Augmenting Evolution with Bio-Inspired "Super Explorers"

Vincent Ragusa[1,3,4] and Cliff Bohm[2,3,4]

[1]Department of Computer Science and Engineering
[2]Department of Integrative Biology
[3]BEACON Center for the Study of Evolution in Action
[4]Michigan State University, East Lansing, MI 48824
ragusavi@msu.edu

Abstract

Natural evolving populations experience constantly fluctuating selection strength, which also creates a fluctuating trade-off between exploration and exploitation. Range expansion, for example, creates semi-persistent spatially-distributed differences in selection strength, particularly among the pioneering agents along the leading edge of each range expansion. The pioneers experience reduced selection strength and in turn experience greater potential for exploration, while selection on the remainder of the population ensures that prior discoveries are not lost.

Here we describe a method to augment pre-existing selection algorithms inspired by the exploration-boosting properties of range expansion events. The key insight is that for productive exploration on deceptive landscapes, mutations must be able to accumulate and persist in some, but not all, lineages. We create artificially drifting lineages of "super explorers" and show that they can be used to improve the performance of another selection algorithm.

Introduction

All search algorithms, be they natural selection or computer models of evolution, are subject to the fundamental limitations of the no-free-lunch theorems (Ho and Pepyne, 2002), and particularly to the explore-exploit tradeoff (Millidge et al., 2021). Managing this tradeoff is typically a main concern in the development of computational selection algorithms. In this work, we introduce the "super-explorer method" which can be sued to augment pre-existing selection algorithms. The super-explorer method allows us to tune the trade-off between exploration and exploitation to better align with the ruggedness of a fitness landscape.

The super-explorer method augments other selection algorithms by adding agents ("super explorers"), that are allowed to drift (i.e, freely accumulate mutations). In essence, our method allows a pre-existing selection algorithm to focus on exploitation, while super explorers enhance exploration. We compare three implementations of the super-explorer method on two different kinds of fitness landscapes: an NK-fitness landscape (Kauffman et al., 1993), and a

saw-tooth fitness landscape (Ragusa and Bohm, 2021), and across a wide array of configurations. We show that for both types of fitness function, there are configurations where evolution augmented with super explorers preforms better than without.

Biological inspiration

Natural selection does not act with consistent strength; Shifting balance theory (Wright, 1932, 1982), range expansion (Slatkin and Excoffier, 2012; Peischl et al., 2013; Peischl and Excoffier, 2015; Peischl et al., 2015; Gilbert et al., 2017; Burton and Travis, 2008), environmental noise (Wang and Zhang, 2011; Van Egeren et al., 2018; Ragusa and Bohm, 2021), population size changes (Jain et al., 2011; Ochs and Desai, 2015; Rozen et al., 2008), sexual selection (Bohm et al., 2019), and mass-extinction events (Mathias and Ragusa, 2016; Engholdt and Mathias, 2021) all describe scenarios where selection strength changes over time or space.

As a consequence of naturally occurring fluctuations in selection strength, a population's ability to explore their fitness landscape also fluctuates. Furthermore, during the periods of increased exploration, the population may discover new fitness peaks that, under stronger selection, would have been unlikely or impossible to discover. Range expansion events are one particular scenario that can cause a reduction in selection strength. As a species enters new territory, a lack of competition can result in an accumulation of deleterious mutations in lineages on the leading edge of the expansion for as long as uncolonized territory remains (Burton and Travis, 2008). The continued accumulation of deleterious mutations in one lineage can result in adaptation via valley-crossing, a process known to be critical for evolutionary adaptation (Covert et al., 2013; Oliveto et al., 2018).

In this work, we present a method designed to augment existing selection algorithms with the exploration-boosting power of range expansion, without requiring that populations have spatial structure. In addition, unlike range expansion, the exploratory advantages are maintained indefinitely. The key insight underlying our method is that for sufficient exploration to occur, mutations must be able to ac-

cumulate in some—but not all—lineages before purifying selection acts on them. To achieve this, we introduce super explorers, agents that always have exactly one offspring every generation. As super explorers propagate regardless of fitness, they experience the same mutation-accumulation as organisms at the leading edge of a range expansion.

Similar selection algorithms

The hybrid selection algorithms resulting from augmentation with the super-explorer method share similarities with previously defined selection algorithms.

Particle Swarm Optimization (PSO) (Poli et al., 2007) models agents as particles and simulates forces that attract the particles to the highest observed fitness (analogous to selection). However, the individual particles in PSO do not move directly to the higher gradients, but instead chaotically trend towards it, often exploring areas of the fitness landscape more distant from known optima.

Lexicase selection (Helmuth et al., 2014; La Cava et al., 2016) and real-valued tournament selection (Ragusa and Bohm, 2021) are examples of selection algorithms that probabilistically ignore some or all fitness gradient information. Novelty search (Lehman and Stanley, 2011) foregoes following the fitness gradient altogether and instead focuses on collecting a catalog of functionally distinct solutions, evaluating them for relevance to the fitness function *ex post facto*.

Island models (Whitley, 2001) protect innovations by removing the competitive interactions of agents between separate mutually-exclusive subpopulations, called islands. When all agents are considered together, the overall effect is a reduction in the selection strength. Restricting competition to be within individual islands also creates small-population drift effects that can facilitate valley-crossing.

Systems augmented with super explores are different from the examples above in key ways. While super explorers do experience a form of occasional selection during the decay-replace process (described in Methods), they do not individually experience a fitness gradient, even in the form of selection for novelty. While some other selection algorithms, such as lexicase and real-valued Tournament, can occasionally allow for several generations of drift on some lineages, super explorers are maintained separately from the main population, ensuring their lineages experience sustained periods free of selection. While island models do break up a population into separate pools, they generally restrict selection and reproduction to be within each island. The super-explorer method makes every agent visible to the selection algorithm in use, allowing for the immediate adoption of a beneficial mutation. Furthermore, although island models facilitate drift, it does not result in sustained drift on lineages for long periods as in the super-explorer method.

Methods
Augmenting evolution with super explorers

Super explorers can augment any well-mixed discrete-generation agent-based selection algorithm. In our implementation, super explorers are added to a selection algorithm by dividing a population into two mutually exclusive pools, the *active selection pool* (or *ASP*) and *super-explorer pool* (or *SEP*) (see Fig. 1). The ASP evolves via the rules of an externally defined selection algorithm (such as, roulette, tournament, lexicase, etc.), except that parents can be drawn not only from the ASP, but also from the SEP. On the other hand, the agents in the SEP each produce one offspring, with the standard mutation load, regardless of fitness. While lineages in the ASP survive and perish following the rules of the selection algorithm in use, lineages in the SEP end only by decay. Whenever an agent in the SEP is about to reproduce, there is a chance (the *decay rate*) that the agent will be removed and replaced by some other agent in the population. The number of agents in the ASP and SEP, the decay rate, the process used to replace decayed SEP agents, the mutation settings, and the selection algorithm used in the ASP (and related settings) are all parameters established by the user.
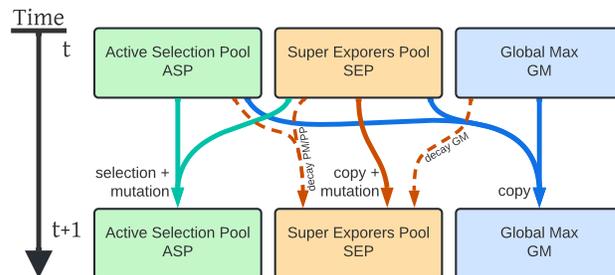

Figure 1: A diagram of an evolving system augmented with super explorers. Here, the population is divided into *the active selection pool* (ASP) and the *super-explorer pool* (SEP). While selection in the ASP (green arrows) is determined by a pre-existing selection algorithm (such as roulette, tournament, lexicase, etc.) with parents drawn from the ASP and SEP, agents in the SEP are free from selection and simply accumulate mutations (red solid arrow). From time to time, agents in the SEP decay and are replaced. Replacements (red dashed arrows) are drawn from either the ASP and SEP, or the global max, depending on the replacement process, where the global max maintains a copy of the highest scoring agent seen to date (blue arrows).

Because the super explorers reproduce regardless of their fitness, they are able to accumulate mutations without consequence. The decoupling of survival from fitness allows super explorers to explore the fitness landscape in directions that selection might prohibit; they may descend into, and possibly cross, fitness valleys. The mutational paths of super explorers are, in fact, random walks (i.e., undirected and unfocused), so they are an inefficient selection algorithm on their own. However, by introducing super explorer decay

and replacement, we provide a degree of focus to help keep the search process "on track." The search pattern of the super explorers can be seen as a genotypic radiation where the center of the radiation is determined by the replacement process, the radiation distance is determined by the decay rate, and the density of the search is determined by the number of agents in the SEP.

Since the selection algorithm used to choose parents in the ASP has access to both the ASP and the SEP, super explorers can migrate into the ASP if their fitness is competitive.Thus, a selection algorithm augmented with super explorers has the advantage of simultaneously exploring freely (in the SEP) while exploiting discoveries (in the ASP), compared to a non-augmented selection algorithm that must balance the explore-exploit trade-off with only an ASP.

Tunable parameters of super explorer systems

When employing the super-explorer method, the user must choose a selection algorithm to augment, as well as a mutation scheme. These choices will introduce a number of parameters. In addition, the supper-explores methods introduces three more parameters: SEP size, decay rate, and replacement process, which we describe below.

Parameter: super-explorer pool size   SEP size sets the number of agents in the SEP and can be set to any integer value, from 0 to population size. SEP size controls the intensity of the search conducted by the super explorers. Note that ASP size is not a parameter and is simply defined as population size minus SEP size.

Parameter: decay rate   Decay rate is the per-generation probability that each agent in the SEP is replaced, and directly controls the explore-exploit trade-off of the SEP. The decay rate can be set to any value r, such that $0 \leq r \leq 1$. The decay rate determines the average amount of time, $\bar{t} \sim 1/r$, that a super-explorer lineage persists before replacement, which in turn determines how far super explorers can drift (i.e., explore). While lower decay rates can result in more exploration, they can also be wasteful if the valleys in the fitness landscape do not require such distant explorations.

Parameter: replacement process   The replacement process defines the process used to replace a decayed agent in the SEP. In this work, we consider three replacement processes, shown in Fig. 1, though many others are possible.

The first process, the *PopSelect* (PS) process, replaces decayed SEP agents with a new agent generated using the same selection algorithm used for reproduction in the ASP.

The second process, the *PopMax* (PM) process, replaces decayed SEP agents with the max of the population (i.e., the max of all the agents in the combined ASP *and* SEP).

The third process is the *GlobalMax* (GM) process. In addition to the ASP and SEP, this replacement requires that the global max, a copy of the highest fitness agent seen to date in either pool, is maintained. The global max is used to replace decayed SEP agents.

Fitness functions

We use two types of fitness functions to collect performance data for the parameter sweeps. First, we chose a typical NK-landscape to provide an intuition regarding the performance of super explores in a familiar context. Then, we turn to a saw-tooth function to provide a different perspective on the operation of the super-explorer method under more controlled conditions.

Fitness function: NK-landscape   The first fitness function is the well studied NK-landscape (Kauffman et al., 1993). This landscape has a rugged topology and a global optimum. The NK function, $\text{NK\_eval}(i, g_i | N, K)$, decides the fitness contribution of the i-th gene, g_i. The fitness landscape is randomly generated with parameters $N = 20$ and $K = 5$ which control genome size, $G = \{g_1, ..., g_N\}$, and strength of epistasis respectively. The score assigned to each organism by NK is given by

$$\text{score}(G) = \frac{1}{N} \sum_{i=1}^{N} \text{NK\_eval}(i, g_i | N, K) \qquad (1)$$

Addressing diminishing returns   The NK-landscape exhibits diminishing returns (Orr, 2005), which means that the ratio of beneficial to deleterious mutations and the fitness gain per beneficial mutation decreases as fitness increases. As a result, the optimal explore-exploit ratio changes over the course of evolution. Strong exploitation initially maximizes the rate at which fitness increases, but later is likely to result in getting stuck on local optima. Conversely, stronger exploration, though slower at first, can achieve higher final performance in the long run, because it is able to escape local optima.

In order to disambiguate the effects of diminishing returns from other dynamics, we included the saw-tooth landscapes, which do not exhibit diminishing returns and allow us to study the behavior of a system with a constant difficulty.

Fitness function: saw-tooth landscape   The second fitness function, the saw-tooth fitness function, (Ragusa and Bohm, 2021) defines a saw-tooth mapping function from genomic values to scores along an infinite set of ever higher fitness peaks (like the teeth of a saw) as shown in Fig. 2.

The saw-tooth landscape has two special properties. First, it has no global optimum. Second, every fitness valley is self-similar; regardless of absolute position on the fitness landscape, the difficulty of descending into and the benefit of crossing every valley is the same. These two properties together result in a landscape that presents a constant

challenge to an evolving population, without diminishing returns. In this landscape, agents have 10 independent genes, $G = \{g_1, ..., g_{10}\}$, where each gene is a single integer value. Applying the mapping function to each gene results in 10 gene scores that are summed to produce the agent's overall score. The saw-tooth function $\text{saw}(x|w, p, b)$ is specified by a valley width (w), a fitness penalty per mutation into each valley (p), and a fitness benefit per valley crossed (b):

$$\text{score}(G) = \sum_{i=1}^{10} \text{saw}(g_i|w, p, b) \quad (2)$$

Here we test four versions of saw-tooth functions. In all four, $p = -0.05$ and the $b = 1.0$. We varied w to create four versions of the function with different difficulties, from $w = 4$ (easy) to $w = 7$ (hard). These functions are shown in Fig. 2.
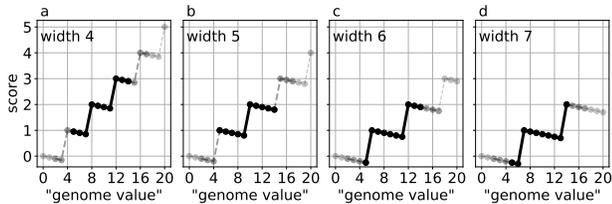

Figure 2: The saw-tooth functions that map gene values to "scores", used in Fig. 4. Panels [a] through [d] show the valley widths (w) 4 through 7, respectively. The values for penalty ($p = -0.05$) and benefit ($b = 1.0$) are the same in all 4 functions.

Roulette selection

We use roulette selection in the ASP to determine which parents in the current population will produce the next generation. The fitness of every agent in the current population (i.e., all agents in the ASP and SEP) is calculated, and then for each new ASP agent, each parent has a chance to reproduce proportional to their share of the total population fitness. Roulette selection is known to suffer from a diminishing selection pressure as the absolute value of agents' scores increases during evolution. In order to ensure that equivalent relative increases to score result in the same relative increase in offspring production, the scores generated from both the NK-landscape and the saw-tooth landscape are exponentiated before fitness shares are determined (i.e., when agent fitness is calculated).

$$f(G) = \exp(\text{score}(G)) \quad (3)$$

Experiment conditions

In order to determine the effectiveness of super explorers, we compare roulette selection with and without super-explorer augmentation on two classes of fitness function (NK and saw-tooth). We use a total population (ASP + SEP)

size of 1024 in all experiments. For each combination of selection regime and function, we run a three-dimensional sweep of SEP size, decay rate, and replacement process. We run decay rates {0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0}, pool sizes {0, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024}, and replacement processes {GM, PM, PS}.

We run 101 replicates for each condition using the NK-fitness landscape ($N = 20$ and $K = 5$) for 10,000 generations and record the final maximum fitness from each replicate. Each replicate evolves on the *same* NK landscape, which is randomly generated *a priori*. A bit genome was used where mutations are introduced with a bit-flip mutation operator, with a 0.0005 per-site mutation rate (i.e., a 0.01 per-agent mutation rate).

We run one replicate for each condition using the saw-tooth landscape for 40,000 generations and record the number of valleys crossed (running one replicate on the saw-tooth function and counting the number of valleys crossed efficiently generates the same results as running many replicates of shorter duration (Ragusa and Bohm, 2021)). In addition to the parameter sweep described above, we used four saw-tooth functions with $w = \{4, 5, 6, 7\}$ (shown in Fig. 2). An integer genome was used where mutations are introduced with a point-offset mutation operator that modifies a genome value by ± 1, with a 0.05 per-site mutation rate (i.e., a 0.5 per-agent mutation rate, as each agent has 10 loci).

The MABE evolution framework was used to run experiments (Bohm et al., 2017). Code and instructions to allow for replication can be found at `https://github.com/cliff-bohm/SuperExp_ALIFE_2022`.

Results

NK landscape

Figure 3 displays the data collected from 101 replicates run on the NK-landscape with parameters $N = 20$ and $K = 5$. The figure presents three grayscale maps (labeled [1], [2], and [3]) each showing results generated using a different replacement process (GM, PM, and PS). The cells in each map show the averages, across replicates, of the maximum fitness detected at the end of each replicate. Note that values in each plot associated with SEP size = 0 are the same, since decay rate only matters if there are super explorers. The fitness differences between the max-fitness, min-fitness, and control configurations of each panel are computed and checked for significance with a two sample z-test (shown in Table 1).

The NK-landscape results show that the GM and PM processes provide similar results, both of which are quite different from the PS data. In the GM and PM results, we see that the addition of any super explorers improves performance under almost all conditions. In addition, as SEP size increases, low decay rates tend to improve adaptation while high decay rates tend to have the opposite effect. In fact, a population made up entirely of super explorers (SEP
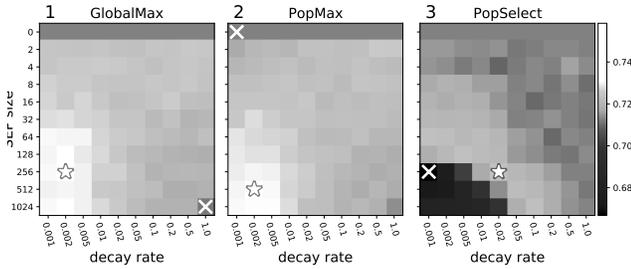
Figure 3: Average maximum final fitness on an NK landscape from 101 replicates using super explorers with three replacement processes: [1] GM, [2] PM, and [3] PS. Lighter shades correspond with higher final fitness, and darker shades correspond with lower final fitness. The color range is set so that gray is associated with the control condition where SEP size = 0 (i.e., no super explorers). The stars and crosses indicate the highest-fitness and lowest-fitness configuration of each panel, respectively. The largest increase and decrease of fitness, relative to the control, are shown in Table 1.

size = 1024) combined with decay rate = 1.0, shows little to no signs of improvement over a non-augmented population.

We note a band of relatively high performance in the PS data, highlighting the conditions with the greatest final scores. This band appears to show a trade-off between SEP size and decay rate. To the right and above the high-performance band, we see values that conform closely to the non-augmented system, while to the left and below the band, we see that final recorded scores are lower than the non-augmented system. The area of low values correlates with a large SEP size and low decay rate.

Note that the values in the column associated with decay rate = 1.0 in the PS data all match the values associated with SEP size = 0. In these conditions, SEP agent replacement happens every generation and uses the same replacement process used in the ASP. As a result, the two pools act as a single ASP (small fluctuations are the result of sampling noise).

Decay Method	Largest Increase	Largest Decrease
GM	+2.76% *	−0.15%
PM	+2.61% *	±0.00%
PS	+1.51% *	−6.44% *

Table 1: The fitness differences between the max-fitness, min-fitness, and control configurations of the NK-landscape data in Figure 3. '*' indicates a p-value of $p < 1. \times 10^{-6}$ (two sample z-test).

Saw-tooth landscape

Figure 4 shows grayscale maps illustrating the number of valleys crossed at the end of the 40,000 generations, with agents evolved on saw-tooth fitness functions. Each of the letters [a] though [d] indicate a different valley width, from 4 to 7. The numbers [1] through [3] indicate the replacement process used (GM, PM, and PS). Note that values in
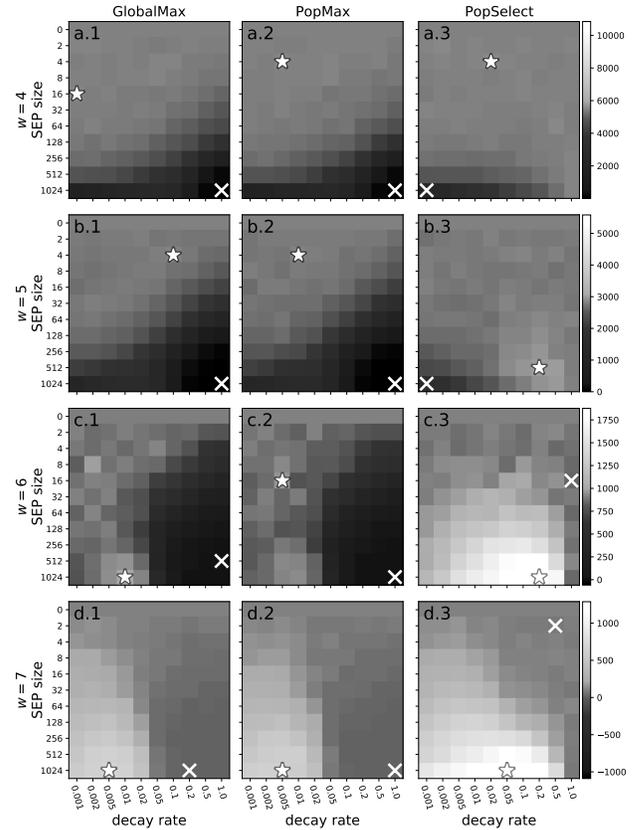

Figure 4: Total number of valleys crossed after 40k generations on saw-tooth landscapes. Horizontal panel groups [a], [b], [c], and [d] show data for $w = \{4, 5, 6, 7\}$ saw-tooth functions, respectively. Vertical panel groups [1], [2], and [3] show data for replacement processes GM, PM, and PS, respectively. The stars and crosses indicate the highest-fitness and lowest-fitness configuration of each panel, respectively. Note: the scales on each color bar (i.e., for each row) are different.

each panel associated with SEP size = 0 are the same within each row, since decay rate only matters if there are super explorers.

As we saw in the NK data, the saw-tooth landscape results show that the GM and PM processes provide similar results that are quite different from the PS data.

Across all panels of the saw-tooth data, there are vertical trends, associated with SEP size, and horizontal trends, associated with decay rate. As we move from top to bottom and introduce a higher ratio of super explorers (larger SEP size) relative to the size of the ASP, we generally see a smooth transition from SEP size = 0 to SEP size = 1024. Conversely, as we move from low decay rates to higher decay rates, particularly larger SEP sizes, we see that the change in performance is not as simple to describe. In all GM and PM results, while we see that large SEP combined with a high decay rate reduces fitness gain relative to the non-augmented system, the effect of large SEP and low decay rate is not

constant. Rather, in the data for large SEP and low decay rate, the final scores flip from worse than non-augmented to better as we move from less deceptive to more deceptive landscapes.

In the PS results Fig. 4[a,b,c,d][3], we see a different trend. As in the NK data, when decay rate = 1.0 agents in the SEP experience the same selection as agents in the ASP (i.e., replaced every generation using the selection algorithm from the ASP). Therefore, as in the NK data, the two pools act as a single ASP (small fluctuations are the result of sampling noise). Unlike the NK data, we see that larger SEP sizes perform as well as or better than the non-augmented system, except in the lower left of [a, b].[3] where SEP is very large, and decay rate is very low. The highest rates of valley crossing appear to correlate with large SEP sizes and middling decay rates (i.e., around 0.05 to 0.1).

Discussion

In this work, we show that augmenting a selection algorithm with super explorers can result in improved adaptation to a range of landscapes and parameter settings. In this discussion, we are primarily interested in describing the super-explorer augmentation method.

The conditions where SEP size = 0 show the behavior of the non-augmented system. In these conditions, there are no super explorers, so decay rate has no effect. Hereafter we will refer to these conditions as the 'control' and they will serve as the baseline for all of our comparisons.

In our experiments, we test conditions with different SEP sizes, but we do not alter the number of agents in the total population or the mutation inflow per agent. Therefore, the populations always experiences the same number of agent evaluations and have the same potential for discovery in the form of mutations. As a result, differences in performance between the control and other conditions must have some other explanation. As we will see, the observed differences can be explained in terms of a trade-off between exploration and exploitation.

The explore-exploit trade-off of super explorers

In evolutionary systems (natural or computational) the explore-exploit trade-off explains how the conditions that allow for effective navigation on smooth fitness landscapes hinder navigation on deceptive fitness landscapes and *vice versa*. On a smooth landscape the paths towards higher fitness are easy to discover, so unfocused distant explorations are wasteful: it is prudent to spend energy exploiting the local fitness gradient. Conversely, on deceptive landscapes the paths to higher fitness often require crossing fitness valleys, so focusing on local search only results in short-term superficial gains: it is prudent to spend time on long shots.

Our initial inspiration for the super-explorer method was the biological phenomenon of range expansion. There, we observed that the freedom from selection along the leading edge of a range expansion creates lineages in which mutations could accumulate, while the remainder of the population experiences purifying selection. To mimic this effect, we designed a method that divides a population into two pools: the SEP that specializes in exploration, and the ASP that specializes in exploitation.

Decay rate Generally speaking, the amount of genetic change that any lineage can accumulate is related to the strength of selection, and how long the lineage can avoid extinction. SEP lineages (unbroken phylogenies in the SEP) are special in that they are free from selection, and so their potential for genetic change is only limited by how long they persist, which in turn depends on the decay rate. Meanwhile, the decay rate *also* establishes the only selective force at work in the SEP: how often replacements occur. Thus, the conditions likely to enhance exploration, high drift potential and low selection strength, are conditions generated by low decay rates while the conditions likely to enhance exploitation, low drift potential and strong selection, are conditions generated by high decay rates.

Replacement process The three replacement processes we chose to test correlate with a range of selection strengths. The GM process provides the strongest selection strength because it always maintains the best solution. The GM process insures that small fitness changes (potentially undetectable by the ASP's selection algorithm) are always exploited, but this comes at a cost: the inability to forget can inhibit escape from local optima. Moreover, when the GM process is combined with an SEP size equal to population size and decay rate equal to 1.0, the resulting system is synonymous with an elitism selection algorithm; every agent tests one mutation, and then either becomes the new global max or is forgotten.

Compared to the GM process, the PS process results in weaker selection, which cannot exceed the selection strength of the ASP's selection algorithm. This is because at decay rate = 1.0, all agents in the population are replaced using the ASP selection algorithm every generation (regardless of SEP size).

The PM process represents a middle ground in terms of selection strength. However, since we are using a relatively large population size relative to decay rate, forgetting high quality solutions is unlikely, so the GM and PM processes generate similar results.

SEP size The SEP size parameter determines the degree to which each pool (ASP and SEP) drives the system. As SEP size increases, particularly when decay rates are low, there are more low quality solutions in the total population, which increases the chance that low quality solutions will be selected during ASP reproduction. As a result, larger SEP sizes result in weaker selection in the ASP. This demon-

strates another way that low decay rates can enhance exploration.

Analyzing the NK data

GM and PM Processes   Fig. 3[1,2] show the results of using the GM and PM processes to evolve populations on the NK-landscape. We see that enhancing exploration with low decay rate improves performance, while maximizing exploitation, with either low SEP size or high decay rate, results in performance that is not significantly different from the control. The fact that the results of elitism are similar to the control suggest that the ASP is already generating elitist-like behavior that hinders exploration. It follows then that improved adaptation associated with lowered decay rates results from increased exploration.

The sudden jump in fitness from SEP = 0 to SEP = 2 in Fig. 3[1,2] occurs because the GM and PM processes select max *perfectly* when making replacements into the SEP. Using the GM process, it is *impossible* to forget a high-value solution once it has been discovered; forgetting is only very unlikely when using the PM process. As a result, the GM and PM process are able to identify and exploit fitness improvements that the ASP selection algorithm alone may not detect.

Finally, the decrease in fitness from top to bottom of the right-most column (decay rate = 1.0) shows that the behavior of the ASP alone is not exactly the same as elitism. While the ASP can allow for the accumulation of small (i.e., nearly-neutral) deleterious mutations, a system experiencing elitism can never move in a direction that results in *any* loss of fitness. As a result, the highest fitness in this column occurs at SEP size = 2, indicating that the best performance is generated by a large ASP (that does have the ability to explore, if only locally), augmented by a SEP that does not forget (but cannot explore).

PS Process   Fig. 3[3] shows the results of the PS process used to evolve populations on the NK-landscape. The low fitness recorded in the low left is the result of weak selection that is unlikely to find effective solutions, and it is likely to forget what it does find. The most interesting feature in this panel is the relatively bright band—indicating the highest scores—that appears to show a trade-off between SEP size and decay rate. SEP size = 1024 and decay rate = 0.1 marks the bottom of this band, and suggests that lineages that survive for about 10 generations are optimal when the entire population is in the SEP. As we decrease the size of the SEP, we see that lower decay rates, (i.e., drifting lineages that persist for longer) are optimal. This is because the larger ASP results in more exploitation and less exploration. Apparently, the trade-off exists because when there are fewer SEP agents they need more time to drift in order to make discoveries, but larger ASPs are better at exploiting beneficial fitness gradients. The entire right side of the panel evolves

similarly to the control because along this edge agents are replaced every generation using the same algorithm used in the ASP.

Analyzing the saw-tooth data

GM and PM Processes   Fig. 4[a].[1,2] show the results of the GM and PM processes on a saw-tooth function with only limited deception. Here we see that small SEP sizes generate the best performance, while larger SEP size degrades performance. This indicates that the ASP selection settings are well tuned for this function, and evolution does not improve with the addition of super explorers. Moreover, elitism results in the lowest levels of performance, since valley crossing is required to optimize this function. In Fig. 4[b,c,d].[1,2] we find the results from the other three saw-tooth landscapes, each with an increasing level of deception. As the functions become harder to adapt to, we see that the control does not produce the best performance because it does not allow enough exploration. As a consequence, larger SEP sizes become more effective. In Fig. 4[d] (the hardest function), the best performance occurs when the entire population is SEP agents and SEP lineages persist for about 200 generations (decay rate = 0.005). This suggests that navigation on this landscape requires a high degree of exploration. The fact that the highest performance values are not at the lowest decay rates means that too much exploration, beyond what is necessary, is wasteful.

PS Process   The results generated by the saw-tooth landscape using the PS process are displayed in Fig. 4[a,b,c,d][3].   As in the NK data, and for the same reason, the entire right side of the panel evolves similarly to the control. However, the main results differ significantly from the NK-landscape PS process results (in part due to the absence of diminishing returns — see Methods). Except in the case of the least deceptive function ([a]), the best performance across the three replacement processes is found in the PS data. This supports the idea that weakened selection is beneficial for valley crossing on the harder saw-tooth functions.

Super explorers alone

The bottom row of each panel in Figures 3 and 4 corresponds to conditions where the entire population is in the SEP; there are no agents in the ASP. The high performance observed in some SEP-only configurations demonstrates that the combination of drift and replacement is a viable search process in its own right. In fact, for several of the fitness functions, a configuration with SEP = population size achieves maximum or nearly maximum performance. We call this new drift-and-replace search process "Drift Pool Optimization" or DPO.

There is a surprising similarity between DPO and particle swarm optimization (PSO). Both DPO and PSO operate as

a swarm of particles, as opposed to implementing evolution by natural selection. The main difference is that while the particles in PSO tend to converge towards known optima, the particles in DPO originate near current optima and are allowed to drift. In both cases, the degree to which solutions can diverge from know optima is critical to success: too little divergence will stifle innovation, while too much diffusion will result in chaotic behavior incapable of effectively making discoveries.

Super explorers and recombination

In addition to the experiments shown in this work, we tested the saw-tooth functions with recombination (results not shown). When producing offspring in the ASP and when picking replacements using the PS process, we select two parents and perform three-point recombination to generate offspring (while GM and PM processes replacements are still asexual). We found that while scores with recombination tended to be higher, the trends in the data were almost identical.

Extending the super-explorer method

In this work, we presented the super-explorer method, an augmentation that can be used to enhance pre-existing selection algorithms. Here we present some potential modifications and extensions.

While a super-explorer-augmented system is already able to simultaneously support enhanced exploration (in the SEP) and exploitation (in the ASP), the method could be enhanced so that it automatically modifies both decay rate and SEP size based on observed rates of fitness gain in an adaptive manner, for example by monitoring the time the population spends in stasis.

Another possible modification to the super explorer-method would be to encourage diversity in the SEP, which would ensure that the SEP explores genetic space more uniformly. Both genetic diversity and phenotypic diversity could be investigated. Furthermore, an archive of high fitness agents, like the kind used in MAP-Elites(Mouret and Clune, 2015), could replace the global max pool and offer a more diverse alternative to elitism.

An interesting alternative to super-explorer augmentation would be to simply use another selection algorithm in place of the SEP. This would allow two selection algorithms with different behaviors to synergize. We are particularly interested in investigating augmenting Particle Swarm Optimization (PSO) with super explorers, and also in using PSO in place of the SEP to augment other selection algorithms.

There is no reason that a system must be limited to only two selection algorithms. A population could be subdivided into any number of pools, each acting as an ASP or SEP with unique selection algorithms and parameterization. Automation in the form of pool-size balancing and parameter adjustments could also be considered.

Finally, super explores could be used to augment selection algorithms that do not have discrete generations, but instead implement overlapping or other less standard evolutionary modules. In these cases, the method would need to be enhanced to determine when reproduction and replacement in the SEP should occur. The method could otherwise be unaltered.

Using the super-explorer method to study evolution

The super-explorer method could be used to further our understanding of general evolutionary processes related to the trade-off between exploration and exploitation. Since the behavior of the SEP can be related directly to the explore-exploit trade-off, super explorers could be used to gauge the relative explore-exploit trade-off of various selection algorithms. For example, in Fig. 4[c,d][1,2], SEP size = 1024 with decay rate = 0.05 performs about as well as the control. We theorize that the explore-exploit trade-off in these conditions is similar, but further investigations would be required to support this conjecture. If correct, we believe this hypothesis could be extended to state that any conditions that share performance values likely also experience similar explore-exploit trade-offs.

In this work, we only considered a single selection algorithm for the ASP: roulette selection. Other configurations of roulette-wheel selection, representing different selection strengths, as well as other selection algorithms should be tested. This approach could allow us to ask a number of different questions, such as: how does tournament size affect the explore-exploit trade-off?

Conclusion

In this work, we introduced a new bio-inspired optimization technique called "the super-explorer method" and we demonstrated the efficacy of the method on a number of deceptive fitness landscapes. There is a wealth of literature exploring how population size, selection strength, and mutation rate affect rates of adaptation. In addition to these, we argue that the frequency at which selection is applied and the distance a lineage can drift before it is evaluated also affect the success of a selection algorithm. Other selection algorithms have experimented with changing the frequency with which selection is applied (e.g., lexicase and real-valued tournament) or ignoring fitness altogether (e.g., novelty search), but we believe we are the first to design a method that intentionally protects lineages in order to promote discovery by drift.

Acknowledgments

References

Bohm, C., Ackles, A. L., Ofria, C., and Hintze, A. (2019). On sexual selection in the presence of multiple costly displays. In *ALIFE 2019: The 2019 Conference on Artificial Life*, pages 247–254. MIT Press.

Bohm, C., C G, N., and Hintze, A. (2017). MABE (modular agent based evolver): A framework for digital evolution research. *Proceedings of the European Conference of Artificial Life*, pages 76–83.

Burton, O. J. and Travis, J. M. (2008). The frequency of fitness peak shifts is increased at expanding range margins due to mutation surfing. *Genetics*, 179(2):941–950.

Covert, A. W., Lenski, R. E., Wilke, C. O., and Ofria, C. (2013). Experiments on the role of deleterious mutations as stepping stones in adaptive evolution. *Proceedings of the National Academy of Sciences*, 110(34):E3171–E3178.

Engholdt, K. and Mathias, H. D. (2021). A biologically-inspired model for mass extinction in genetic algorithms. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, pages 1078–1085. IEEE.

Gilbert, K. J., Sharp, N. P., Angert, A. L., Conte, G. L., Draghi, J. A., Guillaume, F., Hargreaves, A. L., Matthey-Doret, R., and Whitlock, M. C. (2017). Local adaptation interacts with expansion load during range expansion: maladaptation reduces expansion load. *The American Naturalist*, 189(4):368–380.

Helmuth, T., Spector, L., and Matheson, J. (2014). Solving uncompromising problems with lexicase selection. *IEEE Transactions on Evolutionary Computation*, 19(5):630–643.

Ho, Y.-C. and Pepyne, D. L. (2002). Simple explanation of the no-free-lunch theorem and its implications. *Journal of optimization theory and applications*, 115(3):549–570.

Jain, K., Krug, J., and Park, S.-C. (2011). Evolutionary advantage of small populations on complex fitness landscapes. *Evolution: International Journal of Organic Evolution*, 65(7):1945–1955.

Kauffman, S. A. et al. (1993). *The origins of order: Self-organization and selection in evolution*. Oxford University Press, USA.

La Cava, W., Spector, L., and Danai, K. (2016). Epsilon-lexicase selection for regression. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pages 741–748.

Lehman, J. and Stanley, K. O. (2011). Novelty search and the problem with objectives. In *Genetic programming theory and practice IX*, pages 37–56. Springer.

Mathias, H. D. and Ragusa, V. R. (2016). An empirical study of crossover and mass extinction in a genetic algorithm for pathfinding in a continuous environment. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 4111–4118. IEEE.

Millidge, B., Tschantz, A., Seth, A., and Buckley, C. (2021). Understanding the origin of information-seeking exploration in probabilistic objectives for control. *arXiv preprint arXiv:2103.06859*.

Mouret, J.-B. and Clune, J. (2015). Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*.

Ochs, I. E. and Desai, M. M. (2015). The competition between simple and complex evolutionary trajectories in asexual populations. *BMC evolutionary biology*, 15(1):1–9.

Oliveto, P. S., Paixão, T., Heredia, J. P., Sudholt, D., and Trubenová, B. (2018). How to escape local optima in black box optimisation: When non-elitism outperforms elitism. *Algorithmica*, 80(5):1604–1633.

Orr, H. A. (2005). The genetic theory of adaptation: a brief history. *Nature Reviews Genetics*, 6(2):119–127.

Peischl, S., Dupanloup, I., Kirkpatrick, M., and Excoffier, L. (2013). On the accumulation of deleterious mutations during range expansions. *Molecular ecology*, 22(24):5972–5982.

Peischl, S. and Excoffier, L. (2015). Expansion load: recessive mutations and the role of standing genetic variation. *Molecular ecology*, 24(9):2084–2094.

Peischl, S., Kirkpatrick, M., and Excoffier, L. (2015). Expansion load and the evolutionary dynamics of a species range. *The American Naturalist*, 185(4):E81–E93.

Poli, R., Kennedy, J., and Blackwell, T. (2007). Particle swarm optimization. *Swarm intelligence*, 1(1):33–57.

Ragusa, V. R. and Bohm, C. (2021). Connections between noisy fitness and selection strength. In *ALIFE 2021: The 2021 Conference on Artificial Life*. MIT Press.

Rozen, D. E., Habets, M. G., Handel, A., and de Visser, J. A. G. (2008). Heterogeneous adaptive trajectories of small populations on complex fitness landscapes. *PloS one*, 3(3):e1715.

Slatkin, M. and Excoffier, L. (2012). Serial founder effects during range expansion: a spatial analog of genetic drift. *Genetics*, 191(1):171–181.

Van Egeren, D., Madsen, T., and Michor, F. (2018). Fitness variation in isogenic populations leads to a novel evolutionary mechanism for crossing fitness valleys. *Communications biology*, 1(1):1–9.

Wang, Z. and Zhang, J. (2011). Impact of gene expression noise on organismal fitness and the efficacy of natural selection. *Proceedings of the National Academy of Sciences*, 108(16):E67–E76.

Whitley, D. (2001). An overview of evolutionary algorithms: practical issues and common pitfalls. *Information and software technology*, 43(14):817–831.

Wright, S. (1932). The roles of mutation, inbreeding, crossbreeding, and selection in evolution. *Proc. 6th Intern. Congress Genetics*, 1:356–366.

Wright, S. (1982). The shifting balance theory and macroevolution. *Annual review of genetics*, 16(1):1–20.