Alternation-Based Novelty Search

Augusto B. Corrêa[1,2] Jendrik Seipp[3]

[1]University of Oxford, United Kingdom
[2]University of Basel, Switzerland
[3]Linköping University, Sweden

Presented by Remo Christen
@ICAPS'25

Agile Planning

context: agile planning

- solve tasks as fast as you can
- faster $\implies$ more points
- IPC track since 2014
- we consider propositional planning here

last two IPC agile tracks:

2018: only BFWS planners beat LAMA

2023: LAMA baseline outperformed all participants

Agile Planning

context: agile planning

solve tasks as fast as you can

faster $\implies$ more points

IPC track since 2014

we consider propositional planning here

last two IPC agile tracks:

2018: only BFWS planners beat LAMA

2023: LAMA baseline outperformed all participants

this talk: how to combine LAMA and BFWS?

LAMA

LAMA [Richter and Westphal, JAIR 2010]

- still state-of-the-art
- for agile track: "LAMA first"

LAMA

LAMA [Richter and Westphal, JAIR 2010]

- still state-of-the-art
- for agile track: "LAMA first"

LAMA alternates open lists [Röger and Helmert ICAPS 2010]
four open lists, ordered by:

(1) h^{FF}

(2) h^{FF} but only states from preferred operators

(3) h^{LM}

(4) h^{LM} but only states from preferred operators

idea: expand node from (1), then from (2), and so on

BFWS

Best-First Width Search [Lipovetzky and Geffner AAAI 2017]

- different versions and implementations of BFWS
- crucial concept: novelty

intuition: prefer states that have novel information

- novelty $=$ size of smallest tuple that was not seen before
- check only up to a width $k \in \{1, 2\}$

BFWS(f_6)

in practice: calculate novelty only in partition of state space

BFWS(f_6) [Lipovetzky and Geffner AAAI 2017]:
order open-list by $f_6 = \langle w_{\langle h^{\mathsf{LM}}, h^{\mathsf{FF}}\rangle}, \mathit{pref}, h^{\mathsf{LM}}, w_{\langle h^{\mathsf{FF}}\rangle}, h^{\mathsf{FF}}\rangle$, where:

- $w_{\langle h^{\mathsf{LM}}, h^{\mathsf{FF}}\rangle}$: novelty of state with same $(h^{\mathsf{LM}}, h^{\mathsf{FF}})$ value
- pref: 1 if generated by pref. operator; 0 otherwise
- $w_{\langle h^{\mathsf{FF}}\rangle}$: novelty of state with same (h^{FF}) value

Our Method

next: how to combine LAMA and BFWS?

experimental settings:

- agile setting from IPC 2018/2023: 5 min, 8 GiB
- metrics: coverage and agile score
- training set: IPCs 1998–2014 (2 502 tasks)
- test set: IPCs 2018 and 2023 (360 tasks)

why split the data?

- reduce overfitting
- calibrate our method on domains BFWS had access to

But First...

remember: BFWS has this parameter k

in practice, BFWS planners select k dynamically

- if task is too large, $k = 2$ is expensive
- pick based on $V = \#$variables

But First...

remember: BFWS has this parameter k

in practice, BFWS planners select k dynamically

if task is too large, $k = 2$ is expensive

pick based on V = #variables

	Fixed		Dynamic		
BFWS(f_6)	k=1	k=2	V=10	V=100	V=1000
Coverage	2228	2159	2227	**2233**	2182
Agile Score	1786.1	1707.3	1785.6	**1786.4**	1711.5

Combining BFWS and LAMA

idea: insert f_6 open-list in LAMA
spoiler: this doesn't work — too expensive

Combining BFWS and LAMA

idea: insert f_6 open-list in LAMA
spoiler: this doesn't work — too expensive

next idea: simpler BFWS open-lists

- remove individual components of f_6
- LAMA-W(f) = LAMA + open-list ordered by f

Combining BFWS and LAMA

idea: insert f_6 open-list in LAMA
spoiler: this doesn't work — too expensive

next idea: simpler BFWS open-lists

remove individual components of f_6

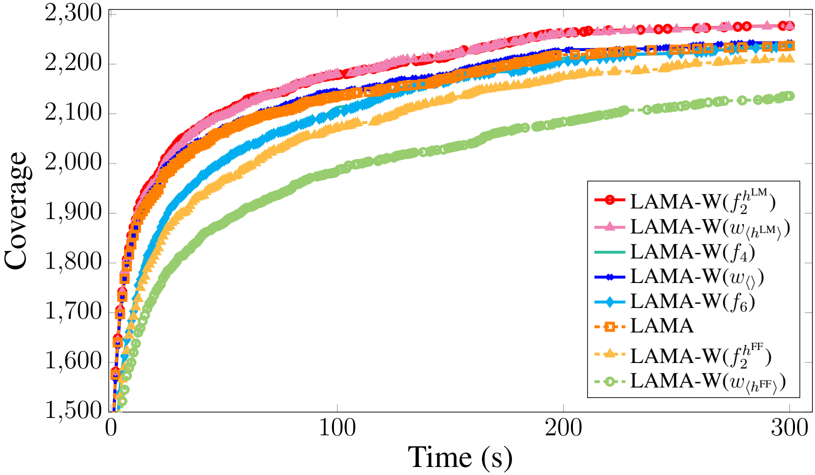
LAMA-W(f) $=$ LAMA $+$ open-list ordered by f

note: some configs correspond to other known version (f_4, f_2)

relevant: $f_2^h = \langle w_{\langle h \rangle}, h \rangle$

much simpler than f_6

Coverage over Time on Training Set



per domain:

LAMA-W($f_2^{h^{\mathrm{LM}}}$) beats LAMA in 14 domains, and only loses in 2

NOLAN

best algorithm LAMA-W($f_2^{h^{\mathsf{LM}}}$) has the following open-lists:

(1) h^{FF}

(2) h^{FF} but only states from preferred operators

(3) h^{LM}

(4) h^{LM} but only states from preferred operators

NOLAN

best algorithm LAMA-W($f_2^{h^{\mathsf{LM}}}$) has the following open-lists:

(1) h^{FF}

(2) h^{FF} but only states from preferred operators

(3) h^{LM}

(4) h^{LM} but only states from preferred operators

(5) $f_2^{h^{\mathsf{LM}}}$

NOLAN $=$ LAMA-W($f_2^{h^{\mathsf{LM}}}$):
we call it NOLAN for NOvelty, Landmarks, AlternatioN

Test Set

next: compare NOLAN with other baselines in the test set

- LAMA, BFWS + IPC 2023 winners
- include other novelty-based planners

Test Set

next: compare NOLAN with other baselines in the test set

LAMA, BFWS + IPC 2023 winners

include other novelty-based planners

	Coverage	Agile Score
NOLAN	**204**	**89.30**
LAMA	199	87.25
BFWS(f_6)	174	75.57
FDSS	171	76.22
DecStar	167	76.05
GBFS-RSL	142	57.87
BFWS-Pref.	133	63.59
Dual-BFWS	131	57.48
OLCFF	124	52.44

Table: Coverage and agile scores on the last two IPC domains.

Outlook

conclusion:

- combine LAMA and BFWS
- obvious combination does not work; have to simplify f_6
- NOLAN improves on LAMA

post-talk extra context:

- NOLAN is *partially* the idea behind Scorpion Maidu
- winner of IPC 2023 Satisficing Track [Corrêa et al. IPC 2023]
- Scorpion Maidu has 12 open-lists — too complicated

The End

Thank You for Your Attention!

Per-Domain

	BFWS(f_6)	LAMA	NOLAN
BFWS(f_6)	—	13	10
LAMA	13	—	2
NOLAN	17	14	—

Table: Per-domain coverage comparison for old IPC tasks.

Coverage over Time on Test Set