Analyzing Deception, Evolvability, and Behavioral Rarity in Evolutionary Robotics

Joel Lehman
IT University of Copenhagen
Copenhagen, Denmark

ABSTRACT

A common aim across evolutionary search is to skillfully navigate complex search spaces, which requires search algorithms that exploit search space structure. This paper focuses on evolutionary robotics (ER) in particular, wherein controllers for robots are evolved to produce complex behavior. One productive approach for probing search space structure is to analyze properties of fitness landscapes; however, this paper argues that ER may require a fresh perspective for landscape analysis, because ER often goes beyond the black-box setting, i.e. evaluations provide useful information about how robots *behave*, beyond scalar performance heuristics. Indeed, some ER algorithms explicitly exploit such behavioral information, e.g. to follow gradients of behavioral novelty rather than to climb gradients of increasing performance. Thus well-motivated behavior-aware metrics may aid probing search-space structure in ER. In particular, this paper argues that behavioral conceptions of *deception*, *evolvability*, and *rarity* may help to understand ER landscapes, and seeks to quantify and explore them within a common ER benchmark task. To help this investigation, an expressive but limited encoding is designed, such that the behavior of all possible individuals in the domain can be precomputed. The result is an efficient platform for experimentation that facilitates (1) probing exact quantifications of deception, evolvability, and rarity in the chosen domain, and (2) the ability to efficiently *drive* search through idealistic ground-truth measures. The results help develop intuitions and suggest possible new ER algorithms. The hope is that the extensible open-source 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

GECCO '17 Companion, Berlin, Germany

978-1-4503-4939-0/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3067695.3082514

ACM Reference format:
Joel Lehman. 2017. Analyzing Deception, Evolvability, and Behavioral Rarity in Evolutionary Robotics. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3067695.3082514

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 a single optimal solution, but other paradigms include accumulating points spanning trade-offs among competing objectives (multiobjective optimization) or collecting a diverse set of individuals that instantiate a wide variety of interesting and innovative behaviors (e.g. as in open-ended evolution or computational creativity). Across nearly all such use cases, a primary challenge is to discover what combination of search algorithm and search heuristic will be effective. Important to this challenge is understanding properties of the experimental domain and of the genetic encoding, which when combined instantiate high-dimensional landscapes of fitness; or similarly, an expansive network of phenotypic behaviors, connected through genotypic mutations. Discovering important properties of landscapes, such as deception [1–3] or ruggedness [4] can catalyze creating new algorithms which better exploit their properties.

This paper focuses on a particular subfield of EC called evolutionary robotics (ER), wherein controllers for robots are evolved, often with the objective of producing complex and functional behavior. While landscape analysis has been effectively applied across EC as a whole, ER may provide unique opportunities for analyzing search space structure [5]. One opportunity in ER is that experimenters can often exploit information *beyond* only scalar fitness values concerning the *behavior* of a robot acting in a domain [6]. Such behavioral information (e.g. a description of what the robot did) enables new degrees of freedom for analyzing landscapes or networks of behaviors, which may well-inform ER algorithm design. A further opportunity is that within ER interest is is growing around algorithms that *accumulate* a diversity of interesting or high-quality solutions [1, 7, 8], as opposed to seeking one optimal one. This distinct focus suggests that a specialized set of search space features may be important to understanding how to effectively design such diversity-seeking algorithms.

In particular, this paper investigates the fitness and behavioral landscapes of a popular benchmark task for *novelty search*, an algorithm often applied in ER and artificial life, that explicitly searches for novel behaviors, instead of for an optimal solution. Interestingly, in deceptive problems, searching for behavioral diversity alone often more effectively leads to evolving solutions than does searching directly for high-quality solutions [1–3]. Hypotheses

about what types of problems and domains are aided by novelty search, and how novelty search functions in practice, often involve appeals to properties like *deception* [1], *evolvability* [9, 10], and *behavioral rarity* [11], although rarely together in the same work. One contribution of this paper is to analyze all of these properties in a common computationally-tractable framework (which is open-sourced with this paper), which can be flexibly extended to include additional ER domains.

The approach is to precompute the *behaviors* of all individuals (e.g. not just their fitnesses) in a large-but-tractable ER search space, evaluated within the benchmark simulator. 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, or how well a particular fitness measure correlates with actual genomic distance to a goal behavior. The hope is not to solve this particular benchmark domain more effectively, but instead to dive into interesting properties of representative ER search spaces, and also to create an experimental playground that can be useful for quickly generating and testing ideas, and building experimenter intuition.

In particular, this paper pairs a maze navigation simulator with a discretized neuroevolution encoding, and caches the results of evaluating all individuals in a database, enabling cheap future evaluation. The resulting search space contains over 30 million artificial neural networks (ANNs), and shares important qualitative characteristics with the canonical domain from which it is inspired. This enumerated search space allows efficient calculation of intractable (and novel) generalizations of evolvability and the exact distribution of specific behaviors (e.g. solutions) within the space. Enough runs to enable statistical significance can be conducted in minutes on one computer (which would otherwise take days or sometimes weeks of computation), enabling quick iteration. Furthermore, highly expensive algorithms (e.g. using evolvability itself as a search heuristic), or impractical ones (e.g. driving search by how objectively rare a behavior is within the space), can be easily implemented and efficiently run, to probe intuitions in a tractable way.

Results affirm that the properties of evolvability, deception, and rarity are important features of this particular ER search space, and new directions for possible algorithms are suggested. The conclusion is that landscapes in ER may have intriguing and relatively unexplored features, and that creating metrics and analyses crafted to ER may result in insights that may guide development of new algorithms.

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 produce unfavorable landscapes for EAs [4, 12, 13], like deception [13] or ruggedness [4]. 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 [4], or constructed bitwise models such as the royal road function [14] or the trap function [13].

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 [15]), or adjust the fitness function through iterations of experiments followed by changes aimed at remedying problematic dynamics [16]. 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 [17, 18].

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 relative groundedness to real problems, thereby enabling fast and flexible idea-generation and investigation.

2.2 Novelty Search

Novelty search is inspired by natural evolution's drive towards novelty, and rewards novel behavior directly *instead* of progress towards a fixed objective [1]. 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

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

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 [1–3]. 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 [1, 9]. In particular, the connection weights of a fixed-topology ANN are evolved; the setup is later described in more detail.

2.3 Evolvability in ER

Natural evolution has produced flexible, highly evolvable representations that facilitate its prolific discovery of diverse organisms; yet this fluid evolvability is often lacking in EC and ER [9]. Thus metrics for exploring evolvability, or methods to directly search for it, are important, because they can help isolate how it is distributed in the search space, which might reveal algorithmic or encoding changes that encourage it.

While there is no overall consensus on evolvability's definition or its measurement [20], one common conception is to consider evolvability as an organism's phenotypic variability [21–24]; 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 [20]. This conception (of evolvability as phenotypic variability) aligns well with the motivation of novelty search, and is adopted here to help explore hypotheses about ER search spaces, in a way similar to previous related studies [9, 25].

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 [9, 10, 25]. That is, the measure attempts to gauge an individual's phenotypic connectivity. However, such measures are expensive because they depend on evaluating the behaviors instantiated by many perturbations of an individual's genome [10]. As a result, calculating evolvability exactly, or considering it over longer evolutionary timescales, is rarely considered. However, the approach here allows tractable explorations with evolvability metrics by exploiting precomputation.

3 PRECOMPUTED DOMAINS

The main idea is to *precompute* the simulated behavior of all possible genotypes in full-fledged domains, leading to evaluation as a lookup table (e.g. as in some experiments in [25]). 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 previously impractical hypotheses become amenable to direct investigation. Note that this idea of precomputed domains is not unique to this paper, the contribution is more specific to its application of studying ER landscapes from a particular angle, and of the uniqueness of the resulting analysis.

3.1 Design Constraints

Precomputing the behavior of all possible genotypes is not generally possible, because most search spaces are impractically large, i.e. 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 relatively 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 maximum-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

3.2 Implementation

The released implementation of precomputed domains consists of: (1) separate in-memory look-up tables for each precalculated property, e.g. rarity, evolvability, and fitness; (2) a mechanism for indexing into such tables given an individual, i.e. mapping a particular genome into a scalar index; (3) a mechanism for enumerating the mutational neighbors of an individual, i.e. what is the connectivity of the search space; and (4) a metric of distance between genomes. These rough tools provide an interface for implementing landscape analyses over the entire search space, and for implementing search algorithms driven to explore within the space.

Given the same notation as in the previous section, a genome is a list of G genes with integer values 0 through $A-1$. We can treat this list as a A-ary number to calculate its scalar index. For example,

given two genes with three alleles, the genome 21 interpreted as a base three number yields an index of 7 in decimal. The mutation operator adopted here perturbs one allele of the genome to any of its A values. The resulting mutuational neighborhood of an individual thus contains all genomes different in one allele, and leads to a genomic distance metric that counts in how many alleles two genomes differ.

3.3 Precalculating Landscape Properties

The next sections motivate the particular properties of interest (e.g. deception, rarity, and evolvability) and describe how they are are efficiently calculated.

3.3.1 Calculating Deception. Deception is the idea that sometimes following the gradient of the heuristic guiding search can actively lead search *away* from a solution. It provides an important motivation for diversity-driven algorithms in ER; the main idea is that as problems grow more ambitious, the heuristic of goal-oriented fitness becomes increasingly deceptive [26]. Diversity-driven algorithms can avoid this pathology by exploiting *behavioral information*. Thus measuring deception is useful to validate that domains in which novelty search succeeds are indeed deceptive.

The quantitative measure of deception adopted here, called fitness distance correlation (FDC; [12]), 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. 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).

3.3.2 Calculating Evolvability. Evolvability is a desirable property for ER, because greater evolvability means a greater range of variation for evolution to select from. Previous work has argued that diversity-driven algorithms encourage greater evolvability than traditional goal-oriented EAs [9, 10]. Thus it is useful to probe whether this is robust across a range of different evolvability metrics.

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 individualfis offspring [9, 10, 25]. To calculate this quantity exactly (i.e. using the entire mutational neighborhood and not a random sample), behaviors are first discretized, by superimposing a regular grid over the space of possible behaviors, where all behaviors contained by a grid square are considered the same. All individuals in the search space can then be mapped into such distinct behavior bins. Next, for each distinct behavior, from each individual the minimal-distance to another individual demonstrating that behavior is calculated, using the same iterative depth-first search procedure above. Thus the products are look-up tables that store the minimum number of mutations needed for any given individual to demonstrate any given behavior.

This approach enables easily calculating *generalizations* of the 1-step (e.g. the mutational neighborhood considering 1 mutation) evolvability measure used for efficiency reasons. E.g. k-step evolvability is calculated 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. Finally, the same tables enable calculating a highly idealized metric of evolvability, *everywhere evolvability*: the average distance to everywhere, i.e. how many mutations are required on average to reach any behavior, which is a novel contribution of this paper.

3.3.3 Calculating Rarity. Behavioral rarity, i.e. the proportion of genotypes in the search space that yield a particular behavior when evaluated, is also an intriguing property in ER. The motivation is that rarity is concept closely adjacent to behavioral novelty, i.e. the reward scheme in novelty search. Note that behavioral novelty is rarity *relative* to what has been previously observed in a particular search. One hypothesis is that 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 diversity-driven algorithms like novelty search. While previous work has attempted to estimate behavioral rarity [11], here we can calculate it exactly through simple queries of the precomputed database. That is, once behaviors are discretized as above, the counts of each distinct behavior can be easily summed.

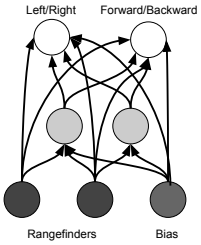
4 PRECOMPUTED MAZE NAVIGATION

This paper adopts a common maze-navigation domain benchmark that is often used to evaluate diversity-driven search algorithms such as novelty search, behavioral diversity, and MAP-ELITES [1, 7, 10, 19].

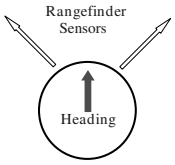
4.1 Domain Details

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 [1, 7]. For measuring evolvability, each grid square within a regular grid ($20x20$) 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 [1].

This domain's canonical setup uses the NEAT neuroevolution encoding [27], 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 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

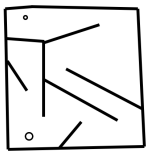


(a) Neural Network

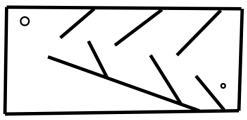


(b) Sensors

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 [1] 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.

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 are 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.

The resulting encoding consists of 16 connections that can each take on 3 distinct weight values, realizing a search space with 3^{16} individuals (34 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 laptop, 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.

4.2 Validating the Precomputed Domain

In contrast to the original NEAT setup, the precomputed encoding is discretized, motivating validation experiments to probe whether qualitative similarity is preserved. To do so, in similar experiments to the domain's introduction [1], 100 runs each of objective-based search, novelty search, and 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.

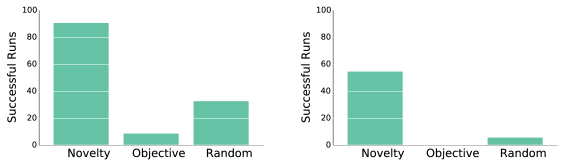
The results are shown in figure 3 for both mazes. Novelty search significantly out-performs the other methods on both mazes, while objective-based search performs worse than random search in both domains (Fisher's exact test; $p < 0.05$). A divergence from results in the canonical (i.e. non-precomputed) domain from [1] is that, there, objective-driven fitness search often does solve the medium maze, although its performance is worse than novelty search, as it is also here. Follow-up experiments in the precomputed domain 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 removing pie-slice sensors entirely and reducing the number of rangefinders makes the problem more difficult in general. Qualitative behavior of evolved solutions is roughly consistent with previous results. From a high-level however, the results are broadly consistent: the medium maze is easier for all methods than is the hard maze, and novelty search outperforms the competing methods in both domains. 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 can serve as a useful, although not perfect, proxy.

5 RESULTS

The next sections present analyses of fitness and behavioral landscapes, as well as novel search heuristics driven by precalculated quantities.

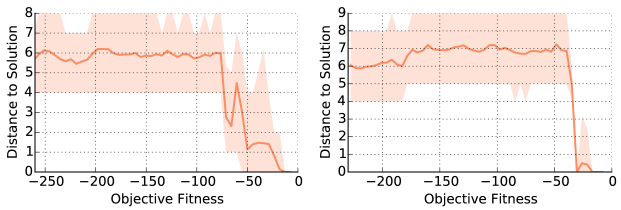
5.1 Exact Quantification of Deception, Evolvability, and Rarity

Interestingly, solutions to either maze are very rare within the search space; only 320 solutions to the medium maze, and 59 solutions to the hard maze exist within the 34 million total individuals in each maze. How objective-based fitness and distance to solution



(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 maze domain, and (b) the precomputed hard maze 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.



(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.

correlate in both mazes is shown in figure 4. FDC (the chosen measure of deception), 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. Results are very similar when using Kendall's Tau, a correlation metric that does not assume linearity.

One intuitive expectation of evolvability is that increasingly evolvable individuals will on average tend to be closer to solutions than less evolvable ones. Figure 5 probes this intuition graphically, across a range of evolvability metrics, showing how increasing evolvability correlates 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 time-scale evolvability highly correlates with being near to a solution.

Intuitively, behavioral rarity would be expected to be higher for behaviors requiring more complicated navigation. Indeed, figure 7 validates this intuition by showing the distribution of behavior density in both mazes.

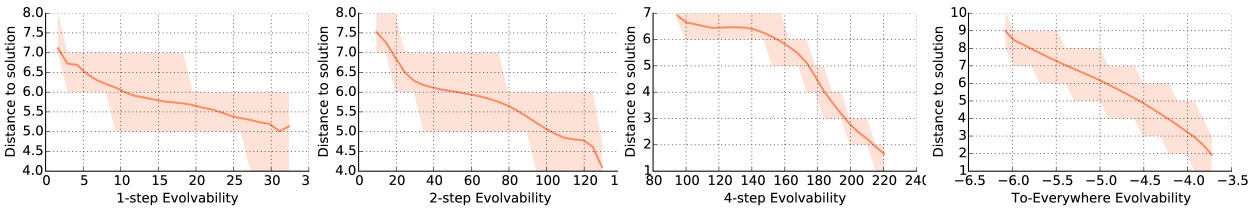
5.2 Driving and Instrumenting Search through Ideal Measures

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 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 (indeed, 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 are instantiations of evolvability search [10], while driving search through rarity has some relation to work on quantifying impressiveness [11].

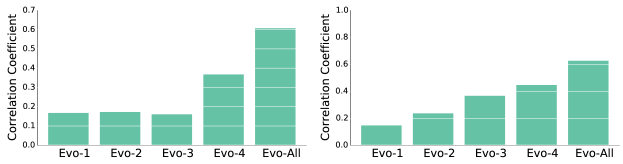
How successful such methods are at evolving solutions is shown in figure 8; reflecting its ideal characteristic and strong correlation with solution distance, searching for everywhere evolvability solves both tasks quickly, as does optimizing 4-step evolvability. As evolvability is considered within 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 yet 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) [9, 19] 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 [9] 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, i.e. increased evolvability is not specific to the 1-step heuristic measures used in the past.



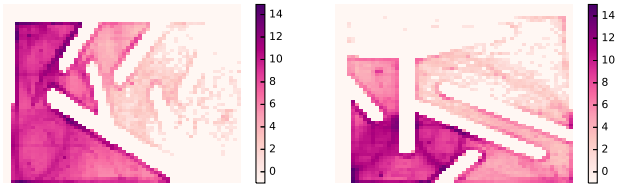
(a) 1-step (b) 2-step (b) 4-step (b) Everywhere

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.



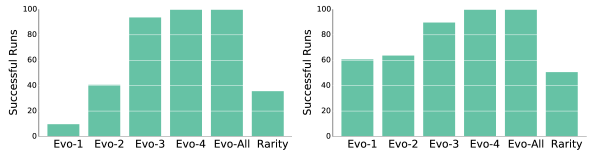
(a) Medium Maze (b) Hard Maze

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 provides 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 34 million individuals in total. The conclusion is that behaviors requiring more complex functionality tend to be rarer.



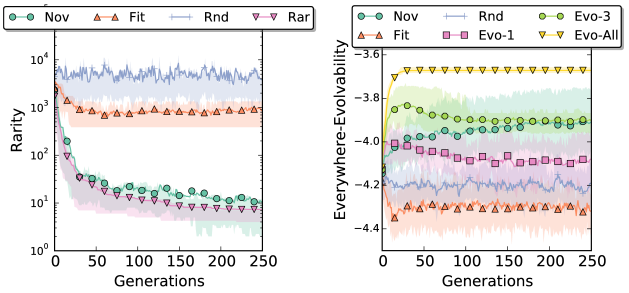
(a) Medium Maze (b) Hard Maze

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 for further study.

6 DISCUSSION

The results show the promise of precomputed domains to investigate properties of ER landscapes and to 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 ground-truth 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 how well the assumptions of ER algorithms hold up in practice. For example, the experiments



(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 is qualitatively connected to seeking rarity, and that novelty search encourages everywhere evolvability.

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

The current implementation is open-source and the precomputed database for the maze domain is available for experimentation (https://github.com/jal278/precomputed_er). Future work aims to release other domains, e.g. a version of biped locomotion simulation that has previously been explored with NEAT [1], to investigate the generality of the results presented here.

7 CONCLUSIONS

This paper applied precomputed domains as a means to explore fitness and behavioral landscapes in ER. 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 diversity-driven search and ER, where evaluation is expensive, and the field is young enough that the space of possible search algorithms likely remains relatively unexplored.

REFERENCES

[1] Joel Lehman and Kenneth O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223, 2011.
[2] Jorge Gomes, Paulo Urbano, and Anders Lyhne Christensen. Evolution of swarm robotics systems with novelty search. *Swarm Intelligence*, 7(2-3):115–144, 2013.
[3] 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.
[4] Stuart Kauffman and Simon Levin. Towards a general theory of adaptive walks on rugged landscapes. *Journal of theoretical Biology*, 128(1):11–45, 1987.
[5] Andrew L Nelson, Gregory J Barlow, and Lefteris Doitsidis. Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Systems*, 57(4):345–370, 2009.
[6] 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.
[7] J-B Mouret and Stéphane Doncieux. Encouraging behavioral diversity in evolutionary robotics: An empirical study. *Evolutionary computation*, 20(1):91–133, 2012.
[8] Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.
[9] 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.
[10] 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.
[11] Joel Lehman and Kenneth O. Stanley. Beyond open-endedness: Quantifying impressiveness. In *Proceedings of Artificial Life Thirteen (ALIFE XIII)*, 2012.
[12] Terry Jones and Stephanie Forrest. Fitness distance correlation as a measure of problem didculty for genetic algorithms. 1995.
[13] David E Goldberg. Simple genetic algorithms and the minimal, deceptive problem. *Genetic algorithms and simulated annealing*, 74:88, 1987.
[14] 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.
[15] 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.
[16] 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.
[17] Brian G Woolley and Kenneth O Stanley. Exploring promising stepping stones by combining novelty search with interactive evolution. *arXiv preprint arXiv:1207.6682*, 2012.
[18] Richard Dawkins. The evolution of evolvability. *On growth, form and computers*, pages 239–255, 2003.
[19] 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.
[20] M. Pigliucci. Is evolvability evolvable? *Nature Reviews Genetics*, 9(1):75–82, 2008.
[21] J.F.Y Brookfield. Evolution: The evolvability enigma. *Current Biology*, 11(3):R106 – R108, 2001.
[22] G.P. Wagner and L. Altenberg. Complex adaptations and the evolution of evolvability. *Evolution*, 50(3):967–976, 1996.
[23] M.L. Dichtel-Danjoy and M.A. Félix. Phenotypic neighborhood and micro-evolvability. *Trends in Genetics*, 20(5):268–276, 2004.
[24] M. Kirschner and J. Gerhart. Evolvability. *Proceedings of the National Academy of Sciences of the United States of America*, 95(15):8420, 1998.
[25] Joel Lehman and Kenneth O. Stanley. Evolvability is inevitable: Increasing evolvability without the pressure to adapt. *PLoS ONE*, 2013.
[26] Joel Lehman and Kenneth O. Stanley. Novelty seach and the problem with objectives. In *Genetic Programming in Theory and Practice IX (GPTP 2011)*, chapter 3, pages 37–56. Springer, 2011.
[27] Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127, 2002.
[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.