Solving the Graph Bisection Problem with Imperialist Competitive Algorithm

Hodais Soltanpoor [1], Shirin Nozarian [1] and Majid VafaeiJahan [2]

[1] Young Researchers Club, Mashhad Branch, Islamic Azad University, Mashhad, Iran

[2] Department of Computer Engineering, Islamic Azad University, Mashhad Branch, Iran

Abstract. Imperialist competitive algorithm is a method in complementary calculations which is dealing with finding an optimum response in different optimization problems. Though its initial versions are introduced in order to solve the optimization problems, it is used in discrete problems, too. The Binary discrete method is offered in this article on the basis of mentioned algorithm to solve the problem of Graph Bisection. Graph Bisection means dividing the graph into two almost equal sections, with minimum connection between them. Considering the results of other suggested methods and their comparison with other optimization algorithms, such as genetic algorithm, particles swarm optimization, ant colony, tabu search, and simulated annealing, it can be deduced that the Binary discrete imperialist competitive algorithm performs better in different tests. Results cleared that this algorithm performs around 11% better than other compared methods.

Keyword: Problem of Graph Bisection, Genetic Algorithm, Tabu Search, Simulated Annealing, Ant Colony, Imperialist Competitive Algorithm.

1. Introduction

This article is dealing with solving the problem of graph bisection and comparing the genetic algorithm (GA), ant colony (ACO), tabu search (TS), simulated annealing (SA), and imperialist competitive algorithm (ICA). Dividing the graph is one of the most important problems which can be applied in different fields. Some applications like scientific calculations, VLSI designing, tasks schedules, and geographical data systems.

The problem can be defined as follow: dividing the graph into P sections almost equally, with minimum connection between them. The efficient execution of most parallel algorithms usually needs a solution for dividing the graph. As the apexes represent the calculations and the connections represent the transacted data. Dividing the calculation graph into P sections is one solution for offering the tasks to P processors. Since this division relates an equal number of calculation tasks to each processor, the performance will be balanced. And since division makes the connection cut minimum, the costs of relating the processors will be the least, too.

The problem of graph division classifies in NP-Complete problems. But many algorithms are designed which offer appropriate divisions for that. Spectrum division methods [1] offer proper divisions of the graph, but they are calculated complicatedly. Geometric division methods [2, 3] are quick, but offer weaker divisions. Furthermore, geometric methods are applicable only when coordinating data are available. After them, a new method is offered to divide the graph. These methods, known as Graph multi layer division, are moderately complicated for calculation [1, 4, 5, 6]. Though the method of multilayer is quicker than spectrum division, the paralleling in this method is more useful. Up to now, lots of researches are accomplished to design the parallel algorithms for dividing the graphs [7, 8, 9].

Dividing the graph into two equal sections, with minimum connection between the apexes in different sections has been one of the points in dividing the graph. Regarding the growing usage of graph bisection problem, and also heuristic methods in solving complicated problems, this article is dealing with solving the problem by genetic algorithm, ant colony, tabu search, simulated annealing, and imperialist competitive algorithm.

An extensive definition of the graph partitioning problem is presented in the second part of the article. Explaining the imperialist competitive algorithm is offered in third, tests and comparing the algorithms to solve the mentioned problem is offered in fourth part of this article. Finally, some results and suggestions for future works are added to section fifth.

2. An Explanatory Definition Of The Problem

The problem generally can be defined as follow: The graph input, without weight and direction $G = (V, E)$ receives some even apexes (n), and in output it offers a division of (v) with two substrates $\{V_1, V_2\}$ in equal sizes. The cost of each bisection equals the number of connections $c = (a, b) \in E$ in which $a \in V_1$ and $b \in V_2$. The main goal is reducing the cost of bisection in minimum.

Saran and Vazirani offered an algorithm in a polynomial to estimate the amount of bisection [11]. They showed that its amount is an index of $n/2$ and there is no better estimation for this algorithm. Then Feige et al. improved this estimation by an index of $n\sqrt{n}$ [12]. The amount of bisection can be calculated in a polynomial time in some classes of this problem. Papadimitriou and Sideri offered this algorithm for a network chart [13]. But generally the problem of graph bisection is classified in NP-Complete problems [14, 10] and approximate solutions are offered for it in several articles like [15, 16, 17]. The absolute solution of this problem was offered by Kemighan et al. in [18]. Marks et al. could offer a new method by combining the Kernighan-Lim algorithm and hill-climbing parallel algorithm, offered in [19], and seed-growth algorithm [20] to solve this problem. They called this method PHC/SG+KL [21]. After them Wang et al. tried to solve the problem in [22] by nervous system. They compared their method with PHC/SG+KL method and Hopfield system. Evaluating the results showed that their method performs better comparing to other methods. Then Chen et al. offered a combined method in [23] on the basis of genetic algorithm. They proved that this method performs better than the previous ones, including PHC/SG+KL or nervous system based method.

3. Imperialist Competitive Algorithm

Like other evolutionary optimization methods, this method begins with some initial population. Every element of the population is called a *country* in this algorithm. Countries are classified in two different groups: *Colony* and *Imperialist*. Every colonialist brings some colonies under its dominion and controls them according to its power [24]. Assimilation policy and imperialist competition is the core of this algorithm. This policy is accomplished by moving the colonies towards empires on the basis of a determined connection which is defined completely in [25].

Generally, imperialist competitive algorithm is unlimitedly used to call every continues optimization problem. Therefore this algorithm is used easily in different fields, such as electrical engineering, mechanics, industries, management, civic developments, artificial intelligence, etc.

4. Simulation Results

In this part of the research the results of the tests will be offered. Mentioned algorithms are executed in a system with 1.67 GHz processor and 1G of RAM memory, in MATLAB programming software. A graph was randomly created at the size of 20*20 for execution, and it was considered as the initial data. Figure 1 depicts the best amounts of response in a cycle of 100 repetitions of algorithms in the same condition. Same condition means the equality in the number of generations and the size of populations in different algorithms. It should be mentioned that global optimization in this problem equals 17 which is clear in all linear D-ICA comparing with other methods. As it can be seen, D-ICA performs more efficiently than other methods. Figure 1 reveals that D-ICA achieves the response most of the time, while GA algorithm never achieves it.

As it can be seen in figure2, ICA achieves the response most of the time, while SA algorithm performs more fluctuation.

Figure 5 depicts the percentage of algorithms ability to approach the global optimum response. As you can see D-ICA shows a great percentage in order to reach the global optimum.

5. Results And Future Works

Firstly, a review of graph bisection problem was accomplished in this article. Then imperialist competitive algorithm was explained to solve the discrete problem of graph bisection. Regarding the accomplished executions, it can be deduced that this algorithm performs better than some other well-known heuristic algorithms. Regarding the proper results of this algorithm in graph bisection problem,

imperialist competitive algorithm can be utilized in future studies for more general problems like Graph Partitioning.
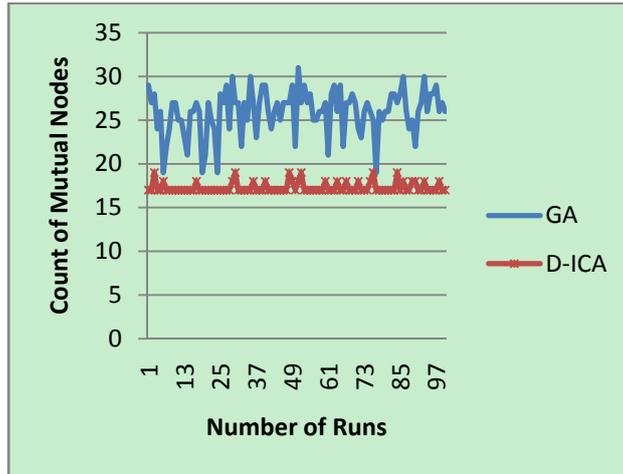
Figure.1: Comparison of ICA and GA in solving Graph Bisection Problem
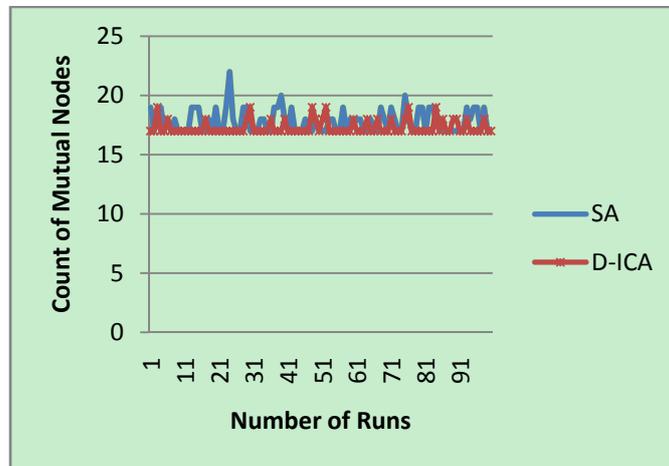
Figure.2: Comparison of ICA and SA in solving Graph Bisection Problem
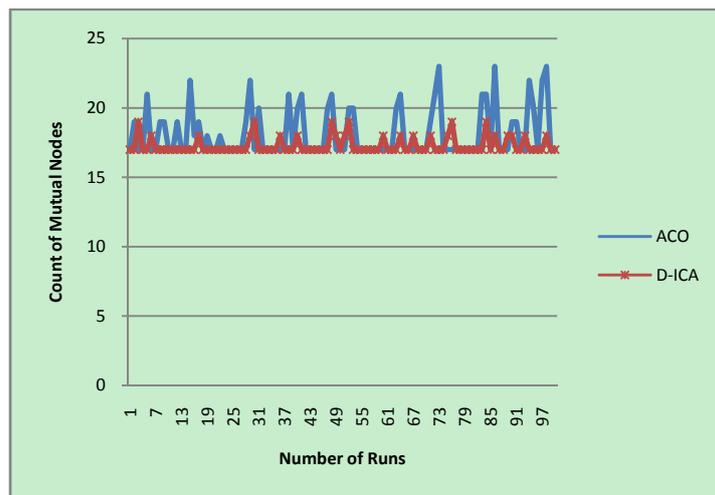
Figure.3: Comparison of ICA and ACO in solving Graph Bisection Problem
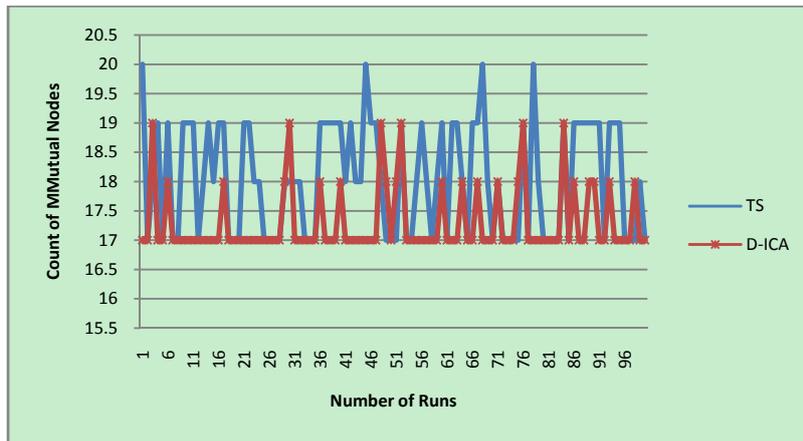
Figure.4: Comparison of ICA and TS in solving Graph Bisection Problem
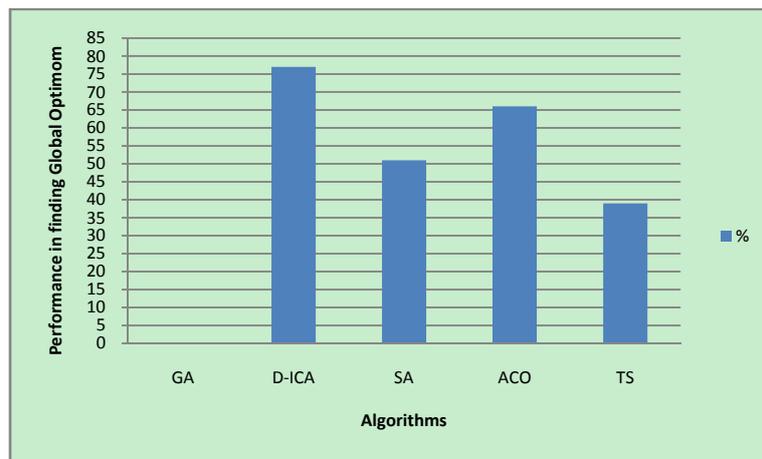
Figure.5: percentage of algorithms ability to approach the global optimum. As you can see genetic algorithm never achieves the optimum and its percentage is zero, in the other hand D-ICA achieves he top percentage in reaching the optimum.

6. References

[1] Hendrickson, B. and R. Leland (1995). A multilevel algorithm for partitioning graphs, ACM.

[2] Heath, M. T. and P. Raghavan (1995). "A Cartesian parallel nested dissection algorithm." SIAM Journal on Matrix Analysis and Applications 16: 235.

[3] Miller, G. L., S. H. Teng, et al. (1991). A unified geometric approach to graph separators, IEEE.

[4] Karypis, G. and V. Kumar (1995). Analysis of multilevel graph partitioning, ACM.

[5] Karypis, G. and V. Kumar (1996). Parallel multilevel k-way partitioning scheme for irregular graphs, IEEE.

[6] Karypis, G. and V. Kumar (1998). "A fast and high quality multilevel scheme for partitioning irregular graphs." SIAM Journal on Scientific Computing 20(1): 359-392.

[7] Ghose, M. and E. Rothberg (1994). A parallel implementation of the multiple minimum degree ordering heuristic, Technical report, Old Dominion University, Norfolk, VA.

[8] Hendricksonz, P. D. S. P. B. and R. Lelandz (1995). Parallel algorithms for dynamically partitioning unstructured grids, Citeseer.

[9] Gilbert, J. R. and E. Zmijewski (1987). "A parallel graph partitioning algorithm for a message-passing multiprocessor." International Journal of Parallel Programming 16(6): 427-449.

[10] Johnson, D. S. and M. R. Garey (1979). "Computers and Intractability: A Guide to the Theory of NP-completeness." Freeman&Co, San Francisco.

[11] Saran, H. and V. V. Vazirani (1995). "Finding $ k $ Cuts within Twice the Optimal." SIAM Journal on Computing 24: 101.

[12] Feige, U., R. Krauthgamer, et al. (2000). Approximating the minimum bisection size, ACM.

[13] Papadimitriou, C. H. and M. Sideri (1996). "The bisection width of grid graphs." Theory of Computing Systems 29(2): 97-110.

[14] Garey, M. R., D. S. Johnson, et al. (1976). "Some simplified NP-complete graph problems." Theoretical computer science 1(3): 237-267.

[15] Bui, T. N., S. Chaudhuri, et al. (1987). "Graph bisection algorithms with good average case behavior." Combinatorica 7(2): 171-191.

[16] Barnes, E. R., A. Vannelli, et al. (1988). "A new heuristic for partitioning the nodes of a graph." SIAM Journal on Discrete Mathematics 1: 299.

[17] Tu, C. C. (1998). "Spectral methods for graph bisection problems." Computers & operations research 25(7-8): 519-530.

[18] Kernighan, B. W. and S. Lin (1970). "An efficient heuristic procedure for partitioning graphs." Bell System Technical Journal 49(2): 291-307.

[19] Tovey, C. A. (1985). "Hill climbing with multiple local optima." SIAM journal on algebraic and discrete methods 6: 384.

[20] Preas, B. T., M. J. Lorenzetti, et al. (1988). Physical design automation of VLSI systems, Benjamin-Cummings Pub Co.

[21] Marks, J., W. Ruml, et al. (1998). "A seed-growth heuristic for graph bisection." Proceedings of Algorithms and Experiments (ALEX98). Italy: Trento: 76-87.

[22] Wang, R. L., Y. Yamanishi, et al. (2007). "A New Neuron Dynamics for Solving the Minimum Graph Bisection Problem." International Journal of Computer Science and Network Security 7(3): 55-58.

[23] Chen, Z. Q., R. L. Wang, et al. (2008). "An efficient genetic algorithm based approach for the minimum graph bisection problem." Int'l Journal of Computer Science and Network Security 8(6): 118-124.

[24] Atashpaz-Gargari, E. and C. Lucas (2007). Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, IEEE.

[25] Atashpaz-Gargari, E (2008) E xpanding the Social Optimization Algorithm and Evaluating its Efficiency: M.S. Thesis in Computer and Electerical Engineeing, Tehran University.