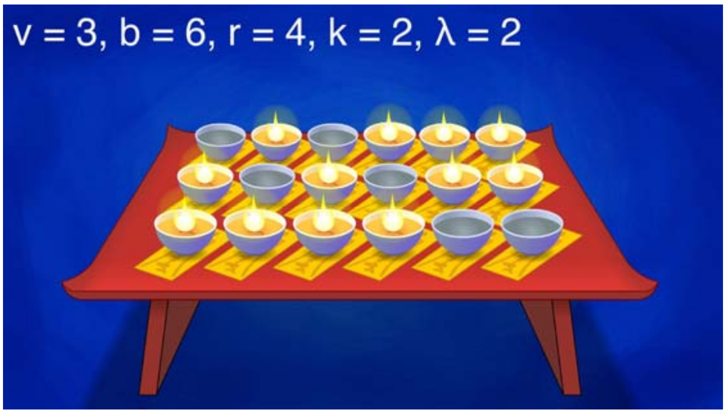
Matrix Model Symmetries

Jimmy Lee & Peter Stuckey

Lamps of the Big Dipper

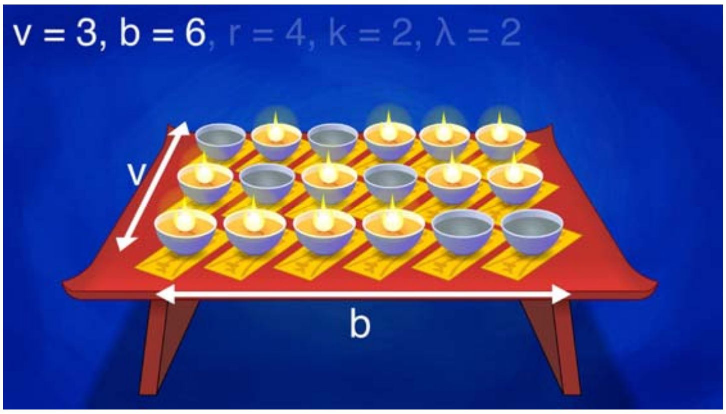
2

Five Parameter Requirements



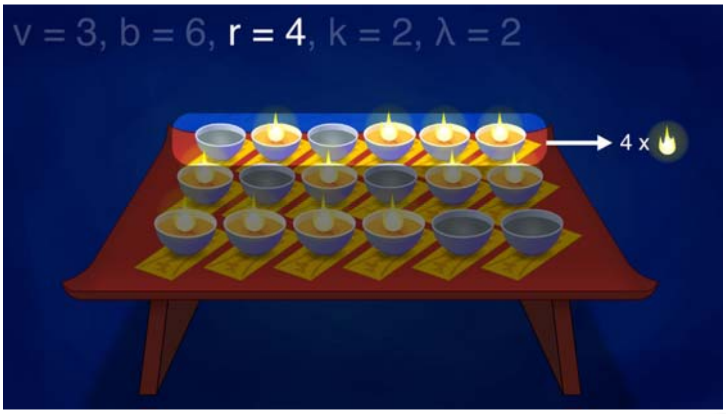
3

Size



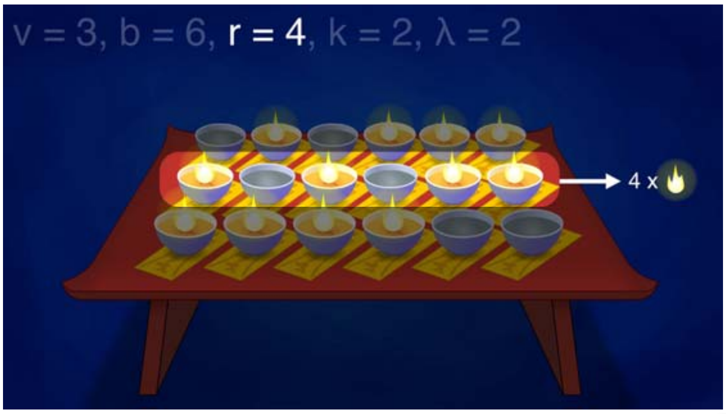
4

Lit Lamps on Each Row



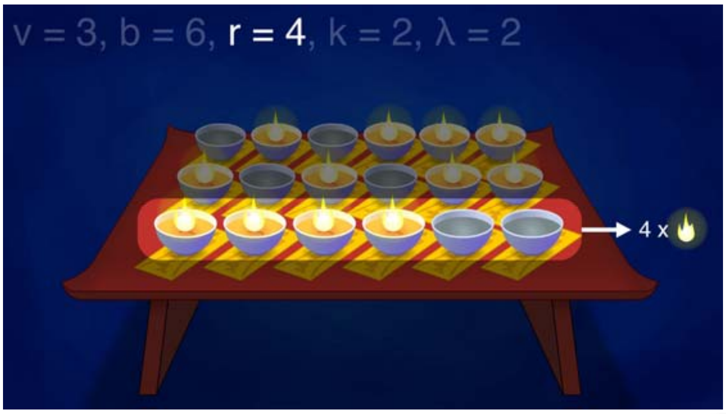
5

Lit Lamps on Each Row

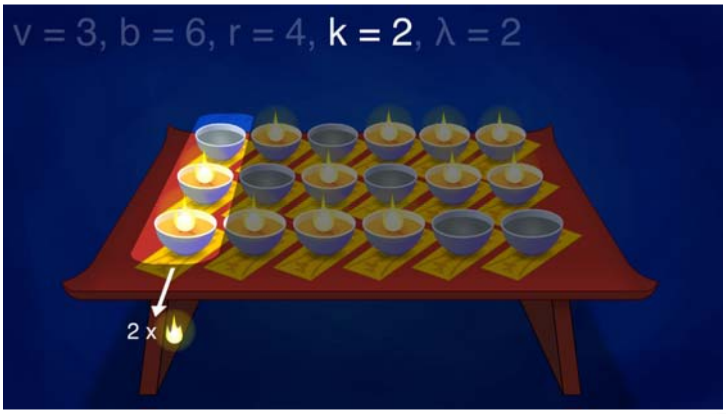


6

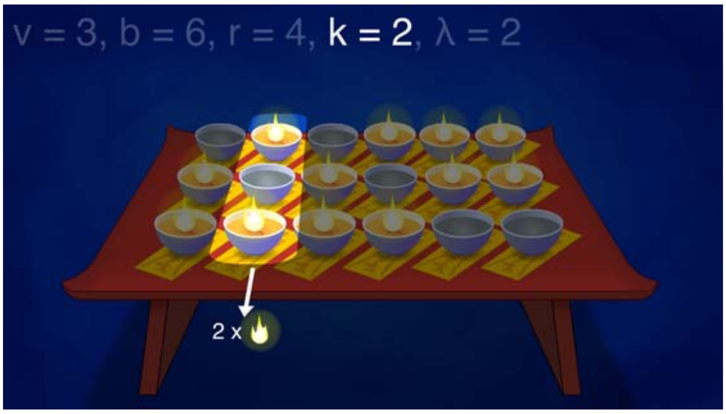
Lit Lamps on Each Row



Lit Lamps on Each Column

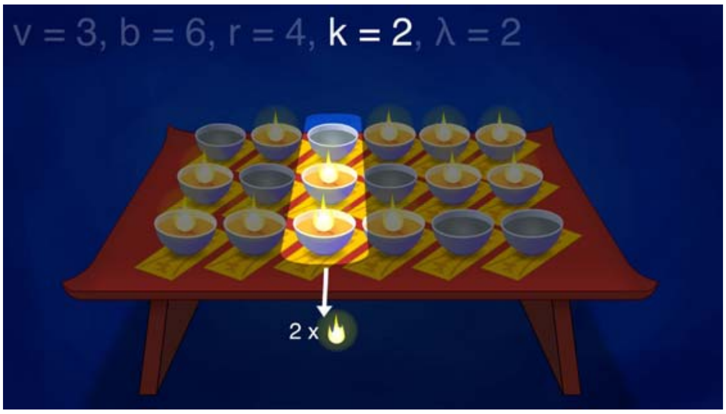


Lit Lamps on Each Column



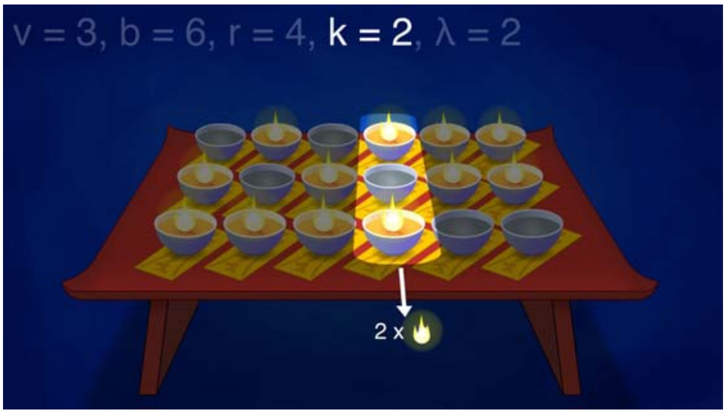
9

Lit Lamps on Each Column



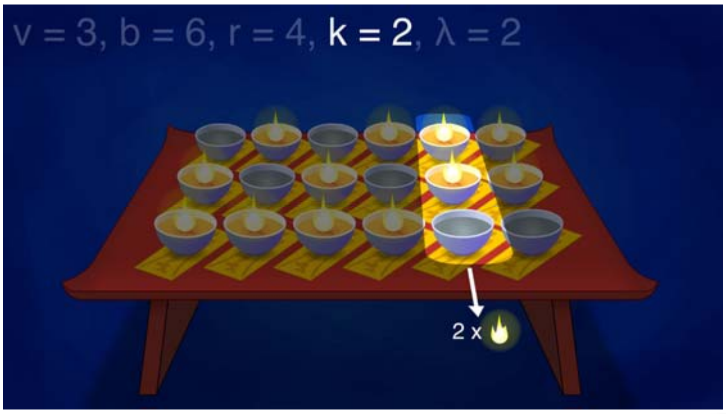
10

Lit Lamps on Each Column



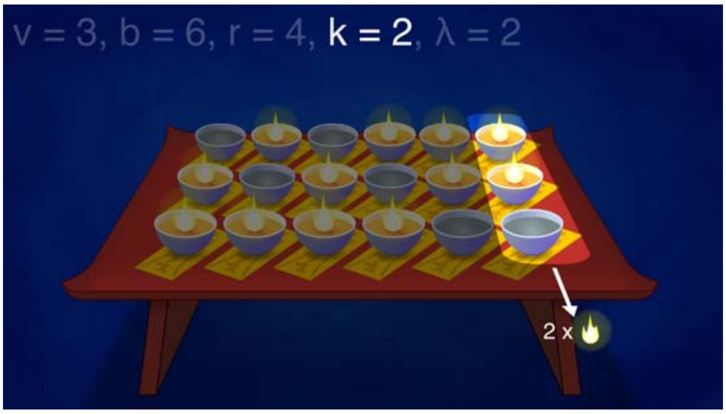
11

Lit Lamps on Each Column



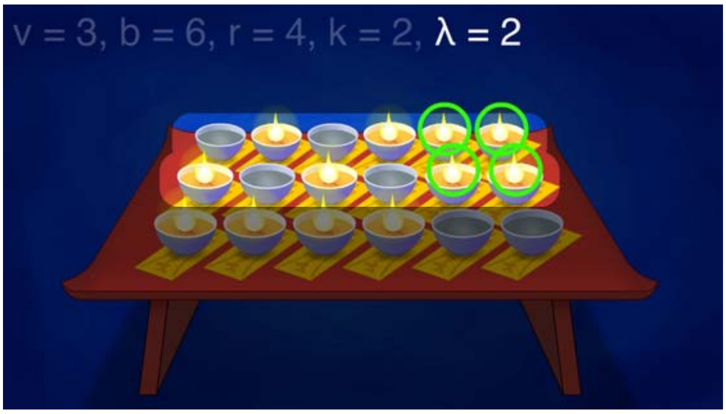
12

Lit Lamps on Each Column



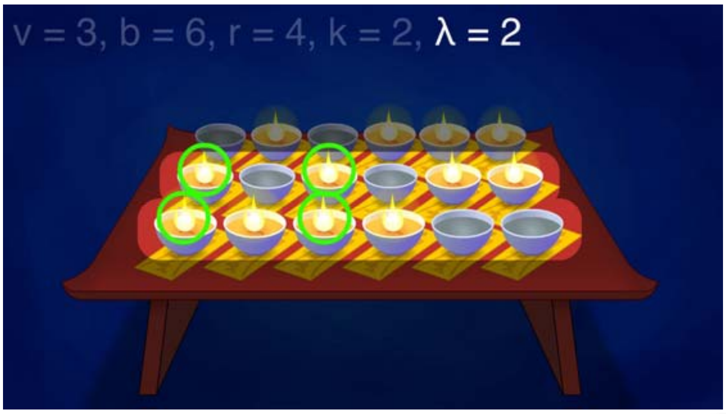
13

Common Lit Lamps Between 2 Rows



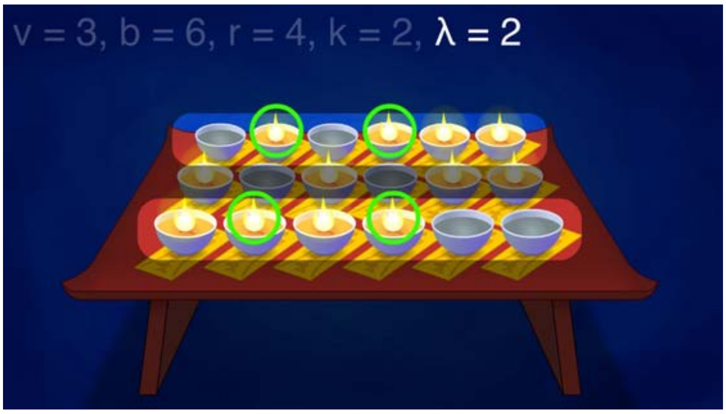
14

Common Lit Lamps Between 2 Rows



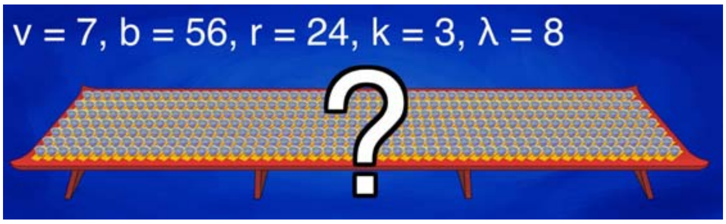
15

Common Lit Lamps Between 2 Rows



16

The Tao of Peace



17

The Lamp Lighting Problem

- In order to stop the rain before the Chibi war, Zhuge Liang decided to lit the Lamps of the Big Dipper with the following requirements:
  - the lamps are arranged in a *v* x *b* matrix
  - each row has exactly *r* lit lamps
  - each column has exactly *k* lit lamps
  - between any two distinct rows, the number of columns containing two lit lamps is λ

18

The Lamp Model (lamp.mzn)

- Data

```
int: v;
set of int: ROW = 1..v;
int: b;
set of int: COL = 1..b;
int: r;
int: k;
int: lambda;
```

- Decisions: which lamps are lit

```
array[ROW,COL] of var bool: m;
solve satisfy;
```

19

The Lamp Model (lamp.mzn)

- Every row has *r* lit lamps

```
forall(i in ROW)(sum(j in COL)(m[i,j]) = r);
```

- Every column has *k* lit lamps

```
forall(j in COL)(sum(i in ROW)(m[i,j]) = k);
```

- The number of common lit lamp positions in any two rows is λ

```
forall(i1, i2 in ROW where i1 < i2)
      (sum(j in COL)
          (m[i1,j] /\ m[i2,j]) = lambda);
```

20

Running the Lamp Model

- With the data file

```
v = 7;
b = 56;
r = 24;
k = 3;
lambda = 8;
```

- No solution in **10** minutes!

- What's the problem?
  - Too many symmetries!

21

Matrix Symmetries

- Interchangeable rows
  - swapping any number of rows in a lamp solution gives another solution

- For example

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T

A	B	C	D	E
P	Q	R	S	T
K	L	M	N	O
F	G	H	I	J

22

Matrix Symmetries

- Interchangeable columns
  - swapping any number of columns in a lamp solution gives another solution
- For example

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T

C	B	A	D	E
H	G	F	I	J
M	L	K	N	O
R	Q	P	S	T

23

Matrix Symmetries

- Composition of symmetries
  - swapping any number of columns, then swapping any number of rows, is also a solution
- For example

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T

C	B	A	D	E
R	Q	P	S	T
M	L	K	N	O
H	G	F	I	J

24

LexLeader Symmetry Breaking

- One lex leader constraint per symmetry

A	B	C
D	E	F

1. ABCDEF $\leq_{lex}$ ABCDEF
2. ABCDEF $\leq_{lex}$ ACBDFE
3. ABCDEF $\leq_{lex}$ BACEDF
4. ABCDEF $\leq_{lex}$ CBAFED
5. ABCDEF $\leq_{lex}$ BCAEFD
6. ABCDEF $\leq_{lex}$ CABFDE
7. ABCDEF $\leq_{lex}$ DEFABC
8. ABCDEF $\leq_{lex}$ DFEACB
9. ABCDEF $\leq_{lex}$ EDFBAC
10. ABCDEF $\leq_{lex}$ FEDCBA
11. ABCDEF $\leq_{lex}$ EFDBCA
12. ABCDEF $\leq_{lex}$ FDECAB

LexLeader Symmetry Breaking

- Totally *n*!*m*! number of symmetries for a *nxm* matrix
- Breaking all symmetries requires an **exponential** number (*n*!*m*!) of LexLeader constraints
- Too **many** constraints to add and handle!
- We can choose only a subset of symmetries to break

LexLeader Symmetry Breaking

- One lex leader constraint per symmetry

A	B	C
D	E	F

1. ABCDEF $\leq_{lex}$ ABCDEF
2. ABCDEF $\leq_{lex}$ ACBDFE
3. ABCDEF $\leq_{lex}$ BACEDF
4. ABCDEF $\leq_{lex}$ CBAFED
5. ABCDEF $\leq_{lex}$ BCAEFD
6. ABCDEF $\leq_{lex}$ CABFDE
7. ABCDEF $\leq_{lex}$ DEFABC
8. ABCDEF $\leq_{lex}$ DFEACB
9. ABCDEF $\leq_{lex}$ EDFBAC
10. ABCDEF $\leq_{lex}$ FEDCBA
11. ABCDEF $\leq_{lex}$ EFDBCA
12. ABCDEF $\leq_{lex}$ FDECAB.

Partial LexLeader Symmetry Breaking

- One lex leader constraint per selected symmetry

A	B	C
D	E	F

- ABCDEF $\leq_{lex}$ DEFABC
- ABCDEF $\leq_{lex}$ ACBDFE
- ABCDEF $\leq_{lex}$ BACEDF

- Simplify the constraints, e.g.
  - ABCDEF $\leq_{lex}$ ACBDFE
  - BCEF $\leq_{lex}$ CBFE removing same positions
  - BE $\leq_{lex}$ CF XY $\leq_{lex}$ YX $\Rightarrow$ X $\leq$ Y

Partial LexLeader Symmetry Breaking

- One lex leader constraint per selected symmetry

A	B	C
D	E	F

- ABCDEF $\leq_{lex}$ DEFABC $\Leftrightarrow$ ABC $\leq_{lex}$ DEF
- ABCDEF $\leq_{lex}$ ACBDFE $\Leftrightarrow$ BE $\leq_{lex}$ CF
- ABCDEF $\leq_{lex}$ BACEDF $\Leftrightarrow$ AD $\leq_{lex}$ BE
- Does not break all symmetries, e.g.
  - ABCDEF = 011100
  - Now 011 $\leq_{lex}$ 100, 10 $\leq_{lex}$ 10, 01 $\leq_{lex}$ 10
  - but not $\leq_{lex}$ 001110 = FEDCBA

29

DoubleLex Symmetry Breaking

- Simply require
  - adjacent rows to be in lexicographic order

A	B	C	$\leq_{lex}$	D	E	F

  - adjacent columns to be in lexicographic order

A	$\leq_{lex}$	B	$\leq_{lex}$	C
D		E		F

- ABCDEF $\leq_{lex}$ DEFABC $\Leftrightarrow$ ABC $\leq_{lex}$ DEF
- ABCDEF $\leq_{lex}$ ACBDFE $\Leftrightarrow$ BE $\leq_{lex}$ CF
- ABCDEF $\leq_{lex}$ BACEDF $\Leftrightarrow$ AD $\leq_{lex}$ BE

30

DoubleLex Symmetry Breaking

- Add the symmetry breaking to the model `(lamp-sym.mzn)`

```
include "lex_lesseq.mzn";
forall(i in 1..v-1)
      (lex_lesseq([ m[i,j] | j in COL],
                  [ m[i+1,j] | j in COL]));
forall(j in 1..b-1)
      (lex_lesseq([ m[i,j] | i in ROW],
                  [ m[i,j+1] | i in ROW]));
```

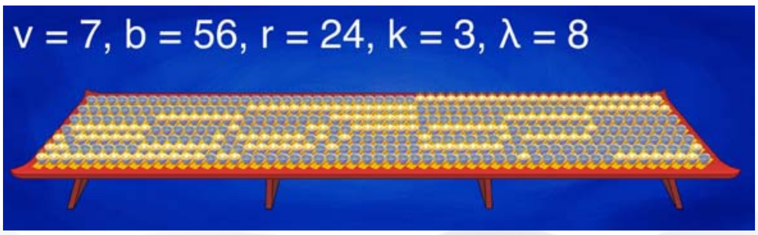
- Or use a global `(lamp-dl.mzn)`

```
include "double_lex.mzn";
double_lex(m);
```

31

DoubleLex Symmetry Breaking

- Does not break all $v! \times b!$ symmetries
  - but breaks sufficiently many
- With the same data file as before
- Solution in 7s



32

DoubleLex Symmetry Breaking

- Does not break all *v*! × *b*! symmetries
  - but breaks sufficiently many
- With the same data file as before
- Solution in 7s

v = 7, b = 56, r = 24, k = 3, λ = 8

```
00000000000000000000000000000000111111111111111111111111
00000000000000001111111111111111000000000000000011111111
00000000111111110000000011111111000000001111111100000000
01111111000000010000001100111111011111110000000100000000
10111111000000101111110101000000100000010111111000000000
11000000111111100111111010000000111111100000000000000001
11111111111111011000000000000000000000001000000011111110
```

33

Summary

- Matrix problems
  - where the answer is a 2D matrix of values
- Often have row and column symmetries
- Usually too expensive to break all
- Global constraint `double_lex`
  - is efficient and breaks many symmetries
- The Lamp model is actually the Balanced Incomplete Block Design (BIBD) in disguise
  - an important problem in experiment design

34

Summary

- Symmetry breaking is a double edge sword
  - Pros: can drastically reduce search space
  - Cons: symmetry breaking constraints can become overheads to slow down computations
- Especially in the case when we want only one solution
  - solving might be slower with symmetry breaking
  - search strategy becomes more important than size of search space in deciding the solving efficiency
  - the important topic of “search strategy” will be discussed in detail in the future

35

Image Credits

All graphics by Marti Wong, ©The Chinese University of Hong Kong and the University of Melbourne 2016

36