P vs NP Solution – Advances in Computational Complexity, Status and Future Scope

Amit Sharma
AMIE CSE Research Scholar
The Institution of Engineers (INDIA), India

Sunil Kr. Singh
CSE Department, CCET
Degree Wing, Chandigarh, India

ABSTRACT
The significance & stature of the **P vs NP** problem is so imperative that even the failed attempts at proof have furnished unprecedented breakthroughs and valuable insights. While the scientists and researchers do not expect the problem to be solved in foreseeable future, the **P vs NP** question has been the harbinger of advancement of the theory of computation and complexity theory in particular. Multitude of research papers have been published on number of topics which have begged numerous accolades and awards. This paper presents and highlights a non-technical review of series of complex mathematical research and enlists the notable awards & advances from each subsequent effort. The paper also presents the limitations of existing and proposed techniques and highlights the direction of active future research towards **P vs NP** solution.

Keywords
P vs NP, Cryptography, Complexity theory, P vs NP attempts and limitations, Geometric complexity theory, quantum computing, One-way function, incompleteness theorem.

1. INTRODUCTION
P versus NP is perhaps the most fundamental and most essential contemporary problem in mathematics and computer science, whose relevance and significance with time has grown beyond bounds. In the first half of the twentieth century, pioneering and prominent research papers on decidability, computability & complexity of algorithmic problems led to the development of computational complexity theory. In 1936, Alan Turing in his historic paper *"On computable numbers, with an application to the Entscheidungsproblem"*[1] presented a formal mathematical model of a computation machine Turing Machine which could simulate any algorithm. Thus, he is regarded as the father of modern theoretical computer science.

In 1961, Stephen Cook in his landmark paper *"The complexity of theorem-proving procedure"*[2] introduced the **P vs NP** problem. Since then, researchers, complexity theorists, mathematicians, programmers, and amateurs have been grappling with the legendary problem and still the possibility of the resolution eludes the brightest of minds.

This question is so fundamental that Clay Mathematics Institute, California has named it among 7-millennium prize problems[3] and offered $1 million to anyone who provides a verified proof. With each attempt the complexity of the problem seems to increase, thereby reiterating the general consensus on the limitation of existing techniques, thus reinstating the requirement of a novel technique. Despite the failures to find a proof, the incessant quest for the Holy Grail of mathematics and computer science, **P vs NP** has bequeathed the beacon amidst the haze of uncertainty over computational complexity perspectives.

2. THE P vs NP PROBLEM
Study of algorithms is a fundamental task in computer science. Researchers & programmers aim to develop efficient algorithms for data processing. The efficiency of an algorithm M is measured in terms of complexity function f(n) which outputs the required running time or storage space in term of size n of the input data, relative to a number of key operations it has to manipulate. By bounding the time or space requirement and model of computation deterministic or non-deterministic Turing machine, algorithms have been classified into various complexity classes.

The general classes of mathematical problems which can be solved by a deterministic Turing machine in a polynomial time are classified as P problems. For instance, the time complexity of bubble sort algorithm for an unsorted array of n numbers is f(n2) and thus its complexity lie in P class. So, the class P is the class of decision problems that are easy for computers to solve.

The general classes of a mathematical problem whose solution is efficiently verifiable by a deterministic Turing machine in polynomial time but not easily solved are classified as NP problems. NP stands for non-deterministic polynomial time i.e. they can be solved by non-deterministic Turing machine in polynomial time. For instance, a brute force algorithm for cracking passwords containing n symbols in the worst possible case will have to go through all the permutations of n symbols. The execution time of such problem is not in polynomial time but in exponential time. So, even with fastest imaginable machines, for the modest value of n (say 30) it would take billions of years to find it. However, for a given password, it is very easy to verify if it correct or incorrect.

NP-Hard is the class of problems such that every problem in NP can be reduced to any problem in NP-hard, i.e. complexity of NP-hard problems is greater than or equal to the hardest problems in NP. Some problems in NP-Hard are in NP but there are problems in NP-hard which are outside of NP, some of which may not even be decidable. NP-complete is the class of problems which lie in both NP and NP-Hard.

The **P** versus **NP** problem is to ascertain -

- Either P = NP i.e. problems being easy to solve are the same as problems having solutions that are easy to check. To put it simply, the **P = NP** problem is the search for a way to solve problems that require full exhaustion without actually having to try each of colossal combinations.
- Or $\mathbf{P} \neq \boldsymbol{NP}$ i.e. to prove that there are some problems that are easily verified but not easily solved. In other words, to prove that **NP complete** problem will never have efficient solutions.

After decades of study and research, a legitimate proof still eludes the likes of genius researchers as it does that of the amateur enthusiasts. For around 3000 important NP-complete problems no one has managed to prove P = NP, so practical experience overwhelmingly suggests that P≠ NP. But in absence of a sound mathematical proof, the legitimacy of the assumption remains questionable. So, P vs NP is still open and up for the challenge, providing an opportunity to the sharpest minds to gain an access to instant fame and riches.

3. SIGNIFICANCE OF P vs NP

The relentless fervor, with which mathematicians and researchers have endeavored the profound pursuit of unlocking the enigmatic intricacies of P vs NP reflects its significance. The rationale it draws so much attention is the stunning implications of the answer. The resolution of the problem either way would foster the understanding of the computation problems and quest for efficient algorithms enormously, and have vast practical consequences. The class of NP-complete problems is extremely rich and diverse. There are numerous NP-complete problems in combinatorics, meteorology, economics, biology, mathematics, string processing, mathematical programming, optimization, artificial intelligence, cryptography, operation research, graph theory, game theory, industrial processes etc. whose efficient solution will have far reaching consequences.

If P = NP, its implications are so profound that the humanity will have a giant leap. Every aspect of knowledge in art & science, from learning to the application will be much more efficient. Resource optimization in logistics, human-like neural network in artificial intelligence, automation of mathematical proofs, correct coding of DNA sequence in biology, scientific theory for a given data in physics, market behavior in economics, design of engineering system with given constraints, predicting earthquakes in meteorology, anything & everything will benefit from fast solution of NP problems. But, it will have its serious repercussions too, cryptography will collapse. All security encryption algorithms will be obsolete. All internet communications, E-commerce transactions, encrypted military secrets would be prone to easy hacking as cracking passwords would actually become trivial. Such cryptosystems would need to be altered or replaced by cryptographic solutions independent of P≠NP assertion.

Owing to startling consequences of P = NP, many experts believe that any P = NP proof would not lead directly to the efficient method as it is more likely to be a non-constructive proof, or rendered to be practically inefficient due to the large size of the bounding polynomial.

If P ≠ NP, it would not have dazzling computational benefits as that of P = NP, but would nevertheless characterize a considerable advance in the theory of computation and complexity theory. With P ≠ NP it will be relatively safe to assume the security and privacy of personal information, internet communication, financial institutions and transactions are protected. As this problem has already done it will continue to provide guidance for future research. Focus would shift to finding techniques & approaches to deal with a hardness of practical NP-complete problems that cannot be avoided. Under the assumption P ≠ NP researchers have already started to actively pursue the direction of determination of new techniques and ways. Combination of several approaches such as brute force, parameterized complexity, approximation algorithms for vertex cover, clustering and TSP, heuristics and average case complexity of TSP, graph partitioning, narrowing the problem space, simulated annealing, random algorithms, improved exponential time algorithms, parallelization, dynamic programming etc. is being used to tackle NP-complete problems.

4. ADVANCES IN COMPLEXITY THEORY

The P vs NP problems stems from the limitation of computation in response to abstract mathematics Entscheidungsproblem.

4.1 Entscheidungsproblem: German mathematician David Hilbert, at the 1928 International Congress of Mathematics, asked three fundamental questions about the completeness, consistency, and decidability of mathematics. The third question is famously known by its German name Entscheidungsproblem – "Is every statement in mathematics decidable?" i.e. is there an algorithm which can be applied to every statement that will tell us in finite time whether or not the statement is true or false. Until 1930, Hilbert himself believed and advocated the answer all three problems is "Yes" and that there is no such thing as an unsolvable problem.

4.2 Gödel's incompleteness theorem: In 1931 Kurt Gödel published a paper On Formally Undecidable Propositions of Principia Mathematica and Related Systems I.[4] In this paper, Gödel's first incompleteness theorem shows that in any consistent effectively generated formal system, there are statements which can neither be proved nor disapproved and is thus incomplete. Gödel's second incompleteness theorem states any formal system can prove its own consistency if and only if it is inconsistent.

4.3 Tarski's undefinability theorem: In 1936, following Gödel's undecidability and incompleteness proofs, Alfred Tarski published Tarski's undefinability theorem[5] which says for any strong formal system, the concept of truth in the standard model of the system is not definable within the system.

In 1936, the answer to Entscheidungsproblem in negation was furnished by the independent works of Alonzo Church, Alan Turing, and Emil Post.

4.4 Effective calculability: Using Gödel-Herbrand's general recursion & Stephen Kleene's work, Alonzo Church in his paper An Unsolvable Problem of Elementary Number Theory [6] proved that no computable function exists which can determine equivalence of two given λ-calculus expressions.

4.5 Turing machine: In 1936, Alan Turing published certainly the most celebrated theoretical paper in the history of computing, On Computable Numbers, with an application to the Entscheidungsproblem[1]. In his paper, Turing came up with the concept of computable numbers which are calculable on the universal machine using some definite rule. Turing then showed that uncomputable numbers could be generated from the computable ones and thus there could be no algorithm for solving all mathematical questions. In this paper, Turing introduced a formal mathematical model of an abstract machine "A-machine" capable of simulating any algorithm. This model is known as Turing machine which is a standard model of computation. Alan Turing is widely known as father of modern computing and artificial intelligence. Since 1966, in his honor, Turing award is given annually by

ACM to an individual for technical contribution to computation community. Often known as "The noble prize of computing" Turing Award is recognized as the highest distinction in computer science.

4.6 Church-Turing Thesis: Equivalence between Church's effective function and Turing's computable function strengthened both their claims to validity and is known as the Church-Turing Thesis. There are many formulations of this theory, of which one of them is every effective computation can be carried out by a Turing machine.

4.7 Post correspondence problem: In 1936, Emil Post published a paper entitled Finite Combinatory Processes-Formulation I.[7] Post introduced a model of the universal machine based on the instructions that make the machine work. He also presented Post Correspondence Problem to obtain unsolvability results.

4.8 John Nash Conjecture: In 2012, NSA declassified letter from Nash to NSA written in 1955, in which he states while it is theoretically possible to decrypt a message without a key, but it would require exponential computation resources. His conjecture says for almost all secure cryptosystems, the computational complexity increases exponentially with the length of the key. It could possibly be the first reference about the complexity of an NP problem and P≠NP.

4.9 Gödel's lost letter: In 1989, a letter written in 1956 from Gödel to von Neumann was discovered. In this letter, he asked Neumann about the computational complexity of a combinatorial problem and inquired if it could have a linear or quadratic time solution on a Turing machine. It could possibly be the first question asked about the time complexity of an NP-complete problem on a deterministic machine.

.10 Non-deterministic Machines: In 1959, Michael O. Rabin and Scott Dana published a paper *Finite automata and their decision problem*[8] which introduced the idea of non-deterministic machines. In 1976, they were presented Turing Award for this paper.

4.11 P Class: In 1965, Jack Edmonds published *Maximum matching and a polyhedron with 0,1-vertices*[9] which defined the concept of polynomial time to differentiate between a practical and an impractical algorithm. In the same year, Alan Cobham published *The intrinsic computational difficulty of functions*[10] in which he mentioned about set of computationally feasible problems which are decidable in polynomial time, thus established the concept of the complexity class **P**. Cobham-Edmonds thesis states that problems can be feasibly computed on a computational machine only if they lie in complexity class **P**.

4.12 Computational Time Complexity: In 1965, Juris Hartmanis with Richard Stearns published a paper *On the Computational Complexity of Algorithms.*[11] In their paper, they defined complexity measure by computation time on Turing machines and developed a theory of complexity classes. Based on time taken by an algorithm they showed that there exist an infinite hierarchy of complexity classes (for example, problems with time complexity proportional to n, n log n, n^2, n^3 ,2^n, and so on).

4.13 Computational Space Complexity: In 1965, Lewis, Stearns, and Hartmanis published a paper *Hierarchies of memory limited computations*[12] establishing a similar hierarchy for space-bounded computation. In this paper, they defined sub-linear space-bounded complexity classes.

In 1993, J. Hartmanis and R. Stearns shared the prestigious Turing Award for establishing the foundations of computational complexity theory.

4.14 Speed-up Theorem: In 1967, Manuel Blum published a paper *A machine-independent theory of the complexity of recursive functions*[13]. In his paper, he presented an axiomatic theory of complexity and proved an important result, well known as speed-up theorem. In 1995, for his contributions to the foundation of computational complexity, Blum received the Turing Award.

4.15 Savitch's Theorem: In 1970, W. J Savitch in his paper *Relationships between nondeterministic and deterministic tape complexities*[14] presented Savitch's Theorem which gives the relationship between non-deterministic and deterministic spaces complexity as NSPACE($f(n)$) ⊆ DSPACE($f(n^2)$).

4.16 NP-Complete Class: In 1971, Stephen A. Cook published a historic paper *The complexity of theorem proving procedures.*[2] In this paper, he formalized the notions of **NP-completeness** and polynomial-time reduction based on Turing reducibility. He proved the existence of an NP-complete problem by showing that the Boolean satisfiability problem and subgraph isomorphism is NP-complete. In 1982, for his significant contribution to computational complexity, he received Turing Award and in 1999, CRM-Fields Institution Award.

In 1973, working independently Leonid Levin published a paper *Universal search problems.*[14]in which he proved the existence of practically relevant NP-complete problems. He considered 6 NP-complete search problems which required solution instead of determining existence. In 2012, for his discovery of NP-completeness and average case complexity, he was awarded Knuth Prize.

4.17 Cook-Levin Theorem: It states that Boolean satisfiability problem is NP-complete and any NP-complete problem can be reduced to Boolean satisfiability problem in polynomial time by a deterministic Turing machine. It is also known as Cook's theorem.

4.18 Karp Reductions: In 1972, following Cook's work Richard Karp published his paper *Reducibility among combinatorial problems.*[16] He selected a set of 21 combinatorial and graph theoretical computational problems and using Cook's theorem showed that there is a polynomial time deterministic many-one reductions from Boolean satisfiability problem to each of 21problems, thereby proving they are all NP-complete. Due to the standard framework provided by his paper, thousand of problems have been added to NP-complete class. For his contribution to the theory of NP-completeness, in 1985 he was awarded Turing Award. He also won the Lanchester Prize in 1977, the Fulkerson Prize in discrete mathematics in 1979, the von Neumann Theory Prize in 1990.

After the celebrated work of Cook and Karp popularized the P vs NP question, computer scientists, mathematicians and researchers immediately tried and have been trying hard to prove P=NP or P≠NP.

5. P vs NP STATUS: ATTEMPTS & LIMITATIONS

Over the period of more than four decades, many researchers have attempted to prove it. Many claims for the solution have been registered but all of them have been discredited so far. The ingenious and noteworthy efforts that have been put in by dedicated researchers to find a solution have not been wasted.

It has lead to the significant advancement of complexity theory. Many beautiful algorithms to cope up with NP-completeness have been designed, novel methods of computations have been devised, wonderful theorems, conjectures, and results have been proved, problems which were previously unsolvable have been solved, numerous awards have been begged by people for their distinguished research papers, for their contribution to the advancement of computation theory. And most importantly, the intricacy and challenge that this problem present has inspired the human endeavor towards the pursuit of relentless quest towards perfection.

So, the status of the problem is – Open and up for the challenge. However, the failed efforts have taught us the limitation of existing techniques and the need to overcome the barriers which hinder the possibility of the solution.

5.1 P = Np Approaches & Claims

5.1.1 Polynomial time algorithm: If someone creates a legible polynomial time algorithm for 3-SAT or any one of the known NP-complete problem then it would conclusively prove P=NP. As polynomial time algorithms are standard for efficiency in industry, the majority of programmers and mathematicians have attempted to find such algorithms for NP-complete problems for few decades without any success. So far, most of the P = NP claims in this category are either flawed or ambiguous. But the quest for finding an efficient algorithm has lead to improvement in algorithm design and analysis and novel techniques such as approximation algorithms, randomized algorithms, LP rounding, semidefinite programming, parametrized algorithms, and much more have been invented. In 2015, Gödel and Knuth prize winner Laszlo Babai[17] produced a breakthrough and found a quasi-polynomial time algorithm for graph isomorphism problem which opened up the possibility of proving NP=QP.

5.1.2 Breaking Cryptographic Code: There exists a strong motivation for the hackers, researchers, and programmers for breaking the cryptographic codes that assume P≠NP for their security. While the attempts to prove P=NP by designing algorithms to break codes hasn't worked so far. In response to it, breakthrough results for more secure systems in cryptography have been achieved.

5.1.3 Non-Constructive Proofs: It is possible that some non-constructive argument could show that a polynomial algorithm for NP problem exists without providing any feasible efficient algorithm. Many experts are of the opinion if, at all P=NP, the proof is more likely to be a non-constructive proof, or rendered to be practically inefficient due to the large size of the bounding polynomial. The non-constructive proof by Robertson and Seymour[18] is an important landmark in computational complexity.

5.2 P ≠ Np Approaches & Claims

5.2.1 Diagonalization with reduction: The classical diagonalization technique dates back to Cantor and was successfully used by Gödel to prove his Incompleteness Theorem and by Turing to prove undecidability results. In complexity theory, it has been successfully used to prove super-exponential lower bounds and for proving time and space hierarchies. Using diagonalization with reduction Rabin[19], Hartmanis and Stearns[11] proved the existence of decidable problems in higher complexity classes. The problem in proving P≠NP is simple diagonalization arguments are based on step by step simulation of machines and it is not known how an NP machine can simulate an arbitrary P machine. Moreover, a diagonalization proof is likely to relativize for lower bounds and therefore it fails to prove P≠NP.

5.2.2 Relativization: In relativized computation, the machine is provided with a black box called oracle which can answer a set of questions in fixed amount of time. Baker, Gill & Solovay[20] demonstrated an oracle A, relative to which $P^A=NP^A$ and another generic oracle P, relative to which P≠NP. Therefore, no relativizable proof can furnish P vs NP solution in either direction. Relativization is a feature shared by most complexity results which make it difficult to prove P≠NP. For some time, it was believed that relativization will prove the independence of P vs NP problem, but after the existence of proof systems that do not relativize all hopes were abandoned.

5.2.3 Interactive Proof systems: In 1985, Shafi GoldwAsser, Silvio Micali and Charles Rackoff[21] introduced an interactive proof system consisting of two machines, a prover, with infinite computational resources and a verifier, a probabilistic polynomial time machine with an access to true random bits. For this paper, they shared first ever Gödel prize in 1993. Through the interaction between two machines and putting certain bounds on verifier, this model provides vital implications for traditional complexity classes, for instance, class of NP is a model with a deterministic polynomial time machine. In 1990, Lund, Furtnow, Karloff and Nisan[22] introduced an indirect method of simulation using algebraic techniques for the construction of interactive proof systems which does not relativize. Using the technique Turing award recipient Shamir[23] proved IP=PSPACE, followed by the proof MIP=NEXP by Babai, Lund and Fortnow[24].

5.2.4 Algebrization: In 2008, Scott Aaoronson and Avi Wigderson[25], extended the relativization barrier and introduced an algebraic oracle which presented a new barrier known as algebrization. They have shown that in presence of such oracle, the interactive proof system results at polynomial time algebrize and thus cannot be used to resolve the lower bound of NP problem.

5.2.5 Zero Knowledge proof systems: Shafi GoldwAsser, Silvio Micali and Charles Rackoff[21] paper also defined a new proof system with randomized and interactive verification procedure. Surprisingly, zero knowledge proof systems have been used to prove abstract as well as concrete problems. Goldreich, Micali and Wigderson[26] proved if one assumes the existence of one-way functions, then every set in NP has a zero-knowledge proof. One way functions conjecture states the existence of functions which are easy to compute but hard to invert. If proven true it would imply FP ≠ FNP, which would imply P ≠ NP. One way functions are widely believed to exist and several candidate functions have been proposed and used in practical applications. Proving the existence of one-way functions is supposed to be a much harder problem than P ≠ NP. Also, Razborov and Rudich[27] proved that existence of one-way functions implies that there cannot be any natural proof for P vs NP. For the same, they received Gödel prize in 2007.

5.2.6 Probabilistically checkable proofs (PCPs): In 1992, Arora and Safra[28] defined probabilistically checkable proofs. It captivated the interest of researchers and successive papers with creative notions culminated in the development of PCPs and PCP theorem. PCPs are proofs which can be verified by a randomized algorithm using bounded random bits and a bounded number of query bits of the proof in such a way that the probability of correct verification is very high. Based on

the amount of randomness and number of queries allowed for verification, various complexity classes have been defined. As per PCP theorem, every problem in NP class has a PCP of the constant query and logarithmic randomness complexity i.e. PCP($O(logn)$,$O(1)$) = NP. An alternate formulation of PCP theorem, the hardness of approximation states that for many NP problems solution cannot be efficiently approximated unless P=NP. In 2001, Arora, Feige, Goldwasser, Lund, Lovasz, Safra, Motwani, Sudan and Szegedy shared Gödel prize for their contribution to PCPs and its applications to hardness of approximation.

5.2.7 Circuit Complexity: In circuit complexity, Boolean functions are classified according to size or depth of circuits of AND, OR and NOT gates. Circuit complexity was introduced by Shannon[29] in 1949 and he proved that for almost all Boolean functions of n variables require circuit with at least $2^n/n$ gates. If superpolynomial lower bound on the size of Boolean circuits, solving any NP-complete problem, can be proved, it would suffice P ≠ NP. Circuit complexity has been successful in proving exponential lower bounds in restricted models.

5.2.8 Bound Depth Circuits: In 1983, Ajatai[30] and independently by Furst, Saxe and Sipser[31] in 1984, showed small circuits having fixed number of gates cannot solve the parity function. In 1987, Håstad[32] through his switching lemma established that any constant depth circuit would require an exponential number of gates. In 1987, Razborov[33] proved the requirement of the exponential size of bound depth circuits over large basis. Building on his work, Smolesky[34] in 1987 proved requirement of exponential size for computing Mod_p function.

5.2.9 Monotone Circuits: In 1985, Razborov[35] proved superpolynomial bound for the k-clique problem using monotone circuits (circuits with AND and OR gates only). In 1987, his result was improved to exponential lower bound by Boppana and Alon[36]. If Razborov's result can be extended to general circuits, it would suffice P ≠ NP.

5.2.10 Natural Proofs: In 1993, Razborov and Rudich[27] studied all important lower bound results in restricted class of circuits and concluded that all the lower bound results were natural. They defined the class of strategies used in lower bound proofs as Natural proofs and showed if one way functions conjecture holds, then no natural proof can distinguish between P and NP classes.

The techniques to prove lower bound results in circuit complexity are either natural proofs or they algebrize. However, in 2010, Williams[37] succeeded in avoiding both the barriers by using non-constructive technique using the special property of ACC^0 circuits and proved NEXP ⊄ ACC^0.

5.2.11 Descriptive Complexity: In descriptive complexity, the complexity classes are characterized by the type of logic needed to express the class of languages representing structures in them. In 1974, Fagin[38] showed that NP equals the class of languages that can be expressed by second order sentences. Later, Immerman[39] extended it to characterize many classes including P, NL. In 1988, Immerman[40] and Szelepcsenyi[41] independently proved NL = coNL which gave descriptive complexity a huge boost. If over finite, ordered structure FO(LFP) = SO then it is equivalent to P = NP.

5.2.12 Proof Complexity: It is a measure of the efficiency of proof systems based on lengths of proofs lower bounds. In 1979, Cook and Reckhow[42] presented propositional proof systems for proving classical propositional tautologies. If there exists a polynomically bound proof system for all tautologies then it is equivalent to NP = coNP else if there are no short proofs for tautologies, it would imply P ≠ NP. In 1985, Haken[43] solved the pigeon hole problem and showed there are unsatisfiability formulas having exponential size resolution proofs. Since then, there have been numerous exponential lower bounds on unsatisfiability proofs.

5.2.13 Randomness in Proofs: Introduction of the degree of randomness, as a part of their logic, into proofs has lead to startling results and a new characterization of traditional complexity classes. A pseudorandom generator provides random bits which speed up the recognition of some languages by randomized algorithms significantly if one permits exponentially small probability of error. The class BPP (bounded-error probabilistic polynomial time) consists of all decision problems that can be recognized by a probabilistic computational machine in polynomial time bounded by small error probability on every input. If randomness is removed from BPP then it is equivalent to P, so P ⊆ BPP. In 1999, Impagliazzo and Wigderson[44] showed that if a sufficiently difficult problem such as 3-SAT requires exponential size circuit to solve then P = BPP else P ≠ NP. If pseudorandom generator with a very short seed exists then it would imply that any problem with efficient randomized algorithm also has an efficient deterministic algorithm. However, finding it is supposed to be a harder problem than P vs NP.

5.2.14 P vs NP Independence[49]: Repeated failures to resolve P vs NP lead to one of the obvious speculation that P vs NP may be unsolvable like Continuum Hypothesis and that it is independent of some standard axiomatic system like ZFC and there is no way to decide if P = NP or P ≠NP. The oracle relativization barrier introduced by Baker, Gill and Solovay[20] in a way suggests a weak independence for P vs NP question. In view of relativization, attempts were made to formulate an axiomatic system on the lines of pure recursive function theory, which could show that P vs NP is undecidable but no one managed to furnish strong independence result in strong axiomatic condition. In 1976, Hartmanis and Hopcroft[45] showed P vs NP is independent of formalized set theory by oracle relativition. In the context of natural proof barrier in circuit complexity, Razborov[46], under cryptographic assumptions, showed unprovability of lower bound on circuits in bounded arithematic. Within weak frameworks of logical theories, non-oracle weak independence results have been established by DeMillo and Lipton[47], and Sazanov[48].

6. TRIUMPH OF AN INCESSANT QUEST

P vs NP, with its beauty and complexity, truly encompass the relentless spirit of human endeavors to strive for perfection and an innate desire to unlock the intimate intricacies of nature. A long tradition of failed attempts and seemingly increasing complexity of the problem, with proof nowhere in sight might make us wonder what did one gains out of a quest for P vs NP? Well, to be very precise, the gains from the quest have been numerous and profound with significant practical and theoretical implications.

P vs NP is one of the fundamental questions which is extensively studied not only in mathematics and computer science but referenced in other domain of knowledge and learning.

On P vs NP hundreds of quality research papers are being published each year that has lead to the advancement of not only complexity and theory of computation but many other

fields notably modern cryptography, algorithm analysis, mathematical models and proof systems, Quantum computing etc.

Numerous awards and prizes of highest distinction have been received by P vs NP researchers. Rabin, Cook, Karp, Hopcraft, Hartmanis, Stearns, Yao, Valiant, Shafi, Micali, Hellman and Diffie have won prestigious Turing Award. Almost all distinguished papers on P vs NP have received Gödel prize. Babai, Hastad, Shafi, Micali, Moran, Immerman, Szelepcsenyi, Shor, Arora, Lund, Safra Razborov, Rudich, Wigderson list of Gödel prize winners is extremely rich and diverse. The list of prizes and awards that went the P vs NP way is endless.

Owing to brilliant work of sharpest minds, important problems have been moved into P class most notably linear programming by Khachiyan & Karmakar, disjoint paths by Robertson & Seymour, primality by Agrawal, Saxena and Kayal etc.

The failure to find a proof has been a blessing for cryptography. It has lead to extensive research and many practical applications have been developed.

The breakthrough research and nontrivial insights have revolutionized the ways to cope up with NP-complete problems. Practically efficient algorithms have been developed and new techniques have been devised. Randomized algorithms, probabilistic algorithms, quantum algorithms, dynamic and semi-definite programming, approximation algorithms etc have been of huge practical relevance and significance.

Researchers have been able to calculate lower bounds on restricted systems, proven weak independence, quasi-polynomial algorithms have been found, barriers and limitations have been identified. All this and much more hints that attempts to prove P vs NP has lead to non-trivial insights which have enhanced our understanding of the problem and methods.

Last but not the least it has attracted the attention and captivated the interest of brilliant minds in Mathematics and Computer science. It has inspired geniuses to go out of the box and achieve miraculous feats.

The holy grail of mathematics and computer task has bequeathed eternal treasures for mankind to cherish and prosper. It is not a saga of doomed failures but the triumph of an epic quest.

7. P vs NP SOLUTION & FUTURE SCOPE

Though it is generally believed $P \neq NP$ but in the absence of a sound proof, the possibility of $P = NP$ still remains open. At this moment, the only thing that scholars are sure about is barriers of relativization, algebrization and natural proofs and any solution technique must go beyond them. Also, the proof must encompass known lower bound results, and prove existing weak results. Let us examine few directions[56] which appear to be more promising approaches towards getting close to the solution of P vs NP problem.

7.1 New Techniques and Hard Math: Many experts are of the opinion that all the current known techniques are not powerful enough to solve P vs NP and it would require major mathematical advances, based on entirely new fundamentals, to find a solution.

7.2 Geometric Complexity Theory: Ketan Mulmuley and Milind Sohoni[50] have proposed an approach to P vs NP problem through algebraic geometry, also known as Geometric complexity theory or GCT. At the present moment, GCT is the only approach which seems viable. Using GCT, two concrete lower bounds, and some important theorems like flip theorem and decomposition theorems have been proved. But this theory is so naïve that it will take deep mathematics that could take decades to progress in GCT.

7.3 Polynomial time algorithms: Over the years, the algorithms design and analysis have been revolutionized by P vs NP problems. Various important problems have been moved into P, numerous new methods and techniques have been devised and deployed. Quasi-polynomial time algorithm for graph isomorphism has been a revelation in itself. So, it is very much possible that someday some genius comes up with a polynomial time or even quasi-polynomial time algorithm for the NP-complete problem.

7.4 Refinement of existing techniques: The proof of equivalence of complexity class of IP and PSPACE has shown that through indirect simulation, it is possible to avoid relativization barrier. Ryan William successfully avoided the barriers of relativization and algebrization by using non-constructive technique and special property of ACC^0 circuits, proved $NEXP \not\subseteq ACC^0$. So it is entirely possible that a clever intuitive diagonalization simulation will go beyond oracle relativization or some special property of complexity classes will provide non-trivial insights into the problem.

7.5 Combination of Existing Techniques: Another way of finding P vs NP solution is by integrating the various existing techniques and subsuming the known weak results, lower bounds, important theorems, and disparate facts. Using some or more of elements of algebraic geometry, combinatorics, graph theory, operation research, model theory, restricted systems, algebraic field functions, logic theory etc can be combined to prove general results.

7.6 Game theoretic logical models: Using game theoretic models of computation researchers have been able to prove some important results and find major barriers in existing techniques. Traditionally, winning strategy in games has always provided the necessary impetus to go beyond the challenges. Moreover, game theory is one of the best contenders that incite originality and creativity.

7.7 Quantum Computing: It is a theoretical model of computation based on principles of quantum mechanics such as superposition and entanglement. Quantum model of computation can be used to create efficient algorithms for NP-complete problems. Peter Shor[51] gave a quantum algorithm for factorization and discrete logarithm solution. Lov Grover[52] achieved a quadratic speed up on general NP-complete problems using the quantum algorithm. But quantum computing research is still in its infancy and has to go a long way. However, it suggests that by creating a different model of computation one can resolve P vs NP by defining new complexity classes.

Whether P = NP or otherwise or it is independent, it is evident that its solution would require novel idea, a breakthrough possibility, and a creative leap whether it is in terms of new mathematics, or new algorithm, or refinement, or integration, or new models of computations and machine learning etc.

8. CONCLUSION

Through the study of the P vs NP problem one can say that P vs NP has become certainly the most fundamental and important mathematical question of our time whose relevance has only grown with time. In this paper, authors have discussed P vs NP problem, its significance, historical overview, and attempts to prove P=NP or P≠NP and the approaches that have been used to deal with the hardness of NP-complete problems. Also, authors have looked at the direction of active & future research which may give in a way to tackle P vs NP problem that nature throws at us.

Most of the work mentioned in the paper stemmed out of long series of intricate mathematics research papers providing breakthroughs and significant insights into the problem. After decades of study and research, the majority of the scientific community believes P≠NP but in absence of a legible proof its validity is still questionable and it is quite possible that one day someone might figure a way to prove P=NP even though the proof may by non-constructive. Attempts have been made to prove P≠NP but with each attempt, the complexity of the problem seem to increase. Anyways, proving P≠NP has remained perplexingly difficult.

The problem as it stands today continues to inspire the brightest minds and incessant research will lead us into yet even new complexities and new opportunities. Though most of the experts believe that chance of P vs NP solution happening in foreseeable future is distantly remote but a spark of genius and a flash of creativity can defy all the known convictions and beliefs as it always has in the past.

9. REFERENCES

[1] A. M. Turing, On computable numbers with an application to the entscheidungs problem, Proceedings of London Mathematical Society ser. 2, 1936.

[2] S. A. Cook , The complexity of theorem proving procedures, Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, 1971.

[3] Official problem description The P versus NP by Stephen Cook, www.claymath.org/millennium-problems/p-vs-np-problem, 2000.

[4] Kurt Gödel, Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I, Volume of Monatshefte für Mathematik, 1931.

[5] A. Tarski, tr J.H. Woodger, The Concept of Truth in Formalized Languages , Logic, Semantics, Metamathematics, Hackett, 1983

[6] Alonzo Church, An Unsolvable Problem of Elementary Number Theory, American Journal of Mathematics, Vol. 58, No. 2, 1936.

[7] Emil L. Post , Finite Combinatory Processes-Formulation I, The Journal of Symbolic Logic, Vol. 1, No. 3, 1936

[8] M. O. Rabin and D. Scott, Finite automata and their decision problem, IBM J. RES. 3, 1959.

[9] Jack Edmonds, Maximum matching and a polyhedron with 0,1-vertices, Journal of Research of the National Bureau of Standards Section B 69, 1965

[10] A. Cobham, The intrinsic computational complexity of functions, proc. Int. Congress on logic, methodology and philosophy of science, Amsterdam, 1965.

[11] J. Hartmanis and R. E. Stearns, On the Computational Complexity of Algorithms, trans. American Mathematics Society 117, 1965.

[12] J. Hartmanis, P. M. Lewis, and R. E. Stearns, Hierarchies of memory limited computations, Proc. 6th Annual IEEE Symp. on Switching Circuit Theory and Logical Design, 1965.

[13] M. Blum, A machine-independent theory of the complexity of recursive functions, Journal of ACM 14:2, 1967.

[14] W. J. Savitch, Relationships between nondeterministic and deterministic tape complexities, Journal of Computer and System Science 4:2, 1970

[15] L. Levin, Universal search problems,Problemy Peredachi Informatsii, 9(3), 1973

[16] R. M. KARP, Reducibility among combinatorial problems, Complexity of Computer Computations, Plenum Press, NY, 1972

[17] Laszlo Babai, Graph Isomorphism in Quasipolynomial Time I: The "Local Certificates Algorithm", Combinatorics and Theoretical Computer Science seminar, 2015

[18] N. Robertson and P. Seymour, Graph Minors. XX. Wagner's conjecture, Journal of Combinatorial Theory, Series B, 92 (2), 2004

[19] M. Rabin, Degree of difficulty of computing a function and a partial ordering of recursive set, Tech. rep. no. 2, Hebrew university, 1960

[20] T. Baker, J. Gill and R. Solovay, Relativization of the P≠NP question, SIAM Journal on Computing 4:4, 1975.

[21] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The Knowledge complexity of interactive proof-systems. Proceedings of 17th ACM Symposium on the Theory of Computation, Providence, Rhode Island. 1985.

[22] C. Lund , L. Fortnow , H. Karloff , and N. Nisan, Algebraic Methods for Interactive Proof Systems, proc. of the 22th Annual ACM symp. On theory of computing, 2-10, 1990.

[23] Adi Shamir, IP = PSPACE, Journal of the ACM, volume 39, issue 4, 1992

[24] L. Babai, L. Fortnow, C. Lund, Non-deterministic exponential time has two-prover interactive protocols, Proc. 31st Ann. IEEE Symp. Found. Comp. Sci., 1990.

[25] S. Aaronson and A. Wigderson, Algebrization: A new barrier in complexity theory, proceedings of the 40th Annual ACM Symposium on Theory of Computing, 2008

[26] O. Goldreich, M. Silvio, and A. Wigderson, Proofs that yield nothing but their validity, Journal of the ACM. 38 (3), 1991.

[27] A. A. Razborov and S. Rudich, Natural proofs, Journal of Computer and System Sciences. 55, 1997

[28] S Arora, and S. Safra, Probabilistic checking of proofs: A new characterization of NP. Journal of the ACM, 45(1), 1998

[29] C. Shannon, The synthesis of two-terminal switching circuits, Bell System Technical Journal. **28** (1), 1949.

[30] M. Ajtai, $\sum_1^1$. formulae on finite structures, Annals of Pure and Applied Logic. **24**: 1983.

[31] M. Furst, J. Saxe, and M. Sipser, Parity, circuits, and the polynomial-time hierarchy, Mathematical Systems Theory 17(1), 1984.

[32] J. Håstad, Computational limitations of small depth circuits, Ph.D. thesis, MIT press, 1987.

[33] AA. Razborov, Lower bounds on the size of bounded depth networks over a complete basis with logical addition, trans. in Math. notes of the Academy of Sciences of the USSR 41, 1987.

[34] R. Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity, in Proc. 19th ACM Symposium on Theory of Computing, 1987.

[35] A. Razborov, Lower bounds on the monotone complexity of some Boolean functions, Mathematics of the USSR, Doklady, 1985

[36] N. Alon, R. Boppana, The monotone circuit complexity of Boolean functions, Combinatorica 7 (1), 1987

[37] R. Williams, Non-Uniform ACC Circuit Lower Bounds, CCC 2011: Proceedings of the 26th Annual IEEE Conference on Computational Complexity, 2011.

[38] R. Fagin, Generalized First-Order Spectra and Polynomial-Time Recognizable Sets, Complexity of Computation, ed. R. Karp, SIAM-AMS Proceedings 7, 1974

[39] N. Immerman, Languages that Capture Complexity Classes, SIAM J. Comput., 16(4), 1987.

[40] N. Immerman, Nondeterministic space is closed under complement, SIAM Journal on Computing,1988.

[41] R. Szelepcsenyi, The method of forced enumeration for nondeterministic automata, Acts Informatica 26, 1988.

[42] S. Cook, and R. Reckhow, The Relative Efficiency of Propositional Proof Systems, J. Symbolic Logic 44, 1979.

[43] A. HAKEN, The intractability of resolution, Theoretical Computer Science 39,1985.

[44] R. Impagliazzo, and A. Wigderson, P = BPP requires exponential circuits: derandomizing the XOR lemma, STOC '97 (El Paso, TX, ACM, New York), 1999.

[45] J. Hartmanis, and J. Hopcroft, Independence results in computer science, ACM SIGACT News 8, no. 4,1976.

[46] A. Razborov. Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic, Izvestiya Math. 59(1), 1995.

[47] R. DeMillo and R. Lipton, The consistency of P=NP and related problems within fragments ofnumber theory, in Proceedings of ACM STOC'79, 1979.

[48] V. Sazanov, A logical approach to the problem "P=NP?", in Mathematical Foundations of ComputerScience, Springer LNCS 88, 1980.

[49] S. Aaronson, Is P vs NP formally Independent?, bulletin of the Europian Association for theoretical computer science,81 , 2003.

[50] K. D. Mulmuley and M. Sohoni, Geometric Complexity Theory I: An Approach to the P vs. NP and Related Problems, SIAM J. Comput. 31(2), 2001.

[51] P. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM Journal on Computing,26(5),1997.

[52] L. Grover, A fast quantum mechanical algorithm for database search, In Proceedings of the 28th ACM Symposium on the Theory of Computing ACM New York, 1996.

[53] Fortnow L., Review Article - The status of P vs NP problem, Communications of the ACM, 52(9), 2009.

[54] Sipser M., The history and status of P vs NP question, 24th annual ACM symp. On theory of computing, 1992.

[55] Fortnow L., and S. Homer, A Short History of Computational Complexity, Bulletin of the European Association for Theoretical Computer Science, 80, 2003.

[56] Hemaspaandra L., SIGACT News Complexity Theory Column 74, Dept. of Computer Science, University of Rochester, 2012.