Universität Basel

Transforming Fact Landmarks to Action Landmarks for Heuristic Search

Bachelor Thesis Presentation

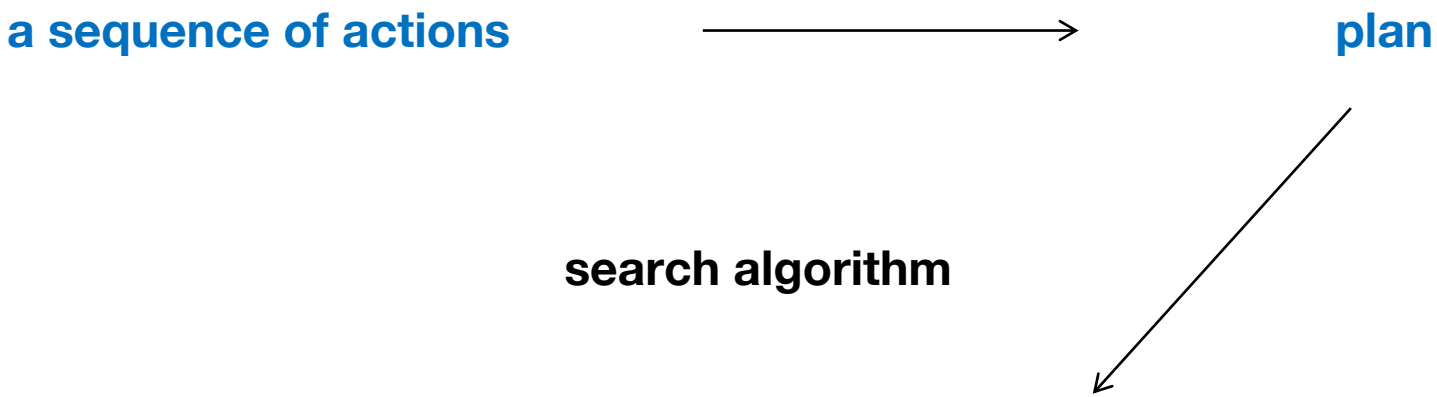
University of Basel - AI Research Group

Yaowen Rui

Supervisor: Dr.Florian Pommerening, Esther Mugdan

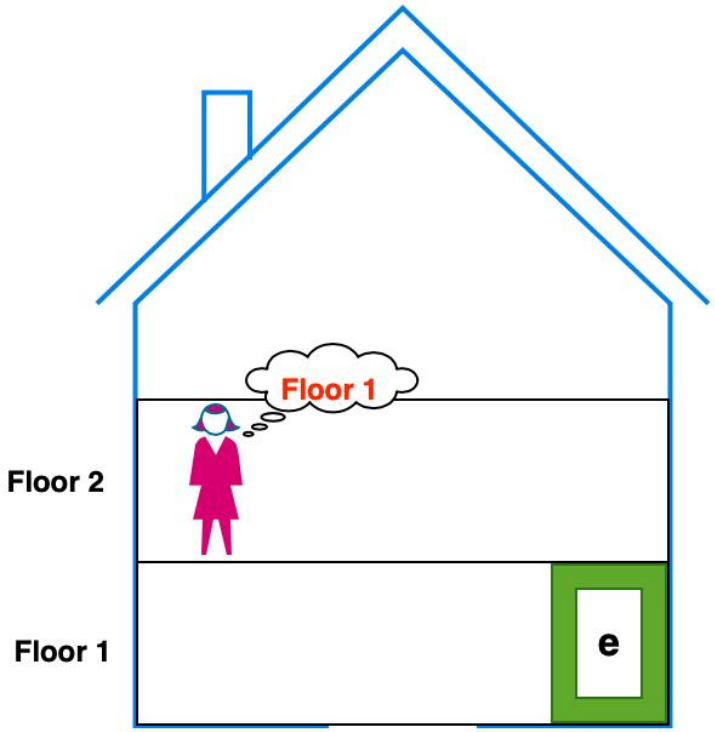
Introduction

What is classical planning ?



Background - Example

Elevator problem



State variable

p-at, e-at, p-in

dom(p-at) = {1, 2}, dom(e-at) = {1, 2}, dom(p-in) = {e, ¬e}

Initial state

Passenger on floor 2: <p-at, 2>

Elevator on floor 1: <e-at, 1>

Passenger not inside the elevator: <p-in, ¬e>

The goal

Passenger on floor 1: <p-at, 1>

Action

p-leave-e-1

pre: <p-in, e>, <e-at, 1>

eff: <p-at, 1>

Background - Landmark

Universität Basel

Definition - Disjunctive Fact Landmark

A disjunctive fact landmark l^F for state s is a set of atoms
such that all s-plans π visit a state containing some atom in l^F.

Definition - Disjunctive Action Landmark

A disjunctive action landmark l^A for state s is a set of actions
such that all s-plans π satisfy $l^A \cap \pi \neq \emptyset$.

Fact Landmark Generation

Preparation: Fact landmarks
Generation of fact landmarks: Backchaining + Fact landmark candidates test

{ <p-at, 1> }

⬆

first achievers: p-leave-e-1

⬆

preconditions: { <p-in, e> }, { <e-at, 1> }

⬆

first achievers: p-board-e-2

⬆

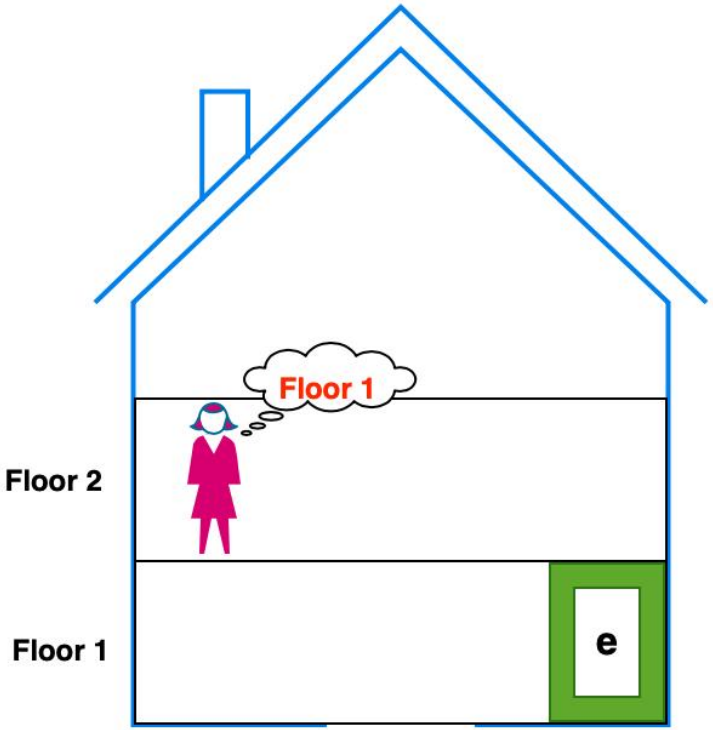
preconditions: {<e-at, 2>} , {<p-at, 2> }

⬆

first achievers: e-move-1-2

⬆

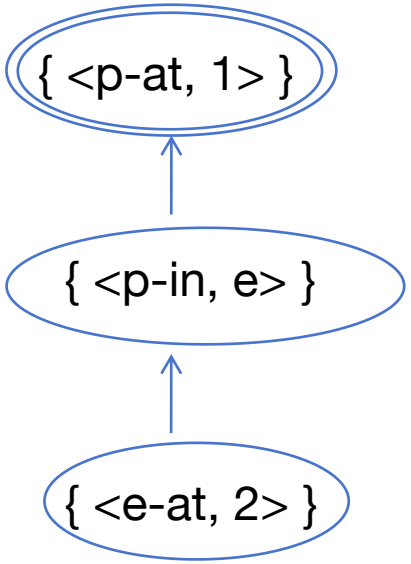
preconditions: { <e-at, 1> }



Action Landmark Generation

Fact Landmark Graph (V, E)

V: a set of fact landmarks, E: a set of edges labeled with natural orderings.



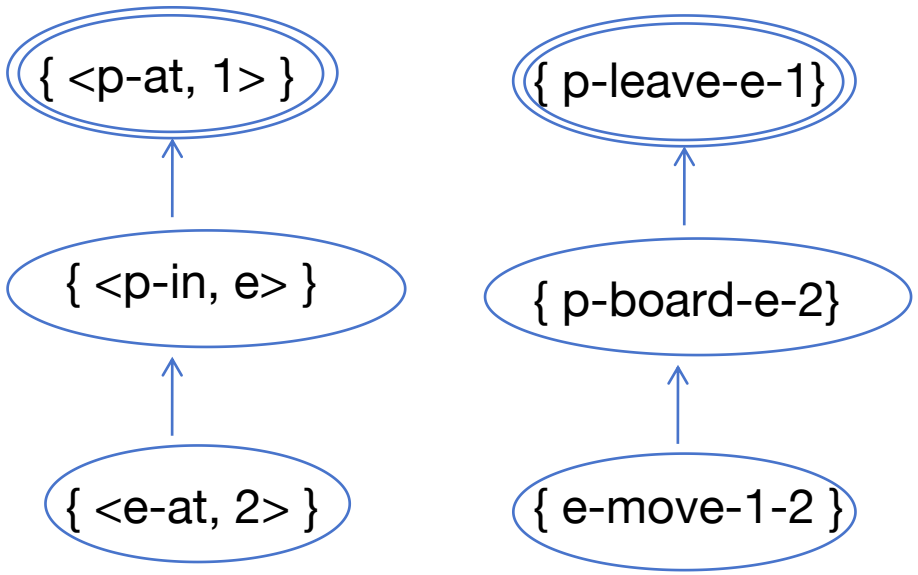
derive action landmarks

$F = \{f_1, \ldots, f_k\}$,

$L^A(F) = \cup_{i=1}^{k}$ first-achievers(f_i)

Action Landmark Generation

Action Landmark Graph (V, E)
V: a set of action landmarks, E: a set of edges labeled with natural orderings.



Fact landmarks F and F':

$$F \rightarrow F'$$

Corresponding action landmarks $L^A(F)$ and $L^A(F')$:

$$L^A(F) \rightarrow L^A(F')$$

Status of Action Landmarks in Search

How a fact landmark holds in a state?
How an action landmark holds in a state?

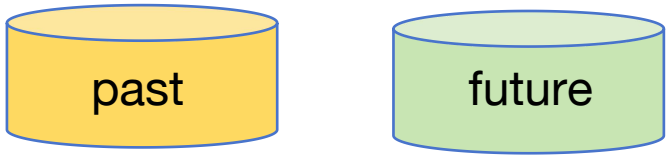
F = {<p-in, e>}
A = { p-board-e-2, e-move-2-1}

Status of Action Landmarks in Search

Transition produced by some action a: s $\xrightarrow{a}$ s'
L = $\{A_1, A_2, \ldots, A_k\}$

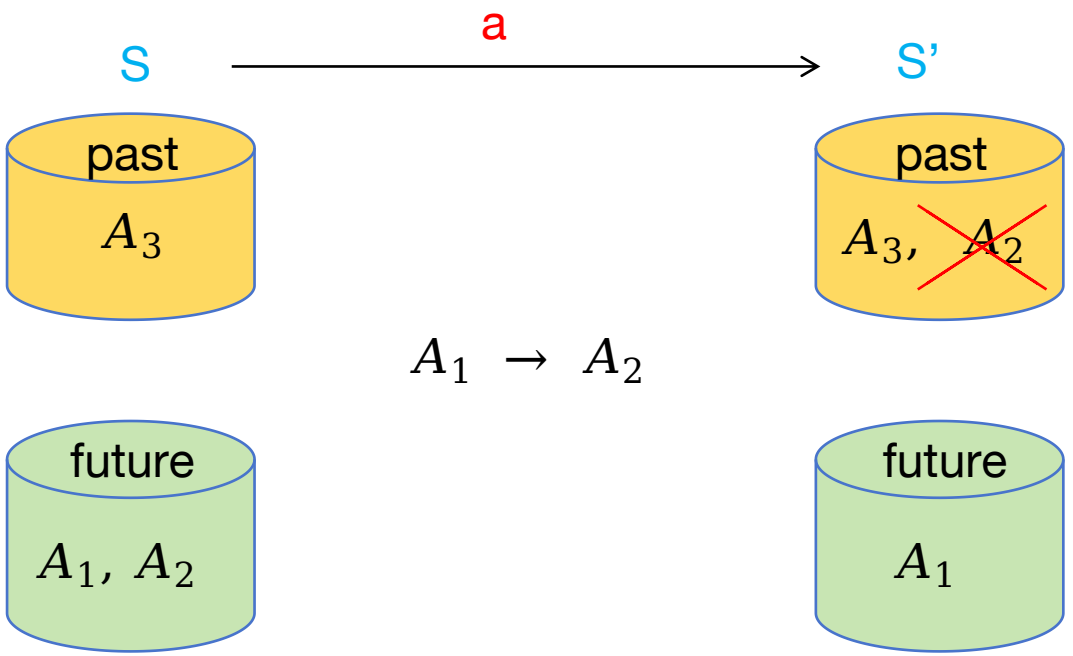
Hit(a) = {A ∈ L | a ∈ A }

To track action landmarks' status at each state s



Example: L = $\{A_1, A_2, A_3\}$,
A_1= {b, c}, A_2= {a}, A_3= {a, d}

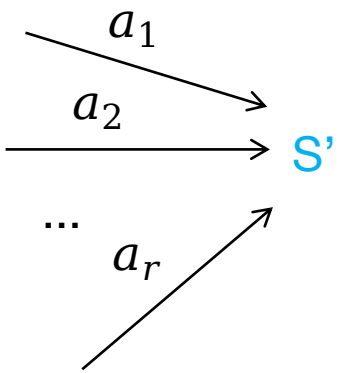
Status of Action Landmarks in Search



Conflict occurs.
$A_2 \in$ Hit(a) is not considered as achieved.

Example: L = {A_1, A_2, A_3},
A_1= {b, c}, A_2= {a}, A_3= {a, d}

Status of Action Landmarks in Search



Conservative merge at s':
$past$ (s') = $\cap_{i=1}^{r}$ $past^{a_i}$(s')
$future$ (s') = $\cup_{i=1}^{r}$ $future^{a_i}$ (s')

Optimistic merge at s':
$past$ (s') = $\cup_{i=1}^{r}$ $past^{a_i}$(s')
$future$ (s') = $\cap_{i=1}^{r} future^{a_i}$ (s')

===> not valid

Example: L = {A_1, A_2, A_3}, A_1= {b, c}, A_2= {a}, A_3= {a, d}
$s_1 \xrightarrow{b} s'$, $s_2 \xrightarrow{d} s'$

After the first transition: past(s') = {A_1}, future(s')= {A_2, A_3}
After the second transition: past(s') = {A_3}, future(s') = {A_1, A_2}
Merge at s' : past(s') = ∅, future(s') = {A_1, A_2, A_3}

Landmark Count Heuristic

Original landmark count heuristic h^{sum}

L^F: a set of fact landmarks that need to be achieved at state s, $F \in L^F$ is a fact landmark, F's first achievers is $o^F = \{o \mid o \in o_F\}$.

$h^{sum}(s) = \sum_{F \in L^F} C(o_F)$

$C(o_F) = min_{o \in o_F} cost(o)$

Adaptation h^{a-sum}

L^A : a set of action landmarks that need to be achieved at state s, $A \in L^A$ is an action landmark

$h^{a-sum}(s) = \sum_{A \in L^A} C(A)$

$C(A) = min_{a \in A} cost(a)$

Experimental results

Universität Basel

What do we compare?

h^{sum} h^{a-sum} 4 fact landmark factories: hm, rhw, exhaust, zg

What search algorithm do we use?

greedy best first search (lazy) algorithm

Resource limit?

- Search time limit: 5 minutes
- Memory limit: 3584MB

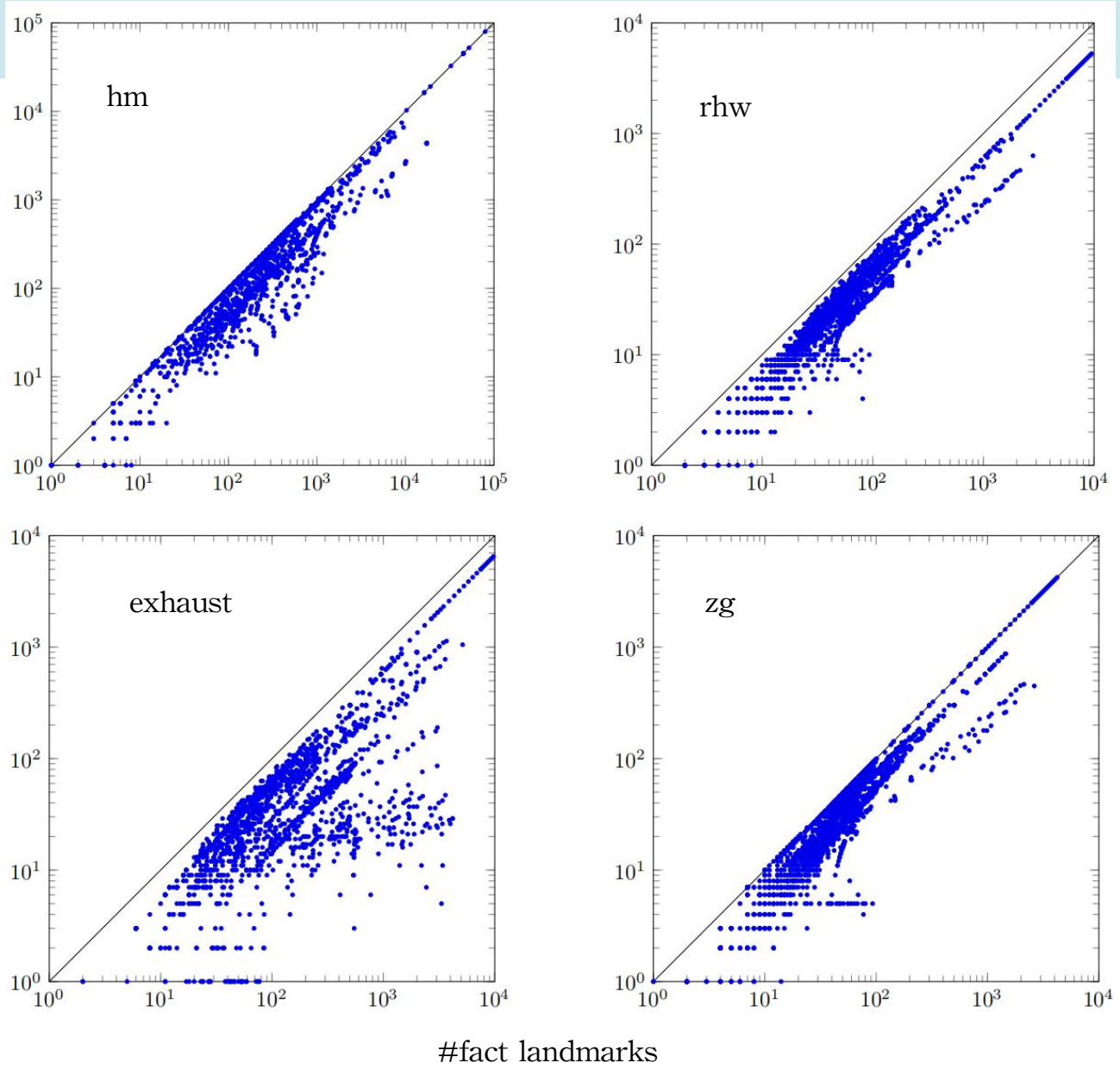
Experimental results - loss of landmarks

$$\text{landmark loss percentage} = \frac{\#\text{fact landmarks} - \#\text{action landmarks}}{\#\text{fact landmarks}}$$

	hm	*rhw*	*exhaust*	*zg*
Average loss per task	33.9%	46.6%	64.5%	37.9%

Universität Basel

Experimental results - loss of landmarks



a high loss ≠ error

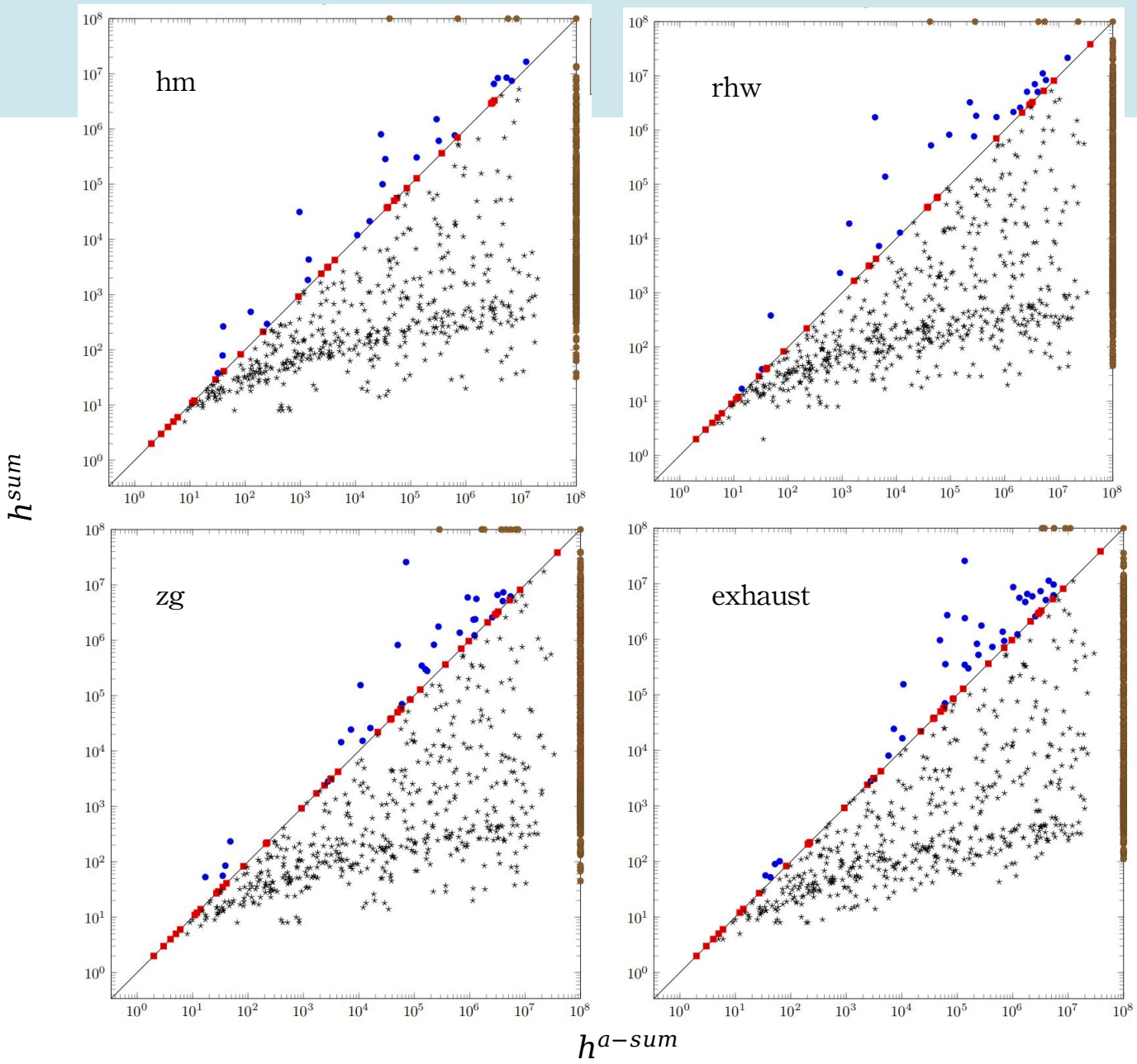
Experimental results - Initial heuristic values

Whether the loss of landmarks introduced by the transformer affects the initial heuristic value ?

$$h^{sum}(s_I) = h^{a-sum}(s_I)$$

	hm	*rhw*	*exhaust*	*zg*
Average initial heuristic value	1765518	59245	79547	34448

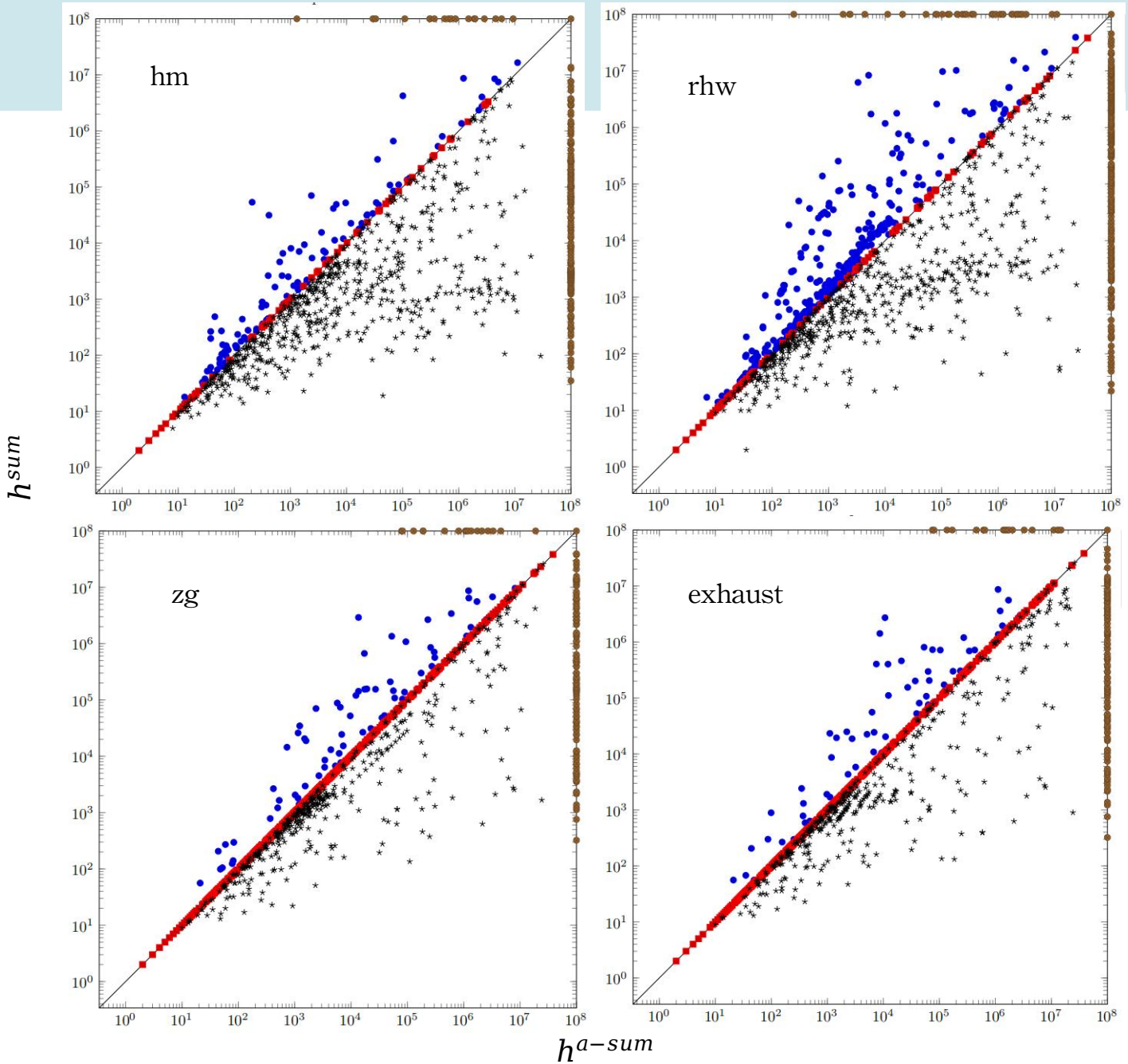
Experimental results - Expansions



h^{sum} VS h^{a-sum} (with conversative merge rule)

When a heuristic can better guide the search, the search algorithm expands fewer states.

Experimental results - Expansions



Percentage of tasks on which the heuristics have identical expansions

	hm	*rhw*	*exhaust*	*zg*
Percentage of tasks	7.2%	8.0%	46.1%	45.7%

h^{sum} VS h^{a-sum} (with optimistic merge rule)

Experimental results - Search time

h^{sum} VS h^{a-sum}(with conversative merge rule)

- Search time score
- Search time per expansion

	$h^{\text{a-sum}}$				h^{sum}			
	hm	*rhw*	*exhaust*	*zg*	*hm*	*rhw*	*exhaust*	*zg*
Mean of search time score	0.22	0.24	0.24	0.24	0.45	0.56	0.50	0.52
Time per expansion	1.43	1.19	0.99	0.10	4.61	4.92	2.88	2.64

Second row: Time per expansion, numbers are multiplied by 10^{-5} and measured in second.

Mean of search time score	0.37	0.47	0.46	0.48

h^{a-sum} (with optimistic merge rule)

Experimental results - Search memory

h^{sum} VS h^{a-sum}(with conversative merge rule)

- Memory score
- Memory per expansion

	$h^{\text{a-sum}}$				h^{sum}			
	hm	*rhw*	*exhaust*	*zg*	*hm*	*rhw*	*exhaust*	*zg*
Mean of memory score	0.15	0.17	0.24	0.24	0.27	0.40	0.50	0.52
Memory per expansion	74.1	38.5	27.4	28.4	681.8	203.7	112.0	109.9

Second row: Memory per expansion is measured in KB

Mean of memory score	0.23	0.36	0.33	0.34

h^{a-sum} (with optimistic merge rule)

Experimental results - Coverage

Coverage records #tasks that search algorithm can solve within the resource limits.

with conservative merge rule

with optimistic merge rule

	$h^{\text{a-sum}}$	h^{sum}	difference	$h^{\text{a-sum}}$	h^{sum}	difference
hm	577	1088	511	913	1088	175
rhw	658	1399	741	1174	1399	225
exhaust	637	1280	643	1160	1280	120
zg	641	1294	653	1180	1294	114

Conclusion and Future work

- action landmark heuristic: less time/memory per state expansion; more expansions; longer search times; higher memory consumption; lower coverages

Future work:

- Let other landmark heuristics use the action landmarks.
- Refine the evaluation of status of action landmarks.
- Build the action landmarks directly on tasks.

```
New best heuristic value for landmark_sum_action_heuristic: 17
g=0, 1 evaluated, 0 expanded
Initial heuristic value for landmark_sum_action_heuristic: 17
New best heuristic value for landmark_sum_action_heuristic: 16
g=1, 2 evaluated, 1 expanded
New best heuristic value for landmark_sum_action_heuristic: 15
g=2, 3 evaluated, 2 expanded
New best heuristic value for landmark_sum_action_heuristic: 14
g=3, 5 evaluated, 4 expanded
New best heuristic value for landmark_sum_action_heuristic: 13
g=4, 6 evaluated, 5 expanded
New best heuristic value for landmark_sum_action_heuristic: 12
g=5, 7 evaluated, 6 expanded
New best heuristic value for landmark_sum_action_heuristic: 11
g=6, 12 evaluated, 11 expanded
New best heuristic value for landmark_sum_action_heuristic: 10
g=7, 13 evaluated, 12 expanded
New best heuristic value for landmark_sum_action_heuristic: 9
g=8, 19 evaluated, 18 expanded
New best heuristic value for landmark_sum_action_heuristic: 8
g=9, 20 evaluated, 19 expanded
New best heuristic value for landmark_sum_action_heuristic: 7
g=10, 28 evaluated, 27 expanded
New best heuristic value for landmark_sum_action_heuristic: 6
g=11, 29 evaluated, 28 expanded
New best heuristic value for landmark_sum_action_heuristic: 5
g=12, 30 evaluated, 29 expanded
New best heuristic value for landmark_sum_action_heuristic: 4
g=13, 40 evaluated, 39 expanded
New best heuristic value for landmark_sum_action_heuristic: 3
g=15, 50 evaluated, 49 expanded
New best heuristic value for landmark_sum_action_heuristic: 2
g=17, 61 evaluated, 60 expanded
New best heuristic value for landmark_sum_action_heuristic: 1
g=19, 74 evaluated, 73 expanded
Solution found!
Actual search time: 0.001113s
```

h^{a-sum}

optimistic merge rule VS conservative merge rule

```
New best heuristic value for landmark_sum_action_heuristic: 17
g=0, 1 evaluated, 0 expanded
Initial heuristic value for landmark_sum_action_heuristic: 17
New best heuristic value for landmark_sum_action_heuristic: 16
g=1, 2 evaluated, 1 expanded
Solution found!
Actual search time: 0.148163s
```

h^{a-sum}

optimistic merge rule VS conservative merge rule

```
Expanded 83 state(s).
Reopened 0 state(s).
Evaluated 84 state(s).
Evaluations: 84
Generated 815 state(s).
Dead ends: 0 state(s).
Number of registered states: 84
Int hash set load factor: 84/128 = 0.656250
Int hash set resizes: 7
Search time: 0.001158s
Total time: 0.009879s
```

```
Expanded 6849 state(s).
Reopened 0 state(s).
Evaluated 6850 state(s).
Evaluations: 6850
Generated 66647 state(s).
Dead ends: 0 state(s).
Number of registered states: 6850
Int hash set load factor: 6850/8192 = 0.836182
Int hash set resizes: 13
Search time: 0.148205s
Total time: 0.157168s
```