# IE 203 PS 1 - Solutions

## Q1. Assignment Problem

### a) General Formulation

**Indices and Parameters:**

- $i, j \in \{1, \ldots, n\}$: workers and tasks
- $c_{ij}$: cost of assigning worker $i$ to task $j$

**Decision Variables:**

- $x_{ij} \in \{0, 1\}$: equals 1 if worker $i$ is assigned to task $j$, 0 otherwise

**Formulation:**

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} = 1, \quad \forall i \in \{1, \ldots, n\} \quad \text{(Worker assignment)} \\
& \sum_{i=1}^{n} x_{ij} = 1, \quad \forall j \in \{1, \ldots, n\} \quad \text{(Task coverage)} \\
& x_{ij} \in \{0, 1\}, \quad \forall i, j
\end{aligned}
$$

**Explanation:** The worker assignment constraints ensure each worker is assigned to exactly one task. The task coverage constraints ensure each task has exactly one worker assigned to it.

### b) Numerical Instance

**Decision Variables:** $x_{ij} \in \{0, 1\}$ for workers $i \in \{A, B, C, D\}$ and procedures $j \in \{1, 2, 3, 4\}$

**Objective:**

$$\min Z = 45x_{A,1} + 60x_{A,2} + \cdots + 40x_{D,4}$$

**Constraints:**

$$
\text{Worker assignment:} \quad
\begin{cases}
x_{A,1} + x_{A,2} + x_{A,3} + x_{A,4} = 1 \\
x_{B,1} + x_{B,2} + x_{B,3} + x_{B,4} = 1 \\
x_{C,1} + x_{C,2} + x_{C,3} + x_{C,4} = 1 \\
x_{D,1} + x_{D,2} + x_{D,3} + x_{D,4} = 1
\end{cases}
$$

$$
\text{Task assignment:} \quad
\begin{cases}
x_{A,1} + x_{B,1} + x_{C,1} + x_{D,1} = 1 \\
x_{A,2} + x_{B,2} + x_{C,2} + x_{D,2} = 1 \\
x_{A,3} + x_{B,3} + x_{C,3} + x_{D,3} = 1 \\
x_{A,4} + x_{B,4} + x_{C,4} + x_{D,4} = 1
\end{cases}
$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j$$

## c) Infeasibility Constraint

**Construction:** To prevent Dr. A from being assigned to Procedure 2, add the prohibition constraint:

$$x_{A,2} = 0 \tag{1}$$

**Rationale:** The explicit constraint $x_{A,2} = 0$ prevents Dr. A from being assigned to Procedure 2. This ensures Dr. A must be assigned to exactly one of Procedures 1, 3, or 4.

## d) Exclusion Constraint

**Construction:** To prevent Dr. B from being assigned to either Procedure 1 or Procedure 3, add the prohibition constraint:

$$x_{B,1} + x_{B,3} = 0 \tag{2}$$

**Rationale:** The explicit constraint $x_{B,1}+x_{B,3} = 0$ prevents Dr. B from being assigned to Procedures 1 or 3. This ensures Dr. B must be assigned to exactly one of Procedures 2 or 4.

## e) Conditional Dependency Constraint

**Construction:** If Dr. C is assigned to Procedure 4, then Dr. D must be assigned to Procedure 2:

$$x_{C,4} \leq x_{D,2} \tag{3}$$

**Rationale:** This inequality enforces logical implication:

- If $x_{C,4} = 1$, then the constraint becomes $1 \leq x_{D,2}$, forcing $x_{D,2} = 1$
- If $x_{C,4} = 0$, then the constraint becomes $0 \leq x_{D,2}$, leaving $x_{D,2}$ free

# Q2. Generalized Assignment Problem

## a) General Formulation

**Indices and Parameters:**

- $i \in \{1, \ldots, m\}$: agents
- $j \in \{1, \ldots, n\}$: tasks
- $c_{ij}$: cost of assigning agent $i$ to task $j$
- $a_{ij}$: resource consumption when agent $i$ performs task $j$
- $b_i$: resource capacity of agent $i$

**Decision Variables:**

- $x_{ij} \in \{0, 1\}$: equals 1 if agent $i$ is assigned to task $j$, 0 otherwise

**Formulation:**

$$\begin{aligned}
\min \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{i=1}^{m} x_{ij} = 1, \quad \forall j \in \{1, \ldots, n\} \quad \text{(Task coverage)} \\
& \sum_{j=1}^{n} a_{ij} x_{ij} \leq b_i, \quad \forall i \in \{1, \ldots, m\} \quad \text{(Agent capacity)} \\
& x_{ij} \in \{0, 1\}, \quad \forall i, j
\end{aligned}$$

**Explanation:** The task coverage constraints ensure each task is assigned to exactly one agent (equality used because all tasks must be completed). The agent capacity constraints ensure total resource consumption by tasks assigned to agent $i$ does not exceed capacity (inequality used because spare capacity is acceptable).

## b) Numerical Instance

**Indices:** $i \in \{1, 2\}$ (servers), $j \in \{1, 2, 3, 4, 5\}$ (jobs)

**Parameters:** $b_1 = 100$ GB, $b_2 = 120$ GB; Job sizes (independent of server): $a_{i,j}$ for $j = 1, 2, 3, 4, 5$ are 30, 50, 40, 20, 60 GB respectively; Processing costs: $c_{1,j} = [8, 12, 10, 6, 15]$ and $c_{2,j} = [10, 9, 11, 8, 13]$ dollars

**Objective:**

$$\min Z = 8x_{1,1} + 12x_{1,2} + 10x_{1,3} + 6x_{1,4} + 15x_{1,5} + 10x_{2,1} + 9x_{2,2} + 11x_{2,3} + 8x_{2,4} + 13x_{2,5}$$

**Constraints:**

$$\text{Task coverage:} \quad \begin{cases} x_{1,1} + x_{2,1} = 1 \\ x_{1,2} + x_{2,2} = 1 \\ x_{1,3} + x_{2,3} = 1 \\ x_{1,4} + x_{2,4} = 1 \\ x_{1,5} + x_{2,5} = 1 \end{cases}$$

$$\text{Server capacity:} \quad \begin{cases} 30x_{1,1} + 50x_{1,2} + 40x_{1,3} + 20x_{1,4} + 60x_{1,5} \leq 100 \\ 30x_{2,1} + 50x_{2,2} + 40x_{2,3} + 20x_{2,4} + 60x_{2,5} \leq 120 \end{cases}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j$$

## c) Conflict Constraint

**Construction:** To prevent Job 1 and Job 2 from being on the same server:

$$x_{i,1} + x_{i,2} \leq 1, \quad \forall i \in \{1, 2\} \tag{4}$$

**Rationale:** For each server $i$, at most one of the two jobs can be assigned. This ensures they cannot both be on Server 1, and they cannot both be on Server 2. The task coverage constraint still requires both jobs to be assigned, so they will necessarily go to different servers.

## d) Cardinality Constraint

**Construction:** To limit each server to at most 3 jobs:

$$\sum_{j=1}^{5} x_{i,j} \leq 3, \quad \forall i \in \{1, 2\} \tag{5}$$

**Rationale:** This is a cardinality constraint that counts the number of tasks, independent of resource consumption. Unlike the capacity constraint which uses weighted sum $\sum a_{ij}x_{ij}$, this uses an unweighted sum to limit task count.

## e) Conditional Resource Overhead

**Construction:** If Server 1 processes Job 3, it requires an additional 10 GB. The Server 1 capacity constraint from part (b) is **modified** as follows:

$$30x_{1,1} + 50x_{1,2} + 40x_{1,3} + 20x_{1,4} + 60x_{1,5} + 10x_{1,3} \leq 100 \tag{6}$$

which simplifies to:

$$30x_{1,1} + 50x_{1,2} + 50x_{1,3} + 20x_{1,4} + 60x_{1,5} \leq 100 \tag{7}$$

**Rationale:** This constraint **replaces** the Server 1 capacity constraint from part (b). This is a fixed-charge concept in resource terms. The resource coefficient for Job 3 effectively increases from 40 GB to 50 GB (40 + 10) when selected. The binary nature of $x_{1,3}$ ensures the overhead is only counted if the job is assigned.

## f) Minimum Utilization Constraint

**Construction:** Server 2 must use at least 70% of its capacity:

$$30x_{2,1} + 50x_{2,2} + 40x_{2,3} + 20x_{2,4} + 60x_{2,5} \geq 0.7 \times 120 = 84 \tag{8}$$

**Rationale:** This provides a lower bound on resource utilization. While standard capacity constraints give upper bounds, this ensures minimum usage for economic efficiency or service level requirements.

# Q3. Traveling Salesperson Problem

## a) General Formulation

**Indices and Parameters:**

- $i, j \in \{1, \dots, n\}$: cities
- $d_{ij}$: distance from city $i$ to city $j$, with $d_{ii} = \infty$ for all $i$

**Note:** Setting $d_{ii} = \infty$ (diagonal elements infinite) prevents self-loops in the solution.
**Decision Variables:**

- $x_{ij} \in \{0, 1\}$: equals 1 if the tour travels directly from city $i$ to city $j$, 0 otherwise
- $u_i \geq 0$: auxiliary position variable representing when city $i$ is visited in the tour

**Formulation:**

$$\begin{aligned}
\min \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} = 1, \quad \forall i \quad \text{(Outgoing flow)} \\
& \sum_{i=1}^{n} x_{ij} = 1, \quad \forall j \quad \text{(Incoming flow)} \\
& u_i - u_j + n \cdot x_{ij} \leq n - 1, \quad \forall i, j \in \{2, \dots, n\}, i \neq j \quad \text{(Subtour elimination)} \\
& u_1 = 0 \\
& 1 \leq u_i \leq n - 1, \quad \forall i \in \{2, \dots, n\} \\
& x_{ij} \in \{0, 1\}, \quad \forall i, j
\end{aligned}$$

**Explanation:** The outgoing and incoming flow constraints ensure each city has exactly one outgoing and one incoming edge. The subtour elimination constraints (Miller-Tucker-Zemlin formulation) use position variables $u_i$ to prevent disconnected subtours: when $x_{ij} = 1$, we require $u_i < u_j$, forcing visits in increasing order from the depot (city 1).

### Alternative Model: Dantzigâ€“Fulkersonâ€“Johnson (DFJ) Formulation

The DFJ formulation eliminates subtours directly without auxiliary variables, using exponential subtour elimination constraints.

**Indices and Parameters:**

- $i, j \in \{1, \dots, n\}$: cities
- $d_{ij}$: distance from city $i$ to city $j$

**Decision Variables:**

- $x_{ij} \in \{0, 1\}$: equals 1 if the tour travels directly from city $i$ to city $j$, 0 otherwise

**Formulation:**

$$\begin{aligned}
\min \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} = 1, \quad \forall i \quad \text{(Outgoing flow)} \\
& \sum_{i=1}^{n} x_{ij} = 1, \quad \forall j \quad \text{(Incoming flow)} \\
& \sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \quad \forall S \subset \{1, \dots, n\}, 2 \leq |S| \leq n - 1 \quad \text{(Subtour elimination)} \\
& x_{ij} \in \{0, 1\}, \quad \forall i, j
\end{aligned}$$

**Example: Handling Subtours in DFJ**

Consider $n = 6$ cities. The flow conservation constraints (outgoing and incoming flow) allow solutions like:

- Subtour 1: $1 \to 2 \to 3 \to 1$ (edges: $x_{1,2} = 1, x_{2,3} = 1, x_{3,1} = 1$)
- Subtour 2: $4 \to 5 \to 6 \to 4$ (edges: $x_{4,5} = 1, x_{5,6} = 1, x_{6,4} = 1$)

This solution satisfies all flow conservation constraints (each city has exactly one incoming and one outgoing edge) but forms two disconnected tours instead of one.

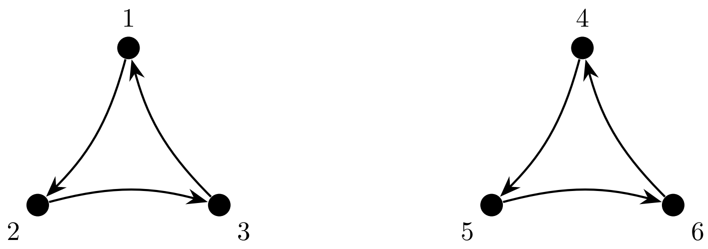


Figure 1: Two disconnected subtours: $1 \to 2 \to 3 \to 1$ and $4 \to 5 \to 6 \to 4$. Each subtour satisfies flow conservation but violates DFJ subtour elimination constraints.

The DFJ subtour elimination constraint for subset $S = \{1, 2, 3\}$ requires:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \implies x_{1,1} + x_{1,2} + x_{1,3} + x_{2,1} + x_{2,2} + x_{2,3} + x_{3,1} + x_{3,2} + x_{3,3} \leq 2$$

The subtour solution violates this: $x_{1,2} + x_{2,3} + x_{3,1} = 3 > 2$. Similarly, for $S = \{4, 5, 6\}$:

$$x_{4,5} + x_{5,6} + x_{6,4} = 3 > 2$$

Thus, the DFJ constraints explicitly prevent disconnected subtours by limiting edges within any proper subset.

**Key Differences from MTZ:**

- **MTZ**: Uses $O(n)$ auxiliary continuous variables ($u_i$) with $O(n^2)$ polynomial constraints; weaker LP relaxation but compact formulation
- **DFJ**: Uses only binary variables ($x_{ij}$) with $O(2^n)$ exponential constraints; stronger LP relaxation but impractical to enumerate

**How DFJ is Actually Solved:**

Because a 50-city problem would require roughly $1.12 \times 10^{15}$ subtour elimination constraints (more than can fit in memory), DFJ uses **branch-and-cut** with lazy constraints:

1. Solve the Assignment Problem (TSP without subtour constraints)
2. Check the solution: if no subtours exist, done!
3. If a subtour is found (e.g., cities $1 \to 2 \to 5 \to 1$), add **only that specific constraint**:

$$x_{1,2} + x_{2,5} + x_{5,1} \leq 2$$

4. Re-solve and repeat until a single tour is formed

This iterative approach only generates the small fraction of constraints actually needed to eliminate subtours in practice, making DFJ the foundation of modern exact TSP solvers. The subtour constraints are added dynamically when violated subtours are detected in LP solutions (separation problem).

## b) Numerical Instance

**Parameters:** $n = 5$ locations, distance matrix $D$ as provided

**Decision Variables:** $x_{ij} \in \{0, 1\}$, $u_i \geq 0$ (position variables)

**Objective:**

$$\begin{aligned}
\min Z = {} & Mx_{1,1} + 10x_{1,2} + 15x_{1,3} + 20x_{1,4} + 25x_{1,5} \\
& + 10x_{2,1} + Mx_{2,2} + 12x_{2,3} + 18x_{2,4} + 22x_{2,5} \\
& + 15x_{3,1} + 12x_{3,2} + Mx_{3,3} + 8x_{3,4} + 16x_{3,5} \\
& + 20x_{4,1} + 18x_{4,2} + 8x_{4,3} + Mx_{4,4} + 15x_{4,5} \\
& + 25x_{5,1} + 22x_{5,2} + 16x_{5,3} + 15x_{5,4} + Mx_{5,5}
\end{aligned}$$

where $M$ is a very large number (e.g., $M = 10000$) that prevents self-loops by making $x_{ii} = 1$ prohibitively expensive.

**Constraints:**

$$\text{Outgoing flow:} \quad \begin{cases} x_{1,1} + x_{1,2} + x_{1,3} + x_{1,4} + x_{1,5} = 1 \\ x_{2,1} + x_{2,2} + x_{2,3} + x_{2,4} + x_{2,5} = 1 \\ x_{3,1} + x_{3,2} + x_{3,3} + x_{3,4} + x_{3,5} = 1 \\ x_{4,1} + x_{4,2} + x_{4,3} + x_{4,4} + x_{4,5} = 1 \\ x_{5,1} + x_{5,2} + x_{5,3} + x_{5,4} + x_{5,5} = 1 \end{cases}$$

$$\text{Incoming flow:} \quad \begin{cases} x_{1,1} + x_{2,1} + x_{3,1} + x_{4,1} + x_{5,1} = 1 \\ x_{1,2} + x_{2,2} + x_{3,2} + x_{4,2} + x_{5,2} = 1 \\ x_{1,3} + x_{2,3} + x_{3,3} + x_{4,3} + x_{5,3} = 1 \\ x_{1,4} + x_{2,4} + x_{3,4} + x_{4,4} + x_{5,4} = 1 \\ x_{1,5} + x_{2,5} + x_{3,5} + x_{4,5} + x_{5,5} = 1 \end{cases}$$

$$\text{Subtour elimination:} \quad \begin{cases} u_2 - u_3 + 5x_{2,3} \leq 4 \\ u_2 - u_4 + 5x_{2,4} \leq 4 \\ u_2 - u_5 + 5x_{2,5} \leq 4 \\ \vdots \\ u_5 - u_4 + 5x_{5,4} \leq 4 \quad \text{(for all } i, j \in \{2, ..., 5\}, i \neq j) \end{cases}$$

$$\begin{aligned}
& u_1 = 0 \\
& 1 \leq u_i \leq 4, \quad \forall i \in \{2, 3, 4, 5\} \\
& x_{ij} \in \{0, 1\}, \quad \forall i, j
\end{aligned}$$

## c) Precedence Constraint

**Construction:** Location 2 must be visited immediately before Location 3:

$$x_{2,3} = 1 \tag{9}$$

**Rationale:** Forcing this edge into the solution ensures direct travel from Location 2 to Location 3 with no intermediate stops.

## d) Early Visit Constraint

**Construction:** Location 5 must be first or second stop after depot:

$$u_5 \leq 2 \tag{10}$$

**Rationale:** Since $u_1 = 0$ (depot) and positions increment, requiring $u_5 \leq 2$ means Location 5 has position 1 or 2.

## e) Path Exclusion

**Construction:** Direct travel between locations 1 and 2, and between locations 3 and 4, cannot both be used:

$$x_{1,2} + x_{2,1} + x_{3,4} + x_{4,3} \leq 1 \tag{11}$$

**Rationale:** At most one of these edges can be in the solution, creating mutual exclusion between route segments.

# Q5. Set Covering Problem

## a) General Formulation

**Indices and Parameters:**

- $i \in I$: elements that must be covered
- $j \in J$: available subsets
- $f_j$: cost of selecting subset $j$
- $a_{ij}$: coverage matrix where $a_{ij} = 1$ if subset $j$ covers element $i$

**Decision Variables:**

- $x_j \in \{0, 1\}$: equals 1 if subset $j$ is selected, 0 otherwise

**Formulation:**

$$\begin{aligned} \min \quad & \sum_{j \in J} f_j x_j \\ \text{s.t.} \quad & \sum_{j \in J} a_{ij} x_j \geq 1, \quad \forall i \in I \quad \text{(Coverage)} \\ & x_j \in \{0, 1\}, \quad \forall j \in J \end{aligned}$$

**Explanation:** The coverage constraints ensure each element is covered by at least one selected subset. The inequality ($\geq$) allows redundant coverage, where multiple subsets can cover the same element.

## b) Numerical Instance

**Parameters:** Costs $f = [30, 25, 35, 20]$, coverage matrix as given
**Objective:**

$$\min Z = 30x_1 + 25x_2 + 35x_3 + 20x_4$$

**Constraints:**

$$\begin{aligned} x_1 + x_2 &\geq 1 \quad \text{(Neighborhood 1)} \\ x_1 + x_3 &\geq 1 \quad \text{(Neighborhood 2)} \\ x_2 + x_3 &\geq 1 \quad \text{(Neighborhood 3)} \\ x_3 + x_4 &\geq 1 \quad \text{(Neighborhood 4)} \\ x_4 &\geq 1 \quad \text{(Neighborhood 5)} \\ x_j &\in \{0, 1\}, \quad \forall j \in \{1, 2, 3, 4\} \end{aligned}$$

## c) Conditional Opening Constraint

**Construction:** If station 2 is opened, then station 4 must also be opened:

$$x_2 \leq x_4 \tag{12}$$

**Rationale:** This is a logical implication constraint. If $x_2 = 1$, then the constraint forces $x_4 \geq 1$, hence $x_4 = 1$ (binary). If $x_2 = 0$, then $x_4$ is free to be 0 or 1.

## d) Redundancy Constraint

**Construction:** Neighborhood 3 must be covered by at least 2 stations. Modify the neighborhood 3 constraint from part (b):

$$x_2 + x_3 \geq 2 \tag{13}$$

**Rationale:** This ensures both stations 2 and 3 must be opened (since they are the only ones covering neighborhood 3). This provides redundancy in case one station becomes unavailable.

# Q6. Facility Location Problem

## a) General Formulation

**Indices and Parameters:**

- $i \in I$: potential facility locations
- $j \in J$: customers
- $f_i$: fixed cost of opening facility $i$
- $K_i$: capacity of facility $i$ (units)
- $d_j$: demand of customer $j$ (units)
- $c_{ij}$: transportation cost per unit from facility $i$ to customer $j$

**Decision Variables:**

- $y_i \in \{0, 1\}$: equals 1 if facility $i$ is opened, 0 otherwise
- $x_{ij} \geq 0$: units of demand of customer $j$ served from facility $i$

**Formulation:**

$$\begin{aligned}
\min \quad & \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{i \in I} x_{ij} = d_j, \quad \forall j \in J \quad \text{(Demand satisfaction)} \\
& \sum_{j \in J} x_{ij} \leq K_i y_i, \quad \forall i \in I \quad \text{(Capacity)} \\
& x_{ij} \geq 0, \quad \forall i, j \\
& y_i \in \{0, 1\}, \quad \forall i
\end{aligned}$$

**Explanation:** The objective minimizes total fixed costs (opening facilities) plus total transportation costs. The demand satisfaction constraints ensure each customer's demand is fully met. The capacity constraints ensure facility $i$ can only serve customers if opened ($y_i = 1$), and total shipments cannot exceed capacity. This is a mixed-integer program (MIP) with both binary ($y_i$) and continuous ($x_{ij}$) variables.

## b) Numerical Instance

**Parameters:** $f = [120, 100, 140, 110]$, $K = [200, 180, 220, 190]$, $d = [60, 50, 70, 55, 65]$, cost matrix $C$ as provided

**Decision Variables:** $y_i \in \{0, 1\}$ for $i \in \{1, 2, 3, 4\}$, $x_{ij} \geq 0$ for all $i, j$

**Objective:**

$$\min Z = 120y_1 + 100y_2 + 140y_3 + 110y_4 + c_{1,1}x_{1,1} + c_{1,2}x_{1,2} + \cdots + c_{4,5}x_{4,5}$$

**Constraints:**

$$\text{Demand satisfaction:} \quad \begin{cases} x_{1,1} + x_{2,1} + x_{3,1} + x_{4,1} = 60 \\ x_{1,2} + x_{2,2} + x_{3,2} + x_{4,2} = 50 \\ x_{1,3} + x_{2,3} + x_{3,3} + x_{4,3} = 70 \\ x_{1,4} + x_{2,4} + x_{3,4} + x_{4,4} = 55 \\ x_{1,5} + x_{2,5} + x_{3,5} + x_{4,5} = 65 \end{cases}$$

$$\text{Capacity constraints:} \quad \begin{cases} x_{1,1} + x_{1,2} + x_{1,3} + x_{1,4} + x_{1,5} \leq 200y_1 \\ x_{2,1} + x_{2,2} + x_{2,3} + x_{2,4} + x_{2,5} \leq 180y_2 \\ x_{3,1} + x_{3,2} + x_{3,3} + x_{3,4} + x_{3,5} \leq 220y_3 \\ x_{4,1} + x_{4,2} + x_{4,3} + x_{4,4} + x_{4,5} \leq 190y_4 \end{cases}$$

$$x_{ij} \geq 0, \quad \forall i, j$$

$$y_i \in \{0, 1\}, \quad \forall i \in \{1, 2, 3, 4\}$$

### c) Strategic Constraints

**Construction:**

*(i) At least one of warehouses 1 and 3 must be opened:*

$$y_1 + y_3 \geq 1 \tag{14}$$

*(ii) Exactly 2 warehouses must be opened:*

$$y_1 + y_2 + y_3 + y_4 = 2 \tag{15}$$

*(iii) Customer 2 must receive all demand from a single warehouse:*
Introduce binary variables $z_{i2} \in \{0, 1\}$ indicating if warehouse $i$ serves customer 2:

$$x_{i2} \leq 50 \cdot z_{i2}, \quad \forall i \in \{1, 2, 3, 4\} \tag{16}$$

$$\sum_{i=1}^{4} z_{i2} = 1 \tag{17}$$

**Rationale:** The first constraint ensures warehouse $i$ can only ship to customer 2 if selected ($z_{i2} = 1$), with shipment capped at the demand (50 units). The second constraint ensures exactly one warehouse is selected to serve customer 2. Together with the demand satisfaction constraint (which requires total shipment to customer 2 equals 50), this forces the selected warehouse to provide all 50 units. This formulation represents single-sourcing requirements common in contractual obligations or delivery efficiency.