SAT Solvers and Computer Algebra Systems: A Powerful Combination for Mathematics

Curtis Bright
University of Waterloo
Waterloo, Ontario, Canada
cbright@uwaterloo.ca

Ilias Kotsireas
Wilfrid Laurier University
Waterloo, Ontario, Canada
ikotsire@uwaterloo.ca

Vijay Ganesh
University of Waterloo
Waterloo, Ontario, Canada
vganesh@uwaterloo.ca

ABSTRACT

Over the last few decades, many distinct lines of research aimed at automating mathematics have been developed, including computer algebra systems (CASs) for mathematical modelling, automated theorem provers for first-order logic, SAT/SMT solvers aimed at program verification, and higher-order proof assistants for checking mathematical proofs. More recently, some of these lines of research have started to converge in complementary ways. One success story is the combination of SAT solvers and CASs (SAT+CAS) aimed at resolving mathematical conjectures.

Many conjectures in pure and applied mathematics are not amenable to traditional proof methods. Instead, they are best addressed via computational methods that involve very large combinatorial search spaces. SAT solvers are powerful methods to search through such large combinatorial spaces—consequently, many problems from a variety of mathematical domains have been reduced to SAT in an attempt to resolve them. However, solvers traditionally lack deep repositories of mathematical domain knowledge that can be crucial to pruning such large search spaces. By contrast, CASs are deep repositories of mathematical knowledge but lack efficient general search capabilities. By combining the search power of SAT with the deep mathematical knowledge in CASs we can solve many problems in mathematics that no other known methods seem capable of solving.

We demonstrate the success of the SAT+CAS paradigm by highlighting many conjectures that have been disproven, verified, or partially verified using our tool MathCheck. These successes indicate that the paradigm is positioned to become a standard method for solving problems requiring both a significant amount of search and deep mathematical reasoning. For example, the SAT+CAS paradigm has recently been used by Heule, Kauers, and Seidl to find many new algorithms for 3×3 matrix multiplication.

ACM Reference Format:
Curtis Bright, Ilias Kotsireas, and Vijay Ganesh. 2019. SAT Solvers and Computer Algebra Systems: A Powerful Combination for Mathematics. In *Proceedings of CASCON'19*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

CASCON'19, November 4–6, 2019, Toronto, Ontario, Canada

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The development of computer science has transformed the practice of mathematics. The practical algorithms designed by computer scientists have profoundly changed how many mathematical conjectures are proposed, studied, and resolved. For example, the fields of *satisfiability checking* and *symbolic computation* have each been paradigm-shifting in this way. They have allowed mathematicians the ability to solve problems much larger than ever dreamt of in the past, the ability to pose and solve entirely new kinds of mathematical conjectures, and the ability to verify their solutions to unprecedented levels.

Despite a common background and over a hundred years of combined successful progress, these two fields have developed mostly independently of each other and have little common overlap [1]. It is in the interest of the working mathematician or computer scientist to have familiarity with the techniques of these fields, as they have broad (and often surprising) applicability. This article provides an overview of these fields with an emphasis on how the techniques of each field have been applied to resolve mathematical conjectures—and how combining the techniques of each field has resolved conjectures and solved problems that were out of reach of both fields.

Satisfiability checking. The Boolean satisfiability (SAT) problem asks if it is possible assign the variables in a Boolean logic expression in such a way that the expression becomes true. In the 1970s, the Cook–Levin theorem demonstrated that the SAT problem is NP-complete resulting in a pessimism that SAT problems are infeasible to solve in practice. Despite this, research in the engineering of SAT solvers has discovered algorithms and heuristics capable of solving enormous SAT instances that cannot currently be solved by any other method. This "SAT revolution" has had dramatic consequences for hardware and software designers who now use SAT solvers on a daily basis [40].

In fact, SAT solvers have become so successful that Heule, Kullmann, and Marek [28] call them the "best solution in most cases" for performing large combinatorial searches. Recently SAT solvers have been spectacularly applied to a number of long-standing mathematical problems including the Erdős discrepancy conjecture (open for 80 years) [29], the Boolean Pythagorean triples conjecture (open for 30 years) [28], and the determination of the fifth Schur number (open for 100 years) [26]. We briefly outline how SAT solvers were successful on these problems in Section 2.1.

Despite these successes, SAT solvers are known to not perform well for all kinds of combinatorial searches such as those that require advanced mathematics. For example, Arunachalam and Kotsireas [2] have shown that searching for mathematical objects defined by autocorrelation relationships are hard for current SAT solvers. Similarly, Van Gelder and Spence [39] have shown that proving the nonexistence of certain combinatorial designs (even some that have intuitively very easy nonexistence proofs) produce small but very difficult instances for SAT solvers.

Symbolic computation. Symbolic computation or computer algebra is the branch of computer science concerned with manipulating algebraic expressions and other mathematical objects. It has been studied for over sixty years and its successes has lead to the development of computer algebra systems (CASs) that can now automatically solve many theoretical and practical mathematical problems of interest. For example, a modern computer algebra system has functionality for things such as Gröbner bases, cylindrical algebraic decomposition, lattice basis reduction, linear system solving, arbitrary and high precision arithmetic, interval arithmetic, linear and nonlinear optimization, Fourier transforms, Diophantine solving, computing automorphism groups, graph algorithms like determining if a graph has a Hamiltonian cycle, and many other basic operations like computing the derivative of a function.

Computer algebra is widely used in engineering and science. For example, the 1999 Nobel prize in physics was awarded to Gerardus 't Hooft and Martinus J. G. Veltman for using computer algebra to place particle physics on "a firmer mathematical foundation". Computer algebra has also been used to resolve a number of long-standing mathematical conjectures. Three well-known examples of this are the alternating sign matrix conjecture (open for 15 years) [43], the Mertens conjecture (open for 100 years) [34], and the Kepler conjecture (open for nearly 400 years) [25]. We briefly discuss how computer algebra was used to solve them in Section 2.2.

Despite these successes, computer algebra systems are not optimized for all types of problems. In particular, they are typically not optimized to perform the kind of general-purpose search with learning that SAT solvers excel at. In other words, problems that require searching through a large combinatorial space will probably not be solved most effectively by a computer algebra system.

The best of both worlds. In this paper we overview the new "SAT+CAS" paradigm that harnesses the search power of SAT solvers and the mathematical abilities of CASs. This approach provides the best aspects of both the SAT and CAS approaches while minimizing the weaknesses of each respective tool. For example, one of the primary drawbacks of SAT solvers is that they lack mathematical expressiveness—many mathematical concepts are difficult or even impossible to efficiently encode in Boolean logic. On the other hand, a huge variety of mathematical concepts can easily be expressed in a CAS. Thus, the SAT+CAS paradigm combines the search power of a SAT solver with the expressive power of a CAS.

Recently the SAT+CAS paradigm has been used to make progress on a number of conjectures from combinatorics, graph theory, and number theory. In particular, it has verified a conjecture of Craigen, Holzmann, and Kharaghani, found three new counterexamples to the good matrix conjecture, verified the smallest counterexample of the Williamson conjecture, and is responsible for the current best known results in the even Williamson, Ruskey–Savage, Norine, and best matrix conjectures. We give an overview of these conjectures and how our SAT+CAS system MathCheck (available at uwaterloo.ca/mathcheck) was used to produce these results in Section 3. A high-level diagram of how MathCheck combines SAT solvers with

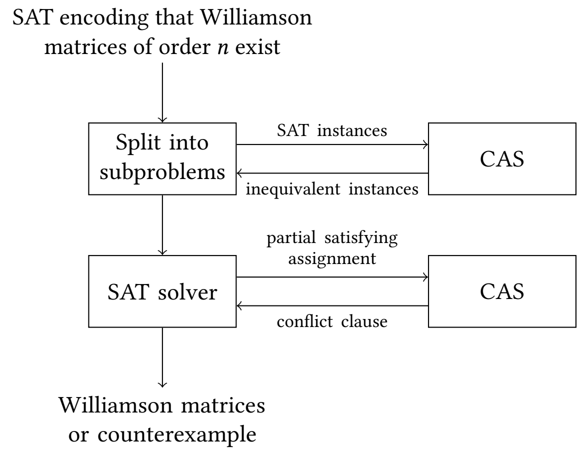

Figure 1: A diagram outlining how the SAT+CAS paradigm is applied to the Williamson conjecture.

CASs is shown in Figure 1. We also briefly discuss how Heule, Kauers, and Seidl have recently used the SAT+CAS paradigm to find numerous new ways of multiplying 3×3 matrices [27]. Finally, we summarize the kinds of problems for which individually the SAT and CAS paradigms are insufficient but for which the SAT+CAS paradigm has been successful in Section 4.

2 PRIOR WORK

In this section we overview the fields of satisfiability checking, symbolic computation, and the kinds of conjectures resolved using the tools of these fields. As we will see, these fields have been applied to resolve an impressive variety of conjectures. Satisfiability checking is particularly good at solving conjectures that can be expressed only using simple constraints but require an enormous search, while symbolic computation is particularly good at solving conjectures that require a lot of complicated mathematical calculations but not a lot of search.

2.1 SAT solving

The techniques developed by the field of satisfiability checking has recently allowed SAT solvers to resolve mathematical conjectures requiring enormous searches. In this section we discuss three of these conjectures.

Erdős discrepancy conjecture. In the 1930s, the prolific mathematician Paul Erdős conjectured that for any infinite $\{\pm 1\}$-sequence $X = (x_1, x_2, \dots)$ the quantity $D_X(n,k) := \left|\sum_{i=1}^{n} x_{ki}\right|$ can be made arbitrarily large by choosing appropriate n and k. In 2010, the Polymath project studied the conjecture and discovered many sequences X of length 1124 with $D_X(n,k)$ at most 2 for all choices of n and k for which this quantity is defined. The sequences were found using a custom computer program and despite expending a lot of computing effort no longer sequences with this property were found. Fields medalist Timothy Gowers would later say "That was enough to convince me that 1124 was the correct bound [for the length of sequences X with $D_X(n,k)$ at most 2]."

In 2014, Konev and Lisitsa [29] showed that 1124 was not the correct bound by using a SAT solver to find a sequence of length 1160 with $D_X(n,k)$ at most 2 for all n and k. Furthermore, they showed that such a sequence of length 1161 could not exist, thereby resolving the smallest open case of the Erdős discrepancy conjecture. The full conjecture was resolved the next year by Terence Tao [38], building on results of the Polymath project.

Boolean Pythagorean triples conjecture. In the 1980s, mathematician Ronald Graham offered a \$100 prize for an answer to the Boolean Pythagorean triples problem: Is it possible to split the natural numbers $\{1, 2, \dots\}$ into two parts so that all triples (a, b, c) with $a^2 + b^2 = c^2$ are separated? In 2008, Cooper and Poirel [10] found a partition of the natural numbers up to 1344 into two parts with no Pythagorean triple in the same part—this required a custom computer program and hundreds of hours of computing time.

In 2016, Heule, Kullmann, and Marek [28] used a SAT solver to find a partition of the natural numbers up to 7824 into two parts that separated all Pythagorean triples. Furthermore, they showed that it was not possible to improve this bound—there is no 2-partition of the natural numbers up to 7825 that separates all Pythagorean triples. The proof found by the SAT solver was over 200 terabytes and was verified in about 4 CPU years. Ronald Graham accepted this as a resolution of the Boolean Pythagorean triples conjecture and awarded his \$100 prize.

Schur number five. In the 1910s, Issai Schur [36] proved that for any $k \geq 1$ there exists a largest set $\{1, \dots, m\}$ that can be partitioned into k parts such that all triples (a, b, c) with $a + b = c$ are separated. The value of m in the above is known as the *Schur number* $S(k)$. It is possible to check that $S(1) = 1$, $S(2) = 4$, $S(3) = 13$ by hand, and Baumert and Golomb [23] showed that $S(4) = 44$ by a computer search in 1965. Furthermore, Exoo [16] showed that $S(5) \geq 160$ in 1994 using a combinatorial optimization algorithm.

In 2017, Heule [26] used a SAT solver to show that any partition of $\{1, \dots, 161\}$ into 5 parts will not separate all triples (a, b, c) with $a + b = c$ and therefore showed that $S(5) = 160$. The proof produced by the SAT solver was two petabytes in size and was verified by a formally-verified proof checker using about 36 CPU years.

2.2 Computer algebra

The techniques developed in the field of computer algebra have been applied to a huge number of engineering, scientific, and mathematical problems. In this section we discuss three conjectures where techniques from computer algebra were essential in the resolution of the conjecture.

Mertens conjecture. In 1885, Thomas Stieltjes conjectured (and later independently by F. Mertens) what is now known as the Mertens conjecture. The Mertens function is defined by $M(x) := \sum_{n \leq x} \mu(n)$ where $\mu(n) := (-1)^k$ if the prime factorization of n consists of k distinct prime factors and $\mu(n) := 0$ if a prime factor appears more than once in the prime factorization of n. The Mertens conjecture is that $|M(x)| < \sqrt{x}$ for all $x > 1$. In the 1970s, the Mertens conjecture was shown to hold for all $x \leq 7.8 \cdot 10^9$.

In 1985, Odlyzko and te Riele [34] showed that the Mertens conjecture was false. Their method used lattice basis reduction and arbitrary-precision arithmetic from the Brent MP package. The smallest counterexample is still unknown but it is known to be larger than 10^{14} and smaller than $\exp(1.59 \cdot 10^{40})$.

Alternating sign matrix conjecture. In the 1980s, Mills, Robbins, and Rumsey [33] studied alternating sign matrices—square $\{0, \pm 1\}$-matrices whose rows and columns sum to 1 and whose nonzero entries alternate sign in each row and column. They noticed that the number of alternating sign matrices of order $n \leq 10$ was $\prod_{k=0}^{n} (3k + 1)!/(n + k)!$ and conjectured that this relationship held for all n.

The conjecture was proven by Doron Zeilberger [43] in the 1990s, crucially relying on the combinatorial functions of the computer algebra system Maple. In fact, a Maple package was distributed with the paper that empirically (and in some cases rigorously) verified every nontrivial fact in the paper.

Kepler conjecture. In 1661, the astronomer and mathematician Johannes Kepler conjectured that the most efficient way of packing spheres in three dimensions is to stack them in a pyramid shape. It was still unsolved in 1900 and David Hilbert included it in his famous list of unsolved problems.

In 1998, the mathematician Thomas Hales and his student Samuel Ferguson [25] proved the Kepler conjecture using a variety of tools such as global optimization, linear programming, and interval arithmetic. Many of the computations in the proof were performed using Mathematica's arbitrary-precision arithmetic and double-checked using Maple. Because of the complexity of the calculations a team of at least thirteen referees could not be certain of the proof's correctness after four years. This lead Hales to start a project to complete a formal verification of the proof; it completed in 2014 after a decade of work [24].

3 SAT+CAS PARADIGM

As we saw in Section 2, the satisfiability checking and symbolic computation approaches have been applied to resolve a variety of mathematical conjectures—but each approach has its own advantages and disadvantages. On the one hand, satisfiability checking is good at solving problems with enormous search spaces and simple constraints. On the other hand, symbolic computation is good at solving problems with sophisticated mathematical calculations.

When a search space becomes too large the overhead associated with a computer algebra system becomes more pronounced, necessitating the usage of a more efficient solver. Currently, SAT solvers are probably the best tools currently available for general purpose search; they are very difficult to beat because of the decades of engineering effort that has been aimed at making them efficient.

Given this, Zulkoski, Ganesh, and Czarnecki in 2015 proposed [45] (and independently by Ábrahám [1]) the *SAT+CAS paradigm* of combining SAT solvers and CASs to solve conjectures that require both efficient search and advanced mathematics. In this section we overview and explain the major successes of the SAT+CAS paradigm over the last four years.

3.1 Williamson conjecture

In 1944, the mathematician J. Williamson studied the Hadamard conjecture from combinatorial design theory. This conjecture says that square $\{\pm 1\}$-matrices with with pairwise orthogonal rows exist in all orders $4n$. He defined a new class of matrices now known as Williamson matrices that he used to construct Hadamard matrices of order $4n$ for certain small values of n. Symmetric $\{\pm 1\}$-matrices A, B, C, D form a *set of Williamson matrices* (each individual matrix itself being *Williamson*) if they are circulant (each row is a cyclic shift of the previous row) and if $A^2 + B^2 + C^2 + D^2$ is the scalar matrix $4nI$. It was once considered likely that Williamson matrices exist for all n and therefore Williamson matrices could provide a route to proving the Hadamard conjecture [22]. The conjecture that Williamson matrices exist in all orders n has since become known as the Williamson conjecture.

The hopes that Williamson matrices exist in all orders were dashed in 1993, when D. Ž. Đoković [14] showed that Williamson matrices of order 35 do not exist by an exhaustive computer search. Đoković noted that this was the smallest *odd* counterexample of the Williamson conjecture but did not specify if it was truly the smallest counterexample. In 2006, Kotsireas and Koukouvinos [30] found no counterexamples in the even orders $n \leq 22$ using the CodeGeneration package of the computer algebra system Maple. In 2016, using an off-the-shelf SAT solver, Bright et al. [5] found no counterexamples in the even orders $n \leq 30$. Despite these successes, both the SAT-only and CAS-only approaches failed to find the smallest counterexample of the Williamson conjecture.

Not only did the SAT+CAS approach successfully find the smallest counterexample, it blew the other approaches out of the water by exhaustively solving all even orders up to seventy [6, 7]. The search space up to order 70 is an astronomical *twenty-five orders of magnitude* larger than the search space up to order 30 because the search space for Williamson matrices grows exponentially in n. Williamson matrices were found to exist in all even orders $n \leq 70$, leading to the *even Williamson* conjecture that Williamson matrices exist in all even orders.

The SAT+CAS approach is able to search such large spaces by exploiting mathematical properties of Williamson matrices that dramatically shrink the search space. In particular, the most important known filtering property is the *power spectral density (PSD) criterion* that says that if A is a Williamson matrix of order n with first row $[a_0, \dots, a_{n-1}]$ then

$$\mathrm{PSD}_A(k) := \left| \sum_{j=0}^{n-1} a_j e^{2\pi ijk/n} \right|^2 \leq 4n$$

for all integers k. This is an extremely strong filtering condition; a random circulant and symmetric $\{\pm 1\}$-matrix A will almost certainty fail it. Thus, a solver that is able to effectively exploit the PSD criterion will easily outperform a solver that does not know about this property. However, to effectively use it we need

(1) an efficient method of computing the PSD values; and
(2) an efficient method of searching while avoiding matrices that fail the filtering criteria.

The fundamental reason for the success of the SAT+CAS paradigm in regard to the Williamson and even Williamson conjectures is that CASs excel at (1) and SAT solvers excel at (2).

The manner in which the SAT and CAS are combined is demonstrated in Figure 1. As the SAT solver completes its search it sends to a CAS the matrices A, B, C, D from partial solutions of the SAT instance. The CAS then ensures that the matrices pass the PSD criterion. If a matrix fails the PSD criterion then a *conflict clause* is generated encoding that fact. The SAT solver adds the conflict clause into its learned clause database, thereby blocking the matrix from being considered in the future.

The search was also parallelized by splitting the search space into many independent subspaces. Each subspace had a separate SAT instance generated for it and the SAT instances were solved in parallel. The CAS was also useful in the splitting phase by removing instances that were found to be equivalent to other instances under the known equivalence operations of Williamson matrices.

In the end, our SAT+CAS system MathCheck found over 100,000 new sets of Williamson matrices among all even orders $n \leq 70$, a new set of Williamson matrices in the odd order 63, and verified that $n = 35$ is the smallest counterexample of the Williamson conjecture.

3.2 Good and best matrix conjectures

Many variants of Williamson matrices exist; two variants are known as *good matrices* (introduced by J. Seberry Wallis [41]) and *best matrices* (introduced by Georgiou, Koukouvinos, and Seberry [20]). There are several slightly different definitions for such matrices, but for our purposes we define them to be circulant matrices A, B, C, $D \in \{\pm 1\}^{n \times n}$ that satisfy $AA^T + BB^T + CC^T + DD^T = 4nI$ where A is skew ($A + A^T = 2I$) and D is symmetric ($D = D^T$). Additionally, B and C are skew (for best matrices) or symmetric (for good matrices).

It is known that if good matrices exist of order n exist then n must be of the form $2r + 1$ (i.e., odd) and if best matrices of order n exist then n must be of the form $r^2 + r + 1$. The good and best matrix conjectures state that good and best matrices exist in *all* orders of these forms. In 2002, the good matrix conjecture was shown to hold for all $n \leq 39$ [21] and in 2001 the best matrix conjecture was shown to hold for all $n \leq 31$ [20]. In 2018, the best matrix conjecture was shown to hold for all $n \leq 43$ and the counterexamples $n = 41$, 47, and 49 were found to the good matrix conjecture [15].

MathCheck has also been applied to the good and best matrix conjectures [3, 4] using a similar method as described in Section 3.1 with some encoding adjustments that are specific to good or best matrices. For example, if $[d_0, \dots, d_{n-1}]$ is the first row of a symmetric best matrix then it is known that $d_{n/3} = d_0$ when n is a multiple of 3. MathCheck found two new sets of good matrices (for $n = 27$ and 57) and three new counterexamples of the good matrix conjecture ($n = 51$, 63, and 69). MathCheck also found three new sets of best matrices in order 57 and showed that the best matrix conjecture holds for all $n \leq 57$ (the best currently known result).

3.3 Craigen–Holzmann–Kharaghani conjecture

In 2002, Craigen, Holzmann, and Kharaghani [11] studied *complex Golay pairs* which are polynomials f, g with $\{\pm 1, \pm i\}$ coefficients such that $|f(z)|^2 + |g(z)|^2$ is constant on the unit circle. This implies that f and g have the same number of terms and this quantity is known as the *length* of the polynomial. Craigen, Holzmann, and Kharaghani performed an exhaustive search for all complex Golay pairs up to length 19 and a partial search up to length 23. They found no complex Golay pairs of length 23 and conjectured that they did not exist. An exhaustive search was performed by F. Fiedler in 2013 [18] that did not find any complex Golay pairs of length 23, though no implementation was provided making it difficult to verify his search.

MathCheck can be used to independently verify the results of Fiedler's searches [8, 9]. The first step is to find all single polynomials f that could appear as a member of a complex Golay pair. A number of known properties of complex Golay pairs are used to cut down the search space, the most important one being that $|f(z)|^2 \leq 2n$ where n is the length of f and z is on the unit circle.

Given a potential f we solve the nonlinear optimization problem of maximizing $|f(z)|^2$ subject to $|z| = 1$ (see Maple's command NLPSolve) and discard the f whose maximum is greater than $2n$. Secondly, we use the known fact that if (f, g) is a complex Golay pair then $N_g(s) = -N_f(s)$ for $s = 1, \dots, n-1$ where N_g is the nonperiodic autocorrelation function of g.

Once f is known and enough of g is known so that $N_g(s) \neq -N_f(s)$ can be determined then a conflict clause is learned blocking the partial solution from ever being tried again. This filtering theorem is very powerful because it often works when only a few coefficients of g are known. For example, the SAT solver is able to learn to never assign both the first and last entries of g to be 1 at the same time.

3.4 Ruskey–Savage conjecture

In 1993, Ruskey and Savage [35] asked if every matching (a set of edges without common vertices) of the hypercube graph with 2^n vertices can be extended into a Hamiltonian cycle of the graph. In 2007, Fink [19] noted that this property holds in the hypercube graphs for $n = 2$, 3, and 4 and he proved a weaker form of the conjecture that he attributes to Kreweras [31].

In 2015, MathCheck was used to show for the first time that the Ruskey–Savage conjecture held for the hypercube graph with $2^5 = 32$ vertices [45]. This was accomplished by using a SAT solver to exhaustively enumerate the matchings of the hypercube graph and then verifying with a CAS that each matching extends to a Hamiltonian cycle. Certain kinds of matchings could be ignored; for example, the SAT solver only enumerates maximal matchings (those which cannot be increased in size while remaining matchings) because if a maximal matching extends to a Hamiltonian cycle then so do all subsets of the matching.

Once the CAS verifies that a given matching extends to a Hamiltonian cycle, a conflict clause is learned that blocks that Hamiltonian cycle (and all subsets of it) from being considered in the search again. Furthermore, it is also effective to have the CAS apply automorphisms of the hypercube graph to the Hamiltonian cycle it finds to generate additional Hamiltonian cycles to be blocked [44].

3.5 Norine conjecture

Consider a 2-colouring of the edges of a hypercube graph such that edges directly opposite each other have opposite colours. Serguei Norine conjectured that in such a colouring it is always possible to find two directly opposite vertices that are joined by a path of edges of a single colour [13]. In 2013, Feder and Subi reported that the conjecture had been verified for hypercube graphs with $n = 2$, 3, 4, and 5, and proved the conjecture for a special class of edge colourings [17].

In 2015, MathCheck was used to show for the first time that the Norine conjecture held for the hypercube graph with $2^6 = 64$ vertices [45]. This was accomplished by using a SAT solver to exhaustively enumerate the edge colourings for which the conjecture was not already known to hold.

Once an edge colouring was found by the SAT solver it was passed to a CAS to verify that the colouring contains at least two directly opposite vertices that are connected by a path of a single colour. If such vertices do not exist then this colouring forms a counterexample to the conjecture; otherwise, a conflict clause is generated that blocks this colouring from appearing in the search again. In fact, any colouring that includes the monochromatic path that was found by the CAS can be blocked, since all such colourings cannot be counterexamples to the Norine conjecture. Similar to in our work on the Ruskey–Savage conjecture, it is also effective to have the CAS apply automorphisms of the hypercube graph to the path that it finds to generate additional colourings to be blocked [44].

3.6 3 by 3 matrix multiplication

The classical way of multiplying two 2×2 matrices uses eight scalar multiplications; in 1969, Strassen discovered a way to do it using just seven scalar multiplications [37]. Two years later, Winograd showed that it is not possible to do it with six multiplications [42] and de Groot [12] showed there is essentially one optimal algorithm.

The optimal algorithm for multiplying 3×3 matrices is still unknown and the best known algorithm uses 23 multiplications [32]. Previously, four inequivalent algorithms were known with this complexity. Recently, Heule, Kauers, and Seidl [27] found over 13,000 additional inequivalent algorithms that use 23 multiplications. This was achieved using the SAT+CAS paradigm in a multistage process.

In the first stage, they reduce the problem of finding a matrix multiplication algorithm using 23 scalar multiplications to solving $3^6 = 729$ cubic equations in $23 \cdot 3^3 = 621$ variables. A SAT instance is generated from these equations by reducing them modulo 2. A solution of the SAT instance then provides a way to multiply 3×3 matrices over the finite field $F_2 = \{0, 1\}$.

By using various simplifications they found over 270,000 solutions of the SAT instance. They then used the computer algebra system Mathematica to determine that over 13,000 of those solutions are inequivalent. Finally, they use a Gröbner basis calculation in the computer algebra system Singular to lift the solutions found for the field F_2 to an arbitrary ring. They report that a small number of solutions over F_2 cannot be lifted in such a way but in most cases each solution provides a new 3×3 matrix multiplication algorithm that works in any ring. None of the algorithms they found could be simplified to use only 22 multiplications making it tempting to conjecture that such an algorithm does not exist.

4 CONCLUSION

In this article we have surveyed the SAT+CAS paradigm of combining SAT solvers and computer algebra systems aimed at resolving mathematical conjectures. It is illuminating to contrast the kind of problems that have been solved by the SAT and CAS paradigms individually, with those that have been solved by the combined SAT+CAS paradigm.

We discussed three long-standing mathematical problems in Section 2.1 for which SAT solvers have been used. For each problem, attempts to use custom-purpose search code or optimization methods ultimately proved to not be as successful as using a SAT solver. This is due to the many efficient search heuristics that have been incorporated in modern solvers, as well as the years of refinements that have gone into these solvers. These heuristics have broad applicability for problems from diverse domains.

Additionally, we saw three long-standing conjectures in Section 2.2 that CAS methods were used to resolve. In each case, very efficient mathematical calculations were necessary but efficient search routines were not the bottleneck in the solutions. These conjectures would not be a good fit for SAT solvers because these problems do not admit natural encodings into Boolean logic.

Note that the eight conjectures from Section 3 would be difficult to resolve using *either* SAT solvers or CASs alone. In each case, the problems have both a significant search component (an exponentially growing search space) and a significant mathematical component (e.g., requiring knowledge of the power spectral density of a circulant matrix or the automorphism group of a graph). As we've seen, the SAT+CAS paradigm is effective at pushing the state-of-the-art in such conjectures. Simply put, the SAT+CAS paradigm allows the mathematician to solve problems that have search spaces too large for CASs and require mathematical calculations too sophisticated for SAT solvers.

REFERENCES

[1] E. Ábrahám. Building bridges between symbolic computation and satisfiability checking. In S. Linton, editor, *Proceedings of the 2015 ACM on International Symposium on Symbolic and Algebraic Computation*, pages 1–6. ACM, 2015.

[2] S. Arunachalam and I. Kotsireas. Hard satisfiable 3-SAT instances via autocorrelation. *Journal on Satisfiability, Boolean Modeling and Computation*, 10:11–22, 2016.

[3] C. Bright, D. Đoković, I. Kotsireas, and V. Ganesh. A SAT+CAS approach to finding good matrices: New examples and counterexamples. In *Thirty-Third AAAI Conference on Artificial Intelligence*, pages 1435–1442. AAAI Press, 2019.

[4] C. Bright, D. Đoković, I. Kotsireas, and V. Ganesh. The SAT+CAS method for combinatorial search with applications to best matrices. *Submitted*, 2019.

[5] C. Bright, V. Ganesh, A. Heinle, I. Kotsireas, S. Nejati, and K. Czarnecki. MathCheck2: A SAT+CAS verifier for combinatorial conjectures. In V. P. Gerdt, W. Koepf, W. M. Seiler, and E. V. Vorozhtsov, editors, *International Workshop on Computer Algebra in Scientific Computing*, pages 117–133. Springer, 2016.

[6] C. Bright, I. Kotsireas, and V. Ganesh. A SAT+CAS method for enumerating Williamson matrices of even order. In S. A. McIlraith and K. Q. Weinberger, editors, *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 6573–6580. AAAI Press, 2018.

[7] C. Bright, I. Kotsireas, and V. Ganesh. Applying computer algebra systems with SAT solvers to the Williamson conjecture. *Journal of Symbolic Computation*, to appear, 2019.

[8] C. Bright, I. Kotsireas, A. Heinle, and V. Ganesh. Enumeration of complex Golay pairs via programmatic SAT. In C. Arreche, editor, *Proceedings of the 2018 ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2018, New York, NY, USA, July 16–19, 2018*, pages 111–118, 2018.

[9] C. Bright, I. Kotsireas, A. Heinle, and V. Ganesh. Complex Golay pairs up to length 28: A search via computer algebra and programmatic SAT. *Journal of Symbolic Computation*, to appear, 2019.

[10] J. Cooper and C. Poirel. Pythagorean partition-regularity and ordered triple systems with the sum property. arXiv 0809.3478, 2008.

[11] R. Craigen, W. Holzmann, and H. Kharaghani. Complex Golay sequences: structure and applications. *Discrete Mathematics*, 252(1-3):73–89, 2002.

[12] H. F. de Groote. On varieties of optimal algorithms for the computation of bilinear mappings II. Optimal algorithms for 2×2-matrix multiplication. *Theoretical Computer Science*, 7(2):127–148, 1978.

[13] M. DeVos. Edge-antipodal colorings of cubes. *Open Problem Garden*, 2008.

[14] D. Ž. Đoković. Williamson matrices of order $4n$ for $n = 33, 35, 39$. *Discrete Mathematics*, 115(1–3):267–271, 1993.

[15] D. Ž. Đoković and I. S. Kotsireas. Goethals–Seidel difference families with symmetric or skew base blocks. *Mathematics in Computer Science*, 12(4):373–388, 2018.

[16] G. Exoo. A lower bound for Schur numbers and multicolor Ramsey numbers of K_3. *Electronic Journal of Combinatorics*, 8:1, 1994.

[17] T. Feder and C. Subi. On hypercube labellings and antipodal monochromatic paths. *Discrete Applied Mathematics*, 161(10–11):1421–1426, 2013.

[18] F. Fiedler. Small Golay sequences. *Advances in Mathematics of Communications*, 7(4):379–407, 2013.

[19] J. Fink. Perfect matchings extend to Hamilton cycles in hypercubes. *Journal of Combinatorial Theory, Series B*, 97(6):1074–1076, 2007.

[20] S. Georgiou, C. Koukouvinos, and J. Seberry. On circulant best matrices and their applications. *Linear and Multilinear Algebra*, 48(3):263–274, 2001.

[21] S. Georgiou, C. Koukouvinos, and S. Stylianou. On good matrices, skew Hadamard matrices and optimal designs. *Computational Statistics & Data Analysis*, 41(1):171–184, 2002.

[22] S. W. Golomb and L. D. Baumert. The search for Hadamard matrices. *The American Mathematical Monthly*, 70(1):12–17, 1963.

[23] S. W. Golomb and L. D. Baumert. Backtrack programming. *Journal of the ACM*, 12(4):516–524, 1965.

[24] T. Hales, M. Adams, G. Bauer, T. D. Dang, J. Harrison, H. Le Truong, C. Kaliszyk, V. Magron, S. McLaughlin, T. T. Nguyen, et al. A formal proof of the Kepler conjecture. In *Forum of Mathematics, Pi*, volume 5. Cambridge University Press, 2017.

[25] T. C. Hales and S. P. Ferguson. *The Kepler Conjecture: The Hales-Ferguson Proof.* Springer New York, 2011.

[26] M. J. H. Heule. Schur number five. In S. A. McIlraith and K. Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 6598–6606. AAAI Press, 2018.

[27] M. J. H. Heule, M. Kauers, and M. Seidl. New ways to multiply 3×3-matrices. arXiv 1905.10192, 2019.

[28] M. J. H. Heule, O. Kullmann, and V. W. Marek. Solving very hard problems: Cube-and-conquer, a hybrid SAT solving method. In C. Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4864–4868. 2017.

[29] B. Konev and A. Lisitsa. Computer-aided proof of Erdős discrepancy properties. *Artificial Intelligence*, 224:103–118, 2015.

[30] I. S. Kotsireas and C. Koukouvinos. Constructions for Hadamard matrices of Williamson type. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 59:17–32, 2006.

[31] G. Kreweras. Matchings and Hamiltonian cycles on hypercubes. *Bulletin of the Institute of Combinatorics and its Applications*, 16:87–91, 1996.

[32] J. D. Laderman. A noncommutative algorithm for multiplying 3×3 matrices using 23 multiplications. *Bulletin of the American Mathematical Society*, 82(1):126–128, 1976.

[33] W. H. Mills, D. P. Robbins, and H. Rumsey Jr. Alternating sign matrices and descending plane partitions. *Journal of Combinatorial Theory, Series A*, 34(3):340–359, 1983.

[34] A. M. Odlyzko and H. J. J. te Riele. Disproof of the Mertens conjecture. *Journal für die reine und angewandte Mathematik*, 357:138–160, 1985.

[35] F. Ruskey and C. Savage. Hamilton cycles that extend transposition matchings in Cayley graphs of S_n. *SIAM Journal on Discrete Mathematics*, 6(1):152–166, 1993.

[36] I. Schur. Über die kongruenz $x^m + y^m = z^m \pmod{p}$. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 25:114–116, 1917.

[37] V. Strassen. Gaussian elimination is not optimal. *Numerische mathematik*, 13(4):354–356, 1969.

[38] T. Tao. The Erdős discrepancy problem. *Discrete Analysis*, 1, 2016.

[39] A. Van Gelder and I. Spence. Zero-one designs produce small hard SAT instances. In O. Strichman and S. Szeider, editors, *International Conference on Theory and Applications of Satisfiability Testing*, pages 388–397. Springer, 2010.

[40] M. Y. Vardi. Boolean satisfiability: Theory and engineering. *Communications of the ACM*, 57(3):5–5, 2014.

[41] J. S. Wallis. *Combinatorial matrices*. PhD thesis, La Trobe University, 1970.

[42] S. Winograd. On multiplication of 2×2 matrices. *Linear algebra and its applications*, 4(4):381–388, 1971.

[43] D. Zeilberger. Proof of the alternating sign matrix conjecture. *Electronic Journal of Combinatorics*, 3(2):R13, 1996.

[44] E. Zulkoski, C. Bright, A. Heinle, I. Kotsireas, K. Czarnecki, and V. Ganesh. Combining SAT solvers with computer algebra systems to verify combinatorial conjectures. *Journal of Automated Reasoning*, 58(3):313–339, 2017.

[45] E. Zulkoski, V. Ganesh, and K. Czarnecki. MathCheck: A math assistant via a combination of computer algebra systems and SAT solvers. In A. P. Felty and A. Middeldorp, editors, *CADE*, volume 9195 of *Lecture Notes in Computer Science*, pages 607–622. Springer, 2015.