Linear Programming and Its Applications

Numeric Programming

Fred Torres Cruz

Computer Science Engineering and Statistics Department
Universidad Nacional del Altiplano de Puno

September 4, 2025

Course Overview

What is Linear Programming?

- A class of optimization problems
- Linear objective function
- Linear constraints
- Powerful mathematical tool for resource allocation
- Applications across multiple disciplines

The Three Essential Questions

1. **What** is linear programming?
2. **Why** should we learn it?
3. **How** are we going to learn it?

Optimization: The Core Concept

General Optimization Problem

Maximize or minimize a function $f(x)$
Subject to constraints $g_i(x) \leq 0$, $h_j(x) = 0$

Most real-world problems are optimization problems!

Why Linear Programming?

Three key properties make LP interesting:

1. **Solvable:** Efficient algorithms exist
2. **Useful:** Many real applications
3. **Beautiful:** Rich mathematical structure (duality theory)

Application Areas

- Resource allocation
- Portfolio management
- Production optimization
- Network design
- Machine learning
- Complexity theory
- Transportation
- Manufacturing

The Brewery Problem

A brewery produces two types of beer:

- Dark beer
- Light beer

Each requires different amounts of:

- Corn
- Hops
- Barley

Resource Requirements

Resource	Dark Beer (per liter)	Light Beer (per liter)
Corn	5 kg	15 kg
Hops	4 g	4 g
Barley	35 kg	20 kg
Profit	\$15	\$20

Inventory Constraints

Available resources:

- Corn: 480 kg
- Hops: 160 g
- Barley: 1190 kg

Question: How to maximize profit?

Mathematical Formulation

Let x = liters of dark beer, y = liters of light beer

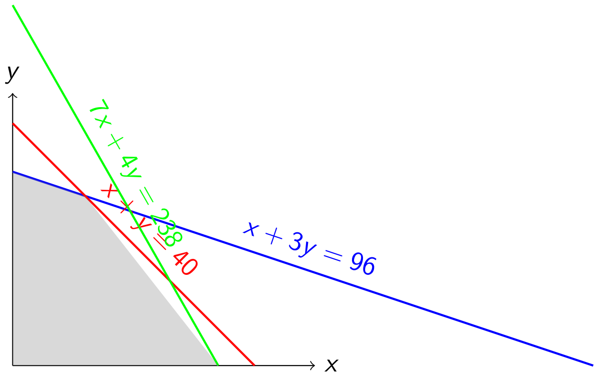
$$\begin{aligned}
\text{Maximize: } & 15x + 20y && (1)\\
\text{Subject to: } & 5x + 15y \leq 480 \text{ (corn)} && (2)\\
& 4x + 4y \leq 160 \text{ (hops)} && (3)\\
& 35x + 20y \leq 1190 \text{ (barley)} && (4)\\
& x, y \geq 0 && (5)
\end{aligned}$$

Graphical Method: Step 1

Convert inequalities to equalities for boundary lines:

- $5x + 15y = 480$ simplifies to $x + 3y = 96$
- $4x + 4y = 160$ simplifies to $x + y = 40$
- $35x + 20y = 1190$ simplifies to $7x + 4y = 238$

Graphical Method: Step 2



The shaded region is the feasible set.

Key Observation

The optimal solution occurs at a **vertex** of the feasible region!
This fundamental property allows us to:

- Check only finitely many points
- Avoid searching the entire region

Finding Vertices

Vertices are intersections of constraint boundaries:

1. $(0, 0)$: Origin
2. $(0, 32)$: y-axis and corn constraint
3. $(12, 28)$: Corn and hops constraints
4. $(34, 0)$: x-axis and barley constraint

Evaluating Profit at Vertices

Vertex (x, y)	Profit $= 15x + 20y$
$(0, 0)$	\$0
$(0, 32)$	\$640
$(12, 28)$	\$740
$(34, 0)$	\$510

Optimal solution: 12 liters dark, 28 liters light beer

Why Graphical Method Isn't Enough

Real-world problems often have:

- Thousands of variables
- Thousands of constraints
- Cannot visualize in high dimensions

We need algebraic methods!

General Linear Program

$$
\begin{aligned}
\text{Maximize:} \quad & c_1x_1 + c_2x_2 + \cdots + c_nx_n && (6)\\
\text{Subject to:} \quad & a_{11}x_1 + a_{12}x_2 + \cdots \leq b_1 && (7)\\
& a_{21}x_1 + a_{22}x_2 + \cdots \geq b_2 && (8)\\
& \vdots && (9)
\end{aligned}
$$

Mix of $\leq$, $\geq$, and $=$ constraints allowed.

Matrix Notation

Compact representation using matrices:

- $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ (variables)
- $\mathbf{c} = [c_1, c_2, \ldots, c_n]^T$ (costs)
- $\mathbf{A}$ = constraint coefficient matrix
- $\mathbf{b}$ = constraint bounds

Standard Form

Every LP can be converted to:

$$\text{Maximize: } \mathbf{c}^T\mathbf{x} \tag{10}$$

$$\text{Subject to: } \mathbf{A}\mathbf{x} = \mathbf{b} \tag{11}$$

$$\mathbf{x} \geq \mathbf{0} \tag{12}$$

Only equality constraints and non-negativity!

Converting to Standard Form

Techniques for conversion:

- Inequality $\leq$: Add slack variable
- Inequality $\geq$: Subtract surplus variable
- Unrestricted variable: Replace with difference of two non-negative variables

Slack Variables Example

Original: $5x + 15y \leq 480$
Add slack variable $s_1 \geq 0$:

$$5x + 15y + s_1 = 480$$

Physical interpretation: $s_1 =$ unused corn

Why Study $\mathbf{Ax} = \mathbf{b}$?

The feasible region is defined by:

- $\mathbf{Ax} = \mathbf{b}$ (linear equations)
- $\mathbf{x} \geq \mathbf{0}$ (non-negativity)

Understanding linear equations is fundamental to LP!

Solution Methods Timeline

- **Simplex Method** (Dantzig, 1947)
  - Not polynomial in worst case
  - Excellent in practice
- **Ellipsoid Algorithm** (Khachikyan, 1979)
  - First polynomial algorithm
  - Poor practical performance
- **Interior Point Methods** (Karmarkar, 1984)
  - Polynomial and practical

Solving Linear Equations

Central problem in linear algebra:

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

where:

- $\mathbf{A}$ is $m \times n$ matrix
- $\mathbf{x}$ is $n \times 1$ vector (unknowns)
- $\mathbf{b}$ is $m \times 1$ vector (constants)

Example System

$$2x + y = 4 \tag{13}$$
$$x + y = 3 \tag{14}$$

Simple observation: Subtracting second from first gives $x = 1$. Back-substitution yields $y = 2$.

Gaussian Elimination Operations

Three elementary row operations:

1. Multiply row by non-zero scalar
2. Add/subtract multiple of one row to another
3. Exchange two rows

These preserve the solution set!

Goal: Upper Triangular Form

Transform system to "good form":

$$\begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & * \end{bmatrix}$$

Then solve by back-substitution.

Three-Variable Example

$$2x + y + z = 5 \tag{15}$$
$$4x - 6y + 0z = -2 \tag{16}$$
$$-2x + 7y + 2z = 9 \tag{17}$$

Goal: Eliminate x from equations 2 and 3.

Step 1: Normalize Leading Coefficient

Divide first equation by 2:

$$x + \frac{1}{2}y + \frac{1}{2}z = \frac{5}{2} \tag{18}$$

$$4x - 6y + 0z = -2 \tag{19}$$

$$-2x + 7y + 2z = 9 \tag{20}$$

Step 2: Eliminate First Column

- $E_2 - 4E_1$: Eliminates x from equation 2
- $E_3 + 2E_1$: Eliminates x from equation 3

Result:

$$x + \frac{1}{2}y + \frac{1}{2}z = \frac{5}{2} \tag{21}$$

$$0 - 8y - 2z = -12 \tag{22}$$

$$0 + 8y + 3z = 14 \tag{23}$$

Step 3: Continue Elimination

Add equations 2 and 3:

$$x + \frac{1}{2}y + \frac{1}{2}z = \frac{5}{2} \tag{24}$$

$$-8y - 2z = -12 \tag{25}$$

$$z = 2 \tag{26}$$

Now in upper triangular form!

Back-Substitution

Working backwards:

- From equation 3: $z = 2$
- Substitute into equation 2: $y = 1$
- Substitute into equation 1: $x = 1$

Solution: $(x, y, z) = (1, 1, 2)$

Possible Outcomes

Gaussian elimination can reveal:

1. **Unique solution:** System in good form
2. **No solution:** Contradictory equations (e.g., $0 = 1$)
3. **Infinite solutions:** Free variables remain

No Solution Example

After elimination:

$$x + y + z = 5 \tag{27}$$
$$3z = -12 \tag{28}$$
$$4z = -16 \tag{29}$$

Last two equations contradict: $z = -4$ and $z = -4$? No!

Infinite Solutions Example

After elimination:

$$x + y + z = 5 \quad (30)$$

$$3z = -12 \quad (31)$$

$$0 = 0 \quad (32)$$

Free variable y; solutions: $(9 - y, y, -4)$ for any y.

Handling Zero Pivots

Problem: Leading coefficient is zero.
Solution strategies:

- Exchange rows (pivot selection)
- Exchange columns (variable reordering)

Always possible unless system is singular.

Alternative View of $\mathbf{Ax} = \mathbf{b}$

Row view (Gaussian elimination):

Set of m linear equations

Column view (vector spaces):

$$x_1\mathbf{v}_1 + x_2\mathbf{v}_2 + \cdots + x_n\mathbf{v}_n = \mathbf{b}$$

where $\mathbf{v}_i$ are columns of $\mathbf{A}$.

Linear Combinations

Definition

A linear combination of vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ is:

$$c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_n\mathbf{v}_n$$

for scalars $c_1, \ldots, c_n$.

Column Space Interpretation

Question: Is $\mathbf{b}$ in the column space of $\mathbf{A}$?
Equivalently: Can we express $\mathbf{b}$ as a linear combination of columns of $\mathbf{A}$?
This is exactly asking if $\mathbf{Ax} = \mathbf{b}$ has a solution!

Linear Independence

Vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are linearly independent if:

$$c_1\mathbf{v}_1 + \cdots + c_n\mathbf{v}_n = \mathbf{0} \implies c_1 = \cdots = c_n = 0$$

Otherwise, they are linearly dependent.

Basis

A basis for a vector space is:

- A set of linearly independent vectors
- That spans the entire space

Number of basis vectors = dimension of space.

Free Variables and Rank

After Gaussian elimination:

- Pivot variables: Determined by the system
- Free variables: Can be chosen arbitrarily
- Rank $=$ number of pivot variables
- Nullity $=$ number of free variables

Rank-Nullity Theorem

For an $m \times n$ matrix $\mathbf{A}$:

$$\text{rank}(\mathbf{A}) + \text{nullity}(\mathbf{A}) = n$$

Fundamental relationship in linear algebra!

Solution Structure

Solutions to $\mathbf{Ax} = \mathbf{b}$:

- If consistent: $\mathbf{x} = \mathbf{x}_p + \mathbf{x}_h$
- $\mathbf{x}_p$: Particular solution
- $\mathbf{x}_h$: Homogeneous solution ($\mathbf{Ax}_h = \mathbf{0}$)

Network Flow Problems

Model flow through networks:

- Nodes: Sources, sinks, junctions
- Edges: Capacity constraints
- Objective: Maximize flow or minimize cost

Applications: Internet routing, transportation, supply chains.

Machine Learning Applications

Linear programming in ML:

- Support Vector Machines (classification)
- Basis pursuit (sparse recovery)
- Robust regression
- Feature selection

Game Theory

Computing Nash equilibria:

- Zero-sum games $\rightarrow$ LP
- Mixed strategies
- Minimax theorem connection

Economics and strategic decision-making.

Approximation Algorithms

LP relaxation technique:

1. Start with hard integer problem
2. Relax to linear program
3. Solve LP (polynomial time)
4. Round solution cleverly

Guarantees approximation ratios.

Primal and Dual Problems

Every LP has an associated dual:

Primal:

$$\begin{aligned} \max \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Dual:

$$\begin{aligned} \min \quad & \mathbf{b}^T\mathbf{y} \\ \text{s.t.} \quad & \mathbf{A}^T\mathbf{y} \geq \mathbf{c} \\ & \mathbf{y} \geq \mathbf{0} \end{aligned}$$

Weak Duality

Theorem (Weak Duality)

If $\mathbf{x}$ is feasible for primal and $\mathbf{y}$ is feasible for dual:

$$\mathbf{c}^T\mathbf{x} \leq \mathbf{b}^T\mathbf{y}$$

Dual provides upper bound on primal!

Strong Duality

Theorem (Strong Duality)

If either primal or dual has optimal solution, then both do, and:

$$\mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^*$$

Optimal values are equal!

Economic Interpretation

Dual variables represent shadow prices:

- Marginal value of resources
- How much objective improves per unit resource increase
- Guides resource allocation decisions

Simplex Method Overview

Key idea: Move from vertex to vertex

1. Start at a vertex (basic feasible solution)
2. Check if current vertex is optimal
3. If not, move to better adjacent vertex
4. Repeat until optimal

Basic Feasible Solutions

For $\mathbf{Ax} = \mathbf{b}$ with m equations, n variables:

- Choose m linearly independent columns (basis)
- Set other $n - m$ variables to zero
- Solve for basic variables

Corresponds to vertex of feasible region.

Optimality Test

At current vertex, compute reduced costs:

$$\bar{c}_j = c_j - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{a}_j$$

If all $\bar{c}_j \leq 0$: Current solution optimal!

Pivoting

If not optimal:

1. Choose entering variable (positive reduced cost)
2. Choose leaving variable (minimum ratio test)
3. Update basis
4. Recompute solution

Tableau Method

Organize computations in tableau:

- Rows: Constraints
- Columns: Variables and RHS
- Bottom row: Reduced costs

Systematic bookkeeping for hand calculations.

Degeneracy

Problem: Multiple constraints active at vertex

- Basic variable $= 0$
- May not improve objective
- Can cause cycling

Solution: Anti-cycling rules (Bland's rule).

Complexity of Simplex

- Worst case: Exponential (Klee-Minty cube)
- Average case: Polynomial (smoothed analysis)
- Practice: Very efficient
- Typical: $2m$ to $3m$ iterations

Interior Point Philosophy

Instead of vertices:

- Stay interior to feasible region
- Follow central path
- Approach optimal vertex

Polynomial worst-case complexity!

Barrier Function

Add logarithmic barrier to objective:

$$\max \quad \mathbf{c}^T\mathbf{x} + \mu \sum_{i=1}^{n} \ln(x_i)$$

As $\mu \to 0$, solution approaches LP optimum.

Central Path

Solutions for different μ values form central path:

- μ large: Deep in interior
- μ small: Near optimal vertex
- Follow path by decreasing μ

Newton's Method

At each μ, solve using Newton's method:

- Compute gradient and Hessian
- Take Newton step
- Stay feasible (step size control)

Quadratic local convergence.

Primal-Dual Methods

Solve primal and dual simultaneously:

- Maintain primal feasibility
- Maintain dual feasibility
- Reduce duality gap
- Most practical approach

Transportation Problem

Ship goods from sources to destinations:

- Sources: Supply constraints
- Destinations: Demand constraints
- Minimize shipping cost

Special structure allows efficient algorithms.

Assignment Problem

Assign n workers to n tasks:

- Each worker: One task
- Each task: One worker
- Minimize total cost

Solvable by Hungarian algorithm.

Network Flow

Max flow problem:

$$\max \quad \text{flow from source to sink}$$

Subject to:

- Capacity constraints on edges
- Flow conservation at nodes

Minimum Cost Flow

Generalization combining:

- Flow requirements
- Capacity constraints
- Edge costs

Many problems reduce to this form.

Numerical Stability

Challenges in practice:

- Floating-point arithmetic
- Round-off errors accumulate
- Ill-conditioned matrices

Need careful implementation.

Scaling

Improve numerical behavior:

- Row scaling: Normalize constraints
- Column scaling: Normalize variables
- Equilibration algorithms

Critical for large problems.

Sparsity

Real problems are sparse:

- Most coefficients zero
- Store only non-zeros
- Sparse matrix techniques
- Huge memory savings

Preprocessing

Simplify before solving:

- Remove redundant constraints
- Fix variables at bounds
- Tighten bounds
- Eliminate variables

Can dramatically reduce size.

Warm Starting

Use previous solution:

- Slightly modified problems
- Sensitivity analysis
- Branch and bound

Much faster than cold start.

Commercial Solvers

Industry standard packages:

- CPLEX (IBM)
- Gurobi
- XPRESS
- MOSEK

Highly optimized, parallel, robust.

Open Source Solvers

Free alternatives:

- GLPK (GNU Linear Programming Kit)
- CLP (COIN-OR)
- lp_solve
- OR-Tools (Google)

Good for education and research.

Modeling Languages

High-level problem specification:

- AMPL
- GAMS
- JuMP (Julia)
- PuLP (Python)

Separate model from data and solver.

Integer Programming

Add integrality constraints:

$$x_i \in \mathbb{Z} \text{ for some } i$$

- Much harder (NP-hard)
- Branch and bound
- Cutting planes
- Many practical applications

Quadratic Programming

Quadratic objective: $\min \quad \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x}$ Linear constraints still.

- Portfolio optimization
- Least squares problems
- Support vector machines

Stochastic Programming

Uncertainty in parameters:

- Two-stage problems
- Recourse decisions
- Scenario analysis
- Risk measures

Semidefinite Programming

Matrix variables with PSD constraint: $\mathbf{X} \succeq 0$ Applications:

- Control theory
- Combinatorial optimization
- Relaxations

Robust Optimization

Handle uncertainty sets: $\min_{\mathbf{x}} \max_{\mathbf{u} \in \mathcal{U}} \mathbf{c}^T \mathbf{x}$

- Worst-case optimization
- Uncertainty sets
- Tractable reformulations

Post-Optimality Analysis

After solving, understand solution stability:

- How sensitive to parameter changes?
- Which constraints are binding?
- Shadow prices and reduced costs

Objective Coefficient Changes

How much can c_j change without changing basis?

- Allowable increase
- Allowable decrease
- Range of optimality

Within range, same solution optimal.

Right-Hand Side Changes

How much can b_i change?

- Shadow price valid range
- Basis remains feasible
- Linear change in objective

Critical for resource planning.

Adding Constraints

New constraint effects:

- May cut off current solution
- Check feasibility
- Resolve if needed
- Warm start helps

Adding Variables

New variable (column) effects:

- Compute reduced cost
- If negative, may improve
- Pricing in column generation

Problem Formulation Tips

Good formulations matter:

- Avoid big-M when possible
- Scale variables appropriately
- Use tight bounds
- Exploit problem structure

Debugging LP Models

Common issues:

- Infeasibility: Check constraints carefully
- Unboundedness: Missing constraints?
- Wrong solution: Formulation error?
- Numerical issues: Scaling needed?

Performance Tuning

Speed up solutions:

- Choose right algorithm
- Tune solver parameters
- Provide good starting point
- Use appropriate tolerances

Large-Scale Problems

Special techniques needed:

- Decomposition methods
- Column generation
- Constraint generation
- Parallel computing

Airline Crew Scheduling

Assign crews to flights:

- Thousands of flights
- Union rules
- Rest requirements
- Minimize cost

Saves millions annually.

Supply Chain Optimization

Coordinate entire supply chain:

- Production planning
- Inventory management
- Distribution routing
- Demand forecasting

Energy Grid Management

Optimize power generation:

- Multiple power sources
- Demand fluctuations
- Transmission constraints
- Environmental regulations

Financial Portfolio Optimization

Markowitz model:

- Risk-return tradeoff
- Diversification constraints
- Transaction costs
- Regulatory requirements

Machine Learning Integration

LP in modern ML:

- Neural network verification
- Adversarial robustness
- Interpretable models
- Structured prediction

Quantum Linear Programming

Quantum algorithms for LP:

- Potential speedups
- Quantum interior point
- Still experimental
- Hardware limitations

Distributed Optimization

Solving across multiple machines:

- Decomposition strategies
- Communication efficiency
- Privacy preservation
- Edge computing

Online Linear Programming

Decisions without complete information:

- Streaming data
- Competitive analysis
- Regret bounds
- Applications in operations

Key Takeaways: Theory

- LP: Optimizing linear functions with linear constraints
- Geometric insight: Solutions at vertices
- Algebraic tools: Gaussian elimination essential
- Duality: Every LP has dual with equal optimum

Key Takeaways: Algorithms

- Simplex: Practical workhorse algorithm
- Interior point: Polynomial guarantee
- Preprocessing: Crucial for efficiency
- Special structure: Exploit when possible