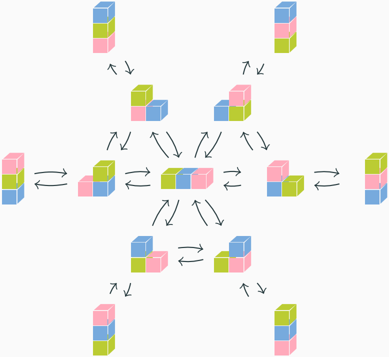
Search Behavior of Greedy Best-First Search

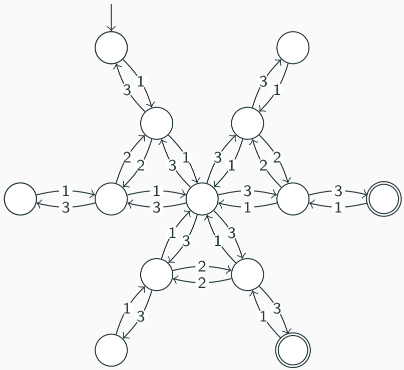
Manuel Heusner

May 10th, 2019

University of Basel

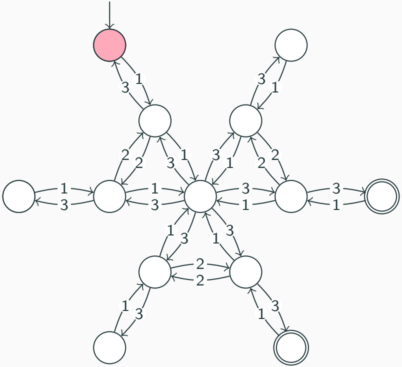
State Spaces




1
3
1
3
2
2
3
1
3
1
2
2
1
3
1
3
3
1
3
1
1
3
1
3
2
2
1
3
1
3

input:

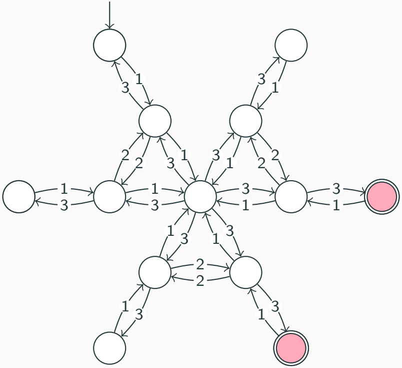
- initial state



State Space Search

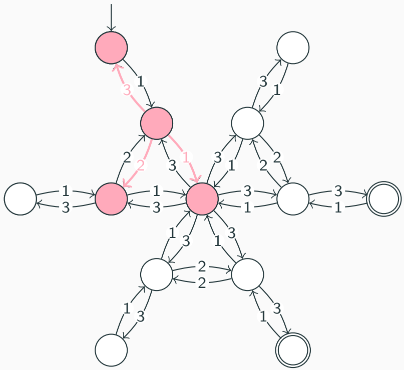
input:

- initial state
- goal test function



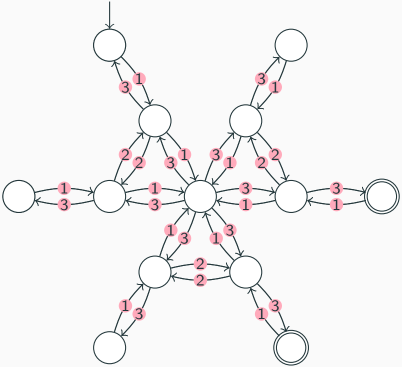
input:

- initial state
- goal test function
- successor generator



input:

- initial state
- goal test function
- successor generator
- transition cost function

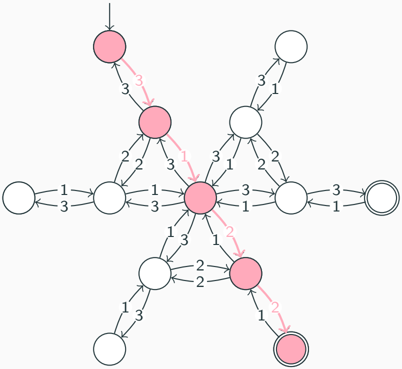


input:

- initial state
- goal test function
- successor generator
- transition cost function

output:

- solution path



input:

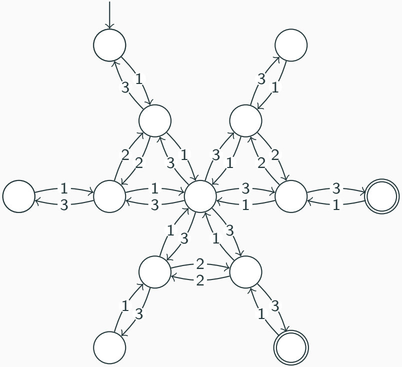
- initial state
- goal test function
- successor generator
- transition cost function

output:

- solution path

additional information:

- heuristic
  - ⇝ heuristic best-first search



information of A*

- c^*: optimal solution path cost
- $f(s)$: estimate of optimal solution path cost

Motivation

information of A*

- c^*: optimal solution path cost
- $f(s)$: estimate of optimal solution path cost

behavior of A*:

- necessary: $f(s) < c^*$
- never: $f(s) > c^*$
- potential: $f(s) = c^*$
- worst case: necessary & potential
- best case: necessary & shortest path of potential states
- progress: increase of f-value

Motivation

information of A*

- c^*: optimal solution path cost
- $f(s)$: estimate of optimal solution path cost

behavior of A*:

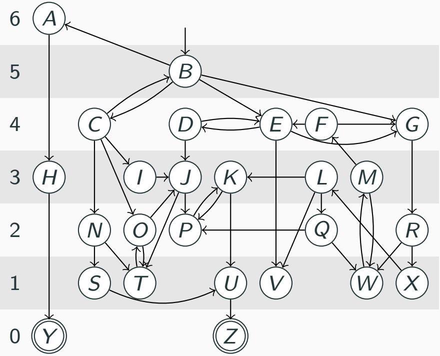
- necessary: $f(s) < c^*$
- never: $f(s) > c^*$
- potential: $f(s) = c^*$
- worst case: necessary & potential
- best case: necessary & shortest path of potential states
- progress: increase of f-value

Can we get similar results for greedy best-first search?

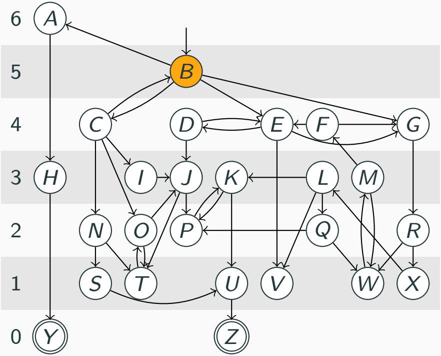
Guiding Questions

Given a state space and a heuristic:

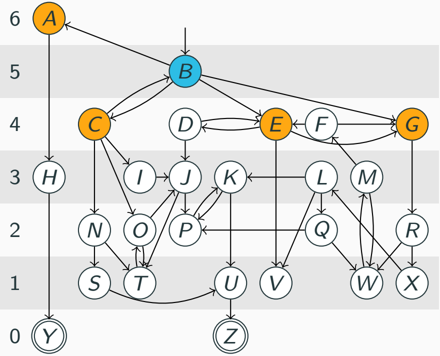
- When does GBFS make search progress?
- Which states does GBFS potentially, never or necessarily expand?
- Which are the best-case and worst-case search runs of GBFS?


6
A
5
B
4
C
D
E
F
G
3
H
I
J
K
L
M
2
N
O
P
Q
R
1
S
T
U
V
W
X
0
Y
Z

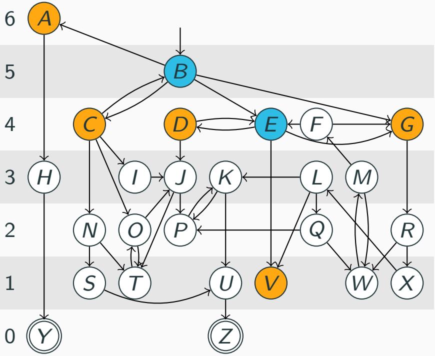
Greedy Best-First Search



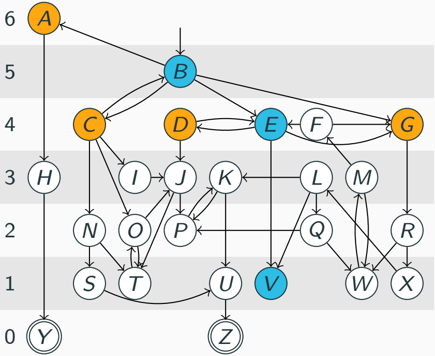
Greedy Best-First Search



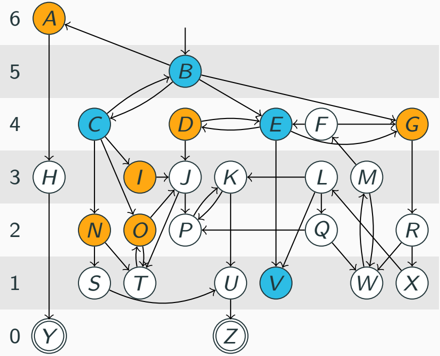
Greedy Best-First Search



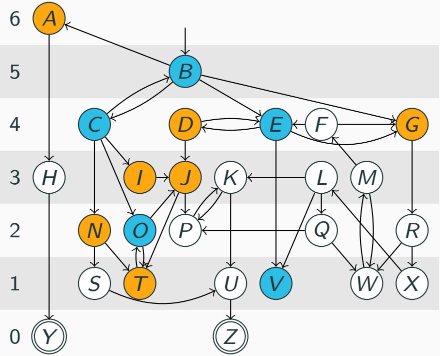
Greedy Best-First Search



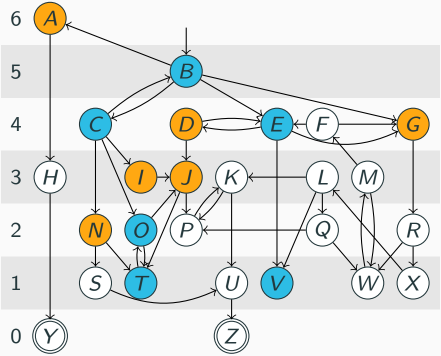
Greedy Best-First Search



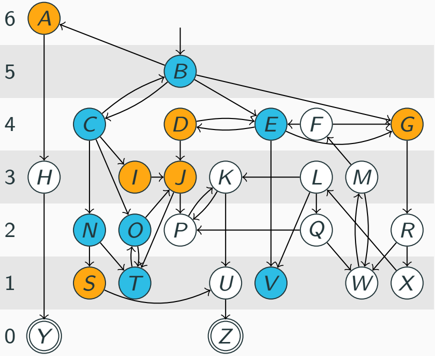
Greedy Best-First Search



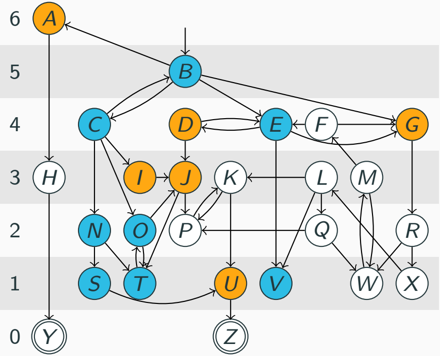
Greedy Best-First Search



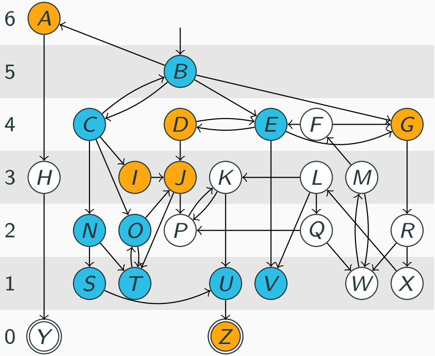
Greedy Best-First Search

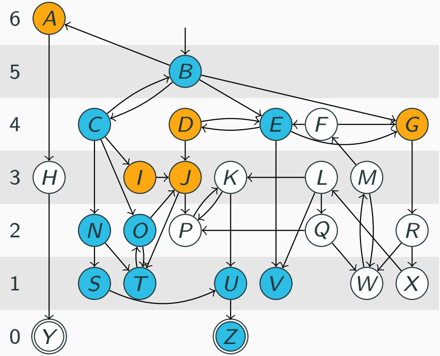


Greedy Best-First Search

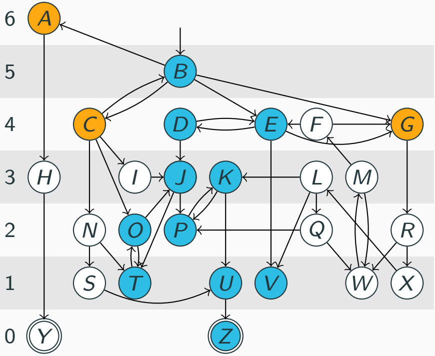


Greedy Best-First Search

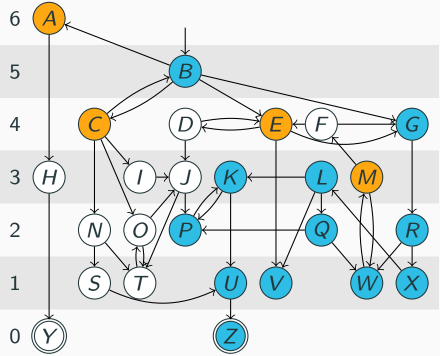



6
5
4
3
2
1
0
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

Greedy Best-First Search



Greedy Best-First Search



When does GBFS make search progress?

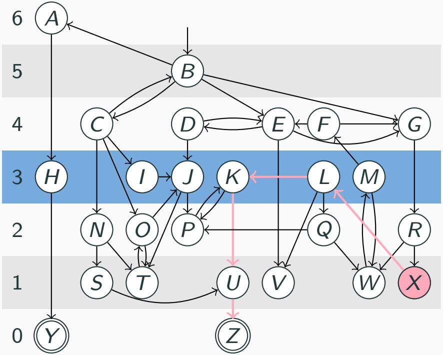
High-Water Mark of State [Wilt & Ruml,2014]

High-Water Mark of State [Wilt & Ruml,2014]

The highest h-value that GBFS reaches during a search run starting in a state.

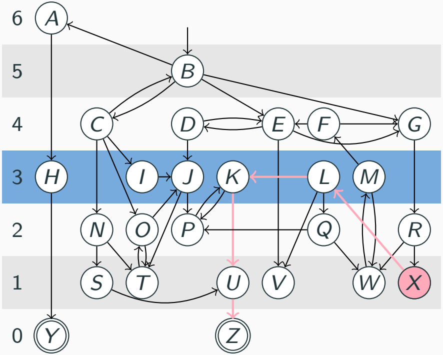
High-Water Mark of State [Wilt & Ruml,2014]

The highest h-value that GBFS reaches during a search run starting in a state.



High-Water Mark of State [Wilt & Ruml,2014]

The highest h-value that GBFS reaches during a search run starting in a state.



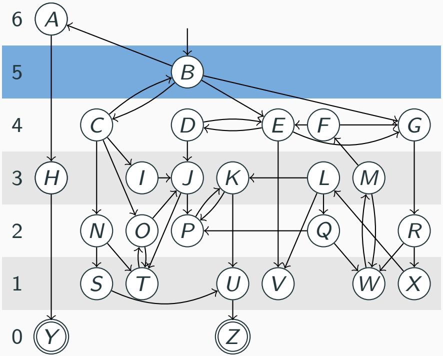
$$hwm(s) := \begin{cases} \min_{\rho \in P(s)}(\max_{s \in \rho} h(s)) & \text{if } P(s) \neq \emptyset \\ \infty & \text{otherwise} \end{cases}$$

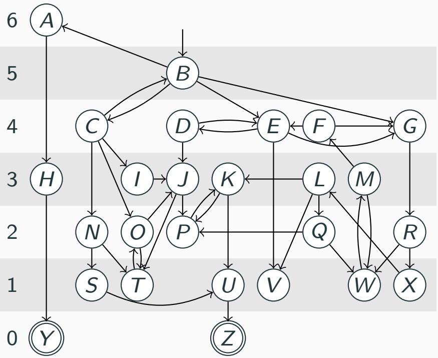
High-Water Mark Pruning [Wilt & Ruml,2014]

GBFS never expands a state s with $h(s) > hwm(s_{init})$.

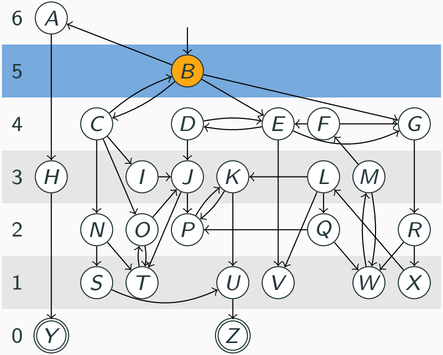
High-Water Mark Pruning [Wilt & Ruml,2014]

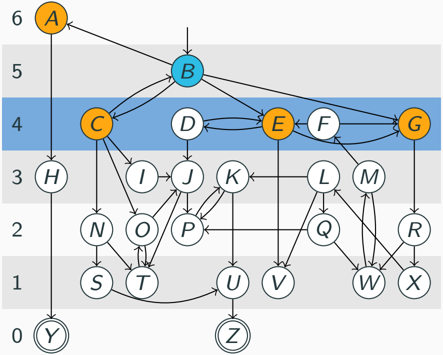
GBFS never expands a state s with $h(s) > hwm(s_{init})$.




6
5
4
3
2
1
0
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

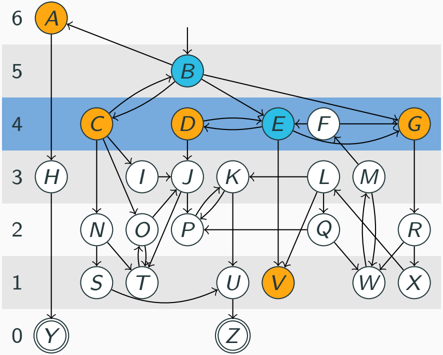
Search Progress




6
5
4
3
2
1
0
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

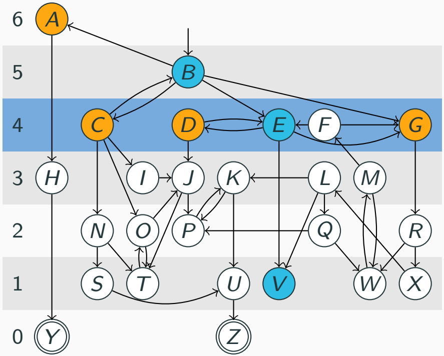
high-water mark of set of states:

$$hwm(S) := \min_{s \in S}(hwm(s))$$



high-water mark of set of states:

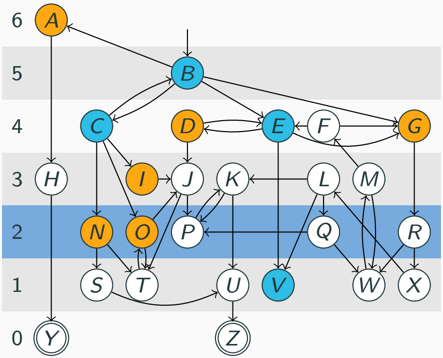
$$hwm(S) := \min_{s \in S}(hwm(s))$$



Search Progress

high-water mark of set of states:

$$hwm(S) := \min_{s \in S}(hwm(s))$$

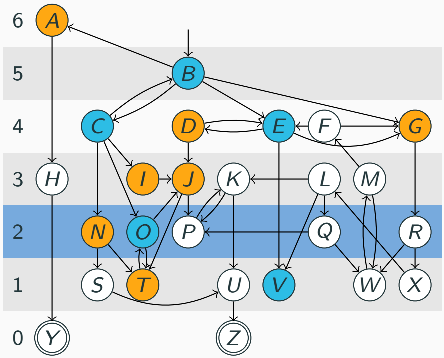


high-water mark of set of states:

$$hwm(S) := \min_{s \in S}(hwm(s))$$

progress state:

$$hwm(s) > hwm(succ(s))$$



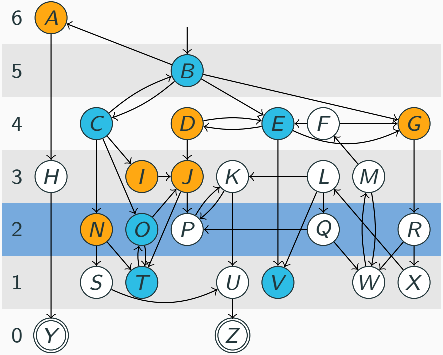
high-water mark of set of states:

$$hwm(S) := \min_{s \in S}(hwm(s))$$

progress state:

$$hwm(s) > hwm(succ(s))$$

episodes of local searches!



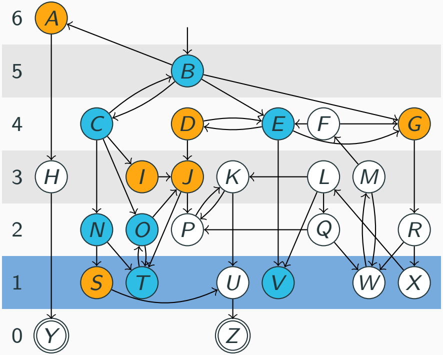
high-water mark of set of states:

$$hwm(S) := \min_{s \in S}(hwm(s))$$

progress state:

$$hwm(s) > hwm(succ(s))$$

episodes of local searches!



Search Progress

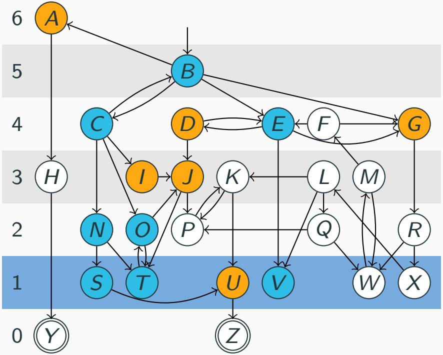
high-water mark of set of states:

$$hwm(S) := \min_{s \in S}(hwm(s))$$

progress state:

$$hwm(s) > hwm(succ(s))$$

episodes of local searches!



Search Progress

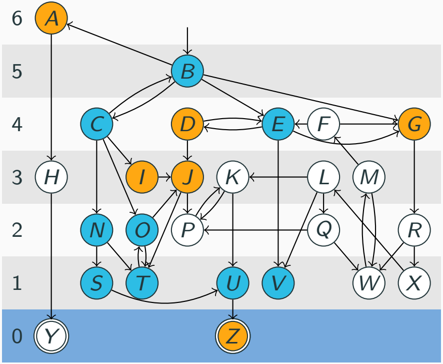
high-water mark of set of states:

$$hwm(S) := \min_{s \in S}(hwm(s))$$

progress state:

$$hwm(s) > hwm(succ(s))$$

episodes of local searches!



high-water mark of set of states:

$$hwm(S) := \min_{s \in S}(hwm(s))$$

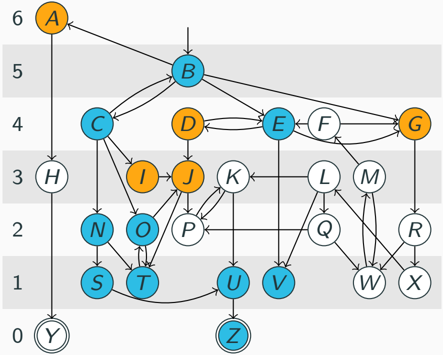
progress state:

$$hwm(s) > hwm(succ(s))$$

episodes of local searches!

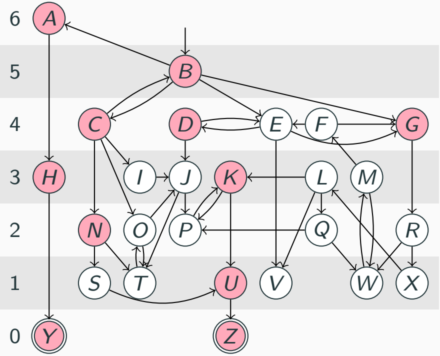
Search Progress

GBFS makes progress when expanding a progress state.

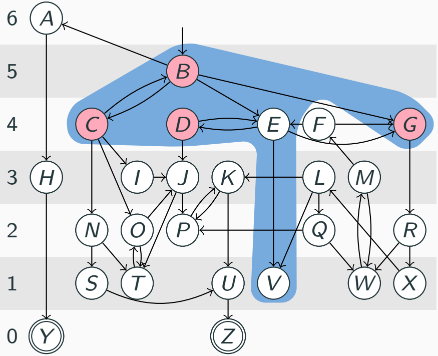


Which states does GBFS potentially or never expand?

Progress States

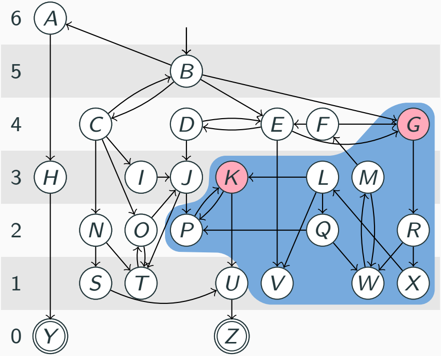


- progress state s induces bench $\mathcal{B}(s)$



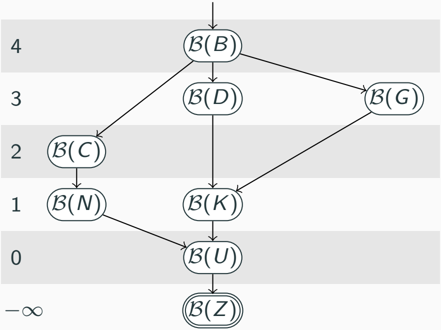
Benches

- progress state s induces bench $\mathcal{B}(s)$



Bench Space

- connects the benches via progress states

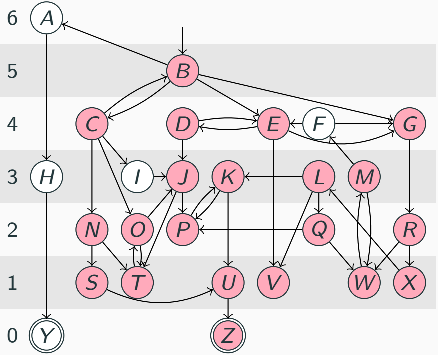


Potentially or Never Expanded States

Potentially and Never Expanded States

GBFS potentially expands a state that is on at least one bench from the bench space.

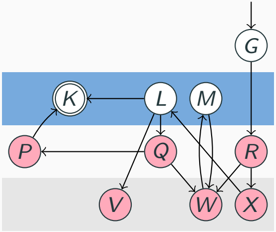
GBFS never expands all other states.



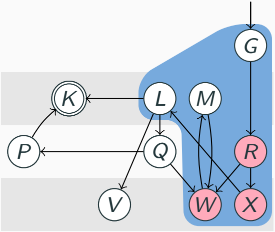
Which states does GBFS necessarily expand?

Crater and Surface States

- crater state: $h(s) < hwm$ of bench
- surface states: all other states on the bench

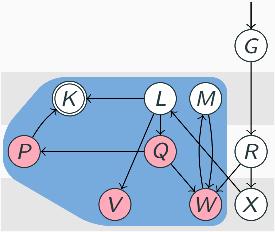


- surface state s induces crater $\mathcal{C}(s)$



Craters

- surface state s induces crater $\mathcal{C}(s)$

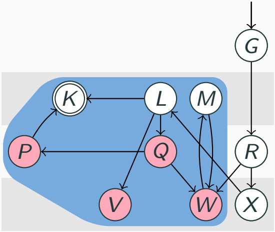


Craters

- surface state s induces crater $\mathcal{C}(s)$

Necessarily Expanded States

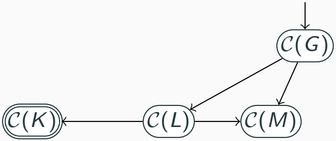
If GBFS expands a surface state s on a bench, then it necessarily expands all the crater states from crater $\mathcal{C}(s)$.



Which is a best-case search run of GBFS?

Crater Space

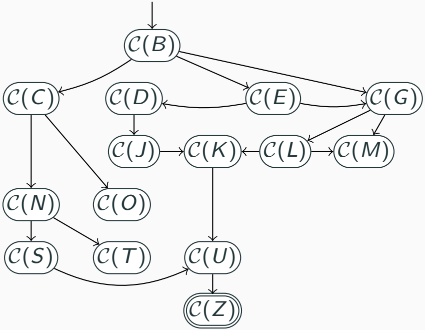
- connects craters of a bench via surface states



Best-Case Search Run

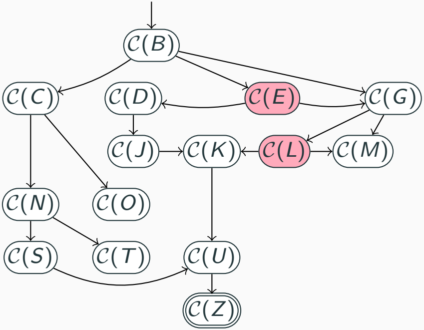
Best-Case Search Run

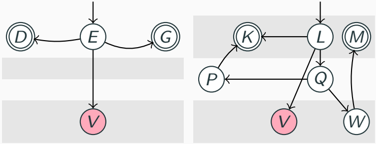
- path in crater space
- minimize length of path and number of crater states



Best-Case Search Run

- path in crater space
- minimize length of path and number of crater states





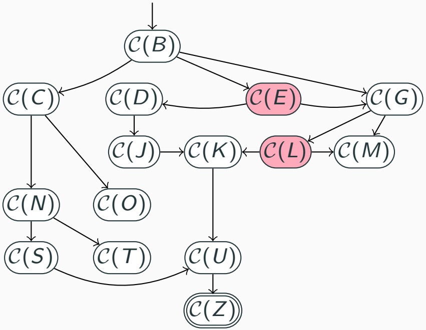
Best-Case Search Run

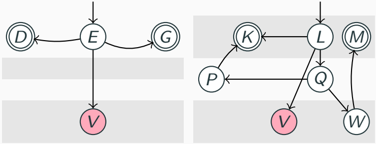
- path in crater space
- minimize length of path and number of crater states

Complexity Results

Given a state space and heuristic:

- NP-complete
- polynomial-time if overlap-free or undirected

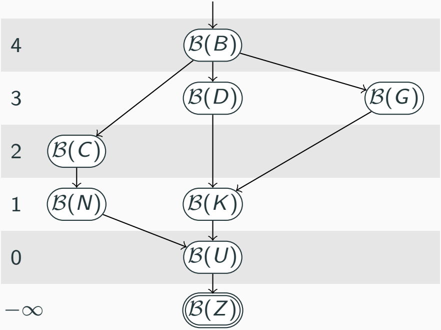




Which is a worst-case search run of GBFS?

Worst-Case Search Run

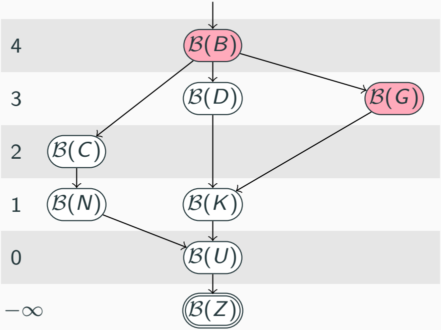
- path in bench space
- maximize length of path and number of non-progress states

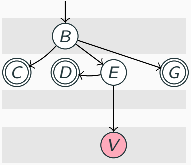


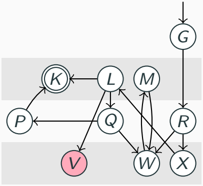
Worst-Case Search Run

Worst-Case Search Run

- path in bench space
- maximize length of path and number of non-progress states



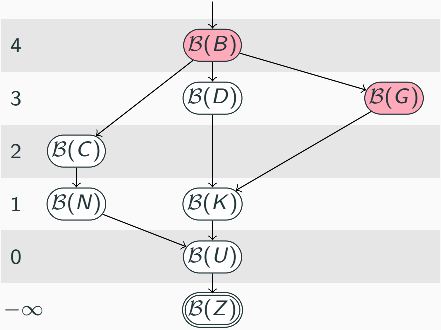




Worst-Case Search Run

Worst-Case Search Run

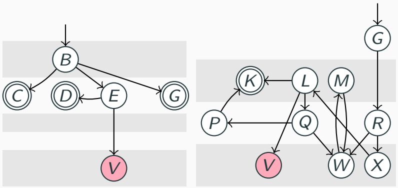
- path in bench space
- maximize length of path and number of non-progress states



Complexity Results

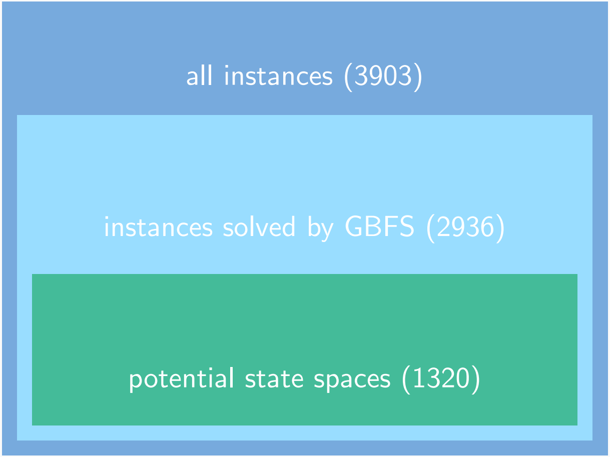
Given a state space and heuristic:

- NP-complete
- polynomial-time if overlap-free or undirected

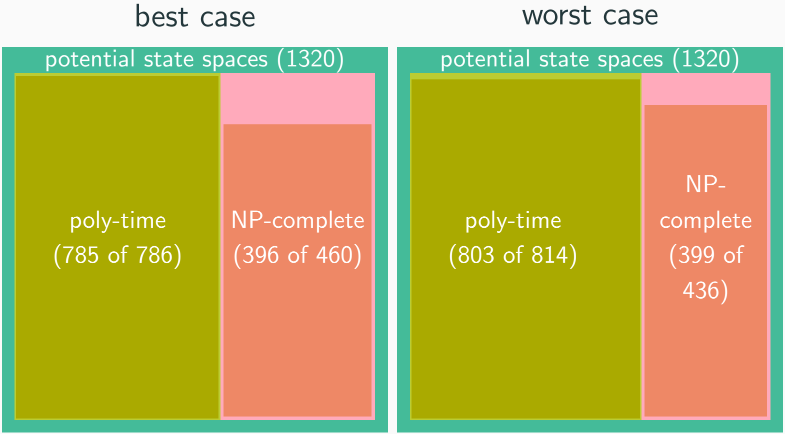


- implemented algorithms for extracting the search behavior
- state spaces: classical planning tasks from international planning competitions
- heuristic: h^{ff}

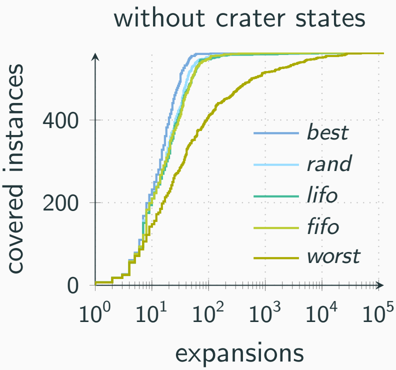
Feasibility: Potential State Space

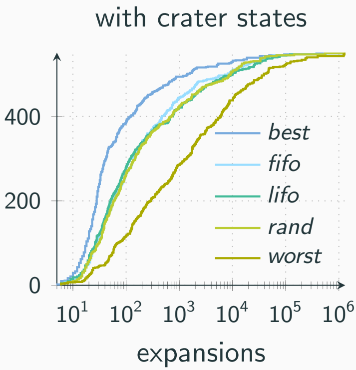


Feasibility: Best-Case and Worst-Case Search Runs



Tie-Breaking Policies





Conclusion

- search progress based on high-water mark
- criterion for expanded states based on benches and craters
- characterization of best-case and worst-case search runs based on bench space and crate space
- demonstrated potential for improvement of tie-breaking