

GraphTea: Interactive Graph Self-Teaching Tool

M. Ali Rostami, Azin Azadi, and Masoumeh Seydi

Abstract—It is becoming increasingly difficult to ignore the importance of graph theory and its application in various scientific areas. There is an urgent need for new tools to investigate this field. Since the aspects of graph algorithms are tricky to be understood by students, there are literatures pointing specially to the educational concepts of this topic. We propose an educational framework, targeting all aspects of graph theory and algorithms. This paper presents the design paradigm of such a framework as well as the teaching strategy using this tool.

Keywords—Graph theory and algorithms, rich graph-editing framework, interactive educational software, GraphTea, self-teaching tool.

I. INTRODUCTION

GRAPH representation is applicable in several scientific fields in which the solutions are heavily based on the structure of the problem. Graph interpretation provides another view of the problem. Neural networks, sparse matrices, routers in internet, social networks, and train networks are just a few examples of the areas which benefit from this theory. Graphs can play an essential role in modeling, better understanding, and simplifying of problems. The importance of this field has motivated the educational system to investigate various ideas and make so many efforts to teach it. Therefore, numerous softwares and libraries have been developed to edit and visualize different classes of graphs. However, a general-purpose software targeting all aspects of graph theory is not easy to implement. Also, a area-specific graph theory software is not meant to cover also all aspects of other areas. In social networks, for example, the dense subgraphs are in the center of interest and in sparse matrices, the connection between matrix and graph. Large graphs and suitable small graph visualization can be intended too. As soon as the visualization of graph is done, the graph algorithm visualization becomes a priority.

The comprehensive understanding of the graph and algorithms in classroom is the aim of this paper. We have designed a software that provides a flexible and easy-to-work environment for teachers so that they can design new courses on graph theory and algorithms using this software. Moreover, since the instructions regarding the algorithm steps are provided to be followed by the student, the software is considered as a self-teaching tool. The new contribution of this software is to provide a rich editing framework while focusing on educational aspects besides.

In this paper, we will present a software and also a process explaining how to utilize this software for teaching and learning graph theory. In Sec II, the related works are discussed and the unique properties of our software. Then we introduce our

tool, called GraphTea, and its abilities in Sec III. Sec IV takes a more detailed look at the algorithm visualization module of GraphTea. Finally, Sec V concludes this paper and discusses future works.

II. RELATED WORK

Wide application of graph theory has been a motivation to design multiple tools to study these topics in academia. Some of these tools are designed for educational purposes, but they are not good enough in visualization. Some others have satisfying visualization features, however they do not provide adequate editing power. In our knowledge, there is no other general-purpose software other than GraphTea with the focus on education, that has rich editorial and visualization options. In addition, most of these softwares are not facilitated for self-teaching and they need to be taught and explained step-by-step by an instructor in a classroom. Here, we look at a few important tools with educational focus.

CABRI-Graph [1] and Gato/CATBox [2] are well-designed for graph algorithm visualization and animation. However, the interactively editing power of the graph structure is very limited. Tulip [3], [4] is another tool for information visualization. This tool could be used efficiently when the user knows how to program Python language. Again, this tool does not have a suitable interactive editing power as well as flexible way of creating graphs. Furthermore, there is software that is specifically designed for teaching a particular area, such as EXPLAIN [5], [6], and Heath [7].

III. GRAPHTEA

GraphTea is a rich graph-editing framework, which has been started to develop under the name of "GraphLab" in the Department of Mathematics in Sharif University of Technology. The authors later continued to develop the software more focusing on educational purposes. The strength of GraphTea is the consideration of different aspects of graphs which would be essential in designing an educational plan. GraphTea attempts to ease the teaching process for both teachers and students by emphasising the visual features and educational ones. Figure 1 shows an overview of GraphTea together with graph coloring results and the independent sets, in which one result is selected. The major parts of graph theory and algorithms are implemented in GraphTea, from which we go through three important ones. The graph generation and modification in GraphTea is explained in Sec. III-A and the graph reports in Sec. III-B. Also, the algorithm visualization and its application are explained in Sec. IV

M. A. Rostami is with the department of Mathematics and Computer Science, Friedrich Schiller University Jena, Jena, Germany. Masoumeh Seydi is with the department of Computer Science, University of Konstanz, Konstanz, Germany. Azin Azadi is working in Jovoto, Berlin, Germany.

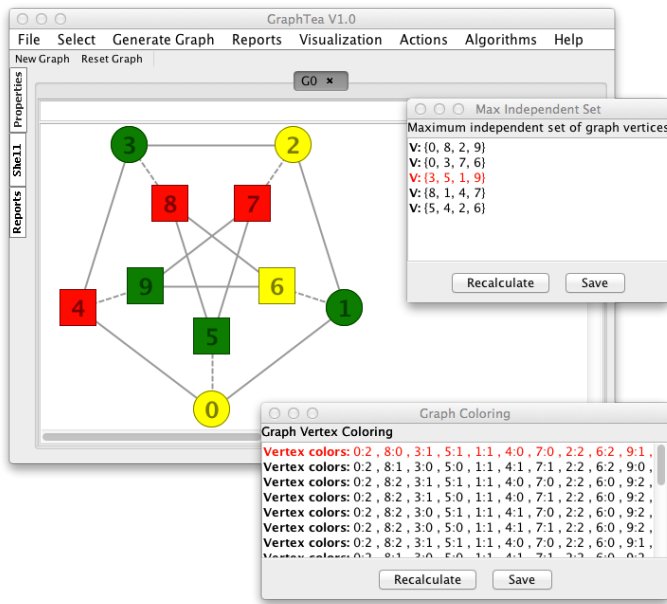


Fig. 1. An overview of GraphTea and two visualized reports on a given graph.

A. Graph Generation and Modification

Graph generation could be named as the main feature in a graph educational software. In GraphTea, a graph can be generated in three ways. The simplest way is to use the mouse. Each mouse click on the main windows adds a new vertex. Two vertices can be connected by click on the source vertex and dragging the mouse to the center of destination vertex. The generated vertices and edges or a group of them can be selected by mouse left-click. The selected vertices and edges can be moved or deleted. The mouse right-click opens a menu, which provides different functionality over the selection.

The second way of generating a graph is to use the predefined ones under the menu "Generate Graph". As it could be seen in Figure 2 (Left), there is a list of highly usable graphs in this menu, which are generated by the software and prepared to be used. When a graph from this list is selected, a window will be opened asking the parameters of the target graph. For example, if a complete graph is selected, the number of vertices would be asked as a parameter.

The graph can also be loaded from a file as a third way. Three types of files are supported so far in GraphTea. GraphML [8] and Matlab file formats are two classic ones. In addition, GraphTea has its own file type for saving and loading, called "SimpleGraph".

As mentioned before, GraphTea is powerful in graph modification. The properties of graph in general as well as vertices and edges are editable. The properties are available in left toolbar "Properties". By selecting an edge or a vertex the corresponding properties will appear. No selection means the generic properties of the graph is demanded. Shapes, colors, sizes, and many other properties are shown by this toolbar. Figure 2 (Right) shows how to alter the properties of a graph. The shape of vertices 0, 6, 2, and 1 has changed from circular to rectangular. Some edges, like the edge between 3 and

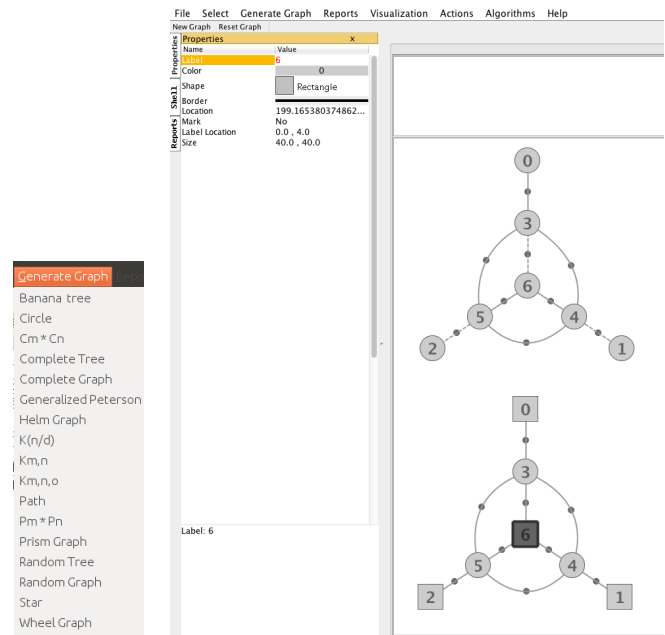


Fig. 2. (Left) List of predefined graphs, accessible under the menu "Generate Graph". (Right) Properties of the selected vertex is shown. Each vertex can be handled alone. The properties of edges are already changed to be curved edges.

4 are curved edges, though some others are straight lines. This confirms the strength of GraphTea and how almost all properties of vertices and edges are editable easily.

B. Graph Reports

GraphTea provides desired information as reports about the graph, which the user would request. Reports in GraphTea are divided into two groups, the one with numbers or string as the results, and the other resulting in a set of vertices or edges. In case that vertices and edges are outputs, there is an option to visualize the result on the corresponding graph. For example, Figure 1 shows the visualized report of independent set and coloring. It is seen that graph has five independent sets and the student can select each of them to see them instantly on the graph. Hence, a report is provided either as a list containing a set of information or in a visualized form (see Figure 3). The reports are available via the menu "Reports" and left toolbar as well, as shown in Figure 3. This toolbar has the option to generate all reports and to prepare them altogether.

IV. ALGORITHM VISUALIZATION: TEACHING DEPTH-FIRST SEARCH (DFS) WITH GRAPHTEA

In GraphTea, an algorithm is a step-by-step procedure, when each step applies an action. An action is either getting some input from the student or modifying the graph. The input could be a mouse click, key input, or any numerical and string input. For example, starting and destination vertices should be selected in shortest path algorithm and are considered as inputs. The changes in the graph can be coloring, removing, adding, or even moving a vertex or an edge.

Since the information of algorithm and its progress are described as instructions, for each step the instruction is

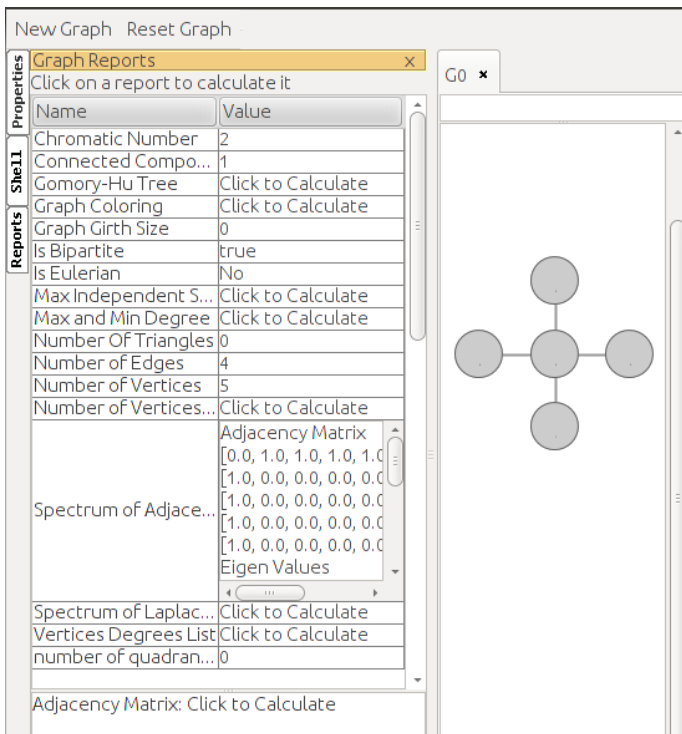


Fig. 3. Left toolbar of GraphTea provides another way to compute reports on the graph.

explaining either the algorithm status in that step or the input required (see Figure 4(Right)). As a consequence, the algorithm education can be done by students without any tutorial sessions. Besides, the instructions are printed along with the algorithm execution and this helps to simultaneously see the algorithm output of each step on the graph and consequently to understand it better. Additionally, GraphTea has the possibility of controlling the speed of algorithm execution and results representation on the graph. This feature prevents the student from getting bored or losing some steps. The speed control can be done by a slide bar as shown in Figure 4(Right). The student could stop the algorithm to analyse specific steps more precisely, such as getting different reports at that stage. When the algorithm is stopped, the student can go forward one step in the algorithm or resume the normal execution.

Here we are going to illustrate this educational process with an example in a classroom, which of course could be done by student themselves. Suppose teaching DFS algorithm [9] is the goal and the given graph is visualized. DFS algorithm should be selected from the list of algorithms listed in "Algorithm" menu (see Figure 4(Left)). Then, the dialog "Algorithm Runner" appears as in Figure 4(Right). This dialog provides the functionality of pausing and playing the steps, and playing just one step of the algorithm. In addition, the speed could be controlled by the existing slide bar. The part "Algorithm Output" contains the instructions. After choosing a vertex, the algorithm starts to run and the visited vertices are colored.

It should be mentioned that the student can edit the graph at any stage and reinstate or continue the algorithm. When the algorithm is completed, the results are still present on the main window until the "Reset Graph" is selected. The resulting

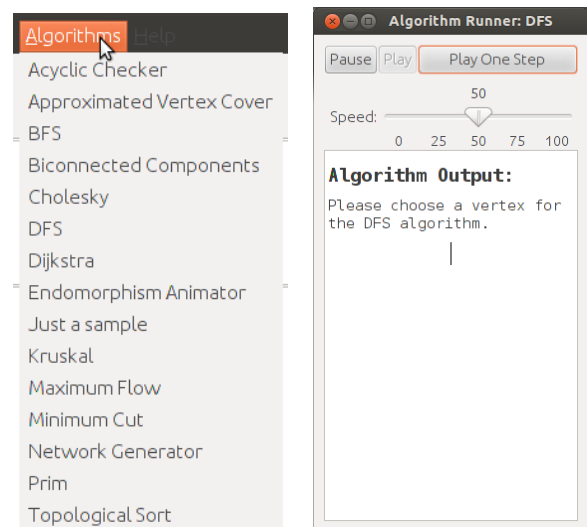


Fig. 4. (Left) List of algorithms accessible under the menu "Algorithms". (Right) The Algorithm Runner dialog.

graph with colors on it can be saved in alternative formats as an image or a graph.

As an example, Figure 5 illustrates the DFS algorithm running on a banana tree. In Figure 5(a), the algorithm asks for a vertex to start the algorithm. After choosing any vertex, the algorithm continues. The vertex is either visited or left in the DFS algorithm. GraphTea uses purple for leaving a vertex and green for visiting a vertex in this algorithm. Color utilization here makes a better representation of the algorithm details and helps to have easier visual perception. Figure 5(c) shows the state in which the algorithm is finished, i.e. all vertices are left.

V. CONCLUSION AND FUTURE WORKS

GraphTea is currently being developed as a rich graph-editing framework focused on educational aspects of graph theory and algorithms. The first version "GraphTea 1.0" is already released and is accessible under the address: *graph-theorysoftware.com*. The software provides the user the ability to design a teaching plan and to add instructions and to engage students more in the teaching process. Famous and widely used graphs, reports on them, and graph algorithms are already implemented in GraphTea. Furthermore, the suitable visualization of graph algorithms in addition to the flexible way of editing, are the power of GraphTea along with educational functionality.

There are other directions to extend GraphTea. Converting the current version to an online version, which can be accessed completely on a website would improve the usage flexibility. Moreover, returning back in algorithm steps could help the students to the better perception of the algorithm idea. Searching for the educational requirements in application areas, like wireless network visualization, social network, and scientific computing, in which there is an urgent need for such a tool in classroom.

ACKNOWLEDGMENT

We would like to thank you Prof. Amir Daneshgar for his continuous support and also the Mathematics department of Sharif University of Technology which provided us with a place and resources to start this project.

We thank you also all students were engaged in the development of the GraphLab, specially Roozbeh Ebrahimi, Reza Mohammadi, and Omid Aladini.

REFERENCES

- [1] Y. Carboneaux, J.-M. Laborde, and R. Madani, *CABRI-Graph: A tool for research and teaching in graph theory*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1996, vol. 1027.
- [2] A. Schliep and W. Hochstättler, “Developing Gato and CATBox with Python: Teaching graph algorithms through visualization and experimentation,” *Multimedia Tools for Communicating Mathematics*, pp. 291–310, 2002.
- [3] D. Auber, D. Archambault, R. Bourqui, A. Lambert, M. Mathiaut, P. Mary, M. Delest, J. Dubois, and G. Melançon, “The Tulip 3 Framework: A Scalable Software Library for Information Visualization Applications Based on Relational Data,” INRIA, Research Report RR-7860, Jan. 2012. [Online]. Available: <http://hal.archives-ouvertes.fr/hal-00659880>
- [4] A. Lambert and D. Auber, “Graph analysis and visualization with Tulip-Python,” in *EuroSciPy 2012 - 5th European meeting on Python in Science*, Bruxelles, Belgique, 2012-08. [Online]. Available: <http://hal.archives-ouvertes.fr/hal-00744969>
- [5] M. Lülfsmann, S. R. Lessenich, and H. M. Bücker, “Interactively exploring elimination ordering in symbolic sparse cholesky factorization,” *International Conference on Computational Science, ICCS 2010*, vol. 1, pp. 867–874, 2010.
- [6] H. Bücker, M. Rostami, and M. Lülfsmann, “An interactive educational module illustrating sparse matrix compression via graph coloring,” in *Interactive Collaborative Learning (ICL), 2013 International Conference on*, 2013, pp. 330–335.
- [7] M. T. Heath, *Scientific Computing: An Introductory Survey*, 2nd ed. McGraw-Hill, 2002.
- [8] R. Tamassia, *Handbook of Graph Drawing and Visualization (Discrete Mathematics and Its Applications)*. Chapman & Hall/CRC, 2007.
- [9] A. Bondy and U. Murty, *Graph Theory*, ser. Graduate Texts in Mathematics. Springer, 2008.

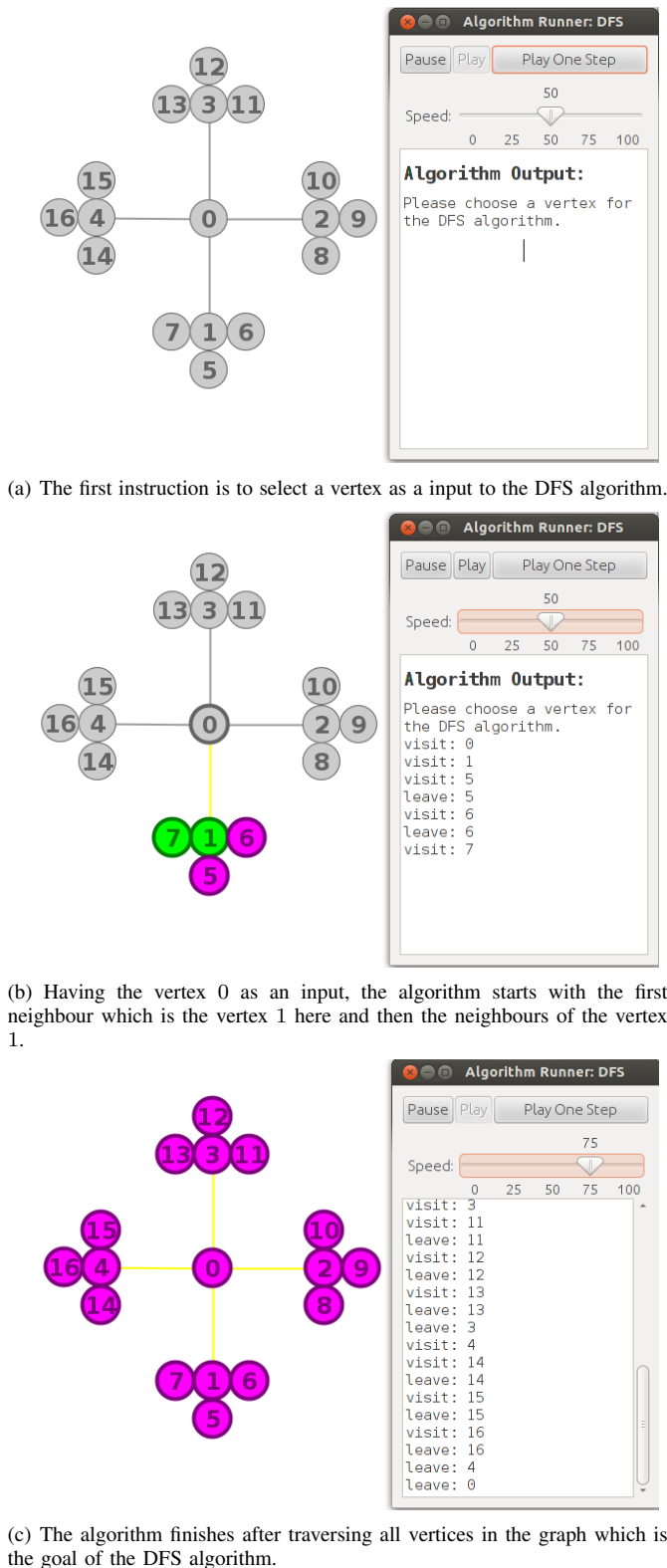


Fig. 5.