2012

Novelty-assisted Interactive Evolution Of Control Behaviors

Brian G. Woolley
University of Central Florida

STARS Citation
Woolley, Brian G., "Novelty-assisted Interactive Evolution Of Control Behaviors" (2012). *Electronic Theses and Dissertations, 2004-2019*. 2256.
https://stars.library.ucf.edu/etd/2256

NOVELTY-ASSISTED INTERACTIVE EVOLUTION OF CONTROL BEHAVIORS

by

BRIAN G. WOOLLEY
B.S. California State University, Sacramento, 2002
M.S. Air Force Institute of Technology, 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

The field of evolutionary computation is inspired by the achievements of natural evolution, in which there is no final objective. Yet the pursuit of objectives is ubiquitous in simulated evolution because evolutionary algorithms that can consistently achieve established benchmarks are lauded as successful, thus reinforcing this paradigm. A significant problem is that such objective approaches assume that intermediate stepping stones will increasingly resemble the final objective when in fact they often do not. The consequence is that while solutions may exist, searching for such objectives may not discover them. This problem with objectives is demonstrated through an experiment in this dissertation that compares how images discovered serendipitously during interactive evolution in an online system called Picbreeder cannot be rediscovered when they become the final objective of the very same algorithm that originally evolved them. This negative result demonstrates that pursuing an objective limits evolution by selecting offspring only based on the final objective. Furthermore, even when high fitness is achieved, the experimental results suggest that the resulting solutions are typically brittle, piecewise representations that only perform well by exploiting idiosyncratic features in the target. In response to this problem, the dissertation next highlights the importance of leveraging human insight *during* search as an alternative to articulating explicit objectives. In particular, a new approach called *novelty-assisted interactive evolutionary computation* (NA-IEC) combines human intuition with a method called novelty search for the first time to facilitate the serendipitous discovery of agent behaviors.

In this approach, the human user directs evolution by selecting what is interesting from the on-screen population of behaviors. However, unlike in typical IEC, the user can then request that the next generation be filled with *novel* descendants, as opposed to only the direct descendants of typical IEC. The result of such an approach, unconstrained by a priori objectives, is that it traverses key stepping stones that ultimately accumulate meaningful domain knowledge.

To establishes this new evolutionary approach based on the serendipitous discovery of key stepping stones during evolution, this dissertation consists of four key contributions: (1) The first contribution establishes the deleterious effects of a priori objectives on evolution. The second (2) introduces the NA-IEC approach as an alternative to traditional objective-based approaches. The third (3) is a *proof-of-concept* that demonstrates how combining human insight with novelty search finds solutions significantly faster and at lower genomic complexities than fully-automated processes, including pure novelty search, suggesting an important role for human users in the search for solutions. Finally, (4) the NA-IEC approach is applied in a challenge domain wherein leveraging human intuition and domain knowledge accelerates the evolution of solutions for the nontrivial octopus-arm control task. The culmination of these contributions demonstrates the importance of incorporating human insights into simulated evolution as a means to discovering better solutions more rapidly than traditional approaches.

To my wife, Amanda and to my precious daughters, Taryn and Aubrey.

ACKNOWLEDGMENTS

Foremost, I would like to thank my advisor Dr. Kenneth O. Stanley, whose patience, guidance, and encouragement has been invaluable to me. Without him this dissertation would not exist. I also owe a special thanks to my other committee members, Dr. Peter A. Hancock, Dr. Charles E. Hughes, Dr. Avelino J. Gonzalez, Dr. Annie S. Wu, for enriching this dissertation with their insights and advice. To the members (past and present) of the Evolutionary Complexity Research Group (Eplex) at the University of Central Florida, thank you for sharing your individual perspectives and ideas so freely. Thanks to the Department of Electrical and Computer Engineering at the Air Force Institute of Technology (AFIT), and the Air Force as a whole[1], for the opportunity to pursue this endeavor. Finally, I want to express my profound gratitude to those around me who love me: To my wife, Amanda for her support and love, I could not have done this without you. To my daughters, Taryn and Aubrey who gave me both focus and balance. And to my friend Gary (and others) who gave me back my life when I had lost my way. This work belongs to all of those who have shaped me into who I am. But for the grace of God go I.

Brian G. Woolley, Capt, USAF

University of Central Florida

June 2012

---

[1] The views expressed in this article are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

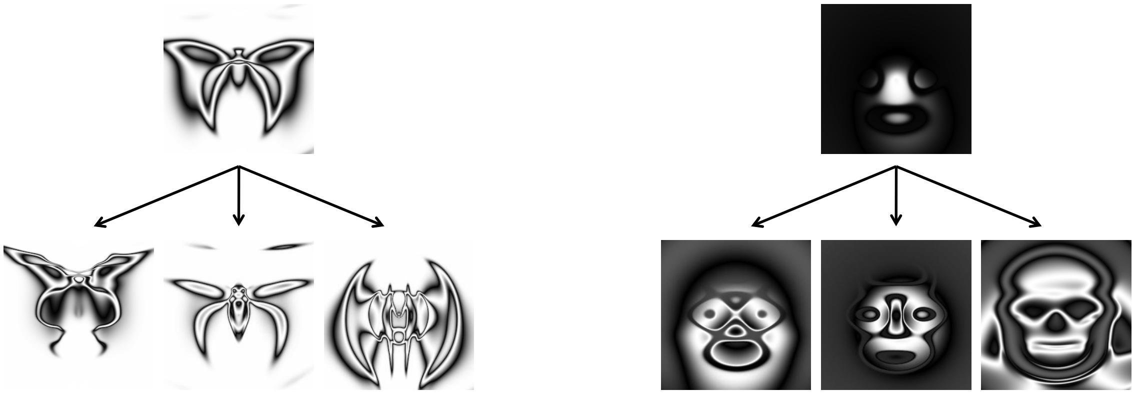
The field of evolutionary computation (EC), inspired by the products of natural evolution, promises to deliver solutions to complex design and control problems. Such evolutionary algorithms (EAs) traditionally find results by shaping a population of candidate solutions over generations based on their performance with respect to an objective. The capability of a particular EA is then evaluated for the speed (i.e. average number of evaluations) within which it finds solutions to benchmark problems (De Jong, 2002; Eiben and Smith, 2003; Fogel, 2006; Goldberg, 1989; Mitchell, 1997). New algorithms are lauded if they can consistently solve benchmarks faster than established methods. But what if this measurement of an algorithm's success is distracting us from the larger potential for discovery within EC?

Consider instead natural evolution, a system capable of discovering a rich and diverse population of powerful designs. Interestingly, in contrast to the usual measure of success in EC, nature does not repeatedly evolve the same organism over and over again in different lineages. Rather, natural evolution is a process with no final objective, and every organism, successful or unsuccessful, is a discovery on the road to nowhere in particular (Dawkins, 1986). Thus, if evolution succeeds in rich discoveries in the absence of any final objective, then how can EC emulate nature's never-ending ratcheting process to evolve its own meaningful artifacts? To address this question, this dissertation investigates the potential of abandoning the reigning objective-driven paradigm in favor of serendipitous discoveries made by humans. This approach is inspired in part by the success of the Picbreeder online service (Secretan et al., 2011, 2008), in which a community of users collaboratively evolve a diverse and growing phylogeny of meaningful images.

Picbreeder is an online system that allows users to interactively evolve images from *scratch* (i.e. from a set of random starter images) or by *branching* from other users' published images. Results from over 8,000 evolved Picbreeder images[1] suggest that the *stepping stones* that lead to major discoveries do not resemble or hint at the innovations to come. That is, users choose such stepping stones (i.e. intermediate images) as parents because they are interesting in their own right, rather than because they compare well to an a priori objective. In fact, new users are encouraged to select what they find *interesting* without an expectation of what they will find, and they often achieve appealing results in a relatively small number of generations (Secretan et al., 2011). Experience with Picbreeder shows how quickly meaningful images like a skull, a butterfly, and a car can be found, i.e. in only 74, 90, and 106 cumulative generations respectively. Furthermore, successful images evolved in Picbreeder, which employs the NeuroEvolution of Augmenting Topologies (NEAT) approach (Stanley and Miikkulainen, 2002, 2004) as its EA, exhibit good underlying representations. In particular, they are represented compactly and support continued elaboration upon established themes. For example, figure 1.1a illustrates how the image known as Butterfly sparked a family of winged creatures (i.e. not just more butterflies), while the Spooky Face, an entirely separate lineage, led to the family of faces shown in figure 1.1b. What is it about the

---

[1] As of May 2012 there were over 8,000 published images on the Picbreeder site.



(a) Winged Creatures **(b)** Family of Faces

Figure 1.1: Successful images have good representations. The collection of winged creatures in (a) emerged from elaborations on the Butterfly. Similarly, the family of faces in (b) emerged from the Spooky Face. These examples reflect the advantages of discovering a good underlying representation.

mode of evolution in Picbreeder that makes the serendipitous discovery of significant images common?

The significance of this question is explored by the first contribution in this dissertation, wherein the deleterious effects of objectives on search are demonstrated by an experiment in which images evolved in Picbreeder become the objective for the very same algorithm that discovered them. In this way, evolution can be run under the same parameters as existed for the original discovery, yet now toward a target image that is known to exist in the space of images. The results demonstrate that NEAT cannot re-evolve many of the very same images it once discovered. Even when the target image is simple enough to be reproduced, the effect is superficial because the underlying representation is a piecewise construction that is up to three times more complex than the original image. The paradox here is that although NEAT can discover meaningful images serendipitously, it *cannot* evolve the very same images when they are the explicit objective. Thus such images are only found effectively when they are *not* the objective.

The aim of this work is to translate the serendipitous phenomenon seen in Picbreeder to support the evolution of principled agent *behaviors* (i.e. not just pictures). The hope is that human evaluators will recognize what is *interesting* in a particular domain and that their selections (unconstrained by any final objective) will lead to *meaningful* behaviors. Additionally, the new evolutionary approach goes beyond simply putting a human in the loop. It also tracks what behaviors have already been evaluated and iteratively generates a collection of *novel* behaviors from the ones previously considered interesting by the user. By interleaving human-driven interactive evolution (Takagi, 2001) with such *novelty search* (Lehman and Stanley, 2008, 2011), two techniques which have never been combined before, a new approach called *novelty-assisted interactive evolutionary computation* (NA-IEC) improves evolution by presenting the human evaluator with a broad set of possible stepping stones, thus inspiring new ideas, and producing better behaviors in a small number of evaluations.

The main hypothesis is that interleaving novelty search with human selection improves evolutionary results over automated approaches. This hypothesis is initially validated by a *proof-of-concept* in which the NA-IEC approach is compared to the best current approach in the deceptive maze domain introduced by Lehman and Stanley (2008, 2011). The expectation is that human intuitions should direct the evolution of the maze navigation behavior more effectively than a completely automated search. In total there are four main contributions of this work: (1) It establishes the problem with objectives; (2) motivated by this problem, it introduces the NA-IEC approach; (3) NA-IEC is then validated by the proof-of-concept in the deceptive maze domain; and finally (4) in a challenge domain.

This fourth and final contribution delivers a *major application* in which kinematic controllers are evolved for a nontrivial task domain, i.e. a simulated octopus arm (Engel et al., 2006; Yekutieli et al., 2005). This culminating experiment extends previous work in an octopus arm domain in which arm controllers were evolved through objective-based evolution (Woolley and Stanley, 2010). While this preliminary work succeeded in evolving arm controllers, this experiment now explores whether leveraging human insights about the domain can evolve control behaviors faster than an approach that maximizes the fitness measure without achieving an intelligent representation of the environment (Gomez and Miikkulainen, 1997). By evolving controllers interactively with interleaved novelty search, i.e. with NA-IEC, the hope is that the human evaluator's ability to see what is interesting in the context of a particular task domain will direct the search toward meaningful behaviors and avoid the deleterious effects of pursuing objectives alone.

The next chapter discusses relevant aspects of EC, including traditional approaches for mitigating deception as well as modern approaches such as novelty search and interactive evolutionary systems like Picbreeder. Chapter 3 then presents an experiment that demonstrates the problem with objectives. Next, the NA-IEC approach is presented in detail (Chapter 4), followed by a proof-of-concept that evolves navigation behaviors for the deceptive maze domain by Lehman and Stanley (2008, 2011) in Chapter 5. An objective-based experiment with another task domain, the octopus arm domain, is introduced in Chapter 6. Then in Chapter 7, the NA-IEC approach is applied to the octopus arm domain as a challenge problem, the results of which confirm the importance of allowing human intuition to direct evolution. Finally, the dissertation concludes with a discussion of how non-objective and serendipitous approaches may impact EC as a whole (Chapter 8) and ends with the overall conclusions (Chapter 9).

CHAPTER 2
BACKGROUND

This chapter reviews foundational work in evolutionary computation that motivates and underlies the new evolutionary approach introduced in this dissertation, including traditional fitness-based approaches and non-objective methods like novelty search and interactive evolutionary computation.

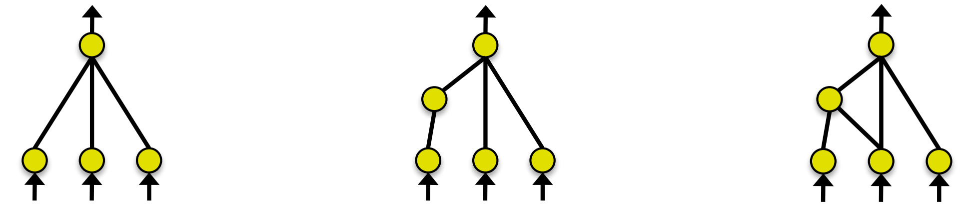
2.1 Evolutionary Computation

In evolutionary computation (EC), it is traditional to evolve the population iteratively by evaluating its members with respect to the objective (De Jong, 2002; Eiben and Smith, 2003; Fogel, 2006; Goldberg, 1989). While initial populations are often largely unfit with respect to the objective, some individuals are generally more fit than others. Inspired by the Darwinian principle of natural selection (Goldberg, 1989), the variation in the fitness of individuals provides a gradient to guide the search. In this way, EC explores multiple candidate solutions in parallel, thereby providing hope that it may avoid the trap of local optima by more fully exploring the search space.

Since its introduction, EC has developed into a number of branches and sub-communities that all follow this general framework, i.e. a cycle of evaluation, selection, and mutation is applied repeatedly to shape a population with respect to an objective (De Jong, 2002; Eiben and Smith, 2003). Some established branches include: *genetic algorithms* (Fogel, 1998; Goldberg, 1989; Holland, 1975), *genetic programming* (Koza, 1992, 1998), *evolutionary programing* (Saravanan and Fogel, 1995; Sinha et al., 2003; Yao et al., 1999), and *neuroevolution* (Gomez et al., 2006; Stanley and Miikkulainen, 2002, 2004). In each of these, the direction of the search, albeit stochastic, is governed by the evaluation and selection of individual candidates with regard to a fitness metric. As an example, to learn a strategy for playing chess, a co-evolutionary system may define fitness as the number of games won against other individuals in the current population (Mitchell, 1997). In this way, individuals that achieve significantly higher fitness scores become *more likely to be selected for reproduction.* Such an approach, which is ubiquitous in EC and even familiar in machine learning (Sutton and Barto, 1998), is predicated on the assumption that higher objective performance indicates approaching a solution.

Genetic algorithm (GA) approaches (Fogel, 1998; Goldberg, 1989; Holland, 1975) optimize strings of numbers that represent parameters in a problem domain. In a simple GA (Goldberg, 1989), the solution to be evolved is a string of binary values that represent the parameters of a black-box system. Evolution begins with an initial population of randomly generated bit-strings. The fitness of each individual is then determined by applying the bit-string within the black-box and evaluating the results with regard to the objective. Next, individual candidates are selected for reproduction. For example, they might be selected in a roulette-style fashion wherein the fitness determines the probability of selection. In this way, strings with higher fitness are more active in the mating pool and thus contribute more offspring to the next generation. Selected strings in the mating pool are ultimately paired stochastically and crossed over. Mutation operations based on random variation are also applied to provide additional diversity. The new population typically becomes the next generation and the process is repeated until a halting criterion is met. In this way, the optimal alleles may be discovered because the objective function rewards the discovery of partial solutions (exploitation) while stochastic selection and random mutations yield diversity (exploration).

A subset of EC, called *neuroevolution*, evolves the configuration of an artificial neural network (ANN) for a given task or environment (Yao et al., 1999). ANNs can be thought of as a little brain-like controllers that map sensory information from input nodes to output nodes, resulting in a particular behavior. ANNs may also contain hidden nodes and recurrent connections to express complex nonlinear behaviors. In many neuroevolution approaches the structure of the network is fixed and the connection weights are evolved to optimize the agent for a given objective (Gomez and Miikkulainen, 1999; Saravanan and Fogel, 1995). The particular approach to neuroevolution in the experiments in this dissertation is *NeuroEvolution of Augmenting Topologies* (NEAT), introduced by Stanley and Miikkulainen (2002, 2004). Unlike the fixed-structure approaches, the premise behind NEAT is that lower-dimensional networks can be optimized before attempting to optimize more complex networks. In this way, NEAT evolves both the connection weights and the ANN structure simultaneously by starting with a population of minimal topology networks (figure 2.1a) and then introduces



(a) Minimal Topology **(b)** Add Node Mutation **(c)** Add Connection Mutation

Figure 2.1: NEAT complexification operations. A key principle in the NEAT approach (Stanley and Miikkulainen, 2002, 2004) is that the topology of the network does not need to be known a priori. Rather, populations of networks increase in complexity from minimal topologies (a) through structural mutations that split existing connections to add new hidden nodes (b) or add connections between nodes that are not already linked (c).

additional structure during evolution, which is called *complexification*, through the *add node* and the *add connection* mutations (figure 2.1b and 2.1c, respectively). Additionally, when the topology of a network changes there is frequently a negative impact on fitness. To protect such structural innovations until they can be optimized, the NEAT approach partitions the population into groups of similar network topologies called*species* to ensure that only similar structures directly compete against each other for survival. Thus by evolving networks from minimal topologies and complexifying them gradually, NEAT does not require that the topology of the network be known a priori.

Other EC approaches follow the same basic evolutionary computing paradigm of selection and reproduction (De Jong, 2002; Eiben and Smith, 2003; Fogel, 2006; Gomez et al., 2006; Koza, 1992; Yao et al., 1999). Thus, while the stochastic elements of EC (i.e. selection and mutation) provide the ability to escape local optima and the population of individuals allows multiple candidate solutions to be explored in parallel, the general approach to EC remains a directed search that is subject to *deception*, which means that the gradient of increasing fitness may not ultimately lead to the global optimum. The next section discusses approaches to mitigating such deception.

2.2 Approaches to Mitigating Deception

Formal research into deception explores what causes evolutionary algorithms (EAs) to fail and how to mitigate such failures (Goldberg, 1989; Whitley, 1991). For the purpose of this work, we are interested in the case in which pursuing what appears to be a reasonable *objective* produces an unreasonable *objective function*. Thus an intuitive definition of deception, as stated by Lehman and Stanley (2011), is: "A deceptive objective function will *deceive* search by actively pointing the wrong way."

The fitness function can point the wrong way because not only must the fitness function reward the objective, but it must also reward the intermediate solutions (i.e. the stepping stones) that lead to the objective. One example of such deception is in the mountain car task (Boyan and Moore, 1995; Sutton and Barto, 1998), in which a car must drive up a steep slope to the goal (figure 2.2). This domain is deceptive by design because the car is underpowered and cannot reach the goal by simply driving forward. Rather, the controller must generate additional momentum by first accelerating backward up the hill behind it. Without such prior knowledge we might assign fitness based on reducing the distance to the

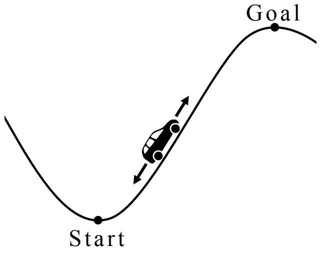


Figure 2.2: Mountain car. This deceptive task requires the controller first to accelerate backward (i.e. become more unfit) before attempting to climb the hill toward the goal (Boyan and Moore, 1995; Sutton and Barto, 1998).

goal, yielding an objective function that prunes out of the search the intermediate solutions (i.e. those that accelerate backwards) needed to reach the global objective.

While there will never be one best method for conquering deception (Wolpert and Macready, 1995), several approaches have been introduced to mitigate it. The most common approaches focus on avoiding the premature convergence of the population to a single solution or type of solution. *Speciation* techniques, inspired by the niching of organisms in nature, are the most common such approach to diversity maintenance (Goldberg and Richardson, 1987; Mahfoud, 1995; Ryan, 1994; Stanley and Miikkulainen, 2004). By creating subpopulations and applying e.g. *explicit fitness sharing* (Goldberg and Richardson, 1987), individuals only compete within their local niche, i.e. species. Speciation has been shown to to protect innovation (Sigrist and Sommer, 1999) by ensuring that "highly fit species cannot crowd smaller species out of the population before they have a chance to reach their potential" (Stanley and Miikkulainen, 2002).

In cases in which the objective is too complex to approach directly, *task decomposition* (Colombetti and Dorigo, 1992; Lin, 1993; Perkins and Hayes, 1996; Singh, 1992) or *incremental evolution* (Elman, 1991; Gomez and Miikkulainen, 1997; Knowles et al., 2001; Mahfoud, 1995; van de Panne and Lamouret, 1995) are often applied to reward the evolution of intermediate goals and make learning tractable. In task decomposition the complex task is broken into subtasks that can be learned separately and then combined to achieve the final solution; in incremental evolution a single system learns through a succession of tasks that are increasingly demanding. For example, Gomez and Miikkulainen applied *delta coding* (Mathias and Whitley, 1993; Whitley et al., 1991) to evolve complex behaviors for the predator-prey (Gomez and Miikkulainen, 1997) and double-pole balancing (Gomez and Miikkulainen, 1999) domains by starting from simple tasks and gradually making the task more challenging. The trick is that following a local search, as in delta coding, requires knowledge of the path to the final goal.

Another approach to mitigating deception is to score individuals in the population over multiple and possibly conflicting objectives through multi-objective evolutionary algorithms (MOEAs; Coello et al., 2007; Deb, 2001). In this way, unique solutions are more distinctly identifiable because they optimize the domain objectives to different degrees and thus exist at different points in the search space. Such an approach is desirable when a monolithic fitness function could make different solutions appear equivalent when in fact they should not compete directly. Rather, by evaluating how solutions score over the set of objectives, a population of candidate solutions emerges that have at least one exemplary quality, i.e. they are on the Parato front (Deb et al., 2002; Horn et al., 1994; Veldhuizen and Lamont, 2000). In this way, two solutions that approach the problem differently are not compared directly. This insight was demonstrated by Knowles et al. (2001) wherein a single-objective optimization problem was decomposed into a multi-objective optimization problem and resulted in a less deceptive search space. Despite this advantage, MOEAs are still vulnerable to deception (Ando and Suzuki, 2006; Brockhoff et al., 2007; Deb, 1999).

Despite extensive research, deception remains a significant problem in the field of EC (Goldberg and Richardson, 1987; Liepins and Vose, 1990; Pelikan and Goldberg, 2001). The problem is that EAs ultimately respond to the selection pressures created by the fitness function. The challenge is to determine how to reward the intermediate steps that are required to reach the goal. Since search landscapes are induced by objective (e.g. fitness) functions, what appears to be a reasonable heuristic may actually *prevent* the objective from being reached because the objective function does not reward the *stepping stones* in the search space that ultimately lead to the desired objective. Thus any similarity metric that guides the search toward an objective is potentially a false compass to the optimal solution (Stanley, 2010).

In this spirit, Lehman and Stanley (2008, 2011) introduced the idea of abandoning objectives as a search heuristic in deceptive domains, electing instead to reward individuals only for novel behaviors. The next section reviews novelty search and some domains where it has been applied successfully.

2.3 Novelty Search

A fundamental dilemma with traditional approaches to EC is that crafting an effective fitness function is akin to understanding the fitness landscape or knowing the stepping stones a priori (Ficici and Pollack, 1998; Zaera et al., 1996). Such a requirement becomes increasingly difficult as objectives become more ambitious because the intermediate steps to the solution are less likely to be known (Ficici and Pollack, 1998). As an alternative, Lehman and Stanley (2008, 2011) demonstrated that searching without regard to the objective, i.e. searching only for novel behavior, is more effective at discovering solutions in some deceptive domains than rewarding objective performance.

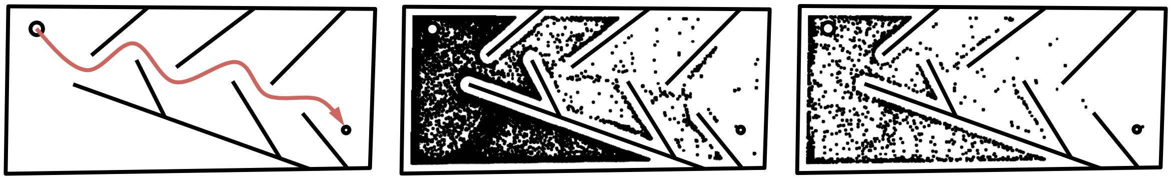
Novelty search works with EAs by replacing the fitness function with a *novelty metric*. The novelty metric is a measure of the uniqueness of an individual's behavior at a given task. Instead of rewarding performance, novelty search rewards individuals in the population for finding new ways to complete the evaluation task, thus creating a constant pressure to do something new (Lehman and Stanley, 2011).

Because novelty search operates in *behavior space*, it is important first to characterize the space of unique behaviors in a way that is meaningful to the domain. The novelty search algorithm then computes the *sparseness* in the behavior space as the average distance to the k-nearest neighbors (Cover and Thomas, 1991) around that behavior. The sparseness ρ of behavior x is given by

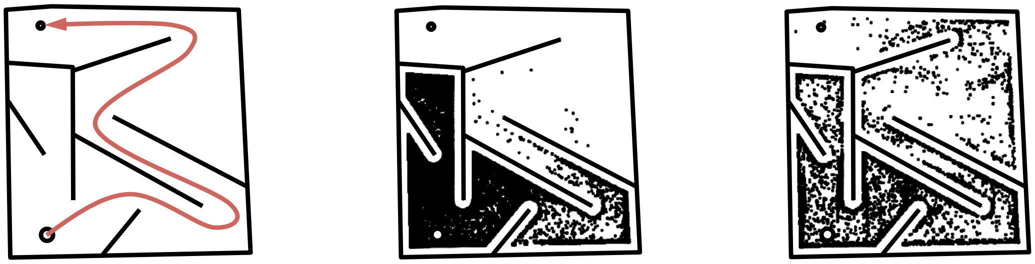
$$\rho(x) = \frac{1}{k}\sum_{i=0}^{k} \mathrm{dist}(x, \mu_i), \tag{2.1}$$

where μ_i is the ith-nearest neighbor of x with respect to the distance function $\mathrm{dist}(x, \mu)$. In this way, if the average distance is large, then the candidate solution is considered to be in a sparse area of the behavior space, thus making it more likely to be selected by the EA. Optionally, as in coevolution (De Jong, 2004), an archive of past behaviors may serve to avoid backtracking through the behavior space. If the novelty metric is sufficiently high for a new individual (i.e. above some minimal threshold $\rho_{\min}$), then the individual may be recorded in the permanent archive to provide a comprehensive sample of where the search has been, thereby increasing the pressure to discover new ways of behaving in the domain (Lehman and Stanley, 2008, 2011).

Characterizing behaviors so that they can be compared is the most challenging aspect of novelty search. In the deceptive maze experiment introduced by Lehman and Stanley (2008), which is an experimental domain in this dissertation, the behavior of a maze navigation robot was defined as its *final position*. In this way, the novelty metric rewards controllers that end at new locations in the maze. At first, the collection of behaviors may include robots that do nothing, get stuck in corners, run in circles, and so forth. However, at some point, the collection of simple behaviors becomes saturated and the pressure to do something new increases, i.e. evolution favors mutations that take the navigator to new places in the maze.



(a) Medium Map **(b)** Objective-based **(c)** Novelty Search



(d) Hard Map **(e)** Objective-based **(f)** Novelty Search

Figure 2.3: Deceptive maze experiment (Lehman and Stanley, 2008, 2011). The task is to evolve controller ANNs that navigate a particular map, e.g. the medium map (a) and the hard map (d). The cul-de-sacs, which are not known a priori, represent local optima in the search space. The final position of each robot in a typical run is marked as a point on the map in (b), (c), (e), and (f). The results in (b) and (e) show how the EA guided by fitness is attracted to and deceived by the cul-de-sacs in each map; the results in (c) and (f) show how the very same EA guided by novelty is more evenly distributed.

Figure 2.3 depicts behaviors (represented as the final point visited by an individual) discovered during typical runs conducted in the Medium Map (figure 2.3a) and the Hard Map (figure 2.3d) from Lehman and Stanley (2008, 2011). From these illustrations it is clear that the objective-based search, which rewards reducing the distance to the goal, finds the the cul-de-sacs in both maps deceiving (figures 2.3b and 2.3e). In contrast, the results from novelty search (figures 2.3c and 2.3f), which rewards ending somewhere new, are more evenly distributed because they are not deceived (Lehman and Stanley, 2008, 2011).

Novelty search was also successfully applied to evolving controllers for a simulated three-dimensional biped robot. While the objective was to walk as far as possible within a given time limit, novelty search found solutions that were qualitatively and quantitatively better by searching the space of biped gaits (i.e. measured as the biped's center of mass at one-second intervals) without regard to the objective (Lehman and Stanley, 2011).

Another deceptive domain where novelty search was shown to be effective is the artificial ant problem (Koza, 1992). In this domain, a simulated ant, driven by a genetic program (GP) tree, follows a trail in an attempt to collect as much food as possible. The gaps in the trail require the agent to *infer* what to do when there are missing steps in the trail. In this domain the ant behaviors are defined by the amount of food collected at evenly spaced times during an evaluation, thus allowing novelty search to differentiate behaviors that ultimately collect the same amount of food by different means (Lehman and Stanley, 2010a).

When applied in the context of adaptive neural networks, i.e. neuromodulated plasticity (Soltoggio, 2008), novelty search escapes the deceptive trap of learning to learn (Risi et al., 2011, 2009). In the T-maze domain (Blynel and Floreano, 2003; Soltoggio et al., 2008), the behavior characterization is based on the amount of reward collected (i.e. *high*, *low*, or *none*) and whether or not the agent crashed in each of 200 trials. In cases where learning was required at a few key points during a trial, novelty search achieved adaptive solutions despite the high-dimensional behavior space, while traditional fitness-based approaches evolved static solutions that satisfied the majority of situations during a trial.

In cases where abandoning the objective completely may be unnecessarily radical, novelty has been applied as an objective in a Pareto-based multi-objective evolutionary algorithm (Coello et al., 2007; Deb, 2001). The idea behind such an approach is that while efforts to optimize fitness (i.e. the traditional objective) are deceived by a multi-modal landscape, the ability to pursue novelty can direct the search into new areas that can attain higher fitness (Graening et al., 2010; Mouret, 2011).

While the idea of selecting anything novel may appear to be an undesirable exhaustive search, searching in the space of *behaviors* is often tractable because many points in the space of possible *genomes* collapse to a single behavior. Furthermore, when applied in conjunction with complexifying algorithms like NEAT (Lehman and Stanley, 2008, 2010b) and GP (Lehman and Stanley, 2010a), simple behaviors become associated with minimal representations, and only mutations that increase the size of the genome and lead to novel behaviors are explored further. Therefore, this approach, operating without regard to an objective, moves into complex spaces in a meaningful way because new behaviors are those that could not be expressed at lower levels of complexity (Ventrella, 1994a, 1995), i.e. complexity is rewarded when it is warranted.

Experimental results with novelty search demonstrate that it often finds solutions in deceptive domains more frequently, more quickly, and at lower genomic complexities than an equivalent objective-based approach. However, experience has also shown that novelty search becomes lost in unrestricted domains (Lehman and Stanley, 2010b). In such domains there is an opportunity to leverage human knowledge rather than exhaustively exploring the space of all possible solutions. For example, in the space of all possible images, humans recognize the importance of symmetry in pictures and are able to relate structural innovations with objects in the real world. Thus the next section provides relevant background on the field of human-led evolution, followed by a description of Picbreeder, a domain in which a community of users interactively evolves a collection of meaningful images without having a formal, unified objective.

2.4 Interactive Evolutionary Computation

In *interactive evolutionary computation* (IEC) the traditional objective-function is replaced by a user who performs selection (Takagi, 2001). IEC is effective in creative domains (Romero and Machado, 2008), including the evolution of images (Hart, 2007; Machado and Cardoso, 2002; Sims, 1993; World, 1996), animations (Draves, 2005; Sims, 1997; Ventrella, 1994b, 1995), structures (Dawkins, 1989; Sims, 1991; Smith, 1991), and music (Biles, 1994; Hoover et al., 2011; Johanson and Poli, 1998; Nelson, 1993; Tokui and Iba, 2000). In such creative domains the term *fitness* is subjective because what people experience as pleasing or interesting is based on individual preferences. Thus when what is good, bad, meaningful, or strange is too broad and complex to encode into a traditional objective function, interactive evolution can provide a means for making significant discoveries in evolutionary systems.

Like traditional EAs, IEC systems also typically begin from a random initial population that evolves over generations by selecting, mating and mutating members. However, IEC differs from traditional automated EAs in that a human user is now responsible for the evaluation and selection of promising candidate solutions. While this change typically leads to smaller population sizes and higher mutation rates, the most profound implication is that evolution is no longer bound to a rigid expression of what is fit and unfit. In fact, the human evaluator's breadth of experience makes it likely that his or her selection criteria will change over the course of evolution. Such an ability to make *serendipitous discoveries*, i.e. identify and pursue important artifacts as they emerge, is the primary motivation for the new evolutionary approach introduced in this dissertation.

To interface with the human evaluator, the majority of IEC systems are modeled after the original Blind Watchmaker *Biomorphs* application by Dawkins (1986, 1989). In this approach the user is presented with a panel of individuals (e.g. 3×4) from which the parents of the next generation are selected. The IEC system then mates, recombines, and mutates the genetic material of the parents to create the next generation, which is then presented to the user. This process is repeated at the user's direction until the user is satisfied.

The results of Dawkins' original nine-gene Biomorphs (Dawkins, 1986) demonstrate that selecting for phenotypic effects, i.e. selecting for how genes are expressed, leads to meaningful discoveries. This work was furthered by Sims (1991), who interactively evolved variable-length expressions for lifelike three-dimensional plant structures. In both cases, the insight is that knowing the underlying genetic encoding is not important. Rather, selecting meaningful phenotype attributes that emerge during evolution does lead to significant results in the vast space of what is possible.

Despite the benefits of having a human in the loop, such IEC systems are limited by user fatigue. According to Takagi (2001), typical IEC processes only last 10–20 generations per session. The problem is that the vast majority of significant discoveries exist beyond the reach of a single-user session. One response, which has become known as *collaborative interactive evolution* (CIE; Szumlanski et al., 2006), is to leverage the efforts of many users.

Sims addressed the issue of user fatigue in the Genetic Images (Sims, 1993) exhibit and the Galapagos (Sims, 1997) exhibit by allowing museum patrons to make selections. In this way, evolution was able to aggregate the evaluations of many visitors. Since the proliferation of the world-wide-web, online CIE systems like Pfeiffer (Langdon, 2005) and Electric Sheep (Draves, 2005) have emerged to allow users to vote remotely for their favorite artifacts. In each of these systems, individual candidate that receive votes are allowed to to stay in the breeding population longer, i.e. votes are a form of subjective fitness. However, such systems rely on *consensus*, which often only achieves mediocre results because users express competing preferences. In contrast, one CIE system that avoids such aggregation, and which inspired the approach presented later, is the Picbreeder project (Secretan et al., 2011, 2008), introduced next.

2.5 Picbreeder: Collaborative Interactive Evolution of Images

Another effectively non-objective search is in *Picbreeder* (`http://picbreeder.org`) (Secretan et al., 2011, 2008), which is studied in the next chapter to evaluate how representations are affected by objective and non-objective evolutionary searches, and is revisited later as the inspiration for the new evolutionary approach introduced by this work. Picbreeder is a distributed community of users that interactively evolve pictures by selecting images that are appealing. Picbreeder is a CIE system because users on Picbreeder *collaborate* by continuing to evolve images previously evolved by other users. The collection of images generated by Picbreeder is significant because it demonstrates how a group of individuals working without a formal unified objective can discover *attractive* and *interesting* areas in the vast desert of all possible images; some such images are shown in figure 2.4. Additionally, the quality of such a serendipitous approach to evolution is evident in the diverse phylogeny of images that have emerged, the compactness of their representations, and the speed (i.e. low number of generations) with which meaningful images are discovered. A crucial aspect of this result, for the purposes of this dissertation, is that the system *as a whole* has no unified a priori objective or objectives. While individual users may sometimes (and sometimes not) arrive with their own objectives, the combination of all users branching off each other is not working towards any unified objective in particular. This fact will turn out critical to the representations that ultimately evolve.

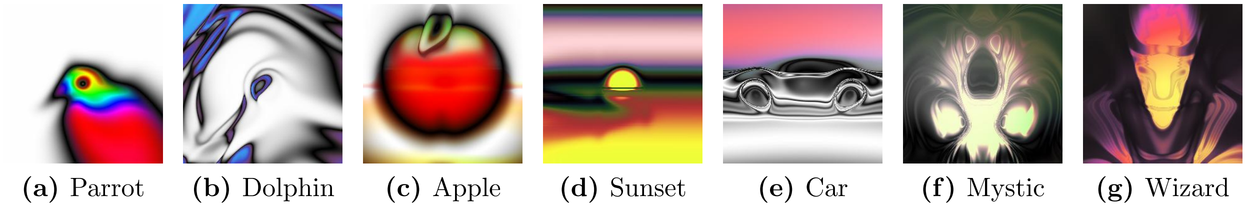
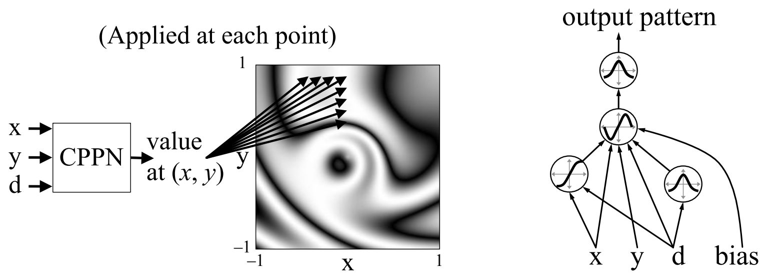

(a) Parrot (b) Dolphin (c) Apple (d) Sunset (e) Car (f) Mystic (g) Wizard

Figure 2.4: Images evolved on Picbreeder. These images were interactively evolved by a community of human users with no explicit objective. They demonstrate the system's ability to discover interesting and meaningful images.

Users evolve images in Picbreeder by selecting ones that appeal to them from among a set of 15 candidates to produce a new generation. As this process is repeated, the individual images in the population evolve to satisfy the user. Once satisfied, the user can *publish* his or her image to the Picbreeder site. Sharing their work with the community then allows others to continue evolving already-published images to form new and more intricate designs (Secretan et al., 2008), which is called *branching*.

Each image in Picbreeder is indirectly encoded by a variant of a neural network called a *compositional pattern-producing network* (CPPN; Stanley, 2007). The idea behind CPPNs is that geometric patterns can be encoded by a *composition of functions* that are chosen to represent common regularities. For example, composing the Gaussian function, which is symmetric, with any other function results in a symmetric pattern. The internal structure of a CPPN is represented as a weighted network, similar to an ANN, that denotes which functions are composed and in what order. The appeal of this encoding is that it can represent a pattern with regularities such as symmetry, repetition, and repetition with variation



(a) Mapping **(b)** Composition

Figure 2.5: CPPN encoding. (a) The CPPN takes arguments x and y, which are coordinates in a two-dimensional space, and d, the distance from (x, y) to the image center. When all the coordinates are drawn with an intensity corresponding to the output of the CPPN, the result is a spatial pattern, which can be viewed as a phenotype whose genotype is the CPPN. (b) Internally, the CPPN is a graph that determines which functions are connected. As in an ANN, the connections are weighted such that the output of a function is multiplied by the weight of its outgoing connection. The CPPN in (b) actually produces the pattern in (a).

through a network of simple functions that can be evolved by existing methods for evolving ANNs.

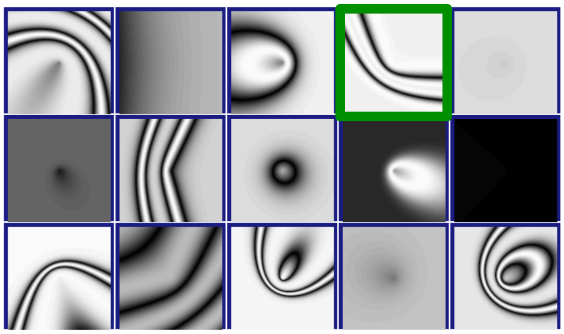
Images are rendered from CPPNs in Picbreeder by querying the network as the function CPPN(x, y, d) to obtain the grayscale value of the pixel located at (x, y) in the image (figure 2.5). The extra input d is the *distance from the center* to the (x, y) location being queried, which gives the CPPN a sense of radial symmetry that provides a bias towards appealing images (Secretan et al., 2011). Because CPPNs are a composition of continuous functions in a geometric space they provide a compact representation of the image at any resolution.

Picbreeder evolves images by evolving the underlying CPPN as if it were an ANN. That way, the NEAT approach (Stanley and Miikkulainen, 2002, 2004) can facilitate the evolutionary step between generations. Following the idea that the network structure can be evolved through complexification, Picbreeder begins with an initial population of images that are simple patterns represented by networks with just five connections and one hidden node. As the underlying networks add complexity (i.e. new nodes and connections), features and structures emerge in the resulting images that could not be expressed by the simpler CPPNs. When interesting or meaningful changes occur, the user selects the images that will reproduce to form the next generation. In this way, users can rapidly move from simple patterns towards a higher-dimensional space by rewarding regularities and structures that have meaning to them.

Figure 2.6 illustrates how images are interactively evolved in Picbreeder. From an initial population (figure 2.6a), the human user can select one or more images to become the parent(s) for the next generation; the example in figure 2.6 is based on selecting one image. Because Picbreeder employs NEAT as its underlying EA, mutations that add structure to the CPPN may also occur, thus allowing the resulting image to increase the complexity of its expressed structure. One such mutation is clearly visible in figure 2.6b, and selecting it yields the population in figure 2.6c. This process of evaluation, selection, and mutation continues at the will of the user. To overcome the potential for fatigue in IEC applications (Takagi, 2001), Picbreeder allows the user to save, publish, or abandon evolution at any time. Published images then become available for other users to continue evolving at the Picbreeder website.

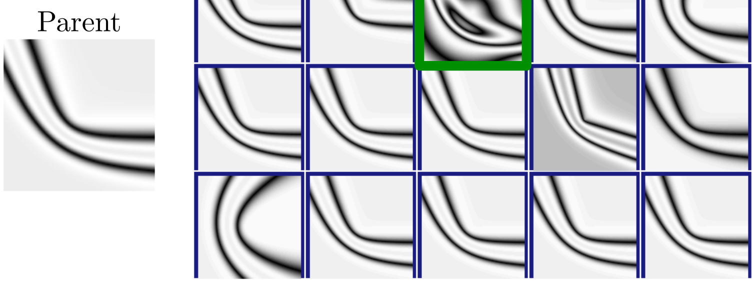
The significance of this example is that every image in Picbreeder ultimately derives from an initial population like the one in figure 2.6a.

In this way, interactive evolution can discover meaningful images in the space of all possible images. Additionally, the number and diversity of compact image representations discovered in Picbreeder suggests that evolving images based on what humans see as interesting actually encodes meaningful structure that in turn leads to serendipitous discoveries. Exploring this process more deeply, the next chapter demonstrates how the mode of evolution affects the representation of the solution and argues for the potential power of serendipitous discovery in evolutionary computation.



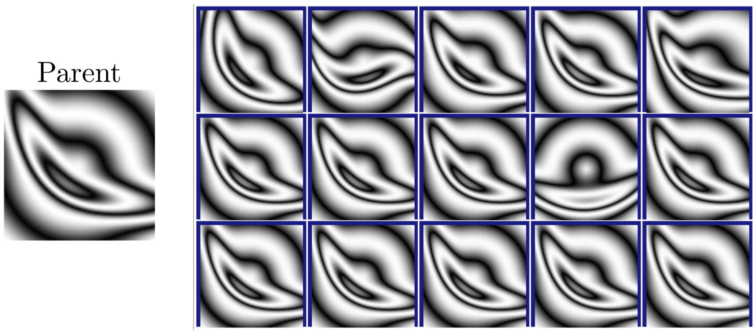
Generation 1:
Randomly-generated images from minimal CPPN topologies.

(a) Users make selections from an initial population of images



Generation 2:
Mutations that add genes cause images to gain structure.

(b) Subsequent populations are based on the parent image(s)



Generation 3:
As *meaningful* structures emerge, interesting images are discovered.

(c) The process is repeated at the user's direction

Figure 2.6: Breeding images from scratch. The sequence illustrates the interactive evolution of images in Picbreeder (Secretan et al., 2011, 2008). (a) Starting from *scratch* creates an initial population of random images from minimal CPPN topologies. Images selected by the user (highlighted) become the parent(s) of the next generation. (b) The new population is created by mutating the parent (highlighted and shown at left), which may yield additional structure in the image. (c) This process of evaluation, selection, and mutation repeats until the user either publishes an image to the Picbreeder website for others to continue its evolution, saves the current evolution, or abandons the effort. This sequence is significant because every image in Picbreeder (e.g. the images in figure 2.4) ultimately began in this way, i.e. from scratch.

CHAPTER 3
THE PROBLEM WITH OBJECTIVES

Evolutionary algorithms (EAs) are often tested on benchmarks to assess their ability to reach a particular objective in the search space. Popular types of benchmarks include optimization (De Jong, 2002), function approximation (Whiteson and Stone, 2006), and control (such as in neuroevolution; Floreano et al., 2008). Reinforcing the field's focus on objective-targeted optimization, theoretical analyses often focus on the probability that EAs will converge to the objective (Droste et al., 2002; Eiben and Smith, 2003; Mühlenbein and Manning, 1999). A fundamental assumption behind this objective-focused paradigm in evolutionary computation (EC) is that the value of EAs lies in their ability to reach objectives that we set for them *a priori*. Paired with this assumption is the idea that when an evolutionary algorithm *consistently fails* to reach its intended objective, then it is not suited for the task. Yet what if this central assumption is wrong? What if consistently reaching the objective could actually obfuscate a deeper underlying pathology, while consistently failing to reach the objective ultimately bears little on the ability of the EA to produce impressive results in general? While perhaps paradoxical, these questions highlight a delicate uncertainty in EC about its relationship to natural evolution, wherein many remarkable phenotypes were discovered even though none of them were explicitly expressed as a priori objectives for the process. Is it possible that we are judging our algorithms wrongly?

To gain a fresh perspective on this question, this chapter, which is based on work published in the Genetic and Evolutionary Computation Conference, GECCO'11 (Woolley and Stanley, 2011), takes an unusual experimental approach: The objective of the EA is to *re-evolve* images that were *already* evolved in Picbreeder (Secretan et al., 2011, 2008). Such an experimental approach yields a unique perspective because these images were not specified as objectives when Picbreeder was first introduced, but in this work they *become* objectives for the very same algorithm and representation already inside Picbreeder. That way, even if the EA fails to reproduce its own original results, the conclusion that the algorithm or representation is insufficient to produce the objective is *precluded*, because the algorithm *did* already produce the objective, only it was originally discovered serendipitously before it was chosen as a benchmark objective. Through this experiment, we will discover that just the act of setting an objective triggers a chain of unintended consequences that confound the usual conclusions drawn from such benchmarks.

In particular, the main result disclosed in in this chapter is that the algorithm inside Picbreeder, *NeuroEvolution of Augmenting Topologies* (NEAT; Stanley and Miikkulainen, 2002, 2004) and the representation inside Picbreeder, *compositional pattern-producing networks* (CPPNs; Stanley, 2007), consistently fail to reproduce the results that they already produced. For the simpler images, the failure is in the quality of the solution, yielding much larger representations than necessary; for the more complex images, the failure is in obtaining the solution at all.

However, the most revealing focus of the analysis is on *why* these failures occur so universally. Because NEAT in fact already discovered the targets, we are forced towards deeper insight into the phenomenon of failure than simply blaming NEAT, as would be customary. In fact, it turns out that it is the very practice of making the images objectives that ultimately causes them not to be rediscovered properly. Not only does evaluating fitness against an objective potentially push it in the wrong direction through deception, but even when it still manages to right itself, the result is often a destructive effect on *representation* that has received little attention before this study.

This insight is sobering because the vast majority of empirical experiments in EC assess their results based on benchmarks with a priori objectives (De Jong, 2002; Goldberg, 1989; Koza, 1992; Saravanan and Fogel, 1995; Stanley and Miikkulainen, 2002; Whiteson and Stone, 2006) and theory often focuses on whether targets can be reached (Droste et al., 2002; Mühlenbein and Manning, 1999). Thus the major implication is that a change in thinking may be warranted about *how EAs should be judged*. Furthermore, on the positive side, the analysis hints at the kinds of situations that ultimately encourage efficient and elegant representations to evolve. As the discussion so far implies, these ideal situations are precisely when there is no specified objective (as in Picbreeder), suggesting that the way we traditionally use these algorithms may not be allowing them to exhibit their full potential. Thus this chapter aims to establish the importance of abandoning the pursuit of specific objectives, which is the reigning paradigm in EC, and instead motivates interactive approaches wherein the human intuitions direct evolution toward meaningful discoveries that were not known to exist a priori, i.e. serendipitous search.

3.1 Motivation

While the idea of searching without an objective may at first seem unusual, even a brief experience with Picbreeder shows that a serendipitous excursion through interesting parts of the search space does not require any specific objective. On the way, stepping stones and key innovations that appear interesting in their own right are discovered that can be elaborated later (perhaps by another user) to reach ambitious ends, even when the intermediate steps do not resemble the ultimate destination.

Furthermore, practical experience with Picbreeder has shown the futility of setting out to evolve a particular type of image, e.g. a flower or a butterfly (Secretan et al., 2011, 2008). In contrast, novice users that are encouraged to evolve patterns without an a priori expectation frequently find appealing images within 10 to 30 generations (often after branching from images evolved by other users that do not resemble the newer result). It turns out that evolving images in this way works because it tends to reward structures that become stepping stones to other meaningful images even if the stepping stones do not resemble their descendants. An example of this phenomenon is the lineage of images in figure 3.1 that ultimately led to the Skull, a seminal Picbreeder image. Its predecessors, which resemble e.g. a crescent,

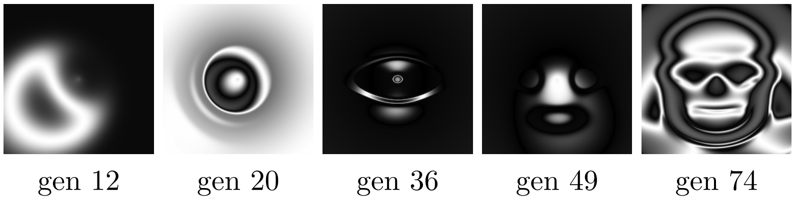


Figure 3.1: Stepping stones to the Skull. This linage illustrates the progression of seemingly unrelated images that led to a seminal Picbreeder image in just 74 generations. As a metaphor for solving ambitious problems, such an example demonstrates the importance of preserving stepping stones even if they do not resemble an a priori objective.

a donut, and a dish, do not hint at the significant discovery to come, yet were nevertheless essential to reaching it. Users involved in this lineage thus selected these images *for their own appeal* rather than because they were searching explicitly for a skull. As a metaphor for finding solutions to ambitious problems, the value of such important intermediate steps often cannot be known in objective-driven search when they are first discovered because essential innovations do not necessarily resemble the given objective.

The hypothesis of the experiment in this chapter is that an objective-based search will construct a piecewise solution that fails to embody key regularities in the problem domain because the fitness function only rewards incremental improvements that *resemble* the objective, which is highly restrictive. By favoring short-term gains rather than developing a broad variety of innovations, early decisions about how to construct the underlying representation of solutions will fail to capture the key organizational concepts in the problem domain.

The experiment described next makes these considerations concrete by showing in practice how setting an objective distorts the search and corrupts the representation, even with the very same algorithm and representation with which the targets were originally discovered. The key lesson will be that *how* something is discovered matters, and discovery through objective optimization is often the wrong way to do it. This lesson is the key motivation for a new evolutionary approach, wherein human intuitions will direct the search in the hope of discovering solutions that have good underlying representations.

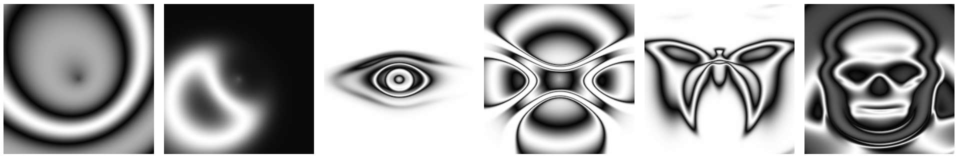
3.2 Image Evolution Experiment

The aim of this experiment is to investigate how pursuing a singular objective impacts the underlying representation of the solution. To achieve this aim, a population of image-generating CPPNs (i.e. with inputs x, y, and d, as in Picbreeder) is evolved towards one of the six target images (which span a range of different complexities) in figure 3.2[1]. It is important to note that each target image was originally evolved by a human user or chain of users on Picbreeder. Thus the task of evolving CPPNs toward these objectives should be feasible because the targets are *known* to exist in the space of solutions and were previously evolved.

To ensure that a direct comparison can be made between the representation of solutions evolved serendipitously and those evolved as objectives, the automated evolution described in this section has the same operational parameters as Picbreeder wherever possible. The

---

[1] While some Picbreeder CPPNs also generate color images, the image chosen as targets are all grayscale.



(a) Simple **(b)** Crescent **(c)** Eye **(d)** Warp **(e)** Butterfly **(f)** Skull

Figure 3.2: Target images. Each image was originally evolved by human users in Picbreeder. The difficulty of re-evolving toward these images with the same algorithm and representation demonstrates that *how* something is discovered matters.

key difference is that in interactive evolution, a human user selects the individuals that will become the parents of the next generation. Thus this experiment adds an evaluation and selection process to replace the human evaluator.

To automate the selection process, the fitness of each evolved candidate image is calculated based on how closely it matches the target image. In effect, the target images become the objective and thereby serve as metaphors for objectives of different levels of complexity. To compare two images and calculate fitness, each is defined by a feature set that includes the grayscale *and* gradient value for each pixel. The candidate image, scaled to match the normalized target image, is then compared by calculating the *degree of difference* (Ralescu, 2003) between corresponding features. The degree of difference, $d(c, t)$, between a particular candidate feature c and the corresponding target feature t is defined as

$$d(c, t) = 1 - e^{-\alpha|c-t|}, \tag{3.1}$$

where $\alpha = 5$ is a modulation parameter. From this equation, images can be described by their feature sets, wherein elements in the candidate feature set, $C = \{c_1, \ldots, c_n\}$, can be compared against the target feature set, $T = \{t_1, \ldots, t_n\}$. Thus the error between the candidate and the target feature sets, $\mathrm{err}(C, T)$, is calculated as

$$\mathrm{err}(C, T) = \frac{1}{n} \sum_{i=1}^{n} d(c_i, t_i), \tag{3.2}$$

which is the average degree of difference across feature sets C and T, where n is the total number of features. Finally, the fitness of the candidate, $f(C)$, is assigned as

$$f(C) = 1 - \mathrm{err}(C, T)^2, \tag{3.3}$$

such that taking the mean-squared error (MSE) generalizes large errors and emphasizes the importance of details.

Inevitably the fitness function in this experiment will be scrutinized for its effectiveness. Some will ask whether a better such function for comparing images could have been chosen. However, it will turn out that ultimately the problem is that no image comparison, no matter how good, can really reward stepping stone images because stepping stones to the target do not resemble the target itself. Thus the problem here is significantly deeper than simply finding a better image-matching heuristic. Nevertheless, to validate the fitness function as a reasonable search heuristic, a preliminary experiment evolves a population of CPPNs towards a randomly-generated image with five connections and one hidden node, i.e. the Simple target image in figure 3.2a. The validation experiment differs from the main experiment in that the correct solution topology for the CPPN is given (i.e. it is taken from the previously-evolved target), and only the connection weights must be evolved (unlike in normal NEAT wherein both weights and structure evolve together). Thus evolution must match the target image by tuning the five connection weights. The results of this preliminary validation of the fitness function showed that the solution threshold (explained below) was reached by 19 of 20 runs in 353 ± 687 (median = 135) generations on average. In this way, the validation experiment demonstrates that the fitness function is a reasonable method for comparing images in the population to the target image.

3.3 Experimental Results

The key question is what happens when NEAT attempts to re-evolve images that were already evolved by Picbreeder users. Such a question is relevant to the need for a new evolutionary approach because it demonstrates the deleterious effects of setting objectives. For consistency, a run is considered *successful* if the fitness score is greater than 0.75 (out of 1.0 maximum). This threshold corresponds to an average error of 5.75% between the source feature set and the target feature set and was verified as reasonable by a qualitative review of the resulting images above this threshold.

To show the contrast between images re-evolved as objectives as opposed to discoveries made on Picbreeder, table 3.1 compares the objective-based results against the statistics of the target images when they were originally evolved. The names that refer to the different targets

Table 3.1: Image evolution results. The number of runs (out of 20) that reached the solution threshold, the number of generations required to reach a solution, and the CPPN solution complexity (i.e. the number of functions and connections) are shown. Values are averaged over runs that achieved the solution threshold. For comparison, the same results for the original discoveries of each target in Picbreeder are shown in parenthesis.

Target	Solved	Generations		Functions		Connections	
Simple	14	3,774±5,902	**(1)**	19.5±2.7	**(6)**	23.4±4.2	**(5)**
Crescent	11	3,500±3,298	**(12)**	20.3±3.2	**(7)**	26.4±4.4	**(7)**
Eye	1	4,840	**(12)**	18	**(10)**	24	**(16)**
Warp	0	-	**(5)**	-	**(9)**	-	**(12)**
Butterfly	0	-	**(90)**	-	**(25)**	-	**(75)**
Skull	0	-	**(74)**	-	**(23)**	-	**(57)**

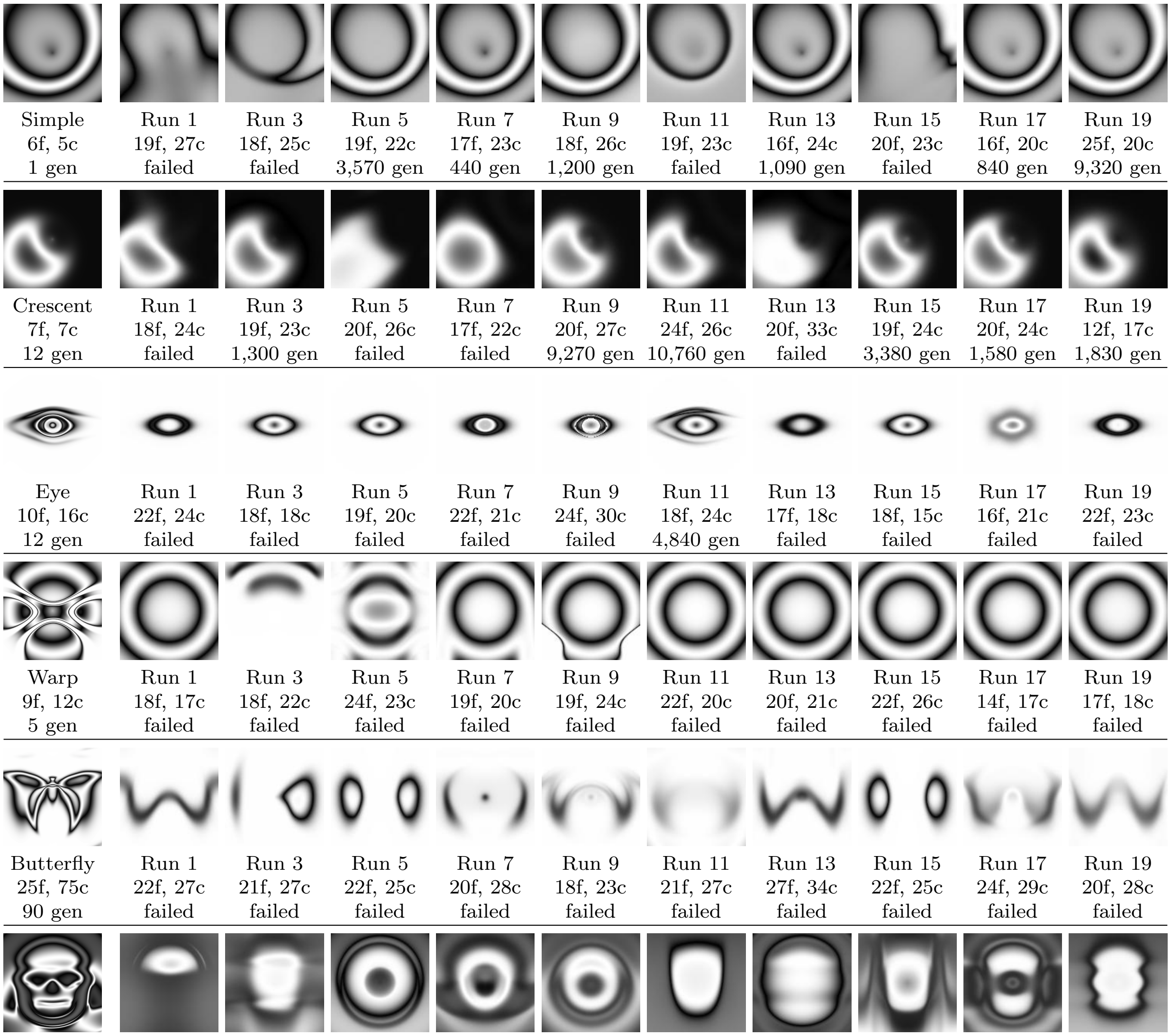
in this chapter are given in figure 3.2. Of runs evolving to the Simple target, 14 of 20 reached the solution threshold in $3,774 \pm 5,902$ (median $= 965$) generations on average (recall that the simple target is from generation 1). Of these, the objective-based solutions were significantly larger (19.5±2.7 functions and 23.4±4.2 connections) than the original Picbreeder discovery (6 functions and 5 connections). Similarly, for runs evolving toward the Crescent, 11 of 20 reached the solution threshold in $3,500 \pm 3,298$ (median $= 2,320$) generations (as opposed to only 12 generations for the original discovery of the Crescent). Of these, the representation was again significantly bloated (20.3 ± 3.2 functions and 26.4 ± 4.4 connections) compared to the target evolved in Picbreeder (7 functions and 7 connections). When evolving toward the Eye, only one of 20 runs reached the solution threshold. This particular run reached the solution criterion at generation 4,840 (as opposed to 12 generations to find the original in Picbreeder). Furthermore, the resulting solution is once again more complex (18 functions and 24 connections) than the target (10 functions and 16 connections).

Beyond these simpler images, at a certain level of complexity it becomes too hard to come even close to hitting the targets: All attempts to recreate the Warp, the Butterfly, and the Skull failed to produce comparable solutions.

A set of champions from the 20 attempts to reach each of the six targets are shown in figure 3.3. To present the results on a single page and to provide a comprehensive sampling of the typical results of evolving to each target, figure 3.3 shows the final result from the ten odd-numbered runs for each target (out of 20 for each). The even runs (not shown) generally follow a similar pattern. The size of the CPPNs of these champions are also shown. In cases labeled as *failed*, the run could not meet the solution criteria even after 30,000 generations. These pictorial results give a qualitative sense of how difficult it is for NEAT to reach the targets it once evolved previously under different conditions. Based on these results, the next section discusses the destructive impact of objective-based search on representation.

3.4 Implications

The experimental results document nothing less than the systematic failure of the objective-driven search. It could not come close to reproducing most of the images. On the other hand, although it could reasonably reproduce the simplest two images (i.e. Simple and Crescent), the re-evolved solutions in those cases contain two to four times more structure than the CPPNs of the original discoveries. Taken together, these dismal results signify a serious



Simple 6f, 5c 1 gen
Run 1 19f, 27c failed
Run 3 18f, 25c failed
Run 5 19f, 22c 3,570 gen
Run 7 17f, 23c 440 gen
Run 9 18f, 26c 1,200 gen
Run 11 19f, 23c failed
Run 13 16f, 24c 1,090 gen
Run 15 20f, 23c failed
Run 17 16f, 20c 840 gen
Run 19 25f, 20c 9,320 gen

Crescent 7f, 7c 12 gen
Run 1 18f, 24c failed
Run 3 19f, 23c 1,300 gen
Run 5 20f, 26c failed
Run 7 17f, 22c failed
Run 9 20f, 27c 9,270 gen
Run 11 24f, 26c 10,760 gen
Run 13 20f, 33c failed
Run 15 19f, 24c 3,380 gen
Run 17 20f, 24c 1,580 gen
Run 19 12f, 17c 1,830 gen

Eye 10f, 16c 12 gen
Run 1 22f, 24c failed
Run 3 18f, 18c failed
Run 5 19f, 20c failed
Run 7 22f, 21c failed
Run 9 24f, 30c failed
Run 11 18f, 24c 4,840 gen
Run 13 17f, 18c failed
Run 15 18f, 15c failed
Run 17 16f, 21c failed
Run 19 22f, 23c failed

Warp 9f, 12c 5 gen
Run 1 18f, 17c failed
Run 3 18f, 22c failed
Run 5 24f, 23c failed
Run 7 19f, 20c failed
Run 9 19f, 24c failed
Run 11 22f, 20c failed
Run 13 20f, 21c failed
Run 15 22f, 26c failed
Run 17 14f, 17c failed
Run 19 17f, 18c failed

Butterfly 25f, 75c 90 gen
Run 1 22f, 27c failed
Run 3 21f, 27c failed
Run 5 22f, 25c failed
Run 7 20f, 28c failed
Run 9 18f, 23c failed
Run 11 21f, 27c failed
Run 13 27f, 34c failed
Run 15 22f, 25c failed
Run 17 24f, 29c failed
Run 19 20f, 28c failed

Skull 23f, 57c 74 gen
Run 1 20f, 24c failed
Run 3 20f, 29c failed
Run 5 19f, 24c failed
Run 7 22f, 28c failed
Run 9 21f, 28c failed
Run 11 16f, 22c failed
Run 13 21f, 27c failed
Run 15 23f, 29c failed
Run 17 18f, 25c failed
Run 19 25f, 28c failed

Figure 3.3: Image evolution results. Results from the ten odd-numbered runs to each target are shown (as they were evaluated, at 128 × 128 pixels). The even runs, not shown to satisfy space constraints, generally follow a similar pattern. The CPPN complexity, i.e. the number of functions and connections, is shown for each image along with the generation when the solution threshold was reached. Runs for which no solution was reached are labeled *failed*. This pictorial perspective demonstrates the difficulty that NEAT has reaching targets that it previously evolved. Even when reasonable reproductions are achieved, their CPPNs are significantly more complex than the the original discoveries. Thus the implication is that evolving toward an objective creates a barrier to discovery by assuming that the stepping stones resemble the final objective.

underlying pathology; because we know that the target images were evolved in the first place, it should have been possible to reproduce them and to reproduce them efficiently.

The key question is whether these results extend beyond the confines of this particular study to imply something about objective-based search in general. The aim of this section is to argue that they do. However, a natural reaction to such poor performance is to dismiss it by pinning it to specific shortcomings of the algorithm and representation or the experimental setup. Yet the unusual experimental setup, in which targets were chosen that were already evolved by the same algorithm and representation, makes it difficult to dismiss the results so easily.

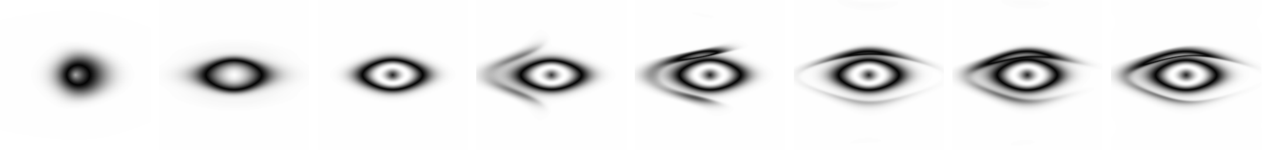
The first objection to drawing general implications is that something might be wrong with NEAT or CPPNs that could be rectified in a different setup. However, it is difficult e.g. to argue that evolving CPPNs with NEAT is somehow ill-equipped to evolve skulls when the Skull has only ever been evolved by NEAT and CPPNs. Not only that, but the Skull was originally discovered in only 74 cumulative generations with a population of 15 on Picbreeder, which makes it hard to argue coherently that it is "difficult" for NEAT to discover. Yet 30,000 generations was not sufficient to rediscover it when it was the objective. Furthermore, although there may be better algorithms, NEAT generally has a good record in a variety of domains (Stanley and Miikkulainen, 2002, 2004; Whiteson and Stone, 2006), diminishing the possible argument that it is a kind of straw man chosen for its weaknesses.

The other potential objection is to the experimental setup. Perhaps equations 3.1–3.3 could somehow better compute image similarity. Yet this objection depends on the assumption that the basic challenge in evolving to a target is to formalize a good *similarity metric*, which turns out not the be the real problem. To see why, observe the stepping stones in figure 3.4 that led to the discovery of the original Eye in just 12 generations (recall that the only successful rediscovery of the eye out of 20 attempts took 4,840 generations). The problem is that the early stepping stones, such as in the first seven generations or so, look nothing like the final eye at all. Given this observation, a "better" image-comparison metric would be even *worse* for this task because it would penalize the essential stepping stones (which do not look like the Eye) severely. Only in retrospect can we see why the stepping stones in figure 3.4 might lead to an eye. The user who originally found them chose them *for their own appeal*, not because they resemble the final published product. In fact, we can see the destructive effect of a "good" comparison metric (and see that the metric in equation 3.3 is actually good) in figure 3.5, which shows stepping stones in the sole 4,840-generation rediscovery of an Eye-like image. These stepping stones *do* resemble the final image, which is exactly why it takes so long to find it: Looking similar is exactly the *wrong* heuristic for identifying the most natural stepping stones.

In fact, the fallacy of the experimental-design objection exposes a fundamental flaw with objective-driven search in general: There is no a priori reason to believe that a metric that measures distance to the objective in *any* domain has a useful relationship to the essential stepping stones. The better the fitness function describes the objective, the more deceptive it

gen 1 gen 2 gen 3 gen 5 gen 7 gen 9 gen 10 gen 12

Figure 3.4: Stepping stones of non-objective image evolution. This sequence shows evolution guided by a single user with no explicit objective. The first image (gen 1) was selected from the initial population and progressed as shown to produce the published image (gen 12) known as the Eye, which emerged after 12 generations and is represented by a network of 10 functions and 16 connections. This example shows why the final objective is a poor heuristic for identifying stepping stones that lead to itself.



gen 39 gen 60 gen 128 gen 590 gen 2,201 gen 2,492 gen 3,429 gen 29,193

Figure 3.5: Evolving to an objective. The depicted sequence shows significant stepping stones reached by following the fitness gradient toward the Eye objective in figure 3.2c. Contrasting these stepping stones with those from the original discovery (figure 3.4) shows why resemblance to the target is not the right heuristic for identifying stepping stones for this problem.

may nevertheless be, which means that effort to better formalize the objective is misguided. The more ambitious and complicated the objective, the more profound this gap will be (as with the Warp, the Butterfly, and the Skull).

Yet while the more sensational result is the failure to reproduce the more complicated images, perhaps most sinister is what happens with the simpler images. They show that even the *appearance* of success is not genuine success. For example, although the Crescent superficially appears to have been rediscovered consistently (figure 3.3), in *every* case the CPPN is at least twice as complex as the original representation (and usually three times more complex).

The same is true for the Simple objective. Thus, *even if* the objective is reached because the problem is simple enough, a price will still be paid in the form of poor representation.

The reason for this representational inefficiency is that objective-based search by its nature encourages a *piecewise* solution because it rewards small changes in parts of the solution that increase its resemblance to the objective, as can be seen in the unhealthy (yet successful) progression in figure 3.5. Each such small change requires adding a small amount of new structure to the representation to capture that piece. The result is a hodgepodge of functions that produce the correct pattern but in an inefficient way, which is why it takes thousands of generations to do what should take only a dozen. The kind of *holistic* optimization that would have been ideal is impossible when the stepping stones that optimize the global structure do not resemble the final objective, as in figure 3.4.

In the short run, while we are only interested in immediate solutions to simple problems, this representational pathology may not matter, but in the long run, when we may want evolution to continue indefinitely or to build upon its past results, it will ultimately obstruct progress in the field. For that reason, the pathological effect of objectives on representation *even when search succeeds*, which has received little attention, merits significant further study. Many positive reported results may be unwittingly subject to this pathology, which is only uncovered in this study because the objectives were already evolved under different conditions.

This last point leads to a final important insight: *How* something is evolved matters. The mode of evolution impacts both the search and the representation, even when successful. Yet almost all our experiments are objectively driven. While the study in this chapter focuses on images, the same principles apply to more practical domains as well, such as evolving controllers. For example, how do we know that the most natural stepping stones to a pole-balancing robot, which is a common benchmark (Gomez and Miikkulainen, 1997; Stanley and Miikkulainen, 2002), are actually increasingly better at balancing a pole? In fact, the objective paradigm is so dominant that even considering the alternatives appears potentially radical. Yet we should not forget that the ultimate inspiration for our field is in nature, where evolution produced such marvels as photosynthesis, the flight of birds, and the human brain. Yet *none* of these discoveries were set as a priori objectives for the search. Instead, they are serendipitous discoveries on the road to nowhere in particular. Picbreeder (Secretan et al., 2011, 2008) is similar; while its users stop at serendipitous waypoints on the road ahead, the system as a whole has no overriding purpose against which it is measured.[2] Is it a coincidence that its representations are so compact and its discoveries so rapid? Novelty search (Lehman and Stanley, 2008, 2011) is another example of a search process without an explicit a priori objective. Thus interactive evolutionary processes like Picbreeder and non-objective automated algorithms like novelty search emerge as possible alternatives to the current objective-driven paradigm that are deserving of further investigation, which

---

[2] It is also interesting to note that users that come to Picbreeder with specific a priori objectives in mind often find the system frustrating (Secretan et al., 2011). Trying to re-evolve the skull from scratch (even as a human), which is virtually impossible, illustrates why discoveries on Picbreeder cannot simply be attributed to an uncanny (e.g. human) objective understanding of the search space.

motivates the idea in this dissertation. In light of the results in this study, it is important to begin asking whether EC is inadvertently distorting a critical aspect of its original inspiration and thereby losing an essential ingredient: Nature has no final objective.

To conclude the discussion, while the discovery of e.g. the Skull could be dismissed by noting that NEAT only discovered it *once* but has shown no propensity to produce the same result again, perhaps our usual meaning of "result" is misguided. Perhaps the *result* in Picbreeder is not an individual image, or even a set of specific images, but rather the fact that it consistently produces interesting images in general (figure 2.4). We could dismiss the Skull, but should we dismiss the Skull, the Car, the Dolphin, the Snake, the faces, the butterflies, the Apple, the Octopus, the insects, the Pig, the planets, the Tiger, the Sunset, the Candle, the Eye, the Penguin, etc. (Secretan et al., 2011)? At what point do we accept that an EA is valuable not for its ability to produce a *particular* artifact that we want, but rather to consistently produce artifacts *in general*? In other words, is the traditional approach to *evaluating* evolutionary algorithms based on their ability to consistently achieve a particular objective flawed? After all, nature, the original inspiration for our field, has never discovered the very same organism in different lineages, yet its profusion of unparalleled discoveries remains no less potent as a clue to what is possible. What service then do we do to our understanding of evolutionary *algorithms* by judging them for their ability to repeat the same trick many times over? Is that really evolution's greatest trick, or is it a distraction from the real engine of evolution, which is creative discovery? Once, after all, is enough for nature.

3.5 Conclusion

The aim of this chapter was to provide a unique perspective on both the downside of objectives and the cost to our field of judging EAs only for their ability to repeat the same achievement multiple times. While evolving without an objective is presently uncommon, such an approach allows fundamental principles (such as bilateral symmetry) to be discovered that may serve as stepping stones to interesting areas in the search space, which often could not have been reached by following a similarity metric. Experience with Picbreeder demonstrates the power of serendipitous discovery and highlights the importance of rewarding stepping stones for what they contribute in their own right. In this context, it may make more sense to judge EAs on their ability to produce a diversity of discoveries rather than on the probability of reaching the same objective over and over again. Motivated by these considerations and drawing on the innate ability of humans to identify what is interesting, the new evolutionary approach introduced in the next chapter opens the door to the serendipitous discovery of *behaviors*. Like Picbreeder, this new approach supports non-objective exploration through interactive evolution, but unlike Picbreeder, it also provides a broader perspective to the user by building the next population presented to the user by first searching for novel behaviors around the points that the human user found interesting. In this way, augmenting interactive evolution with novelty search will create a synergistic system in which human intuition can help to identify innovations from a diverse collection of phenotypic behaviors.

CHAPTER 4
APPROACH: NOVELTY-ASSISTED INTERACTIVE EVOLUTIONARY COMPUTATION

Given the deleterious effects of a priori objectives on what simulated evolution can discover, this chapter motivates and introduces a new evolutionary approach that that combines for the first time the intuitive ability of human users to identify what is interesting and important in a domain, i.e. interactive evolutionary computation (IEC), with a stepping stone generator based on a short-term novelty search to create a synergistic effect that expedites the evolution of controller solutions. The hope is that this approach will deliver meaningful representations of behaviors that can be extended and elaborated in much the same way that Picbreeder allowed a community of users to create a diverse phylogeny of images. Under this new approach, called *novelty-assisted interactive evolutionary computation* (NA-IEC), a human user is asked to select individuals from a population of candidate behaviors. Variations of the user-selected behaviors then become seeds for a larger background population wherein novelty search is applied to find sufficiently novel individuals to fill the next on-screen IEC population. At that point, the novel individuals become the next IEC generation and control is returned to the user. In addition to such interleaved novelty searches, the NA-IEC framework also allows the user to apply a traditional IEC step operation or perform a fitness-based optimization.

4.1 Motivation

The motivation for this new approach is based on several results in recent years that have hinted at the limitations of traditional objective functions, wherein the more a candidate resembles the objective, the higher its fitness. Whether the objective is to evolve a particular behavior like balancing a pole (Gruau et al., 1996; Stanley and Miikkulainen, 2002) or a particular morphology like a French flag via a developmental system (Miller, 2004), such objective-based evolution is the dominant approach across a wide breadth of domains and methods.

An early hint that such an approach to fitness may be flawed was from experiments with the *novelty search* algorithm (Lehman and Stanley, 2008), which rewards novel behaviors instead of rewarding objective performance. Recall from Section 2.3 that novelty search significantly outperformed objective-based fitness in a deceptive maze-navigation domain (Lehman and Stanley, 2008, 2011), showing counter-intuitively that in some deceptive cases it is possible that having no specific objective may work better than rewarding progress toward the objective.

In novelty search (Graening et al., 2010; Lehman and Stanley, 2008, 2010a,b, 2011; Mouret, 2011; Risi et al., 2011, 2009), the way the search space is explored places no restrictions on what behaviors can emerge. Such an approach may appear akin to an exhaustive search of the behavior space; however the pressure to be novel provides an efficient way of identifying which genetic mutations produce innovative behaviors. As an example, when a particular mutation results in a number of distinct behaviors, novelty search inherently identifies this new area of the search space as interesting. Then, as the pressure to be novel in this new space increases, the hope is that another mutation will occur that opens another interesting area of the behavior space. The unique aspect of searching for novelty is that the selection criteria is constantly changing. Further, strings of innovations that repeatedly lead to novel behaviors are establishing concepts that can be extended and elaborated by subsequent mutations. Thus evolving along the gradient of novelty is more apt to incorporate fundamental knowledge about the domain into the genome (Lehman and Stanley, 2011).

Yet novelty search is not the only clue that something is amiss with fitness. Chapter 3 provided another hint, wherein attempts were made to *re-evolve* images that were previously evolved interactively by human users on the Picbreeder online service (Secretan et al., 2011). The strange result was that none of the more interesting images, such as the Butterfly and the Skull, could be re-evolved by the very same evolutionary algorithm when they are made the automated objective. In other words, even though a set of users together evolved a picture of a Skull in only 74 generations, 20 automated attempts of 30,000 generations each were unable to reproduce the result. Remarkably, Picbreeder is full of images that were evolved by users in just a few dozen generations and with no specific objectives, yet each is nearly impossible to reproduce when they are made objectives.

Another intriguing result from Chapter 3 is that when simple images are re-evolved, their underlying representation is at least three times larger than the original image evolved by human user in Picbreeder, suggesting that fitness-based selection all but guarantees that the resulting representation will be a piecewise construction of the a priori objective, if it can be reached at all. Such experimental results suggest that *how* something is evolved has an effect on the resulting representation. In particular, searching for an a priori objective is usually deceptive because stepping stones that *do* capture important concepts often do not resemble the objective. To illustrate this point, figure 4.1 shows a number of significant discoveries in Picbreeder for which a key stepping stone was very different from the resulting image. The plethora of such examples in Picbreeder shows that serendipitous discoveries are *not* accidental, nor are they uncommon.

In one particular account, Stanley (2010) relates that the he branched the Alien Face (figure 4.1b-left) expecting to evolve more alien faces. Instead, what he saw was that the eyes descended to become like the wheels of a car (figure 4.1b-right), an unexpected innovation that he recognized as significant. The point is, if Stanley had intended to evolve a car, then he would *not* have chosen to branch from the Alien Face, nor would he have thought to evolve the Alien Face along the way. From this experience emerged the paradoxical insight that the only way to find images like the Teapot, the Car, the Skull, the Butterfly, the Planet, the Lamp, and many others is by *not* looking for them. Such experiences from Picbreeder suggests that humans are uniquely adept at identifying promising stepping stones, even if their ultimate destination is entirely unclear (Woolley and Stanley, 2011).

(a) Stepping stone to the Teapot

(b) Stepping stone to the Car

(c) Stepping stone to the Skull

(d) Stepping stone to the Butterfly

(e) Stepping stone to Jupiter

(f) Stepping stone to the Lamp

Figure 4.1: Stepping stones in Picbreeder rarely resemble the final product. Some examples include: the Egg Wearing a Hat yielded the Teapot (a), the Alien Face yielded the Car (b), the Dish yielded the Skull (c), Face to Face yielded the Butterfly (d), the G yielded a Jupiter-like planet (e), and an eye yielded the Lamp (f). Nearly every interesting image in Picbreeder emerged as the result of such a serendipitous discovery.

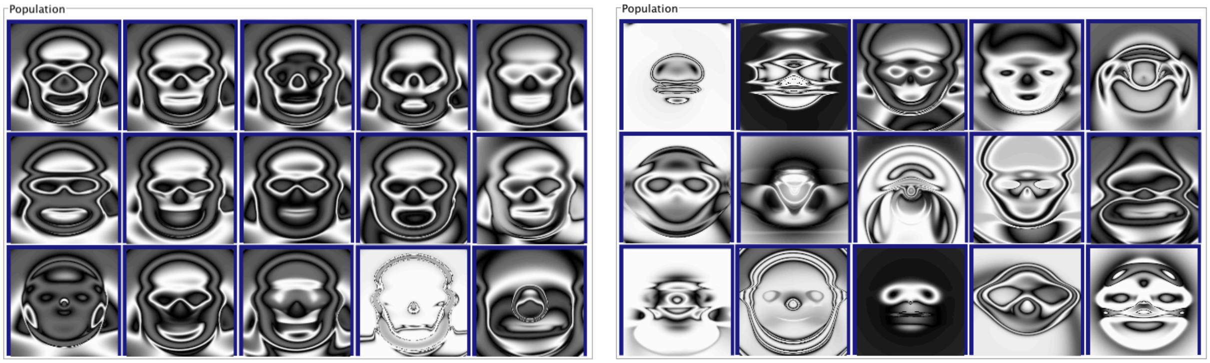
While the results from Picbreeder and novelty search are intriguing, their interpretation has also been controversial. Although they suggest that searches not driven by explicit objectives might sometimes offer more potential than those that are, they seem to offer few alternatives other than searching only for novelty or leaving the search entirely to human guidance. However, work with novelty search has shown that it may become lost in especially large spaces (Kistemaker and Whiteson, 2011; Lehman and Stanley, 2010b), and Takagi (2001) warns that interactive evolutionary computation (IEC) is limited by human fatigue. With such limitations for alternative approaches, the news that traditional objectives offer little hope is not especially encouraging.

One potential response is to hybridize an objective-based search with a search for novelty, as in Novelty-Based Multiobjectivization (Mouret, 2011). Yet while this idea undoubtedly works in some cases, in others the reintroduction of the objective, even partially, only disadvantages the search. After all, as recent critiques of objective-based search have pointed out, the fundamental problem with objectives is that they often penalize essential intermediate stepping stones that lead to the objective because those stepping stones do not resemble the objective (Ficici and Pollack, 1998; Lehman and Stanley, 2008, 2011; Woolley and Stanley, 2011), and reintroducing a deceptive objective back into the search consequently reintroduces the deleterious effects of such objective-based search. Such concerns do not imply that objectives are never useful, or that hybrid objective/non-objective approaches cannot help; rather they open the door to the possibility that more can be done to emphasize the discovery of essential stepping stones.

Thus the aim of the NA-IEC approach is to translate Picbreeder's ability to serendipitously discover interesting areas in the space of evolvable images to support the evolution of principled agent *behaviors*. The intuition is that human users will similarly recognize principled controller policies in the initial generations and later in the search that will establish what is important in the domain and lead to robust solutions. However, because the discoveries of Picbreeder are also the result of many minds with many divergent interests (Stanley, 2010), as a replacement for a community of users laying stepping stones for each other, the NA-IEC approach interleaves short-term novelty searches with such interactive evolution to provide innovative stepping stones from which serendipitous discoveries can lead to robust solutions. The hope is that these two techniques, which have never been combined before, will work together to achieve innovative discoveries that could not be made with a priori objectives. The goal is to discover robust agent behaviors by abandoning the fitness gradient and leveraging human intuitions about the domain, bootstrapped with novelty-driven exploration.

By allowing a human evaluator to explore the space of agent behaviors without regard to an objective, and by augmenting the traditional IEC approach such that the next population is generated by a short-term novelty search, the human evaluator is presented with a broad view of where the evolutionary search can go from its current position. To illustrate this idea, figure 4.2 shows two Picbreeder populations as a metaphor for what will ultimately be *animations* of agent behavior in the final implementation (Chapter 7). Figure 4.2a shows a typical population reproduced from the Skull, where variations are the result of random mutations. Figure 4.2b, which is also derived from the Skull, shows a collection of diverse yet still interesting images that were discovered within the vicinity of the Skull. This example illustrates how interleaving IEC with novelty search can potentially enhance the human-led evolution by providing a summary of the novel artifacts nearby.

In practice, this ability to quickly generate novel stepping stones is important because the populations of animated candidate behaviors must be evaluated individually, which is a



(a) Random mutations of the Skull **(b)** Novel candidate images beyond the Skull

Figure 4.2: Mockup of a novelty-assisted Picbreeder search. A typical population of candidate images branched from the Skull (a) exhibits random mutation alone, giving the human evaluator little opportunity to consider genuinely innovative mutations. Instead, (b) shows a mockup of what running novelty search for just a few generations between user selections might provide. In this way, the human evaluator is presented with a set of agnostic stepping stones in the current search area from which they may identify meaningful innovations and follow the gradient of interestingness toward a principled solution that could not have been known to exist a priori.

burdensome task compared to the quick glance required for Picbreeder. Additionally, this approach also provides the human user with a breadth of stepping stones that they would never have created on their own, thus substituting for the key feature in Picbreeder that enables the proliferation of serendipitous discoveries, i.e. the ability to branch from another user's published image. The next section provides a detailed description of the NA-IEC approach and subsequent framework.

4.2 Novelty-Assisted IEC Framework

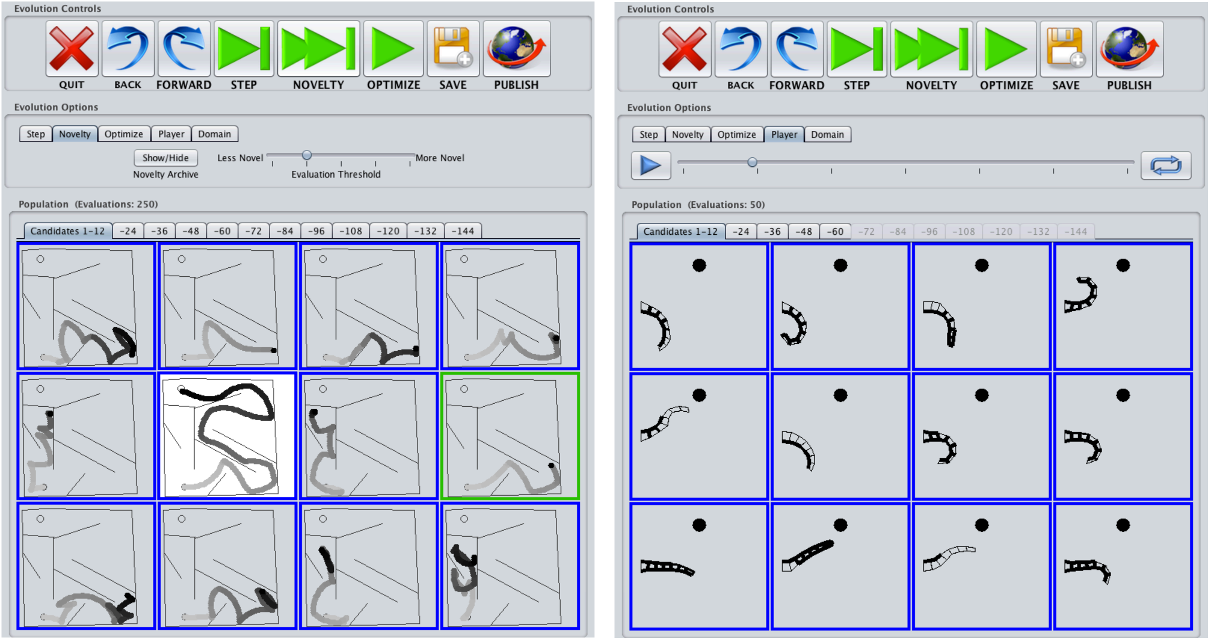
The main idea behind the NA-IEC approach is that the ability of humans to identify promising stepping stones is naturally complemented by the ability of novelty search to generate *sets* of potential stepping stones. In other words, novelty search can mitigate the main weakness of IEC, i.e. that humans grow tired quickly (Takagi, 2001), by offloading most of the exploratory work. This way, novelty search becomes a kind of stepping-stone scavenger that is interleaved with human evaluations that determine which stepping stones are the most promising. Furthermore, neither the human nor the novelty search are guided by any explicit objective, thereby also mitigating the threat of reintroducing deception. Furthermore, in this approach, instead of forcing a human experimenter to *articulate* through a fitness function exactly what should be rewarded in a complex domain, the human can instead leverage highly-nuanced *implicit* hunches that all of us have about what is promising. The result is a powerful synergy between two promising non-objective processes that reintroduces to novelty search a sense of control (i.e. from the human) without reintroducing an explicit objective.

In addition to the ability of humans to recognize key stepping stones (i.e. interesting behaviors), users familiar with EC are also likely to have some sense about what direction evolution ought to go next. In other words, while individual modes of evolution have limitations overall, each is effective under the right conditions. Why then lock ourselves into any one such approach? Thus the NA-IEC framework described here allows the user to specify the mode of evolution that they feel is most appropriate at each step during evolution; the three evolutionary modes provided are: a traditional IEC step, a short-term novelty search, and an objective-based optimization operation.

In this way, the user can apply short-term evolutionary operations where appropriate, even changing the mode of evolution during the course of the search, to reach a satisfactory (or just interesting) solution. The ability of a human user to apply powerful automated approaches like objective-based search (De Jong, 2002; Eiben and Smith, 2003; Fogel, 2006; Goldberg, 1989) and novelty search (Lehman and Stanley, 2008, 2011) in short bursts and when appropriate is a key contribution of the NA-IEC approach. The primary hypothesis is that letting the user make a relatively small number of critical selections during evolution, and leaving the remainder of search to automated approaches seeded by those user selections, can significantly augment the pace of evolution and the quality of its discoveries.

Figure 4.3 shows the main interface for the system, where the user can choose among the *Step*, *Novelty*, and *Optimize* operations. Choosing the *Step* operation creates a new generation of offspring through the recombination and mutation of the selected candidate behaviors. This classic approach to IEC is simple and computationally inexpensive, i.e. it only creates a handful of new candidates.

Choosing the *Novelty* operation causes evolution to explore the space of agent behaviors without regard to an objective and then present the human evaluator with a broad view of where the evolutionary search can go from its current position. To accomplish this aim,



(a) Maze Domain **(b)** Octopus Arm Domain

Figure 4.3: Screenshot of the NA-IEC user interface. The user interface for the NA-IEC framework consists of the *Evolution Controls*, the *Evolution Options*, and the *Evaluation Population*. Candidate solutions are represented in the maze domain (a) by a gradient trail that shown the robot's behavior, and as animations in the octopus arm domain (b). Selected candidates are shown with a green border and solutions are highlighted with a white background. Unlike traditional IEC applications, the user can select one of three evolution modes at each step: *Step*, *Novelty*, or *Optimize*. The *Publish* button saves the results of a completed run for later analysis. In the future it may be connected to the Internet.

the next IEC population is generated by seeding a larger population with variations of the user-selected candidate behaviors and then running novelty search in the background to find novel individuals (in comparison to what has been encountered previously in the search) based on the sparseness measure $\rho(x)$ from equation 2.1 and the threshold $\rho_{\min}$. The underlying evolutionary algorithm is NEAT (Stanley and Miikkulainen, 2002, 2004), which is often the base algorithm under novelty search (Lehman and Stanley, 2008, 2011).

Furthermore, to ensure that novelty is measured with respect to the entire search completed so far, all individuals encountered during *both* traditional IEC steps and interleaved novelty searches throughout an NA-IEC session are measured for their novelty and entered into the permeant archive if their novelty score is greater than the threshold ρ_{min}.

The novelty search runs until at least n new individuals are added to the evaluation population (although it is possible that more than n such novel candidates may be found when the larger population is first created), where n is the size of an on-screen IEC population. At that point, the collected novel individuals become the next IEC generation and control is returned to the user. By convention, the n novel individuals are sorted by their novelty score before the NEAT-based speciation adjustment to place the most novel candidate behaviors on the first visible page of the on-screen IEC population. While the *Novelty* operation is significantly more computationally expensive than the *Step* operation, it provides the human user with a breadth of stepping stones that would have been time-consuming or impossible to discover on his or her own under the narrow view of a traditional single IEC step, which only presents the user with a handful of direct one-generation descendants. In a sense, the set of stepping stones returned to the user by novelty search is like the set of images evolved by *other* users from which a visitor to Picbreeder can branch: In both cases, someone or something else has put in effort to collect a set of interesting jumping-off points and present them to the user.

By augmenting the human-led interactive search with interleaved novelty searches, a small population can be constructed that contains a set of novel stepping stones around the currently-selected candidates. In the event that evolution cannot fill the next generation with a sufficient number of new archive members in a reasonable amount of time, the evaluation threshold can be decreased incrementally to allow the search to conclude quickly.

This approach does *not* imply that the set of novel agent behaviors presented to the evaluator will be good at a potential task. What is important is that they are *behaviorally* diverse; it is the human evaluator who will direct the search by recognizing what is promising for a given domain. The goal is to promote innovation through serendipitous discovery, and presenting the various directions that the search can take leverages the human evaluator's inherent ability to recognize what is important or interesting in a particular domain.

Finally, because objective-based optimization is likely the best option for *perfecting* well-formed behaviors already discovered, the user is also given the option to request seeding a traditional objective-based search with currently-selected individuals. The objective-based search then runs until a specified solution criterion is met or until the user requests it to terminate, at which point the most fit individuals discovered so far will update the on-screen IEC generation. Providing this traditional option allows users to optimize candidates that are near an objective attractor that the user would prefer to approach automatically once it is within striking distance, i.e. once the search is no longer deceptive and the primary discovery is already made.

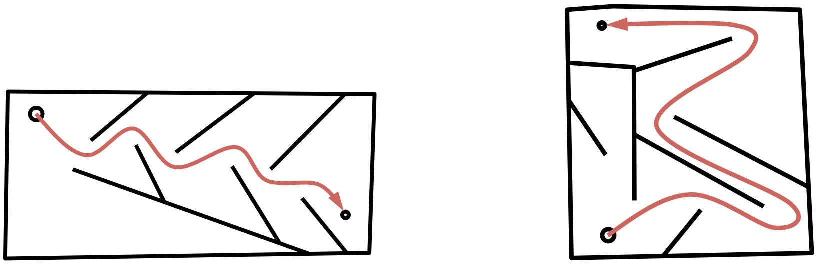
In summary, three types of search will be available: a traditional interactive evolution where human selections become the parents for the next generation, a traditional fitness-based approach where competitive selection is based on performance, and an interleaved novelty search (i.e. the NA-IEC approach) where human selections seed a short-term novelty search that builds the next IEC generation as a collection of novel behaviors starting from what the human user identified as interesting. The user will be free to change the mode of evolution between generations, thus allowing evolution to proceed in the capacity best suited for the current task. In this way, the human user may begin a NA-IEC session by exploring the space of behaviors agnostically. After an interesting behavior is established, the mode of evolution may be changed to optimize it. As a proof-of-concept for this new evolutionary approach, the next chapter revisits the deceptive maze domain (Lehman and Stanley, 2008, 2011) and compares NA-IEC to pure novelty search and traditional fitness-based search.

CHAPTER 5
PROOF-OF-CONCEPT: MAZE NAVIGATION

The purpose of the experiment in this chapter is to establish for the first time the effectiveness of the *novelty-assisted interactive evolutionary computation* (NA-IEC) approach by comparing it to pure novelty search and objective-based search in evolving neurocontrollers for robots in the deceptive mazes of Lehman and Stanley (2011). Interestingly, while novelty search was previously shown significantly more effective than objective-based search in this domain (Lehman and Stanley, 2011), the experimental results here show that NA-IEC outperforms novelty search by a multiple of three to four times, yielding by far the fastest solution on these deceptive problems. Furthermore, NA-IEC is also three times faster in *clock time*, even with the human in the loop, suggesting that perhaps the effort spent crafting objectives functions, which are often deceptive anyway, would be better spent in obtaining a small number of suggestions from a human evaluator during the search process itself. The details of the deceptive maze domain are presented next.

5.1 Deceptive Maze Domain

The deceptive maze domain introduced by Lehman and Stanley (2008, 2011) is the obvious choice as a proving ground for the NA-IEC framework because it is well established as a

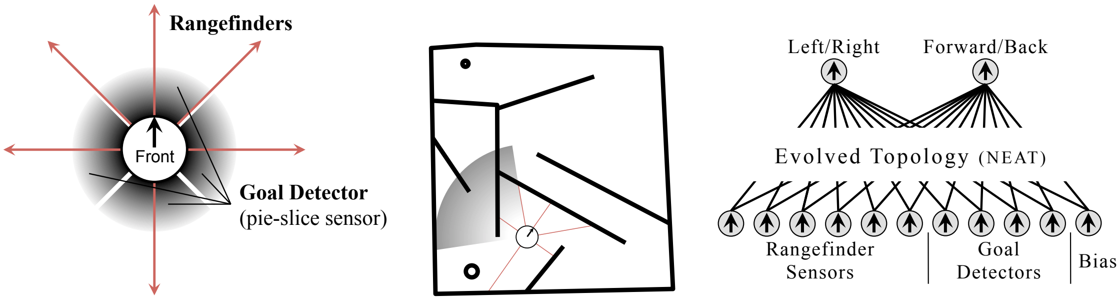


(a) Medium Map **(b)** Hard Map

Figure 5.1: Maze navigation maps (Lehman and Stanley, 2008, 2011). The deceptive maze domain is a metaphor for search and is *not* a path-planning problem. Rather, the aim is to evolve a neural network that drives a robot through the maze; walls represent barriers to search and the cul-de-sacs represent local-optima that can deceive objective-based search.

deceptive benchmark in which novelty search has already been shown to be effective. In this way, the same fitness function and behavior characterization can be applied, thus isolating any performance differences to the individual evolutionary approach.

In the deceptive maze domain a simulated robot must navigate through a maze (figure 5.1) without a priori knowledge of the maze layout. Rather, the agents that act within the maze have a sensor package with six rangefinders that detect the walls and four pie-slice sensors that signal the direction to the goal (figure 5.2a). An example of an agent in the maze is shown in figure 5.2b. Each robot's navigation behavior, encoded as an artificial neural network (ANN), maps sensor inputs to actions, i.e. turn rate (*left*/*right*) and velocity (*forward*/*backward*), as shown in figure 5.2c. Under this construction, navigators must evolve a control policy that traverses the maze based on sensory input.



(a) Sensors **(b)** In the Maze **(c)** Neural Network

Figure 5.2: Maze navigation robot (Lehman and Stanley, 2008, 2011). The sensor package (a) includes six rangefinders that detect walls and four pie-slice sensors that signal the general direction to the goal, e.g. the agent in the maze (b). The navigation behavior, encoded as an ANN (c), maps sensor inputs to actions, i.e. turn rate (*left*/*right*) and velocity (*forward*/*backward*). Under this construction, navigators cannot see the whole maze and must evolve a control policy that traverses the maze based on sensory input.

The medium and hard maps in figure 5.1 are deceptive by design because the maps contain cul-de-sacs that represent local optima in the search space. If fitness is assigned based on reducing the distance to the goal, then the objective function prunes out of the search the deceptive intermediate solutions (i.e. those that move away from the goal location) needed to reach the global objective. While an alternative objective function that rewards specific intermediate solutions is conceivable, the point of this domain is to explore the effect of objective fitness when the precise stepping stones are *not known*, which is the typical predicament in most domains of interest. In such cases, as in the objective function here, performance is generally rewarded for its proximity to the target behavior. Thus evolution driven by proximity to the goal often converges to a cul-de-sac from which the goal is inaccessible.

5.2 Maze Navigation Experiment

The aim of this experiment is to compare the NA-IEC approach against pure novelty search and fitness-based search directly, thus the experiment is conducted in the deceptive maze domain introduced by Lehman and Stanley (2008, 2011) and described in the previous section. In the deceptive maze domain, the goal is to evolve a navigation behavior that drives a robot from the start to the finish of the medium maze or the hard maze shown in figure 5.1, which are constructed with several cul-de-sacs that create local optima in the fitness landscape. Interestingly, these local optima are so deceptive that Lehman and Stanley (2008, 2011) found that novelty search significantly outperforms objective-based search in both mazes. The question here is, can NA-IEC do even better?

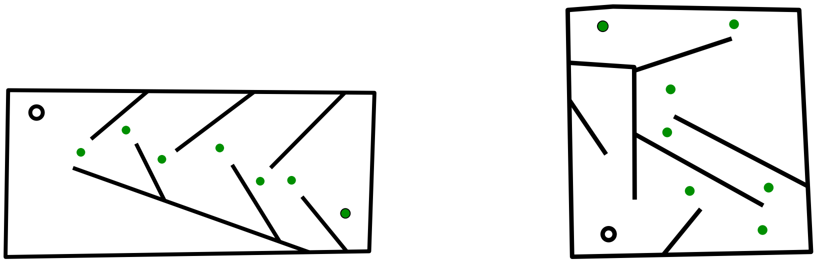
To compare performance, each approach is evaluated over 30 runs on the medium and hard maps. While novelty search and fitness-based search are both automated algorithms, the NA-IEC approach requires a human evaluator. To accomplish the NA-IEC portion of this experiment, six users (not the author) were recruited who were familiar with novelty search and EAs. These users were introduced to the NA-IEC framework and each asked to evolve five solutions to the medium map and five solutions to the hard map. The aim is to characterize the performance that can be reasonably expected from a practitioner in EC when evolving with NA-IEC. Users were permitted to restart if they felt that evolution had become stuck. However, all evaluations before such restarts were recorded as a part of the same run.

Inevitably, some will argue that such human guided runs have an unfair advantage because the user can see the path through the maze. To address this concern, an additional fitness-based experiment, inspired by Risi and Stanley (2011) is conducted. In this additional experiment the primary deceptive element of the maze navigation domain, i.e. the attraction of agents to cul-de-sacs, is removed. In this alternative reward scheme, candidates are rewarded for progressing along a path that actually leads to the goal. Figure 5.3 shows the waypoints (which are invisible to the agent) in the medium and hard maps. In this waypoint-directed version of the experiment, the fitness function f is defined such that agents are rewarded for each waypoint crossed (including the goal), they also receive a partial reward for approaching the next waypoint:

$$f = n + (1 - d), \tag{5.1}$$

where n is the number of waypoints reached and d is the distance to the next waypoint (proportional to distance between waypoint w_i and w_{i-1}, in the range $[0, \ldots, 1]$).

In all variations of this experiment, as in Lehman and Stanley (2008, 2011), the ANN controllers are evolved by the *NeuroEvolution of Augmenting Topologies* (NEAT) approach introduced by Stanley and Miikkulainen (2002, 2004), which was reviewed in Section 2.1. As applied to maze navigation policies, evolution begins with a population of simple behaviors that are represented by fully-connected networks with 22 connections, no hidden nodes, and the inputs/outputs in figure 5.2c. As the underlying networks add complexity (i.e. new nodes and connections), features and nuances emerge in the resulting behaviors that could not be



(a) Medium Map (b) Hard Map

Figure 5.3: Maze navigation waypoints. To compare how much advantage is gained from knowing the path to the goal, waypoints (which are *not* seen by the agent) are provided for the medium and hard maps. In this way, deception is removed by allowing a traditional fitness-based search to reward solutions that discover stepping stones that are on the path to the goal.

expressed by the simpler ANNs. Finally, it is important to note that these experiments, some of which resemble the experiments of Lehman and Stanley (2008, 2011), were *re-run* to ensure a fair comparison and to validate this implementation. A detailed description of the experimental parameters can be found in the Appendix.

5.3 Experimental Results

As with the original experiment by Lehman and Stanley (2008, 2011), a navigation behavior that finishes within five units of the goal location is considered successful. The main result is that NEAT with NA-IEC discovers such solutions in significantly fewer evaluations than both NEAT with novelty search and fitness-based NEAT on the medium and hard maps. Furthermore, despite the expense of waiting on the human to evaluate a panel of candidate solutions, NA-IEC also consumes less clock time in search, suggesting that the value of the user's direction easily offsets the delay of waiting for human input. Another result is that NA-IEC produces solutions with significantly fewer hidden nodes than both novelty search and fitness-based search, further suggesting the importance of allowing a human evaluator to make key decisions about the direction of evolution. While some may dismiss such improvements based on the human evaluator's ability to see the path through the maze, results from the waypoint-directed search, a non-deceptive fitness-based experiment, are on par with NEAT with novelty search, which is still well below the performance of NEAT with NA-IEC. The implication is that NEAT with NA-IEC not only exposes key stepping stones, but also provides evolution with subtle insights about the domain that are not easily incorporated into a traditional fitness function a priori.

On the medium map (table 5.1a), users directing NEAT with NA-IEC found 30 solutions in an average of 6,729 ($\text{sd} = 8,068$) evaluations. These results are significantly ($p < 10^{-6}$; Student's t-test) faster than NEAT with novelty search (22,116 evaluations, $\text{sd} = 10,157$), fitness-based NEAT (55,066 evaluations $\text{sd} = 47,339$), and waypoint-directed NEAT (22,594 evaluations $\text{sd} = 11,982$), each averaged over 30 runs (figure 5.4a). Furthermore, users solved the medium map in an average of 294 ($\text{sd} = 359$) seconds, which is 2.8 times faster than novelty search, 9.1 times faster than fitness-based search, and 2.0 times faster than the waypoint-directed search (figure 5.5a). While solutions from novelty search, fitness-based, and waypoint-directed search have on average 3.2 ($\text{sd} = 1.9$) hidden nodes, 2.9 ($\text{sd} = 1.65$) hidden nodes, and 3.0 ($\text{sd} = 1.8$) hidden nodes respectively, solutions produced by NA-IEC are significantly simpler, averaging just 0.23 (sd = 0.5) hidden nodes per solution ($p < 10^{-10}$; Student's t-test).

On the hard map (table 5.1b), the NA-IEC approach evolved 30 successful navigators in an average of 7,481 (sd $= 6,610$) evaluations, which is a significant ($p < 10^{-5}$; Student's t-test) improvement over not only NEAT with novelty search alone (33,320 evaluations, sd $= 20,949$), but also over the non-deceptive (i.e. waypoint-directed) version of fitness-based NEAT (26,954 evaluations, sd $= 18,464$), each averaged over 30 runs. In the case of fitness-based NEAT, as in Lehman and Stanley (2008, 2011), no comparison could be made because only four of 30 runs evolved solutions for the hard map. Solution rates for the hard map are shown in figure 5.4b. In addition to evolving successful navigators for the hard map in fewer evaluations, NA-IEC did so on average in just 402 (sd = 374) seconds, which is 3.5 times faster than NEAT with novelty search and 2.5 times faster than the waypoint-directed search (figure 5.5b). Regarding complexity, solutions from novelty search have on average 3.3 (sd = 1.8) hidden nodes and solutions from the non-deceptive search have an average of 3.5 (sd = 2.0) hidden nodes, while those evolved by NA-IEC are significantly smaller with 0.5 (sd = 1.01) hidden nodes ($p < 10^{-8}$; Student's t-test).

Typical patterns of exploration for each approach in the medium and hard maps are shown in figure 5.6, which compares the distribution of all ending points visited during a typical run. As Lehman and Stanley (2008, 2011) discovered previously, the traditional fitness-based approach is attracted to the cul-de-sacs in the maze (figures 5.6a and 5.6b), while

Table 5.1: Maze navigation results. The NEAT with NA-IEC approach found solutions to the medium (a) and hard (b) maps in fewer evaluations, with lower complexity (i.e. fewer hidden nodes), and in less time overall than both NEAT with novelty search and fitness-based NEAT. Interestingly, the NA-IEC approach also outperformed a waypoint-directed search that explicitly rewarded agents for progressing along the solution path, thus making it difficult to say that the improvements by NA-IEC are simply the result of the human evaluators ability to see the path through the maze. Such results suggest there is a significant amount of information in this domain that can be leveraged by allowing human insight to guide search.

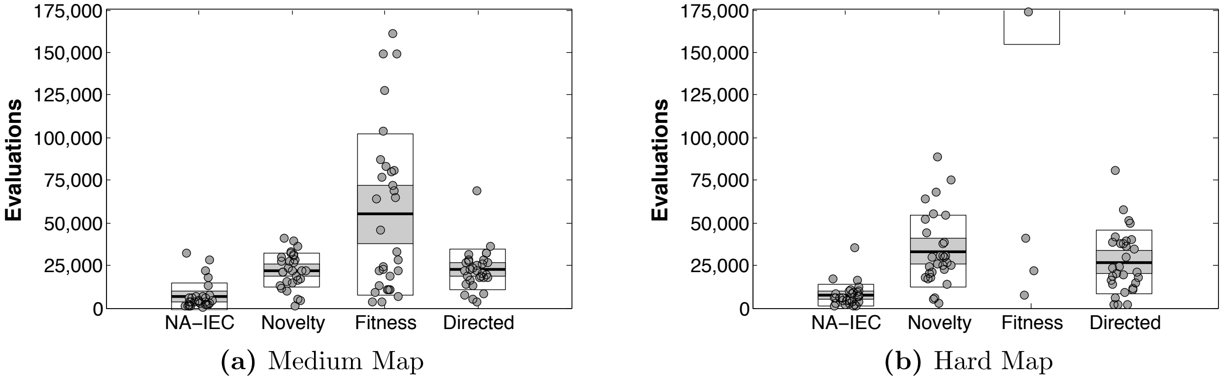
(a) Medium Map

Evolution Mode	Evaluations	Connections	Hidden Nodes	Run Time (sec)
NA-IEC	$6,729 \pm 8,068$	22.5 ± 1.4	0.2 ± 0.5	294 ± 358
Novelty	$22,116 \pm 10,156$	28.4 ± 2.8	3.2 ± 1.9	835 ± 443
Fitness	$55,066 \pm 47,339$	23.3 ± 2.3	2.9 ± 1.6	$2,683 \pm 2,619$
Directed	$22,593 \pm 11,982$	26.8 ± 2.5	3.0 ± 1.8	597 ± 506

(b) Hard Map

Evolution Mode	Evaluations	Connections	Hidden Nodes	Run Time (sec)
NA-IEC	$7,481 \pm 6,610$	22.6 ± 2.3	0.5 ± 1.0	402 ± 374
Novelty	$33,320 \pm 20,949$	28.8 ± 2.8	3.3 ± 1.8	$1,396 \pm 1,058$
Fitness	–	–	–	–
Directed	$26,954 \pm 18,464$	26.8 ± 2.2	3.5 ± 2.0	$1,005 \pm 1,081$

selecting for behavioral novelty allows NEAT to explore the space of possible behaviors more evenly (figures 5.6c and 5.6d). Such search distributions are the result of selection pressure; thus when the objective-function rewards agents for following the solution path (figures 5.6e and 5.6f) the cul-de-sacs no longer deceive evolution. Interestingly, when the points visited during NA-IEC are plotted in this way (figures 5.6g and 5.6h), the signatures of the human selector becomes evident. As expected, the first of these is that there are far fewer points in the cul-de-sacs than in both novelty search and even the waypoint-directed search, demonstrating the intolerance of the human user for behaviors that explore these spaces.

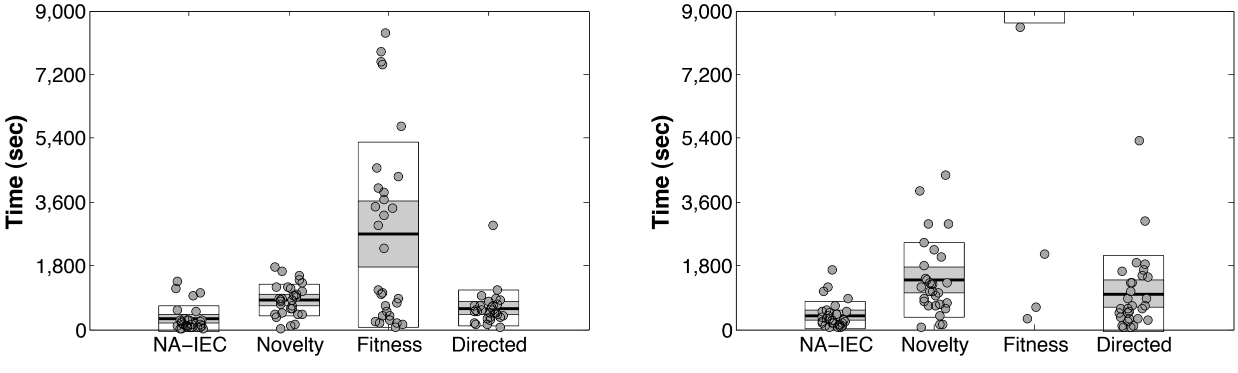


(a) Medium Map **(b)** Hard Map

Figure 5.4: Evaluations required to find solutions. The number of evaluations required by NEAT with NA-IEC, NEAT with novelty search, fitness-based NEAT (pure), and waypoint-directed NEAT to find solutions are shown for the medium (a) and hard (b) maps. The average number of evaluations to reach a solution is marked by a line while the boxed regions extend out to one and two standard-deviations; the distribution of the individual data points is also shown. As in work by Lehman and Stanley (2008, 2011), fitness-based NEAT is generally deceived in the hard map and is unlikely to produce solutions. The main result is that the NA-IEC approach consistently finds solutions for the medium and hard maps in significantly fewer evaluations than not only novelty search and fitness-based search, but is also faster than fitness when the path through the maze is known. Such results suggest that the human user's ability to recognize and select important characteristics as they emerge is directing evolution in a meaningful way.

The second signature is that there are frequently tight groupings of points at key junctions in the map, indicating that the user is probing these areas of the search space for a behavior that can turn a corner and enter a new chamber of the maze. Such observations demonstrate how the human evaluator is contributing his or her insights to the search. Furthermore, it is interesting how these human effects are so readily visible in the points plotted.

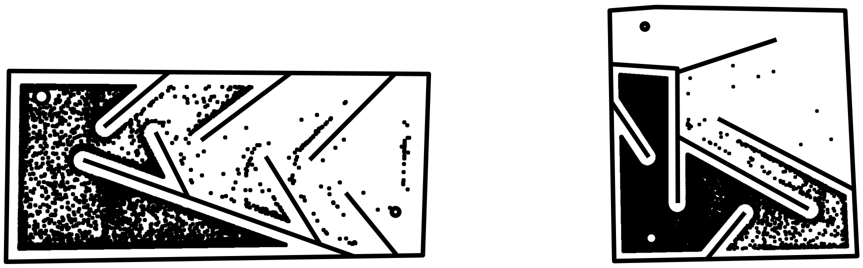
Finally, it is also important to analyze the behavior of the human users, especially in light of the human susceptibility to fatigue in IEC (Takagi, 2001). During the NA-IEC runs on the medium map, users made an average of 30.1 (sd = 40.5) choices, applying the *Step* function



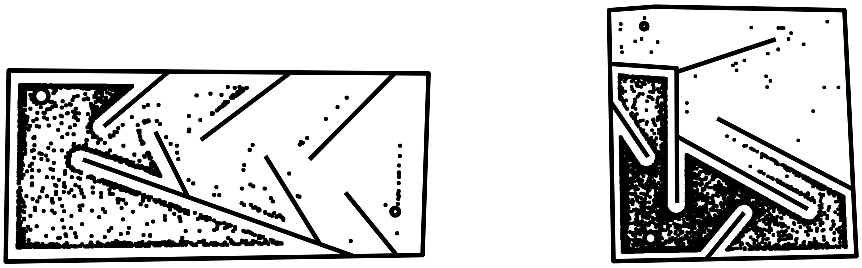
(a) Medium Map **(b)** Hard Map

Figure 5.5: Wall-clock time required to find solutions. The actual time required for different approaches to find solutions are shown for the medium (a) and hard (b) maps. The average number of seconds to reach a solution is marked by a line while the boxed regions extend out to one and two standard-deviations; the distribution of the individual data points is also shown. Surprisingly, the NA-IEC approach finds solutions for the medium and hard maps significantly faster, even with a human in the loop. This result suggests that the human evaluator's time may be better spent looking for solutions serendipitously during evolution instead of spent crafting an appropriate fitness function.

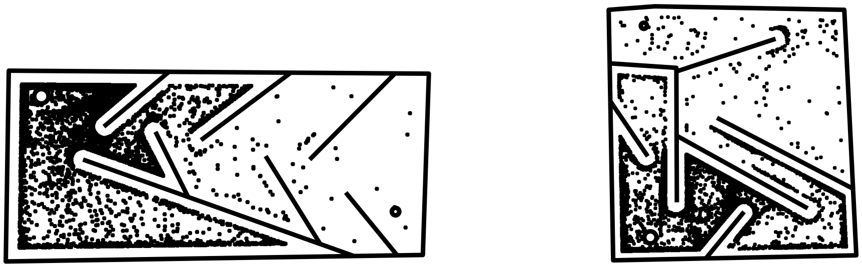
29.8% of the time, the *Novelty* function 47.8% of the time, and the *Optimize* function 22.4% of the time. Similarly, solution were found for the hard map with an average of 32.0 (sd = 23.5) human choices, of which 29.2% were *Step* functions, 58.9% were *Novelty* functions, and 11.9% were *Optimize* functions. These statistics demonstrate that *Novelty* is the preferred operation at most times, and that out of thousands of evaluations, only a few dozen user selections can dramatically reduce the overall cost of a run.



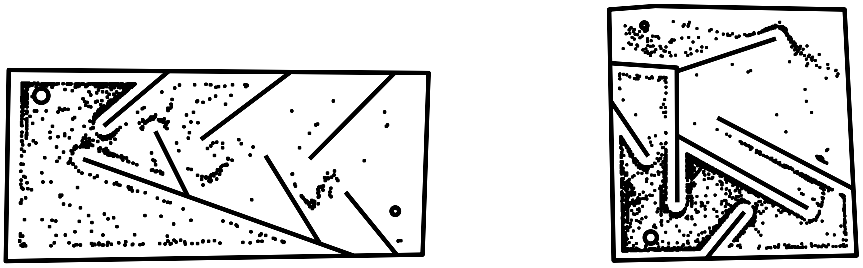
(a) Medium Map Fitness **(b)** Hard Map Fitness



(c) Medium Map Novelty **(d)** Hard Map Novelty



(e) Medium Map Waypoint-Directed **(f)** Hard Map Waypoint-Directed



(g) Medium Map NA-IEC **(h)** Hard Map NA-IEC

Figure 5.6: Distribution of final points visited. Each maze shows the final position for all candidates in a typical run. The density of points shows how NEAT with NA-IEC, NEAT with novelty search, and fitness-based NEAT behave in the deceptive maze domain. As in Lehman and Stanley (2008, 2011), fitness-based search is attracted to the cul-de-sacs, while the points visited by novelty search are more evenly distributed throughout the maze. For the NA-IEC runs, the human user's influence is clearly visible, i.e. there are significantly fewer points in the major cul-de-sacs and tight groupings of points around key junctions. Such characteristics reveal how human selections are impacting evolution.

5.4 Implications

In the deceptive maze domain, humans make a good team with novelty search and objective optimization, which helps to finish the job. In both mazes, users choose *Novelty* to generate the next set of choices significantly more frequently than the other options. The stepping-stone generator of novelty search provides a desirable menu of possibilities to the human user, ultimately exceeding the performance of novelty search alone by several times.[1] Nevertheless, a natural question is whether such results are somehow specific to the maze domain. Perhaps humans harbor a particularly keen insight into the most promising robot behavior in mazes, but would lack such insight in other domains.

For example, one hypothesis might be that humans in effect know the right path through the maze because they can see the whole maze. Yet this interpretation is not entirely accurate. The correct path *through the maze* is not equivalent to the correct path *through the search space.* While some behaviors seem clearly dead ends (such as being caught in the most obvious cul-de-sac in the hard maze), others are less obvious. It is not necessarily the case that just because one behavior drives the robot farther down the correct path that it must be a more promising stepping stone. Some such behaviors are themselves dead ends that cannot push farther. Also, humans perceive more subtle and nuanced indicators that are also important, such as path smoothness or unnecessary loops in the robot trajectory. A behavior in which the robot doubles back on itself and then turns back onto the correct path

---

[1] The results in this chapter also exceed the reported performance of novelty–fitness multi-objective hybrids (Mouret, 2011).

may be just as ominous as being stuck in a dead end. Humans intuitively understand *these* kinds of dangers, yet to articulate them in an objective function would be quite challenging, and would almost certainly take more time than simply guiding the search away from them, whether they are easy to formalize or not.

In this sense, while only future empirical results can settle this issue, there is reason to believe that humans would carry similarly critical insights into other domains. For example, in a biped-walking task (Lehman and Stanley, 2011; Reil and Husbands, 2002; van de Panne and Lamouret, 1995), humans can see that certain kinds of leg oscillations are promising even if the robot falls down. Yet to describe exactly what makes them promising in a fitness function is likely prohibitive. The human's overhead view, and hence knowledge of the mazes, should be viewed metaphorically as like any intuitive understanding of the shape of a particular behavior space. Just as we can see in the maze that certain passageways must precede other passageways, so we can see in a biped robot that oscillations and balance must precede walking. While it is possible that the intuitive insight into some domains is less than in the maze domain, the highly significant advantage provided by such insight in the maze domain suggests that even if the advantage were less elsewhere, it could still be significant.

NA-IEC also may be important for more than just optimization. In some spaces, such as in morphological evolution or with sophisticated encodings, we may be more interested in *what is possible* than in achieving a particular end result. The apparent synergy that results from humans combined with novelty search could be leveraged in the future to show us more about such spaces than trying to solve specific problems. With all the limitations recently shown for objective-based search, NA-IEC provides an alternative without relinquishing our desire to have some say in the process, which is what the traditional fitness function usually facilitates anyway.

Finally, perhaps for some the involvement of a human will be unpalatable, violating a desire for total automation in machine learning. Yet the human must be involved *somewhere*. After all, human researchers at the very least define the traditional fitness function for their experiments. That is one reason NA-IEC was tested with humans with experience in EC. Perhaps our effort and knowledge as researchers would be better applied by providing a modest set of hints to evolution that draw on our rich intuitive understanding of the domain, rather than through trying to articulate at the start of evolution an ad hoc formalization of what kind of behavior should necessarily precede what. Humans in this study only spent up to ten minutes to make a few dozen selections among thousands of evaluations, much of which was automated by novelty search. It is arguable that these few minutes represent time better spent than the time-consuming guesswork usually invested in crafting an objective function. In any case, if our aim is to produce the very best results, as opposed to simply showing that an automated process can achieve a particular benchmark, then what we ultimately discover should matter more than how we get there.

5.5 Conclusion

This chapter presented the proof-of-concept experiment for the NA-IEC approach (Chapter 4), wherein the intuitive ability of human users to identify promising stepping stones is augmented by an agnostic stepping stone generator (i.e. novelty search) seeded with the behaviors selected by the user. In this way, evolution proceeds unconstrained by a priori objectives, but still traverses key stepping stones that are meaningful to the human evaluator. The result was a powerful synergy that allowed human users to realize what was important for a given domain *during* evolution. Furthermore, such serendipitous exploration found solutions in fewer evaluations, at lower genomic complexities, and in significantly less time overall than not only novelty search and fitness-based search alone, but also outperformed a waypoint directed search, suggesting that human direction in NA-IEC eases the need to craft domain-specific fitness functions. Thus the experiment validates the key contribution of the NA-IEC approach, i.e. that it accelerates the rate and quality of evolution by leveraging human-level domain knowledge without burdening the user with the responsibility of evaluating every candidate created during evolution.

The next chapter introduces the octopus arm challenge domain by first approaching it through traditional objective-driven search. The NA-IEC approach is then applied to this domain in Chapter 7. If successful, the results of such a culminating experiment would solidify the importance of allowing human intuition to direct evolution.

CHAPTER 6
A HIGH-DIMENSIONAL CONTROL DOMAIN

This chapter presents a traditional objective-driven experiment in the octopus arm domain, a well recognized machine learning domain that is challenging due its high-dimensional state-action space. The original purpose of this work, which is published in the 11th International Conference on Parallel Problem Solving from Nature, PPSN–XI (Woolley and Stanley, 2010), was to demonstrate how an indirectly-encoded neurocontroller for a simulated octopus arm leverages regularities and domain geometry to capture underlying motion principles and sidestep the superficial trap of dimensionality that overwhelms traditional machine learning approaches. However, while the objective-based experiment presented here successfully answers the problem of dimensionality, the hypothesis in this dissertation is that the results could be better, i.e. the evolved controller solutions suffer from the presence of the objective function. Thus this advanced approach to evolving control behaviors sets the stage for a culminating experiment in the next chapter that compares objective-based solutions against solutions discovered serendipitously by the NA-IEC approach presented in Chapter 4. The hope is that the non-objective approach, guided by the human evaluator's insights and experiences, will discover arm control behaviors in fewer evaluations.

To understand the need for an indirectly encoded controller, note that traditional approaches in evolutionary computation and machine learning often associate the number of dimensions in the state and action space with problem difficulty (Floreano et al., 2008; Kaelbling et al., 1996; Sutton and Barto, 1998; Yao, 1999). Yet the complexity of problems should *not* be determined by the dimensionality of such representations, which are a superficial proxy for the underlying *conceptual* problem. Instead, the problem complexity should correlate with the underlying *principle* to be discovered. The argument behind the octopus arm experiment in this chapter is that indirect encoding, which means describing the solution as a *pattern* through a compressed representation (Bongard and Pfeifer, 2003; Hornby and Pollack, 2002; Stanley et al., 2009; Stanley and Miikkulainen, 2003), is the essential ingredient that will allow traditional learning approaches to transcend the superficial aspects of problem dimensionality.

To make this point, the simulated octopus arm is approximated as a two-dimensional structure of interconnected muscles that must act together to create a coordinated behavior. Thus it induces a high-dimensional state space *and* action space (i.e. because each muscle in each arm segment can be articulated independently).

The details of the original octopus arm experiment (Engel et al., 2006) are interesting because they set a new standard for high-dimensional control that this study pushed even further. The 10-segment arm had a state space with 88 dimensions (i.e. position and velocity for each vertex) that map to the six discrete actions shown in figure 6.1. Engel et al. (2006) chose these six actions to reduce the otherwise prohibitively large action space created by so many muscles. However, the aim of the work in this chapter is to learn from the full unprocessed

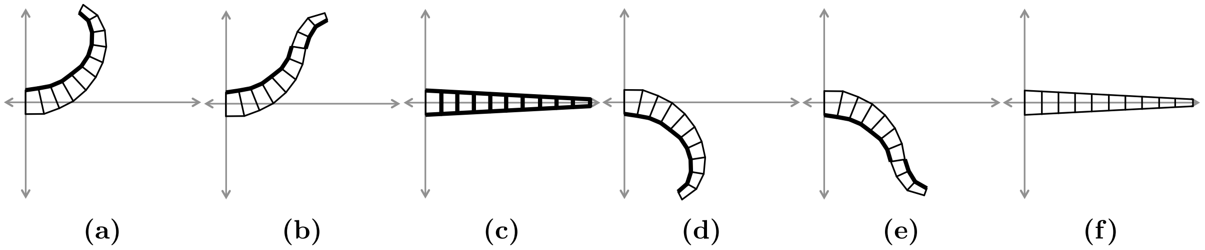


Figure 6.1: Octopus arm actions. While there are theoretically many other combinations of muscle contractions possible, Engel et al. (2006) limited their model to these six to make the domain tractable. Line thickness indicates the strength of the contractive force applied.

action space. Furthermore, unlike any system before, the learned controller is asked to scale to even larger arms *without further learning*.

The octopus arm problem is thus an ideal domain for demonstrating the ability of indirect encoding to transcend such superficial dimensionalities. After all, the underlying kinematic control principle is similar regardless of the precise number of segments (i.e. muscles and sensory inputs), suggesting that an approach that is sensitive to its particular dimensionality is missing something fundamental. This assertion is evident if we consider how the contractive patterns shown in figure 6.1 can be applied to support arms with more segments. To demonstrate this point, an indirect encoding called *Hypercube-based NeuroEvolution of Augmenting Topologies* (HyperNEAT) in this chapter evolves a *description* of how the weights of a neurocontroller relate to each other across the domain geometry irrespective of the arm's precise physical dimensionality (D'Ambrosio and Stanley, 2007; Gauci and Stanley, 2007, 2010; Stanley et al., 2009). This approach means that the HyperNEAT controller actually learns to articulate all the muscles independently without the need to partition the action space up front. Furthermore, as should be the case in learning such problems, the indirect encoding learns controller solutions in equivalent time regardless of the number of segments in the arm, and when transferred *without further training*, solutions evolved on smaller arms retain the fundamental motion model because they simply extend the general kinematic concepts discovered at the original size.

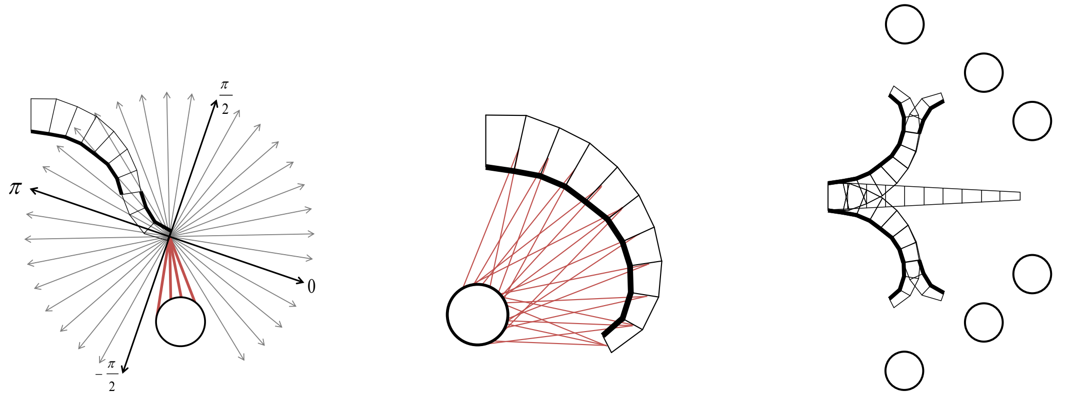
While this work clearly demonstrates the importance of indirect encoding in complex control domains, the experiments are ultimately guided by the fitness function and are consequently subject the deleterious effects described in Chapter 3. In this case, the seemingly successful results may nevertheless be flawed because evolution followed the objective-gradient and failed to establish genuine kinematic principles. Thus, the purpose of this chapter is to introduce the challenge domain in which the NA-IEC approach will be applied to demonstrate later (in the next chapter) how serendipitous discovery via a human-led search can produce better solutions more quickly (i.e. in fewer evaluations) than traditional objective-based approaches. The details of the simulated octopus arm domain are presented next.

6.1 Simulated Octopus Arm

The simulation domain in this work, based on Yekutieli et al. (2005), models the kinematics and dynamics of a two-dimensional *muscular hydrostat*, which is the mechanism of the octopus arm (Kier and Smith, 1985), as a chain of quadrilateral polygons with fixed area connected to a fixed base. The model constructs arms based on length (l), width (w), taper (t), mass (m), and number of segments (n). At the vertex of each quadrilateral is a point mass shared by adjacent segments. The *dorsal*, or upper, and *ventral*, or lower, edges of each segment represent longitudinal muscles while the vertical edges between sections represent transverse muscles. The muscles, modeled as spring-joints, are contracted by increasing the spring constant and relaxed by reducing the spring constant.

The fixed size and incompressible nature of the arm are the key features that enable the dynamic motion of the muscular hydrostat. These attributes are modeled by adjusting each segment's internal pressure: as external forces act to compress a segment, pressure increases; conversely, as forces stretch and expand the segment, internal pressure decreases. Thus segments change shape to restore the equilibrium between surface tension and internal pressure. Figure 6.1 shows the six basic actions utilized in Engel et al. (2006) as examples of the model's motion effects. However, in this work, the neurocontroller has independent control of all $3n$ muscles in the arm, creating a high-dimensional action space.

Because these experiments involve moving towards a perceived object, unlike Engel et al. (2006), the arm state is defined by sensor inputs that allow the controller to infer the position of each segment relative to the target. Range sensors along the arm provide cues about target position. Each sensor at each segment produces 36 radial distance measurements across the range $[-\pi \ldots \pi]$ (figure 6.2a), allowing the target to be seen by multiple beams simultaneously, especially as the sensor approaches the target or as sensor resolution is



(a) Target Sensor Operation **(b)** Target Perception **(c)** Target Placement

Figure 6.2: Target perception and placement. The arm controller perceives the target through range sensors (a) placed at each segment along the arm; the combined effect is shown (b) with non-detecting beams removed for clarity. The training targets in this work (c) are positioned beyond the reach of the simple actions in figure 6.1.

increased. Thus the $36n$ total beams create a high-dimensional input space. Figure 6.2b illustrates the arm's view of the target with the non-detecting beams removed for clarity.

This adaptation of the octopus arm domain (Woolley and Stanley, 2010) has a state-space with $36n$ inputs and and action-space with $3n$ outputs, an arbitrarily complex construction meant to overwhelm traditional machine learning methods. The point is that there are clear geometric relationships that, if preserved, allow the learning approach to focus on the conceptual problem without becoming distracted by the size of the unprocessed state and action space. The next section presents a description of HyperNEAT, the indirect encoding that makes it possible to evolve such high-dimensional controllers without the need to shield the learner from the true dimensionality of the space.

6.2 Hypercube-based NeuroEvolution of Augmenting Topologies

Neuroevolution can produce solutions for a broad array of control tasks (Floreano et al., 2008; Stanley and Miikkulainen, 2002, 2004; Yao, 1999). Many such methods are based on *direct encodings*, which means each piece of structure in the phenotype is encoded by a single gene that is an independent parameter. The ability to tune each parameter independently has kept traditional neural network learning approaches like backpropagation (Rumelhart et al., 1986) and neuroevolution (Stanley and Miikkulainen, 2002; Yao, 1999) from incorporating the domain geometry into the representation. However, as these approaches are applied to high-dimensional domains (e.g. the octopus arm), tuning hundreds of connections independently becomes an intractable task. The problem with direct encoding is that the discovery of repeating motifs is expensive and improbable, a problem that becomes obvious when clear and meaningful relationships exist in the problem domain. Therefore, indirect encodings (Bongard and Pfeifer, 2003; Hornby and Pollack, 2002; Stanley et al., 2009; Stanley and Miikkulainen, 2003) have become a growing area of interest in evolutionary computation.

One such indirect encoding designed explicitly for neural networks is the *Hypercube-based NeuroEvolution of Augmenting Topologies* (HyperNEAT; Gauci and Stanley, 2007; Stanley et al., 2009) approach. Rather than expressing link weights as distinct and independent parameters in the genome, HyperNEAT allows them to vary across the phenotype in a regular pattern. The pattern of connection weights is described by an encoding called a *compositional* *pattern producing network* (CPPN; Stanley, 2007), which is evolved by NEAT (Stanley and Miikkulainen, 2002, 2004).

Recall that images evolved in Picbreeder (Secretan et al., 2011, 2008) were also represented by CPPNs. In HyperNEAT, the CPPN is extended to encode the pattern of connection weights across a neural network. Just as a CPPN with the inputs x and y can paint an intensity pattern over a two-dimensional space, a CPPN with the inputs x_1, y_1, x_2 and y_2 can paint a pattern across a four-dimensional hypercube. In this way, the point (x_1, y_1, x_2, y_2) in the hypercube effectively encodes the weight of a connection between the two-dimensional points (x_1, y_1) and (x_2, y_2). Thus an ANN can be constructed wherein each of its nodes has a position in a geometric space and whose connection weights are thereby set by querying the four-dimensional CPPN with the position of each connection's endpoints (figure 6.3). This fact is significant because connection weights are computed with *knowledge* of the domain geometry.

As a rule of thumb, nodes are placed in a geometric space called the *substrate* to reflect the geometry of the domain, i.e. the state (Clune et al., 2009; Gauci and Stanley, 2010; Stanley et al., 2009). For example, a visual field can be laid out in two dimensions such that nodes that receive input from adjacent locations in the image are literally adjacent in the network geometry (Gauci and Stanley, 2007). In this way, HyperNEAT preserves knowledge of the domain geometry that is then exploited by the the CPPN to encode regularities like adjacency and symmetry; such regularities are invisible to traditional encodings that tune

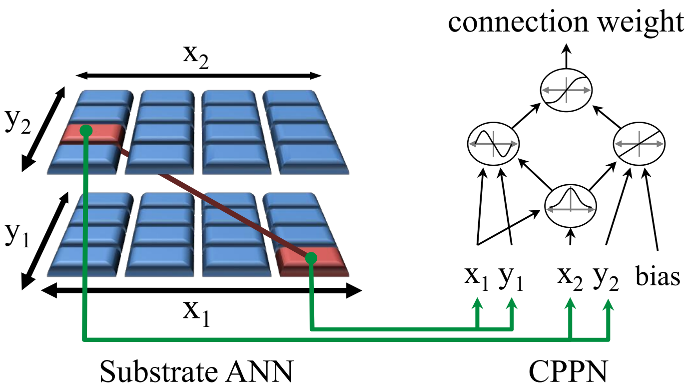


Figure 6.3: Encoding the connectivity pattern. The four-dimensional CPPN encodes the connectivity pattern of the substrate through an evolved network of geometric functions. The substrate ANN is generated by querying the CPPN for the value of each potential connection from (x_1, y_1) to (x_2, y_2). In this way, CPPNs capture patterns and regularities in domain geometry.

each parameter independently. This capability is exploited in the scaleable arm control architecture introduced next.

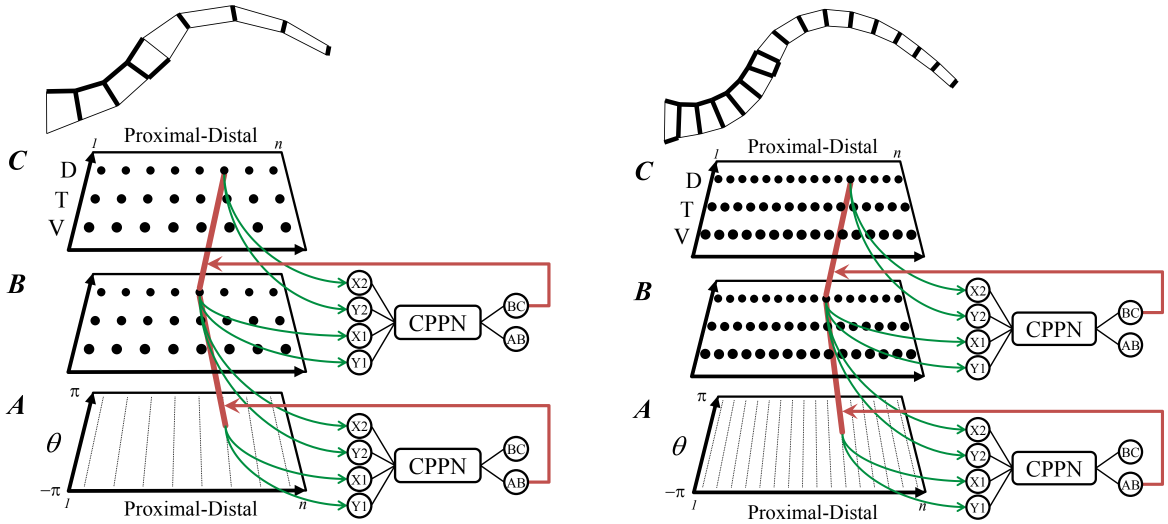
6.3 Scaleable Arm Control Architecture

The octopus arm control architecture presented here leverages the ability of HyperNEAT to encode regularities in the domain geometry and bypass superficial complexity. The architecture shown in figure 6.4 closely couples sensing to acting. The input layer accepts sensor data directly from the range finder sensors at each segment and the output layer provides the contractive response for each muscle. Finally, a hidden layer is provided to support nonlinear operation required by the gravity and buoyancy effects acting on the arm.

To represent the sensor array described above, the controller must interpret 36 rangefinder inputs per segment. The arm model is composed of segments that have a necessary order and relationship to the other segments in the arm, i.e. segment 1 connects to segment 2, segment 2 connects to segment 3, etc. Thus, the perception layer is constructed as a two-dimensional sheet with θ as one axis and the arm's proximal-distal (PD) geometry as the other (figure 6.4, layer A).

To represent the action space, the substrate provides an output for each of the $3n$ muscles in the arm. To take advantage of HyperNEAT's ability to leverage domain geometry, the proximal-distal axis of the sensor layer is mirrored by the output (contractive) layer. Furthermore, note in figure 6.1 how the dorsal and ventral muscles act together to form coordinated reaching behaviors. This configuration suggests aligning the dorsal, transverse, and ventral muscles along the proximal-distal axis (figure 6.4, layer C).

By viewing the substrate architecture as a feedforward network spanning from the sensor input layer (A), to the hidden layer (B), to the contractive output layer (C), a CPPN with inputs (x_1, y_1, x_2, y_2) and outputs (AB, BC) provides a complete encoding of the phenotype. Figure 6.4a illustrates how each of the 7,488 link weights in an eight-segment controller (left) are set by a single CPPN. The hope is that the principle that underlies moving a hydrostat is regular across the segments of the arm and therefore can be captured by the CPPN.



(a) 8-Segments **(b)** 16-Segments

Figure 6.4: Scalable neurocontroller architecture. The eight-segment substrate (a) can be scaled to a 16-segment substrate (b) without further training. The substrate at any resolution contains a two-dimensional input layer A that corresponds to the arm's sensory input, a two-dimensional hidden layer B, and a two-dimensional output layer C that controls the musculature. To query connections between A and B, the proximal-distal (PD) axis is the x_1 input and θ is the y_1 input. To query connections between B and C, the x_1 input is also PD, and DTV (Dorsal-Transverse-Ventral) is the y_2 input. Because the CPPN encodes kinematic principles, resampling with the node positions in (b) can produce a similar contractive pattern and arm pose (shown above).

By constructing the controller architecture (i.e. the substrate) to reflect the domain geometry (figure 6.4a), larger arm controllers are generated without further evolution by *requerying the same CPPN* at higher-resolutions (figure 6.4b). This approach works because adding segments to the arm is analogous to increasing the resolution of the hydrostat model. The CPPN, which encodes a continuous description of the arm's behavior policy, provides similar contractive patterns when applied to an arm with a different number of segments. The next section presents how CPPNs are evolved for this arm controller architecture.

6.4 Scaleable Arm Controller Experiment

The first aim of the experiment is to investigate the ability of indirect encoding to facilitate learning to control a hydrostat with dozens of degrees of freedom that are not restricted or pruned in any way. The second aim is to test the ability of an evolved CPPN to generate controllers for higher resolution arms without further training. From there, the goal of the challenge problem will be to improve upon the consistency and quality of the results learned by HyperNEAT through traditional objective-based search applied in this chapter.

The fitness function is designed to select controller behaviors that approach targets quickly. The simulator records the distance between the tip of the arm and the target at each timestep and calculates the average distance over a trial with a single target as:

$$d_{\text{avg}} = \sum_{t=0}^{t_{\max}} \frac{d_t}{t_{\max}}, \tag{6.1}$$

where d_t is the distance from the tip of the arm to the target surface at time t and $t_{\max}$ is the maximum number of timesteps in the trial. Individuals in the population are evaluated in six trials against six training targets (figure 6.2c) that are beyond the reach of the simple movements shown in figure 6.1. Because the goal is to reduce the average distance, fitness for a single trial can be expressed as:

$$f_{\text{trial}} = d_0 - d_{\text{avg}}^2, \tag{6.2}$$

where d_0 is the initial distance and squaring d_{avg} emphasizes early innovations that move towards the target by providing larger rewards for small improvements. Negative fitness values are set to zero and arms that succeed in touching the target with the tip earn a 25% bonus.

To validate that eight-segment solutions can scale, their evolved CPPNs are then requeried to generate controllers for arms with 10, 12, 14, 16, 18, and 20 segments with no further training. It is important to note that these dimensionalities are indeed high because they impact the necessary dimensionality of the corresponding *neurocontroller*, i.e. an eight-segment controller must set 7,488 connection weights while a 20-segment controller must set 46,800. Also, results cannot be compared directly to controllers trained by Engel et al. (2006) because they seek a single target blindly while those evolved here can actively touch targets at multiple locations based on sensory inputs.

6.5 Experimental Results

Figure 6.5 shows training performance over generations when controllers are separately evolved (i.e. not scaled) for arms with 8, 10, 12, 14, and 16 segments. Twenty runs were completed at each resolution. Remarkably, the number of degrees-of-freedom has *no significant effect* on the training curve, suggesting that indirect encoding really is making it possible to focus on learning the underlying *principle* independent of dimensionality.

Across all variants, CPPNs with an average of only 10.1 connections (stdev = 2.3) encode substrates with between 7,488 (8 segments) and 29,952 connections (16 segments), demonstrating the considerable compression of the indirect encoding.

The main scaling result (figure 6.6) is that the evolved contractive patterns transfer well from controllers trained on eight segments to arms with an increasing number of segments with no additional training. In the figure, the distance from the arm tip to the target surface is graphed over timesteps, demonstrating that controllers maintain the ability to approach targets as the physical structure scales; on average, even the 20-segment (worst) case approaches within 0.084 (±0.05 at 95% confidence) units of the target surface. It is important to note that the qualitative behavior of the arm at all scales in figure 6.6 is the same (i.e. they all still approach the target) although the speed of movement slows gradually and emergent physical characteristics begin to render the original solution less effective.

The sequence shown in figure 6.7 demonstrates a typical scaled reaching behavior. The contractive pattern shown was evolved as an eight-segment arm and applied to a 16-segment arm with no further training.

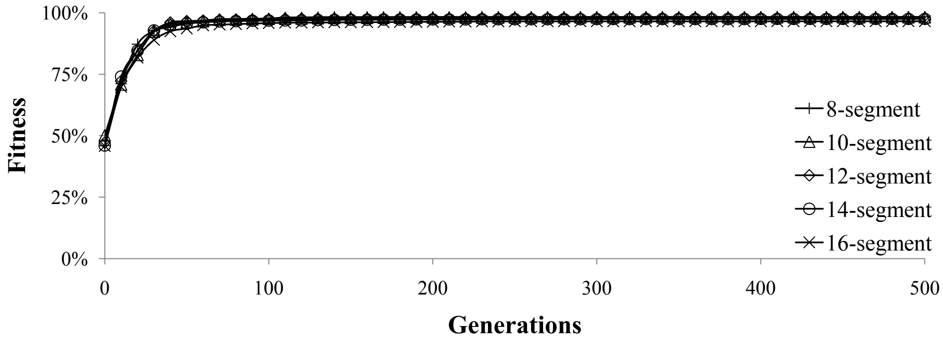


Figure 6.5: Training result: no performance degradation with size. HyperNEAT evolves neurocontrollers for arms with 8, 10, 12, 14 and 16 segments in equivalent time because the CPPN discovers the underlying kinematic patterns. Measurements are averaged over 20 runs; a score of 100% is the theoretical maximum fitness.

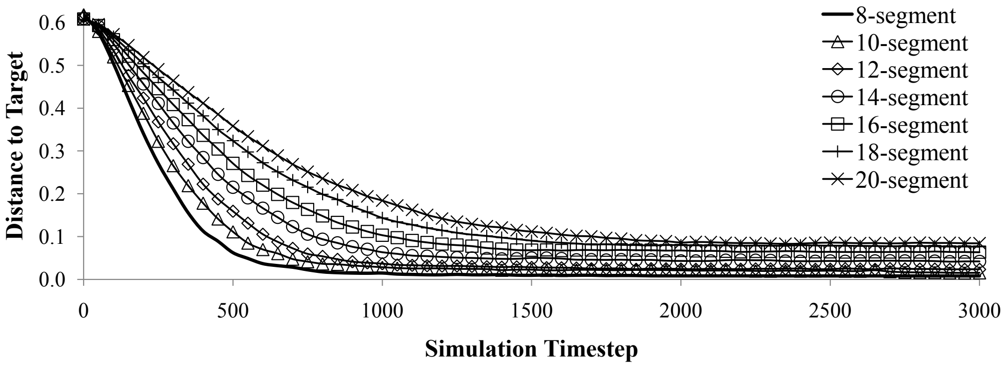


Figure 6.6: Scaling result: arms scale without further evolution. The distance from the target surface at each time-step is shown, demonstrating that the ability to move towards the target is preserved as controllers are scaled to support arms with additional segments. Measurements are averaged over 20 runs.

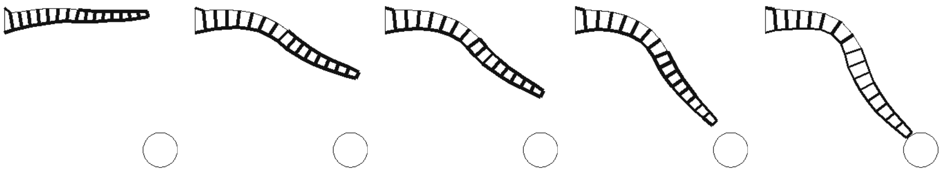


Figure 6.7: Typical reaching motion of a scaled hydrostat. This 16-segment controller is scaled directly from an eight-segment arm solution and illustrates how contracting the transverse muscles allows the arm to extend beyond its relaxed length. For videos of more such evolved and scaled behaviors see `http://eplex.cs.ucf.edu/octopusArm`.

6.6 Solving the Octopus Arm Faster

The original insight behind applying HyperNEAT to this domain is that finding solutions to problems in control should be about discovering an underlying principle and not about the number of dimensions in the action or state representation. Traditional approaches (Stanley and Miikkulainen, 2002, 2004; Sutton and Barto, 1998) map state information to effectors as if each is an independent dimension when in fact they are related. This traditional view of the problem domain ties complexity to the dimensionality of the physical domain, thus obfuscating the underlying concept. The octopus arm model is a good platform to showcase this idea because it can include an increasing number of segments. Some of the common ways that traditional approaches measure problem complexity are listed in table 6.1, all of which increase in scale with the number of arm segments.

Table 6.1: Measures of problem complexity. Traditional approaches are distracted by superficial dimensionality when they ignore fundamental relationships in the domain. Is problem complexity the number of dimensions in the state/action space? Is problem complexity the number of connections in the ANN controller? Failing to preserve key aspects of the domain geometry unnecessarily increases the difficulty of finding solutions.

Segments	(x, y, x', y')	Sensors	Muscles	ANN Connections
8	72	288	24	7,488
10	88	360	30	11,700
12	104	432	36	16,848
14	120	504	42	22,932
16	136	576	48	29,952
18	152	648	54	37,908
20	168	720	60	46,800

However, the experimental results demonstrate how an indirect encoding can evolve controllers for arms with 8, 10, 12, 14, and 16 segments in equivalent time, thus showing that the physical structure's dimensionality is a false measure of the domain complexity. By exploring the space of kinematic principles, the indirect encoding approach bypasses the increasing dimensionality of the physical structure. Similarly, the scaling results demonstrate that solutions evolved specifically for the eight-segment arm model captured fundamental kinematic strategies that apply directly to arms with additional segments.

For many problems, complexity is independent of the number of dimensions. The challenge is to transcend the distraction of superficial dimensionality by preserving meaningful relationships, e.g. geometric principles like order, orientation, and proximity. In this way, indirect encoding becomes and important consideration of any problem in which stat or action dimensionality may be misleading, or for learning scalable control policies. Whether it is a multi-segment arm, a robot hand that can add more fingers, or a centipede with a variable number of legs, indirect encoding shifts the problem away from the precise configuration towards the underlying principle, thereby opening up such problems to machine learning.

However,despite the successes in these experiments, the qualitative results nevertheless hide the subtle cost of objective-based evolution. While this traditional approach seems reasonable, the work in Chapter 3 shows the importance of establishing fundamental concepts early on. Just as the solutions to the Simple and Crescent image targets (figure 3.3) ultimately proved overly complex and inelegant compared to the original serendipitously discovered image, so may the evolved controllers in this chapter ultimately prove to be inefficiently represented by the CPPN when compared to what may be discovered by NA-IEC. The hope is that a human-directed search, interleaved with short-term novelty searches, can similarly evolve arm control behaviors more efficiently than traditional automated approaches. Thus the next chapter presents details of the culminating experiment wherein the ability of HyperNEAT with NA-IEC to evolve arm control behaviors is compared to HyperNEAT with novelty search and fitness-based HyperNEAT.

CHAPTER 7
CHALLENGE PROBLEM: SERENDIPITOUS DISCOVERY OF CONTROL BEHAVIORS

This chapter presents the culminating experiment wherein the benefits of serendipitous discovery are demonstrated in a high-dimensional control domain, i.e. the octopus arm domain. The *novelty-assisted interactive evolutionary computation* (NA-IEC) approach presented in Chapter 4 yielded significantly better results than novelty search alone in the deceptive maze domain by Lehman and Stanley (2008). Now the question becomes, how does human insight affect search when the proverbial "path to the goal" is not evident? The hope is that evolving octopus arm controllers in the NA-IEC framework will leverage highly-nuanced human intuitions to identify significant stepping stones *during* evolution, rather than articulating them explicitly a priori in a formal optimization function. The result is the discovery of principled controllers in fewer evaluations than both a traditional objective-based search and a non-objective search based on novelty.

7.1 Motivation

Up to this point, the capabilities of the NA-IEC approach have been demonstrated in the deceptive maze domain introduced by Lehman and Stanley (2008, 2011), a domain designed to confound traditional fitness-based approaches. Consequently, some may argue that the human user is providing an unfair advantage over both novelty search and traditional fitness-based search. In fact, this dissertation argues that the advantage of NA-IEC over automated EAs is that a human evaluator is indeed basing his selections on additional information. Interestingly, during this process, the human user does not know what information or behaviors he or she is looking for until one appears *serendipitously*, i.e. during evolution. In this way, the abilities of the human user to identify what behaviors are interesting and meaningful in a given domain as they emerge becomes a part of evolution and precludes the need to articulate and formalize sophisticated objective functions a priori. Thus the aim of this chapter is to apply the NA-IEC approach in a complex control domain where the set of solutions is not known nor are the stepping stones that lead to such solutions. The hypothesis is that, as in the maze domain, a human evaluator selecting candidates based on their expressed phenotypic behaviors will discover solutions more quickly than both traditional fitness-based search and novelty-search.

Recall that in the original octopus arm competition at the 2009 International Conference on Machine Learning (ICML 2009), the challenge required learners to map the 88-dimensional state space to the 32-dimensional action space such that the arm would learn to touch a single target with speed and efficiency. Interestingly, all participants opted to simplify the action space by mapping the state space to six pre-coordinated actions, a decision that was necessary to make the task tractable. In contrast, the experiments here and in the previous chapter evolve control behaviors without simplifying the action space a priori, i.e. the controller ANN sets each muscle independently. While this challenge domain was inspired by the original octopus arm competition at ICML 2009, the implementations are different and cannot be compared. Furthermore, the domain in this chapter is a variant (i.e. changed to support more realistic kinematic behaviors) of the simulated octopus arm domain presented in the previous chapter and thus cannot be compared directly to the previous chapter either.

As shown in Chapter 6, the original octopus arm domain from ICML 2009 was adapted so that the the target is *visible*, yielding a more challenging version of the task, which is to evolve a control behavior that can touch targets at arbitrary locations. This work demonstrated the ability of an indirect encoding to sidestep the superficial trap of dimensionality by focusing on evolving kinematic patterns for the arm independently of the number of segments. In fact, the experiment was so successful that the evolved controllers provided reasonable solutions when scaled up to larger arms *without further training.*

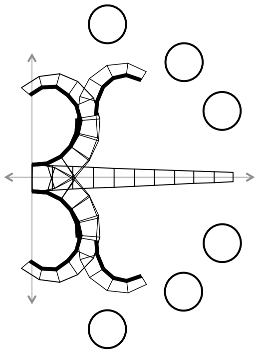
Although successful, evidence suggests that the pursuit of objectives creates deleterious effects, even when the objective function is satisfied (Chapter 3). Yet we regularly accept the solutions that emerge from such objective-driven searches as optimal because they achieve the objective, and may even do so consistently. Furthermore, some may still argue that the interpretation of the image evolution experiment (Section 3.2) and the maze navigation experiment (Section 5.2), are unfounded by blaming the results on a naive fitness function; the experiment in this chapter aims to dispel such doubts because the fitness-function in equation 6.2 was designed to provide a logical search gradient that rewards approaching the target and also encourages speed, i.e. it is a reasonable search heuristic and not a straw man. Therefore, the hypothesis of this experiment is that the human evaluator will recognize behavioral nuances, as in the maze domain, that lead to solutions more quickly even in a complex control domain where the “path to the goal” is unclear.

7.2 Octopus Arm NA-IEC Experiment

The intent of this experiment, as in the proof-of-concept (Chapter 5), is to compare the differences between solutions evolved by the NA-IEC approach, an automated novelty search, and a traditional objective-based search. The experimental domain is the simulated octopus arm environment detailed in Chapter 6, wherein each evaluation consisted of six trials that attempted to touch training targets at different locations. The same approach to evaluation is applied for all runs in this work.

Due to changes in the arm model to make it more stable and realistic, the six training targets were repositioned to be within reach of the arm, but still beyond the reach of the simple arm positions introduced by Engel et al. (2006). The new target positions, which are applied in all runs, are shown relative to the arm in figure 7.1a and listed in figure 7.1b.

As in the previous chapter, the octopus arm controllers are encoded such that they have knowledge of the domain geometry, i.e. they are encoded with a variant of the *Hypercube-based NeuroEvolution of Augmenting Topologies* (HyperNEAT) approach introduced by Stan-



(a) Training Targets

Target	(x, y)	Radius
1	(0.40, 0.80)	0.1
2	(0.80, 0.60)	0.1
3	(1.00, 0.35)	0.1
4	(1.00, −0.35)	0.1
5	(0.80, −0.60)	0.1
6	(0.40, −0.80)	0.1

(b) Target Coordinates

Figure 7.1: Training targets. Following changes that improved the stability and kinematics of the octopus arm model in Chapter 6, the training targets were repositioned to be reachable, but not by the simple actions shown (a). With the arm base positioned at the origin, the location of the training targets is given by the table (b).

ley et al. (2009). In this chapter, the controller implementations are extended to include a *link expression output* (LEO) component. HyperNEAT-LEO is an extension of HyperNEAT introduced by Verbancsics and Stanley (2011) in which the ANN's expressed connectivity pattern (i.e. which connections are expressed) is evolved independently from the pattern of connection weights. The HyperNEAT-LEO-based controller configuration (Appendix A.3) is applied in all versions of the experiment.

7.2.1 Fitness Function

To evaluate fitness in the traditional objective-based runs and the NA-IEC optimization operation, this experiment includes the objective function introduced in Chapter 6. Recall that fitness is assigned as a function of the average distance, d_{avg}, from the arm tip to the target (equation 6.1) during each trial. In this way, the difference between the arm's initial distance and the average distance (equation 6.2) rewards controllers for approaching the target during a trial. Furthermore, squaring d_{avg} emphasizes the fitness assigned to behaviors that begin moving toward the target. As in the original experiment, any negative fitness values are set to zero and touching the target generates a 25% bonus.

7.2.2 Novelty Metric

In contrast to the objective-based experiment in Chapter 6, a behavior characterization for the octopus arm domain is now required to support the fully automated novelty search runs and the NA-IEC short-term novelty search operation. Although this experimentis the first time novelty search has been applied in the octopus arm domain, previous work with novelty search by Lehman and Stanley (2011) has shown that, in most cases, it is reasonable to characterize a motion trajectory simply by its end-point. While some behaviors that reach the same point in different ways will be conflated, this approach emphasizes the discovery of behaviors that move into new positions. While the aim of this experiment is to discover control behaviors that approach a set of six training targets, each trial is independent because the arm position is reset to the initial position. Consequently, the overall controller behavior can be effectively characterized as six independent reaching motions, which are recorded as the final position of the arm at the end of *each* trial. In this experiment, the final position (x, y, θ) is the Cartesian coordinate (x, y) at the tip of the arm, along with θ, the angle of the centerline at the tip of the arm.

For consistency, the behavior space is defined with respect to the target. Hence a relative position $(\Delta x, \Delta y, \Delta\theta)$ is computed for each trial such that

$$(\Delta x, \Delta y, \Delta\theta) = (x_{\mathrm{t}} - x, y_{\mathrm{t}} - y, \theta_{\mathrm{t}} - \theta), \tag{7.1}$$

where $(x_{\mathrm{t}}, y_{\mathrm{t}})$ is the position of the target and θ_{t} is the angle from the arm tip to the target,

$$\theta_{\mathrm{t}} = \tan^{-1} \frac{y_{\mathrm{t}} - y}{x_{\mathrm{t}} - x}. \tag{7.2}$$

Thus the resulting coordinate $(\Delta x, \Delta y, \Delta\theta)$ is the behavior characterization for a single trial wherein the target behavior is effectively located at the center of the space. Finally, combining such characterizations for each of the six trials results in an 18-dimensional behavior space. In this way, arm controllers that move toward the training targets are distinct (i.e. in a different region of the behavior space) from those that are repelled by targets or from those that only approach a single target. Thus the difference between two distinct candidate behaviors, a and b, is computed as the Euclidian distance $\mathrm{dist}(a, b)$ from one to the other in the behavior space such that

$$\mathrm{dist}(a, b) = \sqrt{\sum_{i=1}^{n} (b_i - a_i)^2}, \tag{7.3}$$

where a_i and b_i are corresponding dimension in the n-dimensional behaviors being compared.

To score the novelty for a candidate α, this experiment computes the sparseness $\rho(\alpha)$ by equation 2.1, wherein the average distance to the k-nearest neighbors (Bentley, 1975; Cover and Thomas, 1991) provides a measure of how novel that behavior is compared to those seen previously. In this way, an individual that breaks into a sparse area of the behavior space is rewarded for being novel.

The novelty parameters for this experiment include the k-nearest neighbor value of 15 and a novelty threshold, ρ_{min}, starting at 3.0. The ρ_{min} value is then adjusted after every 2,500 evaluations, either by being increased by 20% (if more than four members enter the archive) or reduced by 5% (in the case that no new behaviors were added to the archive).

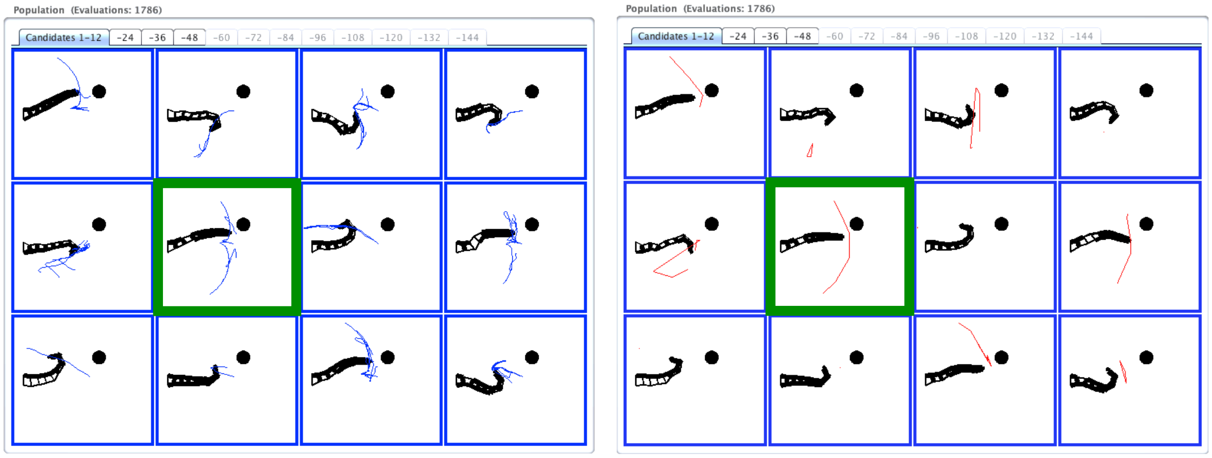
7.2.3 Human Evaluator Interface

As in the maze navigation experiment from Section 5.2, the human evaluator is presented with an onscreen population of candidate behaviors. From here the human user can select those candidates that they feel will be stepping stones to more interesting behaviors. The user then elects one of three evolutionary operations: (1) a traditional IEC step, (2) a short-term novelty search, or (3) an optimization search. The key difference in this experiment is that candidate controllers are evaluated as *animated behaviors*, which makes human fatigue a significant issue.

During initial development of the system, it became evident that evaluating a population of twelve candidates performing six trials each was too difficult. An attempt was made to keep the trials synchronized, but the task was still cumbersome. Another alternative was to evaluate behaviors individually, but the evaluation of just twelve candidates took ten minutes. In addition to the effort required to evaluate the onscreen population, the time required to generate the next population interrupted the human evaluator's train of thought. Although the NA-IEC approach does filter out common behaviors (i.e. solutions that are below the novelty threshold), the user must still to evaluate a significant number of candidates.

Therefore, to assist the human evaluator and ease fatigue, the NA-IEC framework provides the option of superimposing the path of the arm tip during all trials (figure 7.2a) and/or the behavior characterization (figure 7.2b) over the animation. In this way, such depictions provide a visual summary of a behavior, thus allowing the human evaluator to rapidly identify promising candidates before deciding which solutions to inspect more closely. The impact of this approach is evident in figure 7.2, wherein an innovation that led to a solution is made apparent by the behavior summary that is superimposed over the animated behavior.

The power of the NA-IEC approach lies in the ability of a human evaluator to detect important discoveries as they emerge and without regard to fitness. The idea is that a domain expert working with NA-IEC can help to direct the search more effectively than an automated approach alone. To fill the role of the domain expert in this experiment, I take the



(a) Arm Tip Trajectory **(b)** Behavior Characterization

Figure 7.2: Summarizing arm reaching behaviors. To reduce fatigue, a summary of the arm behavior can be overlaid to help identify promising candidates: the arm tip trajectory (a) during all trials provides an indication of how the arm responds to targets at different positions while the behavior characterization (b), which is used by novelty search, connects the position of the arm tip at the end of each trial. The user can also view both simultaneously (not shown). Summarizing the overall behavior in this way reduces fatigue by drawing attention to significant discoveries—one such innovation is highlighted in the population shown.

role of the human evaluator. Thus this experiment provides initial evidence for how such a coupling of human and automated search can perform. An important goal for future work will be to perform a full study with multiple human users, as was performed in the maze domain in Chapter 5. The experimental results are presented next.

7.3 Experimental Results

In this experiment, a controller that touches all six targets is considered a solution and is awarded the maximum fitness score. The main result is that the NA-IEC approach found solutions in significantly fewer evaluations than both the fitness-based runs and the runs guided purely by the novelty metric. Interestingly, fitness-based search found solutions consistently in this domain while pure novelty search did not, suggesting that this is one domain where novelty search suffers from an intractably large set of possible behaviors to discover. However, that human insight combined with novelty search outperforms fitness alone suggests both that human insight is in fact providing evolution with valuable clues about what is important, and that novelty search is still useful in such a domain if it can be guided occasionally to promising areas through human intuition.

Figure 7.3 compares the training performance over evaluations for the NA-IEC approach, the fitness-based search, and the novelty search in the octopus arm domain, averaged over 20 runs. The user-directed evolution with NA-IEC discovered solutions in an average of 3,165 ($\text{sd} = 2,424$) evaluations. These results are significantly ($p < 10^-8$; Student's t-test) faster than fitness-based evolution (8,860 evaluations, $\text{sd} = 2,300$).

Unlike in the previous experiments, the complexity of the solutions discovered in this domain during the human-led NA-IEC sessions were more complex (21.8 connections, sd = 3.6) than those discovered by fitness-based evolution (18.3 connections, sd = 2.9), suggesting that

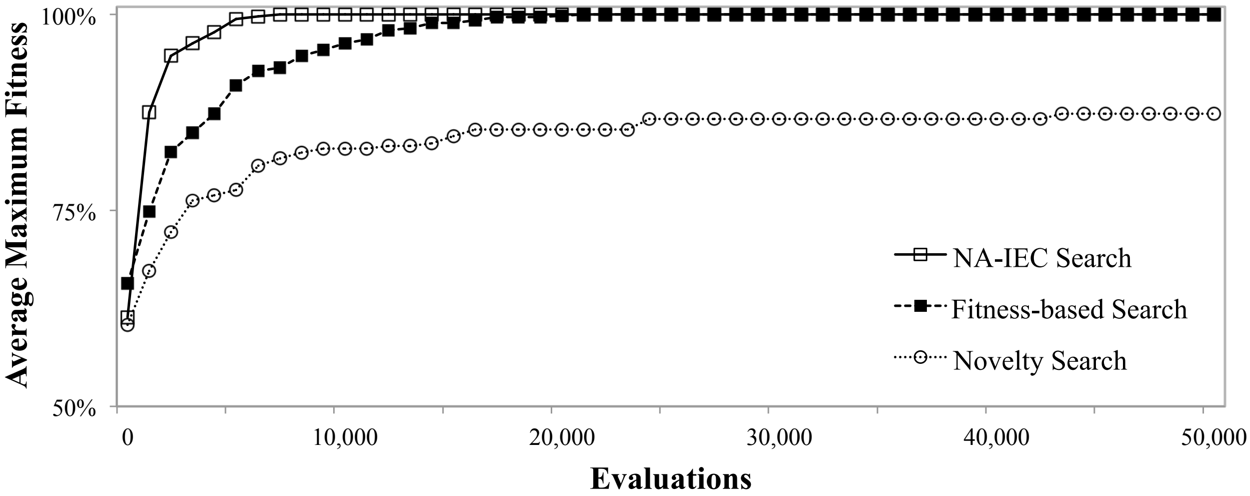


Figure 7.3: Comparing evolutionary learning rates in the octopus arm domain. The average maximum fitness over evaluations for HyperNEAT with NA-IEC, fitness-based HyperNEAT, and HyperNEAT with novelty search is shown. The main result is that runs led by the human evaluator discovered solutions significantly faster than fitness-based runs, while runs guided by novelty search failed to evolve solutions (results are averaged over 20 runs). These results demonstrate the advantage of NA-IEC over automated approaches. In NA-IEC, the human user directs evolution by selecting key innovations *during* evolution.

the representation of solutions discovered through NA-IEC is not necessarily always more compact. However, it is still possible that the representation is better organized, i.e. such that further evolution to perform additional tasks would work better starting from NA-IEC solutions. This hypothesis requires further research to investigate.

Analysis of the NA-IEC usage metrics shows that the human evaluator made an average of 28.9 (sd = 18.9) choices, applying the *Step* operation 10% of the time, the *Novelty* operation 46% of the time, and the *Optimize* operation 44% of the time. Interestingly, the time at which particular operations were applied reveals that the user began his searches with the *Novelty* operation and concluded by tuning his discoveries with the *Optimize* operation (figure 7.4).

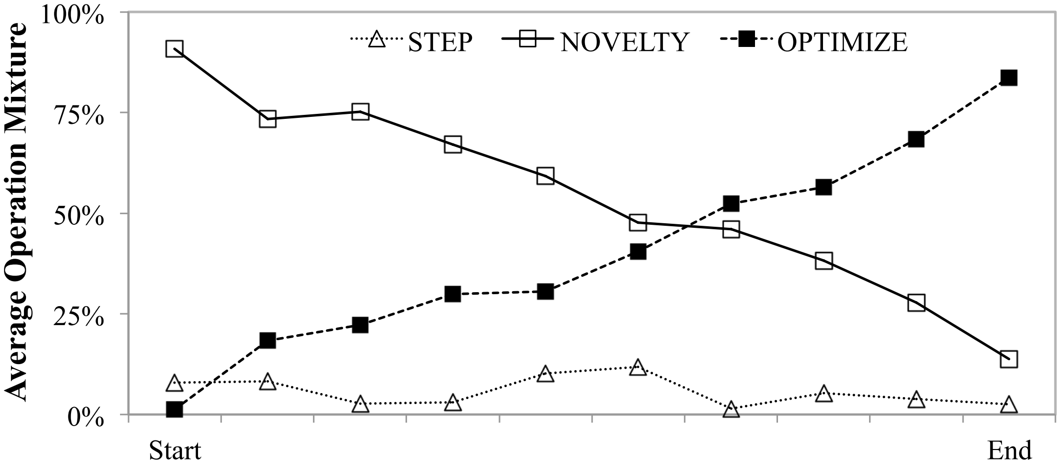


Figure 7.4: NA-IEC operation usage analysis. The NA-IEC user interface allows the human evaluator to select the mode of evolution, i.e. traditional IEC step, short-term novelty search, or fitness-based optimization. Plotting the mixture of operations as evolution progresses shows that the user favored the *Novelty* operation early in evolution, but then choose the *Optimize* operation increasingly more often as evolution progressed. In this way, the human evaluator can began by searching agnostically for a solution and then tuned what he discovered for a specific task.

7.4 Implications

The experimental results in this chapter establish several implications that result from allowing a human evaluator to direct evolution. One implication, as in the maze navigation experiment in Chapter 5, is that human users can take in a rich view of the behavior space and apply a deep understanding of what is meaningful in a given context. Another implication is that naive fitness and novelty functions become useful because the human evaluator is more able to detect and avoid deception in the domain. Furthermore, human insights detected not only promising candidate solutions, but also directed the mode of evolution by electing to execute a traditional IEC step, a short-term novelty search, or fitness-based optimization depending on the effect that the user felt would benefit evolution most at a given time.

While IEC approaches have been applied primarily in domains like art and music where the selection criteria is subjective or difficult to define, the experiment in this chapter demonstrates that leveraging the breadth of human experience can aid the design and discovery of controller solutions. Additionally, these results point to a deficiency with objective functions that is frequently overlooked: that strict mathematical constructs are unlikely to detect and reward subtle aspects of the domain. For example, automated EAs are frequently applied in challenging domains like biped walking and candidates are scored by an overly simple metric like distance traveled. While a well-formed walker would score well by such a metric, the misconception is that this same metric also establishes the intermediate solutions needed to create a successful biped walker. Evidence from Chapter 3 argues that this assumption is incorrect. But what metric would be better? In contrast to the established paradigm, work with NA-IEC demonstrates that a human user can detect significant discoveries *during* evolution, thus precluding the need to craft a more intricate fitness function. In fact, the ability of the human user to see and understand the domain provides experience and knoledge that would be difficult or impossible to formalize. Remarkably, all of this insight is gained from the evaluation of the expressed phenotype behaviors and without knowledge of the underlying implementation or representation.

Under the guidance of the human evaluator, short-term evolutionary operations can now employ simple fitness functions and novelty metrics that might not succeed on their own. Although the automated runs based on novelty search failed to produce solutions for the octopus arm domain, the very same novelty configuration was implemented in the NA-IEC framework. Normally, the failure to evolve solutions would invite criticism of the behavior characterization, the experimental parameters, and perhaps even the idea of searching for novelty at all. Interestingly, usage metrics from the NA-IEC runs show that the short-term novelty search operation played a pivotal role in the discovery of solutions. In fact, the *Novelty* operation was applied almost exclusively at the beginning of evolution, while optimization was applied only after the general form of the solution was established. Thus by allowing human insight about what is meaningful to guide the direction of evolution, simple fitness and novelty metrics that would normally be subject to deception when used alone become effective under the direction of a human evaluator.

Another advantage that the NA-IEC approach has over traditional evolutionary approaches is that the human evaluator can change the mode of evolution to create the desired effect given the state of evolution. Under a traditional fitness-based search evolution follows, albeit stochastically, the fitness gradient. Similarly, a pure novelty search can only reward novel behaviors. Efforts in Novelty-Based Multiobjectivization (Mouret, 2011), wherein one objective is the fitness function and the other is the novelty metric, attempted to provide a construct that could follow an objective gradient while avoiding local optima by simultaneously rewarding novelty. While this approach may work in some cases, the desired effect is more inherent in the NA-IEC approach because not only does the human evaluator bring a perspective on what is reasonable in the domain, but he or she also possesses a sense of how evolution is progressing as a whole.

Thus far NA-IEC has been demonstrated in two domains that have clear and definite objectives. However, the NA-IEC approach was inspired by the success of Picbreeder (Secretan et al., 2011, 2008), wherein surprising and meaningful artifacts were discovered that no one knew existed—yet they were discovered. Furthermore, users continue to evolve interesting and meaningful images in Picbreeder without having a specified objective. Following this inspiration, the hope for the NA-IEC approach is not that it will displace or outperform automated evolutionary methods, but that it will lead to the discovery of agent behaviors in unbounded domains that are as intriguing as the images that emerged serendipitously from Picbreeder.

7.5 Conclusion

The experiment presented in this chapter compared the performance of the NA-IEC approach (Chapter 4) to a traditional fitness-based search and a pure novelty search in a challenge domain (i.e. the octopus arm domain). The main result was that NA-IEC, guided by human insight, evolved solutions in fewer evaluations than a fitness-driven approach that was previously successful. While the performance increase can be attributed to human insight, the influence of the user went beyond recognizing important stepping stones as they emerged by also providing insight about how to direct the mode of evolution. Furthermore, although novelty search alone did not discover solutions consistently, a human user applied the very same novelty metric during NA-IEC runs as a means of discovering meaningful reaching behaviors. Once the general form of the solution was established, the user employed traditional fitness-based optimization to tune their discoveries. The implication is that combining the rich experiences of a human evaluator with automated tools creates a synergistic effect wherein the limitations of the user (Takagi, 2001) are offset by the ability of automated tools to generate novel stepping stones or optimize well-formed solutions. In return, the perspective of the human evaluator protects fitness-based approaches from the effects of deception and keeps novelty search from becoming lost in the space of what is possible. The next chapter presents a discussion of the contributions of the NA-IEC approach and its implications for field of EC.

CHAPTER 8
DISCUSSION

Natural evolution has no *final* objective, yet from it emerged serendipitous innovations like hemoglobin, flight, sonar, the eye, and the human brain. It is remarkable that each is built upon other innovations that are interesting in their own right. Such is the ratcheting process of evolution, in which stepping stones are laid agnostically, without regard to any global objective. In this way, innovations that have a meaningful purpose, function, or behavior become the foundation for future innovations. Through this ongoing process of stepping stones built upon stepping stones, a great diversity of innovations were discovered (Dawkins, 1986) that continue to inspire the greatest of our engineering ambitions.

In contrast, the reigning paradigm in evolutionary computation (EC) evolves solutions for a particular engineering requirement or control task. Under this approach, the evolutionary algorithm searches for solutions that satisfy the objective function (e.g. approaching a target while minimizing time, energy, or cost). When evolutionary algorithms (EAs) fail to solve complex tasks, the traditional view is that the task is too demanding and that the poor performance of all of the individuals traps the evolutionary search in a deceptive region of the solution space (Gomez and Miikkulainen, 1997). Yet the deeper issue is that fruitful regions are difficult to discover because the objective gradient does not reward the stepping stones necessary to reach such areas. Supporting this argument, Chapter 3 exposed the deleterious effects of setting objectives. The reason for this problem, as in nature, is that the stepping stones necessary to reach ambitious ends *do not* resemble the final result; only in hindsight do we see the value of any particular stepping stone. Thus when search is guided by an objective function, the results (even those with high fitness) are piecewise constructions that fail to capture fundamental knowledge about the domain.

Projects like Picbreeder give some insight into how fruitful regions can be discovered without knowing of their existence a priori. The trick is that searching for what is interesting, i.e. searching without expectations, unlocks the ability to make serendipitous discoveries. In this way, fruitful areas are discovered because the innovations that led there encoded meaningful information about the domain. From these well-formed stepping stones, just a few mutations can elaborate on an established theme to create new innovations.

Such insights provoke deep consideration into how the EC community should approach future problems. This consideration is not a call to abandon optimization. Rather, the field needs to incorporate both non-objective and objective-based techniques into the proverbial toolbox. Experience with Picbreeder demonstrated the power of serendipitous search to find meaningful images in the vast and desolate space of possible images. In fact, many of the artifacts discovered in Picbreeder far surpass what was expected to emerge. Such insights suggest that we have been limiting ourselves in our endeavors because we cannot articulate the appropriate path to an ambitious goal without knowing (at least in general) the solution a priori. Thus the argument here is that serendipitous approaches, traditionally reserved for aesthetic domains like art and music, can also benefit evolving agent behaviors without regard to any final objective. Perhaps what emerges will also surpass what was thought possible for a given control domain.

If serendipitous search does indeed provide a means of discovering fruitful regions of agent behaviors, then the field of EC may be able to simulate the powerful ratcheting process in natural evolution to break through the glass celling imposed by the objective-function and create significant solutions to the problems that have eluded us.

The *novelty-assisted interactive evolutionary computation* (NA-IEC) approach presented in Chapter 4 establishes such an interactive framework for evolving agent behaviors, thus allowing evolution to escape the constraints of the fitness gradient. However, simply allowing a human evaluator to explore the space of behaviors is a cumbersome task. Rather, to reach ambitious goals, the single-user IEC system (operating with interleaved novelty search) provides inspiration about what is possible by generating the on-screen IEC population from a short-term novelty search. In this way, the user is presented with a diverse collection of behaviors that are based on what they previously considered interesting. Such an approach, operating without regard to any explicit objective or fitness-function, provides the opportunity for serendipitous discovery. The premise is that successive selections of what is interesting (based on highly-nuanced human intuitions) will discover effective behaviors with meaningful representations more quickly than the traditional objective-based approach in EC.

Such assertions were initially demonstrated in Chapter 5 as a proof-of-concept in the deceptive maze domain (Lehman and Stanley, 2008, 2011), wherein the NA-IEC approach described in Chapter 4 not only provided the user with an initial population of various maze navigation behaviors, but continued to suggest alternative navigation behaviors based on what the user previously identified as important. In this way, interleaving short-term novelty searches with human selection events produced successful navigation behaviors in fewer evaluations than novelty search alone because novelty search, despite being better than fitness for the maze problem, is a broad-beam search of the the behavior space. That is not to say that humans know how to evolve maze navigation policies better than novelty search. In fact, the same task would be much more difficult under a traditional IEC approach. Thus the power of NA-IEC is that search is directed in part by human domain knowledge while innovations are generated by novelty search; it is their unification that makes NA-IEC effective.

Applying NA-IEC to evolve controllers for the octopus arm is a good follow-on experiment because the "path" to the solution is not obvious. In such a dynamic domain there are many arm behaviors that achieve high fitness, some of which were found by the objective-based search in Chapter 6. The fact that the EA did produce solutions with high-fitness scores only obfuscates the idea that better paths to the solution *do* exist, but what are they and how do we find them? The image evolution experiment in Chapter 3 provided insight into the deleterious effects of searching for a priori objectives on solutions, even when the solution criteria is met. Confirming again the benefit of a human in the loop, NA-IEC indeed discovers working arms faster than the fitness-based (or novelty-based) approach alone.

As a general lesson for EC, this work emphasizes the importance of departing the objective gradient, at least temporarily, to allow human domain knowledge to establish meaningful principles early in evolution. In this way, the NA-IEC approach evolves controllers based on what the human evaluator found interesting and not on a performance metric. Consequently, when agent behaviors are evolved under the NA-IEC approach, the resulting solutions actually contain good representations of the problem domain because they are shaped by what the user considers relevant for the task. Furthermore, when such principles are encoded in a meaningful way, future mutations are likely to change and elaborate the established principles in a coherent way, thus leading evolution more quickly into the fruitful regions that are difficult for fitness-based approaches to discover.

CHAPTER 9
CONCLUSIONS

The purpose of this dissertation was to establish the foundational concepts that motivate the need for serendipitous discovery in the field of evolutionary computation (Chapter 3), to introduce a new mode of evolutionary search called *novelty-assisted interactive evolutionary computation* (NA-IEC), and to establish the capabilities of this new evolutionary approach by demonstrating how it improves the discovery of solutions. The capabilities of serendipitous discovery are established by two experiments, the proof-of-concept (Chapter 5) and the culminating experiment (Chapter 7), which compared the NA-IEC approach to pure novelty search and objective-based search.

The premise of this work is that evolutionary algorithms that are guided by an objective function have a negative effect on the resulting solutions—even when the objective is reached. The first contribution (1) is a unique experiment presented in Chapter 3 in which images evolved through serendipitous walks through the space of images become the explicit objective of the *same algorithm* that discovered them. The reported result is that objective-based evolution cannot re-evolve the images that it once discovered. Evolution failed to recreate most images at all, and even when it did recreate them, the failure was in the piecewise representation of the original image. These results call into question the credibility of pursuing a priori objectives in search overall.

The power of serendipitous discovery in Picbreeder, along with results from novelty search, inspired the second contribution (2), the NA-IEC approach presented in Chapter 4. In this approach, a human evaluator selects behaviors that are interesting or meaningful for a given domain without regard to a fitness-gradient. Additionally, combining interactive evolution with novelty search creates a synergistic effect in which evolution is directed by what the human sees as interesting, while an interleaved novelty search explores the area around the user's selection(s). In this way, the on-screen population contains a collection of innovative behaviors that are sufficiently novel, and if they are interesting to the human evaluator, may inspire the discovery of agent behaviors that could not have been known or specified a priori.

The third contribution (3) of this dissertation is a *proof-of-concept* that demonstrates how the NA-IEC approach leverages human insights and domain knowledge to evolve solutions faster than a traditional automated search. In fact, the NA-IEC approach even outperformed a waypoint-directed search, demonstrating that human evaluators are contributing insights that are beyond simply knowing the path that to the goal. The fourth and final contribution (4) is the *culminating experiment* in which control behaviors are evolved serendipitously in a non-trivial task domain, i.e. the octopus arm. In this experiment, the stepping stones required to evolve a solution are not obvious, yet human insights again enabled evolving solutions faster than automated approaches.

Conclusion

The significance of this work is that it demonstrated the deleterious effects of evolving toward an objective and presented a new mode of evolution capable of abandoning the objective gradient. Instead, it favors following the path of *interesting* behaviors (as perceived by humans) toward the discovery of meaningful controller solutions.

APPENDIX: PARAMETERS

This appendix presents the experimental parameters applied in this dissertation, each of which were based either on the NeuroEvolution of Augmenting Topologies (NEAT) approach introduced by Stanley and Miikkulainen (2002, 2004) or the Hypercube-based NeuroEvolution of Augmenting Topologies (HyperNEAT) approach introduced by D'Ambrosio and Stanley (2007); Gauci and Stanley (2007, 2010); Stanley et al. (2009). Recall that HyperNEAT only differs from original NEAT in the set of activation functions that are available. Thus this section first describes the evolutionary parameters for NEAT and HyperNEAT and then specifies the parameter values that were applied, either globally or specifically, in the experiments.

Population Size: The number of individual candidate solutions (i.e. individual hypothesis) maintained at any given time.

Generational Replacement Strategy: The evaluation and selection of individuals in population P_i result in the creation of new population P_{i+1}.

Steady-State Replacement Strategy: The least-fit (or the least-novel) individual is removed from the population to be replaced by a new individual.

Max Number of Evaluations: A halting criteria for the evolutionary search. Under a steady state replacement strategy, the number of evaluations is equal to the size of the initial random population plus the number of replacement events. Under a generational replacement strategy, the maximum number of evaluations is calculated as the number of generations elapsed multiplied the population size.

Survival Rate: The percentage of the population which survives and reproduces between each generation—this parameter is not active under steady-state replacement.

Add Node Probability: The probability that new node will be added by splitting an existing connection within the ANN.

Add Link Probability: The probability that a new connection will be added linking two nodes that were not connected previously.

Remove Link Probability: The probability that a connection will be removed; a possible consequence of such a mutation is that any orphaned nodes will be pruned as well.

Connection weights are mutated as a means of training the network, wherein evolution retains favorable mutations and eliminates degenerate ones.

Weight Mutation Rate: The probability that a connection weight will be mutated.

Weight Mutation StDev: The magnitude of a weight mutation.

Min Connection Weight: The minimum connection weight.

Max Connection Weight: The maximum connection weight.

Due to the various ANN structures that emerge under NEAT, recombination is only permitted between compatible individuals. Compatibility is determined evaluating the number of excess genes, the number of disjoint genes, and the difference between common connection weights—the significance of which is adjusted by the following compatibility coefficients:

Excess Gene Coefficient: Denoted as c_1, this parameter influences the effect of having excess genes on the compatibility of two networks.

Disjoint Gene Coefficient: Denoted as c_2, this parameter influences the effect of having disjoint genes on the compatibility of two networks.

Excess Gene Coefficient: Denoted as c_3, this parameter influences the effect of connection strength differences on the compatibility of two networks.

Compatibility Threshold: Denoted as c_t, this parameter sets the level of difference at which two networks are no longer considered compatible, i.e. they belong to different species.

Truncation Selection Strategy: Specifies that the most fit (or most novel) individuals be selected to reproduce while the rest are eliminated.

Roulette Selection Strategy: Specifies that individuals be selected to reproduce with the probability of being selected proportional to the individuals spectated fitness value (i.e. the individuals raw fitness score divided by the number of individuals in the same species).

Elitism: Specifies if the most fit (or most novel) individual should be copied into the next generation without mutation.

Min Elitism Specie Size: Specifies the minimum number of individuals that must exist in a species for elitism to occur. Note that the use of elitism under a generational replacement strategy reduces the number of individuals that can be selected for reproduction and recombination.

Initial Topology: Specifies the the number of hidden nodes.

Recurrency: Sets wether recurrent connections will be allowed or disallowed in the network.

All of the experiments presented in this dissertation were conducted using an adaptation of the *Another NEAT Java Implementation* (ANJI) package (James and Tucker, 2005–2010) augmented to support steady-state evolution, interactive evolution, and selection based on either the fitness score or the novelty score. Additionally, the ANJI package was extended to implement HyperNEAT, which includes the ability to evolve the CPPNs that indirectly encode larger networks. The following parameters were applied universally in all experiments: The speciation parameters were $c_1 = 1.0$, $c_2 = 1.0$, $c_3 = 0.3$, and $c_t = 0.2$. The survival rate was 20% with the roulette selection strategy and elitism (minimum species size = 1) options were enabled. The chance of adding a new connection was 10%, while the chance of loosing a

Table A.1: Parameter settings in NEAT-based experiments.

Parameter	Image Evolution	Maze Navigation	OctopusArm 1	OctopusArm 2
Population Size	150	250	100	100
Replacement Strategy	Generational	Steady-State	Generational	Steady-State
Max # of Evaluations	4,500,000	250,000	100,000	50,000
Add Node Probability	7%	5%	5%	5%
Weight Mutation Rate	20%	10%	75%	75%
Weight Mutation StDev	1.0	0.8	1.5	1.5
Min Connection Weight	-3.0	-5.0	-5.0	-5.0
Max Connection Weight	3.0	5.0	5.0	5.0
Initial Topology	1 hidden	0 hidden	0 hidden	0 hidden
Recurrency	disallowed	allowed	disallowed	disallowed

connection was 1%. For the experiments where NEAT evolved CPPNs, the available CPPN activation functions were sine, cosine, Gaussian, identity, and sigmoid. These parameters were based on the established parameters for NEAT (Stanley and Miikkulainen, 2002) and were found robust to moderate variation. Table A.1 presents the remaining evolutionary parameters for the experiments in this dissertation.

A.1 NA-IEC Implementation Parameters

While NEAT is the underlying EA for the NA-IEC implementation in this dissertation, NA-IEC run follow some different configuration details. First, in NA-IEC the minimum on-screen population size was at least 12, and no more than 144. Next, during the fitness-based optimization and short-term novelty operations, the background search population was expanded to 250 and then run for up to 500 evaluations. For the *Step* and *Optimize* operators, users expected to see similar behaviors with moderate variation, thus the weight mutation rate is 10% and the weight mutation power is within 0.1 standard deviations of the current weight.

A.2 Novelty Metric Parameters

The parameters specific to novelty were applied by all experiments and were based on the original deceptive maze navigation experiment (Lehman and Stanley, 2008, 2011). They include the nearest neighbors value ($k = 15$) and the novelty threshold, $\rho_{\min}$, which begins at 3.0 and is adjusted after every 2,500 evaluations.

A.3 HyperNEAT with LEO

In all versions of the experiment, controllers for the octopus arm are run under a variant of the HyperNEAT approach introduced by Stanley et al. (2009). Under the traditional implementation of HyperNEAT (including the implementation in Chapter 6), connections in the ANN were not expressed if their connection weight was less than 0.2. The original hope was that particular ANN encoding would suppress patterns of unnecessary connections within the resulting ANN. However, in practice HyperNEAT tends to express most every connection, electing instead to adjust the weights across the entire ANN. Thus, a 10 segment arm controller with a fully connected substrate has 11,700 connections; such fully-connected configurations tend to saturate signals and obscure subtle behaviors.

As an alternative, Verbancsics and Stanley (2011) extended HyperNEAT to include a *link expression output* (LEO) component. In this way, HyperNEAT-LEO is able to evolve the ANN's expressed connectivity pattern (i.e. which connections are expressed) independently from the pattern of connection weights. To accomplish this, the original CPPN that maps the four-dimensional point (x_1, y_1, x_2, y_2) to a connection weight w, is extended to include an additional output-node (the link expression output) that specifies if the connection at (x_1, y_1, x_2, y_2) should be expressed with the weight w. By convention, the LEO node activation function is a step-function with an output in the range $[-1, \ldots, 1]$ and connections are expressed if LEO > 0. Given the geometric properties of the octopus arm domain, the underlying neuroevolutionary algorithm in Chapter 7 is HyperNEAT-LEO because of its ability to express patterns for both link expression and connections with weights.

A.4 Domain Implementation Details

Picbreeder Image Evolution Like ANJI, Picbreeder is written in Java so the code could be combined to support the automated evolution of images previously discovered by users in Picbreeder (Secretan et al., 2008). Target images were rendered from the published CPPN chromosome at a resolution of 128 × 128 pixels, with each pixel having an 8-bit grayscale value. The gradient value of the image was then computed using the Java Advanced Imaging API (JAI). Next, the set of 16,384 grayscale pixel values and the set of 16,384 gradient values were combined to form the *target feature set*. Following the same process for a candidate CPPN evolved by NEAT, a corresponding *candidate feature set* is generated such that the degree of difference (Ralescu, 2003) can be measured for each of the 32,768 corresponding features. Finally, the fitness of a candidate image is computed from the mean squared error between the target feature set and the candidate feature set.

Deceptive Maze Navigation Domain: In order to run with ANJI, which is written Java, the maze navigation domain was recreated in Java based on the original source code released by Lehman and Stanley (2008) at `http://eplex.cs.ucf.edu/software/NoveltySearchC++.zip`. The domain implementation was validated as an equivalent implementation.

Octopus Arm Domain: The physical modeling environment for the octopus arm simulator is based on the Phys2D engine by Galss (2006–2008) at `www.cokeandcode.com`.

LIST OF REFERENCES

S. Ando and E. Suzuki. Distributed multi-objective ga for generating comprehensive pareto front in deceptive optimization problems. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 1569 –1576, 2006.

J. L. Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18:509–517, September 1975.

J. A. Biles. GenJam: A genetic algorithm for generating jazz solos. In *Proceedings of the International Computer Music Conference*, pages 131–137, 1994.

J. Blynel and D. Floreano. Exploring the t-maze: Evolving learning-like robot behaviors using ctrnns. In S. Cagnoni, C. Johnson, J. Cardalda, E. Marchiori, D. Corne, J.-A. Meyer, J. Gottlieb, M. Middendorf, A. Guillot, G. Raidl, and E. Hart, editors, *Applications of Evolutionary Computing*, volume 2611 of *Lecture Notes in Computer Science*, pages 173–176. Springer Berlin / Heidelberg, 2003.

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

J. A. Boyan and A. W. Moore. Generalization in reinforcement learning: Safely approximating the value function. In *Advances in Neural Information Processing Systems 7*, 1995.

D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, and E. Zitzler. Do additional objectives make a problem harder? In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, GECCO '07, pages 765–772, New York, NY, USA, 2007. ACM.

J. Clune, B. E. Beckmann, C. Ofria, and R. T. Pennock. Evolving coordinated quadruped gaits with the HyperNEAT generative encoding. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC-2009) Special Session on Evolutionary Robotics*, Piscataway, NJ, USA, 2009. IEEE Press.

C. Coello, G. Lamont, and D. Van Veldhuizen. *Evolutionary algorithms for solving multi-objective problems.* Springer-Verlag New York Inc, 2007.

M. Colombetti and M. Dorigo. Robot shaping: Developing situated agents through learning. Technical Report TR-92-040, International Computer Science Institute, Berkeley, CA, 1992.

T. M. Cover and J. Thomas. *Elements of Information Theory.* Wiley, Hoboken, NJ, 1991.

D. D'Ambrosio and K. O. Stanley. A novel generative encoding for exploiting neural network sensor and output geometry. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2007)*, New York, NY, 2007. ACM Press.

R. Dawkins. *The Blind Watchmaker.* Longman, Essex, U.K., 1986.

R. Dawkins. The evolution of evolvability. In C. G. Langton, editor, *Artificial Life*, pages 201–220. Addison-Wesley, Reading, MA, 1989.

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

K. A. De Jong. *Evolutionary Computation: A Unified Perspective.* MIT Press, Cambridge, MA, 2002.

K. Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 7(3):205–230, 1999.

K. Deb. *Multi-objective optimization using evolutionary algorithms.* John Wiley & Sons, Chichester ; New York, 2001. Kalyanmoy Deb. : Kalyanmoy Deb.; 50411 Includes bibliographical references (p. [471]-490) and index.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182 –197, Apr. 2002.

S. Draves. The electric sheep screen-saver: A case study in aesthetic evolution. In F. Rothlauf, J. Branke, S. Cagnoni, D. W. Corne, R. Drechsler, Y. Jin, P. Machado, E. Marchiori, J. Romero, G. D. Smith, and G. Squillero, editors, *Applications on Evolutionary Computing*, volume 3449 of *Lecture Notes in Computer Science*, pages 458–467. Springer Berlin / Heidelberg, 2005.

S. Droste, T. Jansen, and I. Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276(1–2):51–81, 2002.

A. Eiben and J. Smith. *Introduction to evolutionary computing.* Springer-Verlag, 2003.

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

Y. Engel, P. Szabo, and D. Volkinshtein. Learning to control an octopus arm with gaussian process temporal difference methods. *Advances in Neural Information Processing Systems*, 18:347–354, 2006.

S. Ficici and J. Pollack. Challenges in coevolutionary learning: Arms-race dynamics, open-endedness, and mediocre stable states. *Artificial life VI*, page 238, 1998.

D. Floreano, P. Dürr, and C. Mattiussi. Neuroevolution: from architectures to learning. *Evolutionary Intelligence*, 1:47–62, 2008.

D. B. Fogel. *Evolutionary Computation: The Fossil Record.* IEEE Press, New York, NY, 1998.

D. B. Fogel. *Evolutionary computation: toward a new philosophy of machine intelligence, third edition.* Wiley-IEEE Press, 2006.

K. Galss. The 2d game physics engine in Java. `http://phys2d.cokeandcode.com`, 2008.

J. Gauci and K. O. Stanley. Generating large-scale neural networks through discovering geometric regularities. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2007)*, New York, NY, 2007. ACM Press.

J. Gauci and K. O. Stanley. Autonomous evolution of topographic regularities in artificial neural networks. *Neural Computation*, 22(7):1860–1898, 2010.

D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, Reading, MA, 1989.

D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In J. J. Grefenstette, editor, *Proceedings of the Second International Conference on Genetic Algorithms*, pages 148–154. San Francisco: Kaufmann, 1987.

F. Gomez and R. Miikkulainen. Incremental evolution of complex general behavior. *Adaptive Behavior*, 5:317–342, 1997.

F. Gomez and R. Miikkulainen. Solving non-Markovian control tasks with neuroevolution. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pages 1356–1361, San Francisco, 1999. Kaufmann.

F. J. Gomez, J. Schmidhuber, and R. Miikkulainen. Efficient non-linear control through neuroevolution. In J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, editors, *Proceedings of the 17th European Conference on Machine Learning*, volume 4212 of *Lecture Notes in Computer Science*, pages 654–662. Springer, 2006.

L. Graening, N. Aulig, and M. Olhofer. Towards directed open-ended search by a novelty guided evolution strategy. In R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, editors, *Parallel Problem Solving from Nature – PPSN XI*, volume 6239 of *Lecture Notes in Computer Science*, pages 71–80. Springer, 2010.

F. Gruau, D. Whitley, and L. Pyeatt. A comparison between cellular encoding and direct encoding for genetic neural networks. In J. R. Koza, D. E. Goldberg, D. B. Fogel, and R. L. Riolo, editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*, pages 81–89, Cambridge, MA, 1996. MIT Press.

D. Hart. Toward greater artistic control for interactive evolution of images and animation. In M. Giacobini, editor, *Applications of Evolutionary Computing*, volume 4448 of *Lecture Notes in Computer Science*, pages 527–536. Springer Berlin / Heidelberg, 2007.

J. 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.

A. K. Hoover, P. A. Szerlip, and K. O. Stanley. Interactively evolving harmonies through functional scaffolding. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '11, New York, NY, USA, 2011. ACM.

J. Horn, N. Nafpliotis, and D. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pages 82 –87 vol.1, June 1994.

G. S. Hornby and J. B. Pollack. Creating high-level components with a generative representation for body-brain evolution. *Artificial Life*, 8(3), 2002.

D. James and P. Tucker. Anji homepage. `http://anji.sourceforge.net/`, 2005–2010.

B. Johanson and R. Poli. GP-music: An interactive genetic programming system for music generation with automated fitness raters. In *Proceedings of the Third Annual Conference on Genetic Programming*, volume 3, pages 181–186. IEEE Press, 1998.

L. P. Kaelbling, M. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence*, 4:237–285, May 1996.

W. M. Kier and K. K. Smith. Tongues, tentacles and trunks: the biomechanics of movement in muscular-hydrostats. *Zoological Journal of the Linnean Society*, 83:307–324, 1985.

S. Kistemaker and S. Whiteson. Critical factors in the performance of novelty search. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, GECCO '11, pages 965–972, New York, NY, USA, 2011. ACM.

J. Knowles, R. Watson, and D. Corne. Reducing local optima in single-objective problems by multi-objectivization. In *Evolutionary Multi-Criterion Optimization*, pages 269–283. Springer, 2001.

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

J. R. Koza. *Genetic Programming II: Automatic Discovery of Reusable Programs.* MIT Press, Cambridge, MA, 1998.

W. B. Langdon. Pfeiffer – A distributed open-ended evolutionary system. In B. Edmonds, N. Gilbert, S. Gustafson, D. Hales, and N. Krasnogor, editors, *Proceedings of the Joint Symposium on Socially Inspired Computing (METAS 2005)*, AISB'05, pages 7–13, University of Hertfordshire, Hatfield, UK, April 2005.

J. Lehman and K. O. Stanley. Exploiting open-endedness to solve problems through the search for novelty. In S. Bullock, J. Noble, R. Watson, and M. Bedau, editors, *Proceedings of the Eleventh International Conference on Artificial Life (Alife XI)*, Cambridge, MA, 2008. MIT Press.

J. Lehman and K. O. Stanley. Efficiently evolving programs through the search for novelty. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, GECCO '10, pages 837–844, New York, NY, USA, 2010a. ACM.

J. Lehman and K. O. Stanley. Revising the evolutionary computation abstraction: minimal criteria novelty search. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, GECCO '10, pages 103–110, New York, NY, USA, 2010b. ACM.

J. Lehman and K. O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223, 2011.

G. E. Liepins and M. D. Vose. Deceptiveness and genetic algorithm dynamics. Technical Report CONF-9007175-1, Oak Ridge National Lab., TN (USA); Tennessee Univ., Knoxville, TN (USA), 1990.

L.-J. Lin. Hierarchical learning of robot skills by reinforcement. In *Proceedings of the International Joint Conference on Neural Networks*, Piscataway, NJ, 1993. IEEE.

P. Machado and A. Cardoso. All the truth about NEvAr. *Applied Intelligence*, 16(2):101–118, 2002.

S. W. Mahfoud. *Niching Methods for Genetic Algorithms.* PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, May 1995.

K. Mathias and D. Whitley. Remapping Hyperspace During Genetic Search: Canonical Delta Folding. *Foundations of genetic algorithms*, 2:167–186, 1993.

J. F. Miller. Evolving a self-repairing, self-regulating, French flag organism. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004)*, Berlin, 2004. Springer Verlag.

T. M. Mitchell. *Machine Learning.* McGraw-Hill, New York, NY, 1997.

J.-B. Mouret. Novelty-based multiobjectivization. In S. Doncieux, N. Bredèche, and J.-B. Mouret, editors, *New Horizons in Evolutionary Robotics*, volume 341 of *Studies in Computational Intelligence*, pages 139–154. Springer Berlin / Heidelberg, 2011.

H. Mühlenbein and T. Manning. Convergence theory and applications of the Factorized Distribution Algorithm. *Journal of Computing and Information Technology*, 7(1):19–32, 1999.

G. L. Nelson. Sonomorphs: An application of genetic algorithms to growth and development of musical organisms. In *Proceedings of the Fourth Biennial Art and Technology Symposium*, pages 155–169, 1993.

M. Pelikan and D. Goldberg. Escaping hierarchical traps with competent genetic algorithms. Technical Report 2001003, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, 2001.

S. Perkins and G. Hayes. Robot shaping–principles, methods, and architectures. Technical Report 795, Department of Artifical Intelligence, University of Edinburgh, 1996.

A. L. Ralescu. Generalization of the Hamming distance using fuzzy sets. Technical Report JSPS Senior Research Fellowship, Laboratory for Mathematical Neuroscience, The Brain Science Institute, RIKEN, Japan, 2003.

T. Reil and P. 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.

S. Risi, C. Hughes, and K. Stanley. Evolving plastic neural networks with novelty search. *Adaptive Behavior*, 2011. To appear.

S. Risi and K. O. Stanley. Enhancing es-hyperneat to evolve more complex regular neural networks. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, GECCO '11, pages 1539–1546, New York, NY, USA, 2011. ACM.

S. Risi, S. D. Vanderbleek, C. E. Hughes, and K. O. Stanley. How novelty search escapes the deceptive trap of learning to learn. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2009)*, New York, NY, USA, 2009. ACM Press.

J. Romero and P. Machado, editors. *The art of artificial evolution: A handbook on evolutionary art and music*. Springer-Verlag, New York, NY, 2008.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, pages 318–362. MIT Press, Cambridge, MA, 1986.

C. Ryan. Pygmies and civil servants. In K. E. Kinnear, Jr., editor, *Advances in Genetic Programming*, chapter 11, pages 243–263. MIT Press, 1994.

N. Saravanan and D. B. Fogel. Evolving neural control systems. *IEEE Expert*, pages 23–27, June 1995.

J. Secretan, N. Beato, D. B. D.Ambrosio, A. Rodriguez, A. Campbell, J. T. Folsom-Kovarik, and K. O. Stanley. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation*, 19(3):373–403, 2011.

J. Secretan, N. Beato, D. B. D'Ambrosio, A. Rodriguez, A. Campbell, and K. O. Stanley. Picbreeder: Evolving pictures collaboratively online. In *CHI '08: Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1759–1768, New York, NY, USA, 2008. ACM.

C. B. Sigrist and R. J. Sommer. Vulva formation in Pristionchus pacificusrelies relies on continuous gonadal induction. *Development Genes and Evolution*, 209:451–459, 1999.

K. Sims. Artificial evolution for computer graphics. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '91)*, pages 319–328, New York, NY, 1991. ACM Press.

K. Sims. Genetic images interactive exhibit. *Centre Georges Pompidou Revue Virtuelle Notebook*, 5, 1993.

K. Sims. Galapagos interactive exhibit, Decordova Museum, Lincoln, MA. `http://www.genarts.com/karl/geneticimages`, 1997.

S. P. Singh. Transfer of learning by composing solutions of elemental sequential tasks. *Machine Learning*, 8:323–339, 1992.

N. Sinha, R. Chakrabarti, and P. Chattopadhyay. Evolutionary programming techniques for economic load dispatch. *Evolutionary Computation, IEEE Transactions on*, 7(1):83 – 94, 2003.

J. R. Smith. Designing biomorphs with an interactive genetic algorithm. In R. K. Belew and L. B. Booker, editors, *Proceedings of the 4th International Conference on Genetic Algorithms (ICGA-91)*, pages 535–538, San Mateo, CA, July 1991. Morgan Kaufmann.

A. Soltoggio. *Evolutionary and Computational Advantages of Neuromodulated Plasticity.* PhD thesis, University of Birmingham, UK, October 2008.

A. Soltoggio, A. J. Bullinaria, C. Mattiussi, P. Drr, and D. Floreano. Evolutionary advantages of neuromodulated plasticity in dynamic, reward-based scenarios. In S. Bullock, J. Noble, R. Watson, and M. Bedau, editors, *Proceedings of the Eleventh International Conference on Artificial Life (Alife XI)*, Cambridge, MA, 2008. MIT Press.

K. O. Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines Special Issue on Developmental Systems*, 8(2):131–162, 2007.

K. O. Stanley. Searching without objectives. Speech presented at SPLASH 2010, Reno/Tahoe, Nevada, October 21, 2010. `http://www.infoq.com/presentations/Searching-Without-Objectives`

K. O. Stanley, D. B. D'Ambrosio, and J. Gauci. A hypercube-based indirect encoding for evolving large-scale neural networks. *Artificial Life*, 15(2):185–212, 2009.

K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127, 2002.

K. O. Stanley and R. Miikkulainen. A taxonomy for artificial embryogeny. *Artificial Life*, 9 (2):93–130, 2003.

K. O. Stanley and R. Miikkulainen. Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21:63–100, 2004.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

S. R. Szumlanski, A. S. Wu, and C. E. Hughes. Conflict resolution and a framework for collaborative interactive evolution. In *Proceedings of the 21st national conference on Artificial intelligence - Volume 1*, pages 512–517. AAAI Press, 2006.

H. Takagi. Interactive evolutionary computation: Fusion of the capacities of EC optimization and human evaluation. *Proceedings of the IEEE*, 89(9):1275–1296, 2001.

N. Tokui and H. Iba. Music composition with interactive evolutionary computation. In *Proceedings of the Third International Conference on Generative Art*, pages 215–226, 2000.

M. van de Panne and A. 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.

D. A. V. Veldhuizen and G. B. Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary Computation*, 8(2):125–147, 2000.

J. J. Ventrella. Disney meets Darwin: an evolution-based interface for exploration and design of expressive animated behavior. Master's thesis, Massachusetts Institute of Technology, June 1994a.

J. J. Ventrella. Explorations in the emergence of morphology and locomotion behavior in animated characters. In *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, pages 436–441, Cambridge, MA, July 1994b. MIT Press.

J. J. Ventrella. Disney meets Darwin: The evolution of funny animated figures. In *Proceedings of the Computer Animation Conference*, CA '95, pages 35–43, Washington, DC, USA, April 1995. IEEE Computer Society.

P. Verbancsics and K. O. Stanley. Constraining connectivity to encourage modularity in hyperneat. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, GECCO '11, pages 1483–1490, New York, NY, USA, 2011. ACM.

S. Whiteson and P. Stone. Evolutionary function approximation for reinforcement learning. *Journal of Machine Learning Research*, 7:877–917, 2006.

D. Whitley, K. Mathias, and P. Fitzhorn. Delta-Coding: An iterative search strategy for genetic algorithms. In R. K. Belew and L. B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 77–84. Kaufmann, 1991.

L. D. Whitley. Fundamental principles of deception in genetic search. *Foundations of genetic algorithms*, 1(3):221–241, 1991.

D. H. Wolpert and W. Macready. No free lunch theorems for search. Technical Report SFI-TR-95-01-010, The Santa Fe Institute, Santa Fe, NM, 1995.

B. G. Woolley and K. O. Stanley. Evolving a single scalable controller for an octopus arm with a variable number of segments. In R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, editors, *Parallel Problem Solving from Nature – PPSN XI*, volume 6239 of *Lecture Notes in Computer Science*, pages 270–279. Springer, 2010.

B. G. Woolley and K. O. Stanley. On the deleterious effects of a priori objectives on evolution and representation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '11, New York, NY, USA, 2011. ACM.

L. World. Aesthetic Selection: The Evolutionary Art of Steven Rooke. *IEEE Computer Graphics and Applications*, 16(1):4–5, 1996.

X. Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447, 1999.

X. Yao, Y. Liu, and G. Lin. Evolutionary programming made faster. *Evolutionary Computation, IEEE Transactions on*, 3(2):82 –102, July 1999.

Y. Yekutieli, R. Sagiv-Zohar, R. Aharonov, Y. Engel, B. Hochner, and T. Flash. Dynamic model of the octopus arm. I. Biomechanics of the octopus reaching movement. *Journal of neurophysiology*, 94(2):1443–1506, 2005.

N. Zaera, D. Cliff, and J. Bruten. (Not) evolving collective behaviours in synthetic fish. In P. Maes, M. J. Mataric, J.-A. Meyer, J. Pollack, and S. W. Wilson, editors, *From animals to animats 4: proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB'96)*, pages 635–642, Cambridge, Mass., 1996. MIT Press.