Assortative partitions of complete graphs

Magdalena Zlatanova
Faculty of Information Sciences and Computer Engineering, Skopje
Ss. Cyril and Methodius University
Republic of North Macedonia
magdalena.zlatanova@students.finki.ukim.mk

***Abstract*—This paper studies the problem of assortative partitions of complete graphs. Assortativity is a measure of the similarity of each node to its neighborhood. The results from numerical simulations suggest that for this class of graphs the assortative partitioning problem becomes more difficult as we increase the assortativity threshold. We observe a significant difference in the performance of the Gradient Descent algorithm when our assortativity threshold is set to 4 instead of 2. This numerically supports the hypothesis that the problem becomes more difficult.**

***Index Terms*—combinatorial optimization, NP class, complete graphs, graph partitioning, statistical physics.**

I. Introduction

Let us have a graph and let each of its nodes belong to one of two types. We want each node to be connected to as many nodes of its own kind as possible, thus we want to minimize edges whose ends are of different kinds. The assortativity of the graph is a measure of the ”happiness” of all nodes. If each node is located in a neighborhood where most nodes are of its own kind, then we will say that the node is happy (the terms friendly and assortative are also found in the literature). The problem becomes trivial if we let all nodes be of the same kind. We introduce the condition that we want the two node partitions to be roughly of the same cardinality. This condition is also called zero magnetization in statistical physics. For the problem defined in this way, we introduce another parameter, the assortativity threshold H. To consider a node happy, we need at least half of its neighbors to be of the same kind as the node we are considering. The threshold of assortativity indicates how much above the half we need to consider the node happy. Example: we have a node with degree 8, and assortativity threshold H=2, then we require this node to have at least 6 nodes like it in its neighborhood to consider it happy. The work in the project will consist of an analysis of the value of this parameter and the complexity of finding a favorable configuration.

II. Related work

For H=0, equivalent to requiring only half of the neighbors to be of the same kind, we have a problem for which the literature [4] indicates that it is easy to find a satisfactory configuration. For H¿0, the problem in some types of graphs exhibits the Overlap Gap property. A detailed interpretation of the Overlap Gap Property can be found in [2]. This property is an obstacle in many optimization problems and is closely related to the NP-complexity of algorithms. The Overlap Gap property refers to a large distance between two solutions that are close to optimal. We want to check whether it is reasonable to expect the Overlap Gap Property and thus high complexity in given classes of graphs that are not sufficiently explored. In a paper [10] from this area, the author investigates the co-evolutionary process based on the Bak-Snappen model, where very high degrees of adaptability are achieved for all kinds. The purpose is to see if a simple model for the co-evolutionary process of species can serve as a basis for a useful technique in Artificial Intelligence. In this paper the authors observed that on geometric graphs the new approach proposed is much faster in reaching a good solution than Simulated Annealing. In Simulated Annealing [1] we have much larger jumps in the beginning and it takes many more iterations to reach a low value of the loss function. This means that the author's two initial hypotheses are correct, but only for the specific problems investigated in the paper and only for geometric graphs. However, it should be noted that not all NP-hard problems will be so prone to Extremal Optimization and not all types of graphs will show similar results. Although the min-bisection problem is mentioned in this paper, it is not the primary focus of the work. However, there are quite a few useful guidelines that can be applied to our specific problem. As an example, the possibility of overcoming large energy barriers (local minima in our local search problem) is quite promising for the different types of graphs that we will explore. Furthermore, the way the numerical simulations are conducted has parallels with our approach.

III. Of practical meaning

The assortativity problem is fundamentally related to the minimum-bisection problem. Its complexity is of particular importance for many practical applications: electrical circuit design, Hopfield networks [7] and partitioning of sparse matrices. Because there are already studies on regular or valence graphs in the literature [3], we want to extend these analyzes to complete graphs. For this, we will use the algorithm based on Gradient Descent and inspired by Extremal Optimization [5].

IV. Hypothesis

Guided by the existing literature, it is intuitive to assume that the problem is indeed hard, and it gets harder as we impose more stringent conditions. However, we divide this assumption into two parts, which we can test experimentally.

A. Overlap Gap Property

Based on the stability analysis of the minimum bisection problem, we assume that the Overlap Gap Property also applies to graphs of a different nature than regular. Overlap Gap Property in our case means that when we find a favorable configuration, if we change only one node, it will start a chain of consequent changes and many other nodes will have to swap partition before we reach a favorable configuration again. If it is true for valence graphs, given that all (and even valence) graphs can be represented in a complete graph with appropriately chosen weights, we assume it will also be true for complete graphs.

B. Clear demarcation of phases

We assume that Gradient Descent on all complete graphs will encounter obstacles in finding a favorable configuration. We expect to clearly differentiate stages of algorithmic complexity for different values of the assortative threshold.

V. Methodology

The question explored in this paper is based on several scientific theories. The hypothesis is set so research can be conducted in two ways: through formal mathematical proofs or through numerical simulations. Numerical simulations alone are not enough to prove or disprove the hypothesis, but because we will work with a large amount of data, the results will at least be meaningful. The data (graphs) will be generated by random generators. For each graph we will pass a specific seed so that we are able to generate the same graph in the future. It follows that we will apply quantitative methodology and we will experimentally test our hypothesis.

VI. Gradient Descent

In the general case, the Gradient Descent algorithm is used when we want to optimize something, usually by looking for the global minimum of some loss function. In our case, the algorithm works a little differently. For a graph that is not initially partitioned, its nodes are randomly divided into two partitions. Of course, these partitions are probably not very assortative. However, it is always possible for the nodes to have high assortativity (half of their degree). Gradient Descent is simply a loop that doesn't end until this is achieved. Additionally, we repeat the procedure as many times as we define restarts. Finally, we select the best value from all restarts for the given graph.

VII. Experiments

For an under-explored class of graphs (complete graphs), we do the following:

- Generating graphs of different sizes (from 32 nodes to 16,777 nodes). 100 different graphs are generated for each size to reduce the probability of random results;
- Setting the threshold for assortativity;
- Running Gradient Descent, which reliably finds an assortative solution (assortativity of at least 1 for each node). This algorithm works with multiple restarts of the same graph, which means that the same graph gets a different initial configuration. This algorithm was made parallel so that we have 10,000 restarts for each instance of the graph;
- Collecting results, pre-processing of data and presenting trends;

A similar approach (but a different class of graphs and different algorithms) can be found in paper [3].

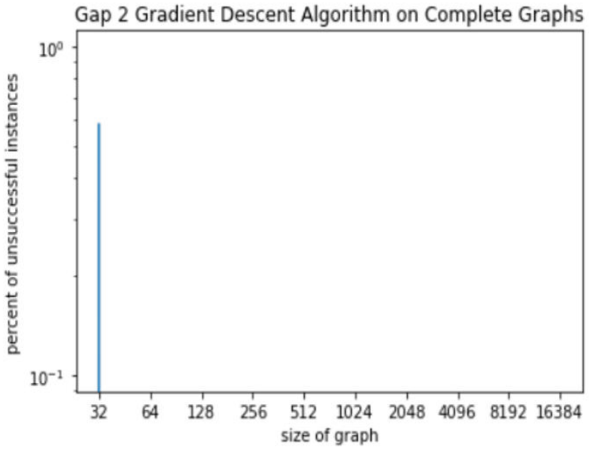

Fig. 1. Percentage of graph instances for which there is not a single successful restart. A successful restart is one where all nodes have an assortativity of at least 2. Failure is concentrated in graphs of size 32.

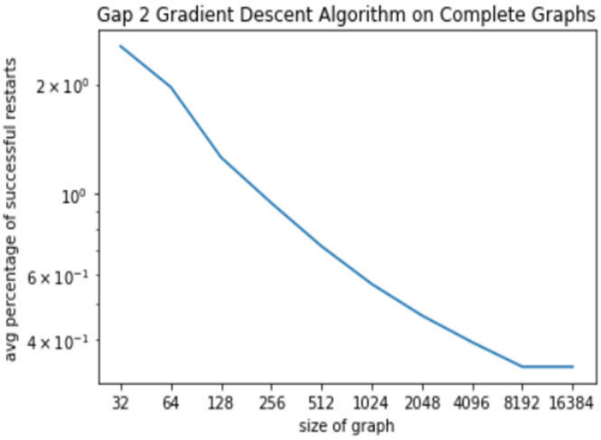

Fig. 2. How many successful restarts we have on average (the percentage) for all instances of a graph of a given size. A successful restart is one where all nodes have an assortativity of at least 2. We observe an exponential decay.

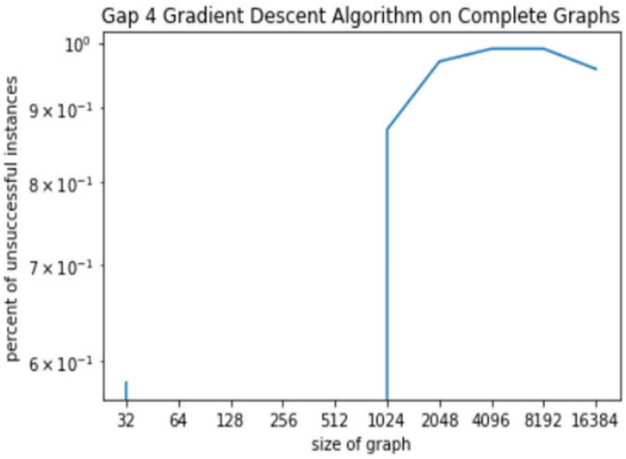

Fig. 3. Percentage of graph instances for which there is not a single successful restart. A successful restart is one where all nodes have an assortability of at least 4.

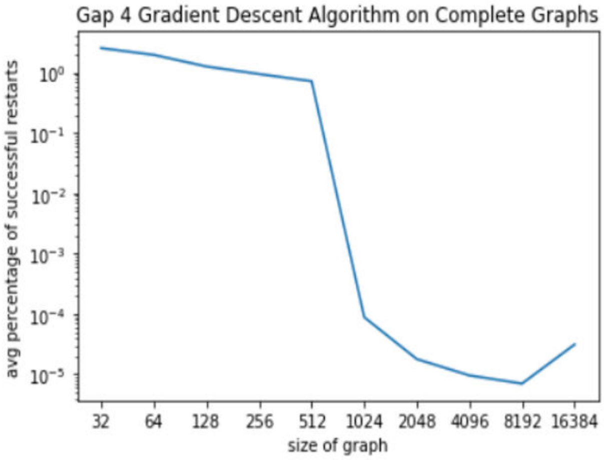

Fig. 4. How many successful restarts we have on average (the percentage) for all instances of a graph of a given size. A successful restart is one where all nodes have an assortability of at least 4.

VIII. Results

In a complete graph, every node is connected to all other nodes. Because our goal is to assign all nodes to a partition so that they are assortative (located in a good neighborhood), to prevent the algorithms from assigning all nodes to one partition we will also introduce edge weights. Our problem now takes on a slightly different definition, although in fact the question, the experiments, and the conclusion are completely equivalent. Every edge is assigned weight of +1 or -1 with equal probability. If an edge has a negative weight, then the two nodes it connects want to belong to a different partition. Accordingly, if the edge is of positive weight, just as in the previous problem definition, the nodes want to be in the same partition. Here it is important to emphasize that we want to have roughly equal numbers of negative and positive weight edges. Since there are already papers in the literature indicating that it is easy to find a solution for a given assortative threshold for given classes of graphs, we now need to set an assortative threshold for complete graphs. There is a difference here for graphs with even and odd number of nodes. For example, if we want half of the neighborhood to agree with our node, then if we have an even graph, the number of edges we will consider is odd, so the half should be defined accordingly. Because we will be working with graph sizes that are powers of 2, all graphs are even-sized. For each node and the corresponding neighbor, we check whether they are in a happy connection or not, and sum the values. For example, if we have a weight on the edge of -1 and both nodes are in a different partition, we add 1 to the assortativity of the node we are looking at, and if they are in the same partition we subtract 1. In even-sized graphs, we will never get an assortativity of 0-it will always be odd. So any good assortativity is of positive value. Additionally, an assortativity of 1 is similar to an assortativity threshold of 0 in valence graphs. That is why our question of interest is whether assortativity of more than 2 can be achieved. Because we want all nodes to have good assortativity, we consider a successful restart of Gradient Descent to be the one in which the worst node has assortativity at least 3. To analyze the data, we look at the percentage of restarts that were successful and the number of graphs for which the algorithm fails to find a satisfactory configuration in all restarts. Figure 4 shows on average how many of the restarts are successful for the parameters defined in this way (assortativity threshold 4). Because we are working with 100 graph instances, it is unlikely that the results are a product of chance. This distribution resembles an exponential, indicating that indeed the problem becomes more difficult for large graphs. On the other hand, if we relax the assortativity condition to 2, we still keep an exponential distribution (Figure 2). So, the problem becomes more difficult for both thresholds, but from the comparison in Figure 5, it can still be seen that the problem with a higher threshold is more difficult

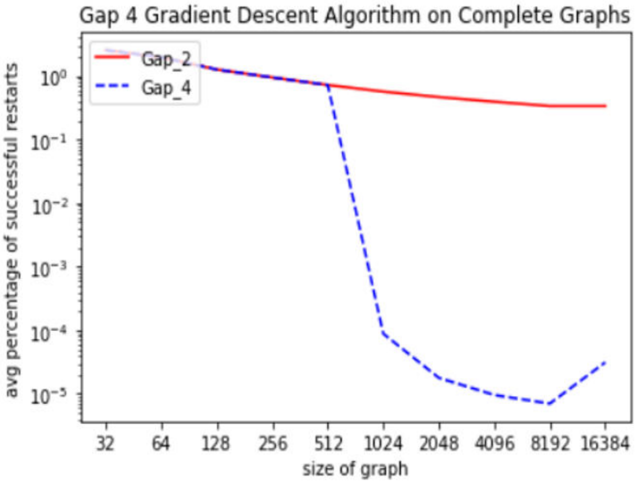

Fig. 5. Comparison of Gradient Descent performance when we change the assortativity threshold. It can be seen that for threshold 4 the performance of the algorithm drops significantly.

and Gradient Descent performs much worse. This supports our hypothesis of differentiating phases of complexity. From Figure 1 we can see that Gradient Descent still manages to find a solution (at least one successful restart) for almost all graph instances when the threshold is set to 2. The exception is graphs of size 32, but this is expected because the graph is too small to reach the thresholds of assortativity that we set. However, the picture is quite different for the threshold of 4, where Gradient Descent often fails completely for graphs with size above 1024. Accordingly, we see a sharp increase in the percentage of graphs for which Gradient Descent fails around graph size 1024 (Figure 3).

IX. Discussion and conclusion

After the experiments, we can see that as the graph increases, we have less and less successful restarts with the Gradient Descent algorithm if we define success as all nodes with assortativity greater than to 2. This provides a numerical basis to assume that our initial hypothesis is correct and that indeed the min-bisection problem for high assortativity thresholds is hard and exhibits the Overlap Gap property. Further experiments with different algorithms are necessary to strengthen this hypothesis, and the proof would undoubtedly be theoretical, but this paper focused on the minimum-bisection problem in complete graphs and shows experimentally that it is reasonable to expect Overlap Gap Property and algorithmic hardness in this problem.

X. Acknowledgments

This paper is the result of a three-month research internship at the Lab of Statistical Physics and Computation at École Polytechnique Fédérale de Lausanne. The author is thankful for the mentorship and guidance of Prof. Lenka Zdeborova and Freya Behrens, as well as the problem idea and additional help from Prof. Stefan Boettcher.

References

[1] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon, ”Optimization by Simulated Annealing: An Experimental Evaluation; Part I Graph Partitioning,” Oper. Res., vol. 37, no. 6, pp. 865-892, Dec. 1989.

[2] D. Gamarnik, A. Jagannath, and S. Sen, ”The overlap gap property in principal submatrix recovery,” Probab. Theory Relat. Fields, vol. 181, pp. 757–814, Sep. 2021.

[3] F. Behrens, G. Arpino, Y. Kivva, L. Zdeborová, ”(Dis)assortative partitions on random regular graphs,” J. Phys. A: Math. Theor., vol. 55, no. 39, p. 395004, Sep. 2022

[4] M. Palassini, ”Ground-state energy fluctuations in the Sherrington–Kirkpatrick model,” J. Stat. Mech.: Theory Exp., vol. 2008, no. 10, p. P10005, Oct. 2008.

[5] S. Boettcher, ”Extremal optimization of graph partitioning at the percolation threshold,” J. Phys. A: Math. Gen., vol. 32, no. 28, pp. 5201–5211, Jul. 1999.

[6] C. Bazgan, Z. Tuza, and D. Vanderpooten, ”Satisfactory graph partition, variants, and generalizations,” Eur. J. Oper. Res., vol. 206, no. 2, pp. 271–280, Oct. 2010.

[7] B. Bollobás, Random Graphs, 2nd ed. Cambridge, England: Cambridge Univ. Press, 1985.

[8] D. J. Watts and S. H. Strogatz, ”Collective dynamics of ’small-world’ networks,” Nature, vol. 393, no. 6684, pp. 440–442, Jun. 1998.

[9] S. Boettcher, ”Extremal optimization: heuristics via coevolutionary avalanches,” Comput. Sci. Eng., vol. 2, no. 6, pp. 75–82, Nov. 2000.