

An Interactive Application to Solve Sudoku Variant Puzzles

Helmut Simonis, Luis Quesada

University College Cork, Cork, Ireland

ModRef 2026, July 2026

A World
Leading SFI
Research
Centre

Science
Foundation
Ireland **sfi**
For what's next

Insight 

SFI RESEARCH CENTRE FOR DATA ANALYTICS

HOST INSTITUTIONS



NUI Galway
O'Gallimh



University College Cork, Ireland
Coláiste na hOllscoile Corcaigh



Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin



FUNDED BY:



Key Points

- Solving Sudoku Variant Puzzles
- Wide variety of constraints besides all different, over 200 handled
- Interested in propagation, not search
- Off-the-shelf solver, Choco
- Human interaction by stating constraints
- Solving 97% of Sudoku 2025 Grand-Prix puzzles
- Help humans to enjoy puzzles, not replace humans!

Insight is was one of the largest data research and innovation centres in Europe...

Insight Today



4
Co-Lead Universities
4 partner institutions

Built on **20** years of
research in Data
Analytics and AI

1250
Academics, Postdocs, PhDs,
RAs

3900+
Scientific conference
and journal papers

175+
Funded collaborations
with industry partners

350+
Research Awards

16
Spin out companies
107 license agreements

135+
H2020 consortia, 500+
collaborations, 40+
countries

1,137+ school
visits, 28,000 students

398
PhDs graduated

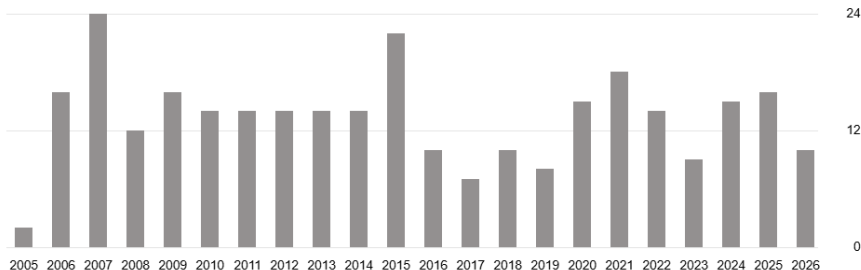
New and Improved: Rinn Artificial Intelligence: Research & Innovation in Data Science and AI

Rinn Artificial Intelligence is designed to serve as a national hub for research and innovation in data science and AI. The programme has two overarching goals: (i) to advance foundational cutting-edge research in Data Science and AI, and (ii) to pursue open, international, and inter- and transdisciplinary research aimed at studying, designing, and implementing strategies to address societal challenges.



Going back 21 years, ModRef 2005

- H. Simonis. Sudoku as a Constraint Problem. ModRef 2005. Sitges, Spain.
- Classification of propagation strength needed to solve regular Sudokus
- Over 300 citations on Google Scholar



Outline

The Problem

Modelling

Manual Constraints

Results

Summary

Running Example ("Second Breakfast" by "99%Sneaky")

Sudoku Variant Solver [Instance Name: 23] "Second Breakfast" by 99%Snaky

File Edit Scenario View Window Help

Problem DrawingArea X

Constraints

	A	B	C	D	E	F	G	H	I
1	1	2	3	4	5	6	7	8	9
2	10	11	12	13	14	15	16	17	18
3	19	20	21	22	23	24	25	26	27
4	28	29	30	31	32	33	34	35	36
5	37	38	39	40	41	42	43	44	45
6	46	47	48	49	50	51	52	53	54
7	55	56	57	58	59	60	61	62	63
8	64	65	66	67	68	69	70	71	72
9	73	74	75	76	77	78	79	80	81

Rules

standard

This is a standard Sudoku, each row, each column, and each major block (outlined with a wider line) must consist of pairwise different values.

sumV

The two values in cells marked by a V must add to five.

sumX

The two values in the cells marked by an X must add to ten.

blackDot

The values in two cells marked by a black dot must be X and 2X, in any order.

whiteDot

The values in the cells marked by a white dot must be consecutive, in any order.

fortress

Each value in a fortress must be bigger than the value on any orthogonal neighbours outside the fortress.

arrow

The sum of the digits on the arrow must sum to the value in the circle. The sequence <7, 4, 2, 1> would form a valid arrow of length four. The value in the circle is always larger than any one of the values on the arrow.

cage

The values inside the cage must add to the given total. All values in the cage must be different from each other.

renban

The values on the line of length n must consist of n consecutive integers, but they can appear in any order. The values <3, 5, 4, 2, 6> for example form a valid renban line of length five.

germanWhisper

Neighbouring cells on the line must differ by at least five. The value five cannot appear on the line. The sequence <1, 6, 1, 8, 2, 7> would be a legal German Whisper line.

quad

The values in the circle must appear at least once in the four connected cells.

Reasoning

DC

Run

Reset

Back

Forward

Propagate

Find

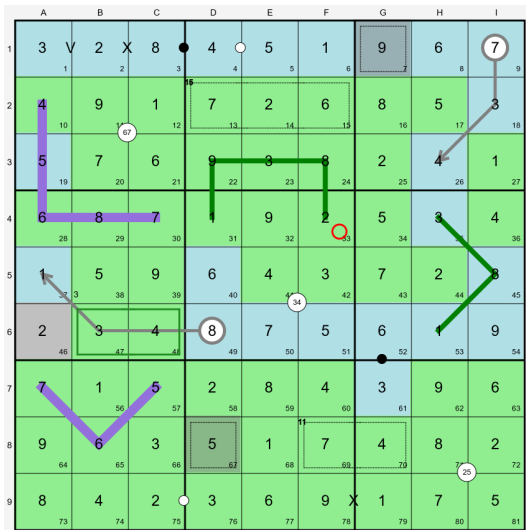
Solve

Explain

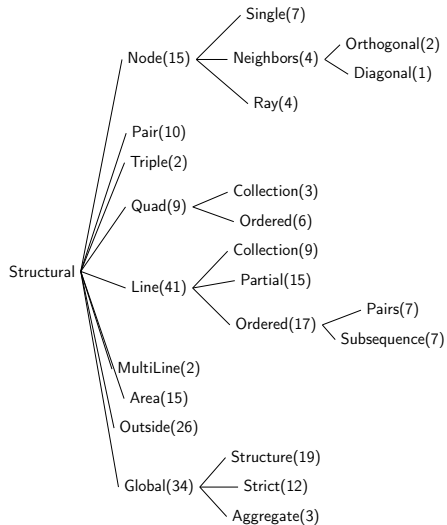
Snapshot

Propagation Step 0 Row 0

What do the constraints mean?



Morphology



Outline

The Problem

Modelling

Manual Constraints

Results

Summary

Modelling Choices

- One-to-one link to global constraint
- Conjunction of basic constraints
- Disjunction
- Tablification
 - Achieve domain consistency of basic constraint

Example: Renban

- Meaning: Values on a line form a collection of distinct, consecutive integers
- Global constraint: `alldifferent_consecutive_values`
- $\text{renban}(X) \equiv \min(X, \mathbf{Min}) \wedge \max(X, \mathbf{Max}) \wedge \mathbf{Max} - \mathbf{Min} = |X| - 1$
- Table: 2 values 16 tuples, 8 values 80,640 tuples

Example: GreatestLine (Line (Collection) Constraint)

- The largest value on the line is greater than the sum of any other two cells on the line

	A	B	C	D	E	F	G	H	I
1	9	4	3	2	1	6	7	5	8
2	2	5	6	7	4	8	9	1	3
3	1	8	7	5	3	9	2	6	4
4	4	1	8	3	9	2	5	7	6
5	3	2	5	6	8	7	1	4	9
6	7	6	9	4	5	1	3	8	2
7	5	7	4	8	2	3	6	9	1
8	6	3	1	9	7	4	8	2	5
9	8	9	2	1	6	5	4	3	7

Implementation: GreatestLine

- Using sort global constraint, plus inequality
- $\text{sort}(X, Y)$: Y is sorted X
- Quite difficult to express with basic constraints alone

$\text{greatestLine}([X_1, X_2, \dots, X_n]) \equiv \text{sort}([X_1, X_2, \dots, X_n], [Y_1, Y_2, \dots, Y_n]) \wedge Y_n \geq Y_{n-1} + Y_{n-2}$

Example: Parity Sum (Outside Constraint)

- The clue outside the grid gives the sum of the initial cells in the grid, up to and including the first parity change.
- A one-digit outside clue could reference just the first cell, since that would be the first odd/even number in that direction.

[illegible]

Parity Sum Code for Choco

```
case ParitySum -> {  
  IntVar[] x = lineVars;  
  int v = oc.getValue();  
  x[0].eq(v).or(  
    x[0].mod(2).ne(x[1].mod(2)).and(x[0].add(x[1]).eq(v)),  
    x[1].mod(2).ne(x[2].mod(2)).and(x[0].add(x[1], x[2]).eq(v)),  
    x[2].mod(2).ne(x[3].mod(2)).and(x[0].add(x[1], x[2], x[3]).eq(v)),  
    x[3].mod(2).ne(x[4].mod(2)).and(x[0].add(x[1], x[2], x[3], x[4]).eq(v)),  
    x[4].mod(2).ne(x[5].mod(2)).and(x[0].add(x[1], x[2], x[3], x[4], x[5]).eq(v)),  
    x[5].mod(2).ne(x[6].mod(2)).and(x[0].  
      add(x[1], x[2], x[3], x[4], x[5], x[6]).eq(v))).post();  
}
```


Table Sizes (Examples)

Table	Number of Variables							
	2	3	4	5	6	7	8	9
Sum+AllDifferent	72	504	3,024	15,120	60,480	181,440	362,880	*
Sum	81	729	6,561	59,049	-	-	-	-
Renban	16	42	144	600	2,880	15,120	80,640	*
Between	-	168	672	3,192	16,800	94,488	556,512	3,390,072
Magic Square	-	-	-	-	-	-	-	41

Outline

The Problem

Modelling

Manual Constraints

Results

Summary

Manual Constraints

- Propagation alone is not sufficient for many instances
- Allow human to control solution process
- More than just fixing or removing values
- User interface components to create constraints on the fly

Proposed Manual Constraints

in($\langle X_1, X_2, \dots, X_n \rangle, \langle V_1, V_2, \dots, V_k \rangle$) We restrict the domain of all X_i variables to the set V .

mustContain($\langle X_1, X_2, \dots, X_n \rangle, \langle V_1, V_2, \dots, V_k \rangle$) For each value V_j , one of the X_i variables must take that value. This leads to a disjunction, but more reasoning can be performed.

mustNotContain($\langle X_1, X_2, \dots, X_n \rangle, \langle V_1, V_2, \dots, V_k \rangle$) None of the values V_j are possible for any of the X_i variables. We can remove these values from their domains.

sameValues($\langle X_1, X_2, \dots, X_n \rangle, \langle Y_1, Y_2, \dots, Y_n \rangle$) The values in vectors X and Y must be the same. This often arises when considering the interaction of two **alldifferent** constraints.

samePairs($\langle X_1, X_2, \dots, X_n \rangle, \langle Y_1, Y_2, \dots, Y_n \rangle, C$) The values must match pairwise, i.e. $X_i = Y_i + c$. In most scenarios offset c is zero.

allEqual($\langle X_1, X_2, \dots, X_n \rangle$) All variables X_i must take the same values. This is a constraint often needed when a lot of initial reasoning is performed to find equivalence classes of cells in the grid.

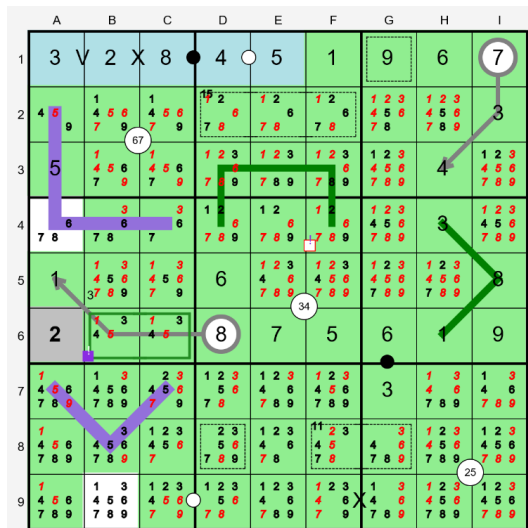
alldifferent($\langle X_1, X_2, \dots, X_n \rangle$) This allows to add additional, redundant **alldifferent** constraints. This can dramatically improve the propagation performed.

Adding Manual Constraint: C47 or C48 must be 3

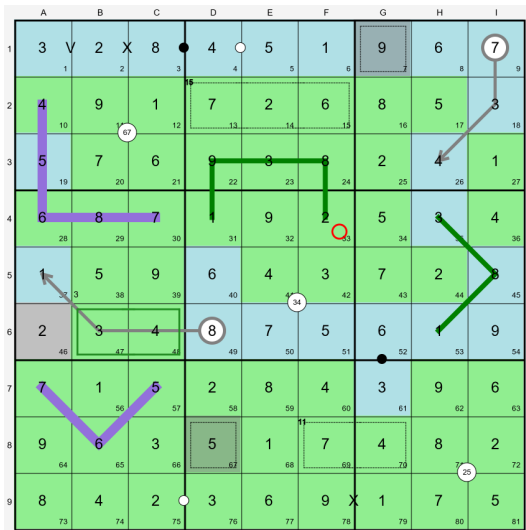
The screenshot displays the 'Sudoku Variant Solver' application window. The main area shows a 9x9 grid with columns labeled A-I and rows labeled 1-9. The grid contains numbers and various symbols (V, X, dots, circles) indicating constraints. A purple path is highlighted on the left side of the grid. The right-hand panel has tabs for 'Rules', 'Interaction', 'Display', and 'Colors'. The 'Interaction' tab is active, showing a list of actions and constraints. The 'Value' section has a dropdown menu set to '3'. The 'Group1' and 'Group2' sections have input fields. The 'Values' section has a list of values including '3'. The 'G1 - V' section has a 'MustContain' button. The 'G1 - G2' section has 'Same Values' and 'Same Pairs' buttons. The 'Global' section has 'Remove All' and 'Undo' buttons. The 'Inactive All' and 'Active All' buttons are also present.

Propagated

After Enforcing Manual Constraint



Solved Grid after Enforcing $C33 \neq 9$



Outline

The Problem

Modelling

Manual Constraints

Results

Summary

Four Benchmark Sets

- Nikoli regular Sudoku puzzle books
 - Can we solve popular, regular Sudokus with our system by propagation alone?
- Genuinely Approachable Sudoku (GAS) Puzzles, daily problems
 - Do we cover the variety of variants encountered?
 - Is our morphology comprehensive enough?
- WPF Sudoku Grand-Prix 2025
 - How do we compare to the best human speed-puzzle solvers?
- "Cracking the Cryptic" YouTube channel
 - Extremely popular, but often complex puzzles
 - Do we have the right manual constraints?

World Puzzle Federation: Sudoku Grand-Prix 2025

- Purely a pen and paper competition for humans
- No computers allowed
- 8 rounds, 90 minutes each for around 15 puzzles
- Points for each problem solved, additional time-points when finishing early

Grand-Prix Results for Rounds 1-8

Round	Puzzles	Points Available	Solved	Points Achieved	Time Points
R1	13	660	13	660	899
R2	13	720	13	720	899
R3	18	600	17	555	0
R4	12	780	11	674	0
R5	15	750	15	750	899
R6	13	750	13	750	899
R7	14	660	14	660	899
R8	14	690	13	631	0
Total	112	5,610	109	5,400	4,495

Grand-Prix Evaluation

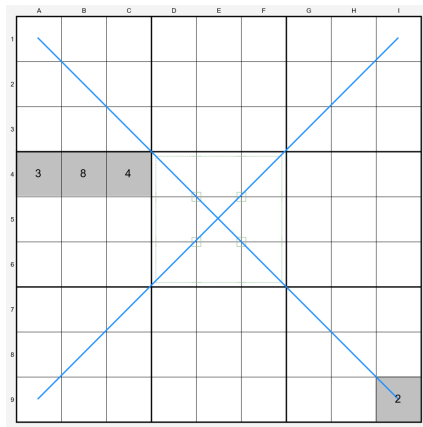
- All but three solved (109 of 112)
- Ranked 5th without including our time-points
 - Compared to human results including their time-points
- Ranked 1st including time-points
 - 9,895 points vs. 6,160.2 for best human

Cracking the Cryptic Examples

Nr	Instance	Cells Filled		=	≠	Manual Constraints Needed						
		Givens	Propagated			In	MC	MNC	SV	SP	AE	AD
52	Magic Square Sudoku	4	15	-	4	-	-	-	-	-	-	-
94	Non Consecutive Killer	1	0	4	1	-	-	-	2	-	3	-
30	The Miracle	2	12	-	15	-	-	-	-	-	-	-
33	Antiknight Sudoku	13	5	-	3	-	1	-	-	-	-	-
51	10 Million Views	4	13	-	-	-	-	3	-	-	-	-
56	Centipede	1	0	3	-	-	3	-	-	-	-	-
60	Spring Has Sprung	0	11	1	1	-	-	-	-	-	-	-
62	Game Of Arrows	3	2	2	1	-	1	-	-	-	-	-
68	The Climb	0	0	2	-	-	4	1	-	-	-	-
69	Simply Beautiful Sudoku	29	0	-	-	-	-	1	-	-	-	-
92	≤ 5 Sudoku	11	7	-	2	-	-	-	-	-	-	-
93	Superking	6	3	1	-	-	-	-	-	-	-	-

Magic Square by Aad van de Wetering (10 million views on YouTube)

- Normal sudoku rules apply.
- In addition, both marked diagonals must also contain the digits 1 to 9.
- Cells that are a knight's move apart (in chess) cannot contain the same digit.
- The central 3x3 box must form a magic square.
- Only four hints are given



After Initial Propagation (Presenting Hint: $C56 \neq 2$)

	A	B	C	D	E	F	G	H	I
1	1 7 8 9	1 3 4 5 6 7 9 2	1 3 5 6 7 8 9 3	3 4 5 7 9 4	1 4 6 8 9 5	1 3 5 6 7 8 6	2 7	1 4 5 6 7 9 8	1 4 6 7 9 9
2	1 2 4 5 6 7 9 10	1 7 9 11	1 2 3 5 6 7 9 12	2 3 5 9 13	1 5 8 9 14	1 3 5 6 8 15	1 3 5 6 7 8 9 16	1 4 6 7 9 17	1 4 5 6 7 8 9 18
3	1 2 5 6 7 8 9 19	1 2 3 4 5 7 9 20	1 6 8 9 21	3 4 7 9 22	2 6 8 23	1 3 7 8 24	1 3 6 25	1 5 6 7 9 26	1 4 5 6 7 9 27
4	3 28	8 29	4 30	6 31	7 32	2 33	1 34	1 5 35	1 5 36
5	2 7 6 37	2 7 38	2 6 39	1 40	5 41	9 42	4 43	8 44	3 45
6	1 5 46	5 9 47	1 9 48	8 49	3 50	4 51	7 52	2 53	6 54
7	1 2 4 5 6 7 8 9 55	1 2 3 4 5 6 7 9 56	1 7 9 57	2 3 7 9 58	2 4 6 8 59	1 6 7 60	1 7 61	1 5 6 7 9 62	1 4 5 6 7 8 9 63
8	1 2 4 5 6 7 8 9 64	1 4 6 7 9 65	1 2 5 6 7 9 66	2 4 5 7 9 67	1 2 6 8 68	1 5 6 7 8 69	1 5 6 7 8 9 70	3 71	1 4 5 6 7 8 9 72
9	1 4 6 7 9 73	1 3 4 5 6 7 9 74	1 3 5 6 7 8 9 75	3 4 5 7 9 76	1 4 6 8 9 77	1 5 6 7 8 78	1 5 6 7 8 9 79	1 4 5 6 7 9 80	2 81

Outline

The Problem

Modelling

Manual Constraints

Results

Summary

Summary

- Presented an interactive tool to solve Sudoku variant puzzles
- Over 200 constraints
- Wide, but still incomplete coverage of constraints found in the field
- Using off-the-shelf constraint solver
- Good results on variety of benchmark problems
- Aim: help human solve puzzles
- Potential for outreach

Acknowledgements

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 12/RC/2289-P2, co-funded under the European Regional Development Fund. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.