

13. Integer Linear Programming

CPSC 535

Kevin A. Wortman



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

Recall: General LP Problem

general-form linear programming problem

input:

- ▶ Boolean for whether f is maximized/minimized
- ▶ vector $c \in \mathbb{R}^n$
- ▶ vector $b \in \mathbb{R}^m$
- ▶ vector $o \in \{\leq, =, \geq\}^m$
- ▶ $m \times n$ matrix A of real numbers

output: one of

1. “unbounded”;
2. “infeasible”; or
3. “solution” with a vector $x \in \mathbb{R}^n$ maximizing the objective function

Recall: General LP Problem

- ▶ integer *linear programming*: like general form, but all variables are integers instead of real
- ▶ i.e. each $x_i \in \mathbb{Z}$
- ▶ *Mixed Integer Programming (MIP)*: mixture of real and integer variables
- ▶ i.e. a subset $I \subseteq \{x_1, \dots, x_n\}$ of variables are restricted to integers

MIP problem

mixed-integer programming problem (MIP)

input:

- ▶ Boolean for whether f is maximized/minimized
- ▶ vector $c \in \mathbb{R}^n$
- ▶ vector $b \in \mathbb{R}^m$
- ▶ vector $o \in \{\leq, =, \geq\}^m$
- ▶ $m \times n$ matrix A of real numbers
- ▶ set $I \subset \{1, \dots, n\}$

output: one of

1. “unbounded”;
2. “infeasible”; or
3. “solution” with a vector $x \in \mathbb{R}^n$ maximizing the objective function; if $i \in I$ then $x_i \in \mathbb{Z}$

MIP Applications

- ▶ **discrete variables:** can formulate a business-logic whole number concept with
 - ▶ variable $x_i, i \in I$
 - ▶ example: you can buy 3 or 4 airplanes but not 3.7
- ▶ **true/false decision:** can formulate a true/false choice with
 - ▶ variable $x_i, i \in I$
 - ▶ constraints $0 \leq x_i$ and $x_i \leq 1$
- ▶ **choose among k alternatives:** more generally, can formulate a choice from $\{a, \dots, b\} \subset \mathbb{Z}$ with
 - ▶ variable $x_i, i \in I$
 - ▶ constraints $a \leq x_i$ and $x_i \leq b$

MIP Hardness

- ▶ Recall: hardness of general LP is an open question
- ▶ not proven in P , not proven NP -hard
- ▶ MIP **is** NP -complete
- ▶ specifying integer variables seems to make the problem substantially harder
- ▶ worst-case MIP programs are intractible
- ▶ **but** MIP solvers use lots of clever heuristics
- ▶ so specific MIP formulations are often computationally feasible in practice

Formulating Sudoku

Sudoku: input is a 9×9 grid, some cells are integers $\{1, \dots, 9\}$, others are blank

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

Rules:

1. Objective: fill every blank
2. Each row contains $\{1, \dots, 9\}$
3. Each column contains $\{1, \dots, 9\}$
4. Each 3×3 subgrid contains $\{1, \dots, 9\}$
5. (implies none of these regions has duplicates)

Formulating Sudoku: Variables

Create binary decision variables

$$x_{ijv} = 1 \Leftrightarrow \text{row } i, \text{ column } j, \text{ is assigned value } v$$

Specify that every x_{ijv} is an integer variable.

Add constraints for the variables to be used properly:

$$\begin{array}{ll} 0 \leq x_{ijv} \leq 1 & \forall i, j, v \in \{1, \dots, 9\} \quad (0 \text{ or } 1 \text{ indicator}) \\ \sum_{v=1}^9 x_{ijv} = 1 & \forall i, j \in \{1, \dots, 9\} \quad (\text{each cell has exactly one value}) \end{array}$$

Rule 1: Pre-Filled Cells

For each pre-filled cell at row i , column j , filled with value v , add one constraint

$$x_{ijv} = 1$$

Rules 2, 3: Each Row, Column is Filled Properly

“Row i is filled in properly” \Leftrightarrow each value v appears exactly once
in row i
(and for columns, resp.)

Add constraints:

$$\begin{aligned}\sum_{j=1}^9 x_{ijv} &= 1 & \forall i, v \in \{1, \dots, 9\} & \text{ rows are filled properly} \\ \sum_{i=1}^9 x_{ijv} &= 1 & \forall j, v \in \{1, \dots, 9\} & \text{ columns are filled properly}\end{aligned}$$

Rule 4: Each Subgrid is Filled Properly

For $r, c \in \{1, 2, 3\}$, let

$G(r, c) = \{(i, j) : i, j \in \{1, \dots, 9\} \text{ and } (i, j) \text{ is a cell of subgrid } r, c\}$.

Add constraints:

$$\sum_{(i,j) \in G(r,c)} x_{ijv} = 1 \quad \forall v \in \{1, \dots, 9\}; r, c \in \{1, 2, 3\} \quad \text{subgrids}$$

Objective Function

- ▶ Those constraints model all the rules of Sudoku!
- ▶ Still need an objective function
- ▶ Sudoku does not involve minimizing or maximizing anything
- ▶ Any arbitrary objective function works
- ▶ Define objective:
 maximize 0

Outcomes of MLP

- ▶ **Infeasible:**
 - ▶ it is impossible to fill the grid without breaking a rule
 - ▶ the pre-filled cells must break a rule and be invalid
- ▶ **Unbounded:** the objective function maximize 0 is a constant function, so is certainly bounded. So our MIP will never be unbounded.
- ▶ **Solution:** To fill in the grid: for each row i and column j , search for the v such that

$$x_{ijv} = 1$$

and then write v into cell (i, j) .

References

http://profs.sci.univr.it/~rrizzi/classes/PLS2015/sudoku/doc/497_Olszowy_Wiktor_Sudoku.pdf

<https://towardsdatascience.com/using-integer-linear-programming-to-solve-sudoku-puzzles-1>

<https://dingo.sbs.arizona.edu/~sandiway/sudoku/examples.html>