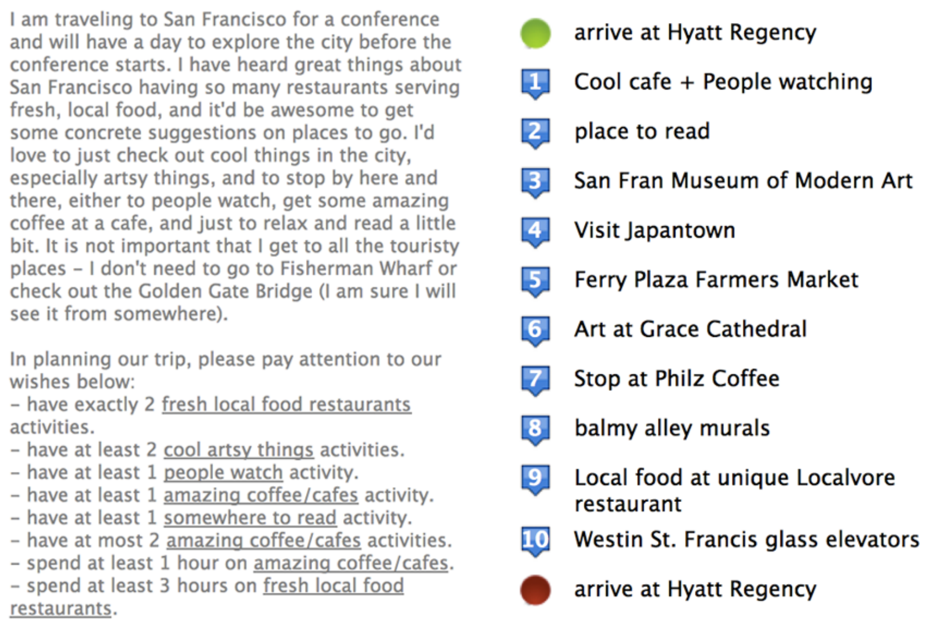
Chapter 3

Human Computation with Global Constraints

Within studies of human computation, an important class of underexplored tasks are those in which the solution must satisfy a set of global requirements. For example, in leveraging the crowd to write an essay, a requester may want to specify requirements on the desired tone, tense, length, structure of arguments, and style of exposition that must hold consistently throughout a piece of writing. Some requirements, e.g., presenting a balanced perspective on a situation, touch upon different components of the essay and depend on the essay as a whole. Similar considerations arise in creative tasks such as graphic design and more mundane tasks such as meeting scheduling.

As good solutions rely on the composition as a whole and are marked by interdependence among solution components, tasks involving global constraints are difficult to decompose into subtasks that can be independently assigned to individuals in the crowd. This raises a significant challenge for human computation algorithms, as such



I am traveling to San Francisco for a conference and will have a day to explore the city before the conference starts. I have heard great things about San Francisco having so many restaurants serving fresh, local food, and it'd be awesome to get some concrete suggestions on places to go. I'd love to just check out cool things in the city, especially artsy things, and to stop by here and there, either to people watch, get some amazing coffee at a cafe, and just to relax and read a little bit. It is not important that I get to all the touristy places – I don't need to go to Fisherman Wharf or check out the Golden Gate Bridge (I am sure I will see it from somewhere).

In planning our trip, please pay attention to our wishes below:
- have exactly 2 fresh local food restaurants activities.
- have at least 2 cool artsy things activities.
- have at least 1 people watch activity.
- have at least 1 amazing coffee/cafes activity.
- have at least 1 somewhere to read activity.
- have at most 2 amazing coffee/cafes activities.
- spend at least 1 hour on amazing coffee/cafes.
- spend at least 3 hours on fresh local food restaurants.

● arrive at Hyatt Regency
1 Cool cafe + People watching
2 place to read
3 San Fran Museum of Modern Art
4 Visit Japantown
5 Ferry Plaza Farmers Market
6 Art at Grace Cathedral
7 Stop at Philz Coffee
8 balmy alley murals
9 Local food at unique Localvore restaurant
10 Westin St. Francis glass elevators
● arrive at Hyatt Regency

Figure 3.1: Planning mission (left) and itinerary (right)

tasks are not amenable to the divide and conquer approach introduced in chapter 2 that is used in most crowdsourcing systems.

As a focal example, consider the problem of crowdsourcing itinerary planning. Planning events such as vacations, outings, and dates often involve an *itinerary* (Figure 3.1), which contains an ordered list of activities that are meant to be executed in sequence over the course of an event. People going on a trip have preferences and constraints over the types of activities of interest (e.g., "I want a coffee break right after lunch"), how long to spend on different activities (e.g., "I want to spend at least 2 hours in parks"), the composition of activities (e.g., "I want to focus on art galleries and museums for the day"), the budget, and the total time available, which define a set of global requirements that an itinerary should satisfy.

Decisions on any particular activity in the itinerary may naturally influence other decisions. As simple examples, spending time on one activity leaves less time for another, and moving to one location introduces distances to other locations.

To handle tasks with global requirements, we introduce in this chapter a *crowdware* design that provides a single workspace in which a crowd of individuals contribute opportunistically based on their knowledge and expertise and the current solution context, and in which the system (indirectly) coordinates the crowd problem-solving effort by focusing the crowd's attention on what needs work. Crowdware takes inspiration from *groupware* [25], which suggest principles and ideas on communication and collaboration within a shared context that help a group to accomplish a joint task. We consider how to apply such principles and ideas to crowd workers, who differ from groups in that individuals may only be briefly involved, may be less willing to spend time grasping the solution context or take meta-level actions, and may not consider the desires of other crowd workers when making decisions.

We focus on itinerary planning as a case study of coordinating a crowd to tackle tasks with global constraints. We introduce a collaborative itinerary planning system called Mobi. Mobi takes a planning mission containing a set of qualitative and quantitative constraints as articulated by the user as input and produces an itinerary that satisfies the mission as output. The crowd participates via a single interface—displaying the current itinerary and a stream of ideas generated thus far—that allows individuals to contribute opportunistically given the current context and to see their contributions incorporated into the solution in real-time. Mobi focuses the crowd's attention on aspects of the evolving plan that needs work by prominently displaying a list of automatically generated *todo items*, which point out violated constraints, provide suggestions on how to address them, and promote activities directed at the refinement of the itinerary.

Mobi allows users to specify their desires and needs in *natural language*, thereby enabling complex constraints and preferences to be expressed and used in the planning process. We present two studies, which show that Mobi's design promotes a collaborative planning environment in which the crowd can effectively produce custom itineraries that satisfy the global constraints stated in user missions.

In the first study, we test the effect of displaying todo items on the rate at which quantitative constraints are resolved by the crowd, and measure the contribution patterns of crowd workers. We find that the display of todo items promotes satisfaction of constraints at a significantly faster rate than when todo items are not displayed, and that the crowd's editing patterns show evidence of both collaboration and opportunistic planning. In the second study, we seek to understand whether the end users believe that crowd-generated itineraries satisfy their stated requirements. Users report that the itineraries contain many activities of interest, mostly or fully satisfy their mission requirements, and are useful for their actual trips.

The chapter is organized as follows. Section 3.1 presents related work. Section 3.2 introduces the Mobi system. Section 3.3 and Section 3.4 describe our two studies. Section 3.5 revisits the elements of Mobi's design and discusses how these elements may in general inform the design of systems that facilitate a crowd to tackle problems involving global constraints. Section 3.6 summarizes our results and presents directions for future work.

3.1 Related Work

Planning can be viewed as an *iterative task* in which workers make successive edits to improve the solution. There has been some attention on iterative tasks in human computation [60], and an interesting recent example is work by Kittur [45] that recruits workers to collaborate in Etherpad to translate a poem. Workers were able to see their edits reflected in real time and could communicate via chat to explain their edits. One difference in Mobi is that Mobi uses its sense of the progress made so far (e.g., how full the itinerary is, which constraints are violated, etc.) to prompt users on what needs work so as to guide the problem-solving process.

Wikipedia can be viewed as an example of a system in which (mostly expert and highly dedicated) contributors write and edit articles to resolve a set of global constraints as defined by Wikipedia's standards. Much like the way todo items are used in Mobi to drive progress, template messages and cleanup tags are used in Wikipedia to alert editors of changes that need to be made to improve an article.[1] Such messages are typically managed by human contributors, whereas in Mobi todo items are managed in an automated manner whenever possible.

Several models have been proposed to describe how people generate plans to achieve goals. The *successive refinement* model advocates a top-down approach, where a high-level goal is decomposed into subgoals iteratively, down to a sequence of elementary actions [81]. In contrast, the planning of many everyday activities (e.g., errands) is often *opportunistic*. In other words, planning decisions happen whenever opportunities arise [30, 44], so that a decision or observation in one part of the plan

[1]See `http://en.wikipedia.org/wiki/Wikipedia:Template_messages/Cleanup`

may suggest new ideas or illuminate problems in a different part of the plan, causing the planner to refocus his attention. Opportunistic planning may involve both top-down and bottom-up processing. For example, in an errand planning experiment, Hayes-Roth and Hayes-Roth [30] found that subjects would start making detailed plans (e.g., sequencing individual errands), and then switch to planning on a more abstract level (e.g., by discovering clusters of errands), and back and forth as they refined the plan. Mobi is designed with the opportunistic planning model in mind, where individuals in the crowd are allowed to contribute freely as they see fit based on their observations of what needs work given the current solution context.

Real-life planning is a difficult problem for computers. Despite advances in automated planning [67], a major challenge is making sense of people's goals, preferences and other "soft" considerations [13]. Currently, the automated planner in Mobi supports workers by automatically checking constraints and computing trip times and routes. In the future, automation may play a more active role in the planning process by learning about different requirements, suggesting activities and their composition in the itinerary, or even detecting and adding important constraints that may have been missed by the requester.

There are several existing commercial systems that allow groups to plan trips for themselves or to ask friends and other members for suggestions. Examples include Gogobot, Triporama, Kukunu, and FriendTripper. Mobi differs from these systems in that it produces not only suggestions for activities, but an itinerary satisfying a set of global requirements. By using todo items, Mobi can also focus the crowd on making contributions where they are most needed.

3.2 Mobi: A System for Crowd Itinerary Planning

Mobi takes a planning mission consisting of preferences and constraints as input, and generates an itinerary by having a crowd plan asynchronously using a shared interface. Workers invited to contribute can view the current plan and all ideas proposed thus far, and make contributions as they see fit. Edits can be made at any time and without restrictions. The itinerary is automatically saved after each change. We now describe Mobi's interfaces for specifying the planning mission and assembling the itinerary, and discuss how these two interfaces support the process of generating itineraries and resolving constraints.

3.2.1 Specifying the Planning Mission

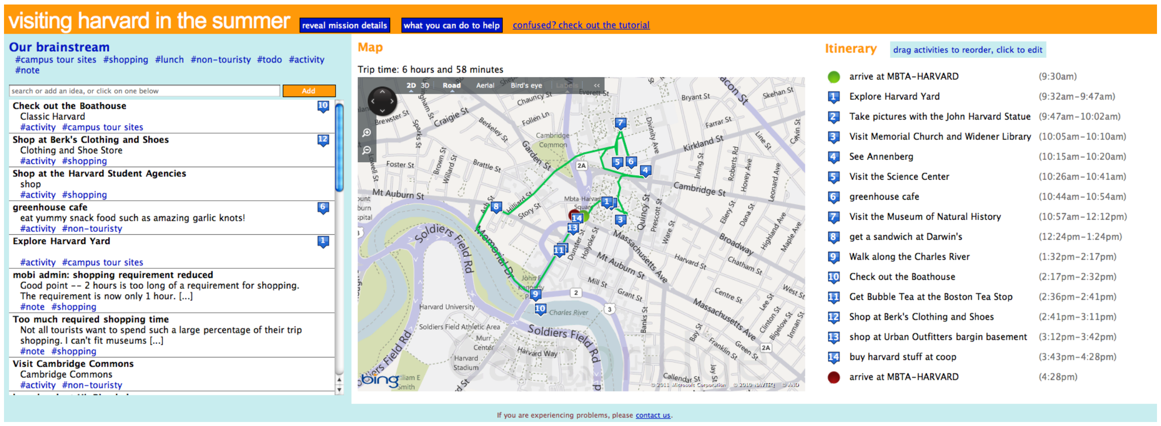
Our target users, also referred to as *requesters*, are people who are interested in planning a trip. To start planning, the requester enters a planning mission using a simple web interface, by specifying the title and description of the trip, start/end locations and times, and whether he or she will use public transit or drive between locations in addition to walking.

Requesters can express two kinds of constraints: *qualitative* and *quantitative*. Figure 3.1 (page 60) shows an example of a planning mission that includes both types of constraints. Qualitative constraints are specified in natural language (e.g., in a paragraph). They can describe, for example, the nature of the trip, what the user hopes to accomplish, and who they are traveling with. Quantitative constraints are specified by creating *categories* using arbitrary natural language phrases (e.g., "cool art," "by the ocean"), and assigning preferences and limitations over categories. One can specify constraints on the number of activities in each category (e.g., "I want to visit up to two *museums*"), as well as the amount of time to spend on activities in each category (e.g., "I want to spend at least two hours on *cool art*"). Such constraints can also be used to express the preferred combination of activities in the plan (e.g., "I want to spend half of my time on activities *by the ocean*, and the other half on activities *in the city*"). In our prototype, the domain-specific language for quantitative constraints allows for constraints encoded in the form of "*I want* {*at most, at least, exactly*} *[number]* {*activities, hours*} *of* $\{cat_1, cat_2, \ldots, cat_n\}$," where cat_i refers to the i-th requester-defined category.

Both qualitative and quantitative constraints contain natural language, and can express "soft" considerations that the computer cannot tackle alone. In addition to these constraints, the system maintains a pair of time constraints, which state that the cumulative duration of the activities in the itinerary should not be greater than, or significantly less than, the duration of the trip specified by the user.

3.2.2 Assembling the Itinerary

Once a requester specifies a planning mission, workers can use Mobi's planning interface to view the mission by clicking on the "reveal mission details" button in the *information panel* (Figure 3.2, on top). The planning interface consists of two key components: the *brainstream* and the *itinerary viewer*.


visiting harvard in the summer
reveal mission details
what you can do to help
confused? check out the tutorial
Add
Map
If you are experiencing problems, please contact us.

Our brainstream
#campus tour sites #shopping #lunch #non-touristy #todo #activity #note

search or add an idea, or click on one below

Check out the Boathouse 10
Classic Harvard
#activity #campus tour sites

Shop at Berk's Clothing and Shoes 12
Clothing and Shoe Store
#activity #shopping

Shop at the Harvard Student Agencies
shop
#activity #shopping

greenhouse cafe 6
eat yummy snack food such as amazing garlic knots!
#activity #non-touristy

Explore Harvard Yard 1
#activity #campus tour sites

mobi admin: shopping requirement reduced
Good point -- 2 hours is too long of a requirement for shopping. The requirement is now only 1 hour. [...]
#note #shopping

Too much required shopping time
Not all tourists want to spend such a large percentage of their trip shopping. I can't fit museums [...]
#note #shopping

Visit Cambridge Commons
Cambridge Commons
#activity #non-touristy

Trip time: 6 hours and 58 minutes

Itinerary drag activities to reorder, click to edit

●	arrive at MBTA-HARVARD	(9:30am)
1	Explore Harvard Yard	(9:32am-9:47am)
2	Take pictures with the John Harvard Statue	(9:47am-10:02am)
3	Visit Memorial Church and Widener Library	(10:05am-10:10am)
4	See Annenberg	(10:15am-10:20am)
5	Visit the Science Center	(10:26am-10:41am)
6	greenhouse cafe	(10:44am-10:54am)
7	Visit the Museum of Natural History	(10:57am-12:12pm)
8	get a sandwich at Darwin's	(12:24pm-1:24pm)
9	Walk along the Charles River	(1:32pm-2:17pm)
10	Check out the Boathouse	(2:17pm-2:32pm)
11	Get Bubble Tea at the Boston Tea Stop	(2:36pm-2:41pm)
12	Shop at Berk's Clothing and Shoes	(2:41pm-3:11pm)
13	shop at Urban Outfitters bargin basement	(3:12pm-3:42pm)
14	buy harvard stuff at coop	(3:43pm-4:28pm)
●	arrive at MBTA-HARVARD	(4:28pm)

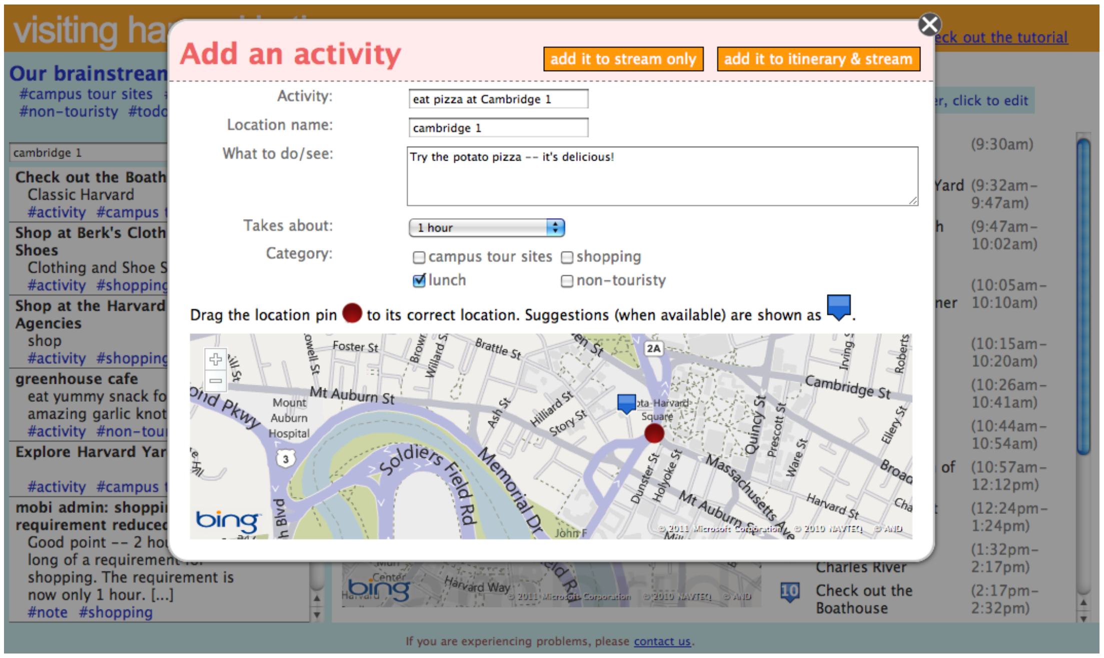
Figure 3.2: The Mobi planning interface consists of the information panel (top), the brainstream (left), and the itinerary viewer (right).

Brainstream

The *brainstream* (Figure 3.2, on left) is a collection of everyone's ideas. An idea can be an activity ("something to do or see") or a note ("a thought about the plan").

To view ideas in the brainstream, one can either scroll down the list, click on a hashtag to display ideas belonging to a particular category, or use the autocomplete search box. Clicking on an idea reveals a dialog box with additional details, an option to edit the idea, and in the case of an activity, an option to add it to or remove it from the current itinerary. A blue badge next to an activity indicates that it is already in the current itinerary.

To add a new idea (an activity or a note), one can type a title into the search box and click "add." If similar ideas already exist, a drop down list will appear, which helps to prevent duplicates and promote editing. For notes, workers can fill in a description. For activities, the *activity editor* (Figure 3.3) asks workers to provide the name of the location, what to do or see, the activity's duration, and the (requester-



Add an activity

Activity: eat pizza at Cambridge 1

Location name: cambridge 1

What to do/see: Try the potato pizza -- it's delicious!

Takes about: 1 hour

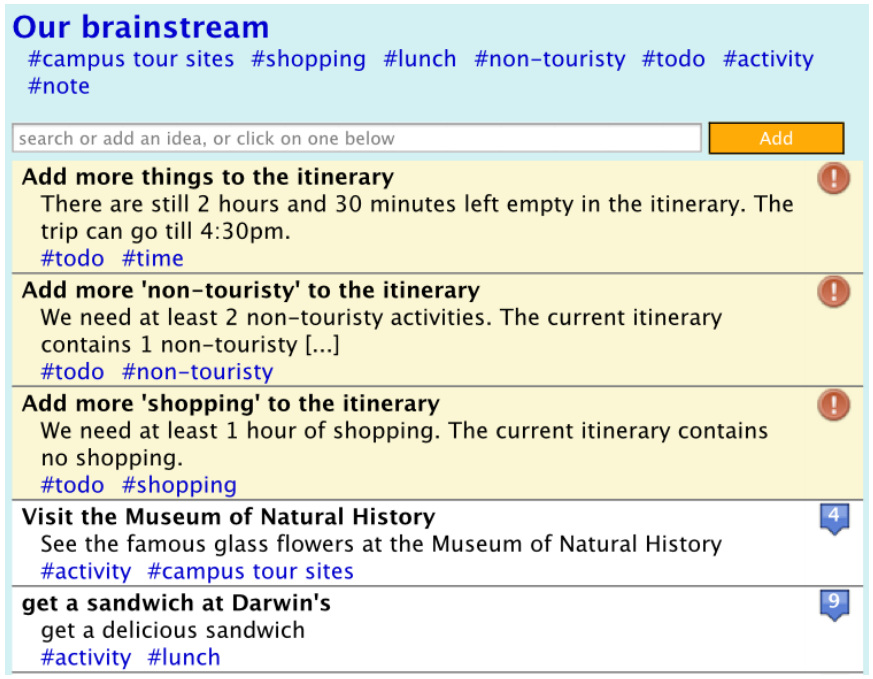
Category: campus tour sites, shopping, [✓ lunch], non-touristy

Drag the location pin to its correct location. Suggestions (when available) are shown as .

Figure 3.3: Adding a new activity to the brainstream

defined) categories that the activity belongs in. In the same editor, workers can view a map, which allows them to mark the location of the point of interest. Workers can decide to add the activity to both the itinerary and the brainstream, or only to the brainstream for the time being.

The brainstream allows workers to brainstorm together and build upon each other's ideas. It keeps around all suggested activities, and allows workers to quickly access them through the hashtags and the search box. By adding notes, workers can identify areas that need work or raise questions about the plan's feasibility, which other workers or the requester can then help to address or provide comments on. The brainstream's design draws inspirations from social technologies such as Twitter and Piazza, that aggregate information into a feed or stream that one can easily process.


Our brainstream
#campus tour sites #shopping #lunch #non-touristy #todo #activity #note
search or add an idea, or click on one below
Add

Add more things to the itinerary
There are still 2 hours and 30 minutes left empty in the itinerary. The trip can go till 4:30pm.
#todo #time

Add more 'non-touristy' to the itinerary
We need at least 2 non-touristy activities. The current itinerary contains 1 non-touristy [...]
#todo #non-touristy

Add more 'shopping' to the itinerary
We need at least 1 hour of shopping. The current itinerary contains no shopping.
#todo #shopping

Visit the Museum of Natural History
See the famous glass flowers at the Museum of Natural History
#activity #campus tour sites

get a sandwich at Darwin's
get a delicious sandwich
#activity #lunch

Figure 3.4: The brainstream displays system-generated todo items which alert workers to what needs work.

If the current itinerary does not satisfy a quantitative constraint or is over time or under time, the violated constraints are automatically turned into *todo items* that are displayed at the top of the brainstream with exclamation marks (Figure 3.4). Todo items alert workers to what needs work. They suggest specific actions, such as “Add a ‘lunch’ activity” or “The itinerary is over time. Try reordering itinerary items. You can also edit or remove items.” Todo items also provide natural language explanations of how the current itinerary violates particular constraints. For example, a todo item may explain that “You need a ‘lunch’ activity but there is currently none in the itinerary” or “The itinerary is over time because the trip must end by 9pm.”

We note that the system is able to check arbitrary quantitative constraints and generate todo items without understanding the meaning of the natural language categories. This is because workers associate activities with the categories they belong in when the activities are suggested. As we will show in the next section, todo items are an important design element that accelerates the speed at which quantitative constraints are resolved.

Itinerary Viewer

The *itinerary viewer* (Figure 3.2, on right) consists of an itinerary and a map. The itinerary displays the activities in order, with times during which they are scheduled to take place. Travel times between locations are automatically computed and accounted for. The map displays the activities' locations and the routes between locations.

The map and itinerary allow crowd workers to see at a glance whether the plan is coherent. A worker may notice activities that are out of order, for example by seeing on the itinerary that lunch is happening too early or seeing on the map that activities can be reordered to avoid unnecessary travel. A worker can also use the itinerary to detect if too much or too little time is spent on an activity.

The itinerary doubles as an editor. Workers can drag and drop activities to rearrange their order, and click an activity to see its details, edit it, or remove it from the itinerary. On any itinerary change (i.e., via adding, removing, editing, or reordering of activities), the itinerary, activity times, map display, trip time, and todo items automatically update, which provides direct feedback to the workers as they refine the itinerary.

Mobi promotes collaboration by making the plan always visible and editable by everyone. This follows the WYSIWIS (What You See Is What I See) principle [88], which ensures that all participants have equal access to shared information. Mobi also supports *opportunistic planning*, by providing support for both top-down and bottom-up planning, and a fluid way to move back and forth between the two. For example, as workers plan at a detailed level (e.g., suggesting activities in the brainstream), they may become aware of shortcomings of the current itinerary, which in turn prompts them to start considering the itinerary as a whole. Likewise, when workers refine the itinerary, they may think of new activities to add to the brainstream, or ways to elaborate on the details of a particular activity in the current itinerary.

3.3 Experiment: Todo or Not Todo

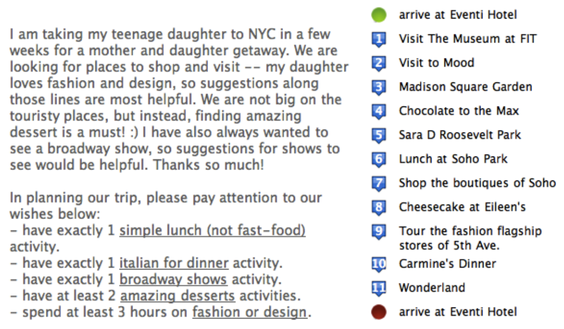
We hypothesize that elements of Mobi's design, namely the todo items and having a shared interface in which the crowd can work off the current solution context and existing ideas, promotes the crowd to effectively and collaboratively resolve the users' stated constraints so as to produce itineraries that satisfy planning missions. In this section, we consider an experiment using two versions of Mobi—one that displays todo items and one that does not—to evaluate the effect of todo items on how quickly the crowd can reach solutions that satisfy the stated quantitative constraints.

3.3.1 Method

We created custom day-trip planning missions for each of eight major U.S. cities: New York, Chicago, Washington DC, Las Vegas, Los Angeles, San Francisco, Seattle, and San Diego. We recruited Amazon Mechanical Turk workers (Turkers) in the U.S. with 95% or higher approval rating to contribute to the planning missions by working on human intelligence tasks (HITs) in which the Mobi interface was fully embedded. The interface is nearly identical to that shown in Figure 3.2, with differences being the addition of a "submit" button on the bottom, a "HIT instructions" button replacing the "what you can do to help" button in the information panel, and the addition of a "continue to improve the itinerary" todo item that displays only when there are no other todo items (all quantitative constraints are satisfied). Turkers were asked to make "micro-contributions" as they plan the trip with other Turkers, and were told that they can submit a HIT as soon as they have made *any* contribution. Turkers were paid 15 cents per HIT, and no verification was used other than requiring Turkers to have made some edit (however small) to the brainstream or itinerary before submitting the task. For half of the cities, the version with todo items was posted prior to the version without todo items, and the order of posting was reversed for the other cities. Missions were posted for up to four days. Other than the display of todo items, the interface, job description, and instructions were identical in the two conditions.

3.3.2 Results I: The Generated Itineraries

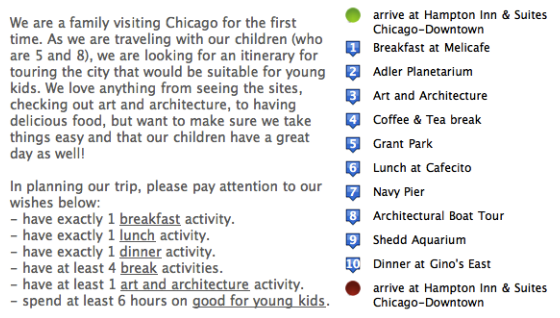
In the todo condition, all eight itineraries satisfied the stated quantitative constraints. Figure 3.5 provides four examples of planning missions and the corresponding itineraries generated by Turkers. From the itineraries, it appears that Turkers not only pay attention to the quantitative constraints, but also to the mission description, for example by including educational activities for the kids on a family vacation to Washington DC.


I am taking my teenage daughter to NYC in a few weeks for a mother and daughter getaway. We are looking for places to shop and visit -- my daughter loves fashion and design, so suggestions along those lines are most helpful. We are not big on the touristy places, but instead, finding amazing dessert is a must! :) I have also always wanted to see a broadway show, so suggestions for shows to see would be helpful. Thanks so much!

In planning our trip, please pay attention to our wishes below:
- have exactly 1 simple lunch (not fast-food) activity.
- have exactly 1 italian for dinner activity.
- have exactly 1 broadway shows activity.
- have at least 2 amazing desserts activities.
- spend at least 3 hours on fashion or design.

● arrive at Eventi Hotel
1 Visit The Museum at FIT
2 Visit to Mood
3 Madison Square Garden
4 Chocolate to the Max
5 Sara D Roosevelt Park
6 Lunch at Soho Park
7 Shop the boutiques of Soho
8 Cheesecake at Eileen's
9 Tour the fashion flagship stores of 5th Ave.
10 Carmine's Dinner
11 Wonderland
● arrive at Eventi Hotel

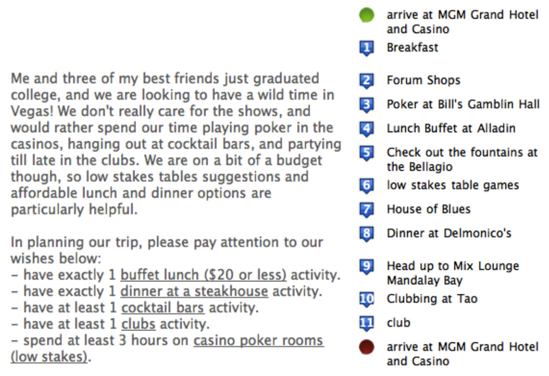
(a) Mother/Daughter NYC


We are a family visiting Chicago for the first time. As we are traveling with our children (who are 5 and 8), we are looking for an itinerary for touring the city that would be suitable for young kids. We love anything from seeing the sites, checking out art and architecture, to having delicious food, but want to make sure we take things easy and that our children have a great day as well!

In planning our trip, please pay attention to our wishes below:
- have exactly 1 breakfast activity.
- have exactly 1 lunch activity.
- have exactly 1 dinner activity.
- have at least 4 break activities.
- have at least 1 art and architecture activity.
- spend at least 6 hours on good for young kids.

● arrive at Hampton Inn & Suites Chicago-Downtown
1 Breakfast at Melicafe
2 Adler Planetarium
3 Art and Architecture
4 Coffee & Tea break
5 Grant Park
6 Lunch at Cafecito
7 Navy Pier
8 Architectural Boat Tour
9 Shedd Aquarium
10 Dinner at Gino's East
● arrive at Hampton Inn & Suites Chicago-Downtown

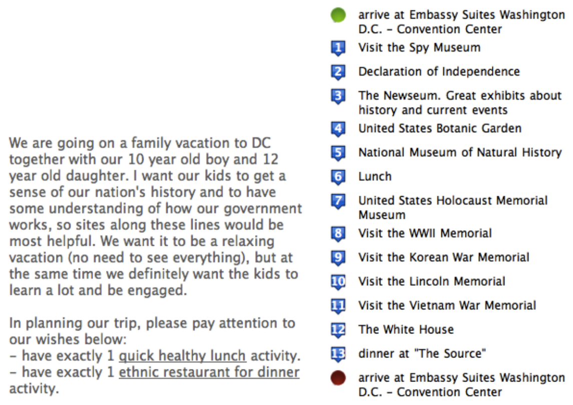
(b) Chicago with young children


Me and three of my best friends just graduated college, and we are looking to have a wild time in Vegas! We don't really care for the shows, and would rather spend our time playing poker in the casinos, hanging out at cocktail bars, and partying till late in the clubs. We are on a bit of a budget though, so low stakes tables suggestions and affordable lunch and dinner options are particularly helpful.

In planning our trip, please pay attention to our wishes below:
- have exactly 1 buffet lunch ($20 or less) activity.
- have exactly 1 dinner at a steakhouse activity.
- have at least 1 cocktail bars activity.
- have at least 1 clubs activity.
- spend at least 3 hours on casino poker rooms (low stakes).

● arrive at MGM Grand Hotel and Casino
1 Breakfast
2 Forum Shops
3 Poker at Bill's Gamblin Hall
4 Lunch Buffet at Alladin
5 Check out the fountains at the Bellagio
6 low stakes table games
7 House of Blues
8 Dinner at Delmonico's
9 Head up to Mix Lounge Mandalay Bay
10 Clubbing at Tao
11 club
● arrive at MGM Grand Hotel and Casino

(c) Vegas with buddies


We are going on a family vacation to DC together with our 10 year old boy and 12 year old daughter. I want our kids to get a sense of our nation's history and to have some understanding of how our government works, so sites along these lines would be most helpful. We want it to be a relaxing vacation (no need to see everything), but at the same time we definitely want the kids to learn a lot and be engaged.

In planning our trip, please pay attention to our wishes below:
- have exactly 1 quick healthy lunch activity.
- have exactly 1 ethnic restaurant for dinner activity.

● arrive at Embassy Suites Washington D.C. - Convention Center
1 Visit the Spy Museum
2 Declaration of Independence
3 The Newseum. Great exhibits about history and current events
4 United States Botanic Garden
5 National Museum of Natural History
6 Lunch
7 United States Holocaust Memorial Museum
8 Visit the WWII Memorial
9 Visit the Korean War Memorial
10 Visit the Lincoln Memorial
11 Visit the Vietnam War Memorial
12 The White House
13 dinner at "The Source"
● arrive at Embassy Suites Washington D.C. - Convention Center

(d) Family in DC

Figure 3.5: Experiment: planning missions and the corresponding itineraries generated by the crowd in the todo items condition

Table 3.1 summarizes, for each of the examples shown in Figure 3.5, statistics about the final itineraries, the types of edits Turkers made, and the amount of money paid to workers. We see that the final itineraries contain original ideas from multiple workers. Turkers generated just over twice as many ideas for activities as are in the final itineraries, and generally used notes sparingly. When notes were added, they provided commentary on alternative suggestions ("They are a better place then Pasty's by far, and have better service, plus that perfect dessert."), noted errors in activities ("Barbary Coast isn't called 'Barbary Coast' anymore"), presented general

	NYC	Chicago	Las Vegas	DC
# unique workers	17	15	16	21
# workers with winning ideas	6	5	7	8
# activities in brainstream	35	16	18	28
# activities in itinerary	11	10	11	13
# edits in brainstream	50	54	43	57
# edits in itinerary	193	140	75	154
# notes in brainstream	1	0	9	1
# of HITs	64	31	47	50
Total cost	\$9.60	\$4.65	\$7.05	\$7.50

Table 3.1: Summary statistics about the final itineraries of the examples shown in Figure 3.5, including contributions by and payments to Turkers. Winning ideas are activity suggestions that are in the final itinerary.

advice ("You can buy a MealTicket which will allow you to eat free at many places."), or pointed out problems with the plan ("why are we eating so much dinner?").

3.3.3 Results II: Impact of Todo Items

Results show that when prompting workers with todo items, quantitative constraints are satisfied significantly more quickly than when todo items are not displayed. We measure the speed at which constraints are satisfied in number of HITs performed. One worker in the no todo condition attempted to game the system by submitting multiple HITs for a single piece of work (e.g., by adding an activity, filling in its details, and placing it into the itinerary in three separate HITs). For this worker, only the itinerary-changing HITs were counted, but for all other workers, all HITs were counted.

We make three observations. First, we found a significant difference ($t(7) = 3.65$, $p = 0.0082$) in the number of HITs it took to satisfy (for the first time) all of the stated quantitative constraints between the todo condition ($\mu = 16.5$, $\sigma = 9.65$) and the no

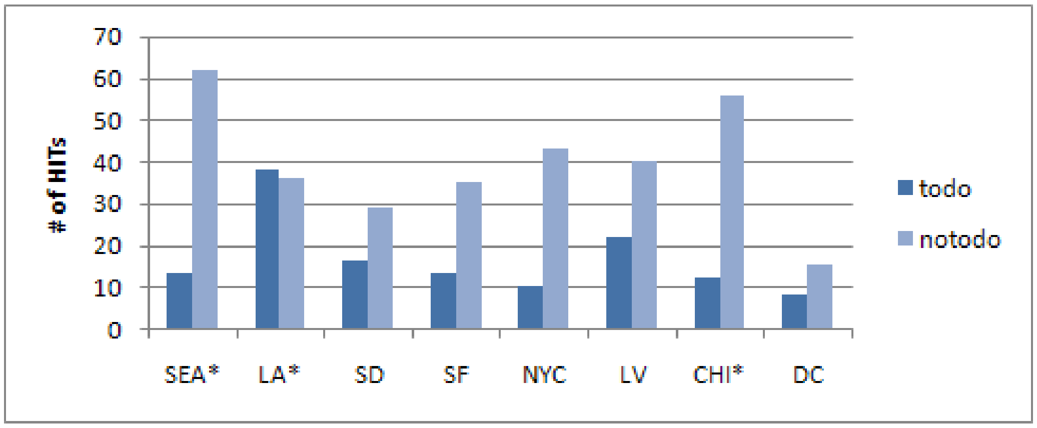


Figure 3.6: Number of HITs required to satisfy all quantitative and system generated time constraints for each city in the todo and no todo conditions. For cities marked by an asterisk, itineraries in the no todo condition still had violated constraints; in such cases we reported the number of HITs thus far.

todo condition ($\mu = 39.5$, $\sigma = 14.8$).[2] See Figure 3.6 for a city-by-city breakdown.

Second, there is also a significant difference ($t(7) = 4.247$, $p = 0.0038$) in the number of HITs it took to satisfy all constraints for the first time (this includes system generated time constraints) between the todo condition ($\mu = 22.5$, $\sigma = 8.5$) and the no todo condition ($\mu = 45.38$, $\sigma = 13.9$).

Finally, as constraints can be violated and satisfied repeatedly throughout the planning process, we sought to understand how quickly constraints are satisfied on average. We introduce the notion of the *violation duration* of a constraint, which is the number of HITs it takes for a constraint to be satisfied by the itinerary since it was last violated (which could be when it was first introduced). The average violation duration of quantitative constraints is shorter for the todo condition ($\mu = 5.64$, $\sigma = 6.34$) than for the no todo condition ($\mu = 10.5$, $\sigma = 10.97$); the result is statistically significant

[2]In some cases for the no todo condition, no itinerary satisfied all the stated requirements in the course of the experiment. In such cases the number of HITs completed thus far was used as a lower bound for comparison.

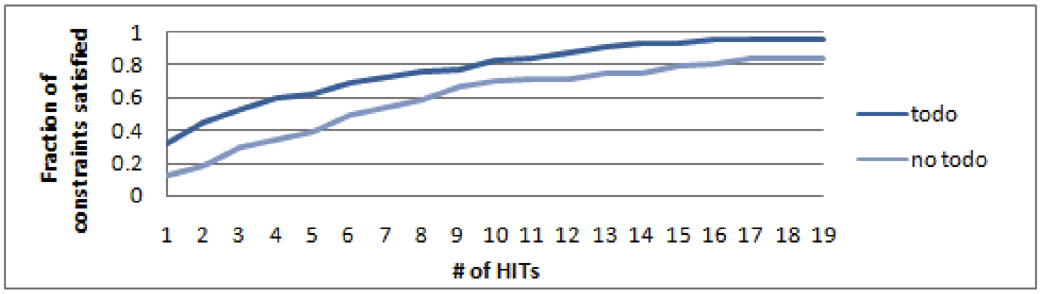
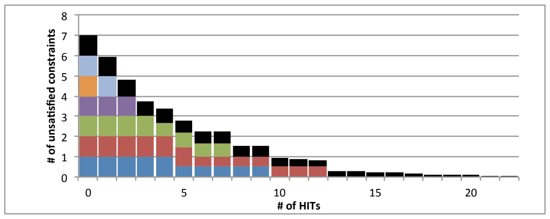


Figure 3.7: Cumulative distribution of the violation duration of constraints in the todo versus no todo conditions, showing the fraction of constraints satisfied after at most k HITs since the time it was last violated.

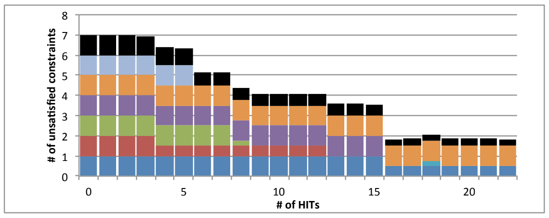
($t(134) = 3.206$, $p = 0.0017$).

Figure 3.7 shows the cumulative distribution of the violation durations of constraints in the todo versus no todo conditions. We observe that for any violation duration we may consider (in number of HITs), a larger fraction of the constraints are satisfied within that duration in the todo condition than the no todo condition. We also see that more than half of all violated constraints were satisfied after three or fewer HITs in the todo condition.

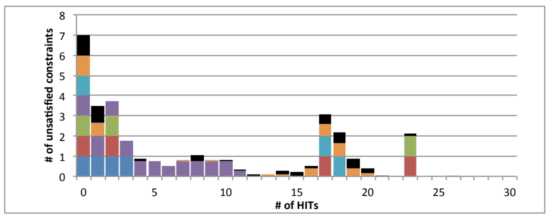
Figure 3.8 shows, for the todo versus no todo conditions, the rate at which each constraint gets satisfied as workers contribute to the planning effort for the Seattle and Chicago planning missions. We observe that constraints were satisfied much more quickly in the todo condition. The Chicago case is particularly interesting. In the todo condition, a worker violated a previously satisfied constraint while editing and proceeded to make successive edits that led to the satisfaction of all constraints. In the no todo condition, a satisfied constraint was violated and then left unaddressed. This example illustrates the power of immediate feedback. When an edit to the itinerary violates some constraint(s), the automatically generated todo items are able to not



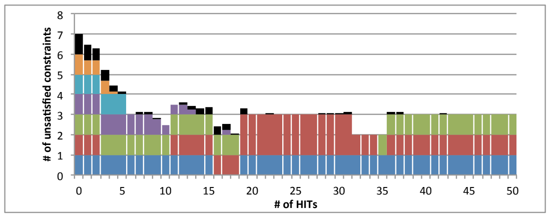
(a) Seattle with todo items



(b) Seattle with no todo items



(c) Chicago with todo items



(d) Chicago with no todo items

Figure 3.8: The unsatisfied constraints during the planning process for the Seattle and Chicago missions. The height of each bar indicates the number of constraints unsatisfied after k HITs. Each colored segment represents a particular quantitative constraint, and its height indicates the extent to which it is violated. The black segment represents the percent by which the itinerary is over time or under time (when it is greater or less than 5%).

only alert workers to what needs fixing, but also make them aware that their edits have a direct effect on the satisfaction of constraints associated with the planning mission.

3.3.4 Results III: Editing Patterns

Having shown that todo items play an important role in focusing the crowd's effort towards satisfying the quantitative constraints, we turn to investigate the crowd's work process while using Mobi in the todo condition. In particular, we look for evidence of collaborative behavior from the crowd and examine the way that they plan using the current context of the plan.

We focus first on the process of generating ideas for activities. We observe that roughly half (52%) of the contributions to the brainstream contain new suggestions while the other half (48%) are edits to existing ideas in the itinerary. Of the edits, 72% are edits on ideas that originated from someone other than the person editing, which suggests that workers are working off others' contributions when they refine ideas and the itinerary. When editing an activity, we see that edits are mostly to an activity's duration (80%), but there are also edits to change titles/descriptions (7%) and to correct an activity's location (12%). Edits to the title, description, and location are encouraging to see as they suggest that the brainstream and itinerary viewer are providing means for users to discover and improve existing ideas.

Turning to the patterns of itinerary edits, we observe that while most of the contributions come from adding (31%) and reordering activities (32%), workers also edit existing ideas (22%) and remove activities (14%). This is encouraging to see because workers are using the different actions available to them to improve the itinerary as they see fit. When tasks are left to run after the quantitative constraints are all satisfied, we observe that itineraries continue to evolve; workers replace activities in the itinerary with other activities, reorder the itinerary, edit existing items, and so on. While constraints may be violated during such edits, todo items reminded workers of such violations and violated constraints were quickly satisfied again (e.g., see Figure 3.8(c)). Workers are encouraged to continuously generate new ideas and incorporate them into the itinerary both because we pay them for such contributions and because Mobi displays a todo item that asks workers to continue improving the itinerary whenever all quantitative constraints are met.

We saw very few Turkers who blatantly tried to game the system. The kinds of gaming behavior we did observe generally fell into two categories. In one, a Turker underspecifies an activity, either by creating an activity without filling in its description and location, or by adding a note containing a suggestion for an activity instead of just adding the suggested activity. In the other, a Turker would fully specify an activity, but use up to three HITs to do so—by spending a HIT on creating the activity, another to edit its details, and another to add it to the itinerary—when all this can be accomplished with a single "add activity" action.

While it is certainly useful to consider refinements that would curb such behaviors (e.g., by requiring activities to contain descriptions; by not allowing workers to submit HITs in which they have only edited their own ideas; etc.), such gaming behaviors from a few Turkers did not seem to have a negative influence on the planning processes nor the resulting solutions. In particular, we saw that poorly formed ideas were simply ignored, removed from the itinerary, or edited by another worker who discovered them via the autocomplete search box in the brainstream, all of which occurred as natural parts of the iterative process through which workers improved the itinerary.

3.4 End-to-End User Study

Having seen that workers can resolve quantitative constraints effectively using Mobi, we conducted a user study to evaluate how well the generated itineraries satisfy not only quantitative constraints, but also the stated qualitative constraints, from the perspective of requesters.

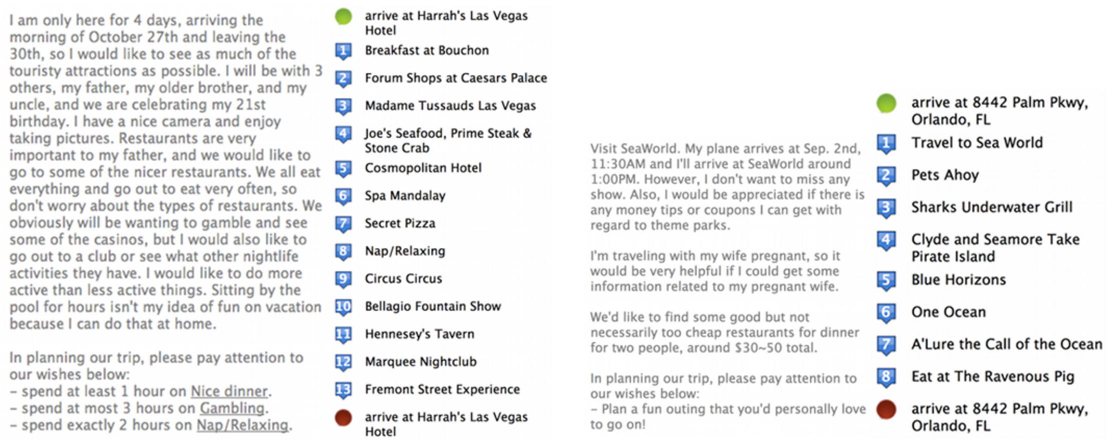
3.4.1 Method

We recruited 10 subjects from university mailing lists to participate in the study. Individuals were eligible if they were actually planning a forthcoming trip to a major U.S. city. Recruited subjects were a mix of undergraduate students, graduate students, and research scientists. Subjects were instructed to describe their planning mission, which includes qualitative and quantitative preferences and constraints. Participants were given unlimited access to Mobi for a week, during which they were free to modify their planning mission and participate in the planning process. Missions were crowdsourced on Mechanical Turk as was done in the todo versus no todo experiment. At the end of the study, subjects completed a questionnaire, which asked them to evaluate the final itinerary and to describe their experience using Mobi. Subjects each received a $30 Amazon Gift Card for their participation.

The trip destinations specified by the users included Boston, New York City, San Francisco, Las Vegas, Orlando, and Washington DC. The planning missions varied in length and specificity. Figure 3.9 provides two examples of user missions and the generated itineraries.

3.4.2 Results

To assess how well the generated itineraries satisfy the users' requirements, we consider three measures of the *quality* of an itinerary, namely the extent to which it (1) contains activities that the requester likes, (2) satisfies the qualitative and quantitative constraints specified in the planning mission, and (3) serves its purpose as a plan that is feasible, useful, and executable in real life.


I am only here for 4 days, arriving the morning of October 27th and leaving the 30th, so I would like to see as much of the touristy attractions as possible. I will be with 3 others, my father, my older brother, and my uncle, and we are celebrating my 21st birthday. I have a nice camera and enjoy taking pictures. Restaurants are very important to my father, and we would like to go to some of the nicer restaurants. We all eat everything and go out to eat very often, so don't worry about the types of restaurants. We obviously will be wanting to gamble and see some of the casinos, but I would also like to go out to a club or see what other nightlife activities they have. I would like to do more active than less active things. Sitting by the pool for hours isn't my idea of fun on vacation because I can do that at home.

In planning our trip, please pay attention to our wishes below:
- spend at least 1 hour on Nice dinner.
- spend at most 3 hours on Gambling.
- spend exactly 2 hours on Nap/Relaxing.

● arrive at Harrah's Las Vegas Hotel
1. Breakfast at Bouchon
2. Forum Shops at Caesars Palace
3. Madame Tussauds Las Vegas
4. Joe's Seafood, Prime Steak & Stone Crab
5. Cosmopolitan Hotel
6. Spa Mandalay
7. Secret Pizza
8. Nap/Relaxing
9. Circus Circus
10. Bellagio Fountain Show
11. Hennesey's Tavern
12. Marquee Nightclub
13. Fremont Street Experience
● arrive at Harrah's Las Vegas Hotel

Visit SeaWorld. My plane arrives at Sep. 2nd, 11:30AM and I'll arrive at SeaWorld around 1:00PM. However, I don't want to miss any show. Also, I would be appreciated if there is any money tips or coupons I can get with regard to theme parks.

I'm traveling with my wife pregnant, so it would be very helpful if I could get some information related to my pregnant wife.

We'd like to find some good but not necessarily too cheap restaurants for dinner for two people, around $30~50 total.

In planning our trip, please pay attention to our wishes below:
- Plan a fun outing that you'd personally love to go on!

● arrive at 8442 Palm Pkwy, Orlando, FL
1. Travel to Sea World
2. Pets Ahoy
3. Sharks Underwater Grill
4. Clyde and Seamore Take Pirate Island
5. Blue Horizons
6. One Ocean
7. A'Lure the Call of the Ocean
8. Eat at The Ravenous Pig
● arrive at 8442 Palm Pkwy, Orlando, FL

(a) Subject 1: Las Vegas (b) Subject 2: Orlando

Figure 3.9: User study: planning missions and corresponding itineraries generated by the crowd

1. Do itineraries contain activities that the requesters like?

Users were shown information about each of the itinerary activities (title, description, start time, end time, duration) and asked to rate how much they think they would enjoy each activity on a 5-point scale (1="hate it", 5="love it").

Figure 3.10 shows a histogram of the activity ratings across all 10 participants. The mean rating was 4.03 ($\sigma = 0.44$). Users also mentioned that the activities are diverse, interesting, and often unknown to them prior to using Mobi.

2. Do itineraries satisfy the qualitative and quantitative constraints specified in the planning mission?

All of the users answered that their itinerary fulfilled most or all of the requirements they had specified. Some users noted specific activities that they did not like, such as one who commented "I just happen to be afraid of bungee jumping because it seems so unsafe, but a similar activity would be fun" and another who commented "I am under age so the wine thing would not be great for me but everything else

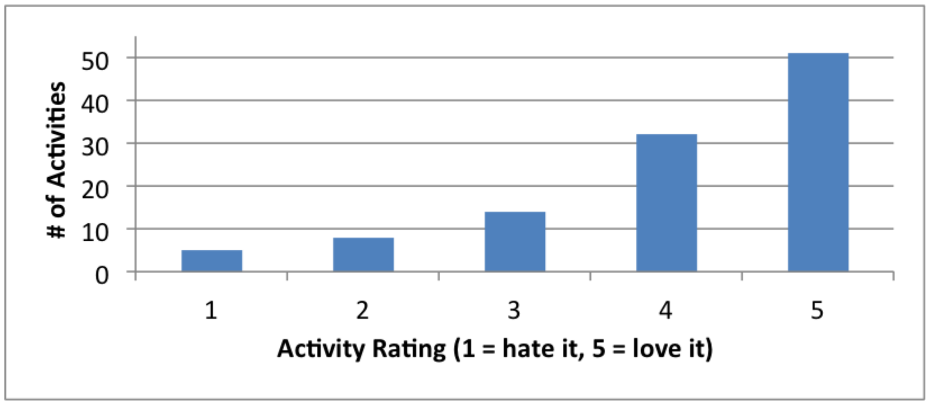


Figure 3.10: Histogram of activity ratings

sounds great." Another user complained about too many activities: "There was far far far too much packed into a single day, but the ideas were all totally interesting." Yet another user felt they were visiting too many parts of a city in one day: "For the most part, it was a good mix of things to do. I was not expecting to travel so much uptown/downtown in one day though."

These problems can be explained in part by the fact that some constraints, such as the notion that an itinerary shouldn't be too packed, are assumed or missed and therefore not explicitly stated by the users. One potential solution is for requesters to evaluate the itineraries as they are being created and add the missing constraints to the planning mission. In fact, as a preliminary test, we took two users' feedback and entered them as todo items (i.e., "Let's just stay midtown and remove downtown activities," "The harbor island suggestions are great but one island would be enough. Please adjust time durations accordingly so the day is not so packed."). We observe that after just a few HITs, workers have already addressed the issue by removing offending activities, reordering activities (so that meals occur at reasonably hours), and adding additional activities (replacing a concert downtown with a Broadway show in midtown).

3. Are the itineraries feasible, useful, and executable in real-life settings?

We asked users if they *would* or *did* use the itinerary in real-life. All users expressed that they would use the itinerary as is, some version of the itinerary, or selected ideas from the itinerary. When asked *"If Mobi were made available for general use, how likely would you want to use such a tool again for recruiting the crowd to help you plan a trip?"*, 7 out of 10 users answered *likely* or *very likely*, 2 answered *neutral* and only 1 answered *unlikely*.

Three users actually followed the itinerary or used the ideas in the itinerary in their real-life trips. One user reported that "having other people involved in the idea-creation process was extremely helpful. It sparked all sorts of ideas that I kept in the back of my head throughout the weekend." Another user remarked that his "trip was mostly in the plan," although his restaurant plans changed during the trip.

We found a dichotomy of users: those who are interested in obtaining a *fully-specified itinerary* and those who are interested in a *loose itinerary* that contains an unordered set of suggested activities that leaves room for exploration. A possible solution is to allow requesters to choose between a fully specified or loose itinerary, which in turn translate into constraints that specify the maximum number of activities in the itinerary, the amount of buffer time between activities, and the extent to which activities need to be ordered.

One of the most frequently mentioned benefits of Mobi is that both the *idea generation* and the *planning* are fully automated, thereby "integrating all the factors one would consider in planning an itinerary," yet making "the time spent creating the plan minimal." Most users (7 out of 10) reported that they were comfortable with an anonymous crowd planning their trip. Furthermore, results show that requesters mostly left the planning up to the crowd. In particular, 3 out of 10 users reported that they never or rarely checked on the progress of the itinerary, 5 did so occasionally, and only 2 did so frequently. Likewise, 7 out of 10 users said that they never went back to modify the mission details or add notes. As one user noted: "the process seemed to work smoothly without my intervention."

3.5 Discussion

Having demonstrated the effectiveness of Mobi for helping the crowd to resolve qualitative and quantitative constraints in the itinerary planning setting, we now revisit the elements of Mobi's design and discuss how these elements may in general inform the design of systems that facilitate a crowd to tackle problems involving global constraints.

Keeping the crowd, the solution, and the context together

Compared to the design of most other crowdsourcing systems for tackling complex tasks, Mobi is distinguished in its use of a single structured interface through which the crowd is exposed to the current solution and the global problem-solving context. This unified view provides a shared context that allows contributors to coordinate and communicate more effectively with one another than approaches where participants work on different subtasks in separate contexts.

Interactions are less controlled, but still structured

Mobi allows workers to choose how they want to contribute to the task. In our studies, we found that workers generate diverse sets of ideas, and make various types of contributions while problem solving. This freedom is particularly important for resolving global constraints as we do not know *a priori* the specific contributions that are needed. Rather, contributions are context dependent. While interactions are less controlled this way, they are still highly structured. The crowd selects from a well-specified set of actions, todo items guide the crowd towards useful actions, and the system provides real-time feedback on the effects of actions.

A language for human-computer communication

In the background, Mobi's automation computes routes and times, checks for violated constraints, and generates todo items. Mobi understands, for example, when all of the quantitative constraints are satisfied. This ability enables Mobi to take actions such as prompting the crowd for future revisions and asking the crowd or requester to check for potential problems. Mobi can do these things without knowing what the constraints mean, because the inputs that it seeks from the crowd include the categories of suggested activities. This information is sufficient for the system to check for violated constraints and therefore assist in the planning process.

A fluid way to refine goals

With complex problems, requirements can change over time as ideas and partial solutions stream in. In Mobi, a requester can add or revise requirements, write notes, or even directly alter the plan during the planning process. The crowd can react to such changes just as they react to the current solution at any other point in the planning process. The iterative nature of the task and the ease with which workers can grasp the current solution and access alternative suggestions make it easy for the crowd to see and respond to such changes.

3.6 Summary and Research Directions

To date, many human computation systems have relied on the assumption that problems can be solved in an algorithmic manner, using explicit procedures that outline the operations that need to be done and how they are ordered. In this chapter, we argue for an alternative *crowdware* approach, in which workers contribute to solving a complex problem in a less controlled environment that allows them to view and build upon each other's ideas and to contribute as they wish, while system-generated alerts and advice guide them towards a solution.

Using itinerary planning as a case study, we introduce Mobi, a system that draws on groupware ideas and uses explicit processes such as the automatic generation of todo items to generate itineraries that satisfy complex, interdependent constraints. Our results show that constraints are resolved efficiently using this design, and that end user found that the generated itineraries satisfied their stated quantitative and qualitative constraints.

On Mobi, we are interested in studying ways to handle the implied constraints that are assumed or missed. The challenge is to make implied constraints visible so they can be tackled like other constraints; possible approaches include having the crowd identify them, using automated procedures to detect and learn about such constraints, and asking requesters to provide feedback. In a related direction, we can also attempt to encapsulate qualitative constraints in todo items, which would allow workers to see everything that needs work in one place. In addition, we envision rich opportunities to integrate different types of automation into Mobi—to detect failures, handle uncertainties, incorporate richer forms of user preferences, and combine automated and human planners in a synergistic way.

On crowdware more generally, we are interested in understanding how to focus the crowd's attention on not only resolving global constraints, but on taking actions that are most likely to lead to high quality solutions. This may involve generalized uses of todo items to focus the crowd's attention and effort on where they are mostly likely to matter, for example by taking into account the potential value of possible refinements. Such value judgments may be guided by heuristic evaluations made by the requester, the crowd, or the automated system. Focusing the crowd on solution quality may also involve taking steps to ensure that a solution is coherent and consistent, and is not suffering from issues often associated with "design by committee." Steps may include explicitly promoting consistency checks, engaging the crowd in making high-level judgments about the coherence of the solution, and having the crowd and automated system make decisions about when to seek feedback from the requester.

In addition to these directions, there are opportunities to explore novel combinations of crowdware and workflow approaches that can enhance the ability of participants to effectively contribute to solving complex problems that are hard to decompose. We elaborate on this direction in Chapter 9.