ROSE – Assisting Pedestrians to Find Preferred Events and Comfortable Public Transport Connections

Bjørn Zenker
FAU Erlangen-Nürnberg
Haberstr. 2
91058 Erlangen
+49 9131 85 28714
bjoern.zenker@informatik.uni-erlangen.de

Bernd Ludwig
FAU Erlangen-Nürnberg
Haberstr. 2
91058 Erlangen
+49 9131 29911
bernd.ludwig@informatik.uni-erlangen.de

ABSTRACT

In this paper, we describe ROSE (Routing Service), a mobile application which combines pedestrian navigation with event recommendation and live public transport support. It determines the best possible transport link and accompanies passengers throughout their entire journey. Our motivation is to free the passenger from as many tedious tasks as possible in finding an interesting event and on the way to it. 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 two step algorithm for incorporating non monotone and non optimistic multi dimensional user preferences in an A*-like algorithm. We also present an assignment of theoretical foundations to real world route planning problems.

Categories and Subject Descriptors

H.1.2 [**Information Systems**]: User/machine systems – *human factors, user preferences*

General Terms

Algorithms, Human Factors

Keywords

pedestrian navigation, public transport, multi-attributive decision making, user preferences

1. INTRODUCTION

With the rapid developments of mobile hardware, more and better navigation systems are developed. They guide the user from one point to another, while accounting to the users preferences. This point-to-point navigation has the drawback, that the user has to know, where he wants to go to. This is not always the case: tourists in unfamiliar cities can only use the navigation system, after they have looked up, where they want to go, i.e. to see a famous castle. But also residents often don't know, where to go or what to do. If they want to enjoy an jazz concert, they first have to look up, when and where there is a concert and then they have to plan the trip to it. For this, they have to use at least two different services: one to lookup the concert and a navigation tool. To ease the whole process of trip planning, we are developing ROSE which combines the recommendation of locations and events 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.

1.1 Presentation of the Problem: Combining Pedestrian Navigation with Event Recommendation and Live Public Transport Routing

After the user entered a query, like 'eat pizza', the recommender generates a list of suggestions based on the user input and the users preferences. In this example it would likely be a list of restaurants which sell pizza. After the user choose an alternative, the system calculates a route from the current location to the selected goal. To consider a diverse set of user preferences in route generation, we propose a two steps method in section 2.

To ease traveling, public transportation is also considered. The system calculates a route to next best public transport stop, 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. The ROSE project also integrates live public data, which gives rise to new problems: what to do, when a bus scheduled in a travel route is known to be delayed?

As was shown, the system can be separated in three main services: recommendation, route generation and navigation. All services can be loosely coupled: 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 lacks the flexibility needed in many spacial situations, especially when errors occur, or the user behaves in an unpredicted manner. For example, if the user misses a bus, the system has to decide, what to do: wait for the next bus, take another line, walk,... Or if the recommender suggest the best events, but they are situated too far away from each other.

To address these problems, we suggest a close coupling of recommendation, route generation and navigation, which results in a theoretical problem formulation, as it can be seen in section 3.

Mobility 2009, Sep 2-4, Nice, France Copyright © 2009
ACM 978-1-60558-536-9/00/0009……$5.00

1.2 Current State of the ROSE System

To address the limitations of mobile devices like limited computational power, slow and expensive Internet access, we constructed a client/server-architecture. Expensive calculations are moved to the server and the transferred amount of data is minimized.

The ROSE system consists of the ROSE server, a J2EE application which integrates different services from multiple service providers and offers them as web services to the ROSE client. Routes, timetables and live public transport data is obtained via VPN from a local public transport company.

At the moment, we are developing two prototypes of a thin client: one running on J2ME enabled mobile phones with a GPS receiver and one for Android. We have an integration of different location providers to allow high compatibility between different devices and to allow navigation in various locations. As the start and end of a travel is often in buildings, and to support navigation also in subways we need an suitable indoor localization technique. Therefore we incorporate Fraunhofer WiFi localization [13], which is unique in that way as it is working autonomous on the mobile terminal. Besides this, it doesn't need access to the WiFi access points and needs no information, where the WiFi routers are located.

2. MAIN CONTRIBUTION

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, e.g. 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
- Duration of the trip
- Length of foot walks
- Number of changes
- No long waiting time during changes

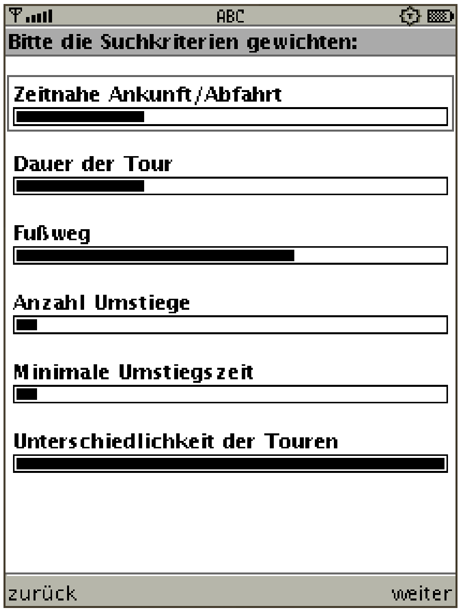
Optimistic estimates for these criteria are just the function f(x)=0; this amounts to omitting the criterion completely – an undesired consequence.

The second important observation is that users do not evaluate the utility of a route on a one-dimensional scale (where there is always an optimum in a finite set or closed interval of utility values). Users consider multiple attributes as equally important and try to find a compromise between options that are not comparable among each other, but taken individually are not dominated by another proposal.

This decision strategy leads to selecting proposals that may be locally sub-optimal, but from a global point of view are among those proposals that optimize the benefits of a proposal and minimize its risk.

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.

Figure 1: Configuration Dialog



Bitte die Suchkriterien gewichten:

Zeitnahe Ankunft/Abfahrt

Dauer der Tour

Fußweg

Anzahl Umstiege

Minimale Umstiegszeit

Unterschiedlichkeit der Touren

From an algorithmic point of view, this means that 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 even on mobile devices. The key to an efficient solution that retains the complexity results, soundness and completeness of fast algorithms such as A* is therefore to use additive heuristics for computing correct solutions and to incorporate multi-attribute heuristics into the search procedure when the search space is extended non-deterministically. This means that the algorithm with high priority finds solutions that are correct in terms of the search problem and sort solutions that are equivalent in terms of reaching the search goal according to a multi-attribute decision that represents the user's needs and preferences. Any time, the search procedure non-deterministically chooses an extension of the search space to test next, it evaluates the partial solution attributed to an extension according to the user preferences and then selects one of the pareto-optimal extensions.

This behavior can be simulated by letting the path search procedure compute a N-best list of solutions instead of just finding the optimal one. Each of these N solutions can be evaluated according to the user preferences. The final set of M<N solutions is the set of M routes which optimize both the monotone or even optimistic criteria and the non-monotone compromise between the user's preferences and their particular values for the routes under consideration.

2.1 Proposed Solution

In our first prototype, we implemented the N-best approach in order to obtain an evaluation platform as fast as possible.

a) 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 a better idea to compute N routes using N different (optimistic) heuristics and to compare the N resulting best routes.

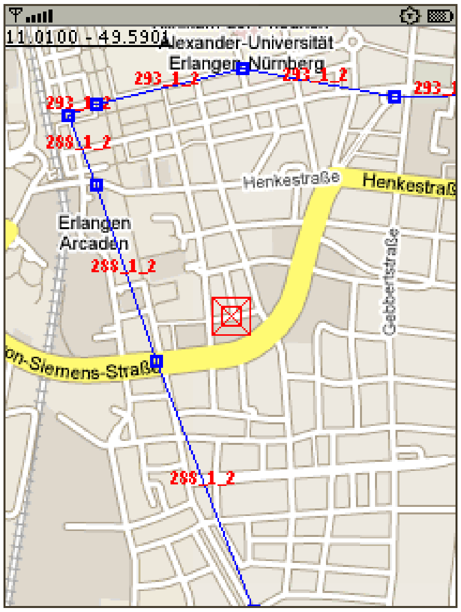
b) Rank the N best lists

The easiest approach to take multiple criteria into account is to compute a weighted sum of all criteria. In order to get a global score for each (user) criterion 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 by multiplying each rating with the weight for the criterion as entered by the user in the configuration dialog for the ROSE system (see Fig. 1).

We have implemented the described approach in JavaME for a standard Nokia N95 with 2GB of main memory. Fig. 2 shows a screenshot for a calculated route in downtown Erlangen (Germany).

Figure 2: Calculated route on mobile phone



2.2 Multi-Attribute Decisions

As known from decision theory, weighted sums are problematic as incommensurable data is mapped onto a unique scale. For example, the duration of a trip and the amount of space in a bus available for a passenger's luggage do not have the same units. Therefore, just multiplying such a value with a weight factor and adding it to some other weighted value makes it hard for the user to understand the system's decision. It is equally difficult for the designer of such a system to improve the selection process by adjusting the weights: no preferences of one criterion over others can be expressed explicitly.

In order to simulate better the way users take their decisions among several proposed routes, we implemented a module for multi-attribute decisions. Its basis is formed by a decision-oriented programming language.

Figure 3 shows a code fragment for a decision in this programming language, called MADL (Multi Attributive Decision Language). The decision named 'doWhat', using the pareto decision rule, consists of three alternatives, namely taking bus 293, waiting and taking bus 288 or walking and the goal is to minimize the three given attributes 'waiting Time', 'duration' and 'trip duration'.

Figure 3: MADL decision

```
ParetoDecision doWhat
{
ALT [bus293]
ALT [wait, bus288]
ALT [walk]

GOAL MIN!(waitingTime)
GOAL MIN!(changes)
GOAL MIN!(tripDuration)
}
```

The advantages of using MADL are, that all decision specific information is stored in one designated place and easily understandable and adaptable by humans. It allows to configure the parameters influencing a decision, the strategy used to make a decision in a given situation and also allows combination and inheritance of decisions. [11]

2.3 Comparison

ROSE's application domain is also treated by other researchers. Similar aproaches for recommendation and navigation systems have been implemented in COMPASS [4] and Magitti [2]. They employ different prediction strategies in recommendation and rate the yielded results based on user preferences. P-TOUR [5] 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 [6] 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 [7] studies, which decision rules bicyclists 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 [12] is a mobile personalisable navigation system, which advices 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, not taking any bus, ...) and ranks them according to user preferences (walking preferences, number of bus changes, arrival at destination, sightseeing).

3. ASSIGNMENT TO THEORETICAL PROBLEMS

The presentation of the ROSE system and its capabilities highlights the fact that applications such as ROSE should offer two modes of usage:

Single destination mode: the user wants to reach a single location or complete just one tasks and needs assistance in finding the appropriate location and a transportation link to it.

Multiple destination mode: the user wants to reach more then one location or complete several tasks. The locations or the respective tasks are partially ordered leading to some time constraints that have to be respected by ROSE when it recommends locations and transportation links.

In order to give an overview about the algorithmic complexity for the functionality that has to be provided by ROSE, in this section we sketch the graph theoretical problem for each (level of) functionality.

The most fundamental problem finding the shortest path from one location to another is known as the shortest path problem and investigated in depth. It solves the problem of finding a transportation link without respecting a time table.

An extension to this problem includes public transport. A theoretical definition of this problem known as time-dependent shortest path problem (TDSP) and according algorithms can be found in [9] and [8].

Table 1. Assignment of route planning problems to theoretical problems

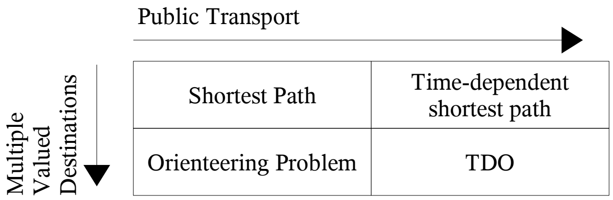
Route planning problem	Theoretical Problem	Algorithm
Single destination	Shortest Path	Dijkstra, A*
Single destination with public transport	Time-dependent shortest path (TDSP)	PFS[9] , Ding [8]
Multiple destination recommendation	Orienteering Problem (OP)	Fischetti, Salazar, Toth [10]
Multiple destination recommendation with time windows	Orienteering Problem with time windows(OPTW)	For an overview see [1]
Multiple destination recommendation with time windows and constrained attributes	Multi-constrained team orienteering problem with time windows (MCTOPTW)	Garcia et al. [1]
Multiple destination recommendation with public transport	Time-dependent orienteering problem (TDO) [15]	
Multi destination recommendation with public transport, time windows and constrained attributes	Proposal: Time dependent orienteering problem with time windows (MCTDOPTW)	

A different extension is to find a best path between multiple, differently valued locations. This is known as the orienteering problem (OP) [10], which is NP-hard.

Furtheron, considering opening hours of locations leads to the orienteering problem with time windows (OPTW). [1] gives an overview. This problem is extended for the tourist route recommendation context by [14] which takes multiple constrained attributes into account, e.g. a maximum price of the trip. [1] presents an iterated local search based algorithm which yields good results ('[...] below 2 seconds for problems with up to 100 locations [...]').

For our application scenario we need another extension to support public transport in route finding. As [9] shows, normal shortest path problems have little in common with public transport best path problems. [15] introduces the the time-dependent orienteering problem (TDO) which adds public transport support to the OP. Figure 4 shows how the TDO evolves from combining TDSP and OP.

Figure 4: Dimensions of Shortest Path Problem Extensions



Shortest Path	Time-dependent shortest path
Orienteering Problem	TDO

In further steps, it should be extended, to also account for opening and closing times of events and locations (TDOPTW) and to multiple constrained attributes (MCTDOPTW).

4. CONCLUSION AND OUTLOOK

We introduced ROSE, a mobile system for recommending events and planning routes. For considering arbitrary user preferences when using A*-like search, we developed a two step method. This was only the first step towards solving the more complex problems in personal navigation.

Our next steps are the formulation of the TDOPTW problem and constructing an algorithm for it. After that, we want to extend this problem by additionally considering multiple constrained attributes. It is to be studied, if our proposed two step method for incorporating non monotone and non optimistic multi dimensional user preferences can be integrated in an algorithm to solve these problems.

5. REFERENCES

[1] Garcia, A., Vansteenwegen, P., Souffriau, W., Liniza, M. T., Arbelaitz, O. 2009 Iterated Local Search applied to the Multi-Constrained Team Orienteering Problem with Time Windows, submitted to Computers & Industrial Engineering

[2] Bellotti, V. et al., 2008. Activity-Based Serendipitous Recommendations with the Magitti Mobile Leisure Guide. CHI 2008 Proceedings. DOI=http://doi.acm.org/10.1145/1357054.1357237

[3] Ricci, F., Nguyen, Q. NContext-Aware Recommendations in the Mobile Tourist Application COMPASS. Critique-based Mobile Recommender Systems. ÖGAI Journal, 24(4): 19-25. ÖGAI Pres

[4] van Setten, M., Pokraev, S., Koolwaaij, J. 2004. Context-Aware Recommendations in the Mobile Tourist Application COMPASS. Adaptive Hypermedia and Adaptive Web-Based Systems, pp. 235-244.

[5] Maruyama, A., Shibata, N., Murata, Y., Yasumoto K., Ito M. 2004. P-Tour: A Personal Navigation System for Tourism. Proc. of 11th World Congress on ITS

[6] Völkel, T., Weber, G. 2008. RouteCheckr: personalized multicriteria routing for mobility impaired pedestrians. Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility. pp. 185-192

[7] 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, pp. 697-706

[8] Ding, B., Xu Yu, J., Qin, L. 2008. Finding Time-Dependent Shortest Paths over Large Graphs. EDBT Proceedings. DOI = http://doi.acm.org/10.1145/1353343.1353371

[9] Huang, R. 2007. A Schedule-based Pathfinding Algorithm for Transit Networks Using Pattern First Search. Geoinformatica, Volume 11, Issue 2, pp. 269 - 285

[10] Fischetti, M., Salazar, J.J., Toth, P. 1998. Solving the orienteering problem through branch-and-cut. INFORMS J. Comput., 10:133–148.

[11] Zenker, B. MADL – Decision based programming language for human computer interaction. To appear.

[12] Tumas, G., Ricci, F. 2009. Personalized Mobile City Transport Advisory System. ENTER Conference 2009

[13] Meyer, S., Vaupel, T. 2008. WI-FI Coverage and Propagation for Localization Purposes in Permanently Changing Urban Areas. Proceedings, IADIS International Conference Wireless Applications and Computing

[14] Vansteenwegen, P., Souffriau, W., Garcia, A. 2009. Personalised Tourist Guide: Multi-Constraint Team Orienteering Problem with Time Windows. Proceedings of ORBEL2009. 103:650-669

[15] Fomin, F. V., Lingas, A. 2002. Approximation algorithms for time-dependent orienteering, Information Processing Letters. Elsevire, 83:57-62