B-ACO for the Irregularity Sum Problems

Ay-Hwa Andy Liou[1] Tzong-Heng Chi[2] Kun-Yi Lin[3]

[1]Department of Information Management, [3]Department of Computer Science and Information Engineering, Tamkang University, Taiwan

[2]Department of Information Management, Aletheia University, Taiwan

Abstract

This paper proposes a B-ACO algorithm to construct valid irregularity assignments under the minimal irregularity sum for graphs. The proposed algorithm is ACO-based but generalized with binary encoding. The implementation is compared with the exhaustive search and random walk to examine the algorithm's efficiency and effectiveness. Finally, the preliminary results obtained on computational tests are provided.

Keywords: ACO, Graph Theory, Irregularity Sum, Irregularity Strength.

1. Introduction

1.1. The Problems

The results have been extensive and abundant in studying of regular graphs which can be dated back to Petersen [**Error! Reference source not found.**][**Error! Reference source not found.**]. On the contrary, the irregular properties of graphs were not investigated till Alavi et al. [**Error! Reference source not found.**]. For a simple graph G, it is regular if its vertices have the same degree. A network is a simple graph to which each edge is assigned a positive integer value or weight. The degree of a vertex in a network is the sum of the weights of its incident edges. A network is local or highly irregular if it is connected and the vertices in the neighborhood of any vertex have mutually distinct degrees. Otherwise, a network is global irregular or just irregular if all the vertices have distinct degrees [**Error! Reference source not found.**][**Error! Reference source not found.**][**Error! Reference source not found.**]. For digraphs, the similar definitions can refer to [**Error! Reference source not found.**]. Given an assignment which is a function $w:E(G) \rightarrow \{1, 2, 3, \ldots, s\}$, the weight of a vertex v is $w(v)= \Sigma_{v\in e}w(e)$. The strength of a network is the maximum weight assigned to any edge. Also, an irregular assignment is valid or admissible if the weight of each vertex is different and thus the graph G is distinguishable. The irregularity strength $s(G)$ of graph G is the minimum strength among irregular networks with underlying graph G. The study of irregular strength stems from problems related to highly irregular graphs and multigraphs first introduced by Chartrand et al. [**Error! Reference source not found.**]. The irregularity sum of a graph G is the minimum sum of the induced weights of all vertices in an irregular assignment for G originally considered by [**Error! Reference source not found.**]. We abbreviate the problem of finding the minimal irregularity sum as ISP, irregularity sum problems.

Problems important to irregular assignments includes those to determine $s(G)$, the irregularity sum, the number of irregular assignments, the maximum and minimum size for the n-vertex graph G, etc. [**Error! Reference source not found.**][**Error! Reference source not found.**]. The problem of studying $s(G)$ was proved by Chartrand et al. to be rather hard, even for very simple graphs [**Error! Reference source not found.**][**Error! Reference source not found.**].

Besides, the study of irregular assignments also has been stretched to hypergraph theory, integer matrix designs, finite projective geometry, irregular coloring, asymmetric graphs, etc. [**Error! Reference source not found.**][**Error! Reference source not found.**][**Error! Reference source not found.**]. Most of mentioned researches have done well to prove the existence of lower or upper bound of $s(G)$. However, there was less study in the construction of valid assignments under the consideration of irregularity sum. Though [**Error! Reference source not found.**][**Error! Reference source not found.**] has done many good results, they focused on trees only. For these reasons, in this paper, we shift our attention to the construction of solutions in finding the irregular assignments related to the irregular strength and irregular sum for graphs. That is, we have developed a general ACO-based probability algorithm called B-ACO to construct valid assignments and find the minimal irregularity sum for graphs.

1.2. The Ant Algorithm

The algorithm used in this paper is based on the paradigm of Ant colony optimization (ACO). The ACO was originally introduced by M. Dorigo [**Error! Reference source not found.**] as a metaheuristic that uses artificial ants to find (near-) optimal solutions for the well-known traveling salesman problem (TSP) [**Error! Reference source not found.**]. It also has proved successful to cope with many static and dynamic combinatorial optimization problems such as the quadratic assignment, the vehicle routing, the network routing, the single machine tardiness, the power economic dispatch, the graph coloring, etc. [**Error! Reference source not found.**] [**Error! Reference source not found.**].

The behavior of artificial ants is based on the traits of real ants in food foraging with additional capabilities such as a memory of past action to make them more effective. Each ant of the colony builds a solution to the problem under consideration, and uses information collected on the problem characteristics and its own performance to change how other ants see the problem. This indirect communication mediated by pheromone secretion called stigmergy is one of the two important ACO's characteristics. While the random event happened such as an ant getting lost but finding a new food source, the fluctuation can possibly be amplified by recruitment of other ants, the positive feedback. Negative feedback induced by the depletion of food source is usually an answer to the limitation on behavior of exploitation. In ACO, the mechanism is implemented by the evaporation of pheromone. Shortly, the second characteristic is the results of self-organization, which are global in nature, but come about from interactions based only on local information [**Error! Reference source not found.**].

Searching is usually used to find the solution for problems without complete analytical solution. There are two ways to do, i.e., by exhaustive (only effective for small problems) or by heuristic. Both of them are investigated in this paper to find the (near-)optimal irregularity strength and sum for sample graphs.

This paper is structured as follows. In section 2 we introduce the associated mathematical notation. The previous research results are introduced here, too. Section 3 provides the details of the algorithm. In Section 4, the details of the experiment are given and we present the computational results on different test sets. Finally, In Section 5 we draw the conclusions from our work.

2. Mathematic Notations and Previous Results

A graph G=(V, E) is a simple graph if it has no loops and multiple edges, where V is the vertex set and E is the edge set of G separately. An edge is called a loop if both of its endpoints are the same vertex. A graph is said to have multiple edges if there is a pair of vertices with more than one edge joining them. This graph is called multigraph. The number of V is called order represented as $|V|$ and the number of E is called size as $|E|$. The pendant vertex is one which has degree one and the edge adjacent to it is called pendant edge. A tree T_n on n vertices is a connected graph with no cycles. A complete graph K_n on n vertices is a graph with all possible edges; every vertex is joined to every other vertex by a single edge. A graph G is bipartite if for vertex classes V_1 and V_2, $V(G)=V_1\cup V_2$ and $V_1\cap V_2=\phi$ and each edge of a vertex of V_1 is joined to a vertex of V_2. The graph is a complete bipartite graph if, in addition to being bipartite, it contains all possible edges from V_1 to V_2. The number of edges incident with any vertex v is said to be the degree of v. A graph in which all vertices have the same degree r is an r-regular graph. Unless stated otherwise, graphs mentioned in this paper are simple and undirected.

The study of irregularity strength was inspired by one of the oldest fact in graph theory that, in a simple graph, at least two vertices must have the same degree. This fact no longer holds for multigraphs. A question proposed in [**Error! Reference source not found.**] to give another view to this problem: What is the least number of edges needed to be added to a graph in order to convert a simple graph into an irregular multigraph? To answer this question is via an assignment of integer weights to the edge of graph.

Very few types of graphs have their exact irregularity strengths known. Most of researches focused on the lower or upper bounds have been proposed for limited family of graphs such as paths, complete graphs, cycles, most bipartite graphs, Turan graphs, wheel, k-cubes, grids, etc. [**Error! Reference source not found.**], which are useful both in assessing the quality of approximate solutions and in limiting the search in the solution space for optimal assignments. The number of irregular assignments can also be useful for estimating the reasonable hit ratio in finding $s(G)$.

The following results most quoted in [**Error! Reference source not found.**] had been reported:

- For any graph, the strength $s(G) \geq 2$ [**Error! Reference source not found.**].
- For any connected graph with at least three vertices, $s(G) \leq 2n - 3$ [**Error! Reference source not found.**]. It was strengthened by [**Error! Reference source not found.**] as $s(G) \leq n + 1$ for any graph with $s(G) < \infty$. For any connected graph on $n \geq 4$, $s(G) \leq n- 1$.
- The strength of a path of length n is $n/2$.

- The irregularity strength of a full d-ary tree (d=2, 3) is its number of pendant vertices.
- For any tree T, both n_1, if $n_1 \geq n_2$, and $\lceil (n_1 + n_2)/2 \rceil$, are lower bound for the irregularity strength where n_i denotes the number of vertices of degree i. However, this was an underestimate and refuted in [**Error! Reference source not found.**] by a counter evidence when vertices are infinite.
- The strength of a complete graph of order $n \geq 3$ is 3.
- For complete bipartite graphs $k_{m,n}$, $s(k_{m,n})$ is determined in [**Error! Reference source not found.**] except for the case when $m = n = 2k + 1$. $s(k_{3,3}) = 4$ for m odd was proved in [**Error! Reference source not found.**].
- The strength of 2-regular graphs of order $n \geq 3$ is $[n/2] + c$, c= 0, 1, 2.

  For an r-regular graph G of order n, $r \geq 2$, $n \geq 3$, $s(G) \leq [\ n/2\ ] + 9$, where $s(G) \leq [\ n/r\ ] + c$ was conjectured by Faudree and Lehel [**Error! Reference source not found.**]. It has been verified when $r \geq n/2$ and, $r \geq n/3$ when the graph is 2-connected. Moreover, $s(G) \leq c_1(\ n/r\ )$ when $r \leq \sqrt{n}$, and $s(G) \leq c_2(\ n\log n\ )/r$ when $r \geq n$ [**Error! Reference source not found.**].

  For an (n-3)-regular graph G of order $n \geq 5$, $s(G) = 3$, except when $G = k_{3,3}$ and $s(k_{3,3}) = 4$.

  For an (n-4)-regular graph G of order $n \geq 8$, $s(G) = 3$.

3. B-ACO

As mentioned, application of an ACO algorithm to a combinatorial optimization problem requires giving the definition of a constructive algorithm. Accordingly, we have designed an ant algorithm called B-ACO algorithm in which a set of artificial ants builds feasible solutions to the ISP.

B-ACO works based on the encoding of a given problem instance as a constructive graph, usually, an undirected complete graph. For the sake of efficiency in both memory and performance, and also the convenience in programming, we replace it with a special 0-1-graph C, called zipper graph here, such as Figure 1. The stepwise construction of a solution is presented by a model-biased random walk in C. Since the solution space is usually large, we apply the rule: for any connected graph on $n \geq 4$, $s(G) \leq n + 1$, as the upper bound for the encoding. That is, for a graph on order n, we calculate the minimal bits B needed for representing integer n+1. Hence the length L of the zipper is B*n. The binary encoding approach is important to B-ACO and that is where the name B-ACO comes from.

Informally, B-ACO works as follows. Each ant iteratively starts from node b_{00} or b_{01} and adds new nodes until to the end node b_{L0} or b_{L1}. When in node b_{i*}, * means don't care, possibly 0 or 1, an ant applies transition rule, that is, it probabilistically chooses the next node j, i.e. $b_{(i+1)*}$, based on the pheromone trail and additional heuristics. For most of ACO algorithms, the size of next candidate node set $|\ j\ |$ is usually large and dynamically changed with time. This is often troublesome for the algorithm designing and programming. Especially, it consumes lots of running time in searching and maintaining necessary data structures. By the zipper, our ants only consider $b_{(i+1)0}$ and $b_{(i+1)1}$ as the next hop. Obviously, it is easier.

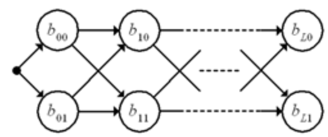


Figure 1

The zipper graph is as the basis of encoding and as the construction graph ants walk on.

The ants choose, with probability q, the node j, $j \in \{\ b_{(i+1)0}, b_{(i+1)1}\}$, for which the product $\tau_{ij} \cdot \eta_{i+1}$ is the highest, with the probability 1- q the node j is chosen with the probability given by

$$\underset{j \in N(i)}{p_{ji}} = \frac{\tau_{ij} \cdot \eta_{i+1}}{\sum_{k \in N(i)} \tau_{ik} \cdot \eta_{i+1}} \tag{1}$$

where $N(i)= \{b_{(i+1)0}, b_{(i+1)1}\}$, τ_{ij} is the pheromone on the arc from the node i to j, and η_{i+1} is the weight of the position in B of the next node j. The value q represents the probabilistic transition rule to follow. Besides, the reason to take the bit position as the heuristic information, i.e. η_{i+1}, is for speeding up the process of converging to the smaller arc value. Though [**Error! Reference source not found.**] mentioned that the greedy algorithms works well for finding the minimal irregularity strength, it is not true for irregularity sum problems. We give a counterexample as Figure 2.

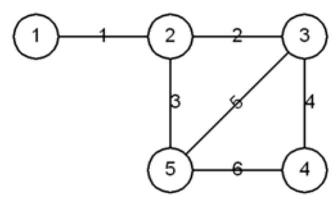


Figure 2

It is a connected undirected graph with cycles. The irregularity assignment is 2, 1, 2, 1, 2, 2 for edges 1, 2, 3, 4, 5, 6, respectively, and therefore 2, 5, 4, 3, 6 for nodes 1 to 5. By this assignment, the irregularity strength is 2 and the irregularity sum is 20. However, for another valid assignment, 1, 1, 2, 1, 3, 1 for edges 1 to 6 and 1, 4, 5, 2, 6 for nodes 1 to 5, the irregularity strength is 3 but the irregularity sum is 18. The strength of the former assignment is better but the sum of the latter one is better. In short, though the vertex label is summed with the incident edges, the pursuit of the minimal sum is conflicting with the pursuit of the minimal strength. Therefore, information about strength should be used carefully or the algorithm will be trapped in an inappropriate state.

When an ant completes a tour, it is constructed a candidate as a feasible solution. Then, the algorithm will verify it to see whether it is a valid walk or not. That is, vertices' labels formed by the assignment are distinguishable. If it is valid, then the strength and sum are calculated as the basis of pheromone value to deposit. That is, the trace can be more visible as a memory storing better step (b_{i*}, $b_{(i+1)*}$) than it has been in previous runs. Also, this strategy to deposit pheromone is called local and online-delayed because it is using only local information and not to deposit till the tour is finished. The pheromone update rule is as follows.

$$\tau_{ij}(t) = \tau_{ij}(t) + \Delta\tau^k(t) \quad \textbf{(2)}.$$

where $\Delta\tau^k$ is the pheromone value to deposit for ant k.

Finally, evaporation decreases the pheromone trails when any ant deposits pheromone but not interleaving as usual ACO algorithms. B-ACO applies the following rule:

$$\tau_{ij} = (1-\rho)\cdot \tau_{ij} \quad \textbf{(3)},$$

where $\rho \in (0, 1)$ is a parameter or pheromone evaporation. Therefore, the complete cycle of an iteration of B-ACO involves ants' movement, solution evaluation, pheromone deposit, and pheromone evaporation. The algorithm is as shown in Figure 3.

B-ACO algorithm

1. (Initialization)
   Compute the upper bound to define the coding length
   Encode the problem to zipper construction graph
2. (Construction)
   For each ant k do
   Repeat
   Toil a coin, c,
   If c< q, choose the next node with highest pheromone trail
   Else, choose the node to move with probability given by (1)
   Until ant k completes its solution
   Update the pheromone with online-delayed rule by (2)
   End For
   Update the pheromone with off-line rule (3)
3. (Terminating condition)
   If not(end_stop) **goto** step 2.

Figure 3

4. Experiments with B-ACO

Our experiments were aimed at (1) finding the best parameters for the B-ACO in solving ISP and (2) comparing B-ACO with exhaustive search if possible, and random walk. Experiments were run on a PC with Pentium 2.4 GHz CPU and 512MB main memory. The code was written in C++. For the test problems, we first examined problems with known results as mentioned above including paths, trees, and complete graphs. Also, we have tested randomly generated graphs with designated order and density. Most of them are conformed to the previous research results especially for paths, trees, and complete graphs. We found the ratio of the valid assignments to the solution space is usually tiny. Take the Figure 2 as sample, it needs 3 bits because the order is 5. Since there are 6 edges, the solution space is 2^{18}=262144. It is huge and computationally hard. We examine the enumeration, only 63100 valid assignments were found. Hence the ratio is 0.24. Comparatively, for 10000 ant steps, the average number of valid assignment is almost 9500. The ratio is near 95%. That is, the B-ACO is efficient. In most cases, B-ACO is also effective to find the optimal or near-optimal solutions.

The Executions of B-ACO with different number of ants are also examined and confirmed the tendency of better results with increasing ants.

Finally, the B-ACO is compared with random walking of ants with no pheromone deposition to understand its effectiveness. With 13 ants doing 10000 runs each time, the results of 100 times experiments on Figure 4 is presented in Figure 5 and 6.

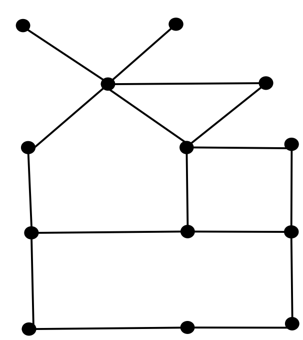
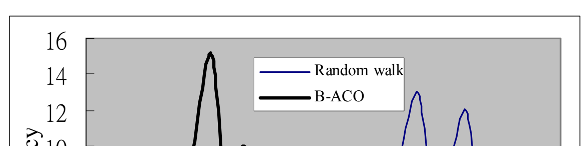


Figure 4



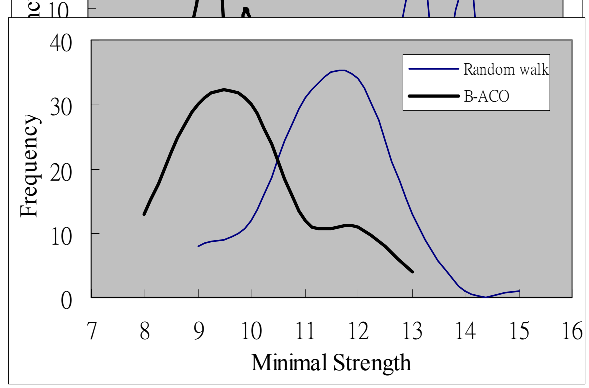


Figure 5

Figure 6

These results shown the frequencies of minimal sum and minimal strengths found during the experiment. B-ACO is apparently showing its superiority.

5. Conclusion

By encoding the problem to the zipper graph, we have shown how the B-ACO algorithm finds the minimal irregularity sum and the detail irregular assignments. By the comparisons with the exhaustive search and usual ACO algorithms, the proposed B-ACO has shown its convenience in algorithm design and programming, efficiency in running time steps, and effectiveness in optimal solution finding. We have briefly addressed the hardness in solving irregularity sum problems. In fact, it needs more rigid proof. However, by our experiments, it provides some useful insights. Moreover, the counterexample was given to present the conflict between finding minimal irregularity strength and irregularity sum. Finally, for the future work, more experiments and test problems need to be done and examined to make B-ACO encouraging.

6. References

[1] M. Aiger and E. Triesch, "Irregular assignments of trees and forests," *SIAM Journal of Discrete Mathematics*, vol. 3, pp. 439-449, 1990.

[2] Y. Alavi, G. Chartrand, F. R. K. Chung, P. Erdos, R. L. Graham and O. R. Oellermann, "Highly irregular graphs," *J. Graph Theory*, vol. 11, pp. 235-249, 1987.

[3] D. Amar, "Irregularity strength of regular graphs of large degree," *Discrete* Mathematics, vol. 114, pp. 9-17, 1993.

[4] T. Bohman and D. Kravitz, "On the irregularity strength of trees," *Journal of Graph Theory*, vol. 45, pp. 241-254, 2004.

[5] A. C. Burris, "The irregular coloring number of trees, *Discrete Mathematics*," vol. 141, pp. 279-283, 1995.

[6] G. Chartrand, M. S. Jacobsen, J. Lehel, O. R. Oellermann, S. Ruiz, and F. Saba, "Irregular Networks," *Congressus Numerantium*, vol. 64, pp. 187-192, 1988.

[7] P. R. Christopher, "Highly irregular asymmetric graphs," *Congressus Numerantium*, vol. 91, pp. 93-97, 1992.

[8] M. Dorigo and Gianni Di Caro, "The Ant Conlony Optimization Meta-Heuristic," D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pp. 11-32, 1999.

[9] M. Dorigo and T. Stützle, *Ant* Colony *Optimization*, MIT Press, 2004.

[10] G. Ebert, J. Hemmeter, F. Lazebnik, and A. Woldar, "On the number of irregular assignments on a graph," *Discrete Mathematics*, vol. 93 , pp. 131-142, 1991.

[11] R. Faudree, M. Jacobson, J. Lehel, and R. Schlp, "Irregular networks, regular graphs and integer matrices with distinct row and column sums," *Discrete Mathematics*, vol. 76, pp. 223-240, 1989.

[12] A. Frieze, R. J. Gould, M. Karoński, and F. Pfender, "On graph irregularity strength," *Journal of Graph Theory*, vol. 41, pp. 120-137, 2002.

[13] A. Gyarfas, "The irregularity strength of $K_{m,m}$ is 4 for m odd," *Discrete Mathematics*, vol. 71, pp. 273-274, 1988.

[14] M. S. Jacobson, E. Kubicka, and G. Kubicki, "Irregularity Sum for graphs," *Vishwa International Journal of Graph Theory*, vol. 1, no. 2, pp. 159-175, 1992.

[15] D. Kravitz, "Investigation of the Irregularity Strength of Trees," Undergraduate thesis, University of Delaware, 2000.

[16] J. Lehel, "Facts and quests on degree irregular assignments, Graph theory," *Combinatorics, and Applications*, Vol. 2 (Kalamazoo, MI, 1988), Wiley-Intersci. Publ., pp. 765--781, 1991.

[17] Z. Majcher, J. Michael, Joanna Górska, and Z. Skupień, "The minimum size of fully irregular oriented graphs," *Discrete Mathematics*, vol. 236, 263-272, 2001.

[18] J. Petersen, "Die Theorie der regulären Graphen, " *Acta Math.*, vol. 15, pp. 193-220, 1891.

[19] T. Stützle and M. Dorigo, "ACO algorithms for the traveling salesman problem", In K. Miettinen, M. Makela, P. Neittaanmaki, J. Periaux, editors, *Evolutionary Algorithms in Engineering and Computer Science*, Wiley, 1999.

[20] P. Tarasewich and P. McMullen, "Swarm intelligence power in numbers", *Communications of the ACM*, vol. 45, issue 8, pp. 62-67, August 2000.