University of Saida Dr Moulay Tahar
Faculty of M.I.T
Computer Science Department

FINAL EXAM – SOLUTION

Module: Operations Research

Instructor: Dr. SOUIDI

Academic Year: 2025-2026 **Promotion:** 4^{th} Year Engineer

Total Points: 20

Exercise 1: [5 Points]

Each answer is worth 0.5 points. Full credit requires clear, accurate definitions and explanations.

1. **What does ILP mean and when should it be used?**

   Answer: ILP stands for **Integer Linear Programming**. It is used when we need to solve optimization problems with linear objective functions and linear constraints, where some or all decision variables must take integer values (whole numbers). ILP is essential for problems involving discrete decisions such as:

   - Binary choices (yes/no, select/don't select)
   - Counting discrete items (number of projects, machines, etc.)
   - Network flow with discrete units

   Example: Resource allocation, scheduling, facility location problems.

2. **Why do we need to use Binary Integer Linear Programming (Binary ILP)?**

   Answer: Binary ILP is needed when decision variables can only take values 0 or 1, representing yes/no decisions or on/off states. Applications include:

   - Project selection (fund or not fund)
   - Assignment problems (assign or not assign)
   - Network design (include edge or not)
   - Logical constraints (conditional constraints like "if-then" statements)

   Binary variables allow us to model logical and Boolean constraints that cannot be expressed in continuous LP.

3. **What is the key difference between CPM (MPM) and PERT?**

   Answer:

   - **CPM (Critical Path Method / MPM):** Uses Activity-on-Node (AoN) representation, where nodes depict activities and arrows show dependencies, making it ideal for construction or repetitive projects with predictable timelines. Better for projects with well-known, stable durations.

- **PERT (Program Evaluation and Review Technique):** Uses Activity-on-Arrow (AoA) representation. Activities are represented as arrows, and nodes represent events (start/completion of activities).Pert is better for research and development projects with uncertain timelines.

4. **What does Convex Programming mean?**

   Answer: Convex Programming is the optimization of a convex objective function over a convex feasible region. A function f is convex if for any two points x, y and $\lambda \in [0, 1]$:

   $$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

   Key advantages:

   - Any local minimum is a global minimum
   - Widely applicable to real-world problems

5. **What is the key concept of Gradient Descent and how does it work?**

   Answer: Gradient Descent is an iterative optimization algorithm that minimizes a function by moving in the direction of steepest descent (negative gradient).

   Algorithm: $x^{k+1} = x^k - \alpha_k \nabla f(x^k)$

   Where:

   - $\nabla f(x^k)$ is the gradient at iteration k
   - α_k is the step size (learning rate)
   - The algorithm repeats until convergence

   Key features: Simple, scalable, requires only gradient computations, but can be slow for ill-conditioned problems.

6. **What do Heuristics and Metaheuristics mean?**

   Answer:

   - **Heuristics:** Problem-specific algorithms that find good (not necessarily optimal) solutions quickly. Examples: nearest neighbor for TSP, greedy algorithms.
   - **Metaheuristics:** General-purpose problem-solving frameworks that guide heuristics to escape local optima. Examples: Genetic Algorithms, Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Simulated Annealing.

   Metaheuristics are more flexible and can be applied to diverse problem classes.

7. **What does Branch and Bound mean and what is its main purpose?**

   Answer: Branch and Bound is an exact algorithm for solving integer and combinatorial optimization problems. It works by:

   - **Branching:** Divide the problem into smaller subproblems (creating a tree)
   - **Bounding:** Compute upper/lower bounds for each subproblem
   - **Pruning:** Eliminate subproblems that cannot contain an optimal solution

   Purpose: Find optimal solutions to ILP problems by systematically exploring the solution space while pruning infeasible branches. Guarantees optimality but can be computationally expensive.

8. **When do we need to use direct search methods instead of gradient-based methods?**

   Answer: Use direct search (also called derivative-free methods) when:

   - The function is non-differentiable or gradients are unavailable
   - The function is non-smooth or has discontinuities
   - Computing gradients is computationally expensive
   - The problem is noisy or has black-box structure
   - The objective function is discrete or non-convex

   Examples of direct search: Nelder-Mead simplex, pattern search, Bayesian optimization.

9. **What is the key difference between the Frank–Wolfe method and Kelley's Cutting Plane Method?**

   Answer:

   - **Frank–Wolfe:** A conditional gradient method that solves a linear approximation at each iteration (Linear Minimization Oracle). Iterates: $x^{k+1} = x^k + \gamma_k(s^k - x^k)$ where s^k minimizes the linear function over the feasible set. Slower but simpler.
   - **Kelley's Cutting Plane:** Builds a piecewise linear approximation of the objective function by adding cutting planes (linear constraints). Solves a sequence of LPs with increasing numbers of constraints. Converges faster but requires more complex LP solvers.

10. **Explain the concept of feasible region in linear programming and its role in finding optimal solutions.**

    Answer: The feasible region is the set of all points that satisfy all constraints in an LP problem. It is always a convex polyhedron (polytope) in linear programming.

    Key properties:

    - Defined by the intersection of half-spaces (linear inequalities)
    - If empty, the problem is infeasible
    - Extreme points (vertices) of the feasible region are candidate solutions
    - For LP, the optimal solution always occurs at a vertex (if bounded)

    Role in optimization: The Simplex algorithm exploits this by moving from vertex to vertex, improving the objective function until reaching optimality.

Exercise 2: Binary Integer Linear Programming [3 Points]

Problem: Research Grant Allocation

Question 2.1 [0.25 points]: Define binary decision variables x_i for $i = 1, \ldots, 6$.
Solution:

$$x_i = \begin{cases} 1 & \text{if project } i \text{ is funded} \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } i = 1, 2, 3, 4, 5, 6$$

Question 2.2 [2.75 points]: Formulate the complete binary ILP model.

Solution:

Objective Function (maximize total impact):

$$\text{maximize} \quad f = 12x_1 + 10x_2 + 14x_3 + 9x_4 + 16x_5 + 11x_6$$

[0.25 points]

Subject to Constraints:

(1) **Funding constraint:** $25x_1 + 18x_2 + 22x_3 + 15x_4 + 30x_5 + 20x_6 \leq 80$ [0.25 points]

(2) **At most 4 projects:** $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 4$ [0.25 points]

(3) **Conflict constraint (If P5, then not P2):** $x_5 + x_2 \leq 1$ [0.5 points]

(4) **At least one of P1 or P3:** $x_1 + x_3 \geq 1$ [0.5 points]

(5) **High-risk projects limit (P3, P5 are high-risk):** $x_3 + x_5 \leq 2$ [0.25 points]

(6) **Prerequisite dependency (If P6, then P4):** $x_6 \leq x_4$ [0.5 points]

(7) **Binary constraints:** $x_i \in \{0, 1\}$ for $i = 1, \ldots, 6$ [0.25 points]

Exercise 3: Project Scheduling Using MPM and PERT [6 Points]

Problem: University Registration System Project

Question 3.1 [2,5 points]: Construct the MPM network.
Solution:

Activity	Predecessors	Duration	ES	EF
A	—	3	0	3
B	A	4	3	7
C	A	3	3	6
D	B	5	7	12
E	B, C	6	7	13
F	D, E	4	13	17
G	F	3	17	20
H	G	2	20	22

Project completion time: $EF(H) = 22$ days

Activity	Successors	LS	LF	Slack
H	—	20	22	0
G	H	17	20	0
F	G	13	17	0
E	F	7	13	0
D	F	8	13	1
C	E	4	7	1
B	D, E	3	7	0
A	B, C	0	3	0

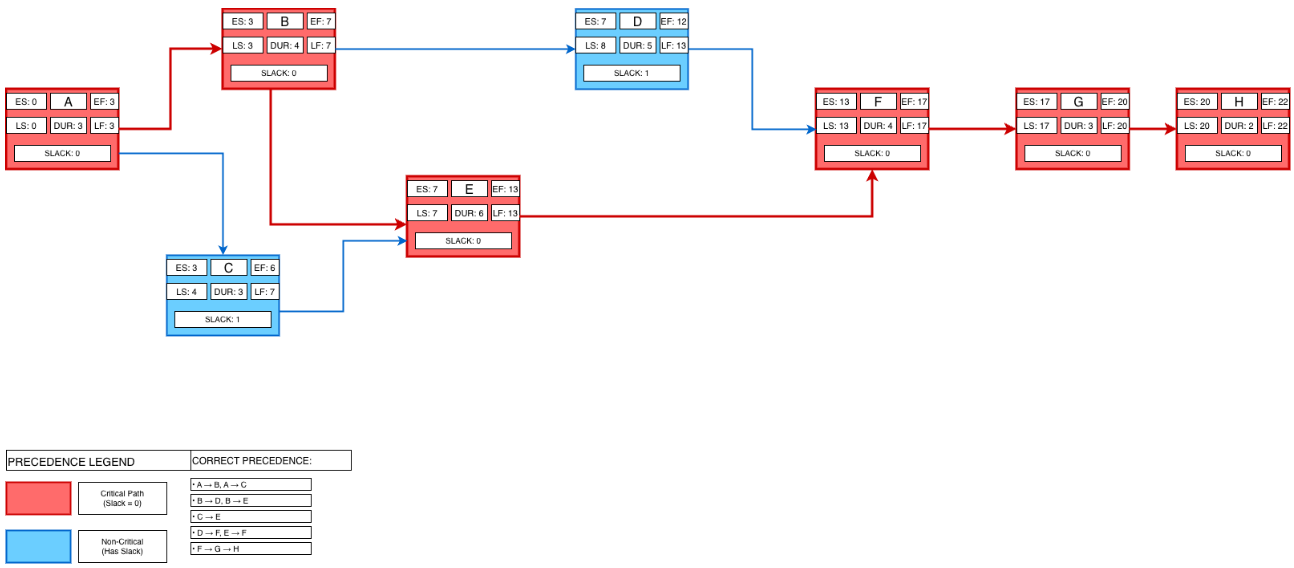

Figure 1: MPM Network Diagram

MPM Network Diagram:

The network should be drawn with each activity in a box:

Question 3.2 [2.5 points]: Construct the PERT network.
Solution:

PERT uses Activity-on-Arrow (AoA) representation. Activities are represented as arrows, and nodes represent events (start/completion of activities).
The network structure:

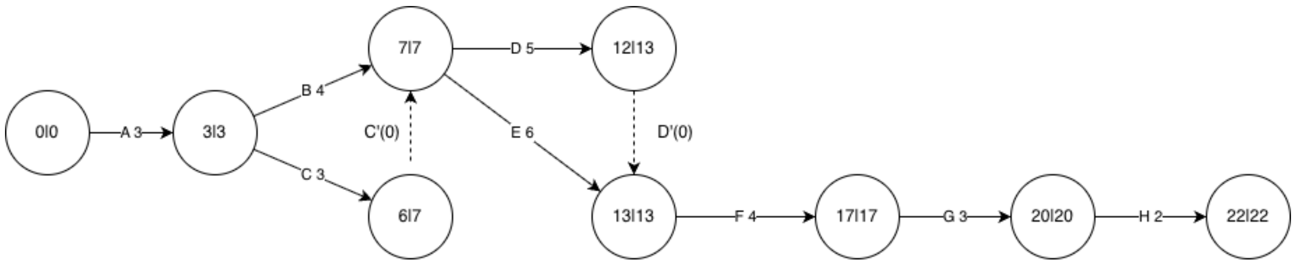

Figure 2: PERT Network Diagram

The PERT diagram should show all activities as labeled arrows between numbered nodes, with durations labeled on each arrow.

Question 3.3 [0,5 point]: Identify and highlight the critical path.
Solution:

The **critical path** consists of all activities with zero slack (total float = 0):

$$\boxed{\text{Critical Path: } A \to B \to E \to F \to G \to H}$$

These activities are:

- Activity A (3 days): 0 to 3
- Activity B (4 days): 3 to 7
- Activity E (6 days): 7 to 13
- Activity F (4 days): 13 to 17
- Activity G (3 days): 17 to 20

- Activity H (2 days): 20 to 22

Activities NOT on critical path:

- Activity C: 1 day slack (can be delayed by 1 day)
- Activity D: 1 day slack (can be delayed by 1 day)

Question 3.4 [0,5 point]: Determine total project duration and explain significance.
Solution:
Total Project Duration: $\boxed{\text{22 days}}$
Significance of the Critical Path:

1. **Minimum project completion time:** The critical path determines the shortest possible time to complete the project (22 days). The project cannot be completed faster unless activities on the critical path are accelerated.
2. **Resource focus:** Any delay in critical path activities directly delays the entire project. Management should prioritize resources and monitoring on these activities (A, B, E, F, G, H).
3. **Slack activities:** Activities with slack (C and D) have flexibility. They can be delayed without impacting the overall project deadline, allowing for resource trade-offs.
4. **Project control:** Identifying the critical path enables managers to focus on cost reduction and schedule compression where it matters most.

Exercise 4: Frank–Wolfe Algorithm [6 Points]

Problem Statement

$$\begin{aligned} \text{minimize} \quad & f(x_1, x_2) = (x_1 - 3)^2 + (x_2 - 2)^2 \\ \text{subject to:} \quad & x_1 + x_2 \leq 4 \\ & x_1 \geq 0, \quad x_2 \geq 0 \end{aligned}$$

Question 4.1 [0.75 points]: Show convexity and compute gradient.
Solution:
Convexity:
The objective function is $f(x_1, x_2) = (x_1 - 3)^2 + (x_2 - 2)^2$, which is a sum of two squared terms (convex functions). The sum of convex functions is convex. Alternatively, we can verify by computing the Hessian matrix:

$$\nabla^2 f = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

The Hessian is positive definite (eigenvalues are 2 and 2, both positive), so f is strictly convex.
Gradient Computation:

$$\nabla f(x_1, x_2) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 2(x_1 - 3) \\ 2(x_2 - 2) \end{pmatrix}$$

Question 4.2 [0.75 points]: Sketch feasible region and identify extreme points.
Solution:
Feasible Region:
The constraints define a triangular region:

- $x_1 + x_2 \leq 4$ (below the line $x_1 + x_2 = 4$)
- $x_1 \geq 0$ (right of the x_2-axis)
- $x_2 \geq 0$ (above the x_1-axis)

Extreme Points (Vertices):
The vertices of the feasible region are found by solving pairs of active constraints:

1. Intersection of $x_1 = 0$ and $x_2 = 0$: $(0, 0)$
2. Intersection of $x_1 = 0$ and $x_1 + x_2 = 4$: $(0, 4)$
3. Intersection of $x_2 = 0$ and $x_1 + x_2 = 4$: $(4, 0)$

$\boxed{\text{Extreme Points: } (0,0), (0,4), (4,0)}$

Question 4.3 [3.5 points]: Apply Frank–Wolfe algorithm for 2 iterations.
Solution:
Initial Point: $x^0 = (0, 0)$

Iteration 0 (k = 0):

Step 1: Compute gradient at x^0

$$\nabla f(0,0) = \begin{pmatrix} 2(0-3) \\ 2(0-2) \end{pmatrix} = \begin{pmatrix} -6 \\ -4 \end{pmatrix}$$

Step 2: Solve Linear Minimization Oracle (LMO)
Find $s^0 = \arg\min \langle \nabla f(x^0), s \rangle$ subject to $s \in$ Feasible Set

$$\min \langle (-6, -4), (s_1, s_2) \rangle = \min -6s_1 - 4s_2$$

subject to: $s_1 + s_2 \leq 4$, $s_1 \geq 0$, $s_2 \geq 0$
Since we want to minimize $-6s_1 - 4s_2$, we want to maximize $6s_1 + 4s_2$. This is maximized at a vertex of the feasible region:

- At $(0, 0)$: objective $= 0$
- At $(4, 0)$: objective $= 24$ (maximum)
- At $(0, 4)$: objective $= 16$

Therefore, $s^0 = (4, 0)$
Step 3: Compute step size

$$\gamma_0 = \frac{2}{0+2} = \frac{2}{2} = 1$$

Step 4: Update iterate

$$x^1 = x^0 + \gamma_0(s^0 - x^0) = (0,0) + 1 \cdot ((4,0) - (0,0)) = (4,0)$$

Iteration 1 (k = 1):

Step 1: Compute gradient at $x^1 = (4, 0)$

$$\nabla f(4, 0) = \begin{pmatrix} 2(4-3) \\ 2(0-2) \end{pmatrix} = \begin{pmatrix} 2 \\ -4 \end{pmatrix}$$

Step 2: Solve Linear Minimization Oracle

$$\min\langle(2, -4), (s_1, s_2)\rangle = \min 2s_1 - 4s_2$$

Evaluate at vertices:

- At $(0, 0)$: objective $= 0$
- At $(4, 0)$: objective $= 8$
- At $(0, 4)$: objective $= 0 - 16 = -16$ (minimum)

Therefore, $s^1 = (0, 4)$

Step 3: Compute step size

$$\gamma_1 = \frac{2}{1+2} = \frac{2}{3}$$

Step 4: Update iterate

$$x^2 = x^1 + \gamma_1(s^1 - x^1) = (4, 0) + \frac{2}{3}((0, 4) - (4, 0))$$

$$x^2 = (4, 0) + \frac{2}{3}(-4, 4) = (4, 0) + \left(-\frac{8}{3}, \frac{8}{3}\right) = \left(4 - \frac{8}{3}, \frac{8}{3}\right)$$

$$x^2 = \left(\frac{12-8}{3}, \frac{8}{3}\right) = \left(\frac{4}{3}, \frac{8}{3}\right)$$

Question 4.4 [0.5 points]: Report numerical values of x^1 and x^2.
Solution:

$$\boxed{x^1 = (4, 0)}$$

$$\boxed{x^2 = \left(\frac{4}{3}, \frac{8}{3}\right) \approx (1.333, 2.667)}$$

Objective function values:

- $f(x^0) = f(0, 0) = 9 + 4 = 13$
- $f(x^1) = f(4, 0) = 1 + 4 = 5$
- $f(x^2) = f(1.333, 2.667) = (1.333 - 3)^2 + (2.667 - 2)^2 = (-1.667)^2 + (0.667)^2 \approx 2.778 + 0.445 = 3.223$

Question 4.5 [0.5 points]: Comment on convergence toward unconstrained minimizer.
Solution:
Unconstrained Minimizer:
The unconstrained minimum occurs where $\nabla f = 0$:

$$\nabla f = (2(x_1 - 3), 2(x_2 - 2)) = (0, 0) \implies x^* = (3, 2)$$

However, $(3, 2)$ is **not feasible** because $3 + 2 = 5 > 4$.
Analysis of Frank–Wolfe Convergence:

1. **Monotonic improvement:** $f(x^0) = 13 \rightarrow f(x^1) = 5 \rightarrow f(x^2) \approx 3.223$ the sequence is consistently decreasing, indicating progress toward optimality.