Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT
Theoretical Computer Science 310 (2004) 181–231

Theoretical Computer Science

www.elsevier.com/locate/tcs

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

Lothar M. Schmitt

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

Received 6 May 2002; received in revised form 4 July 2003; accepted 11 July 2003
Communicated by M. Ito

Dedicated to Chie Fukuchi in Koga

Abstract

We present a theoretical framework for an asymptotically converging, scaled genetic algorithm which uses an arbitrary-size alphabet and common scaled genetic operators. The alphabet can be interpreted as a set of equidistant real numbers and multiple-spot mutation performs a scalable compromise between pure random search and neighborhood-based change on the alphabet level. We discuss several versions of the crossover operator and their interplay with mutation. In particular, we consider uniform crossover and gene-lottery crossover which does not commute with mutation. The Vose–Liepins version of mutation-crossover is also integrated in our approach. In order to achieve convergence to global optima, the mutation rate and the crossover rate have to be annealed to zero in proper fashion, and unbounded, power-law scaled proportional fitness selection is used with logarithmic growth in the exponent. Our analysis shows that using certain types of crossover operators and large population size allows for particularly slow annealing schedules for the crossover rate. In our discussion, we focus on the following three major aspects based upon contraction properties of the mutation and fitness selection operators: (i) the drive towards uniform populations in a genetic algorithm using standard operations, (ii) weak ergodicity of the inhomogeneous Markov chain describing the probabilistic model for the scaled algorithm, (iii) convergence to globally optimal solutions. In particular, we remove two restrictions imposed in Theorem 8.6 and Remark 8.7 of (Theoret. Comput. Sci. 259 (2001) 1)

E-mail address: lothar@u-aizu.ac.jp (L.M. Schmitt).

0304-3975/$ - see front matter
doi:10.1016/S0304-3975(03)00393-1

where a similar type of algorithm is considered as described here: mutation need not commute with crossover and the fitness function (which may come from a coevolutionary single species setting) need not have a single maximum.

Keywords: Asymptotic convergence of genetic algorithms; Coevolution; Unbounded power-law scaled proportional fitness selection; Non-commuting crossover and mutation operators; Neighborhood-based search; Simulated annealing

0. Introduction

The main purpose of this exposition is to extend the presentation in [56,62][1] in regard to mixing operators in genetic algorithms and to remove two shortcomings in [56, Theorem 8.6, Remark 8.7] which impose certain conditions on the genetic operators in order to obtain a scaled genetic algorithm which converges to global optima. In fact, we remove the conditions that (1) mutation commutes with crossover and that (2) the fitness function has a sole maximum. We achieve these goals by presenting a mostly self-contained, streamlined theory of convergent, scaled genetic algorithms that use a general-size alphabet, standard crossover-mutation operators and unbounded power-law scaled proportional fitness selection. Overall, we pursue the following major objectives:

1. The alphabet $\mathscr{A}$ is primarily interpreted as a finite set of equidistant real numbers such that the genetic algorithms discussed here are well-suited for optimization in a compact domain in $\mathbf{R}^\ell$, $\ell \in \mathbf{N}$. This follows an idea formulated, e.g., by Markus et al. in [40] and in [56]. On the level of the alphabet, we discuss the spot-mutation operator $\mathbf{m}_{\mu_0}^{(1)}$ which performs a scalable compromise between pure random change and a neighborhood-based hill-climbing strategy analogous to a similar strategy for the simulated annealing algorithm. See the paper by Aarts and van Laarhoven [1] for an overview on simulated annealing.
2. Based upon the analysis of the spot-mutation operator $\mathbf{m}_{\mu_0}^{(1)}$, we study multiple-spot mutation in regard to certain contraction properties that ensure the following: (1) weak ergodicity of the inhomogeneous Markov chain describing the probabilistic model for the scaled genetic algorithm and, together with a contraction property of the fitness selection operator, (2) convergence towards uniform populations.
3. We allow essentially all standard types of crossover operators under the following three conditions: (1) the crossover operation leaves uniform populations invariant, (2) the entries of the stochastic matrix describing the crossover operation are rational functions in the crossover rate without singularities and (3) the crossover operation converges to the identity operation, if the crossover rate converges to zero. In particular, we allow appropriately scaled uniform crossover in our setting. In addition, we present an analysis of gene-lottery crossover which yields an example

[1] We point out to the reader that it is desirable but *not necessary* to read [56,62] in order to understand most of this presentation. We shall only refer to [56,61,62], if a side-track result can be obtained from results presented here by already well-established straightforward techniques.

Table 1

Step 0	Initialize population $p = (c_1, \ldots, c_s)$ with creatures $c_1, \ldots, c_s \in \mathscr{C}$.
Step 1	Apply crossover to creatures in p.
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.

of non-commuting crossover and mutation operators. Our discussion also contains an improved model for one-cutpoint regular crossover compared to the analysis by Schmitt et al. [62, Section 2.2] and a model for the Vose–Liepins version of mutation-crossover [68, p. 44], [69].

4. Asymptotic convergence of genetic algorithms to global optima with *not-necessarily-commuting crossover and mutation operators* is shown under the condition that the mutation and crossover rates are properly annealed to zero and unbounded power-law scaled proportional fitness selection is used with logarithmic growth in the exponent. The *condition of a single maximum* imposed upon the fitness function in [56, Theorem 8.6, Remark 8.7] is thereby removed. Thus, our analysis is, in particular, more general than the approach taken in Vose's book where it is always assumed that the fitness function is *injective* (cf. [68, p. 25, footnote]).

As Mühlenbein [43] points out in the introduction to his survey on genetic algorithms, evolutionary algorithms based upon "mutation, mating and selection" were already introduced in the 1960s as a tool for optimization. One example of such work is the paper by Bremermann et al. [8]. Genetic algorithms, a particular case of evolutionary algorithms, were invented by Holland [29] and are by now a well-established tool for search and optimization. A given optimization task is encoded in such a way that candidate solutions are understood as elements in a finite collection $\mathscr{C}$ of creatures in a model "world" and a fitness function $f : \mathscr{C} \to \mathbf{R}^+$ exists which has to be maximized. In the model for genetic algorithms presented in this exposition, creatures (candidate solutions) are identified with their genetic information which typically consists of an ordered string of coefficients selected from an appropriately chosen alphabet $\mathscr{A}$ such as, e.g., $\{0, 1\}$ or an equidistant set of real numbers. Non-binary genetic algorithms have previously been investigated, e.g., by Bhattacharyya and Koehler [7], Koehler et al. [32], Leung et al. [35] and Nomura and Shimohara [45]. The collection of creatures in the current population p of fixed size s is subject to three operations: crossover, mutation and selection which are applied cyclically and iteratively until a termination condition is satisfied. Overall, the genetic algorithm is described in Table 1.

A genetic algorithm is called *simple*, if all operations in steps 1–3 stay constant over the course of the algorithm.

In this work, we shall be interested in the asymptotic behavior of the genetic algorithm defined by the above table, i.e., the probabilistic behavior of the algorithm if never halted in step 4 above. Asymptotic behavior of genetic algorithm has been investigated by many authors: Agapie [2], Aytug and Koehler [4], Cerf [14,15], Davis [16], Davis and Principe [17,18], Fogel [20], Goldberg [24], He and Kang [28],

Holland [29], Leung et al. [35], Liepins and Vose [69], Lozano et al. [36], Mahfoud [37], Mahfoud and Goldberg [38,39], Nix and Vose [44], Rudolph [50], Suzuki [65,66], and Vose [68]. See also related work in the case of genetic programming by Poli [46] and Poli and Langdon [47]. The analysis presented in this work sets boundary conditions for proper design and implementation of genetic algorithms that actually do stop after a finite but large number t of cycles, $t \in \mathbf{N}$. The proofs of Theorem V.4.3 (p. 160) in the book by Isaacson and Madsen [30] or [59, Theorem 3.3.2] show that for large t the probability distribution describing the state of the algorithm after t steps is close to the limit of the steady-state distributions of the individual steps (see the discussion in Sections 2.6, 3.3, Theorem 3.3.2). In fact, the algorithm follows the trajectory of steady-state distributions of the individual steps. This, in principle, allows for development of a stopping-criterion for a genetic algorithm which is scaled as in Theorem 3.3.2 and its Corollaries. However, such a stopping-criterion should depend upon an analysis of the problem instance.

Aarts and van Laarhoven describe in [1, pp. 41–43; p. 49: first •] the difficulty to obtain applicable, general "black-box-scenario" stopping criteria for the simulated annealing algorithm. Work by Catoni [10–13] constitutes a major advance in this regard and should be generalized to the case of genetic algorithms in future work. In fact, Catoni analyses stopping the simulated annealing algorithm after finitely many steps and finding an optimal cooling schedule for that purpose rather than studying the asymptotic behavior of simulated annealing. A similar analysis (which the author considers an outstanding problem) should be carried out for genetic algorithms preferably with non-fully positive mutation. For genetic algorithms, the stopping-criterion by Aytug and Koehler [4] is a valuable step. But since this result is based upon the mutation rate, it is the most general result possible and therefore of limited practical use. A similar approximation result for the steady-state distribution based upon the mutation rate is obtained by Leung et al. [35].

After discussing asymptotics, let us discuss the steps of the genetic algorithm as outlined in the above table. We note that the combined crossover-mutation phase of a genetic algorithm is also called the generator phase or mixing phase of the algorithm, cf. [68, p. 32]. Genetic information is recombined by the crossover operation and is slightly altered under mutation.

Crossover models the exchange of genetic information of creatures and is inspired by exchange of genetic information in living organisms, e.g., during the process of sexual reproduction. It is discussed in Sections 2.3–2.5. The notion needed to show convergence of genetic algorithms is that of a rational generalized crossover defined in Section 2.3. Examples for rational generalized crossover operators can be found in the literature, e.g., in [7,35]. We shall mainly be concerned with four examples for crossover methods: one/two-cutpoint regular crossover, uniform regular crossover and gene-lottery crossover. First, we consider one-cutpoint regular crossover as in [62, Section 2.2, "simple crossover"]; [56, Section 5.2], i.e., a procedure where the creatures $c_1, \ldots, c_s$ in the even-size population are sequentially paired and the one-cutpoint crossover operation is then applied to each of the pairs (c_1, c_2), $\ldots$ (c_{s-1}, c_s) with probability χ. This follows, e.g., Goldberg's approach [23, pp. 16–17]. The new model for one-cutpoint regular crossover presented in Section 2.4 yields a significant

improvement in regard to determining the spectrum of the crossover matrix. The result is in complete correspondence (not identical though) with Koehler's Theorem [31, p. 419]. It illustrates in a significant way why Koehler's Theorem actually holds and in which way mutation and crossover contribute to the spectrum of the mixing matrix. Our analysis of one-cutpoint regular crossover also yields a convenient way to describe two-cutpoint regular crossover and uniform regular crossover using the tensor-space description of the vector-space $\mathscr{V}_{\wp}$ underlying our model (see Eq. (7) below). A second principal model for crossover that we shall consider is the so-called gene-lottery crossover where for every creature in the next generation (population), the letter (allele) at a given spot λ is selected probabilistically from the alleles at spots λ of all creatures in the present population. This provides an explicit example of a reasonable crossover operation that *does not commute* with the mutation operation. However, a strong convergence result can be obtained for a properly scaled genetic algorithm that employs the quite "destructive" gene-lottery crossover procedure (see Corollary 3.3.4). Observe that in every example for crossover discussed here, the given crossover operation may alter every creature in the population. Thus, the population before crossover and the population after crossover may be disjoint, if they are seen as sets of creatures.

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. The spot mutation matrix $\mathbf{m}_{\mu_0}^{(1)}$ which models change by mutation on the level of the alphabet is discussed in Section 2.1. The multiple-spot mutation operator $M_{\mu_0,\mu}^{(m)}$ which acts on entire populations and is based upon the local action of $\mathbf{m}_{\mu_0}^{(1)}$ is discussed in Section 2.2. Multiple-spot mutation has been studied in a Markov chain framework over finite populations by many authors. Among earlier references that have been used by others are the work of Davis and Principe [17,18] and Nix and Vose [44].

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 work, we shall restrict the analysis to scaled proportional fitness selection based upon a given fitness function f (consult, e.g., Goldberg's book [23, p. 16]; [54, Section 2.3 or 58, Section 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. Thereby, the value $f(c,p)$ of the fitness function f for a particular creature c may depend upon the population p the creature resides in. Thus, our analysis includes the case of rank of creatures within a population based upon a given raw fitness function $\mathscr{C} \to \mathbf{R}^+$. See, e.g., [58, Section 7.3] for a suitable definition[2] of rank. Using a selection method based upon rank induced by a given raw fitness function was proposed by Baker [5]. Scaled proportional fitness selection is discussed in Section 2.6.

[2] Note that our definition of rank simply assigns an integer value in $[1,s]$ to creatures in the populations (with largest value for maximal creatures in the population) depending upon the quasi order induced by the fitness function on the creatures in the *current* population. This is different from the definition of "ranking selection" given in [68, p. 25].

De Jong [19] stressed the need for a theoretical framework for coevolutionary genetic algorithms and possible convergence theorems in regard to coevolutionary optimization ("arms races") which require treatment of a population-dependent fitness function. We note at this point that the main results of this exposition (Theorem 3.3.2 and its corollaries) solve the coevolutionary optimization problem for a single-species setting, if a set of agents exist that are strictly superior in every population they reside in. Furthermore, [56, Theorem 8.6, Remark 8.7] already solve the coevolutionary optimization problem for a single-species setting, if a single dominant agents exists. These results have recently been generalized for multi-species settings in [58–60].

The mathematical model presented in this exposition uses an inhomogeneous Markov chain over a finite set $\wp$ of populations. Consequently, the state space $\mathscr{S}_\wp$ of the genetic algorithm consists of probability distributions over populations $p \in \wp$. These probability distributions are considered as elements of the free vector space $\mathscr{V}_\wp$ over $\wp$, i.e., $\mathscr{S}_\wp \subset \mathscr{V}_\wp$. Thus, the stochastic matrices representing the genetic operators crossover, mutation and selection in the model for a scaled genetic algorithm presented here act on $\mathscr{V}_\wp$. Consult Section 5 for a concrete example that illustrates this setting.

Most authors represent populations as multi-sets following the work of Davis and Principe [17,18], Liepins and Vose [69] and Nix and Vose [44]. A more general Markovian framework for stochastic search methods using multi-sets is given by Vose's theory of random heuristic search [67,68]. As in [56,61,62], the representation used in this exposition considers populations as strings of letters in the underlying alphabet and not as multi-sets. As outlined in [56, Section 2.9], the multi-set representation can easily be embedded into the tensor-string representation considered here. Rudolph [50] developed his Markov chain model for genetic algorithms over the tensor-string representation for populations approximately around the same time as [62].

A specific example that shows the concurrent use of the tensor-string representation for populations and the stochastic matrices modeling the genetic operators is listed in Section 5.

What makes our approach different is that we do not attempt to unite the genetic operators crossover, mutation and selection to *one* operator which is subsequently analyzed. We rather analyze the genetic operators separately to isolate key properties: (1) crossover plays a dual role enhancing mutation in the mixing phase of the algorithm ([56, Theorem 6.1]) as well as enhancing selection in the contraction phase of the algorithm in some cases (end of Section 2.5 and Proposition 2.6.2); (2) mutation is responsible for weak ergodicity (Theorem 3.2.1) and the flow away from uniform populations (Proposition 2.2.3); (3) fitness selection is responsible for contraction towards uniform populations (Proposition 2.6.1); (4) mutation-selection is responsible for convergence to uniform populations in the zero mutation rate limit (Theorem 3.1.1); (5) all three genetic operators act together to obtain the steady-state flow inequality which shows convergence to global optima (proof of Theorem 3.3.2, Part 3c).

Various convergence results for genetic algorithms can be found in the literature. However, a *specific condition* limiting generality is attached to most of these results. Rudolph [50] studied genetic algorithms together with an *elitist strategy* that always keeps the best creature found so far. Obviously, this algorithm must converge eventu-

ally. However, Example 2 in [56, Section 8.3] shows that an adverse fitness landscape may lead an ill-designed genetic algorithm away from global optima. In such cases, the simple genetic algorithm with elitist strategy may behave worse than enumerative search. He and Kang [28] have developed estimates for the rate of convergence of elistist-type genetic algorithms. Their estimates depend upon *first visit times*. With similar strategy, Greenwood and Zhu [26] investigate the (μ, λ) and $(\mu + \lambda)$ evolutionary strategy algorithms with self-adaptation[3] for $(\mu + \lambda) = (1+1)$ and $(\mu, \lambda) = (1, 1)$. Greenwood and Zhu consider functions over Lebesque-measurable domains with *non-isolated singularities* and find the optimum *up to error* $\varepsilon > 0$. In principle, their analysis relies on eventual mutation into a proper neighborhood of global optima similar to Rudolph's case. Extending Rudolph's analysis of elitist-strategy genetic algorithms, Agapie [2] presents an analysis of a mutation-adaptive simple genetic algorithm that is convergent under certain conditions [2, Theorem 4.3]. Agapie requires that every non-optimal populations p is *inessential*, i.e., there is another population q (sink) that can be reached from p but offers no way to return to p. Agapie presents an estimate of the convergence rate in [2, Theorems 2.9 and 5.2]. Cerf [14,15] has developed a framework for a convergent genetic algorithm which requires a (larger) *populations size that strongly depends upon the optimization problem*. Vose's approach to convergence of genetic algorithms under the framework of random heuristic search [68, Chapter 3] requires an *infinite population limit* for convergence to stable fixed points [68, p. 147].

A proof of asymptotic convergence for a genetic algorithm of fixed, relatively small population size using scaled proportional fitness selection based upon an inhomogeneous Markov chain model has only recently been obtained in [56, Theorem 8.6, Remark 8.7] essentially for fitness functions $\mathscr{C} \to \mathbf{R}^+$ with a single global maximum or rank based upon such fitness functions. We remark that, unfortunately, the proof of convergence in [65,66] fails for principal reasons, cf. [56, Section 8.3]. As outlined in [62, pp. 120–121], the condition of an injective fitness function is not so much of a restriction in regard to function optimization. However, in regard to ease of implementation[4] and applicability, in regard to mathematical generality and in regard to the analogue with the artificial cooling process in the simulated annealing[5] algorithm [1] as advocated by Davis and Principe [18], it is desirable to remove the condition of a single global maximum or injectivity for the given (raw) fitness function.

The main results of this exposition, Theorem 3.3.2 and its two Corollaries, achieve the following goals: (1) a general-purpose, scaled, converging genetic algorithm is

[3] The $(\mu+\lambda)$ evolutionary strategy generates $\lambda \geqslant \mu$ offspring from μ creatures and selects the μ fittest among the available $\mu + \lambda$ creatures. The (μ, λ) evolutionary strategy generates $\lambda \geqslant \mu$ offspring from μ creatures and selects the μ fittest among the λ offspring. Self-adaptation modifies Rechenberg's $\frac{1}{5}$-rule [48].

[4] One avoids to implement a procedure which makes the raw fitness function artificially injective at runtime, or to assign "maxrank+ε" to the best creature found thus far in the course of the algorithm (consult [56, Theorem 8.6, Remark 8.7]). Such techniques may bias the outcome of the genetic algorithm considerably.

[5] Speculating, one may anticipate to implement a genetic algorithm in the near future based upon DNA/RNA-based encoding and computing where the parameters such as mutation rate and selection pressure are controlled, e.g., by the abundance of certain enzymes in a chemical solution, the temperature, radiation or sound. In such a setting, it seems impractical to attempt to implement special book-keeping procedures.

described whose setup is quite similar to that of the simulated annealing algorithm; (2) explicit cooling schedules for not-necessarily commuting mutation-crossover and exponentiation schedules for fitness selection are given; (3) no conditions are attached, in particular, the fitness function need not be injective and the population size s can stay small and controllable. In fact, s can be set to $\ell+1$ where ℓ is the length of the genome of creatures (candidate solutions). The genetic algorithm presented in this paper consequently satisfies *all* goals formulated by Davis and Principe [18, p. 270].

Finally, let us mention that Lozano et al. [36] have developed a genetic algorithm with a simulated-annealing-type selection strategy which converges asymptotically to global optima. This provides an alternative to the selection mechanism developed in this exposition.

1. Notation and preliminaries

Before we describe the proposed scaled genetic algorithm, investigate its components and prove its asymptotic convergence, we need to collect a number of definitions and elementary facts in this section. The notation used here is essentially the same as in [56, Section 2] up to some additions and simplifications. To keep this presentation mostly self-contained, we include a listing which is complete in regard to almost the entire exposition for the convenience of the reader. Consult also Section 5 for a tutorial on the notation used in this exhibition.

1.1. Symbols and keywords

The following index of mathematical symbols is listed in the order of appearance of items: 1.2: $\mathbf{Z}$, $\mathbf{R}$, $\mathbf{R}^+$, $\mathbf{C}$, $\mathbf{R}^+_*$, $\mathbf{N}_0$, $\mathbf{N}$, $\delta_{n,m}$, $\langle\cdot\rangle$, $\|\cdot\|_1$, $(\cdot)^*$, e. 1.3: $\mathbf{M}_k(\Omega)$, $\mathbf{M}_k$, $X^{[\cdot]}$, operator norm $\|\cdot\|_1$, $\mathbf{1}$, P_e, $\prod_{\tau=t}^{s} X_\tau$, stochastic, irreducible, fully positive. 1.5: $\mathscr{A}$, α, $a(\imath)$, $d_{\mathscr{A}}$, n, close neighbors, $\mathscr{N}_n(\imath)$, $\mathscr{V}_1$. 1.6: ℓ, $\mathscr{C}$, $\wp$, s, L, J, set(p), $p\wedge J$, $c\in p$, spot, Δ, $\Delta_{\pm n}$, $\mathscr{V}_\wp$, $\mathscr{V}_C$, $\mathscr{U}$, $\mathscr{S}_\wp$,. 1.7: GFV$_u$, mean$_{\mathbf{u}}$. 2.1: $\mathbf{m}^{(1)}_{\mu_0}$, $\mathbf{f}$. 2.2: $M^{(m)}_{\mu_0,\mu}$. 2.3: $C(\chi)$. 2.4: $C^{(1)}_{\mathrm{reg}}(\chi)$, $C^{(2)}_{\mathrm{reg}}(\chi)$, $C^{(u)}_{\mathrm{reg}}(\chi)$, 2.5. $C^{(m)}_{\mathrm{glc}}(\chi)$, 2.6. f, D_f, $\mathscr{C}_{\max}$, $\rho_2(f)$, unbounded power-law scaling f_t, logarithmic exponentiation schedule g, proportional fitness selection S^f_t. 3.2: annealing schedules for mutation, ϕ_M, ϕ_0, ϕ'_0, t_0. 3.3: SMC genetic, algorithm, SCM genetic algorithm, annealing schedules for crossover (3.3.2). 4.2: P_Π, $\bar{D}$, D. 4.3: r, VLGA.

1.2. Scalars and vectors

Let $\mathbf{Z}$, $\mathbf{R}$, $\mathbf{R}^+$ and $\mathbf{C}$ denote the integers, the real numbers, the non-negative real numbers and the complex numbers, respectively. Let $\mathbf{R}^+_* = \mathbf{R}^+\setminus\{0\}$, $\mathbf{N}_0 = \mathbf{Z}\cap\mathbf{R}^+$ and $\mathbf{N} = \mathbf{Z}\cap\mathbf{R}^+_*$. For elements k,m of a set, let $\delta_{k,m}=1$, if $k=m$, and let $\delta_{k,m}=0$ otherwise, i.e., δ is the Kronecker delta. Recall that for $v=(v_\kappa)_{\kappa=0}^{k-1}, w=(w_\kappa)_{\kappa=0}^{k-1}\in\mathbf{C}^k$, $k\in\mathbf{N}$, the canonical inner product of v and w and the usual Hamming-norm or ℓ^1-norm of v are

given by

$$\langle w, v\rangle = \sum_{\kappa=0}^{k-1} \bar{w}_\kappa v_\kappa \quad \text{and} \quad \|v\|_1 = \sum_{\kappa=0}^{k-1} |v_\kappa|. \tag{1}$$

We shall use the notation x^* to denote the adjoint of a vector or matrix x. Let $e = k^{-1} \cdot (1, 1, \ldots, 1)^* \in \mathbf{R}^k$. Observe that e is the maximal entropy probability distribution (over k elements) in the positive part of the $\|\cdot\|_1$-unit sphere of $\mathbf{R}^k$ (see, e.g., [56, Section 2.8]). The notation for e differs by a factor 2^{-L} from the notation introduced in [52, p. 104], but coincides with the notation used in [56,61].

1.3. Matrices

Let $\mathbf{M}_k(\Omega)$, $k \in \mathbf{N}$, denote the $k \times k$ matrices with entries in a set Ω. Let $\mathbf{M}_k = \mathbf{M}_k(\mathbf{C})$. We shall enumerate the coefficients of a matrix in $\mathbf{M}_k(\Omega)$ with indices running from 0 through $k-1$. A matrix in $\mathbf{M}_k$ will operate by matrix multiplication from the left on column vectors in $\mathbf{C}^k$. If $X \in \mathbf{M}_k$ and $v \in \mathbf{C}^k$ is a row vector, then we shall write $X^{[v]}$ for the matrix obtained from X by replacing the first row of X with v. The operator norm of $X \in \mathbf{M}_k$ is given by $\|X\|_1 = \sup\{\|Xv\|_1 : v \in \mathbf{C}^k, \ \|v\|_1 = 1\}$ as in [53, p. 5, Eq. (5)]. The matrix associated with the identity map $\mathbf{C}^k \to \mathbf{C}^k$ will be denoted by $\mathbf{1}$. Let $P_e = (e, e, \ldots, e) \in \mathbf{M}_k$ be the orthogonal projection onto $\mathrm{span}_{\mathbf{C}}(\{\mathrm{e}\})$. In contrast to some conventions, we shall use the $\prod$-symbol to denote products of possibly *non-commuting* matrices as follows: for $X_\tau \in \mathbf{M}_k$, $s, t \in \mathbf{Z}$, $s \neq t$, let

$$\prod_{\tau=t}^{s} X_\tau = X_t \cdot X_{t+\mathrm{sign}(s-t)} \cdots X_s, \qquad \prod_{\tau=s}^{s} X_\tau = X_s. \tag{2}$$

A matrix in $\mathbf{M}_k(\mathbf{R}^+)$ is called column-stochastic or for short *stochastic*, if each of its columns sums to 1. See Schaefer's book [53] for a good and short introduction to theoretical aspects of stochastic matrices.

$X \in \mathbf{M}_k$ shall be called *reducible* or decomposable, cf. [53, p. 19], if there exists a permutation matrix $U \in \mathbf{M}_k(\{0,1\})$ such that

$$X = U \cdot \begin{pmatrix} X_1 & X_2 \\ 0 & X_3 \end{pmatrix} \cdot U^{-1} = U \cdot \begin{pmatrix} X_1 & X_2 \\ 0 & X_3 \end{pmatrix} \cdot U^*, \tag{3}$$

where $X_1 \in \mathbf{M}_m$ for some $m \in \mathbf{N}$, $1 \leqslant m < k$. Otherwise, $X \in \mathbf{M}_k$ shall be called *irreducible* or indecomposable. A matrix in $\mathbf{M}_k(\mathbf{R}_*^+)$ will be called *fully positive* and is obviously irreducible.

In order to treat weak ergodicity of the inhomogeneous Markov chain underlying the model for the scaled genetic algorithm considered in this exposition, we can take into account that multiple-spot mutation contributes fully positive matrices to this Markov chain (see Proposition 2.2.2.1). The following, possibly known Lemma shortens the discussion of weak ergodicity considerably compared to the discussion in the books by Isaacson and Madsen [30, pp. 142–151, p. 151: Theorem V.3.2] or Seneta [63, pp. 85–86, 134–142, p. 137: Theorem 4.8, p. 141: Theorem 4.9].

Lemma 1.3.1. *Let* $k \in \mathbf{N}$ *and* $\varepsilon_t \in (0, 1/k)$ *for* $t \in \mathbf{N}$ *such that* $\sum_{t\in\mathbf{N}} \varepsilon_t = \infty$. *Let* $M_t \in \mathbf{M}_k([\varepsilon_t, 1])$ *and* $X_t, Y_t \in \mathbf{M}_k$ *be stochastic matrices. Let* $v, w \in (\mathbf{R}^+)^k$ *such that* $\|v\|_1 = \|w\|_1 = 1$, i.e., v *and* w *are probability distributions. Then we have*
1. $\|M_t(v-w)\|_1 \leqslant (1-\varepsilon_t k)\|v-w\|_1$.
2. $\lim_{t\to\infty} (\prod_{\tau=t}^{1} X_\tau \cdot M_\tau \cdot Y_\tau)(v-w) = 0$.

Proof. One has $M_t = \varepsilon_t k P_e + M_t'$ with $M_t' \in \mathbf{M}_k(\mathbf{R}^+)$. Using [53, p. 5, Eq. (7′)], we conclude that $\|M_t'\|_1 \leqslant 1-\varepsilon_t k$. Hence, $\|M_t(v-w)\|_1 = \|M_t'(v-w)\|_1 \leqslant (1-\varepsilon_t k)\|v-w\|_1$, since $v-w \in e^\perp$ and, consequently, $P_e(v-w) = 0$. This shows statement (1). Applying statement (1) to the M_τ, $\tau \in \mathbf{N}$, we have

$$\left\| \left(\prod_{\tau=t}^{1} X_\tau \cdot M_\tau \cdot Y_\tau \right) (v-w) \right\|_1 \leqslant \|v-w\|_1 \cdot \prod_{\tau=1}^{t} (1-\varepsilon_\tau k),$$

since the X_τ and Y_τ are stochastic and have operator norm equal to 1 by [53, p. 5, Eq. (7′)]. The latter product converges to 0 for $t \to \infty$, if $\sum_{\tau\in\mathbf{N}} \varepsilon_\tau = \infty$. In the non-trivial case $\lim_{\tau\to\infty} \varepsilon_\tau = 0$, this can be easily shown by taking logarithms. □

1.4. Application of Frobenius theory

The following collection of arguments constitutes a well-known consequence of Frobenius' celebrated result [53, p. 22, Theorem 6.5]. See also [57, Section 1.3].

Lemma 1.4.1. *Let* $X \in \mathbf{M}_k$ *be a stochastic matrix*, $k \in \mathbf{N}$.
1. Ref. [53, p. 7, Theorem 2.3] *shows that* X *has an eigenvector* v^+ *to eigenvalue* 1 *with positive entries.* [53, p. 13, Proposition 4.2] *shows that* X *has spectral radius* 1. *Since the sequence* $(X^m)_{m\in\mathbf{N}}$ *is bounded, we conclude by* [53, p. 11, Proposition 3.4] *that* 1 *is a simple pole of the resolvent.*
2. *Suppose that for every pair of indices* (κ, κ'), $0 \leqslant \kappa, \kappa' < k$, *there exists* $m \in \mathbf{N}$ *such that* $(X^m)_{\kappa,\kappa'} > 0$. *Then* X *is irreducible as remarked in the discussion following* [53, p. 20, Proposition 6.2].
3. *Suppose that* X *is irreducible and at least one diagonal entry of* X *is strictly positive. By* [53, p. 23, Corollary 2], *one obtains that* 1 *is the only eigenvalue of absolute value* 1 *and a simple root of the characteristic equation. By the remarks following* [53, p. 9, Proposition 2.8], *one obtains that the eigenspace pertaining to eigenvector* 1 *is one-dimensional and spanned by* v^+ *obtained in statement* (1) *above.*

The following Lemma will be used to show that the stochastic matrix associated with an individual step of a scaled genetic algorithm has a positive, invariant eigenvector which is uniquely determined up to scalar multiples.

Lemma 1.4.2. *Let* $M \in \mathbf{M}_k(\mathbf{R}_*^+)$ *and* $X \in \mathbf{M}_k$ *be stochastic matrices. Then we have*
1. $MX \in \mathbf{M}_k(\mathbf{R}_*^+)$. *Consequently*, MX *has an invariant eigenvector* $v = MXv \in \mathbf{C}^k$ *which as such is uniquely determined up to scalar multiples. Furthermore*, v *can be chosen such that* $v \in (\mathbf{R}_*^+)^k$ *and* $\|v\|_1 = 1$.

2. *Suppose that v obtained in statement (1) satisfies $v \in (\mathbf{R}_*^+)^k$ and $\|v\|_1 = 1$. In addition, assume that M is invertible. Then, $w = M^{-1}v$ is an invariant eigenvector of XM which as such is uniquely determined up to scalar multiples. Furthermore, $w \in (\mathbf{R}^+)^k$ and $\|w\|_1 = 1$.*

Proof. MX is fully positive since M is such. Lemma 1.4.1.3 shows that MX has an invariant eigenvector $v \in \mathbf{C}^k$ which is uniquely determined up to scalar multiples, and one can assume $v \in (\mathbf{R}^+)^k$. v can be normalized such that $\|v\|_1 = 1$. The identity $MXv = v$ shows that $v \in (\mathbf{R}_*^+)^k$. This completes the proof of statement (1). Let w be any possible invariant eigenvector of XM. Then we have $MXMw = Mw$. Thus, $Mw = \zeta v$ for a $\zeta \in \mathbf{C}$. Since M is invertible, one obtains that w is uniquely determined up to scalar multiples. By Lemma 1.4.1.1, we know that XM has an invariant eigenvector $w \in (\mathbf{R}^+)^k$. Suppose that w is normalized such that $\|w\|_1 = 1$. Then, Mw is a probability distribution since M is stochastic. Hence, $Mw = v$. This completes the proof of statement (2). □

We point out to the reader that [57, Sections 1.1–1.3, 5] presents a collection of simple, likely known arguments that provide coverage and detailed proof for the facts listed in Sections 1.2–1.4 of this exposition.

1.5. The alphabet and the basic vector space

The letters in the alphabet $\mathscr{A}$ of size $\alpha, 2 \leqslant \alpha \in \mathbf{N}$, underlying the mathematical model for the optimization algorithm described in this work shall be denoted by $a(0), a(1), \ldots, a(\alpha - 1)$. Sometimes but only if explicitly stated, we shall identify $\mathscr{A}$ with $\mathbf{Z}_\alpha = \mathbf{Z}/\alpha\mathbf{Z}$ such that under this identification $a(\imath) \equiv \imath$, $0 \leqslant \imath \leqslant \alpha - 1$. This identification allows to define a shortest cyclic distance function $d_{\mathscr{A}} : \mathscr{A} \times \mathscr{A} \to [0, \lfloor \alpha/2 \rfloor] \cap \mathbf{N}_0$ by setting

$$d_{\mathscr{A}}(a(\imath), a(\imath')) = d_{\mathbf{R}}(\imath + \alpha\mathbf{Z}, \imath' + \alpha\mathbf{Z}). \tag{4}$$

Let $n \in \mathbf{N}$ such that $n < \alpha/2$. We shall say that $a(\imath), a(\imath') \in \mathscr{A}$ are *close neighbors*, if $\imath \neq \imath'$ and $d_{\mathscr{A}}(a(\imath), a(\imath')) \leqslant n$. Let $\mathscr{N}_n(\imath)$ be the set of close neighbors of $a(\imath)$, $0 \leqslant \imath \leqslant \alpha - 1$.

Let $\mathscr{V}_1$ be the free vector space over $\mathscr{A}$. We shall identify $\mathscr{V}_1$ with $\mathbf{C}^\alpha$, i.e., define a linear map $\mathscr{V}_1 \to \mathbf{C}^\alpha$, such that $a(\imath) \in \mathscr{V}_1$ is mapped to $b_\imath$, where $b_\imath = (\delta_{\imath,\imath'})_{\imath'=0}^{\alpha-1} \in \mathbf{C}^\alpha$ is a standard unit base vector, $0 \leqslant \imath \leqslant \alpha - 1$.

1.6. Creatures and populations

We shall consider creatures or candidate solutions in the model world to which the genetic algorithm is applied as ℓ-tuples over the alphabet $\mathscr{A}$ where $2 \leqslant \ell \in \mathbf{N}$. Thus, creatures have a genome of length ℓ. Let $\mathscr{C} = \mathscr{A}^\ell$ denote the set of possible creatures. The number of elements in $\mathscr{C}$ is given by $\#(\mathscr{C}) = \alpha^\ell$.

The set of populations $\wp$, to which the genetic algorithm is applied, is the set of s-tuples of creatures, $s \in \mathbf{N}$. We shall assume that s is even and $s \geqslant 4$, if not explicitly stated otherwise. Let $L = \ell \cdot s$. Then, $\#(\wp) = \alpha^L$.

Let $J \subset \{1,\ldots,s\}$. The set J will act as a selector mask. We shall mainly consider $J = \{1,\ldots,s\}$ as the regular setting for this exposition. We shall set $J = \{1,\ldots,s\} \cap 2\mathbf{N}$ for incorporating the Vose–Liepins version of crossover-mutation in a genetic algorithm. The latter is discussed in Section 4.3. If $p = (c_1, c_2, \ldots, c_s)$ is a population, $c_\sigma \in \mathscr{C}$, $1 \leqslant \sigma \leqslant s$, then we define $\mathrm{set}(p) = \{c_\sigma\colon\ 1 \leqslant \sigma \leqslant s\}$ and $p \wedge J = (c_\sigma)_{\sigma \in J}$. If $c \in \mathscr{C}$, then we shall write $c \in p$, if $c \in \mathrm{set}(p)$.

A *spot* in the genome is, by definition, the position of one of the letters in a word over $\mathscr{A}$ representing a creature or population. For $p, q \in \wp$, we define the Hamming distance $\Delta(p,q)$ as the number of spots in the genome where p and q differ. Similarly, we define $\Delta_{\pm n}(p,q)$ as the number of spots in the genome where p and q differ by a close neighbor in the sense of Section 1.5. Thus, for $p = (c_1, c_2, \ldots, c_s)$, $q = (d_1, d_2, \ldots, d_s)$, $c_\sigma, d_\sigma \in \mathscr{C}$, $1 \leqslant \sigma \leqslant s$, $c_\sigma = (a(\imath_1^{(0,\sigma)}), \ldots, a(\imath_\ell^{(0,\sigma)}))$, $d_\sigma = (a(\imath_1^{(1,\sigma)}), \ldots, a(\imath_\ell^{(1,\sigma)}))$, $\imath_\lambda^{(\kappa,\sigma)} \in [0, \alpha - 1] \cap \mathbf{N}$, $1 \leqslant \lambda \leqslant \ell$, $\kappa = 0, 1$, $I = \{(\sigma,\lambda)\colon\ 1 \leqslant \sigma \leqslant s, 1 \leqslant \lambda \leqslant \ell\}$

$$\Delta(p,q) = \#\{(\sigma,\lambda) \in I\colon\ \imath_\lambda^{(0,\sigma)} \neq \imath_\lambda^{(1,\sigma)}\}, \tag{5}$$

$$\Delta_{\pm n}(p,q) = \#\{(\sigma,\lambda) \in I\colon\ \imath_\lambda^{(0,\sigma)} \neq \imath_\lambda^{(1,\sigma)} \text{ and } d_{\mathscr{A}}(a(\imath_\lambda^{(0,\sigma)}), a(\imath_\lambda^{(1,\sigma)})) \leqslant n\}. \tag{6}$$

We define the vector-space $\mathscr{V}_\wp$ underlying our model for genetic algorithms as the free complex vector space over $\wp$. Thus, $\dim(\mathscr{V}_\wp) = \#(\wp) = \alpha^L$. Every population p can be identified canonically with an integer in $[0, \alpha^L - 1]$, i.e., the letters comprising p are used as digits to describe a number in the α-adic number system. This induces a natural order on $\wp$ and is used to index matrices acting on $\mathscr{V}_\wp$. See Section 5.1 for an example in this regard. $\mathscr{V}_\wp$ is further identified with the L-fold tensor product of $\mathscr{V}_1$ as follows:

$$\mathscr{V}_\wp = \otimes_{\lambda=1}^{L} \mathscr{V}_1. \tag{7}$$

With the tensor-space identification in Eq. (7) set $\mathscr{V}_\mathscr{C} = \mathscr{V}_\wp|_{s=1}$. Then one has

$$\mathscr{V}_\wp = \otimes_{\sigma=1}^{s} \mathscr{V}_\mathscr{C}. \tag{8}$$

The tensor product description of $\mathscr{V}_\wp$ in Eq. (7) allows for an elegant way to analyze the mutation operator (see Proposition 2.2.2.2 and work by Griffiths and Taveré [27]), the crossover operator for regular crossover (see Eqs. (18), (20) and (22)) and gene-lottery crossover (see Eq. (25)). This is useful for computation of spectra and in, e.g., verifying that mutation commutes with some of the crossover operators mentioned above.

Let $\mathscr{U} \subset \mathscr{V}_\wp$ be the free vector space over all populations which are uniform, i.e., which consist of s copies of a single creature. Consequently, $\wp \cap \mathscr{U}$ equals[6] the set of uniform populations. In addition, $P_\mathscr{U}$ shall denote the orthogonal projection onto $\mathscr{U}$.

[6] By the construction of $\mathscr{V}_\wp$ as free vector space over $\wp$, the latter becomes the set of base vectors in $\mathscr{V}_\wp$ and thus $\wp \subset \mathscr{V}_\wp$. Hence, the above set-intersection is well defined and yields the set (of base vectors) of uniform populations.

Lemma 1.6.1. *Let $X : \mathscr{V}_\wp \to \mathscr{V}_\wp$ be a linear map such that $Xp = p$ for every $p \in \wp \cap \mathscr{U}$. Then X satisfies $XP_\mathscr{U} = P_\mathscr{U}$ and $(\mathbf{1} - P_\mathscr{U})X = (\mathbf{1} - P_\mathscr{U})X(\mathbf{1} - P_\mathscr{U})$.*

Proof. We have $XP_\mathscr{U} p = Xp = p = P_\mathscr{U} p$ for $p \in \wp \cap \mathscr{U}$ and $XP_\mathscr{U} p = 0 = P_\mathscr{U} p$ for $p \in \wp \setminus \mathscr{U}$. Hence, $XP_\mathscr{U} = P_\mathscr{U}$. In addition, we have $(\mathbf{1} - P_\mathscr{U})X = (\mathbf{1} - P_\mathscr{U})XP_\mathscr{U} + (\mathbf{1} - P_\mathscr{U})X(\mathbf{1} - P_\mathscr{U}) = (\mathbf{1} - P_\mathscr{U})P_\mathscr{U} + (\mathbf{1} - P_\mathscr{U})X(\mathbf{1} - P_\mathscr{U}) = (\mathbf{1} - P_\mathscr{U})X(\mathbf{1} - P_\mathscr{U})$. □

Let $\mathscr{S}_\wp \subset \mathscr{V}_\wp$ be the set of probability distributions over $\wp$. $\mathscr{S}_\wp$ is the relevant state space in this investigation where the stochastic matrices representing the individual steps of the probabilistic algorithm act by matrix multiplication from the left.

1.7. Gene frequency vectors

The map GFV_u as defined below measures allele/gene frequencies within populations. For $p = (c_1, c_2, \ldots, c_s) \in \wp$, $c_\sigma \in \mathscr{C}$, $1 \leqslant \sigma \leqslant s$, let

$$\mathrm{GFV}_u(p) = s^{-1} \sum_{\sigma=1}^{s} c_\sigma \in \bigoplus_{\lambda=1}^{\ell} \mathscr{V}_1. \tag{9}$$

We shall use GFV_u in this exposition as a means to analyze scaled gene-lottery crossover (see Section 2.5). Defining a mean-value in the ℓ-dimensional, positive unit-cube for binary genetic algorithms, GFV_u was introduced in [62, Section 1.3] under the terminology $\mathrm{mean}_\mathbf{u}$. The latter notation was kept for $\alpha \geqslant 2$ in [56][7].

GFV_u determines invariant subspaces of $\mathscr{V}_\wp$ in regard to certain crossover operations. More precisely, GFV_u pertains to what is called unrestricted crossover in [52, Section 2.2, p. 117 ff.; 56, Section 5.3] and to regular crossover in the multi-set representation as outlined in [56, Section 5.2.2]. GFV_u is a very useful tool in analyzing the interplay crossover vs. mutation. In that regard, the reader is referred to the results [56, Propositions 3.5, 3.8, Section 5.2.1.3, Lemma 5.1, Section 5.3.1.3, Section 5.4, Theorem 6.1].

2. The genetic operators

2.1. The spot mutation matrix

In this section, let $\mu_0 \in [0, 1]$ be a fixed parameter. Note however, that in the subsequent sections of this exposition we shall always assume $\mu_0 > 0$. The spot mutation matrix $\mathbf{m}_{\mu_0}^{(1)} \in \mathbf{M}_\alpha$ models *change* within the alphabet $\mathscr{A}$ at a single spot $\hat{\lambda}$ in the combined genome of a population, $1 \leqslant \hat{\lambda} \leqslant L$. We model local change determined by $\mathbf{m}_{\mu_0}^{(1)}$ as a continuous function of μ_0 such that $\mu_0 = 0$ corresponds to uniform change within a preferred "small neighborhood" $\mathscr{N}_n(\imath)$ of the current letter $a(\imath)$ at spot $\hat{\lambda}$ in the genome, while $\mu_0 = 1$ corresponds to change within the alphabet $\mathscr{A}$ which is spread out uniformly. Thus, we model change on the alphabet level determined by $\mathbf{m}_{\mu_0}^{(1)}$ as a scalable

[7] Ref. [56, p. 16, line 4] contains a typographical error: the tensor-symbol should be replaced by a direct-sum symbol, cf. [55, p. 13].

compromise between a neighborhood-based hill-climbing strategy in the spirit of the simulated annealing algorithm [1] and pure random change as essentially advocated in Rudolph's work [51, p. 140]. In fact, Rudolph argues for a mutation operation that is determined by a maximal entropy distribution.

In general, we can model mutation on the level of a single spot in the genome of a creature by any stochastic matrix $\mathbf{m}_{\mu_0}^{(1)} \in \mathbf{M}_\alpha$ with zero entries on the diagonal. Thus, for $\alpha = 2$ the spot mutation matrix $\mathbf{m}_{\mu_0}^{(1)}$ models flipping a single bit in the genome and we have

$$\mathbf{m}_{\mu_0}^{(1)} = \mathbf{f} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \in \mathbf{M}_2. \tag{10}$$

Next, we define a spot mutation matrix $\mathbf{m}_1^{(1)}$ for $\alpha > 1$ that models change which is spread out as uniformly as possible within $\mathscr{A}$. In fact, we set

$$\mathbf{m}_1^{(1)} = (1-\alpha)^{-1}(\mathbf{1} - P_e) + P_e. \tag{11}$$

This description of $\mathbf{m}_1^{(1)}$ shows that its spectrum is given by $\{(1-\alpha)^{-1}, 1\}$. Furthermore, $\mathbf{m}_1^{(1)}$ commutes with every symmetric stochastic matrix since P_e does so.

In order to define the localized portion of the spot mutation operator, let $\mathbf{u}$ denote the unitary, stochastic matrix describing the cyclic shift, i.e.,

$$\langle a(\imath+1), \mathbf{u}a(\imath)\rangle = 1, \quad \imath \in \mathbb{Z}_\alpha. \tag{12}$$

Observe that $\mathbf{m}_1^{(1)}$ is a linear combination of powers of $\mathbf{u}$. The spot mutation matrix $\mathbf{m}_0^{(1)}$ is defined in terms of $\mathbf{u}$ as follows:

$$\mathbf{m}_0^{(1)} = (2n)^{-1} \sum_{\nu=1}^{n} (\mathbf{u}^\nu + \mathbf{u}^{-\nu}), \tag{13}$$

where $n < \alpha/2$ determines the size of the local neighborhood of a letter in the underlying alphabet $\mathscr{A}$ as in Section 1.5. The matrix $\mathbf{m}_0^{(1)}$ describes switching to the $2n$ close neighbors of a letter $a(\imath)$ with equal probability. If n is small compared with α, then $\mathbf{m}_0^{(1)}$ represents a neighborhood-based search in the spirit of the simulated annealing algorithm [1]. The spectrum of $\mathbf{m}_0^{(1)}$ and, consequently, the spectrum of $\mathbf{m}_{\mu_0}^{(1)}$ as in definition (14) and the spectrum of the multiple-spot mutation operator $M_{\mu_0,\mu}^{(m)}$ given by Definition 2.2.1 can be computed by straightforward spectral calculus from $\mathrm{sp}(\mathbf{u}) = \{\exp(2\pi i\imath/\alpha)\colon 0 \leqslant \imath < \alpha\}$. One may apply, e.g., [49, Theorems 10.28, 11.23; 62, p. 105]. The reader may carry this out along the lines of [56, Propositions 3.3.3, 3.6.3] where explicit similar computations are listed. As shown in [56, Theorem 6.1], such computation of spectra has implications for estimates of contraction/mixing properties of the combined crossover-mutation operator in a genetic algorithm. This also effectively generalizes Koehler's Theorem [31, p. 419].

Finally, using definitions (11) and (13), we set

$$\mathbf{m}_{\mu_0}^{(1)} = (1-\mu_0)\mathbf{m}_0^{(1)} + \mu_0 \mathbf{m}_1^{(1)}, \quad \mu_0 \in (0,1). \tag{14}$$

For $\alpha = 2, 3$, we have $\mathbf{m}_{\mu_0}^{(1)} = \mathbf{m}_0^{(1)} = \mathbf{m}_1^{(1)}$. Otherwise, we have the following direct consequences of definition (14).

Lemma 2.1.2. *Let* $a(\imath), a(\imath') \in \mathscr{A}$ *such that* $\imath \neq \imath'$, $0 \leqslant \imath, \imath' \leqslant \alpha - 1$. *Let* $n \in \mathbf{N}$ *such that* $n < \alpha/2$. *Then we have*

1. $\langle a(\imath), \mathbf{m}_{\mu_0}^{(1)} a(\imath) \rangle = 0$,
2. *if* $d_{\mathscr{A}}(a(\imath), a(\imath')) \leqslant n$, *then* $\langle a(\imath'), \mathbf{m}_{\mu_0}^{(1)} a(\imath) \rangle = (1 - \mu_0)/(2n) + \mu_0/(\alpha - 1)$,
3. *if* $d_{\mathscr{A}}(a(\imath), a(\imath')) > n$, *then* $\langle a(\imath'), \mathbf{m}_{\mu_0}^{(1)} a(\imath) \rangle = \mu_0/(\alpha - 1)$.

In definition (12) and, consequently, in (13) and (14), we have identified the alphabet $\mathscr{A}$ with $\mathbb{Z}_\alpha$ for reason of mathematical convenience such that, in particular, the first and last letter of the alphabet become close neighbors. In applications of genetic algorithms to optimization problems, one may be interested in considering **R**-valued parameters or coefficients as entries in the genome of creatures. If a regular programming language such as Fortran or C is employed for the implementation of the genetic algorithm, then only a finite set $R_0 \subset \mathbf{R}$ of real numbers is used. In many cases, the search space can be restricted further by a rough analysis of the given optimization problem to a finite interval $\mathscr{A} = \{a(\imath) = x_0 + \imath\delta \colon 0 \leqslant \imath < \alpha\} \subset R_0$, $x_0, \delta \in \mathbf{R}$. Such an approach is formulated, e.g., in work by Markus, Renner, & Vanza [40, p. 48] and in [58, p. 16]. See also work by Nomura and Shimohara [45]. In this situation, the above identification $\mathscr{A} \equiv \mathbb{Z}_\alpha$ may be seen as an artificial structure. However, if we assume that the maxima of the fitness function occur for parameter values (i.e., letters) well within the interior of $\mathscr{A}$ and the fitness function assumes relatively low values at the boundary, then it is not significant that mutation is defined cyclically symmetric since the selection mechanism will force the algorithm away from the boundary of the domain of definition. Consequently, the analysis presented here can be applied to a large class of optimization problems where **R**-valued parameters are optimized in a compact domain. Alternatively, one may employ an asymmetric spot mutation matrix $\mathbf{m}_0^{(1)}$ which implements change within "one-sided" neighborhoods for parameter values (i.e., letters) close to the boundaries x_0 and $x_0 + (\alpha - 1)\delta$ of an alphabet $\mathscr{A}$ as above. In such a setting, the main results of this exposition, Theorem 3.3.2 and its Corollaries, stay valid. In the situation of a particular asymmetric spot mutation matrix, one has (as single most important change) to adapt the mutation flow inequality obtained in Proposition 2.2.3. See [57, Proposition 3.1.1] for a general type of mutation flow inequality that should cover most conceivable cases.

2.2. Multiple-spot mutation

Multiple-spot mutation $M_{\mu_0,\mu}^{(m)}$ has been studied theoretically by many authors. Earlier references include the work of Davis and Principe [17,18], Vose and Liepins [69] and Nix and Vose [44]. We continue the analysis in [62, Section 2.1, p. 110 *ff*., "multiple-bit mutation"; 56, Section 3.3]. However, our discussion here will be limited to the absolute minimum. Multiple-spot mutation is the most commonly used procedure for mutation in implementations of genetic algorithms.

Definition 2.2.1 (multiple-spot mutation $M^{(m)}_{\mu_0,\mu}$). In what follows, suppose that the spot mutation matrix $\mathbf{m}^{(1)}_{\mu_0}$ is given as in definition (14) with $\mu_0 \in (0,1]$. Let $\mu \in \left(0, \frac{1}{2}\right)$ denote the mutation rate, and for $\hat{\lambda} = 1 \cdots L$ execute the following two steps.

Step 1: Decide probabilistically whether or not to change the letter at spot $\hat{\lambda}$ in the current population. The decision for change is made positively with probability μ.

Step 2: If the decision has been made positively in step 1; then the letter at spot $\hat{\lambda}$ is altered in accordance with the transition probabilities for letters set by $\mathbf{m}^{(1)}_{\mu_0}$.

In this work, we shall mainly be interested in genetic algorithms using small mutation rates. In order to avoid some technicalities in our presentation, we have therefore restricted the mutation rate to $\mu \in \left(0, \frac{1}{2}\right)$ in Definition 2.2.1. If necessary, the reader may adapt the results [56, Propositions 3.6, 3.7, Theorems 6.1, 6.2] to obtain some of the results listed there and in what follows here for larger μ, in particular, conditions that ensure key properties of $M^{(m)}_{\mu_0,\mu}$ such as being invertible.

Proposition 2.2.2. *Let $\mu_0 \in (0,1]$ determine the balance between neighborhood-based search and uniform change in the spot mutation matrix $\mathbf{m}^{(1)}_{\mu_0}$ as in definition (14). Let $\mu \in \left(0, \frac{1}{2}\right)$ denote the mutation rate for multiple-spot mutation $M^{(m)}_{\mu_0,\mu}$. Let $M^{(m)}_{\mu_0,\mu}$ also denote the stochastic matrix associated with multiple-spot mutation. $M^{(m)}_{\mu_0,\mu}$ acts on $\mathscr{V}_{\wp}$ and describes transition probabilities for entire populations. Then we have*

1. *Let $p, q \in \wp$. The coefficients of $M^{(m)}_{\mu_0,\mu}$ are given as follows*:

$$\langle q, M^{(m)}_{\mu_0,\mu} p\rangle = \mu^{\Delta(p,q)} \cdot \left(\frac{1-\mu_0}{2n} + \frac{\mu_0}{\alpha-1}\right)^{\Delta_{\pm n}(p,q)} \cdot \left(\frac{\mu_0}{\alpha-1}\right)^{\Delta(p,q)-\Delta_{\pm n}(p,q)} \cdot (1-\mu)^{L-\Delta(p,q)} > 0.$$

   In particular, $M^{(m)}_{\mu_0,\mu}$ is a fully positive, symmetric matrix with entries bounded below by $K\mu_0^L\mu^L$ where $K \in \mathbf{R}^+_*$ is a suitable constant.
2. $M^{(m)}_{\mu_0,\mu} = \bigotimes_{\lambda=1}^{L} ((1-\mu)\,\mathbf{1} + \mu\,\mathbf{m}^{(1)}_{\mu_0})$.
3. *$M^{(m)}_{\mu_0,\mu}$ is an invertible matrix.*

Proof. Statement (1) follows directly from the Definition 2.2.1 and Lemma 2.1.1. Statement (2) follows by comparing statement (1) and the action of the matrix listed on the right-hand side of the formula. Finally, we show statement (3). $\mathbf{m}^{(1)}_{\mu_0}$ is a stochastic matrix and, consequently, has operator norm 1 by [53, p. 5, Eq. (7')]. By Rudin [49, Theorem 10.13.2], we conclude that the spectrum of $\mathbf{m}^{(1)}_{\mu_0}$ is contained in the closed unit disk. Elementary spectral calculus shows now that the spectrum of $(1-\mu)\mathbf{1}+\mu\mathbf{m}^{(1)}_{\mu_0}$ is contained in the open right half-plane of $\mathbf{C}$. In particular, $(1-\mu)\mathbf{1}+\mu\mathbf{m}^{(1)}_{\mu_0}$ is invertible. Thus, $M^{(m)}_{\mu_0,\mu}$ is invertible as a tensor product of invertible matrices. □

Observe that for an asymmetric spot mutation matrix which drives the algorithm away from the boundaries $a(0)$ and $a(\alpha-1)$ of the underlying alphabet as discussed

at the end of Section 2.1, the results obtained in Proposition 2.2.2.2.-3 stay valid. A tensor-product description of mutation is known in Theoretical Biology. See, e.g., work by Griffiths and Taveré [27]. The next result is one of the key ingredients in showing convergence to uniform populations by properly scaled genetic algorithms. See [57, Propositions 3.1.1] for a generalization.

Proposition 2.2.3 (mutation flow inequality). *Let $\mu_0 \in (0,1]$ determine the balance between neighborhood-based search and uniform change in the spot mutation matrix $\mathbf{m}_{\mu_0}^{(1)}$ as in definition (14). Let $\mu \in \left(0, \frac{1}{2}\right)$ denote the mutation rate for multiple-spot mutation $M_{\mu_0,\mu}^{(m)}$. As throughout this exposition let $n < \alpha/2$ denote the size of local neighborhoods as in Section 1.5 and let*

$$\beta = \left((1-\mu)^s + \mu^s \cdot \left(2n\left(\frac{1-\mu_0}{2n} + \frac{\mu_0}{\alpha-1}\right)^s + (\alpha - 2n - 1)\left(\frac{\mu_0}{\alpha-1}\right)^s\right)\right)^\ell.$$

Then $\beta \in (0,1)$, and we have for $v \in \mathscr{S}_\wp$

$$\|(\mathbf{1} - P_{\mathscr{U}})M_{\mu_0,\mu}^{(m)} v\|_1 \leqslant 1 - \beta + \beta\|(\mathbf{1} - P_{\mathscr{U}})v\|_1.$$

Proof. Let $p \in \wp$ be uniform. In order to produce a uniform population from p under multiple-spot mutation, one selects λ spots to be changed in the first creature of p and then has to change $s \cdot \lambda$ corresponding spots in p where $0 \leqslant \lambda \leqslant \ell$. From the λ spots in the first creature selected for change, one selects v spots that are changed to close neighbors in the sense of Section 1.5 where $0 \leqslant v \leqslant \lambda$. For the close neighbors, one has $(2n)^v$ distinct choices. For the remaining spots to be changed, one has $(\alpha - 2n - 1)^{\lambda - v}$ distinct choices. Thus, the *probability* of producing a uniform population from p via $M_{\mu_0,\mu}^{(m)}$ is given by

$$\sum_{\lambda=0}^{\ell} \binom{\ell}{\lambda} (1-\mu)^{s(\ell-\lambda)} \mu^{s\lambda} \cdot \left(\sum_{v=0}^{\lambda} \binom{\lambda}{v} (2n)^v \left(\frac{1-\mu_0}{2n} + \frac{\mu_0}{\alpha-1}\right)^{sv} \times (\alpha - 2n - 1)^{\lambda - v} \left(\frac{\mu_0}{\alpha-1}\right)^{s(\lambda-v)}\right) = \beta \in [0,1].$$

Clearly $0 < \beta$. Since for $\mu > 0$ there is also a positive probability for generating a non-uniform population from p, we have $\beta < 1$.

The probability for producing a non-uniform population from p via $M_{\mu_0,\mu}^{(m)}$ is given by $1 - \beta$. Let $v \in \mathscr{S}_\wp$. Then we obtain

$$\begin{aligned}
&\|(\mathbf{1} - P_{\mathscr{U}})M_{\mu_0,\mu}^{(m)} v\|_1 \\
&\quad = \sum_{q \in \wp \setminus \mathscr{U}} \sum_{p \in \wp} \langle q, M_{\mu_0,\mu}^{(m)} p\rangle \langle p, v\rangle \\
&\quad = \sum_{p \in \wp \cap \mathscr{U}} \|(\mathbf{1} - P_{\mathscr{U}})M_{\mu_0,\mu}^{(m)} p\|_1 \langle p, v\rangle + \sum_{p \in \wp \setminus \mathscr{U}} \|(\mathbf{1} - P_{\mathscr{U}})M_{\mu_0,\mu}^{(m)} p\|_1 \langle p, v\rangle
\end{aligned}$$

$$\leqslant \sum_{p\in\wp\cap\mathscr{U}} (1-\beta)\langle p,v\rangle + \sum_{p\in\wp\setminus\mathscr{U}} \langle p,v\rangle$$
$$= (1-\|(\mathbf{1}-P_{\mathscr{U}})v\|_1)(1-\beta) + \|(\mathbf{1}-P_{\mathscr{U}})v\|_1$$
$$= 1-\beta+\beta\|(\mathbf{1}-P_{\mathscr{U}})v\|_1. \qquad \square$$

2.3. Rational generalized crossover

In order to handle the crossover operation C for the purpose of obtaining a convergent, scaled genetic algorithm, we can essentially adopt the definition of a continuous, generalized crossover operation given in [61, Section 4.1; 56, Section 5.1.1]. Note however, that we *drop the condition* that the crossover operation is given by a symmetric matrix. The condition of a symmetric crossover matrix is used in [56] only in the proof of Theorem 8.2.2, formula (30)]. We shall illustrate in Section 4.1 that the proof of [56, Theorem 8.2.2] essentially stays valid without this condition. Thus, the results in [56], in particular, those on genetic drift and positive limit-mutation rate [56, Sections 7.5, 8.1] stay valid for scaled genetic algorithms that use not-necessarily symmetric, continuous generalized crossover operators. See Section 4.1 for more details in this regard.

Definition 2.3.1 (rational generalized crossover). For crossover rate $\chi\in[0,1]$, we suppose that $C=C(\chi)$ is not a necessarily symmetric, but stochastic matrix that acts on $\mathscr{V}_\wp$ via the identifications in Section 1.6 and describes transition probabilities for entire populations. In addition, $C(\chi)$ satisfies the following conditions:

1. $\chi\mapsto\langle q,C(\chi)p\rangle$ is a rational function in χ for $p,\ q\in\wp$ which obviously has no singularity in $[0,1]$,
2. $C(0)=\mathbf{1}$,
3. $C(\chi)p=p$ for $p\in\wp\cap\mathscr{U}$.

Clearly, $C(\chi)^k$ is also a rational generalized crossover operation, $k\in\mathbf{N}$. Combining parts 1 and 2 of Definition 2.3.1, we can write $C(\chi)$ in the following way:

$$C(\chi)=\mathbf{1}+\chi C_0(\chi), \tag{15}$$

where $C_0(\chi)$ is a matrix acting on $\mathscr{V}_\wp$ with bounded, rational entries in χ. In addition, we have the following result.

Lemma 2.3.2. *Let $\chi\mapsto C(\chi)$ be a rational generalized crossover as in Definition* 2.3.1. *Then there exists $\chi_0\in(0,1]$ such that*

1. $\langle p,C(\chi)p\rangle\geqslant\frac{1}{2}$ *for* $p\in\wp$, $\chi\in[0,\chi_0]$, *and*
2. $C(\chi)$ *is invertible for* $\chi\in[0,\chi_0]$.

Proof. Let $p\in\wp$. We have $\lim_{\chi\to0}\langle p,C(\chi)p\rangle=\langle p,p\rangle=\mathbf{1}$. Thus, there exists $\chi_p\in(0,1]$ such that $\langle p,C(\chi)p\rangle\geqslant\frac{1}{2}$ for $\chi\in[0,\chi_p]$. The determinant is a continuous function

in its argument. Hence, $\lim_{\chi\to 0}\det(C(\chi))=\det(\mathbf{1})=1$. Thus, there exists $\chi_1\in(0,1]$ such that $\det(C(\chi))\neq 0$ for $\chi\in[0,\chi_1]$. Set $\chi_0=\min\{\chi_1,\chi_p:\ p\in\wp\}>0$. □

In order to illustrate Definition 2.3.1, let us consider the following two examples. Section 2.4 presents a new, simplified model for one-cutpoint regular crossover which allows for an improved version of the analysis in [62, Section 2.2]. As a by-product of the analysis, we obtain simple descriptions of two-cutpoint regular crossover and uniform regular crossover with respect to the tensor-space description of $\mathscr{V}_\wp$ as in Eq. (7). These simple descriptions show, in particular, that these regular crossover operators commute with mutation. Section 2.5 shows that gene-lottery crossover in two scaled versions also fits the requirements for a generalized crossover operation listed in Definition 2.3.1 above. Gene-lottery crossover operators *do not commute* with mutation. However, their use yields a convergent, scaled genetic algorithm as formulated in Corollary 3.3.4 where the crossover rate can stay larger than any power of the mutation rate over the entire course of the algorithm. In fact, the discussion in Sections 2.4 and 2.5 shall yield significant improvements of the main result of this exposition (Theorem 3.3.2) in Corollary 3.3.3 (for regular crossover) and Corollary 3.3.4 (for gene-lottery crossover). In all examples introduced in Sections 2.4 and 2.5, the functions $\chi\mapsto\langle q,C(\chi)p\rangle$, $p,\ q\in\wp$, are actually polynomials.

Variations of the crossover operator for $\alpha=2^k$, $k\in\mathbf{N}$, that fit Definition 2.3.1 have been previously studied by several authors, e.g., Bhattacharyya and Koehler [7] as well as Leung et al. [35]. See also work by Koehler et al. [32] in this regard.

2.4. Regular crossover

Recall that the size $s\geqslant 4$ of populations is supposed to be an even integer. Regular crossover shall refer to a procedure where the creatures $c_1,\ldots,c_s$ in the population are sequentially paired, and a specific crossover operation is then applied to each of the pairs $(c_1,c_2),\ldots,(c_{s-1},c_s)$ with probability χ. This follows, e.g., Goldberg's approach [23, pp. 16–17].

One-cutpoint regular crossover has previously been studied in the tensor-string representation for populations in [62, Section 2.2, "simple crossover"; 56, Section 5.2]. We first define the elementary one-cutpoint crossover operation $C(\sigma;\lambda)$ for $1\leqslant\sigma\leqslant s/2$ and $1\leqslant\lambda\leqslant\ell$. $C(\sigma;\lambda)$ exchanges "heads" (leading letters in spots before and including spot λ) of creatures $c_{2\sigma-1}$ and $c_{2\sigma}$ in the current population. The case $\lambda=\ell$ is included in order to present an analysis that completely covers the discussion in [62, Section 2.2] and for mathematical convenience as discussed there (see [62, p. 113; footnote, Proposition 7.6]). We shall also treat the case $1\leqslant\lambda<\ell$ in what follows below.

Definition 2.4.1 (elementary one-cutpoint crossover $C(\sigma;\lambda)$). Let $1\leqslant\sigma\leqslant s/2$ and $0\leqslant\lambda\leqslant\ell$. Let $p=(c_1,\ldots,c_s)\in\wp$ be the current population, $c_{\sigma'}\in\mathscr{C}$, $1\leqslant\sigma'\leqslant s$. Then the elementary one-cutpoint crossover operation $C(\sigma;\lambda)$ is defined by the following three steps.

Step 1: Pick creatures $c_{2\sigma-1}=(a(\imath_1),\ldots,a(\imath_\ell))$ and $c_{2\sigma}=(a(\imath'_1),\ldots,a(\imath'_\ell))$ from p where $a(\imath_\lambda),a(\imath'_\lambda)\in\mathscr{A}$, $1\leqslant\lambda\leqslant\ell$.

Step 2: For $v=1,\ldots,\ell$ do: ((If $v\leqslant\lambda$, then switch letters by setting $\bar{a}_v=a(\iota'_v)$ and $\bar{a}'_v=a(\iota_v)$. If $v>\lambda$, then copy letters by setting $\bar{a}_v=a(\iota_v)$ and $\bar{a}'_v=a(\iota'_v)$.))

Step 3: Replace $c_{2\sigma-1}$ by $(\bar{a}_1,\ldots,\bar{a}_\ell)$ and replace $c_{2\sigma}$ by $(\bar{a}'_1,\ldots,\bar{a}'_\ell)$ in p.

$C(\sigma;0)$ is the identity operation. We shall also denote the symmetric, stochastic matrix associated with the elementary one-cutpoint crossover operation by $C(\sigma;\lambda)$. The matrix $C(\sigma;\lambda)$ acts on $\mathscr{V}_\wp$ and describes transition probabilities for entire populations. $C(\sigma;\lambda)$ commutes with $M^{(m)}_{\mu_0,\mu}$ since $C(\sigma;\lambda)$ moves letters around but does not alter them; and it does not matter whether the entire collection of letters in a population is mutated spot-wise before or after being rearranged. Consequently, *all* crossover operators considered in Section 2.4 commute with $M^{(m)}_{\mu_0,\mu}$. Similar considerations can be made for a single-spot mutation operation that uses $\mathbf{m}^{(1)}_{\mu_0}$ similar to the single-spot mutation considered in [56, Section 3.2].

Clearly, one has $C(\sigma;\lambda)p=p$ for $p\in\wp\cap\mathscr{U}$. One also has $C(\sigma;\lambda)^2=\mathbf{1}$ which shows that—up to a rearrangement of the basis of $\mathscr{V}_\wp$—$C(\sigma;\lambda)$ is a block diagonal matrix consisting of $\mathbf{1}$ of proper dimension and flip matrices $\mathbf{f}$ as defined in Eq. (10). See Section 5.4 for examples. This shows $\mathrm{sp}(C(\sigma;\lambda))=\{-1,1\}$. One then randomizes the choice of the cut-cutpoint λ giving every possible value for λ equal probability. This yields the *averaged one-cutpoint crossover operation* $\bar{C}(\sigma)$ which is given by the following symmetric, stochastic matrix:

$$\bar{C}(\sigma)=\ell_0^{-1}\sum_{\lambda=1}^{\ell_0}C(\sigma;\lambda). \tag{16}$$

In Eq. (16), one has $\ell_0=\ell$, if a "cutpoint" $\lambda=\ell$ is permitted and $\ell_0=\ell-1$ otherwise. The spectrum of $\bar{C}(\sigma)$ satisfies

$$\mathrm{sp}(\bar{C}(\sigma))\subset\{-1\}\cup[-1+2/\ell_0,1-2/\ell_0]\cup\{1\}. \tag{17}$$

This follows from the fact that $\bar{C}(\sigma)$ is a convex combination of commuting matrices $C(\sigma;\lambda)$ and [49, Theorem 11.23]. For $p\in\wp\cap\mathscr{U}$, one obtains as immediate consequence of $C(\sigma;\lambda)p=p$ and definition (16) that $\bar{C}(\sigma)p=p$.

Example. Let $p=((a(0),a(0),\ldots,a(0)),(a(1),a(0),\ldots,a(0)))$ where the letter $a(1)$ occurs at spot $\ell+1$ of p, i.e., the first spot of the second creature. Let $q=((a(1),a(0),a(0),\ldots,a(0)))\in\wp$ and $x=p-q=1\cdot p+(-1)\cdot q\in\mathscr{V}_\wp$. Then $C(1;\lambda)p=q$ and $C(1;\lambda)q=p$ for every $\lambda,1\leqslant\lambda\leqslant\ell$. Hence, $\bar{C}(1)x=-x$ and, consequently, $-1\in\mathrm{sp}(\bar{C}(1))$. However, this example for an eigenvector to eigenvalue -1 will disappear in the multi-set representation for populations since then p and q represent the same population. (See also [56, Theorem 6.2]).

Finally, we have:

Definition 2.4.2 (one-cutpoint regular crossover $C^{(1)}_{\mathrm{reg}}$). Let $\chi\in[0,1]$ denote the crossover rate. For $\sigma=1\cdots s/2$ do the next two steps.

Step 1: Decide probabilistically whether or not crossover takes place in the current population involving parent creatures $c_{2\sigma-1}$ and $c_{2\sigma}$. The decision for crossover to take place is made positively with probability χ.

Step 2: If the decision for crossover involving creatures $c_{2\sigma-1}$ and $c_{2\sigma}$ has been made positively in step 1, then execute $\bar{C}(\sigma)$.

The case of a negative decision in step 1 above is referred to as *cloning* in [68, p. 43]. Suppose that the tensor-factors $\mathscr{V}_1$ in Eq. (7) are canonically arranged corresponding to sequentially listing creatures as in Eq. (8), i.e., tensor-factor $\hat{\lambda} = (\sigma-1)\ell+\lambda$ corresponds to spot λ in creature c_σ in a population $p=(c_1,c_2,\ldots,c_s)$, $c_\sigma \in \mathscr{C}$, $1 \leqslant \sigma \leqslant s$. The symmetric, stochastic matrix $C^{(1)}_{\mathrm{reg}}(\chi)$ describing transition probabilities for entire populations under one-cutpoint regular crossover is then given by the following expression:

$$C^{(1)}_{\mathrm{reg}}(\chi) = \prod_{\sigma=1}^{s/2} ((1-\chi)\mathbf{1} + \chi\bar{C}(\sigma)) = \bigotimes_{\sigma=1}^{s/2} ((1-\chi)\mathbf{1} + \chi\bar{C}(1)|_{s=2}). \tag{18}$$

A direct verification based upon Eq. (18) shows that $C^{(1)}_{\mathrm{reg}}(\chi)$ satisfies the requirements for a rational generalized crossover operation set in Section 2.3. Using Eq. (17) and [49, Theorem 11.23], one obtains (for sufficiently small χ) a better estimate $1-2\chi/\ell_0$ for the second largest modulus of elements in the spectrum of $C^{(1)}_{\mathrm{reg}}(\chi)$ than [62, Proposition 7.7]. This yields improvements of results in [62, Proposition 10; 58, Theorem 6.1].

Koehler's Theorem [31, p. 419] proves the Vose–Liepins conjecture by determining the spectrum of the crossover-mutation matrix in the Vose–Liepins model for binary genetic algorithms [69,68]. Up to a leading factor $\frac{1}{2}$, the factors $(1-2\mu)^k$ contributed by the mutation matrix coincide in the Vose–Liepins model and the model based upon the tensor-string representation for populations used here (cf. [62, Proposition 3.4]). The improved estimate $1-2\chi/\ell_0$ obtained above corresponds to (but does not equal) the factor $1-\chi/(\gamma-1)$ contributed by crossover in the third largest eigenvalue obtained in Koehler's Theorem. The reason for the factors $\frac{1}{2}$ will become apparent in Section 4.3. See [56, Theorem 6.2] for a related result where the spectrum of the combined crossover-mutation matrix is considered acting on the multi-set representation for populations (as a projection of the crossover-mutation matrix in the tensor-string representation for populations).

Next, let us sketch a model for two-cutpoint regular crossover $C^{(2)}_{\mathrm{reg}}$. Assume first that $s=2$. Then two-cutpoint regular crossover $C^{(2)}_{\mathrm{reg}}|_{s=2}$ can be described as follows:

$$C^{(2)}_{\mathrm{reg}}|_{s=2}(\chi) = (1-\chi)\mathbf{1} + \chi\,\frac{\ell}{\ell-1}\cdot(\bar{C}(1)^2 - \ell^{-1}\mathbf{1}) \tag{19}$$

with $\ell_0=\ell$ in Eq. (16). Suppose that the tensor-factors $\mathscr{V}_1$ in Eq. (7) are canonically arranged as for Eq. (18). Then we have

$$C^{(2)}_{\mathrm{reg}}(\chi) = \bigotimes_{\sigma=1}^{s/2} C^{(2)}_{\mathrm{reg}}|_{s=2}(\chi). \tag{20}$$

Here, we suppose that a probabilistic decision to apply two-cutpoint regular crossover is made separately for each of the pairs $(c_1, c_2)\cdots(c_{s-1}, c_s)$.

In contrast to one-cutpoint regular crossover which is often used in implementations of genetic algorithms, two-cutpoint regular crossover $C_{\mathrm{reg}}^{(2)}$ has the advantage of being "cyclicly symmetric": the genome of a creature can be seen bent to a circle; corresponding pieces of the genome are exchanged between creatures during crossover; but in regard to this operation there are no preferred head and tail which are usually "separated" in a non-trivial elementary one-cutpoint crossover operation. In other words, there is no linkage "directed towards the head of creatures" in the sense of Geiringer [21, p. 33] as discussed in [56, p. 26: footnote 4, Section 5.4]. See also related work by Vose and Wright [71].

Finally, let us sketch a model for uniform regular crossover $C_{\mathrm{reg}}^{(u)}$. Uniform regular crossover switches letters of the parents creatures $c_{2\sigma-1}$ and $c_{2\sigma}$ at corresponding spots λ, $1 \leqslant \lambda \leqslant \ell$, with probability $\frac{1}{2}$. Uniform regular crossover has been discussed, e.g., by Nomura and Shimohara [45] and Vose [68, p. 43]. Assume first that $s = 2$. Then uniform regular crossover $C_{\mathrm{reg}}^{(u)}|_{s=2}$ can be described as follows:

$$C_{\mathrm{reg}}^{(u)}|_{s=2}(\chi) = (1-\chi)\mathbf{1} + 2^{-\ell}\chi \prod_{\lambda=1}^{\ell} (\mathbf{1} + C(1;\lambda)C(1;\lambda-1)). \tag{21}$$

Suppose that the tensor-factors $\mathscr{V}_1$ in Eq. (7) are canonically arranged as for Eq. (18). Then we have

$$C_{\mathrm{reg}}^{(u)}(\chi) = \bigotimes_{\sigma=1}^{s/2} C_{\mathrm{reg}}^{(u)}|_{s=2}(\chi). \tag{22}$$

We suppose again that a probabilistic decision to apply uniform regular crossover is made separately for each of the pairs $(c_1, c_2)\cdots(c_{s-1}, c_s)$.

The descriptions in Eqs. (20) and (22) allow for verification of commutation relations involving single/multiple spot mutation and two-cutpoint/uniform regular crossover based upon the discussion for $C(\sigma;\lambda)$ after Definition 2.4.1. Such results have been obtained for one-cutpoint regular crossover in Proposition 2.2.2.2, [62, Lemma 5.9, Proposition 7.1.4; 56, Section 5.2.1.2]. This in turn allows for analysis of spectral properties and contracting properties of two-cutpoint/uniform regular crossover with respect to the Euclidean norm using the Spectral Mapping Theorem [49, Theorem 10.28]. See [52, Propositions 7, 10], [56, Theorems 6.1, 6.2] for applicable results related to these aspects.

2.5. Gene-lottery crossover

Gene-lottery crossover is a variant of the crossover operation which accelerates convergence to uniform populations by selecting letters (genes/alleles) probabilistically from the distribution of letters at corresponding spots in the creatures of the current population.

Definition 2.5.1 (scaled, spot-wise gene-lottery crossover $C^{(m)}_{\mathrm{glc}}$). Let $\chi \in [0,1]$ denote the crossover rate. Let $p=(c_1,c_2,\ldots,c_s)$ be the current population, $c_\sigma \in \mathscr{C}$, $1 \leqslant \sigma \leqslant s$. Let $\xi=(\xi_\lambda)_{\lambda=1}^{\ell}=\mathrm{GFV}_u(p)$. For $1 \leqslant \lambda \leqslant \ell$, let

$$\xi_\lambda = \sum_{\iota=0}^{\alpha-1} \xi(\lambda,\iota)\cdot a(\iota) \in \mathscr{V}_1, \quad \xi(\lambda,\iota) \in [0,1]. \tag{23}$$

Now, execute for $\sigma = 1\cdots s$ and for $\lambda = 1\cdots\ell$ the following two steps.

Step 1: Decide probabilistically whether or not to apply gene-lottery crossover at spot λ in creature c_σ. The decision for application of gene-lottery crossover is made positively with probability χ.

Step 2: If the decision has been made positively in step 1, then probabilistically select a letter for that spot such that $a(\iota)\in\mathscr{A}$ has probability $\xi(\lambda,\iota)$ of being selected, $0 \leqslant \iota \leqslant \alpha-1$.

We shall denote the stochastic matrix associated with the scaled gene-lottery crossover operation by $C^{(m)}_{\mathrm{glc}}=C^{(m)}_{\mathrm{glc}}(\chi)$ as well. The matrix $C^{(m)}_{\mathrm{glc}}$ acts on $\mathscr{V}_\wp$ and describes transition probabilities for entire populations. It is immediately clear, that $C^{(m)}_{\mathrm{glc}}$ leaves uniform populations invariant. Also, $C^{(m)}_{\mathrm{glc}}(0)=\mathbf{1}$ follows directly from Definition 2.5.1.

In order to verify Definition 2.3.1.1 for $C^{(m)}_{\mathrm{glc}}(\chi)$, suppose first that $\ell=1$. Now, let $p=(a(\iota_1),a(\iota_2),\ldots,a(\iota_s))$, $q=(a(\iota'_1),a(\iota'_2),\ldots,a(\iota'_s))$ with $a(\iota_\sigma),a(\iota'_\sigma)\in\mathscr{A}=\mathscr{C}$, $1\leqslant\sigma\leqslant s$. Let $\mathrm{GFV}_u(p)=\xi=(\xi_1)$ and let ξ_1 be as in Eq. (23). Then, the stochastic matrix $C^{(m)}_{\mathrm{glc}}(\chi)|_{\ell=1}$ describing probabilistic passage from p to q under gene-lottery crossover is given by

$$\langle q,(C^{(m)}_{\mathrm{glc}}(\chi)|_{\ell=1})p\rangle = \prod_{\sigma=1}^{s}((1-\chi)\delta(a(\iota_\sigma),a(\iota'_\sigma))+\chi\xi(1,\iota'_\sigma)). \tag{24}$$

For arbitrary $\ell\in\mathbf{N}$, consider the order of tensor-factors $\mathscr{V}_1$ in Eq. (7) arranged in such a way that s-tuples of λth spots in creatures are combined, and then these s-tuples are listed sequentially for $1\leqslant\lambda\leqslant\ell$, i.e., tensor-factor $\hat{\lambda}=(\lambda-1)s+\sigma$ corresponds to spot λ in creature c_σ in a population $p=(c_1,c_2,\ldots,c_s)$, $c_\sigma\in\mathscr{C}$, $1\leqslant\sigma\leqslant s$. With this identification of $\mathscr{V}_\wp$, we have

$$C^{(m)}_{\mathrm{glc}}(\chi)=\bigotimes_{\lambda=1}^{\ell} C^{(m)}_{\mathrm{glc}}(\chi)|_{\ell=1}. \tag{25}$$

Definition 2.5.1 shows that gene-lottery crossover is a quite "destructive" procedure. For example in the case of the binary genetic algorithm, gene-lottery crossover can generate any population in $\wp$ from a population p such that $\mathrm{GFV}_u(p)$ is an interior point of the 2^ℓ-dimensional, positive $\|\cdot\|_1$-unit-sphere. Since this includes generating uniform populations, we can conclude for arbitrary size α of the underlying alphabet that $C^{(m)}_{\mathrm{glc}}(\chi)$ is not symmetric for $\chi>0$.

Definition 2.5.2 (alternate scaled gene-lottery crossover).

Step 1: Decide probabilistically whether or not to apply gene-lottery crossover globally to the current population. The decision to apply gene-lottery crossover is made positively with probability $\chi \in [0,1]$.

Step 2: If the decision has been made positively in step 1, then gene-lottery crossover corresponding to $C_{\mathrm{glc}}^{(m)}(1)$ in the sense of Eq. (25) is applied.

In case of Definition 2.5.2, the stochastic transition matrix is given by $(1-\chi)\mathbf{1} + \chi C_{\mathrm{glc}}^{(m)}(1)$ with $C_{\mathrm{glc}}^{(m)}(1)$ as in Eq. (25). The two crossover operations just considered and the population-wise single/multiple-spot mutation operators based upon $\mathbf{m}_{\mu_0}^{(1)}$, in general, do not commute. This is shown by the following and similar examples.

Example. Let $\alpha = 2$, $\ell = 1$, $s = 2$. Let $p = (a(0), a(0))$, $q = (a(1), a(0))$, $q_0 = (a(0), a(1)) \in \wp$. Then

$$\langle q, M_{\mu_0,\mu}^{(m)} C_{\mathrm{glc}}^{(m)}(1)\, p\rangle = \langle q, M_{\mu_0,\mu}^{(m)} p\rangle = \mu(1-\mu). \tag{26}$$

On the other hand, we have

$$\langle q, C_{\mathrm{glc}}^{(m)}(1) M_{\mu_0,\mu}^{(m)} p\rangle = \mu(1-\mu)(\langle q, C_{\mathrm{glc}}^{(m)}(1) q\rangle + \langle q, C_{\mathrm{glc}}^{(m)}(1) q_0\rangle) = \frac{\mu(1-\mu)}{2} \tag{27}$$

Eqs. (26) and (27) show that $M_{\mu_0,\mu}^{(m)}$ and $C_{\mathrm{glc}}^{(m)}(\chi)$ do not commute for the setting of the example, if $\chi > 0$.

For $\chi > 0$ and $p \in \wp$, there is a positive probability to generate uniform populations from p under gene-lottery crossover. Thus, $C_{\mathrm{glc}}^{(m)}(\chi)$ satisfies the following inequality:

$$\|(\mathbf{1} - P_{\mathscr{U}}) C_{\mathrm{glc}}^{(m)}(\chi) v\|_1 \leqslant \theta_c \cdot \|(\mathbf{1} - P_{\mathscr{U}}) v\|_1. \tag{28}$$

A similar estimate as inequality (28) holds, if gene-lottery crossover is given by Definition 2.5.2. Inequality (28) can be obtained as the proof of Proposition 2.6.1.3. θ_c is a function of χ and $\lim_{\chi \to 0} \theta_c(\chi) = 1$. Thus, gene-lottery crossover alone (as function of χ_t, $t \in \mathbf{N}$) fits the definition of a generalized fitness scaling given in [56, Definition 7.1]. In fact, one has $\wp^{\mathrm{I}} = \wp$, $\wp^{\mathrm{II}} = \emptyset$ and $\theta_t = \theta_c(\chi_t)$ in the sense of [56, Definition 7.1].

Inequality (28) has two major consequences: (1) The effect of genetic drift (i.e., *non-ergodic* convergence to uniform populations in a genetic algorithm without mutation) as discussed in [61, Section 6; 56, Section 7.5] is accelerated. (2) The contraction property of the combined crossover-selection operator $\mathscr{F}_t$ as discussed in Proposition 2.6.2 is enlarged by a factor $\theta_c(\chi_t)$.

Using gene-lottery crossover, e.g., in a simple genetic algorithm with extremely low mutation rate may virtually be equivalent to implementing genetic drift. If no specific precautions are taken, then an observed accelerated "convergence" to uniform

populations induced by θ_c could be very misleading in regard to the goal of rather "fully" (ergodically) exploring the search space and finding globally optimal creatures. However, use of gene-lottery crossover in a properly scaled, asymptotically converging genetic algorithm as described in Corollary 3.3.4 yields the possibility that gene-lottery crossover accelerates the accumulation of "good genes" considerably since the crossover rate can stay larger than any power of the mutation rate over the entire course of the algorithm.

The gene-lottery crossover operators considered above are "multiple-spot" operators as they are applied sequentially to every spot in the population. Another possible gene-lottery crossover operator would be a "single-spot" gene-lottery crossover operator in analogy to single-spot mutation in the sense of [56, Section 3.2].

2.6. The fitness function and selection

We shall assume that there is a given non-constant fitness function $f : D_f \to \mathbf{R}^+$ where $D_f \subset \mathscr{C} \times \wp$ is the set of all pairs (c, p) such the $c \in p$. Suppose that for every $p \in \wp$ one has $\max\{f(c, p): c \in p\} > 0$. In addition, we shall assume that a non-empty set $\mathscr{C}_{\max} \subset \mathscr{C}$ exists such that for any population $p \in \wp$, $c \in \mathscr{C} \setminus \mathscr{C}_{\max}$ and d, $d' \in \mathscr{C}_{\max}$:

$$c, d \in p \Rightarrow f(c, p) < f(d, p) \quad \text{and} \quad d, d' \in p \Rightarrow f(d, p) = f(d', p), \tag{29}$$

i.e., the elements of $\mathscr{C}_{\max}$ behave strictly superior in every population they reside in.

Typical examples for fitness functions satisfying (29) are: (1) a fitness function whose values are independent of the population and (2) rank based upon such a (raw) fitness function. In that case, $\mathscr{C}_{\max}$ is the set of creatures where the (raw) fitness function attains maximal value. See [56, Section 7.3] for a suitable definition of rank. Using a selection method based upon rank induced by a given raw fitness function was proposed by Baker [5]. Another example for a fitness function satisfying (29) arises in a coevolutionary, single-species setting (e.g., game-playing programs of finite length), if a group of strictly superior creatures (agents) exists.

The optimization algorithm is supposed to *maximize* f in the sense of finding an element of $\mathscr{C}_{\max}$. What we shall show in Theorem 3.3.2 is that a properly scaled genetic algorithm will do this asymptotically: if $w_\infty \in \mathscr{S}_\wp$ is the limit of the steady-state distributions $w_t = G_t w_t \in \mathscr{S}_\wp$ for the individual steps G_t, $t \in \mathbf{N}$, of the underlying inhomogeneous Markov chain describing the algorithm (see line (38)), then w_∞ is non-zero only over uniform populations containing members of $\mathscr{C}_{\max}$. Let $v_t = \prod_{\tau=t}^{1} G_\tau v_0$ where $v_0 \in \mathscr{S}_\wp$ is initially fixed. Asymptotically, the probability distributions v_t describing the state of the scaled genetic algorithm for individual steps $t \in \mathbf{N}$ approach the trajectory of the w_t even though thermal equilibrium (i.e., steady state $v_t = w_t$) is not necessarily reached in an individual step. These facts follow from an inspection of the proofs of [30, p. 160: Theorem V.4.3] or [57, Theorem 3.3.2]. See Theorem 3.3.2 for additional details.

Let

$$\rho_2(f) = \min\{f(c,p)/f(d,p):\ p \in \wp,\ c \in \mathrm{set}(p) \cap \mathscr{C}_{\max} \neq \emptyset,\ d \in \mathrm{set}(p)\backslash\mathscr{C}_{\max} \neq \emptyset\} > 1. \tag{30}$$

$\rho_2(f)$ measures the "strength" of second-to-best creatures in populations containing elements of $\mathscr{C}_{\max}$. $\rho_2(f)$ is easy to determine, if the fitness function f is given by rank.

Next, we define *power-law scaling* of the fitness function in accordance with, e.g., [23, p. 124; 62, Section 2.3; 56, Section 7.1; 65, p. 65; 66, p. 100]. In fact, we set

$$f_t(c,p) = (f(c,p))^{g(t)} \quad \text{for } (c,p) \in D_f,\ t \in \mathbf{N},\ g: \mathbf{N} \to \mathbf{R}_*^+. \tag{31}$$

In addition, let $f_t(c,p) = 0$, if $(c,p) \in (\mathscr{C} \times \wp)\backslash D_f$. We shall say that a power-law scaling is *unbounded*, if $\lim_{t\to\infty} g(t) = \infty$. In this exposition, we shall only consider unbounded, *logarithmic scalings* g given by the following expression:

$$g(t) = B\ \log(t - t_0 + 2) \quad \text{for } B \in \mathbf{R}_*^+,\ t \in \mathbf{N} \cap [t_0, \infty). \tag{32}$$

It has been shown in [56, Theorem 8.5], that fast 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 [56, Definition 8.4] where one cycle of the algorithm consists of the mutation-step and picking maximal creatures in the current population.

Finally, scaled proportional fitness selection is defined as follows (see, e.g., [56, Section 7.1] for more details): For $p = (c_1, c_2, \ldots, c_s)$, $q = (d_1, d_2, \ldots, d_s) \in \wp$ with $c_\sigma, d_\sigma \in \mathscr{C}$, $1 \leqslant \sigma \leqslant s$, let $\#(d_\sigma, p)$ denote the number of copies of d_σ in the population p. In this situation, the stochastic matrix S_t^f describing probabilistic passage from p to q under scaled proportional fitness selection is given by

$$\langle q, S_t^f\, p\rangle = \left(\sum_{\sigma=1}^{s} f_t(c_\sigma, p)\right)^{-s} \cdot \prod_{\sigma=1}^{s} \#(d_\sigma, p) f_t(d_\sigma, p). \tag{33}$$

The following proposition collects basic properties of the scaled fitness selection operators S_t^f. Proposition 2.6.1.3 improves [62, Proposition 11.6].

Proposition 2.6.1. *Let p and q be populations as above and let the scaled fitness selection operator S_t^f be defined as in Eq. (33). Let $\theta = 1 - s^{-s+1}$. Then we have*

1. *If p is a uniform population, then $S_t^f\, p = p$.*
2. $\|P_{\mathscr{U}} S_t^f\, p\|_{\mathbf{1}} \geqslant \mathbf{1} - \theta$
3. *If $v \in \mathscr{S}_\wp$, then $\|(\mathbf{1} - P_{\mathscr{U}}) S_t^f v\|_1 \leqslant \theta \cdot \|(\mathbf{1} - P_{\mathscr{U}}) v\|_1$.*

Proof. Definition (33) shows statement (1). To show statement (2), assume first that $f(c_1, p) = f(c_\sigma, p)$ for every σ with $2 \leqslant \sigma \leqslant s$ and all creatures in p are different. Then, one can generate exactly s uniform populations from p with proportional fitness selection. Hence, $\|P_{\mathscr{U}} S_t^f\, p\|_1 = 1 - \theta$ in this situation. In any other situation, we obtain

from (33):

$$\|P_{\mathscr{U}} S_t^f p\|_1 \geqslant \left(\sum_{\sigma=1}^{s} f_t(c_\sigma, p)\right)^{-s} \cdot \sum_{\sigma=1}^{s} f_t(c_\sigma, p)^s$$
$$= \sum_{\sigma=1}^{s} \left(f_t(c_\sigma, p) \Big/ \left(\sum_{\sigma'=1}^{s} f_t(c_{\sigma'}, p) \right) \right)^s .$$

The latter expression takes minimal value $1-\theta$, if $f(c_1, p) = f(c_\sigma, p)$ for every σ with $2 \leqslant \sigma \leqslant s$ as a discussion employing elementary means of calculus shows. This completes the proof of statement (3). Let us finally show statement (1). Using statement (3), Lemma 1.6.1 and statement (2), we obtain for $v \in \mathscr{S}_\wp$

$$\|(\mathbf{1} - P_{\mathscr{U}}) S_t^f v\|_1 = \|(\mathbf{1} - P_{\mathscr{U}}) S_t^f (\mathbf{1} - P_{\mathscr{U}}) v\|_1 = \sum_{q,p \in \wp \setminus \mathscr{U}} \langle q, S_t^f p \rangle \langle p, v \rangle$$
$$= \sum_{p \in \wp \setminus \mathscr{U}} (1 - \|P_{\mathscr{U}} S_t^f p\|_1) \langle p, v \rangle \leqslant \theta \cdot \|(\mathbf{1} - P_{\mathscr{U}}) v\|_1 .$$

This completes the proof of statement (3). □

The following proposition illustrates the dual nature of the crossover operation. Combined with mutation, crossover can be seen as enhancing mixing/randomizing properties of mutation as formulated in [56, Theorem 6.1]. Combination with the selection operators S_t^f yields a generalized fitness scaling $\mathscr{F}_t = S_t^f \cdot C(\chi_t)$ in the sense of [56, Definition 7.1]. In fact, one has in the sense of [56, Definition 7.1]: $\wp^{\mathrm{i}} = \wp$, $\wp^{\mathrm{ii}} = \emptyset$ and $\theta_t \leqslant 1 - s^{-s+1}$ (shown in Proposition 2.6.2.2).

Proposition 2.6.2. *Let $\theta \in [0,1)$. Let C and F be stochastic matrices acting on $\mathscr{V}_\wp$ such that*

$$CP_{\mathscr{U}} = P_{\mathscr{U}}, \quad FP_{\mathscr{U}} = P_{\mathscr{U}} \quad \text{and} \quad \|(\mathbf{1} - P_{\mathscr{U}}) F v\|_1 \leqslant \theta \|(\mathbf{1} - P_{\mathscr{U}}) v\|_1 .$$

Consider the two cases $\mathscr{F} = C \cdot F$, or $\mathscr{F} = F \cdot C$. Then we have
1. *If p is a uniform population, then $\mathscr{F} p = p$.*
2. *If $v \in \mathscr{S}_\wp$, then $\|(\mathbf{1} - P_{\mathscr{U}}) \mathscr{F} v\|_1 \leqslant \theta \cdot \|(\mathbf{1} - P_{\mathscr{U}}) v\|_1$.*

Proof. The prerequisites for C and F show that $\mathscr{F} p = p$ for either definition of $\mathscr{F}$. Lemma 1.6.1 and the inequality for F show

$$\|(\mathbf{1} - P_{\mathscr{U}}) FCv\|_1 \leqslant \theta \|(\mathbf{1} - P_{\mathscr{U}}) Cv\|_1 = \theta \|(\mathbf{1} - P_{\mathscr{U}}) C (\mathbf{1} - P_{\mathscr{U}}) v\|_1$$
$$\leqslant \theta \|(\mathbf{1} - P_{\mathscr{U}}) v\|_1 \quad \text{and}$$

$$\|(\mathbf{1} - P_{\mathscr{U}}) CFv\|_1 = \|(\mathbf{1} - P_{\mathscr{U}}) C (\mathbf{1} - P_{\mathscr{U}}) Fv\|_1 \leqslant \|(\mathbf{1} - P_{\mathscr{U}}) Fv\|_1$$
$$\leqslant \theta \|(\mathbf{1} - P_{\mathscr{U}}) v\|_1$$

since both $\mathbf{1} - P_{\mathscr{U}}$ and C have operator norm 1 by [53, p. 5, Eq. (7′)]. □

3. Convergence of scaled genetic algorithms

3.1. The drive towards uniform populations

The mutation flow inequality established in Proposition 2.2.3 shows how the mutation operation controls the balance between uniform and non-uniform populations in a genetic algorithm. If the mutation flow inequality is combined in a proper way with the contraction of the selection operator towards uniform populations established in Propositions 2.6.1.3 and 2.6.2, then this ensures that the combined probability over non-uniform populations in the steady-state distribution of a simple genetic algorithm becomes small for small mutation rates. This fact is shown with stronger statement and simplified proof in the next Theorem. See [62, Theorem 15; 56, Theorems 8.1.3, 8.2.3–4] for results related to these aspects.

Theorem 3.1.1. *Let $\theta \in [0,1)$. Let $\mathscr{F}$ and X be stochastic matrices acting on $\mathscr{V}_\wp$ such that*

$$XP_{\mathscr{U}} = P_{\mathscr{U}}, \quad \mathscr{F}P_{\mathscr{U}} = P_{\mathscr{U}} \quad \text{and} \quad \|(\mathbf{1}-P_{\mathscr{U}})\mathscr{F}v\|_1 \leqslant \theta\|(\mathbf{1}-P_{\mathscr{U}})v\|_1$$

for every $v \in \mathscr{S}_\wp$. Let $\mu_0 \in (0,1]$ determine the balance between neighborhood-based search and uniform change in the spot mutation matrix $\mathbf{m}_{\mu_0}^{(1)}$ as in definition (14). Let $\mu \in (0, \frac{1}{2})$ denote the mutation rate for multiple-spot mutation $M_{\mu_0,\mu}^{(m)}$ as in Definition 2.2.1. Suppose that $\beta \in (0,1)$ is given as in Proposition 2.2.3. Then we have for $v \in \mathscr{S}_\wp$:

1. $\|(\mathbf{1}-P_{\mathscr{U}})\mathscr{F}M_{\mu_0,\mu}^{(m)}Xv\|_1 \leqslant \theta\cdot(1-\beta+\beta\|(\mathbf{1}-P_{\mathscr{U}})v\|_1)$,
2. $\|(\mathbf{1}-P_{\mathscr{U}})(\mathscr{F}M_{\mu_0,\mu}^{(m)}X)^k v\|_1 \leqslant (1-\beta)\theta/(1-\beta\theta) + (\beta\theta)^k\|(\mathbf{1}-P_{\mathscr{U}})v\|_1$ *for* $k \in \mathbf{N}$,
3. *If v is an invariant vector of $\mathscr{F}M_{\mu_0,\mu}^{(m)}X$, then* $\|(\mathbf{1}-P_{\mathscr{U}})v\|_1 \leqslant (1-\beta)\theta/(1-\beta\theta)$.

Proof. Using Propositions 2.6.2.2, 2.2.3 and Lemma 1.6.1, one has

$$\begin{aligned}
\|(\mathbf{1}-P_{\mathscr{U}})\mathscr{F}M_{\mu_0,\mu}^{(m)}Xv\|_1 &\leqslant \theta\cdot\|(\mathbf{1}-P_{\mathscr{U}})M_{\mu_0,\mu}^{(m)}Xv\|_1 \\
&\leqslant \theta\cdot(1-\beta+\beta\|(\mathbf{1}-P_{\mathscr{U}})Xv\|_1) \\
&= \theta\cdot(1-\beta+\beta\|(\mathbf{1}-P_{\mathscr{U}})X(\mathbf{1}-P_{\mathscr{U}})v\|_1) \\
&\leqslant \theta\cdot(1-\beta+\beta\|(\mathbf{1}-P_{\mathscr{U}})v\|_1).
\end{aligned}$$

since $\mathbf{1}-P_{\mathscr{U}}$ and X have operator norm 1 by [53, p. 5, Eq. (7′)]. This shows statement (1). Statement (1) shows statement (2) for $k=1$ since $1/(1-\beta\theta) \geqslant 1$. To complete the proof of statement (2), we proceed by induction

$$\begin{aligned}
\|(\mathbf{1}-P_{\mathscr{U}})(\mathscr{F}M_{\mu_0,\mu}^{(m)}X)^{k+1}v\|_1 &\leqslant \theta\cdot(1-\beta+\beta\|(\mathbf{1}-P_{\mathscr{U}})(\mathscr{F}M_{\mu_0,\mu}^{(m)}X)^k v\|_1) \\
&\leqslant \theta\cdot(1-\beta+\beta((1-\beta)\theta/(1-\beta\theta) \\
&\quad +(\beta\theta)^k\|(\mathbf{1}-P_{\mathscr{U}})v\|_1)) \\
&= (1-\beta)\theta/(1-\beta\theta) + (\beta\theta)^{k+1}\|(\mathbf{1}-P_{\mathscr{U}})v\|_1.
\end{aligned}$$

Statement (3) is now obtained as follows:

$$\begin{aligned}\|(\mathbf{1}-P_{\mathscr{U}})v\|_1 &= \lim_{k\to\infty}\|(\mathbf{1}-P_{\mathscr{U}})(\mathscr{F}M^{(m)}_{\mu_0,\mu}X)^k v\|_1\\ &\leqslant \lim_{k\to\infty}((1-\beta)\theta/(1-\beta\theta)+(\beta\theta)^k\|(\mathbf{1}-P_{\mathscr{U}})v\|_1)\\ &= (1-\beta)\theta/(1-\beta\theta). \qquad \square\end{aligned}$$

The results [54, Theorems 8.1, 8.2], [62, Theorem 17] and quite drastically [62, Theorem 8.3] show that a genetic algorithm with strictly positive mutation-rate limit cannot asymptotically converge to a probability distribution over populations containing only globally optimal creatures. Consequently, in order to obtain asymptotic convergence, the mutation rate has to be annealed to zero. Theorem 3.1.1 shows that in this situation the algorithm must converge to a probability distribution over uniform populations only. Thus, even though the goal of an optimization algorithm should be to find just one copy of a globally optimal creature, the fabric of the algorithm will asymptotically deliver a uniform population containing globally optimal creatures. We point out that a properly designed, scaled, asymptotically converging genetic algorithm as in Theorem 3.3.2 and its Corollaries allows for probabilistic estimates in regard to running the algorithm only a finite (but larger) number of cycles and approaching the limit probability distribution over uniform populations containing globally optimal creatures, cf. [30, p. 160: proof of Theorem V.4.3] or [57, Theorem 3.3.2].

3.2. Weak ergodicity

The following Theorem extends a result by Suzuki [65, p. 60, Lemma 1; 66, p. 98, Lemma 1], a result of the discussion in [61, Section 3; 56, Theorem 4.1].

Theorem 3.2.1. *Suppose that X_t, Y_t are sequences of stochastic matrices acting on $\mathscr{V}_{\wp}$, $t\in\mathbf{N}$. Let μ_0 determine the balance between neighborhood-based search and uniform change in the spot mutation matrix $\mathbf{m}^{(1)}_{\mu_0}$ as in definition (14). Let μ denote the mutation rate for multiple-spot mutation $M^{(m)}_{\mu_0,\mu}$ as Definition 2.2.1. Let $\phi_M,\phi_0\in\mathbf{R}^+_*$, $\phi'_0\in(0,1]$. Let ϕ_M and the initial value $t_0\in\mathbf{N}$ be determined such that $\phi_M\cdot t_0^{-1/(\kappa_0 L)}<\frac{1}{2}$ where $\kappa_0\in[1,\infty)$ is set as described below. Now, set $\mu(t)=\phi_M\cdot t^{-1/(\kappa_0 L)}$ for $t\in\mathbf{N}\cap[t_0,\infty)$. Let $\mu_0=\mu_0(t)$ be determined by one of the following annealing schedules*:

1. Local balance which is bounded below. *Set $\kappa_0=1$. Choices are*:
   - $\mu_0\in(0,1]$ *is kept constant. This includes the case $\alpha=2$ and for our choice of spot mutation matrix also the case $\alpha=3$.*
   - $\mu_0(t)=\phi'_0-\phi_0\cdot\mu(t)\in(0,1]$ *where $\phi_0<\phi'_0/\mu(t_0)$.*
2. Decreasing local noise. *Choose $\kappa_0\in(1,\infty)$ and set $\mu_0(t)=\phi_0\cdot\mu(t)^{\kappa_0-1}\in(0,1]$ where $\phi_0\leqslant\mu(t_0)^{1-\kappa_0}$.*

Given the above cooling schedule, set $G_t=X_t\cdot M^{(m)}_{\mu_0(t),\mu(t)}\cdot Y_t$ for $t\in\mathbf{N}\cap[t_0,\infty)$. Then the inhomogeneous Markov chain $\prod_{\tau=t}^{t_0}G_\tau$ is weakly ergodic.

Proof. For the proof one combines Lemma 1.3.1 and Proposition 2.2.2.1. □

3.3. Convergence to global optima

Theorem 3.3.2 and its corollaries are the main results of this exposition. They show that a carefully scaled genetic algorithm with *not-necessarily commuting mutation and crossover* converges *for arbitrary fitness function* to (a probability distribution over) uniform populations containing only elements of $\mathscr{C}_{\max}$ (see the beginning of Section 2.6). This simplifies and strengthens [56, Theorem 8.6, Remark 8.7] considerably in regard to applicability and implementation. However, there is a price to pay in that we require the crossover rate being annealed to 0 for the algorithm described below. Such a condition is not required in [56, Theorem 8.6, Remark 8.7]. Note that our analysis in Theorem 3.3.2 and its corollaries is, in particular, more general than the approach taken in Vose's book [68] where it is always assumed that the fitness function is *injective* [68, p. 25, footnote]. To obtain our main result, we first need the following Lemma.

Lemma 3.3.1. *Let* $k \in \mathbf{N}$, $r_0 \in [1,\infty)$ *and* $\mu_{\max}, \rho \in \mathbf{R}_*^+$. *Let* $x_{\kappa,\kappa'}, x_0 \in \mathbf{C}$, $x_0 \neq 0$ *and* $r_{\kappa,\kappa'}, \rho_\kappa \in \mathbf{R}^+$ *for* $1 \leqslant \kappa, \kappa' \leqslant k$. *Assume that all* $r_{\kappa,\kappa'}$ *are distinct. Assume that all* ρ_κ *are distinct. Let* $\hat{x}_{v,v'} \in \mathbf{C}$ *and* $\hat{r}_{v,v'}, \hat{\rho}_v \in \mathbf{R}^+$ *for* $1 \leqslant v, v' \leqslant k$. *Assume that all* $\hat{r}_{v,v'}$ *are distinct. Assume that all* $\hat{\rho}_v$ *are distinct. Suppose that* $h_p : (0,1] \to \mathbf{C}$ *satisfies*

- $h_p(\mu) = h(\mu)/\hat{h}(\mu)$ *for* $\mu \in (0, \mu_{\max}]$, *where*

$$h(\mu) = \sum_{\kappa=1}^{k} \mu^{\rho_\kappa} \cdot \left(\sum_{\kappa'=1}^{k} x_{\kappa,\kappa'} \cdot (1 + x_0 \mu^{r_0})^{r_{\kappa,\kappa'}} \right),$$

$$\hat{h}(\mu) = \sum_{v=1}^{k} \mu^{\hat{\rho}_{v'}} \cdot \left(\sum_{v'=1}^{k} \hat{x}_{v,v'} \cdot (1 + x_0 \mu^{r_0})^{\hat{r}_{v,v'}} \right) \neq 0.$$

- $|h_p(\mu)| \leqslant \rho$ *for* $\mu \in (0, \mu_{\max}]$.

In this situation, we have

1. h_p *can be extended to a continuous function in* $\mu = 0$,
2. *there exists* $j \in \mathbf{N}$ *and* $\zeta_0 > 0$ *such that the function* $\zeta \mapsto h_p(\zeta^j)$ *is continuously differentiable in* $[0, \zeta_0]$.

Proof. Suppose first that $r_0 \in \mathbf{N}$. If the complex logarithm [34, p. 111] is defined in such a way that it becomes discontinuous along $-\mathbf{R}^+$ and analytic in $\mathbf{C}\backslash(-\mathbf{R}^+)$, then the function $z \mapsto (1+z)^r$, $r \in \mathbf{R}^+$, shall be analytic in $\mathbf{C}\backslash(-\infty, -1]$. Consequently, this function has an absolutely converging power series expansion in $\mathbf{D}_{\mathbf{C}}^0 = \{z \in \mathbf{C} : |z| < 1\}$ which is uniformly converging on any compact subset of $\mathbf{D}_{\mathbf{C}}^0$ by [34, pp. 49–52, Theorems 2.4–2.6]. Combine the sums

$$\sum_{\kappa'=1}^{k} x_{\kappa,\kappa'} \cdot (1 + x_0 \mu^{r_0})^{r_{\kappa,\kappa'}} = \sum_{i=0}^{\infty} y_{\kappa,i} \mu^i \quad \text{and} \quad \sum_{v'=1}^{k} \hat{x}_{v,v'} \cdot (1 + x_0 \mu^{r_0})^{\hat{r}_{v,v'}} = \sum_{i=0}^{\infty} \hat{y}_{v,i} \mu^i$$

to absolutely converging power series for $\mu \in |x_0|^{-1/r_0} \cdot \mathbf{D}_{\mathbf{C}}^0$, where $y_{\kappa,i}, \hat{y}_{v,i} \in \mathbf{C}$ for $1 \leqslant \kappa, v \leqslant k,\ i \in \mathbf{N}_0$.

Let $\Xi = \{i + \rho_\kappa:\ 1 \leqslant \kappa \leqslant k,\ i \in \mathbf{N}_0\} \subset \mathbf{R}^+$ and $\hat{\Xi} = \{i + \hat{\rho}_v : 1 \leqslant v \leqslant k,\ i \in \mathbf{N}_0\} \subset \mathbf{R}^+$. Then one has

$$h(\mu) = \sum_{\kappa=1}^{k} \sum_{i=0}^{\infty} y_{\kappa,i} \mu^{i+\rho_\kappa} = \sum_{\rho \in \Xi} y(\rho) \mu^{\rho},$$
$$\hat{h}(\mu) = \sum_{v=1}^{k} \sum_{i=0}^{\infty} \hat{y}_{v,i} \mu^{i+\hat{\rho}_v} = \sum_{\hat{\rho} \in \hat{\Xi}} \hat{y}(\hat{\rho}) \mu^{\hat{\rho}}, \tag{34}$$

where $y(\rho), \hat{y}(\hat{\rho}) \in \mathbf{C}$ for $\rho \in \Xi,\ \hat{\rho} \in \hat{\Xi}$. All series shown in line (34) are converging absolutely and uniformly for $\mu \in [0, |x_0|^{-1/r_0}/2]$. Let

$$\rho_0 = \min\{\rho, \hat{\rho}:\ y(\rho) \neq 0 \neq \hat{y}(\hat{\rho}), \rho \in \Xi, \hat{\rho} \in \hat{\Xi}\} \geqslant 0,$$
$$i_0 = \lfloor \rho_0 \rfloor + 1,\ \ \rho_0' = i_0 - \rho_0 > 0. \tag{35}$$

Then $h_p(\mu) = (\mu^{-\rho_0} h(\mu))/(\mu^{-\rho_0} \hat{h}(\mu))$. Let $\Xi' = \{i + \rho_\kappa : 1 \leqslant \kappa \leqslant k,\ 0 \leqslant i < i_0\} \subset \Xi$. We have

$$\mu^{-\rho_0} h(\mu) = \sum_{\rho \in \Xi'} y'(\rho) \mu^{\rho - \rho_0} + \sum_{\kappa=1}^{k} \mu^{\rho_\kappa + \rho_0'} \sum_{i=i_0}^{\infty} y_{\kappa,i} \mu^{i-i_0}, \tag{36}$$

where $y'(\rho) \in \mathbf{C}$ for $\rho \in \Xi'$. All exponents for μ that are used in the finite sum over Ξ' and the power-series in line (36) are positive. The theory of power series shows that the power-series representation for $\mu^{-\rho_0} h(\mu)$ in line (36) is converging uniformly and absolutely for $\mu \in [0, |x_0|^{-1/r_0}/2]$. In fact, the radii of convergence for the power-series are determined by $(y_{\kappa,i+i_0})_{i=0}^{\infty}$, $1 \leqslant \kappa \leqslant k$ using [34, p. 52, Theorem 2.6] and stay $|x_0|^{-1/r_0}$.

A similar power-series representation as in line (36) can be derived for $\mu^{-\rho_0} \hat{h}(\mu)$. If $\mu^{-\rho_0} \hat{h}(\mu)$ does not have a non-zero constant term in its power-series representation, then ρ_0 was chosen among the $\rho \in \Xi'$. Thus, $\mu^{-\rho_0} h(\mu)$ must have a non-zero constant term $y'(\rho_0)$. In this case, h_p would be unbounded which contradicts the hypothesis of the Lemma. Hence, $\mu^{-\rho_0} \hat{h}(\mu)$ has a non-zero constant term and h_p can be extended continuously to $\mu = 0$. This completes the proof of statement (1).

Let $j \in \mathbf{N}$. Consider the following expression for $\zeta \in [0, \zeta_0]$, $\zeta_0 = (\min\{|x_0|^{-1/r_0}/2, \mu_{\max}\})^{1/j}$:

$$(\zeta^j)^{-\rho_0} h(\zeta^j) = \sum_{\rho \in \Xi'} y'(\rho) \zeta^{j(\rho - \rho_0)} + \sum_{\kappa=1}^{k} \zeta^{j(\rho_\kappa + \rho_0')} \sum_{i=i_0}^{\infty} y_{\kappa,i} \zeta^{j(i-i_0)}, \tag{37}$$

where the notation defined in Eq. (36) is used. If we chose j large enough, then every exponent in the series shown in Eq. (37) is either 0 or larger than 1. A similar consideration holds for $\zeta \mapsto (\zeta^j)^{-\rho_0} \hat{h}(\zeta^j)$. Hence for sufficiently large j, the function $\zeta \mapsto h_p(\zeta^j)$ is continuously differentiable for $\zeta \in [0, \zeta_0]$ by the quotient rule of differentiation. This proves Lemma 3.3.1 in case $r_0 \in \mathbf{N}$.

To obtain Lemma 3.3.1 in the case $r_0 \in (1,\infty)$, compose the function h_p given in Lemma 3.3.1 for $r_0 = 1$ with the function $\mu \mapsto \mu^{r_0}$ and apply the chain rule. □

A genetic algorithm as in the table in the introduction shall be called an SMC *genetic algorithm* in accordance with the action of the corresponding stochastic matrices. If the order of crossover and mutation are reversed in every cycle of the algorithm, then we shall speak of an SCM *genetic algorithm*.

Theorem 3.3.2 (Convergence to global optima). *Let* $t_0 \in \mathbf{N}$ *be fixed and* $t \in \mathbf{N} \cap [t_0,\infty)$ *enumerate the individual steps* G_t *of the scaled genetic algorithm as described below. Suppose that the genetic operators mutation, crossover and selection satisfy the following conditions*:

- $\mu_0(t)$ *determines the balance between neighborhood-based search and uniform change in the spot mutation matrix* $\mathbf{m}_{\mu_0}^{(1)}$ *as in definition* (14). $\mu(t)$ *denotes the mutation rate for multiple-spot mutation* $M_{\mu_0(t),\mu(t)}^{(m)}$ *as in Definition* 2.2.1. $\kappa_0 \in [1,\infty)$ *and the annealing schedules* $t \mapsto \mu_0(t)$ *and* $t \mapsto \mu(t)$ *are given by one of the choices listed in Theorem* 3.2.1.
- *A rational generalized crossover operation* $C(\chi_t)$ *as in Definition* 2.3.1 *is used. Suppose that the crossover rates satisfy* $\chi_t = \phi_C \mu(t)^{\kappa_0 \ell + 1}$ *where* $\phi_C \in (0, 2^{\kappa_0 \ell + 1}]$. *We shall not assume that mutation and crossover commute as operators.*
- *Power-law scaled proportional fitness selection* S_t^f *as defined in Section* 2.6 *is used with logarithmic exponentiation* $g(t) = B\log(t - t_0 + 2)$ *as in Eq.* (32). *Let* $\rho_2(f)$ *be given by definition* (30). $\rho_2(f)$ *is easy to determine if the fitness function is given by rank. Suppose that* B *satisfies*

$$\kappa_0 \ell < \kappa_0 L B \log(\rho_2(f)) + 1.$$

- *The population size satisfies* $s > \kappa_0 \ell$.

In case of an SMC *genetic algorithm, set* $k = 1$. *In case of an* SMC *genetic algorithm, set* $k = 0$. *Let the stochastic matrices* G_t *describing the individual steps of the scaled genetic algorithm be given by*

$$G_t = S_t^f \cdot C(\chi_t)^{1-k} \cdot M_{\mu_0(t),\mu(t)}^{(m)} \cdot C(\chi_t)^k, \quad k = 0, 1. \tag{38}$$

Then we have:

1. *The inhomogeneous Markov chain* $H_t = \prod_{\tau=t}^{t_0} G_\tau$ *describing the scaled genetic algorithm is strongly ergodic.*
2. *Let* $w_t = G_t w_t \in \mathscr{S}_\wp$ *denote a steady-state distribution of an individual step* G_t *of the scaled genetic algorithm. For sufficiently large* t, w_t *is uniquely determined up to scalar multiples as invariant eigenvector of* G_t. *Then* $w_\infty = \lim_{t\to\infty} w_t \in \mathscr{S}_\wp$ *exists.* w_∞ *is strictly positive only over uniform populations generated by creatures in* $\mathscr{C}_{\max}$.
3. *Let* $v_0 \in \mathscr{S}_\wp$ *be the probability distribution for the selection of the initial population. Let* $v_t = H_t v_0 \in \mathscr{S}_\wp$ *describe the state of the algorithm after step* t. *Then we*

have $\lim_{t\to\infty} v_t = w_\infty$ *as is shown in the proofs of* [30, p. 160, *Theorem* V.4.3] *or* [57, *Theorem* 3.3.2]. *Consequently, the states of the scaled genetic algorithm converge to* (*a probability distribution over*) *uniform populations generated by globally optimal creatures.*

Proof. In what follows, we consider $\mu \in (0, \phi_M \cdot t_0^{-1/(\kappa_0 L)}]$ and $t \in [t_0, \infty)$ as $\mathbf{R}$-valued parameters satisfying $\mu = \phi_M \cdot t^{-1/(\kappa_0 L)}$. G_t is well-defined for $\mathbf{R}$-valued $t \in [t_0, \infty)$.

Part 1: *Proof of uniqueness of* w_t. If $k = 1$, then $M^{(m)}_{\mu_0(t),\mu(t)} \cdot C(\chi_t)$ is fully positive for any stochastic crossover matrix $C(\chi_t)$ by Proposition 2.2.2.1. In addition, $M^{(m)}_{\mu_0(t),\mu(t)}$ is invertible by Proposition 2.2.2.3, and $C(\chi_t)$ is invertible for sufficiently large t by Lemma 2.3.2.2. Hence in both cases $k = 0, 1$, we obtain by Lemma 1.4.2.2 that w_t is uniquely defined as an invariant eigenvector of G_t up to scalar multiples for large t.

Part 2: *Proof of statement* (1). Theorem 3.2.1 shows that the inhomogeneous Markov chain H_t is weakly ergodic for $k = 0, 1$. In order to show that H_t is strongly ergodic, we shall apply [30, p. 160, Theorem V.4.3] or [57, Theorem 3.3.2]. In order to complete Part 2, we only have to show that the sequence $(\|w_{t+1} - w_t\|_1)_{t\in\mathbf{N}\cap[t_0,\infty)}$ is summable, i.e., $\sum_{t=t_0}^{\infty} \|w_{t+1} - w_t\|_1 < \infty$. Since $\wp$ is finite, the latter is achieved by showing that $(|\langle p, (w_{t+1} - w_t)\rangle|)_{t\in\mathbf{N}\cap[t_0,\infty)}$ is summable for every $p \in \wp$.

By Part 1 of the proof, we know that the kernel of $G_t - \mathbf{1}$ is generated by $w_t \in \mathscr{S}_\wp \subset (\mathbf{R}^+)^{\alpha^L}$ for large t. Adding the rows of $G_t - \mathbf{1}$ to the first row, we see that the kernel of $(G_t - \mathbf{1})^{[0]}$ is generated by w_t as well. Hence, $(G_t - \mathbf{1})^{[e^*]}$ has kernel $\{0\}$ and the equation $(G_t - \mathbf{1})^{[e^*]} w_t = (\alpha^{-L}, 0, \ldots, 0)^*$ uniquely determines w_t. In this situation, w_t can be computed using Cramer's Rule [33, p. 182, Theorem 3].

Let $T = t - t_0 + 2 = \phi_M^{\kappa_0 L} \mu^{-\kappa_0 L} - t_0 + 2$. Expressing the coefficients of the fitness selection operator in terms of T, we obtain from Eq. (33):

$$\langle q, S_t^f\, p\rangle = \left(\sum_{\sigma=1}^{s} T^{B\log(f(c_\sigma,p))}\right)^{-s} \cdot \prod_{\sigma=1}^{s} \#(d_\sigma, p) T^{B\log(f(d_\sigma,p))} \tag{39}$$

for $p = (c_1, c_2, \ldots, c_s)$, $q = (d_1, d_2, \ldots, d_s) \in \wp$, $c_\sigma, d_\sigma \in \mathscr{C}$, $\#(d_\sigma, p) \in \mathbf{N}_0$, $1 \leqslant \sigma \leqslant s$. For $p \in \wp$, set $h_p(\mu) = \langle p, w_t\rangle$ with $t = \phi_M^{\kappa_0 L} \mu^{-\kappa_0 L}$. By Proposition 2.2.2.1, Definition 2.3.1.1 and Eq. (39), we conclude that the computation of $h_p(\mu)$ using Cramer's rule yields a function that satisfies the prerequisites of Lemma 3.3.1 with $r_0 = \kappa_0 L$, $\mu_{\max} \leqslant \phi_M t_0^{-1/(\kappa_0 L)}$, $\rho = 1$ and $x_0 = (2 - t_0)/\phi_M^{\kappa_0 L}$. By Lemma 3.3.1.2, there exists $j \in \mathbf{N}$ and $\zeta_0 > 0$ such that the function $\zeta \mapsto \Psi_p(\zeta) = h_p(\zeta^j)$ is continuously differentiable in $[0, \zeta_0]$. Let $t_1 \in (\zeta_0^{-j\kappa_0 L}, \infty) \cap \mathbf{N}$. Let $K \in \mathbf{R}_*^+$ be such that $|(\mathrm{d}/\mathrm{d}\zeta)\Psi_p(\zeta)| \leqslant K$ on $[0, \zeta_0]$. Then we have

$$\begin{aligned}\sum_{t\in[t_1,\infty)\cap\mathbf{N}} |\langle p, (w_{t+1} - w_t)\rangle| &= \sum_{t=t_1}^{\infty} |\Psi_p(\phi_M^{1/j}(t+1)^{-1/(j\kappa_0 L)}) - \Psi_p(\phi_M^{1/j} t^{-1/(j\kappa_0 L)})| \\ &\leqslant \sum_{t=t_1}^{\infty} K \cdot \phi_M^{1/j} \cdot |(t+1)^{-1/(j\kappa_0 L)} - t^{-1/(j\kappa_0 L)}| < \infty.\end{aligned}$$

Part 3: *Proof of statement* (2). The first part of statement (2) concerning uniqueness of the w_t has already been shown in Part 1 of the proof. The limit $w_\infty \in \mathscr{S}_\wp$ exists since the $w_t \in \mathscr{S}_\wp$, $t \in \mathbf{N}$, satisfy $\sum_{t=t_0}^{\infty} \|w_{t+1} - w_t\|_1 < \infty$ and, consequently, form a Cauchy sequence. Theorem 3.1.1.3 implies

$$\|(\mathbf{1} - P_{\mathscr{U}})w_\infty\|_1 = \lim_{t\to\infty} \|(\mathbf{1} - P_{\mathscr{U}})w_t\|_1 \leqslant \lim_{t\to\infty} (1 - \beta(\mu(t)))\theta/(1-\theta) = 0,$$

where $\beta = \beta(\mu(t)) \in (0,1)$ is given as in Proposition 2.2.3, and θ as in Proposition 2.6.1.3. This shows that w_∞ is non-zero only over uniform populations. To complete Part 3 of the proof, we show now that w_∞ is strictly positive only over populations in $\mathscr{C}^s_{\max} \subset \wp$, i.e., populations that contain only globally optimal creatures.

The idea for the following argument is to derive an estimate for the probabilistic flow between $\Omega = \mathscr{C}^s_{\max}$ and $\Omega' = \wp \setminus \mathscr{C}^s_{\max}$, if the homogeneous Markov chain defined by G_t is in steady state. This is based upon the fact that $w_t = G_t w_t$. Let P_Ω be the orthogonal projection onto $\mathrm{span}_{\mathbf{C}}(\Omega)$ and let $P_{\Omega'}$ be the orthogonal projection onto $\mathrm{span}_{\mathbf{C}}(\Omega')$. Let $\omega(t) = \|P_\Omega w_t\|_1$. Let $\Omega^+ = \{p \in \wp : \mathrm{set}(p) \cap \mathscr{C}_{\max} \neq \emptyset\}$.

Part 3a: *The flow towards* Ω. In order to make a transition under mutation from $q' \in \Omega'$ to a population $q^+ \in \Omega^+$, one has to change at most the letters in the spots corresponding to a single creature in q. For each of these changes, the probability under mutation is at least $\mu_0/(\alpha-1)$, cf. Lemma 2.1.1.3. If t is chosen sufficiently large, then the crossover rate is sufficiently small such that Lemma 2.3.2.1 can be applied to $C(\chi_t)$. A transition from $q^+ \in \Omega_+$ to a population $p \in \Omega \cap \mathscr{U}$ under fitness selection occurs now with probability bounded below by s^{-s+1}, cf. Proposition 2.6.1.2. Hence, we have for $K_1 \in \mathbf{R}^+_*$ and summations over $p \in \Omega \cap \mathscr{U}$, $q^+ \in \Omega^+$, $q' \in \Omega'$

$$\begin{aligned}
\|P_\Omega G_t P_{\Omega'} w_t\|_1 &\geqslant \sum_{p,q'} \langle p, S_t^f C(\chi_t)^{1-k} M^{(m)}_{\mu_0(t),\mu(t)} C(\chi_t)^k q' \rangle \langle q', w_t \rangle \\
&\geqslant \sum_{p,q^+,q'} \langle p, S_t^f q^+ \rangle \langle q^+, C(\chi_t)^{1-k} q^+ \rangle \langle q^+, M^{(m)}_{\mu_0(t),\mu(t)} q' \rangle \\
&\quad \times \langle q', C(\chi_t)^k q' \rangle \langle q', w_t \rangle \\
&\geqslant \sum_{q^+,q'} s^{-s+1} \frac{1}{2} \langle q^+, M^{(m)}_{\mu_0(t),\mu(t)} q' \rangle \langle q', w_t \rangle \\
&\geqslant \frac{1}{2} s^{-s+1} (\mu\mu_0/(\alpha-1))^\ell \sum_{q'} \langle q', w_t \rangle \geqslant K_1 \mu^{\kappa_0 \ell} \cdot \|P_{\Omega'} w_t\|_1. \qquad (40)
\end{aligned}$$

Hence, we have

$$\begin{aligned}
\|P_{\Omega'} G_t P_{\Omega'} w_t\|_1 &= \|P_{\Omega'} w_t\|_1 \cdot \|P_{\Omega'} G_t (\|P_{\Omega'} w_t\|_1^{-1} P_{\Omega'} w_t)\|_1 \\
&= \|P_{\Omega'} w_t\|_1 \cdot (1 - \|P_{\Omega'} w_t\|_1^{-1} \|P_\Omega G_t P_{\Omega'} w_t\|_1) \\
&\leqslant \|P_{\Omega'} w_t\|_1 \cdot (1 - K_1 \mu^{\kappa_0 \ell}). \qquad (41)
\end{aligned}$$

Let $q^+ = (c_1, c_2, \ldots, c_s) \in \Omega^+$, $c_\sigma \in \mathscr{C}$, $1 \leqslant \sigma \leqslant s$. Let $\nu_{\max}$ denote the number of spots in p that are occupied by elements in $\mathscr{C}_{\max}$. The probability for selecting an arbitrary

element $d \in \mathrm{set}(q^+) \cap \mathscr{C}_{\max}$ in the process of the scaled proportional fitness selection operation is then given by

$$v_{\max} f_t(d, q^+) \Big/ \left(\sum_{\sigma=1}^{s} f_t(c_\sigma, q^+) \right). \tag{42}$$

The expression in line (42) is bounded below by $(1 + (s-1)T^{-B\log(\rho_2(f))})^{-1}$. Hence we have

$$\|P_\Omega S_t^f q^+\|_1 \geqslant (1 + (s-1)(\phi_M^{\kappa_0 L}\mu^{-\kappa_0 L} + 2 - t_0)^{-B\log(\rho_2(f))})^{-s}.$$

Hence, there exists $K_2 \in \mathbf{R}_*^+$ such that for sufficiently small μ:

$$\|P_{\Omega'} S_t^f q^+\|_1 = 1 - \|P_\Omega S_t^f q^+\|_1 \leqslant K_2 \mu^{\kappa_0 LB\log(\rho_2(f))}. \tag{43}$$

Part 3b: *The flow towards* Ω'. In order to estimate the probabilistic flow from Ω to Ω' in application of G_t to w_t, we distinguish two cases:

Case 1: *Initial crossover-mutation step destructs all globally optimal creatures.* In order to make a transition from $p \in \Omega$ to a population $q^c \in \wp \backslash \Omega^+$ via crossover-mutation, one has to change every creature in p in appropriate fashion. In that case, a subsequent selection operation cannot generate an element of Ω^+. If $k = 1$, i.e., crossover is applied first, then the crossover operation alone may achieve changing every creature in p. By Eq. (15), the combined probability for this to happen is bounded by $K_3\mu^{\kappa_0 \ell + 1}$, $K_3 \in \mathbf{R}_*^+$. Mutation may then keep the resulting $q^c \in \wp \backslash \Omega^+$. If crossover did not change $p \in \Omega$, then mutation has to alter at least one spot in every creature in p. By Proposition 2.2.2.1, the combined probability for this to happen is bounded by $K_4\mu^s$, $K_4 \in \mathbf{R}_*^+$. If crossover did only change some creatures in $p \in \Omega$, then mutation has to alter at least one spot in every unaltered creature in p. This happens with probabilities that are bounded by higher-order terms than the previous two cases discussed. If $k = 0$, then a similar argument can be used.

Case 2: *Initial crossover-mutation step retains globally optimal creatures.* An initial application of $C(\chi_t)^{1-k} M^{(m)}_{\mu_0(t),\mu(t)} C(\chi_t)^k$ to $p \in \Omega$ yields elements $q^+ \in \Omega^+ \backslash \Omega$ with probability bounded from above by $K_5\mu$, $K_5 \in \mathbf{R}_*^+$, since at least one spot in p must be changed by mutation, or crossover must be applied. If selection is applied to q^+, then the combined probability to generate elements of Ω' is bounded from above by $K_2\mu^{\kappa_0 LB\log(\rho_2(f))}$ as was shown in line (43).

Hence, we have for $K_6 \in \mathbf{R}_*^+$ and summations over $q' \in \Omega'$, $q^c \in \wp \backslash \Omega^+$, $q^+ \in \Omega^+ \backslash \Omega$, $p \in \Omega$:

$$\begin{aligned}
&\|P_{\Omega'} G_t P_\Omega w_t\|_1 \\
&\quad = \sum_{q', q^c, p} \langle q', S_t^f q^c\rangle \langle q^c, C(\chi_t)^{1-k} M^{(m)}_{\mu_0(t),\mu(t)} C(\chi_t)^k p\rangle \langle p, w_t\rangle \\
&\qquad + \sum_{q', q^+, p} \langle q', S_t^f q^+\rangle \langle q^+, C(\chi_t)^{1-k} M^{(m)}_{\mu_0(t),\mu(t)} C(\chi_t)^k p\rangle \langle p, w_t\rangle
\end{aligned}$$

$$\begin{aligned}
&\leqslant \sum_{q^c,p} \langle q^c, C(\chi_t)^{1-k} M^{(m)}_{\mu_0(t),\mu(t)} C(\chi_t)^k p\rangle \langle p, w_t\rangle \\
&\quad + \sum_{q^+,p} K_2 \mu^{\kappa_0 LB \log(\rho_2(f))} \langle q^+, C(\chi_t)^{1-k} M^{(m)}_{\mu_0(t),\mu(t)} C(\chi_t)^k p\rangle \langle p, w_t\rangle \\
&\leqslant \sum_{p} K_6(\mu^{\kappa_0\ell+1} + \mu^s)\langle p, w_t\rangle + \sum_{p} K_2 K_5 \mu^{\kappa_0 LB\ \log(\rho_2(f))+1} \langle p, w_t\rangle \\
&= (K_6\mu^{\kappa_0\ell+1} + K_6\mu^s + K_2K_5\mu^{\kappa_0 LB\log(\rho_2(f))+1})\omega(t). \qquad (44)
\end{aligned}$$

Part 3c: *The steady-state flow inequality.* Combining inequalities (41) and (44) yields the steady-state flow inequality as follows:

$$\begin{aligned}
1-\omega(t) &= \|P_{\Omega'} w_t\|_1 = \|P_{\Omega'} G_t P_\Omega w_t\|_1 + \|P_{\Omega'} G_t P_{\Omega'} w_t\|_1 \\
&\leqslant (K_6\mu^{\kappa_0\ell+1} + K_6\mu^s + K_2K_5\mu^{\kappa_0 LB\log(\rho_2(f))+1})\omega(t) \\
&\quad + (1 - K_1\mu^{\kappa_0\ell})(1-\omega(t)). \qquad (45)
\end{aligned}$$

Inequality (45) shows that $\lim_{t\to\infty} \omega(t) = 1$. This completes Part 3 of the proof and the proof of Theorem 3.3.2. □

Theorem 3.3.2 requires that the crossover rate is annealed rather fast compared with mutation. If we use a regular crossover operation, then Theorem 3.3.2 can be strengthened considerably in this regard as the following Corollary shows.

Corollary 3.3.3 (Convergence to global optima). *Let* $m \in [1,\infty)$. *Suppose that the prerequisites of Theorem* 3.3.2 *hold except for the following changes*:

- *The crossover operation is given by* (*scaled*) *one-*, *two-cutpoint or uniform regular crossover* (*see Definitions* 2.4.2, (20) *and* (22)). *The crossover rate is given by* $\chi_t = \phi_C \mu(t)^{1/m}$ *where* $\phi_C \in (0, 2^{1/m}]$.
- *B satisfies*: $\kappa_0\ell < \kappa_0 LB \log(\rho_2(f)) + 1/m$.
- *The population size satisfies*: $s > 2m\kappa_0\ell$.

Then the conclusions of Theorem 3.3.2 *hold.*

Proof. To prove Corollary 3.3.3, the proof of Theorem 3.3.2 can be copied except for the discussion in Part 3b, Case 1 and resulting estimates in the steady-state flow inequality. With the notation used in the proof of Theorem 3.3.2 we have

Case 1′: In order to make a transition from $p \in \Omega$ to a population $q^c \in \wp\backslash\Omega^+$ via crossover-mutation, one has to change every creature in p in appropriate fashion. In that case, a subsequent selection operation cannot generate an element of Ω^+. If $k = 1$, i.e., crossover is applied first, then the crossover operation alone may achieve changing every creature in p. By Eq. (18) or (20), the combined probability for this to happen is bounded by $K_3\chi_t^{s/2} = K_3\phi_C^{s/2}\mu(t)^{s/(2m)}$, $K_3 \in \mathbf{R}_*^+$. Mutation may then keep the resulting $q^c \in \wp\backslash\Omega^+$. If crossover only changed σ creatures in $p \in \Omega$ which is bounded from above by terms in the order of $\mu(t)^{\sigma/(2m)}$, then mutation has to alter at least one spot in

the unchanged $s-\sigma$ creature in p. By Proposition 2.2.2.1, the combined probability for the latter to happen is bounded by $K_4\mu^{s-\sigma}$, $K_4 \in \mathbf{R}_*^+$. The asymptotically largest estimate (i.e., for sufficiently small μ) obtained in this discussion is the term $K_3\phi_C^{s/2}\mu(t)^{s/(2m)}$. The case $k=0$ need not be discussed since mutation and crossover commute. Proof of the latter statement is discussed after Definition 2.4.1.

Now, the steady-state flow inequality is obtained as follows with $K_6 \in \mathbf{R}_*^+$:

$$\begin{aligned} 1-\omega(t) \leqslant{}& (K_6\mu^{s/(2m)} + K_2K_5\mu^{\kappa_0 LB\log(\rho_2(f))+1/m})\omega(t) \\ &+ (1-K_1\mu^{\kappa_0\ell})(1-\omega(t)), \end{aligned} \tag{46}$$

which yields $\lim_{t\to\infty}\omega(t)=1$. □

If we use scaled, spot-wise gene-lottery crossover, then Theorem 3.3.2 can be strengthened too. This is shown in the following corollary.

Corollary 3.3.4 (Convergence to global optima). *Let* $m \in [1,\infty)$. *Suppose that the prerequisites of Theorem* 3.3.2 *hold except for the following changes*:

- *The crossover operation is given by scaled, spot-wise gene-lottery crossover* (*see Definitions* 2.5.1). *The crossover rate is given by* $\chi_t = \phi_C\mu(t)^{1/m}$ *where* $\phi_C \in (0, 2^{1/m}]$.
- *B satisfies*: $\kappa_0\ell < \kappa_0 LB\log(\rho_2(f)) + 1/m$.
- *The population size satisfies*: $s > m\kappa_0\ell$.

Then the conclusions of Theorem 3.3.2 *hold.*

Proof. To prove Corollary 3.3.4, the proof of Theorem 3.3.2 can again be copied except for the discussion in Part 3b, Case 1 and resulting estimates in the steady-state flow inequality. With the notation used in the proof of Theorem 3.3.2 we have

Case 1″: In order to make a transition from $p \in \Omega$ to a population $q^c \in \wp\backslash\Omega^+$ via crossover-mutation, one has to change every creature in p in appropriate fashion. Thus, one has to alter at least one spot in every creature of p. In that case, a subsequent selection operation cannot generate an element of Ω^+. If $k=1$, i.e., crossover is applied first, then the crossover operation alone may achieve changing every creature in p. By Eqs. (24) and (25), the combined probability for this to happen is bounded by $K_3\chi_t^s = K_3\phi_C^s\mu(t)^{s/m}$, $K_3 \in \mathbf{R}_*^+$. Mutation may then keep the resulting $q^c \in \wp\backslash\Omega^+$. If crossover did only change σ creatures in $p \in \Omega$, then mutation has to alter at least one spot in the unchanged $s-\sigma$ creature in p. By Proposition 2.2.2.1, the combined probability for this to happen is bounded by $K_4\mu^{s-\sigma}$, $K_4 \in \mathbf{R}_*^+$. Overall, the required change can be estimated for small μ by an upper bound of order $\mu(t)^{s/m}$. The case $k=0$ is discussed similarly.

Now, the steady-state flow inequality is obtained as follows with $K_6 \in \mathbf{R}_*^+$:

$$\begin{aligned} 1-\omega(t) \leqslant{}& (K_6\mu^{s/m} + K_2K_5\mu^{\kappa_0 LB\log(\rho_2(f))+1/m})\omega(t) \\ &+ (1-K_1\mu^{\kappa_0\ell})(1-\omega(t)), \end{aligned} \tag{47}$$

which yields $\lim_{t\to\infty}\omega(t)=1$. □

Remark 3.3.5. Corollaries 3.3.3 and 3.3.4 show (with mathematical theory and not experimentally) the quite remarkable effect that with increasing population size, one is allowed to use a more relaxed cooling schedule for crossover. Thus for larger population size, the part of the algorithm design, i.e., definition of creatures (data structures), which is exploited by crossover plays a more important role. Overall, crossover has more time and opportunity to perform its enhancement of the mixing phase of the genetic algorithm. See [56, Theorem 6.1] where this statement is given a precise meaning in terms of contraction properties of the combined crossover-mutation operator in case of regular crossover.

Remark 3.3.6. Anily and Federgruen [3, Theorem 2] have shown for the simulated annealing algorithm that one can use certain non-monotone sequences for the cooling parameter and still obtain an asymptotically converging simulated annealing algorithm. The reader may adapt Anily and Federgruen's work to the situation of Theorem 3.3.2 to obtain asymptotically converging genetic algorithms with more general annealing schedules for mutation than presented in this work.

The author conjectures that Theorem 3.3.2 and its corollaries can possibly be generalized to the following situations with random-nature annealing schedules for the mutation rate μ and the spot mutation rate μ_0: μ is chosen at runtime depending upon the state of the algorithm. If the algorithm repeatedly returns to the same state (i.e., population), then μ is increased by a certain magnitude. Overall, μ follows one of the trajectories proposed in Theorem 3.2.1. For μ_0 we have: (1) *Local balance which is bounded below*. Set $\kappa_0 = 1$. $\mu_0(t) \in (\varphi, 1]$ is chosen at runtime depending upon the state of the algorithm, $\varphi \in (0, 1)$. If the algorithm repeatedly returns to the same state (i.e., population), then the local noise is increased. (2) *Decreasing local noise*. Set $\kappa_0 = 2$. $\mu_0(t) \in [\mu(t), 1]$ is chosen at runtime depending upon the state of the algorithm. Overall, $\mu_0(t)$ follows the trajectory of the $\mu(t)$ with occasional bursts.

Section 4 of [57] discusses a number of other possible generalizations and continuations of the work presented in this exposition. In the opinion of this author, adapting the approach to theoretical treatment of scaled genetic algorithms presented here to the case of a non-fully positive mutation matrix (see, e.g., [56, Section 3.2]) is the most challenging and important next advance.

4. Additional extensions of the theory

4.1. Not-necessarily symmetric generalized crossover operators

In [56, Section 5.1.1] generalized crossover is considered with the additional assumption that the stochastic matrices $C(\chi)$ are symmetric. We shall outline in this section that this condition can actually be dropped almost entirely in [56]. The main results of [56], in particular, in regard to strictly positive limit mutation rate stay valid. This yields *valuable examples for non-converging genetic algorithms* that use non-symmetric crossover operations such as gene-lottery crossover.

4.1.1. Genetic drift

The discussion in [56, Section 7.5] holds without change. We observe that simple genetic algorithms with zero mutation rate and any crossover are dangerous, non-ergodic procedures. Thus, an extremely low mutation rate in a simple genetic algorithm may as well yield misleading experimental results. This point of view is in accordance with experimental results by, e.g., Banzhaf et al. [6].

In this context, the author notes that investigations of genetic algorithms without mutation as in [9,22,64,73] must fail to produce explicit convergence theorems with scaling schedules such as Theorem 3.3.2 for the principal reason that these paradigms rely on a *non-ergodic* exploration of the search space via genetic drift. Similarly, schema theory for genetic programming without mutation as in [46, Theorems 1–4] must fail to produce explicit convergence theorems. This concurs with the point of view expressed by Vose [68, p. 211, lines 1–4].

4.1.2. Strictly positive limit mutation rate and single-spot mutation

In the case of the SCM genetic algorithm, [56, Theorem 8.1] stays valid and can even be extended to the case of a single-spot mutation as defined in [56, Section 3.2] which uses the spot mutation matrix given by definition (14). Ref. [56, Theorem 8.1] can be extended to the case of the SMC genetic algorithm too. In that case, [56, Theorem 8.1.3] is obtained with the help of Theorem 3.1.1.3.

4.1.3. Strictly positive limit mutation rate and multiple-spot mutation

In the case of the SCM genetic algorithm, [56, Theorems 8.2.1,3,4] hold without change and can even be extended in that $M^{(m)}_{\mu_0,\mu}$ as discussed here is used as mutation operation. In order to obtain [56, Theorem 8.2.2], one has to suppose that $C_{\chi_\infty} - \varepsilon\mathbf{1}$ is a positive matrix for some $\varepsilon \in (0,1]$ similar to the conditions for [56, Theorem 8.2.4]. Such a condition is satisfied for scaled, spot-wise gene-lottery crossover by Eqs. (24) and (25). If $p \in \wp$ is a population of uniform fitness as in the proof of [56, Theorem 8.2.2, pp. 46 (line 28)–47 (line 5)], then $\langle p, C_{\chi_\infty} M^{(m)}_{\mu_{0,\infty},\mu_\infty} v_\infty \rangle \neq 0$ for $v_\infty \in \mathscr{S}_\wp$ simply because $M^{(m)}_{\mu_{0,\infty},\mu_\infty}$ is fully positive and C_{χ_∞} has positive diagonal.

In the case of the SMC genetic algorithm, [56, Theorem 8.2] holds without change and for $M^{(m)}_{\mu_0,\mu}$ as mutation operation. For the proof, one notes that $M^{(m)}_{\mu_{0,\infty},\mu_\infty} C_{\chi_\infty}$ is a fully positive matrix. This yields strong ergodicity in [56, Theorem 8.2.1] and $\langle p, M^{(m)}_{\mu_{0,\infty},\mu_\infty} C_{\chi_\infty} v_\infty \rangle \neq 0$ in the proof of [56, Theorem 8.2.2, pp. 46 (line 28)–47 (line 5)]. Again, [56, Theorem 8.2.3] follows from Theorem 3.1.1.3. The argument for the proof of [56, Theorem 8.2.4, p. 47] can be easily modified to the case of the SMC genetic algorithm. Note that [56, p. 47] contains a systematic typographical error: $M^{(1)}_{\mu_\infty}$ should always be replaced by $M^{(m)}_{\mu_\infty}$.

Ref. [56, Theorem 8.3] holds for both the SCM- and SMC-genetic algorithm. Note that [56, Theorem 8.3.3] does not suppose that the fitness function induces an order on the set of creatures $\mathscr{C}$.

4.1.4. Zero-limit mutation rate and multiple-spot mutation

Ref. [56, Theorem 8.5] stays valid for commuting crossover and mutation operators. The property of crossover being symmetric is used nowhere in the proof. Note that

[56, Theorem 8.5] does not even suppose that crossover is a continuous function of the crossover rate.

Ref. [56, Theorem 8.6, Remark 8.7] also stay valid for not-necessarily-symmetric crossover matrices. Again, symmetry of the crossover matrices is used nowhere in the proof.

4.2. Invariant subspaces for commuting crossover and mutation

The following proposition simplifies and strengthens [62, Propositions 1.7, 3.5; 58, Propositions 3.5, 3.8]. It states that any (mutation) matrix that commutes with a specific crossover operator maps the space of invariant vectors of that crossover operator into itself. Note that for the proof of Proposition 4.2.1, one only has to know that $\bar{D}$ and D (defined below) are the spaces of invariant vectors for the respective crossover operators.

Let Π_s denote the group of permutations of s elements. Elements of Π_s act canonically on $\wp$ by shuffling creatures (see [56, Section 2.9] for a more detailed definition). Let P_Π be the average/mean over those linear, stochastic operators on $\mathcal{V}_\wp$ that are induced by the canonical action of the group Π_s on $\wp$. The maps $v \mapsto P_\Pi v$, $v \in \mathcal{V}_\wp$ and $X \mapsto P_\Pi X P_\Pi$ for matrices X acting on $\mathcal{V}_\wp$ embed the model for genetic algorithms based upon the multi-set representation for populations into the model presented in this exposition. For more details on this claim and the projection P_Π see [56, p. 13].

Proposition 4.2.1. *Let $\chi \in (0,1]$ be fixed. Let M be a matrix acting on $\mathcal{V}_\wp$. Suppose that M commutes with one of the crossover matrices* (1)–(7) *listed below.*

1. (1) *The one-cutpoint regular crossover matrix $C^{(1)}_{\mathrm{reg}}(\chi)$ given by Eq.* (18), (2) *the two-cutpoint regular crossover matrix $C^{(2)}_{\mathrm{reg}}(\chi)$ given by Eq.* (20), (3) *the uniform regular crossover matrix $C^{(u)}_{\mathrm{reg}}(\chi)$ given by* Eq. (22).

   Let $\chi < 1$. Let $\mathscr{D} = \bar{D}$ with $\bar{D}$ as in [56, p. 15]. *$\bar{D}$ is the space of invariant vectors for the matrices* (1)–(3). *This follows from* [62, Proposition 7.6][8] *which holds also in the case of two-cutpoint/uniform regular crossover.*
2. (4) *The one-cutpoint regular crossover matrix in the model based upon multi-set representations for populations given by $P_\Pi C^{(1)}_{\mathrm{reg}}(\chi) P_\Pi$ ($\chi < 1$),* (5) *the two-cutpoint regular crossover matrix in the model based upon multi-set representations for populations given by $P_\Pi C^{(2)}_{\mathrm{reg}}(\chi) P_\Pi$ ($\chi < 1$),* (6) *the uniform regular crossover matrix in the model based upon multi-set representations for populations given by $P_\Pi C^{(u)}_{\mathrm{reg}}(\chi) P_\Pi$ ($\chi < 1$),* (7) *the unrestricted crossover matrix $C^{(u)}_\chi$ in the sense of* [62, Section 2.2, p. 117].

   Let $\mathscr{D} = D$ with D as in [56, p. 14]. *D is the space of invariant vectors for the matrices* (4)–(7). *This follows from* [58, Lemma 5.1] *which holds for one/two-cutpoint and uniform regular crossover and from* [62, Proposition 9.6] *for unrestricted crossover.*

Then $M\mathscr{D} \subset \mathscr{D}$.

[8] The last equation listed in the statement of [62, Proposition 7.6] should better be: $\mathbf{C}_\chi \mathbf{P}_D = \mathbf{P}_D = \mathbf{P}_D \mathbf{C}_\chi$. This also holds for [62, Proposition 9.6].

Proof. Let $v \in \mathscr{D}$. Let C stand for one of the crossover operators listed above. Then we have for $k \in \mathbf{N}$

$$Mv = Mk^{-1} \sum_{\kappa=1}^{k} C^{\kappa} v = k^{-1} \sum_{\kappa=1}^{k} C^{\kappa} Mv = \lim_{k \to \infty} k^{-1} \sum_{\kappa=1}^{k} C^{\kappa} Mv \in \mathscr{D} \tag{48}$$

by [61, p. 10, Theorem 3.1]. □

4.3. The Vose-Liepins version of mutation-crossover

Vose [68, Section 5.4, p. 44] describes one cycle of the classical simple genetic algorithm as follows: (a) obtain two parents by the selection function, (b) mutate the parents by the mutation function, (c) produce the mutated parent's child by the crossover function, (d) put the child into the next generation, (e) if the next generation contains less than r members, go to step (a). r is the population size. The acronym VLGA shall refer to a genetic algorithm whose cycle is described by steps (a)–(e) as above. See also the analysis by Vose and Liepins [69].

The procedure of pairing creatures to produce offspring in the VLGA produces one child at a time while regular crossover as defined in Section 2.4, in [56, Section 5.2.1], [62, Section 2.2] following, e.g., Goldberg's book [23, pp. 16–17] produces two offspring from two parents in a single crossover-step. Mitchell [42, p. 139, line 19] writes that the procedural difference was assumed for reason of mathematical simplicity. In what follows, we shall show that the VLGA can be easily embedded into the model developed here and in [61,62].

If $r \in \mathbf{N}$ denotes the finite population size which corresponds to a computer-implementation of a VLGA, then set $s = 2 \cdot r$. Suppose that $f : \mathscr{C} \to \mathbf{R}_*^+$ is the originally given fitness function in the sense of [68, p. 25]. We need not assume the f is injective as in all of [68].

Let $p = (c_1, c_2, \ldots, c_s), q = (d_1, d_2, \ldots, d_s) \in \wp$ be populations, $c_\sigma, d_\sigma \in \mathscr{C}$, $1 \leqslant \sigma \leqslant s = 2r$. In order to embed the VLGA into the model developed here, we modify the selection procedure given by the definition prior to line (33) in such a way that creatures for the next population q are selected only from positions σ in the present population p where σ is even. That means: one only selects creatures from positions σ in $J = 2\mathbf{N} \cap [2, s]$. This yields a selection operator S_t^f whose stochastic matrix is given as follows:

$$\langle q, S_t^f\, p \rangle = \left(\sum_{\sigma' \in J} f_t(c_{\sigma'}, p) \right)^{-s} \cdot \prod_{\sigma=1}^{s} \#(d_\sigma, p \wedge J) f_t(d_\sigma, p). \tag{49}$$

Now, we can restate the definition of one cycle of the VLGA as follows:

1. Given $p \in \wp$, apply proportional fitness selection[9] as defined in line (49). The fitness function f can be scaled or unscaled ($g(t) = 1$, cf. Eq. (31)). Line (49) shows that

[9] We can also use other selection methods here such as tournament fitness selection involving only the $c_{2\sigma}$ in p, $1 \leqslant \sigma \leqslant r$. See [25,40,41,61,54, p. 78; p. 170; p. 59; p. 153; Section 7.2].

this step effectively creates r pairs of "parents" from r creatures $(c_2, c_4, \ldots, c_{2s})$ in p. The creatures $(c_1, c_2, \ldots, c_{2s-1})$ in p are disregarded.

2. Apply the mutation operator to the population. One choice for a mutation operator is multiple-spot mutation $M^{(m)}_{\mu_0,\mu}$. Vose [68, p. 42] allows for additional choices for mutation. We leave a discussion of these possibilities to the reader.
3. Apply regular crossover (or gene-lottery crossover as in Section 2.5) to the population obtained after the mutation operation.

Note that the above list describes a mathematical model for the VLGA and *not* a proposed method of implementation. After every cycle, the next fitness selection step (1) will disregard the creatures $c_{2\sigma-1}$ for $1 \leqslant \sigma \leqslant r$ obtained through mutation-crossover in the previous cycle and randomly arrange the chosen parents. Thus, we do not have to perform a "selection step from the children" as in [68, p. 43, line 25]. The algorithm whose model is described in steps (1)–(3) above is in complete accordance with the definition of the VLGA.

The new model for the VLGA presented in steps (1)–(3) above allows for application/adaptation of the results in [56], their extensions as discussed in Section 4.1 and the main results of this work in Section 3.3. A detailed mathematical treatment incorporating both definitions of the selection operator given in lines (33) and (49) can be found in [57,59,60]. In particular, [56, Theorems 8.2, 8.3] and its extension discussed in Section 4.1 shows ergodicity but non-convergence to global optima for the VLGA with strictly positive mutation limit which includes the case of the simple VLGA. On the other hand, [56, Theorems 8.5, 8.6] and their extensions discussed in Section 4.1 as well as Theorem 3.3.2, Corollaries 3.3.3 and 3.3.4 show convergence to global optima of the appropriately scaled VLGA.

5. Appendix: examples for stochastic matrices modeling genetic operators

This appendix is included here in order to address the concerns of one of the referees and possibly to ease accessibility to the overall mathematical framework used in this exhibition. In what follows, we shall explicitly list the setup via tensor-string representation of populations and corresponding stochastic matrices that model some of the genetic operators discussed above. We do this for the smallest reasonable setting. In fact, we set $\alpha = 2$ and, consequently, $\mu_0 = 1$, $\ell = 2$ and $s = 2$. Then $L = 4$.

5.1. Tensor-string representation of populations

Let the alphabet be given by $\mathscr{A} = \{\hat{\mathrm{o}}, \hat{|}\}$. Then

$$\mathscr{V}_1 = \mathbf{C} \cdot \hat{\mathrm{o}} + \mathbf{C} \cdot \hat{|} = \{x_0 \cdot \hat{\mathrm{o}} + x_1 \cdot \hat{|} : x_{0,1} \in \mathbf{C}\} = \mathbf{C}^2 = \mathbf{C}^\alpha. \tag{50}$$

Here, base vector $\hat{\mathrm{o}} \in \mathscr{V}_1$ is identified with $(1,0)^* \in \mathbf{C}^2$ and base vector $\hat{|} \in \mathscr{V}_1$ is identified with $(0,1)^* \in \mathbf{C}^2$. We have $4 = \alpha^\ell$ possible creatures: $(\hat{\mathrm{o}}, \hat{\mathrm{o}})$, $(\hat{\mathrm{o}}, \hat{|})$, $(\hat{|}, \hat{\mathrm{o}})$ and $(\hat{|}, \hat{|})$.

And we have the following $16=\alpha^L$ populations:

$$\begin{aligned}
p_0 &= ((\hat{\mathrm{o}},\hat{\mathrm{o}}),(\hat{\mathrm{o}},\hat{\mathrm{o}})); & p_1 &= ((\hat{\mathrm{o}},\hat{\mathrm{o}}),(\hat{\mathrm{o}},\hat{|})); \\
p_2 &= ((\hat{\mathrm{o}},\hat{\mathrm{o}}),(\hat{|},\hat{\mathrm{o}})); & p_3 &= ((\hat{\mathrm{o}},\hat{\mathrm{o}}),(\hat{|},\hat{|})); \\
p_4 &= ((\hat{\mathrm{o}},\hat{|}),(\hat{\mathrm{o}},\hat{\mathrm{o}})); & p_5 &= ((\hat{\mathrm{o}},\hat{|}),(\hat{\mathrm{o}},\hat{|})); \\
p_6 &= ((\hat{\mathrm{o}},\hat{|}),(\hat{|},\hat{\mathrm{o}})); & p_7 &= ((\hat{\mathrm{o}},\hat{|}),(\hat{|},\hat{|})); \\
p_8 &= ((\hat{|},\hat{\mathrm{o}}),(\hat{\mathrm{o}},\hat{\mathrm{o}})); & p_9 &= ((\hat{|},\hat{\mathrm{o}}),(\hat{\mathrm{o}},\hat{|})); \\
p_{10} &= ((\hat{|},\hat{\mathrm{o}}),(\hat{|},\hat{\mathrm{o}})); & p_{11} &= ((\hat{|},\hat{\mathrm{o}}),(\hat{|},\hat{|})); \\
p_{12} &= ((\hat{|},\hat{|}),(\hat{\mathrm{o}},\hat{\mathrm{o}})); & p_{13} &= ((\hat{|},\hat{|}),(\hat{\mathrm{o}},\hat{|})); \\
p_{14} &= ((\hat{|},\hat{|}),(\hat{|},\hat{\mathrm{o}})); & p_{15} &= ((\hat{|},\hat{|}),(\hat{|},\hat{|})).
\end{aligned}$$

p_0, p_5, p_{10} and p_{15} are uniform populations. A general element of $\mathscr{V}_\wp=\mathbf{C}^{16}$ is now simply given by

$$\begin{aligned}
v &= v_0 p_0 + v_1 p_1 + v_2 p_2 + v_3 p_3 + v_4 p_4 + v_5 p_5 + v_6 p_6 + v_7 p_7 \\
&\quad + v_8 p_8 + v_9 p_9 + v_{10} p_{10} + v_{11} p_{11} + v_{12} p_{12} + v_{13} p_{13} + v_{14} p_{14} + v_{15} p_{15} \\
&= (v_0, v_1, \ldots, v_{15})^*, \qquad (51)
\end{aligned}$$

where $v_k\in\mathbf{C}$, $0\leqslant k\leqslant 15$. Note that the summation in line (51) understands the p_k as pure symbols. In particular, no additions or other operations involving the four components of p_k in $\mathscr{A}$ are performed. The latter identity in line (51) is by canonical identification $\mathscr{V}_\wp=\mathbf{C}^{16}$. Thus, e.g., $p_0\in\mathscr{V}_\wp$ is canonically identified with $(1,0,\ldots,0)^*\in\mathbf{C}^{16}$. By definition, we have $\langle p_i, p_k\rangle=\delta_{i,k}$, $0\leqslant i,k\leqslant 15$, and the inner product $\langle\cdot,\cdot\rangle$ is conjugate-linear in the first argument. We have, $e=\frac{1}{16}\sum_{k=0}^{15} p_k\in\mathscr{S}_\wp$. An arbitrary element of $\mathscr{U}$ is given by

$$v_0 p_0 + v_5 p_5 + v_{10} p_{10} + v_{15} p_{15} \in \mathscr{U}. \qquad (52)$$

In particular, $p_0 = p_0 + 0\,p_5 + 0\,p_{10} + 0\,p_{15}\in\mathscr{U}$. The probability vector which represents equal probability over only uniform populations is given by

$$\frac{1}{4}\,p_0 + \frac{1}{4}\,p_5 + \frac{1}{4}\,p_{10} + \frac{1}{4}\,p_{15} \in \mathscr{S}_\wp \cap \mathscr{U}. \qquad (53)$$

We note that $p_k\in\mathscr{V}_\wp$, since p_k is a base vector in $\mathscr{V}_\wp$, $0\leqslant k\leqslant 15$. Thus, the notations $\wp\subset\mathscr{V}_\wp$, $\wp\cap\mathscr{U}=\{p_0, p_5, p_{10}, p_{15}\}$ and $\wp\backslash\mathscr{U}$ are justified.

In regard to the tensor-product representation of $\mathscr{V}_\wp=\mathscr{V}_1\otimes\mathscr{V}_1\otimes\mathscr{V}_1\otimes\mathscr{V}_1$ stated in line (7), we list the following examples of tensor representations of populations *and* base vectors of $\mathscr{V}_\wp$ in terms of base vectors $\hat{\mathrm{o}},\hat{|}\in\mathscr{V}_1$:

$$p_0 = (\hat{\mathrm{o}})\otimes(\hat{\mathrm{o}})\otimes(\hat{\mathrm{o}})\otimes(\hat{\mathrm{o}}) = \hat{\mathrm{o}}\otimes\hat{\mathrm{o}}\otimes\hat{\mathrm{o}}\otimes\hat{\mathrm{o}} \quad\text{and}\quad p_{11} = \hat{|}\otimes\hat{\mathrm{o}}\otimes\hat{|}\otimes\hat{|}. \qquad (54)$$

Let us compute GFV_u for some populations

$$\mathrm{GFV}_u(p_0) = (\hat{\mathsf{o}}, \hat{\mathsf{o}}) \in \mathscr{V}_1 \times \mathscr{V}_1 = \mathscr{V}_1^2, \quad \mathrm{GFV}_u(p_{11}) = \left(\hat{|}, \frac{1}{2}\,\hat{\mathsf{o}} + \frac{1}{2}\,\hat{|}\right). \tag{55}$$

Any linear operator X acting on $\mathscr{V}_\wp$ is given by a 16×16 matrix $(X_{i,k})_{i,k=0}^{15}$ such that $X_{i,k} = \langle p_i, X p_k \rangle$.

5.2. Some auxiliary matrices

We note that the matrix P_e is given in the setting of the present example by the 16×16 matrix with constant entries $\frac{1}{16}$. $P_{\mathscr{U}}$ is given by the diagonal 16×16 matrix whose diagonal equals $(1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1)$.

5.3. Example for the mutation matrix

As discussed in Section 2.1, line (10), the spot mutation matrix $\mathbf{m}_{\mu_0}^{(1)}$ is given by the unitary, stochastic flip matrix $\mathbf{f}$ that acts on $\mathscr{V}_1 = \mathbf{C}^2$. We can assume any μ_0 or fix $\mu_0 = 1$.

Let, as usual, μ denote the mutation rate and let $\hat{\mu} = 1 - \mu$. The following computation shows the determination of the $(0,2)$- and $(2,11)$-coefficients of the mutation matrix $M_{1,\mu}^{(m)}$ in accordance with Proposition 2.2.2.1:

$$\begin{aligned} \langle p_0, M_{1,\mu}^{(m)} p_2 \rangle &= \mu^{\Delta(p_0,p_2)} \cdot \hat{\mu}^{L-\Delta(p_0,p_2)} = \mu \cdot \hat{\mu}^3 \quad \text{and} \\ \langle p_2, M_{1,\mu}^{(m)} p_{11} \rangle &= \mu^{\Delta(p_2,p_{11})} \cdot \hat{\mu}^{L-\Delta(p_2,p_{11})} = \mu^2 \cdot \hat{\mu}^2. \end{aligned} \tag{56}$$

Altogether, $M_{1,\mu}^{(m)}$ is given as follows:

$$\left(\begin{array}{cccc|cccc|cccc|cccc}
\hat{\mu}^4 & \mu\hat{\mu}^3 & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu^3\hat{\mu} & \mu^4 \\
\mu\hat{\mu}^3 & \hat{\mu}^4 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^4 & \mu^3\hat{\mu} \\
\mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \hat{\mu}^4 & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu^4 & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} \\
\mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu\hat{\mu}^3 & \hat{\mu}^4 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^4 & \mu^3\hat{\mu} & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 \\
\hline
\mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \hat{\mu}^4 & \mu\hat{\mu}^3 & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu^3\hat{\mu} & \mu^4 & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} \\
\mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \hat{\mu}^4 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^4 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 \\
\mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \hat{\mu}^4 & \mu\hat{\mu}^3 & \mu^3\hat{\mu} & \mu^4 & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 \\
\mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu\hat{\mu}^3 & \hat{\mu}^4 & \mu^4 & \mu^3\hat{\mu} & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 \\
\hline
\mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu^3\hat{\mu} & \mu^4 & \hat{\mu}^4 & \mu\hat{\mu}^3 & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} \\
\mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^4 & \mu^3\hat{\mu} & \mu\hat{\mu}^3 & \hat{\mu}^4 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 \\
\mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu^4 & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \hat{\mu}^4 & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 \\
\mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^4 & \mu^3\hat{\mu} & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu\hat{\mu}^3 & \hat{\mu}^4 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 \\
\hline
\mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu^3\hat{\mu} & \mu^4 & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \hat{\mu}^4 & \mu\hat{\mu}^3 & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 \\
\mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^4 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \hat{\mu}^4 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 \\
\mu^3\hat{\mu} & \mu^4 & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \hat{\mu}^4 & \mu\hat{\mu}^3 \\
\mu^4 & \mu^3\hat{\mu} & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^3\hat{\mu} & \mu^2\hat{\mu}^2 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu^2\hat{\mu}^2 & \mu\hat{\mu}^3 & \mu\hat{\mu}^3 & \hat{\mu}^4
\end{array}\right).$$

Note that if $M_{1,\mu}^{(m)}$ is seen as a 2×2 matrix of 8×8 matrices, then the *same* 8×8 matrix M_8 is multiplied by $\hat{\mu}$ for the $(0,0)$- and $(1,1)$-components and by μ for the $(0,1)$- and $(1,0)$-components of $M_{1,\mu}^{(m)}$. This corresponds to one of the tensor-factors $((1-\mu)\mathbf{1}+\mu\mathbf{m}_{\mu_0}^{(1)})=\hat{\mu}\mathbf{1}+\mu\mathbf{f}$ occurring in Proposition 2.2.2.2. The matrix M_8 is given by the last tensor factor in the formula stated in line (57) below. In fact, we have the following tensor-product decomposition of $M_{1,\mu}^{(m)}$:

$$
\begin{aligned}
M_{1,\mu}^{(m)} &= (\hat{\mu}\mathbf{1}+\mu\mathbf{f})\otimes M_8\\
&= \begin{pmatrix} \hat{\mu} & \mu \\ \mu & \hat{\mu} \end{pmatrix} \otimes \left(\begin{array}{cccc|cccc}
\hat{\mu}^3 & \mu\hat{\mu}^2 & \mu\hat{\mu}^2 & \mu^2\hat{\mu} & \mu\hat{\mu}^2 & \mu^2\hat{\mu} & \mu^2\hat{\mu} & \mu^3 \\
\mu\hat{\mu}^2 & \hat{\mu}^3 & \mu^2\hat{\mu} & \mu\hat{\mu}^2 & \mu^2\hat{\mu} & \mu\hat{\mu}^2 & \mu^3 & \mu^2\hat{\mu} \\
\mu\hat{\mu}^2 & \mu^2\hat{\mu} & \hat{\mu}^3 & \mu\hat{\mu}^2 & \mu^2\hat{\mu} & \mu^3 & \mu\hat{\mu}^2 & \mu^2\hat{\mu} \\
\mu^2\hat{\mu} & \mu\hat{\mu}^2 & \mu\hat{\mu}^2 & \hat{\mu}^3 & \mu^3 & \mu^2\hat{\mu} & \mu^2\hat{\mu} & \mu\hat{\mu}^2 \\
\hline
\mu\hat{\mu}^2 & \mu^2\hat{\mu} & \mu^2\hat{\mu} & \mu^3 & \hat{\mu}^3 & \mu\hat{\mu}^2 & \mu\hat{\mu}^2 & \mu^2\hat{\mu} \\
\mu^2\hat{\mu} & \mu\hat{\mu}^2 & \mu^3 & \mu^2\hat{\mu} & \mu\hat{\mu}^2 & \hat{\mu}^3 & \mu^2\hat{\mu} & \mu\hat{\mu}^2 \\
\mu^2\hat{\mu} & \mu^3 & \mu\hat{\mu}^2 & \mu^2\hat{\mu} & \mu\hat{\mu}^2 & \mu^2\hat{\mu} & \hat{\mu}^3 & \mu\hat{\mu}^2 \\
\mu^3 & \mu^2\hat{\mu} & \mu^2\hat{\mu} & \mu\hat{\mu}^2 & \mu^2\hat{\mu} & \mu\hat{\mu}^2 & \mu\hat{\mu}^2 & \hat{\mu}^3
\end{array}\right). \qquad (57)
\end{aligned}
$$

This can be continued to obtain the tensor-product decomposition of $M_{1,\mu}^{(m)}$ in Proposition 2.2.2.2. To illustrate how Proposition 2.2.2.2 and the formulas in lines (18), (20), (22) and (25) can be checked, let us repeat the computation carried out in line (58) using the tensor-product decomposition of $M_{1,\mu}^{(m)}$, cf. Proposition 2.2.2.2. Then we have

$$
\begin{aligned}
&\langle \hat{\mathtt{o}}\otimes\hat{\mathtt{o}}\otimes\hat{\mathtt{o}}\otimes\hat{\mathtt{o}}, ((\hat{\mu}\mathbf{1}+\mu\mathbf{f})\otimes(\hat{\mu}\mathbf{1}+\mu\mathbf{f})\otimes(\hat{\mu}\mathbf{1}+\mu\mathbf{f})\otimes(\hat{\mu}\mathbf{1}+\mu\mathbf{f}))(\hat{\mathtt{o}}\otimes\hat{\mathtt{o}}\otimes\hat{|}\otimes\hat{\mathtt{o}})\rangle\\
&\quad = \langle \hat{\mathtt{o}}, (\hat{\mu}\mathbf{1}+\mu\mathbf{f})\hat{\mathtt{o}}\rangle\cdot\langle\hat{\mathtt{o}}, (\hat{\mu}\mathbf{1}+\mu\mathbf{f})\hat{\mathtt{o}}\rangle\cdot\langle\hat{\mathtt{o}}, (\hat{\mu}\mathbf{1}+\mu\mathbf{f})\hat{|}\rangle\cdot\langle\hat{\mathtt{o}}, (\hat{\mu}\mathbf{1}+\mu\mathbf{f})\hat{\mathtt{o}}\rangle\\
&\quad = \hat{\mu}^2\cdot\mu\cdot\hat{\mu}
\end{aligned}
$$

and

$$
\begin{aligned}
&\langle \hat{\mathtt{o}}\otimes\hat{\mathtt{o}}\otimes\hat{|}\otimes\hat{\mathtt{o}}, ((\hat{\mu}\mathbf{1}+\mu\mathbf{f})\otimes(\hat{\mu}\mathbf{1}+\mu\mathbf{f})\otimes(\hat{\mu}\mathbf{1}+\mu\mathbf{f})\otimes(\hat{\mu}\mathbf{1}+\mu\mathbf{f}))(\hat{|}\otimes\hat{\mathtt{o}}\otimes\hat{|}\otimes\hat{|})\rangle\\
&\quad = \langle \hat{\mathtt{o}}, (\hat{\mu}\mathbf{1}+\mu\mathbf{f})\hat{|}\rangle\cdot\langle\hat{\mathtt{o}}, (\hat{\mu}\mathbf{1}+\mu\mathbf{f})\hat{\mathtt{o}}\rangle\cdot\langle\hat{|}, (\hat{\mu}\mathbf{1}+\mu\mathbf{f})\hat{|}\rangle\cdot\langle\hat{\mathtt{o}}, (\hat{\mu}\mathbf{1}+\mu\mathbf{f})\hat{|}\rangle\\
&\quad = \mu\cdot\hat{\mu}^2\cdot\mu. \qquad (58)
\end{aligned}
$$

5.4. Examples for the crossover matrix

We shall list below the elementary one-cutpoint crossover matrices $C(1;1)$ and $C(1;2)$ in regard to the example setting established in Section 5.1 above and Definition 2.4.1. The reader may then easily synthesize the crossover operators $\bar{C}(1)=\frac{1}{2}(C(1;1)+C(1;2))$ in the sense of line (16) and $C_{\mathrm{reg}}^{(1)})(\chi)$ in the sense of Definition 2.4.2 and line (18).

The elementary one-cutpoint crossover matrix $C(1;1)$ which is unitary and stochastic is given as follows:

$$\left(\begin{array}{cccc|cccc|cccc|cccc}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
\hline
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
\hline
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
\hline
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{array}\right).$$

Similarly, the elementary one-cutpoint crossover matrix $C(1;2)$ is given as follows:

$$\left(\begin{array}{cccc|cccc|cccc|cccc}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
\hline
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
\hline
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
\hline
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{array}\right).$$

5.5. Example for the 'take-the-best' selection matrix

We finally note that for the fitness function defined in [56, p. 55, Section 8.3, Example 1] the *take-the-best* selection matrix $\lim_{t\to\infty} S_t^f$ is listed and permanently available online in a Mathematica notebook in [54].

6. Conclusion

The main results of this exposition, Theorem 3.3.2, Corollaries 3.3.3 and 3.3.4 achieve the following goals:

1. A general-purpose, scaled genetic algorithm is described that converges asymptotically to global optima. There are no special requirements in regard to the fitness function or the fitness landscape. The size s of populations that are strings over an arbitrary-size alphabet can stay small and can be set as low as $\ell+1$ where ℓ is the length of the genome of creatures (candidate solutions).
2. Explicit schedules for scaled mutation, crossover and selection are given as follows:
   - Annealing schedule for the mutation rate $\mu(t)$, $t\in\mathbf{N}\cap[t_0,\infty)$, for multiple-spot mutation.
     $\mu(t)=\phi_M\cdot t^{-1/(\kappa_0\ell_s)}<\frac{1}{2}$, where $t_0\in\mathbf{N}$, $\phi_M\in\mathbf{R}_*^+$, $\kappa_0\in[1,\infty)$ can be chosen (see Theorem 3.2.1).
   - Annealing schedule for the crossover rate χ_t.
     Rational generalized crossover: $\chi_t=\phi_C\mu(t)^{\kappa_0\ell+1}$, where $\phi_C\in(0,2^{\kappa_0\ell+1}]$ can be chosen (see Theorem 3.3.2).
     Regular pair-wise crossover, or gene-lottery crossover: $\chi_t=\phi_C\mu(t)^{1/m}$ where both $\phi_C\in(0,2^{1/m}]$ and $m\in[1,\infty)$ can be chosen (see Corollaries 3.3.3 and 3.3.4).
   - Exponentiation schedules $f^{g(t)}$ for the fitness function f and scaling of the corresponding proportional fitness selection operator S_t^f.
     $g(t)=B\log(t-t_0+2)$ where $1<sB\log(\rho_2(f))$ and $B>0$ can be chosen.
     $\rho_2(f)$ given in line (30) is easy to determine, if fitness selection is based upon scaled, i.e., exponentiated rank.
3. The genetic algorithm presented in this exposition is well suited for optimization in a compact domain of $\mathbf{R}^k$, $k\in\mathbf{N}$, in particular, since mutation incorporates a neighborhood-based search.
4. The genetic algorithm presented in this exposition satisfies *all* goals (1)–(4) formulated by Davis and Principe in [18, p. 270] and is much in the spirit of the simulated annealing algorithm.
5. The two major shortcomings of [56, Theorem 8.6, Remark 8.7] are removed: mutation need not commute with crossover anymore and the fitness function need not have a sole maximum.

The extension of the theory of genetic algorithms [56] presented in this exposition contains besides the above strong convergence results the following new elements: improved spectral estimates for one-cutpoint regular crossover (see Section 2.4) which show one-to-one correspondence with results in Koehler's Theorem [31, p. 419]; iden-

tities for stochastic matrices that model two-cutpoint and uniform regular crossover and act on the free vector space over the tensor-string representation for populations which allows for simple verification of commutation relation with mutation and spectral estimates (see Section 2.4); an example of an asymmetric crossover matrix that does not commute with mutation (see Section 2.5); extension of various results of [56] to the situation of asymmetric crossover matrices (see Section 4.1); embedding of the Vose–Liepins crossover-mutation method into the model based upon the tensor-string representation for populations (see Section 4.3); a simpler approach to the mutation flow equation (see Proposition 2.2.3); a simplified and more general approach to weak ergodicity based upon annealing the mutation rate (see Theorem 3.2.1); and, a simplified and strengthened approach to convergence to uniform populations of genetic algorithms in the zero mutation-rate limit (see Theorem 3.1.1)).

Acknowledgements

- The main result of this work and [56] are the foundation of a presentation "Theoretical Aspects of Genetic Algorithms" by the author in a recent seminar "*Theory of Evolutionary Computation 2002*" at the Max Planck Institute for Computer Science Conference Center, Schloß Dagstuhl, Saarland, Germany. The author wishes to thank the organizers H.G. Beyer (Universität Dortmund), K. De Jong (George Mason University), C. Reeves (University of Coventry) and I. Wegener (Universität Dortmund) for the opportunity to participate in the event which yielded an enriched presentation in this exhibition.
- The author also wishes to thank the two referees for carefully reviewing the manuscript.

References

[1] E.H.L. Aarts, P.J.M. van Laarhoven, Simulated annealing: an introduction, Statist. Neerlandica 43 (1989) 31–52.

[2] A. Agapie, Theoretical analysis of mutation-adaptive evolutionary algorithms, Evol. Comput. 9 (2001) 127–146.

[3] S. Anily, A. Federgruen, Ergodicity in parametric non-stationary markov chains: an application to simulated annealing methods, Oper. Res. 35 (1987) 867–874.

[4] H. Aytug, G.J. Koehler, Stopping criteria for finite length genetic algorithms, INFORMS J. Comput. 8 (1996) 183–191.

[5] J.E. Baker, Reducing bias and inefficiency in the selection algorithm, in: J.J. Grefenstette (Ed.), Genetic Algorithms and Their Applications: Proc. Second Internat. Conf. on Genetic Algorithms, Lawrence Erlbaum Publishers, Mahwah, NJ, USA, 1987.

[6] W. Banzhaf, F.D. Francone, P. Nordin, The effect of extensive use of the mutation operator on generalization in genetic programming using sparse data sets, in: H.-M. Voigt, et al. (Eds.), Proc. Internat. Conf. on Evolutionary Computation and Fourth Conf. Parallel Problem Solving from Nature (PPSN IV), Berlin, Germany, September 1996, Springer, Berlin, Germany, 1996, pp. 300–309.

[7] S. Bhattacharyya, G.J. Koehler, An analysis of non-binary genetic algorithms with cardinality 2^v, Complex Systems 8 (1994) 227–256.

[8] H.J. Bremermann, J. Rogson, S. Salaff, Global Properties of Evolution Processes, in: H.H. Pattee, et al. (Eds.), Natural Automata and Useful Simulations, Spartan Books, Washington, DC, USA, 1966, pp. 3–42.
[9] C.L. Bridges, D.E. Goldberg, An analysis of reproduction and crossover in a binary-coded genetic algorithm, in: J.J. Grefenstette (Ed.), Genetic Algorithms and Their Applications: Proc. of the Second Internat. Conf. on Genetic Algorithms, Lawrence Erlbaum, Mahwah, NJ, USA, 1987, pp. 9–13.
[10] O. Catoni, Large deviations for annealing, Ph.D. Dissertation, Université Paris XI, 1990.
[11] O. Catoni, Sharp large deviations estimates for simulated annealing algorithms, Ann. Inst. H. Poincaré. Probab. Statist. 27 (1991) 291–383.
[12] O. Catoni, Applications of sharp large deviations estimates to optimal cooling schedules, Ann. Inst. H. Poincaré. Probab. Statist. 27 (1991) 493–518.
[13] O. Catoni, Rough large deviations estimates for simulated annealing—application to exponential schedules, Ann. Probab. 20 (1992) 1109–1146.
[14] R. Cerf, An asymptotic theory for genetic algorithms. in: J.M. Alliot, et al. (Eds.), Artificial Evolution European Conf. AE '95, Brest, France, September 4–6, 1995, Selected Papers, Lecture Notes in Computer Science, Vol. 1063, Springer, Berlin, Germany, 1996, pp. 37–53.
[15] R. Cerf, Asymptotic convergence of genetic algorithms, Adv. Appl. Probab. 30 (1998) 521–550.
[16] T.E. Davis, Toward an extrapolation of the simulated annealing convergence theory onto the simple genetic algorithm, Ph.D. Dissertation, University of Florida, 1991.
[17] 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. Fourth Internat. Conf. on Genetic Algorithms '91, Morgan Kaufmann Publishers, San Francisco, CA, USA, 1991, pp. 174–181.
[18] T.E. Davis, J.C. Principe, A Markov chain framework for the simple genetic algorithm, Evol. Comput. 1 (1993) 269–288.
[19] K. De Jong, Lecture on coevolution, in: H.-G. Beyer, et al. (Chairs), Theory of Evolutionary Computation 2002, Max Planck Institute for Computer Science Conference Center, Schloß Dagstuhl, Saarland, Germany, 2002.
[20] D.B. Fogel, Asymptotic convergence properties of genetic algorithms and evolutionary programming: analysis and experiments, Cybernet. Systems 25 (1994) 389–407.
[21] H. Geiringer, On the probability theory of linkage in mendelian heredity, Ann. of Math. Statist. 15 (1944) 25–57.
[22] D.E. Goldberg, Simple genetic algorithms and the minimal deceptive problem, in: L. Davis (Ed.), Genetic Algorithms and Simulated Annealing, Hyperion Books, New York, NY, USA, Morgan Kaufmann, San Francisco, CA, USA, 1987, pp. 74–88.
[23] D.E. Goldberg, Genetic Algorithms, in Search, Optimization & Machine Learning, Addison-Wesley, Boston, MA, USA, 1989.
[24] D.E. Goldberg, A note on Boltzmann tournament selection for genetic algorithms and population oriented simulated annealing, Complex Systems 4 (1990) 445–460.
[25] D.E. Goldberg, K. Deb, A comparative analysis of selection schemes used in genetic algorithms, in: G.J.E. Rawlins (Ed.), Foundations of Genetic Algorithms, Morgan Kaufmann, San Francisco, CA, USA, 1991, pp. 69–93.
[26] G.W. Greenwood, Q.J. Zhu, Convergence in evolutionary programs with self-adaptation, Evol. Comput. 9 (2001) 147–157.
[27] R.C. Griffiths, S. Taveré, Computational methods for the coalescent, in: P. Donnelly, S. Taveré (Eds.), Progress in Population Genetics and Human Evolution, IMA Volumes in Mathematics and its Applications, Vol. 87, Springer, Berlin, Germany, 1997, pp. 165–182.
[28] J. He, L. Kang, On the convergence rates of genetic algorithms, Theoret. Comput. Sci. 229 (1999) 23–39.
[29] J.H. Holland, Adaptation in natural and artificial systems, University of Michigan Press, Ann Arbor, MI, USA, 1975 (extended new edition, MIT Press, Cambridge, MA, USA, 1992).
[30] D.L. Isaacson, R.W. Madsen, Markov chains: theory and applications, Prentice-Hall, Upper Saddle River, NJ, USA, 1961.
[31] G.J. Koehler, A proof of the Vose–Liepins conjecture, Ann. Math. Artif. Intell. 10 (1994) 408–422.

[32] G.J. Koehler, S. Bhattacharyya, M.D. Vose, General cardinality genetic algorithms, Evol. Comput. 5 (1998) 439–459.
[33] S. Lang, Linear Algebra, 2nd Ed., Addison-Wesley, Boston, MA, USA, 1970.
[34] S. Lang, Complex Analysis, Addison-Wesley, Boston, MA, USA, 1977.
[35] Y. Leung, Z.-P. Chen, Z.-B. Xu, K.-S. Leung, Convergence rate for non-binary genetic algorithms with different crossover operators. The Chinese University of Hong Kong, preprint, 1998.
[36] J.A. Lozano, P. Larrañaga, M. Graña, F.X. Albizuri, Genetic algorithms: bridging the convergence gap, Theoret. Comput. Sci. 229 (1999) 23–39.
[37] S.W. Mahfoud, Finite Markov chain models of an alternative selection strategy for genetic algorithms, Complex Systems 7 (1993) 155–170.
[38] S.W. Mahfoud, D.E. Goldberg, A genetic algorithm for parallel simulated annealing, in: R. Männer, B. Manderick (Eds.), Parallel Problem Solving from Nature, Vol. 2, Elsevier, Amsterdam, The Netherlands, 1992, pp. 301–310.
[39] S.W. Mahfoud, D.E. Goldberg, Parallel recombinative simulated annealing: a genetic algorithm, Parallel Comput. 21 (1995) 1–28.
[40] A. Márkus, G. Renner, J. Vanza, Spline interpolation with genetic algorithms, in: T.L. Kunii, et al. (Chairs), Proc. Internat. Conf. Shape Modeling, The University of Aizu, Aizu-Wakamatsu, Japan, March 3–6, 1997, IEEE Computer Society Press, Washington, DC, USA, 1997, pp. 47–54.
[41] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, 2nd, extended ed., Springer, Berlin, Germany, 1994.
[42] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Cambridge, MA, USA, 1996.
[43] H. Mühlenbein, Genetic algorithms, in: E. Aarts, L. Lenstra (Eds.), Local Search in Combinatorial Optimization, Wiley Publishers, Hoboken, NJ, USA, 1997, pp. 131–171.
[44] A.E. Nix, M.D. Vose, Modeling genetic algorithms with Markov chains, Ann. Math. Artif. Intel. 5 (1992) 79–88.
[45] T. Nomura, K. Shimohara, An analysis of two-parent recombinations for real-valued chromosomes in an infinite population, Evol. Comput. 9 (2001) 283–308.
[46] R. Poli, Exact schema theory for genetic programming and variable-length genetic algorithms with one-point crossover, Genetic Programming Evol. Mach. 2 (2001) 123–163.
[47] A. Poli, M. Langdon, Schema theory for genetic programming with one-point crossover and point mutation, Evol. Comput. 6 (3) (1998) 231–252.
[48] I. Rechenberg, Evolutionsstrategie. Optimierung technischer Systeme nach Prinzipien der biologischen Evolution, Frommann-Holzboog Verlag, Stuttgart, Germany, 1973.
[49] W. Rudin, Functional Analysis, McGraw-Hill, New York, NY, USA, 1973.
[50] G. Rudolph, Convergence analysis of canonical genetic algorithms, IEEE Trans. Neural Networks 5 (1994) 96–101.
[51] G. Rudolph, An evolutionary algorithm for integer programming, in: Y. Davidor, et al. (Eds.), Proc. Third Internat. Conf. Parallel Problem Solving From Nature (PPSN III), Springer, Berlin, Germany, 1994, pp. 139–148.
[52] G. Rudolph, Finite Markov chain results in evolutionary computation: a tour d'Horizon, Fund. Inform. 35 (1998) 67–89.
[53] H.H. Schaefer, Banach lattices and positive operators, in: Die Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen, Vol. 215, Springer, Berlin, Germany, 1974.
[54] L.M. Schmitt, Mathematica Computation, ftp://ftp.u-aizu.ac.jp/u-aizu/doc/Tech-Report/1999/99-2-004.tar.gz.
[55] L.M. Schmitt, Theory of genetic algorithms, Tech. Report 2000-2-002, The University of Aizu, Aizu-Wakamatsu, Japan, 2000, pp. 1–61.
[56] L.M. Schmitt, Theory of genetic algorithms, Theoret. Comput. Sci. 259 (2001) 1–61.
[57] L.M. Schmitt, Asymptotic convergence of scaled genetic algorithms to global optima—a gentle introduction to the theory—, in: A. Menon (Ed.), Computing Horizons: the Hilbert Challenge Essays on the Frontiers of Evolutionary Computation, in preparation; D.E. Goldberg (Ed.), Kluwer Series in Evolutionary Computation, Kluwer, Dordrecht, The Netherlands, 2003, pp. 178–221.
[58] L.M. Schmitt, Coevolutionary convergence to global optima, Technical Report 2003-2-002, The University of Aizu, Aizu-Wakamatsu, Japan, 2003, pp. 1–12, in: Erick Cantu-Paz, et al. (Eds.), Proc.

Genetic and Evolutionary Computation Conf. GECCO 2003 (coevolution track: poster), The Holiday Inn, Chicago, IL, USA, July 12–16, 2003, Lecture Notes in Computer Science, Vol. 2723, Springer, Berlin, Germany, 2003, pp. 375–376.

[59] L.M. Schmitt, Optimization with genetic algorithms in multi-species environments, in: L. Jiao, et al. (Eds.), Proc. Fifth Internat. Conf. Computation Intelligence and Multimedia Applications (ICCIMA 2003), Xidian University, Xi'an, China, September 27–30, 2003, Sponsored by IEEE, The Printing House (publisher), Quincy, FL, USA, pp. 194–199, ISBN 0–7695–1957–1.

[60] L.M. Schmitt, Theory of coevolutionary genetic algorithms, in: M. Guo, L.T. Yang (Eds.), Proc. of 'The 2003 Internat. Symp. Parallel and Distributed Processing and Applications' ISPA 2003, The University of Aizu, Aizu-Wakamatsu, Japan, July 2–4, 2003, Sponsored by ACM and IPSJ, Lecture Notes in Computer Science, Vol. 2745, Springer, Berlin, Germany, 2003, pp. 285–293.

[61] 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, in: C.L. Nehaniv (Ed.), Mathematical and Computational Biology: Computational Morphogenesis, Hierarchical Complexity and Digital Evolution, An International Workshop, The University of Aizu, Aizu-Wakamatsu, Japan, October 21–25, 1997, Lectures on Mathematics in the Life Sciences Series, Vol. 26, American Mathematical Society, Providence, RI, USA, 1999, pp. 147–166.

[62] L.M. Schmitt, C.L. Nehaniv, R.H. Fujii, Linear analysis of genetic algorithms, Theoret. Comput. Sci. 200 (1998) 101–134.

[63] E. Seneta, Non-Negative Matrices and Markov Chains, Springer Series in Statistics, Springer, Berlin, Germany, 1981.

[64] C.R. Stephens, H. Waelbroeck, Effective degrees of freedom in genetic algorithms and the block hypothesis, in: T. Back (Ed.), Proc. Seventh Internat. Conf. Genetic Algorithms, Lawrence Erlbaum, Mahwah, NJ, USA, 1997, pp. 34–40.

[65] J. Suzuki, A further result on the Markov chain model of genetic algorithms and its application to a simulated annealing-like strategy, in: R.K. Belew, M.D. Vose (Eds.), Foundations of Genetic Algorithms, Vol. 4, Morgan Kaufmann, San Francisco, CA, USA, 1997, pp. 53–72.

[66] J. Suzuki, A further result on the Markov chain model of genetic algorithms and its application to a simulated annealing-like strategy, IEEE Trans. Systems Man Cybernet. B 28 (1998) 95–102.

[67] M.D. Vose, Random heuristic search, Theoret. Comput. Sci. 229 (1999) 103–142.

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

[69] M.D. Vose, G.E. Liepins, Punctuated equilibria in genetic search, Complex Systems 5 (1991) 31–44.

[70] M.D. Vose, A.H. Wright, The simple genetic algorithm and the Walsh transform: part I, theory, Evol. Comput. 6 (1998) 253–273.

[71] M.D. Vose, A.H. Wright, The simple genetic algorithm and the Walsh transform: part II, the inverse, Evol. Comput 6 (1998) 275–289.

[72] M.D. Vose, A.H. Wright, Form invariance and implicit parallelism, Evol. Comput. 9 (2001) 355–370.

[73] D. Whitley, An executable model of a simple genetic algorithm, in: L.D. Whitley (Ed.), Foundations of Genetic Algorithms, Vol. 2, Morgan Kaufmann, San Francisco, CA, USA, 1993, pp. 45–62.