Understanding Integer Programming

Vitthal Srinivasan
CO-FOUNDER, LOONYCORN
www.loonycorn.com

Overview

Integer programming problems stipulate that decision variables be integers

Integer problems are even more widely used in business than LPPs

Solving some integer problems can be very mathematically complex

The LP-relaxation of an integer problem is the LPP that drops the integer constraint

LP-relaxations, if used right, greatly simplify solving integer problems

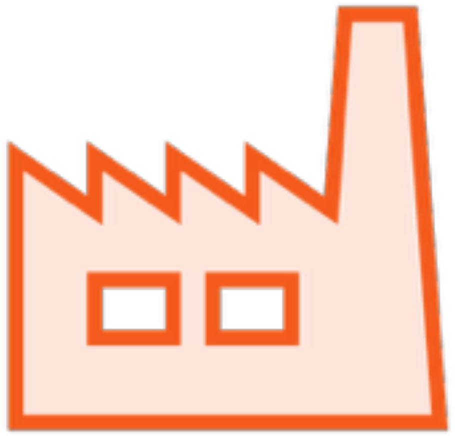
Integer Programming: Intuition

Summary

Winder glass - pg 518

Introduction - pg 507 - 509

A Famous Case Study: Wyndor Glass



Three Factories

Different plants for wood, aluminium and glass

Two Products

Glass doors and glass windows

Cost and Profit

Profit and effort per unit product are known

A Famous Case Study: Wyndor Glass

Production Facility	Production Time per Batch (Hours)		Production Time available per Week (hours)
	Product x_1	Product x_2	
Plant y_1	1	0	4
Plant y_2	0	2	12
Plant y_3	3	2	18
Profit per Batch	$3,000	$5,000	

Tweak production to maximise profits

Manufacturing as an Optimization Problem

Objective Function

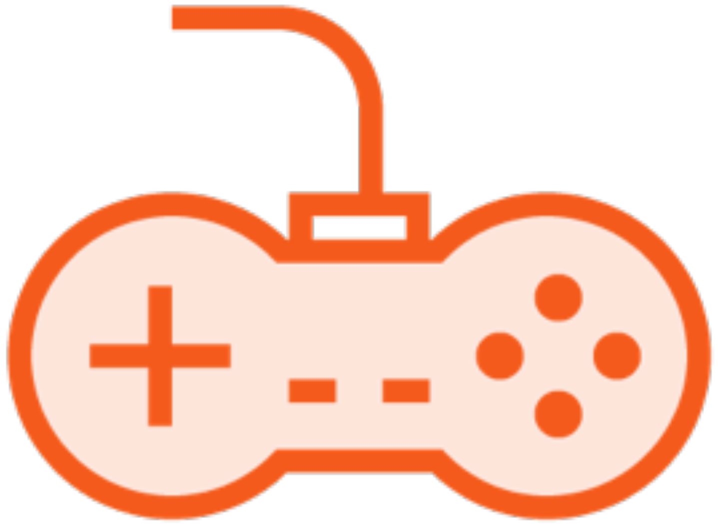
Maximize profits

Constraints

Plant capacity constraints

Decision Variables

How many batches of each product to produce



Decision Variables

x_1 = Number of batches of product 1 to produce

x_2 = Number of batches of product 2 to produce

A Famous Case Study: Wyndor Glass

	Production Time per Batch (Hours)		Production Time available per Week (hours)
	Product x_1	Product x_2	
Plant y_1	1	0	4
Plant y_2	0	2	12
Plant y_3	3	2	18
Profit per Batch	$3,000	$5,000	

Batches of Product 1 = x_1

A Famous Case Study: Wyndor Glass

	Production Time per Batch (Hours)		Production Time available per Week (hours)
	Product x_1	Product x_2	
Plant y_1	1	0	4
Plant y_2	0	2	12
Plant y_3	3	2	18
Profit per Batch	$3,000	$5,000	

Batches of Product 2 = x_2

Objective Function

Maximize profit Z

Z is total profit per week, in thousands of dollars

$Z = 3x_1 + 5x_2$

A Famous Case Study: Wyndor Glass

	Production Time per Batch (Hours)	
	Product x_1	Product x_2
Plant y_1	1	0
Plant y_2	0	2
Plant y_3	3	2

Production Time available per Week (hours)
4
12
18

Profit per Batch	$3,000	$5,000

$3x_1 \quad + \quad 5x_2$

A Famous Case Study: Wyndor Glass

	Production Time per Batch (Hours)		Production Time available per Week (hours)
	Product x_1	Product x_2	
Plant y_1	1	0	4
Plant y_2	0	2	12
Plant y_3	3	2	18

Profit per Batch	\$3,000	\$5,000

$$\text{Profit } Z = 3x_1 + 5x_2$$

Constraints

Infinite production is not possible

The production time available in the factories limits x_1 and x_2

A Famous Case Study: Wyndor Glass

	Production Time per Batch (Hours)			Production Time available per Week (hours)
	Product x_1	Product x_2		
Plant y_1	1 x_1 +	0 x_2	<=	4
Plant y_2	0	2		12
Plant y_3	3	2		18
Profit per Batch	$3,000	$5,000		

A Famous Case Study: Wyndor Glass

	Production Time per Batch (Hours)		Production Time available per Week (hours)
	Product x_1	Product x_2	
Plant y_1	1	0	4
Plant y_2	0	2	12
Plant y_3	3	2	18
Profit per Batch	$3,000	$5,000	

Constraint 1: $x_1 <= 4$

A Famous Case Study: Wyndor Glass

	Production Time per Batch (Hours)			Production Time available per Week (hours)
	Product x_1	Product x_2		
Plant y_1	1	0		4
Plant y_2	0 x_1 +	2 x_2	<=	12
Plant y_3	3	2		18
Profit per Batch	$3,000	$5,000		

A Famous Case Study: Wyndor Glass

	Production Time per Batch (Hours)		Production Time available per Week (hours)
	Product x_1	Product x_2	
Plant y_1	1	0	4
Plant y_2	0	2	12
Plant y_3	3	2	18
Profit per Batch	$3,000	$5,000	

Constraint 2: $2x_2 <= 12$

A Famous Case Study: Wyndor Glass

	Production Time per Batch (Hours)					Production Time available per Week (hours)
	Product x_1			Product x_2		
Plant y_1	1			0		4
Plant y_2	0			2		12
Plant y_3	3 x_1	+		2 x_2	<=	18
Profit per Batch	\$3,000			\$5,000		

A Famous Case Study: Wyndor Glass

	Production Time per Batch (Hours)		Production Time available per Week (hours)
	Product x_1	Product x_2	
Plant y_1	1	0	4
Plant y_2	0	2	12
Plant y_3	3	2	18

Profit per Batch	$3,000	$5,000

Constraint 3: $3x_1 + 2x_2 <= 18$

Linear Programming Problem Formulation

Maximize

$$Z = 3x_1 + 5x_2$$

Subject to constraints:

$$x_1 <= 4$$

$$2x_2 <= 12$$

$$3x_1 + 2x_2 <= 18$$

$x_1 , x_2 >= 0$ (Non-negativity constraints)

Constraints

Management decides to set aside slots for new product trials

The trials will neither increase profit nor costs for now

Each trial requires 6 hours of time in factory y_3

Linear Programming Problem Formulation

Maximize

$$Z = 3x_1 + 5x_2$$

Subject to constraints:

$$x_1 <= 4$$

$$2x_2 <= 12$$

$$\boxed{3x_1 + 2x_2 <= 18}$$

$$x_1 , x_2 >= 0$$ **(Non-negativity constraints)**

Linear Programming Problem Formulation

Maximize

$$Z = 3x_1 + 5x_2$$

Subject to constraints:

$$x_1 <= 4$$

$$2x_2 <= 12$$

$$\boxed{3x_1 + 2x_2 <= 18}$$

$$x_1 , x_2 >= 0$$ (Non-negativity constraints)

Linear Programming Problem Formulation

Maximize

$$Z = 3x_1 + 5x_2$$

Subject to constraints:

$$x_1 <= 4$$

$$2x_2 <= 12$$

A new type of constraint

$3x_1 + 2x_2 = 6$ OR $3x_1 + 2x_2 = 12$ OR $3x_1 + 2x_2 = 18$

$x_1 , x_2 >= 0$ (Non-negativity constraints)

An Integer Constraint

$3x_1 + 2x_2 = 6$ OR $3x_1 + 2x_2 = 12$ OR $3x_1 + 2x_2 = 18$

Define auxiliary variables z_1, z_2, z_3

Redefine original constraint as $3x_1 + 2x_2 = 6z_1 + 12z_2 + 18z_3$

Add constraints on the auxiliary variables

$z_1 + z_2 + z_3 = 1$

z_1, z_2, z_3 are binary

i.e. $z_1, z_2, z_3 \in \{0,1\}$

Linear Programming Problem Formulation

Maximize

$$Z = 3x_1 + 5x_2$$

Subject to constraints:

$$x_1 <= 4$$

$$2x_2 <= 12$$

$$3x_1 + 2x_2 <= 18$$

$x_1 , x_2 >= 0$ (Non-negativity constraints)

Integer Programming Problem Formulation

Maximize

$$Z = 3x_1 + 5x_2$$

Subject to constraints:

$x_1 <= 4$

$2x_2 <= 12$

$3x_1 + 2x_2 = 6z_1 + 12z_2 + 18z_3$

$z_1 + z_2 + z_3 = 1$

$z_1, z_2, z_3 \in \{0,1\}$ (Binary integer constraint)

$x_1 , x_2 >= 0$ (Non-negativity constraints)

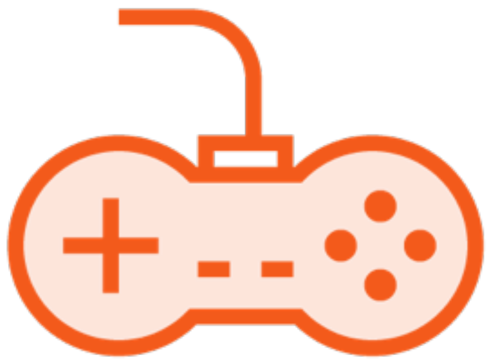
Micro-economic Assumptions: Linear Programming

Proportionality Assumption

No start-up costs, constant returns to scale

Additivity Assumption

Products are neither complements nor substitutes



Divisibility Assumption

Fractional production is possible

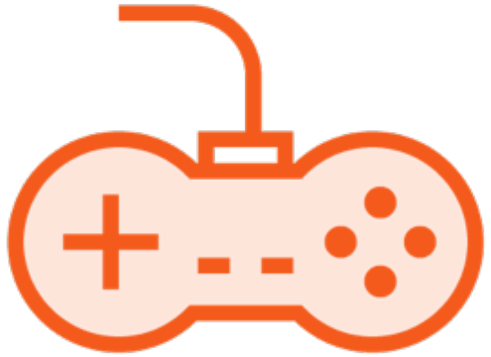
Micro-economic Assumptions: Linear Programming

Proportionality Assumption

No start-up costs, constant returns to scale

Additivity Assumption

Products are neither complements nor substitutes



Divisibility Assumption

Fractional production is possible

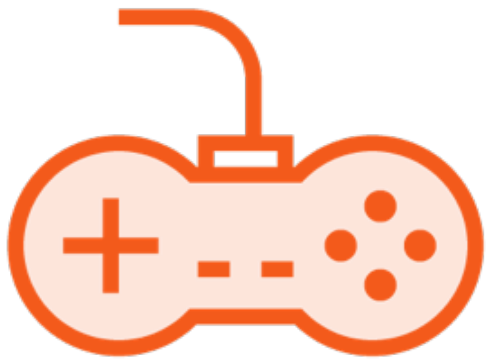
Micro-economic Assumptions: Integer Programming

Proportionality Assumption

No start-up costs, constant returns to scale

Additivity Assumption

Products are neither complements nor substitutes



No Divisibility Assumption

Fractional production is not possible

Integer Linear Programming

Pure integer programming

All decision variables must be integers

Mixed integer programming

Some decision variables must be integers

Binary integer programming

All integer variables must be 0 or 1

Integer programming still requires objective and constraints to be linear

Integer programming still requires objective and constraints to be linear

Integer Programming: Solutions

Summary

Difficulties in solving - pg 529 to 531

LP-relaxation - pg 531

Rounding off - problem 1 - pg 532

Rounding off - problem 2 - pg 533

Linear Programming Problem Formulation

Maximize

$$Z = 3x_1 + 5x_2$$

Subject to constraints:

$$x_1 <= 4$$

$$2x_2 <= 12$$

$$3x_1 + 2x_2 <= 18$$

$$x_1 , x_2 >= 0$$ (Non-negativity constraints)

Integer Programming Problem Formulation

Maximize

$$Z = 3x_1 + 5x_2$$

Subject to constraints:

$$x_1 <= 4$$

$$2x_2 <= 12$$

$$3x_1 + 2x_2 = 6z_1 + 12z_2 + 18z_3$$

$$z_1 + z_2 + z_3 = 1$$

$z_1, z_2, z_3 \in \{0,1\}$ **(Binary integer constraint)**

$x_1 , x_2 >= 0$ **(Non-negativity constraints)**

Integer programming can be far more difficult to solve efficiently than LPPs

Linear Programming Problem Formulation

Maximize

$$Z = 3x_1 + 5x_2$$

Subject to constraints:

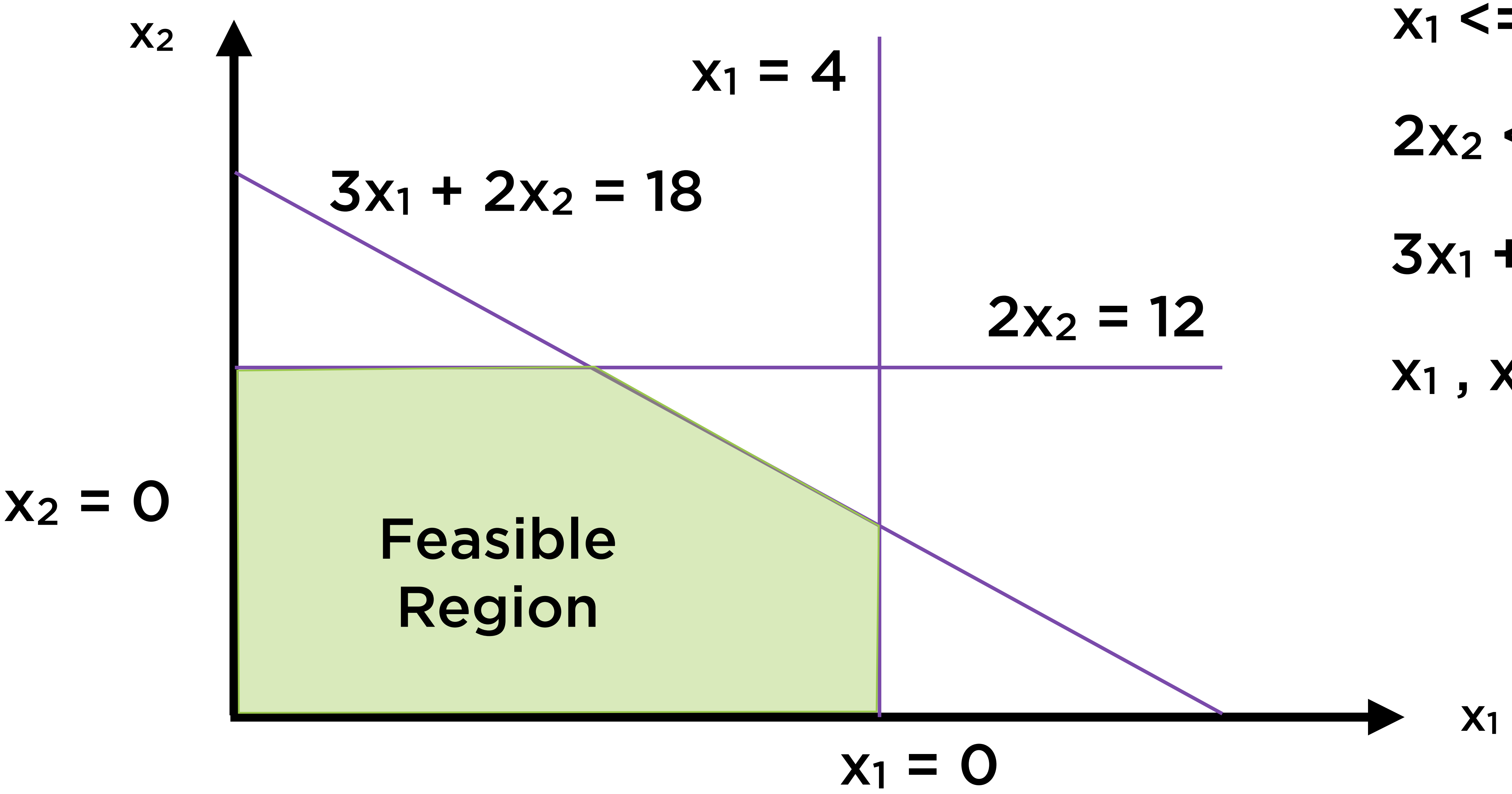
$$x_1 <= 4$$

$$2x_2 <= 12$$

$$3x_1 + 2x_2 <= 18$$

$$x_1 , x_2 >= 0$$ (Non-negativity constraints)

LPP Constraints in Space



$x_1 <= 4$

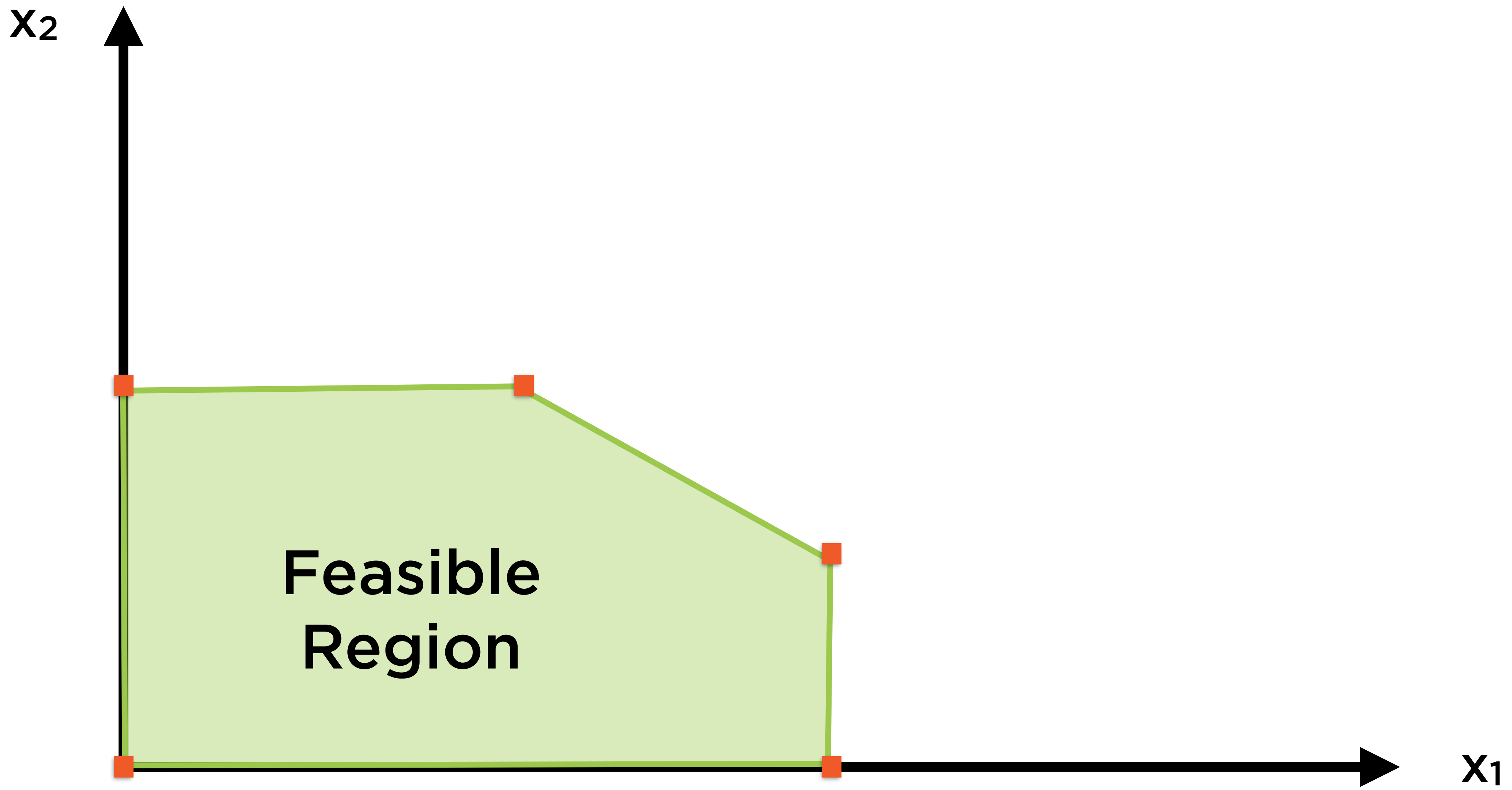
$2x_2 <= 12$

$3x_1 + 2x_2 <= 18$

$x_1 , x_2 >= 0$

Each constraint bounds the feasible region

LPP Constraints in Space



The optimal solution will always* be a corner point of this feasible region

The optimal solution will always* be a corner point of the feasible region

* Not guaranteed to hold for integer programming problems

Constraints

LPP solution algorithms heavily rely on the corner-point property

Integer problems may need to search the entire feasible region

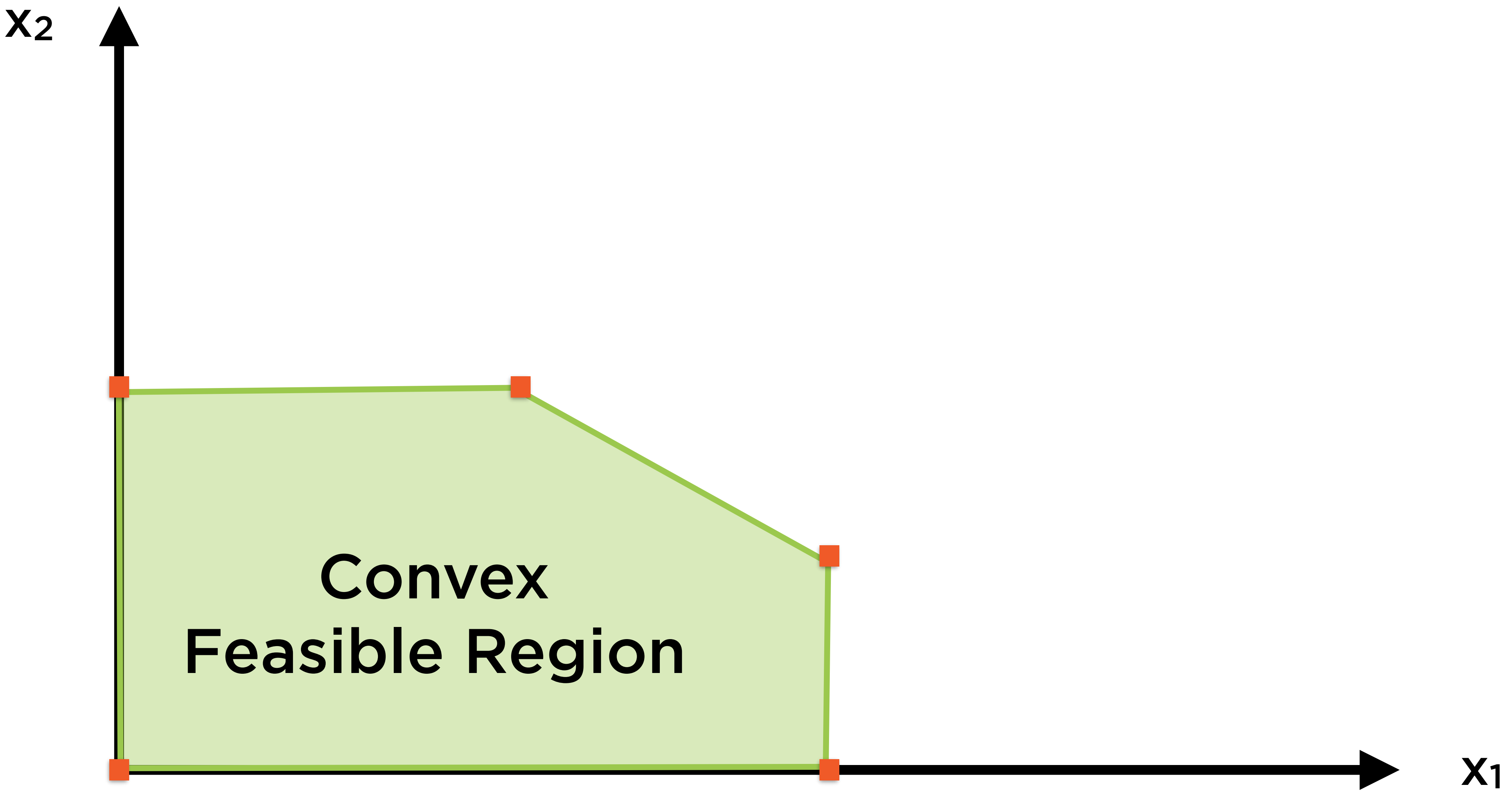
This can make it very very difficult to solve integer problems

Constraints

LPP solution algorithms heavily rely on the corner-point property

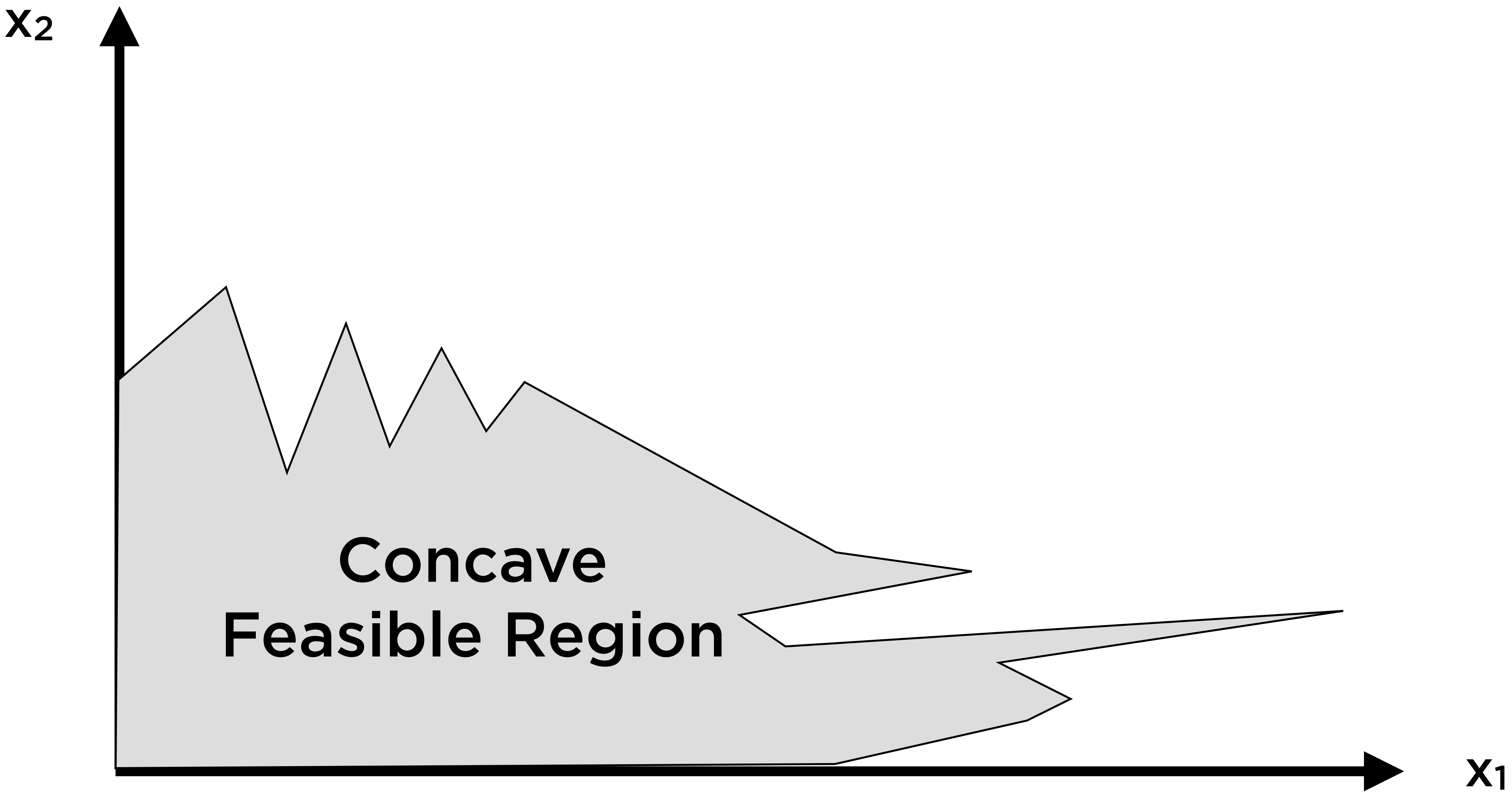
This only holds when the feasible region is convex

Convex Feasible Region



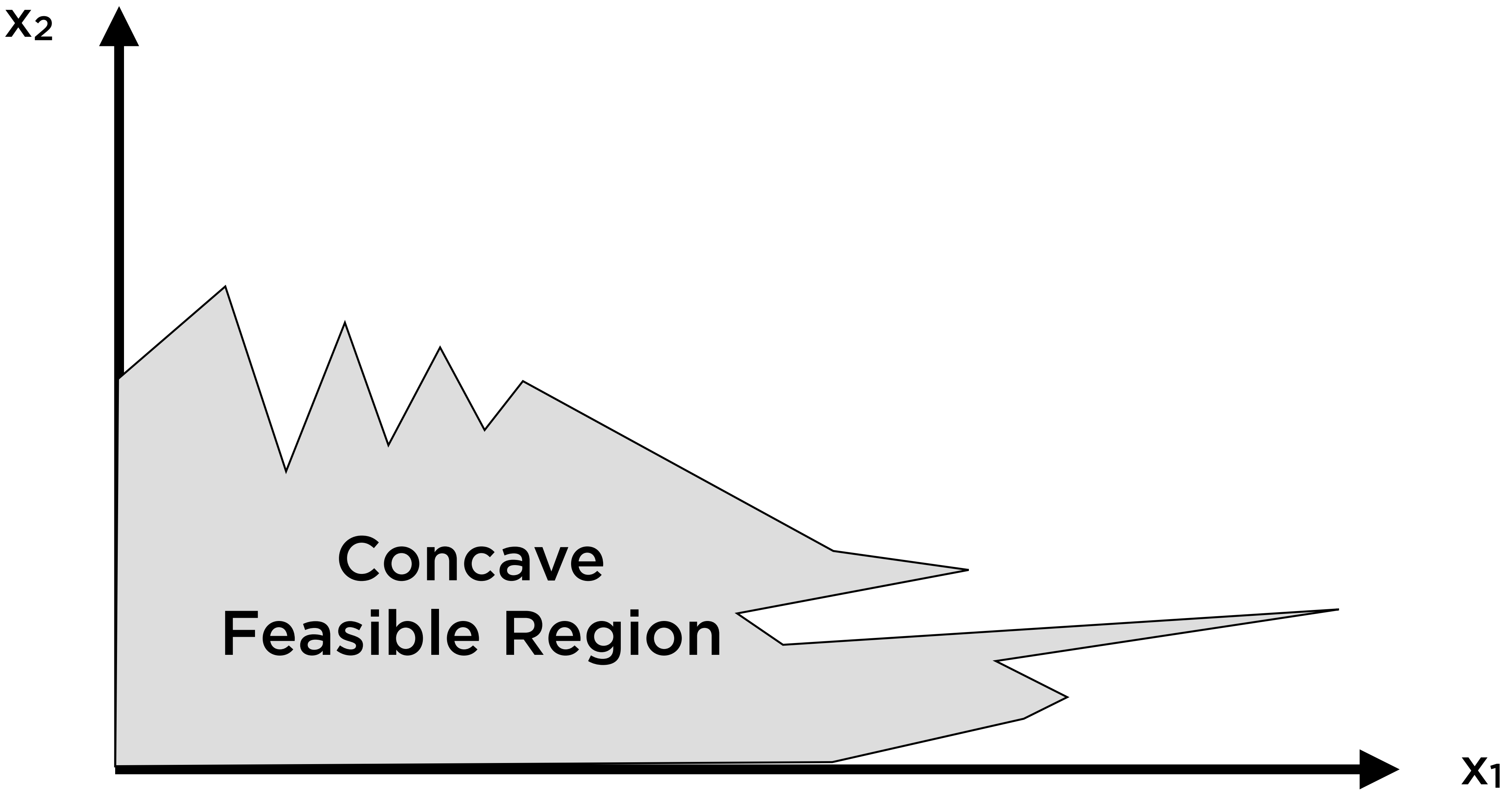
The optimal solution will always* be a corner point of this feasible region

Concave Feasible Region



The optimal solution **need not be** be a corner point of this feasible region

Concave Feasible Region



Integer constraints lead to feasible regions with jagged edges, just like this one

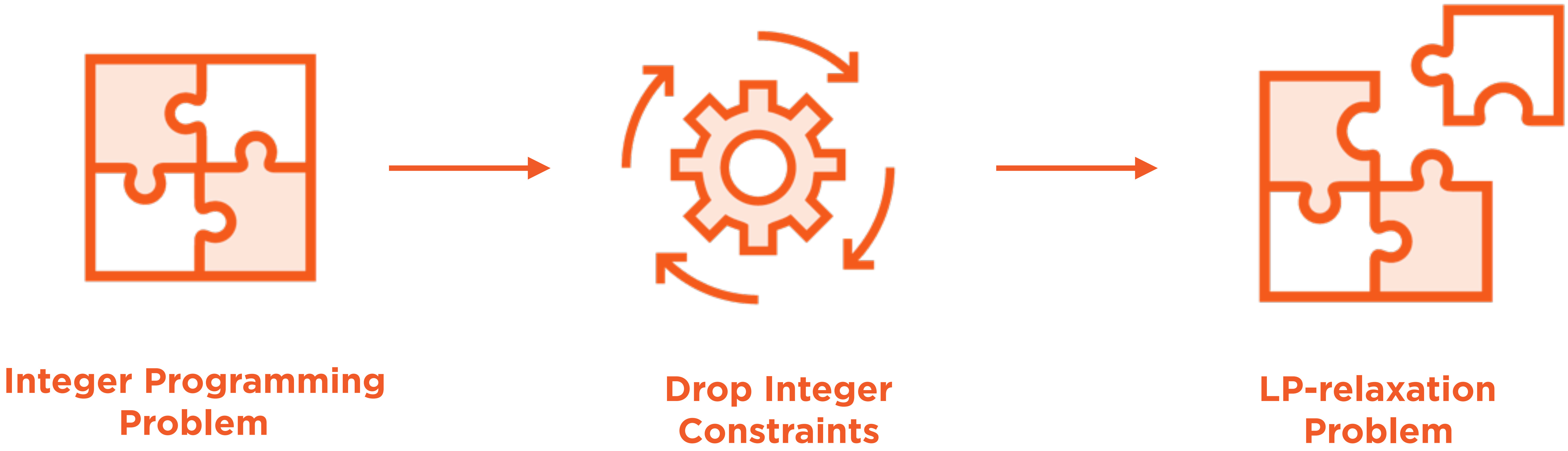
LP-relaxation

Take an integer programming problem...

...Relax the integer restriction

The resulting LPP is called the LP-relaxation of the integer problem

LP-relaxation of Integer Problem



The LP-relaxation is used as a **starting point** in solving the original integer problem

Integer Programming Problem Formulation

Maximize

$$Z = 3x_1 + 5x_2$$

Subject to constraints:

$$x_1 <= 4$$

$$2x_2 <= 12$$

$$3x_1 + 2x_2 = 6z_1 + 12z_2 + 18z_3$$

$$z_1 + z_2 + z_3 = 1$$

$z_1, z_2, z_3 \in \{0,1\}$ (Binary integer constraint)

$x_1 , x_2 >= 0$ (Non-negativity constraints)

LP-relaxation of Integer Problem

Maximize

$$Z = 3x_1 + 5x_2$$

Subject to constraints:

$$x_1 <= 4$$

$$2x_2 <= 12$$

$$3x_1 + 2x_2 = 6z_1 + 12z_2 + 18z_3$$

$$z_1 + z_2 + z_3 = 1$$

$\cancel{z_1, z_2, z_3 \in \{0,1\}}$ ~~(Binary integer constraint)~~

$x_1, x_2 >= 0$ (Non-negativity constraints)

Original LPP $\neq$ LP-relaxation

Maximize

$$Z = 3x_1 + 5x_2$$

Subject to constraints:

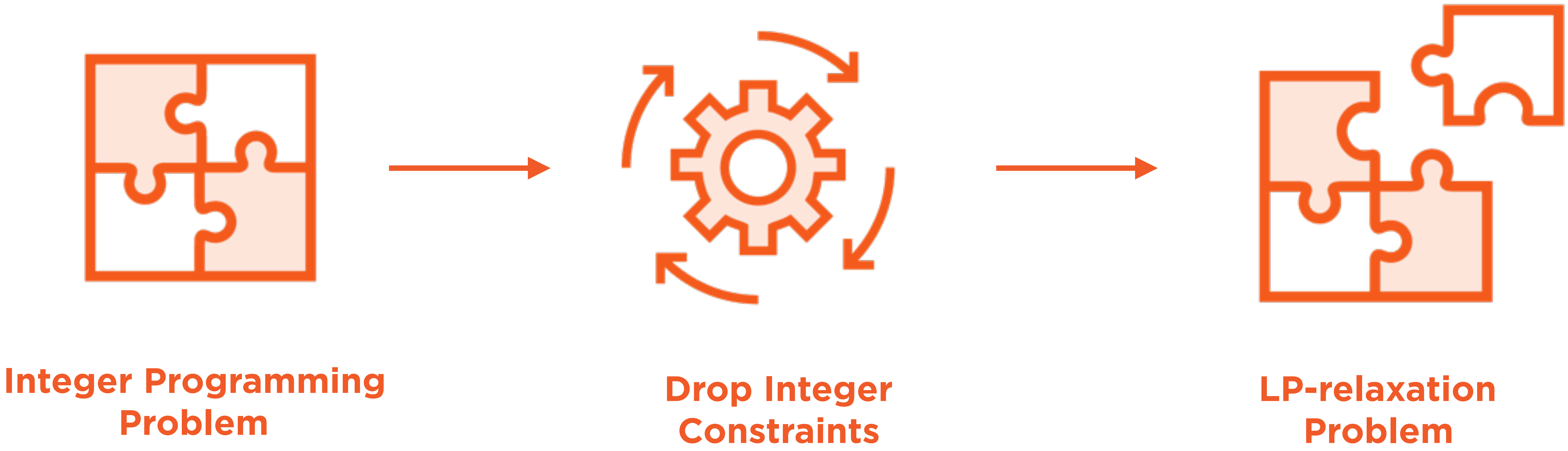
$$x_1 <= 4$$

$$2x_2 <= 12$$

$3x_1 + 2x_2 = 6$ OR $3x_1 + 2x_2 = 12$ OR $3x_1 + 2x_2 = 18$

$x_1 , x_2 >= 0$ (Non-negativity constraints)

LP-relaxation of Integer Problem



The LP-relaxation is used as a starting point in solving the original integer problem

Constraints

It is tempting but wrong to solve the LPP and round off the answers

Perils of Rounding Off

May not be feasible

Rounding optimal solution of LP-relaxation may not even be feasible for integer problem

May not be optimal

Rounding optimal solution of LP-relaxation may not be even close to optimal for integer problem

It is tempting but wrong to solve the LPP and round off the answers

Integer Problem

Maximize

$$Z = x_2$$

Subject to constraints:

$$2x_2 - 2x_1 <= 1$$

$$2x_2 + 2x_1 <= 7$$

x_1, x_2 are integers (Integer constraint)

$x_1 , x_2 >= 0$ (Non-negativity constraints)

LP-relaxation of Integer Problem

Maximize

$$Z = x_2$$

Subject to constraints:

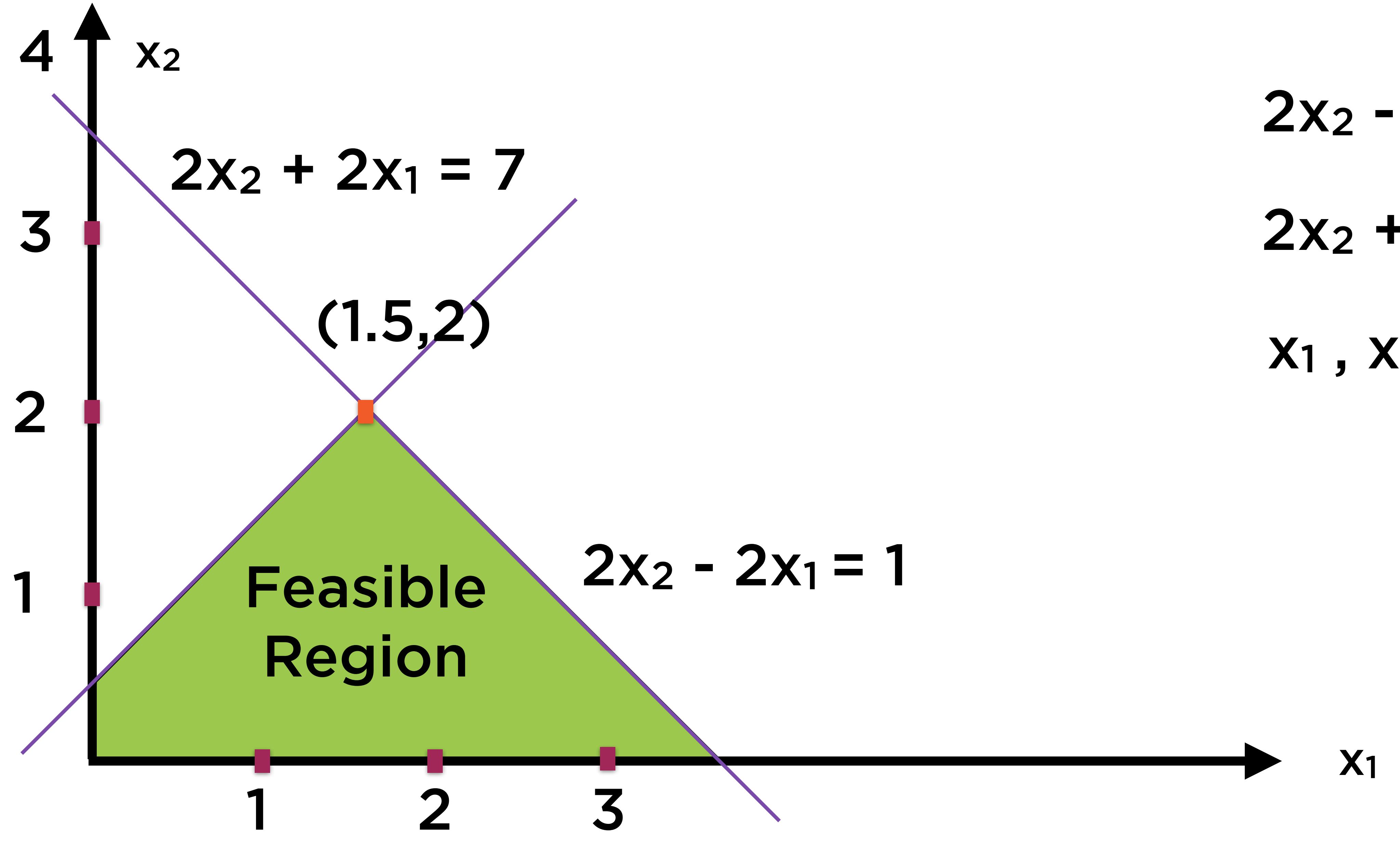
$$2x_2 - 2x_1 <= 1$$

$$2x_2 + 2x_1 <= 7$$

~~x_1, x_2 are integers (Integer constraint)~~

$x_1 , x_2 >= 0$ (Non-negativity constraints)

Solution of LP-relaxation



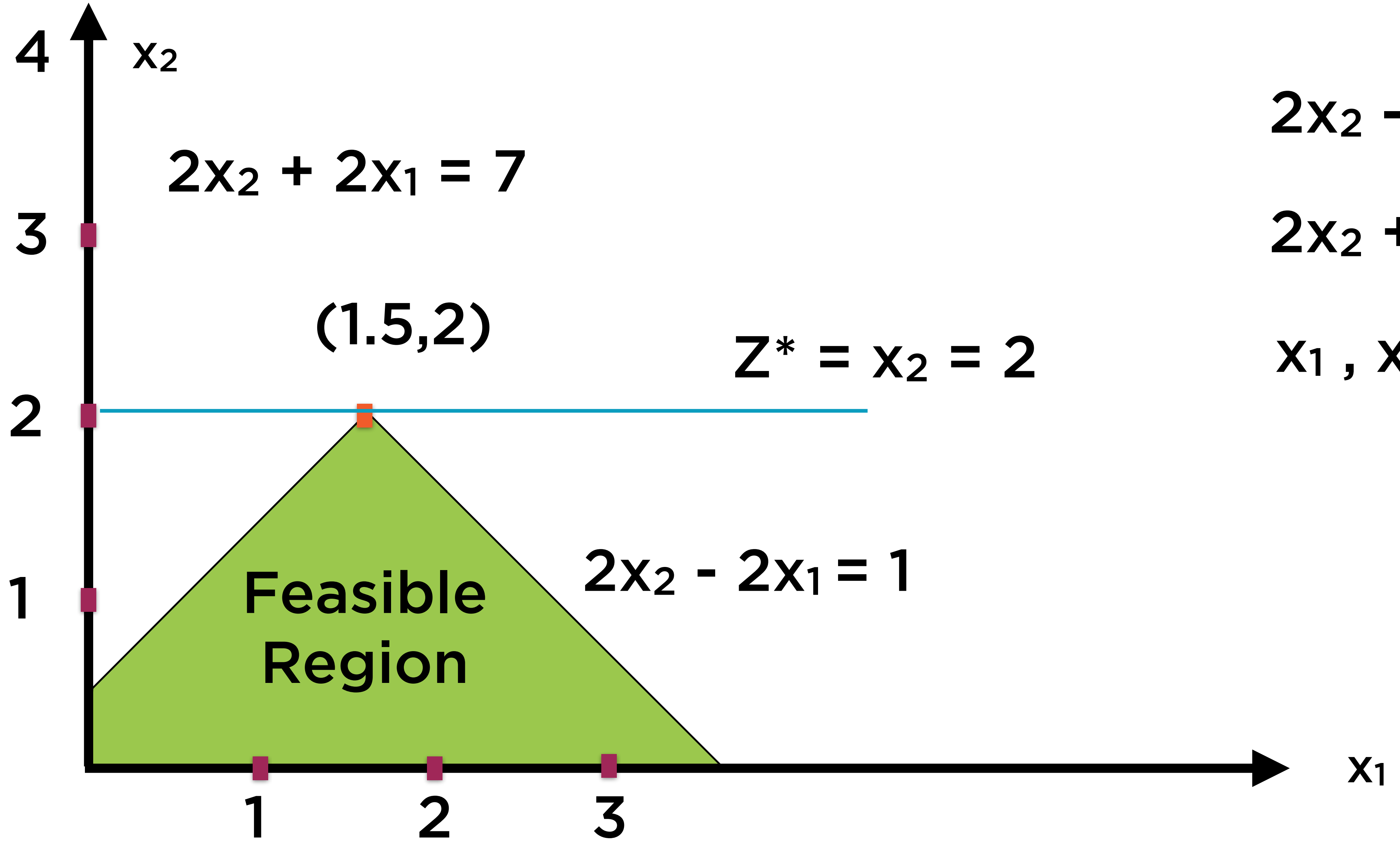
$2x_2 - 2x_1 <= 1$

$2x_2 + 2x_1 <= 7$

$x_1 , x_2 >= 0$

Represent the constraints as boundaries of the feasible region

Solution of LP-relaxation



$2x_2 - 2x_1 <= 1$

$2x_2 + 2x_1 <= 7$

$x_1 , x_2 >= 0$

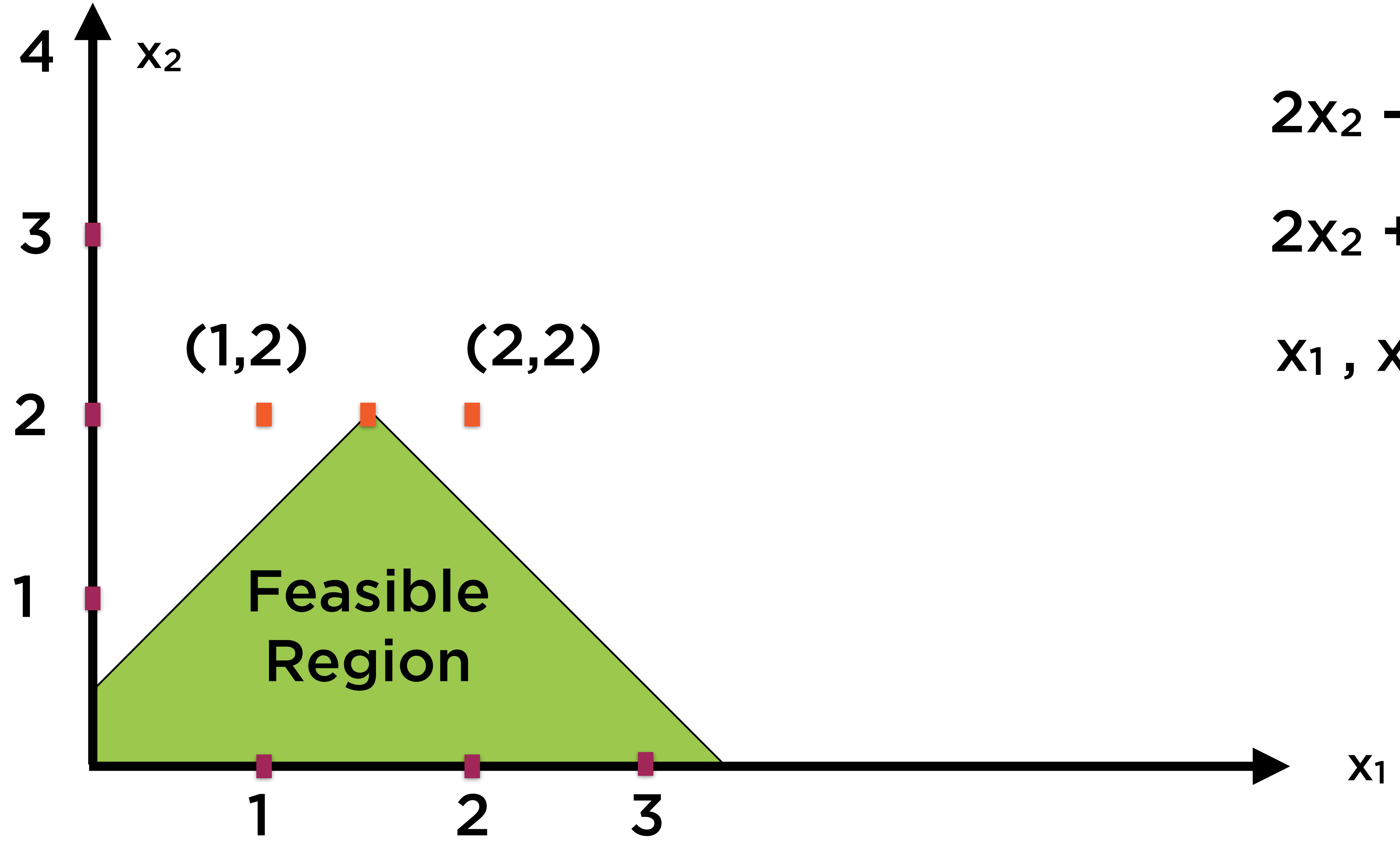
Optimal solution of the LP-relaxation is the point (1.5,2), where Z = 2

Constraints

Rounding the solution (1.5,2) gives us 2 candidate solutions

- Candidate 1: (1,2)
- Candidate 2: (2,2)

Solution of LP-relaxation



$2x_2 - 2x_1 <= 1$

$2x_2 + 2x_1 <= 7$

$x_1 , x_2 >= 0$

Rounding the solution (1.5,2) gives us 2 candidate solutions

Constraints

Try plugging these into the original problem

- Candidate 1: $x_1 = 1, x_2 = 2$
- Candidate 2: $x_1 = 2, x_2 = 2$

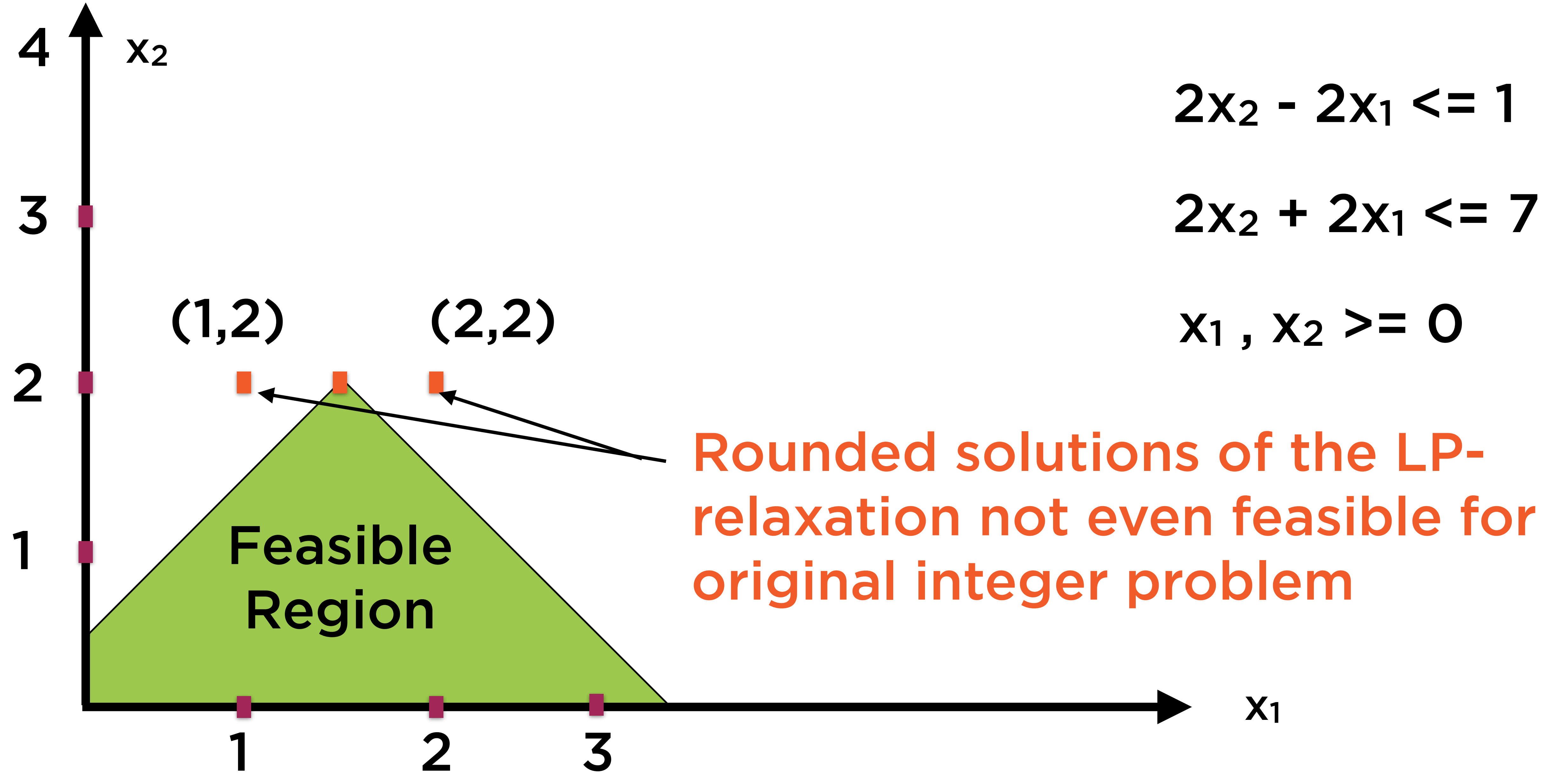
Candidate 1 violates the constraint $2x_2 - 2x_1 <= 1$

- Because $2x2 - 2x1 = 4 - 2 = 2 > 1$

Candidate 2 violates the constraint $2x_2 + 2x_1 <= 7$

- Because $2x2 + 2x2 = 4 + 4 = 8 > 7$

Solution of LP-relaxation



$2x_2 - 2x_1 <= 1$

$2x_2 + 2x_1 <= 7$

$x_1 , x_2 >= 0$

Optimal solution of the LP-relaxation is the point (1.5,2), where Z = 2

Constraints

Rounded solution of the LP-relaxation are not even feasible for original integer problem

Perils of Rounding Off

May not be feasible

Optimal solution of LP-relaxation may not even be feasible for integer problem

May not be optimal

Optimal solution of LP-relaxation may not be even close to optimal for integer problem

It is tempting but wrong to solve the LPP and round off the answers

Integer Problem

Maximize

$$Z = x_1 + 5x_2$$

Subject to constraints:

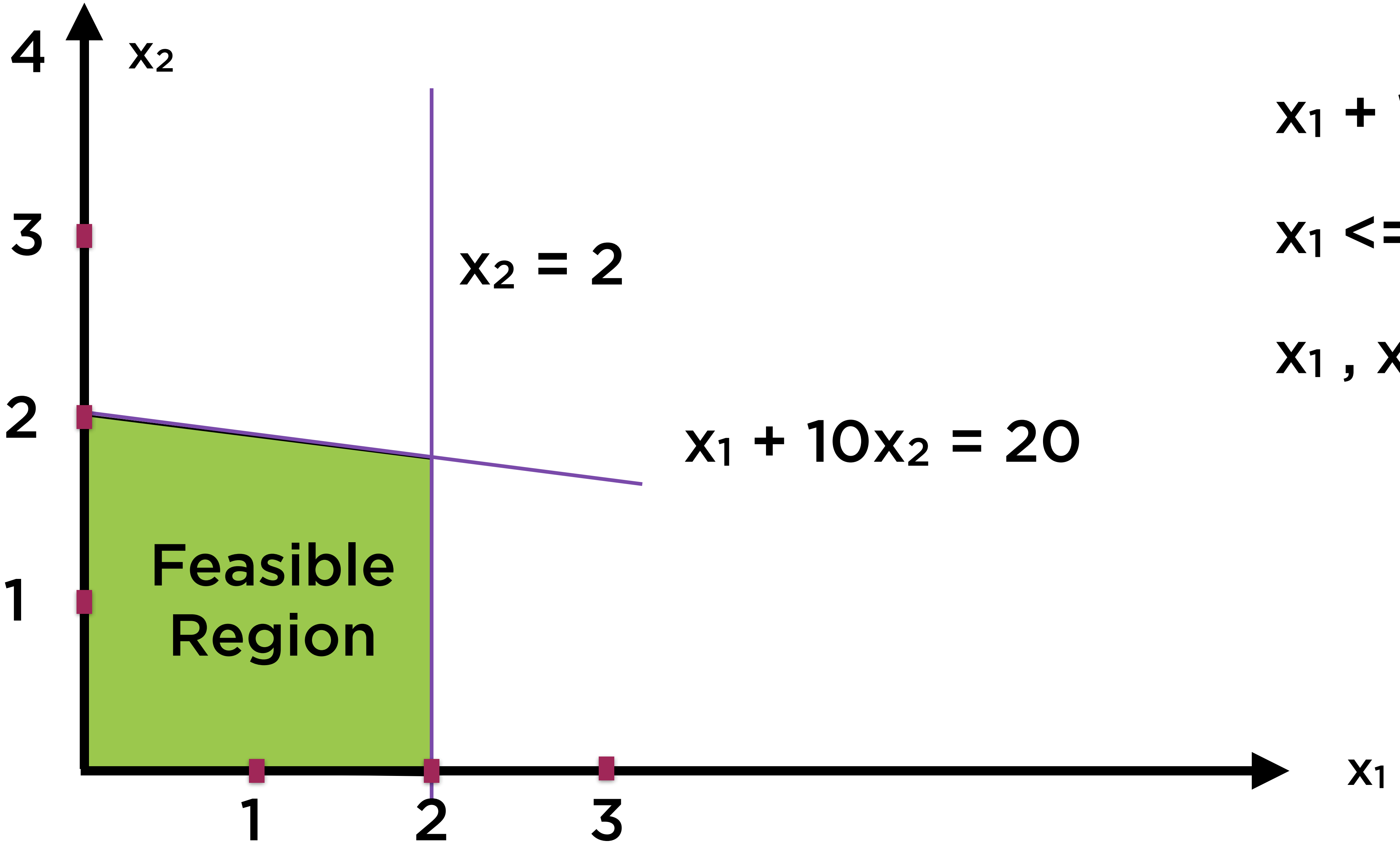
$$x_1 + 10x_2 <= 20$$

$$x_1 <= 2$$

x_1, x_2 **are integers** **(Integer constraint)**

$x_1 , x_2 >= 0$ **(Non-negativity constraints)**

Solution of LP-relaxation



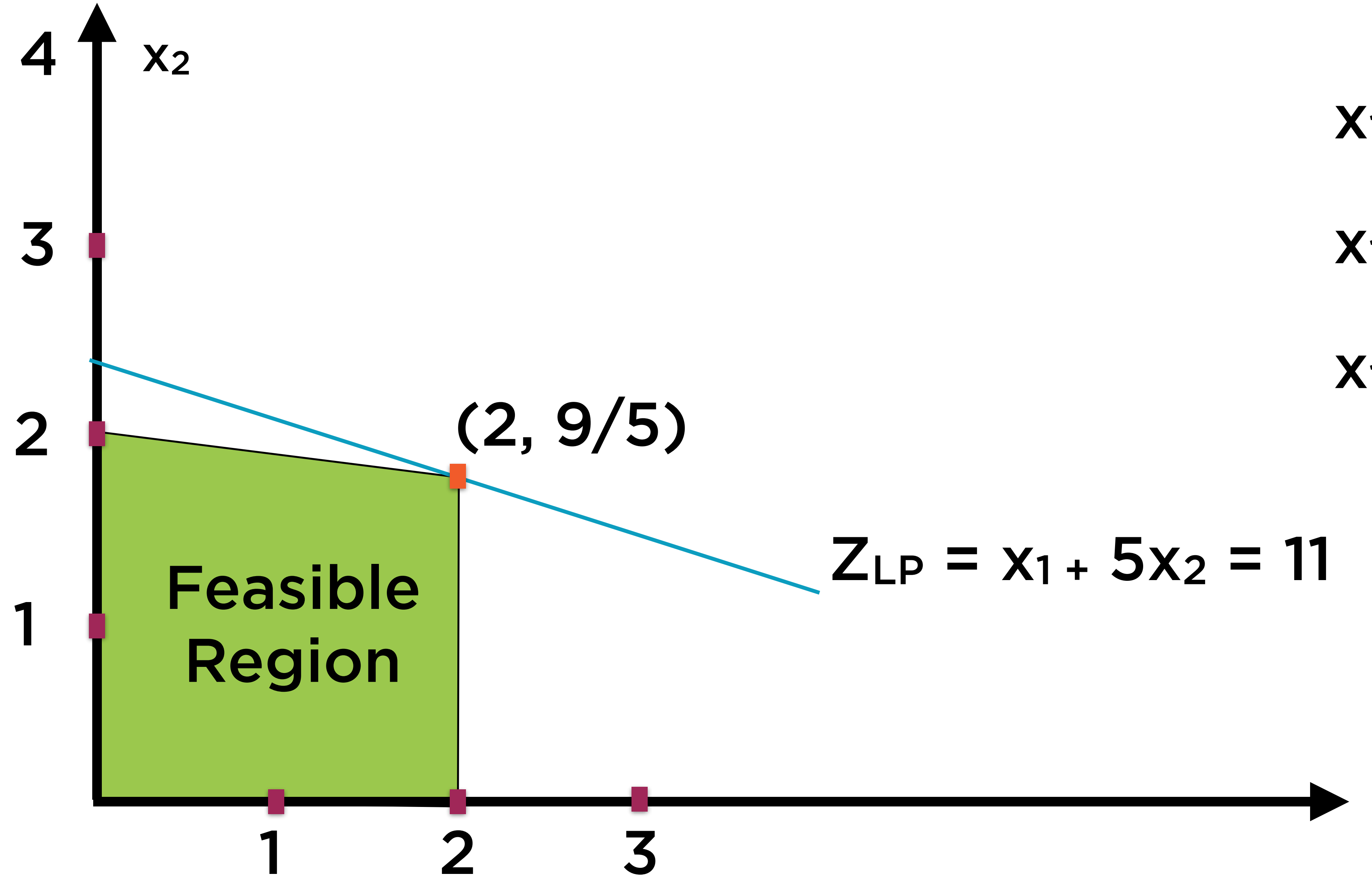
$x_1 + 10x_2 <= 20$

$x_1 <= 2$

$x_1 , x_2 >= 0$

Represent the constraints as boundaries of the feasible region

Solution of LP-relaxation



$x_1 + 10x_2 <= 20$

$x_1 <= 2$

$x_1 , x_2 >= 0$

Optimal solution of the LP-relaxation is the point (2,9/5), where $Z_{LP} = 11$

Constraints

Rounding the solution (2, 9/5) gives us 2 candidate solutions

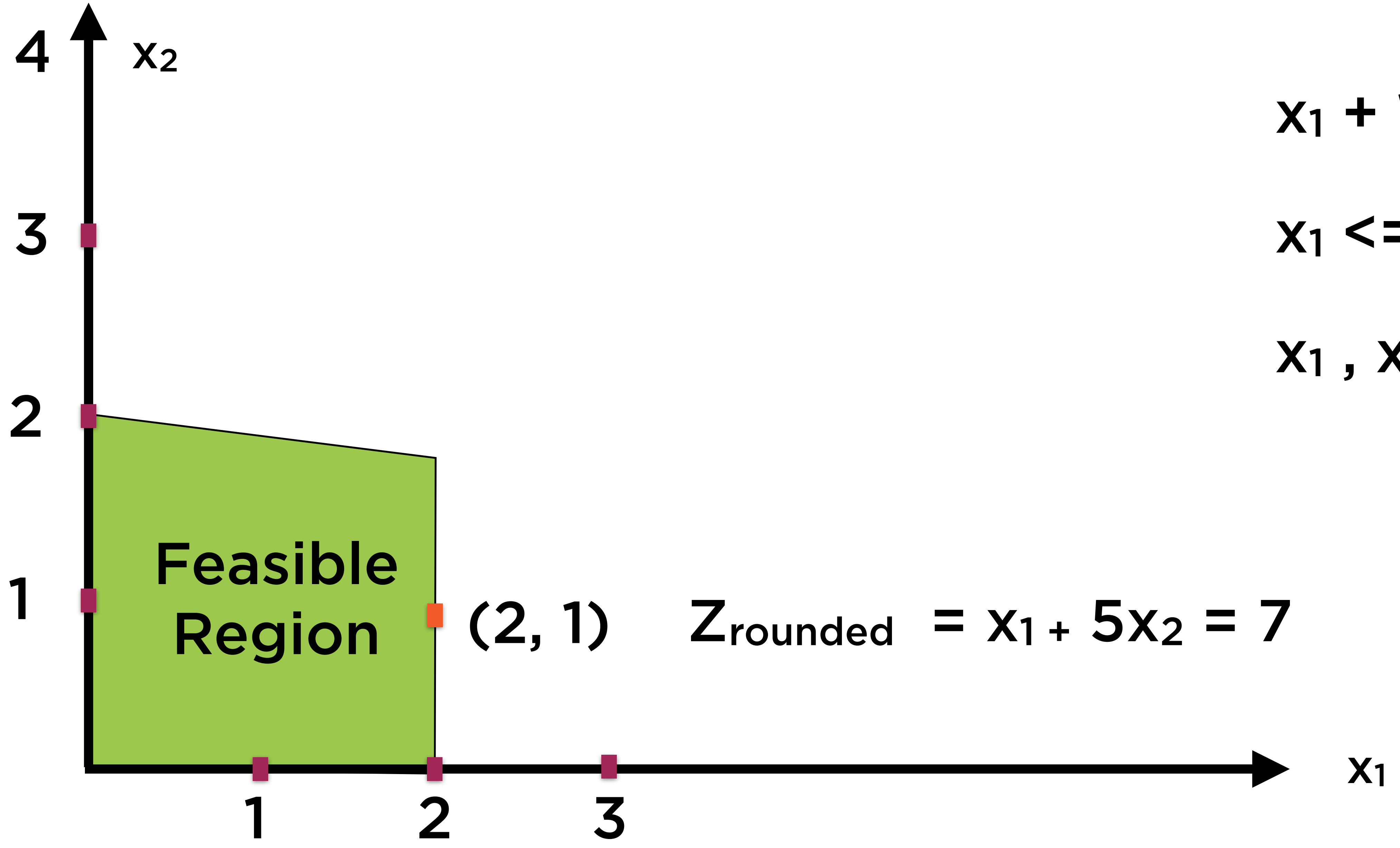
- Candidate 1: (2,2)
- Candidate 2: (2,1)

Candidate 1 violates a constraint, so abandon it

- Because 2 + 2x10 = 2 + 20 = 22 > 20

Rounded solution = (2,1)

Solution of LP-relaxation



$x_1 + 10x_2 <= 20$

$x_1 <= 2$

$x_1 , x_2 >= 0$

$Z_{rounded} = x_1 + 5x_2 = 7$

At rounded solution, $Z_{rounded} = 7$

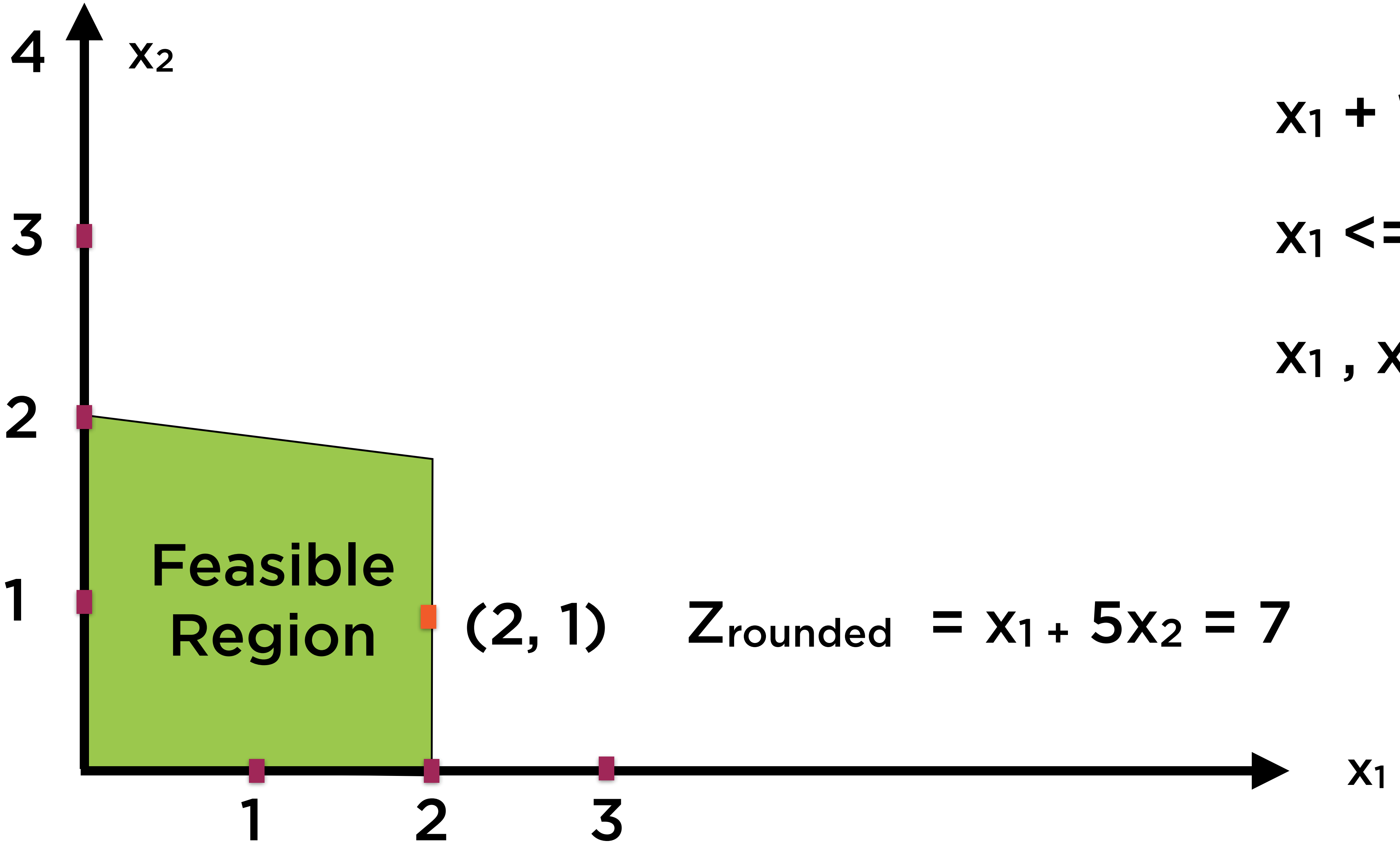
Constraints

But, the actual optimal solution of the integer problem is (0,2)

Ironically, this is a corner-point solution!

At (0,2), $Z = x_1 + 5x_2 = 10$

Solution of LP-relaxation



$x_1 + 10x_2 <= 20$

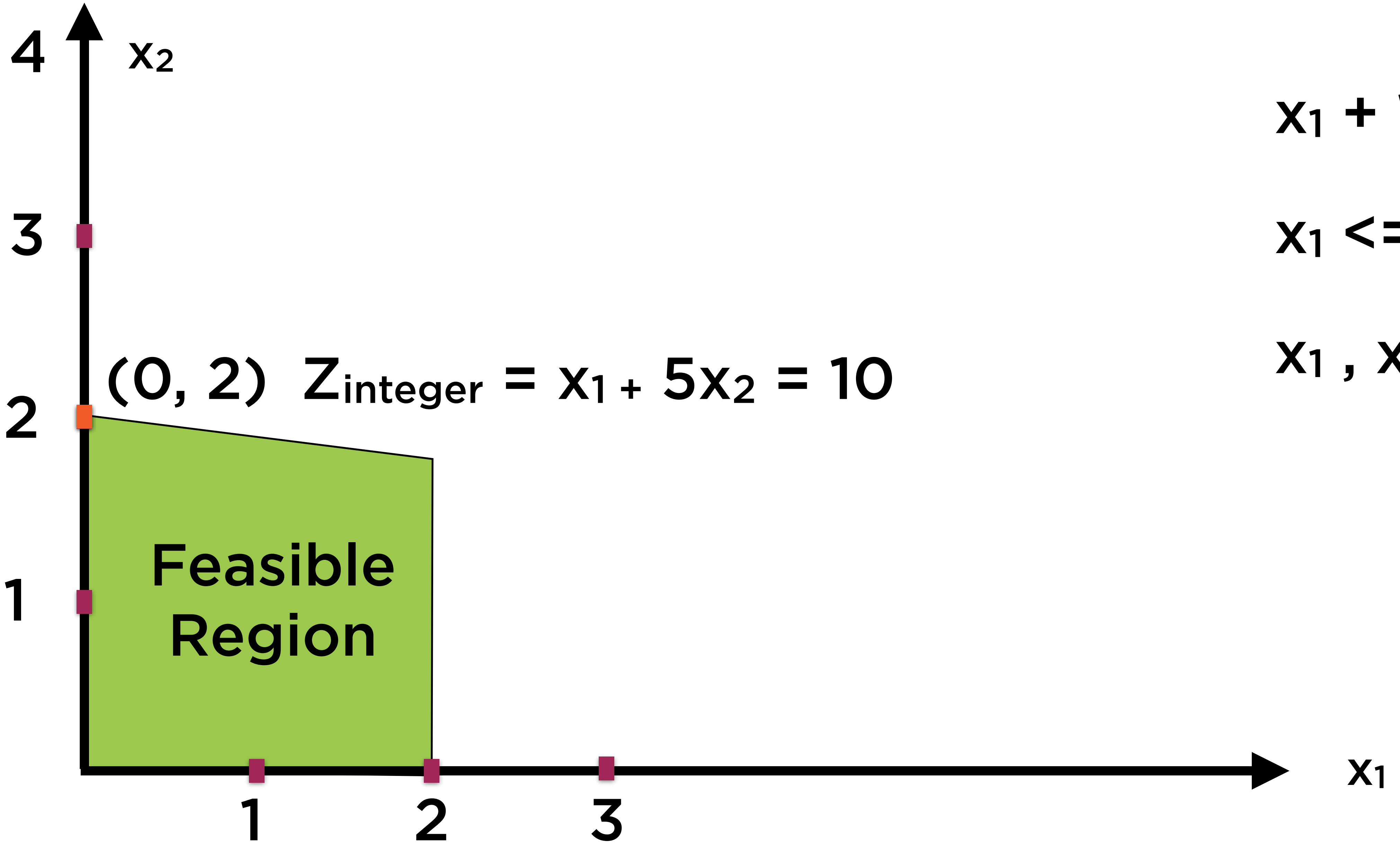
$x_1 <= 2$

$x_1 , x_2 >= 0$

$Z_{rounded} = x_1 + 5x_2 = 7$

At rounded solution, $Z_{rounded} = 7$

Solution of LP-relaxation



$x_1 + 10x_2 <= 20$

$x_1 <= 2$

$x_1 , x_2 >= 0$

Actual optimal solution, $Z_{integer} = 10$

Objective Function

Actual optimal solution:

$Z_{integer} = 10$

Optimal solution of the LP-relaxation:

$Z_{LP} = 11$

Rounded optimal solution of the LP-relaxation:

$Z_{rounded} = 7$

Perils of Rounding Off

May not be feasible

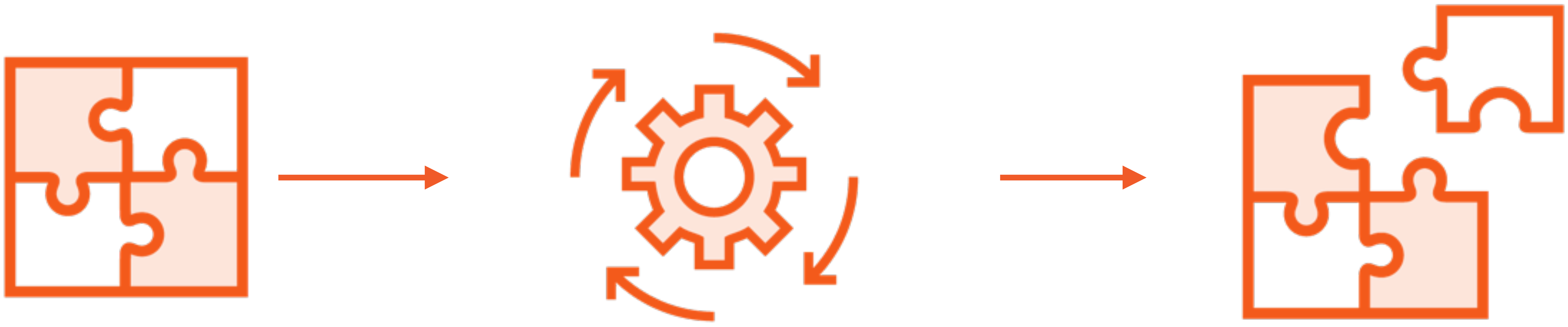
Optimal solution of LP-relaxation may not even be feasible for integer problem

May not be optimal

Optimal solution of LP-relaxation may not be even close to optimal for integer problem

It is tempting but wrong to solve the LPP and round off the answers

LP-relaxation of Integer Problem



Integer Programming Problem

Drop Integer Constraints

LP-relaxation Problem

The LP-relaxation is used as a starting point in solving the original integer problem

Constraints

However, if the solution of LP-relaxation satisfies integer constraint…

…then it is an optimal solution for integer problem as well!

Constraints

Many optimization problems are constructed to be solvable this way

- transportation problem
- assignment problem
- shortest path problem
- maximum flow problem

Integer Programming: Applications

Summary

Assignment problem - pg 361

Investment analysis - pg 511

Site selection - pg 512

Production-and-distribution - pg 512

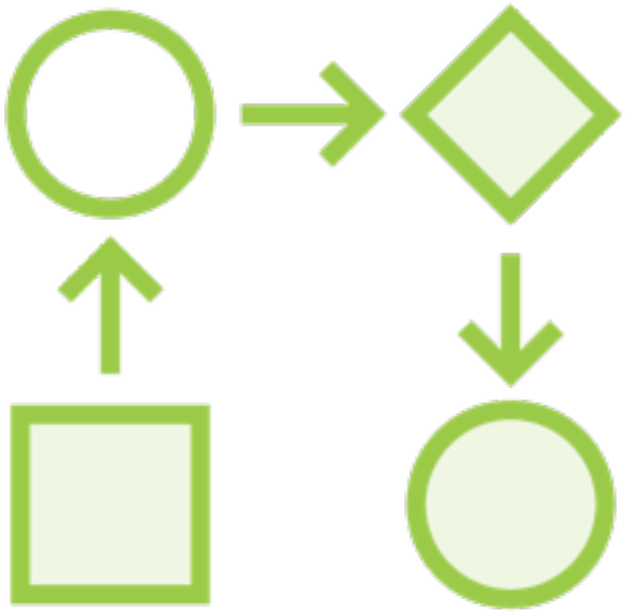
Shipments - pg 513

Inter-related activities - pg 513

Matching People and Problems

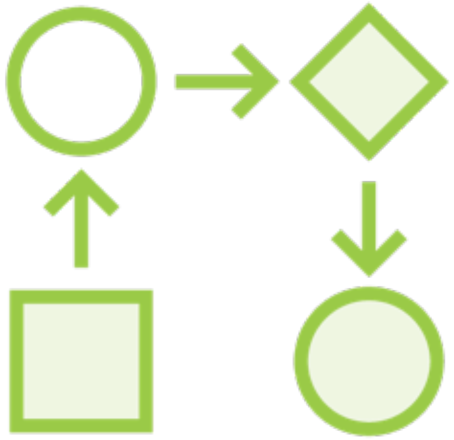
People

Each team-member has unique skills, likes and talents



Problems

Tasks that need to be completed, each with different costs



People-problem matching

n people, n tasks

Each person assigned 1 task

Each task performed by 1 person

Minimize total cost, given that...

...Cost of person i performing task j is given by $c_{i,j}$

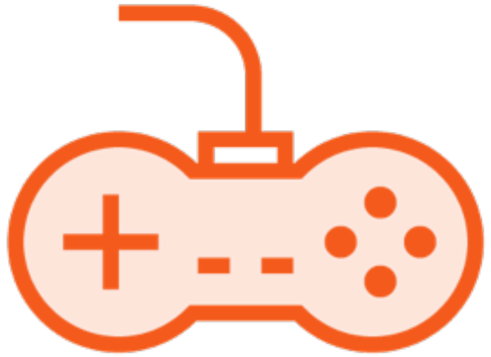
Assignment Problem

Objective Function

Minimize total cost

Constraints

1:1 mapping between tasks and persons



Decision Variables

Binary variables matching people and tasks

Decision Variables

$$x_{i,j} = \begin{cases} 1 & \text{if person i performs task j} \\ 0 & \text{otherwise} \end{cases}$$

i and j each vary from 1 to n

n^2 decision variables in total

Objective Function

Cost of person i performing task j is given by $c_{i,j}$

$$x_{i,j} = \begin{cases} 1 & \text{if person i performs task j} \\ 0 & \text{otherwise} \end{cases}$$

Total cost is sum-product of the decision variables and the costs

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_{i,j} x_{i,j}$$

Constraints

Each person will be assigned 1 task

$$\sum_{j=1}^{n} x_{i,j} = 1$$

(for $j = 1, 2 \ldots n$)

Each task will be performed by 1 person

$$\sum_{i=1}^{n} x_{i,j} = 1$$

(for $i = 1, 2 \ldots n$)

Assignment Problem Formulation

Minimize $$\sum_{i=1}^{n} \sum_{j=1}^{n} c_{i,j} x_{i,j}$$

Subject to constraints:

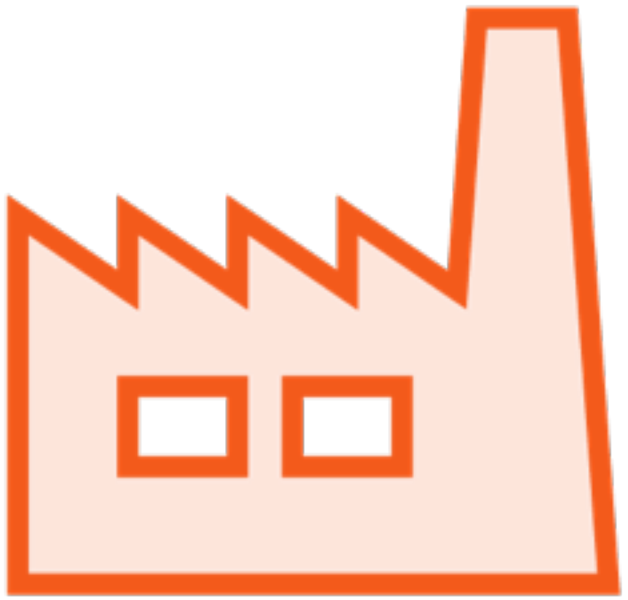
$$\sum_{j=1}^{n} x_{i,j} = 1$$

$$\sum_{i=1}^{n} x_{i,j} = 1$$

$x_{i,j}$ are binary

Integer programming can be used even in commonplace business situations

Factories and Warehouses



Factory Decisions

Suitable locations - City A and/or City B

Warehouse Decisions

Build 1 warehouse in city where factory is located

OK to build up to 2 factories, but exactly 1 warehouse

Factories and Warehouses

Decision Variable	Yes-or-no Decision	Total Lifetime Benefit	Upfront Investment
X_1	Build factory in City A?	$9 million	$6 million
X_2	Build factory in City B?	$5 million	$3 million
X_3	Build warehouse in City A?	$6 million	$5 million
X_4	Build warehouse in City B?	$4 million	$2 million

Capital available for upfront investment = $ 10 million

Factories and Warehouses

Decision Variable	Yes-or-no Decision	Total Lifetime Benefit	Upfront Investment
x_1	Build factory in City A?	$9 million	$6 million
x_2	Build factory in City B?	$5 million	$3 million
x_3	Build warehouse in City A?	$6 million	$5 million
x_4	Build warehouse in City B?	$4 million	$2 million

Capital available for upfront investment = $ 10 million

x_i are binary - each is a yes/no decision

Factories and Warehouses

Decision Variable	Yes-or-no Decision	Total Lifetime Benefit	Upfront Investment
x_1	Build factory in City A?	$9 million	$6 million
x_2	Build factory in City B?	$5 million	$3 million
x_3	Build warehouse in City A?	$6 million	$5 million
x_4	Build warehouse in City B?	$4 million	$2 million

Capital available for upfront investment = $ 10 million

$$\text{Maximize } Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$$

Factories and Warehouses

Decision Variable	Yes-or-no Decision	Total Lifetime Benefit	Upfront Investment
x_1	Build factory in City A?	$9 million	$6 million
x_2	Build factory in City B?	$5 million	$3 million
x_3	Build warehouse in City A?	$6 million	$5 million
x_4	Build warehouse in City B?	$4 million	$2 million

Capital available for upfront investment = $ 10 million

$$6x_1 + 3x_2 + 5x_3 + 2x_4 <= 10$$

Factories and Warehouses

Decision Variable	Yes-or-no Decision	Total Lifetime Benefit	Upfront Investment
x_1	Build factory in City A?	$9 million	$6 million
x_2	Build factory in City B?	$5 million	$3 million
x_3	Build warehouse in City A?	$6 million	$5 million
x_4	Build warehouse in City B?	$4 million	$2 million

Capital available for upfront investment = $ 10 million

If $x_1 = 1$, then x_3 can be 0 or 1

If $x_1 = 0$, then x_3 can only be 0

Factories and Warehouses

Decision Variable	Yes-or-no Decision	Total Lifetime Benefit	Upfront Investment
x_1	Build factory in City A?	\$9 million	\$6 million
x_2	Build factory in City B?	\$5 million	\$3 million
x_3	Build warehouse in City A?	\$6 million	\$5 million
x_4	Build warehouse in City B?	\$4 million	\$2 million

Capital available for upfront investment = \$ 10 million

$x_3 <= x_1$

$x_3 - x_1 <= 0$

Factories and Warehouses

Decision Variable	Yes-or-no Decision	Total Lifetime Benefit	Upfront Investment
x_1	Build factory in City A?	\$9 million	\$6 million
x_2	Build factory in City B?	\$5 million	\$3 million
x_3	Build warehouse in City A?	\$6 million	\$5 million
x_4	Build warehouse in City B?	\$4 million	\$2 million

Capital available for upfront investment = \$ 10 million

If $x_2 = 1$, then x_4 can be 0 or 1

If $x_2 = 0$, then x_4 can only be 0

Factories and Warehouses

Decision Variable	Yes-or-no Decision	Total Lifetime Benefit	Upfront Investment
x_1	Build factory in City A?	$9 million	$6 million
x_2	Build factory in City B?	$5 million	$3 million
x_3	Build warehouse in City A?	$6 million	$5 million
x_4	Build warehouse in City B?	$4 million	$2 million

Capital available for upfront investment = $ 10 million

$x_4 <= x_2$

$x_4 - x_2 <= 0$

Factories and Warehouses

Maximize

$$Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$$

Subject to constraints:

$$6x_1 + 3x_2 + 5x_3 + 2x_4 <= 10$$

$$x_3 - x_1 <= 0$$

$$x_4 - x_2 <= 0$$

x_1, x_2, x_3, x_4 are binary

Integer programming can be used even in commonplace business situations

Integer Programming: Unusual Formulations

Summary

Either-or constraints - pg 516

K-of-N constraints - pg 517

N possible values - pg 518

Startup costs - pg 519

Unusual Integer Programming Formulations

Either-or Constraints

Specific Allowable Values

Start-up Costs

Linear Programming Problem Formulation

Maximize

$$Z = 3x_1 + 5x_2$$

Subject to constraints:

$$x_1 <= 4$$

$$2x_2 <= 12$$

$$3x_1 + 2x_2 <= 18$$

$$x_1 , x_2 >= 0$$ (Non-negativity constraints)

Introducing Either-or Constraints

Maximize

$$Z = 3x_1 + 5x_2$$

Subject to constraints:

$$x_1 <= 4$$

$$2x_2 <= 12$$

A new type of constraint

$3x_1 + 2x_2 <= 18$ **OR** $x_1 + 4x_2 <= 16$

$x_1 , x_2 >= 0$ (Non-negativity constraints)

Either-or Constraints

$$3x_1 + 2x_2 <= 18 \quad \text{OR} \quad x_1 + 4x_2 <= 16$$

Define auxiliary binary variable z_1

Introduce a very large positive number M

Replace original constraint with two additional constraints

$$3x_1 + 2x_2 <= 18 + Mz_1$$

$$\text{AND} \quad x_1 + 4x_2 <= 16 + M(1-z_1)$$

Exactly one of the two constraints will be satisfied

Constraints

Adding M is equivalent to creating slack in a particular constraint

Doing so makes that constraint non-binding

Constraints

M should also be large enough that no feasible solution is eliminated by it

The feasible solution set must be bounded

Either-or Constraints

$z_1 = 1, 1-z_1 = 0$

$3x_1 + 2x_2 <= 18 + Mz_1$

$x_1 + 4x_2 <= 16 + M(1-z_1)$

(non-binding)

$\cancel{3x_1 + 2x_2 <= 18 + M}$

$x_1 + 4x_2 <= 16 + 0$

(binding)

$x_1 + 4x_2 <= 16$

$z_1 = 0, 1-z_1 = 1$

$3x_1 + 2x_2 <= 18 + Mz_1$

$x_1 + 4x_2 <= 16 + M(1-z_1)$

(non-binding)

$\cancel{x_1 + 4x_2 <= 16 + M}$

$3x_1 + 2x_2 <= 18 + 0$

(binding)

$3x_1 + 2x_2 <= 18$

Constraints

Either-or constraints are 1-of-2 constraints

Can extend to K-of-N constraints by introducing K binary auxiliaries

Unusual Integer Programming Formulations

Either-or Constraints

Specific Allowable Values

Start-up Costs

Linear Programming Problem Formulation

Maximize

$$Z = 3x_1 + 5x_2$$

Subject to constraints:

$$x_1 <= 4$$

$$2x_2 <= 12$$

$$3x_1 + 2x_2 <= 18$$

$x_1 , x_2 >= 0$ (Non-negativity constraints)

Constraints

Management decides to set aside slots for new product trials

The trials will neither increase profit nor costs for now

Each trial requires 6 hours of time in factory y_3

Linear Programming Problem Formulation

Maximize

$$Z = 3x_1 + 5x_2$$

Subject to constraints:

$$x_1 <= 4$$

$$2x_2 <= 12$$

$$\boxed{3x_1 + 2x_2 <= 18}$$

$$x_1 , x_2 >= 0$$ (Non-negativity constraints)

Linear Programming Problem Formulation

Maximize

$$Z = 3x_1 + 5x_2$$

Subject to constraints:

$$x_1 <= 4$$

$$2x_2 <= 12$$

$$\boxed{3x_1 + 2x_2 <= 18}$$

$$x_1 , x_2 >= 0$$ (Non-negativity constraints)

Linear Programming Problem Formulation

Maximize

$$Z = 3x_1 + 5x_2$$

Subject to constraints:

$$x_1 <= 4$$

$$2x_2 <= 12$$

A new type of constraint

$3x_1 + 2x_2 = 6$ OR $3x_1 + 2x_2 = 12$ OR $3x_1 + 2x_2 = 18$

$x_1 , x_2 >= 0$ (Non-negativity constraints)

An Integer Constraint

$3x_1 + 2x_2 = 6$ OR $3x_1 + 2x_2 = 12$ OR $3x_1 + 2x_2 = 18$

Define auxiliary variables z_1, z_2, z_3

Redefine original constraint as $3x_1 + 2x_2 = 6z_1 + 12z_2 + 18z_3$

Add constraints on the auxiliary variables

$z_1 + z_2 + z_3 = 1$

z_1, z_2, z_3 are binary

i.e. $z_1, z_2, z_3 \in \{0,1\}$

An Integer Constraint

$z_1 + z_2 + z_3 = 1$

z_1, z_2, z_3 are binary

i.e. $z_1, z_2, z_3 \in \{0,1\}$

$$3x_1 + 2x_2 = 6z_1 + 12z_2 + 18z_3$$

$z_1 = 1$ OR $z_2 = 1$ OR $z_3 = 1$

$3x_1 + 2x_2 = 6$

$3x_1 + 2x_2 = 12$

$3x_1 + 2x_2 = 18$

An Integer Constraint

$z_1 + z_2 + z_3 = 1$

z_1, z_2, z_3 are binary

i.e. $z_1, z_2, z_3 \in \{0,1\}$

$3x_1 + 2x_2 = 6z_1 + 12z_2 + 18z_3$

$z_1 = 1$ OR $z_2 = 1$ OR $z_3 = 1$

$3x_1 + 2x_2 = 6$ OR $3x_1 + 2x_2 = 12$ OR $3x_1 + 2x_2 = 18$

Linear Programming Problem Formulation

Maximize

$Z = 3x_1 + 5x_2$

Subject to constraints:

$x_1 <= 4$

$2x_2 <= 12$

A new type of constraint

$3x_1 + 2x_2 = 6$ OR $3x_1 + 2x_2 = 12$ OR $3x_1 + 2x_2 = 18$

$x_1 , x_2 >= 0$ (Non-negativity constraints)

Integer Programming Problem Formulation

Maximize

$$Z = 3x_1 + 5x_2$$

Subject to constraints:

$$x_1 <= 4$$

$$2x_2 <= 12$$

$$3x_1 + 2x_2 = 6z_1 + 12z_2 + 18z_3$$

$$z_1 + z_2 + z_3 = 1$$

$z_1, z_2, z_3 \in \{0,1\}$ **(Binary integer constraint)**

$x_1 , x_2 >= 0$ **(Non-negativity constraints)**

Unusual Integer Programming Formulations

Either-or Constraints

Specific Allowable Values

Start-up Costs

Micro-economic Assumptions

Proportionality Assumption

No start-up costs, constant returns to scale

Additivity Assumption

Products are neither complements nor substitutes

Divisibility Assumption

Fractional production is possible

Cost Minimization

Minimize

$$W = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

Subject to constraints:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n <= b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n <= b_2$$

$$\vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n <= b_m$$

$x_1 , x_2 \dots x_n >= 0$ (Non-negativity constraints)

Introducing Start-up Costs

Minimize

$$W = f(x_1) + f(x_2) + \dots + f(x_n)$$

Subject to constraints:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n <= b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n <= b_2$$

$$\vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n <= b_m$$

$x_1 , x_2 \dots x_n >= 0$ (Non-negativity constraints)

Start-up Costs

$f(x_1), f(x_2)...f(x_m)$ include the start-up costs of each activity

$$f(x_i) = \begin{cases} k_i + c_i x_i & \text{if } x_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

Actual start-up costs are $k_1, k_2, k_3 \dots k_m$

Start-up Costs

Define m auxiliary variables $y_1, y_2, y_3 \ldots y_m$

$$y_i = \begin{cases} 1 & \text{if } x_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

Redefine the objective function multiplying k_i by y_i

Minimize

$$W = \sum (k_i\, y_i + c_i\, x_i)$$

Start-up Costs

Add constraints

$$x_i <= My_i$$

Where M is a very large positive number (e.g. M = 100 Billion)

y_i : Answer to the question "Should activity i be started up?"

x_i : Total units of activity i to be undertaken

The constraint will ensure that

If $x_i > 0$, then $y_i = 1$

If undertaking activity x_i, then start-up costs will be incurred

Cost Minimization with Start-up Costs

Minimize

$$W = \sum (k_i\, y_i + c_i\, x_i)$$

Subject to additional constraints:

$$x_i <= M y_i$$

y_i are binary

Note:

The objective function must be minimization, else y_i could be 1 when x_i is 0

Unusual Integer Programming Formulations

Either-or Constraints

Specific Allowable Values

Start-up Costs

Summary

- Variants of simplex (m2)
- Upper bound variables - pg 311
- Dual Simplex - pg 303
- Interior Point Algorithms - pg 313

Problems

- Transportation problem - pg 336 - 338
- Assignment problem - pg 361 - 363

Network Optimisation Problems

- Intro - 395
- Shortest Path - pg 398
- MST - pg 403
- Maximal Flow - pg 407

Non-linear Programming

- Wyndor - pg 596
- Unconstrained - pg 601

Summary

Integer programming problems stipulate that decision variables be integers

Integer problems are even more widely used in business than LPPs

Solving some integer problems can be very mathematically complex

The LP-relaxation of an integer problem is the LPP that drops the integer constraint

LP-relaxations, if used right, greatly simplify solving integer problems