JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Algorithms for Data Science

Optimization: Integer Programming

Integer Programming

Integer programming (IP) is an extension of linear programming where some or all variables are restricted to integer values.

- **Standard Form:**

$$\text{Maximize or Minimize } c^T x \text{ s.t. } Ax \leq b, x \in \mathbb{Z}^n$$

- **Key Differences from LP:**

  LP allows for continuous variables, IP requires discrete variables.
  Integer constraints make the problem combinatorial.

Types of Integer Programming

Pure IP

- All decision variables are integers.
- E.g. Assigning workers to tasks where partial assignments are not allowed.
- Use Case: Resource Allocation

Mixed IP (MIP)

- Some variables are integers, others continuous.
- E.g. Optimizing costs with integer constraints but continuous costs.
- Use Case: Portfolio Optimization

Binary IP

- Variables are restricted to 0 or 1 (binary).
- E.g. Selecting projects to fund given budget.
- Use Case: Project Selction

Methods for Solving IP Problems

Branch & Bound

- Systematically divides the problem into smaller subproblems to exclude infeasible solutions.
- Process:
  - Solve the relaxed LP problem.
  - Identify fractional variables.
  - Prune subproblems

Cutting Plane

- Iteratively refines the feasible region by adding linear constraints to exclude non-integer solutions.
- Process:
  - Solve the relaxed LP problem.
  - Identify violated integer constraints.
  - Add cutting planes.

Heuristic Methods

- Approximation techniques that provide near-optimal solutions quickly.
- Examples:
  - Genetic Algorithms
  - Simulated Annealing

Integer Programming Example

- **Objective:** Assign tasks to workers while minimizing the total costs.
- **Constraints:**
  - Each task must be assigned to exactly one coworker.
  - Each worker can handle only a limited number of tasks.
  - Assignments must be binary.
- **Mathematical Formulation:**
  - Let x_{ij} represent whether task j is assigned to worker i or not.
  - Objective Function: $\text{Minimize} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$
  - Constraints:

$$\sum_{i=1}^{m} x_{ij} = 1, \forall j$$

(Each task assigned to one worker)

$$\sum_{j=1}^{n} x_{ij} \leq k_i, \forall i$$

(Worker capacity constraint)

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

(Binary decision variables)

JOHNS HOPKINS

WHITING SCHOOL *of* ENGINEERING