Precomputation for Efficient Exploration of Hypotheses about Novelty Search and Evolvability

ABSTRACT

A common aim in evolutionary search is to skillfully navigate complex search spaces. Achieving this aim requires creating search algorithms that exploit the structure of such spaces. However, to exhaustively analyze such structure is generally intractable, due to the expansiveness of most search spaces. Researchers thus typically develop intuitions about complex search spaces indirectly through experimentation in involved domains, or through light-weight theoretical models. However, empirical work sacrifices ground truth about the search space's true connectivity, and theoretical models risk disconnection from actual problem domains. In the context of evolutionary robotics and artificial life, this paper suggests a middle-ground approach, which combines a full-fledged domain with an expressive but limited encoding, and then precomputes the behavior of all possible individuals, enabling evaluation as a look-up table. The product is an experimental playground in which search is non-trivial yet which offers extreme computational efficiency and ground truth about search-space structure. The framework is open-sourced and released with this paper, which describes the approach and demonstrates its usefulness with applications to evolvability and novelty search, evaluated in a popular benchmark task. The hope is that the extensible framework enables quick experimentation and idea generation, aiding brainstorming of new search algorithms and measures.

CCS CONCEPTS

•**Theory of computation** → *Evolutionary algorithms;* •**Computer systems organization** → *Neural networks; Evolutionary robotics;*

KEYWORDS

Evolvability, Novelty Search, Neural Networks

ACM Reference format:
. 2017. Precomputation for Efficient Exploration of Hypotheses about Novelty Search and Evolvability. In *Proceedings of Genetic and Evolutionary Computation Conference, Berlin, Germany, July 2017 (GECCO'17),* 8 pages.
DOI: 10.475/123_4

1 INTRODUCTION

Broadly across evolutionary computation (EC) it is important to navigate complex search spaces to find individuals with rare properties. Most commonly, evolutionary algorithms (EAs) search for an optimal point in the search space, but other paradigms include accumulating points that span optimal trade-offs among competing objectives (multiobjective optimization [1]); or collecting a diverse set of individuals that instantiate a wide variety of interesting and innovative behaviors (e.g. as in open-ended evolution [2] or computational creativity [3]). Across nearly all such use cases, a primary challenge is to create search algorithms that respect the structure of the search space. In other words, what combination of algorithm and incentives will enable skillful navigation of the space, to uncover relevant points of interest?

Although building intuitions about the structure of complex search spaces is important, such intuitions are often difficult to acquire because search-space structure resists direct investigation. One challenge is that interesting search spaces are often effectively infinite, due to continuous-valued parameters and expanding variable-length representations; such infinitude frustrates exhaustively enumerating and evaluating all possible individuals, which might allow definitive analysis of a search space's connectivity and structure. Thus in practice, researchers learn indirectly about search space structure through iterative empirical work, through theoretical models, or through thought experiments. These options can be viewed along a gradient of abstractness, where direct empirical work in domains of interest is completely grounded, and thought experiments are the loosest (but most free). Each of these approaches have benefits and drawbacks, and a main contribution of this paper is to propose a principled intermediate that exploits advances in computation to trade up-front computation to maximize groundedness and tractability.

The main idea is to precompute the behaviors of all individuals in a barely-tractable search space, evaluated within a grounded domain. By evaluating all individuals once and storing the results, evaluation becomes computationally trivial (e.g. a look-up table), and it becomes possible to calculate ground-truth quantities, such as the absolute potential of a particular encoding for evolvability within that domain, or how well a particular fitness measure correlates with actual genomic distance to a goal behavior. The hope is to create an experimental playground that can be useful for quickly testing ideas and building experimenter intuition.

As a proof-of-concept implementation of precomputed domains, this paper adopts a common maze navigation benchmark domain common within artificial life and evolutionary robotics (ER), pairs it with a discretized neuroevolution encoding, and stores the results of evaluating all individuals in a database that fits in RAM. The enumerated search space allows various ground-truth properties of the search-space to be explored, including otherwise intractable generalizations of evolvability and the exact distribution of specific behaviors like solutions within the space. Enough runs can be conducted in minutes on one computer to generate statistically significant data, enabling quick iteration.

GECCO'17, Berlin, Germany
© 2017 ACM. 123-4567-24-567/08/06...$15.00
DOI: 10.475/123_4

To highlight its potential, the implementation is applied here to test hypotheses related to novelty search and evolvability. Additionally, the performance of idealistic search algorithms that exploit generally-intractable information is probed, leading to speculation about new algorithms driven by approximations of such quantities. This range of applications suggests the value of precomputed domains for idea-generation and initial exploration of hypotheses, which is critical in early stages of research. The implementation is open-source and available for download (http://goo.gl/JTSTGs[1]), ideally to serve as an extensible framework that allows other precomputed domains to be easily distributed among researchers.

2 BACKGROUND

The next section first reviews existing methods for probing the structure of search spaces, then reviews the novelty search algorithm that provides a setting for testing costly hypotheses. Finally, the concept of evolvability is reviewed, which acts as a concrete example of an expensive measure that precomputed domains can render tractable.

2.1 Exploring Search Space Structure

Because understanding search space structure is fundamental to designing effective EC algorithms, there are a range of formal and informal techniques to quantify or explore it. For example, one line of research aims to investigate what properties of search spaces make problems difficult to solve for EAs [4–6], like deception [6] or ruggedness [5]. The idea is that if one suspects a problem of interest has such properties, that understanding can guide algorithmic design or focus future research. Most often, mathematical models or toy domains are used to make analysis tractable, such as the popular NK model of fitness landscape ruggedness [5], or constructed bitwise models such as the royal road function [7] or the trap function [6].

Less formal methods include problem-specific human analysis, or iterative sequences of experimentation, analysis, and tweaking. For example, researchers often embed their knowledge of a domain into the encoding (e.g. locomoting biped agents might more easily realize stable cyclic gaits if oscillatory patterns are provided as a basic element [8]), or adjust the fitness function through iterations of experiments followed by changes aimed at remedying problematic dynamics [9]. Interactive evolution, or combinations of interactive evolution and mechanical evaluation can also yield insights into search spaces by enabling humans to more directly probe them [10, 11].

The method proposed here attempts to enable leveraging the benefits both of formal and informal methods more easily. In particular, it aims to create domains that are tractable to measure ground-truth formal properties, such as ruggedness or deception, while maintaining computational efficiency and groundedness to real problems, thereby enabling fast and flexible experimentation.

2.2 Novelty Search

In this paper, the precomputed domain approach is applied to a common benchmark task for divergent search methods like novelty search, which in particular serves as a focal point for experimentation as a prototypical use case. To provide necessary context, this section reviews the novelty search method.

Novelty search is inspired by natural evolution's drive towards novelty, and rewards novel behavior directly *instead* of progress towards a fixed objective [12]. Tracking novelty requires little change to any evolutionary algorithm aside from replacing the objective-based fitness function with a *novelty metric*. Such a metric measures how different an individual is from other individuals, thereby creating a constant pressure to produce something new. The key idea is that instead of rewarding performance on an objective, novelty search rewards diverging from prior behaviors. Therefore, novelty in behavior needs to be *measured*.

The novelty metric characterizes how far away the new individual is from the rest of the population and its predecessors in *behavior space*, i.e. the space of unique behaviors. A good metric should thus compute the *sparseness* at any point in the behavior space. Areas with denser clusters of visited points are less novel and therefore rewarded less.

A simple measure of sparseness at a point is the average distance to the k-nearest neighbors of that point. Intuitively, if the average distance to a given point's nearest neighbors is large then it is in a sparse area; if the average distance is small, it is in a dense region. The sparseness ρ at point x is given by

$$\rho(x) = \frac{1}{k}\sum_{i=0}^{k} \mathrm{dist}(x, \mu_i), \tag{1}$$

where μ_i is the ith-nearest neighbor of x with respect to the distance metric dist, which is a domain-dependent measure of behavioral difference between two individuals in the search space. Candidates from more sparse regions of the behavior space thus receive higher novelty scores.

With fixed probability an individual is entered into the permanent archive that characterizes the distribution of prior solutions in behavior space. The current generation plus the archive constitute a comprehensive sample of where the search has been and where it currently is; that way, by attempting to maximize the novelty metric, the gradient of search is simply towards what is *new*, with no other explicit objective. However, even without an explicit objective, novelty search is still driven by meaningful information; that is, behaving in a novel way often requires learning the structure of the domain.

Once objective-based fitness is replaced with novelty, the underlying EA operates as usual, selecting the most novel individuals to reproduce. Over generations, the population spreads out across the space of possible behaviors.

While novelty search imposes no direct pressure to achieve any particular objective, it has been successfully applied in a range of domains [12–14]. Most relevant to this paper, it has previously been shown to lead to increased evolvability relative to objective-driven search [15–17]; this link between novelty search and evolvability is explored in more depth in this paper's experiments. Note that the experiments here apply novelty search to evolve artificial neural networks (ANNs) that control the behavior of a simulated robot, as is common in previous such experiments [12, 15]. In particular, the connection weights of a fixed-topology ANN are evolved; the setup is described in more detail in the approach section.

[1] Temporary anonymous download site

2.3 Estimating Evolvability

Natural evolution has produced flexible, highly evolvable representations that facilitate its prolific discovery of diverse organisms. An important question with implications for EC is: What properties of natural evolution led to such evolvability? If well-understood, such properties could be built into EAs to enable more powerful algorithms. Thus tools for exploring evolvability, such as quantitative measures and estimates of it, or methods to directly search for it, can help isolate its characteristics and the evolutionary features that encourage evolvability.

While there is no overall consensus on evolvability's definition or its measurement [18], one common conception is to consider evolvability as an organism's phenotypic variability [19–22]; that is, the capacity of an organism's lineage to generate novel phenotypic traits captures some significant part of what enables some lineages to adapt more quickly than others, although there exist alternative definitions that focus on different or overlapping aspects of evolvability [18]. This conception (of evolvability as phenotypic variability) aligns well with the motivation of novelty search, and is adopted here to help explore hypotheses about novelty search, in a way similar to previous related studies [15, 23].

The evolvability measure most often used in prior novelty search studies *estimates* an individual's evolutionary potential by counting the number of unique behaviors exhibited by samples of offspring within its immediate mutational neighborhood [15, 15, 17]. That is, the measure attempts to gauge an individual's phenotypic connectivity. But such measures are expensive because they depend on evaluating the behaviors instantiated by many perturbations of an individual's genome; i.e. 200 evaluations are often required to reliably estimate a single individual's evolvability [17]. This makes efforts to measure evolvability frequently during search, or to drive search explicitly to maximize evolvability [17], painfully expensive. Furthermore, because such measures of evolvability take into account only the local mutational neighborhood, it may not well-reflect an individual's medium-term or long-term evolutionary potential. The approach here enables efficient precalulation of exact evolvability, across longer time-scales, and is described next.

3 PRECOMPUTED DOMAINS

Precomputed domains are motivated by the importance of developing intuitions about complex search spaces, which aides inventing new algorithms and understanding existing ones more deeply. The primary idea is to stake out a useful middle ground between mathematical models and full-complexity challenge domains.

The benefit of mathematical models is their elegance, computational efficiency, and the access they often provide to ground-truth metrics, such as the distribution of solutions or evolvability. However, they sometimes run the risk of begging the question, i.e. they are often based on axioms that may or may not reflect search spaces of interest, and thus transferring insights learned from them to practical domains may fail. On the other hand, the benefits of direct empirical investigations into domains of interest are their pragmatism and groundedness. That is, physical simulations ensure that individuals must overcome common constraints that naturally emerge across real-world situations, like navigating around obstacles or coordinating robotics limbs to locomote. However, the complexity of rich simulations incurs computational expense, which slows the iterative loop through which researchers develop new techniques and understanding; complicating such expense, to make reliable statistical judgments requires many independent runs, meaning that research may be gated to those having access to large clusters. Additionally, by definition ground-truth is absent in cutting edge domains, e.g. if the distribution of solutions were known, the problem ceases to be cutting-edge, or is itself computationally expensive even to estimate, e.g. an individual's evolvability.

The main idea is to *precompute* the behavior of all possible genotypes in full-fledged domains, leading to evaluation as a look-up table. Thus many runs can be quickly completed on consumer hardware, enabling more easily testing hypotheses that depend on an otherwise exorbitant number of runs. Furthermore, if all genotypes are enumerated, it then becomes possible to compute the ground-truth distance from an individual to the objective of search, or to any other possible behavior of interest. Thus some hypotheses become more amenable to direct investigation. Note that while this methodology is generally applicable to search in EC, the case study presented here assumes a setting beyond black-box optimization [24], i.e. there is access to information beyond scalar fitness values concerning the *behavior* of an individual in its domain, as is typical in ER or artificial life.

A side-benefit from calculating such exact quantities for each individual is that search can then be efficiently *driven* to maximize them, or easily instrumented by them. For example, evolvability search is an interesting algorithm for exploring ideas related to evolvability [17], but is much slower than a typical search process, because it requires the expensive approximation of each individual's evolvability (which requires many domain evaluations). However, beyond enabling quick experimentation with existing algorithms like evolvability search, precomputed domains further enable optimizing fanciful measures, such as an ideal generalization of evolvability, e.g. the average genotypic distance from an individual to *every* behavior, or directly optimizing for behavioral rarity. While these quantities may be entirely impractical to measure and optimize in practice, seeing their promise (or lack thereof) may guide the construction of future practical measures or approximations.

It is important to acknowledge that precomputing the behavior of all possible genotypes is not possible in general, because most search spaces are impractically large, e.g. they are often effectively infinite because of continuous parameters, or mutations that iteratively extend the length of the genotype. As a result, the approach taken here is to construct a search space that stretches tractability towards reasonable limits of computation and memory. In particular, one explicit design consideration is that the precomputed search space should fit in RAM on a modern computer, to maximize computational efficiency; note that the discussion section discusses how the search space can be further stretched by relaxing this in-memory constraint.

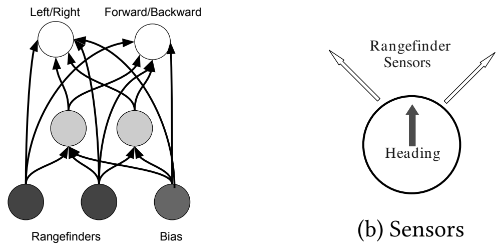
Thus it is important to examine how such considerations limit the number of parameters that can realistically be evolved in a precomputed domain. Assuming a fixed-length discrete representation in which each of G genes has A possible alleles, the resulting search space will contain A^G distinct individuals. Because this quantity is exponential in G, there are strong limits on how many genes can be added. There is a significant cost to added alleles as well, as the search space grows with them relative to the Gth power. As a result, an important design consideration when adopting an encoding with continuous parameters (e.g. the neural network encoding adopted in this paper's experiments) is how few parameters are necessary, and how granularly those parameters can be discretized without rendering the search space impassable or uninteresting. Given a relatively modern computer with 8GB of RAM, and 16 bytes of storage for each precomputed individual, the magnitude of an enumerable search space is 500 million individuals, which while not infinite, provides a non-trivial search space. Note that the proof of concept domain explored here contains 34 million individuals, meaning there is still some room to further scale the approach.

4 PRECOMPUTED MAZE NAVIGATION

As a proof-of-concept implementation that introduces the precomputed domain approach, this paper adopts a common maze-navigation domain benchmark that is often used to evaluate non-objective search algorithms such as novelty search, behavioral diversity, and MAP-ELITES [12, 16, 17, 25]. As in previous work, we conduct experiments comparing novelty search and traditional objective-based search.

In the maze navigation domain, a simulated wheeled robot (figure 1) is embedded in a two-dimensional maze (figure 2). The objective for the robot is to traverse the maze and arrive at a fixed goal point. Thus, the objective-based fitness function f of an individual for objective-based search is $f = -d_g$, where d_g is the distance of the robot to the goal at the end of the evaluation. For novelty search evolution instead requires a characterization of behavior. Because ending location is a critical factor in navigating mazes, the behavior of a robot is defined as its location in the maze at the end of the evaluation [12, 25]. For measuring evolvability, each grid square within a regular grid superimposed over all ending locations acts a discrete niche. Offspring are mapped into the niche that contains the behavior they exhibit when evaluated. The precomputed domain mirrors the canonical setup introduced in Lehman and Stanley [12].

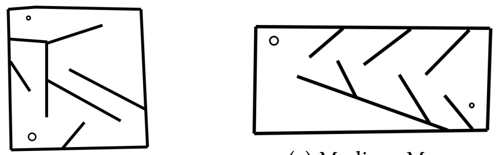
The canonical setup of this domain applied the NEAT neuroevolution encoding [26], which features continuous-valued evolvable weights and mutations that add new neurons and connections to the ANN. Because such features manifest a search space containing effectively infinite individuals, NEAT, at least in its usual form, is incompatible with precomputing the behavior of all individuals. Thus a discretized and bounded ANN encoding is adopted in the experiments here. In particular, weights take on the discrete values of -1, 0, and 1, and a feed-forward two-layer fully-connected topology with two hidden neurons is employed (figure 1a). Further, the agent's sensors were reduced to a minimal set, to restrict the size of the search space, which grows exponentially in the number of connections. In particular, the agent's pie-slice radar sensors are removed, and the number of range-finder sensors is reduced from six to two, as shown in figure 1. This reduction of sensor information increases the difficulty of navigation, as the agent can no longer discern directly in which direction the goal lies; to partially offset such difficulty, the evaluation time in each maze is extended from 400 timesteps to 600.



(a) Neural Network

(b) Sensors

Figure 1: A Maze-Navigating Robot. The artificial neural network that controls the maze navigating robot is shown in (a). The layout of the sensors is shown in (b). Both arrows outside of the robot's body in (b) are rangefinder sensors that indicates the distance to the closest obstacle in that direction. The solid arrow indicates the robot's heading. Note that the sensors of the robot are reduced from the setup in Lehman and Stanley [12] to limit the size of the search space, and that the neural network has a fixed topology, instead of an evolved topology as when using the NEAT algorithm.



(a) Hard Maze

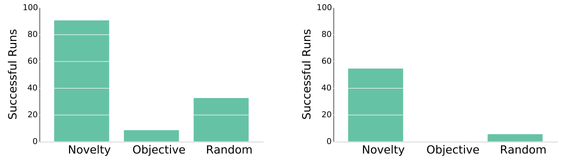
(a) Medium Maze

Figure 2: Maze Navigation Maps. In both maps, the larger circle represents the starting position of the robot and the smaller circle represents the goal. To solve the task, the robot must navigate around obstacles, which requires the evolution of non-trivial behavior. The (a) medium map has a series of cul-de-sacs that instantiate local optima with objective-based fitness, while the (b) hard map has a highly deceptive cul-de-sac that requires significant further navigation before a robot can achieve a higher objective-based fitness score.

The resulting encoding consists of 16 connections that can each take on 3 distinct weight values, realizing a search space with 3^{16} individuals (43 million). Each of these individuals were separately evaluated in both mazes, and their behavior (the point within the maze they ended upon) and whether they solved the maze, was recorded in a binary data file. Evaluation was conducted on a single multi-core desktop machine, and took approximately one hour to complete when parallelized over eight threads. Because fitness in this case can be calculated as a byproduct from an individual's behavior, there was no need to separately store such information.

4.1 Validating the Precomputed Domain

In contrast to the original NEAT setup, the precomputed encoding is streamlined and heavily discretized, motivating validation experiments to probe whether qualitative similarity is preserved. To do so, fifty runs each of objective-based search, novelty search, and



(a) Medium Maze (b) Hard Maze

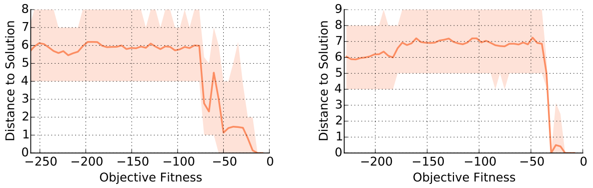
Figure 3: Precomputed Maze Navigation Validation. The number of successful runs out of 100 is shown in (a) the precomputed medium map domain, and (b) the precomputed hard map domain. Consistent with previous results, novelty search performs the best in both domains, and the performance of both methods decreases when evaluated in the hard maze relative to the medium maze.

random search were run for 250 generations with a population size of 500 individuals. The EA is a simple generational model that uses tournament selection, protects the champion with elitism, and has no crossover or diversity maintenance. Mutation is performed on 80% of offspring, and replaces a the weight of a randomly chosen connection with a value chosen at random. Due to evaluation as a look-up table, these 600 runs (100 for each method across two mazes) took under 12 minutes on a modern laptop using a single core; all other experiments described in this paper required similarly trivial runtime.

The results are shown in figure 3 for both mazes. Novelty search significantly out-performs the other methods on both mazes, while objective-based search performs worse than random search in both domains (Fisher's exact test; $p < 0.05$). One divergence from results in the canonical (i.e. non-precomputed) domain is that objective-based fitness usually can solve the medium maze, albeit more slowly than can novelty search. Follow-up experiments revealed that the precomputed encoding rendered the initial cul-de-sac significantly more deceptive than in the canonical setup; one cause may be a lack of diversity maintenance in the EA, although preliminary experiments that reduced selection pressure or rewarded genotypic diversity did not outperform random search. A reasonable hypothesis is that the lack of pie-slice sensors and the reduced number of rangefinders may make it more difficult for mutations to generate significant behavioral diversity by chance. However, consistent with previous results, the hard maze more actively leads objective-based search astray (which never solves the task), and is more difficult for novelty search as well. In this way, the results of evolution in the precomputed encoding are coherent and share significant qualitative traits with the original setup, implying that it likely can serve as a useful proxy.

4.2 Exact Quantification of Deception, Evolvability, and Rarity

One quantitative measure of deception, called fitness distance correlation (FDC; [4]), calculates the correlation between the fitness of an individual and the minimal genomic distance from it to a solution. In other words, an ideal fitness function would incentivize moving in the genotypic space towards a solution, e.g. an easy non-deceptive problem has a large and *negative* FDC (because distance to solution should *decrease* with higher fitness. While for full-fledged domains it is generally intractable to calculate the minimal distance to a solution, precalculated domains provide complete knowledge of the search-space, enabling identifying all solutions, and measuring shortest-path distances from all individuals to solutions.



(a) Medium Maze (b) Hard Maze

Figure 4: Fitness Distance Correlation in Precomputed Mazes. How objective-based fitness values relate to true genomic distance to a solution is shown for the (a) Medium Maze and (b) Hard Maze. Fitness scores are discretized into fifty uniformly-size intervals; the mean fitness value is plotted as a solid line, and the surrounding red fill encompasses 95% of the distribution within each interval. The conclusion is that objective-based fitness offers only weak signal in the medium maze, and is actively deceptive in the hard maze until a navigator is already very close to the goal.

We calculate such shortest-path distance using an iterative depth-first search, which starts from the set of solution individuals (which can be identified through a simple query of the precomputed database). Interestingly, solutions to either maze are very rare within the search space; there are only 320 solutions to the medium maze, and 59 solutions to the hard maze exist within the 43 million total individuals. How fitness and distance to solution correlate in both mazes is shown in figure 4. FDC, calculated as the Pearson correlation coefficient between fitness score and solution distance, is slightly negative in the medium maze ($r = -0.001$), indicating a near-lack of correlation between fitness and distance to goal, while the hard maze has a larger positive correlation ($r = 0.043$), validating the natural intuition that the hard maze is the more deceptive map.

A variation of the same approach can be applied to quantify exact evolvability of individuals. As reviewed in the background, one popular evolvability estimate in ER is to measure how many distinct behaviors occur among a random sample of an individual's offspring [15, 17, 23]. The idea is that an evolvable individual is one that provides a stepping stone to many other phenotypes. Using the same minimal-distance approach, but applying it once to each distinct behavior in the domain (as discretized into 10 unit x 10 unit grid-squares containing all individuals with evaluations ended within that square), we can create look-up tables that store the minimum number of mutations needed for any given individual to demonstrate any given behavior. This approach enables efficient calculation of a *generalization* of the 1-step evolvability measure typically used in practice, i.e. the *k-step* evolvability, by querying how many behaviors are within k mutations of a given individual; intuitively, the larger the k, the longer the time-scale across which evolvability is considered. With ground-truth distance of distance from a individual to all behaviors, it also becomes possible to calculate a fanciful idealized metric of evolvability, *everywhere evolvability*: the average distance to everywhere, i.e. how many mutations are required on average to reach any behavior. Figure 5 shows graphically how these metrics correlate with distance to the solution in the medium maze (results are similar in the hard maze), while figure 6 demonstrates the intuitive notion that in both mazes longer-scale evolvability highly correlates with being near to a solution.

A final idealized measure is *behavioral rarity*, i.e. the proportion of genotypes in the search space that yield a particular behavior when evaluated. Some behaviors (such as not moving at all) may be common, while others (such as successful navigation through the maze) are objectively rare within the search space. Such rarity is a concept closely adjacent to behavioral novelty, i.e. the reward scheme in novelty search. In particular, behavioral novelty is rarity *relative* to what has been previously observed in a particular search.

This relation is interestingly deep, in that it has been previously shown that novelty search does indeed uncover objectively rare behaviors [27], e.g. the complex functionalities needed to perform non-trivial tasks; yet novelty search does not directly optimize a heuristic of absolute rarity itself, which could prove deceptive (i.e. the gradient of behavioral rarity may not be well-behaved). Instead, through feedback, novelty search can avoid such deception, as repeatedly visiting behaviors in novelty search gradually reduces their reward. In some sense, novelty search may approximately follow many divergent gradients of increasing rarity, exhausting one line when rarity gradients lead to a local optimum, staying until novelty is exhausted. For this reason, understanding the structure of rare behaviors may be useful to understanding or improving non-objective search algorithms like novelty search. While previous work has attempted to estimate objective rarity [27], here we can calculate it exactly through simple queries of the precomputed database. Figure 7 shows the distribution of behavior density in both mazes. The next section applies these metrics to instrument search, and to drive it.

4.3 Driving and Instrumenting Search through Ideal Measures

One advantage of precomputed domains is that expensive and ideal measures can also be precomputed, and then can efficiently either instrument search (e.g. does novelty search encourage to-everywhere evolvability?) or drive search (e.g. does directly optimizing behavioral rarity itself instantiate an effective search algorithm?). While many possible permutations of measures and drives could be explored within this framework (this diversity of experimental possibilities is a keystone of the value that it provides), this section shows only a few examples to highlight its potential.

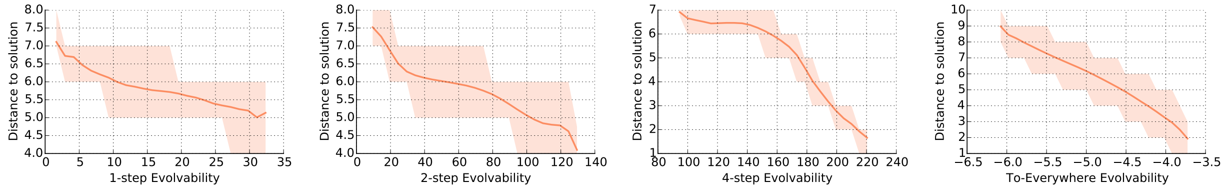
First, search algorithms are explored that are driven by the measures described in the previous section. Behavioral rarity, exact k-step evolvability, and everywhere evolvability are calculated for each genotype, and are then used as incentives to drive the same simple evolutionary algorithm applied to validate the precomputed domain. Driving search by directly incentivizing measures of evolvability can be seen as alternative instantiations of evolvability search [17], while driving search through rarity has some relation to work on quantifying impressiveness [27]. How successful such methods are at evolving solutions is shown in figure 8; reflecting its ideal characteristic and strong correlation with solution distance, searching for everywhere evolvability solves both tasks quickly, as does optimizing 4-step evolvability. As evolvability is considered within smaller mutational neighborhoods, its success rate declines, suggesting that efficient approximations of longer-range evolvability could increase the potential of the evolvability search method, which maximizes an estimate of 1-step evolvability. Rarity search is less consistently successful, although it outperforms objective-based search and is competitive with novelty search in the hard maze; preliminary follow-up experiments (and instrumentation results discussed next) support the intuitive hypothesis that rarity search can converge to behaviors that are exceedingly rare but that do not solve the task.

A final experimental exploration instruments search algorithms by two of the ideal metrics, i.e. behavioral rarity and everywhere evolvability. The idea is to explore how quickly different search algorithms discover rare behaviors, and to probe whether previous results showing that novelty search encourages evolvability (as measured by heuristic estimates of 1-step evolvability) [15, 16] generalize to an ideal measure of evolvability. Fifty runs are conducted for each approach. Figure 9a instruments search with rarity, and echoes the result of Lehman and Stanley [15] where novelty search quickly discovers rare behavior; it also suggests support for the hypothesis that there is a strong conceptual connection between novelty and rarity (given that both algorithms demonstrate similar performance by this metric). Figure 9b instruments search with everywhere evolvability, and supports the case that novelty search may encourage holistic evolvability that is not specific to the 1-step heuristic measures used in the past.

5 DISCUSSION

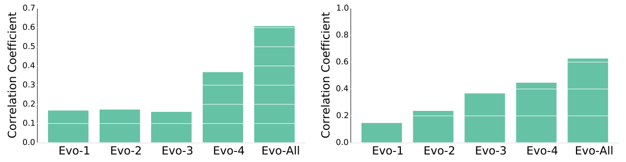
The results show the promise of precomputed domains to help explore costly hypotheses. For example, calculating ideal evolvability metrics such as the average distance to everywhere can reveal interesting properties of search spaces, and can aid researchers in attempts to find tractable approximations of them, and to investigate how well common search algorithms align with such metrics. It also enables exploring compelling (if unrealistic) best-case scenarios, such as whether directly incentivizing multi-step evolvability would indeed lead to effective search, which might motivate trying to adjust current algorithms such that they somehow better-align with the hard-to-compute metric (e.g. perhaps MCTS-like roll-outs [28] of mutated genotypes can provide approximate estimates of multi-step evolvability).

Additionally, having a true measure of how far a given individual is to a goal behavior enables direct observation of deception, i.e. when increasing fitness demonstrably moves a population further away in the search space from any solution individuals. In this way, having the true connectivity of the space allows for a deeper understanding of what assumptions certain search algorithms exploit. For example, the experiments with rarity search hint at the potential importance of rarity gradients for novelty search, and at a potentially interesting algorithm (e.g. driving search through a direct estimate of behavioral rarity). Precomputed domains could easily be adapted for multiobjective optimization or quality diversity algorithms [29],



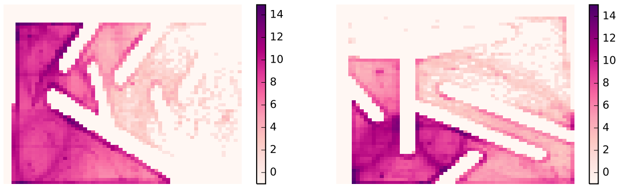
(a) 1-step (b) 2-step (b) 4-step (b) Everywhere

Figure 5: Generalized Evolvability Measures in the Medium Maze. The relationship between solution distance and (a) 1-step, (b) 2-step, (c) 4-step, and (d) everywhere evolvability is shown for the medium maze. The solid line indicates the mean solution distance, and the red fill spans the top and bottom quartiles. The conclusion is that across all evolvability measures, increasing evolvability decreases distance to a solution.



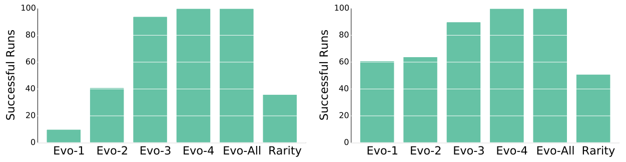
(a) Medium Maze (b) Hard Maze

Figure 6: Correlation Coefficients between Evolvability and Solution Distance. The negation of the Pearson correlation coefficient between evolvability measures and solution distance is shown for the (a) Medium Maze and (b) Hard Maze (e.g. higher means that increased evolvability is associated with being genotypically nearer to a solution). The Evo-k label indicates k-step evolvability, while Evo-All indicates Everywhere evolvability. All measures demonstrate relatively strong correlation, and in general correlation increases with the size of the mutational neighborhood considered. The conclusion is that considering evolvability over longer timescales may provide stronger signal about an individual's potential.



(a) Medium Maze (b) Hard Maze

Figure 7: Rarity of Behaviors in the Maze Domain. How rare behaviors are in the enumerated search space is shown for the (a) Medium Maze and (b) Hard Maze. The coloration of a point indicates how many individuals instantiate that behavior. The scale is logarithmic, i.e. 12 indicates e^{12}, or approximately $160,000$ individuals. There are 43 million individuals in total. The conclusion is that behaviors requiring more complex functionality tend to be rarer.



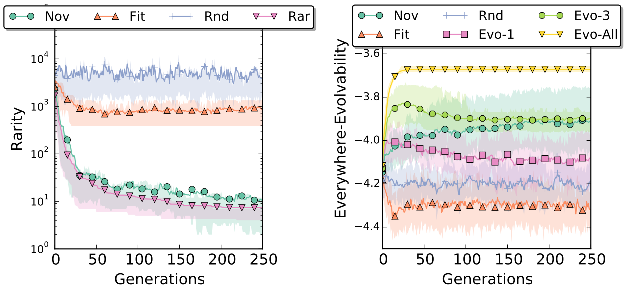
(a) Medium Maze (b) Hard Maze

Figure 8: Driving Search through Ideal Measures. The number of successful runs out of 100 is shown for variations of evolvability search and rarity search in the (a) Medium Maze and (b) Hard Maze. Longer-term evolvability measures ($k \geq 3$ and Everywhere evolvability) are never statistically outperformed, but interestingly, rarity search performs as well as novelty search in the Hard Maze (Fisher's exact test). The conclusion is that optimizing or encouraging longer-term notions of evolvability may be useful, and that rarity search may be an interesting algorithm to study further.

or to investigate the importance of population-level evolvability [30] (the focus in this paper was on individual-level evolvability).

While precomputed domains show promise for helping experimenters develop intuitions and quickly test new ideas, the trade-off they strike imposes strong limitations by necessity. For example, the discreteness of its search space may artificially inflate ruggedness. Further, an enumerated space can simulate the variable-length encoding of models like NEAT in only a limited way, e.g. by initializing search with zero weights for all connections beyond a minimal connectivity, and allowing zero weights to mutate to non-zero only with special topology-altering mutations.

A further problem is that the current implementation stores only certain aspects of simulated behavior (e.g. only the ending point of the robot at the end of simulation), which limits what fitness functions and behavior characterizations experimenters can easily implement without re-running the precomputation process. Keeping only limited parts of simulated behavior reflects a deliberate engineering choice, one which enables the precomputed search-space database to fit entirely in RAM for a modern computer; however, more complete behavior traces could be precomputed and stored in a disk-based database, enabling a wider range of fitness functions and behavioral properties, at the cost of slower evaluation. This undertaking is left to future work, and would provide most value



(a) Rarity

(b) Everywhere Evolvability

Figure 9: Instrumenting Search through Ideal Measures. Instrumentation of evolution over generations by (a) behavioral rarity (lower is more rare), and (b) everywhere evolvability (higher means more evolvable), is shown for experiments in the Hard Maze. In both plots, *Rnd* is random search, *Nov* is novelty search, *Evo-k* indicates evolvability search with k-step evolvability, *Evo-All* indicates everywhere evolvability, and *Rar* indicates rarity search. Both instrumentations record the score of the most rare (e.g. lowest occurrence) or most evolvable individual in the population. The solid lines indicate the mean value across the 50 independent runs, while the filled-in areas include the lowest and highest quartiles. The conclusion is that seeking novelty has some connection to seeking rarity, and that novelty search encourages everywhere evolvability.

when domain evaluations are very expensive, which would make re-running the precomputing process particularly undesirable.

While the current implementation is open-source and the precomputed database for the maze domain is available for experimentation (http://goo.gl/JTSTGs), the value of the approach would be increased with more domains, enabling probing the generality of the hypotheses explored here. Future work aims to release other domains, e.g. a version of biped locomotion simulation that has previously been explored with NEAT [12].

6 CONCLUSIONS

This paper introduced precomputed domains as a principled intermediate between theoretical models and the full complexity of modern encodings and domains. By limiting the encoding in a well-motivated way and precomputing all individuals, the benefit is ground-truth and extremely fast runs. The conclusion is that precomputed domains provide an interesting experimental playground for developing intuitions and testing hypotheses about complex search spaces, especially in areas such as non-objective search and evolutionary robotics, where evaluation is expensive, and the field is young enough that the space of possible search algorithms likely remains relatively unexplored. The current implementation is open-sourced and available for download; ideally it will serve as an extensible framework for other precomputed domains that could be shared among researchers, enabling easy exploration of ideas across domains and the possibility of performing meaningful research without access to large-scale computational resources.

REFERENCES

[1] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.

[2] Wolfgang Banzhaf, Bert Baumgaertner, Guillaume Beslon, René Doursat, James A Foster, Barry McMullin, Vinicius Veloso De Melo, Thomas Miconi, Lee Spector, Susan Stepney, et al. Defining and simulating open-ended novelty: requirements, guidelines, and challenges. *Theory in Biosciences*, 135(3):131–161, 2016.

[3] Peter Bentley and David Corne. *Creative evolutionary systems.* Morgan Kaufmann, 2002.

[4] Terry Jones and Stephanie Forrest. Fitness distance correlation as a measure of problem didculty for genetic algorithms. 1995.

[5] Stuart Kauffman and Simon Levin. Towards a general theory of adaptive walks on rugged landscapes. *Journal of theoretical Biology*, 128(1):11–45, 1987.

[6] David E Goldberg. Simple genetic algorithms and the minimal, deceptive problem. *Genetic algorithms and simulated annealing*, 74:88, 1987.

[7] Melanie Mitchell, Stephanie Forrest, and John H Holland. The royal road for genetic algorithms: Fitness landscapes and ga performance. In *Proceedings of the first european conference on artificial life*, pages 245–254, 1992.

[8] Daniel Hein, Manfred Hild, and Ralf Berger. Evolution of biped walking using neural oscillators and physical simulation. In *RoboCup 2007: Proceedings of the International Symposium*, LNAI. Springer, 2007.

[9] Nahum Zaera, Dave Cliff, et al. Not) evolving collective behaviours in synthetic fish. In *In Proceedings of International Conference on the Simulation of Adaptive Behavior.* Citeseer, 1996.

[10] Brian G Woolley and Kenneth O Stanley. Exploring promising stepping stones by combining novelty search with interactive evolution. *arXiv preprint arXiv:1207.6682*, 2012.

[11] Richard Dawkins. The evolution of evolvability. *On growth, form and computers*, pages 239–255, 2003.

[12] Joel Lehman and Kenneth O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223, 2011.

[13] Jorge Gomes, Paulo Urbano, and Anders Lyhne Christensen. Evolution of swarm robotics systems with novelty search. *Swarm Intelligence*, 7(2-3):115–144, 2013.

[14] Heather J Goldsby and Betty HC Cheng. Automatically discovering properties that specify the latent behavior of uml models. In *International Conference on Model Driven Engineering Languages and Systems*, pages 316–330. Springer, 2010.

[15] Joel Lehman and Kenneth O. Stanley. Improving evolvability through novelty search and self-adaptation. In *2011 IEEE Congress on Evolutionary Computation (CEC)*, pages 2693–2700. IEEE, 2011.

[16] Joel Lehman and Risto Miikkulainen. Enhancing divergent search through extinction events. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2015)*, Madrid, Spain, 2015.

[17] Henok Mengistu, Joel Lehman, and Jeff Clune. Evolvability search: Directly selecting for evolvability in order to study and produce it. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2016).* ACM, 2016.

[18] M. Pigliucci. Is evolvability evolvable? *Nature Reviews Genetics*, 9(1):75–82, 2008.

[19] J.F.Y Brookfield. Evolution: The evolvability enigma. *Current Biology*, 11(3):R106 – R108, 2001.

[20] G.P. Wagner and L. Altenberg. Complex adaptations and the evolution of evolvability. *Evolution*, 50(3):967–976, 1996.

[21] M.L. Dichtel-Danjoy and M.A. Félix. Phenotypic neighborhood and microevolvability. *Trends in Genetics*, 20(5):268–276, 2004.

[22] M. Kirschner and J. Gerhart. Evolvability. *Proceedings of the National Academy of Sciences of the United States of America*, 95(15):8420, 1998.

[23] Joel Lehman and Kenneth O. Stanley. Evolvability is inevitable: Increasing evolvability without the pressure to adapt. *PLoS ONE*, 2013.

[24] Stephane Doncieux and Jean-Baptiste Mouret. Beyond black-box optimization: a review of selective pressures for evolutionary robotics. *Evolutionary Intelligence*, 7(2):71–93, 2014.

[25] J-B Mouret and Stéphane Doncieux. Encouraging behavioral diversity in evolutionary robotics: An empirical study. *Evolutionary computation*, 20(1):91–133, 2012.

[26] Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127, 2002.

[27] Joel Lehman and Kenneth O. Stanley. Beyond open-endedness: Quantifying impressiveness. In *Proceedings of Artificial Life Thirteen (ALIFE XIII)*, 2012.

[28] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

[29] Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40, 2016.

[30] Bryan Wilder and Kenneth Stanley. Reconciling explanations for the evolution of evolvability. *Adaptive Behavior*, 23(3):171–179, 2015.