Width-Based Backward Search:

Chao Lei and **Nir Lipovetzky**

School of Computing and Information Systems, The University of Melbourne, Australia

Introduction

- **We studied the performance of width-based planners over Duality and Regression,**
- **We proposed backward and bidirectional width-based search algorithms,**
- **Challenges of regression in classical planning were discussed.**

Main Findings

- **Width-based search solves dual problems efficiently when |G| = 1, but challenging when |G| >1.**
- **Backward BFWS(f5) and k-BFWS(f5) perform worse than their forward versions but can be orthogonal.**
- **Width-based bidirectional search algorithms show trade-off in terms of coverage and meeting-in-the-middle.**
- **FB: forward *k*-BFWS(f5) running first and then backward *k*-BFWS(f5), *k*=1, can be integrated as a quick preprocessing step.**
- **Modeling matters: partial states, missing invariants and underspecified action schemas cause unnecessary challenges on regression.**

THE UNIVERSITY OF MELBOURNE

Backward width-based search solves more problems over the **STRIPS regression model** than using **forward width-based search** over **dual problems. Regression / progression** models / algorithms are **orthogonal** when integrated over different **bidirectional algorithms** and **state-of-the-art planners**.

Take a picture to download the full paper

Duality

- $P^d = < F, O^d, I^d, G^d >$
- $a^d \in O^d$, $a^d = <del, add, pre>$
- $I^d \subseteq F \setminus G$
- $G^d \subseteq F \setminus I$

RESULTS 1

P: original problem; P^d dual problem; |G|=1 instances with single-goal; |G|>1 instances conjunctive goals.

	IW		BFWS(f5)	
	|G| = 1		|G| > 1	
	P	P^d	P	P^d
Instances	37921	88856	1095	1095
Solved	35918	87830	765	67
Percentage	95%	99%	70%	6%

Regression State Model

- $S^R = < S, s_0, S_G, A, f, c >$
- $A(s) = \{a \mid del(a) \cap s = \emptyset, add(a) \cap s \neq \emptyset\}$
- $s_0 = G$
- $S_G = \{s \mid s \subseteq I\}$
- $f(a, s) = s \setminus add(a) \cup pre$ (a)

RESULTS 2

Solved instances by backward (B) vs. forward (F) k-BFWS(f5) and BFWS(f5).

Domain	F-k	B-k	F	B
Solved (1095)	734	372	765	391
Average time	13.39	46.26	13.10	69.53
Average quality	154.93	176.78	167.13	189.55

Bidirectional Search

1) ***front-to-end*** estimates the heuristic value of a state *s* according to how close it is to the **closest goal state**.

2) ***front-to-front*** estimates the heuristic value of a state *s* according to how close it is to the **opposite search frontier**.

RESULTS 3

k-BDWS: the combination of **forward and backward** k-BFWS(f5) and checks intersection with respect to the **novelty 1 frontier** of the opposite direction.

R: checks with random states in the *Close* list; **H** checks with the last expanded state; **C**: checks with the full *Close* list. **M**: solved by meet-in-the-middle.

	Dual	Dual-FB	k-BDWS-e	k-BDWS-e R	k-BDWS-e H	k-BDWS-e C
Solved	841	864	713	629	714	596
F-Solved	-	841	410	418	536	298
B-solved	-	23	41	95	146	31
M-solved	-	-	262	116	32	267