Graph isomorphism recognition using scheme matrices

M Kh Dudov
Karachay-Cherkess State Technological Academy, 36, Stavropol st., Cherkessk, Karachay-Cherkess Republic, 369000, Russia
E-mail: dudovm@mail.ru

Abstract. The issue of isomorphism graphs recognition is as follows: whether there is a polynomial algorithm that recognizes isomorphism of each pair of graphs. The issue of isomorphism recognition is one of the most important in the graph theory and belongs to the class of NP-complete problems which includes problems for which polynomial algorithms are unknown. Despite numerous attempts, the problem of verifying the isomorphism of graphs belongs to those ones that cannot be classified according to their complexity. The issue of computational complexity of the problem remains open, although for some special cases efficient algorithms with polynomial computational complexity have been constructed. The paper develops and justifies a graph isomorphism recognition method whose computational complexity is polynomial.

1. Introduction
The issue of isomorphism recognition is one of the most important in the graph theory and belongs to the class of NP-complete problems which includes problems for which polynomial algorithms are unknown. The problem of analysis and synthesis of fault-tolerant systems includes many problems of the graph theory. It is important to recognize graph isomorphism of the degrading system to the finite number of subgraphs with parameters sufficient to maintain system’s performance [1].

The need to verify isomorphism of graphs arises when solving problems of chemo-informatics, mathematical chemistry [2], electronic circuit design automation, program optimization, robotic recognition, etc. Some applied problems whose solution is based on the isomorphic graph recognition algorithms are as follows:

- CAD-based control automation;
- integrity control automation for the structure of various objects, in particular, databases;
- analysis and management of system survivability;
- analysis and synthesis of computing systems;
- research and identification of molecular structures of chemical compounds;
- recognition of isomorphism of Boolean functions.

A lot of theoretical and applied researches deal with development and research of methods and algorithms for recognition of graph isomorphism [3-6]. The problem of graph isomorphism recognition is the main procedure or subtask for complex and applied recognition of isomorphic graph embedding.

Algorithms for recognition of isomorphism and isomorphic graph embedding are used when solving the following applied problems [1, 4, 5]:

- development of information retrieval systems and logical inference systems based on comparison with samples;
- organization of a logical conclusion based on the semantic network;
• syntactic and structural pattern recognition;
• automatic analysis of documents;
• recognition of similarity of cognitive maps and identification of adequacy of the developed models of complex systems;
• optimization of computing in the network of processors;
• identification of the structure of discrete systems, in particular, logical networks and Boolean functions;
• study of the topology of computer networks and multiprocessor systems;
• study of molecular structures of chemical compounds.

These examples show a variety of applications of the isomorphism recognition algorithms.

The problem of graph isomorphism aims at finding a polynomial algorithm that recognizes isomorphism of each pair of graphs.

No polynomial isomorphism recognition algorithm has been developed. Moreover, it is not known whether this algorithm can be developed.

2. Problem statement

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are graphs. It is required to determine whether $G_1$ and $G_2$ are isomorphic, i.e. if there is function $f : V_1 \rightarrow V_2$ such that $(u, v) \in E_1$ if and only if $(f(u), f(v)) \in E_2$ [7-5].

The problem of graph isomorphism recognition is one of the few classical problems of the graph theory for which it was not possible to construct a polynomial algorithm. Only for some special classes of graphs it was possible to construct polynomial algorithms [3-6]. This fact supports researchers’ optimism in the search for polynomial algorithms.

The main method is construction of non-choice algorithms which recursively improve their efficiency in terms of completeness or sensitivity of the graph characteristics which are invariant under the isomorphism of graphs and are referred to as invariants. Since the graph invariant does not change its values on isomorphic graphs, equality of the invariants is a requirement for the graph isomorphism.

Examples of graph invariants can be the number of vertices, the number of edges, the vector of local degrees of the vertices, the number of cycles of a given length in the graph, etc. The most important graph invariants are: density - the number of vertices of the largest complete subgraph; leakiness - the largest number of pairwise non-adjacent vertices of the graph, the chromatic number of the graph, the number of components of the graph; the Hadwiger number - the number of vertices of the largest complete graph to which the original graph can be drawn. Along with these invariants, it is possible to consider a system of numbers. For example, as an invariant, one can choose a vector of values of invariants. All these invariants do not guarantee graph isomorphism, i.e. they are not complete.

Examples of the full graph invariant are maximum and minimum binary codes of the adjacency matrices of this graph. To obtain a binary code by the adjacency matrix of the graph, let us number its elements in a row from left to right and from top to bottom. Then this matrix can be considered as a bitwise binary number for a graph. The binary codes of all adjacency matrices that are isomorphic to a given one are generally different. The smallest code is a minimum code of the graph, and the largest one is the maximum code. For any of these codes and the number of graph vertices, one of its adjacency matrices and the graph itself can be reconstructed. However, calculation of the minimum or maximum binary code of the adjacency matrix for a given vertex graph is as difficult as the complete set $n$ of matches of two graph vertices, since you have to choose the smallest (or largest) of the binary codes of all adjacency matrices of the given graph.

Efforts of many researchers were directed to the search for such an invariant graph, which would be easily computed from a given graph and complete. A number of papers [1, 4, 8] are devoted to the study of the completeness of graph invariants and development of new systems of invariants for solving the problem.

Any invariant of a graph is either not complete, or requires an inefficient procedure for its calculation.

The issue of computational complexity of the problem remains open, although for some particular cases efficient algorithms with polynomial computational complexity have been constructed [9–13]. The method for isomorphic graphs recognition with computational complexity of no more $O(n^3)$ is developed and justified below.
3. Schematic graph matrices
Let \( S \) be the adjacency matrix of graph \( G \).

Let us construct matrix \( Y = \|y_{ij}\|, i, j = 1,n \) such that
\[
\begin{align*}
    y_{ij} &= -s_{ij}, i \neq j, \\
    y_{ii} &= \text{deg} \; v_i + 1
\end{align*}
\]

Where \( \text{deg} \; v_i \) is the degree of the \( i \)-th vertex of graph \( G \).

Matrix \( Y \) is referred to as the schematic matrix of graph \( G \) [14]. It is known that circuit matrices are symmetric and, in contrast to adjacency matrices, are always non-singular [15].

Let \( A, B \) be the schematic matrices of graphs \( G_1 \) and \( G_2 \), respectively.

For each graph, we find the solution of \( n \) systems of linear algebraic equations
\[
A \cdot U_i = D_i, \\
B \cdot U_i = D_i,
\]
where \( D_i \) is a vector whose components, except for the \( i \)-th one, are equal to zero, and \( d_i = 1 \).

Let us denote elements of matrices \( M_1 \) and \( M_2 \) by \( m_{ij}^{(1)} \) and \( m_{ij}^{(2)} \). Then according to the Kramer’s formula:
\[
m_{ij}^{(1)} = A_{ij} / \text{det}(A); \quad m_{ij}^{(2)} = B_{ij} / \text{det}(B),
\]
where \( A_{ij} \), \( B_{ij} \) are the minors of order \( (n-1) \) corresponding to the elements of matrices \( A \) and \( B \);
\[
\text{det}(A), \text{det}(B) \]
are determinants of matrices \( A \) and \( B \).

Obviously, the necessary condition for the graphs to be isomorphic is equality of the determinants of the diagram matrices of graphs. Indeed, when rearranging any two matrix lines or columns, the determinant changes the sign, therefore simultaneous permutation of an equal number of lines and columns does not change the value of the determinant.

4. Sufficient condition for equality of schematic matrices
Let us denote the elements of the main diagonal of the matrices \( A \) and \( B \) by \( a_{ii} \) and \( b_{ii} \), \( A_{ii} \), \( B_{ii} \) are the corresponding principal minors, and \( A_{ij} \), \( B_{ij} \) are the minors of order \( (n-1) \) corresponding to the elements of the \( i \)-x columns of matrices \( A \) and \( B \). Let \( \ell = 1 \).

Statement 1. If \( a_{ii} = b_{ii}, \ A_{ii} = B_{ii} \) and \( A_{ij} = B_{ij} \) for all \( i = 1, n \), \( A = B \).

Let us prove this statement by induction for \( n \).
For \( n = 1 \) the validity of the statement is obvious.

Let \( n = 2 \), then the condition \( A_{ii} = B_{ii} \) and \( A_{ij} = B_{ij} \) for all \( i = 1, n \) can be written as:
\[
A_{11} = B_{11} \Rightarrow a_{22} = b_{22}; \\
A_{22} = B_{22} \Rightarrow a_{11} = b_{11}; \\
A_{21} = B_{21} \Rightarrow a_{12} = b_{12}.
\]

Matrices \( A \) and \( B \) symmetric, that is \( a_{21} = a_{12} \), \( b_{21} = b_{12} \), therefore, \( A = B \).

Let us show that from the validity of statement \( n = k \) follows the validity of statement \( n = k + 1 \).
Each of the minors that meet the requirement of the statement can be represented as the sum of two determinants that differ from the minor only by the last \( (k+1) \) columns so that the \( (k+1) \)-th column of the minor is equal to the sum of the \( (k+1) \) x columns of the determinants.

For example, the minor of the \( a_{11} \) matrix element \( A^{(k+1)} \) (at \( n = k + 1 \)) can be represented as:
\[
A^{(k+1)}_{11} = A^{(k+1)}_{11} + A^{(k+1)}_{11},
\]
where \( A'_1^{(k+1)} \) differs from \( A_1^{(k+1)} \) only by the last \((k+1)\) column, all elements of which, except for the diagonal, are equal to «0», and the diagonal is equal to \( a_{(k+1)(k+1)} \); and \( A''_1^{(k+1)} \) differs from \( A_1^{(k+1)} \) only by the diagonal element of the \((k+1)\)-th column equal to «0».

Then, expanding \( A'_1^{(k+1)} \) by the \((k+1)\)th column, we get \( A'_1^{(k+1)} = a_{(k+1)(k+1)} \cdot A_1^{(k)} \), where \( A_1^{(k)} \) is the minor element \( a_i \) of the matrix \( A^{(k)} \) (for \( n = k \)). Since for circuit matrices \( a_{(k+1)(k+1)} > 0 \), it is valid that \( A_1^{(k+1)} = a_{(k+1)(k+1)} A_1^{(k)} + A''_1^{(k+1)} \) or \( A''_1^{(k+1)} = \frac{1}{a_{(k+1)(k+1)}} (A_1^{(k+1)} - A''_1^{(k+1)}) \).

Using similar transformations for minor element \( b_{1i} \) of matrix \( B^{(k+1)} \), we have \( B''_1^{(k)} = \frac{1}{b_{1i}} (B_1^{(k+1)} - B''_1^{(k+1)}) \). By the condition of the statement \( A_1^{(k+1)} = B_1^{(k+1)} \) and \( a_{(k+1)(k+1)} = b_{1i} \).

Minors \( A''_1^{(k+1)} \) and \( B''_1^{(k+1)} \) differ from \( A_1^{(k+1)} \) and \( B_1^{(k+1)} \) only by one element of the last \((k+1)\)nd column, therefore \( A''_1^{(k+1)} = B''_1^{(k+1)} \) and \( A_1^{(k)} = B_1^{(k)} \) follow from \( A''_1^{(k+1)} = B''_1^{(k+1)} \).

Similarly, all other minors can be decomposed from the condition of the statement, therefore if \( a_i = b_i \), \( A''_1^{(k+1)} = B''_1^{(k+1)} \), \( A''_1^{(k+1)} = B''_1^{(k+1)} \) for all \( i = 1, n \) at \( n = k \), then \( A^{(k)} = B^{(k)} \).

Matrices \( A^{(k+1)} \) and \( B^{(k+1)} \) differ from matrices \( A^{(k)} \) and \( B^{(k)} \) by the presence of the \((k+1)\) column and line. But for circuit matrices, the sum of all elements of any column (line) is equal to one, therefore, if \( A^{(k)} = B^{(k)} \), then \( A^{(k+1)} = B^{(k+1)} \). In other words, if the statement \( n = k \) is valid, \( n = k + 1 \) is valid as well.

5. Algorithm of graph isomorphism recognition

The validity of Statement 1 allows us to construct the following algorithm for determining function \( f : V'_1 \rightarrow V_2 \), as a substitution \( \pi \).

Let us denote by \( m_{(1)}^k \) and \( m_{(2)}^k \) - the \( k \)-th and \( l \)-th columns of matrices \( M_1 \) and \( M_2 \) respectively. Let \( k \) and \( l \) are such that \( m_{(1)}^k = m_{(2)}^l \). We construct matrix \( C^{kl} \) such that \( c_{ij}^{kl} = 1 \) if \( m_{ij}^k = m_{ij}^l \), \( m_{ij}^k = m_{ij}^l \) and \( a_i = b_i \), all the other elements of matrix \( C^{kl} \) are equal to zero. If \( m_{ij}^k \) and \( m_{ij}^l \) are equal, for \( C^{kl} \) there is at least one set \( n \) of independent units. If this set is unique, then it uniquely defines the renumbering order which allows to obtain matrix \( M_2 \) from matrix \( M_1 \). Matrix \( C^{kl} \) is the permissibility matrix of permutation \( m_{ij}^{(k+1)} \) and \( m_{ij}^{(l+1)} \) [16].

If this set is not the only one, then there are lines with more than one unit in matrix \( C^{kl} \), let be \( s \) is the number of one of these lines.

Then let us form corresponding permutation admissibility matrix \( C^{kl} \), and construct matrix \( C = C^{kl} + C^{l \pi} \), \( k \neq s \), \( l \neq t \) for any non-zero element \( c_{ji}^{kl} \) of matrix \( C^{kl} \). Any set of \( n \) independent twos for \( C \) does not mean that both for \( C^{kl} \) and \( C^{l \pi} \) there is the same \( n \) set of independent units. If this set \( n \) of independent twos is unique, then it uniquely defines the renumbering order which allows to obtain matrix \( M_1 \) from matrix \( M_2 \). If this set is not unique, then we continue similar constructions. Let \( C = \sum_{i=1}^{n} C^s_{pi}^{(i)} \). If for \( C \) there exists a set \( n \) of independent "\( s \)" units, then the indices of this set uniquely determine desired function \( f : V'_1 \rightarrow V_2 \) as substitution \( \pi \), and matrix \( P = \|p_{ij}\| \), \( p_{ij} = 1 \) with \( j = \pi(i) \) and \( p_{ij} = 0 \) with \( j \neq \pi(i) \) for all \( i, j = 1, n \), and \( M_1 = PM_2P^T \).
6. Computational complexity
Calculation of matrices $M_1$ and $M_2$ requires no more than $O(n^3)$ operations. Then to build each of $n$ matrices $C$ $O(n^3)$ operations are required. Consequently, computational complexity of the proposed method for recognizing isomorphic graphs not more than $O(n^3)$ is polynomial.

7. Conclusion
The issue of isomorphism recognition is one of the most important in the graph theory and belongs to the class of NP-complete problems which includes problems for which polynomial algorithms are unknown.

The issue of computational complexity of the problem remains open, although for some special cases efficient algorithms with polynomial computational complexity have been constructed.

The fastest known algorithm for determining graph isomorphism was developed by Babai and Lax [17, 18].

This paper proposes and justifies a method for recognizing isomorphic graphs, the computational complexity of which is polynomial.

The paper developed and justified a graph isomorphism recognition method whose computational complexity is polynomial.

References
[1] Melentiev V A 2004 Structural fault tolerance, survivability and reliability of systems Proc. of the III Int. Conf. "System Identification and Control Problems" SICPRO 4 January 28–30 Moscow: V.A. Trapeznikova RAS p 1545–62.
[2] Trofimov M I and Smolensky E A 2005 The use of electronegativity indices of organic molecules in chemical informatics problems News of the Academy of Sciences. Chemical series
[3] Nechepurenko M I et al. 1990 Algorithms and programs for solving problems using graphs and networks (Novosibirsk: Science)
[4] Karelin V P 1995 Theory and decision support tools in organizational and technological systems (Taganrog: TSURE)
[5] Melikhov A N and Karelin V P 1995 Methods for recognizing isomorphism and isomorphic embedding of clear and fuzzy graphs: study guide (Taganrog: Publishing house of TSURE)
[6] Bershtein L S, Karelin V P and Tselykh A N 1999 Models and methods of decision making in integrated intelligent systems (Rostov D: RTU Publishing House)
[7] Gehry M and Johnson D 1982 Computers and inaccessible tasks (Moscow: Mir)
[8] Pinchuk V P 2001 Table invariants on graphs and their application Cybernetics and Systems Analysis 4
[9] Karp R M 1975 Compatibility of combinatorial problems (Moscow: Mir)
[10] Corneil D G and Gotlieb C C 1970 An efficient algorithm for graph isomorphism ACM 17 51 – 64
[11] Weinberg L 1966 A simple and efficient algorithm for determining isomorphism of planar triply connected graphs IEEE Trans. Circuit Theory 13 142–148
[12] Hopcroft J E and Tarjan R E 1973 A VlogV algorithm for isomorphism of triconnected planar graphs J. Comput. Syst. 7 323–331.
[13] J E Hopcroft and J K Wong 1974 Linear time algorithm for isomorphism on planar graphs Proc. 6th Annual ACM Symp. On Theory of Computing p 172–184.
[14] Venikov V A, Zuev E N, Litkens I V, Markovich I M, Melnikov N A, Soldatkina L A and Stroyev V A 1981 Electrical systems. Mathematical problems of electric power (Moscow: Higher School)
[15] Swami M and Tkhalasiraman K 1984 Graphs, networks and algorithms (Moscow: Mir)
[16] Dudov M Kh 2013 Computational complexity of recognition of isomorphic graphs (Moscow: IP RadioSoft)
[17] Babai L 1983 Permutation Groups, Coherent Configurations and Graph Isomorphism, Doctor Dissertation, Hungarian Academy of Sciences.
[18] Babai L, Kantor W M, Eugene M Luks 1983 Computational complexity and the classification of finite simple groups In: Proc. 24th IEEE FOCS pp 162-171.