MATH 380

Hemanshu Kaul

kaul@iit.edu

Useful Ideas for Linear Programs

① $\max f(\vec{x})$ is equivalent to $\min -f(\vec{x})$

② $a_1x_1+a_2x_2+\cdots+a_nx_n=b$ is equivalent to $\begin{array}{l} a_1x_1+a_2x_2+\cdots+a_nx_n\leq b \\ a_1x_2+a_2x_2+\cdots+a_nx_n\geq b \end{array}$

③ $a_1x_1+a_2x_2+\cdots+a_nx_n\geq b$ is equivalent to $(-a_1)x_1+(-a_2)x_2+\cdots+(-a_n)x_n\leq -b$

④ $\min \left(\max\limits_{i=1,\dots,m} (c_i^T\vec{x}+d_i)\right)$
s.t. $A\vec{x}\geq\vec{b}$

is equivalent to

$\min z$
s.t.
$z\geq c_i^T\vec{x}+d_i \ \forall i$
$A\vec{x}\geq\vec{b}$

Recall CAC

⑤ $\min \left(\sum\limits_{i=1}^{n}|x_i|\right)$
s.t. $A\vec{x}\geq\vec{b}$

is equivalent to

$\min \sum\limits_{i=1}^{n} z_i$
s.t. $A\vec{x}\geq\vec{b}$
$z_i\geq x_i$, $i=1,\dots,n$
$z_i\geq -x_i$, $i=1,\dots,n$

Recall minimize avg. deviation criterion

non-linear programs as linear programs

Geometry and Algorithms for Linear Optimization

Any linear optimization problem can be written in the form

$$\boxed{\begin{aligned} \min\ & c_1x_1 + c_2x_2 + \dots + c_nx_n \\ \text{s.t. } & a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \\ & a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2 \\ & \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m \\ & x_1, x_2, \dots, x_n \geqslant 0 \end{aligned}}$$

→

$$\min\ \vec{c}^T\vec{x}$$

$$\text{s.t. } A\vec{x} \leq \vec{b}$$

$$\vec{x} \geqslant 0$$

where $\vec{x} \in \mathbb{R}^n$

A is $m \times n$

$\vec{b} \in \mathbb{R}^m$

n = # variables

m = # constraints

$A, \vec{b}$ are given data and $\vec{x}$ is the unknown.

Any $\vec{x} \in \mathbb{R}^n$ that satisfies all the constraints is called a feasible solution.

Feasible set is the collection of all feasible solutions.

Geometrically it looks like a polyhedron.

e.g. $\min \; -x_1 - x_2$

s.t. $x_1 + 2x_2 \leqslant 3$

$2x_1 + x_2 \leqslant 3$

$x_1, x_2 \geqslant 0$

e.g. $\min \; -x_1 - x_2$ ← Objective function defines a family of lines $-x_1 - x_2 = k$

s.t. $x_1 + 2x_2 \leq 3$ ← Each constraint defines a half-plane (region with boundary given by the straight line $x_1 + 2x_2 = 3$)

$2x_1 + x_2 \leq 3$

$x_1, x_2 \geqslant 0$ ← restricts the feasible set to the non-negative quadrant. Boundary given by $x_1 = 0$ & $x_2 = 0$.

e.g. $\min \; -x_1 - x_2$

s.t. $x_1 + 2x_2 \leq 3$

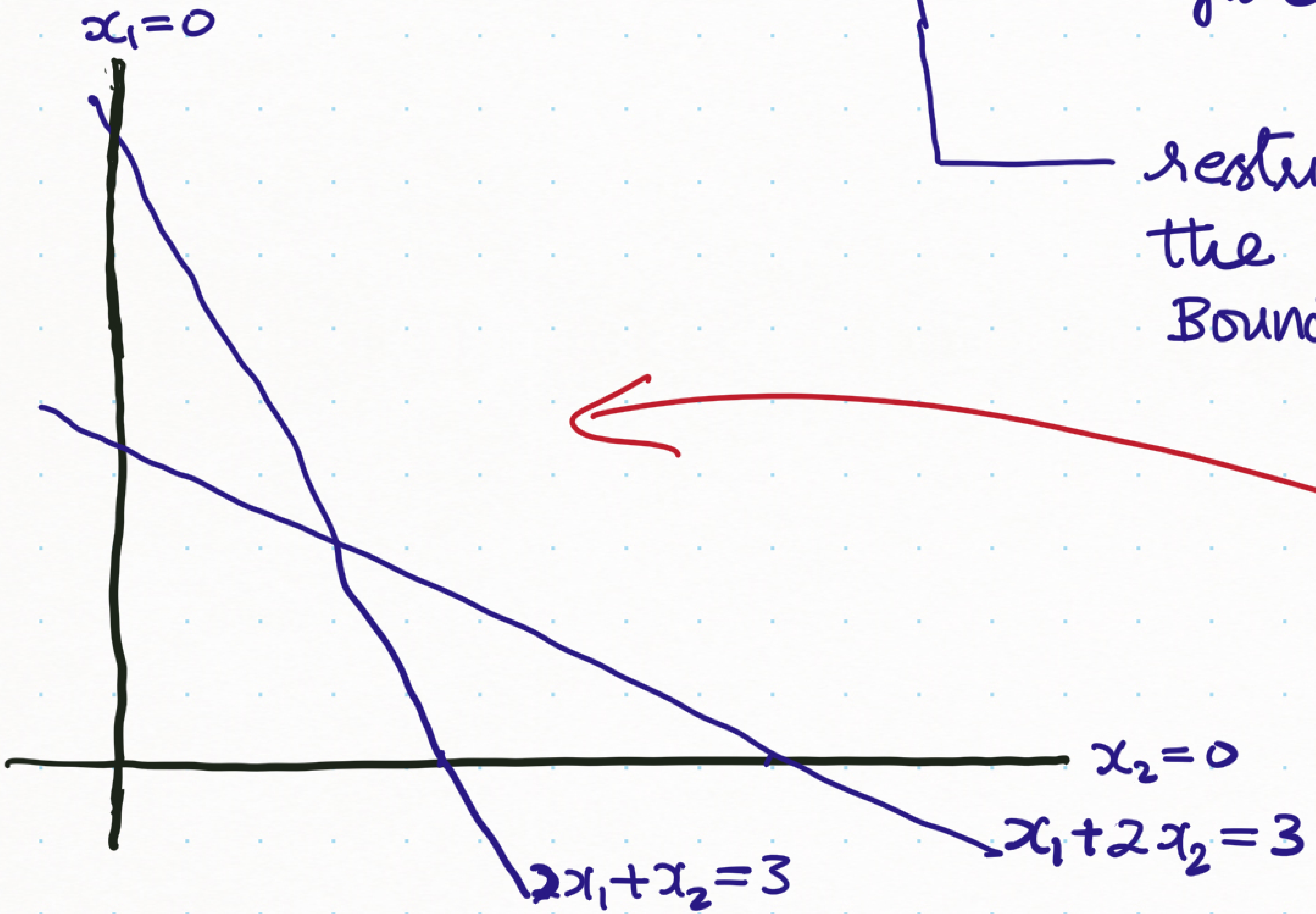
$2x_1 + x_2 \leq 3$

$x_1, x_2 \geqslant 0$

Objective function defines a family of lines $-x_1 - x_2 = k$

Each constraint defines a half-plane (region with boundary given by the straight line $x_1 + 2x_2 = 3$)

restricts the feasible set to the non-negative quadrant Boundary given by $x_1 = 0$ & $x_2 = 0$.



Where is the feasible region?

e.g. $\min \; -x_1 - x_2$

s.t. $x_1 + 2x_2 \leqslant 3$

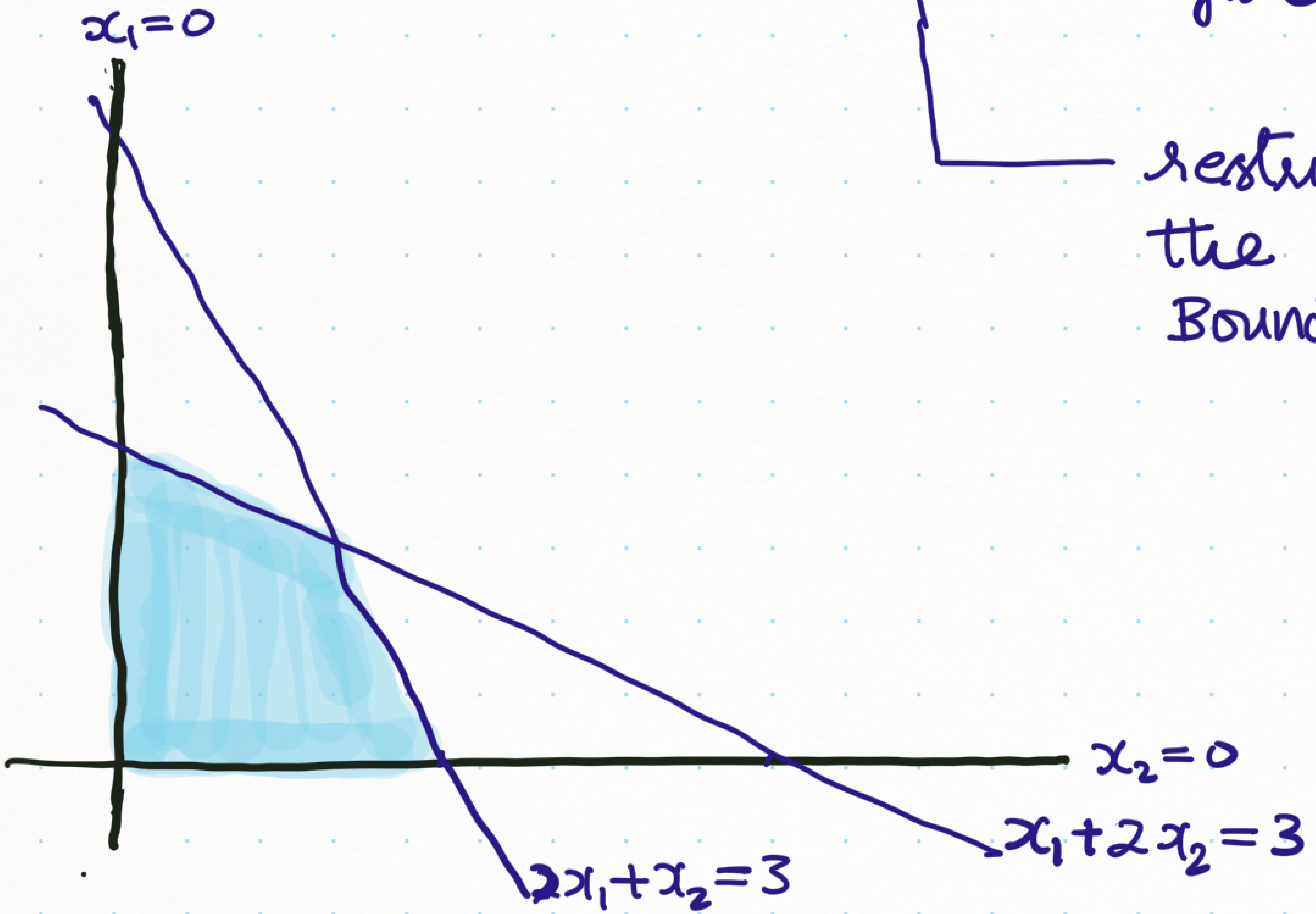
$2x_1 + x_2 \leqslant 3$

$x_1, x_2 \geqslant 0$

← Objective function defines a family of lines $-x_1 - x_2 = k$

← Each constraint defines a half-plane (region with boundary given by the straight line $x_1 + 2x_2 = 3$)

← restricts the feasible set to the non-negative quadrant Boundary given by $x_1 = 0$ & $x_2 = 0$.



Where is an optimal solution?

e.g. $\min \; -x_1 - x_2$

s.t. $x_1 + 2x_2 \leq 3$

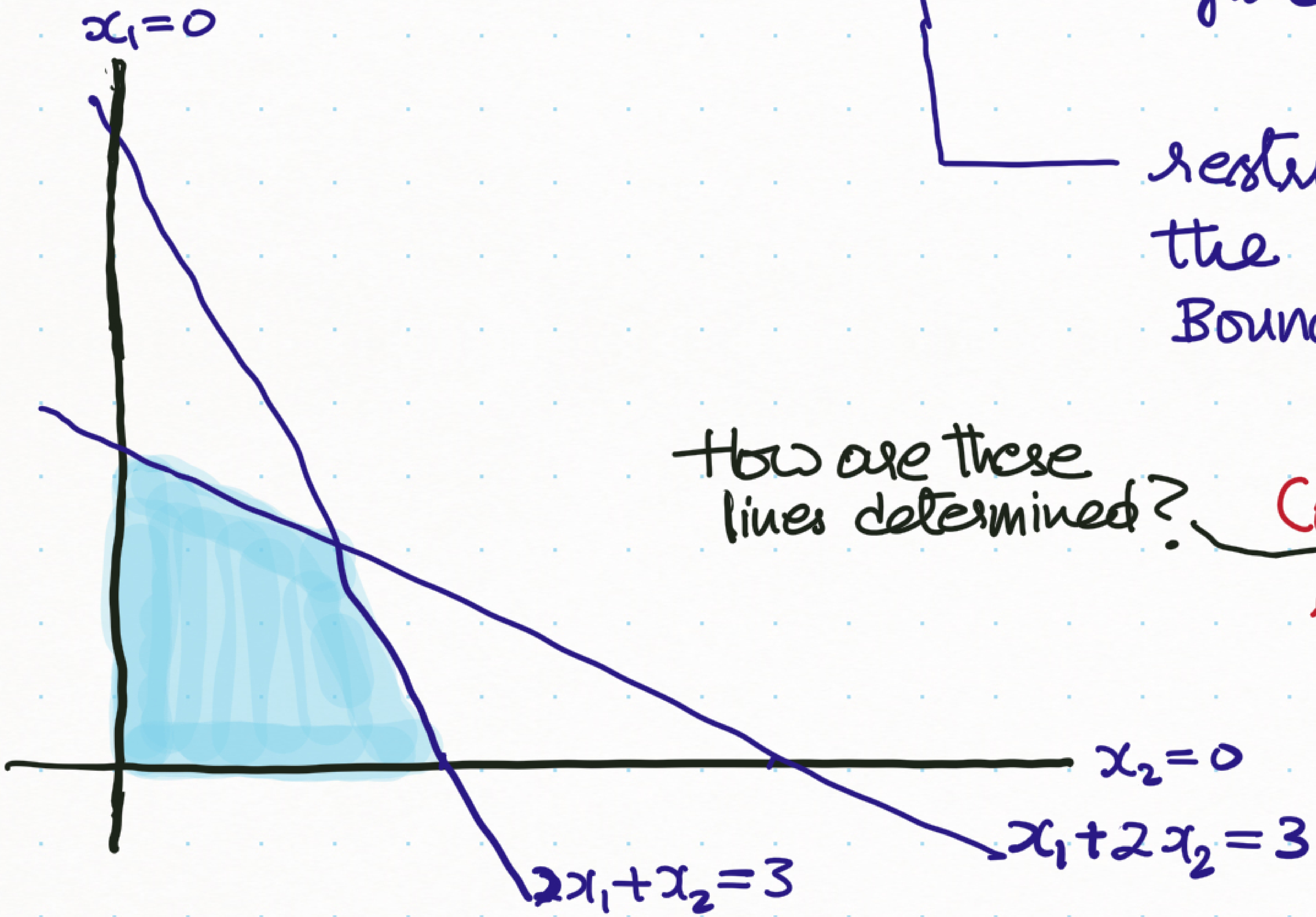
$2x_1 + x_2 \leq 3$

$x_1, x_2 \geqslant 0$

← Objective function defines a family of lines $-x_1 - x_2 = k$

← Each constraint defines a half-plane (region with boundary given by the straight line $x_1 + 2x_2 = 3$)

← restricts the feasible set to the non-negative quadrant Boundary given by $x_1 = 0$ & $x_2 = 0$.



How are these lines determined?

Consider the family of lines $\boxed{-x_1 - x_2 = k}$ as they pass through the feasible region.

e.g. $\min \; -x_1 - x_2$

s.t. $x_1 + 2x_2 \leq 3$

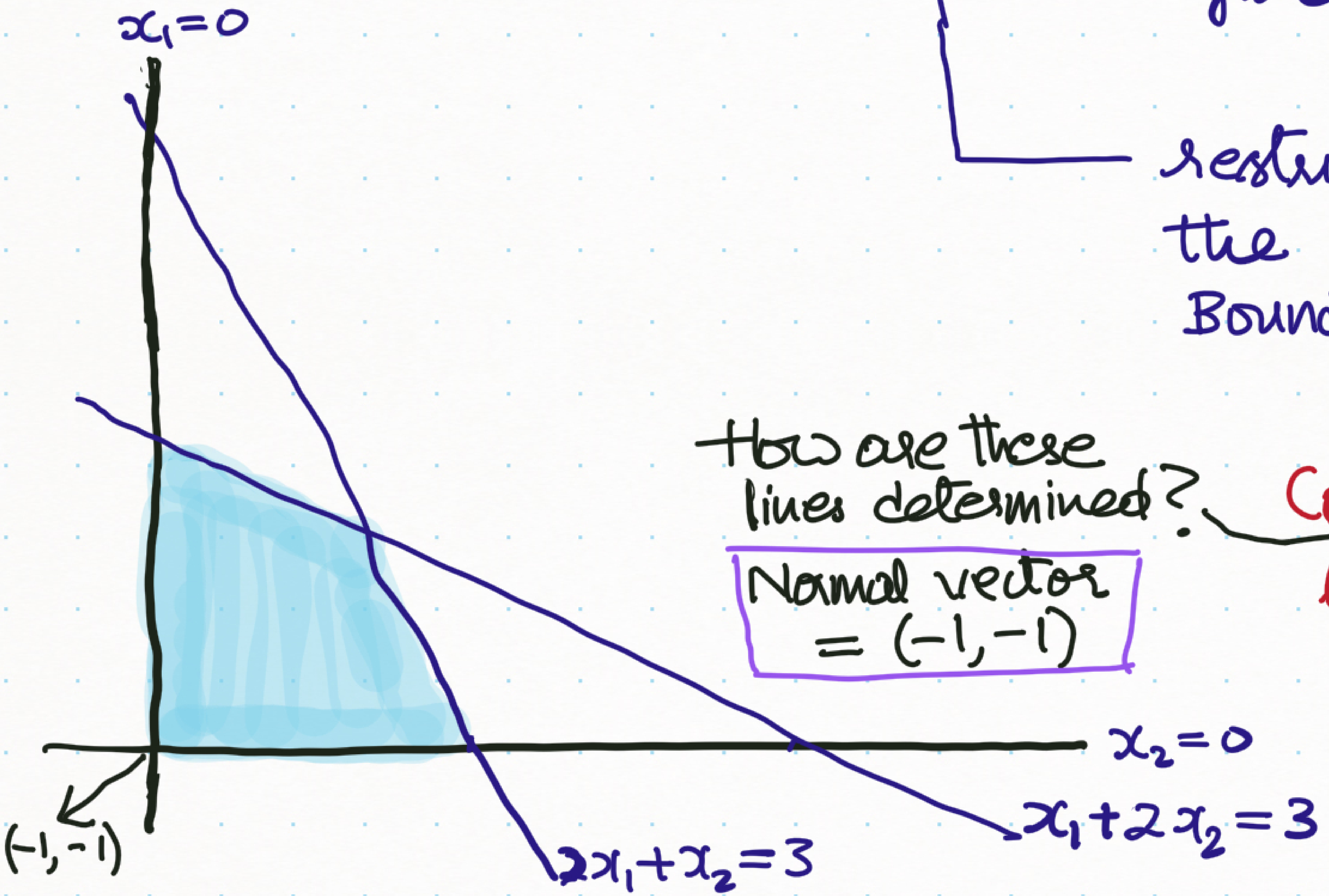
$2x_1 + x_2 \leq 3$

$x_1, x_2 \geqslant 0$

← Objective function defines a family of lines $-x_1 - x_2 = k$

← Each constraint defines a half-plane (region with boundary given by the straight line $x_1 + 2x_2 = 3$)

← restricts the feasible set to the non-negative quadrant. Boundary given by $x_1 = 0$ & $x_2 = 0$.



How are these lines determined?

$\boxed{\text{Normal vector} = (-1,-1)}$

Consider the family of lines $\boxed{-x_1 - x_2 = k}$ as they pass through the feasible region.

e.g. min $-x_1 - x_2$

s.t. $x_1 + 2x_2 \leq 3$

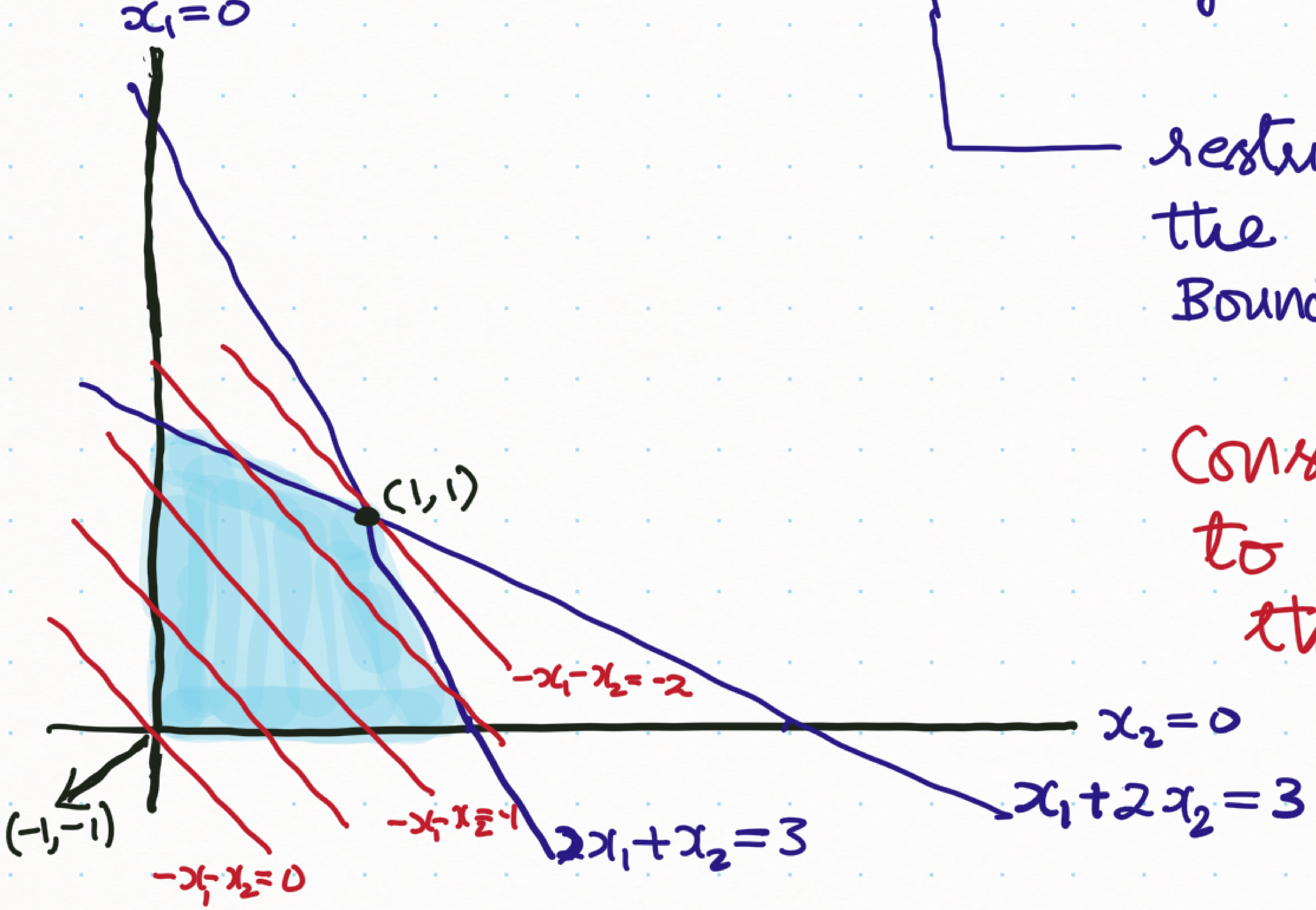
$2x_1 + x_2 \leq 3$

$x_1, x_2 \geqslant 0$

← Objective function defines a family of lines $-x_1 - x_2 = k$

← Each constraint defines a half-plane (region with boundary given by the straight line $x_1 + 2x_2 = 3$)

restricts the feasible set to the non-negative quadrant Boundary given by $x_1 = 0$ & $x_2 = 0$.



Consider lines perpendicular to (−1,−1) as they pass through the feasible region.

e.g. min $-x_1 - x_2$

s.t. $x_1 + 2x_2 \leq 3$

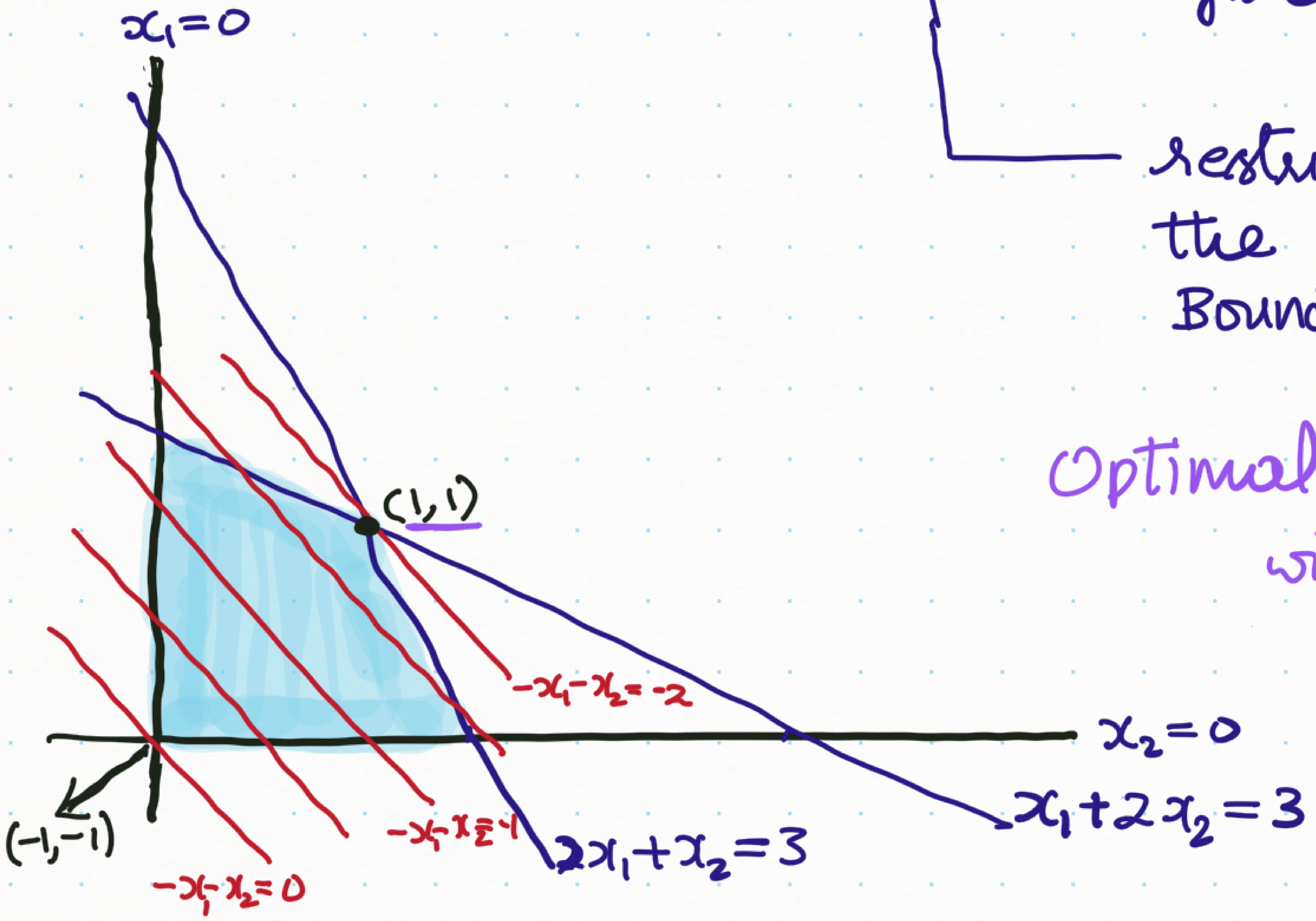
$2x_1 + x_2 \leq 3$

$x_1, x_2 \geqslant 0$

← Objective function defines a family of lines $-x_1 - x_2 = k$

← Each constraint defines a half-plane (region with boundary given by the straight line $x_1 + 2x_2 = 3$)

← restricts the feasible set to the non-negative quadrant Boundary given by $x_1 = 0$ & $x_2 = 0$.



Optimal solution $(1,1)$ with value $-(1)-(1) = -2$

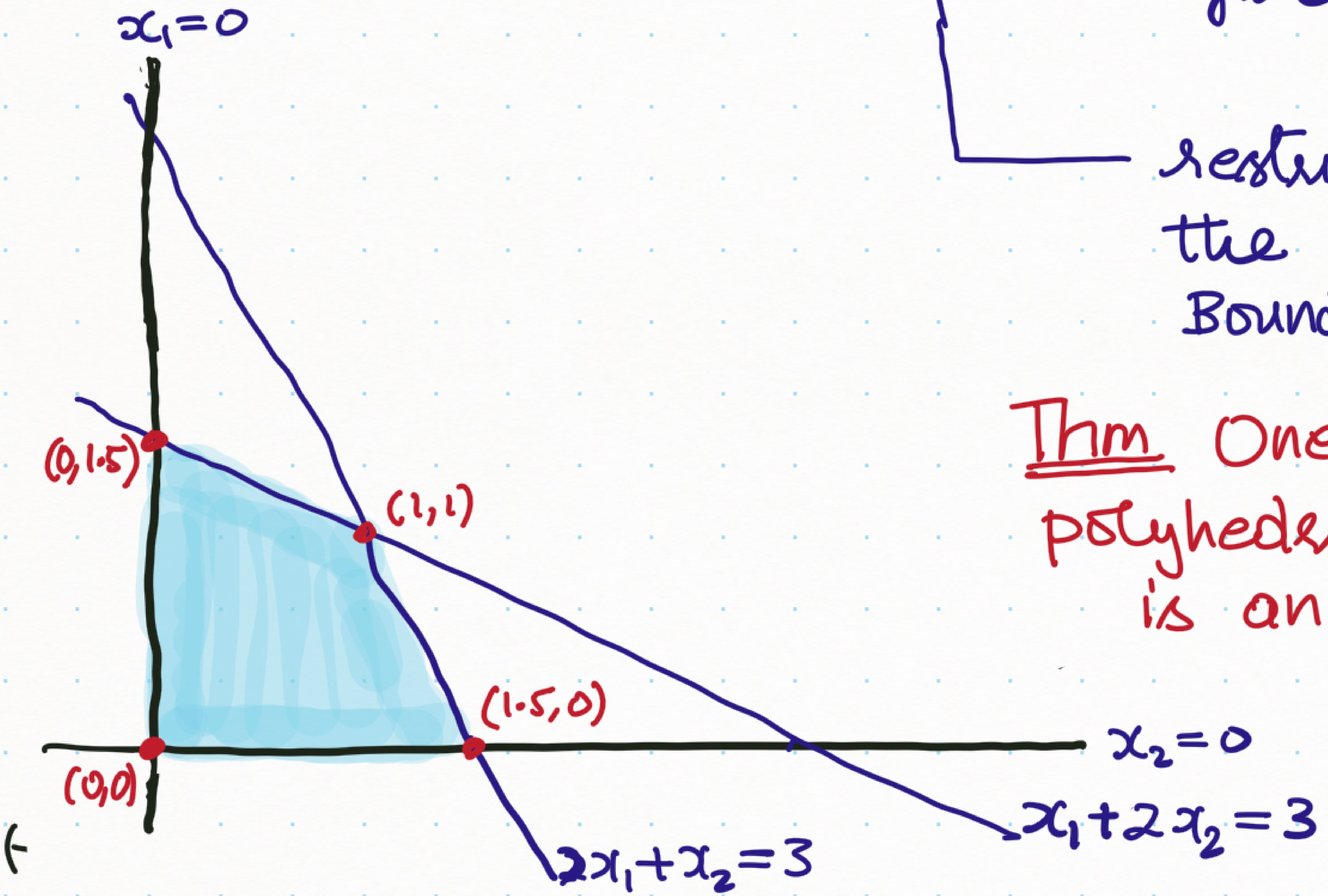
e.g.

$$\begin{aligned} \min\ & -x_1 - x_2 \\ \text{s.t.}\ & x_1 + 2x_2 \leq 3 \\ & 2x_1 + x_2 \leq 3 \\ & x_1, x_2 \geq 0 \end{aligned}$$

← Objective function defines a family of lines $-x_1 - x_2 = k$

← Each constraint defines a half-plane (region with boundary given by the straight line $x_1 + 2x_2 = 3$)

← restricts the feasible set to the non-negative quadrant Boundary given by $x_1 = 0$ & $x_2 = 0$.



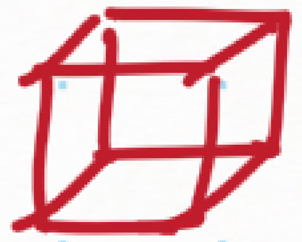
Thm One of the corners of the polyhedral feasible region is an optimal solution (if it exists).

"Infinite problem reduced to finite problem"

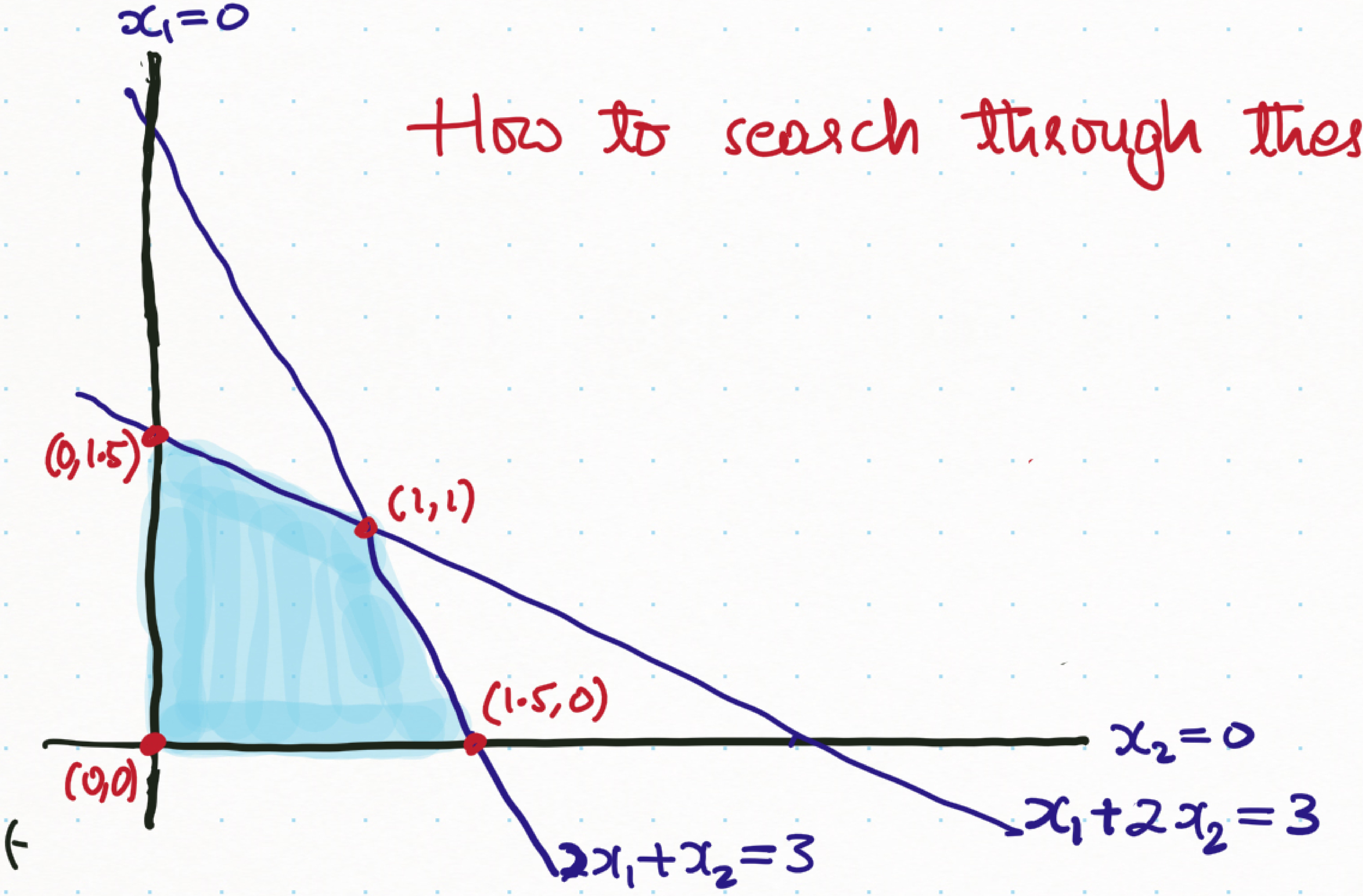
There are infinitely many feasible solutions.

There can be infinitely many optimal solutions.

But there will be only finitely many "corners" of the feasible region.

However, there can be exponentially many "corners" in terms of the size of the problem ($n = \#$ vars) e.g.  $0 \le x_i \le 1$, $i=1,\dots,n$ has 2^n corners

How to search through these corners systematically and efficiently?

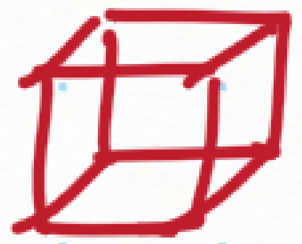


There are infinitely many feasible solutions.

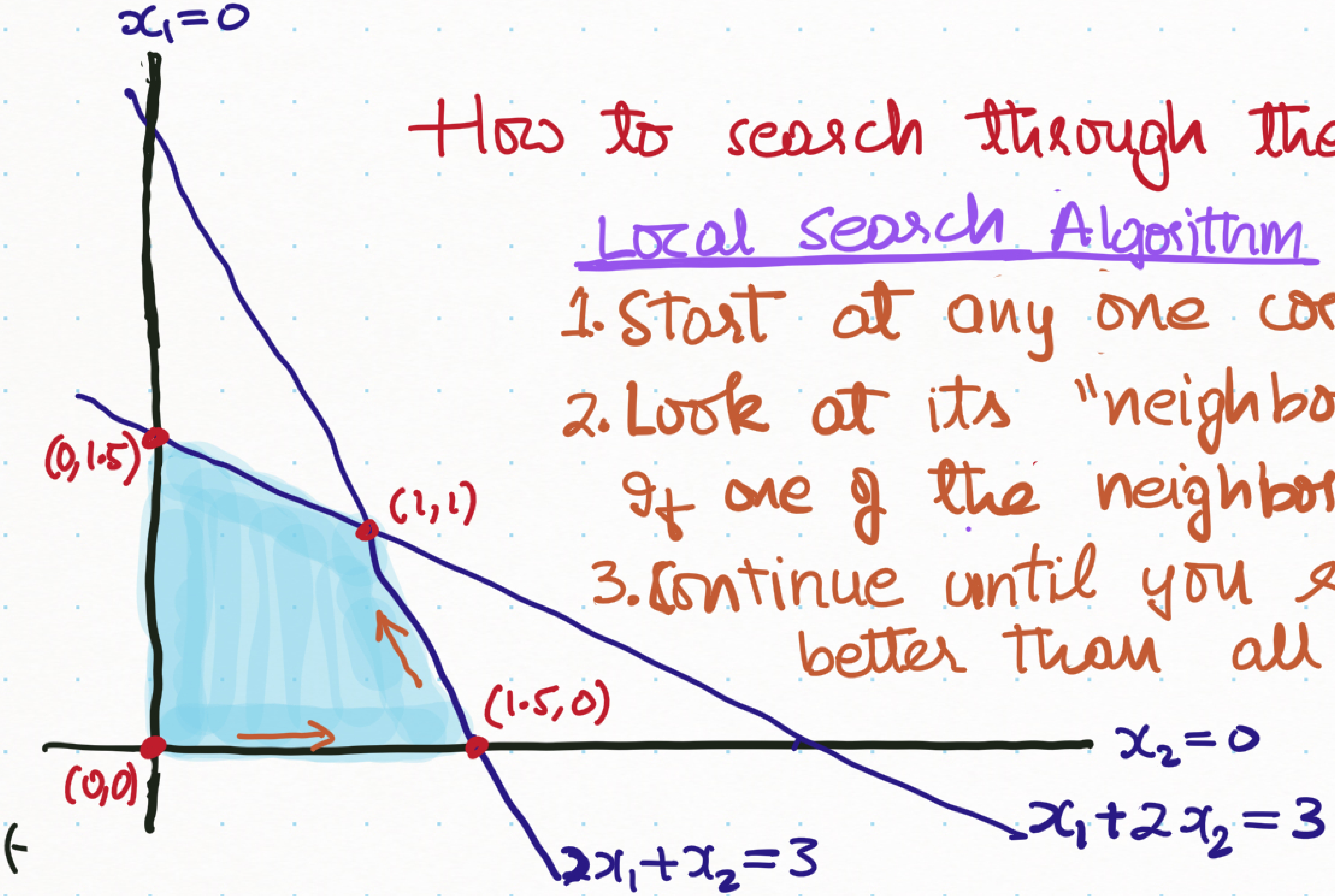
There can be infinitely many optimal solutions.

But there will be only finitely many "corners" of the feasible region.

However, there can be exponentially many "corners" in terms of the size of the problem ($n = \#$ vars) e.g.



$0 \le x_i \le 1$, $i = 1, \dots, n$ has 2^n corners



How to search through these corners systematically and efficiently?

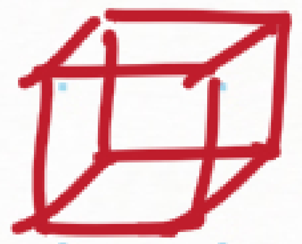
Local Search Algorithm

1. Start at any one corner solution, say (0,0)
2. Look at its "neighboring" corners
   If one of the neighbors is better move to that corner
3. Continue until you reach a corner that is better than all its neighboring corners.

Local Optimum

There are infinitely many feasible solutions.
There can be infinitely many optimal solutions.
But there will be only finitely many "corners" of the feasible region.

However, There can be exponentially many "corners" in terms of the size of the problem (n = # vars) e.g.



$0 \le x_i \le 1$, $i = 1, \dots, n$
has 2^n corners

How to search through these corners systematically and efficiently?

Local Search Algorithm

1. Start at any one corner solution
2. Look at its "neighboring" corners
   If one of the neighbors is better move to that corner
3. Continue until you reach a corner that is better than all its neighboring corners.

→ Translating this into linear Algebra gives us Simplex Algorithm for solving Linear programs to Global (!!) optimality.

Modeling using (Combinatorial) Graphs & Networks

Graphs are the mathematical structures underlying networks, they model relationships between entities.

This flexibility makes them ubiquitous in mathematical models arising in Computer Science (algorithms, data science, ML), Physics (Statistical, Quantum, ..), Biology (Systems, Genetics, ..), Chemistry, Sociology (Social Networks), Anthropology (Kinship networks), Epidemiology, Linguistics, Statistics (Network models and data), and as a tool in other parts of mathematics.

Modeling using (Combinatorial) Graphs & Networks

Graphs are the mathematical structures underlying networks, they model relationships between entities.

This flexibility makes them ubiquitous in mathematical models arising in Computer Science (algorithms, data science, ML), Physics (Statistical, Quantum,..), Biology (Systems, Genetics,..), Chemistry, Sociology (Social Networks), Anthropology (Kinship networks), Epidemiology, Linguistics, Statistics (Network models and data), and as a tool in other parts of mathematics.

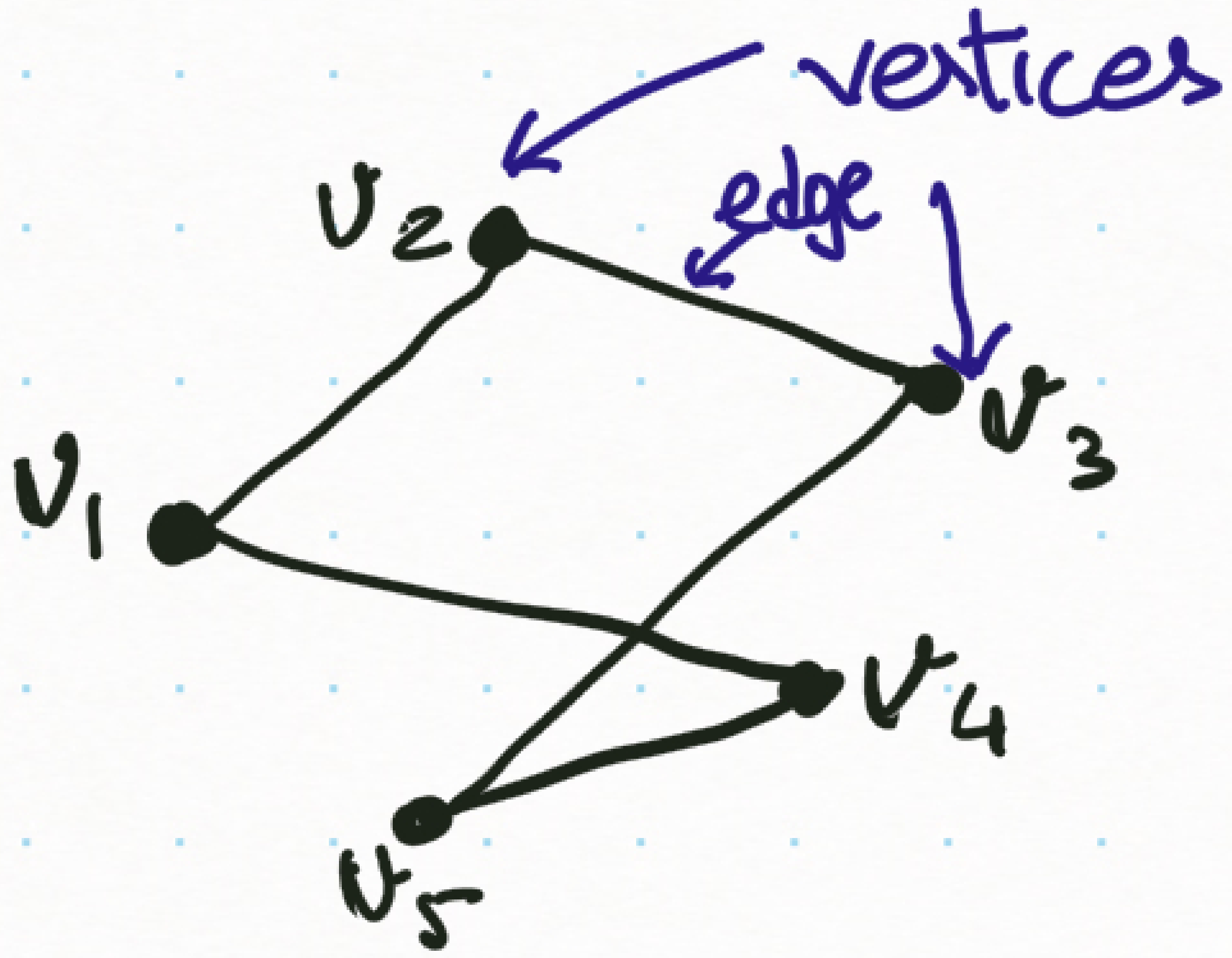
entities being studied

A graph / network consists of vertices / nodes

edges / arcs

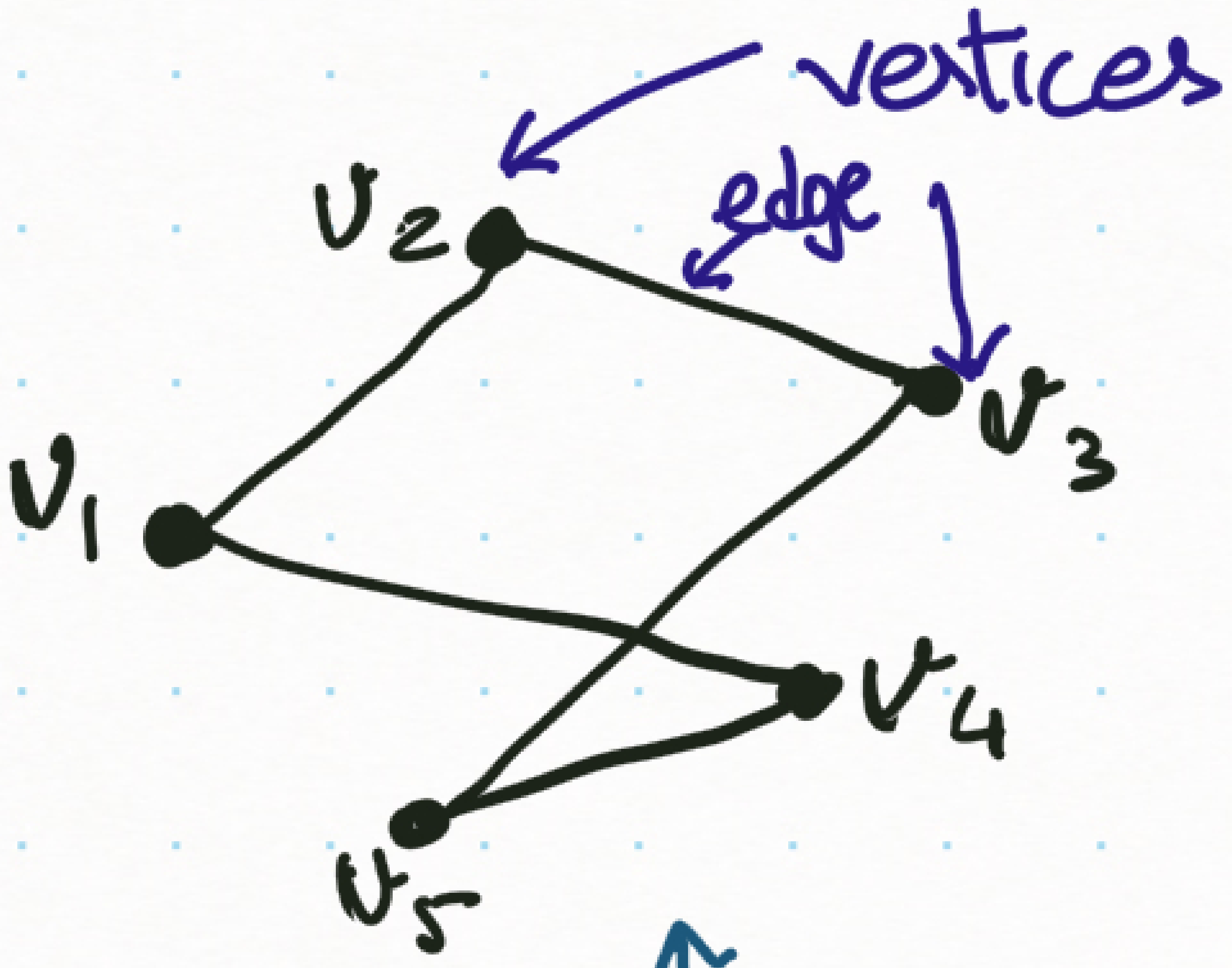
pairwise relations between the entities.

Visually



Algebsaicly

Visually



↕

Algebraicly

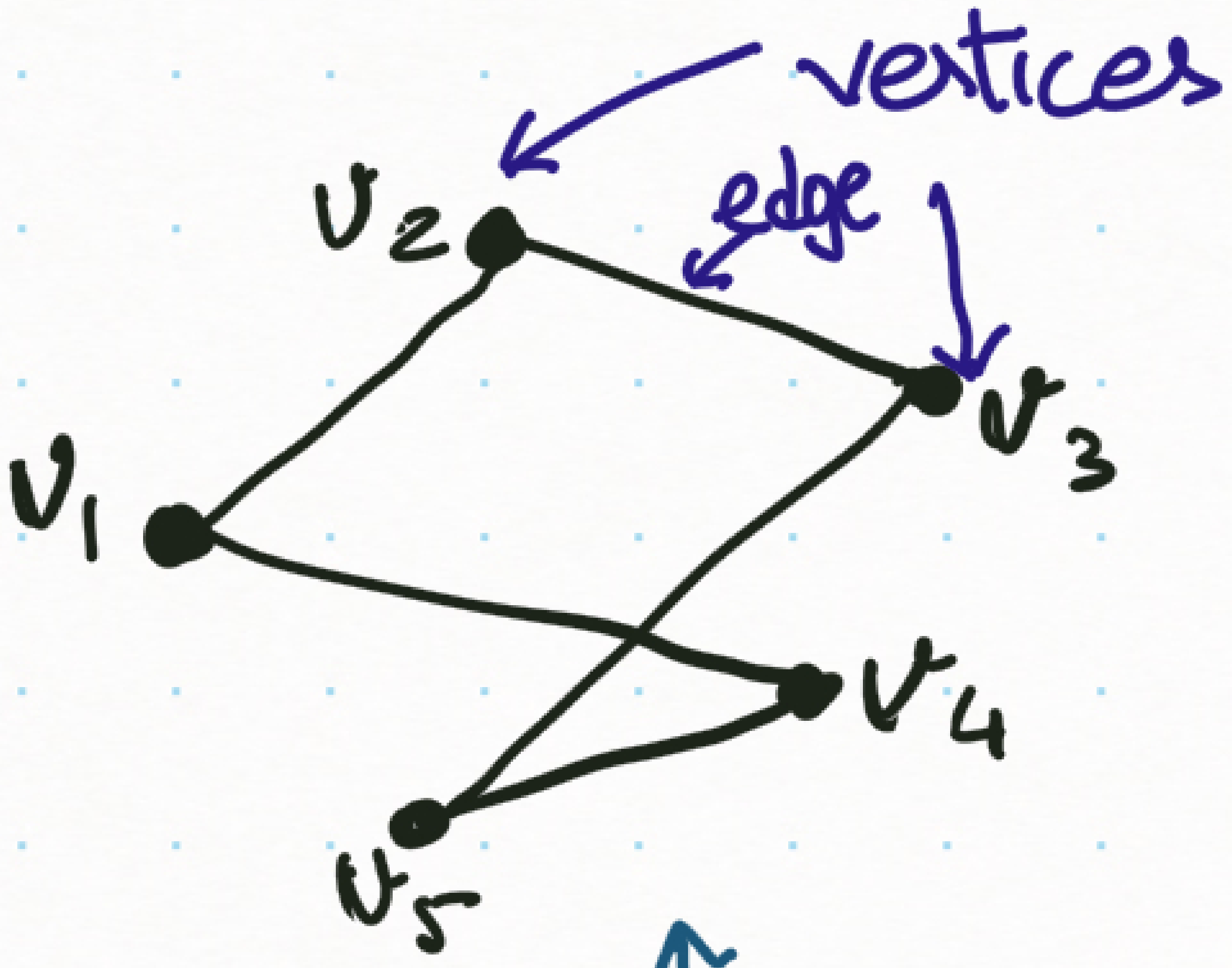
Adjacency Matrix

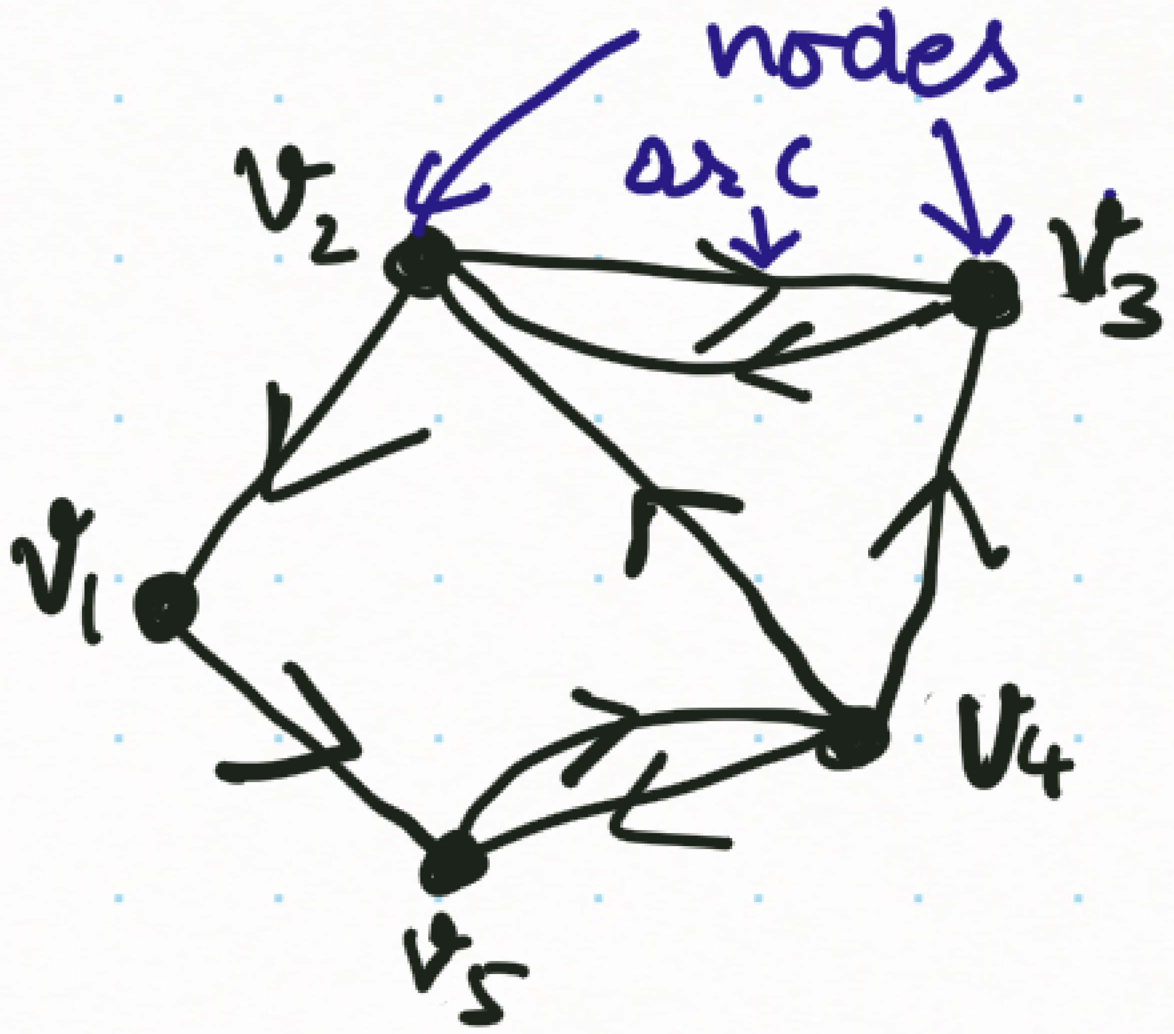
$$\begin{array}{c|ccccc} & v_1 & v_2 & v_3 & v_4 & v_5 \\ \hline v_1 & 0 & 1 & 0 & 1 & 0 \\ v_2 & 1 & 0 & 1 & 0 & 0 \\ v_3 & 0 & 1 & 0 & 0 & 1 \\ v_4 & 1 & 0 & 0 & 0 & 1 \\ v_5 & 0 & 0 & 1 & 1 & 0 \end{array}$$

$$(i,j) = \begin{cases} 1 & \text{if } v_i \Leftrightarrow v_j \\ 0 & \text{if } v_i \not\Leftrightarrow v_j \end{cases}$$

Adjacency lists ....

Visually





$\updownarrow$

Algebraicly

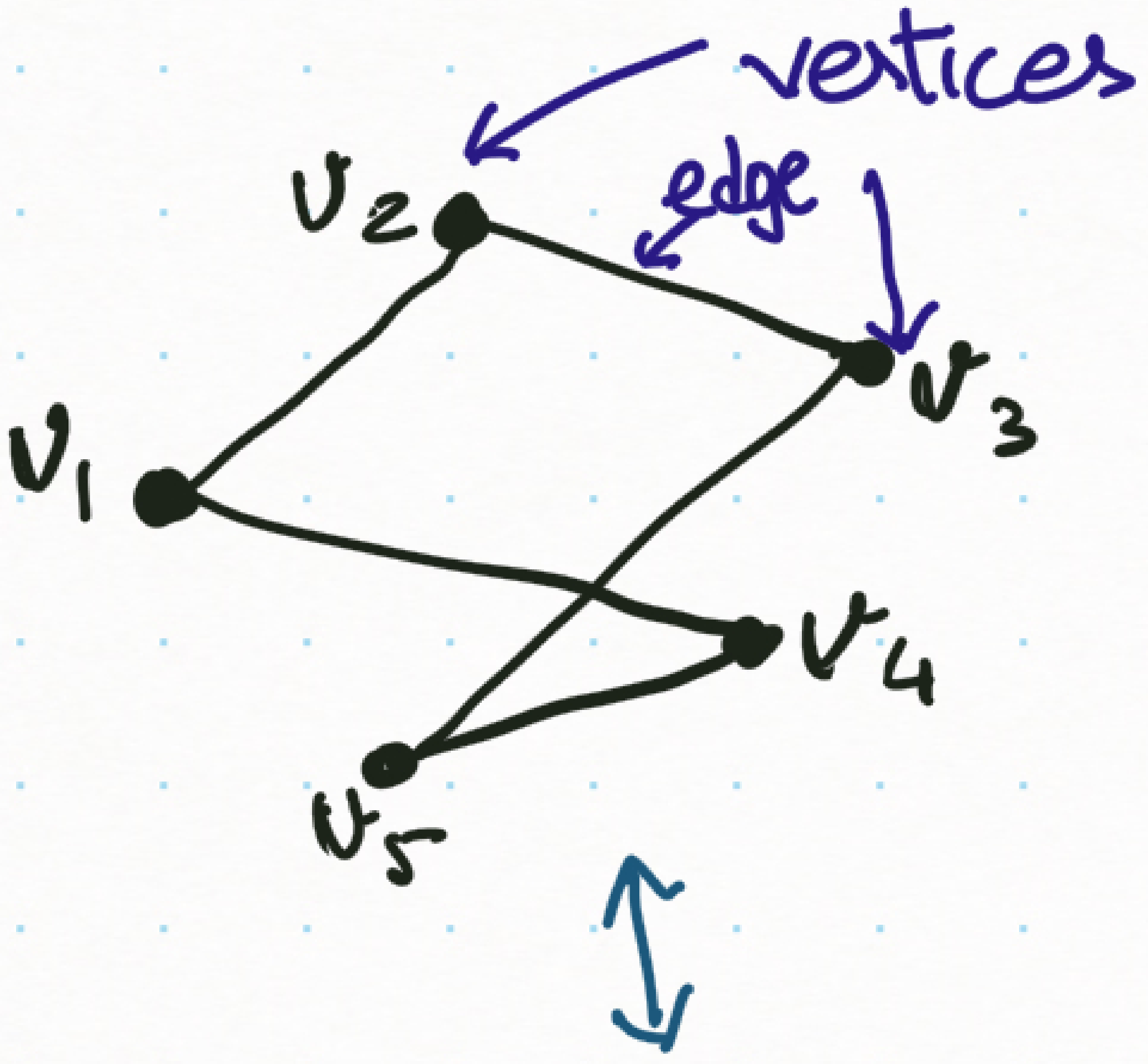
Adjacency Matrix

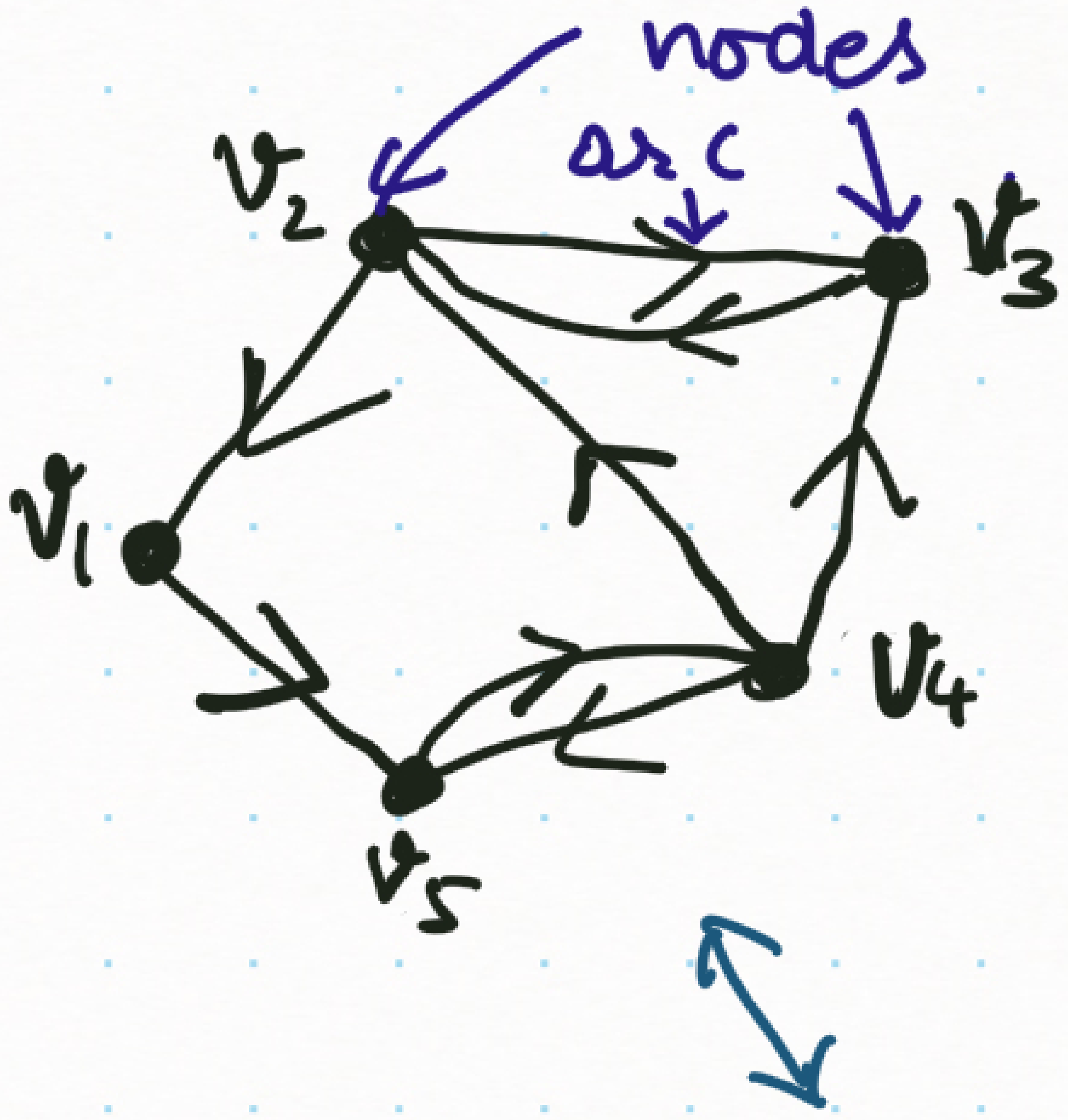
$$\begin{array}{c|ccccc} & v_1 & v_2 & v_3 & v_4 & v_5 \\ \hline v_1 & 0 & 1 & 0 & 1 & 0 \\ v_2 & 1 & 0 & 1 & 0 & 0 \\ v_3 & 0 & 1 & 0 & 0 & 1 \\ v_4 & 1 & 0 & 0 & 0 & 1 \\ v_5 & 0 & 0 & 1 & 1 & 0 \end{array}$$

$$(i,j) = \begin{cases} 1 & \text{if } v_i \leftrightarrow v_j \\ 0 & \text{if } v_i \not\leftrightarrow v_j \end{cases}$$

Adjacency lists ....

Visually





Algebraicly

Adjacency Matrix

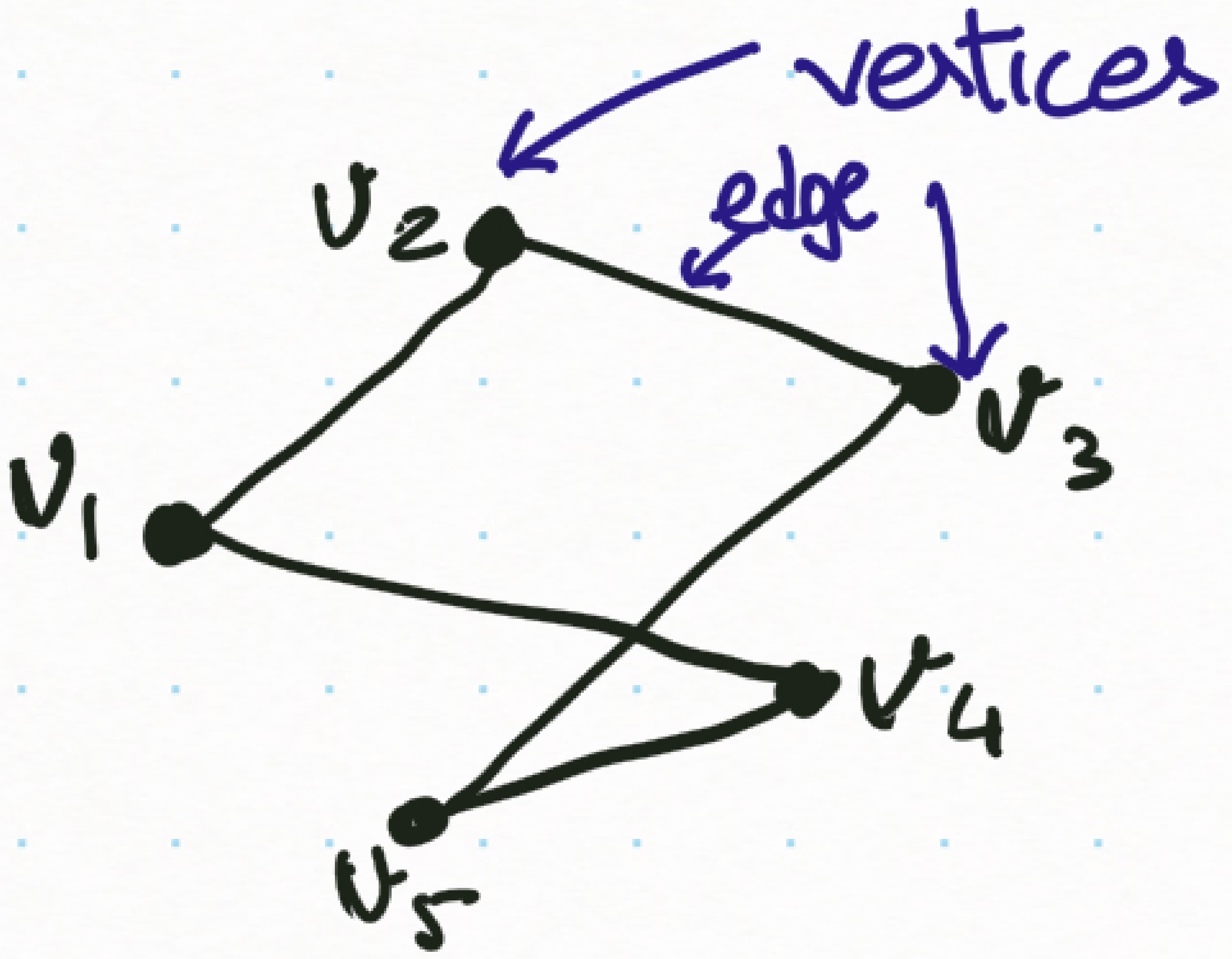
$$\begin{array}{c|ccccc} & v_1 & v_2 & v_3 & v_4 & v_5 \\ \hline v_1 & 0 & 1 & 0 & 1 & 0 \\ v_2 & 1 & 0 & 1 & 0 & 0 \\ v_3 & 0 & 1 & 0 & 0 & 1 \\ v_4 & 1 & 0 & 0 & 0 & 1 \\ v_5 & 0 & 0 & 1 & 1 & 0 \end{array}$$

$$(i,j) = \begin{cases} 1 & \text{if } v_i \Leftrightarrow v_j \\ 0 & \text{if } v_i \not\Leftrightarrow v_j \end{cases}$$

$$\begin{array}{c|ccccc} & v_1 & v_2 & v_3 & v_4 & v_5 \\ \hline v_1 & 0 & 0 & 0 & 0 & 1 \\ v_2 & 1 & 0 & 1 & 0 & 0 \\ v_3 & 0 & 1 & 0 & 0 & 0 \\ v_4 & 0 & 1 & 1 & 0 & 1 \\ v_5 & 0 & 0 & 0 & 1 & 0 \end{array}$$

$$(i,j) = \begin{cases} 1 & \text{if } v_i \rightarrow v_j \\ 0 & \text{if } v_i \not\rightarrow v_j \end{cases}$$

Adjacency lists ....



G graph

consists of $(V(G), E(G))$

$V(G)$ = vertex set

$E(G)$ = edge set

degree of a vertex $d(v) = \#$ edges incident to v

$= |N(v)| = \#$ neighbors of v

e.g. $d(v_1) = 2$

$N(v_1) = \{v_2, v_4\}$

← Neighborhood of $v = \{u : uv \in E(G)\}$

path is a sequence of vertices $u_1, u_2, u_3, .., u_k$ such that there is an edge between each consecutive pair u_i & u_{i+1}

e.g $v_1\, v_2\, v_3\, v_5\, v_4$ is a path

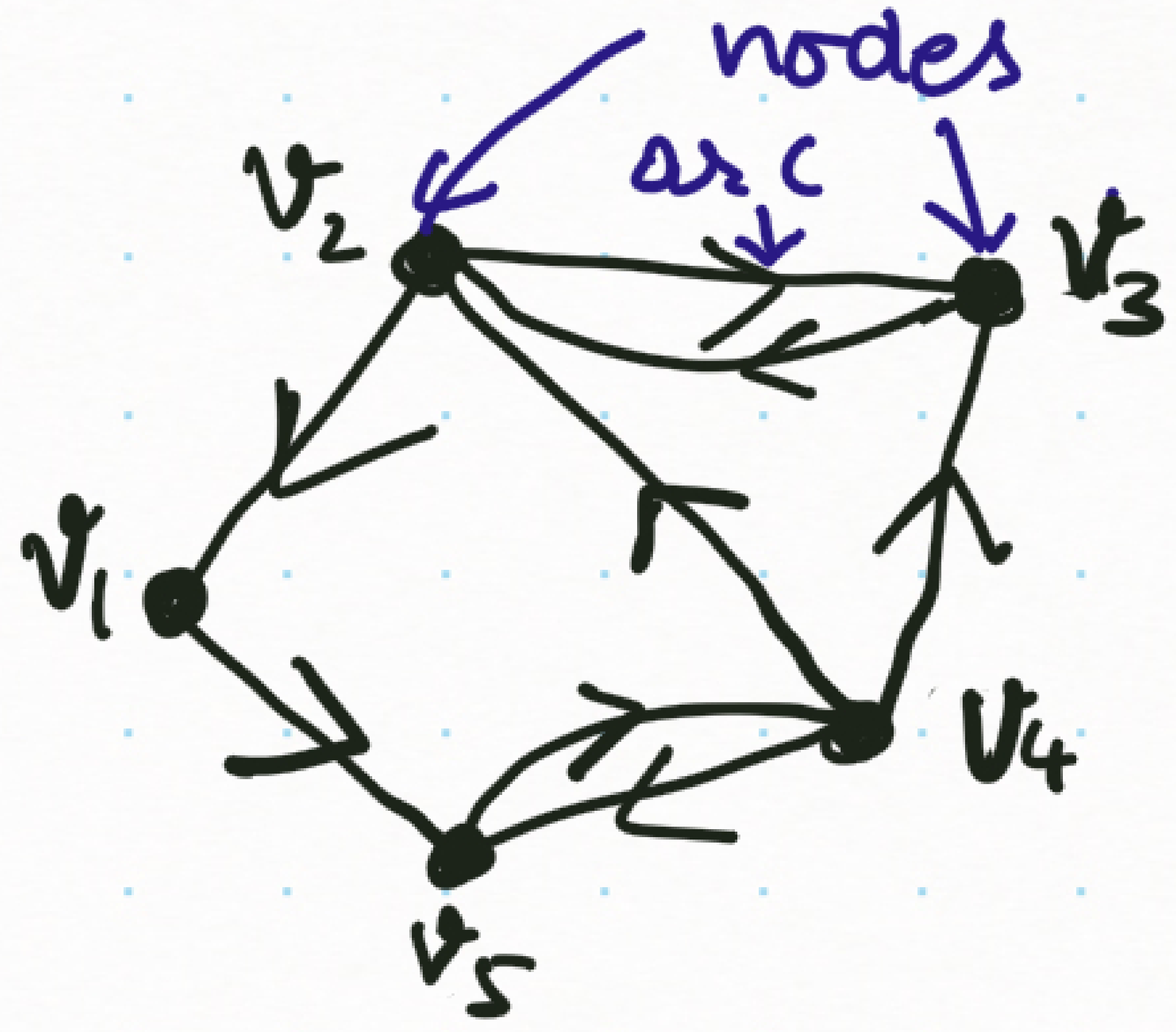
Graph is connected if there a path between every pair of vertices.

Directed graph / Network

$D = (N(D), A(D))$

$N(D)$ = Node set

$A(D)$ = Arc set



Out-degree of a vertex, $d^+(v)$ = # edges outgoing from v

$= |N^+(v)|$ = # out-neighbors of v

Out-Neighborhood of v $= \{u : (v,u) \in A(D)\}$

e.g. $d^+(v_2) = 2$

$N^+(v_2) = \{v_1, v_3\}$

In-degree of v, $d^-(v)$ = # edges incoming to v

$= |N^-(v)|$ = # in-neighbors of v

In-Neighborhood of v $= \{u : (u,v) \in A(D)\}$

e.g. $d^-(v_2) = 2$

$N^-(v_2) = \{v_3, v_4\}$

Directed path: $u_1 \longrightarrow u_2 \longrightarrow u_3 \longrightarrow u_4 \longrightarrow u_5 \cdots$

e.g. $v_1 v_5 - v_4 v_3$

Some examples

Vertices	Edges
individuals in a population	"friendship" "acquaintance" "interaction" "genetic relation" "proximity"
Websites	Direct links
Routers	Direct connection
Facebook accounts	"Friends"
Cities	Direct flight

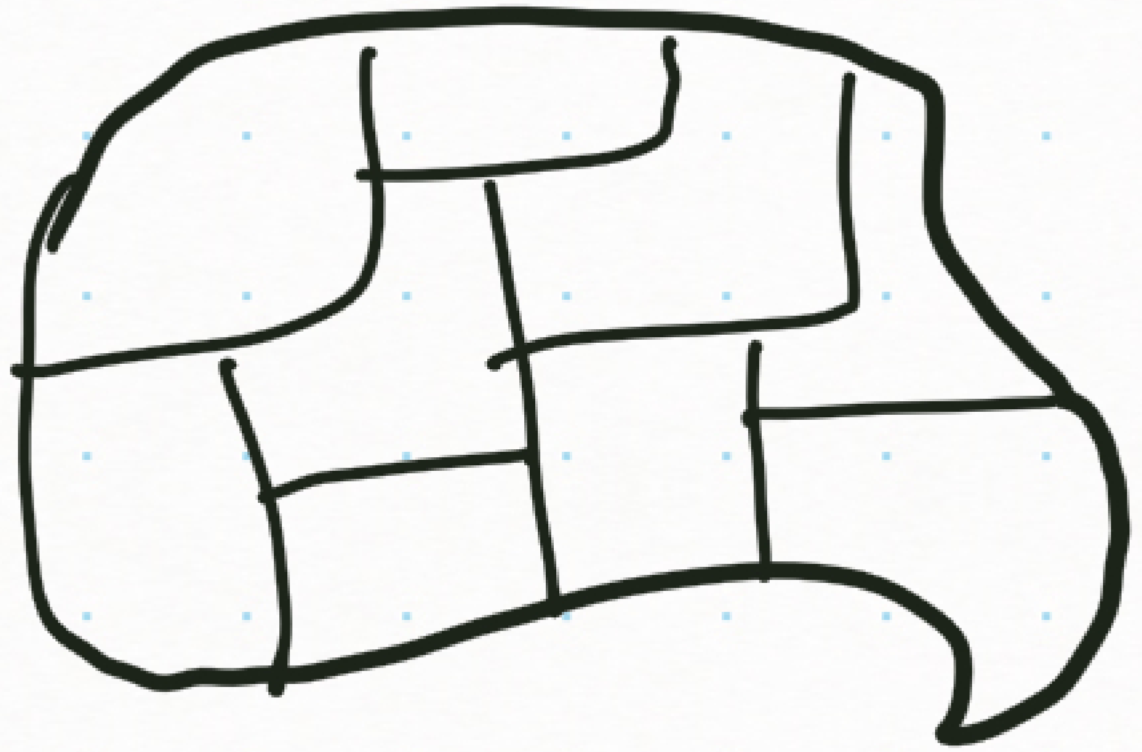
Vertices	Edges
Locations	Road connections
Wireless towers	Proximity
"Jobs"/"processes"	Precedence Orders
Courses	Time conflict
Geographical regions	Common boundary
circuit joints	direct wires
people vs. jobs	Qualified?

Conflict-free allocation of scarce resources

① How many colors (resource) should be used for regions (entities) in a map so that regions with common boundary (conflict) receive different colors ?

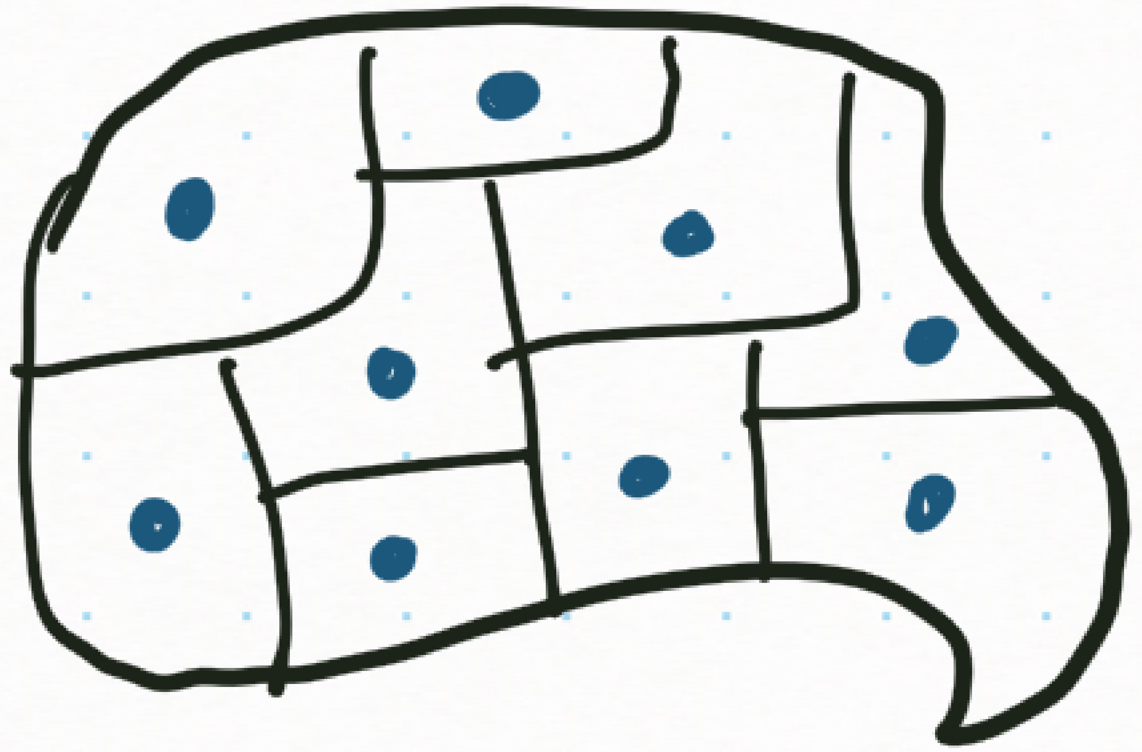
Conflict-free allocation of scarce resources

① How many colors (resource) should be used for regions (entities) in a map so that regions with common boundary (conflict) receive different colors?



Conflict-free allocation of scarce resources

① How many colors (resource) should be used for regions (entities) in a map so that regions with common boundary (conflict) receive different colors?

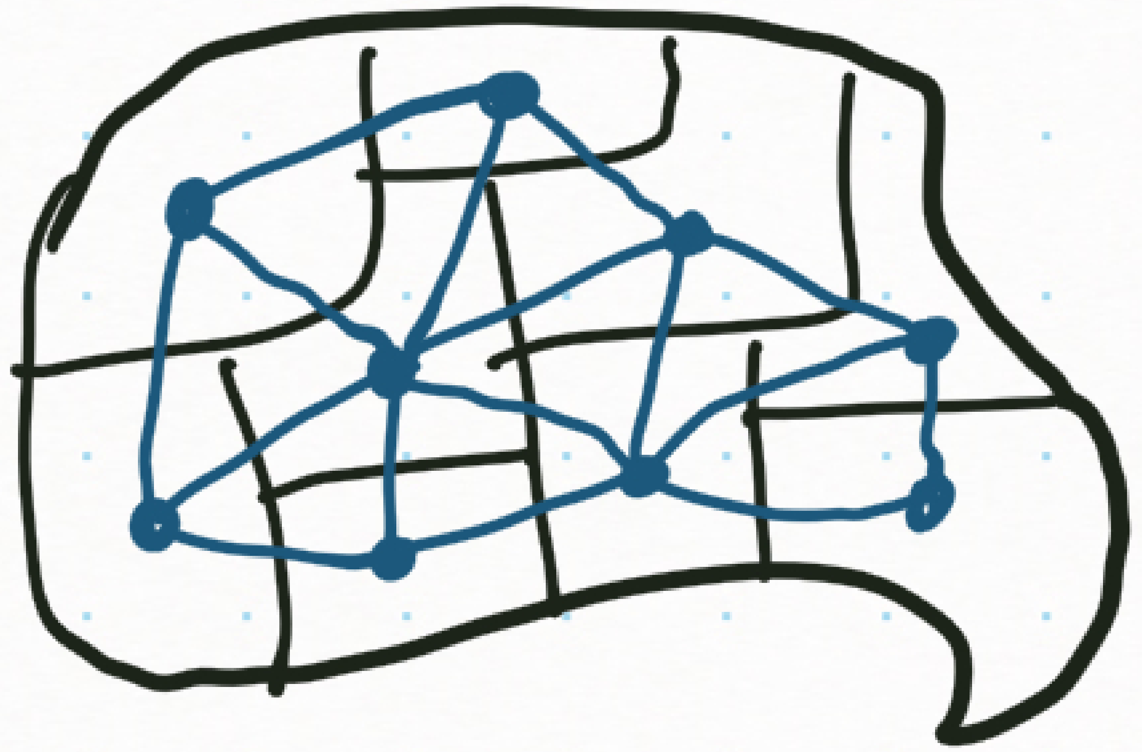


entities ⟷ vertices

conflicts ⟷ edges

Conflict-free allocation of scarce resources

① How many colors (resource) should be used for regions (entities) in a map so that regions with common boundary (conflict) receive different colors?

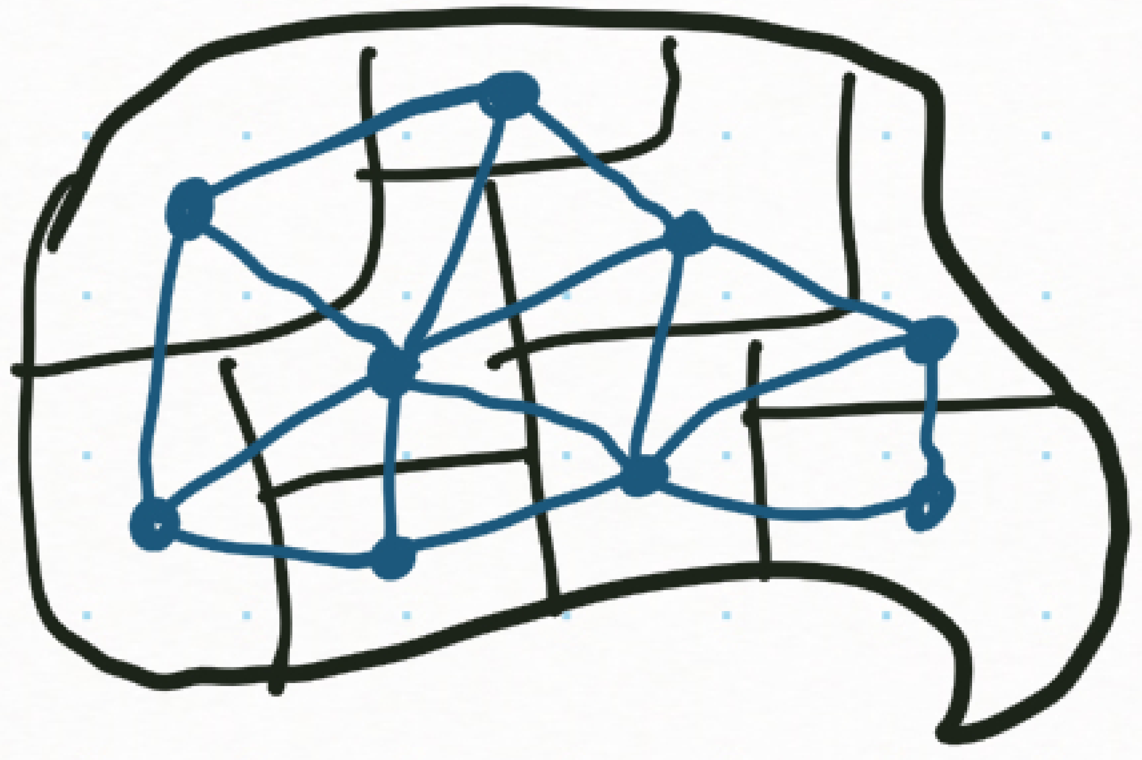


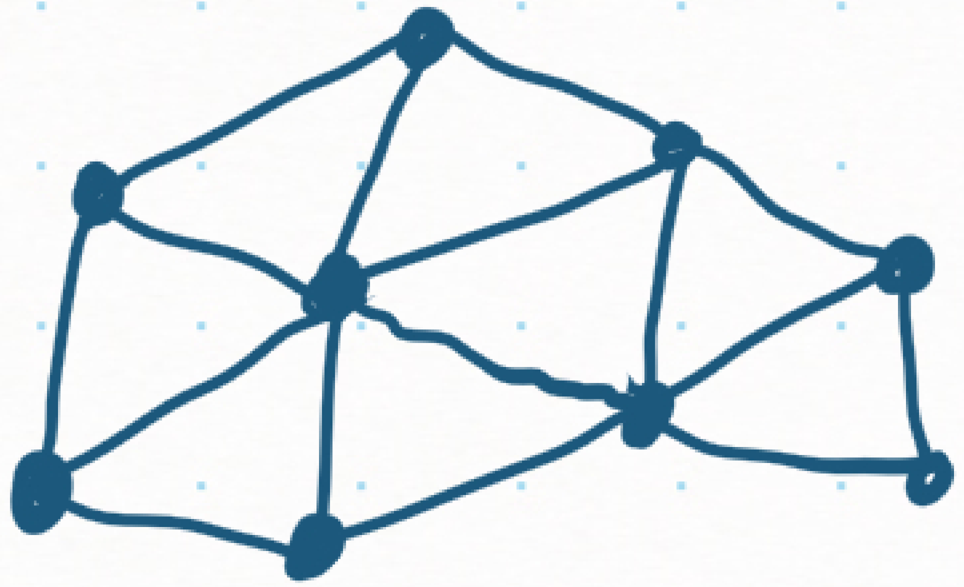
entities ⟷ vertices

conflicts ⟷ edges

Conflict-free allocation of scarce resources

① How many colors (resource) should be used for regions (entities) in a map so that regions with common boundary (conflict) receive different colors?





entities ⟷ vertices

conflicts ⟷ edges

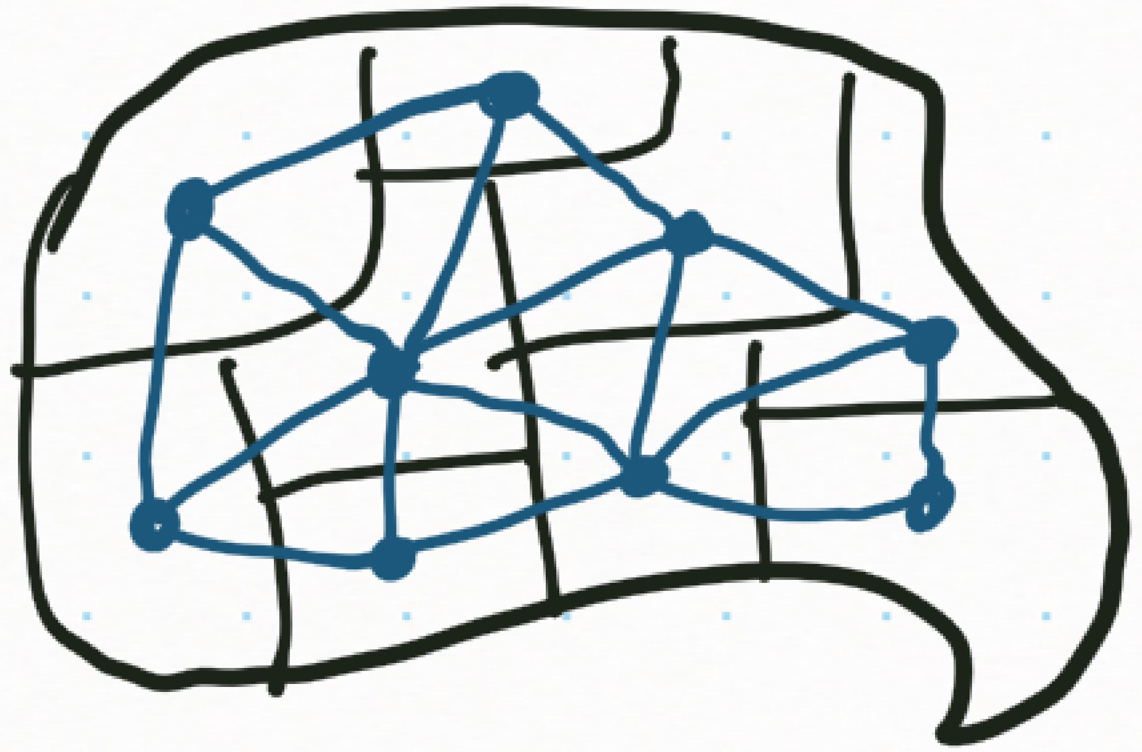
Assign colors to vertices so that vertices with an edge between them get different colors.

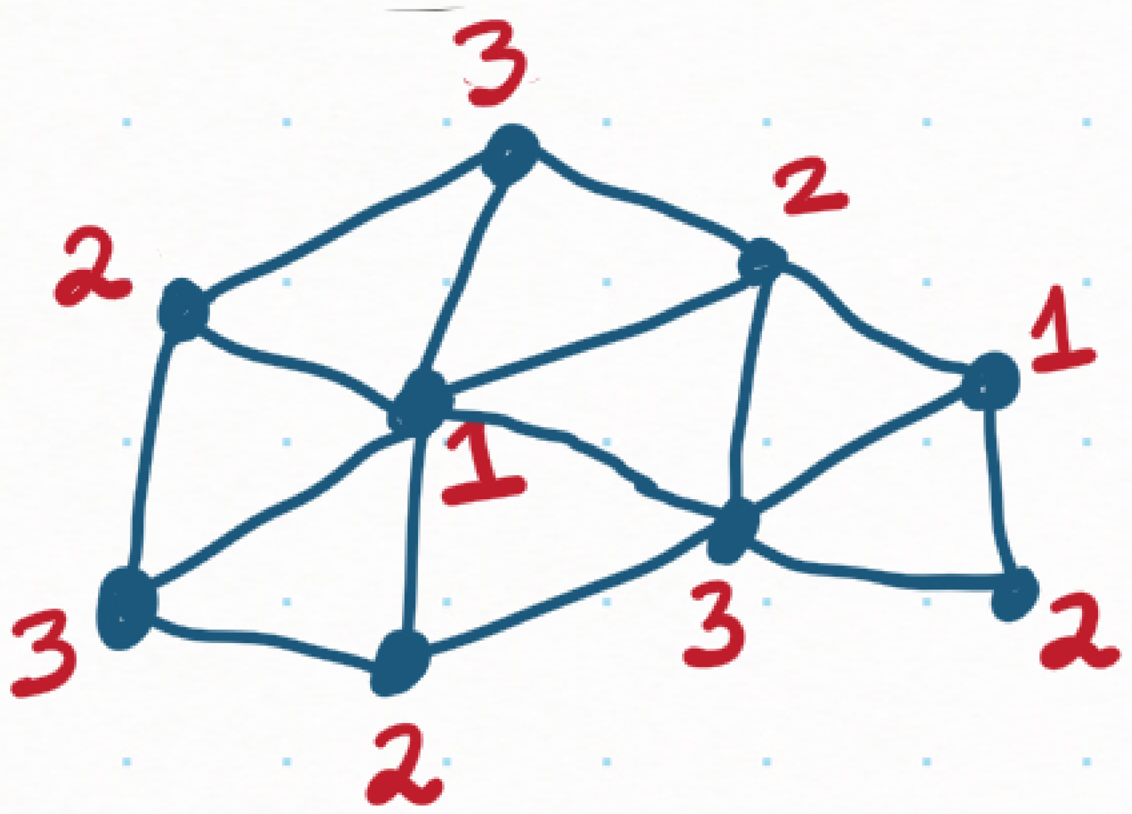
↕

resources

Conflict-free allocation of scarce resources

① How many colors (resource) should be used for regions (entities) in a map so that regions with common boundary (conflict) receive different colors?





entities ↔ vertices

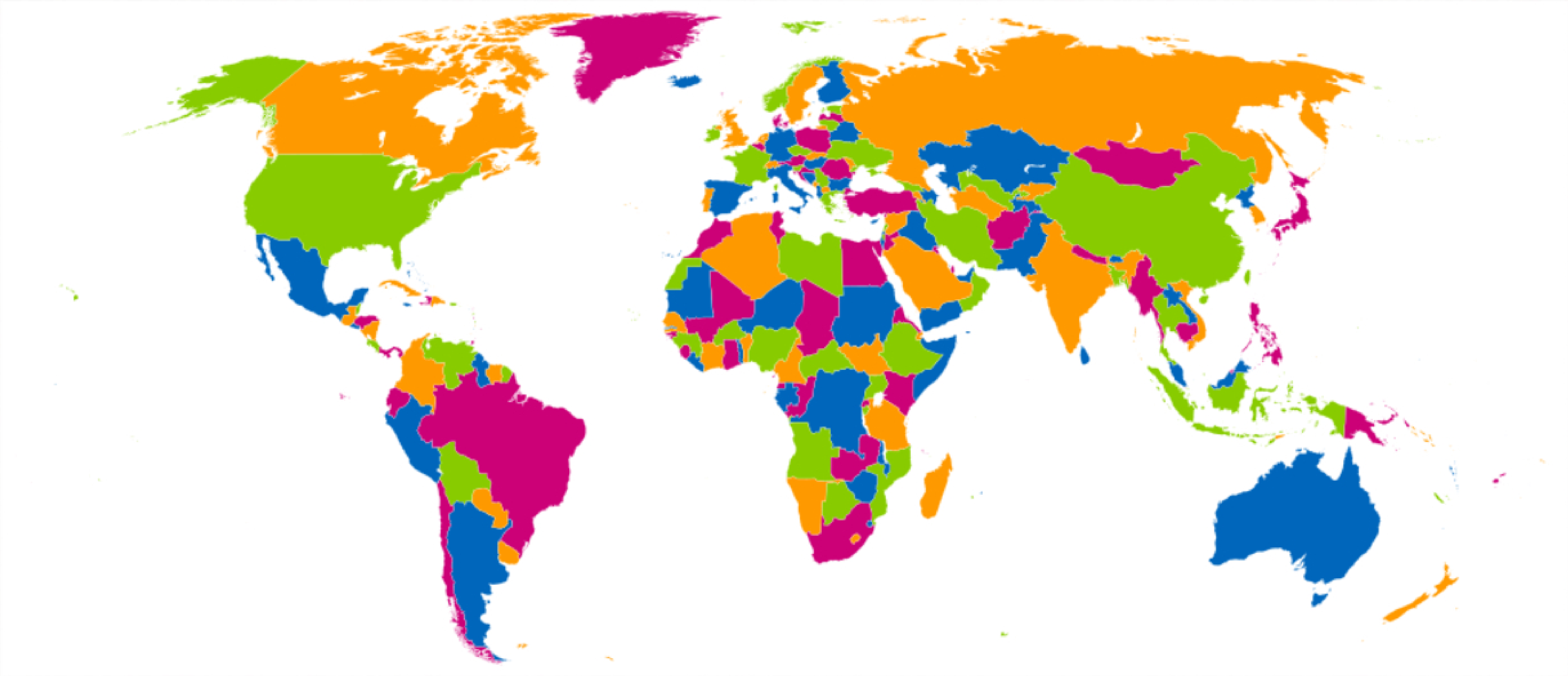
conflicts ↔ edges

Assign colors to vertices so that vertices with an edge between them get different colors.

↕

resources

Can we use less than 3 colors?



Here regions ≡ countries of the world

Colored using 4 colors, as guaranteed by the Four Color Thm.

Look this up.

Conflict-free allocation of scarce resources

② Allocate radio channels (resource) to radio stations (entities) so that stations with proximity interference (conflict) get different channels.

entities ↔ vertices

conflicts ↔ edges

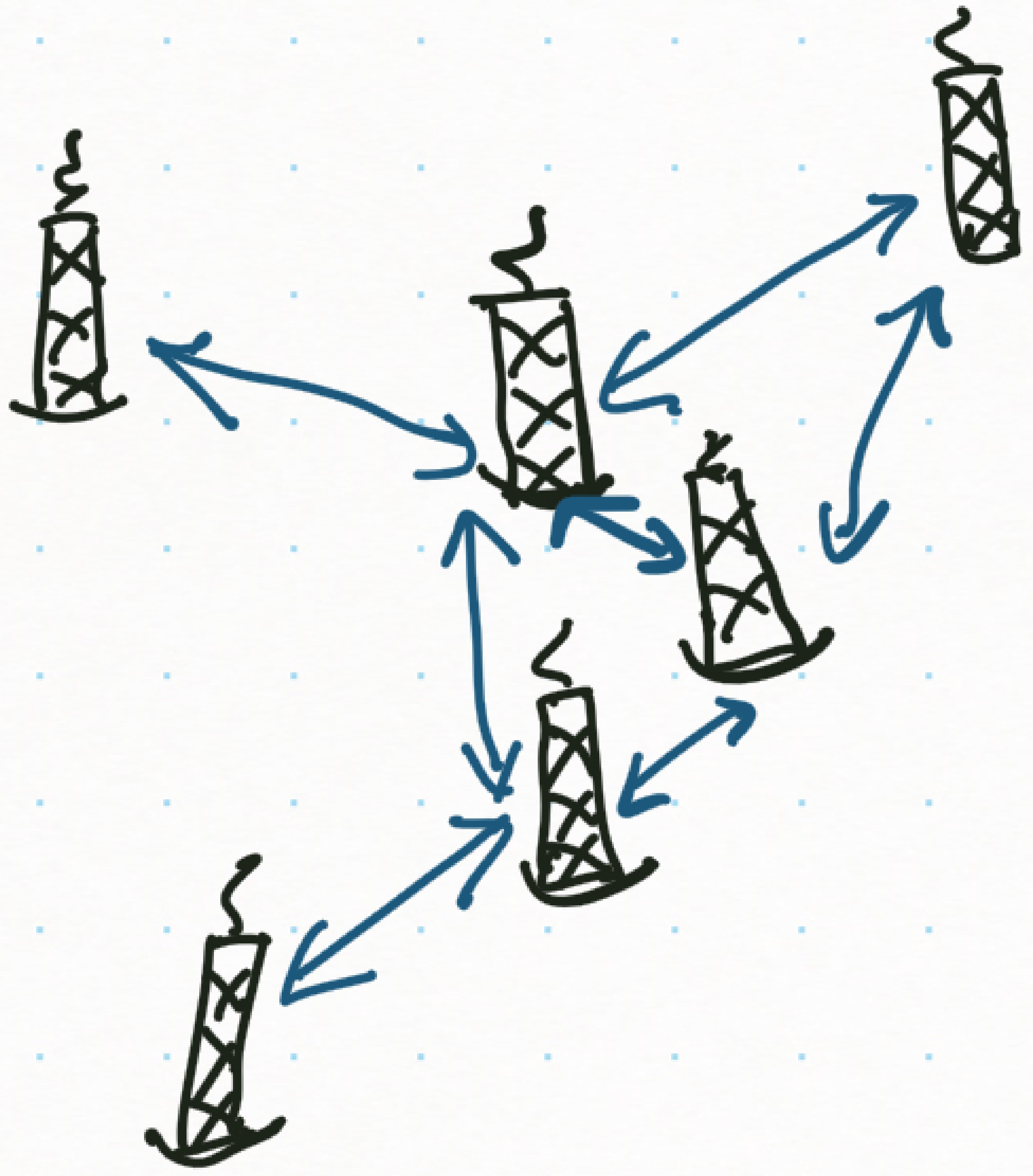
Assign colors to vertices so that vertices with an edge between them get different colors.

↕

resources

Conflict-free allocation of scarce resources

② Allocate radio channels (resource) to radio stations (entities) so that stations with proximity interference (conflict) get different channels.



entities ⟷ vertices

conflicts ⟷ edges

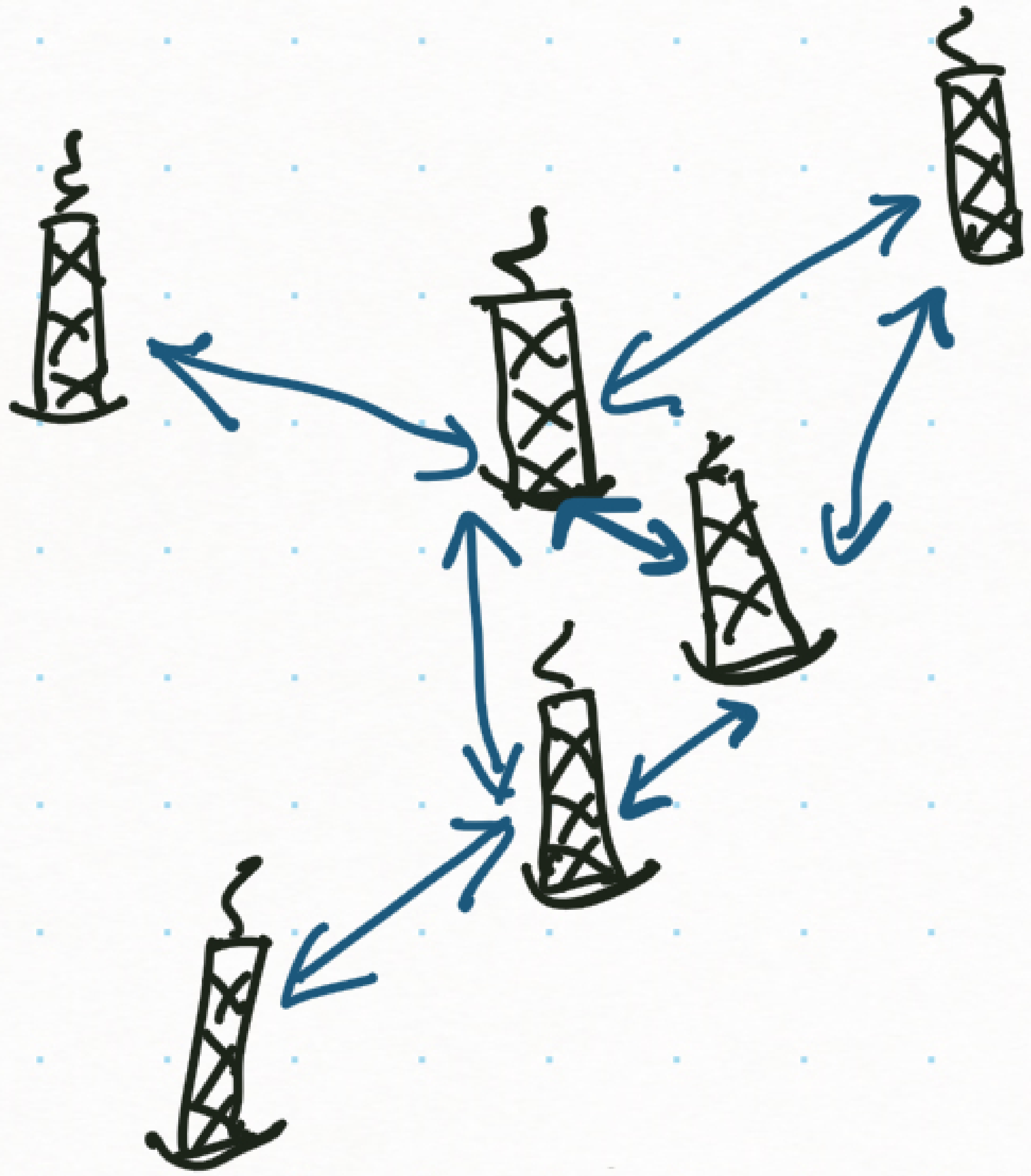
Assign colors to vertices so that vertices with an edge between them get different colors.

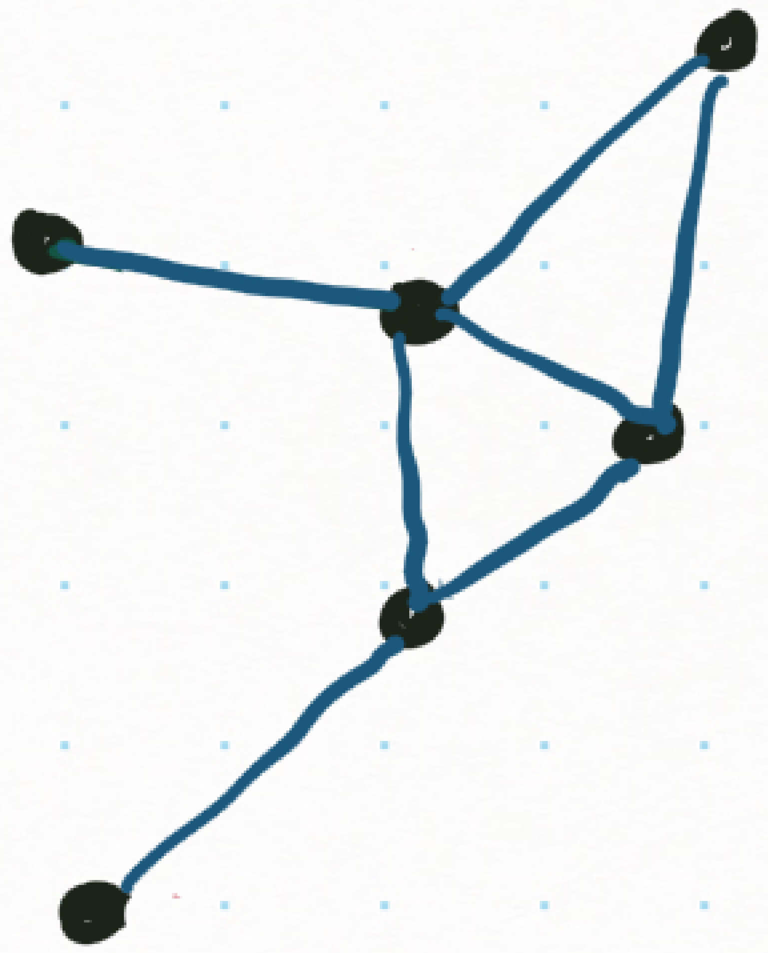
↕

resources

Conflict-free allocation of scarce resources

② Allocate radio channels (resource) to radio stations (entities) so that stations with proximity interference (conflict) get different channels.





entities ↔ vertices

conflicts ↔ edges

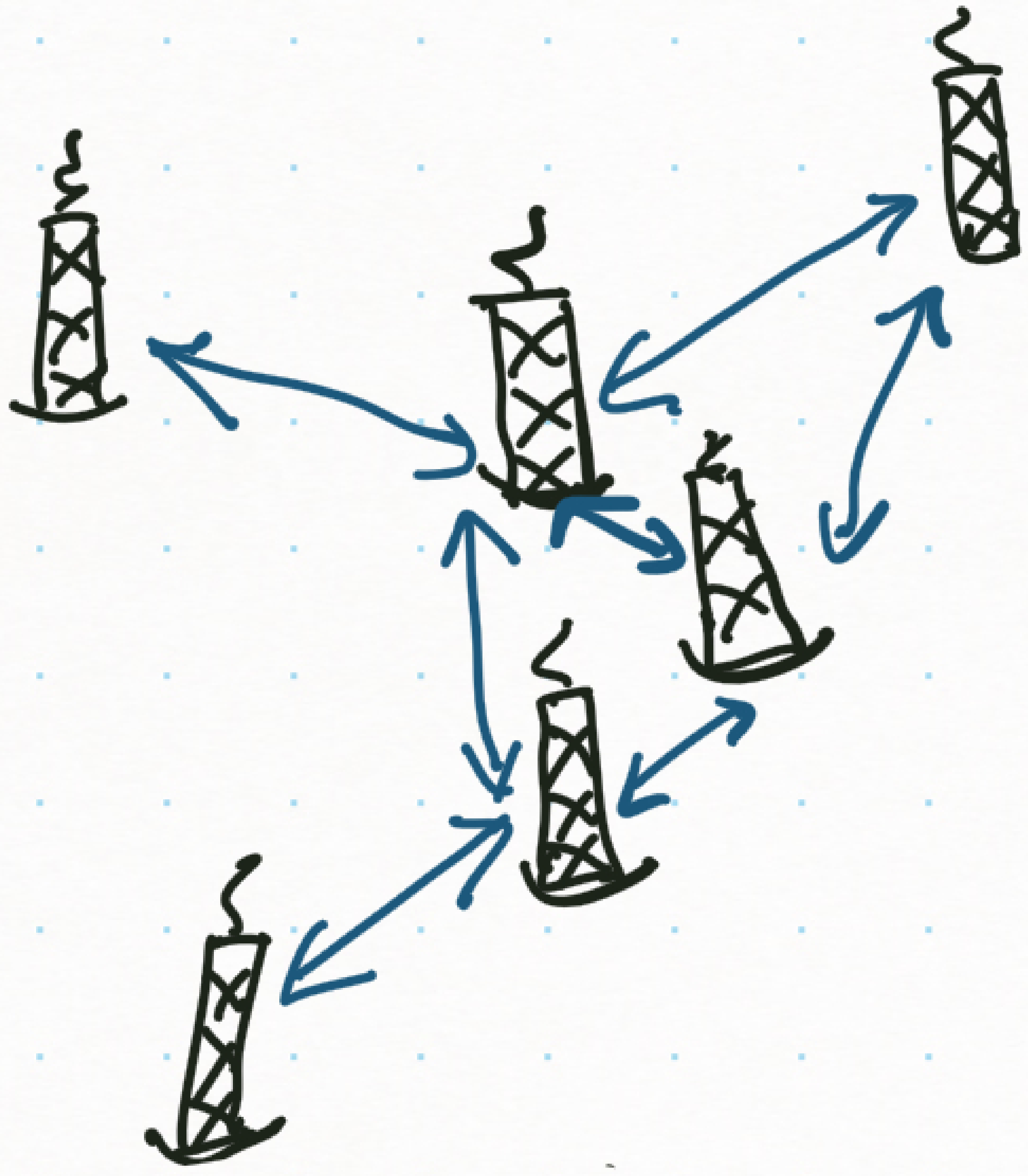
Assign colors to vertices so that vertices with an edge between them get different colors.

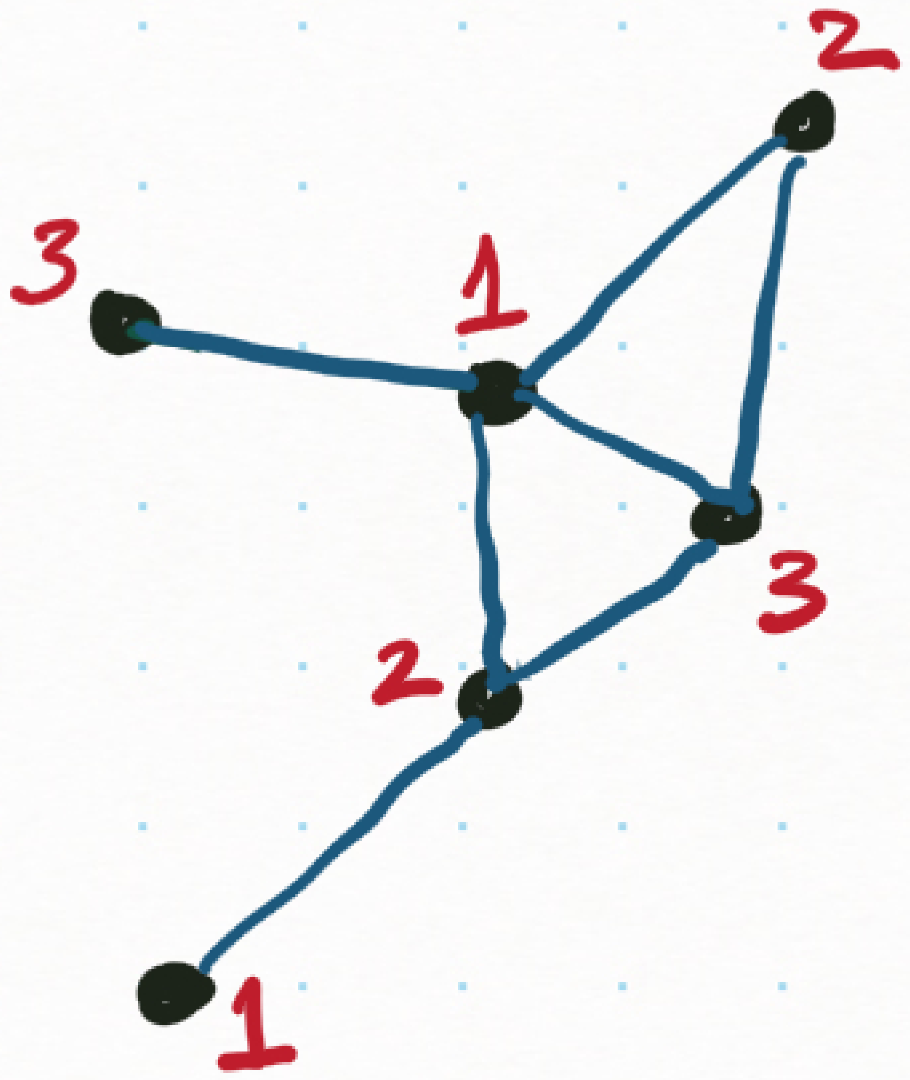
↕

resources

Conflict-free allocation of scarce resources

② Allocate radio channels (resource) to radio stations (entities) so that stations with proximity interference (conflict) get different channels.





entities ↔ vertices
conflicts ↔ edges

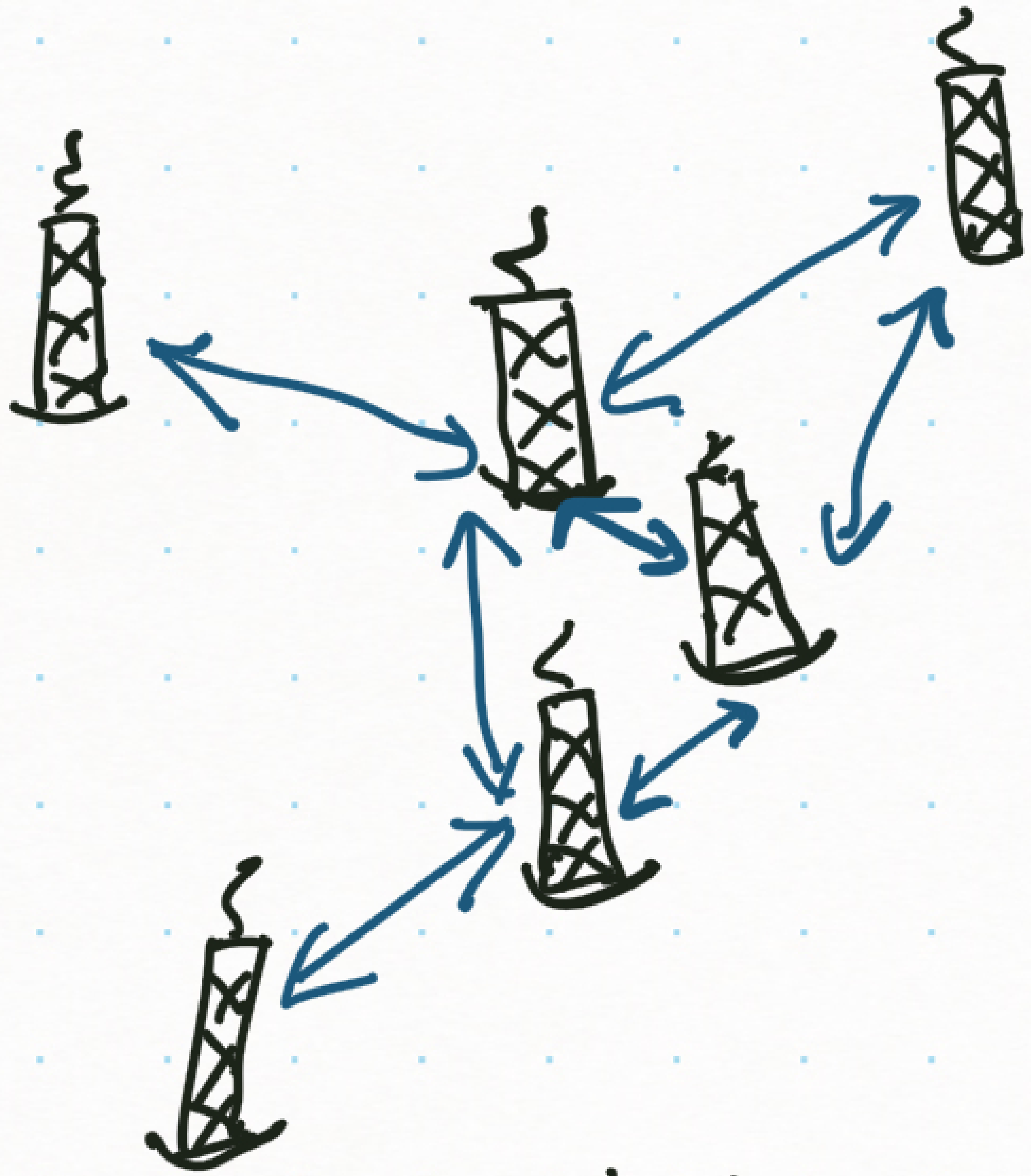
Assign colors to vertices so that vertices with an edge between them get different colors.

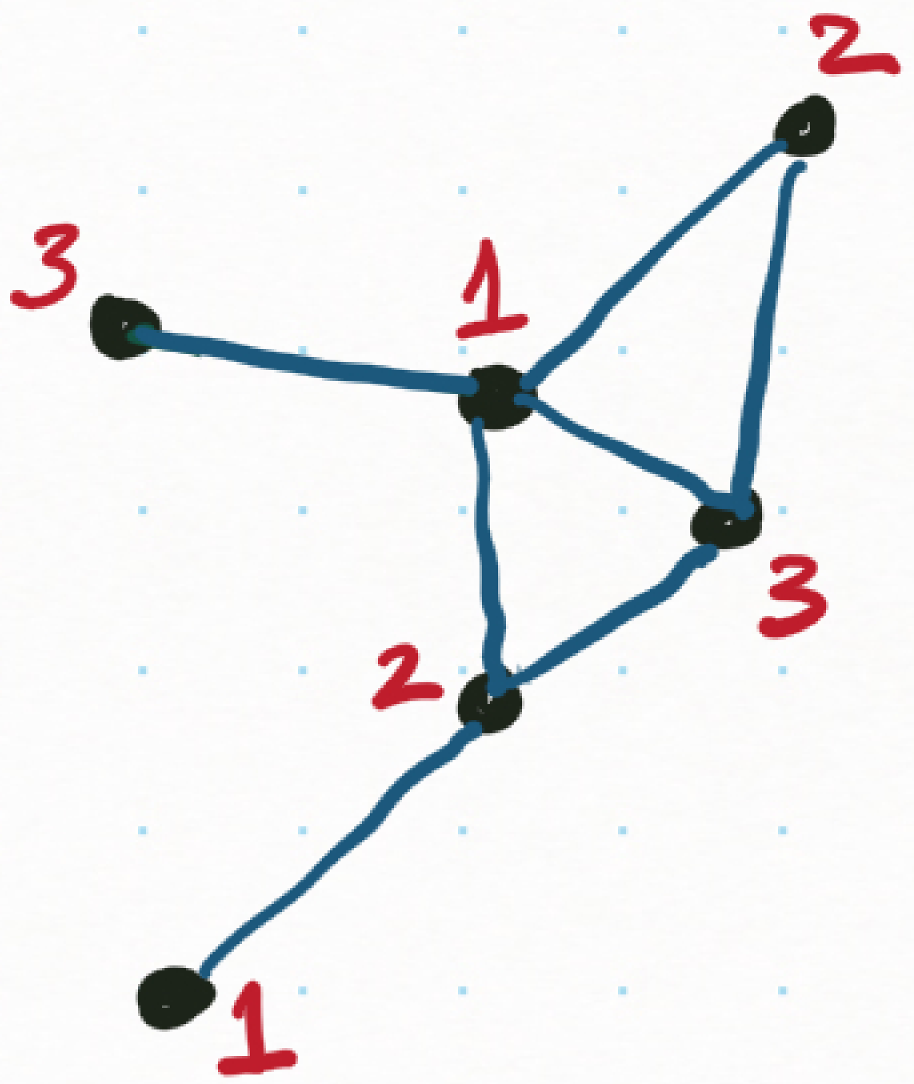
↕

resources

Conflict-free allocation of scarce resources

② Allocate radio channels (resource) to radio stations (entities) so that stations with proximity interference (conflict) get different channels.





entities ⟷ vertices

conflicts ⟷ edges

Assign colors to vertices so that vertices with an edge between them get different colors.

↕

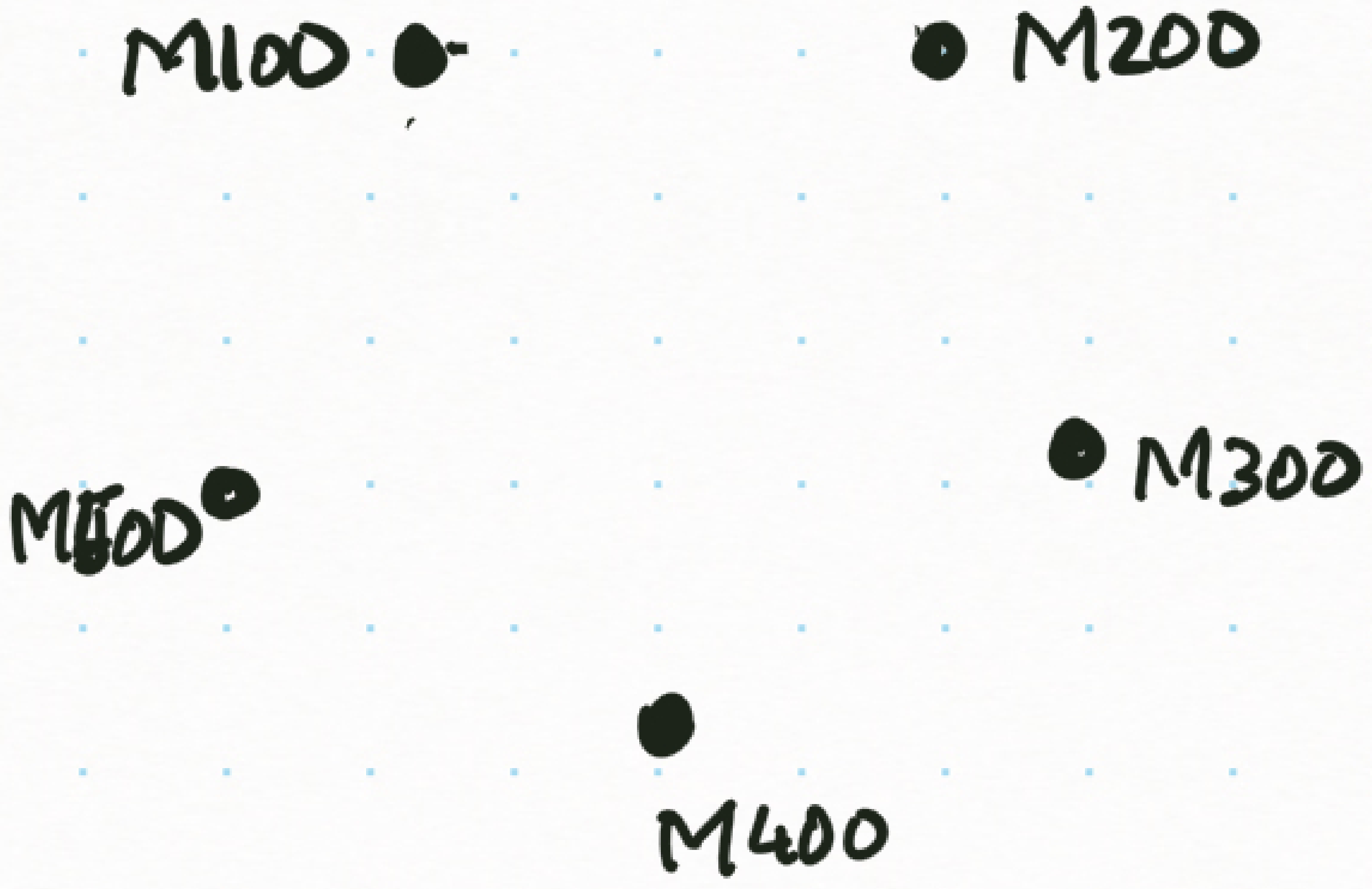
resources

Data in form of a distance matrix between pairs of radio stations ⟶ Structure of the interference graph in form of an adjacency matrix

Conflict-free allocation of scarce resources

③ Allocate classrooms (resource) to courses (entities) so that courses with overlapping-time (conflict) are given different rooms.

Courses	Time
M100	10-11am
M200	10:30-11:30am
M300	12-1pm
M400	12:30-1:30pm
M500	10am-2pm



entities ↔ vertices

conflicts ↔ edges

Assign colors to vertices so that vertices with an edge between them get different colors.

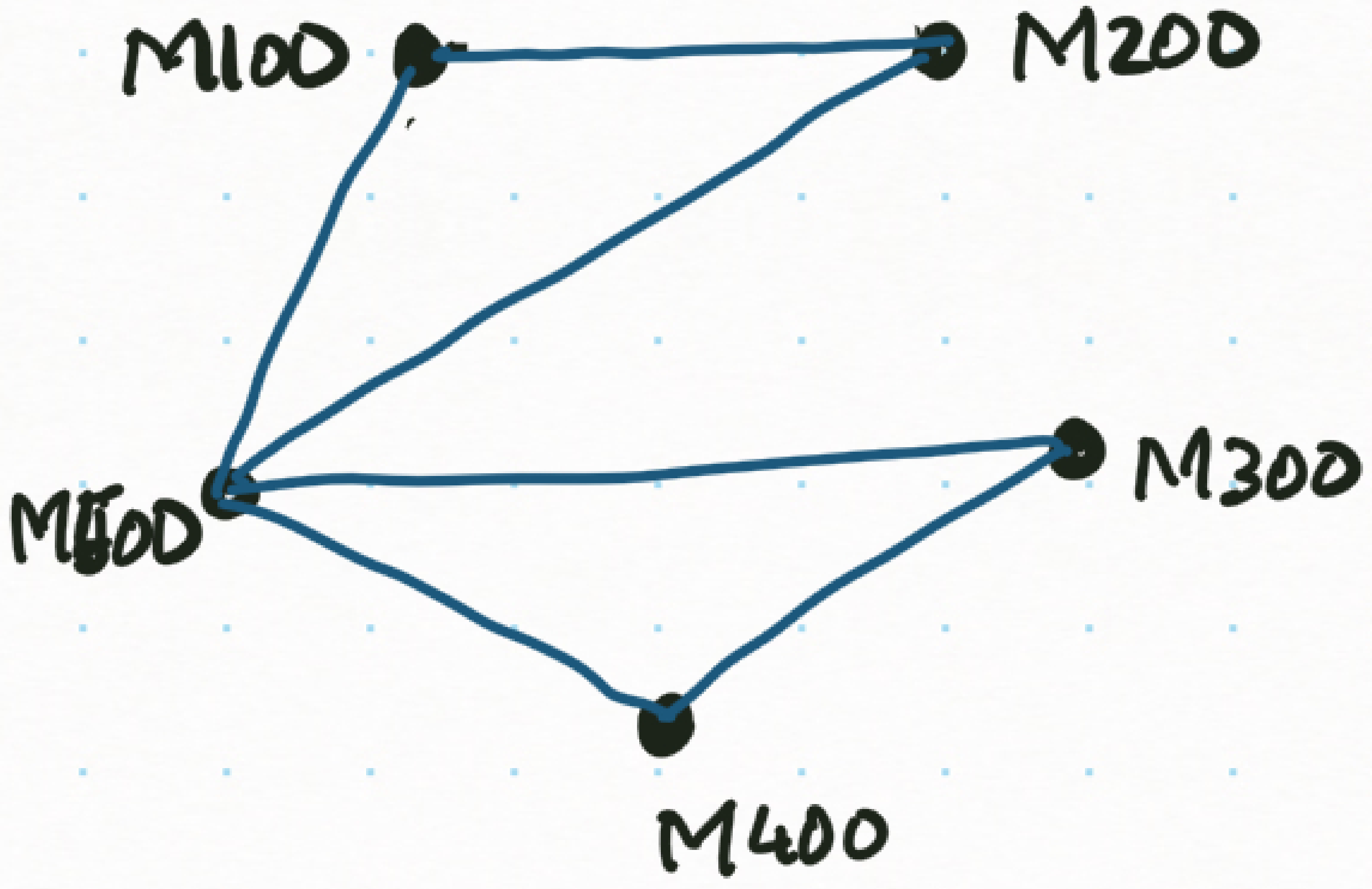
↕

resources

Conflict-free allocation of scarce resources

③ Allocate classrooms (resource) to courses (entities) so that courses with overlapping-time (conflict) are given different rooms.

Courses	Time
M100	10-11am
M200	10:30-11:30am
M300	12-1pm
M400	12:30-1:30pm
M500	10am-2pm



entities ↔ vertices

conflicts ↔ edges

Assign colors to vertices so that vertices with an edge between them get different colors.

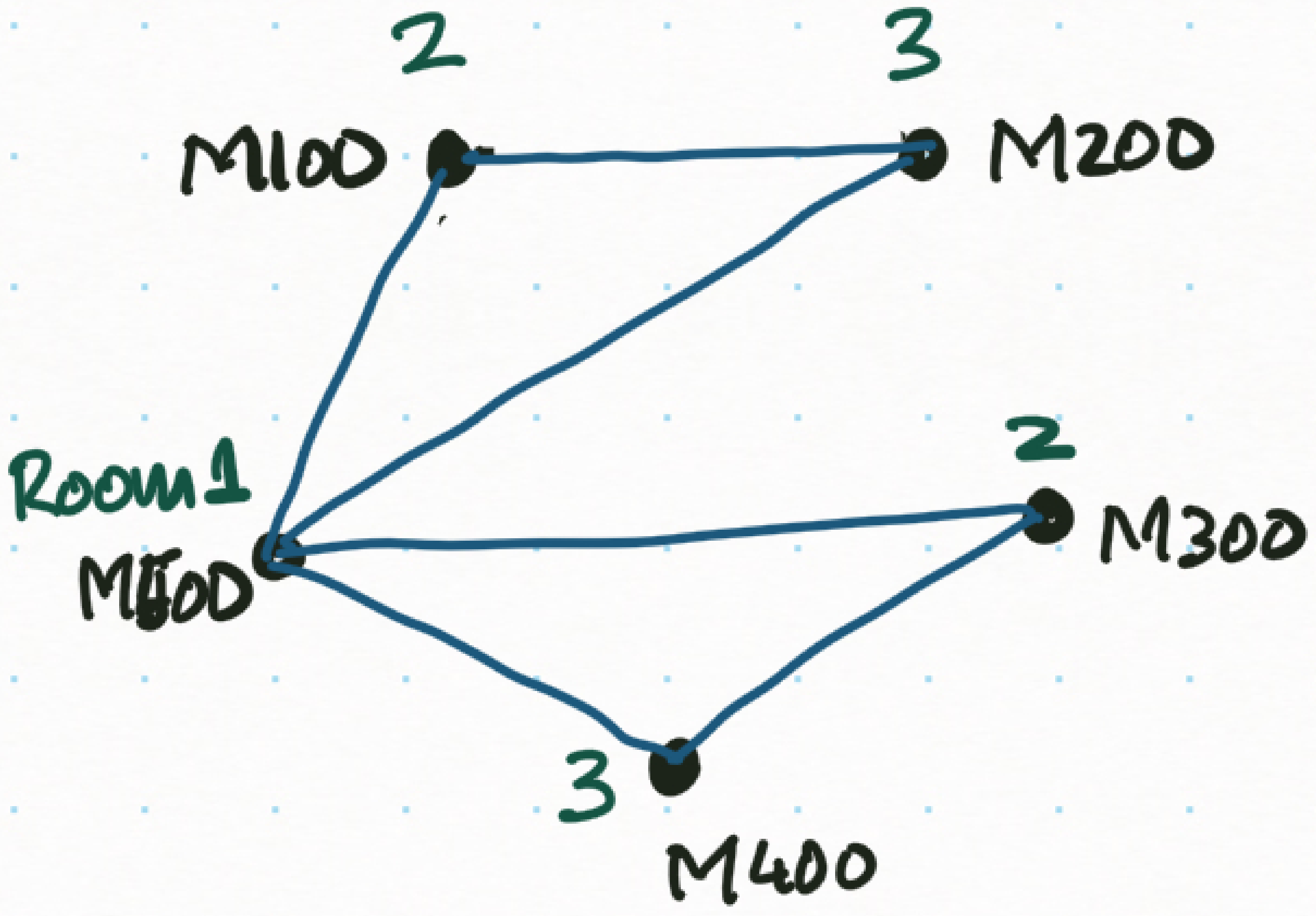
↕

resources

Conflict-free allocation of scarce resources

③ Allocate classrooms (resource) to courses (entities) so that courses with overlapping-time (conflict) are given different rooms.

Courses	Time
M100	10-11am
M200	10:30-11:30am
M300	12-1pm
M400	12:30-1:30pm
M500	10am-2pm



entities ↔ vertices

conflicts ↔ edges

Assign colors to vertices so that vertices with an edge between them get different colors.

↕

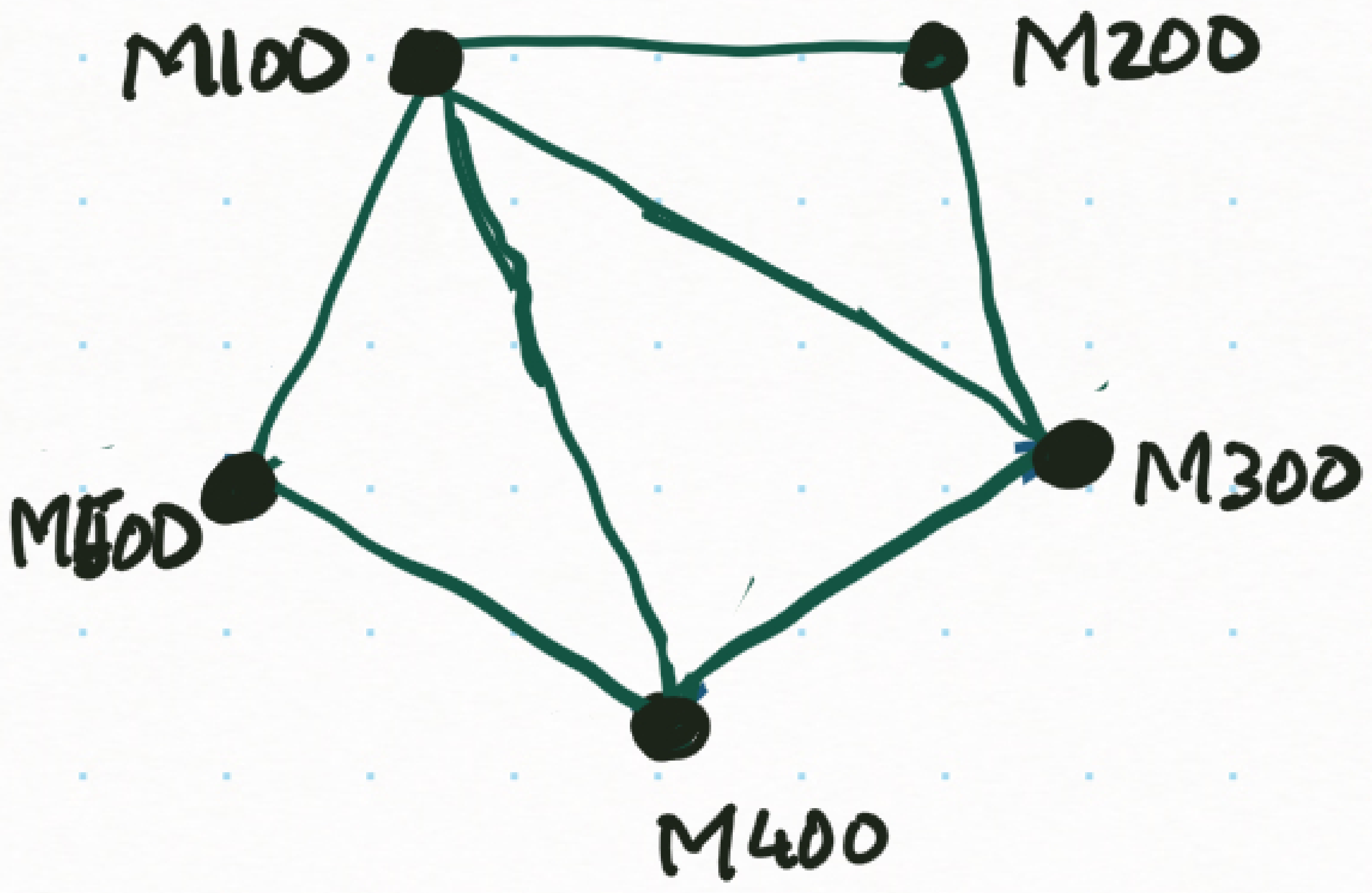
resources

Conflict-free allocation of scarce resources

④ Allocate Final Exam time-slots (resource) to courses (entities) so that courses with common students (conflict) are given different time-slots.

Courses	Students
M100	A, B, C, D, E
M200	A, B, E
M300	B, D, E
M400	C, D
M500	C

Anna, Bob, Cathy, Don, Edith



entities ↔ vertices

conflicts ↔ edges

Assign colors to vertices so that vertices with an edge between them get different colors.

↕

resources

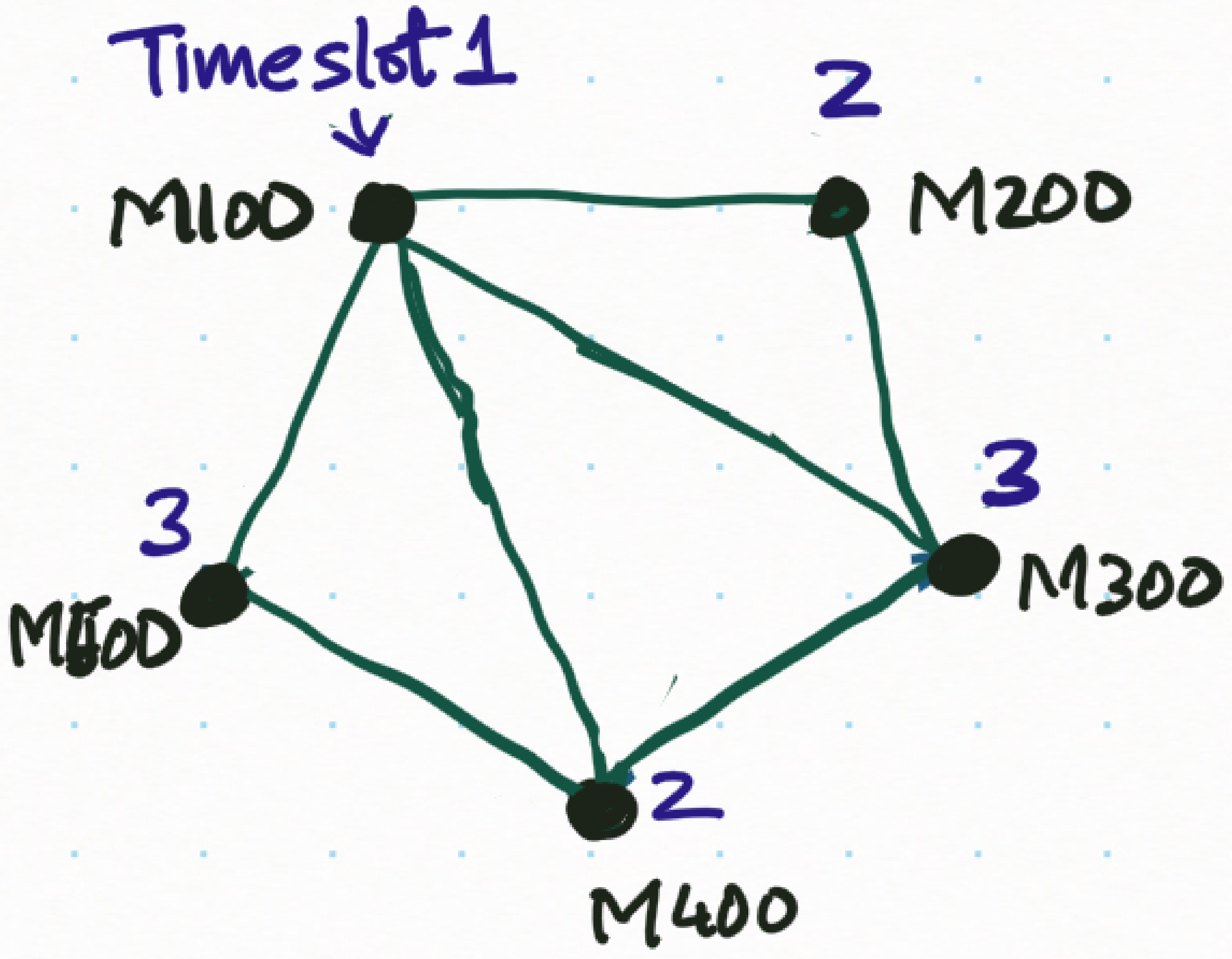
Conflict-free allocation of scarce resources

④ Allocate time-slots (resource) to courses (entities) so that courses with common students (conflict) are given different time-slots.

Final Exam

Courses	Students
M100	A, B, C, D, E
M200	A, B, E
M300	B, D, E
M400	C, D
M500	C

Anna, Bob, Cathy, Don, Edith



Timeslot #1: 10am–12pm
Time slot #2: 2pm–4pm
Time slot #3: 6pm–8pm

entities ↔ vertices
conflicts ↔ edges

Assign colors to vertices so that vertices with an edge between them get different colors.

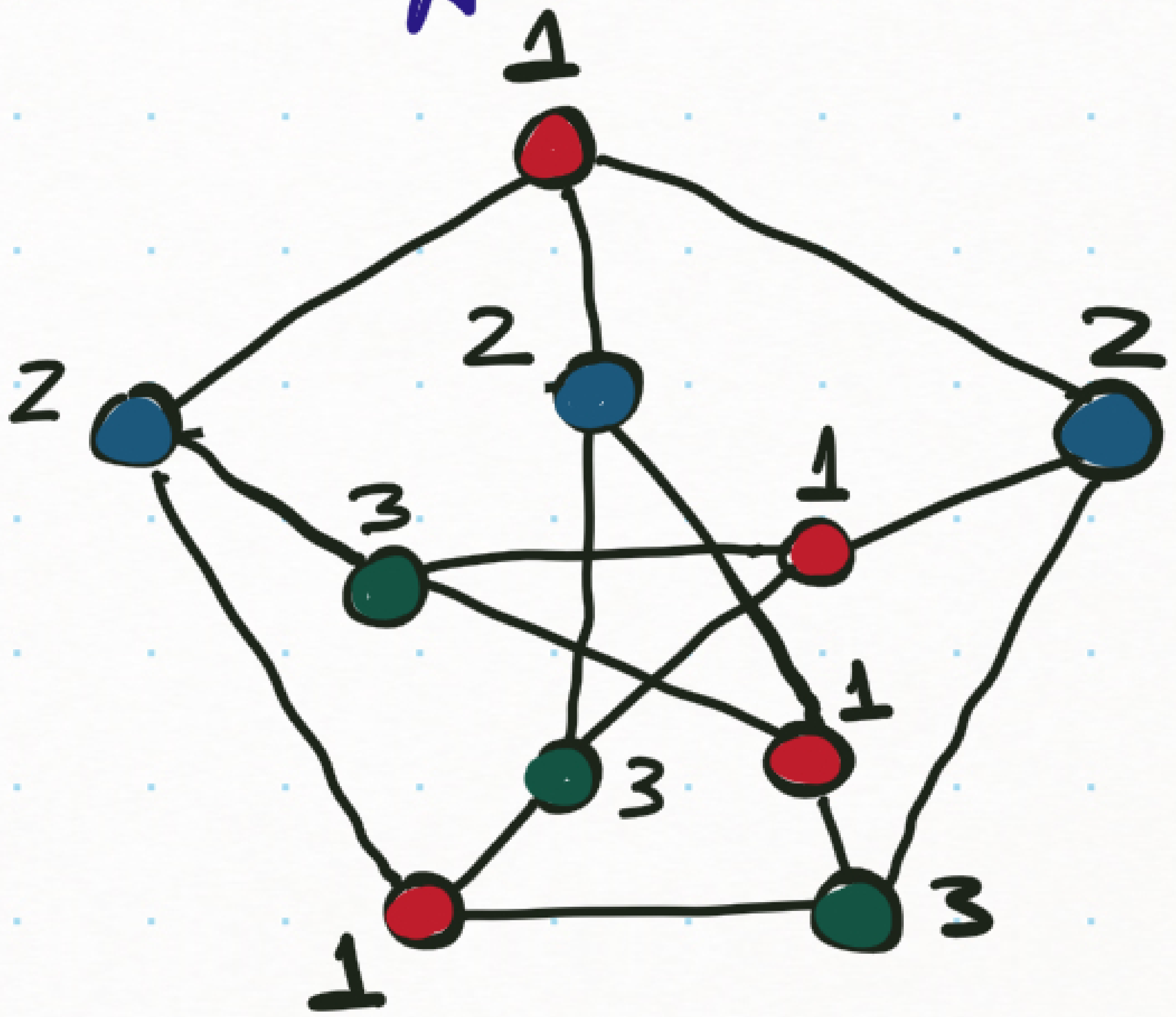
↕

resources

Proper coloring

We color vertices in such a way that any pair of vertices with an edge between them must receive different colors.

Vertices receiving the same color have no edges in between them (conflict-free allocation of resource)

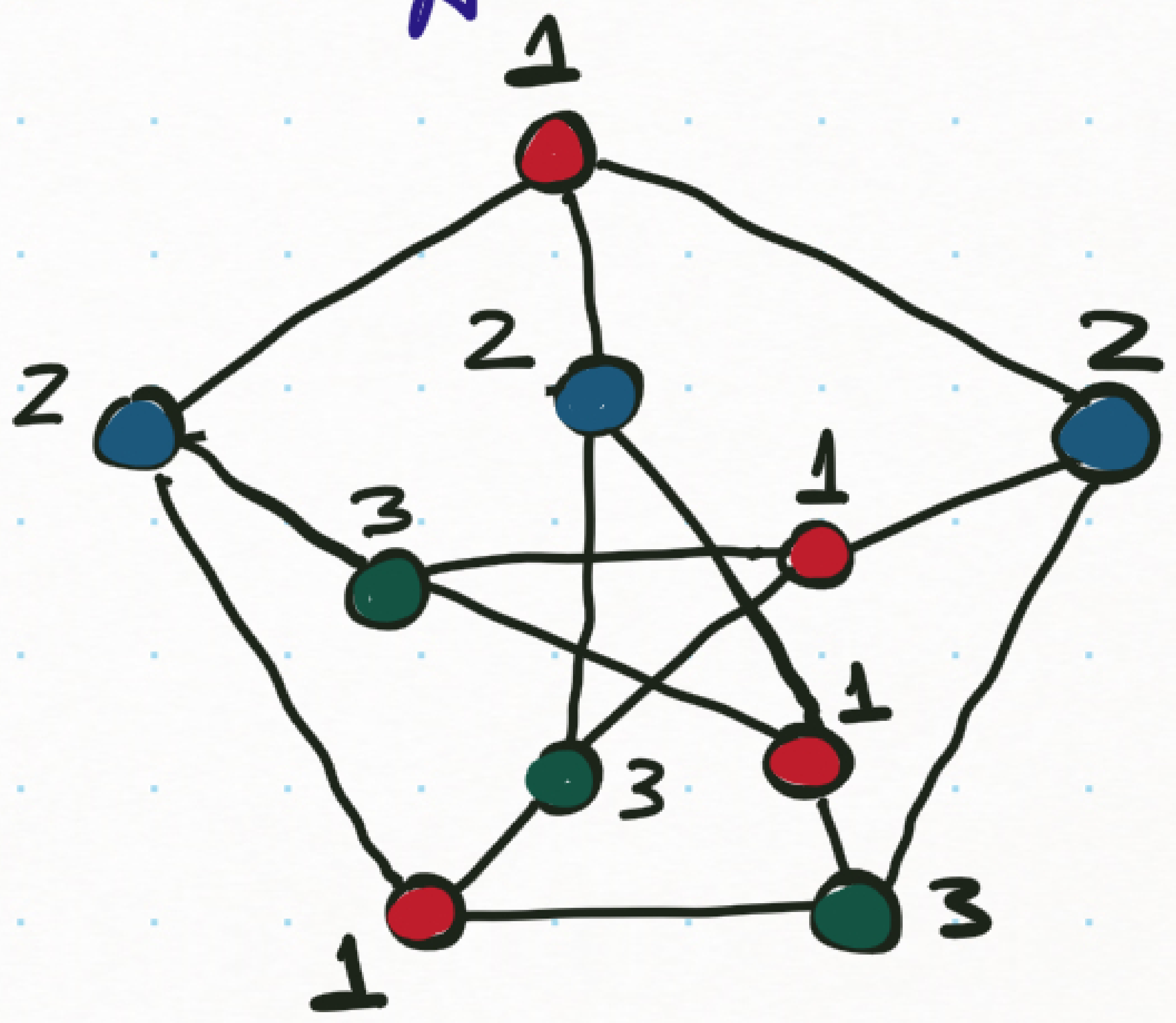


We are partitioning the vertices into color classes, each are independent sets (no edges within).

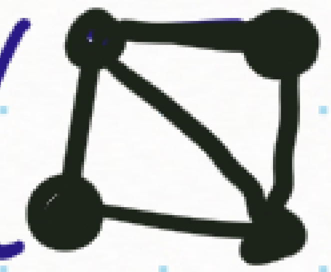
Proper coloring

We color vertices in such a way that any pair of vertices with an edge between them must receive different colors.

Vertices receiving the same color have no edges in between them (conflict-free allocation of resource)



We are partitioning the vertices into color classes, each are independent sets (no edges within).

The least number of colors need for a graph G, is called the chromatic number of G, $\chi(G)$. e.g. χ() $=3$