Towards Robust Constraint Satisfaction in Hybrid Hierarchical Planning

Tobias Schwartz, Michael Sioutis, Diedrich Wolter

Challenges

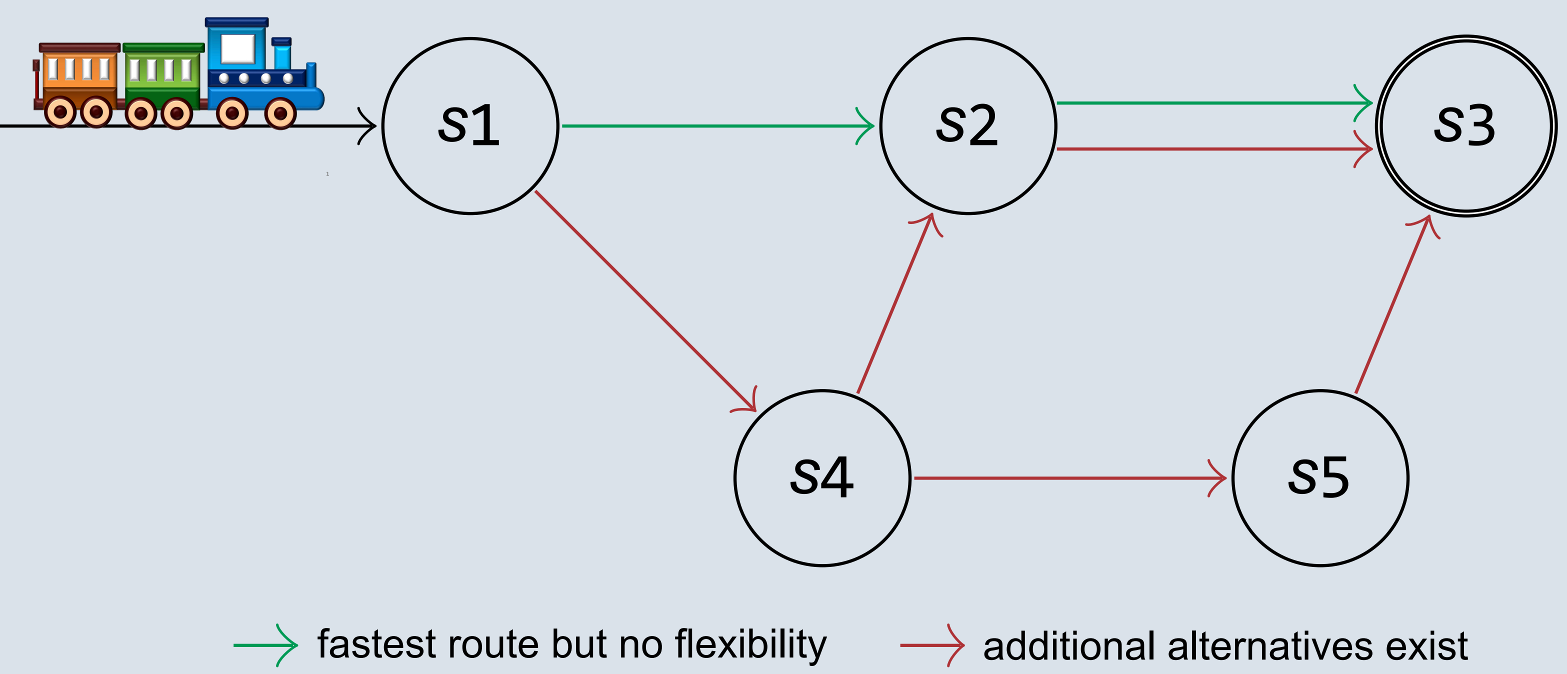
How to

- do **robust planning** in the face of a dynamic execution environment?
- best **define and model robustness** in the context of hierarchical planning?
- **measure robustness of a solution** within existing planning benchmarks?

Example & Motivation

The task of driving trains in a train network from one station to a particular destination is complex, because many requirements, constraints and uncertainties exist.

Many subtasks are part of the abstract task of driving a train from one station to another, e.g., `driveTo(?train, ?destination)`.



Robustness vs. Flexibility

Robustness

The ability of a system to resist all possible changes given by a model →Proactive

Flexibility

The ability of a system to quickly generate a new solution in case of change → Reactive

Robustness in Planning

It is important how changes are modelled and at what time they are known to the planner (before or during plan execution).

Robustness can be achieved by combining flexibility found in partial-order planning with plan execution monitoring [3].

A viable partial-order plan (POP), i.e., where at least one valid linearization exists, may be found by exploiting *state relevance*.

In hierarchical planning, performing tasks has been combined with the ability to react to events in the Refinement Acting Engine (RAE) [1]. Currently, only efficiency is considered in the online planner UPOM [4].

Case Study: Meta-CSPs

Meta-CSPs [2] are used to reason about many different forms of knowledge at once. Actions are modelled as methods $(f, (\mathcal{F}, \mathcal{C}))$, where fluent $f{=}(A, \cdot, \cdot, u)$ indicates that action A is being executed; $\mathcal{F}$ is the set of fluents, where $\mathcal{F}_p$ are preconditions, and $\mathcal{F}_+$, $\mathcal{F}_-$ positive and negative effects. $\mathcal{C}$ is a set of causal (CC), temporal (TC), and symbolic (BC) constraints [6]:

$$
\begin{aligned}
f &= (!driveTo(?t_1, ?s_2), [0, \infty], [0, 30], u(track1) = 1) \\
F_p &= \{f_1 = (At(?t_2, ?s_1), \cdot, \cdot)\} \\
F_- &= \{f_1\} \\
F_+ &= \{f_2 = (At(?t_3, ?s_3), \cdot, \cdot)\} \\
CC &= \{f_1 \text{ pre } f, f \text{ closes } f_1, f \text{ opens } f_2, f \text{ planned } f\} \\
BC &= \{S_1^{(f)} = S_1^{(f_1)}, S_1^{(f)} = S_1^{(f_2)}, S_2^{(f)} = S_2^{(f_2)}\} \\
TC &= \{I^{(f)} \ oi \ I^{(f_1)}, I^{(f)} \ fi \ I^{(f_2)}\}
\end{aligned}
$$

Robustness in QCNs by measuring *similarity* of all satisfiable scenarios. The solution with the highest similarity to all others is the most robust [5]. Can be a proactive measure, guiding the planning process actively. As such, it is especially useful in scenarios, where changes are known or can be predicted in advance.

References

[1] Ghallab, M.; Nau, D. S.; and Traverso, P. 2016. Automated Planning and Acting. Cambridge University Press.
[2] Mansouri, M.; and Pecora, F. 2016. A robot sets a table: a case for hybrid reasoning with different types of knowledge. J. Exp. Theor. Artif. Intell. 28: 801–821.
[3] Muise, C. 2014. Exploiting Relevance to Improve Robustness and Flexibility in Plan Generation and Execution. Ph.D. thesis, University of Toronto.
[4] Patra, S.; Mason, J.; Kumar, A.; Ghallab, M.; Traverso, P.; and Nau, D. S. 2020. Integrating Acting, Planning, and Learning in Hierarchical Operational Models. In ICAPS.
[5] Sioutis, M.; Long, Z.; and Janhunen, T. 2020. On Robustness in Qualitative Constraint Networks. In IJCAI.
[6] Stock, S. 2016. Hierarchische hybride Planung für mobile Roboter. Ph.D. thesis, University of Osnabrück.