Seyed Mostafa Mansourfar, Fardin Esmaeeli Sangari, Mehrdad Nabahat / International Journal of Engineering Research and Applications (IJERA)ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 4, June-July 2012, pp.655-657

Genetic Algorithms Vs. Niching Methods On Clustering Undirected Weighted Graph

Seyed Mostafa Mansourfar*, Fardin Esmaeeli Sangari, Mehrdad Nabahat*****

*Sama technical and vocational training school, Islamic Azad University, Mamaghan branch, Mamaghan, Iran
**Sama technical and vocational training school, Islamic Azad University, Urmia branch, Urmia, Iran
***Sama technical and vocational training school, Islamic Azad University, Urmia branch, Urmia, Iran

ABSTRACT

In this paper, clustering undirected, weighed graphs will be under focus, whose decision problem is known to be NP-complete. Genetic algorithm and deterministic crowding technique of niching methods have been applied, and then we have compared performance of these two paradigms.

Keywords: Clustering, Genetic algorithms, graphs, Niching methods, NP-Complete

I. Introduction

Genetic algorithms (GAs) are generally used as an optimization technique to search the global optimum of a function. However, this is not the only possible use for GAs. Other fields of applications where robustness and global optimization are needed could also benefit greatly from GAs [2]. However, GAs has been successful in many of human competitive problems like optimization problems, classification problems, time series analysis, etc.

In this paper we have applied Genetic Algorithm and niching methods-a variety of genetic algorithms- to compare which one acts better on the issue. *Clustering* tries to divide a graph into k pieces, such that the pieces are of about the same size or weight and there are few connections between the pieces. An important application of graph partitioning is load balancing for parallel computing [9].

Complex problems such as clustering, however, often involve a significant number of locally optimal solutions. In such cases, traditional GAs cannot maintain controlled competitions among the individual solutions and can cause the population to converge prematurely. To improve the situation, various methods including niching methods have been proposed [1].

II. Related works

In recent years clustering and partitioning problem have great importance; among the proposed algorithms there are several evolutionary algorithms [9, 10, 11, and 12] that focused on the case.

III. Niching Methods

Main problem with Genetic Algorithm is premature convergence, that is, a non-optimal genotype taking over a population resulting in every individual being either identical or extremely alike, the consequences of which is a population that does not contain sufficient genetic diversity to evolve further.

Simply increasing the population size may not be enough to avoid the problem, while any increase in population size will incur the twofold cost of both extra computation time and more generations to converge on an optimal solution [3].

Niching methods have been developed to maintain the population diversity and permit the GA to investigate many peaks in parallel. On the other hand, they prevent the GA from being trapped in local optima of the search space [4].

Deterministic Crowding (DC) is an implicit neighborhood technique of niching methods that Mahfoud improved it by introducing competition between parents and children of identical niche. “Fig.1” (replacement process of DC) [5].

IV. Problem Definition

Clustering graph problem studied in this paper is defined as follow:

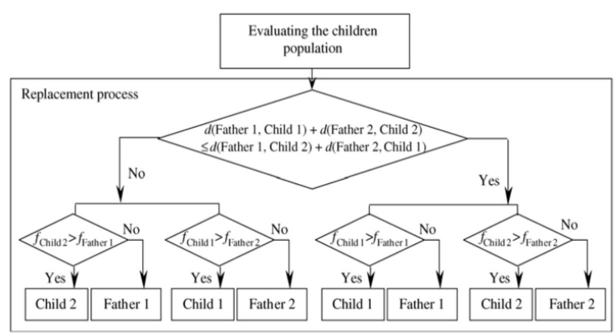


Figure 1. Replacement process in Deterministic Crowding Method

Consider a graph $G = (V, E)$, where V denotes the set of n vertices and E the set of edges. For a (k, v) balanced partition problem, the objective is to partition G into k components of at most size $v.(n/k)$, while minimizing the capacity of the edges between separate components. Also, given G and an integer $k > 1$, partition V into k parts (subsets) $V_1, V_2, \ldots, V_k$ such that the parts are disjoint and have equal size, and the number of edges with endpoints in different parts is minimized such that

- $\cup v_i = V$ And $v_i \cap v_j = \emptyset$ For $i \neq j$.

- $\boldsymbol{IN} = \sum_k \left(\frac{N_k(N_k-1)}{2} - A_{total}\right) + \sum \boldsymbol{external\ interconnections}$
  - *K:* maximum number of sub graph
  - N_k: Number of nodes of sub graphs
  - A_{total} : Internal interconnections

We need to minimize *IN* index.

V. Chromosome Representation

Assignment representation is used in this algorithm. Each chromosome, a string with length N, consists of nonnegative integer numbers less than or equal to K [8]. K is the number of clusters in the chromosome.

VI. Breeding operation

The breeding process is heart of the evolutionary algorithms. The search process creates new and hopefully fitter individuals.

Selection operation

The most widely used selection mechanisms are the *roulette-wheel selection* and *tournament selections* [6]. Roulette wheel selection could lead to high fitness individual of population dominated the direction of population evolution and eventually occupied the population [7]; Unlike, the Roulette wheel selection, the tournament selection strategy provides selective pressure by holding a tournament competition among N_u individuals, that is more efficient and leads to an optimal solution[3]. We use tournament selection that holds a competition between randomly selected portions of population and returns the winner for mating pool. Moreover elitism is also comes along with tournament selection to improve the performance.

Crossover and mutation operation

Crossover is the process of taking two parent solutions and producing children from them. In our algorithm we have applied Two-point crossover which two crossover points are chosen and the contents between these points are exchanged between two mated parents.

We have applied random mutation

.

VII. Experimental results

Implementation and test environment

Algorithms have been implemented in C# using Socket and Net namespaces. The architecture of all experiments consist Intel Pentium 4 processors running in 2.2GHz with 512MB of RAM.

Test graphs

We have tested the algorithms on 2 undirected graphs from DIMACS instance [13] which vertices' and edge's weights assigned with randomly integer numbers between 1 and 5. Both graphs adjacency matrices can be accessed in [15].

Test results

The following settings used for each of algorithms:

- GA: Population size=200; number of generations=500; elitism=10; Crossover algorithm and rate= Two-point crossover with 75%, Selection algorithm=Tournament method.
- DC: Population size=200; number of generations=500; Crossover algorithm and rate= Two-point crossover with 100%, Selection algorithm=Tournament method. In addition, Hamming distance is used in replacement process.

Experiments include comparing performance of DC and GA; all experiments have been repeated for 20 times.

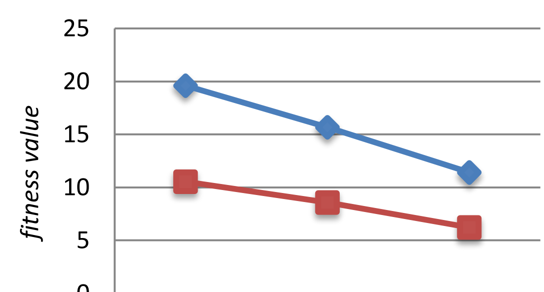
"Table I" shows characteristic of each test graphs.

"Fig.2" and "Fig.3" show obtained results of this experiment with both DC and GA.

"Table II" compares DC and GA from point of run time, clearly DC method is more demanding and takes considerably more time; of course reason is performing much more crossover operator also carrying out distance operator on chromosomes.

Table I
Test graph characteristics

Graph	vertices	edges	K
MYCIEL3	11	40	4
ANNA	138	986	10



	15%	30%	45%
GA	19.61	15.68	11.412
DC	10.546	8.5741	6.23

Figure 2. Fitness values for MYCIEL3

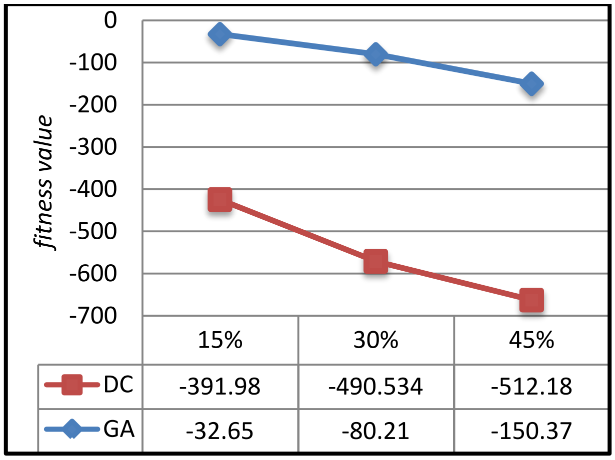


Figure 3. Fitness values for ANNA

Table II
Comparing DC and PGA from Run Time View

	ANNA	MYCIEL3
DC	35.78(s)	0.92(s)
PGA	4.1(s)	0.03(s)

VIII. Conclusions

In this paper we have reviewed a popular NP-complete problem, clustering on undirected, weighted graphs. We have used two practical heuristic algorithms, Deterministic Crowding of niching methods and Genetic Algorithm. As the results show DC method acts better than genetic algorithm; it is due to mechanisms of breeding this algorithm uses and survives better solutions for rest of operation.

Focusing on run time, DC method is more demanding than GA, which clearly is the only weak point of this method.

REFERENCES

[1] Weiguo Sheng, Xiaohui Liu, Mike Fairhurst, "A Niching Memetic Algorithm for Simultaneous Clustering and Feature Selection," IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 7, pp. 868-879, Jan. 2008

[2] Settea, Boullart, “Genetic programming: principles and applications”, Engineering Applications of Artificial Intelligence 14 (2001) pp.727–736

[3] S.N.Sivanandam, S.N.Deepa, *Introduction to Genetic Algorithms*, Springer Berlin Heidelberg New York, 2008, pp.10-20.

[4] B. Sareni, L. Krähenbühl, and A. Nicolas, “Efficient Genetic Algorithms for Solving Hard Constrained Optimization Problems," IEEE Transactions on Magnetics, vol. 36, no. 4, pp. 1027-1030 ,JULY, 2000.

[5] E.Perez, F.Hererra, C.Hernandez, “Finding multiple solutions in job shop scheduling by niching genetic algorithms,” Journal of intelligent manufacturing, vol. 14, no. 3-4, pp. 323-339, June 2003

[6] Petra Kudova, "Clustering Genetic Algorithm," dexa, pp.138-142, 18th International Conference on Database and Expert Systems Applications (DEXA 2007), 2007

[7] Da-jiang jin, Ji-ye zhang, “A New Crossover Operator For Improving Ability Of Global Searching,” Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, pp. 19-22, August 2007.

[8] Kazem Pourhaji, Shahriar Lotfi, “An Evolutionary Approach for Partitioning Weighted Module Dependency Graphs,” 4th International Conference on Innovations in Information Technology, 2007.

[9] Harpal Maini , Kishan Mehrotra , Chilukuri Mohan , Sanjay Ranka, “Genetic algorithms for graph partitioning and incremental graph partitioning,” Proceedings of the 1994 conference on Supercomputing, p.449-457, December 1994, Washington, D.C., United States

[10] Sahar Shazely , Hoda Baraka , Ashraf Abdel-Wahab, “Solving Graph Partitioning Problem Using Genetic Algorithms,” Proceedings of the 1998 Midwest Symposium on Systems and Circuits, p.302, August 09-12, 1998

[11] A. Cincotti, V. Cutello, M. Pavone, "Graph partitioning using genetic algorithms with ODPX," Proceedings of the 2002 Congress on Evolutionary Computation, vol. 1, pp.402-406.

[12] Siming Lin, Xueqi Cheng, BC-GA: A Graph Partitioning Algorithm for Parallel Simulation of Internet Applications , 16th Euromicro Conference on Parallel, Distributed and Network-Based Processing, pp. 358-365, Feb 2008.

[13] http://mat.gsia.cmu.edu/COLOR/instances.html

[14] http://en.wikipedia.org/wiki/Graph_partitioning

[15] http://sites.google.com/site/gppdataset/