Minnesota Department of Transportation

RESEARCH SERVICES

Office of Policy Analysis, Research & Innovation

Bike, Bus, and Beyond: Extending Cyclopath to Enable Multi-Modal Routing

Loren Terveen, Principal Investigator
Department of Computer Science & Engineering
University of Minnesota

July 2013

Research Project
Final Report 2013-19

Your Destination...Our Priority

Technical Report Documentation Page

1. Report No. MN/RC 2013-19	2.	3. Recipients Accession No.
4. Title and Subtitle Bike, Bus, and Beyond: Extending Cyclopath to Enable Multi-Modal Routing		5. Report Date July 2013 6.
7. Author(s) Loren Terveen		8. Performing Organization Report No.
9. Performing Organization Name and Address Department of Computer Science & Engineering University of Minnesota 200 Union St. Minneapolis, MN 55455		10. Project/Task/Work Unit No. CTS Project #2011007 11. Contract (C) or Grant (G) No. (C) 89261 (WO) 193
12. Sponsoring Organization Name and Address Minnesota Department of Transportation Research Services 395 John Ireland Boulevard, MS 330 St. Paul, MN 55155		13. Type of Report and Period Covered Final Report 14. Sponsoring Agency Code
15. Supplementary Notes http://www.lrrb.org/pdf/201319		

16. Abstract (Limit: 250 words)

This project addressed several MnDOT critical issues and strategic directions: easing congestion and improving mobility by making multi-modal transit options more easily accessible. It did so by extending the Cyclopath bicycle route-finding system to support multi-modal routes that combine cycling and transit. This required extensions to the route-finding algorithm – to combine biking and transit appropriately – and user interface – to let users express modal preferences and enhance the visual presentation of routes, e.g., to mark mode shifts. We used the Graphserver open-source software to as the routing engine; by building on existing software, we reduced development effort. We conducted interviews with a number of multimodal transit users to guide the design of the user interface.

17. Document Analysis/Descriptors Bicycling, Bikeways, Market research, Multimodal transportation, Crowdsourcing, Route-finding, Personalization, Wikis		18. Availability Statement No restrictions. Document available from: National Technical Information Services, Alexandria, VA 22312	
19. Security Class (this report) Unclassified	20. Security Class (this page) Unclassified	21. No. of Pages 18	22. Price

Bike, Bus, and Beyond: Extending Cyclopath to Enable Multi-Modal Routing

Final Report

Prepared by:

Loren Terveen

Department of Computer Science & Engineering
University of Minnesota

July 2013

Published by:

Minnesota Department of Transportation
Research Services
395 John Ireland Boulevard, Mail Stop 330
St. Paul, Minnesota 55155

Table of Contents

1. Introduction 1
2. Route finding in Cyclopath: From Cycling-specific to Multi-modal 2
    Graphserver 2
3. Requirements Revisited 4
4. Multi-Modal Interface Design 5
5. User Evaluation 8
    Results 8
6. Deployment and Use 11
7. Summary 12

List of Figures

Figure 1: Multi-modal route request interface 5

Figure 2: Step-by-step instructions 6

Figure 3: Map display of a route, including a zoomed-in section showing additional details obtainable by "mousing over" the initial transit stop. 7

Figure 4: Redesigned multi-modal route finding interface 10

Figure 5: Multi-modal route display 11

1. Introduction

This report provides a comprehensive overview of the technical work that was done as part of this project and the results of deploying multimodal routing within the Cyclopath route finder. In the summer of 2011, we implemented and deployed a multimodal routing algorithm and updated the Cyclopath user interface to enable users to request multimodal routing. Since then, about 15% of route requests have been for multimodal routes.

2. Route Finding in Cyclopath: From Cycling-specific to Multi-modal

The original Cyclopath route-finding algorithm was custom code developed by the Cyclopath team and tailored for bicycle route-finding. The code implements the A* algorithm. Rather than extend our custom code to do multi-modal routing, we choose to use and modify existing open-source multi-modal routing software, thus conserving our personnel time and effort.

Graphserver

Graphserver is an open-source multimodal routing backend. It uses Dijkstra's algorithm to find the shortest path from a starting point to an ending point through multiple linked graphs: in our case, one graph represents the Twin Cities (Metro Transit) bus network, and the other graph represents the Twin Cities Metro Area road and bicycle trail network. The software was designed to work the General Transit Feed Specification (GTFS), which was originally designed by Google, and the OpenStreetMap format, but we adapted it to work with the Twin Cities GTFS and Cyclopath's sql street database. Graphserver's core code is written in C, which makes it efficient. This is an advantage over the original Cyclopath code, which is written in Python. However, Graphserver "wraps" the core functionality with Python, which makes it easy and convenient for us to integrate into the rest of the Cyclopath software.

Some Graphserver features of particular importance for the Cyclopath multimodal implementation are the "slog" option and a set of parameters that fine-tune the non-transit segments of the route. "Slog" is an additional weight added to an undesirable road so that Dijkstra's algorithm will attempt to avoid it even if it is on the shortest path. This is similar to the weight applied to poorly rated roads in Cyclopath's original routing algorithm. Graphserver also uses a class called *WalkOptions* to specify parameters for the non-transit sections; in our case, *WalkOptions* actually correspond to the cycling parts of the route. *WalkOptions* lets the programmer specify useful parameters such as average biking speed, how much faster a cyclist is on a downhill slope, and the number of extra seconds required to make a turn. It also let the programmer specify preferences with parameters such as the maximum cycling distance, aversion to uphill climbs, and aversion to cycling in general. One parameter that deserves special consideration is the transfer penalty, which is incurred every time one boards a bus or train. This has an important purpose beyond the obvious one; this purpose is related to the shortest path Dijkstra's algorithm selects when there is more than one.

Dijkstra's algorithm is guaranteed optimal for a graph with nonnegative edge weights. However, if there is more than one shortest path, it will return the one that starts first. Dijkstra's algorithm is a greedy algorithm, so at every step, the algorithm chooses the path segment that appears best at the time, meaning the one that leaves first. On an all-cycling path this makes perfect sense: if you depart earlier you will arrive earlier. However, this has an undesirable effect for a multi-modal route: it leads to long waits at bus and train stops. For example, if a bus leaves in 45 minutes and the stop is a 5-minute bike ride away, Dijkstra's will suggest that you leave immediately and wait at the stop for forty minutes. Similarly, if you need to transfer to a train that only runs every hour and there are four buses that will get you there in time, Dijkstra's will pick the earliest time. Clearly, this doesn't align with most users' route preferences. Dijkstra's greediness also may result in frivolous transfers. Suppose you are trying to get from point A to point C, and there is a potential intermediate stop at point B. Bus 1 leaves first and goes from A to B. Bus 2 leaves later and goes from A to B to C. Since cyclists in particular dislike transfers (since they have to unload and reload their bike), the preferred route is Bus 2 all the way from A to C. Without fine-tuning, Dijkstra's algorithm will direct you to take Bus 1 to point B, get off, wait, then transfer to Bust 2 to complete the trip to point C. Since both routes arrive at the same time, Dijkstra's is free to choose either as the shortest path. By using the transfer penalty to add weight to the latter path, we can force the algorithm to choose Bus 2 for the entire route.

3. Requirements Revisited

As we noted in our first milestone report, we conducted interviews with 10 multi-modal transit users in August of 2010 to gather requirements for multimodal routing feature in Cyclopath. A key finding was that an effective multi-modal routing system can be quite simple. In particular, when requesting a route, the interview participants had minimal requirements; namely, they wanted some way to specify a desired balance between cycling and transit for the route. And it appeared that the simplest way to do this – specifying a maximum cycling distance – was adequate.

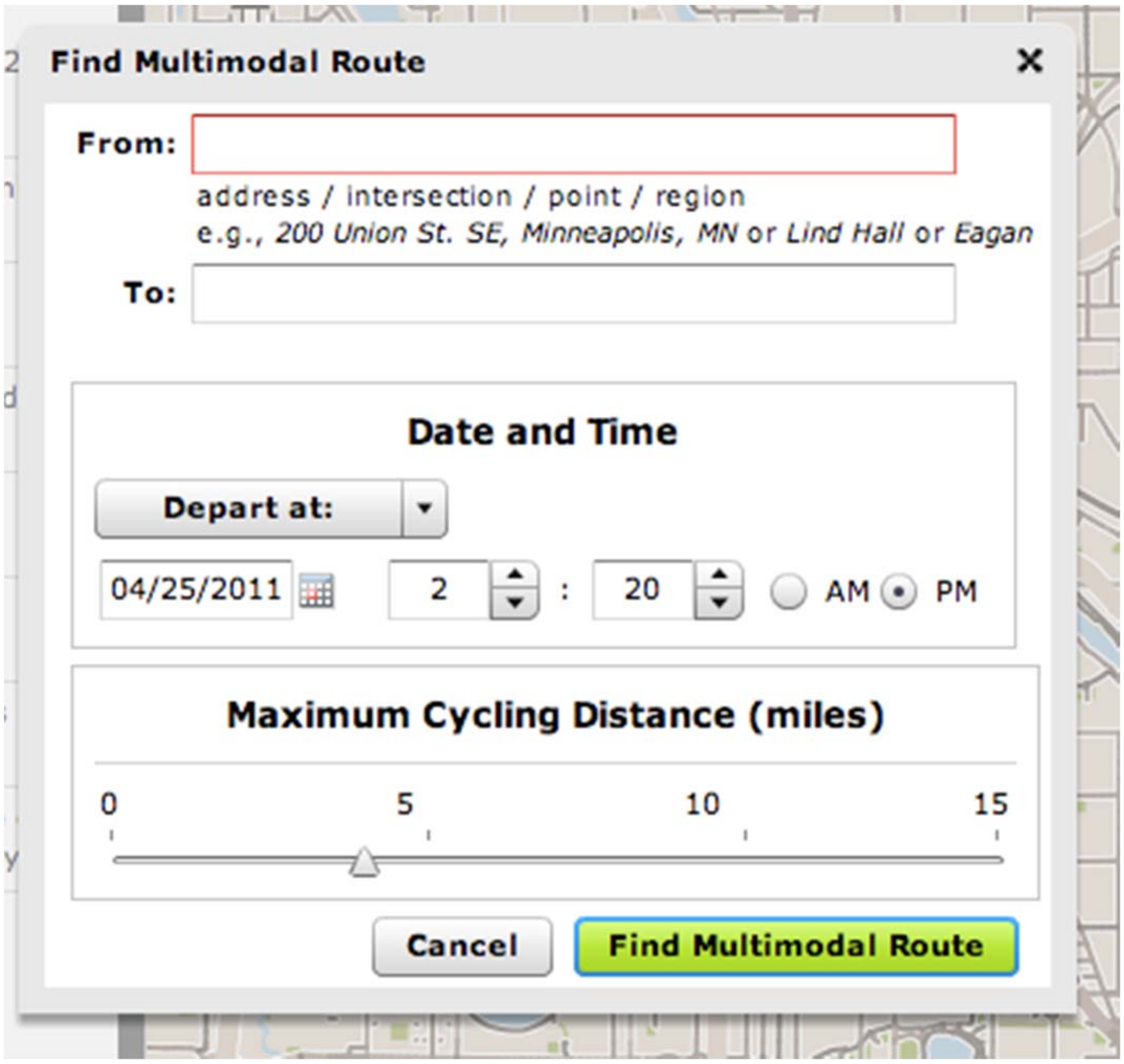
On the other hand, participants did want rather rich information displayed about generated routes, including:

- Road conditions, notably in the winter (icy, snowpacked, etc.);
- Safe or unsafe to bicycle at night;
- Locations of transit stops near the route;
- Locations of bike racks near transit stops.

4. Multi-Modal Interface Design

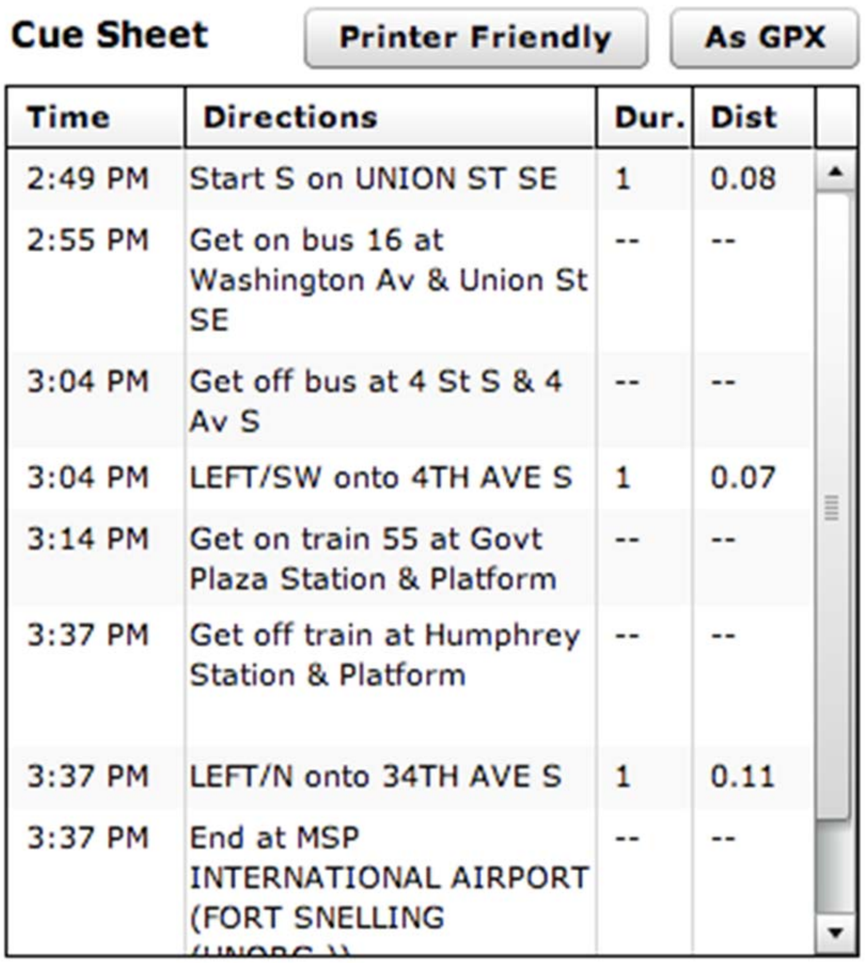
Route Request Interface Prototype.

As noted, the interviews suggested that requesting a route could be a simple process. Therefore, in our prototype interface we implemented only a control to let users specify a maximum desired biking distance for a route. Of course, we will iterate as necessary based on any user feedback we receive. Figure 1 shows the route request interface.


Find Multimodal Route
From:
address / intersection / point / region
e.g., 200 Union St. SE, Minneapolis, MN or Lind Hall or Eagan
To:
Date and Time
Depart at:
Maximum Cycling Distance (miles)
0
5
10
15
Cancel
Find Multimodal Route

04/25/2011 2 : 20 AM [✓ PM]

Figure 1: Multi-modal route request interface



Time	Directions	Dur.	Dist
2:49 PM	Start S on UNION ST SE	1	0.08
2:55 PM	Get on bus 16 at Washington Av & Union St SE	--	--
3:04 PM	Get off bus at 4 St S & 4 Av S	--	--
3:04 PM	LEFT/SW onto 4TH AVE S	1	0.07
3:14 PM	Get on train 55 at Govt Plaza Station & Platform	--	--
3:37 PM	Get off train at Humphrey Station & Platform	--	--
3:37 PM	LEFT/N onto 34TH AVE S	1	0.11
3:37 PM	End at MSP INTERNATIONAL AIRPORT (FORT SNELLING	--	--

Figure 2: Step-by-step instructions

Step-by-step instructions. We wanted step-by-step instructions to be clear and simple: users should be able to see all the information necessary to decide whether or not to take a suggested route. The instructions also had to be specific enough to allow for accurate navigation. For example, to ensure that a user gets off at the right stop, the name of the bus stop is listed along with the estimated time of arrival. Figure 2 shows the step-by-step instructions display.

Map display. We wanted the map display to stand alone. A user should be able to get all the necessary information from the map alone, without looking at the written directions. Thus the locations and names of transit stops should be obvious from the map alone, along with important times (start time, board time, alight time and end time). Similarly, a user should be able to determine the balance between cycling and transit, and which bus or train she should take, from a glance at the map.

We also did not want to clutter the display with too much information. Therefore, some details such as the departure and arrival time for a bus are shown only when a user “mouses over” the bus stop icon.

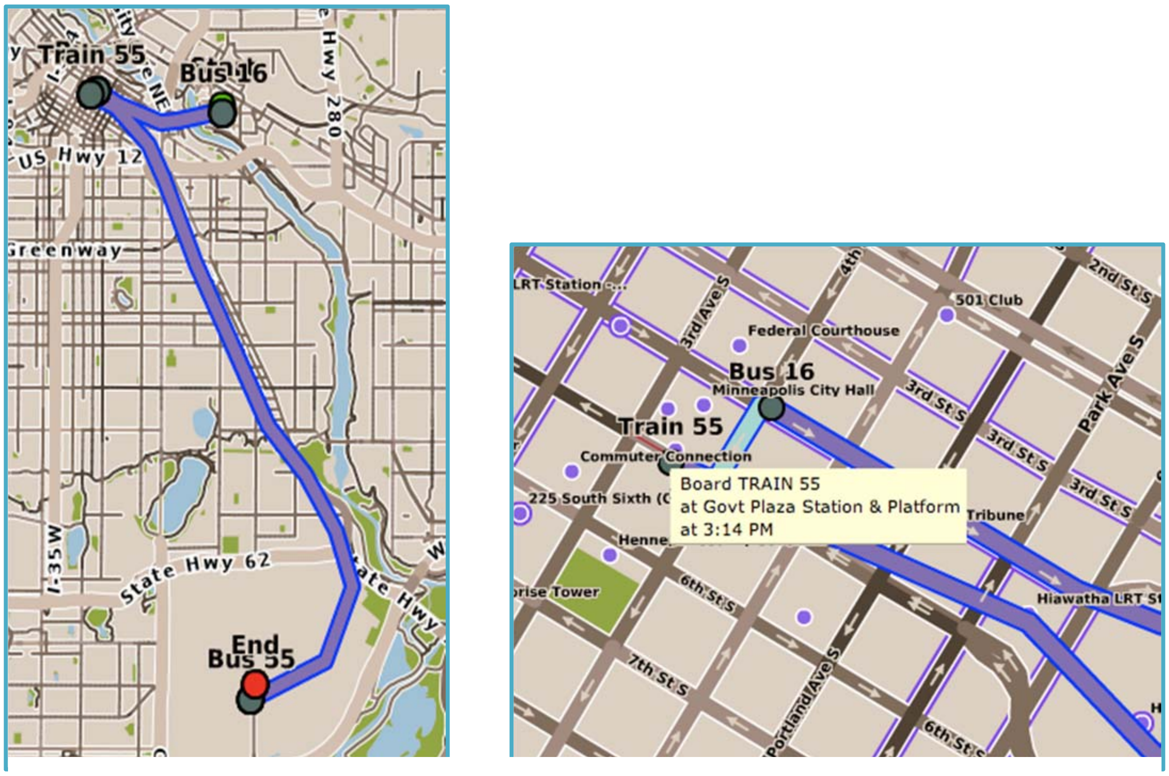


Figure 3: Map display of a route, including a zoomed-in section showing additional details obtainable by "mousing over" the initial transit stop.

Figure 3 shows the visual display of a map, including "zooming in" on a section of the route to illustrate details that can be obtained by mousing over.

5. User Evaluation

In April 2011 we conducted user evaluations of the multi-modal route finding interface. 7 people participated; the participants were frequent cyclists who sometimes used multimodal routes. All had some familiarity with the Cyclopath project; six of the seven had used Cyclopath sometime in the past but only one was a regular user. All had experience using the Metro Transit site to find bus routes. Six were male and one was female.

Goal. The goal of this evaluation was to investigate whether our design concepts were on the right track and identify any essential changes and interface improvements that were needed.

Procedure. Participants were instructed to find three different routes of their choice. Afterwards, we asked which of their routing preferences they had been able to enter; if there were preferences they wished they could enter but were unable to; what information they were able to learn from the written directions and the map; if there was nonessential information displayed; what information they would need to know about a multimodal route to compare it to another multimodal route; and how they would compare the Cyclopath multimodal interface they had just tried with Metro Transit's route generation tool.

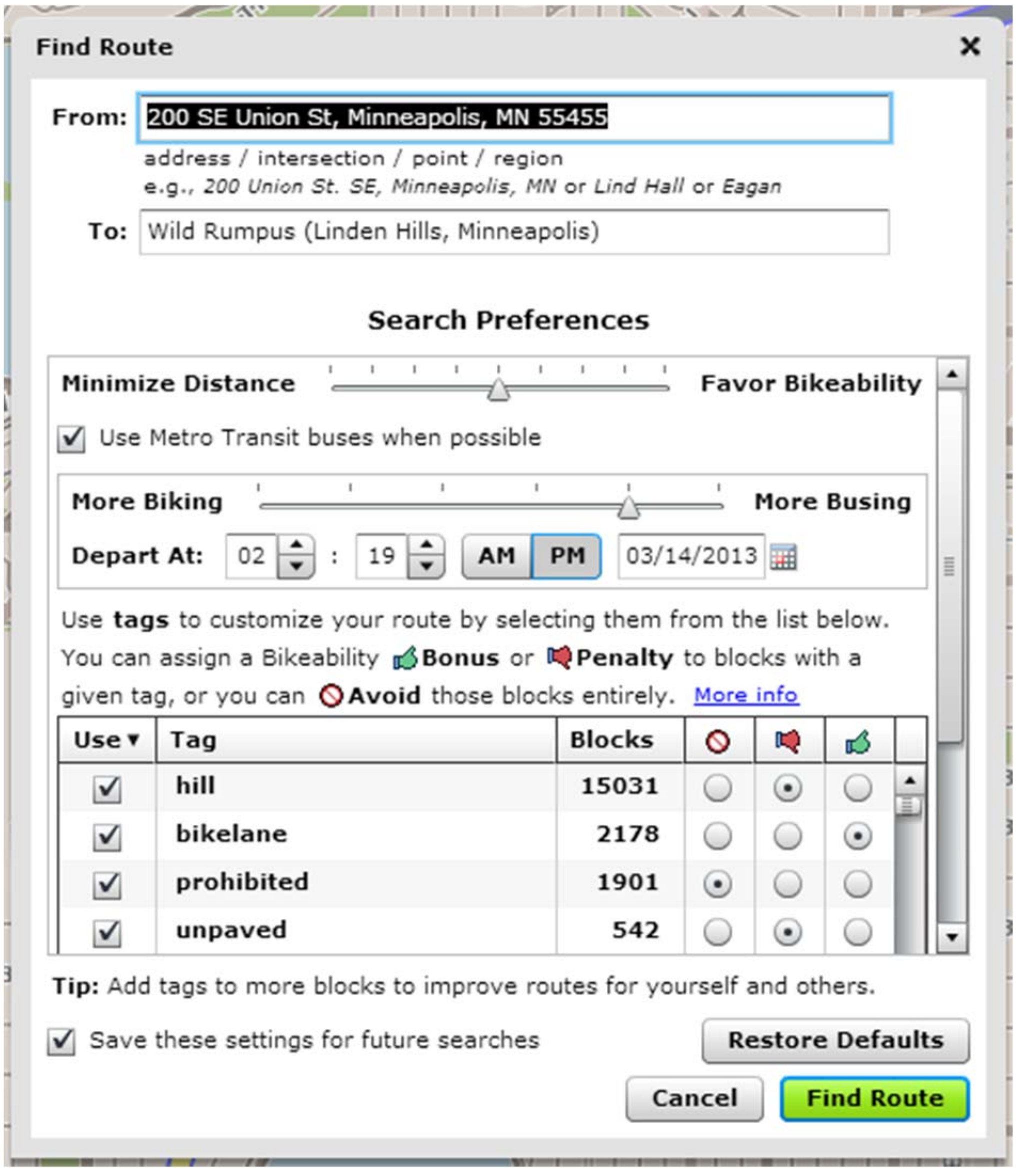
Results

In addition to more than three dozen specific user interface suggestions, the testing revealed user confusion on the function of the "max cycling distance" scrollbar and persistent concerns about the optimality of the suggested route. We made a number of design changes to respond to these findings.

The testers directly suggested some of the possible user interface changes, while we articulated other changes by observing areas of confusion or struggle. Many of the suggestions involved making the step-by-step instructions clearer. About half of the testers avoided looking at the step-by-step instructions entirely, and the other half were frustrated that they couldn't immediately pick up on the transition between transit and bike the way they could see it on the map. The most frequent suggestions were changing the color of the text or background for the transit sections and providing a summary with the most important information such as bus numbers and the amount of time spent waiting during transfers. We collected these and other potential design changes for consideration as we continue to improve the Cyclopath user experience.

All of the participants noticed the different colors of the route lines and understood that different colors indicated different modes. Similarly, the bus number labels were noticeable and clear. However, it was a different story when it came to revealing information on mouse rollover. Only one user discovered that additional information was displayed when the mouse rolled over the start, end, board or alight points along the route. Possible design fixes include increasing the "sensitive" area around the route, making it easier to reveal the information. Or we could automatically popup a window explaining the 'mouse-over-to-get-details' feature the few times a user views a multi-modal route.

The "maximum cycling distance" parameter generated some controversy. We chose not to instruct participants to use the control slider as part of the tasks; this was because we hadn't found the proper balance of the other WalkOptions parameters to make it useful. Nevertheless, we asked about this feature indirectly to probe how participants expected or wanted it to function. There seemed to be a split between those who believed it was a strict maximum and expected the routing system to return the most efficient route with less cycling than the maximum and those who expected the returned route to have approximately that much cycling. Three of the participants never adjusted the slider; three adjusted the slider once; and one adjusted the slider frequently, comparing the returned routes to determine the effect of the slider. Those who tried to use the slider found the results confusing due to the way the multi-modal system was tuned at the time: its parameters were set to greatly prefer riding a bus to cycling. This is partly because riding a bus is generally much faster, so when finding the route that arrives soonest, bus-heavy routes are preferred; this is particularly true since the default cycling speed is set very low at the moment. However, it doesn't explain why the system preferred a half-hour wait for a transfer instead of cycling a mile and a half. An additional complication is that unsafe byways are heavily penalized in the routing algorithm, but waiting for a bus is not penalized. The participants' reactions gave us valuable feedback: most preferred a routing algorithm that included more cycling to one that had long waits at bus stops. Based on these findings, we redesigned the control to specify the balance between biking and busing: we now simply use a slider with "More biking" at one end of the scale and "More busing" at the other end": see Figure 4.



From: 200 SE Union St, Minneapolis, MN 55455

address / intersection / point / region

e.g., *200 Union St. SE, Minneapolis, MN* or *Lind Hall* or *Eagan*

To: Wild Rumpus (Linden Hills, Minneapolis)

Search Preferences

Minimize Distance — **Favor Bikeability**

☑ Use Metro Transit buses when possible

More Biking — **More Busing**

Depart At: 02 : 19 AM [✓ PM] 03/14/2013

Use **tags** to customize your route by selecting them from the list below. You can assign a Bikeability 👍**Bonus** or 👎**Penalty** to blocks with a given tag, or you can 🚫**Avoid** those blocks entirely. More info

Use ▾	Tag	Blocks	🚫	👎	👍
☑	hill	15031	○	●	○
☑	bikelane	2178	○	○	●
☑	prohibited	1901	●	○	○
☑	unpaved	542	○	●	○

Tip: Add tags to more blocks to improve routes for yourself and others.

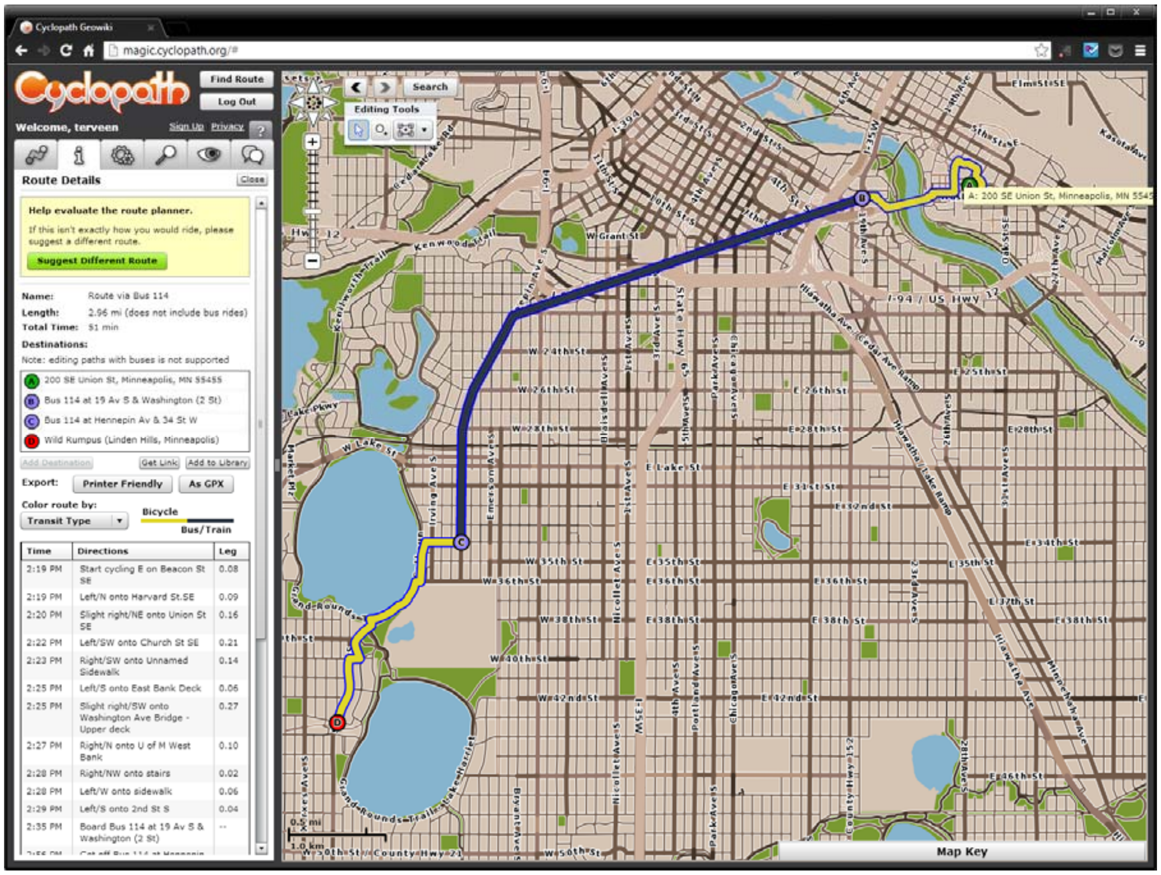
☑ Save these settings for future searches

Figure 4: Redesigned multi-modal route finding interface

Half of the participants were unsatisfied with at least one of the routes returned, audibly wondering if it really was the best option. Problems included routes with long waits at the bus stop during transfers, routes that didn’t seem to take the first available bus, routes that arrive first but have a long in-transit time, routes that don’t take a preferred bus and routes that had a cyclist ride away from their destination to get to a bus stop. We have considered a set of solutions to these problems. One option is add additional preference controls, possibly hidden under an “advanced features” label. Another option is to generate several possible routes using different parameters, as the Metro Transit and Google Maps websites do, and offer them all to the user. We think this is preferable from a user experience point of view, but only if this does not significantly increase the amount of time a user has to wait for a route request to be satisfied.

6. Deployment and Use

We deployed multimodal routing in the public version of Cyclopath in the summer of 2011. The deployed interface is much like that of the prototype, with the modifications described above. Figure 5 below shows the display of a multimodal route request from “200 Union Street S.E., Minneapolis MN” to “Wild Rumpus” (a user-defined point of interest).



Route Details

Help evaluate the route planner.

If this isn't exactly how you would ride, please suggest a different route.

Suggest Different Route

Name:	Route via Bus 114
Length:	2.96 mi (does not include bus rides)
Total Time:	51 min

Destinations:

Note: editing paths with buses is not supported

- A 200 SE Union St, Minneapolis, MN 55455
- B Bus 114 at 19 Av S & Washington (2 St)
- C Bus 114 at Hennepin Av & 34 St W
- D Wild Rumpus (Linden Hills, Minneapolis)

Export: Printer Friendly | As GPX

Color route by: Transit Type — Bicycle / Bus/Train

Time	Directions	Leg
2:19 PM	Start cycling E on Beacon St SE	0.08
2:19 PM	Left/N onto Harvard St.SE	0.09
2:20 PM	Slight right/NE onto Union St SE	0.16
2:22 PM	Left/SW onto Church St SE	0.21
2:23 PM	Right/SW onto Unnamed Sidewalk	0.14
2:25 PM	Left/S onto East Bank Deck	0.06
2:25 PM	Slight right/SW onto Washington Ave Bridge - Upper deck	0.27
2:27 PM	Right/N onto U of M West Bank	0.10
2:28 PM	Right/NW onto stairs	0.02
2:28 PM	Left/W onto sidewalk	0.06
2:29 PM	Left/S onto 2nd St S	0.04
2:35 PM	Board Bus 114 at 19 Av S & Washington (2 St)	--

Figure 5: Multi-modal route display

During the prime bicycle riding season, Cyclopath receives about 100-150 route requests per day. About 15% of these requests are for multimodal routes. This number may seem low; however, given that Cyclopath is well known and is positioned as a route finding system for bicyclists, we do not think this total is low.

7. Summary

We extended the Cyclopath route finding system to enable multimodal – bicycle + transit – routing. We tested the system in a user study and iterated the design based on the results. We then deployed the multimodal enabled version of Cyclopath, and multimodal routes have come to constitute a small but persistent proportion of the route requests we service.