Lecture 2

INTRO TO OPTIMIZATION

Warmup!

mathematical optimization problem

$$\begin{aligned} \text{minimize:} \quad & f(\mathbf{x}) \\ \text{subject to:} \quad & g_i(\mathbf{x}) \leq b_i, \; i = 1, \ldots, m \end{aligned}$$

$\mathbf{x} = (x_1, \ldots, x_n)$: optimization variables

$f : \mathcal{R}^n \to \mathcal{R}$: objective function

$g_i : \mathcal{R}^n \to \mathcal{R}, \; i = 1, \ldots, m$: constraint functions

Boolean satisfiability problem (SAT)

- Boolean Satisfiability or simply **SAT** is one of the most famous and most studied in theoretical computer science
- SAT is the problem of determining if a **Boolean formula** is satisfiable or unsatisfiable

SAT applications

- Zhang, H. **Generating college conference basketball schedules by a SAT solver**. In *Proceedings of the 5th International Symposium on Theory and Applications of Satisfiability Testing.* (Cincinnati, OH, 2002).

- Nam, G.-J., Sakallah, K. A., and Rutenbar, R.A. Satisfiability-based layout revisited: **Detailed routing of complex FPGAs via search-based Boolean SAT**. *International Symposium on Field-Programmable Gate Arrays* (Monterey, CA, 1999).

- Kautz, H. and Selman, B. **Planning as satisfiability**. *European Conference on Artificial Intelligence*, 1992.

Boolean formula

1. Variables

   e.g., x_1, x_2, x_3 that take only two possible values: true (1) or false (0)

2. Operator "not" written as $\neg$

   if x_1= true, then $\neg x_1$= false (equivalently, $x_1 = 1 \rightarrow \neg x_1 = 0$)

3. Operator "and" written as $\wedge$

   a *conjunction* of x_1 and x_2: $x_1 \wedge x_2$

4. Operator "or" written as $\vee$

   a *disjunction* of x_1 and x_2: $x_1 \vee x_2$

$$\boxed{\begin{array}{l} 0 \wedge 0 = 0 \\ 0 \wedge 1 = 0 \\ 1 \wedge 0 = 0 \\ 1 \wedge 1 = 1 \end{array}}$$

$$\boxed{\begin{array}{l} 0 \vee 0 = 0 \\ 0 \vee 1 = 1 \\ 1 \vee 0 = 1 \\ 1 \vee 1 = 1 \end{array}}$$

Example Boolean formula: $\neg x_1 \vee (x_2 \vee x_3)$

SAT problem

- Input: a Boolean formula with n variables
- Problem: can you set the variables to a combination of true/false so that the formula becomes true?

$$(x_1 \land x_2) \lor (x_1 \land \neg x_1) \lor x_2$$

$$(x_1 \lor x_3) \land (x_1 \land (x_2 \lor \neg x_3)) \land (\neg x_2 \lor \neg x_3) \land (\neg x_1 \lor \neg x_2)$$

simple SAT problem

$$f(\mathbf{x}) = (x_1 \vee x_3) \wedge (x_1 \wedge (x_2 \vee \neg x_3)) \wedge (\neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2)$$

x_1	x_2	x_3	$x_1 \vee x_3$	$x_1 \wedge (x_2 \vee \neg x_3)$	$\neg x_2 \vee \neg x_3$	$\neg x_1 \vee \neg x_2$	$f(\mathbf{x})$
TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE
TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE
TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
	…						

Bonus material: Conjuntive Normal Form (CNF)

A statement is in conjunctive normal form if it is a conjunction of clauses, where a clause is a disjunction of literals (i.e., statement letters and negations of statement letters) → ANDs of ORs

You can convert any Boolean formula to CNF.

(However, sometimes this conversion leads to exponential growth → the size of the CNF formula may be much larger than the size of the original formula)

Example conjunctive normal form:

ANDs

$$\underbrace{(x_1 \vee x_3)} \wedge \underbrace{(x_1 \vee \neg x_2 \vee x_3)} \wedge \underbrace{(\neg x_2 \vee \neg x_3)} \wedge \underbrace{(\neg x_1 \vee \neg x_2)}$$

ORs

Bonus material: k-SAT

If a boolean formula is in CNF and every clause consists of exactly k literals, we say the boolean formula is an instance of k-SAT

Example 3-CNF formula:

$$(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_4) \wedge (x_3 \vee \neg x_6 \vee \neg x_7) \wedge (\neg x_1 \vee x_8 \vee \neg x_9)$$

Super bonus material!

2-SAT is in **P**

3-SAT is **NP-complete**

A final word on SAT

- The notation that I showed you is only one example…
- E.g., multiplication and addition are commonly used to express “and” and “or”, respectively

$$(x_1 \lor x_2 \lor x_3) \land (x_1 \lor \neg x_2 \lor x_4) \land (x_3 \lor \neg x_6 \lor \neg x_7) \land (\neg x_1 \lor x_8 \lor \neg x_9)$$

$$=$$

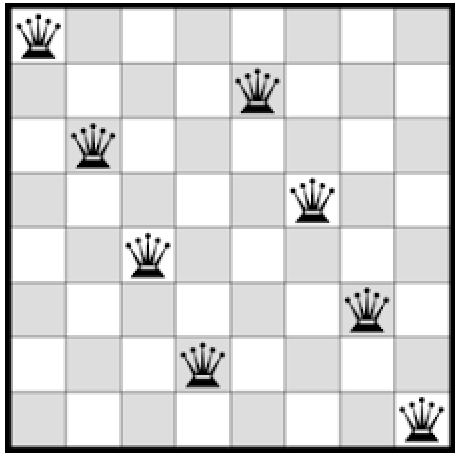
$$(x_1 + x_2 + x_3)(x_1 + \overline{x_2} + x_4)(x_3 + \overline{x_6} + \overline{x_7})(\overline{x_1} + x_8 + \overline{x_9})$$

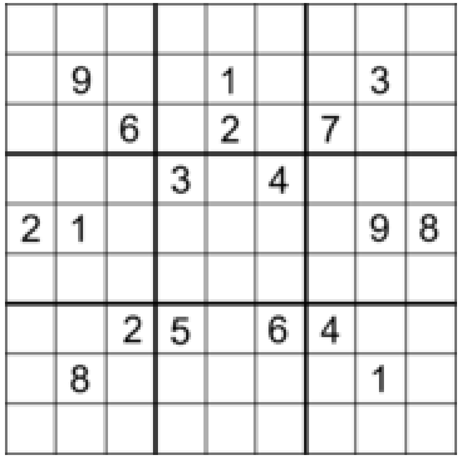
Constraint Satisfaction Problems (CSP)

- CSP is a generalization of the SAT
- Defined by a set of variables, $x_1, x_2, \ldots, x_n$, and constraints, $C_1, C_2, \ldots, C_m$
- Each variable x_i has nonempty domain D_i of possible values
- Each constraint C_i involves some subset of the variables and specifies the allowable combinations of values for that subset
- A state of the problem is defined by an assignment of values to some or all of the variables, $\{x_i = v_i, x_j = v_j, \ldots\}$
- A complete assignment is one in which every variable is mentioned
- A solution is a complete assignment that satisfies all the constraints

Example CSPs

- 8-queens problem
- Map-coloring
- Sudoku

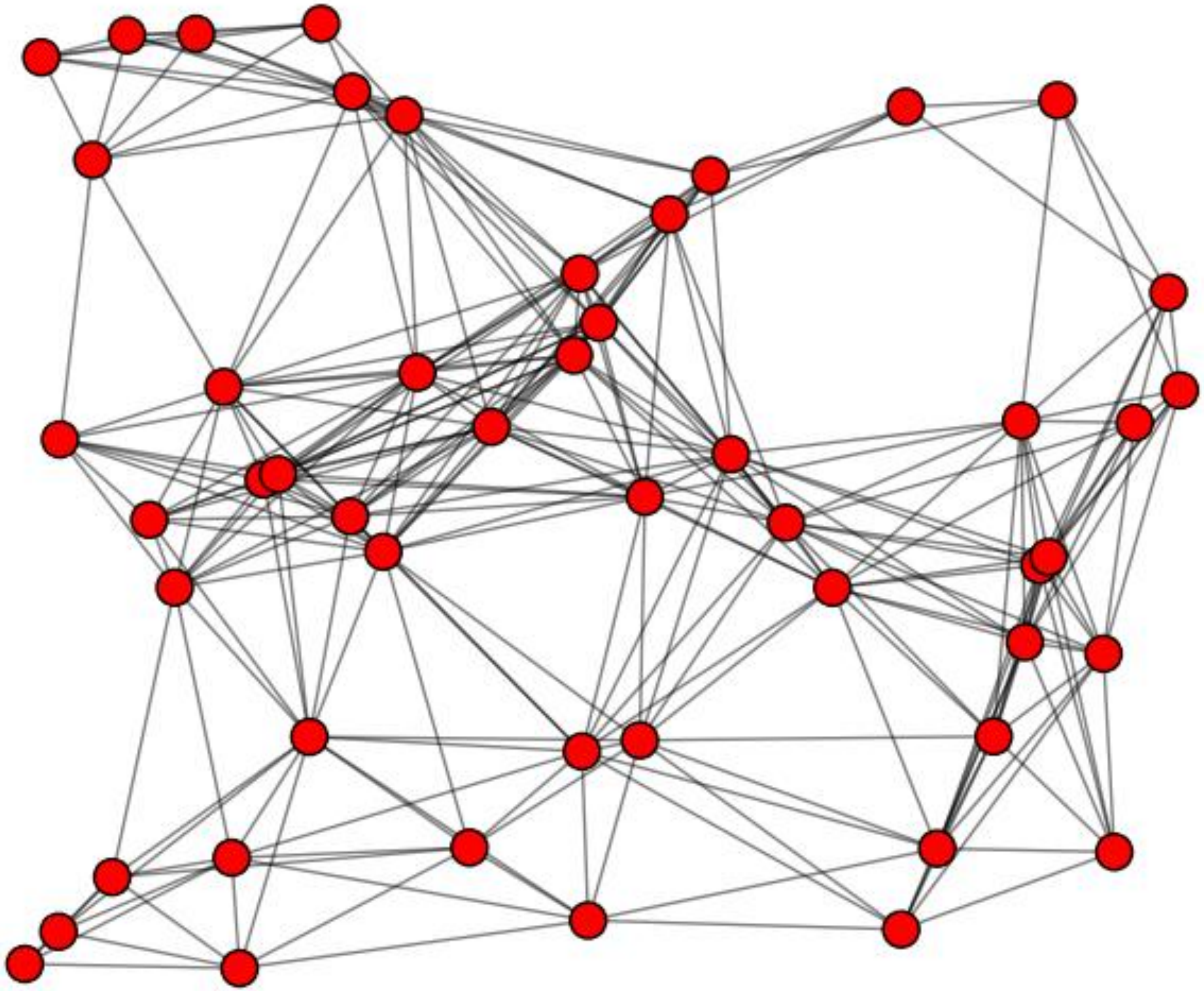


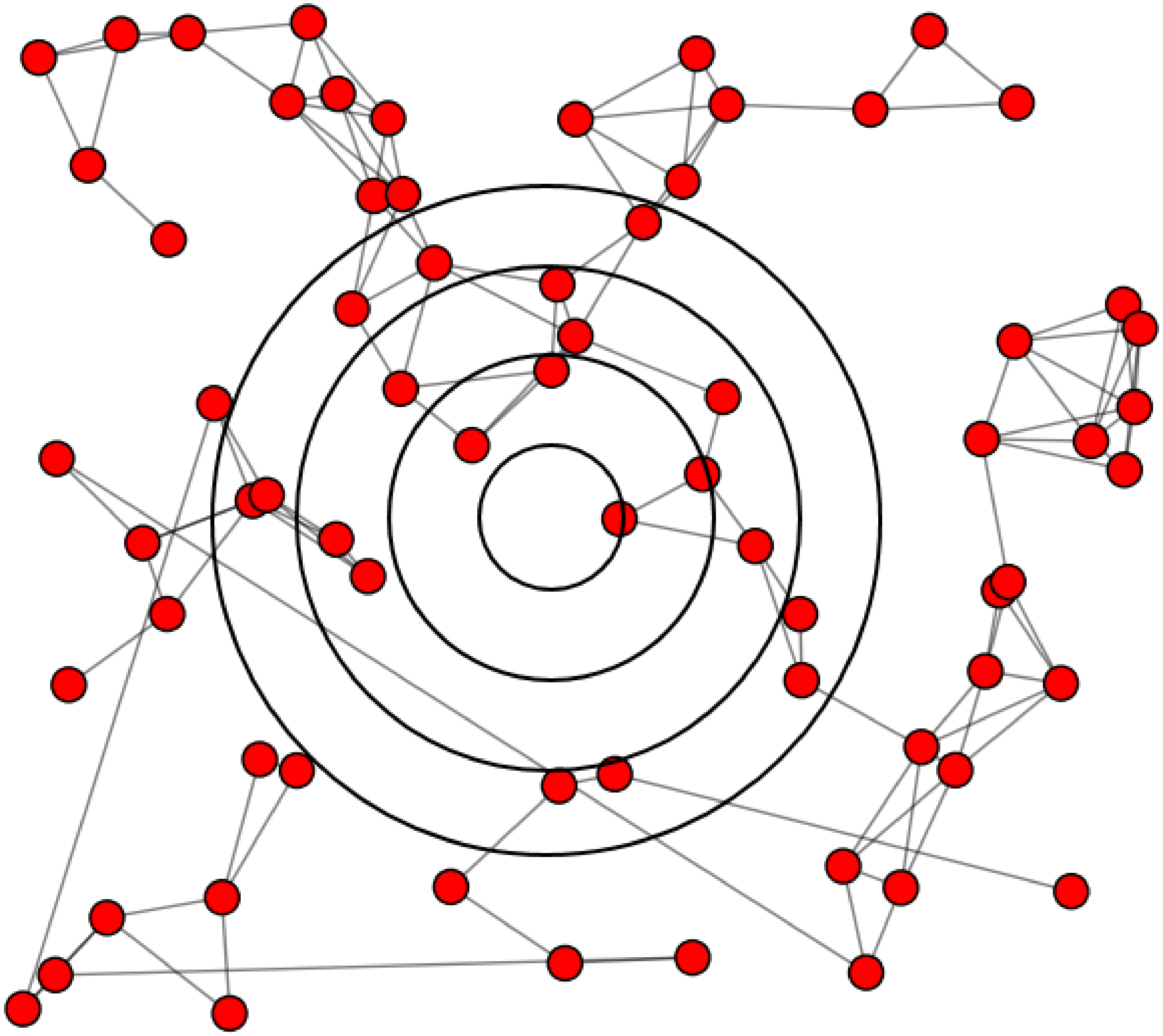


- Note: Linear Programming is a special case of CSPs with continuous variable domains, constraints are linear inequalities forming convex feasible region, and includes a linear objective function

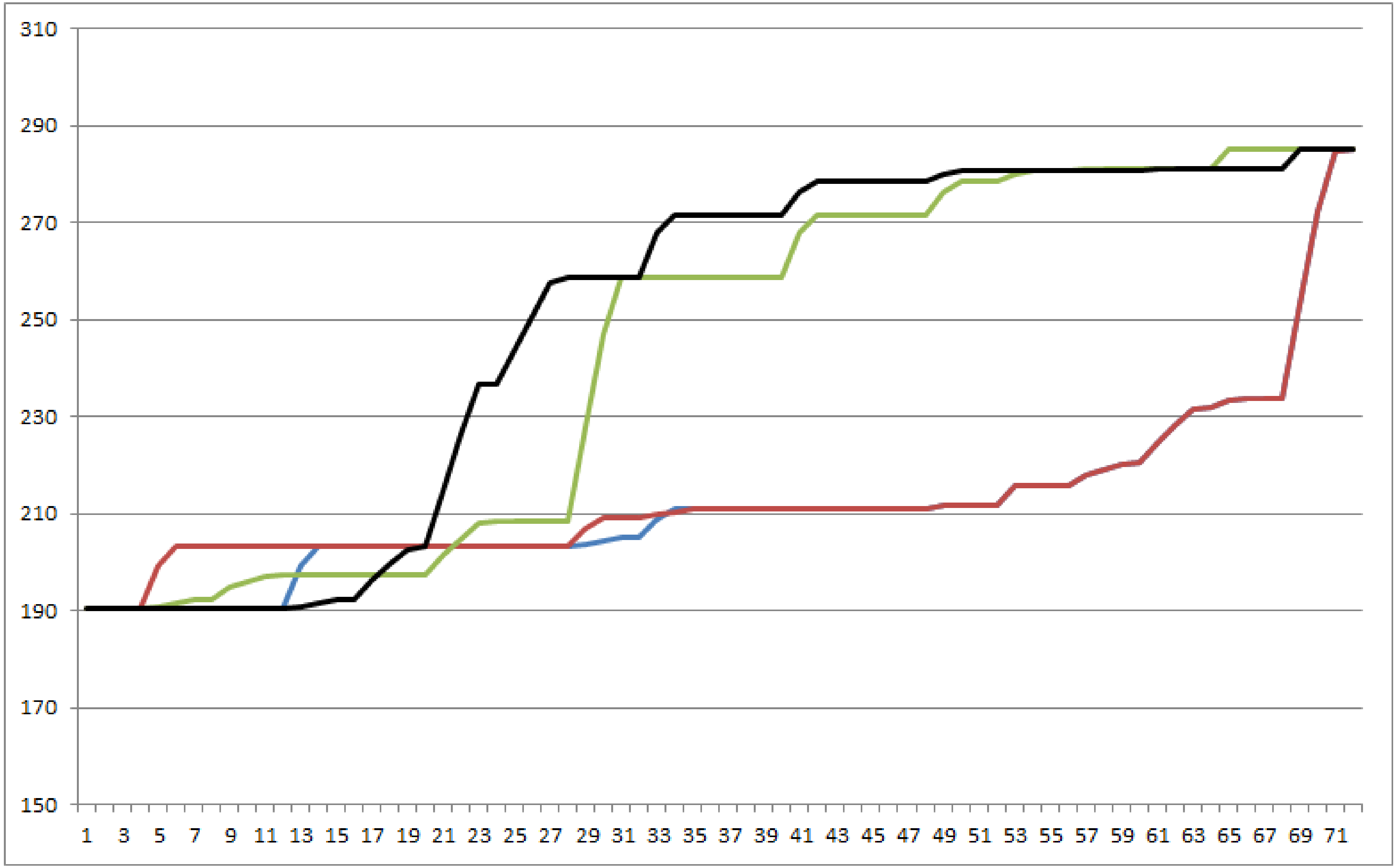
Satisfying Constraints: Course Timetabling

Assign each lecture to some period of the week in such a way that no student is required to take more than one lecture at a time.





Recovery profile based on sequencing



Visualize, assume, simplify, formulate

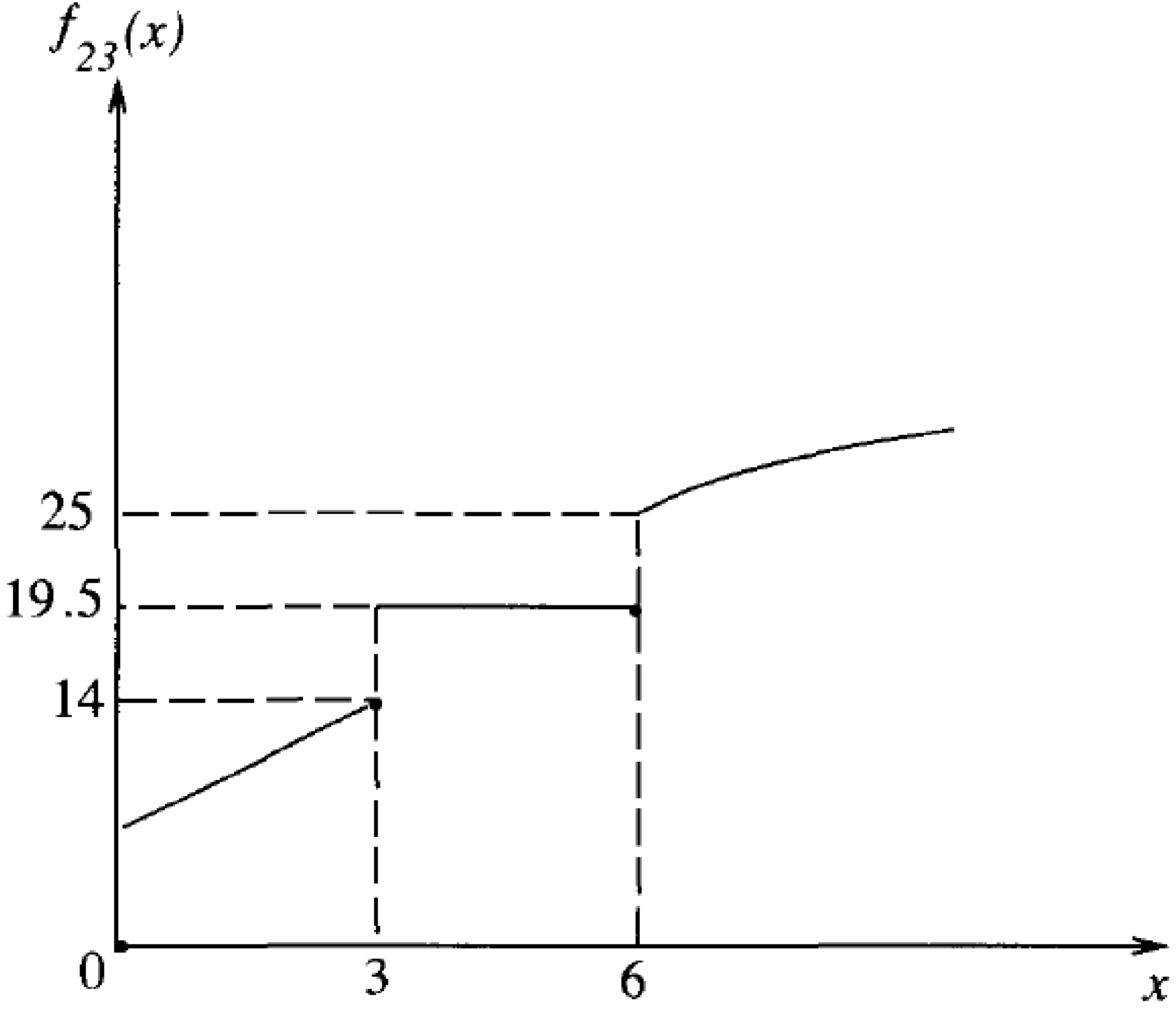
Problem → Model → Solution

Decide on approach, solve

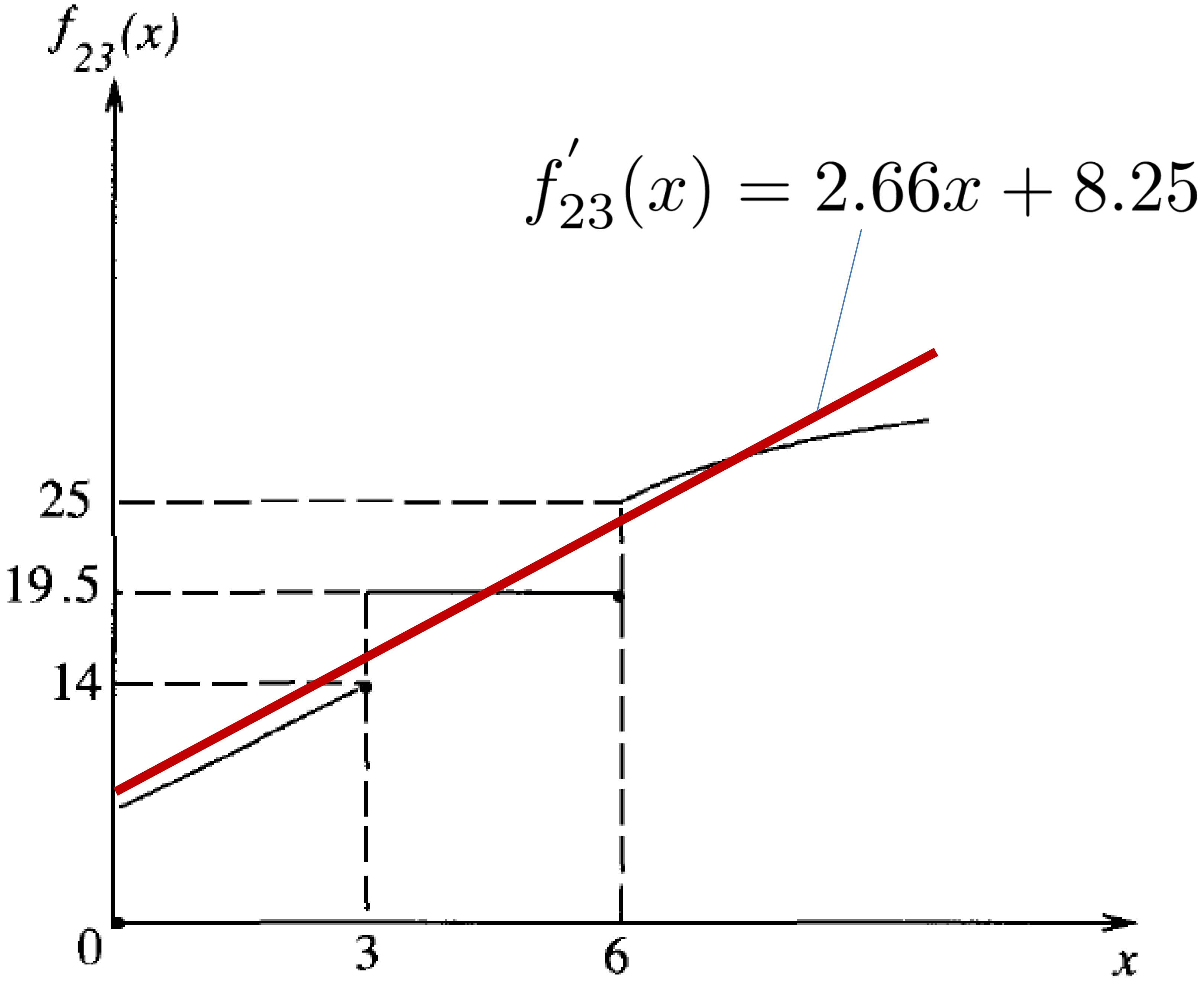
Suppose a company has n warehouses that store paper supplies in reams. These supplies are to be delivered to k distribution centers. Every possible delivery route between a warehouse i and a distribution center j has an associated transportation cost, f_{ij}.

Example from HTSI chapter 1, page 17

$$f_{23}(x) = \begin{cases} 0 & \text{if } x = 0 \\ 4 + 3.33x & \text{if } 0 < x \leq 3 \\ 19.5 & \text{if } 3 < x \leq 6 \\ 0.5 + 10\sqrt{x} & \text{if } x > 6 \end{cases}$$



$f_{23}(x)$
25
19.5
14
0
3
6
x



$f_{23}(x)$
$f'_{23}(x) = 2.66x + 8.25$
25
19.5
14
0
3
6
x

Problem → Model (approximate) → Solution (precise)

or

Problem → Model (precise) → Solution (approximate)

portfolio optimization

- variables: amounts invested in different assets
- constraints: budget, max/min investment per asset, min return
- objective: overall risk or return variance

device sizing in electronic circuits

- variables: device widths and lengths
- constraints: manufacturing limits, timing requirements, max area
- objective: power consumption

data fitting

- variables: model parameters
- constraints: prior information, parameter limits
- objective: measure of misfit or prediction error

general optimization problem

- very difficult to solve
- methods involve some compromise, e.g.
- very long computation times
- not finding optimum, only approximate solutions

exceptions: certain problem classes can be solved efficiently and reliably, e.g.

- least-square problems
- linear programming problems

least-squares linear regression

$$\text{minimize } ||Y - \mathbf{X}\beta||_2^2$$

$$\text{minimize } \sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$$

linear programming (LP) problem

$$\begin{aligned} \text{maximize } & \mathbf{c}^T\mathbf{x} \\ \text{subject to } & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq 0 \end{aligned}$$

solving linear programs

- no analytical formula for solution
- reliable, efficient algorithms and software
- mature technology

Slide from Stephen Boyd, Convex Optimization I, Stanford

$$\begin{array}{ll} \text{Objective} & \max \quad c_1x_1 + c_2x_2 + \cdots + c_nx_n \\ \text{Constraints} & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1 \\ & a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2 \\ & \ldots \\ & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_m \\ & x_1 \geq 0 \\ & x_2 \geq 0 \\ & \ldots \\ & x_n \geq 0 \end{array}$$

$\mathbf{c}^T\mathbf{x}$

ve max $c_1x_1 + c_2x_2 + \cdots + c_nx_n$

Constraints

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\le b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\le b_2 \\
\cdots& \\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\le b_m
\end{aligned}$$

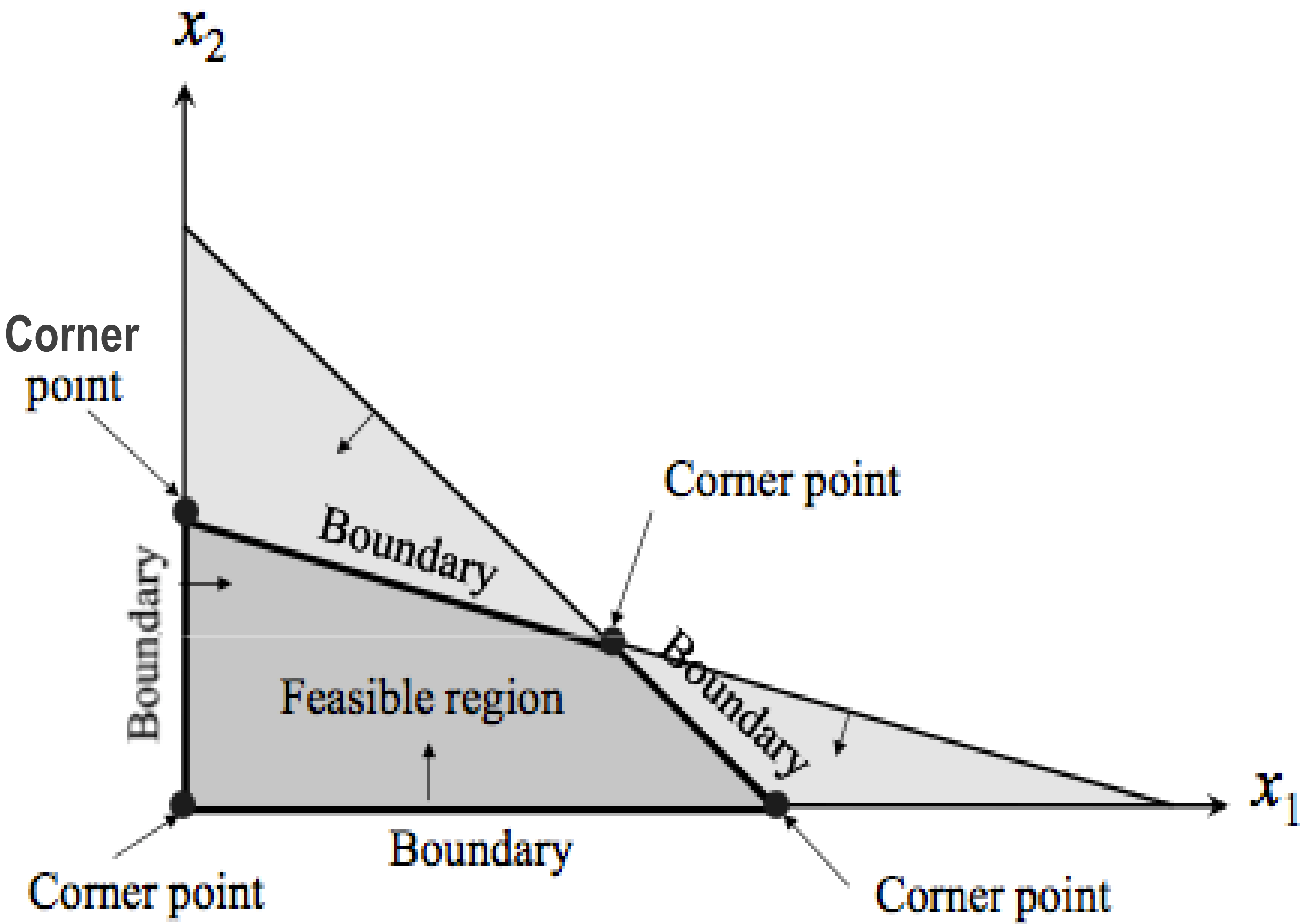
$\mathbf{Ax}$

$\mathbf{b}$

$$\begin{aligned}
x_1 &\ge 0 \\
x_2 &\ge 0 \\
\cdots& \\
x_n &\ge 0
\end{aligned}$$

$\mathbf{x} \ge 0$

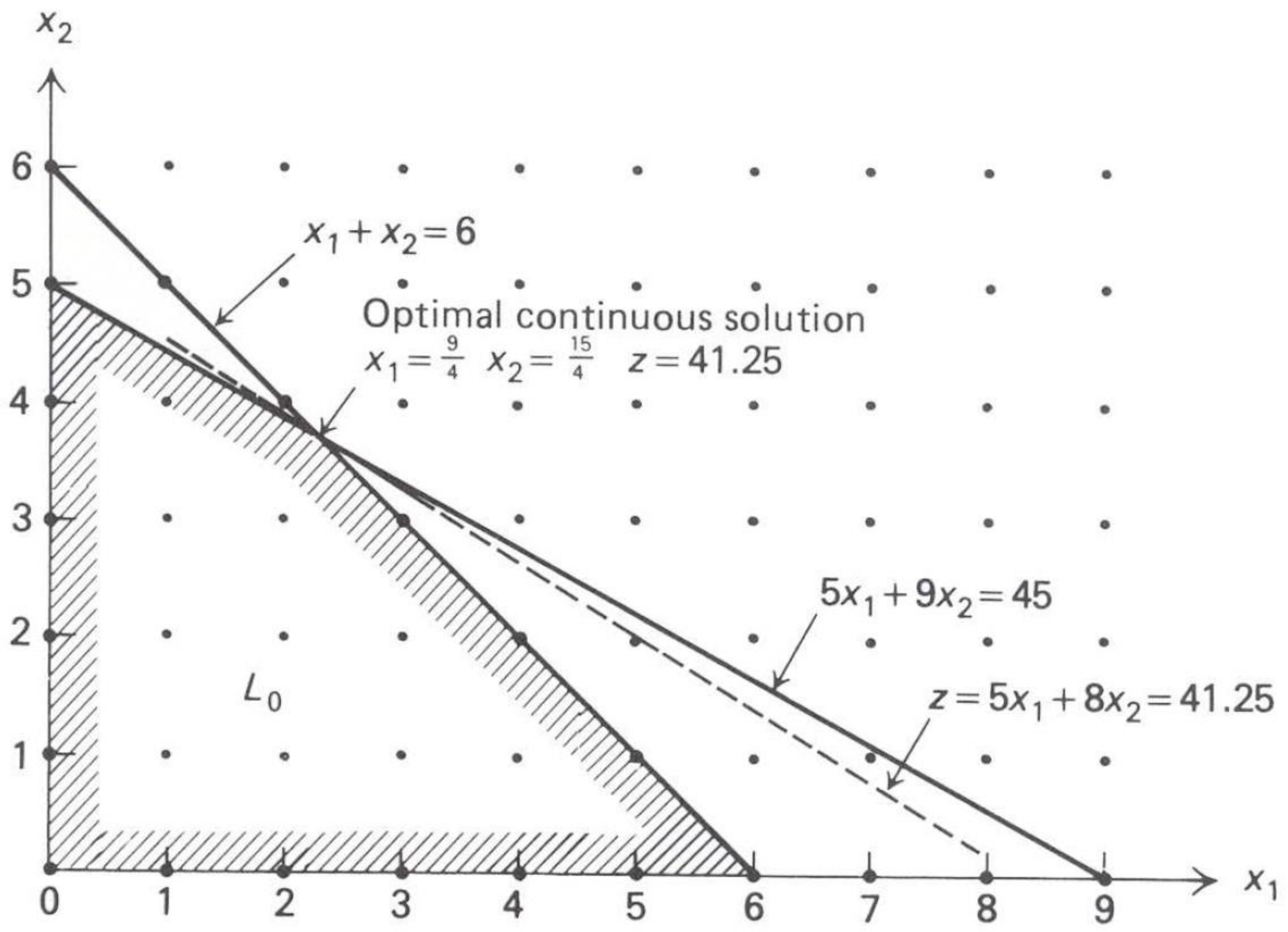
Graphical view of an LP problem



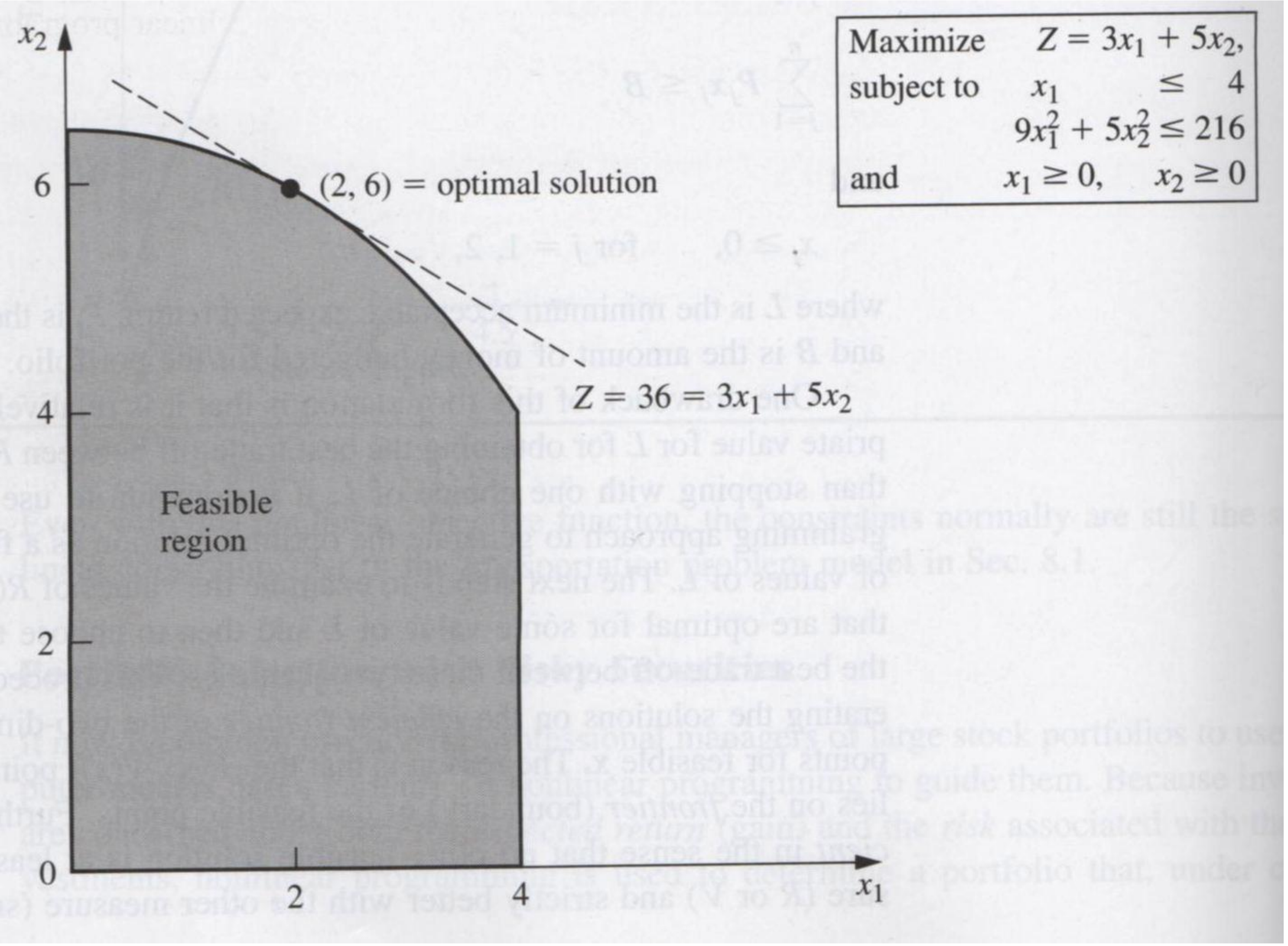
integer programming (IP) problem

$$
\begin{aligned}
\text{maximize } & \mathbf{c}^T\mathbf{x} \\
\text{subject to } & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\
& \mathbf{x} \geq 0 \\
& \mathbf{x} \in \mathcal{Z}^n
\end{aligned}
$$

Graphical illustration of integer program problem

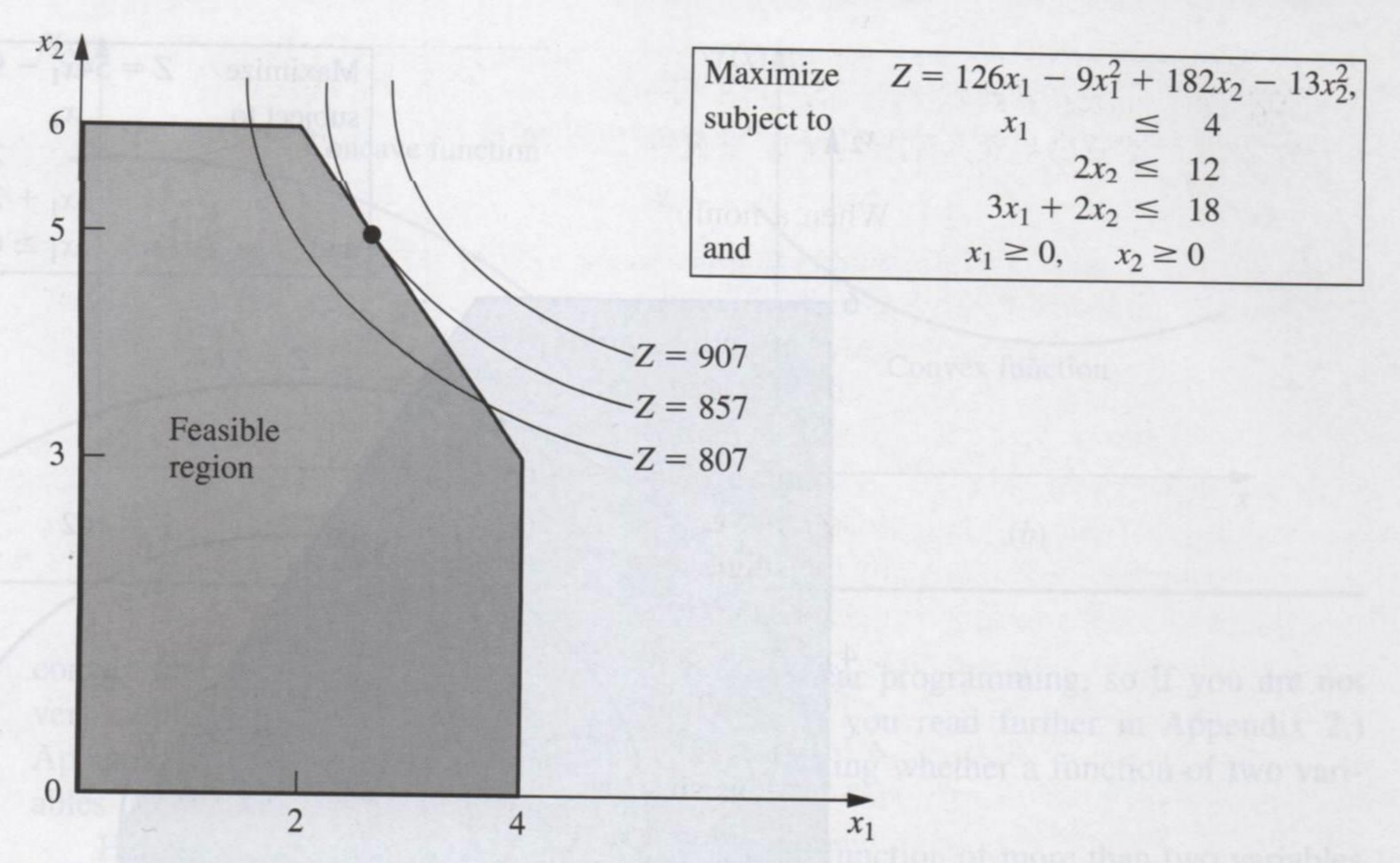


Graphical illustration of nonlinear programs



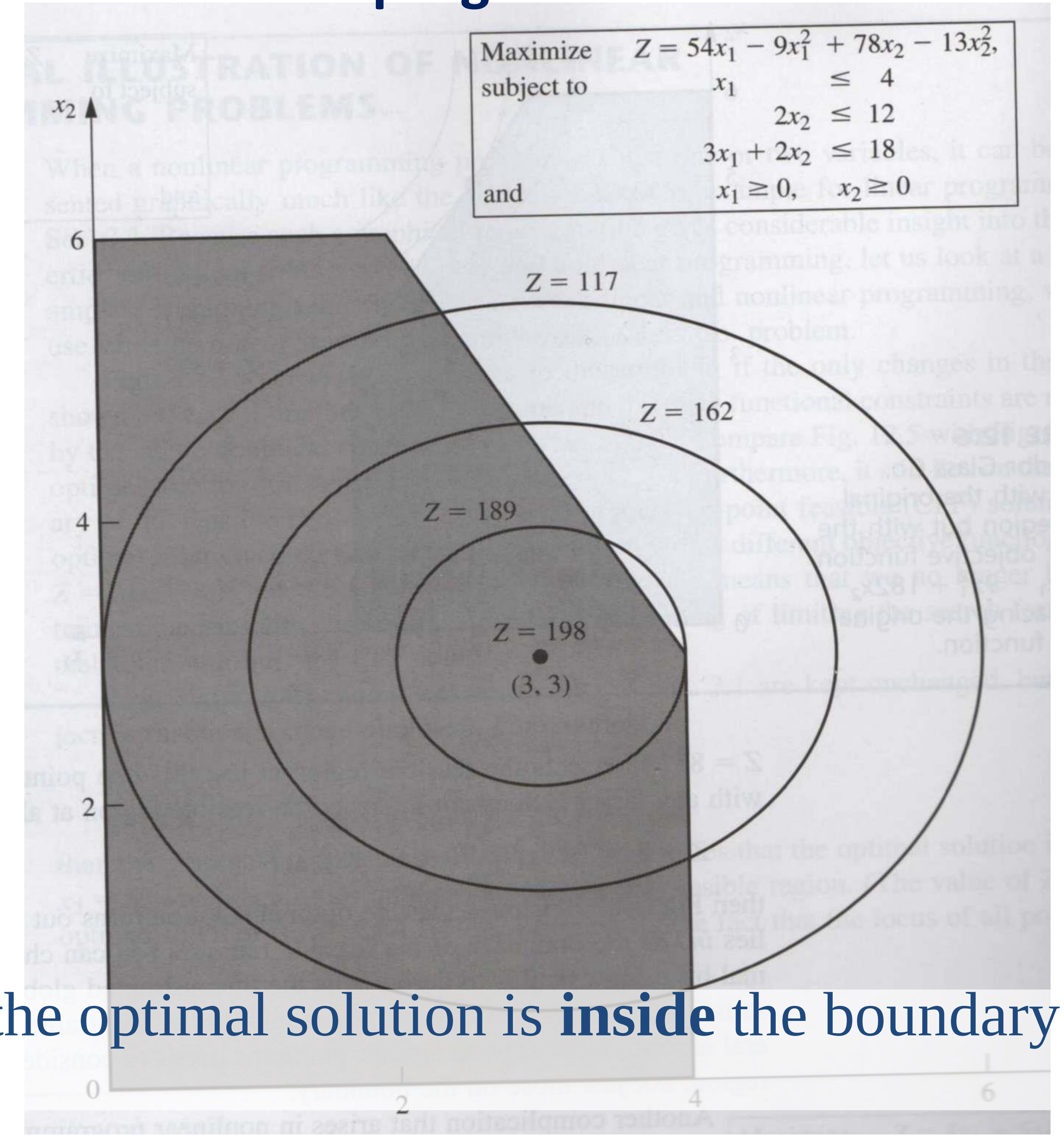
An example with nonlinear constraints when the optimal solution is **not** a corner point feasible solution.

Graphical illustration of nonlinear programs



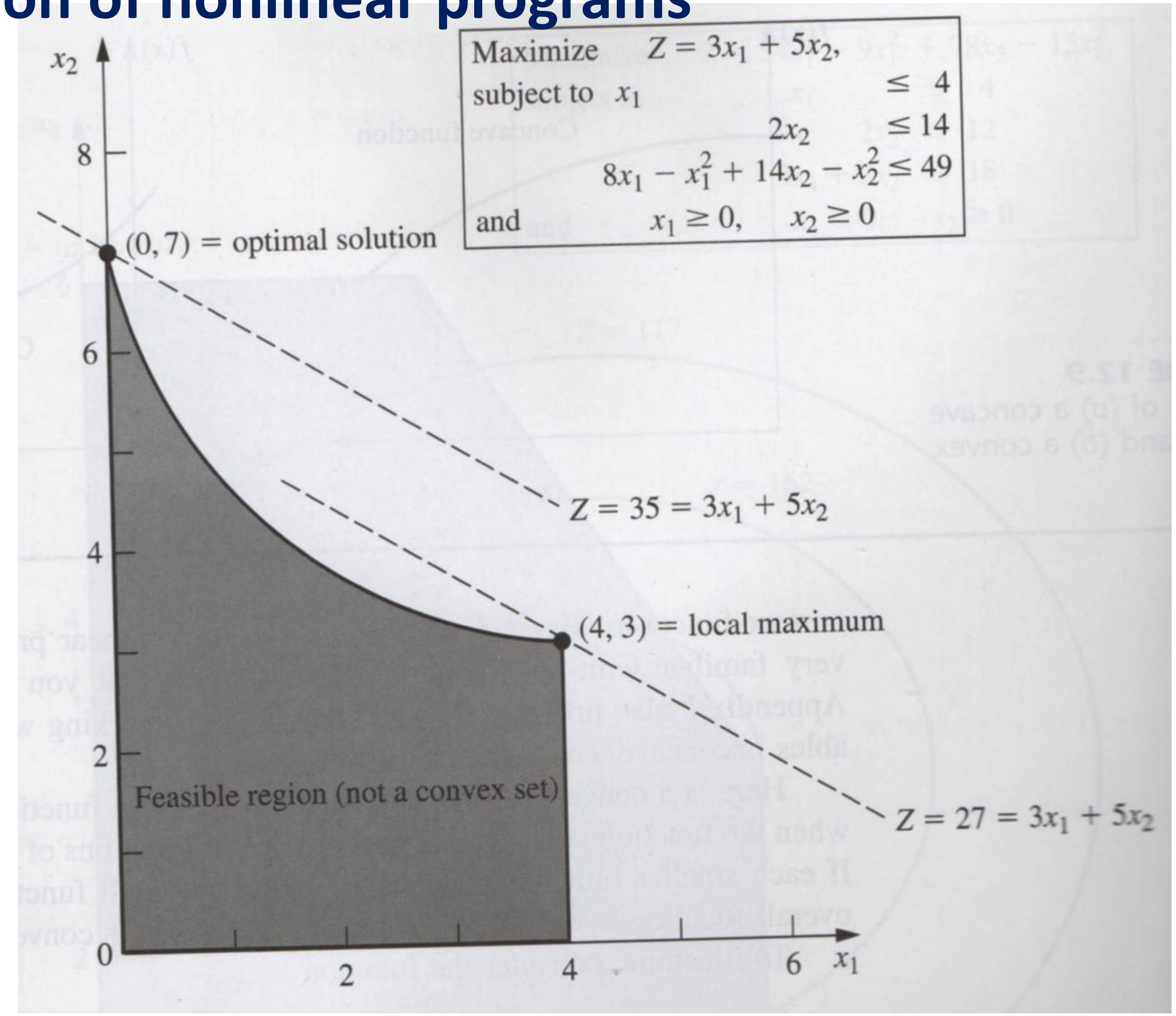
An example with linear constraints but nonlinear objective function when the optimal solution is **not** a corner point feasible solution.

Graphical illustration of nonlinear programs



An example when the optimal solution is **inside** the boundary of the feasible region.

Graphical illustration of nonlinear programs



An example when a local maximum is **not** a global maximum (the feasible region is **not** a convex set).