ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 10, October 2013

Solving the Graph Partitioning based on Cuckoo Optimization Algorithm (COA)

Sina Zangbari Kouhi[1], Faranak Nejati[2], Dr. Jaber Karimpour[3]

Department of Computer Science, University of Tabriz, Tabriz, Iran [1]

Department of Computer Science, University of Tabriz, Tabriz, Iran [2]

Professor, Department of Computer Science, University of Tabriz, Tabriz, Iran[3]

Abstract: In this paper a new efficient approach to solving the balanced connected partitioning is presented. The graph partitioning problem has been used in many areas of computer science like LSI design; electrical power networks (EPNs) and ect. The problem aims to obtaining sub graphs of a graph which include balance connected vertices. The proposed solution is based on Cuckoo Optimization Algorithm (COA) that is a novel evolutionary optimization algorithm. This method is inspired by the life structure of a bird family which is called cuckoo. To prove the performance of this approach two hardness instances are used.

Keyword: Graph partitioning, COA, optimization algorithm, meta-hubristic.

I. INTRODUCTION

The balanced partitioning problem [1, 2, 3 and 8] of a graph arises in many areas of computer science, like image processing, VLSI design, electrical power networks (EPNs), operating systems, clustering analysis and many other subjects. The problem that we focus in this work, called k-connected problem (k-cp) and defined as follow:

Let G be a connected graph with a set of vertices $V = \{V_i, i = 1 \dots n\}$, then G has a k-connected sub graph (G_1, G2 … G_k), $k \geq 2$, such that the following value is maximized:

$$\text{Min } \{|V_1|, |V_2| \dots |V_k|\}$$

Basically, the problem [11, 12, 13 and 14] is to break a connected graph into k balanced connected sub graphs such that the total number of vertices in each sub graphs is (optimally) equal. The need for solving this problem a number of different approaches has been suggested by the research community. Examples include METIS [5, 10 and 11], Genetic Algorithm [7], Genetic Algorithm with novel crossover (ODPX) [6], JOSTEL [9] and ect. In this paper we employ Cuckoo Optimization Algorithm (COA) [4] for solving the Balanced Graph Partitioning problem. The results of COA compare with classical method PMETIS, Genetic Algorithm with novel crossover (ODPX) that found in [6] and this method apply to grid graph that is one of hardness instances.

The remainder of the paper is organized as follows: In section 2, gives short presentation of (COA). In section 3, COA is applied to the graph partitioning problem and in section 4, we present experiments and comparisons. Finally, we have conclusion of remarks.

II. CUCKOO OPTIMIZATION ALGORITHM

Cuckoo Optimization Algorithm is inspired by the life method of a bird family which is called cuckoo. Special behaviours of cuckoos and their manners in egg laying and breeding has been the basic motivation for development of this new evolutionary optimization algorithm. In this model, cuckoos appear in two models: mature cuckoos and eggs. Mature cuckoos lay eggs in number of host bird's nest. The grown eggs determine the suitability of the environment and if an area includes the more live eggs then the more profit is assigned in that area and we can say COA going to optimize.

Evolutionary algorithms like PSO and ACO usually start with initial population. This population includes values of problem variables that formed as an array. This array in cuckoo optimization is called habitat which representing current living position of cuckoo. In an N_{var}-dimensional optimization problem a habitat is an array of $1 \times N_{var}$ and defined as follow [4]:

$$\text{Habitat} = [x_1, x_2 \dots x_{N_{var}}] \quad (1)$$

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 10, October 2013

The profit of a habitat obtained as follow [4]:

$$\text{Profit} = fp\,(\text{habitat}) = fp\,(x_1, x_2 \ldots x_{Nvar}) \quad (2)$$

After generated matrix of initial cuckoo's habitat some randomly produced number of eggs for each of them will be considered. Cuckoos lay eggs within a maximum distance from their habitat which is called "Egg Laying Radius (ELR)". ELR is defined as [4]:

$$ELR = \alpha \times \frac{\#current\_cuckoo's\_eggs}{Total\_number\_of\_eggs} \times (var_{Hi} \times var_{Lo}) \quad (3)$$

Where α supposed to handle the maximum value of ELR and var_{Hi}, var_{Lo} are upper limit and lower limit of variables.

So after egg laying process, **p%** of all eggs (usually 10%), with less profit values and have no chance to grow, will be killed. When the remained eggs grow and turn into mature they make society in own area to live in for some time. The society with best profit value will be the destination for the cuckoos in other society to immigrate. All cuckoos will be immigrated toward the best habitat then they will inhabit somewhere near the goal point and starts to lay eggs in Your paper must be in two column format with a space of 4.22mm (0.17") between columns.

Random number of host bird's nest inside her ERL. This process continues until convergence of all cuckoos in one society becomes more than 95%.

III. PROPOSED METHOD

First of all, we generate initial habitat of cuckoos which are randomized solutions. We must produce $N_{numCuckoos}$ of them. In this problem each habitat is array includes a permutation from graph vertices. We set the value of COA parameters that depicts in TABLE 1.

TABLE 1: THE VALUE OF COA PARAMETERS

Parameters of COA	Values
Optimization parameter	array
Number of initial population of cuckoo	5
Minimum number of eggs for each cuckoo	2
Maximum number of eggs for each cuckoo	4
Number of clusters	1
Maximum number of cuckoos	50
maximum value of ELR (α)	2

To evaluate profit must create graph partition by permutation of vertices that denoted in cuckoo's habitats. In order to, we select K element of these habitats and set them as partitions leader of main graph. Then, the remaining N-K elements must be assigned to each part. We allocate each element to the part that is smaller than others and one of its members has common edge with that element. Sudoku of create partition is the following:

Create Partition Algorithm

For each element of cuckoo's habitat: i ε [k+1 …N]
While there are free vertices do
If P_i is free then
Assign P_i to the smallest adjacent partition;

Our cost function according as (4) that suggested in [5], in the rest of the paper, the minimization problem will be considered:

$$MIN\ F(X) = \frac{K\,.\,M}{|V|} \quad (4)$$

Which the $|V|$ is number of graph vertices, K is the number of partitions and M is total number of vertices in the largest sub graph. The profit of each cuckoo's habitat evaluate by $F(x)$.
To immigrate, we change each element of array with respect their ELR toward the best habitat. More details regarding the COA algorithm to solve this problem can be found in [4].

IV. EXPERIMENTS AND RESULTS

To investigate the efficiency of the COA, we used random graph of 5000 vertices and grid graph. In Table 2, we compare results of COA with PMETIS and Genetic Algorithm (ODPX). The results show that COA can produce accurate solution than classic method PMETIS and Genetic Algorithm (ODPX).

Table 2: Results of PMETIS, ODPX and COA

Number of Partitions	Methods		
	PMETIS	Using Genetic Algorithm (ODPX)	Using COA
50	1.01	1	1
100	1.02	1	1
200	1.08	1.02	1
400	1.18	1.04	1.02
600	1.34	1.08	1.08
800	1.40	1.12	1.10
1000	1.44	1.20	1.16

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 10, October 2013

Following figures show the convergence history of random graph obtained by different partitions. Normally, 100 runs are executed for each *k* and we selected the best results. This method produces better results than other but it is slow.

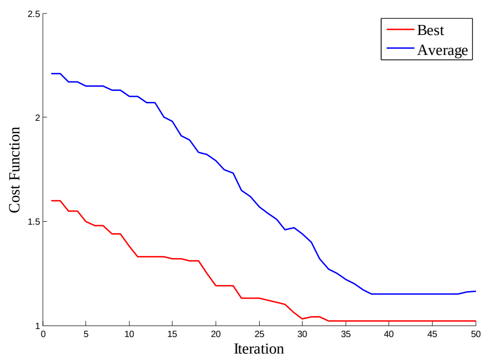

Figure 1: The convergence history, k = 200.

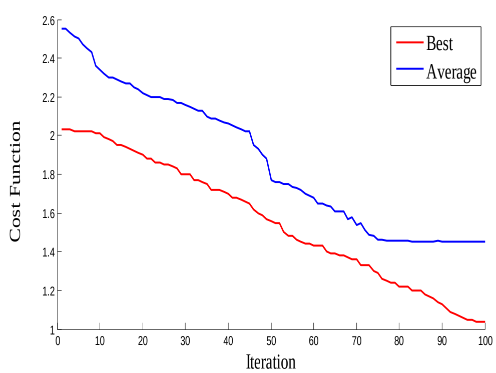

Figure 1: The convergence history, k = 400.

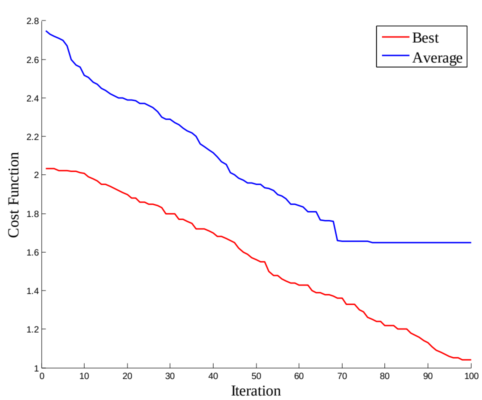

Figure 1: The convergence history, k = 800.

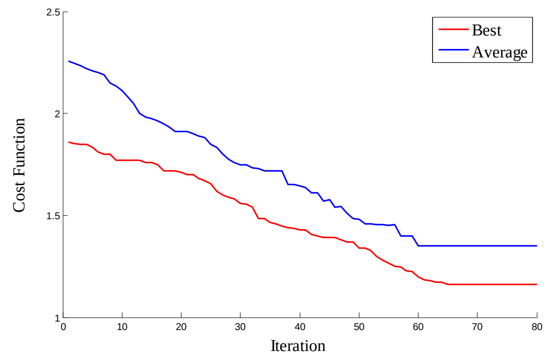

Figure 1: The convergence history, k = 1000.

Another type of graph that we used to demonstrate the performance of the COA is grid graph (mesh graph). Grid graphs are the hardest test samples of the k-cardinality tree problem (KCTP) that presented in [15, 16]. It does not mean that this type of graphs is too hard for our problem. Figure 5 depicts a two dimension grid graph. Each node of them can present a task and have a number.

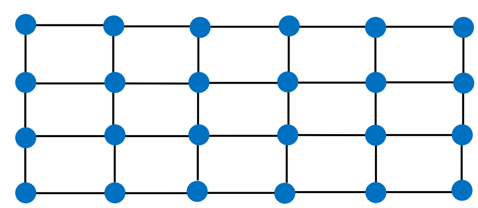

Figure 5: Two dimension grid graph $_{4\times6}$.

We apply COA to create balance partition of grid graph. The best results of 100 runs are shown in the following tables.

Table 3: Results of COA in grid graph $_{5\times5}$

Graph $_{m\times n}$	**Num of vertices**	**Num of edges**	**Num of partition**	***F(x)***
05 × 05	25	40	5	1.02

Table 4: Results of COA in grid graph $_{5\times10}$

Graph $_{m\times n}$	**Numb of vertices**	**Numb of edges**	**Numb of partition**	***F(x)***
05 × 10	50	85	5	1.02
			10	1.04

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 10, October 2013

Table 5: Results of COA in grid graph $_{5 \times 20}$

Graph $_{m \times n}$	**Numb of vertices**	**Numb of edges**	**Numb of partition**	***F(x)***
05 × 20	100	175	2	1.02
			5	1.04
			10	1.06

Table 6: Results of COA in grid graph $_{15 \times 15}$

Graph $_{m \times n}$	**Numb of vertices**	**Numb of edges**	**Numb of partition**	***F(x)***
15 × 15	225	420	3	1.04
			5	1.04
			15	1.06
			45	1.08

V. CONCLUDING REMARKS

In this paper, we presented a new approximation approach for solving k-balanced partitioning problem. The approach extends with Cuckoo Optimization Algorithm (COA) to break a graph into k (optimally) equal parts. The goal is to balance number of connected vertices among the partitions. This method applies to a random graph with 5000 vertices and grid graph with different vertices. The obtained experimental results of random graph compared with METIS and ODPX method. The results prove that proposed method produce accurate solutions than METIS and ODPX. Also, this method can produces the better results in small *K*s and the *K*s that are considered multiples of vertices number. Regarding future work, we are desired in testing this approach on a set of standard benchmark instances.

REFERENCES

[1] Chleb´ıkov´a, J.: Approximating the Maximally Balanced Connected Partition Problem in Graphs. Information Processing Letters, Vol. 60, 1996, pp. 225, pp. 230.
[2] L. R. B. Salgado. Algoritmos de Aproximac¸ ˜ao para Partic¸ ˜oes Conexas em Grafos. PhD thesis, Universidade de S˜ao Paulo, 2004.
[3] L. R. B. Salgado and Y. Wakabayashi. Approximation results on balanced connected partitions of graphs. In Latin-American Conference on Combinatorics, Graphs and Applications, volume 18 of Electron. Notes Discrete Math., pages 207–212 (electronic). Elsevier, Amsterdam, 2004.
[4] R. Rajabioun "Cuckoo Optimization Algorithm" ; Elsevier; Applied Soft Computing 11 (2011) 5508–5518; 2011.
[5] G. Karypis and V. Kumar "MÉTISUnstructured Graph Partitioning and Sparse Matrix Ordering System"-Version 2.0GEORGE KARYPIS AND VIPIN KUMAR, *University of Minnesota, Department of Computer Science,* Minneapolis, August 26, 1995.
[6] A. Cincotti, V.Cutello, M. Pavone" Graph partitioning using genetic algorithm with ODPX,".Dept of Mathematics and Computer Science, *University of Catania*.
[7] B. Djuri´C, J. Kratica, D. Toˇsi´c, V. Filipovi´, "Solving the Maximally Balanced Connected Partition Problem in Graphs by Using Genetic Algorithm"; Computing and Informatics, Vol. 27, 341–354, 2008.
[8] E. Gy˝ori. On division of graphs to connected subgraphs. In Combinatorics (Proc. Fifth Hungarian Colloq., Keszthely, 1976), Vol. I, volume 18 of Colloq. Math. Soc. J´anos Bolyai, pages 485–494. North-Holland, Amsterdam, 1978.
[9] Chris Walshaw and M. Cross. Mesh partitioning: A multilevel balancing and refinement algorithm. *SIAM Journal on Scientific Computing,* 22:63–80, 2000.
[10] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal of Scientific Computing*, 20(1):359–392, 1998.
[11] George Karypis and Vipin Kumar. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing*, 48(1):96–129, 1998.
[12] K. Andreev" - balanced graph partitioning," *Mathematics Department, Carnegie-Mellon University, HaraldR¨acke, Mathematics Department, Carnegie-Mellon University,* November 2006.
[13] Charles-Edmond Bichot. A new method, the fusion fission, for the relaxed k-way graph partitioning problem, and comparisons with some multilevel algorithms. *Journal of Mathematical Modeling and Algorithms (JMMA)*, 6(3):319–344, 2007.
[14] G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski, "Pregel: A System for Large-Scale Graph Processing," in Proceedings of the 28th ACM symposium on Principles of distributed computing (PODC'09). New York, NY, USA: ACM, 2009, pp. 6–6.
[15] Blum, C.—Ehrgott, M.: Local Search Algorithms for the k-Cardinality Tree Problem. Discrete Applied Mathematics, Vol. 128, 2003, No. 2–3, pp. 511–540.
[16] Brimberg, J.—Uroˇsevi´c, D.—Mladenovi´c, N.: Variable Neighborhood Search for the Vertex Weighted k-Cardinality Tree. European Journal of Operational Re- search, Vol. 171, 2006, pp. 74–84.