Best-First Width Search for Lifted Classical Planning

Augusto B. Corrêa[1] Jendrik Seipp[2]

[1]University of Basel, Switzerland
[2]Linköping University, Sweden
augusto.blaascorrea@unibas.ch, jendrik.seipp@liu.se

This Talk

in this talk:

- efficient implementation of lifted best-first width search
- new ways to combine width search with other heuristics
- state-of-the-art lifted planner

Lifted Planning

we consider lifted classical planning:

- planning only with the PDDL description
  - predicate symbols, objects, action schemas, initial state, goal

heuristic search:

- actions are lifted
- states are ground

we do not know all possible ground atoms

Best-First Width Search

best-first width search (BFWS):

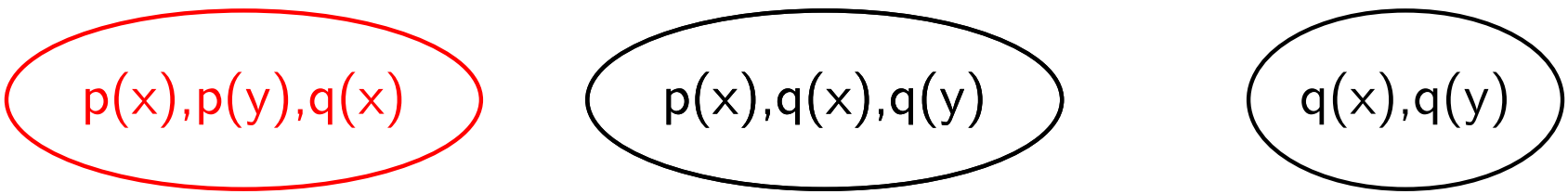
- based on novelty of a state
  - size of the smallest set of atoms not seen before
- smaller set $\Longrightarrow$ "more novel"
- prioritize "more novel states"
- in practice: check only sets up to size k

see: Lipovetzky and Geffner (2012)

In Practice

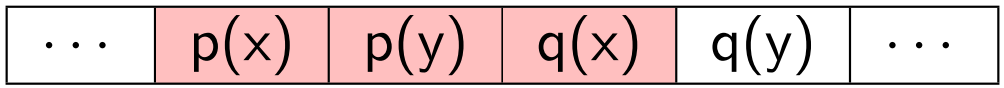
ground implementation: $(k = 1)$

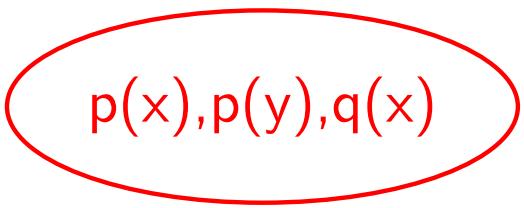
···	p(x)	p(y)	q(x)	q(y)	···



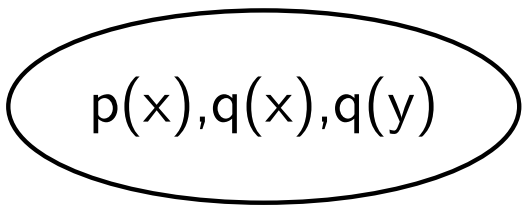
In Practice

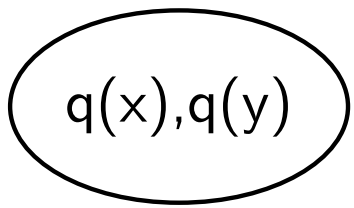
ground implementation: $(k = 1)$





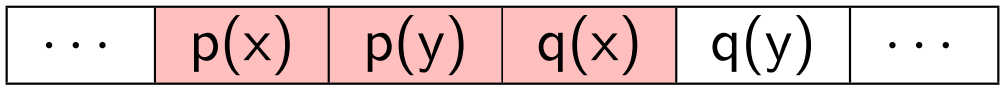
novel!

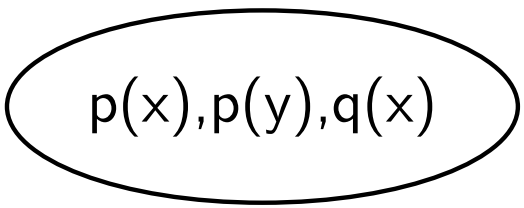




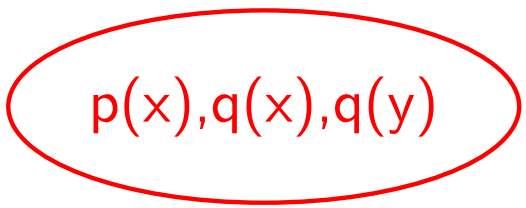
In Practice

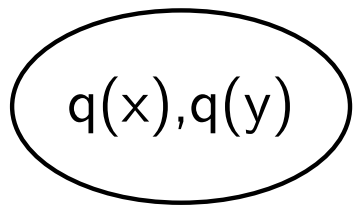
ground implementation: $(k = 1)$





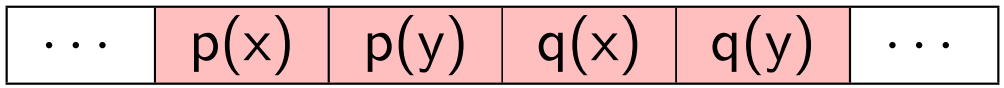
novel!

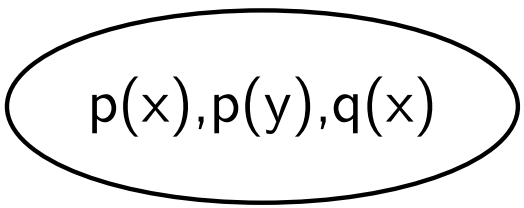




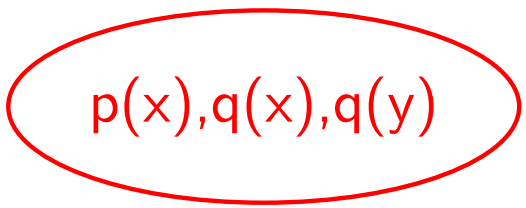
In Practice

ground implementation: $(k = 1)$

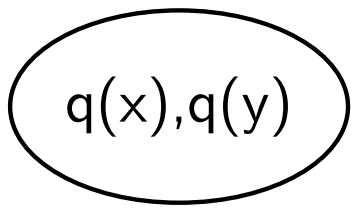




novel!

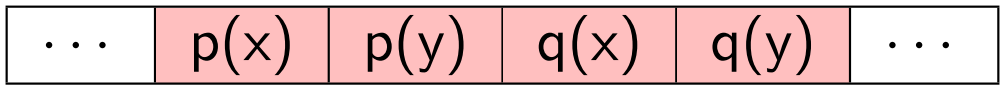


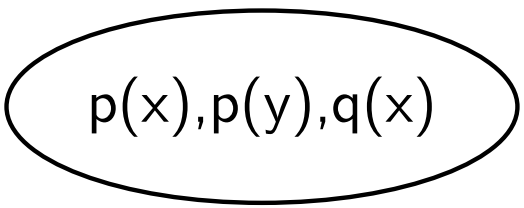
novel!



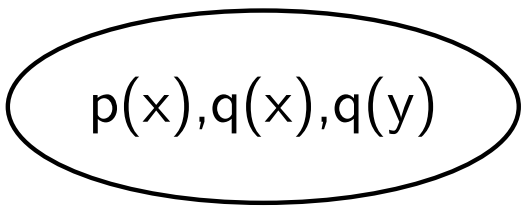
In Practice

ground implementation: $(k = 1)$

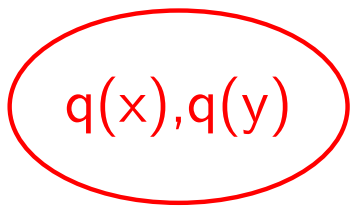




novel!



novel!



not novel!

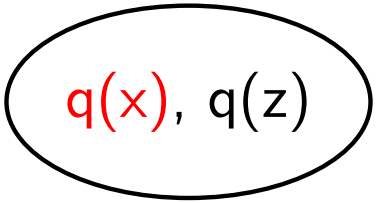
w-value will be higher than previous states!

Lifted Implementation

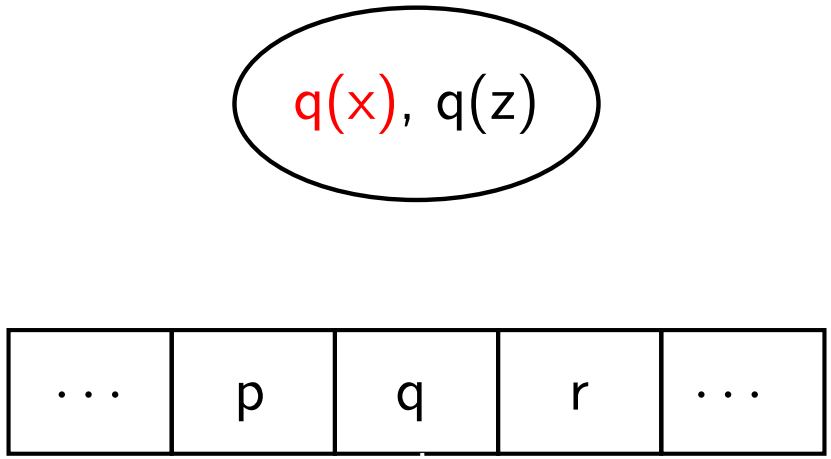
does not work directly on lifted planners:

- needs set of possible ground atoms
- tasks are too large to precompute it

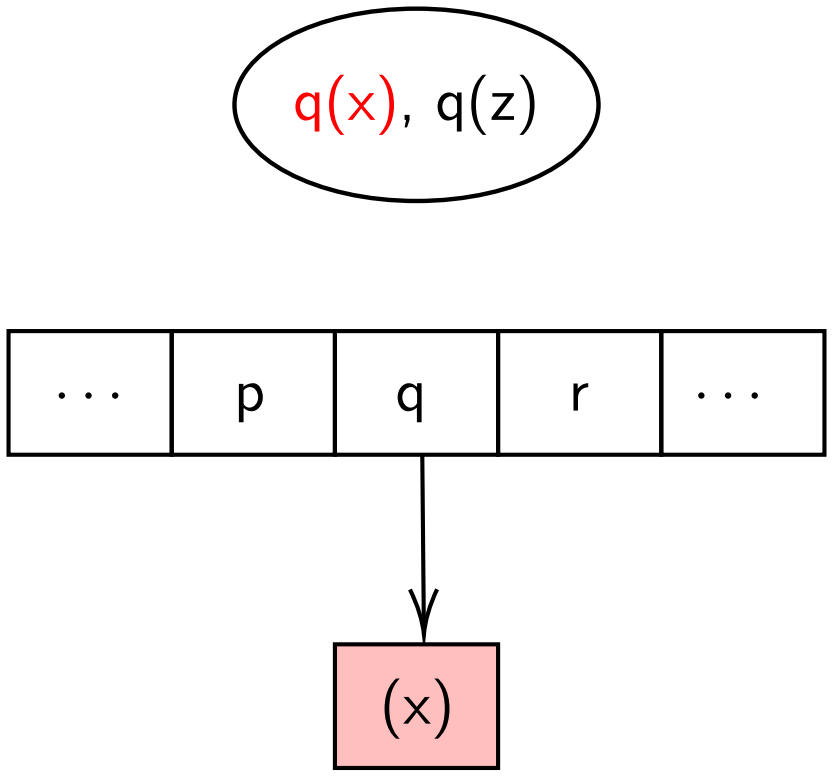
Lifted Implementation



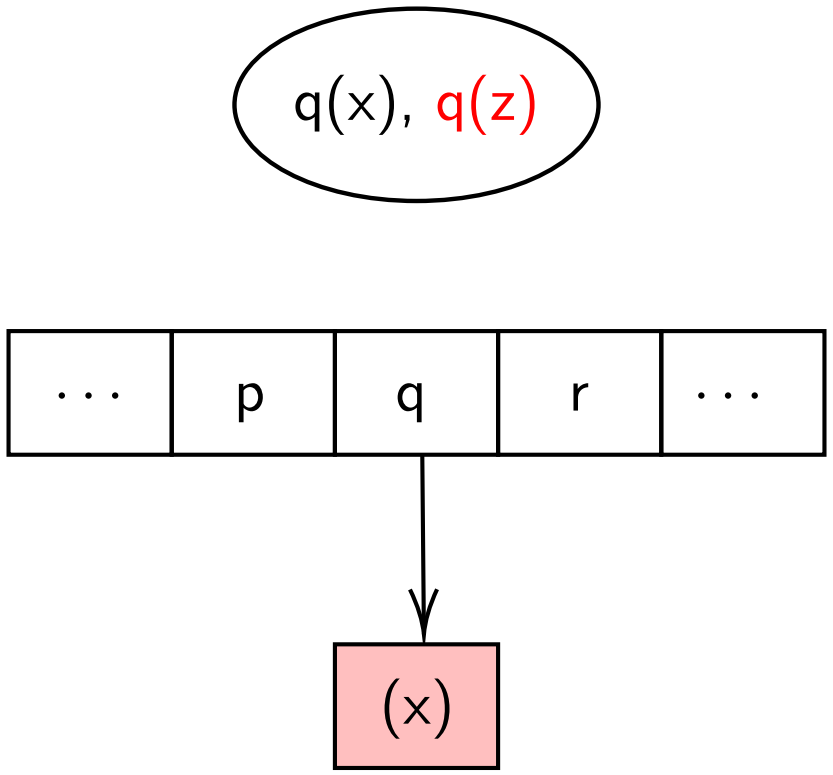
Lifted Implementation



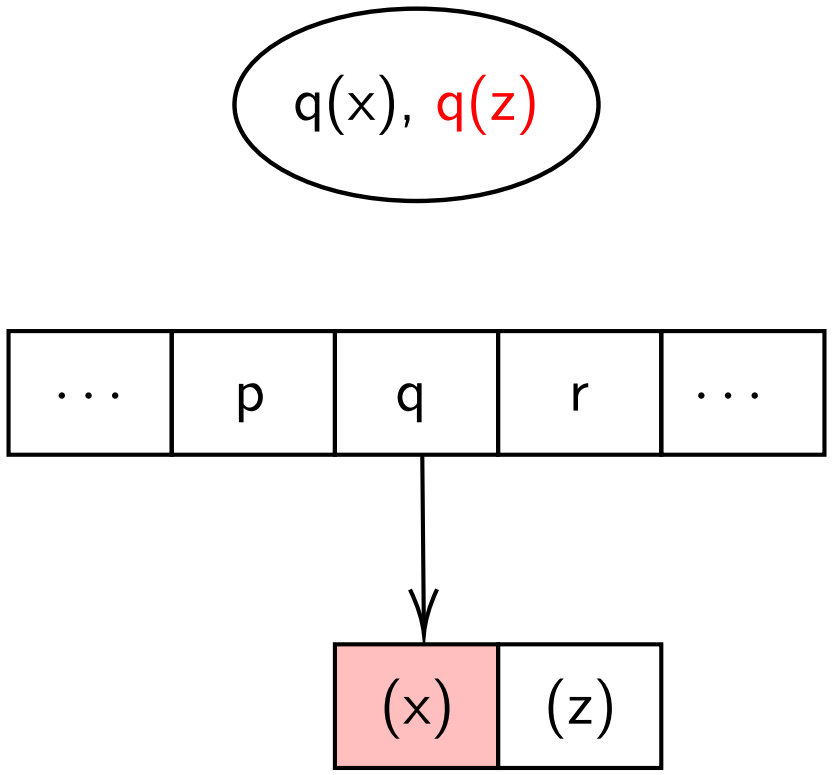
Lifted Implementation



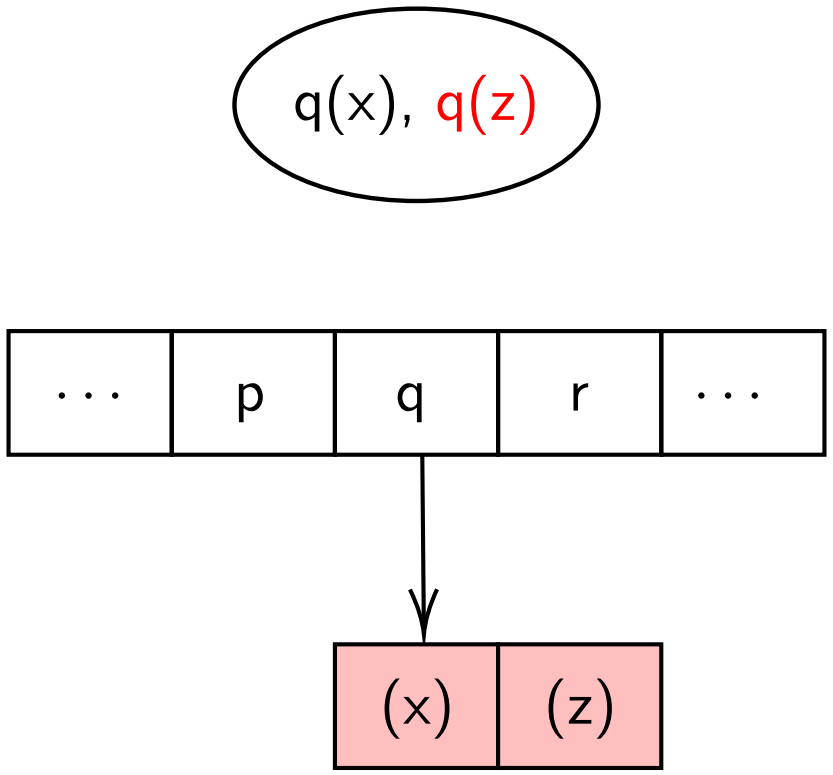
Lifted Implementation



Lifted Implementation



Lifted Implementation



Lifted Implementation

basic idea:

- one table of reached tuples per predicate symbol
- on-demand indexation

More Sophisticated Novelty Measures

partition functions:

- use functions $f_1, \ldots, f_n$ to partition search-space
- compute novelty based on states in the same partition

usually use $\#r$ and $\#g$ as partition functions

- $\#r$: number of relevant atoms that are true in s
- $\#g$: number of goal atoms that are true in s

how do we define relevant atoms?

Relevant Atoms

we use two approaches to define relevant atoms:

- R_0: $r = \emptyset$
- R_X: $r =$ useful atoms from a relaxed plan from initial state

notation:

- BFWS(R_0): use $\#R_0$ and $\#g$ as partition functions
- same for BFWS(R_X)

see Francès et al. (2017) for other definitions of relevant atoms

Experiments

how does a lifted implementation compare to a ground one?

- using $k = 2$, and two different sets of domains

Experiments

how does a lifted implementation compare to a ground one?

- using $k = 2$, and two different sets of domains

	FS-blind	Lifted BFWS(R_0)
IPC (1001)	714	**725**
blocksworld (40)	0	**6**
childsnack (144)	**73**	60
genome-edit-dist. (312)	**312**	307
logistics (40)	0	**10**
organic-synthesis (56)	0	**48**
pipesworld-tankage (50)	18	**43**
rovers (40)	**2**	0
visitall-multidim. (120)	37	**108**
visitall-5-dim (60)	–	48
HTG Total (862)	442	**630**

Experiments

	Baselines			Lifted BFWS	
	LAMA	Dual-BFWS	L-h^{FF}	R_0	R_X
IPC (1001)	917	**953**	821	725	741
blocksworld (40)	**12**	4	9	6	5
childsnack (144)	**116**	109	72	60	67
genome-edit-dist. (312)	**312**	**312**	311	307	**312**
logistics (40)	36	4	**40**	10	31
organic-synthesis (56)	21	20	48	48	**49**
pipesworld-tankage (50)	18	18	27	43	**47**
rovers (40)	16	13	**40**	0	1
visitall-multidim. (120)	60	36	98	108	**111**
visitall-5-dim (60)	12	6	42	**48**	**48**
HTG Total (862)	603	522	**687**	630	671

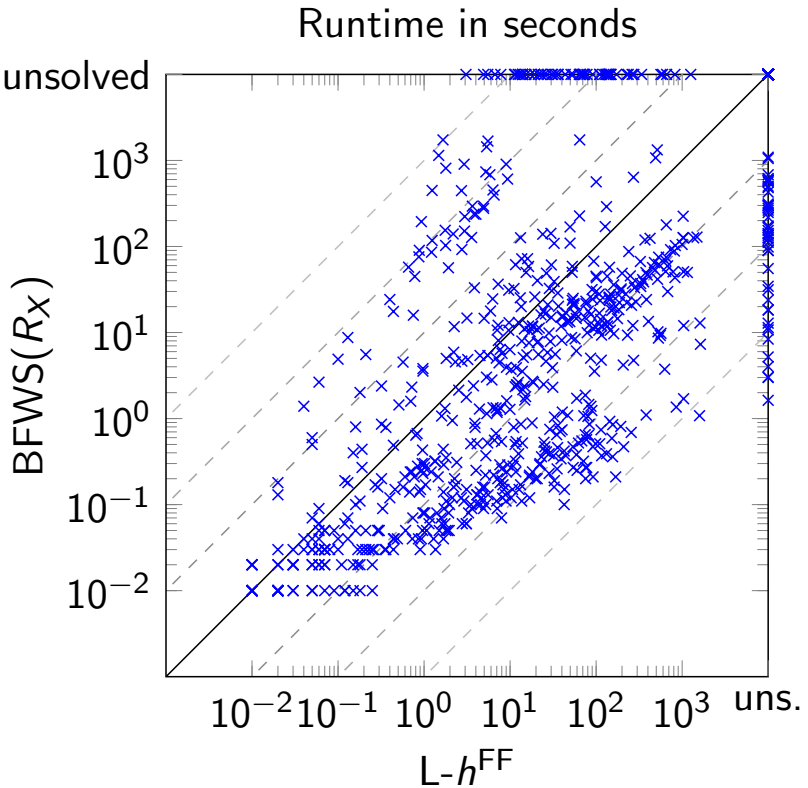
Experiments

	Baselines			Lifted BFWS	
	LAMA	Dual-BFWS	L-h^{FF}	R_0	R_X
IPC (1001)	917	**953**	821	725	741
blocksworld (40)	**12**	4	9	6	5
childsnack (144)	**116**	109	72	60	67
genome-edit-dist. (312)	**312**	**312**	311	307	**312**
logistics (40)	36	4	**40**	10	31
organic-synthesis (56)	21	20	48	48	**49**
pipesworld-tankage (50)	18	18	27	43	**47**
rovers (40)	16	13	**40**	0	1
visitall-multidim. (120)	60	36	98	108	**111**
visitall-5-dim (60)	12	6	42	**48**	**48**
HTG Total (862)	603	522	**687**	630	671

L-h^{FF} vs. BFWS(R_X)

L-h^{FF} and BFWS(R_X) perform well in different domains

- L-h^{FF} exploits domain-structure (e.g., useful atoms)
- BFWS(R_X) explores state-space very quickly



Combining L-h^{FF} and BFWS(R_X)

alternation between open-lists:

- evaluate nodes using multiple functions
- use one open-list for each function
- choose one open-list at a time for expansion
- balance exploration and exploitation

our alternation algorithms:

- $[R_X, h^{\mathrm{add}}]$ and $[R_X, h^{\mathrm{FF}}]$
- caveat: better performance with $k = 1$

see Röger and Helmert (2010)

Experiments

	Baselines			Lifted BFWS			
	LAMA	Dual-BFWS	L-h^{FF}	R_0	R_X	$[R_X, h^{\mathrm{add}}]$	$[R_X, h^{\mathrm{FF}}]$
IPC (1001)	917	**953**	821	725	741	838	857
blocksworld (40)	12	4	9	6	5	**21**	19
childsnack (144)	**116**	109	72	60	67	100	101
genome-edit-dist. (312)	**312**	**312**	311	307	**312**	309	309
logistics (40)	36	4	**40**	10	31	**40**	**40**
organic-synthesis (56)	21	20	48	48	49	**50**	**50**
pipesworld-tankage (50)	18	18	27	43	47	**48**	47
rovers (40)	16	13	**40**	0	1	**40**	**40**
visitall-multidim. (120)	60	36	98	108	**111**	101	101
visitall-5-dim (60)	12	6	42	48	**48**	42	41
HTG Total (862)	603	522	687	630	671	**751**	748

Experiments

	Baselines			Lifted BFWS			
	LAMA	Dual-BFWS	L-h^{FF}	R_0	R_X	$[R_X, h^{\mathrm{add}}]$	$[R_X, h^{\mathrm{FF}}]$
IPC (1001)	917	**953**	821	725	741	838	857
blocksworld (40)	12	4	9	6	5	**21**	19
childsnack (144)	**116**	109	72	60	67	100	101
genome-edit-dist. (312)	**312**	**312**	311	307	**312**	309	309
logistics (40)	36	4	**40**	10	31	**40**	**40**
organic-synthesis (56)	21	20	48	48	49	**50**	**50**
pipesworld-tankage (50)	18	18	27	43	47	**48**	47
rovers (40)	16	13	**40**	0	1	**40**	**40**
visitall-multidim. (120)	60	36	98	108	**111**	101	101
visitall-5-dim (60)	12	6	42	48	**48**	42	41
HTG Total (862)	603	522	687	630	671	**751**	748

Conclusions

- BFWS works well in the lifted setting
- nice fit with with delete-relaxation heuristics
- state-of-the-art lifted planner

The End

Thank You for Your Attention!