HILL CRUNCHING CLUSTERED GENETIC SEARCH AND ITS IMPROVEMENTS

Henryk Telega[*], **Igor Podolak**[**]
telega@wsb-nlu.edu.pl, uipodola@theta.uoks.uj.edu.pl
[*] *Department of Computer Science*
Wyższa Szkoła Biznesu – National Louis University, Nowy Sącz
[**] *Institute of Computer Science, Jagiellonian University, Cracow*

ABSTRACT

Two modifications to the Hill Crunching Clustered Genetic Search (*HC-CGS*) algorithm are proposed in this paper. *HC-CGS* (see [Telega 1999], [Adamska et al. 2004] is a global optimization algorithm that was designed in order to solve such parameter inverse problems in which an approximation of certain level sets (central parts of basins of attractions of local minimizers) is required. The approximation of these sets can be useful when some additional criteria of optimization are considered after main results of parameter identification are obtained. The approximation is also helpful in stability analysis. In spite of some good properties of *HC-CGS*, tests have shown that its original version can be not effective for problems with more than 4 dimensions.

Two modifications of *HC-CGS* are proposed in order to overcome the dimensionality limitation. In the first one clusters are remembered as ellipsoids. The first modification is based on the idea of cluster recognition with the use of Kohonen Self Organizing Maps (*SOM*) neural networks [Kaski 1997]. In the second one clusters are remembered as ellipsoids.

INTRODUCTION

Many global optimization methods are aimed at finding a single local minimizer or the global minimizer. One can consider analogous maximization tasks. However, in many problems there is a need to look for all local minimizers or such minimizers which fulfill additional criteria. Parameter inverse problems are classical examples.

HC-CGS algorithm that is considered in this paper has been originally designed for parameter inverse problems that were formulated as global optimization tasks (see [Telega 2000, AST2004]. This algorithm is especially suitable for problems, in which an approximation of certain level sets (central parts of basins of attractions of local minimizers) is required. The approximation of these sets can be useful when additional criteria of optimization are considered after main results of parameter identification are obtained. Such criteria can express in some way for instance the availability and/or the

cost of materials. When one knows approximation of central parts of the basins, he or she can give an approximate answer to the question: how much one can change the value of a parameter with "not too high" change of the objective.

The *HC-CGS* strategy is inspired with clustering methods in global optimization [Rinnooy Kan, Timmer 1987a & b] and genetic algorithms [Vose 1999], [Goldberg 1989].

Generally, cluster analysis includes a specific group of pattern recognition methods (see [Jain et al.]). It consists in unsupervised exploration of a given data set, aimed at discovering groups in the data. Clustering results of partitioning of a discrete data set $X = \{x_1,...,x_m\}$ into nonempty mutually exclusive sets $X_1,...,X_k$, $k \le m$ called *clusters*. Clustering algorithms has been also proposed in continuous global optimization. The idea is to find groups of points from which local searches can be started. It is desirable that the number of local searches is diminished to one in each basin of attraction of a local minimum.

The role of genetic algorithms in clustering has increased for recent years (see [Adamska et al. 2004] and articles cited there). In most papers the term *genetic clustering* is used for such strategies in which genetic algorithm is used in order to determine clusters. In some papers this term is used also for strategies in which genetic algorithms are used to generate input data for clustering. In [Adamska et al. 2004] the latter strategies have been called Clustered Genetic Search and *HC-CGS* is a representative of this category.

To be more formal let us introduce some terms and give definitions. We focus on global minimization problems with continuous objective functions of the form $\Phi : D \to \Re$, $D \subset \Re^n$, $\forall x \in D \quad k \le \Phi(x) \le K$ which have only isolated minimizers in the interior of the domain and for which one can construct equivalent maximization problems: $\widetilde{\Phi} = M - m - \Phi$ (see also [Adamska et al. 2004]).

Let $L(y) = \{x \in D : \Phi(x) \le y\}$ and $\hat{L}(y) = \{x \in D : \Phi(x) < y\}$ denote two types of level sets of the function Φ. $L_x(y)$ and $\hat{L}_x(y)$ stand for simply connected components of $L(y)$ and $\hat{L}(y)$ (respectively) that contain x. For an arbitrary fixed point x^* being a minimizer of Φ let $\bar{y}(x^*)$ be defined as follows:

$$\bar{y}(x^*) = \begin{cases} \min\{y : \exists x^{**} \text{ isolated local minimizer of } \Phi,\ x^{**} \neq x^*, x^{**} \in L_{x^*}(y)\} & \text{if } x^{**} \text{ exists} \\ \min\limits_{x \in \partial D} \Phi(x) & \text{otherwise} \end{cases}$$

where ∂D stands for the boundary of the domain. The basin of attraction B_{x^*} of a local minimizer x^* is the simply connected component of the interior of $\hat{L}_{x^*}(\bar{y}(x^*))$ such that $x^* \in B_{x^*}$ (see also [Rinnooy Kan, Timmer 1987a & b], [Adamska et al. 2004]).

For global optimization problems that are being considered it is desired that a cluster contains such points from the domain that belong to the same basin of attraction. By a *cluster extension* we mean a closed positive measured set that contains one local minimizer x^* in its interior and is included in B_{x^*}.

HILL CRUNCHING CLUSTERED GENETIC SEARCH ( )

The aim of the original version of *HC-CGS* [Telega 1999] is to find all local minima that have adequately large basins of attraction with sufficiently large objective variability. The algorithm also gives information about basins. *HC-CGS* utilizes Simple Genetic Algorithm (*SGA* [Goldberg 1989]) as the genetic engine. This choice allowed to obtain some theoretical results concerning the stop criterion and asymptotic behavior of *HC-CGS*.
The idea of *HC-CGS* is presented in Scheme 1. Initially the set of cluster extensions is empty.

$$\overset{f}{\widetilde{\Phi}}$$

Scheme 1. The main scheme of *HC-CGS*

The domain of searches D is divided into hypercubes that constitute a grid. In this approach, the cluster extensions are unions of hypercubes. Each cluster extension is recognized in a stepwise manner. Each step of this process is performed after the *SGA* is stopped (that means after step 4). Step 5 can be explained as it is shown in Scheme 2.

Scheme 2. Recognizing of cluster extensions in *HC-CGS*.

The fitness modification results in repelling individuals from cluster extensions (or their parts) that are already found. The global stop criterion distinguishes two basic kinds of *SGA* behavior. The first one is that *SGA* finds parts of cluster extensions after few generations, and the second is that *SGA* "converges" to the uniform distribution of individuals. This corresponds to the recognition of plateau (or areas where the fitness has small variability) outside of the already known cluster extensions. Other cases are treated as the situation when the *SGA* does not fit to the particular problem, and a refinement of *SGA* parameters is suggested.

PROPERTIES OF

Each population with a finite number of binary coded individuals can be identified with a vector which *i*-th coordinate stands for the occurrence frequency of the *i*-th individual in the population. Lets by *r* denote the length of an individual. The frequency vector belongs to the unit simplex Λ in $\Re^{r-1}$. All possible populations of the size *n* correspond to a finite subset S_n in Λ [Vose 1999].

SGA with a finite population constitutes a stationary Markov chain with states from S_n. By non-zero mutation it is ergodic, and there exists a weak limit

$$\pi_n^k \underset{k\to\infty}{\overset{w}{\to}} \pi_n$$

of probability distributions π_n^k on S_n in *k*-th generation (see [Vose 1999]).

In the case of an infinite population $n=\infty$ *SGA* is a deterministic dynamic system with states in Λ, governed by the genetic search operator $G:\Lambda\to\Lambda$. The sequence of the limit probability distributions π_n has a weak limit distribution π^* when the size of population goes to infinity $n\to\infty$. Moreover if *G* is focused, and *K* stands for the set of fixed points of *G*, then $\pi^*(K)=1$ [Vose 1999].

Let F_ε be the ε-envelope of the set *K*

$$F_\varepsilon = \{x \in \Lambda; \exists y \in K; d(x,y) < \varepsilon\}$$

where *d* is a distance function in $\Re^{r-1}$.

Theorem 1 [Cabib et al. 1998]: $\forall\varepsilon>0\ \ \forall\varsigma>0\ \ \exists N>0$ such that $\forall n>N\ \ \exists W(n)$ and $\forall k>W(n)\ \ \pi_n^k(F_\varepsilon)>1-\varsigma$.

It means, that if the population is sufficiently large and a sufficiently large number of generations have been evaluated, then the population is arbitrary close to the fixed one with the arbitrary large probability.

Stop criteria can be justified with the use of the Vose theory of *SGA*. We try to detect the situation in which the population is sufficiently concentrated in basins so that density cluster recognition is possible. The state in which an arbitrary rate of raster cells contains the assumed number (much less than the average) of individuals can be handled as the local stop criterion. The above situation is asymptotically highly probable if there exists at least one basin of attraction out of the union of cluster extensions that are already recognized (see [Schaefer, Jab o ski 2002] and Theorem 3 in [Adamska et al. 2004]). The chart of the modified fitness function becomes sufficiently flat at the end of computations. This corresponds to the unique fixed point of G at the center of Λ (see [Vose 1999] Theorem 10.8). If a sufficiently large population that starts from the center of Λ (uniform distribution of individuals) does not leave its neighborhood sufficiently long, this implies that the center of Λ is the fixed point of G (with the arbitrarily large probability). This follows from the Theorem 1 and corresponds to the situation that the probability of finding new local minimizers is arbitrarily small.

One can say, that there is an analogy between the way in which mutation and crossover rates in *SGA* influences *HC-CGS* algorithm and the way in which the reduction phase influences *DC* and *SL* clustering algorithms described in [Rinnooy Kan, Timmer 1987a & b]. Both factors cause that some minima can be undetected. However, unlike the *DC* and *SL* with the reduction phase, the *HC-CGS* constitutes a filter that eliminates local minima with small fitness variability and shallow basins of attraction (see tests in [Telega 1999], [Telega 2002]). *HC-CGS* strategy is also less sensitive to fitness values in local minimizers. Such filtering property can be useful in some cases. Another interesting feature of *HC-CGS* is that it should be especially convenient for functions with large areas of small variability (areas similar to plateaus) which can be difficult for other methods.

Tests have shown that *HC-CGS* can be effective in solving some inverse problems for instance the problem of optimal pretraction design of a network structure made of elastic unconnected fibers fastened at their ends to a square rigid frame ([Telega 1999], [Telega 2002]).

However, the original version of *HC-CGS* is not effective for problems with more than 4 dimensions. This follows from the representation of clusters – they are remembered as unions of small hypercubes that constitute a regular grid in the domain of searches.

In order to overcome the above limitation two modifications to the *HC-CGS* algorithm are proposed in this paper. They will be described in following sections.

CLUSTERS RECOGNITION AND REPRESENTATION WITH THE USE OF NEURAL NETWORKS

There is a need to represent clusters or clusters' extensions found. We propose to use self-organizing maps (SOM's) ([Kohonen 1995], [Haykin 1999]) so that, when joined with the *HC-CGS* algorithm, high-dimensional input space clusters could be represented.

Self-organizing maps are organized of 2 layers:

- Input layer which represent vectors of features
- Output neurons organized in a form of a lattice, usually one- or two-dimensional, with a neighborhood relation defined (a *topographic map*)

Through the process of learning consecutive input vectors of features are passed on to neurons in the output layer which behave in a *competitive* way: only one (the one with the highest activation) is activated (a *winner-takes-all* strategy) and incoming synapses weights are modified. Together with the weight of the winning neuron weights of output neurons within the neighborhood are modified too. Thanks to that strategy *the spatial location of an output neuron in a topographic map corresponds to a particular domain or feature of data drawn from the input space* ([Kohonen, 1990]). Additionally, activations of neurons which are close by correspond to similar inputs.

The input data is represented in neurons by a vector w_i (reference vector), whose components correspond to synaptic weights. Neurons can be indexed with k. The winner neuron is determined from the formula:

$$k = k(x) = \arg\min_i \left\{ \|x - w_i\|^2 \right\}$$

That means the winner is this neuron, whose reference vector is closest to the input data x. This neuron and its neighbors modify their reference vectors according do the following formula.

$$w_i(t+1) = w_i(t) + h_{ki}(t)[x(t) - w_i(t)]$$

Neighbors are determined by so called neighborhood kernel function h_{ki}.

In the simplest case the neighborhood function can be defined as follows:

$$h_{ij} = \begin{cases} 1 & \|r_i - r_j\| \le \lambda \\ 0 & \|r_i - r_j\| > \lambda \end{cases}$$

or

$$h_{ij} = \exp\left(-\frac{\|r_i - r_j\|^2}{2\lambda^2} \right)$$

where r_i and r_j are vectors that represent location of neurons in the lattice, t denotes time and decreases (exponentially with time) during training. Thanks to that the training process constitutes of two phases

- *Ordering phase* during which a topological ordering of the neuron nodes are organized

- *Convergence phase* when output nodes activations are fine tuned

In such environment two possible choices for input vectors are possible:

- each function input constitutes the whole *SOM* input vector – in such case the *SOM* simply finds clusters of data, in this situation the whole input space is divided into Voronoi cells (a partition of the space according to the *nearest-neighbor* rule so that each cell contains those points from input space that are closest to the Voronoi vector, i.e. the reconstruction vector, among all of the points),
- or, preferably, the function data is augmented with information whether the point belongs to a minimum found with the *HC-CGS* algorithm or whether it lays outside of any minima known; in such case more patterns which are sums of Voronoi cells (in the original input space) corresponding to minima basins of attraction are found

After the training process all the output neurons need to be labeled. In our case they would be labeled with each minima identifier, so that afterwards, during actual use, the *SOM* would be able to classify an instance. If the second input data represenation is used, then the user would not know in advance if a point belongs to some minima or not (if he knew, then the whole process would be spurious) and would not be able to augment the input information as it was described above. Instead of that, that "minima" feature information would be zeroed equal to passing information "not known". The *SOM* network is able to distinguish and cluster the input data by other features.

If the original Kohonen *SOM* architecture is used, then the number of clusters does not need to be known in advance, it only has to be lower then the whole number of output neurons (preferably lower than the square root of that number). On the other hand, some extensions of the *SOM* algorithm are known in which the number of output neurons adapts, eg. the *growing neural gas* or *growing* grid algorithms [Fritzke 1995]. Both algorithms start from output layers organized into most simple lattices (either two neurons or a 2 by 2 neuron grid) which grow during training up to the moment that input points are categorized with sufficiently small error.

Another architecture possible to use would be to employ the *counter-propagation* network which is an extension of a SOM. In a counter-propagation network all of the *SOM* output neurons are connected to vector of linear neurons with dimensionalty equal to the original function dimensionality, which are trained with a supervised training algorithm after training of the *SOM* is finished. In that case a simple approximation of the original function is possible, in addition to input points classification as in the original SOM.

CLUSTERS REPRESENTATION WITH THE USE OF ELLIPSOIDS

Onother simpler method that can be proposed to overcome the problem with high dimensionality is to represent clusters by ellipsoids. This approach is the theme of the current research. The similar approach to clusters in global optimization is known in so called *Density Clustering* (*DC*) rule described in [Horst et al. 1995]. Some good

properties of this version of *DC* are proven in [Rinnooy Kan, Timmer 1987a]. The version of *DC* proposed by Rinnooy Kan and Timmer assumes that the reduction phase is applied, that means the initial sample is drawn from the uniform distribution over *D* and all sample points for which the value of the objective function is below certain threshold are rejected. A key assumption is that the objective function is well approximated by a quadratic function in a neighborhood of a local minimizer. This implies that central parts of basins are approximated by ellipsoids. Clusters' extensions are recognized iteratively in the following way: the seed point $\bar{x}$ of a cluster is the result of local optimization started from the unclustered best point of the reduced sample (the unclustered point with the smallest value of the objective function). Lets by T_0 denote the set $\{\bar{x}\}$ with the seed of the cluster. In consecutive steps next points of the sample are joined to the cluster. These points belong to subsets T_i of D, $i=1,2,\ldots$, $T_i \subset T_{i+1}, i = 1,2,\ldots$. These subsets correspond to certain level sets. When $f \in C^2$ we can approximate level sets by

$T_i = \left\{x \in D \middle| (x-\bar{x})^T H(x_s)(x-\bar{x}) \le r_i^2\right\}$, where H denotes hessian.

All points that are within T_i which is described by a critical distance $r_i(\bar{x})$ of the seed are joined to the cluster. The distance $d(x_1, x_2)$ is defined as follows: for points x_1, x_2 from a neighborhood of $\bar{x}$

$$d(x_1, x_2) = \left[(x_1 - x_2)^T H(\bar{x})(x_1 - x_2)\right]^{\frac{1}{2}},$$

(an approximation of hessian can be obtained as a byproduct of quasi-Newton local methods). The parameter $r_i(\bar{x})$ is increased stepwisely (with increasing i) until there is no unclustered point from the reduced sample within $r_i(\bar{x})$. Rinnooy Kan and Timmer gave the formula for the critical distance:

$$r_i(\bar{x}) = \pi^{-\frac{1}{2}}\left(i\Gamma\left(1+\tfrac{d}{2}\right)\det\left(H(\bar{x})\right)^{\frac{1}{2}} m(D)\frac{\sigma \log kN}{kN}\right)^{\frac{1}{d}}, \quad (*)$$

where Γ denotes here the Gamma function, m denotes the Lebesgue measure, N is the sample size and σ is a constant. The whole process of sampling, reduction and clusters recognition is repeated (k denotes the number of the iteration) until a global stop criterion is satisfied. The formula (*) assures that the probability of erroneous termination of the cluster recognition procedure (the procedure is terminated too early, see [Rinnooy Kan, Timmer 1987a] for details) in step i decreases polynomially fast with increasing k.

This version of *DC* has also other advantages:

- It has the property of asymptotic correctness in the sense that it finds global minimum with the probability 1 as k increases to infinity.
- It is possible (and relatively easy) to apply bayesian stopping rules [Horst et al. 1995], [Rinnooy Ka, Timmer 1987a].

The main drawbacks are obvious:

- The success of the method depends on how well the assumed approximation is.

- In fact each recognized cluster can contain more than one minimizer.

Applying similar approach to *HC-CGS* can diminish disadvantages that are caused by high dimensionality. Clusters are parametrized by the central point and radiuses. Each point generated by *SGA* can be classified as belonging to some cluster or not, so the idea of fitness modification can be almost unchanged.

However, the Bayesian stopping rules derived for *DC* cannot be applied directly to *HC-CGS*, because these rules assume uniform distribution of sample points. Also such good properties of *DC* as mentioned above cannot be directly attributed to *HC-CGS*. Analogous estimations for *HC-CGS* are still open problems, because it is difficult to predict and calculate the exact distribution of points after some genetic epochs.

Introducing such cluster representation to *HC-CGS* causes also that stopping strategy from *HC-CGS* should be modified. Under the assumption that clusters cannot intersect, the criterion "the whole domain has been processed" should be removed.

REFERENCES

[Adamska et al. 2004] R. Schaefer, K. Adamska, H. Telega,: *Clustered genetic search in continuous landscape exploration*. Engeneering Applications of Artificial Intelligence no 17, Elsevier 2004, pp. 407-416.

[Cabib et al. 1998] E. Cabib, R. Schaefer, H. Telega, *A parallel genetic clustering for inverse problems*. Lecture Notes in Computer Science 1541, Springer 1998, pp. 551-556.

[Fritzke 1995] B. Fritzke, *A growing neural gas network learns topologies*, in Advances in Neural Information Processing Systems 7, eds. G. Tesauro, D. Touretzky, T. Leen, MIT Press, Cambridge MA, 1995

[Fritzke 1995] B. Fritzke, *Growing grid – a self organizing network with constant neghborhood range and adaptation strength*, Neural Processing Letters, 2(5), 1995, pp. 9-13.

[Goldberg 1989] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA 1989.

[Jain et al. 1999] A.K. Jain, M.N. Murty, P.J. Flynn, *Data clustering - a review*, ACM Computing Surveys 31 (3), 1999, pp. 264–323.

[Haykin 1999] S. Haykin, Neural Networks, a comprehensive foundation, 2nd ed., Prentice Hall, Upper Saddle NJ, 1999.

[Horst et al. 1995] R. Horst, Pardalos M. Panos, Handbook of Global Optimization, Kluwer Ac. Publisher 1995.

[Kaski 1997] S. Kaski, *Data Exploration Using Self-Organizing Maps*, Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series No. 82, 1997.

[Kohonen 1995] T. Kohonen, Self Organizing Maps, Springer, Berlin, 1995.

[Rinnooy Kan, Timmer 1987a] A.H.G. Rinnooy Kan, G. T. Timmer, *Stochastic global optimization methods. Part I: Clustering methods*, Mathematical Programming 39, 1987, pp. 27-56.

[Rinnooy Kan, Timmer 1987b] A. H. G. Rinnoy Kan, G. T. Timmer, *Stochastic global optimization. Part 2: Multi level methods*. Mathematical Programming 39, 1987, p. 57-78.

[Schaefer, Jab o ski 2002] R. Schaefer , Z.J. Jab o ski, *On the convergence of sampling measures in the global genetic search*. Lecture Notes in Computer Science, Vol. 2328. Springer, Berlin, 2002, pp. 593–600.

[Telega 2002] H. Telega, *Adaptive Parallel Genetic Clustering in Parameter Inverse Problems*, Inverse Problems in Engineering Mechanics III, M. Tanaka, G. S. Dulikravich Eds., Elsevier 2002, pp. 315-325.

[Telega 1999] H. Telega, Parallel algorithms for solving selected inverse problems, PhD Thesis, AGH Krakow, Poland (in polish), 1999.

[Vose 1999] M. D. Vose, The Simple Genetic Algorithm - Foundation and Theory. The MIT Press, 1999.