Feasible Approaches to Convergence Results for Evolutionary Algorithms Part I: Introductory overview and analysis of scaled genetic algorithms

Lothar M. Schmitt[+] and Stefan Droste[*]

[+]School of Computer Science and Engineering, The University of Aizu
Aizu-Wakamatsu City, Fukushima Prefecture 965-8580, Japan
lothar@u-aizu.ac.jp

[*]LS Informatik 2, Universität Dortmund, 44221 Dortmund, Germany
stefan.droste@udo.edu

Abstract. Despite many successes of evolutionary algorithms (EAs) in real-world applications, theoretical knowledge in regard to these algorithms is still in its infancy. In this work, we discuss a number of approaches to theory for EAs in regard to strengths and weaknesses of statements for convergence-speed obtained with these methods. This includes the general convergence-analysis of a broad class of EAs in an arbitrary-fitness-function black-box scenario similar to the setting for the simulated annealing algorithm, and the runtime-analysis of specific EAs on limited classes of fitness-functions within the framework of asymptotic runtime-analysis for randomized algorithms.
We propose that a suitable merger of ideas put forward through the latter two types of convergence-analysis may yield substantial progress towards understanding convergence behavior of EAs. In particular, this may yield a unified theoretical framework for EAs as well as probabilistic estimates for runtimes of EAs used in real-world applications.

Introduction

The lack of a comprehensive approach to analysis of EAs. Evolutionary algorithms (EAs) have already a long history of successful application for optimization purposes. See, *e.g.*, [62]. Successful applications of EAs have been especially noticeable and valuable in situations when there exists no exactly solvable mathematical model for the problem at hand, or there is no time to develop such a model and/or a problem-specific algorithm. Although there are many different approaches to develop a theory for EAs, none of the existing

[*] Stefan Droste was supported in part by the Deutsche Forschungsgemeinschaft through the Collaborative Research Center "*Computational Intelligence*" (SFB 531). This work was done while he was visiting The University of Aizu whose financial support is also kindly acknowledged.

theories gives practitioners direct hints or guidelines showing which problems can be successfully tackled by what kind of EA in a manner comparable to the theory of NP-completeness for classical optimization algorithms. (See [22] for an introduction to NP-completeness). In particular, results in [45, 48–52, 54–57] which understand EAs as analogues of the simulated annealing algorithm and analyze asymptotic behavior show a *very slow* convergence of certain classes of EAs in a general black-box scenario. On the other hand, results in [17–19, 21, 29, 23, 65–67] that understand EAs as randomized algorithms and apply results for asymptotic runtime-analysis of such as in [38] show better types of convergence only on *limited classes* of fitness-functions. In addition, research on schemata as described, *e.g.*, in [24, p. 19 *ff.*, p. 40 *ff.*] only delivers results in regard to *one-step improvements* for EAs and no results in regard to finite but larger advance in time or asymptotic behavior. This holds as well for approaches to analysis of EAs based upon Price's theorem [16]. Thus, lack of an EA-theory with direct consequences for practitioners —comparable to the theory of NP-completeness— is possibly the greatest challenge in the field of theory for EAs today.

Biology-inspired yet ordinary randomized algorithms. There are at least two principle points of view regarding the use of EAs: some researchers see them primarily as a means to model and study evolutionary principles in nature, others such as the authors of this work see them primarily as a computational optimization method which is biology-inspired.

The authors of this work believe that biology plays and should play an important role in motivating EAs and finding new ideas for genetic operators or their representation on data-structures. Nevertheless, we regard EAs as ordinary randomized, heuristic optimization algorithms; and we think that, primarily, they should be analyzed as such. Hence, the most important issues are *general convergence* (Is the algorithm designed to find or converge to at least one optimal solution similar to convergence of classical discrete or numerical algorithms?) and specific *runtime analysis* (How many computation steps does the algorithm need to find an optimum with regard to the problem dimension similar to classical runtime-analysis?). In trying to answer these questions, we can and should use methods and techniques developed in established works of convergence analysis for efficient classical deterministic algorithms such as [30–32], the analysis of efficient randomized algorithms such as [38]), and the analysis of finite-length behavior of probabilistic algorithms such as [6–9].

Weaknesses and strengths of existing approaches. For most of the discussion following below, we shall consider EAs as randomized algorithms whose goal is to find globally optimal elements in a set $\mathcal{C}$ of candidate solutions (creatures) in regard to a given *fitness-function* $f: \mathcal{C} \to \mathbf{R}$. As already indicated above, there are a number of general convergence results for EAs in the spirit of convergence analysis for the simulated annealing algorithm (see [14, 15, 45, 48–52, 54–57] among others) and specific runtime analyses for EAs (see [17–19, 21, 29, 23, 46, 65–67] among others). However, particular weaknesses of these re-

sults have so far prevented them from becoming easily applicable to real-world problem settings:

- General convergence results for EAs as mentioned above, usually, do not make any statement in regard to the runtime needed to find an optimum.
- Runtime analyses for EAs as mentioned above are usually limited to specific classes of fitness-functions and specific EAs.

On the other hand, these directions of research have also particular strengths:

- General convergence results for EAs as mentioned above describe EAs that are designed and can be implemented in any situation to find optima asymptotically with probability one.
- Runtime analyses for EAs as mentioned above that are limited to specific classes of fitness-functions and specific EAs gives the practitioner rather pricise estimates for the computational costs one must bear on average in order to find optima.

Hence, the weaknesses and strengths of these two approaches are somewhat complementary which suggests that a merger of the two approaches, or the attempt to close the "gap" between the two approaches holds some promise for theoretical advance.

In addition, one should note that the two approaches discussed above have some strengths in common:

- They do not rely on any hidden or particular assumptions which are difficult to validate, or experimental results whose error is not estimated in the corresponding model.
- They are not limited to the one-step behavior of EAs.
- They yield procedural descriptions of EAs and, thus, have been successfully applied leading to mathematically proven results about the convergence or the runtime, *i.e.*, they are not theories only for their own sake.

In view of the arguments put forward in the above discussion, we believe that it is a promising research goal to find new ways of merging the two approaches discussed above. In this paper, we shall present some ideas how to extend the collection of methods already in use in order to possibly make general convergence results more precise in regard to statement of runtime, and existing runtime analyses possibly applicable to larger classes of fitness-functions and EAs.

Taking into account a lot of work that has been done in algorithm analysis within the last decades (see almost any standard textbook on the subject such as [12, 59]), the authors of this work believe that it is unrealistic to attempt to prove results which are very general in scope but also precise, optimal and easy to apply in regard to runtime. Observe that for the simulated annealing algorithm this point is quite well understood and yields the following, possibly surprising answer: [25] lists explicitly the optimal cooling schedule for simulated annealing in terms of "depth of local minima" of the fitness-landscape, *i.e.*, this determines the fastest possible runtime of the algorithm in general. However, it is mentioned in [1] that determining the "depth of local minima" is as computationally complex as solving the original optimization problem and, thus, *not* easy to apply. Thus in order to achieve our goal, we must obviously weaken the

strengths of the two approaches, *i.e.*, convergence results will be less general while runtime analyses will be not so precise anymore.

The analysis of randomized algorithms and processes requires an array of quite sophisticated mathematical techniques. For a successful analysis, one has to apply estimation and simplification techniques in order to master the probabilistic behavior of the algorithm. See, *e.g.*, estimates in regard to simplification of Markov chain analysis in [18, Lemma 5], or the mutation-flow and steady-state flow inequalities in [52, Prop. 3.1.1] and [51, Prop. 2.2.3, lines 45, 46, 47]. Usually, the difficulty is to find proper estimates and simplifications while still being able to bound the error introduced through them. In this paper, we want to point out some general methods for convergence resp. runtime analysis and possible generalizations that close the "gap" between the two approaches mainly discussed above.

Organization of the remainder of the paper. In the next section, we shall give a short but more detailed overview of existing theoretical approaches for analysis of EAs. In sections I.2 and 3, we shall present a family of basic ideas and methods for general convergence analysis of EAs resp. runtime analysis of specific EAs. The final section lists conclusions from our previous discussion.

I.1. Overview of Existing Approaches

In this section, we shall outline in more detail some of the existing approaches to theory for EAs from a somewhat historical perspective. Analysis of general convergence results (from "outside-in") can be seen as following the historic route of analysis of the simulated annealing algorithm. Runtime analysis for specific EAs (from "inside-out") has been developed by continuously enlarging a repertoire of mathematical techniques studied on simple problem instances which currently yields applicable results for well-known real-world problem instances such as finding a maximum matching in a graph [23]. In the course of the presentation, we emphasize some directions of inquiry that we find especially interesting for future research. The discussion based in this section is an extension of the discussion in [52, Sec. 4].

Theory for simulated annealing. The simulated annealing algorithm was first described in [37]. Similar to the GA, the simulated annealing algorithm was initially investigated in regard to its asymptotic behavior using a finite state, inhomogeneous Markov chain as mathematical framework. See, *e.g.*, [1] for an excellent introduction and overview in regard to the simulated annealing algorithm. Probably the most comprehensive analysis in regard to infinite-length cooling schedules for simulated annealing can be found in [25].

Note at this point that in the case of the simulated annealing algorithm, the steady-state distributions of the individual steps v_t, $t \in \mathbf{N}$, of the algorithm can be determined in a closed mathematical form, *cf.* [1]. This makes establishment of

strong ergodicity of the inhomogeneous Markov chain describing the algorithm[1] quite simple via coordinate-wise monotone behavior of the sequence v_t and use of [27, p. 160: Thm. V.4.3] or [52, Thm. 3.3.2].

Work in [6–9] took analysis of the simulated annealing algorithm to a higher level. Here, it is emphasized to find the "*best possible cooling schedule over a finite time-interval*" for a given finite computational time limit. Incidentally (or not?), this yields the same results as in the case of infinite-length analysis. This work constitutes a major advance in regard to developing probabilistic estimates for finite-length simulated annealing algorithms, *i.e.*, stopping criteria for these algorithms. Analysis for scaled GAs and for general EAs as well should take the approach in [6–9] into account.

References [35, 36, 33] discuss hybrid simulated annealing algorithms with GA-mutation/GA-crossover as mixing/generating procedures. In particular, an asymptotic convergence result for such algorithms to global optima is obtained in [33].

We should point out to the reader that the $(1 + 1)$ EA is the limit of the single steps of a regular simulated annealing algorithm. Consequently in view of [27, p. 160: Thm. V.4.3] or [52, Thm. 3.3.2], some of the analysis developed in [17–19, 21, 66] may have applications to the runtime analysis of the simulated annealing algorithm on certain function classes.

Theory for genetic algorithms. Genetic Algorithms (GAs) constitute a widely used subclass of EAs which have probably been applied and investigated among EAs for the longest period of time. Theoretical description of GAs can be roughly classified in two categories: The first category is characterized by schema-theory following [26] including the variant of building block hypothesis, *cf.* [24, p. 41–45]. The second category is characterized by Markov-chain analysis.

Schema-theory has overall failed to produce general convergence results to global optima for GAs. This is mainly due to the fact that it is limited in its basic form to one-step behavior. This does not say that schema-theory may not be useful in explaining few-step behavior of a GA in an environment where the fitness function is changing over time. An example where a fitness landscape is considered that changes over time depending upon the behavior of the creatures in a population can be found in [34].

Markov chain analysis was initiated notably through work in [63, 39, 13–15]. However, even though it is fairly simple (as demonstrated in [52, Sec. 4] and outlined below) to set up a mathematical model for genetic algorithms based upon Markov chains, it has taken quite some time that non-elementary convergence results with correct proofs appeared in the literature. See, *e.g.*, [10, 11, 50, 51].

If one accepts that the results in [10, 11, 50, 51] at least to some degree settle the question of infinite-length asymptotics for GAs in regard to finding global optima, then the next step is to analyze finite-length GAs in regard to a "*best possible cooling/scaling schedule over a finite time-interval*" following the ideas and the mathematical framework of [6–9]. Combining this with the point of

[1] See [51, Lemma 3.3.1, p. 213: proof of Thm. 3.3.2] for a quite general analogous framework in the context of GAs.

view put forward in work in [17–19, 21, 29, 23, 65–67] as well as [52, Sec. 4], let it be stated that future theoretical research on EAs —and GAs in particular— should primarily deal with finite-length algorithms on finite-state machines and estimates in regard to approaching infinite-length asymptotics and global optima using a probabilistic framework. If one looks at theory of EAs this way, then theory of EAs is certainly at its beginning.

Analysis from outside-in and inside-out. There are, in principle, two major ways to analyze EAs: from *inside-out* and *outside-in*. Following the *outside-in* view, the EA is understood as an all-purpose tool which is used in a black-box scenario, *i.e.*, on a fitness function of largely unknown behavior and characteristics. Thus, this analysis could be characterized as "finding a most general scenario" or an "upper bound envelope" in which convergence results still hold. As depicted in Section I.2 the EAs are seen primarily as an ergodic "cooling procedure" similar to the simulated annealing algorithm setting. The analysis of the simulated annealing algorithm serves as example and inspiration for corresponding analysis of (scaled) EAs within a probabilistic framework that involves analysis of inhomogeneous Markov chains for infinite-length algorithm asymptotics and, likely, estimates for large deviations for finite-length optimal annealing schedules.

The opposite way to analyze genetic algorithms is from *inside-out*, *i.e.*, to systematically analyze the behavior of possibly different, specially designed EAs for specific classes of problem instances. Such a way of analysis is even more in the spirit of [30–32]. It could be characterized as "finding the precisest asymptotic bounds" for specific classes of problem instances and corresponding EAs. In section 3 of [3], this approach to theoretical analysis of genetic or evolutionary algorithms is advocated and illustrated in detail. It is pointed out that this direction of research leads to a unknown territory waiting for exploration. While many of the results developed earlier analyze simple EAs and problems (see, *e.g.*, [21]), the methods used for their analysis lay the foundation for the analysis of more complicated EAs with crossover [29], or for more complicated problems from combinatorial optimization [23], or noisy resp. dynamic variants of simple problems [18, 19].

Extensions of EA-analysis to a theory for evolution strategies. Many applications of EAs, in particular, in engineering are numerical and rather non-discrete optimization problems where $\mathbf{R}$-valued parameters are optimized in a bounded or unbounded domain. Historically, $\mathbf{R}$-valued EAs are called evolution strategies (*cf.* [58]). In this scenario, asymptotic convergence results are even more interesting, since the optimum is usually approximated with increasing degree of accuracy without being attained.

There exists a considerable amount of research for evolution strategies as outlined in [3, Sec. 2]. Most of these results deal with local measures, like quality gain or local progress in one step, making it hard to generalize them to global measures. Only recently, work in [28] has presented a successful analysis of the number of steps necessary to halve the distance to the optimum (with special

consideration of the 1/5-rule [44]). Although using different techniques this work is in the same spirit than much of the runtime analyses performed for discrete EAs mentioned above. In the field of real-valued optimization, this approach besides an extension of the convergence methods presented later in the present paper seem to be worthwhile and promising areas for future research.

I.2. General Convergence Analysis for Scaled GAs

In this section, we shall discuss in detail an approach towards general asymptotic convergence analysis for a certain class of EAs, namely scaled GAs. This is based upon results in [48, 50–52] which develop a convergence analysis for a scaled version of the regular GA with standard operations, and results in [54–57] which develop a convergence analysis for several scaled versions of a GA in a coevolutionary setting. There are three major aspects that contribute to the proof of a convergence result for scaled GAs: (1) ergodicity which yields a unique limit probability distribution for the asymptotic behavior of the scaled GA that is independent of any initial population, (2) the mutation flow inequality which essentially yields convergence to uniform population through its interplay with a contraction property of the selection operator, and (3) the steady-state flow inequality which shows convergence to global optima by discussing the action of the stochastic matrices G_t, $t \in \mathbf{N}$, associated with the individual steps t of the algorithm on their steady-state probability distributions $v_t = G_t v_t$.

Modeling genetic algorithms similar to simulated annealing. The main direction of research carried out in [48–52, 54–57] tries to achieve convergence results for genetic algorithms by scaling of the GA-operators mutation, crossover and proportional fitness selection in accordance with predetermined scaling-schedules for the mutation-rate $\mu = \mu(t)$, the crossover-rate $\chi = \chi(t)$, and the exponentiation of the originally-given fitness function f. See, *e.g.*, [51, line (32), Thm. 3.3.2] in regard to the latter and (•) below. The fitness function f is thereby allowed to be population-dependent. Convergence is understood as asymptotic convergence of the probability distributions $w_t = G_t \cdot G_{t-1} \cdots G_1 w_o$, $t \in \mathbf{N}$, describing the state of the algorithm after a finite number t of steps towards a limit-probability-distribution $\lim w_t = \lim v_t = v_\infty$ that is non-zero only over populations containing globally optimal creatures (candidate solutions).

This approach follows and actually implements a course of action proposed in [15, p. 270] which was also pursued in [60, 61]. However, the main results in [60, 61] are contradicted by the examples in [50, Sec. 8.3].

One can argue that in an implementation of a real-world GA, one would always save the best element found so far in the GA-run, and, consequently, the GA need not be directed as a process itself towards globally optimal creatures. The authors think that this is, however, not an appropriate point of view for the following reasons: (1) Classical algorithms are directed towards finding a solution of a problem instance. (2) The simulated annealing algorithm which has been used in many real-world applications is directed towards finding optimal solutions in the same sense as convergence is understood in [50–52, 54–57] and

here. (3) Evolutionary algorithms such as the (1+1) EA are directed towards the optimum by an a-priory built-in memory mechanism. (4) Finding the maximum in regard to the occurring fitness-values in the population consumes computation time. (5) In view of the advances in DNA-computing, *cf.* [40], it is imaginable that genetic algorithms shall be implemented in the future using such techniques. In such a scenario, implementing a procedure for finding maxima in the population (*i.e.*, chemical solution) seems impractical. A much more practical goal would be to let the algorithm "freeze" in optimal state where "freezing" could be achieved by removing certain reactants from the chemical solution in which the DNA-computing operations take place.

The necessity of annealing. (*cf.* [53, p. 116]) Let G be the stochastic matrix associated with one cycle of a simple genetic algorithm. Study of the simple genetic algorithm with regular operators shows that a finite number of iterates yields an associated stochastic matrix G^t, $t \in \mathbf{N}$, which is fully positive. Thus, the associated invariant probability distribution $v = Gv = G^t v$ (*cf.* [52, Prop. 1.3.2] or [47, p. 7: Prop. 2.3]) is fully positive, and the algorithm does not asymptotically converge to optima, *cf.* [50, Thms. 8.1–2].

The preceding observation makes it necessary to vary at least some of the genetic operators over time in order to obtain an algorithm that converges asymptotically towards global maxima. The main result of [14] shows that annealing the mutation rate to 0 alone and consequently having the mutation operator approaching the identity operator $\mathbf{1}$ does not yields success. Thms. 8.2–3 of [50] show that increasing the selection pressure alone fails to achieve the goal of asymptotic global optimization. Varying the crossover operator alone is also of limited interest since crossover does not completely control mixing nor does it control the selection pressure towards optimal solutions in any way. Consequently, one has to satisfy the following conditions:

(M) one has to anneal the mutation rate to 0 in order to avoid asymptotically a positive probability for suboptimal solutions;

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

(C) possibly, one has to anneal the crossover rate to 0 such that the crossover operator approaches asymptotically the identity operator $\mathbf{1}$ as well.

In what follows, we shall discuss how one can satisfy the above conditions in order to obtain a scaled genetic algorithm that asymptoticly converges to global optima. We note that [50, Thm. 8.6] shows that under quite reasonable but not absolutely general circumstances condition **(C)** need not be satisfied.

Ergodicity. If the mutation-rate $\mu = \mu(t)$, the crossover-rate $\chi = \chi(t)$, and the exponentiation of the originally-given fitness function f change over the course of the algorithm, then one has to describe every cycle $t \in \mathbf{N}$ of the scaled GA by a separate stochastic matrix G_t. The family $(G_t)_{t \in \mathbf{N}}$ constitutes

an inhomogeneous Markov chain. There are two major issues to be settled in regard to $(G_t)_{t\in\mathbf{N}}$: weak ergodicity and strong ergodicity.

Essentially, weak ergodicity means that starting the scaled GA from two distinct populations p_0 and p_1, the probabilistic trajectories $w_t^{(0)}$ and $w_t^{(1)}$ with

$$w_t^{(\nu)} = G_t \cdot G_{t-1} \cdots G_1 p_\nu, \quad \nu = 0, 1, \quad t \in \mathbf{N},$$

which describe the state of the algorithm after t steps will become arbitrarily close. Weak ergodicity is obtained by using the combined contraction properties of the fully positive mutation matrices M_t, *i.e.*, every stochastic matrix M_t describing mutation in step t of the scaled GA satisfies $||M_t(v-w)||_1 \leq \epsilon_t \cdot ||v-w||_1$ for arbitrary probability distributions v, w over the set of populations, $\epsilon_t > 0$. The combined product $\prod_{\tau=1}^{t} \epsilon_\tau$ then converges to 0, if the mutation rate satisfies $\mu(t) = t^{-1/L}$ where L is the length of the combined genome in the population, *cf.* [51, Thm. 3.2.1].

Essentially, strong ergodicity means that every probabilistic trajectory $w_t^{(0)}$ as above has the same limit probability distribution v_∞. Strong ergodicity follows from weak ergodicity somewhat automatically by applying [27, p. 160: Thm. V.4.3] or [52, Thm. 3.3.2]. What one has to verify is that $\sum_{t=1}^{\infty} ||v_{t+1} - v_t||_1 < \infty$ where $v_t = G_t v_t$ are the steady-state probability distributions of the G_t. The v_t are uniquely determined for small $\mu(t)$ using [51, Lemma 1.4.2]. This allows the v_t to be computed via Cramer's rule from a fixed set of entries of the G_t, *cf.* [51, p. 213: proof of Thm. 3.3.2]. The summability condition shown above is then verified by establishing that the coordinates of the v_t show monotone behavior as $t \to \infty$. Observe that the summability condition shown above implies that $v_\infty = \lim v_t$ exists.

The mutation-flow inequality. The mutation flow inequality 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 property of the selection operator towards uniform populations, then this ensures that the combined probability over non-uniform populations in the steady-state distribution v_t of G_t becomes small for small mutation rate $\mu(t)$, *cf.* [52, Sec. 3.1].

A population p is called uniform, if it contains only copies of one single creature or candidate solution. Let $P_{\mathcal{U}}$ be the projection (*i.e.*, diagonal matrix with entries in $\{0, 1\}$) onto the subspace generated by uniform populations in the free vector space over all populations. In order to establish the mutation flow inequality, one determines

$$\beta(t) = \min\{||P_{\mathcal{U}} M_t p||_1 : p \text{ a uniform population}\} \in (0, 1).$$

Here, $||P_{\mathcal{U}} M_t p||_1$ is the combined probability over uniform populations in the probability distribution $M_t p$ (*i.e.*, the p-column of M_t). One has $\beta(t) \to 1$ as $t \to \infty$, and

$$||(\mathbf{1} - P_{\mathcal{U}}) M_t w||_1 \leq 1 - \beta(t) + \beta(t)||(\mathbf{1} - P_{\mathcal{U}}) w||_1$$

for any probability distribution w over populations [52, Prop. 3.1.1].

One can now show for the steady-state distributions v_t as above that

$$||(\mathbf{1} - P_{\mathcal{U}})v_t||_1 \leq \theta(1 - \beta(t))/(1 - \theta\beta(t))$$

where $\theta \in (0, 1)$ is a fixed constant depending upon the selection operator [51, Thm. 3.1.1]. This shows that the limit probability distribution $v_\infty = \lim v_t$ which determines the asymptotic behavior of the scaled GA is non-zero only over uniform populations.

The steady-state flow inequality. The final major ingredient in the proof of asymptotic convergence to global optima for scaled GA is the steady-state flow inequality. In order to establish this inequality, denote by Ω the set of populations that contain only globally optimal creatures. Elements in Ω need not be uniform. Let P_Ω be the projection onto the subspace generated by Ω. By discussing the separate actions of the genetic operators mutation, crossover and selection on the steady-state probability distributions $v_t = G_t v_t$, one obtains for $\omega(t) = ||P_\Omega v_t||_1$ the following inequality:

$$1 - \omega(t) \leq K_1(\mu(t)^{s/2} + \mu(t)^{LB\log(\rho_2(f))+1})\omega(t) + (1 - K_2\mu^\ell)(1 - \omega(t))$$

where $K_1, K_2 > 0$ are fixed constants, and $\rho_2(f) > 1$ is a fixed constant that depends upon the fitness-function and measures strength of second-to-best creatures in populations containing globally optimal creatures. $\rho_2(f)$ is easy to determine, if the fitness-function is given by rank. See, *e.g.*, [51, Thm. 3.3.2] and [52, Thm. 3.4.1]. In addition, ℓ denotes the length of creatures (candidate solutions) as strings in their encoding, s denotes the population size, and $L = \ell \cdot s$. The particular simple form of the steady-state flow inequality shown above supposes that $\chi(t) = \mu(t)$, and the fitness function f is exponentiated in accordance with the following schedule

$$f_t = f^{B\cdot\log(t+1)} \quad , t \in \mathbf{N}, \qquad (\bullet)$$

where $B > 0$ is a constant the user is free to chose. It is easy to see that $\lim_{t\to\infty} \omega(t) = ||P_\Omega v_\infty||_1 = 1$, if $\ell < s/2$ and $\ell < LB\log(\rho_2(f)) + 1$.

Applications to runtime analysis. The steady-state flow inequality and similarly the mutation-flow inequality provide explicit bounds for probability over populations containing globally optimal elements (resp. uniform populations) that can be applied also in a homogeneous Markov chain setting. It seems an interesting challenge to study analogues of the above inequalities for EAs other than GAs with proportional fitness selection, and possible applications to runtime estimates.

References

See the bibliography of "S. Droste & L.M. Schmitt, *Feasible Approaches to Convergence Results for Evolutionary Algorithms. Part II: Runtime analysis of evolutionary algorithms and summary.*" which is also published in these workshop proceedings and contains references of all papers listed in both parts.