Problem Difficulty and the Phase Transition in Heuristic Search

Eldan Cohen and **J. Christopher Beck**
Department of Mechanical & Industrial Engineering
University of Toronto
Toronto, Canada
{ecohen, jcb}@mie.utoronto.ca

Abstract

In the recent years, there has been significant work on the difficulty of heuristic search problems, identifying different problem instance characteristics that can have a significant impact on search effort. Phase transitions in the solubility of random problem instances have proved useful in the study of problem difficulty for other classes of computational problems, notably SAT and CSP, and it has been shown that the hardest problems typically occur during this rapid transition. In this work, we perform the first empirical investigation of the phase transition phenomena for heuristic search. We establish the existence of a rapid transition in the solubility of an abstract model of heuristic search problems and show that, for greedy best first search, the hardest instances are associated with the phase transition region. We then perform a novel investigation of the behavior of heuristics of different strength across the solubility spectrum. Finally, we demonstrate that the behavior of our abstract model carries over to commonly used benchmark problems including the Pancake Problem, Grid Navigation, TopSpin, and the Towers of Hanoi. An interesting deviation is observed and explained in the Sliding Puzzle.

1 Introduction

A recent line of research in heuristic search aims to develop of an understanding of empirical problem difficulty. Several factors affecting search effort have been identified including the ratio of operator costs (Wilt and Ruml 2011), the correlation between path cost and length (Wilt and Ruml 2012; 2015) and the existence of uninformed heuristic regions (Xie, Müller, and Holte 2015).

The phase transition in problem solubility has been a central tool in the study of problem difficulty for computational problems. In seminal work, Cheeseman, Kanefsky and Taylor (1991) empirically showed that several NP-complete problems exhibit an abrupt *phase transition* from underconstrained to overconstrained problems as a problem-generation control parameter is varied, changing the probability of a solution from nearly zero to nearly one. They discovered that the hardest problem instances occur during this abrupt change.

Subsequently, the phase transition was extensively studied in problems including SAT (Mitchell, Selman, and Levesque 1992; Crawford and Auton 1996), CSP (Smith and Dyer 1996; Prosser 1996), quantified boolean formula (Gent and Walsh 1999), and classical planning (Bylander 1996; Rintanen 2004). Interestingly, despite Cheeseman et al.'s conjecture that the phase transition was relevant to a number of AI problems, it does not appear to have been studied for heuristic search problems.

In this work, we introduce the tool of phase transition to heuristic search using an abstract model of a heuristic search problem that is based on a random graph representation of the state space. We demonstrate an abrupt transition in solubility as a parameter controlling the density of the transitions in the state space is varied and observe the accompanying easy-hard-easy pattern of problem difficulty across the transition region. Building on these results, we make the following further contributions:

- Exploiting our random graph model, we provide analytical bounds on the "mushy region" between the fully soluble and fully insoluble problems.
- We demonstrate how to transfer the abstract graph model to existing heuristic search benchmark problems, allowing the generation of versions of each problem across the phase transition and demonstrating both the phase transition and the easy-hard-easy pattern on the five standard heuristic search benchmarks.
- We study the behavior of systematically stronger heuristics across the phase transition region and show that the reduction in search effort for strong heuristics is orders of magnitude smaller for the hard soluble instances at the phase transition than in the underconstrained regions.

2 Background

For many NP-complete problems, we can define a *control parameter* for which a critical interval of values separates two regions: one that is underconstrained with high density of solutions and one that is overconstrained with low likelihood that a solution exists (Cheeseman, Kanefsky, and Taylor 1991). Empirically, the hardest problems occur over this critical interval. Mitchell, Selman and Levesque (1992) identified a phase transition for critical values of the clause-to-variable ratio of 3-SAT problems, with search effort peaking at a ratio of approximately 4.27. Smith and Dyer (1996) investigated phase transitions in CSPs and showed a phase transition in problem solubility for critical values of constraint tightness. The hardest problems occurred during this rapid transition with the median search effort peaking at the point in which 50% of the problems are soluble. Many other works through the late 1990s confirmed these results and extended them to other classes of problems.

For PSPACE-complete problems, Gent and Walsh (1999) investigated the phase transition phenomenon on quantified boolean formulae. They witnessed a clear phase transition in solubility and an easy-hard-easy pattern associated with a critical value of constrainedness. They therefore conjectured that similar phase transition behavior will occur in other PSPACE problems. In planning, Rintanen (2004) found a transition in solubility as the operators-to-variables ratio is varied. However, the effort peak was located earlier than expected and the analytical derivation of tight upper and lower bounds for this transition remained an open problem.

The phase transition in heuristic search does not appear to have been investigated. While a reference to heuristic search occurs in an early work (Huberman and Hogg 1987), the work actually concerns tree search and is therefore more relevant to SAT and CSP rather than a best-first search.

3 Analytical Framework

In this section, we present the analytical framework we use to investigate the phase transition in heuristic search.

3.1 Random Heuristic Search Problems

In search problems, the problem space is often represented as a graph (Pohl 1970) with states as vertices and the transitions as edges. Given a finite state space S of cardinality n: $S = \{s_1, s_2, ..., s_n\}, |S| = n$, and a successor function $\Gamma : S \to 2^S$, we can use the following graph representation (Pohl 1970): $G\langle V, E\rangle$, $V = \{v_i : s_i \in S\}$, $E : \{(v_i, v_j) : s_i, s_j \in S, s_j \in \Gamma(s_i)\}$.

Random Problem Space. With such a representation, generating a random problem requires generating a random graph with characteristics similar to the search space of heuristic search problems (Erdős and Rényi 1959; Karp 1990; Barabási and Albert 1999; Watts and Strogatz 1998).

Common heuristic search problems often exhibit symmetric transitions (or have a similar distribution of in- and out-degrees) along with other properties that can be well-modeled by random graphs. For example, in the Sliding Puzzle, the Pancake Problem, and the Rubik's Cube, the transition function is symmetric ($\forall i, j : s_i \in \Gamma(s_j) \iff s_j \in \Gamma(s_i)$), and most of the states have the same degree, mapping to an undirected model with nearly constant degree distribution. Grid Navigation and Vacuum World, in which a cell is blocked with probability p, as well as other similar problems, have a degree distribution centered around an expected degree that varies based on p. Map navigation problems, e.g. the Arad-Bucharest example (Russell and Norvig 2003), are often symmetric and have a different degree distribution for every map, with no specific structure.

As our focus is not on a specific class of problem, we use the random digraph model (Karp 1990) as a general model. While not committing to symmetry, it maintains an even distribution of in- and out-degrees and a Poisson degree distribution centered around a certain degree. We expect this to be a good general model for random heuristic search problems and, in Section 5, we examine its applicability to real heuristic search problems. When investigating a specific class of problems, using a random graph model that better matches the nature of the class may yield more precise results.

Random Problem Instances. Our model for generating random problem instances is based on a random directed transition graph, a randomly chosen initial state, and a randomly chosen goal state. However, an important technical consideration in problem generation is the avoidance of so-called "flawed" models which include many trivially insoluble instances (Gomes and Walsh 2006). A naive instance generator can easily generate instances for which either the initial state or goal state has no out- or in-transitions, respectively. While such instances do not form an asymptotically dominant proportion of the instance space (Achlioptas et al. 2001), they can nonetheless be easily filtered. Therefore, we propose the following problem generation model.

Model 1. *Let $n \in \mathbb{Z}^+$ be the number of states in the problem space $S = \{s_1, s_2, ..., s_n\}$ and $p \in [0, 1]$ be the connectivity density of the problem space. The class $Q_{n,p}$ consists of all problem instances $\langle T, S_i, S_g\rangle$ such that:*

1. *T is a random transition graph drawn from $D_{n,p}$, the probability space of all random digraphs (Karp 1990).*
2. *$S_i \in S$ is a randomly chosen initial state such that $\exists k \neq i : (S_i, S_k) \in T$.*
3. *$S_g \in S$ is a randomly chosen goal state such that $S_g \neq S_i$ and $\exists k \neq g : (S_k, S_g) \in T$.*

As this model requires a minimal number of edges to provide sufficient variation, we arbitrarily consider instances that have more than 1000 edges.

The Control Parameter. In our investigation, we use the following control parameter:[1]

$$\gamma := \frac{\text{Expected number of edges in the transition graph}}{\text{Number of states}}$$

As the numerator is in the range of [0, $n(n-1)$] (no self-loops), then $\gamma \in [0, n-1]$. When no edges exists ($\gamma = 0$), no path between the initial and the goal states exists. When all edges exist ($\gamma = n - 1$), every potential path between the initial and goal states is a feasible solution. As the expected number of edges (and γ) increases, problems become less constrained and the expected number of solutions increases.

The Region of the Phase Transition. While the phase transition is asymptotically instantaneous, in finite problems the region in which the probability that a problem is soluble changes from nearly zero to nearly one is referred to as the *mushy region* (Smith and Dyer 1996). The point in which this probability is 0.5 is referred to as the *crossover point* (Crawford and Auton 1996). Here, we define the mushy

[1] For digraphs, γ is equivalent to c, often used in the research on phase transitions in random graphs (e.g., Karp (1990)).

region and the crossover point based on the observed proportion of soluble problems as follows.

Definition 1. (Mushy region) *The range of γ in which the observed portion of soluble problems is between 0.1% and 99.9%.*

Definition 2. (Crossover point) *The γ value in which the observed portion of soluble problems is 50%.*

3.2 Search Algorithm

In our analysis, we use no-reopening, unicost greedy best-first search (GBFS) (Doran and Michie 1966). Consistent with all previous phase transition work of which we are aware, this algorithm aims to find a feasible solution.

The Heuristic Function. To investigate the effect of γ on search effort and whether an easy-hard-easy pattern emerges in the phase transition region, we need to use a heuristic that is general enough to apply to our abstract problems but that also embodies useful search guidance. We therefore consider the following set of heuristics $H = \{h_0, h_1, h_2, ...\}$, based on $c(x)$, the real cost from x to goal:

$$h_i(x) = \begin{cases} c(x), & \text{if } c(x) < i \\ i, & \text{otherwise} \end{cases}$$

H is a set of admissible, increasingly informed heuristics. h_0 is the completely uninformed zero heuristic, h_1 only incorporates information on the goal state, while h_i incorporates information on states up to $i-1$ steps away from the goal. We note that $\forall i < j : h_j$ dominates h_i, and since all actions have the same cost, h_j will have a higher rank correlation with the true distance-to-go (Wilt and Ruml 2015).

These heuristic functions are admissible and consistent and we expect they will demonstrate an easy-hard-easy pattern as γ varies: in underconstrained problems, the goal state or one of its close neighbors will be added to the open list quickly, and expanded shortly thereafter; in overconstrained problems, regardless of the heuristic, it should be easy to exhaust the relatively small part of the problem space that is accessible from the initial state; in the mushy region neither of these easy cases will apply, requiring more effort to find a path to the goal or to prove that none exists.

4 Results on the Abstract Model

4.1 Analytical Bounds on the Mushy Region

In the previous work on the phase transition, the mushy region has been defined, as above, using arbitrary thresholds for the observed percentage of soluble instances. An immediate benefit of our random graph model is that we can derive bounds on the mushy region using the theory of random graphs.

Two well-studied phenomena in random graphs are the emergence of a giant component and the connectivity threshold (Frieze and Karoński 2015). Let n be the number of vertices in the digraph, $p = \frac{\gamma}{n}$ be the connectivity density and ω be a non-decreasing function such that $\omega(n) \to \infty$ as $n \to \infty$. For $\gamma < 1$, all components of $D_{n,p}$ are either single vertices or components smaller than ω for any ω, and so we expect the solubility in this region to be asymptotically zero. For $\gamma > \frac{\ln n + \omega}{n}$, the graph is fully-connected, and we expect the solubility in this region to be asymptotically one. Therefore, we can use $\gamma \approx 1$ and $\gamma \approx \frac{\ln n}{n}$ as principled bounds on the mushy region.

4.2 Empirical Results

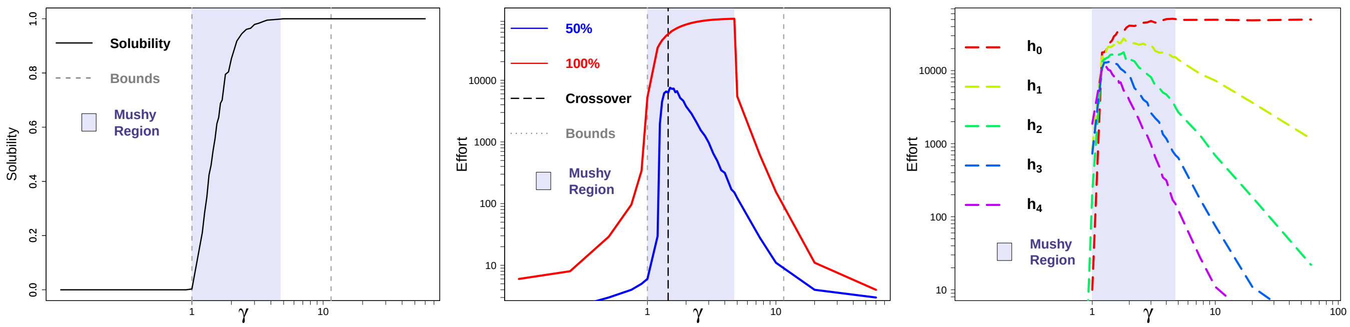
Using our model for random problem generation $Q_{n,p}$, we carried out a series of experiments for $n = \{10000,\ 100000,\ 1000000\}$, and for 35 γ values in $[0, n-1]$, non-uniformly distributed, with higher density within the mushy region boundaries. For each value of n and p, we generated 1000 random instances. For each instance, we record whether or not a solution was found and the number of nodes expanded in order to find a solution or prove that no solution exists. We present results only for $n = 100000$ as the other plots show the same behavior.

The Phase Transition. Figure 1a shows the probability that a solution is found plotted against γ for 100K-state random problems. As we increase γ, there is a clear phase transition in solubility. The observed mushy region is approximately bounded by the analytical bounds, and as problem size grows (not shown), these bounds become looser.

The Hardness of Problems. Figure 1b shows the median (50%-percentile) search effort required to find a first solution using h_4 plotted against γ. There is a clear easy-hard-easy pattern in search effort and the hard problems are associated with the phase transition in solubility. The peak in search effort is located in close proximity to the crossover point ($\gamma \approx 1.65$). These results are consistent with the behavior observed for CSPs (Smith and Dyer 1996).

The hardest instances, however, do not occur near the peak of the median search effort. Figure 1b also shows the 100%-percentile search effort which peaks close to the end of the phase transition at the 99% solubility point. This can be explained, based on our model. The insoluble instances are the rare cases in which a very large giant component is formed with an initial state inside the giant component and a goal state outside. For a forward search, these are the hardest instances to solve. We expect to find similarly hard instances in the same region of high solubility for other types of search. For a backward search, the hardest instances will be the exact opposite, in which the goal state is inside a large component and the initial state outside. For bidirectional search, we expect the hardest instances to be those in which two similarly large components are formed, with the goal state in one and the initial state in another.[2] The hardest soluble instances are the rare cases of a very large giant component that is still very sparsely connected, which requires the search to exhaust most of it in order to find the solution. While we might be able to significantly reduce the required search effort for the hardest soluble instances using a better heuristic function, the hardest insoluble instances require exhausting the accessible portion of the state space to prove infeasibility and cannot be eliminated or improved using a different heuristic function.

[2] A different generator is required to generate such instances.



(a)
Solubility plotted against γ (log scale).

(b) The 50%-, 100%-percentile effort in number of nodes expanded (log-log scale).

(c)
Median effort (log-log scale).

Figure 1: Empirical results for the abstract model for 100K-state random instances.

Similar behavior of the median and 100%-percentile instances have been observed by Gent and Walsh (1994a; 1994b) for different classes of SAT problems, including k-SAT, and by Hogg and Williams (1994) for graph coloring.

The Impact of the Heuristic. While it is standard to use instances across the phase transition to compare heuristic quality (e.g., Gent et al. (1996a), McCreesh et al. (2016)), to our knowledge, previous work has not studied the behavior of a set of heuristics with a known, analytical quality ranking (such as our, h_i, heuristic family) across the phase transition.

Figure 1c shows the effort curve for $h_0, h_1, ..., h_4$ for the *soluble* random instances with 100K states. While for h_0 there is no reduction at all as we move towards less constrained regions, h_4 expands at the crossover approximately 65 times more nodes than at the end of the phase transition ($\gamma \approx 4.75$), and approximately 2,500 times more nodes than at $\gamma \approx 20$. Also, while h_1 expands only twice the number of nodes expanded by h_4 at the crossover, it expands approximately 100 times more nodes at $\gamma \approx 4.75$, and approximately 900 times more nodes at $\gamma \approx 20$.

5 Empirical Results for Benchmarks

In this section, we evaluate the applicability of the results of the abstract model to the analysis of existing heuristic search benchmark problems. We do not expect to see the exact phenomena observed on the abstract model because a set of random variants of an existing problem will have a much smaller variation than our abstract problem generator. However, we expect to observe similar phenomena in close proximity to their occurrence in the abstract model.

As with our study of heuristics of systematically differing strength, we are unaware of phase transition work on other problems that has attempted to directly apply the abstract models of phase transition behavior to existing benchmarks.

5.1 Random Instance Generator

We propose a model for generating restricted or relaxed instances of an existing heuristic search problem at varying constrainedness level. We start by considering the transition graph induced by the problem's state space and create increasingly restricted or increasingly relaxed variants of this graph by removing or adding edges. These variants are proper subgraphs or supergraphs of the original transition graph and the instances near the original constrainedness level should have an almost identical connectivity structure.

Definition 3. (Observed connectivity density) *Let $G\langle V, E\rangle$ be an arbitrary transition graph. We define the observed connectivity density of this graph* $\mathcal{P}(G) = \frac{|E|}{|V|\cdot(|V|-1)}$.

Definition 4. (Restricted instance) *Let $G\langle V, E\rangle$ be an arbitrary transition graph. $\hat{G}\langle V, \hat{E}\rangle$ is considered a restricted instance of G if $\mathcal{P}(\hat{G}) < \mathcal{P}(G)$ and $\hat{E} \subseteq E$.*

Definition 5. (Relaxed instance) *Let $G\langle V, E\rangle$ be an arbitrary transition graph. $\hat{G}\langle V, \hat{E}\rangle$ is considered a relaxed version of G if $\mathcal{P}(\hat{G}) > \mathcal{P}(G)$ and $\hat{E} \supseteq E$.*

Our random model generates restricted or relaxed instances of the original problem with the required connectivity density. Consistent with the abstract model, we eliminate trivially insoluble instances.

Model 2. *Given an existing problem's transition graph $G\langle V, E\rangle$ and the required connectivity density p, the class $R_{G,p}$ consists of all problem instances $\langle T, S_i, S_g\rangle$ such that:*

1. *T, the transition graph, is a restricted instance of G if $p < \mathcal{P}(G)$, or a relaxed instance otherwise. $\mathcal{P}(T) = p$.*
2. *$S_i \in S$, a randomly chosen initial state, $\exists k : (S_i, S_k) \in T$*
3. *$S_g \in S$, a randomly chosen goal state such that $S_g \neq S_i$ and $\exists k : (S_k, S_g) \in T$*

We use domain-specific heuristic functions to solve the benchmark problems. It should be noted that these heuristics remain admissible for the restricted instances but not for the relaxed instances. However, admissibility is not required for our analysis.

5.2 Benchmark Problems

In this section, we describe the empirical results for five benchmark domains. In order to perform experiments with thousands of random instances, we use relatively small versions of these problems.

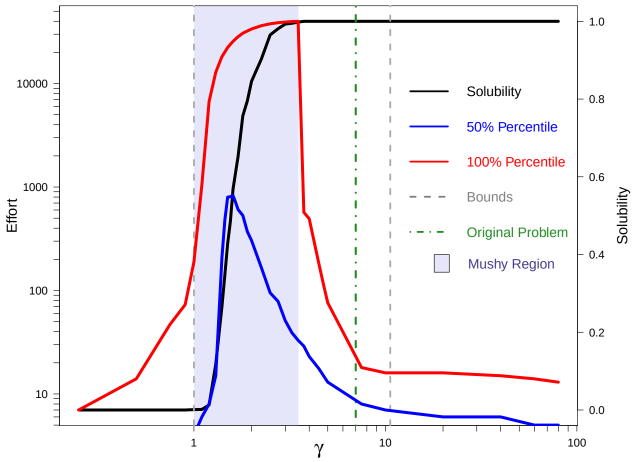


Figure 2: 8-Pancake Problem: Solubility and search effort (50% and 100% percentile) plotted against γ (log-log scale).

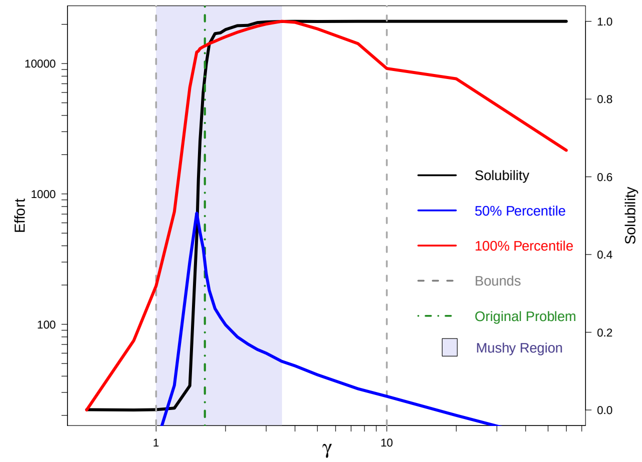


Figure 3: 150×150 Grid Navigation Problem: Solubility and search effort plotted against γ (log-log scale).

The Pancake Problem. We consider the 8-Pancake Problem, for which the observed connectivity density is $\mathcal{P} = \frac{8! \cdot 7}{8! \cdot (8! - 1)} = \frac{7}{8! - 1}$ ($\gamma \approx 7$). For $p < \mathcal{P}$, we generate restricted instances of the problem, in which some of the flip operators are not allowed. For $p > \mathcal{P}$, we generate relaxed instances, in which there exist additional operators that do not correspond to valid flip. We use h^{gap} (Helmert 2010), a landmark heuristic based on the number of gaps in the pancake stack.

Figure 2 shows the probability of a solution and the required search effort. There is a clear phase transition in solubility and the mushy region is bounded by the analytical bounds. The median effort peaks near the crossover point with the hardest problems near the end of the mushy region. Since $\mathcal{P}$ is relatively high, we can see that restricting the problem, does not immediately reduce the solubility. These results are in agreement with our abstract model.

Grid Navigation. We consider a 150×150 Grid Navigation Problem with 36% randomly chosen blocked cells, solved using the Manhattan distance heuristic. We estimate $\mathcal{P} \approx 1.63$ based on the observed average number of edges in the transition graph of 1000 random instances.

Figure 3 shows the probability of solution and the required search effort. For the original problem, 85% of the instances were soluble. The peak of the median effort is located near the crossover point and the hardest problems are located near the end of the phase transition. We can also see that the mushy region is bounded by the analytical bounds.

Sliding Puzzle. For the Sliding Tile 8-Puzzle, we observe a different behavior (Figure 4). In its original form, the 8-Puzzle is not always soluble if the initial and goal states are randomly chosen because the state space consists of two equal-sized connected components. Consequently, the solubility at the original constrainedness $\mathcal{P}$ is approximately 50%. Furthermore, the behavior in the near-$\mathcal{P}$ region is different. As the problem is relaxed, it immediately becomes 100% soluble, since every edge that connects the two large components results in a fully connected state space. Alternatively, as we restrict the problem, the solubility does not immediately decline, and remains approximately 50% for a short range of γ, and the median effort therefore belongs to either a soluble or insoluble instance. Due to the unique structure of two large components, almost all the insoluble instances in the near-$\mathcal{P}$ region require near-maximal effort, as we have to exhaust one of the components. The result is a very noisy median as shown in Figure 4. As we further restrict the problem, the two large components break into smaller components and the effort decreases.

It is interesting to observe that, unlike the other benchmarks we have studied, the original state space of the 8-Puzzle is harder than any restricted or relaxed version of it.

Other Benchmark Problems. Experiments on Towers of Hanoi and TopSpin showed similar patterns to the Pancake

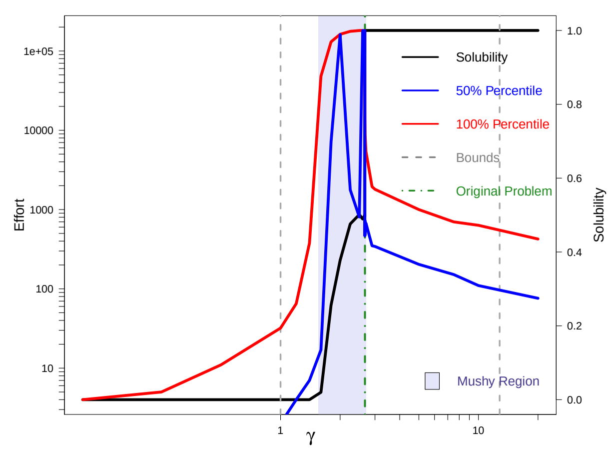


Figure 4: 8-Puzzle: Solubility and search effort plotted against γ (log-log scale).

Problem and the Grid Navigation and were in agreement with the abstract model. We omit their results due to space.

5.3 The Impact of the Heuristic Function

In order to evaluate the impact of the heuristic's quality on the search effort for existing heuristic search problems, we carried out a series of experiments on the Pancake Problem.

We consider the original h^{gap} heuristic (Helmert 2010), and propose $H'^{gap} = \{h_0'^{gap}, h_1'^{gap}, h_2'^{gap}, ...\}$, a set of partial and increasingly stronger versions of h^{gap}:

$$h_i'^{gap}(x) = h^{gap}(\text{bottom } i \text{ pancakes in } x)$$

For an n-Pancake problems, $h_0'^{gap}$ is the uninformed zero heuristic, while $h_n'^{gap}$ is the original h^{gap} heuristic.

Figure 5 and Table 1 show the median effort for soluble instances. Similar to the abstract model, the impact of a better heuristic is much stronger outside of the phase transition. $h_8'^{gap}$ is approximately 1.75 times better than $h_2'^{gap}$ at the crossover point ($\gamma \approx 1.55$), approximately 11.5 times better at the end of the mushy region ($\gamma \approx 3.5$), and approximately 69 times better at $\gamma \approx 60$. Note the very small difference in effort among $\{h_i'^{gap} | i > 0\}$ at the crossover point with $h_2'^{gap}$ even expanding fewer median nodes than $h_4'^{gap}$.

6 Discussion

The successful prediction of the behavior of the benchmark problems indicates some structural properties of these problems are properly modeled by our abstract model. As all these benchmark problems are essentially puzzles, their state space is symmetric, they have roughly the same number of operators for every state, and they have one goal state. The fact that the 8-Puzzle results do not match our abstract model is due to a mismatch with the connectivity assumptions in our model. The 8-Puzzle demonstrates interestingly different behavior and the use of the phase transition framework leads us to useful insights into the structure of the problem.

While the analytical bounds on the mushy region seem to hold, even on the benchmarks, predicting the location of the crossover remains an open question. For our model, the number of solutions corresponds to the number of s-t paths in a directed graph, a well-studied problem (Valiant 1979). According to the theory of constrainedness (Gent et al. 1996b), the crossover can be predicted at the point where the expected number of solutions is exactly one. Unfortunately, calculation of the expected number of s-t paths in random digraph remains an open problem with some potentially useful estimation results (Roberts and Kroese 2007).

Our investigation also showed that the effort improvement due to a more informed heuristic is much smaller for more constrained problems, especially for problems inside the phase transition – exactly the hardest problems. If we can hope to generalize empirical comparisons of heuristics and develop an understanding of problem difficulty for heuristic search, we should, therefore, take into account the location of the problem sets on the phase transition as is done, for example, in CSPs (Gent et al. 1996a).

As a major direction of our future work, we will investigate the application of the results above to other heuristic search algorithms such as cost-based GBFS and A*. Given the interest in characteristics that correlate with problem difficulty noted in Section 1, we believe that an analysis based on the phase transition will provide valuable insight.

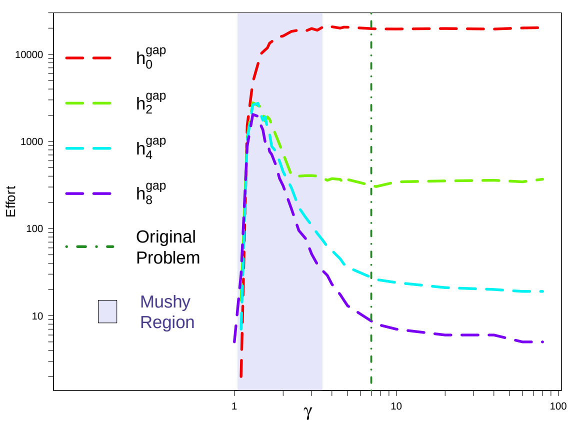


Figure 5: 8-Pancake Problem: Search effort for soluble instances using $h_0'^{gap}, h_2'^{gap}, h_4'^{gap}, h_8'^{gap}$ (log-log scale).

γ value	$h_0'^{gap}$	$h_2'^{gap}$	$h_4'^{gap}$	$h_8'^{gap}$
$\gamma \approx 1.55$	10,647	1,715	1,921	981
$\gamma \approx 3.5$	20,500	380	74	33
$\gamma \approx 20$	19,798	352	21	6
$\gamma \approx 60$	20,098	344	19	5

Table 1: 8-Pancake Problem: Median effort

7 Conclusion

We performed the first empirical analysis of the phase transition in heuristic search, focusing on greedy best first search. Our results establish the existence of a rapid transition in the solubility of heuristic search problems and the occurrence of the hardest problems during this transition. These results connect heuristic search to a variety of problems and body of literature for which similar results have been obtained.

We also performed a novel investigation of the behavior of systematically stronger heuristics across the phase transition region and showed that the benefit of more informed heuristics is orders of magnitude smaller for hardest problem instances (i.e., those at the phase transition).

Finally, we demonstrated the practicality of these results by successfully predicting the behavior of restricted and relaxed versions of common benchmark problems. We believe that this is the first time that phase transition analysis has been applied to existing benchmark problems for any type of problem.

Acknowledgements We would like to thank the anonymous reviewers whose valuable feedback helped improve the final paper. The authors gratefully acknowledge funding from the Natural Sciences and Engineering Research Council of Canada.

References

Achlioptas, D.; Molloy, M. S.; Kirousis, L. M.; Stamatiou, Y. C.; Kranakis, E.; and Krizanc, D. 2001. Random constraint satisfaction: A more accurate picture. *Constraints* 6(4):329–344.

Barabási, A.-L., and Albert, R. 1999. Emergence of scaling in random networks. *Science* 286:509–512.

Bylander, T. 1996. A probabilistic analysis of propositional STRIPS planning. *Artificial Intelligence* 81(1):241–271.

Cheeseman, P.; Kanefsky, B.; and Taylor, W. M. 1991. Where the really hard problems are. In *The Twelfth International Joint Conference on Artificial Intelligence (IJCAI)*, volume 91, 331–337.

Crawford, J. M., and Auton, L. D. 1996. Experimental results on the crossover point in random 3-SAT. *Artificial Intelligence* 81(1):31–57.

Doran, J. E., and Michie, D. 1966. Experiments with the graph traverser program. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 294, 235–259. The Royal Society.

Erdős, P., and Rényi, A. 1959. On random graphs. *Publicationes Mathematicae Debrecen* 6:290–297.

Frieze, A., and Karoński, M. 2015. *Introduction to random graphs*. Cambridge University Press.

Gent, I. P., and Walsh, T. 1994a. Easy problems are sometimes hard. *Artificial Intelligence* 70(1):335–345.

Gent, I. P., and Walsh, T. 1994b. The hardest random SAT problems. In *KI-94: Advances in Artificial Intelligence*, 355–366. Springer.

Gent, I. P., and Walsh, T. 1999. Beyond NP: the QSAT phase transition. In *The Sixteenth National Conference on Artificial Intelligence (AAAI)*, 648–653.

Gent, I. P.; MacIntyre, E.; Presser, P.; Smith, B. M.; and Walsh, T. 1996a. An empirical study of dynamic variable ordering heuristics for the constraint satisfaction problem. In *International Conference on Principles and Practice of Constraint Programming (CP'96)*, 179–193. Springer.

Gent, I. P.; MacIntyre, E.; Prosser, P.; and Walsh, T. 1996b. The constrainedness of search. In *The Thirteenth National Conference on Artificial Intelligence (AAAI)*, 246–252.

Gomes, C., and Walsh, T. 2006. Randomness and structure. In Rossi, F.; Van Beek, P.; and Walsh, T., eds., *Handbook of constraint programming*. Elsevier. 639–664.

Helmert, M. 2010. Landmark heuristics for the pancake problem. In *Third Annual Symposium on Combinatorial Search (SoCS)*, 109–110.

Hogg, T., and Williams, C. P. 1994. The hardest constraint problems: A double phase transition. *Artificial Intelligence* 69(1-2):359–377.

Huberman, B. A., and Hogg, T. 1987. Phase transitions in artificial intelligence systems. *Artificial Intelligence* 33(2):155–171.

Karp, R. M. 1990. The transitive closure of a random digraph. *Random Structures & Algorithms* 1(1):73–93.

McCreesh, C.; Prosser, P.; and Trimble, J. 2016. Heuristics and really hard instances for subgraph isomorphism problems. In *The 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 631–638.

Mitchell, D.; Selman, B.; and Levesque, H. 1992. Hard and easy distributions of SAT problems. In *The Tenth National Conference on Artificial Intelligence (AAAI)*, 459–465.

Pohl, I. 1970. Heuristic search viewed as path finding in a graph. *Artificial Intelligence* 1(3):193–204.

Prosser, P. 1996. An empirical study of phase transitions in binary constraint satisfaction problems. *Artificial Intelligence* 81(1):81–109.

Rintanen, J. 2004. Phase transitions in classical planning: An experimental study. In *The Fourteenth International Conference on Automated Planning and Scheduling (ICAPS)*, volume 2004, 101–110.

Roberts, B., and Kroese, D. 2007. Estimating the number of s-t paths in a graph. *Journal of Graph Algorithms and Applications* 11(1):195–214.

Russell, S. J., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach.* Pearson Education, 2nd edition.

Smith, B. M., and Dyer, M. E. 1996. Locating the phase transition in binary constraint satisfaction problems. *Artificial Intelligence* 81(1):155–181.

Valiant, L. G. 1979. The complexity of enumeration and reliability problems. *SIAM Journal on Computing* 8(3):410–421.

Watts, D. J., and Strogatz, S. H. 1998. Collective dynamics of "small-world" networks. *Nature* 393:440–442.

Wilt, C. M., and Ruml, W. 2011. Cost-based heuristic search is sensitive to the ratio of operator costs. In *The Fourth Annual Symposium on Combinatorial Search (SoCS)*, 172–179.

Wilt, C. M., and Ruml, W. 2012. When does weighted A* fail? In *The Fifth Annual Symposium on Combinatorial Search (SoCS)*, 137–144.

Wilt, C. M., and Ruml, W. 2015. Building a heuristic for greedy search. In *The Eighth Annual Symposium on Combinatorial Search (SoCS)*, 131–139.

Xie, F.; Müller, M.; and Holte, R. 2015. Understanding and improving local exploration for GBFS. In *The Twenty-Fifth International Conference on Automated Planning and Scheduling (ICAPS)*, 244–248.