Computer Assisted Mechanics and Engineering Sciences, 9: 135–149, 2002.

Simple taxonomy of the genetic global optimization

Robert Schaefer
Institute of Computer Science, Jagiellonian University
Nawojki 11, 30-072 Kraków, Poland

(Received August 8, 2001)

The paper tries to show the role that can be played by genetic optimization strategies in solving huge global optimization problems in computational mechanics and other branches of high technology. Genetic algorithms are especially recommended as the first phase in two-phase stochastic optimization. The self-adaptability of genetic search is shown on the basis of the mathematical model introduced by M. Vose. Main goals of adaptation are used as leading criteria in the simple taxonomy of genetic strategies.

Keywords: genetic algorithms, stochastic search, two-phase strategies

1. Introduction

Global optimization problems are one of most difficult numerical problems that find their origin in computational mechanics. The main difficulties come from such reasons as nonlinearity, the existence of many solutions, enormous computational and memory complexity and general bad conditioning. Genetic algorithms (GAs) and evolutionary strategies (EAs) constitute the group of method strongly developed in the last twenty years that can be helpful by solving ill posed global optimization problems. GAs and EAs are of a very delicate nature as the methods performing stochastic search with the variable sampling measure.

We try to show, that GAs and EAs are self-adaptive, ergodic searching processes by using the Markov theory of Simple Genetic Algorithm (SGA) introduced by Vose [24, 42, 43]. We also show why GA can not been a good local optimization method. The simple taxonomy of adaptive strategies in genetic search will be also introduced. It takes into account the goal of adaptation and the course and "depth" of modifications of basic genetic mechanisms.

The another possible way to overcome difficulties met in global optimization is to apply two-phase stochastic global optimization strategies. Roughly saying, they consist in performing the refined global stochastic search in the first phase, and the set of local convex optimizations (using maximum slope methods) in the second phase. The necessary result of the first phase is the set of starting points for local methods. Some information about the shape and volume of local attractors is also desired. We will discuss the group of adaptive GAs as the methods that can be good candidates for the first phase.

The paper can help to understand the real nature of GAs and EAs and their possible role in solving complex global optimization problems that appear in computational mechanics (e.g. optimal shape design problems). It may be also helpful by selecting the proper GA or EA adaptive policy to the particular problem.

2. Single- and two-phase stochastic global optimization strategies

Let us denote by $D \subset \Re^n$ the set of all admissible design parameters and by $Obj : D \to \Re$ the objective function, that may express the inverse of identification error or the maximum internal

energy or the global cost of the whole structure. We intend to maximize the objective, more correctly we intend to solve one of the following problems:

Find all pairs $(x, Obj(x)) \in D \times \Re$ (or pairs $(z, Obj(z))$ for z from the small neighborhood of x) for which:

- $Obj(x) \geq Obj(y) \forall y \in D$, or,
- $Obj(x)$ is the value of Obj at the local optimizer that satisfy some prescribed conditions (e.g. $Obj(x) \geq Obj_{threshold}$), or,
- $(x, Obj(x))$ belongs to the level set of Obj, or to the local attractor.

The global optimization problems that appears in the mechanics of continua exhibit the huge computational and memory complexity mainly because of:

- multimodality and the total lack of information about the behavior of Obj.
- usual low regularity of Obj (Obj is sometimes discontinuous),
- non-deterministic (fuzzy) character of Obj (e.g. in case if we try to scalarize the objective vector),
- The high cardinality of the set of all discrete representations of elements in D (e.g. meshes using for encoding the design parameters).

The method that we need for solving global optimization problems should comprehend the possibility of revising all local optimizers together with the sufficient local accuracy. Let us define the following two strategies:

Definition 1. *Single-phase stochastic algorithms consist in performing in the loop the following two steps:*

a) *make the sample set S which is the finite subset of D,*

b) *evaluate the set S.*

Example 1. Pure Random Search (PRS)

ad a) perform sampling of elements of the set S according to the uniform probability distribution on D,

ad b) sort elements in S according to the value of Obj.

Definition 2. *Two-phase stochastic strategies consist in performing in the loop the following three steps:*

a) *make the sample set S which is the finite subset of D,*

b) *reduce the sample set and make $S_r \subseteq S$,*

c) *perform local convex optimizations starting from points that belong to the selection $S_{rs} \subseteq S_r$.*

Example 2. Multistart

ad a) perform sampling of elements of the set S according to the uniform probability distribution on D,

ad b), c) $S_{rs} = S_r = S$.

Example 3. Clustering

ad a) perform sampling of elements of the set S according to the uniform probability distribution on D,

ad b) $S_r = \bigcup_{i=1}^{B} C_i$, where $C_i \subset S$ is the set of points that belong to the attractor of the i-th local optimizer of Obj,

ad c) $S_{rs} = \bigcup_{i=1}^{B} \{x_i\}; \quad x_i \in C_i; \quad Obj(x_i) \geq Obj(y), \quad y \in C_i, \quad i = 1, \ldots, B.$

Stochastic searching strategies have some significant advantages with respect to enumerative, deterministic methods. In particular they involve non-determinism and a weak regularity of Obj, moreover the random sample better covers the admissible set $D \subset \Re^n$ as regular meshes in $\Re^n$.

3. Adaptability as the preferred path towards the high efficient search

We intend to show the *adaptation strategies* in stochastic search as the external modification of a measure or the measure vector towards the measures highly concentrated in the central parts of basins of attraction. This evolving measure sequence is utilize for sampling making this process more effective. Another words the probability that a point in the sample is located close to the optimizer asymptotically growth, even if some modifications may decrease average objective for several iteration steps.

The *self-adaptive stochastic algorithms* can collect and store the information about the problem to be solved during iteration. This information mobilized the built-in mechanism of sampling measure modification which affects on the computational efficiency.

We try to explain the mechanism of genetic self-adaptability by using the Markov chain theory of Simple Genetic Algorithm introduced by Vose [24, 42, 43].

4. Simple Genetic Algorithm (SGA) as the Markov dynamic system

4.1. The background of SGA stochastic model

At the start of our consideration let us recall the main steps of Simple Genetic Algorithm (see e.g. Goldberg [18]):

A. Establish the uncoding and decoding mappings

$$\text{code} : \Omega \to D, \qquad \text{decode} : D \to \Omega \tag{1}$$

where D is the admissible set to a global optimization problem. The genetic space Ω is defined as the product group

$$\Omega = \underbrace{\mathbf{Z}_2 \times \ldots \times \mathbf{Z}_2}_{r \text{ times}} \tag{2}$$

where $\mathbf{Z}_2$ is the additive group of integers modulo 2. The group operators $\oplus$ and $\otimes$ denote component-wise addition and multiplication respectively. It is also convenient to treat $\Omega = [0, \ldots, r-1]$, $r = 2^l$ as the space of binary codes of the length l. We assume, that code is injective and decode surjective and they satisfied the coherency condition: decode $\circ$ code $= I_\Omega$.

B. Establish fitness f to be maximize. Assuming $Obj > 0$ we may put

$$f : \Omega \to \Re_+, \qquad f = Obj \circ \text{code}. \tag{3}$$

C. Generate initial population as a multiset P_0 of elements from Ω, $\#P_0 = n$.

D. Perform "genetic operations" that transform one population P_k to the next epoch population P_{k+1}:

- proportional selection,
- mixing operation e.g. multipoint mutation and single-point crossover,

until a stop criterion is satisfied.

Usually the stop criteria are as follows: the average fitness does not change (there is no progress in evolution) or an individual with a satisfactory fitness is found.

In order to pass to the Markov Chain model of SGA we unambiguously identify each population P of n elements from Ω with a frequency vector $x = [x_0, \ldots, x_{r-1}]$, $x_j \geq 0$, $\sum x_j = 1$, where x_i denotes the frequency of the individual of genotype $j \in \Omega$ in P. Frequency vectors for all populations of n individuals constitute a discrete subset X_n in the unit $r-1$ dimensional simplex $\Lambda^{r-1} \subset \Re^r$. SGA can be understood as a stationary Markov chain of states from X_n (see [24, 43]). Let us denote by π_n^k the probability distribution of the occurrence of n-sized population in the k-th evolution epoch. The Markov law of evolution consists in the linear dependence between the consecutive probability distributions

$$\pi_n^{k+1} = Q\pi_n^k \qquad k = 0, 1, 2, \ldots \tag{4}$$

where Q denotes the $\#X_n \times \#X_n$ transition probability matrix. Q does not depend on the evolutionary epoch k.

We define now the *genetic operator*

$$G = (F \circ M) : \Lambda^{r-1} \to \Lambda^{r-1}. \tag{5}$$

The component F is called the *proportional selection operator*

$$F(x) = (f, x)^{-1}(\operatorname{diag} f)x, \qquad f_i = f(i), \qquad i \in \Omega. \tag{6}$$

For any vector x representing some population in k-th evolution epoch $F(x)$ can be interpreted as a vector which component $F_i(x)$ is the probability, that the individual i is selected to next genetic operations.

The second component M in Eq. (5) called *recombination operator* comprehends both the mutation and crossover. M is a composition of quadratic forms,

$$M(x) = \left((\sigma_0 x)^T \mathbf{M} \sigma_0 x, \ \ldots, \ (\sigma_{r-1} x)^T \mathbf{M} \sigma_{r-1} x\right), \tag{7}$$

where $\sigma_j : \Lambda^{r-1} \to \Lambda^{r-1}$ is a permutation operator such, that $\sigma_j(x_0, \ldots, x_{r-1}) = (x_{j \oplus 0}, \ldots, x_{j \oplus (r-1)})$.

The $r \times r$ mixing matrix $\mathbf{M}$ depends only upon the mutation rate μ and crossover rate χ, $(\mu, \chi \in (0,1))$. μ is the probability of single bit inversion in each individual genotype $j \in \Omega$ while χ determines probability of crossover occurrence. We refer to [24] for details of mixing matrix computation with respect to μ and χ.

The essential of the Markov theory of SGA is the following:

Theorem 1. (Nix and Vose [24]) *Let x denote the population vector in the k-th evolutionary epoch. The expected population vector in the $(k+1)$-th evolution epoch is $G(x)$. Moreover, the transition matrix of the Markov chain can be expressed by*

$$Q_{xy} = n! \prod_{i=0}^{r-1} \frac{[(G(x))_i]^{r\, y_i}}{(r\, y_i)!}, \qquad x, y \in X_n\,. \tag{8}$$

Let us denote by $M(D)$ the space of probabilistic measures on the admissible set D. We may introduce the mapping

$$\Theta : \Lambda^{r-1} \to M(D) \tag{9}$$

such that for the arbitrary $A \subset D$ measurable set

$$\forall x \in \Lambda^{r-1} \quad \Theta(x)(A) = \sum_{I_A} x_i \tag{10}$$

where $I_A \subset \Omega$ is the maximum set of all genotypes such that code$(i) \in A$ for $i \in I_A$.

Observation 1. (see Schaefer, Jabłoński [30]) For $x \in X_n$ the measure $\Theta(x)$ is the discrete counting measure on D, and $\Theta(x)(A)$ for any Lesbegue measurable set $A \subset D$ may be calculated by counting the individual which genotypes $i \in I_A$ and normalizing the result by the population size n.

Observation 2. (see Vose [44]) If $x \in X_n$ is the population vector in the k-th genetic epoch, then the next population is obtained by the n-time sampling according to the measure $\Theta(G(x))$. Notice, that for any fixed point $z \in \Lambda^{r-1}$ of the operator G we have $\Theta(G(z)) = \Theta(z)$, so z can be understand as the limit sampling measure on D.

4.2. SGA as a dynamic system

The *dynamic semi-system* (see Pelczar [29]) may be characterized as the triple $(\Im, \mathrm{B}, \phi)$, where $\Im$ is a topological space, $(\mathrm{B}, +)$ the topological, alternating semigroup and $\phi : \Im \times \mathrm{B} \to \Im$ the mapping such that:

1. $\forall x \in \Im \quad \phi(\bullet, e) = I(\bullet)$ where e stands for the neutral element in B,
2. $\forall p, t \in \mathrm{B}, \ \forall x \in \Im \quad \phi(\phi(x,p),t) = \phi(x, p+t)$,
3. ϕ is continuous with respect to both variables.

If $n < +\infty$ (finite population) π_n^k is a occurrence probability distribution of occurrence of each population from X_n in k-th genetic epoch. So π_n^k is a probabilistic measure on $\Lambda^{r-1} \subset \Re^r$ such that π_n^k is concentrated on X_n. Taking into account Eq. (4) we have

$$\pi_n^{k+p} = Q^p \pi_n^k, \qquad Q^{k+p} = Q^k Q^p = Q^p Q^k, \quad \forall k, p \text{ integers},$$

then $\mathrm{B} = \{Q^p\}, p = 0, 1, 2, \ldots$, generates an alternating semigroup of transformations of probabilistic measures on X_n. All $\{\pi_n^k\}$, $k = 1, 2, 3, \ldots$, can be identified with vectors form the unit simplex in the $\Re^{\#X_n}$ space. We introduce $\Im$ as the space of discrete measures on $\Lambda^{r-1} \subset \Re^r$ concentrated on X_n with the topology induced from $\Re^{\#X_n}$. Each iterate Q^p, $p = 0, 2, 3, \ldots$, generates the continuous mapping $\Re^{\#X_n} \to \Re^{\#X_n}$, so B is a semigroup of continuous transformations on $\Im$. Let us define $\phi(x, Q^p) = Q^p x$, $x \in X_n$, $p = 0, 1, 2, \ldots$. It is proven, that ϕ is continuous with respect to the first variable. The regularity with respect to the second variable is trivial, because B is a discrete set and the continuity in discrete topology on B can be justified. Finally we have:

Observation 3. (see Schaefer, Telega, Kołodziej [32]) Finite population SGA can be understood as a dynamic semi-system in a set of probabilistic measures on $\Lambda^{r-1} \subset \Re^r$ concentrated on X_n.

In the case of finite population the passage from the population $x \in X_n$ to $y \in X_n$ is performed with the probability Q_{xy}. If the number of individuals is infinite ($n = +\infty$) a population $x \in \Lambda^{r-1}$ is replaced in the next epoch by $G(x)$ with the probability 1, according to the theorem of great numbers and the Theorem 1.

Now $\Im = \Lambda^{r-1} \subset \Re^r$ with an arbitrary topology induced from $\Re^r$ and $\mathrm{B} = \{G^p\}$, $p = 0, 1, 2, \ldots$ (G^0 is the identity on Λ^{r-1}). B is the alternating semigroup of transformations on Λ^{r-1}. G is continuous on Λ^{r-1} as the composition of continuous mappings (5) and (6), so B is the semigroup of continuous transformations on Λ^{r-1}. B may be also equipped with the discrete topology as a discrete set of transformations. If we put now $\phi(x, G^p) = G^p(x)$, $x \in \Lambda^{r-1}$, $p = 0, 1, 2, \ldots$, then we can obtain:

Observation 4. (see Grygiel [19]) Infinite population SGA constitutes a dynamic semi-system in $\Lambda^{r-1} \subset \Re^r$.

4.3. Asymptotic

Vose and Nix have proved the important results concerning limit points of genetic algorithm in case of finite and infinite populations (see Nix and Vose [24]):

Theorem 2. *If the mutation is non-zero $\mu > 0$ then the Markov chain which models SGA is ergodic (see e.g. Feller [12]), so for any initial distribution π_n^0 there exists a weak limit π_n*

$$\lim_{k \to +\infty} \pi_n^k = \lim_{k \to +\infty} Q^k \pi_n^0 = \pi_n \,.$$

Theorem 3. *When the size of population increases ($n \to +\infty$) then by the theorem of Prohorov (see also Feller [12]) there exists a subsequence $\{\pi_{n_\xi}\}$ in $\{\pi_n\}$ which converges weakly to some limit point π^*,*

$$\pi_{n_\xi} \xrightarrow{w} \pi^* .$$

Theorem 4. *If G is focused and K is the set of fixed points of G ($\forall x \in \Lambda^{r-1}$, $\lim_{p \to +\infty} G^p(x) \in K$) then $\pi^*(K) = 1$.*

The stability of fixed points of the operator G has been also studied in [42].

4.3. Approximation

Approximation results can be presented as answers to the following questions:

- *How close can SGA approach to K after a finite number of evolution epochs and if population is finite?*

  The answer is partially given by the lemma (see Telega [30], Telega, Schaefer, Cabib [9]):

  Lemma 1. *Let $K_\varepsilon = \{x \in \Lambda^{r-1};\ \exists y \in K$ such that $d(x, y) < \varepsilon\}$, d is the distance in $\Re^r$, then $\forall \varepsilon > 0$, $\forall \eta > 0$ $\exists$ integer N, $\exists$ integer $W(N)$ such that*

  $$\forall n > N, \quad \forall k > W(N) \qquad \pi_n^k(K_\varepsilon) > 1 - \eta.$$

  If G is focused, than a sufficiently large population can be concentrated arbitrary close to the set of fixed points K with an arbitrary high probability $1 - \eta$, after a sufficient number of evolution epochs.

- *How close we can approximate the limit sampling measure on the admissible set D?*

  The answer is given by the following lemma (see Schaefer, Jabłoński [30]) being the consequence of the Lemma 1 and Observations 1 and 2:

Lemma 2. $\forall \varepsilon > 0, \ \forall \eta > 0 \quad \exists$ *integer* $N, \ \exists$ *integer* $W(N)$ *such that*

$$\forall n > N, \ \forall k > W(N), \ \forall A \subset D$$

$$\textit{Lesbegue measurable set} \quad P\left(\left|\Theta(x^k)(A) - \Theta(z)(A)\right| < \varepsilon\right) > 1 - \eta$$

where x^k, $k = 0, 1, 2, \ldots$ *stand for the n-sized population vectors in the k-th evolution epoch.*

In other words, there is arbitrary large probability $1 - \eta$ that the counting measure determined by the current population x^k is sufficiently close to the limit sampling measure if the population size and the number of evolutionary epochs are sufficiently large.

- *Does a finite population SGA follow the trajectory of infinite one in the initial genetic epochs?*

The answer is given by the theorem (Vose and Nix [24]):

Theorem 5.

$$\forall T > 0, \quad \forall \varepsilon > 0, \quad \forall \gamma < 1 \quad \exists N > 0; \qquad \forall k \in [0, T] \quad n > N \Rightarrow \|x^k - G^k(x^0)\| < \varepsilon$$

with the probability at least γ. *Vectors* x^k, $k = 0, 1, 2, \ldots$ *stand for the n-sized population in the k-th evolution epoch.*

If we assume the finite time interval T, then the finite population SGA will follow infinite population one arbitrary close, with the arbitrary large probability during the epochs $k \in [0, T]$, if the population is sufficiently large.

5. Why GA cannot been the well posed local optimization routine?

The above theory, especially the Theorem 2 shows, that if SGA is well posed and ergodic, than there is no unique limit population for the infinite number of genetic epoch. Even if the global optimizer occurs in the particular genetic epoch P_k, there is no way to distinct such situation in order to stop the algorithm.

If SGA is ergodic and its parameters are well tuned to the problem, than it is able to penetrate the whole admissible domain D and the preferred (more probable) populations are ones that concentrate close to the optimizers (according to the limit probability distribution $\Theta(z)$, $z \in K$). The similar features may be conjectured for the wide class of genetic algorithms.

We may summarize the above discussion in form of the *genetic search paradox*:

If the genetic algorithm is convergent (in sense of Theorem 2) than it is not convergent (in the sense of local convergence of individuals). From the opposite side, if GA is locally convergent, that it is ill posed as the global optimization routine, because it does not check the whole admissible domain.

6. Main directions of GA's self-adaptability

The Genetic Algorithm is the stochastic search that allow to utilize more general coding (e.g. real valued one) and more general genetic operations than SGA (see Michalewicz [23]). We will extend the meaning of Ω to the general (not necessary binary) space of genotypes. The well known block diagram of GA is presented at the Fig. 1. The self-adaptability of GA's may be explained by the convergence $\pi_n^k \to \pi_n$ and the convergence of counting measures $\Theta\left(G(x^k)\right)$ to the limit sampling measure $\Theta(z)$, $z \in K$. It causes, that if we go ahead with genetic epochs, than the more probable sampling will prefer points that are close to the local optimizers. After T epochs GA's

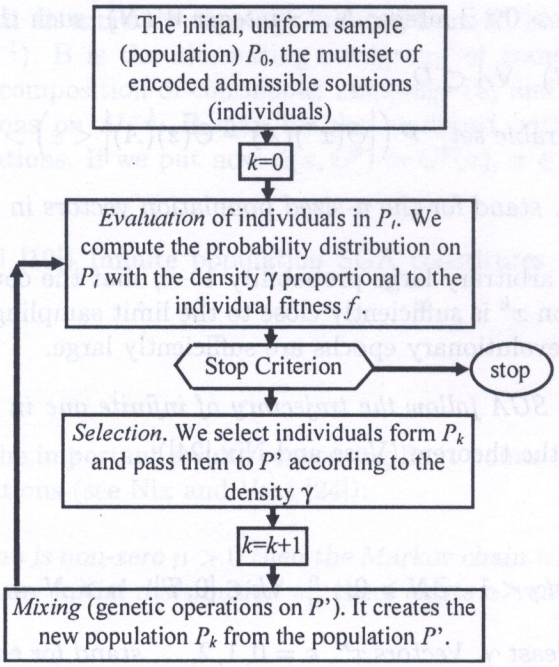

Fig. 1. The basic idea of genetic search

may be understood as the stochastic search, which sample $S_T = \bigcup_{k=1}^{T} P_k$. Assuming the uniform probability distribution for creating $P_0 = S_0$ we obtain quite different distribution of sampling in S_T for $T > 0$. It depends on the current fitness and the initial genetic parameter setting. If we assume, that the fixed points of G are populations sufficiently concentrated near optimizers, than the sampling in S_T is much more effective than in PRS after this same number of epochs.

Notice, that the adaptation of the measure is performed without any external control, by the constant values of genetic parameters. Moreover the convergence $\pi_n^k \to \pi_n$ do not force the monotone growth and convergence of the average and maximum fitness in evolving population.

The above theoretical discussion of GA's advantages show us two contradictory goals of self-adaptation for this class of algorithms:

- enlarge the increment of the average fitness in population,
- reinforce the ergodicity in order to effective check the whole admissible domain D.

7. The simple taxonomy of adaptive GA strategies

The general motivation for designing the adaptive GA strategies is to decrease significantly the computational complexity of genetic search. The adaptive GA strategies may be classified with respect to the *structure of evolving set of individuals*. We may distinct the class of low-structured (single or twin population) strategies, and the group of high-structured strategies operating as the colony of concurrent, collaborating populations.

The another possibility is to take into account the *main goal of adaptation*. This taxonomy partially follow main directions of self-adaptation mentioned previously. The adaptation strategy may strength only the maximum or average fitness improvement. The another possibility is to accelerate the population discrepancy or population mobility in order to pass saddles in optimization landscape and visit the whole admissible region. High-structured strategies are usually designed for total search which is performed everywhere in the admissible set D with a high local accuracy.

Let us denote by $\mathcal{A}$–$\mathcal{E}$ sets of GA adaptive strategies which belong do classes mentioned above (see Table 1). Although sets $\mathcal{A}$ and $\mathcal{B}$ are rather disjoint, there exist many adaptation techniques which may fall in to set $\mathcal{C}$ or $\mathcal{D}$ according to their parameters.

We present in the sequel several important approaches modifying classical GA model, and try to classify them using the taxonomy introduced above. We have intensively used the information and hints contained in monographs: Goldberg [18], Bäck, Fogel, Michalewicz [6] and Arabas [3].

Table 1. Various kinds of GA adaptive strategies

Taxonomy criterion	Sets of GA adaptive strategies		
Population structure	Single- or twin-population strategies $\mathcal{A}$		High-structured set of individuals $\mathcal{B}$
The goal of adaptation	Strength the maximum or average fitness improvement $\mathcal{C}$	Reinforce the population mobility or population discrepancy $\mathcal{D}$	Perform the total concurrent search $\mathcal{E}$

7.1. Single- or twin-population strategies

7.1.1. Dynamic parameter control of genetic operators

This strategy affects on a single population, so it belongs to the set $\mathcal{A}$. Each individual in a particular genetic epoch may be understood as the pair (i, s) where $x \in \Omega$ stands for its genotype and s is the set of parameters that control genetic operators which can transform i to i' in the next epoch. In the classical GA model s is constant. In case of dynamic parameter control the parameter setting obtain different values according to deterministic schedule prescribed by the user.

The main parameters to be controlled are the mutation parameters. They may depend on the number of generation k (see Fogarty [13]). References to another control methods may be found in Bäck [4]. This strategy generally falls in to the intersection of classes $\mathcal{A} \cap \mathcal{C}$.

7.1.2. Adaptive parameter control of genetic operators

This model of adaptation may be explained using the above representation (i, s) of the individual in the current genetic epoch as in previous case. New values of parameters s may be obtained by a feedback mechanism that monitors evolution and explicitly rewards or punishes operators according to their impact on the objective function rate.

Examples of such mechanism may be found in Davis [10] which adapt the operator probabilities according to their observed success in objective improvement. The similar mechanism were used by Stańczak [40]. The genetic operator which transform the particular individual is selected from the larger set of operators according to their efficiency in previous epochs. The efficiency parameters are updated after validation, at the end of each epoch.

The strategies described above are members of $\mathcal{A} \cap \mathcal{C}$. The next two examples of ESSS-SVA and ESSS-FDM algorithms (see Obuchowicz and Patan [26, 27], respectively) fall into class $\mathcal{A} \cap \mathcal{D}$ and leads to mobilize the population which concentrates near the local optimizers for the long time. The common assumption necessary for the successful applying of both techniques is the symmetry of the central part of attractor to be occupied.

ESSS-SVA (Evolutionary Search with Soft Selection – Simple Variance Adoption) accelerates the saddle crossing by adaptation of the modification radius. First the trap test is performed which

determine whether the population quality changes substantially for a given number of epochs. If the evolutionary trap is detected, then the mutation variance is enlarged, which disperse the population and allow to find and cross the saddle. If the evolutionary trap is not detected in the next steps, then the initial variance returns.

ESSS-FDM (Evolutionary Search with Soft Selection – Forced Direction of Mutation) consists in mutating the selected individuals by adding to each component a randomly distributed variable with a given variance and the non-zero expectation called "drift of expectation". The drift vector may be computed in the epoch k as $C^k(E(P^k) - E(P^{k-1}))$ where C^k is the positive normalizing number.

7.1.3. Self-adaptive parameter control of genetic operators

We will used the above representation (i, s) of individual in the evolving population. This mechanism is typically implemented by first recombining and mutating (according to the prescribed probability density function) the set s, yielding the set s', and next using the updated parameters s' to recombine and mutate the genotype i yielding i'. It is applied mostly for the unary operators like mutation.

Notice, that the transformation $s \to s'$ is performed locally for each individual, so it plays another role that the adaptation of the proportional selection operator in SGA, which depends on the information stored in population.

Several examples of self-adaptive mutation operators may be found in Bäck [5]. Although the strategy may satisfy various goals of GA adaptation, most examples belong to the class $\mathcal{A} \cap \mathcal{C}$.

7.1.4. Metaevolution

Metaevolution is the process of obtaining the optimal genetic algorithm that is, the best type of genetic operator, and their parameter setting for a given problem. The basic idea is to consider the search for the best GA as an optimization problem and use another GA to solve it. Another words, in metaevolutionary approach a metalevel GA operates on a population on basic GA's which in turn solve the problem under consideration.

This seems to be the generalization of the above dynamic-adaptive and self-adaptive parameter strategies. Operators as well as their parameters on a basic level may be selected and recombined globally.

More detailed characterization of this strategy and the review of metaevolutionary algorithms known from the literature are presented in [14]. Interesting example of genetic operators breeding which is performed in each $k > 1$ epoch of the basic level is given in [39]. The both mataevolutionary algorithms are of the class $\mathcal{A} \cap \mathcal{C}$.

7.1.5. Local methods as a part of stochastic genetic search

Local convex optimization methods can be twofold incorporated in to the genetic computation scheme (see Fig. 1). Let us denote by $L : D \to D$ the mapping which returns the outcome of local method $L(x)$ started at the point x.

The first possibility is to use the local method by the individual evaluation, so we put $Obj(L(\text{code}(i)))$ instead of $f(i)$ for each individual of the genotype $i \in \Omega$. This approach leads to flatten individual evaluation in area of local attractors (the so called Baldwin effect). It was described by Anderson [1], Whitley, Gordon, Mathias [45]. This effect allow to locate individuals close to the border of the attractor with this same probability as in the center, by presence of the high selective pressure. In consequence, the jump in to the neighboring better attractor is much more probable, so this strategy may be classified to the group $\mathcal{A} \cap \mathcal{D}$.

The second possibility may consist in particular in defining the new unary genetic operator called "gradient mutation". It replaces the individual of genotype $i \in \Omega$ by the individual of genotype decode(L(code(i))). We refer to Littman and Ackley [21] for the first result of this strategy. The interesting version of "gradient mutation" was implemented by Burczyński and Orantek [8] to the shape optimization op plastically deformable structures. The gradient operator was applied only to the one best individual in the population in each genetic epoch.

Also the crossover operator may be performed by using local optimization. Smith [37] introduce the special type of crossover in form of the single step of Nelder–Mead simplex method.

The second group of strategies which significantly strength local search fall in to group $\mathcal{A} \cap \mathcal{C}$ and its often called Lamarckian evolution.

7.1.6. Impatience operators

This strategy base also on the local fitness modification. It intends to disperse the population symmetrically with respect to the top of the local attractor if the trap-test is detected. It is performed by multiplying the fitness function of the particular individual by the coefficient which is proportional to the normalized distance between the phenotype and the "center of gravity" of the whole population. The crossover operator polarize the population into two subsets that are located on the opposite sides of local optimizer, and surround it during the next genetic epochs. If one of this subsets find the saddle, then the population can pass quickly to the basin of attraction of the better local optimizer. When the trap test is not satisfied, the impatience operator is deactivated. The impatience operator strategy was invented by Galar and Kopciuch [15] and can be classified as the element of $\mathcal{A} \cap \mathcal{D}$.

7.1.7. "Hill crunching" strategy

This strategy performs the fitness modification during the evolution. The trap test is performed in order to detect the long-term oscillation around the local optimizer, then the best individual is stored as the potential global solution and fitness is decreased in some neighborhood of the best individual (hill crunching). Such fitness modification make the population difficult to concentrate near the optimizers already recognized and push it to review another promising regions. Such kind of strategy was introduced by Beasley, Bull and Martin [7] in 1993 and was classified as the sequential niching procedure.

There are several methods of hill crunching. Obuchowicz [20] proposed in his ESSS-FDM (Evolutionary Search with Soft Selection – Deterioration of the Objective Function) decreasing the fitness by subtracting Gauss "hut" function with a center in the best individual.

Telega and Schaefer [33, 34] use hill crunching as the part of genetic clustering strategy. Clustering can be defined as the process of finding approximations to basins of attraction of local maxima. The main goal of this process is to reduce the number of local searches, perfectly to one in each basin (see e.g. [20] and references inside). However, sometimes it is also desirable to store and utilize rough information about basins.

Genetic clustering strategy consists in performing three steps in the loop:

a) performing GA steps that start from the uniformly distributed population P_0,

b) cluster recognition by density clustering which is performed using the current population as the measure estimator,

c) fitness modification which "crunches" clusters already recognized, until the global stop criterion is satisfied.

The point a) is stopped after SGA concentrates sufficiently on the basins (parts of basins). Another words the measure is locally sufficiently concentrated. Density clustering in the point b) consists in the following strategy: the domain D is divided into hypercubes. Hypercubes with the best individuals are seeds of clusters. Neighbor hypercubes are joined to clusters if the density of individuals inside them is greater than a certain threshold. In the point c) fitness is modified in such a way, that its value in the already recognized parts of clusters is set to the maximum value so far found. This is responsible for the pushing of individuals away from the clusters in the subsequent generation.

The algorithm should be stopped if there is no attractor that can be recognized by density analysis. For a class of fitness functions the global stop criterion can be defined as recognition of plateau outside of already known clusters. In this case the Cauchy-like convergence criterion can be justified on the base of presented theory if we assume that the plateau corresponds to the only fixed point of G in the center of Λ^{r-1}. More details can be found in the paper [9].

Approaches that contain hill crunching are rather classified in $\mathcal{A} \cap \mathcal{D}$.

7.1.8. Varying population size (life-time algorithms)

This is perhaps the oldest and most popular adaptive strategy in genetic computation. Population size is the important parameter that affects on the population mobility and the computational complexity. There are two contradictory opinions about the influence of the population size on the behavior of the genetic search. The first one represented by Michalewicz [23] finds large population better suited to the global search, because it can better fill the whole admissible set D. Small populations converge quickly, so they are more effective at the final phase of the search, when the basin of the global optimizer is already recognized. The second one introduced by Galar [16] underlines the high mobility of small populations. It is because a mutation of the single individual can affect on the whole population in several steps and then reinforce the saddle crossing. Large populations tends rather to occupy the basin of attraction of the local optimizer and the single individual mutation can not destabilized its equilibrium.

The main tool to obtain the varying population size is to introduce the new parameter which is called the individual life time. Examples of life time modification strategies leading to increase the population size at the introductory phase of computation and decrease it later are given in [23]. Another approaches may be found in [2, 35]. This algorithms may be classified in the set $\mathcal{A} \cap \mathcal{C}$. Obuchowicz and Korbicz in [25] set the life time proportional to the ratio between the current fitness and the maximum fitness from the previous epochs. Such setting avoids the precocious convergence of individuals. The population size increases when the average fitness increases and decreases when the population is trapped around the local optimum, or average fitness decreases. This strategy belongs to $\mathcal{A} \cap \mathcal{D}$.

7.1.9. Sharing function and another niching methods

This is the simplest strategy that allow to spread the single population and force the individuals to concentrate near more than one global optimizers. Individuals are penalized if they are bound one to each another. If this mechanism is well tuned, and fitness is locally convex, then individuals can concentrate only close to the global optimizers, and have large discrepancy out of their basins of attraction. Basic ideas of sharing function strategy are presented in [17].

Sharing function strategy may be generalized to the niching strategy which tends to spread population in to several niches, that can cover several local or global optima. Significant results in this direction (*crowding algorithms*) are obtained by De Jong [11]. Another concept of niching called *speciation method* was introduced by Spears [38].

Niching seems to be the part of the class $\mathcal{A} \cap \mathcal{E}$ of genetic algorithms.

7.2. Concurrent multipopulation cooperative strategies

7.2.1. Island model

The island model involves running several single population genetic algorithms in parallel. Each island is GA of this same type with its own population. Some individuals may migrate between population in order to improve the genetic material. The migration process may follow different topologies, and may involve different number of individuals. Whitley and Scott [46] applied SGA and GENITOR algorithm for each island and allow to migrate the single, best individual from each island according to the ring topology. Each island is started randomly and independently form each other, then it can potentially concentrate near different local optimizer. The experimental results that confirm this feature as well as the high efficiency of island strategy applied to linearly separable problems were presented in [46]. Whitley, Soraya and Heckendorn try to explain the mechanism of island collaboration basing on the Markov model for each island. This strategy belongs to $\mathcal{B} \cap \mathcal{E}$.

The review of another island strategies, their efficiency and applications may be found in Seredyński [36] and Martin, Lienig and Cohoon [22]. All this results show the high efficiency of the island strategy with respect to the multimodal problems that is mainly the issue of the concurrent search, not only the consequence op parallel implementation (implementation in the multiprocessor computer environment).

7.2.2. Hierarchic genetic algorithm

This is quite recent strategy introduced by Kołodziej and Schaefer [31] that is able to overcome evolutionary barriers and increase the efficiency of classical GA, especially in case of solving multimodal problems. It consists in running a parallel set of dependent genetic processes. The dependency relation among processes has a tree or a forest structure. The lowest order processes represent basic, chaotic search with lowest accuracy. The higher order ones represent more accurate search and are introduced when promising region on the optimization landscape has been found. The hierarchic genetic search aspire to be a member of $\mathcal{B} \cap \mathcal{E}$.

Main differences between island and hierarchic strategy may be stressed as follows:

- All island processes are active during the whole computation period. Hierarchic genetic algorithm performs only one process of most chaotic search with lowest accuracy up to the end of computation. Another processes are started only if the new promising region is found. Each higher order process is killed just after it founds the local optimizer, or recognizes plateau. Moreover two processes of this same order are reduces to the single search if they fall into this same region.
- Islands are introduces randomly. The higher then one order processes in hierarchic genetic algorithm are started by the lowest order ones by "sprouting" its best individual into the new population of longer genotype, that enables more effective and more local search.

8. Conclusions

1. Two-phase stochastic optimization strategies are well suited for solving inverse problems and optimization problems in mechanics.
2. The Markov model of SGA exhibits the real nature of the convergence of genetic search. It can not been the well posed local optimizer, because there is impossible to define the correct stop criterion (GA may pass the global optimizer, but we have no information when it happened).
3. Genetic algorithms transform measures in regular way, so they can be utilized in the first phase in two-phase global optimization stochastic strategies. In particular, they may effectively recognize attractors of the multimodal objective function.

4. The main advantage of genetic search seems to be the significantly greater efficiency then in the case of a simple stochastic search (e.g. PRS) by preserving the ergodicity. This is due to the self-adaptability feature of genetic algorithm (the sampling measure is automatically adopted in each step).

5. The computational efficiency of genetic algorithms may increase if the proper adaptation techniques are applied. There are three basic directions in GAs adaptation: fasten the local optimizer recognition, mobilize the population in order to pass the nearest saddle in the optimization landscape and perform the concurrent global search.

6. The multipopulaion genetic strategies are much better suited to solve multimodal problems then the single- or twin-population ones. It is mainly due to the internal concurrent search that is performed even in case of the serial, single processor implementation.

7. Another possibility to increase the computational efficiency is to utilize the multiprocessor computer environment. This possibility is discussed in the paper [36].

ACKNOWLEDGEMENTS

Research supported in part by Polish Scientific Committee (KBN) Grant No. 1191/T07/18

REFERENCES

[1] R.W. Anderson. The Baldwin effect. C3.4.1. in [42].
[2] J. Arabas, Z. Michalewicz, J. Mulawka. GAVaPS – a genetic algorithm with varying population size. *Proc. of the 1st IEEE Conf. on Evolutionary Computation, Orlando, FL, June 1994*, pp. 73-78. Piscataway, NJ, IEEE, 1994.
[3] J. Arabas. *Lectures on Evolutionary Algorithms* (in Polish). WNT, 2001.
[4] T. Bäck. Mutation parameters. E1.2 in [6].
[5] T. Bäck. Self-adaptation. C7.1 in [6].
[6] T. Bäck, D.B. Fogel, Z. Michalewicz, eds. *Handbook of Evolutionary Computation.* Oxford Univ. Press, New York, Oxford, 1997.
[7] D. Beasley, D.R. Bull, R.R. Martin. A sequential niche technique for multimodal function optimization. *Evolutionary Computation*, 1(2): 101–125, 1993.
[8] T. Burczyński, P. Orantek. The composition of genetic and gradient algorithms (in Polish). *Proc. of the KAEGiOG'99, Potok Złoty 1999*, pp. 47–54, 1999.
[9] Cabib, R. Schaefer, H. Telega. A parallel genetic clustering for inverse problems. *Lecture Notes in Computer Science* 1541, pp. 551–556. Springer, 1998.
[10] L. Davis. Adapting operator probabilities in genetic algorithms. In: J.D. Schaeffer, ed., *Proc. of 3rd Int. Conf. on Genetic Algorithms, Fairfax, VA, June 1989*, pp. 61–69. San Mateo, CA, Morgan Kaufman, 1989.
[11] K.A. De Jong. *An Analysis of the Behavior of a class of Genetic Adaptive Systems.* PhD Thesis, University of Michigan, 1975.
[12] W. Feller. *An Introduction to Probability, Theory and its Applications*, vol. I, 3rd ed., vol. II, 2nd ed. J. Wiley, NY, 1968, 1971; Polish ed.: PWN, Warszawa, 1987.
[13] T.C. Fogarty. Varying the probability of mutation in the genetic algorithm. In: J.D. Schaeffer, ed., *Proc. of 3rd Int. Conf. on Genetic Algorithms, Fairfax, VA, June 1989*, pp. 104–109. San Mateo, CA, Morgan Kaufman, 1989.
[14] B. Friesleben. Metaevolutionary approaches. C7.2 in [6].
[15] R. Galar, R. Kopciuch. Impatience and polarization in evolutionary processes (in Polish). *Proc. of the KAEGiOG'99, Potok Złoty 1999*, pp. 115–122, 1999.
[16] R. Galar. The practice of governing of research and development processes in light of experience in evolutionary algorithms (in Polish). *Proc. of the KAGiOG'97 (Tutorial), Rytro 1997*, pp. 7–22, 1997.
[17] D. Goldberg, J. Richardson. Genetic algorithms with sharing for multimodal function optimization. Genetic Algorithms and their Applcations. *Proc.of the 2nd Int. Conf. on Genetic Algorithms*, pp. 41–49, 1987.
[18] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Adisson Wesley, 1989.
[19] K. Grygiel. On asymptotic properties of a selection-with-mutation operator, *Proc. of the 1st Polish Conference on Evolutionary Algorithms, Murzasichle*, 1996.

[20] R. Horst, P.M. Pardalos. *Handbook of Global Optimization.* Kluwer Academic Publishers, 1995.
[21] M. Littman, D. Ackley. Adaptation in constant utility nonstationary environment. *Proc. Of ICGA'91 Conf.* Morgan Kaufman Publisher, 1991.
[22] W.N. Martin, J. Lienig, J.P. Cohoon. Island (migration) models: evolutionary algorithms based on punctuated equilibria. C6.3 in [6].
[23] Z. Michalewicz. *Genetic algorithms + Data Structures = Evolution Programs.* Springer, 1996.
[24] A.E. Nix, M.D. Vose. Modeling genetic algorithms with Markov chains. *Annals of Mathematics and Artificial Intelligence*, **5**(1): 79–88, 1992.
[25] A. Obuchowicz, J. Korbicz. Evolutionary search with soft selection algorithms in parameter optimization. *Proc. of the Conf. PPAM'99, Kazimierz Dolny, Poland*, 1999.
[26] A. Obuchowicz, K. Patan. An algorithm of evolutionary search with soft selection for training multilayered feedforeward naural networks. *Proc. of 3rd Conf. Neural Networks and their Application, KSN'97, Kule, Poland*, pp. 123–128, 1997.
[27] A. Obuchowicz. Adaption in time-varying landscape using an evolutionary search with soft selection algorithm. *Proc. of the KAEGiOG'99, Potok Złoty 1999*, pp. 245–251, 1999.
[28] A. Obuchowicz. The evolutionary search with soft selection and deterioration of the objective function. *Proc. of the 6th Int. Symp. Intelligent Information Systems, IIS'97, Zakopane, Poland*, pp. 288–295, 1997.
[29] A. Pelczar. *Introduction to the Theory of Differential Equations, Part II: The Qualitative Theory of Differential Equations*, (in Polish). PWN, Warszawa, 1989.
[30] R. Schaefer, Z.J. Jabłoński. On the convergence of sampling measures in the global genetic search. Submitted to the *Proceedings of PPAM'01*, September 2001.
[31] R. Schaefer, J. Kołodziej, R. Gwizdała, J. Wojtusiak. How simpletons can increase the community development – an attempt to hierarchical genetic computation. *Proc. of the KAEGiOG'2K, Lądek Zdrój 2000*, pp. 187–198, 2000.
[32] R. Schaefer, H. Telega, J. Kołodziej. Genetic algorithm as a Markov dynamic system. *Proc. of the Int. Conf. on Intelligent Techniques in Robotics, Control and Decision Making, 22-23 February 1999*, pp. 82-85. Polish-Japanese Institute of Information Technology Press, Warsaw, 1999.
[33] R. Schaefer, H. Telega. Advantages and drawbacks of a genetic clustering strategy. *Proc. of the KAEGiOG'99, Potok Złoty 1999*, pp. 291–300, 1999.
[34] R. Schaefer, H. Telega. Testing the Genetic Clustering with SGA evolutionary engine. *Proc. of the KAEGiOG'2K, Lądek Zdrój 2000*, pp. 227–236, 2000.
[35] D. Schlierkamp-Voosen, H. Müchlenbein. Adaptation of population sizes by competing subpopulation. *Proc. of the 1st IEEE Conf. on Evolutionary Computation, Orlando, FL, June 1996*, pp. 330–335. Piscataway, NJ, IEEE, 1996.
[36] F. Seredyński. New trends in parallel and distributed evolutionary computing. *Fundamenta Informaticae*, **35**: 211–230, 1998.
[37] S. Smith. The simplex method and evolutionary algorithms. *Proc of ICEC'98 Conf., Alaska USA, May 1998.* IEEE Press, 1998.
[38] W.M. Spears. Simple subpopulation schemes. *Proc. of 3rd Annual Conf. on Evolutionary Programming, San Diego, CA, 1994*, pp. 296–307. World Scientific, 1994.
[39] J. Stańczak. *Concept and Algorithms Development for Self Adapting Evolutionary Systems*, (in Polish). PhD Thesis, Warsaw University of Technology, Warszawa, 1999.
[40] J. Stańczak. Evolutionary algorithm with population of "smart" solutions (in Polish). *Proc. of the KAEGiOG'2K, Lądek Zdrój 2000*, pp. 207–218, 2000.
[41] H. Telega. *Parallel Algorithms for Solving Selected Inverse Problems* (in Polish). PhD Thesis, Academy of Mining and Metalurgy, Cracow, Poland, 1999.
[42] M.D. Vose, A.D. Wright. Stability of vertex fixed points and applications. *Evolutionary Computations*, 1995.
[43] M.D. Vose. Formalizing genetic algorithms. *Proc. IEEE wrksp. on GAs, NNs & SA Applied to Problems in Signal and Image Processing.* Glasgow, U.K., May 1990.
[44] M. Vose. *The simple genetic algorithm.* MIT Press 1999.
[45] D. Whitley, V. Gordon, K. Mathias. Lamarckian evolution, the Baldwin effect and function optimization. In: Y. Davidor, H.P. Schwefel, eds., *Proc. Of PPSN'94, Berlin, July 1992.* Springer-Verlag, 1992.
[46] D. Whitley, R. Soraya, R.B. Heckerdorn. Island model genetic algorithms and linearly separable problems. *Proc. of AISB 1997 Workshop on Evolutionary Computing, Manchester*, pp. 112–129, 1997.