Comparing performance ofparallel genetic algorithm and niching methods on partitioning connected, weighted graph

S. M. Mansourfar
Sama technical and vocational training college
Islamic Azad University, Urmia branch, Urmia, Iran
E-mail: mostafa.mansourfar@gmail.com

FardinEsmaeeliSangari
Sama technical and vocational training college
Islamic Azad University, Urmia branch, Urmia, Iran
E-mail: fardin_e_s@yahoo.com

MehrdadNabahat
Sama Technical and Vocational Training college
Islamic Azad University, Urmia branch Urmia, Iran
E-mail: m_nteach2009@yahoo.com

Abstract--In this paper, partitioning problem on graphs will be studied, whose decision problem is known to be NP-complete. Graphs used in this paper are connected and weighed. We have reviewed parallel genetic algorithm and niching methods, focusing on basic and important parameters; moreover we chose Deterministic Crowding (DC) of niching methods as main target of review. Deterministic Crowding is an implicit neighborhood technique that makes competition between parents and children of identical niche. We applied both parallel genetic algorithm and deterministic crowding on specified problem, then compared performance of these two methods. In addition, a new mutation algorithm named GPBM has been proposed. GPBM mainly focuses on balancing vertices between clusters of graph. As results showing GPBM rate have direct impact on finding better solutions. Tests applied for couple of times with different values so that obtained results be more mature. Some important characteristics of genetic algorithms were also reviewed.

***Keywords*- partitioning problem; genetic algorithm; graphs; GPBM**

I. Introduction

The basic concept of GA's designation was simulating processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. However, GAs has been successful in many of human competitive problems like optimization problems, classification problems, time series analysis, etc but they're suffering long computation time facing harder and complicated ones. Therefore applying parallel model of this method will be quite a good way to overcome drawbacks.

In this paper we have applied Parallel Genetic Algorithm to represent a new algorithm solving k-way graph partitioning problem (k-GPP).K-GPP 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 PGAs 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].

This paper also proposes a new algorithm which applies niching method as a second possible solution algorithm for K-GPP, and contains a comprehensive comparison between these two evolutionary algorithms.

This paper is organized as follow:we will talk about related works, and then introduce niching methods and its techniques, next we have problem definition, and detailed descriptions about our proposed algorithms, finally represent experimental evaluations and conclusion.

II. Related works

Over the years the graph partitioning problem has received a lot of attention, among the proposed algorithms there are several evolutionary algorithms [10, 11, 12, and 13] that focused on the case. However most of them are successful but important resources (CPU speed, network bandwidth) affecting load balancing haven't taken into accounts. It's the reason we have taken graph partitioning problem with complete set of elements influencing performance.

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].

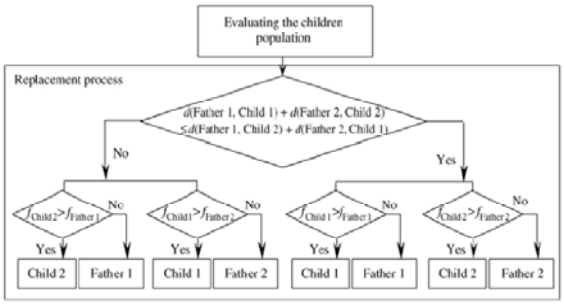

Fig 1. Replacement process in Deterministic Crowding Method

IV. ProblemDefinition

K-way graph partitioning problem studied in this paper is defined as follow:

Let G= (V, E) be a connected graph, where *V* denotes the set of vertices and *E* the set of edges. Let $w_1 : V \rightarrow R^+$is a positive weight function defined on set of vertices and $w_2 : E \rightarrow R^+$is a positive weight function defined on set of edges. Let k ≥ 2 be given integer, find a partition v_1, v_2, v_3, ...,v_k of the set of vertices such that

- $\cup v_i = V \; And \;\; v_i \cap v_j = \emptyset For \; i \neq j.$
- $F_v = \sum_{i=1}^{k} |w(i) - \frac{W_v}{k}|$**(1)**

  Where W_v is total sum of weights of vertices in V and *W*(i) is sum of weights of vertices in V_i.
- $F_E = \sum_{i,j=1}^{k} E_{i,j} - \sum_{i=1}^{k} A_i$**(2)**

  Where A_i is sum of weights of intra connections in V_i and $E_{i,j}$ is sum of weights of interconnections between $V_i$andV_j.

V. ProposedAlgorithm

In order to achieve better results, some parameters and specifications need to be explained. Encoding in genomes, methods for Genetic Operators, selection function, defining a proper fitness function, and migration topology have considerable influence on how much the results are optimal. In this section we try to explain some of these parameters in our PGA and DC.

V.1. Chromosome Representation

Assignment representation is used in this algorithm. Eachchromosome, a string with length N, consists of nonnegative integer numbers less than or equal to K [9]. K is the number of clusters in the chromosome. For example graph in "Fig.2" is encoded as following string:

Chromosome: 2 1 2 1 3 1 (k=3)

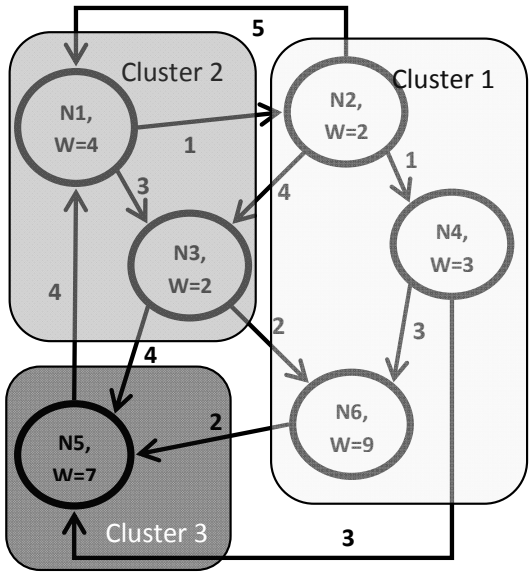

Fig2. Sample Graph

V.2. Fitness Function

As mentioned in problem definition, both vertices and edges have direct impact in quality of obtained results; therefore fitness function must provide a tradeoff between these two factors. Considering "(1)" and "(2)" we can use average of F_v and F_E as a fitness functions. So we can calculate fitness as:

In addition, the fitness functions is a minimization function, which means results with minimum fitness function is better.

Fitness Function=$\frac{F_v+F_E}{2}$ **(3)**

V.3. Breeding operation

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

V.3.1. 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.

V.3.2. Crossover operation

Crossover is the process of taking two parent solutions and producing children fromthem.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.

V.3.3. Mutationoperation

Mutation prevents the algorithm to be trapped in a local minimum and is viewed as a background operator to maintain genetic diversity in the population. We have proposed a new algorithm named *GPBM* for concerning problem. This algorithm focuses on balancing vertex weighswhichtries to smartly generate a child solution based on parent solution in a way that good characteristic of current solution is not lost. "Fig.3" shows GPBM algorithm.

GPBM (chromosome)
1. M=Determine Number of changing genomes in chromosome
2. Sort clusters based on intra cluster vertices weight
3. Repeat for M times
   Begin
   - ✓ Choose cluster C_B with biggest intra weight and C_L with lowest intra weight
   - ✓ Change genomes of C_B in chromosome with C_L, starting at a random bit of chromosome string, provided that $|C_B|>=1$ and
   - ✓ Sum of changed vertex's (that assigned to another cluster) weights $< \frac{w(C_B)}{2}$

   End.

Fig 3. GPBM mutation algorithm

Example

As we can see in "Fig.2" encoded chromosome is <2 1 2 1 3 1>. M, number of changing genomes in chromosome is considered 1. That is:

$$W(\ Cluster\ 1) = 14$$

$W(\ Cluster\ 2) = 6$

$W(\ Cluster\ 3) = 7$

$F_v = |14 - 9| + |6 - 9| + |7 - 9| = 10$

$$F_E\ = 25 - 7 = 18$$

$fittness\ function = (10 + 18\ )/2 = 14$

So cluster 1 should be substituted with cluster 2, if 2nd bit of chromosome is randomly chosen, chromosome is changed to <2 **2** 2 **2** 3 1>base on stopping criteria (see "Fig.4"); resulting chromosome's intra cluster vertices weight is calculated:

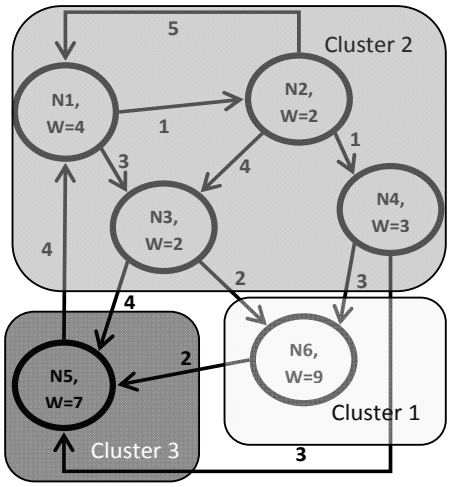

Fig 4.Sample graph after applying GPBM mutation

$W(\ Cluster\ 1) = 9$

$W(\ Cluster\ 2) = 11$

$$W(\ Cluster\ 3) = 7$$

$$F_v = |9 - 9| + |11 - 9| + |7 - 9| = 4$$

$$F_E\ = 18 - 14 = 4$$

$fittness\ function = (4 + 4\ )/\ 2 = 4$

V.4. Migration policy and migration topology

There are some new parameters introduced with PGA, which must be carefully considered while designing parallel algorithm. Migration process tries to prevent premature convergence and share high quality solutions. Chromosome migrations occur in fixed intervals with each deme sending a copy of its locally selected chromosome to another deme. There are four popular migration policies, good migrants replace bad individuals, good migrants replace random individuals, bad migrants replace random individuals and random migrants replace random individuals [8].

Another important component is topology of the interconnection between demes. It determines how fast (or how slow) a good solution disseminates to other demes. If the topology has a dense connectivity (or a short diameter, or both) good solutions will spread fast to all the demes and may quickly take over the population. On the other hand, if the topology is sparsely connected (or has a long diameter), solutions will spread slower and the demes will be more isolated from each other, permitting the appearance of different solutions. The general trend on coarse-grained parallel GAs is to use static topologies that are specified at the beginning of the run and remain unchanged. Hypercube and ring topologies commonly applied in most implementations [2].

VI. Experimental results

VI.1. Implementation and test environment

Algorithms have been implemented in C# using Socket and Net namespaces. The architecture of all experiments consists of 4demes utilizing withIntel Pentium 4processors running in 2.2GHz with 512MB of RAM.Ring topology is used for communication between demes.

VI.2. Test graphs

We have tested the algorithms on 2directed graphs from DIMACS instance [14] which vertices' and edge's weights assigned with randomly integer numbersbetween1 and5. Both graphs adjacency matrices can be accessed in [16].

VI.3. Test results

The following settings used for each of algorithms:

- PGA:
  Population size=200; number of generations=500; elitism=10; number of Island=4; migration interval =20; migration frequency=15%;Crossover algorithm and rate= Two-point crossover with 75%, Mutation algorithm and rate=GPBMwith30%; Selection algorithm=Tournament method.
- DC:
  Population size=200; number of generations=500; number of Island=4; migration interval =20; migration frequency=15%; Crossover algorithm and rate= Two-point crossover with 100%, Mutation algorithm and rate= GPBM with 30%; Selection algorithm=Tournament method.
  In addition, Hamming distance is used in replacement process.

Experiments include comparing performance of DC and PGA;italso checksimpact of proposed mutation algorithm, GPBM, on obtained results; finally different migration policies will be investigated in both mentioned algorithms. All experiments have been repeated for 20 times.

"Table I" shows characteristic of each test graphs.

Table 1 :Test graph characteristics

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

Taking GPBM operator under focus, we have conducted several runs with three different mutation rates; "Fig.5" and "fig.6" show obtained results of this experiment with both DC and PGA, as it indicates any increase in rate of mutation has considerable effect in results.

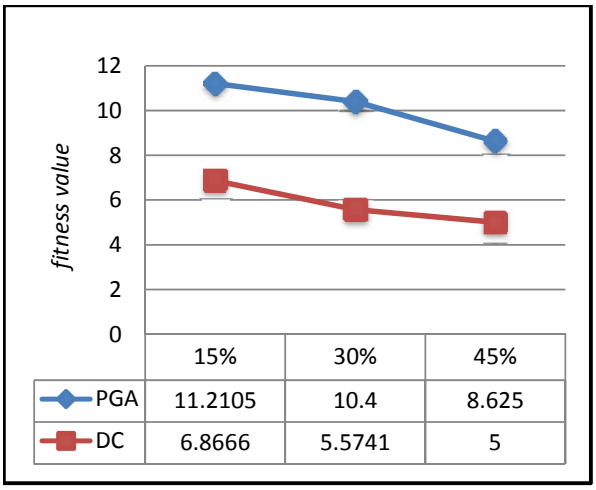

Fig5. Fitness values, increasing GPBM mutation rate for MYCIEL3

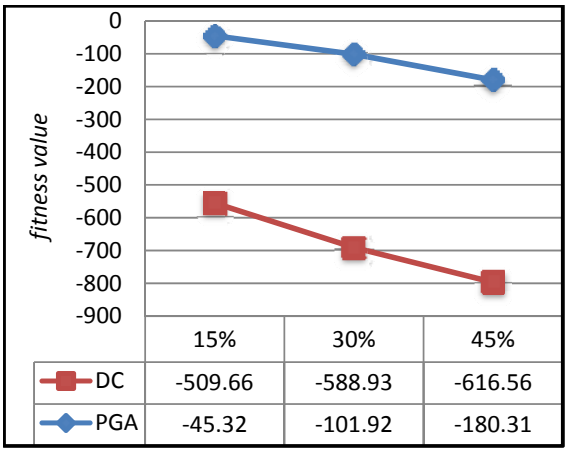

Fig6. Fitness values, increasing GPBM mutation rate for ANNA

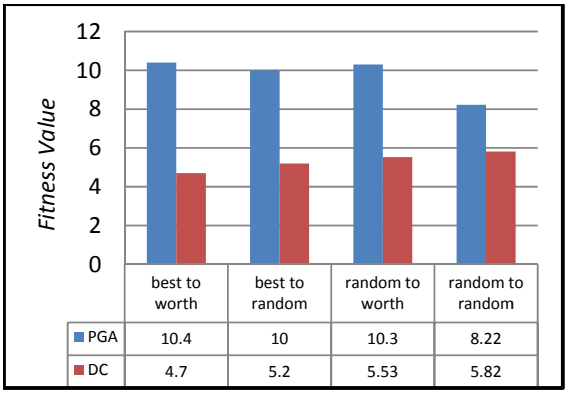

Fig7. Comparing effect of each migration strategy in MYCIEL3

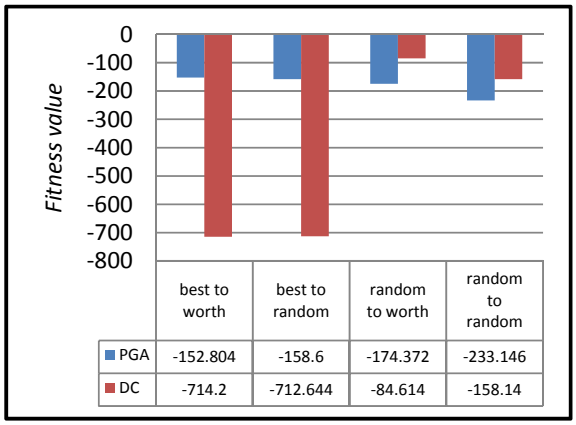

Fig8. Comparing effect of each migration strategy in ANNA

"Fig.7" and "fig.8" show effect of each migration strategy on test graphs, as we can see *"Best to Worth"* strategy works better for DC method and *"Random to Random"* strategy is more efficient in PGA."Table2" compares DC and PGA 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.

VII. Conclusions

Table 2: Comparing DC and PGA from Run Time View

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

In this paper we have reviewed a popular NP-complete problem, K-way graph partitioning on directed, weighted graphs. We have used two practical heuristic algorithms, Deterministic Crowding of niching methods and Parallel Genetic Algorithm; both of these algorithms have been implemented on coarse-grained parallel model.UsingDC we were able to achieve even better results over the PGA in this problem and the difference got bold as test graphs grow. Moreover, a new mutation algorithm represented which considerably has positive effect on obtained results. Finally migration strategies on parallel evolutionary algorithms were on sight. As experiments show different migration strategies are suitable for different algorithms.

Focusing on run time, DC method is more demanding and time consumer than PGA, 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] Erick Cantú-Paz, "A Survey of Parallel Genetic Algorithms," Illinois Genetic Algorithms Laboratory University of Illinois at Urbana-Champaign

[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-jiangjin, 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] Erick Cantú-Paz, MigrationPolicies and TakeoverTimesinParallelGenetic Algorithms**,** IlliGAL Report No. 99008. January, 1999. Illinois Genetic AlgorithmsLaboratory University of Illinois at Urbana-Champaign

[9] KazemPourhaji, ShahriarLotfi, "An Evolutionary Approach for Partitioning Weighted Module Dependency Graphs**,"**4th International Conference on Innovations in Information Technology, 2007.

[10] HarpalMaini , KishanMehrotra , 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

[11] SaharShazely , 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

[12] 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.

[13] SimingLin, 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.
[14] http://mat.gsia.cmu.edu/COLOR/instances.html
[15] http://en.wikipedia.org/wiki/Graph_partitioning
[16] http://sites.google.com/site/gppdataset/