Minimal Criterion Coevolution: A New Approach to Open-Ended Search

To appear in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2017). New York, NY: ACM

Jonathan C. Brant
Dept. of Computer Science, University of Central Florida
Orlando, Florida 32816-2362
jbrant@cs.ucf.edu

Kenneth O. Stanley
Dept. of Computer Science, University of Central Florida
Orlando, Florida 32816-2362
kstanley@cs.ucf.edu

ABSTRACT

Recent studies have emphasized the merits of search processes that lack overarching objectives, instead promoting divergence by rewarding behavioral novelty. While this less objective search paradigm is more open-ended and divergent, it still differs significantly from nature's mechanism of divergence. Rather than measuring novelty explicitly, nature is guided by a single, fundamental constraint: survive long enough to reproduce. Surprisingly, this simple constraint produces both complexity and diversity in a continual process unparalleled by any algorithm to date. Inspired by the relative simplicity of open-endedness in nature in comparison to recent non-objective algorithms, this paper investigates the extent to which interactions between two coevolving populations, both subject to their own constraint, or *minimal criterion*, can produce results that are both functional and diverse even without any behavior characterization or novelty archive. To test this new approach, a novel maze navigation domain is introduced wherein evolved agents must learn to navigate mazes whose structures are simultaneously coevolving and increasing in complexity. The result is a broad range of maze topologies and successful agent trajectories in a single run, thereby suggesting the viability of *minimal criterion coevolution* as a new approach to non-objective search and a step towards genuinely open-ended algorithms.

CCS CONCEPTS

•Theory of computation → Evolutionary algorithms; **•Computing methodologies → Artificial life; Evolutionary robotics;** *Neural networks;*

KEYWORDS

novelty search; non-objective search; open-ended evolution; artificial life; NEAT; coevolution;

ACM Reference format:
Jonathan C. Brant and Kenneth O. Stanley. 2017. Minimal Criterion Coevolution: A New Approach to Open-Ended Search. In *Proceedings of GECCO '17, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3071178.3071186

GECCO '17, Berlin, Germany
© 2017 ACM. 978-1-4503-4920-8/17/07...$15.00
DOI: http://dx.doi.org/10.1145/3071178.3071186

1 INTRODUCTION

A delicate tension has persisted historically between the practically-oriented pursuits of evolutionary computation (EC) and the more abstract questions of artificial life (alife) that in recent years may be starting to fade. The focus in EC on solving problems has traditionally excluded more philosophical issues in alife like the pursuit of open-endedness [27] (i.e. trying to understand processes that continue to generate interesting artifacts indefinitely). Historically, open-ended evolution like that seen in nature has been studied as a side-effect of Earth-like domains with organism-like occupants and ecosystems [19, 23], a distant cousin to evolving targeted solutions to problems like walking or maze-navigation. However, this dichotomy has begun to close with the recent rise of non-objective algorithms that began with the introduction of novelty search [8, 10].

This new class of algorithms focuses on divergence, i.e. on moving the search away from previously-visited locations, which aligns more closely with the idea of open-ended discovery. And yet, interestingly, such algorithms can be applied to domains outside the traditional confines of alife. In recent years these algorithms have increased in sophistication as researchers have learned to balance divergent pressure with a more objective fitness-like counterpart to produce a process called *quality diversity* (QD) that yields both high quality and diversity simultaneously in one run [22]. QD algorithms such as novelty search with local competition (NSLC) [11] and the Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) [17] have produced arguably practical results, such as a repertoire of robot walking strategies [29], even as they incorporate some of the flavor of alife research into open-endedness.

However, despite this progress, there remain limits to any analogy between QD and open-endedness in the spirit of nature. First, to date QD algorithms generally require defining a *behavior characterization* (BC), which is a descriptor of the phenotype behavior (and sometimes physical properties) of individuals in the search space. This requirement diverges from nature, where there is no need for anyone to develop a formalism for describing organism behavior simply to allow evolution to proceed. Second, though it may be vast, the descriptor space of BCs is intrinsically finite. For example, if the behavior of an individual is its average velocity and size, once the full spectrum of velocities and sizes has been sampled, the pressure toward novelty by necessity diminishes significantly. While there are interesting hints that BCs may be possible to expand over time [13], allowing the behaviors to expand *unboundedly* remains a significant challenge. Finally, QD algorithms generally require an archive of some sort – in NSLC it is literally a record of past points visited, while in MAP-Elites it is points already visited in the existing map. In nature, no such archiving strategy is needed.

Nature thereby becomes inspiration for seeking approaches to open-ended discovery without the need for such mechanisms. To explore alternative paradigms for open-endedness outside typical alife worlds, this paper pursues a strategy relying heavily on *drift*. QD algorithms to date rely on identifying and hence rewarding novelty to facilitate divergence, which leads to the need for BCs (descriptors), but divergence is also in principle a byproduct of simple drift [12] subject to a minimal criterion (MC) constraint for reproduction. However, while researchers have previously explored MCs [9, 14], it is difficult to know a priori what the MC should be (or how it should change over time) to facilitate finding interesting new behaviors over a long period of time. The main contribution of this paper is to show that *coevolution* may help to address this problem by forcing the MCs of two interlocking populations to satisfy minimal criteria with respect to each other, even as both populations are gradually shifting simultaneously.

This idea, called *minimal criterion coevolution* (MCC), is tested by coevolving a population of mazes with a population of maze navigating neural networks, where the MC is that each maze must be solved by at least one neural network, and each neural network must solve at least one maze. Anyone who satisfies that constraint can reproduce. For the first time (to our knowledge), a proliferation of diverse solutions (and diverse mazes) of varying complexity is shown to emerge in a single run from an algorithm without any behavioral description or archive. To provide guidance for future applications, two variants of MCC are tested to expose the conditions that yield best performance.

2 BACKGROUND

This section reviews the history behind the MC in the fields of EC and alife. Coevolution in non-objective search is also discussed.

2.1 Minimal Criteria in EC

The field of EC has historically focused on the idea of selecting individuals for reproduction based on an objective performance scale and iteratively converging on a solution [2]. The less common notion of imposing a lower bound on individual performance was introduced early on by Mattiussi and Floreano [15] and later elaborated as the sole driver of evolution by Maesani et al. [14]. Their proposed algorithm, called *viability evolution* (ViE), introduces a threshold on one or more constraints that individuals must meet to survive and reproduce. At the beginning of evolution, these thresholds, or *viability boundaries*, encompass the entire population but are then incrementally tightened until the algorithm converges on a set of individuals who satisfy all of the constraints.

One benefit of this approach is that it avoids the often tricky art of crafting an objective function, particularly when that function considers multiple objectives that must be appropriately weighted. Another benefit is that the loose criterion of viability means that the reproductive pool is generally larger and less biased, allowing more stepping stones to be sampled. MCC draws original inspiration from the generally fitness-agnostic approach of ViE, but fundamentally differs in that it has no drive toward convergence.

Like ViE, novelty search [8, 10] and QD algorithms [11, 17, 22] have challenged conventional views of objective-driven evolutionary search. By design, novelty search and QD tend to spread the search to anywhere novel. A side effect and common criticism of this dynamic is the algorithm's consequent susceptibility to expending resources exploring unimportant areas of vast behavior spaces. To address this inefficiency and to better align novelty search with selection processes in natural evolution, minimal criteria novelty search (MCNS) was introduced [9]. MCNS bounds the behavior space by tagging individuals who traverse beyond the boundary (i.e. the MC) as non-viable, meaning they are not eligible to reproduce. This approach avoids establishing new lineages of individuals that would primarily explore areas of the behavior space that are orthogonal to the desired search results. MCC in the present paper preserves the concept of the MC, but demonstrates how the MC *alone* (i.e. without a BC, without narrowing viability boundaries, *and* without the need for an archive) can efficiently produce open-ended evolutionary artifacts, even in non-alife domains.

2.2 The MC and Open-endedness in Alife

Soros and Stanley [26] recently suggested that the MC is a reasonable interpretation of natural evolution in the sense that *making a copy of oneself* is the MC for continuing one's lineage. They introduced an alife world called Chromaria to highlight this point – in Chromaria there is no explicit fitness function and *everyone* who satisfies the MC reproduces, yet evolution still progresses. In fact, constraining evolution through the MC rather than explicit fitness in effect helps to avoid convergence and promote open-endedness. However, a key factor in Chromaria's success is that organisms *interact* with each other to satisfy the MC, which allows its difficulty to vary organically over time as the population changes. This idea raises the interesting question of whether a similarly open-ended dynamic can be created outside of a conventional alife world.

2.3 Coevolution

Coevolution in the field of EC has a rich history [20] focused on attributing fitness when it is prohibitive to evaluate individuals on their own through an extrinsic absolute measure of performance. These types of scenarios are traditionally divided between *competitive coevolution*, where individuals are rewarded according to their performance against other individuals in a competition [3], and *cooperative coevolution*, where instead fitness is a measure of how well an individual performs in a cooperative team with other evolving individuals [30]. Coevolution itself has inspired interest in open-endedness because of the potential for a never-ending arms race, but in practice has proven difficult to sustain perpetually [3]. An important contribution of the present work is to introduce a new kind of coevolution (*minimal criterion coevolution*) that does not fit neatly into the traditional competitive versus cooperative dichotomy. Instead, the idea is to provoke a Chromaria-like divergent search dynamic through the interaction of individuals under a MC, but in a more general context that is independent from any overarching alife world. While this idea bears some resemblance to previous work in co-evolving learners and tests [1], a key distinction is that it does not require an archive for non-dominated learners, nor does it impose a ranking among solutions. Moreover, rather than placing an emphasis on evolving the context (i.e. the test) for a single population, the focus of this work is in producing a broad array of functional diversity in both populations.

3 APPROACH: MINIMAL CRITERION COEVOLUTION

In nature, each organism or species has the potential to traverse its own unique path to satisfying the MC. For example, the developmental path toward reproductive viability for bacteria is far shorter and less complex than for mammals. Furthermore, the interactions between individuals and other coevolving populations results in an on-going flux of each species' path towards the MC.

Inspired by this perspective, minimal criterion coevolution (MCC) is proposed as a dual-population coevolutionary algorithm; however, given its unorthodox evaluation process and reward mechanism, it does not fall into the traditional categories of cooperative or competitive coevolution. That is, individuals from one population are not rewarded based on the success or failure of another; rather, they are selected as parents based solely on their satisfaction of the MC. A notable characteristic of this process is that it imposes no ranking among individuals (that is, they either satisfy the MC or not), necessitating a selection method that is free of bias. One such method that has been demonstrated to be effective in alife is storing the population in a fixed-size queue, called the "parent queue" in the alife world of Chromaria [26]. The queue retains individuals who satisfy the MC in the order of insertion, and a queue pointer points to the location of the individual next in line for reproduction. If the pointer reaches the end of the queue, it simply wraps back around to the beginning. Additionally, if the insertion of a new individual exceeds the queue's capacity, the oldest individual in the queue is removed to make room. This setup overall ensures that every individual added to the queue gets *at least one* chance to reproduce. Other than that, there is no attempt to say any candidate is better than any other. This agnosticism means that the algorithm keeps many divergent paths open simultaneously without convergence, as long as they satisfy the MC.

Most evolutionary algorithms begin evolution with a randomly generated population; however, the requirement that queue membership be predicated on satisfying the MC necessitates special consideration for MCC initialization. In particular, it is unlikely that a randomly generated individual will be capable of meeting a non-trivial MC, thereby denying admittance to any of the population queues. Therefore, each population queue must undergo a *bootstrap* process wherein the requisite number of seed genomes required by the applicable population queue are pre-evolved (novelty search serves as the bootstrap in this paper).

MCC's selection process closely resembles that of a steady-state evolutionary algorithm; however, its absolute measure of performance (i.e. the MC) allows large chunks of a population to be evaluated in parallel (their fitness never needs to be computed or compared). This freedom facilitates a distributed execution paradigm wherein evaluations are spread across multiple nodes, allowing MCC to scale when simulating evolutionary processes. Algorithm 1 formalizes the MCC selection, evaluation, and removal process, which is alternately applied to both of the coevolving populations.

Algorithm 1 MCC Evaluation Process

```
Require:
  batchSize - # of individuals to evaluate simultaneously
  numSeeds - # of seed genomes to evolve that satisfy the MC

  ▷ Evolve seed genomes that satisfy MC
  randPop ← GenerateRandomPopulation()
  viablePop ← EvolveSeedGenomes(randPop, numSeeds)

  loop
    ▷ Reproduce children and add parents back into queue
    parents ← viablePop.Dequeue(batchSize)
    children ← Reproduce(parents)
    viablePop.Enqueue(parents)

    for all child ∈ children do
      ▷ MC involves interaction with the other coevolving pop.
      mcSatisfied ← EvaluateMC(child)
      if mcSatisfied then
        viablePop.Enqueue(child)
      end if
    end for

    ▷ Removed oldest if queue capacity exceeded
    if viablePop.Size > viablePop.Capacity then
      numRemovals ← viablePop.Size − viablePop.Capacity
      RemoveOldest(viablePop, numRemovals)
    end if
  end loop
```

In addition to this base MCC algorithm, speciation can be introduced as a lightweight method to sustain diversity. It is lightweight because, unlike in other QD algorithms, it can be based entirely on the genome with no need to characterize or compare actual phenotypes or behaviors. While individuals from a given population remain physically stored in a queue structure, they are clustered into separate logical groups, or *species*, based on their genetic similarity and tagged with a species identifier. The seed genomes for a given population constitute the centroids of these species clusters. Additionally, the queue capacity is evenly distributed among species such that each species i has a maximum size equivalent to $\text{capacity}(i) = \frac{n}{s}$, where n is the number of individuals in the population, and s is the number of species.

In the speciated version, instead of simply including the next batch of individuals in the queue order, a proportionate number of individuals are aggregated from each species for reproduction (though queue insertion order still dictates the order of selection). Recall that only individuals who satisfy the MC are in the queue. The offspring of the batch who satisfy the MC are then added to the queue as normal but also assigned to the species that shares the greatest genetic similarity. If any species has exceeded its capacity as a result of speciating the offspring, the oldest individuals *assigned to those species* are removed from the queue. This process also ensures that the queue remains at or below capacity. Overall, the hope is that both populations will in effect diverge through drift, but the MC will force them only to drift toward further desirable interactions with each other, yielding a continual, virtually open-ended process. The importance of speciation to this process will be addressed in the experiment.

4 EXPERIMENT

Thanks to their relatively light computational load and easily visualized and adjustable level of deception, maze domains have proven instrumental in the initial investigations into various non-objective search algorithms [9, 10, 16, 18, 22]. In particular, mazes make deceptive and non-optimal trajectories (usually obtained by evolved neurocontrollers) through the solution space visually explicit in the form of dead ends and wandering paths, thereby facilitating the identification of principled strategies.

Mazes are also a good choice for investigating MCC, though their implications in its context are different. Rather than revealing an ability to explore diverse paths through a single maze, with MCC mazes demonstrate something new: *solutions to numerous mazes of varying complexity* in a single run, all without the need for any kind of BC or description. This novel capability portends a new, open-ended approach to QD with broad potential applications. For example, robot morphologies could be coevolved with controllers in an analogous realization of MCC in the future.

Maze-navigating agents use NEAT-evolved artificial neural networks (ANNs) to navigate from their starting location to the target location. The NeuroEvolution of Augmenting Topologies (NEAT) [28] algorithm is a common choice for evolving the agent's ANNs based on its long track record in control tasks, particularly in navigation domains [8, 10]. NEAT begins evolution with a population of networks that have minimal structure (i.e no hidden layer) and incrementally complexifies their topology by adding connections and nodes. Each structural addition is encoded in the NEAT genome and assigned a historical marking, allowing genomes to be compared based on the genes that they have in common. In conventional NEAT, genomes that are topologically similar are grouped together, or *speciated*, thereby protecting new innovations by reducing competition with more established genomes [28]. Importantly, ANN complexification increases the number of free parameters, which will often lead to increasingly complex behaviors.

The maze domain used for the experiments in this paper is a square grid with the start location in the upper-left corner and the target location in the lower right; both points remain fixed throughout evolution. An agent, controlled by a NEAT-evolved ANN, begins a trial at the start location and is evaluated on its ability to reach the target location within the time allotted for a single trial. Multiple obstructions and cul-de-sacs impede a direct path to the target, necessitating the evolution of nontrivial navigation strategies. The agent ANN architecture is identical to that used in the original novelty search studies [8]: six rangefinder sensors measure distance to line-of-sight obstructions while four pie slice radar sensors activate when the target location is within the sensor's arc. Motion is controlled by two actuators that can rotate and propel the agent. The sections that follow describe key elements of the experimental setup in detail.

4.1 Maze Evolution Methodology

While one population is the maze-navigating agents, the other is the mazes. Evolving mazes requires that the maze structure be encoded in a representation on which evolutionary operators can be applied. For this purpose, a maze in this work is a collection of component walls. Fixed exterior walls bound the maze space while interior walls introduce obstructions and generally increase the difficulty of navigating from one location to another. For the purposes of this paper, the complexity of a maze is estimated as the number of interior walls it contains. The fixed canvas size bounds the permissible number of interior walls, thereby imposing an upper limit on maze complexity that keeps this initial investigation tractable. (Future work will focus on removing this constraint.)

For each wall, the genome encodes the relative position of the wall within the maze and the relative position of an opening, or "passage." Like the NEAT connection gene weights, maze genomes adopt a real-valued encoding in the interval $[0, 1]$. The combination of *wall position* and *passage position* define a single maze gene. All maze genome reproduction is asexual, with three new parameters controlling mutations: The *wall mutation* probability impacts the relative position of the wall within the maze sub-space while the *passage mutation* probability affects the relative position of the passage within a wall. The *add wall probability* is a complexification operation, controlling the addition of wall genes. When the genome reaches maximum complexity, the add wall operation is disabled.

Mazes are decoded into their phenotype by iteratively bisecting the maze space and scaling the relative wall and passage position to the dimensions of the affected sub-space. For example, the first gene in the genome bisects the entire maze space, creating two sub-spaces with a passage through which the agent can traverse between the two. If the bisection is vertical, the second gene bisects the left-most sub-space, creating two additional sub-spaces. Similarly, if the original bisection is horizontal, the second gene bisects the uppermost sub-space. The orientation of each wall is dictated by the dimensions of its assigned sub-space such that the wall is placed perpendicular to the dimension of greatest magnitude. This process continues until all genes are decoded. The maze generation approach is inspired by the recursive division algorithm [4], but uses a breadth-first implementation to avoid unbalanced wall placement for mazes that have not yet reached maximum complexity. Figure 1 depicts the maze complexification process. As with the evolution of ANNs in NEAT, mazes begin evolution in this experiment with few walls and increase in complexity as evolution progresses.

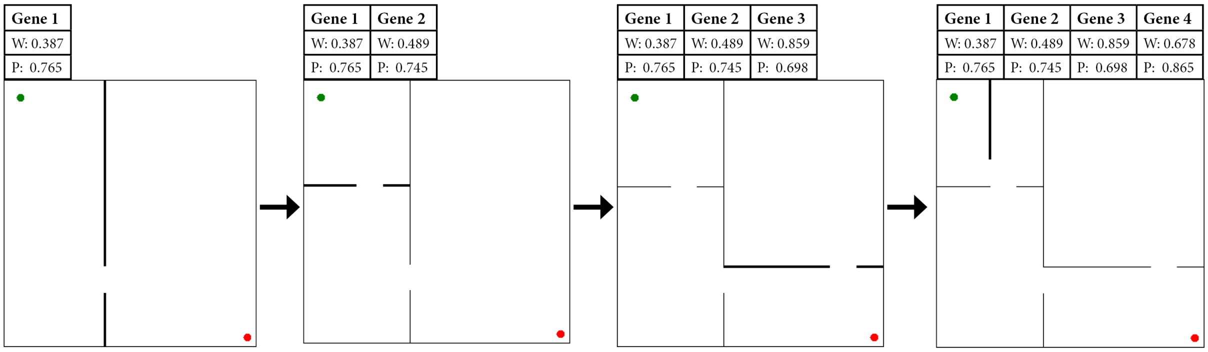
4.2 Minimal Criterion Definition

Soros and Stanley [26] have suggested that a nontrivial MC is a necessary condition for open-ended evolution. The motivation behind the MCs in this work is that the ability to navigate mazes is nontrivial. In these experiments, each agent is evaluated on potentially many different mazes from the maze population queue, and the MC is that the agent must solve *at least one of those mazes*. Conversely, the maze population MC is a successful navigation by at least one agent from the current agent queue.

Recall that for evolution to begin, the individuals in the initial population must themselves meet the MC. Given its ability to efficiently discover diverse solutions, the novelty search algorithm serves as a "bootstrap" method to evolve a requisite number of seed genomes that meet the MC for a set of simple mazes (parameters are given in section 4.4).

4.3 Experimental Methods

Given that MCC represents a significant departure from both objective and, to a lesser extent, non-objective search algorithms published to date, a direct, quantitative comparison to those algorithms in their current form is not feasible. For example, typical



Gene 1
W: 0.387
P: 0.765

Gene 1	Gene 2
W: 0.387	W: 0.489
P: 0.765	P: 0.745

Gene 1	Gene 2	Gene 3
W: 0.387	W: 0.489	W: 0.859
P: 0.765	P: 0.745	P: 0.698

Gene 1	Gene 2	Gene 3	Gene 4
W: 0.387	W: 0.489	W: 0.859	W: 0.678
P: 0.765	P: 0.745	P: 0.698	P: 0.865

Figure 1: Maze genotype (top) and phenotype (bottom). An example of the maze complexification process is shown. The table above each maze enumerates the genes within the genome that decode to the phenotype immediately below them. Each gene contains a wall position and passage position, both real numbers in the interval $[0, 1]$. The wall position describes the relative position of a wall within its subspace while the passage position describes the relative position of a passage within that wall. Each gene maps to one wall in the phenotype. The wall that corresponds to the rightmost gene in its genotype is emboldened in the phenotype depiction. The starting position and target position are fixed and are shown in the phenotype at the upper left and lower right respectively.

QD algorithms invent many new behaviors for a given maze, but they do not invent new mazes and new behaviors at the same time. Instead, the experiments that follow survey for the first time the artifacts produced by an almost entirely undirected and potentially open-ended search process, and elucidate some of the factors that can enhance such a process.

The first experiment admits agents to a global agent queue and mazes to a global maze queue based on satisfaction of their respective MC (detailed in section 4.2). As MCC is a parallel algorithm, a "batch" of individuals is selected from both the maze and agent parent queues on each iteration and evaluated asynchronously. When either queue reaches its capacity, excess individuals are removed based on their comparative age (i.e. oldest first) to make room for new, functional offspring. This configuration represents a simple baseline operation of an MCC algorithm, and acts as a control against which the second experiment is compared.

While MCC has no explicit objective or drive toward novel behaviors, some individuals will be better at producing competent offspring than others. Over time, this disparity could result in a coevolutionary convergence. In nature, individuals are segregated into species, each of which fills a particular ecological niche [6]. Every niche has a finite capacity, thereby allowing for a more localized regulation of population growth, which in turn leaves room for founding a wide array of diverse niches. Accordingly, genetics-based speciation is a long-standing popular mechanism in evolutionary computation for preserving diversity [10, 21, 28].

The second experiment models this notion of local regulation by speciating both the agent queue and the maze queue. Given that agents are evolved using the NEAT algorithm, they adopt NEAT's built in speciation mechanism [28]. Mazes are speciated based on a cantor pairing of each gene's relative wall position and relative passage position. More concretely, both gene components (wall position and passage position) are combined to give each gene m_i in maze genome m a scalar value:

$$m_i = \frac{1}{2}(w_i + p_i)(w_i + p_i + 1) + p_i, \tag{1}$$

where w_i and p_i are the relative wall and passage positions specified by gene i, respectively. Genetic similarity is then based on computing the Euclidean distance between two full maze position vectors. If two vectors have different sizes, the missing genes in the shorter vector are assigned position zero. Individuals in both the agent queue and the maze queue are grouped with genetically similar agents or mazes respectively. Selection and removal processes are then carried out on a per-species basis (as explained in section 3). It is important to note that this kind of genetics-based speciation *does not* require characterizing behavior or maintaining an archive [8, 10, 24] or map [17], and therefore sustains the unique nature of MCC as a QD algorithm without a behavior characterization.

4.4 Experimental Parameters

The novelty search bootstrap is executed with 250 agents and NEAT parameters identical to those in Lehman and Stanley [10]. Agents are evaluated on ten randomly-generated mazes, each with 8 walls of random composition. Execution halts when 20 distinct agents are evolved that can solve one or more mazes, and each of the 10 mazes is solved by at least one agent. The 20 agents and 10 mazes then seed their respective MCC queues. In the speciated variant, the seed genomes are the initial centroids of each species cluster.

Both configurations are executed for 20 runs of 2,000 batches each. The agent queue is seeded with 20 genomes and has a maximum capacity of 250 with a 0.7 probability of mutating connection weights, 0.1 of adding a connection, 0.01 of adding a neuron, and 0.0001 of deleting a connection. The maze queue is seeded with 10 genomes and has a maximum capacity of 50 with a 0.05 probability of mutating a wall location, 0.05 of mutating a passage location, and 0.7 of adding a wall. These values produced reasonably diverse and complex results in an initial parameter sweep.

Agents have a maximum velocity of 3 units per second and are allotted up to 600 time steps to complete a 320 × 320 unit maze. SharpNEAT version 3.0 [5] is the neuroevolution platform, and was extended to implement the MCC and novelty search algorithms as well as to support encoding and evolving maze genomes. Source code for the experiments is available at http://bit.ly/2oM22YK.

5 RESULTS

One of the primary hypothesized benefits of MCC is its ability to produce a broad diversity of functional solutions to a range of different problems. To that end, this section showcases the range and complexity of navigable mazes and the learned agent trajectories through those mazes. The quantitative portion of the analysis focuses on maze complexity trends over evolution and the comparative diversity of trajectories between the control and speciated variant.

5.1 Qualitative Results

Figure 2 depicts a sample of the mazes evolved by a typical *single* run of the MCC control. While the mazes often reach maximum complexity with effective agent trajectories, the trajectories that the maze structures permit tend to be similar. Interestingly, different runs of the control converge to different such consistent structures, suggesting that the agent and maze populations are converging in control runs to trajectories and maze structures that become standards across the run.

Figure 3 depicts a sample of the mazes evolved by a single run of the MCC speciated variant. Maze structures and agent trajectories are significantly more diverse in these runs, suggesting that enforcing genetic diversity through maze and agent queue speciation is a viable approach to maintaining divergent and functional phenotypic diversity in a minimal criterion coevolutionary system.

5.2 Quantitative Results

To give a more holistic and comprehensive view of system dynamics, a quantitative analysis is performed to capture global diversity trends and increases in system complexity over evolution.

Population diversity is computed by measuring the Euclidean distance between each point on a given agent's trajectory and the corresponding point *at the same time step* on every other successful agent's trajectory. Formally, the diversity score of an agent trajectory is given by

$$\text{div}(a) = \frac{1}{n}\sum_{i=1}^{n}(\frac{1}{m}\sum_{t=1}^{m}\text{dist}(a_t, \mu_{i,t})), \qquad (2)$$

where a represents the agent trajectory whose diversity is being measured, a_t is the position of the agent at time step t, $\mu_{i,t}$ corresponds to the position of the agent against which a is compared at time step t, m is the number of time steps in the simulation, n corresponds to the number of agents who also successfully navigated a maze in the current maze population, and *dist* represents the function that computes the distance between any two agent's trajectories (in this case, the Euclidean distance). The overall diversity is simply the average of all agent trajectory diversity measurements. This approach has been shown (and taken before) to be effective for measuring phenotypic and behavioral diversity [7, 18, 25].

Ultimately the primary concern of open-endedness or QD is to accumulate diverse, functional artifacts throughout the entirety of the run. Figure 4 (which encompasses maze-solving agents discovered over *entire runs*) demonstrates that speciating the maze and agent queues results in a significantly larger diversity of maze solutions ($p < 0.001$; Welch's t-test).

An interesting side effect of a lack of diversity in the control is a comparatively high proportion of mazes that are navigable by the existing agents at any point in time (figure 5). The speciated variant indicates a significantly lower proportion of mazes that are successfully navigated by each agent at every point in evolution ($p < 0.001$; Welch's t-test), a side effect of more diverse maze structures and navigation strategies. An important implication is that the mazes being evolved are nontrivial to solve because there is not a single navigation strategy that can solve more than a small fraction of them.

In addition to producing functional diversity, MCC was able to evolve increasingly complex behavioral artifacts, quickly reaching maximum maze complexity. Figure 6 depicts a population-wide drift toward higher maze complexity, which implies the development of increasingly advanced control policies to solve those mazes. This result raises the interesting question of whether complexity would continue to increase unboundedly if mazes did not have a size cap.

6 DISCUSSION AND CONCLUSIONS

Of fundamental importance is the extent to which MCC using *either* experimental method was able to induce a reciprocating increase in complexity between both populations, ultimately reaching the maximum complexity supported by the domain constraints. This phenomenon persisted consistently in all runs even without any BCs or behavior archives. Speciation effectively adds the ability also to explore divergent lines simultaneously, validating the existence of an entirely new kind of divergent search algorithm.

In all runs, MCC evolved agents able to solve mazes at their maximum level of complexity (i.e. with the maximum number of walls supported by mazes of the given dimensions) well before the allotted time. This complexity ceiling was imposed to ensure the computational tractability of this intial investigation, wherein the dynamics of MCC were previously unknown. However, as an algorithm designed to facilitate open-ended evolutionary dynamics even beyond conventional QD approaches, MCC should in the future be tested for its capacity to support the continual, unbounded complexification of all coevolving populations. The hope is that MCC will serve as a scalable and flexible platform for the production and investigation of both QD and open-ended evolution.

REFERENCES

[1] Edwin D De Jong. 2004. The Incremental Pareto-Coevolution Archive. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004)*. Springer Verlag, Berlin.

[2] Kenneth A De Jong. 2006. *Evolutionary computation: a unified approach.* MIT press.

[3] Sevan G Ficici and Jordan B Pollack. 1998. Challenges in Coevolutionary Learning: Arms–Race Dynamics, Open–Endedness, and Mediocre Stable States. In *Proceedings of the Sixth International Conference on Artificial Life*, Adami et al (Ed.). MIT Press, Cambridge, MA, 238–247.

[4] Martin Foltin. 2011. Automated Maze Generation and Human Interaction. *Brno: Masaryk University Faculty Of Informatics* (2011).

[5] Colin Green. 2016. SharpNEAT v3. *URL: http://sharpneat.sourceforge.net* (2016).

[6] Marc Kirschner and John Gerhart. 1998. Evolvability. *Proceedings of the National Academy of Sciences* 95, 15 (1998), 8420–8427.

[7] Bakir Lacevic and Edoardo Amaldi. 2010. On population diversity measures in Euclidean space. In *IEEE Congress on Evolutionary Computation.* IEEE, 1–8.

[8] Joel Lehman and Kenneth O Stanley. 2008. Exploiting Open-Endedness to Solve Problems Through the Search for Novelty.. In *Proceedings of the Eleventh International Conference on Artificial Life (Alife XI).* MIT Press, 329–336.

[9] Joel Lehman and Kenneth O Stanley. 2010. Revising the evolutionary computation abstraction: minimal criteria novelty search. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2010).* ACM, 103–110.

[10] Joel Lehman and Kenneth O Stanley. 2011. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation* 19, 2 (2011),

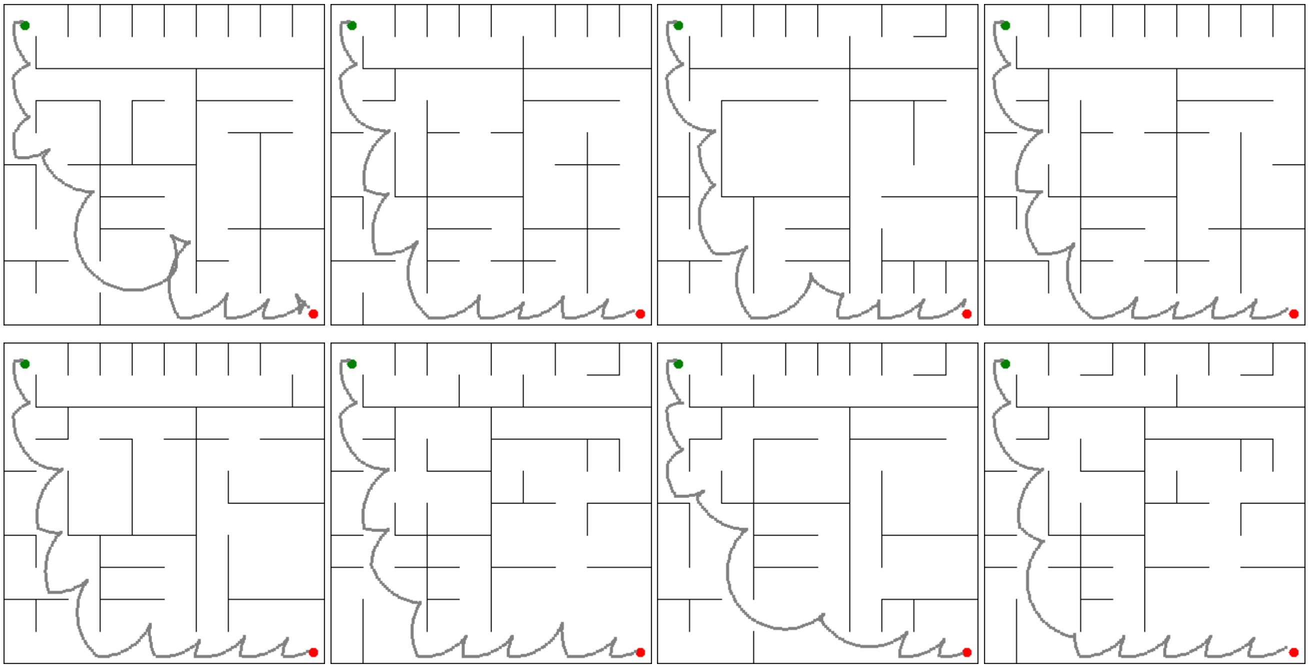


Figure 2: Agent Trajectories from the MCC Control Experiment. A sample of eight agent trajectories through different mazes evolved by the MCC control experiments during the same run are shown. These exemplify how solved mazes from a run of the control tend to look similar.

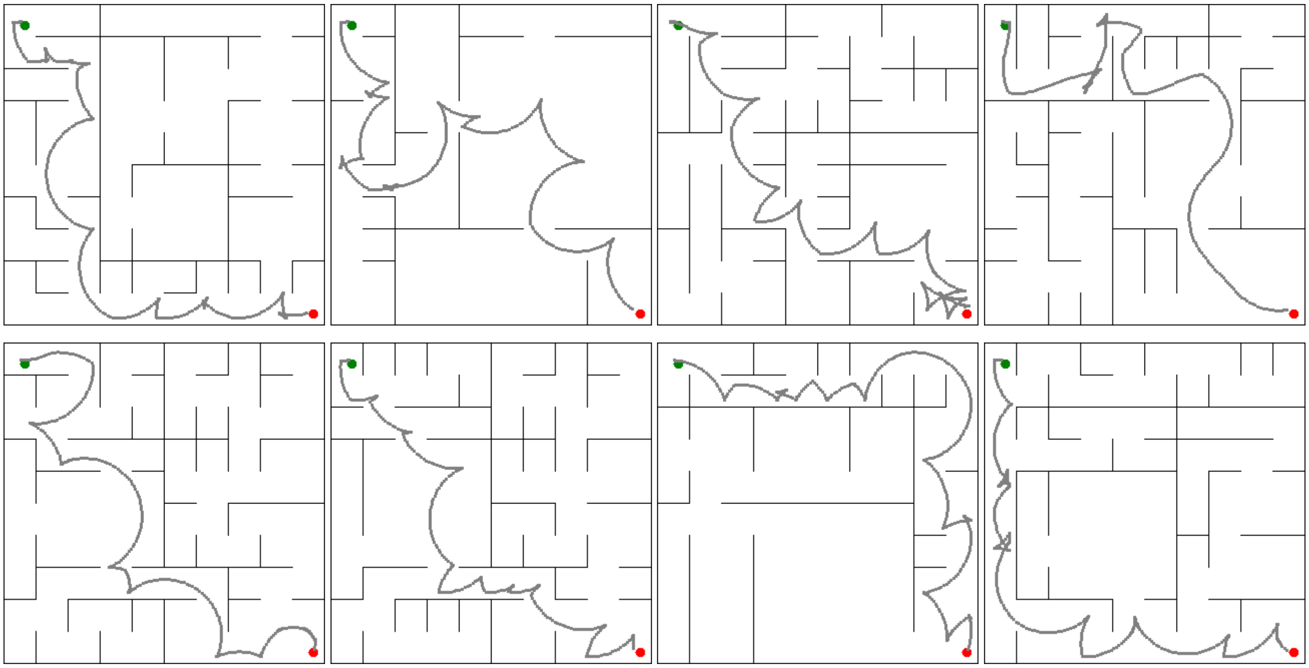


Figure 3: Agent Trajectories from the MCC Speciated Variant. A sample of eight agent trajectories through different solved mazes evolved by the MCC speciated variant during the same run are shown. Intriguingly, this collection exhibits signs of open-ended divergence.

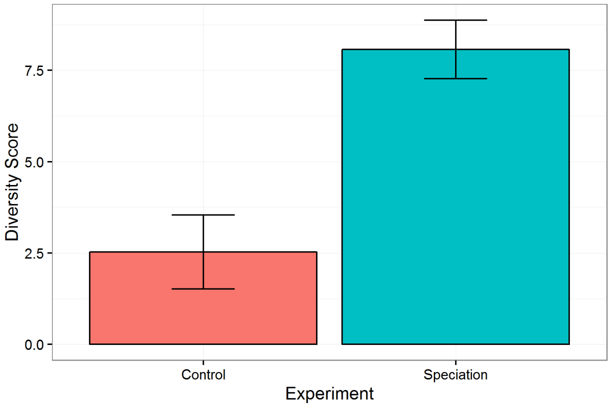


Figure 4: Diversity of Agent Trajectories collected over Full Run (averaged over 20 runs of both variants). Speciated variant configurations produce significantly more diverse solutions over the entirety of a given run.

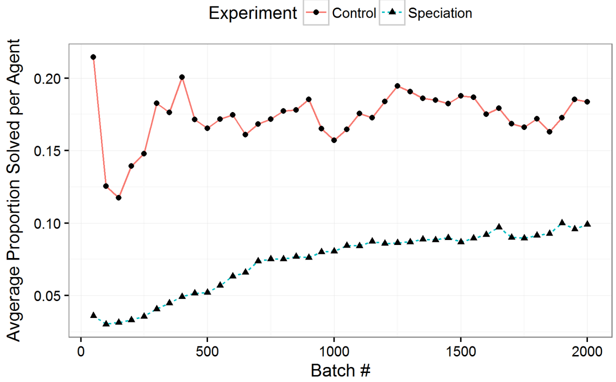


Figure 5: Average Proportion of Mazes Solved per Agent over Evolution (over 20 runs of both variants). The agents in the control population demonstrate the ability to solve a higher proportion of mazes as a result of convergence in the maze population to similar structures, as well as over-training of agents. Conversely, speciated agents solve a lower proportion because their population maintains greater heterogeneity.

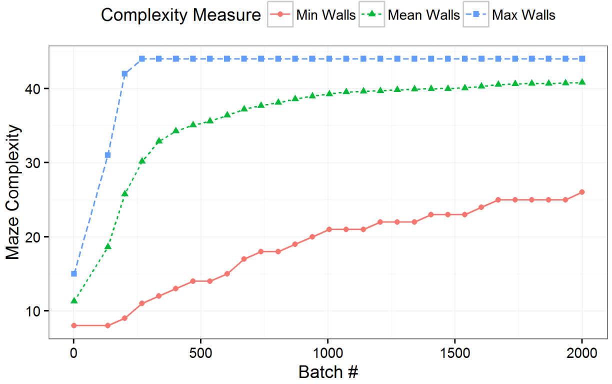


Figure 6: Speciated Maze Complexity Trend. The average minimum, mean, and maximum maze complexity over evolution is shown for the speciated variant. Note the rapid increase in complexity throughout the entire population while also maintaining a healthy *range* of complexities throughout the run. The leveling off of the maximum and mean maze complexities is a side effect of the fixed-sized maze domain, an artificial constraint that was imposed only for this initial investigation.

189–223.

[11] Joel Lehman and Kenneth O Stanley. 2011. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, 211–218.

[12] Joel Lehman and Kenneth O Stanley. 2013. Evolvability is inevitable: Increasing evolvability without the pressure to adapt. *PloS one* 8, 4 (2013), e62186.

[13] Antonios Liapis, Héctor P. Martınez, Julian Togelius, and Georgios N. Yannakakis. 2013. Transforming Exploratory Creativity with DeLeNoX. In *Proceedings of the Fourth International Conference on Computational Creativity*.

[14] Andrea Maesani, Pradeep Ruben Fernando, and Dario Floreano. 2014. Artificial evolution by viability rather than competition. *PloS one* 9, 1 (2014), e86831.

[15] Claudio Mattiussi and Dario Floreano. 2003. *Viability Evolution: Elimination and Extinction in Evolutionary Computation*. Technical Report. EPFL.

[16] Jean-Baptiste Mouret. 2011. Novelty-based multiobjectivization. In *New Horizons in Evolutionary Robotics*. Springer, 139–154.

[17] Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909* (2015).

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

[19] Charles Ofria and Claus O Wilke. 2004. Avida: A software platform for research in computational evolutionary biology. *Artificial life* 10, 2 (2004), 191–229.

[20] Elena Popovici, Anthony Bucci, R Paul Wiegand, and Edwin D De Jong. 2012. Coevolutionary principles. In *Handbook of Natural Computing*. Springer, 987–1033.

[21] Mitchell A Potter and Kenneth A De Jong. 1995. Evolving Neural Networks with Collaborative Species. In *Proceedings of the 1995 Summer Computer Simulation Conference*. 340–345.

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

[23] Thomas S Ray. 1991. An approach to the synthesis of life. In *Artificial Life II*. Addison-Wesley, 371–408.

[24] Sebastian Risi, Sandy D Vanderbleek, Charles E Hughes, and Kenneth O Stanley. 2009. How novelty search escapes the deceptive trap of learning to learn. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2009)*. ACM, 153–160.

[25] Eivind Samuelsen and Kyrre Glette. 2014. Some distance measures for morphological diversification in generative evolutionary robotics. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2014)*. ACM, 721–728.

[26] Lisa B Soros and Kenneth O Stanley. 2014. Identifying necessary conditions for open-ended evolution through the artificial life world of Chromaria. In *Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems (ALife 14)*. Citeseer, 793–800.

[27] Russell K Standish. 2003. Open-ended artificial evolution. *International Journal of Computational Intelligence and Applications* 3, 02 (2003), 167–175.

[28] Kenneth O Stanley and Risto Miikkulainen. 2002. Evolving neural networks through augmenting topologies. *Evolutionary computation* 10, 2 (2002), 99–127.

[29] Danesh Tarapore, Jeff Clune, Antoine Cully, and Jean-Baptiste Mouret. 2016. How do different encodings influence the performance of the MAP-Elites algorithm?. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2016)*. ACM, 173–180.

[30] R Paul Wiegand, William C Liles, and Kenneth A De Jong. 2001. An empirical analysis of collaboration methods in cooperative coevolutionary algorithms. In *Proceedings of the genetic and evolutionary computation conference (GECCO)*, Vol. 2611. 1235–1245.