CS:4980 Topics in Computer Science II

Introduction to Automated Reasoning

Theory Solvers II

Cesare Tinelli

Spring 2024

The University of Iowa

Credits

These slides are based on slides originally developed by **Cesare Tinelli** at the University of Iowa, and by **Clark Barrett**, **Caroline Trippel**, and **Andrew (Haoze) Wu** at Stanford University. Adapted by permission.

Overview

SMT solvers can be used to solve arithmetic problems

Linear Programs (LPs) are a particularly interesting class of arithmetic problems, with stand-alone solvers

Many interesting applications: robotic planning, formal verification, operations research

Some of the slides are contributed by Guy Katz.

Outline

- QF_LRA
- Linear Programming
- The Simplex algorithm

Readings: DP 5.1-5.2

Review: Theory of Real Arithmetics ($\mathcal{T}_{\mathsf{RA}}$)

$\mathcal{T}_{\mathsf{RA}} = \langle \Sigma_{\mathsf{RA}}, \boldsymbol{M}_{\mathsf{RA}} \rangle$

$$\Sigma^S_{\mathsf{RA}} = \{ \mathsf{Real} \} \qquad \Sigma^F_{\mathsf{RA}} = \{ +, -, *, \leq \} \cup \{ q \mid q \text{ is a decimal numeral} \}$$

All $\mathcal{I} \in \boldsymbol{M}_{\mathsf{RA}}$ interpret Real as the set $\mathbb{R}$ of real numbers, and the function symbols in the usual way

Quantifier-free linear real arithmetic (QF_LRA):

1. no quantifiers
2. all occurrences of $*$ have at least one argument that is a decimal numeral

Many SMT solvers (e.g., cvc5, Z3) use a version of the Simplex method as the theory solver for QF_LRA

Review: Theory of Real Arithmetics ($\mathcal{T}_{\mathsf{RA}}$)

$\mathcal{T}_{\mathsf{RA}} = \langle \Sigma_{\mathsf{RA}}, \boldsymbol{M}_{\mathsf{RA}} \rangle$

$$\Sigma^S_{\mathsf{RA}} = \{ \mathsf{Real} \} \qquad \Sigma^F_{\mathsf{RA}} = \{ +, -, *, \leq \} \cup \{ q \mid q \text{ is a decimal numeral} \}$$

All $\mathcal{I} \in \boldsymbol{M}_{\mathsf{RA}}$ interpret Real as the set $\mathbb{R}$ of real numbers, and the function symbols in the usual way

Quantifier-free linear real arithmetic (QF_LRA):

1. no quantifiers
2. all occurrences of $*$ have at least one argument that is a decimal numeral

Many SMT solvers (e.g., cvc5, Z3) use a version of the Simplex method as the theory solver for QF_LRA

Review: Theory of Real Arithmetics ($\mathcal{T}_{\mathsf{RA}}$)

$\mathcal{T}_{\mathsf{RA}} = \langle \Sigma_{\mathsf{RA}}, \boldsymbol{M}_{\mathsf{RA}} \rangle$

$$\Sigma^S_{\mathsf{RA}} = \{ \mathsf{Real} \} \qquad \Sigma^F_{\mathsf{RA}} = \{ +, -, *, \leq \} \cup \{ q \mid q \text{ is a decimal numeral} \}$$

All $\mathcal{I} \in \boldsymbol{M}_{\mathsf{RA}}$ interpret Real as the set $\mathbb{R}$ of real numbers, and the function symbols in the usual way

Quantifier-free linear real arithmetic (QF_LRA):

1. no quantifiers
2. all occurrences of $*$ have at least one argument that is a decimal numeral

Many SMT solvers (e.g., cvc5, Z3) use a version of the Simplex method as the theory solver for QF_LRA

Linear Programming

A *linear program (LP)* consists of:

1. An $m \times n$ matrix $\boldsymbol{A}$, the *constraint matrix*
2. An m-dimensional vector $\boldsymbol{b}$
3. An n-dimensional vector $\boldsymbol{c}$, the *objective function*

Let $\boldsymbol{x}$ a vector of n variables

Goal: Find a solution $\boldsymbol{x}$ that maximizes $\boldsymbol{c}^T\boldsymbol{x}$ subject to the linear constraints $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$

Linear Programming

A *linear program (LP)* consists of:

1. An $m \times n$ matrix $\boldsymbol{A}$, the *constraint matrix*
2. An m-dimensional vector $\boldsymbol{b}$
3. An n-dimensional vector $\boldsymbol{c}$, the *objective function*

Let $\boldsymbol{x}$ a vector of n variables

Goal: Find a solution $\boldsymbol{x}$ that maximizes $\boldsymbol{c}^T\boldsymbol{x}$ subject to the linear constraints $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$

Linear Programming

A *linear program (LP)* consists of:

1. An $m \times n$ matrix $\boldsymbol{A}$, the *constraint matrix*
2. An m-dimensional vector $\boldsymbol{b}$
3. An n-dimensional vector $\boldsymbol{c}$, the *objective function*

Let $\boldsymbol{x}$ a vector of n variables

Goal: Find a solution $\boldsymbol{x}$ that maximizes $\boldsymbol{c}^T\boldsymbol{x}$ subject to the linear constraints $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$

Example and Terminology

Maximize $2x_2 - x_1$ subject to:

$$x_1 + x2 \leq 3$$
$$2x_1 - x_2 \leq -5$$

Here:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \quad b = \begin{bmatrix} 3 \\ -5 \end{bmatrix} \quad c = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

Find $\boldsymbol{x}$ that maximizes $\boldsymbol{c}^T\boldsymbol{x}$, subject to $\boldsymbol{Ax} \leq \boldsymbol{b}$

Example and Terminology

Maximize $2x_2 - x_1$ subject to:

$$x_1 + x2 \leq 3$$
$$2x_1 - x_2 \leq -5$$

An assignment of $\boldsymbol{x}$ is a *feasible solution* if it satisfies $\boldsymbol{Ax} \leq \boldsymbol{b}$
Otherwise, it is an *infeasible solution*

Is $\langle 0, 0 \rangle$ a feasible solution? ✗
Is $\langle -2, 1 \rangle$ a feasible solution? ✓

For a given assignment of $\boldsymbol{x}$, the value of $\boldsymbol{c}^T\boldsymbol{x}$ is the *objective value*, or *cost*, of $\boldsymbol{x}$

What is the objective value of $\langle -2, 1 \rangle$? 4

Example and Terminology

Maximize $2x_2 - x_1$ subject to:

$$
\begin{aligned}
x_1 + x2 &\leq 3 \\
2x_1 - x_2 &\leq -5
\end{aligned}
$$

An assignment of $\boldsymbol{x}$ is a *feasible solution* if it satisfies $\boldsymbol{Ax} \leq \boldsymbol{b}$
Otherwise, it is an *infeasible solution*

Is $\langle 0, 0 \rangle$ a feasible solution? ✗
Is $\langle -2, 1 \rangle$ a feasible solution? ✓

For a given assignment of x, the value of $c^T x$ is the *objective value*, or *cost*, of x

What is the objective value of $\langle -2, 1 \rangle$? 4

Example and Terminology

Maximize $2x_2 - x_1$ subject to:

$$x_1 + x2 \leq 3$$
$$2x_1 - x_2 \leq -5$$

An assignment of $\boldsymbol{x}$ is a *feasible solution* if it satisfies $\boldsymbol{Ax} \leq \boldsymbol{b}$
Otherwise, it is an *infeasible solution*

Is $\langle 0, 0 \rangle$ a feasible solution? ✗
Is $\langle -2, 1 \rangle$ a feasible solution? ✓

For a given assignment of $\boldsymbol{x}$, the value of $\boldsymbol{c}^T\boldsymbol{x}$ is the *objective value*, or *cost*, of $\boldsymbol{x}$

What is the objective value of $\langle -2, 1 \rangle$? 4

Example and Terminology

Maximize $2x_2 - x_1$ subject to:

$$x_1 + x2 \leq 3$$
$$2x_1 - x_2 \leq -5$$

An assignment of $\boldsymbol{x}$ is a *feasible solution* if it satisfies $\boldsymbol{Ax} \leq \boldsymbol{b}$
Otherwise, it is an *infeasible solution*

Is $\langle 0, 0 \rangle$ a feasible solution? ✗
Is $\langle -2, 1 \rangle$ a feasible solution? ✓

For a given assignment of $\boldsymbol{x}$, the value of $\boldsymbol{c}^T\boldsymbol{x}$ is the *objective value*, or *cost*, of $\boldsymbol{x}$

What is the objective value of $\langle -2, 1 \rangle$? 4

Example and Terminology

Maximize $2x_2 - x_1$ subject to:

$$x_1 + x2 \leq 3$$
$$2x_1 - x_2 \leq -5$$

An assignment of $\boldsymbol{x}$ is a *feasible solution* if it satisfies $\boldsymbol{Ax} \leq \boldsymbol{b}$
Otherwise, it is an *infeasible solution*

Is $\langle 0, 0 \rangle$ a feasible solution? ✗
Is $\langle -2, 1 \rangle$ a feasible solution? ✓

For a given assignment of x, the value of $c^T x$ is the *objective value*, or *cost*, of x

What is the objective value of $\langle -2, 1 \rangle$? 4

Example and Terminology

Maximize $2x_2 - x_1$ subject to:

$$
\begin{aligned}
x_1 + x2 &\leq 3 \\
2x_1 - x_2 &\leq -5
\end{aligned}
$$

An assignment of $\boldsymbol{x}$ is a *feasible solution* if it satisfies $\boldsymbol{Ax} \leq \boldsymbol{b}$
Otherwise, it is an *infeasible solution*

Is $\langle 0, 0 \rangle$ a feasible solution? ✗
Is $\langle -2, 1 \rangle$ a feasible solution? ✓

For a given assignment of $\boldsymbol{x}$, the value of $\boldsymbol{c}^T\boldsymbol{x}$ is the *objective value*, or *cost*, of $\boldsymbol{x}$

What is the objective value of $\langle -2, 1 \rangle$? 4

Example and Terminology

Maximize $2x_2 - x_1$ subject to:

$$x_1 + x2 \leq 3$$
$$2x_1 - x_2 \leq -5$$

An assignment of $\boldsymbol{x}$ is a *feasible solution* if it satisfies $\boldsymbol{Ax} \leq \boldsymbol{b}$
Otherwise, it is an *infeasible solution*

Is $\langle 0, 0 \rangle$ a feasible solution? ✗
Is $\langle -2, 1 \rangle$ a feasible solution? ✓

For a given assignment of $\boldsymbol{x}$, the value of $\boldsymbol{c}^T \boldsymbol{x}$ is the *objective value*, or *cost*, of $\boldsymbol{x}$

What is the objective value of $\langle -2, 1 \rangle$? 4

Example and Terminology

Maximize $2x_2 - x_1$ subject to:

$$x_1 + x2 \leq 3$$
$$2x_1 - x_2 \leq -5$$

An assignment of $\boldsymbol{x}$ is a *feasible solution* if it satisfies $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$
Otherwise, it is an *infeasible solution*

Is $\langle 0, 0 \rangle$ a feasible solution? ✗
Is $\langle -2, 1 \rangle$ a feasible solution? ✓

For a given assignment of $\boldsymbol{x}$, the value of $\boldsymbol{c}^T\boldsymbol{x}$ is the *objective value*, or *cost*, of $\boldsymbol{x}$

What is the objective value of $\langle -2, 1 \rangle$? 4

Example and Terminology

An *optimal solution* is feasible solution with a maximal objective value, over all feasible solutions

If a linear program has no feasible solutions, the linear program is *infeasible*

The linear program is *unbounded* if the objective value of the optimal solution is ∞

Example and Terminology

An *optimal solution* is feasible solution with a maximal objective value, over all feasible solutions

If a linear program has no feasible solutions, the linear program is *infeasible*

The linear program is *unbounded* if the objective value of the optimal solution is ∞

Example and Terminology

An *optimal solution* is feasible solution with a maximal objective value, over all feasible solutions

If a linear program has no feasible solutions, the linear program is *infeasible*

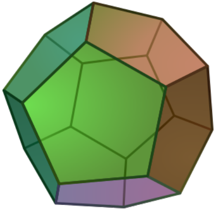
The linear program is *unbounded* if the objective value of the optimal solution is ∞

Geometric Interpretation

A *polytope* the generalization of polyhedron from 3-dimensional space to higher dimensions

A polytope P is *convex* if for all $v_1, v_2 \in \mathbb{R}^n \cap P$, $\lambda v_1 + (1 - \lambda) v_2 \in P$ for all $\lambda \in [0, 1]$

In other words, every point on the line segment connecting two points in P is also in P



Note: For an $m \times n$ constraint matrix $\boldsymbol{A}$, the set of points $P = \{ x \mid \boldsymbol{Ax} \leq \boldsymbol{b} \}$ form a *convex polytope* in n-dimensional space

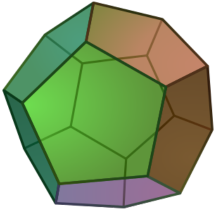
LP goals: find a point in the polytope that maximizes $c^\top x$ for a given c

Geometric Interpretation

A *polytope* the generalization of polyhedron from 3-dimensional space to higher dimensions

A polytope P is *convex* if for all $v_1, v_2 \in \mathbb{R}^n \cap P$, $\lambda v_1 + (1 - \lambda) v_2 \in P$ for all $\lambda \in [0, 1]$

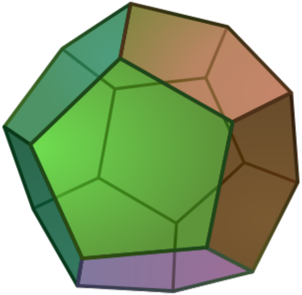
In other words, every point on the line segment connecting two points in P is also in P



Note: For an $m \times n$ constraint matrix $\boldsymbol{A}$, the set of points $P = \{ x \mid \boldsymbol{Ax} \leq \boldsymbol{b} \}$ form a *convex polytope* in n-dimensional space

LP goals: find a point in the polytope that maximizes $\boldsymbol{c}^T\boldsymbol{x}$ for a given $\boldsymbol{c}$

Geometric Interpretation



The LP is infeasible iff the polytope is empty

The LP is unbounded iff the polytope is open in the direction of the objective function

The optimal solution for a bounded LP must lie on a vertex of the polytope

Satisfiability as Linear Programming

Our goal: use LP to check the satisfiability of sets of linear $\mathcal{T}_{\text{RA}}$-literals

Satisfiability as Linear Programming

Our goal: use LP to check the satisfiability of sets of linear $\mathcal{T}_{\mathsf{RA}}$-literals

Step 1: convert equalities to inequalities

Satisfiability as Linear Programming

Our goal: use LP to check the satisfiability of sets of linear $\mathcal{T}_{\mathsf{RA}}$-literals

Step 1: convert equalities to inequalities

A linear $\mathcal{T}_{\mathsf{RA}}$-equality can be written to have the form $\boldsymbol{a}^T\boldsymbol{x} = b$

Satisfiability as Linear Programming

Our goal: use LP to check the satisfiability of sets of linear $\mathcal{T}_{\mathsf{RA}}$-literals

Step 1: convert equalities to inequalities

A linear $\mathcal{T}_{\mathsf{RA}}$-equality can be written to have the form $\boldsymbol{a}^T\boldsymbol{x} = b$

We rewrite this further as $\boldsymbol{a}^T\boldsymbol{x} \geq b$ and $\boldsymbol{a}^T\boldsymbol{x} \leq b$

And finally to $-\boldsymbol{a}^T\boldsymbol{x} \leq -b$, $\boldsymbol{a}^T\boldsymbol{x} \leq b$

Satisfiability as Linear Programming

Our goal: use LP to check the satisfiability of sets of linear $\mathcal{T}_{\mathsf{RA}}$-literals

Step 2: handle inequalities

Satisfiability as Linear Programming

Our goal: use LP to check the satisfiability of sets of linear $\mathcal{T}_{\mathrm{RA}}$-literals

Step 2: handle inequalities

A $\mathcal{T}_{\mathrm{RA}}$-literal of the form $\boldsymbol{a}^T\boldsymbol{x} \leq b$ is already in the desired form

Satisfiability as Linear Programming

Our goal: use LP to check the satisfiability of sets of linear $\mathcal{T}_{\mathrm{RA}}$-literals

Step 2: handle inequalities

A $\mathcal{T}_{\mathrm{RA}}$-literal of the form $\boldsymbol{a}^T\boldsymbol{x} \leq b$ is already in the desired form

A $\mathcal{T}_{\mathrm{RA}}$-literal of the form $\neg(\boldsymbol{a}^T\boldsymbol{x} \leq b)$ is transformed as follows

$$\neg(\boldsymbol{a}^T\boldsymbol{x} \leq b) \longrightarrow \boldsymbol{a}^T\boldsymbol{x} > b \longrightarrow -\boldsymbol{a}^T\boldsymbol{x} < -b \longrightarrow -\boldsymbol{a}^T\boldsymbol{x} + y \leq -b,\ y > 0$$

where y is a fresh variable used for all negated inequalities

Satisfiability as Linear Programming

Our goal: use LP to check the satisfiability of sets of linear $\mathcal{T}_{\mathsf{RA}}$-literals

Step 2: handle inequalities

A $\mathcal{T}_{\mathsf{RA}}$-literal of the form $\boldsymbol{a}^T\boldsymbol{x} \leq b$ is already in the desired form

A $\mathcal{T}_{\mathsf{RA}}$-literal of the form $\neg(\boldsymbol{a}^T\boldsymbol{x} \leq b)$ is transformed as follows

$$\neg(\boldsymbol{a}^T\boldsymbol{x} \leq b) \longrightarrow \boldsymbol{a}^T\boldsymbol{x} > b \longrightarrow -\boldsymbol{a}^T\boldsymbol{x} < -b \longrightarrow -\boldsymbol{a}^T\boldsymbol{x} + y \leq -b,\ y > 0$$

where y is a fresh variable used for all negated inequalities

Example: $\neg(2x_1 - x_2 \leq 3)$ rewrites to $-2x_1 + x_2 + y \leq -3,\ y > 0$

Satisfiability as Linear Programming

Our goal: use LP to check the satisfiability of sets of linear $\mathcal{T}_{\mathsf{RA}}$-literals

Step 2: handle inequalities

A $\mathcal{T}_{\mathsf{RA}}$-literal of the form $\boldsymbol{a}^T\boldsymbol{x} \leq b$ is already in the desired form

A $\mathcal{T}_{\mathsf{RA}}$-literal of the form $\neg(\boldsymbol{a}^T\boldsymbol{x} \leq b)$ is transformed as follows

$$\neg(\boldsymbol{a}^T\boldsymbol{x} \leq b) \longrightarrow \boldsymbol{a}^T\boldsymbol{x} > b \longrightarrow -\boldsymbol{a}^T\boldsymbol{x} < -b \longrightarrow -\boldsymbol{a}^T\boldsymbol{x} + y \leq -b,\ y > 0$$

where y is a fresh variable used for all negated inequalities

If there are no negated inequalities, add the inequality $y \leq 1$ where y is a fresh var

Satisfiability as Linear Programming

Our goal: use LP to check the satisfiability of sets of linear $\mathcal{T}_{\mathsf{RA}}$-literals

Step 2: handle inequalities

A $\mathcal{T}_{\mathsf{RA}}$-literal of the form $\boldsymbol{a}^T\boldsymbol{x} \leq b$ is already in the desired form

A $\mathcal{T}_{\mathsf{RA}}$-literal of the form $\neg(\boldsymbol{a}^T\boldsymbol{x} \leq b)$ is transformed as follows

$$\neg(\boldsymbol{a}^T\boldsymbol{x} \leq b) \longrightarrow \boldsymbol{a}^T\boldsymbol{x} > b \longrightarrow -\boldsymbol{a}^T\boldsymbol{x} < -b \longrightarrow -\boldsymbol{a}^T\boldsymbol{x} + y \leq -b,\ y > 0$$

where y is a fresh variable used for all negated inequalities

If there are no negated inequalities, add the inequality $y \leq 1$ where y is a fresh var

In either case, we have the set of the form $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \cup \{\, y > 0 \,\}$

Satisfiability as Linear Programming

Our goal: use LP to check the satisfiability of sets of linear $\mathcal{T}_{\mathsf{RA}}$-literals

Step 3: check the satisfiability of $\boldsymbol{Ax} \leq \boldsymbol{b} \cup \{ y > 0 \}$

Satisfiability as Linear Programming

Our goal: use LP to check the satisfiability of sets of linear $\mathcal{T}_{\mathsf{RA}}$-literals

Step 3: check the satisfiability of $\boldsymbol{Ax} \leq \boldsymbol{b} \cup \{ y > 0 \}$

Encode that as the LP: maximize y subject to $\boldsymbol{Ax} \leq \boldsymbol{b}$

Satisfiability as Linear Programming

Our goal: use LP to check the satisfiability of sets of linear $\mathcal{T}_{\mathsf{RA}}$-literals

Step 3: check the satisfiability of $\boldsymbol{Ax} \le \boldsymbol{b} \cup \{ y > 0 \}$

Encode that as the LP: maximize y subject to $\boldsymbol{Ax} \le \boldsymbol{b}$

The final system is satisfiable iff the optimal value for y is positive

Methods for solving LP problems

- *Simplex* (Dantzig, 1949) Exponential time (probably)
- *Ellipsoid* (Khachian, 1979) Polynomial time
- *Interior-point* (Karmarkar, 1984) Polynomial time

Although the Simplex method is the oldest and the least efficient in theory
it can be implemented to be quite efficient in practice

It remains the most popular and we focus on it next

Methods for solving LP problems

- *Simplex* (Dantzig, 1949) Exponential time (probably)
- *Ellipsoid* (Khachian, 1979) Polynomial time
- *Interior-point* (Karmarkar, 1984) Polynomial time

Although the Simplex method is the oldest and the least efficient in theory
it can be implemented to be quite efficient in practice

It remains the most popular and we focus on it next

Methods for solving LP problems

- *Simplex* (Dantzig, 1949) Exponential time (probably)
- *Ellipsoid* (Khachian, 1979) Polynomial time
- *Interior-point* (Karmarkar, 1984) Polynomial time

Although the Simplex method is the oldest and the least efficient in theory
it can be implemented to be quite efficient in practice

It remains the most popular and we focus on it next

Standard Form

The general form of LP is to maximize objective function subject to a system of inequalities

However, the Simplex method is easier to present
if we make the additional assumption that all variables are non-negative:

$$
\begin{aligned}
\text{maximize} \quad & \sum_{j=1}^{n} c_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad \text{for } i = 1, \dots, m \\
& x_j \ge 0 \quad \text{for } j = 1, \dots, n
\end{aligned}
$$

We call this the *standard form*

This causes no loss of generality since any LP can be transformed to standard form

Standard Form

The general form of LP is to maximize objective function subject to a system of inequalities

However, the Simplex method is easier to present
if we make the additional assumption that all variables are non-negative:

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad \text{for } i = 1, \ldots, m \\ & x_j \ge 0 \quad \text{for } j = 1, \ldots, n \end{aligned}$$

We call this the *standard form*

This causes no loss of generality since any LP can be transformed to standard form

Standard Form

The general form of LP is to maximize objective function subject to a system of inequalities

However, the Simplex method is easier to present
if we make the additional assumption that all variables are non-negative:

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for } i = 1, \ldots, m \\ & x_j \geq 0 \quad \text{for } j = 1, \ldots, n \end{aligned}$$

We call this the *standard form*

This causes no loss of generality since any LP can be transformed to standard form

Standard Form

The general form of LP is to maximize objective function subject to a system of inequalities

However, the Simplex method is easier to present
if we make the additional assumption that all variables are non-negative:

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for } i = 1, \ldots, m \\ & x_j \geq 0 \quad \text{for } j = 1, \ldots, n \end{aligned}$$

We call this the *standard form*

This causes no loss of generality since any LP can be transformed to standard form

Standard Form

The general form of LP is to maximize objective function subject to a system of inequalities

However, the Simplex method is easier to present
if we make the additional assumption that all variables are non-negative:

$$
\begin{aligned}
\text{maximize} \quad & \sum_{j=1}^{n} c_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for } i = 1, \ldots, m \\
& x_j \geq 0 \quad \text{for } j = 1, \ldots, n
\end{aligned}
$$

We call this the *standard form*

This causes no loss of generality since any LP can be transformed to standard form

Standard Form

$$
\begin{aligned}
\text{maximize} \quad & \sum_{j=1}^{n} c_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for } i = 1, \ldots, m \\
& x_j \geq 0 \quad \text{for } j = 1, \ldots, n
\end{aligned}
$$

Running example:

$$
\begin{aligned}
\max \quad & 5x_1 + 4x_2 + 3x_3 \\
\text{s.t.} \quad & \begin{cases} 2x_1 + 3x_2 + x_3 \leq 5 \\ 4x_1 + x_2 + 2x_3 \leq 11 \\ 3x_1 + 4x_2 + 2x_3 \leq 8 \\ x_1, x_2, x_3 \geq 0 \end{cases}
\end{aligned}
$$

Slack Variables

Observe the first inequation

$$2x_1 + 3x_2 + x_3 \leq 5$$

$$\max \quad 5x_1 + 4x_2 + 3x_3$$
$$\text{s.t.} \quad \begin{cases} 2x_1 + 3x_2 + x_3 \leq 5 \\ 4x_1 + x_2 + 2x_3 \leq 11 \\ 3x_1 + 4x_2 + 2x_3 \leq 8 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

Define a new variable to represent the *slack*:

$$x_4 = 5 - 2x_1 - 3x_2 - x_3, \quad x_4 \geq 0$$

Do this to every each constraint so everything becomes equalities

Define a new variable to represent the objective value: $z = 5x_1 + 4x_2 + 3x_3$

Slack Variables

Observe the first inequation

$$2x_1 + 3x_2 + x_3 \leq 5$$

Define a new variable to represent the *slack*:

$$x_4 = 5 - 2x_1 - 3x_2 - x_3, \quad x_4 \geq 0$$

Do this to every each constraint so everything becomes equalities

$$\max \quad 5x_1 + 4x_2 + 3x_3$$

$$\text{s.t.} \quad \begin{cases} 2x_1 + 3x_2 + x_3 \leq 5 \\ 4x_1 + x_2 + 2x_3 \leq 11 \\ 3x_1 + 4x_2 + 2x_3 \leq 8 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

Define a new variable to represent the objective value: $z = 5x_1 + 4x_2 + 3x_3$

Slack Variables

$$\max \quad 5x_1 + 4x_2 + 3x_3$$
$$\text{s.t.} \quad \begin{cases} 2x_1 + 3x_2 + x_3 \leq 5 \\ 4x_1 + x_2 + 2x_3 \leq 11 \\ 3x_1 + 4x_2 + 2x_3 \leq 8 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

$$\longrightarrow$$

$$\max \quad z$$
$$\text{s.t.} \quad \begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 0 + 5x_1 + 4x_2 + 3x_3 \\ x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 \end{cases}$$

New variables are called *slack variables*

Optimal solution remains optimal for the new problem

Slack Variables

$$
\max \quad 5x_1 + 4x_2 + 3x_3
$$
$$
\text{s.t.} \quad \begin{cases} 2x_1 + 3x_2 + x_3 \leq 5 \\ 4x_1 + x_2 + 2x_3 \leq 11 \\ 3x_1 + 4x_2 + 2x_3 \leq 8 \\ x_1, x_2, x_3 \geq 0 \end{cases}
$$

$$
\longrightarrow
$$

$$
\max \quad z
$$
$$
\text{s.t.} \quad \begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 0 + 5x_1 + 4x_2 + 3x_3 \\ x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 \end{cases}
$$

New variables are called *slack variables*

Optimal solution remains optimal for the new problem

The Simplex Strategy

- Start with a feasible solution
  - For our example, assign 0 to all original variables
    - $x_1 \mapsto 0, x_2 \mapsto 0, x_3 \mapsto 0$
  - Assign the introduced vars their computed value
    - $x_4 \mapsto 5, x_5 \mapsto 11, x_6 \mapsto 8, z \mapsto 0$
- Iteratively improve the objective value
  - Go from $\boldsymbol{x}$ to $\boldsymbol{x}'$ only if $z(\boldsymbol{x}) \leq z(\boldsymbol{x}')$

$$
\begin{cases}
x_4 = 5 - 2x_1 - 3x_2 - x_3 \\
x_5 = 11 - 4x_1 - x_2 - 2x_3 \\
x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\
z = 5x_1 + 4x_2 + 3x_3
\end{cases}
$$

What can we improve here?

One option: make x_1 larger, leave x_2, x_3 unchanged

- $x_1 = 1 \Rightarrow x_4 = 3, x_5 = 7, x_6 = 1, z = 5$ ✓
- $x_1 = 2 \Rightarrow x_4 = 1, x_5 = 3, x_6 = 2, z = 10$ ✓
- $x_1 = 3 \Rightarrow x_4 = -1, \ldots$ ✗ no longer feasible!

The Simplex Strategy

- Start with a feasible solution
  - For our example, assign 0 to all original variables
    - $x_1 \mapsto 0, x_2 \mapsto 0, x_3 \mapsto 0$
  - Assign the introduced vars their computed value
    - $x_4 \mapsto 5, x_5 \mapsto 11, x_6 \mapsto 8, z \mapsto 0$
- Iteratively improve the objective value
  - Go from $\boldsymbol{x}$ to $\boldsymbol{x}'$ only if $z(\boldsymbol{x}) \leq z(\boldsymbol{x}')$

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

What can we improve here?

One option: make x_1 larger, leave x_2, x_3 unchanged

- $x_1 = 1 \quad \Rightarrow \quad x_4 = 3, x_5 = 7, x_6 = 1, z = 5$ ✓
- $x_1 = 2 \quad \Rightarrow \quad x_4 = 1, x_5 = 3, x_6 = 2, z = 10$ ✓
- $x_1 = 3 \quad \Rightarrow \quad x_4 = -1, \ldots$ ✗ no longer feasible!

The Simplex Strategy

- Start with a feasible solution
  - For our example, assign 0 to all original variables
    - ► $x_1 \mapsto 0, x_2 \mapsto 0, x_3 \mapsto 0$
  - Assign the introduced vars their computed value
    - ► $x_4 \mapsto 5, x_5 \mapsto 11, x_6 \mapsto 8, z \mapsto 0$
- Iteratively improve the objective value
  - Go from $\boldsymbol{x}$ to $\boldsymbol{x}'$ only if $z(\boldsymbol{x}) \leq z(\boldsymbol{x}')$

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

What can we improve here?

One option: make x_1 larger, leave x_2, x_3 unchanged

- $x_1 = 1 \quad \Rightarrow \quad x_4 = 3, x_5 = 7, x_6 = 1, z = 5$ ✓
- $x_1 = 2 \quad \Rightarrow \quad x_4 = 1, x_5 = 3, x_6 = 2, z = 10$ ✓
- $x_1 = 3 \quad \Rightarrow \quad x_4 = -1, \ldots$ ✗ no longer feasible!

The Simplex Strategy

- Start with a feasible solution
  - For our example, assign 0 to all original variables
    - $x_1 \mapsto 0, x_2 \mapsto 0, x_3 \mapsto 0$
  - Assign the introduced vars their computed value
    - $x_4 \mapsto 5, x_5 \mapsto 11, x_6 \mapsto 8, z \mapsto 0$
- Iteratively improve the objective value
  - Go from $\boldsymbol{x}$ to $\boldsymbol{x}'$ only if $z(\boldsymbol{x}) \leq z(\boldsymbol{x}')$

$$
\begin{cases}
x_4 = 5 - 2x_1 - 3x_2 - x_3 \\
x_5 = 11 - 4x_1 - x_2 - 2x_3 \\
x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\
z = 5x_1 + 4x_2 + 3x_3
\end{cases}
$$

What can we improve here?

One option: make x_1 larger, leave x_2, x_3 unchanged

- $x_1 = 1 \quad \Rightarrow \quad x_4 = 3, x_5 = 7, x_6 = 1, z = 5$ ✓
- $x_1 = 2 \quad \Rightarrow \quad x_4 = 1, x_5 = 3, x_6 = 2, z = 10$ ✓
- $x_1 = 3 \quad \Rightarrow \quad x_4 = -1, \ldots$ ✗ no longer feasible!

The Simplex Strategy

- Start with a feasible solution
  - For our example, assign 0 to all original variables
    - ▶ $x_1 \mapsto 0, x_2 \mapsto 0, x_3 \mapsto 0$
  - Assign the introduced vars their computed value
    - ▶ $x_4 \mapsto 5, x_5 \mapsto 11, x_6 \mapsto 8, z \mapsto 0$
- Iteratively improve the objective value
  - Go from $\boldsymbol{x}$ to $\boldsymbol{x}'$ only if $z(\boldsymbol{x}) \leq z(\boldsymbol{x}')$

$$
\begin{cases}
x_4 = 5 - 2x_1 - 3x_2 - x_3 \\
x_5 = 11 - 4x_1 - x_2 - 2x_3 \\
x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\
z = 5x_1 + 4x_2 + 3x_3
\end{cases}
$$

What can we improve here?

One option: make x_1 larger, leave x_2, x_3 unchanged

- $x_1 = 1 \quad \Rightarrow \quad x_4 = 3, x_5 = 7, x_6 = 1, z = 5$ ✓
- $x_1 = 2 \quad \Rightarrow \quad x_4 = 1, x_5 = 3, x_6 = 2, z = 10$ ✓
- $x_1 = 3 \quad \Rightarrow \quad x_4 = -1, \ldots$ ✗ no longer feasible!

The Simplex Strategy

- Start with a feasible solution
  - For our example, assign 0 to all original variables
    - $x_1 \mapsto 0, x_2 \mapsto 0, x_3 \mapsto 0$
  - Assign the introduced vars their computed value
    - $x_4 \mapsto 5, x_5 \mapsto 11, x_6 \mapsto 8, z \mapsto 0$
- Iteratively improve the objective value
  - Go from $\boldsymbol{x}$ to $\boldsymbol{x}'$ only if $z(\boldsymbol{x}) \leq z(\boldsymbol{x}')$

$$
\begin{cases}
x_4 = 5 - 2x_1 - 3x_2 - x_3 \\
x_5 = 11 - 4x_1 - x_2 - 2x_3 \\
x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\
z = 5x_1 + 4x_2 + 3x_3
\end{cases}
$$

What can we improve here?

One option: make x_1 larger, leave x_2, x_3 unchanged

- $x_1 = 1 \quad \Rightarrow \quad x_4 = 3, x_5 = 7, x_6 = 1, z = 5$ ✓
- $x_1 = 2 \quad \Rightarrow \quad x_4 = 1, x_5 = 3, x_6 = 2, z = 10$ ✓
- $x_1 = 3 \quad \Rightarrow \quad x_4 = -1, \ldots$ ✗ no longer feasible!

The Simplex Strategy

- Start with a feasible solution
  - For our example, assign 0 to all original variables
    - $x_1 \mapsto 0, x_2 \mapsto 0, x_3 \mapsto 0$
  - Assign the introduced vars their computed value
    - $x_4 \mapsto 5, x_5 \mapsto 11, x_6 \mapsto 8, z \mapsto 0$
- Iteratively improve the objective value
  - Go from $\boldsymbol{x}$ to $\boldsymbol{x}'$ only if $z(\boldsymbol{x}) \leq z(\boldsymbol{x}')$

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

What can we improve here?

One option: make x_1 larger, leave x_2, x_3 unchanged

- $x_1 = 1 \Rightarrow x_4 = 3, x_5 = 7, x_6 = 1, z = 5$ ✓
- $x_1 = 2 \Rightarrow x_4 = 1, x_5 = 3, x_6 = 2, z = 10$ ✓
- $x_1 = 3 \Rightarrow x_4 = -1, \ldots$ ✗ no longer feasible!

The Simplex Strategy

Moral of the story:

- Can't increase x_1 too much
- Increase it as much as possible, without compromising feasibility

$x_1 \mapsto 0, x_2 \mapsto 0, x_3 \mapsto 0$

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases} \longrightarrow \quad x_1 \le \frac{5}{2},\ x_1 \le \frac{11}{4},\ x_1 \le \frac{8}{3}$$

Select the tightest bound, $x_1 \le \frac{5}{2}$

- New assignment: $x_1 \mapsto \frac{5}{2}, x_2 \mapsto x_3 \mapsto x_4 \mapsto 0, x_5 \mapsto 1, x_6 \mapsto \frac{1}{2}, z \mapsto \frac{25}{2}$
- This indeed improves the value of z

The Simplex Strategy

Moral of the story:

- Can't increase x_1 too much
- Increase it as much as possible, without compromising feasibility

$$x_1 \mapsto 0, x_2 \mapsto 0, x_3 \mapsto 0$$

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases} \longrightarrow \quad x_1 \le \tfrac{5}{2},\ x_1 \le \tfrac{11}{4},\ x_1 \le \tfrac{8}{3}$$

Select the tightest bound, $x_1 \le \frac{5}{2}$

- New assignment: $x_1 \mapsto \frac{5}{2}, x_2 \mapsto x_3 \mapsto x_4 \mapsto 0, x_5 \mapsto 1, x_6 \mapsto \frac{1}{2}, z \mapsto \frac{25}{2}$
- This indeed improves the value of z

The Simplex Strategy

Moral of the story:

- Can't increase x_1 too much
- Increase it as much as possible, without compromising feasibility

$$x_1 \mapsto 0, x_2 \mapsto 0, x_3 \mapsto 0$$

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases} \longrightarrow \quad x_1 \leq \frac{5}{2},\ x_1 \leq \frac{11}{4},\ x_1 \leq \frac{8}{3}$$

Select the tightest bound, $x_1 \leq \frac{5}{2}$

- New assignment: $x_1 \mapsto \frac{5}{2}, x_2 \mapsto x_3 \mapsto x_4 \mapsto 0, x_5 \mapsto 1, x_6 \mapsto \frac{1}{2}, z \mapsto \frac{25}{2}$
- This indeed improves the value of z

The Simplex Strategy

Moral of the story:

- Can't increase x_1 too much
- Increase it as much as possible, without compromising feasibility

$$x_1 \mapsto 0, x_2 \mapsto 0, x_3 \mapsto 0$$

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases} \longrightarrow \quad x_1 \leq \frac{5}{2},\ x_1 \leq \frac{11}{4},\ x_1 \leq \frac{8}{3}$$

Select the tightest bound, $x_1 \leq \frac{5}{2}$

- New assignment: $x_1 \mapsto \frac{5}{2}, x_2 \mapsto x_3 \mapsto x_4 \mapsto 0, x_5 \mapsto 1, x_6 \mapsto \frac{1}{2}, z \mapsto \frac{25}{2}$
- This indeed improves the value of z

The Simplex Strategy

Currently,

$x_1 \mapsto \frac{5}{2}, x_2 \mapsto x_3 \mapsto x_4 \mapsto 0, x_5 \mapsto 1, x_6 \mapsto \frac{1}{2}, z \mapsto \frac{25}{2}$

How do we continue?

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

For the first iteration we had:

- A feasible solution ✓
- An equation system, where
  - variables with positive value are expressed in terms of variables with 0 value

Does the current equation system satisfy this property? No

Need to update the equations

The Simplex Strategy

Currently,

$x_1 \mapsto \frac{5}{2}, x_2 \mapsto x_3 \mapsto x_4 \mapsto 0, x_5 \mapsto 1, x_6 \mapsto \frac{1}{2}, z \mapsto \frac{25}{2}$

How do we continue?

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

For the first iteration we had:

- A feasible solution ✓
- An equation system, where
  - variables with positive value are expressed in terms of variables with 0 value

Does the current equation system satisfy this property? No

Need to update the equations

The Simplex Strategy

Currently,

$x_1 \mapsto \frac{5}{2}, x_2 \mapsto x_3 \mapsto x_4 \mapsto 0, x_5 \mapsto 1, x_6 \mapsto \frac{1}{2}, z \mapsto \frac{25}{2}$

How do we continue?

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

For the first iteration we had:

- A feasible solution ✓
- An equation system, where
  - variables with positive value are expressed in terms of variables with 0 value

Does the current equation system satisfy this property? No

Need to update the equations

The Simplex Strategy

Currently,

$x_1 \mapsto \frac{5}{2}, x_2 \mapsto x_3 \mapsto x_4 \mapsto 0, x_5 \mapsto 1, x_6 \mapsto \frac{1}{2}, z \mapsto \frac{25}{2}$

How do we continue?

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

For the first iteration we had:

- A feasible solution ✓
- An equation system, where
  - variables with positive value are expressed in terms of variables with 0 value

Does the current equation system satisfy this property? No

Need to update the equations

The Simplex Strategy

Currently,

$x_1 \mapsto \frac{5}{2}, x_2 \mapsto x_3 \mapsto x_4 \mapsto 0, x_5 \mapsto 1, x_6 \mapsto \frac{1}{2}, z \mapsto \frac{25}{2}$

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

How do we continue?

For the first iteration we had:

- A feasible solution ✓
- An equation system, where
  - variables with positive value are expressed in terms of variables with 0 value

Does the current equation system satisfy this property? No

Need to update the equations

The Simplex Strategy

Currently,

$x_1 \mapsto \frac{5}{2}, x_2 \mapsto x_3 \mapsto x_4 \mapsto 0, x_5 \mapsto 1, x_6 \mapsto \frac{1}{2}, z \mapsto \frac{25}{2}$

How do we continue?

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

For the first iteration we had:

- A feasible solution ✓
- An equation system, where
  - variables with positive value are expressed in terms of variables with 0 value

Does the current equation system satisfy this property? No

Need to update the equations

The Simplex Strategy

$$x_1 \mapsto \frac{5}{2}, x_2 \mapsto 0, x_3 \mapsto 0, x_4 \mapsto 0$$

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

What should we change?

- **Initially:** x_1 was 0, x_4 was positive
- **Now:** x_1 is positive, x_4 is 0

Isolate x_1, eliminate from right-hand-side

$$x_4 = 5 - 2x_1 - 3x_2 - x_3 \longrightarrow x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4$$

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases} \longrightarrow \begin{cases} x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \\ x_5 = 1 + 5x_2 + 2x_4 \\ x_6 = \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \\ z = \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4 \end{cases}$$

The Simplex Strategy

$$x_1 \mapsto \frac{5}{2}, x_2 \mapsto 0, x_3 \mapsto 0, x_4 \mapsto 0$$

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

What should we change?

Initially: x_1 was 0, x_4 was positive
Now: x_1 is positive, x_4 is 0

Isolate x_1, eliminate from right-hand-side

$$x_4 = 5 - 2x_1 - 3x_2 - x_3 \longrightarrow x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4$$

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases} \longrightarrow \begin{cases} x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \\ x_5 = 1 + 5x_2 + 2x_4 \\ x_6 = \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \\ z = \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4 \end{cases}$$

The Simplex Strategy

$$x_1 \mapsto \tfrac{5}{2}, x_2 \mapsto 0, x_3 \mapsto 0, x_4 \mapsto 0$$

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

What should we change?

Initially: x_1 was 0, x_4 was positive
Now: x_1 is positive, x_4 is 0

Isolate x_1, eliminate from right-hand-side

$$x_4 = 5 - 2x_1 - 3x_2 - x_3 \longrightarrow x_1 = \tfrac{5}{2} - \tfrac{3}{2}x_2 - \tfrac{1}{2}x_3 - \tfrac{1}{2}x_4$$

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases} \longrightarrow \begin{cases} x_1 = \tfrac{5}{2} - \tfrac{3}{2}x_2 - \tfrac{1}{2}x_3 - \tfrac{1}{2}x_4 \\ x_5 = 1 + 5x_2 + 2x_4 \\ x_6 = \tfrac{1}{2} + \tfrac{1}{2}x_2 - \tfrac{1}{2}x_3 + \tfrac{3}{2}x_4 \\ z = \tfrac{25}{2} - \tfrac{7}{2}x_2 + \tfrac{1}{2}x_3 - \tfrac{5}{2}x_4 \end{cases}$$

The Simplex Strategy

$x_1 \mapsto \frac{5}{2}, x_2 \mapsto 0, x_3 \mapsto 0, x_4 \mapsto 0$

What should we change?

Initially: x_1 was 0, x_4 was positive
Now: x_1 is positive, x_4 is 0

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

Isolate x_1, eliminate from right-hand-side

$$x_4 = 5 - 2x_1 - 3x_2 - x_3 \longrightarrow x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4$$

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases} \longrightarrow \begin{cases} x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \\ x_5 = 1 + 5x_2 + 2x_4 \\ x_6 = \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \\ z = \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4 \end{cases}$$

The Simplex Strategy

$x_1 \mapsto \frac{5}{2}, x_2 \mapsto 0, x_3 \mapsto 0, x_4 \mapsto 0$

How can we improve z further?

$$\begin{cases} x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \\ x_5 = 1 + 5x_2 \qquad\qquad + 2x_4 \\ x_6 = \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \\ z = \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4 \end{cases}$$

Option 1: decrease x_2 or x_4
but we can't since $x_2, x_4 \geq 0$

Option 2: increase x_3
By how much?

x_3's bounds: $x_3 \leq 5, x_3 \leq \infty, x_3 \leq 1$

So we increase x_3 to 1

- New assignment: $x_1 \mapsto 2, x_2 \mapsto 0, x_3 \mapsto 1, x_4 \mapsto 0, x_5 \mapsto 0, x_6 \mapsto 0$
- This gives $z = 13$, which is again an improvement

The Simplex Strategy

$$x_1 \mapsto \frac{5}{2}, x_2 \mapsto 0, x_3 \mapsto 0, x_4 \mapsto 0$$

$$\begin{cases} x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \\ x_5 = 1 + 5x_2 + 2x_4 \\ x_6 = \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \\ z = \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4 \end{cases}$$

How can we improve z further?

Option 1: decrease x_2 or x_4
but we can't since $x_2, x_4 \geq 0$

Option 2: increase x_3
By how much?

x_3's bounds: $x_3 \leq 5, x_3 \leq \infty, x_3 \leq 1$

So we increase x_3 to 1

- New assignment: $x_1 \mapsto 2, x_2 \mapsto 0, x_3 \mapsto 1, x_4 \mapsto 0, x_5 \mapsto 0, x_6 \mapsto 0$
- This gives $z = 13$, which is again an improvement

The Simplex Strategy

$$x_1 \mapsto \tfrac{5}{2}, x_2 \mapsto 0, x_3 \mapsto 0, x_4 \mapsto 0$$

How can we improve z further?

Option 1: decrease x_2 or x_4
but we can't since $x_2, x_4 \geq 0$

Option 2: increase x_3
By how much?

$$\begin{cases} x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \\ x_5 = 1 + 5x_2 + 2x_4 \\ x_6 = \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \\ z = \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4 \end{cases}$$

x_3's bounds: $x_3 \leq 5, x_3 \leq \infty, x_3 \leq 1$

So we increase x_3 to 1

- New assignment: $x_1 \mapsto 2, x_2 \mapsto 0, x_3 \mapsto 1, x_4 \mapsto 0, x_5 \mapsto 0, x_6 \mapsto 0$
- This gives $z = 13$, which is again an improvement

The Simplex Strategy

$$x_1 \mapsto \tfrac{5}{2}, x_2 \mapsto 0, x_3 \mapsto 0, x_4 \mapsto 0$$

How can we improve z further?

Option 1: decrease x_2 or x_4
but we can't since $x_2, x_4 \geq 0$

Option 2: increase x_3
By how much?

$$\begin{cases} x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \\ x_5 = 1 + 5x_2 + 2x_4 \\ x_6 = \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \\ z = \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4 \end{cases}$$

x_3's bounds: $x_3 \leq 5$, $x_3 \leq \infty$, $x_3 \leq 1$

So we increase x_3 to 1

- New assignment: $x_1 \mapsto 2, x_2 \mapsto 0, x_3 \mapsto 1, x_4 \mapsto 0, x_5 \mapsto 0, x_6 \mapsto 0$
- This gives $z = 13$, which is again an improvement

The Simplex Strategy

$x_1 \mapsto \frac{5}{2}, x_2 \mapsto 0, x_3 \mapsto 0, x_4 \mapsto 0$

$$\begin{cases} x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \\ x_5 = 1 + 5x_2 + 2x_4 \\ x_6 = \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \\ z = \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4 \end{cases}$$

How can we improve z further?

Option 1: decrease x_2 or x_4
but we can't since $x_2, x_4 \geq 0$

Option 2: increase x_3
By how much?

x_3's bounds: $x_3 \leq 5$, $x_3 \leq \infty$, $x_3 \leq 1$

So we increase x_3 to 1

- New assignment: $x_1 \mapsto 2, x_2 \mapsto 0, x_3 \mapsto 1, x_4 \mapsto 0, x_5 \mapsto 0, x_6 \mapsto 0$
- This gives $z = 13$, which is again an improvement

The Simplex Strategy

$x_1 \mapsto \frac{5}{2}, x_2 \mapsto 0, x_3 \mapsto 0, x_4 \mapsto 0$

$$\begin{cases} x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \\ x_5 = 1 + 5x_2 + 2x_4 \\ x_6 = \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \\ z = \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4 \end{cases}$$

How can we improve z further?

Option 1: decrease x_2 or x_4
but we can't since $x_2, x_4 \geq 0$

Option 2: increase x_3
By how much?

x_3's bounds: $x_3 \leq 5$, $x_3 \leq \infty$, $x_3 \leq 1$

So we increase x_3 to 1

- New assignment: $x_1 \mapsto 2, x_2 \mapsto 0, x_3 \mapsto 1, x_4 \mapsto 0, x_5 \mapsto 0, x_6 \mapsto 0$
- This gives $z = 13$, which is again an improvement

The Simplex Strategy

Analogously to before, we switch x_6 and x_3, and eliminate x_3 from the right-hand sides

$$\begin{cases} x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \\ x_5 = 1 + 5x_2 \qquad\qquad + 2x_4 \\ x_6 = \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \\ z = \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4 \end{cases} \longrightarrow \begin{cases} x_1 = 2 - 2x_2 - 2x_4 + x_6 \\ x_5 = 1 + 5x_2 + 2x_4 \\ x_3 = 1 + x_2 + 3x_4 - 2x_6 \\ z = 13 - 3x_2 - x_4 - x_6 \end{cases}$$

The Simplex Strategy

$x_1 \mapsto 2, x_2 \mapsto 0, x_3 \mapsto 1$
$x_4 \mapsto 0, x_6 \mapsto 0$

$$\begin{cases} x_1 = 2 - 2x_2 - 2x_4 + x_6 \\ x_5 = 1 + 5x_2 + 2x_4 \\ x_3 = 1 + x_2 + 3x_4 - 2x_6 \\ z = 13 - 3x_2 - x_4 - x_6 \end{cases}$$

Can we improve z again?

- No, because $x_2, x_4, x_6 \geq 0$ and
- all appear with negative signs in the objective function

So we are done, and the optimal value of z is 13

The optimal solution is then $x_1 \mapsto 2, x_2 \mapsto 0, x_3 \mapsto 1$

The Simplex Strategy

$$x_1 \mapsto 2, x_2 \mapsto 0, x_3 \mapsto 1$$
$$x_4 \mapsto 0, x_6 \mapsto 0$$

Can we improve z again?

- No, because $x_2, x_4, x_6 \geq 0$ and
- all appear with negative signs in the objective function

$$\begin{cases} x_1 = 2 - 2x_2 - 2x_4 + x_6 \\ x_5 = 1 + 5x_2 + 2x_4 \\ x_3 = 1 + x_2 + 3x_4 - 2x_6 \\ z = 13 - 3x_2 - x_4 - x_6 \end{cases}$$

So we are done, and the optimal value of z is 13

The optimal solution is then $x_1 \mapsto 2, x_2 \mapsto 0, x_3 \mapsto 1$

The Simplex Strategy

$x_1 \mapsto 2, x_2 \mapsto 0, x_3 \mapsto 1$
$x_4 \mapsto 0, x_6 \mapsto 0$

$$\begin{cases} x_1 = 2 - 2x_2 - 2x_4 + x_6 \\ x_5 = 1 + 5x_2 + 2x_4 \\ x_3 = 1 + x_2 + 3x_4 - 2x_6 \\ z = 13 - 3x_2 - x_4 - x_6 \end{cases}$$

Can we improve z again?

- No, because $x_2, x_4, x_6 \geq 0$ and
- all appear with negative signs in the objective function

So we are done, and the optimal value of z is 13

The optimal solution is then $x_1 \mapsto 2, x_2 \mapsto 0, x_3 \mapsto 1$

The Simplex Strategy

$x_1 \mapsto 2, x_2 \mapsto 0, x_3 \mapsto 1$
$x_4 \mapsto 0, x_6 \mapsto 0$

$$\begin{cases} x_1 = 2 - 2x_2 - 2x_4 + x_6 \\ x_5 = 1 + 5x_2 + 2x_4 \\ x_3 = 1 + x_2 + 3x_4 - 2x_6 \\ z = 13 - 3x_2 - x_4 - x_6 \end{cases}$$

Can we improve z again?

- No, because $x_2, x_4, x_6 \geq 0$ and
- all appear with negative signs in the objective function

So we are done, and the optimal value of z is 13

The optimal solution is then $x_1 \mapsto 2, x_2 \mapsto 0, x_3 \mapsto 1$

The Simplex Algorithm

$$\text{maximize} \quad \sum_{j=1}^{n} c_j x_j$$

$$\text{s.t.} \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for } i = 1, \ldots, m$$

$$x_j \geq 0 \quad \text{for } j = 1, \ldots, n$$

1. Introduce slack variables $x_{n+1}, \ldots, x_{n+m}$
2. Set $x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij} x_j$ for $i = 1, \ldots, m$
3. Start with initial, feasible solution ($x_1 \mapsto 0, \ldots, x_n \mapsto 0$ in our example)
4. If some addends in the current objective function have positive coefficients, update the feasible solution to improve the objective value; otherwise, stop
5. Update the equations to maintain the invariant that all right-hand side vars have value 0
6. Go to step 4

Updating the Equations: Pivoting

As we progress towards the optimal solution, equations are updated

This computational process of constructing the new equation system is called *pivoting*

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

Invariants:

- Number of equations (m) never changes
- Variables are either on the left-hand side or the right-hand side, never both
  - Left-hand side variables are called *basic*
  - Right-hand side variables are called *non-basic*
- Non-basic variables always pressed against their bounds (always 0)
- Basic variable assignment determined by non-basic assignment and equations

Updating the Equations: Pivoting

As we progress towards the optimal solution, equations are updated

This computational process of constructing the new equation system is called *pivoting*

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

Invariants:

- Number of equations (m) never changes
- Variables are either on the left-hand side or the right-hand side, never both
  - Left-hand side variables are called *basic*
  - Right-hand side variables are called *non-basic*
- Non-basic variables always pressed against their bounds (always 0)
- Basic variable assignment determined by non-basic assignment and equations

Updating the Equations: Pivoting

As we progress towards the optimal solution, equations are updated

This computational process of constructing the new equation system is called *pivoting*

$$
\begin{cases}
x_4 = 5 - 2x_1 - 3x_2 - x_3 \\
x_5 = 11 - 4x_1 - x_2 - 2x_3 \\
x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\
z = 5x_1 + 4x_2 + 3x_3
\end{cases}
$$

Invariants:

- Number of equations (m) never changes
- Variables are either on the left-hand side or the right-hand side, never both
  - Left-hand side variables are called *basic*
  - Right-hand side variables are called *non-basic*
- Non-basic variables always pressed against their bounds (always 0)
- Basic variable assignment determined by non-basic assignment and equations

Updating the Equations: Pivoting

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

The set of basic variables is the *basis*

In the pivoting step:

- A non-basic variable enters the basis (the *entering* variable)
- A basic variable leaves the basis (the *leaving* variable)

How is the entering variable chosen? To increase the value of z

One strategy (*Dantzig's rule*) picks the variable with the largest coefficient

How is the leaving variable chosen? To maintain feasibility

Select the basic variable corresponding to the tightest upper-bound

Updating the Equations: Pivoting

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

The set of basic variables is the *basis*

In the pivoting step:

- A non-basic variable enters the basis (the *entering* variable)
- A basic variable leaves the basis (the *leaving* variable)

How is the entering variable chosen? To increase the value of z

One strategy (*Dantzig's rule*) picks the variable with the largest coefficient

How is the leaving variable chosen? To maintain feasibility

Select the basic variable corresponding to the tightest upper-bound

Updating the Equations: Pivoting

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

The set of basic variables is the *basis*

In the pivoting step:

- A non-basic variable enters the basis (the *entering* variable)
- A basic variable leaves the basis (the *leaving* variable)

How is the entering variable chosen? To increase the value of z

One strategy (*Dantzig's rule*) picks the variable with the largest coefficient

How is the leaving variable chosen? To maintain feasibility

Select the basic variable corresponding to the tightest upper-bound

Updating the Equations: Pivoting

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

The set of basic variables is the *basis*

In the pivoting step:

- A non-basic variable enters the basis (the *entering* variable)
- A basic variable leaves the basis (the *leaving* variable)

How is the entering variable chosen? To increase the value of z

One strategy (*Dantzig's rule*) picks the variable with the largest coefficient

How is the leaving variable chosen? To maintain feasibility

Select the basic variable corresponding to the tightest upper-bound

Updating the Equations: Pivoting

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

The set of basic variables is the *basis*

In the pivoting step:

- A non-basic variable enters the basis (the *entering* variable)
- A basic variable leaves the basis (the *leaving* variable)

How is the entering variable chosen? To increase the value of z

One strategy (*Dantzig's rule*) picks the variable with the largest coefficient

How is the leaving variable chosen? To maintain feasibility

Select the basic variable corresponding to the tightest upper-bound

Updating the Equations: Pivoting

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases}$$

The set of basic variables is the *basis*

In the pivoting step:

- A non-basic variable enters the basis (the *entering* variable)
- A basic variable leaves the basis (the *leaving* variable)

How is the entering variable chosen? To increase the value of z

One strategy (*Dantzig's rule*) picks the variable with the largest coefficient

How is the leaving variable chosen? To maintain feasibility

Select the basic variable corresponding to the tightest upper-bound

Tableau and Implementation

We have presented the equation system as a *dictionary*

A more popular version uses a matrix, or a *tableau*:

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases} \longrightarrow$$

x_1	x_2	x_3	x_4	x_5	x_6	z	b
2	3	1	1	0	0	0	5
4	1	2	1	1	0	0	11
3	4	2	1	0	1	0	8
5	4	3	0	0	0	0	0
−5	−4	−3	0	0	0	1	0

The pivoting process can be understood as a series of matrix operations

See [Guoqing Hu] for a description and example

Tableau and Implementation

We have presented the equation system as a *dictionary*

A more popular version uses a matrix, or a *tableau*:

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases} \longrightarrow$$

x_1	x_2	x_3	x_4	x_5	x_6	z	b
2	3	1	1	0	0	0	5
4	1	2	1	1	0	0	11
3	4	2	1	0	1	0	8
5	4	3	0	0	0	0	0
−5	−4	−3	0	0	0	1	0

The pivoting process can be understood as a series of matrix operations

See [Guoqing Hu] for a description and example

Tableau and Implementation

We have presented the equation system as a *dictionary*

A more popular version uses a matrix, or a *tableau*:

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases} \longrightarrow$$

x_1	x_2	x_3	x_4	x_5	x_6	z	b
2	3	1	1	0	0	0	5
4	1	2	1	1	0	0	11
3	4	2	1	0	1	0	8
5	4	3	0	0	0	0	0
−5	−4	−3	0	0	0	1	0

The pivoting process can be understood as a series of matrix operations

See [Guoqing Hu] for a description and example

Tableau and Implementation

We have presented the equation system as a *dictionary*

A more popular version uses a matrix, or a *tableau*:

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3 \end{cases} \longrightarrow$$

x_1	x_2	x_3	x_4	x_5	x_6	z	b
2	3	1	1	0	0	0	5
4	1	2	1	1	0	0	11
3	4	2	1	0	1	0	8
5	4	3	0	0	0	0	0
−5	−4	−3	0	0	0	1	0

The pivoting process can be understood as a series of matrix operations

See [Guoqing Hu] for a description and example

Some Challenges

Possible problems of the procedure that we described so far:

- **Initialization**: how to obtain an initial feasible solution?
- **Termination**: can we generate an infinite sequence of dictionaries, without reaching an optimal z?

Some Challenges

Possible problems of the procedure that we described so far:

Initialization: how to obtain an initial feasible solution?

Termination: can we generate an infinite sequence of dictionaries, without reaching an optimal z?

Challenges: initialization

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad \text{for } i = 1, \dots, m \\ & x_j \ge 0 \quad \text{for } j = 1, \dots, n \end{aligned}$$

Easy when all b_i's are non-negative (set all x_j to 0)

What can we do for negative b_i's?

Challenges: initialization

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for } i = 1, \ldots, m \\ & x_j \geq 0 \quad \text{for } j = 1, \ldots, n \end{aligned}$$

Easy when all b_i's are non-negative (set all x_j to 0)

What can we do for negative b_i's?

Challenges: initialization

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for } i = 1, \ldots, m \\ & x_j \geq 0 \quad \text{for } j = 1, \ldots, n \end{aligned}$$

Easy when all b_i's are non-negative (set all x_j to 0)

What can we do for negative b_i's?

Challenges: initialization

Solution: switch to an auxiliary problem with a known feasible solution

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad \text{for } i = 1, \ldots, m \\ & x_j \ge 0 \quad \text{for } j = 1, \ldots, n \end{aligned}$$

becomes

$$\begin{aligned} \text{maximize} \quad & -x_0 \\ \text{s.t.} \quad & -x_0 + \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad \text{for } i = 1, \ldots, m \\ & x_j \ge 0 \quad \text{for } j = 0, 1, \ldots, n \end{aligned}$$

Challenges: initialization

Solution: switch to an auxiliary problem with a known feasible solution

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for } i = 1, \ldots, m \\ & x_j \geq 0 \quad \text{for } j = 1, \ldots, n \end{aligned}$$

becomes

$$\begin{aligned} \text{maximize} \quad & -x_0 \\ \text{s.t.} \quad & -x_0 + \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for } i = 1, \ldots, m \\ & x_j \geq 0 \quad \text{for } j = 0, 1, \ldots, n \end{aligned}$$

Challenges: initialization

$$\begin{aligned} \text{maximize} \quad & -x_0 \\ \text{s.t.} \quad & -x_0 + \sum_{j=1}^{n} a_{ij}x_j \le b_i \quad \text{for } i = 1, \ldots, m \\ & x_j \ge 0 \quad \text{for } j = 0, 1, \ldots, n \end{aligned}$$

For the auxiliary problem, a feasible solution is easy to find:

set $x_1, \ldots, x_n$ to 0, and make x_0 sufficiently large

Original problem is feasible iff the optimal solution for the auxiliary problem has $x_0 \mapsto 0$

Challenges: initialization

$$\begin{aligned} \text{maximize} \quad & -x_0 \\ \text{s.t.} \quad & -x_0 + \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for } i = 1, \ldots, m \\ & x_j \geq 0 \quad \text{for } j = 0, 1, \ldots, n \end{aligned}$$

For the auxiliary problem, a feasible solution is easy to find:

set $x_1, \ldots, x_n$ to 0, and make x_0 sufficiently large

Original problem is feasible iff the optimal solution for the auxiliary problem has $x_0 \mapsto 0$

Challenges: initialization

$$\begin{aligned} \text{maximize} \quad & -x_0 \\ \text{s.t.} \quad & -x_0 + \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad \text{for } i = 1, \dots, m \\ & x_j \ge 0 \quad \text{for } j = 0, 1, \dots, n \end{aligned}$$

For the auxiliary problem, a feasible solution is easy to find:

set $x_1, \dots, x_n$ to 0, and make x_0 sufficiently large

Original problem is feasible iff the optimal solution for the auxiliary problem has $x_0 \mapsto 0$

Initialization: example

$$
\text{maximize} \quad x_1 + 2x_2 \qquad \text{s.t.} \quad \begin{cases} 2x_1 - 3x_2 \le -2 \\ 4x_1 - x_2 \le -4 \\ x_1, x_2 \ge 0 \end{cases}
\quad \longrightarrow \quad
\text{maximize} \quad -x_0 \qquad \text{s.t.} \quad \begin{cases} 2x_1 - 3x_2 - x_0 \le -2 \\ 4x_1 - x_2 - x_0 \le -4 \\ x_0, x_1, x_2 \ge 0 \end{cases}
$$

The dictionary of the auxiliary problem:

$$
\begin{aligned} x_3 &= -2 - 2x_1 + 3x_2 + x_0 \\ x_4 &= -4 - 4x_1 + x_2 + x_0 \\ z &= -x_0 \end{aligned}
$$

Initial feasible solution: $x_0 \mapsto 4, x_1 \mapsto 0, x_2 \mapsto 0$

Any issues? Variables on the right-hand side need to be 0

Solution: perform a pivot step to move x_0 into the basis

$$
\begin{aligned} x_3 &= 2 + 2x_1 + 2x_2 + x_4 \\ x_0 &= 4 + 4x_1 - x_2 + x_4 \\ z &= -4 - 4x_1 + x_2 - x_4 \end{aligned}
$$

Initialization: example

$$\text{maximize} \quad x_1 + 2x_2 \qquad \text{s.t.} \begin{cases} 2x_1 - 3x_2 & \leq -2 \\ 4x_1 - x_2 & \leq -4 \\ x_1, x_2 & \geq 0 \end{cases}$$

$$\longrightarrow$$

$$\text{maximize} \quad -x_0 \qquad \text{s.t.} \begin{cases} 2x_1 - 3x_2 - x_0 & \leq -2 \\ 4x_1 - x_2 - x_0 & \leq -4 \\ x_0, x_1, x_2 & \geq 0 \end{cases}$$

The dictionary of the auxiliary problem:

$$\begin{aligned} x_3 &= -2 - 2x_1 + 3x_2 + x_0 \\ x_4 &= -4 - 4x_1 + x_2 + x_0 \\ z &= -x_0 \end{aligned}$$

Initial feasible solution: $x_0 \mapsto 4, x_1 \mapsto 0, x_2 \mapsto 0$

Any issues? Variables on the right-hand side need to be 0

Solution: perform a pivot step to move x_0 into the basis

$$\begin{aligned} x_3 &= 2 + 2x_1 + 2x_2 + x_4 \\ x_0 &= 4 + 4x_1 - x_2 + x_4 \\ z &= -4 - 4x_1 + x_2 - x_4 \end{aligned}$$

Initialization: example

$$\text{maximize} \quad x_1 + 2x_2 \qquad \text{s.t.} \begin{cases} 2x_1 - 3x_2 & \leq -2 \\ 4x_1 - x_2 & \leq -4 \\ x_1, x_2 & \geq 0 \end{cases}$$

$$\longrightarrow$$

$$\text{maximize} \quad -x_0 \qquad \text{s.t.} \begin{cases} 2x_1 - 3x_2 - x_0 & \leq -2 \\ 4x_1 - x_2 - x_0 & \leq -4 \\ x_0, x_1, x_2 & \geq 0 \end{cases}$$

The dictionary of the auxiliary problem:

$$\begin{aligned} x_3 &= -2 - 2x_1 + 3x_2 + x_0 \\ x_4 &= -4 - 4x_1 + x_2 + x_0 \\ z &= -x_0 \end{aligned}$$

Initial feasible solution: $x_0 \mapsto 4, x_1 \mapsto 0, x_2 \mapsto 0$

Any issues? Variables on the right-hand side need to be 0

Solution: perform a pivot step to move x_0 into the basis

$$\begin{aligned} x_3 &= 2 + 2x_1 + 2x_2 + x_4 \\ x_0 &= 4 + 4x_1 - x_2 + x_4 \\ z &= -4 - 4x_1 + x_2 - x_4 \end{aligned}$$

Initialization: example

$$\text{maximize} \quad x_1 + 2x_2 \qquad\qquad \text{maximize} \quad -x_0$$

$$\text{s.t.} \begin{cases} 2x_1 - 3x_2 \leq -2 \\ 4x_1 - x_2 \leq -4 \\ x_1, x_2 \geq 0 \end{cases} \longrightarrow \quad \text{s.t.} \begin{cases} 2x_1 - 3x_2 - x_0 \leq -2 \\ 4x_1 - x_2 - x_0 \leq -4 \\ x_0, x_1, x_2 \geq 0 \end{cases}$$

The dictionary of the auxiliary problem:

$$\begin{aligned} x_3 &= -2 - 2x_1 + 3x_2 + x_0 \\ x_4 &= -4 - 4x_1 + x_2 + x_0 \\ z &= -x_0 \end{aligned}$$

Initial feasible solution: $x_0 \mapsto 4, x_1 \mapsto 0, x_2 \mapsto 0$

Any issues? Variables on the right-hand side need to be 0

Solution: perform a pivot step to move x_0 into the basis

$$\begin{aligned} x_3 &= 2 + 2x_1 + 2x_2 + x_4 \\ x_0 &= 4 + 4x_1 - x_2 + x_4 \\ z &= -4 - 4x_1 + x_2 - x_4 \end{aligned}$$

Initialization: example

$$\text{maximize} \quad x_1 + 2x_2 \qquad \text{s.t.} \begin{cases} 2x_1 - 3x_2 & \leq -2 \\ 4x_1 - x_2 & \leq -4 \\ x_1, x_2 & \geq 0 \end{cases} \quad \longrightarrow \quad \text{maximize} \quad -x_0 \qquad \text{s.t.} \begin{cases} 2x_1 - 3x_2 - x_0 & \leq -2 \\ 4x_1 - x_2 - x_0 & \leq -4 \\ x_0, x_1, x_2 & \geq 0 \end{cases}$$

The dictionary of the auxiliary problem:

$$\begin{aligned} x_3 &= -2 - 2x_1 + 3x_2 + x_0 \\ x_4 &= -4 - 4x_1 + x_2 + x_0 \\ z &= -x_0 \end{aligned}$$

Initial feasible solution: $x_0 \mapsto 4, x_1 \mapsto 0, x_2 \mapsto 0$

Any issues? Variables on the right-hand side need to be 0

Solution: perform a pivot step to move x_0 into the basis

$$\begin{aligned} x_3 &= 2 + 2x_1 + 2x_2 + x_4 \\ x_0 &= 4 + 4x_1 - x_2 + x_4 \\ z &= -4 - 4x_1 + x_2 - x_4 \end{aligned}$$

Initialization: example

$$\text{maximize} \quad x_1 + 2x_2 \qquad\qquad \text{maximize} \quad -x_0$$

$$\text{s.t.} \begin{cases} 2x_1 - 3x_2 & \leq -2 \\ 4x_1 - x_2 & \leq -4 \\ x_1, x_2 & \geq 0 \end{cases} \longrightarrow \quad \text{s.t.} \begin{cases} 2x_1 - 3x_2 - x_0 & \leq -2 \\ 4x_1 - x_2 - x_0 & \leq -4 \\ x_0, x_1, x_2 & \geq 0 \end{cases}$$

The dictionary of the auxiliary problem:

$$\begin{aligned} x_3 &= -2 - 2x_1 + 3x_2 \quad +x_0 \\ x_4 &= -4 - 4x_1 + x_2 \quad +x_0 \\ z &= \quad -x_0 \end{aligned}$$

Initial feasible solution: $x_0 \mapsto 4, x_1 \mapsto 0, x_2 \mapsto 0$

Any issues? Variables on the right-hand side need to be 0

Solution: perform a pivot step to move x_0 into the basis

$$\begin{aligned} x_3 &= 2 + 2x_1 + 2x_2 \quad +x_4 \\ x_0 &= 4 + 4x_1 - x_2 \quad +x_4 \\ z &= -4 - 4x_1 + x_2 \quad -x_4 \end{aligned}$$

Challenges: Termination

Recall the goal of every iteration is to increase the objective function z

In each pivoting step, we swap a non-basic variable with a basic variable:

- The non-basic (entering) variable has a positive coefficient in the objective function
- If no such variable exists, the objective function is optimal and we can stop
- The leaving variable is the one imposing the tightest constraint

An iteration will never make z worse

So when might we not converge to the optimal z?

Challenges: Termination

Recall the goal of every iteration is to increase the objective function z

In each pivoting step, we swap a non-basic variable with a basic variable:

- The non-basic (entering) variable has a positive coefficient in the objective function
- If no such variable exists, the objective function is optimal and we can stop
- The leaving variable is the one imposing the tightest constraint

An iteration will never make z worse

So when might we not converge to the optimal z?

Challenges: Termination

Recall the goal of every iteration is to increase the objective function z

In each pivoting step, we swap a non-basic variable with a basic variable:

- The non-basic (entering) variable has a positive coefficient in the objective function
- If no such variable exists, the objective function is optimal and we can stop
- The leaving variable is the one imposing the tightest constraint

An iteration will never make z worse

So when might we not converge to the optimal z?

Challenges: Termination

Recall the goal of every iteration is to increase the objective function z

In each pivoting step, we swap a non-basic variable with a basic variable:

- The non-basic (entering) variable has a positive coefficient in the objective function
- If no such variable exists, the objective function is optimal and we can stop
- The leaving variable is the one imposing the tightest constraint

An iteration will never make z worse

So when might we not converge to the optimal z?

Challenges: Termination

Recall the goal of every iteration is to increase the objective function z

In each pivoting step, we swap a non-basic variable with a basic variable:

- The non-basic (entering) variable has a positive coefficient in the objective function
- If no such variable exists, the objective function is optimal and we can stop
- The leaving variable is the one imposing the tightest constraint

An iteration will never make z worse

So when might we not converge to the optimal z?

Challenges: Termination

Theorem 1
The simplex method fails to terminate iff it cycles, i.e., it generates the same dictionary infinitely often.

Proof sketch:

1. There are only finitely many bases;
2. each bases uniquely defines the dictionary;
3. therefore, there are only finitely many values of z to try

If Simplex is cycling, then z has to stop increasing

Challenges: Termination

Theorem 1

The simplex method fails to terminate iff it cycles, i.e., it generates the same dictionary infinitely often.

Proof sketch:

1. There are only finitely many bases;
2. each bases uniquely defines the dictionary;
3. therefore, there are only finitely many values of z to try

If Simplex is cycling, then z has to stop increasing

Challenges: Termination

Theorem 1

The simplex method fails to terminate iff it cycles, i.e., it generates the same dictionary infinitely often.

Proof sketch:

1. There are only finitely many bases;
2. each bases uniquely defines the dictionary;
3. therefore, there are only finitely many values of z to try

If Simplex is cycling, then z has to stop increasing

Degenerate Pivots

Example: Current feasible solution: $x_1 \mapsto 0, x_2 \mapsto 0, x_3 \mapsto 0, x_4 \mapsto 0$

$$
\begin{aligned}
x_1 &= -2x_2 + 3x_3 \\
z &= 5x_2 - x_3 + 4x_4
\end{aligned}
$$

Dantzig's rule: pick x_2 as the entering variable

Leaving variable is x_1, but the highest x_2 can be is 0

So the value of z does not change after switching x_1 and x_2

A pivot is *degenerate* if it does not change the objective value

Cycling can only occur in the presence of a degenerate pivot

Note: Degenerate pivots are empirically rare

Degenerate Pivots

Example: Current feasible solution: $x_1 \mapsto 0, x_2 \mapsto 0, x_3 \mapsto 0, x_4 \mapsto 0$

$$
\begin{aligned}
x_1 &= -2x_2 + 3x_3 \\
z &= 5x_2 - x_3 + 4x_4
\end{aligned}
$$

Dantzig's rule: pick x_2 as the entering variable

Leaving variable is x_1, but the highest x_2 can be is 0

So the value of z does not change after switching x_1 and x_2

A pivot is *degenerate* if it does not change the objective value

Cycling can only occur in the presence of a degenerate pivot

Note: Degenerate pivots are empirically rare

Degenerate Pivots

Example: Current feasible solution: $x_1 \mapsto 0, x_2 \mapsto 0, x_3 \mapsto 0, x_4 \mapsto 0$

$$\begin{aligned} x_1 &= -2x_2 + 3x_3 \\ z &= 5x_2 - x_3 + 4x_4 \end{aligned}$$

Dantzig's rule: pick x_2 as the entering variable

Leaving variable is x_1, but the highest x_2 can be is 0

So the value of z does not change after switching x_1 and x_2

A pivot is *degenerate* if it does not change the objective value

Cycling can only occur in the presence of a degenerate pivot

Note: Degenerate pivots are empirically rare

Degenerate Pivots

Example: Current feasible solution: $x_1 \mapsto 0, x_2 \mapsto 0, x_3 \mapsto 0, x_4 \mapsto 0$

$$\begin{aligned} x_1 &= -2x_2 + 3x_3 \\ z &= 5x_2 - x_3 + 4x_4 \end{aligned}$$

Dantzig's rule: pick x_2 as the entering variable

Leaving variable is x_1, but the highest x_2 can be is 0

So the value of z does not change after switching x_1 and x_2

A pivot is *degenerate* if it does not change the objective value

Cycling can only occur in the presence of a degenerate pivot

Note: Degenerate pivots are empirically rare

Degenerate Pivots

Example: Current feasible solution: $x_1 \mapsto 0, x_2 \mapsto 0, x_3 \mapsto 0, x_4 \mapsto 0$

$$
\begin{aligned}
x_1 &= -2x_2 + 3x_3 \\
z &= 5x_2 - x_3 + 4x_4
\end{aligned}
$$

Dantzig's rule: pick x_2 as the entering variable

Leaving variable is x_1, but the highest x_2 can be is 0

So the value of z does not change after switching x_1 and x_2

A pivot is *degenerate* if it does not change the objective value

Cycling can only occur in the presence of a degenerate pivot

Note: Degenerate pivots are empirically rare

Degenerate Pivots

Example: Current feasible solution: $x_1 \mapsto 0, x_2 \mapsto 0, x_3 \mapsto 0, x_4 \mapsto 0$

$$\begin{aligned} x_1 &= -2x_2 + 3x_3 \\ z &= 5x_2 - x_3 + 4x_4 \end{aligned}$$

Dantzig's rule: pick x_2 as the entering variable

Leaving variable is x_1, but the highest x_2 can be is 0

So the value of z does not change after switching x_1 and x_2

A pivot is *degenerate* if it does not change the objective value

Cycling can only occur in the presence of a degenerate pivot

Note: Degenerate pivots are empirically rare

Pivoting Strategies

There are variable selection strategies that guarantee termination

Bland's Rule (1977): the simplex method terminates as long as the entering and leaving variables are selected by the smallest-subscript rule in each iteration

Pivoting Strategies

There are variable selection strategies that guarantee termination

Bland's Rule (1977): the simplex method terminates as long as the entering and leaving variables are selected by the smallest-subscript rule in each iteration

Example: $z = -5x_1 - 3x_2 + 4x_3 + 40x_4$

The entering variable is: x_3

Leaving variable: still the one imposing the tightest constraint, but break tie by picking the smaller subscript

Pivoting Strategies

There are variable selection strategies that guarantee termination

Bland's Rule (1977): the simplex method terminates as long as the entering and leaving variables are selected by the smallest-subscript rule in each iteration

Modern solvers use more sophisticated heuristics (e.g., Steepest Edge) that might not prevent cycling

Pivoting Strategies

There are variable selection strategies that guarantee termination

Bland's Rule (1977): the simplex method terminates as long as the entering and leaving variables are selected by the smallest-subscript rule in each iteration

Modern solvers use more sophisticated heuristics (e.g., Steepest Edge) that might not prevent cycling

When cycling is detected: switch to Bland's rule for a while

Pivoting Strategies

There are variable selection strategies that guarantee termination

Bland's Rule (1977): the simplex method terminates as long as the entering and leaving variables are selected by the smallest-subscript rule in each iteration

Modern solvers use more sophisticated heuristics (e.g., Steepest Edge) that might not prevent cycling

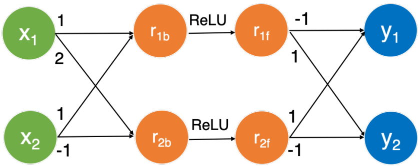
When cycling is detected: switch to Bland's rule for a while

Complexity: the common strategies all have worse-case exponential time

Possible improvements

- More sophisticated pivoting strategy
- Use rational-number instead of floating-point representation
  (to handle numerical instability and avoid solutions unsoundness)
- Handle general Linear Programs
  (variables can have non-zero lower bounds and/or finite upper bounds)
- Extract irreducible infeasible subset in case of infeasibility
  (theory explanations)
- …

Application: Neural Network Verification



Property to verify: $\forall x_1 . x_2 . (x_1 \in [-2, 1] \wedge x_2 \in [-2, 2] \Rightarrow y_1 < y_2)$

1. Encoding of the neural network α_n (linear + Rectified Linear Units):

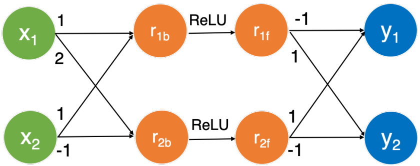
$$r_{1b} = x_1 + x_2 \quad r_{2b} = 2x_1 - x_2 \qquad (r_{1b} \leq 0 \wedge r_{1f} = 0) \vee (r_{1b} \geq 0 \wedge r_{1f} = r_{1b})$$

$$y_1 = -r_{1f} + r_{2f} \quad y_2 = r_{1f} - r_{2f} \qquad (r_{2b} \leq 0 \wedge r_{2f} = 0) \vee (r_{2b} \geq 0 \wedge r_{2f} = r_{2b})$$

2. Encoding of the the property α_p: $\quad -2 \leq x_1 \leq 1 \quad -2 \leq x_2 \leq 2 \quad y_1 >= y_2$

3. Property holds iff $\alpha_n \wedge \alpha_p$ is unsatisfiable

Application: Neural Network Verification



Property to verify: $\forall x_1 . x_2 . (x_1 \in [-2, 1] \land x_2 \in [-2, 2] \Rightarrow y_1 < y_2)$

1. Encoding of the neural network α_n (linear + Rectified Linear Units):

$$r_{1b} = x_1 + x_2 \quad r_{2b} = 2x_1 - x_2 \qquad (r_{1b} \leq 0 \land r_{1f} = 0) \lor (r_{1b} \geq 0 \land r_{1f} = r_{1b})$$

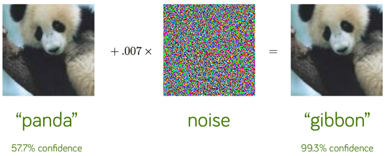
$$y_1 = -r_{1f} + r_{2f} \quad y_2 = r_{1f} - r_{2f} \qquad (r_{2b} \leq 0 \land r_{2f} = 0) \lor (r_{2b} \geq 0 \land r_{2f} = r_{2b})$$

2. Encoding of the the property α_p: $\quad -2 \leq x_1 \leq 1 \quad -2 \leq x_2 \leq 2 \quad y_1 >= y_2$

3. Property holds iff $\alpha_n \land \alpha_p$ is unsatisfiable

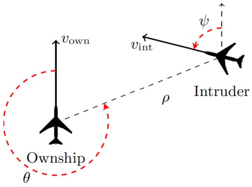
Practical properties

Robustness: $\forall x'.\ \|\mathbf{x} - x'\| < \epsilon \Rightarrow \|N(\mathbf{x}) - N(x')\| < \delta$



There is no adversarial input within ϵ distance

Reachability: $\forall x.\ x \in [x_l, x_u] \Rightarrow y \in [y_l, y_u]$

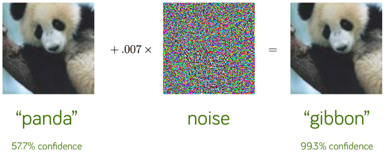


Whenever intruder is near and to the right advise strong left

A lot of attention in recent years

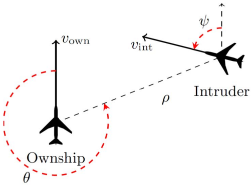
Practical properties

Robustness: $\forall x'.\ \|\mathbf{x} - x'\| < \epsilon \Rightarrow \|N(\mathbf{x}) - N(x')\| < \delta$



There is no adversarial input within ϵ *distance*

Reachability: $\forall x.\ x \in [x_l, x_u] \Rightarrow y \in [y_l, y_u]$



Whenever intruder is near and to the right advise strong left

A lot of attention in recent years