Effective Mutation Rate Adaptation through Group Elite Selection

Akarsh Kumar, Bo Liu, Risto Miikkulainen, Peter Stone

Abstract

Evolutionary algorithms are sensitive to the mutation rate (MR); no single value of this parameter works well across domains. Self-adaptive MR approaches have been proposed but they tend to be brittle; for example, they sometimes decay the MR to zero, thus halting evolution. To make self-adaptive MR robust, this paper introduces the Group Elite Selection of Mutation Rates (GESMR) algorithm. GESMR co-evolves a population of solutions and a population of MRs, such that each MR is assigned to a group of solutions. The resulting *best* mutational change in the group, instead of average mutational change, is used for MR selection during evolution, thus avoiding the vanishing MR problem. With the same number of function evaluations and with almost no overhead, GESMR converges faster and to better solutions than previous approaches on a wide range of continuous test optimization problems. GESMR also scales well to high-dimensional neuroevolution for supervised image-classification tasks and for reinforcement learning control tasks. Analysis of the distribution of function changes during mutation explains why self-adaptation is prone to premature convergence and how GESMR overcomes this issue. Empirically, GESMR produces MRs that are *optimal in the long-term*, as demonstrated through a comprehensive look-ahead grid search. GESMR and the analysis have theoretical and practical implications for the fields of artificial life and evolutionary computation.

1 Introduction

Biological evolution has produced an incredible diversity of life that is seen everywhere. In this process, the solutions and the mechanisms co-evolve end-to-end, including the mutation rate (MR; Metzgar and Wills 2000). Self-adaptation of MRs (SAMR) is a technique common in the literature of genetic algorithms (GA) that encapsulates this idea of end-to-end evolution of the MR along with the individuals (Meyer-Nieberg and Beyer 2007; Bäck 1992; Smith and Fogarty 1996; Dang and Lehre 2016). The idea is to assign each individual its own MR, creating a pair. The pairs are then evolved end-to-end using the assigned MR for mutating the individual and a "meta" MR for mutating the assigned MR.

However, this approach often runs into the problem that the MRs produced decay to zero, causing evolution to stop at a sub-optimal value. If instead the MR were fixed at some moderate value, evolution would continue and find a better function value (Clune et al. 2008; Rudolph 2001; Glickman and Sycara 2000). This premature convergence can be attributed to the fact that most mutations hurt the fitness of an individual (Clune et al. 2008), and thus an effective way for an individual to preserve its fitness into the next generation is to have no mutation. Thus, SAMR ignores the long-term goal of evolution to explore the fitness landscape and find better solutions in future generations (Clune et al. 2008).

To counteract this effect, this paper proposes a novel GA based on supportive co-evolution (Goldman and Tauritz 2012) of solutions and MRs, entitled Group Elite Selection of Mutation Rates (GESMR). After assigning each MR to a group of solutions, the solutions are evolved using that MR, and the MRs are evolved according to the *best* change in function value from the MR's solution group, defined as the "group elite". By targeting the MR that produces the *best* change in function value, given many mutation samples, GESMR can mitigate the vanishing MR problem. Additionally, GESMR is straightforward to implement and requires no more function evaluations than a fixed MR GA, and thus can be applied to a wide range of GA problems.

In prior work, a related approach using the idea of group elites was formulated as a multi-armed bandit problem and applied to entire genetic operators in an ad-hoc manner (Fialho et al. 2008; Whitacre, Pham, and Sarker 2009). In contrast, this paper demonstrates that the approach is most effective when focused on MRs, and it also makes it possible to understand this result both empirically and theoretically.

Evaluation of GESMR is performed on common benchmark test optimization problems from the GA literature. To show that the method scales well to harder problems, it is also evaluated on neuroevolution for image classification in the MNIST/Fashion-MNIST domain and on reinforcement learning for control in the CartPole, Pendulum, Acrobot, and MountainCar domains. For comparison, results of several adaptive MR algorithms including an oracle optimal fixed MR, an oracle look-ahead MR (that uses foresight to determine MR), self-adaptive MR, the multi-armed bandit method (Fialho et al. 2008), and some common heuristic methods (Rechenberg 1978) are also reported.

GESMR outperforms other algorithms in most tasks. Even when SAMR prematurely converges, like in problems with especially rugged fitness landscapes (Clune et al.

2008), GESMR does not. As a matter of fact, GESMR performs as well as the oracle look-ahead MR in function value and even matches the MR to the empirically estimated *long-term optimal MR*. To explain why, the statistical distribution of the change in function value for a spectrum of MRs for different function landscapes is empirically analyzed and visualized. This analysis shows that SAMR is minimizing an MR objective whose optimal MR is zero in rugged landscapes, while GESMR is minimizing an objective whose optimal MR is nonzero.

2 Related Work

Research on mutation rates (MRs) is one of the most studied sub-fields of genetic algorithms (Aleti and Moser 2016; Eiben, Hinterding, and Michalewicz 1999; Karafotias, Hoogendoorn, and Eiben 2015; Kramer 2010; Hassanat et al. 2019; Bäck and Schütz 1996).

Fixed MRs: Lots of theoretical and empirical work has been done on finding the optimal fixed MR for specific problems (Greenwell, Angus, and Finck 1995; Böttcher, Doerr, and Neumann 2010), finding heuristics like the MR should be proportional to $1/L$ where L is the length of the genotype (Ochoa 2002; Doerr, Doerr, and Lengler 2019). Evolutionary bilevel optimization tries to find the optimal evolutionary parameters, including MR, by running an inner evolution with an outer loop searching over parameters (Sinha, Malo, and Deb 2018; Liang and Miikkulainen 2015). However, it is commonly known that the optimal MR is constantly changing during evolution (Patnaik and Mandavilli 1986).

Deterministic MRs: Deterministic MRs are common but these are ad hoc functions to change the MR as a function of the number of generations, and may not generalize to unseen problems with different landscapes (Aleti and Moser 2016).

Adaptive MRs: Adaptive MRs are also common (Thierens 2002; Srinivas and Patnaik 1994; Patnaik and Mandavilli 1986; Doerr, Doerr, and Lengler 2019; Sewell et al. 2006) but these rely on another ad hoc system to determine how to alter the MR given feedback from the evolution. A common technique is to maintain a MR that produces mutations of which only one-fifth are beneficial (Karafotias, Hoogendoorn, and Eiben 2015; Rechenberg 1978), by increasing MR when the percentage of successful mutations is greater than $1/5$ (and vice versa). Although this technique is based on empirical findings, it is ad-hoc, does not generalize to different landscapes, requires a hard-coded threshold, and has been shown to lead to premature convergence when elitism is employed (Rudolph 2001).

Self-Adaptive MRs: Perhaps the most promising and evolutionarily plausible class of adapting MRs is that of self-adapting MRs (Kramer 2010; Aleti and Moser 2016; Bäck 1992; Gomez 2004; Thierens 2002). This technique concatenates an MR to each individual and evolves the MRs and individuals in one end-to-end evolutionary process. However, many previous works have shown this process to be brittle and lead to premature convergence of evolution as the MRs decay and vanish (Rudolph 2001; Glickman and Sycara 2000; Clune et al. 2008; Meyer-Nieberg and Beyer 2007). In the instances where self-adapting MRs succeed, the authors attribute the cause to be from a relatively smooth fitness landscape (Clune et al. 2008; Glickman and Sycara 2000), or high selection pressure (Maschek 2010). The cause of general premature convergence in rugged landscapes is attributed to the fact that most mutations are deleterious, causing self-adaptation to prefer solutions that mutate less and preserve the fitness of each individual (Clune et al. 2008; Glickman and Sycara 2000). Clune et al. (2008) mention that, in this way, evolution is short-sighted: it cannot adapt MRs to be optimal for the long-term, only optimizing for short-term performance.

Outlier-Based MRs: Some works have proposed looking at the best mutation produced by a certain mutation operator to judge the quality of the operator (Fialho et al. 2008; Whitacre, Pham, and Sarker 2009), with the motivation that an operator that produces infrequent large fitness gains is preferred to one that produces frequent small fitness gains. However, these works model the operator selection as a multi-armed bandit problem. This technique is not only unnatural to evolution, it is also limited by the expressiveness of the arms used and assumes independent arms, thus failing to capture the continuous spectrum that the MR exists in.

3 Method

This section first provides the formal problem definition, a discussion of the general class of genetic algorithms, and then briefly describes a previous adaptive mutation rate (MR) method and its associated vanishing MR problem. Finally this section proposes the Group Elite Selection of Mutation Rates (GESMR) algorithm that addresses this problem with better performance and almost no extra overhead.

3.1 Problem Formulation

Consider the general optimization problem where the goal is to find the best decision variable $x^* \in \mathbb{R}^d$ that minimizes a target function f (e.g. the negative fitness function in the genetic algorithm literature). The objective is therefore

$$\underset{x \in \mathbb{R}^d}{\arg\min} f(x). \tag{1}$$

3.2 Genetic Algorithms and the Mutation Rate

A genetic algorithm (GA) evolves a population of $N+1$ candidate solutions/individuals $x_0, \ldots, x_N$ over time that progressively minimize the objective in Eq. 1. At each evolution time step t, the current population is $\{x_i^{(t)}\}_{i=0}^N$.

To produce the next generation, a GA consists of 1) selection of individuals, 2) mutation of individuals, and 3) crossover of individuals.

The common truncation selection method with one elite is used in this paper. Truncation selection creates a new set of $N+1$ solutions by keeping the single best "elite" solution from the population (known as *elitism*) and uniformly sampling the rest of the N solutions from the top η_x portion of the population with replacement (better solution has lower $f(x)$ value) (Such et al. 2017).

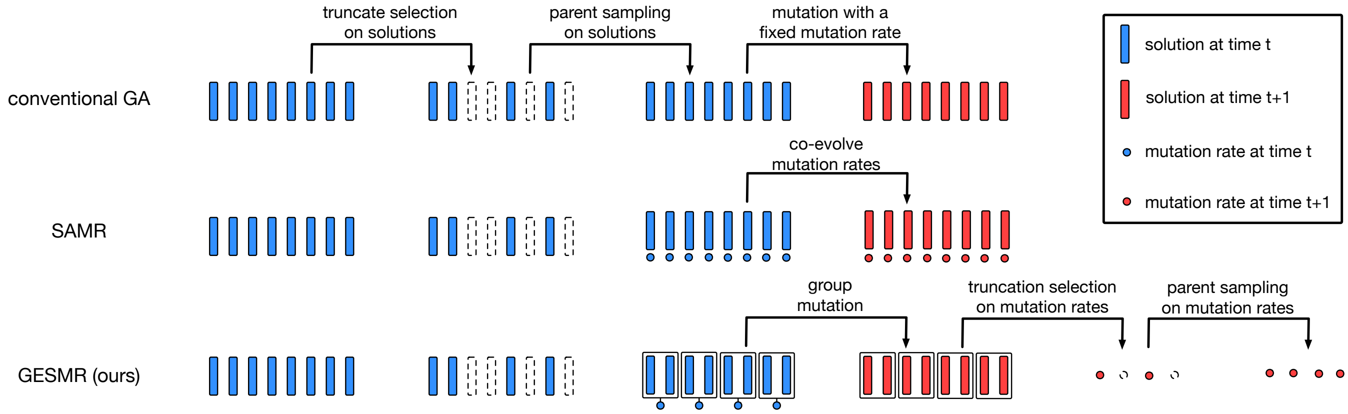

Figure 1: Comparison of GESMR against a fixed MR GA and SAMR. Fixed MR GA only evolves the solution with a given MR. SAMR evolves pairs of solutions and MRs. GESMR co-evolves a population of solutions and a population of MRs separately. Each MR is assigned a group and the MRs are evolved using the best function value gain in the MR's corresponding group.

Since it is a common way to mutate a continuous genotype x (Such et al. 2017), the Gaussian mutation operator $M : \mathbb{R}^d \to \mathbb{R}^d$ is used, which produces x' with

$$x' \sim M(x;\sigma) \triangleq x + \sigma\epsilon, \text{ and } \epsilon \sim \mathcal{N}(0, I). \tag{2}$$

where $\mathcal{N}(0, I)$ denotes a standard multi-variate normal distribution in $\mathbb{R}^d$. $\sigma \in \mathbb{R}_{\geq 0}$ represents the **mutation rate** (MR), which constrains how different x' could be from x.

Crossover is used to mix information between solutions, essentially allowing traits to be transferred to another solution. For the sake of simplicity and to isolate the mutation operator, which is the main focus of this work, no crossover operator is used since crossover is not a necessary mechanism in GAs (Such et al. 2017).

For conventional GA algorithms, a fixed MR is chosen a priori based on the user's preference or prior knowledge. Clearly, a too small σ will slow down evolution and a too large σ will tend towards random search, a tuned σ is needed. It has also been shown that the optimal σ changes over the course of evolution, e.g. a small σ is often needed to "fine tune" the solutions at the end of evolution (Cervantes and Stephens 2009). As a result, the adaptive MR field studies how to dynamically adapt this σ for faster learning and better convergence. Among previous adaptive MR methods, a well-known and commonly used method is the self-adaptation of MR (SAMR) (Kramer 2010; Aleti and Moser 2016; Bäck 1992; Gomez 2004; Thierens 2002). This method attaches to each solution x_i its own MR, σ_i. These pairs $\{(x_i, \sigma_i)\}$ are then evolved, by selection on the pairs and mutating the x_i using σ_i and mutating σ_i using an external fixed meta MR τ.

In practice, a well-known drawback of SAMR is that the MRs produced could prematurely converge to zero over time (Rudolph 2001; Clune et al. 2008; Glickman and Sycara 2000), which is referred to here as the **vanishing mutation rate problem** (VMRP). One might try to simply clip the MR to a lower bound, but a single lower bound that maintains exploration early on while still allowing for fine tuning later may not exist (Cervantes and Stephens 2009). Therefore, there exists a need for a better adaptive MR strategy.

Algorithm 1: One step of GESMR

Input: current solutions $\{x_i^{(t)}\}_{i=0}^N$, current mutation rates $\{\sigma_k^{(t)}\}_{k=1}^K$, the selection rates η_x, η_σ, and the meta mutation rate, τ.

Output: next generation of solutions $\{x_i^{(t+1)}\}_{i=0}^N$ and mutation rates $\{\sigma_k^{(t+1)}\}_{k=1}^K$.

```
// 1. Evolve the solutions
{x̂_i^(t)}_{i=0}^N ← sort {x_i^(t)}_{i=0}^N with ascending f(x̂_i^(t))
Generate {x̃_i^(t)}_{i=0}^N according to Eq. 3 {Selection}
Generate {x_i^(t+1)}_{i=0}^N according to Eq. 4{Mutation}
// 2. Evolve the mutation rates
Calculate Δ_k^(t) according to Eq. 5 {MR worth}
{σ̂_k^(t)}_{k=1}^K ← sort {σ_k^(t)}_{k=1}^K with ascending Δ_k^(t)
Generate {σ̃_k^(t)}_{k=1}^K according to Eq. 6 {Selection}
Generate {σ_k^(t+1)}_{k=1}^K according to Eq. 7{Mutation}
return {x_i^(t+1)}_{i=1}^N and {σ_j^(t+1)}_{j=1}^K
```

3.3 Group Elite Selection of Mutation Rates

This section presents Group Elite Selection of Mutation Rates (GESMR), to adapt MRs on the fly, along with empirical evidence that GESMR mitigates the VMRP and outperforms previous adaptive MR methods. For visualization of GESMR, refer to Fig. 1.

GESMR keeps a set of K positive scalar MRs $\{\sigma_k\}_{k=1}^K$, where $N \equiv 0 \pmod{K}$, and co-evolves them with the $N+1$ candidate solutions, so that the σs do not decay to zero.

At each optimization step t, the current population, $\{x_i^{(t)}\}_{i=0}^N$ is first sorted in ascending order of $f(x_i^{(t)})$, giving $\{\hat{x}_i^{(t)}\}_{i=0}^N$. Truncation selection with one elite is applied to get the next generation parents, $\{\tilde{x}_i^{(t)}\}_{i=0}^N$, with

$$\tilde{x}_i^{(t)} = \begin{cases} \hat{x}_0^{(t)} & i = 0 \\ \sim \mathcal{U}\{\hat{x}_0^{(t)}, \ldots, \hat{x}_{m-1}^{(t)}\} & i = 1, \ldots, N \end{cases} \tag{3}$$

and $m = \eta_x N$ (number of solutions for parent selection).

Then, the *non-elite* solutions, $\{\tilde{x}_1^{(t)}\}_{i=1}^{N}$ are split into K groups of equal size (i.e. each group has N/K solutions) and each group is assigned a different σ_k. Without loss of generality, σ_k corresponds to $\{\tilde{x}_{(k-1)N/K+1}^{(t)}, \dots, \{\tilde{x}_{kN/K}^{(t)}\}$. To form the next generation, each $\tilde{x}_i^{(t)}$ is then mutated according to its corresponding σ_k, while the elite is unaltered:

$$x_i^{(t+1)} = \begin{cases} \tilde{x}_0^{(t)} & i = 0 \\ \sim M(\tilde{x}_i^{(t)}; \sigma_{\lfloor iK/N \rfloor}) & i = 1, \dots, N \end{cases} \tag{4}$$

After the next generation of $\{x_i^{(t+1)}\}_{i=0}^{N}$ are found, GESMR evolves the MRs, $\{\sigma_k\}_{k=1}^{K}$ using another separate but similar GA with one elite, truncation selection, and a different mutation operator.

For each σ_k, its negative fitness is calculated by considering the *best* change in function value it has produced:

$$\Delta_k^{(t)} \triangleq \Delta(\sigma_k^{(t)}) = \min_{i=(k-1)N/K+1}^{kN/K} \big(f(x_i^{(t+1)}) - f(\tilde{x}_i^{(t)})\big). \tag{5}$$

First the MR population is sorted by this $\Delta_k^{(t)}$, producing $\{\hat{\sigma}_{k=1}^{K}\}$. Truncation selection with one elite is applied to get the next generation parent MRs $\{\sigma_k\}_{k=1}^{K}$ with

$$\tilde{\sigma}_k^{(t)} = \begin{cases} \hat{\sigma}_1^{(t)} & k = 1 \\ \sim \mathcal{U}\{\hat{\sigma}_1^{(t)}, \dots, \hat{\sigma}_l^{(t)}\} & k = 2, \dots, K \end{cases} \tag{6}$$

and $l = \eta_\sigma K$ (number of MRs for parent selection). The mutation operator associated with the σs is

$$\sigma' \sim M_\sigma(\sigma; \tau) \triangleq \sigma\tau^{\epsilon} \text{ and } \epsilon \sim \mathcal{U}(-1, 1)$$

where $\mathcal{U}(-1, 1)$ represents a continuous uniform distribution on $\mathbb{R}$ and τ represents a fixed meta mutation rate.

The next generation of MRs is produced by mutating the parent MRs, while the elite parent is unaltered:

$$\sigma_i^{(t+1)} = \begin{cases} \tilde{\sigma}_1^{(t)} & i = 1 \\ \sim M_\sigma(\tilde{\sigma}_i^{(t)}; \tau) & i = 2, \dots, K \end{cases} \tag{7}$$

One full step of GESMR is described in Alg. 1.

The performance of GESMR depends on the number of groups, K. When $K = 1$, GESMR recovers the fixed-MR method. When $K = N$, each solution aside from the elite is assigned a different MR, a method reminiscent of the SAMR method. The experiment section shows that in practice the optimal K lies between 1 and N, and uncovers a heuristic on how to choose such a K.

4 Experiment

The experiments in this section are designed to answer the following questions:

1. How does GESMR compare to other methods in terms of the quality of function values found and how quickly it converges to those values?
2. Does SAMR suffer from the Vanishing Mutation Rate Problem (VMRP)? Does GESMR solve this problem, and can it produce MRs that are optimal in a long-term sense?
3. What parts of GESMR are vital to its success?
4. Why is GESMR more successful than SAMR?
5. What is the optimal group size in GESMR and how much does this parameter matter?
6. Does GESMR generalize to the high-dimensional loss landscapes of neuroevolution?
7. Does GESMR generalize to neuroevolution for reinforcement learning control tasks?

4.1 Comparison Algorithms

For comparison, the following MR selection and adaptation algorithms are evaluated in various optimization problems:

- †**OFMR**: Optimal fixed MR found with a grid search;
- †**LAMR-G**: MR determined at every G generations by "looking ahead," that is, by running a grid search multiple times and picking the MR that produces the best elite in another evolution run (initialized with the current population and run for G generations);
- **FMR**: A fixed MR of $\sigma = 0.01$;
- **1CMR** A fixed MR of $\sigma = 1/d$ (Ochoa 2002);
- **15MR**: MR is doubled if the percentage of beneficial mutations is above 1/5 in the current generation and cut in half if not (Rechenberg 1978);
- **UCB/R**: The adaptive MR method proposed by Fialho et al. (2008), implemented with a multi-armed bandit with R arms (each corresponding to a different MR), and sampling an arm every generation using the upper confidence bound algorithm (Fialho et al. 2008);
- **SAMR**: Self-adaptation of MR, where each solution is assigned its own MR and evolved end-to-end;
- **GESMR**: The method of Algorithm 1;
- **GESMR-AVG**: The method of Algorithm 1 with the min in Eq. 5 replaced with the mean;
- **GESMR-FIX**: The method of Algorithm 1 with the MRs fixed to the initial population and not evolved further.

Details for the parameters of these algorithms are provided in the Appendix. The †represents that the algorithm is an oracle using foresight (looking ahead of the current evolution step) to determine the MR and should not be compared against directly. Note that **LAMR-G** specifically uses foresight to determine the best MR for the next G generations. With sufficiently large G, its MRs thus serve as an empirical estimate of the optimal long-term MRs at any point during evolution.

4.2 Test Optimization Functions

All algorithms are evaluated on common test functions: Ackley, Griewank, Rastrigin, Rosenbrock, Sphere, and Linear (Surjanovic and Bingham 2013). Definitions of these test functions are provided in the Appendix. Each function is evaluated for dimension $d \in \{2, 10, 100, 1000\}$, with the

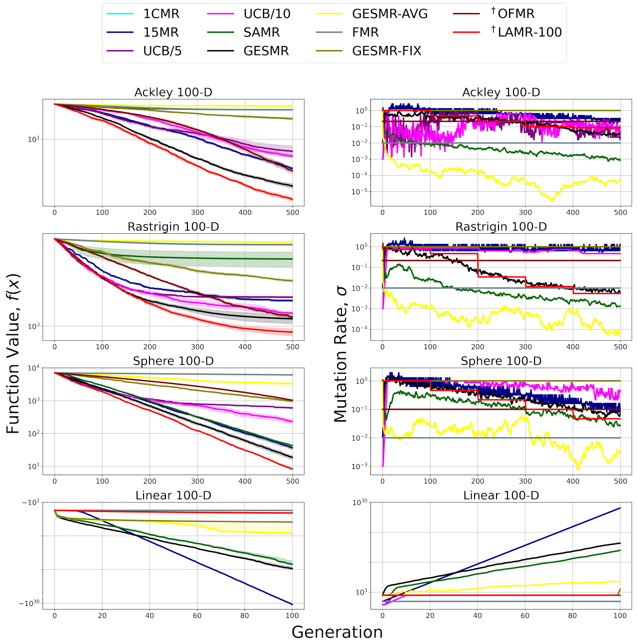

Figure 2: Elite function value and average mutation rate (MR) over generations of evolution by different adaptive MR methods, applied to four test optimization problems. Notice GESMR outperforms other methods in function value and is able to match its MR to the one from LAMR-100.

initial population sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\mathcal{N}(\mathbf{0}, 10^2\mathbf{I})$ (referenced in table as std with values 1 and 10). These functions were chosen because they are common in the GA literature and they span a diverse range of ruggedness for function landscapes (Malan and Engelbrecht 2009). All results are averaged over five seeds.

Fig. 2 shows selected runs from this experiment, displaying the elite function value and the average MR over generations. The full list of final elite function values are reported in Table 1 in the Appendix, serving as a statistic on how good the final solution is. The full list of average elite function values over all evolution iterations are reported in Table 2 in the Appendix, serving as a statistic on how quickly the algorithm converges to a good solution. Mean squared error between the log MR of an algorithm and the log MR of LAMR-100 (averaged over generations) are reported in Table 3 in the Appendix, serving as a statistic on how close to optimal the MRs are.

To answer Question 1, GESMR outperforms other methods, excluding the oracles, in almost all domains both in terms of the final function value and in terms of quickness of convergence to good values.

To answer Question 2, SAMR only succeeds and matches the performance of LAMR when the function landscape is relatively non-rugged, like in the Rosenbrock and Sphere functions. In the rugged functions, SAMR consistently produces MRs that are sub-optimal and smaller than those produced by even OFMR, and thus also lags behind in elite function value during evolution. Thus, SAMR struggles with the VMRP, as shown in previous work (Rudolph 2001; Clune et al. 2008; Meyer-Nieberg and Beyer 2007). However, GESMR overcomes this phenomenon and surprisingly consistently *matches its average MR to the long-term optimal MR* produced by LAMR-100 (i.e. red and black lines match in Fig. 2, and GESMR has consistently the lowest error in Table 3 in the Appendix).

The limitations of of all methods except 15MR, SAMR, and GESMR can be seen in the linear test function. The optimal MR for this case is $\sigma \to \infty$, but other methods are unable to approximate this result because they limit themselves to an upper bound (ex. UCB-R is limited by the largest MR in its arms). On the other hand, GESMR quickly keeps scaling up the MR until reaching a very large MR. GESMR is also arbitrarily precise, fine tuning MRs with an evolutionary process. In contrast, UCB-R and the grid search methods constrain the MRs to a quantized range.

To answer Question 3, GESMR-AVG and GESMR-FIX were run as an ablation of GESMR, with the results shown in Fig. 2 and Tables 1, 2, 3 in the Appendix. GESMR outperforms both of them, suggesting that the use of the best mutation statistic and the evolution of MRs are both vital to its success.

4.3 Empirical Analysis of GESMR vs. SAMR

To answer the Question 4, two objectives for σ are defined based on a change of function value, and these objectives are shown to be related to the methods of GESMR-AVG, GESMR, and SAMR. These objectives are then statistically analyzed over different function landscapes to explain the behavior of the algorithms.

Consider the change in function value for a mutation given a solution and MR:

$$\Delta(x, \sigma) \sim f(M(x; \sigma)) - f(x). \tag{8}$$

For simplicity, this variable will be denoted as Δ. Let $\{\Delta_q\}_{q=1}^{N/K}$ represent independently and identically distributed instances of Δ where q indexes an individual within its group. To minimize $f(x)$ in evolution, a σ must be chosen to minimize $\Delta(x, \sigma)$ in some capacity (denoted as an "MR objective"). Consider the MR objectives

- $\sigma^*_\mu = \arg\min_\sigma \mathbb{E}_{x,\epsilon}[\Delta(x, \sigma)]$ i.e. the mean objective; and
- $\sigma^*_{\text{min}} = \arg\min_\sigma \mathbb{E}_{x,\epsilon}[\min_q \Delta_q(x, \sigma)]$, i.e. the outlier objective.

The expectations in the objectives are over x sampled from the current population and the noise in the mutation operator, ϵ. For simplicity, these objectives are denoted as $\arg\min_\sigma \mathbb{E}[\Delta]$ and $\arg\min_\sigma \mathbb{E}[\min_q \Delta_q]$, respectively. The mean objective corresponds to the algorithm GESMR-AVG, which directly selects σs to minimize a sample average of Δ. The outlier objective corresponds to the algorithm GESMR, which directly selects σs to minimize the *best* (lowest-value) sample of $\{\Delta_q\}$. SAMR does not select σs directly, but rather selects (x_i, σ_i) pairs to minimize $f(x_i)$. However, because x_i is produced using the parent of σ_i, SAMR also indirectly selects pairs (x_i, σ_i) based on σ_is that produce non-

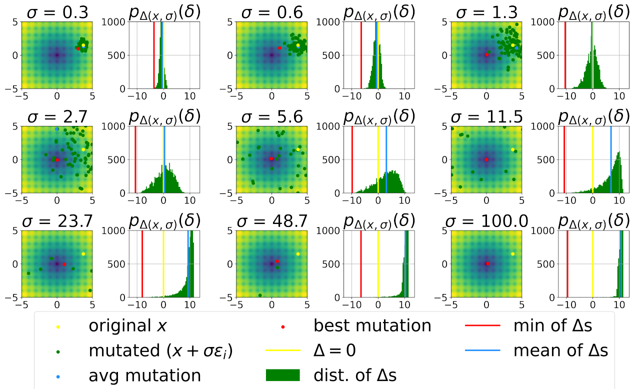

Figure 3: Visualization of mutations and the distribution of the change in function value from the mutations, $\Delta(x, \sigma)$ (defined in Eq. 8), for nine labeled mutation rates, σ, at one point, x, on the 2-D Ackley function. The left plots show an image representation of the 2-D function landscape where lighter colors are higher values and annotates the original solution and some mutated solutions. The right plots show the empirical histogram of $\Delta(x, \sigma)$ and annotates the mean and minimum samples of this histogram. Only moderate σs are able to mutate to the global minimum.

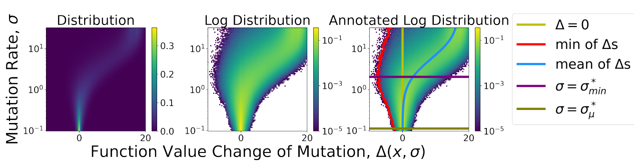

Figure 4: A representation of σ versus $\Delta(x, \sigma)$ (defined in Eq. 8) colored by the empirical probability density function, $p_\Delta(\delta; \sigma)$ and the respective log distribution for the 2-D Ackley function. Many samples of $\Delta(x, \sigma)$ are generated from $x \sim \mathcal{N}(0, I)$, and a logarithmic range of σs, and put into bins of a σ-Δ grid, colored by the number of samples the bin has. Annotated are the σ versus $\mathbb{E}[\Delta; \sigma]$ (mean of Δs) and $\mathbb{E}[\min_q \Delta_q; \sigma]$ (min of Δs) curves, and the σs that minimize them. Importantly, notice that $\sigma^*_\mu = 0$ and $\sigma^*_{\min} > 0$.

deleterious mutations over generations consistently. This mechanism is intuitively associated with the mean objective.

To analyze general function landscapes outside of evolution, x is is either fixed to a point or sampled from a distribution, and many more samples for $\{\Delta_q\}$ are used. Fig. 3 shows a histogram of samples from Δ and visualizes their respective mutations across values of σ for a single x in the Ackley 2-D function, highlighting that the best mutation comes from a σ that is not too small and not too large. Fig. 4 represents this same information, but sampling $x \sim \mathcal{N}(0, I)$, for a continuous range of σ as a visualization of the probability density function (PDF), $p_\Delta(\delta; \sigma)$. The sigma versus the mean objective and the outlier objective curves as well as their optimal σ solutions, σ^*_μ and $\sigma^*_{\min}$ are shown over the PDF. Fig. 5 displays this same plot for several other test optimization problems.

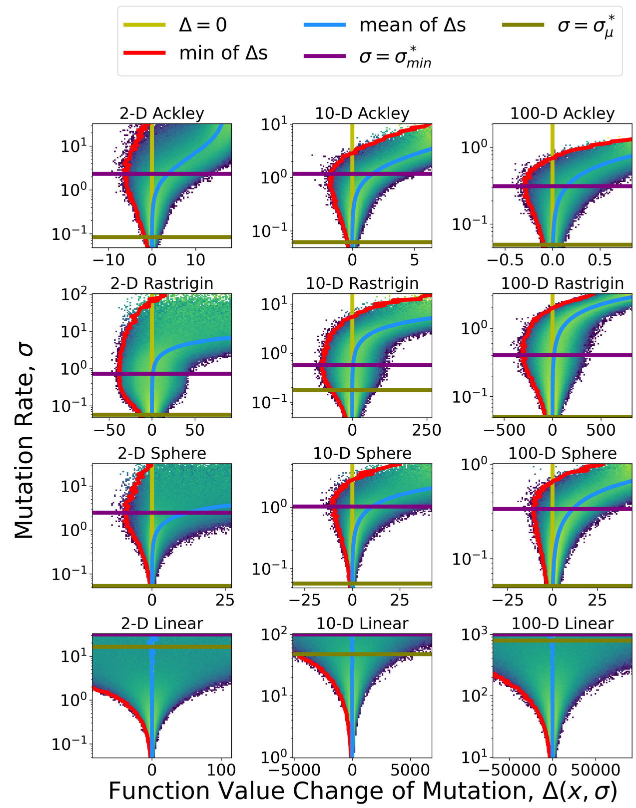

Figure 5: A representation of the σ versus $\Delta(x, \sigma)$(defined in Eq. 8) colored by the empirical probability density function $p_\Delta(\delta; \sigma)$, and the respective log distribution for several different test optimization functions of different dimensionality. Many samples of $\Delta(x, \sigma)$ are generated from $x \sim \mathcal{N}(0, I)$ and a logarithmic range of σs and put into bins of a σ versus Δ 2-D grid, colored by the number of samples the bin has. Annotated are the σ versus $\mathbb{E}[\Delta; \sigma]$ (mean of Δs), $\mathbb{E}[\min_q \Delta_q; \sigma]$ (min of Δs), and $\mathbb{E}[\max_q \Delta_q; \sigma]$ (max of Δs) curves, and the optimal σ that minimizes the first two curves. All curves show that $\sigma^*_\mu \to 0$ and $\sigma^*_{\min} > 0$.

As Fig. 5 shows $\mathbb{E}[\Delta]$ often increases monotonically with σ. As a result, the optimal MR tends to go to zero, i.e. $\sigma^*_\mu \to 0$. Interestingly, $\mathbb{E}[\min_q \Delta_q]$ is zero for $\sigma = 0$, and *decreases* monotonically as σ increases until $\sigma = \sigma^{\min}_*$, and then increases monotonically with σ, leading to $\sigma^*_{\min} \not\to 0$. These behaviors hold true for all landscapes tested, except for the non-rugged linear landscape. Intuitively, this finding makes sense. As $\sigma \to \infty$, $\mathbb{E}[\Delta] = \mathbb{E}_{x'}[f(x')] - \mathbb{E}_x[f(x)]$ (first expectation over all x' possible by mutation) becomes a a constant (i.e. there is no search) and $\mathbb{E}[\min_q \Delta_q]$ becomes random search over the entire function landscape. Both values can be assumed to be worse than any partially optimized solutions during evolution, so both MR objectives will tend towards positive values. As $\sigma \to 0$ (i.e. no mutation), both MR objectives vanish. If $\sigma < \sigma_c$ for some σ_c such that the function landscape can be approximated as linear, it can be shown that $\Delta(x, \sigma) \sim \mathcal{N}(0, \sigma^2 \|\nabla f(x)\|^2)$, where $\nabla f(x)$ is

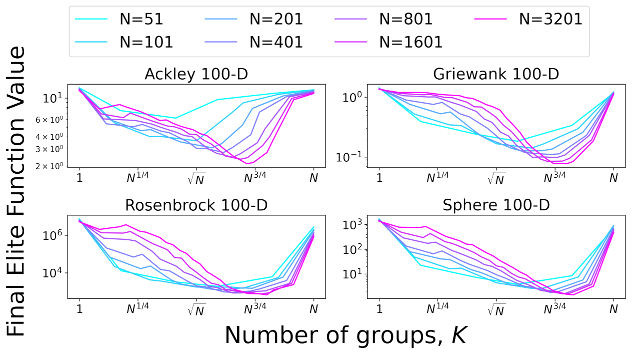

Figure 6: Elite final function value of GESMR versus the number of groups, K, as the population size, N increases in the Ackley 100-D function. As $N \to \infty$, the optimal $K \to N^{3/4}$, suggesting K does not need tuning.

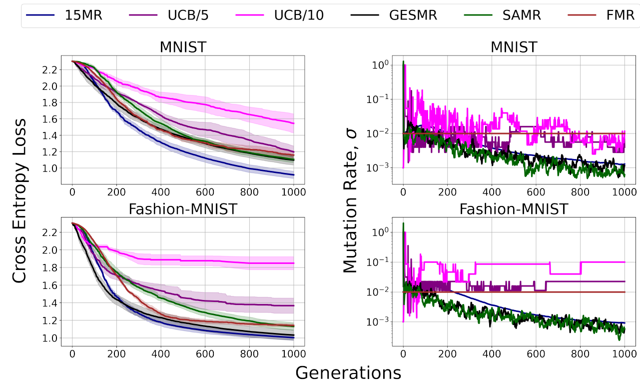

Figure 7: Elite function value and average mutation rate (for different mutation rate control strategies) versus generations of neuroevolution applied to image classification in MNIST and Fashion-MNIST. GESMR outperforms most methods except 15MR.

the gradient of f, leading to $\mathbb{E}[\Delta] = 0$ and $\mathbb{E}[\min_q \Delta_q] < 0$. Therefore, the outlier objective will have $\sigma^*_{\text{min}} > 0$.

These results explain empirically *why* GESMR-AVG and SAMR often suffer from the VMRP in rugged landscapes, and how GESMR may be able to overcome this limitation. Underneath the empirical analysis lies the fundamental flaw in GESMR-AVG and SAMR: the assumption that a σ should consistently produce non-deleterious mutations. Because most mutations are deleterious (Clune et al. 2008), this condition is possible only if $\sigma \to 0$. On the other hand, GESMR incorporates this assumption into the algorithm itself, by considering only the *best* mutation from a σ, and ignoring the other mostly deleterious mutations.

4.4 Ablation on the Group Size Parameter

To answer Question 5, and to evaluate the optimal number of groups, K, evolution was run on the Ackley, Griewank, Rosenbrock, and Sphere functions with $d = 100$ and K equal to all factors of N for various values of N. It turns out that if the number of groups is too small, i.e. $K \to 1$, or too big, i.e. $K \to N$, the performance drops very fast (Fig. 6). In general, $K = \sqrt{N}$ is a reasonable value, but as $N \to \infty$, the optimal $K \to N^{3/4}$. This finding suggests that the number-of-groups hyperparameter can be set according to N and does not need tuning.

4.5 Neuroevolution for Image Classification

To answer Question 6, the algorithms were run on the high dimensional loss landscapes of neuroevolution for image classification with the common MNIST and Fashion-MNIST datasets (LeCun 1998; Xiao, Rasul, and Vollgraf 2017). The details of the datasets, the NN architecture evolved, and the experimental setup are provided in the Appendix. Each algorithm was run independently five times and the mean loss and the standard error measured.

GESMR outperforms all other methods, including FMR and SAMR, but does not beat 15MR (Fig. 7). Presumably, 15MR's hyperparameter of 1/5 is especially suited to the MNIST loss landscapes but might have trouble generalizing to other problems, like the test optimization problems and the reinforcement learning control problems.

4.6 Neuroevolution for Reinforcement Learning

Reinforcement learning (RL) tasks are amenable to the neuroevolution approach because the approach tolerates long time-horizon rewards well (Salimans et al. 2017; Such et al. 2017). To answer Question 7, the algorithms were evaluated on four common RL control tasks: CartPole, Pendulum, Acrobot, and MountainCar (Brockman et al. 2016). In all these tasks, a controller maps the robot's input observations to either continuous or discrete actions to maximize a cumulative reward. The details of these environments, the neural architecture evolved, and the experimental setup are provided in the Appendix. Each algorithm was run independently five times and the mean and standard error of performance was measured.

The results are shown in Fig. 8 in the Appendix. GESMR generally outperformed other methods including the baseline fixed MR and SAMR. Presumably, GESMR fails in MountainCar because the reward signal is very sparse (zero rewards provide no way to appropriately select for MRs).

5 Conclusion

In this paper, a novel and simple adaptive mutation rate (MR) method, group elite selection mutation rate (GESMR), was proposed to mitigate the vanishing mutation rate problem (VMRP), along with empirical analysis that grounds its success over self-adaptation of mutation rates (SAMR). Comprehensive experiment results showed that GESMR outperforms previous adaptive MR methods in final value and convergence speed. GESMR also consistently matches its MRs to the empirically estimated long-term optimal MR. Thus, this work provides the next step in designing self-adaptive machine learning algorithms.

References

Aleti, A.; and Moser, I. 2016. A Systematic Literature Review of Adaptive Parameter Control Methods for Evolutionary Algorithms. *ACM Comput. Surv.*, 49(3): 1–35.

Bäck, T. 1992. Self-Adaptation in Genetic Algorithms. In *Proceedings of the First European Conference on Artificial Life*. Citeseer.

Bäck, T.; and Schütz, M. 1996. Intelligent mutation rate control in canonical genetic algorithms. In *Foundations of Intelligent Systems*, 158–167. Springer Berlin Heidelberg.

Böttcher, S.; Doerr, B.; and Neumann, F. 2010. Optimal Fixed and Adaptive Mutation Rates for the LeadingOnes Problem. In *Parallel Problem Solving from Nature, PPSN XI*, 1–10. Springer Berlin Heidelberg.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym.

Cervantes, J.; and Stephens, C. R. 2009. Limitations of Existing Mutation Rate Heuristics and How a Rank GA Overcomes Them. *IEEE Trans. Evol. Comput.*, 13(2): 369–397.

Clune, J.; Misevic, D.; Ofria, C.; Lenski, R. E.; Elena, S. F.; and Sanjuán, R. 2008. Natural selection fails to optimize mutation rates for long-term adaptation on rugged fitness landscapes. *PLoS Comput. Biol.*, 4(9): e1000187.

Dang, D.-C.; and Lehre, P. K. 2016. Self-adaptation of Mutation Rates in Non-elitist Populations. In *Parallel Problem Solving from Nature – PPSN XIV*, 803–813. Springer International Publishing.

Doerr, B.; Doerr, C.; and Lengler, J. 2019. Self-adjusting mutation rates with provably optimal success rules. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '19, 1479–1487. New York, NY, USA: Association for Computing Machinery.

Eiben, A. E.; Hinterding, R.; and Michalewicz, Z. 1999. Parameter control in evolutionary algorithms. *IEEE Trans. Evol. Comput.*, 3(2): 124–141.

Fialho, Á.; Da Costa, L.; Schoenauer, M.; and Sebag, M. 2008. Extreme Value Based Adaptive Operator Selection. In *Parallel Problem Solving from Nature - PPSN X, 10th International Conference Dortmund, Germany, September 13-17, 2008, Proceedings*, volume 5199, 175–184. unknown.

Glickman, M. R.; and Sycara, K. 2000. Reasons for premature convergence of self-adapting mutation rates. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, volume 1, 62–69 vol.1. ieeexplore.ieee.org.

Goldman, B. W.; and Tauritz, D. R. 2012. Supportive coevolution. In *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*, GECCO '12, 59–66. New York, NY, USA: Association for Computing Machinery.

Gomez, J. 2004. Self Adaptation of Operator Rates in Evolutionary Algorithms. In *Genetic and Evolutionary Computation – GECCO 2004*, 1162–1173. Springer Berlin Heidelberg.

Greenwell, R. N.; Angus, J. E.; and Finck, M. 1995. Optimal mutation probability for genetic algorithms. *Math. Comput. Model.*, 21(8): 1–11.

Hassanat, A.; Almohammadi, K.; Alkafaween, E.; Abunawas, E.; Hammouri, A.; and Prasath, V. B. S. 2019. Choosing Mutation and Crossover Ratios for Genetic Algorithms—A Review with a New Dynamic Approach. *Information*, 10(12): 390.

Karafotias, G.; Hoogendoorn, M.; and Eiben, A. E. 2015. Parameter Control in Evolutionary Algorithms: Trends and Challenges. *IEEE Trans. Evol. Comput.*, 19(2): 167–187.

Kramer, O. 2010. Evolutionary self-adaptation: a survey of operators and strategy parameters. *Evol. Intell.*, 3(2): 51–65.

LeCun, Y. 1998. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

Liang, J. Z.; and Miikkulainen, R. 2015. Evolutionary Bilevel Optimization for Complex Control Tasks. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO '15, 871–878. New York, NY, USA: Association for Computing Machinery.

Malan, K. M.; and Engelbrecht, A. P. 2009. Quantifying ruggedness of continuous landscapes using entropy. In *2009 IEEE Congress on Evolutionary Computation*, 1440–1447.

Maschek, M. K. 2010. Intelligent mutation rate control in an economic application of genetic algorithms. *Comput. Econ.*, 35(1): 25–49.

Metzgar, D.; and Wills, C. 2000. Evidence for the adaptive evolution of mutation rates. *Cell*, 101(6): 581–584.

Meyer-Nieberg, S.; and Beyer, H.-G. 2007. Self-Adaptation in Evolutionary Algorithms. In Lobo, F. G.; Lima, C. F.; and Michalewicz, Z., eds., *Parameter Setting in Evolutionary Algorithms*, 47–75. Berlin, Heidelberg: Springer Berlin Heidelberg.

Ochoa, G. 2002. Setting the mutation rate: Scope and limitations of the 1/L heuristic. In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, 495–502. maths.stir.ac.uk.

Patnaik, L. M.; and Mandavilli, S. 1986. Adaptation in Genetic Algorithms. In *Genetic Algorithms for Pattern Recognition*, 45–64. CRC Press.

Rechenberg, I. 1978. Evolutionsstrategien. In *Simulationsmethoden in der Medizin und Biologie*, 83–114. Springer Berlin Heidelberg.

Rudolph, G. 2001. Self-adaptive mutations may lead to premature convergence. *IEEE Trans. Evol. Comput.*, 5(4): 410–414.

Salimans, T.; Ho, J.; Chen, X.; Sidor, S.; and Sutskever, I. 2017. Evolution Strategies as a Scalable Alternative to Reinforcement Learning.

Sewell, M.; Samarabandu, J.; Rodrigo, R.; and McIsaac, K. 2006. The rank-scaled mutation rate for genetic algorithms. *Int. J. Inform. Technol.*, 3(1): 32–36.

Sinha, A.; Malo, P.; and Deb, K. 2018. A Review on Bilevel Optimization: From Classical to Evolutionary Approaches and Applications. *IEEE Trans. Evol. Comput.*, 22(2): 276–295.

Smith, J.; and Fogarty, T. C. 1996. Self adaptation of mutation rates in a steady state genetic algorithm. In *Proceedings of IEEE International Conference on Evolutionary Computation*, 318–323. ieeexplore.ieee.org.

Srinivas, M.; and Patnaik, L. M. 1994. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Trans. Syst. Man Cybern.*, 24(4): 656–667.

Such, F. P.; Madhavan, V.; Conti, E.; Lehman, J.; Stanley, K. O.; and Clune, J. 2017. Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning.

Surjanovic, S.; and Bingham, D. 2013. Optimization Test Functions and Datasets. https://www.sfu.ca/~ssurjano/optimization.html. Accessed: 2021-9-6.

Thierens, D. 2002. Adaptive mutation rate control schemes in genetic algorithms. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, volume 1, 980–985 vol.1. ieeexplore.ieee.org.

Whitacre, J. M.; Pham, T. Q.; and Sarker, R. A. 2009. Use of statistical outlier detection method in adaptive evolutionary algorithms.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms.