Graph-based Orders for Saturated Cost Partitioning

Paul Höft, Linköping University

Supervisors: Jendrik Seipp, David Speck

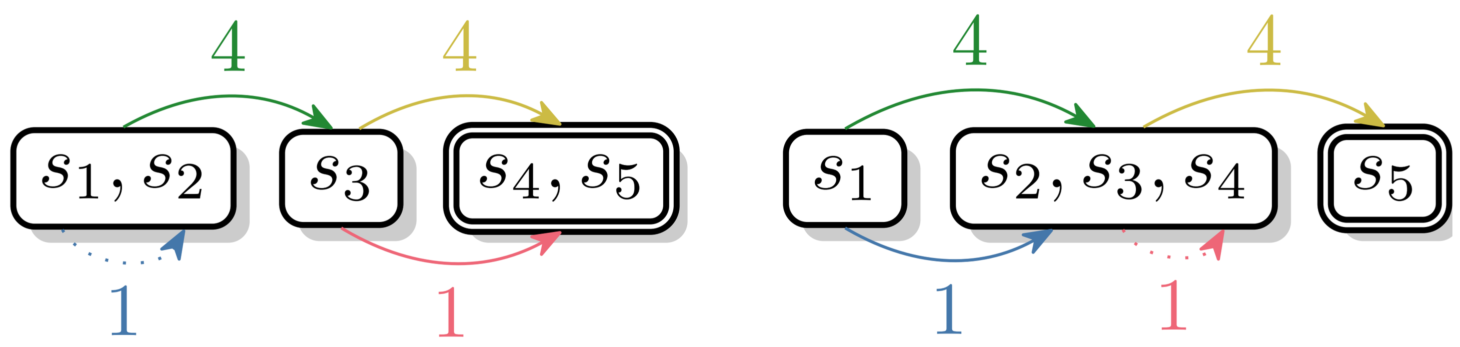
Acknowledgments: Thomas Keller

Optimal Classical Planning

- domain independent input language: PDDL
- A^*-search+admissible heuristic=optimal solution

Saturated Cost Partitioning (SCP)

- partition costs greedily following an order ω
- heuristic quality depends on order
- previous works use sampling and heuristics for finding good orders



Problem: Factorial growth of all SCP orders

Idea:

- Express SCP Heuristic as compact computational graph
- Use order-independence between heuristics to reduce graph size

SCP Computational Graphs

Rooted directed acyclic graph with max and sum nodes

More compact representation of SCP Heuristics

Graphs can be reduced by eliminating equivalent orders

Reducing Graph Size

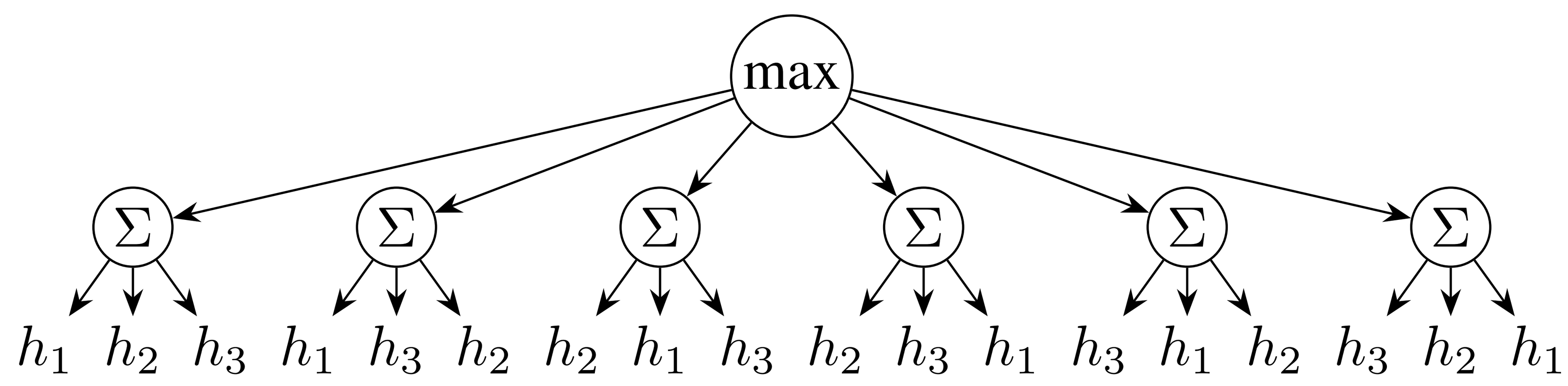
Approximations to avoid computation of orders

Algorithm:

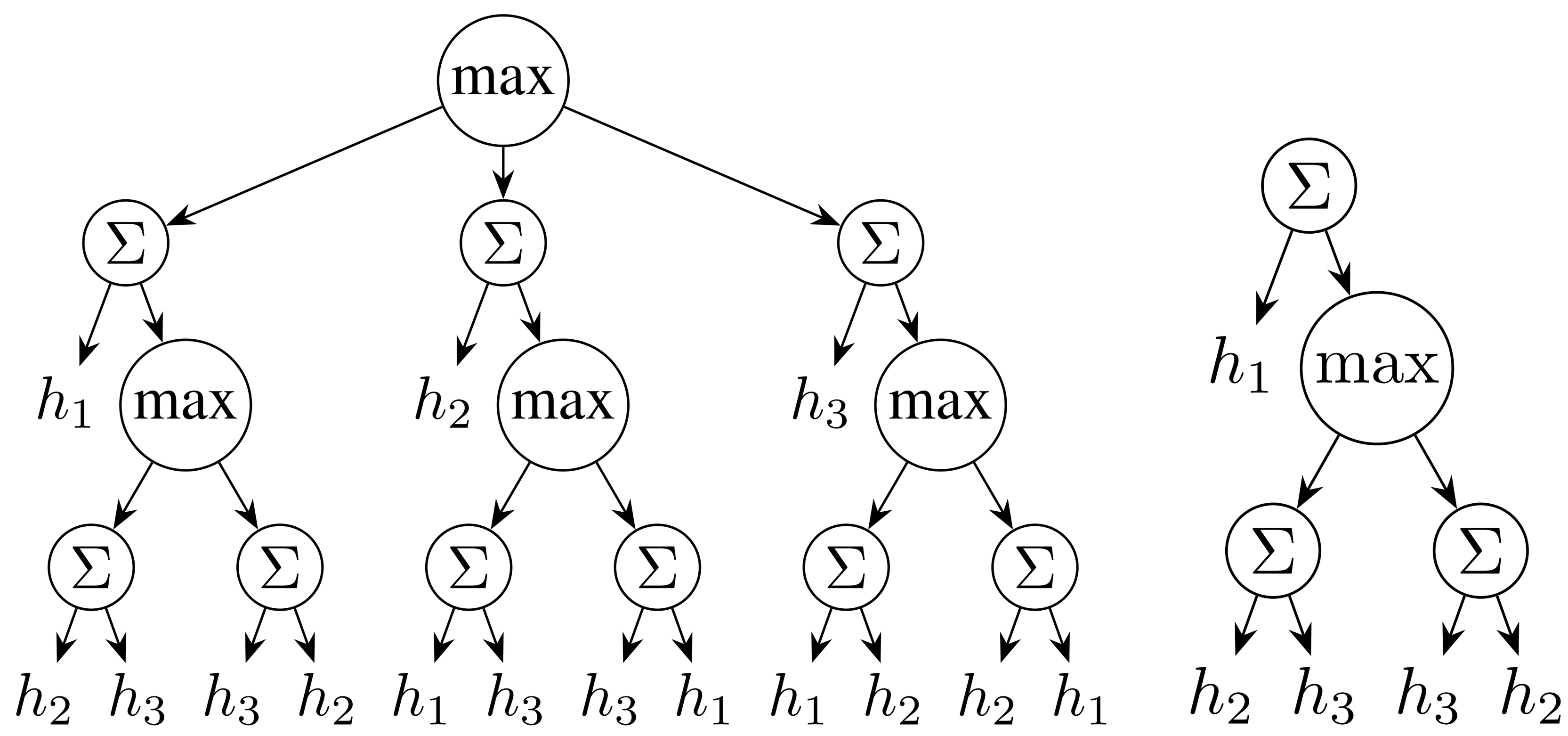
- graph G_C with heuristics as nodes
- add edge if two heuristics are in conflict
- find connected components in G_C
- build first layer of subgraph for each component
- repeat algorithm for heuristics in each component
- combine subgraphs with sum

Results

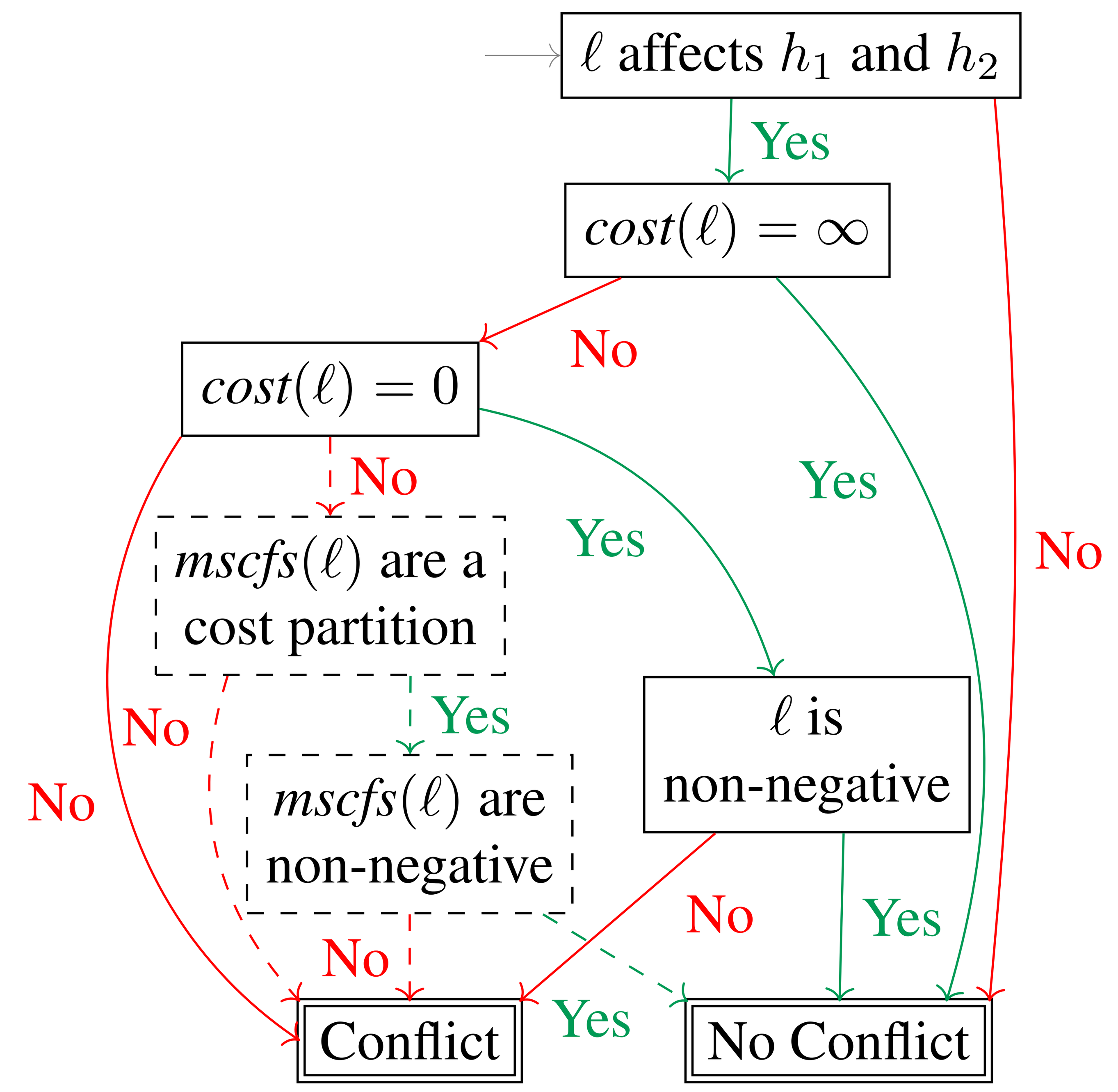
- optimized computation of SCP Heuristics
- first possible computation of optimal SCP Heuristic



Naive SCP computation tree



Prefix-merged computation tree and reduction for $\mathcal{H}_1 = \{h_1\}, \mathcal{H}_2 = \{h_2, h_3\}$



Conflict approximation between two heuristics

$$\omega_1 = \langle h_1, h_2 \rangle, c_{h_1} = \langle 0, 4, 1, 1 \rangle, c_{h_2} = \langle 0, 0, 3, 0 \rangle$$
$$\omega_2 = \langle h_2, h_1 \rangle, c_{h_1} = \langle 0, 3, 0, 0 \rangle, c_{h_2} = \langle 1, 1, 4, 0 \rangle$$

$$h^{SCP}_{\omega_1}(s_2) = 5 + 3 = 8 \quad h^{SCP}_{\omega_1}(s_4) = 0 + 3 = 3$$
$$h^{SCP}_{\omega_2}(s_2) = 3 + 4 = 7 \quad h^{SCP}_{\omega_2}(s_4) = 0 + 4 = 4$$

LINKÖPING UNIVERSITY

WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM