Introduction to ZIMPL and SCIP

Lê Xuân Thanh
Institute of Mathematics, VAST

Trường Đông về Lý thuyết trò chơi và Tối ưu Tổ hợp
VIASM, 05.12.2023

Contents

Contents

Sudoku puzzle

Given a 9×9 grid (as a composition of nine 3×3 subgrids).

Rules:

- Only digits from 1 to 9 can be filled in the grid
- Each cell of the grid must be filled by exactly one digit
- Each digit appears exactly once in each column, row, and 3×3 subgrid

Input: a partially complete grid

Objective: completely fill the grid

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

Sudoku puzzle

Given a 9×9 grid (as a composition of nine 3×3 subgrids).

Rules:

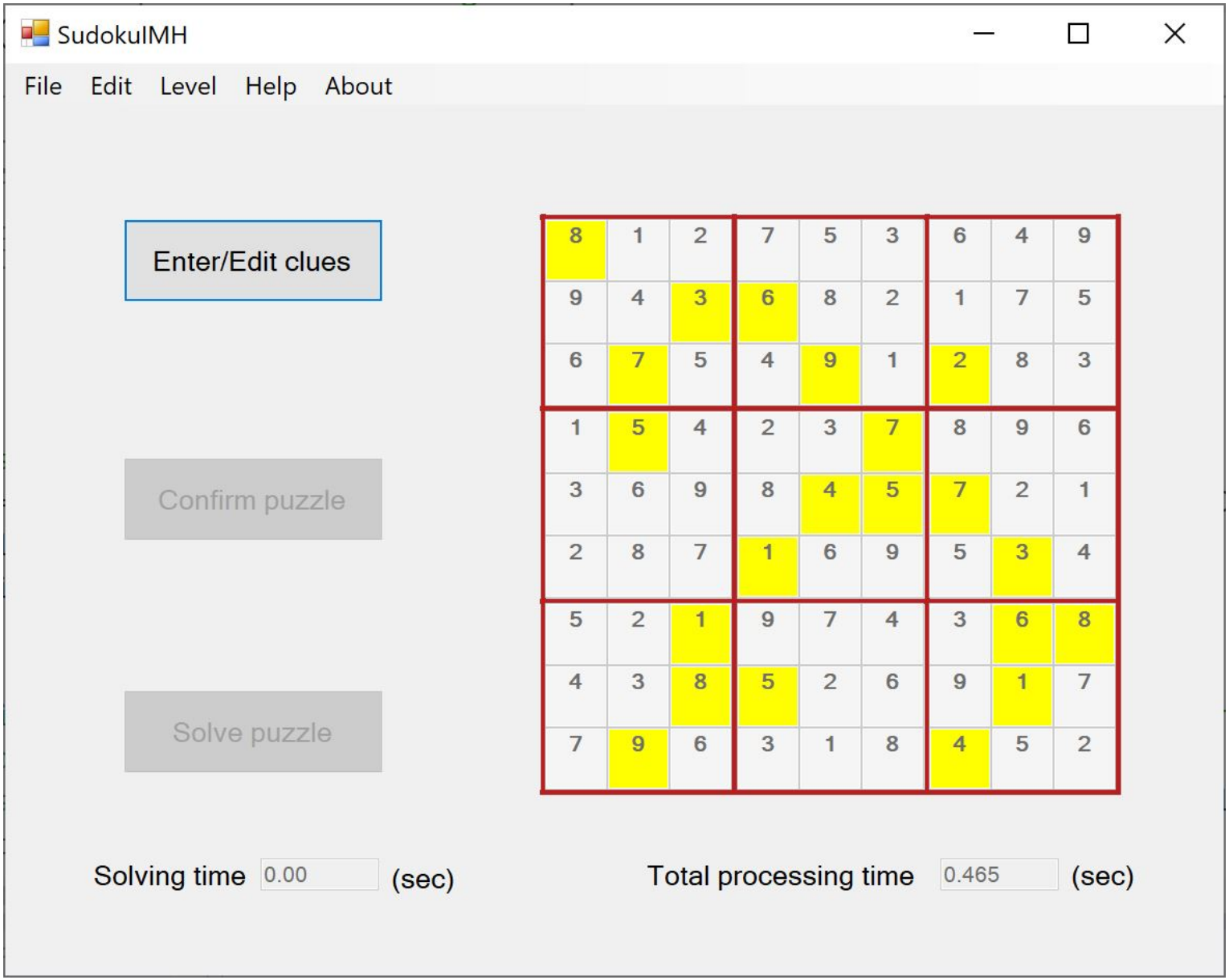
- Only digits from 1 to 9 can be filled in the grid
- Each cell of the grid must be filled by exactly one digit
- Each digit appears exactly once in each column, row, and 3×3 subgrid

Input: a partially complete grid

Objective: completely fill the grid

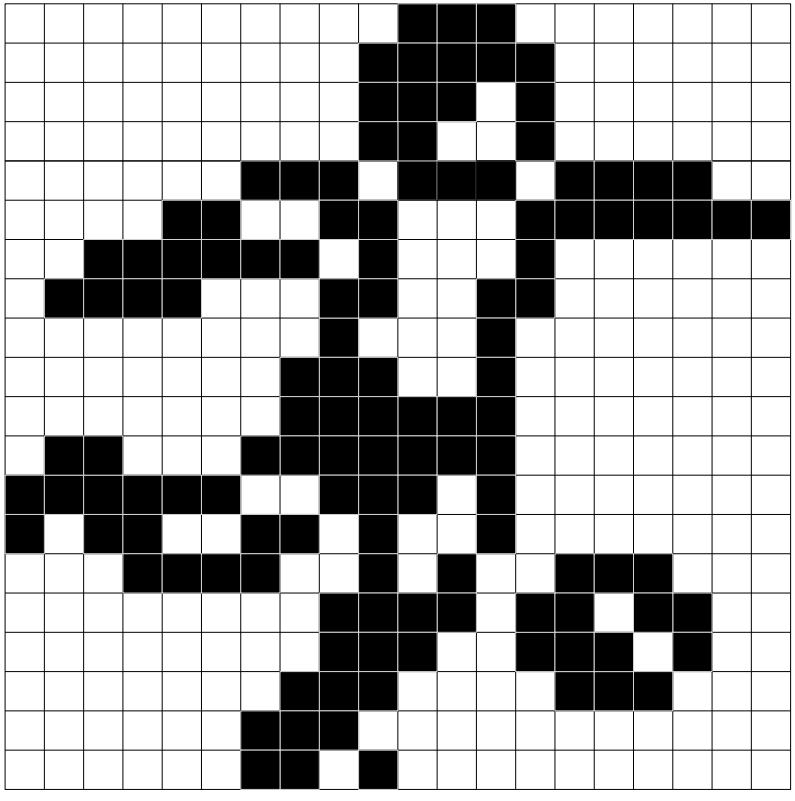
3	6	7	4	2	5	**8**	**9**	1
4	2	**1**	**3**	8	**9**	5	7	6
8	**5**	9	6	7	1	**2**	3	**4**
5	7	4	1	3	**2**	**6**	**8**	9
9	1	8	**7**	4	**6**	3	2	**5**
2	**3**	**6**	**5**	9	8	4	1	7
1	8	**3**	9	6	4	7	**5**	2
6	9	2	**8**	5	**7**	**1**	4	3
7	**4**	**5**	2	1	3	9	6	8

SudokuIMH: a Sudoku solver



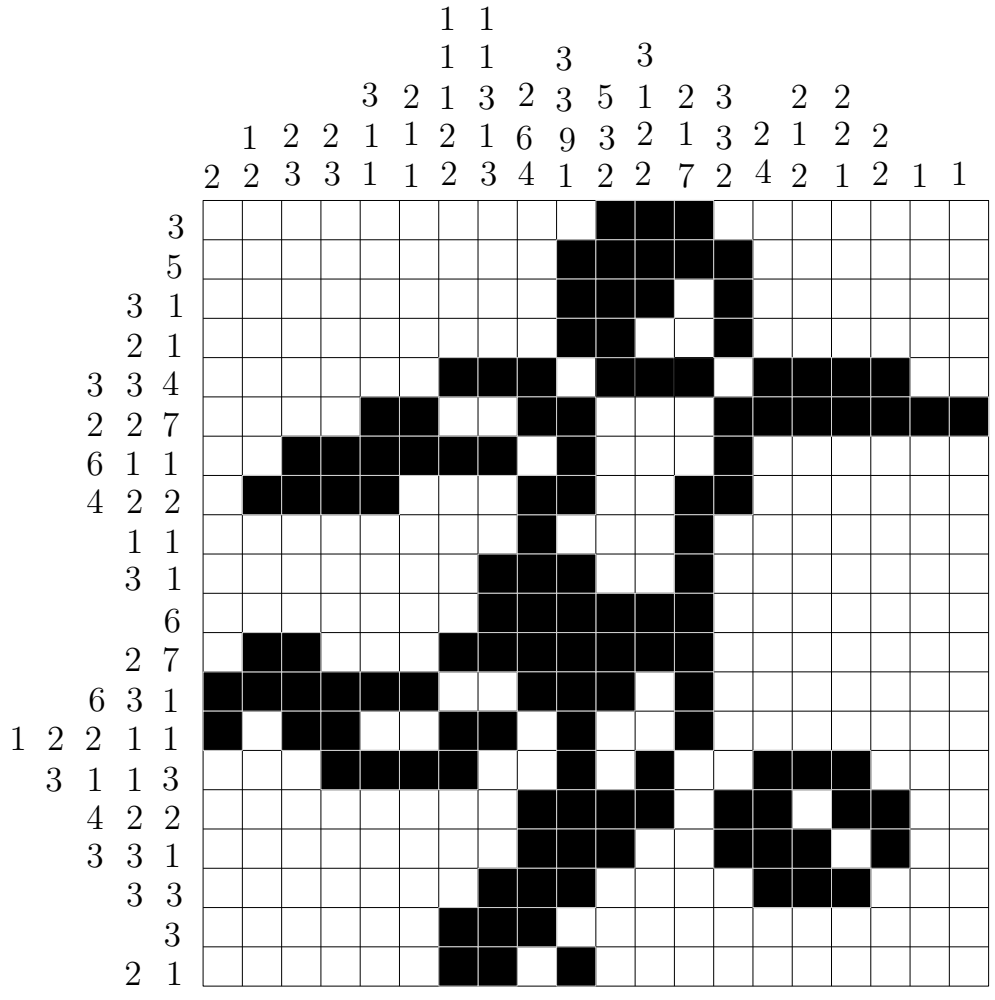
Contents

Example



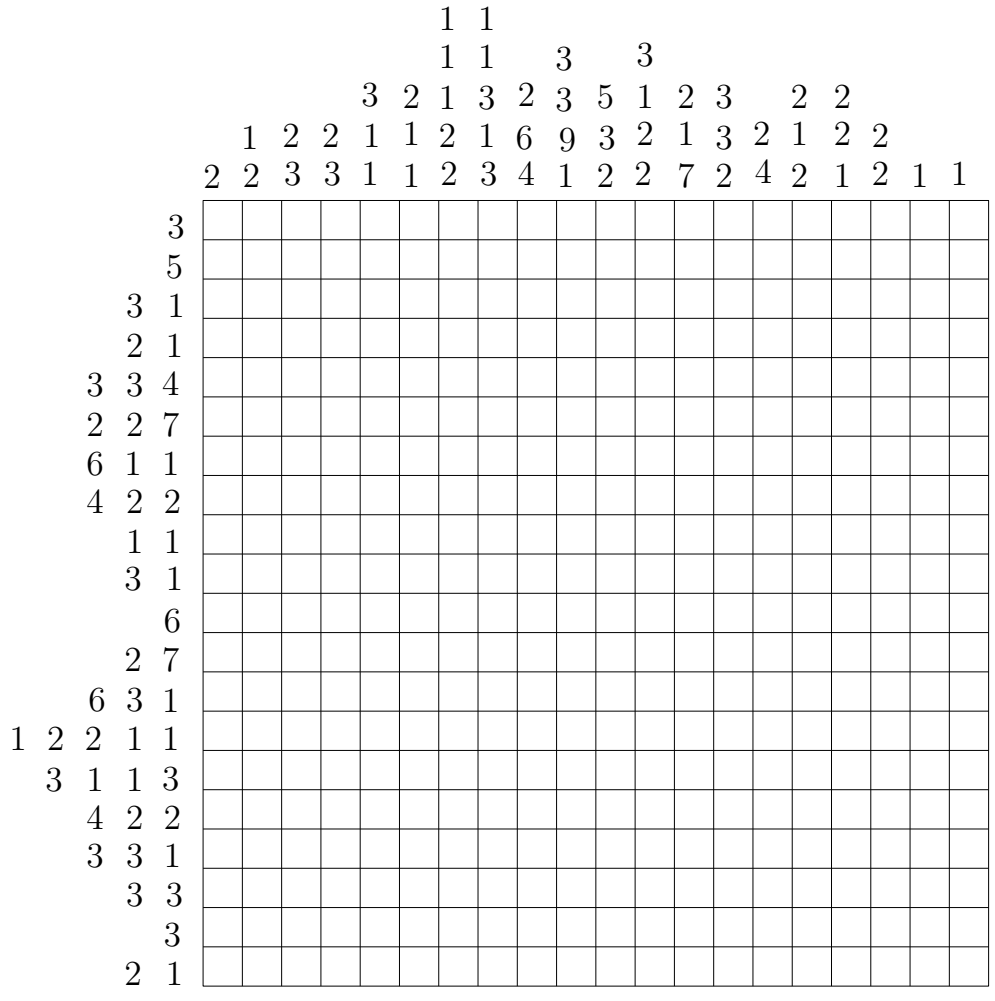
A grid of black and white cells

Example



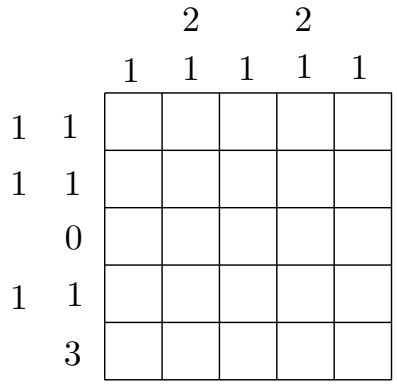
From picture to clues: easy

Example



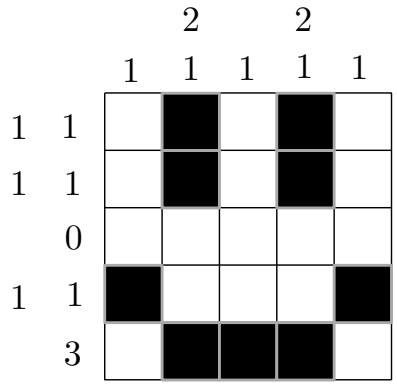
From clues to picture: hard

Picross puzzle description



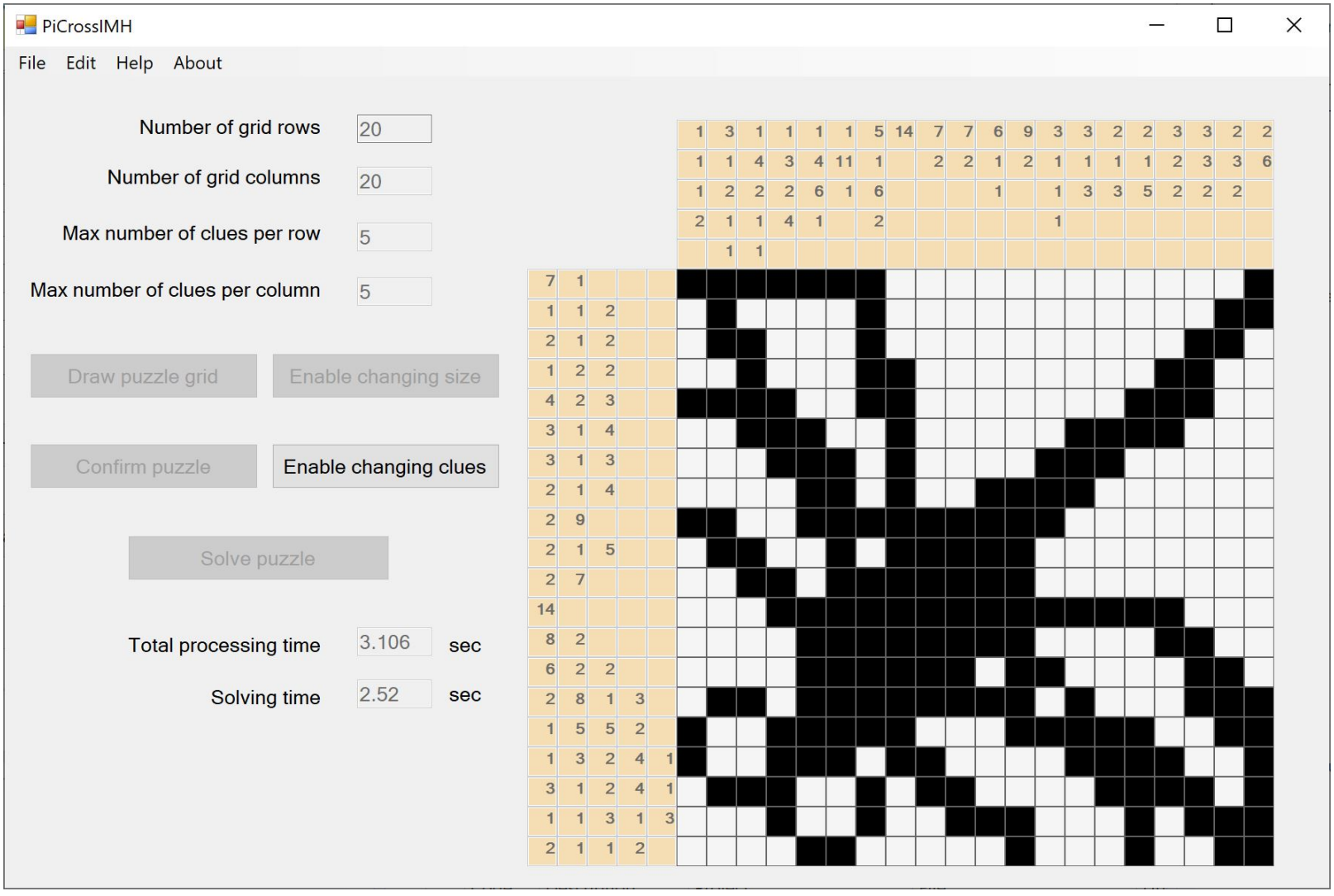
- Setup
  - A rectangle grid consists of $m \times n$ square cells
  - Clues (non-negative integers) for each row / column
- Terms
  - Cells are either colored (black) or blank (white)
  - Block = black cells without white in between (on same row / column)
- Rules
  - Each clue $\leftrightarrow$ to exactly one block
  - Clue = number of black cells in the corresponding block
  - On same row / column, blocks are separated by ≥ 1 white cell
  - Order of clues $\equiv$ order of blocks

Picross puzzle description



- Setup
  - A rectangle grid consists of $m \times n$ square cells
  - Clues (non-negative integers) for each row / column
- Terms
  - Cells are either colored (black) or blank (white)
  - Block = black cells without white in between (on same row / column)
- Rules
  - Each clue $\leftrightarrow$ to exactly one block
  - Clue = number of black cells in the corresponding block
  - On same row / column, blocks are separated by ≥ 1 white cell
  - Order of clues $\equiv$ order of blocks

PicrossIMH: a Picross solver



Contents

Linear Programs

- In words:
  Optimize a linear function
  subject to linear constraints on real variables

- Explicit general formulation:

$$
\begin{aligned}
\max \mid \min \quad & c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \\
\text{subject to} \quad & a_{i1} x_1 + a_{i2} x_2 + \ldots + a_{in} x_n \leq b_i \quad (i = 1, \ldots, m_1) \\
& a_{i1} x_1 + a_{i2} x_2 + \ldots + a_{in} x_n \geq b_i \quad (i = m_1 + 1, \ldots, m_2) \\
& a_{i1} x_1 + a_{i2} x_2 + \ldots + a_{in} x_n = b_i \quad (i = m_2 + 1, \ldots, m) \\
& x_j \in \mathbb{R} \quad (j = 1, \ldots, n)
\end{aligned}
$$

Linear Integer Programs

- In words:
  Optimize a linear function
  subject to linear constraints on integer variables

- Explicit general formulation:

$$
\begin{aligned}
\max \mid \min \quad & c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \\
\text{subject to} \quad & a_{i1} x_1 + a_{i2} x_2 + \ldots + a_{in} x_n \leq b_i \quad (i = 1, \ldots, m_1) \\
& a_{i1} x_1 + a_{i2} x_2 + \ldots + a_{in} x_n \geq b_i \quad (i = m_1 + 1, \ldots, m_2) \\
& a_{i1} x_1 + a_{i2} x_2 + \ldots + a_{in} x_n = b_i \quad (i = m_2 + 1, \ldots, m) \\
& x_j \in \mathbb{Z} \quad (j = 1, \ldots, n)
\end{aligned}
$$

Mixed Integer Programs

- In words:
  Optimize a linear function
  subject to linear constraints on integer and/or real variables

- Explicit general formulation:

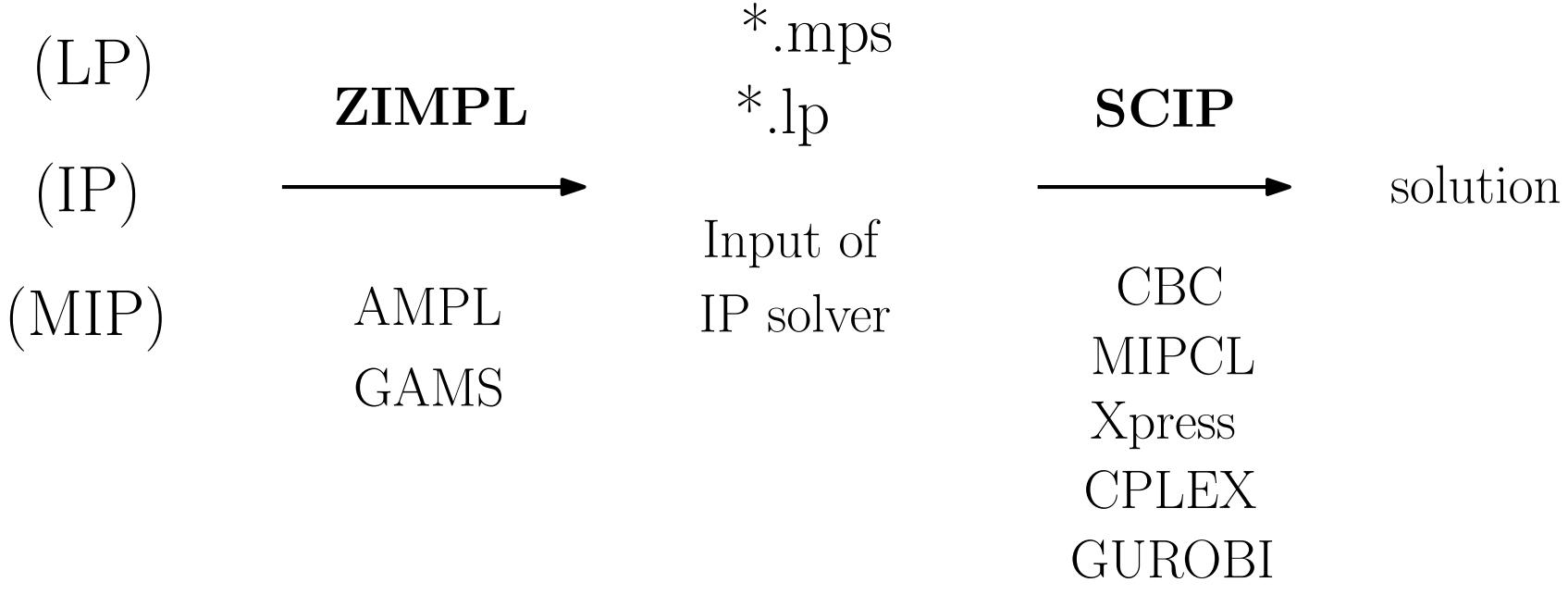
$$
\begin{aligned}
&\max \mid \min && c_1 x_1 + c_2 x_2 + \ldots + c_n x_n && \\
&\text{subject to} && a_{i1} x_1 + a_{i2} x_2 + \ldots + a_{in} x_n \leq b_i && (i = 1, \ldots, m_1) \\
& && a_{i1} x_1 + a_{i2} x_2 + \ldots + a_{in} x_n \geq b_i && (i = m_1 + 1, \ldots, m_2) \\
& && a_{i1} x_1 + a_{i2} x_2 + \ldots + a_{in} x_n = b_i && (i = m_2 + 1, \ldots, m) \\
& && x_j \in \mathbb{Z} \text{ or } \mathbb{R} && (j = 1, \ldots, n)
\end{aligned}
$$

From real-life problems ...

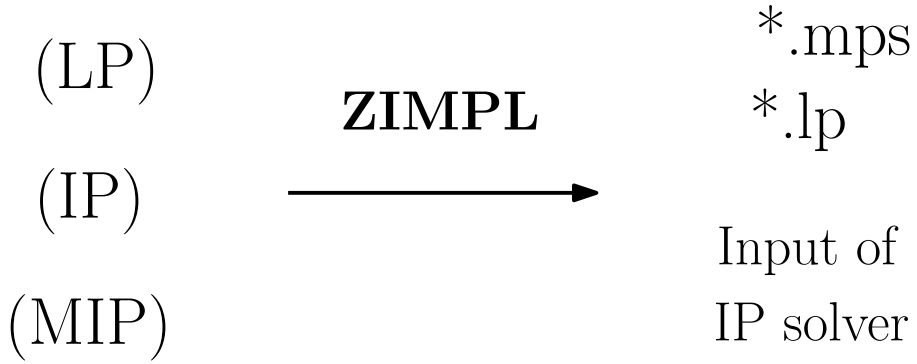
Many real-life problems can be modeled as

Linear Program (LP)	Integer Program (IP)	Mixed Integer Program (MIP)
$\min \mathbf{c}^t\mathbf{x}$	$\min \mathbf{c}^t\mathbf{x}$	$\min \mathbf{c}^t\mathbf{x}$
$\text{s.t. } A\mathbf{x} \leq \mathbf{b}$	$\text{s.t. } A\mathbf{x} \leq \mathbf{b}$	$\text{s.t. } A\mathbf{x} \leq \mathbf{b}$
$\mathbf{x} \in \mathbb{R}^n$	$\mathbf{x} \in \mathbb{Z}^n$	$\mathbf{x} \in \mathbb{Z}^p \times \mathbb{R}^q$

... to their solutions



What is ZIMPL?



Prof. Thorsten Koch

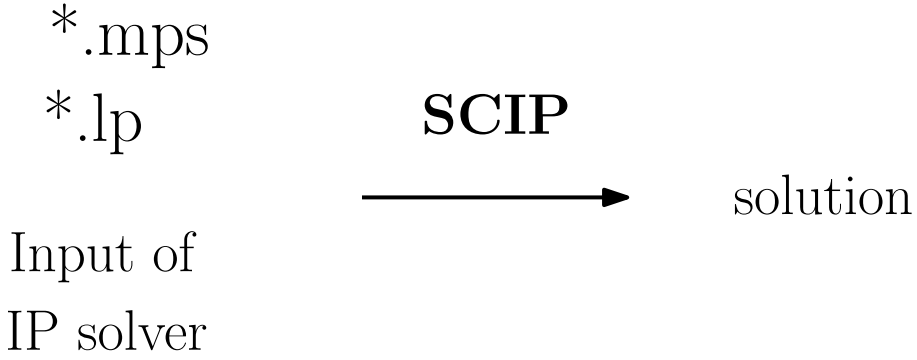
- ZIMPL[1]:
  - Zuse Institut Mathematical Programming Language
  - https://zimpl.zib.de/

[1] Thorsten Koch. *Rapid Mathematical Programming.* PhD thesis, Technische Universität Berlin, 2004

Why ZIMPL?

	ZIMPL	AMPL, GAMS
Distribution	free, open-source	commercial
Rational arithmetic	infinite precision	finite precision

What is SCIP?

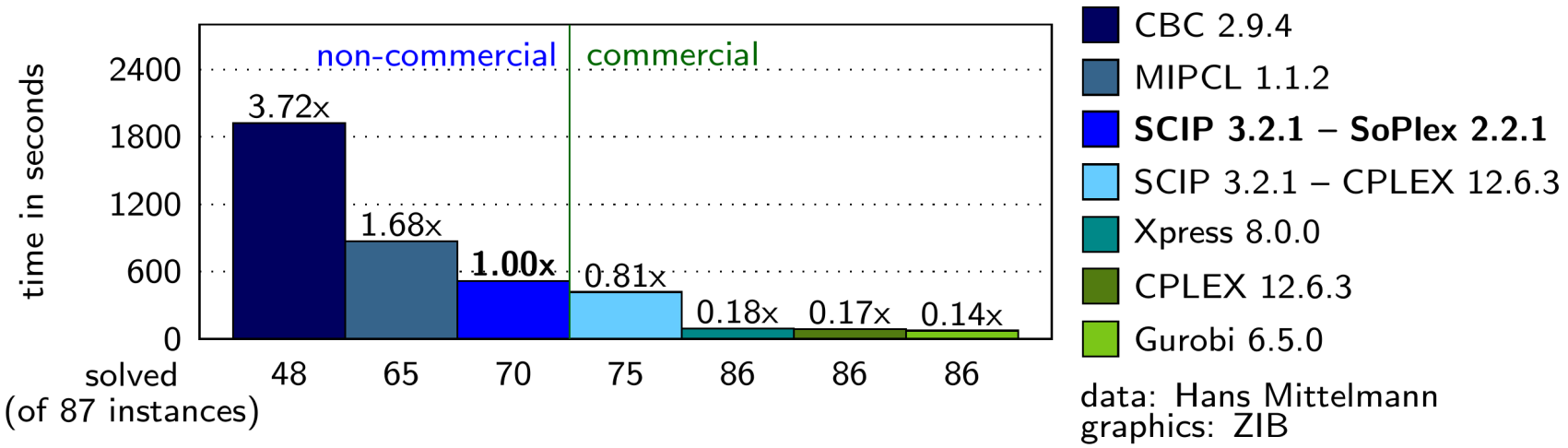


Dr. Tobias Achterberg

- SCIP[2]:
  - Solving Constraint Integer Programs
  - https://scip.zib.de/

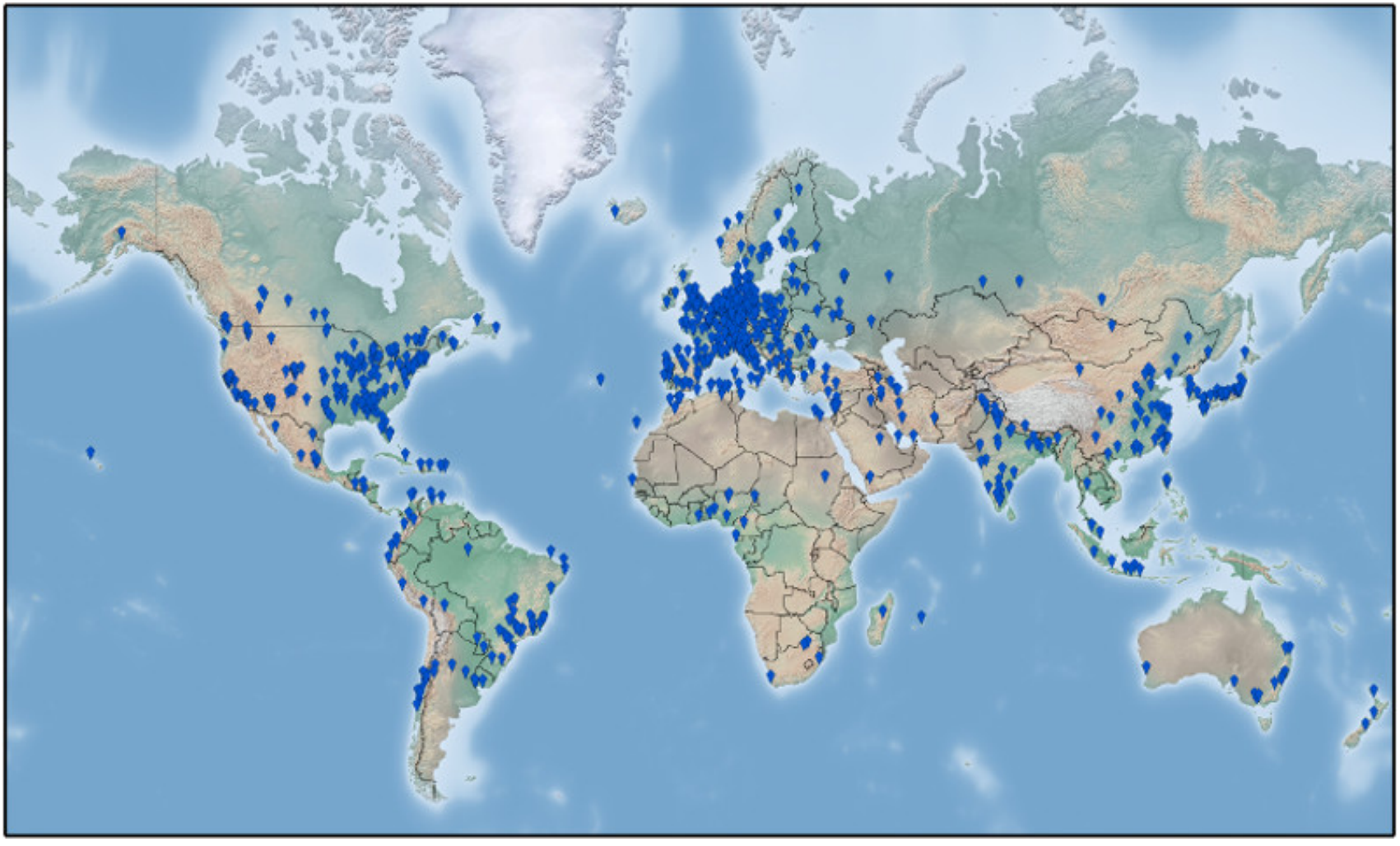
[2] Tobias Achterberg. *Constraint Integer Programming.* PhD thesis, Technische Universität Berlin, 2007

Why SCIP?



Source: https://scip.zib.de/

Why SCIP?



Locations of registered SCIP downloads.
Source: https://scip.zib.de/

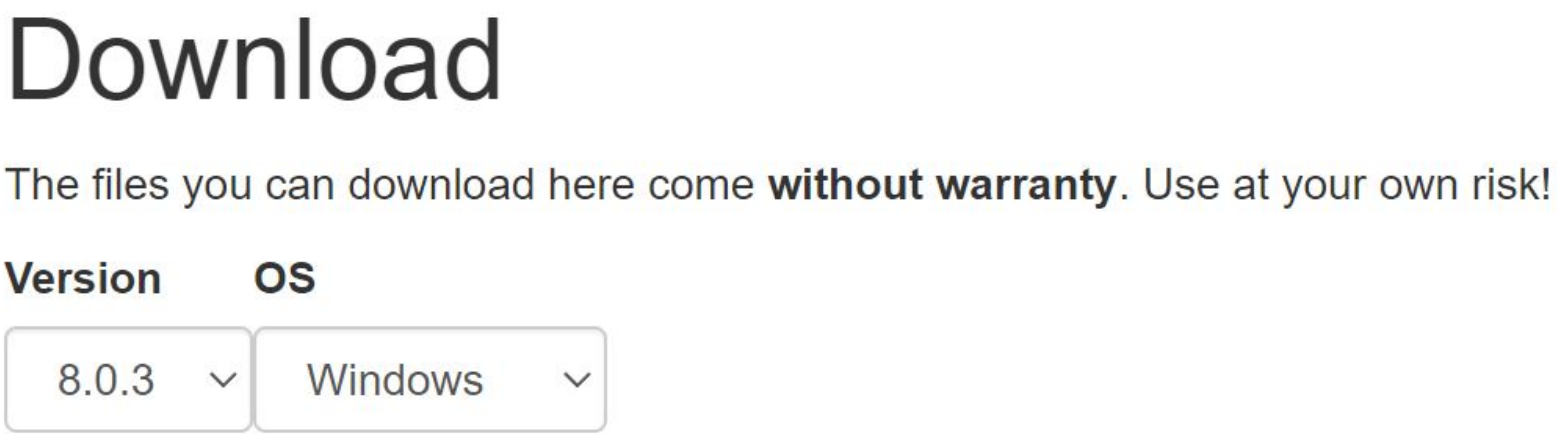
Contents

Download ZIMPL and SCIP

Step 1: Click

`https://www.scipopt.org/index.php#download`

then choose Version 8.0.3 (latest version) and OS Windows



Download ZIMPL and SCIP (cont.)

Step 2: Go to **Precompiled Packages** and choose

- SCIPOptSuite-8.0.3-win64-VS15.exe
  (for Windows 64 bits)
- SCIPOptSuite-8.0.3-win32-VS15.exe
  (for Windows 32 bits)

Precompiled Packages

You can also download precompiled executables of SCIP with which you can solve MIP, MIQCP, CIP, SAT, or PBO instances in MPS, LP, RLP, ZIMPL, flatzinc, CNF, OPB, WBO, PIP, or CIP format.
Note that these executables do not include the readline features (i.e., command line editing and history) due to license issues. However, you can download the free readline wrapper rlwrap to provide this missing feature to the executables.

Installers (install the scipoptsuite in your computer, without source files):

	SoPlex	ZIMPL	GCG	UG	PaPILO	Ipopt	CPPAD	zlib	bliss	MPIR
SCIPOptSuite-8.0.3-win64-VS15.exe Windows 64bit installer (MSVS15); requires the Visual C++ Redistributable Packages	6.0.3	3.5.3			2.1.2	3.12.9	20180000.0		0.77	3.0.0
SCIPOptSuite-8.0.3-win32-VS15.exe Windows 64bit installer (MSVS15); requires the Visual C++ Redistributable Packages	6.0.3	3.5.3			2.1.2	3.12.9	20180000.0		0.77	3.0.0

Download ZIMPL and SCIP (cont.)

Step 3: Register before downloading

Download

File name:	SCIPOptSuite-8.0.3-win64-VS15.exe
File size:	47949 kb
MD5 sum:	35fc8fd940295cf5e221b0fc5ecc9ece
Downloads:	0

Releases up to and including Version 8.0.2 remain under the ZIB Academic License as indicated by the files contained in the releases. The Apache 2.0 License applies from Version 8.0.3 onwards.

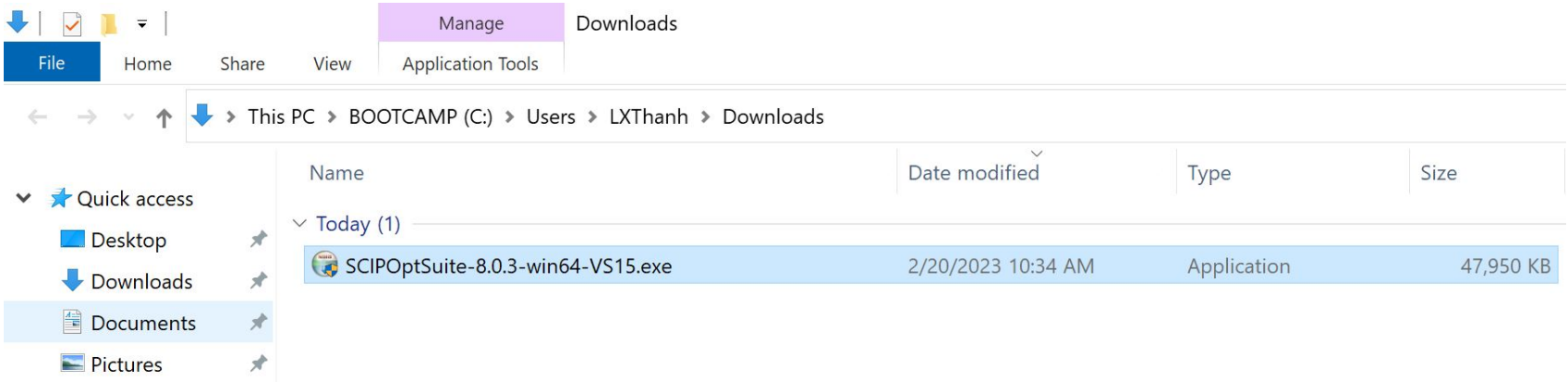
Please enter your information in the form below. Mandatory fields are marked with an (*) asterisk. The input data are only used for internal evaluation purposes.

Name*	Le Xuan Thanh
Email*	lxthanh@math.ac.vn
Institution	Institution
City	City
Country*	Vietnam

Start Download

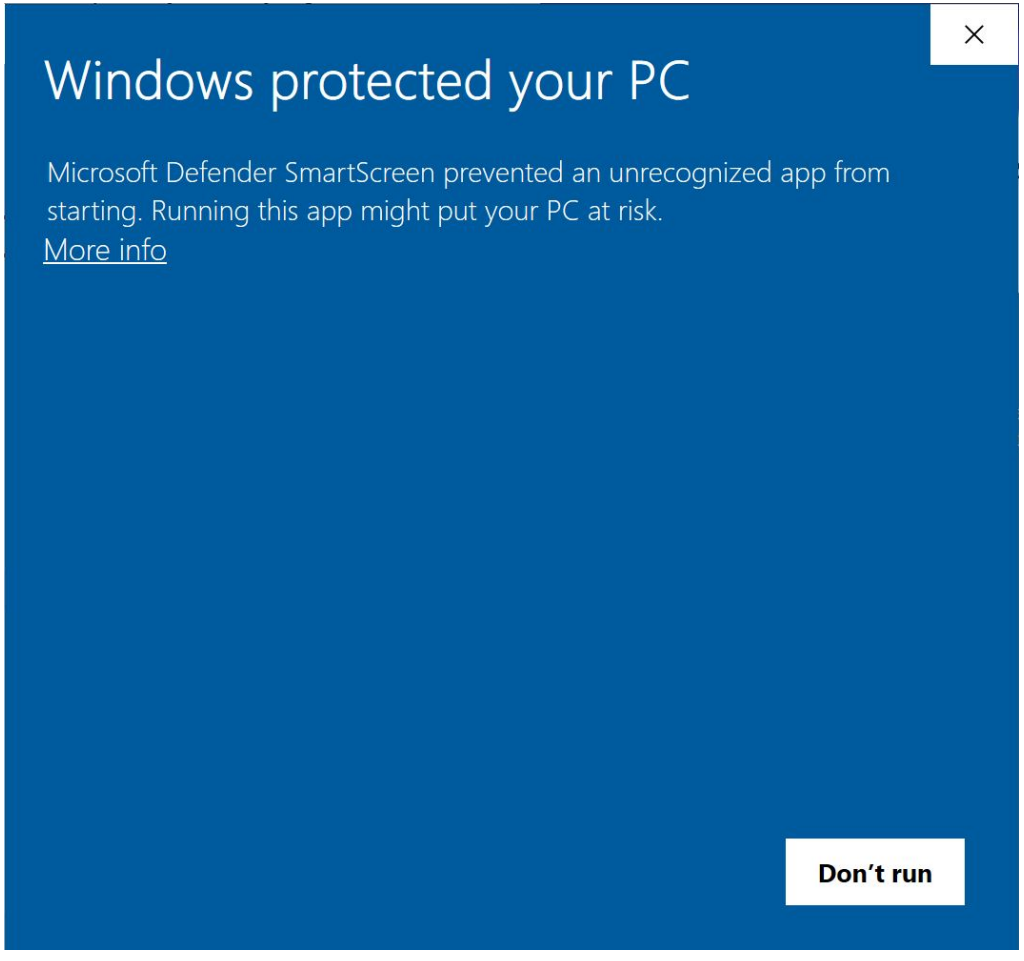
Install ZIMPL and SCIP

Step 4: Find the downloaded file and double click



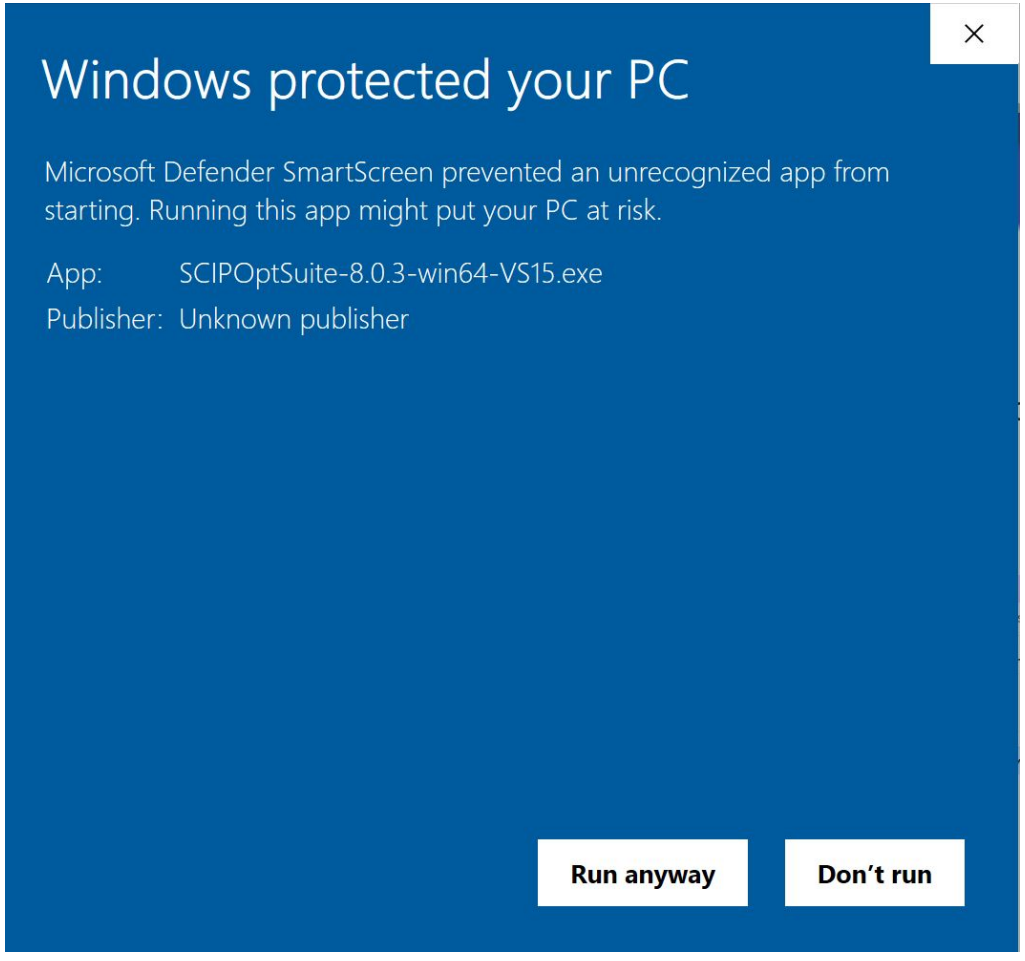
Install ZIMPL and SCIP (cont.)

Step 5: In case of unrecognized app:
choose More info then Run anyway
and follow default instruction



Install ZIMPL and SCIP (cont.)

Step 5: In case of unrecognized app:
choose More info then Run anyway
and follow default instruction

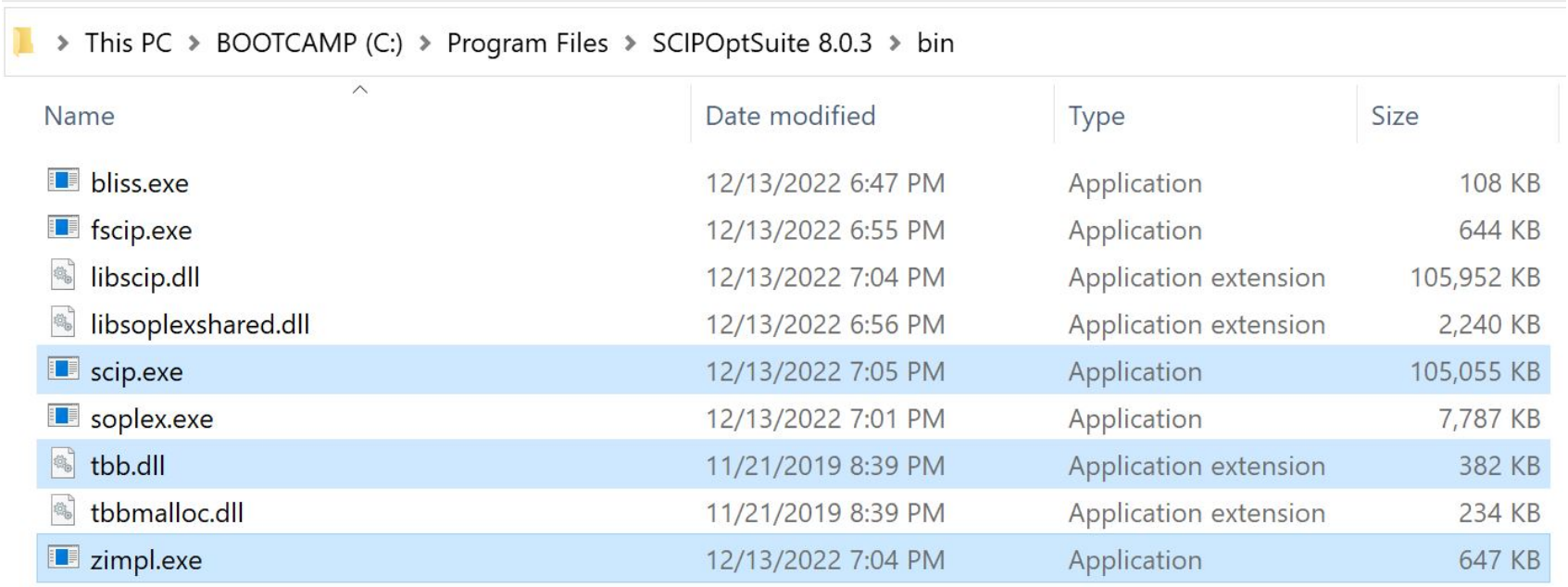


Install ZIMPL and SCIP (cont.)

Step 6: Go to

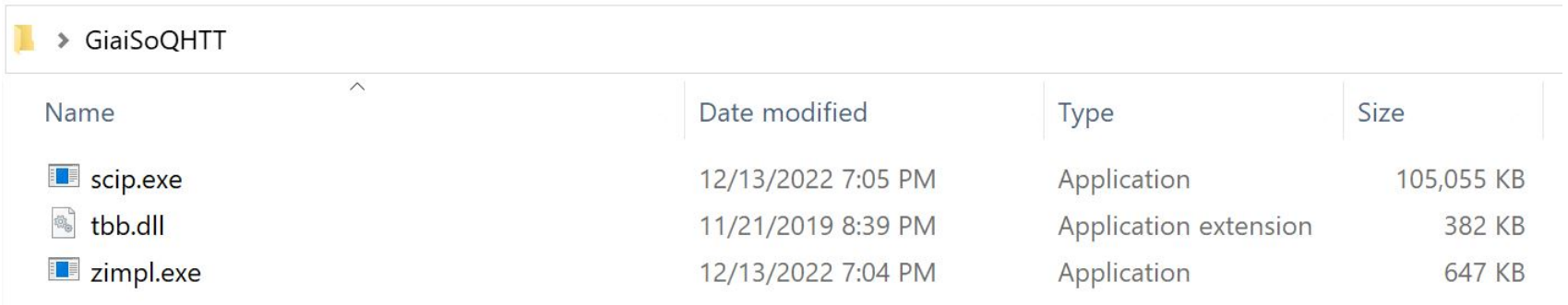
```
C:\Program Files\SCIPOptSuite 8.0.3\bin
```

and copy 3 files: scip.exe, tbb.dll, zimpl.exe



Install ZIMPL and SCIP (cont.)

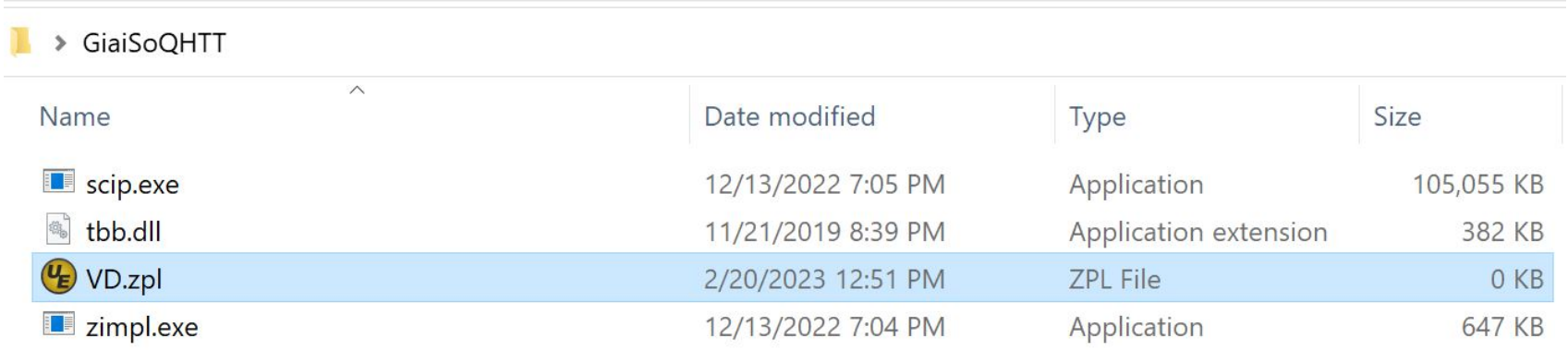
Step 7: Create a folder (should be on Desktop)
and paste 3 files: scip.exe, tbb.dll, zimpl.exe into this folder



Contents

Procedure

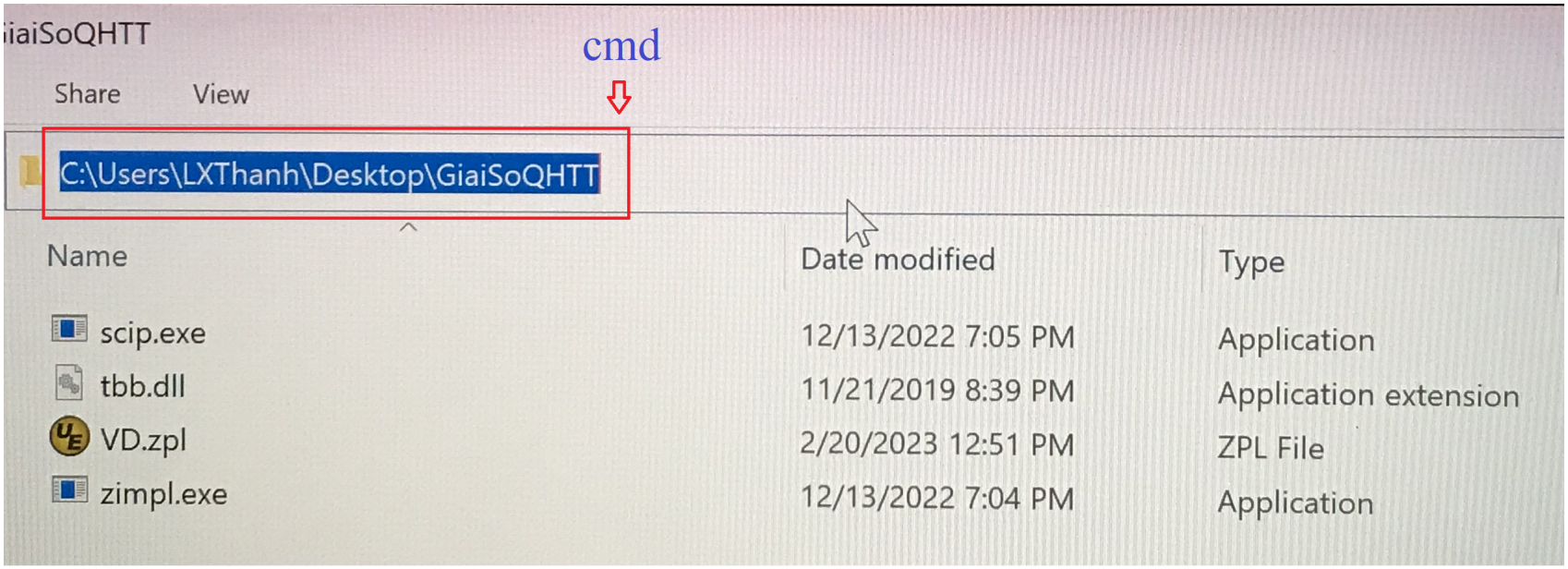
- **Step 1:**
  In the folder containing 3 files scip.exe, tbb.dll, zimpl.exe, create a text file *.zpl in which * is the file name



- **Step 2:**
  In file *.zpl, write down the ZIMPL code for the linear program to be solved
  (see examples later)

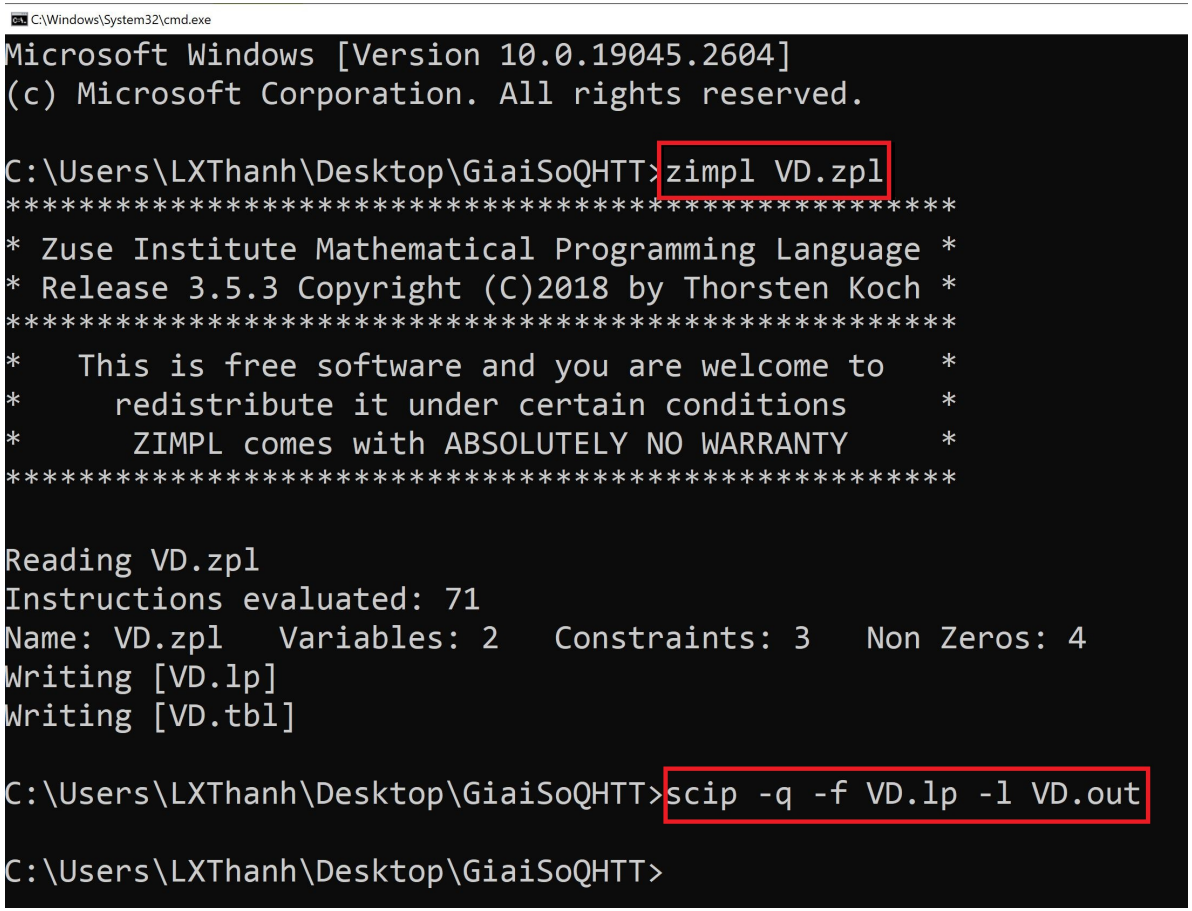
Procedure (cont.)

- **Step 3:**
  Go to the title box of the folder, type cmd then hit Enter (to open Command Prompt terminal)



Procedure (cont.)

- **Step 4:**
  In Command Prompt terminal, type the commands
  zimpl *.zpl then scip -q -f *.lp -l *.out
  in which * is the file name



Procedure (cont.)

- **Step 5:**
  Find solution in file *.out in the same folder

VD.out

```
34
35 SCIP Status        : problem is solved [optimal solution found]
36 Solving Time (sec) : 0.00
37 Solving Nodes      : 0
38 Primal Bound       : +1.9200000000000e+03 (2 solutions)
39 Dual Bound         : +1.9200000000000e+03
40 Gap                : 0.00 %
41
42 primal solution (original space):
43 ================================
44
45 objective value:                                1920
46 x1                                                30   (obj:50)
47 x2                                                10   (obj:42)
```

Contents

Example 1

Solve

$$
\begin{aligned}
\max \quad & 50x_1 + 42x_2 \\
\text{s.t.} \quad & x_1 + x_2 \leq 40 \\
& x_1 \leq 30 \\
& x_2 \leq 30 \\
& x_1 \geq 0 \\
& x_2 \geq 0
\end{aligned}
$$

ZIMPL code

Program:

$$\begin{aligned} \max \quad & 50x_1 + 42x_2 \\ \text{subject to} \quad & \\ & x_1 + x_2 \leq 40 \\ & x_1 \leq 30 \\ & x_2 \leq 30 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{aligned}$$

ZIMPL code:

```
# Variables
var x1 >= 0;
var x2 >= 0;

# Objective
maximize profit:
50 * x1 + 42 * x2;

# Constraints

subto Constraint1:
  x1 + x2 <= 40;

subto Constraint2:
  x1 <= 30;

subto Constraint3:
  x2 <= 30;
```

Structure of ZIMPL code

Code:

```
# Variables
var x1 >= 0;
var x2 >= 0;

# Objective
maximize profit:
50 * x1 + 42 * x2;

# Constraints

subto Constraint1:
  x1 + x2 <= 40;

subto Constraint2:
  x1 <= 30;

subto Constraint3:
  x2 <= 30;
```

Structure:

- Comment lines start with #
- Declare variables
  var [Declare variables];
- Objective:
  maximize [Name]: [Objective];
  or
  minimize [Name]: [Objective];
- Constrains:
  subto [Name]: [Constraints];

Remark:
Names of variables, objective, constraints start with alphabet characters

Results obtained by SCIP

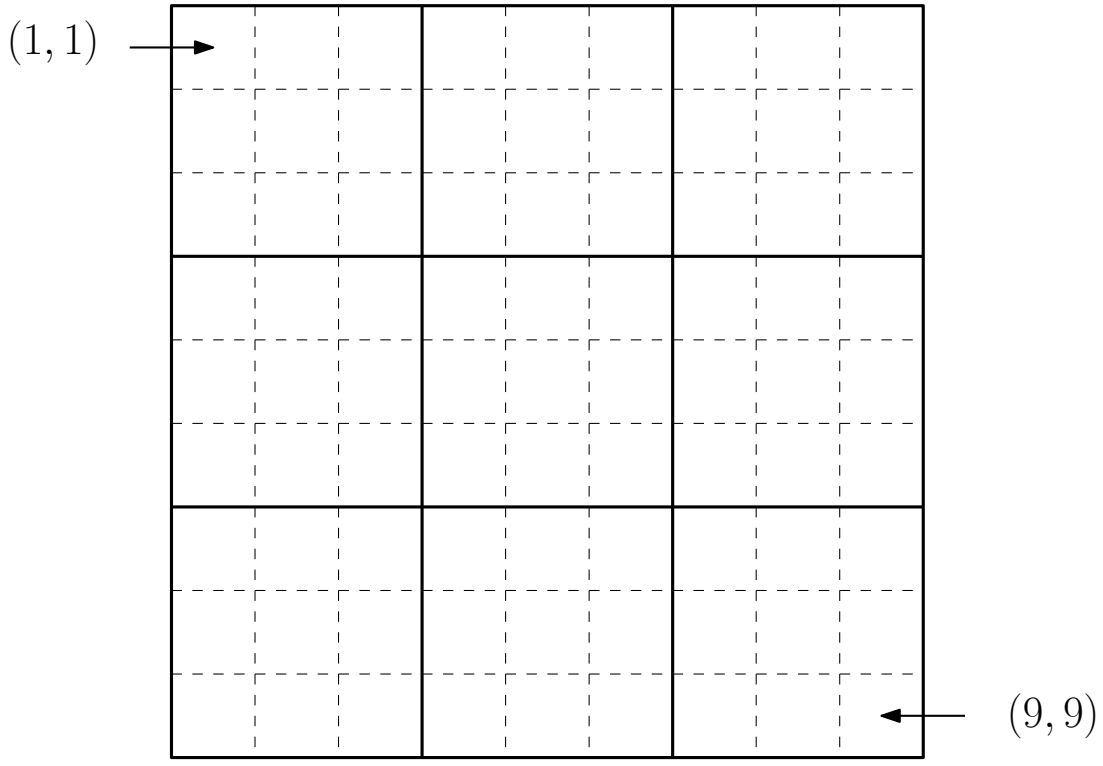
VD.out

```
34
35 SCIP Status        : problem is solved [optimal solution found]
36 Solving Time (sec) : 0.00
37 Solving Nodes      : 0
38 Primal Bound       : +1.9200000000000e+03 (2 solutions)
39 Dual Bound         : +1.9200000000000e+03
40 Gap                : 0.00 %
41
42 primal solution (original space):
43 ================================
44
45 objective value:                                1920
46 x1                                               30   (obj:50)
47 x2                                               10   (obj:42)
```

Contents

Contents

Variables



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

A mathematical formulation for Sudoku

$$G = \{(i,j,k) \mid k \text{ is already filled into cell } (i,j)\}$$
$$S = \{1,\ldots,9\}, \quad N = \{1,\ldots,3\}$$

Formulations:

$$x_{ijk} = 1 \qquad \forall (i,j,k) \in G$$

$$\sum_{k=1}^{9} x_{ijk} = 1 \qquad \forall i \in S, j \in S$$

$$\sum_{i=1}^{9} x_{ijk} = 1 \qquad \forall j \in S, k \in S$$

$$\sum_{j=1}^{9} x_{ijk} = 1 \qquad \forall i \in S, k \in S$$

$$\sum_{j=3q-2}^{3q} \sum_{i=3p-2}^{3p} x_{ijk} = 1 \qquad \forall k \in S, p \in N, q \in N$$

$$x_{ijk} \in \{0,1\} \qquad \forall i \in S, j \in S, k \in S$$

Data file

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

#	Row	Column	Clue
1		7	8
1		8	9
2		3	1
2		4	3
2		6	9
3		2	5
3		7	2
3		9	4
4		6	2
4		7	6
4		8	8
5		1	9
5		4	7
5		6	6
5		9	5
6		2	3
6		3	6
6		4	5
7		1	1
7		3	3
7		8	5
8		4	8
8		6	7
8		7	1
9		2	4
9		3	5

ZIMPL model

```
# Auxiliary sets
set N := {1 .. 3};
set S := {1 .. 9};

# Import Sudoku clues from input data file
param fileClue := "Sudoku_Clues.dat";
set G := {read fileClue as "<1n, 2n, 3n>" comment "#"};

# Variables
var x[S * S * S] binary;

# Constraints

# For the already-filled cells in the grid
subto clues:
  forall <i, j, k> in G do x[i, j, k] == 1;

# One cell one value
subto cells:
  forall <i, j> in S * S do sum <k> in S: x[i, j, k] == 1;

# Each value appears once on each column
subto columns:
  forall <j, k> in S * S do sum <i> in S: x[i, j, k] == 1;

# Each value appears once on each row
subto rows:
  forall <i, k> in S * S do sum <j> in S: x[i, j, k] == 1;

# Each value appears once on each block
subto blocks:
  forall <p, q, k> in N * N * S do
    sum <i, j> in N * N: x[3*p-3+i, 3*q-3+j, k] == 1;
```

(First part of) result obtained by SCIP

```
solution status: optimal solution found
objective value:   0
x#1#1#3              1  (obj:0)
x#1#2#6              1  (obj:0)
x#1#3#7              1  (obj:0)
x#1#4#4              1  (obj:0)
x#1#5#2              1  (obj:0)
x#1#6#5              1  (obj:0)
x#1#7#8              1  (obj:0)
x#1#8#9              1  (obj:0)
x#1#9#1              1  (obj:0)
x#2#1#4              1  (obj:0)
x#2#2#2              1  (obj:0)
x#2#3#1              1  (obj:0)
x#2#4#3              1  (obj:0)
x#2#5#8              1  (obj:0)
x#2#6#9              1  (obj:0)
x#2#7#5              1  (obj:0)
x#2#8#7              1  (obj:0)
x#2#9#6              1  (obj:0)
x#3#1#8              1  (obj:0)
x#3#2#5              1  (obj:0)
x#3#3#9              1  (obj:0)
x#3#4#6              1  (obj:0)
x#3#5#7              1  (obj:0)
x#3#6#1              1  (obj:0)
x#3#7#2              1  (obj:0)
x#3#8#3              1  (obj:0)
x#3#9#4              1  (obj:0)
```

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

Contents

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

Thank you for your attention!