MATH 380

Hemanshu Kaul

kaul@iit.edu

Optimization based decision making

Variables: $\vec{x} = (x_1, \ldots, x_n)$ ← decisions to be made

reals, integers, binary

chosen so that

objective functions

"Optimize" $\underbrace{f_1(\vec{x}), f_2(\vec{x}), \ldots, f_m(\vec{x})}$

maximize or minimize

subject to "constraints"

$$\left.\begin{aligned} g_1(\vec{x}) &\geq b_1 \\ g_2(\vec{x}) &\leq b_2 \\ g_3(\vec{x}) &= b_3 \\ &\vdots \\ g_k(\vec{x}) &\gtreqless b_k \end{aligned}\right\}$$

constraints which can be a mix of "$\geq$", "$\leq$", "$=$" inequalities

A linear Optimization problem (aka Linear Program)
has one objective function and all f_i, g_i are linear $\vec{a}^T\vec{x} = a_1x_1 + a_2x_2 + \cdots + a_nx_n$
and variables are real valued.

example A firm produces 2 different goods (G1, G2) using 3 different raw materials, R1 & R2 & R3.
G1 per unit amount needs 4 units of R1, 3 units of R2 & 4 units of R3.
G2 ——— " ——— 1 unit of R1, 5 units of R2 & 3 units of R3.
We have 23 units of R1, 26 of R2 and 21 of R3 available.

G1 sells for \$40/unit amount and G2 sells for \$30/unit amount.

How much amount of G1 & G2 should we produce?

Decision variables x_i = amount of good Gi, i=1,2, to be produced

Objective function

Constraints

Decision variables x_i = amount of good G_i, $i=1,2$, to be produced

Objective function maximize revenue

Constraints

production requirements

for R1

for R2

for R3

Variables

Decision variables x_i = amount of good Gi , $i=1,2$, to be produced

Objective function maximize revenue max $40x_1 + 30x_2$

Constraints subject to

production requirements

for R1 $4x_1 + 1x_2 \leq 23$

for R2 $3x_1 + 5x_2 \leq 26$

for R3 $4x_1 + 3x_2 \leq 21$

Variables $x_1 \geq 0$

$x_2 \geq 0$

A linear Optimization problem (aka Linear Program) has one objective function and all f_i, g_i are linear and variables are real valued.

example A firm produces 2 different goods (G1, G2) using 3 different raw materials, R1 & R2 & R3.
G1 per unit amount needs 4 units of R1, 3 units of R2 & 4 units of R3.
G2 ——— " ——— 1 unit of R1, 5 units of R2 & 3 units of R3.
We have 23 units of R1, 26 of R2 and 21 of R3 available.

G1 sells for \$40/unit amount and G2 sells for \$30/unit amount.

How much amount of G1 & G2 should we produce?

A linear Optimization problem (aka Linear Program) has one objective function and all f_i, g_i are linear and variables are real valued.

example A firm produces 2 different goods (G1, G2) using 3 different raw materials, R1 & R2 & R3.
G1 per unit amount needs 4 units of R1, 3 units of R2 & 4 units of R3.
G2 ——— " ——— 1 unit of R1, 5 units of R2 & 3 units of R3.
We have 23 units of R1, 26 of R2 and 21 of R3 available to buy.
Each unit amount of R1 cost \$4, of R2 costs \$3, of R3 costs \$3.
We have an operating budget of \$100.
G1 sells for \$40/unit amount and G2 sells for \$30/unit amount.

How much amount of G1 & G2 should we produce?

Decision variables x_i = amount of good Gi, $i=1,2$, to be produced

Objective function maximize revenue $\max\ 40x_1 + 30x_2$

Constraints subject to

production requirements

for R1 $4x_1 + 1x_2 \leq 23$

for R2 $3x_1 + 5x_2 \leq 26$

for R3 $4x_1 + 3x_2 \leq 21$

Variables

$x_1 \geq 0$

$x_2 \geq 0$

Decision variables x_i = amount of good G_i, $i=1,2$ to produce.
y_i = amount of raw mat. R_i, $i=1,2,3$ to buy.

Objective function maximize revenue max $40x_1 + 30x_2$

Constraints subject to

production requirements

for R1 $4x_1 + 1x_2 \leq 23$

for R2 $3x_1 + 5x_2 \leq 26$

for R3 $4x_1 + 3x_2 \leq 21$

Budget for buying R1, R2 & R3

Market availability of R1, R2 & R3

Variables $x_1 \geq 0$

$x_2 \geq 0$

Decision variables x_i = amount of good G_i, $i=1,2$ to produce.
y_i = amount of raw mat. R_i, $i=1,2,3$ to buy.

Objective function maximize revenue $\max\ 40x_1 + 30x_2$

Constraints

subject to

production requirements

for R1 $4x_1 + 1x_2 \leq 23$

for R2 $3x_1 + 5x_2 \leq 26$

for R3 $4x_1 + 3x_2 \leq 21$

Budget for buying R1, R2 & R3 $4y_1 + 3y_2 + 3y_3 \leq 100$

Market availability of R1, R2, & R3 $y_1 \leq 23,\ y_2 \leq 26,\ y_3 \leq 21$

Variables $x_1 \geq 0$

$x_2 \geq 0$

Decision variables x_i = amount of good G_i, $i=1,2$ to produce.
y_i = amount of raw mat. R_i, $i=1,2,3$ to buy.

Objective function maximize revenue $\max\ 40x_1 + 30x_2$

Constraints subject to

production requirements

for R1 $4x_1 + 1x_2 \leq y_1$

for R2 $3x_1 + 5x_2 \leq y_2$

for R3 $4x_1 + 3x_2 \leq y_3$

Budget for buying R1, R2 & R3 $4y_1 + 3y_2 + 3y_3 \leq 100$

Market availability of R1, R2, R3 $y_1 \leq 23,\ y_2 \leq 26,\ y_3 \leq 21$

Variables $x_1, x_2, y_1, y_2, y_3 \geq 0$

Decision variables x_i = amount of good G_i, $i=1,2$ to produce.
y_i = amount of raw mat. R_i, $i=1,2,3$ to buy.

Objective function maximize profit
= revenue − expenditure

Constraints subject to

production requirements

for R1 $4x_1 + 1x_2 \leq y_1$

for R2 $3x_1 + 5x_2 \leq y_2$

for R3 $4x_1 + 3x_2 \leq y_3$

Budget for buying R1, R2 & R3 $4y_1 + 3y_2 + 3y_3 \leq 100$

Market availability of R1, R2, R3 $y_1 \leq 23$, $y_2 \leq 26$, $y_3 \leq 21$

Variables $x_1, x_2, y_1, y_2, y_3 \geq 0$

Decision variables

x_i = amount of good G_i, $i=1,2$ to produce.

y_i = amount of raw mat. R_i, $i=1,2,3$ to buy.

Objective function

maximize profit = revenue − expenditure

$$\max\ \left[(40x_1+30x_2)-(4y_1+3y_2+3y_3)\right]$$

Constraints

subject to

production requirements

for R1: $4x_1+1x_2 \le y_1$

for R2: $3x_1+5x_2 \le y_2$

for R3: $4x_1+3x_2 \le y_3$

Budget for buying R1, R2 & R3: $4y_1+3y_2+3y_3 \le 100$

Market availability of R1, R2 & R3: $y_1 \le 23,\ y_2 \le 26,\ y_3 \le 21$

Variables: $x_1, x_2, y_1, y_2, y_3 \ge 0$

Pay attention to variables: $x_1, x_2, y_1, y_2, y_3 \geq 0$

underlying assumption?

what if our goods were G1 = tables, G2 = chairs

& raw materials R1 = wood, R2 = iron, R3 = cloth.

Pay attention to variables: $x_1, x_2, y_1, y_2, y_3 \geq 0$

underlying assumption?

what if our goods were G1 = tables, G2 = chairs

& raw materials R1 = wood, R2 = iron, R3 = cloth.

Then we need, $x_1, x_2 \in \mathbb{Z}$

$$x_1, x_2 \geq 0$$

$$y_1, y_2, y_3 \geq 0$$

what if, in addition, we have signed a contract to deliver 2 tables and 8 chairs to a customer

Pay attention to variables: $x_1, x_2, y_1, y_2, y_3 \geq 0$

underlying assumption?

what if our goods were $G1$ = tables, $G2$ = chairs

& raw materials $R1$ = wood, $R2$ = iron, $R3$ = cloth.

Then we need, $x_1, x_2 \in \mathbb{Z}$

$$x_1, x_2 \geq 0$$

$$y_1, y_2, y_3 \geq 0$$

what if, in addition, we have signed a contract to deliver 2 tables and 8 chairs to a customer

Then we need $\left.\begin{array}{l} x_1 \geq 2 \\ x_2 \geq 8 \end{array}\right\}$ part of main constraints.

$$x_1, x_2 \in \mathbb{Z}$$

$$y_1, y_2, y_3 \geq 0$$

Integer Optimization Problem

Some (or all) variables are restricted to take only integer values.

Mixed Integer Linear Program is a linear optimization problem with some variables restricted to be integer valued.

0-1 or binary Linear Program has integer variables restricted to take only 0 or 1 as values.

Binary variables are a very powerful tool (as they model "Yes" / "No" decisions).

Pay attention to Objective function:

we had "max revenue" as our initial obj. function

What if we wanted "max revenue" and "min costs" both as our objective functions?

Or,

$$\left.\begin{aligned} &\max \; F_1(\vec{x}) \\ &\max \; F_2(\vec{x}) \\ &\max \; F_3(\vec{x}) \\ &\quad\vdots \\ &\max \; F_k(\vec{x}) \\ &\text{s.t. } G_i \geq b_i, \ldots \end{aligned}\right\}$$

Such an optimization problem is called an <u>multiobjective optimization problem.</u>

→ How to solve such problems?

Modeling Multiobjective Optimization problems

① Combine the objective functions into a single objective function.

Modeling Multiobjective Optimization problems

① Combine the objective functions into a single objective function.

↳ e.g. $f(\vec{x}) = \omega_1 f_1(\vec{x}) + \omega_2 f_2(\vec{x}) + \dots + \omega_k f_k(\vec{x})$

is the new single objective function using a linear combination with weights $\omega_1, \omega_2, \dots, \omega_k$

Recall Revenue − Expenditure

↳ Treat each objective function as a player in a k-player game . . . . .

Modeling Multiobjective Optimization problems

① Combine the objective functions into a single objective function.

↳ e.g. $f(\vec{x}) = \omega_1 f_1(\vec{x}) + \omega_2 f_2(\vec{x}) + \dots + \omega_k f_k(\vec{x})$

is the new single objective function using a linear combination with weights $\omega_1, \omega_2, \dots, \omega_k$

Caution!

<u>Recall</u> Revenue − Expenditure

↳ Treat each objective function as a player in a k-player game . . . . .

All objective functions must be converted into the same units of measurement } How?

Modeling Multiobjective Optimization problems

① Combine the objective functions into a single objective function.

e.g. Transportation Network Design

Objectives are:
- min total Travel time delays (hours)
- min fuel consumption (gallons)
- min loss of life due to accidents (#persons)
- min Total cost of chosen projects ($)
- ⋮

Constraints are:
- Network design requirements
- Traffic flow requirements
- ⋮

Modeling Multiobjective Optimization problems

① Combine the objective functions into a single objective function.

e.g. Transportation Network Design

Objectives are:

- min total Travel time delays (hours) — economic cost
- min fuel consumption (gallons) — cost of fuel
- min loss of life due to accidents (#persons) — actuary table
- min Total cost of chosen projects ($) — $

⋮

$

Constraints are:

Network design requirements

Traffic flow requirements

⋮

Modeling Multiobjective Optimization problems

② Put an order on objective function valued vectors

How to compare $(f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x}))$

with $(f_1(\vec{y}), f_2(\vec{y}), \dots, f_k(\vec{y}))$?

Modeling Multiobjective Optimization problems

(2) Put an order on objective function valued vectors

How to compare $(f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x}))$

with $(f_1(\vec{y}), f_2(\vec{y}), \ldots, f_k(\vec{y}))$?

↳ Lexicographic order on $(f_1, \ldots, f_k)$ so that

f_1 is more important than f_2 which is more important than $f_3, \ldots$

e.g. $(3, 0, 1, 0, 2) \underset{L}{\geqslant} (2, 25, 93, 48, 16)$

$(1, 5, 0, 0, 0) \underset{L}{\geqslant} (1, 4, 97, 68, 14)$

Modeling Multiobjective Optimization problems

② Put an order on objective function valued vectors

How to compare $(f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x}))$

with $(f_1(\vec{y}), f_2(\vec{y}), \dots, f_k(\vec{y}))$?

↳ Lexicographic order on $(f_1, \dots, f_k)$ so that f_1 is more important than f_2 which is more important than f_3, …

e.g. $(3, 0, 1, 0, 2) \geq (2, 25, 93, 48, 16)$

$(1, 5, 0, 0, 0) \geq (1, 4, 97, 68, 14)$

↳ minimize l_p-distance to "ideal point"

↳ $(OPT_{f_1}, OPT_{f_2}, \dots, OPT_{f_k})$

where OPT_{f_i} is the optimal solution of the problem with only f_i as the single obj. function.

Modeling Multiobjective Optimization problems

③ Keep only one objective function as the single obj. ftn. and convert the rest into appropriate constraints

$$
\begin{aligned}
\max\ & f_1(\vec{x}) \\
\text{s.t.}\ & f_2(\vec{x}) \le B_2 \\
& f_3(\vec{x}) \le B_3 \\
& \vdots \\
& f_k(\vec{x}) \le B_k \\
& G_i(\vec{x}) \le b_i, \ldots
\end{aligned}
$$

e.g. For a fire station location problem,

min (construction costs)
(response time)
(loss of lives)

$\downarrow$

min (loss of lives)
s.t. (response time) ≤ mandate
(construction costs) ≤ budget

Decision making models using binary variables

"Choosing between two alternatives" $\longleftrightarrow$ set $x = 0$ or 1

Alternative #1   Alternative #2

0-1 Knapsack Model for Budgeting

Available: n items each with cost c_j & value v_j, $j = 1, \dots, n$.

Budget: B, total amount available to buy the items

Decision making models using binary variables

"Choosing between two alternatives" $\longleftrightarrow$ set $x = 0$ or 1

each alternative

0-1 Knapsack Model for Budgeting

Available: n items each with cost c_j & value v_j, $j = 1, \dots, n$.

Budget: B, total amount available to buy the items

Variables

Objective

Constraints

Decision making models using binary variables

"Choosing between two alternatives" $\longleftrightarrow$ set $x = 0$ or 1

each alternative

0-1 Knapsack Model for Budgeting

Available: n items each with cost c_j & value v_j, $j = 1, \ldots, n$.

Budget: B, total amount available to buy the items

Variables $x_1, x_2, \ldots, x_n$ $\quad x_i = \begin{cases} 1 & \text{if we pick item } i \\ 0 & \text{if we don't pick item } i \end{cases}$

Objective maximize total value of the picked items

Constraints Don't exceed the budget

Decision making models using binary variables

"Choosing between two alternatives" $\longleftrightarrow$ set $x = 0$ or 1

each alternative

0-1 Knapsack Model for Budgeting

Available: n items each with cost c_j & value v_j, $j = 1, \dots, n$.

Budget: B, total amount available to buy the items

Variables $x_1, x_2, \dots, x_n$ $\quad x_i = \begin{cases} 1 & \text{if we pick item } i \\ 0 & \text{if we don't pick item } i \end{cases}$

Objective maximize total value of the picked items $\longrightarrow$

Constraints Don't exceed the budget $\longrightarrow$

$$\boxed{\begin{aligned} \max \ & \sum_{i=1}^{n} v_i x_i \\ \text{s.t.} \ & \sum_{i=1}^{n} c_i x_i \le B \\ & x_i \in \{0,1\} \ \forall i \end{aligned}}$$

Out of 5 yes/no decisions, we want exactly one to be yes, how can we ensure this ?

$$\text{Let } x_i = \begin{cases} 1 & \text{if decision } i \text{ is "yes"} \\ 0 & \text{if decision } i \text{ is "no"} \end{cases} \quad \text{for } i = 1, 2, 3, 4, 5$$

Out of 5 yes/no decisions, we want exactly one to be yes,
how can we ensure this?

Let $x_i = \begin{cases} 1 & \text{if decision } i \text{ is "yes"} \\ 0 & \text{if decision } i \text{ is "no"} \end{cases}$ for $i=1,2,3,4,5$

$$\boxed{x_1 + x_2 + x_3 + x_4 + x_5 = 1 \quad \text{for } x_1, x_2, \dots, x_5 \in \{0,1\}.}$$

What if atleast one should be "yes"?

What if atmost one should be "yes"?

How can we restrict a variable x to only take either 0 or 9 as its value?

How can we restrict a variable x to only take either 0 or 9 as its value?

$$x = 9y \quad \text{with} \quad y \in \{0, 1\}$$

How can we restrict a variable x to either be 0 or take a value in the interval $[3, 4]$?

How can we restrict a variable x to only take either 0 or 9 as its value?

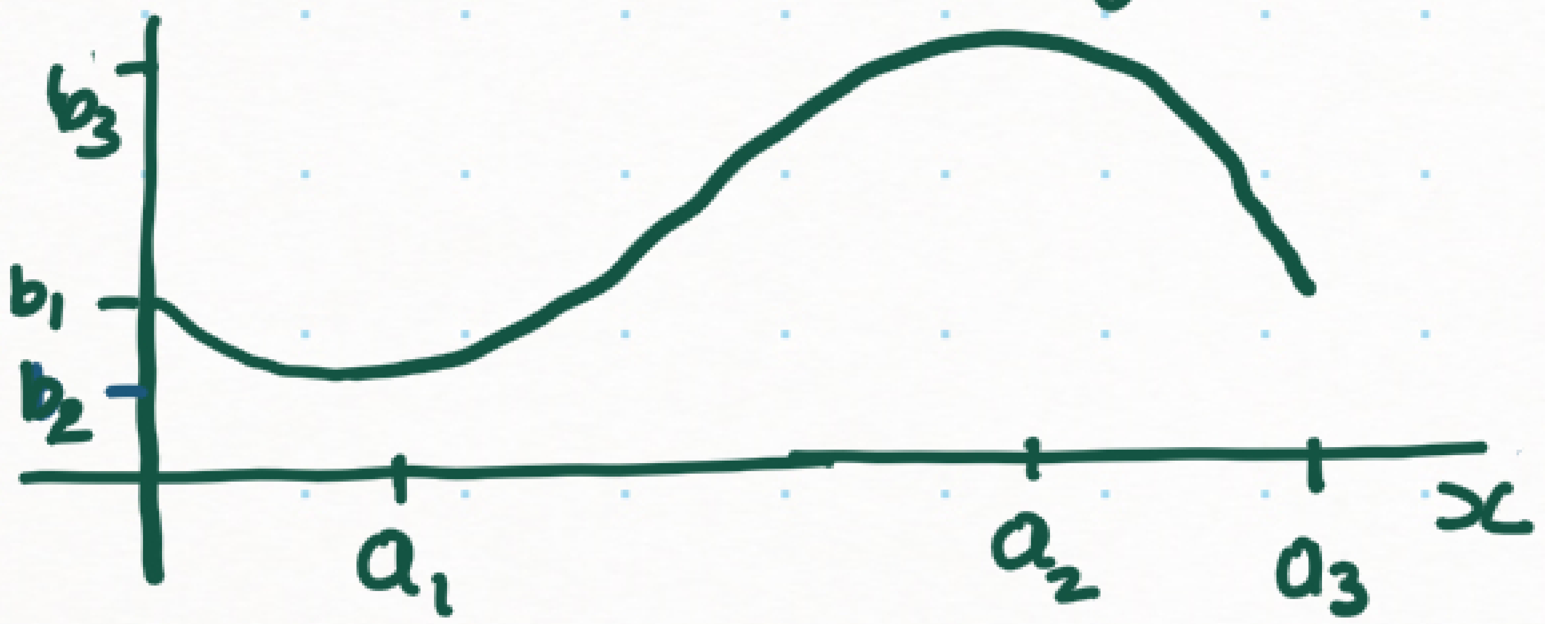
$$x = 9y \quad \text{with} \quad y \in \{0, 1\}$$

How can we restrict a variable x to either be 0 or take a value in the interval $[3, 4]$?

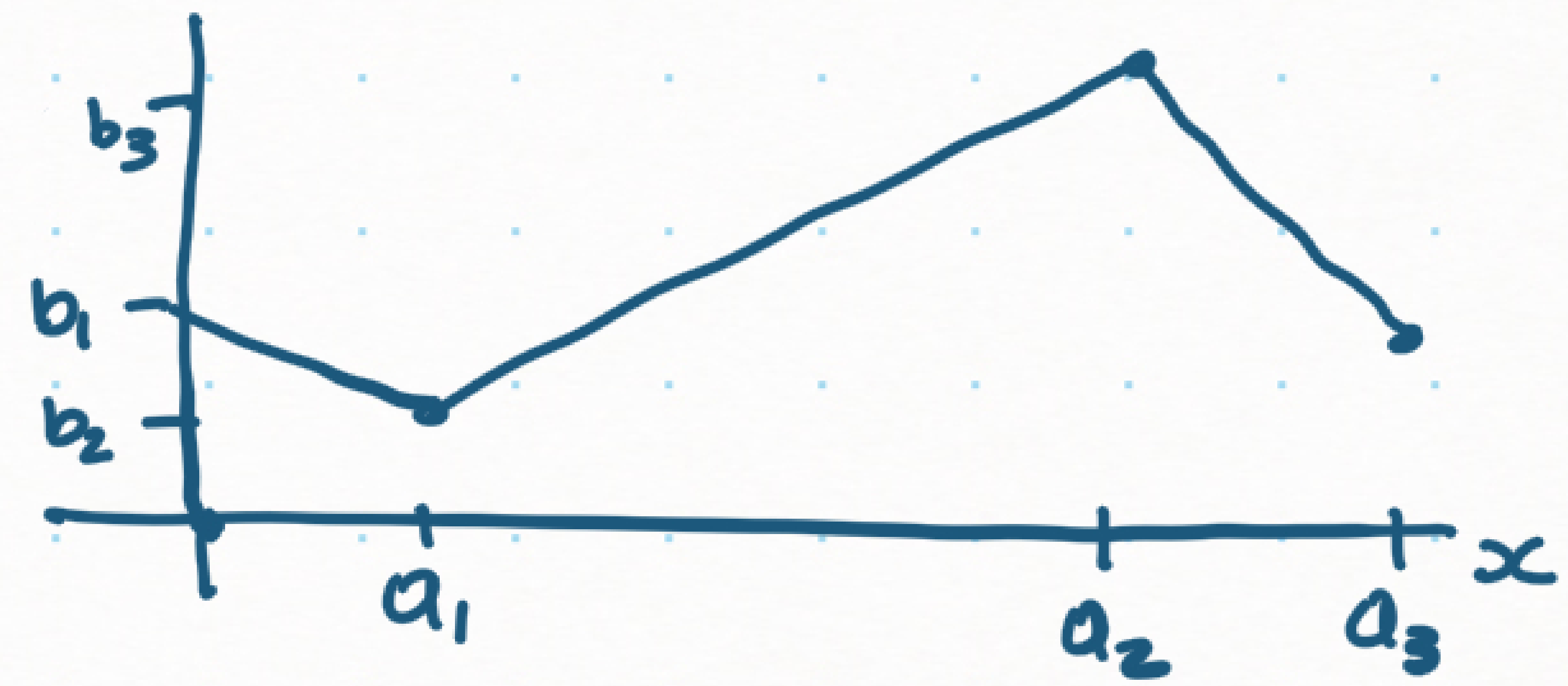
$$x \leq 4y$$
$$x \geq 3y \quad \text{with} \quad y \in \{0, 1\}$$

Read example 3 on page 247 of the textbook

We want to approximate a non-linear function



by a piecewise linear function



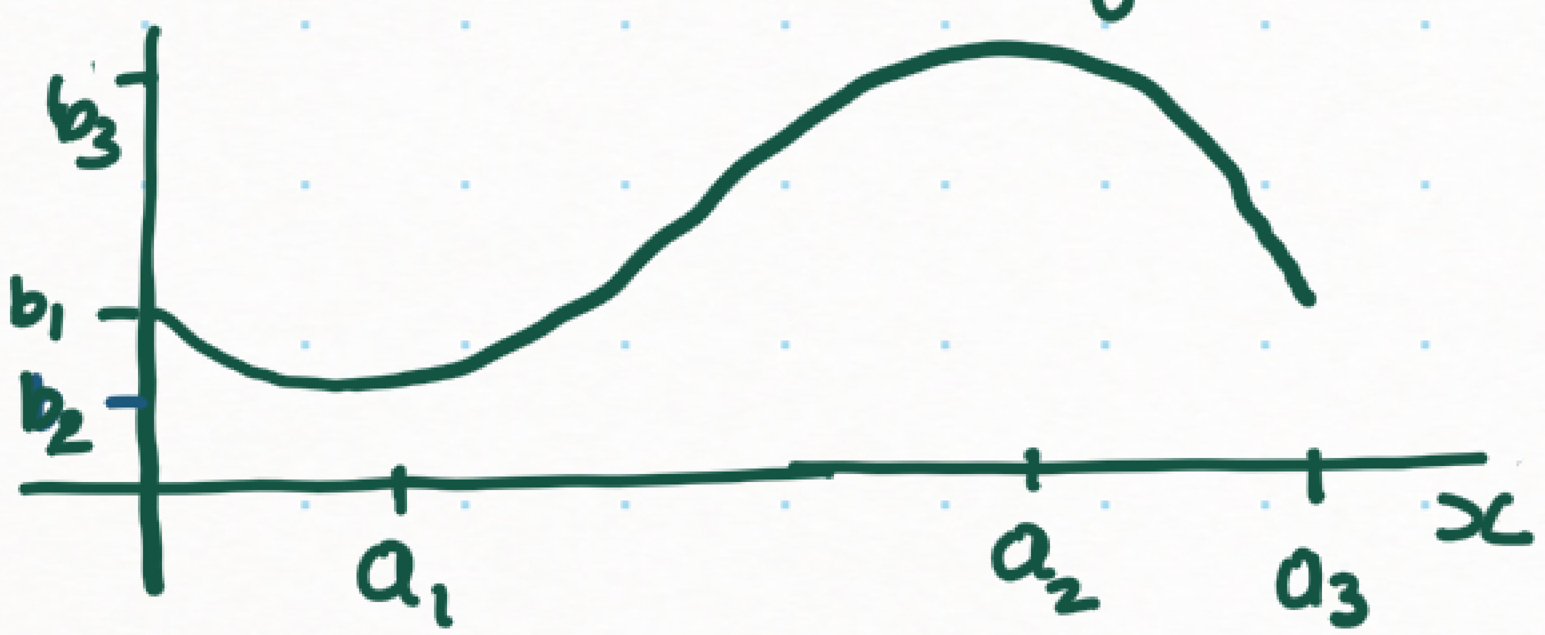
Can we express this piecewise linear function as a single (lin.) function?

$$g(x) = \begin{cases} b_1 + k_1 x & \text{if } 0 \le x \le a_1 \\ b_2 + k_2 (x - a_1) & \text{if } a_1 \le x \le a_2 \\ b_3 + k_3 (x - a_2) & \text{if } a_2 \le x \le a_3 \end{cases}$$

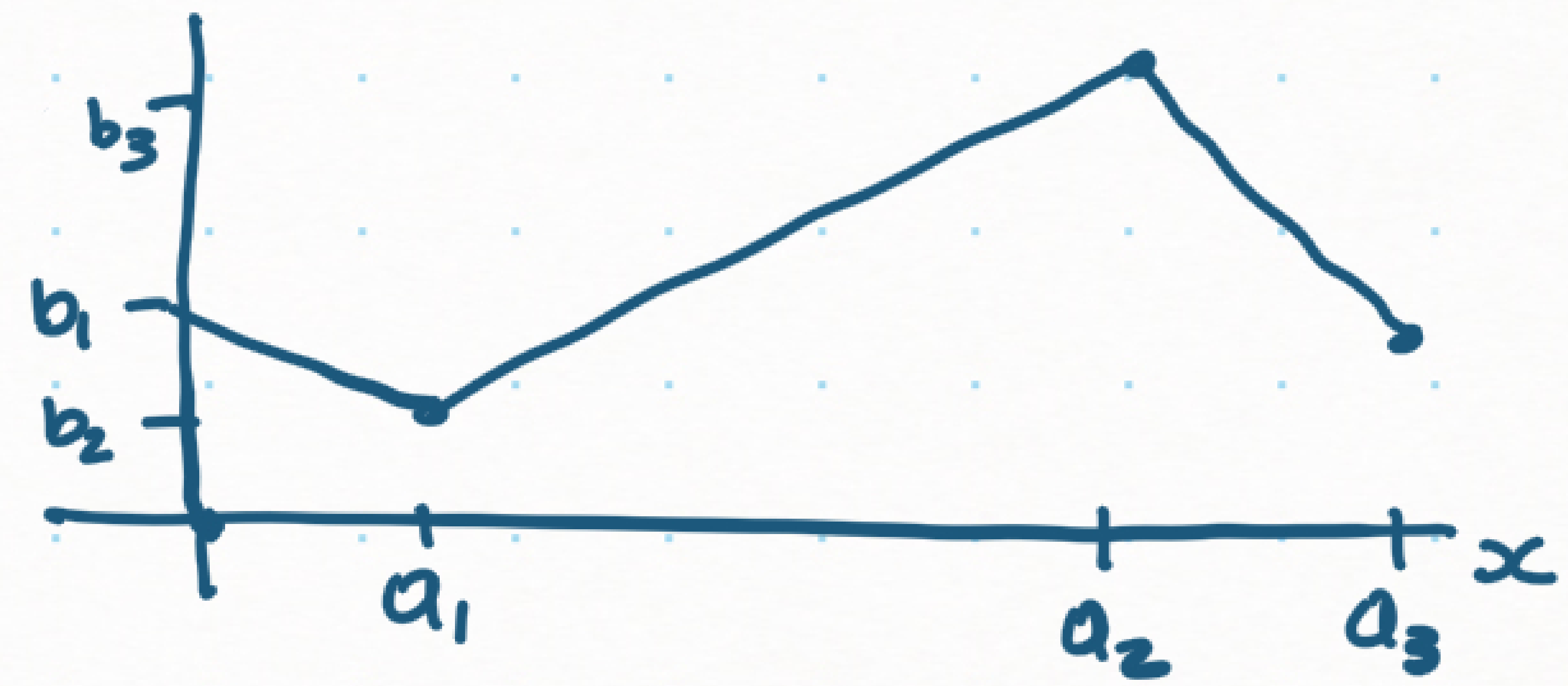
can be expressed as

Read example 3 on page 247 of the textbook

We want to approximate a non-linear function



by a piecewise linear function



Can we express this piecewise linear function as a single (linear) function ?

$$g(x) = \begin{cases} b_1 + k_1 x & \text{if } 0 \le x \le a_1 \\ b_2 + k_2(x - a_1) & \text{if } a_1 \le x \le a_2 \\ b_3 + k_3(x - a_2) & \text{if } a_2 \le x \le a_3 \end{cases}$$

can be expressed as

$$g(x) = y_1(b_1 + k_1 x_1) + y_2(b_2 + k_2 x_2) + y_3(b_2 + k_3 x_3)$$

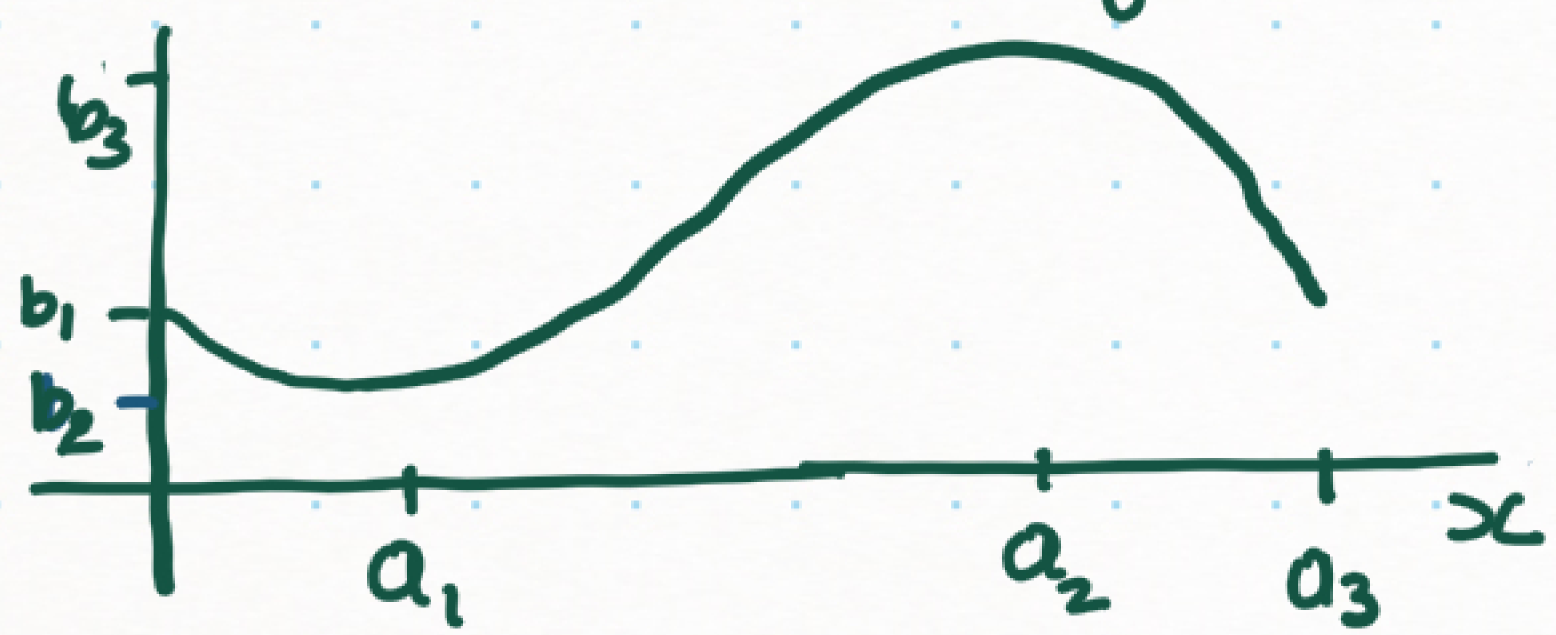
← What is the value of x_i when $y_i = 0$?
So can we simplify $x_i y_i$?

where ? → $y_1 + y_2 + y_3 = 1$, $y_1, y_2, y_3 \in \{0, 1\}$

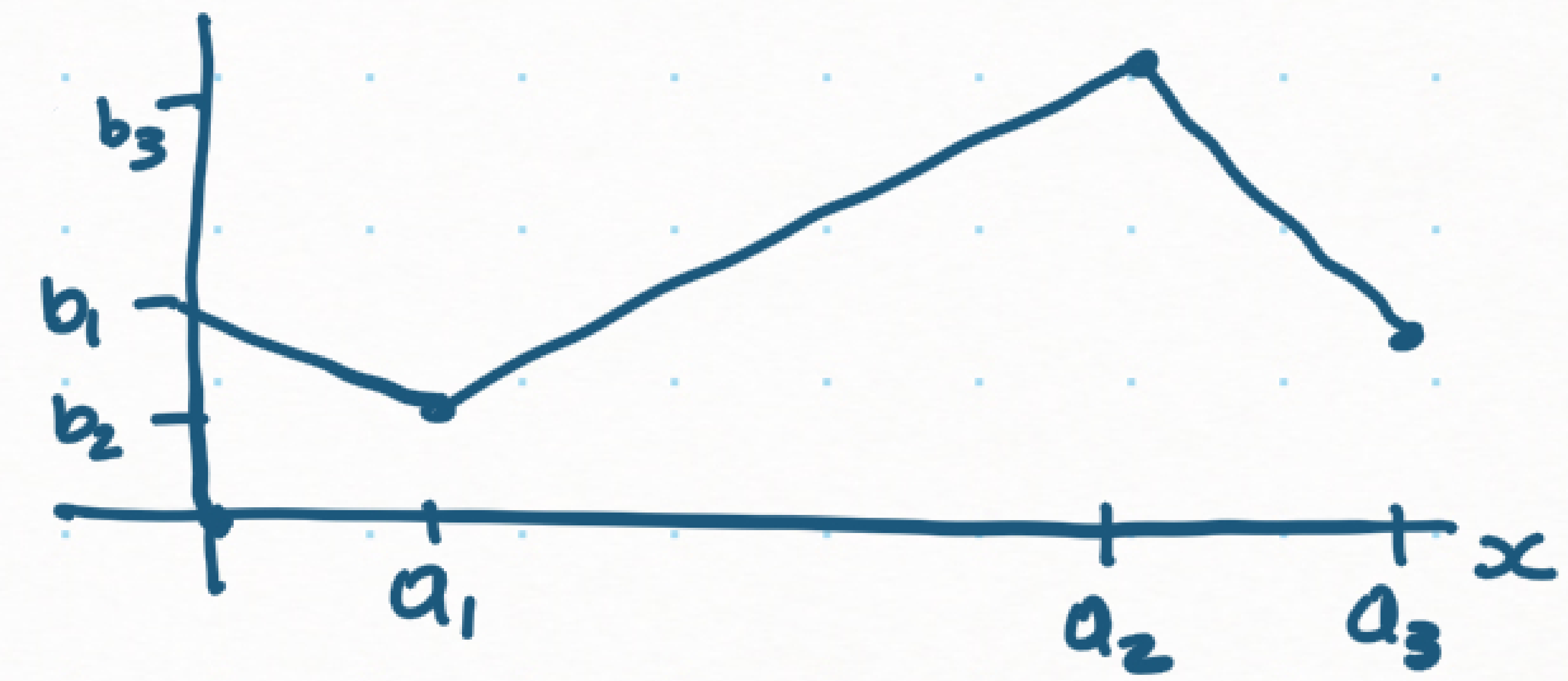
? → $0 \le x_1 \le a_1 y_1$, $0 \le x_2 \le (a_2 - a_1) y_2$, $0 \le x_3 \le (a_3 - a_2) y_3$

Read example 3 on page 247 of the textbook

We want to approximate a non-linear function



by a piecewise linear function



Can we express this piecewise linear function as a single (lin.) function?

$$g(x) = \begin{cases} b_1 + k_1 x & \text{if } 0 \le x \le a_1 \\ b_2 + k_2(x - a_1) & \text{if } a_1 \le x \le a_2 \\ b_3 + k_3(x - a_2) & \text{if } a_2 \le x \le a_3 \end{cases}$$

can be expressed as

$$g(x) = k_1 x_1 + k_2 x_2 + k_3 x_3 + b_1 y_1 + b_2 y_2 + b_3 y_3$$

where $y_1 + y_2 + y_3 = 1$, $y_1, y_2, y_3 \in \{0, 1\}$

$0 \le x_1 \le a_1 y_1$, $0 \le x_2 \le (a_2 - a_1) y_2$, $0 \le x_3 \le (a_3 - a_2) y_3$

Decisions are often carried out in a particular order.

i.e., in order to make "Decision A" we must already have made "Decision B"

Decisions are often carried out in a particular order,
i.e., in order to make "Decision A" we must already have made "Decision B"

$$\text{Decision A} \longleftrightarrow x = \begin{cases} 0 & \text{if decision A is not made} \\ 1 & \text{if decision A is made} \end{cases}$$

$$\text{Decision B} \longleftrightarrow y = \begin{cases} 0 & \text{if decision B is not made} \\ 1 & \text{if decision B is made} \end{cases}$$

Decisions are often carried out in a particular order,

i.e., in order to make "Decision A" we must already have made "Decision B"

Decision A $\longleftrightarrow$ $x = \begin{cases} 0 & \text{if decision A is not made} \\ 1 & \text{if decision A is made} \end{cases}$

Decision B $\longleftrightarrow$ $y = \begin{cases} 0 & \text{if decision B is not made} \\ 1 & \text{if decision B is made} \end{cases}$

If $y=0$ then x must also equal 0

Decisions are often carried out in a particular order,

i.e., in order to make "Decision A" we must already have made "Decision B"

Decision A $\longleftrightarrow$ $x = \begin{cases} 0 & \text{if decision A is not made} \\ 1 & \text{if decision A is made} \end{cases}$

Decision B $\longleftrightarrow$ $y = \begin{cases} 0 & \text{if decision B is not made} \\ 1 & \text{if decision B is made} \end{cases}$

If $y=0$ then x must also equal 0

$$\boxed{x \le y \quad \text{where } x, y \in \{0, 1\}.}$$

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
$$a_1x_1 + a_2x_2 + \cdots + a_nx_n \leq b$$
$$a_1x_2 + a_2x_2 + \cdots + a_nx_n \geq b$$

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

is equivalent to

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

Recall CAC

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

is equivalent to

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

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

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

$\longrightarrow$

$$\min \ \vec{c}^T\vec{x}$$
$$\text{s.t. } A\vec{x} \leq \vec{b}$$
$$\vec{x} \geq 0$$

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 <u>feasible solution</u>.

<u>Feasible set</u> is the collection of all feasible solutions.

Geometrically it looks like a polyhedron.

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

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

$2x_1 + x_2 \leq 3$

$x_1, x_2 \geqslant 0$

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 $x_1 = 0$ & $x_2 = 0$.

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

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

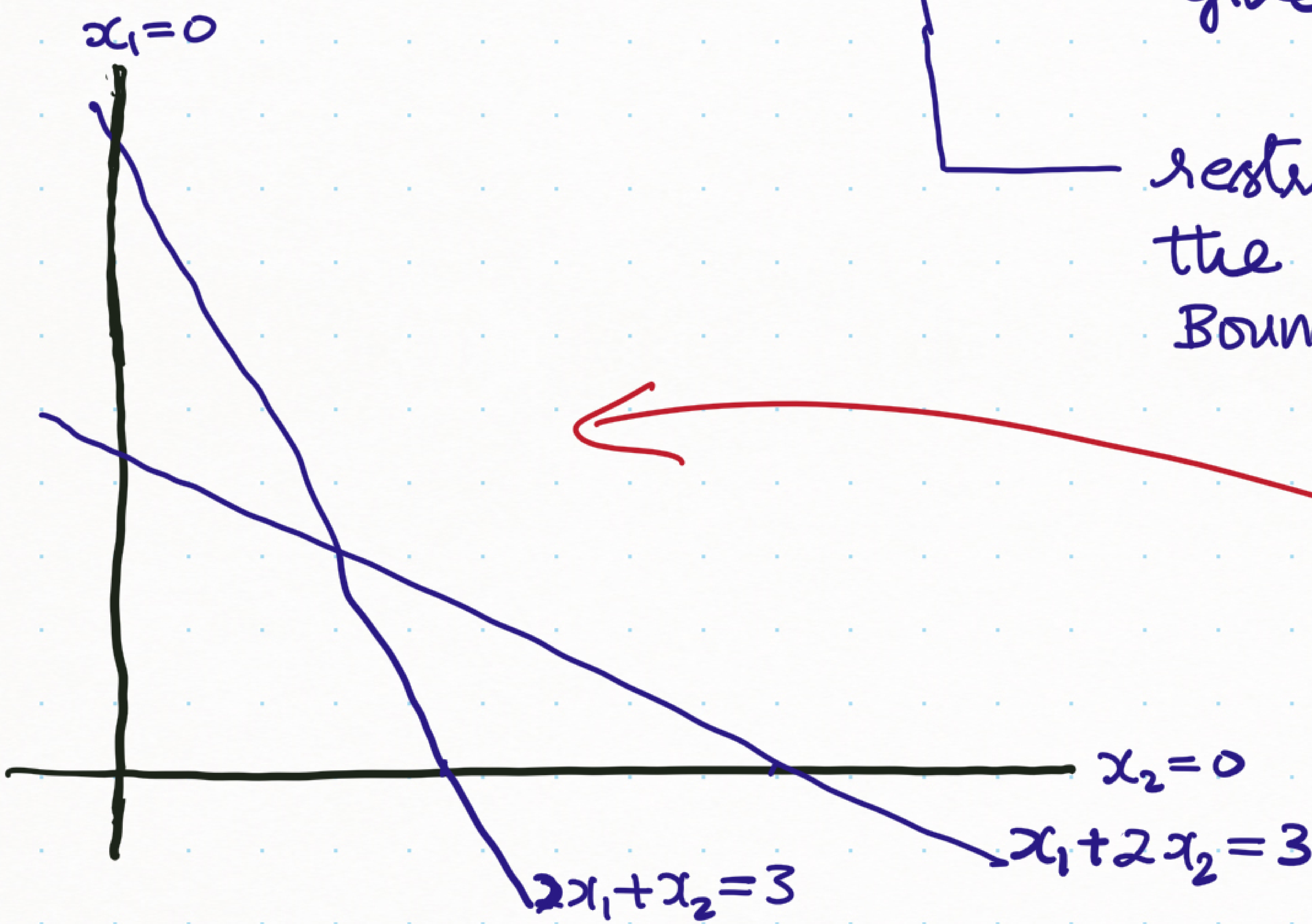
$2x_1 + x_2 \leq 3$

$x_1, x_2 \geq 0$

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. $x_1 + 2x_2 \leq 3$

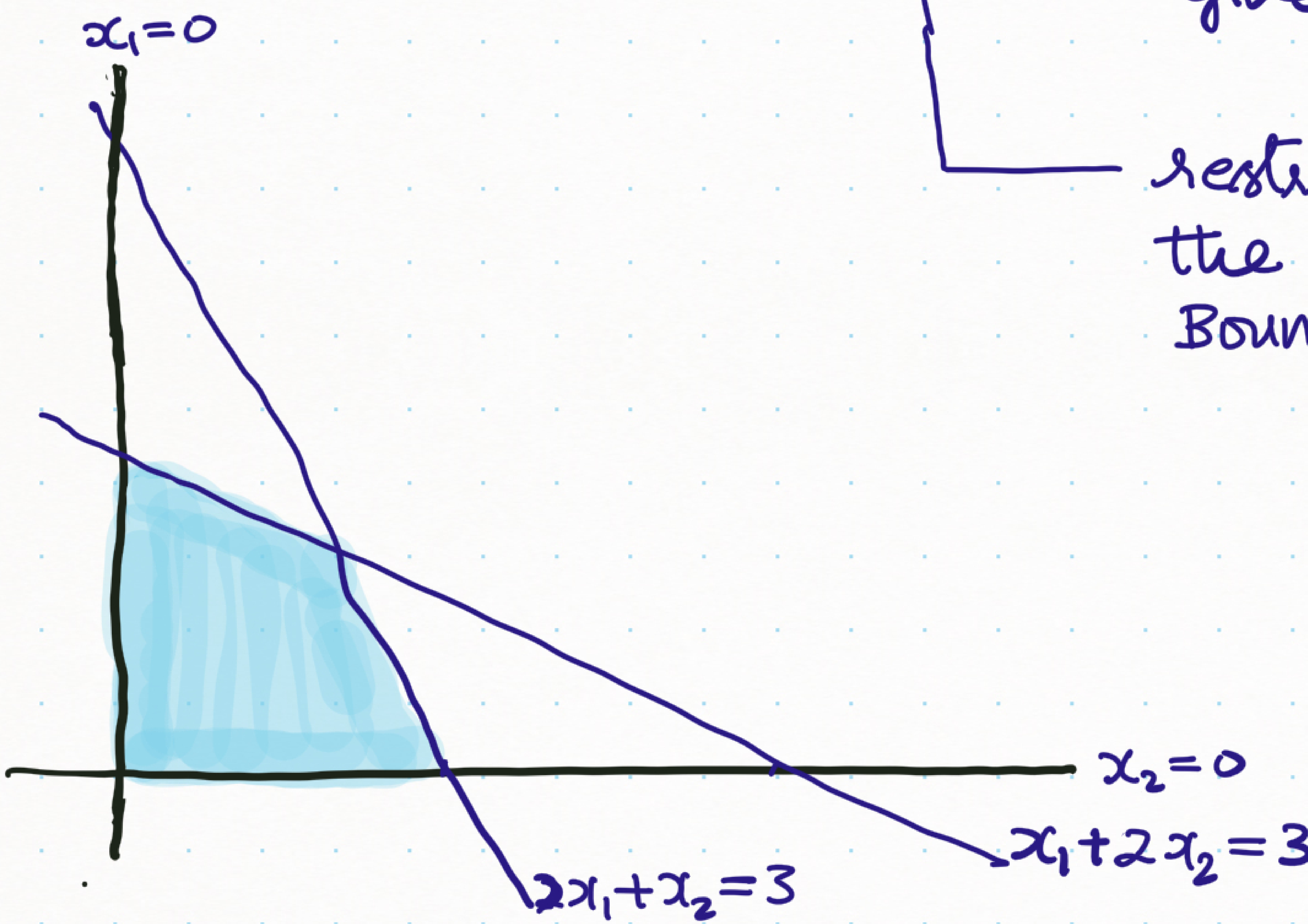
$2x_1 + x_2 \leq 3$

$x_1, x_2 \geqslant 0$

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 an optimal solution?

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

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

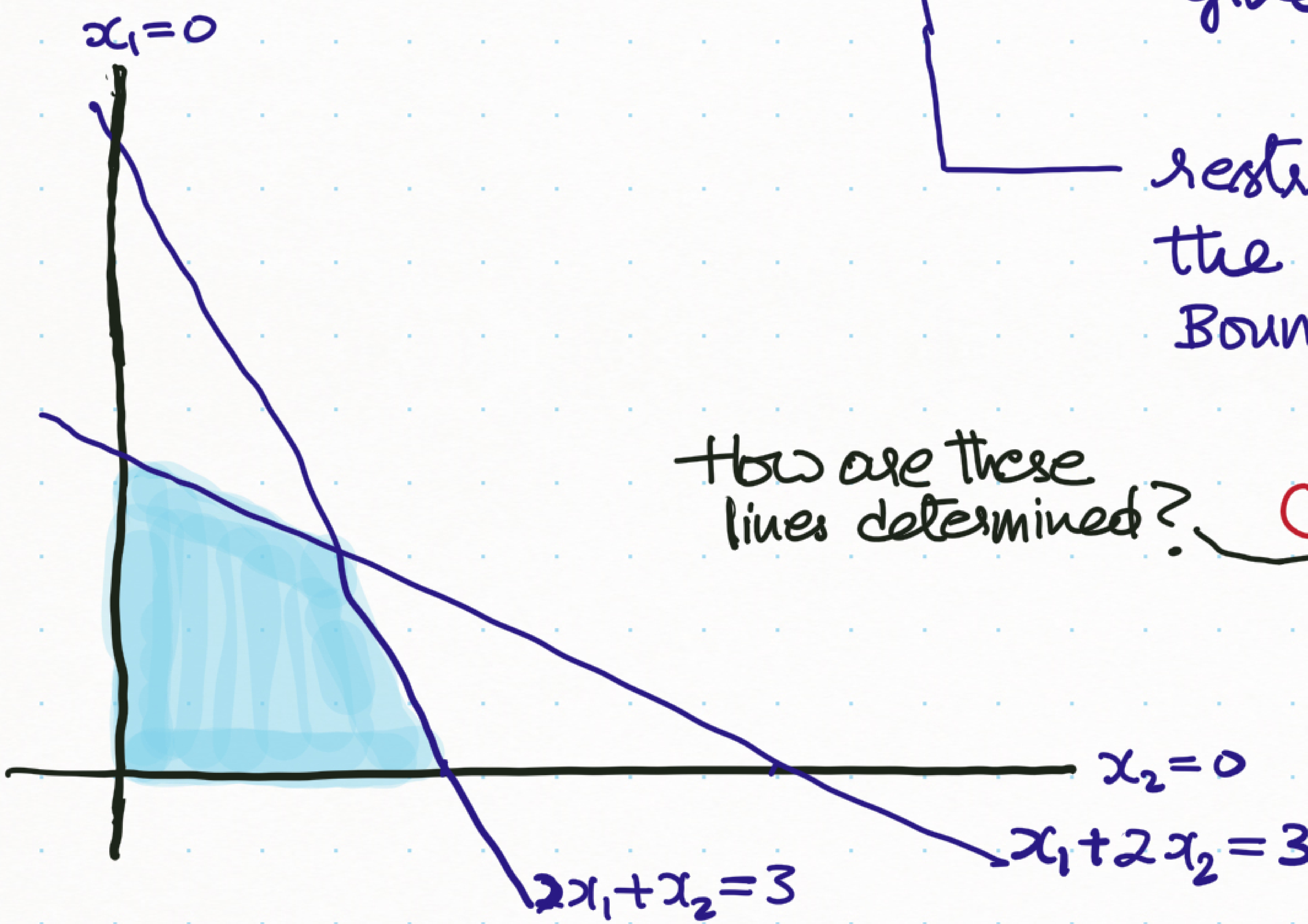
$2x_1 + x_2 \leq 3$

$x_1, x_2 \geqslant 0$

← 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. $x_1 + 2x_2 \leq 3$

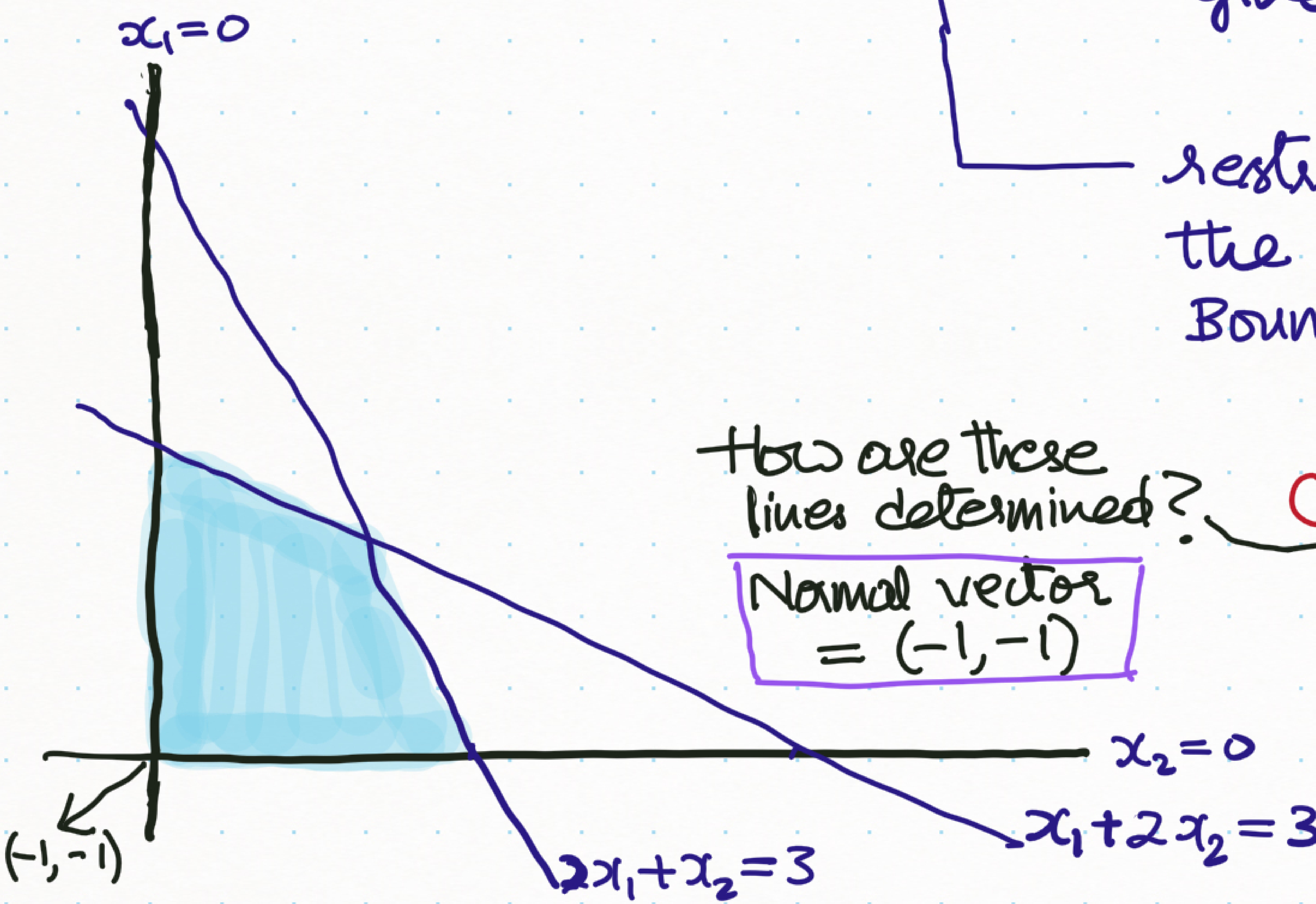
$2x_1 + x_2 \leq 3$

$x_1, x_2 \geq 0$

← 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. $x_1 + 2x_2 \leq 3$

$2x_1 + x_2 \leq 3$

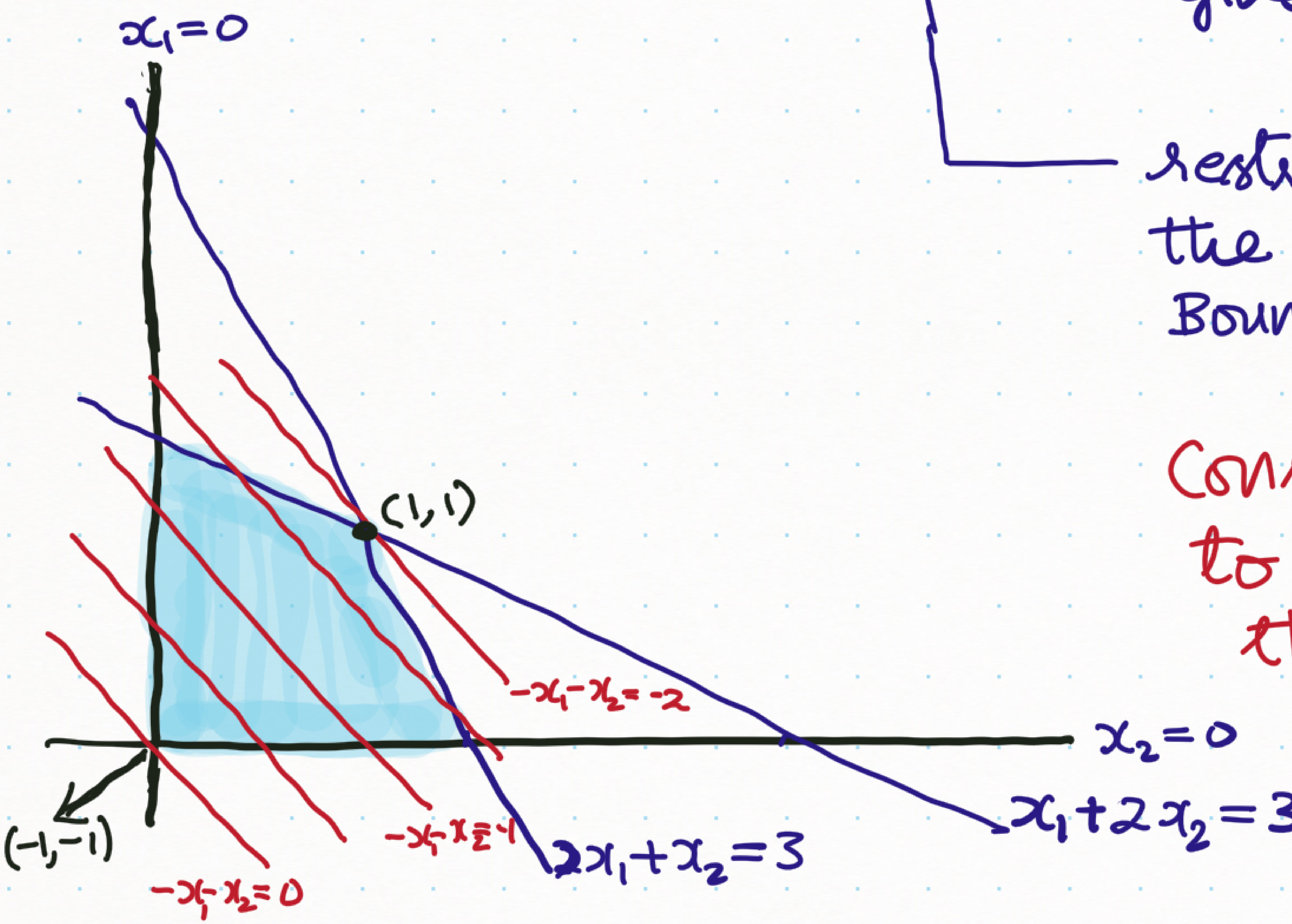
$x_1, x_2 \geq 0$

← 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. $x_1 + 2x_2 \leq 3$

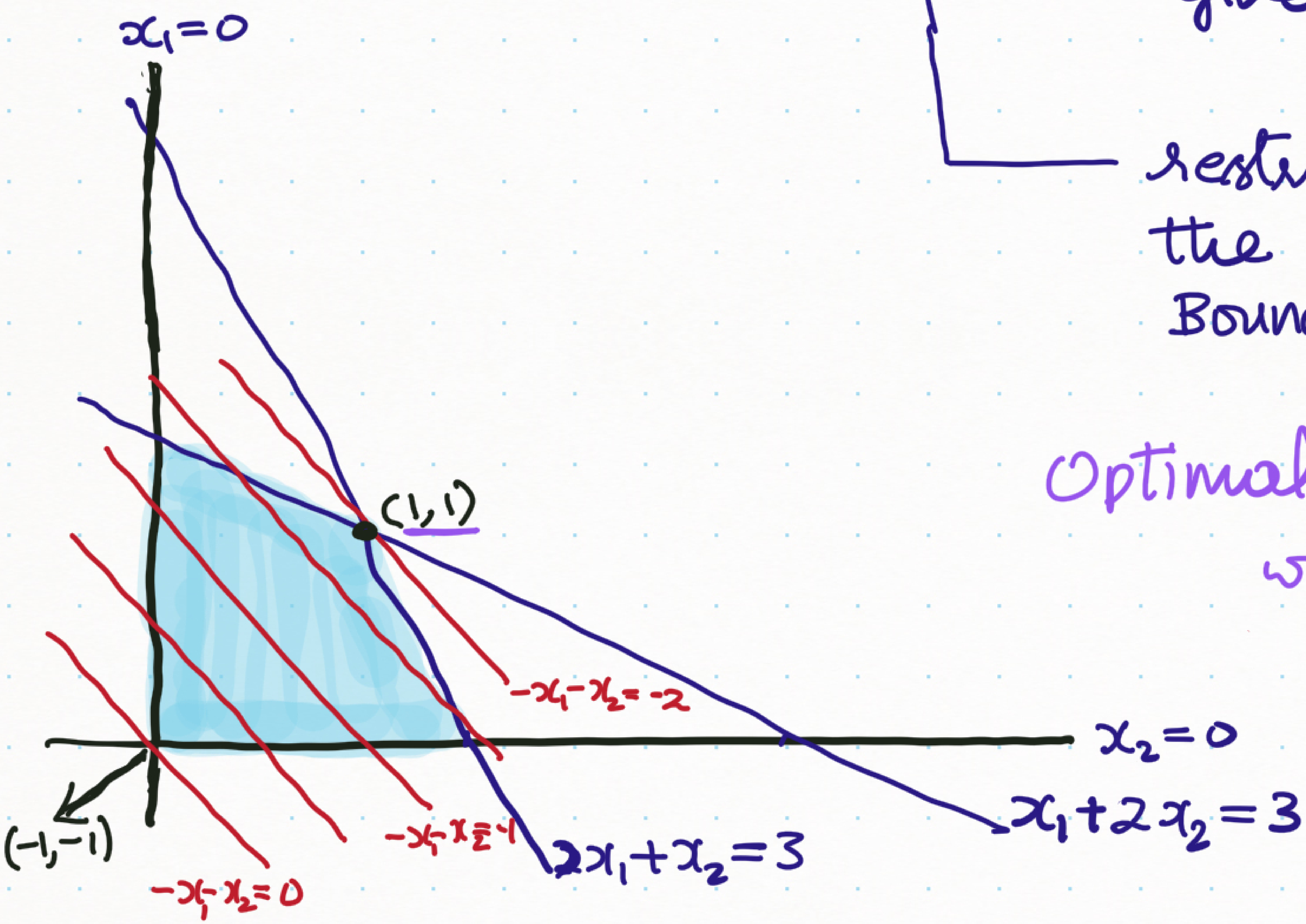
$2x_1 + x_2 \leq 3$

$x_1, x_2 \geq 0$

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. $\min \ -x_1 - x_2$

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

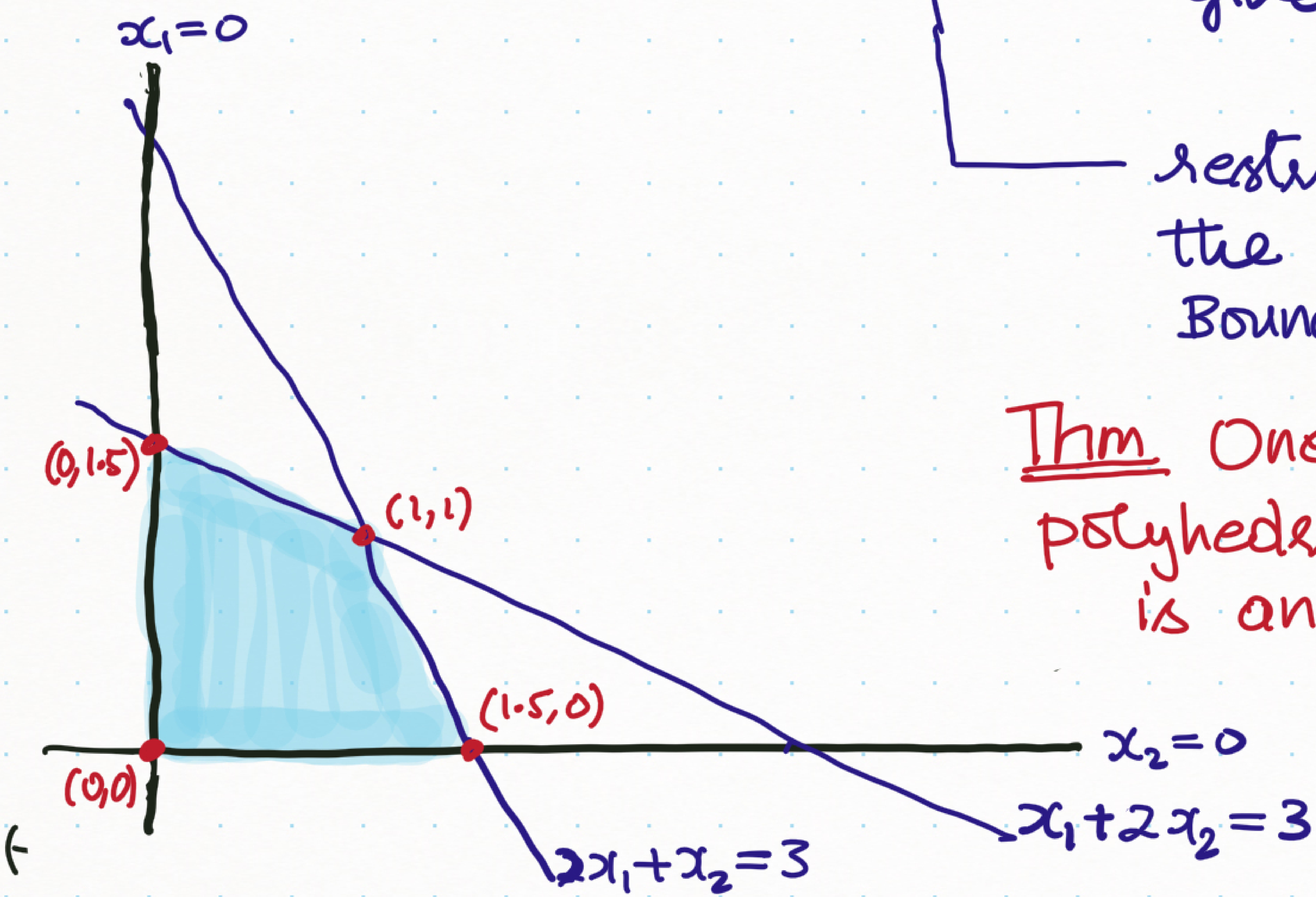
$2x_1 + x_2 \leq 3$

$x_1, x_2 \geq 0$

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

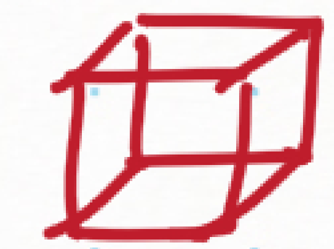
← Each constraint defines a half-plane (region with boundary given by the straight line $x_1 + 2x_2 = 3$)

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

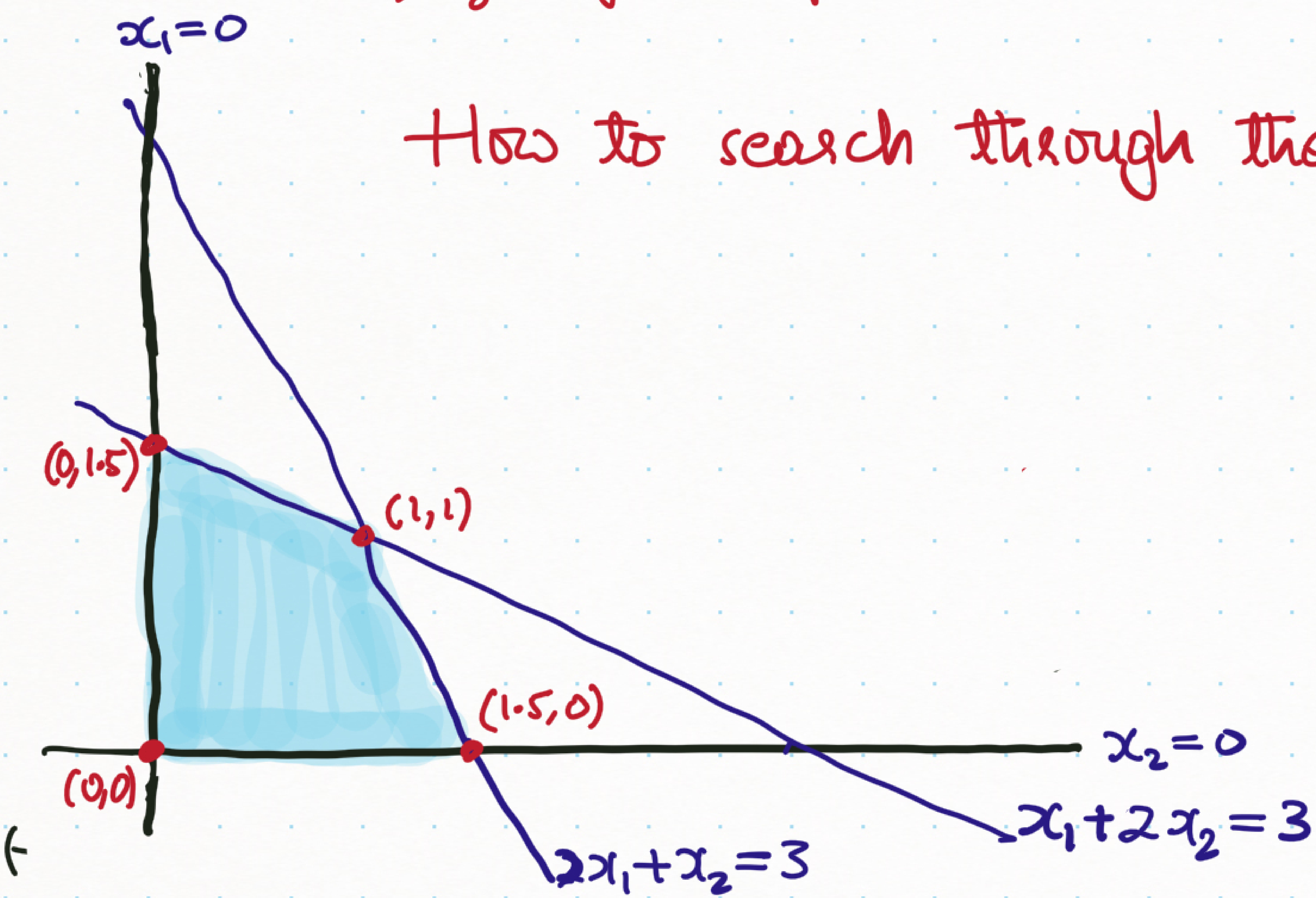


Thm One of the corners of the polyhedral feasible region is an optimal solution (if it exists).

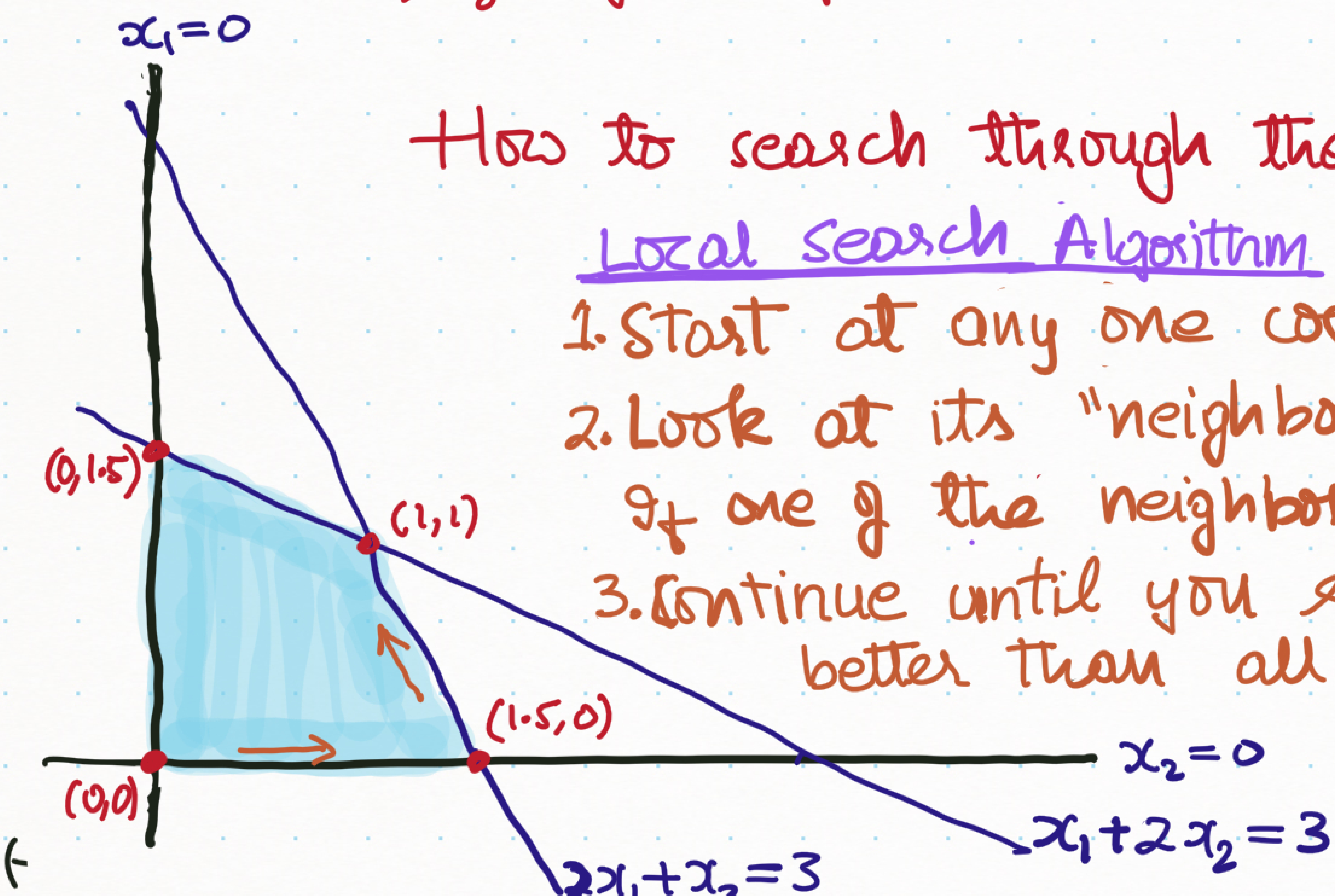
"Infinite problem reduced to finite problem"

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



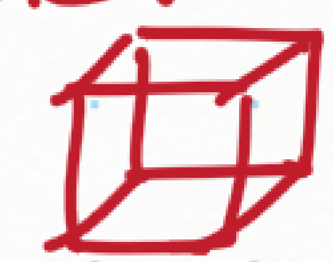
How to search through these corners systematically and efficiently?

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?

<u>Local Search Algorithm</u>

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 optimality.
Global (!!)