Dynamic Programming Applications

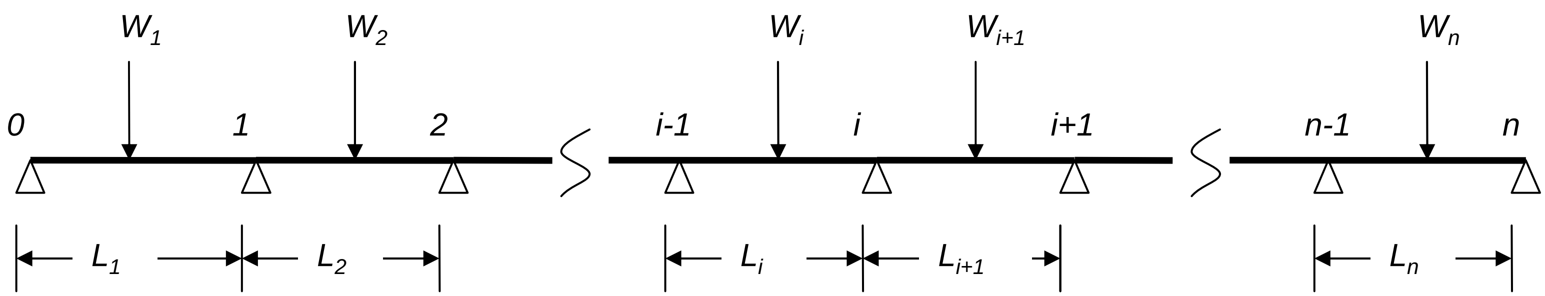
Design of Continuous Beam

Objectives

- To discuss the design of continuous beams
- To formulate the optimization problem as a dynamic programming model

Design of Continuous Beam

- Consider a continuous beam having n spans with a set of loadings W_1, W_2,…, W_n



- Beam rests on $n+1$ rigid supports
- Locations of the supports are assumed to be known

Design of Continuous Beam …contd.

- Objective function: To minimize the sum of the cost of construction of all spans
- Assumption: Simple plastic theory of beams is applicable
- Let the reactant support moments be represented as m_1, m_2, …, m_n
- Complete bending moment distribution can be determined once these support moments are known
- Plastic limit moment for each span and also the cross section of the span can be designed using these support moments

Design of Continuous Beam …contd.

- Bending moment at the center of the i^{th} span is $-W_iL_i/4$
- Thus, the largest bending moment in the i^{th} span can be computed as

$$M_i = \max\left\{ |m_{i-1}|, |m_i|, \left| \frac{m_{i-1} + m_i}{2} - \frac{W_i L_i}{4} \right| \right\} \qquad for \;\; i = 1, 2, \ldots n$$

- Limit moment for the i^{th} span, m_lim_i should be greater than or equal to M_i for a beam of uniform cross section
- Thus, the cross section of the beam should be selected such that it has the required limit moment

Design of Continuous Beam …contd.

- The cost of the beam depends on the cross section
- And cross section in turn depends on the limit moment
- Thus, cost of the beam can be expressed as a function of the limit moments
- *Let X* represents the vector of limit moments

$$X = \begin{Bmatrix} m\_\lim_1 \\ m\_\lim_2 \\ \vdots \\ m\_\lim_n \end{Bmatrix}$$

Design of Continuous Beam …contd.

- The sum of the cost of construction of all spans of the beam is

$$\sum_{i=1}^{n} C_i(X)$$

- Then, the optimization problem is to find X to

Minimize $\sum_{i=1}^{n} C_i(X)$

satisfying the constraints .

$$m\_\lim_i \geq M_i \qquad for \;\; i = 1, 2, ..., n$$

- This problem has a serial structure and can be solved using dynamic programming

Thank You