UF UNIVERSITY of FLORIDA

The Foundation for The Gator Nation

Linear Programming

When objective and constraints are linear,

the global optimum exists on a vertex

LINEAR PROGRAMMING (LP)

- Old name for linear optimization
  - Linear objective functions and constraints
- Optimum always at boundary of feasible domain
- First solution algorithm, Simplex algorithm developed by George Dantzig, 1947
  - What is a simplex (e.g. triangle, tetrahedron)?
- Applications
  - Resource allocation, transportation, product mix, scheduling, networking, etc.
- We will study limit design of skeletal structures as an application of LP.

Linear programming (LP) problem

$$\begin{aligned} &\min f(\mathbf{x}) \\ &\text{s.t. } g_i(\mathbf{x}) = 0, \qquad i = 1, \cdots, k \\ &\quad\;\; g_i(\mathbf{x}) \le 0, \qquad i = k+1, \cdots, m \end{aligned}$$

- Linear function $f = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n = \sum_{i=1}^{n} c_i x_i = \mathbf{c}^{\mathrm{T}}\mathbf{x}$
- Standard LP problem

$$\boxed{\begin{aligned} &\min f(\mathbf{x}) = \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ &\text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b} \\ &\mathbf{x} \ge 0, \mathbf{b} \ge 0 \\ &\mathbf{x} \in R^n, \mathbf{b} \in R^n, \mathbf{A} \in R^m \times R^n \end{aligned}}$$

- Inequality constraints

$\sum_{j=1}^{n} a_{ij} x_j \le b_i \;\; \rightarrow \sum_{j=1}^{n} a_{ij} x_j + s_i = b_i,$ $\qquad s_i \ge 0$: slack variable

$\sum_{j=1}^{n} a_{ij} x_j \ge b_i \;\; \rightarrow \sum_{j=1}^{n} a_{ij} x_j - s_i = b_i,$ $\qquad s_i \ge 0$: surplus variable

Linear programming (LP) problem *cont.*

- Free in sign
  - If x is free in sign, let $x = y - z,\ \text{where}\ \ y, z \geq 0$
- All linear problems are convex
  - Feasible set with linear constraints is convex, linear objective is convex
  - Any local minima is also a global minimum
- Optimum solutions are on the constraint boundary
  - Any solution of LP must satisfy $\mathbf{A}_{m\times n}\mathbf{x}_{n\times 1} = \mathbf{b}_{m\times 1}$
  - $m < n$ is the usual case, and assume $\mathrm{rank}(\mathbf{A}) = m$
  - When $\mathrm{rank}(\mathbf{A}) = m$, the system is called consistent

Canonical representation

- Decompose $\mathbf{A}$ into m dependent and $n-m$ independent

$$\mathbf{A} = [\underset{m\times m}{\mathbf{A}_1} \quad \underset{m\times(n-m)}{\mathbf{A}_2}], \quad \mathbf{x} = \begin{Bmatrix} \mathbf{x}_D \\ \mathbf{x}_I \end{Bmatrix}, \quad \Rightarrow [\mathbf{A}_1 \quad \mathbf{A}_2]\begin{Bmatrix} \mathbf{x}_D \\ \mathbf{x}_I \end{Bmatrix} = \mathbf{b}$$

$$\mathbf{A}_1\mathbf{x}_D + \mathbf{A}_2\mathbf{x}_I = \mathbf{b} \quad \Rightarrow \quad \mathbf{x}_D = \mathbf{A}_1^{-1}(\mathbf{b} - \mathbf{A}_2\mathbf{X}_I)$$

- Let $rank(\mathbf{A}) = r < m$, after Gauss elimination

$$\begin{bmatrix} \mathbf{I}_r & \mathbf{Q}_{r\times(n-r)} \\ \mathbf{0}_{(m-r)\times r} & \mathbf{0}_{(m-r)\times(n-r)} \end{bmatrix} \begin{Bmatrix} \mathbf{x}_r \\ \mathbf{x}_{(n-r)} \end{Bmatrix} = \begin{Bmatrix} \mathbf{q}_r \\ \mathbf{p}_{(n-r)} \end{Bmatrix}$$

  – The system is consistent if $\mathbf{p} = 0$ → the system has infinite solutions

  – If $r = m$, $\mathbf{I}\mathbf{x}_m + \mathbf{Q}_{m\times(n-m)}\mathbf{x}_{(n-m)} = \mathbf{q}_m$ Canonical representation

  – $\mathbf{x}_{(n-m)}$ is arbitrary and $\mathbf{x}_m = \mathbf{q}_m - \mathbf{Q}_{m\times(n-m)}\mathbf{x}_{(n-m)}$

Canonical representation *cont.*

- A particular solution is when $\mathbf{x}_{(n-m)} = 0$ → Basic solution
- Basic feasible solutions also satisfy $\mathbf{x} \geq 0$
  - $\mathbf{x}_m = \mathbf{q}$: Basic variable
  - $\mathbf{x}_{(n-m)} = \mathbf{0}$: non-basic variable
- No. of basic solutions for $\boldsymbol{A}\mathbf{x} = \mathbf{b}$ is finite

$$\binom{n}{m} = \frac{n!}{m!\,(n-m)!}$$

- Minimizing solution is one of basic feasible solution

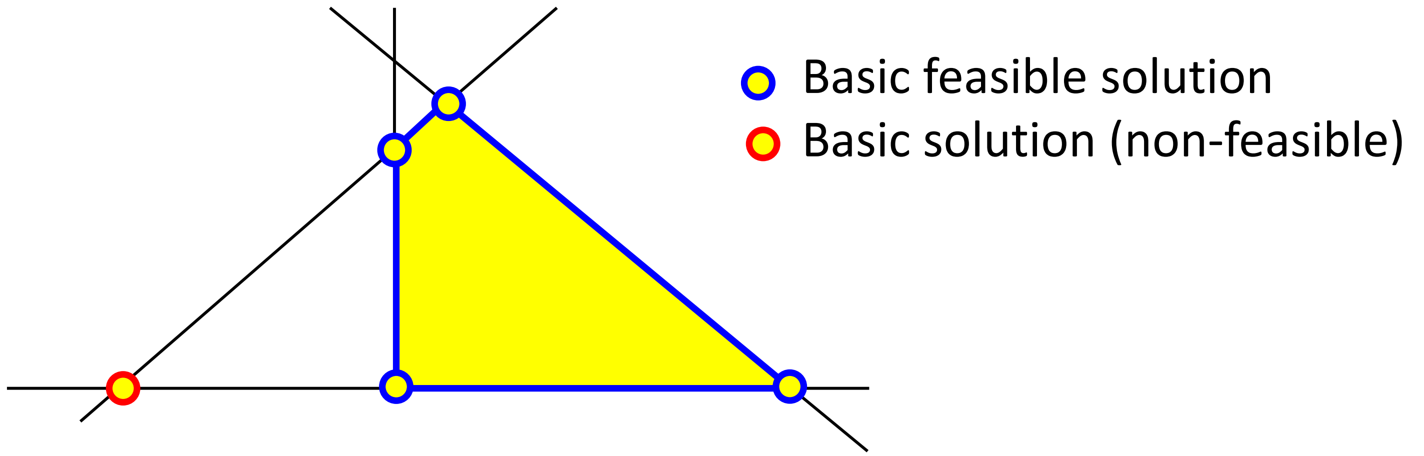
Canonical representation *cont*.

$$\underset{\uparrow}{\mathbf{I}\mathbf{x}_{(m)}} + \underset{\uparrow}{\mathbf{Q}\mathbf{x}_{(n-m)}} = \mathbf{q}_{(m)}$$

Dependent basic variable

Independent non-basic variable

- Basic feasible solutions: the corner points of convex feasible set of constraints
- Simplex algorithm: Starting from one corner point, move to next corner point to reduce the objective function



Example of basic feasible solution

$$\min f(\mathbf{x}) = -4x_1 - 5x_2$$
$$\text{s.t.}\ \ g_1(\mathbf{x}) = -x_1 + x_2 \le 4$$
$$g_2(\mathbf{x}) = x_1 + x_2 \le 6$$
$$x_1, x_2 \ge 0$$

$$\min f(\mathbf{x}) = -4x_1 - 5x_2$$
$$\text{s.t.}\ \ g_1(\mathbf{x}) = -x_1 + x_2 + x_3 = 4$$
$$g_2(\mathbf{x}) = x_1 + x_2 + x_4 = 6$$
$$x_1, x_2, x_3, x_4 \ge 0$$

– Basic solution of $Ax = b$

$$\mathbf{A} = \begin{bmatrix} -1 & 1 & \boxed{1} & \boxed{0} \\ 1 & 1 & \boxed{0} & \boxed{1} \end{bmatrix}$$

Basic	x_1	x_2	x_3	x_4	Value	
x_3	-1	1	1	0	4	Initial basic feasible solution
x_4	1	1	0	1	6	
x_3	-2	0	1	-1	-2	Infeasible
x_2	1	1	0	1	6	
x_1	1	0	-0.5	0.5	1	Feasible
x_2	0	1	0.5	0.5	5	
x_1	1	1	0	1	6	Feasible
x_3	0	2	1	1	10	

Sequential linear programming (SLP)

- Solving nonlinear problem using a sequence of LP

$$\min f(\mathbf{x})$$
$$\text{s.t. } g_i(\mathbf{x}) = 0, \qquad i = 1, \cdots, k$$
$$g_i(\mathbf{x}) \leq 0, \qquad i = k+1, \cdots, m$$

Nonlinear programming problem

1. Start with $x^{(0)}$

2. Linearize the problem using Taylor series expansion

$$f\left(\mathbf{x}^{(0)} + \Delta\mathbf{x}\right) \cong f\left(\mathbf{x}^{(0)}\right) + \nabla f\left(\mathbf{x}^{(0)}\right) \cdot \Delta\mathbf{x}$$

$$g\left(\mathbf{x}^{(0)} + \Delta\mathbf{x}\right) \cong g\left(\mathbf{x}^{(0)}\right) + \nabla g\left(\mathbf{x}^{(0)}\right) \cdot \Delta\mathbf{x}=0$$

Linear w.r.t. $\Delta\mathbf{x}$

3. Solve the LP problem w.r.t. Δx and update $\mathbf{x}^n$

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \Delta\mathbf{x}^{(n)}$$

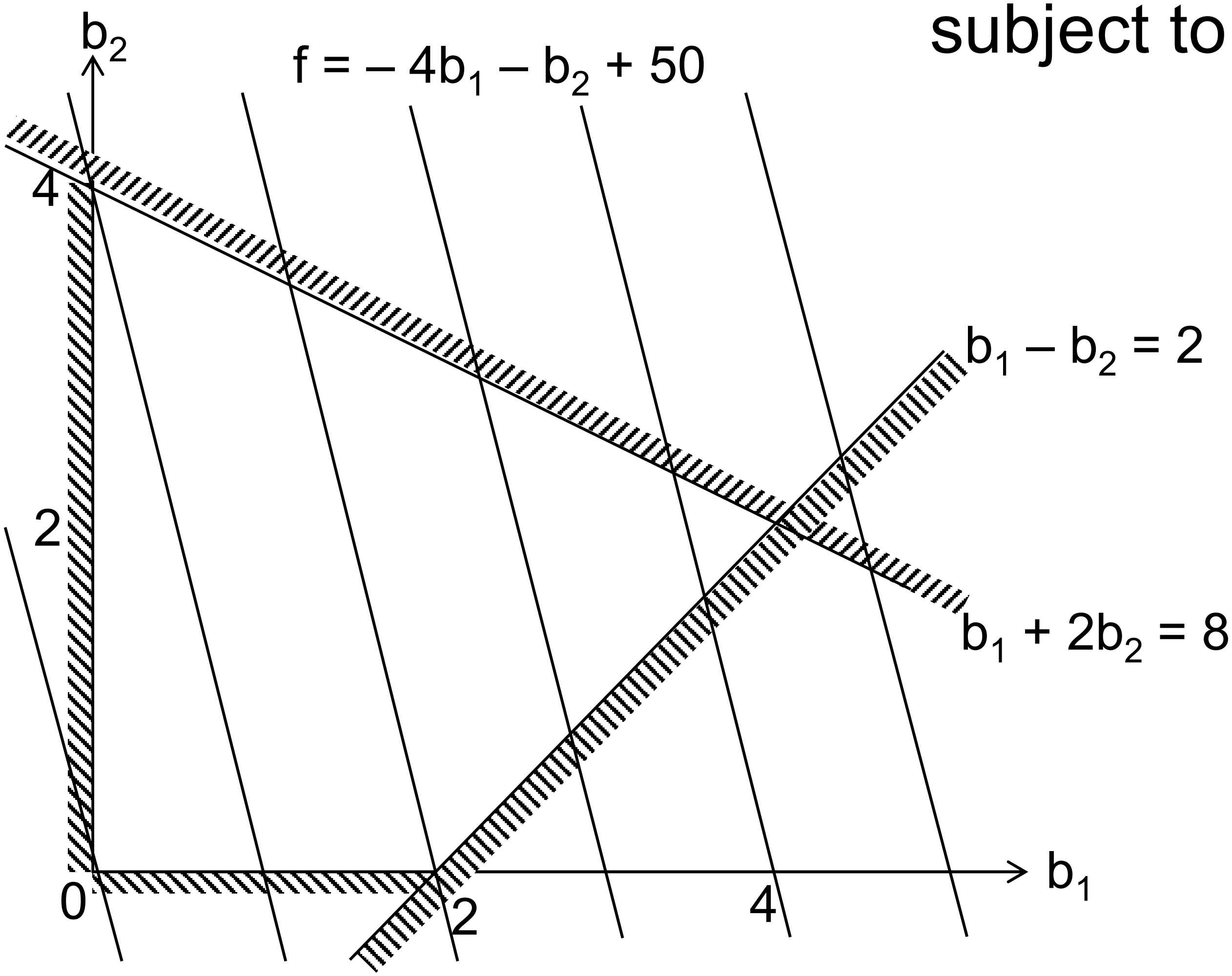
4. Introduce move limits

$$\left|\Delta\mathbf{x}^{(n)}\right| \leq \boldsymbol{\delta}$$

EXAMPLE: LP

- Linear optimization problem

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{b}) = -4b_1 - b_2 + 50 \\ \text{subject to} \quad & g_1(\mathbf{b}) = b_1 - b_2 \le 2 \\ & g_2(\mathbf{b}) = b_1 + 2b_2 \le 8 \\ & b_1 \ge 0, b_2 \ge 0 \end{aligned}$$



SOLUTION WITH MATLAB linprog

- Simplest form solves

$$\underset{\mathbf{b}}{\text{Minimize}} \quad \mathbf{f}^T\mathbf{b}$$

$$\text{subject to} \quad \mathbf{Ab} \le \mathbf{c}$$

```
f=[-4 -1];
A=[1 -1; 1 2; -1 0; 0 -1];
c=[2 8 0 0]';
[b,obj]=linprog(f,A,c)
Optimization terminated.
b =4.0000
   2.0000
obj =-18.0000
```

Matrix form

$$\text{Minimize} \quad f = -4b_1 - b_2 + 50$$

$$\text{subject to } b_1 - b_2 \le 2$$

$$b_1 + 2b_2 \le 8$$

$$b_1 \ge 0 \quad b_2 \ge 0$$

$$f = [-4 \; -1]$$

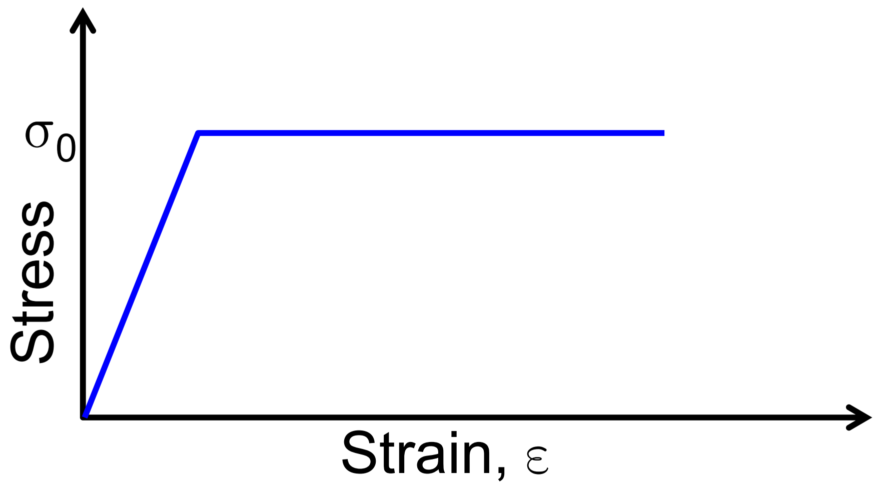
$$A = \begin{bmatrix} 1 & -1 \\ 1 & 2 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \quad c = \begin{bmatrix} 2 \\ 8 \\ 0 \\ 0 \end{bmatrix}$$

EXERCISE: linprog

- Solve the following problem using linprog and also graphically (do not use the equality constraint to reduce the number of variables)

$$
\begin{aligned}
\text{Maximize } & b_1 + 4b_2 \\
\text{subject to } & b_1 + 2b_2 \le 5 \\
& b_1 + b_2 = 4 \\
& b_1 - b_2 \ge 3 \\
& b_1, b_2 \ge 0
\end{aligned}
$$

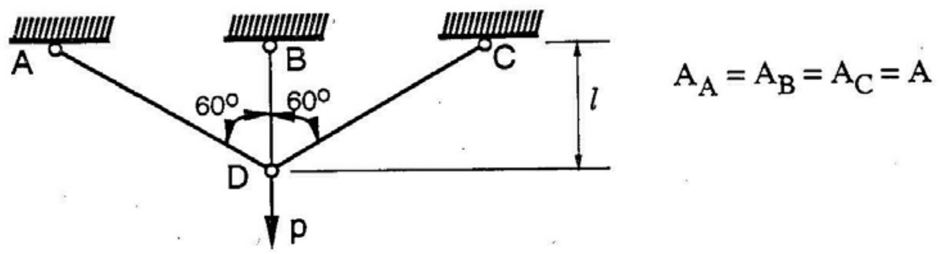
- Elastic-perfectly plastic behavior



Stress-strain curve for elasto-perfectly plastic material

- Normally, beyond yield the stress will continue to increase, so the assumption is conservative.
- We will see it will simplify estimating the collapse load of a truss.

THREE BAR TRUSS EXAMPLE



Elongations due to vertical displacement v at D:

$e_B = v, e_A = e_C = 0.5v$ (Recall $\cos 60^\circ = 0.5$)

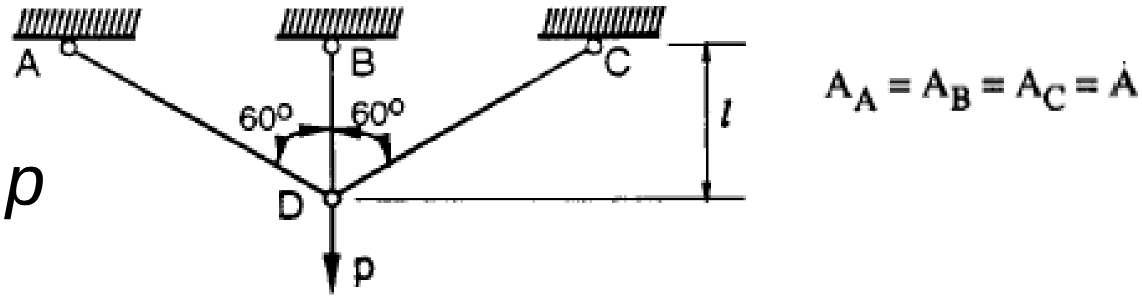
Strains: $\varepsilon_B = v / \ell$ $\varepsilon_A = \varepsilon_C = v / 4\ell$

Member forces: $n_B = \dfrac{EA}{\ell} v$ $n_A = n_C = \dfrac{EA}{4\ell} v = 0.25 n_B$

Equilibrium $n_A = n_C$ $p = n_B + 0.5(n_A + n_C) = 1.25 n_B$

Elastic solution: $n_A = n_C = 0.2p, n_B = 0.8p$

- Recall



$$n_A = n_C = 0.2p,\ n_B = 0.8p$$

- Member B yields first

$$n_B = 0.8p = \sigma_0 A, \quad p_{yield} = 1.25\sigma_0 A$$

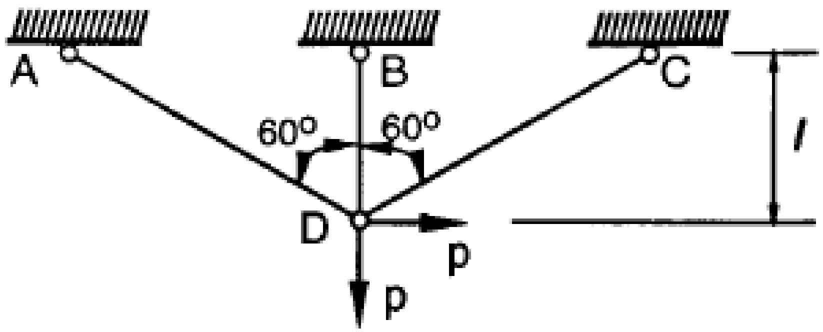
- However, load can be increased until members A and C also yield

$$n_A = n_B = n_C = \sigma_0 A$$
$$p = n_B + 0.5(n_A + n_C)$$
$$p_{collapse} = 2\sigma_0 A$$

LOWER BOUND THEOREM

- The Lower Bound Theorem: If a stress distribution can be found that is in equilibrium internally and balances the external loads, and also does not violate the yield conditions, these loads will be carried safely by the structure.
- Leads to an optimization problem with equations of equilibrium as equality constraints, and yield conditions as inequality constraints.

LP FORMULATION OF TRUSS COLLAPSE LOAD



$A_A = A_B = A_C = A$

$$n_B + 0.5(n_A + n_C) = p$$
$$0.5\sqrt{3}(n_A - n_C) = p$$

$$-A\sigma_0 \le n_A, n_B, n_C \le A\sigma_0$$

- Implication of lower bound theorem: Any *p* for which we can find *n*’s that satisfy the equation is safe
- LP problem: Find loads to maximize *p* subject to above constraints
- Non-dimensionalize! $N_A = \dfrac{n_A}{A\sigma_0}, \quad P = \dfrac{p}{A\sigma_0}$

NON-DIMENSIONAL FORM

- LP problem

Maximize P

subject to $N_B + 0.5(N_A + N_C) - P = 0$

$$0.5\sqrt{3}(N_A - N_C) - P = 0$$

$$-1 \le N_A, N_B, N_C \le 1$$

$$\mathbf{b} = \begin{Bmatrix} N_A \\ N_B \\ N_C \\ P \end{Bmatrix}$$

f=[0 0 0 -1]; A=eye(4); c=[1 1 1 1000]';

Aeq=[0.5 1 0.5 -1; sqrt(3)/2 0 -sqrt(3)/2 -1]; ceq=zeros(2,1);

lb=-[1 1 1 0]; b=linprog(f,A,c,Aeq,ceq,lb)’

Optimization terminated.

b =1.0000   1.0000   -0.4641   1.2679

EXERCISE: LIMIT DESIGN

- Limit design is to select truss cross sectional areas to minimize the weight of the truss subject to a given collapse load *p*. Formulate the limit design of the truss for given loads *p* as an LP and solve using linprog.
- Define a nominal area: $A_{nom} = p / \sigma_0$
- The non-dimensional design variables will now be the areas divided by *A* and the three member loads, divided by $p = A_{nom}\sigma_0$