On Extensibility and Personalizability of Multi-Modal Trip Planning

Xudong Liu
Department of Computer Science
University of Kentucky
329 Rose Street
Lexington, KY 40506
USA

PARC People
Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304
USA

Abstract

Trip planning is a practical task that has drawn extensive attention from the AI planning and scheduling community and the industrial/commercial sectors. In this paper, we consider the setting of the *multi-modal* trip planning, where users can exploit different transportation modes, such as walking, biking, public transit, and taxi. In such a context, it would be of benefit if the user was able to extend the *cost base*, including traveling time and fare, of the planner, and to personalize the planner according to her own constraints and preferences. To this end, we designed and developed a hipergraph-based multi-modal trip planner that allows users to upload auxiliary cost metrics (e.g., crime rates), to specify constraints as a theory in the *linear temporal logic*, and to express preferences as a *preferential cost function*.

Introduction

Trip planning, as an application of planning and scheduling, has seen substansive implementations by researchers and developers(Bast et al. 2015). Some of the planning systems are *multi-modal*; that is, combining distinct transportation modes, the trip planners compute optimal routes from sources to destinations. This notion of "optimality" generally refers to the computed routes having minimal total time or total fare.

However, in the eyes of a user, it may be more faceted than just "fastest" or "cheapest." For instance, for a college student who specifies that she will only walk or take public transit in a trip from Palo Alto to San Francisco, the computed plan is not necessarily the fastest (e.g., taking a cab could be faster) or the cheapest (e.g., walking all the way has no fare). This happens when a user tells the planner her hard constraints, called *constraints*. The planner then needs to either satisfy the constraints in the search process or return failure because they are over-restrictive. Moreover, the user might want to further customize the planner by describing soft constraints, called *preferences*. For example, an agent wants to travel from school to downtown, and prefers biking to taking the bus. Thus, a trip with more biking than bus may be considered better for the agent than the one with more bus than biking. In this case, the planner will need to accommodate user preferences whenever possilbe in the search of optimal solutions.

Representing and reasoning about constraints and preferences are fundamental to decision making in automated planning and scheduling in artificial intelligence. Son and Pontelli presented a declarative preference language, $\mathcal{PP}$, to represent ordinal preferences (e.g., basic desires, atomic preferences and general preferences) between trajectories of a planning problem(Son and Pontelli 2004). Bienvenu et al. proposed a first-order preference language, $\mathcal{LPP}$, extending $\mathcal{PP}$ by allowing the specification of quantified qualitative preferences(Bienvenu, Fritz, and McIlraith 2011). Modesti and Sciomachen introduced an ad hoc utility function for weighing the arcs both with their time and fare(Modesti and Sciomachen 1998). This relationship between time and fare in the setting of transportation was studied, and economic models capturing the trade-off between the two was presented(Antoniou, Matsoukis, and Roussi 2007).

However, relatively limited effort has been devoted to designing and implementing real-world multi-modal trip planners that captures user constraints and preferences over the *cost base*, possibly extended from the user with auxiliary *cost metrics*, such as crime rates and pollution statistics.

Using a high-performance graph search engine(Zhou and Hansen 2011), we designed and implemented a multi-modal trip planner that uses pure graph-search. This allows us to flexibly combine various modes (e.g., walk, bike, car, public transit, and taxi) and declaratively change constraints and preferences. The planner allows the user to upload *new* mapping data over which to express constraints and preferences. For instance, a user might upload a map of crime in the city, and ask the trip planner to avoid areas where crime is frequent. Then, the planner takes constraints expressed in *linear temporal logic* to constrain the search space (e.g., never bike after transit, and never walk through bad neighborhoods). Finally, the planner uses a weighted sum, called *preferential cost function*, over several cost metrics (e.g., time spend biking, fare on public transit, and overall crimes walking through) which can be re-weighted based on different user preferences.

Extensibility

Allowing users to upload their own data sets of interests is an important step towards customization of a trip planner.

We designed a framework where a user can upload auxiliary cost metric data (e.g., crime statistics and pollution data) into the planner, and the planner will compute an optimal route accordingly.

The user-created data are the auxiliary data that is represented as pairs of latitude and longitude degrees. To merge these lat-long pairs into the planner, we performed a neighborhood search to calculate the total score of auxiliary data for each lat-long pair already in our planner. It might be of strong interest to some user for our planning system to take care of criminal statistics so that some level of safety of the resulting routes is guaranteed. For instance, a user traveling through the downtown area of San Francisco around midnight may want to upload a data set of crimes (cf. Figure 1), and express her constraints and preferences in hope of a preferred trip plan.

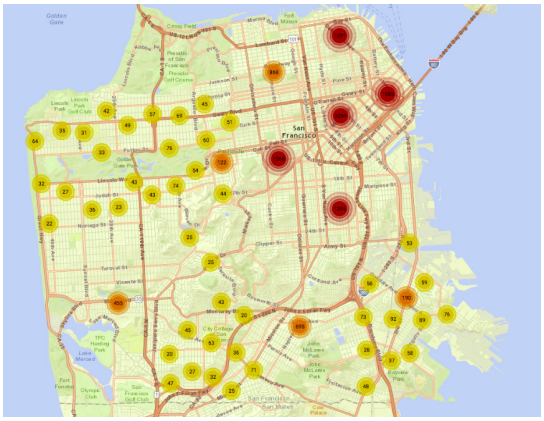

Figure 1: Crime rates in San Francisco

Formally, let $\mathcal{U}$ be the uploaded auxiliary dataset of lat-long pairs, $N = (x_N, y_N)$ a node in our planner and r the effective radius. We first compute the set $\mathcal{P} \subseteq A$ of auxiliary points such that, for every node $P \in \mathcal{P}$, the Euclidean distance between N and P is within r. Then, the score of auxiliary data for N with respect to $\mathcal{P}$ and r, denoted by $S(N, \mathcal{P}, r)$, is computed as follows.

$$S(N, \mathcal{P}, r) = \sum_{P \in \mathcal{P}} \left(1 - \frac{\sqrt{(x_N - x_P)^2 + (y_N - y_P)^2}}{r}\right)$$

Now we turn to Figure 2 for an instance to show how auxiliary data are integrated into the graph using the equation above. In Figure 2, we have the green nodes denoting the graph nodes, and the red nodes the new auxiliary nodes that represent locations of criminal events. For node N, there are three red nodes within its neighborhood of radius of r; thus, we have $\mathcal{P} = \{P_1, P_2, P_3\}$. We assume r is 100 feet, and the distances from N to P_1, P_2 and P_3 are 25, 80 and 90 feet, respectively. Then, the crime score of $S(N, \mathcal{P}, r)$ is 1.05.

Personalizability

Personalizability consists of two aspects: constraints and preferences. From the viewpoint of the planner, constraints, also referred to as hard constraints, are statements that the planner has to satisfy during the planning process; whereas preferences, also called soft constraints, are specifications that the planner will need to optimize. We formulated constraints using linear temporal logic (LTL) and preferences as a preferential cost function (PCF), and implemented our planner leveraging the widely-used graph search algorithm the A*.

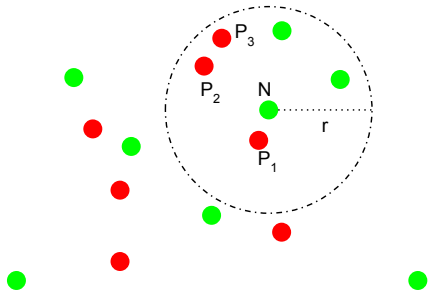

Figure 2: Integrating uploaded data

Constraints

As constraints in the setting of trip planning are often declarative and temporal, our choice of LTL is straightforward. We now give a brief review of linear temporal logic (LTL). Let f be a propositional formula over a finite set L of Boolean variables. LTL formulas are defined recursively as follows.

$$\varphi = f | \varphi_1 \wedge \varphi_2 | \varphi_1 \vee \varphi_2 | \neg\varphi | \bigcirc \varphi | \Box\varphi | \Diamond\varphi | \varphi_1 \mathcal{A} \varphi_2 \quad (1)$$

Note that we have $\varphi_1 \mathcal{A} \varphi_2$, and it means that "$\varphi_2$ holds right after φ_1 holds."

A natural constraint in trip planning could be "In this trip I will not drive a car after biking or taking the public transit." In LTL, such constraint can be translated into an LTL formula

$$\psi = ((M = b) \vee (M = p)) \, \mathcal{A} \, (\Box(\neg(M = c))).$$

As the actions in trip planning is limited to taking different transportation modes, in our definition of the semantics of LTL these actions are subsumed into the interpretations of L, or *states*. The semantics of LTL is defined with regard to trajectories of states. Let σ be a trajectory of states $S_0, a_1, S_1, \ldots, a_n, S_n$, and $\sigma[i]$ a suffix $S_i, a_{i+1}, S_{i+1}, \ldots, a_n, S_n$. We have

$$\begin{aligned}
\sigma &\models f \ \textit{iff}\ S_0 \models f, \\
\sigma &\models \varphi_1 \wedge \varphi_2 \ \textit{iff}\ \sigma \models \varphi_1 \textit{ and } \sigma \models \varphi_2, \\
\sigma &\models \varphi_1 \vee \varphi_2 \ \textit{iff}\ \sigma \models \varphi_1 \textit{ or } \sigma \models \varphi_2, \\
\sigma &\models \neg\varphi \ \textit{iff}\ \sigma \not\models \varphi, \\
\sigma &\models \bigcirc\varphi \ \textit{iff}\ \sigma[1] \models \varphi, \\
\sigma &\models \Box\varphi \ \textit{iff}\ \forall 0 \leq i \leq n (\sigma[i] \models \varphi), \\
\sigma &\models \Diamond\varphi \ \textit{iff}\ \exists 0 \leq i \leq n (\sigma[i] \models \varphi), \\
\sigma &\models \varphi_1 \mathcal{A} \varphi_2 \ \textit{iff}\ \forall 0 \leq i < n (\textit{if } \sigma[i] \models \varphi_1, \sigma[i+1] \models \varphi_2).
\end{aligned}$$

For example, we are given an LTL constraint ψ and three trajectories (σ_1, σ_2 and σ_3) as shown in Figure 3. Clearly, we have $\sigma_1 \models \psi$ because, after public transit in S_2 and S_3, traveling by car has never been taken place. Moreover, we have $\sigma_2 \not\models \psi$ because we have $M = c$ hold in S_6 and S_7 after having $M = p$ hold in S_2 and S_5. Finally, we know $\sigma_3 \models \psi$, as the mode is always neither biking nor public transit.

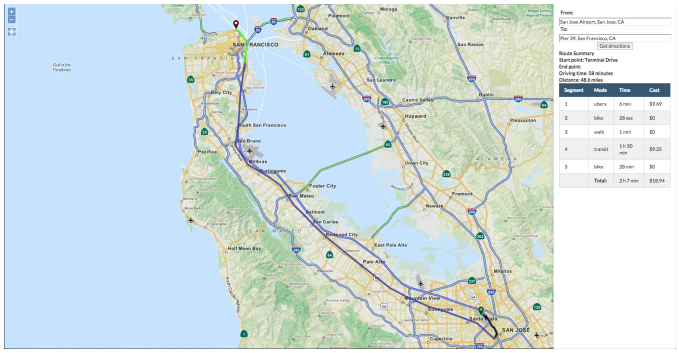

Figure 3: State transition diagram

Preferences

A state is described as a set of *state variables*. The state variables of a state S include the transportation mode M that led to S, time spent T so far per mode (e.g., T_{bike} for biking and T_{public} for public transit), fare D spent so far per mode (e.g., D_{gas} for driving and D_{taxi} for taking a cab), and variables related to the auxiliary data once uploaded. These extra data related variables are metrics such as the sum (A_{sum}), the maximum (A_{max}), the minimum (A_{min}), and the average (A_{avg}) data along the path.

We focused on weighted functions over state variables and designed the cost function, called preferential cost function (PCF), that guides the graph-based search engine in our trip planner as follows.

$$\begin{aligned} PCF(S) =& \beta_1 * (\alpha_1 \cdot T_{walk} + \alpha_2 \cdot T_{wait} + \ldots) \\ &+ (D_{gas} + D_{public} + \ldots) \\ &+ \beta_2 * (A_{sum} + \ldots), \end{aligned} \tag{2}$$

where α_i are real numbers representing the relations among different time pieces, and β_1 (β_2) is the ratio that essentially describes how much in dollars a user would pay to save an hour (an auxiliary data, respectively).

Preference Elicitation To gather these coefficients (α_i and β_i) in our PCF, we designed interface to elicit these numbers from the user.

Reasoning with Constraints and Preferences

We leveraged the widely-used A* search algorithm on top of our high-performance graph search engine (cf. Figure 4). The A* algorithm incorporates the following cost function.

$$f(S) = g(S) + h(S), \tag{3}$$

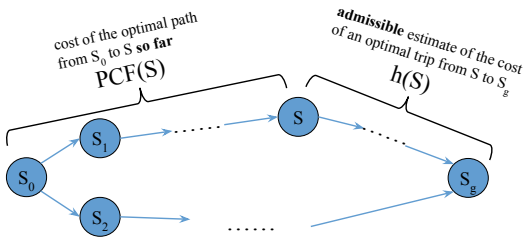

Figure 4: Adjusted A*

where $g(S)$ is the cost of an optimal trip from the initial state to S, and $h(S)$ is an admissible estimate of the cost of an optimal trip from S to goal.

To prune the search space, we check satisfiability of the temporal constraints in LTL at expansion of the search tree. To guide the search engine, we set $g(S) = PCF(S)$ and $h(S)$ the minimum estimate among all available modes in S.

Implementation and Results

System Architecture

Results

First, we show resulting route for Alice in Figure 6.

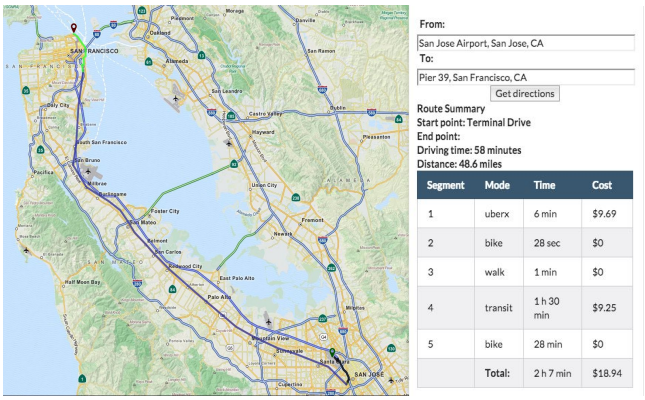

Figure 6: Resulting route for Alice

Second, we show resulting route for Bob in Figure 7.
Third, we show resulting route for Cal in Figure 8.

Conclusion and Future Work

References

[Antoniou, Matsoukis, and Roussi 2007] Antoniou, C.; Matsoukis, E.; and Roussi, P. 2007. A methodology for the estimation of value-of-time using state-of-the-art econometric models. *Journal of Public Transportation* 10(3):1.

[Bast et al. 2015] Bast, H.; Delling, D.; Goldberg, A.; Müller-Hannemann, M.; Pajor, T.; Sanders, P.; Wagner, D.; and Werneck, R. F. 2015. Route planning in transportation networks. *arXiv preprint arXiv:1504.05140*.

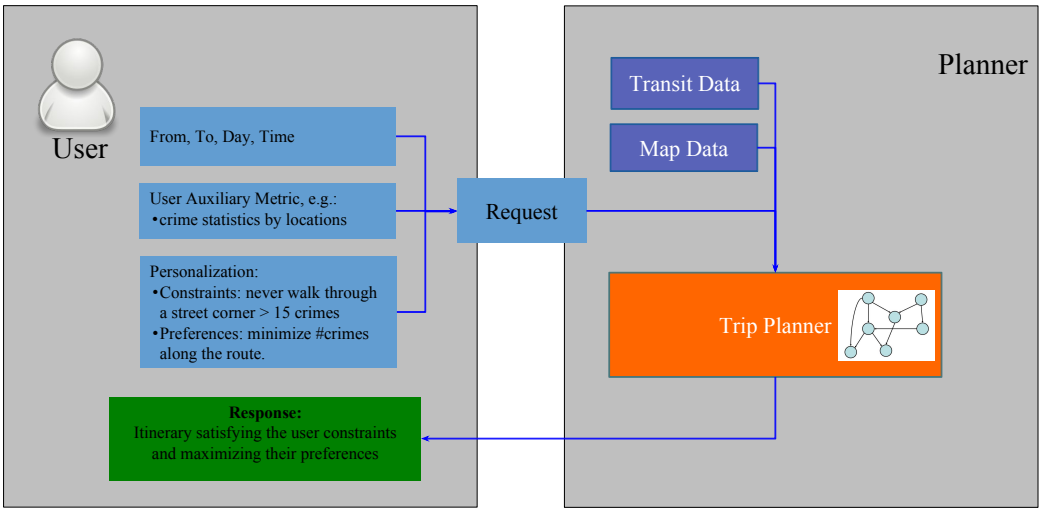

Figure 5: System Overview

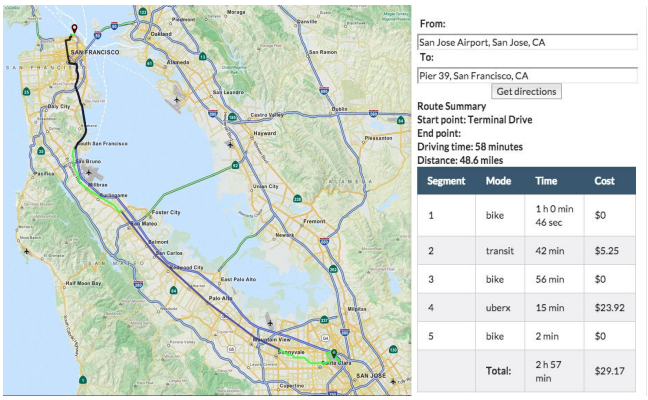

Figure 7: Resulting route for Bob

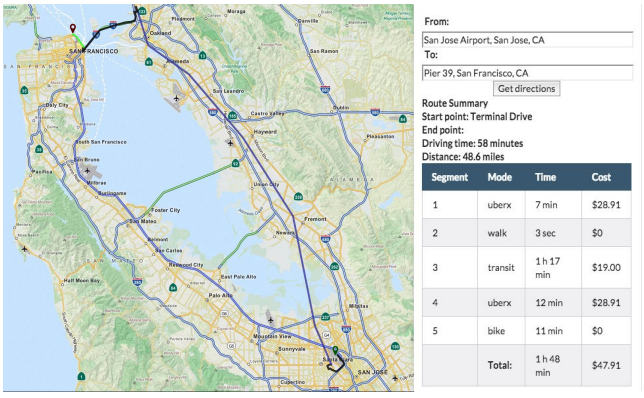

Figure 8: Resulting route for Cal

[Bienvenu, Fritz, and McIlraith 2011] Bienvenu, M.; Fritz, C.; and McIlraith, S. A. 2011. Specifying and computing preferred plans. *Artificial Intelligence* 175(7):1308–1345.

[Modesti and Sciomachen 1998] Modesti, P., and Sciomachen, A. 1998. A utility measure for finding multiobjective shortest paths in urban multimodal transportation networks. *European Journal of Operational Research* 111(3):495–508.

[Son and Pontelli 2004] Son, T. C., and Pontelli, E. 2004. Planning with preferences using logic programming. In *Logic Programming and Nonmonotonic Reasoning*. Springer. 247–260.

[Zhou and Hansen 2011] Zhou, R., and Hansen, E. A. 2011. Dynamic state-space partitioning in external-memory graph search. In *ICAPS*.