Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)

# Search Progress and Potentially Expanded States in Greedy Best-First Search

**Manuel Heusner, Thomas Keller** and **Malte Helmert**
University of Basel, Switzerland
{manuel.heusner,tho.keller,malte.helmert}@unibas.ch

## Abstract

A classical result in optimal search shows that A* with an admissible and consistent heuristic expands every state whose $f$-value is below the optimal solution cost and no state whose $f$-value is above the optimal solution cost. For satisficing search algorithms, a similarly clear understanding is currently lacking. We examine the search behavior of greedy best-first search (GBFS) in order to make progress towards such an understanding.

We introduce the concept of *high-water mark benches*, which separate the search space into areas that are searched by a GBFS algorithm in sequence. High-water mark benches allow us to exactly determine the set of states that are expanded by at least one GBFS tie-breaking strategy and give us a clearer understanding of search progress.

## 1 Introduction

Many classical algorithms for state-space search, such as greedy best-first search [Doran and Michie, 1966], A$^*$ [Hart *et al.*, 1968], Weighted A$^*$ [Pohl, 1970] and IDA$^*$ [Korf, 1985], are representatives of a general family of uni-directional, expansion-based heuristic search algorithms. Such algorithms are largely agnostic to the state space to be searched, only requiring two pieces of information to be applicable in a given domain: a *generative model* of the state space that provides the *initial state*, tests if a state is a *goal state* and produces all *successor states* of a given state along with the *action costs*; and a *heuristic function* which estimates the cost-to-go or distance-to-go from a given state.

Optimal search algorithms in this family have a fairly well-developed theory [e.g., Dechter and Pearl, 1985]. For example, we know that the A$^*$ algorithm [Hart *et al.*, 1968] is optimal with *admissible* heuristics and never reexpands states with *consistent* heuristic. Moreover, for admissible and consistent heuristics $h$, it is well-known that A$^*$ will expand a state $s$ if $f(s) < c^*$, and A$^*$ will not expand $s$ if $f(s) > c^*$, where $c^*$ is the optimal solution cost, $f(s) = g(s) + h(s)$ and $g(s)$ is the shortest-path cost from the initial state to $s$.

While this criterion is not perfect – it does not predict whether or not states $s$ with $f(s) = c^*$ are expanded – it goes a large way towards explaining the search behavior of A$^*$. Theoretical results of this kind are useful to explain cases where A$^*$ performs poorly [e.g., Helmert and Röger, 2008]. They can also show how to improve the performance of A$^*$-style search algorithms, for example by emphasizing the importance of *tie-breaking* [Asai and Fukunaga, 2017].

In A$^*$ searches with admissible heuristics, expanding a state with a higher $f$-value than any previously expanded state is a meaningful event because every time it happens, a new lower bound for the optimal solution cost has been proven. Consequently, such an occurrence is often interpreted as a measure of *progress* of the search.

For *satisficing* (non-optimal) algorithms in the family, a comparably deep understanding of search progress and states that will or will not be expanded is currently lacking. Many new algorithms for satisficing search have been proposed in recent years [e.g., Imai and Kishimoto, 2011; Xie *et al.*, 2014b; 2014a; Valenzano *et al.*, 2014], yet our understanding of the behavior of such algorithms is still quite limited. For example, a recent study by Wilt and Ruml [2015] demonstrates that (and why) improving the accuracy of an admissible heuristic can be highly detrimental for greedy search, while being extremely beneficial for A$^*$.

This paper contributes to the understanding of satisficing search by studying the search behavior of greedy best-first search (GBFS), the most commonly considered satisficing search algorithm. We show that GBFS runs are *episodic* in the sense that each algorithm run can be understood as a sequence of smaller subsearches, where each subsearch occurs within a single *bench* of the underlying state space. Each search episode begins and ends by expanding a so-called *progress state*, which can be characterized in terms of *high-water marks*. Based on this observation, we exactly characterize the set of states that are *potentially expanded* by GBFS with a given heuristic. i.e., the set of states that are expanded by at least one tie-breaking strategy. This paper is based on a longer paper published at SoCS 2017 [Heusner *et al.*, 2017], and we refer to the SoCS paper for proofs.[1]

[1] We define some concepts slightly differently here compared to the SoCS paper. Specifically, we make a state that induces a bench part of the bench, and the benches defined here correspond to *reduced* benches in the SoCS paper. These changes simplify and streamline the presentation, but care is needed when considering the proofs of the SoCS paper with the definitions of this paper.

## 2 Background

**State Space Topology** We consider search algorithms that operate on a *state space* $\mathcal{S} = \langle S, s_I, S_\star, succ \rangle$, where $S$ is a finite set of states, $s_I \in S$ is the initial state, $S_\star \subseteq S$ is the set of goal states and $succ$ is a successor function that maps each state $s \in S$ to a set of successor states.

A sequence of pairwise distinct states $\rho = \langle s_0, \ldots, s_n \rangle$ is a (cycle-free) *path* from $s_0$ to $s_n$ if $s_i \in succ(s_{i-1})$ for $i = 1, \ldots, n$. A path $\langle s_0, \ldots, s_n \rangle$ is an *$s$-plan* if $s_0 = s$ and $s_n \in S_\star$. With $P(s)$, we denote the set of $s$-plans for $s$. State $s'$ is *reachable* from $s$ if there is a path from $s$ to $s'$.

A *state space topology* $\mathcal{T} = \langle \mathcal{S}, h \rangle$ is a state space $\mathcal{S}$ combined with a *heuristic function* $h : S \to \mathbb{R}_0^+$. Typically, $h(s)$ estimates the cost of a cheapest $s$-plan. In this work, it suffices to regard the heuristic function as an arbitrary black-box function that assigns some non-negative, finite real number to each state. To simplify some definitions, we assume that the initial state has a larger heuristic value than all other states and that all goal states have lower heuristic values than all non-goal states. This does not affect the behavior of GBFS: the initial state is always the first expanded state regardless of its heuristic value, and the heuristic values of goal states never need to be considered because GBFS may terminate as soon as a goal state is generated.

**Greedy Best-First Search** The input to GBFS is a state space topology, and the output is an $s_I$-plan if one exists and "unsolvable" otherwise. GBFS is driven by the assumption that states with lower heuristic values are closer to a goal state. In each step, it expands a state that has the lowest heuristic value among all states that have been generated before but have not been expanded yet, until a goal state is generated. Because of its greediness, it provides no quality guarantee of the computed $s_I$-plan. Due to heuristic inaccuracy, most challenging search problems contain (possibly prohibitively large) heuristic plateaus or local minima where the heuristic does not provide guidance. Then, GBFS often faces situations where it has to decide which state to expand next among a set of states with identical heuristic value.

For this reason, GBFS is actually not a well-defined algorithm but rather a *family* of algorithms that differ in a single parameter, the *tie-breaking strategy*. Formally, a GBFS tie-breaking strategy $\tau$ for a state space topology $\langle \mathcal{S}, h \rangle$ with states $S$ maps all possible non-empty sets $S^k \subseteq \{s \in S \mid h(s) = k\}$ to a state $s \in S^k$. We refer to GBFS coupled with a specific tie-breaking strategy as an *instance* of GBFS.

In every iteration of GBFS where generated but unexpanded states (*open states*) still exist and no solution has yet been found, GBFS with tie-breaking strategy $\tau$ expands the state $\tau(S_{\min})$, where $S_{\min}$ is the set of all open states with minimal $h$-value. The tie-breaking strategy is the only parameter that sets different instances of GBFS apart and uniquely determines the sequence of state expansions. The *search realization* of a GBFS search with tie-breaking strategy $\tau$ is a sequence of states $r^\tau = \langle s_1, \ldots, s_n \rangle$, where $s_1 = s_I$, $s_i$ is the $i$-th expanded state following $\tau$, and $s_n$ is either a goal state, or $\{s_1, \ldots, s_n\}$ is the set of all reachable states in case no solution exists.

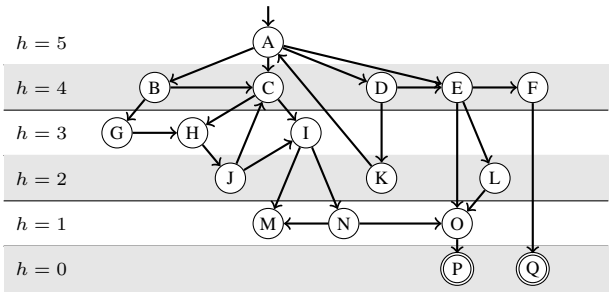


Figure 1: State space topology of our running example.

**High-Water Mark** An earlier attempt to explain the behavior of GBFS is due to Wilt and Ruml [2014], who base their analysis on the *high-water mark* of a state.

**Definition 1** (High-water mark)**.** *Let $\langle \mathcal{S}, h \rangle$ be a state space topology with states $S$. The* high-water mark *of $s \in S$ is*

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

*We define the high-water mark of a set of states $S' \subseteq S$ as*

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

Intuitively, the high-water mark of a state $s$ measures how high the heuristic values of expanded states must climb before a solution can be found in a search starting from $s$. Wilt and Ruml define high-water marks for individual states. We extend this definition to sets of states $S'$ by selecting the minimum high-water mark of any of the states in $S'$. This reflects the intuition that if search begins with a set of candidate states $S'$, then search will eventually follow the "path of least resistance" among the states in $S'$.

**Example 1.** *As a running example, consider the search space topology $\mathcal{T} = \langle \langle \{\mathrm{A}, \ldots, \mathrm{Q}\}, \mathrm{A}, \{\mathrm{P}, \mathrm{Q}\}, succ \rangle, h \rangle$ with $succ$ given by the arcs and $h$ by the shaded regions in Figure 1.*

*The set of* J*-plans in $\mathcal{T}$ is $P(\mathrm{J}) = \{\langle \mathrm{J}, \mathrm{C}, \mathrm{I}, \mathrm{N}, \mathrm{O}, \mathrm{P} \rangle, \langle \mathrm{J}, \mathrm{I}, \mathrm{N}, \mathrm{O}, \mathrm{P} \rangle\}$, and the set of* M*-plans is $P(\mathrm{M}) = \emptyset$. Therefore, $hwm(\mathrm{J}) = 3$, $hwm(\mathrm{M}) = \infty$, and $hwm(\{\mathrm{J}, \mathrm{M}\}) = 3$.*

*The search realization $r^\tau = \langle \mathrm{A}, \mathrm{D}, \mathrm{K}, \mathrm{C}, \mathrm{H}, \mathrm{J}, \mathrm{I}, \mathrm{M}, \mathrm{N}, \mathrm{O}, \mathrm{P} \rangle$ for a GBFS tie-breaking strategy $\tau$ computes the* A*-plan $\langle \mathrm{A}, \mathrm{C}, \mathrm{I}, \mathrm{N}, \mathrm{O}, \mathrm{P} \rangle$.*

Wilt and Ruml use the high-water mark of the initial state to distinguish two kinds of states: ones that are *certainly not* expanded by any GBFS instance regardless of the used tie-breaking criterion (all $s \in S$ with $h(s) > hwm(s_I)$) and one for which we *do not know* if they are expanded or not (all remaining states). In the following we refine this result to obtain a clear classification for a larger number of states.

## 3 Search Progress

Expanding a state with higher $f$-value than any previously expanded state is often regarded as a measure of progress of A$^*$ search with an admissible heuristic. In GBFS-style searches, somewhat analogously, an event that is often considered meaningful is when the search expands a state with a

| $s$ | **A** | D | K | **C** | H | J | **I** | M | N | **O** | P |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $h(s)$ | **5** | **4** | **2** | 4 | 3 | 2 | 3 | **1** | 1 | 1 | **0** |
| $hwm(s)$ | **5** | **4** | 5 | 4 | **3** | 3 | 3 | $\infty$ | **1** | 1 | **0** |

Table 1: Heuristic values and high-water marks for our example realization $r^\tau$. Bold underlined states are progress states. Bold values for $h(s)$ and $hwm(s)$ indicate all-time lows.

lower heuristic value than all previously expanded states. For example, the boosted dual-queue search algorithm by Richter and Helmert [2009] uses such an occurrence to further prioritize expansions based on preferred operators, and the GBFS-LE family of algorithms by Xie *et al.* [2014a] uses it to determine when a GBFS search has become stalled.

Although it sounds intuitively appealing, reaching a new lowest $h$-value in GBFS does not necessarily translate into meaningful, quantifiable progress in finding a solution. In the search realization $r^\tau$ of our running example, K is the first state with a heuristic value of 2 that is expanded, but the expansion of K certainly does not lead to meaningful progress, as the only successor of K is the initial state A (Table 1). In contrast, the states C and I, which are expanded after K and have heuristic values larger than $h(\mathrm{K})$, are part of the computed A-plan and do intuitively look much more like states where progress is made than K.

Given the role of high-water marks in Wilt and Ruml's study of GBFS, another plausible idea is to focus on states that have a lower high-water mark than any previously expanded states, and indeed these states correspond to a somewhat more meaningful measure of progress: whenever we expand a state with a new lowest high-water mark $k$, we are guaranteed that after this point, no state with a heuristic value larger than $k$ will be expanded. However, this notion of progress is still somewhat vague (again, see Table 1). In the running example, we see that D and H are among the expanded states with a high-water mark lower than that of any previously expanded state, but they do not participate in the solution found by the algorithm and do not play a critical role in the search realization.

A slight tweak of the high-water mark idea gives us the desired progress criterion: rather than focus on the algorithm steps where a state with a new lowest high-water mark is *expanded*, we focus on algorithm steps where a state with a new lowest high-water mark is *generated*. Perhaps surprisingly, it turns out that states with this property can be characterized by a test that is completely independent of the tie-breaking strategy or search history: a state expansion generates a state with a high-water mark lower than that of any previously expanded or generated state. This test applies to all states $s$ from any search realization of every state space topology and is exactly expressed with $hwm(succ(s)) < hwm(s)$, i.e., state $s$ has a successor whose high-water mark is lower than the high-water mark of $s$ itself. We call states that pass this test *progress states*.

**Definition 2.** *Let $\langle \mathcal{S}, h \rangle$ be a state space topology with set of states $S$. A* progress state *is a state $s \in S$ with $hwm(s) > hwm(succ(s))$.*

In the running example, the progress states are A, C, I, and O (Table 1).

It turns out that progress states play a very important role in the behavior of GBFS. Consider a situation during the execution of GBFS where the set of states in the open list of GBFS is $S'$, and the algorithm expands the progress state $s \in S$. Then it is guaranteed that *none of the states $s' \in S' \setminus \{s\}$ will ever be expanded.*

This is due to two observations. Firstly, because GBFS chooses to expand $s$, all states $s' \in S' \setminus \{s\}$ satisfy $h(s') \geq h(s)$. Secondly, because $s$ is a progress state, we have $hwm(succ(s)) < h(s)$, which implies that $s$ has a successor (generated upon expanding $s$) from which a goal state can be reached while only expanding states with heuristic values strictly lower than $h(s)$. Therefore, none of the states in $S'$ can ever become a candidate for expansion again. We will use this insight to better understand the set of states that are potentially expanded by GBFS (i.e., expanded by at least one tie-breaking strategy).

## 4 Potentially Expanded States

The properties of progress states imply that every GBFS realization can be understood as a sequence of search episodes, where each episode begins when a progress state is expanded and ends when the next progress state is expanded. Our technical restrictions on heuristic values ensure that the initial state is always a progress state and that every predecessor state of a goal state is a goal state, which means that we do not need special treatment for the first or last episode. (The last episode is simply the one that ends with the expansion of a predecessor state of a goal state.)

Therefore, to analyze which states can be potentially expanded by GBFS, it is sufficient to understand (A) how a search from one progress state to the next proceeds, and (B) in which order different progress states can be reached. For (A), we define *high-water mark benches* (or simply *benches*), which represent a single search episode, beginning with the expansion of a progress state and ending with the expansion of another progress state.

**Definition 3.** *Let $\langle \mathcal{S}, h \rangle$ be a state space topology with set of states $S$. Let $s \in S$ be a progress state.*

*The* bench level *of $s$ is $level(s) = hwm(succ(s))$.*

*The* inner bench states *$inner(s)$ for $s$ consist of all states $s'' \neq s$ that can be reached from $s$ on paths on which all states $s' \neq s$ (including $s''$ itself) are non-progress states and satisfy $h(s') \leq level(s)$.*

*The* bench exit states *$exit(s)$ for $s$ consist of all progress states $s'$ with $h(s') \leq level(s)$ that are successors of $s$ or of some inner bench state of $s$.*

*The* bench states *$states(s)$ for $s$ are $\{s\} \cup inner(s) \cup exit(s)$.*

*The* bench *induced by $s$, denoted by $\mathcal{B}(s)$, is the state space with states $states(s)$, initial state $s$, goal states $exit(s)$, and whose successor function is the successor function of $\mathcal{S}$ restricted to $states(s)$.*

**Example 2.** *In our running example, consider the bench $\mathcal{B}(\mathrm{A})$. It has the initial state* A*, $level(\mathrm{A}) = 4$, $inner(\mathrm{A}) = \{\mathrm{D}, \mathrm{K}\}$ and $exit(\mathrm{A}) = \{\mathrm{B}, \mathrm{C}, \mathrm{E}\}$. State* F *is not on this bench*

*because it is only reachable from* A *via exit state* E*, and no GBFS tie-breaking exists where* F *can be expanded.*

*Bench* $\mathcal{B}(\mathrm{E})$ *has initial state* E*, level*$(\mathrm{E}) = 1$*, inner*$(\mathrm{E}) = \emptyset$ *and exit*$(\mathrm{E}) = \{\mathrm{O}\}$*. State* L *is not on this bench because its heuristic value is larger than the level of the bench.*

Benches capture the possible local behaviors of GBFS after expanding a given progress state and before expanding the next progress state. In Heusner *et al.* [2017], we show that whenever state $s$ is a progress state that is potentially expanded by GBFS, all states of $\mathcal{B}(s)$ are also potentially expanded by GBFS. Moreover, after expanding state $s$ and before expanding the next progress state $s'$ in the GBFS realization, *only* states in $\mathcal{B}(s)$ can be expanded.

In summary, once a GBFS run reaches a bench, i.e., expands the progress state that forms the initial state of the bench, it remains on the bench until it expands an exit state of this bench (which, in turn, is a progress state and hence begins another bench). Every time a GBFS search moves on to a new bench, it has made progress towards finding a goal, as subsequent benches have strictly decreasing levels. This also means that the search never returns to a bench it has exited.[2]

Therefore, the set of states potentially expanded by GBFS follows from the inner structure of benches (which states are included in which bench) and the connectivity between benches (i.e., which progress states occur as bench exit states of which other progress states). Our next definition formalizes the connectivity between benches.

**Definition 4.** *Let* $\mathcal{T} = \langle \mathcal{S}, h \rangle$ *be a state space topology with initial state* $s_I$*. The* bench transition system $\mathcal{B}(\mathcal{T})$ *of* $\mathcal{T}$ *is a directed graph* $\langle V, E \rangle$ *whose vertices are benches. The vertex set* $V$ *and directed edges* $E$ *are inductively defined as the smallest sets that satisfy the following properties:*

1. $\mathcal{B}(s_I) \in V$
2. *If* $\mathcal{B}(s) \in V$*,* $s' \in$ *exit*$(s)$*, and* $s'$ *is a non-goal state, then* $\mathcal{B}(s') \in V$ *and* $\langle \mathcal{B}(s), \mathcal{B}(s') \rangle \in E$*.*

In words, the bench transition system can be constructed by starting from the bench of the initial state and then iteratively adding all further benches induced by exit states of previously generated benches.

**Example 3.** *The bench transition system of our running example* $\mathcal{T}$ *is depicted in Figure 2. Starting from the initial bench* $\mathcal{B}(\mathrm{A})$*, which is as described in Example 2, it can be constructed by iteratively selecting an exit state* $s$ *that has not been considered yet and by creating* $\mathcal{B}(s)$ *along with a directed edge that links both benches. Note that benches can share states (like* $\mathcal{B}(\mathrm{B})$ *and* $\mathcal{B}(\mathrm{C})$ *and some states are not part of any bench (like state* F *or* L*).*

Bench transition systems are always acyclic because we must have *level*$(s) >$ *level*$(s')$ whenever the bench transition system contains a directed edge from $\mathcal{B}(s)$ to $\mathcal{B}(s')$ [Heusner *et al.*, 2017]. We are now ready to state our main result, which characterizes the set of states potentially expanded by GBFS.

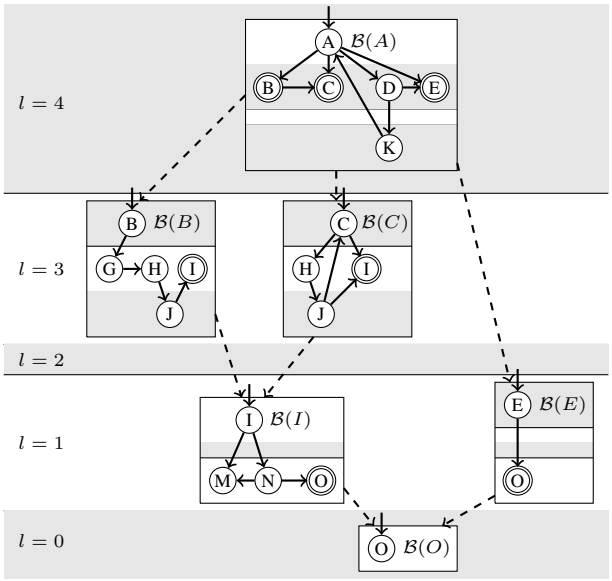


Figure 2: Bench transition system of our running example.

**Theorem 1.** *Let* $\mathcal{T} = \langle \mathcal{S}, h \rangle$ *be a state space topology with bench transition system* $\langle V, E \rangle$*. Then the set of states potentially expanded by GBFS is* $\bigcup_{\mathcal{B}(s) \in V}$ *states*$(s)$*.*

In other words, there exists a GBFS tie-breaking strategy that expands a given state $s$ iff the bench transition system for the given state space topology includes a bench containing $s$.

**Example 4.** *In Example 3, we see that state* F *is not part of any bench in the bench transition system. Even though one of the two shortest* A*-plans is via state* F*, there is no tie-breaking strategy under which GBFS expands* F*.*

In particular, this allows us to compute the set of potentially expanded states in polynomial time in the number $N$ of states of the state space: precompute the high-water mark values by backchaining from the goal states, construct the bench transition system (which consists of at most $N$ benches, each consisting of at most $N$ states), then compute the union of all states of all benches. More efficient algorithms are possible.

## 5 Conclusion

We identified the critical role of *progress states* in GBFS, which are states whose high-water mark exceeds the high-water mark of their successors. Whenever GBFS expands a progress state, all other open states become ineligible for expansion forever: the search effectively behaves as if the open list were cleared with the progress state as a new initial state.

It follows that greedy best-first search is *episodic* in nature, with progress states separating different search episodes. *High-water mark benches* capture the behavior of one episode, and the *bench transition system* captures episode sequencing. Together, they allow us to exactly characterize the states expanded by GBFS under any tie-breaking strategy.

[2] We point out that benches can overlap, i.e., the same state can occur in multiple benches. Therefore, benches do not form a partitioning of the state space. In the presence of *craters*, it is even possible for the same state to be part of different benches that are visited in a single GBFS run. These aspects of benches are not important for the results we present here, and we refer to Heusner *et al.* [2017] for a more detailed discussion.

## Acknowledgments

This work was supported by the European Research Council as part of the project "State Space Exploration: Principles, Algorithms and Applications".

## References

[Asai and Fukunaga, 2017] Masataro Asai and Alex Fukunaga. Tie-breaking strategies for cost-optimal best first search. *Journal of Artificial Intelligence Research*, 58:67–121, 2017.

[Dechter and Pearl, 1985] Rina Dechter and Judea Pearl. Generalized best-first search strategies and the optimality of A*. *Journal of the ACM*, 32(3):505–536, 1985.

[Doran and Michie, 1966] James E. Doran and Donald Michie. Experiments with the graph traverser program. *Proceedings of the Royal Society A*, 294:235–259, 1966.

[Hart *et al.*, 1968] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[Helmert and Röger, 2008] Malte Helmert and Gabriele Röger. How good is almost perfect? In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI 2008)*, pages 944–949. AAAI Press, 2008.

[Heusner *et al.*, 2017] Manuel Heusner, Thomas Keller, and Malte Helmert. Understanding the search behaviour of greedy best-first search. In Alex Fukunaga and Akihiro Kishimoto, editors, *Proceedings of the 10th Annual Symposium on Combinatorial Search (SoCS 2017)*, pages 47–55. AAAI Press, 2017.

[Imai and Kishimoto, 2011] Tatsuya Imai and Akihiro Kishimoto. A novel technique for avoiding plateaus of greedy best-first search in satisficing planning. In Wolfram Burgard and Dan Roth, editors, *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2011)*, pages 985–991. AAAI Press, 2011.

[Korf, 1985] Richard E. Korf. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial Intelligence*, 27(1):97–109, 1985.

[Pohl, 1970] Ira Pohl. Heuristic search viewed as path finding in a graph. *Artificial Intelligence*, 1:193–204, 1970.

[Richter and Helmert, 2009] Silvia Richter and Malte Helmert. Preferred operators and deferred evaluation in satisficing planning. In Alfonso Gerevini, Adele Howe, Amedeo Cesta, and Ioannis Refanidis, editors, *Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling (ICAPS 2009)*, pages 273–280. AAAI Press, 2009.

[Valenzano *et al.*, 2014] Richard Valenzano, Nathan R. Sturtevant, Jonathan Schaeffer, and Fan Xie. A comparison of knowledge-based GBFS enhancements and knowledge-free exploration. In *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling (ICAPS 2014)*, pages 375–379. AAAI Press, 2014.

[Wilt and Ruml, 2014] Christopher Wilt and Wheeler Ruml. Speedy versus greedy search. In Stefan Edelkamp and Roman Barták, editors, *Proceedings of the Seventh Annual Symposium on Combinatorial Search (SoCS 2014)*, pages 184–192. AAAI Press, 2014.

[Wilt and Ruml, 2015] Christopher Wilt and Wheeler Ruml. Building a heuristic for greedy search. In Levi Lelis and Roni Stern, editors, *Proceedings of the Eighth Annual Symposium on Combinatorial Search (SoCS 2015)*, pages 131–139. AAAI Press, 2015.

[Xie *et al.*, 2014a] Fan Xie, Martin Müller, and Robert C. Holte. Adding local exploration to greedy best-first search in satisficing planning. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI 2014)*, pages 2388–2394. AAAI Press, 2014.

[Xie *et al.*, 2014b] Fan Xie, Martin Müller, Robert C. Holte, and Tatsuya Imai. Type-based exploration with multiple search queues for satisficing planning. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI 2014)*, pages 2395–2401. AAAI Press, 2014.