On Satisficing Planning with Admissible Heuristics

R. Bahumi C. Domshlak M. Katz

Faculty of Industrial Engineering and Management
Technion - Israel Institute of Technology

Heuristic Search

! Inadmissible Heuristic $\rightsquigarrow$ Satisficing Search

! Admissible Heuristic $\rightsquigarrow$ Cost-Optimal Search

? Admissible Heuristic $\rightsquigarrow$ Satisficing Search

Heuristic Search

! Inadmissible Heuristic $\rightsquigarrow$ Satisficing Search

! Admissible Heuristic $\rightsquigarrow$ Cost-Optimal Search

? Admissible Heuristic $\rightsquigarrow$ Satisficing Search

Heuristic Search

! Inadmissible Heuristic $\rightsquigarrow$ Satisficing Search

! Admissible Heuristic $\rightsquigarrow$ Cost-Optimal Search

? Admissible Heuristic $\rightsquigarrow$ Satisficing Search

Heuristic Search

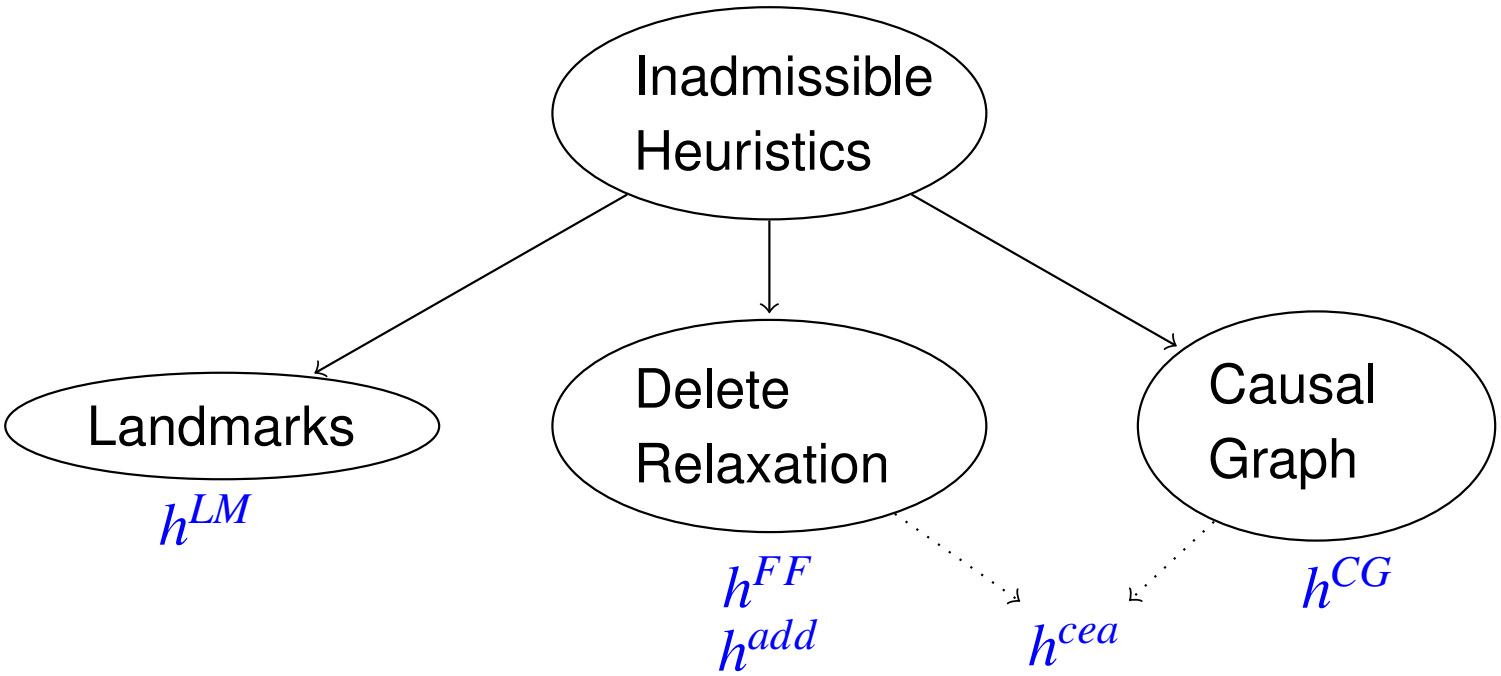
! Inadmissible Heuristic $\leadsto$ Satisficing Search

! Admissible Heuristic $\leadsto$ Cost-Optimal Search

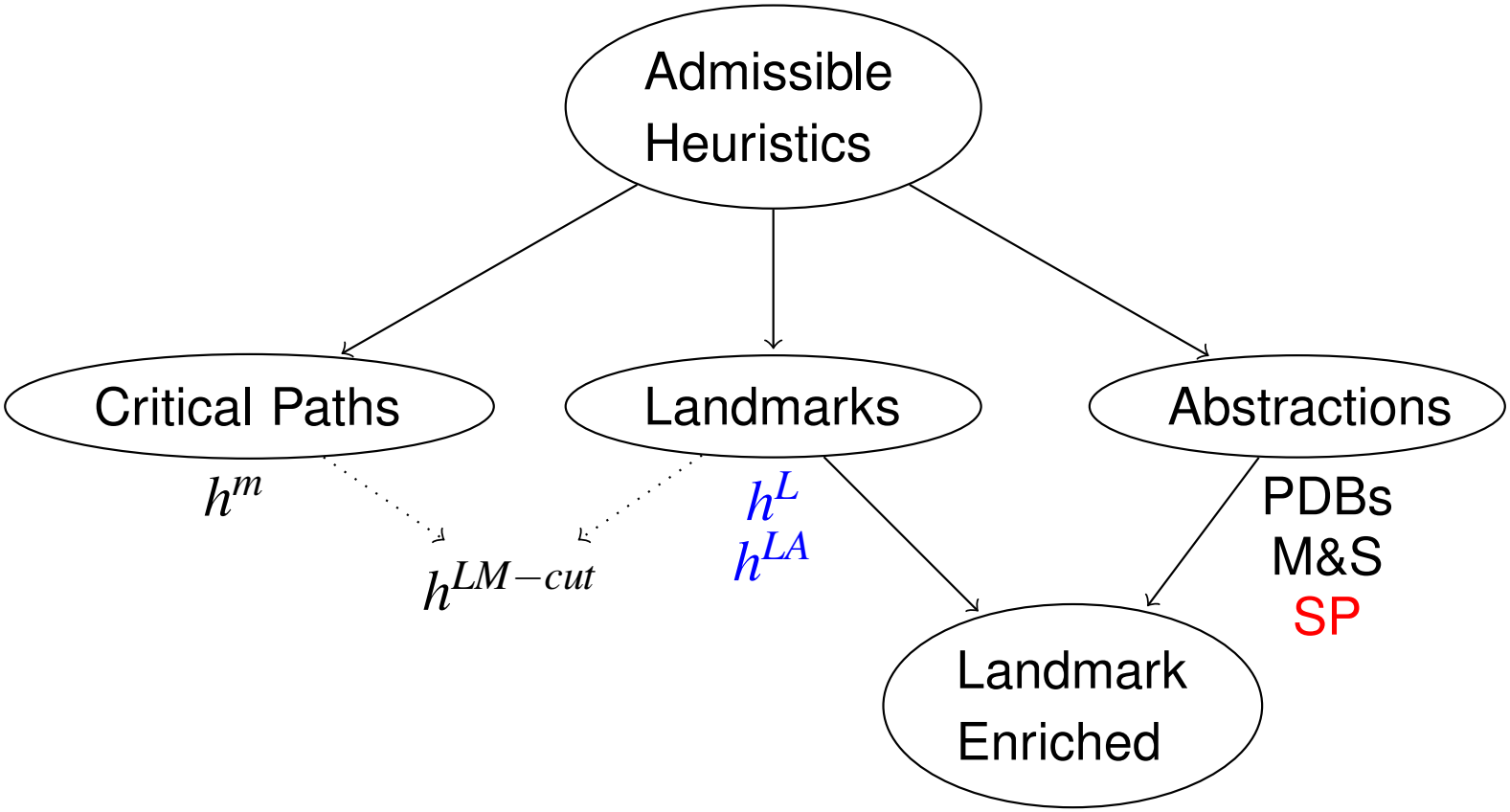
? Admissible Heuristic $\leadsto$ Satisficing Search

Search enchancements: Preferred Operators

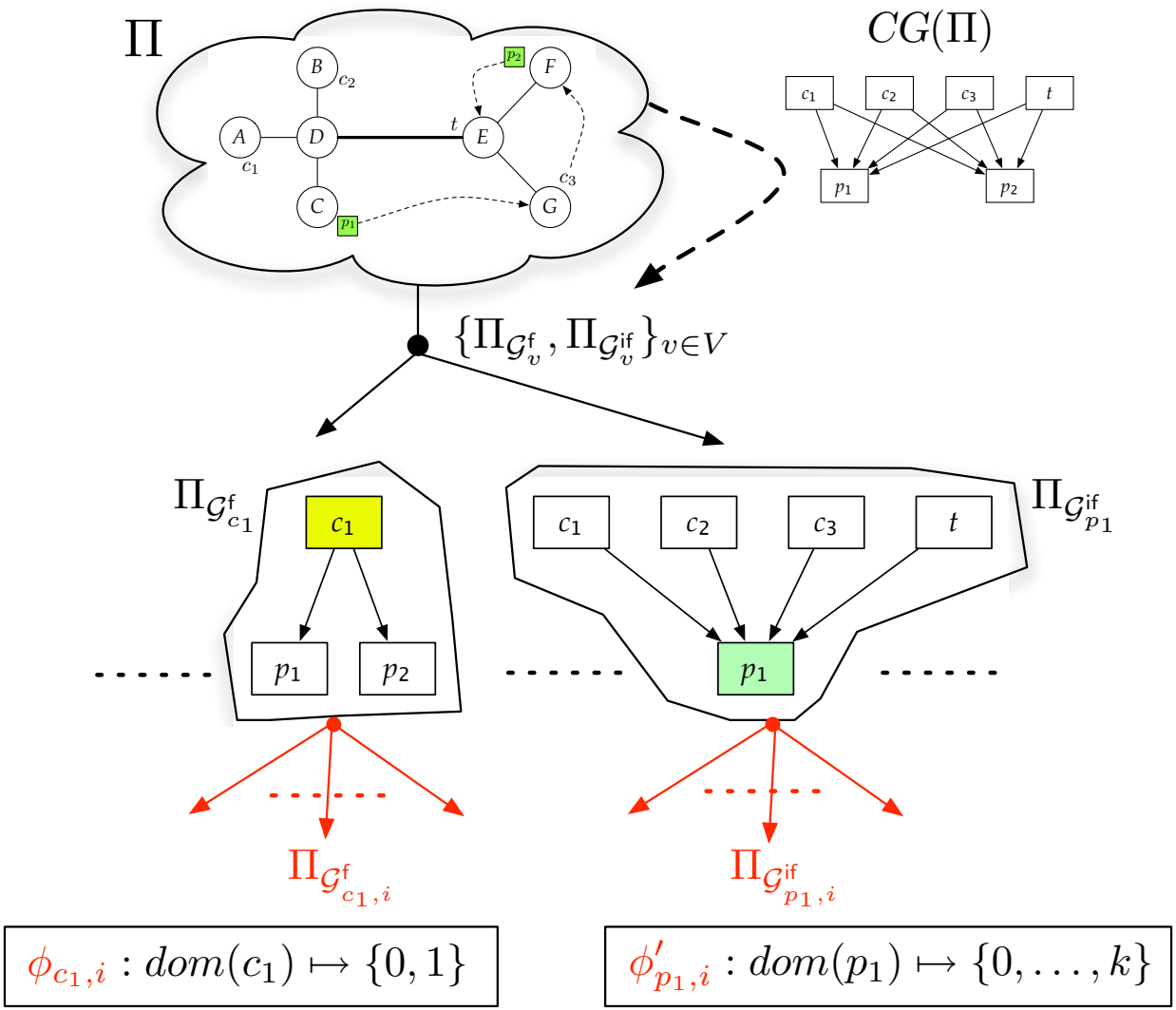
Inadmissible Heuristics



Admissible Heuristics



Fork Abstractions
(K & Domshlak, ICAPS08)



+ ensuring proper action cost partitioning

Dominating Actions

- A dominating action is an action starting some cost-optimal plan
- The notion of dominating actions complements the notion of *useless actions* (Wehrle, Kupferschmid, & Podelski, 2008)
- Deciding whether an action is useless (dominating) is in general as hard as planning itself
- Calculating the set of all dominating actions is poly-time for explicit abstractions (PDB, M&S)

Dominating Actions in Implicit Abstractions

Theorem

Given a state s, calculating the set of all dominating actions for a given (inverted) fork is poly-time

Proof

For each state s, an applicable action a is dominating iff

$$h^*(s) = h^*(s[\![a]\!]) + Cost(a)$$

- In practice can be done with little additional effort

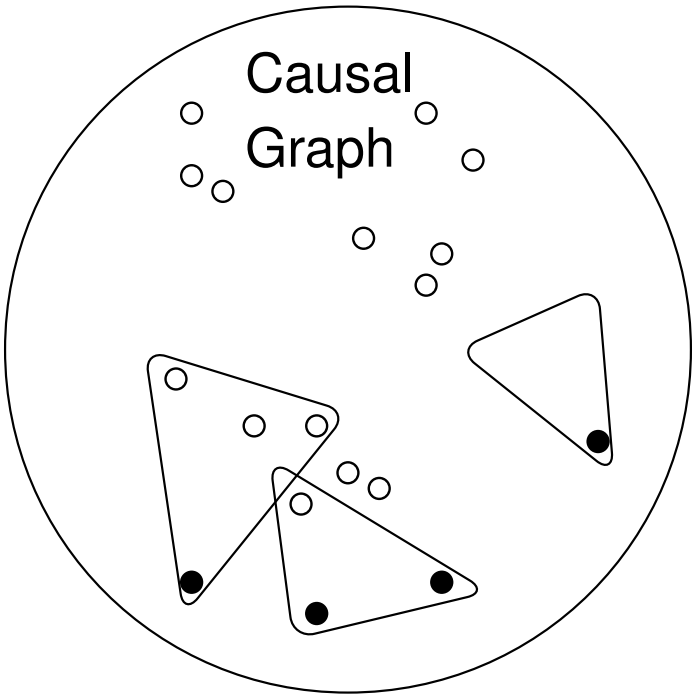
Dominating Actions in Implicit Abstractions

- Actions are unary effect
- (Inverted) Forks variables have either in-degree 0 (Up) or out-degree 0 (Lo)
- Variables with out-degree 0 are goal-variables
- Actions can be partitioned into those changing upper and lower variables
- Some of these actions may be more helpful in guidance towards the goal than other (should be checked)

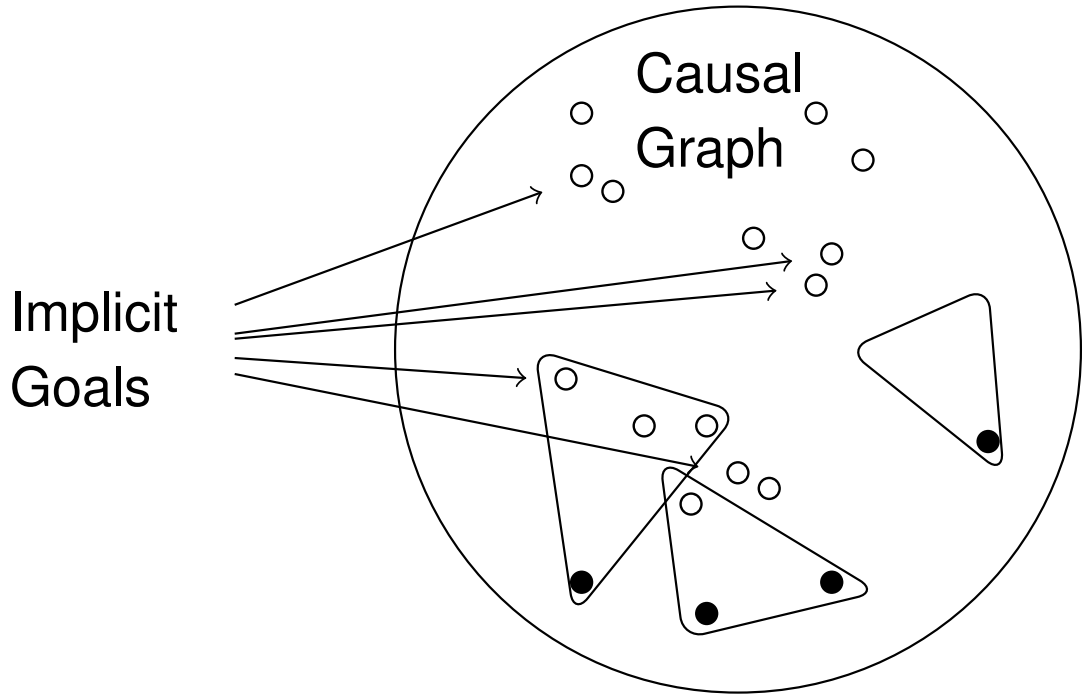
Empirical Evaluation

domain	h_{FF}		$h^{\mathcal{F}}$			$h^{\mathcal{J}}$			$h^{\mathcal{FJ}}$		
	No Pref	All Pref	No Pref	Up Pref	FF Pref	No Pref	All Pref	FF Pref	No Pref	Up Pref	FF Pref
blocks-00	**34.81**	34.24	32.56	30.69	32.84	31.88	31.32	32.67	31.27	29.91	32.06
elevators	27.26	**29.32**	11.20	16.94	12.74	8.33	8.58	13.57	24.29	20.62	26.43
logistics-98	22.55	**32.79**	20.52	18.53	28.28	20.15	26.96	31.35	20.24	18.70	28.49
openstacks	**29.52**	29.27	29.03	27.94	29.07	23.55	23.99	29.13	29.08	28.19	28.96
pegsol	**30.00**	29.85	29.95	29.00	29.00	29.95	29.00	29.95	29.95	28.75	29.90
woodworking	12.43	**27.72**	5.00	5.00	13.11	5.00	6.00	15.67	5.00	5.00	13.08
logistics-00	27.15	27.76	**27.96**	27.91	27.78	26.91	27.33	27.29	27.22	26.58	27.42
openstacks-adl	29.14	29.18	23.73	22.29	**29.22**	13.80	14.33	15.00	25.48	23.94	29.15
parcprinter	14.00	14.00	12.00	20.73	22.80	13.00	26.95	23.93	13.00	26.97	**28.88**
scanalyzer	24.38	25.15	22.53	21.81	21.65	22.36	**25.33**	22.28	21.43	21.70	22.47
sokoban	26.83	26.88	23.00	24.98	23.93	**28.83**	27.73	27.96	24.75	23.96	24.91
transport	12.16	18.29	19.44	17.44	**19.67**	8.30	8.34	8.94	13.22	12.61	17.82
	290.23	**324.45**	256.91	263.25	290.08	232.06	255.87	277.74	264.93	266.94	309.58

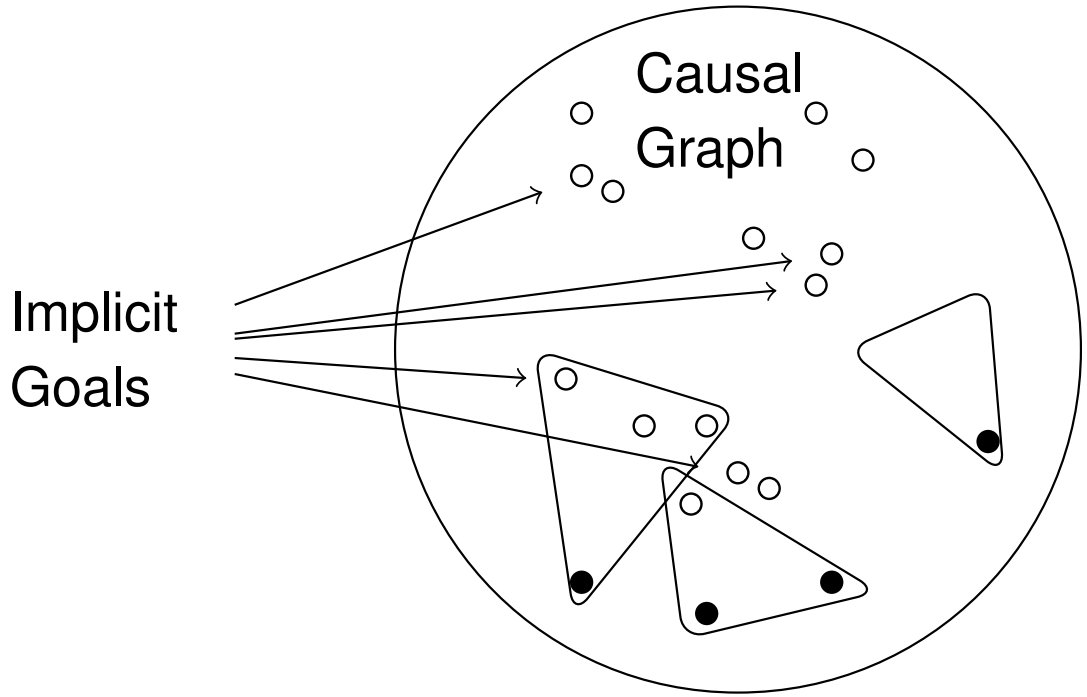
Goal Sensitivity



Possible Solution - Landmark Enriched Problem

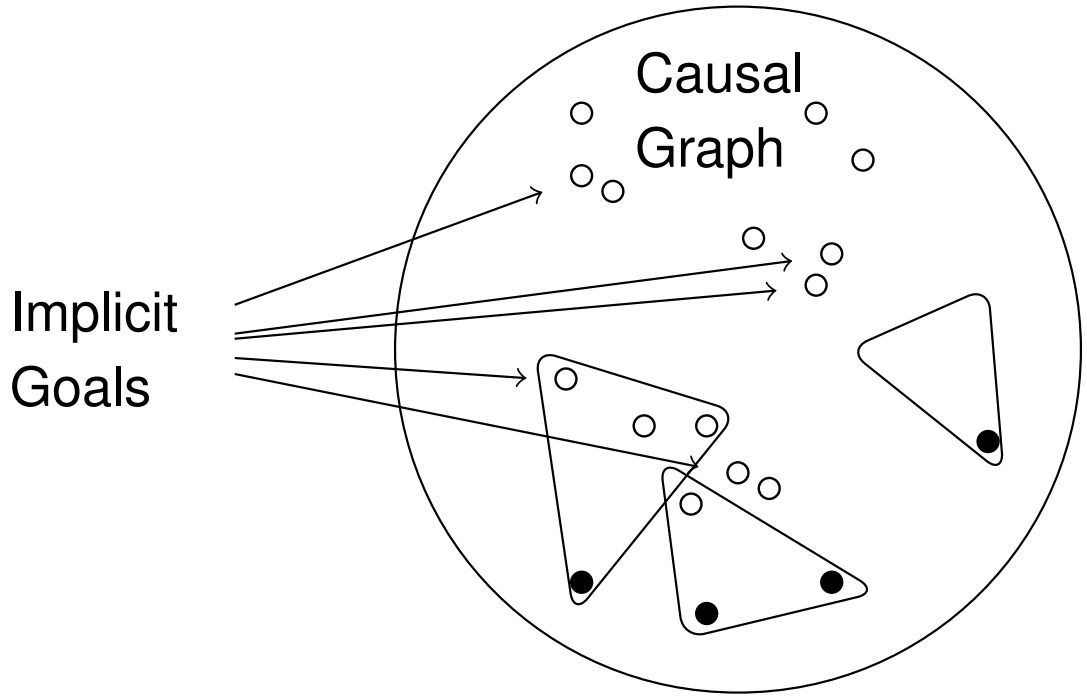


Possible Solution - Landmark Enriched Problem



- Landmarks found by backchaining or forward propagation are close to goals

Possible Solution - Landmark Enriched Problem



- ♠ Landmarks found by backchaining or forward propagation are close to goals
- ♠ Other (different) methods for finding landmarks are needed

Summary and Future Work

Conclusions

1. Preferred Operators for Structural Patterns
2. Combining evaluation and Preferred Operators from different heuristics may improve the overall performance

Future Work

1. Better coverage of the task's actions
2. Composition of dominating actions sets