# Local Minima, Heavy Tails, and Search Effort for GBFS

**Eldan Cohen** and **J. Christopher Beck**

Department of Mechanical & Industrial Engineering, University of Toronto, Toronto, Canada

{ecohen, jcb}@mie.utoronto.ca

## Abstract

Problem difficulty for greedy best first search (GBFS) is not entirely understood, though existing work points to deep local minima and poor correlation between the $h$-values and the distance to goal as factors that have significant negative effect on the search effort. In this work, we show that there is a very strong exponential correlation between the depth of the single deepest local minimum encountered in a search and the overall search effort. Furthermore, we find that the distribution of local minima depth changes dramatically based on the constrainedness of problems, suggesting an explanation for the previously observed heavy-tailed behavior in GBFS. In combinatorial search, a similar result led to the use of randomized restarts to escape deep subtrees with no solution and corresponding significant speed-ups. We adapt this method and propose a randomized restarting GBFS variant that improves GBFS performance by escaping deep local minima, and does so even in the presence of other, randomization-based, search enhancements.

## 1 Introduction

In recent years, several works have conducted empirical studies of problem difficulty in greedy best first search (GBFS), highlighting factors that have negative effect on the search effort. One key observation is that the search can get stuck in large local minima ("bad subtrees" [Xie *et al.*, 2014b]): regions of the state space that have no solution, due to an early mistake in the search ("low-$h$ bias" [Richter *et al.*, 2010]). Avoiding such local minima would result in a much lower runtime. Wilt and Ruml [2016] also observed that GBFS performance is negatively effected when the $h$-value difference in a local minimum is high and when it is not possible to reach the goal via a path along which $h$ monotonically decreases.

In a recent study, Cohen and Beck [2018] found that the search effort of GBFS exhibits a heavy-tailed behavior. In combinatorial search, the heavy-tailed behavior was shown to be related to the distribution of depth of subtrees with no solution (*inconsistent subtrees*) [Gomes *et al.*, 2005]. In critically constrained problems, the probability of deep inconsistent subtree is very high, while in relaxed problems there is a

low, but non-negligible probability of a deep inconsistent subtree. These results led to the development of techniques such as randomized restarts and tailored portfolios [Gomes, 2003] that improve search performance by jumping out of deep inconsistent subtrees.

In this work, we show that the analysis of the distribution of inconsistent subtree depths can be applied to GBFS as an explanatory framework for the observed algorithm performance phenomena. We extend Cohen and Beck's framework and introduce the notion of local minimum $h$-depth and $h$-backtracks. We use this framework to study the distribution of local minima $h$-depths in planning problems of different constrainedness levels and its effect on the search effort. We provide explanation and deeper understanding of previous observations on the behavior of GBFS, and show how these insights can be exploited to improve GBFS performance. We make the following contributions:

1. We find an exponential correlation between the $h$-depth of the single deepest local minimum encountered and the total search effort (i.e., number of expanded nodes).

2. We find an exponential correlation between the $h$-depth of local minima and the number of $h$-backtracks.

3. We show that the distribution of local minima $h$-depth in planning problems depends on the constrainedness of problems, and that heavy-tailed behavior appears when there is a low, but non-negligible, probability of encountering a deep local minimum during search.

4. Inspired by combinatorial search, we propose *RR-GBFS*, a randomized restarting GBFS that outperforms GBFS by escaping deep local minima.

## 2 Background

### 2.1 Local Minima, Backtracking, and GBFS

A recent line of work in heuristic search has focused on empirical characterizations of problem difficulty and highlighted different factors affecting the search effort. Wilt and Ruml [2016] analyzed the behavior of GBFS and made three observations. First, that GBFS "tends to work well when it is possible to reach the goal from every node via a path where $h$ monotonically decreases along the path"[1] (*Observation 1*, p.

---

[1] More precisely, the authors actually refer to the case where $h$ is monotonically non-increasing.

283) and that a heuristic with no local minima is an extreme example of such heuristic. They also observe that GBFS "tends to work well when the difference between the minimum $h$ value of the nodes in a local minimum and the minimum $h$ that will allow the search to escape from the local minimum and reach a goal is low" (*Observation 3*, p. 288).[2]

In a previous work, Wilt and Ruml [2014] showed that the negative effect of operator cost ratio in satisficing search is due to the deepening of the local minima. Cohen and Beck [2017] found that the effect of operator cost ratio depends on the constrainedness of problems and therefore conjectured that there is a connection between the existence and extent of local minima and the constrainedness of problems.

## 2.2 Heavy-Tailed Behavior and GBFS

The study of runtime distributions of several computational problems (notably SAT and CSP) found *exceptionally hard instances*, that were attributed to a fat- or heavy-tailed behavior in ensembles of random problems [Gomes *et al.*, 2005]. In ensembles of critically constrained instances, for which the median search effort is very high and all instances are uniformly hard, heavy-tailed behavior does not appear. However, relaxing the problems results in statistical regime in which the median effort is low, but the hardest problems are very hard. The distribution in this regime can be modeled by a fat- or heavy-tailed behavior, indicated by a slow decay of the tail of the survival function and high kurtosis [Gomes, 2003]. Formally, a random variable $X$ with distribution $F(x)$ is considered heavy-tailed if there exists some $x_l > 0$, $c > 0$, $\alpha > 0$ such that $1 - F(x) = P[X > x] = cx^{-\alpha}, x > x_l$ [Resnick, 2007].

Later work found that the heavy-tailed behavior is also observed in the runtime distribution of multiple runs of a randomized search procedure on *one* instance and that the exceptionally hard problems are easily solved by minor changes in the search [Gomes *et al.*, 2005]. This result inspired new algorithms such as randomized restarts and portfolios that aim at eliminating the heavy-tailed behavior [Gomes, 2003].

For GBFS, Cohen and Beck [2018] found heavy-tailed behavior in both ensembles of random planning problems and multiple runs of a randomized heuristic search on one instance. In order to randomize the heuristic search, they introduced a limited amount of randomization in the heuristic. Given a heuristic $h$ and a parameter $p \geq 0$, they defined $h_{\Delta_p}$ to be a $p$-randomized version of $h$ as: $h_{\Delta_p}(s) = h(s) + \Delta_p^h$ where $\Delta_p^h$ is a random number in the range $[-p \cdot h(s), p \cdot h(s)]$.

To illustrate heavy-tailed behavior, we present an example from Cohen and Beck [2018]. Figure 1 shows the survival function for the search effort over 1000 runs of a randomized search on a single instance of the Rovers domain for different levels of constrainedness. When the problem is highly constrained ($C=1$), there is no apparent heavy-tailed behavior. As the problem is relaxed we see a fat and heavy-tailed behavior, as most instances become easy while the hardest problems are very hard. For $C=6$, a clear heavy tailed behavior is indicated by approximately linear behavior on the log-log plot [Gomes *et al.*, 2005].

---

[2]Wilt and Ruml's [2016] Observation 2 is implied by Observation 1 and so omitted.
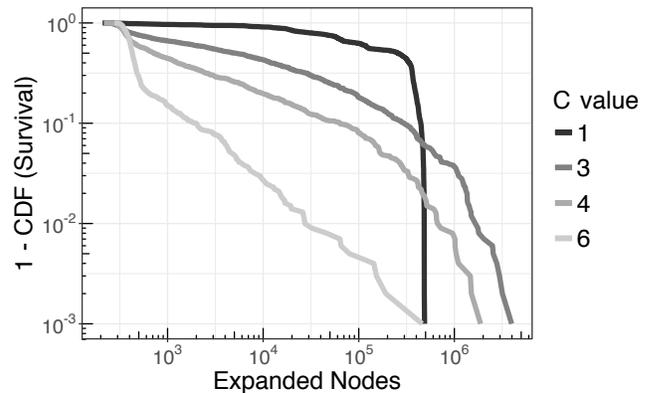


Figure 1: Rovers: Multiple runs on one instance.

## 3 Preliminaries

Following Bonet and Geffner [2001], a planning problem $\mathcal{P} = \langle \mathcal{A}, \mathcal{O}, \mathcal{I}, \mathcal{G} \rangle$ defines a state space $\mathcal{S}_\mathcal{P} = \langle S, s_I, S_G, A, f, c \rangle$ where the states $s \in S$ are collection of atoms in $\mathcal{A}$, $s_I$ is the initial state $\mathcal{I}$, goal states $s \in S_G$ are all states such that $\mathcal{G} \subseteq s$, actions $a \in A(s)$ are the operators in $\mathcal{O}$ such that the preconditions $Prec(o) \subseteq s$, the function $f(s, a)$ maps a state $s$ and an action $a$ into a state $s'$ based on the delete and add effects $s' = (s \setminus Del(a)) \cup Add(a)$, and all action costs $c(a, s) = 1$.

We use the state representation to formally define a set of terms used in our analysis. Informally, we consider a local minimum to be a set of states that were expanded between two consecutive expanded states on the solution path found by a search algorithm (we assume nodes are not re-opened).

**Definition 1.** (Solution vector) *Let $\mathcal{P} = \langle s_1, ..., s_P \rangle$ be a vector of states in $S$. We consider $\mathcal{P}$ to be a (feasible) solution vector if $s_1 = S_I$, $s_P \in S_G$ and for each consecutive pair of states $s_i, s_j$ in $\mathcal{P}$ there exists action $a$ such that $f(s_i, a) = s_j$.*

**Definition 2.** (Expansion vector) *Let $\mathcal{E} = \langle s_1, ..., s_E \rangle$ be a vector of states in $S$. We consider $\mathcal{E}$ to be a (feasible) expansion vector if $s_1 = S_I$, $s_E \in S_G$ and for every state $s_x \in \mathcal{E}, x \geq 2$ there exists state $s_y \in \mathcal{E}, y < x$ and action $a \in A$ such that $f(s_y, a) = s_x$.*

**Definition 3.** (Local Minimum) *Let $\mathcal{E}$ be an expansion vector and $\mathcal{P}$ be a solution vector, such that $\mathcal{P}$ is a non-contiguous subsequence of $\mathcal{E}$. For every consecutive pair $s_i, s_j$ in $\mathcal{P}$ we define the local minimum $\mathcal{L}_{s_i, s_j}$ to be the sub-vector of all states between $s_i$ and $s_j$ in the vector $\mathcal{E}$ (including $s_i, s_j$).*

**Definition 4.** ($h$-depth) *Let $s_i, s_j$ be two consecutive states in a solution vector and $\mathcal{L}_{s_i, s_j} = \langle s_1, s_2...s_L \rangle$, a vector of states, be the local minimum between $s_i$ and $s_j$. We define the $h$-depth of the local minimum $d_{s_i, s_j}^h = h(s_j) - min_h(\mathcal{L}_{s_i, s_j})$.*

**Definition 5.** ($h$-backtrack) *Let $s_i, s_{i+1}$ be two consecutive states in an expansion vector. We define the expansion of $s_{i+1}$ to be an $h$-backtrack if $h(s_{i+1}) > h(s_i)$.*

Note that the definition of local minimum $h$-depth closely matches the $h$ difference in Observation 1 and the definition of $h$-backtrack is closely related to the non-monotonicity of $h$-values in Observation 3.

## 4 Empirical Analysis

In this section we present an empirical analysis of the local minima phenomenon in GBFS. We present results for the FF heuristic [Hoffmann and Nebel, 2001] with deferred ("lazy") heuristic evaluation [Richter and Helmert, 2009], however experiments with standard ("eager") evaluation and other heuristics (causal graphs [Helmert, 2004], landmark count [Richter *et al.*, 2008], landmark cut [Helmert and Domshlak, 2009]) yielded similar trends. We use Fast Downward [Helmert, 2006], configured not to re-open nodes.

In our analysis, we use the six benchmark domains in Cohen and Beck [2018], for which the constrainedness of problems can be controlled either by resource-constrainedness parameter (denoted $C$) or by goal-constrainedness parameter (denoted $\lambda$): NoMystery, Rovers, TPP, Maintenance, Parking, and Freecell. We present a thorough analysis of one domain (NoMystery) and summarize the results for the rest. For each domain, we present results for both an ensemble of 1000 random problems and for 1000 runs of a $p$-randomized search on a single problem ($0.05 \leq p \leq 0.1$). Following Cohen and Beck [2018], the single problem is the median instance of the ensemble $C=1$ and the relaxed instances are generated by increasing $C$ (analysis of starting problems from quantiles other than the median yields similar trends).

### 4.1 Problem Difficulty and Local Minima

We start by analyzing the connection between the $h$-depth and the size (i.e., number of expanded nodes) of a local minimum for different constrainedness levels. Figure 2a shows the distribution of local minima size vs. $h$-depth in ensemble of 1000 random problems in both the non-heavy-tailed regime ($C=1$) and the heavy-tailed regime ($C=2$). We can clearly see an exponential correlation, similar to inconsistent subtrees [Gomes *et al.*, 2005]. Figure 2b shows similar results for 1000 runs of a randomized search on one instance. While the absolute numbers change between different levels of constrainedness, the qualitative trends remains similar for different $C$ values.

In combinatorial search, this result is due to backtracking behavior. Figure 2c shows the distribution of number of $h$-backtracks vs. $h$-depth in an ensemble of random problems in both the non-heavy-tailed and the heavy-tailed regimes. Again, we find an exponential correlation suggesting that the observed effort is due to a backtracking behavior based on $h$-values. The fact that we would observe such a correlation for GBFS is not a priori obvious given its best-first search style.

Surprisingly, we also find a strong correlation between the $h$-depth of the *single* deepest local minimum encountered in an instance and the *total* search effort for that instance. Table 1 reports this correlation for all the six benchmark domains, in both the heavy-tailed and the non-heavy-tailed regimes and over all problem instances (of mixed constrainedness values). The reported values are the Pearson correlation coefficient between the $h$-depth of the deepest local minimum and the $log$(search effort). We use $log$ since Pearson measures linear correlation, and linear correlation between $x$ and $log(y)$ indicates an exponential correlation between $x$ and $y$. To make sure we are not biased by a population that is centered on a small range of $h$-values we also calculated the weighted Pearson correlation (all the instances of a given $h$-depth sum up



(a) 1000 random instances: local minima size vs. $h$-depth.



(b) 1000 runs of randomized search: local minima size vs. $h$-depth.



(c) 1000 random instances: number of $h$-backtracks vs. $h$-depth.
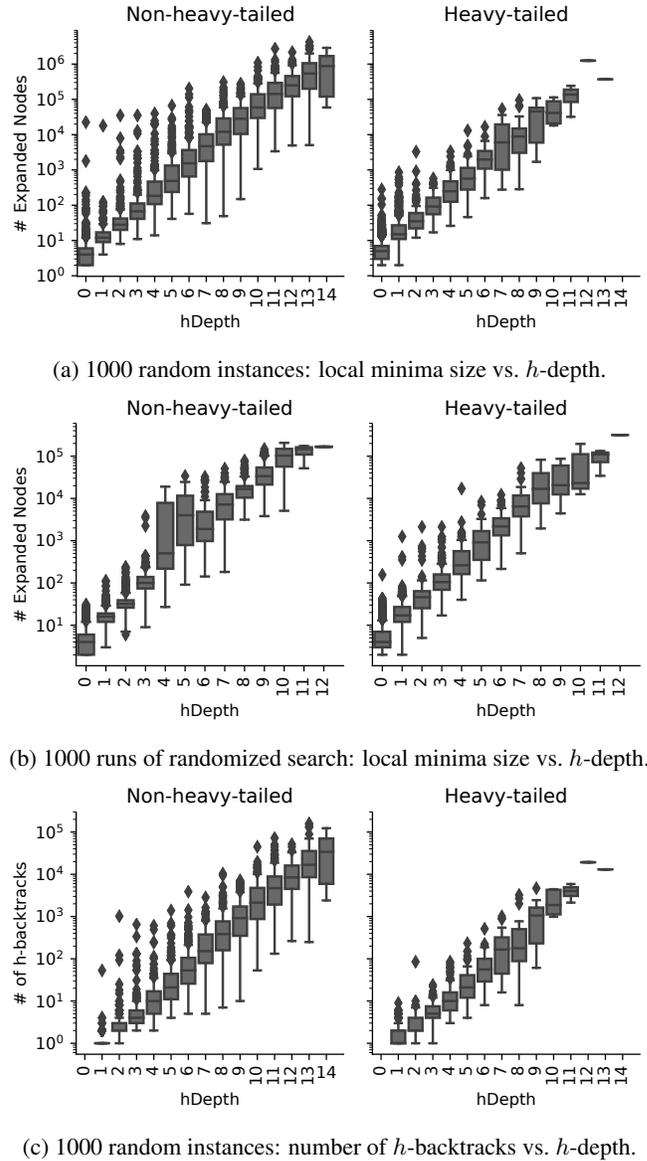
Figure 2: Empirical results for NoMystery.

to the same weight) and found similarly high values. These results indicate a strong exponential correlation and suggest that the deepest local minimum encountered in a search is an important factor in determining the total search effort.

Given this strong correlation, a natural hypothesis, that is consistent with the results for CSP and SAT, is that the heavy-tailed behavior in planning problems is due to the distribution of $h$-depth of local minima in such problems (i.e., there is a low probability of getting into a deep local minimum that leads to few, very hard, instances). In the next section, we will analyze this distribution and test this hypothesis.

### 4.2 The Distribution of Local Minima $h$-Depth

We now analyze the distribution of $h$-depth of the deepest local minimum encountered in each problem for ensembles of

| Domain | $log$(effort) | | $log$($h$-backtracks) | |
|---|---|---|---|---|
| | N-HT | HT | N-HT | HT |
| NoMyst (1000) | 0.92 | 0.92 | 0.93 | 0.95 |
| NoMyst (one) | 0.87 | 0.94 | 0.97 | 0.96 |
| Rovers (1000) | 0.96 | 0.84 | 0.98 | 0.94 |
| Rovers (one) | 0.98 | 0.95 | 0.98 | 0.98 |
| TPP (1000) | 0.89 | 0.86 | 0.92 | 0.95 |
| TPP (one) | 0.80 | 0.77 | 0.84 | 0.93 |
| Maint (1000) | 0.93 | 0.95 | 0.96 | 0.98 |
| Maint (one) | 0.93 | 0.98 | 0.93 | 0.99 |
| Parking (1000) | 0.97 | 0.93 | 0.96 | 0.96 |
| Parking (one) | 0.97 | 0.92 | 0.97 | 0.90 |
| Freecell (1000) | 0.79 | 0.83 | 0.82 | 0.87 |
| Freecell (one) | 0.73 | 0.82 | 0.75 | 0.83 |

Table 1: Pearson correlation coefficient between $h$-depth and $log$(Search effort) ($log$($h$-backtracks)) in the non-heavy-tailed (N-HT) and heavy-tailed (HT) regimes for ensembles of random problems and for multiple runs on one problem.

| Domain | N-HT | | HT | |
|---|---|---|---|---|
| | Med | Max | Med | Max |
| NoMyst (1000) | 9 | 14 | 0 | 13 |
| NoMyst (one) | 10 | 12 | 3 | 12 |
| Rovers (1000) | 12 | 25 | 0 | 24 |
| Rovers (one) | 21 | 22 | 7 | 23 |
| TPP (1000) | 9 | 13 | 0 | 11 |
| TPP (one) | 11 | 12 | 1 | 9 |
| Maint (1000) | 10 | 16 | 0 | 11 |
| Maint (one) | 12 | 16 | 0 | 11 |
| Parking (1000) | 4 | 19 | 1 | 14 |
| Parking (one) | 9 | 13 | 2 | 13 |
| Freecell (1000) | 14 | 35 | 8 | 32 |
| Freecell (one) | 22 | 33 | 7 | 27 |

Table 2: Median and maximal value of deepest local minimum encountered in instances in the non-heavy-tailed (N-HT) and the heavy-tailed (HT) regimes for both ensembles of 1000 random problems and for multiple runs on one problem.
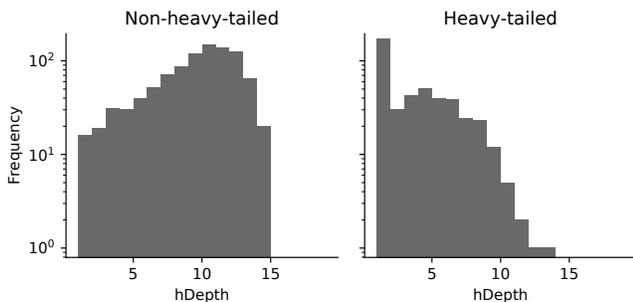


(a) 1000 random instances.
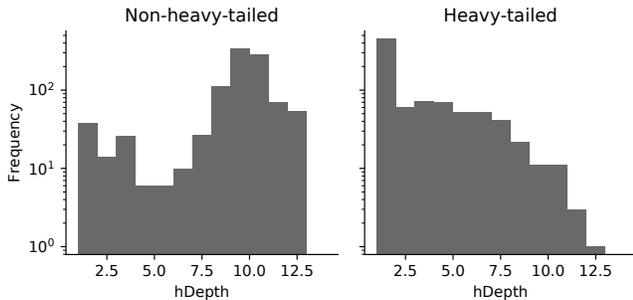


(b) 1000 runs of randomized search.

Figure 3: NoMystery: Local minima size vs. $h$-depth.

random problems of different constrainedness levels. Figure 3a shows the distribution in the non-heavy-tailed ($C$=1) and in the heavy-tailed regime ($C$=2). In the non-heavy-tailed regime, we find that GBFS on the majority of the problems encounters a deep local minimum (the peak of the histogram is at 10). In the heavy-tailed regime, where most problems are easy but there are few *exceptionally* hard problems, we find that the distribution changes dramatically. The majority of the searches do not encounter a deep local minimum at all (the peak of the histogram is at 1), however a few instances do

encounter local minima that are almost as deep as the deepest local minima in the non-heavy-tailed regime.

More interesting is the corresponding distribution for multiple runs of a randomized search procedure on one problem instance. Figure 3b shows the results when using a randomized search procedure on the same instance for different values of $C$. We find similar trends: in the non-heavy-tailed regime ($C$=1) we find that the majority of runs encounter a deep local minimum. As we relax the problem and move to the heavy-tailed regime and easier problems on average, the majority of runs do not encounter a deep local minimum, however a few encounter a very deep local minimum.

To summarize similar results for the different domains, Table 2 reports the median and the maximal value for the deepest local minimum encountered in an instance. As demonstrated for NoMystery, in the non-heavy-tailed regime the median and the maximal value are much closer, compared to the heavy-tailed regime where the median is very low but there are few very deep local minima. In many cases the maximal value for the heavy-tailed regime is very close and in Rovers (one) even higher than the one of the non-heavy-tailed regime, while the median is significantly lower.

### 4.3 Discussion

Our analysis suggests that the $h$-depth of the local minima encountered in the search is a key factor in problem difficulty for different kinds of planning problems (we analyzed six domains with different characteristics and constrainedness type). Furthermore, we find an exponential correlation between the depth of a local minimum and its size, as often observed in tree search. These results are consistent with combinatorial search problems (e.g., in [Gomes *et al.*, 2005]) and further establish the connection between satisficing planning and a large body of literature on combinatorial search.

These results provide explanation and deeper understanding of several previously observed phenomena in GBFS:

1. Our results explain the heavy-tailed behavior observed in satisficing planning using GBFS [Cohen and Beck,

2018]. In relaxed problems, there is a low, but non-negligible, probability of encountering a deep local minimum (consistent with combinatorial search).

2. The exponential correlation between $h$-depth and search effort explains Observations 1 on the effect of large $h$ difference in a local minimum [Wilt and Ruml, 2016].

3. The exponential correlation between $h$-depth and $h$-backtracks suggest Observation 3 is connected to Observation 1. Extensive backtracking behavior is the result of a deep local minimum [Wilt and Ruml, 2016].

4. The $h$-depth distribution suggests that the existence and extent of the factors highlighted by Wilt and Ruml (deep local minima and extensive backtracking behavior) depend on the constrainedness of problems. This work establishes the conjecture by Cohen and Beck [2017] that there is a connection between the existence and extent of local minima and the constrainedness of problems.

The heavy-tailed behavior observed in the investigated domains stems primarily from local minima, one type of uninformative heuristic region [Xie *et al.*, 2014a]. A second type, plateaus, are a different kind of search inefficiency that is not addressed in this work. The effect of constrainedness on domains with significant plateaus and no local minima in GBFS is an interesting direction for future work.

## 5 Exploiting the Heavy-tailed Behavior: Randomized Restarting GBFS

Following the work on randomized restarts in combinatorial search [Gomes *et al.*, 1998], in this section we demonstrate how such randomized restarts can be integrated into a GBFS to gain significant speed-ups.

A restart strategy is a sequence $(t_1, t_2, ...)$ of cutoff values, i.e., run lengths (often expressing number of backtracks) after which the search restarts. As we have established the connection between the $h$-backtracks and the heavy-tailed behavior, we use cutoff values based on $h$-backtracks. Algorithm 1 presents pseudocode for a randomized restarting GBFS (RR-GBFS). In each iteration, we run a GBFS with a given $h$-backtrack cutoff. If the cutoff is reached before a solution is found, the randomized search is restarted with a different seed and the next cutoff value. We use randomized heuristic search with a geometric restart policy [Walsh, 1999] with an initial value of 16, increasing with a factor of 1.5: $(16, 24, 36, ...)$.

Figure 4a and 4b compare the runtime distribution of NoMystery for multiple runs on one instance and for an ensemble of random instances respectively, using randomized restarting GBFS vs. standard GBFS. The randomized-restarting GBFS manages to significantly reduce the tail in the fat- and heavy-tailed regimes, outperforming GBFS when $C > 1$. However, for the most constrained problems ($C = 1$), where the probability of a deep local minimum is very high and, therefore, restarting is likely to lead to another deep local minimum, RR-GBFS underperforms GBFS, as expected.

Table 3 reports the maximum local minimum $h$-depth encountered for GBFS and for the last iteration of RR-GBFS in both the heavy-tailed and non-heavy-tailed regimes. It also reports the maximal number of restarts needed by RR-GBFS.

**Algorithm 1** Randomized restarting GBFS

```
function RR-GBFS (seed, cutoff)
    while GBFS(seed, cutoff) = NO_SOLUTION do
        seed ← ChooseRandomSeed()
        cutoff ← updateCutoff()
```

These results explain the success of RR-GBFS for the less constrained problems. Exploiting the distribution of local minima $h$-depth in the heavy-tailed regime, RR-GBFS needs fewer restarts to successfully escape deep local minima in most of the domains, reducing the maximum $h$-depth significantly compared to GBFS. In the non-heavy-tailed regime, the number of restarts is much higher and the maximum $h$-depth of RR-GBFS is much closer to GBFS.

Cohen and Beck [2018] previously noted that $\epsilon$-GBFS [Valenzano *et al.*, 2014] and Type-GBFS [Xie *et al.*, 2014b], GBFS variants that incorporate random exploration in the search, reduce heavy-tailed behavior while increasing the effort for median problems. A comparison between RR-GBFS, $\epsilon$-GBFS, and Type-GBFS found that while they all tend to reduce the heavy-tailed behavior, their performance differs based on the domain, the constrainedness of problems, and the effect on the easy problems. In general we find that random exploration tends to work better for the most constrained problems (as RR-GBFS applies unneeded restarts) and RR-GBFS tends to perform better for more relaxed problems (as $\epsilon$-GBFS and Type-GBFS apply unneeded exploration).
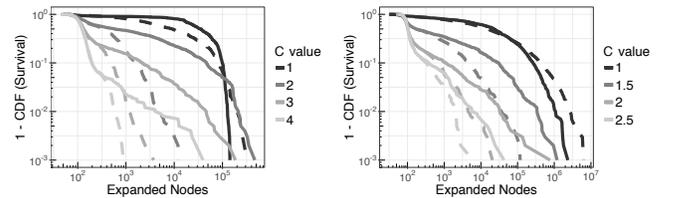


(a) Multiple runs on one instance.    (b) 1000 random instances.

Figure 4: NoMystery: GBFS (solid) vs RR-GBFS (dashed).

| Domain | Non-Heavy-tailed | | | Heavy-tailed | | |
|---|---|---|---|---|---|---|
| | $D_G$ | $D_{RR}$ | #R | $D_G$ | $D_{RR}$ | #R |
| NoMyst (1000) | 14 | 15 | 27 | 13 | 6 | 10 |
| NoMyst (one) | 12 | 11 | 18 | 12 | 5 | 5 |
| Rovers (1000) | 25 | 21 | 28 | 24 | 13 | 17 |
| Rovers (one) | 22 | 18 | 23 | 23 | 9 | 9 |
| TPP (1000) | 13 | 11 | 23 | 13 | 6 | 6 |
| TPP (one) | 12 | 12 | 15 | 9 | 6 | 3 |
| Maint (1000) | 16 | 15 | 19 | 11 | 7 | 14 |
| Maint (one) | 16 | 15 | 17 | 11 | 5 | 7 |
| Parking (1000) | 19 | 18 | 23 | 12 | 11 | 16 |
| Parking (one) | 13 | 10 | 12 | 13 | 6 | 4 |
| Freecell (1000) | 35 | 36 | 24 | 32 | 25 | 21 |
| Freecell (one) | 33 | 32 | 17 | 27 | 19 | 10 |

Table 3: The maximum $h$-depth for GBFS ($D_G$) and for RR-GBFS ($D_{RR}$) and the maximal number of restarts in RR-GBFS (#R).

(a) NoMystery: GBFS (solid), $\epsilon$-GBFS ($\epsilon$=.5, dotted), $\epsilon$-RR-GBFS (dashed).

(b) Maintenance: GBFS (solid), $\epsilon$-GBFS ($\epsilon$=.2, dotted), $\epsilon$-RR-GBFS (dashed).

(c) TPP: GBFS (solid), Type-GBFS (dotted), Type-RR-GBFS (dashed).
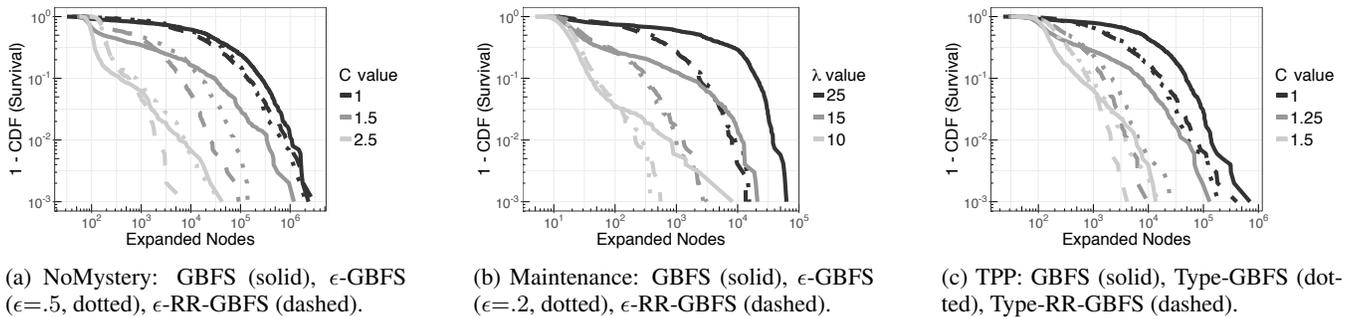
Figure 5: Comparison between RR-GBFS to Type-GBFS and $\epsilon$-GBFS on ensembles of 1000 random problems.

We can easily combine randomized restarts with random exploration (by applying both techniques) as shown in Figure 5 for NoMystery, TPP, and Maintenance. The combined versions seem to be as good or better than either of the methods alone. A detailed investigation of the differences between random exploration and randomized restarts is an interesting direction for future work.

Our results support the hypothesis that randomized restarts exploit the distribution of local minima $h$-depth to escape deep local minima in the heavy-tailed regime and show that randomized restarting can improve performance of GBFS and of GBFS variants that incorporate random exploration. In combinatorial search, much work has been done on randomized restarts, including the use of dynamic and learning restart policies (e.g., [Kautz *et al.*, 2002]) that are updated in real-time. Investigating ways to incorporate such policies in GBFS is an interesting direction for future work.

We considered two common exploration methods: $\epsilon$-GBFS and Type-GBFS. Other methods that incorporate exploration in GBFS include GBFS-LS [Xie *et al.*, 2014a], DBFS [Imai and Kishimoto, 2011], and IP-diversification [Asai and Fukunaga, 2017]. Analyzing the effect of restarts in the presence of these exploration methods is potential future work.

## 6 Related Work

Several works have dealt with the state space topology of satisficing planning. Hoffmann [2005] discussed the local search topology of heuristic search and analyzed plateaus and local minima (defined differently) for the idealized $h^+$. Heusner et al. [2017] used high-water mark benches which separate the search space into areas that are searched by a GBFS in sequence, allowing to determine nodes that will not be expanded and nodes that will be expanded only if certain conditions are met. Using the theory developed in this work to derive theoretical bounds on the size of local minima is a potential future work. Wilt and Ruml [2014] measured the effect of cost-based GBFS on the *size* of local minima (number of nodes) in different domains using different heuristics. In a later work, they provide several observations (see Section 2.1) on the behavior of GBFS [Wilt and Ruml, 2016].

Several works suggested using restarts in different planning algorithms and scenarios. Nakhost et al. [2009] used restarts in Monte-Carlo Random Walk planning after reaching a dead end or a maximum number of consecutive non-

improving steps. Coles et al. [2007] proposed a local search planning algorithm together with global restarts. Richter et al. [2010] used restarts to improve solution quality in an anytime weighted A* by restarting the search each time a new solution is found, while keeping track of previously seen nodes. While these algorithms show the value of restarts, none do so in the context of GBFS nor do they provide a precise empirical understanding of how randomized restarting can lead to stronger search performance as we have done here.

## 7 Conclusion

We performed an empirical study of local minima in ensembles of random planning problems and multiple runs of a randomized heuristic search on a single planning problem instance. We defined the depth of local minima based on the $h$-value differences and analyzed the connection between $h$-depth and search effort and the distribution of local minima $h$-depth in problems of different constrainedness.

Our empirical results indicate a strong exponential correlation between the $h$-depth of a local minimum and the associated search effort and a similar strong correlation between the $h$-depth and the number of $h$-backtracks. Furthermore, we found that the probability of entering a deep local minimum depends on the constrainedness of the problems.

Inspired by combinatorial optimization, we proposed a randomized restarting GBFS variant and show that it successfully escapes deep local minima in the heavy-tailed regime, resulting in better search performance.

These results provide a deeper understanding of GBFS and its search topology and explain several previous observations and conjectures on the behavior of GBFS. We have demonstrated a simple way that these insights can be used to enhance search performance and believe that they can be further exploited to develop additional improvements to heuristic search algorithms.

## Acknowledgments

## References

[Asai and Fukunaga, 2017] Masataro Asai and Alex S. Fukunaga. Exploration among and within plateaus in

greedy best-first search. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 2017.

[Bonet and Geffner, 2001] Blai Bonet and Hector Geffner. Planning as heuristic search. *Artificial Intelligence*, 129(1):5–33, 2001.

[Cohen and Beck, 2017] Eldan Cohen and J Christopher Beck. Cost-based heuristics and node re-expansions across the phase transition. In *Symposium on Combinatorial Search (SoCS)*, pages 11–19, 2017.

[Cohen and Beck, 2018] Eldan Cohen and J Christopher Beck. Fat- and heavy-tailed behavior in satisficing planning. In *AAAI Conference on Artificial Intelligence (AAAI)*, page In press, 2018.

[Coles *et al.*, 2007] Andrew Coles, Maria Fox, and Amanda Smith. A new local-search algorithm for forward-chaining planning. In *International Conference on Automated Planning and Scheduling (ICAPS)*, pages 89–96, 2007.

[Gomes *et al.*, 1998] Carla P Gomes, Bart Selman, and Henry Kautz. Boosting combinatorial search through randomization. *National Conference on Artificial Intelligence (AAAI)*, pages 431–437, 1998.

[Gomes *et al.*, 2005] Carla P Gomes, Cèsar Fernández, Bart Selman, and Christian Bessière. Statistical regimes across constrainedness regions. *Constraints*, 10(4):317–337, 2005.

[Gomes, 2003] Carla P Gomes. Randomized backtrack search. In Michela Milano, editor, *Constraint and integer programming: Toward a unified methodology*, pages 233–291. Springer Science & Business Media, 2003.

[Helmert and Domshlak, 2009] Malte Helmert and Carmel Domshlak. Landmarks, critical paths and abstractions: what's the difference anyway? In *International Conference on Automated Planning and Scheduling (ICAPS)*, pages 162–169, 2009.

[Helmert, 2004] Malte Helmert. A planning heuristic based on causal graph analysis. In *International Conference on Automated Planning and Scheduling (ICAPS)*, pages 161–170, 2004.

[Helmert, 2006] Malte Helmert. The fast downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, 2006.

[Heusner *et al.*, 2017] Manuel Heusner, Thomas Keller, and Malte Helmert. Understanding the search behaviour of greedy best-first search. In *Symposium on Combinatorial Search (SoCS)*, 2017.

[Hoffmann and Nebel, 2001] Jörg Hoffmann and Bernhard Nebel. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14:253–302, 2001.

[Hoffmann, 2005] Jörg Hoffmann. Where 'ignoring delete lists' works: local search topology in planning benchmarks. *Journal of Artificial Intelligence Research*, 24:685–758, 2005.

[Imai and Kishimoto, 2011] Tatsuya Imai and Akihiro Kishimoto. A novel technique for avoiding plateaus of greedy best-first search in satisficing planning. In *Symposium on Combinatorial Search (SoCS)*, 2011.

[Kautz *et al.*, 2002] Henry Kautz, Eric Horvitz, Yongshao Ruan, Carla Gomes, and Bart Selman. Dynamic restart policies. *National Conference on Artificial Intelligence (AAAI)*, pages 674–681, 2002.

[Nakhost and Müller, 2009] Hootan Nakhost and Martin Müller. Monte-carlo exploration for deterministic planning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 9, pages 1766–1771, 2009.

[Resnick, 2007] Sidney I Resnick. *Heavy-tail phenomena: probabilistic and statistical modeling*. Springer Science & Business Media, 2007.

[Richter and Helmert, 2009] Silvia Richter and Malte Helmert. Preferred operators and deferred evaluation in satisficing planning. In *International Conference on Automated Planning and Scheduling (ICAPS)*, pages 273–280, 2009.

[Richter *et al.*, 2008] Silvia Richter, Malte Helmert, and Matthias Westphal. Landmarks revisited. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 975–982, 2008.

[Richter *et al.*, 2010] Silvia Richter, Jordan Tyler Thayer, and Wheeler Ruml. The joy of forgetting: Faster anytime search via restarting. In *International Conference on Automated Planning and Scheduling (ICAPS)*, pages 137–144, 2010.

[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 *International Conference on Automated Planning and Scheduling (ICAPS)*, pages 375–379, 2014.

[Walsh, 1999] Toby Walsh. Search in a small world. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1172–1177, 1999.

[Wilt and Ruml, 2014] Christopher Wilt and Wheeler Ruml. Speedy versus greedy search. In *Symposium on Combinatorial Search (SoCS)*, pages 184–192, 2014.

[Wilt and Ruml, 2016] Christopher Wilt and Wheeler Ruml. Effective heuristics for suboptimal best-first search. *Journal of Artificial Intelligence Research*, 57:273–306, 2016.

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

[Xie *et al.*, 2014b] Fan Xie, Martin Müller, Robert Holte, and Tatsuya Imai. Type-based exploration with multiple search queues for satisficing planning. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 2395–2402, 2014.