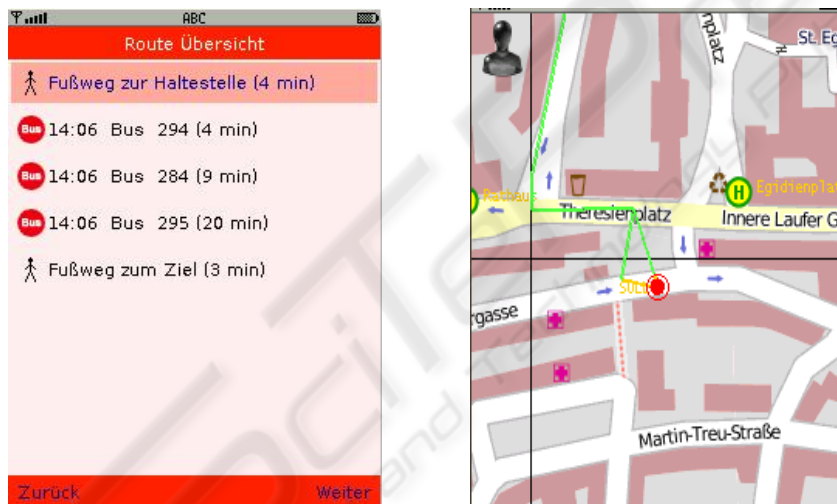
3.3 The Core Issue: User Adaptive Optimization

We structured the system into three main services: recommendation, route generation and navigation. In the current release of the system, all services are coupled loosely: the results of the recommendation are the input (goals) of the route generation. The result of the route generation is the input (way) of the navigation service. Such a loose combination suffers from a number of drawbacks:

- It does not provide for replanning when some unforeseen event (e.g. missing a bus, the printer paper shop being closed exceptionally) happens.
- The algorithm is challenged by a huge search space: the fact that busses travel according to time tables results in a search graph with a quite tremendous number of edges.
- In most cases, there are many locations meeting the user's preferences. For example, many restaurants sell good pizza. As a consequence, even the problem an optimal solution is searched for is not defined uniquely.
- Finally, the selected locations may be far away from each other, allowing for many degrees of freedom in ordering them to form a tour.

Our conclusion from all these issues is that a close coupling of recommendation is needed which interleaves route generation and navigation. This results in a complex graph theoretical optimization problem like the Multi Path Orienteering Problem with Time Windows (MPOPTW) (Garcia et al., 2009). (Ludwig et al., 2009) presents an hierarchical overview (compare Figure 4.) of different problem classes in the routing context.

Additionally, the impact of close and loose coupling of the recommendation and route finding services on the user satisfaction shall be researched. As

Figure 2: *left*: Query, *right*: List of recommendations.Figure 3: *left*: Details of recommendation, *right*: Route overview.

we presume that the algorithmic complexity of close coupling is too big and that loose coupling is not satisfactory, we want to investigate, how an intermediate coupling between these the extremes could be realized.

4 ALGORITHMS FOR COMFORTABLE ROUTES

Criteria for evaluating the quality of a route are limited mostly by formal constraints dictated by the algorithm used to find optimal paths. Efficient greedy graph search algorithms require the heuristic function to be monotone; A* even requires the heuristics to be

optimistic, i.e. to never overestimate real costs of a path. In practice however, such constraints for heuristics are not adequate to reason about user preferences. In a survey conducted at our computer science institute among public transport users, the following criteria were marked as important by the test candidates:

- No long waiting time until departure
- Short duration of the trip
- Short foot walks
- Few changes of transportation
- No long waiting time during changes

Optimistic estimates for these criteria are just the function $f(x) = 0$; this amounts to omitting the criterion completely, which is an undesired consequence.

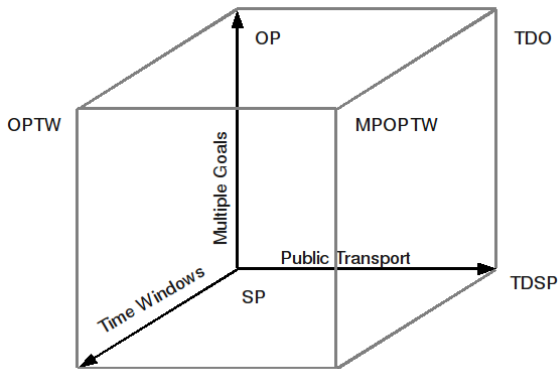


Figure 4: Routing Problems for Different Needs.

A second important finding of our study is that users do not evaluate the utility of a route on a one-dimensional scale (where there is always an optimum), but try to find a compromise between multiple attributes that are not comparable among each other. They accept locally sub-optimal proposals optimizing the benefits of a proposal and minimizing its risk globally.

For example, somebody traveling with a lot of luggage accepts using a bus line that arrives some minutes later at the train station than the fastest one, but is much less crowded. Obviously, in this context, the comfort of the trip is valued higher than the duration. From an algorithmic point of view, this means that the standard shortest path approach cannot be applied successfully in order to satisfy the user needs as good as possible. However, searching according to a heuristic function that forces the search procedure to visit (almost) the whole search space is no attractive option for developing programs intended to run in real-time.

4.1 An $h_\epsilon u$ -optimal Algorithm for Comfortable Routes

The key to an efficient solution that belongs to the complexity class as A^* and retains its soundness and completeness is therefore to use a) a monotone heuristic function h for computing correct solutions and b) to incorporate a non-monotone, multi-attribute heuristic u for the user preferences into the search procedure.

In Fig. 5 you see two paths from the start node s to a goal node g . g_i denotes the actual costs from s to the node which is currently being expanded, h_i the estimated costs to g according to the heuristic function h . u_i denotes the actual valuation from s to the node which is currently being expanded, t_i the esti-

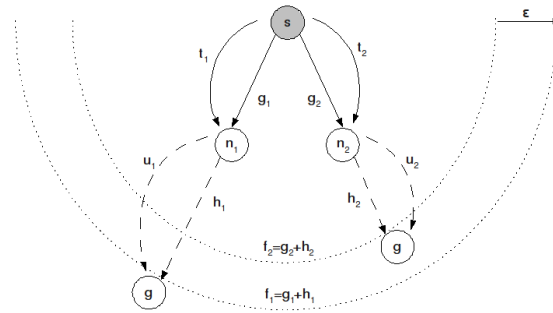


Figure 5: $h_\epsilon u$ -optimal Expansion of the Search Space.

mated valuation to g according to the multi-attribute heuristic function u (user preferences).

If there is exactly one optimal successor state with $|g_1 + h_1 - g_2 - h_2| > \epsilon$, the search procedure works as usual. In the case of $|g_1 + h_1 - g_2 - h_2| \leq \epsilon$, both paths from s to g in Fig. 5 are undistinguishable in h .

In practical applications, if $|g_1 + h_1 - g_2 - h_2| \leq \epsilon$ then often other criteria become relevant for decisions. In order to explain the meaning of ϵ , let us consider the two public transport connections from s to g in Fig. 5: one takes 50 minutes, the other one 47 minutes. However, the first connection requires fewer changes of transportation than the second one. If the user dislikes changes, the first connection is optimal according to the user preferences u under the assumption of an ϵ -tolerance $\epsilon \geq 3$ minutes.

In order to select a path that is both optimal in the sense of h given ϵ and in the sense of u , we compute the ranking $p_1 = t_1 + u_1$ and $p_2 = t_2 + u_2$ of both options. The vectors p_1 and p_2 represent all user preferences in u . Each dimension represents a single preference (cf. the list of criteria in section 4). As the preference space to which p_1 and p_2 belong is partially ordered, in general there is no unique minimal element. Therefore, we compute a pareto-optimal set which includes all paths which cannot be distinguished neither by h nor by u . The final selection of a path which is h_ϵ -optimal and u -optimal is performed by applying a decision procedure. For calculating decision procedures, we use the MADL programming language, as described in (Ludwig et al., 2009). We call the result $h_\epsilon u$ -optimal.

4.2 An Approximation of the $h_\epsilon u$ -optimal Algorithm

Instead of implementing the $h_\epsilon u$ -optimal algorithm directly, we simulated its behavior by having the path search procedure compute the n -best list of solutions for a given destination. Each of these n solutions is

evaluated according to the user preferences. The final set of $m < n$ solutions is the set of m routes which are $h_{\epsilon u}$ -optimal.

In our first prototype, we implemented the n -best approach in order to obtain an evaluation platform as fast as possible. The algorithm works in two steps:

1. Compute n Best Results

Just computing the n best routes using always the same heuristics often leads to proposals that only differ minimally among each other — in particular, if just one line serves as public transport to the destination. Therefore, it is better to compute n routes using n different (optimistic) heuristics and to compare the n resulting best routes. This observation has also been made by (Tumas and Ricci, 2009).

2. Rank the n Best Lists

In order to get a global score for each user preference and each route, we sum up the contributions of each segment of the route to each criterion. The sum is called the rating of the route corresponding to the criterion under investigation. Finally, a total score is computed: The easiest approach to take multiple criteria into account is to compute a weighted sum of all criteria by multiplying each rating with the weight for the preference as entered by the user in the configuration dialog for the ROSE system (see Figure 1 left).

A more elaborate approach is to compute a pareto-optimum for the rating of the n routes. Beyond that, it is possible in our system to apply other multi-attribute decision rules, as the *Take The Best* decision rule proposed by (Gigerenzer and Goldstein, 1996).

5 CONCLUSIONS AND OUTLOOK

ROSE is a recommendation, route planning and navigation system for mobile devices. To integrate user multiple preferences and decision rules into the route planning process, we developed a $h_{\epsilon u}$ -optimal algorithm.

We tested ROSE on a major event with 20.000 people to recommend subevents and calculate routes using the local bus network. We conducted a user study, from which we hope to get more information about the relevance of different user preferences in recommendation and navigation.

REFERENCES

- Bellotti, V. (2008). Activity-based serendipitous recommendations with the magitti mobile leisure guide. *CHI 2008 Proceedings*, 21:498–516.
- Garcia, A., Arbelaitz, O., Otaeguui, O., Vansteenwegen, P., and Linaza, M. (2009). Public transportation algorithm for an intelligent routing system. *16th ITS world congress, September 21-25, Stockholm, Sweden*.
- Gigerenzer, G. and Goldstein, D. (1996). Reasoning the fast and frugal way: Models of bounded rationality. *Psychological Review*, 103:650–669.
- Hochmair, H. (2004). Decision support for bicycle route planning in urban environments. *Proceedings of the 7th AGILE Conference on Geographic Information Science, Crete University Press, Heraklion, Greece*, pages 697–706.
- Ludwig, B., Zenker, B., and Schrader, J. (2009). Recommendation of personalized routes with public transport connections.
- Maruyama, A., Shibata, N., Murata, Y., and Yasumoto, K. (2004). P-tour: A personal navigation system for tourism. *Proc. of 11th World Congress on ITS*, pages 18–21.
- Tumas, G. and Ricci, F. (2009). Personalized mobile city transport advisory system. *ENTER Conference 2009*.
- van Setten, M., Pokraev, S., and Koolwaaij, J. (2004). Context-aware recommendations in the mobile tourist application compass. *Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 235–244.
- Voelkel, T. and Weber, G. (2008). Routechecker: personalized multicriteria routing for mobility impaired pedestrians. *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, pages 185–192.