Optical Graph 3-Colorability

Sama Goliaei and Saeed Jalili

SML Lab, Electrical and Computer Engineering Department,
Tarbiat Modares University, Tehran, Iran
{goliaei,sjalili}@modares.ac.ir

Abstract. The graph 3-colorability problem is a decision problem in graph theory which asks if it is possible to assign a color to each vertex of a given graph using at most three colors, satisfying the condition that every two adjacent vertices have different colors. It has been proved that the graph 3-colorability problem belongs to NP-complete class of problems which no polynomial resources solution is found for them yet.

In this paper, a novel optical solution to the graph 3-colorability problem is provided. In this solution, polynomial number of black filters are created in preprocessing phase each of which has exponential size and requires exponential time to be created. After preprocessing phase, the provided solution takes $O(n + m)$ time to decide if a given graph is 3-colorable or not, where the given graph has n vertices and m edges.

Keywords: Unconventional Computing, Optical Computing, NP-Complete, Graph Coloring, Graph 3-Colorability.

1 Introduction

Graph coloring problems is a class of problems in graph theory seeking to assign a color to each vertex of a given graph in such a way that every two adjacent vertices have different colors. Graph coloring problems arise from many real-world applications such as scheduling [1] making it necessary to find efficient solutions for these problems.

Graph 3-colorability problem is one of the graph coloring problems having specific applications in resource allocation and scheduling [2]. The graph 3-colorability problem asks if it is possible to assign a color to each vertex of a given graph in such a way that every two adjacent vertices (two vertices which are connected via an edge) have different colors, using at most three colors. It has been proved that 3-colorability problem is an NP-complete problem and like every other NP-complete problems, no polynomial resources solution is found yet.

Light is a natural phenomenon used in computation because of its special physical properties such as its parallel motion. Many NP-complete problems have recently investigated in optical computing such as the 3-SAT problem [3], the Hamiltonian path problem [4], the exact cover problem [5], the subset sum problem [6], the maximum clique problem, the vertex cover problem, the partition

S. Dolev and M. Oltean (Eds.): OSC 2010, LNCS 6748, pp. 16–22, 2011.

problem, the 3D-matching problem, the permanent problem, and the traveling salesman problem (TSP) [7].

New computational capabilities of optical computing in comparison to conventional computing have resulted to obtain more efficient solutions for these NP-complete problems, brings the idea that using optical computing to solve 3-colorability problem will also result to obtain more efficient solutions. Although the 3-colorability problem have been investigated in other branches of unconventional computing such as DNA computing [8] and quantum computing [9], but it seems that no optical solution for the graph 3-colorability problem (based on natural properties of light) is provided yet.

In this paper, a novel optical solution for the graph 3-colorability problem is provided. The solution takes just polynomial time to solve each problem instance, but exponential time in preprocessing phase. In the next section, the graph 3-colorability problem is defined. Provided optical solution and its complexity analysis are explained in sections 3 and 4, respectively. Finally, the conclusion of the paper is provided in section 5.

2 The Graph 3-Colorability Problem

In graph theory, graph coloring means to assign a color to each vertex of a graph. A proper graph coloring is assigning different colors to every two adjacent vertices (two vertices are adjacent if and only if they are connected via an edge). A proper 3-coloring for a given graph G, is a proper coloring for G using just three colors. A given graph G is 3-colorable, if and only if there exists at least one proper 3-coloring for G. Fig. 1 shows a 3-colorable graph and a proper 3-coloring solution. Fig. 2 shows a graph which is not 3-colorable.

The graph 3-colorability problem is a decision problem (requires answer "yes" or "no") which is looking to find if a given graph G is 3-colorable or not [10]. As the graph 3-colorability problem is a decision problem, it is not seeking to find a proper 3-coloring, but just requires answer "yes" if the given graph is 3-colorable, and answer "no" otherwise.

It has been proved that the graph 3-colorability problem is an NP-complete problem [10] and the best algorithms to solve this problem is exponential time in the conventional computers.

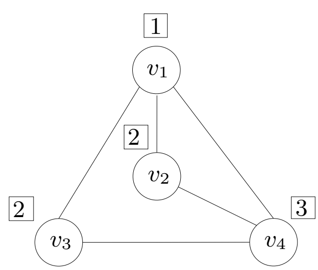

Fig. 1. Example of a 3-colorable graph and a possible proper 3-coloring solution. The colors assigned to the vertices are specified in the squares.

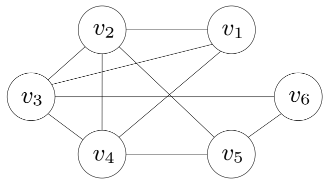

Fig. 2. Example of a graph which is not 3-colorable

3 The Optical Solution

In this section, the optical solution to the graph 3-colorability problem is explained. We first show how to represent a 3-coloring of a given graph having n vertices as a binary sequence having length $2n$. Then we explain how to use light and optical devices to generate all binary sequences having length $2n$ and filter improper sequences to obtain exactly those binary sequences representing a proper 3-coloring for the given graph in efficient time.

3.1 Graph 3-Colorings as Binary Sequences

It is possible to represent each 3-coloring of a graph G having n vertices $v_1, \cdots, v_n$ as a binary sequence having length $2n$ in the form of $a_1b_1a_2b_2 \cdots a_nb_n$ where $\overline{a_ib_i}$ represents the index of the color assigned to vertex v_i $(0 < i \leq n)$ in binary format. In the other words, $\overline{a_ib_i}$ is "01" if v_i is colored by the first color, "10" if it is colored by the second color, and "11" if v_i is colored by the third color. For example, the provided 3-coloring for the given graph in Fig. 1 is represented as "01101011".

Now, the question is how we can confirm if a given binary sequence $a_1b_1 \cdots a_nb_n$ represents a proper 3-coloring for a given graph G with n vertices $v_1 \cdots v_n$ and m edges $e_1, \cdots c_m$. Since $\overline{a_ib_i}$ $(0 < i \leq n)$ represent the color of v_i, two conditions should be satisfied (and it is also sufficient) in order the given binary sequence represents a proper 3-coloring for G:

- The color of each vertex should be one of three colors, so for each i $(0 < i \leq n)$, $\overline{a_ib_i}$ should be one of "01", "10", or "11". In the other words, for each i $(0 < i \leq n)$ it is needed $a_i \neq 0 \vee b_i \neq 0$.
- For each edge e of the given graph G connecting two vertices v_i and v_j, the color of v_i should be different from the color of v_j. So $\overline{a_ib_i}$ and $\overline{a_jb_j}$ should be different, or in the other words, it is needed $a_i \neq a_j \vee b_i \neq b_j$.

It can be easily seen that these two conditions are also sufficient for the given binary sequence having length $2n$ to represent a proper 3-coloring for G, as they guarantee a color is assigned to each vertex using at most three colors and every two adjacent vertices have different colors.

3.2 Finding Proper Binary Sequences

To solve the graph 3-colorability problem for a given graph G having n vertices $v_1, \cdots, v_n$, we are looking to find proper binary sequences representing proper 3-colorings for G among all binary sequences having length $2n$. The main idea is to divide a square of light into 2^{2n} square sections (number of all possible binary sequences with length $2n$) and consider each section as a binary sequence having length $2n$. Then we use some black filters which do not let light to pass from the sections not representing a proper 3-coloring for G.

Now we want to create some black filters which be punched in sections corresponding to binary sequences representing proper 3-colorings (In the other words, if a section is correspond to a binary sequence not representing a proper coloring, then the section is black in at least one filter). Note that a binary sequence in form of $a_1b_1 \cdots a_nb_n$ represents a proper 3-coloring for G if and only if it satisfies the two conditions explained in section 3.1. So we create two classes of filters, vertex filters to filter binary sequences not satisfying the first condition, and edge filters to filter binary sequences not satisfying the second condition.

Vertex Filters. The first condition described in section 3.1 expresses that the color of each vertex should be one of three colors 01, 10, or 11. Each vertex filter blocks those light rays which does not assign a valid color to one vertex. So we create n vertex filters denoted by $f_{v_1}, \cdots, f_{v_2}$, in such a way that for each vertex v_i $(0 < i \leq n)$, f_{v_i} is divided into 2^{2n} sections (representing all possible binary sequences having length $2n$ in form of $a_1b_1 \cdots a_nb_n$), and the filter is black in those sections where $a_i = 0 \wedge b_i = 0$. As f_{v_i} is punched in other sections, it filters those binary sequences which does not assign a valid color to v_i.

These vertex filters are independent from the structure of the given graph, so they can be created in preprocessing phase and be used to solve many problem instances.

Edge Filters. The edge filters are created to block light rays in the sections corresponding to binary sequences not satisfying the second condition described in section 3.1, which expresses that the color of each two adjacent vertices should be different. So for each edge $e_u = (v_i, v_j)$ $(0 < u \leq m)$ in G, we create an edge filter denoted by f_{e_u} and divide it into 2^{2n} sections (representing all possible binary sequences having length $2n$ in form of $a_1b_1 \cdots a_nb_n$), and make the filter black in those sections where $a_i = a_j \wedge b_i = b_j$ (the filter is punched in other sections). Hence f_{e_u} filters those binary sequences assigning the same color to v_i and v_j.

Note that these edge filters are dependent just on the corresponding edge and are independent from the whole structure of the graph, so they can be created in preprocessing phase and be used to solve many problem instances.

3.3 Solving the Problem Instances

To solve a problem instance for a given graph G having n vertices $v_1, \cdots, v_n$ and m edges $e_1, \cdots, e_m$, we place n vertex filters $f_{v_1}, \cdots, f_{v_n}$ and m edge filters

$f_{e_1}, \cdots, f_{e_m}$ one after each other (the order of placing filters is not important), and emit light to the filters as is shown in Fig. 3. Using a convex lens and an optical sensor, we examine if there are some light rays passing through the filters or not. If some light rays have been passed, so there are some sections which are punched in all $n + m$ filters, thus the binary sequences corresponding to these sections satisfy both conditions described in section 3.1. This means that there is at least one binary sequence representing a proper 3-colorings for G and hence G is a 3-colorable graph. If no light rays passes through the filters, no binary sequence with length $2n$ satisfies the conditions described in section 3.1 and G is not a 3-colorable graph.

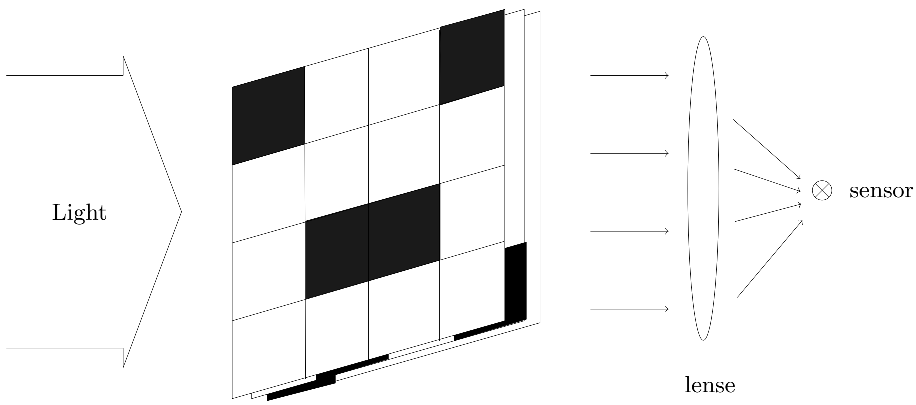

Fig. 3. Arrangement of optical devices in the provided optical solution for the 3-colorability problem

4 Complexity

In the provided optical solution, vertex filters and edge filters are created in preprocessing phase and are used to solve many problem instances. To solve a problem instance for graphs having at most n vertices, n vertex filters should be created, each of which has 2^{2n} sections and requires $O(2^{2n})$ time to be created.There are $\binom{n}{2}$ possible edges in graph having at most n vertices, so $\frac{n(n-1)}{2}$ edge filters are required to be created in preprocessing phase each of which requires $O(2^{2n})$ time to be created. Hence the provided solution requires $O(n^2 2^{2n})$ time in preprocessing phase to create $O(n^2)$ black filters (vertex and edge filters). As the shape of the filters is square and each filter has 2^{2n} sections, the side length of a filter is 2^n times side length of a section.

To solve each instance of the problem for a given graph having n vertices and m edges, $n + m$ black filters (n vertex filters and m edge filters) are placed next to each other and light is passed through them. So the provided solution takes $O(m + n)$ time to find if a given graph is 3-colorable or not.

Note that the exponential size of the black filters may cause some difficulties in implementation of the provided solution. Considering side length 30 micro meters for each section obtained by laser micro drilling technology, the side length of each black filter is 1.3 meters for $n = 15$ and 2.6 meters for $n = 16$. Hence, the provided solution is not practicable for large values of n.

5 Conclusion

In graph theory, the graph 3-colorability problem asks if it is possible to assign a color to each vertex of a given graph using just three colors in such a way that every two adjacent vertices have different colors. The graph 3-colorability problem arises in many real-world applications makes it necessary to find efficient solutions for this problem. It has been proved that the graph 3-colorability problem is an NP-complete problem, which no polynomial time and other resources solution is found yet.

In this paper, an optical solution to the graph 3-colorability problem is provided requiring polynomial time to solve each problem instance after an exponential time preprocessing phase. In the provided solution, each graph 3-coloring for a graph having n vertices is represented as a binary sequence with length $2n$. A square shape space is considered to be divided into 2^{2n} sections and each section is considered as a possible binary sequence having length $2n$. Black filters created in preprocessing phase are placed each after other in such a way that each section corresponding to a binary sequence not representing a proper 3-coloring for the given graph is black in at least one black filter. Passing light through the filters, if at least some light rays pass through all filters detecting by an optical sensor, the given graph is 3-colorable, and if no light ray passes through the filters, every sections are black in at least one black filter and hence the given graph is not a 3-colorable graph.

The black filters are created in preprocessing phase and may be used to solve many problem instances. To solve the problem instances for the graphs having at most n vertices, the provided solution requires to create $O(n^2)$ black filters each of which having exponential size and requires $O(2^{2n})$ time to be created. After preprocessing phase, the provided solution requires $O(m+n)$ time to solve each problem instance for a given graph having n vertices and m edges.

The exponential size of each black filter causes difficulties in implementation of the provided solution and the solution is not practicable for large values of n.

Acknowledgments. This work is partially supported by Iranian National Institute of Elites.

References

1. Myszkowski, P.B.: Solving scheduling problems by evolutionary algorithms for graph coloring problem. SCI, vol. 128, pp. 145–167 (2008)
2. Zweben, M., Fox, M.: Intelligent Scheduling. Morgan Kaufmann, San Francisco (1998)
3. Goliaei, S., Jalili, S.: An optical wavelength-based solution to the 3-SAT problem. In: Dolev, S., Oltean, M. (eds.) OSC 2009. LNCS, vol. 5882, pp. 77–85. Springer, Heidelberg (2009)
4. Oltean, M.: Solving the hamiltonian path problem with a light-based computer. Natural Computing: an International Journal 7(1), 57–70 (2008)
5. Oltean, M., Muntean, O.: Exact cover with light. New Generation Computing 26(4), 327–344 (2008)
6. Oltean, M., Muntean, O.: Solving the subset-sum problem with a light-based device. Natural Computing: an International Journal 8(2), 321–331 (2009)
7. Dolev, S., Fitoussi, H.: Masking traveling beams:optical solutions for np-complete problems, trading space for time. Theoretical Computer Science 411, 837–853 (2010)
8. Lin, M.: 3d dna self-assembly model for graph vertex coloring. Journal of Computational and Theoretical Nanoscience 7(1), 246–253 (2010)
9. DHondt, E.: Quantum approaches to graph colouring. Theoretical Computer Science 410(4-5), 302–309 (2009)
10. Goldreich, O.: Computational Complexity: a conceptual perspective. Cambridge University Press, Cambridge (2008)