Collaborative Planning with Encoding of Users' High-level Strategies

Joseph Kim
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139
joseph_kim@csail.mit.edu

Christopher J. Banks
Norfolk State University
700 Park Ave
Norfolk, VA 23504
cjbanks@mit.edu

Julie A. Shah
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139
julie_a_shah@csail.mit.edu

Abstract

The generation of near-optimal plans for multi-agent systems with numerical states and temporal actions is computationally challenging. Current off-the-shelf planners can take a very long time before generating a near-optimal solution. In an effort to reduce plan computation time, increase the quality of the resulting plans, and make them more interpretable by humans, we explore collaborative planning techniques that actively involve human users in plan generation. Specifically, we explore a framework in which users provide high-level strategies encoded as soft preferences to guide the low-level search of the planner. Through human subject experimentation, we empirically demonstrate that this approach results in statistically significant improvements to plan quality, without substantially increasing computation time. We also show that the resulting plans achieve greater similarity to those generated by humans with regard to the produced sequences of actions, as compared to plans that do not incorporate user-provided strategies.

Introduction

Automated planning is an important field of AI research that addresses many real-world optimization problems, including vehicle routing, job-shop scheduling, and mission planning for military engagements and disaster response. Substantial research efforts have focused on improving the performance of domain-independent generative planners. These planners automatically produce *plans*, or sequences of actions that transform the initial state into a state that satisfies the goal specification.

Even in its simplest form, automated planning is computationally hard — to be precise, it is PSPACE-complete (Bylander 1991). With the inclusion of numerical variables and temporal actions, the computational complexity increases further to EXPSPACE-complete (Rintanen 2007). Currently, there are only a small number of planners available that can handle numerical and temporal problems, and designing planners that generate high-quality plans and perform efficiently across various domains remains an open challenge (Chrpa, Vallati, and McCluskey 2014). Greedy search heuristics often yield low-quality plans that can result in wasted resources; also, even in the event that an adequate plan is generated, users may have difficulty interpreting the reason why the plan performs well and trusting it.

Such challenges motivate the development of a form of collaborative planning, where instead of treating the planner as a "black box," users are actively involved in the plan generation process. Results from prior research have indicated that human experts can positively aid machine learning classifiers by providing feedback about feature selection (Cheng, Wei, and Tseng 2006; Raghavan, Madani, and Jones 2006), scheduling algorithms (by dynamically modifying weights of objective functions) (Clare et al. 2012) and search algorithms (by highlighting local search regions) (Anderson et al. 2000). Similarly, we expect that users can augment automated planners — for example, by providing insights gained from spatial reasoning or by communicating strategies that generalize across problems — in order to produce higher-quality plans than those generated by an automated planner alone.

In this work, we explore a type of interaction through which users provide high-level strategies to a planner. Humans can readily describe their dominant strategies during problem-solving scenarios (Hayes-Roth 1980; Nielsen 1994), and such communication at a strategic level is more succinct than explicitly enumerating the steps of a plan. This method of communication is similar to the way in which a supervisor would instruct his or her subordinates about a new problem (Du Boulay and Luckin 2001). The strategies derived for a specific problem could also generalize across other problem instances, potentially improving learner's performance in different circumstances.

We encoded users' strategies using *preferences*, which are composed of soft goals and soft trajectory constraints, using the Planning Domain Descriptional Language (PDDL) 3.0 (Gerevini and Long 2006). Trajectory constraints incorporate linear temporal logic to constrain the plan structure. We used soft constraints, as opposed to hard constraints, because the latter could over-constrain the problem and potentially render it unsolvable — circumstances we would want to avoid when users' elicited strategies are infeasible or poor. Preferences also allow for the association of a reward for preserving user's strategies, or a penalty for breaking them. The quality of the resulting plan is then assessed quantitatively in terms of its *net benefit* — minimizing the original problem cost while maximizing reward associated with the preservation of users' strategies.

We evaluated our collaborative model on a human planning dataset involving 36 participants, each of whom solved two benchmark problems from the International Planning Competition (IPC). Along with the generated plans, participants also provided their own high-level strategies; we manually encoded these strategies as preferences and provided them as input to a state-of-the-art PDDL 3.0 planner. We found that the quality of the resulting plans improved by a statistically significant 12.6% on average, without substantially increasing plan computation time. These plans also achieved greater action similarity to users' manually-generated plans, as compared to plans that were generated without the use of user-provided strategies. To the best of our knowledge, this is the first work to propose and empirically demonstrate the potential benefits of a collaborative planning framework in which users' suggested problem-solving strategies are incorporated into a domain-independent planner.

Related Work

Early works in collaborative or mixed-initiative planning date back to the mid-1990's, when the authors of the TRAINS (Ferguson et al. 1996) and TRIPS (Ferguson and Allen 1998) projects sought to provide users with the means to interactively guide plan construction. In these projects, users interacted with a planning assistant via a combination of natural language and graphical displays. While TRAINS and TRIPS introduced early prototypes of interactive planners, their focus was on HCI functionalities (e.g., language processing, dialogue management and multi-media). Similarly, PASSAT (Myers et al. 2002) is a plan-authoring system that features interactive tools such as the ability to modify problem files and access to a pre-existing library of plan templates. These systems provide authoring environments for user-driven planning, while our work considers user participation in automated planning, where the user provides input at a higher level of abstraction.

Other systems have led to support for certain low-level interactions with automated planners, such as ordering goals for expansion, selecting which actions to apply and choosing instantiations for planning variables (Tate 1994; Wilkins 1999). Systems such as AI-MIX (Manikonda et al. 2014) and RADAR (Vadlamudi et al. 2016) help users compose plans while the system performs constraint checking and offers constructive critiques (e.g., suggested actions that users can either accept or reject). AI-MIX also introduced a planning system that steers users' planning process to better mimic a database of plans collected from a crowd. The aforementioned systems provide decision support for the user, who acts as the primary planning agent — in short, the planner assists the human user. Conversely, our work aims to facilitate human support of an automated planning process.

The idea of users providing high-level strategies to the planner was first introduced by Myers (Myers 1996); however, no empirical evaluations analyzing the benefits of such a model were conducted at the time. We consider the proposed domain-independent collaborative planning framework and its empirical evaluation to be the primary contribution of our work.

The concept of incorporating user knowledge to augment automated planning has also been explored by the *knowledge-based* planning community.[1] Knowledge-based planners rely upon experts to hand-code domain-specific information. The form of these user inputs is quite different from standard planner inputs, and crafting them typically requires expert knowledge about the domain and/or the search algorithm used by the automated planner. The "knowledge" provided by the user has typically been encoded as task decomposition schemas via hierarchical task network (HTN) planning (Nau et al. 2003). In contrast, our work incorporates user input via a standard planning domain description language — specifically, PDDL 3.0. We encode strategies as preferences that are specified using linear temporal logic (LTL).

We used PDDL 3.0 primarily because it is domain- and planner-independent, as well as a standardized language used in all IPCs. This enables seamless implementations of various models of collaborative interaction using state-of-the-art planners. We manually encoded natural language descriptions of users' strategies into preferences, and leave automatic translation for future work. (We also leave for future work investigation into the potential benefit of using languages other than PDDL 3.0.)

Related works in the field of scheduling have demonstrated the benefits of collaborative systems: situational awareness and user satisfaction improved when users were able to set priorities for different objective functions (Clare et al. 2012), and task performance improved when users were able to accept or reject generated schedules and select from a library of heuristics (Cummings and Bruni 2010). Another recent work proposed a method to automatically learn scheduling heuristics from expert demonstrations (Gombolay et al. 2016). However, the encoding of user-provided high-level strategies for scheduling problems is an underexplored area. Since scheduling and automated planning communities often consider the same real-world problems, findings related to the type of interaction explored in this work may generalize across the two.

Preliminaries and System Architecture

In this section, we describe the planning problem and preferences, as well as our system architecture for collaborative planning.

Planning Problem

A planning problem is a tuple, $\langle F, I, G, A\rangle$, where F is a set of fluents (state variables), $I \subseteq F$ and $G \subseteq F$ are initial state and goal states and A is a set of actions with precondition, add and delete lists $Pre(a)$, $Add(a)$ and $Del(a)$, respectively, all of which are subsets of F. An action sequence $\pi = (a_o, ...a_n)$ is identified as a *valid* plan when the resulting state achieves each goal g in G. In propositional planning, the cost of the plan is represented by its

[1] Knowledge-based planners were prevalent until the third IPC in 2002; the track was removed from subsequent competitions.

number of actions, $|\pi|$. In numerical planning, action costs $c : A \mapsto \mathbb{R}^+$ and numeric fluents are introduced. In temporal planning, actions can temporally overlap with durations and objective functions include makespan. Formally, these planning aspects are well-defined in PDDL 2.1 (Fox and Long 2003).

Preferences

PDDL 3.0 introduces soft goals and soft trajectory constraints, which together denote preferences. Preferences, P, are subsets of fluents that a valid plan need not necessarily achieve, but penalty functions, λ, are assigned for their violation (which we denote as $\pi \not\models p$). The planning problem then seeks to minimize the following total cost as follows:

$$\min \sum_{i=1}^{|\pi|} c(a_i) + \sum_{p:\pi \not\models p} \lambda(p) \tag{1}$$

This enforces planners to minimize action costs while satisfying preferences as much as possible. Notice that priorities can be set to favor individual preferences by varying λ.

While soft goals are defined at the final state, soft trajectory constraints assert the conditions that must be met by the entire sequence of states visited during plan execution. These constraints are expressed using temporal modal operators over first-order logic involving state variables. Basic operators include `always, sometime, at-most-once` and `at-end`. Precedence relationships are set using `sometime-after, sometime-before` and `imply`. Operators such as `within, hold-after` and `hold-during` represent deadlines with lower and upper bounds. For example, a preference indicating that the fuel level of all vehicles should never drop below a threshold of 5 units can be represented by the following: `(always (forall (?v - vehicle) (>= (fuel ?v) 5))`. Operators can be nested to form more complex expressions. For the full list of operators and their LTL specifications, we refer readers to work by Gerevini and Long (Gerevini and Long 2006).

Unlike hard constraints, preferences cannot be directly used to prune the search space; however, they can serve as forward search heuristics wherein several techniques use relaxed planning graphs and sum the layers containing preference facts (Baier, Bacchus, and McIlraith 2009). These are then used to estimate goal distances and preference satisfaction potential. Other approaches represent preferences as automata, with the position of each preference (satisfied, violated or eternally-violated) stored in a state alongside other fluents (Coles and Coles 2011; Benton, Coles, and Coles 2012). Updates to these are synchronized with the application of actions to states, and mixed-integer programs are utilized for cost minimization. Prior results have indicated that preferences can be tested for satisfiability in polynomial time, and that their integration does not necessarily increase computational complexity (Domshlak et al. 2003).

System Architecture

Figure 1 depicts an overview of our collaboration model. First, the original domain and problem files are graphically

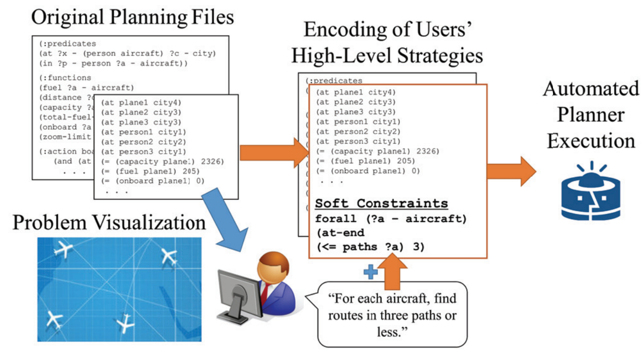

Figure 1: Collaborative planning with users' high-level strategies encoded as preferences.

visualized for the human user. Information visualization is important to amplify users' comprehension and organization of the problem (Lino, Tate, and Chen-Burger 2005) and visual representations (as opposed to pure text) allow users to better see the "big picture," providing semantic support for deriving high-level strategies.

Users' problem-solving strategies are elicited as free-form text responses and encoded as preferences. The preferences are then appended to the problem file given as input to the automated planner. For this work, preference encoding was performed manually by two experimenters (inter-rater agreement = 90%). Although automatic translation of users' natural language into machine-understandable form (Nikora and Balcom 2009; Dzifcak et al. 2009) is an important research area, we view this as a separate problem and did not focus on it in this work.

Human Planning Dataset

In this section, we demonstrate our collaboration model using benchmark problems from the International Planning Competitions (IPCs). We selected problems from two domains: "Zenotravel" and "Satellite," both presented during the third IPC. "Zenotravel" models a multi-vehicle routing problem requiring the transportation of passengers to their destinations. Relevant actions involve embarking and disembarking passengers onto aircraft that can fly at two alternative speeds between multiple cities. The planes consume fuel at different rates according to speed, and the distances between the cities vary. "Satellite" models a task-allocation problem involving the use of multiple satellites to observe space phenomena. The satellites are equipped with instruments that vary according to their calibration targets, image types, data storage and fuel usage. Problem instances in both domains seek to minimize linear combinations of fuel use and makespan.

We tested the problems using three different planning complexities: propositional, numerical and temporal. For each domain-complexity combination, we selected moderate-sized problem files[2] — problems that human users

[2]We used the following benchmark problems for Zenotravel: pfile15 (propositional), pfile13 (numerical) and pfile14 (temporal).

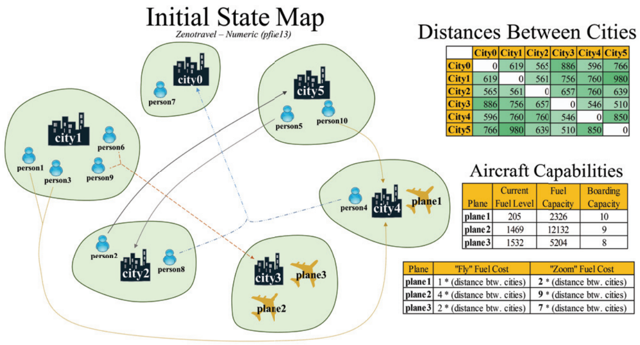

Figure 2: An example of a visualization file used for the Zenotravel–Numerical problem. The map shows the initial state and the arcs represent goal conditions. Other planning information was presented through tables.

could reasonably solve within 30 minutes of planning time (as determined through results obtained from six subjects during our pilot study).

For problem visualization, we used 2D maps to represent initial and goal states where co-located objects signified their pairwise relationship (e.g., a person situated in a city or a satellite pointing at a star), and used tables to organize information about agent capabilities, action costs and constraints on state variables. Figure 2 depicts a visualization file used for the Zenotravel–Numerical problem.

Study Protocol

We employed a 2x3 mixed study with domain as the within-subject variable and complexity as the between-subject variable. Each complexity block consisted of 12 participants, for a total of 36 participants.

At the start of each planning problem, participants read introductory paragraphs (taken directly from the IPC domain description) describing the context and goal of the planning problem. Participants were then presented with visualization files and allowed to ask the experimenters general questions about the problem. This briefing period took no more than 5 minutes across all participants.

Participants then received 30 minutes to derive high-level strategies and produce a plan implementing those strategies. Their strategies, along with the plan, were recorded as free-form text responses. We asked participants to estimate and report the times they spent on assessing the problem, deriving strategies, and producing a plan. Participants then repeated the same process for the second domain. (The ordering of domains was randomized across participants.)

Participants spent about 5 minutes on average assessing the problem, 6 minutes on strategy development and 12 minutes on plan production. The remaining time was used to manually record their plans. These average times did not differ significantly across domain-complexity conditions.

The participants were between 19 and 48 years of age (M=24.1, SD=6.7) and were mostly students from the local campus. The entire study took approximately 75 minutes to complete, and each participant was compensated $15, with a bonus +$5 if their plans were valid and a +$25 prize if their plans resulted in the lowest costs compared with the plans generated by other participants.

For Satellite, we used pfile12 (p), pfile5 (n), and pfile9 (t).

Table 1: Examples of strategies and their preference encodings.

User Strategy	Preference Encoding
"Try to fly using plane1 as much as possible"	`forall (?a - aircraft ?p - person) (always (imply (in ?p ?a) (= ?a plane1)))`
"I tried to have every passenger meet in city2"	`(sometime (forall (?p - person) (at ?p city2)))`
"Do not make any redundant trips (i.e., planes should visit each city at most once)"	`forall (?a - aircraft ?c - city) (at-most-once (at ?a ?c))`
"If plane3 visits city1 then it should visit city2"	`(sometime-after (at plane3 city1) (at plane3 city2)))`
"For each satellite, find routes in three turns or less"	`(forall (?s - satellite) (at-end (<= (turns ?s) 3))`
"Each target should have at most one satellite pointing at the same time"	`(forall ?s1 ?s2 - satellite, ?t - target) (always (implies (and (pointing ?s1 ?t) (pointing ?s2 ?t)) (= ?s1 ?s2)))`
"Whenever data storage exceeds 500, point towards the ground station within 5 time units"	`(forall (?s - satellite) (always-within 5 (>= (data ?s) 500) (pointing ?s groundstation))`

Preference Encodings

The user-provided strategies were manually encoded as preferences by two experimenters. The translation took 3 minutes on average (SD=1.3) strategy.[3] Table 1 depicts a subset of our collected strategies and their corresponding preference encodings. From the full set of 72 user-generated strategies, 53 were successfully encoded in PDDL 3.0 in the form of preferences. The remaining strategies could not be encoded due to the users' generic responses, such as *"try to make all trips continuous"* and *"pass by any bottlenecks"*.

In the above examples, the meanings of "continuous" and "bottlenecks" are ambiguous and difficult for translators to map to features in the original problem specification. We could have employed a template from which participants could select pre-populated parameters and thereby compose their strategies, but we chose not to restrict participants' choice of expression.

[3]Note that the experimenters were previously familiar with PDDL and LTL languages and had practiced translations on the pilot study data.

Table 2: Summary of planning results.

		(1) Planner Only - Baseline			(2) Planner + User Strategies				(3) User Plans	Difference btw. costs of (1) and (2)
Domain	Planning Complexity	*Plan Cost*	*Sim. Recall*	*Sim. Rel. Seq.*	*Plan Cost*	*N*	*Sim. Recall*	*Sim. Rel. Seq.*	*Plan Cost*	*% improve / P-value*
Zenotravel	Propositional	53	0.24	0.58	52.1 (0.88)	7	0.28	0.61	44.7 (0.93)	1.7% / p=0.5
	Numerical	18210	0.12	0.28	**15840** (1210)	11	0.24	0.64	7869 (1284)	13.0% / p=0.027*
	Temporal	806	0.18	0.81	**707** (31)	7	0.21	0.87	656 (35)	11.5% / p=0.031*
Satellite	Propositional	43	0.58	0.62	44.1 (2.8)	10	0.59	0.77	47.5 (1.33)	-2.5% / p=0.25
	Numerical	199.8	0.36	0.71	**168** (12.4)	9	0.44	0.80	157.8 (17.2)	15.9% / p=0.020*
	Temporal	51	0.23	0.88	**46** (1.9)	9	0.31	0.95	40.3 (2.5)	9.8% / p=0.027*

Parentheses denote standard errors of the mean. N refers to the number of strategies for each domain-complexity condition. * represents statistical significance.

Thirteen out of 53 user strategies involved modification of the objective functions to include certain state variables. Responses such as *"reduce the number of refuels"* and *"minimize the number of total flights"* required state variables to be added to the optimization metric (e.g., `cost metric: 1*makespan + 3*fuel` **`+ flights`**). Weights of appended state variables were set to 1.

Fifteen of the 53 strategies required the creation of new state variables for successful encodings. For example, the response *"the total number of calibrations should be less than 5"* required the planner to keep track of calibration counts, which was a variable not originally defined in the Satellite domain. The inclusion of new state variables required adjustments to the domain description file.

Penalty weights (λ) on encoded preferences were set to equal 20% of the estimated problem cost. If a strategy involved several preferences (which occurred 17 times out of 53), the weight was evenly distributed. The new problem file was then sent as input, along with the domain description file, to the automated planner. We employed the following state-of-the-art planners able to incorporate preferences: LPRPG-P (Coles and Coles 2011), for propositional and numerical planning; and OPTIC (Benton, Coles, and Coles 2012), for temporal planning. The costs of the generated plans were then assessed based on the original problem file without the preferences and penalty weights. We used VAL4 tool (Howey, Long, and Fox 2004) for plan validation and evaluation.

Results and Discussion

In this section, we empirically validate that collaborative planning with user-provided strategies yields higher-quality plans than those produced by an automated planner alone. We ran the automated planners (anytime planners) and report the best-quality plans generated within 30 min of CPU-time. We selected a fixed budget of 30 min because this is the standard employed in the IPC and to draw fair comparison to the human planning session, which took 30 min.[4]

[4]We also ran the planners for 60 and 120 min and found consistent results. All tests were performed using an Intel Xeon Processor (2.27 GHz, 12MB Cache, 16 cores) with 16GB of RAM.

Table 2 summarizes our results when comparing plans generated by (1) the automated planner only, which represents our baseline; (2) the automated planner incorporating users' strategies; and (3) the users only.

Plan Costs

Within the Zenotravel domain, we found that collaborative planning (2) improved cost minimization on average across all three planning complexities. The cost of generated plans improved by 1.7%, 13.0% and 11.5% for propositional, numerical and temporal problems, respectively. Wilcoxon signed-rank tests with a significance level at 0.05 indicated statistically significant improvements for numerical and temporal problems (p-values are shown in the last column of Table 2). Similar results were observed within the Satellite domain: plan costs significantly improved by 15.9% and 9.8% for numerical and temporal problems.

Note that only the collaborative plans for the Satellite–Propositional problem performed worse than the baseline; in this case, the plan generated by the baseline was actually optimal.[5] This was the only problem for which the users themselves produced poorer plans on average than the baseline. Since there was no effective room for improvement, users' strategies perturbed generated plans away from the optimal baseline. However, the difference in cost was not statistically significant, and four of the 10 user strategies yielded plans that maintained the optimal cost of 43. Such behavior exemplifies the benefit of employing preferences, indicating robustness for situations in which users' strategies may be deemed poor or unnecessary.

Our findings suggest that human users can outperform state-of-the-art planners on a subset of IPC benchmark problems (the differences between plan costs in (1) and (3) were all statistically significant, with $p<0.01$). Note, however, that we selected moderate-sized problems, and whether similar patterns would result for smaller- or larger-sized prob-

[5]This was the only problem out of the six for which the baseline produced an optimal plan. The true optimal cost for the Zenotravel–Propositional problem was 41. Optimal costs for numerical and temporal problems could not be determined even after a week of computation (the programs eventually crashed due to memory limit).

lems remains to be investigated. We hypothesize that users could suffer from information overload if problem instances become too large, potentially hindering users' ability to provide useful strategies; meanwhile, smaller problems could be solved effectively with automated planners alone. Nevertheless, our results demonstrate that users need not be domain experts in order to assist automated planners — novices can provide useful strategies quickly (in approximately 6 min) and positively influence plan generation even when they are being introduced to the given problem for the first time.

Six out of 72 user-generated plans were found to be invalid due to violated constraints on state variables (likely due to arithmetic errors) and/or unsatisfied goal conditions. However, users' high-level strategies still proved to be beneficial, producing lower-cost plans compared to the baseline. This shows that even in cases where humans are holistically worse at the planning task than automated planners, they can still potentially provide input that complements the automated planner and improves performance.

We also found that collaborative planning did not significantly influence propositional planning. We suspect this may be due to the fact that propositional planning is a much easier problem to solve than numerical/temporal planning, and that state-of-the-art planners generate satisficing solutions reasonably well within a short amount of time. The search space for numerical/temporal planning (or other complex fields, such as probabilistic planning) is exponentially greater; we show that our collaborative framework, utilizing users' high-level strategies, could be a promising avenue for improving low-level search and producing higher-quality plans.

Varying Preference Weights

We ran a sensitivity analysis varying λ from 1–50% and found that the produced plans were robust to the variation. This is likely due to the fact that almost all user-provided strategies elicited in our study were helpful in solving the planning problems and the planners readily incorporated the strategies even with λ set to 1%. We also synthetically generated infeasible strategies and confirmed that the planners correctly rejected the strategies across the tested range for λ. We would like to further explore the relationship between the utility of a strategy and the setting of λ in future work.

Plan Similarity

Next, we evaluated our framework's ability to better mimic (or reproduce) human plans. Machine-generated plans that better match the expected plan-making process of a user would naturally translate to increased plan interpretability and comprehension. We explored how well adding in the preferences (2) makes the generated plan more like the user's plan (3) vs. not having any input from the human (1). Therefore we directly compared measures of plan similarity between the pairs (2)–(3) and (1)–(3).

We used the following measures to compute plan similarity: *Recall* represents the percentage of the user's plan recovered in the machine-generated plan. This percentage is calculated by comparing the set of grounded actions. If even a single parameter for an action differs between the two actions, a mismatch is recorded. Meanwhile, the percent accuracy of plan sequencing is computed by the ratio $\frac{\text{\# of correctly ordered pairs of grounded actions}}{\text{total \# of pairs of grounded actions}}$. We used this *relative sequencing* measure on bigrams of actions because it does not compound sequence errors (e.g., scenarios in which an error in the ordering of a single action early in the plan shifts the position of all subsequent actions).

Recall and relative sequencing measures for pairs (1)–(3) and (2)–(3) are shown in Table 2. We found that both recall and relative sequencing increased with collaborative planning (2) over the baseline of automated planning (1). This behavior was consistent across all domain-complexity conditions. These results provide a promising initial indication that produced plans may better match the human planner's mental model and benefit users' comprehension of machine-generated plans.

Limitations and Future Work

Our collaboration architecture relies upon a visualization environment in order to show the "big picture" of the problem to the user. We believe that if our participants were instead presented with original PDDL problems through raw text (the input language for the planner), their planning performance and the efficacy of their strategies would have suffered. In order to reduce the translation burden, future work exploring domain-independent approaches to automatic generation of visualization environments for planning problems would be paramount. Additionally, we would like to integrate automatic techniques to encode users' strategies as preferences. Such tools would be useful for future IPCs to introduce a track on human-machine collaborative planning; this would allow for a more seamless comparison of planning performance against fully automated planners.

Our model assumes that users are capable planning agents and their provided strategies are sound. Although preferences allow a form of robustness, weights on the preferences are set a priori. Work focused on dynamically assessing the utility of user-provided strategies and appropriately auto-adjusting the weights would be beneficial. We could also estimate a scoring function from features of users' written text or conduct a study eliciting the weight directly from the users themselves.

Conclusion

In this paper, we explored a collaborative planning framework incorporating human users' high-level strategies as preferences to guide the low-level search of automated planners. We collected a human planning dataset and found that useful strategies can be readily provided by non-experts within a short amount of time and successfully encoded as preferences using the PDDL 3.0 language. With the use of benchmark problems from the International Planning Competition, we empirically validated that users' high-level strategies aided automated planners in the generation of better-quality plans within a fixed amount of computation time. Plan improvements generalized across two planning domains: multi-vehicle routing and task allocation. We showed that our approach yields plans that better mimic those that users themselves would generate, signifying potential improvements to plan interpretability and comprehension among users.

References

Anderson, D.; Anderson, E.; Lesh, N.; Marks, J.; Mirtich, B.; Ratajczak, D.; and Ryall, K. 2000. Human-guided simple search. In *Proc. AAAI*.

Baier, J. A.; Bacchus, F.; and McIlraith, S. A. 2009. A heuristic search approach to planning with temporally extended preferences. *Artificial Intelligence* 173(5):593–618.

Benton, J.; Coles, A. J.; and Coles, A. 2012. Temporal planning with preferences and time-dependent continuous costs. In *Proc. ICAPS*.

Bylander, T. 1991. Complexity results for planning. In *Proc. IJCAI*, 274–279.

Cheng, T.-H.; Wei, C.-P.; and Tseng, V. S. 2006. Feature selection for medical data mining: comparisons of expert judgment and automatic approaches. In *Proc. CBMS*, 165–170.

Chrpa, L.; Vallati, M.; and McCluskey, T. L. 2014. *Eight International Planning Competition: Deterministic Part*. Post-competition report.

Clare, A. S.; Cummings, M. L.; How, J. P.; Whitten, A. K.; and Toupet, O. 2012. Operator object function guidance for a real-time unmanned vehicle scheduling algorithm. *Journal of Aerospace Computing, Information, and Communication* 9(4):161–173.

Coles, A. J., and Coles, A. 2011. LPRPG-P: Relaxed plan heuristics for planning with preferences. In *Proc. ICAPS*, 37–45.

Cummings, M. L., and Bruni, S. 2010. Human-automated planner collaboration in complex resource allocation decision support systems. *Intelligent Decision Technologies* 4(2):101–114.

Domshlak, C.; Rossi, F.; Venable, K. B.; and Walsh, T. 2003. Reasoning about soft constraints and conditional preferences: complexity results and approximation techniques. In *Proc. IJCAI*, 215–220.

Du Boulay, B., and Luckin, R. 2001. Modelling human teaching tactics and strategies for tutoring systems. *International Journal of Artificial Intelligence in Education* 12(3):235–256.

Dzifcak, J.; Scheutz, M.; Baral, C.; and Schermerhorn, P. 2009. What to do and how to do it: Translating natural language directives into temporal and dynamic logic representation for goal management and action execution. In *Proc. ICRA*, 4163–4168.

Ferguson, G., and Allen, J. F. 1998. TRIPS: An integrated intelligent problem-solving assistant. In *Proc. AAAI/IAAI*, 567–572.

Ferguson, G.; Allen, J. F.; Miller, B. W.; et al. 1996. TRAINS-95: Towards a mixed-initiative planning assistant. In *Proc. AIPS*, 70–77.

Fox, M., and Long, D. 2003. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research* 20:61–124.

Gerevini, A., and Long, D. 2006. Preferences and soft constraints in PDDL3. In *ICAPS Workshop on Planning with Preferences and Soft Constraints*, 46–53.

Gombolay, M.; Jensen, R.; Stigile, J.; and Shah, J. 2016. Apprenticeship scheduling: Learning to schedule from human experts. In *Proc. IJCAI*.

Hayes-Roth, B. 1980. Human planning processes. Technical report, No R-2670-ONR.

Howey, R.; Long, D.; and Fox, M. 2004. VAL: Automatic plan validation, continuous effects and mixed initiative planning using PDDL. In *IEEE International Conference on Tools with Artificial Intelligence*, 294–301.

Lino, N.; Tate, A.; and Chen-Burger, Y. 2005. Semantic support for visualisation in collaborative AI planning. In *ICAPS Workshop on the Role of Ontologies in AI Planning and Scheduling*.

Manikonda, L.; Chakraborti, T.; De, S.; Talamadupula, K.; and Kambhampati, S. 2014. AI-MIX: using automated planning to steer human workers towards better crowdsourced plans. In *Second AAAI Conference on Human Computation and Crowdsourcing*.

Myers, K. L.; Tyson, W. M.; Wolverton, M. J.; Jarvis, P. A.; Lee, T. J.; and desJardins, M. 2002. PASSAT: A user-centric planning framework. In *Proc. 3rd International NASA Workshop on Planning and Scheduling for Space*, 1–10.

Myers, K. L. 1996. Advisable planning systems. *Advanced Planning Technology* 206–209.

Nau, D. S.; Au, T.-C.; Ilghami, O.; Kuter, U.; Murdock, J. W.; Wu, D.; and Yaman, F. 2003. SHOP2: An HTN planning system. *Journal of Artificial Intelligence Research* 20:379–404.

Nielsen, J. 1994. Enhancing the explanatory power of usability heuristics. In *SIGCHI Conference on Human Factors in Computing Systems*, 152–158.

Nikora, A. P., and Balcom, G. 2009. Automated identification of LTL patterns in natural language requirements. In *International Symposium on Software Reliability Engineering*, 185–194.

Raghavan, H.; Madani, O.; and Jones, R. 2006. Active learning with feedback on features and instances. *Journal of Machine Learning Research* 7(Aug):1655–1686.

Rintanen, J. 2007. Complexity of concurrent temporal planning. In *Proc. ICAPS*, 280–287.

Tate, A. 1994. Mixed initiative planning in O-Plan2. In *Proc. ARPA/Rome Laboratory Planning Initiative Workshop*.

Vadlamudi, S. G.; Chakraborti, T.; Zhang, Y.; and Kambhampati, S. 2016. Proactive decision support using automated planning. *arXiv preprint:1606.07841*.

Wilkins, D. E. 1999. Using the SIPE-2 planning system. *Artificial Intelligence Center, SRI International, Menlo Park, CA*.