EVOLUTIONARY PROTOTYPE SELECTION

by
D. Dumitrescu, Károly Simon

Abstract: Evolutionary algorithms are useful tools for clustering and learning purposes. A new evolutionary multi-modal optimization technique is used to derive a new clustering algorithm. The optimal number of clusters is an outcome of the method.
Keywords and phrases: dynamic evolutionary clustering, Genetic Chromodynamic, GC-based clustering

1 Introduction

Recently a new evolutionary search and multi-modal optimization metaheuristics - called Genetic Chromodynamics (GC) [3] - was proposed. This metaheuristics could be used to derive dynamic evolutionary clustering algorithms.

By clustering a data set is divided into regions of high similarity, as defined by a distance metric. In most instances, a prototypical vector (the cluster center) identifies a cluster. Hence, the problem of cluster optimization is twofold: optimization of cluster centers and determination of number of clusters. The latter aspect has often been neglected in standard approaches (static clustering methods) (see [8, 9]), as these typically fix the number of clusters *a priori*.

In case of practical problems the number of existing clusters is generally unknown. Opposed to static, dynamic clustering does not require *a priori* specification of the number of clusters. Some tentative to develop dynamic evolutionary clustering algorithms are known (see [4 – 6, 10]).

Based on the GC theory a new evolutionary clustering method has been developed and used to solve practical problems (see [5, 6, 10]). An improved version of this GC-based dynamic clustering method (GCDC) is presented and some experiments are described.

In Section 2 GC related theoretical notions are presented. The GCDC method is described in section 3, in section 4 numerical experiments are presented.

2 Genetic Chromodynamics

GC is a new kind of evolutionary search and optimization metaheuristics [3]. GC is a metaheuristics for maintaining population diversity and for detecting multiple optima. The main idea of the strategy is to force the formation and maintenance of stable sub-populations.

GC-based methods use a variable-sized population, a stepping-stone search mechanism, a local interaction principle and a new operator for merging very close individuals.

Corresponding to the stepping-stone technique each individual in the population has the possibility to contribute to the next generation and thus to the search progress. Corresponding to the local interaction principle the recombination mate of a given individual is selected within a determined mating region. Only short range interactions between solutions are allowed. Local mate selection is done according to the values of the fitness function. An adaptation mechanism can be used to control the interaction range, so as to support sub-population stabilization. Within this adaptation mechanism the interaction radius of each individual could be different.

To enhance GC, micropopulation models [7] can be used. Corresponding to these models, for each individual a local interaction domain is considered. Individuals within this domain represent a micropopulation. All solutions from a micropopulation are recombined using local tournament selection. When the local domain of an individual is empty the individual is mutated.

Within GC sub-populations co-evolve and eventually converge towards several optima. The number of individuals in the current population usually changes with the generation. A merging operator is used for merging very close individuals. At convergence, the number of sub-populations equals the number of optima. Each final sub-population hopefully contains a single individual representing an optima, a solution of the problem.

GC allows any data structure suitable for the problem together with any set of meaningful variation/search operators. For instance solutions may be represented as real-component vectors. Moreover the proposed approach is independent of the solution representation.

3 GC-based Dynamic Clustering

GC can be used to derive evolutionary clustering algorithms. In the following a GC-based dynamic clustering algorithm – called GCDC - is described.

3.1 Solution Representation

Let

$$X = \{x_1,...,x_m\}, x_i \in \mathbf{R}^s, s \geq 1,$$

be the data set for clustering. The cluster structure of X is given by a fuzzy partition $P = \{A_1,...,A_n\}$ of X. Every class A_i is represented by a prototype $L_i \in \mathbf{R}^s$. $L = \{L_1,...,L_n\}$ is the representation of the partition P.

In the proposed clustering technique each prototype is encoded into a chromosome. Totality of these chromosomes represents a generation.

The idea of the method is to determine formations of evolving chromosomes converging towards prototypes of real clusters.

The initial population is randomly generated and it contains a large number of individuals. The operations involved in the searching process are selection, crossover, mutation and merging.

3.2 Fitness Function

The fitness value of chromosome L can be calculated using the following fitness function:

$$f(L) = \sum_{i=1}^{m} \frac{1}{d^{\alpha}(x_i, L) + C}, \qquad (1)$$

where $\alpha \geq 1$ and $C > 0$.

The role of the constant C is to prevent infinite or too great values for the fitness function and together with α controls the granularity of the clusters.

Generally the above-mentioned fitness function is used in standard evolutionary clustering algorithms. In order to improve GCDC a new fitness function is proposed. The fitness value of chromosome L is calculated using the following Gaussian function:

$$f(L) = \sum_{i=1}^{m} e^{-\frac{\|x_i - L\|^2}{\sigma^2}} \qquad (2).$$

The Gaussian fitness function is centered to chromosome L and its variance is σ. The variance can be considered as the standard deviation of all points. Some adaptation mechanism also could be used.

In the next section numerical experiments are described. A comparison between these fitness functions is presented.

3.3 Interaction Range

For each individual in the population (a chromosome representing a prototype) a mating region is considered as the closed ball with radius d^*, where the *interaction radius* d^* depends on the chromosome.

Initially we consider the neighborhood distance for each chromosome as the standard deviation of all points. For a chromosome L the mean distance $\bar{\delta}$ is calculated between the points in $V(L, d^*)$ and L:

$$\bar{\delta} = \sum_{i=1}^{n_{d^*}} \frac{d(x_i, L)}{n_{d^*}},$$

where $x_1, ..., x_{n_{d^*}}$ are the points in the neighborhood with radius d^* of L.

When the points in $V(L, d^*)$ are uniformly distributed, the value of $\bar{\delta}$ is $\frac{d^*}{\beta}$, where $\beta \in (1,2]$ is a fixed number, which depends on the dimension s of the search space (generally the best value for β is $\sqrt[s]{2}$). d^* is adjusted such that $\bar{\delta}$ to be equal

with $\frac{d^*}{\beta}$, so if $\bar{\delta} \le \frac{d^*}{\beta}$ then the next value for d^* will be $\beta\bar{\delta}$, else $\bar{\delta}$. If there are not at least two points in the neighborhood of the chromosome, then the previous distance value will be not modified.

3.4 Selection, Crossover, Mutation and Merging

A micro population model is used. At each step of the generation process every chromosome is selected to produce an offspring through crossover or mutation. An individual can be involved into a crossover operation only with individuals that are at smaller distance than d^*.

The mate for the crossover operation for an individual is selected among the chromosomes in its neighborhood with a proportional selection. Later the mate will be selected as first parent to produce its offspring. For this reason at crossover only one new chromosome is generated. If there is no mate for the crossover operation in the neighborhood of radius d^* of an individual, then the mutation operator will be applied.

At each generation every chromosome is involved in crossover or mutation. An offspring can replace only its parent. When an offspring is produced, it is compared with the parent and the best (with better fitness) is introduced in the new generation.

An effect of the crossover operation is that the chromosomes in the same subpopulation are overlapping after a number of iterations. When the distance between two chromosomes is smaller than a considered value ε (*merging radius*) they are merged. In this way the size of the population decreases during the process until *n* individuals remain, where *n* is the optimal cluster number.

3.5 Termination and Fuzzy Class Detection

If no more changes occur in the population through a fixed number of iterations then the process will stop. The individuals constituting the last population are considered as prototypes of the detected clusters. For all points the fuzzy membership to the clusters determined by the prototypes is calculated.

4 Numerical Experiments

From the two-dimensional input space data points ((*x*,*y*) pairs, where $x \in \{100,...,300\}$ and $y \in \{100,...,300\}$ are considered. Data sets for clustering are constructed using these points. Several tests for different input data sets are performed.

GCDC is used for data clustering. The crossover operation is a convex combination of the codes of the genes. The coefficient of the combination is a randomly generated number for each gene. Mutation is an additive perturbation of the genes with a randomly chosen value from a $N(0,\sigma')$ normal distribution, where σ' is a control parameter called *mutation step size*. The parameters of the method are:

- parameters for the fitness function (1): α=2, *C*=140;

- mutation step size: $\sigma' = 15$;
- merging radius: $\varepsilon = 20$.

The variance for the Gaussian fitness function (2) is considered as the standard deviation of all points.

4.1 Experiment 1

19 data points organized in 5 clusters are considered. GCDC is used for clustering this data set. The initial population contains 38 individuals.

Experiment 1.1

The fitness function (1) is used. After 56 iterations the correct number of clusters is determined by GCDC. The algorithm detects existing clusters and corresponding centers. Obtained results are depicted in Figure 1.

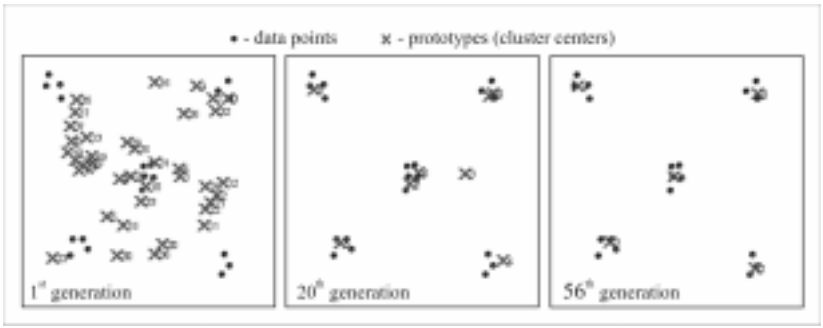
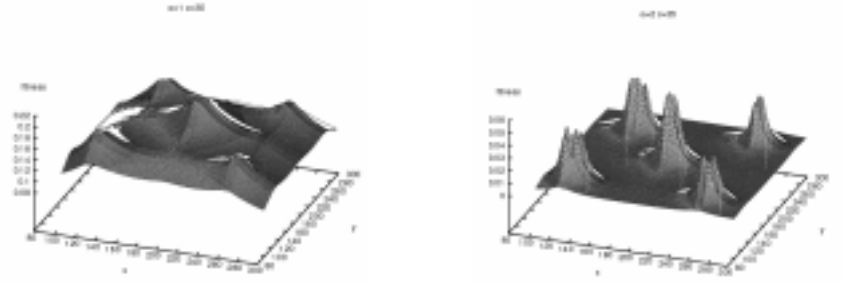


Fig. 1. Convergence of the GCDC algorithm: two-dimensional input data, 19 data points organized in 5 clusters

More tests with different parameters for the fitness function are performed. The behavior of the fitness function is presented in Figure 2.



Experiment 1.2

The Gaussian fitness function (2) is used. After 38 iterations the correct number of clusters is determined. The algorithm detects existing clusters and corresponding centers. Obtained results are depicted in Figure 3. The surface determined by the fitness function is presented in Figure 4.

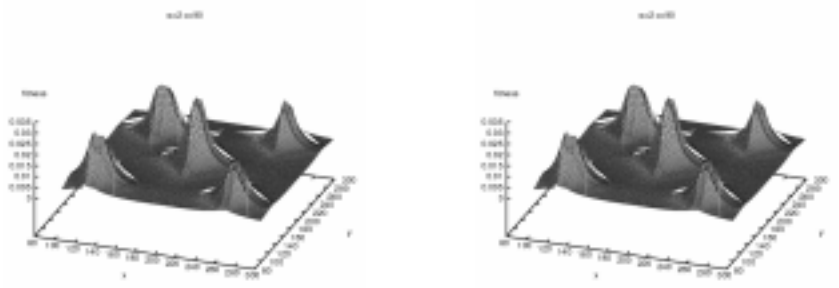


Fig. 2. Fitness functions for different parameter values.

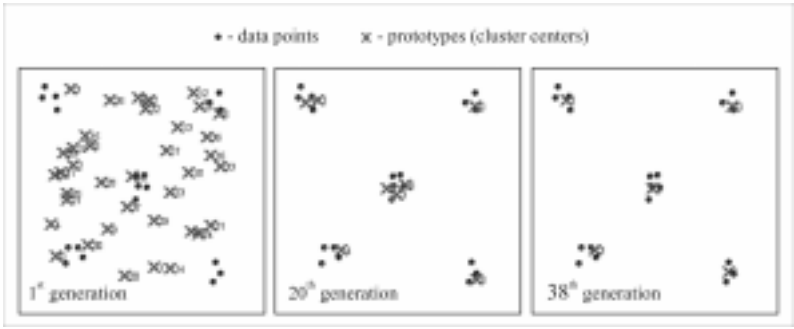


Fig. 3. Convergence of GCDC using Gaussian fitness function: 19 data points organized in 5 clusters

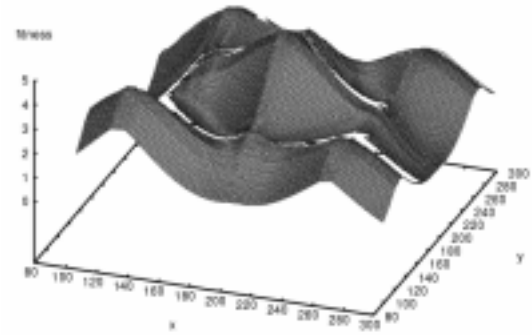


Fig. 4. The surface determined by the Gaussian fitness function.

4.2 Experiment 2

46 data points organized in 4 clusters are considered. There are essential differences between the diameters of clusters. GCDC is used for clustering this data set. The number of individuals in the first population is 92.

Experiment 2.1

The fitness function (1) is used. After 60 iterations the algorithm detects 5 clusters and corresponding centers. For the largest cluster two centers are considered. Obtained results are depicted in Figure 5.

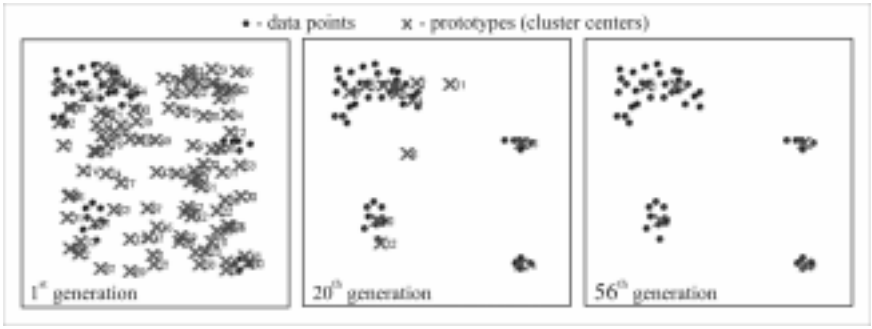


Fig. 5. Convergence of GCDC: 46 data points organized in 4 clusters (5 centers are detected)

Experiment 2.2

The Gaussian fitness function (2) is used. After 45 iterations the algorithm detects 4 clusters and corresponding centers. The method is able to determinate the correct number of clusters. Obtained results are depicted in Figure 6.

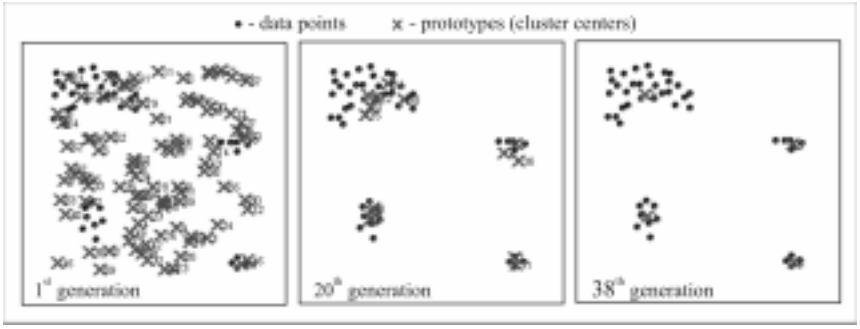


Fig. 6. Convergence of GCDC using Gaussian fitness function: 46 data points organized in 4 clusters

4.3 Obtained Results and Conclusions

When the fitness function (1) is used the results obtained by GCDC are influenced by the parameters of this function. These parameters must be fixed manually or a specific parameter setting technique has to be considered. Such a technique is not easy to be obtained. Using the Gaussian fitness function these parameters are not required. Only the variance of the fitness function must be fixed. The standard deviation of all points can be used for determining the value of the

variance. Better results are obtained using the Gaussian fitness function.

5 Conclusions

GC can be successfully used to derive new dynamic evolutionary clustering algorithms. These techniques are able to determinate the optimal number of clusters.

GCDC is a dynamic clustering algorithm based on the GC metaheuristics. Experiments show that this new evolutionary technique can be successfully used for dynamic data clustering. The method is able to determinate the optimal number of clusters and detects existing centers and corresponding classes into the input data set. It could be a powerful instrument for solving clustering problems.

References

[1] Dumitrescu D., Algoritmi Genetici şi Strategii Evolutive - Aplicaţii în Inteligenţa Artificialã şi în Domenii Conexe, Editura Albastră, Cluj Napoca, 2000.
[2] Dumitrescu D., Lazzerini B., Jain L. C., Dumitrescu A., Evolutionary Computation, CRC Press, Boca Raton, 2000.
[3] Dumitrescu D., Genetic Chromodynamics, Studia Univ. Babeş-Bolyai, Ser. Informatica, 35 (2000), pp. 39-50.
[4] Dumitrescu D., Bodrogi L., A New Evolutionary Method and its Applications in Clustering, Babeş-Bolyai University, Seminar on Computer Science, 2 (1998), pp. 127-134.
[5] Dumitrescu D., Simon K., Reducing Complexity of RBF Neural Networks by Dynamic Evolutionary Clustering Techniques, Proceedings of CAIM,1 (2003), pp. 83-89.
[6] Dumitrescu D., Simon K., Genetic Chromodynamics for Designing RBF Neural Networks, Proceedings of SYNASC, (2003), pp. 91-101.
[7] Oltean M., Groşan C., Genetic Chromodynamics Evolving Micropopulations, Studia Univ. Babeş-Bolyai, Ser. Informatica, (2000).
[8] Schreiber T., A Voronoi Diagram Based Adaptive *k*-means Type Clustering Algorithm for Multidimensional Weighted Data, Universitat Kaiserslautern, Technical Report, (1989).
[9] Selim S. Z., Ismail M. A., *K*-means Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality, IEEE Tran. Pattern Anal. Mach. Intelligence, PAMI-6, 1 (1986), pp. 81-87.
[10] Simon K., Evolutionary Clustering for Designing RBF Neural Networks, Babeş-Bolyai University, MSc. Thesis, 2003.

Authors:

D. Dumitrescu, ddumitr@cs.ubbcluj.ro, Károly Simon,
ksimon@cs.ubbcluj.ro, Computer Science Department, "Babes-Bolyai" University, Cluj-Napoca, Romania.