M - 22

APPLICATING CVD ALGORITHM ON EDGE-COLORING OF SPECIAL GRAPHS

Nur Insani

Deparment of Mathematics Education, Yogyakarta State University

Abstract

Heuristics algorithm is a soultion method that typically relatively quick to find a feasibel soloution with reasonable time and quality though there are no guarantees about if the quality of the solution is bad. This research explores the application of Conflicting Vertex Displacement (CVD) algorithm on edge-coloring of special graphs. This algorithm found by Fiol and Vilaltella [2] 6in 2012 and uses the idea of recolor of two “conflicts” edges (edges that are incident to a vertex) along the paths of adjacent vertices. The research tests the algorithm on special graphs, ie. bipartite graphs.

Keywords: edge-coloring, heuristics, Kempe chain, bipartite graphs.

INTRODUCTION

Graph coloring is one of the greatest topics on the graph problem. Though many people do not consider it as an important problem, but in fact it is a very important ones. Edge coloring is a problem of coloing the edges of a graph with a minimum number of colors as possible. As it is known to be $NP-Complete$ problem (even for maximum degree $\Delta = 3$), many scientists has find ways to resolve this problem either with exact algorithms or heuristics ones. Heuristics algorithm is a method designed to solve problems that are not emphasized in proving whether the solution obtained is correct, but produce a quick solution. Hopcroft and Karp [6] tried to solve the edge coloring problem for bipartite graphs by using a heuristics algorithm, while Marathe and Panconeso [11] tried to introduce another simple and distributed edge coloring algorithm in 2000. Nemhauser ad Park [13] intoduced a plyhedral approach to edge coloring problem and this approach solve the problem heuristically.

In 2012, Fiol and Vilatella [2] has constructed a simple heuristics algorithm called CVD (Conflicting Vertex Displacement). This algorithm is very simple but emperically efficient for the edge coloring of a graph. By assuming that $P \neq NP-Complete$ and admitting the probalistics solutions, they proposed a simple and fast heuristics algorithm to 3-edge-color a cubic graph. From the experimental data used, it was suggested that when G is 3-edge colorable, their algorithm efficiently gives an explicit solution and allows them to break intractability. Then they modified and applied the algorithm for Δ - reguler graphs ad graphs wiht maximum degree Δ. Looking at this simple CVD algorithm and the advantages obtained, the writer is interested to apply the CVD algorithm for a bipartite graph. A bipartite graph, also known as a bigraph, is a graph with set of vertices that are decomposed into two disjoint sets such that no two vertices same set are adjacent. There are many occasions in life where you need to match up the elements of two sets. On most occasions this is a problem you will probably tackle without really considering that you are doing it such as timetabling, radio frequency communications systems, communication network and computer science.

Edge Coloring

Consider an undirected graph $G = (V, E)$ where V is a set of vertices (nodes) and E is the set of edges, with $|V| = n$ and$|E| = m$. Edge coloring is the coloring of each edge in the graph such that no two adjacent edges has the same color. Edge coloring is usually aims to get the minimum number of colors to color all edges of a graph. A k-edge coloring for graph G is to use of part or all of k colors to color all the edges in G so that every tow adjacent edges have a different color. If G has k-edge coloring, it is said edges in G colored with k colors (k-colourable).

The edge chromatic number of graph G is the minimum number of colors with which all edges of graph G can be colored. The chromatic number of edge coloring is usually denoted by $\chi(G)$. In general, the colors used to color the edges of a graph are represented by numbers such as 1, 2, 3, ..., k. Then the edge coloring problem is to find a coloring of G with $\chi(G)$. Let $\Delta(G)$ be the maximum degree over the vertices of G. Thus, $\chi(G) \geq \Delta(G)$. The edge coloring problem for simple graphs, i.e. those without loops or parallel edges, appears to be greatly simplified by the following theorem of Vizing [18].

Theorem 1. If G is a simple graph, then $\chi(G) = \Delta(G)$ or $\Delta(G) + 1$.

By this Vizing's theorem, the decision problem for 3-regular graphs is just to determine whether the chromatic index is 3 or 4. There are some proof in some literature for this Vizing theorem that gives a polynomial time algorirhtm for coloring any simple graph with $\Delta(G) + 1$, such as Faber, Ehrenfeucht and Kierstead [18] and Lovasz and Plummer [19] but there is unresolved problem for a simple graph which is to determine whether it can be colored with $\Delta(G)$ color. As shown by Holyer [7], this problem is $NP - complete$, even for 3-regular graphs. In this paper, all graphs are assumed to be connected and with no loops or multiple edges.

CVD Algorithm

Good heuristic algorithms are essential to tackle hard optimization problems (Insani et all [8]). Using heuristics one might be able to find good feasible solution quickly. Moreover, heuristics are useful to tighten the bounds and consequently to reduce the search space. This motivates the large amount of literature concerning the heuristic and metaheuristic approaches for edge coloring problem. In this paper, we present an application of CVD heuristics algorithm found by Fiol and Vilatella [2] to solve the edge coloroing problem and apply it to bipartite graphs. To solve the edge coloring probkem, CVD heuristics algorithm starts the algorithm with a random coloring of the edges of G using 3 colors. Thus, the graph will probably have several edges that have the same color. Then the basic idea of CVD algorithm is to re-color the conflicting vertex along the path. A conflicting vertex is a vertex whose incident edges are not properly colored. In other word, at least two edges that are incident to this vertex have the same color.

The main loops in the CVD algorithm consists of two steps. The first step is choosing randomly (if exists) a conflicting vertex and take one the unproperly colored edges as the starting point of a Kempe Chain. A Kempe chain is a 2-edge colored path whose two locors are the original unproper color of the starting edge and a new color that makes the vertex unconficting, or at least reduces the number of edges with repeated color. Here, a new color always exists and can be randomly chose when it is not unique. Kempe chain was found by

Alfred Kempe in 1879 to proof the four color conjecture. The original Kempe chains were used in the contecx of colorings of countries on a map, or in other word to face colorings of a plane graph such that no two adjacent faces have the same color. Suppose G is a graph (V, E) and a colouring function is given as $C: E \rightarrow S$ where S is a finite set of colors which contains at least two distinct colors i and j. If e is an edge with color i, then the (i, j)-Kempe chain of G containing e is the maximal connected subset of V which contains e and whose edges are all colored either i or j. This definition is usually applied where S has three elements, say i, j, and k, and where G is a cubic graph. A cubic graph is a graph having 3 edges for every its vertex. If such a graph is properly coloured, then each vertex must have edges of three distinct colours, and Kempe chains end up being paths. Another simple definition of a Kempe chain is given as follow: let $G(a, b)$ be the subgraph consisting of all the edges of color i or j, with any vertices incident to them.Kempe chian is any connected component $H(i, j)$ of $G(i, j)$. By the definition, at any vertex of $H(i, j)$, there will be at most one dge of color i and one edge of color j with at most two edges in all. There will be two types of chain will happen: 1) Every vertex in the chain has two such edges, means that it forms a closed path (cycle). Being colored alteratingly with two different colors, the number of edges must now be even; 2) A vertex misses out on having an edge of one of those colors, and the chain stops. In a finite graph, the other end of the chain must now also terminate somewhere (at a vertex that misses out either one of the colors). The chain is an open path of one or more edges (its length can be even or odd).

The second main step in the CVD algorithm is swapping the two colos along the chain. Kempe chain arguments is a technique that can be used by swapping the colors in one $H(i, j)$. Swapping can be used to free up a color somewhere. It does not create a new clnflict but it moves the confliciting vertex along the path. The algorithm will stop the color swapping when 1) another conflicting vertex is reached; 2) where the conflict may cancel out, or 3) the number of swaps reaches a given limit. We set the given limit at most the number of vertices in the graph. The algorithm keeps running until one of these condition is reached:1) there is no conflicting vertex anymore or 2) all conflics are solved (getting a 3-edge colored), or 3) it reaches fixed limit.

Accodirng to Rauf & Insani [15], the basic steps of CVD algorithm for edge coloring are as follow:

1. Color the all the edges with a minimum number of color (the maximum degree (Δ) vertices of the graph).
2. Find conflicting vertices:
    a. If a conflicting vertex is found, then go to step 3.
        i. If the conflicting vertex is not found, then check if the number of colors for the edge is optimal.
            1. If the number of colors arc is not optimal, then randomly find a possible edge-color that is likely to be swap with another color that does not make a new conflict we obtain an optimum solution. Then go to step 2.b.2.
            2. If the number of colors arc is optimal, then stop the process as it has reached the desired optimal solution.
3. Select one of the conflicting vertices as a starting point for the Kempe chain, says vertex a.
4. Choose an edge that incident with the chosen vertex, says edge e_1.
5. Give a new color (i.e. color i) for edge e_1. Then check whether the edge e_1 directly

related to another conflicting vertex.

   a. If edge e_1 adjacent to another conflicting vertex, then back again to step 2.
   b. If edge e_1 is not adjacent to another conflicting vertex, then go to step 6.

6. Determine the next edge that is adjacent with e_1 which will be used as the second edge in the Kempe chain, says e_2.
7. Give color to e_2, could be the available color or a new ones, as long as different should stick with the original color with a new color or a different color as long as the color of e_1, says color j. Thus we form a simple Kempe chain $H(i, j)$ that is constructed by color i and j.
8. Swap the of two colors (colors i and j) along the Kempe chain along the path, stop and go to step 9 if one of these conditions is reached:
   a. Onother conficiting vertex is connected.
   b. The number of swapping of color had been reached a certain limit (the number of vertices (n) in the graph).
   c. Formed a closed path (cycle).
9. Start the procedure again from step 2 and repeat the Kempe chain argumet with a different Kempe chain.

Bipartite Graphs

A bipartite graph is a graph whose vertices can be divided into two independent sets, U and V such that every edge (u, v) either connects a vertex from U to V or a vertex from V to U. In other words, for every edge (u, v), either u belongs to U and v to V, or u belongs to V and v to U. There is no edge that connects vertices of same set. The intereseting point of bipartite graphs is these graphs are equivalent with to two-colorabe graphs (in terms of vertex coloring). Another interesting thing about bipartite graphs is all acyclic graph graphs are bipartite and it is a class 1 graph.

There are many kinds of bipartite graphs, the most known are: singleton graph, square graph, claw graph, utility graph, cubical graph, Herschel graph, Franklin graph, Heawood graph, tesseract graph, Möbius-Kantor graph, Hoffman graph, Pappus graph, Folkman graph, Desargues graph, truncated octahedral grap, Walther graph, Levi graph, Dyck graph, great rhombicuboctahedral graph, Gray graph, Balaban 10-cage, Foster graph, Horton 96-graph, great rhombicosidodecahedral graph, and Tutte 12-cage.

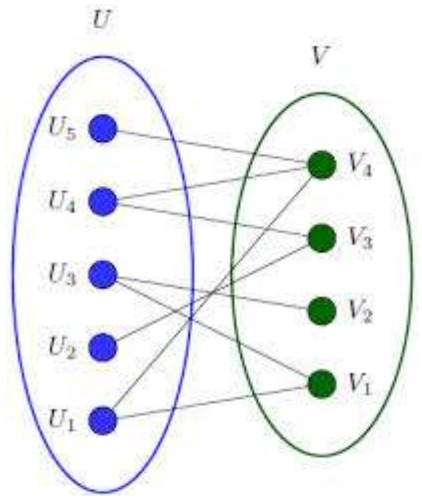

Figure 1- An example of a bipartite graph

RESULT AND DISCUSSION

Performance Results

We tested the performance of the CVD algorithm by measuirng the time needed to several types of bipartite graphs. We measured the average time per iteration and plotted the result as shown in the picture below. We implemented the CVD algorithm on edge-coloring of bipartite graphs using Python languange that is written by Fiol and Vilatella [2] for a random reguler graph. We generated 20 different types of bipartites graphs with number of vertices between 10 - 100 and maximum degree of 3, 7, 11, and 15, using the built-in generator available NetworkX.

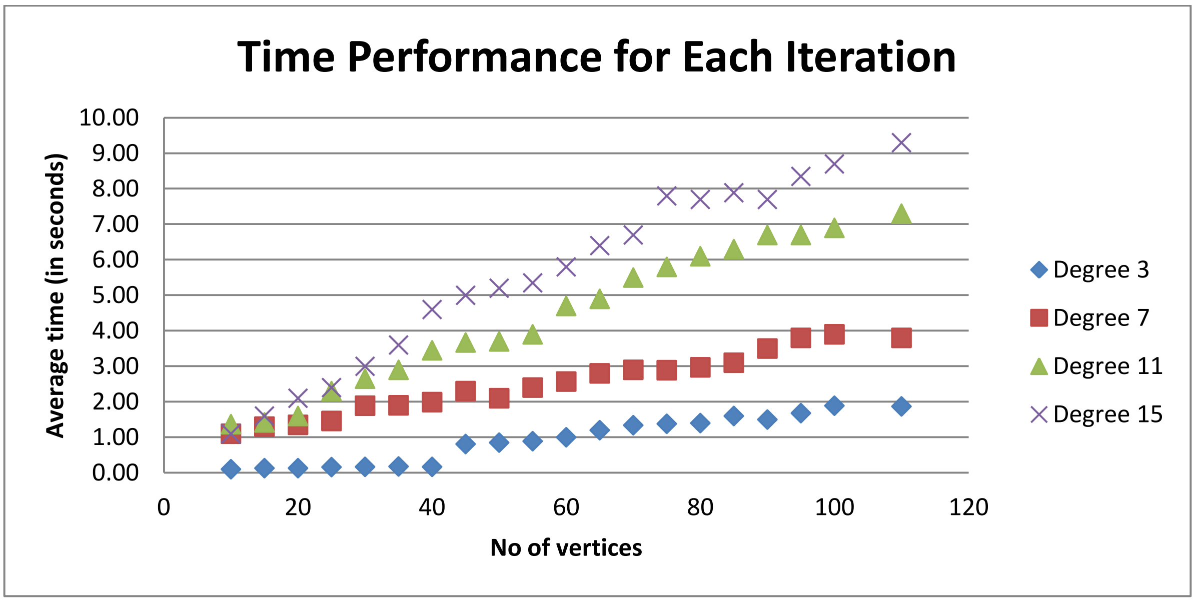

Figure 2 - Time Performance

The results obtained is this paper are similar with the results obtained on Fiol and Vilatella [2] paper. The behaviour of average average running time appears to grow linearly along with the number of vertices. The higher the maximum degree of the graphs the longer is the running time to solve the edge coloring problem of the graph.

CONCLUSION AND SUGGESTION

From the results above, we can see that the CVD algorithm can solve the edge coloring problem in a very reasonable time not only for cubic and reguler random graph, but also a different type of graph which consider very important in a real life, bipartite graphs. This heuristics algorithm could be use also to solve other types of graphs and might be implemented in a real life problem.

REFERENCES

[1] Fiol, M.A. (eds). (1991). *A Boolean algebra approach to the construction of snarks, in Graph Theory, Combinatorics and Applications*. vol.1 (eds. Y. Alavi, G. Chartrand, O.R. Oellermann, and A.J. Schwenk). John Wiley dan Sons. (1991). Hlm. 493-524.

[2] Fiol, M.A. and Vilatella, J. (2012). *A Simple and Fast Heuristic Algorithm for Edge-coloring*

of Graphs. Jurnal Metematika Universitas Politeknik Catalonia, Departemen Matematika Terapan IV, Barcelona, Catalonia (arXiv:1210.5176v1 [math.CO]). Hlm. 1-10.

[3] Fiorini, S., and Wilson, R.J. (1977). *Edge-colouring of graphs*. Pitman.

[4] Frank, H. (1994). *Graph Theory*. Addison-Wesley Company.

[5] Gardner, M. (1976). *Mathematical Games: Snarks, Boojums and other conjectures related to the four-color-map theorem*. Sci. Amer. 234. Hlm. 126-130.

[6] Hopcroft, J. E., & Karp, R. M. (1973). *An n^5/2 algorithm for maximum matchings in bipartite graphs*. SIAM Journal on computing, *2*(4), 225-231.

[7] Holyer, I. (1981). *The NP-completeness of edge-colouring*. SIAM J. Comput. 10. Hlm. 718-720.

[8] Insani, N. (2012). *Pemilihan Algoritma Heuristik Terbaik untuk Suatu Masalah Graf Berdasarkan Sifat/Karakteristiknya(Instance Features).* Prosiding, Seminar Nasional Penelitian, Pendidikan dan Penerapan MIPA. Yogyakarta: FMIPA UNY.

[9] Isaacs, R. (1975*). In_nite families of nontrivial trivalent graphs which are not Tait colorable*. Am. Math. Monthly82. No.3. Hlm. 221-239.

[10] Lee, T.T., Wan, Y. dan Guan, H. (2011). *Randomized Δ-edge-Coloring via quaternion of complex colors (Extended Abstract).* The Journal of Mathematics Science (arXiv:1104.1852v1 [cs.DS]).

[11] Marathe, M. V., Panconesi, A., & Risinger, L. D. (2000, July). *An experimental study of a simple, distributed edge coloring algorithm*. In Proceedings of the twelfth annual ACM symposium on Parallel algorithms and architectures (pp. 166-175). ACM.

[12] Munoz S, Ortuno M.T, Javier R dan Yanez J. (2005). Colouring Fuzzy Graph. *The Journal of Management Science.* 33. Hlm. 211-221.

[13] Nemhauser, G. L., & Park, S. (1991). *A polyhedral approach to edge coloring*. Operations Research Letters, *10*(6), 315-322.

[14] Robinson, W. dan Wormald,N.C. (1994). *Almost all cubic graphs are hamiltonian*. Random struct. & alg. 5 (Nomor 10.1002/rsa.3240050209). Hlm. 363-374.

[15] Rauf, A. and Insani, N. (2014). *Pewarnaan Busur Graf Dengan Algoritma Heuristik Conflicting Vertex Displacement (CVD).* Skripsi. Universitas Negeri Yogyakarta

[16] Theresia, M.H. dan Seputro, T. (1992). *Graf Pengantar*. University Press IKIP: Surabaya.

[17] Wilson, R.J. dan Watkins, J.J. (1990). *Graph: An Introduction Approach: A first Course In Discrete Mathematics*. New York: John Wiley & Sons. Inc.

[18] Vizing, V. G. *On an estimate of the chromatic class of a p-graph*. Diskret Analiz 3 (1964) 25–30. *Mathematical Reviews (MathSciNet): MR180505*.