A SAT+CAS Approach to Finding Good Matrices:

New Examples and Counterexamples

Curtis Bright	University of Waterloo
Dragomir Đoković	University of Waterloo
Ilias Kotsireas	Wilfrid Laurier University
Vijay Ganesh	University of Waterloo

SAT:

Boolean satisfiability problem

SAT:

Boolean satisfiability problem

SAT solvers: Glorified brute force

CAS:

Computer algebra system

CAS:

Computer algebra system

Mathematical expression manipulators

SAT + CAS

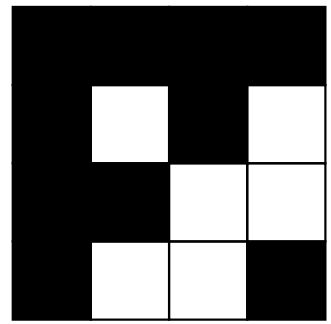
Brute force + Knowledge

The research areas of SMT [SAT Modulo Theories] solving and symbolic computation are quite disconnected. [. . .] More common projects would allow to join forces and commonly develop improvements on both sides.

Erika Ábrahám.
Building bridges between symbolic computation and satisfiability checking.
ISSAC invited talk, 2015.

Hadamard matrices

- 125 years ago Jacques Hadamard defined what are now known as *Hadamard matrices*.
- Square matrices with ± 1 entries and pairwise orthogonal rows.



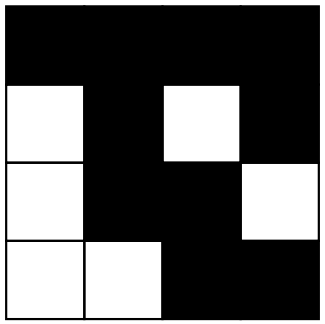
Jacques Hadamard. Résolution d'une question relative aux déterminants.
Bulletin des sciences mathématiques, 1893.

The Hadamard conjecture

- The *Hadamard conjecture* says that Hadamard matrices exist in order $4n$ for all positive integers n.
- Strongly expected to hold but still open after 125 years.

The skew Hadamard conjecture

- A matrix is *skew* if its diagonal entries are 1 and its entry at (i, j) is the negative of its entry at (j, i).
- The skew Hadamard conjecture says that skew Hadamard matrices exist in order $4n$ for all positive integers n.



Good matrices

In 1970, Jennifer Seberry Wallis discovered a way to construct skew Hadamard matrices of order $4n$ using four "good" matrices A, B, C, D of order n with ± 1 entries.

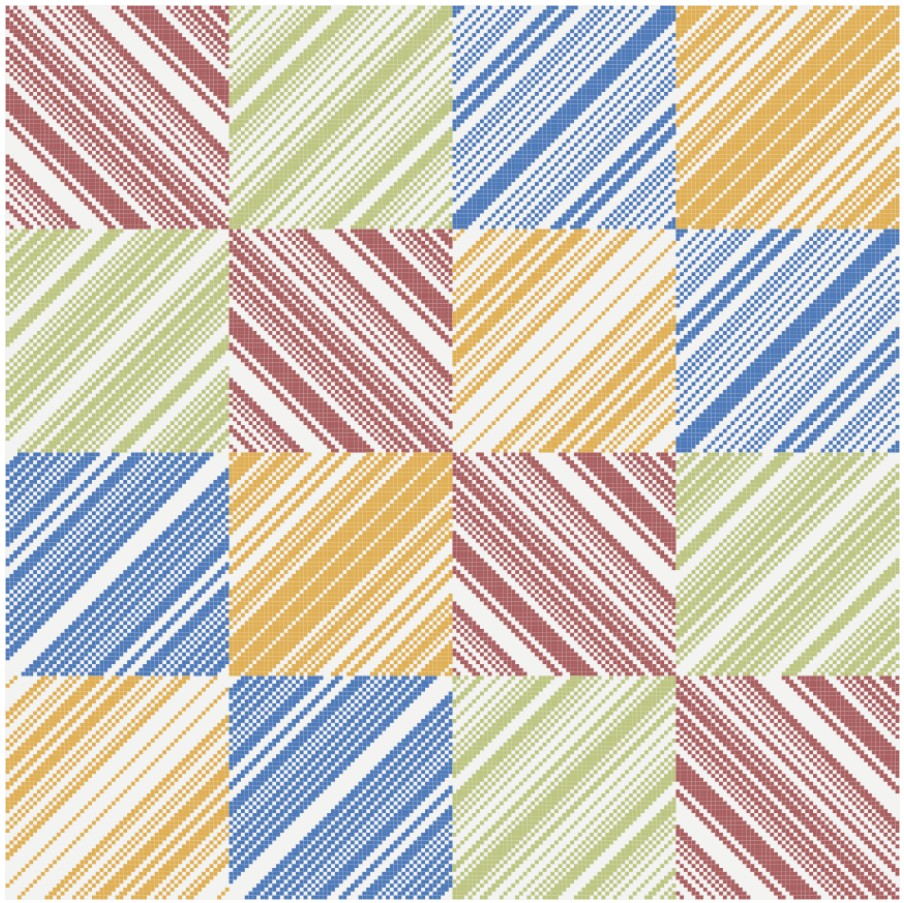
Good matrices

In 1970, Jennifer Seberry Wallis discovered a way to construct skew Hadamard matrices of order $4n$ using four "good" matrices A, B, C, D of order n with ± 1 entries.

Properties

- A is skew and B, C, D are symmetric.
- Every row is a shift of the previous row.
- $AA^T + B^2 + C^2 + D^2$ is the identity matrix scaled by $4n$.

A skew Hadamard matrix of order $4 \cdot 57 = 228$



Constructed using the good matrices A, B, C, D.

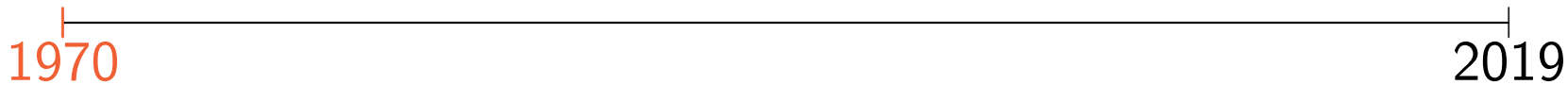
The good matrix conjecture

. . . it is conceivable that [good matrices] exist for all $n = 2m + 1$, $m \geq 1$ *and it is worth testing this hypothesis at least for those orders which are accessible to present day computers. . .*

George Szekeres.
A note on skew type orthogonal ± 1 matrices.
Combinatorics, Colloquia Mathematica Societatis János Bolyai, 1988.

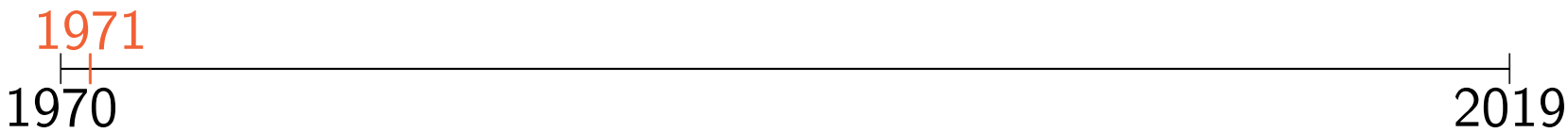
Known good matrices

In 1970, Seberry found good matrices in the orders 3, 5, 7, 9, 11, 13, 15, and 19.



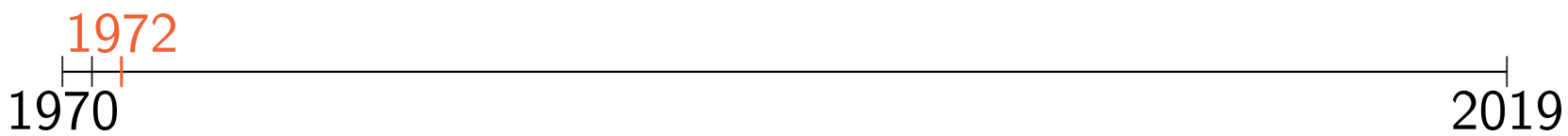
Known good matrices

In 1971, Seberry found a set of good matrices in order 23.



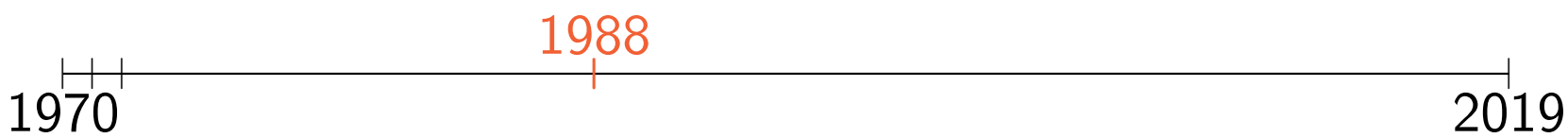
Known good matrices

In 1972, Hunt found new good matrices in the orders 7, 11, 13, 15, 17, 19, 21 (via a complete search) and order 25.



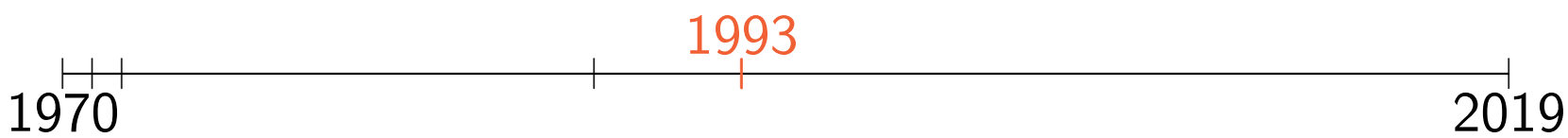
Known good matrices

In 1988, Szekeres found new good matrices in the orders 23, 25, 27, 29, and 31 (via a complete search).



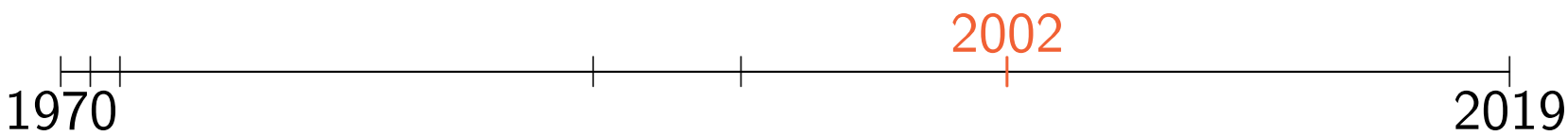
Known good matrices

In 1993, Đoković found new good matrices in the orders 33, 35, and 127.



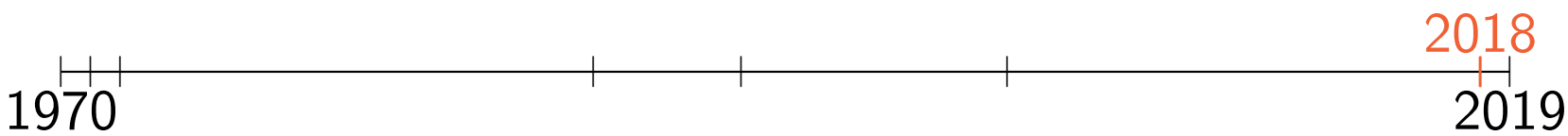
Known good matrices

In 2002, Georgiou, Koukouvinos, and Stylianou found new good matrices in the orders 33, 35, 37, and 39 (via a complete search) showing that the good matrix conjecture holds for $n < 40$.



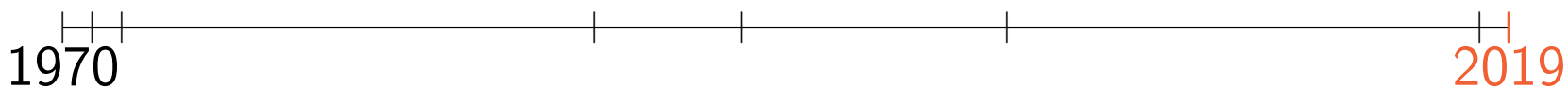
Known good matrices

In 2018, Đoković and Kotsireas found new good matrices in the orders 43 and 45 (via a complete search) and found that 41, 47, and 49 are counterexamples to the good matrix conjecture.

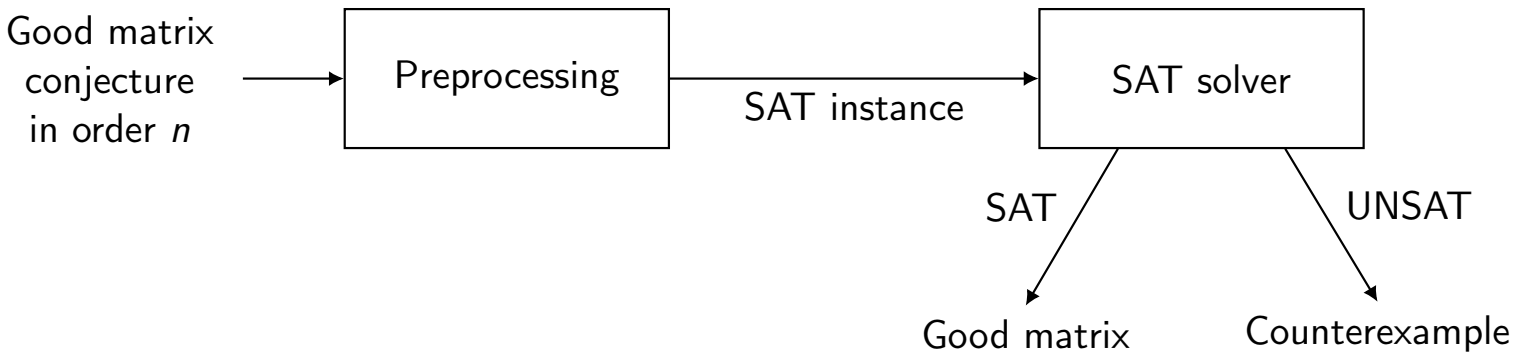


Known good matrices

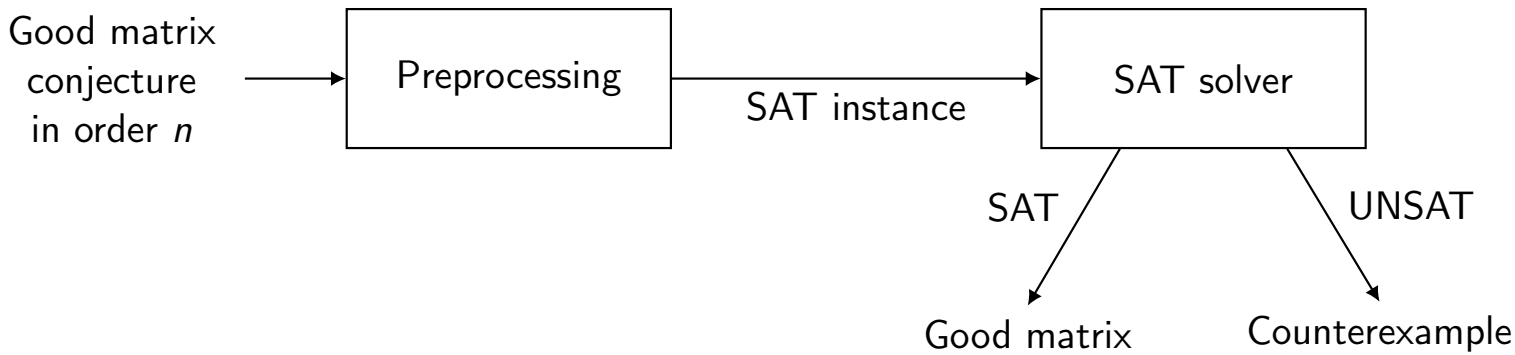
In our paper we find new good matrices in the orders 27 and 57 (via a complete search) and found that 51, 63, and 69 are counterexamples to the good matrix conjecture.



System overview

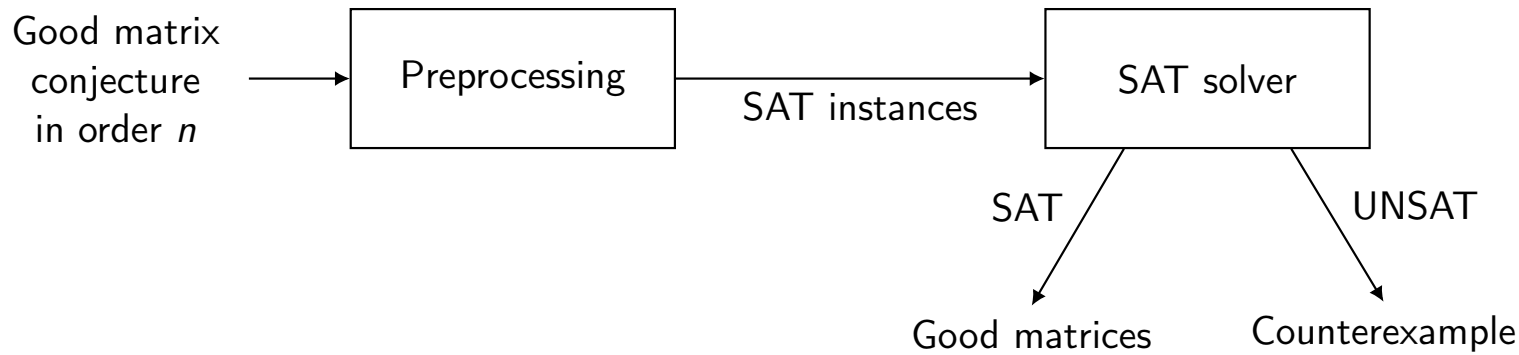


System overview



This setup is simple but only works for small n.

System overview



Split up the search space during preprocessing:

Solvers perform better on smaller search spaces and the subspaces are independent so can be solved in parallel.

Splitting

The simplest thing would be to fix the first entries of A, but this does not perform well.

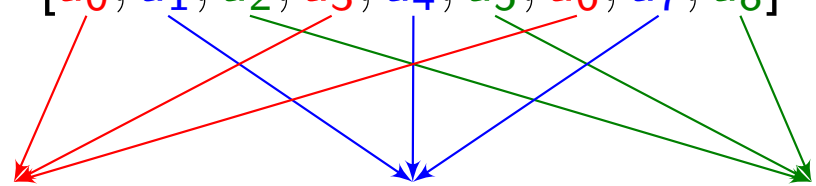
Splitting

The simplest thing would be to fix the first entries of A, but this does not perform well.

Compression

- Instead, we fix the entries of the *compression* of A.
- Compression of a row of order n is defined as follows:

$$A = [a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8]$$



$$A' = \left[a_0 + a_3 + a_6, \quad a_1 + a_4 + a_7, \quad a_2 + a_5 + a_8\right].$$

Uncompression

Let the Boolean variables $a_0, \ldots, a_{n-1}$ represent the entries of A with true representing 1 and false representing -1.

Encoding in SAT

- Say the first entry in the 3-compression of A is 3, i.e.,

$$a_0 + a_{n/3} + a_{2n/3} = 3.$$

- We encode this in Boolean logic as the three unit clauses

$$a_0, \quad a_{n/3}, \quad a_{2n/3}.$$

Encoding in SAT

- Say the first entry in the 3-compression of A is 1, i.e.,

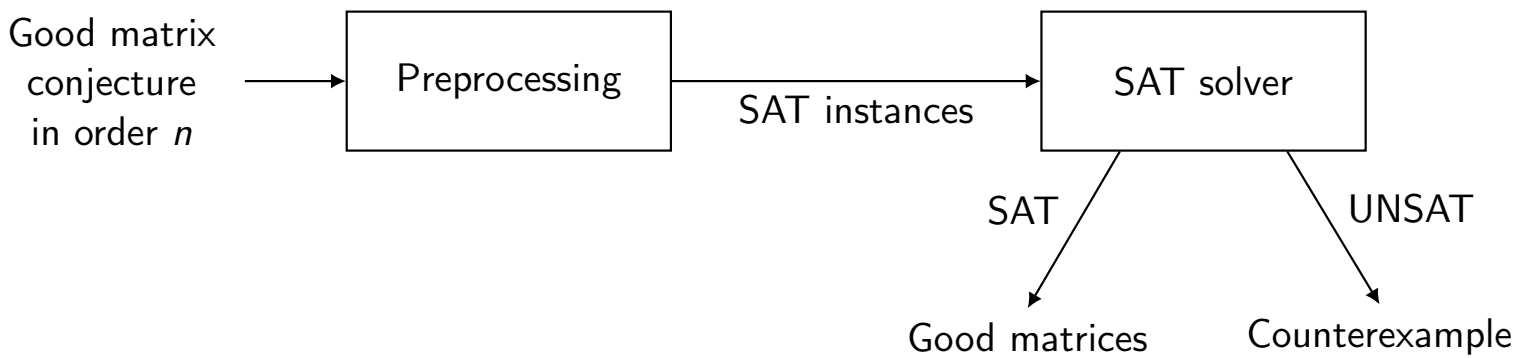
$$a_0 + a_{n/3} + a_{2n/3} = 1.$$

- We encode this in Boolean logic as the four clauses

$$\neg a_0 \vee \neg a_{n/3} \vee \neg a_{2n/3},$$

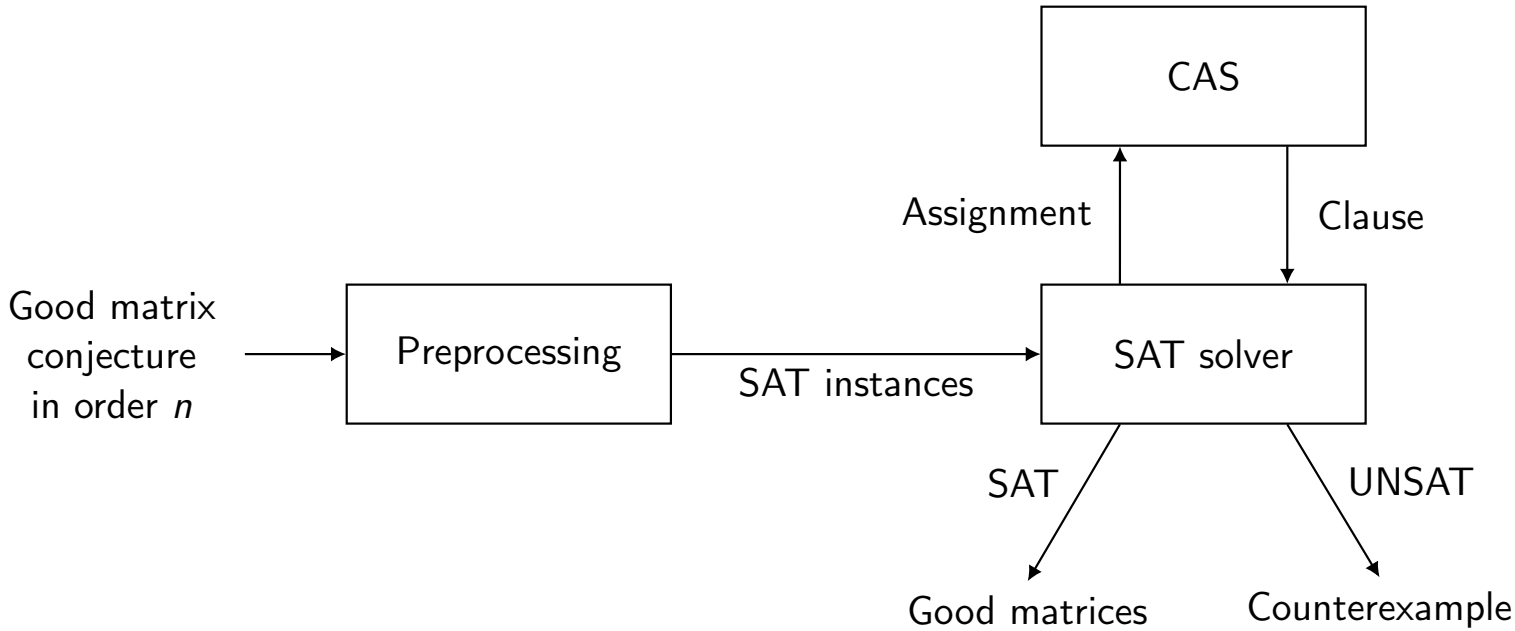
$$a_0 \vee a_{n/3}, \quad a_0 \vee a_{2n/3}, \quad a_{n/3} \vee a_{2n/3}.$$

System overview



This works better but does not exploit theorems about good matrices that cannot easily be encoded in Boolean logic.

System overview



Encode some knowledge *programmatically*:

Allows encoding much more expressive constraints.

Power spectral density

- The *power spectral density* $\mathrm{PSD}_A(k)$ of $A = [a_0, \ldots, a_{n-1}]$ is the value

$$\left| \sum_{j=0}^{n-1} a_j \omega^{jk} \right|^2$$

  where $\omega := \exp(2\pi i/n)$.
- Can be computed very efficiently by CAS functions.

Power spectral density

- The *power spectral density* $\mathrm{PSD}_A(k)$ of $A = [a_0, \ldots, a_{n-1}]$ is the value

$$\left| \sum_{j=0}^{n-1} a_j \omega^{jk} \right|^2$$

  where $\omega := \exp(2\pi i/n)$.
- Can be computed very efficiently by CAS functions (but *not* SAT solvers)!

PSD filtering

If a sequence has a PSD value larger than $4n$ then **it cannot be a row of a good matrix**.

Example

- Let $n = 2m + 1$.
- Say the SAT solver assigns the first $m + 1$ entries of A to 1 (true) and the last m entries of A to -1 (false).

Example

- Let $n = 2m + 1$.
- Say the SAT solver assigns the first $m + 1$ entries of A to 1 (true) and the last m entries of A to -1 (false).
- In this case we can compute that $\mathrm{PSD}_A(1) \approx 0.4n^2$ which is larger than $4n$ for large n.

Example

- Let $n = 2m + 1$.
- Say the SAT solver assigns the first $m + 1$ entries of A to 1 (true) and the last m entries of A to -1 (false).
- In this case we can compute that $\mathrm{PSD}_A(1) \approx 0.4n^2$ which is larger than $4n$ for large n.

Consequence

A cannot be a row of a good matrix, so the SAT solver learns the clause blocking A:

$$\neg a_0 \vee \cdots \vee \neg a_m \vee a_{m+1} \vee \cdots \vee a_{n-1}$$

Filtering results

- A simple filtering approach would require knowing all values of A, B, C, and D and blocking clauses would be of length $4n$.
- The programmatic PSD filtering approach was hugely successful, usually allowing the SAT solver to learn a blocking clause just of size n.

Filtering results

- A simple filtering approach would require knowing all values of A, B, C, and D and blocking clauses would be of length $4n$.
- The programmatic PSD filtering approach was hugely successful, usually allowing the SAT solver to learn a blocking clause just of size n.
- The programmatic approach was over 10 times faster in order 33 and the speedup looked exponential in n.

Enumeration results

- Two new sets of good matrices: One of order 27 (missed by Szekeres' search) and one of order 57.
- Three new counterexamples: No good matrices exist in the orders 51, 63, and 69. (Independent verification requested!)
- Code available from the MathCheck website:

`uwaterloo.ca/mathcheck`

Conclusion

- The SAT+CAS paradigm is very general and can be applied to problems in many domains, especially "needle-in-haystack" problems that require rich mathematics.

Conclusion

- The SAT+CAS paradigm is very general and can be applied to problems in many domains, especially "needle-in-haystack" problems that require rich mathematics.
- Make use of the immense amount of engineering effort that has gone into CAS and SAT solvers.

Conclusion

- The SAT+CAS paradigm is very general and can be applied to problems in many domains, especially "needle-in-haystack" problems that require rich mathematics.
- Make use of the immense amount of engineering effort that has gone into CAS and SAT solvers.
- Splitting up the problem in a way that takes advantage of this requires domain knowledge.