AMERICAN Journal of Engineering, Mechanics and Architecture
Volume 3, Issue 1, 2025 ISSN (E): 2993-2637

Method for Determining Non-Isomorphism of Graphs

Urakova Dilnoza Karimovna
Navoi Architectural and Construction Technical School

Abstract: Among the numerous examples of application areas of algorithms for solving the problem of determining graph isomorphism, we note the problem of syntactic and structural pattern recognition, some problems of mathematical chemistry and chemoinformatics (study of the molecular structures of chemical compounds), problems related to the study of social networks (for example, linking several accounts of one user on Facebook).

Keywords: Graph isomorphism, non -isomorphic graphs, and graph invariants, and isomorphism testing algorithms, adjacency matrix, incidence matrix, spectral analysis of graphs, graph theory.

The presented work considers the problem of checking graph isomorphism and various approaches to its solution. The isomorphism relation between two graphs (undirected and without vertex and edge weights) is a bijection between the sets of graph vertices that preserves vertex adjacency. If the isomorphism relation holds between two graphs, then such graphs are called isomorphic. For undirected and directed graphs, the problems of determining isomorphism are practically identical. When determining isomorphism for directed or weighted graphs, additional restrictions are imposed on preserving the values of weights and arc orientations.

An example of one of the possible algorithms for determining the isomorphism of two directed graphs is described in the article. In this algorithm, using the graph distance matrix (a matrix in which each element represents the length of the shortest path between two graph vertices), the search tree for possible correspondences between vertices is limited when determining the isomorphism of two directed graphs.

It is easy to show that the isomorphism relation between graphs is reflexive, symmetric, and transitive, i.e., it is an equivalence relation. Therefore, the class of all graphs can be divided into nonempty and pairwise disjoint subclasses, called isomorphism classes or classes of isomorphic graphs. Two arbitrary graphs belong to the same isomorphism class if and only if they are isomorphic to each other. On In practice, for most cases, partitioning graphs into isomorphism classes is an unsolvable problem.

The isomorphism testing problem has wide practical application and is an important problem in algorithm complexity theory. This problem belongs to the class IR, but it is unknown whether it belongs to the class P - if we assume that P ^ NR. At present, it is unknown whether this problem is NR-complete [9], but, for example, it is known that the problem of finding an isomorphic subgraph in a graph is NR-complete (the input data for this problem are graphs O and H, it is required to determine whether graph O contains a subgraph isomorphic to graph H) [12]. Thus, the studies currently being conducted that are aimed at solving the isomorphism testing problem for both arbitrary graphs and graphs of a special type are relevant (in practice, both exact and heuristic algorithms can be used for such studies, examples of which are given in Chapter 1).

To solve many practical problems, it is often necessary to show that the graphs under consideration are not isomorphic. This makes it possible to cut off obviously non-isomorphic graphs in the set of graphs under consideration. The problem of checking the non-isomorphicity of graphs, studied in the presented work, can be considered equivalent to the problem of checking the isomorphism of graphs. It is just as relevant and has just as wide practical application.

Level of development

One of the common approaches to the problem of checking the isomorphism of graphs is the use of heuristics. Heuristics for solving the isomorphism problem usually consist of attempts to show that the graphs in question are not isomorphic [57]. To do this, a list of different invariants is compiled in an order usually determined by the complexity of calculating this invariant. Then the values of the parameters of the presented graphs are sequentially compared. If two different values of the same parameter are found, it is concluded that the presented graphs are not isomorphic. Such an algorithm for establishing the isomorphism of two

graphs is called heuristic. An example of a heuristic algorithm for checking graph isomorphism is given in Chapter 1. The approach described in this paper can be considered as an improved version of the heuristic algorithm.

Another area of research is the solution of the problem of determining isomorphism for certain classes of graphs. It is currently unknown whether the problem of checking graph isomorphism is solvable in polynomial time [11, 31], but it is known that this problem can be solved in polynomial time for some classes of graphs. For

- planar graphs;
- graphs with limited degree of vertices;
- graphs with limited multiplicity of eigenvalues from the spectrum of the adjacency matrix as well as some other classes, efficient algorithms for solving this problem are known.

These algorithms exploit specific structural characteristics of graphs, which limits their scope of application. Therefore, there is a need for an algorithm that would find a solution to the graph isomorphism problem for as wide a class of graphs as possible, while remaining polynomial in both time and memory. An example of an isomorphism testing algorithm for a class of graphs defined by spectral characteristics is given in Chapter 1 .

More recently, at the end of 2015, the famous mathematician Laszlo Babai presented a new fast algorithm for solving the graph isomorphism problem [17, 55]. The proposed algorithm allows one to establish the isomorphism of two graphs in a smaller number of steps (compared to the methods currently used). Laszlo Babai promises to publish a more detailed description of this algorithm in the near future. If its successful operation is confirmed, this algorithm will allow one to more effectively operate with a large array of data.

related to natural sciences. And it may also contribute to the revision of the principles of data encryption, since the process of decrypting data encrypted with factors (when instead of a certain number or group of numbers, when transmitting information, the factors of this number or group of numbers are transmitted) will be significantly simplified.

The purpose and objectives of the study

The object of study of the presented work is graphs, as well as graph characteristics that are their invariants.

The subject of the research is algorithms for calculating graph invariants, as well as graph generation algorithms.

The main objective of the work is to develop a method for solving the problem of checking the non-isomorphism of graphs and its study based on the use of random graph generation algorithms.

As the main result of the work, we consider obtaining algorithms for checking the non-isomorphism of graphs, which are some sequences of comparison of the values of invariants. To test various algorithms for checking the non-isomorphism of graphs, sets of input data are needed. In most real situations, the storage of input data is limited by the size of the system memory. One of the methods that allows solving this problem is random data generation. In this paper, it is assumed that for the discrete optimization problem under consideration, the optimal algorithm for solving it depends on the method of generating input data. To conduct computational experiments, it is assumed to use a set of graphs obtained using certain generation algorithms (including graph generation algorithms developed by the author).

Main objectives of the research.

1. Develop a method for solving the problem of checking the non-isomorphism of graphs based on the selected assumption about the applied generation algorithm.
2. Describe the new graph invariant introduced by the author - the second-order degree vector.
3. Develop an algorithm for generating graphs from a given vector of first-order degrees using the branch and bound method with additional heuristics.
4. Develop an algorithm for generating graphs from a given vector of second-first-order degrees using the branch and bound method with additional heuristics.
5. Develop a software system for organizing computational experiments to evaluate the effectiveness of using various algorithms that represent certain sequences of invariants for determining the non-isomorphism of graphs.

References

1. Abrosimov, M. Practical tasks on graphs, 2nd edition: Textbook / M. B. Abrosimov, A. A. Dolgov. - Saratov: Publishing house "Scientific book", 2009. - 76 p.
2. Baumgertner, S. Additional heuristics in the problem of star-height minimization of a nondeterministic finite automaton / S. Baumgertner // Vector of Science of Togliatti State University. - 2010. - No. 3 (13). - P. 37-39.
3. Balinova, V. Statistics in questions and answers / V. Balinova. - M.: TK Velbi, Prospect Publishing House, 2004. - 344 p.
4. Belsky, A. Graph Theory and Combinatorics [Electronic resource]. - Access mode: http://belsky. narod.ru/v2/rus/mathemat/tgik.html (16.02.2015).
5. Bolshakova, E. I. Algorithms for constructing a computer dictionary of Russian letter paronyms and its application / E. I. Bolshakova, I. A. Bolshakov // Heuristic algorithms and distributed computing. - 2015. - Vol. 2, No. 3. - P. 8-22.
6. Breer, V. V. Stochastic models of social networks / V. V. Breer // Management of large systems. - 2009. - No. 27. - P. 169-204.
7. Bryuske, E. Ya. To a chemist about graph theory: graphs in chemical nomenclature / E. Ya. Bryuske // Bulletin of Tambov University. Series: Natural and technical sciences. - 2003. - Vol. 8, issue 5. - P. 840-847.
8. Calculation of the determinant by the Kraut method [Electronic resource]. - Access mode: http: //e-maxx.ru/al go/determinant crout (12.04.2016).

9. Gromkovich, Yu. Theoretical computer science. Introduction to automata theory, computability theory, complexity theory, theory algorithms, randomization, communication theory and cryptography / Yu. Gromkovich. - SPb.: BHV-Petersburg, 2010. - 336 p.
10. Goodman, S. Introduction to the Development and Analysis of Algorithms / S. Goodman, S. Hidetniemi. - M.: Mir, 1981. - 368 p.
11. Gary, M. Computing machines and intractable problems / M. Gary, D. Johnson. - M.: Mir, 1982. - 416 p.
12. Zykov, A. Fundamentals of graph theory / A. Zykov. - M.: Nauka, 1986. - 384 pp.
13. Kirsanov, M. N. Graphs in Maple / M. N. Kirsanov. - M.: Fizmatlit, 2007. - 168 p.
14. Cormen, T. Algorithms - construction and analysis / T. Cormen, C. Leiserson, R. Rivest, K. Stein. - M.: Williams, 2005. - 1296 p.
15. Levitin, A. Algorithms: Introduction to Development and Analysis / A. Levitin. - M.: "Williams", 2006. - 576 p.
16. Mainika, E. Optimization Algorithms on Networks and Graphs / E. Mainika. - M.: Mir, 1981. - 323 p.
17. Materials from the portal "Scientific Russia". Scientific algorithm promises to simplify the problem of "graph isomorphism". [Electronic resource]. - Access mode: http://scientificrussia.ru/articles/izomorfizm-grafov (08.09.2017)
18. Melnikov, B. F. Algorithm for checking the equality of infinite iterations of finite languages / B. F. Melnikov // Bulletin of Moscow University. Ser. Comput. math. and kib-ka. - 1996. - No. 4. - P. 49-54.
19. Melnikov, B. F. Multiheuristic approach to discrete optimization problems / B. F. Melnikov // Cybernetics and systems analysis (NAS of Ukraine). - 2006. - No. 3. - P. 32-42.
20. Melnikov, B. F. Subclasses of the class of context-free languages / B. F. Melnikov. - M.: Moscow State University, 1995. - 174 p.
21. Melnikov, B. F. Application of algorithms for generating random graphs for studying the reliability of communication networks / B. F. Melnikov, E. F. Saifullina, Yu. Yu. Terentyeva, N. N. Churikova // Informatization and communication. - 2018. - No. 1. - P. 71-80.
22. Melnikov, B. F. Application of a multi-heuristic approach for random generation of a graph with a given degree vector / B. F. Melnikov, E. F. Saifullina // News of higher educational institutions. Volga region. Physical and mathematical sciences. - 2013. - No. 3 (27). - P. 69-82.
23. Melnikov, B. Heuristics in programming nondeterministic games / B. Melnikov // Programming. Bulletin of the Russian Academy of Sciences. - 2001. - No. 5. - P. 63-80.
24. Melnikova, E. A. Approach to isomorphism verification using invariant construction / E. A. Melnikova, E. F. Saifullina // Vector of Science of Togliatti State University. - 2013. - No. 1(23). - P. 113-120.
25. Melnikova, E. A. Application of various graph invariants to checking the isomorphism of some types of graphs / E. A. Melnikova, E. F. Saifullina // Problems of informatics in education, management, economics and engineering: tr. XII Int. scientific and technical. conf. - Penza: Privolzhsky House of Knowledge, 2012. - P. 40-42.
26. Molodtsov, S. G. Generation of molecular graphs with given structural constraints: dis. ... Cand. of Phys. and Mathematics: 05.13.16: defended 21.10.1997 / Molodtsov Sergey Georgievich. - Novosibirsk: NIOKh SB RAS, 1997. - 79 p.