Integer and Constraint Programming Revisited for Mutually Orthogonal Latin Squares (Student Abstract)

Noah Rubin,[1] Curtis Bright,[2,1] Brett Stevens,[1] Kevin Cheung[1]

[1]Carleton University, School of Mathematics and Statistics
[2]University of Windsor, School of Computer Science
noahrubin@cmail.carleton.ca, cbright@uwindsor.ca, brett@math.carleton.ca, kevincheung@cunet.carleton.ca

Abstract

We use integer programming (IP) and constraint programming (CP) to search for sets of mutually orthogonal latin squares (MOLS). We improve the performance of the solvers by formulating an extended symmetry breaking method and provide an alternative CP encoding which performs much better in practice. Using state-of-the-art solvers we are able to quickly find pairs of MOLS (or prove their nonexistence) in all orders up to and including eleven. We also analyze the effectiveness of using CP and IP solvers to search for triples of MOLS and estimate the running time of using this approach to resolve the longstanding open problem of determining the existence of a triple of MOLS of order ten.

Introduction

A *latin square of order* n is an $n \times n$ array, L, of symbols $\{0, 1, \ldots, n-1\}$ in which each symbol appears exactly once in each row and column. The entry in row i and column j of a square L is denoted L_{ij}. Two latin squares of the same order, L and M, are said to be *orthogonal* if there is a unique solution $L_{ij} = a$, $M_{ij} = b$ for every pair of $a, b \in \{0, 1, \ldots, n-1\}$. A set of k latin squares of order n, is called a set of *mutually orthogonal latin squares* (MOLS) if all squares are pairwise orthogonal—in which case we label the system as $k\,\mathrm{MOLS}(n)$.

There is a long history of using automated reasoning tools like constraint and integer programming solvers to search for and construct latin squares (Gomes 2000). In this project we use off-the-shelf constraint programming (CP) and integer programming (IP) solvers to find mutually orthogonal latin squares (or disprove their existence) in all orders up to and including eleven. We also develop two main improvements to the IP and CP models originally described by Appa *et al.* (Appa, Magos, and Mourtos 2006). First, we improve the constraint programming model by using indexing constraints to encode orthgogonality, which performed significantly better than the standard constraint programming encoding—using a CP solver we were able to find or prove the nonexistence of $2\,\mathrm{MOLS}(n)$ for $n \leq 11$ with our improved encoding compared to $n \leq 8$ with the usual encoding. Our improved constraint programming formulation is described below. Table 1 compares the running times across all three models encoding the $2\,\mathrm{MOLS}(n)$ problem, and shows the superiority of this encoding. Second, we developed an improved symmetry breaking method (described below) that removes more symmetry from the search space than the "domain reduction" symmetry breaking method used in previous searches (Appa, Magos, and Mourtos 2006; Appa, Mourtos, and Magos 2002). Our new symmetry breaking method reduces the amount of symmetry present in the $k\,\mathrm{MOLS}(n)$ search by an exponential factor in n when compared with domain reduction symmetry breaking. The new symmetry breaking method performs well in practice, particularly with the integer programming model—using an IP solver we were able to find or prove the nonexistence of $2\,\mathrm{MOLS}(n)$ for $n \leq 10$ with our improved symmetry breaking compared to $n \leq 8$ with domain reduction symmetry breaking.

Constraint and Integer Programming Models

Let X and Y be two mutually orthogonal latin squares of order n with entries X_{ij} and Y_{ij} for $0 \leq i, j < n$. One method of approaching the $2\,\mathrm{MOLS}(n)$ problem is to express it as a pure binary linear program and use an integer programming (IP) solver to generate solutions (Dantzig 1963, §26.3.IV). Our IP model for a set of two mutually orthogonal latin squares contains the n^4 binary variables

$$x_{ijkl} \coloneqq \begin{cases} 1 & \text{if } X_{ij} = k \text{ and } Y_{ij} = l \\ 0 & \text{otherwise} \end{cases}$$

for all $i, j, k, l \in \{0, 1, \ldots, n-1\}$. The latin and orthogonality constraints are expressed as six sets of n^2 equalities grouped by which subscripts of x_{ijkl} are fixed. A CP solver allows a more natural formulation of the $2\,\mathrm{MOLS}(n)$ problem using the $2n^2$ integer-valued variables

$$X_{ij} \coloneqq \text{value of cell } (i, j) \text{ in square } X,$$
$$Y_{ij} \coloneqq \text{value of cell } (i, j) \text{ in square } Y,$$

where $i, j, X_{ij}, Y_{ij} \in \{0, 1, \ldots, n-1\}$. The squares X and Y can be forced to be latin squares via "AllDifferent" constraints which specify that the rows and columns of the squares each contain different values. Appa *et al.* provide an encoding of orthogonality by defining $Z_{ij} \coloneqq X_{ij} + nY_{ij}$ and imposing AllDifferent$(Z_{ij}\ \forall i, j)$; we call this the CP-linear encoding.

Viewing the rows of X and Y as permutations on $\{0, 1, \ldots, n-1\}$ we define XY as the square whose ith row is row i of Y applied to row i of X. We also define X^{-1} as the square whose ith row is the inverse permutation of row i of X. Our alternative orthogonality encoding is based on (Laywine and Mullen 1998, Theorem 6.6) which implies two latin squares X and Y are orthogonal if and only if there is a latin square Z such that $XZ = Y$. The additional variables in our alternative orthogonality encoding are

$$Z_{ij} \coloneqq \text{value of cell } (i, j) \text{ in square } Z = X^{-1}Y,$$

where $Z_{ij} \in \{0, 1, \ldots, n-1\}$. In order to ensure $Y = XZ$ the (i, j)th entry of Y is set equal to the (i, X_{ij})th entry of Z using the "element indexing" constraint $Z_i[X_{ij}] = Y_{ij}$ where Z_i is the vector of variables corresponding to row i of Z. The constraints encoding that the squares X and Y are orthogonal are then AllDifferent$(Z_{ij}\ \forall j)$ and AllDifferent$(Z_{ij}\ \forall i)$. We call this the CP-index encoding and altogether it uses $6n$ AllDifferent constraints and n^2 element indexing constraints. This encoding also may be extended to encode the $3\,\mathrm{MOLS}(n)$ problem by adding another n^2 variables encoding the entries of a third latin square U and using the same orthogonality constraints as above between X and U and Y and U.

Symmetry Breaking

There are a large number of symmetries in the $k\,\mathrm{MOLS}(n)$ problems. In any set of $k\,\mathrm{MOLS}(n)$, the rows and columns of squares can be permuted simultaneously. The symbol sets within each square can be permuted independently and the squares may all be replaced with their transposes. All of these operations preserve the latin properties and orthogonality of the set of squares (Colbourn and Dinitz 2006). The search space can be reduced significantly for any elimination of symmetries that still permits finding an isomorphic representative of any solution. Appa *et al.* fix the first row of every square to be in lexicographic order which eliminates the permutations of the columns and fixes the symbol permutations to be the same in each square. They fix the first column of the first square to be in lexicographic order which eliminates permutations of the rows. This reduces the possible symmetries to $2k!n!$. With these cells fixed, the first column of Y must be a permutation where $Y_{i0} \neq i$ for $1 \leq i < n$.

Appa *et al.* (Appa, Mourtos, and Magos 2002) use domain reduction to further reduce the number of possibilities. We show that the number of possibilities for the first column of Y is reduced to F_{n-2}, the $(n-2)$th Fibonacci number. By exploiting the disjoint cycle structure of these permutations, we are able to further reduce this to the number of partitions of $n-1$ into parts of size greater than 1. This quantity is $e^{O(\sqrt{n})}$ and is exponentially smaller than F_{n-2} (OEIS Foundation Inc. 2021).

In detail, let X and Y be a pair of orthogonal latin squares in *standard form*: the first rows of both are in lexicographic order and the first column of X is in lexicographic order. The *first column permutation of* Y is $\rho(i) = Y_{i0}$. We prove that any pair of orthogonal latin squares is isomorphic to a pair (X, Y) in standard form where ρ's disjoint cycles, when listed in lexicographic order, are from shortest to longest. This fact applies equally well to a set of $k\,\mathrm{MOLS}(n)$, and implies that we can restrict the first column of the second square to be in a fixed set of canonical representatives (one from each cycle type). Figure 1 contains a plot comparison of the running times observed by using each of our available symmetry breaking methods. More detailed timings, proofs and complete implementations of our programs can be found at https://github.com/noahrubin333/CP-IP.

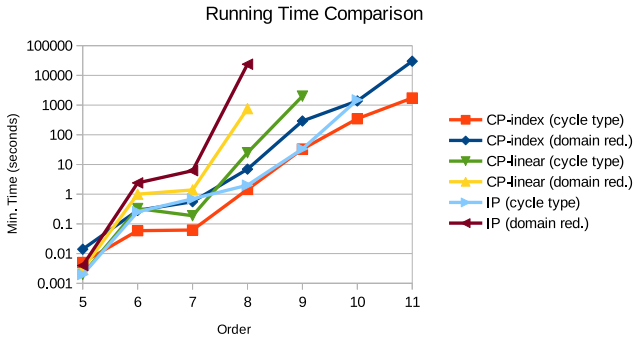

Figure 1: A comparison of the running times for the various models and symmetry breaking methods that we considered in the orders $5 \leq n \leq 11$.

Model	5	6	7	8	9	10
IP	0.1	Timeout	3.2	6.4	344.5	3,046.4
CP-linear	0.0	Timeout	8.0	1,967.1	58,637.8	Timeout
CP-index	0.0	Timeout	7.8	36.3	378.7	214.8

Table 1: Timings in seconds for orders $5 \leq n \leq 10$ with no symmetry breaking. All models timed out at 60,000 seconds for $n = 11$.

References

Appa, G.; Magos, D.; and Mourtos, I. 2006. Searching for Mutually Orthogonal Latin Squares via integer and constraint programming. *European Journal of Operational Research*, 173(2): 519–530.

Appa, G.; Mourtos, I.; and Magos, D. 2002. Integrating Constraint and Integer Programming for the Orthogonal Latin Squares Problem. In *Lecture Notes in Computer Science*, 17–32. Springer Berlin Heidelberg.

Colbourn, C. J.; and Dinitz, J. H. 2006. *Handbook of combinatorial designs*. CRC press.

Dantzig, G. 1963. *Linear programming and extensions*. Number 48 in Princeton Landmarks in Mathematics and Physics. Princeton University Press.

Gomes, C. P. 2000. Structure, duality, and randomization: Common themes in AI and OR. In *AAAI-00 Proceedings*, 1152–1158.

Laywine, C. F.; and Mullen, G. L. 1998. *Discrete mathematics using Latin squares*. John Wiley & Sons.

OEIS Foundation Inc. 2021. *The On-Line Encyclopedia of Integer Sequences*. http://oeis.org/A002865.