The Simplex Method: Theory, Algorithms, and Applications

A Introduction

This document provides a comprehensive overview of the Simplex Method, a foundational algorithm in linear programming. It explores the mathematical depths of the method, covering problem formulation, core algorithmic concepts, pivot operations, duality theory, and computational considerations. The discussion emphasizes the theoretical underpinnings and practical implications of the Simplex Method. The Simplex Method is a cornerstone of optimization theory, widely used in various fields such as operations research, economics, and engineering. It transforms linear programming problems into a series of manageable steps, allowing for efficient computation of optimal solutions.

B Linear Programming (LP) Basics

Linear Programming (LP) is a mathematical method for determining a way to achieve the best outcome in a given mathematical model. Its function and constraints are linear, and it is used to maximize or minimize a linear objective function subject to linear equality and inequality constraints.

B.1 Standard Form

A linear programming problem in standard form is expressed as:

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

where:

- $x \in \mathbb{R}^n$ is the vector of variables.
- $c \in \mathbb{R}^n$ is the cost vector.
- $A \in \mathbb{R}^{m \times n}$ is the constraint matrix.
- $b \in \mathbb{R}^m$ is the right-hand side vector.

The goal is to find a vector x that minimizes the objective function $c^T x$ while satisfying the constraints given by $Ax = b$ and $x \geq 0$.

B.2 Converting Inequalities to Equalities

Inequality constraints can be transformed into equalities by introducing additional variables:

- For a "less than or equal to" constraint ($\sum a_{ij}x_j \leq b_i$), add a non-negative **slack variable** ($s_i \geq 0$), converting it to $\sum a_{ij}x_j + s_i = b_i$.
- For a "greater than or equal to" constraint ($\sum a_{ij}x_j \geq b_i$), subtract a non-negative **surplus variable** ($s_i \geq 0$), converting it to $\sum a_{ij}x_j - s_i = b_i$. If an initial basic feasible solution is not available, an artificial variable may also be introduced.

The objective function can be either minimized or maximized by adjusting the sign of the cost vector c. If the problem is to maximize, it can be converted to a minimization problem by negating the cost vector:

$$\text{maximize} \quad c^T x \quad \Leftrightarrow \quad \text{minimize} \quad -c^T x$$

Example

Consider a simple LP problem:

$$\begin{aligned} \text{minimize} \quad & 2x_1 + x_2 + 3x_3 + x_4 \\ \text{subject to} \quad & x_1 + 2x_2 + x_3 \geq 5 \\ & 2x_1 - x_2 + 3x_4 = 7 \\ & x_2 + x_3 + x_4 \leq 4 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

This problem can be converted to standard form by introducing slack and surplus variables as needed:

- Surplus variable $s_1 \geq 0$ for the first ($\geq$) constraint,
- Slack variable $s_2 \geq 0$ for the third ($\leq$) constraint.

The standard form becomes:

$$\begin{aligned} \text{minimize} \quad & 2x_1 + x_2 + 3x_3 + x_4 \\ \text{subject to} \quad & x_1 + 2x_2 + x_3 - s_1 = 5 \\ & 2x_1 - x_2 + 3x_4 = 7 \\ & x_2 + x_3 + x_4 + s_2 = 4 \\ & x_1, x_2, x_3, x_4, s_1, s_2 \geq 0 \end{aligned}$$

In this example, the objective function is to minimize the cost associated with the variables x_1, x_2, x_3, x_4, while satisfying the constraints that define the feasible region of the problem. The introduction of slack and surplus variables allows us to convert inequalities into equalities, which is a key step in applying the Simplex Method.

B.3 Polyhedron

Definition 1 (Polyhedron)**.** *A* polyhedron *is the set of solutions to a finite system of linear inequalities and equalities. Formally, a set $P \subseteq \mathbb{R}^n$ is a polyhedron if there exist matrices $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{p\times n}$, and vectors $b \in \mathbb{R}^m$, $d \in \mathbb{R}^p$ such that*

$$P = \{x \in \mathbb{R}^n : Ax \leq b,\ Bx = d\}.$$

In the context of linear programming, the feasible region defined by $Ax = b$, $x \geq 0$ is a polyhedron.

B.4 Feasible Region

Definition 2 (Feasible Region)**.** *The feasible region of an LP is the set of all points that satisfy the constraints. It is a convex polyhedron in $\mathbb{R}^n$. The vertices (or extreme points) of this polyhedron correspond to basic feasible solutions (BFS) of the LP.*

Theorem 1. *The feasible region of a linear program, defined by $Ax = b$, $x \geq 0$, is a convex set.*

Proof. Let $x^{(1)}$ and $x^{(2)}$ be any two feasible solutions, i.e., $Ax^{(1)} = b$, $x^{(1)} \geq 0$ and $Ax^{(2)} = b$, $x^{(2)} \geq 0$. For any $\lambda \in [0, 1]$, consider the convex combination $x = \lambda x^{(1)} + (1 - \lambda)x^{(2)}$.
We have:

$$\begin{aligned} Ax &= A(\lambda x^{(1)} + (1-\lambda)x^{(2)}) \\ &= \lambda A x^{(1)} + (1-\lambda) A x^{(2)} \\ &= \lambda b + (1-\lambda) b = b \end{aligned}$$

Also, since $x^{(1)} \geq 0$ and $x^{(2)} \geq 0$, x is a non-negative combination of non-negative vectors, so $x \geq 0$.
Therefore, x is feasible. Since any convex combination of feasible points is feasible, the feasible region is convex. □

B.5 Basic Solutions and Extreme Points

In the context of linear programming, a basic solution is a solution that corresponds to a vertex of the feasible region. The Simplex Method operates on these basic solutions, moving from one vertex to another along the edges of the feasible region.

Definition 3 (Extreme Point)**.** *A point $x^* \in \mathbb{R}^n$ is called an* extreme point *(or vertex) of the feasible region $\{x \in \mathbb{R}^n : Ax = b,\, x \geq 0\}$ if x^* cannot be written as a convex combination of two distinct feasible points. That is, if $x^* = \lambda x^{(1)} + (1-\lambda)x^{(2)}$ for some $x^{(1)}, x^{(2)}$ feasible and $0 < \lambda < 1$, then $x^{(1)} = x^{(2)} = x^*$.*

Definition 4 (Basic Solution)**.** *Given a system $Ax = b$ with $A \in \mathbb{R}^{m\times n}$ of rank m ($m < n$), a vector $x \in \mathbb{R}^n$ is called a* basic solution *if it is obtained by selecting m linearly independent columns of A (forming a basis B), setting the remaining $n-m$ variables to zero (the non-basic variables), and solving $Bx_B = b$ for the basic variables x_B.*

Theorem 2. *Every basic solution of $Ax = b$ corresponds to an extreme point (vertex) of the feasible region $\{x \in \mathbb{R}^n : Ax = b,\, x \geq 0\}$.*

Proof. Let x^* be a basic solution. Suppose, for contradiction, that x^* is not an extreme point. Then x^* can be written as a nontrivial convex combination $x^* = \lambda x^{(1)} + (1-\lambda)x^{(2)}$ for some $x^{(1)} \neq x^{(2)}$ in the feasible region and $0 < \lambda < 1$.
Since $Ax^* = b$ and A is linear, $Ax^{(1)} = Ax^{(2)} = b$. Also, $x^* \geq 0$ implies $x^{(1)}, x^{(2)} \geq 0$.
By the construction of a basic solution, x^* has at most m nonzero components, corresponding to the basis B. For the non-basic variables, $x_j^* = 0$, so $x_j^{(1)} = x_j^{(2)} = 0$ (otherwise, their convex combination would be positive). Thus, $x^{(1)}$ and $x^{(2)}$ have nonzero entries only in the positions corresponding to B.
But B is invertible, so the equation $Bx_B = b$ has a unique solution. Therefore, $x^{(1)} = x^{(2)} = x^*$, contradicting the assumption that $x^{(1)} \neq x^{(2)}$. Thus, x^* is an extreme point. □

B.6 Optimal Solutions

Definition 5 (Optimal Solution)**.** *An* optimal solution *to a linear programming problem is a feasible solution x^* (i.e., $Ax^* = b$, $x^* \geq 0$) that achieves the best possible value of the objective function. For a minimization problem, x^* is optimal if $c^T x^* \leq c^T x$ for all feasible x; for a maximization problem, x^* is optimal if $c^T x^* \geq c^T x$ for all feasible x.*

Existence and Uniqueness of Optimal Solutions A linear program may have:

- **No optimal solution:** if the feasible region is empty (infeasible problem) or if the objective function is unbounded over the feasible region.
- **A unique optimal solution:** if the optimal value is achieved at a single vertex (extreme point) of the feasible region.
- **Multiple optimal solutions:** if the objective function is constant along an edge or face of the feasible region, so that more than one point (possibly infinitely many) achieves the optimal value.

Theorem 3 (Alternative Optimal Solutions)**.** *If there are multiple optimal solutions to a linear program, then any convex combination of optimal extreme points is also optimal. This occurs when the objective function is parallel to a constraint boundary over a face of the feasible region.*

Proof. Let $x^{(1)}$ and $x^{(2)}$ be two distinct optimal extreme points, i.e., $c^T x^{(1)} = c^T x^{(2)} = z^*$ and both are feasible. For any $\lambda \in [0,1]$, the convex combination $x = \lambda x^{(1)} + (1-\lambda)x^{(2)}$ is also feasible, since the feasible region is convex. The objective value at x is $c^T x = \lambda c^T x^{(1)} + (1-\lambda)c^T x^{(2)} = \lambda z^* + (1-\lambda)z^* = z^*$. Thus, x is also optimal. This situation arises when the objective function is constant along a face (edge or higher-dimensional face) of the feasible region, i.e., when it is parallel to a constraint boundary. □

Theorem 4. *If an optimal solution exists for a linear programming problem, then at least one optimal solution is an extreme point (vertex) of the feasible region.*

Proof. Let $P = \{x \in \mathbb{R}^n : Ax = b,\, x \geq 0\}$ be the feasible region, which is a convex polyhedron. Suppose x^* is an optimal solution, i.e., $c^T x^* \leq c^T x$ for all $x \in P$ (for minimization). Consider the set $F = \{x \in P : c^T x = c^T x^*\}$, which is a face of P containing all optimal solutions.
If x^* is not an extreme point, then it can be written as a convex combination $x^* = \lambda x^{(1)} + (1-\lambda)x^{(2)}$ for some $x^{(1)}, x^{(2)} \in F$, $x^{(1)} \neq x^{(2)}$, and $0 < \lambda < 1$. If $x^{(1)}$ or $x^{(2)}$ is not an extreme point, repeat this process. Since P is a polyhedron, this process terminates in finitely many steps at an extreme point in F. Thus, there exists an extreme point that is optimal. □

Unboundedness If the feasible region is non-empty and the objective function can be decreased (for minimization) indefinitely without violating any constraints, the LP is said to be unbounded and has no finite optimal solution.

C Mathematical Foundations of the Simplex Method

The Simplex Method, developed by George Dantzig in 1947, is a systematic procedure for solving LP problems. It iteratively moves along the edges of the feasible region (polyhedron) defined by the constraints, from one basic feasible solution (BFS) to another, improving the objective function at each step until an optimal solution is reached or unboundedness is detected.

C.1 Overview

The Simplex Method operates on the vertices of the feasible region, which correspond to basic feasible solutions. It uses pivot operations to transition between these vertices, guided by the objective function's coefficients (reduced costs). The method is efficient for large-scale problems and is widely used in practice.

C.2 Key Concepts

The Simplex Method relies on several key concepts:

- **Basic Feasible Solution (BFS):** A solution that satisfies all constraints and has a number of non-zero variables equal to the number of constraints.
- **Pivot Operation:** The process of moving from one BFS to another by changing the basic and non-basic variables.
- **Reduced Costs:** The coefficients of the non-basic variables in the objective function after accounting for the current basis.

Definition 6 (Reduced Cost)**.** *Let $A = [B\ N]$ where B is the current basis and N the non-basic columns. The reduced cost vector for the non-basic variables is given by:*

$$\bar{c}_N = c_N - B^{-1} N^T c_B$$

or, equivalently,

$$\bar{c}_N = c_N - c_B^T B^{-1} N$$

where c_B and c_N are the cost vectors for basic and non-basic variables, respectively. The reduced cost $\bar{c}_j$ for a non-basic variable x_j is the j-th entry of $\bar{c}_N$.

Let's say we have a basic feasible solution $x^{(1)}$ with basis matrix B, and corresponding basic and nonbasic variables split as $x^{(1)} = (x_B^{(1)}, x_N^{(1)})$. Then this basic solution satisfies:

$$x_B^{(1)} = B^{-1}b, \quad x_N^{(1)} = 0.$$

The optimal objective value is:

$$z^{(1)} = c_B^T x_B^{(1)} = c_B^T B^{-1}b.$$

Now consider any feasible $x^{(2)}$ with $x_N^{(2)} \geq 0$, but potentially not basic. Write $x^{(2)} = (x_B^{(2)}, x_N^{(2)})$, and the constraint $Ax^{(2)} = b$ becomes:

$$Bx_B^{(2)} + Nx_N^{(2)} = b \quad \Rightarrow \quad x_B^{(2)} = B^{-1}b - B^{-1}Nx_N^{(2)}.$$

The objective function value at this feasible point is:

$$z^{(2)} = c_B^T x_B^{(2)} + c_N^T x_N^{(2)} = c_B^T B^{-1}b + (c_N^T - c_B^T B^{-1}A_N)x_N^{(2)}.$$

The term $(c_N^T - c_B^T B^{-1}A_N)$ is the vector of reduced costs $\bar{c}_N$. Thus, the objective value can be expressed as:

$$z^{(2)} = c_B^T B^{-1}b + \bar{c}_N^T x_N^{(2)}.$$

We get the following relationship:

$$\begin{aligned} z^{(2)} - z^{(1)} &= (c_B^T B^{-1}b + \bar{c}_N^T x_N^{(2)}) - c_B^T B^{-1}b \\ &= \bar{c}_N^T x_N^{(2)}. \end{aligned}$$

Theorem 5 (Optimality Condition via Reduced Costs)**.** *If, at a basic feasible solution, all reduced costs $\bar{c}_j \geq 0$ for all non-basic variables x_j, then the current basic feasible solution is optimal for the minimization problem.*

Proof. Let x^* be the current basic feasible solution with basis B and non-basic variables $x_N = 0$. The objective function can be written as

$$z = c_B^T x_B + c_N^T x_N = c_B^T B^{-1}b + (c_N^T - c_B^T B^{-1}N)x_N.$$

The term $(c_N^T - c_B^T B^{-1}N)$ is the vector of reduced costs $\bar{c}_N$. For any feasible x with $x_N \geq 0$, the difference in objective value is

$$z - z^* = \bar{c}_N^T x_N.$$

Since all $\bar{c}_j \geq 0$ and $x_N \geq 0$, it follows that $z \geq z^*$. Therefore, no feasible direction can decrease the objective, and the current basic feasible solution is optimal. If all $\bar{c}_j \geq 0$ and $x_N \geq 0$, then $z \geq z^*$. Thus, no feasible direction can decrease the objective, so x^* is optimal. □

C.3 Basis Change and Pivoting

At each iteration of the Simplex Method, a **pivot** operation is performed, which changes the current basis by exchanging one basic variable (leaving variable) with one non-basic variable (entering variable).

Choosing the Entering Variable For minimization problems, the entering variable is selected among the non-basic variables with a *negative* reduced cost ($\bar{c}_j < 0$). The reduced cost $\bar{c}_j$ indicates how much the objective function will decrease per unit increase in x_j. If all reduced costs are non-negative, the current solution is optimal.

Impact of Entering Variable When a non-basic variable x_k with $\bar{c}_k < 0$ enters the basis, increasing x_k from zero will decrease the objective value at a rate of $|\bar{c}_k|$ per unit. However, the increase is limited by the feasibility constraints: as x_k increases, some basic variable may reach zero and must leave the basis.

Determining the Leaving Variable (Minimum Ratio Test) To maintain feasibility, we solve for the maximum allowable increase in x_k:

$$x_B = B^{-1}b - B^{-1}a_k x_k \geq 0 \implies x_k \leq \min_{i:(B^{-1}a_k)_i > 0} \frac{(B^{-1}b)_i}{(B^{-1}a_k)_i}$$

where a_k is the k-th column of A. The largest permissible value of x_k is:

$$\theta^* = \min_{i:(B^{-1}a_k)_i > 0} \frac{(B^{-1}b)_i}{(B^{-1}a_k)_i}$$

The basic variable corresponding to the minimum ratio is the leaving variable.

Pivot Operation After determining the entering and leaving variables, the basis is updated: the entering variable becomes basic, and the leaving variable becomes non-basic. The new basic solution is computed, and the process repeats until no further improvements can be made.

D The Simplex Algorithm

The main steps of the Simplex Algorithm are as follows:

1. Convert the LP to standard form.
2. Identify an initial BFS (using slack variables if necessary).
3. At each iteration:
   - Compute the reduced costs.
   - If all reduced costs are non-positive, the current BFS is optimal.
   - Otherwise, select an entering variable (with positive reduced cost).
   - Determine the leaving variable using the minimum ratio test.
   - Pivot to update the BFS.
4. Repeat until optimality or unboundedness is detected.

The convertion to standard form has been discussed in the previous sections. In this section we will discuss o different strategies to find an initial basic feasible solution (BFS) and the pivot operations in detail.

D.1 Finding an Initial Basic Feasible Solution

Finding an initial basic feasible solution (BFS) is crucial for starting the Simplex Method. There are several strategies to achieve this:

- **Obvious Solution:** Some LPs have an obvious initial BFS, such as when all constraints are satisfied by setting some variables to zero. Such cases can be directly identified from the problem formulation.
- **Two-Phase Simplex Method:** If the LP does not have an obvious initial BFS, the Two-Phase Simplex Method can be used:

1. In Phase 1, introduce artificial variables to create an initial BFS. The objective is to minimize the sum of these artificial variables.
2. Solve this auxiliary problem to find a feasible solution for the original LP.
3. In Phase 2, remove the artificial variables and solve the original LP using the obtained feasible solution as the starting point.

- **Big M Method:** This method introduces a large penalty (M) for artificial variables in the objective function, effectively forcing them to zero in the optimal solution. The initial BFS is constructed similarly to the Two-Phase Method.

When an initial basic feasible solution (BFS) is not readily available, two classical strategies are used: the **Two-Phase Method** and the **Big M Method**. Both approaches introduce *artificial variables* to construct an initial BFS by allowing the exploration of infeasible regions, but their strategies for reducing infeasibility differ.

Two-Phase Method (Infeasibility Reduction Strategy)

- **Phase 1:** Artificial variables are added to constraints where a BFS is not obvious. The auxiliary objective is to minimize the sum of artificial variables. The algorithm seeks a feasible solution to the original constraints by driving all artificial variables to zero.
- **Phase 2:** If the minimum value of the Phase 1 objective is zero (i.e., all artificial variables are zero), a feasible solution to the original problem has been found. The artificial variables are removed, and the original objective function is optimized starting from this feasible point.
- **Main idea:** The method explicitly separates the process of finding feasibility (Phase 1) from optimizing the original objective (Phase 2).

Example

Consider the following LP:

$$\begin{aligned} \text{minimize} \quad & x_1 + 2x_2 \\ \text{subject to} \quad & x_1 + x_2 \geq 3 \\ & x_1 - x_2 = 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

To apply the Two-Phase Method:

1. Introduce slack and artificial variables:

$$\begin{aligned} \text{minimize} \quad & x_1 + 2x_2 \\ \text{subject to} \quad & x_1 + x_2 - s_1 + a_1 = 3 \\ & x_1 - x_2 + a_2 = 1 \\ & x_1, x_2, s_1, a_1, a_2 \geq 0 \end{aligned}$$

2. Solve the auxiliary problem:

$$\begin{aligned} \text{minimize} \quad & a_1 + a_2 \\ \text{subject to} \quad & x_1 + x_2 - s_1 + a_1 = 3 \\ & x_1 - x_2 + a_2 = 1 \\ & x_1, x_2, s_1, a_1, a_2 \geq 0 \end{aligned}$$

3. If a feasible solution is found (i.e., $a_1 = a_2 = 0$), remove artificial variables and solve the original problem using the obtained BFS.

Big M Method (Penalty Strategy)

- Artificial variables are added as in the Two-Phase Method, but instead of a separate phase, a large penalty M is assigned to each artificial variable in the original objective function (e.g., minimize $c^T x + M \sum a_i$).
- The Simplex Method is applied directly. The large penalty forces artificial variables to zero in the optimal solution, provided M is sufficiently large.
- **Main idea:** Infeasibility is penalized within a single-phase optimization, so the algorithm simultaneously seeks feasibility and optimality.

Example

Consider the same LP as before:

$$\begin{aligned} \text{minimize} \quad & x_1 + 2x_2 \\ \text{subject to} \quad & x_1 + x_2 \geq 3 \\ & x_1 - x_2 = 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

To apply the Big M Method:

1. Introduce slack and artificial variables and modify the objective function:

$$\begin{aligned} \text{minimize} \quad & x_1 + 2x_2 + M(a_1 + a_2) \\ \text{subject to} \quad & x_1 + x_2 - s_1 + a_1 = 3 \\ & x_1 - x_2 + a_2 = 1 \\ & x_1, x_2, s_1, a_1, a_2 \geq 0 \end{aligned}$$

2. Solve the modified problem using the Simplex Method.

Limitations and Considerations

- **Big M Method:**
  - Choosing M is delicate: if M is too small, artificial variables may remain positive in the solution, leading to incorrect results; if M is too large, numerical instability and round-off errors can occur.
  - The method can be less reliable in practice due to these numerical issues.
- **Two-Phase Method:**
  - More robust and numerically stable, as it separates feasibility and optimality.
  - Slightly more computational effort due to the two-phase process, but preferred in most modern implementations.
- **General:** Both methods start from an infeasible solution (with artificial variables) and systematically reduce infeasibility, but the Two-Phase Method does so explicitly, while the Big M Method does so implicitly via penalties.

D.2 Proof of Finite Termination

Assume the feasible region is non-empty and bounded. At each iteration, the Simplex Method moves from one basic feasible solution (BFS) to another, improving or maintaining the objective value. Let us prove that, with an anti-cycling rule (e.g., Bland's Rule), the algorithm terminates in a finite number of steps.

Proof. Let $P = \{x \in \mathbb{R}^n : Ax = b,\ x \geq 0\}$ be the feasible region. The number of possible bases (sets of m linearly independent columns of A) is finite: at most $\binom{n}{m}$. Each BFS corresponds to a basis. The Simplex Method never visits the same basis twice if an anti-cycling rule is used. Since the objective value does not decrease and the number of bases is finite, the algorithm must terminate after finitely many steps. □

D.3 Degeneracy and Cycling

Definition 7 (Degeneracy)**.** *A basic feasible solution (BFS) is* degenerate *if at least one of its basic variables is zero, i.e., $x_B = B^{-1}b$ has some zero entries. Degeneracy is common in practical LPs, especially when the feasible region has vertices where more than n constraints are active (i.e., the intersection of more than n hyperplanes).*

Consequences of Degeneracy

- **Stalling:** The Simplex Method may perform a pivot that does not improve the objective value (the value of z remains unchanged). This is called a *degenerate pivot*.
- **Cycling:** In rare cases, the Simplex Method may revisit the same BFS multiple times, leading to an infinite loop (cycling). This is a theoretical concern, as practical implementations use anti-cycling rules.

Example of Degeneracy Consider the following tableau after some iterations:

	x_1	x_2	s_1	RHS
x_3	1	2	0	0
s_2	0	1	1	2
z	-1	0	0	0

Here, x_3 is a basic variable with value zero, so the BFS is degenerate. If x_1 enters and x_3 leaves, the new BFS may still have a zero basic variable, and the objective may not increase.

Anti-Cycling Rules To prevent cycling, several rules have been developed:

- **Bland's Rule:** Always choose the entering and leaving variables with the smallest index among all eligible candidates. This guarantees finite termination.
- **Lexicographic Rule:** Break ties in the minimum ratio test using lexicographic ordering of the tableau rows. This also prevents cycling.

Impact on Efficiency Degeneracy can cause the Simplex Method to take many more iterations than the number of vertices in the feasible region, due to repeated visits to the same BFS (stalling). However, with anti-cycling rules, the method is guaranteed to terminate.

D.4 Implementation Considerations

A practical implementation of the Simplex Method typically involves the following steps:

1. **Input and Standardization:** Read the LP problem and convert it to standard form (minimization, equality constraints, non-negative variables).
2. **Tableau Construction:** Build the initial simplex tableau, which encodes the constraints, objective function, and basic/non-basic variable structure.
3. **Initialization:** Identify an initial basic feasible solution (BFS), possibly using the Two-Phase or Big M method if one is not obvious.
4. **Pivot Operations:**
   - Compute reduced costs to select the entering variable (most negative for minimization).
   - Use the minimum ratio test to select the leaving variable.

- Perform a pivot (Gaussian elimination) to update the tableau and basis.

5. **Termination:** Repeat pivoting until all reduced costs are non-negative (optimality), or detect unboundedness/infeasibility.

6. **Solution Extraction:** Read the optimal variable values and objective from the final tableau.

Efficient implementations use data structures to track the basis, update the tableau incrementally, and exploit sparsity. Libraries such as `GLPK`, `COIN-OR`, and `CPLEX` provide robust, optimized Simplex solvers. The detailed implementation of these algorithms and their advanced features goes beyond the scope of this document.

Implementation Considerations When implementing the Simplex Method, several practical considerations arise:

- **Numerical Stability:** Care must be taken to avoid numerical instability, especially in degenerate cases. Using exact arithmetic or specialized libraries can help.
- **Sparse Matrices:** Many LPs have a sparse constraint matrix. Exploiting this sparsity can lead to more efficient implementations.
- **Warm Starts:** If solving a sequence of related LPs, using the solution from the previous LP as a starting point can significantly speed up convergence.

E A Note on LPs with bounded Variables

In many practical applications, linear programming problems may have variables that are not only non-negative but also have explicit upper bounds, i.e., $0 \leq x_j \leq u_j$ for some $u_j < +\infty$. Introducing these upper bounds as additional constraints in the tableau increases the problem size and can slow down computations.
Instead of adding explicit constraints for upper bounds, the Simplex Method can be adapted to handle variable bounds directly. The key idea is to allow variables to be either at their lower bound ($x_j = 0$), at their upper bound ($x_j = u_j$), or strictly between bounds ($0 < x_j < u_j$). In this approach:

- A variable at its lower bound is treated as non-basic at zero (as in the standard simplex).
- A variable at its upper bound is treated as non-basic at u_j.
- Only variables strictly between bounds are basic.

Pivoting with Upper Bounds When a non-basic variable at its lower bound is chosen to enter the basis, it increases from zero as usual. If it reaches its upper bound before any basic variable leaves the basis, it becomes non-basic at its upper bound, and the process continues. Similarly, a basic variable may hit its upper or lower bound during the minimum ratio test, and in that case, it leaves the basis and becomes non-basic at the corresponding bound.

Advantages This approach avoids increasing the tableau size and can be implemented efficiently. Most modern simplex solvers (e.g., revised simplex) support upper and lower bounds natively using this technique.

Summary

- Do not introduce explicit upper bound constraints into the tableau.
- Track whether each variable is at its lower bound, upper bound, or basic.
- Adapt the minimum ratio test and pivoting rules to account for both lower and upper bounds.