Second Assignment: Linear, Integer, and Mixed-Integer Programming

1 Linear Programming Basics

1.1 Conceptual Questions

1. Explain what a linear programming (LP) problem is. A linear programming (LP) problem is an optimization problem in which a *linear* objective function of decision variables is minimized or maximized subject to a set of *linear* equality and/or inequality constraints, where the feasible set is a convex polyhedron (feasible region) in $\mathbb{R}^n$.[1]

2. What are decision variables, objective function, and constraints in LP? In LP:

- *Decision variables* are the unknown quantities the model seeks to determine (e.g., production quantities) and are the coordinates of the design vector $x \in \mathbb{R}^n$.[2]
- The *objective function* is a linear function $c^\top x$ that represents the performance criterion (such as cost, profit, or time) to be minimized or maximized.[3]

[1]See, e.g., Kamran Iqbal, *Fundamental Engineering Optimization Methods*, Ch. 5; S. S. Rao, *Engineering Optimization: Theory and Practice*, Ch. 3; J. Nocedal and S. Wright, *Numerical Optimization*, Ch. 13.

[2]Iqbal, *Fundamental Engineering Optimization Methods*, Ch. 1.

[3]Rao, *Engineering Optimization: Theory and Practice*, Sec. 1.4.

- The *constraints* are linear equations or inequalities of the form $Ax \leq b$, $Ax = b$, or $Ax \geq b$ that restrict the feasible values of the decision variables, typically representing resource limits or physical requirements.[4]

3. Why is it important to write all constraints in linear form for LP solvers? LP solvers assume that both the objective and constraints are linear, so that:

- The feasible region is a convex polyhedron and any local optimum is also a global optimum.[5]
- Algorithms such as the simplex method and interior-point methods can exploit linearity to guarantee convergence and global optimality.[6]

If constraints are not linear, the problem is no longer an LP, convexity may be lost, and LP algorithms and correctness guarantees no longer apply.[7]

4. What is an auxiliary variable and when do we use it? An auxiliary variable is an additional decision variable introduced to transform a nonlinear, logical, or composite expression (e.g., absolute values, min, max, piecewise-linear functions) into an equivalent system of linear constraints, so that the problem fits the LP/MILP framework.[8] Examples include slack/surplus variables to convert inequalities to equalities, or variables representing $\min(x_1, x_2)$, $\max(x_1, x_2)$, or segment costs in piecewise-linear approximations.[9]

1.2 Modeling Exercise (P1, P2, P3)

A factory produces three products P_1, P_2, P_3 with profits per unit $40, 30, 50$ and resource usage as given in the assignment. Labor limit = 150 hours, material limit = 200 units.

[4] Nocedal and Wright, *Numerical Optimization*, Ch. 13.
[5] Iqbal, Ch. 4; Rao, Ch. 3.
[6] Nocedal and Wright, Chs. 13–14.
[7] Rao, Chs. 3–4.
[8] Rao, Ch. 12 (separable and piecewise-linear programming); Iqbal, Ch. 6.
[9] Nocedal and Wright, Sec. 12.4; Rao, Sec. 12.2.

1. Decision variables. Let

x_1 = number of units of product P_1 produced, x_2 = number of units of product P_2 produced,

All decision variables are continuous and nonnegative:

$$x_1, x_2, x_3 \geq 0.$$

2. Objective function. Total profit equals profit per unit times quantity for each product, summed over products, hence

$$\max \; z = 40x_1 + 30x_2 + 50x_3.$$

This is linear because each term is a profit coefficient multiplied by a decision variable, and we assume profit is proportional to production.[10]

3. Resource constraints. Labor usage per unit: 2 (P1), 1 (P2), 3 (P3). Labor availability 150 hours, so

$$2x_1 + 1x_2 + 3x_3 \leq 150.$$

Plain explanation: the left-hand side is total labor consumed (sum of "hours per unit × units" over all products); the inequality ensures total labor used does not exceed the available 150 hours.[11]

Material usage per unit: 3 (P1), 2 (P2), 4 (P3). Material availability 200 units, so

$$3x_1 + 2x_2 + 4x_3 \leq 200.$$

Plain explanation: the left-hand side is total material consumed across all products; the constraint limits that total to at most the material availability 200 units.[12]

Full LP model:

$$\begin{aligned} \max \; & z = 40x_1 + 30x_2 + 50x_3, \\ \text{s.t.} \; & 2x_1 + x_2 + 3x_3 \leq 150, \\ & 3x_1 + 2x_2 + 4x_3 \leq 200, \\ & x_1, x_2, x_3 \geq 0. \end{aligned}$$

[10] Cf. the "simplified manufacturing problem" in Iqbal, Sec. 1.2.

[11] Standard LP "tables and chairs" example; see Iqbal, Example 1.6.

[12] Similar resource-constraint formulations in Rao, Sec. 1.4.

2 Integer and Binary Variables with Logical Constraints

2.1 Conceptual Questions

1. What is an integer programming (IP) problem and how does it differ from LP? An integer programming (IP) problem is an optimization problem in which some or all decision variables are constrained to take integer values (e.g., $x_i \in \mathbb{Z}$ or $x_i \in \{0, 1\}$), while the objective and constraints are usually linear.[13] It differs from LP in that the feasible region is discrete rather than a continuous polyhedron, which makes IP problems NP-hard in general and requires combinatorial algorithms such as branch-and-bound, cutting planes, or branch-and-cut.[14]

2. What is a binary variable and when do we use it? A binary variable is an integer decision variable that can only take values 0 or 1, i.e., $x \in \{0, 1\}$.[15] Binary variables are used to model yes/no decisions (e.g., open/close a facility, select/not select a project, assign/not assign a product to a machine) and to encode logical conditions in MILP models.[16]

3. Example of a real-world situation requiring integer variables. A classical example is the knapsack problem: selecting a subset of indivisible items (e.g., projects, products, or investments) to maximize total value subject to a weight or budget constraint, where each item is either chosen or not, modeled via binary variables.[17] Other examples include crew scheduling, machine assignment, shift planning, and network design where decisions are inherently discrete.[18]

4. How can logical conditions be modeled using linear inequalities with binary variables? Logical conditions can be translated into linear constraints on binary variables by exploiting the fact that 0 and 1 encode

[13] Rao, Ch. 10; Iqbal, Ch. 6.
[14] Rao, Sec. 10.6; Nocedal and Wright, Sec. 16.2.
[15] Rao, Sec. 10.5; Iqbal, Sec. 6.4.
[16] Rao, Sec. 10.2; Nocedal and Wright, Sec. 16.1.
[17] Iqbal, Problem 12 (Knapsack); Rao, Sec. 10.5.
[18] Rao, Ch. 10.

truth values, and by using "big-M" bounds as needed.[19] Typical patterns include:

- Implication "if-then": $x = 1 \Rightarrow y = 1$ can be modeled as $x \leq y$ for binary variables.
- Either-or: "$x = 1$ or $y = 1$" modeled as $x + y \geq 1$.
- At most r variables equal to 1: $\sum_i x_i \leq r$.
- At least r variables equal to 1: $\sum_i x_i \geq r$.
- Conditional bounds via big-M: e.g., $x \leq My$ forces $x = 0$ when $y = 0$.

These constructions are standard in MILP modeling.[20]

2.2 Logical Constraints Exercises

Let $x_1, \ldots, x_n$ be binary variables.

1. If $x_1 = 0$, then $x_2 = 1$. Logical form:

$$x_1 = 0 \Rightarrow x_2 = 1.$$

Linear constraint:

$$x_2 \geq 1 - x_1.$$

If $x_1 = 0$, then $x_2 \geq 1 \Rightarrow x_2 = 1$; if $x_1 = 1$, the constraint is $x_2 \geq 0$, nonrestrictive beyond binarity.[21]

2. Either $x_1 = 1$ or $x_2 = 1$ (at least one is 1). Logical form:

$$x_1 = 1 \vee x_2 = 1.$$

Linear constraint:

$$x_1 + x_2 \geq 1.$$

[19] Rao, Sec. 10.5; Nocedal and Wright, Sec. 16.1.
[20] See also Nocedal and Wright, Sec. 12.4 on complementarity and logical reformulations.
[21] Rao, Sec. 10.5.

3. Exactly one of x_1 and x_2 equals 1. Logical form:

$$(x_1 = 1, x_2 = 0) \vee (x_1 = 0, x_2 = 1).$$

Linear constraint:

$$x_1 + x_2 = 1.$$

4. Both x_1 and x_2 must be 1. Logical form:

$$x_1 = 1 \wedge x_2 = 1.$$

Linear constraints:

$$x_1 = 1, \quad x_2 = 1.$$

5. If $x_1 = 0$, then at least two of x_2 and x_3 must be 1. Logical form:

$$x_1 = 0 \Rightarrow x_2 + x_3 \geq 2.$$

Linear constraint:

$$x_2 + x_3 \geq 2(1 - x_1).$$

6. At most one variable can be 1 (n variables).

$$\sum_{i=1}^{n} x_i \leq 1.$$

7. At most r variables can be 1 (n variables).

$$\sum_{i=1}^{n} x_i \leq r.$$

8. At least r variables must be 1 (n variables).

$$\sum_{i=1}^{n} x_i \geq r.$$

9. If $x_1 = 0$, then at least r of $x_2, \ldots, x_n$ must be 1.

$$\sum_{i=2}^{n} x_i \geq r(1 - x_1).$$

10. If $x_1 = 1$, then at most r of $x_2, \dots, x_n$ can be 1.

$$\sum_{i=2}^{n} x_i \le r + (n-1-r)(1-x_1),$$

with $n-1-r$ serving as a problem-specific big-M constant.

2.3 Modeling Exercise: Store Opening with Logical Constraints

A company can open up to 4 stores from 6 locations $L_1, \dots, L_6$. Rules: L_1 and L_2 cannot both open; L_3 and L_4 must open together if either is opened; each open store must have between 3 and 10 staff.

Decision variables. Store opening:

$$y_i = \begin{cases} 1, & \text{if store at location } L_i \text{ is opened,} \\ 0, & \text{otherwise,} \end{cases} \qquad i = 1, \dots, 6.$$

Staffing:

$$s_i = \text{number of staff assigned to store } L_i, \quad i = 1, \dots, 6.$$

Domains: $y_i \in \{0, 1\}$, $s_i \in \mathbb{Z}_{\ge 0}$.

Constraints. Open at most 4 stores:

$$\sum_{i=1}^{6} y_i \le 4.$$

L_1 and L_2 cannot both open:

$$y_1 + y_2 \le 1.$$

L_3 and L_4 must open together:

$$y_3 - y_4 = 0.$$

Staff bounds per open store:

$$3y_i \le s_i \le 10y_i, \quad i = 1, \dots, 6.$$

Objective. Let F_i be fixed opening cost and c_i cost per staff member for store i. Minimize

$$\min\ z = \sum_{i=1}^{6} \big(F_i y_i + c_i s_i\big).$$

3 Big-M and Conditional Constraints (MILP)

3.1 Conceptual Questions

1. What is the Big-M method in MILP? The Big-M method introduces large constants M in linear inequalities to activate or deactivate constraints conditionally based on the value of a binary variable, thereby modeling logical implications or conditional bounds within a mixed-integer linear framework.[22]

2. Where and why is it used? It is used to model constraints of the form "if $y = 1$, then a certain relation must hold; if $y = 0$, the constraint is relaxed," such as conditional capacities, setup-dependent production, or piecewise definitions.[23] By using sufficiently large M, the inactive side of the implication becomes nonbinding, while keeping the formulation linear and compatible with MILP solvers.[24]

3. Simple illustrative example of Big-M usage. If a machine can produce quantity x only if it is on ($y = 1$), with capacity U, we can write

$$0 \le x \le Uy.$$

Here U acts as M: when $y = 0$, $x = 0$; when $y = 1$, x can range up to U.[25]

3.2 Modeling Exercise (Machine on/off)

A machine can be turned on/off. If on, minimum production 20, maximum 100; if off, production 0. Production cost 5\$/unit, fixed cost 100\$.

[22] Rao, Sec. 7.12 and Ch. 10; Nocedal and Wright, Sec. 12.4.
[23] Rao, Ch. 10; Iqbal, Ch. 7.
[24] Nocedal and Wright, Sec. 12.4.
[25] Standard MILP modeling pattern; see Rao, Sec. 10.5.

Variables. Binary on/off:

$$y = \begin{cases} 1, & \text{machine on,} \\ 0, & \text{machine off,} \end{cases}$$

production quantity:

$$x \geq 0.$$

Conditional production via Big-M.

$$x \leq 100y, \quad x \geq 20y.$$

Objective.

$$\min \ z = 100\, y + 5x.$$

4 Linearizing Min/Max and Piecewise Functions

4.1 Conceptual Questions

1. Why is it important to linearize min and max operators in MILP? MILP solvers require linear constraints and linear objectives; operators like min, max, and nonlinear piecewise definitions are not directly admissible.[26] Linearizing min, max, and piecewise-linear functions using auxiliary variables and additional linear constraints allows us to keep the model within the MILP class so that standard branch-and-bound/interior-point techniques can be applied.[27]

2. How can an auxiliary variable represent min or max? For $z = \min(x_1, x_2)$, introduce z and enforce

$$z \leq x_1, \quad z \leq x_2,$$

[26] Rao, Sec. 12.2; Nocedal and Wright, Sec. 12.4.
[27] Iqbal, Ch. 7.

plus additional constraints (possibly with binaries and big-M) so that z equals the smaller argument.[28] For $w = \max(x_1, x_2)$, enforce

$$w \geq x_1, \quad w \geq x_2,$$

with similar logic.

3. How can piecewise-linear cost functions be handled in MILP? They are modeled by dividing the domain into segments, introducing auxiliary segment variables, and using binaries (or SOS2 sets) to ensure only relevant segments are active; the cost is then a linear combination of these segment variables.[29]

4.2 Modeling Exercise (Projects A and B)

Let x_A, x_B be budgets and $z = \min(x_A, x_B)$.

1. Linearize $z = \min(x_A, x_B)$. Simplest formulation (relying on objective to tighten):

$$z \leq x_A, \quad z \leq x_B.$$

If we need an exact min independent of the objective, introduce binary y and big-M:

$$\begin{aligned} z &\geq x_A - M(1-y), \\ z &\geq x_B - My, \\ z &\leq x_A, \\ z &\leq x_B. \end{aligned}$$

2. Piecewise bonus for project A.

$$\text{bonus}_A = \begin{cases} 0, & x_A \leq 50, \\ 10, & x_A > 50. \end{cases}$$

Binary indicator:

$$y_A = \begin{cases} 1, & x_A > 50, \\ 0, & x_A \leq 50. \end{cases}$$

[28] Rao, Sec. 12.2.

[29] Rao, Sec. 12.2; Nocedal and Wright, Sec. 12.4.

Big-M_A bound on x_A:

$$x_A \le 50 + M_A y_A, \quad x_A \ge 50 y_A, \quad \text{bonus}_A = 10 y_A.$$

3. Same for project B. Introduce y_B, bonus$_B$, and:

$$x_B \le 50 + M_B y_B, \quad x_B \ge 50 y_B, \quad \text{bonus}_B = 10 y_B.$$

5 A MILP Example (3 products, 2 machines)

5.1 Model Formulation

Products P_1, P_2, P_3 and machines M_1, M_2. Each product can be produced on one machine only; each machine can be on/off with setup time 10 hours. Product data (profit, labor, minimum, maximum) are as in the assignment.

Variables. Machine on/off:

$$y_j \in \{0, 1\}, \quad j = 1, 2.$$

Assignment:

$$a_{ij} \in \{0, 1\}, \quad i = 1, 2, 3,\ j = 1, 2.$$

Production:

$$x_{ij} \in \mathbb{Z}_{\ge 0}, \quad i = 1, 2, 3,\ j = 1, 2.$$

Constraints. Each product assigned to exactly one machine:

$$\sum_{j=1}^{2} a_{ij} = 1, \quad i = 1, 2, 3.$$

Assignment allowed only if machine is on:

$$a_{ij} \le y_j, \quad i = 1, 2, 3,\ j = 1, 2.$$

Production bounds (min/max per product):

$$\underline{q}_i\, a_{ij} \le x_{ij} \le \overline{q}_i\, a_{ij}, \quad i = 1, 2, 3,\ j = 1, 2.$$

Total labor including setup (setup time 10 per active machine, labor per unit ℓ_i):

$$10(y_1 + y_2) + \sum_{i=1}^{3} \sum_{j=1}^{2} \ell_i x_{ij} \le 200.$$

Objective. Let p_i be profit per unit of P_i and F_j fixed setup cost of machine M_j. Maximize net profit:

$$\max\ z = \sum_{i=1}^{3} \sum_{j=1}^{2} p_i x_{ij} - \sum_{j=1}^{2} F_j y_j.$$