

Article

Not peer-reviewed version

Three Problems in Graph Imbedding to Show P Does Not Equal NP

[R. Dharmarajan](#)^{*} and D. Ramachandran

Posted Date: 27 February 2026

doi: 10.20944/preprints202602.1776.v1

Keywords: algorithm; polynomial time; graph imbedding; cliques; Hamiltonian graphs; dominating sets; classes P and NP



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Three Problems in Graph Imbedding to Show P Does Not Equal NP

R. Dharmarajan * and D. Ramachandran

Niels Abel Foundation, Palakkad 678001, Kerala, India

* 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 $P = NP$?* 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? In this article we formulate three problems in graph imbedding to show $P \neq NP$.

Keywords: algorithm; polynomial time; graph imbedding; cliques; Hamiltonian graphs; dominating sets; classes P and NP

MSC: 2020; 05C38; 05C45; 05C69; 11Y16; 68W40

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 thirteen 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 on which the three problems hinted at in the abstract are formulated.

The first of these problems is discussed in Section 3 through Section 6; the second in Section 7 through Section 9; and the third in Section 10 through Section 12.

The discussions of the three problems also include algorithms (one under each problem) that contribute to proving $P \neq NP$. Each problem is shown to be in NP but not in P (Section 6, Section 9 and Section 12). The concluding remarks are in Section 13.

These 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 [1,3–5].
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 [4,5].
8. Atomic sub-outputs of solutions [4,5].
9. Bijections (or, bijective maps) [6,8].

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 are well-formed definitions that decide conclusively whether or not (i) a given object is an element of X and (ii) two given elements of X are distinct or the same. Also, such definitions are made explicit when necessary. 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 necessitates $E = \phi$, 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 (it is understood that) $x \in G$ and $y \in G$.

The *order* of G is denoted by $|G|$ and is defined to be the number of vertices in G - i.e., $|G| = |V|$. The *girth* of G is the number of edges in G and is denoted by $e(G)$ or by e as convenient.

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 $\frac{n(n-1)}{2}$ [6]. All the graphs considered in this article are assumed simple and loop-free with positive orders.

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 $|N_G(x)|$ ($= |N(x)|$) is the *degree* of x in G , and is denoted by $dx(G)$ or dx .

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 $|J| = |G|$ (i.e., $|W| = |V|$) 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, providing no ambiguity arises. In particular, if $S = V - \{x\}$ then $G[S]$ will also 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* (written $G_1 \cong G_2$) if there exists a bijection [8] $f: V_1 \rightarrow V_2$ 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.

A graph $G = (V, E)$ is said to be *imbedded* in a graph H if H contains a subgraph that is isomorphic to G . G is *span-imbedded* in H if (i) G is imbedded in H and (ii) $|G| = |H|$. Every graph is span-imbedded in itself.

2.1. Importing Edges into Induced Subgraphs

Let $G = (V, E)$ and $S \in \mathcal{P}(V) - \{\phi\}$. Suppose $\{x_1, y_1\}, \dots, \{x_q, y_q\}$ are distinct members of $\mathcal{P}(V) - \{\phi\}$, each of cardinality 2, such that for each $j = 1, \dots, q$:

- (i) $x_j \in S$ and $y_j \in S$ and
- (ii) x_j and y_j are nonadjacent in the induced subgraph $G[S]$ of G .

Then for each $j = 1, \dots, q$, an edge can be introduced between x_j and y_j . Obviously none of these q edges is in G . Each of these q edges introduced this way into $G[S]$ will be called *an edge imported into $G[S]$* (or, simply, an *imported edge*, if the graph into which this edge is introduced is clear from the context). Clearly these q imported edges result in a graph - say, H - such that V is the vertex set of H and G is span-imbedded in H .

Any activity of importing an edge between two nonadjacent vertices of an induced subgraph $G[S]$ will be called a *legal edge-import*. Obviously, then, if a legal edge-import is done between two vertices x and y of $G[S]$ then no more edges are allowed to be imported into $G[S]$ between x and y , since we insist that every graph in this article be simple and loop-free. Every edge-import considered in this article will be assumed legal.

2.2. Note on Importing Edges

Let $G = (V, E)$ be a graph. Throughout this article, the following are to be noted in the context of importing edges into G .

- (i) An edge will be imported only into an induced subgraph of G .
- (ii) If an edge $\{x, y\}$ is given to be imported into $G[S]$ for some $S \in \mathcal{P}(V) - \{\phi\}$, then x and y are non-adjacent vertices of $G[S]$ (and hence non-adjacent in G) prior to the import of this edge.
- (iii) We have a graph H resulting from this import of the edge $\{x, y\}$ such that G is span-imbedded in H .
- (iv) Hereafter, given a graph G , any import of finitely many edges into any induced subgraph of G will be assumed legal, so that the graph resulting from this import process will, like G , be simple and loop-free.

3. Span-Imbedding and Cliques (SIC)

Throughout this section, assume $G = (V, E)$ and $|G| \geq 1$. If $S \in \mathcal{P}(V) - \{\phi\}$ then S is a *clique* of G if either (i) $|S| > 1$ and the vertices of S are pairwise adjacent in G or (ii) $|S| = 1$. G is a *complete graph* if V is a clique of G .

Proposition 3.1. Let $S \in \mathcal{P}(V) - \{\phi\}$ and $e(S)$ denote the girth of $G[S]$. Then a finite number of edges can be imported into $G[S]$ so that:

- (C-1) G is span-imbedded in a graph $H = (V, F)$ (with $E \subseteq F$) and
- (C-2) S is a clique of H - i.e., the induced subgraph $H[S]$ (of H) is a complete graph.

Proof. Suppose S is a clique of G . Then importing no edge into $G[S]$ leaves G span-imbedded in itself (meaning $H \cong G$) from which (C-1) and (C-2) are immediate.

Suppose S is not a clique of G . Let $|S| = p$ and $t = \frac{p(p-1)}{2} - e(S)$. Clearly $t > 0$. Further, exactly t edges can be imported into $G[S]$ such that the p elements of S become pairwise adjacent with no loops and no multiple edges.

These t edges can also be deemed to have been imported into G . Let E_1 denote the set of these t imported edges and let $F = E \cup E_1$. Then the import of these t edges results in the graph $H = (V, F)$ with $E \subsetneq F$. (C-1) and (C-2) follow immediately. •

Proposition 3.2. Let $S \in \mathcal{P}(V) - \{\phi\}$. Then the number of edges that can be imported into $G[S]$ is not more than $\frac{p(p-1)}{2} - e(S)$ where p and $e(S)$ are, respectively, $|S|$ and the girth of $G[S]$.

Proof. Let $t = \frac{p(p-1)}{2} - e(S)$. Upon importing t edges into $G[S]$, we have a graph H such that (C-1) and (C-2) of Proposition 3.1 are true. Then no more edges can be legally imported into $G[S]$ since (by (iv) of Subsection 2.2) H and $H[S]$ are required to be simple and loop-free. •

Corollary 3.3. The number of edges that can be imported into G is not more than $\frac{n(n-1)}{2} - e$ where $n = |G|$ and e is the girth of G .

Proof. Set $S = V$ in Proposition 3.2. •

Proposition 3.4. Let $|G| = n$ and e be the girth of G . Let $S \in \mathcal{P}(V) - \{\phi\}$, $|S| = p$ and $e(S)$ be the girth of $G[S]$. Then $\frac{p(p-1)}{2} - e(S) \leq \frac{n(n-1)}{2} - e$.

Proof. We use induction on n . Let $T(n)$ denote the statement to prove. For $n = 1$, we have $e = 0$, $p = 1$ and $e(S) = 0$, whence $\frac{p(p-1)}{2} - e(S) = 0 = \frac{n(n-1)}{2} - e$, proving $T(1)$.

Assume $T(q)$ is true. We next prove $T(q+1)$ - i.e., when $n = q+1$. We consider the following cases.

Case 1: $p = q+1$. Here $e(S) = e$, whence $\frac{p(p-1)}{2} - e(S) = \frac{n(n-1)}{2} - e$ is immediate.

Case 2: $p < q+1$. Let $z \in V - S$ and let $J = (W, F)$ denote the induced subgraph $G - z$ and let e_1 be the girth of J . Note that $W = V - \{z\}$ so that $|J| = q$. If dz is the degree of z in G , then obviously $dz \leq q$ and $e_1 = e - dz$. By induction hypothesis, $\frac{p(p-1)}{2} - e(S) \leq \frac{q(q-1)}{2} - e_1$ for J (owing to $S \in \mathcal{P}(W) - \{\phi\}$). This gives $\frac{p(p-1)}{2} - e(S) \leq \frac{q(q-1)}{2} - (e - dz) = \frac{q(q-1)}{2} - e + dz \leq \frac{q(q-1)}{2} - e + q$.

Now, $\frac{q(q-1)}{2} - e + q = \frac{q(q-1)}{2} - e + \frac{2q}{2} = \frac{(q+1)q}{2} - e$. Using $n = q+1$, it is straightforward that $\frac{p(p-1)}{2} - e(S) \leq \frac{n(n-1)}{2} - e$, completing the induction. •

Proposition 3.5. To each $S \in \mathcal{P}(V) - \{\phi\}$, there exists a unique $t \in \mathbb{W}$ such that the import of t edges into $G[S]$ endows the properties (C-1) and (C-2) (seen in the statement of Proposition 3.1) to G and S , respectively.

Proof. From the proof of Proposition 3.1, the existence of $t \in \mathbb{W}$ that endows (C-1) to G and (C-2) to S is clear, as is $t = \frac{p(p-1)}{2} - e(S)$ (where $p = |S|$). More than t edges cannot be imported legally into $G[S]$ (Proposition 3.2) whereas if fewer than t edges are imported into $G[S]$ then S will not acquire the property (C-2), whence t is unique for S . •

3.1. Clique-Upgradation Number

The unique t corresponding to S in Proposition 3.5 is the *clique-upgradation number* of S , and is denoted by $\omega^+(S)$.

Proposition 3.6. (i) Each nonempty subset S of V has a clique-upgradation number $\omega^+(S) \in \mathbb{W}$.

(ii) $\omega^+(S) = \frac{p(p-1)}{2} - e(S)$ where $p = |S|$ and $e(S)$ denotes the girth of $G[S]$.

Proof. (i) follows from Proposition 3.1 and the definition of clique-upgradation number.

(ii) Clearly $\omega^+(S) \leq \frac{p(p-1)}{2} - e(S)$. Suppose the conclusion were false. Then (by Proposition 3.2) we would have $\omega^+(S) = \frac{p(p-1)}{2} - e(S) - q$ for some $q \in \mathbb{N}$. Let the graph resulting from the import of $\omega^+(S)$ edges (into $G[S]$) be H . Then $H[S]$, which by the definition of $\omega^+(S)$ is complete after the edge-import process, would have $e(S) + \frac{p(p-1)}{2} - e(S) - q$ edges, which number is less than $\frac{p(p-1)}{2}$. But this patently contradicts the completeness of $H[S]$. •

Let G be of order n and girth e . For G , we define the set $L[G]$ as $L[G] = \left\{ k \in \mathbb{W} : 0 \leq k \leq \frac{n(n-1)}{2} - e \right\}$. For each $k \in L[G]$, we define the set $G^{\omega^+}(k)$ as $G^{\omega^+}(k) = \{ S \in \mathcal{P}(V) - \{\phi\} : \omega^+(S) = k \}$. Two given members of $G^{\omega^+}(k)$ are deemed distinct if they are distinct nonempty subsets of V . The following are immediate:

- (i) $G^{\omega^+}(k)$ is a finite set for each $k \in L[G]$.
- (ii) There exists $k \in L[G]$ such that $G^{\omega^+}(k)$ is nonempty.
- (iii) It may happen that $G^{\omega^+}(m) = \phi$ for some $m \in L[G]$.

For the remainder of this section, let $G = (V, E)$ be of order n and girth e , and let $r = \frac{n(n-1)}{2} - e$.

Proposition 3.7. (i) The set union $\bigcup_{k=0}^r G^{\omega^+}(k)$ equals $\mathcal{P}(V) - \{\phi\}$.

(ii) If $k_1, k_2 \in L[G]$ with $k_1 \neq k_2$ then $G^{\omega^+}(k_1) \cap G^{\omega^+}(k_2) = \phi$.

Proof. (i) $\bigcup_{k=0}^r G^{\omega^+}(k) \subseteq \mathcal{P}(V) - \{\phi\}$ is clear from the definition of $G^{\omega^+}(k)$.

On the other hand, suppose $S \in \mathcal{P}(V) - \{\phi\}$ with $|S| = p$ and let $e(S)$ be the girth of $G[S]$. By Proposition 3.2 and the definition of $\omega^+(S)$ it follows $\omega^+(S) = \frac{p(p-1)}{2} - e(S)$. Invoking Proposition 3.4, we have $S \in G^{\omega^+}(m)$ for some $m \in \mathbb{W}$ with $0 \leq m \leq r$, from which $\mathcal{P}(V) - \{\phi\} \subseteq \bigcup_{k=0}^r G^{\omega^+}(k)$ as well.

(ii) If $S \in G^{\omega^+}(k_1)$ then $\omega^+(S) = k_1 \neq k_2$, whence $S \notin G^{\omega^+}(k_2)$. •

Corollary 3.8. $\sum_{k=0}^r |G^{\omega^+}(k)| = 2^n - 1$.

Proof. Consequence of Proposition 3.6 and Proposition 3.7. •

Proposition 3.9. $|G^{\omega^+}(k)| \geq \frac{2^n - 1}{r + 1}$ for some $k \in L[n]$.

Proof. If each of the $r + 1$ non-negative integers $|G^{\omega^+}(0)|, \dots, |G^{\omega^+}(r)|$ were less than $\frac{2^n - 1}{r + 1}$ then we would have $\sum_{k=0}^r |G^{\omega^+}(k)| < (r + 1) \frac{2^n - 1}{r + 1}$, from which $\sum_{k=0}^r |G^{\omega^+}(k)| < 2^n - 1$. But this contradicts Corollary 3.8. •

4. Computation and Decision Variants of a Problem in SIC

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 \mathcal{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.

We shall be concerned only with computation and decision variants of the three problems we discuss in the coming sections. It is common to formulate a computation problem \mathcal{Q} as a decision problem to find out if \mathcal{Q} is in NP (see [2–5,9,11]).

4.1. Computation Variant of the Problem PSIC

The following is a computation variant of the problem we name PSIC:
If $G = (V, E)$ is a graph of order n and girth e , then find a nonempty set $G^{\omega^+}(k) \subseteq \mathcal{P}(V) - \{\phi\}$ such that $|G^{\omega^+}(k)| \geq \frac{2^n - 1}{r + 1}$ where $r = \frac{n(n-1)}{2} - e$ and $0 \leq k \leq r$.

4.2. A Decision Variant of PSIC

Inputs: (i) Graph $G = (V, E)$, (ii) $n = |G| \in \mathbb{N}$, (iii) $e = |E|$, (iv) $r = \frac{n(n-1)}{2} - e$.

Question: Does there exist a nonempty set $G^{\omega^+}(k) \subseteq \mathcal{P}(V) - \{\phi\}$ such that $|G^{\omega^+}(k)| \geq \frac{2^n - 1}{r + 1}$ for some $k \in \mathbb{W}$ with $0 \leq k \leq r$?

Output: YES (a desired set $G^{\omega^+}(k)$ exists) or NO (no such set $G^{\omega^+}(k)$ exists), as appropriate.

Note. There is no need of a certificate candidate since $n \in \mathbb{N}$ ensures the output is YES by dint of Proposition 3.9. As for the inputs (i) through (iv), it is mandatory that they be free from any error, as also that they be logically consistent with one another.

5. Algorithm PSIC-in- NP

The following algorithm will be referred to as PSIC-in- NP . The input is (G, n, e, r) . G, n, e and r , as well as the decision question and the required output, are given in Subsection 4.2.

Algorithm PSIC-in- NP

BEGIN

1. print “YES, there exists $G^{\omega^+}(k) \subseteq \mathcal{P}(V) - \{\phi\}$ with $|G^{\omega^+}(k)| \geq \frac{2^n - 1}{r + 1}$ since G is of positive order” and STOP

STOP

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

Proof. The algorithm evidently returns only one output. Printing each decision clearly terminates in a finite number of steps. Consequently, PSIC-in- NP is feasible.

Next, we assert its correctness. Since $n \in \mathbb{N}$, the YES decision is correct by Proposition 3.9. •

Proposition 5.2. Given an input (G, n, e, r) , PSIC-in- NP runs in polynomial time in n .

Proof. The total number (say, T_c) of steps executed by the algorithm PSIC-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, line 1 is executed once. Hence, $t_1 r_1 = t_1$. Then $T_c = t_1$.

The number of steps required for line 1 is at most n^2 . Hence $T_c \leq n^2$. •

6. PSIC Is in NP but Not in P

Let G, n, e and r be as in Subsection 4.2.

Proposition 6.1. The ratio $\frac{2^n - 1}{r + 1}$ cannot be bounded by an^b for any positive real constants a and b as n increases over \mathbb{N} .

Proof. Were the conclusion false, then there would exist real constants $c > 1$ and $d > 1$ such that $\frac{2^n - 1}{r + 1} \leq cn^d$ for all $n \in \mathbb{N}$. Simple calculations from here lead to $2^n \leq (2c + 1)n^{d+2}$. This leads to $\frac{n}{\log_2 n} \leq d + 3$ for all $n \geq 2c + 1$. But this contradicts the fact that $\frac{n}{\log_2 n}$ is unbounded as n increases over \mathbb{N} . •

Proposition 6.2. The problem PSIC is in NP.

Proof. Consequence of Proposition 5.1 and Proposition 5.2.

Proposition 6.3. The problem PSIC is not in P.

Proof. Let \mathcal{C} be a given feasible algorithm that outputs a desired set $G^{\omega^+}(k)$ (where G is the given input graph of order n).

Let $|G^{\omega^+}(k)| = q$. Note that $q \geq \frac{2^n - 1}{r + 1}$ by Proposition 3.9. The q members of $G^{\omega^+}(k)$ are the atomic sub-outputs [4,5] of this solution to the instance G . Name these sub-outputs S_1, \dots, S_q in the order that \mathcal{C} follows in computing $G^{\omega^+}(k)$.

For $j = 1, \dots, q - 1$, having taken t_j steps for only the computation of S_j , suppose \mathcal{C} takes another t_{j+1} steps to compute S_{j+1} ; in other words, once \mathcal{C} executes t_j steps to compute S_j , then beginning with the next step, \mathcal{C} 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 $T_{\mathcal{C}}$ is the total number of steps taken by \mathcal{C} to compute and output the members S_1 through S_q of $G^{\omega^+}(k)$, then $T_{\mathcal{C}} \geq t_1 + \dots + t_q \geq q \geq \frac{2^n - 1}{r + 1}$. By Proposition 6.1, \mathcal{C} cannot run in polynomial time (in n), from which the conclusion follows. •

7. Span-Imbedding and Dominating Sets (SID)

Throughout this section, assume $G = (V, E)$ and $|G| \geq 1$. A nonempty subset D of V is a *dominating set* of G if either $D = V$ or each element of $V - D$ is adjacent (in G) to some element of D .

Proposition 7.1. Let $D \in \mathcal{P}(V) - \{\emptyset\}$. Then a finite number of edges can be imported into G so that:

(D-1) G is span-imbedded in a graph $H = (V, F)$ (with $E \subseteq F$) and

(D-2) D is a dominating set of H .

Proof. Suppose D is a dominating set of G . Then importing no edge into G leaves G span-imbedded in itself (meaning $H \cong G$), from which (D-1) and (D-2) follow.

Suppose D is not a dominating set of G . Then let A denote the set of all the elements of $V - D$ that are not adjacent to any element of D . Let $|A| = p$ and write $A = \{y_1, \dots, y_p\}$. Clearly $p > 0$. Fix an element $x \in D$. Import the p edges $\{x, y_j\}$, ($j = 1, \dots, p$) into G . (D-1) and (D-2) follow immediately. •

Proposition 7.2. To each $D \in \mathcal{P}(V) - \{\emptyset\}$, there corresponds a least $t \in \mathbb{W}$ such that the import of t edges into G endows the properties (D-1) and (D-2) (seen in the statement of Proposition 7.1) to G and D , respectively.

Proof. Consequence of Proposition 7.1. •

7.1. Domination-Upgradation Number

The least $t \in \mathbb{W}$ corresponding to D in Proposition 7.2 is the *domination-upgradation number* (or, *dom-upgradation number*) of D , and is denoted by $\gamma^+(D)$.

Proposition 7.3. (i) Each nonempty subset D of V has a dom-upgradation number $\gamma^+(D)$.
(ii) $\gamma^+(D) \leq |V - D|$.

Proof. (i) Follows from Proposition 7.2 and the definition of dom-upgradation number.

(ii) If D is a dominating set of G then $\gamma^+(D) = 0 \leq |V - D|$.

So suppose D is not a dominating set of G . Let A be the set $\{y_1, \dots, y_p\}$ seen in the proof of Proposition 7.1. Clearly $A \subseteq V - D$ and so $|A| \leq |V - D|$.

Importing an edge from each element of A to a fixed element of D results in a graph H such that D is a dominating set of H . Next, if fewer than p edges are imported into G then at least one element of A will remain nonadjacent to every element of D and so D cannot be a dominating set of H ; so $\gamma^+(D) \geq |A|$.

Next, having imported an edge from each element of A to the fixed element of D , avoiding the import of any other edge results in a graph H of which D is clearly a dominating set. So $\gamma^+(D) \leq |A|$.

Thus $\gamma^+(D) = |A| \leq |V - D|$. •

Let G be of order n and girth e . For G , let the set $L[G]$ be as defined immediately after the proof of Proposition 3.6. For each $k \in L[G]$, we define the set $G^{\gamma^+}(k)$ as $G^{\gamma^+}(k) = \{D \in \mathcal{P}(V) - \{\phi\} : \gamma^+(D) = k\}$. Two given members of $G^{\gamma^+}(k)$ are deemed distinct if they are distinct nonempty subsets of V . The following are immediate:

- (i) $G^{\gamma^+}(k)$ is a finite set for each $k \in L[G]$.
- (ii) There exists $k \in L[G]$ such that $G^{\gamma^+}(k)$ is nonempty.
- (iii) It may happen that $G^{\gamma^+}(m) = \phi$ for some $m \in L[G]$.

For the remainder of this section, let $G = (V, E)$ be of order n and girth e , and let $r = \frac{n(n-1)}{2} - e$.

Proposition 7.4. (i) The set union $\bigcup_{k=0}^r G^{\gamma^+}(k)$ equals $\mathcal{P}(V) - \{\phi\}$.
(ii) If $k_1, k_2 \in L[G]$ with $k_1 \neq k_2$ then $G^{\gamma^+}(k_1) \cap G^{\gamma^+}(k_2) = \phi$.

Proof. (i) $\bigcup_{k=0}^r G^{\gamma^+}(k) \subseteq \mathcal{P}(V) - \{\phi\}$ is clear from the definition of $G^{\gamma^+}(k)$.

On the other hand, suppose $D \in \mathcal{P}(V) - \{\phi\}$. Let $|D| = p$. Since $0 \leq \gamma^+(D) \leq r$ (in view of Corollary 3.3) and $p \leq n$, we have $D \in G^{\gamma^+}(m)$ for some $m \in \mathbb{W}$ with $0 \leq m \leq r$, from which $\mathcal{P}(V) - \{\phi\} \subseteq \bigcup_{k=0}^r G^{\gamma^+}(k)$ as well.

(ii) If $D \in G^{\gamma^+}(k_1)$ then $\gamma^+(D) = k_1 \neq k_2$, from which $D \notin G^{\gamma^+}(k_2)$. •

Corollary 7.5. $\sum_{k=0}^r |G^{\gamma^+}(k)| = 2^n - 1$.

Proof. Consequence of Proposition 7.3 and Proposition 7.4. •

Proposition 7.6. $|G^{\gamma^+}(k)| \geq \frac{2^n - 1}{r + 1}$ for some $k \in L[n]$.

Proof. If each of the $r + 1$ non-negative integers $|G^{\gamma^+}(0)|, \dots, |G^{\gamma^+}(r)|$ were less than $\frac{2^n - 1}{r + 1}$ then we would have $\sum_{k=0}^r |G^{\gamma^+}(k)| < (r + 1) \frac{2^n - 1}{r + 1}$, from which $\sum_{k=0}^r |G^{\gamma^+}(k)| < 2^n - 1$. But this contradicts Corollary 7.5. •

7.2. Computation Variant of a Problem in SID

The following is a computation variant of the problem we name PSID:
 If $G = (V, E)$ is a graph of order n and girth e then find a nonempty set $G^{\gamma^+}(k) \subseteq \mathcal{P}(V) - \{\phi\}$ such that $|G^{\gamma^+}(k)| \geq \frac{2^n - 1}{r + 1}$ where $r = \frac{n(n - 1)}{2} - e$ and $0 \leq k \leq r$.

7.3. A Decision Variant of PSID

Inputs: (i) Graph $G = (V, E)$, (ii) $n = |G| \in \mathbb{N}$, (iii) $e = |E|$, (iv) $r = \frac{n(n - 1)}{2} - e$.

Question: Does there exist a nonempty set $G^{\gamma^+}(k) \subseteq \mathcal{P}(V) - \{\phi\}$ such that $|G^{\gamma^+}(k)| \geq \frac{2^n - 1}{r + 1}$ for some $k \in \mathbb{W}$ with $0 \leq k \leq r$?

Output: YES (a desired set $G^{\gamma^+}(k)$ exists) or NO (no such set $G^{\gamma^+}(k)$ exists), as appropriate.

Note. There is no need of a certificate candidate since $n \in \mathbb{N}$ ensures the output is YES by dint of Proposition 7.6. As for the inputs (i) through (iv), it is mandatory that they be free from any error, as also that they be logically consistent with one another.

8. Algorithm PSID-in-NP

The following algorithm will be referred to as PSID-in-NP. The input is (G, n, e, r) . G, n, e and r , as well as the decision question and the required output, are given in Subsection 7.3.

Algorithm PSID-in-NP

BEGIN

1. print "YES, there exists $G^{\gamma^+}(k) \subseteq \mathcal{P}(V) - \{\phi\}$ with $|G^{\gamma^+}(k)| \geq \frac{2^n - 1}{r + 1}$ since G is of positive order" and STOP

STOP

Proposition 8.1. The algorithm PSID-in-NP is feasible and correct.

Proof. The algorithm evidently returns only one output. Printing the decision clearly terminates in a finite number of steps. Consequently, PSID-in-NP is feasible.

Next, we assert its correctness. Since $n \in \mathbb{N}$, the YES decision is correct by Proposition 7.6. •

Proposition 8.2. Given an input (G, n, e, r) , PSID-in-NP runs in polynomial time in n .

Proof. The total number (say, T_d) of steps executed by the algorithm PSID-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, line 1 is executed once and so $T_d = t_1$. The number of steps required for line 1 is at most n^2 . Hence $T_d \leq n^2$. •

9. PSID Is in NP but not in P

Proposition 9.1. The problem PSID is in NP.

Proof. Consequence of Proposition 8.1 and Proposition 8.2. •

Proposition 9.2. The problem PSID is not in P.

Proof. Let \mathcal{D} be a given feasible algorithm that outputs a desired set $G^{\gamma^+}(k)$ (where G is the given input graph of order n).

Let $|G^{\gamma^+}(k)| = q$. Note that $q \geq \frac{2^n - 1}{r + 1}$ by Proposition 7.6. The q members of $G^{\gamma^+}(k)$ are the atomic sub-outputs of this solution to the instance G . Name these sub-outputs S_1, \dots, S_q in the order that \mathcal{D} follows in computing the members of $G^{\gamma^+}(k)$.

For $j = 1, \dots, q - 1$, having taken t_j steps for only the computation of S_j , suppose \mathcal{D} takes another t_{j+1} steps to compute S_{j+1} ; in other words, once \mathcal{D} executes t_j steps to compute S_j then beginning with the next step \mathcal{D} 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 $T_{\mathcal{D}}$ is the total number of steps taken by \mathcal{D} to compute and output the members S_1 through S_q of $G^{\gamma^+}(k)$, then $T_{\mathcal{D}} \geq t_1 + \dots + t_q \geq q \geq \frac{2^n - 1}{r + 1}$. By Proposition 6.1, \mathcal{D} cannot run in polynomial time (in n), from which the conclusion follows. •

10. Span-Imbedding and Hamiltonian Graphs (SIH)

Throughout this section, assume $G = (V, E)$. A *path* in G is a sequence x_1, \dots, x_k of k distinct vertices of G such that $\{x_j, x_{j+1}\} \in E$ for each $j = 1, \dots, k - 1$. A *cycle* or a *closed path* in G is a path x_1, \dots, x_k in G having the additional properties that $k \geq 3$ and $\{x_k, x_1\} \in E$. Such a cycle is also called a *k-cycle*. A *k-cycle* in G where $k = |G|$ is a *spanning cycle* or *Hamiltonian cycle* in G . G is a *Hamiltonian graph* if there is a Hamiltonian cycle in G .

Proposition 10.1. Let $|G| \geq 3$. Let $S \in \mathcal{P}(V) - \{\phi\}$.

(i) If $|S| \geq 3$, then a finite number of edges can be imported into $G[S]$ so that:

(H-1) G is span-imbedded in a graph $H = (V, F)$ (with $E \subseteq F$) and

(H-2) the induced subgraph $H[S]$ (of H) is a Hamiltonian graph.

(ii) If $|S| \leq 2$ then (H-2) seen in (i) above is never true.

Proof. (i) Suppose $G[S]$ is Hamiltonian. This necessitates $|S| \geq 3$. Then importing no edge into $G[S]$ leaves G is span-imbedded in itself (i.e., $H \cong G$) from which (H-1) and (H-2) follow immediately.

Suppose $G[S]$ is not Hamiltonian. Let $|S| = p$. Write $S = \{y_1, \dots, y_p\}$. For each $j = 1, \dots, p - 1$, if y_j and y_{j+1} are adjacent in $G[S]$ then do not import any edge between y_j and y_{j+1} ; else import one edge between y_j and y_{j+1} .

Next, import an edge between y_p and y_1 if they are not adjacent in $G[S]$; else do not import any edge between the two.

Let E_1 denote the set of all the edges imported into $G[S]$ by the above process. Note that $E \cap E_1 = \phi$. Let $F = E \cup E_1$. We now have a graph $H = (V, F)$.

Clearly, then, G is span-imbedded in H . Further, the vertices y_1, \dots, y_p (in that order) form a Hamiltonian cycle in $H[S]$. (H-1) and (H-2) follow immediately.

(ii) If $|S| \leq 2$ then the elements of S can never form a Hamiltonian cycle by themselves. •

Proposition 10.2. To each $S \in \mathcal{P}(V) - \{\phi\}$ such that $|S| \geq 3$, there corresponds a least $t \in \mathbb{W}$ such that the import of t edges into $G[S]$ endows the properties (H-1) and (H-2) (seen in the statement of Proposition 10.1) to G and $H[S]$, respectively.

Proof. Consequence of Proposition 10.1. •

10.1. Hamilton-Upgradation Number

The least t corresponding to S in Proposition 10.2 is the *Hamilton-upgradation number* or *H-upgradation number* of S , and is denoted by $\theta^+(S)$.

Proposition 10.3. Let $G = (V, E)$ have order $n \geq 3$.

(i) Each nonempty subset S of V such that $|S| \geq 3$ has a H-upgradation number $\theta^+(S)$.

(ii) $\theta^+(S) \leq \frac{p(p-1)}{2} - e(S)$ where $p = |S|$ and $e(S)$ denotes the girth of $G[S]$.

Proof. (i) Follows from Proposition 10.1 and the definition of H-upgradation number.

(ii) Suppose the conclusion were false. Then $\theta^+(S) = \frac{p(p-1)}{2} - e(S) + k$ for some $k \in \mathbb{N}$. Let H be the graph resulting from importing $\frac{p(p-1)}{2} - e(S) + k$ edges into $G[S]$. Then there would be $\frac{p(p-1)}{2} + k$ edges in $H[S]$, a patent impossibility since $H[S]$ is simple and loop-free of order p . •

Let G be of order $n \geq 3$ and girth e . For G , we define the set $L[G]$ as in Subsection 3.1. For each $k \in L[G]$, we define the set $G^{\theta^+}(k)$ as $G^{\theta^+}(k) = \{S \in \mathcal{P}(V) - \{\emptyset\} : \theta^+(S) = k\}$. Two given members of $G^{\theta^+}(k)$ are deemed distinct if they are distinct nonempty subsets of V . The following are immediate:

(i) $G^{\theta^+}(k)$ is a finite set for each $k \in L[G]$.

(ii) There exists $k \in L[G]$ such that $G^{\theta^+}(k)$ is nonempty.

(iii) It may happen that $G^{\theta^+}(m) = \emptyset$ for some $m \in [G]$.

(iv) If $S \in G^{\theta^+}(k)$ for some $k \in L[n]$ then $|S| \geq 3$.

For the remainder of this section, $G = (V, E)$ is of order $n \geq 3$ and girth e , and $r = \frac{n(n-1)}{2} - e$.

Proposition 10.4. (i) The set union $\bigcup_{k=0}^r G^{\theta^+}(k)$ equals $\{S \in \mathcal{P}(V) : |S| \geq 3\}$.

(ii) If $k_1, k_2 \in L[G]$ with $k_1 \neq k_2$ then $G^{\theta^+}(k_1) \cap G^{\theta^+}(k_2) = \emptyset$.

Proof. (i) The inclusion $\bigcup_{k=0}^r G^{\theta^+}(k) \subseteq \{S \in \mathcal{P}(V) : |S| \geq 3\}$ is clear from the definition of $G^{\theta^+}(k)$.

On the other hand, suppose $S \subseteq V$ and $S \neq \emptyset$ with $|S| = p \geq 3$ and let $e(S)$ be the girth of $G[S]$. By Proposition 10.3(ii), $\theta^+(S) \leq \frac{p(p-1)}{2} - e(S)$. Since $p \leq n$, invoking Proposition 3.4 we have $S \in G^{\theta^+}(m)$ for some $m \in \mathbb{W}$ with $0 \leq m \leq r$, from which $\{S \in \mathcal{P}(V) : |S| \geq 3\} \subseteq \bigcup_{k=0}^r G^{\theta^+}(k)$ as well.

(ii) If $S \in G^{\theta^+}(k_1)$ then $\theta^+(S) = k_1 \neq k_2$, from which $S \notin G^{\theta^+}(k_2)$. •

Corollary 10.5. $\sum_{k=0}^r |G^{\theta^+}(k)| = 2^n - 1 - n - \frac{n(n-1)}{2}$.

Proof. Consequence of Proposition 10.3 and Proposition 10.4. •

Proposition 10.6. $|G^{\theta^+}(k)| \geq \frac{2^n - 1 - n - \frac{n(n-1)}{2}}{r+1}$ for some $k \in L[n]$.

Proof. Let Ω_n denote the real number $\frac{2^n - 1 - n - \frac{n(n-1)}{2}}{r+1}$. If $|G^{\theta^+}(k)| < \Omega_n$ for each $k \in L[G]$ then $\sum_{k=0}^r |G^{\theta^+}(k)| < (r+1)\Omega_n$ would ensue, giving $\sum_{k=0}^r |G^{\theta^+}(k)| < 2^n - 1 - n - \frac{n(n-1)}{2}$, patently contradicting Corollary 10.5. •

10.2. Computation Variant of a Problem in SIH

The following is a computation variant of the problem we name PSIH:

If $G = (V, E)$ is a graph of order $n \geq 3$ and girth e then find a nonempty set $G^{\theta+}(k) \subseteq \{S \in \mathcal{P}(V) : |S| \geq 3\}$ such that $|G^{\theta+}(k)| \geq \frac{2^n - 1 - n - \frac{n(n-1)}{2}}{r+1}$ where $r = \frac{n(n-1)}{2} - e$ and $0 \leq k \leq r$.

10.3. A Decision Variant of PSIH

Inputs: (i) Graph $G = (V, E)$, (ii) $n = |G| \in \mathbb{N}$, (iii) $e = |E|$, (iv) $r = \frac{n(n-1)}{2} - e$

Question: Is there a nonempty set $G^{\theta+}(k) \subseteq \{S \in \mathcal{P}(V) : |S| \geq 3\}$ with $|G^{\theta+}(k)| \geq \frac{2^n - 1 - n - \frac{n(n-1)}{2}}{r+1}$ for some $k \in \mathbb{W}$ with $0 \leq k \leq r$?

Certificate candidate: $C(G) = n$.

Output: YES (a desired set $G^{\theta+}(k)$ exists) or NO (no such set $G^{\theta+}(k)$ exists), as appropriate.

11. Algorithm PSIH-in-NP

The following algorithm will be referred to as PSIH-in-NP. The input is (G, n, e, r) . G, n, e and r , as well as the decision question and the required output, are given in Subsection 10.3.

Algorithm PSIH-in-NP

BEGIN

1. **if** $n \geq 3$

2. **then** print "YES, exists $G^{\theta+}(k) \subseteq \{S \in \mathcal{P}(V) : |S| \geq 3\}$ with $|G^{\theta+}(k)| \geq \frac{2^n - 1 - n - \frac{n(n-1)}{2}}{r+1}$ since $n \geq 3$ " and STOP

3. **else** print "NO, no nonempty set $G^{\theta+}(k) \subseteq \{S \in \mathcal{P}(V) : |S| \geq 3\}$ with $|G^{\theta+}(k)| \geq \frac{2^n - 1 - n - \frac{n(n-1)}{2}}{r+1}$ since $n \leq 2$ " and STOP

4. **endif**

STOP

Proposition 11.1. The algorithm PSIH-in-NP is feasible and correct.

Proof. Line 1 of the algorithm checks if $n \geq 3$. 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, PSIH-in-NP is feasible.

Next, we assert its correctness. If $n \geq 3$, then the algorithm decides YES. This is correct by Proposition 10.6.

If $n \leq 2$, then the algorithm decides NO. This output is correct by Proposition 10.1(ii). •

Proposition 11.2. Given an input (G, n, e, r) , PSIH-in-NP runs in polynomial time in n .

Proof. The total number (say, T_n) of steps executed by the algorithm PSIH-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 $j = 1, \dots, 4$, $t_j r_j = t_j$.

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 $T_h \leq 3n^2 + 1$. •

Proposition 11.3. To each instance G of PSIH such that $|G| = n \geq 3$ there is a certificate that is verified by PSIH-in- NP in polynomial time in the size (n) of G .

Proof. $C(G) = n \geq 3$ is the required certificate. •

12. PSIH Is in NP but Not in P

Let G, n, e and r be as in Subsection 10.3.

Proposition 12.1. The ratio $\frac{2^n - 1 - n - \frac{n(n-1)}{2}}{r+1}$ cannot be bounded by an^b for any positive real constants a and b as n increases over \mathbb{N} .

Proof. Were the conclusion false, then $\frac{2^n - 1 - n - \frac{n(n-1)}{2}}{r+1} \leq cn^d$ for some real constants $c > 1$ and $d > 1$ and for all $n \in \mathbb{N}$. Simple calculations from here lead to $2^n < (c+2)n^{2d}$, giving $2^n < n^{2d+1}$ for all $n \in \mathbb{N}$ such that $n > c+2$. Then $\frac{n}{\log_2 n} < 2d+1$ would ensue, But this contradicts the fact that $\frac{n}{\log_2 n}$ cannot be bounded as n increases over \mathbb{N} . •

Proposition 12.2. The problem PSIH is in NP .

Proof. Consequence of Proposition 11.1, Proposition 11.2 and Proposition 11.3. •

Proposition 12.3. The problem PSIH is not in P .

Proof. Let \mathcal{H} be a given feasible algorithm that outputs a desired set $G^{\theta+}(k)$ (where G is the given input graph of order n).

Let $|G^{\theta+}(k)| = q$. Note that $q \geq \frac{2^n - 1 - n - \frac{n(n-1)}{2}}{r+1}$ by Proposition 10.6. The q members of $G^{\theta+}(k)$ are the atomic sub-outputs of this solution to the instance G . Name these sub-outputs S_1, \dots, S_q in the order that \mathcal{H} follows in computing $G^{\theta+}(k)$.

For $j = 1, \dots, q-1$, having taken t_j steps for only the computation of S_j , suppose \mathcal{H} takes another t_{j+1} steps to compute S_{j+1} ; in other words, once \mathcal{H} executes t_j steps to compute S_j then beginning with the next step \mathcal{H} 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 $T_{\mathcal{H}}$ is the total number of steps taken by \mathcal{H} to compute and output the members S_1 through S_q of $G^{\theta+}(k)$, then $T_{\mathcal{H}} \geq t_1 + \dots + t_q \geq q \geq \frac{2^n - 1 - n - \frac{n(n-1)}{2}}{r+1}$. By Proposition 12.1, \mathcal{H} cannot run in polynomial time (in n), from which the conclusion follows. •

13. Conclusions

$P \neq NP$ follows from Proposition 6.2 and Proposition 6.3; or from Proposition 9.1 and Proposition 9.2; or from Proposition 12.2 and Proposition 12.3.

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](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.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.