

Finding the shortest path in a graph and its visualization using C# and WPF

Radoslav Mavrevski¹, Metodi Traykov², Ivan Trenchev³

^{1,2}Department of Informatics, South-West University "Neofit Rilski", Bulgaria

²Department of Informatics, New Bulgarian University, Bulgaria

³Department of Electrical Engineering, Electronics and Automatics, South-West University "Neofit Rilski", Bulgaria

³University of Library Studies and Information Technologies Sofia, Bulgaria

Article Info

Article history:

Received Apr 9, 2019

Revised Nov 20, 2019

Accepted Dec 11, 2019

Keywords:

Dijkstra's algorithm

Visual Studio

Visualization

ABSTRACT

The shortest path problem is a classic problem in mathematics and computer science with applications in Economics (sequential decision making, analysis of social networks, etc.). The presented work is an example of realizing and applying of Dijkstra's algorithm to find the shortest path between two vertices in a connected, undirected graph, which is often a solved problem at a time annual International Olympiad in Informatics. For this purpose, are used the technologies, .NET 4.0, Visual Studio 2010, and WPF for the graphical user interface. The implemented program allows drawing an undirected graph, visualizing the shortest path between two vertices and finding its value. This software is a valuable tool for the study of Dijkstra's algorithm and is a great pedagogic instrument. All figures of path visualization included in this paper are actual screenshots of our visualization program.

Copyright © 2020 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Radoslav Mavrevski,
Department of Informatics,
South-West University "Neofit Rilski",
66 Ivan Michailov st., 2700 Blagoevgrad, Bulgaria.
Email: mavrevski@swu.bg

1. INTRODUCTION

The shortest path (SP) in the graph between two vertices is called a path that starts from one vertex and ends in the other, with the sum of the weights of the edges involved in it being minimal [1-3]. For some problems, the length of the path may be defined not as a sum, but as some other function of the weights of the edges involved in path (and even vertices). Since there may be no restriction the path to be simple, we must be careful when the graph contains a cycle. For example, if we are finding a minimal path, and in it there is a negative cycle (cycle with negative length), we will be able to "rotate" on this cycle any number of times, in which the length of the path (equal to the sum of the edges in it) will decrease arbitrarily much to minus Infinity. Similarly, if we find a maximum path and there is a positive cycle, for each path that contains it, we will be able to "rotate" on this cycle whereby we get an arbitrary great length [1]

The most efficient method of finding a shortest path from a particular vertex to everyone else is Dijkstra's algorithm. Without applying optimization, this algorithm has a complexity of the order of $O(N^2)$, with N is the number of vertices in a given graph [1-3]. Although it is the most efficient algorithm for finding the shortest path, this algorithm is not applicable to every graph. The graphs in which used are not to have negative cycles. In this work, we finding the shortest paths in undirected graph without cycles.

In the present work is presented .NET implementation of Dijkstra's algorithm to find the shortest path between two vertices in a connected undirected graph. For this purpose, has been used the technology .NET 4.0, Visual Studio 2010 and Windows Presentation Foundation (WPF) for the graphical user interface.

One vertex of a graph in this particular development has several important properties – a label (number) that distinguishes it from all other vertices, a set of edges that connects it to other vertices, a property that will show whether the vertex has been visited or not, its coordinates, which will be used in its graphical representation, and the total value (length) of the found path.

2. MATERIAL AND METHOD

2.1. Dijkstra's algorithm

In this algorithm are assumed, that the lengths c_{ij} of all edges to be nonnegative. The algorithm can be considered as the process of sequentially marking the vertices of a graph with corresponding numbers. Generally, the marking number $d(x)$ at the vertex x is temporary and gives the upper limit of the path length from s to x . When performing the algorithm, the values of the markup numbers are reduced, with at each step exactly one of the temporary marker numbers being constant (coloring) [1]. In this case, the permanent (colored) marker number $d(x)$ is no longer an upper limit, but the exact length of the SP from s to x . For colored we also consider the corresponding vertex x . When in addition to the length of the SP the path itself is searching is also colored and one of the edges of the graph, thus it is included in the search path.

2.2. .NET 4.0

The Microsoft .NET Framework is a platform created by Microsoft that provides a programming model, a library of classes, the Framework Class Library (FCL), and an environment for the implementation of written specifically for it program code, Common Language Runtime (CLR). It is a key element of Microsoft's development strategy, which aims to make most new Windows applications based on a .NET Framework 3.5. .NET applications are written in high level languages (C#, VB.NET, C++ / CLI, etc.) and compiled into a platform-independent intermediate language called Common Intermediate Language (CIL). The C# language is used in this work. During execution, the CIL code is automatically compiled by CLR for the specific hardware platform and operating system the user is use. The .NET Framework architecture is often considered at levels as shown in Figure 1.

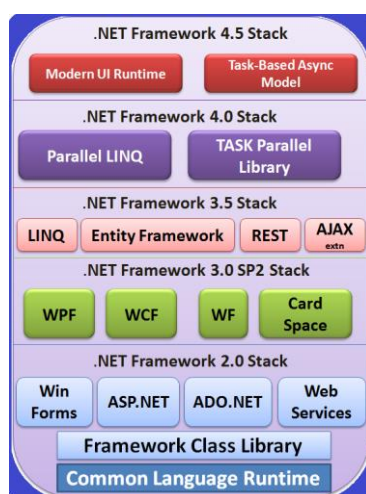


Figure 1. The architecture of the .NET framework

2.3. Visual Studio 2010

VS.NET is one of the world's leading integrated application development environments - Integrated Development Environment (IDE). With its help, we can do each of the typical tasks related to building an application-writing code, creating user interface, compiling, running and testing, debugging, error tracking, creating installation packages, viewing documentation and others [4-8]. The Visual Studio .NET 2010 package supports Microsoft C# .NET, Microsoft Visual Basic .NET, Microsoft C++ .NET (managed/unmanaged) and Microsoft Visual J# programming languages as standard. In the present work, program for the finding and visualization of SP is written in the C# programming language. The VS.NET code editor supports all approved advanced source code editors functions-syntax coloring for easier visualization of the code and error reduction, automatic completion of the sentence, automatic extraction of auxiliary data during writing and many more. Also, IntelliSense functionality for selecting classes, methods, and variables is supported. It provides great ease for the incoming .NET programmers, as it allows them to look at

the possibilities and choose from the list the one that interests them. This saves effort, the time to write the name, and reduces the likelihood of unpleasant "wrong" errors. Figure 2 gives a visual view of the Visual Studio .NET 2010 Code Editor.

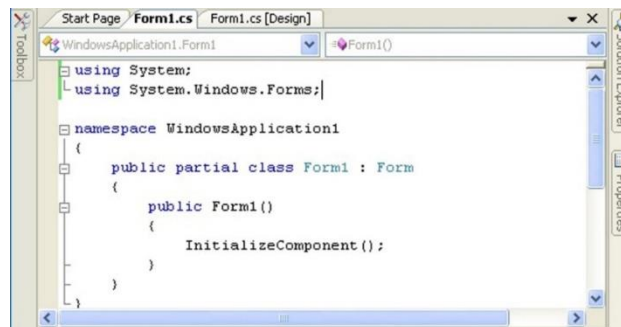


Figure 2. The view of the visual studio .NET 2010 code editor

2.4. Windows presentation foundation (WPF)

Windows Presentation Foundation is an entirely new graphical system for building modern desktop applications that provides both developers and designers with a unified platform for creating rich media applications based on XAML language. WPF is designed for the .NET platform, influenced by modern HTML and Flash technologies, and hardware accelerated. WPF is a new generation platform designed to facilitate the creation of a graphical user interface for desktop applications. In this development, the WPF was used to implement the graphical user interface of the program located and visualizing the SP.

3. RESULTS

The implemented software program allows drawing an undirected graph using the mouse. Drawing the vertex of a graph is done with a click of the mouse, and the setting of the edge and the corresponding weight is done by sequentially choosing the two vertices between which we set the edge by clicking with mouse on them. The general view of the SP finding program is shown in Figure 3. To switch from a plotting mode to a SP search mode, the program has a special button "SP finding". In this mode, the program allows us to select the starting and ending edges, between which we are searching SP. The program finds the SP between the two vertices of the Dijkstra's algorithm by visualizing it and coloring the vertices and edges involved in it in green color and find value of the path.

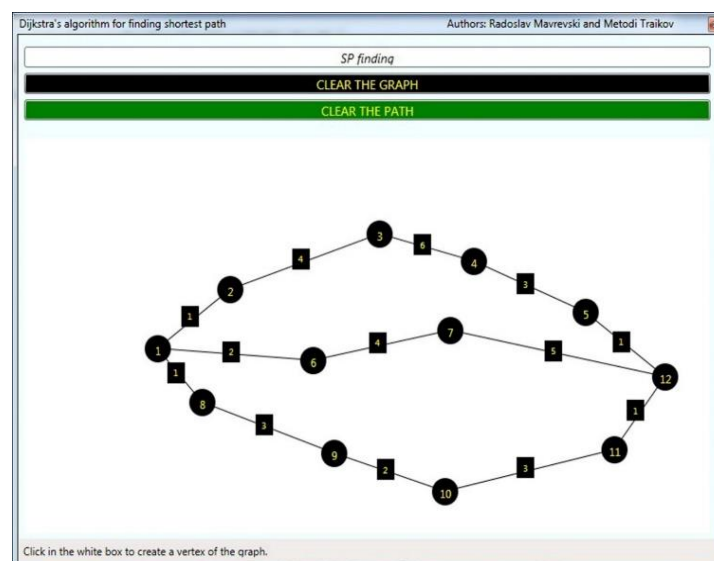


Figure 3. The general view of the SP finding program

The result of using the program to find the SP in a given graph is shown in Figure 4 and Figure 5. For example, for Figure 4 the graph has 15 vertices and 16 edges. The SP between vertex 1 and vertex 15 has a length of 14. The developed software program for calculation of SP by Dijkstra's algorithm, significantly facilitates the application of this algorithm on any connected, undirected graph and allows for easier interpretation of the result. This software program may be used as an educational software can give a big clarity, regarding the algorithm and programming [7-18].

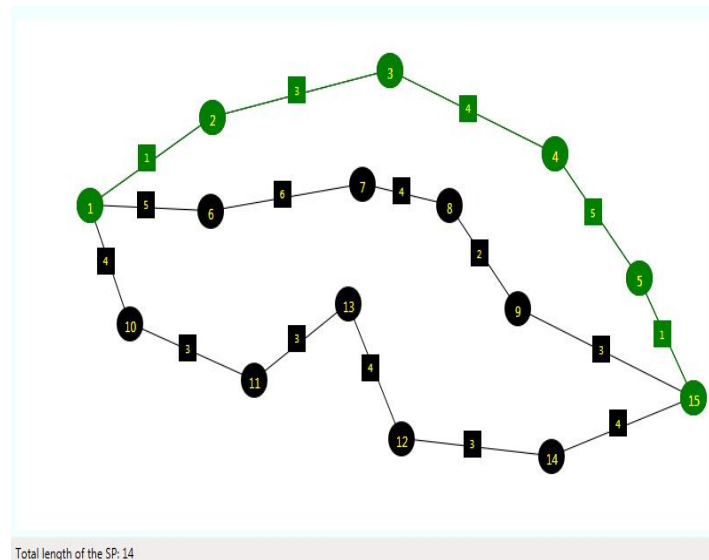


Figure 4. Result from using the SP finding program: SP14

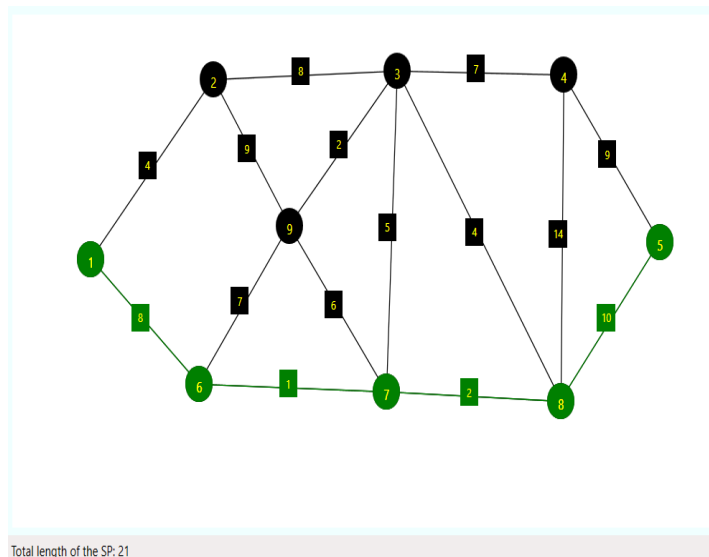


Figure 5. Result from using the SP finding program: SP21

4. CONCLUSION

Very often a wealth different informatics Olympiads are held national, regional or international levels, in which are decided problems that requiring programming knowledge and various algorithm techniques. The creating high quality problem solution can be a difficult and time-consuming process [19-25]. In this paper we explored known technique for find the shortest path, with the help of the powerful programming language C# and modern technologies such as WPF, which can be used to find new ideas for problems and refine these ideas into problems suitable for the International Olympiad in

Informatics (IOI) annually organized. In addition this article can help students in secondary and higher schools to acquire additional skills in programming and to learn more about algorithms and their application.

ACKNOWLEDGEMENTS

This paper is partially supported by the National Scientific Program "Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICTinSES)", financed by the Ministry of Education and Science. This paper is partially supported by the French-Bulgarian contract "RILA", 01/4, 2018, Project title: Integer Programming Approaches for Long-Read Genome Sequencing and Assembly.

REFERENCES

- [1] N. Moschovakis, "What is an algorithm?". In *Engquist, B.; Schmid, W. Mathematics Unlimited – 2001 and beyond*. Springer. pp. 919–936 (Part II), 2001, ISBN 9783540669135.
- [2] C. Jing-Chao, "Dijkstra's Shortest Path Algorithm", *Journal of formalized mathematics*, 15, 2003.
- [3] T. Carmen et al., "Introduction to Algorithms" (3rd ed.), London, MIT Press and McGraw-Hill, 2009.
- [4] G Arora. et al., "Microsoft C# Professional Projects", Portland, United States, Premier Press, 2002.
- [5] P. Krishna, "Announcing the .NET Framework 4.7.1". .NET Blog. Retrieved 17 October, 2017.
- [6] E. Börger et al., "A high-level modular definition of the semantics of C#" *Theoretical Computer Science* 336(2-3), 235–284, 2005.
- [7] R. Mavrevski and M. Traykov, "Visualization software for hydrophobic-polar protein folding model," *Scientific Visualization*, 11(1), 11-19, 2019, DOI: 10.26583/sv.11.1.02. ISSN: 2079-3537.
- [8] R. Mavrevski et al. "Interactive approach to learning of sorting algorithms," *International Journal of Online and Biomedical Engineering*, 15(8), 120-134., 2019, eISSN: 2626-8493.
- [9] M. Traykov et al. "Modeling of digital converter for GSM signals with MATLAB" *International Journal of Electrical and Computer Engineering*, 9(5), 4417-4422, 2019, DOI: 10.11591/ijece.v9i5.pp4417-4422. ISSN: 2088-8708.
- [10] R. Mavrevski et al., "Multitasking in Embedded System Designs," *International Journal of Online and Biomedical Engineering*, 15(9), 44-57, 2019, eISSN: 2626-8493.
- [11] M. Valinataj, et al., "Fault-aware and reconfigurable routing algorithms for Networks-on-Chip," *IETE Journal of Research*, vol. 57, no. 3, pp. 215-223, 2011.
- [12] T. Lehtonen, D. Wolpert, P. Liljeberg, J. Plosila, and P. Ampadu, "Self-adaptive system for addressing permanent errors in on-chip interconnects," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 18, no. 4, pp. 527-540, Apr. 2010.
- [13] I. Kostadinova, et al., "Model For Building Competencies Through An Electronic System For Assessment Of Knowledge," *10th International Conference Of Education, Research And Innovation (ICERI2017)*, Edited by: Chova, LG; Martinez, AL; Torres, IC, Book Series: ICERI Proceedings, Pages: 430-436, Seville, SPAIN, NOV 16-18, 2017, ISBN: 978-84-697-6957-7 / ISSN: 2340-1095.
- [14] I. Kostadinova, et al., "Residence And The Social Environment As A Factor Affecting Digital Skills And Competence In Bulgaria. Steps And Methods To Overcome The Problem," *Conference: ICERI 2016*, October 2016, ISBN: 978-84-617-5895-1, ISSN: 2340-1095, doi: 10.21125/iceri.2016.
- [15] Hristov et al., "The blockchain as a backbone of GDPR compliant frameworks," *8th International Multidisciplinary Symposium. SIMPRO*, 56-61, 2018.
- [16] S. Denchev and T. Trencheva, "Intellectual Property as a Basic Part of the University's Information Literacy", *Conference Proceedings: 2nd International Conference on Education and Management Science (ICEMS 2016)*, 28-29 May 2016, Beijing, China, DEStech Publications, China, pp. 74-78, 2016.
- [17] E. Zdravkova, "Media Literacy as a Key Competence for the Safe and Effective use of Media", *ICERI 2019 Proceedings*, 7467-7473, 2019.
- [18] A. W. Yin et al, "Hierarchical agent monitoring NoCs: a design methodology with scalability and variability," *Proc. 26th NORCHIP Conf*, pp. 202-207, 2008.
- [19] S. Dimitrova, "Distance Learning in Bulgarian Universities - Step forward in the Modern Education," *ICERI2019 Proceedings*, 8127-8132, 2019.
- [20] Trencheva, T., et al., "Intellectual Property Training of Library and Information Management Bachelor's Students", *Information Literacy: Key to an Inclusive Society: 5th European Conference on Information Literacy (ECIL): Revised Selected Papers*. Eds. by S. Kurbanoğlu et. al., ECIL, Saint Malo, 2017, Springer-Verlag, Heidelberg, 2017, 294-302 - (Communication in Computer and Information Science, Vol. 810).
- [21] W. Dimitrov, et al., "Standards And Interoperability Within Smart City Ecosystem", *Conference: 10th annual International Conference of Education, Research and Innovation*, November 2017, 4346-4355, 2017, 10.21125/iceri.2017.1157.
- [22] G. Siva et al., "Dynamic Time Slice Calculation for Round Robin Process Scheduling Using NOC," *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 5, No. 6, pp. 1480-1485, 2015.

- [23] A. Brata, et al, "Software Development of Automatic Data Collector for Bus Route Planning System," *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 5, No. 1, pp. 150-157, 2015.
- [24] A. Kohler, G. Schley, and M. Radetzki, "Fault tolerant network on chip switching with graceful performance degradation," *IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no.6, 2010.
- [25] S. Kendaganna, et al., "Platform level design for Network on Chips", *2015 IEEE International Advance Computing Conference (IACC)*, Bangalore, pp. 16-19 2015.

BIOGRAPHIES OF AUTHORS



Radoslav Mavrevski, associate professor in Department of Informatics, Faculty of Mathematics and Natural Sciences, member of University Center for Advanced, Bioinformatics Research, South-West University "Neofit Rilski", 66 Ivan Mihaylov Str., Blagoevgrad, Bulgaria. PhD on Informatics. Scientific Interest: programming, computer modelling, applied statistics and bioinformatics. He is one of the organizers of the South Eastern European Mathematical Olympiad for University Students (SEEMOUS) with International Participation, 2012, <http://seemous2012.swu.bg/> and XXVII REPUBLICAN STUDENT PROGRAMMING OLYMPIAD, 2015, <http://bcpc.eu/XXVII/>.



Metodi Traykov, assistant in Department of Informatics in New Bulgarian University, Sofia, Bulgaria. Also assistant in Department of Informatics, Faculty of Mathematics and Natural Sciences, member of University Center for Advanced, Bioinformatics Research, South-West University "Neofit Rilski", 66 Ivan Mihaylov Str., Blagoevgrad, Bulgaria. PhD on Informatics. Scientific Interest: programming and bioinformatics. He is one of the organizers of the XXVII REPUBLICAN STUDENT PROGRAMMING OLYMPIAD, 2015, <http://bcpc.eu/XXVII/>.



Ivan Trenchev, associate professor in Department of Electrical Engineering, Electronics and Automatics, Faculty of Engineering, member of University Center for Advanced, Bioinformatics Research, South-West University "Neofit Rilski", 66 Ivan Mihaylov Str., Blagoevgrad, Bulgaria and Associate professor in University of Library Studies and Information Technologies Sofia, Bulgaria. PhD on Informatics. Scientific Interest: virtual reality (VR), computer modelling and bioinformatics. He is one of the organizers of the XXVII REPUBLICAN STUDENT PROGRAMMING OLYMPIAD, 2015, <http://bcpc.eu/XXVII/>.