IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 November 2023, accepted 13 January 2024, date of publication 18 January 2024, date of current version 29 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3355540

TOPICAL REVIEW

Computer-Aided Verification of P/NP Proofs: A Survey and Discussion

STEFAN RASS[1,2], (Member, IEEE), MAX-JULIAN JAKOBITSCH[2], STEFAN HAAN[2], AND MORITZ HIEBLER[2]

[1] LIT Secure and Correct Systems Laboratory, Johannes Kepler University Linz, 4040 Linz, Austria
[2] Institute for Artificial Intelligence and Cybersecurity, Universität Klagenfurt, 9020 Klagenfurt am Wörthersee, Austria

Corresponding author: Stefan Rass (stefan.rass@jku.at)

This work was supported by the Austrian Science Fund (FWF) under Grant TAI 201-N.

ABSTRACT We survey a collection of proofs towards equality, inequality, or independence of the relation of P to NP. Since the problem has attracted much attention from experts, amateurs, and in-betweens, this work is intended as a pointer into directions to enable a "self-assessment" of ideas laid out by people interested in the problem. To this end, we identify the most popular approaches to proving equality, inequality, or independence. Since the latter category appears to be without any attempts to follow the necessary proof strategies, we devote a section to an intuitive outline of how independence proofs would work. In the other cases of proving equality or inequality, known barriers like (affine) relativization, algebrization, and others are to be avoided. The most important and powerful technique available in this regard is a formalization of arguments in automated proof assistants. This allows an objective self-check of a proof before presenting it to the scientific community.

INDEX TERMS P/NP question, proofs, barriers, relativization, proof assistants.

I. INTRODUCTION

Among the most famous open questions of computer science is whether the complexity classes P and NP are equal or not, or if this relation is provable at all (say, within Zermelo-Fraenkel set theory). Given a decision problem, whose size is measured by an integer n, we may qualitatively (and informally) think of P as the set of all decision problems that are solvable in a number of steps that is at most some polynomial in n, whereas NP is different in asking (only) for the verifiability of a given answer in a polynomial amount of steps (depending on n), based on a certificate string whose size is as well at most polynomial in n. More formally, let Σ be a finite alphabet, and let $L \subseteq \Sigma^*$ be a formal language. For a given word $w \in \Sigma^*$, let the problem be the decision about whether or not $w \in L$ holds. In that context, a language L is in the complexity class P if and only if there is an algorithm A that outputs $A(w) = 1 \iff w \in L$ (and zero otherwise), after at most $p(|w|)$ many steps, where p is some polynomial (that depends only on L) and $|w|$ is the number of symbols in w. It is important to note that A takes only w and no other input. The class NP is characterized by allowing algorithm A to take a limited amount of auxiliary information for the decision. Formally, let q be another polynomial, and let the algorithm take two strings w, x to output $A(w, x) = 1 \iff w \in L$. Herein, x may explicitly depend on w, and must not have more than a length of $|x| \leq q(|w|)$, where q is a polynomial. Other than that, the same constraint on the running time of A holds, i.e. $A(w, x)$ must terminate with 0 or 1 after at most $p(|w|)$ many steps,[1] where the polynomial p again depends on L (only).

Despite its conceptual simplicity, the problem of whether the inclusion P $\subseteq$ NP is strict or not has yet escaped all attempts to answer whether the two classes are equal or not. Many domain experts believe that the two classes are distinct [1], but there are also a considerable lot of papers that claim equality of the two classes.

The associate editor coordinating the review of this manuscript and approving it for publication was Byung-Gyu Kim.

[1] The dependence of the auxiliary string on w is what distinguishes non-determinism from probabilism: if we let A run with an auxiliary input string x that is allowed to be (stochasticall) independent of w, then A is a said to work probabilistic. These are out of our scope here.

A. PAPER SEARCH AND SELECTION METHODOLOGY

This work presents some summary statistics about the developments of proofs around P-vs-NP towards equality, inequality, or unprovability of a solution, which starts from a collection by Woeginger [2], and the survey by Aaronson [3], and extends this list by papers appeared after these references. We herein do not provide a full account of each solution but rather seek to overview the entirety of attempts that have been made. As Gasarch has eloquently put it [1]: "The practical impact would come not from the result itself, but from the new ideas needed to achieve it.".

In turn, and despite all heuristic arguments to not become overly excited about yet-another-proof-of-P-vs-NP, the problem has attracted many people with many potentially interesting ideas, but the number of proofs coming up simply exceeds the community's capacities of verifying them.

Starting from the aforementioned surveys, we extended the list by querying the following digital libraries:

- IEEE Xplore
- ACM Digital Library
- ScienceDirect
- Web of Science
- Google Scholar
- Citeseer

For the following keywords "P-vs-NP", "Cook's conjecture" (with and without the apostrophe), and "P/NP question", "P equal to NP", "P not equal to NP" and some slight variations thereof, all using full phrase search (not letting words separated by a space being treated as independent search terms), we used Boolean connectives or conditions between these keywords and "proof" as the only second keyword (using the keyword "answer" delivered too many results unrelated to our goal, to be considered as useful). We intentionally did not specify additional keywords like "equal", "not equal", "unprovable" or "independence", since we were interested in working on all these cases, so no restriction based on such keywords appeared necessary. Given the relevance of the topic, the resulting number of papers referring to the issue was then narrowed down by excluding papers that matched one or more of the following criteria:

- Reference to the question as an unproven hypothesis, but not attempting to answer the question itself (for example, speaking about intractability or the hypothesis that some problem is not solvable in polynomial time unless P is equal to NP), except for other surveys about the same topic.
- References that used P and/or NP separately in their arguments, but not speak about the relation between these two explicitly, except for known conditions (such as the inclusion of P in NP).
- References that we already knew from the previous lists (see above).

To screen the papers of the existing lists (such as Woeginger's [2]), we queried normal i.e., not scientifically specialized, search engines to locate papers that were published on personal websites, blogs, and others. Works that we could not locate in this way were searched using www.archive.org and with the help of weblinks that were provided in Woeginger's list. This lets us retrieve almost all (except for a few) papers that disappeared from the web as of today.

As of the time of writing this article, we have collected a total of 126 papers dealing with how P relates to NP, which can roughly be divided into three classes[2]:

- Proofs that $\mathrm{P} = \mathrm{NP}$, proposed in a total of 67 papers,
- Proofs that $\mathrm{P} \neq \mathrm{NP}$, proposed in a total of 55 papers,
- and proofs that the problem itself is unsolvable, as only 4 papers claim.

This is in considerable contrast to the (third) poll made by Gasarch [1], which found $\approx$ 88% of people believe $\mathrm{P} \neq \mathrm{NP}$, (only) 12% believe $\mathrm{P} = \mathrm{NP}$, but nobody believed in independence (presumably from Zermelo-Fraenkel set theory with the axiom of choice (ZFC)) or void of an opinion (different to earlier polls about the same question, also conducted by W. Gasarch[3]).

Taking these numbers as statistics about the community's belief about the answer to the question, however, could be misleading, since many proof proposals have been contributed, but some lack the required rigor in their definitions and reasoning.

Nonetheless, the problem remains important and outstanding, however, due to its apparent simplicity may be in danger of going unresolved forever if we think about it as a "queuing problem": if $\lambda > 0$ papers about P-vs-NP come up per time unit (undefined, but different choices may only scale λ accordingly), and $0 < \mu < \lambda$ undergo reviews to identify flaws, then the number of unverified proofs will grow towards infinity. Hence, even if the solution *is* found someday, there would be a decent chance for it to vanish in the vast amount of competitor work on the topic. To get some numbers, let us decompose the above figures into more details, to get the average papers count to appear, versus the average number of papers to be verified. Figure 1 displays the counts, excluding (sometimes frequent) updates or revisions of papers (and excluding work that escaped our eyes[4]).

We believe that all proof attempts deserve scientific attention and review in the first place, and not be rejected because appearing to be from an "overly ambitious amateur". History has lots of examples of groundbreaking accomplishments made by people from all professions or institutions and at all levels of education. However, time for peer review is usually scarce, and resources need to be primarily dedicated to the most promising proof proposals among the many. To this end, we believe that putting proofs to automated

[2] It should be noted that not all of these references are explicitly after a solution to P-vs-NP, but in many cases, discuss other problems whose solutions subsequently imply an answer to P-vs-NP.

[3] Whose guest column [1] is a wonderful case of an article that is interesting *and* fun to read at the same time.

[4] For which we apologize to all here unnamed authors having made their own contributions that would have deserved a mentioning.

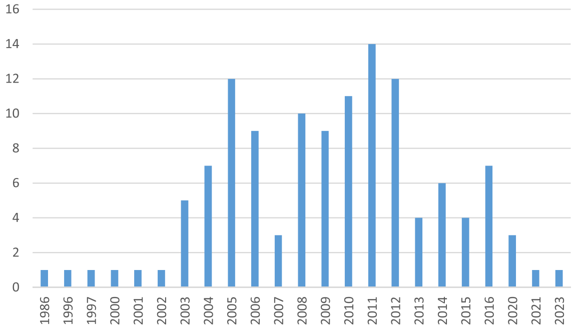

FIGURE 1. **Average number of papers published on P-vs-NP over time.**

proof assistants can be a first step of "self-assessment" to accomplish a baseline level of scientific rigor before approaching the community for a peer review. This may save time for professional peer reviews and avoids flaws in the construction of proofs at an early stage. The ambition to ultimately answer P-vs-NP must not be under-appreciated, making the empowerment of the community a desirable goal (motivating this work).

II. TOO MANY PROOFS MAY LEAVE A PROBLEM UNSOLVED TOO

The number of papers having received peer review (or at least some form of scrutiny) is considerably smaller, but it is worth mentioning that some papers indeed got to publication following scientific processes. Over the range of years from 1986 until 2023, only 28 papers seem to have received (documented) consideration by the scientific community. Taking an average of the counts in Figure 1, we find $\lambda >$ 5 papers to appear per year on average. In the 23 years that the collection covers (which excludes the exceptionally quiet period from 1987 until 1995), about 28 papers have received some sort of attention from the scientific community; Table 1 overviews these with the status as far as we could determine it. This means that $\mu \approx$ 28 papers/23 years $\approx$ 1.22, i.e., less than 2 papers are being reviewed per year. If we view the incoming new proofs as filling a queue that requires peer reviews to handle each paper, the capacities of the scientific peer review system will necessarily lag behind. Conversely, and based on well-known facts from queuing theory, the expected number of papers queued for verification will long-run diverge towards infinity.

It might be for this (among other) reasons that some venues like the ACM Journal of Computing have adopted a designated "P/NP policy" [44], which, once an author submits a proof claim for a review, explicitly forbids any further submission speaking about P-vs-NP by the same author for two years (simply to avoid a "paper overflow"). In light of the above numbers, this appears reasonable and also helps to stay away from known routes to failure. Specifically, some "meta-results" are known to be barriers to avoid for a decent attempt, such as relativization, algebrization, natural proofs, and others that we will discuss below. These allow us to cross out many candidate proofs from the list in a review, but still, a vast amount of work remains unverified as of today, leaving the possibility that a correct argument and answer may already have been found.

For the particular P-vs-NP problem, a considerable body of research deals with the identification of "dead-ends" by classifying certain arguments or proof techniques as incapable of settling the question in the first place. These so-called barriers are primarily used as heuristics for a quick judgment about whether or not some new proof attempt deserves a deeper inspection. In light of a critical discussion about the use of such barriers that we let follow in Section IV-A, we made our screening of proofs agnostic of these heuristic conditions and instead focused on the possibility of formalizing proofs in automated proof assistants like Isabelle/HOL or Coq.

We did our screening without alluding to the known barriers against proving P-vs-NP, receiving more attention in Section IV-A since we were interested in the exploration of the idea of using *proof assistants* to help with independent verification of the (too many) proofs around P-vs-NP.

A. THE POSSIBILITY TO "OBJECTIVELY SELF-REVIEW" ONES PAPER

Based on the collection of proofs, we believe that the scientific community would simply be overwhelmed by the sheer flood of papers coming out, why not empower the ones interested in the problem with running their own objective and independent reviews?

Clearly, a human reader, if it were the author itself, is biased, but formal proof assistants like Isabelle/HOL [45] or Coq [46] may help out here.

While it is prestigious to present a mathematical proof to the community, the equally important task of independent verification, today almost in all cases done by a peer-review, is far less "attractive" and offers only little incentive to domain experts to invest lots of time here without any revenue for it. Somewhat ironically, the P-vs-NP question is again special in this regard, since the aforementioned intuition behind NP is it capturing all problems to which a given solution is efficiently verifiable. So, the question is whether a proof about P and NP can itself be verified in reasonable (e.g., polynomial) time by humans or a machine. A machine-verification has the appeal of being objective by construction.

Of course, objectivity only holds to the extent of the human accurately mapping human-made proof into a machine-readable form that allows an automated verification. However, with the goal being a relief for domain experts from the burden to review P-vs-NP proofs, the author of such a proof has a natural interest in an accurate representation of the proof to a machine, who can then subsequently do an independent verification. This idea of assigning the verification back to the author, but obliging the person with the provisioning of a machine-verifiable proof has been

TABLE 1. Papers having undergone some (peer) review.

Ref.	Claim	Scientific Attention	Status
[4]	equal	commented [5] and refuted [6]	flaws identified
[7]	equal	open letter to the scientific community [8]	counterexamples given
[9]	equal	published, but indirectly refuted by [10],	implied flaws
[11], [12]	equal	studied and refuted by other scientists [13]	flaws identified
[14]	equal	studied in [15]	refuted
[16]	equal	patent applied	patent granted
[17]	equal	replied in [18] and rebuttal in [19]	rebuttal not commented so far
[20]	equal	published without official review [21]	not officially verified
[22]	equal	refuted by [6]	refuted
[23]	equal	appeared in the Southwest Journal of Pure and Applied Mathematics (discontinued)	reviewed
[24]	equal	appeared	reviewed
[25]	equal	published	reviewed
[26]	equal	Published as a book by WorldScientific	reviewed
[27]	equal	published	reviewed
[28]	equal	published	reviewed
[29]	equal	refuted by [30]	refuted
[31]	not equal	underwent a peer review, but was felt not convincing	published, but has negative reviews
[22]	not equal	refuted by [6]	refuted
[32]	not equal	published in the Journal ISRN Computational Mathematics 2012 ID: 321372, 1-15	not found on the web, only on arxiv
[33]	not equal	refuted by [34]	flaws identified
[35]	not equal	refuted by [36]	flaws identified
[37]	not equal	retracted (by the author)	flaws identified
[38]	not equal	discussed intensively online (by experts)	issues identified but left unresolved
[39]	not equal	published	removed from the internet
[40]	not equal	published	reviewed
[41]	not equal	published	reviewed
[42]	not equal	published	reviewed
[43]	not equal	published	reviewed

investigated along a research project about which this paper in parts will report.

B. FORMALIZING PROOFS

"Proof assistants", or interactive theorem provers hereafter, refer to software that aids in the construction of mathematical proofs. Unlike automated theorem provers, which try to prove theorems without further instruction from humans, proof assistants tell the user, acting like a programmer applying proof techniques or tactics, which claims or statements are left open to prove as sub-goals, until the list of open goals becomes empty, which finishes the overall proof. These interactive assistants may also opportunistically use automated provers to propose proof strategies or prove small sub-goals directly.

Most modern proof assistants base their foundation on some variant of typed lambda calculus rather than ZFC, as they seem to provide a much more suitable environment for automatic solvers. There exist projects that formalize mathematics in ZFC [47] but this is the exception, rather than the rule.

Our choice of proof assistant for this project has been Isabelle/HOL [45], as it aims to provide a logic and language similar to that typically seen in publications, in addition to powerful tooling. We believe it to be more pleasant and intuitive to work with than systems like Coq [46] and Lean, which provide powerful dependent type theories and in some cases enable much more concise and elegant definitions, but lag behind in readability and automation.

C. FORMALIZATION FOR AN "OBJECTIVE SELF-REVIEW"

The motivation to look into proof assistants for verification of arguments has various appealing aspects, such as:

- It is not possible to omit implicit assumptions, since the proof assistant will throw errors if an attempt is made to use an assumption that was not stated in the initial hypotheses or some sub-goal is left unproven (for instance, consider an induction that is not properly started, or an inequality that may be intuitive but still not proven rigorously). And though there exist debugging commands that allow developers to skip the proof of a statement, proof documents containing them are not considered valid, and it is easy to determine if any such command was used.[5]
- Especially for complexity theoretic arguments, it is easy to overlook matters like the explicit construction of the Turing machines (TMs) involved, if a proof is just done on paper; the proof assistant's requirement to present all

[5] An example for this would be the Isabelle command `sorry` which only works in interactive use, while throwing an error if the given document is checked properly.

arguments fully formally, while leading to significant increases in development time, naturally avoids such pitfalls.
- The computer has no personal interests or bias towards or against certain affiliations or backgrounds and cannot be convinced by authority, reputation, or other subjective (human) factors. This enables anyone (who is willing to learn to work with a proof assistant) to get fast, direct, and objective feedback.

The last two points imply a degree of objectiveness when somebody formalizes one's own proof about P-vs-NP since even though the person may have a strong personal wish for the work to be correct, the proof assistant will mercilessly point out any error. Even if the author then attempts to manipulate the proof's code towards making the proof assistant accept, it will end up in either (i) adding tweaks like skipped (sub-)proofs when compared to the paper version, which clearly marks weak spots in the line of argument, or (ii) changing the proof entirely, such that it no longer corresponds to the paper version. The second case is not necessarily problematic, since as long as the deviating proof is correct, the intended result has still been proven. The consistent translation problem then remains, only in the converse direction, since we then need to convert the machine-readable and -checkable proof back into a text version that a human reader can understand and verify.

The consistency issue, however, is vital in terms of definitions, axioms, and basic assumptions. Definitions of concepts on the paper may – even slightly – differ from the way they are formalized, i.e., "programmed", in the proof assistant. Hence, the equivalence of concepts on paper and their counterparts (of the same name) in the proof document needs to be verified manually by humans. Otherwise, the truth asserted by the proof assistant may have little to say about the correctness of the corresponding proof on paper.

At least for the P-vs-NP question, the problem of how accurately a textbook proof is mapped into its formalized version is crucial in terms of how the underlying definitions are implemented. Provided that the theoretical and implemented concepts (definitions, axioms, etc.) are verifiably consistent, the match between the argument flow on paper and in the proof assistant becomes secondary, since if the formalized proof is correct, we *do* have a correct proof.

III. OVERVIEW OF PROOF ATTEMPTS

Many arguments for equality are based on seemingly polynomial-time algorithms for some NP-complete problem, whose existence is known to imply equality of the two classes. Arguments for inequality have various roots, some identifying certain properties that all problems in NP must have, but which are absent at some problems in P (thus concluding inequality) or extending the known proper inclusions between complexity classes or lower bounds on the complexity required to solve certain problems, towards a proper inclusion of P inside NP (the inclusion $P \subseteq NP$ is trivially true). Arguments for independence, i.e., unprovability of either relation ($=$ or $\neq$), would require (i) a choice for an axiomatic system relative to which independence is concluded, and (ii) two models of that axiom system, one in which $P = NP$, and another one in which $P \neq NP$. At least the (four) papers [48], [49], [50], [51] mentioned in the above summary statistics do not follow this general line of arguments, and applications of rigorous techniques (see Section III-D) seem to have not been tried so far, except to study the barriers [52], [53].

A. FORMALIZED PROOFS AROUND P-VS-NP

A few of the proofs we found have been fully or partly formalized, such as [54] (partly, but verified to the extent it was formalized and revised to fix mistakes that the formalization revealed), or [55] and [56] (both for which the formalization disclosed flaws).

Probably the most complete formalization of any $P \neq NP$ proof attempt is due to René Thiemann who formalized the paper "On P Versus NP" by Lev Gordeev using Isabelle [57]. In his paper, Gordeev claims to have shown that no circuit of polynomial size can solve CLIQUE. Gordeev's approach is to generalize "Razborov's theorem" which proves this fact for monotone circuits (i.e. those only composed of $\vee$ and $\wedge$) to non-monotone circuits which can also contain negation. The attempts to verify the paper in Isabelle uncovered problems with the proof. As a byproduct, an Isabelle formalization of Razborov's theorem was published in a computer-verified version [58].

B. APPROACHES TO PROVE EQUALITY

Given that any polynomial-time algorithm to decide any NP-complete problem would be sufficient to equalize the classes, the natural approach to proving equality is by exhibiting an algorithm to solve any of the known NP-complete or -hard problems. Among the several hundred candidates [59], a few turn out to be particularly popular for this goal: these are the clique or independent set problem [11], [12], [23], [24], [56], [60], [61], satisfiability of logical formulas [4], [7], [9], [14], [17], [20], [27], [61], [62], [63], [64], [65], [66], [67], [68], [69], [70], [71], [72], [73], [74], [75], [76], [77], [78], [79], [80], [81], [82], [83], [84], [85], [86], [86], [87], Hamiltonian circuits and the traveling salesperson problem [16], [22], [25], [29], [72], [88], [89], [90], [91], [92], [93], [94], [95], [96], as well as the quadratic assignment problem [25], [97], subset-sum [97], graph isomorphism [67], the polynomial hierarchy and enumerations of perfect matchings [17], diophantine equations [98], maximum cuts [75], constraint satisfaction [82], partition [98], [99], bin packing [28], or fast multiplication of long integers [43] (which is an example of a work concluding equality from a speedup to a computational, not a decision, problem, which is also not asserted as NP-complete).

Some work, unfortunately, seemingly disappeared from the internet [20], [63], [66], [81], were retracted (perhaps by the authors themselves) [84], or were published in (today) inaccessible venues [69].

Among the methods to tackle the problems, linear programming turned out as popular, and used in, e.g., [9], [25], [26], [86], [87], [96], [100], all of which formulate an NP-complete problem as a linear program. The necessary polynomial size of the resulting linear program is in contrast to the results of Yannakakis [10], who proved that any such linear programming formulation of the traveling salesperson problem would be of exponential size. Further methods include modifications of the TMs themselves, such as implementing an oracle query mechanism efficiently [74], converting nondeterminism into determinism while preserving polynomial complexities [100], [101], proving the equality of P and NP to a third class that is introduced newly for this purpose [73], [102], or using category theory [85].

Some works, interestingly, do not allude to P-vs-NP at all, except in the title, such as [103]: this work states the equality in the subtitle, however, considers a linear optimization problem without any integer constraints and with no obvious relation to either P or NP.

C. APPROACHES TO PROVE INEQUALITY

Inequality of P and NP are based on showing strict inclusions [62], exhibiting problems with super-polynomial lower bounds to their complexity [5], [32], [35], [42], [55], [104], [105], [106], [107], [108], [109], [110], [111], [112], [113], [114], [115], [116], such as integer factorization [117], clique problems [118], subset-sum [119], problems in quantum computing [120], [121].

Some methods are model- or proof-theory related [50], [122], [123], [124], such as, e.g., reasoning about the length of proofs [125]. Other attempts dig into the relation between NP and co-NP, resp. other classes (such as Exptime [62]), or new interpretations of NP: [33], [37], [41], [118], [126], [127], [128], [129], [130], [131], [132], [133], [134], up to attempts of refuting the NP-completeness of certain problems [135].

Another popular line goes over one-way functions [41], [54], [96], [136], [137], [138], and a relatively small fraction employs different or heterogeneous combinations of arguments [38], [40], [139], [140], [141], besides some work [37], [38], [39], [41], [104], [141], [142], [143], [144] that seemingly disappeared from the internet so that we cannot make reliable claims about the methods applied therein.

D. APPROACHES TO PROVE UNPROVABILITY (INDEPENDENCE)

Proving that something is not provable in some given axiomatic system is called an *independence proof*. This is a relatively rare kind of argument found throughout mathematics, but a few notable examples do exist such as (mentioning only a small sample here):

- The continuum hypothesis, to which half of the answer was contributed by Gödel [145], and the other half is due to Cohen [146], [147]
- The independence of the axiom of choice from the remaining Zermelo-Fraenkel axioms of set theory was also proven by Paul Cohen,
- The independence of the parallel axiom and Euclid's other axioms of plane geometry was proven by Eugenio Beltrami.

A general proof of independence is a model-theoretic argument formulated relative to a certain set of axioms, such as for example, Zermelo-Fraenkel (ZF), perhaps including the axiom of choice (ZFC). If ψ is a logical formula of which we seek to prove that ϕ cannot be proven from a set of axioms $\mathcal{A}$, we need to prove two things:

1) The existence of a model for $\mathcal{A}$ under which ψ is provably true,
2) and the existence of (another) model for $\mathcal{A}$ under which ψ is provably false,

so that in total, we can conclude that there is no way to logically deduce ψ from $\mathcal{A}$, nor is there a way to refute ψ by proving that the truth of $\neg\psi$ is implied by $\mathcal{A}$.

Hence, like a proof of logical equivalence, an independence proof has "two directions" to show, and we can outline the argument easiest letting ψ be the *parallel postulate*, stating that

> ψ: Given any straight line and a point not on it, there "exists one and only one straight line which passes" through that point and never intersects the first line, no matter how far they are extended [148].

A model under which ψ is true is the plane geometry, which is intuitively easy to believe (yet considerably harder to prove formally). Another model under which ψ is false is the geometry on the surface of a sphere: thinking of a straight line to be a set of points without a defined start or end, it is not difficult to imagine a given circle (as the "line" about which ψ speaks) and a point not on the circle through which we can easily draw two further circles on the surface of a sphere that does not touch the given first line. The other four of Euclid's postulates remain true in both models, the plane and the surface of a sphere. Hence, ψ cannot be logically derived from the other axioms.

In total, we found only four papers claiming the unprovability of P-vs-NP in any sense. However, none of these works applied the proving strategy of constructing models in which both, equality and inequality would hold. Some works [48], [149] argue the impossibility of an answer due to the problem itself being ill-posed. In light of rigorous definitions of both classes, P and NP, together with a well-formed axiomatic foundation of set theory (such as ZFC), this argument would not hold. The work of [49] refers to ZFC and presents a reformulation of the question similar as for relativization along a self-assessment, but was also counter-argued soon after by AMS Mathematical Reviews [2]. The work of [50] modified the Peano axioms for the natural numbers towards two different axiomatic systems, one in which equality of the classes holds, the other, in which P and NP are not equal. In being a change of the axiomatic system, this does not count as an independence proof. It is, however, noteworthy to mention that also relativization (discussed later in Section IV-A) can be interpreted as an extension to the axiomatic system, via the additional oracle assumption, which allows to prove equality or inequality, depending on how the extension (e.g., the oracle) looks like. The authors of [68] and [123] identified a logical inconsistency in ZFC itself and stated the invalidity of the continuum hypothesis (CH) [123] or Cook's theorem that satisfiability is NP-complete [68]. Given that CH already has been proven as independent of ZFC by Kim, and that Cook's theorem has also undergone countless verifications, the far-reaching consequences that [123] gives, including P $\neq$ NP and the invalidity of many fundamental results of complexity theory despite there being manifold verifications and proofs of the opposite, makes this work interesting to mention, yet invalid from a scientific perspective. Finally, the work of [51] more or less philosophically speaks about the problem, unfortunately, neither following the required proof strategy of constructing models, nor being in itself consistent, since the paper title suggests the impossibility of proving how P relates to NP, but the paper itself concludes with the final line that "nonetheless P $\neq$ NP".

IV. FINDINGS

When going into a self-verification of a proof using formalization in languages like Isabelle/HOL, it nonetheless is necessary to understand what dead-ends need to be avoided. To ease following the upcoming descriptions, we refer to Table 2 for a list of symbols.

TABLE 2. List of symbols.

Symbol	Meaning
P	class of decision problems solvable in determinstic polynomial time
NP	class of decision problems solvable in non-determinstic polynomial time
Dtime(t)	class of decision problems L for which at most $t(n)$ deterministic steps to are enough to decide $w \in L$
O, A	general oracle, being a formal language or set whose characteristic function is assumed as computable in constant time
$\tilde{A}$	algebraic version of the oracle A; roughly speaking, this replaces the characteristic function by a low-degree polynomial (interpolating it)
M^A,M^A	Turing machine M with additional ? state to run a query $w \in A$? or $w \in \tilde{A}$? that takes only constant time (by assumption)
$C^A, C^{\tilde{A}}$	extension of a complexity class C, by allowing Turing machines to have access to a normal or algebraic oracle (see above)
ZFC	Zermelo-Fraenkel Set Theory, including the axiom of choice (C)
CT(O_0)	"complexity theory using the oracle O_0", i.e., a version of complexity theory in which all Turing machines have access to the oracle O_0
CT$(*)$	complexity theory using an arbitrary oracle (used to formalize relativization against any oracle)
FP	generic symbol to formulate logical propositions; no meaning by itself, but just a "placeholder" to be filled with a symbol with semantics such as "TheFP" (see next)
TheFP	class of languages decidable in polynomial time (with or without using oracle Turing machines); can be P but also something more general (using oracles)
ψ, ϕ	general logical formulae

A. BARRIERS: (AFFINE) RELATIVIZATION, ALGEBRIZATION AND NATURAL PROOFS

The earliest barrier against separating or equalizing P and NP is based on the concept of oracle TMs: Fix any formal language $A \subseteq \Sigma^*$ over a (fixed) alphabet Σ, and define a modified TM with the ability to decide $w \overset{?}{\in} A$ in a constant amount of steps for any w that it produces on its tape. We call A an *oracle*, and a Turing machine M endowed with the capability of querying A is an *oracle-TM* denoted as M^A. Complexity classes are definable in a canonical way by allowing the defining TM to access A, which naturally leads to generalizations of P and NP as P^A and NP^A. In such worlds, which we call relativized, a famous result is the following:

Theorem 1 (Baker, Gill, and Solovay [150]): There are oracle sets A, B for which $\mathrm{P}^A = \mathrm{NP}^A$ and $\mathrm{P}^B \neq \mathrm{NP}^B$.

This has several consequences, among them:

1) Oracles, as a proof technique, apparently cannot settle the issue between P and NP, since there are oracles that lead to either possible outcome.
2) Given any *PROOF* towards some (any) relation between P and NP, one may simply rephrase *PROOF* into speaking about oracle-TM whenever it uses a (normal) TM. If the proof remains intact under such generalizations, in which case it is said to *relativize*, it will – no matter what it originally claimed about P-vs-NP– be in contradiction with Theorem 1. Hence, the usual conclusion is that any such relativizing proof cannot be effective against the original question.
More generally (without alluding to P-vs-NP), if a result is such that its proof generalizes towards another setting (e.g., a different space, weaker hypotheses, or similar), but the conclusion is provably wrong in the generalized context, then the original proof must be flawed.

Such relativization, for example, applies to arguments based on diagonalization. The technique itself does not hinge on (not) using oracles, so it is straightforward to modify well-known results, such as the deterministic time hierarchy theorem (DTHT).

Theorem 2: Let $f : \mathbb{N} \to \mathbb{N}$ be time-constructible, and let $t, T : \mathbb{N} \to \mathbb{N}$ be such that $\liminf_{n\to\infty} t(n) \log t(n)/T(n) = 0$ and $t(n) \geq n$ for all n. Then,

$$\mathrm{Dtime}(t) \subsetneq \mathrm{Dtime}(T), \qquad (1)$$

whose proof uses diagonalization, can be modified into stating, under the same hypotheses, that

$$\mathrm{Dtime}^A(t) \subsetneq \mathrm{Dtime}^A(T), \qquad (2)$$

for complexity classes that have access to the oracle A; and the oracle A can be *any language* here.

Proofs have therefore been said to "relativize" if they go through just as usual, under *every possible oracle*, i.e.,

$$\left.\begin{array}{l}\text{IF } \mathit{PROOF} \text{ is correct}\\ \text{THEN } \forall \text{ oracles } O:\\ \qquad [\text{ the relativized } \mathit{PROOF}^{O} \text{ is correct}],\end{array}\right\} \tag{3}$$

or by contraposition

$$\left.\begin{array}{l}\text{IF } \exists \text{ an oracle } O\colon [\mathit{PROOF}^{O} \text{ is wrong}]\\ \text{THEN the (original) } \mathit{PROOF} \text{ is wrong}\end{array}\right\} \tag{4}$$

No matter what PROOF^{O} thus concludes about P-vs-NP, it will contradict Theorem 1 and hence be found wrong in (4) by putting $O = A$ or $O = B$ with the oracles from the Baker-Gill-Solovay theorem. Then, (4) indicates to abandon *PROOF* for this reason. This is how the relativization barrier is usually applied.

In the past, some results were found to not relativize because their reasoning about Turing machines is so specific that the oracle query mechanism, e.g., modeled by a designated query state "?", is not trivially considerable in the proof without substantially changing the argument. An example of such a result was the Cook-Levin theorem. The discovery of more general barriers, such as algebrization [53], [151], however, exhibited also these results as relativizing, only under a modified form of oracle. Technically, the oracle was changed from a set of strings to a family of low-degree polynomials that extend the space of possible queries and enable reasoning with techniques like arithmetization, as introduced in the context of proving the famous equality IP $=$ PSpace. Using the resulting generalized concept of *algebraic oracles*, we have a sibling to the second part of Theorem 1:

Theorem 3 (Aaronson, Widgerson [151]): There exists an algebraic oracle $\tilde{A}$ such that $\mathrm{P}^{\tilde{A}} = \mathrm{NP}^{\tilde{A}}$. As a consequence, any proof of $\mathrm{P} \neq \mathrm{NP}$ will require non-algebrizing techniques.

Further refinements and other barriers (not chronologically mentioned here) are local checkability [152], and natural proofs [153]. The most recent unified account for relativization and algebrization was presented in [52] and [154], which described how to integrate the oracle assumption in the axiomatic system from which P-vs-NP shall be analyzed. Specifically, introducing the concept of an affine oracle, they call a proof relativizing if it is a theorem of the axiomatic system extended by "the oracle assumption" TheOA,

$$\mathrm{ZFC} + \underbrace{[O \text{ is a language}] + [\text{FP equals (TheFP)}^{O}]}_{=:\ \text{TheOA}} \tag{5}$$

in which

- ZFC is the usual Zermelo-Fraenkel axiomatization of set theory including the axiom of choice,
- O is a language in the complexity-theoretic sense, considered as a mapping $\{0, 1\}^* \to \{0, 1\}$ (letting the set be represented by its characteristic function acting on strings)
- FP is a symbol of the signature to formulate propositions, and taken to be the class "TheFP" of polynomial time computable functions (defined in the standard way using Turing machines or any equivalent thereof), with explicit oracle access to O.

In the notation of [52], the collection of all theorems implied by the axiomatic system (5) will depend on the oracle, and therefore be denoted as the relativized complexity theory $CT(O_0)$ for the fixed oracle language O_0. The derived symbols $CT(*)$ then denote the *relativized complexity theory* (against all possible oracles), while $CT(0)$ is the non-relativized universe, represented by the empty oracle $O(x) = 0$ for all $x \in \{0, 1\}^*$, in which classes defined with access to the oracle, such as FP, match their conventional counterparts.

In other words, the axiomatic system is extended by letting the property of "polynomial time computability" be *re-defined* under the additional capability of evaluating the function O on any input, counting time- and space complexity for algorithms that can use access to O in their "basic instruction set" (thereby also naturally settling questions of how the oracle tape is used, and whether its use counts towards time or space complexity, as [155] raised as an issue). Then, one can rigorously define a statement to *relativize against a specific oracle* O_0 if it is a theorem of (5) with O_0 substituted for O [52, Def.3], or just as *relativizing* if it is a theorem relative to every language [52, Def.4]. This view allows to reproduce various past results from the literature that are known to (not) relativize, plus discover new relativizing and non-relativizing results. The beauty of this approach from first principles (i.e., axioms) is that no change to the usual definitions from complexity theory is required, since the re-interpretation of the symbol FP as using (or not using) an oracle naturally endows all computational mechanisms, formulated with use of the symbol FP and/or O, with the power to query the oracle.

This naturally covers the unrelativized world by using an empty oracle as a function that returns a constant value for all arguments, and also not requires to leave the unrelativized world at any time, since we can just re-interpret the symbol FP in the statements to be proven. This models relativization as a syntactic change of letting FP change its meaning throughout the entire *PROOF* from the unrelativized version (e.g., with an empty oracle), to the oracle-enhanced version with an explicit additional capability to evaluate the function O_0, i.e., equivalently, querying the oracle O_0. Consequently, and continuing the previous thought, the definition of, for example, Dtime(t) under FP becomes the familiar concept based on Turing machines, but the identical definition under TheFP will lead to $\mathrm{Dtime}^{O}(t)$ with the oracle O coming in via TheFP.

Under a proper modification of the proof to take explicit advantage of the oracle, the conceptualization of oracles as provided by [52] together with proof assistants can address prior criticism about oracle results as uttered by R. Lipton [156], who raised the question about which would be the specific predicate to decide whether *PROOF* relativizes or not.

B. OPEN RESEARCH QUESTIONS

The guidance that barriers provide is to avoid known dead-end arguments, such as the fact that a linear program to describe the traveling salesperson problem is necessarily of exponential size [10] and hence takes longer than polynomial, even if solved by a polynomial-time algorithm. Thus, the known barriers to avoid are at least:

1) Affine relativization (including the (original) relativization and (younger) algebrization barrier) [52]
2) Naturalization [153] (for example, by diagonalization, but this needs care to not fall victim to the relativization barrier)
3) Linear programming, as it is known to fail for at least some NP-complete problems like Traveling Salesperson [10]

There are several recommendations about how to bypass relativization barriers, such as the interpolation technique [52], [151] or using methods from communication complexity [52, Sec.7.2]. Conversely, some methods are explicitly deprecated, such as diagonalization [150], although specifically this technique is known to bypass another barrier known as naturalization [153].

In addition, exact predicates and conditions to recognize techniques and arguments that fall under the above dead-ends are an open issue of research. More precisely, how can we "automatically" in the sense of algorithmically, analyze a given series of (logical) arguments to recognize it as algebrizing, relativizing, affinely relativizing, or similar? The power of proof assistants is interesting to explore to this end, a starting point of which has partially been made in past work. While the above are already explicit routes to fail, a general classification of what techniques could work towards settling P-vs-NP is another challenge for future research.

A potential route to explore further is analogous to oracle extensions, but rather directed to the opposite of limiting the underlying axiomatic foundation or model mathematics intentionally. For example, by removing the axiom of choice or generally asking for purely constructive arguments (diagonalization would then also naturally fall out of such considerations). While we did not explicitly screen the papers here for proofs under such limitations, some work does explicitly consider the axiom of choice [49], [134], respectively consistency thereof with the P-vs-NP question [85], [123], occasionally also with far-reaching claims of fundamental results of complexity theory (including Cook's and Fagin's theorems) to be wrong and ZFC to be overall inconsistent [68], [123]. Despite any of these past claims, the idea of trying to prove equality, inequality, or independence of the question from a reduced axiomatic system or "non-standard" models (of mathematics or complexity theory) seems widely unexplored.

C. USING PROOF ASSISTANTS

The actual value of a proof assistant remains in its objectiveness of checking, being unbiased even against the user of the proof assistant itself. Some proof assistants like Isabelle/HOL provide instructions to skip parts of the proof (in Isabelle/HOL by the `sorry` keyword) to make the system accept unproven statements. The use of the `sorry` keyword in Isabelle/HOL is to be considered as dangerous as using words like "trivial" or "obviously" in mathematics, since what hides behind what has been skipped can be arbitrarily small, arbitrarily large, or even impossible to prove. That said, if the entirety of a proof is formalized in an automated proof assistant and verified without any skipped parts (by `sorry` or comparable constructs), then its correctness would be strongly certified (up to possible errors in the proof assistant software itself).

If a counter-argument is made based on barriers like the above, then either the counterargument is itself wrong, or the gap in the proof would have been identifiable as a `sorry` or missing details or inaccuracies in the underlying definitions (cf. the above discussion about consistency between definitions on paper and in the proof assistant, for example, the size of a problem instance upon a Karp reduction may have been missed. In a case where an NP-complete problem is transformed into a linear program, this could result in worst-case exponential size, thus making the proof fail against the aforementioned barrier, and similarly, for the relativization and other barriers).

Leaving "gaps" in the formalized proof is not per se an indication of incorrectness; any part verifiable by the proof assistant already saves a human's time for peer review, and at the same time, provides the (only) reasonable points to counter-argue.

The manual labor left, by formalization, thus boils down to:

- writing the formalized proof itself,
- checking consistency between the definitions used in a paper version of a proof versus the definitions used inside the proof assistant
- arguing about the remaining "gaps" in the arguments (skipped sub-proofs).

The last point is most considerable in terms of efforts. In the project underlying this paper, we selected two papers (one [56] claiming P $=$ NP, the other [54] claiming P $\neq$ NP, both from the set of candidate proofs; no paper was selected from the "unprovable" category since no work showed the required structure of an independence proof), and formalized them in Isabelle/HOL.[6] For [56], issues with the arguments were revealed by the formalization.[7] For [54], the formalization was accomplished only partially (up to approximately 30% of 50 pages at an effort of (equivalently) 23.6 person-months). Issues were found, but could all be fixed, so that [54] is formalized up to $\approx$ 30%, with only a few `sorry`s that require special attention.

[6] The choice was made from a much longer list of candidate systems that Wikipedia surveys (https://en.wikipedia.org/wiki/Proof_assistant) as of 2023.

[7] The authors of this work contacted the author of [56] for involvement to fix the issues, but received no response.

The formalizations are all open access available in a public `github` organization,[8] with the possibility and explicit *invitation* to add one's own formalization as a new repository, continuing the proposal of this work. The two formalizations reported above are included as examples there.

D. CONCLUSION

We believe that automated proof assistants can be one possible way to handle the lot of ideas versus the relatively smaller capabilities of expert peer reviews. The number of proofs being proposed per year has apparently led to some sort of "proof-fatigue", at least visible in the insignificant excitement of the community about yet another proof to be published. The Riemann-hypothesis is another example of a conjecture whose fate will depend on the right idea to not go unseen in a flood of flawed attempts.

Our hope in this work is to empower anyone aiming to contribute solutions at the public `github` organization[8], to self-check their work, with some guidance towards known pitfalls and dead-ends that have already been explored in the literature. Equally important are exact predicates and conditions to unambiguously and objectively decide about relativization or other barriers.

Automated proof assistants can offer (i) objective checking, ignorant of affiliations, education, subjective opinions, and other factors, and (ii) help to identify errors in a seemingly logical argument, very much like examples and exercises help a student to develop deep knowledge and insights.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for valuable and much-appreciated suggestions to earlier versions of this manuscript.

REFERENCES

[1] W. I. Gasarch, "Guest column: The third P=?NP poll," *ACM SIGACT News*, vol. 50, pp. 38–59, Mar. 2019.

[2] G. J. Woeginger. *The P-Versus-NP Page*. Accessed: Jan. 8, 2020. [Online]. https://www.win.tue.nl/~gwoegi/P-versus-NP.htm

[3] S. Aaronson. *My 116-Page Survey Article on P VS. NP: Better Late Than Never*. Accessed: Dec. 9, 2019. [Online]. Available: https://www.scottaaronson.com/blog/?p=3095

[4] S. Gubin, "Complementary to Yannakakis' theorem," *J. Combinat. Math. Combinat. Comput.*, vol. 74, pp. 313–321, Jul. 2010.

[5] R. Hofman, "Complexity considerations, cSAT lower bound," 2007, *arXiv:0704.0514*.

[6] I. Christopher, D. Huo, and B. Jacobs, "A critique of a polynomial-time SAT solver devised by Sergey Gubin," 2008, *arXiv:0804.2699*.

[7] V. F. Romanov, "Non-orthodox combinatorial models based on discordant structures," 2010, *arXiv:1011.3944*.

[8] D. Gusev. (Sep. 25, 2013). *Open Letter | 3-SAT: Novel Model*. Accessed: Apr. 11, 2023. [Online]. Available: https://romvf.wordpress.com/2011/01/19/open-letter/

[9] A. A. Maknickas, "Linear programming formulation of Boolean satisfiability problem," in *Transactions on Engineering Technologies*, G.-C. Yang, S.-I. Ao, X. Huang, and O. Castillo, Eds. Dordrecht, The Netherlands: Springer, 2014, pp. 305–321.

[10] M. Yannakakis, "Expressing combinatorial optimization problems by linear programs," *J. Comput. Syst. Sci.*, vol. 43, no. 3, pp. 441–466, Dec. 1991.

[11] P. Tamta, B. P. Pande, and H. S. Dhami, "A polynomial time solution to the clique problem," 2014, *arXiv:1403.1178*.

[12] M. LaPlante, "A polynomial time algorithm for solving clique problems," 2015, *arXiv:1503.04794*.

[13] H. A. Cardenas, C. Holtz, M. Janczak, P. Meyers, and N. S. Potrepka, "A refutation of the clique-based P=NP proofs of LaPlante and Tamta-Pande-Dhami," 2015, *arXiv:1504.06890*.

[14] C. Sauerbier, "Three complete deterministic polynomial algorithms for 3SAT," Dec. 2019, *arXiv:cs/0205064*.

[15] D. Zhu, J. Luan, and S. Ma, "Hardness and methods to solve CLIQUE," *J. Comput. Sci. Technol.*, vol. 16, no. 4, pp. 388–391, Jul. 2001.

[16] C. A. H. Krieger, "Polynomial method for detecting a Hamiltonian circuit," U.S. Patent 20 080 071 849, Mar. 20, 2008.

[17] J. Aslam, "An extension of the permutation group enumeration technique (collapse of the polynomial hierarchy: **NP = P**)," 2008, *arXiv:0812.1385*.

[18] F. Ferraro, G. Hall, and A. Wood, "Refutation of Aslam's proof that NP = P," 2009, *arXiv:0904.3912*.

[19] J. Aslam, "Response to refutation of Aslam's proof that NP = P," 2009, *arXiv:0906.5112*.

[20] M. Telpiz. *Miron Telpiz's P=NP Page*. Accessed: Apr. 11, 2023. [Online]. Available: https://web.archive.org/web/20071005034903/ and http://www.tarusa.ru/~mit/ENG/eng.html

[21] Ž. Hutinski and M. Malkoevic, "From the editor," *J. Inf. Organizational Sci.*, vol. 29, no. 2, Mar. 2012.

[22] S. Gubin, "A polynomial time algorithm for the traveling salesman problem," 2006, *arXiv:0610042*.

[23] A. D. Plotnikov, "Polynomial time partition of a graph into cliques," *Electron. J., Southwest J. Pure Appl. Math.*, vol. 1, pp. 16–29, Nov. 1996.

[24] T. Pushan, "An algorithm with polynomial time complexity for finding clique in a graph," in *Proc. 5th Int. Conf. (CAD&CG)*, Shenzhen, China, 1997, pp. 500–505.

[25] M. Diaby, "Equality of complexity classes P and NP: Linear programming formulation of the quadratic assignment problem," May 2007, *arXiv:cs/0609004*.

[26] M. Diaby and M. H. Karwan, *Advances in Combinatorial Optimization: Linear Programming Formulations of the Traveling Salesman and Other Hard Combinatorial Optimization Problems*. Singapore: World Scientific, Apr. 2016.

[27] M. Mueller, "Polynomial exact-3-SAT-solving algorithm," *Int. J. Eng. Technol.*, vol. 9, no. 3, p. 670, Aug. 2020.

[28] V. Voinov and M. Rahmanov, "2-, 3-, and 4-partition problems and their relation to the equality P=NP," *Central Asia Bus. J.*, vol. 11, no. 2, pp. 34–46, 2020.

[29] V. Yatsenko, "Fast exact method for solving the travelling salesman problem," Feb. 2007, *arXiv:cs/0702133*.

[30] C. Clingerman, J. Hemphill, and C. Proscia, "Analysis and counterexamples regarding Yatsenko's polynomial-time algorithm for solving the traveling salesman problem," 2008, *arXiv:0801.0474*.

[31] K.-B. Nam, S. H. Wang, and Y. G. Kim, "Linear algebra, lie algebra and their applications to P versus NP," *J. Appl. Algebra Discrete Struct.*, vol. 2, no. 1, pp. 1–26, 2004.

[32] A. Annila, "Physical portrayal of computational complexity," *ISRN Comput. Math.*, vol. 2012, pp. 1–15, Mar. 2012.

[33] J. Kim, "P is not equal to NP by modus tollens," 2014, *arXiv:1403.4143*.

[34] D. Hassin, A. Scrivener, and Y. Zhou, "Critique of J. Kim's 'P is not equal to NP by modus Tollens,'" 2014, *arXiv:1404.5352*.

[35] A. L. Barbosa, "P != NP proof," 2009, *arXiv:0907.3965*.

[36] L. A. Hemaspaandra, K. Murray, and X. Tang, "Barbosa, uniform polynomial time bounds, and promises," 2011, *arXiv:1106.1150*.

[37] A. Blinder, "A possible new approach to resolving open problems in computer science," 2009.

[38] V. Deolalikar, "$P \neq NP$," 2010.

[39] B. Wen and Y. Lin, "The answer to the P/NP problem is $P \neq NP$!" 2010.

[40] G. R. Diduch, "P vs NP," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 12, pp. 165–167, Jan. 2012.

[41] F. Vega, "P versus UP," *IEEE Latin Amer. Trans.*, vol. 10, no. 4, pp. 2006–2009, Jun. 2012.

[42] R. C. Valeyev, "The lower border of complexity of algorithm of the elementary NP-complete task (The most condensed Version)," *World Appl. Sci. J.*, vol. 24, no. 8, 2013, Art. no. 10721083.

[43] Z. A. Chotchaeva, "P vs NP: P is equal to NP: Desired proof," *Global J. Comput. Sci. Technol.*, vol. 21, pp. 1–11, Sep. 2021.

[44] *P/NP Policy*, ACM, New York, NY, USA, 2023.

[8] https://github.com/CAVE-PNP/cave-pnp

[45] T. Nipkow, L. C. Paulson, and M. Wenzel, *Isabelle/HOL: A Proof Assistant for Higher-Order Logic* (Lecture Notes in Computer Science), no. 2283. Berlin, Germany: Springer, 2002.

[46] S. Böhne and C. Kreitz, "Learning how to prove: From the coq proof assistant to textbook style," *Electron. Proc. Theor. Comput. Sci.*, vol. 267, pp. 1–18, Mar. 2018.

[47] S. Kolodynski. *IsarMathLib*. Accessed: Jun. 20, 2023. [Online]. Available: https://isarmathlib.org/

[48] N. Argall. (2003). *P = NP—An Impossible Question*. [Online]. Available: https://www.win.tue.nl/~wscor/woeginger/P-versus-NP/argall.txt

[49] N. C. A. da Costa and F. A. Doria, "Consequences of an exotic definition for P=NP," *Appl. Math. Comput.*, vol. 145, nos. 2–3, pp. 655–665, Dec. 2003.

[50] S. Jaeger, "Computational complexity on signed numbers," 2011, *arXiv:1104.2538*.

[51] N. L. Malinina, "On the principal impossibility to prove P=NP," 2012, *arXiv:1211.3492*.

[52] B. Aydinlioğlu and E. Bach, "Affine relativization: Unifying the algebrization and relativization barriers," *ACM Trans. Comput. Theory*, vol. 10, no. 1, pp. 1–67, Jan. 2018.

[53] R. Impagliazzo, V. Kabanets, and A. Kolokolova, "An axiomatic approach to algebrization," in *Proc. 41st Annu. ACM Symp. Theory Comput.*, May 2009, p. 695.

[54] S. Rass, "On the existence of weak one-way functions," 2016, *arXiv:1609.01575*.

[55] L. Gordeev, "On p versus NP," 2020, *arXiv:2005.00809*.

[56] Z. O. Akbari, "A polynomial-time algorithm for the maximum clique problem," in *Proc. IEEE/ACIS 12th Int. Conf. Comput. Inf. Sci. (ICIS)*, Jun. 2013, pp. 503–507.

[57] L. Gordeev, "On P versus NP," *ArXiv*, Oct. 2022, doi: 10.48550/arXiv.2005.00809.

[58] R. Thiemann, "Clique is not solvable by monotone circuits of polynomial size," *Arch. Formal Proofs*, May 2022. [Online]. Available: https://isa-afp.org/entries/Clique_and_Monotone_Circuits.html

[59] M. R. Garey and D. S. Johnson, *Computers and Intractability*. New York, NY, USA: Freeman, 1979.

[60] J. I. Alvarez-Hamelin, "Is it possible to find the maximum clique in general graphs?" 2011, *arXiv:1110.5355*.

[61] A. D. Plotnikov, "Experimental algorithm for the maximum independent set problem," 2007, *arXiv:0706.3565*.

[62] S. Gram, "Redundancy, obscurity, self-containment & independence," in *Information and Communications Security*, S. Qing, T. Okamoto, and J. Zhou, Eds. Berlin, Germany: Springer, 2001, pp. 495–501.

[63] L. Kolukisa, "Two dimensional formulas and tautology checking," 2005. Accessed: Jan. 22, 2024. [Online]. Available: https://web.archive.org/web/20051229233612/ and http://geocities.com/lkoluk2003/

[64] D. Uyar, "Tautology problem and two dimensional formulas," Nov. 2017. Accessed: Apr. 11, 2023. [Online]. Available: https://vixra.org/abs/1711.0113

[65] F. Capasso, "A polynomial-time heuristic for circuit-SAT," 2005, *arXiv:cs/0511071*.

[66] M. I. Telpiz, "Sigma-notation and the equivalence of P and NP Classes," *J. Inf. Organizational Sci.*, vol. 29, pp. 1–12, Dec. 2005.

[67] M. Delacorte, "Graph isomorphism is PSPACE-complete," 2007, *arXiv:0708.4075*.

[68] R. E. Kamouna, "The Kleene-Rosser paradox, the Liar's paradox & a fuzzy logic programming paradox imply SAT is (NOT) NP-complete," 2008, *arXiv:0806.2947*.

[69] R. V. Hidalgo-Gato, "Método de solución para sistemas de ecuaciones simultáneas sobre un Campo de Galois Y aplicaciones en inteligencia artificial (solution method for systems of simultaneous equations over a Galois Field and artificial intelligence applications)," *Cuban Academy Sci. Ed., Annu. Rep.*, vol. 2, p. 274, Jan. 1985.

[70] L. Salemi, "Method of resolution of 3SAT in polynomial time," 2009, *arXiv:0909.3868*.

[71] N. S. Chaudhari, "Computationally difficult problems: Some investigations," *J. Indian Acad. Math.*, vol. 31, pp. 407–444, 2009.

[72] L. Du, "A polynomial time algorithm for Hamilton cycle with maximum degree 3, 3SAT," 2010, *arXiv:1004.3702*.

[73] C. Wan and Z. Shi, "A proof for P =? NP problem," 2010, *arXiv:1005.3010*.

[74] H. Xiao Wen, "Mirrored language structure and innate logic of the human brain as a computable model of the Oracle Turing machine," 2010, *arXiv:1006.2495*.

[75] M. Katkov, "Polynomial-time approximation scheme for max-cut problem," 2010, *arXiv:1007.4257*.

[76] A. Mukherjee, "The 3-satisfiability problem," 2011, *arXiv:1104.4490*.

[77] M. A. Weiss. *A Polynomial Algorithm for 3-SAT*. Accessed: Apr. 11, 2023. [Online]. Available: https://www.ime.usp.br/~weiss/

[78] M. Groff, "Towards p = NP via k-SAT: A k-SAT algorithm using linear algebra on finite fields," 2011, *arXiv:1106.0683*.

[79] S. Kardash, "Algorithmic complexity of pair cleaning method for k-satisfiability problem. (draft version)," 2011, *arXiv:1108.0408*.

[80] J. W. Steinmetz, "Algorithm that solves 3-SAT in polynomial time," 2011, *arXiv:1110.1658*.

[81] F. Gillet, "Solving 3-SAT and 3-dimensional matching in polynomial time," 2013, *arXiv:1310.1971*.

[82] P. Cui, "Approximation resistance by disguising biased distributions," 2014, *arXiv:1401.6520*.

[83] A. S. Guinea, "Understanding SAT is in P," 2015, *arXiv:1504.00337*.

[84] E. Halylaurin. (Jan. 22, 2024). *An Attempt to Demonstrate P=NP*. Accessed: Apr. 11, 2023. [Online]. Available: https://vixra.org/abs/1605.0278

[85] D. Topchyi, "The theory of plafales. P Vs NP problem solution. Sections 1–7," *OSF Preprints*, Dec. 2020. [Online]. Available: 10.31219/osf.io/9wkxj

[86] A. A. Maknickas, "How to solve kSAT in polynomial time," 2012, *arXiv:1203.6020*.

[87] M. Feldmann, "Solving satisfiability by Bayesian inference," 2012, *arXiv:1205.6658*.

[88] J. Merz. *Merlins-World*. Accessed: Apr. 5, 2023. [Online]. Available: http://www.merlins-world.de/

[89] H. Kleiman, "The asymmetric traveling salesman problem," Dec. 2006, *arXiv:math/0612114*.

[90] K. Riaz and M. S. H. Khiyal, "Finding Hamiltonian cycle in polynomial time," *Inf. Technol. J.*, vol. 5, no. 5, pp. 851–859, Aug. 2006.

[91] G. Zhu, "The complexity of HCP in digraps with degree bound two," 2007, *arXiv:0704.0309*.

[92] X. Jiang, "A polynomial time algorithm for the Hamilton circuit problem," 2013, *arXiv:1305.5976*.

[93] W.-Q. Duan, "A constructive algorithm to prove P=NP," 2012, *arXiv:1208.0542*.

[94] D. Nuriyev, "A DP approach to Hamiltonian path problem," 2013, *arXiv:1301.3093*.

[95] H. Liu, "A algorithm for the Hamilton circuit problem," 2014, *arXiv:1401.6423*.

[96] A. Panyukov, "Polynomial solvability of *NP*-complete problems," 2014, *arXiv:1409.0375*.

[97] A. Bianchini. (2013). *A Polynomial-Time Exact Algorithm for The Subset Sum Problem*. Accessed: Apr. 5, 2023. [Online]. Available: https://www.semanticscholar.org/paper/A-polynomial-time-exact-algorithm-for-the-Subset-Bianchini/e5ac316007199706e9560362ba277e45d45e4eca

[98] Y. Dujardin, "Résolution du 'partition problem' par une approche arithmétique," 2009, *arXiv:0909.3466*.

[99] M. Diaby, "Linear programming formulation of the set partitioning problem," *Int. J. Oper. Res.*, vol. 8, no. 4, p. 399, 2010.

[100] S. V. Yakhontov, "P = NP," 2012, *arXiv:1208.0954*.

[101] Y. Huang, "Testing a new idea to solve the P = NP problem with mathematical induction," PeerJ Inc., London, U.K., Tech. Rep. e1813, Oct. 2015.

[102] F. Vega, "Solution of P versus NP problem," Jun. 2015. Accessed: Apr. 11, 2023. [Online]. Available: https://hal.science/hal-01161668

[103] G. Bolotashvili, "Solution of the linear ordering problem (NP=P)," 2003, *arXiv:cs/0303008*.

[104] M. N. Kupchik, "P versus np problem solution," 2004.

[105] V. Ivanov, "A short proof that NP is Not P," *Int. J. Pure Apllied Math.*, vol. 94, no. 1, pp. 81–88, Jul. 2014.

[106] B. S. Anand, "Is the halting problem effectively solvable non-algorithmically, and is the goedel sentence in NP, but not in P?" 2005, *arXiv:math/0506126*.

[107] J. Meek, "P is a proper subset of NP," 2008, *arXiv:0804.1079*.

[108] S.-A. Tarnlund, "P is not equal to NP," 2008, *arXiv:0810.5056*.

[109] C. Barron-Romero, "The complexity of Euclidian 2 dimension travelling salesman problem versus general assign problem, NP is not P," 2010, *arXiv:1101.0160*.

[110] R. Liao, "The complexity of 3SAT_N and the P versus NP problem," 2011, *arXiv:1101.2018*.

[111] R. V. Yampolskiy, "Construction of an NP problem with an exponential lower bound," 2011, *arXiv:1111.0305*.

[112] C. A. Feinstein, "The computational complexity of the traveling salesman problem," Jan. 2012, *arXiv:cs/0611082*.

[113] K. Kobayashi, "Topological approach to solve P versus NP," 2012, *arXiv:1202.1194*.

[114] J. Fukuyama, “Computing cliques is intractable,” 2013, *arXiv:1305.3218*.

[115] S. Tazawa, “Relationship between circuit complexity and symmetry,” 2012, *arXiv:1207.2171*.

[116] D. Uribe, “P vs. NP,” 2016, *arXiv:1601.03619*.

[117] B. S. Anand, “A density-based approach for non-heuristic approximations of prime counting functions,” 2015, *arXiv:1510.04225*.

[118] G. R. Renjit, “Fixed type theorems,” Feb. 2005, *arXiv:cs/0502030*.

[119] J. Jormakka, “On the existence of polynomial-time algorithms to the subset sum problem,” 2008, *arXiv:0809.4935*.

[120] R. V. Ramos, “Using disentangled states and algorithmic information theory to construct a not P problem,” Dec. 2006, *arXiv:quant-ph/0612001*.

[121] D. Song, “The P versus NP problem in quantum physics,” 2014, *arXiv:1402.6970*.

[122] B. S. Anand, “An elementary proof that $P \neq NP$,” Feb. 2007, *arXiv:math/0603605*.

[123] M. Kim, “Inconsistency of the Zermelo–Fraenkel set theory with the axiom of choice and its effects on the computational complexity,” 2012, *arXiv:1203.0494*.

[124] O. V. German, “Postulate-based proof of the P != NP hypothesis,” 2020, *arXiv:2011.02868*.

[125] C. A. Feinstein, “Complexity science for simpletons,” Jun. 2012, *arXiv:cs/0507008*.

[126] R. A. Cohen, “Proving that P is not equal to NP and that P is not equal to the intersection of NP and Co-NP,” Nov. 2005, *arXiv:cs/0511085*.

[127] G. R. Renjit, “P is not equal to NP,” *ArXiv*, Aug. 2009, doi: 10.48550/arXiv.cs/0611147.

[128] C. Barron-Romero, “The complexity of the NP-class,” 2010, *arXiv:1006.2218*.

[129] A. D. Plotnikov, “On the relationship between classes P and NP,” 2011, *arXiv:1109.5531*.

[130] A. D. Plotnikov, “On the structure of the class NP,” 2013, *arXiv:1304.1307*.

[131] K. Kobayashi, “NP is not AL and P is not NC is not NL is not L,” 2011, *arXiv:1110.0200*.

[132] F. Vega. *Is P Equal to NP*. Accessed: Jun. 15, 2023. [Online]. Available: https://hal.science/hal-01270398/document

[133] M. Hauptmann, “On alternation and the union theorem,” 2016, *arXiv:1602.04781*.

[134] K. E. Kyritsis, “Review of the solutions of the clay millennium problem about P? NP =EXPTIME,” *World J. Res. Rev.*, vol. 13, no. 3, pp. 21–36, Sep. 2021.

[135] J. Meek, “Analysis of the postulates produced by Karp’s theorem,” 2008, *arXiv:0808.3222*.

[136] M. C. Ionescu, “NP–P is not empty,” Sep. 2016, *arXiv:cs/0409039*.

[137] J. A. Arroyo-Figueroa, “The existence of the tau one-way functions class as a proof that P != NP,” 2016, *arXiv:1604.03758*.

[138] A. D. Plotnikov, “About set-theoretic properties of one-way functions,” 2011, *arXiv:1110.3189*.

[139] M. A. P. Moscu, “On invariance and convergence in time complexity theory,” Nov. 2004, *arXiv:cs/0411033*.

[140] L. Gordeev and A. Krebs, “Elementary interpretations of NP vs. P,” Univ. Tübingen, 2004. Accessed: Jan. 22, 2024. [Online]. Available: https://uni-tuebingen.de/securedl/sdl-eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE3MDU5MTYzNzUsImV4cCI6MTcwNjAwNjM3NCwidXNlciI6MCwiZ3JvdXBzIjpbMCwtMV0sImZpbGUiOiJmaWxlYWRtaW5cL1VuaV9UdWViaW5nZW5cL0Zha3VsdGFldGVuXC9JbmZvS29nbmlcL1dTSVwvTFNcL2dvcmRlZXdcL3B1YmxpY2F0aW9uc1wvZTQucGRmIiwicGFnZSI6MzU2NTl9.4DicUQJd3GlY4wWkFG5d3y_ja2PWDyPvL8uZucGyqJM/e4.pdf

[141] *Just How Random are Your Answers*, Holcomb Technologies, Irving, TX, USA, 2011.

[142] C. A. Feinstein, “Evidence that P is not equal to NP and P is not equal to NP,” *CoRR*, vol. cs.CC/0305035, 2003.

[143] B. S. Anand, “A trivial solution to the PVNP problem,” in *Proc. 2008 Int. Conf. Found. Comput. Sci. (FCS)*, Las Vegas, NV, USA, Jul. 2008. [Online]. Available: https://api.semanticscholar.org/CorpusID:1374965

[144] F. V. Delgado, “A solution to the p versus NP problem,” 2010, *arXiv:1011.2730*.

[145] K. Gödel, *The Consistency Axiom Choice Generalized Continuum-Hypothesis With Axioms Set Theory*, no. 3. Princeton, NJ, USA: Princeton Univ. Press, 1940.

[146] P. J. Cohen, “The independence of the continuum hypothesis,” *Proc. Nat. Acad. Sci. USA*, vol. 50, no. 6, pp. 1143–1148, Dec. 1963.

[147] P. J. Cohen, “The independence of the continuum hypothesis, II*,” *Proc. Nat. Acad. Sci. USA*, vol. 51, pp. 105–110, Jan. 1964.

[148] E. W. Weisstein, *Parallel Postulate*. Champaign, IL, USA: Wolfram Research, 2023.

[149] S. Meyer, “Philosophical solution to P=?NP: P is equal to NP,” 2016, *arXiv:1603.06018*.

[150] T. Baker, J. Gill, and R. Solovay, “Relativization of the P?=NP question,” *SIAM J. Comput.*, vol. 4, no. 4, pp. 431–442, 1975.

[151] S. Aaronson and A. Wigderson, “Algebrization,” *ACM Trans. Comput. Theory*, vol. 1, no. 1, pp. 1–54, Feb. 2009.

[152] S. Arora, R. Impagliazzo, and U. Vazirani, “Relativizing versus nonrelativizing techniques: The role of local checkability,” 2007. Accessed: Jan. 22, 2024. [Online]. Available: https://people.eecs.berkeley.edu/~vazirani/pubs/relativizing.pdf

[153] A. A. Razborov and S. Rudich, “Natural proofs,” *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 24–35, 1997.

[154] B. Aydinlioglu and E. Bach, “Corrigendum to affine relativization: Unifying the algebrization and relativization barriers,” *ACM Trans. Comput. Theory*, vol. 11, no. 3, p. 16:1, 2019.

[155] L. Fortnow, “The role of relativization in complexity theory,” *Bull. EATCS*, vol. 52, pp. 229–244, Feb. 1994.

[156] L. Rjlipton. *I Hate Oracle Results*. Blog Entry. Accessed: Jan. 22, 2024. [Online]. Available: https://rjlipton.wpcomstaging.com/2009/05/21/i-hate-oracle-results/

STEFAN RASS (Member, IEEE) received the degree in mathematics and computer science from Universität Klagenfurt (AAU). He is currently a Full Professor with Johannes Kepler University Linz (JKU), Austria, and a member with the Secure and Correct Systems Laboratory. He has authored numerous articles related to practical security, security infrastructures, robot security, and applied statistics and decision theory in security. He participated in various nationally and internationally funded research projects as well as being a contributing researcher in many EU projects and offering consultancy services to the industry. His research interests include decision theory and game-theory with applications in system security, especially robotics security, as well as complexity theory, statistics, and information-theoretic security.

MAX-JULIAN JAKOBITSCH received the B.Sc. degree in computer science from the University of Klagenfurt. He is currently a member with the Institute for Artificial Intelligence and Cybersecurity. He has been involved in various projects at the institute, related to implementations of cryptographic algorithms, and also involved on formal methods for program and proof checking.

STEFAN HAAN received the B.Sc. degree in computer science from the University of Klagenfurt. He is a member with the Institute for Artificial Intelligence and Cybersecurity. He studies mathematics and computer science and also involved on formal methods and program verification using Coq.

MORITZ HIEBLER is currently a former member with the Institute for Artificial Intelligence and Cybersecurity. He studied mathematics and has a particular interest in number theory, algebra, and geometry.