CLUSTER ANALYSIS OF GENETIC ALGORITHM RESULTS

KATARZYNA ADAMSKA
INSTITUTE OF COMPUTER SCIENCE, JAGIELLONIAN UNIVERSITY
NAWOJKI 11, 30-072 KRAKOW
POLAND
ADAMSKA@II.UJ.EDU.PL

Summary

The work is concerned on the problem of approximation of central parts of basins of attraction of an objective in continuous global optimization problems. It presents the general strategy of Clustered Genetic Search (CGS), which consists in finding clusters in a genetic sample to get the approximations of basins of attraction of an objective. DR-CGS is an instance of CGS which utilizes a construction of a Finite Mixture Model of normal componenets as a clustering method. CR-CGS brings wide opportunities of asymptotic analysis. Due to features of a normal mixture, it also allows for very easy definition of approximations of basins of attraction Presented computational tests illustrate how the method works and are a practical evidence of its good results.

Key words: genetic algorithms, global optimization, clustering, finite mixture model

1. Motivation

This work[1] is concerned on a problem of an approximation of central parts of basins of attraction of an objective in continuous global optimization problems. Knowledge about basins of attraction can be useful at least due two the following reasons:

- Once basins are located, any local gradient method can be run inside them. It is guaranteed that such a method will be convergent and will lead in a few steps to an exact solution. In that sense, looking for basins of attraction can be treated as a first of two steps of solving global optimization problem.
- Approximate shape and location of basins of attraction give an opportunity of an objective's landscape examination. Information about an objective's landscape is crucial in stability analysis of solutions. It is important in many applications; as examples we can mention the computational modeling of chemical reactivity or engineering computation of stability and durability of constructions.

As a tool for approximation of basins of attraction we propose a new strategy of Clustered Genetic Search (CGS). Some introductory considerations and tests can be found in author's previous work [7]. This work defines CGS as a general class of methods which make use of cooperation between a genetic algorithm and a clustering method to achieve the goal of approximating basins of attraction. While using genetic algorithms, it can be hardly expected to find an exact solution to an optimization problem. However, genetic algorithms generate populations of approximate solutions. We use a class of genetic algorithms which tend to concentrate approximate solutions inside central parts of basins of attraction of local maximizers. This feature differs genetic algorithms from classical gradient optimization methods and makes them a useful tool for CGS. The clusters of genetic individuals can be detected with the use of a selected approach from a wide range of clustering methods. The clusters become a base for determining approximations of basins of attraction.

[1] The work is sponsored by Jagiellonian University grant.

Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. Vol. 9 No 28 (2005), pp. 25-32.
ISSN: 1137-3601. ©AEPIA (http://www.aepia.org)

2. The concept of Clustered Genetic Search

As mentioned in the introduction, Clustered Genetic Search is a general name for a class of methods which utilize cooperation between a genetic algorithm and a cluster analysis tool to find approximations of basins of attraction of an objective in a global optimization problem. There can be many instances of CGS, depending on particular algorithms selected to use. Some of them are discussed in the next sections. This section is to present a general formal definition of CGS strategy, which is independent from any particular algorithms.

Let us start with some assumptions and introductory definitions. Our deliberation is limited to continuous problems of minimization which can be converted to equivalent maximization problems. We consider an objective $\Phi : D \to \mathbb{R}$, $D \subset \mathbb{R}^n$, $m \leq \Phi(x) \leq M, \forall x \in D$ which has isolated minimizers only in the interior of D.

Definition 2.1. A multiset is a pair $M = (M^*, f)$ where M^* is a generator set and $f : M^* \to \mathbb{N}$ is a multiplicity function.

Intuitively, a multiset is a set which an element can be included more then once in. A multiplicity function defines the number of occurrences of an element in a multiset. In particular, if a generating set is finite, the cardinality of a multiset will be stated as $card\, M = \sum_{x \in M^*} f(x)$.

Definition 2.2. Let us introduce a relation between a multiset $M = (M^*, f)$ and a set B:

$$M \prec B \iff M^* \subset B$$

A genetic algorithm operates on a population which is a multiset of individuals from some fixed genetic universe U. In the sense of the relation 2.2 a population $P \prec U$.

For a given admissible set D, we define a phenotype set $\widetilde{D} \subset D$ such that $card\, U = card\, \widetilde{D}$. Once $\widetilde{D}$ is fixed, we can introduce a bijective mapping between genotypes and phenotypes:

$$code : U \hookrightarrow \widetilde{D} \tag{2.1}$$

It will be useful to define an analogous $code^*$ mapping between multisets:

Definition 2.3. Let $\mathbb{M}(U), \mathbb{M}\left(\widetilde{D}\right)$ stand for sets of all m-elemented multisets over U and $\widetilde{D}$, respectively.

$$code^* : \mathbb{M}(U) \to \mathbb{M}\left(\widetilde{D}\right)$$

is such a mapping that $\forall M_1 = (M_1^*, f_1) \in \mathbb{M}(U)$

$$code^*(M_1) = (M_2^*, f_2) \iff$$

$$code(M_1^*) = M_2^* and \forall x \in M_1^* \; f_1(x) = f_2(code(x))$$

Definition 2.4. Let $P \prec U$ stand for a population. A multiset $code^*(P) \prec \widetilde{D}$ is called a genetic sample.

A genetic algorithm used for CGS is required to be well-tuned (see sect. 4), which means that genetic samples in succeeding epochs tend to gather in central parts of basins of attraction. A genetic sample is an input data for a clustering algorithm. In CGS a genetic algorithm can be named a *genetic engine* which plays a role of input data generator for a clustering. Recognition of approximation of basins of attraction is done using clustering.

Let us introduce the following equivalence relation:

Definition 2.5. ([7]) Let $\Phi : D \to \mathbb{R}$ be a continuous multimodal objective, and $X \prec \widetilde{D} \subset D$ is a genetic sample. Two elements $x_i, x_j \in X^*$ are in relation R, $x_i \, R \, x_j$, if and only if x_i and x_j belong to the same basin of attraction of a local minimizer of the function Φ or they don't belong to any basin of attraction.

The above relation divides a genetic sample generator set into equivalence classes which correspond to distinct basins of attraction. There can be also a class which includes elements of a generator set that are out of all basins of attraction. The clustering in CGS means creating multisets $X_1, \ldots, X_k$ such that $\forall X_i \; X_i^* \in X^*/R$ and $f_i^* = f \, | X_i^*$.

However, clusters are discrete finite sets included in an admissible set D so they are not sufficient as approximations of basins of attraction. Thus, CGS strategy employs another step. This step is a definition of cluster extensions.

Definition 2.6. If $X_1, \ldots, X_k$ are clusters defined for a genetic sample X, such that $\forall i \; X_i$ includes phenotypes that belong to some basin of attraction then cluster extensions are sets $D_1, \ldots, D_k$ such that $\forall 1 \leq i \leq k \; X_i \subset D_i \subset D$, D_i are dense sets in D and they are pairwise disjoint.

As mentioned, there can be one cluster that includes phenotypes which are not included in any basin of attraction. There is no cluster extension defined for such a cluster. If such a cluster exists, the following idea can be introduced:

Definition 2.7. Let $X_1, \ldots, X_k$ be clusters defined for a genetic sample X and without loss of generality, X_k is the cluster whose elements are not in any basin of attraction. A cluster background is a set

$$D \setminus \bigcup_{i=1}^{k-1} D_i$$

where D_i are cluster extensions for clusters X_i, $i = 1, \ldots, k-1$.

3. An Instance of Clustered Genetic Search strategy

The ideas presented in the previous chapter constitute general rules of CGS strategy. They do not force any particular algorithms to be used. To define of a concrete CGS method a particular clustering tool and a genetic engine must be selected. The first instance of CGS was Hill Crunching CGS defined and analyzed by Telega [10, 7]. The method utilizes Simple Genetic Algorithm (SGA) as a genetic engine. Clustering is performed with the use of specific author's method based on the idea of sequential niching. Hill Crunching CGS provides very easy-interpretable approximations of basins of attraction, however the method becomes too complex and thus impractical while the dimension of a problem grows. That was the motivation to look for another approach to CGS practical performance.

This section is to describe selected methods of a class named Density Restoring CGS (DR-CGS). DR-CGS is a subclass of CGS where the clustering job is done by constructing a Finite Mixture Model (FMM, see sect. 3.2). A genetic engine in DR-CGS methods can be selected arbitrary. A general scheme of DR-CGS consist of three consecutive steps:

1. generate genetic sample using a selected genetic engine
2. cluster a genetic sample by constructing a Finite Mixture Model (FMM)
3. determine cluster extensions on the base of FMM parameters

For DR-CGS methods with a genetic engine is based on the Simple Genetic Algorithm (SGA) scheme there is an opportunity of theoretical analysis, which will be featured in section 4. In the examples presented in this work we utilize a genetic engine the Hierarchical Genetic Strategy (HGS, see the next section) and the evolutionary version of this algorithm (HGS-RN).

3.1. Genetic engine. Hierarchical Genetic Strategy (HGS, see [9]) is an effective multideme parallel global optimization algorithm. Calculation in HGS goes synchronously on several levels. For each level a set of parameters is defined. The higher level is, the lower mutation rate, the smaller population and the longer chromosome is set. As a consequence, low level populations have a sufficient mobility and play a role of a control system, which searches a domain and refers populations of a higher level into areas which seem to be interesting. Populations on high levels are responsible for more accurate calculations and are expected to gather close to extrema. Every single population evolves according to SGA rules. Due to this fact, the theoretical analysis (see sect 4) of DR-CGS with this algorithm is possible. A run of HGS can be characterized as follows:

The computation starts with a single population of the lowest level 1 called root. After the fixed number of evolution epochs the best adapted individual is selected. This procedure is called a metaepoch of the fixed period. The selected individual becomes a seed of a new population of the second order. This procedure is called sprouting operation. Sprouting can be generalized to higher accuracy levels, up to some fixed value. Metaepoch calculation is continued for all existing populations. Sprouting is performed conditionally, according to the outcome of the branch comparison operation. Details of both operations depend on the HGS implementation. After stopping the algorithm, one gets on the highest accuracy level a set of populations occupying neighborhoods of different extrema in an admissible domain. In compare to other genetic algorithms HGS is noteworthy faster and enables finding many local extrema in a single run.

The first of the examples presented in section 5 has bee calculated using DR-CGS with HGS genetic engine. The second employs HGS-RN where a single population evolves according to Evolutionary Algorithm (EA) with floating point coding.

3.2. Clustering tool. On the contrary to freedom of choice of genetic algorithm, a clustering tool in DR-CGS is strictly determined. As a clustering tool, DR-CGS methods employ the method of construction a Finite Mixture Model (FMM, see for ex. [4]) with normal components.

The construction of a finite mixture model is a statistical method. Its base idea is that distribution of clustered data can be described with some density function ρ. This density function is supposed to be represented by the convex combination of some components (3.1):

$$\rho(x;q) = \sum_{k=1}^{s} \gamma_k g_k(x;q_k) \tag{3.1}$$

where $g_k(x;q_k)$ stands for a component function, depending on input data x and a set of specific parameters q_k. $\rho(x;q)$ is the function of the argument $x \in D \subset \mathbb{R}^n$, q denotes a collective vector of all parameters to be estimated. Coefficients γ_k in (3.1) are called mixing proportions and the function ρ is named a mixture. Each component should describe one cluster. The number of clusters s must be predicted.

The functional form of components g_k is assumed, but their specific parameters remain unknown. As in the most practical works, in this paper components are assumed to be the normal distributions. Thus a covariance matrix C_k and a mean vector m_k are characteristic parameters of a k^{th} component, $q_k = (C_k, m_k)$. The mixing proportions γ_k have to be also evaluated. Thus a parameter vector q in (3.1) is an object composed of all mixing parameters, mean vectors and covariance matrices; $q = (\gamma_1, m_1, C_1, \dots, \gamma_s, m_s, C_s)$. In case of DR-CGS data to be clustered is produced from a genetic sample X by taking $\forall x \in X^*$ $f(x)$ copies of the element x.

The mixture parameters are estimated according to maximum likelihood estimation approach. We denote with L the likelihood function to be maximized. The practical calculation method is the EM algorithm [2]. The EM algorithm operates on two sets of parameters. The first set is an auxiliary probability matrix $\Delta = [\delta_{ij}]_{j=1\dots m}^{i=1\dots s}$ where δ_{ij} stands for a probability that j-th data element belongs to an i-th cluster. The second set is the vector q of mixture (3.1) characteristic parameters. The both sets of parameters are dependent on each other through a Bayes rule, such that given one of them the second can be unambiguously calculated. A rough scheme of the EM can be drafted as follows:

```
1 assign initial values for probability
matrix $\Delta$

2 repeat

    2.1 compute mixture parameters $q$
using $\Delta$ (M-step)

    2.2 compute expected values of $\Delta$
entries using $q$ (E-step)

    2.3 compute likelihood L

3 until change of L <$\epsilon$
```

3.3. Cluster extensions. The location of cluster extensions is quite easy in DR-CGS due to the features of the normal components. There are as many cluster extensions as the number of components of a mixture and for each component a cluster extension is determined individually. Cluster extensions are level sets of a mixture density. To define a cutting level, we have to determine points where mixture density function changes the most rapidly. Mixture values in these points define a cutting level. The reason for this approach will be given in sect. 4.

Let us consider for example the easiest one-dimensional case. A derivative of a normal distribution function has extremal points in $m+\sigma$ and $m-\sigma$, where m is a mean value and $\sigma$is a standard deviation. Set of cluster extensions for one-dimensional mixture is a set of intervals $\{[m_k-\sigma_k, m_k+\sigma_k]\}_{k=1}^{s}$. For a two-dimensional normal distribution $\rho(x_1, x_2)$ a cluster extension is an ellipse with a center in a point (m_1, m_2) and major sub-axes of the length σ_1 and σ_2. The points on the ends of ellipse's axes are extremal points of derivatives of the limit distributions (in eigenvectors coordinate system). The ellipse is rotated around its center with an angle θ which is dependent on covariance value between $x_1$and x_2; $\theta = 0{,}5 \arctan\left(\frac{-2\sigma_1\sigma_2}{\sigma_2^2-\sigma_1^2}\frac{cov\, x_1x_2}{var\, x_1 var\, x_2}\right)$.

In a general n-dimensional case, a cluster extension for a given component $\rho_k(x)$ is an n-dimensional hyper-ellipsoid. The center of ellipsoid is a mean vector. Subaxes of the hyper-ellipsoid are calculated in the eigenvectors coordinate system $\{\xi_i\}$ and then the hyper-ellipsoid is rotated into the original system using a transition matrix. End points of an i-th subaxis of an ellipsoid in coordinate system $\{\xi_i\}$ are extremal points of i-th limit distribution derivative $D\rho_k(\xi_i)$.

4. Notes on asymptotic analysis

Asymptotic analysis of DR-CGS properties refers to the case of SGA and SGA-based algorithms, where genetic universum is a finite set of binary strings of a fixed length l:

$$U = \Omega = \{(a_0, \dots a_{l-1}) : a_i \in \{0,1\} \, \forall i = 0, \dots, l-1\}$$

We reserve the symbol Ω specially for the binary universum. The analysis is based on the SGA model introduced by Vose [12]. Let us denote the cardinality of the universum with $r = 2^l$. Every population over Ω can be represented with a frequency vector $\omega = (\omega_0, \dots, \omega_{r-1})$, where ω_i, $i = 0, \dots, r-1$, is a contribution of the i-th genotype to a given population. Genotypes are supposed to be numbered according to their decimal values. The frequency vector can be treated as a discrete probabilistic measure on Ω. Due to the bijective coding (2.1) between genotypes and phenotypes, that measure can be transported into phenotype set $\widetilde{D}$, which will be here denoted D_r to emphasize that in this particular case it is a finite r-elemented set. It has been shown that such a discrete measure over D_r can be extended to a probabilistic measure with a density d of a class L^p or H^1, defined over a whole admissible set D ([8, 1]). The density d is named a *brightness* function.

The idea of brightness is used to define a well-tuning condition. Well-tuning is a feature of a genetic algorithm. Only well-tuned genetic algorithms are valid for CGS strategy. Note, that the brightness is a kind of continuous representation of a genetic sample distribution on an admissible set. Brightness has big values in these areas where the genetic sample concentration is high. A genetic algorithm is well-tuned if it can produce such a population that its brightness is significantly higher in central parts of basins of attraction (formal definition can be found in [7, 5]). Areas of high brightness values are defined in two ways, depending on the brightness type. For a brightness of a class L^p there are level sets for some arbitrary threshold. For a H^1 brightness we define contours according to extremal values of the brightness derivatives and consider sets bordered by these contours.

DR-CGS is an interesting approach, since constructing FMM can be treated as "restoring" the brightness. This is why cluster extensions are defined as presented in sect. 3.3. By comparing a mixture density and a brightness we can define the error of DR-CGS approach. Let us now draft how this comparison is done.

According to Vose's model, the evolution of SGA population can be described with a heuristic operator:

$$G : \Lambda^{r-1} \to \Lambda^{r-1}$$

where Λ^{r-1} is r-dimensional simplex that includes all population vectors. Additionally, we have to assume G is focused (see for ex. [12, 8, 7]), which means its set of fixed points $K \neq \emptyset$ and $\forall \omega \in \Lambda^{r-1} \, \exists z \in K : \text{lím}_{t\to\infty} G^t(\omega) = z$. For the purpose of CGS analysis we suppose $K = \{z\}$.

The asymptotic analysis of DR-HGS consists in comparison between a normal mixture density obtained from EM computation and the brightness distribution that corresponds to a population z of a fixed point of G, i.e. $G(z) = z$. We define our comparison as:

$$\left|d_z - \rho^i\left(Ey_m^t\right)\right| \tag{4.1}$$

where d_z is a brightness for a population of a fixed point z and $\rho^i\left(Ey_m^t\right)$ is normal mixture density calculated after i EM steps for an expected m-elemented population after t epochs. The difference (4.1) can be evaluated as follows:

$$\left|d_z - \rho^i\left(Ey_m^t\right)\right| \leq \tag{4.2}$$

$$\left|d_z - \rho^*(z)\right| + \left|\rho^*(z) - \rho^*\left(Ey_m^t\right)\right| + \left|\rho^*\left(Ey_m^t\right) - \rho^i\left(Ey_m^t\right)\right|$$

In (4.2) $\rho^*(z)$ stands for a normal mixture in a fixed point of EM algorithm defined for a population of a fixed point z of G and $\rho^*\left(Ey_m^t\right)$ denotes a normal mixture in a fixed point of EM algorithm defined for an expected m- elemented population after t epochs.

It can be shown (details in [6]) using Banach theorem about a fixed point of contraction that the last component in (4.2) can be arbitrary small if i is enough large. Moreover, the second component on the right side of (4.2) can also be arbitrary small for large enough values of m and t. The proof of this fact is quite complex and utilizes the weak convergence of population probability distributions over Λ^{r-1}(see [12]), as well as the fact that normal mixture is given by a maximum likelihood estimator. It employs a theorem about convergence of minimizers for a Γ-convergent function sequence (see for ex. [3]). Due to the size limits of this paper, we have to skip the details of the analysis.

The first component in the estimation (4.2) is an unavoidable error of the method and arises from a selected problem representation. It depends on degree of fit of a genetic algorithm to a given optimization task. The estimation of this component requires to answer two fundamental questions of genetic algorithm analysis and statistical data analysis: what information about the optimization problem is represented with fixed points of G and how good estimator for a brightness function is a normal mixture. The theoretical estimation of the first component in (4.2) is a very difficult problem and no effective method is known yet. The only available tool in this case is a simulative analysis.

The conclusion from the above is that if the first component in estimation (4.2) is small, a method of DR-CGS class with a SGA-based genetic engine will work well in the asymptotic sense.

5. Computational tests

DR-CGS has been successfully tested for several benchmark functions used typically in global optimization problems as well as for some simple examples of modeling of optimal geometry of chemical compounds (see for ex. [6, 7]). Here the strategy will be illustrated with two examples.

The first example concerns the maximization problem of the function

(5.1)

$$f(x,y)=\sin(x*y)+1 \quad (x,y)\in[-3,3]\times[-3,3]$$

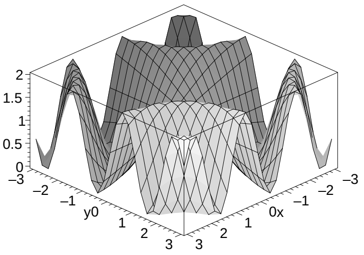

Figura 5.1. The graph of the objective (5.1)

The objective is shown in Fig. 5.1. The multiple maxima of the function constitute one-dimensional manifolds, which makes this example difficult to optimize. In general, DR-CGS is not recommended for such problems, but for this particular task the method is partially successful. A genetic engine is HGS (sect. 3.1). Due to the difficulty of the problem, the final concentration of genetic sample was not high enough in all interesting areas and clustering was performed for three clusters only. Fig. 5.2 presents the obtained mixture density function.

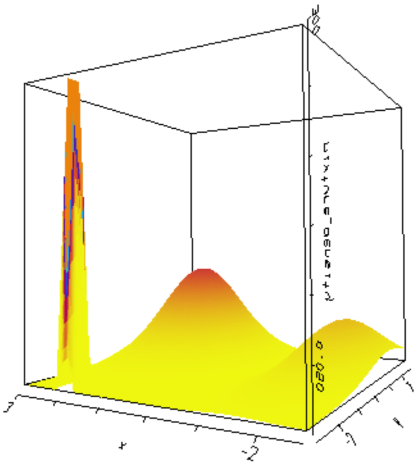

Figura 5.2. Mixture density function for objective (5.1)

The most of genetic sample is gathered under two higher components of the mixture and cluster extensions for these two peaks (see Fig. 5.3) are properly located inside the basins of attraction. The location of the third cluster extension is disturbed by a group of phenotypes concentrated in another basin (however, that group is too small to define a fourth cluster extension).

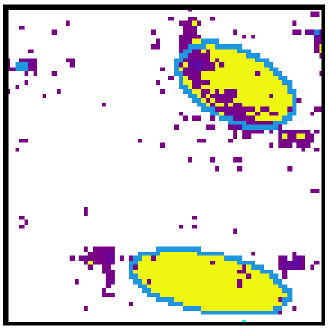

Figura 5.3. Cluster extensions for (5.1)

The second example is the minimization task of the ten-dimensional Griewank function:

$$(5.2)\quad g\left(x\right) = \frac{1}{4000}\sum_{i=1}^{10} x_i^2 - \prod_{i=1}^{10} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$$

$$x \in \left[-512, 512\right]^{10}$$

The Griewank function is a typical benchmark function used to test if a method stops or not in local minima. To give the idea of the function shape, one-dimensional graph is shown in Fig. 5.4

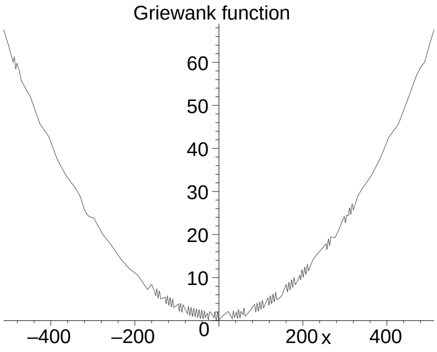

Figura 5.4. One-dimensional Griewank function

In this case the genetic engine was HGS-RN (with the floating point coding). As in many other examples computed with this engine, the genetic sample is strictly concentrated in a small neighborhood of minimum which ensures an excellent convergence of any gradient method, but limits the opportunity of landscape exploration. Fig. 5.5 shows a limit distribution $\rho\left(x_6, x_{10}\right)$ of a calculated ten-dimensional normal density function. Two next figures (5.6 and 5.7) cluster extensions for selected limit distributions. The domain of this graphs is given in the figure description. It gives a good impression about the sample concentration.

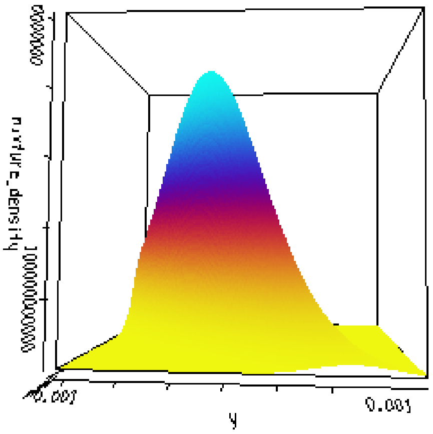

Figura 5.5. Limit distribution $\rho\left(x_6, x_{10}\right)$ of the normal density for Griewank function

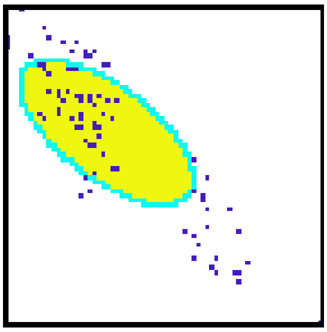

Figura 5.6. Cluster extension for the limit distribution $\rho\left(x_6, x_{10}\right)$, domain on the display $\left(x_6, x_{10}\right) \in [9{,}4407 * 10^{-4}, 9{,}4226 * 10^{-4}] \times [1{,}02 * 10^{-3}, 1{,}09 * 0^{-3}]$

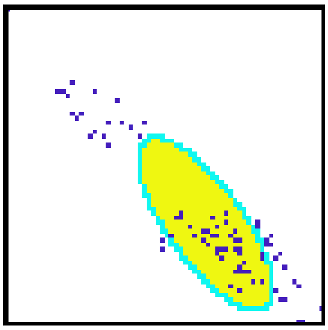

FIGURA 5.7. Cluster extension for the limit distribution $\rho(x_2, x_6)$, domain on the display $(x_2, x_6) \in [1{,}3416 * 10^{-4}, 1{,}3472 * 10^{-4}] \times [\in [9{,}4407 * 10^{-4}, 9{,}4226 * 10^{-4}]$

The plots of mixture density functions and cluster extensions have been prepared with the help of VisADJava component free library [11].

6. Conclusions

1. Clustered Genetic Search is a global optimization heuristic strategy, which consists in finding clusters in a genetic sample to get the approximations of basins of attraction of an objective. The approximations are given with cluster extensions which are defined on the base of discrete clusters. Cluster extensions are dense sets in an admissible set. CGS can be defined for many possible genetic engines and clustering tools.
2. Although CGS is a wide class of methods, there is a precise formal definition of the CGS strategy. The definition uses the idea of a genetic sample which is a multiset of phenotypes included (in the sense of def. 2.2) in an admissible set. A CGS method is expected to cluster a genetic sample according to the criterion of affiliation to separate basins of attraction.
3. Density Restoring CGS is class of CGS instances which characterizes with using a construction of Finite Mixture Model as a clustering tool. The choice of a genetic engine is limited only by a rule of well-tuning. In this paper HGS and HGS-RN have been used to present computational examples.
4. DR-CGS brings wide opportunities of asymptotic analysis while used with an SGA-based genetic engine. Due to the features of a normal mixture, it enables also a straightforward definition of cluster extensions which requires no additional computational effort. Presented computational tests illustrate how the method works and are a practical evidence of its good results.

Referencias

[1] Adamska K., Description of SGA population with countinuous H^1 brightness distribution, *Proc. of Symposium on Methods of Artificial Intelligence AIMETH'04*, Gliwice, Poland, November 17-19, 2004

[2] Dempster A.P., Laird N.M., Rubin D.B.; Maximum likelihood from incomplete data via EM algorithm, *J. Royal Statis. Soc., Ser. B*, vol. 39,1977, pp. 1-38

[3] Z. Denkowski, S. Migorski and N.S. Papageorgiou, *An Introduction to Nonlinear Analysis: Applications*, Kluwer Academic/Plenum Publishers, sect. 3.7 Boston, Dordrecht, London, New York, 2003.

[4] McLachlan G., Peel D., *Finite Mixture Models*, Wiley, 2000

[5] Schaefer R., Adamska K., Well-tuned genetic algorithm and its advantage in detecting basins of attraction, *Proceedings of 7th KAEiOG*, pp.149-154, Kazimierz Dolny, 2004

[6] Schaefer R., Adamska K., Jabłoński Z.J., Clustering driven by the genetic sampling measure, *Methods of Artificial Intelligence*, Proceedings of the Symposium on Methods of Artificial Intelligence AI-METH 2002, pp. 361-366, Gliwice, Poland, 2002

[7] Schaefer R., Adamska K. Telega H., Clustered genetic search in continuous landscape exploration, *Engineering Applications of Artificial Intelligence*, vol 17/4, pp.407-416, 2004

[8] Schaefer R., Jabłoński Z.J., On the convergence of sampling measures in the global genetic search. *Lecture Notes in Computer Science*, vol. 2328, pp. 593-600, Springer 2002.

[9] Schaefer R., Kołodziej J., Genetic search reinforced by the population hierarchy. in De Jong K. A., Poli R., Rowe J. E. eds. *Foundations of Genetic Algorithms 7*, Morgan Kaufman Publisher 2003, pp. 383-399.

[10] Telega H.; Adaptive parallel genetic clustering in parameter inverse problems, *Inverse Problems in Engineering Mechanics III,* M. Tanaka, G.S. Dulikravich (eds.), Elsevier, 2002, 315-324.

[11] VisAD Java library, http://www.ssec.wisc.edu/˜billh/visad.html

[12] Vose M. D.: The Simple Genetic Algorithm, MIT Press, 1999