10-2014

Theoretical Analysis of Steady State Genetic Algorithms

Alexandru Agapie
Bucharest University of Economic Studies

Alden H. Wright
University of Montana - Missoula, alden.wright@umontana.edu

59 (2014) APPLICATIONS OF MATHEMATICS No. 5, 509–525

THEORETICAL ANALYSIS OF STEADY STATE GENETIC ALGORITHMS

Alexandru Agapie, Bucharest, Alden H. Wright, Missoula

(Received March 24, 2013)

Abstract. Evolutionary Algorithms, also known as Genetic Algorithms in a former terminology, are probabilistic algorithms for optimization, which mimic operators from natural selection and genetics. The paper analyses the convergence of the heuristic associated to a special type of Genetic Algorithm, namely the Steady State Genetic Algorithm (SSGA), considered as a discrete-time dynamical system non-generational model. Inspired by the Markov chain results in finite Evolutionary Algorithms, conditions are given under which the SSGA heuristic converges to the population consisting of copies of the best chromosome.

Keywords: genetic algorithm; Markov chain; random heuristic search

MSC 2010: 60J10, 68W20, 90C59

1. Introduction

In the finite-population paradigm, a Genetic Algorithm (GA) is *convergent* if the probability of containing the global optimum (best chromosome) inside the current generation tends to one as the generation index tends to infinity. In short, the algorithm's convergence can be related to the asymptotic behavior of a finite homogeneous Markov chain (MC) by the following condensed procedure.

(1) Identify the MC set of states to the set of all possible fixed-size GA populations.

(2) Check if the associated transition matrix is of the form $P = \begin{pmatrix} R & 0 \\ Q & T \end{pmatrix}$.

(3) If the matrix is of the form P (the MC is called *reducible* in this case), and if all non-optimal populations are in T, then GA is convergent.

The research has been supported by grant of the Romanian National Authority for Scientific Research, CNCS-UEFISCDI, project number PNII-ID-PCCE-2011-0015, and by a COBASE grant from the National Science Foundation, USA.

(4) If the matrix cannot be put into the form P (the MC is called *ergodic*), then GA is *not* convergent.

The first analysis exposing the reducible behavior of the MC was performed by Rudolph, for the case of an *elitist* GA—that is, a GA maintaining the best solution from a generation to another [8]. Successive refining of the elitist case finally concluded in the simple convergence condition presented [1], [9]. Different paths emerged from that simple conditions, leading to various convergence analyses for either adaptive algorithms [2], continuous space algorithms [4], [5] or even marginal distribution algorithms [3].

A totally different interpretation of the GA theory came from dynamical systems, when Vose [11] regarded GA populations as points in the simplex. He developed the theory of *Random Heuristic Search* and built his analysis on a two-limit behavior. First, on each fixed population size the corresponding MC must approach its limit distribution from the ergodic case presented above. Second, as the population size tends to infinity the corresponding limits gather into a sequence of distributions which, under specific requirements, will approach a distribution concentrated on one of the 'fixed point' populations, say x_0. The fact that x_0 contains copies of the best individual is then a simple consequence of the large population size and of x_0 being situated in the interior of the simplex.

It is worth noticing that the second modeling presented is not primarily intended for studying 'convergence' in the finite-population sense, but in the sense of discrete time dynamical systems (that is, with respect to stable/unstable fixed-points inside the simplex). Within this framework the theory splits into the expected value and infinite population models, mainly concentrated on generational genetic algorithms.

Yet, many practitioners advocate the use of steady-state genetic algorithms where a single individual is replaced at each step. Discrete-time expected value models are described in this paper, where each time step corresponds to the replacement of an individual.

The steady-state model that uses random deletion has a very close correspondence with the generational model that uses the same crossover, mutation, and selection. It is a remarkable result that a SSGA with random deletion has the same fixed-points as a generational GA with common heuristic function $\mathcal{G}$, as shown in [7], [13].

Let Ω denote the search space for a search problem. We identify Ω with the integers in the range from 0 to $n-1$, where n is the cardinality of Ω. We assume a real-valued nonnegative fitness function f over Ω. We will denote $f(i)$ by f_i. Our objective is to model population-based search algorithms that search for elements of Ω with high fitness. Such algorithms can be *generational*, where a large proportion of the population is replaced at each time step (or generation). Or they can be *steady-state*, where only a single or small number of population members are replaced in a time step.

A population is a multiset (set with repeated elements) with elements drawn from Ω. We will represent populations over Ω by nonnegative vectors indexed over the integers in the interval $[0, n)$ whose sum is 1. If a population of size r is represented by a vector p, then rp_i is the number of copies of i in the population. For example, if $\Omega = \{0, 1, 2, 3\}$, and the population is the multiset $\{0, 0, 1, 2, 2\}$, then the population is represented by the vector $\langle 2/5, 1/5, 2/5, 0\rangle$.

Let us define

$$\Lambda = \left\{ x \colon \sum_{i=0}^{n-1} x_i = 1 \text{ and } x_i \geqslant 0 \text{ for all } i \right\}.$$

Then all populations over Ω are elements of Λ, and Λ can also be interpreted as the set of probability distributions over Ω. It is natural to think of elements of Λ as *infinite populations*. Geometrically, Λ is the unit simplex in $\mathbb{R}^n$.

The ith unit vector in $\mathbb{R}^n$ is denoted by e_i. The Euclidean norm on $\mathbb{R}^n$ is denoted by $\|\cdot\| = \|\cdot\|_2$, the max norm by $\|\cdot\|_\infty$, and the sum norm by $\|\cdot\|_1$. The Euclidean norm is the default.

Vose's random heuristic search algorithm describes a class of generational population-based search algorithms. The model is defined by a *heuristic function* $\mathcal{G} \colon \Lambda \to \Lambda$. If x is a population of size r, then the next generation population is obtained by taking r independent samples from the probability distribution $\mathcal{G}(x)$. When random heuristic search is used to model the simple genetic algorithm, $\mathcal{G}$ is the composition of a selection heuristic function $\mathcal{F} \colon \Lambda \to \Lambda$ and a mixing heuristic function $\mathcal{M} \colon \Lambda \to \Lambda$. The mixing function describes the properties of crossover and mutation. Properties of the functions $\mathcal{M}$ and $\mathcal{F}$ are explored in detail in [11].

Given a population $x \in \Lambda$, it is not hard to show that the expected next generation population is $\mathcal{G}(x)$. As the population size goes to infinity, the next generation population converges in probability to its expectation, so it is natural to use $\mathcal{G}$ to define an infinite population model. Thus, $x \mapsto \mathcal{G}(x)$ defines a discrete-time dynamical system on Λ that we will call the *generational model*. Given an initial population x, the trajectory of this population is the sequence $x, \mathcal{G}(x), \mathcal{G}^2(x), \mathcal{G}^3(x), \ldots$

Note that after the first step, the populations produced by this model do not necessarily correspond to populations of size r. Building on previous analysis of the Steady State GA (SSGA) [12], [13], this paper aims to bring together for the first time the finite and infinite population paradigms. Namely, we tackle two versions of the steady-state algorithm (represented by a heuristic function) and give conditions for their convergence to the uniform population consisting of copies of the global optimum (similar to the finite-population approach).

2. Steady-state genetic algorithms

Whitley's Genitor algorithm [12] was the first "steady state" genetic algorithm. Genitor selects two parent individuals by ranking selection and applies mixing to them to produce one offspring, which replaces the worst element of the population. Syswerda [10] described variations of the steady-state genetic algorithm and empirically compared various deletion methods. Davis [6] also empirically tested steady-state genetic algorithms and advocates them as being superior to generational GAs when combined with a feature that eliminates duplicate chromosomes.

In this section, we describe two versions of steady-state search algorithms. Both the algorithms start with a population η of size r. In most applications, this population would be chosen randomly from the search space, but there is no requirement for a random initial population. At each step of both algorithms, an element j is removed from the population, and an element i of Ω is added to the population, The selection of element i is described by a heuristic function $\mathcal{G}$. (For a genetic algorithm, $\mathcal{G}$ will describe crossover, mutation, and usually selection.) The selection of element j is described by another heuristic function $\mathcal{D}_r$. (We include the population size r as a subscript, since there may be a dependence on population size.)

In the first algorithm, the heuristic functions $\mathcal{G}$ and $\mathcal{D}_r$ both depend on x, the current population. Thus, i is selected from the probability distribution $\mathcal{G}(x)$, and j is selected from the probability distribution $\mathcal{D}_r(x)$.

Steady-state random heuristic search algorithm 1:

1. Choose an initial population η of size r.
2. $x \leftarrow \eta$.
3. Select i from Ω using the probability distribution $\mathcal{G}(x)$.
4. Select j using the probability distribution $\mathcal{D}_r(x)$.
5. Replace x by $x - e_j/r + e_i/r$.
6. Go to step 3.

The second algorithm differs from the first by allowing for the possibility that the newly added element i might be deleted. Thus, j is selected from the probability distribution $\mathcal{D}((rx + e_i)/(r+1))$. This algorithm is an $(r+1)$ algorithm in evolution strategy notation.

Steady-state random heuristic search algorithm 2:

1. Choose an initial population η of size r.
2. $x \leftarrow \eta$.
3. Select i from Ω using the probability distribution $\mathcal{G}(x)$.

4′. Select j using the probability distribution $\mathcal{D}_r((rx + e_i)/(r+1))$.

5. Replace x by $x - e_j/r + e_i/r$.
6. Go to step 3.

Some heuristics that have been suggested for the $\mathcal{D}_r$ function include worst-element deletion, where a population element with the least fitness is chosen for deletion, reverse proportional selection, reverse ranking deletion, and random deletion, where the element to be deleted is chosen randomly from the population. Random deletion was suggested by Syswerda [10]. He points out that random deletion is seldom used in practice. Because of this, one of the reviewers of this paper objected to the use of the term "steady-state genetic algorithm" for an algorithm that used random deletion. However, we feel that the term can be applied to any genetic algorithm that replaces only a few members of the population during a time step of the algorithm.

Random deletion can be modeled by choosing $\mathcal{D}_r(x) = x$.

If the fitness function is injective (the fitnesses of elements of Ω are distinct), then reverse ranking and worst-element deletion can be modeled using the framework developed for ranking selection in [11],

$$\mathcal{D}_r(x)_i = \int_{\sum_{\{j\colon f_j < f_i\}} x_j}^{\sum_{\{j\colon f_j \leqslant f_i\}} x_j} \varrho(s)\,\mathrm{d}s.$$

The probability density function $\varrho(s)$ can be chosen to be $2s$ to model standard ranking selection, and $2 - 2s$ to model reverse ranking deletion. To model worst-element deletion, we define $\varrho(s)$ as follows:

$$\varrho(s) = \begin{cases} r & \text{if } 0 \leqslant s \leqslant 1/r, \\ 0 & \text{otherwise.} \end{cases}$$

As an example, let $n = 3$, $x = \langle \frac{1}{3}\ \frac{1}{6}\ \frac{1}{2} \rangle^{\mathrm{T}}$, $f = \langle 2\ 1\ 3 \rangle^{\mathrm{T}}$, and $r = 4$. Then $\varrho(s) = 4$ if $0 \leqslant s \leqslant 1/4$ and $\varrho(s) = 0$ if $1/4 < s \leqslant 1$. (The population x does not correspond to a real finite population of size 4. However, this choice leads to a more illustrative example. Also, if $\mathcal{D}_r$ is iterated, after the first iteration the populations produced will not necessarily correspond to finite populations of size r.) Then

$$\mathcal{D}_r(x)_1 = \int_0^{x_1} \varrho(s)\,\mathrm{d}s = \int_0^{1/6} 4\,\mathrm{d}s = 2/3,$$
$$\mathcal{D}_r(x)_0 = \int_{x_1}^{x_1+x_0} \varrho(s)\,\mathrm{d}s = \int_{1/6}^{1/2} \varrho(s)\,\mathrm{d}s = \int_{1/6}^{1/4} 4\,\mathrm{d}s = 1/3,$$

and

$$\mathcal{D}_r(x)_2 = \int_{x_1+x_0}^{x_1+x_0+x_2} \varrho(s)\,\mathrm{d}s = \int_{1/2}^{1} \varrho(s)\,\mathrm{d}s = 0.$$

For random deletion and reverse ranking deletion, $\mathcal{D}_r(x)$ does not depend on the population size and can be shown to be differentiable as a function of x.

For worst-element deletion, $\mathcal{D}_r(x)$ does depend on the population size, and is continuous but not differentiable.

Lemma 2.1. *If $\mathcal{D}_r$ is defined as above for worst-element deletion, then $\mathcal{D}_r$ satisfies a Lipschitz condition. In other words, there is a constant L_r such that $\|\mathcal{D}_r(x) - \mathcal{D}_r(y)\| \leqslant L_r\|x - y\|$ for all $x, y \in \Lambda$.*

P r o o f. Let $x, y \in \Lambda$. Then for an arbitrary index i we have

$$\begin{aligned}
|\mathcal{D}_r(x)_i - \mathcal{D}_r(y)_i| &= \left| \int_{\sum_{\{j\colon f_j<f_i\}} x_j}^{\sum_{\{j\colon f_j\leqslant f_i\}} x_j} \varrho(s)\,\mathrm{d}s - \int_{\sum_{\{j\colon f_j<f_i\}} y_j}^{\sum_{\{j\colon f_j\leqslant f_i\}} y_j} \varrho(s)\,\mathrm{d}s \right| \\
&= \left| \int_{\sum_{\{j\colon f_j<f_i\}} x_j}^{\sum_{\{j\colon f_j<f_i\}} y_j} \varrho(s)\,\mathrm{d}s - \int_{\sum_{\{j\colon f_j\leqslant f_i\}} x_j}^{\sum_{\{j\colon f_j\leqslant f_i\}} y_j} \varrho(s)\,\mathrm{d}s \right| \\
&\leqslant r \sum_{\{j\colon f_j<f_i\}} |y_j - x_j| + r \sum_{\{j\colon f_j\leqslant f_i\}} |y_j - x_j| \\
&\leqslant 2r \sum_{j=0}^{n-1} |y_j - x_j| = 2r\|x - y\|_1.
\end{aligned}$$

Thus, $\|\mathcal{D}_r(x) - \mathcal{D}_r(y)\|_\infty \leqslant 2r\|x - y\|_1$. Since all norms are equivalent up to a constant, $\|\mathcal{D}_r(x) - \mathcal{D}_r(y)\|_2 \leqslant 2rK\|x - y\|_2$ for some constant K. □

The function

$$\mathcal{H}_r(x) = x + \frac{1}{r}\mathcal{G}(x) - \frac{1}{r}\mathcal{D}_r(x) \tag{2.1}$$

gives the expected population for Algorithm 1 at the next time step, and the function

$$\mathcal{K}_r(x) = x + \frac{1}{r}\mathcal{G}(x) - \frac{1}{r}\mathcal{D}_{r+1}\Big(\frac{rx + \mathcal{G}(x)}{r+1}\Big) \tag{2.2}$$

gives the expected population for Algorithm 2 at the next time step.

Thus, $x \mapsto \mathcal{H}_r(x)$ and $x \mapsto \mathcal{K}_r(x)$ define discrete-time expected-value models of the above steady-state algorithms. We will call them the *discrete-time steady-state models*.

The following is straightforward.

Lemma 2.2. *If the deletion heuristic $\mathcal{D}_r$ of the discrete-time steady-state models satisfies*

$$\mathcal{D}_r(y) \leqslant rx + \mathcal{G}(x), \tag{2.3}$$

where $y = x$ for (2.1) and $y = (rx + \mathcal{G}(x))/(r+1)$ for (2.2), then the trajectories of the systems defined by $\mathcal{H}_r$ and $\mathcal{K}_r$ remain in the simplex Λ.

The models for random deletion, reverse ranking deletion, and worst-element deletion all satisfy the hypotheses of Lemma 2.2.

3. Convergence of the $\mathcal{K}_r$ heuristic

In this section we assume that the deletion heuristic is defined by worst element deletion. We give conditions on the fitness function and on $\mathcal{G}$ that ensure that $\lim_{t\to\infty} \mathcal{K}_r^t(x)$ exists and is the uniform population consisting of copies of the global optimum.

In evolution strategy terminology, this is an $(r+1)$-ES algorithm which uses an elitist selection method. Rudolph [9] has shown that for this class of algorithms, if there is mutation rate which is greater than zero and less than one, then the finite population algorithm converges completely and in mean. These are statements about the best element in the population rather than the whole population, so these results do not imply our result.

We assume that the fitness function is injective. In other words, we assume that if $i \neq j$, then $f_i \neq f_j$. Since we will not be concerned with the internal structure of Ω, without loss of generality we can assume that $f_0 < f_1 < \ldots < f_{n-1}$. This assumption will simplify notation.

Under this assumption, we can give a simplified definition for the worst-element deletion heuristic $\mathcal{D}_{r+1}$ that is used in the definition of $\mathcal{K}_r$:

$$\mathcal{D}_{r+1}(y)_i = \begin{cases} (r+1)y_i & \text{if } \sum_{j\leqslant i} y_j \leqslant \dfrac{1}{r+1}, \\ 1-(r+1)\sum_{j<i} y_j & \text{if } \sum_{j<i} y_j \leqslant \dfrac{1}{r+1} < \sum_{j\leqslant i} y_j, \\ 0 & \text{if } \dfrac{1}{r+1} < \sum_{j<i} y_j. \end{cases}$$

Now let us define $m(x) = \min\{i\colon\ x_i > 0\}$.

Theorem 3.1. *If there is a $\delta > 0$ such that for all $x \in \Lambda$,*

$$\sum_{j>m(x)} \mathcal{G}_j(x) \geqslant \delta,$$

then $\lim_{t\to\infty} \mathcal{K}_r^t(x) = e_{n-1}$ for all $x \in \Lambda$.

This condition says that $\mathcal{G}(x)$ has a combined weight of at least δ at those points of Ω whose fitness is higher than the worst-fitness element of x. (By "element of x",

we mean any $i \in \Omega$ such that $x_i > 0$.) This condition would be satisfied by any $\mathcal{G}$ heuristic that allowed for a positive probability of mutation between any elements of Ω. The proof of Theorem 3.1 will follow from Lemma 3.3.

Lemma 3.2. *For any $x \in \Lambda$, if $j < m(x)$, then $\mathcal{K}_r(x)_j = 0$.*

P r o o f. To simplify notation, let m stands for $m(x)$.

Let $y = (rx + \mathcal{G}(x))/(r+1)$. Then $\sum\limits_{j<m} \mathcal{G}(x)_j \leqslant 1/(r+1)$, since $\sum\limits_{j<m} x_j = 0$ and $\sum\limits_{j<m} \mathcal{G}_j \leqslant 1$.

Thus, for $j < m$, $\mathcal{D}_{r+1}(y)_j = y_j$, and $\mathcal{K}_r(x)_j = y_j - \mathcal{D}_{r+1}(y)_j = 0$. □

Define $M(x) = 2m(x) + 1 - x_{m(x)}$.

Lemma 3.3. *For any $x \in \Lambda$, if there is a $\delta > 0$ such that $\sum\limits_{j>m(x)} \mathcal{G}(x)_j \geqslant \delta$, then*

$$M(\mathcal{K}_r(x)) \geqslant M(x) + \frac{\delta}{r}.$$

P r o o f. To simplify notation, again let m stands for $m(x)$. Let $y = (rx+\mathcal{G}(x)) \times (r+1)^{-1}$.

Case 1: $\sum\limits_{j\leqslant m} y_j \leqslant 1/(r+1)$. Then

$$\mathcal{D}_{r+1}(y)_m = (r+1)y_m = rx_m + \mathcal{G}(x)_m,$$

and

$$\mathcal{K}_r(x)_m = x_m + \frac{1}{r}\mathcal{G}(x)_m - \frac{1}{r}(rx_m + \mathcal{G}(x)_m) = 0.$$

Thus

$$M(\mathcal{K}_r(x)) \geqslant 2(m+1) + 1 - x_{m+1} \geqslant 2m + 2 \geqslant M(x) + 1.$$

Case 2: $\sum\limits_{j<m} y_j \leqslant 1/(r+1) < \sum\limits_{j\leqslant m} y_j$. Then

$$\mathcal{D}_{r+1}(y)_m = 1 - (r+1)\sum_{j<m} y_j = 1 - \sum_{j<m} \mathcal{G}(x)_j.$$

Thus

$$\begin{aligned}
\mathcal{K}_r(x)_m &= x_m + \frac{1}{r}\mathcal{G}(x)_m - \frac{1}{r}\mathcal{D}_{r+1}(y)_m \\
&= x_m - \frac{1}{r}\left(1 - \sum_{j\leqslant m} \mathcal{G}(x)_j\right) \\
&= x_m - \frac{1}{r}\sum_{j>m} \mathcal{G}(x)_j \\
&\leqslant x_m - \frac{\delta}{r}.
\end{aligned}$$

Also note that

$$\frac{1}{r+1} < \sum_{j \leqslant m} y_j \Longrightarrow 1 < \sum_{j \leqslant m} (rx_j + \mathcal{G}(x)_j)$$
$$\Longrightarrow x_m - \frac{1}{r}\left(1 - \sum_{j \leqslant m} \mathcal{G}(x)_j\right) > 0 \Longrightarrow \mathcal{K}_r(x)_m > 0.$$

Thus

$$\begin{aligned} M(\mathcal{K}_r(x)) &= 2m + 1 - x_m + \frac{1}{r}\sum_{j>m} \mathcal{G}(x)_j \\ &= M(x) + \frac{1}{r}\sum_{j>m} \mathcal{G}(x) \\ &\geqslant M(x) + \frac{\delta}{r}. \end{aligned}$$

Case 3: $1/(r+1) < \sum\limits_{j<m} y_j$. In this case, $1 < \sum\limits_{j<m} (rx_j + \mathcal{G}(x)_j)$, which implies $1 < \sum\limits_{j<m} \mathcal{G}(x)_j$. This is impossible, so the case never happens. □

4. Bounded-convergence of the $\mathcal{H}_r$ heuristic

In this section we assume that the deletion heuristic is defined by worst element deletion. We give conditions on the heuristic $\mathcal{G}$ that ensure that $\mathcal{H}_r^t(x)_{n-1} \geqslant \sigma$ for all $t \geqslant T_0$, where T_0 is a positive integer and $\sigma > 0$ is a constant depending on the population size r. We also assume that the fitness function is injective and that $f_0 < f_1 < \ldots < f_{n-1}$.

Then the worst-element deletion heuristic $\mathcal{D}_r$ used in the definition of $\mathcal{H}_r$ simplifies to

$$\mathcal{D}_r(y)_i = \begin{cases} ry_i & \text{if } \sum\limits_{j \leqslant i} y_j \leqslant \dfrac{1}{r}, \\ 1 - r\sum\limits_{j<i} y_j & \text{if } \sum\limits_{j<i} y_j \leqslant \dfrac{1}{r} < \sum\limits_{j \leqslant i} y_j, \\ 0 & \text{if } \dfrac{1}{r} < \sum\limits_{j<i} y_j. \end{cases} \tag{4.1}$$

First, we show that without imposing any condition on heuristic $\mathcal{G}$, if the starting point has a positive last component ($x_{n-1} > 0$), then the same property will be shared by all subsequent iterations of heuristic $\mathcal{H}$. This is the correspondent of '*elitism*', in the finite population EA theory.

Proposition 4.1. *If $x_{n-1} = \sigma \in (0, (r-1)/r)$, then $\mathcal{H}_r^t(x)_{n-1} \geqslant \sigma$ for all $t \geqslant 0$.*

P r o o f. We have

$$x_{n-1} \leqslant \frac{r-1}{r} \Rightarrow 1 - x_{n-1} \geqslant 1 - \frac{r-1}{r} \Rightarrow \sum_{j<n-1} x_j \geqslant \frac{1}{r} \Rightarrow \mathcal{D}_r(x)_{n-1} = 0,$$

which leads to

$$\mathcal{H}_r(x)_{n-1} = x_{n-1} + \frac{1}{r}\mathcal{G}(x)_{n-1} - \frac{1}{r}\mathcal{D}(x)_{n-1} = x_{n-1} + \frac{1}{r}\mathcal{G}(x)_{n-1} - 0 \geqslant x_{n-1} = \sigma.$$

Let us see what happens if $\mathcal{H}_r(x)_{n-1} = y > (r-1)/r$. By taking the second branch in the deletion heuristic one gets

$$\mathcal{H}_r(y)_{n-1} = y_{n-1} + \frac{1}{r}\mathcal{G}(y)_{n-1} - \frac{1}{r} + 1 - y_{n-1} = 1 - \frac{1}{r} + \frac{1}{r}\mathcal{G}(y)_{n-1} \geqslant 1 - \frac{1}{r} > \sigma.$$

□

Next, the interesting problem would be to find conditions on heuristic $\mathcal{G}$ that ensure the positivity of the last component of $\mathcal{H}_r^t$ for some $t > 0$, regardless of the starting point x. One way to do this is by copying the *positive mutation assumption* from the finite population case [9], [8], by imposing $\mathcal{G}(x)_i \geqslant \delta$ for some appropriate fixed $\delta > 0$ and for all i, $0 \leqslant i \leqslant n-1$.

Yet, one can find weaker assumptions on heuristic $\mathcal{G}$ that ensure the same behavior of $\mathcal{H}_r$, and this will be proved in the rest of this section.

We start with a simple counter-example, showing that the convergence condition for heuristic $\mathcal{K}_r$ (proved in the previous section) is no longer valid for the $\mathcal{H}_r$ heuristic.

E x a m p l e 1. Let heuristic $\mathcal{H}$ be defined by

$$\mathcal{H}_r(x) = x + \frac{1}{r}\mathcal{G}(x) - \frac{1}{r}\mathcal{D}_r(x),$$

where deletion is defined by (4.1), and $\mathcal{G}$ is constrained by $\sum_{j>o(x)} \mathcal{G}_j(x) \geqslant \delta$ only, for some $\delta > 0$. Then the evolution described in the table below precludes convergence. Let $\Omega = \{0, 1, 2\}$ and let $r = 3$.

Ω	0	1	2
rx	0	1	2
$\mathcal{G}(x)$	$1-\delta$	0	δ
$\mathcal{D}_r(x)$	0	1	0
$r\mathcal{H}_r(x) = ry$	$1-\delta$	0	$2+\delta$
$\mathcal{G}(y)$	$1-\delta$	δ	0
$\mathcal{D}_r(y)$	$1-\delta$	0	δ
$r\mathcal{H}_r^2(x) = r\mathcal{H}_r(y)$	$1-\delta$	δ	2

In order to obtain convergent behavior for this heuristic, we must first define another index function, say $o(x)$. This will be related to the population size r in the following manner:

$$o(x) = o_r(x) = \min\left\{j\colon \sum_{i\leqslant j} x_i \geqslant \frac{1}{r}\right\}.$$

We can now state the main result on the $\mathcal{H}$ heuristic.

Theorem 4.2. *If there is a $\delta > 0$ such that for all $x \in \Lambda$,*

$$\sum_{j>o_r(x)} \mathcal{G}_j(x) \geqslant \delta,$$

then there is a positive integer T_{n-2} such that for all $t \geqslant T_{n-2}$,

$$\sum_{i\leqslant n-2} \mathcal{H}_r^t(x)_i \leqslant \frac{1}{r}.$$

P r o o f. We start by tackling the case where x has the first component greater than $1/r$ (that is, $o_r(x) = 0$), all other components being arbitrary.

Lemma 4.3. *Let $x \in \Lambda$, $x = \langle 1/r + \varepsilon, \ldots, \ldots\rangle$, with $\varepsilon > 0$, under the hypothesis of Theorem 4.2. Then there is a positive integer T_0 such that for all $t \geqslant T_0$,*

$$\mathcal{H}_r^t(x)_0 \leqslant \frac{1}{r}.$$

P r o o f. Since $x_0 > 1/r$, we have $\mathcal{D}_r(x) = \langle 1, 0, \ldots, 0\rangle$ and $r(x) = 0$, thus

$$\mathcal{H}_r(x)_0 = \frac{1}{r} + \varepsilon + \frac{1}{r}\mathcal{G}(x)_0 - \frac{1}{r} = \varepsilon + \frac{1}{r}\mathcal{G}(x)_0 \leqslant \varepsilon + \frac{1-\delta}{r}.$$

Let us assume that $\mathcal{H}_r(x)_0 > 1/r$. This yields

$$\mathcal{H}_r^2(x)_0 \leqslant \varepsilon + \frac{1-\delta}{r} + \frac{1}{r}\mathcal{G}(\mathcal{H}(x))_0 - \frac{1}{r} \leqslant \varepsilon + \frac{2(1-\delta)}{r} - \frac{1}{r} = \varepsilon + \frac{1-\delta}{r} - \frac{\delta}{r}.$$

By repeating this reasoning under the assumption $\mathcal{H}_r^s(x)_0 > 1/r$, for all $s \in \{0, 1, \ldots, t-1\}$, we obtain

$$\mathcal{H}_r^t(x)_0 \leqslant \varepsilon + \frac{1}{r} - \frac{t\delta}{r}.$$

So, as t increases, the iterations $\mathcal{H}_r^t(x)_0$ will descend under $1/r$, starting with some index T_0. Let us see what happens at the next iteration (we denote $\mathcal{H}_r^{T_0}(x) = y$)

$$\mathcal{H}_r^{T_0+1}(x)_0 = y_0 + \frac{1}{r}\mathcal{G}(y)_0 - y_0 = \frac{1}{r}\mathcal{G}(y)_0 \leqslant \frac{1-\delta}{r} < \frac{1}{r}.$$

So, $\mathcal{H}_r^t(x)_0 < 1/r$ for all $t \geqslant T_0$. □

The following result generalizes Lemma 4.3 to the case $o_r(x) = k$, $k \leqslant n-2$.

Lemma 4.4. *Assume the hypothesis of Theorem* 4.2. *If* $o_r(x) = k$, $1 \leqslant k \leqslant n-2$, *then there is a positive integer* T_k *such that for all* $t \geqslant T_k$,

$$\sum_{i \leqslant k} \mathcal{H}_r^t(x)_i \leqslant \frac{1}{r}.$$

P r o o f. Looking at the deletion operator (4.1), $o_r(x) = k$ implies

$$\mathcal{D}_r(x)_i = \begin{cases} rx_i & \text{if } i < k, \\ 1 - r \sum\limits_{j<k} x_j & \text{if } i = k, \\ 0 & \text{if } i > k. \end{cases}$$

Thus $\sum\limits_{i \leqslant k} \mathcal{D}_r(x)_i = 1$. Next,

$$\begin{aligned} \sum_{i \leqslant k} \mathcal{H}_r(x)_i &= \sum_{i \leqslant k} x_i + \frac{1}{r} \sum_{i \leqslant k} \mathcal{G}(x)_i - \frac{1}{r} \sum_{i \leqslant k} \mathcal{D}_r(x)_i \\ &\leqslant \sum_{i \leqslant k} x_i + \frac{1}{r}(1 - \delta - 1) = \sum_{i \leqslant k} x_i - \frac{\delta}{r}. \end{aligned}$$

Now, if we suppose that $o_r(\mathcal{H}_r(y)) = k$, we get

$$\sum_{i \leqslant k} \mathcal{H}_r^2(x)_i \leqslant \sum_{i \leqslant k} x_i - \frac{\delta}{r} + \frac{1-\delta}{r} - \sum_{i<k} x_i' + \frac{1}{r} + \sum_{i<k} x_i' = \sum_{i \leqslant k} x_i - \frac{2\delta}{r},$$

where we denoted $\mathcal{H}_r(x) = x'$. As in the proof of Lemma 4.3, there will be a positive integer T_k such that $\sum\limits_{i \leqslant k} \mathcal{H}_r^{T_k}(x)_i \leqslant 1/r$ and all successive iterations will preserve the inequality. □

Now, turning back to the proof of Theorem 4.2, induction on the assertion *If* $o_r(x) = k$, *then there is a positive integer* T_k *such that* $o_r(\mathcal{H}_r^{T_k}(x)) > k$, *and this holds also for all* $t \geqslant T_k$ ensures the result. Lemmas 4.3 and 4.4 have proved the assertion up to $k = n-2$. Thus, starting with an arbitrary $x \in \Lambda$, by iterating $\mathcal{H}_r$ the index-string $\{o_r(\mathcal{H}_r^t(x)\}_{t \geqslant 0}$ increases (not necessarily strictly), until it reaches the value $n-1$, which will be never left. □

R e m a r k 1. Obviously, the constant $\sigma > 0$ that we announced at the beginning of the section as a lower bound for $\{\mathcal{H}_r^t(x)_{n-1}\}_{t \to \infty}$ is given by

$$\sigma = \frac{r-1}{r}.$$

5. Convergence of the $\mathcal{H}_r$ heuristic

In this section we prove that by imposing stronger assumptions on the heuristic $\mathcal{G}$ one can ensure that $\lim_{s\to\infty} \mathcal{H}_r^s(x)$ exists (at least for a subsequence $\{s_k\}_{k>0}$, and regardless of the starting point x) and is the uniform population consisting of copies of the global optimum, e_{n-1}.

Under the same hypothesis on the fitness function and the simplified form of the deletion heuristic (see the previous section), we now introduce in a different manner the index function, say $p(x)$. Namely, we put

$$p_s(x) = \min\left\{i\colon \sum_{j\leqslant i} x_j \geqslant \frac{1}{s}\right\}.$$

We can now state the main convergence result.

Theorem 5.1. *If for all $x \in \Lambda$ and all $s > 1$,*

$$\sum_{j>p_s(x)} \mathcal{G}_j(x) > 1 - \frac{1}{s+1}, \tag{5.1}$$

then there is a subsequence $\{s_k\}_{k>0}$ such that $\lim_{k\to\infty} \mathcal{H}_r^{s_k}(x) = e_{n-1}$.

P r o o f. As the population size r is considered fixed, it will be omitted in the sequel when referring to heuristic $\mathcal{H}$. We start with a useful result.

Lemma 5.2. *Under the hypothesis of Theorem* 5.1, *the following inequalities hold:*

(1) $\sum_{i\leqslant p_s(x)} \mathcal{H}(x)_i < 1/(s+1)$,

(2) $p_{s+1}(\mathcal{H}(x)) > p_s(x)$.

P r o o f. Let us start with a fixed s. As the interesting case corresponds to $s > r$ (and thus $p_s(x) \leqslant p_r(x)$), we shall make this assumption in the sequel. By applying the deletion operator (4.1) and assumption (5.1) one gets

$$\begin{aligned}
\sum_{i>p_s(x)} \mathcal{H}(x)_i &= \sum_{i>p_s(x)} (x)_i + \frac{1}{r}\sum_{i>p_s(x)} \mathcal{G}(x)_i - \frac{1}{r} + \sum_{i<p_r(x)} (x)_i - \sum_{i\in(p_s(x),p_r(x))} (x)_i \\
&= \sum_{i>p_s(x)} (x)_i + \sum_{i\leqslant p_s(x)} (x)_i + \frac{1}{r}\sum_{i>p_s(x)} \mathcal{G}(x)_i - \frac{1}{r} \\
&> 1 + \frac{1}{r}\Big(1 - \frac{1}{s+1}\Big) - \frac{1}{r} \geqslant 1 - \frac{1}{s+1} \\
&\Leftrightarrow \sum_{i\leqslant p_s(x)} \mathcal{H}(x)_i < \frac{1}{s+1},
\end{aligned}$$

which proves the first part of the lemma. Next, the second part is obvious. □

Turning back to the proof of Theorem 5.1, let us have a look at the string $\{p_s(\mathcal{H}^s(x))\}_{s\geqslant 0}$. By Lemma 5.2 this string increases up to $n-1$—that is, there is an index s_1 such that $p_{s_1}(\mathcal{H}^{s_1}(x)) = n-1$. From that point further, one can distinguish two cases.

Case 1: For all $h > 0$,

$$p_{s_1+h}(\mathcal{H}^{s_1+h}(x)) = n-1.$$

Then by applying Lemma 5.2 we get

$$\mathcal{H}^{s_1+h}(x)_{n-1} = 1 - \sum_{i\leqslant n-2} \mathcal{H}^{s_1+h}(x)_i \geqslant 1 - \frac{1}{s_1+h+1} \to 1 \quad (h\to\infty).$$

Thus, $\lim_{h\to\infty} \mathcal{H}^{s_1+h}(x) = e_{n-1}$.

Case 2: There is an index $s_2 = s_1 + h$ such that $p_{s_2}(\mathcal{H}^{s_2}) \leqslant n-2$. Then, by applying once more the above reasoning, one will find a greater index s_3 such that $p_{s_3}(\mathcal{H}^{s_3}) = n-1$, which leads to

$$\mathcal{H}^{s_3+h}(x)_{n-1} \geqslant 1 - \frac{1}{s_3+1}.$$

By reiterating the above procedure, we obtain a string $\{s_{2k+1}\}_k \to \infty$ for which $\mathcal{H}^{s_{2k+1}}(x) \to e_{n-1}$. And this takes place regardless of the starting point x. □

Two will be our goals for the rest of this section. First, to prove that condition (5.1) is essential for the convergence of heuristic $\mathcal{H}$—by showing that a weaker (in a certain sense) version of (5.1) does *not* ensure convergence. Second, we shall give an example of operator $\mathcal{G}$ which fulfils condition (5.1), and thus the hypothesis of convergence of Theorem 5.1.

Let us introduce a weaker form of condition (5.1), by allowing the sum counter on the right-hand side of the inequality to take also the value $p_s(x)$. That is,

$$\sum_{j\geqslant p_s(x)} \mathcal{G}_j(x) > 1 - \frac{1}{s+1}. \tag{5.2}$$

One can easily prove the following equivalent of Lemma 5.2.

Lemma 5.3. *Under the hypothesis of Theorem* 5.1, *the following inequalities hold:*

(1) $\sum_{i\leqslant p_s(x)} \mathcal{H}(x)_i < 1/(s+1)$,

(2) $p_{s+1}(\mathcal{H}(x)) > p_s(x)$.

Yet, as one will see below, this result is *not* strong enough in order to ensure convergence.

Theorem 5.4. *If for all $x \in \Lambda$ and all $s > 1$ condition* (5.2) *holds, then the iterates of heuristic $\mathcal{H}$ do not necessarily converge to e_{n-1}, not even on subsequences.*

P r o o f. Let us start with an $x \in \Lambda$ such that $p_2(x) < n-1$. According to Lemma 5.3 we have, for all $s \geqslant 2$, $\sum\limits_{i \leqslant p_s(x)} \mathcal{H}^s(x)_i < 1/(s+1)$ and $p_s(\mathcal{H}(x)) \geqslant p_2(x)$. This means that, when looking at the vector

$$\mathcal{H}^s(x) = \langle \mathcal{H}^s(x)_0, \mathcal{H}^s(x)_1, \ldots, \mathcal{H}^s(x)_{p_s-1}, \mathcal{H}^s(x)_{p_s}, \ldots \rangle ,$$

its first $p_s(x)$ components $\to 0$ as $s \to \infty$, but what happens to the rest? In the most unfavorable case allowed by condition (5.2), one can concentrate the whole mass of heuristic $\mathcal{H}$ on the $p_2(x)$ component, as $s \to \infty$. It is obvious that such a heuristic converges to $e_{p_2(x)}$, which is not e_{n-1}. □

Now let us turn again to condition (5.1) and give an example of heuristic $\mathcal{G}$ that satisfies the condition. However, we must admit that this example (and this stands for all the results in this section) is of purely theoretical interest only, that is, we do not expect the practical algorithms to meet the very strong assumption we formulate here.

Let $x = \langle x_0, x_1, \ldots, x_{n-1} \rangle \in \Lambda$ be an arbitrary starting point. Obviously, the sequence $\{p_s(x)\}_{s>1}$ is non-increasing, so one can define a function M on $\{0, 1, \ldots, p_2(x)\}$ in the following manner:

$$M(i) = \begin{cases} \max\{k\colon\ p_k(x) = i\}, \\ 0, \text{ if } \{k\colon\ p_k(x) = i\} = \emptyset. \end{cases}$$

Next, let us take the following pointwise definition for heuristic $\mathcal{G}$:

$$\mathcal{G}(x)_i = \begin{cases} 0, & \text{if } M(i) = 0, \\ \dfrac{1}{M(i)+2}, & \text{for the first } i \text{ such that } M(i) > 0, \\ \dfrac{1}{M(i)+2} - \dfrac{1}{M(h_i)+2}, & \text{if } 0 < i \leqslant p_2(x), M(i) > 0 \text{ and } h_i > 0, \\ \dfrac{M(i-1)+1}{M(i-1)+2}, & \text{if } i = p_2(x)+1, \\ 0, & \text{if } i > p_2(x)+1, \end{cases} \tag{5.3}$$

where, for each i, h_i is defined by

$$h_i = \begin{cases} \max\{j\colon\ j < i,\ M(j) > 0\}, \\ 0, \quad \text{if the set is empty.} \end{cases}$$

Let us illustrate the construction above by a numerical example.

E x a m p l e 2.

$p_2(x)$	$p_3(x)$	$p_4(x)$	$p_5(x)$	$\ldots$	$p_{24}(x)$	$p_{25}(x)$	$\ldots$	p_∞
10	10	8	7	7	7	1	1	1

i	0	1	2	3	4	5	6	7	8	9	10
$M(i)$	0	25	0	0	0	0	0	24	4	0	3
$\mathcal{G}(x)_i$	0	$\frac{1}{25+2}$	0	0	0	0	0	$\frac{1}{24+2} - \frac{1}{27}$	$\frac{1}{4+2} - \frac{1}{24+2}$	0	$\frac{1}{3+2} - \frac{1}{4+2}$

Turning back to the general case, we have the following.

Proposition 5.5. *If heuristic $\mathcal{G}$ is given by equation* (5.3)*, then $\mathcal{G}(x) \in \Lambda$.*

P r o o f. One has to show that the sum over all components of $\mathcal{G}$ is one. But this comes from

$$\sum_{i \leqslant p_2(x)} \mathcal{G}(x)_i = \frac{1}{M(p_2(x)) + 2},$$

which is a straightforward consequence of (5.3). □

One must notice that the dependence on x is very strong in the definition of heuristic $\mathcal{G}$, cf. formula (5.3). Also, we required a big jump from x to $\mathcal{G}(x)$, namely we moved the whole mass

$$\frac{M(p_2(x)) + 1}{M(p_2(x)) + 2}$$

(which can be very close to one) one position to the right; and this must happen for all $x \in \Lambda$!

Unfortunately, this condition is very hard to achieve for a practical algorithm, so one should prefer, instead of looking for *pure convergence*, the *bounded-convergence* property introduced earlier. In this concern, we conjecture that *bounded-convergence* of heuristic $\mathcal{H}$ is sufficient to ensure the convergence of the implemented finite population algorithm, with respect to the usual convergence definition employed by the finite population EA theory, see [9].

References

[1] *A. Agapie*: Modelling genetic algorithms: From Markov chains to dependence with complete connections. Lect. Notes Comput. Sci. *1498* (1998), 3–12.

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

[3] *A. Agapie*: Estimation of distribution algorithms on non-separable problems. Int. J. Comput. Math. *87* (2010), 491–508. zbl MR

[4] *A. Agapie, M. Agapie, G. Rudolph, G. Zbaganu*: Convergence of evolutionary algorithms on the n-dimensional continuous space. IEEE Trans. Cybern. *43* (2013), 1462–1472.

[5] *A. Agapie, M. Agapie, G. Zbaganu*: Evolutionary algorithms for continuous space optimization. Int. J. Syst. Sci. *44* (2013), 502–512.

[6] *L. Davis*: Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York, 1991.

[7] *B. Mitavskiy, J. Rowe, A. H. Wright, L. Schmitt*: Quotients of Markov chains and asymptotic properties of the stationary distribution of the Markov chain associated to an evolutionary algorithm. Genet. Program. Evolv. Mach. *9* (2008), 109–123.

[8] *G. Rudolph*: Convergence Properties of Evolutionary Algorithms. Verlag Dr. Kovać, Hamburg, 1997.

[9] *G. Rudolph*: Stochastic convergence. Handbook of Natural Computing (G. Rozenberg, T. H. W. Bäck, J. N. Kok, eds.). Springer, Berlin, 2012. zbl

[10] *G. Syswerda*: A study of reproduction in generational and steady state genetic algorithms. Foundations of Genetic Algorithms. San Mateo, Morgan Kaufman, San Francisco, 1991, pp. 94–101.

[11] *M. D. Vose*: The Simple Genetic Algorithm. Foundations and Theory. MIT Press, Cambridge, 1999. zbl MR

[12] *D. Whitley*: The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. Proceedings of the Third International Conference on Genetic Algorithms. Morgan Kaufman, San Francisco, 1989, pp. 116–123.

[13] *A. H. Wright, J. Rowe*: Continuous dynamical system models of steady-state genetic algorithms. Foundations of Genetic Algorithms—6. Proc. FOGA-6, Morgan Kaufmann Publishers, Orlando, 2002, pp. 209–225. zbl

Authors' addresses: *Alexandru Agapie*, Bucharest University of Economic Studies, Calea Dorobantilor 15-17, Bucharest, 010552, Romania, and Institute of Mathematical Statistics and Applied Mathematics, Bucharest e-mail: `agapie@clicknet.ro`; *Alden H. Wright*, Computer Science, University of Montana, Missoula, MT 59812 USA, e-mail: `wright@cs.umt.edu`.