

# AI-MIX: Using Automated Planning to Steer Human Workers Towards Better Crowdsourced Plans

Lydia Manikonda Tathagata Chakraborti Sushovan De  
Kartik Talamadupula Subbarao Kambhampati

Department of Computer Science and Engineering  
Arizona State University, Tempe AZ 85287 USA  
{lmanikon, tchakra2, sushovan, krt, rao} @ asu.edu

## 1 Introduction

Some of the crowdsourcing applications that deal with computationally hard problems are directed towards planning (e.g. tour planning) and scheduling (e.g. conference scheduling). Studies (Zhang et al. 2012; Talamadupula and Kambhampati 2013) have shown that involving even primitive forms of automated components can help improve the efficiency of the crowd in these applications. Our ongoing work on developing the **AI-MIX** (Automated Improvement of Mixed Initiative eXperiences) system<sup>1</sup> (Manikonda et al. 2014) foregrounds the types of roles an automated planner can play in such human computation systems. The architecture of **AI-MIX** as shown in Figure 1, supports two crucial roles of the automated planning component:

**Interpretation:** This is a process of understanding the requester’s goals as well as the crowd’s plans from semi-structured or unstructured natural language input. This challenge arises because human workers find it most convenient to exchange / refine plans expressed in a representation close to natural language, while automated planners typically operate on more structured plans and actions.

**Steering with Incompleteness:** This process guides the collaborative plan generation process with the use of incomplete models of the scenario dynamics and preferences. These challenges are motivated by the fact that an automated planner operating in a crowdsourced planning scenario cannot be expected to have a complete model of the domain and the preferences.

We now describe how interpretation and steering are handled in the current **AI-MIX** prototype. The main components of the **AI-MIX** interface are:

1. **Requester Specification:** The task description is provided by the *requester* (who is seeking help from the crowd to prepare a tour plan), in the form of a brief description of their preferences, followed by a list of activities they want to do as part of the tour, each associated with a suitable hashtag. These tags are used internally by the system to map turker suggestions to specific tasks.



Figure 1: The **AI-MIX** interface

2. **Turker Inputs:** Amazon MTurk workers (Turkers) have two choices in terms of the kinds of responses (HITs) they can provide to the system: (i) they may add a new activity (action) in response to one of the existing to-do tags; or (ii) they may critique the existing activities (actions).
3. **Existing Activities:** This box displays a full list of the current activities that are part of the plan. New turkers may look at the contents of this box in order to establish the current state of the plan.
4. **Planner Critiques:** The to-do items include automated critiques of the current plan produced by the planner.

Finally, the interface also provides a map that can be used by turkers to find nearby points of interest, infer routes of travel or the feasibility of existing suggestions, or even discover new activities that may satisfy some outstanding tags. **AI-MIX** is similar to Mobi (Zhang et al. 2012) in terms of the types of inputs it can handle and the constraint checks it can provide. However, instead of using only structured input, which severely restricts the crowd and limits the scope of their contributions, our system is able to parse natural language from user inputs and reference it against the relevant actions in a tour planning domain model in order to be able to reason with the inputs at a deeper level.

**Action Extraction** The system performs parts of speech (POS) tagging on the activities to identify the name of the activity and the places that turkers are referring to; currently, we tag *verbs* and *nouns* using the *Stanford Log-Linear Part-of-Speech* tagger (Toutanova et al. 2003).

**Sub-Goal Generation** **AI-MIX** uses a primitive PDDL description of activities in the tour planning domain to de-

termine whether the planner has additional subgoal annotations on an activity provided by turkers. The actions in the model are all sufficiently high-level; for e.g. *visit*, *lunch*, *shop* etc. Each action comes with a list of synonyms, that help the planner in identifying similar activities. Each action also comes with generic preconditions. When the planner determines that a turker generated activity matches one of the actions in its model, it generates sub-goals to be added as to-do items in the interface.

**Constraint Checking** Along with generating sub-goals for existing activities, our system automatically checks if constraints on duration and cost given by the requester are met. If these constraints are violated, then that violation is automatically added to the to-do stream of the interface, along with a description of the constraint that was violated.

**Plan Generation** While suggestions are being provided by the crowd, a knowledge base is built in the background that augments this information with common sense knowledge in the form of temporal, existential, contiguity and non-concurrency constraints. The knowledge base also contains domain independent axioms, object and state declarations along with the effects and preconditions of the actions. We use Answer Set Programming (*clingo*) to build the final schedule based on this knowledge base.

A video run-through of **AI-MIX** can be found at: <http://bit.ly/1nHxBN7>. Also, a version of our system is currently active at <http://bit.ly/1qD539I>.

## 2 Preliminary Evaluation

Here, we present a preliminary evaluation of **AI-MIX** on Amazon’s MTurk platform. For our study, HITs were made available to all US residents with a HIT approval rate greater than 50%; workers were paid 20 cents for each HIT. We used tour planning scenarios for six major US cities, reused from (Zhang et al. 2012). We will compare the results from two experimental conditions:

- C1: Turkers quantify their suggestions in terms of cost and duration, and the system checks these constraints for violations with respect to the requester demands.
- C2: In addition to C1, we process free-form text from turkers to extract actions and map them to our PDDL model to generate alerts for sub-goals and missing preconditions.

Both conditions were uploaded at the same time, with the same task description and HIT parameters and compared on 6 scenarios and given 2 days before the HITs were expired. The interface for both C1 and C2 is identical to eliminate any bias. More than 150 turkers responded to our HITs.

**Generated Tour Plan Quality** We noticed that the quality of the plans, in terms of detail and description as shown in Table 1, seems to increase in C2. This is because in C2 we now have users responding to planner critiques to further qualify suggested activities. For example, a turker suggested “not really fun, long lines and cannot even go in and browse around” in response to a planner generated tag related to a “fun club” activity suggested previously, while another suggested a “steamer” in response to a planner alert

<b>Lunch:</b> Alice’s Tea Cup #lunch(20 \$)
<b>Show:</b> ABSOLUTELY CANNOT go wrong with Phantom of the Opera #show(3 hrs)(200 \$)
<b>Activity Steamer:</b> #lunch #lunch_todo1
<b>Paradis To-Go:</b> Turkey & Gruyere is pretty delicious. The menu is simple, affordable, but certainly worth the time #lunch(1 hrs)(10 \$)
<b>watch Wicked (musical):</b> Do watch Wicked the musical. It’s a fantastic show and one of the most popular on Broadway right now! #broadwayshow(3 hrs)(150 \$)

Table 1: Sample activity suggestions from turkers for the two conditions: C1 (top) and C2 (bottom).

Critiques by	Suggested	Addressed
Humans	8	0
Planner	45	7

Table 2: Statistics of critiques generated and addressed.

about “what to eat for lunch”. This seems to indicate that including a domain description (generic preconditions of actions in PDDL model) in addition to the simplistic constraint checks increases the plan quality.

**Role Played by the Planner Module** We received a total of 31 new activity suggestions from turkers, of which 5 violated constraints. The C2 interface attracted 39 responses, compared to 28 for C1, which may indicate that the planner tags encouraged turker participation.

Note that in the **AI-MIX** interface, there is no perceptual difference between the critiques generated by the planner and the critiques suggested by humans. With this in mind, as shown in Table 2 there were 8 flaws pointed out by humans, but none were acted upon by other turkers; the planner on the other hand generated 45 critiques, and 7 were acted upon and fixed by turkers. This seems to indicate that turkers consider the planner’s critiques more instrumental to the generation of a high quality plan than those suggested by other turkers. Though these results are not entirely conclusive, there is enough evidence to suggest that the presence of an automated critiquing system does help to engage and guide the focus of the crowd.

**Acknowledgments.** This research is supported in part by the ARO grant W911NF-13-1-0023, the ONR grants N00014-13-1-0176 and N0014-13-1-0519, and a Google Research Grant.

## References

Manikonda, L.; Chakraborti, T.; De, S.; Talamadupula, K.; and Kambhampati, S. 2014. AI-MIX: Using automated planning to steer human workers towards better crowd sourced plans. In *IAAI*.

Talamadupula, K., and Kambhampati, S. 2013. Herding the crowd: Automated planning for crowdsourced planning. *CoRR* abs/1307.7720.

Toutanova, K.; Klein, D.; Manning, C. D.; and Singer, Y. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic dependency network. In *Proc. of HLT-NAACL*.

Zhang, H.; Law, E.; Miller, R.; Gajos, K.; Parkes, D.; and Horvitz, E. 2012. Human Computation Tasks with Global Constraints. In *Proc. of CHI*, 217–226.