EVOLUTION THROUGH THE SEARCH FOR NOVELTY

by

JOEL LEHMAN
B.S. Ohio State University, 2007

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2012

Major Professor: Kenneth O. Stanley

ABSTRACT

I present a new approach to evolutionary search called *novelty search*, wherein *only* behavioral novelty is rewarded, thereby abstracting evolution as a search for novel forms. This new approach contrasts with the traditional approach of rewarding progress towards the objective through an *objective function*. Although they are designed to light a path to the objective, objective functions can instead *deceive* search into converging to dead ends called *local optima*.

As a significant problem in evolutionary computation, deception has inspired many techniques designed to mitigate it. However, nearly all such methods are still ultimately susceptible to deceptive local optima because they still measure progress with respect to the objective, which this dissertation will show is often a broken compass. Furthermore, although novelty search completely abandons the objective, it counterintuitively often *outperforms* methods that search directly for the objective in deceptive tasks and can induce evolutionary dynamics closer in spirit to natural evolution. The main contributions are to (1) introduce novelty search, an example of an effective search method that is not guided by actively measuring or encouraging objective progress; (2) validate novelty search by applying it to biped locomotion; (3) demonstrate novelty search's benefits for evolvability (i.e. the ability of an organism to further evolve) in a variety of domains; (4) introduce an extension of novelty search called minimal criteria novelty search that brings a new abstraction of natural evolution to evolutionary computation (i.e. evolution as a search for many ways of meeting the minimal criteria of life); (5) present a second extension of novelty search called novelty search with local competition that abstracts evolution instead as a process driven towards diversity with competition playing a *subservient* role; and (6) evolve a diversity of functional virtual creatures in a single run as a culminating application of novelty search with local competition. Overall these contributions establish novelty search as an important new research direction for the field of evolutionary computation.

To my parents

ACKNOWLEDGMENTS

Sincere thanks go to my advisor Dr. Kenneth O. Stanley for tireless patience, countless hours of challenging and productive discussions, and the many marked up drafts of papers that so shaped my ability to write coherently. Without him I would not be the researcher that I am today nor would this dissertation exist.

Thanks also go to my other committee members Dr. R. Paul Wiegand, Dr. Avelino Gonzalez, and Dr. Eric Hoffman. Their insightful feedback has made this a stronger work.

Perhaps most significantly, my parents surely deserve my thanks for putting up with me. I always have had their support and encouragement in all my endeavors. Who knows where I would be today without someone patiently looking over my shoulder as I typed in a BASIC program from 3-2-1 Contact magazine, or if I was not forgivingly allowed to constantly monopolize the family computer.

Thanks also are due to past and present members of the Evolutionary Complexity Research Group (Eplex) at University of Central Florida (http://eplex.cs.ucf.edu), our occasional visitors from afar, and honorary member Dr. Charles Bailey for his unwavering faith in our work and his generous financial support. In particular thanks go to Dr. Sebastian Risi and Dr. David D'Ambrosio for being sounding boards for fledgling ideas, providing insightful suggestions, and offering company during post-meeting lunches and coffee. Thanks also to Amy Hoover for leveling out our fearsome coffee mug gang and infusing music into our robot-centric world. To visiting post-docs Dr. Andrea Soltaggio and Dr. Ben Jones, your fresh perspectives and engaging conversations brought new color to the lab.

Special thanks to Russell Smith and the other developers that contributed to the Open Dynamics Engine, which provided the physics engine underlying the biped experiment in this dissertation. Likewise to Dr. Kenneth O. Stanley for implementing the NEAT C++ package used in many of the experiments. In the same vein, thanks to the developers of Linux and in general all the open-source developers that provided the high-quality free software that facilitated my experiments.

Thanks also to DARPA and ARO for funding the majority of this research through DARPA grants HR0011-09-1-0045 and N11AP20003 (Computer Science Study Group Phases 2, and 3), and US Army Research Office grant Award No. W911NF-11-1-0489. This dissertation does not necessarily reflect the position or policy of the government, and no official endorsement should be inferred.

Finally, if the research behind this dissertation has taught me anything on a personal level, it is the value of appreciating where you are for its own merits and the difficulty in predicting where a particular path will lead you. In that spirit, though I did not know it at the time, one of the best decisions I have made is moving to Orlando five years ago after being inspired by a book on game programming.

Joel Lehman

University of Central Florida

April 2012

TABLE OF CONTENTS

LIST OF FIGURES xiv

LIST OF TABLES xxv

CHAPTER 1 INTRODUCTION 1

1.1 Contributions 8

1.2 Outline 10

CHAPTER 2 BACKGROUND 11

2.1 Evolutionary Computation 11

2.2 Deception In Evolutionary Computation 14

2.2.1 Deception and Problem Difficulty 15

2.2.2 Mitigating Deception 20

2.3 Complexity in Natural Evolution 28

2.4 Open-ended Evolutionary Computation 31

2.5 NeuroEvolution of Augmenting Topologies (NEAT) 34

CHAPTER 3 APPROACH: NOVELTY SEARCH 37

3.1 The Search for Novelty 37

3.2 The Novelty Search Algorithm 41

CHAPTER 4 INITIAL MAZE EXPERIMENTS 46

4.1 Maze Experiment 46

4.2 Maze Results 51

4.2.1 Typical Behavior 53

4.3 Additional Maze Studies 53

4.3.1 Bounding the Size of the Archive 55

4.3.2 Conflating Behaviors in the Maze 56

4.4 Implications 66

CHAPTER 5 GENERALITY OF NOVELTY SEARCH 68

5.1 GP Experiments 68

5.1.1 GP Maze Experiment 70

5.1.2 Artificial Ant Experiment 72

5.1.3 GP Parameters 74

5.1.4 GP Results 76

5.1.5 GP Interpretation 77

5.2 Scaling Problem Complexity 80

5.2.1 Scaling Experiment 81

5.2.2 Scaling Results 83

5.3 Results from other Researchers 84

5.4 Implications 87

CHAPTER 6 EVOLVING BIPED WALKING GAITS WITH NOVELTY SEARCH 89

6.1 Biped Experiment 89

6.2 Biped Results 94

6.3 Implications 95

CHAPTER 7 NOVELTY SEARCH AND EVOLVABILITY 98

7.1 Evolvability in Natural Evolution and EC 99

7.2 Measuring Evolvability 101

7.3 Evolvability Experiments 103

7.3.1 Maze Experiment 103

7.3.2 Biped Experiment 104

7.4 Evolvability Results 105

7.5 Self-Adaptation Experiments 106

7.6 Self-Adaptation Results 111

7.7 Implications 113

CHAPTER 8 MINIMAL CRITERIA NOVELTY SEARCH 115

8.1 The Minimal Criteria Novelty Search Algorithm 115

8.2 MCNS Experiments 120

8.2.1 Maze Navigation Experiment 121

8.2.2 Two-point Navigation Experiment 123

8.2.3 MCNS Parameters 126

8.3 MCNS Results 127

8.4 Implications 128

CHAPTER 9 NOVELTY SEARCH WITH LOCAL COMPETITION 130

9.1 Evolving Virtual Creatures 130

9.1.1 Evolution of Robotic Organisms (ERO) 132

9.2 The NSLC Algorithm 134

9.2.1 Balancing Achievement with Novelty 136

9.2.2 Local Competition 138

9.3 Virtual Creatures Experiment 141

9.3.1 Virtual Creature Parameters 144

9.4 Virtual Creatures Results 145

9.5 Typical Run 149

9.6 Implications 151

CHAPTER 10 DISCUSSION 154

10.1 Abandoning Objectives 154

10.2 Novelty search and open-ended evolution 158

10.3 Alternate Abstractions of Natural Evolution 160

10.4 Future Work 163

CHAPTER 11 CONCLUSION 165

11.1 Contributions 165

11.2 Conclusion 167

APPENDIX: EXPERIMENTAL PARAMETERS 169

A.1 Parameters for NEAT-based experiments 170

A.2 Parameters for GP-based experiments 173

LIST OF REFERENCES 177

LIST OF FIGURES

Figure 4.1 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). Each arrow outside of the robot's body in (b) is a rangefinder sensor that indicates the distance to the closest obstacle in that direction. The robot has four pie-slice sensors that act as a compass towards the goal, activating when a line from the goal to the center of the robot falls within the pie-slice. The solid arrow indicates the robot's heading. 48

Figure 4.2 Maze Navigation Maps. In both maps, the large circle represents the starting position of the robot and the small circle represents the goal. Cul-de-sacs in both maps that lead toward the goal create the potential for deception. 49

Figure 4.3 Comparing Novelty Search to Fitness-based Search. The change in fitness over time (i.e. number of evaluations) is shown for NEAT with novelty search, fitness-based NEAT, and NEAT with random selection on the medium (a) and hard (b) maps, both averaged over 40 runs of each approach. The horizontal line indicates at what fitness the maze is solved. The main result is that novelty search is significantly more effective. Only the first 75,000 evaluations (out of 250,000) are shown because the dynamics remain stable after that point. 51

Figure 4.4 Final Points Visited Over Typical Runs. Each maze depicts a typical run, stopping at either 250,000 evaluations or when a solution is found. Each point represents the end location of a robot evaluated during the run. Novelty search is more evenly distributed because it is not deceived. 54

Figure 4.5 Increasing the Dimensionality of the Behavioral Characterization. The average number of evaluations taken to solve the task is shown for different numbers of samples of the navigator's location (i.e. different behavioral characterization dimensionalities) taken during an evaluation of NEAT with novelty search on the hard maze. For comparison, the baseline performance when the behavior characterization is only the final point visited is shown as a separate line. The dimensionality of the behavioral characterization is twice the number of samples because a location is composed of x and y coordinates. Each point represents the average of 40 runs. The main result is that even with a behavioral characterization with 200 samples (400 dimensions), the performance of novelty search is largely unaffected. The hard maze was solved in under $500,000$ evaluations in each of the 520 runs. 61

Figure 4.6 Comparing Conflation Across Different Behavioral Characterizations. Typical conflations of behavior are indicated in gray for reducing the amount of information in the behavioral characterization (a) and characterizing behavior by fitness (b). In (a) only geographically similar behaviors are conflated by a rectangle that is a part of a regular grid, whereas in (b), behaviors that end in very different locations (and are likely specified by very different policies) are conflated by the circle centered on the goal with radius equal to a particular fitness value. . 63

Figure 4.7 Reducing the Amount of Information in the Behavioral Characterization. The average number of evaluations taken to solve the hard maze is shown for behavioral characterizations that purposefully conflate behaviors that end in the same square of a virtual grid overlaid uniformly on the maze. Each point represents the average of 40 runs. The dotted line indicates the baseline performance at full resolution. The main result is that even with a behavioral characterization that only recognizes four behaviors (2^1 by 2^1), novelty search still can consistently solve the maze that fitness-based search is nearly always unable to solve. . . . 64

Figure 5.1 GP Maze Navigation Maps. In both maps, the circle represents the starting position of the robot and the grey square represents the goal that the robot attempts to reach. Black squares are walls. Cul-de-sacs and occlusions in both maps that prevent direct passage to the goal create the potential for deception. 71

Figure 5.2 Artificial Ant Trails. In both trails, the ant starts in the upper left hand corner. Black squares represent food that the ant attempts to collect, while grey squares illustrate gaps in the trail that the ant must infer. 73

Figure 5.3 Performance Comparison. For each variation of the test domains, the proportion of the 200 runs that are successful are shown for fitness-based GP and GP with novelty search. The main result is that GP with novelty search evolves solutions more consistently than fitness-based GP. 76

Figure 5.4 Bloat Comparison. For each variation of the test domains, the average size (number of nodes) of solutions evolved by GP with novelty search and fitness-based GP is compared (lower is better). In every variation, novelty search evolves significantly smaller program trees. 77

Figure 5.5 Program Growth During Search. The average size (number of nodes) of a program tree in the population is shown as evolution progresses for GP with novelty search and fitness-based GP on the medium (a) and hard (b) mazes, and the Santa Fe (c) and Los Altos (d) ant trails, averaged over the first 25 runs of each approach. The main result is that populations do not continually bloat with novelty search. 78

Figure 5.6 Two randomly-generated maze problems created by the recursive division algorithm. In both mazes, the filled circle represents the starting location and the unfilled circle represents the goal location. The maze shown in (a) has fewer subdivisions and a shorter optimal path to the goal than the maze shown in (b). 82

Figure 5.7 **Reward scheme comparisons.** The effectiveness of novelty search, fitness-based search, and random search at solving problems of increasing complexity is plotted along with linear regressions in (a). Novelty search is the most effective although its performance also degrades with increasing problem complexity. Each plotted point is from ten runs, each lasting 600 generations. The effect on performance of varying the amount of generations for novelty search, fitness-based search, and random search is shown in (b). Novelty search exploits additional evaluations more effectively than fitness-based search or random search. 84

Figure 6.1 Biped Robot. A visualization of the biped robot controlled by an evolved ANN in the experiments in this dissertation. 92

Figure 6.2 Comparing Novelty Search to Fitness-based Search in Biped Walking. The change in distance traveled by the best biped over time (i.e. number of evaluations) is shown for NEAT with novelty search and NEAT with fitness-based search on the biped task. Results are averaged over 50 runs. The main result is that as in the maze domain, novelty search is significantly more effective. 96

Figure 7.1 Comparing the evolvability of novelty search and fitness-based search. The average evolvability of evolved solutions is shown in (a). The average and maximum evolvability of all individuals in the population are shown for novelty search and fitness-based search in (b) the hard maze and (c) the biped walking experiments. Results are averaged over the initial 20 runs. Error bars indicate standard deviation. 107

Figure 7.2 Effects of self-adaptation on evolvability. In all figures, methods with self-adaptation are indicated by the abbreviation *SA*. The average evolvability of all individuals in the final population for each variant are shown in (a), the maximum evolvability of all individuals in the final population are shown in (b), and the average evolvability of discovered solutions are in (c). Results are averaged over 20 runs and error bars indicate standard deviation. The main implication is that self-adaptive parameters help evolvability in novelty search but not in fitness-based search. 112

Figure 8.1 Navigation Maps. In both maps, unfilled circles represent the starting position of the robot and filled circles represent goal points. The maze navigation map (a) is unenclosed, which enlarges the behavior space because the robot can travel outside the maze without restriction. The two-point navigation map (b) has two goal points that a successful navigational robot must reach; the first goal point is the larger filled circle, the second is the smaller filled circle. 122

Figure 8.2 Comparing MCNS, Novelty Search and Fitness-based Search. The change in probability of success over time (i.e. number of evaluations) is shown for NEAT with MCNS, NEAT with novelty search, and fitness-based NEAT on the maze navigation (a) and two-point navigation (b) tasks, both averaged over 100 runs of each approach. The main result is in both tasks MCNS evolves solutions with significantly higher probability after $500,000$ evaluations ($p < 0.01$; Fisher's exact test). . 127

Figure 9.1 ERO Encoding. A nested graph genotype is shown in (a) that consists of outer nodes (i.e. the larger circles) that represent morphological parts, connections among outer nodes (e.g. F and G) that represent joints, inner nodes (i.e. the smaller circles) that represent neurons in an ANN, and neural connections among inner nodes (e.g. the connection between B and E). The phenotype shown in (b) is unrolled from the genotype (a) according to flags associated with the connections between outer nodes that can encode symmetry (e.g. the joint F and its children are reflected over the x-axis) and repetition (e.g. joints can be repeated to create articulated appendages like fingers), yielding a hierarchical repeating structure. 133

Figure 9.2 Absolute Performance Comparison (larger is better). For each setup, the maximum fitness discovered in a particular run is shown (averaged over 15 runs). The main result is that novelty search with global competition discovers the most fit individuals ($p < 0.001$). 147

Figure 9.3 Niche Sparsity (larger is worse). For each setup, the average coverage of morphology space of the final population of a particular run is shown (lower is better; averaged over 15 runs). The main result is that novelty with local competition and novelty search alone cover the niche space the best ($p < 0.001$). 148

Figure 9.4 Niche Capacity. The capacity of evolution to exploit different morphological niches is illustrated above. Each square represents a segment of morphology space (only two out of three morphological dimensions are visualized), and its darkness is proportional to the logarithm of the highest fitness value found within that segment of morphology space (i.e. darker means more fit), over all runs of all variants; thus it is an estimate of the upper bound of fitness that the niche supports. 148

Figure 9.5 Niche Exploitation (larger is better). For each setup, the average niche exploitation is shown (averaged over 15 runs). The main result is that novelty with local competition exploits niches significantly better on average than the other setups ($p < 0.001$). 149

Figure 9.6 Comparing exploration and exploitation of morphological niches. The final distribution of virtual creatures from a single run in two morphological dimensions (height and mass) is shown for typical runs of (a) novelty only, (b) fitness only, (c) novelty search with global competition, and (d) novelty search with local competition. The darkness of each point indicates how fit each virtual creature is. The main result is that local competition effectively exploits more of the morphological niche space than the other methods. 150

Figure 9.7 Diverse competent morphologies discovered within a typical *single* run of local competition. Various creatures are shown that have specialized to effectively exploit particular niches of morphology space. These creatures were all found in the final population of a typical run of local competition. The hopper (a) is a unipedal hopper that is very tall, (b) is a heavy short crab-like creature, and (c) and (d) are distinct quadrupeds. Creature (c) drives a large protrusion on its back to generate momentum, and (d) has a tail for balance. 151

LIST OF TABLES

Table A.1 NEAT Parameter Settings. This table shows the NEAT parameters for the maze experiments in Chapter 4, the biped experiment in Chapter 6, and the mazes in the MCNS experiment in Chapter 8. Archive threshold and k-nearest neighbor parameters apply only to NEAT with novelty search, while time constant and bias parameters apply only to the biped experiment. 172

Table A.2 Biped Simulation Parameter Settings. Parameters are shown for the physical simulation of the biped robot from Chapter 6, implemented using the freely available Open Dynamics Engine library. Maximum torque is the most torque a proportional controller can apply. The proportional constant is multiplied by the disparity between the actual angle of the joint and the angle that the ANN requests to derive the torque to be applied at that joint. 173

Table A.3 Maze Problem. The parameters for the GP maze problem are shown. . 175

Table A.4 Artifical Ant Problem. The parameters for the GP artificial ant problem are listed. 175

Table A.5 Scalable-difficulty Maze parameters. The parameters for the variant of the GP maze problem with scalable problem difficulty are shown. . . . 176

CHAPTER 1
INTRODUCTION

At present, the primary abstraction of natural evolution in evolutionary computation (EC; [14, 70, 39]) is to optimize the fitness function, which abstracts the concept of biological fitness. The implicit assumption is that optimization, i.e. encouraging progress as measured by the fitness function, is the key ingredient to natural evolution's prolific creativity. It is tempting to believe that natural evolution works in this way, and that in general encouraging progress towards an objective is the best way to light a path to it through the search space. Yet work in this dissertation demonstrates that sometimes *not* looking for the objective can more readily reveal its path, and suggests that natural evolution's creativity may result more from its lack of an overall objective than from optimization.

The concept of the *objective function*, which rewards getting closer to the goal, is ubiquitous in machine learning [116]. The overriding intuition behind the idea of the objective function, which is widely accepted, is that the best way to improve performance is to reward improving performance with respect to the objective. In EC, this objective measure of performance is called the *fitness function* [116, 70, 39], which is a metaphor for the pressure to adapt in nature. Pressure to optimize fitness in evolutionary algorithms (EAs) traditionally causes them to converge to a single solution, in stark contrast to natural evolution's drive towards diversity.

So although they are pervasive, objective functions often suffer from the pathology of *local optima*, i.e. dead ends from which no local step in the the search space can improve the performance on the objective function. Convergence to such local optima is problematic in EC as well as machine learning as a whole. To understand intuitively a major cause of local optima, consider fingers stuck within a Chinese finger trap. While the goal is to free one's fingers, performing the most direct action of pulling them apart yields no progress. Rather, the necessary precursor to solving the trap is to push one's fingers *together*, which seems to entrap them more severely. In this way, the trap is *deceptive* because one must seemingly move farther from the goal to ever have the hope of reaching it. Another example is navigating in a city: In a modern city with streets laid out in a regular grid pattern, traveling along roads in the *direction* of the destination will often result in efficiently reaching the destination. However, as many tourists are painfully aware, in cities with irregular roads employing the same strategy will not always work: Heading towards the destination may mean traveling along deceptive roads that initially appear promising but instead twist further from the destination or towards dead ends. In both of these examples, the heuristic measure of distance to the goal can be misleading.

Similarly, landscapes induced by objective (e.g fitness) functions are often deceptive [53, 124, 103, 100]. Although researchers are still working to characterize the various reasons that evolutionary methods may fail to reach the objective [107, 53, 115, 60, 87, 71, 32, 169, 109, 82, 16], as a rule of thumb, the more ambitious the goal, the more difficult it may be to articulate an appropriate objective function and the more likely it is that search can be deceived by local optima [47, 176]. The problem is that the objective function does not necessarily reward the *stepping stones* in the search space that ultimately lead to the objective. As objectives become more ambitious, it becomes increasingly difficult to craft an objective function such that the resulting landscape induces a monotonically improving path through the search space with no dead ends. For example, while it makes sense for a beginner-level program learning to play checkers to be evaluated against another beginner, it would be foolish to evaluate it against the world champion, because it would simply lose every game terribly without exposing a gradient for improvement; any small change in the beginner's gameplay would be obfuscated to the objective function by a similar crushing defeat from the champion. A search guided by such an uninformative metric would likely fail no matter how powerful the underlying search algorithm is.

For these reasons, ambitious objectives in EC are often carefully sculpted through a curriculum of graded tasks, each chosen delicately to build upon the prior [45, 56, 162]. Yet such *incremental evolution* is difficult and ad hoc, requiring intimate domain knowledge and careful oversight, opposing the ideal of automated *machine learning.* Other methods, such as simulated annealing or tabu search, accept deleterious changes or backtrack in an attempt to avoid local optima [116, 51, 89]. However, such methods, along with others employed in EC to deal with deception [107, 53, 76, 73, 52, 124, 160, 8, 92], do not ameliorate the underlying problem of local optima; if local optima are pervasive, search will still likely be deceived. Analogously, a treasure map long weathered by the elements may be unintelligible even to the most experienced treasure hunter. In this scenario, the treasure hunter is not at fault;

the problem is the lack of discernable information in the map. Likewise, deception may be an unavoidable consequence of certain objective functions irrespective of the underlying search algorithm. Paradoxically, in these cases pursuing the objective may prevent the objective from being reached.

Interestingly, in contrast to the focus on objective optimization in machine learning and EC, researchers in artificial life often study systems without explicit objectives, such as in *open-ended evolution* [150, 21, 108]. An ambitious goal of this research is to reproduce the unbounded innovation of natural evolution. A typical approach is to create a complex artificial world in which there is no final objective other than survival and reproduction [174, 21, 2, 148, 161]. Such models assume that biologically-inspired evolution supports creating an open-ended dynamic that leads to unbounded increasing complexity [108, 21, 11].

However, a growing yet controversial view in biology is that the drive towards complexity in natural evolution is a passive force, i.e. not driven primarily by selection [106, 112, 113, 58]. In fact, in this view, the path towards complexity in natural evolution can sometimes be *inhibited* by selection pressure. If selection pressure is too high, then any deviation from a locally optimal behavior will be filtered out by selection. Thus, in this view, instead of being a byproduct of selection, perhaps the accumulation of complexity is better explained by a different characteristic of natural evolution and open-ended evolutionary systems in general: They continually produce novel forms [150]. That is, complexity could simply be the byproduct of continual passive accumulation of phenotypic variation [58].

This perspective leads to a key idea in this dissertation that approaches the problems in both EC and artificial life in a new way: Instead of modeling natural evolution with the hope that novel individuals will be continually discovered, it is possible to create an open-ended dynamic by simply searching *directly* for novelty. Thus this dissertation introduces the *novelty search* algorithm, which searches with no objective; rather, it continually searches for novel behaviors in the search space. That is, the algorithm searches explicitly for behaviors not demonstrated by prior individuals. By defining novelty in this domain-independent way, novelty search can be applied to real world problems, bridging the gap between open-ended evolution and the practical application of EC. Interestingly, because there are only so many ways to behave, some of which must be more complex than others [35, 58], the passive force in nature that leads to increasing complexity is potentially *accelerated* by searching for behavioral novelty.

Paradoxically, as shown in this dissertation, the search for novelty often evolves objectively superior behavior to evolution that is actually driven by the objective. To demonstrate this phenomenon, novelty search is compared to objective-based evolution in a deceptive two-dimensional robot maze navigation task and a variety of other experiments that establish novelty search's generality. Counterintuitively, novelty search, which ignores the objective, evolves successful maze navigators that reach the objective in significantly fewer evaluations than the objective-based method. Furthermore, the solutions novelty search finds are more elegantly represented. For harder mazes, the objective-based method almost always fails, while novelty search is successful in nearly every attempt. This result holds in a different representation (genetic programming) and in hundreds of additional random mazes of varying difficulty. Additionally, outside researchers have tested a variety of other domains in which novelty search is beneficial [136, 91, 55, 117, 120, 102, 97, 43]. , including virtual creatures [97], artificial ants finding food [43], and discovering bugs in computer programs [55].

To provide evidence of its ability to avoid deception even in real-world problems, in this dissertation I also apply novelty search to a challenging three-dimensional biped locomotion domain. In the biped domain, novelty search evolves both more elegant representations and controllers that walk significantly farther than those evolved by the the objective-based method. The culmination of all these results challenges the premise that the objective is always the proper impetus for search. If the objective function does not align with the stepping stones needed to reach the objective, then directly searching for the objective may be futile, no matter the search algorithm. However, in such cases, by discarding the false security of the objective function, as novelty search does, it *may* still be possible to reach the objective.

Furthermore, because novelty search indirectly rewards lineages better able to produce novelty, it may also encourage evolvability, or the ability to further evolve, which is a desirable property that is apparent in natural evolution but rarely in EC. In contrast to novelty search, the pressure to optimize an objective can *discourage* evolvability when it converges upon a local optimum. Supporting this idea, experiments in the maze navigation and biped domains in this dissertation provide evidence that often novelty search discovers more evolvable solutions than the the traditional search for the objective. This drive towards evolvability, in combination with its tendency to evolve more elegant representations, suggests that novelty search may also often provide secondary benefits in addition to its ability to circumvent deception.

Finally, because it offers a different perspective on natural evolution from traditional EC approaches, novelty search can also lead to different abstractions of natural evolution. While the traditional search for the objective in EC is based on an abstraction focusing on natural selection's ability to optimize, novelty search instead abstracts evolution as a generator of novel forms. Extending this abstraction in two ways leads to two additional algorithms intended to yield dynamics closer in spirit to natural evolution. The first, minimal criteria novelty search (MCNS), abstracts evolution as a search for many ways of doing the same thing (i.e. to survive and reproduce). This algorithm is shown to solve classes of mazes that otherwise stymie both the raw search for novelty and objective-based search. The second, novelty search with local competition (NSLC), abstracts evolution as an accumulator of novel and locally-competitive organisms. The key insight is that niches in nature have significantly different capacities for supporting fitness, and competition between niches in nature not *global*. However, global competition is ubiquitous in EC. As a culminating application of this dissertation, NSLC is applied to evolve a significant diversity of locomoting virtual creatures within a *single* evolutionary run, bringing EC a step closer towards creating a "complexity explosion" reminiscent of natural evolution.

The conclusion is that by abstracting the process and constraints through which natural evolution discovers novelty, it is possible to derive a variety of open-ended search algorithms that operate without pressure towards the ultimate objective. Novelty search is immune to the problems of deception and local optima inherent in objective optimization because it entirely ignores the objective, suggesting the surprising conclusion that ignoring the objective in this way may often *benefit* the search for the objective. Yet apart from being a practical tool for solving problems or a means of illustrating deep-seated limitations of objective-based search, as a significant departure from previous methods in EC novelty search provides a fresh perspective on evolutionary algorithms and may bring EC closer to matching nature's vast creativity.

1.1 Contributions

The research hypothesis of this dissertation is that searching without an ultimate objective, as novelty search does, is often a viable alternative to the objective-based abstraction of evolution prevalent in EC. This hypothesis is supported by the following contributions:

1. This dissertation is the first to introduce novelty search, a search algorithm without an ultimate objective, which I co-invented with Dr. Kenneth O. Stanley.

2. Novelty search is validated by applying it to biped locomotion, a challenging machine learning domain, where it outperforms the control objective-based algorithm.

3. The potential of novelty search to increase evolvability is demonstrated through experiments in maze navigation and biped locomotion.

4. An extension of novelty search called minimal criteria novelty search (MCNS) is introduced that brings a new abstraction of natural evolution to EC: evolution as a search for novel ways of doing the same thing. This extension also suggests a way to constrain the raw search for novelty that may sometimes be necessary in practice.

5. Another extension of novelty search is introduced called novelty search with local competition (NSLC), which abstracts natural evolution in a different way than novelty search or MCNS. NSLC combines the drive to accumulate novelty with pressure to achieve in a way that respects diversity.

6. As a culminating application, novelty search with local competition is used to evolve a wide diversity of well-adapted locomoting virtual creatures in a single run.

1.2 Outline

The dissertation begins with background on evolutionary computation and the relationship between natural evolution and artificial life. Chapter 3 then introduces novelty search. In Chapter 4 novelty search is applied to a maze navigation domain. Chapter 5 presents additional results in a variety of domains and EC paradigms that establish the generality of novelty search. As a capstone to these experiments, Chapter 6 then applies novelty search to a challenging biped walking domain and chapter 7 demonstrates novelty search's tendency towards increasing evolvability. Algorithmic variants of novelty search are presented in chapters 8 and 9 that are motivated by different abstractions of natural evolution than the traditional fitness-centric abstraction prevalent in EC. In particular, chapter 9 applies such a variant to the challenging prolem of evolving virtual creatures, where it evolves a wide diversity of functional creatures in a single evolutionary run. Finally, Chapter 10 discusses the significant implications of novelty search to the fields of EC and artificial life, and chapter 11 concludes the dissertation.

CHAPTER 2
BACKGROUND

To provide context for understanding novelty search, this chapter first reviews evolutionary computation and the significant problem of deception within it. Next, complexity in natural evolution is reviewed, as is the field of open-ended evolution. The chapter concludes by reviewing the NEAT neuroevolution method used in most of the experiments in this dissertation.

2.1 Evolutionary Computation

Evolutionary computation (EC; [14, 70, 39]) attempts to algorithmically abstract the process of natural evolution, inspired by its drive towards innovative and complex solutions to the problems of life. The general idea is to capture the essence of such creativity and unleash it in domains other than natural life. In particular, EC researchers often apply evolutionary algorithms (EAs) to practical problems such as numerical optimization [168] or automatic design [75]. While there are many potential algorithmic abstractions of natural evolution (this dissertation introduces two new such abstractions in chapters 8 and 9), EC is generally driven by a particular dominant abstraction that this dissertation argues may entail significant limitations.

This dominant abstraction of natural evolution in EC is based on the concept of biological fitness and its relation to natural selection. That is, natural selection implies that the most biologically fit, i.e. those organisms that on average produce the most offspring, will tend to displace those that are less fit. The motivating idea is that evolution *optimizes* fitness and that this optimization is what leads to evolution's impressive solutions to the problems of life. However, it is important to note that attributing the impressiveness of evolution's products to the optimizing force of natural selection is not universally accepted within biology [106, 58]. Still, in EC biological fitness is a central focus and is abstracted in EAs as the *fitness function*.

While in nature fitness is defined as the reproductive success of an organism, in an algorithmic abstraction of evolution, fitness can be redefined task-specifically. Such redefinition can be seen as similar in spirit to how an animal breeder can artificially choose among animals to breed by arbitrary criteria, e.g. a horse breeder could breed horses for their speed. That is, in practice an experimenter crafts a fitness function for a particular task that will yield what they are looking for if successfully optimized by the EA. For example, if the task is to design a controller for a biped robot that can walk a long distance, the fitness function might measure how far the biped traveled before falling over. The general idea is that higher values of the fitness function imply better performance. Thus when derived from this abstraction, EAs act as optimizers, searching for maximal fitness.

Importantly, the concept of a fitness function is a key unifying feature between superficially distinct flavors of EAs such as genetic algorithms [70], evolutionary programming [48], and evolutionary strategies [129]. Each of these variants is driven by pressure to optimize a fitness function. Additionally, such fitness functions are examples of a wider class of *objective functions* common in the larger field of machine learning [89, 116, 51]. Thus the term *objective-based search* in this dissertation refers to a search method explicitly searching for a particular objective, as an EA does when maximizing a fitness function.

In practice, the fitness function acts as a mechanism for automatically breeding a solution to a problem. Given a means of *representing* a space of possible solutions to a problem (i.e. an artificial genetic encoding), a fitness function can provide objective scores for a given set of genes and thereby guide the search towards promising areas of the genetic search space.

The general formulation of an EA is that first a *population* of random solutions (also called individuals or candidates) is generated. Then for each successive generation, individuals scoring among the highest by the metric of the fitness function are chosen and bred together to yield new variants that constitute a new population. The hope is that over time variants of high-fitness individuals will lead to ever-further improvements until finally an individual that implements a desired solution to the task is produced.

If such an approach always worked, then EAs would be a panacea for creating anything for which a fitness function could be imagined. However, as practitioners in EC are aware, trying to optimize an arbitrary fitness function does not always succeed. The problem is that fitness functions do not always induce a smooth path through the search space for evolution to traverse. In particular, ambitious fitness functions often can cause evolution, with its lack of foresight, to converge upon a deceptive cul-de-sac in the search space from which no immediate improvement is possible.

Thus the next section further reviews the significant problem of deception that this dissertation argues is generally intractable yet problematically inherent in the abstraction of evolution as an optimizer. Because this dissertation introduces a novel means of circumventing such deception by entirely abandoning objectives, previous attempts to mitigate deception are also reviewed.

2.2 Deception In Evolutionary Computation

The term *deception* is coined by Goldberg [53] in the context of EC, building upon work by Bethke [14] to describe fitness landscapes in which seemingly promising directions ultimately lead away from global optima. The study of deceptive landscapes is part of a larger study by EC researchers into what may cause an evolutionary algorithm to fail and how to remedy such failures. For the purpose of this dissertation, it is instructive to study the role of the objective (fitness) function in such failures and remedies. Thus this section reviews the investigation of deception and problem difficulty as well as methods designed to mitigate deceptiveness in practice.

2.2.1 Deception and Problem Difficulty

The motivation behind characterizing deception and problem difficulty is to understand what properties of problems may cause EAs to fail to reach the objective. If these properties are understood, they can potentially be remedied or avoided.

The original definition of deception by Goldberg [53] is based on the building blocks hypothesis, in which small genetic building blocks (called schemata) may be integrated by crossover to form larger blocks [70]. In the original conception, a problem is deceptive if lower-order building blocks, when combined, do not lead to a global optimum. A variety of work has further refined this definition and investigated performance on deceptive problems [53, 124, 103, 100]. Whitley [171] reached the strong conclusion that the *only* challenging problems for genetic algorithms (GAs) are those with some degree of deception.

However, Grefenstette [60] suggested that deception was neither necessary nor sufficient for a problem to be difficult for a GA. Similarly, Mitchell et al. [115] argued that deception was only one of many characteristics of fitness landscapes that can make optimization difficult for GAs. Interestingly, no measure of problem difficulty can be perfect because in general it is impossible to know the outcome of an algorithm on a particular set of data without actually running it [62]. Hence, many different metrics of EA problem hardness have been explored [87, 71, 32, 169, 109, 82, 16].

Some alternative measures of problem difficulty attempt to model or quantify the ruggedness of the fitness landscape (i.e. the landscape resulting from mapping every genotype to its corresponding fitness [173]), motivated by the intuition that optimizing more rugged landscapes is more difficult [87, 71, 32, 169, 109]. Such approaches are often based on the concepts of *correlation* or *epistasis*. Correlation metrics typically measure the degree to which the fitness of individuals are well correlated to their neighbors in the search space; the idea is that if correlation is high, the fitness landscape is likely smooth and easy to optimize, while low correlation implies a rough landscape that may be difficult to optimize [169, 87, 109, 71]. Epistasis is a term from biology describing the situation in which the effects of one gene are influenced by the effects of other genes. That is, if there is no epistasis, genes may be predictably adjusted independently, while a large amount of epistasis implies unpredictable non-linear interactions among genes. As linkage between genes' effects increases, search becomes more difficult because the resulting landscape is more rugged [32, 87]. Importantly, because the fitness landscape is induced by the objective function, the problem of ruggedness, presupposing reasonable settings for the EA, can be attributed to the objective function itself.

Interestingly, other researchers suggest that ruggedness is overemphasized and that neutral fitness plateaus (neutral networks) are key influences on evolutionary dynamics [7, 163, 66]. The importance of neutrality is supported by the theory that most genetic changes in natural evolution are neutral with respect to fitness [88]. However, even neutral networks suggest a deficiency in the objective function: By definition a neutral part of the search space contains no gradient information with respect to the objective function. That is, in a neutral network the map defined by the objective function is ambiguous with respect to which way search should proceed.

A substantially different approach is taken by Borenstein and Poli [16], in which the fitness landscape is completely ignored and instead problem difficulty is measured based only on the distribution of fitness values. The measurement entails running a GA many times on a particular problem over a sampling of all possible representations. The idea is to isolate problem difficulty more generally without fixing a representation, although it leads to the counterintuitive conclusion that the needle in a haystack problem is *easier* than the one-max problem. This method illustrates the importance of choosing sensible representations, without which common intuitions about search may no longer hold *regardless* of the objective or search algorithm.

Another approach, by Jones and Forrest [82], suggests that the degree to which the heuristic of fitness relates to the real distance to the goal is a good measure of difficulty, although it (unsurprisingly) mispredicts in certain situations [4]. This perspective perhaps most clearly demonstrates that a large factor in what makes a given problem difficult may be an uninformative objective function: When the heuristic has little basis in reality (i.e. when the objective function does not agree with the real distance to the objective) there is little reason to expect search to perform well. What is problematic, as will be echoed by the experiments in this dissertation, is that sometimes pursuing what appears to be a reasonable *objective* produces an unreasonable *objective function*.

Finally, Guo and Hsu [62] clarify EA problem difficulty by examining it within the framework of the No Free Lunch (NFL) theorem for black box optimization. The authors state that what makes a particular problem difficult for a particular EA may stem from three possibilities. The first is that the fitness landscape lacks information that *any* search algorithm could exploit (e.g. a fitness function that yields a landscape similar to that of the needle-in-the-haystack problem); in this case the objective-based paradigm is completely impotent, but there may yet be hope for search methods that exploit *additional* information such as the behavior of individuals. The second possible cause of problem difficulty is when the search algorithm could, but does not, exploit information that would allow the algorithm to perform better than random search on a particular problem. In this case, it is tempting to blame the search algorithm and revise it so that it *does* integrate the helpful information. However, by specializing the algorithm, it is then vulnerable to the third source of problem difficulty: deceptive problems that are structured based on *conflicting* information from that to which the search algorithm was specialized. Such problems are relatively common even when the search algorithm makes reasonable assumptions, as will be shown in the experiments in this dissertation. Thus, deception cannot be generally avoided within the objective-based paradigm, suggesting the need for alternative paradigms that exploit *additional information*.

In summary, there are many ways to consider, measure, and model the difficulty of problems for EAs. While in general the exact properties of a problem that make it difficult for EAs are still a subject of research, in this dissertation the term deception will refer to an intuitive definition of problem hardness: A deceptive problem is one in which a reasonable EA (with a reasonable representation, parameters, and search operators) will not reach the desired objective in a reasonable amount of time. That is, a deceptive problem is simply a problem in which following the gradient of the objective function leads to local optima. It is important to note that this definition of deception is different from the traditional definition [53] and is not meant to trivialize the impact or difficulty of choosing the correct representation [138], parameters [38], or search operators [175], all of which affect the performance of the EA and the structure of the fitness landscape. Rather, the intuitive approach helps to isolate the general problem with particular objective functions because the word "deception" itself reflects a fault in the *objective function* (as opposed to in the algorithm itself): An objective function with the pathology of deceptiveness will *deceive* search by actively pointing the wrong way.

Next, previously developed methods to mitigate deception are reviewed.

2.2.2 Mitigating Deception

Ideally, there would exist a silver bullet method immune to the problem of deception such that any objective would be reachable in a reasonable amount of time. This hypothetical ideal method is attractive because it would not require any conceptual change to the comfortable paradigm of objective-based search. Thus, although it is impossible that any such silver bullet method exists [172], researchers strive to create methods that can overcome deception in practice.

A common way to prevent premature convergence to local optima in EC is by employing a diversity maintenance technique [107, 54, 76, 73, 78]. Many of these methods are inspired by speciation or niching in natural evolution, wherein competition may be mostly restricted to occur *within* the same species or niche, instead of encompassing the entire population. The benefit of such restricted competition is that if one species is centered on a deceptive local optimum, there is still hope that another species may reach the global optimum. In effect, the search is less greedy and more explorative. For example, fitness sharing [54] enforces sharing of fitness (i.e. competition) between similar solutions so that there is pressure to find solutions in distant parts of the search space even if the fitness values encountered there initially seem less promising. While fitness sharing forces similar genotypes to compete, Hierarchical Fair Competition (HFC [76]) enforces competition among individuals with similar fitness scores by creating an explicit hierarchy of populations. Similarly, the Age-Layered Population

Structure (ALPS [73]) approach enforces competition between genomes of different genetic ages.

Interestingly, *fitness uniform selection* [78] removes the direct pressure to increase fitness entirely: Selection pressure favors a uniform distribution of fitness values in the population instead of pressure towards only higher fitness values. That is, an individual is not rewarded for higher fitness, but for a *unique* fitness value compared to the rest of the population. This approach is fundamentally different from most other techniques that deal with deception and can be viewed as a search for novel fitness scores, which is related to the approach in this dissertation; however, as will be demonstrated later (Chapter 4.3.2.3), the conflation of vastly different behaviors that receive the same fitness score can sometimes be deleterious. Relatedly, Juric [83] argues for reducing selection pressure via an anti-adaptionist evolutionary algorithm that emphasizes minimal satisfactory fitness (as in natural evolution) over optimality. Although these methods do encourage exploration (typically genotypically), if local optima are pervasive or genotypic difference is not well-correlated with phenotypic/behavioral difference, these methods may still be deceived.

Other methods for avoiding deception tackle the problem of ruggedness (i.e. epistasis) by rearranging the genome so that crossover respects genes whose effects are linked [52], or by building models of interactions between genes [124]. When successful, the effect is to smooth a rugged fitness landscape. This approach can be be viewed as an attempt to derive additional information from an imperfect objective function. However, given a sufficiently uninformative objective function, the advantage of such modeling is impaired.

Still other methods, taking into account neutral networks, seek to accelerate search through such networks [160, 8]. Stewart [160] explicitly rewards drifting further away in genotype space from the center of the population. The idea is that once a neutral area of the search space is recognized, instead of aimlessly meandering through the area, search will progress quickly to find an area with a gradient of improvement to follow. In practice it may be difficult to identify when search is stalled in a neutral network, and encouraging only genetic novelty in variable-length encodings may lead to unnecessarily large genomes (i.e. by needlessly increasing the dimensionality of the search space). Barnett [8] similarly seeks to accelerate movement through a neutral area of search space, but takes an orthogonal approach: When neutrality is detected, the population shrinks to one individual that crawls over the neutral network. While these methods may decrease the amount of meandering that occurs in neutral networks, if an unlikely series of specific mutations is needed to exit the neutral network then search may still be stalled for a long time.

In direct response to the problem of local optima when evolving towards sophisticated behaviors, some researchers incrementally evolve solutions by sequentially applying carefully crafted objective functions [45, 56, 162]. This approach is similar to how children learn simple tasks before more complex tasks [45]. For example, Elman [45] reported that training a neural network to recognize a complex grammar failed when employing a static approach, but was successful in two incremental variants: The first variant trained with only simple patterns at first and then incrementally added more difficult examples, while the second variant at first limited the memory of the network so that it would only learn from simple examples and then increased the memory to its full capability over time. Similar performance gains by training incrementally were noted by Gomez and Miikkulainen [56] and van de Panne and Lamouret [162]. These efforts demonstrate that sometimes it is not enough to carefully craft a *single* objective function: To avoid deception it may be necessary to first *deeply understand* the domain in question so that a series of objective functions can be crafted that respect the stepping stones that ultimately lead to the objective. However, with ambitious objectives these stepping stones may be difficult or impossible to determine a priori. Additionally, the requirement of such intimate domain knowledge conflicts with the aspiration of *machine* learning.

In addition to single-objective optimization, there also exist evolutionary methods that aim to optimize several objectives at once: Multi-Objective Evolutionary Algorithms (MOEAs) [164]. There are different varieties of MOEAs, but pareto-dominance-based methods such as NGSA-II are most popular and often effective [164, 42, 72]. Pareto-based methods are an elegant alternative to the quagmire of integrating several objectives together into a single monolithic objective. The key insight is a reformulation of the concept of superiority: In a problem with more than one objective, two individuals that each perform better than the other in at least one of potentially many objectives may be seen as incomparable, or mutually *non-dominated*. When one individual is better than the other individual in at least one objective, and at least equal in the rest, then the first individual *dominates* the second. These MOEAs are not immune to the problem of deception [41], and adding objectives does not always make a problem easier [17], but the idea is that perhaps deception is less likely when optimizing multiple objectives because progress can be made with respect to an alternate objective if a local optimum has been reached in a different objective [92]. In effect, the search may be more open because a wider class of behaviors are seen as equal in performance. In this spirit, some researchers have experimented with *multi-objectivization*, i.e. extending single-objective problems into multi-objective problems to avoid deception [92]. There are generally two approaches, which are either to decompose a problem into multiple sub-problems that are each made objectives [63, 92], or to add new objectives that may indirectly encourage progress [61, 37]. Decomposing a problem can either make it easier or harder [64], and it is necessary to verify that the single-objective optima are multi-objective pareto-optima in the transformed multi-objective problem [92]. There have been several successful applications of multiobjectivization [92, 63, 61, 37], but as in other reviewed methods, the fundamental pathology of deception remains.

Yet another avenue of research in EC related to deception is coevolution. Coevolutionary methods in EC attempt to overcome the limitations of a single static fitness function by making interactions between individuals contribute towards fitness. In this way, the fitness landscape is constantly changing with the population. The benefit of this approach is that while an objective function based on a single fixed grandmaster opponent may not provide a gradient of improvement for a beginner to learn against, many co-evolved opponents of varying skill levels may provide such a desirable gradient. The hope is that such competition between individuals will spark an evolutionary *arms race* in which the interactions between individuals continually creates a smooth gradient for better performance [25]. There have been several impressive successes [69, 147, 23], perhaps most notably the evolution of an expert level checkers player [23]. However, a common problem is that in practice such arm races may converge to mediocre stable-states [47, 126, 127, 167], cycle between various behaviors without further progress [25, 20, 167], or one species may so far out-adapt another that they are evolutionarily disengaged [19, 20, 167]. These coevolutionary difficulties are somewhat analogous to the problem of local optima in traditional objective-based search; the difficulty for practitioners in coevolution, much like the difficulty of crafting an effective fitness function facing researchers in standard EC, is to construct an environment that provides sustained *learnability* in which the gradient of improvement is always present [47].

Finally, outside of EC there is also general interest in machine learning in avoiding local optima. For example, simulated annealing is inspired by the annealing process that occurs as metals cool, and probabilistically accepts deleterious changes with respect to the objective function based on a cooling schedule [89]. Initially, the cooling schedule accepts deleterious changes often, but as search progresses the rate of acceptance drops. Another machine learning technique called tabu search avoids re-searching certain areas of the search space [51]. However, if local optima are pervasive, then these methods too can fail to reach global optima.

In summary, because deception is a significant problem in EC, there are *many* methods that have been designed to mitigate deception in practice. However, while they may work for some problems, ultimately such methods do not ameliorate the underlying pathology of the objective function that causes deception: The gradient of the objective function may be misleading or uninformative. Instead, current methods that deal with deception attempt to glean as much information as possible from an imperfect objective function or encourage exploration in the search space. Given a sufficiently uninformative objective function, it is an open question whether *any* method relying solely on the objective function will be effective. Thus an interesting yet sobering conclusion is that some objectives may be unreachable by direct objective-based search alone. Furthermore, as task complexity increases it is more difficult to successfully craft an appropriate objective function [47, 176]. For example, Zaera et al. [176], in a rare publication of a negative result, acknowledge explicitly the difficulty of crafting objective functions:

> The problem appears to be due to the difficulty of formulating an evaluation function which captures what schooling is. We argue that formulating an effective fitness function for use in evolving controllers can be at least as difficult as hand-crafting an effective controller design. Although our paper concentrates on schooling, we believe this is likely to be a general issue, and is a serious problem which can be expected to be experienced over a variety of problem domains.

Similarly, Ficici and Pollack [47] expound on why objective functions fail for more ambitious problems:

> Most machine learning (ML) systems operate by optimizing to a fixed fitness function ... and typically require considerable inductive bias in order to succeed; this inductive bias takes the form of either a learner that is *pre-adapted* to the learning environment, or a carefully *gradient-engineered* fitness landscape that provides the learner with a clear path towards a global optimum. ... As learning domains become more intricate ... both methods of bias engineering quickly become infeasible: gradient engineering turns overwhelmingly complex, and, following the observation that "you can only learn what you almost already know," pre-adaptation requires the learning problem to be already substantially solved.

These insights match many EC practitioners' experience that the difficulty in ambitious experiments is often in crafting a sufficient fitness function. Thus, the ultimate conclusion is that the more ambitious the experiment, the less likely it is that an objective function will correctly recognize the stepping stones that lead to the objective and the more likely it is that search will lead to local optima as opposed to the desired goal behavior.

The next section discusses complexity in natural evolution, which, in contrast to traditional EAs, is a search process *without* any final objective.

2.3 Complexity in Natural Evolution

Natural evolution fascinates practitioners of search because of its profuse creativity, lack of volitional guidance, and perhaps above all its apparent drive towards complexity. Incredibly, this unguided process produced the human brain, an artifact that has resisted all attempts to recreate the entirety of its complex functionality in software.

A subject of longstanding debate is the arrow of complexity [11, 113], i.e. the idea that evolutionary lineages sometimes tend towards increasing complexity. For example, it is hard to dispute that humans are more complex than their bacterial ancestors. What about evolutionary search in nature causes complexity to increase? This question is important because the most difficult problems in search, e.g. an intelligent autonomous robot, may require discovering a prohibitive level of solution complexity.

The topic of complexity in natural evolution is much in contention across biology, artificial life, and evolutionary computation [106, 118, 113, 156, 58, 5, 112]. One important question is whether there is a selective pressure towards complexity in evolution (i.e. does selection drive natural evolution to create "higher" forms). While even Darwin himself may have believed that evolution was a progressive process ("as natural selection works solely by and for the good of each being, all corporeal and mental endowments will tend to progress towards perfection" [31, p. 489]), such beliefs, especially when human-centric (i.e. human beings are the pinnacle or inevitable result of evolution), have since been disputed [135, 33, 79, 58, 140, 112]. A potentially heretical view that is gaining attention is that progress towards higher forms is not mainly a direct consequence of selection pressure, but rather an inevitable passive byproduct of random perturbations [106, 113, 58]. Researchers like Miconi [113] in artificial life, Sigmund [145] in evolutionary game theory, and Gould [58], McShea [112], and Lynch [106, 105] in biology are arguing that natural selection does not always explain increases in evolutionary complexity. In fact, some argue that to the extent that fitness (i.e. in nature, the ability to survive and reproduce) determines the direction of evolution, it can be deleterious to increasing complexity [106, 113, 145]. In other words, rather than laying a path towards the next major innovation, fitness (like the objective function in machine learning) in effect prunes that very path away.

In particular, a quote from Miconi [113] illustrates this point of view:

> ... the Darwinian heuristic actually imposes a *restriction* on the search process: the search process will now follow certain directions at the expense of others. This restriction constrains both the scope and the direction of the search process. The restriction in scope comes from the fact that, because evolution proceeds by small jumps from existing designs, newer designs can only be chosen in the neighborhood of the current population. The restriction in direction comes from the fact that only those designs which exhibit a high fitness will be used as starting points for further exploration.

Sigmund [145, p. 85], describes how high levels of selection pressure in nature can *oppose* innovation:

> Indeed, an improvement of the genome can sometimes be reached only through intermediate steps which are by themselves disadvantageous. This is like the Rubix cube, whose parts have to be twisted until every face shows one colour only. In order to get a better combination, one must destroy in part what has been achieved so far.

Gould [58], a biologist, goes even further, and argues that an increasing upper bound in complexity is not a product of selection, but simply a result of a drift in complexity space limited by a hard lower bound (the minimal complexity needed for a single cell to reproduce). Lynch [106], another biologist, argues that selection pressure in general does not explain innovation, and that non-adaptive processes (i.e. processes not driven by selection) are often undeservedly ignored: "What is in question is whether natural selection is a necessary or sufficient force to explain the emergence of the genome and cellular features central to the building of complex organisms." Similarly, the role of genetic drift (a non-adaptive process) and neutral mutations are stressed by Kimura's influential neutral theory of molecular evolution [88]. Interestingly, Huyen [80] suggests that neutral mutations in nature allow for a nearly limitless indirect exploration of phenotype space, a process that this dissertation seeks to directly accelerate.

These arguments coupled with the fundamental pathology of deception lead to the main idea in this dissertation that abandoning the idea of directly optimizing performance (i.e. searching for fitter organisms) and instead simply searching explicitly for novel behaviors may itself exemplify a powerful search algorithm.

2.4 Open-ended Evolutionary Computation

The *open-ended evolution* community in artificial life aims to produce simulated worlds that allow a similar degree of unconstrained exploration as Earth. While a precise definition of open-ended evolution is not uniformly accepted [108, 150], the general intuition is that it should continually create individuals "of a greater complexity and diversity than the initial individuals of the system" [108]. Tierra [128], PolyWorld [174] and Geb [21] are typical attempts to attain such a dynamic. In such systems, there is no objective beyond that of survival and reproduction (as in nature). The motivation behind this approach is that as evolution explores an unbounded range of life forms, complexity will inevitably increase [21, 113].

Tierra [128] is an open-ended evolutionary model in which self-replicating digital programs compete for resources on a virtual computer. Individuals in the system consist of a sequence of basic instructions, somewhat analogous to the set of amino acids that make up DNA in biology. Mutations and probabilistic flaws in the execution of instructions generate variation upon which selection can act. Evolution is seeded with a simple self-replicating program, and proceeds until the simulation is ended. Interestingly, various types of parasites and corresponding immunity to parasites arose in several runs of the system, demonstrating a basic coevolutionary arms race. However, Tierra and other similar systems such as Avida [2], the Computer Zoo [148], and Cosmos [161] all inevitably struggle to continually produce novelty [150, 22, 84].

Polyworld [174] is also a simulation of an ecological system of competing agents, but gives these basic agents embodiment in a two-dimensional world, a brain (i.e. a neural network), and basic actions it may execute (e.g. eating, mating, fighting, moving). The only goal is survival. Interestingly, behaviors seen in real life organisms, such as flocking, grazing, and foraging, although not directly specified by the system, emerge from the interactions and evolution of agents. However, as in the Tierra-like systems, eventually innovation appears to slow, leaving open the challenge of one day achieving a true open-ended evolutionary system [22]. Geb [21], which addresses criticisms of Polyworld and also exhibits emergent behaviors, follows a similar philosophy.

Bedau and Packard [9] and Bedau et al. [10] have contributed to formalizing the notion of unbounded open-ended dynamics by deriving a test (called *activity statistics*) that classifies evolutionary systems into categories of open-endedness. Geb and others are distinguished by passing this test [21, 108], but the results nevertheless do not appear to achieve the levels of diversity or (especially) complexity seen in natural evolution. This apparent deficiency raises the question of what element is missing from current models [150, 108]. Many suggest that more detailed, lifelike domains must be constructed to facilitate the open-ended dynamic of natural evolution [174, 114, 84, 108]. For example, Kampis and Gulyás [84] argue that it is not even possible to achieve a true open-ended dynamic without phenotypes that have "in the broad sense of the word, a full material body with 'depth', which is linked to the germs by nontrivial development as in real organisms."

However, this dissertation presents a more general approach to open-ended evolution that is motivated well by the following insight from Standish [150]: "The issue of *open-ended evolution* can be summed up by asking under what conditions will an evolutionary system continue to produce novel forms." Thus, instead of modeling natural selection, the idea in this dissertation is that it may be more efficient to search directly for novel behaviors. While not intended to replace previous approaches to open-ended evolution, the advantage of this approach is that it decouples the concept of open-endedness from the problem domain because novelty can be sought in *any* domain. Therefore, it can apply to real-world tasks as easily as artificial life worlds. It is important to acknowledge that this view of open-endedness contrasts with the more commonly accepted notion of prolonged production of *adaptive* traits [9, 10]. Nevertheless, the simpler view of open-endedness merits consideration on the chance that a dynamic that *appears* adaptive might be possible to capture in spirit with a simpler process.

Many of the experiments in this dissertation combine this approach to open-ended evolution with the NEAT method, which is explained next.

2.5 NeuroEvolution of Augmenting Topologies (NEAT)

Many experiments in this dissertation evolve behaviors that are controlled by artificial neural networks (ANNs). Thus a neuroevolution (NE) method is needed to underpin these experiments. The NEAT method serves this purpose in this dissertation in part because it is widely applied [155, 157, 151, 1, 131, 158, 152, 153, 170, 3] and well understood. However, it is important to emphasize that the aim is not to validate the capabilities of NEAT yet again. In fact, in some of the experiments NEAT performs poorly. Rather, the interesting insight that NEAT helps to elucidate is that the same algorithm can appear ineffective in an objective-based context yet excel when the search is open-ended. Thus, as a common approach to NE, NEAT is a natural conduit to reaching this conclusion.

The NEAT method was originally developed to evolve ANNs to solve difficult control and sequential decision tasks [155, 157, 151]. Evolved ANNs control agents that select actions based on their sensory inputs. Like the SAGA method [65] introduced before it, NEAT begins evolution with a population of small, simple networks and *complexifies* the network topology into diverse species over generations, leading to increasingly sophisticated behavior.

A similar process of gradually adding new genes has been confirmed in natural evolution [110, 166], and fits well with the idea of open-ended evolution.

However, a key feature that distinguishes NEAT from prior work in complexification is its unique approach to maintaining a healthy diversity of complexifying structures simultaneously, as this section reviews. Complete descriptions of the NEAT method, including experiments confirming the contributions of its components, are available in Stanley et al. [151], Stanley and Miikkulainen [155], and Stanley and Miikkulainen [157]. Next, the key ideas on which the basic NEAT method is based are reviewed.

To keep track of which gene is which while new genes are added, a historical marking is uniquely assigned to each new structural component. During crossover, genes with the same historical markings are aligned, producing meaningful offspring efficiently. Speciation in NEAT protects new structural innovations by reducing competition among differing structures and network complexities, thereby giving newer, more complex structures room to adjust. Networks are assigned to species based on the extent to which they share historical markings. Complexification, which resembles how genes are added over the course of natural evolution [110], is thus supported by both historical markings and speciation, allowing NEAT to establish high-level features early in evolution and then later elaborate on them. In effect, then, NEAT searches for a compact, appropriate network topology by incrementally complexifying existing structure.

It is important to note that a complexifying neuroevolutionary algorithm like NEAT induces an *order* over the complexity of behaviors discovered during search from simple to complex. An important difference between an ANN with five connections and one with five million connections is that the larger network, by virtue of having more free parameters (i.e. connection weights), can exhibit more complex behaviors. For example, a network with five connections cannot encode a master chess player because the available amount of information is too little to encode a behavior of such complexity.

Therefore, as new nodes and connections are added over the course of evolution, the potential complexity of the behaviors that the network can represent increases. Thus, in effect, the increasing complexity of the representation imposes an order on the search from simple behaviors to potentially more complex. Searching behaviors from simple to complex is important because it means that simpler behaviors are encountered first. This consideration is significant because, as a rule of thumb, the *most* complex behaviors are often undesirable because the extreme end of behavioral complexity is associated with irregularity and chaos. That is, the most complex behaviors are completely ad-hoc and specify no reuse of information; we are often impressed by the regularities of biological organisms and their complex *functionality* rather than the complexity of their construction for its own sake. Thus a search heuristic that encounters the most complex ANNs last makes sense.

In many experiments that follow, NEAT is combined with the approach introduced in this dissertation, novelty search, which is explained next.

CHAPTER 3
APPROACH: NOVELTY SEARCH

A new approach to circumventing the problem of deception in EC is introduced in this chapter, which is to abandon the search for the objective completely (i.e. to give up the fitness-based abstraction of evolution) and to search instead only for behavioral novelty. The next sections respectively explain in detail the search for novelty and formalize the novelty search algorithm.

3.1 The Search for Novelty

Recall that the problem identified with the objective fitness function in EC is that it does not necessarily reward the intermediate stepping stones that lead to the objective. The more ambitious the objective, the harder it is to identify *a priori* these stepping stones.

The approach introduced here is to identify novelty as a *proxy* for stepping stones. That is, instead of searching for a final objective, the learning method is rewarded for finding any instance whose functionality is significantly different from what has been discovered before (note that this idea is related to the concept of *curiosity* and seeking novelty in reinforcement learning research [142, 143]). Thus, instead of a traditional objective function, evolution employs a *novelty metric*. That way, no attempt is made to measure overall progress. In effect, such a process performs explicitly what natural evolution does passively, i.e. gradually accumulating novel forms that ascend the complexity ladder.

For example, in a biped locomotion domain, initial attempts might simply fall down. The novelty metric would reward simply falling down in a different way, regardless of whether it is closer to the objective behavior or not. In contrast, an objective function may explicitly *reward* falling the farthest, which likely does not lead to the ultimate objective of walking and thus exemplifies a deceptive local optimum. In contrast, in the search for novelty, a set of instances are maintained that represent the most novel discoveries. Further search then jumps off from these representative behaviors. After a few ways to fall are discovered, the only way to be rewarded is to find a behavior that does *not* fall right away. In this way, behavioral complexity rises from the bottom up. Eventually, to do something new, the biped would have to successfully walk for some distance even though it is not an objective!

At first glance, this approach may seem naive. What confidence can we have that a search process can solve a problem when the objective is not provided whatsoever? Where is the pressure to adapt? Yet its appeal is that it rejects the misleading intuition that objectives are an essential means to discovery. The idea that the objective may be the enemy of progress is a bitter pill to swallow, yet if the proper stepping stones do not lie conveniently along its gradient then it provides little more than false security.

Still, what hope is there that novelty is any better when it contains no information about the direction of the solution? Is not the space of novel behaviors unboundedly vast, creating the potential for endless meandering? One might compare novelty search to exhaustive search: Of course a search that enumerates all possible solutions will eventually find the solution, but at enormous computational cost.

Yet there are good reasons to believe that novelty search is not like exhaustive search, and that in fact the number of novel behaviors is reasonable and limited in many practical domains. The main reason for optimism is that task domains on their own provide sufficient constraints on the kinds of behaviors that can exist or are meaningful, without the need for further constraint from an objective function.

For example, a biped robot can only enact so many behaviors related to locomotion; the robot is limited in its motions by physics and by its own morphology. Although the search space is effectively infinite if the evolutionary algorithm can add new genes (like NEAT or tree-based GP), the *behavior space* into which points in the search space collapse is limited. For example, after an evaluation, a biped robot finishes at a specific location. Suppose the robot's behavior is characterized only by this ending location. While there are many ways to encode a policy that arrives at a particular point, under this measure of novelty, they *all* collapse to the *same* behavior. In fact, the search space collapses into a manageable number of novelty points, significantly differentiating novelty search from exhaustive enumeration.

Furthermore, novelty search succeeds where objective-based search fails by rewarding the stepping stones. That is, anything that is genuinely different is rewarded and promoted as a jumping-off point for further evolution. While we cannot know which stepping stones are the right ones, if we accept that the primary pathology in objective-based search is that it cannot detect the stepping stones at all, then that pathology is remedied.

A natural question about novelty search is whether it follows any principle beyond naively enumerating all possible behaviors. The answer is that while it does attempt to find all possible behaviors over time, when combined with a complexifying algorithm like NEAT, the *order* in which they are discovered *is* principled and not random. To understand why, recall that NEAT evolves *increasingly complex* ANNs. That way, the amount of nodes and connections and thus the maximal complexity of ANNs discovered by novelty search increases over time, ensuring that simple behaviors must be discovered before more complex behaviors. Note that a similar constraint exists in a more limited form in tree-based GP (an alternate EC paradigm explored with novelty search in chapter 5) because the initial population is initialized with small program trees which can also add new nodes and otherwise grow over generations. Regardless of the particular encoding, this ordering from simple to complex is generally beneficial because of Occam's razor, i.e. the idea that the simplest satisfying description is usually the best, which motivates the minimum description length principle in machine learning [137]. Intuitively, this approach makes sense because the *most* complex behaviors are chaotic and lack regularities.

Thus there *is* an order in the search for novelty; it is just a different one then in fitness-based search. While fitness-based search generally orders the search from low to high fitness, novelty search orders it from low to high *complexity*, which is principled in a different way.

The next section introduces the novelty search algorithm by replacing the objective function with the novelty metric and formalizing the concept of novelty itself.

3.2 The Novelty Search Algorithm

Evolutionary algorithms like NEAT are well-suited to novelty search because the population of genomes that is central to such algorithms naturally covers a wide range of expanding behaviors. In fact, tracking novelty requires little change to any evolutionary algorithm aside from replacing the fitness function with a *novelty metric*.

The novelty metric measures how different an individual is from other individuals, creating a constant pressure to do something new. The key idea is that instead of rewarding performance on an objective, the novelty search rewards diverging from prior behaviors. Therefore, novelty needs to be *measured*.

There are many potential ways to measure novelty by analyzing and quantifying behaviors to characterize their differences. Importantly, like the fitness function, this measure must be fitted to the domain.

The novelty of a newly generated individual is computed with respect to the *behaviors* (i.e. *not* the genotypes) of an *archive* of past individuals whose behaviors were highly novel when they originated. In addition, if the evolutionary algorithm is steady state (i.e. one individual is replaced at a time) then the current population can also supplement the archive by representing the most recently visited points. The aim is to characterize 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, where k is a fixed parameter that is determined experimentally. Intuitively, if the average distance to a given point's nearest neighbors is large then it is in a sparse area; it is in a dense region if the average distance is small. The sparseness ρ at point x is given by

$$\rho(x) = \frac{1}{k}\sum_{i=0}^{k} \mathrm{dist}(x, \mu_i), \tag{3.1}$$

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. The nearest neighbors calculation must take into consideration individuals from the current population and from the permanent archive of novel individuals. Candidates from more sparse regions of this behavioral search space then receive higher novelty scores. It is important to note that this novelty space cannot be explored purposefully, that is, it is not known *a priori* how to enter areas of low density just as it is not known a priori how to construct a solution close to the objective. Therefore, moving through the space of novel behaviors requires exploration. In effect, because novelty is measured relative to other individuals in evolution, it is driven by a coevolutionary dynamic.

If novelty is sufficiently high at the location of a new individual, i.e. above some minimal threshold ρ_{min}, then the individual is entered into the permanent archive that characterizes the distribution of prior solutions in behavior space, similarly to archive-based approaches in coevolution [36]. Note that the threshold ρ_{min} changes dynamically during evolution to match the difficulty in generating novelty by search over time. If 2,500 evaluations pass and no new individuals have been added to the archive, the threshold is lowered by 5%. If over four are added in the same amount of evaluations, it is raised by 20%. The current generation plus the archive give 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 explicit objective.

It is important to note that novelty search resembles prior diversity maintenance techniques (i.e. speciation) popular in evolutionary computation [30, 54, 73, 76, 107]. The most well known are variants of fitness sharing [30, 54]. These also in effect open up the search by reducing selection pressure. However, in these methods, as in fitness uniform selection [78], the search is still ultimately guided by the fitness function. Diversity maintenance simply keeps the population more diverse than it otherwise would be. (Also, most diversity maintenance techniques generally measure genotypic diversity as opposed to behavioral diversity

[30, 107].) In contrast, novelty search takes the radical step of *only* rewarding behavioral diversity with no concept of fitness or a final objective, inoculating it to traditional deception. In other words, it is logically impossible for novelty search to be deceived *by* the objective because the novelty search algorithm ignores the objective completely. Nevertheless, it is of course important to acknowledge that ignoring the objective as novelty search does will not *always* facilitate the objective's discovery. Also, although novelty search cannot be deceived by the objective, other notions of deception might be argued to impact novelty search, e.g. dimensions of behavior that intuitively seem relevant to a domain might sometimes prove deceptively irrelevant in practice. However, the success of intuitive behavior characterizations in the experiments in this dissertation give some evidence that in practice characterizing behavior is not too challenging.

It is also important to note that novelty search is not a random walk; rather, it explicitly maximizes novelty. Because novelty search includes an archive that accumulates a record of where search has been, backtracking, which can happen in a random walk, is effectively avoided in behavioral spaces of any dimensionality.

The novelty search approach in general allows any behavior characterization and any novelty metric. Although generally applicable, novelty search is best suited to domains with deceptive fitness landscapes, intuitive behavioral characterization, and domain constraints on possible expressible behaviors.

One might argue that this approach simply replaces one objective function with another, because novelty is being maximized just as fitness was. However, conceptually the two are different; novelty creates a gradient of behavioral difference, whereas fitness creates a gradient towards the objective. Maximizing fitness is done with the intent of bringing the search towards the goal (i.e. a specific region of the search space), while maximizing novelty is done without any concept of where the search should terminate or even what general direction it should take within the search space. In other words, novelty search does not describe a *point* (i.e. the objective) in the search space at which the search is aimed; it is a relative measure that is undefined outside the context of a particular search. In other words, the novelty of an individual cannot be compared *between* searches, or even among different generations *within* a particular novelty search.

Once objective-based fitness is replaced with novelty, the NEAT algorithm operates as normal, selecting the highest-scoring individuals to reproduce. Over generations, the population spreads out across the space of possible behaviors, continually ascending to new levels of complexity (e.g. by expanding the neural networks in NEAT) to create novel behaviors as the simpler variants are exhausted. The power of this process is demonstrated in this dissertation through several experiments with surprising results. The first set of such experiments, which are conducted in a maze-navigation domain, are described in the next chapter.

CHAPTER 4
INITIAL MAZE EXPERIMENTS

As an easily-understood example domain of when searching for the objective can prove misleading, this chapter presents experiments in evolving controllers with NEAT for robots that must navigate deceptive mazes. I presented work from this chapter at Artificial Life XI [101] and elaborated upon that conference paper in an article published in Evolutionary Computation journal [102].

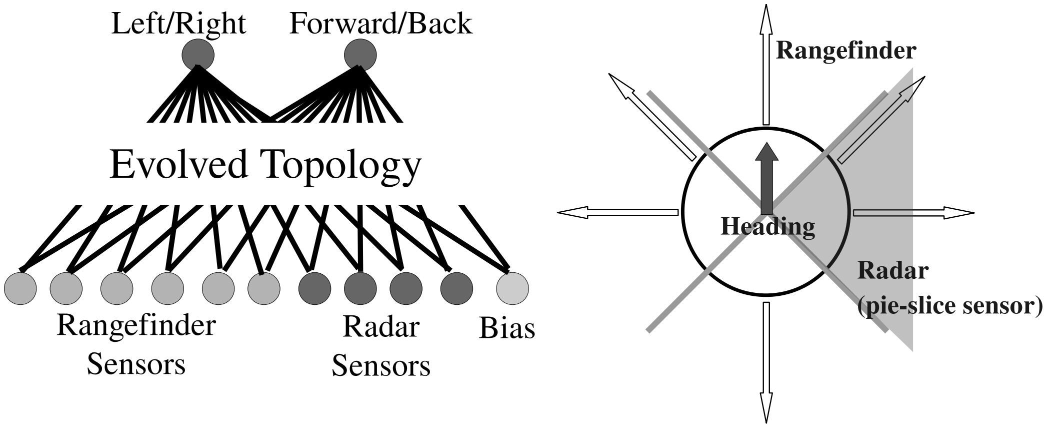
4.1 Maze Experiment

An illustrative domain for testing novelty search should have a deceptive fitness landscape. In such a domain, a search algorithm following the fitness gradient may perform worse than an algorithm following novelty gradients because novelty cannot be deceived with respect to the objective; it ignores objective fitness entirely.

A compelling, easily-visualized domain with this property is a two-dimensional maze navigation task, wherein a robot must navigate through a maze to a chosen goal point. A reasonable fitness function for such a domain is how close the maze navigator is to the goal at the end of the evaluation. Thus, dead ends that lead close to the goal are local optima to which an objective-based algorithm may converge, which makes a good model for deceptive problems in general. Moreover, by varying the structure of the maze and the starting and goal point of the robot, various classes of problems can be modeled (e.g. removing the exterior walls of a maze results in a more unconstrained problem). Because of these favorable properties, the maze navigation domain introduced in this chapter is often used within this dissertation as a representative example of deception or adapted to model particular kinds of domains. For example, in Chapter 5 maze navigation is revisited with a different search algorithm, while in Chapter 8 it is used both to model unconstrained domains and extended to act as a representative example of a more complex domain.

However, the main idea of the experiment in this chapter is to contrast the performance of NEAT with novelty search with that of traditional fitness-based NEAT. In other words, the experiment is a first exploration into the effect in deceptive domains of changing the reward scheme to novelty search, the method this dissertation introduces.

In more detail, the maze domain works as follows. A robot controlled by an ANN must navigate from a starting point to an end point in a fixed time. The task is complicated by cul-de-sacs that prevent a direct route and that create local optima in the fitness landscape. The robot (figure 4.1) has six rangefinders that indicate the distance to the nearest obstacle and four pie-slice radar sensors that fire when the goal is within the pie-slice. The robot's two effectors result in forces that respectively turn and propel the robot. This setup is similar to the successful maze navigating robots in NERO [151].

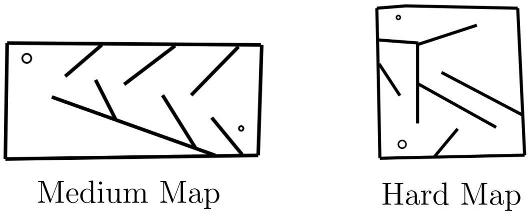


(a) Neural Network

(b) Sensors

Figure 4.1: 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). Each arrow outside of the robot's body in (b) is a rangefinder sensor that indicates the distance to the closest obstacle in that direction. The robot has four pie-slice sensors that act as a compass towards the goal, activating when a line from the goal to the center of the robot falls within the pie-slice. The solid arrow indicates the robot's heading.

Two maps are designed to compare the performance of NEAT with fitness-based search and NEAT with novelty search. The first (figure 4.2a) has deceptive dead ends that lead the robot close to the goal. To achieve a higher fitness than the local optimum provided by a dead end, the robot must travel part of the way through a more difficult path that requires a weaving motion. The second maze (figure 4.2b) provides a more deceptive fitness landscape that requires the search algorithm to explore areas of significantly lower fitness before finding the global optimum (which is a network that reaches the goal). Note that this task is *not* pathfinding. Rather, NEAT is searching for an *ANN* that itself can navigate the maze, which is more challenging and less direct (i.e. it is not immediately obvious how to construct an



Medium Map

Hard Map

Figure 4.2: Maze Navigation Maps. In both maps, the large circle represents the starting position of the robot and the small circle represents the goal. Cul-de-sacs in both maps that lead toward the goal create the potential for deception.

ANN that can successfully navigate a maze, while the path that a robot can take to reach the goal in either of the two mazes in this chapter is readily apparent).

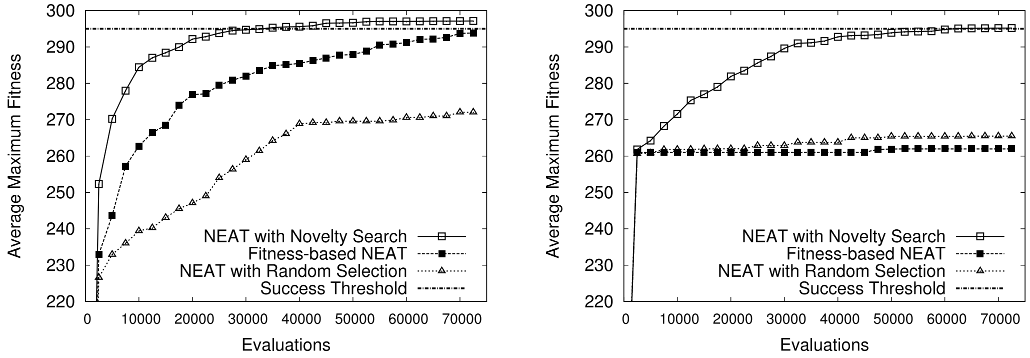
Fitness-based NEAT, which will be compared to novelty search, requires a fitness function to reward maze-navigating robots. Because the objective is to reach the goal, the fitness f is defined as the distance from the robot to the goal at the end of an evaluation: $f = b_f - d_g$, where b_f is a constant bias and d_g is the distance from the robot to the goal. Given a maze with no deceptive obstacles, this fitness function defines a monotonic gradient for search to follow. The constant b_f ensures all individuals will have positive fitness.

NEAT with novelty search, on the other hand, requires a novelty metric to distinguish between maze-navigating robots. Defining the novelty metric requires careful consideration because it biases the search in a fundamentally different way than the fitness function. The novelty metric determines the behavior-space through which search will proceed. Thus, it is important that the types of behaviors that one hopes to distinguish are recognized by the metric.

Therefore, because what is important in a maze is where the traverser ends, for the maze domain, the behavior of a navigator is defined as its ending position. The novelty metric is then the squared Euclidean distance between the ending positions of two individuals. For example, by such a metric two robots stuck in the same corner appear similar. In contrast, a robot that simply sits at the start position looks very different from one that reaches the goal, though both behaviors are equally viable to the novelty metric.

The effect of this novelty metric is to reward the robot for ending in a place where none have ended before; the method of traversal is ignored. This measure reflects that what is important is reaching a certain location (i.e. the goal) rather than the method of locomotion. Thus, although the novelty metric has *no knowledge* of the final goal, a solution that reaches the goal can appear novel. In addition, the comparison between fitness-based and novelty-based search is fair because both scores are computed only based on the distance of the final position of the robot from other points. Furthermore, NEAT is given exactly the same settings in both (see Appendix), so the only difference is the reward scheme.

Finally, to confirm that novelty search is indeed not anything like random search, NEAT is also tested with a random fitness assigned to every individual regardless of performance, which means that selection is random. In all cases, if the maze is solved, the number of evaluations is recorded.



(a) Medium Map (b) Hard Map

Figure 4.3: Comparing Novelty Search to Fitness-based Search. The change in fitness over time (i.e. number of evaluations) is shown for NEAT with novelty search, fitness-based NEAT, and NEAT with random selection on the medium (a) and hard (b) maps, both averaged over 40 runs of each approach. The horizontal line indicates at what fitness the maze is solved. The main result is that novelty search is significantly more effective. Only the first 75,000 evaluations (out of 250,000) are shown because the dynamics remain stable after that point.

4.2 Maze Results

On both maps, a robot that finishes within five units of the goal counts as a solution. On the medium map, both fitness-based NEAT and NEAT with novelty search were able to evolve solutions in every run (figure 4.3a). Novelty search took on average $18,274$ evaluations ($sd = 20,447$) to reach a solution, while fitness-based NEAT was three times slower, taking $56,334$ evaluations ($sd = 48,705$), averaged over 40 runs. This difference is significant ($p < .0001$). NEAT with random selection performed much worse than the other two methods, finding successful navigators in only 21 out of 40 runs, which confirms the difference between novelty search and random search.

Interestingly, the average genomic complexity (i.e. number of connections in the ANN) of solutions evolved by fitness-based NEAT for the medium map (66.74 connections, $sd = 56.7$) was almost three times greater ($p < 0.05$) than those evolved by NEAT with novelty search (24.6 connections, $sd = 4.59$), even though both share the same parameters. In other words, novelty search discovers more elegant solutions to the medium maze. What is particularly interesting is that genomic complexity is a representational property that is not directly subject to selection in either fitness-based search or novelty search; however, such reduced genomic complexity of evolved solutions for novelty search is also seen in chapters 5 and 6, hinting that it may generally hold true.

On the hard map, fitness-based NEAT was only able to evolve a successful navigator in three out of 40 runs, while NEAT with random selection fared marginally better, succeeding in four out of 40 runs, showing that deception in this map renders the gradient of fitness no more helpful than random search. However, novelty search was able to solve the same map in 39 out of 40 runs, in $35,109$ evaluations ($sd = 30,236$) on average when successful, using 33.46 connections on average ($sd = 9.26$). Figure 4.3b shows this more dramatic divergence. Remarkably, because the second maze is so deceptive, the same NEAT algorithm can almost never solve it when solving the maze is made the explicit objective, yet solves it almost every time when finding novel behavior is the objective.

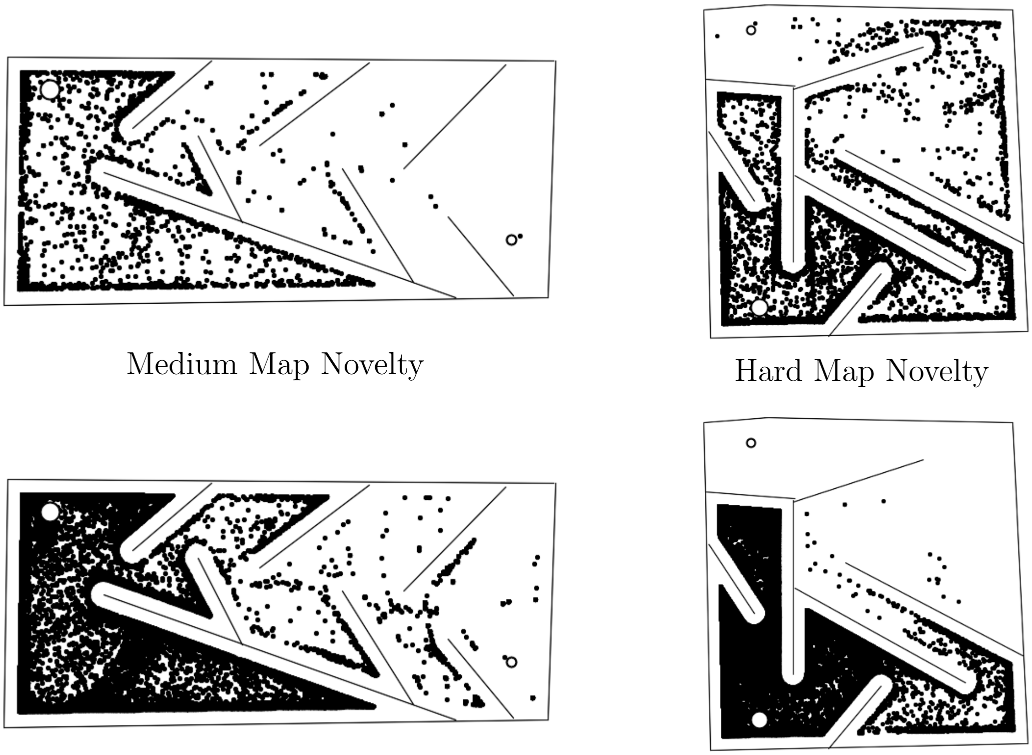
4.2.1 Typical Behavior

Figure 4.4 depicts behaviors (represented as the final point visited by an individual) discovered during typical runs of NEAT with novelty search and fitness-based NEAT on each map. Novelty search exhibits a more even distribution of points throughout both mazes because behaviors are explicitly rewarded for diverging from those that have already been encountered. On the other hand, Fitness-based NEAT shows areas of density around local optima in the maze, reflecting evaluations wasted in areas of the search space that appear superficially promising to an objective-driven search. The typical behavior of a successful robot on either maze was to directly traverse the maze for both methods.

The results so far raise a number of important questions and concerns, which the the next section addresses.

4.3 Additional Maze Studies

To better understand important components of novelty search such as the archive and the choice of behavior characterization, additional experiments are detailed in this section. In particular, first the effect of bounding the size of the archive to limit computational cost is investigated. Then, the effect is measured of applying different behavior characterizations that conflate maze navigation behaviors to various degrees and in different ways. The



Medium Map Novelty

Hard Map Novelty

Medium Map Fitness

Hard Map Fitness

Figure 4.4: Final Points Visited Over Typical Runs. Each maze depicts a typical run, stopping at either 250,000 evaluations or when a solution is found. Each point represents the end location of a robot evaluated during the run. Novelty search is more evenly distributed because it is not deceived.

motivation is that investigating such variation may provide general insight into potential limitations of novelty search or how to pick appropriate behavioral characterizations for particular domains.

4.3.1 Bounding the Size of the Archive

A possible concern about the computational effort required to search for novelty is that the archive of past behaviors may grow without limit as the search progresses. As the size of the archive grows, the nearest-neighbor calculations that determine the novelty scores for individuals become more computationally demanding. This occurs because a new individual's novelty can potentially be affected by any of the individuals in the archive. While the implementation of novelty search in these experiments calculates the nearest-neighbors naively, there exist specialized data structures such as kd-trees that can significantly improve performance [13]. Although in most complex domains the evaluation of individuals will likely be the computational bottleneck, it is true that the nearest-neighbor calculation increases the amount of computation beyond that required for a simple objective-based search method. Thus, it is interesting to consider ways in which the archive's size may be limited.

A simple approach is to change the structure of the archive from an ever-expanding list to a queue of limited size. When a new individual is added to the archive once the archive has reached its full size, the earliest entry is overwritten by the new individual instead of the new individual being appended to the end of the list. While this approach may lead to some backtracking through behavior space (which the archive is designed to avoid), the amount of backtracking may be limited because the dropped behaviors may no longer be *reachable* from the current population. That is, the population as a whole may have thoroughly explored the particular area of behavior space represented by the dropped individual and progressed sufficiently beyond it. In such a situation, there may be no likely sequence of mutations from any individual in the current population to the particular area of behavior space once occupied by the overwritten archive point, such that the intermediate steps also appear novel. Also, interestingly, some degree of backtracking may be beneficial to the search for novelty: If a necessary stepping stone reachable from a certain behavior was missed during novelty search, dropping the behavior from the archive may give search a second chance to discover the missed stepping stone.

To explore the effects of limiting the archive size, 40 additional runs of novelty search were conducted in the hard maze with the size of the archive limited to the same size as the population (250). The hard maze was solved in all 40 runs, in $38,324$ evaluations ($sd = 42,229$) on average, which is slightly worse but not significantly different from the original results of novelty search on the hard maze without a bounded archive. This result demonstrates that it in some domains it is possible to limit the archive size, and thus the additional computational effort, without significantly decreasing the performance of novelty search.

4.3.2 Conflating Behaviors in the Maze

As noted earlier, behavior was characterized in the maze experiment as the location of the maze-navigating robot at the end of an evaluation. Although this characterization is only an approximation of the complete behavior of the navigator (which would include the navigator's location and heading at every timestep of the simulation), it reflects that what is important in solving a maze is reaching a particular location (i.e. the goal).

Because the behavioral characterization is an approximation, all navigators that end at the same point are *conflated* as equivalent even if they take entirely different routes. Thus they appear identical to the novelty search algorithm. Intuitively, a natural assumption is that in the maze domain, this conflation makes the search for novelty more efficient: By conflating individuals that end at the same location, the behavior space is greatly *reduced* and many possible uninteresting meandering behaviors that end at the same location are not rewarded for being different.

An important implication of this assumption is that lengthening the behavioral characterization (e.g. by taking multiple samples of the navigator's location during an evaluation) might render the search for novelty intractable because the number of possible behaviors grows exponentially as the dimensionality of the characterization increases. On the other hand, if the behavioral characterization does not include *enough* information, search might also be hindered: If a behavior that is a stepping stone is conflated with an already-discovered behavior, it will receive a low novelty score and may thus be discarded. It is also interesting to consider characterizing behavior by the fitness measure, which conflates all behaviors that would receive the same fitness score; if fitness as a behavioral characterization were always effective then it would be trivial to apply novelty search to any domain that employs an objective function (applying novelty search in this way is very similar to fitness uniform selection [78]).

However, as explained in Section 3.1, the hypothesis in this dissertation is that intuitive assumptions about the effect of the size of the behavior space do not take into account the effect of complexification and therefore wrongly predict that novelty search should fail in high-dimensional behavior spaces. Rather, the general issue of behavioral conflation is interesting because it provides a potential bias that can be exploited to increase performance and also gives insight into when a search for novelty may be successful. Three additional experiments in the maze domain are thus conducted to explore the effect of behavioral conflation by increasing the behavioral characterization's dimensionality, decreasing the amount of information in the behavioral characterization, and characterizing behavior by the fitness measure.

4.3.2.1 High-Dimensional Behavior Space

To increase the dimensionality of the behavioral characterization in the maze domain, the location of the navigator is sampled multiple times during an evaluation. For example, the dimensionality of the behavioral characterization in Section 4.1 is two, which includes the navigator's x and y coordinates at the end of the evaluation. If instead 200 samples of the navigator's location are taken during an evaluation (i.e. one every other timestep), the effective dimensionality of the behavioral characterization becomes 400, which entails a vast space of behaviors. The question is whether an algorithm that simply searches for behavioral novelty can hope to find the objective behavior in such a large behavior space.

To answer this question, additional runs of novelty search in the hard maze were conducted with behavioral characterizations of various sizes. In each such characterization, k equally-spaced samples of the navigator's (x, y) position are taken during an evaluation. Each (x_i, y_i) pair corresponds to the ith sample of the navigator's position in the maze. The resulting behavioral characterization is then the vector that concatenates all of these pairs, where $1 \leq i \leq k$, to form $(x_1, y_1, x_2, y_2, ..., x_k, y_k)$. The novelty metric is unchanged, i.e. it remains the sum of the squared distances between the controllers' behavioral characterization vectors. The only difference is that these vectors can now be longer (i.e. up to 400-dimensional).

Figure 4.5 shows the results of novelty search with different characterization dimensionalities. Contrary to intuition, even in the 400-dimensional behavior space, novelty search still consistently finds controllers that reach the goal in the hard maze *without any significant decrease in performance*. In fact, the added information of the higher-dimensional characterization even slightly improves performance. Novelty search is largely unaffected by the vastness of the 400-dimensional behavioral characterization and is not lost exploring endless variants of uninteresting behaviors because of the underlying order of the search for novel behaviors imposed by a complexifying algorithm, from simple to more complex. A simple neural network with few connections cannot encode a complex policy so erratic meandering behaviors are not initially encountered. In this way, genomic complexity places an upper bound on behavioral complexity. Because NEAT begins evolution with minimal neural networks and slowly grows them during evolution, simple behaviors are encountered before complex behaviors. As the behavioral characterization becomes higher-dimensional, conflation is reduced, but chaotic behaviors with no guiding principle still require a high level of complexity to be expressed in an ANN controller. Thus, such unprincipled policies will be encountered *later* in search than a simpler policy that exploits regularities in the environment (such as walls) and sensory feedback to reach the goal. In summary, this experiment supports the hypothesis that a high-dimensional behavior characterization is *not* a sufficient basis for predicting that novelty search should fail.

4.3.2.2 Increased Conflation through Reduced Precision

The previous experiment demonstrated that reducing the amount of behavioral conflation in the hard maze did not hurt novelty search's performance. However, what if conflation is *increased* such that some navigators that end in different locations within the maze nonetheless receive *identical* behavioral characterizations? This idea would effectively reduce the size of the behavior space and would provide an upper bound on the possible size of the archive (see Section 4.3.1). The idea is to study the effect of reduced information on novelty search; the danger of conflation is that a necessary stepping stone may not be rewarded if

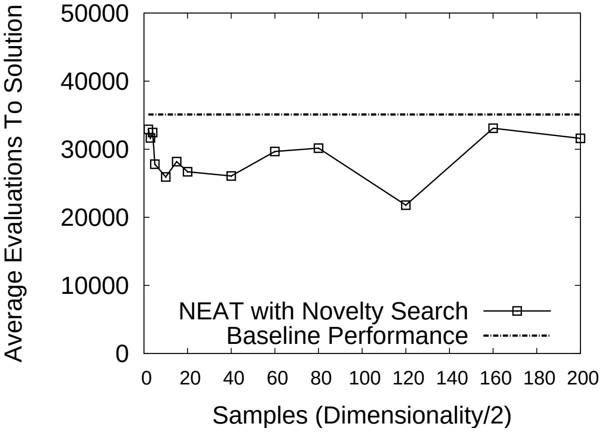
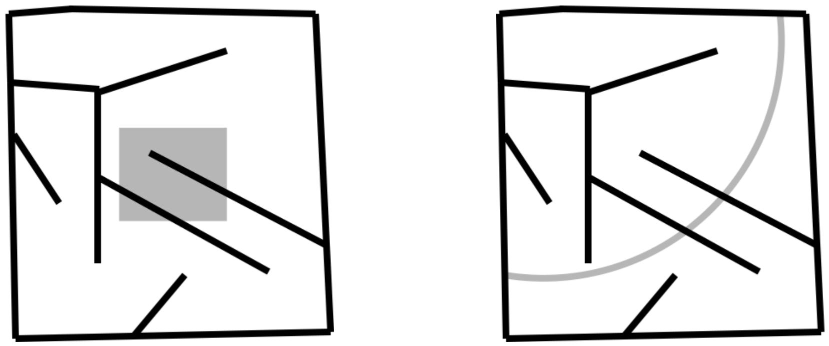


Figure 4.5: Increasing the Dimensionality of the Behavioral Characterization. The average number of evaluations taken to solve the task is shown for different numbers of samples of the navigator's location (i.e. different behavioral characterization dimensionalities) taken during an evaluation of NEAT with novelty search on the hard maze. For comparison, the baseline performance when the behavior characterization is only the final point visited is shown as a separate line. The dimensionality of the behavioral characterization is twice the number of samples because a location is composed of x and y coordinates. Each point represents the average of 40 runs. The main result is that even with a behavioral characterization with 200 samples (400 dimensions), the performance of novelty search is largely unaffected. The hard maze was solved in under $500,000$ evaluations in each of the 520 runs.

a conflated behavior is discovered and thoroughly explored earlier in search. Thus the new behavior, which is a useful stepping stone, might not be further explored.

To investigate this issue, the hard maze is discretized into a two-dimensional grid such that any two navigators that end in the same grid square are assigned the same behavioral characterization (i.e. the characterization is the position of the center of the grid square). This kind of behavioral conflation is illustrated in figure 4.6a. By increasing the size of the grid squares, the robustness of novelty search with respect to increasing conflation can be measured. In effect, as the grid becomes coarser, the resolution of the behavioral measure decreases. For example, if the grid is two by two, then only four distinct behaviors are recognized by the characterization; all behaviors are thus mapped onto one of these four grid squares. In this context, the original behavioral characterization for the maze domain had effective resolution of two single-precision floating point numbers (24 bits each) and thus could be approximately represented by a 2^{24} by 2^{24} grid. Conversely, the most trivial yet interesting grid that can be constructed is a two-by-two grid. To obtain a representative sampling of this wide range of grid sizes, 24 different behavioral characterizations are constructed. Of these, the ith characterization is the center of the final grid square encountered by the navigator on a 2^i by 2^i grid overlaid on the hard maze. Note that these 24 characterizations cover an immense range of grid sizes, from two-by-two to 2^{24} by 2^{24}. For each of these 24 characterizations, 40 additional runs of novelty search on the hard maze were conducted. If the maze is not solved within $500,000$ evaluations, novelty search is restarted and the evaluations continue to accumulate until the maze is solved.

The results (figure 4.7) indicate that for all but the smallest grids (two-by-two and four-by-four), the performance of novelty search is largely unaffected. Only four runs (one in the two-by-two grid and three in the four-by-four grid) out of 960 total runs failed to solve the hard maze in under $500,000$ evaluations and required a restart (which counted towards the average number of evaluations). The worst-performing grid (four-by-four) recognized only 16 behaviors and required on average $117,107$ evaluations to solve, which is about three times slower than the original results, but still performed better than fitness-based search, which usually does not solve the hard maze at all. This result is surprising because it



(a) Reducing Information (b) Characterizing Behavior by Fitness

Figure 4.6: Comparing Conflation Across Different Behavioral Characterizations. Typical conflations of behavior are indicated in gray for reducing the amount of information in the behavioral characterization (a) and characterizing behavior by fitness (b). In (a) only geographically similar behaviors are conflated by a rectangle that is a part of a regular grid, whereas in (b), behaviors that end in very different locations (and are likely specified by very different policies) are conflated by the circle centered on the goal with radius equal to a particular fitness value.

might be expected that discretizing the behavioral characterization would create plateaus in behavior space that would reduce novelty search in those areas to random search. However, even in these cases, as soon as the next plateau is discovered, novelty search then favors exploration there. Even if these plateaus are very large (as in the two-by-two grid), the search process will eventually escape. Furthermore, the behaviors are conflated in a sensible way: Although nearby behaviors appear identical, distant behaviors still appear distant. Thus, novelty search still works well with large amounts of such behavioral conflation because the stepping stones are still respected, although at a lower resolution. This result suggests again that the archive size can be limited (because reduced grid sizes also reduce the amount of unique individuals in the archive, e.g. with a two-by-two grid only four unique archive individuals could exist), thereby limiting its computational expense without significant loss

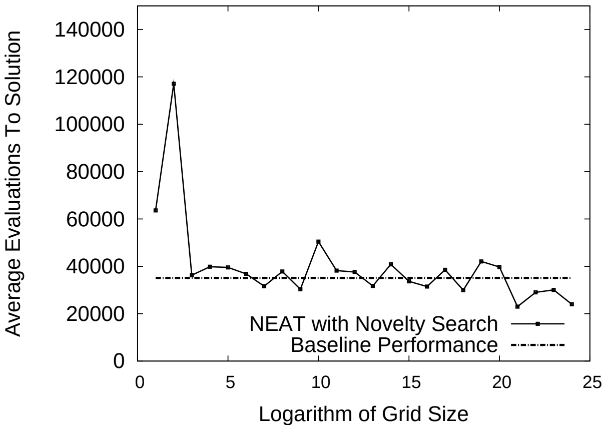


Figure 4.7: Reducing the Amount of Information in the Behavioral Characterization. The average number of evaluations taken to solve the hard maze is shown for behavioral characterizations that purposefully conflate behaviors that end in the same square of a virtual grid overlaid uniformly on the maze. Each point represents the average of 40 runs. The dotted line indicates the baseline performance at full resolution. The main result is that even with a behavioral characterization that only recognizes four behaviors (2^1 by 2^1), novelty search still can consistently solve the maze that fitness-based search is nearly always unable to solve.

of performance. What would happen, however, if distant behaviors were sometimes conflated with nearby behaviors? The next section examines this question.

4.3.2.3 Characterizing Behavior as the Fitness Measure

In traditional single-objective evolutionary algorithms, fitness measures closeness to the objective and in effect summarizes behavior with a single number. It is interesting to consider also how fitness itself conflates behavior. In the maze domain, fitness measures how close to the goal location the navigator ends. Thus, from a fitness standpoint, a navigator that ends at a specific point in the maze is conflated with all navigators that end on the circle of points that are at the same distance from the goal (figure 4.6b). Therefore, some behaviors are conflated that are very different from each other (i.e. they end at points separated by a large distance). In this way, the stepping stones may *not* be respected; if dead-end behaviors are discovered that mask distant stepping stones, the search for novelty may fail. To test this hypothesis, a series of runs with fitness (i.e. the distance from the navigator's ending position to the goal) as the behavioral characterization were conducted in the maze domain. This kind of novelty search (with a behavioral characterization of fitness) is very similar to fitness-uniform selection [78]. The results validate the hypothesis that this type of conflation can be disruptive to novelty search: the maze was solved in only 11 out of 40 runs (which is still better than fitness-based NEAT's performance, however, which only solved the hard maze in four out of 40 runs). This result suggests that characterizing behavior by the fitness measure may be a simple way to improve performance over an objective-based method in a deceptive domain, although carefully constructing a more appropriate behavioral characterization may yield better results.

4.3.2.4 Conflation Summary

The conclusion is that behavioral conflation is complex, and cannot be characterized simply by the idea that more is worse, and less is better. It is more informative to think of conflation in terms of stepping stones. Conflation is harmful to the search for novelty if behaviors are conflated in a way that interferes with the discovery of stepping stones. Such harmful conflation can arise when characterizing the behavior of a navigator in terms of its fitness: By thoroughly exploring behaviors near the easily-reachable cul-de-sac in the hard maze, a quagmire of low novelty is created in a ring through the corridor that ultimately leads to the goal location. Conversely, if a robot can vary behavior in a way that is completely *orthogonal* to the objective, it would be better to conflate all such orthogonal dimensions of behavior rather than to explore these less informative dimensions.

The next section discusses the implications of the initial studies of novelty search described in this chapter.

4.4 Implications

The initial experiments in this chapter demonstrate concretely that novelty search can sometimes circumvent the problem of deception in EC. The hard maze is a problem for which the gradient of fitness is unhelpful because it will consistently deceive search into local optima. However, novelty search was *not* deceived because it pursued many gradients of novelty simultaneously, and was able to solve both mazes consistently. Higher performance is achieved in the maze when the only incentive given to NEAT is to produce individuals that are different functionally from those seen before.

The additional studies in the maze domain addressed potential objections to novelty search: High-dimensional behavior characterizations do not necessarily stymie a search for novelty, nor does reducing the precision of the behavioral characterization. Also, the archive size may sometimes be limited without affecting performance. These studies also revealed insights into the important role of behavioral conflation in novelty search.

While these initial experiments begin to establish the problem with the objective-based paradigm dominant in EC and how novelty search can sometimes circumvent it, naturally questions on the generality of such results and the scalability of novelty search remain. While the additional experiments with a fixed-size archive and high-dimensional behavior characterizations hint at novelty search's robustness, the next chapter more definitively demonstrates the generality of novelty search through several independent means.

CHAPTER 5
GENERALITY OF NOVELTY SEARCH

This chapter provides evidence of novelty search's generality, i.e. its ability to apply across domains and methods, by applying it to experiments in an alternate EC paradigm and examining how its performance scales with increasing problem complexity. Further evidence is provided by reviewing additional novelty search results by other researchers.

5.1 GP Experiments

The last chapter presented foundational novelty search experiments with neuroevolution, which is a particular approach to evolving controllers within EC. One aspect of establishing novelty search's generality is to show that it can apply to EC paradigms other than neuroevolution. Thus this chapter investigates novelty search experiments that evolve controllers with genetic programming (GP; [93]), which is an approach to EC that differs significantly from neuroevolution. The key difference between neuroevolution and GP is their disparate representations; neuroevolution evolves ANNs while GP evolves computer programs. In particular, the GP algorithm applied in this chapter evolves LISP-like program trees.

As in neuroevolution, deception is a prominent problem in GP that often causes premature convergence to local optima [99, 104, 141, 49]. Such premature convergence causes genetic diversity to be extinguished before the objective is discovered, which leaves the search trapped in a dead end. So a divergent search technique like novelty search, which may often circumvent deception, is also of interest to researchers in GP.

Interestingly, another significant issue in GP is that of *program bloat* [98, 149, 144], wherein program trees tend to accumulate a vast amount of non-functional nodes, slowing search and wasting computation. Importantly, sometimes the seemingly disparate problems of deception and bloat are connected; selection in a fitness-based search converged to a local optimum can favor buffering a genome against behavioral change by accumulating introns. In this way, crossover and mutation are less likely to affect functional nodes, which can only decrease fitness from a locally-optimal point in the search space. However, because novelty search continually selects for phenotypic divergence, selecting for novelty does not similarly favor such bloat; behavioral change is exactly what *is* selected for, which indirectly favors unbloated representations in which mutations readily result in behavioral change. Thus novelty search may also help with the problem of bloat in GP. This hypothesis is also investigated by the experiments in this chapter.

Good domains for testing novelty search in GP follow the same motivation as those in neuroevolution: They should have deceptive fitness landscapes. In such domains, search algorithms following the gradient of fitness may perform worse than the search for novelty because novelty search cannot be deceived; it ignores the objective entirely.

For this reason, maze navigation is as appropriate a choice for testing novelty search with GP as it was in neuroevolution, and allows for an intuitive comparison with the results from the previous chapter. Also, there is precedent for similar maze domains in GP [149]. To further test novelty search in GP in this chapter it is also applied to another deceptive GP domain, the artificial ant problem, which is also a common GP benchmark [93, 134, 24, 99]. In this domain an ant controlled by an evolved program attempts to navigate a trail collecting food. It is so deceptive that GP often performs little better than random search [99].

The GP experiments in this chapter were conducted with a version of the lilGP genetic programming distribution [177] extended with an implementation of novelty search that is generational as in Mouret [117], as opposed to the steady-state implementation introduced previously in chapter 3.

5.1.1 GP Maze Experiment

The GP maze domain works as follows. A robot controlled by a genetic program must navigate from a starting point to an end point within a fixed amount of time. The task is complicated by occlusions and cul-de-sacs that prevent a direct route and create local optima in the fitness landscape. The robot can move forward, turn, and act conditionally based on whether there is a wall directly in front of it or not. A robot is successful in the task if it reaches the goal location. This GP setup is purposefully analogous to the neu-

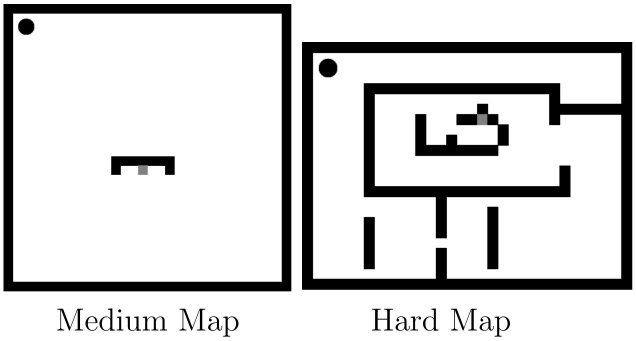


Figure 5.1: GP Maze Navigation Maps. In both maps, the circle represents the starting position of the robot and the grey square represents the goal that the robot attempts to reach. Black squares are walls. Cul-de-sacs and occlusions in both maps that prevent direct passage to the goal create the potential for deception.

roevolution maze experiment in the previous chapter. Table A.3 in the Appendix describes the parameters of the experiment.

Two maps are designed to compare the performance of GP with fitness-based search and GP with novelty search, as was done in the previous chapter but with NEAT. The first (figure 5.1a) has an occlusion that blocks the most direct path to the goal. To reach the goal, the robot must learn a behavior unrelated to the locally optimal behavior of simply crashing into the occlusion. The second maze (figure 5.1b) provides a more deceptive fitness landscape that requires the search algorithm to explore areas of significantly lower fitness before finding the global optimum (which is a network that reaches the goal).

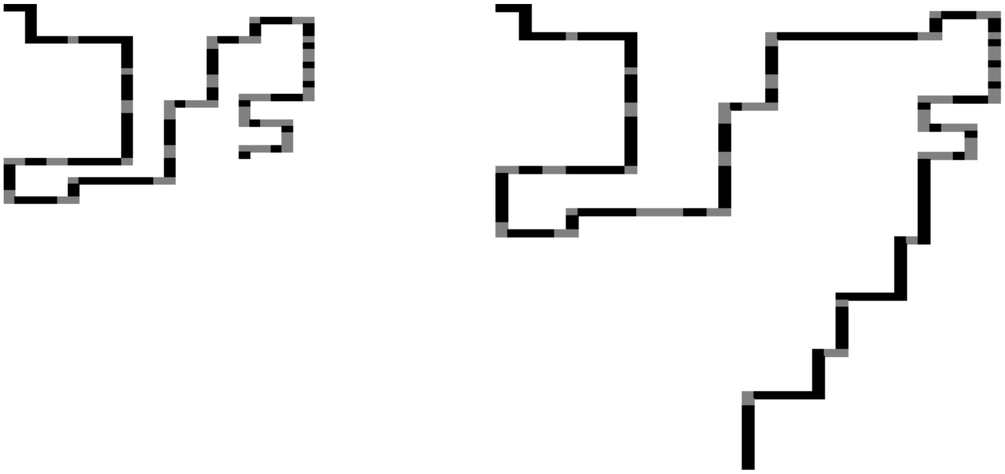
Fitness-based GP, which will be compared to novelty search, requires a fitness function to reward maze-navigating robots. Because this experiment is derived from the maze domain in the last chapter, the same fitness function applies. Recall that the fitness f is defined as the distance from the robot to the goal at the end of an evaluation: $f = b_f - d_g$, where b_f is the maximum distance possible and d_g is the distance from the robot to the goal. Given a maze with no deceptive obstacles, this fitness function defines a monotonic gradient for search to follow. The constant b_f ensures all individuals will have positive fitness.

GP with novelty search, on the other hand, requires a behavior characterization and novelty metric to distinguish between maze-navigating robots. For consistency, the same characterization and metric as in the neuroevolution maze domain are applied, i.e. the behavior of a maze navigator is defined as its ending position and the novelty metric is the Euclidean distance between the ending positions of two individuals. In this way two robots crashing into the same cul-de-sac appropriately appear similar, and two robots ending at opposite corners within the maze will appear different.

The next section describes the other deceptive GP domain, the artificial ant, in more detail.

5.1.2 Artificial Ant Experiment

In the artificial ant problem, a simulated ant embedded in a toroidal two-dimensional grid attempts to collect as much food as possible [93, 134, 24, 99]. The food is laid out in a trail in the grid, but there are gaps in the trail such that the agent must *infer* the missing steps of the trail. The problem is deceptive because policies with no principle can nonetheless capture



Santa Fe Trail Los Altos Trail

Figure 5.2: Artificial Ant Trails. In both trails, the ant starts in the upper left hand corner. Black squares represent food that the ant attempts to collect, while grey squares illustrate gaps in the trail that the ant must infer.

many units of food by chance although they are not ultimately precursors to a policy able to capture *all* the food. The ant is controlled by a genetic program that allows it to turn, move, and conditionally act based upon whether food is in front of it or not. The parameters of the artificial ant experiment are described in table A.4 in the Appendix.

The artificial ant experiment in this paper follows the formulation in Koza [93] and is attempted on both the Sante Fe and Los Altos trails. The Santa Fe trail is shorter, while the Los Altos trail is longer and harder to follow. In both maps, a successful individual is one that collects all of the food.

Fitness-based GP requires a fitness function to reward ants. Because the objective is to collect all of the food in the trail, the fitness function is the amount of food that the ant collects, which is customary on this problem [93].

GP with novelty search requires a novelty metric to distinguish between ant behaviors so that novelty can be quantified. A simple way to characterize behavior in this domain is to sample how much food has been collected by the ant over several evenly-spaced intervals. This measure frames behavior in terms of collecting food, the task of interest; a novel individual must do something *different* with respect to collecting food.

Sampling this value over time allows search to differentiate behaviors that ultimately collect the same amount of food by different means. It is important to note that characterizing behavior as the amount of food collected does *not* mean that novelty search will necessarily seek higher values of food collected, but rather *different* temporal sequences of food collected during an evaluation. The novelty metric is the same as in the maze domain (i.e. the Euclidean distance between two behavioral characterization vectors).

5.1.3 GP Parameters

All experiments were run with a modified version of the lilGP package with the standard settings according to Koza [94], including tournament selection. The number of generations was significantly extended from the formulation in Koza [93] from 50 to $1,000$ so that long-term program growth patterns could be observed.

Because GP with novelty search differs from traditional GP only in substituting a novelty metric for a fitness function, they share the same parameters from Koza [93] except that novelty search is given a smaller tournament size (two), which reduces selection pressure so that what is novel does not drastically change from generation to generation. It is important to note that when traditional fitness-based GP was given a smaller tournament size in preliminary experiments, overall performance was worse and the dynamics of program growth were unaffected.

The number of neighbors checked in novelty search, k, was set to 25, and is robust to moderate variation. An individual has a 0.05% chance to be added to the archive, which means a new archive point is added every two generations on average.

In the artificial ant problem, the amount of food the ant collected was sampled eight times during an evaluation to construct the behavioral characterization vector, and proved robust to moderate variation during initial experiments.

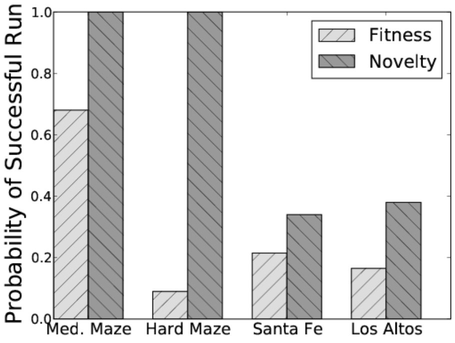


Figure 5.3: Performance Comparison. For each variation of the test domains, the proportion of the 200 runs that are successful are shown for fitness-based GP and GP with novelty search. The main result is that GP with novelty search evolves solutions more consistently than fitness-based GP.

5.1.4 GP Results

Two hundred trials were run for each variation of the maze and artificial ant problems with both fitness-based GP and GP with novelty search. As illustrated by figure 5.3, in each task novelty search solves the task significantly more often than fitness-based GP ($p < 0.001$).

Supporting one of the hypotheses in this chapter, that novelty search may be less prone to the problem of program bloat in GP, an analysis of evolved solutions shows that for each of the domains, novelty search tends to evolve smaller program trees (figure 5.4). Note that only individuals from runs that solve the problem are included in these averages and that only the first evolved solution is considered. The differences are significant for each domain ($p < 0.01$).

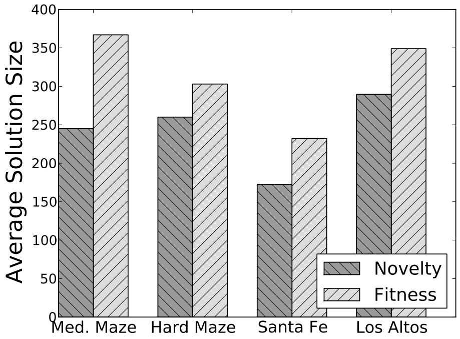
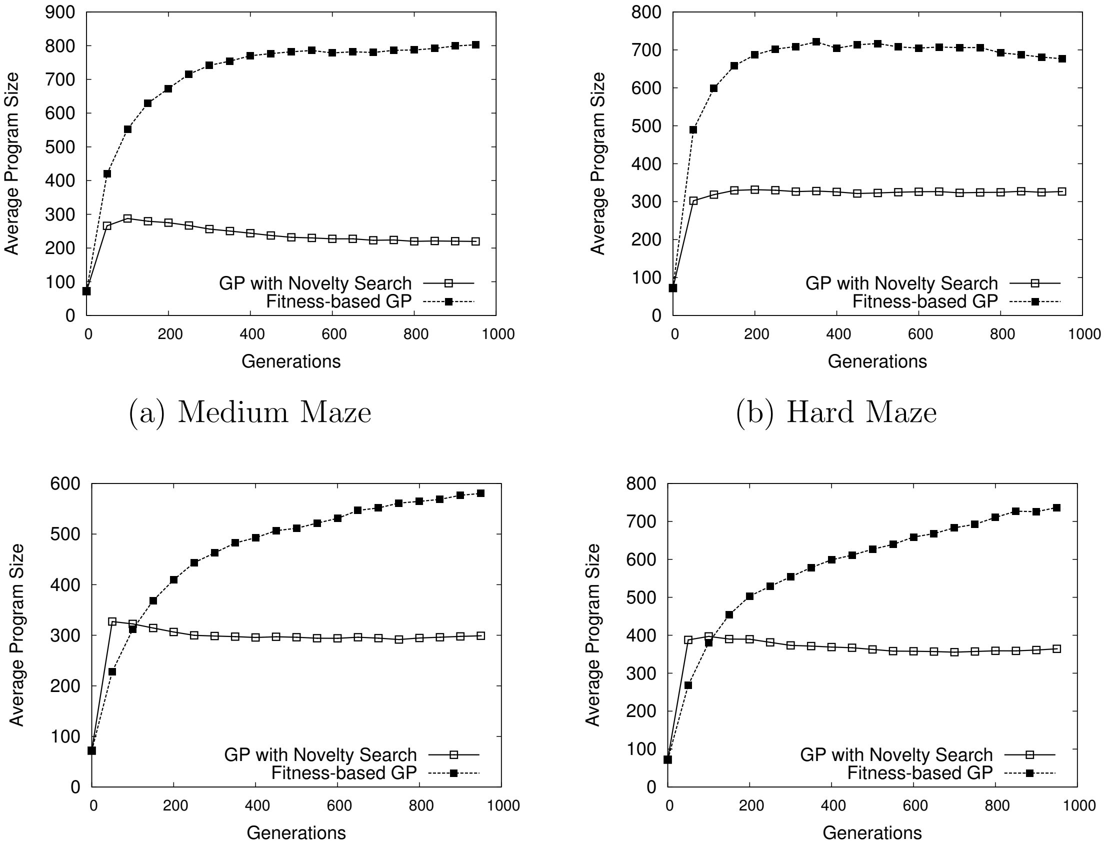


Figure 5.4: Bloat Comparison. For each variation of the test domains, the average size (number of nodes) of solutions evolved by GP with novelty search and fitness-based GP is compared (lower is better). In every variation, novelty search evolves significantly smaller program trees.

The dynamics of program growth during evolution are shown in figure 5.5. Interestingly, instead of growing larger as in fitness-based GP, program length stagnates and sometimes even *declines* in runs of GP with novelty search.

5.1.5 GP Interpretation

The failure of fitness-based GP on the harder maze might lead one to conclude that the Koza-style GP algorithm is at fault; however, the problem is solved consistently when all is kept constant except that behavioral novelty is rewarded instead of estimated progress to the objective. This reconfirms the conclusion of the neuroevolution maze experiments in the



(a) Medium Maze

(b) Hard Maze

(c) Santa Fe Trail

(d) Los Altos Trail

Figure 5.5: Program Growth During Search. The average size (number of nodes) of a program tree in the population is shown as evolution progresses for GP with novelty search and fitness-based GP on the medium (a) and hard (b) mazes, and the Santa Fe (c) and Los Altos (d) ant trails, averaged over the first 25 runs of each approach. The main result is that populations do not continually bloat with novelty search.

previous chapter: The problem is not in the search algorithm itself but in *how the search is guided.*

In GP as well as neuroevolution, the problem of deception proves difficult for objective-based search to overcome. Novelty search, by virtue of not searching directly for the objective, is not deceived and is therefore sometimes more successful at solving such deceptive problems.

However, the claim is not that GP with novelty search will always be better than an objective-based GP algorithm; as in neuroevolution, there will be cases where fitness is not deceptive and easily maximized or the behavior space will be immense and without restrictions. Yet as problems increase in complexity and difficulty, as will be explored in the next section with mazes of parametrized difficulty, it becomes increasingly difficult to craft a sufficiently accurate heuristic that does not deceive search by leading it to locally optimal dead ends [47, 176].

It is also interesting that novelty search evolves smaller solutions than fitness-based search in every domain. Because bloat is a prominent problem in GP [98, 149, 144], this result further recommends novelty search for GP. Furthermore, the growth curves in figure 5.5 demonstrate that the reduced bloat in novelty search is not a simple consequence of it solving the problems faster. Rather, novelty search effectively constrains bloat throughout the runs in every domain. The reason is that while static fitness functions promote bloat to protect program trees from destructive crossover [98], in novelty search this "protective" bloat would actually be *maladaptive*. That is, bloat buffers *against* behavioral change, while in the search for novelty, behavioral change is rewarded. Tentative evidence for this hypothesis is found in Schmidt and Lipson [144], where a coevolutionary GP algorithm (which also changes what is rewarded as search progresses) also evolved smaller programs when compared to GP with a static fitness function.

The successful application of novelty search to GP, following success in neuroevolution adds to evidence for the generality of the search technique itself and hints again at the limitations of the objective-based paradigm which is currently dominant in EC. The next section investigates how well novelty search and objective-based search comparatively scale up to solve increasingly complicated problems.

5.2 Scaling Problem Complexity

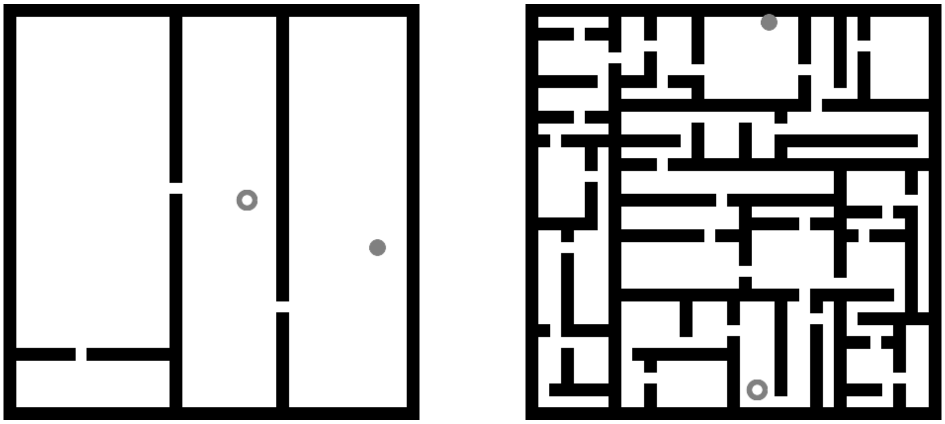
A hypothesis advanced by this dissertation is that as problems grow more difficult, the gradient defined by measuring distance to the objective becomes increasingly deceptive and thereby less informative. Thus as deceptiveness increases, non-objective search methods like novelty search may outperform more traditional objective-based search methods. However, while not susceptible to traditional deception, novelty search also is not guaranteed to consistently find *specific* objectives as problems become more complex.

Therefore, an interesting experiment is to compare how the relationship between problem complexity and performance varies in both traditional objective-based search and novelty search, which serves as an example of a non-objective search algorithm. Maze navigation is a natural choice of domain for such an investigation because it is a good model for search problems in general and because it is easy to generate mazes of parametrized complexity.

5.2.1 Scaling Experiment

The same GP maze setup applied in the previous section provides the basis for this experiment as well. The general idea is that a maze navigating robot must navigate to the goal in a maze. Fitness-based GP and GP with novelty search are to be compared over a wide range of mazes with varying complexity. The fitness function for fitness-based GP, and the behavior characterization and novelty metric for GP with novelty search are the same as in the previous section. Table A.5 in the Appendix describes the parameters of the experiment.

To compare how effectively fitness-based search and novelty search evolve navigational policies for increasingly complex maze problems, both search methods were tested on 360 randomly-generated mazes. These mazes were created by a recursive division algorithm [133], which divides an initially empty maze (i.e. without any interior walls) into two subareas by randomly adding a horizontal or vertical wall with a single randomly-located hole in it (which makes all open points reachable from any other open point in the maze.) This process continues recursively within each subarea until no areas can be further subdivided without making the maze untraversable, or until the limit for subdivisions (chosen randomly between 2 and 50 for each maze in this experiment) is exceeded. The starting position of the maze navigating robot and the goal position it is trying to reach are also chosen randomly. Examples of mazes generated by such recursive division are shown in figure 5.6.



(a) Simple Maze Problem

(b) More Complex Maze Problem

Figure 5.6: Two randomly-generated maze problems created by the recursive division algorithm. In both mazes, the filled circle represents the starting location and the unfilled circle represents the goal location. The maze shown in (a) has fewer subdivisions and a shorter optimal path to the goal than the maze shown in (b).

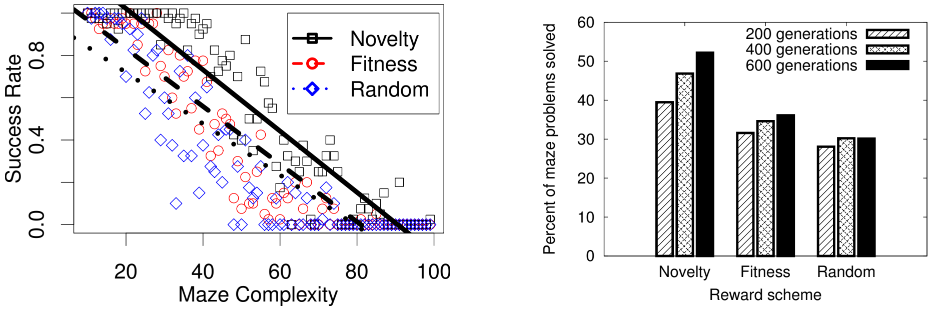
The length of the shortest possible path between the start and goal position was found to be a good heuristic for problem complexity. Intuitively, longer paths potentially require more complex navigational policies. In addition, increasing path length was highly correlated with decreasing performance for all of the search methods (adjusted $R^2 > 0.75$ for each method). Thus mazes were sampled such that 4 maze problems were chosen for each shortest-path length between 10 and 100. For each of the 360 mazes, 10 independent runs were conducted for both fitness-based search, novelty search, and GP with random selection. Random selection was considered as a control to differentiate novelty search from random exploration of the search space. Experiments were conducted with limits of 200, 400, and 600 generations. A given run is considered successful if a navigator was evolved that reaches the goal within the time limit of 200 steps.

5.2.2 Scaling Results

The main result, as illustrated by figure 5.7, is that novelty search solves significantly more instances of the generated maze problems ($p < 0.001$, Fischer's exact test) and that it scales to solving more complex instances significantly better than objective fitness-based search or random search ($p < 0.001$, the intercept values of the linear regression models are significantly different according to an ANCOVA test). In addition, figure 5.7b shows that novelty search better exploits additional evaluations than fitness-based search or random search. While random search may waste many evaluations with policies that are the same and fitness-based search may waste many evaluations attempting to escape from deceptive local optima, novelty search constantly incentivizes discovering new behaviors.

It is important to note that the performance of each of the three compared methods decreases with increasingly complex maze instances. Few instances are reliably solved by any of the methods with optimal path length greater than 80. Thus while novelty search may outperform the other methods in this domain, it too struggles to discover *specific* ambitious objectives from first principles; this result tentatively supports the hypothesis that in some cases attempting to achieve specific objectives from a random starting point may ultimately be futile beyond a certain level of problem complexity.

This experiment adds to the accumulating evidence presented in this chapter of novelty search's generality because it better scales to difficult instances within the maze – although



(a) Performance Comparison

(b) Performance versus Evaluations

Figure 5.7: **Reward scheme comparisons.** The effectiveness of novelty search, fitness-based search, and random search at solving problems of increasing complexity is plotted along with linear regressions in (a). Novelty search is the most effective although its performance also degrades with increasing problem complexity. Each plotted point is from ten runs, each lasting 600 generations. The effect on performance of varying the amount of generations for novelty search, fitness-based search, and random search is shown in (b). Novelty search exploits additional evaluations more effectively than fitness-based search or random search.

it too suffers as problems get more difficulty. In addition to results that I have personally collected, the next section reviews results with novelty search from other researchers.

5.3 Results from other Researchers

The neuroevolution maze results from chapter 4 were also reproduced in combination with a multi-objective EA by Mouret [117]. His results revealed that a raw novelty search would discover an approximate solution most quickly, while optimizing fitness in addition to novelty would more quickly discover a more exact solution. This result makes sense because once novelty search discovers an individual in the neighborhood of a solution, further optimization from that point is no longer subject to deception. That is, optimizing from such a point is easy because the gradient is not misleading, yet reaching that point in itself is difficult without exploring novel behaviors.

In further experiments in a deceptive four-bit boolean classification problem and a ball-gathering evolutionary robotics domain Ollion and Doncieux [120] show that the most promising approach is novelty search coupled with pressure to optimize the objective. The general conclusion might be that such combination may often prove practical although as the previous section suggests, as objectives get more ambitious even such coupling will likely fail.

Novelty-related methods have also been shown beneficial in evolving plastic neural networks that learn from experience by Risi et al. [136]. In a simulated T-maze similar to those employed to investigate learning in rats, novelty search was more consistently able to evolve robot behaviors that learned from experience to solve the task. The main idea is that stepping stones on the way to networks that successfully learn from experience may not themselves be successful. For example, a behavior that consistently moves *away* from a large reward may be very similar genotypically to a desired behavior of running *towards* it; however, such a behavior may appear superficially undesirable to an objective function and thus may likely be pruned away by objective-based search. Similarly, for tasks that require adaptation to solve consistently, the best *non-adaptive* solutions (i.e. those that employ a mediocre fixed policy that solves the task only in some cases) can act as deceptive local optima that objective-based search may converge to. Thus, novelty search may often prove advantageous in training plastic neural networks because of its immunity to both of these kinds of deception common in adaptive tasks and its general tendency to accumulate stepping stones.

Krcah [97] compared fitness-based search and novelty search in two domains evolving both the morphology and control policy of virtual creatures in a three-dimensional physically-realistic simulator. In a simple non-deceptive swimming task, fitness-based search slightly outperformed novelty search. However, in a deceptive barrier-avoidance task novelty search was consistently able to solve the task while fitness-based search failed consistently.

Additionally, other works have attempted to analyze when novelty search may be particularly appropriate or ineffective [91, 28]. In particular, Kistemaker and Whiteson [91] attempt to characterize what factors are critical to novelty search's success or failure to solve a particular problem. They suggest that one such factor is what types of genomes are conflated to the same point in behavior space; ideally, points with the same potential to evolve to a solution should be conflated. This insight makes sense when applying novelty search to practical problems although its relevance is less clear to domains with no clear objective.

There have also been further results from other researchers in evolutionary paradigms other than neuroevolution. In GP, Doucette [43] discovered benefits for novelty search in the artificial ant benchmark and maze navigation. Also, Goldsby and Cheng [55] showed that novelty search with GP can help to find latent bugs in software models. Additionally, in results with evolutionary strategies (ES), Cuccu et al. [29] demonstrate that a novelty-based approach can help in some optimization problems. The main idea is that when evolution fails it may be a good idea to restart in a novel area of the search space.

The conclusion from reviewing other researchers' work with novelty search is that novelty search is applicable to a wide variety of domains and evolutionary approaches, and that despite its relatively recent introduction other researchers in EC have already begun to be influenced by it.

5.4 Implications

The evidence presented in this chapter demonstrates the accumulation of novelty search results spanning different EC paradigms, problem difficulties, and researchers. Such an accumulation lays to rest the criticism that novelty search's applicability is limited to specific mazes or that it is specific only to neuroevolution. Furthermore, this accumulation of results helps establish novelty search as a viable new practical tool for EC researchers. Most importantly, however, the generality of novelty search reinforces the hypothesis that there may be profound limitations to the objective-based paradigm that is dominant in EC.

Interestingly, these results also hint that there may often be secondary benefits (i.e. not relating only to solving a problem) that result from applying novelty search. In the GP domains, it was demonstrated that novelty search was not as prone to the problem of bloat, where evolved programs tend to accumulate non-functional nodes. Likewise, in the maze neuroevolution experiments in Chapter 4, novelty search evolved more compact solutions. A hypothesis in this dissertation is that such results are characteristic of a general tendency for novelty search to encourage evolvable representations. That is, selecting for novelty may indirectly select for lineages better able to generate novelty. In particular, compact, non-bloated representations may be more suited to creating novelty. More comprehensive evidence for this hypothesis is explored in Chapter 7, where evolvability is quantified and compared between fitness-based search and novelty search.

Though the results in this chapter demonstrate novelty search's generality, one might still wonder if a search technique without an objective can effectively solve more challenging, real-world tasks. Thus to address such a concern, the next chapter applies novelty search to biped walking, a difficult problem in machine learning still subject to active research.

CHAPTER 6
EVOLVING BIPED WALKING GAITS WITH NOVELTY SEARCH

While in this dissertation novelty search has been shown successful in many domains, it has not been applied to a difficult problem of significant real-world interest. Thus this chapter describes novelty search experiments in such a challenging domain, evolving walking gaits for an unstable biped robot. Note that the experiment described in this chapter was first published in Evolutionary Computation journal [102].

6.1 Biped Experiment

While the initial maze experiments described in chapter 4 and the further diversity of results presented in the previous chapter illustrate that the search for novelty can often circumvent deception, a reasonable question is whether novelty search can successfully be applied to more difficult real-world domains.

Thus there is a need to test novelty search on a well-known problem of considerable difficulty. The intuition behind such an attempt is that novelty search may succeed in such a domain because problem difficulty is generally correlated with deceptiveness. That is, what makes a particular problem intractable to objective search is often that the gradient defined by the objective function nearly always leads to local optima. Because novelty search ignores the objective, it may be able to succeed even in difficult domains simply by searching for novelty. The challenge domain in this chapter is biped locomotion, a difficult control task that is popular within machine learning [162, 130, 67, 111]. The key issue is whether the evolutionary algorithm can overcome the deception inherent in the domain. Furthermore, the problem confronting novelty search is that the space of behaviors is far greater than in domains such as the hard maze, and that the solution, whatever gait is chosen, significantly more brittle from the need for balance and oscillation [86].

In this domain, a three-dimensional biped robot in a realistic physics simulation is controlled by a type of ANN called a *continuous time recurrent neural network* (CTRNN) that is able to express the non-linear dynamics found in natural gaits and is common in other biped experiments [130, 111]. The objective is to walk as far as possible within a given time limit. The task is difficult because it requires coordination, balance, and the discovery of oscillatory patterns. Initial random controllers for biped robots tend to provide a bad gradient for search because they and all of their immediate perturbations tend to simply fall [162]. Thus, even if oscillatory behavior is discovered (which could be useful), it is penalized and therefore ignored for falling down.

To mitigate deception, most methods that evolve controllers for biped robots implement domain-specific biases like enforcing symmetry [67, 123, 119, 3] or oscillation [67, 81], simplifying the task [111, 12], or initially making the task easier [3, 162, 81]. These biases mitigate the deceptiveness of walking by restricting the search and introducing domain knowledge. However, they also impose a priori expectations on the creative evolutionary process. For example, imposing a constant oscillatory period may make the first few steps awkward and the restriction precludes considering alternatives. Thus to make the problem as difficult and general as possible so that novelty search is forced to consider the entire space of possible behaviors, unlike typical approaches to biped walking, *no domain-specific biases* are implemented in the experiment in this dissertation.

The biped domain works as follows. A biped robot is controlled by an ANN for a fixed duration (15 simulated seconds). The evaluation is terminated if the robot falls or after the allocated time expires. The objective is that the robot travel the greatest possible distance from the starting location.

The ANN that controls the biped has only two inputs, which are contact sensors that signal for each foot whether it is touching the ground. The sparsity of input makes the problem more difficult because the ANN has *no information* on the orientation of the robot or on the current angles of its joints.

The biped robot (figure 6.1) has a total of six degrees of freedom (DOF): two degrees in each hip joint (pitch and roll) and one degree in each knee joint (pitch). Simple sphere-shaped feet make ankle joints unnecessary in this model, although the lack of typical feet or a torso (which could provide a counterbalance) require knees to bend backwards (as in birds) to balance, adding to the challenge of the model.

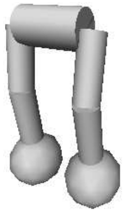


Figure 6.1: Biped Robot. A visualization of the biped robot controlled by an evolved ANN in the experiments in this dissertation.

The ANN outputs movement requests for each degree of freedom (DOF) in the model, i.e. for each independent axis of rotation for all joints in the model. The outputs are scaled to match the angular range of the corresponding DOF, which is interpreted as the angle that the neural controller is requesting. The difference between the requested angle and the current orientation of the DOF denotes the disparity between the state the neural net is requesting and the current state of the model. A proportional controller applies torque to reduce this disparity. In other words, the ANN directs the low-level controllers towards a particular state. This model and method of control are similar to those in Reil and Husbands [130].

Like the maze experiment in Chapter 4, this experiment compares fitness-based NEAT to NEAT with novelty search. It is important to note that NEAT was extended for these experiments to evolve CTRNNs. Parameter settings for NEAT in this experiment (which were the same for fitness and novelty) are given in the appendix. A natural fitness function is the squared distance traveled from the starting location. This distance is measured by recording the location of the center of mass of the biped robot before and after evaluation, then calculating the distance between the two points. Distance traveled is a standard measure among evolutionary biped locomotion experiments [130, 67, 111, 162], and matches the intuitive notion of learning to walk with increasing stability and efficiency.

In contrast, NEAT with novelty search requires a behavioral characterization to distinguish biped gaits. The behavioral characterization in this domain is the offset of the biped's center of mass sampled at one second intervals during the evaluation:

$$x'_k = \operatorname{sign}(x_k - x_0) * (x_k - x_0)^2, \tag{6.1}$$

$$y'_k = \operatorname{sign}(y_k - y_0) * (y_k - y_0)^2, \tag{6.2}$$

where x_0 and y_0 correspond to the initial planar center of mass (i.e. ignoring the vertical z component) of the biped robot, and x_k and y_k correspond to the center of gravity sample taken after the kth second of simulation. The magnitude of the offsets is squared just as the fitness measure is to make the comparison uniform. If the robot falls, the center of gravity for all remaining samples is set to the robot's center of gravity when it fell. The behavioral characterization of a particular controller is then the vector that concatenates all pairs (x'_i, y'_i), where $1 \leq i \leq m$ and m is the final sample taken, to form $(x'_1, y'_1, x'_2, y'_2, ..., x'_m, y'_m)$.

The novelty metric for two different controllers is the same as in chapter 4, i.e. the sum of the squared distances between the controllers' behavioral characterization vectors. However, unlike in those maze experiments, temporal sampling is necessary because a temporal pattern is fundamental to walking. This additional information allows the novelty metric to differentiate two gaits that end up at the same location by different means. However, most importantly, note that novelty search is ignorant of the objective of walking a long distance. In fact, to novelty search, walking clumsily and falling down can be rewarded just as much as walking well.

Furthermore, the comparison between novelty search and fitness-based search is fair because both the fitness function and novelty metric are calculated based on the robot's position. Although the novelty metric observes additional temporal information to differentiate biped gaits, the fitness function could not integrate this extra information without introducing a bias towards a certain style of walking. In contrast, novelty search integrates this information while remaining entirely agnostic about what is good or bad.

6.2 Biped Results

Over 50 runs, novelty search evolved controllers that traveled on average 4.04 meters ($sd = 2.57$) in the allocated 15 seconds while solutions evolved by fitness-based search traveled 2.88 meters ($sd = 1.04$) on average. The difference in performance is significant ($p < 0.01$, Student's t-test). Furthermore, even in the beginning of a run, when one might expect a greedy search to temporarily show advantage, novelty search still performs better (figure 6.2).

More dramatically, the *best* gait discovered by novelty search traveled 13.7 meters, while the best gait discovered by fitness-based search traveled only 6.8 meters. In fact, this latter solution was among only three gaits (in 50) discovered by fitness-based search that traveled a distance over four meters (which was average for novelty search). Videos of these best evolved gaits are available at:

`http://eplex.cs.ucf.edu/noveltysearch/`

Qualitatively, the solutions evolved by both methods were different. A large proportion of runs of novelty search discovered oscillatory gaits (80%), while more than 40% of runs of fitness-based search converged to non-oscillatory gaits (i.e. fewer than four steps) corresponding to deceptive local optima of the objective function.

As in the neuroevolution and GP maze domain from chapters 4 and 5, the genomic complexity of the best walkers evolved by fitness-based NEAT (272.90 connections, $sd = 178.97$) was significantly larger ($p < 0.0001$) than those evolved by NEAT with novelty search (87.52 connections, $sd = 48.22$), even though both share the same parameters.

6.3 Implications

The comparison between novelty search and objective-based search in the biped walking domain shows that novelty search is not only suited for simpler domains like maze navigation

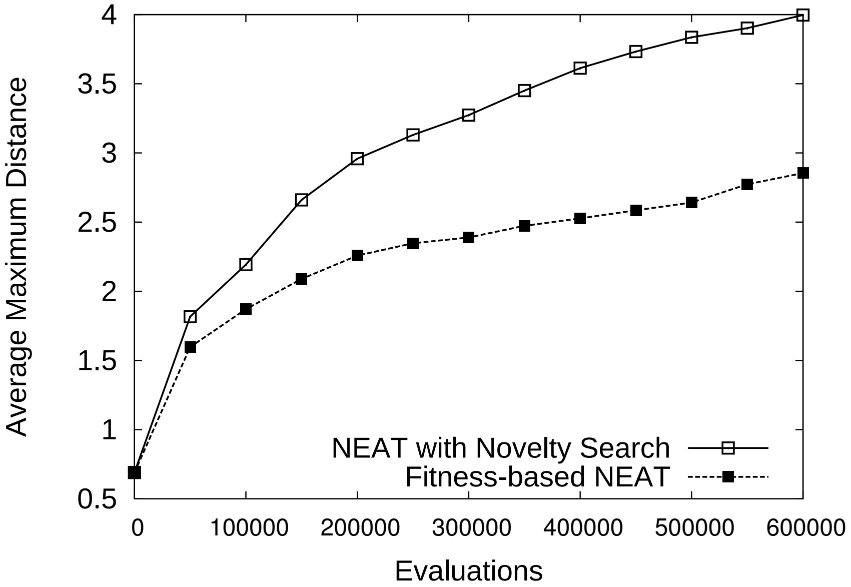


Figure 6.2: Comparing Novelty Search to Fitness-based Search in Biped Walking. The change in distance traveled by the best biped over time (i.e. number of evaluations) is shown for NEAT with novelty search and NEAT with fitness-based search on the biped task. Results are averaged over 50 runs. The main result is that as in the maze domain, novelty search is significantly more effective.

but can also succeed in a much more difficult real-world task. Such a result is important because, like the problem scalability result from the previous chapter, it addresses the common critique of novelty search that it will not scale to more complex problems. That is, because the implications of novelty search contradict common intuitions about search, it is particularly strengthened by succeeding in a popular problem like biped walking. Thus this biped walking result is interesting not only because it further demonstrates the generality of novelty search but also because it makes the intellectual implications of novelty search for objective-based search harder to ignore.

Interestingly, in this domain, as in the neuroevolution and GP maze navigation experiments, novelty search evolves *more compact* solutions than fitness-based search. Because novelty search is not searching consistently and relentlessly to achieve a particular goal, it is not subject to the pressure to continually expand the current-best walker into a better one. That is, evolution's designs may be most elegant when it is not always chained to a specific overarching objective. The idea is that novelty search may discover more elegant representations because it better recognizes the stepping stones torwards which a particular representation leads. In other words, novelty search more naturally expands a given behavior in its most natural directions instead of always forcing it in the same prescribed direction as objective-based search does.

In general, it is possible that novelty search is predisposed to favorable properties of evolved solutions that are nevertheless not directly subject to selection. In other words, there was no selection pressure for small solutions in the maze experiments in Chapter 4, the GP experiments in Chapter 5, or in the biped walking domain in this chapter. Yet in all of these instances, novelty search did find smaller solutions. The next chapter explores this idea further by examining the effect of novelty search on evolvability, another important property of evolved representations that is normally not easily made subject to direct selection pressure.

CHAPTER 7
NOVELTY SEARCH AND EVOLVABILITY

The effect of novelty search on *evolvability*, i.e. the ability of an individual within evolutionary search to *further* evolve, is explored in this chapter. Evolvability is an important property in EC and artificial life because higher evolvability may facilitate evolving more functionally complex individuals. Furthermore, an interesting open question is what causes natural evolution's products to be significantly more evolvable than the frequently fragile representations uncovered by EC. The main insight highlighted by this chapter is that the *type* of evolutionary reward scheme used during search can impact important high-level representational qualities like evolvability evident only over evolutionary time (i.e. not exhibited during a particular evaluation and thereby not *directly* subject to selection).

In particular, novelty search may increase evolvability because it selects for diverging from previous behaviors; thus lineages better at generating novelty, i.e. those that are more evolvable, may be indirectly selected for. In contrast, by its drive to converge to the highest fitness, fitness-based search may sometimes instead select for representations resistant to behavioral change, i.e. that are *less* evolvable. Thus experiments in this chapter quantify and compare the evolvability of individuals evolved by novelty search and fitness-based search.

An interesting secondary discovery in this chapter is that the type of reward scheme can also influence the effectiveness of self-adaptation, where certain parameters influencing reproduction are encoded within an individual's genome. Self-adaptation is a promising idea because it allows evolution to decide parameters that otherwise must be chosen by an experimenter. However, such self-adaptation has not always lived up to its potential in past experiments [26, 95, 50, 139]. A potential reason for past failures is that fitness-based search may exploit self-adaptation to better converge to local optima, an undesirable evolutionary outcome. Indeed, such a hypothesis is supported by the results of the experiments in this chapter.

However, before introducing the experiments, the next section reviews evolvability in natural evolution and also previous studies on evolvability in EC, to put into context the main investigation of this chapter into evolvability,

7.1 Evolvability in Natural Evolution and EC

Natural evolution has discovered flexible, highly evolvable representations that have facilitated the discovery of widely diverse organisms. An important question that could inform EC is what properties of natural evolution led to such evolvability? At the same time, EC enables studies of evolvability that can potentially inform biology through computational experiments impossible in nature.

From a biological perspective, Kirschner and Gerhart [90] describe evolvability as "an organism's capacity to generate heritable phenotypic variation," and suggest that evolvability in natural evolution results from an accumulation of flexible building blocks that are heavily conserved. However, they can be combined and regulated in many ways to yield substantial phenotypic variety with few mutations. Further, they argue that such evolvable lineages may be selected for; a lineage able to discover and exploit new niches or to quickly radiate through existing niches following extinctions will itself be less likely to go completely extinct.

Examining evolvability from both biology and EC, Wagner and Altenberg [165] similarly describe evolvability as relating to the phenotypic variability of a genome, and argue that EC can potentially provide insight into evolvability that biology cannot; the effect of alternate genetic representations on evolvability can be tested within EC but are not easily explored in nature. Furthermore, the authors argue that the structure of the genotype to phenotype mapping is fundamental to evolvability. This mapping, which includes the mechanism for the reproduction and mutation of an organism, is itself subject to selection and evolution in nature.

Studies in EC have described a lack of evolvability in practice [165, 132] and have noted possible directions for increasing evolvability. Some argue that static fitness functions, which are prevalent in most of EC, do not encourage evolvability [132, 44] and instead suggest that fitness functions should change over the course of evolution. Other suggestions for improving evolvability are to allow adaptation of the genetic operator set [59], to increase the extent of neutral networks [44], or to employ indirect encodings that allow more plastic genotype to phenotype mappings [165, 132]. However, prior studies have not examined the effects of alternate reward schemes like novelty search on evolvability, as is undertaken in this chapter.

7.2 Measuring Evolvability

The primary hypothesis in this chapter is that rewarding novelty increases evolvability when compared with rewarding progress towards a fixed objective. Therefore, an important aspect of evaluating this hypothesis is quantifying evolvability.

One definition of evolvability is “an organism’s capacity to generate heritable phenotypic variation” [90]. Brookfield [18] similarly suggests evolvability is “the proportion of radically different designs created by mutation that are viable and fertile.” These definitions reflect a growing consensus in biology that the ability to generate phenotypic variation is fundamental to evolvability [90, 18, 165, 125].

From these definitions, a principled way of estimating an individual’s evolvability is to generate many children from it and then measure the degree of phenotypic variation among those offspring. In effect, this measure quantifies how well the organization of the individual’s representation has internalized domain information to enable more behaviorally diverse mutations. This measure is similar in intention to that in Reisinger et al. [132] but is more granular because it quantifies how well an *individual’s* evolved representation exploits domain structure instead of measuring an *encoding’s* ability to do the same.

Accordingly, the measure of evolvability in this chapter is to count how many distinct behaviors there are among a series of generated offspring of a particular individual. An individual whose offspring tend to display many distinct behaviors is capable of generating much heritable phenotypic variation, and is thus evolvable. In practice, such a measure requires a domain-fitted test between two individuals to determine whether their behaviors are distinct.

The domain-specific novelty metric in novelty search that measures behavioral distance between two individuals can naturally be adapted for this purpose: Two behaviors are distinct if the behavioral distance between them is greater than a fixed threshold.

Using this test of behavioral distinction, the number of distinct behaviors among a list of behaviors can be calculated by the following greedy algorithm that accumulates a list of distinct behaviors: The first behavior is added to this list by default, and each subsequent behavior is added if it is distinct from each behavior already in the distinct behavior list. The size of this filtered list is the number of distinct behaviors, an estimate of the individual's evolvability.

At regular intervals during a run, the evolvability of each individual in the population is measured. That is, for each individual in the population many offspring are sequentially generated by first cloning the individual and then mutating the clone; the idea is to sample the genotypic space around an individual and examine the distribution of those samples in behavior space. The behaviors of these perturbed clones are then concatenated to form a list, with the number of distinct behaviors in the concatenated list acting as an indicator of that individual's phenotypic variability, i.e. its evolvability.

7.3 Evolvability Experiments

To compare the *evolvability* of individuals evolved by both novelty search and traditional objective fitness-based search, experiments are conducted in two domains previously employed in this dissertation to compare the *performance* of the two varieties of search, i.e. maze-navigation (see chapter 4) and biped locomotion (see chapter 6).

In both experiments in this chapter, the evolvability of individuals evolved with NEAT with novelty search is compared to that of traditional fitness-based NEAT. The next sections describe these maze navigation and biped locomotion experiments in detail.

7.3.1 Maze Experiment

The maze navigation domain from Chapter 4 is a good model for behavior spaces in general because it is easy to understand and visualize [102, 117]. Recall that in this domain, a robot controlled by an ANN must navigate in a maze from a starting point to an end point in a fixed time. In particular, the *hard map* from Chapter 4 (figure 4.2b) is used in this experiment.

Like in Chapter 4, the fitness function for fitness-based NEAT rewards a robot's nearness to the goal at the end of an evaluation, reflecting that the objective of the robot is to reach the goal. The behavioral characterization and novelty metric for novelty search is also unchanged from the previous maze experiments. That is, the behavior of a robot is defined as its location in the maze at the end of the evaluation and the novelty metric is the squared Euclidean distance between the ending positions of two individuals. This same novelty metric also distinguishes behaviors when calculating evolvability; two behaviors are distinct if the metric between them is greater than a fixed threshold.

Population size for the hard maze was 250 and each run lasted $1,000$ generations. When measuring evolvability, 200 offspring were created for each individual in the population and evolvability was measured every 50 generations. Solution criteria in the maze domain are the same as in Chapter 4 (i.e. ending within five units of the goal) as are all other parameters (as detailed in the Appendix).

7.3.2 Biped Experiment

A more challenging domain is biped locomotion, which was the subject of Chapter 6. Recall that in this domain, a three-dimensional biped robot (figure 6.1) in a realistic physics simulation is controlled by an ANN with the objective of walking as far as possible. The task is difficult because it requires coordination, balance, and the discovery of oscillatory patterns.

This experiment also compares the evolvability of fitness-based NEAT to NEAT with novelty search. The same fitness function for fitness-based NEAT is used as in Chapter 6, which is the squared distance traveled from the starting location. Likewise, the same behavioral characterization and novelty metric are used for novelty search to distinguish biped gaits, i.e. the offset of the biped's center of mass sampled at one-second intervals during the evaluation. The novelty metric for two different controllers is the same as in the maze domain, i.e. the sum of the squared distances between the controllers' behavioral characterization vectors. Like in the maze evolvability experiment, this novelty metric also acts as a behavioral distinction test when measuring evolvability; two behaviors are distinct if the behavioral distance between them is greater than a fixed threshold.

The population had 500 individuals and each run lasted $2,000$ generations. When measuring evolvability, 200 offspring were cloned from each individual in the population and evolvability was measured every 100 generations. The walker traveling the farthest in each run is considered the solution. Other parameters are the same as the biped experiment from chapter 6 and are listed in the Appendix.

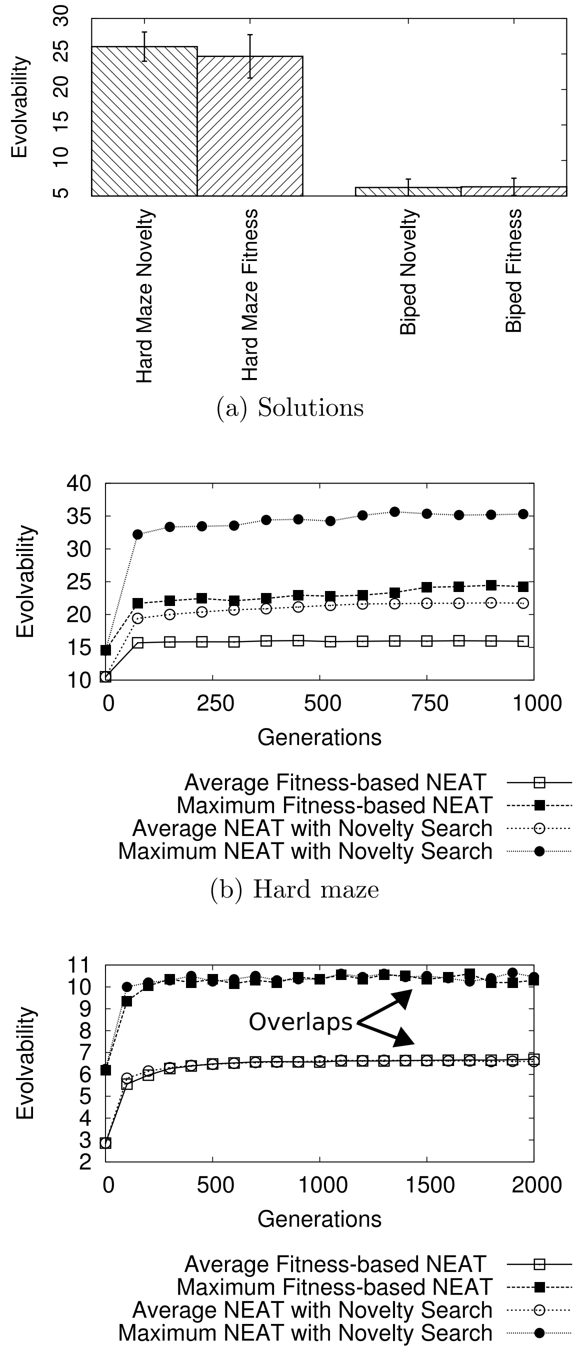
7.4 Evolvability Results

In the hard maze experiment, novelty search discovered solutions in every one of the 20 runs while fitness-based search solved it only once. Thus an additional 580 runs of fitness-based search were conducted to accumulate a more representative sample of solutions. Out of the combined 600 runs, 41 were successful. The average evolvability of *solutions* (figure 7.1a) discovered by novelty search in the hard maze ($\mu = 26.03$, $sd = 2.026$) was significantly higher ($p < 0.05$; Mann-Whitney U-test) than that that of solutions found by fitness-based search ($\mu = 24.76$, $sd = 2.97$). In addition, reflecting evolvability across evolution, after the 25th generation, the evolvability of the most evolvable individual and the average evolvability of individuals (figure 7.1b) in the population were significantly higher ($p < 0.01$; Mann-Whitney U-test) in novelty search than in fitness-based search.

In the biped experiment, novelty search outperformed fitness-based search by evolving walkers that traveled farther on average ($p < 0.01$; Mann-Whitney U-test). Yet, unlike in the maze, the *evolvability* of biped champions (figure 7.1a) was not significantly higher in novelty search than in fitness-based search. Neither were the evolvability of the most evolvable individual or the average evolvability of individuals in the population (figure 7.1c) significantly different between novelty search and fitness-based search (Mann-Whitney U-test).

7.5 Self-Adaptation Experiments

The results in the hard maze conflict with those in the biped domain; though novelty search evolved more evolvable individuals in the maze domain, those evolved in the biped domain were not significantly more evolvable. Further examination shows that individuals evolved



(a) Solutions

(b) Hard maze

(c) Biped

Figure 7.1: Comparing the evolvability of novelty search and fitness-based search. The average evolvability of evolved solutions is shown in (a). The average and maximum evolvability of all individuals in the population are shown for novelty search and fitness-based search in (b) the hard maze and (c) the biped walking experiments. Results are averaged over the initial 20 runs. Error bars indicate standard deviation.

by novelty search and fitness-based search in the biped domain are very fragile; nearly all mutations of such individuals immediately fall, suggesting that the mutation and recombination operators are ill-fitted to the evolved representations.

A potential remedy to such fragility is self-adaptation: Augmenting a genome with additional parameters can allow evolution to adapt parameters governing reproduction such as mutation rate [95, 26]. A genome with such self-adaptive capabilities has greater influence over how its offspring are created. The idea is that evolution can select for self-adaptive parameters that better complement a genome's evolved representation and mitigate domain fragility.

An additional fragile domain is crafted to more generally test the new hypothesis that self-adaptation may encourage evolvability in fragile domains. In the original conception of the maze domain (Chapter 4) there is no penalty for colliding with walls, and all evolved solutions collide with walls as they navigate to the goal. Thus, a more difficult variant of the maze domain is created that terminates an evaluation when an individual collides with a wall; a crashed individual receives a very low fitness (for fitness search) and a dummy behavior characterization to a particular point outside the map (for novelty search). The motivation is to make gradients of both improvement (for fitness-based search) and behavioral change (for novelty search) more brittle, mirroring the inherent brittleness of the biped domain.

To implement self-adaptation in NEAT, genomes are augmented with floating point parameters that influence the mutation of connection weights. Two such parameters are considered here: Connection mutation power, which modulates the magnitude of a potential change to a weight, and connection mutation rate, which specifies the probability that a particular connection weight will be changed at all.

Initial experiments revealed that evolving a single mutation power and rate for each entire genome is too coarse; certain connections in the neural network may be more or less sensitive to perturbation, a scenario impossible to exploit with only a single set of mutation parameters per genome [50]. At the opposite extreme, evolving separate sets of mutation parameters for every connection in an ANN greatly expands the parameter space evolution searches through. A balance is achieved by augmenting NEAT's genome with a fixed-length set of mutation parameters for each genome among which each connection within the ANN can choose. That is, with this kind of self-adaption each connection weight in a genome is mutated according to a particular rule within the set of rules the genome contains. This setup is similar to that employed by Stanley and Miikkulainen [154] in the context of rules for adaptation in plastic networks.

Thus, each genome is given a list of three pairs of mutation settings and each connection is augmented with an integer parameter that indexes within the list. When an individual is mutated, every one of its links are usually mutated by adding to its weight a random number chosen from the same uniform distribution. With self-adaptation, each connection of the neural network may be *exempted* from mutation with probability determined by the connection mutation rate. If it is chosen to be mutated, the power of the mutation is modulated by multiplying it with the connection mutation power. Thus, a self-adaptive genome can potentially craft a mutation distribution that complements the genome's evolved representation.

It is important to note that these self-adaptation experiments do not use crossover, as in some other studies [26, 50]; the reason here is that crossover facilitates mixing of self-adaptive parameters between different lineages, which can overpower weaker lineage-level selection pressure on such parameters that have no immediate impact on fitness [165, 125]. The previous experiments, with the addition of the new fragile hard maze domain, are repeated with self-adaptation to investigate the effect of such self-adaptation on the evolvability of both fitness-based NEAT and NEAT with novelty search.

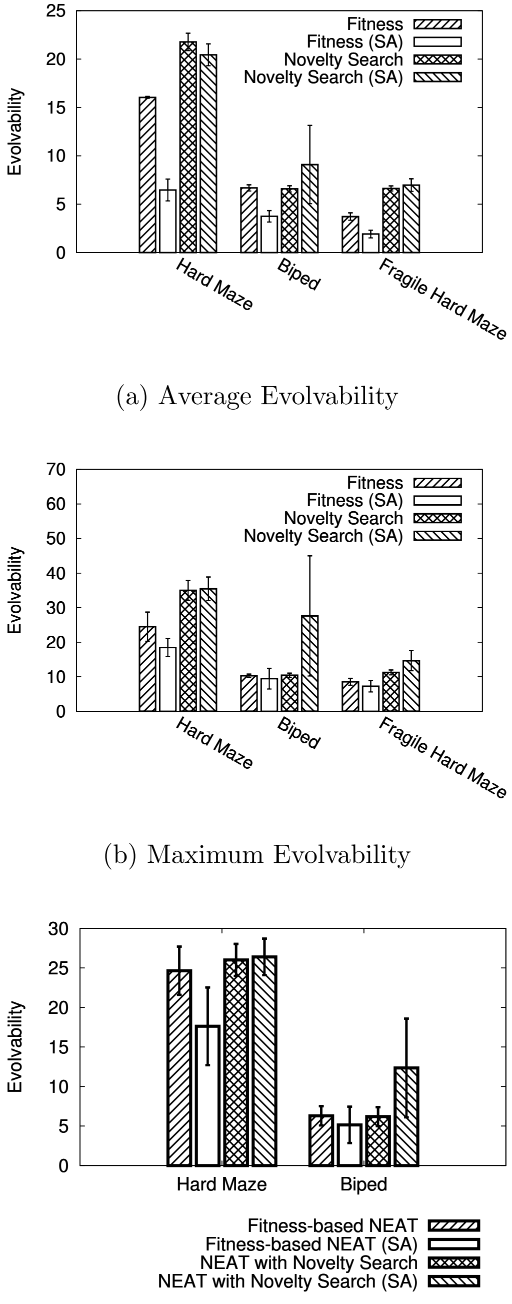
The added self-adaptive rules that influenced mutation of ANN connection weights themselves changed probabilistically at *fixed* rates [26, 95]. During mutation of a genome, a randomly-selected rule would be changed with probability 0.75. Each of the two floating point parameters making up such a rule are then perturbed with probability 0.2 by a random number uniformly selected between -0.2 and 0.2. Each genome also had a 10% chance of changing to which rule a randomly-chosen connection was linked.

The population size for the fragile hard maze was 250 and evolution lasted $2,000$ generations. Evolvability was measured every 100 generations. Other parameters were the same as for the normal hard maze experiment and are listed in the Appendix.

7.6 Self-Adaptation Results

Figure 7.2 shows how self-adaptation effects novelty search and fitness-based search in the hard maze, biped, and fragile hard maze domains. The main result is that self-adaptation allows novelty search to discover significantly more evolvable representations in the biped and fragile hard maze ($p < 0.05$; Mann-Whitney U-test), while combining fitness-based search and self-adaptation is *universally* detrimental across every measure and domain ($p < 0.05$; Mann-Whitney U-test).

In addition, novelty search combined with self-adaptation increased *performance* in the fragile hard maze as well; in 300 additional runs it was solved 31 times by that setup, while it was solved at best only 14 times out of 300 by other setups. The difference is significant ($p < 0.05$; Fisher's exact test). There was no significant difference in the average distance traveled by biped champions between novelty search with self-adaptation and without it, but the two novelty search setups significantly outperformed the two fitness-based setups ($p < 0.05$; Mann-Whitney U-test). Additionally, combining fitness-based search with self-adaptation hurt performance when compared to fitness-based search without self-adaptation ($p < 0.05$; Mann-Whitney U-test).



(a) Average Evolvability

(b) Maximum Evolvability

(c) Solution Evolvability

Figure 7.2: Effects of self-adaptation on evolvability. In all figures, methods with self-adaptation are indicated by the abbreviation *SA*. The average evolvability of all individuals in the final population for each variant are shown in (a), the maximum evolvability of all individuals in the final population are shown in (b), and the average evolvability of discovered solutions are in (c). Results are averaged over 20 runs and error bars indicate standard deviation. The main implication is that self-adaptive parameters help evolvability in novelty search but not in fitness-based search.

7.7 Implications

The results support the idea that novelty search may often discover more evolvable representations than fitness-based search, although the ability of any method to discover evolvable representations is likely stymied by a poor fit between reproductive operators, the representation, and the domain. In addition, while novelty search can effectively mold mutational parameters to match the representations it discovers, fitness-based search instead suffers from such an additional capability.

More significantly, the inability of fitness-based search to exploit self-adaptation and its lesser evolvability may illustrate the greater *conflict* between greedily pursuing the objective and desirable representation. That is, a lack of representational variety prevents discovering *better* representations. This handicap is a significant challenge that evolution may often face when employed as an optimizer; by considering only current fitness, traditional EAs often converge to a particular lineage and its corresponding representation without regard to higher-level properties such as evolvability. Evolution must then elaborate on what may be a poor representation.

What is in question is whether fitness should often be expected to be a reasonable heuristic for quality of representation and evolvability? If the fitness function is static and its landscape sufficiently rugged, then there may be no reason to expect such a correlation [26]. If the fitness function changes over time, depending on its character and the character of the chosen encoding, such correlation may exist [132, 59]. Because what novelty search rewards is highly dynamic, though always incentivizing new phenotypes, it may often encourage discovery of flexible representations and thereby lead to increasing evolvability even if the task is inherently static.

Interestingly, the results in this chapter may suggest revisiting prior disappointing results from testing EAs with self-adaptive mutation rates on non-trivial fitness landscapes [26, 95, 50, 139]. Often in such cases self-adaptive mutation rates are prematurely minimized (which accelerates premature convergence) because mutations leading to further fitness increases become exceedingly rare [26, 95, 50, 139]. In contrast, coupling self-adaptation with novelty search facilitates an *improved* balance between mutation rate and innovation, even in the deceptive and challenging biped locomotion domain. Thus prior experimental results that might be interpreted as illustrating the *limitations* of self-adaptation could be partially misleading. That is, the tendency of self-adaptation to exacerbate premature convergence in prior experiments may be an incidental byproduct of the reward scheme currently dominant in EC.

CHAPTER 8
MINIMAL CRITERIA NOVELTY SEARCH

Novelty search is extended in this chapter to produce a new approach called *minimal criteria novelty search* (MCNS) that serves two purposes. The first is to introduce a new abstraction of natural evolution to EC that interprets evolution as a search for many novel ways to do the same thing, i.e. to survive and reproduce. The motivation is that such an abstraction may potentially better capture natural evolution's prolific creativity than the dominant fitness-based abstraction. The second purpose of MCNS is to provide practical constraints for novelty search in domains where a behavior space is dominated by largely uninteresting areas. The idea is that pruning uninteresting behaviors through applying a set of minimal criteria may also sometimes yield performance benefits.

The next section motivates and describes MCNS in more detail.

8.1 The Minimal Criteria Novelty Search Algorithm

Although natural evolution is *not* overall explicitly a search for novelty, there are several mechanisms in nature that do encourage novelty, both explicitly and implicitly. Explicit novelty search in natural evolution does sometimes result from *negative frequency dependent selection*, in which phenotypically rare individuals are more successful [46, 77]. On a more fundamental level, two properties of natural evolution, *novelty preservation* and *niching*, lead to an implicit search for novelty over evolutionary time scales.

First, novelty in natural evolution is generally preserved as long as a novel organism's lineage continues to meet the minimal criteria for selection. Second, a complementary process, niching, encourages novelty. That is, by founding a new niche (i.e. a new way to make a living), an organism may *avoid* competition and exploit untapped resources [84]. A new niche may also lead to other new niches that are only reachable from or depend upon the precursor niche. The result of preservation and niching is an accumulation of novel lifestyles; thus natural evolution can be abstracted at a high level as a passive kind of novelty search subject to minimal criteria, which is an alternative to the more common interpretation of evolution as an adaptive competition that underpins most of EC.

This view illustrates that biological evolution is not always about competition; selection merely specifies that a lineage that does not meet the minimal criterion of reproduction will end. Competition, either from its own or competing species, is only one reason that an individual may fail to meet this minimal criterion; if the environment precludes an individual from reproducing or mutations render an organism sterile, then its lineage will also lose to natural selection.

The critical functionality of all organisms is their ability to survive until they are able to reproduce. Thus, from this point of view, all successful organisms ultimately have the *same* functionality (i.e. to reproduce) but may execute this functionality in vastly *different* ways (e.g. by being a beet or a bear, but nonetheless reproducing). Furthermore, unlike the objective of a typical fitness function in EC, this functionality has existed since the very *beginning* of evolution.

Contrary to the discredited egocentric view of humanity as the pinnacle of evolution [33, 58, 140, 112], the complexity of macro-scale organisms such as humans is orthogonal to the fundamental evolutionary goal of efficient reproduction. That is, it is unnecessary for an organism to grow from a single cell (the fertilized egg) to the scale of trillions of cells (a developed human), when ultimately the organism's contribution to its offspring is yet again only the size of one cell (i.e. a sperm or egg cell) [34, p. 259]. Thus, the complex development cycle of large animals is a kind of Rube-Goldbergian *digression.* Furthermore, when compared to the offspring of macro-scale organisms, offspring of bacteria are more robust [58] (i.e. they are able to survive in many environments unsuitable for larger animals), and are produced much more quickly (i.e. on the order of minutes instead of years) and efficiently (i.e. without the need to build trillions of additional cells).

Hence, any perceived *forward progress* in evolution is orthogonal to meeting the minimal criterion of reproduction. However, it is clear that the diversity of living forms has increased. The reason such a diversity including inefficient replicators such as mammals continues to exist is precisely because natural evolution is *not* a free-for-all competition: There are many niches that support life, but only limited competition among them (e.g. grass does not compete with grasshoppers). In fact, filling one niche often leads implicitly to further niches that depend on the first. A proliferation of niches, as seen in natural evolution, implies a corresponding proliferation of varied lifestyles.

The conclusion is that perhaps natural evolution can be abstracted computationally as a process that finds many *novel* ways to express the *same* functionality. While in nature the search is guided only through the passive filter provided by the minimal criteria of survival and reproduction, in a computational abstraction the passive accumulation in nature can be *accelerated* by searching directly for novelty through the novelty search method. While in nature the minimal criterion is always reproduction, in a computational abstraction, the minimal criteria can be *anything*. In fact, it may be possible to choose minimal criteria in a practical domain such that the "epiphenomenal complexity" that results from novelty search subject to such criteria actually solves real-world problems.

While the choice of minimal criteria in a particular domain may require careful consideration because they add restrictions to the search space, such minimal criteria may provide a principled way to reduce the behavior space relative to novelty search alone, which may be sometimes infeasible. This perspective motivates studying the search for novelty subject to minimal criteria.

Extending novelty search to incorporate minimal criteria is straightforward. The evaluation of an individual in the domain need only additionally return whether the minimal criteria are met, thereby complementing the individual's behavioral characterization. If the minimal criteria are not met, then novelty search simply assigns a novelty score of zero to the individual and changes the failed individual's behavioral characterization to a dummy value that reflects its failure. For those individuals that meet the minimal criteria, novelty search operates as normal, assigning novelty scores to individuals based solely on relative behavioral novelty.

A practical concern when enforcing minimal criteria is that until an individual is found that meets the criteria, search is effectively random. That is, all individuals not meeting the minimal criteria are viewed equally as failures and only considered for reproduction if there are no individuals that do pass the criteria. Thus, if an individual meeting the minimal criteria is unlikely to appear in the initial population, it may be necessary to *seed* MCNS with a genome specifically evolved that does meet the criteria.

While enforcing minimal criteria involves only minor changes to the novelty search algorithm, the conceptual advance is significant. It is possible that natural evolution's ability to create an explosion of *interesting* complexity and diversity is an inevitable result of a search for many ways to do the same thing. That "something" is the minimal criteria.

The next section describes experiments in a maze navigation domain designed to test the potential of minimal criteria novelty search (MCNS).

8.2 MCNS Experiments

One practical motivation for enforcing minimal criteria during a search for novelty is to reduce the size of a vast behavior space so that finding the goal behavior can become more tractable. To verify the hypothesis that MCNS can be more efficient than either the raw search for novelty or the direct search for the objective in such large spaces, the maze navigation domain introduced in Chapter 4 is revisited with a maze containing vast uninteresting areas orthogonal to successful navigation.

Another motivation for MCNS is that, as a new abstraction of natural evolution, the search for many ways to do the same thing (upon which MCNS is based) can sometimes efficiently solve difficult problems by discovering complex policies. To explore this hypothesis, a more difficult extension of the maze navigation domain is considered in which a robot must navigate to two points instead of just the goal. This task is hard because after reaching one of the points, the robot must effectively adopt a different policy to reach the remaining point, which is non-trivial. If the minimal criterion is that an individual must always reach one of the points to be reproductively viable, MCNS will in effect search for many ways to reach that point, some of which may become more complex and eventually reach the other point as well even though it is not an explicit objective for MCNS.

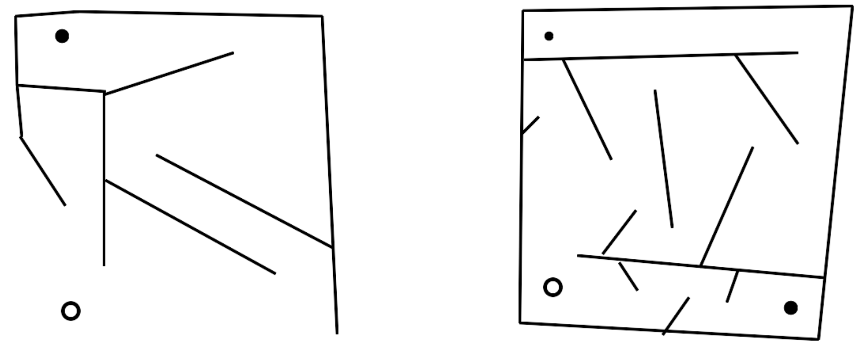
This chapter's experiments again utilize NEAT. In both experiments, NEAT with MCNS will be compared to NEAT with novelty search and traditional fitness-based NEAT. The next sections describes these experiments in detail.

8.2.1 Maze Navigation Experiment

The maze navigation domain from Chapter 4 is a good model domain because it is easy to understand and visualize. Recall that the main idea of the maze navigation domain is that a robot (figure 4.1b) controlled by an ANN (figure 4.1a) must navigate in a maze from a starting point to an end point in a fixed time.

Importantly, if the behavior of a robot is characterized as its final coordinate within the maze, large empty areas of the maze represent uninteresting areas of the behavior space that a raw search for novelty may waste evaluations exploring. Thus the basic idea of this experiment is to create a map in which there exists a large area orthogonal to reaching the goal that can be pruned from the behavior space through enforcing minimal criteria.

Therefore, as a preliminary test for MCNS, this experiment introduces a variation of the "hard maze" (figure 4.2b) that has a larger behavior space (figure 8.1a). The only difference from the original hard maze is that the new maze is no longer fully enclosed; a navigational robot may travel into the vast empty region outside the maze.



(a) Maze Navigation Map (b) Two-Point Map

Figure 8.1: Navigation Maps. In both maps, unfilled circles represent the starting position of the robot and filled circles represent goal points. The maze navigation map (a) is unenclosed, which enlarges the behavior space because the robot can travel outside the maze without restriction. The two-point navigation map (b) has two goal points that a successful navigational robot must reach; the first goal point is the larger filled circle, the second is the smaller filled circle.

Problematically, a raw search for novelty may expend unnecessary effort exploring this vast empty region of the map. Additionally, because the deceptive cul-de-sacs in the maze may confound an objective-based search (with the fitness function of minimizing distance to the goal), fitness-based NEAT may often fail to discover a solution as well. On the other hand, if a minimal criterion for reproduction is enforced such that an individual must end within the original bounds of the maze to be viable, the reduced behavior space may be tractable for MCNS, which may thus more consistently solve the maze than the other approaches.

Note that the same fitness function as in Chapter 4 is used for fitness-based NEAT in this experiment, which rewards a robot's nearness to the goal at the end of an evaluation. Likewise, the same behavior characterization is used for novelty search, i.e. a robot's behavior is defined as its location in the maze at the end of the evaluation.

Finally, NEAT with MCNS distinguishes between behaviors by using the same novelty metric as NEAT with novelty search, but additionally requires a set of minimal criteria to restrict the behavior space. Because behaviors that end outside the maze generally will have no relation to behaviors that navigate the maze, a simple criterion is that to be reproductively viable, an individual must end an evaluation within the maze. In this way, the behavior space is reduced to a subset that has a clear relation to successful maze navigating, although no behavior within this subset is a priori favored over another. It is important to note that because individuals that meet this particular minimal criterion are common in the initial population, it is unnecessary to seed evolution with a specifically-evolved genome.

The next section describes in detail the experiment in the two-point navigation domain.

8.2.2 Two-point Navigation Experiment

This second experiment elaborates on the maze navigation domain by introducing a more ambitious objective; instead of navigating to a single goal, the navigational robot must reach two separate goal points in the maze. This subtle change to the task relegates the objective behavior to a smaller portion of the behavior space, especially if both goal points are distant from each other and the starting location. It is important to note that which goal point the robot reaches first does not matter as long as both are reached.

To facilitate the robot's ability to navigate to both goal points, the network in figure 4.1 is augmented with an additional set of four pie-slice radar sensors that indicate the direction of the second goal point. Also, an input for both goal points is added that is continually activated after the corresponding goal is reached, enabling the robot to potentially change its policy and proceed to the unreached goal.

A new map is introduced (figure 8.1b) that has two goal points; the first point is in the lower-right corner of the map while the second is in the upper-left. The maze is deceptive because there are cul-de-sacs that lead close to both goal points; these cul-de-sacs create barriers in the fitness landscape that may stymie a traditional objective-based search. In contrast, the raw search for novelty must explore a vast behavior space, and thus may often fail to find a solution in a reasonable amount of time. For example, the raw search for novelty will likely discover separate behaviors, i.e. one that the leads to one goal point and one that leads to the other goal point, but may take unreasonably long to discover the highly-specific combined behavior in the vast sea of potential combinations. The main hypothesis is that only MCNS can search a tractable area of the behavior space to more consistently solve the problem.

Fitness-based NEAT, which will be compared to novelty search and MCNS, requires a fitness function to reward the navigational robots. Because the objective is to navigate to both goal points, the fitness f is awarded based on how close during an evaluation the robot approaches both goal points: $f = b_f - d_{g_1} + b_f - d_{g_2}$, where b_f is a constant bias, d_{g_1} is the closest the robot navigated to the first goal point, and d_{g_2} is the closest the robot navigated to the second goal point.

NEAT with novelty search, on the other hand, requires a novelty metric to distinguish between navigational robots. Because the trajectory of the robot is important in the two-point navigation domain, the behavior of the robot is defined as its location in the maze sampled uniformly a fixed number of times over the course of an evaluation (similar to experiments in chapter 4.3.2.1). The novelty metric is then the squared Euclidean distance between the position vectors of two individuals. Thus, an individual that reaches both points, although there is no direct bias to find such an individual, will appear novel.

Finally, NEAT with MCNS uses the same novelty metric as NEAT with novelty search, but additionally requires a set of minimal criteria to restrict the behavior space. If the minimal criterion is that the navigator must at least reach the first goal point, then the knowledge of how to traverse the maze and complete one of the two objectives is always maintained. MCNS will in effect search for novel ways to navigate that preserve the crucial functionality of reaching the first goal point.

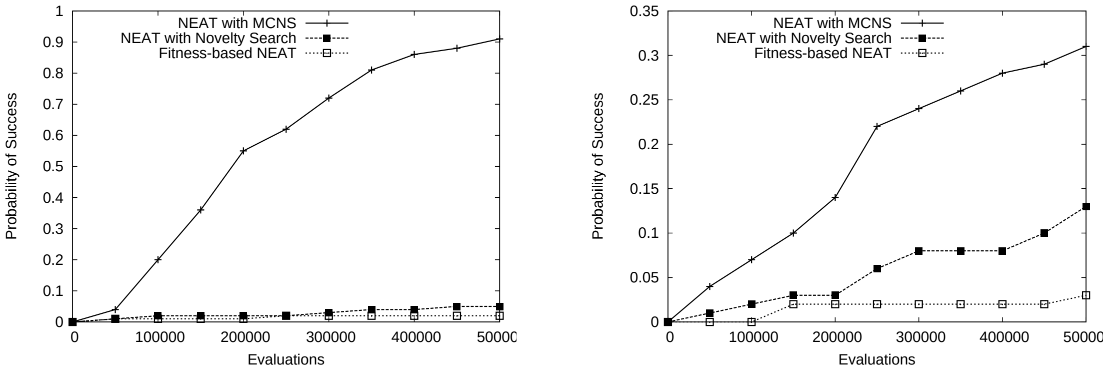
By enforcing this minimal criterion, the effective behavior space is reduced to a smaller subset with clear relation to the ultimate objective. The hypothesis is that MCNS can efficiently explore such a subset to discover a behavior that reaches both goal points, while a raw search for novelty may take longer to succeed because the full behavior space is significantly larger and contains mainly behaviors *completely orthogonal* to two-point navigation.

Note that this minimal criterion, unlike the criterion in the first experiment, requires seeding MCNS with an individual that can already reach the first goal point because such behaviors are unlikely to be found in the starting population. Thus, prior to running this experiment, for each of the 100 runs a seed genome was evolved that can reach the first goal point (seen in the lower right of figure 8.1b) by using the raw search for novelty and the same procedure from the first experiment. To make the comparison fair, all three of the compared methods are started with the seed genome able to reach the first goal point.

8.2.3 MCNS Parameters

Because NEAT with MCNS, NEAT with novelty search, and fitness-based NEAT differ only in whether they enforce minimal criteria or in substituting a novelty metric for a fitness function, they use the same parameters, which are listed in the Appendix. Runs consisted of 500,000 evaluations, which is equivalent to 1,000 generations of 500 individuals in a generational EA.

In the unenclosed hard maze experiment, as in the original formulation of the hard maze in chapter 4, the robot has 400 timesteps to navigate through the maze and the fitness bias f_b was 300. In the two-point navigation experiment, the robot has 1,200 timesteps to reach both points, and the fitness bias f_b was 500. The position of the robot is sampled three times during an evaluation to construct the behavioral characterization.



(a) Maze Navigation

(b) Two-point Navigation

Figure 8.2: Comparing MCNS, Novelty Search and Fitness-based Search. The change in probability of success over time (i.e. number of evaluations) is shown for NEAT with MCNS, NEAT with novelty search, and fitness-based NEAT on the maze navigation (a) and two-point navigation (b) tasks, both averaged over 100 runs of each approach. The main result is in both tasks MCNS evolves solutions with significantly higher probability after $500,000$ evaluations ($p < 0.01$; Fisher's exact test).

8.3 MCNS Results

The maze navigation task and the two-point navigation task were run 100 times with fitness-based NEAT, NEAT with novelty search, and NEAT with MCNS. Figure 8.2 highlights that in both tasks NEAT with MCNS evolves solutions with significantly higher probability than the other two approaches after 500,000 evaluations ($p < 0.01$; Fisher's exact test).

8.4 Implications

The results support the hypothesis that reducing the behavior space through enforcing minimal criteria can sometimes increase the efficiency of novelty search. They also further highlight the vulnerability of traditional fitness-based evolution. Thus, MCNS is a way to focus the search for novelty on an interesting subset of behaviors while still remaining agnostic to which of the behaviors in the viable subset will ultimately lead to the objective.

Furthermore, the two-point navigation experiment demonstrates that the search for many ways to do the same thing, an abstraction of how natural evolution accumulates novelty and complexity, can discover policies that exhibit complexity orthogonal to meeting the minimal criteria but essential to solving the problem. Thus, MCNS connects a coarse abstraction of natural evolution to solving real world problems.

While the empirical results show that MCNS can sometimes solve problems more consistently than other methods, the claim is not that MCNS will always work well; if the minimal criteria is too stringent or if evolution is seeded with an individual that is fragile and cannot be mutated without violating the minimal criteria, then most evaluations may be wasted and search will be inefficient. The minimal criteria and the seed genome must be carefully considered just like the fitness function in standard EC models.

Although MCNS is a practical search technique for solving difficult EC problems, it also demonstrates that EC models need not always be derived from the prevalent fitness-pressure-based abstraction of natural evolution. Because the drive to complexity in natural evolution may in fact be opposed by selection pressure, further pursuing the pervasive objective-based paradigm may yield little progress. Instead, alternate abstractions such as MCNS that more accurately abstract the high-level process by which natural evolution discovers complexity may be a more appropriate basis for solving highly ambitious problems.

Both the raw search for novelty and MCNS exhibit an open-ended search process that in principle could continue indefinitely to discover novel policies of increasing complexity, which makes them relevant to artificial life research as well. Yet the maze navigation and two-point navigation domains offer limited potential for behaviors that a human observer would find *interesting*. This limitation is in stark contrast to natural evolution, which exhibits a vast diversity of interesting forms and behaviors. An open question is how to further align the measure of novelty with such interestingness.

In summary, paradoxically, MCNS demonstrates that the best way to learn to do something new may often be to try to do the same thing again.

CHAPTER 9
NOVELTY SEARCH WITH LOCAL COMPETITION

Another extension of novelty search is introduced in this chapter called *novelty search with local competition* (NSLC). The main idea is that nature's creativity is facilitated by limited competition between niches. Inspired by this insight, NSLC augments the raw search for novelty with an additional drive for competition that is restricted locally among a space of niches. The effectiveness of NSLC is demonstrated in this chapter in the culminating application for this dissertation of evolving a diversity of locomoting virtual creatures.

Thus the next section reviews previous attempts to evolve virtual creatures including the Evolving Robotic Organisms (ERO; [96]) system that is used in the experiment in this chapter.

9.1 Evolving Virtual Creatures

The most well-known example of evolving virtual creatures is the seminal work of Sims [146]. In this work, an EA evolves the morphology and control policy of virtual creatures specified by an encoding able to represent hierarchical, symmetric, and repeating structures. Creatures are simulated in a realistic three-dimensional environment and learn to locomote in different ways. The results have since been replicated [96, 114]. A problem with Sims' system is that a particular run *converges* to a single morphology [146], contrasting with nature's accumulation of diverse organisms.

Of the systems that have been designed since, most focus on alternate ways of representing organisms, e.g. L-systems [74] or genetic regulatory networks [15]. A recent more granular approach to representation is applying compositional pattern producing networks (CPPNs [159]) to specify growth patterns of three-dimensional structures [6] or density patterns in three-dimensional space for soft robotics [68]. Such systems can in theory scale indefinitely to represent creatures with a large quantity of morphological parts. Though equipped with more advanced representations, these systems also tend to converge to single morphologies [74, 15]. That is, convergence results more from selection than representation.

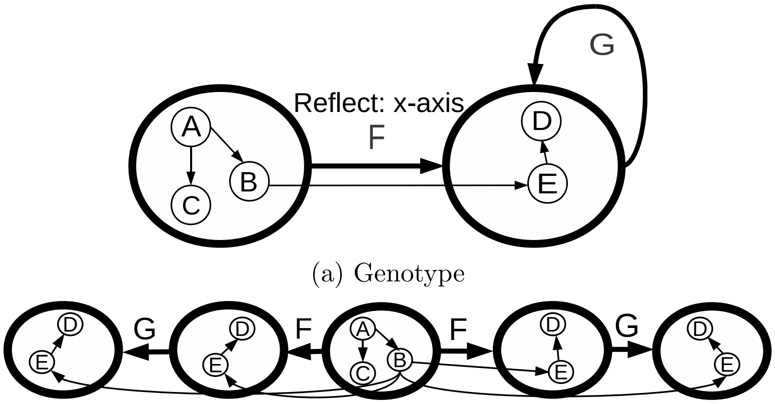
The particular system extended for the experiments in this chapter is called ERO [96], a modern re-implementation of Sims [146] with extensions. Although newer encodings may provide greater evolvability or potential, ERO is extensively tested and therefore provides a good experimental platform for the local competition approach, which can in principle be generally applied to any creature representation to mitigate morphological convergence. Thus the next section briefly reviews ERO (for a full description see Krcah [96]).

9.1.1 Evolution of Robotic Organisms (ERO)

Krcah [96] describes the ERO system, applies ERO to recreate the experiments of Sims [146], and tests extensions designed to encourage diversity and enhance crossover. ERO applies an EA to a complex encoding derived from that in Sims [146] with extensions inspired by NEAT [155]; the encoding specifies both the morphology and control policy of a virtual creature. Fitness evaluation takes place in a realistic three-dimensional simulation where creatures are rewarded for their ability to effectively locomote.

ERO's encoding unrolls an evolved genotypic graph structure into a coupled body plan and control policy; this unrolling process yields hierarchical and symmetric repeating structures. In ERO, an organism's genome is represented as a nested graph structure that has outer nodes and inner nodes (figure 9.1). The outer nodes and the connections among them represent different morphological parts comprising a creature's body, while the inner nodes and the connections among them represent the artificial neural network (ANN) that will control the creature.

Each outer node represents a body part of a creature, and each directed edge between two outer nodes represents a physical joint. Connections can be recursively applied to generate hierarchical structures, and can also be mirrored over different axes to generate symmetrical structures. The idea is to provide evolution with a set of useful regularities similar to those seen in natural evolution, which also exploits symmetry and hierarchical structures.



(a) Genotype

(b) Phenotype Blueprint

Figure 9.1: ERO Encoding. A nested graph genotype is shown in (a) that consists of outer nodes (i.e. the larger circles) that represent morphological parts, connections among outer nodes (e.g. F and G) that represent joints, inner nodes (i.e. the smaller circles) that represent neurons in an ANN, and neural connections among inner nodes (e.g. the connection between B and E). The phenotype shown in (b) is unrolled from the genotype (a) according to flags associated with the connections between outer nodes that can encode symmetry (e.g. the joint F and its children are reflected over the x-axis) and repetition (e.g. joints can be repeated to create articulated appendages like fingers), yielding a hierarchical repeating structure.

ERO also encodes the control policy of a creature in the nested graph. Each inner node represents a node in an ANN, and each weighted directed edge between two inner nodes is a connection in the ANN. As the genotypic graph is unrolled into the creature, the ANN represented by the inner nodes and connections is unrolled as well, generating modular ANNs that repeat in the same way as the morphological nodes they are nested within. ANN nodes can represent sensory organs of the creature, motors driving the joints of the creature, or intermediate functions of sensory signals.

By applying an EA to this representation, the experiments by Krcah [96] evolve creatures that locomote in various ways (walking, swimming, and jumping) and that can reactively follow a light. Even though ERO extends Sims [146] with genotypic speciation (possible in such a graph encoding by exploiting historical markings as in Stanley and Miikkulainen [155]), which does increase performance, preliminary experimentation revealed that such diversity maintenance is typically ineffective at preventing convergence to a single morphology by the end of an evolutionary run, like other related systems [146, 15, 74, 114].

Because it derives from a well-tested and validated model of virtual creatures, ERO is an ideal system to test the hypothesis advanced in this paper: Morphologically local competition combined with a drive for novelty will more effectively explore and exploit a diversity of morphological niches than more traditional approaches.

The next section motivates and describes the NSLC approach in detail.

9.2 The NSLC Algorithm

One reason virtual creature experiments are interesting is the morphological and functional variety that evolution might discover in such experiments. However, in most existing systems, uncovering such diversity becomes almost prohibitively computationally expensive; though *different* runs may yield varied solutions, a particular run will usually converge to a single morphology [74, 15, 146, 114]. This dynamic also departs strikingly from natural evolution, which concurrently explores an increasing diversity of morphologies. Thus an important result more in line with the creativity of natural evolution would be to evolve a wide diversity of functional morphologies within a single run.

To remedy the problem of morphological convergence, the idea in this chapter is to apply novelty search to reward *novel morphologies* by defining a space of morphological niches and a novelty metric that quantifies how different a new morphology is from previous morphologies in this space. Note that this approach differs from prior novelty search experiments in which novelty search rewards novel *behaviors*; however the underlying mechanism is exactly the same because both are simply abstract spaces of phenotypic properties of evolved individuals. For example, a two-dimensional morphology space can be constructed by considering the height and mass of a virtual creature. A search for novelty within this space will effectively explore the space of morphologies spanning those that are short and light to tall and heavy.

However, a raw search for morphological novelty does not guarantee that any of the discovered morphologies are *functional*. That is, rewarding pure morphological novelty does not address the control policy of the virtual creature; all of the creatures discovered might not move at all. Such disconnect between morphology and functionality differs from prior experiments with novelty search in which the space it explores also distinguishes *behaviors* of interest [102, 136]. Thus what is needed is an extension to novelty search that encourages functionality when exploring dimensions like morphology that themselves do not distinguish functionality.

The next section describes one approach to balancing competing drives for diversity and functionality through a Pareto-based Multi-objective Evolutionary Algorithm (MOEA).

9.2.1 Balancing Achievement with Novelty

Multi-objective optimization is a popular paradigm within EC that addresses how to optimize more than one objective at the same time in a principled way [27]. Such multi-objective search suggests a simple way to combine the drive to optimize performance with the search for novelty: Reward both performance and novelty at the same time by making them separate objectives in a multi-objective search [117]. Yet while it is tempting simply to combine these different objectives and novelty through an MOEA, it is important to understand what this approach might inadvertently imply.

The concept of dominance is central to Pareto-based multi-objective search; the key insight is that when comparing the performance of two individuals over multiple objectives, if both individuals are better on different subsets of the objectives then there is no meaningful way to directly rank such individuals because neither entirely *dominates* the other. That is, ranking such mutually non-dominating individuals would require placing priority or weight on one objective at the cost of another; traditionally one individual dominates another only if it is no worse than the other over all objectives and better than the other individual on at least one objective.

Thus the best individuals in a population are those that are not dominated by any others. Such best individuals form the *non-dominated front*, which defines a series of trade-offs in the objective space. That is, the non-dominated front contains individuals that specialize in various combinations of optimizing the set of all objectives; some will maximize one at the expense of all the rest, while some may focus equally on all of the objectives.

Therefore if novelty and fitness are simply two separate objectives, then at one extreme of the resulting non-dominated front, maximizing novelty at the expense of performance will be rewarded. At the other extreme of the front, maximizing performance at the expense of novelty will be rewarded. In between these two extremes will be various trade-offs between performance and novelty. In effect, this type of search is a mixture of traditional objective-based search and novelty search; this simple combination can be seen as an objective-based search that encourages novelty to maintain behavioral diversity.

While this option is viable for search and may sometimes be more effective at achieving higher performance than searching only for novelty or only for functionality, it fails to exploit the fact that some niches may naturally support different levels of fitness than others. For example, bacteria reproduce dramatically faster than humans, and thus technically have higher fitness, though both are viable ways of life in nature. Sometimes an interesting goal is not narrowly to achieve the *highest* fitness, but rather to discover a wide diversity of individuals well-adapted to *their own niches*, like the diversity seen in natural evolution.

In nature, the success of one particular niche does not affect the success of another unrelated niche; that is, bears do not generally suffer from the reproductive success of bacteria. In contrast, in EC a global trade-off between fitness and novelty implies that the success of one niche may render other niches unappealing. For example, in the context of a virtual world, a medium-size, medium-mass organism may be more optimal for traveling quickly than a tall, low-mass organism; a global trade-off between fitness (i.e. locomotion speed) and novelty means that most resources will be spent on the medium morphology even though the way that the tall morphology would eventually perform at its best relative to nearby niches may ultimately be just as interesting. Thus a Pareto front trading off fitness and novelty does not really capture the spirit of the diverse set of locally-optimal designs we might hope to emerge in an artificial world.

9.2.2 Local Competition

The suggested response is to limit competition among organisms *locally* within a niche-space (i.e. not *physical* locality); that is, individuals compete only with other individuals nearby in niche-space. For example, in an artificial world a space of niches can be constructed by considering the space of possible morphologies. Local competition within such a space implies that individuals compete for fitness only with other individuals in similar morphological niches.

The key change is that local competition within a niche space transforms the fitness objective from being a global measure to one relative to an individual's neighbors in niche space. In effect, the search pressure changes from managing a global trade-off between fitness and novelty that is biased towards particular niches to encouraging both novelty and better performance relative to an individual's niche.

The idea is thus to explore the merits of *each niche* rather than to exploit greedily only the best niches. This strategy reflects that what is most interesting about some domains may not always be discovering the most optimal behavior. That is, among the various strategies for achieving the highest performance score in a given domain, the factors that decide which is ultimately optimal may only be an incidental side effect of how the domain is constructed. For example, one particular artificial world may have a physics engine in which quadruped locomotion is optimal, and another may have an engine in which worm-like locomotion is optimal. Thus the particular optimality of one strategy or another may be uninteresting; instead, it may be more interesting to observe the creative force of evolution exploiting a wide variety of niches than it is to see only the optimal product of evolution narrowly converging.

It is important to note the differences between local competition and existing diversity maintenance techniques in EC [76, 73, 54]. Some such techniques do restrict competition, e.g. by level of fitness [76] or by age of genotype [73], but not in a way that necessarily encourages finding and maintaining *functionally* distinct solutions, i.e. genotypic age and fitness level have no inherent link to an individual's *means* of functionality (e.g. its particular strategy or morphology). Other methods modulate fitness based on genotypic diversity [54] but generally do not change the *scale* of fitness itself; though rare genotypes may have a better chance of surviving, such methods do not acknowledge that different niches may have widely differing distributions of fitness values that they are capable of supporting (e.g. bacterial niches support absolute fitness values of several orders of magnitude higher than mammalian niches).

In practice, transforming a global competition score (e.g. the fitness function) into a local competition score requires a comparison of an individual's performance to that of its nearest neighbors in niche space. The more neighbors it outperforms, the higher its local competition score.

Therefore, it is straightforward to implement local competition as an extension of novelty search, which already calculates nearest neighbors in morphology space. When determining the novelty of an individual by measuring the average distance to its nearest neighbors in morphology space, the number of such nearest neighbors with lower fitness than that of the individual is also counted. This number is assigned as the *local competition objective* for that individual, which measures an individual's performance *relative* to its niche. Thus two objectives in the MOEA become novelty and local competitiveness. The expectation is that balancing achievement and novelty in this way will lead to more natural evolutionary dynamics through a gradual accumulation of functionally-diverse well-adapted individuals.

9.3 Virtual Creatures Experiment

The experiment in this chapter extends ERO with Pareto multi-objective search, novelty search, and local competition. It is important to note that although this composite system has several components, the central underlying intuition motivating the system is simple: Encouraging both morphological novelty and local competition may lead to a greater diversity of functional creatures.

The MOEA with which ERO is extended is based on NSGA-II, a well-known MOEA [40]. One difference is that while NSGA-II has a mechanism to encourage diversity along the non-dominated front, the implementation described in this chapter *replaces* that mechanism with a separate objective explicitly rewarding *genotypic* diversity. This departure is logical because both novelty and local competition are relative measures; two individuals with exactly the same novelty or local competition scores that would be conflated by encouraging diversity on the Pareto front may actually be quite different morphologically or with respect to performance.

The characterization of morphology space has important consequences for evolution and strongly biases the types of creatures evolution will discover. In this paper the morphological space that novelty search explores and within which competition is localized has three dimensions: height, mass, and number of active joints. This space encourages creatures of different sizes, with different distributions of mass, and that exploit different means of locomotion.

In all experimental setups, the fitness function rewards virtual creatures according to how far they are able to travel within an evaluation. The novelty metric is the squared Euclidean distance separating two individuals in morphology space, and thus the novelty of a creature is proportional to how different its morphology is from that of other creatures currently in the population.

The genotypic diversity objective encourages elaborating upon existing genotypes by assigning higher scores to more novel genotypes. A rough characterization of a genotype in ERO is the number of outer genotypic nodes it has; a genome starts with a small number of outer genotypic nodes, more of which can be added through mutation as evolution progresses, and the amount of which limit the maximal morphological complexity a genotype can represent. Because adding a new outer genotypic node often initially disrupts existing functionality, it is beneficial if such innovation is temporarily protected. Thus genotypic diversity is rewarded inversely proportionally to how many other genotypes in the population have the same amount of outer genotypic nodes.

Four experimental setups are tested: (1) **fitness alone**, (2) **novelty alone**, (3) **novelty search with global fitness competition**, and (4) **novelty search with local fitness competition**. The only difference between these setups is the choice of objectives given to the MOEA. Genotypic diversity is an objective in all variants.

The first setup, fitness alone, has a global fitness objective and a genotypic diversity objective. This setup is a control, roughly corresponding to the original ERO experiment, which employed fitness-based search coupled with speciation to encourage genotypic diversity.

The second setup, novelty alone, has a novelty objective and a genotypic diversity objective. This setup is a second control designed to show that without additional pressure to adapt, a search for morphological novelty alone will not generate creatures able to effectively locomote.

The third setup, novelty with global competition, has a novelty objective, a global fitness objective, and a genotypic diversity objective. This setup is much like the combination of novelty and fitness in Mouret [117]. The idea is to extend standard objective-based search with an additional drive for novelty that may encourage diversity, thereby reaping some of the benefit of novelty search while mostly maintaining the familiar objective-based search paradigm.

The final setup, novelty with local competition, abstracts competition between niches in nature; instead of a global fitness objective between all morphologies, evolution is provided a local competition objective that restricts competition within the morphology space. That is, an individual receives a local competition score relative to how many of its nearest neighbors in morphological space it outperforms.

Additionally, in the final setup the genotypic diversity objective is also localized within the morphology space; similar in motivation to that of local competition, local diversity ensures that genotypic diversity is not only exploited in those morphological niches in which such diversity is incidentally most easily expressed.

In all setups, the distribution of individuals in morphology space is recorded, as well as the final generation of the individuals themselves and the progression of highest fitness individuals throughout the run. The idea is to quantify how well each morphological niche is being exploited and how much morphological diversity is discovered and *maintained* throughout a run.

9.3.1 Virtual Creature Parameters

The population size for all experiments is $1,000$, and a run consists of 500 generations. The nearest-neighbor size for novelty search and local competition is 15. The three morphology dimensions are rescaled so that their values fill the range between zero and three; the height dimension originally spans between zero and 1.5 meters, the mass dimension originally spans between zero and 3.0 kilograms, and the active joint dimension originally spans between one and four active joints. The crossover rate was 75% and mutated parameters would be perturbed with power 0.25. The selection method was changed to tournament selection (with tournament size two) to agree with NSGA-II [40]. Other parameters are the same as in Krcah [96].

9.4 Virtual Creatures Results

Three metrics are described next that respectively quantify the ability of each setup to discover high fitness individuals, to maintain morphological diversity, and to exploit morphological niches. For each of the metrics, the Kruskal-Wallis test was first applied across the four experimental setups to demonstrate that the distributions are significantly different ($p < 0.001$). The remainder of the statistical tests below are Mann-Whitney U tests measuring the significance of pair-wise differences between experimental setups.

The first metric is the maximum fitness discovered by a particular setup. These results are shown in figure 9.2. The worst-performing setup for this metric is morphological novelty alone because it enforces no selection pressure towards functionality (of course, another kind of novelty, such as behavioral, might have produced better fitness, as previous results in this dissertation have shown, but would not produce the desired morphological diversity). The global competition setup, which heavily favors morphological niches that most easily facilitate locomotion, significantly outperforms the other setups ($p < 0.001$). Fitness alone and local competition do not perform significantly differently from each other, but perform better than novelty alone ($p < 0.001$).

The second metric is niche coverage, which quantifies how well a setup has encouraged and *maintained* morphological diversity until the end of a run. The morphology space is overlaid with a regularly-spaced grid. From each point in the grid, the distance in morphology space to the nearest individual in the final population is calculated. The more well-covered the niche space is, the less the sum of all such nearest distances will be; if the population has spread throughout niche space, then on average there will be an individual near to each point in the grid. The results are shown in figure 9.3. Note that the larger the niche coverage metric is, the *worse* the morphological space is covered. Thus the worst-performing setup is fitness alone, which has no direct selection pressure towards discovering morphological diversity beyond that useful in maximizing fitness ($p < 0.001$). The best performing setups are novelty alone and local competition ($p < 0.001$), which do not perform significantly differently from each other. Local competition outperforms global competition on this metric because it allocates resources more fairly among morphological niches.

The last metric is niche exploitation, which quantifies how well on average a setup has accumulated and maintained diverse individuals that exploit niches. The morphological space is first divided into equal-sized bins. Next, each individual in the final population is placed into the bin that matches its morphology. Then, for each bin the best fitness among all individuals in it is recorded; these represent the best solutions for each niche that evolution was able to find. If all niches supported the same level of fitness, then taking the average of these best fitnesses would be a reasonable measure of niche exploitation. However, figure 9.4, which illustrates niche capacities, demonstrates the uneven distribution of solutions discovered across morphology space. Intuitively, some morphological niches are simply better suited to locomotion than others; e.g. it is difficult to construct an effective, very tall, very light creature.

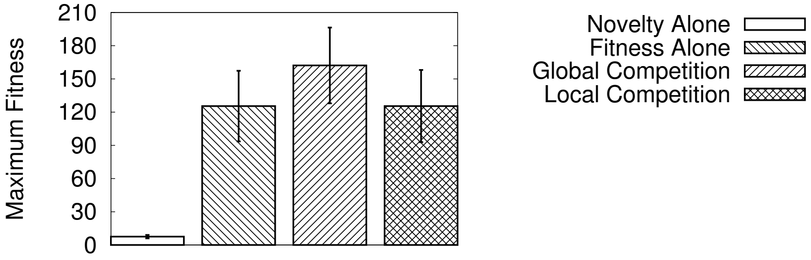


Figure 9.2: Absolute Performance Comparison (larger is better). For each setup, the maximum fitness discovered in a particular run is shown (averaged over 15 runs). The main result is that novelty search with global competition discovers the most fit individuals ($p < 0.001$).

Because of the uneven distribution of maximum fitnesses across niches, which span several orders of magnitude, fitness values for each niche are normalized first logarithmically, then are divided by the *best* fitness values for each niche (also normalized logarithmically) over *all* runs; i.e. each niche is normalized by its capacity for fitness. Finally, these normalized fitness values are then averaged over all of the niches, yielding the average niche exploitation; this number indicates for a particular run the average percentage of a niche's capacity evolution was able to exploit.

The results are shown in figure 9.5. This figure illustrates the trade-off between localized and globalized competition: While global competition is better able to find the maximum fitness, local competition is better able to exploit all of the morphological niches.

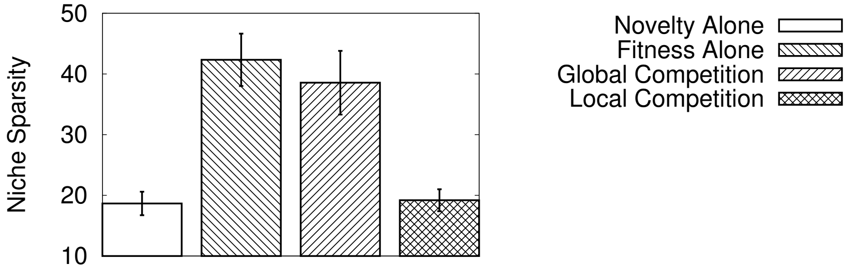


Figure 9.3: Niche Sparsity (larger is worse). For each setup, the average coverage of morphology space of the final population of a particular run is shown (lower is better; averaged over 15 runs). The main result is that novelty with local competition and novelty search alone cover the niche space the best ($p < 0.001$).

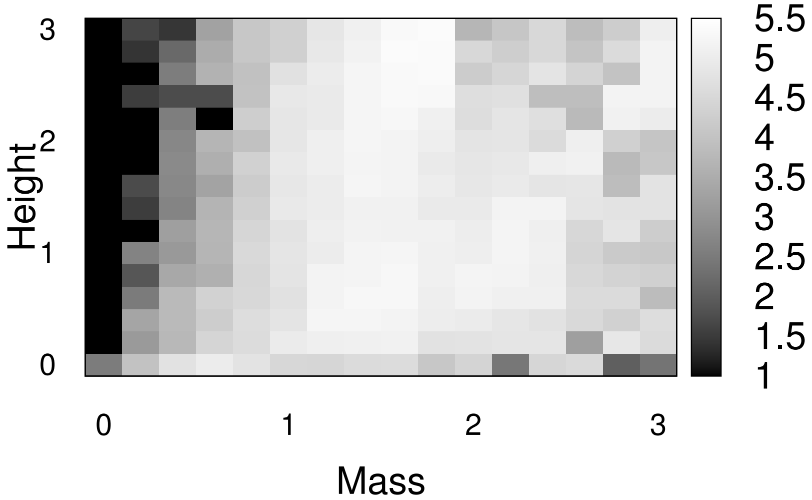


Figure 9.4: Niche Capacity. The capacity of evolution to exploit different morphological niches is illustrated above. Each square represents a segment of morphology space (only two out of three morphological dimensions are visualized), and its darkness is proportional to the logarithm of the highest fitness value found within that segment of morphology space (i.e. darker means more fit), over all runs of all variants; thus it is an estimate of the upper bound of fitness that the niche supports.

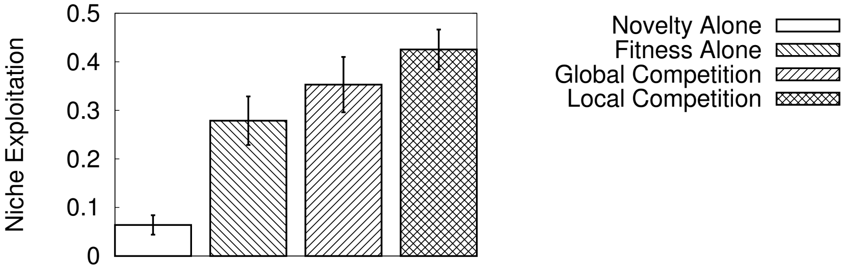


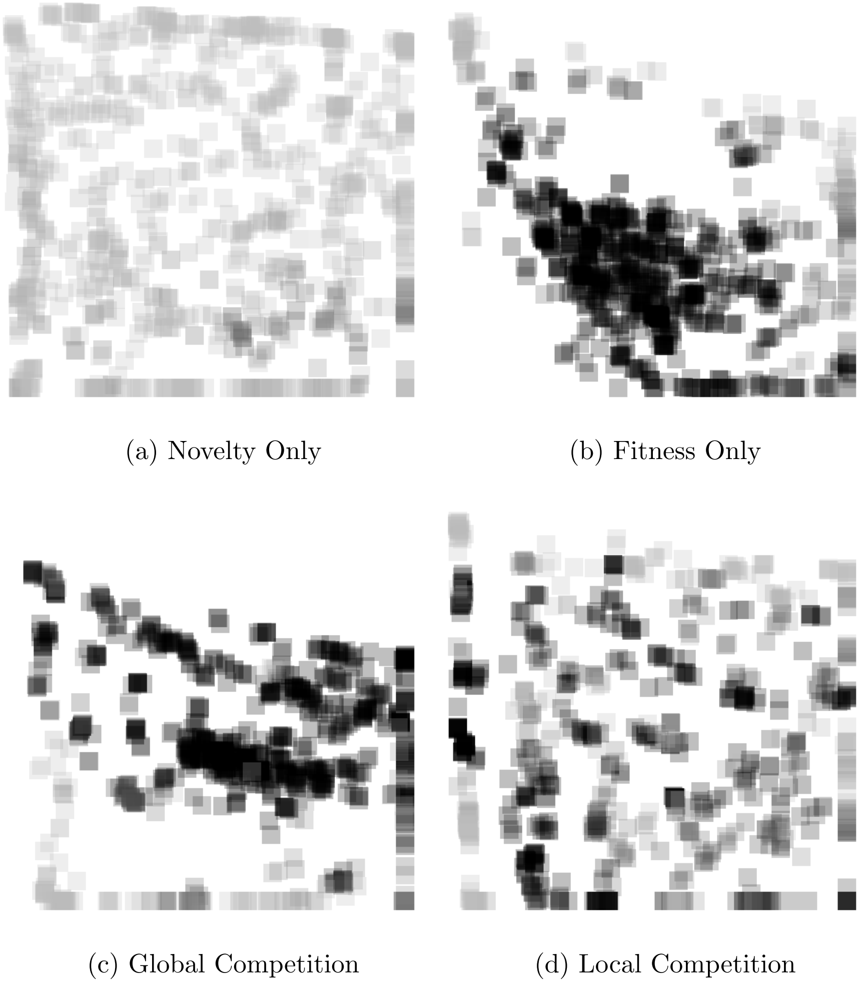
Figure 9.5: Niche Exploitation (larger is better). For each setup, the average niche exploitation is shown (averaged over 15 runs). The main result is that novelty with local competition exploits niches significantly better on average than the other setups ($p < 0.001$).

9.5 Typical Run

Typical runs of novelty alone accumulate diverse morphologies, but none are functional (figure 9.6a). In contrast, typical runs of fitness alone (figure 9.6b) and novelty with global competition (figure 9.6c) generally converge to a single body plan. However, global competition *stretches* that body plan further through morphology space than fitness alone because of novelty pressure. Finally, local competition (figure 9.6d) typically finds diverse functional morphologies, often maintaining worm-like, unipedal, and quadrupedal locomotion throughout a run, all at once; such diversity would otherwise require many independent runs [96, 146, 15, 74, 114].

Images of typical such creatures discovered within a single run are shown in figure 9.7, and videos of typical ones discovered within a single run are available at:

`http://eplex.cs.ucf.edu/ns/vc.html`



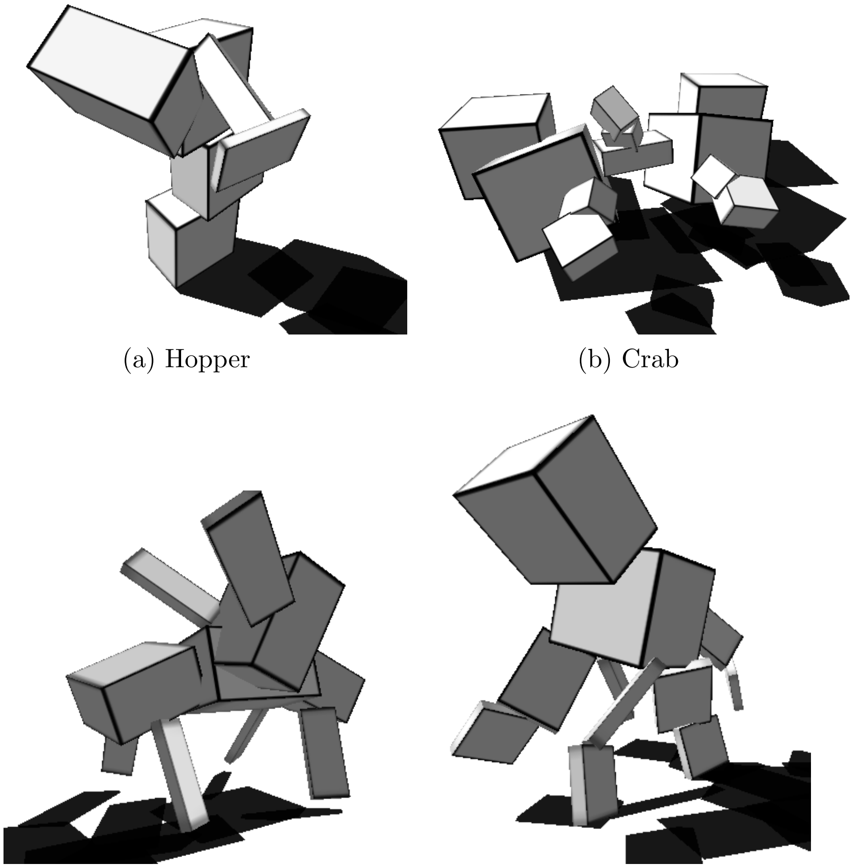
(a) Novelty Only

(b) Fitness Only

(c) Global Competition

(d) Local Competition

Figure 9.6: Comparing exploration and exploitation of morphological niches. The final distribution of virtual creatures from a single run in two morphological dimensions (height and mass) is shown for typical runs of (a) novelty only, (b) fitness only, (c) novelty search with global competition, and (d) novelty search with local competition. The darkness of each point indicates how fit each virtual creature is. The main result is that local competition effectively exploits more of the morphological niche space than the other methods.



(a) Hopper

(b) Crab

(c) Quadruped

(d) Tailed Quadruped

Figure 9.7: Diverse competent morphologies discovered within a typical *single* run of local competition. Various creatures are shown that have specialized to effectively exploit particular niches of morphology space. These creatures were all found in the final population of a typical run of local competition. The hopper (a) is a unipedal hopper that is very tall, (b) is a heavy short crab-like creature, and (c) and (d) are distinct quadrupeds. Creature (c) drives a large protrusion on its back to generate momentum, and (d) has a tail for balance.

9.6 Implications

The results support the hypothesis in this chapter that local competition combined with novelty search can better maintain and exploit a diversity of individuals than can global competition between all individuals combined with novelty, albeit at a cost in absolute performance. This result is potentially important given the ubiquity of global competition objectives throughout EC and the difficulty for artificial life systems to maintain a wide diversity of functional organisms.

While this chapter combines drives towards novelty and achievement, prior successes with searching for novelty alone (e.g. Chapters 4 and 6) illustrate that such a combination is not always necessary. The poor relative performance of searching for novel morphologies alone in this chapter is not surprising nor does it contradict such prior successes. That is, of course morphological novelty does not distinguish competent creatures as another type of novelty might (e.g. behavioral novelty). Rather, the idea in this chapter is that sometimes it is most natural to express and encourage the desired outcome of evolution (e.g. a morphologically diverse collection of competent creatures) as a combination of loosely connected drives.

In the context of artificial life, it is interesting to further consider the relationship between local competition and niches in nature. Morphologically localized competition is an abstraction of interacting niches in nature. However, the abstraction is inexact: In nature, niches are not aligned in fixed-dimensional spaces, and interactions between niches, although *restricted*, may not always be *localized*. For example, a large predator may impinge on the fitness of small prey, or a small parasite may make a larger animal ill.

That is, natural niches are not fundamentally functions of morphology but of ways of life, which are defined implicitly and tend to grow through a process of bootstrapping. One way of life may provide the substrate for many others. For example, the stomach of an animal may be a novel niche, and the same animal's waste products can found yet another niche. An interesting future research direction, though it is unclear how to accomplish it, is to construct a similar system wherein niches and interactions are implicitly defined and thereby lead to indefinite open-ended discovery of new niches.

It is important to note that lack of exact fidelity to nature does not imply that local competition may not often be a useful tool in encouraging functional diversity whenever a niche space can defined. In fact, this formulation may in general be a good way of merging novelty search with the drive to optimize an objective. It may be particularly important in the virtual worlds of artificial life wherein the hope is to encourage an explosion of diverse and competent (though not necessarily globally optimal) creatures.

The idea of local competition maintaining a diversity of interesting functional individuals aligns well with the dynamics of natural evolution. Thus the culminating application of this dissertation, successfully evolving a wide diversity of functional creatures within a single run, may bring the dynamics of EAs one step closer to that of natural evolution. Additionally, the results presented here may provide anecdotal evidence of a problem in how evolutionary algorithms are nearly always applied: Evolution may ultimately prove better suited to progressively uncovering a widening interconnected web of interesting, complex artifacts than it is to the constrained pursuit of a single optimal individual.

CHAPTER 10
DISCUSSION

A new evolutionary search paradigm was introduced in this dissertation that is unencumbered by the long-standing assumption that measuring and rewarding progress with respect to an objective is always the best approach. By ignoring the objective and instead rewarding only behaviorial novelty, novelty search is immune to the problem of deception that often prevents ambitious objectives from being reached. This chapter begins with a discussion of the importance of abandoning objectives, and then explains what novelty search reveals about open-ended evolution. The following section then contemplates the two extensions of novelty search: minimal criteria novelty search and novelty search with local competition. The final section covers future possibilities created by the results in this dissertation.

10.1 Abandoning Objectives

Novelty search casts the performance of evolutionary algorithms in a new perspective. For example, based on the performance of fitness-based search in the many experiments in this dissertation, the usual conclusion would be that the underlying search algorithms are ineffective for solving these problems. Yet the results from the same experiments with novelty search, which changes the reward function while preserving the rest of the algorithm, shows that the pathology is *not* in the search algorithms but rather in the pursuit of the objective itself. Higher performance is achieved when the only incentive given to the search algorithm is to produce individuals that are different functionally from those seen before. It is also notable that the problem is not simply diversity maintenance: NEAT itself, which is used in many of the experiments in this dissertation, already employs the diversity maintenance technique called explicit fitness sharing [54], yet it is still fundamentally deceived when seeking higher fitness.

Furthermore, this dissertation has shown that searching without objectives may have implications beyond just performance. For example, fitness may not always be a good heuristic for the evolvability of an individual's representation or of the appropriateness of self-adapted reproduction parameters. As demonstrated in Chapter 7, novelty search may favor greater evolvability and more readily exploit the benefits of self-adaptation than the traditional search for the objective. An important conclusion is that traditional comparisons within EC over properties such as performance, evolvability, or the potential of self-adaptation may themselves be deceptive because all of these properties also depend on the reward scheme. Thus novelty search calls into question past such comparisons that inferred the ineffectiveness of a particular algorithm or feature based on its ability to optimize fitness.

However, novelty search also faces potential limitations. For example, because it ignores the objective, there is no bias towards optimization once a solution is found. An optimized solution may be produced by novelty search only if an individual can appear novel by reaching such performance. However, it may be more efficient to take the most promising results from novelty search and further optimize them based on an objective function. This idea exploits the strengths of both approaches: Novelty search effectively finds approximate solutions, while objective optimization is good for *tuning* approximate solutions. One potential such approach is to make novelty an objective and fitness another objective in a multi-objective formulation of the problem as in Mouret [117]. Another such approach more in line with the motivations of artificial life is given by the NSLC extension of novelty search introduced in Chapter 9.

Although counterintuitive, the idea that the search for novelty can outperform the search for the objective introduces critical insight: Objective fitness by necessity instantiates an imposing landscape of peaks and valleys. For complex problems it may be impossible to define an objective function through which these peaks and valleys create a direct route through the search space. Yet in novelty search, the rugged landscape evaporates into an intricate web of paths leading from one idea to another; the concepts of higher and lower ground are replaced by an agnostic landscape that points only along the gradient of novelty. What were once barriers become smooth links among nodes in in a large lattice.

The problem with the objective is that it fails to identify the stepping stones. The more ambitious and complex the problem, the more difficult it is to formalize an objective that rewards the stepping stones along the way. Yet it is exactly those stepping stones that ultimately must be identified and rewarded if search is to find its way up the ladder of complexity [113]. Novelty search is designed to build gradients that lead to stepping stones. By abandoning the objective, all the steps along the way come into greater focus. While the trade-off is a more expansive search, it is better to search far and wide and eventually reach a summit than to search narrowly and single-mindedly yet never come close.

Of course, there are likely domains for which the representation is not suited to discovering the needed behavior or in which the space of behaviors is too vast for novelty search to reliably discover the desired ones. For example, the task of pattern classification lacks the inherent *domain restrictions* of control tasks (e.g. simulated physics restricts the possible behaviors of the biped and walls restrict the movement of the maze navigator) and thus may not be directly amenable to a search for novelty without introducing some artificial constraints. Additionally, there are some domains in which it may be difficult to define behavior or the novelty metric; interestingly, Gomez [57] defines a universal behavioral distance metric based on algorithmic information theory, and research in adaptive curiosity in developmental robotics may also provide insight into defining appropriate characterizations of behavior [121, 122, 85]. Characterizing when and for what reason novelty search fails is an important future research direction. Yet its performance has proved robust since it was first introduced in a conference paper in 2008 [136, 91, 55, 117, 120, 102, 97, 43].

However, because the results in this dissertation challenge common intuitions, it is important to interpret them carefully. It would be a mistake to read them as a typical comparison between competing methodologies. In fact, the message is not that novelty search is "better" than objective-based search. Rather, a deeper interpretation is that search is about much more than objectives alone and ultimately can be guided by a diverse range of available *information*, much of it orthogonal to any explicit objective. Yet the field of EC, and even machine learning, has historically focused almost exclusively on this single guiding principle (i.e. objectives) among the many other kinds of searches that are possible, and hence the field is beginning to encounter the limits of what the objective-based paradigm has to offer. The results in this dissertation serve to confirm that indeed the objective is not the only impetus for search that can lead to interesting results. Thus this dissertation only scratches the surface of such alternative paradigms, which include natural evolution itself (which has no final objective in the search space).

10.2 Novelty search and open-ended evolution

Novelty search suggests a perspective on open-endedness that is fitness-agnostic. Rather than viewing open-ended evolution as an adaptive competition, it can be viewed simply as a passive drift through the lattice of novelty. While this perspective bypasses a long-standing notion of adaptive innovation in open-ended evolution [9, 10, 108], it offers a complementary view that is recommended by its intuitive simplicity: Open-endedness can be defined simply as the continual production of novelty.

The benefit of this view is that it means that we can now endow *any* domain with this kind of open-endedness. No longer are we restricted to complex artificial life worlds in our pursuit of open-ended discovery. As long as novelty can be defined (which will not always be easy), it can be sought explicitly in every domain from simple XOR to the most complex artificial world, putting many practical problems in machine learning within its reach.

For example, it is difficult to evolve a checkers player from scratch against a fixed world-class opponent because early generation individuals are always completely defeated. Yet novelty search abandons the idea that winning is the goal; rather it can simply try to *lose in a different way.* As the approaches to losing are exhausted one by one, eventually it may cross the path to winning, avoiding all deception and providing an entirely new kind of practical search that is nevertheless open-ended.

Novelty search also helps to illustrate that true open-ended evolution is only a part of what makes natural evolution interesting. Although novelty search can discover novel behaviors of increasing complexity in any domain in which behavior can be quantified, ultimately complexity and diversity of behavior alone are not necessarily interesting. In other words, though there is theoretically no limit to the behaviors that can be discovered by novelty search in the "hard maze," (by virtue of there being infinite discrete points within any continuous space) few such behaviors, despite their complexity, will appear purposeful or interesting to us. Thus there appears to be an missing ingredient orthogonal to open-endedness that makes evolution's products so impressive.

The two extensions of novelty search introduced in this dissertation begin to address this missing ingredient by constraining search to ensure an evolved artifact is interesting in at least a minimal way. For example, an evolved artifact in MCNS must pass the minimal criteria, which means it at least has some minimal level of functionality. In NSLC, an evolved organism must maintain some locally-competitive functionality to pass the threshold of selection. However, the mystery of interestingness and how to better align it with the gradients of novelty remains an open philosophical question.

10.3 Alternate Abstractions of Natural Evolution

Novelty search opens up new possibilities for EC in general by suggesting new abstractions of natural evolution that can serve as the basis of evolutionary algorithms. That is, most EAs treat evolution as an objective-based optimizer by abstracting biological fitness as a *fitness function* that measures progress to an objective. The problem is that natural evolution does not search for an overall globally optimal organism like such an abstraction implies. The vast diversity of life stands as a living counterexample. Instead, evolution accumulates a wide variety of solutions to life's problems; the organisms in a particular niche may be driven to local optimality, but do not compete globally with all other niches. The most telling symptom that something may be wrong with the dominant abstraction in EC is that a characteristic feature of natural evolution is its drive to *diversity* while in contrast EAs almost universally *converge*. Thus it is possible that EC's foundation lies on the wrong bedrock if it is aspiring to truly recreate the creative power of evolution.

Novelty search suggests a number of alternate abstractions of natural evolution and algorithmic implementations for each that root the abstraction in computational reality. The first abstraction is evolution as a search for novel forms, which is the motivation behind the novelty search algorithm. The idea is to completely abandon the objective-driven abstraction of evolution for a diversity-driven one, with the intuition that accumulating a diversity of novel forms may be more central to evolution's creativity than its tendency to optimize biological fitness. This abstraction, as implemented by novelty search, has proven successful in many experiments in this dissertation and also those undertaken by other researchers [136, 91, 55, 117, 120, 97, 43].

The other two abstractions of natural evolution suggested by this dissertation reintegrate forms of selection pressure into a search for novelty in different ways. Minimal criteria novelty search does so in a coarse way by treating such pressure as a binary threshold. More specifically, MCNS prunes away behaviors from search that do not meet such a threshold. In effect it abstracts evolution as a search for many ways of doing the same thing; those behaviors that do not implement that *same thing* are considered inviable. That is, in nature, all organisms can be seen as surviving and reproducing in different ways, and those organisms that cannot survive and reproduce go extinct. What makes MCNS unique is that instead of abstracting the concept of *biological fitness* as a fitness function, MCNS abstracts the concept of *surviving and reproducing* as a set of minimal criteria that constrain the raw search for novel forms. This abstraction was tested in two experiments in this dissertation in Chapter 8.

The last abstraction of natural evolution introduced in this dissertation is less coarse; it integrates a continuous value of success (much like a fitness function) with the drive to accumulate novelty. However, importantly the drive to achieve is *secondary* to the drive towards diversity in contrast to most diversity maintenance techniques in which the reverse is true, i.e. diversity is only encouraged insofar as it may help increase performance. While in most EC approaches competition between evolved individuals is global, i.e. resource allocation is stilted towards the most easily-exploited niches (this is generally true even when applying diversity maintenance techniques), the *scale* of viable fitness values in nature varies enormously between biological niches. That is, bacteria reproduce in hours, humans in decades, and yet both are viable ways of life preserved by evolution. Novelty search with local competition introduces a local competition objective that drives towards optimality *within* a particular niche but not towards global convergence. NSLC thus makes it possible to evolve a wide diversity of individuals well-adapted to their particular niches, a step towards more life-like evolutionary dynamics. NSLC was validated in Chapter 9 in a culminating application of virtual creatures, where a diversity of locomoting creatures was evolved in a single run.

By abstracting the process by which natural evolution discovers novelty and enforcing minimal criteria or local competition in a spirit similar to the criterion of viability in natural evolution, this dissertation may bring EC one step closer to finally creating an explosion of diversity and complexity reminiscent of what happened in natural evolution. This exciting possibility is the most ambitious goal of this line of research: Novelty search may help us understand at a high level the "laws" driving the emergence of interesting complexity from the low-level principles of Darwinian evolution. Claims of such high-level understanding can be strengthened through reproducibility; to my knowledge, there has been no such recreation of such a complexity explosion in EC. Though it is a far cry from the grandeur of nature, the culminating application in this dissertation of evolving in a single run locomoting creatures in diverse niches is more reminiscent of evolutionary dynamics than prior virtual creature works, and may thus be a step along the way towards a true complexity explosion. This possibility is notable because if and when such a complexity explosion is created, it would widely impact the broader research community and could further deepen understanding of biological evolution and consequently, our own origins.

10.4 Future Work

The results in this dissertation demonstrate a new approach to open-ended evolution and practical EC, but do not meet the high bar of evolutionary impressiveness set by natural evolution. In particular this dissertation demonstrates that open-ended evolution does not necessarily connect with intuitions about interestingness: The products of an open-ended process need not always be impressive. Though novelty search can discover solutions to the “hard maze,” if the open-ended search continues beyond this point many other behaviors would be uncovered. Though initially novel, most of these maze navigators that ended in different points within the maze would not impress a human observer in the same way that novel biological organisms would. Thus a future research direction illuminated by this dissertation is to explore how to incentivize interestingness or impressiveness within evolutionary search to address this gap.

Also, though novelty search is in effect a coevolutionary algorithm because the discovery of one behavior can change the novelty of other behaviors, there is no *direct* interaction between evolved behaviors as in competitive coevolution. An interesting idea is to combine novelty search with competitive coevolution because there exist coevolutionary equivalents of deception that often prevent continually escalating evolutionary arms races [47, 25, 167]. Because novelty search can often circumvent deception in static evolutionary environments it may also be able to induce better coevolutionary dynamics as well. Although it is not immediately clear what the best way is to combine novelty search with competitive coevolution, a simple approach might be to characterize behavior as the outcome precipitated by two (or more) evolved behaviors interacting. For example, the behavior of two opposing checkers players might be characterized as its ending position.

CHAPTER 11
CONCLUSION

The main idea in this dissertation is a new evolutionary paradigm called novelty search that overcomes the problem of deception faced by evolutionary learning methods by abandoning objectives and instead searching only for behavioral novelty. Novelty search offers an alternative to the long-standing perspective that directly measuring and encouraging progress towards a goal is the best approach. Pursuing this alternative approach may help bring EC closer to the ambitious goal of replicating the apparent power of the process of natural evolution. This chapter summarizes the main contributions of the dissertation.

11.1 Contributions

In summary, this dissertation provides significant evidence for the research hypothesis that searching without objectives, as novelty search does, can be an effective alternative to the more familiar objective-based search paradigm:

1. This dissertation was the first to introduce novelty search, a search algorithm that I co-invented with Dr. Kenneth O. Stanley. Despite only being introduced in 2008 [101], novelty search has proven successful in a large number of domains [136, 91, 55, 117, 120, 102, 97, 43] by both researchers at UCF and from outside. The main

intellectual contribution of novelty search is to demonstrate that algorithms with no explicit objective can sometimes counter-intuitively be as or more effective than an algorithm explicitly seeking the objective.

2. Novelty search was validated by applying it to biped locomotion, a challenging machine learning domain. In experiments evolving a walking gait for a biped robot, novelty search significantly outperformed objective-based search. This result is important because it shows novelty search's ability to scale to real-world domains and its viability as a tool in the EC practitioner's toolbox.

3. The potential of novelty search to increase evolvability was demonstrated through experiments in maze navigation and biped locomotion. An important issue in EC is the discrepancy between natural evolution's high evolvability and the often-fragile representations evolved in EC. The results in the evolvability experiments show that novelty search may often uncover more evolvable solutions and better exploit self-adaptation, establishing that the choice of *reward scheme* and not only that of the search algorithm, can influence evolvability and the consequences (positive or negative) of features like self-adaption.

4. An extension of novelty search called minimal criteria novelty search (MCNS) was introduced that brings a new abstraction of natural evolution to EC. The dominant abstract of evolution in EC is based on optimizing a fitness function that serves as a heuristic of distance to a particular goal. MCNS introduces the alternative abstraction

of natural evolution as *many novel ways to do the same thing*, i.e. to survive and reproduce. Two maze-navigation experiments demonstrate the extension's potential.

5. Another extension of novelty search called novelty search with local competition (NSLC) was introduced, which abstracts natural evolution in a different way, stressing local achievement over global progress. The main idea is that competition in nature is not global (i.e. each kind of organism does not compete with *every* other kind). Such global competition would encourage convergence. Instead, competition is localized within niches, which encourages divergence through escaping from competition.

6. As a culminating application, NSLC was used to evolve a wide diversity of virtual creatures. In a step towards more powerful, creative evolutionary algorithms, NSLC evolved this wide diversity of functional organisms well-adapted to their particular niches in a single run. This result is significant because nature too preserves and accumulates a wide diversity of well-adapted novelty as it progresses, although EAs typically do not.

11.2 Conclusion

Novelty search is a viable new technique for evolving behaviors in EC. Counterintuitively, by abandoning objectives it is often able to outperform the direct search for the objective. Such an outcome was demonstrated through maze experiments in neuroevolution and in GP, biped locomotion, and evolving the gaits of virtual creatures. Novelty search was also demonstrated to increase evolvability, and to lead to two new abstractions of natural evolution that may bring EC closer to the ambitious goal of recreating processes with power and creativity similar in magnitude to that of natural evolution.

APPENDIX: EXPERIMENTAL PARAMETERS

A.1 Parameters for NEAT-based experiments

In many experiments in this dissertation, novelty search was implemented as an extension of NEAT, which when run with a traditional objective function also served as the control algorithm. Thus to facilitate a fair comparison, NEAT with novelty search and objective-based NEAT have *identical* settings for all NEAT-specific parameters. Because the biped experiment is more challenging and operated with CTRNNs, the NEAT parameters required some adjustment, as did the MCNS mazes in Chapter 8, but they were always identical for both NEAT with novelty search and objective-based NEAT. The software package used in all such NEAT-based experiments, Novelty Search C++, which is an extended version of the real-time NEAT (rtNEAT) software package, is available at `http://eplex.cs.ucf.edu`. Table A.1 shows the parameters used in the NEAT-based experiments. NEAT has been found to be robust to moderate variations in parameters [155, 157, 151]. Additionally, table A.2 describes the physical parameters particular to the biped experiment. This section also provides a detailed explanation of each NEAT parameter.

1. **Population Size**: The number of ANNs that compose the population.

2. c_1: When calculating the compatibility distance between genomes defines the weight given to excess genes.

3. c_2: When calculating the compatibility distance between genomes defines the weight given to disjoint genes.

4. c_3: When calculating the compatibility distance between genomes defines the weight given to connection strength differences.

5. C_t: The compatibility threshold. Two individuals are considered to be in separate species if their compatibility distance is greater than this threshold. In the experiments in this dissertation this value is adjusted dynamically to maintain a target number of species.

6. **Add Link Probability**: Specifies how often mutations that add a new connection to an ANN occur.

7. **Add Node Probability**: Specifies how often mutations that add a new node to an ANN occur.

8. **Mutate Time Constant Probability**: Specifies how often mutations that change the time constant of a node in a CTRNN occur (only applies to the biped experiment).

9. **Mutate Bias Probability**: A probability that defines how often mutations that change the bias of a node in a CTRNN occur (only applies to the biped experiment).

Parameter	**Mazes (ch. 4)**	**Biped (ch. 6)**	**MCNS mazes (ch. 8)**
Pop. Size	250	500	500
c_1	1.0	1.0	1.0
c_2	1.0	1.0	1.0
c_3	3.0	3.0	3.0
C_t	variable	variable	variable
Prob. Add Link	0.1	0.06	0.05
Prob. Add Node	0.005	0.005	0.02
Prob. Mutate Time Constant	*N/A*	0.3	*N/A*
Prob. Mutate Bias	*N/A*	0.3	*N/A*
Initial Archive Threshold	6.0	1.0	6.0
K-Nearest Neighbors	15	15	15

Table A.1: NEAT Parameter Settings. This table shows the NEAT parameters for the maze experiments in Chapter 4, the biped experiment in Chapter 6, and the mazes in the MCNS experiment in Chapter 8. Archive threshold and k-nearest neighbor parameters apply only to NEAT with novelty search, while time constant and bias parameters apply only to the biped experiment.

10. **Initial Archive Threshold**: A novelty search parameter that defines the threshold level of novelty an individual must exceed to be added to the archive. It is dynamically adjusted during search to encourage consistent growth of the archive.

11. **K-Nearest Neighbors**: A novelty search parameter that determines how large of a neighborhood (i.e. how many nearest neighbors) to consider when calculating the novelty of a new individual.

Parameter	**Value**
Foot Radius	0.17 meters
Foot Density	1.0 kilograms per cubic meter
Torso Radius	0.1 meters
Torso Length	0.33 meters
Torso Density	1.0 kilograms per cubic meter
Leg Segment Radius	0.2 meters
Leg Segment Length	0.33 meters
Leg Segment Density	1.0 kilograms per cubic meter
Maximum Torque	5.0 newton meters
Proportional Constant	9.0

Table A.2: Biped Simulation Parameter Settings. Parameters are shown for the physical simulation of the biped robot from Chapter 6, implemented using the freely available Open Dynamics Engine library. Maximum torque is the most torque a proportional controller can apply. The proportional constant is multiplied by the disparity between the actual angle of the joint and the angle that the ANN requests to derive the torque to be applied at that joint.

A.2 Parameters for GP-based experiments

In addition to NEAT, novelty search was also implemented as an extension of GP in several experiments in this dissertation. In such novelty search GP experiments, traditional objective-based GP served as the control algorithm. As in the NEAT-based experiments, to facilitate a fair comparison GP with novelty search and objective-based GP have *identical* settings for all GP-specific parameters. Table A.3 shows the parameters used for the GP maze experiments, table A.4 shows those for the artificial ant experiments, and table A.5 shows those for the maze experiments with scalable problem difficulty. In addition, this section also explains the meaning of each parameter.

1. **Objective**: This parameter defines what must be achieved to successfully conclude an experiment.

2. **Terminal Set**: The set of possible program nodes that can be leaves (i.e. nodes without children) in a valid GP program tree for a particular experiment.

3. **Functions Set**: The set of possible nodes that can be interior nodes (i.e. nodes with children) in a valid GP program tree for a particular experiment.

4. **Fitness Cases**: The variations of a domain to be considered in an experiment (e.g. different mazes for the maze experiment or different trails for the artificial ant experiment).

5. **Wrapper**: Describes how a GP program tree is evaluated, i.e. how an evolved program is applied in a particular experiment.

6. **Population Size**: The number of GP program trees that compose the population and are evaluated in each generation.

7. **Termination**: The conditions to be met before an experiment will be terminated.

Objective:	Find a robot that navigates the maze
Terminal set:	Left (turn left), Right (turn right), Move (move forward one square)
Functions set:	IfWallAhead (execute one of two child instructions based on whether there is a wall directly ahead), Prog2 (sequentially execute the two child instructions)
Fitness cases:	Medium Maze and Hard Maze
Wrapper:	Program repeatedly executed for 100 time steps for the medium maze or 400 time steps for the hard maze
Population Size:	1,000
Termination:	Maximum number of generations = 1,000

Table A.3: Maze Problem. The parameters for the GP maze problem are shown.

Objective:	Find an ant that follows food trails
Terminal set:	Left (turn left), Right (turn right), Move (move forward one square)
Functions set:	IfFoodAhead (execute one of two child instructions based on if there is food directly ahead), Prog2 (sequentially execute the two children instructions)
Fitness cases:	Santa Fe Trail and Los Altos Trail
Wrapper:	Program repeatedly executed for 400 time steps for Santa Fe Trail or 3,000 time steps for Los Altos Trail
Population Size:	1,000
Termination:	Maximum number of generations = 1,000

Table A.4: Artifical Ant Problem. The parameters for the GP artificial ant problem are listed.

Objective:	Find a robot that navigates the maze
Terminal set:	Left (turn left), Right (turn right), Move (move forward one square)
Functions set:	IfWallAhead (execute one of two child instructions based on whether there is a wall directly ahead), IfGoalAhead (execute one of two child instructions based on whether the goal is within a 90 degree cone projected outwards from where the robot is facing), Prog2 (sequentially execute the two child instructions)
Fitness cases:	One of 360 randomly-generated mazes
Wrapper:	Program repeatedly executed for 200 time steps
Population Size:	500
Termination:	Maximum number of generations = 200, 400 and 600

Table A.5: Scalable-difficulty Maze parameters. The parameters for the variant of the GP maze problem with scalable problem difficulty are shown.

LIST OF REFERENCES

[1] T. Aaltonen et al. Measurement of the top quark mass with dilepton events selected using neuroevolution at CDF. *Physical Review Letters*, 2009.

[2] Christoph Adami, Charles Ofria, and Travis C. Collier. Evolution of biological complexity. *Proceedings of the National Academy of Sciences, USA*, 97:4463–4468, 2000.

[3] Brian Allen and Petros Faloutsos. Complex networks of simple neurons for bipedal locomotion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.

[4] Lee Altenberg. Fitness distance correlation analysis: An instructive counterexample. In *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 57–64. Morgan Kaufmann, 1997.

[5] WB Arthur. On the evolution of complexity. In G Cownan, D Pines, and D Meltzer, editors, *Complexity: Metaphors, Models and Reality*, pages 65–81. Addison-Wesley, Reading, MA, 1994.

[6] J.E. Auerbach and J.C. Bongard. Dynamic Resolution in the Co-Evolution of Morphology and Control. In *Proceedings of the Twelfth International Conference on Artificial Life (ALIFE XII)*, 2010.

[7] Lionel Barnett. Ruggedness and neutrality - the NKp family of fitness landscapes. In *ALife VI: Sixth International Conference on Articial Life*, pages 18–27. MIT Press, 1998.

[8] Lionel Barnett. Netcrawling - optimal evolutionary search with neutral networks. In *Proceedings of the 2001 IEEE International Conference on Evolutionary Computation*, pages 30–37. IEEE Press, 2001.

[9] M. A. Bedau and N. H. Packard. Measurement of evolutionary activity, teleology, and life. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Proceedings of Artificial Life II*, pages 431–461, Redwood City, CA, 1991. Addison-Wesley.

[10] M. A. Bedau, E. Snyder, and N. H. Packard. A classification of longterm evolutionary dynamics. In C. Adami, R. Belew, H. Kitano, and C. Taylor, editors, *Proceedings of Artificial Life VI*, pages 228–237, Cambridge, MA, 1998. MIT Press.

[11] Mark Bedau. Four puzzles about life. *Artificial Life*, 4:125–140, 1998.

[12] Hamid Benbrahim. *Biped dynamic walking using reinforcement learning.* PhD thesis, Durham, NH, USA, 1996. Director-Miller,III, W. Thomas.

[13] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, 1975.

[14] A. D. Bethke. *Genetic Algorithms as Function Optimizers.* Phd thesis, University of Michigan, 1981.

[15] J.C. Bongard and R. Pfeifer. Evolving complete agents using artificial ontogeny. *Morpho-functional machines: The new species (designing embodied intelligence)*, pages 237–258, 2003.

[16] Yossi Borenstein and Riccardo Poli. Fitness distributions and GA hardness. In *Proceedings of PPSN*, pages 11–20. Springer, 2004.

[17] Dimo Brockhoff, Tobias Friedrich, Nils Hebbinghaus, Christian Klein, Frank Neumann, and Eckart Zitzler. Do additional objectives make a problem harder? In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 765–772, New York, NY, USA, 2007. ACM.

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

[19] John Cartlidge and Seth Bullock. Combating coevolutionary disengagement by reducing parasite virulence. *Evolutionary Computation*, 12(2):193–222, 2004.

[20] John Cartlidge and Seth Bullock. Unpicking tartan CIAO plots: Understanding irregular coevolutionary cycling. *Adaptive Behavior*, 12(2):69–92, 2004.

[21] A. Channon. Passing the alife test: Activity statistics classify evolution in geb as unbounded. In *Proceedings of the European Conference on Artificial Life(ECAL-2001)*. Springer, 2001.

[22] A. D. Channon and R. I. Damper. Towards the evolutionary emergence of increasingly complex advantageous behaviours. *International Journal of Systems Science*, 31(7): 843–860, 2000.

[23] Kumar Chellapilla and David B. Fogel. Evolving neural networks to play checkers without relying on expert knowledge. *IEEE Transactions on Neural Networks*, 10(6): 1382–1391, 1999.

[24] Steffen Christensen and Franz Oppacher. Solving the artificial ant on the santa fe trail problem in 20,696 fitness evaluations. In *GECCO '07: Proceedings of the 9th annual*

conference on Genetic and evolutionary computation, pages 1574–1579, New York, NY, USA, 2007. ACM.

[25] Dave Cliff and Geoffrey Miller. Tracking the red queen: Measurements of adaptive progress in co-evolutionary simulations. pages 200–218. 1995.

[26] J. Clune, D. Misevic, C. Ofria, R.E. Lenski, S.F. Elena, and R. Sanjuán. Natural selection fails to optimize mutation rates for long-term adaptation on rugged fitness landscapes. *PLoS Computational Biology*, 4(9):1145–1146, 2008.

[27] C.A.C. Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information systems*, 1(3):129–156, 1999.

[28] G. Cuccu and F. Gomez. When novelty is not enough. *Applications of Evolutionary Computation*, pages 234–243, 2011.

[29] G. Cuccu, F. Gomez, and T. Glasmachers. Novelty-based restarts for evolution strategies. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 158–163. IEEE, 2011.

[30] Paul Darwen and Xin Yao. Every niching method has its niche: Fitness sharing and implicit sharing compared. In *In Proc. of Parallel Problem Solving from Nature (PPSN) IV - Lecture Notes in Computer Science 1141*, pages 398–407. Springer-Verlag, 1996.

[31] Charles Darwin. *On the Origin of Species by Means of Natural Selection or the Preservation of Favored Races in the Struggle for Life.* Murray, London, 1859.

[32] Yuval Davidor. Epistasis variance: A viewpoint on GA-hardness. In *Proceedings of the First Workshop on Foundations of Genetic Algorithms.*, pages 23–35. Morgan

Kaufmann, 1990.

[33] R. Dawkins. Human chauvinism. *Evolution*, 51(3):1015–1020, 1997.

[34] Richard Dawkins. *The Selfish Gene.* Oxford University Press, Oxford, UK, 3rd edition, 1989.

[35] Richard Dawkins. Genetic and evolutionary computation conference (GECCO-2007) Keynote Debate, July 2007.

[36] E. D. De Jong. The incremental pareto-coevolution archive. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004)*, Berlin, 2004. Springer Verlag.

[37] Edwin D. De Jong and Jordan B. Pollack. Multi-objective methods for tree size control. *Genetic Programming and Evolvable Machines*, 4(3):211–233, September 2003.

[38] Kenneth A. De Jong. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems.* PhD thesis, The University of Michigan, Ann Arbor, MI, 1975. University Microfilms No. 76-09381.

[39] Kenneth A. De Jong. *Evolutionary Computation: A Unified Approach.* MIT Press, 2006.

[40] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2): 182–197, 2002.

[41] Kalyanmoy Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 7:205–230, 1999.

[42] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2000.

[43] John Doucette. Novelty-based fitness measures in genetic programming. Master of science in computer science, Dalhouise University, 2010.

[44] M. Ebner, M. Shackleton, and R. Shipman. How neutral networks influence evolvability. *Complexity*, 7(2):19–33, 2001.

[45] Jeffrey L. Elman. Incremental learning, or the importance of starting small. Technical Report 9101, CRL, La Jolla, CA, 1991.

[46] J. A. Endler and J. J. D. Greenwood. Frequency-Dependent Predation, Crypsis and Aposematic Coloration [and Discussion]. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 319(1196):505–523, 1988.

[47] Sevan Ficici and Jordan B. Pollack. Challenges in coevolutionary learning: Arms-race dynamics, open-endedness, and mediocre stable states. In *Proceedings of the Sixth International Conference on Artificial Life*, pages 238–247. MIT Press, 1998.

[48] L.J. Fogel, A.J. Owens, and M.J. Walsh. Artificial intelligence through simulated evolution. 1966.

[49] Chris Gathercole and Peter Ross. An adverse interaction between crossover and restricted tree depth in genetic programming. In *GECCO '96: Proceedings of the First Annual Conference on Genetic Programming*, pages 291–296, Cambridge, MA, USA, 1996. MIT Press.

[50] M.R. Glickman and K. Sycara. Reasons for premature convergence of self-adapting mutation rates. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 1, pages 62–69. IEEE, 2002.

[51] F. Glover. Tabu search, part I. *ORSA Journal on Computing*, 1(3):190–206, Summer 1989.

[52] David Goldberg, K. Deb, and B. Korb. Messy genetic algorithms: motivation, analysis, and first results. *Complex Systems*, (3):493–530, 1989.

[53] David E. Goldberg. Simple genetic algorithms and the minimal deceptive problem. In L. D. Davis, editor, *Genetic Algorithms and Simulated Annealing, Re- search Notes in Artificial Intelligence*. Morgan Kaufmann, 1987.

[54] David E. Goldberg and Jon Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms*, pages 41–49, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc.

[55] H.J. Goldsby and B.H.C. Cheng. Automatically Discovering Properties that Specify the Latent Behavior of UML Models. In *Proceedings of MODELS 2010*, 2010.

[56] Faustino Gomez and Risto Miikkulainen. Incremental evolution of complex general behavior. *Adaptive Behavior*, 5:317–342, 1997.

[57] F.J. Gomez. Sustaining diversity using behavioral information distance. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 113–120. ACM, 2009.

[58] Steven Jay Gould. *Full House: The Spread of Excellence from Plato to Darwin.* Harmony Books, 1996.

[59] J.J. Grefenstette. Evolvability in dynamic fitness landscapes: A genetic algorithm approach. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 3. IEEE, 2002.

[60] John J. Grefenstette. Deception considered harmful. In *Foundations of Genetic Algorithms 2*, pages 75–91. Morgan Kaufmann, 1992.

[61] David Greiner, Jose M. Emerador, Gabriel Winter, and Blas Galvan. Improving computational mechanics optimum design using helper objectives: An application in frame bar structures. In *Evolutionary Multi-Criterion Optimization*, pages 575–589. 2007.

[62] Haipeng Guo and William H. Hsu. Ga-hardness revisited. In *Genetic and Evolutionary Computation - GECCO 2003, Genetic and Evolutionary Computation Conference*, pages 12–16. Springer, 2003.

[63] Julia Handl, Simon C. Lovell, and Joshua Knowles. Investigations into the effect of multiobjectivization in protein structure prediction. In *Proceedings of the 10th international conference on Parallel Problem Solving from Nature*, pages 702–711, Berlin, Heidelberg, 2008. Springer-Verlag.

[64] Julia Handl, Simon C. Lovell, and Joshua Knowles. Multiobjectivization by decomposition of scalar cost functions. In *Proceedings of the 10th international conference on Parallel Problem Solving from Nature*, pages 31–40, Berlin, Heidelberg, 2008. Springer-Verlag.

[65] Inman Harvey. *The Artificial Evolution of Adaptive Behavior*. PhD thesis, School of Cognitive and Computing Sciences, University of Sussex, Sussex, 1993.

[66] Inman Harvey and Adrian Thompson. Through the labyrinth evolution finds a way: A silicon ridge. In *Proceedings of the First International Conference on Evolvable Systems: From Biology to Hardware*, pages 406–422. Springer-Verlag, 1996.

[67] 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.

[68] J.D. Hiller and H. Lipson. Evolving amorphous robots. In *Proceedings of the 12th International Conference on Artificial Life (ALIFE XII)*, 2010.

[69] W. D. Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. In J. D. Farmer, C. Langton, S. Rasmussen, and C. Taylor, editors, *Artificial Life II*. Addison-Wesley, Reading, MA, 1991.

[70] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. University of Michigan Press, Ann Arbor, MI, 1975.

[71] Wim Hordijk. A measure of landscapes. *Evolutionary Computation*, 4:335–360, 1995.

[72] Jeffrey Horn, Nicholas Nafpliotis, and David E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *In Proceedings of the First IEEE International Conference on Evolutionary Computation*, pages 82–87, 1994.

[73] Gregory S. Hornby. ALPS: the age-layered population structure for reducing the problem of premature convergence. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2006)*, pages 815–822, New York, NY, USA, 2006. ACM.

[74] G.S. Hornby, H. Lipson, and J.B. Pollack. Evolution of generative design systems for modular physical robots. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 4146–4151. IEEE; 1999, 2001.

[75] G.S. Hornby, J.D. Lohn, and D.S. Linden. Computer-automated evolution of an x-band antenna for nasa's space technology 5 mission. *Evolutionary Computation*, 19(1): 1–23, 2011.

[76] Jianjun Hu, Erik Goodman, Kisung Seo, Zhun Fan, and Rondal Rosenberg. The hierarchical fair competition (HFC) framework for sustainable evolutionary algorithms. *Evolutionary Computation*, 13(2):241–277, 2005.

[77] Kimberly A. Hughes, Linh Du, F. Helen Rodd, and David N. Reznick. Familiarity leads to female mate preference for novel males in the guppy, poecilia reticulata. *Animal Behavior*, 58(4):907–916, 1999.

[78] Marcus Hutter and Shane Legg. Fitness uniform optimization. *IEEE Transactions on Evolutionary Computation*, 10:568–589, 2006.

[79] Julian Huxley. *Essays of a biologist, by Julian Huxley.* Chatto & Windus, London, 1923.

[80] Martijn Huyen. Exploring phenotype space through neutral evolution. Working Papers 95-10-100, Santa Fe Institute, October 1995.

[81] Akio Ishiguro, Akinobu Fujii, and Peter Eggenberger Hotz. Neuromodulated control of bipedal locomotion using a polymorphic CPG circuit. *Adaptive Behavior*, 11(1): 7–17, 2003.

[82] Terry Jones and Stephanie Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 184–192. Morgan Kaufmann, 1995.

[83] Mark Juric. An anti-adaptationist approach to genetic algorithms. In *Proceedings of IEEE Conference on Evolutionary Computation*, volume 2, pages 619–623. IEEE Press, 1994.

[84] George Kampis and László Gulyás. Full body: The importance of the phenotype in evolution. *Artificial Life*, 14(3):375–386, 2008.

[85] F. Kaplan and V.V. Hafner. Information-theoretic framework for unsupervised activity classification. *Advanced Robotics*, 20(10):1087–1103, 2006.

[86] Duško Katić and Miomir Vukobratović. Survey of intelligent control techniques for humanoid robots. *Journal of Intelligent Robotics Systems*, 37(2):117–141, 2003.

[87] Stuart A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution.* Oxford University Press, USA, 1 edition, June 1993.

[88] Motoo Kimura. *The neutral theory of molecular evolution.* Cambridge University Press, 1983.

[89] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

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

[91] S. Kistemaker and S. Whiteson. Critical factors in the performance of novelty search. In *GECCO 2011: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 965–972. ACM, 2011.

[92] Joshua D. Knowles, Richard A. Watson, and David Corne. Reducing local optima in single-objective problems by multi-objectivization. In *EMO '01: Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, pages 269–283, London, UK, 2001. Springer-Verlag.

[93] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, 1992.

[94] John R. Koza. *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, Cambridge Massachusetts, May 1994.

[95] Oliver Kramer. Evolutionary self-adaptation: a survey of operators and strategy parameters. *Evolutionary Intelligence*, 3:51–65, 2010. 10.1007/s12065-010-0035-y.

[96] P. Krcah. Evolutionary development of robotic organisms. *Master's thesis, Charles University in Prague*, 2007.

[97] Peter Krcah. Solving deceptive tasks in robot body-brain co-evolution by searching for behavioral novelty. In *ISDA*, pages 284–289. IEEE, 2010.

[98] W. B. Langdon and R. Poli. Fitness causes bloat. In *Soft Computing in Engineering Design and Manufacturing*, pages 23–27. Springer-Verlag, 1997.

[99] W. B. Langdon and R. Poli. Why ants are hard. In *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pages 193–201, University of Wisconsin, Madison, Wisconsin, USA, 22-25 1998. Morgan Kaufmann.

[100] Benoit Leblanc and Evelyne Lutton. Bitwise regularity and GA-hardness. In *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*, 1998.

[101] Joel Lehman and Kenneth O. Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *Proceedings of the Eleventh International Conference on Artificial Life (ALIFE XI)*, Cambridge, MA, 2008. MIT Press.

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

[103] Gunar E. Liepins and Michael D. Vose. Deceptiveness and genetic algorithm dynamics. In Gregory J. E. Rawlins, editor, *FOGA*, pages 36–50. Morgan Kaufmann, 1990.

[104] Sean Luke. Genetic programming produced competitive soccer softbot teams for RoboCup97. In *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pages 214–222, University of Wisconsin, Madison, Wisconsin, USA, 1998. Morgan Kaufmann.

[105] Michael Lynch. The evolution of genetic networks by non-adaptive processes. *Nature Reviews Genetics*, 8:803–813, 2007.

[106] Michael Lynch. The frailty of adaptive hypotheses for the origins of organismal complexity. In *Proc Natl Acad Sci USA*, volume 104, pages 8597–8604, 2007.

[107] Samir W. Mahfoud. *Niching methods for genetic algorithms.* PhD thesis, Champaign, IL, USA, 1995.

[108] C. C. Maley. Four steps toward open-ended evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference(GECCO-1999)*, volume 2, pages 1336–1343, Orlando, Florida, USA, 1999. IEEE Press.

[109] Bernard Manderick, Mark K. de Weger, and Piet Spiessens. The genetic algorithm and the structure of the fitness landscape. In Richard K. Belew and Lashon B. Booker, editors, *ICGA*, pages 143–150. Morgan Kaufmann, 1991.

[110] Andrew P. Martin. Increasing genomic complexity by gene duplication and the origin of vertebrates. *The American Naturalist*, 154(2):111–128, 1999.

[111] Gary McHale and Phil Husbands. Gasnets and other evolvable neural networks applied to bipedal locomotion. In *From Animals to Animats 8*, 2004.

[112] Daniel W. McShea. Complexity and evolution: What everybody knows. *Biology and Philosophy*, 6(3):303–324, 1991.

[113] Thomas Miconi. Evolution and complexity: The double-edged sword. *Artificial Life: Special Issue on the Evolution of Complexity*, 2007.

[114] Thomas Miconi. *THE ROAD TO EVERYWHERE: Evolution, Complexity and Progress in Nature and in Computers.* PhD thesis, University of Birmingham, 2007.

[115] Melanie Mitchell, Stephanie Forrest, and John H. Holland. The royal road for genetic algorithms: Fitness landscapes and ga performance. In F. J. Varela and P. Bourgine, editors, *Proceedings of the First European Conference on Artificial Life*, Cambridge,

MA, 1992. MIT Press.

[116] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, New York, NY, 1997.

[117] Jean-Baptiste Mouret. Novelty-based multiobjectivization. In *Proceedings of the Workshop on Exploring New Horizons in Evolutionary Design of Robots,2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[118] C. L. Nehaniv and J. L. Rhodes. On the manner in which biological complexity may grow. In *Mathematical and Computational Biology*, volume 26 of *Lectures on Mathematics in the Life Sciences*, pages 93–102. American Mathematical Society, 1999.

[119] Sooyol Ok and DuckSool Kim. Evolution of the CPG with sensory feedback for bipedal locomotion. In *Advances in Natural Computation*. Springer Berlin / Heidelberg, 2005.

[120] C. Ollion and S. Doncieux. Why and how to measure exploration in behavioral space. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 267–274. ACM, 2011.

[121] P.Y. Oudeyer, F. Kaplan, and V.V. Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, 11(2):265–286, 2007.

[122] P.Y. Oudeyer, F. Kaplan, V.V. Hafner, and A. Whyte. The playground experiment: Task-independent development of a curious robot. In *Proceedings of the AAAI Spring Symposium on Developmental Robotics*, pages 42–47, 2005.

[123] Chandana Paul. Bilateral decoupling in the neural control of biped locomotion. In *Proc. 2nd International Symposium on Adaptive Motion of Animals and Machines*,

2003.

[124] Martin Pelikan, Martin Pelikan, David E. Goldberg, and David E. Goldberg. Escaping hierarchical traps with competent genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 511–518. Morgan Kaufmann, 2001.

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

[126] Jordan Pollack, Alan D. Blair, and Mark Land. Coevolution of a backgammon player. In *Proceedings of Artificial Life V*, pages 92–98. MIT Press, 1996.

[127] Elena Popovici and Kenneth De Jong. Understanding cooperative co-evolutionary dynamics via simple fitness landscapes. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 507–514, New York, NY, USA, 2005. ACM.

[128] T. Ray. Evolution, ecology and optimization of digital organisms. Technical Report Working paper 92-08-042, Santa Fe Institute, 1992.

[129] I. Rechenberg. Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution. frommann-holzbog, stuttgart, 1973. *Step-Size Adaptation Based on Non-Local Use of Selection Information. In Parallel Problem Solving from Nature (PPSN3)*, 1994.

[130] Torsten Reil and Phil Husbands. Evolution of central pattern generators for bipedal walking in a real-time physics environment. *IEEE Transactions on Evolutionary Computation*, 6(2):159–168, April 2002.

[131] J. Reisinger, E. Bahceci, I. Karpov, and R. Miikkulainen. Coevolving strategies for general game playing. In *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on*, pages 320–327, 2007.

[132] Joseph Reisinger, Kenneth O. Stanley, and Risto Miikkulainen. Towards an empirical measure of evolvability. In *Genetic and Evolutionary Computation Conference (GECCO2005) Workshop Program*, pages 257–264, Washington, D.C., 2005. ACM Press.

[133] AM Reynolds. Maze-solving by chemotaxis. *Physical Review E*, 81(6), 2010.

[134] Esteban Ricalde and Katya Rodríguez Vázquez. A GP neutral function for the artificial ant problem. In *GECCO '07: Proceedings of the 9th conference companion on Genetic and evolutionary computation*, pages 2565–2571, New York, NY, USA, 2007. ACM.

[135] Robert J. Richards. The moral foundations of the idea of evolutionary progress: Darwin, spencer, and the neo-darwinians. In Matthew Nitecki, editor, *Evolutionary Progress*. University of Chicago Press, Chicago, 1988.

[136] S. Risi, C.E. Hughes, and K.O. Stanley. Evolving plastic neural networks with novelty search. *Adaptive Behavior*, 2010.

[137] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465 – 471, 1978.

[138] Franz Rothlauf and David E. Goldberg. *Representations for Genetic and Evolutionary Algorithms*. Physica-Verlag, 2002.

[139] G. Rudolph. Self-adaptive mutations may lead to premature convergence. *Evolutionary Computation, IEEE Transactions on*, 5(4):410–414, 2002.

[140] Michael Ruse. Evolution and progress. *Tree*, 8(2):55–60, 1993.

[141] Conor Ryan. Pygmies and civil servants. pages 243–263, 1994.

[142] J. Schmidhuber. Exploring the predictable. In S. Ghosh and Tsutsui S., editors, *Advances in Evolutionary Computing: theory and applications*, pages 579–612. Springer-Verlag New York, 2003.

[143] J. Schmidhuber. Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts. *Connection Science*, 18(2):173–187, 2006.

[144] M.D. Schmidt and H. Lipson. Coevolution of fitness predictors. *Evolutionary Computation, IEEE Transactions on*, 12(6):736–749, Dec. 2008.

[145] Karl Sigmund. *Games of Life: Explorations in Ecology, Evolution and Behaviour*. Oxford University Press, Inc., New York, NY, USA, 1993.

[146] K. Sims. Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 15–22. ACM, 1994.

[147] Karl Sims. Evolving 3D morphology and behavior by competition. In Rodney A. Brooks and Pattie Maes, editors, *Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems (Artificial Life IV)*, pages 28–39. MIT Press, Cambridge, MA, 1994.

[148] J. Skipper. The computer zoo–evolution in a box. In *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 355–364. MIT Press, 1992.

[149] Terence Soule. *Code Growth in Genetic Programming.* PhD thesis, University of Idaho, Moscow, Idaho, USA, 15 May 1998.

[150] Russell Standish. Open-ended artificial evolution. *International Journal of Computational Intelligence and Applications*, 3(167), 2003.

[151] Kenneth O. Stanley, Bobby D. Bryant, and Risto Miikkulainen. Real-time neuroevolution in the NERO video game. *IEEE Transactions on Evolutionary Computation Special Issue on Evolutionary Computation and Games*, 9(6):653–668, 2005.

[152] Kenneth O. Stanley, Nate Kohl, and Risto Miikkulainen. Neuroevolution of an automobile crash warning system. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2005.

[153] Kenneth O. Stanley and Risto Miikkulainen. The dominance tournament method of monitoring progress in coevolution. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002) Workshop Program*, San Francisco, 2002. Kaufmann.

[154] Kenneth O. Stanley and Risto Miikkulainen. Efficient evolution of neural network topologies. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02)*, Piscataway, NJ, 2002. IEEE. In press.

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

[156] Kenneth O. Stanley and Risto Miikkulainen. A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130, 2003.

[157] Kenneth O. Stanley and Risto Miikkulainen. Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21:63–100, 2004.

[158] Kenneth O. Stanley and Risto Miikkulainen. Evolving a roving eye for Go. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004)*, Berlin, 2004. Springer Verlag.

[159] K.O. Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131–162, 2007.

[160] T. C. Stewart. Extrema selection: Accelerated evolution on neutral networks. In *Proceedings of the 2001 IEEE International Conference on Evolutionary Computation*. IEEE Press, 2001.

[161] Tim Taylor and John Hallam. Studying evolution with self-replicating computer programs. In *Fourth European Conference on Artificial Life*, pages 550–559. MIT Press, 1997.

[162] Michiel van de Panne and Alexis Lamouret. Guided optimization for balanced locomotion. In D. Terzopoulos and D. Thalmann, editors, *Sixth Eurographics Workshop on Animation and Simulation*, volume 95, pages 165–177. Springer-Verlag, 1995.

[163] Vesselin K. Vassilev, Terence C. Fogarty, and Julian F. Miller. Smoothness, ruggedness and neutrality of fitness landscapes: from theory to application. pages 3–44, 2003.

[164] David A. Van Veldhuizen and Gary B. Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary Computation*, 8(2):125–147, 2000. PMID: 10843518.

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

[166] James D. Watson, Nancy H. Hopkins, Jeffrey W. Roberts, Joan A. Steitz, and Alan M. Weiner. *Molecular Biology of the Gene Fourth Edition.* The Benjamin Cummings Publishing Company, Inc., Menlo Park, CA, 1987.

[167] Richard Watson and Jordan Pollack. Coevolutionary dynamics in a minimal substrate. pages 702–709. Morgan Kaufmann, 2001.

[168] D.S. Weile and E. Michielssen. Genetic algorithm optimization applied to electromagnetics: A review. *Antennas and Propagation, IEEE Transactions on*, 45(3):343–353, 1997.

[169] Edward Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics*, 63(5):325–336, 1990.

[170] Shimon Whiteson and Peter Stone. Evolutionary function approximation for reinforcement learning. *Journal of Machine Learning Research*, 7:877–917, May 2006.

[171] L. Darrell Whitley. Fundamental principles of deception in genetic search. In *Foundations of Genetic Algorithms*, pages 221–241. Morgan Kaufmann, 1991.

[172] David H. Wolpert and William Macready. No free lunch theorems for search. Technical Report SFI-TR-95-01-010, The Santa Fe Institute, Santa Fe, NM, 1995.

[173] Sewall Wright. The roles of mutation, inbreeding, crossbreeding, and selection in evolution. *Proceedings of the Sixth International Congress on Genetics*, 1932.

[174] Larry Yaeger. Computational genetics, physiology, metabolism, neural systems, learning, vision and behavior or polyworld: Life in a new context. In C. G. Langton, editor, *Artificial Life III, Proceedings Volume XVII*, pages 263–298. Addison-Wesley, 1994.

[175] Xin Yao. An empirical study of genetic operators in genetic algorithms. *Microprocessing and Microprogramming*, 38(1-5):707–714, 1993.

[176] N. Zaera, D. Cliff, and J. Bruten. (Not) evolving collective behaviours in synthetic fish. In *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*. MIT Press Bradford Books., 1996.

[177] Douglas Zongker and Bill Punch. lilgp 1.01 user's manual. Technical report, Michigan State University, USA, 26 March 1996.