SEARCHING FOR COMPLEX GOLAY SEQUENCES USING A SAT SOLVER

CURTIS BRIGHT, VIJAY GANESH, ALBERT HEINLE, JIA HUI LIANG

University of Waterloo, Ontario, Canada

ILIAS KOTSIREAS

Wilfrid Laurier University, Ontario, Canada

Abstract. We propose a research project to complete the search for complex Golay sequences of order 23. Our method employs a novel approach which uses the techniques and domain knowledge found in computer algebra systems coupled with the sophisticated searching abilities of modern SAT solvers.

Golay introduced the sequences which were later named in his honor in 1949. Since then many types of so-called *complementary sequences* have been studied, both for their elegant theoretical properties and for their applications in diverse fields such as spectrometry, communications, and coding theory. One class of complementary sequences known as *complex Golay sequences* were defined in 1992. Two sequences A and B of length n are said to be a complex Golay pair of order n if their entries are from the set $\{\pm 1, \pm i\}$ and the sequences satisfy

$$\sum_{j=1}^{k} A_j \overline{A}_{n-k+j} + \sum_{j=1}^{k} B_j \overline{B}_{n-k+j} = 0 \quad \text{for} \quad k = 1, \ldots, n-1.^{1}$$

In 2001 a search was performed for complex Golay sequences for all orders up to 23 and this search uncovered many previously unknown complex Golay sequences. However, no complex Golay sequences of order 23 were found and it was conjectured that no such sequences exist. The conjecture remains open today and we propose to resolve it by either finding such a sequence or searching the remaining space and thereby proving that no complex Golay sequence of order 23 exists.

We have developed a methodology of searching for complementary sequences which uses the techniques and domain knowledge found in computer algebra systems coupled with the sophisticated searching abilities of modern SAT solvers. To date, we have used these methods to construct many previously unknown Hadamard matrices and prove that there are no Williamson matrices of order 35. Furthermore, our results showed for the first time that 35 is the smallest number with this property.

Date: August 14, 2016.

[1]Here $\overline{A}$ represents the complex conjugate of A.

(Our results and code may be found on our website `https://sites.google.com/site/uwmathcheck/`.)

SAT solvers are highly efficient at searching domains specified by boolean constraints but lack the high-level domain-specific knowledge which can be used to remove parts of the search space which provably do not contain a solution. Our method uses a custom programmatic SAT solver in which domain knowledge can be embedded to automatically perform a filtering process as the search is progressing; this allows the search to complete significantly faster than one only using a naive encoding.

Once the space of complex Golay sequences of order 23 has been completely searched we plan to submit our result to a mathematics or computer science journal such as *Mathematics in Computer Science*; our code will also be made available on our website. In addition to resolving a conjecture which has been open for over 15 years, the method by which we propose to do it should be of significant interest to both the symbolic computation and satisfiability checking communities. Proposals for bridging these two areas have recently received considerable interest as demonstrated by initiatives such as the SC^2 project (`http://www.sc-square.org/`).

We estimate that our methods can resolve the order 23 complex Golay conjecture using approximately 500,000 hours of compute time. Our system translates the conjecture into a SAT instance which has a solution if and only if a complex Golay sequence of order 23 exists and then uses domain-specific knowledge to split this SAT instance into millions of smaller subinstances. Our work is therefore highly parallelizable and sees an essentially linear speedup in the number of processors available. For example, with 1000 processors the search should complete in under a month. Our work does not require a significant amount of bandwidth or storage (at most a few gigabytes for storing log files).

Currently we are using a slightly modified version of the SAT solver MAPLE-COMSPS for our work. At present we are compiling and running this on Linux systems with Intel Xeon 3.3GHz processors although the code is portable and should compile and run on a variety of operating systems and architectures.

The lead and principal investigator for the project is professor Vijay Ganesh. The implementation work will be completed by PhD students Curtis Bright, Albert Heinle, and Jia Hui Liang, under the guidance of professor Ilias Kotsireas.