On Extensibility and Personalizability of Multi-Modal Trip Planning

Xudong Liu
School of Computing
University of North Florida
Jacksonville, FL
xudong.liu@unf.edu

Christian Fritz*
Savioke Inc.
San Jose, CA
cfritz@savioke.com

Matthew Klenk
PARC, a Xerox Company
Palo Alto, CA
klenk@parc.com

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, an application of planning and scheduling, has seen substantive 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 possible in the search of optimal solutions.

In the work by Yang et al. (Yang et al. 2009), a pilot study was conducted to suggest favored transport modes among the population in the Lisbon Metropolitan Area. The study includes a survey involving 150 respondents, sampled to roughly represent the socio-economic aspects of the local population. Their results revealed that at least 72% of the population picked multiple travel modes (e.g., bus combined with heavy modes including subway, train and ferry) over singular travel modes (e.g., private car, carpool and taxi). The results also presented that almost half of the population had some constraints on traveling time (e.g., departure times to/from work). Furthermore, the pilot survey suggested correlations between travel safety and travel modes, and between environmentally friendliness and travel modes. To this end, our trip planning model is designed and developed in line with these results.

Representing and reasoning about constraints and preferences are fundamental to decision making in automated planning and scheduling in artificial intelligence. Label-constrained multi-modal planning framework was proposed (Barrett et al. 2008; Dibbelt, Pajor, and Wagner 2015), where constraints are formal languages over, and limited to, edge labels that are the modalities in the planner. 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, Fritz and McIlraith 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).

*Work was done while at PARC, a Xerox Company.

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. One notable work by Nina et al. (Nina et al. 2016) introduced system *Autobahn* for generating scenic routes using Google Street View images to train a deep neural network to classify route segments based on their visual features. Although *Autobahn* computes scenic routes using computer vision techniques, it does not account for extensibility and personalizability.

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 (i.e., walking, biking, driving, public transit, and taxi) and to declaratively specify constraints and preferences. The planner also allows the user to upload *new* mapping data over which constraints and preferences can be expressed. 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. To handle user constraints, the planner takes constraints (e.g., never bike after transit, and never walk through bad neighborhoods) expressed in *linear temporal logic* to restrain the search space. As with user preferences, the planner uses a *preferential cost function*, a weighted sum 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.

Our paper is organized as follows. In the next section, we present what it means for a planner to be extensible and formally define the method to incorporating new metrics into the planner. In the next section, we discuss the two aspects of personalization in trip planning: constraints and preferences, and how they are represented and reasoned with in the setting of multi-modal trip planning. We then move on to describe the system structure of our graph-search based planner, and show results obtained from our planner in various occasions. Finally, we conclude by outlining some future research directions.

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 safer trip plan. Note that in Figure 1 bad areas are the colored circles, whose integer labels represent numbers of crimes in corresponding areas, the bigger and darker the circle, the worse the neighborhood.

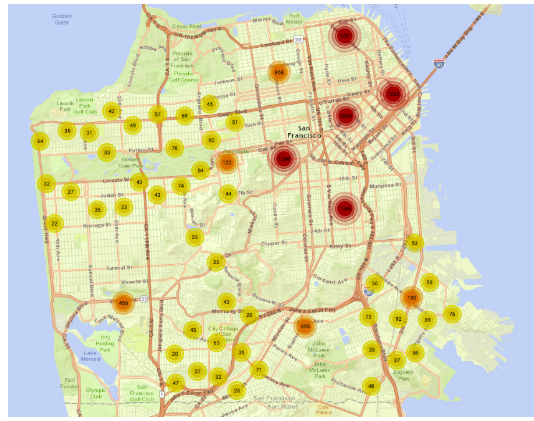


Figure 1: Crime rates in San Francisco

Formally, we denote by $\mathcal{A}$ the set of auxiliary points uploaded by the user, and $\mathcal{N}$ the set of points in our planner. Given a point $N = (x_N, y_N) \in \mathcal{N}$ in our planner, an auxiliary point $A = (x_A, y_A) \in \mathcal{A}$ and an effective radius r, we compute the auxiliary score $S(N, A, r)$ of N contributed by A with respect to r:

$$S(N, A, r) = \begin{cases} 1 - \frac{ED(N,A)}{r} & \text{if } ED(N, A) \leq r, \\ 0 & \text{otherwise,} \end{cases}$$

where $ED(N, A)$ is the Euclidean distance between two points.

Thus, the auxiliary score $S(N, \mathcal{A}, r)$ of N for $\mathcal{A}$ with respect to r can be computed: $S(N, \mathcal{A}, r) = \sum_{A \in \mathcal{A}} S(N, A, r)$.

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. Say, we set radius r to 100 feet, and the distances from N to A_1, A_2 and A_3 are 25, 80 and 90 feet, respectively. For node N, these are the only auxiliary nodes within its neighborhood of radius of r. Thus, the auxiliary score, in this case, the crime score, $S(N, \mathcal{A}, r)$ 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

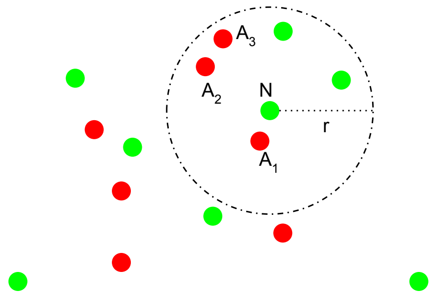


Figure 2: Integrating uploaded data

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*.

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

$$((M = \mathit{bike}) \vee (M = \mathit{public}))\, \mathcal{A}\, (\Box(\neg(M = \mathit{car}))). \quad (2)$$

Note that LTL allows agents to describe constraints over the entirety of the search tree, not just limited to mode labels on edges. For instance, an agent may also express "In this trip I will bike for at least one hour but not more than two," which in LTL would be

$$(\Diamond(T_{bike} \geq 1)) \wedge (\Box(T_{bike} \leq 2)), \quad (3)$$

where T_{bike} denotes the total time spent so far per *bike*.

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

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

For example, we are given an LTL constraint ψ (Equation 2) 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 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.

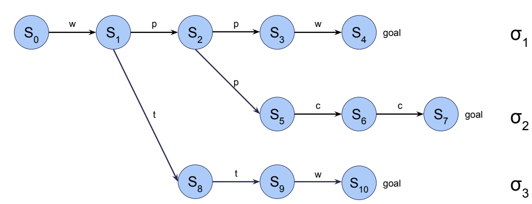


Figure 3: State transition diagram

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 T_M spent so far per mode M (e.g., T_{public} for public transit), fare D_M spent so far per mode M (e.g., 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 assume the fares D_{walk} and D_{bike} are zeros.

We denote by $\mathcal{M} = \{\mathit{walk}, \mathit{bike}, \mathit{car}, \mathit{public}, \mathit{taxi}\}$ the set of transportation modes and focus 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.

$$\begin{aligned}
\mathit{PCF}(S) =& \beta_T \cdot \sum_{M \in \mathcal{M}} (\alpha_M \cdot T_M) + \sum_{M \in \mathcal{M}} D_M \\
&+ \beta_A \cdot A_{sum},
\end{aligned} \quad (4)$$

where α_M is the coefficient of T_M specifying the relationship between M and *car*, and β_T (β_A) is the ratio that describes how much in dollars a user would pay to save an hour (an auxiliary datum, respectively). Note that the PCF can be easily adjusted to cases when no auxiliary dataset or multiple auxiliary datasets uploaded.

Clearly, to any given state our PCF assigns a monetary value, the overall cost that drives our search algorithm in the planner.

Preference Elicitation To gather these coefficients (α_i's and β_i's) in our PCF, we designed an interface to elicit them from the user. The planner asks the user questions and collect answers from the user to derive the coefficients. These questions are as follows.

1. How many hours of driving do you think are equivalent to one hour of walking?
2. How many hours of driving do you think are equivalent to one hour of biking?
3. How many hours of driving do you think are equivalent to one hour of public transit?
4. How many hours of driving do you think are equivalent to one hour of taxi?
5. How much in dollars would you pay to save an hour in traveling?
6. How much in dollars would you pay to avoid an auxiliary datum (e.g., crime or pollution) in traveling?

For instance, Alice, an agent, answers 3, 2, 0.25, 0.5, 20 and 1 to the questions in the list above. Intuitively, the numbers indicate that she prefers public transit the most, followed by taxi, driving, biking and walking, in order. We show how we can now derive α_i's in Equation 4. We start with setting $\alpha_{car} = 1$. Now, since one hour of walking is equivalent to 3 hours of walking, we have $\alpha_{walk} \times 1 = \alpha_{car} \times 3$; hence, we derive $\alpha_{walk} = 3$. Similarly, we have $\alpha_{bike} = 2$, $\alpha_{public} = 0.25$, and $\alpha_{taxi} = 0.5$.

As with the other two coefficients β_1 and β_1, we know one travel hour is worth 20 dollars and one auxiliary event 1 dollar. We then have $\beta_1 \times 1$ hour $= 1 \times 20$ dollars and $\beta_2 \times 1$ aux $= 1 \times 1$ dollars; therefore, we derive $\beta_1 = 20$ dollars/hour and $\beta_2 = 1$ dollar/aux. Indeed, function PCF with the input of time, fare and auxiliary metric pieces boils down to monetary cost, and the planner computes the best path by optimize based on this overall monetary cost in the searching process.

Reasoning with Constraints and Preferences

We leveraged the widely-used A* search algorithm (Hart, Nilsson, and Raphael 1968) on top of our high-performance graph search engine (cf. Figure 4). The A* algorithm incorporates the following cost function.

$$f(S) = PCF(S) + h(S), \tag{5}$$

where $PCF(S)$ is the overall 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. We set $h(S)$ the minimum estimate among all available modes in S. To prune the search space, we check satisfiability of the temporal constraints in LTL at expansion of the search tree.

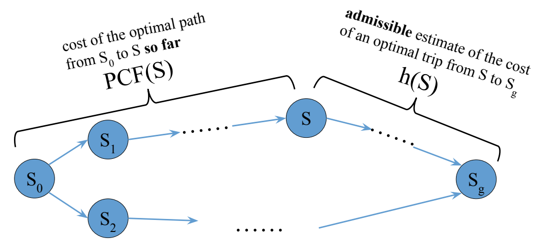


Figure 4: Adjusted A*

Implementation

We designed and implemented a multi-modal trip planning system (cf. Figure 5) based on a high-performance graph search engine. The planner allows user uploads, as well as declarative constraints and preferences. We now describe the structure of the planning system.

The trip planner takes two types of data as input: static data and user-specified request. The static input includes Map Data and Transit Data. Map Data describes the map, a directed graph where nodes are street corners, bus stops and train stations. Transit Data is a set of schedules for the buses and trains On the other hand, a user provides her request, composed of three parts. First, the user enters *from* and *to* locations on the map together with day and time of the start of the trip. Second, the user may upload her auxiliary metric dataset, e.g., crime rates. Lastly, the user specifies her constraints in LTL and preferences as a PCF. For example, the constraint could be "never walk through a bad neighborhood." Given these inputs, our planner computes an optimal path satisfying all the constraints and optimizing the preferences. Note that the request from the user is encapsulated into a JSON object.

For instance, the JSON object for the constraint ψ in Equation 2 is shown in Figure 6.

Results

We present and analyze resulting routes for three agents: Alice, Bob, and Cal. This is assuming no auxiliary metric datasets uploaded so that the agents focus on time and fare. Fixing their where and when information, we show how agents' different constraints and preferences affect their optimal routes computed by our planner. Their where and when are set so that they all plan to travel from San Jose International Airport (SJC) in San Jose, to Pier 39 in San Francisco. Note that the natural constraint ψ in Equation 2 is implicitly imposed on all cases, and that we consider uberX for the taxi mode.

Agent Alice has only one constraint that she does not have a car; thus, driving to her is never available during this trip. This constraint is represented as a LTL formula $\Box(\neg(M = car))$. Per her preferences, Alice provides her thoughts as earlier; that is, public transportation (0.25) is preferred to taxi (0.5), which is better than biking (2), preferred to walking (3). What's more, she

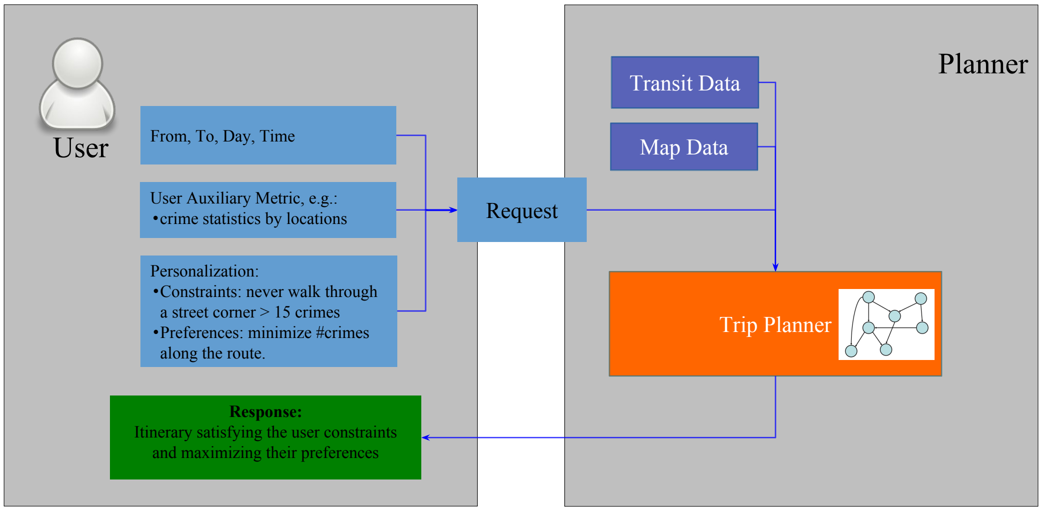


Figure 5: System Overview

```
{
  "constraints": {
    "type":"after",
    "value": [
      {
        "type":"or",
        "value": [
          {"type":"=", "variable":"M", "value":"b"},
          {"type":"=",  "variable":"M", "value":"p"}
        ]
      },
      {
        "type":"always",
        "value": {
          "type":"not",
          "value": {"type": "=", "variable": "M", "value": "c"}
        }
      }
    ]
  }
}
```

Figure 6: JSON object for constraint ψ in Equation 2

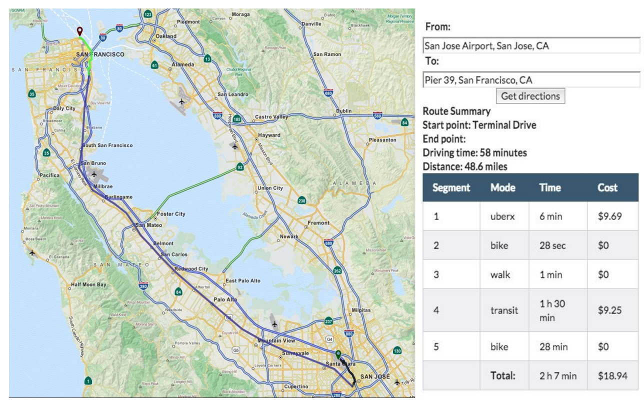


Figure 7: Resulting route for Alice

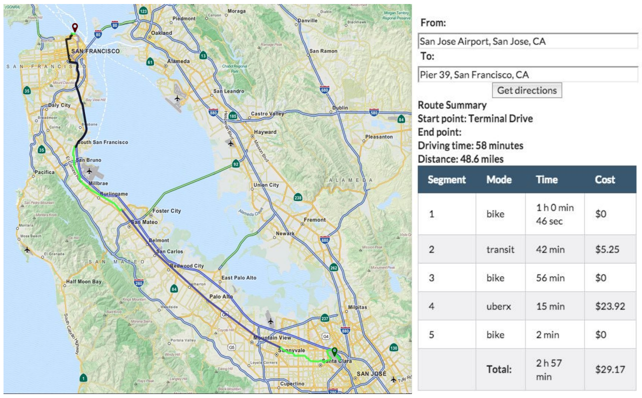
decides that she would sacrifice thirty dollars to save one hour during the trip.

The resulting route for Alice is included in Figure 7. It contains four trainsit modes: *walk*, *bike*, *public* and *taxi*, and takes 2 hours 7 minutes and 18 dollars 94 cents. Alice's travel constraints are clearly met by the result. As for her preferences, most of the time in this route is spent on the public transit that is her most preferred travel mode. We see there is biking for 28 seconds and walking for 1 minute. Although they may not be of of interest from the user's view, the two segments are part of the optimal route the route with minimal combined cost of time and fare among all paths from SJC to Pier 39.

Like Alice, Bob is constrained that he will not drive a car in his travel, and his dollar per hour is thirty. Moreover, Bob makes his mind to have some workout with his bike and dictates that he will bike for between one and two hours. So, Bob's constraint is specified as $(\Box(\neg(M = car))) \wedge (1 \leq T_{bike} \leq 2)$. Then, he expresses his preferences: biking and public transit are the most preferred, next is taxi, and the least preferred is walking. Similarly, he has done so by answering the aforementioned elicitation questions, and here we omit the detailed answers.

The result for Bob is depicted in Figure 8. It spans 2 hours 57 minutes in time with the fare of 29 dollars 17 cents. It is so, seemingly worse than what Alice achieved, only because Bob has the constraint that he will for sure bike for one to two hours (the solution provides under, but very close to, 2-hour biking), and the preferences that put biking the most satisfying mode.

Finally, as part of his constraints, agent Cal will not use a car in the trip. He also affirms that his budget is

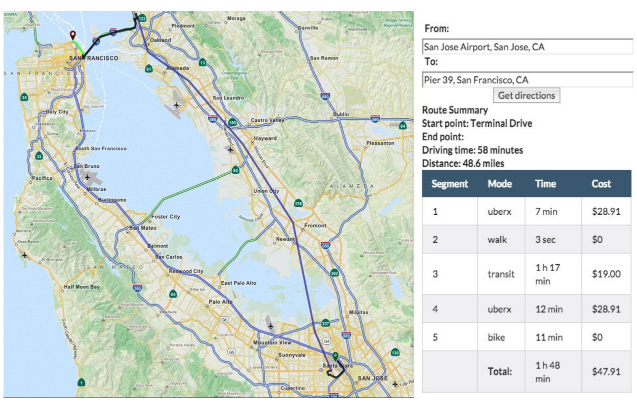


Segment	Mode	Time	Cost
1	bike	1 h 0 min 46 sec	$0
2	transit	42 min	$5.25
3	bike	56 min	$0
4	uberx	15 min	$23.92
5	bike	2 min	$0
	Total:	2 h 57 min	$29.17

Figure 8: Resulting route for Bob

restricted to 50 dollars. These constraints in LTL are $(\Box(\neg(M = car))) \wedge (D_{total} \leq 50)$. Cal's preferences are that the most preferred are public transit and taxi, and the next preferred are walking and biking that are equivalent, and that his one hour in traveling is as valuable as 500 dollars.

Refer to Figure 9 for the optimal path for Cal. This route, stretching 1 hour 48 minutes, is the most time-saving one compared with the previous two, at the price of 49 dollars 91 cents. This is due to the constraint that Cal can spend up to 50 dollars, as well as his preferences and vale of time being high.



Segment	Mode	Time	Cost
1	uberx	7 min	$28.91
2	walk	3 sec	$0
3	transit	1 h 17 min	$19.00
4	uberx	12 min	$28.91
5	bike	11 min	$0
	Total:	1 h 48 min	$47.91

Figure 9: Resulting route for Cal

Auxiliary Metric

When the user of the planner is interested in metrics other than the ones offered already (i.e., time and fare), she might discover new metrics (e.g., crime rates and pollution statistics), upload them into the planner, and retrieve optimal plans taking these metrics into account. One scenario of this approach is the following.

Our agent needs to travel without a car across San Francisco downtown at night. For her, safety is important. Having found the crime statistics for the area, the agent uploads the data as a new auxiliary metric into the map. By specifying that she will never walk through a neighborhood with more than fifteen crimes over the last month, and that she would sacrifice a quarter to avoid one crime incident, the agent tells the planner to come up with a relatively safe route. An example is shown in Figure 10, where the agent needs to start at the east of downtown and travel across the area to arrive at the west side. The computed path is represented by the line colored by black, blue and green, denoting taxi, public transit and biking, respectively. Clearly, this path routes away from crime-heavy areas and achieves optimality in that the combined metrics – time, fare and crime rates, uploaded and personalized by the user – is minimal among all possibilities.

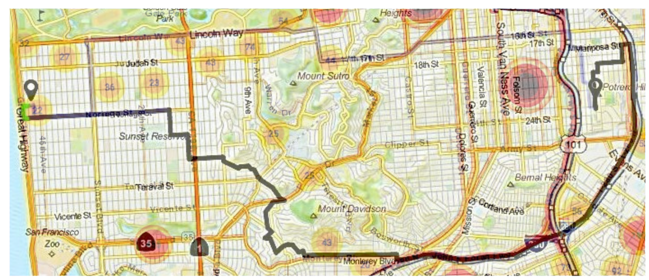


Figure 10: Optimal route considering crime rates

Conclusion and Future Work

In multi-modal trip planning, it is vital for the planner to enable users to integrate new metrics of interest into the existing map, and to express constraints and preferences over which optimal routes are computed. In our work, we designed and built a graph-search based, extensible and personalizable multi-modal trip planner that utilizes linear temporal logic (LTL) and a preferential cost function (PCF) to model user constraints and preferences.

As a note on the future work, we plan to explore techniques in planning for computing multiple routes that are diverse with bounded difference of costs. Continuing this direction, we intend to study the problem of learning the *PCF* coefficients using the observations of the decisions the user made among the computed paths. (It is reasonable to assume that routes picked by the user are preferred to others, and these observations could contribute to the learning problem.) Also interesting is to introduce traffic information into the planner to support real-time multi-agent concurrent trip planning.

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.

[Barrett et al. 2008] Barrett, C.; Bisset, K.; Holzer, M.; Konjevod, G.; Marathe, M.; and Wagner, D. 2008. Engineering label-constrained shortest-path algorithms. *Algorithmic aspects in information and management* 27–37.

[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*.

[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.

[Dibbelt, Pajor, and Wagner 2015] Dibbelt, J.; Pajor, T.; and Wagner, D. 2015. User-constrained multimodal route planning. *Journal of Experimental Algorithmics (JEA)* 19:3–2.

[Hart, Nilsson, and Raphael 1968] Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics* 4(2):100–107.

[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.

[Nina et al. 2016] Nina, R.; Samsonov, P.; Degraen, D.; and Schöning, J. 2016. No more autobahn! scenic route generation using googles street view.

[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.

[Yang et al. 2009] Yang, L.; Choudhury, C. F.; Ben-Akiva, M.; Abreu e Silva, J.; and Carvalho, D. 2009. Stated preference survey for new smart transport modes and services: Design, pilot study and new revision. *Massachusetts Institute of Technology*.

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