Who Needs These Operators Anyway: Top Quality Planning with Operator Subset Criteria

Michael Katz and **Shirin Sohrabi**

IBM Research, Yorktown Heights, NY, USA
michael.katz1@ibm.com, ssohrab@us.ibm.com

Abstract

Top-quality planning in general and quotient top-quality planning in particular deal with producing multiple high-quality plans while allowing for their efficient generation, skipping equivalent ones. Prior work has explored one equivalence relation, considering two plans to be equivalent if their operator multi-sets are equal. This allowed omitting plans that are reorderings of previously found ones. However, the resulting sets of plans were still large, in some domains even infinite.

In this paper, we consider a different relation: two plans are related if one's operator multiset is a subset of the other's. We propose novel reformulations that forbid plans that are related to the given ones. While the new relation is not transitive and thus not an equivalence relation, we can define a new subset top-quality planning problem, with finite size solution sets. We formally prove that these solutions can be obtained by exploiting the proposed reformulations. Our empirical evaluation shows that solutions to the new problem can be found for more tasks than unordered top-quality planning solutions. Further, the results shows that the solution sizes significantly decrease, making the new approach more practical, particularly in domains with redundant operators.

Introduction

Top-quality planning deals with generating all high-quality plans up to a certain bound (Katz, Sohrabi, and Udrea 2020). The need for producing such a collection of plans is well established for many applications, including plan recognition (Sohrabi, Riabov, and Udrea 2016), malware detection (Boddy et al. 2005), business process automation (Chakraborti et al. 2020), and automated machine learning (Katz et al. 2020). Top-quality planning serves as a basis for solving other computational problems, such as quality-aware diverse planning (Nguyen et al. 2012; Vadlamudi and Kambhampati 2016; Katz, Sohrabi, and Udrea 2022). In these and other applications, the choices of planning models, whether avoidable or not, may have unintended effects on plans. These include (zero-cost) loops, unnecessary repeated operator applications, continuing after the goal is reached.

While producing *all* high-quality plans can be challenging, and even impossible when there are infinitely many such plans, one practical approach suggested to generate all plans that are not equivalent when ignoring operator ordering (Katz, Sohrabi, and Udrea 2020). However, this so-called unordered top-quality planning solution can still be very large. Further, this approach only reduces the solution size by a finite number, not helping in domains with infinite size solutions. Such domains are not uncommon, and even appear at International Planning Competitions (IPC). Also, previous work did not address unnecessary long plans.

In this work, we propose a novel computational problem under the umbrella of top-quality planning that allows to further restrict the solution set. We consider the following relation: a plan is related to another if its operator multiset is a subset of the other's. We propose a novel planning task reformulation that forbids plans whose operator multisets are supersets of the given ones. We prove that the solution sets of the new *subset top-quality planning* problem are of finite size. We propose a similar to the previously proposed solution scheme, iteratively finding and forbidding plans, exploiting the new reformulation. We empirically evaluate our approach, depicted by top-MSQ, comparing to the unordered top-quality planner, showing a better coverage and significantly reducing the solution size. Additionally, we propose a reformulation that forbids plans that are supersets as sets rather than multisets, depicted by top-SQ. This stricter reformulation top-SQ, can be preferred if it is important to forbid alternative but similar ways of achieving subgoals. Our evaluation of the second method (depicted by top-SQ) shows that while being simpler, and thus easier to solve, it results in very similar solution sizes to the one of top-MSQ. Both proposed approaches show promising results, in particular, being able to generate subset top-quality solutions in domains with infinite size unordered top-quality solutions. While each of the approaches to top-quality planning has its own motivating scenario, the two new ones show a great promise, particularly in domains with redundant and zero-cost operators.

Background

A *planning task* in the SAS$^+$ formalism (Bäckström and Nebel 1995), extended with operator costs, is $\Pi = \langle \mathcal{V}, \mathcal{O}, s_0, s_\star, cost \rangle$ with a finite set of finite-domain *state* *variables* $\mathcal{V}$, a finite set of *operators* $\mathcal{O}$, an *initial state* s_0, and the *goal* $s_\star$. Each variable $v \in \mathcal{V}$ is associated with a finite domain $\mathcal{D}(v)$ of variable values. These variable and value pairs are called *facts*. A *partial assignment* p maps a subset of variables $\mathcal{V}_p \subseteq \mathcal{V}$ to values in their domains. For a variable $v \in \mathcal{V}$ and partial assignment p, the value of v in p is denoted by $p[v]$ if $v \in \mathcal{V}_p$ and we say $p[v]$ is *undefined* if $v \notin \mathcal{V}_p$. A partial assignment s with $\mathcal{V}_s = \mathcal{V}$ is called a *state*. State s is *consistent* with partial assignment p if they agree on all variables in $\mathcal{V}_p$, denoted by $p \subseteq s$. s_0 is a state and $s_\star$ is a partial assignment. A state s is called a *goal state* if $s_\star \subseteq s$ and the set of all goal states is denoted by $\mathcal{S}_{s_\star}$. Each operator o in $\mathcal{O}$ is a pair $\langle pre(o), \mathit{eff}(o)\rangle$ where $pre(o)$ is a partial assignment called *precondition* and $\mathit{eff}(o)$ is a partial assignment called *effect*. Further, o has an associated natural number $cost(o)$, called *cost*. An operator o is applicable in state s if $pre(o) \subseteq s$. Applying operator o in state s results in a state denoted by $s[\![o]\!]$ where $s[\![o]\!][v] = \mathit{eff}(o)[v]$ for all $v \in \mathcal{V}_{\mathit{eff}}$ and $s[\![o]\!][v] = s[v]$ for all other variables. An operator sequence $\pi = \langle o_1, \cdots, o_n\rangle$ is applicable in state s if there are states $s_0, \cdots, s_n$ such that o_i is applicable in s_{i-1} and $s_{i-1}[\![o_i]\!] = s_i$ for $0 \leq i \leq n$. We denote s_n by $s[\![\pi]\!]$. An operator sequence with $s_0[\![\pi]\!] \in \mathcal{S}_{s_\star}$ is called a *plan*. The cost of a plan π, denoted by $cost(\pi)$ is the summed cost of its operators. For a planning task Π, the set of all plans is denoted by $\mathcal{P}_\Pi$. A plan π is *optimal* if its cost is minimal among all plans in $\mathcal{P}_\Pi$. For a plan π, $\mathrm{MS}(\pi)$ denotes the multiset of operators in π. The *multiplicity*, that is, the number of occurrences, of the element $o \in X$ in a multiset is denoted by $m_X(o)$. For multisets X, Y, we say that X is a subset of Y ($X \subseteq Y$) if $m_X(o) \leq m_Y(o)$ for all o. $X = Y$ if $X \subseteq Y$ and $Y \subseteq X$, and $X \subset Y$ if $X \subseteq Y$ and $X \neq Y$.

The definitions of top-quality planning problems were given by Katz, Sohrabi, and Udrea (2020).

Top-quality: Given a planning task Π and a natural number q, find the set of plans $P = \{\pi \in \mathcal{P}_\Pi \mid cost(\pi) \leq q\}$.

Quotient top-quality: Given Π, an equivalence relation N over its set of plans $\mathcal{P}_\Pi$, and a natural number q, find a set of plans $P \subseteq \mathcal{P}_\Pi$ such that $\bigcup_{\pi \in P} N[\pi]$ is the solution to the top-quality planning problem.

Unordered top-quality: Given Π and a natural number q, find a set of plans $P \subseteq \mathcal{P}_\Pi$ such that P is a solution to the quotient top-quality planning problem under the equivalence relation $\mathrm{U}_\Pi = \{(\pi, \pi') \mid \pi, \pi' \in \mathcal{P}_\Pi, \mathrm{MS}(\pi) = \mathrm{MS}(\pi')\}$.

Forbidding Plans as Super-Multisets

We start by defining the computational problem of interest as well as a method for solving it. First, let $R_\subset = \{(\pi, \pi') \mid \mathrm{MS}(\pi) \subset \mathrm{MS}(\pi')\}$ denote the relation defined by the subset operation over plan operator multisets.

Definition 1 (subset top-quality planning problem) *Given a planning task* Π *and a natural number* q*, find a set of plans* $P \subseteq \mathcal{P}_\Pi$ *s.t. (i)* $\forall \pi \in P$*,* $cost(\pi) \leq q$*, and (ii)* $\forall \pi' \in \mathcal{P}_\Pi \setminus P$ *with* $cost(\pi') \leq q$*,* $\exists \pi \in P$ *s.t.* $(\pi, \pi') \in R_\subset$*.*

In words, a plan π with $cost(\pi) \leq q$ may not be part of the solution to the subset top-quality planning problem only if its subset is part of the solution. Note, while a top-quality and unordered top-quality solutions are also subset top-quality solutions, we are interested in finding smallest (in the number of plans) such solutions. While unordered top-quality solutions can be of infinite size, smallest subset top-quality planning solutions are always finite.

Theorem 1 *Given a planning task* Π *and a natural number* q*, a smallest subset top-quality planning problem solution* P *is of finite size.*

Proof: For a plan π that contains a cycle, if π' is obtained from π by removing all cycles, then we have $(\pi', \pi) \in R_\subset$. Thus, P is a subset of the set of all cycle-free plans. Since Π has only a finite number of states, the number of cycle-free plans is also finite. $\square$

We now turn our attention to finding smallest subset top-quality planning problem solutions. Following the line of work on reformulating planning tasks to forbid a set of plans (Katz et al. 2018; Katz and Sohrabi 2020; Katz, Sohrabi, and Udrea 2020, 2022), we present a reformulation that, given a set of plans, forbid all plans that are supersets (as operator multisets) of some plan in the given set. In other words, the reformulation preserves all plans that are *not* supersets of all the plans in the given set. That allows for both ignoring plan reorderings and unnecessary operators. We call it Forbidding Multiple Plans as Operator Super-Multi-Sets (FOSMS).

Definition 2 *Let* $\Pi = \langle \mathcal{V}, \mathcal{O}, s_0, s_\star, cost\rangle$ *be a planning task,* $\mathcal{M} = \{\mathcal{M}_1, \dots \mathcal{M}_k\}$ *be a set of operator multisets, and* $U \subseteq \mathcal{O}$ *be the set of all operators in the multisets in* $\mathcal{M}$*. The task* $\Pi^-_{\mathcal{M}^+} = \langle \mathcal{V}', \mathcal{O}', s'_0, s'_\star, cost'\rangle$ *is defined as follows.*

- $\mathcal{V}' = \mathcal{V} \cup \{\overline{v}\} \cup \{\overline{v}_o \mid o \in U\}$*, where the variable* $\overline{v}$ *has the domain* $dom(\overline{v}) = \{0 \dots k\}$ *and* $dom(\overline{v}_o) = \{0, \dots, \overline{m}(o)\}$*, where* $\overline{m}(o) = \max_{i=1}^{k}\{m_{\mathcal{M}_i}(o)\}$*,*
- $\mathcal{O}' = \{\overline{o}_i \mid o \in \mathcal{O}, 0 \leq i \leq \overline{m}(o)\} \cup \{o_{i,j} \mid o \in \mathcal{M}_j, 1 \leq j \leq k, 0 \leq i < m_{\mathcal{M}_j}(o)\}$*, where*

$$pre(\overline{o}_i) = pre(o) \cup \{\langle \overline{v}, 0\rangle\} \cup \{\langle \overline{v}_o, i\rangle \mid o \in U\},$$
$$\mathit{eff}(\overline{o}_i) = \mathit{eff}(o) \cup \{\langle \overline{v}_o, i+1\rangle \mid o \in U, i < \overline{m}(o)\},$$
$$o_{i,j} = \langle s_\star \cup \{\langle \overline{v}_o, i\rangle, \langle \overline{v}, j-1\rangle\}, \{\langle \overline{v}, j\rangle\}\rangle, \text{and}$$

$cost'(\overline{o}_i) = cost(o)$*,* $cost'(o_{i,j}) = 0$*,*
- $s'_0 = s_0 \cup \{\langle \overline{v}, 0\rangle\} \cup \{\langle \overline{v}_o, 0\rangle \mid o \in U\}$*, and*
- $s'_\star = s_\star \cup \{\langle \overline{v}, k\rangle\}$*.*

In words, the reformulation keeps track of the original operators from the planning task Π applied on the way to the goal (i.e., $\overline{o}_i$) with the extra effects counting the number of applications of each operator applied to reach the goal. Once the goal is reached, if there is an operator with the number of applications lower than the number of appearances in a multiset, then the found plan is not a superset of that multiset. The reformulation then switches to the second phase in which it ensures that there exists at least one operator o from the multiset $\mathcal{M}_j$, whose corresponding operator $\overline{o}_i$ was not applied and now $o_{i,j}$ can be applied. Note, the goal can only be reached if $\langle \overline{v}, k\rangle$ is true in addition to the original goal, $s_\star$. That is the goal can be reached by applying the additional operators $o_{i,j}$, which are applicable only when the original goal was achieved and the corresponding original operator was not applied $m_{\mathcal{M}_j}(o)$ times.

Algorithm 1: Iterative subset top-quality planning.

```
Input: Planning task $\Pi$, quality bound $q$
  $P \leftarrow \emptyset$, $\Pi' \leftarrow \Pi$, $\pi \leftarrow$ shortest cost-optimal plan of $\Pi'$
  while $cost(\pi) \leq q$ do
      $\overline{\pi} \leftarrow$ MAPPLANBACK($\pi$)
      $P \leftarrow P \cup \{\overline{\pi}\} \cup \{\pi' \mid \pi' \sim \overline{\pi}, \mathrm{MS}(\pi') \neq \mathrm{MS}(\overline{\pi})\}$
      $\Pi' \leftarrow \Pi^{-}_{\mathcal{M}+}$, where $\mathcal{M} = \{\mathrm{MS}(\pi') \mid \pi' \in P\}$
      $\pi \leftarrow$ shortest cost-optimal plan to $\Pi'$
  return $P$
```

Consider an example task Π with $\mathcal{O} = \{a, b, c, d\}$, plans aba and ca and their corresponding multisets $\mathcal{M} = \{\{a, a, b\}, \{a, c\}\}$. The reformulated task $\Pi^{-}_{\mathcal{M}+}$ has extra variables: ternary $\overline{v}$ and $\overline{v}_a$, as well as binary $\overline{v}_b$ and $\overline{v}_c$. $\mathcal{O}'$ includes multiple copies of the original operators that appear in U. Here, we have $\mathcal{O}' = \{\overline{a}_0, \overline{a}_1, \overline{a}_2, \overline{b}_0, \overline{b}_1, \overline{c}_0, \overline{c}_1, \overline{d}_0\} \cup \{a_{0,1}, a_{0,2}, a_{1,1}, a_{1,2}, b_{0,1}, c_{0,2}\}$. Consider a plan $\pi = abd$ for Π and a sequence of operators $\overline{\pi} = \overline{a}_0\overline{b}_0\overline{d}_0 a_{1,1} c_{0,2}$. Note that $\overline{\pi}$ is a plan for $\Pi^{-}_{\mathcal{M}+}$: $\overline{a}_0\overline{b}_0\overline{d}_0$ achieves the original goal, while $a_{1,1}$ and $c_{0,2}$ are changing the value of $\overline{v}$ from 0 to 1 and then to 2. Thus, π is not forbidden.

To solve the subset top-quality planning problem, one can exploit the reformulation FOSMS in Definition 2 in a fashion very similar to the one suggested by Katz, Sohrabi, and Udrea (2020): find a plan, (possibly) extend it to a set of plans with the help of structural symmetries (Shleyfman et al. 2015), reformulate the original planning task using the set of plans found so far, extend the set by solving the reformulated task optimally, adding the cost optimal plan (and possibly its symmetric plans to the set), continue until there are no more plans of cost lower or equal to q. The detailed scheme is depicted in Algorithm 1.

Theorem 2 *Algorithm 1 is sound and complete for subset top-quality planning when using cost-optimal planners that find shortest cost-optimal plans.*

Proof: First, according to Theorem 1, Algorithm 1 will terminate and return a set of plans.

Let $P = \{\pi_1, \dots, \pi_k\}$ be plans for Π with their corresponding multisets $\mathcal{M} = \{\mathrm{MS}(\pi_1), \dots, \mathrm{MS}(\pi_k)\}$. Let $\pi = a^1 \dots a^n$ be some plan for Π such that $\mathrm{MS}(\pi_i) \not\subset \mathrm{MS}(\pi)$ for $1 \leq i \leq k$. We show that there exists a corresponding to π plan $\overline{\pi}$ for $\Pi^{-}_{\mathcal{M}+}$. Let $\overline{\pi}_1 = \overline{a^1}_{i_1} \dots \overline{a^n}_{i_n}$ be the applicable sequence of operators from $\mathcal{O}'$ that corresponds to $\pi = a^1 \dots a^n$ and s_1 be the end state of applying $\overline{\pi}_1$ in s'_0. Then, $s_\star \subseteq s_1$ and $s_1[\overline{v}] = 0$. Since $\mathrm{MS}(\pi_i) \not\subset \mathrm{MS}(\pi)$, there exists an operator $b^i \in \mathrm{MS}(\pi_i)$ such that $b^i \notin \mathrm{MS}(\pi)$. Thus, we have $s_1[\overline{v}_{b^i}] = 0$, $1 \leq i \leq k$. Then, $\overline{\pi}_2 = b^1_{0,1} \dots b^k_{0,k}$ is applicable in s_1 and results in a state s_2 such that $s_\star \subseteq s_2$ and $s_2[\overline{v}] = k$. In other words, $\overline{\pi} = \overline{\pi}_1\overline{\pi}_2$ is a plan for $\Pi^{-}_{\mathcal{M}+}$.

The relation $R_\subset$ is reflexive, antisymmetric, and transitive and thus defines a partial order over plans. Let π and π' be two plans such that $(\pi, \pi') \in R_\subset$. Then $cost(\pi) \leq cost(\pi')$ and the plan π is strictly shorter than π' and therefore a planner that finds shortest among the optimal plans will find π before π'. Thus, Algorithm 1 finds plans that are minimal under the partial order defined by $R_\subset$ and will terminate when it finds all such plans of cost less or equal q. □

Forbidding Plans as Super-Sets

The reformulation in Definition 2 might in some cases be unnecessarily permissive, allowing plans that are supersets as sets but not as multisets. The stricter reformulation will, therefore, given a set of plans, forbid all plans that are supersets (as operator sets) of some plan in the given set. We call it Forbidding Multiple Plans as Operator Super-Sets (FOSS).

Definition 3 *Let $\Pi = \langle \mathcal{V}, \mathcal{O}, s_0, s_\star, cost \rangle$ be a planning task, $\mathcal{X} = \{\mathcal{X}_1, \dots \mathcal{X}_k\}$ be a set of operator sets, and $U = \bigcup_{i=1}^{k} \mathcal{X}_i \subset \mathcal{O}$ be the union of these sets. The task $\Pi^{-}_{\mathcal{X}+} = \langle \mathcal{V}', \mathcal{O}', s'_0, s'_\star, cost' \rangle$ is defined as follows.*

- *$\mathcal{V}' = \mathcal{V} \cup \{\overline{v}\} \cup \{\overline{v}_o \mid o \in U\}$, where the variable $\overline{v}$ has the domain $dom(\overline{v}) = \{0 \dots k\}$ and all additional variables $\overline{v}_o$ being binary variables,*
- *$\mathcal{O}' = \{\overline{o} \mid o \in \mathcal{O}\} \cup \{o_i \mid o \in \mathcal{X}_i, 1 \leq i \leq k\}$, where*

$$\overline{o} = \langle pre(o) \cup \{\langle \overline{v}, 0 \rangle\}, eff(o) \cup \{\langle \overline{v}_o, 1 \rangle \mid o \in U\} \rangle,$$
$$o_i = \langle s_\star \cup \{\langle \overline{v}_o, 0 \rangle, \langle \overline{v}, i-1 \rangle\}, \{\langle \overline{v}, i \rangle\} \rangle, \text{ and}$$

  $cost'(\overline{o}) = cost(o)$, $cost'(o_i) = 0$,
- *$s'_0 = s_0 \cup \{\langle \overline{v}, 0 \rangle\} \cup \{\langle \overline{v}_o, 0 \rangle \mid o \in U\}$, and*
- *$s'_\star = s_\star \cup \{\langle \overline{v}, k \rangle\}$.*

$\mathcal{X}_i$, $1 \leq i \leq k$ are the operator sets associated with the plans seen so far for the original planning task Π. In words, the reformulation keeps track of the original operators from the planning task Π applied on the way to the goal (i.e., $\overline{o}$) with the extra effect $\langle \overline{v}_o, 1 \rangle$ for the operators in the set U, and then switches to the second phase, in which it ensures that there exists at least one operator from the original plans, that was not applied and now can be applied. Note, k is the size of the set $\mathcal{X}$, and the goal can only be reached if $\langle \overline{v}, k \rangle$ is true in addition to the original goal, $s_\star$. Thus, the goal can be reached by applying the additional operators o_i, which are applicable only when the original goal was achieved and the corresponding original operator was not applied.

Consider the same example planning task Π over operators $\mathcal{O} = \{a, b, c, d\}$, the same two plans aba and ca and their corresponding sets $\mathcal{X} = \{\{a, b\}, \{a, c\}\}$. The reformulated task $\Pi^{-}_{\mathcal{X}+}$ would then have extra ternary variable $\overline{v}$ and binary variables $\overline{v}_a$, $\overline{v}_b$ and $\overline{v}_c$. $\mathcal{O}'$ includes multiple copies of the original operators that appear in U. Here, we have $\mathcal{O}' = \{\overline{a}, \overline{b}, \overline{c}, \overline{d}\} \cup \{a_1, a_2, b_1, c_2\}$. Consider again the plan $\pi = abd$ for Π and a sequence of operators $\overline{\pi} = \overline{abd}$. Applying $\overline{\pi}$ to the initial state of $\Pi^{-}_{\mathcal{X}+}$ results in a state $s' = s_\star \cup \{\langle \overline{v}, 0 \rangle, \langle \overline{v}_a, 1 \rangle, \langle \overline{v}_b, 1 \rangle, \langle \overline{v}_c, 0 \rangle, \langle \overline{v}_d, 1 \rangle\}$. Note that none of $\{a_1, a_2, b_1, c_2\}$ operators are applicable in s' and therefore there is no corresponding to π plan of $\Pi^{-}_{\mathcal{X}+}$.

Observe that the reformulation, while forbidding plans that are supersets of elements in $\mathcal{X}$, allows plans that are proper subsets of these sets in $\mathcal{X}$. For example, a plan abc is a superset (as operator set) of aba. Assuming the reformulation is used within Algorithm 1, if aba is found first, abc is forbidden, but not vice versa. Thus, Algorithm 1 can be

Domain	top-uQ	top-MSQ	top-SQ
childsnack14	0	**6**	**6**
data-network18	0	**4**	**4**
elevators08	0	**2**	**2**
gripper	5	**16**	**16**
miconic	15	**17**	16
movie	**1**	0	0
openstacks08	**2**	1	1
parcprinter08	**25**	24	24
parcprinter11	**18**	17	17
pipesworld-notankage	**8**	7	7
psr-small	45	45	**47**
satellite	2	**4**	**4**
scanalyzer08	4	**6**	**6**
scanalyzer11	1	**3**	**3**
sokoban08	0	0	**1**
storage	**11**	9	10
Sum	337	361	**364**

Table 1: Per-domain coverage for the three configurations.

slightly adapted to check whether the newly found plans are subsets of previously found ones, discarding the plans that are supersets. Note that this could not happen in the case of Definition 2, since a plan that corresponds to a proper subset (as multiset) of another one will be found earlier.

It is worth noting here that in some cases FOSS might be preferred over FOSMS or vice versa. One example of such a case is whether it is important to differentiate alternative but similar ways of achieving subgoals.

Experimental Evaluation

To empirically evaluate the feasibility of our suggested approach, we have implemented our diverse planners on top of the ForbidIterative planners collection (Katz, Sohrabi, and Udrea 2019), on top of the Fast Downward planning system (Helmert 2006). ForbidIterative planner implements a shortest cost-optimal planner as a modified orbit A^* search (Domshlak, Katz, and Shleyfman 2015), considering both the cost and the distance of the search nodes. The code is available at https://github.com/ibm/forbiditerative. The experiments were performed on Intel(R) Xeon(R) CPU E7-8837 @2.67GHz machines, with the time and memory limit of 30min and 3.5GB, respectively. The benchmark set consists of all STRIPS benchmarks from optimal tracks of IPC 1998-2018, a total of 1797 tasks in 64 domains.

We have compared the two proposed reformulations depicted in Definition 2 (denoted by top-MSQ) and Definition 3 (denoted by top-SQ) to the unordered top-quality planning (top-uQ) (Katz, Sohrabi, and Udrea 2020). In our evaluation we focused on cost-optimal plans, e.g., q=1. First, Table 1 depicts the per-domain coverage of the three approaches, with the largest coverage bolded. A planner gets coverage of 1 for a task if it was able to find the solution for its corresponding computational problem. For top-uQ it means finding all cost-optimal plans up to reorderdings. For top-MSQ (top-SQ) it means finding all cost-optimal plans up to sub-multi-sets (subsets). While on most of the 64 domains there is no change in the coverage, there are 16 domains depicted

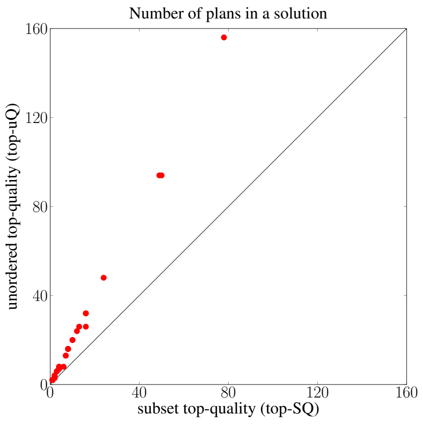

Figure 1: Solution size, comparing top-SQ to top-uQ.

in the table where the coverage differs. Note that the overall coverage significantly increases compared to the baseline, for both approaches. There are four domains where the previous approach could not solve any task, but our proposed approach can. Out of these, *elevators* and *sokoban* could not be solved by the previous approach in principle, as the unordered top-quality solution has an infinite number of plans. The superset top-quality one is however finite in these domains. Still, in a few domains (movie, openstacks, parcprinter, pipesworld, storage) the baseline top-uQ still dominates. This is not surprising, as the corresponding reformulation is simpler than the suggested ones, and therefore the classical planners are expected to be able to solve these reformulated tasks quicker. If the number of iterations does not (significantly) decrease when attempting to forbid more plans, the baseline should perform better than our approaches. These new suggested approaches thrive when the solution size decreases significantly.

To verify that the solution sizes indeed decrease, we compare the number of plans in the solution for the three approaches. Out of 327 tasks solved by all three approaches, on 277 the plan sets are of the same size. The remaining 50 tasks are depicted in Figure 1, comparing top-SQ to top-uQ. We do not show a comparison to top-MSQ, as only on 5 of these tasks the solution size of top-SQ is strictly smaller than of top-MSQ, namely one *ged* (49 vs. 50) and four *psr* tasks (2, 16, 2, 4 vs. 3, 26, 3, 7). Each point in the plot corresponds to a planning task that was solved by both approaches, and depicting the number of plans in each solution. The 50 tasks depicted in the figure all belong to the following four domains: *ged*, *pegsol*, *psr*, and *spider*. For these tasks, the solution size is decreased by half on average.

Conclusions and Future Work

We have presented new computational problems under the umbrella of top-quality planning, as well as corresponding planners. Our empirical evaluation shows these planners to be more practical than the previously existing tools.

For future work, we would like to investigate ways of improving the performance of these planners, possibly by reusing information from one iteration for the next one.

References

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

Boddy, M.; Gohde, J.; Haigh, T.; and Harp, S. 2005. Course of Action Generation for Cyber Security Using Classical Planning. In Biundo, S.; Myers, K.; and Rajan, K., eds., *Proceedings of the Fifteenth International Conference on Automated Planning and Scheduling (ICAPS 2005)*, 12–21. AAAI Press.

Chakraborti, T.; Isahagian, V.; Khalaf, R.; Khazaeni, Y.; Muthusamy, V.; Rizk, Y.; and Unuvar, M. 2020. From Robotic Process Automation to Intelligent Process Automation. In *Business Process Management: Blockchain and Robotic Process Automation Forum: BPM 2020 Blockchain and RPA Forum*, volume 393, 215–228. Springer Nature.

Domshlak, C.; Katz, M.; and Shleyfman, A. 2015. Symmetry Breaking in Deterministic Planning as Forward Search: Orbit Space Search Algorithm. Technical Report IS/IE-2015-03, Technion.

Helmert, M. 2006. The Fast Downward Planning System. *Journal of Artificial Intelligence Research*, 26: 191–246.

Katz, M.; Ram, P.; Sohrabi, S.; and Udrea, O. 2020. Exploring Context-Free Languages via Planning: The Case for Automating Machine Learning. In Beck, J. C.; Karpas, E.; and Sohrabi, S., eds., *Proceedings of the Thirtieth International Conference on Automated Planning and Scheduling (ICAPS 2020)*, 403–411. AAAI Press.

Katz, M.; and Sohrabi, S. 2020. Reshaping Diverse Planning. In Conitzer, V.; and Sha, F., eds., *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2020)*, 9892–9899. AAAI Press.

Katz, M.; Sohrabi, S.; and Udrea, O. 2019. ForbidIterative planners for top-k, top-quality, and diverse planning problems. https://doi.org/10.5281/zenodo.3246773.

Katz, M.; Sohrabi, S.; and Udrea, O. 2020. Top-Quality Planning: Finding Practically Useful Sets of Best Plans. In Conitzer, V.; and Sha, F., eds., *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2020)*, 9900–9907. AAAI Press.

Katz, M.; Sohrabi, S.; and Udrea, O. 2022. Bounding Quality in Diverse Planning. In Honavar, V.; and Spaan, M., eds., *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI 2022)*. AAAI Press.

Katz, M.; Sohrabi, S.; Udrea, O.; and Winterer, D. 2018. A Novel Iterative Approach to Top-k Planning. In de Weerdt, M.; Koenig, S.; Röger, G.; and Spaan, M., eds., *Proceedings of the Twenty-Eighth International Conference on Automated Planning and Scheduling (ICAPS 2018)*. AAAI Press.

Nguyen, T. A.; Do, M. B.; Gerevini, A.; Serina, I.; Srivastava, B.; and Kambhampati, S. 2012. Generating diverse plans to handle unknown and partially known user preferences. *Artificial Intelligence*, 190: 1–31.

Shleyfman, A.; Katz, M.; Helmert, M.; Sievers, S.; and Wehrle, M. 2015. Heuristics and Symmetries in Classical Planning. In Bonet, B.; and Koenig, S., eds., *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI 2015)*, 3371–3377. AAAI Press.

Sohrabi, S.; Riabov, A. V.; and Udrea, O. 2016. Plan Recognition as Planning Revisited. In Kambhampati, S., ed., *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*, 3258–3264. AAAI Press.

Vadlamudi, S. G.; and Kambhampati, S. 2016. A Combinatorial Search Perspective on Diverse Solution Generation. In Schuurmans, D.; and Wellman, M., eds., *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI 2016)*, 776–783. AAAI Press.