DegreeLive

Bachelor of Business Administration (BBA) | Fifth Semester | UK5DSCMGT303

Management Science

Course Code: UK5DSCMGT303 • Module 2 Notes

degreelive.in

Module 2: Linear Programming Problems (LPP)

Course Code: UK5DSCMGT303 • Module 2 Notes

Chapter 1: Conceptual Foundations and Mathematical Formulation of Linear Programming

1.1 Meaning, Significance, and Resource Allocation Paradigm

Linear Programming (LP) is a sophisticated mathematical optimization technique designed to allocate scarce, heterogeneous organizational resources—including productive machine hours, skilled human labor, raw materials, warehouse storage volume, and financial working capital—among competing productive activities in an optimal manner. The primary objective is either to **maximize an organizational payoff** (such as total net contribution, operational profit, or production throughput) or to **minimize an operational sacrifice** (such as aggregate manufacturing cost, transportation expense, material scrap, or total production cycle time), subject to a system of linear structural constraints imposed by technological, market, or policy limitations.

Conceived originally by Soviet mathematician Leonid Kantorovich in 1939 and independently formalized by American mathematician George B. Dantzig in 1947 through the breakthrough development of the Simplex Algorithm, linear programming represents one of the most widely applied quantitative tools in industrial management, corporate supply chain planning, energy grid dispatching, and investment portfolio optimization.

1.2 Core Structural Components of a Linear Programming Model

Every linear programming problem consists of four mandatory mathematical structural elements:

1. **Decision Variables ($x_1, x_2, \dots, x_n$):** Continuous, controllable mathematical variables that represent the quantifiable levels of commercial activity chosen by the decision-maker (e.g., number of units of Product A and Product B to manufacture per month).
2. **The Objective Function:** A mathematical expression representing the managerial goal to be optimized. It takes the form of a linear equation expressing total profit or cost as a function of the decision variables:

   > $\text{Optimize (Max or Min) } Z = c_1x_1 + c_2x_2 + \cdots + c_nx_n = \sum_{j=1}^{n} c_jx_j$

   Legend: Z = Total objective value; c_j = Contribution margin or unit cost coefficient of decision variable j; x_j = Level of activity for decision variable j.
3. **Structural (Technological) Constraints:** A system of linear equations or inequalities reflecting physical, operational, technological, and legal limitations restricting resource consumption:

DegreeLive

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad (\text{for } i = 1, 2, \dots, m)$$

Legend: a_{ij} = Technological consumption coefficient (amount of resource i consumed per unit of activity j); b_i = Maximum available supply or capacity limit of resource i.

4. **Non-Negativity Restrictions:** Physical activities cannot occur in negative quantities; producing negative quantities of an automobile or baking negative loaves of bread is physically impossible:

$$x_j \geq 0 \quad (\text{for all } j = 1, 2, \dots, n)$$

Legend: Ensures decision variables remain in the non-negative first quadrant of Cartesian coordinate space.

1.3 Fundamental Assumptions of Linear Programming

The mathematical validity and computational solvability of linear programming models rest upon five rigid structural assumptions:

Underlying Assumption	Mathematical Formulation	Managerial & Practical Significance
1. Linearity (Proportionality)	The contribution of each variable to the objective function and resource usage is strictly proportional: $f(k \cdot x) = k \cdot f(x)$.	Rules out economies of scale, quantity discounts, learning curve effects, and diminishing marginal returns. Unit costs and revenues remain constant regardless of production volume.
2. Additivity	Total payoff and total resource consumption represent the exact sum of individual contributions: $f(x_1 + x_2) = f(x_1) + f(x_2)$.	Rules out synergistic interactions and cross-product cannibalization. Products do not interact chemically, physically, or commercially to boost or depress shared consumption.
3. Divisibility (Continuity)	Decision variables are continuous real numbers: $x_j \in R+$, admitting fractional and decimal values.	Permits solutions like producing 14.75 metric tons of chemical fertilizer. If variables must be strict integers (e.g., building whole ships), Integer Programming (IP) must be deployed.
4. Determinism (Certainty)	All technological coefficients (a_{ij}), cost/profit coefficients (c_j), and resource limits (b_i) are known with certainty.	No random statistical fluctuations exist during the planning horizon. If parameters fluctuate probabilistically, Stochastic Programming must be utilized.
5. Finiteness	The number of decision variables (n) and structural constraints (m) is strictly finite.	Ensures that algorithmic search spaces are mathematically bounded and

Underlying Assumption	Mathematical Formulation	Managerial & Practical Significance
		terminate within finite computational steps.

Chapter 2: Linear Programming Formulation Mechanics

2.1 Step-by-Step Problem Formulation Protocol

Formulating a linear programming model from an unstructured commercial narrative requires a rigorous four-step analytical protocol:

- **Step 1: Identify Key Managerial Decisions:** Determine the exact operational choices management must execute and define unambiguous decision variables with explicit physical and time dimensions (e.g., let x1 = units of Executive Desks manufactured per week).
- **Step 2: Formalize the Objective Function:** Express the overriding organizational goal (maximizing net contribution or minimizing production expenditure) as a linear algebraic sum of decision variables multiplied by unit parameters.
- **Step 3: Establish the Constraint Set:** Systematically inventory every physical limitation (labor hours, raw material supplies, machine capacities, quality standards, statutory safety caps, minimum contractual delivery obligations) and write corresponding linear inequalities (≤, ≥) or equalities (=).
- **Step 4: Declare Non-Negativity:** State explicitly that all decision variables must be non-negative ($x_j \geq 0$).

2.2 Exemplar Formulations Across Industrial Domains

Case A: Industrial Product Mix Formulation: An engineering firm manufactures two precision components: Component 1 (x1) and Component 2 (x2). Manufacturing involves three departments: Fabrication, Assembly, and Quality Inspection. Fabricating one unit of Component 1 requires 3 hours of machining and 2 hours of assembly; Component 2 requires 2 hours of machining and 4 hours of assembly. Total available machine hours per week are 180; assembly capacity is 240 hours. Quality inspection can process at most 70 total components per week. The unit profit contribution is Rs 50 for Component 1 and Rs 60 for Component 2.

```
Maximize Z = 50x1 + 60x2 (Total Weekly Contribution in Rupees)
Subject to structural constraints:
1. Machining Capacity: 3x1 + 2x2 ≤ 180 hours
2. Assembly Capacity: 2x1 + 4x2 ≤ 240 hours
3. Quality Inspection Limit: x1 + x2 ≤ 70 components
4. Non-negativity: x1, x2 ≥ 0
```

DegreeLive

Legend: x_1 = Weekly units of Component 1; x_2 = Weekly units of Component 2; Z = Aggregate weekly profit contribution.

Case B: Agricultural Feed Mix (Cost Minimization) Formulation: An agricultural cooperative prepares animal feed blending two raw grains: Grain X (x1) and Grain Y (x2). Each kilogram of Grain X costs Rs 25 and provides 3 units of Protein and 2 units of Carbohydrates. Grain Y costs Rs 30 per kg and provides 4 units of Protein and 5 units of Carbohydrates. The daily nutritional requirements mandate a minimum of 24 units of Protein and 30 units of Carbohydrates. To prevent digestive toxicity, total grain blend cannot exceed 12 kg per day.

```
Minimize Z = 25x1 + 30x2 (Total Feed Cost in Rupees)
Subject to nutritional constraints:
1. Minimum Protein Intake: 3x1 + 4x2 ≥ 24 units
2. Minimum Carbohydrate Intake: 2x1 + 5x2 ≥ 30 units
3. Maximum Toxicity Ceiling: x1 + x2 ≤ 12 kilograms
4. Non-negativity: x1, x2 ≥ 0
```

Legend: x_1 = Kilograms of Grain X; x_2 = Kilograms of Grain Y; Z = Total daily expenditure on feed blend.

Chapter 3: Graphical Solution Method for Two-Variable Linear Programs

3.1 Mathematical Theory of the Graphical Solution

The Graphical Method is a powerful visual optimization approach applicable to linear programming problems containing exactly two decision variables ($x1, x2$). The method relies on two foundational theorems of mathematical optimization:

- **Convex Polygon Theorem:** The set of all feasible solutions satisfying a system of linear inequalities forms a **convex polygonal set** (or convex polyhedron). A set is convex if the straight line segment joining any two points within the set lies entirely within the set.
- **Fundamental Theorem of Linear Programming:** If a linear programming problem possesses a finite optimal solution, that optimal solution will always occur at one or more **extreme points (corner vertices)** of the feasible region.

3.2 Systematic Step-by-Step Graphical Algorithm

To solve a two-variable linear program graphically, execute the following six steps:

1. **Establish Cartesian Axes:** Represent decision variable x1 on the horizontal axis and variable x2 on the vertical axis. Restrict graphing entirely to the positive first quadrant ($x1 \geq 0, x2 \geq 0$).
2. **Convert Inequalities to Boundary Lines:** Treat each structural inequality as an exact equality boundary line. For example, convert $3x1 + 2x2 \leq 180$ into the linear equation $3x1 + 2x2 = 180$.
3. **Plot Boundary Lines:** Find the axis intercepts for each boundary line by setting $x1 = 0$ to solve for x2$, and setting $x2 = 0$ to solve for x1$. Draw straight lines connecting intercept pairs across the quadrant.

4. **Identify the Feasible Space:** Determine the permissible half-plane for each constraint. For a ≤ constraint, test the origin point (0, 0); if satisfied, shade towards the origin. For a ≥ constraint, shade away from the origin. The mutual intersection of all shaded half-planes defines the **Feasible Region**.
5. **Locate Corner Points (Vertices):** Identify all extreme corner points enclosing the feasible polygon. Calculate the exact algebraic coordinates of each vertex by solving the simultaneous linear equations of the intersecting boundary lines.
6. **Evaluate the Objective Function:**
   - *Corner Point Method:* Substitute the coordinates of every corner point into the objective function Z. Select the vertex yielding the maximum Z (for profit maximization) or minimum Z (for cost minimization).
   - *Iso-Profit / Iso-Cost Line Method:* Select an arbitrary profit value Z0 and plot the parallel contour line $c_1x_1 + c_2x_2 = Z_0$. Shift this line parallelly away from the origin (for maximization) until it touches the last corner vertex of the feasible region before exiting. That contact vertex is the unique optimal solution.

3.3 Comprehensive Worked Numerical Problem: Product Mix Maximization

Consider the industrial component optimization model formulated in Section 2.2:

> $\text{Maximize } Z = 50x_1 + 60x_2$
> $\text{Subject to: (1) } 3x_1 + 2x_2 \le 180;\ \text{(2) } 2x_1 + 4x_2 \le 240;\ \text{(3) } x_1 + x_2 \le 70;\ \text{and } x_1, x_2 \ge 0.$

Step 1: Calculating Axis Intercepts for Boundary Lines

- **Line 1 (Machining):** $3x_1 + 2x_2 = 180$.
  - If $x_1 = 0$, then $2x_2 = 180 \implies x_2 = 90$. Point: (0, 90).
  - If $x_2 = 0$, then $3x_1 = 180 \implies x_1 = 60$. Point: (60, 0).
- **Line 2 (Assembly):** $2x_1 + 4x_2 = 240$.
  - If $x_1 = 0$, then $4x_2 = 240 \implies x_2 = 60$. Point: (0, 60).
  - If $x_2 = 0$, then $2x_1 = 240 \implies x_1 = 120$. Point: (120, 0).
- **Line 3 (Quality Inspection):** $x_1 + x_2 = 70$.
  - If $x_1 = 0$, then $x_2 = 70$. Point: (0, 70).
  - If $x_2 = 0$, then $x_1 = 70$. Point: (70, 0).

Step 2: Identifying Vertices of the Feasible Polygon

The feasible region is bounded by the non-negativity axes and the three linear constraints. It forms a 5-sided convex polygon with vertices labeled **O, A, B, C, D**:

1. **Point O:** Origin (0, 0).

2. **Point A:** On the vertical axis. Bounded by the tightest x2 intercept: Point $(0, 60)$ from Assembly Line.
3. **Point B:** Intersection of Line 2 (Assembly: $2x1 + 4x2 = 240$) and Line 3 (Inspection: $x1 + x2 = 70$).
   - From Line 3: $x1 = 70 - x2$. Substitute into Line 2:
   - $2(70 - x2) + 4x2 = 240 \implies 140 - 2x2 + 4x2 = 240 \implies 2x2 = 100 \implies x2 = 50$.
   - Substitute back: $x1 = 70 - 50 = 20$. Vertex B coordinates: $(20, 50)$.
4. **Point C:** Intersection of Line 3 (Inspection: $x1 + x2 = 70$) and Line 1 (Machining: $3x1 + 2x2 = 180$).
   - From Line 3: $x2 = 70 - x1$. Substitute into Line 1:
   - $3x1 + 2(70 - x1) = 180 \implies 3x1 + 140 - 2x1 = 180 \implies x1 = 40$.
   - Substitute back: $x2 = 70 - 40 = 30$. Vertex C coordinates: $(40, 30)$.
5. **Point D:** On the horizontal axis. Bounded by the tightest x1 intercept: Point $(60, 0)$ from Machining Line.

Step 3: Corner Point Objective Function Evaluation

Corner Vertex	Coordinates (x1, x2)	Objective Value Calculation: Z = 50x1 + 60x2	Total Profit (Rs)
Vertex O	$(0, 0)$	$50(0) + 60(0) = 0 + 0$	Rs 0
Vertex A	$(0, 60)$	$50(0) + 60(60) = 0 + 3600$	Rs 3,600
Vertex B	$(20, 50)$	$50(20) + 60(50) = 1000 + 3000$	**Rs 4,000 (OPTIMAL)**
Vertex C	$(40, 30)$	$50(40) + 60(30) = 2000 + 1800$	Rs 3,800
Vertex D	$(60, 0)$	$50(60) + 60(0) = 3000 + 0$	Rs 3,000

Managerial Conclusion and Resource Slacks

The unique mathematically optimal product mix is to manufacture **20 units of Component 1** and **50 units of Component 2** per week, achieving a maximum profit contribution of **Rs 4,000**.

Evaluation of Resource Utilization:

- **Machining Hours:** $3(20) + 2(50) = 60 + 100 = 160$ hours used out of 180 available. Slack = 20 unused machining hours (Non-binding constraint).
- **Assembly Hours:** $2(20) + 4(50) = 40 + 200 = 240$ hours used out of 240 available. Slack = 0 (Binding bottleneck constraint).

DegreeLive

- **Inspection Hours:** $20 + 50 = 70$ units inspected out of 70 capacity. Slack = 0 (Binding bottleneck constraint).

Chapter 4: Special Computational Cases and Boundary Conditions in LPP

4.1 Multiple (Alternative) Optimal Solutions

Multiple optimal solutions occur when the slope of the objective function is exactly identical to the slope of a binding structural constraint line. Mathematically, the iso-profit contour line coincides perfectly with an entire edge of the feasible polygon rather than touching a single discrete vertex.

In such instances, any point along the line segment connecting the two optimal corner vertices yields the exact same maximum objective value $Z*$. If XA and XB are optimal extreme vertices, then any convex linear combination:

```
X(λ) = λXA + (1 − λ)XB   (where 0 ≤ λ ≤ 1)
```

Legend: X(\lambda) = Infinite set of alternate optimal production allocations along the boundary edge.

Managerial Advantage: Multiple optima afford executives immense strategic flexibility. If secondary operational preferences (such as client relationship preservation or workforce specialization) favor one product over another, management can shift along the optimal edge without sacrificing a single rupee of profit.

4.2 Infeasible (Non-Feasible) Solutions

An infeasible solution arises when there is no single coordinate point $(x1, x2)$ that simultaneously satisfies all structural constraints and non-negativity conditions. The mathematical intersection of the constraint half-planes is the **empty set ($\emptyset$)**.

Causes: Infeasibility stems from mutually contradictory operational requirements, such as requiring high minimum contractual delivery volumes that mathematically exceed maximum available manufacturing factory capacity. Management must relax one or more rigid constraints or inject additional capital resources.

4.3 Unbounded Solutions

An unbounded solution occurs when the feasible region is open-ended and extends infinitely in the direction of optimization, allowing the objective function value to grow without finite limit ($Z \to \infty$ for maximization or $Z \to -\infty$ for minimization).

Causes: Unbounded solutions never occur in real physical manufacturing systems; they signify a fundamental error in problem formulation where the analyst inadvertently omitted a critical limiting constraint (such as assuming unlimited market demand or zero raw material supply bottlenecks).

4.4 Redundant Constraints

A constraint is mathematically redundant if its removal does not alter the geometry or boundaries of the feasible region. It represents an operational capacity that is so abundant that it will never become a binding bottleneck under any circumstances.

While redundant constraints do not invalidate the mathematical solution, identifying and removing them simplifies computational processing in large-scale enterprise linear programming models.

Special Case	Geometric Graphic Manifestation	Managerial Diagnostics & Action
Multiple Optima	Iso-profit line lies parallel to and coincides along an entire bounding constraint segment.	Exploit strategic managerial flexibility; select optimal mix based on non-quantifiable qualitative criteria.
Infeasibility	No overlapping shaded area; empty intersection of constraint half-planes.	Contradictory policy demands; renegotiate contractual commitments or expand physical capacity.
Unboundedness	Feasible polygon is open-ended; iso-profit line can move indefinitely without boundary contact.	Formulation defect; identify and insert the missing real-world operational constraints.
Redundancy	Constraint boundary line lies completely outside the active boundaries of the feasible region.	Prune redundant equation from linear solver to reduce computational overhead.

Chapter 5: Managerial Applications, Sensitivity Analysis, and Duality

5.1 Diverse Managerial Applications of Linear Programming

Linear programming serves as an indispensable decision-support engine across modern commercial and public sectors:

- **Manufacturing Production Planning:** Determining optimal monthly product mixes to maximize net factory contributions under machine, labor, and warehouse limitations.
- **Supply Chain and Oil Refinery Blending:** Determining optimal blending ratios of crude petroleum fractions to produce aviation fuel, diesel, and gasoline satisfying octane and sulfur limits at minimum blending cost.
- **Financial Portfolio Optimization:** Allocating capital across corporate equities, government treasury bonds, and money market instruments to maximize yield subject to portfolio beta risk constraints.
- **Marketing Media Selection:** Allocating corporate advertising budgets across digital search, streaming video, television, and print media to maximize consumer impressions subject to frequency caps and platform budgets.

DegreeLive

- **Agricultural Crop Allocation:** Allocating farmland acreage across competing seasonal crops to maximize agricultural yield subject to soil moisture, irrigation quotas, and tractor hours.

5.2 Concept of Duality and Economic Shadow Prices

Every linear programming problem (termed the **Primal Problem**) has an associated, mathematically symmetric twin problem known as the **Dual Problem**. If the primal problem seeks to maximize profit subject to resource limits, the dual problem seeks to minimize the total imputed economic valuation of those resources.

The decision variables of the dual problem are designated as **Shadow Prices (Dual Prices)**. A shadow price represents the marginal increase in the maximum objective function value resulting from an additional one-unit relaxation in the capacity of that specific constraint ($\Delta Z / \Delta bi$).

- For non-binding constraints with unused slack (e.g., machining hours in our worked problem with 20 hours idle), the shadow price is **identically zero (Rs 0 per hour)** because having more unused capacity adds zero incremental profit.
- For binding bottleneck constraints (e.g., assembly hours), the shadow price is strictly positive. If the shadow price of assembly is Rs 10 per hour, management knows it can economically justify paying up to Rs 10 in overtime premiums to secure additional assembly labor.

Chapter 6: Graphical Solution for Minimization Problems

6.1 Distinctive Attributes of Cost Minimization Problems

While profit maximization linear programs typically feature less-than-or-equal-to ($\leq$) resource constraints that produce a bounded feasible region anchored to the coordinate origin, cost minimization problems (such as diet planning, fertilizer blending, and transportation sourcing) are characterized primarily by greater-than-or-equal-to ($\geq$) requirements. These specifications ensure that minimum nutritional thresholds, tensile strength criteria, or statutory customer demands are satisfied.

Consequently, the feasible region in pure minimization models is often **unbounded from above**. The graphical algorithm must systematically examine the lower boundary vertices of this convex polygon, moving the iso-cost line towards the origin to identify the lowest possible cost contour that remains in physical contact with the feasible region.

6.2 Complete Worked Numerical Problem: Feed Mix Cost Minimization

Let us solve the agricultural feed formulation problem formalized in Section 2.2:

```
Minimize Z = 25x1 + 30x2 (Total Feed Expenditure in Rupees)
Subject to:
1. Protein Requirement: 3x1 + 4x2 ≥ 24
2. Carbohydrate Requirement: 2x1 + 5x2 ≥ 30
```

DegreeLive

```
3. Digestive Ceiling Constraint: x1 + x2 ≤ 12
4. Non-negativity: x1, x2 ≥ 0
```

Step 1: Calculating Axis Intercepts for Boundary Lines

- **Line 1 (Protein):** $3x1 + 4x2 = 24$.
  - If $x1 = 0 \implies 4x2 = 24 \implies x2 = 6$. Intercept Point: (0, 6).
  - If $x2 = 0 \implies 3x1 = 24 \implies x1 = 8$. Intercept Point: (8, 0).
  - Since this is a ≥ constraint, the feasible half-plane lies above and away from the origin (0, 0).
- **Line 2 (Carbohydrates):** $2x1 + 5x2 = 30$.
  - If $x1 = 0 \implies 5x2 = 30 \implies x2 = 6$. Intercept Point: (0, 6).
  - If $x2 = 0 \implies 2x1 = 30 \implies x1 = 15$. Intercept Point: (15, 0).
  - Since this is a ≥ constraint, the feasible half-plane lies above and away from the origin.
- **Line 3 (Digestive Ceiling):** $x1 + x2 = 12$.
  - If $x1 = 0 \implies x2 = 12$. Intercept Point: (0, 12).
  - If $x2 = 0 \implies x1 = 12$. Intercept Point: (12, 0).
  - Since this is a ≤ constraint, the feasible half-plane lies below and towards the origin.

Step 2: Identifying the Extreme Corner Vertices

The mutual intersection of these three half-planes and the non-negativity axes forms a bounded convex polygon enclosed by four corner vertices labeled **P, Q, R, S**:

1. **Vertex P:** Upper vertical intercept bounded by the digestive ceiling line: Point (0, 12).
2. **Vertex Q:** Lower vertical intercept shared by both nutritional constraint lines on the vertical axis: Point (0, 6). At this point, $x1 = 0$ and $x2 = 6$.
3. **Vertex R:** Intersection of Line 1 (Protein: $3x1 + 4x2 = 24$) and Line 2 (Carbohydrates: $2x1 + 5x2 = 30$).
   - Multiply Line 1 by 2: $6x1 + 8x2 = 48$.
   - Multiply Line 2 by 3: $6x1 + 15x2 = 90$.
   - Subtract the first equation from the second: $7x2 = 42 \implies x2 = 6$.
   - Substitute back into Line 1: $3x1 + 4(6) = 24 \implies 3x1 + 24 = 24 \implies 3x1 = 0 \implies x1 = 0$.
   - Notice that Vertex R coincides identically with Vertex Q at (0, 6), confirming that both nutritional lines intersect exactly on the vertical axis!
4. **Vertex S:** Intersection of Line 2 (Carbohydrates: $2x1 + 5x2 = 30$) and Line 3 (Digestive Ceiling: $x1 + x2 = 12$).
   - From Line 3: $x1 = 12 - x2$. Substitute into Line 2:
   - $2(12 - x2) + 5x2 = 30 \implies 24 - 2x2 + 5x2 = 30 \implies 3x2 = 6 \implies x2 = 2$.

DegreeLive

- Substitute back: $x_1 = 12 - 2 = 10$. Vertex S coordinates: $(10, 2)$.

5. **Vertex T:** Intersection of Line 3 (Digestive Ceiling: $x_1 + x_2 = 12$) and horizontal axis at Point $(12, 0)$. However, at $(12, 0)$, we test the carbohydrate requirement: $2(12) + 5(0) = 24$, which fails the minimum requirement of 30! Thus $(12, 0)$ is infeasible.

Step 3: Corner Point Cost Evaluation

Corner Vertex	Coordinates (x_1, x_2)	Cost Calculation: $Z = 25x_1 + 30x_2$	Total Cost (Rs)
Vertex P	$(0, 12)$	$25(0) + 30(12) = 0 + 360$	Rs 360
Vertex Q	$(0, 6)$	$25(0) + 30(6) = 0 + 180$	**Rs 180 (OPTIMAL MINIMUM)**
Vertex S	$(10, 2)$	$25(10) + 30(2) = 250 + 60$	Rs 310

Managerial Solution and Nutritional Interpretation

The optimal minimum-cost feed blending strategy is to purchase **0 kg of Grain X** and **6 kg of Grain Y** per day, resulting in a minimum daily expenditure of **Rs 180**.

Nutritional Verification:

- Protein provided: $3(0) + 4(6) = 24$ units (exactly meets minimum threshold).
- Carbohydrates provided: $2(0) + 5(6) = 30$ units (exactly meets minimum threshold).
- Total feed weight: $0 + 6 = 6$ kg (well within the 12 kg toxicity ceiling, leaving 6 kg of safety headroom).

Chapter 7: Beyond Graphical Analysis — Conceptual Bridge to the Simplex Method

7.1 Severe Dimensional Limitations of Graphical Optimization

While the graphical method provides incomparable visual intuition into the geometry of optimization, convex polygons, and extreme corner points, its practical commercial applicability is severely constrained. In real-world enterprise operations, decision problems rarely involve merely two products. A modern petrochemical refinery optimizes over 200 fuel fractions; an automobile assembly plant schedules thousands of distinct sub-assemblies; a global supply chain balances hundreds of warehouse distribution nodes.

Because physical human geometry cannot visualize spatial dimensions beyond three coordinate axes (x_1, x_2, x_3), graphical linear programming cannot solve problems involving three or more decision

variables. This critical barrier necessitates an algebraic, algorithmic optimization method that operates seamlessly in multi-dimensional vector space—the **Simplex Method**.

7.2 Standard Form and Algebraic Transformation Variables

Before any linear programming problem can be processed by algebraic simplex algorithms, it must be converted from its original inequality formulation into the **Standard Form of LPP**, where all structural constraints are expressed as exact mathematical equalities with non-negative right-hand sides ($b_i \geq 0$). This transformation requires introducing specialized algebraic variables:

Variable Classification	Algebraic Mathematical Role	Managerial & Economic Interpretation
Slack Variables ($s_i \geq 0$)	Added to the left side of a $\leq$ inequality: $\sum a_{ij}x_j + s_i = b_i$. Objective coefficient = 0.	Represents unused, idle capacity of resource i (e.g., unutilized machine hours or surplus warehouse square footage).
Surplus Variables ($e_i \geq 0$)	Subtracted from the left side of a $\geq$ inequality: $\sum a_{ij}x_j - e_i = b_i$. Objective coefficient = 0.	Represents the excess amount by which operational activity surpasses the bare minimum statutory requirement.
Artificial Variables ($A_i \geq 0$)	Added to $\geq$ and = constraints to provide an initial identity matrix basis for Simplex iteration.	Possesses zero real physical meaning; penalized heavily in the objective function (via Big-M or Two-Phase method) to force it to zero in the final optimal basis.

7.3 Basic Feasible Solutions and Extreme Points Equivalence

The mathematical elegance of Dantzig's Simplex Algorithm lies in the fundamental algebraic proof that **every extreme corner vertex of the geometric feasible polygon corresponds exactly to a Basic Feasible Solution (BFS)** of the transformed equality system:

- In a system of m constraint equations with n variables ($n > m$), a **Basic Solution** is obtained by setting $n - m$ variables (designated as *Non-Basic Variables*) equal to zero and solving the remaining m equations for the m *Basic Variables*.
- If all basic variable values are non-negative (≥ 0), the solution is a **Basic Feasible Solution**.
- Instead of evaluating all possible corner points, the Simplex Method starts at an initial BFS (typically the origin with all slacks as basic variables), systematically assesses adjacent vertices along connecting edges, and transitions iteratively to neighboring vertices that yield improved objective function values until the optimality criterion ($c_j - z_j \leq 0$ for maximization) is fully satisfied.

DegreeLive