Article

Not peer-reviewed version

Three Problems in Graph Coteries to Show P Does not Equal NP

Dharmarajan R * and Ramachandran D

Posted Date: 24 July 2025

doi: 10.20944/preprints202507.2089.v1

Keywords: Algorithm; polynomial time; connectedness; Eulerian graphs; cliques; class P; class NP

Article

Three Problems in Graph Coteries to Show *P* Does not Equal *NP*

R. Dharmarajan [1,*] and D. Ramachandran [2]

1 Niels Abel Foundation, Palakkad 678011, Kerala, India
2 Sathyabama University
* Correspondence: mathnafrd@gmail.com

Abstract

The ***P*** versus ***NP*** problem, a conjecture formulated by Stephen Cook in 1971, is one of the most challenging problems in contemporary mathematics and theoretical computer science. A concise mathematical formulation of the problem reads: ***is*** $\boldsymbol{P = NP}$? In longer phrasing, this asks: given a problem instance, if some additional data can be recognized fast enough as logically implying the existence of a solution (to the instance), then can a solution be computed fast enough without the aid of any such additional data? In this article we present the idea of graph coteries that are sets whose members are sets with specified graph-theoretic properties. Then we formulate three problems in graph coteries to show $\boldsymbol{P \neq NP}$.

Keywords: Algorithm; polynomial time; connectedness; Eulerian graphs; cliques; class ***P***; class ***NP***

MSC: 11Y16; 05C38; 05C45; 05C69

---

1. Introduction

More on terminologies, symbols and notations used in this article can be found in [3] or [6] or [8]. Throughout this article, $\mathbb{N}$ will denote the set of positive integers, $\mathbb{W}$ the set of non-negative integers (i.e., $\mathbb{W} = \mathbb{N} \cup \{0\}$), $\mathbb{Z}$ the set of integers and $\mathbb{R}$ the set of real numbers.

This article has eight sections. Section 1 directs the reader to some references for relevant details on algorithms, the problem classes ***P*** and ***NP***, attempted solutions and feasible solutions. Section 2 contains graph-theoretical concepts essential for the three problems hinted at in the abstract. Section 3 hints at the types of problem variants that are widely used.

The three problems are presented in Section 4. The first problem is centered on connectedness of graphs; the second on Eulerian graphs; and the third on cliques. Each of these three problems is shown to be in ***NP*** but not in ***P*** (Section 5 and Section 6). Section 7 mentions a few other possible graph coteries. Section 8 presents the conclusion of our discussions.

Our discussions also draw on the following topics that can be found in some detail in the suggested references:

1. Algorithms [3–5,10].
2. Problem instance and size of an instance [3–5].
3. A step in an algorithm [3–5].
4. Polynomial-time algorithm [2–5,10].
5. Class ***P*** and class ***NP*** of problems [2–5,9,11].
6. Feasible solutions and proposed solutions [3–5,7].
7. Certificates and certificate candidates; check and verification [4,5].
8. Atomic sub-outputs of solutions [4,5].
9. Bijective maps (or, bijections) [6,8].

 doi:10.20944/preprints202507.2089.v1

2. Essential Graph Theory

A *set* is a collection of definite and distinguishable objects [3,8]. Each object in a set is an *element* or a *member* of the set. It is taken for granted that if X is a given set then there is a well-formed definition that decides conclusively whether or not two given elements of X are distinct. Wherever necessary, such a definition is made explicit. There is a unique set that contains no members, and this is the *empty set*, denoted by ϕ.

The *cardinality* (or, *size*) of a set X is the number of elements in X, and is denoted by $|X|$. Obviously $|X| \geq 0$. If $|X| \in \mathbb{W}$ then X is a *finite set*; else X is an *infinite set*.

The set of all the subsets of a set X, including the empty set ϕ and the set X itself, is denoted by $\mathcal{P}(X)$ and is the *power set* of X. If X is finite of cardinality n then $\mathcal{P}(X)$ is finite [3,8] of cardinality 2^n. The set $\mathcal{P}(X) - \{\phi\}$ denotes the set of all nonempty subsets of X.

A *simple graph* G is a pair $G = (V, E)$ where:
(i) V is a finite set, each element of which is called a *vertex* of G and
(ii) E is a finite set, each element of which is a nonempty subset X of V such that $|X| \leq 2$ and is called an *edge* of G.

If G is a simple graph then neither V nor E can have any repeated elements. V is the *vertex set* of G and E is the *edge set* of G. $V = \phi$ is possible, which necessiates $E = \phi$, but the converse is not true.

The expressions $x \in V$ and $x \in G$ will both mean x is a vertex of G. The expression $\{x, y\} \in E$ will mean $\{x, y\}$ is an edge of G where $x \in G$ and $y \in G$.

The *order* of G is denoted by $|G|$ and is defined to be the number of vertices in G. Obviously, $|G| = |V|$. The *girth* of G is the number of edges in G and is denoted by $e(G)$ or by e as convenient, providing no ambiguity arises.

Two distinct vertices x and y of G are *adjacent* in G if $\{x, y\} \in E$. If $\{x, y\} \notin E$ then x and y are *nonadjacent*. The vertex x is *self-adjacent* if $\{x\} \in E$. G is *loop-free* if it has no self-adjacent vertex (meaning, $|X| = 2$ whenever $X \in E$).

By the above definitions, if $G = (V, E)$ is a simple loop-free graph of order n, it is immediate that the girth of G cannot exceed $\dfrac{n(n-1)}{2}$ [6].

Let $x \in G$. The *neighbourhood* of x in G is denoted by $N_G(x)$ or $N(x)$, and is defined to be the set of all $y \in V$ such that y is adjacent to x in G. The non-negative integer $|N_G(x)|$ is the *degree* of x in G, and is denoted by $dx(G)$ or dx.

All graphs considered in this article are assumed simple and loop-free with positive orders.

A graph $J = (W, F)$ is a *subgraph* of a graph $G = (V, E)$ if (i) $W \subseteq V$ and (ii) $F \subseteq E$. In this case J is said to be contained in G or G is said to contain J. J is a *proper subgraph* of G if J is a subgraph of G with either $W \subsetneq V$ or $F \subsetneq E$. If J is a subgraph of G such that $|J| = |G|$ then J is a *spanning subgraph* of G.

Let S be a nonempty subset of V - i.e., $S \in \mathcal{P}(V) - \{\phi\}$. The *subgraph of G induced by S* is denoted by $G[S]$ and is defined to be the graph $G[S] = (S, E[S])$ where $E[S]$ is the set of all the edges $\{x, y\} \in E$ such that $x \in S$ and $y \in S$. The order of $G[S] = (S, E[S])$ is (obviously) $|S|$. For convenience, we will denote the girth of $G[S]$ by $e(S)$ (rather than $e(G[S])$) in the coming discussions. In particular, if $S = V - \{x\}$ then $G[S]$ will be denoted by $G - x$. In this case $e(S) = e - dx(G)$ where e is the girth of G.

Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are *isomorphic* if there exists a bijection [6] $f : V_1 \to V_2$ with the property that x and y are adjacent vertices in G_1 if and only if $f(x)$ and $f(y)$ are adjacent vertices in G_2. In this case the graph-theoretic properties of G_1 are exactly those of G_2, and vice-versa.

Let $G = (V, E)$ be a graph. In Subsection 2.1 through Subsecion 2.3, we define coteries of G based on specific graph theoretic properties possessed (or not) by subsets of V. For the remainder of this section, assume $G = (V, E)$ and $|G| \geq 1$.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 July 2025 doi:10.20944/preprints202507.2089.v1

2.1. Connect Coteries

A *path* in G is a sequence $x_1, \ldots, x_k$ of k distinct vertices of G such that $k \geq 2$ and $\{x_j, x_{j+1}\} \in E$ for each $j = 1, \ldots, k-1$. Such a sequence is also called a path between x_1 and x_k. G is a *connected graph* if there is a path between x and y whenever x and y are distinct vertices of G. A graph that is not connected is *disconnected*. A graph of order 1 is assumed connected.

The set of all $S \in \mathcal{P}(V) - \{\phi\}$ such that the induced subgraph $G[S]$ is connected will be called the *connect coterie* of G, and will be denoted by $\Lambda^+(G)$.

The set of all $S \in \mathcal{P}(V) - \{\phi\}$ such that $G[S]$ is disconnected will be called the *disconnect coterie* of G, and will be denoted by $\Lambda^-(G)$.

Clearly $\Lambda^+(G) \cup \Lambda^-(G) = \mathcal{P}(V) - \{\phi\}$ and $\Lambda^+(G) \cap \Lambda^-(G) = \phi$.

Either of $\Lambda^+(G)$ and $\Lambda^-(G)$ will be called a Λ *coterie* of G. These two coteries together will be referred to as *the pair of antithetic* Λ *coteries* (of G).

2.2. Euler Coteries

G is an *Eulerian graph* [6] if $dx(G)$ is even for each vertex x of G.

The set of all $S \in \mathcal{P}(V) - \{\phi\}$ such that $G[S]$ is an Eulerian graph will be called the *Euler coterie* of G, and will be denoted by $\mathcal{E}^+(G)$.

The set of all $S \in \mathcal{P}(V) - \{\phi\}$ such that $G[S]$ is not an Eulerian graph will be called the *non-Euler coterie* of G, and will be denoted by $\mathcal{E}^-(G)$.

Obviously $\mathcal{E}^+(G) \cup \mathcal{E}^-(G) = \mathcal{P}(V) - \{\phi\}$ and $\mathcal{E}^+(G) \cap \mathcal{E}^-(G) = \phi$.

Either of $\mathcal{E}^+(G)$ and $\mathcal{E}^-(G)$ will be called an $\mathcal{E}$ *coterie* of G. These two coteries together will be referred to as *the pair of antithetic* $\mathcal{E}$ *coteries* (of G).

2.3. Clique Coteries

A nonempty subset M of V is a *clique* of G if either (i) $|M| > 1$ and the vertices of M are pairwise adjacent in G or (ii) $|M| = 1$. G is a *complete graph* if V is a clique of G.

The set of all $S \in \mathcal{P}(V) - \{\phi\}$ such that S is a clique of G (i.e., the $G[S]$ is a complete graph) will be called the *clique coterie of* G, and will be denoted by $\Omega^+(G)$.

The set of all $S \in \mathcal{P}(V) - \{\phi\}$ such that S is not a clique of G will be called the *non-clique coterie of* G, and will be denoted by $\Omega^-(G)$.

It is immediate that $\Omega^+(G) \cup \Omega^-(G) = \mathcal{P}(V) - \{\phi\}$ and $\Omega^+(G) \cap \Omega^-(G) = \phi$.

Either of $\Omega^+(G)$ and $\Omega^-(G)$ will be called a Ω *coterie* of G. These two coteries together will be referred to as *the pair of antithetic* Ω *coteries* (of G).

2.4. On Cardinalities Of Graph Coteries

Proposition 2.1. Given a graph G of positive order, at least one coterie in each of the pairs of antithetic coteries (of G) seen in Sections 2.1 through 2.3 exists.

Proof. It suffices to prove the statement for the pair of antithetic Λ coteries of G; a similar reasoning applies to each of the other pairs of antithetic coteries.

The conclusion holds since $|\Lambda^+(G)| + |\Lambda^-(G)| = |\mathcal{P}(V) - \{\phi\}| = 2^n - 1$ from the definitions of the Λ coteries (Subsection 2.1.). •

Proposition 2.2. Let $G = (V, E)$ have order n. Then:
(i) Either $|\Lambda^+(G)| \geq 2^{n-1}$ or $|\Lambda^-(G)| \geq 2^{n-1}$.
(ii) Either $|\mathcal{E}^+(G)| \geq 2^{n-1}$ or $|\mathcal{E}^+(G)| \geq 2^{n-1}$.
(iii) Either $|\Omega^+(G)| \geq 2^{n-1}$ or $|\Omega^-(G)| \geq 2^{n-1}$.

 doi:10.20944/preprints202507.2089.v1

Proof. It suffices to prove (i); a similar reasoning applies to each of (ii) and (iii). By the defintions in Subsection 2.1, we have $|\Lambda^+(G)| + |\Lambda^-(G)| = 2^n - 1$. Then either $|\Lambda^+(G)| \geq \frac{2^n-1}{2}$ or $|\Lambda^-(G)| \geq \frac{2^n-1}{2}$.

Next, $\frac{2^n-1}{2} = 2^{n-1} - \frac{1}{2} \notin \mathbb{N}$, whereas $|\Lambda^+(G)| \in \mathbb{N}$ and $|\Lambda^-(G)| \in \mathbb{N}$. Then (i) follows immediately. •

Proposition 2.3. The cardinality of one Λ coterie exceeds that of the other. A similar conclusion holds for each of the other two pairs of antithetic coteries

Proof. $|\Lambda^+(G)| + |\Lambda^-(G)| = 2^n - 1$, an odd positive integer. Then $|\Lambda^+(G)| \neq |\Lambda^-(G)|$. •

2.5. Superior Coteries

Of the two Λ coteries of G, the one with the larger cardinality (see Proposition 2.2 and Proposition 2.3) will be called *the superior Λ coterie*. A similar definition applies to each of the other types of coteries.

3. Problem Variants

A *variant* of a problem $\mathcal{Q}$ is a formulation of $\mathcal{Q}$ that seeks a desired type of solution to a given instance of $\mathcal{Q}$. Types of variants that are widely studied and used are: optimization, computation and decision.

An *optimization variant* of $\mathcal{Q}$ is a formulation of Q that asks for a solution of an optimum measure (which is either the maximum or the minimum of the concerned measure) to each instance of $\mathcal{Q}$ [1].

A *computation variant* of $\mathcal{Q}$ is a formulation that asks for a solution (to each instance of $\mathcal{Q}$) subject to finitely many conditions.

A *decision variant* of $\mathcal{Q}$ is a formulation of $\mathcal{Q}$ using a decision question that asks for either a "yes" or a "no" answer to each instance. The basic ingredients [1] of a decision variant are: the set of instances, the set of attempted solutions (i.e., certificate candidates [4,5]) and the predicate that decides whether an input certificate candidate yields a solution.

In the context of the ***P*** versus ***NP*** problem, we shall be concerned with optimization and decision variants only. It is common to formulate an optimization problem $\mathcal{Q}$ as a decision problem to find out if $\mathcal{Q}$ is in ***NP***.

4. Variants Of Problems In Graph Coteries

We formulate three problems that we name Problem Λ, Problem $\mathcal{E}$ and Problem Ω. We give their optimization and decision variants.

In each problem the input graph is $G = (V, E)$ of order $n \in \mathbb{N}$.

4.1. Optimization Variants

1. **Problem Λ.** Find the superior Λ coterie of G.
2. **Problem $\mathcal{E}$.** Find the superior $\mathcal{E}$ coterie of G.
3. **Problem Ω.** Find the superior Ω coterie of G.

4.2. Decision Variants of Problems Λ, $\mathcal{E}$ and Ω

In each of the following decision versions:
(i) the inputs are (i-a) graph G and (i-b) $n = |G|$,
(ii) the certificate candidate is $C(G) = n$ and
(iii) the output is either **YES** (i.e., the desired coterie exists) or **NO** (i.e., no such coterie exists).

1. **A decision variant of Problem Λ.**
   Decision question: Does there exist a Λ coterie of G of cardinality $\geq 2^{n-1}$?
2. **A decision variant of Problem $\mathcal{E}$.**

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 July 2025 doi:10.20944/preprints202507.2089.v1

Decision question: Does there exist an $\mathcal{E}$ coterie of G of cardinality $\geq 2^{n-1}$?

3. <u>**A decision variant of Problem Ω.**</u>

Decision question: Does there exist a Ω coterie of G of cardinality $\geq 2^{n-1}$?

In the next section we give an algorithm that will prove Problem Λ is in the class ***NP***. By analogy, we will also show that the other two problems (of Subsection 4.1) are also in ***NP***.

5. Algorithm to Show Problem Λ is in *NP*

The following algorithm will be referred to as Λ-in-***NP***. The input is (G, n). G and n, as well as the decision question, the certificate candidate $C(G)$ and the possible outputs, are given in Subsection 4.2.

<u>Algorithm Λ-in-***NP***</u>

BEGIN
1. **if** $n \in \mathbb{N}$
2. **then** print "YES, exists a major Λ coterie of G" and STOP
3. **else** print "NO, such a graph has no coteries" and STOP
4. **endif**

STOP

Proposition 5.1. The algorithm Λ-in-***NP*** is feasible and correct.

Proof. Line 1 of the algorithm checks if $n \in \mathbb{N}$. This check clearly terminates in a finite number of steps.

The possible outputs are all accounted for in two lines - line 2 and line 3. So the algorithm returns only finitely many outputs. Printing each decision clearly terminates in a finite number of steps. Consequently, Λ-in-***NP*** is feasible.

Next, we assert its correctness. If $n \in \mathbb{N}$, then the algorithm decides YES. This is correct by Proposition 2.1.

If $n \notin \mathbb{N}$, then the algorithm decides NO. This output is correct since a graph of such an order n has no vertices and hence no coteries at all. •

Proposition 5.2. Given an input (G, n), Λ-in-***NP*** runs in polynomial time in n.

Proof. The total number (say, T_Λ) of steps executed by the algorithm Λ-in-***NP*** is the sum of the numbers of steps for all the lines executed. Suppose that one execution of the line j requires t_j steps and that this line is executed exactly r_j times when the algorithm is executed once. Then $t_j r_j$ is the number of steps consumed by the line j in one execution of the algorithm.

In one execution of the algorithm, each line is executed once if at all. Hence, for $j = 1, \ldots, 4$, $t_j r_j = t_j$.

We suppose each **endif** line takes constant time, independent of n.

The number of steps required for line 1 is at most n^2. Likewise for lines 2 and 3. Hence $T_\Lambda \leq 3n^2 + 1$. •

Proposition 5.3. To each instance G of Problem Λ there is a certificate that is verified by Λ-in-***NP*** in polynomial time in the size (n) of G.

Proof. $C(G) = n \in \mathbb{N}$ is the required certificate.

Proposition 5.4. Suppose replacements are done in the name and line 2 of the algorithm Λ-in-***NP*** to obtain two other algorithms as folllows:
(i) Λ is replaced by $\mathcal{E}$ to get an algorithm named Algorithm $\mathcal{E}$-in-***NP***.

 doi:10.20944/preprints202507.2089.v1

(ii) Λ is replaced by Ω to get Algorithm Ω-in-***NP***.

Then, like Λ-in-***NP***, each of the resulting algorithms in (i) and (ii) above satisfies the propositions 5.1 through 5.3.

Proof. Straightforward. •

6. *NP*, *P*, the Problems Λ, $\mathcal{E}$ and Ω

Proposition 6.1. 2^{n-1} is unbounded above as n increases over $\mathbb{N}$.

Proof. Straightforward.

Proposition 6.2. The problems Λ, $\mathcal{E}$ and Ω are all in ***NP***.

Proof. Consequence of Proposition 5.1 through Proposition 5.4.

Proposition 6.3. None of the problems Λ, $\mathcal{E}$ and Ω can be in ***P***.

Proof. It suffices to show Problem Λ is not in ***P***. Suppose $\mathcal{A}(\Lambda)$ is a given feasible algorithm that outputs the superior Λ coterie of the input graph G of order n. This output coterie should be either $\Lambda^+(G)$ or $\Lambda^-(G)$, but we shall name it $\mathcal{C}(G)$ for convenience. Let $|\mathcal{C}(G)| = q$. Then $q \geq 2^{n-1}$ (by Proposition 2.2). Each of the q members of $\mathcal{C}(G)$ is an atomic sub-output of $\mathcal{C}(G)$. Name these sub-outputs $S_1, \ldots, S_q$ in the order that $\mathcal{A}(\Lambda)$ follows in computing the members of $\mathcal{C}(G)$.

For $j = 1, \ldots, q-1$, having taken t_j steps for only the computation of S_j, suppose $\mathcal{A}(\Lambda)$ takes another t_{j+1} steps to compute S_{j+1}; in other words, once $\mathcal{A}(\Lambda)$ executes t_j steps to compute S_j then beginning with the next step $\mathcal{A}(\Lambda)$ executes t_{j+1} steps to compute S_{j+1}, allowing that any of the already-computed sub-outputs S_1 through S_j may be used anywhere in the computation of S_{j+1}. Obviously, then, each $t_j \geq 1$ and $t_q \geq 1$.

If T_Λ is the total number of steps taken by $\mathcal{A}(\Lambda)$ to compute and output the members S_1 through S_q of $\mathcal{C}(G)$, then $T_\Lambda \geq t_1 + \cdots + t_q \geq q \geq 2^{n-1}$. By Proposition 6.1, $\mathcal{A}(\Lambda)$ cannot run in polynomial time (in n), from which the conclusion follows. •

7. Other Possible Coteries

Let $G = (V, E)$ be a simple loop-free graph. There are coteries of G other than the ones seen in Section 2, as shown in the following subsections.

7.1. Independence Coteries

Let $S \in \mathcal{P}(V) - \{\phi\}$. Then S is an *independent set* of G if no two elements of S are adjacent in G.

The set of all $S \in \mathcal{P}(V) - \{\phi\}$ such that S is an independent set of G is the *independence coterie* of G, denoted by $\mathcal{I}^+(G)$.

The set of all $S \in \mathcal{P}(V) - \{\phi\}$ such that S is not an independent set of G is the *non-independence coterie* of G, denoted by $\mathcal{I}^-(G)$.

Clearly $\mathcal{I}^+(G) \cup \mathcal{I}^-(G) = \mathcal{P}(V) - \{\phi\}$ and $\mathcal{I}^+(G) \cap \mathcal{I}^-(G) = \phi$.

Either of $\mathcal{I}^+(G)$ and $\mathcal{I}^-(G)$ is an $\mathcal{I}$ *coterie* of G. These two coteries together are referred to as *the pair of antithetic* $\mathcal{I}$ *coteries* (of G).

7.2. Vc Coteries

Let $S \in \mathcal{P}(V) - \{\phi\}$. Then S is a *vertex cover* of G if for every edge $\{x, y\}$ of G it is true that either $x \in S$ or $y \in S$.

The set of all $S \in \mathcal{P}(V) - \{\phi\}$ such that S is a vertex cover of G is the *vc coterie* of G, denoted by $\mathcal{B}^+(G)$.

 doi:10.20944/preprints202507.2089.v1

The set of all $S \in \mathcal{P}(V) - \{\phi\}$ such that S is not a vertex cover of G is the *non-vc coterie* of G, denoted by $\mathcal{B}^-(G)$.

Clearly $\mathcal{B}^+(G) \cup \mathcal{B}^-(G) = \mathcal{P}(V) - \{\phi\}$ and $\mathcal{B}^+(G) \cap \mathcal{B}^-(G) = \phi$.

Either of $\mathcal{B}^+(G)$ and $\mathcal{B}^-(G)$ is a $\mathcal{B}$ *coterie* of G. These two coteries together are referred to as *the pair of antithetic* $\mathcal{B}$ *coteries* (of G).

7.3. More Coterie Problems

Along the lines of the problems Λ, $\mathcal{E}$ and Ω presented in Subsection 4.1, problems that are in ***NP*** but not in ***P*** can be formulated in the coteries seen in Subsection 7.1 and Subsection 7.2.

8. Conclusions

$\boldsymbol{P} \neq \boldsymbol{NP}$ follows from Proposition 6.2 and Proposition 6.3.

Acknowledgments: This research was fully funded by Niels Abel Foundation (NAF), Palakkad, Kerala State, India. The authors are especially grateful to everyone of the referees – all anonymous – who returned highly favourable reports about the correctness of the proofs in this article, as a result of which NAF gave the authors substantial financial rewards.

References

1. Bovet, D. P. and Crescenzi, P., 1994, Introduction to the theory of complexity, Prentice Hall, London, UK.
2. Cook, S., 2000, The P versus NP problem, Available online: Http://www.claymath.org/millennium/P vs NP/pvsNP.pdf.
3. Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C., 2009, Introduction to Algorithms, MIT Press, USA.
4. Dharmarajan, R. and Ramachandran, D., A problem in Moon-Moser graphs to show *P* does not equal *NP*, *Glob. J. Pure Appl. Math.* **21** (1) (2025) 51-73.
5. Dharmarajan, R. and Ramachandran, D., A problem in power sets shows *P* does not equal *NP*, Available online: Http://www.preprint.org. Doi: 10.20944/preprints202503.1321.v2
6. J.M. Harris, J.L. Hirst, M.J. Mossinghoff, Combinatorics and Graph Theory, second ed., Springer, New York, 2008. doi: 10.1007/978-0-387-79711-3
7. Savage, J. E., 1998, Models of Computation, Addison Wesley, Reading, USA.
8. Stoll, R. R., 2012, Set Theory and Logic, Courier Corporation, USA.
9. Trakhtenbrot, B. A., 1984, "A Survey of Russian Approaches to Perebor (Brute-Force Search) Algorithms," IEEE Ann. Hist. Comput. 6(4), pp. 384-400.
10. Wilf, H. S., 1994, Algorithms and Complexity, Internet Edition, Summer.
11. Woeginger, G. J., 2003, "Exact algorithms for NP-hard problems: A survey," Combinatorial Optimization - Proc. 5th Intl. Workshop, Aussois, France, pp. 185–207.