Master of Science Examination, 2024

(Second Year, First Semester)

Mathematics

Paper : DSE - 03 (A7)

(Nonlinear and Dynamic Programming)

Full Marks : 40 *Time* : *Two Hours*

The figures in the margin indicate full marks.

Symbols / Notations have their usual meaning.

Use separate answer script for each group.

GROUP—A (20 Marks)

Answer ***any two*** questions

1. *(a)* Define convex set. Let $S \subseteq R^n$ be a convex set and $f : S \rightarrow R$. Then show that f is a convex function on S iff its epigraph E_f is a convex set.

(b) Using Steepest Descent method, minimize

$$f(x_1, x_2) = x_1^2 + x_2^2 + 2gx_1 + 2fx_2 + c$$

starting from the point $\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$. 5+5

2. *(a)* Define convex function. Let f be a convex function on a convex set $S \subseteq R^n$. Then show that every local minimum of f on S is the global minimum.

(b) Solve the following optimization problem using Kuhn-Tucker (KT) conditions :

Minimize $f = x_1^2 + x_2^2 - 2x_1$

subject to $x_1^2 + x_2 \leq 1$ 5+5

3. *(a)* Define conjugate directions. Find the conjugate directions for the matrix $\begin{bmatrix} 4 & 5 \\ 5 & 4 \end{bmatrix}$.

(b) Use Davidon-Fletcher Powell method to minimize $f(x_1, x_2) = 2x_1 + 3x_2 + 8x_1^2 + 12x_1x_2 + 9x_2^2$.

Take $\begin{bmatrix} 1/2 \\ 1/3 \end{bmatrix}$ as the starting point. 3+7

4. *(a)* Minimize $f(x_1, x_2) = x_1^2 + x_2^2$ subject to $1 - x_1 - x_2 \leq 0$, using Barrier function method with the calculus method of unconstrained minimization.

(b) Using Wolfe's method, solve the following quadratic programming problem (QPP) :

Maximize $z = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$

subject to $x_1 + 2x_2 \leq 2$

$x_1, x_2 \geq 0$ 4+6

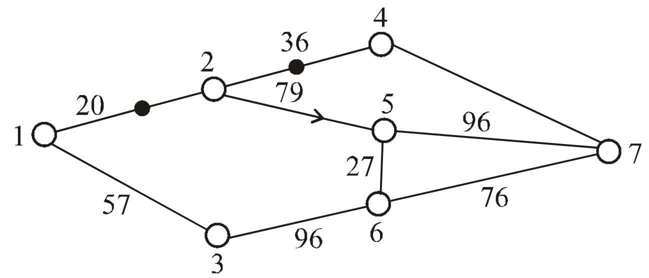
GROUP—B (20 Marks)

Attempt ***any two*** questions. Each question carries **10** marks.

1. Let X = "XMJAUZ" & Y = "MZJAWXU". Find the longest common subsequence of X & Y. 10

2. Define Catalan numbers. Find the first TEN Catalan numbers. State its time complexity. 10

3.



Find the shortest path from vertex 1 to vertex 7. Find the time complexity of the algorithm you used. 10

★ ★ ★