Search Behavior of Greedy Best-First Search

Manuel Heusner
May 10th, 2019
University of Basel
State Spaces
State Space Search

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

- output:
  - solution path

- additional information:
  - heuristic
    - heuristic best-first search
State Space Search

input:

- initial state
State Space Search

input:

- initial state
- goal test function
State Space Search

input:

- initial state
- goal test function
- successor generator

output:

- solution path

additional information:

- heuristic \(\Rightarrow\) heuristic best-first search
State Space Search

input:

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

additional information:
- heuristic
  ⇝ heuristic best-first search
State Space Search

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
  \(\rightarrow\) heuristic best-first search
information of $A^*$

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

Can we get similar results for greedy best-first search?
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?
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
Greedy Best-First Search
Greedy Best-First Search
When does GBFS make search progress?
High-Water Mark of State [Wilt & Ruml, 2014]

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

\[
h_{wm}(s) := \begin{cases} 
\min_{\rho \in P(s)} \max_{s' \in \rho} h(s') & \text{if } P(s) \neq \emptyset \\
\infty & \text{otherwise}
\end{cases}
\]
High-Water Mark of State [Wilt & Ruml, 2014]

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

\[ \text{hwm}(s) := \begin{cases} \min_{\rho \in \mathcal{P}(s)} (\max_{s \in \rho} h(s)) & \text{if } \mathcal{P}(s) \neq \emptyset \\ \infty & \text{otherwise} \end{cases} \]
The highest $h$-value that GBFS reaches during a search run starting in a state.

$$h_{wm}(s) := \begin{cases} \min_{\rho \in \mathcal{P}(s)} \max_{s \in \rho} h(s) & \text{if } \mathcal{P}(s) \neq \emptyset \\ \infty & \text{otherwise} \end{cases}$$
The highest h-value that GBFS reaches during a search run starting in a state.

\[ hwm(s) := \begin{cases} 
\min_{\rho \in P(s)} (\max_{s \in \rho} h(s)) & \text{if } P(s) \neq \emptyset \\
\infty & \text{otherwise} 
\end{cases} \]
GBFS never expands a state $s$ with $h(s) > hwm(s_{init})$. 
GBFS never expands a state $s$ with $h(s) > hwm(s_{init})$. 
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: $hwm(s) > hwm(succ(s))$

GBFS makes progress when expanding a progress state.
high-water mark of set of states: \( hwm(S) := \min_{s \in S} hwm(s) \)

progress state: \( hwm(s) > hwm(\text{succ}(s)) \)

Search Progress

GBFS makes progress when expanding a progress state.
high-water mark of set of states:

\[ hwm(S) := \min_{s \in S} hwm(s) \]
high-water mark of set of states:

\[ hwm(S) := \min_{s \in S} hwm(s) \]
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!
high-water mark of set of states:

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

progress state:

\[ hwm(s) > hwm(\text{succ}(s)) \]

episodes of local searches!
Search Progress

high-water mark of set of states:

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

progress state:

$$hwm(s) > hwm(\text{succ}(s))$$

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

A → B

C → D → E → F → G

H → I → J → K

N → O → P → Q → R

S → T → U → V → W → X

Y → Z
• progress state $s$ induces bench $B(s)$
• progress state $s$ induces bench $B(s)$
Bench Space

- connects the benches via progress 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 \( C(s) \)
• surface state $s$ induces crater $C(s)$
- surface state $s$ induces crater $C(s)$

Necessarily Expanded States

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

Diagram:

- Nodes: $P, Q, V, W, K, L, M, R, X$
- Edges: connecting the nodes as shown in the graph.
Which is a best-case search run of GBFS?
• connects craters of a bench via surface 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
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

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

[Diagram of a search tree with states and transitions labeled, showing states such as B(B), B(D), B(G), B(C), B(N), B(K), B(U), B(Z), C, D, E, G, K, L, M, P, Q, R, V, W, X, G.]

Complexity Results

Given a state space and heuristic:

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

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

- All instances (3903)
  - Instances solved by GBFS (2936)
  - Potential state spaces (1320)
Feasibility: Best-Case and Worst-Case Search Runs

**Best Case**
- Potential state spaces: 1320
- Poly-time: (785 of 786)
- NP-complete: (396 of 460)

**Worst Case**
- Potential state spaces: 1320
- Poly-time: (803 of 814)
- NP-complete: (399 of 436)
Tie-Breaking Policies

without crater states

covered instances

expansions

with crater states

covered instances

expansions

- best
- rand
- lifo
- fifo
- worst

- best
- fifo
- lifo
- rand
- worst
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