Lecture 5:
Dual Simplex, Primal – Dual And Karmarkar's Algorithms

Prof. Krishna R. Pattipati
Dept. of Electrical and Computer Engineering
University of Connecticut
Contact: krishna@engr.uconn.edu; (860) 486-2890

Outline

- Review of duality
- Dual simplex algorithm
  - Revised simplex: primal feasibility $\xrightarrow{\text{work towards}}$ dual feasibility
  - Dual simplex: dual feasibility $\xrightarrow{\text{work towards}}$ primal feasibility
- Primal-dual algorithm
  - Enforce complementary slackness conditions over subsets of $\{1,2, \dots, n\}$
  - Widely used to solve network flow, assignment & transportation problems
- Interior point methods
  - The primal path following algorithm
  - Affine scaling methods (see notes. Will not be covered)
  - The potential reduction algorithm
  - The primal-dual path following algorithm
  - Implementation issues
- Comparison of revised simplex and Interior point methods
- Summary

Review of uality

- Duality
  - SLP and its dual

$$\begin{aligned} &\min \underline{c}^T \underline{x} \\ &s.t.\ A\underline{x} = \underline{b} \\ &\underline{x} \geq \underline{0} \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} &\max \underline{\lambda}^T \underline{b} \\ &s.t.\ \underline{\lambda} \text{ unrestricted} \\ &\underline{\lambda}^T A \leq \underline{c}^T \end{aligned}$$

    - Asymmetric form of the dual
  - Inequality constrained LP and its dual

$$\begin{aligned} &\min \underline{c}^T \underline{x} \\ &s.t.\ A\underline{x} \geq \underline{b} \\ &\underline{x} \geq \underline{0} \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} &\max \underline{\lambda}^T \underline{b} \\ &s.t.\ \underline{\lambda} \geq \underline{0} \\ &\underline{\lambda}^T A \leq \underline{c}^T \end{aligned}$$

    - Symmetric form of the dual
  - For all feasible $\underline{x}$ in primal and $\underline{\lambda}$ in dual
    - $\underline{\lambda}^T \underline{b} \leq \underline{c}^T \underline{x}$ $\Rightarrow$ dual feasible solution is always a lower bound on the primal
    - Dual unbounded $\Rightarrow$ primal infeasibility
    - Primal unbounded $\Rightarrow$ dual infeasibility
    - Primal infeasibility may imply dual infeasibility and vice-versa
    - When dual and primal have finite optimal solution, max of the dual, $\underline{\lambda}^T \underline{b}$ = min of the primal, $\underline{c}^T \underline{x}^*$

Complementary Slackness & Sensitivity

- Complementary slackness conditions

$$(\underline{c}^T - \underline{\lambda}^{*T} A)\underline{x}^* = 0 \Rightarrow x_i^* > 0 \Rightarrow c_i = \underline{\lambda}^T \underline{a}_i$$

(or relative cost = 0 or x_i^* in basis)

$$\Rightarrow x_i^* = 0 \Rightarrow c_i > \underline{\lambda}^T \underline{a}_i$$

(or relative cost > 0 or x_i^* is nonbasic)

$\Rightarrow$ **true cost > synthetic cost**

- For inequality constrained problem

$$(\underline{\lambda}^*)^T (A\underline{x} - \underline{b}) = 0$$

$$\Rightarrow \lambda_i^* > 0 \Rightarrow \quad \underline{a}_i^T \underline{x} = b_i \quad \text{(nonbasic surplus)}$$

$$\lambda_i^* = 0 \Rightarrow \quad \underline{a}_i^T \underline{x} > b_i \quad \text{(basic surplus)}$$

- Simplex multipliers λ_j are the costs of $\underline{e}_j$, the jth unit vector
- Cost of any other vector $\underline{a}_k$ is $\sum \lambda_j a_{jk} = \underline{\lambda}^T \underline{a}_{k\ldots}$ synthetic cost of vector $\underline{a}_k$
  - $\lambda_j^* = \dfrac{\partial f}{\partial b_j}; \; x_j^* = \dfrac{\partial f}{\partial c_j}$

Dual Simplex Algorithm

- In the shortest path problem, λ_j can be interpreted as the length of the shortest path from source to node j
  - If $\lambda_j - \lambda_i = c_{ij}$, edge (i, j) is in the shortest path
  - If $\lambda_j - \lambda_i < c_{ij}$, edge (i, j) is not in the shortest path
  - $\underline{\lambda}^*$ and $\underline{x}^*$ are saddle points of

$$L(\underline{x}, \underline{\lambda}) = \underline{c}^T \underline{x} - \underline{\lambda}^T A\underline{x} + \underline{\lambda}^T \underline{b}$$

$$\Rightarrow \min_{\underline{x}\geq 0} \max_{\underline{\lambda}} L(\underline{x}, \underline{\lambda}) = \max_{\underline{\lambda}} \min_{\underline{x}\geq 0} L(\underline{x}, \underline{\lambda})$$

- **Dual Simplex Algorithm**
  - Primal revised simplex starts with a primal feasible solution $\underline{x}$ s.t. $A\underline{x} = \underline{b}, \underline{x} > \underline{0}$ and work towards $(\underline{c}^T - \underline{\lambda}^T A) = \underline{p}^T \geq \underline{0} \Rightarrow$ dual feasibility

$$A\underline{x} = \underline{b}, \quad \underline{x} \geq \underline{0} \quad \xrightarrow{\text{update } \underline{x}} \quad \underline{c}^T - \underline{\lambda}^T A \geq \underline{0}, \quad \underline{\lambda} = \underline{c}_B^T B^{-1}$$

  - Note
    - Basic $\Rightarrow$ equality
    - Non-basic $\Rightarrow$ strict inequality

From Dual Feasibility to Primal Feasibility

- What if we tried another approach?

$$\begin{array}{ccc} \text{From Dual Feasibility} & \rightarrow & \text{Primal Feasibility} \\ \underline{c}^T - \underline{\lambda}^T A \geq \underline{0} & \xrightarrow{\text{update } \lambda} & \underline{x}_B = B^{-1}\underline{b},\ \underline{x}_B \geq \underline{0} \end{array}$$

- The latter approach leads to the *Dual Simplex Algorithm*
- Key ideas:
  - Suppose $\underline{\lambda}$ is dual feasible

    $$\Rightarrow \underline{\lambda}^T A \leq \underline{c}^T \text{ or } \underline{\lambda}^T \underline{a}_j \leq c_j \forall j$$

  - Suppose our basis B consists of the first m columns

    $$(\underline{a}_1,\ \underline{a}_2,\ \ldots,\ \underline{a}_m)$$

  - From revised simplex and complementary slackness conditions, we know

    $$\underline{\lambda}^T \underline{a}_j = c_j;\ 1 \leq j \leq m \quad \Rightarrow \underline{\lambda}^T = \underline{c}_B^T B^{-1}$$

    $$\underline{\lambda}^T \underline{a}_j < c_j;\ m+1 \leq j \leq n \quad \text{(barring degeneracy)}$$

- What is the corresponding $\underline{x}_B = B^{-1}\underline{b}$ (is it primal feasible?)

Need not be Primal Feasible!!

- Suppose $x_{Bl} < 0$, we must remove the corresponding column $\underline{a}_l$ from the basis
  - $x_{Bl} = \left[\text{row } l \text{ of } \left(B^{-1}\right)\right] * \underline{b}$

Dual Step Size Selection

- Since want to maximize the dual, what if I perturb $\underline{\lambda} \to \underline{\hat{\lambda}}$ s. t.

  $$\underline{\hat{\lambda}}^T \underline{b} = \underline{\lambda}^T \underline{b} - \varepsilon x_{Bl} > \underline{\lambda}^T \underline{b},\ \varepsilon > 0, = (\underline{\lambda}^T - \varepsilon \text{ row } l(B^{-1}))\underline{b}$$

- So, $\underline{\hat{\lambda}}^T = \underline{\lambda}^T - \varepsilon \text{ row } l(B^{-1}) = (\underline{c}_B^T - \varepsilon \underline{e}_l^T)B^{-1}$

- *Q: How far to go?*
- *A: Only so far as to maintain dual feasibility*

$$(\underline{c}^T - \underline{\hat{\lambda}}^T A) \geq \underline{0}^T$$

$$\underline{\hat{\lambda}}^T \underline{a}_j = c_j,\ j \neq l,\ j = 1, \ldots, m$$

$$\underline{\hat{\lambda}}^T \underline{a}_l = c_l - \varepsilon < c_l \quad \text{(out of the basis)}$$

$$\underline{\hat{\lambda}}^T \underline{a}_j = \underline{\lambda}^T \underline{a}_j - \varepsilon \underline{e}_l^T B^{-1} \underline{a}_j,\quad j = m+1, \ldots, n$$

$$= z_j - \varepsilon \alpha_{lj},\quad j = m+1, \ldots, n \quad \text{where } z_j < c_j$$

- *What does this mean:* $\underline{\hat{\lambda}}^T \underline{a}_l < c_l \Rightarrow$ strict inequality or column $\underline{a}_l$ left the basis
- *Q: Which column should we bring into the basis?*
- *A: The one that makes* $z_j - \varepsilon \alpha_{lj} = c_j$ *first*
- What if all $\alpha_{lj} \geq 0$?

  $\Rightarrow$ Can never make $c_j = z_j - \varepsilon \alpha_{lj}$ since $z_j < c_j$

  $\Rightarrow$ Dual unbounded, since $\underline{\hat{\lambda}}$ is feasible $\forall \varepsilon$

Dual Simplex Algorithm Steps

- If any $\alpha_{lj} < 0$, can move until $\varepsilon_j = \frac{z_j - c_j}{\alpha_{lj}} = \frac{-p_j}{\alpha_{lj}}$

$\Rightarrow$Among these ε, pick one that reaches c_j first $\varepsilon = \frac{z_k - c_k}{\alpha_{lk}} = \frac{-p_k}{\alpha_{lk}} = \min_j \left\{ \frac{z_j - c_j}{\alpha_{lj}} : \alpha_{lj} < 0 \right\}$

- Update basis $B = B - \text{column } \underline{a}_l + \text{column } \underline{a}_k$ as in revised simplex and compute $\underline{x}_B = B^{-1}\underline{b}$

- **Dual simplex algorithm steps**:

<u>Step 1</u>: Given a dual feasible solution $\underline{x}_B = B^{-1}\underline{b}$

if $\underline{x}_B \geq \underline{0}$ then the solution is optimal

else select an index l such that $x_{Bl} < 0$

<u>Step 2</u>: If all $\alpha_{lj} = \left[\text{row } l \text{ of } (B^{-1})\right] * \underline{a}_j \geq 0$ for all non-basic columns $\underline{a}_j$,

then unbounded dual (or infeasible primal)

else $\varepsilon = \min_j \left\{ \frac{z_j - c_j}{\alpha_{lj}} = \frac{-p_j}{\alpha_{lj}} : \alpha_{lj} < 0 \right\} = \frac{z_k - c_k}{\alpha_{lk}} = \frac{-p_k}{\alpha_{lk}}$

<u>Step 3</u>: Update $\underline{\lambda}$, basis B, and $\underline{x}_B$

$$\underline{\lambda}^T \leftarrow \underline{\lambda}^T - \varepsilon \text{ row } l(B^{-1})$$

$$B \leftarrow B - \text{column } \underline{a}_l + \text{column } \underline{a}_k \text{ (or propogate } B^{-1} \text{ or } LU \text{ or } QR \text{ factors)}$$

Go back to Step 1

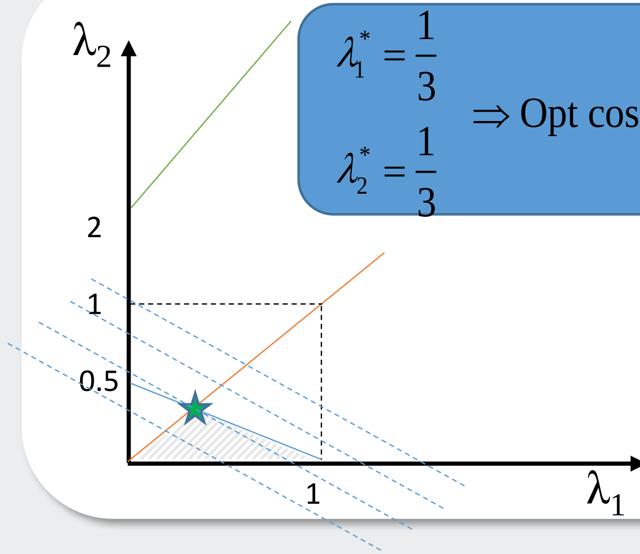
Optimality ⇒ Dual Feasibility & Primal Feasibility

- Why does it converge?
  - Maintain dual feasibility at each stage
  - Choice of $x_{Bl} < 0 \Rightarrow$ dual objective increases
  - Cannot terminate at a non-optimum point (because all we require for optimum is dual and primal feasibility)
  - Finite number of extreme points ⇒ must terminate in a finite number of steps
- Example: Primal

$$\min x_1 + 2x_2$$
$$s.t.\ x_1 - 2x_2 + x_3 \geq 4$$
$$2x_1 + x_2 - x_3 \geq 6$$
$$x_1, x_2, x_3, \geq 0$$

Dual

$$\max 4\lambda_1 + 6\lambda_2$$
$$s.t.\ \lambda_1 + 2\lambda_2 \leq 1$$
$$-2\lambda_1 + \lambda_2 \leq 2$$
$$\lambda_1 - \lambda_2 \leq 0$$
$$\lambda_1, \lambda_2 \geq 0$$



$$\lambda_1^* = \frac{1}{3} \quad \lambda_2^* = \frac{1}{3} \Rightarrow \text{Opt cost} = \frac{10}{3}$$

- Graphical Solution:

$$B = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \qquad B^{-1} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{2}{3} & -\frac{1}{3} \end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{2}{3} & -\frac{1}{3} \end{bmatrix} \begin{bmatrix} 4 \\ 6 \end{bmatrix} = \begin{bmatrix} \frac{10}{3} \\ \frac{2}{3} \end{bmatrix} \qquad x_1 = \frac{10}{3},\ x_2 = 0,\ x_3 = \frac{2}{3}$$

$$\underline{\lambda}^T = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{2}{3} & -\frac{1}{3} \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \end{bmatrix} \qquad \text{opt cost} = \frac{10}{3}$$

Illustration of Dual Simplex Algorithm

- Example:

Primal

$$\min 3x_1 + 4x_2 + 5x_3$$
$$s.t.\;\; x_1 + 2x_2 + 3x_3 \geq 5$$
$$2x_1 + 2x_2 + x_3 \geq 6$$
$$x_i \geq 0$$

Dual

$$\max 5\lambda_1 + 6\lambda_2$$
$$s.t.\;\; \lambda_1 + 2\lambda_2 \leq 3$$
$$2\lambda_1 + 2\lambda_2 \leq 4$$
$$3\lambda_1 + \lambda_2 \leq 5$$
$$\lambda_1, \lambda_2 \geq 0$$

Optimal Solution:

$$\lambda_1 = 1,\; \lambda_2 = 1 \Rightarrow x_1 = 1,\; x_2 = 2,\; x_3 = 0$$
$$\text{optimal cost} = 11$$

Iteration 0:

(1):

$$\lambda_1 = \lambda_2 = 0 \Rightarrow z_j = 0\; \forall j$$
$$x_1 + 2x_2 + 3x_3 - s_1 = 5$$
$$2x_1 + 2x_2 + x_3 - s_2 = 6$$
$$\Rightarrow B = -I \text{ is the basis}$$
$$\underline{x}_B = -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 5 \\ 6 \end{bmatrix} = \begin{bmatrix} -5 \\ -6 \end{bmatrix}$$

Select the most negative one : s_2

(2):

$$p_1 = c_1 - z_1 = 3;\; p_2 = c_2 - z_2 = 4;\; p_3 = c_3 - z_3 = 5$$
$$(\text{row } l \text{ of } B^{-1})\underline{a}_j = -\begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 & 3 \\ 2 & 2 & 1 \end{bmatrix} = \begin{bmatrix} -2 & -2 & -1 \end{bmatrix}$$
$$\varepsilon = \min_j \left\{ \frac{z_j - c_j}{\alpha_{lj}} : \alpha_{lj} < 0 \right\} = \min \begin{bmatrix} \frac{3}{2} & \frac{4}{2} & \frac{5}{1} \end{bmatrix} = \frac{3}{2}$$

Dual Simplex Algorithm Steps

(3) Update $\underline{\lambda}$, B and $\underline{x}_B$

$$\Rightarrow \text{column 1 comes into the basis} \Rightarrow \text{basis} \begin{pmatrix} s_1 \\ x_1 \end{pmatrix}$$

$$\text{or } \underline{\lambda}^T = \underline{\lambda}^T - \varepsilon(\text{row}_l \text{ of } (B^{-1})) = \begin{bmatrix} 0 & 0 \end{bmatrix} - \tfrac{3}{2}\begin{bmatrix} 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & \tfrac{3}{2} \end{bmatrix}$$

$$\text{new } B = \begin{bmatrix} -1 & 1 \\ 0 & 2 \end{bmatrix} \qquad \text{new } B^{-1} = \begin{bmatrix} -1 & \tfrac{1}{2} \\ 0 & \tfrac{1}{2} \end{bmatrix}$$

$$\underline{\lambda}^T = \begin{bmatrix} 0 & 3 \end{bmatrix} \begin{bmatrix} -1 & \tfrac{1}{2} \\ 0 & \tfrac{1}{2} \end{bmatrix} = \begin{bmatrix} 0 & \tfrac{3}{2} \end{bmatrix}$$

$$\underline{x}_B = \begin{bmatrix} -1 & \tfrac{1}{2} \\ 0 & \tfrac{1}{2} \end{bmatrix} \begin{bmatrix} 5 \\ 6 \end{bmatrix} = \begin{bmatrix} -2 \\ 3 \end{bmatrix} \Rightarrow \begin{pmatrix} s_1 \\ x_1 \end{pmatrix}$$

<u>Iteration 1</u>:

(1) s_1 goes out of basis

(2) $$(\text{row}_1 \text{ of } B^{-1}) \underbrace{\begin{bmatrix} 2 & 3 & 0 \\ 2 & 1 & -1 \end{bmatrix}}_{N} = \begin{bmatrix} -1 & \tfrac{1}{2} \end{bmatrix} \begin{bmatrix} 2 & 3 & 0 \\ 2 & 1 & -1 \end{bmatrix} = \begin{bmatrix} -1 & -\tfrac{5}{2} & -\tfrac{1}{2} \end{bmatrix}$$

Dual Simplex Algorithm

$$(\text{row}_1 \text{ of } B^{-1})N = \begin{bmatrix} -1 & -\frac{5}{2} & -\frac{1}{2} \end{bmatrix}$$

$$\underline{\lambda}^T A - \underline{c} = \begin{bmatrix} 0 & \frac{3}{2} \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 & -1 & 0 \\ 2 & 2 & 1 & 0 & -1 \end{bmatrix} - \begin{bmatrix} 3 & 4 & 5 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} \overset{\downarrow}{0} & -1 & -\frac{7}{2} & \overset{\downarrow}{0} & -\frac{3}{2} \end{bmatrix} = -\underline{p}$$

$$\varepsilon = \min\begin{bmatrix} \frac{1}{1} & \frac{7}{5} & 3 \end{bmatrix} \Rightarrow \text{column 2 enters the basis}$$

(3) $\underline{\lambda}^T = \underline{\lambda}^T - \varepsilon(\text{row}_1 \text{ of } B^{-1}) = \begin{bmatrix} 0 & \frac{3}{2} \end{bmatrix} - 1\begin{bmatrix} -1 & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 1 & 1 \end{bmatrix}$

$$\text{new } B = \begin{bmatrix} 2 & 1 \\ 2 & 2 \end{bmatrix} \qquad \text{new } B^{-1} = \begin{bmatrix} 1 & -\frac{1}{2} \\ -1 & 1 \end{bmatrix}$$

$$\text{Old } B^{-1} = \begin{bmatrix} -1 & \frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix}$$

$$\text{check}: \quad c_B^T B^{-1} = \begin{bmatrix} 4 & 3 \end{bmatrix} \begin{bmatrix} 1 & -\frac{1}{2} \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

$$\underline{x}_B = B^{-1}\underline{b} = \begin{bmatrix} 1 & -\frac{1}{2} \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{pmatrix} x_2 \\ x_1 \end{pmatrix}$$

$x_1 = 1, x_2 = 2, x_3 = 0$ Done!!!

Another Example of Dual Simplex Algorithm

- Example:

Primal:

x_1 = number of barrels of light crude

x_2 = number of barrels of heavy crude

$$\min\ 56x_1 + 50x_2$$

$$s.t.\ 0.3x_1 + 0.3x_2 \ge 900{,}000$$
$$0.2x_1 + 0.4x_2 \ge 800{,}000$$
$$0.3x_1 + 0.2x_2 \ge 500{,}000$$
$$x_1 \ge 0; x_2 \ge 0$$

optimal point: $(0, 3M)$; Cost: \$150M

Dual:

$$\max\ 100{,}000[9\lambda_1 + 8\lambda_2 + 5\lambda_3]$$

$$s.t.\ 0.3\lambda_1 + 0.2\lambda_2 + 0.3\lambda_3 \le 56$$
$$0.3\lambda_1 + 0.4\lambda_2 + 0.2\lambda_3 \le 50$$
$$s.t.\ \lambda_1 \ge 0; \lambda_2 \ge 0; \lambda_3 \ge 0$$

optimal point: $(500/3 \;\; 0 \;\; 0)$

Cost: \$150M

Iteration 0:

(1):

$$\lambda_1 = \lambda_2 = \lambda_3 = 0 \Rightarrow z_j = 0\ \forall j$$
$$0.3x_1 + 0.3x_2 - s_1 = 900{,}000$$
$$0.2x_1 + 0.4x_2 - s_2 = 800{,}000$$
$$0.3x_1 + 0.2x_2 - s_3 = 500{,}000$$
$$x_1 \ge 0; x_2 \ge 0; s_i \ge 0$$

$\Rightarrow B = -I$ is the basis

$$\underline{x}_B = -\begin{bmatrix} -900{,}000 \\ -800{,}000 \\ -500{,}000 \end{bmatrix}$$

Select the most negative one : s_1

(2):

$$p_1 = c_1 - z_1 = 56;\ p_2 = c_2 - z_2 = 50$$

(row l of B^{-1})$N = [-0.3 - 0.3]$

$$\varepsilon = \min_j \left\{ \frac{z_j - c_j}{\alpha_{lj}} : \alpha_{lj} < 0 \right\} = \min\left[\frac{56}{0.3} \quad \frac{50}{0.3} \right] = \frac{500}{3}$$

Dual Simplex Algorithm Steps

(3):

$$\Rightarrow \text{column 2 comes into the basis} \Rightarrow \text{basis} \begin{bmatrix} x_2 \\ s_2 \\ s_3 \end{bmatrix}$$

$$\text{or } \underline{\lambda}^T = \underline{\lambda}^T - \varepsilon(\text{row}_l \text{ of } (B^{-1})) = [0 \quad 0 \quad 0] - \frac{500}{3}[-1 \quad 0 \quad 0] = \begin{bmatrix} \frac{500}{3} & 0 & 0 \end{bmatrix}$$

$$\text{new } B = \begin{bmatrix} 0.3 & 0 & 0 \\ 0.4 & -1 & 0 \\ 0.2 & 0 & -1 \end{bmatrix} \quad \text{new } B^{-1} = \begin{bmatrix} 10/3 & 0 & 0 \\ 4/3 & -1 & 0 \\ 2/3 & 0 & -1 \end{bmatrix}$$

$$\underline{\lambda}^T = [50 \quad 0 \quad 0] \begin{bmatrix} 10/3 & 0 & 0 \\ 4/3 & -1 & 0 \\ 2/3 & 0 & -1 \end{bmatrix} = [500/3 \quad 0 \quad 0]$$

$$\underline{x}_B = \begin{bmatrix} 10/3 & 0 & 0 \\ 4/3 & -1 & 0 \\ 2/3 & 0 & -1 \end{bmatrix} \begin{bmatrix} 900{,}000 \\ 800{,}000 \\ 500{,}000 \end{bmatrix} = \begin{bmatrix} 3{,}000{,}000 \\ 400{,}000 \\ 100{,}000 \end{bmatrix} \Rightarrow \begin{bmatrix} x_2 \\ s_2 \\ s_3 \end{bmatrix}$$

$$\Rightarrow \textit{Optimal} \Rightarrow f^* = \$150M$$

Key Idea of Primal-Dual Algorithm

- Idea for Primal-Dual Algorithm
  - To set the stage, consider the SLP and its dual

Primal		*Dual*
$\min \underline{c}^T \underline{x}$ $s.t.\ A\underline{x} = \underline{b}$ $\underline{x} \geq \underline{0}$	$\Leftrightarrow$	$\max \underline{\lambda}^T \underline{b}$ $s.t.\ \underline{\lambda}$ unrestricted $\underline{\lambda}^T A \leq \underline{c}^T$

  - At optimum:

    $\underline{\lambda}^T (A\underline{x} - \underline{b}) = 0\ldots$ satisfied for any feasible $\underline{x}$ in primal and

    $(\underline{c}^T - \underline{\lambda}^T A)\underline{x} = 0\ldots$ satisfied at optimum
  - Suppose we have a feasible $\underline{\lambda}$ for the dual problem

    $\Rightarrow \underline{\lambda}^T A \leq \underline{c}^T$

    $\Rightarrow$ Some of these inequalities will be equalities

    $\Rightarrow$ Define the subset P of $\{1,\ldots,n\}$ by $i \in P$

    $$P = \left\{ i : \underline{\lambda}^T \underline{a}_i = c_i \right\}$$

    If none, set $P = \varnothing$
  - For optimality, we need:

    $x_i > 0$ if $\underline{\lambda}^T \underline{a}_i = c_i \Rightarrow i \in P$

    $x_i = 0$ if $\underline{\lambda}^T \underline{a}_i < c_i \Rightarrow i \notin P$ $\Rightarrow$ so, if we can find x_i s.t. $x_i = 0$ for $i \notin P$, we are done!!

Maintaining Dual and Primal Feasibility

- What does it mean?
  - This amounts to searching for $\underline{x}$ such that

$$\sum_{i\in P} \underline{a}_i x_i = \underline{b} \qquad x_i \geq 0, i \in P\,; \quad x_i = 0, i \notin P$$

$\Rightarrow$ Nonnegative linear combinations of columns in P=$\underline{b}$

P = set of admissible columns

- But, this is simply phase I of LP … restricted primal (RP)

$$\min_{\underline{x},\underline{y}} \sum_{i=1}^{m} y_i = \underline{e}^T \underline{y} = \begin{bmatrix} \underline{0}^T & \underline{e}^T \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{y} \end{bmatrix} = \underline{c}^T \underline{x}_a$$

$$s.t. \quad \sum_{i\in P} \underline{a}_i x_i + \underline{y} = \underline{b}$$

$$x_i \geq 0, i \in P\,; x_i = 0, \ i \notin P \text{ (implicit)};\ \underline{y} \geq \underline{0}$$

- Dual of the restricted primal (DRP)

$$\max_{\underline{\mu}} \ \underline{\mu}^T \underline{b}$$

$$s.t. \quad \underline{\mu}^T \underline{a}_i \leq 0; i \in P$$

$$\underline{\mu} \leq \underline{e}$$

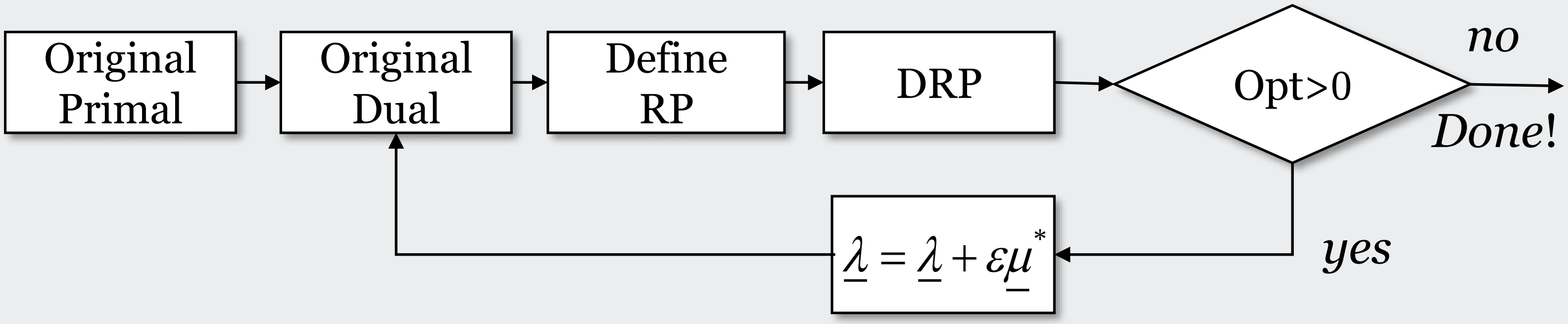
- Given a feasible $\underline{\lambda}$, we can find a feasible solution $\underline{x}$ to the associated RP
- If optimum solution of RP = 0, then found an optimum:

  $\underline{x}$ from RP & original $\underline{\lambda}$ are optimum

- Else, update $\underline{\lambda}$ via $\underline{\lambda} = \underline{\lambda} + \varepsilon \underline{\mu}^*$ where $\underline{\mu}^*$ = vector of simplex multipliers at the termination of RP

Primal-Dual Algorithm Graphically

- Graphically, the idea is this:



- Key questions
  - What is the sign of ε?
  - What is the largest ε I can take? … must maintain dual feasibility
  - Can I detect infeasibility?
  - Does the algorithm converge?
- Sign of ε
  - $\underline{\mu}^{*T}\underline{b} \geq 0$ since $\underline{\mu} = \underline{0}$ is feasible for DRP
  - New dual cost:

$$\underline{\lambda}^T\underline{b} = \underline{\lambda}^T\underline{b} + \varepsilon\underline{\mu}^{*T}\underline{b} = \underline{\lambda}^T\underline{b} + \varepsilon(\text{optimum solution of RP (or DRP)}) > \underline{\lambda}^T\underline{b} \text{ if } \varepsilon > 0$$

  - Must take $\varepsilon > 0$ to increase the cost of original dual

Step Size in Primal-Dual Algorithm

- Step size and detection of infeasibility
  - What is the effect of ε on feasibility?

    Need $\underline{\lambda}^{T}\underline{a}_i = \underline{\lambda}^T\underline{a}_i + \varepsilon\underline{\mu}^{*T}\underline{a}_i \le c_i \;\; \forall i = 1,\ldots,n$

    If $\underline{\mu}^{*T}\underline{a}_i < 0 \Rightarrow$ *No Problem*

    However, if $\underline{\mu}^{*T}\underline{a}_i < 0 \;\forall i$ then

    $\Rightarrow$ we can increase ε indefinitely, while maintaining dual feasibility

    $\Rightarrow$ dual is unbounded $\Rightarrow$ primal is infeasible
  - If optimal solution in RP>0 and the optimal dual satisfies $\underline{\mu}^{*T}\underline{a}_i < 0 \;\forall i \notin P$, then the original problem is infeasible (or original dual is unbounded)
  - If original problem has finite optimum
    - At least some $\underline{\mu}^{*T}\underline{a}_i > 0$ for $i \notin P$
    - ε should be chosen such that the equality is met by one of the constraints first

$$\varepsilon = \min_{i \notin P}\left\{\frac{c_i - \underline{\lambda}^T\underline{a}_i}{\underline{\mu}^{*T}\underline{a}_i} : \underline{\mu}^{*T}\underline{a}_i > 0\right\}$$

  - The dual cost increases to $\underline{\lambda}^{T}\underline{b} = \underline{\lambda}^T\underline{b} + \varepsilon\underline{\mu}^{*T}\underline{b}$
  - The set P changes to $P \leftarrow P \cup \{k\}$ where $k = \arg\min_{i \notin P}\left\{\frac{c_i - \underline{\lambda}^T\underline{a}_i}{\underline{\mu}^{*T}\underline{a}_i} : \underline{\mu}^{*T}\underline{a}_i > 0\right\}$

Primal-Dual Algorithm Steps

- **Primal-Dual Algorithm**

  Step 1:

  Given a feasible $\underline{\lambda}$ to the dual problem

  $$\max \underline{\lambda}^T \underline{b}$$
  $$s.t. \quad \underline{\lambda}^T A \le \underline{c}^T$$

  Determine the restricted primal problem:

  - Find set P
  - Formulate restricted primal: $\min \underline{e}^T \underline{y}$

    $$\text{s.t.} \quad \sum_{i \in P} \underline{a}_i x_i + \underline{y} = \underline{b}$$
    $$x_i \ge 0, i \in P; \quad x_i = 0, i \notin P \text{ (implicit)}; \ \underline{y} \ge \underline{0}$$

  - Note: $\underline{b} \ge \underline{0}$, if not, multiply corresponding Eq. by –1

  Step 2:

  Optimize the restricted primal (phase I of LP)

  If optimal solution = 0, then done

  Else go to Step 3

  Step 3:

  Compute $\underline{\mu}^{*T} \underline{a}_i$ for $i \notin P$

Illustration of Primal-Dual Algorithm

Step 3 (cont'd): If all $\underline{\mu}^{*T}\underline{a}_i < 0$ for $i \notin P$, then primal is infeasible

Else update $\underline{\lambda} \leftarrow \underline{\lambda} + \varepsilon\underline{\mu}^*$

Where $$\varepsilon = \frac{c_k - \underline{\lambda}^T\underline{a}_k}{\underline{\mu}^{*T}\underline{a}_k} = \min_{i \notin P}\left\{\frac{c_i - \underline{\lambda}^T\underline{a}_i}{\underline{\mu}^{*T}\underline{a}_i} : \underline{\mu}^{*T}\underline{a}_i > 0\right\}$$

$$P \leftarrow P \cup \{k\}$$

Go back to Step 1

Primal-Dual:

$$\begin{aligned} &\min 3x_1 + 4x_2 + 5x_3 \\ &s.t.\ x_1 + 2x_2 + 3x_3 \geq 5 \\ &\quad 2x_1 + 2x_2 + x_3 \geq 6 \\ &\quad x_i \geq 0 \end{aligned}$$

$$\begin{aligned} &\max 5\lambda_1 + 6\lambda_2 \\ &s.t.\ \lambda_1 + 2\lambda_2 \leq 3 \\ &\quad 2\lambda_1 + 2\lambda_2 \leq 4 \\ &\quad 3\lambda_1 + \lambda_2 \leq 5 \\ &\quad \lambda_1, \lambda_2 \geq 0 \end{aligned}$$

Iteration 0:

Let $\underline{\lambda} = 0,\ \{c_i - \underline{\lambda}^T\underline{a}_i\} = [3\ \ 4\ \ 5] \Rightarrow P = \phi$

Restricted primal:

$RP : \min \underline{e}^T\underline{y}$ s.t. $\underline{y} = \underline{b}$; $\underline{y} \geq \underline{0}$

$DRP : \max \underline{\mu}^T\underline{b}$ s.t. $\underline{\mu} \leq \underline{e}$ $\Rightarrow$ $\underline{y} = \underline{b},\ \underline{\mu}^T = \underline{e}^T$

$\underline{\mu}^T\{\underline{a}_i\} = [3\ \ 4\ \ 4]$

$\varepsilon = \min\left[\frac{3}{3}\ \ \frac{4}{4}\ \ \frac{5}{4}\right]$ $\Rightarrow$ Both 1 & 2 can enter basis

$P = \{1,2\};\ \underline{\lambda}^T = \underline{\lambda}^T + \varepsilon\underline{\mu}^T = [0\ \ 0] + 1[1\ \ 1] = [1\ \ 1]$

Property of Primal-Dual Algorithm

Iteration 1:

RP:

$$\min \underline{e}^T \underline{y}$$
$$\text{s.t.} \begin{bmatrix}1\\2\end{bmatrix} x_1 + \begin{bmatrix}2\\2\end{bmatrix} x_2 + \underline{y} = \underline{b} = \begin{bmatrix}5\\6\end{bmatrix}$$

DRP:

$$\max 5\mu_1 + 6\mu_2$$
$$\text{s.t.} \quad \mu_1 + 2\mu_2 \le 0$$
$$2\mu_1 + 2\mu_2 \le 0 \qquad \Rightarrow \mu_1 = \mu_2 = 0$$
$$\mu_1 \le 1$$
$$\mu_2 \le 1$$
$$\Rightarrow \underline{\lambda}^T = [1 \quad 1]; \text{optimal basis}, B = \begin{bmatrix}1 & 2\\2 & 2\end{bmatrix}; B^{-1} = \begin{bmatrix}-1 & 1\\1 & \frac{-1}{2}\end{bmatrix}$$
$$\underline{x}_B = B^{-1}\underline{b} = [1 \quad 2]^T \qquad \Rightarrow \underline{x}^T = [1 \quad 2 \quad 0]$$

- **Property of primal-dual algorithm**
  - Every column $i \in P$ in the optimal basis of restricted primal (RP) remains in set *P* at the start of next iteration
  - Proof:
    - If a column *i* is in the optimal basis of RP, $(\underline{\mu}^*)^T \underline{a}_i = 0$

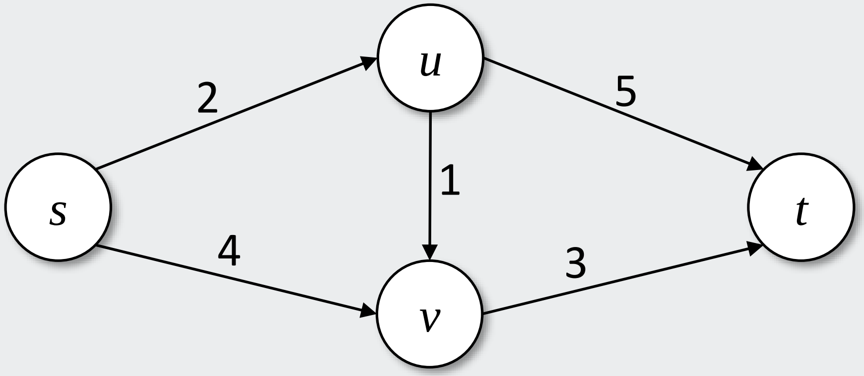
$$\Rightarrow \underline{\lambda}^T \underline{a}_i = \underline{\lambda}^T \underline{a}_i + \varepsilon \underline{\mu}^{*T} \underline{a}_i = \underline{\lambda}^T \underline{a}_i = c_i, \text{since } i \in P$$

- The algorithm must converge
  - No primal basis is repeated

Primal-Dual Algorithm for Shortest Path Problem

  - Pivoting on $\underline{a}_k$ will decrease restricted primal cost (since $(\underline{\mu}^*)^T \underline{a}_k > 0$)
  - There are only a finite number of bases

- Application to shortest path problem… Dijkstra's algorithm



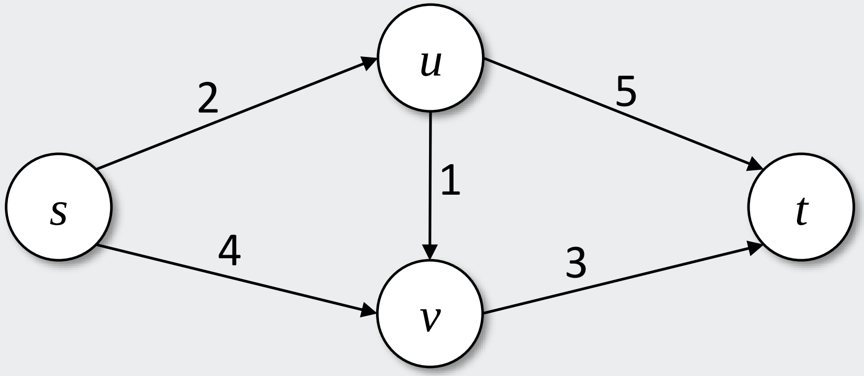
  - *s*, *u*, *v*, *t* are computers, edge lengths are costs of sending a message between them
  - Let x_{sv} be the fraction of messages sent from *s* to *v*

- **Primal**

$$\min \quad 2x_{su} + 4x_{sv} + x_{uv} + 5x_{ut} + 3x_{vt}$$
$$\text{s.t.} \quad x_{su}, x_{sv}, x_{uv}, x_{ut}, x_{vt} = 0 \text{ or } 1$$

Primal-Dual Algorithm for Shortest Path Problem

$$A\underline{x} = \begin{bmatrix} 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_{su} \\ x_{sv} \\ x_{uv} \\ x_{ut} \\ x_{vt} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \underline{b}$$



- **Dual**
  - λ_s = Price of a message at node *s* (buying or selling) = 0
  - λ_t = Price of a message at node *t* (buying or selling)

$$\begin{aligned} \max\ & \lambda_t \\ \text{s.t.}\ & \lambda_u \le 2 \\ & \lambda_v \le 4 \\ & \lambda_v - \lambda_u \le 1 \\ & \lambda_t - \lambda_u \le 5 \\ & \lambda_t - \lambda_v \le 3 \end{aligned}$$

- **Crude way**
  - Start with $\underline{\lambda}^T = [0\ \ 0\ \ 0]; P = \phi$

Primal-Dual Algorithm for Shortest Path Problem

$\Rightarrow$ RP has solution $\underline{y} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

$\Rightarrow$ Optimal cost=1

$$\text{Basis} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow \underline{\mu}^* = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

because $\max \mu_t$ s.t. $\mu_u \le 1,\ \mu_v \le 1,\ \mu_t \le 1$

<u>Iteration 1</u>: $(\underline{\mu}^*)^T \underline{a}_i = [1 \quad 1 \quad 0 \quad 0 \quad 0]$ for $i \notin P$

$$\varepsilon = \arg\min_{i \notin P} \left\{ \frac{c_i - \underline{\lambda}^T \underline{a}_i}{\underline{\mu}^T \underline{a}_i} : \underline{\mu}^T \underline{a}_i > 0 \right\} = \min[2 \quad 4 \quad x \quad x \quad x]$$

$\Rightarrow$ pick column 1 to enter admissible column set $P \Rightarrow P\{1\}$

- Update $\underline{\lambda} \Rightarrow \underline{\lambda}^T = [0 \quad 0 \quad 0] + 2[1 \quad 1 \quad 1] = [2 \quad 2 \quad 2]$
- x_{su}=1
- Dual of RP $\max \mu_t$

$$\begin{aligned} s.t.\ \ \mu_u &\le 0 \\ \mu_v &\le 1 \\ \mu_t &\le 1 \end{aligned} \qquad \Rightarrow \underline{\mu}^* = [0 \quad 1 \quad 1]$$

Primal-Dual Algorithm for Shortest Path Problem

Iteration 2:

$$\overline{P} = \{2 \quad 3 \quad 4 \quad 5\}$$

$$(\underline{\mu}^*)^T \underline{a}_i = [1 \quad 1 \quad 1 \quad 0] \quad \text{for } i \notin P$$

$$\varepsilon = \min\left\{\frac{4-2}{1}, \frac{1}{1}, \frac{5-2}{1}\right\} = 1 \Rightarrow P = \{1,3\}$$

$$\Rightarrow \underline{\lambda}^T = [2 \quad 2 \quad 2] + 1[0 \quad 1 \quad 1] = [2 \quad 3 \quad 3]$$

$$\Rightarrow x_{uv} = 1$$

Iteration 3:

$$\overline{P} = \{2 \quad 4 \quad 5\}$$

$$\begin{aligned} &\max \mu_t \\ &\text{s.t. } \mu_u \le 0 \\ &\quad \mu_v - \mu_u \le 0 \Rightarrow \underline{\mu}^* = [0 \quad 0 \quad 1] \\ &\quad \mu_v, \mu_t \le 1 \end{aligned}$$

$$(\underline{\mu}^*)^T \underline{a}_i = [0 \quad 0 \quad 1]; i \notin P$$

$$\Rightarrow \varepsilon = \min\left\{\frac{3-0}{1}\right\} = 3$$

$$\Rightarrow \underline{\lambda}^T = [2 \quad 3 \quad 3] + 3[0 \quad 0 \quad 1] = [2 \quad 3 \quad 6]$$

$$\Rightarrow x_{vt} = 1$$

Iteration 4:

$$\begin{aligned} &\max \mu_t \\ &\text{s.t. } \mu_u \le 0 \\ &\quad \mu_v - \mu_u \le 0 \Rightarrow \underline{\mu}^* = 0 \Rightarrow \text{optimal} \\ &\quad \mu_t - \mu_v \le 0 \end{aligned}$$

There is a method to our madness

- Shortest path from $s - t$: $s \rightarrow u \rightarrow v \rightarrow t$
  - $s \rightarrow u = 2 = \lambda_u$
  - $s \rightarrow v = 3 = \lambda_v$
  - $s \rightarrow t = 6 = \lambda_t$
- There is a method to our madness .... Related to Dijkstra's Algorithm
  - $\underline{\mu}^*$ at stage i, where j columns (or arcs) are in the admissible set is defined as follows:

    $\underline{\mu}^* = 0$ for all nodes reachable by paths from source s using arcs in P

    $\underline{\mu}^* = 1$ for all other nodes
  - Iteration 1: Since P is empty $\underline{\mu}^*=[1\ 1\ 1]$
  - Iteration 2: Since P includes column 1 (arc(s, u)), $\underline{\mu}^*=[0\ 1\ 1]\ldots$
  - Iteration 3: Since P includes columns 1 and 3 (arcs (s,u), (u,v)), $\underline{\mu}^*=(0\ 0\ 1)$
  - Iteration 4: Since P includes columns 1,3 and 5 (arcs (s,u), (u,v) and (v,t)), $\underline{\mu}^*=(0\ 0\ 0)$
- What about step size ε?

$$\varepsilon = \min_{\text{arcs} \notin P}\{\text{cost of arc} - (\lambda_{\text{end node of arc}} - \lambda_{\text{start node of arc}})\}$$

  - Note: Denominator $(\underline{\mu}^*)^{\mathrm{T}} \underline{a}_i$ is always 1 or 0. Recall unimodularity of A
  - So consider arcs with $\mu^*_{\text{end node of arc}} - \mu^*_{\text{start node of arc}} > 0$ (in this case 1)

Relation to Dijkstra's Algorithm

- Since $\underline{\mu}^* = 0$ for all nodes reachable by s using arcs in P, λ_i for these nodes remains fixed from the time node i enters the feasible set P until the conclusion of the algorithm
  - Note the evolution of $\underline{\lambda}$

    $[0\ 0\ 0] \rightarrow [2\ 2\ 2] \rightarrow [2\ 3\ 3] \rightarrow [2\ 3\ 6]$
- If we let w be the set of nodes reachable through arcs in P, λ_i for these nodes remains constant till the end of the algorithm
- At each iteration, one node is added to w until w becomes the entire set of nodes $s \rightarrow (s, u) \rightarrow (s, u, v) \rightarrow (s, u, v, t)$
- Looks like we terminate in $(n - 1)$ steps where n is the number of nodes… with some streamlining, this is DIJKSTRA's algorithm…Lecture 6
- λ_u, λ_v and λ_t are the lengths of the shortest paths from start node s
- Interior Point Algorithms
- Three major types
  - The primal and primal-dual path following algorithms
  - Affine scaling algorithms
  - Potential Reduction Algorithms

Interior Point Methods

- Path following algorithms
  - Discuss not the original Interior point algorithm, but an equivalent (and more general) formulation based on **Barrier functions**

SLP

$$\min_{\underline{x}} \underline{c}^T \underline{x}$$
$$s.t.\ A\underline{x} = \underline{b}$$
$$\underline{x} \geq \underline{0}$$
optimal solution $\underline{x}^*$

$\Leftrightarrow$

Barrier

$$\min_{\underline{x}} f(\underline{x}, \mu) = \underline{c}^T \underline{x} - \mu \sum_{j=1}^{n} \ln x_j$$
$$s.t.\ A\underline{x} = \underline{b}$$
$$\mu > 0$$
optimal solution $\underline{x}^*(\mu)$

  - Key: $\underline{x}^*(\mu) \rightarrow \underline{x}^*$ as the Barrier parameter $\mu \rightarrow 0$
  - $\exists$ many variations of Barrier function formulations… we will discuss them later or see references

- Consider the general NLP

$$\min_{\underline{x}} f(\underline{x})$$
$$s.t.\ A\,\underline{x} = \underline{b}$$

  - Suppose $\underline{x}$ is feasible, then $\bar{\underline{x}} = \underline{x} + \alpha\,\underline{d}$, $\underline{d}$ ~ search direction
  - Pick α s.t. $A\bar{\underline{x}} = \underline{b}$ (new point is feasible) and $f(\underline{x}) < f(\bar{\underline{x}})$

Newton's Method for NLP

- What does Newton's method do for this problem?
  - Feasibility $\Rightarrow A\underline{\bar{x}} = A\underline{x} + \alpha A\underline{d} = 0 \Rightarrow A\underline{d} = 0$
  - Newton's method fits a quadratic to $f(\underline{x})$ at the current point and takes $\alpha = 1$

$$f(\underline{x}+\underline{d}) = f(\underline{x}) + \underline{g}^T\underline{d} + \frac{1}{2}\underline{d}^T H \underline{d}, \quad \text{where} \quad \underline{g} = \nabla f(\underline{x}); H = \nabla^2 f(\underline{x})$$

  - **Newton's method solves a quadratic problem to find $\underline{d}$** (⇒ a weighted least squares problem)
  - Consider

$$\begin{aligned}&\min_{\underline{d}}\ \underline{g}^T\underline{d} + \tfrac{1}{2}\underline{d}^T H \underline{d}\\ &s.t.\ A\underline{d} = \underline{0}\end{aligned} \quad \Rightarrow \quad \begin{aligned}&\min_{\underline{d}}\ \tfrac{1}{2}\left\| H^{\frac{1}{2}}\underline{d} - H^{\frac{1}{2}}\underline{g}\right\|_2^2\\ &s.t.\quad A\underline{d} = \underline{0}\\ &\qquad H^{\frac{1}{2}} \text{ symmetric square root}\end{aligned}$$

  - Define Lagrangian function:

$$L(\underline{d},\underline{\lambda}) = \underline{g}^T\underline{d} + \tfrac{1}{2}\underline{d}^T H \underline{d} - \underline{\lambda}^T A\underline{d}; \quad \underline{\lambda} \sim \text{ Lagrange multiplier}$$

  - Karush-Kuhn-Tucker necessary conditions of optimality:

$$\Rightarrow \frac{\partial L}{\partial \underline{d}} = 0 \Rightarrow \underline{g} + H\underline{d} - A^T\underline{\lambda} = \underline{0}$$

$$\Rightarrow \frac{\partial L}{\partial \underline{\lambda}} = 0 \Rightarrow -A\underline{d} = \underline{0}$$

KKT Conditions for the Barrier Problem

- Special NLP = Barrier formulation of LP:

$$\underline{g} = \nabla f(\underline{x}) = \underline{c} - \mu D^{-1}\underline{e} \text{ and } H = \nabla^2 f(\underline{x}) = \mu D^{-2}$$

where

$$D = \text{Diag}(x_j);\ j = 1, 2, \ldots, n$$

$$\underline{e} = [1 \quad 1 \quad 1 \quad \cdots \quad 1]^T$$

- Karush-Kuhn-Tucker conditions for special NLP are:

$$\mu D^{-2}\underline{d} + (\underline{c} - \mu D^{-1}\underline{e} - A^T\underline{\lambda}) = \underline{0}$$

$$A\underline{d} = \underline{0}$$

- So,

$$\underline{d} = \frac{-1}{\mu} D^2(\underline{c} - \mu D^{-1}\underline{e} - A^T\underline{\lambda}) \qquad (1)$$

- Using $A\underline{d} = \underline{0}$ in (1), we get

$$\underline{\lambda} = (AD^2A^T)^{-1}AD^2(\underline{c} - \mu D^{-1}\underline{e}) \qquad (2)$$

$$\text{or } \underline{\lambda} = (AD^2A^T)^{-1}A(D^2\underline{c} - \mu D\underline{e}) \qquad (3)$$

$$\underline{d} = [I - D^2A^T(AD^2A^T)^{-1}A](D\underline{e} - \frac{1}{\mu}D^2\underline{c}) \qquad (4)$$

Path Following Algorithm

- So, $\underline{\lambda}$ is the solution of weighted least square (WLS) problem:

$$\min_{\underline{\lambda}} \left\| D[\underline{c} - \mu D^{-1}\underline{e} - A^T \underline{\lambda}] \right\|_2^2$$

- Barrier function (Path following) Algorithm:
  - Choose a strictly feasible solution and constant $\mu > 0$
  - Let the tolerance parameter be ε and a parameter associated with the update of μ be σ

for $k = 0,1,\ldots.,k_{\max}$

let $D = \text{Diag}(x_j)$

Compute the solution $\underline{\lambda}$ to $(AD^2A^T)\underline{\lambda} = AD^2(\underline{c} - \mu D^{-1}\underline{e})$...WLS solution

let $\underline{p} = \underline{c} - A^T\underline{\lambda}$

$$\underline{d} = \frac{-D^2(\underline{p} - \mu D^{-1}\underline{e})}{\mu} = -\frac{(D^2\underline{p} - \mu D\underline{e})}{\mu}$$

$\underline{x} = \underline{x} + \underline{d}$

if $\underline{x}^T\underline{p} < \varepsilon \rightarrow$ stop : $\underline{x}$ is near-optimal solution... complementary slackness condition

else $\mu \leftarrow (1 - \frac{\sigma}{\sqrt{n}})\mu$

end if

end

$k_{max} \approx 50$
$\sigma \approx 1/4 - 1/6$

Finding a Feasible Point

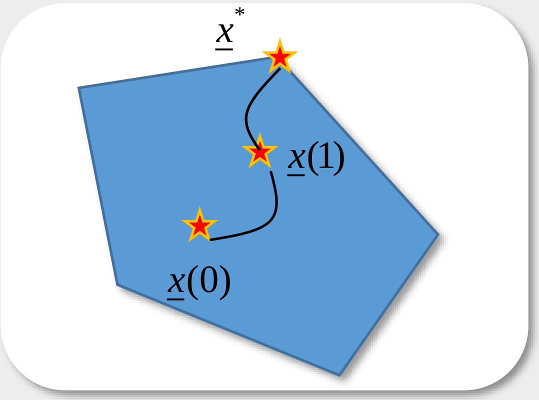


Illustration of Path Following Algorithms

- **Remarks**:
- Finding a feasible point

  Method 1

  - Select any $\underline{x}_0 > \underline{0}$ and define $\xi_0 \underline{y} = \underline{b} - A\underline{x}_0$ with $\|\underline{y}\|_2 = 1$

    $\Rightarrow \xi_0 = \|\underline{b} - A\underline{x}_0\|_2$ and solve:

$$
\begin{aligned}
&\min_{\underline{x},\xi} \xi \\
&s.t. \begin{bmatrix} A & \underline{y} \end{bmatrix} \begin{bmatrix} \underline{x} \\ \xi \end{bmatrix} = \underline{b} \\
&\quad \underline{x} \geq \underline{0} \\
&\quad \xi \geq 0
\end{aligned}
$$

$$
\begin{aligned}
&Initial: \underline{x}_0 = \| \underline{b} \|_2 \, \underline{e} \\
&\xi = \| \underline{b} - A\underline{x}_0 \|_2 \\
&\underline{y} = \frac{\underline{b} - A\underline{x}_0}{\| \underline{b} - A\underline{x}_0 \|_2}
\end{aligned}
$$

Finding Feasible Point using M Method - 1

- The solution: $\xi = 0$ or when ξ starts becoming negative → stop
- Suggest $\underline{x}_0 = \|\underline{b}\|\underline{e}$

Method 2: … big M method

Primal	**Dual**
$\min \underline{c}^T \underline{x}$	$\max \underline{\lambda}^T \underline{b}$
s.t. $A\underline{x} = \underline{b}$	s.t. $\underline{\lambda}$ unrestricted
$\underline{x} \geq \underline{0}$	$\underline{\lambda}^T A \leq \underline{c}^T \Rightarrow \underline{\lambda}^T A + \underline{p}^T = \underline{c}^T$

- Assume A, $\underline{b}$ and $\underline{c}$ are integers with absolute values bounded by U (Can always do this by scaling numbers by 10^t, $t \sim 3-6$)
- Then,

$$\sum_{j=1}^{n} x_j = \underline{e}^T \underline{x} \leq n(mU)^m \text{ (very loose bound)}$$

- Let $\bar{\underline{b}} = \underline{b}(n+2)/n(mU)^m;\ x_i \leftarrow x_i(n+2)/n(mU)^m$

Finding Feasible Point using M Method - 2

- Finding a feasible point - Method 2 (cont'd...)

Primal

$$\begin{aligned}
&\min\ \underline{c}^T x + M x_{n+1} \\
&\text{s.t. } A\underline{x} + (\overline{b} - A\underline{e}) x_{n+1} = \overline{b} \\
&\underline{e}^T \underline{x} + x_{n+1} + x_{n+2} = n+2 \\
&\underline{x} \geq \underline{0} \\
&x_{n+1} \geq 0;\ x_{n+2} \geq 0
\end{aligned}$$

Dual

$$\begin{aligned}
&\max\ \underline{\lambda}^T \overline{b} + \lambda_{m+1}(n+2) \\
&\text{s.t. } \underline{\lambda}^T A + \lambda_{m+1}\underline{e}^T + \underline{p}^T = \underline{c}^T \\
&\underline{\lambda}^T \left(\overline{b} - A\underline{e}\right) + \lambda_{m+1} + p_{n+1} = M \\
&\lambda_{m+1} + p_{n+2} = 0 \\
&p_1, p_2, \ldots, p_{n+1}, p_{n+2} \geq 0
\end{aligned}$$

- If we let $\mu_0 = 4\sqrt{\|\underline{c}\|^2 + M^2}$

  $\left(\underline{x} \quad x_{n+1} \quad x_{n+2}\right)_0 = \left(\underline{e} \quad 1 \quad 1\right)$ and

  $\left(\underline{\lambda} \quad \lambda_{m+1} \quad \underline{p} \quad p_{n+1} \quad p_{n+2}\right) = \left(\underline{0} \quad -\mu_0 \quad \underline{c} + \mu_0\underline{e} \quad M + \mu_0 \quad \mu_0\right)$ are feasible solutions

- Since the method uses Newton's directions, expect quadratic convergence near minimum

Major Computational Step: WLS

- Major computational step: Weighted Least-squares subproblem

$$(AD^2A^T)\underline{\lambda} = AD^2(\underline{c} - \mu D^{-1}\underline{e})$$

  - Generally A is sparse
  - We will discuss the computational aspects of Least-squares subproblem later

- The algorithm (theoretically) requires $O(\sqrt{n}L)$ iterations with overall complexity $O(n^3L)$ where

$$L = \sum_{i=0}^{m}\sum_{j=1}^{n}\left[\log\left|a_{ij}\right| + 1\right] + 1$$

- In practice, the method typically takes 20 – 50 iterations even for very large problems (> 20,000 variables). Simplex, on the other hand, takes increasingly large numbers of iterations with the problem size n
- Initialize $\mu = 2^{O(L)}$ and $\sigma \approx \frac{1}{4}$ to $\frac{1}{6}$. In practice, we need to experiment with the parameters

Other Potential Functions

- Other potential functions:

$$f(\underline{x},\ q) = r\ln(\underline{c}^T\underline{x} - q) - \sum_j \ln x_j$$

where $r = n + \sqrt{n}$ and

q = a lower-bound on the optimal cost

- Problem with Barrier function approach:
  - Update of μ
  - Selection of initial μ and parameter σ

• **Dual Affine scaling**:

- Typically, the affine scaling methods are used on the dual problem

Primal		*Dual*		*Modified Dual*
$\min_{\underline{x}}\ \underline{c}^T\underline{x}$		$\max_{\underline{\lambda}}\ \underline{\lambda}^T\underline{b}$		$\max_{\underline{\lambda}}\ \underline{\lambda}^T\underline{b}$
s.t. $A\underline{x} = \underline{b}$	$\Leftrightarrow$	s.t. $A^T\underline{\lambda} \le \underline{c}$	$\Leftrightarrow$	s.t. $A^T\underline{\lambda} + \underline{p} = \underline{c}$
$\underline{x} \ge \underline{0}$				$\underline{p} \ge \underline{0}$

Dual problem and scaled reduced costs

- Suppose we have a strictly feasible $\underline{\tilde{\lambda}}$ and the corresponding reduced cost vector (slack vector) is $\underline{\tilde{p}}$
  - Define

$$\underline{\hat{p}} = P^{-1}\underline{p}$$

where

$$P = \text{Diag}\left[\, \tilde{p}_1, \tilde{p}_2, \dots, \tilde{p}_n \right]$$

- So, the dual problem is:

$$\begin{aligned} &\max \underline{\lambda}^T \underline{b} \\ &\text{s.t.} \quad A^T \underline{\lambda} + P\underline{\hat{p}} = \underline{c} \\ &\underline{\hat{p}} \geq \underline{0} \end{aligned}$$

- From the equality constraint:

$$\begin{aligned} &\underline{\hat{p}} = P^{-1}(\underline{c} - A^T \underline{\lambda}) \\ &\Rightarrow P^{-1} A^T \underline{\lambda} = (P^{-1}\underline{c} - \underline{\hat{p}}) \end{aligned}$$

- Assuming full column rank of A^T or row rank of A

  $\Rightarrow$ linearly independent constraints in primal

LP for Scaled Reduced Costs

$$AP^{-2}A^T\underline{\lambda} = AP^{-1}(P^{-1}\underline{c} - \underline{\hat{p}})$$

$$\Rightarrow \underline{\lambda} = \left(AP^{-2}A^T\right)^{-1} AP^{-1}(P^{-1}\underline{c} - \underline{\hat{p}}) = M(P^{-1}\underline{c} - \underline{\hat{p}})$$

note that $\underline{\lambda} \in R(AP^{-1}) = R(M)$

- Eliminating $\underline{\lambda}$ from the dual problem we have:

$$\begin{aligned} &\max_{\underline{\hat{p}}} \ \underline{b}^T M(P^{-1}\underline{c} - \underline{\hat{p}}) = f(\underline{\hat{p}}) \\ &\text{s.t.} \quad H(\underline{\hat{p}} - P^{-1}\underline{c}) = \underline{0} \\ &\qquad \underline{\hat{p}} \geq \underline{0} \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} &\min_{\underline{\hat{p}}} \ \underline{b}^T M \underline{\alpha} \\ &\text{s.t.} \ H\underline{\alpha} = \underline{0} \\ &\text{where } \underline{\alpha} = \underline{\hat{p}} - P^{-1}\underline{c} \end{aligned}$$

and where

$H = I - P^{-1}A^T M$, a symmetric projection matrix

$\Rightarrow H^2 = H$

- In addition, we have

$$AP^{-1}H = 0 \Rightarrow \text{columns of } H \in N\left(AP^{-1}\right)$$

Direction to Update Dual Variables

- Note that we want $\underline{\alpha} \in N(H) \Rightarrow \underline{\alpha} \in R(P^{-1}A^T)$
- But $R\left(P^{-1}A^T\right) = R\left(M^T\right)$
- The gradient of $f(\underline{\hat{p}})$ w.r.t. the scaled reduced costs $\underline{\hat{p}}$ is

$$\underline{\hat{g}}_p = -M^T\underline{b} \in R\left(M^T\right) = R\left(P^{-1}A^T\right)$$

$\Rightarrow$ **Results**: The gradient w.r.t. the scaled reduced costs, $\underline{\hat{p}}$, already lies in the range space of $P^{-1}A^T$... *making the projection unnecessary*

- In terms of the original unscaled reduced costs, the projected gradient is:

$$\underline{g}_p = P\underline{\hat{g}}_p = -A^T\left(AP^{-2}A^T\right)^{-1}\underline{b}$$

- The corresponding feasible direction with respect to $\underline{\lambda}$ is:

$$\underline{d}_\lambda = -MM^T\underline{\hat{g}}_p = \left(AP^{-2}A^T\right)^{-1}\underline{b}$$

$$\Rightarrow \quad \underline{g}_p = -A^T\underline{d}_\lambda$$

- If $\underline{g}_p \geq \underline{0} \Rightarrow$ dual problem is unbounded $\Rightarrow$ primal is infeasible (assuming $\underline{b} \neq \underline{0}$)

Dual Affine Scaling Algorithm Steps - 1

- Otherwise, we replace $\underline{\lambda}$ by

$$\underline{\lambda} \leftarrow \underline{\lambda} + \alpha \underline{d}_\lambda$$

where $\alpha = \beta\alpha_{\max;}$ $\beta \approx 0.95$

$$\alpha_{\max} = \min\left\{ \frac{-p_i}{g_{p_i}} : g_{p_i} < 0, i = 1, 2, \ldots, n \right\}$$

- Note that primal solution $\underline{x}$ is:

$$\underline{x} = -P^{-2}\underline{g}_p = P^{-2}A^T\left(AP^{-2}A^T\right)^{-1}\underline{b}$$

since it satisfies $A\underline{x} = \underline{b}$

- **Dual Affine Scaling Algorithm**:
  - Start with a strictly feasible $\underline{\lambda}$, stopping criterion ε and β

$z_{old} = \underline{\lambda}^T \underline{b}$

for $k = 0, 1, \ldots k_{\max}$

$p = \underline{c} - A^T\underline{\lambda}$

$P = \text{Diag}\begin{bmatrix} p_1 & p_2 & \cdots & p_n \end{bmatrix}$

Compute the solution $\underline{d}_\lambda$ to

$\left(AP^{-2}A^T\right)\underline{d}_\lambda = \underline{b}$

$\underline{g}_p = -A^T\underline{d}_\lambda$

Dual Affine Scaling Algorithm Steps - 2

if $\underline{g}_p \geq \underline{0}$

Stop $\rightarrow$ unbounded dual solution $\Rightarrow$ primal is infeasible

else

$$\alpha = \beta \min\left\{ \frac{-p_i}{g_{p_i}} : g_{p_i} < 0, i = 1, 2, \cdots, n \right\}$$

$$\underline{\lambda} \leftarrow \underline{\lambda} + \alpha \underline{d}_\lambda \left(\Rightarrow \underline{p} \leftarrow \underline{p} + \alpha \underline{g}_p \text{ next step} \right)$$

$$z_{new} = \underline{\lambda}^T \underline{b}$$

if $\dfrac{\left| z_{new} - z_{old} \right|}{\max\left(1, \left| z_{old} \right|\right)} < \varepsilon$

stop $\rightarrow$ found an optimal solution $\underline{x} = -P^{-2} \underline{g}_p$

else

$z_{old} \leftarrow z_{new}$

end if

end if

end do

UCONN

Initial Feasible Solution for Dual Affine Scaling Algorithm

- Finding an initial strictly feasible solution for the dual affine scaling algorithm

$$\underline{\lambda}_0 = \left(\frac{\|\underline{c}\|_2}{\|A^T \underline{b}\|_2} \right) \underline{b}$$

  - Want to find a $\underline{p}$ s.t. $\underline{p} = -\xi \underline{e}$
  - Select initial ξ_0 as

$$\xi_0 = -2 \min \left\{ \left(\underline{c} - A^T \underline{\lambda} \right)_i : i = 1, 2, \cdots, m \right\}$$

  - Solve an $(m+1)$ variable LP:

$$\max_{\underline{\lambda},\xi} \quad \underline{\lambda}^T \underline{b} - \mu \xi$$
$$\text{s.t.} \quad A^T \underline{\lambda} - \xi \underline{e} \le \underline{c}$$

  - Select $\mu = \gamma \dfrac{\underline{\lambda}_0^T \underline{b}}{\xi_0}; \quad \gamma = 10^5$
  - The initial $(\underline{\lambda}_0, \xi_0)$ are feasible for the problem
  - Note :
    - ❖ If $\xi < 0$ at iteration $k \Rightarrow$ found a feasible $\underline{\lambda}$
    - ❖ If the algorithm is such that optimal $\xi < \varepsilon \Rightarrow$ dual is infeasible $\Rightarrow$ primal is unbounded

Primal Affine Scaling

- Primal affine scaling
  - Starting with $\underline{x}_0 \to \underline{x}_1 \to \cdots \to \underline{x}_k \to \underline{x}_{k+1} \to \cdots \underline{x}^*$
  - $\underline{x}_{k+1} = \underline{x}_k + \underline{d}_k \ni \left\| D_k^{-1}\underline{d}_k \right\| \le \beta; \;\; \beta < 2/3; \;\; D_k = \text{Diag}(\underline{x}_k)$
  - $\underline{d}_k$ is the solution of

$$\begin{aligned} &\min \underline{c}^T \underline{d} \\ &\text{s.t. } A\underline{d} = \underline{0} \qquad\qquad \text{recall } A\underline{x} = \underline{b} \Rightarrow A\underline{d}_k = \underline{0} \\ &\quad \left\| D_k^{-1}\underline{d} \right\| \le \beta \end{aligned}$$

  - Lagrangian:

$$L(\underline{d}, \underline{\lambda}, \mu) = \underline{c}^T \underline{d} - \underline{\lambda}^T A\underline{d} + \tfrac{\mu}{2}(\underline{d}^T D_k^{-2}\underline{d} - \beta^2)$$

$$\Rightarrow \mu D_k^{-2}\underline{d} + \underline{c} - A^T\underline{\lambda} = \underline{0} \qquad \Rightarrow \underline{d} = -\tfrac{1}{\mu} D_k^2(\underline{c} - A^T\underline{\lambda})$$

$$A\underline{d} = \underline{0}$$

$$\underline{d}^T D_k^{-2}\underline{d} = \beta^2$$

$$\Rightarrow \tfrac{1}{\mu^2}(\underline{c} - A^T\underline{\lambda})^T D_k^2(\underline{c} - A^T\underline{\lambda}) = \beta^2$$

$$\Rightarrow \mu = \frac{\left\| D_k(\underline{c} - A^T\underline{\lambda}) \right\|_2}{\beta}$$

$$\Rightarrow \underline{\lambda}_k = (AD_k^2A^T)^{-1}AD_k^2\underline{c}; \quad \underline{d}_k = -\beta \frac{D_k^2(\underline{c} - A^T\underline{\lambda})}{\left\| D_k(\underline{c} - A^T\underline{\lambda}) \right\|_2}$$

Primal Affine Scaling Algorithm Steps

- Affine Scaling Algorithms

Start with $\underline{x}_0 > \underline{0}$

for $k = 0, 1, 2, \cdots k_{\max}$

$D_k = \text{Diag}(\underline{x}_k)$

$(AD_k^2 A^T)\underline{\lambda}_k = AD_k^2 \underline{c}$

$\underline{p}_k = \underline{c} - A^T \underline{\lambda}_k$

If $\underline{p}_k \geq \underline{0}$ and $\underline{e}^T D_k \underline{p}_k < \varepsilon$, stop $\rightarrow$ found optimal solution

else if $-D_k^2 \underline{p}_k \geq \underline{0} \Rightarrow$ primal is unbounded (cost $= -\infty$)

else

$$\underline{x}_{k+1} = \underline{x}_k - \beta \frac{D_k^2 \underline{p}_k}{\left\| D_k^2 \underline{p}_k \right\|_2}$$

end if

end

- Initialize via big-M method

Potential Reduction Algorithm

- Potential Reduction Algorithm

Primal	Dual
$\min\ \underline{c}^T \underline{x}$	$\max_{\underline{\lambda},\underline{p}}\ \underline{\lambda}^T \underline{b}$
s.t. $A\underline{x} = \underline{b}$	s.t. $\underline{\lambda}^T A + \underline{p}^T = \underline{c}^T$
$\underline{x} \geq \underline{0}$	$\underline{p}^T \geq \underline{0}$

- Modified Barrier Function $f(\underline{x}, \underline{p}) = q\ln(\underline{p}^T \underline{x}) - \sum_{j=1}^{n} \ln x_j - \sum_{j=1}^{n} \ln p_j$

  Note: $\underline{c}^T \underline{x} - \underline{\lambda}^T \underline{b} = (\underline{p}^T + \underline{\lambda}^T A)\underline{x} - \underline{\lambda}^T A\underline{x} = \underline{p}^T \underline{x}$

  Duality gap if $\underline{x}$ is primal feasible and $(\underline{\lambda}, \underline{p})$ are dual feasible

  Idea: Starting with $\underline{x}_k > 0$ and $\underline{p}_k \geq \underline{0}$, find a direction $\underline{d}_k$ such that

$$\min_{\underline{d}}\ \nabla \underline{f}_k^T \underline{d}$$

$$\text{s.t.}\quad A\underline{d} = 0$$

$$\left\| D_k^{-1}\underline{d} \right\| \leq \beta < 1$$

$$\nabla_{\underline{x}}\ \underline{f}_k = \frac{q}{\underline{p}_k^T \underline{x}_k}\underline{p}_k - D_k^{-1}\underline{e} = \underline{\hat{c}}$$

Solution:

$$\underline{d}_k = -\beta D_k \frac{\underline{u}}{\|\underline{u}\|}$$

$$\underline{u} = D_k\left(\underline{\hat{c}}_k - A^T\left(AD_k^2A^T\right)^{-1} AD_k^2\underline{\hat{c}}_k\right)$$

Potential Reduction Algorithm Steps

- Start with $\underline{x}_0 > 0,\ \underline{p}_0 > 0,\ \underline{\lambda}_0,\ \beta < 1,\ \gamma < 1,\ q$

for $k = 0,1,2,\ldots k_{\max}$

If $\underline{p}_k^T \underline{x}_k < \varepsilon$ stop, found optimal solution.

Else $\quad D_k = \text{Diag}(\underline{x}_k)$

$$\underline{\hat{c}}_k = \frac{q}{\underline{p}_k^T \underline{x}_k} \underline{p}_k - D_k^{-1} \underline{e}$$

$$\underline{u} = D_k \left(I - A^T \left(A D_k^2 A^T \right)^{-1} A D_k^2 \right) \underline{\hat{c}}_k; \quad \underline{d}_k = -\beta D_k \frac{\underline{u}}{\|\underline{u}\|}$$

If $\|\underline{u}\| \geq \gamma \Rightarrow$ perform primal step

$$\underline{x}_{k+1} = \underline{x}_k + \underline{d}_k$$

$$\underline{p}_{k+1} = \underline{p}_k$$

$$\underline{\lambda}_{k+1} = \underline{\lambda}_k$$

Else $\quad \underline{x}_{k+1} = \underline{x}_k$

$$\underline{p}_{k+1} = \frac{\underline{p}_k^T \underline{x}_k}{q} D_k^{-1} \left(\underline{u}_k + \underline{e} \right)$$

$$\underline{\lambda}_{k+1} = \underline{\lambda}_k + \left(A D_k^2 A^T \right)^{-1} A D_k \left(D_k \underline{p}_k - \frac{\underline{p}_k^T \underline{x}_k}{q} \underline{e} \right)$$

end if

end if

end

See page 415 of Bertsimas & Tsitsiklis

Primal-dual Path following Algorithms

- Primal-dual path following algorithms

Barrier formulation of primal

$$\min \; \underline{c}^T \underline{x} - \mu \sum_{j=1}^{n} \ln x_j$$
$$\text{s.t.} \quad A\underline{x} = \underline{b}$$

Barrier formulation of dual

$$\max_{\underline{\lambda}, \underline{p}} \; \underline{\lambda}^T \underline{b} + \mu \sum_{j=1}^{n} \ln p_j$$
$$\text{s.t.} \quad \underline{\lambda}^T A + \underline{p}^T = \underline{c}^T$$

- Optimality Conditions

$$\left.\begin{aligned}
&A\underline{x} = \underline{b}\\
&A^T\underline{\lambda} + \underline{p} = \underline{c}\\
&\underline{c} - \mu D^{-1} e - A^T\underline{\lambda} = \underline{0}\\
&\Rightarrow \underline{c} - \mu D^{-1} e - \underline{c} + \underline{p} = \underline{0}\\
&\Rightarrow \mu\underline{e} = D\underline{p} = DP\underline{e}\\
&P = \text{Diag}(\underline{p})
\end{aligned}\right\} \Rightarrow \begin{cases} A\underline{x} - \underline{b} = \underline{0}\\ A^T\underline{\lambda} + \underline{p} - \underline{c} = \underline{0}\\ DP\underline{e} - \mu\underline{e} = \underline{0}\end{cases}$$

- Nonlinear equation because of $Dp\underline{e} = \mu\underline{e}$ (complementary slackness condition when μ=0)

This is a nonlinear equation! We will revisit this issue later

Primal-dual Path following Algorithms

- Solve via Newton's Method

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ P_k & 0 & D_k \end{bmatrix}\begin{bmatrix} \underline{d}_x \\ \underline{d}_\lambda \\ \underline{d}_p \end{bmatrix} = -\begin{bmatrix} A\underline{x}_k - \underline{b} \\ A^T\underline{\lambda}_k + \underline{p}_k - \underline{c} \\ D_k P_k \underline{e} - \mu_k \underline{e} \end{bmatrix}$$

Basis of **infeasible** primal-dual method with $\underline{x}_k > \underline{0}$, $\underline{p}_k > \underline{0}$, and $\underline{\lambda}_k$

$$\Rightarrow \begin{cases} A\underline{d}_x = \underline{0} \\ A^T\underline{d}_\lambda + \underline{d}_p = \underline{0} \\ P_k\underline{d}_x + D_k\underline{d}_p = \mu_k\underline{e} - D_k P_k\underline{e} \end{cases}$$

Basis of **feasible** primal-dual method

- Solution:

$$\underline{d}_x = E_k\left(I - R_k\right)\underline{v}_k$$

$$\underline{d}_\lambda = -\left(AE_k^2A^{\mathrm{T}}\right)^{-1}AE_k\underline{v}_k$$

$$\underline{d}_p = E_k^{-1}P_k\underline{v}_k$$

where

$$E_k = D_k P_k^{-1}$$

$$R_k = E_k A^T\left(AE_k^2A^T\right)^{-1}AE_k$$

$$\underline{v}_k = D_k^{-1}E_k(\mu_k\underline{e} - D_kP_k\underline{e})$$

use $\mu_k = \dfrac{\underline{x}_k^T\underline{p}_k}{n}$

Primal-dual Path following Algorithm Steps

- Initialize

  $\underline{x}_0 > 0,\ \underline{p}_0 > 0,\ \underline{\lambda}_0, (\alpha < 1)$

for $k = 0, 1, 2, \ldots k_{\max}$

If $\underline{p}_k^T \underline{x}_k < \varepsilon$, stop

else (compute Newton directions)

$$\mu_k = \frac{\underline{x}_k^T \underline{p}_k}{n}$$

$$D_k = \mathrm{Diag}(\underline{x}_k)$$

$$P_k = \mathrm{Diag}(\underline{p}_k)$$

compute $\underline{d}_x, \underline{d}_\lambda$ and $\underline{d}_p$

find step lengths via

$$\beta_p = \min\left\{1, \alpha \min_{(i:d_{xi}<0)} \left(\frac{-x_{ki}}{d_{xi}}\right)\right\}$$

$$\beta_d = \min\left\{1, \alpha \min_{(i:d_{pi}<0)} \left(\frac{-p_{ki}}{d_{pi}}\right)\right\}$$

$$\underline{x}_{k+1} = \underline{x}_k + \beta_p \underline{d}_k$$

$$\underline{\lambda}_{k+1} = \underline{\lambda}_k + \beta_d \underline{d}_\lambda$$

$$\underline{p}_{k+1} = \underline{p}_k + \beta_d \underline{d}_p$$

end

Relationships among Path following Algorithms

- <u>Relationships</u>:
  - $\underline{d}_{affine} = -D^2\left(I - A^T\left(AD^2A^T\right)^{-1}AD^2\right)\underline{c}$
  - $\underline{d}_{primal\ path-following} = \left(I - D^2A^T\left(AD^2A^T\right)^{-1}A\right)\left(D\underline{e} - \frac{1}{\mu}D^2\underline{c}\right)$
  - When $\mu = \infty$, the corresponding direction is called *centering direction* because in this case $\underline{x}(\mu)$ is the *analytic center* of the feasible set.

$$\underline{d}_{centering} = \left(I - D^2A^T\left(AD^2A^T\right)^{-1}A\right)D\underline{e}$$

$$\Rightarrow \underline{d}_{primal\ path-following} = \underline{d}_{centering} + \frac{1}{\mu}\underline{d}_{affine}$$

$$\underline{d}_{potential} = \underline{d}_{centering} + \frac{q}{p^T\underline{x}}\underline{d}_{affine}$$

- Both potential and path following algorithms have polynomial complexity. There is no such result for affine scaling.
  - ⇒ centering directions are responsible for polynomiality of path following and potential reduction algorithms.

Implementation Issues

- Least-squares subproblem: Implementation Issues
  - Generally A is sparse
  - Major computational step at each iteration

    $AP^{-2}A^T\underline{d} = \underline{b}$ ⋯ Affine scaling

    $AD^2A^T\underline{\lambda} = AD^2(\underline{c} - \mu D^{-1}\underline{e}) = AD(D\underline{c} - \mu\underline{e})$ ⋯Barrier function method

    Similar equations in path following and potential reduction algorithms.
  - Key: Need to solve a symmetric positive definite system $\Sigma\underline{y} = \underline{b}$
- Solution Approaches:
- Direct methods:

  a) Cholesky factorization: $\Sigma = SS^T$, $S = \Delta_{lower}$

  b) ***LDL^T*** factorization: $\Sigma = LDL^T$; $L = \text{unit } \Delta_{lower}$

  c) ***QR*** factorization of $P^{-1}A^T$ or DA^T
- Methods to speed up factorization
  - During each iteration only D or P^{-1} changes, while A remains unaltered
    - Nonzero structure of Σ is static throughout
    - So, during the first iteration, keep track of the list of numerical operations performed

Factorization Methods

- Perform factorization only if the diagonal scaling matrix has changed significantly
  - Consider $\Sigma = AP^{-2}A^T$
  - Replace P by $\bar{P}$ where

$$\bar{P}_{ii}^{new} = \begin{cases} \bar{P}_{ii}^{old} & \text{if } \dfrac{\left|P_{ii} - \bar{P}_{ii}^{old}\right|}{\left|\bar{P}_{ii}^{old}\right|} < \delta \\ P_{ii} & \text{otherwise} \end{cases}$$

  - $\delta \sim 0.1$
  - Define $\Delta P_{ii} = \bar{P}_{ii}^{new} - \bar{P}_{ii}^{old}$
  - Then $\Sigma^{new} = \Sigma^{old} + \sum_{\{i:\Delta P_{ii} \neq 0\}} \Delta P_{ii} \underline{a}_i \underline{a}_i^T \quad \underline{a}_i = i^{th}$ column of A
  - So, use rank-one modification methods (ECE6435, Lecture 8)
- Perform pivoting to reduce fill-ins ⇒ having nonzero elements in factors where there are zero elements in Σ
  - Recall that $(P\Sigma P^T)P\underline{y} = P\underline{b}$
  - Unfortunately, finding the optimal permutation matrix to reduce fill-in is NP-complete
  - However, ∃ heuristics
    - ❖ Minimum degree
    - ❖ Minimum local fill-in

Incomplete Cholesky Algorithm

- Combine with an iterative method, if we have a few dense columns in A that will make impractically dense Σ (recall the outer product representation)
  - $\Rightarrow$ Hybrid factorization and conjugate gradient method called a preconditioned conjugate gradient method works well
- <u>Idea</u>: At iteration k, split columns of A into two parts $[S\ \bar{S}]$ where columns of A_s are sparse (i.e., have density < $\lambda(\approx 0.3)$)
  - Form $A_s P^{-2} A_s^T$
  - Find <u>incomplete</u> Cholesky factor L such that $Z_s = A_s P^{-2} A_s^T = LL^T$
  - Basically the idea is to step through the Cholesky decomposition, but setting $l_{ij} = 0$ if the corresponding $\Sigma_{s_{ik}} = 0$

<u>Incomplete Cholesky Algorithm</u>

for $k = 1, \ldots, m$ do
  $l_{kk} = \sqrt{\Sigma_{s_{kk}}}$
  for $i = k+1, \cdots, m$ do
    if $\Sigma_{s_{ik}} \neq 0$
      $l_{ik} = \frac{\Sigma_{s_{ik}}}{l_{kk}}$
    end if
  end do
  for $j = k+1, \ldots, m$ do
    for $i = j, \ldots, m$ do
      if $\Sigma_{s_{ij}} \neq 0$
        $\Sigma_{s_{ij}} = \Sigma_{s_{ij}} - l_{ik} l_{jk}$
      end if
    end do
  end do
end do

Conjugate Gradient Algorithm

- Now consider the original problem $\Sigma \underline{y} = A^T P^{-2} A \underline{y} = \underline{b}$

$$L^{-1}\Sigma\left(L^{-1}\right)^T L^T \underline{y} = L^{-1}\underline{b}$$

$$\Rightarrow Q\underline{u} = \underline{f}$$

where $\quad Q = L^{-1}\Sigma(L^{-1})^T;\ \underline{u} = L^T \underline{y};\ \underline{f} = L^{-1}\underline{b}$

- Solve $Q\underline{u} = \underline{f}$ via conjugate gradient algorithm … ECE6435
- Conjugate Gradient Algorithm:

$\underline{u} = \underline{f}$ …initial solution

$c = \|\underline{f}\|_2$ …norm of RHS

$\underline{r} = \underline{f} - Q\underline{u}$ …initial residual

(negative gradient of $\left(\frac{1}{2}\underline{u}^T Q\underline{u} - \underline{u}^T \underline{f}\right)$)

$p = \|\underline{r}\|_2^2$ …square norm of initial residual

$\underline{d} = \underline{r}$ …initial direction

$k = 0$

while $\frac{\sqrt{p}}{c} \geq \varepsilon$ and $k \leq k_{max}$ do

$\underline{\omega} = Q\underline{d}$

$\alpha = \frac{r}{\underline{d}^T Q\underline{d}} \cdots$ step length

$\underline{u} = \underline{u} + \alpha\underline{d} \cdots$ new solution

$\underline{r} = \underline{r} - \alpha\underline{w} \cdots$ new residual, $\underline{r} = \underline{f} - Q\underline{u}$

$\beta = \frac{\|\underline{r}\|_2^2}{p} \cdots$ parameter to update direction

$\underline{d} = \underline{r} + \beta\underline{d} \cdots$ new direction

$p = \|\underline{r}\|_2^2$

$k = k + 1$

end do

Computational load … $O(m^2 + 10m)$

Need to store only four vectors: $\underline{u}$, $\underline{r}$, $\underline{d}$ and $\underline{w}$

Mehrotra's Correction

Recall $Dp\underline{e} = \mu e$ is a nonlinear equation

$$DP\underline{e} = \mu_k \underline{e}$$
$$D = D_k + \Delta D_k ; P = P_k + \Delta P_k$$
$$\left(D_k + \Delta D_k\right)(P_k + \Delta P_k)\underline{e} = \mu_k \underline{e}$$

$$P_k \underline{d}_x + D_k \underline{d}_p = \mu_k \underline{e} - D_k P_k \underline{e} - \Delta D_k \Delta P_k \underline{e} = \mu_k \underline{e} - D_k P_k \underline{e} - \underline{d}_x \circ \underline{d}_p$$
$$\underline{d}_x \circ \underline{d}_p = \text{Hadamard Product} = [d_{x1} d_{p1} \; d_{x2} d_{p2} \; \text{........} d_{xn} d_{pn}]$$

Mehrotra's Correction: Solve for directions twice

1. Predictor step: First solve by setting $\underline{d}_x = \underline{d}_p = 0$ in RHS
2. Corrector step: Solve it again by plugging the values from step 1 in RHS

- Factorization makes this easy to implement
- Speeds up convergence

Simplex versus Interior Point Methods

- Comparison of simplex and dual affine scaling methods
  - Three types of test problems
- NETLIB test problems
  - 31 test problems
  - The library and test problem can be accessed via electronic mail: *netlib@anl-mcs* (ARPANET/CSNET) or research! netlib (UNIX network)
  - # of variables n ranged from 51 to 5533
  - # of constraints m ranged from 27 to 1151
  - # of non-zero elements in A ranged from 102 to 16276
  - Comparisons on IBM 3090

	Simplex	**Affine Scaling**
Iterations	(6,7157)	(19,55)
Ratio of time per iteration	(0.093, 0.356)	1
Total cpu time range (secs)	(0.01, 217.67)	(0.05, 31.70)
Ratio of cpu time (Simplex/Affine)	(0.2, 10.7)	1

Simplex versus Interior Point Methods

- Multi-commodity Network Flow problem
  - Specialized LP algorithms exist that are better than simplex
  - ∃ a program to generate random multi-commodity network flow problem called MNETGN
  - 11 problems were generated
  - # of variables $n \in (2606,8800)$
  - # of constraints $m \in (1406,4135)$
  - Non-zero elements in A ranged from 5212 to 22140

	Simplex	Specialized Simplex	
	MINOS 4.0	MCNF 85	Affine Scaling
Total # of iterations	(940, 21915)	(931, 16624)	(28, 35)
Ratios of time per iteration (w.r.t. Affine Scaling)	(0.010, 0.069)	(0.0018, 0.0404)	1
Total CPU time (secs)	(12.73, 1885.34)	(7.42, 260.44)	(6.51, 309.50)
Ratios of CPU times w.r.t. Affine Scaling	(1.96, 11.56)	(0.59, 4.15)	1

Simplex versus Interior Point Methods

- Timber Harvest Scheduling problems
  - 11 timber harvest scheduling problems using a program called FOR-PLAN
  - # of variables ranged from 744 to 19991
  - # of constraints ranged from 55 to 316
  - Non-zero elements in *A* ranged from 6021 to 176346

	Simplex (MINOS 4.0) Default Pricing	**Affine Scaling**
Total # of iterations	(534, 11364)	(38,71)
Ratio of time per iteration	(0.0141, 0.2947)	1
Total CPU time (secs)	(2.74, 123.62)	(0.85, 43.80)
Ratios of CPU times	(1.52, 5.12)	1

Summary and References

- Promising approach for large real-world LP problems
- Summary
  - Reviewed duality
  - Dual simplex and primal-dual algorithm
  - Interior point methods
    - Path following (primal, primal-dual)
    - Affine scaling
    - Potential reduction
- References

1) D. Goldfarb and M. J. Todd, "Linear Programming," ch. II in (Eds.) G. L. Nemhauser, A. H. G. Rinnoy Kan and M. J. Todd, Optimization, vol. I., North-Holland, pp. 73-170.
2) I. Adler, M. G. C. Resende, G. Vega and N. Karmarkar, "An implementation of Karmarkar's Algorithm for Linear Programming," Mathematical Programming, vol. 44, 1989, pp. 297-335.
3) I. Adler, N. Karmarkar, M. G. C. Resende, and G. Vega, "Data Structures and Programming Techniques for the Implementation of Karmarkar's Algorithm," ORSA Journal on Computing, Vol. 1, No. 2, 1989.
4) G. Golub and C. Van Loan, Matrix Computations, John Hopkins University Press, 1989.
5) D. Bertsimas and J. N. Tsitsiklis, Introduction to Linear Optimization, Athena Scientific, 1997.