Friday, October 29

Program of the day:

- Efficient solution of problems
- The simplex algorithm is not efficient
- Convex hull and totally unimodular (TU) matrices
- Good and bad formulations (Williams chap. 10.1)
- Simplifying an IP model (Williams chap. 10.2)
- Applications: Three-dimensional noughts and crosses

Efficient solution of problems

- Efficient algorithm: bounded by a polynomium

$$n^3 + n^2, n^{100}, \sin(n) n^5$$

- Not efficient algorithm:

$$2^n, n!$$

Moore: speed of computers get doubled every second year

- Efficient algorithm $f(n) = n^3$

$$2 \cdot f(n) = 2 \times n^3 = \left(\sqrt[3]{2} n\right)^3 = f\left(\sqrt[3]{2} \cdot n\right)$$

multiplicative increase (exponential growth)

- Exponential algorithm $f(n) = 2^n$

$$2 \cdot f(n) = 2 \times 2^n = 2^{n+1} = f(n+1)$$

additive increase (linear growth)

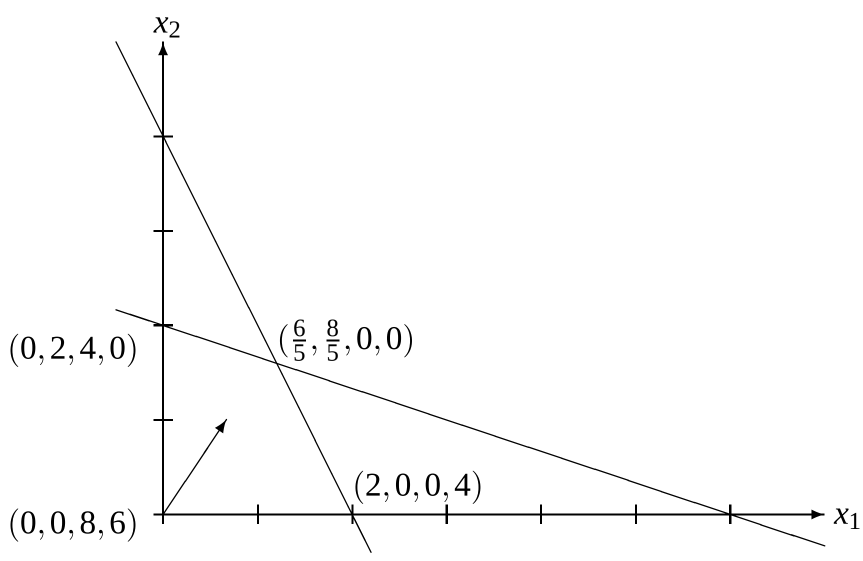
Linear Programming

$$\begin{array}{lrcrcl} \text{maximize} & 2x_1 & + & 3x_2 & & \\ \text{subject to} & 4x_1 & + & 2x_2 & \leq & 8 \\ & x_1 & + & 3x_2 & \leq & 6 \\ & & & x_1, x_2 & \geq & 0 \end{array}$$

Add slack variables

$$\begin{array}{lrcrcrcrcl} \text{maximize} & 2x_1 & + & 3x_2 & & & & & & \\ \text{subject to} & 4x_1 & + & 2x_2 & + & x_3 & & & = & 8 \\ & x_1 & + & 3x_2 & & & + & x_4 & = & 6 \\ & & & & & & x_1, x_2, & x_3, x_4 & \geq & 0 \end{array}$$

The set of constraints form a polyhedral.
Optimal solution is found at extreme points



Extreme points:

$$\begin{array}{ccc} (0,0,8,6) & (0,4,0,-6) & (0,2,4,0) \\ (2,0,0,4) & (6,0,-16,0) & (\frac{6}{5},\frac{8}{5},0,0) \end{array}$$

Extreme point

- Extreme points appear by setting $n-m$ variables to 0 and solving the remaining m equations with m variables to optimality.
- Choose m linearly independant columns in A. The corresponding set $B=\{i_1,i_2,\ldots,i_m\}$ is called a *basis*.
- A simple algorithm: Search through all extreme points

  Basis can be chosen in $\binom{n}{m}$ ways (i.e. exponential).

- Two basis feasible solutions x^1 and x^2 are adjacent if B^1 and B^2 have $m-1$ common elements.
- *Simplex algorithm* is a greedy algorithm which works as follows: Move from basis feasible solution to adjacent basis feasible solution such that objective function is ”increased most possible” in each step.
  - Initial solution
  - Iterative step
  - Optimality criteria

Simplex in Matrix Form (Taha Chapter 6)

LP-model

$$\begin{array}{ll} \text{maximize} & cx \\ \text{subject to} & Ax \leq b \\ & x \geq 0 \end{array}$$

Formulation after adding slack variables (new A, c, x)

$$\begin{array}{ll} \text{maximize} & cx \\ \text{subject to} & Ax = b \\ & x \geq 0 \end{array}$$

If we have m constraints, the Simplex algorithm chooses m linearly independant columns in A (the basis). The corresponding variables are x_B the remaining variables x_N

$$\begin{array}{ll} \text{maximize} & c_B x_B + c_N x_N \\ \text{subject to} & A_B x_B + A_N x_N = b \\ & x_B, x_N \geq 0 \end{array}$$

Solve for x_B

$$x_B = A_B^{-1}(b - A_N x_N)$$

setting the non-basis variables to zero $x_N = 0$ we get

$$x_B = A_B^{-1} b$$

which is a basis solution. Objective function

$$c_B x_B = c_B A_B^{-1} b$$

Complexity of Simplex

Klee and Minty (1975) proved that the Simplex algorithm may use exponential time

$$
\begin{array}{l}
\text{maximize} \\
2^{n-1}x_1 \;+\; 2^{n-2}x_2 \;+\; \dots \;+\; 2x_{n-1} \;+\; 1x_n \\
\text{subject to} \\
\begin{array}{rcrcrcrcrcl}
1x_1 & + & & + & & + & & + & & \leq & 5 \\
4x_1 & + & 1x_2 & + & & + & & + & & \leq & 5^2 \\
8x_1 & + & 4x_2 & + & 1x_3 & + & & + & & \leq & 5^3 \\
\vdots & + & & + & & + & & + & & \leq & \vdots \\
2^n x_1 & + & 2^{n-1}x_2 & + & \dots & + & 4x_{n-1} & + & 1x_n & \leq & 5^n
\end{array} \\
x_i \geq 0, i = 1, \dots, n
\end{array}
$$

The problem has

- n variables
- n constraints
- 2^n extreme points
- Simplex, starting at $x = (0, \dots, 0)$, visits all extreme points
- optimal solution $(0, 0, \dots, 0, 5^n)$

Complexity of Simplex

For $n = 3$ simplex visits $2^3 = 8$ extreme points
Assume (s_1, s_2, s_3) slack variables:

	nonbasis			
basis	x_1	x_2	x_3	RHS
s_1	1^*			5
s_2	4	1		25
s_3	8	4	1	125
$-z$	4	2	1	0

	nonbasis			
basis	s_1	x_2	x_3	RHS
x_1	1			5
s_2	-4	1^*		5
s_3	-8	4	1	85
$-z$	-4	2	1	-20

	nonbasis			
basis	s_1	s_2	x_3	RHS
x_1	1^*			5
x_2	-4	1		5
s_3	8	-4	1	65
$-z$	4	-2	1	-30

	nonbasis			
basis	x_1	s_2	x_3	RHS
s_1	1			5
x_2	4	1		25
s_3	-8	-4	1^*	25
$-z$	-4	-2	1	-50

	nonbasis			
basis	x_1	s_2	s_3	RHS
s_1	1^*			5
x_2	4	1		25
x_3	-8	-4	1	25
$-z$	4	2	-1	-75

	nonbasis			
basis	s_1	s_2	s_3	RHS
x_1	1			5
x_2	-4	1^*		5
x_3	8	-4	1	65
$-z$	-4	2	-1	-95

	nonbasis			
basis	s_1	x_2	s_3	RHS
x_1	1^*			5
s_2	-4	1		5
x_3	-8	4	1	85
$-z$	4	-2	-1	-105

	nonbasis			
basis	x_1	x_2	s_3	RHS
s_1	1^*			5
s_2	4	1		25
x_3	8	4	1	125
$-z$	-4	-2	-1	-125

Complexity of Simplex

Worst-case complexity is exponential

Average number of iterations required by ”largest-coefficient rule”:

$m \backslash n$	10	20	30	40	50
10	9.4	14.2	17.4	19.4	20.2
20		25.2	30.7	38.0	41.5
30			44.4	52.7	62.9
40				67.6	78.7
50					95.2

Source: Avis and Chvatal (1978).

Interior-point methods

Karmarkar (1984), many later improvements

- Does not examin basis solutions
- Polynomial running times can be proved
- Taha section 7.7

Solving IP models

Some IP naturally lead to integer solutions

- Totally unimodular (TU) matrices
- Several transportations problems and network problems are totally unimodular.

Preprocessing and reformulation

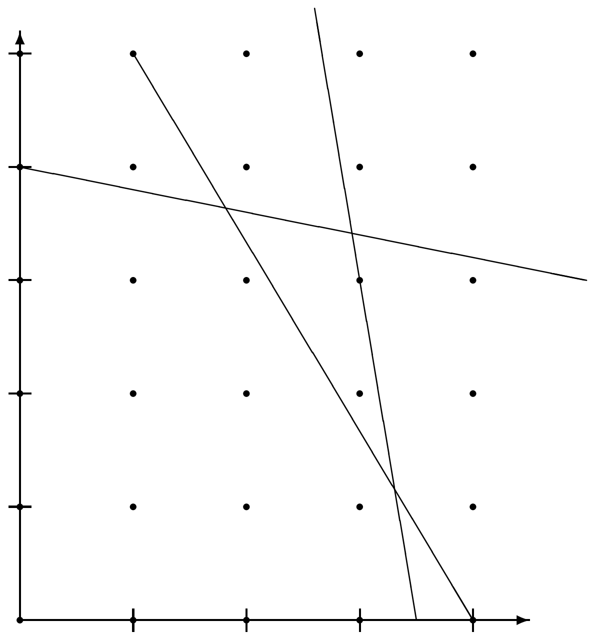
- Reformulation of constraints to TU
- Tightening M, m
- Fixation of variables
- Tightening of single constraints

Branch-and-bound methods

- Branching strategy
- Dual simplex

Convex hull

Smallest convex polyhedral which contains all integer points



feasible solutions	$\{x \in \mathbb{N}^n : Ax \leq b\}$
linear relaxation	$\{x \in \mathbb{R}^n : Ax \leq b\}$
convex hull	$\text{conv}\{x \in \mathbb{R}^n : Ax \leq b\}$

If constraints of an IP-model define the convex hull, then we can solve the problem efficiently.

Totally Unimodularity

Definition 1 An $m \times n$ integral matrix A is called *totally unimodular* (TU) if the determinant of each square submatrix of A is equal to 0, 1 or -1.

Obviously a_{ij} must be 0, 1, -1

Recognising whether A is TU demands an exponential number of steps

Proposition 1 If A is TU then A_B is also TU

Proof: If A is TU then the determinant of each square submatrix of A is equal to 0, 1 or -1. This also holds when restricted to columns in A_B.

Proposition 2 If A is TU then A^{-1} is also TU

Proof: From Cramer's rule $A_{ij}^{-1} = C_{ji}/\det(A)$ where C_{ji} is the adjoint matrix

$$C_{ji} = (-1)^{i+j} \det(A_{\text{row } i, \text{ column } j \text{ removed}})$$

Proposition 3 If A is TU and b is integral then any basis solution x_B is integral

Proof:

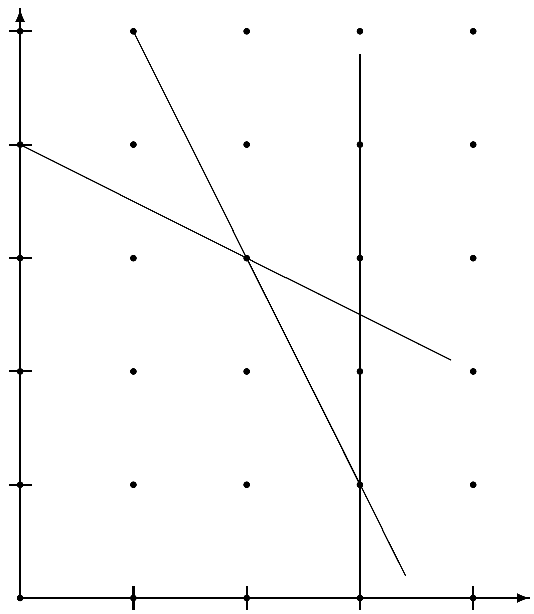
$$x_B = A_B^{-1} b$$

Totally Unimodularity

Proposition 4 If A is TU and b is an integral vector, then the polyhedral defined by

$$\{x \in \mathbb{R}^n : Ax \leq b\}$$

is integeral (i.e. all corner points are integral), or empty.



Proof: The corner points are the basis solutions

Totally Unimodularity

Proposition 5 ("*Property P*") Let A be a $(0, 1, -1)$ matrix with no more than two nonzero elements in each column. Then A is TU if and only if the rows of A can be divided in two subsets P_1 and P_2 such that if a column contains two nonzero elements, the following statements are true:

1 If both nonzero elements have the same sign, then one is in a row contained in P_1 and the other is in a row contained in P_2.

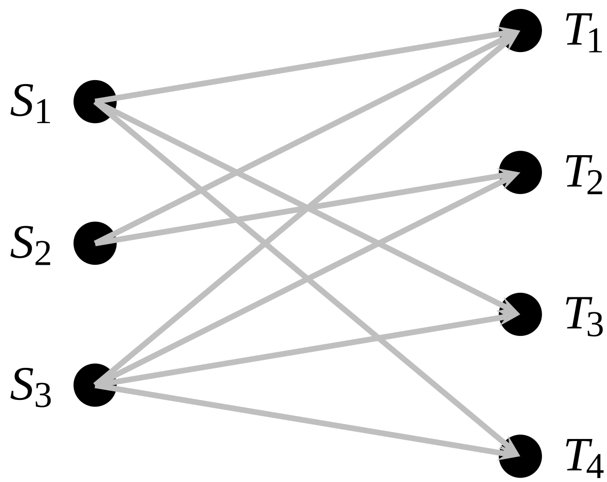
2 If the two nonzero elements have opposite sign, then both are in rows contained in the same subset.

Example

$$\begin{pmatrix} & & -1 & & & & & \\ & 1 & & 1 & & & 1 & \\ -1 & & & -1 & 1 & & & \\ & & & & & 1 & & \\ & 1 & -1 & & & & & -1 \\ & & & & & -1 & & \end{pmatrix}$$

Example of TU

Three suppliers (S_1, S_2, S_3) should provide four customers (T_1, T_2, T_3, T_4) with a particular commodity.



Commodities cannot be split, hence IP problem.

Supplier	S_1	S_2	S_3
Capacity	135	56	93

Customer	T_1	T_2	T_3	T_4
Requirements	62	83	39	91

	Customer			
Supplier	T_1	T_2	T_3	T_4
S_1	132	—	97	103
S_2	85	91	—	—
S_3	106	89	100	98

Example of TU

Variables: x_{ij} is flow from supplier i to customer j

IP-model

minimize

$$132x_{11}+Mx_{12}+97x_{13}+103x_{14}+85x_{21}+91x_{22}+Mx_{23}+Mx_{24}+106x_{31}+89x_{32}+100x_{33}+98x_{34}$$

subject to

$$\begin{array}{lllllllllllll}
x_{11} & +x_{12} & +x_{13} & +x_{14} & & & & & & & & & \leq 135 \\
& & & & x_{21} & +x_{22} & +x_{23} & +x_{24} & & & & & \leq 56 \\
& & & & & & & & x_{31} & +x_{32} & +x_{33} & +x_{34} & \leq 93 \\
x_{11} & & & & +x_{21} & & & & +x_{31} & & & & \geq 62 \\
& x_{12} & & & & +x_{22} & & & & +x_{32} & & & \geq 83 \\
& & x_{13} & & & & +x_{23} & & & & +x_{33} & & \geq 39 \\
& & & x_{14} & & & & +x_{24} & & & & +x_{34} & \geq 91
\end{array}$$

$x_{ij} \in \mathbb{N}$

The model is TU, hence LP returns IP solutions

Notice

1 Problem can be formulated by a TU matrix

2 Problem can be solved efficiently

$1 \Rightarrow 2$ is obvious, but frequently also $2 \Rightarrow 1$.
(e.g. flow conservation)

Good and bad formulations (Williams)

i) The straightforward formulation results in an IP model where the feasible region is already the convex hull of integer points.

ii) The problem can fairly easily be reformulated to give a feasible region corresponding to the convex hull of integer points.

iii) By reformulation it is possible to reduce the feasible region of the LP problem to nearer that of the convex hull of integer points.

Good and bad formulations

i) The straightforward formulation defines the convex hull

- LP-solver will automatically return integer solution
- Important to know if a problem is NP-hard
- If we can prove that constraint matrix is TU then polynomially solvable

Good and bad formulations

ii) Reformulate to convex hull

$$(\delta_1 = 1 \lor \delta_2 = 1 \lor \ \ldots \ \lor \delta_n = 1) \ \Rightarrow \ \delta = 1$$

Can be written

$$\delta_1 + \delta_2 + \ldots + \delta_n \geq 1 \ \Rightarrow \ \delta = 1$$

LP-model

$$(\delta_1 + \delta_2 + \ldots + \delta_n) - n\delta \leq 0$$

Better formulation

$$\begin{aligned} &\delta_1 = 1 \ \Rightarrow \ \delta = 1 \\ &\delta_2 = 1 \ \Rightarrow \ \delta = 1 \\ &\vdots \\ &\delta_n = 1 \ \Rightarrow \ \delta = 1 \end{aligned}$$

LP-model

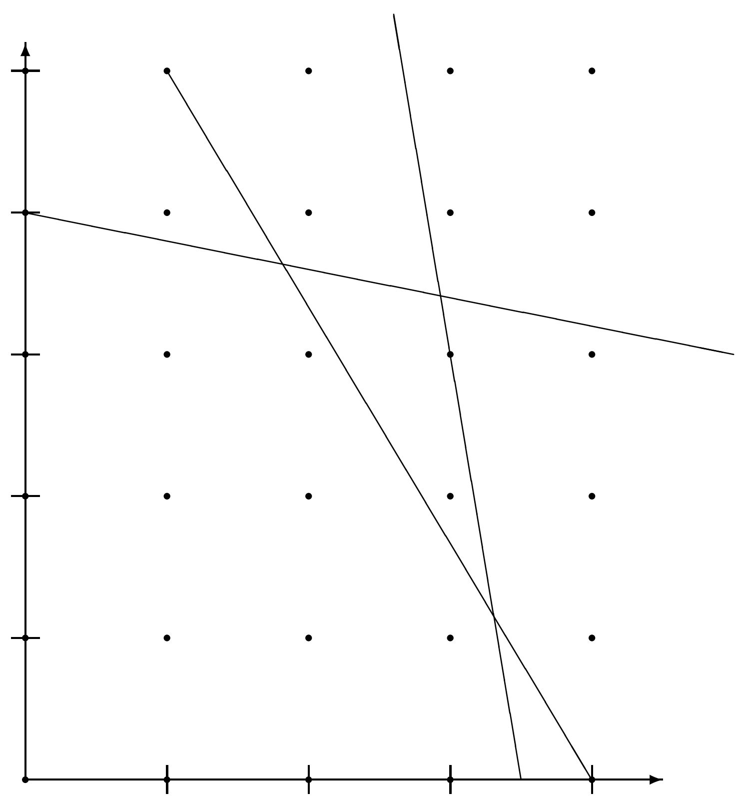
$$\begin{aligned} &\delta_1 - \delta \leq 0 \\ &\delta_2 - \delta \leq 0 \\ &\vdots \\ &\delta_n - \delta \leq 0 \end{aligned}$$

Has “property P”

Good and bad formulations

iii) Reformulate so closer to convex hull

- LP-solution closer to IP-solution
- Better upper bounds



Choose M and m as tight as possible

Good and bad formulations

To model $\boxed{x > 0 \;\Rightarrow\; \delta = 1}$ we use $\boxed{x - M\delta \leq 0}$

$$S_1 = \left\{ (x, \delta) \mid x - M_1\delta \leq 0, \delta \in \{0, 1\} \right\}$$
$$S_2 = \left\{ (x, \delta) \mid x - M_2\delta \leq 0, \delta \in \{0, 1\} \right\}$$

where $M_1 < M_2$. Consider *LP-relaxation* of S_1 and S_2

Will show: $S_1 \subset S_2$

- Solutions in S_1 are also solutions in S_2
  Consider $(x, \delta) \in S_1$

$$x \leq M_1\delta \leq M_2\delta \quad \Rightarrow \quad x - M_2\delta \leq 0 \qquad (x, \delta) \in S_2$$

- Solutions in S_2 exists which are not solutions in S_1
  Consider (x, δ) where $\delta = \frac{x}{M_2}$ and $x > 0$

$$x - M_2\delta \leq 0 \qquad (x, \delta) \in S_2$$
$$x - M_1\delta = x - M_1\frac{x}{M_2} = x\left(1 - \frac{M_1}{M_2}\right) > 0 \quad (x, \delta) \notin S_1$$

Simplifying an IP model

$$\begin{array}{lrcrcrcrcrcrl}
\min & 5\delta_1 & + & 7\delta_2 & + & 10\delta_3 & + & 3\delta_4 & + & 1\delta_5 & & & \\
\text{s.t.} & \delta_1 & - & 3\delta_2 & + & 5\delta_3 & + & \delta_4 & - & \delta_5 & \geq & 2 & (1) \\
& -2\delta_1 & + & 6\delta_2 & - & 3\delta_3 & - & 2\delta_4 & + & 2\delta_5 & \geq & 0 & (2) \\
& & - & \delta_2 & + & 2\delta_3 & - & 2\delta_4 & - & \delta_5 & \geq & 1 & (3) \\
& \multicolumn{12}{l}{\delta_1, \delta_2, \delta_3, \delta_4, \delta_5 \in \{0, 1\}}
\end{array}$$

- Using (3) we have

  $$2\delta_3 \geq 1 + \delta_2 + 2\delta_4 + \delta_5 \geq 1$$

  hence $\delta_3 \geq \frac{1}{2}$.
- Using (2) we have

  $$6\delta_2 \geq 3 + 2\delta_1 + 2\delta_4 - 2\delta_5 \geq 1$$

  hence $\delta_2 \geq \frac{1}{6}$.
- Using (3) we have

  $$2\delta_4 \leq -\delta_5 \leq 0$$

  hence $\delta_4 = 0$
- Using (3) $\delta_5 \leq 0$.
- By inspection $\delta_1 = 0$.

Three-dimensional noughts and crosses (Williams)

27 cells are arranged in a $(3 \times 3 \times 3)$-dimensional array.

Three cells are regarded as laying in the same line if they are on the same horizontal or vertical line or on the same diagonal. There are 49 lines altogether

×	×	×
o	o	×
×	o	o

×	o	×
o	×	×
×	×	o

o	×	o
o	o	×
×	×	o

Three-dimensional noughts and crosses

- the player getting three balls on one line, wins
- is it possible to play “remis”?
- i.e. what is the minimum number of covered lines during a game

Thus: given 13 white balls (noughts) and 14 black balls (crosses), arrange them one to a cell, so as to minimize the number of lines with balls all of one colour.

Three-dimensional noughts and crosses

Each cell gets a number

$$1, 2, 3, \ldots, 27$$

Notice that all the 27 balls are arranged. Boolean variable

$$\delta_j = \begin{cases} 1 & \text{if cell } j \text{ contains a black ball} \\ 0 & \text{if cell } j \text{ contains a white ball} \end{cases}$$

There are 49 lines, e.g.

$$\begin{array}{ll} 1, 2, 3 & 1, 4, 9 \\ 3, 14, 25 & 9, 18, 27 \end{array}$$

We introduce an indicator variable γ_i for each line i saying

$$\gamma_i = \begin{cases} 1 & \text{if all balls in line } i \text{ have the same colour} \\ 0 & \text{otherwise} \end{cases}$$

Thus

$$\gamma_i = 0 \;\Rightarrow\; \begin{cases} \delta_{i1} + \delta_{i2} + \delta_{i3} \geq 1 \\ \delta_{i1} + \delta_{i2} + \delta_{i3} \leq 2 \end{cases}$$

Can be modeled as

$$\begin{array}{l} \delta_{i1} + \delta_{i2} + \delta_{i3} + \gamma_i \geq 1 \\ \delta_{i1} + \delta_{i2} + \delta_{i3} - \gamma_i \leq 2 \end{array}$$

Objective function

$$\text{minimize } z = \sum_{i=1}^{49} \gamma_i$$

Model has 99 constraints, 76 boolean variables

Three-dimensional noughts and crosses

Solved by CPLEX, mixed-integer programming (built-in branch-and-bound code).

Solution

$$z = \sum_{i=1}^{49} \gamma_i = 4$$

395 branching nodes.

The optimal solution

×	×	o
o	o	×
×	o	×

×	o	×
o	o	×
×	×	o

o	×	o
×	×	o
o	o	×

The four lines are

		1,2
3		
4		

	1	
	2,3	
	4	

1		
		3
2		4