Integer programming approaches for solving variants of Sudoku puzzle

Kieu Thu Ha

Advisor: Dr. Le Xuan Thanh (IM, VAST)

June 24th, 2020

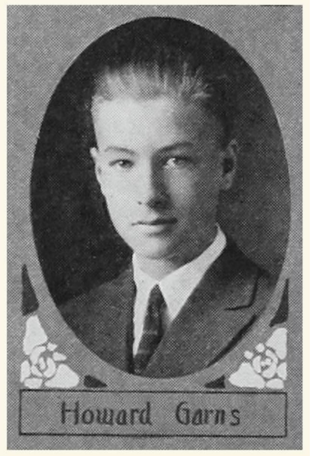
Contents

1. Introduction
   - Classical Sudoku
   - Some variants of Sudoku
2. IP-based approaches for Classical Sudoku
   - A binary linear programming formulation
   - An integer programming formulation
3. IP-based approaches for variants of Sudoku
   - Binary linear programming approach
   - Integer programming approach
4. Numberical experiments
5. Summary

Contents

1. Introduction
   - Classical Sudoku
   - Some variants of Sudoku
2. IP-based approaches for Classical Sudoku
   - A binary linear programming formulation
   - An integer programming formulation
3. IP-based approaches for variants of Sudoku
   - Binary linear programming approach
   - Integer programming approach
4. Numberical experiments
5. Summary

Introduction



Howard Garns (1905 - 1989): father of modern Sudoku

Classical Sudoku rule for size 9×9

(S1) Each cell of grid must be filled by exactly one digit from 1 to 9.

(S2) Each digit from 1 to 9 appears exactly once in each row of the grid.

(S3) Each digit from 1 to 9 appears exactly once in each column of the grid.

(S4) Each digit from 1 to 9 appears exactly once in each of the 3×3 blocks of the grid.

			5				6	
8		9					1	
1	6			8	7			
3				2	6			
		7		1		6		
			8	5				3
			4	7			2	1
	4					9		8
	8				3			

4	7	2	5	3	1	8	6	9
8	5	9	6	4	2	3	1	7
1	6	3	9	8	7	2	5	4
3	1	8	7	2	6	4	9	5
5	9	7	3	1	4	6	8	2
6	2	4	8	5	9	1	7	3
9	3	6	4	7	8	5	2	1
7	4	1	2	6	5	9	3	8
2	8	5	1	9	3	7	4	6

Variants of Sudoku

- X-Sudoku
- Windoku
- Killer Sudoku
- Jigsaw Sudoku

X-Sudoku

(X1) Each digit from 1 to 9 appears exactly once in each of the two main diagonals of the grid.

5							3	4
	7				5			
4				8				1
			4		6		9	2
6	4	2		3			1	
9			1	2				
			6					
						6		
						3		

5	8	9	7	6	1	2	3	4
1	7	3	2	4	5	9	8	6
4	2	6	9	8	3	7	5	1
3	1	8	4	7	6	5	9	2
6	4	2	5	3	9	8	1	7
9	5	7	1	2	8	4	6	3
7	3	5	6	9	2	1	4	8
8	9	4	3	1	7	6	2	5
2	6	1	8	5	4	3	7	9

Windoku

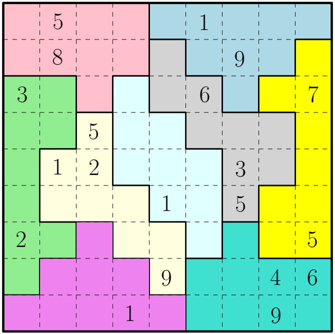
(W1) Each digit from 1 to 9 appears exactly once in each of the four additional blocks of the grid.

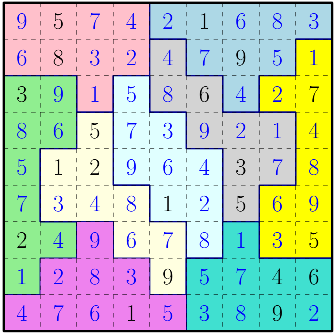
9			4		3			8
	1			9				
3		4				6		2
				7			4	
			5		8			
	2			3				
2		5				3		6
				4			8	
1			3		6			7

9	6	2	4	5	3	1	7	8
8	1	7	6	9	2	5	3	4
3	5	4	8	1	7	6	9	2
6	9	3	2	7	1	8	4	5
4	7	1	5	6	8	9	2	3
5	2	8	9	3	4	7	6	1
2	4	5	7	8	9	3	1	6
7	3	6	1	4	5	2	8	9
1	8	9	3	2	6	4	5	7

Jigsaw Sudoku

(J1) Each digit from 1 to 9 appears exactly once in each of the nine nonominos.

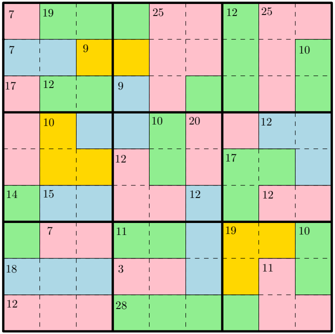


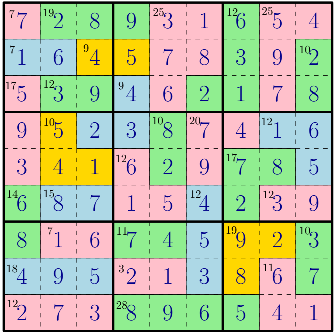


Killer Sudoku

(K1) No digit appears more than once in a cage.

(K2) The sum of all digits filled in a cage must equal the associated number of the cage.





Contents

1 Introduction
- Classical Sudoku
- Some variants of Sudoku

2 IP-based approaches for Classical Sudoku
- A binary linear programming formulation
- An integer programming formulation

3 IP-based approaches for variants of Sudoku
- Binary linear programming approach
- Integer programming approach

4 Numberical experiments

5 Summary

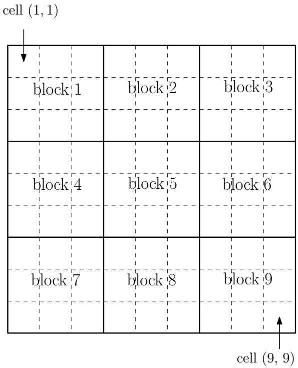
Framework

Formulate Sudoku problem as an IP

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \in \mathbb{Z}^n \end{aligned}$$

then solve numerically by some IP solver

Classical Sudoku



$$S = \{1, \ldots, 9\}$$

$$B := \{(1,1), (1,4), (1,7), (4,1), (4,4), (4,7), (7,1), (7,4), (7,7)\}$$

Contents

1. Introduction
   - Classical Sudoku
   - Some variants of Sudoku
2. IP-based approaches for Classical Sudoku
   - A binary linear programming formulation
   - An integer programming formulation
3. IP-based approaches for variants of Sudoku
   - Binary linear programming approach
   - Integer programming approach
4. Numberical experiments
5. Summary

A binary integer linear program

$$G = \{(i,j,k) \mid k \text{ is clue in cell } (i,j)\}, \quad x_{ijk} = \begin{cases} 1 & \text{if } k \text{ is filled in cell } (i,j) \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned}
(SUDOKU-BLP) \qquad & \min \sum_{i,j,k \in S} 0 \cdot x_{ijk} & & (1)\\
\text{s.t.} \qquad & x_{ijk} = 1 & \forall (i,j,k) \in G \qquad & (2)\\
& \sum_{k \in S} x_{ijk} = 1 & \forall i \in S, j \in S \qquad & (3)\\
& \sum_{i \in S} x_{ijk} = 1 & \forall j \in S, k \in S \qquad & (4)\\
& \sum_{j \in S} x_{ijk} = 1 & \forall i \in S, k \in S \qquad & (5)\\
& \sum_{i=p}^{p+2} \sum_{j=q}^{q+2} x_{ijk} = 1 & \forall k \in S, (p,q) \in B \qquad & (6)\\
& x_{ijk} \in \{0,1\} & \forall i \in S, j \in S, k \in S \qquad & (7)
\end{aligned}$$

Contents

1 Introduction
- Classical Sudoku
- Some variants of Sudoku

2 IP-based approaches for Classical Sudoku
- A binary linear programming formulation
- An integer programming formulation

3 IP-based approaches for variants of Sudoku
- Binary linear programming approach
- Integer programming approach

4 Numberical experiments

5 Summary

A non-linear integer programming formulation

F is the set of clue cells
$g(i,j)$ is clue inside cell (i,j)

$$
\begin{aligned}
(SUDOKU-NLIP) \qquad \min \; & \sum_{i,j \in S} 0 \cdot x_{ij} && && (8)\\
\text{s.t.} \qquad x_{ij} &= g(i,j) && \forall (i,j) \in F && (9)\\
|x_{ij} - x_{ik}| &\geq 1 && \forall i,j,k \in S, j < k && (10)\\
|x_{ij} - x_{kj}| &\geq 1 && \forall i,j,k \in S, i < k && (11)\\
|x_{p+i,q+j} - x_{p+k,q+l}| &\geq 1 && \forall (p,q) \in B, && \\
& && i,j,k,l \in \{0,1,2\} : 3i+j < 3k+l && (12)\\
x_{ij} &\in S && \forall i,j \in S && (13)
\end{aligned}
$$

Linearize non-linear constraints

$$y^+ = \begin{cases} y & \text{if } y > 0, \\ 0 & \text{if } y \leq 0, \end{cases} \qquad y^- = \begin{cases} 0 & \text{if } y \geq 0, \\ -y & \text{if } y < 0, \end{cases}$$

$$a^+ = \begin{cases} 1 & \text{if } y > 0, \\ 0 & \text{if } y \leq 0, \end{cases} \qquad a^- = \begin{cases} 1 & \text{if } y < 0, \\ 0 & \text{if } y \geq 0. \end{cases}$$

Since $|y| = y^+ + y^-$, non-linear constraint

$$1 \leq |y| \leq 8, y \in \mathbb{Z}$$

can be linearized by

$$y^+ - y^- = y \tag{14}$$

$$y^+ + y^- \geq 1 \tag{15}$$

$$a^+ \leq y^+ \leq 8a^+ \tag{16}$$

$$a^- \leq y^- \leq 8a^- \tag{17}$$

$$a^+ + a^- = 1 \tag{18}$$

$$y^+, y^- \in \mathbb{Z} \tag{19}$$

$$a^+, a^- \in \{0, 1\} \tag{20}$$

Contents

1. Introduction
   - Classical Sudoku
   - Some variants of Sudoku
2. IP-based approaches for Classical Sudoku
   - A binary linear programming formulation
   - An integer programming formulation
3. IP-based approaches for variants of Sudoku
   - Binary linear programming approach
   - Integer programming approach
4. Numberical experiments
5. Summary

Contents

1. Introduction
   - Classical Sudoku
   - Some variants of Sudoku
2. IP-based approaches for Classical Sudoku
   - A binary linear programming formulation
   - An integer programming formulation
3. IP-based approaches for variants of Sudoku
   - Binary linear programming approach
   - Integer programming approach
4. Numberical experiments
5. Summary

X-Sudoku

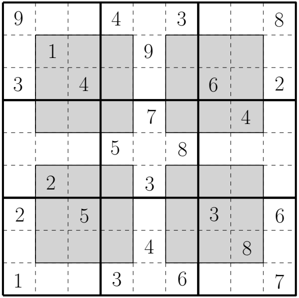
5							3	4
	7				5			
4				8				1
			4		6		9	2
6	4	2		3			1	
9			1	2				
			6					
						6		
						3		

(X1) Each digit from 1 to 9 appears exactly once in each of the two main diagonals of the grid.

$$\sum_{i \in S} x_{iik} = 1 \qquad \forall k \in S \tag{21}$$

$$\sum_{i \in S} x_{i,10-i,k} = 1 \qquad \forall k \in S \tag{22}$$

Windoku

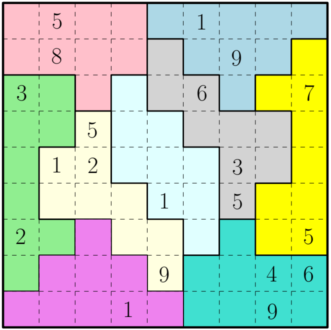


(W1) Each digit from 1 to 9 appears exactly once in each of the four additional blocks of the grid.

$$B_W := \{(2,2),(2,6),(6,2),(6,6)\}$$

$$\sum_{i=p}^{p+2} \sum_{j=q}^{q+2} x_{ijk} = 1 \quad \forall k \in S, (p,q) \in B_W \quad (23)$$

Jigsaw Sudoku

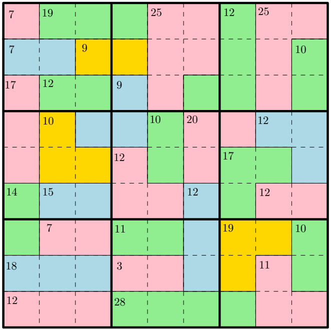


(J1) Each digit from 1 to 9 appears exactly once in each of the nine jigsaw blocks.

Let $\mathcal{H}$ be the set of jigsaw blocks.

$$\sum_{(i,j)\in H} x_{ijk} = 1 \quad \forall k \in S, H \in \mathcal{H} \qquad (24)$$

Killer Sudoku



(K1) No digit appears more than once in a cage.

$$\sum_{(i,j)\in C} x_{ijk} \leq 1 \quad \forall C \in \mathcal{C}, k \in S \qquad (25)$$

(K2) The sum of all digits filled in a cage must equal the associated number of the cage.

$$\sum_{(i,j)\in C} \sum_{k\in S} k x_{ijk} = n(C) \quad \forall C \in \mathcal{C} \qquad (26)$$

where $\mathcal{C}$ is the set of all cages,
$n(C)$ is the number associated with cage C

Contents

1. Introduction
   - Classical Sudoku
   - Some variants of Sudoku
2. IP-based approaches for Classical Sudoku
   - A binary linear programming formulation
   - An integer programming formulation
3. IP-based approaches for variants of Sudoku
   - Binary linear programming approach
   - Integer programming approach
4. Numberical experiments
5. Summary

X-Sudoku

5							3	4
	7				5			
4				8				1
			4		6		9	2
6	4	2		3			1	
9			1	2				
			6					
						6		
						3		

(X1) 'Each digit from 1 to 9 appears exactly once in each of the two main diagonals of the grid'

$$|x_{ii} - x_{jj}| \geq 1 \quad \forall i, j \in S : i < j \quad (27)$$

$$|x_{i,10-i} - x_{j,10-j}| \geq 1 \quad \forall i, j \in S : i < j \quad (28)$$

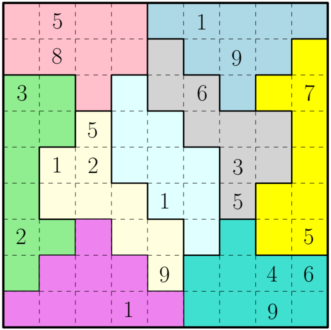
Windoku

9			4		3			8
	1			9				
3		4				6		2
				7			4	
			5		8			
	2			3				
2		5				3		6
				4			8	
1			3		6			7

(W1) 'Each digit from 1 to 9 appears exactly once in each of the four additional blocks of the grid'

$$\left|x_{p+i,q+j} - x_{p+k,q+l}\right| \geq 1$$
$$\forall (p,q) \in B \cup B_W,$$
$$i,j,k,l \in \{0,1,2\} : 3i+j < 3k+l \qquad (29)$$

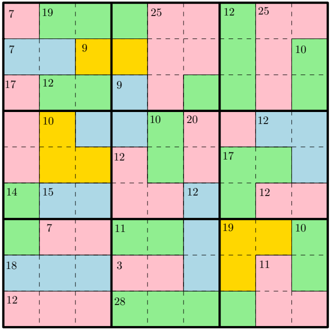
Jigsaw Sudoku



(J1) 'Each digit from 1 to 9 appears exactly once in each of the nine jigsaw blocks'

$$|x_{ij} - x_{kl}| \geq 1$$
$$\forall (i,j),(k,l) \in H : 9i + j < 9k + l \qquad (30)$$

Killer Sudoku



(K1) No digit appears more than once in a cage.

$$|x_{ij} - x_{kl}| \geq 1$$
$$\forall (i,j),(k,l) \in C : 9i + j < 9k + l, \quad (31)$$

(K2) The sum of all digits filled in a cage must equal the associated number of the cage.

$$\sum_{(i,j)\in C} \sum_{k \in S} k x_{ijk} = n(C) \quad \forall C \in \mathcal{C}. \quad (32)$$

Contents

1. Introduction
   - Classical Sudoku
   - Some variants of Sudoku
2. IP-based approaches for Classical Sudoku
   - A binary linear programming formulation
   - An integer programming formulation
3. IP-based approaches for variants of Sudoku
   - Binary linear programming approach
   - Integer programming approach
4. Numberical experiments
5. Summary

Numerical experiments

- Model by ZIMPL; IP solver: SCIP
- Intel Core i3-4005U CPU, 64 GHz Processor, 4 GB RAM
- Performance (average running time in seconds) of IP formulations:

Variant	(BLP)	$(NLIP)$
Classical Sudoku	-	31.7
X-Sudoku	-	26.7
Windoku	-	298.8
Jigsaw Sudoku	-	220.5
Killer Sudoku	7.6	475.5

Contents

1 Introduction
- Classical Sudoku
- Some variants of Sudoku

2 IP-based approaches for Classical Sudoku
- A binary linear programming formulation
- An integer programming formulation

3 IP-based approaches for variants of Sudoku
- Binary linear programming approach
- Integer programming approach

4 Numberical experiments

5 Summary

Summary

- Classical Sudoku and some variants
  - X-Sudoku, Windoku, Jigsaw Sudoku, Killer Sudoku
- IP formulations
  - BLP: three - index binary variables
  - NLIP: two-index integer variables
- Numerical experiments
  - BLP outperforms NLIP
  - Killer Sudoku is hardest
  - Classical Sudoku & X-Sudoku is easiest

Thank you for your attention!