Iterative Planning with MUGS Explanations: Exploring the Design Space

Johannes Schmalz[1], David Groß[2], Rebecca Eifler[3], Julián Méndez[4], Raimund Dachselt[4], Stefan Gumhold[2], Jörg Hoffmann[1,5]

[1]Saarland University, Saarland Informatics Campus, Germany
[2]Faculty of Computer Science, TUD Dresden University of Technology, Germany
[3]Work performed while at LAAS-CNRS, Toulouse, France
[4]Interactive Media Lab Dresden, TUD Dresden University of Technology, Germany
[5]German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany
{schmalz, hoffmann}@cs.uni-saarland.de, {david.gross1, julian.mendez2, raimund.dachselt, stefan.gumhold}@tu-dresden.de, rebecca.eifler@laas.fr

Abstract

Oversubscription planning assumes fixed utilities for goals. The trade-offs between such goals are hard to understand and therefore hard to fix a priori. Iterative planning, where users make and modify choices of which subset of goals to enforce based on example plans, was proposed as a remedy. A subsequent line of work enriched this process with minimum-unsolvable-goal-set (MUGS) explanations, which make the dependencies across goals explicit. Here we further explore this design space. First, we ask the question whether, given the MUGS explanations, example plans are actually still useful. Second, list representations of MUGS can be cumbersome when lengthy, so we introduce a compact visual representation based on set-of-set visualizations. We run a user study evaluating these two changes. The results suggest that neither example plans nor our visualization yield significant advantages. These insights can guide further work on iterative planning.

1 Introduction

Oversubscription planning (OSP, e. g. (Smith 2004; Domshlak and Mirkis 2015)) assumes fixed utilities for goals. Yet, as argued by Smith (2012), the trade-offs between such goals are sometimes hard to understand for users, making it difficult to fix an a-priori trade-off. Smith proposes, as a remedy, an iterative planning process where users make and modify choices of which subset of goals to enforce based on example plans. Subsequently, Eifler et al. (2020a; 2020b) (henceforth: Eif20) enriched this iterative planning process with explanations in the form of *minimum unsolvable goal sets (MUGS)*. A MUGS is a set of OSP goals that is unsolvable, but where every strict subset is solvable. MUGS are precomputed prior to the iterative planning process, and are used during that process to explicate dependencies across goals. To evaluate the benefits of this approach, Eifler et al. (2022) (henceforth: Eif22) conducted a large online user study – $N = 100$ for each of three benchmark domains – in Prolific (https://www.prolific.co/), comparing user performance (value of achieved goals) for users with vs. without access to the MUGS explanations. Their results show that users with access to MUGS tend to identify better trade-offs between the plan properties, indicating an improved understanding of the planning task.

Here we further explore the design space for iterative planning with MUGS explanations. First, *are example plans actually still useful when MUGS explanations are present?* This question is relevant as MUGS already contain information about solvability, and as computing example plans – done online as a function of the goals selected by the user – can take substantial time, hindering the interactive nature of the process. Second, Eif22's presentation of MUGS is limited to simply showing a textual list. But there can be many MUGS, making such a list cumbersome to read. *Can this be improved through visualization?* A set of MUGS is a set of sets, a fairly common data structure for which visualization techniques have been proposed (Lex et al. 2014; Alsallakh et al. 2016). We drew from those techniques to design a visual presentation of MUGS in iterative planning.

We follow Eif22's design to conduct a user study evaluating these two changes, challenging both plan generation and the presentation of MUGS as a list. The results show some differences across the three designs compared, but these are per-domain and with low statistical confidence, so no consistent advantages emerge. For our first question, this provides evidence that example plans may indeed not be needed. For our second question, we conclude that a deeper integration of the visualization to the overall iterative planning is needed. Future work should explore an iterative planning process interacting directly with the visual representation, rather than confining it to just one part of the process.

2 Background and Prior Work

Oversubscription Planning (OSP). We consider the OSP setting from Eif20. Our OSP tasks are defined by the tuple $\tau = \langle V, A, c, I, G^{\text{hard}}, G^{\text{soft}}, b \rangle$, where V, A, c, I are the same as in a finite-domain representation (FDR) (Bäckström and Nebel 1995; Helmert 2009) with variables V that induce a set of states S, actions A, action cost c, and initial state I. G^{hard} and G^{soft} can be understood as properties, where all hard goals G^{hard} must be fulfilled and G^{soft} are additional goals that do not need to be achieved but are desired. A solution for such tasks is a sequence of actions (called a plan) that can be applied to I, and lead to a state that satisfies

G^{hard}. Additionally, b is an action-cost budget that disallows plans whose cost exceeds b. In contrast to typical OSP, we do not have utilities over G^{soft}, because these represent properties whose values are not directly comparable — their comparative values are determined online by domain experts.

Minimal Unsolvable Goal Subset (MUGS). Given an OSP planning task τ and $G \subseteq G^{\text{soft}}$, we can *enforce* G by modifying τ with $G^{\text{hard}} \cup G$ as hard goals. $G \subseteq G^{\text{soft}}$ is a MUGS (Eif20) if the task with G enforced is unsolvable, but it is solvable for any $G' \subsetneq G$. Intuitively, this means that for each MUGS G, we must stop enforcing at least one contained soft goal $g \in G$ in order to get a solvable problem. MUGS can be used to analyse dependencies between different subsets of soft goals, and Eif22 built their iterative planning process on this idea.

Iterative Planning Process. Eif22's iterative planning process works in the following steps: (1) the user selects $G_{\text{enf}} \subseteq G^{\text{soft}}$; (2) a planner is called on the task with $G^{\text{hard}} \cup G_{\text{enf}}$ enforced; (3) if a plan π was found, the user can ask about remaining goals $p \in G^{\text{soft}}$ "why does π not satisfy p?" There are two cases: either $G_{\text{enf}} \cup \{p\}$ can be safely enforced, and the answer is "you can enforce this goal," or $G_{\text{enf}} \cup \{p\}$ is unsatisfiable and the answer is a list of MUGS with the semantic "you must remove from $G_{\text{enf}} \cup \{p\}$ at least one goal from each MUGS before $G_{\text{enf}} \cup \{p\}$ is satisfiable." Otherwise, if no plan was found, the user can ask "why is there no plan?" The answer is again a list of MUGS with the semantic "you must remove from G_{enf} at least one goal from each MUGS before G_{enf} is satisfiable." Next, the user may choose to loop to step (1) or exit. The MUGS needed to answer the questions in step (3) are computed before the iterative planning process starts, using algorithms from Eif20.

Eif22's User Study Design.[1] Eif22's user study was designed to determine whether Eif20's explanation facility helps users to produce higher quality goal selection in the iterative planning framework. Participants were divided randomly into two groups: $\mathbb{Q}+$ and $\mathbb{Q}-$ where participants either did, or did not have access to the explanation facility, respectively. In each session, a participant was asked to do the following for one of the considered domains: (1) read textual domain and tool description; (2) solve a small problem from the domain to get familiar with the tool; (3) solve the evaluation problem from the domain to the best of their ability within 30 minutes; (4) answer a questionnaire about how useful they found the tool.

Three domains were considered: (1) *Transport*. Inspired by the IPC NoMystery benchmark. The goals specify locations that delivery trucks need to visit or where packages need to be delivered, and more complex goals specify in what order packages should be delivered. Limited fuel makes it impossible to achieve all goals. (2) *Parent's Afternoon*. A parent must decide which errands to fit into their afternoon, e. g., driving their kids to soccer practice and going shopping. All errands take time, and some errands need to be performed at specific times, e. g., soccer practice begins at 6pm. The timing constraints make it impossible to do everything. (3) *Mars Rover*. Based on the IPC Rovers domain, the goals correspond to a Mars Rover's daily scientific missions, such as taking pictures or samples. The rover is constrained by limited battery-life and data-storage capacity, and time windows when certain tasks can be completed, e. g., pictures can only be taken when there is sufficient sunlight. One small problem for training, and one larger problem for evaluation were selected from each domain. To be able to quantitatively measure user performance, Eif20 assigned a utility to each soft goal in the evaluation problem, and asked users to maximize the summed-up utility. This is not what happens in the targeted application scenario (where users need to make up their minds about which soft goals they prefer) but, as Eif20 argue, this is meaningful for evaluation as test persons need to understand goal dependencies to perform well. Thus, the quantitative measurement is goal utility achieved over time in the evaluation problem.

The questionnaire consisted of two parts. The first half were Likert scale questions, where participants gave a response on a 7-point scale. Some example questions: *"How difficult was the task for you?"* and *"The ability to ask questions helped you."* The second half of the questionnaire consisted of questions with free-text answers, e.g., *"Is there anything we can improve, with respect to the interface?"*

Participants were recruited with Prolific (Palan and Schitter 2018). Eif22 made sure to have at least 50 participants per group for each domain. Participants that made meaningful progress in the evaluation problem of one domain were invited to perform the experiment for other domains, with the intention of allowing participants to get familiar with the tool. For each session, users were given a base pay of $5£$. Additionally, as motivation to look for high-quality solutions, there was bonus reward of up to $2.50£$, earned by obtaining a better goal selection (to earn the maximum reward of $2.50£$, the participant had to achieve the maximum possible utility).

Results of Eif22's User Study. Eif22's results showed that, overall, the explanation facility helped users. The mean utility over domains was higher for $\mathbb{Q}+$ than $\mathbb{Q}-$ across the entire timeline. Breaking down by domains: in Parent's Afternoon the advantage was statistically significant over the whole timeline; in the other domains it was only statistically significant over parts of the timeline. Participants' subjective satisfaction tended to be in favor of the explanation facility, but with no statistically significant results. Eif22 observed that familiarity with the tool can have a significant impact on performance: the results for $\mathbb{Q}+$ in Mars Rover showed that re-invited participants performed better than those who saw the tool for the first time, with higher mean utility over the entire timeline, and the advantage was statistically significant towards the end of the timeline.

3 Removing the Example Plans

Eif22's questions for interacting with the soft goals are framed in terms of a plan, but we argue that plans are probably not needed. Each time that a user makes a selection of enforced goals G_{enf}, Eif22's tool attempts to compute a new plan that satisfies $G^{\text{hard}} \cup G_{\text{enf}}$. The plan itself is not shown to the user; rather, the user is shown which goals

[1] A more detailed description of the user study can be found in Eif22's main author's dissertation (Eifler 2025).

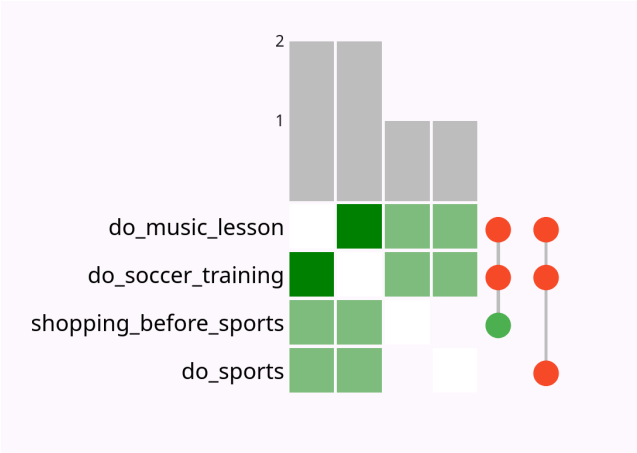

Figure 1: Visualization of a small MUGS instance from the Parent's Afternoon domain. The bar chart at the top and central matrix view count the occurrence of individual goals and goal pairs respectively. The beads chart to the right lists the actual MUGS.

were satisfied (if a plan was found), and any further interaction is done via the explanations, which are extracted from precomputed MUGS. Thus, the main function of the planning step is to determine if the enforced goals are satisfiable. However, with access to the precomputed MUGS, we do not require a planner because $G^{\text{hard}} \cup G_{\text{enf}}$ is satisfiable iff $\exists G \in \text{MUGS}$ s.t. $G_{\text{enf}} \subsetneq G$. With this, we can use the precomputed MUGS to efficiently compute satisfiability (takes a few seconds), instead of computing plans from scratch (takes a mean of 25 seconds per plan in our experiments, up to 5 minutes). Based on this, we introduce a modified interface where the user can ask "why is p not satisfied when enforcing G_{enf}?" and "why is enforcing G_{enf} unsatisfiable?" instead of asking analogous questions about a plan.

Note that computing an example plan has the potential advantage that the plan may satisfy additional goals beyond those enforced by the user. But this is purely incidental (except for the rare case in which the enforced goals G_{enf} entail fulfillment of other goals). Therefore, our hypothesis is that access to such additionally satisfied goals does not yield a systematic advantage.

4 Visualizing a Set of MUGS

Our visualization encodes the MUGS and goals in a compact representation so that users can quickly identify conflicting goals. Since the MUGS as computed by the solver do not possess any additional properties other than the contained goals, the visualization focuses on the occurrence and combination of goals across sets. A small instance of our visualization is shown in Figure 1.

We propose a tabular arrangement loosely inspired by (Lex et al. 2014), incorporating three aligned charts: a central *matrix view*, a *beads view* to its right, and a *frequency bar chart* on top. Each matrix row is assigned a single goal. The beads view lists MUGS in columns aligned to the matrix, with visual markers (colored circles) indicating when a goal is contained in a set. Vertical connections (grey lines) further emphasize the assignment of goals to their respective MUGS. The color of the circles encodes goal *types*: an optional domain-dependent property which categorizes goals into, e.g., activities (*do_music_lesson*) or constraints (*shopping_before_sports*), and might be relevant to the user's choices.

The matrix and bar chart show computed properties that aim to inform the user of any conflicts among the goals. Examining goals individually, the frequency bar chart gives the number of involved conflicts, i.e., the number of MUGS in which the relevant goal occurs. This provides a user with information about the overall conflict potential of a single goal. The conflict count also governs the horizontal arrangement of bars and vertical order of goals, showing more conflict-rich goals at the left and top respectively. However, MUGS are generally comprised of more than one goal and, as such, the combination of certain goals is the main cause of conflicts in the first place. Considering this, we additionally count the pairwise shared conflicts between two goals and plot the results in the central matrix view. Each matrix cell (diagonal excluded) represents a pair of goals with the color coding depicting the number of pairwise shared conflicts, i.e., in how many MUGS the relevant pairs of goals occur (darker color means higher count). Goals are mirrored along the matrix diagonal to map to the bar chart above. The matrix chart thus offers information on which pairs of goals often appear together and may hint at large overlaps in resource requirements that cannot be satisfied or leave too little resources remaining for the rest of the goals.

Note that the matrix is symmetric, i.e., both the lower and upper triangular matrix currently represent the same correlation. In next versions, we will explore showing different pairwise measures. Furthermore, we currently only count occurrences of *pairs* of goals, which is intuitively suited to the 2D matrix. However, future work might research the usefulness of counting larger goal combinations, e.g., on user demand.

To aid the visual assignment of chart elements to goals, hovering the mouse pointer over any element will highlight all related entries in the charts. This is especially useful in the matrix chart, as this will highlight the two goals represented by the hovered cell, supporting quick cross-reference.

Our visualization is integrated directly into the iterative planning tool, replacing the previous textual representation. When an unsolvable set G_{enf} of enforced goals occurs (see Section 2: either from direct selection by the user, or when asking a question about an additional goal p to enforce), the relevant MUGS are supplied to a separate component that builds the visual representation. Users can then inspect the charts and gain understanding of the existing conflicts and involved goals, aiding the decision on which goals are best to forego to make the plan satisfiable again. As the individual and pairwise conflict counts together indicate goals with high potential of conflict, a user may consider these first while selecting goals to no longer be enforced in the next iteration step.

5 User Study Design and Results

User Study Design. We compare three variants of the tool: (1) the version used by Eif22 without a visualization and computing plans for each iteration step, (2) our extension with a visualization, (3) our extension where plans are not computed (and without the visualization). This gives us three groups: V−P+, V+P+, V−P−, respectively. We largely follow the user study design of Eif22, given that we build on their work. We used the same three domains, with the same problems for Parent's Afternoon and Mars Rover, and a variant for Transport.

The main departure from Eif22's user study design was to the structure of experiment sessions. We asked participants to: (1) watch an instructional video and pass comprehension checks about it, (2) solve a small instance of Transport to get familiar with the tool, (3) solve all the evaluation instances to the best of their ability, with 20 minutes per instance, for Transport, Parent's Afternoon, and finally Mars Rover (in that order), (4) fill in the questionnaire. Each participant used the same tool variant for all tasks, according to their assigned group. The comprehension check filtered our participants who did not understand the instructions, and were unlikely to make meaningful progress in the study. In contrast to Eif22, our step (3) had participants solve an evaluation problem from each domain in one sitting. This builds on Eif22's observation that there was a significant difference between reinvited participants already familiar with the tool, and new participants who saw it for the first time — we wanted to focus on users familiar with the tool. Otherwise, there is the danger that our study evaluates how intuitive the variants are to learn, rather than how helpful the variants are to experienced users.[2] We gave a base reward of $18£$ and a bonus reward of up to $3£$ per evaluation problem (so a maximum bonus reward of $9£$ is possible).

We removed data for a participant in a specific task when the participant asked fewer than 2 questions on that evaluation task (i.e., did not make use of the tool). We had 30 participants per group; after applying our filter, for (V−P+, V+P+, V−P−) there remained (28, 28, 29) participants for Transport, (29, 28, 28) for Parent's Afternoon, and (26, 28, 30) for Mars Rover. Note that each participant only filled in one questionnaire, not one per task. We kept the questionnaire data for all 30 participants per group — we filtered all participants that asked 0 questions over all evaluation tasks, but there were no such participants in our data.

User Study Results. We evaluate the effectiveness of the three tool variants by comparing the max. utility achieved over time by each group (V−P+, V+P+, V−P−). The mean, 95% CI, and median results per group are shown in Figure 2 for each task. Overall, there is no significant difference between the performance of the three groups, except that V−P+ performed better than V−P− in Transport.

In our analysis, we followed Eif22: with 1-minute intervals, we considered each participant's maximum achieved utility up to that point in time, and then applied the *Student's t-test* and *Wilcoxon rank-sum test* (also known as the *Mann-Whitney U* test) to these values, comparing the baseline V−P+ pairwise with V+P+ and V−P−. We consider the difference between groups to be significant when both *Student's t-test:* $p < 0.05$ and *Wilcoxon test:* $p < 0.05$. The only significant difference in our data is on Parent's Afternoon between V−P+ and V−P− from 12 minutes onwards, with *Student's t-test:* $p = 0.022$ and *Wilcoxon test:* $p = 0.025$ at 12 minutes, and no larger p-values later, in favor of the baseline V−P+.

For the questionnaire data, we compared the Likert scale responses with the same two tests, and similarly only consider a difference between groups to be significant if both $p < 0.05$ for both tests. The only case where this occurs in our data, is in response to *In solvable steps, the question "Why is 'goal X' not satisfied?" helped you* with *Student's t-test:* $p = 0.019$ and *Wilcoxon test:* $p = 0.023$ in favor of V+P+ over V−P+. The free text responses were diverse, without any clear themes or consistent criticisms.

Comparing V−P+ with V−P−. In the Transport task, V−P+ performs better. Recall that the only potential advantage of the tool for V−P+ over V−P− is that there may be additionally satisfied goals, which suggests that the additionally satisfied goals are helpful in achieving higher utility in this particular task, but this advantage does not appear in the other tasks. This likely occurs because in this task, more than in the other tasks, it happens in more cases that a participant makes a selection of goals that are suboptimal, but with additionally satisfied goals it becomes an optimal solution. There is no significant difference in the other tasks, nor in the questionnaire. This suggests that having additionally satisfied goals on the other tasks is approximately equally useful as solving the problem faster without additionally satisfied goals.

Comparing V−P+ with V+P+. There was no discernible difference in performance on the evaluation tasks between the two groups. One possible factor why V+P+ did not yield the expected improvement, is that our non-expert participants potentially did not fully understand the visualization, and therefore failed to make full use of the information it provides. Another possible factor, is that the tasks were relatively small, so the visualization's additional information may not have been necessary. In terms of the questionnaire, there was no significant difference except that participants in V+P+ found the response to *"Why is 'goal X' not satisfied?"* for solvable problems more useful than those in V−P+. It is surprising that there was significant difference here, but not for *In unsolvable steps, the question "Why is the selection of enforced goals unsolvable?" helped you,* given that the visualization is the same, but there are usually more MUGS displayed as a response to the latter question. Without making strong claims, this hints that participants do in fact find the visualization more helpful at a subjective level, but we would require a larger sample size to confirm.

[2]The focus of our work is on the usefulness of the tool in the hands of users who are familiar with it. For example, this line of research was initiated by Smith (2012) in the context of planning Mars Rover missions, which would be performed daily by specialists — clearly, they would be familiar with the tool they use.

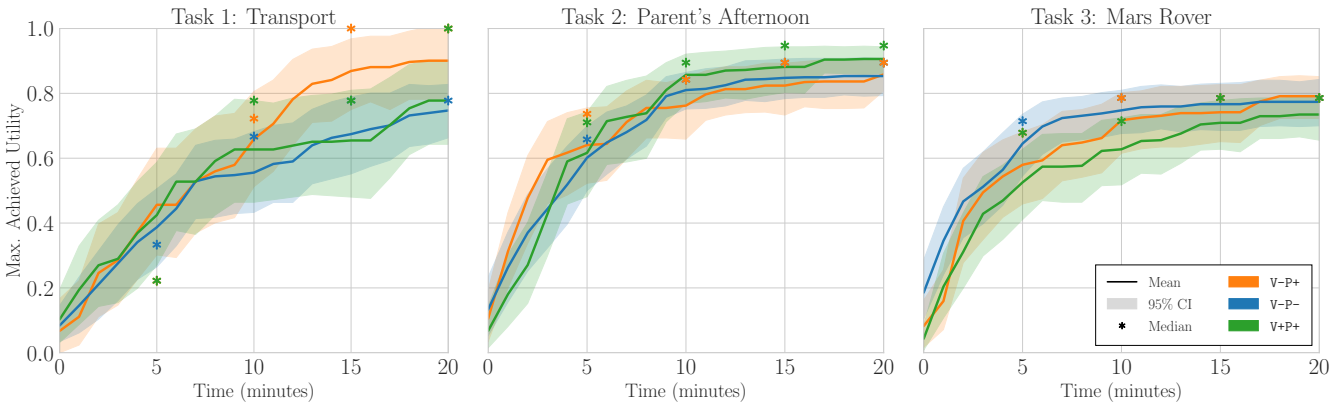

Figure 2: Utility over time: x-axis is time (mins) and y-axis is max utility achieved up to that point in time.

6 Conclusion

Iterative planning has been introduced to enable users to identify their preferred trade-offs between goals in oversubscription planning. MUGS explanations have been shown to help with that, but some design choices deserve further attention. Here we focused on the use of example plans, and the presentation of MUGS. The results indicate that 1. example plans offer no advantage in the presence of MUGS, and 2. using visualization only for MUGS presentation does not seem useful.

Future work should explore an iterative planning process interacting directly with the visual representation, rather than confining it to just one part of the process. This will require interaction paradigms for our visualization, e.g. clicking on goals to enforce or de-enforce them, permitting to explore the effects of the selection. Also, there is still room for incorporating more information into the visualisation, e.g. including a second pairwise measure in the matrix, or showing information about goal subsets of size > 2, as previously mentioned. Furthermore, future work should investigate the effect of additionally satisfied goals. In particular, we note that algorithms for computing MUGS also compute maximal solvable goal subsets (Eifler et al. 2020b), which can be used as maximal additionally satisfied goals.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 389792660 – TRR 248 (see https://perspicuous-computing.science).

References

Alsallakh, B.; Micallef, L.; Aigner, W.; Hauser, H.; Miksch, S.; and Rodgers, P. 2016. The State-of-the-Art of Set Visualization. *Computer Graphics Forum*, 35(1): 234–260.

Bäckström, C.; and Nebel, B. 1995. Complexity Results for SAS$^+$ Planning. *Computational Intelligence*, 11(4): 625–655.

Domshlak, C.; and Mirkis, V. 2015. Deterministic Oversubscription Planning as Heuristic Search: Abstractions and Reformulations. *JAIR*, 52: 97–169.

Eifler, R. 2025. *Explaning Goal Conflicts in Oversubscription Planning*. Ph.D. thesis, Saarland University, Saarland Informatics Campus, Germany.

Eifler, R.; Cashmore, M.; Hoffmann, J.; Magazzeni, D.; and Steinmetz, M. 2020a. A New Approach to Plan-Space Explanation: Analyzing Plan-Property Dependencies in Oversubscription Planning. In *AAAI*.

Eifler, R.; and Hoffmann, J. 2022. In *Proceedings of the 32nd International Conference on Automated Planning and Scheduling (ICAPS'22)*, 687–691.

Eifler, R.; Steinmetz, M.; Torralba, A.; and Hoffmann, J. 2020b. Plan-Space Explanation via Plan-Property Dependencies: Faster Algorithms & More Powerful Properties. In *IJCAI*, 4091–4097.

Helmert, M. 2009. Concise Finite-Domain Representations for PDDL Planning Tasks. *AI*, 173: 503–535.

Lex, A.; Gehlenborg, N.; Strobelt, H.; Vuillemot, R.; and Pfister, H. 2014. UpSet: Visualization of Intersecting Sets. *IEEE Transactions on Visualization and Computer Graphics*, 20(12): 1983–1992.

Palan, S.; and Schitter, C. 2018. Prolific.ac — A subject pool for online experiments. *Journal of Behavioral and Experimental Finance*, 17: 22–27.

Smith, D. 2012. Planning as an Iterative Process. In *AAAI*, 2180–2185.

Smith, D. E. 2004. Choosing Objectives in Over-Subscription Planning. In *ICAPS*, 393–401.