Loopless Top-K Planning

Julian von Tschammer[1] **Robert Mattmüller**[1] **David Speck**[1,2]

[1]University of Freiburg, Germany

[2]Linköping University, Sweden

Motivation

Airport domain [THN04]

Elevators domain [KS00]

Motivation

Airport domain [THN04]

Elevators domain [KS00]

Many similar plans visiting the same states multiple times!

Top-K Solution Concepts

Graph Theory

- k shortest path problem [BKH57]

⇝ Find the shortest k paths

Planning

- Top-k planning [RSU14]

⇝ Find the cheapest k plans

Top-K Solution Concepts

Graph Theory

- k shortest path problem [BKH57]

⇝ Find the shortest k paths

- k shortest simple path problem [Yen71]

⇝ Find the shortest k simple paths

Planning

- Top-k planning [RSU14]

⇝ Find the cheapest k plans

Top-K Solution Concepts

Graph Theory

- k shortest path problem [BKH57]

⇝ Find the shortest k paths

- k shortest simple path problem [Yen71]

⇝ Find the shortest k simple paths

Planning

- Top-k planning [RSU14]

⇝ Find the cheapest k plans

???

Top-K Solution Concepts

Graph Theory

- k shortest path problem [BKH57]
- ⇝ Find the shortest k paths
- k shortest simple path problem [Yen71]
- ⇝ Find the shortest k simple paths

Planning

- Top-k planning [RSU14]
- ⇝ Find the cheapest k plans
- Loopless top-k planning
- ⇝ Find the cheapest k loopless plans

Loopless Plans

Plans

- Sequence of applicable operators $\pi = \langle o_0, \ldots, o_{n-1} \rangle$
  - Generates a sequence of states $\text{states}(\pi) = \langle s_0, \ldots, s_n \rangle$
  - Initial state s_0 and goal state s_n

Loopless Plans

Plans

- Sequence of applicable operators $\pi = \langle o_0, \ldots, o_{n-1} \rangle$
  - Generates a sequence of states $\text{states}(\pi) = \langle s_0, \ldots, s_n \rangle$
  - Initial state s_0 and goal state s_n

$\rightsquigarrow$ A plan π is called loopless if all states of $\text{states}(\pi)$ are distinct.

- Airport domain: No planes driving in circles
- Elevator domain: No passengers entering/exiting the elevator multiple times

Problem Definition

Top-k Planning

- Given: a planning task Π and a natural number $k \in \mathbb{N}$
- Wanted: a subset of all plans $P \subseteq P_\Pi$ where
  - there exists no plan $\pi' \in P_\Pi$ with $\pi' \notin P$ that is cheaper than some plan $\pi \in P$, and
  - $|P| = k$ if $|P_\Pi| \geq k$, and $|P| = |P_\Pi|$ otherwise.

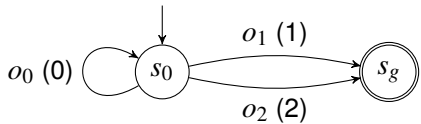
Problem Definition

Loopless Top-k Planning

- Given: a planning task Π and a natural number $k \in \mathbb{N}$
- Wanted: a subset of all loopless plans $P \subseteq P^{\ell\ell}_{\Pi}$ where
  - there exists no plan $\pi' \in P^{\ell\ell}_{\Pi}$ with $\pi' \notin P$ that is cheaper than some plan $\pi \in P$, and
  - $|P| = k$ if $|P^{\ell\ell}_{\Pi}| \geq k$, and $|P| = |P^{\ell\ell}_{\Pi}|$ otherwise.

Loopless Top-K Planning

Example



- Operators $\mathcal{O} = \{o_0, o_1, o_2\}$ with $\text{cost}(o_i) = i$
- Infinitely many optimal plans with a cost of 1
- Two loopless plans $P^{\ell\ell}_{\Pi} = \{\langle o_1 \rangle, \langle o_2 \rangle\}$

Generate-and-Test Approach

Idea

- Using a top-k planner with an anytime behavior
- Request a sufficiently large number of plans ($k = \infty$)
- Enumerate all plans with increasing costs
  - Add loopless plans to the solution set
  - Discard plans with loops

Generate-and-Test Approach

Pros & Cons

- ✔ Straightforward approach
- ✔ Any top-k planner with an anytime behavior can be used
- ✔ Sound approach

Generate-and-Test Approach

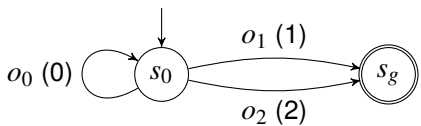
Pros & Cons

- ✔ Straightforward approach
- ✔ Any top-k planner with an anytime behavior can be used
- ✔ Sound approach
- ✗ Performance strongly depends on the number of discarded plans
- ✗ Not complete when there are zero cost loops

Generate-and-Test Approach

Pros & Cons

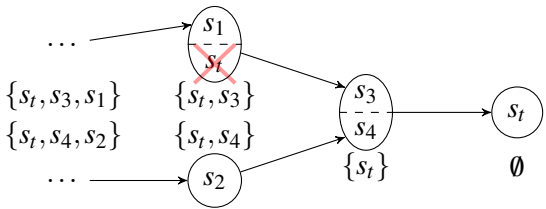
- ✔ Straightforward approach
- ✔ Any top-k planner with an anytime behavior can be used
- ✔ Sound approach
- ✗ Performance strongly depends on the number of discarded plans
- ✗ Not complete when there are zero cost loops



Symbolic Search Approach

Idea

- Use symbolic search (SYM-K algorithm)
- Modify the plan reconstruction phase
  - Keep track of visited states
  - Ignore already visited states



UNI FREIBURG

Symbolic Search Approach

Pros & Cons

- ✔ Sound and complete approach
- ✔ Generates only relevant plans

Symbolic Search Approach

Pros & Cons

- ✔ Sound and complete approach
- ✔ Generates only relevant plans
- ✗ More complex
- ✗ Unnecessary overhead when few or no loopy plans exist

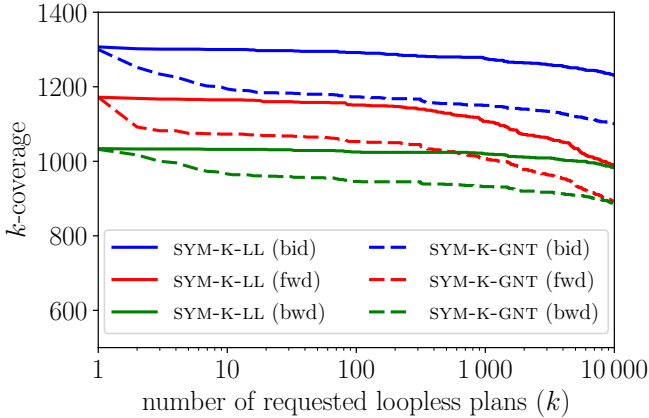
Empirical Evaluation

Setup

- Implemented both approaches in SYM-K
  - SYM-K-GNT and SYM-K-LL
  - Forward, backward and bidirectional search
- k-Coverage with values between $1 \leq k \leq 10000$
  - #Tasks solved when k loopless plans are requested
- 2262 Planning Tasks from 74 domains (optimal track of IPCs)

Empirical Evaluation

K-Coverage



Diverse Planning Zoo

- Objective: interesting and useful plans
- Useful set of plans [KSU20]
  - Equivalence relation over plans
  - One representative of each equivalence class

Diverse Planning Zoo

- Objective: interesting and useful plans
- Useful set of plans [KSU20]
  - Equivalence relation over plans
  - One representative of each equivalence class

Loopless Top-K Planning

- Generation of loopless plans with increasing costs
- Each loopless plan forms a representative
- Further diversification according to a diversity metric

Conclusion

- Introduced Loopless Top-K Planning
- All visited states of a plan must be distinct
- Two different approaches
  - Generate-and-test approach
  - Symbolic search approach

Conclusion

- Introduced Loopless Top-K Planning
- All visited states of a plan must be distinct
- Two different approaches
  - Generate-and-test approach
  - Symbolic search approach
- Practice: Possible to determine a set of k best loopless plans
- Symbolic search approach performs best overall

References I

[BKH57] Frederick Bock, Harold Kanter, and John Haynes, *An algorithm (the r-th best path algorithm) for finding and ranking paths through a network*, Armour Research Foundation, 1957.

[KS00] Jana Koehler and Kilian Schuster, *Elevator control as a planning problem*, Proc. AIPS 2000, 2000, pp. 331–338.

[KSU20] Michael Katz, Shirin Sohrabi, and Octavian Udrea, *Top-quality planning: Finding practically useful sets of best plans*, Proc. AAAI 2020, 2020, pp. 9900–9907.

[RSU14] Anton V. Riabov, Shirin Sohrabi, and Octavian Udrea, *New algorithms for the top-k planning problem*, ICAPS 2014 Scheduling and Planning Applications woRKshop, 2014, pp. 10–16.

References II

[THN04] Sebastian Trüg, Jörg Hoffmann, and Bernhard Nebel, *Applying automatic planning systems to airport ground-traffic control – A feasibility study*, Proc. KI 2004, 2004, pp. 183–197.

[Yen71] Jin Y. Yen, *Finding the k shortest loopless paths in a network*, Management Science **17** (1971), no. 11, 712–716.