Proceedings of the 5th WSEAS International Conference on Telecommunications and Informatics, Istanbul, Turkey, May 27-29, 2006 (pp92-97)

The Genetic Chromodynamics Metaheuristic

D. DUMITRESCU[1] and CATALIN STOEAN[2]
[1]Department of Computer Science
Babes-Bolyai University of Cluj-Napoca
Str Mihail Kogalniceanu, No 1B, 400084, Cluj-Napoca
ROMANIA
http://www.cs.ubbcluj.ro/~ddumitr/
[2]Department of Computer Science
University of Craiova
Str A. I. Cuza, No 13, 200585, Craiova
ROMANIA
http://inf.ucv.ro/~cstoean/

Abstract: - An evolutionary metaheuristic called *genetic chromodynamics* and its applications to optimization, clustering and classification are presented in current paper. Genetic chromodynamics aims at maintaining population diversity and detecting multiple optima. All algorithms derived from genetic chromodynamics use a variable-sized population of solutions and a local interaction principle as selection for reproduction. Sub-population formation is achieved through the interaction between individuals, without any modification of the objective function. Sub-populations evolve and eventually converge to several optima. Very close individuals are merged and thus population size may be decreased with each generation. At convergence, each final subpopulation contains a single individual which corresponds to one optimum (solution of the problem). The model can be successfully applied to various optimization issues in telecommunication.

Key-Words: - Genetic Chromodynamics, multimodal optimization, evolutionary computation, function optimization, clustering, classification.

1 Introduction

Most of the real world problems we are trying to solve have more than one solution; meanwhile, the majority of evolutionary techniques [4] developed so far are dedicated to solve problems with a single solution only, therefore, the need for powerful techniques able to detect all solutions or only a selection of the best of them is very high. In addition, the hitch of the models intended to solve unimodal problems is that search may block into a suboptimal solution; when using a multimodal technique, chances to find the global optimum are a lot higher.

Genetic Chromodynamics (GC) [3], [6] represents a metaheuristic for solving (multimodal) optimization/search problems. It embodies an implicit combination of a global evolutionary searcher with a local one and mixes the advantages of both. In order to detect the correct number of optima, it is necessary to have one solution per optimum. For this purpose a new operator called *merging* is introduced. It merges very similar individuals, keeping only the most promising ones in the population. Results obtained by using GC for the optimization of numerous tricky functions and through its application to clustering and classification confirm the viability of the model.

A field that requests application of multimodal optimization engines is telecommunications. With the increased number of services and users, the discovery of all (sub) optimal potential solutions is a major advantage, as providers can subsequently select the one which is advantageous to implement. The optimal design of error correcting codes for a fast and reliable transmission of messages through noisy channels, the optimal routing of packets through a network or the optimal coverage of a maximal area with a minimum set of transmitters are just a few common optimization problems in telecommunications. On the other hand, the optimal filtering (classification) of unwanted e-mails or the identification of optimal clusters of nodes for the design of logical rings in advanced optical fiber systems are also of major importance. Through its behaviour and validated by its applications in function optimization, clustering and classification encourage the belief of the ability and suitability of GC for telecommunications issues.

The paper is organized as follows: next section presents the core of the GC metaheuristic, while section 3 includes some variants of GC meant to improve and generalize proposed model. Section 4 puts together some applications of GC for several benchmark problems. Last section ends the discussion by formulating some conclusions.

2 Standard GC model

Unlike most evolutionary techniques, GC uses a variable sized population, as the number of individuals decreases with the advance in evolution. This is a very important point as, when diversity decreases, there is no need for many similar individuals in the population anymore; their recombination will not direct search towards unexplored areas from the search space. Moreover, the decrease in the number of individuals will significantly minimize the number of fitness evaluations, leading therefore to a smaller amount of computations.

GC forces the formation and the maintenance of stable subpopulations from the early search stages of the process and each subpopulation is connected to local or global optima of the problem to be solved. This is achieved by introducing a set of restrictions such as the way selection is applied or the way recombination takes place. For selection, each individual represents a stepping-stone for the establishment of the new generation, i.e. each chromosome is taken into account for reproduction purposes. Additionally, a local interaction principle is considered in this respect, meaning that only chromosomes similar under a given threshold recombine. If a chromosome *c* has no similar individuals to it, i.e. there is not any other individual in the current population similar to *c* under the considered threshold, then *c* suffers mutation. After recombination or mutation takes place, the offspring fights for survival with the stepping-stone parent.

At a first glance, one can easily notice that an uncommon selection procedure takes place within the GC model, as each individual from the current population participates in forming the population that will form the next generation. This is the reason why each individual is called a *stepping stone* at certain moments.

A description of the algorithm is illustrated in Fig. 1. The algorithm begins with the initialization of the population; at this point, each individual is considered to have a different color. Therefore, the search process starts with a population whose all individuals have different colors. Afterwards, the color of each individual changes into the color of the fittest individual within its mating range. By applying merging, many individuals are removed from the current population; therefore from the early stages each sub-population will contain only individuals colored likewise.

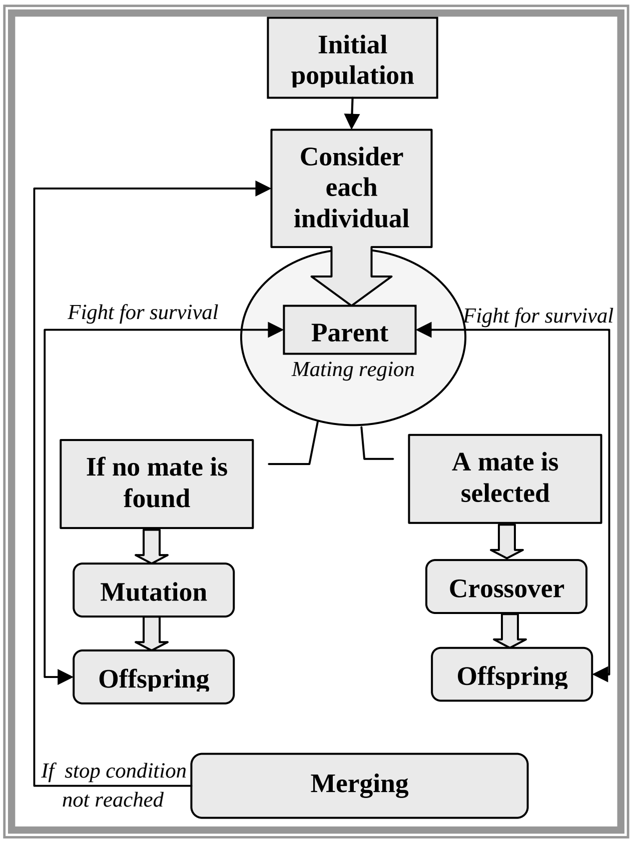

Fig. 1 General scheme of GC algorithm

For each individual, the dissimilarity between it and all the other individuals in the population is computed; all individuals which are at a lower distance than the *mating radius r* form the mating region of the current individual. Any distance to characterize the dissimilarity between individuals can be chosen, depending on representation.

If there is not any individual in the mating area of the current individual *c* (left branch of Fig. 1) mutation is applied to it and obtained offspring fights for survival with the parent. This situation happens when the individual remains alone in a search sub-space corresponding to an optimum and only some fine tuning as a result of mutation is henceforth performed to it. Mutation step size is chosen as for the offspring to remain in the mating area of the parent.

If there are some individuals in the mating area of the current individual *c* (right branch in Fig. 1), one of them is selected (using any selection scheme) and crossover takes place one offspring resulting. The offspring fights for survival with the stepping stone parent, *c*. Fig. 2 illustrates the way mutation, crossover and merging happen.

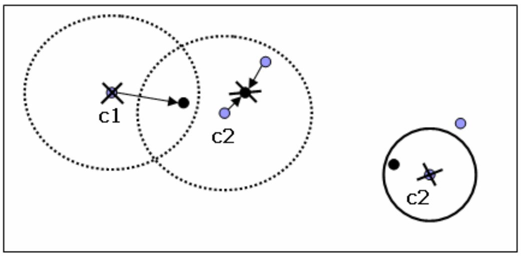

Fig. 2 Mating (left) and merging (right) within GC. Chromosome c_1 produces one offspring by mutation; chromosome c_2 selects another chromosome from its mating region (dotted circle) and produces one offspring by crossover. Crossed lines indicate replaced chromosomes with worse fitness. During merging, c_2 is deleted because there is another chromosome with better fitness in its merging region (solid circle).

GC introduces a new operator that *merges* very similar individuals into a single one that is often chosen to be the best one of them with respect to the fitness evaluation (Algorithm 1).

As a consequence of all these phenomena, subpopulations independently evolve and become better separated with each iteration and lead, at convergence, each one to an optimum.

```
Repeat
        A chromosome c is considered to be the current one;
        Select all m individuals in the merging region of c, including itself;
        Remove all but the best chromosome from the selection;
Until merging cannot be applied at all
```

Algorithm 1 Merging procedure within GC

Usually, the value of the *merging radius* is considered to be lower than the mating radius in order to permit crossover to take place multiple times between individuals during the evolution process; at the same time, one would not choose a high value for the merging radius as population size would be drastically decreased and this would not allow enough time for the exploration of the search space to take place.

Summing up, there are some principles that, as a whole, make GC unique in comparison to any other evolutionary algorithm:

1. *Population size is variable.*

2. *Sub-population structure is not predefined, but emergent.*

Sub-populations are formed and maintained due to the stepping stone principle connected to a local interaction scheme. GC starts with a large population of arbitrary solutions and population size may change at each generation, due to merging. Consequently there is not a predefined structure for sub-populations, but they are naturally formed and maintained.

3. *Each individual within the current population is considered a stepping-stone for the evolution process.*

Each individual is involved in the search process. This scheme has a very important role in maintaining diversity in the population as particularities of each individual are evolved at each step.

4. *A new operator is used for fusing very close individuals.*

A merging operator is used for reducing population size towards the final goal of containing only the optima.

5. *Apply merging and variation operators are applied until reaching multimodal convergence.*

At the end of the algorithm, each individual within the final population corresponds to an optimum.

```
Initialize population;
While termination condition is not satisfied
    Evaluate each chromosome;
    For all chromosomes c in the population do
        If mating region of c is empty then
            Apply mutation to c;
            If obtained chromosome is fitter than c then replace c;
            End if
        Else
            Select one chromosome from the mating region of c for crossover;
            Obtain and evaluate one offspring;
            If offspring is fitter than c then replace c;
            End if
        End if
    End for
    Merging
End while
```

Algorithm 2 GC Algorithm

A description of the pseudocode of GC algorithm is pointed out in Algorithm 2. The stop condition of the algorithm may refer either to a pre-specified number of steps (generations) for which the algorithm will run or to a fixed number of generations without any improvement for which the algorithm should run or to a specified accuracy of the optimum that the algorithm must reach etc.

3 Other Algorithms in GC Framework

This section overviews some algorithms that were developed with GC as a starting point. The reasons for building these models were various, among which the better exploration of the search space, the maintenance of diversity in the population or the speed up of convergence time were the most important targets.

3.1 A Crowding Procedure within GC ([8])

The motivation for developing this new algorithm within the GC framework involves the preservation or even the improvement the ability of the standard technique to properly locate several or all optima within one go, and in addition speed up this process.

In comparison to the standard approach, within present algorithm, an offspring can enter the current population and immediately take part to the evolution process; thus, the algorithm gains a lot in dynamism. In contrast to the stepping stone mechanism, the first parent is selected randomly. The offspring obtained after crossover does not replace any of the parents particularly, but the worst individual (with respect to fitness values) within its *replacement radius*, a new parameter of the algorithm. Thus, weak individuals are removed more aggressively (together with the effect merging has in this respect) from the current population. The local interaction principle and merging still hold.

In present algorithm, diversity is not maintained by the fact that each individual represents a stepping stone, but because of the way replacement takes place, which happens as in crowding ([7]).

The stepping stone principle does not apply here anymore, but *n* (where *n* is the number of chromosomes in the population) random individuals are selected instead. The reason for randomness is that individuals may be replaced by some offspring without ever being selected for recombination. An important aspect of the algorithm is the choice of the replacement radius value. If picked properly, this parameter may lead to improved convergence speed.

Radii-based evolution in the context of the new algorithm is depicted in Figure 3.

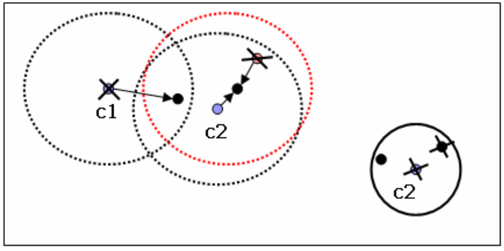

Fig. 3 Mating (left) and merging (right) within the new algorithm. As in Figure 2, c_1 and c_2 each produce one offspring. This time, the second offspring replaces its other parent – one of the chromosomes in its replacement area (lighter dotted circle) - because the latter is the worst chromosome within its replacement region. During merging, two chromosomes are removed, c_2 and one offspring, because now three chromosomes are within merging radius from c_2.

The algorithm was applied for function optimization, as well as for two classification problems; obtained results were slightly better than those of the classical algorithm in the GC framework [9].

3.2 Cloning within GC [9]

A cloning procedure is introduced within the algorithm presented in previous subsection in order to force a better exploration of the search space near the optima; this is performed by introducing more copies of an individual connected to a current optimum and applying mutation to them. Individuals resulting after mutation that are better than the initial one are introduced in the next generation. Through retaining only better individuals, exploitation of search space is also done.

Cloning draws its roots from the theory of artificial immune systems ([2]). Cloning (Algorithm 3) appears only if there is no individual left in the mating region of the current one. Thus, new individuals can appear in the population of the next generation and will be part of the same mating region; as a consequence, crossover will take place once more for the current individual.

```
Begin
        An individual is considered to be the current one;
        A fixed number of clones (copies) of the current individual are made;
        Mutation is applied to all copies;
        Mutated copies are evaluated and all individuals that have a higher value for
        the fitness function than the one of the current individual are introduced in the
        population of the next generation;
End
```

Algorithm 3 Cloning procedure

In GC algorithm, when an individual, *c*, remains alone in a region, i.e. there is no other individual in its mating area, mutation will be applied to it until the algorithm finally stops; this leads to a very high number of mutations that are applied to *c*. By introducing new individuals that are in the same mating region with the current one, crossover will take place between them and a better exploration takes place. There is a fixed number of clones that may be introduced at each step (usually, between 1 and 4). The cloning procedure introduces thus new individuals in the population, so the population size varies but, ultimately, the population will contain the pursued optima with significantly higher accuracies.

The algorithm was applied for the optimization of three multimodal, two-dimensional functions ([9]). Results emphasize the fact that obtained accuracies are considerably better but at the expense of a higher computational time.

4 Applications of GC

GC and variants of the original algorithm were successfully applied for the optimization of multimodal and highly multimodal functions, for clustering or for classification. Next subsections sketch out some of the interesting results obtained.

4.1 Function Optimization

Different types of functions were chosen for optimization using GC or variants of the standard algorithm. Some of them are outlined below:

- *Schaffer function* on the [-20, 20]^2 domain:

$$f(x,y) = -0.5 - \frac{\sin^2(\sqrt{x^2+y^2}) - 0.5}{(1+0.001(x^2+y^2))^2}$$

- *Himmelblau function* on [-4, 4] × [-6, 6]:

$$f(x,y) = 200 - (\mathrm{x}^2 + \mathrm{y} - 11)^2 \quad (\mathrm{x} + \mathrm{y}^2 - 7)^2$$

- *Six-Hump Camel Back function* on [-2, 2]^2

$$f(x,y) = -[(4 - 2.1\mathrm{x}^2 + \frac{\mathrm{x}^4}{3})\mathrm{x}^2 + \mathrm{xy} + 4(\mathrm{y}^2 - 1)\mathrm{y}^2]$$

- *Schwefel function* on [-500, 500]$^{\mathrm{n}}$

$$\sum_{i=1}^{n} x_i \sin(\sqrt{|x_i|})$$

Schaffer and Schwefel functions have one global optimum surrounded by several local optima and the task was not to escape the global one. In both cases, GC located the global optimum with high accuracies. Although radii-based evolutionary methods usually attain poor results for highly multimodal problems, results obtained for Schwefel function, which was considered for up to 100 dimensions, proved that GC remained very stable.

When the task was to find several global optima (like in Himmelblau function case) or find a fixed number of global and local optima (i.e. Six-Hump Camel Back function), the individuals that formed the final population represented exactly the expected optima. In order to achieve that, some tuning for the parameters which determine the mating and merging radii was performed. Complete results regarding function optimization may be found in [8] and [9].

4.2 Classification

Instead of using a unimodal genetic algorithm for the development of a learning classifier system (LCS), GC was chosen to be applied for classifying data by means of machine learning. The rules (if-then type) represent a population that is evolved by GC; the rules cover the space of possible inputs and they are evolved in order to successfully be applied for the problem to be solved - the problems may range from data mining to robotics.

Consider a data set and divide it into two parts – a training set and a test set. An LCS uses the samples in the training set i.e. the values of their attributes and the corresponding class for producing classification rules that are used in the decision-making process. The rules are finally used to predict the class for each of the samples in the test set.

An individual $c = (c_1, c_2, ..., c_n)$ is a string where each of the first n - 1 genes corresponds to an attribute of the samples from the data set. The last gene represents the outcome. Thus, an individual encodes an if-then rule. The condition is a conjunction of the first n – 1 attributes values and the conclusion is the class of the sample. Rules are evolved against the training set. The fitness of an individual is computed as its distance to all samples in the training set that have the same outcome (or class). Minimizing distances leads to good rules (representing cluster centers in the n-attribute variable space) for that outcome. A rule is considered successful if it matches the condition part of all data in the training set with the same outcome as itself. The match is determined using a distance measure between the individual (rule) and the object from the training set, experimentally defined as:

$$d(c,p) = \sum_{i=1}^{n-1} \frac{|c_i - p_i|}{b_i - a_i},$$

where a_i and b_i are the lower and upper bounds, respectively, of the i-th attribute. As the values for the n - 1 attributes belong to different intervals, the distance measure has to refer interval bounds.

At termination of the algorithm, population size (number of rules) equals the number of classes at least. For each test sample, the rule which is closest will provide the corresponding class.

This new LCS that may use the standard algorithm in GC or the variant with the crowding procedure within as engine has been successfully applied for several classification problems. Data sets considered dealt diabetes diagnosis for Pima-Indian data set or the iris classification in Fisher's data set [9]. The algorithm performed well and, in comparison to other methods, obtained accuracies were above average. An important result was its application on the issue of spam filtering [10], where results were very competitive.

4.2 Clustering [5], [6]

Other interesting attempts of dealing with clustering are presented in [1], [11]. In order to achieve a GC based unsupervised learning process upon a data set considered as in subsection above, individuals will encode cluster prototypes. They will be subsequently evolved by the means of GC, leading at convergence each to a cluster center. The number of cluster prototypes will be optimal with respect to the task at hand, as will the structure of each center.

An individual $c = (c_1, c_2, ..., c_{n-1})$ is therefore a string where each of the genes corresponds to an attribute of the samples from the data set. The expression of the fitness of an individual x_i is written as a sum of radial functions, centered in the samples to be clustered, and has to be maximized:

$$f(x_i) = \sum_{j=1}^{\text{popution size}} \frac{1}{e^{d(x_i, x_j)}}$$

Following experiments, the distance between two individuals is considered normalized Manhattan, as in previous section, since individuals encode attributes of samples. After centers are determined by GC, the assignation of samples to clusters is achieved by the method of closest prototype.

5 Conclusions

The main features of a recently proposed metaheuristic called genetic chromodynamics are presented in current paper. GC allows for a variable sized and emergent population. Solutions in the initial population are supposed to have different colors. Population dynamics is thus accompanied by color dynamics.

The metaheuristic involves a local interaction principle among individuals. Sub-populations co-evolve towards global and local optima. The final population contains as many solutions as (global and local) optima are detected.

Obtained results for the application of GC for various problems like function optimization, data classification or data clustering sustain that GC represents a metaheuristic rather than a common evolutionary technique.

References:

[1] J. Aguilar, A Data Mining technique for Data Clustering based on Genetic Algorithm, *Proceedings of the 6th WSEAS Int. Conf. on Evolutionary Computing*, Lisbon, Portugal, June 16-18, 2005, pp. 269-274.

[2] de Castro L. N., Von Zuben F. J., Learning and Optimization using the Clonal Selection Principle, *IEEE Transactions on Evolutionary Computation*, Special Issue on Artificial Immune Systems, vol. 6, No. 3, pp. 239-251, 2002.

[3] D. Dumitrescu, Genetic Chromodynamics, *Studia Universitatis Babes-Bolyai*, Ser. Informatica, Vol. 45, No.1, 2000, pp. 39-50.

[4] D. Dumitrescu, B. Lazzerini, L.C. Jain, A. Dumitrescu, *Evolutionary Computation*, CRC Press, Boca Raton, Florida, 2000.

[5] D. Dumitrescu, R. Gorunescu, Evolutionary clustering using adaptive prototypes, *Studia Universitatis Babes - Bolyai*, Seria Informatica, Volume XLIX, Number 1, 2004, pp. 15 – 20.

[6] D. Dumitrescu, *Genetic Algorithms and Evolution Strategies*, Blue Publishing House, Cluj – Napoca, 2000.

[7] K.E. De Jong, An analysis of the behavior of a class of genetic adaptive systems, *University of Michigan Press*, Ann Arbor, 1975.

[8] C. Stoean, M. Preuss, R. Gorunescu, D. Dumitrescu, Elitist Generational Genetic Chromodynamics - a New Radii-Based Evolutionary Algorithm for Multimodal Optimization, *The 2005 IEEE Congress on Evolutionary Computation*, 2005, pp. 1839-1846.

[9] C. Stoean, D. Dumitrescu, A New Algorithm within Genetic Chromodynamics. Applications to Function Optimization and Classification, *Technical Report*, Department of Computer Science, Faculty of Mathematics and Computer Science, Babes-Bolyai University, http://www.cir.cs.ubbcluj.ro.

[10] C. Stoean, R. Gorunescu, M. Preuss, D. Dumitrescu, An Evolutionary Learning Classifier System Applied to Text Categorization, *Annals of West University of Timisoara*, Mathematics and Computer Science Series, vol. XLII, 2004, pp. 265-278.

[11] Z.W. Zhong, N.G. Shankar, A Clustering Technique for Defect Inspection, *Proceedings of the 6th WSEAS Int. Conf. on Evolutionary Computing*, Lisbon, Portugal, June 16-18, 2005, pp. 281-284.