Trends in Mathematics
Information Center for Mathematical Sciences
Volume 6, Number 2, December, 2003, Pages 113–123

MODELING GENETIC ALGORITHMS FROM A LINEAR OPERATOR POINT OF VIEW. A SURVEY OF RECENT ADVANCES AND FUTURE PERSPECTIVES

LOTHAR M., SCHMITT

Abstract. We present an overview over recent advances in stochastic modeling of genetic algorithms. In addition, we discuss possible future research activities in this field.

1. Introductory overview

What is a genetic algorithm? Genetic algorithms are a simulation method that borrows its fundamental ideas from principles which allegedly govern evolution in nature. An early reference to the idea of biology-inspired computing and optimization is, *e.g.*, [3]. A genetic algorithm usually comprises three operators: mating (crossover), mutation, and selection. These operators act on finite populations consisting of creatures (candidate solutions) taken from a set $\mathcal{C}$. Crossover models the exchange and recombination of genetic information of creatures and is inspired by exchange and recombination of genetic information in living organisms, *e.g.*, during the process of sexual reproduction. Mutation models random change in the genetic information of creatures and is inspired by random change of genetic information in living organisms, *e.g.*, through the effects of radiation or chemical mismatch. Fitness selection models reproductive success of adapted organisms in their environment and, in computer implementations, usually includes a random rearrangement of the creatures/individuals in a population. After initializing a population, a genetic algorithm iterates crossover, mutation, and selection in accordance with the following table until a termination condition is satisfied:

Step 0	Initialize population $p = (c_1, \ldots, c_s)$ with $c_1, \ldots, c_s \in \mathcal{C}$.
Step 1	Apply crossover to pairs $(c_{2\sigma-1}, c_{2\sigma})$ of creatures in p, $1 \leq \sigma \leq s/2$.
Step 2	Apply mutation to the genetic information in p.
Step 3	Apply the selection mechanism to the family of creatures in p.
Step 4	**If** termination condition is satisfied, **then** stop, **else** continue at step 1.

Typical termination conditions are: 'the best creature seen so far does not change for a long period of time', 'the best creature seen so far meets a certain standard',

'an analysis has determined that after a certain number of cycles the probability to have seen the best creature is sufficiently close to 1' (*cf.* [2], [12, proof of Thm. V.4.3 (p. 160)], [26, proof of Thm. 3.3.2]) or simply 'end of allocated computation time'.

There are several principle attitudes to look at genetic algorithms: some researchers see them primarily as a means to model and study evolutionary principles in nature, others such as this author see them primarily as a computational optimization method which is biology-inspired.

From now on, we shall limit the discussion to the second point of view, *i.e.*, genetic algorithms whose goal is to find globally optimal creatures in $\mathcal{C}$ in regard to a *fitness function* $f : \mathcal{C} \to \mathbf{R}$.

The tensor-string model for populations. As a first step in developing a mathematical description for the behavior of genetic algorithms, many researchers have studied the asymptotics of these algorithms using a Markov chain model as primary tool. A quite extensive listing of related references in this regard can be found in [24, 25, 26]. This approach follows, in principle, the historic route for handling the simulated annealing algorithm [1] where substantial results on asymptotics were obtained first (*e.g.*, [11]) and subsequently finite-length algorithms were mastered (*cf.* [4, 5, 6, 7]).

Before one can set up a Markov chain model, one has to define the underlying state space, *i.e.*, one has to specify a model for populations. There are, essentially, two models for populations in use: the multi-set model and the tensor-string model.

The multi-set model describes a population p with s creatures as a family of pairs $(c, \#(c,p))$ where $c \in \mathcal{C}$ and $\#(c,p) \in \mathbf{N} \cap [1,s]$ denotes the number of copies of c in p. This description of populations was initiated in research in [8, 9, 16, 30] and has been used in many subsequent studies. See also [31].

An alternate model to describes a population is the tensor-string model initiated independently in [18, 21]. Here, the population is simply an ordered tuple (list) of creatures which themselves are words (list of genes) over the underlying alphabet in accordance with usual representation of populations in computer memory. One advantage of this approach is the possibility to introduce spatial structure into the model by considering local-dependent genetic operators. Another advantage is that the genetic operators mutation and crossover on populations can be described by suitable tensor-products of simpler basic components (see, *e.g.*, [24, Prop. 3.6.2], [29, Sec. 2.1, formula (4)] in regard to mutation, and [21, p. 114, line 5], [25, Sec. 2.4–5] in regard to crossover). A third advantage of the tensor-string model is that it can easily be extended to populations containing several types of creatures (species) in a coevolutionary setting that allows optimization via "arms-races" of species. See [27, 28, 29].

The multi-set model in the single-species setting can be recovered from the tensor-string model via projection into the quotient space over the kernel of permutations on these populations. This is outlined in [24, Sec. 2.9].

After a model for populations is fixed, the vector space $\mathcal{V}_\wp$ underlying the model becomes the free vector space over the set of populations[1] $\wp$ containing the simplex $\mathcal{S}_\wp$ of probability distributions over populations. $\mathcal{S}_\wp$ is the relevant state space for investigating genetic algorithms where the stochastic matrices representing the genetic operators act.

Spectral and geometric properties of mixing. The tensor-string model allows to investigate spectral and geometric/contraction properties of the stoch- astic matrix $M_{\mu_o,\mu} \cdot C_\chi$ associated with the crossover-mutation operator in a genetic algorithm with general-size alphabet in a natural fashion. In regular circumstance, the stochastic matrices $M_{\mu_o,\mu}$ and C_χ associated with mutation and crossover are self-adjoint and commute (*cf.* [24, Secs. 3.2–3, 5]). Thus, their combined spectrum can be computed from the individual spectra. By representing mutation and crossover on populations through suitable tensor-products of simpler basic components, the individual spectra of crossover and mutation can be explicitly computed for a variety of settings (see additional comments below). Such findings extend and contribute to understanding results in [14]. By computing spectral estimates and analyzing invariant subspaces for crossover and mutation, one can show how the crossover operator enhances the averaging procedure of the mutation operator in the mixing phase (random generator phase, *i.e.*, steps 1–2 in the table shown above) of the genetic algorithm. See [21, Prop. 10] and [24, Thm. 6.1] for results in this regard.

We note that the Vose-Liepins version of crossover-mutation as advocated in [31, Sec. 5.4: p. 44] is also explicitly integrated in our approach using the tensor-string model (*cf.* [26, Sec. 3.5]). By mapping the tensor-string model to the multi-set model through the projection procedure mentioned above, we can compute corresponding spectral estimates for crossover-mutation in the multi-set model as, *e.g.*, [24, Thm. 6.2] and thus can rigorously extend [14].

Let us further note at this point that under exceptional circumstances a properly chosen crossover operator may yield dramatic improvement of a genetic algorithm. One example of such a situation is analyzed in [13].

The analysis of invariant subspaces for crossover (see, *e.g.*, [21, Prop. 7.5]) also shows that crossover alone is not a suitable mixing operator in regard to optimization in contrast to popular belief and some results in the literature [17] even if one starts the algorithm from a "genetically rich" population. Genetic algorithms without mutation do not yield ergodic procedures (Markov chains). This should be apparent by considering an initial uniform population filled with copies of a single suboptimal solution which can never be changed neither by common crossover operators nor by one of the commonly used selection operators. Genetic algorithms without mutation rather show a possibly misleading convergence effect called *genetic drift*. A model for genetic drift is discussed in detail in [22, Sec. 6] based upon the analysis of contraction properties of the selection operator towards uniform populations.

[1]We shall identify populations $p \in \wp$ and base-vectors in what follows.

The necessity of annealing. A genetic algorithm is called *simple*, if all operators in steps 1–3 of the table shown above stay constant over the course of the algorithm. Let G be the stochastic matrix associated with a simple genetic algorithm. Study of the simple genetic algorithm with regular operators shows that a finite number of iterates yields a associated stochastic matrix G^t, $t \in \mathbf{N}$, which is fully positive. Consequently, the associated invariant probability distribution $v = Gv \in \mathcal{S}_\wp$ (*cf.* [26, Prop. 1.3.2] or [19, Ch. 1]) is fully positive and the algorithm does not asymptotically converge to optima. See [24, Thms. 8.1–2].

The preceding simple observation makes it necessary to vary at least some of the genetic operators over time in order to obtain an algorithm that converges asymptotically towards global maxima. The main result of [8] shows that annealing the mutation rate to 0 alone and consequently having the mutation operator approaching the identity operator **1** does not yields success. Thms. 8.2–3 of [24] show that increasing the selection pressure alone does also fail to achieve the goal of asymptotic global optimization. Varying the crossover operator alone is also of limited interest since it does not completely control mixing nor does it control the selection pressure towards optimal solutions in any way. Consequently, one has to

(Condition M) anneal the mutation rate to 0 in order to avoid asymptotically a positive probability for suboptimal solutions,

(Condition S) increasing the selection pressure in an unbounded fashion (see details below) in order to stir the algorithm towards a limit probability distribution which is positive only over populations containing optimal creatures, and

(Condition C) (possibly) anneal the crossover rate to 0 such that the crossover operator approaches asymptotically the identity operator **1** as well.

In what follows, we shall be dealing with satisfying the above conditions **(M)**, **(S)** and **(C)** in order to obtain a scaled genetic algorithm that asymptoticly converges to global optima. We note that [24, Thm. 8.6] shows that under quite reasonable but not absolutely general circumstances condition **(C)** need not be satisfied. Mathematically, the above conditions means dealing with an inhomogeneous Markov chain in order to describes the probabilistic behavior of the scaled genetic algorithm both step-by-step and asymptotically.

2. Some Notation

The underlying alphabet and definition of creatures. Extending the discussion of binary genetic algorithms (but also including it), the alphabet $\mathcal{A} = \{a(0) \dots a(\alpha-1)\}$ of size α, $2 \leq \alpha \in \mathbf{N}$, is primarily interpreted here as a set of equidistant real numbers. This is inspired by optimization of real-valued functions in compact domains of $\mathbf{R}^\ell$, $\ell \in \mathbf{N}$ (see, *e.g.*, [20, 23]), *i.e.*, creatures are defined as words of length ℓ over $\mathcal{A}$. Thus, $\mathcal{C} = \mathcal{A}^\ell$. For certain definitions such as the definition of the spot mutation matrix $\mathbf{m}(\mu_o)$ given below, we shall assume for reason of mathematical convenience that the alphabet is cyclic, *i.e.*, isomorphic to $\mathbf{Z}_\alpha$ and assume that optima of the fitness function are located away from the

boundaries of the "cube" described by $\mathcal{C} = \mathcal{A}^\ell$, *i.e.*, away from points containing $a(0)$ and $a(\alpha{-}1)$ as coordinates. Let $\mathcal{V}_1$ denote the free vector space over $\mathcal{A}$.

Populations. The set of populations $\wp$ is now simply defined as $\wp = \mathcal{C}^s = \mathcal{A}^L$, $4 \leq s \in 2\mathbf{N}$, $L = \ell \cdot s$. Let $\mathcal{V}_\wp$ denote the free vector space over $\wp$.

Let $\mathcal{U} \subset \mathcal{V}_\wp$ be the free vector space over all populations which are uniform, *i.e.*, which consist of s copies of a single creature. Consequently, $\wp \cap \mathcal{U}$ shall denote the set of uniform populations (identifying populations and base-vectors). In addition, $P_\mathcal{U}$ shall denote the orthogonal projection onto $\mathcal{U}$.

3. Scaling the genetic operators and convergence

The spot mutation matrix $\mathbf{m}(\mu_o)$ and neighborhood-based search. Let $\mu_o \in [0,1]$. In order to let mutation perform a scalable compromise between pure random search (μ_o=1) and neighborhood-based change (μ_o=0) on the alphabet level, one defines the stochastic spot mutation matrix $\mathbf{m}(\mu_o)$ as follows: for $0 \leq \iota \neq \iota' \leq \alpha{-}1$ let

$$<a(\iota'), \mathbf{m}(\mu_o) a(\iota)> \; = (1-\mu_o)/(2n) + \mu_o/(\alpha-1), \text{ if } d(a(\iota'), a(\iota)) \leq n. \tag{1}$$

$$<a(\iota'), \mathbf{m}(\mu_o) a(\iota)> \; = \mu_o/(\alpha-1), \text{ if } d(a(\iota'), a(\iota)) > n. \tag{2}$$

Here d is the cyclic distance on $\mathbf{Z}_\alpha$ and $n \in \mathbf{N}$ is fixed describing the size of a "preferred" small neighborhood of a letter $a(\iota)$. In case μ_o=0, change is only possible with equal probability $1/(2n)$ to one of the neighbors of $a(\iota)$. In case μ_o=1, change is possible with equal probability $1/(\alpha{-}1)$ to any element of $\mathcal{A}$.

Multiple-spot mutation $M_{\mu_o,\mu}$ and weak ergodicity. Multiple-spot mutation applies the probabilistic change determined by the spot mutation matrix $\mathbf{m}(\mu_o)$ sequentially with probability and *mutation-rate* $\mu \in (0, 1/2)$ to every letter $a(\iota_{\hat{\lambda}})$, $1 \leq \hat{\lambda} \leq L$, in the combined word of length L over $\mathcal{A}$ representing the current population. For a more formal definition see [24, Sec. 3.3] or [26, Sec. 2.1.1]. The associated symmetric stochastic matrix $M_{\mu_o,\mu}$ is then fully positive, invertible and satisfies:

$$M_{\mu_o,\mu} = \bigotimes_{\hat{\lambda}=1}^{L} ((1-\mu)\mathbf{1} + \mu\, \mathbf{m}(\mu_o)). \tag{3}$$

In particular, there exist $K_o \in \mathbf{R}_*^+$ such that

$$K_o \mu^L \mu_o^L \leq <q, M_{\mu_o,\mu} p> \text{ for every } p, q \in \wp. \tag{4}$$

In order to make the inhomogeneous Markov chain representing the scaled genetic algorithm *weakly ergodic*[2], one then schedules the mutation-rate μ and the balance μ_o between local and random search on the alphabet level in accordance with the following two possibilities:

$$\kappa_o = 1,\ \mu_o \in (0,1] \text{ is fixed, } \mu(t) = (t+1)^{-1/L}/2. \tag{5}$$

[2]See [12, pp. 142–151, p. 151: Thm. V.3.2] for treatment of this well-known topic in full generality. Alternatively, [26, Prop. 1.3.1, Sec. 3.2] treats weak ergodicity with simpler means particularly tailored towards the situation of the inhomogeneous Markov chain considered in this exposition.

$\kappa_o \in (1,\infty)$ is fixed, $\mu_o(t) = \mu(t)^{\kappa_o - 1}$, $\mu(t) = (t+1)^{-1/(\kappa_o L)}/2$. (*6*)

For a proof of the fact that these annealing schedules yield a weakly ergodic inhomogeneous Markov chain that describes the resulting scaled genetic algorithm use line (*4*) and [24, Thm. 4.2].

Crossover. In order to limit this presentation, let us only discuss the case of single-cutpoint regular crossover C_χ. See [21, Sec. 2.2] [24, Sec. 5] [26, Sec. 2.2] [25, Sec. 2.3–5] for further details and generalizations.

In order to execute single-cutpoint regular crossover C_χ, the current population $p = (c_1, \ldots, c_s)$, $c_1, \ldots, c_s \in \mathcal{C}$, is divided into pairs $(c_{2\sigma-1}, c_{2\sigma})$, $1 \leq \sigma \leq s/2$. To each of the $s/2$ pairs, the single-cutpoint operation $\bar{C}$ is then applied with probability and *crossover-rate* $\chi \in [0,1]$.

The single-cutpoint operation $\bar{C}$ randomly selects a cutpoint $\lambda \in [1, \ell-1]$ and then exchanges the genes/letters of $c_{2\sigma-1}$ and $c_{2\sigma}$ in the first λ components. Thus, $\lambda=1$ corresponds to exchanging only the first letter of $c_{2\sigma-1}$ and $c_{2\sigma}$, while $\lambda=\ell-1$ corresponds to exchange of all but the last letter. We have

$$\bar{C} = (\ell-1)^{-1} \textstyle\sum_{\lambda=1}^{\ell-1} C(\lambda) \tag{7}$$

where $C(\lambda)$ is a unitary stochastic matrix acting on $\mathcal{V}_\wp|_{s=2}$ that corresponds to switching letters in the first λ components within pairs of creatures. With this description of $\bar{C}$, we have

$$C_\chi = \textstyle\bigotimes_{\sigma=1}^{s/2} ((1-\chi)\mathbf{1} + \chi\, \bar{C}) \tag{8}$$

Estimating the spectrum of $\bar{C}$ using line (*7*), and then using line (*8*), one obtains that the second largest eigenvalue of C_χ is given by $1 - 2\chi/(\ell-1)$ for sufficiently small χ. This fact can be combined with the spectral analysis of mutation via line (*3*) to obtain spectral estimates for the second largest eigenvalue of crossover-mutation. Consult [24, Thm. 6.2] in this regard.

Except for the very next section of this presentation, we shall suppose that the crossover rate χ satisfies

$$\chi(t) = \mu(t)^{1/m},\ m \in \mathbf{N} \text{ fixed.} \tag{9}$$

The mutation flow inequality. The mutation flow inequality listed in line (*12*) is a simple but important insight that together with a contraction property of fitness selection towards uniform populations listed below assures asymptotic convergence of the scaled genetic algorithm considered here to a probability distribution v_∞ that is strictly positive only over $\wp \cap \mathcal{U}$. In fact using the notation established thus far, we have (*cf.* [26, Prop. 3.1.1], [25, Prop. 2.2.3, partial proof]):

$$\beta_\mu = \beta(M_{\mu_o,\mu}) = \min\{||P_{\mathcal{U}} M_{\mu_o,\mu} p||_1 : p \in \wp \cap \mathcal{U}\} \in (0,1), \tag{10}$$

$$\lim_{\mu \to 0} \beta_\mu = 1, \text{ and} \tag{11}$$

$$||(\mathbf{1} - P_{\mathcal{U}}) M_{\mu_o,\mu} C_\chi v||_1 \leq 1 - \beta_\mu + \beta_\mu ||(\mathbf{1} - P_{\mathcal{U}}) v||_1, \qquad \forall v \in \mathcal{S}_\wp. \tag{12}$$

The constant β_μ in line (*10*) has been explicitly computed for some cases in [21, Prop. 4.4], [24, Prop. 3.7.4], [25, Prop. 2.2.3], and the statement in line (*12*) can be somewhat strengthened. The mutation flow inequality shows how the mutation operation controls the balance between uniform and non-uniform populations in a genetic algorithm.

Proportional fitness selection. Fitness selection models reproductive success of adapted organisms in their environment and, usually, includes a random rearrangement of the creatures/individuals in a population. In this exposition, we shall restrict the discussion to scaled proportional fitness selection based upon a given fitness function $f : \mathcal{C} \to \mathbf{R}_*^+$ (consult, *e.g.*, [10, p. 16], [21, Sec. 2.3], or [24, Sec. 7.1]) which is used in standard applications of genetic algorithms to select the creatures in the future population from the creatures in the present population after the crossover-mutation operation. The presentation given here generalizes to a population-dependent fitness function such as rank based upon an $f : \mathcal{C} \to \mathbf{R}_*^+$ as above (*cf.* [25]). Consult [24, Sec. 7.3] for a suitable definition of rank.

We note that a typical population-dependent fitness function arises, if one considers coevolutionary optimization among creatures from a single "species". For example, game-playing strategies that represent creatures in a population are evaluated by playing each other. Thus, the convergence results mentioned below solve the *coevolutionary optimization problem for one species*. See [27, 28, 29] for generalizations of these results to a multi-species setting.

Let $\mathcal{C}_{\max} \subset \mathcal{C}$ be the set of creatures where f assumes maximal value. The optimization algorithm is supposed to find elements in $\mathcal{C}_{\max}$. Let

$$\rho_2(f) = \min\{f(c)/f(d) : c \in \mathcal{C}_{\max}, d \in \mathcal{C} \setminus \mathcal{C}_{\max} \neq \emptyset\} > 1. \tag{13}$$

$\rho_2(f)$ measures the "strength" of second-to-best creatures d.

Next, we define *power-law scaling* of the fitness function in accordance with, *e.g.*, [10, p. 124], [21, Sec. 2.3], [24, Sec. 7.1]. Let κ_o as in lines (*5–6*), m as in line (*9*), and $B \in \mathbf{R}_*^+$ such that

$$\kappa_o \ell < \kappa_o L B \log(\rho_2(f)) + 1/m. \tag{14}$$

Now, we set for $c \in \mathcal{C}$ and $t \in \mathbf{N}$:

$$f_t(c) = (f(c))^{g(t)}, \qquad g(t) = B \cdot \log(t+1). \tag{15}$$

In this exposition, we shall only consider *logarithmic scalings* $g(t)$ as listed above which are unbounded. It has been shown in [24, Thm. 8.5] with a remarkably simple "linear operator type" argument, that faster scalings with, *e.g.*, linear growth $g(t) = at + b$ in the exponent are of limited value, in particular, in regard to the use of a crossover operation. In fact, such algorithms are asymptotically equivalent to a "take-the-best" algorithm [24, Def. 8.4] where one cycle of the algorithm consists of the mutation-step and picking maximal creatures in the current population.

In cycle $t \in \mathbf{N}$ of the algorithm, scaled proportional fitness selection S_t selects creatures c in the current population p with a probability that is proportional to $f_t(c)$ and the number of occurrence $\#(c, p)$ of c in p for the subsequent population

q (which is then subject to the next cycle of the algorithm corresponding to $t+1$). Consult, *e.g.*, [24, Sec. 7.1], [26, Sec. 2.3] for more detail. Let S_t also denote the stochastic matrix associated with scaled proportional fitness selection. S_t acts on $\mathcal{V}_\wp$ and describes transition probabilities for entire populations. The following facts describe basic properties of S_t. For $p = (c_1, c_2, \ldots, c_s), q = (d_1, d_2, \ldots, d_s) \in \wp$, $c_\sigma, d_\sigma \in \mathcal{C}$, $1 \leq \sigma \leq s$, one has:

$$<q, S_t p> = (\textstyle\sum_{\sigma=1}^{s} f_t(c_\sigma))^{-s} \cdot \prod_{\sigma=1}^{s} \#(d_\sigma, p)\, f_t(d_\sigma). \tag{16}$$

$$\text{If } p \in \wp \cap \mathcal{U}, \text{ then } S_t p = p. \tag{17}$$

$$||P_{\mathcal{U}} S_t p||_1 \geq 1 - \theta, \quad \text{with } \theta = 1 - s^{-s}. \tag{18}$$

$$\text{If } v \in \mathcal{S}_\wp, \text{ then } ||(\mathbf{1} - P_{\mathcal{U}}) S_t v||_1 \leq \theta \cdot ||(\mathbf{1} - P_{\mathcal{U}}) v||_1. \tag{19}$$

Strong ergodicity and convergence to uniform populations. Let $(G_t)_{t \in \mathbf{N}}$ denote the inhomogeneous Markov chain that describes the probabilistic behavior of the scaled genetic algorithm considered here:

$$G_t = S_t M_{\mu_o(t), \mu(t)} C_{\chi(t)}. \tag{20}$$

Strong ergodicity as discussed in [12, p. 157: Sec. V.4] follows now from the functional form of the stochastic matrices involved in the model here and weak ergodicity discussed above. In fact, one verifies the conditions of [12, p. 160: Thm. V.4.3] or [26, Thm. 3.3.2]. This verification uses the techniques in [24, Thm. 8.6.1, proof, pp. 53–54] to show:

$$\textstyle\sum_{t=1}^{\infty} ||v_{t+1} - v_t||_1 < \infty \text{ where } v_t = G_t v_t \in \mathcal{S}_\wp, \textit{ cf.} \text{ [19, p. 7: Prop. 2.3].} \tag{21}$$

v_t is uniquely determined for $\mu < 1/2$ using [25, Lemma 1.3.2.2] or [26, 1.3.2].

Convergence to uniform populations is shown by a remarkably simple "geometric series" type argument (*cf.* [24, Thm. 8.2.3, proof], [26, Thm. 3.1.2]) that combines the mutation flow inequality given in line (*12*) and the contraction property of the selection operator given in line (*19*).

Convergence to optima. In order to be able to show convergence to optima, one needs to add one additional ingredient to the conditions already established in lines (*5*), (*6*), (*9*) and (*14*). In fact, one needs

$$2m\kappa_o \ell < s. \tag{22}$$

This inequality shows that for large population size s one is allowed a large m and consequently (*cf.* line (*9*)) a more relaxed annealing schedule for crossover. Thus, crossover (which is related to the algorithm-design since it exchanges structural elements of creatures) is allowed to perform its enhancement of mutation during the mixing phase of the genetic algorithm in more significant way.

Convergence towards optima is shown by establishing a "steady-state flow inequality" similar to [24, p. 54: line (43)] and [26, proof of Thm. 3.4.1, line (26)]. In fact, one analyzes the probabilistic flow between $||P_\Omega v_t||_1$ and $||(\mathbf{1} - P_\Omega) v_t||_1$ under a single application of G_t to v_t, where as above $v_t = G_t v_t$, and P_Ω is the projection

onto the space generated by populations containing only optimal creatures. This yields an inequality implying $||P_\Omega v_t||_1 \to 1$ as $t \to \infty$.

4. Challenges for further research

In order to keep the size of this exposition limited, we can only roughly sketch some ideas for subsequent research activities that continue the line of activities presented thus far. A detailed exposition of open research challenges is presented in [26, Chp. 4].

Non-fully-positive mutation. In the opinion of this author, the most important challenge at this point in theory of genetic algorithms is to find a mathematical approach to the situation where the mutation operator does not yield a fully positive matrix. Providing a framework that would establish convergence results for scaled genetic algorithms with non-fully-positive mutation similar to the results presented here, would —in the opinion of this author— constitute a major mathematical achievement as well as a significant contribution to the field of probabilistic algorithms. Techniques involving modeling via linear operators and Banach spaces ($\ell^{1.2}$) similar to some applications in treating simulated annealing [4, 5, 6, 7] may be of value.

Finite length analysis and noise. After asymptotics for scaled genetic algorithms with non-fully-positive mutation is established satisfactory (presumed easier), one should pursue transition of analysis to finite length algorithms (presumed more difficult) following the historic route for handling the simulated annealing algorithm [1] where substantial results on asymptotics were obtained first (*e.g.*, [11]) and subsequently finite-length algorithms were mastered (*cf.* [4, 5, 6, 7]).

Another interesting point to treat is adding sampling noise following [15]. This is motivated by the fact, that fitness evaluation of creature may be done by executing a simulation with random elements that add "noise" to the fitness value.

Modeling programming. In models for genetic programming, the length of creatures (*i.e.*, programs) is usually not bounded even though realisticly, all programs are of finite length with global upper bound. Thus, the space of possible populations becomes infinite in an idealized setting. It is an interesting mathematical challenge with significant relations to important application aspects to adapt the framework presented above to the general setting for genetic programming.

Going even further, it is, in the opinion of this author, a quite interesting and rewarding challenge to develop a comprehensive operator framework for theoretical treatment of "computability, algorithms and data-structures." Such research should open a larger field of applications to operator theory and yield new insights into problem instances of practical value.

References

[1] E.H.L. Aarts, P.J.M. van Laarhoven. Simulated Annealing: An Introduction. *Statist. Neerlandica* **43** (1989), pp. 31–52

[2] H. Aytug, G.J. Koehler. Stopping Criteria for Finite Length Genetic Algorithms. *INFORMS J. on Computing* **8** (1996), pp. 183–191

[3] H.J. Bremermann, J. Rogson, S. Salaff. Global Properties of Evolution Processes. In: H.H. Pattee, E. Edelsack, L. Fein, A. Callahan (eds.). *Natural Automata and Useful Simulations.* Spartan Books, Washington, DC, USA (1966), pp. 3–42

[4] O. Catoni. *Large Deviations for Annealing.* Ph.D. Dissertation, Université Paris XI, Paris, France (1990)

[5] O. Catoni. Sharp Large Deviations Estimates for Simulated Annealing Algorithms. *Ann. de ℓ'Inst. Henri Poincaré. Prob. & Stat.* **27** (1991), pp. 291–383

[6] O. Catoni. Applications of Sharp Large Deviations Estimates to Optimal Cooling Schedules. *Ann. de ℓ'Inst. Henri Poincaré. Prob. & Stat.* **27** (1991), pp. 493–518

[7] O. Catoni. Rough Large Deviations Estimates for Simulated Annealing — Application to Exponential Schedules. *Ann. of Probability* **20** (1992), pp. 1109–1146

[8] T.E. Davis, J.C. Principe. A Simulated Annealing-like Convergence Theory for the Simple Genetic Algorithm. In: R.K. Belew, L.B. Booker (eds.), *Proc. 4th Int. Conf. on Genetic Algorithms '91*, Morgan Kaufmann Publ., San Francisco, CA, USA (1991), pp. 174–181

[9] T.E. Davis, J.C. Principe. A Markov Chain Framework for the Simple Genetic Algorithm. *Evol. Comput.* **1** (1993), pp. 269–288

[10] D.E. Goldberg. *Genetic Algorithms, in Search, Optimization & Machine Learning.* Addison-Wesley Publ., Boston, MA, USA (1989)

[11] B. Hajec. Cooling Schedules for Optimal Annealing. *Mathematics of Operations Research* **13** (1988), pp. 311–329

[12] D.L. Isaacson, R.W. Madsen. *Markov Chains: Theory and Applications.* Prentice-Hall Publ., Upper Saddle River, NJ, USA (1961)

[13] T. Jansen, I. Wegener. Real royal road functions — where crossover provably is essential. In: L. Spector, E.D. Goodman (eds.). *Proc. GECCO 2001.* Morgan Kaufmann Publ., San Francisco, CA, USA (2001), pp. 375-382

[14] G.J. Koehler. A Proof of the Vose-Liepins Conjecture. *Ann. Math. Artif. Intell.* **10** (1994), pp. 408–422

[15] H.J. Kushner. Asymptotic Global Behavior for Stochastic Approximation and Diffusions with Slowly Decreasing Noise Effects: Global Minimization via Monte Carlo. *SIAM J. Appl. Math.* **47** (1987), pp. 169–185

[16] A.E. Nix, M.D. Vose. Modeling Genetic Algorithms with Markov Chains. *Ann. Math. Artif. Intell.* **5** (1992), pp. 79–88

[17] R. Poli. Exact Schema Theory for Genetic Programming and Variable-Length Genetic Algorithms with One-Point Crossover. *Genetic Programming and Evolvable Machines* **2** (2001), pp. 123–163

[18] G. Rudolph. Convergence Analysis of Canonical Genetic Algorithms. *IEEE Trans. on Neural Networks* **5** (1994), pp. 96–101

[19] H.H. Schaefer. *Banach Lattices and Positive Operators.* Die Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen **215**, Springer Verlag, Berlin, Germany (1974)

[20] V. Savchenko, L.M. Schmitt. Reconstructing Occlusal Surfaces of Teeth using a Genetic Algorithm with Simulated Annealing Type Selection. In: D.C. Anderson, K. Lee (eds.). *Proc. of the Sixth ACM Symp. on Solid Modeling and Appl.* ACM Press, New York, NY, USA (2001), pp. 39–46

[21] L.M. Schmitt, *et al.*. Linear Analysis of Genetic Algorithms. *Theoret. Comput. Sci.* **200** (1998), pp. 101–134

[22] L.M. Schmitt, C.L. Nehaniv. The Linear Geometry of Genetic Operators with Applications to the Analysis of Genetic Drift and Genetic Algorithms using Tournament Selection. Lect. on Math. in the Life Sci. **26**, AMS, Providence, RI, USA (1999), pp. 147–166

[23] L.M. Schmitt, T. Kondoh. Optimization of Mass Distribution in Articulated Figures with Genetic Algorithms. In: M.H. Hamza (ed.). *Proc. IASTED Int. Conf. 'Applied Simulation and Modelling'.* IASTED/ACTA Press, Calgary, AB, Canada (2000), pp. 191–197

[24] L.M. Schmitt. Theory of Genetic Algorithms. *Theoret. Comput. Sci.* **259** (2001), pp. 1–61

[25] L.M. Schmitt. Theory of Genetic Algorithms II. —Models for genetic operators over the string-tensor representation of populations and convergence to global optima for arbitrary fitness function under scaling. *Theoret. Comput. Sci.* **310** (2004), 181–231

[26] L.M. Schmitt. Asymptotic Convergence of Scaled Genetic Algorithms to Global Optima —A gentle introduction to the theory. In: A. Menon (ed.), *Frontiers of Evolutionary Computation.*, Kluwer Ser. in Evol. Comput. (D.E. Goldberg, ed.), Kluwer, Dordrecht, The Netherlands (2004)

[27] L.M. Schmitt: Coevolutionary Convergence to Global Optima. *Proceedings of the Genetic and Evolutionary Computation Conference GECCO 2003*, Chicago, IL, USA(July 12–16, 2003), Erick Cantu-Paz etAI (eds.), Lecture Notes in Computer Science 2723, Springer Verlag, Berlin, Germany(2003), 373-375.

[28] L.M. Schmitt: Optimization with Genetic Algorithms in Multi-Species Environments. *Proceedings of the Fifth International Conference on Computation Intelligence and Multimedia Applications (ICCIMA 2003)*, Xidian University, Xi'an, China (September 27-30, 2003), Sponsored by IEEE, L. Jiao, H. Selvaraj, B. Verma and X. Yao (eds.), IEEE Computer Society (publisher), ISBN 0-7695-1957-1, 194–199

[29] L.M. Schmitt: Theory of Coevolutionary Genetic Algorithms. *Proceedings of 'The 2003 International Symposium on Parallel and Distributed Processing and Applications' ISPA 2003*, The University of Aizu, Aizu-Wakamatsu City, Fukushima, Japan (July 2–4, 2003). M. Guo, L.T. Yang (eds.). Sponsored by ACM and IPSJ. Lecture Notes in Computer Science 2745, Springer Verlag, Berlin, Germany (2003), pp. 285–293

[30] M.D. Vose, G.E. Liepins. Punctuated Equilibria in Genetic Search. *Complex Systems* **5** (1991), pp. 31–44

[31] M.D. Vose. *The Simple Genetic Algorithm: Foundations and Theory.* MIT Press, Cambridge, MA, USA (1999)

The University of Aizu School of Computer Science and Engineering Aizu-Wakamatsu City, Fukushima Prefecture 965-8580, Japan

E-mail address: lothar@u-aizu.ac.jp