

Graph isomorphism problem

From Wikipedia, the free encyclopedia

The **graph isomorphism problem** is the computational problem of determining whether two finite graphs are isomorphic.

Besides its practical importance, the graph isomorphism problem is a curiosity in computational complexity theory as it is one of a very small number of problems belonging to NP neither known to be solvable in polynomial time nor NP-complete: it is one of only 12 such problems listed by Garey & Johnson (1979), and one of only two of that list whose complexity remains unresolved (the other being integer factorization).^[1] As of 2008 the best algorithm (Eugene Luks, 1983) has run time $2^{O(\sqrt{n \log n})}$ for graphs with n vertices.^{[2][3]}

List of unsolved problems in computer science

Is the graph isomorphism problem in P, or NP-complete, or neither?

It is known that the graph isomorphism problem is in the low hierarchy of class NP, which implies that it is not NP-complete unless the polynomial time hierarchy collapses to its second level.^[4]

At the same time, isomorphism for many special classes of graphs can be solved in polynomial time, and in practice graph isomorphism can often be solved efficiently.^[5]

This problem is a special case of the subgraph isomorphism problem,^[6] which is known to be NP-complete. It is also known to be a special case of the non-abelian hidden subgroup problem over the symmetric group.^[7]

Contents

- 1 State of the art
- 2 Solved special cases
- 3 Complexity class GI
 - 3.1 GI-complete and GI-hard problems
 - 3.1.1 Isomorphism of other objects
 - 3.1.2 GI-complete classes of graphs
 - 3.1.3 Other GI-complete problems
 - 3.1.4 GI-hard problems
- 4 Program checking
- 5 Applications
- 6 See also
- 7 Notes
- 8 References
 - 8.1 Surveys and monographs

State of the art

The best current theoretical algorithm is due to Eugene Luks (1983), and is based on the earlier work by Luks (1981), Babai & Luks (1982), combined with a *subfactorial* algorithm due to Zemlyachenko (1982). The algorithm relies on the classification of finite simple groups. Without CFSG, a slightly weaker bound $2^{O(\sqrt{n} \log^2 n)}$ was obtained first for strongly regular graphs by László Babai (1980), and then extended to general graphs by Babai & Luks (1982). Improvement of the exponent \sqrt{n} is a major open problem; for strongly regular graphs this was done by Spielman (1996). For hypergraphs of bounded rank, a subexponential upper bound matching the case of graphs, was recently obtained by Babai & Codenotti (2008).

On a side note, the graph isomorphism problem is computationally equivalent to the problem of computing the automorphism group of a graph, and is weaker than the permutation group isomorphism problem, and the permutation group intersection problem. For the latter two problems, Babai, Kantor and Luks (1983) obtained complexity bounds similar to that for the graph isomorphism.^[8]

There are several competing practical algorithms for graph isomorphism, due to McKay (1981), Schmidt & Druffel (1976), Ullman (1976), etc. While they seem to perform well on random graphs, a major drawback of these algorithms is their exponential time performance in the worst case.^[9]

Solved special cases

A number of important special cases of the graph isomorphism problem have efficient, polynomial-time solutions:

- Trees^[10]
- Planar graphs^[11] (In fact, planar graph isomorphism is in log space,^[12] a class contained in P.)
- Interval graphs^[13]
- Permutation graphs^[14]
- Partial k -trees^[15]
- Bounded-parameter graphs
 - Graphs of bounded genus^[16] (Note: planar graphs are graphs of genus 0)
 - Graphs of bounded degree^[17]
 - Graphs with bounded eigenvalue multiplicity^[18]
 - k -Contractible graphs (a generalization of bounded degree and bounded genus)^[19]
 - Color-preserving isomorphism of colored graphs with bounded color multiplicity (i.e., at most k vertices have the same color for a fixed k) is in class NC, which is a subclass of P.^[20]

Complexity class GI

Since the graph isomorphism problem is neither known to be NP-complete nor to be tractable, researchers have sought to gain insight into the problem by defining a new class **GI**, the set of problems with a polynomial-time Turing reduction to the graph isomorphism problem.^[21] If in fact the graph isomorphism problem is solvable in polynomial time, **GI** would equal **P**.

As is common for complexity classes within the polynomial time hierarchy, a problem is called **GI-hard** if there is a polynomial-time Turing reduction from any problem in **GI** to that problem, i.e., a polynomial-time solution to a GI-hard problem would yield a polynomial-time solution to the graph isomorphism problem (and so all problems in **GI**). A problem **X** is called complete for **GI**, or **GI-complete**, if it is both GI-hard and a polynomial-time solution to the GI problem would yield a polynomial-time solution to **X**.

The graph isomorphism problem is contained in both **NP** and co-**AM**. GI is contained in and low for Parity P, as well as contained in the potentially much smaller class SPP

(http://qwiki.stanford.edu/wiki/Complexity_Zoo:S#spp).^[22] That it lies in Parity P means that the graph isomorphism problem is no harder than determining whether a polynomial-time nondeterministic Turing machine has an even or odd number of accepting paths. GI is also contained in and low for ZPP^{NP} .^[23] This essentially means that an efficient Las Vegas algorithm with access to an NP oracle can solve graph isomorphism so easily that it gains no power from being given the ability to do so in constant time.

GI-complete and GI-hard problems

Isomorphism of other objects

There are a number of classes of mathematical objects for which the problem of isomorphism is a GI-complete problem. A number of them are graphs endowed with additional properties or restrictions:^[24]

- digraphs^[24]
- labelled graphs, with the proviso that an isomorphism is not required to preserve the labels,^[24] but only the equivalence relation consisting of pairs of vertices with the same label
- "polarized graphs" (made of a complete graph K_m and an empty graph K_n plus some edges connecting the two; their isomorphism must preserve the partition)^[24]
- 2-colored graphs^[24]
- explicitly given finite structures^[24]
- multigraphs^[24]
- hypergraphs^[24]
- finite automata^[24]
- Markov Decision Processes^[25]
- commutative class 3 nilpotent (i.e., $xyz = 0$ for every elements x, y, z) semigroups^[24]
- finite rank associative algebras over a fixed algebraically closed field with zero squared radical and

commutative factor over the radical^{[24][26]}

- context-free grammars^[24]
- balanced incomplete block designs^[24]
- Recognizing combinatorial isomorphism of convex polytopes represented by vertex-facet incidences.^{[2][27]}

This list is incomplete; you can help by expanding it (https://en.wikipedia.org/w/index.php?title=Graph_isomorphism_problem&action=edit).

GI-complete classes of graphs

A class of graphs is called GI-complete if recognition of isomorphism for graphs from this subclass is a GI-complete problem. The following classes are GI-complete:^[24]

- connected graphs^[24]
- graphs of diameter 2 and radius 1^[24]
- directed acyclic graphs^[24]
- regular graphs.^[24]
- bipartite graphs without non-trivial strongly regular subgraphs^[24]
- bipartite Eulerian graphs^[24]
- bipartite regular graphs^[24]
- line graphs^[24]
- chordal graphs^[24]
- regular self-complementary graphs^[24]
- polytopal graphs of general, simple, and simplicial convex polytopes in arbitrary dimensions^[27]

This list is incomplete; you can help by expanding it (https://en.wikipedia.org/w/index.php?title=Graph_isomorphism_problem&action=edit).

Many classes of digraphs are also GI-complete.

Other GI-complete problems

There are other nontrivial GI-complete problems in addition to isomorphism problems.

- The recognition of self-complementarity of a graph or digraph.^[28]
- A clique problem for a class of so-called M -graphs. It is shown that finding an isomorphism for n -vertex graphs is equivalent to finding an n -clique in an M -graph of size n^2 . This fact is interesting

because the problem of finding an $(n - \epsilon)$ -clique in a M -graph of size n^2 is NP-complete for arbitrarily small positive ϵ .^[29]

- The problem of homeomorphism of 2-complexes.^[30]

GI-hard problems

- The problem of counting the number of isomorphisms between two graphs is polynomial-time equivalent to the problem of telling whether even one exists.^[31]
- The problem of deciding whether two convex polytopes given by either the V-description or H-description are projectively or affinely isomorphic. The latter means existence of a projective or affine map between the spaces that contain the two polytopes (not necessarily of the same dimension) which induces a bijection between the polytopes.^[27]

Program checking

Blum and Kannan^[32] have shown a program checker for graph isomorphism. Suppose P is a claimed polynomial-time procedure that checks if two graphs are isomorphic, but it is not trusted. To check if G and H are isomorphic:

- Ask P whether G and H are isomorphic.
 - If the answer is "yes":
 - Attempt to construct an isomorphism using P as subroutine. Mark a vertex u in G and v in H , and modify the graphs to make them distinctive (with a small local change). Ask P if the modified graphs are isomorphic. If no, change v to a different vertex. Continue searching.
 - Either the isomorphism will be found (and can be verified), or P will contradict itself.
 - If the answer is "no":
 - Perform the following 100 times. Choose randomly G or H , and randomly permute its vertices. Ask P if the graph is isomorphic to G and H . (As in AM protocol for graph nonisomorphism).
 - If any of the tests are failed, judge P as invalid program. Otherwise, answer "no".

This procedure is polynomial-time and gives the correct answer if P is a correct program for graph isomorphism. If P is not a correct program, but answers correctly on G and H , the checker will either give the correct answer, or detect invalid behaviour of P . If P is not a correct program, and answers incorrectly on G and H , the checker will detect invalid behaviour of P with high probability, or answer wrong with probability 2^{-100} .

Notably, P is used only as a blackbox.

Applications

In cheminformatics and in mathematical chemistry, graph isomorphism testing is used to identify a chemical compound within a chemical database.^[33] Also, in organic mathematical chemistry graph isomorphism testing is useful for generation of molecular graphs and for computer synthesis.

Chemical database search is an example of graphical data mining, where the graph canonization approach is often used.^[34] In particular, a number of identifiers for chemical substances, such as SMILES and InChI, designed to provide a standard and human-readable way to encode molecular information and to facilitate the search for such information in databases and on the web, use canonization step in their computation, which is essentially the canonization of the graph which represents the molecule.

In electronic design automation graph isomorphism is the basis of the Layout Versus Schematic (LVS) circuit design step, which is a verification whether the electric circuits represented by a circuit schematic and an integrated circuit layout are the same.^[35]

See also

- Graph automorphism problem
- Graph canonization

Notes

1. ^ The latest one resolved was minimum-weight triangulation, proved to be NP-complete in 2006. Mulzer, Wolfgang; Rote, Günter (2008), "Minimum-weight triangulation is NP-hard", *Journal of the ACM* 55 (2): 1, arXiv:cs.CG/0601002 (<https://arxiv.org/abs/cs.CG/0601002>), doi:10.1145/1346330.1346336 (<http://dx.doi.org/10.1145%2F1346330.1346336>).
2. ^ ^a ^b Johnson 2005
3. ^ Babai & Codenotti (2008).
4. ^ Uwe Schöning, "Graph isomorphism is in the low hierarchy", Proceedings of the 4th Annual Symposium on Theoretical Aspects of Computer Science, 1987, 114–124; also: *Journal of Computer and System Sciences*, vol. 37 (1988), 312–323
5. ^ McKay 1981
6. ^ Ullman (1976).
7. ^ Cristopher Moore; Alexander Russell; Schulman, Leonard J. (2005). "The Symmetric Group Defies Strong Fourier Sampling: Part I". arXiv:quant-ph/0501056v3 (<http://arxiv.org/abs/quant-ph/0501056v3>) [quant-ph (<http://arxiv.org/archive/quant-ph>)].
8. ^ László Babai, William Kantor, Eugene Luks, Computational complexity and the classification of finite simple groups (<http://portal.acm.org/citation.cfm?id=1382437.1382817>), Proc. 24th FOCS (1983), pp. 162-171.
9. ^ P. Foggia, C.Sansone, M. Vento, A Performance Comparison of Five Algorithms for Graph Isomorphism (http://www.engr.uconn.edu/~vkk06001/GraphIsomorphism/Papers/VF_SD_NAUTY_Ullman_Experiments.pdf), Proc. 3rd IAPR-TC15 Workshop Graph-Based Representations in Pattern Recognition, 2001, pp. 188-199.

10. ^ P.J. Kelly, "A congruence theorem for trees" *Pacific J. Math.*, 7 (1957) pp. 961–968; Aho, Hopcroft & Ullman 1974.
11. ^ Hopcroft & Wong 1974
12. ^ Datta, S.; Limaye, N.; Nimbhorkar, P.; Thierauf, T.; Wagner, F. (2009). "2009 24th Annual IEEE Conference on Computational Complexity". p. 203. doi:10.1109/CCC.2009.16 (<http://dx.doi.org/10.1109%2FCCC.2009.16>). ISBN 978-0-7695-3717-7. |chapter= ignored (help)
13. ^ Booth & Lueker 1979
14. ^ Colbourn 1981
15. ^ Bodlaender 1990
16. ^ Miller 1980; Filotti & Mayer 1980.
17. ^ Luks 1982
18. ^ Babai, Grigoryev & Mount 1982
19. ^ Gary L. Miller: Isomorphism Testing and Canonical Forms for k -Contractable Graphs (A Generalization of Bounded Valence and Bounded Genus). *Proc. Int. Conf. on Foundations of Computer Theory*, 1983, pp. 310–327 (*Lecture Notes in Computer Science*, vol. 158, full paper in: *Information and Control*, 56(1–2):1–20, 1983.)
20. ^ Eugene Luks, "Parallel algorithms for permutation groups and graph isomorphism", *Proc. IEEE Symp. Foundations of Computer Science*, 1986, 292-302
21. ^ Booth & Colbourn 1977; Köbler, Schöning & Torán 1993
22. ^ Köbler, Schöning & Torán 1992; Arvind & Kurur 2006
23. ^ Arvind & Köbler 2000
24. ^ *a b c d e f g h i j k l m n o p q r s t u v w x* Zemlyachenko, Korneenko & Tyshkevich 1985
25. ^ "On the hardness of finding symmetries in Markov decision processes", by SM Narayanamurthy, B Ravindran (<http://www.cse.iitm.ac.in/~ravi/papers/Shravan-ICML08.pdf>), *Proceedings of the Twenty Fifth International Conference on Machine Learning (ICML 2008)*, pp. 688–696.
26. ^ D.Yu.Grigor'ev, "Complexity of “wild” matrix problems and of isomorphism of algebras and graphs", *Journal of Mathematical Sciences*, Volume 22, Number 3, 1983, pp. 1285–1289, doi:10.1007/BF01084390 (<http://dx.doi.org/10.1007%2FBF01084390>) (translation of a 1981 Russian language article)
27. ^ *a b c* Volker Kaibel, Alexander Schwartz, "On the Complexity of Polytope Isomorphism Problems" (<http://eprintweb.org/S/authors/All/ka/Kaibel/16>), *Graphs and Combinatorics*, 19 (2):215 —230, 2003.
28. ^ Colbourn M.J., Colbourn Ch.J. "Graph isomorphism and self-complementary graphs", *SIGACT News*, 1978, vol. 10, no. 1, 25–29.
29. ^ Kozen 1978.
30. ^ J. Shawe-Taylor, T.Pisanski, "Homeomorphism of 2-Complexes is Graph Isomorphism Complete", *SIAM Journal on Computing*, 23 (1994) 120 – 132 .
31. ^ R. Mathon, "A note on the graph isomorphism counting problem", *Information Processing Letters*, 8 (1979) pp. 131–132; Johnson 2005.
32. ^ Designing Programs that Check their Work (<ftp://ftp.cis.upenn.edu/pub/kannan/jacm.ps.gz>)
33. ^ Christophe-André Mario Imniger (2005) "Graph Matching: Filtering Databases of Graphs Using Machine Learning", ISBN 1-58603-557-6

34. ^ "Mining Graph Data", by Diane J. Cook, Lawrence B. Holder (2007) ISBN 0-470-07303-9, pp. 120–122, section 6.2.1. "Canonical Labeling" (http://books.google.com/books?id=bHGY0_H0g8QC&pg=PA119&dq=%22canonical+labeling%22+graphs#PPA120,M1)
35. ^ Baird, HS and Cho, YE (1975). "An artwork design verification system" (<http://dl.acm.org/citation.cfm?id=809095>). Proceedings of the 12th Design Automation Conference. IEEE Press. pp. 414–420.

References

- Aho, Alfred V.; Hopcroft, John; Ullman, Jeffrey D. (1974), *The Design and Analysis of Computer Algorithms*, Reading, MA: Addison–Wesley.
- Arvind, Vikraman; Köbler, Johannes (2000), "Graph isomorphism is low for ZPP(NP) and other lowness results.", *Proceedings of the 17th Annual Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science **1770**, Springer-Verlag, pp. 431–442, ISBN 3-540-67141-2, OCLC 43526888 (<https://www.worldcat.org/oclc/43526888>).
- Arvind, Vikraman; Kurur, Piyush P. (2006), "Graph isomorphism is in SPP", *Information and Computation* **204** (5): 835–852, doi:10.1016/j.ic.2006.02.002 (<http://dx.doi.org/10.1016%2Fj.ic.2006.02.002>).
- Babai, László; Codenotti, Paolo (2008), "Isomorphism of Hypergraphs of Low Rank in Moderately Exponential Time", *FOCS '08: Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society, pp. 667–676, ISBN 978-0-7695-3436-7.
- Babai, László; Grigoryev, D. Yu.; Mount, David M. (1982), "Isomorphism of graphs with bounded eigenvalue multiplicity", *Proceedings of the 14th Annual ACM Symposium on Theory of Computing*, pp. 310–324, doi:10.1145/800070.802206 (<http://dx.doi.org/10.1145%2F800070.802206>), ISBN 0-89791-070-2.
- Bodlaender, Hans (1990), "Polynomial algorithms for graph isomorphism and chromatic index on partial k -trees", *Journal of Algorithms* **11** (4): 631–643, doi:10.1016/0196-6774(90)90013-5 (<http://dx.doi.org/10.1016%2F0196-6774%2890%2990013-5>).
- Booth, Kellogg S.; Colbourn, C. J. (1977), *Problems polynomially equivalent to graph isomorphism*, Technical Report CS-77-04, Computer Science Department, University of Waterloo.
- Booth, Kellogg S.; Lueker, George S. (1979), "A linear time algorithm for deciding interval graph isomorphism", *Journal of the ACM* **26** (2): 183–195, doi:10.1145/322123.322125 (<http://dx.doi.org/10.1145%2F322123.322125>).
- Boucher, C.; Loker, D. (2006), *Graph isomorphism completeness for perfect graphs and subclasses of perfect graphs* (<http://www.cs.uwaterloo.ca/research/tr/2006/CS-2006-32.pdf>), University of Waterloo, Technical Report CS-2006-32.
- Colbourn, C. J. (1981), "On testing isomorphism of permutation graphs", *Networks* **11**: 13–21, doi:10.1002/net.3230110103 (<http://dx.doi.org/10.1002%2Fnet.3230110103>).
- Filotti, I. S.; Mayer, Jack N. (1980), "A polynomial-time algorithm for determining the isomorphism of graphs of fixed genus", *Proceedings of the 12th Annual ACM Symposium on Theory of Computing*, pp. 236–243, doi:10.1145/800141.804671 (<http://dx.doi.org/10.1145%2F800141.804671>), ISBN 0-89791-017-6.
- Garey, Michael R.; Johnson, David S. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, ISBN 978-0-7167-1045-5, OCLC 11745039 (<https://www.worldcat.org/oclc/11745039>).

- Hopcroft, John; Wong, J. (1974), "Linear time algorithm for isomorphism of planar graphs", *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing*, pp. 172–184, doi:10.1145/800119.803896 (<http://dx.doi.org/10.1145%2F800119.803896>).
- Köbler, Johannes; Schöning, Uwe; Torán, Jacobo (1992), "Graph isomorphism is low for PP", *Computational Complexity* 2 (4): 301–330, doi:10.1007/BF01200427 (<http://dx.doi.org/10.1007%2FBF01200427>).
- Köbler, Johannes; Schöning, Uwe; Torán, Jacobo (1993), *The Graph Isomorphism Problem: Its Structural Complexity*, Birkhäuser, ISBN 978-0-8176-3680-7, OCLC 246882287 (<https://www.worldcat.org/oclc/246882287>).
- Kozen, Dexter (1978), "A clique problem equivalent to graph isomorphism", *ACM SIGACT News* 10 (2): 50–52, doi:10.1145/990524.990529 (<http://dx.doi.org/10.1145%2F990524.990529>).
- Luks, Eugene M. (1982), "Isomorphism of graphs of bounded valence can be tested in polynomial time", *Journal of Computer and System Sciences* 25: 42–65, doi:10.1016/0022-0000(82)90009-5 (<http://dx.doi.org/10.1016%2F0022-0000%2882%2990009-5>).
- McKay, Brendan D. (1981), "Practical graph isomorphism" (<http://cs.anu.edu.au/~bdm/nauty/PGI/>), *Congressus Numerantium* 30: 45–87, 10th. Manitoba Conference on Numerical Mathematics and Computing (Winnipeg, 1980).
- Miller, Gary (1980), "Isomorphism testing for graphs of bounded genus", *Proceedings of the 12th Annual ACM Symposium on Theory of Computing*, pp. 225–235, doi:10.1145/800141.804670 (<http://dx.doi.org/10.1145%2F800141.804670>), ISBN 0-89791-017-6.
- Schmidt, Douglas C.; Druffel, Larry E. (1976), "A Fast Backtracking Algorithm to Test Directed Graphs for Isomorphism Using Distance Matrices", *J. ACM (ACM)* 23 (3): 433–445, doi:10.1145/321958.321963 (<http://dx.doi.org/10.1145%2F321958.321963>), ISSN 0004-5411 (<https://www.worldcat.org/issn/0004-5411>).
- Spielman, Daniel A. (1996), "Faster isomorphism testing of strongly regular graphs", *STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, ACM, pp. 576–584, ISBN 978-0-89791-785-8.
- Ullman, Julian R. (1976), "An algorithm for subgraph isomorphism", *Journal of the ACM* 23: 31–42, doi:10.1145/321921.321925 (<http://dx.doi.org/10.1145%2F321921.321925>).

Surveys and monographs

- Read, Ronald C.; Corneil, Derek G. (1977), "The graph isomorphism disease", *Journal of Graph Theory* 1 (4): 339–363, doi:10.1002/jgt.3190010410 (<http://dx.doi.org/10.1002%2Fjgt.3190010410>), MR 0485586 (<https://www.ams.org/mathscinet-getitem?mr=0485586>).
- Gati, G. "Further annotated bibliography on the isomorphism disease." – *Journal of Graph Theory* 1979, 3, 95–109.
- Zemlyachenko, V. N.; Korneenko, N. M.; Tyshkevich, R. I. (1985), "Graph isomorphism problem", *Journal of Mathematical Sciences* 29 (4): 1426–1481, doi:10.1007/BF02104746 (<http://dx.doi.org/10.1007%2FBF02104746>). (Translated from *Zapiski Nauchnykh Seminarov Leningradskogo Otdeleniya Matematicheskogo Instituta im. V. A. Steklova AN SSSR* (Records of Seminars of the Leningrad Department of Steklov Institute of Mathematics of the USSR Academy of Sciences), Vol. 118, pp. 83–158, 1982.)

- Arvind, V.; Jacobo Torán (2005), "Isomorphism Testing: Perspectives and Open Problems" (<http://theorie.informatik.uni-ulm.de/Personen/toran/beatcs/column86.pdf>), *Bulletin of the European Association for Theoretical Computer Science* (no. 86): 66–84 . (A brief survey of open questions related to the isomorphism problem for graphs, rings and groups.)
- Köbler, Johannes; Uwe Schöning, Jacobo Torán (1993), *Graph Isomorphism Problem: The Structural Complexity*, Birkhäuser Verlag, ISBN 0-8176-3680-3, OCLC 246882287 (<https://www.worldcat.org/oclc/246882287>) . (*From the book cover*: The books focuses on the issue of the computational complexity of the problem and presents several recent results that provide a better understanding of the relative position of the problem in the class NP as well as in other complexity classes.)
- Johnson, David S. (2005), "The NP-Completeness Column", *ACM Transactions on Algorithms* **1** (no. 1): 160–176, doi:10.1145/1077464.1077476 (<http://dx.doi.org/10.1145%2F1077464.1077476>). (This 24th edition of the Column discusses the state of the art for the open problems from the book *Computers and Intractability* and previous columns, in particular, for Graph Isomorphism.)
- Torán, Jacobo; Fabian Wagner (2009), "The Complexity of Planar Graph Isomorphism" (<http://theorie.informatik.uni-ulm.de/Personen/toran/beatcs/column97.pdf>), *Bulletin of the European Association for Theoretical Computer Science* (no. 97) .

Software

- Graph Isomorphism (<http://www.cs.sunysb.edu/~algorithm/files/graph-isomorphism.shtml>), review of implementations, The Stony Brook Algorithm Repository (<http://www.cs.sunysb.edu/~algorithm>).

Retrieved from "http://en.wikipedia.org/w/index.php?title=Graph_isomorphism_problem&oldid=637746824"

Categories: Graph algorithms | Morphisms | Computational problems in graph theory
| Unsolved problems in computer science

-
- This page was last modified on 12 December 2014 at 08:48.
 - Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.