A Method For Methodoku

Stan Wagon, Department of Mathematics, Macalester College, St. Paul MN 55105 ⟨wagon@macalester.edu⟩

The recent book, *Methodoku Mayhem,* by Mark Davies presents many puzzles, each based on a ringing method. My wife, Joan Hutchinson, has been ringing for over 50 years; with some help from Joan and Mark, I could learn the terminology and bring some mathematics to bear on the puzzles. Here I will show how linear programming can be used to analyze the various situations, with some surprising results. The main difficulty is working out how to describe the basic concepts (such as palindromic and double symmetry) in terms of linear inequalities. Once that is done, the method is very fast and not only solves Davies's puzzles, but also leads to new and surprising minimalist versions of some of them.

Figure 1 shows puzzle 30, with location clues in the grid and additional method clues in the caption. The grid represents one lead and the object is to fill in all the squares with bells.

1	2	3	4	5	row 0
	1				row 1
		1			row 2
			1		row 3
				1	row 4
				1	row 5
			1		row 6
		1			row 7
	1		2		row 8
1					row 9
1	4				row 10

Figure 1. Davies's Puzzle 30. Additional clues: the method has palindromic symmetry, there are no long places, and the plain course consists of four leads. The row labels show the indexing used.

When the changes of a lead are repeated, they eventually lead to rounds and that larger grid is the plain course. The number of repetitions is the number of times the permutation represented by the last row must be repeated until it becomes the identity permutation; this is called the *order* of the permutation. So in Figure 1, the last row will be a permutation of order 4.

These puzzles use standard change-ringing terminology, with two caveats: Davies's *lead-end* is the last row, while some others call this the lead head; and his use of symmetry conforms to convention and is therefore not the most general form (details below). The interaction between palindromic symmetry and double symmetry is a key aspect of many of his puzzles. Another assumption in all these puzzles is that the lead-end is not just rounds.

Any change is determined by the locations where places are made and so a method is determined by place-notation: a list of place locations (with "×" indicating no places are made). The solution to Puzzle 30 is 3 . 1 . 5 . 1 . 5 . 1 . 5 . 1 . 3 . 145, the Chase Bob Doubles method. The last row is 14532, which, in the usual mathematical view of a permutation, is the 4-cycle $2 \to 4 \to 3 \to 5 \to 2$. Each of his puzzles corresponds to a known ringing method.

Palindromic symmetry means that the place-notation for the lead, viewed as a circular list, reads the same forward or backward when one starts at the half-lead or lead end (Fig. 2).

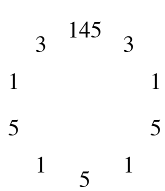

Figure 2. A palindromic set of changes.

Double symmetry is similar: each change in the lead is replaced with its reflection (e.g., for 8 bells, places at 14 become places at 58) and the resulting sequence of places is the same as the original, after no translation or translation by a half-lead, and viewed in circular form. For example, Puzzle 62 from the book has the solution ×14.58.36.14.58×18×58.14.36.58.14×18 and has both palindromic and double symmetry. Applying reflection gives ×58.14.36.58.14×18×14.58.36.14.58×18, the same as the original after a half-lead rotation.

There are several facts (some obvious, some not) that can be deduced from the symmetry properties. For example, for double symmetry with a half-lead translation, the permutation corresponding to the reverse of the half-lead handstroke row, when applied twice, equals the permutation at the lead end. In the other translation case, double symmetry means that every row equals its reflection. Another interesting property is that if a method has a plain bob lead end and palindromic symmetry, then the last change (when the number of bells, n, is even) must be 12 or $1n$. A proof of this can be found at Martin Bright's post. If n is odd then the last change is 1 or $12n$.

No long places means that a bell cannot be in the same position for three consecutive rows, referring to all the rows in the plain course; therefore the place-notation, viewed circularly, must not have the same place entry in two consecutive changes.

Davies's grids have modest size, so it is not surprising that some sort of computer search (such as backtracking) can be used. Here we will describe an extremely fast computational method that can

- solve a given puzzle;
- prove that the solution is unique;
- determine smaller sets of clues that lead to a unique solution.

My method uses ILP, for integer-linear programming. The details are in the next section, but the basic idea is that the conditions of the puzzles can be described by equations and inequalities that are purely linear, such as $3x + y = 13$ or $3x + y \leq 7$. This approach solved the two dozen of the book's puzzles that I tried, including the difficult bonus puzzle, taking only about one second for each. But it took days, not seconds, to set everything up.

I found that many puzzles from Davies's book have unique solutions when the clue set is smaller than the one presented. For Puzzle 30, two much-reduced versions (all treble clues are removed) are shown in Figure 3; the solution to each is unique. The 45 one is not too hard but the 53 one is quite difficult. Davies solved it by hand, commenting that "there are quite a few cases to iterate through, but not so many that it is tedious. Each nonsolution case differs nicely in terms of the type of reasoning required to rule it out. The line that leads to the eventual solution has the most steps, with the other cases leading to dead ends more quickly. So overall I would say it is a very nice puzzle, if extremely intimidating to start with. I think this is definitely the hardest five-bell puzzle I have yet seen. But I like it a lot." Another striking example is the bonus puzzle in Davies's book; it uses 12 bells and has 16 location clues. If one deletes the clue for the tenor (bell 12, notated T) in the penultimate row and also both clues in the row containing 1 and 2, the puzzle is still valid and solvable by hand, and the location-clue count is reduced from 16 to 13.

1	2	3	4	5
		4		
5				

1	2	3	4	5
5				
				3

Figure 3. Two versions of Puzzle 30 with a much smaller set of clues. Each has a unique solution with no long places, palindromic symmetry, and a four-lead plain course.

A new puzzle, by Mark Davies and me, is in Figure 4. This one is remarkable as it has only one location clue. The clue about the lead end means that the last row is 13527486, 15738264, 17856342, 18674523, 16482735, or

14263857. This puzzle has a unique solution using only the clues in the caption and the single location clue that bell 5 is at the lead in row 10. It works equally well with these five other single location clues: bell 2 at (3, 9) or (4, 10); bell 5 at (5, 1) or (4, 6); or bell 4 at (5, 14). To solve it by hand, the fact mentioned earlier about the final change being 12 or 18 (place notation) is helpful. And while the published puzzles all take about one second for ILP to solve, this one took 2200 seconds (about 37 minutes).

1	2	3	4	5	6	7	8
5							

Figure 4. The clues for this minimalist Methodoku are: palindromic and double symmetry, no long places, a plain bob lead end, and just a single location clue.

Applying ILP to Methodoku Puzzles

Here we show how to express all the ringing concepts in a linear fashion, which is the key to using ILP. Some of the steps are tricky and readers can skip this section.

I use n for the number of bells and (x, y) for a grid location, where x runs from 1 to n and y runs from 0 to $k-1$, where k is the number of rows in the lead. Thus row 0 is rounds. The main task is to represent the various clues in terms of linear equations and inequalities. For such a system of equations and inequalities, integer-linear programming can be used to find a solution or determine that none exists. ILP is an extremely important extension of linear programming (LP). Because LP, developed 75 years ago, is very very fast, ILP, which uses many calls to LP, works well on problems of modest size. For Puzzle 30, the number of variables required for all the constraints is 575; even several thousand variables is considered to be of modest size.

Here is how ILP is set up. The main variable is $B[x, y, i]$ where (x, y) is the grid position and i is one of the bells. One can ask that $B[x, y, i]$ be an integer (the "I" of ILP) and also $0 \le B[x, y, i] \le 1$; these force the value to be 0 or 1, with a 1 indicating that bell i is at position (x, y). First there are several routine constraints, all of which are linear.

- Each grid square contains exactly one bell: $B[x, y, 1] + \ldots + B[x, y, n] = 1$, for each position (x, y).
- Start with rounds: $B[1, 0, 1] = B[2, 0, 2] = \ldots = B[n, 0, n] = 1$.
- Each bell appears in each row: $B[1, y, i] + B[2, y, i] + \ldots + B[n, y, i] = 1$ for each bell i and row y.
- A bell moves at most one position: this is handled by looking at all pairs (x, y, i), $(X, y+1, i)$ where x and X differ by two or more. For each pair, use $B[x, y, i] + B[X, y+1, i] \le 1$; that is, bell i occupies at most one of the two distant positions.
- The location clues: $B[x, y, i] = 1$ for each specification that bell i is at (x, y).

And now the method clues. Palindromic symmetry is delicate and to express it linearly requires a trick. We want to introduce $P[x, y, i]$ to mean that bell i appears at both (x, y) and $(x, y+1)$; i.e., it makes places at (x, y). But how to relate P to B? The equation $P[x, y, i] = B[x, y, i] \cdot B[x, y+1, i]$ works, but multiplication is illegal in the linear world. The trick is to use four inequalities to force the multiplicative equation: $P[x, y, i] \geq B[x, y, i] + B[x, y+1, i] - 1$, $P[x, y, i] \geq 0$, $P[x, y, i] \leq B[x, y, i]$, and $P[x, y, i] \leq B[x, y+1, i]$. It is easy to see that these four inequalities force $P[x, y, i]$ to be 1 if and only if $B[x, y, i]$ and $B[x, y+1, i]$ are both 1. So $P[x, y, i]$ will correctly determine whether bell i makes places at (x, y). We then set $P[x, y]$ to be the sum of the $P[x, y, i]$ over the bells i; this will indicate whether any bell makes places at (x, y). Now the palindromic equations are easy: $P[x, y] = P[x, k+1-y]$ for any position with $0 \leq y \leq k-3$ (the -3 is because the last change is ignored). Double symmetry is similar, with the translations through a half-lead and whole lead treated separately.

No long places is expressed by $P[x, y] + P[x, y+1] \leq 2$ for all locations, and also $P[x, k-2] + P[x, 0] \leq 2$ for the wraparound change.

Some deviousness is need for the condition that there is a plain bob lead-end or that the full course involves a specific number of leads, but these can be handled linearly. For example, suppose a clue for an 8-bell puzzle is that there are seven leads in the plain course. The permutation corresponding to the lead end is then a 7-cycle. (Any permutation is made up of disjoint cycles.) One can add a constraint that the lead-end is not a 3-cycle by saying that, for example, the bells in positions 2, 5, and 7 contain at least one bell not equal to 2, 5 or 7. That means that the cycle $2 \rightarrow 5 \rightarrow 7 \rightarrow 2$ does not occur. Do this for all possible 3-cycles, and then do something similar for all possible 2-cycles and 1-cycles (fixed points). If there are no such short cycles, the final permutation must be a 7-cycle, because any smaller cycle is ruled out (e.g., a 6-cycle is ruled out because that would mean a fixed point exists). Forcing a plain bob lead end is trickier and the details are omitted (the key point is that certain bell pairs are always within two positions of each other). But it can be done with a single set of inequalities (as opposed to trying every possible plain bob lead end, which works but is slower).

With the above setup one can call on ILP. A solution to the linear equations and inequalities gives bell locations consistent with all clues; the nonexistence of a solution to the equations means there is no way to place the bells. This approach solves any of Davies's puzzles. For ILP I use *Mathematica*®, which makes use of the state-of-the-art *GUROBI* program; there are several free ILP solvers available.

Generating Surprising New Puzzles

Once the setup is working to find solutions, one can take things to a deeper level in order to prove uniqueness and reduce the clue set. Consider Puzzle 30 (Fig. 1). In the solution, places are made at a set S of 12 specific locations. These are derived from the place-notation 3 . 1 . 5 . 1 . 5 . 1 . 5 . 1 . 3 . 145: S consists of (3, 0), (1, 1), (5, 2), (1, 3), (5, 4), (1, 5), (5, 6), (1, 7), (3, 8), (1, 9), (4, 9), (5, 9). So one can first search for a solution using all the clues for the puzzle together with the condition that there are 11 or fewer places made. If there is no solution for that, then check for the same with 13 or more places. If those two have no solution, then any solution using the clues must have places made exactly 12 times. To finish, search for a solution where places are made at fewer than the 12 locations in S. If this situation cannot occur, then any solution using the clues must have places made 12 times and those 12 must include all the locations in S. So places are made at the locations in S and nowhere else. Because the place locations uniquely determine the entire lead, this proves that the solution is unique. One can set this up using fewer clues, and that is how minimalist puzzles such as those in Figures 3 and 4 were discovered.

There are unsolved problems in change-ringing, the most prominent being whether a bobs-only peal is possible for Erin Triples (see the paper by Haythorpe and Johnson). ILP might or might not be useful in attacking them. But for Methodoku, ILP is the ultimate solver. Many industries use ILP for scheduling and optimization; for example, it is used to construct the season schedules for professional sports leagues and by airlines to manage routes and schedules. Indeed, I use it myself in a consulting business where we assign college students to classes with limited size so as to maximize student happiness. It is satisfying to now add bell ringing to the list of applications.

Acknowledgment. I am extremely grateful to Mark Davies for help with bell-ringing terms and verifying the uniqueness property by hand for the variants of his puzzles. And thanks to Rob Pratt for improving my method of relating P to B.

References

Martin Bright, Plainbob leadheads <https://lists.ringingworld.co.uk/pipermail/ringing-theory_bellringers.org/2021-February/027165.html>

<https://cccbr.github.io/method_ringing_framework/classification.html#definitions>

Mark B. Davies, *Methodoku Mayhem,* Ringing World, Andover, U.K. 2020.

John Harrison, Glossary of Ringing Terms ⟨http://jaharrison.me.uk/Ringing/Glossary⟩.

Michael Haythorpe and Andrew Johnson, Change ringing and Hamiltonian cycles: The search for Erin and Stedman triples, *Electronic Journal of Graph Theory and Applications* **7**:1 (2019) 61–75 ⟨https://arxiv.org/pdf/1702.02623.pdf⟩