ORF 522: Lecture 11

Linear Programming: Chapter 13
Network Flows: Algorithms

Robert J. Vanderbei

October 17, 2013

Slides last edited at 11:18am on Wednesday 16th April, 2014
Agenda

- Primal Network Simplex Method
- Dual Network Simplex Method
- Two-Phase Network Simplex Method
- One-Phase Primal-Dual Network Simplex Method
- Planar Graphs
- Integrality Theorem
Primal Network Simplex Method

Used when all primal flows are nonnegative (i.e., primal feasible).

Pivot Rules:

**Entering arc:** Pick a nontree arc having a negative (i.e. infeasible) dual slack.

**Leaving arc:** Add entering arc to make a cycle. Leaving arc is an arc on the cycle, pointing in the opposite direction to the entering arc, and of all such arcs, it is the one with the smallest primal flow.


d| f | 15 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>27</td>
</tr>
<tr>
<td>b</td>
<td>9</td>
</tr>
<tr>
<td>a</td>
<td>19</td>
</tr>
<tr>
<td>g</td>
<td>32</td>
</tr>
<tr>
<td>c</td>
<td>33</td>
</tr>
</tbody>
</table>

Obj value = 724


d| f | 15 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>27</td>
</tr>
<tr>
<td>b</td>
<td>9</td>
</tr>
<tr>
<td>a</td>
<td>19</td>
</tr>
<tr>
<td>g</td>
<td>32</td>
</tr>
<tr>
<td>c</td>
<td>33</td>
</tr>
</tbody>
</table>

Obj value = 559
Primal Method—Second Pivot

Explanation of leaving arc rule:

- Increase flow on (d,e).
- Each unit increase produces a unit increase on arcs pointing in the same direction.
- Each unit increase produces a unit decrease on arcs pointing in the opposite direction.
- The first to reach zero will be the one pointing in the opposite direction and having the smallest flow among all such arcs.
Primal Method—Third Pivot

Entering arc: \((c, g)\)
Leaving arc: \((c, e)\)

Optimal!
Dual Network Simplex Method

Used when all dual slacks are nonnegative (i.e., dual feasible).

Obj value = \(-62\)

Leaving arc: \((g,a)\)
Entering arc: \((d,e)\)

Pivot Rules:

*Leaving arc:* Pick a tree arc having a negative (i.e. infeasible) primal flow.

Obj value = \(106\)

*Entering arc:* Remove leaving arc to split the spanning tree into two subtrees. Entering arc is an arc reconnecting the spanning tree with an arc in the opposite direction, and, of all such arcs, is the one with the smallest dual slack.
Dual Network Simplex Method—Second Pivot

Obj value = 106

Leaving arc: (d,a)
Entering arc: (b,c)

Obj value = 316

Leaving arc: (d,a)
Entering arc: (b,c)

Optimal!
Explanation of Entering Arc Rule

Recall initial tree solution:

- Leaving arc: (g,a)
- Entering arc: (d,e)

- Remove leaving arc. Need to find a reconnecting arc.
- Since the leaving arc has a negative flow, there is a net supply at the subtree attached to the head node and a net demand at the subtree attached to the tail node.
- So, reconnecting with an arc that spans in the same direction does not improve anything.
- Hence, only consider arcs spanning the two subtrees in the opposite direction.

- Consider a potential arc reconnecting in the opposite direction, say (b,c).
  - Its dual slack will drop to zero.
  - All other reconnecting arcs pointing in the same direction will drop by the same amount.
  - To maintain nonnegativity of all the others, must pick the one that drops the least.
Example.

- Turn off display of dual slacks.
- Turn on display of artificial dual slacks.
Two-Phase Method–First Pivot

Use dual network simplex method.
Leaving arc: (d,e)  Entering arc: (e,f)

Dual Feasible!
Two-Phase Method–Phase II

- Turn off display of artificial dual slacks.
- Turn on display of true dual slacks.
Two-Phase Method—Second Pivot

Entering arc: (g,b)
Leaving arc: (g,f)

Obj value = 500

Obj value = 290
Two-Phase Method—Third Pivot

- Entering arc: (f,c)
- Leaving arc: (a,f)

**Obj value = 290**

**Optimal!**

**Obj value = -46**
Click here (or on any displayed network) to try out the online network simplex pivot tool.
• Artificial flows and slacks are multiplied by a parameter $\mu$.
• In the Figure, $6,1$ represents $6 + 1\mu$.
• **Question:** For which $\mu$ values is dictionary optimal?
• **Answer:**

\[
\begin{align*}
1 + \mu & \geq 0 \quad (a, b) & \mu & \geq 0 \quad (f, b) \\
-2 + \mu & \geq 0 \quad (a, c) & 20 + \mu & \geq 0 \quad (c, e) \\
\mu & \geq 0 \quad (a, d) & -1 + \mu & \geq 0 \quad (f, c) \\
\mu & \geq 0 \quad (e, a) & -9 + \mu & \geq 0 \quad (g, d) \\
-3 + \mu & \geq 0 \quad (a, g) & 12 + \mu & \geq 0 \quad (f, e) \\
\mu & \geq 0 \quad (b, c) & 6 + \mu & \geq 0 \quad (g, e) \\
3 + \mu & \geq 0 \quad (b, d)
\end{align*}
\] (1)

• That is, $9 \leq \mu < \infty$.
• Lower bound on $\mu$ is generated by arc $(g,d)$.
• Therefore, $(g,d)$ enters.
• Arc $(a,d)$ leaves.
Second Iteration

- Range of $\mu$ values: $2 \leq \mu \leq 9$.
- Entering arc: (a,c)
- Leaving arc: (b,c)

New tree:
Third Iteration

- Range of $\mu$ values: $1.5 \leq \mu \leq 2$.
- Leaving arc: (a,g)
- Entering arc: (g,e)

New tree:
Fourth Iteration

- **Range of $\mu$ values:**
  \[ 1 \leq \mu \leq 1.5. \]

- **A tie:**
  - Arc $(f,b)$ enters, or
  - Arc $(f,c)$ leaves.

- **Decide arbitrarily:**
  - Leaving arc: $(f,c)$
  - Entering arc: $(f,b)$
Fifth Iteration

- Range of $\mu$ values: $1 \leq \mu \leq 1$.
- Leaving arc: (f,b)
- Nothing to Enter.

Primal Infeasible!
Click here (or on any displayed network) to try out the online network simplex pivot tool.
**Definition.** Network is called **planar** if can be drawn on a plane without intersecting arcs.

**Theorem.** Every planar network has a geometric dual—dual nodes are faces of primal network.

Notes:
- Dual node $A$ is “node at infinity”.
- Primal spanning tree shown in red.
- Dual spanning tree shown in blue (don’t forget node $A$).

**Theorem.** A dual pivot on the primal network is exactly a primal pivot on the dual network.
**Definition.** Network is called planar if can be drawn on a plane without intersecting arcs.

**Theorem.** Every planar network has a geometric dual—dual nodes are faces of primal network.

Notes:
- Dual node A is “node at infinity”.
- Primal spanning tree shown in red.
- Dual spanning tree shown in blue (don’t forget node A).

**Theorem.** A dual pivot on the primal network is exactly a primal pivot on the dual network.
**Planar Networks**

**Definition.** *Network is called planar if can be drawn on a plane without intersecting arcs.*

**Theorem.** *Every planar network has a geometric dual—dual nodes are faces of primal network.*

Notes:
- Dual node $A$ is “node at infinity”.
- Primal spanning tree shown in red.
- Dual spanning tree shown in blue (don’t forget node $A$).

**Theorem.** *A dual pivot on the primal network is exactly a primal pivot on the dual network.*
Planar Networks—Algebraic Dual

Primal flow problem:

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = -b \\
& \quad x \geq 0
\end{align*}
\]

There is one redundant equation. Drop it and rewrite the equations:

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad A_0 x = -b_0 \\
& \quad x \geq 0
\end{align*}
\]

A spanning tree corresponds to a basic solution:

\[A_0 = [B \ N].\]

The dual has free variables and slacks:

\[
\begin{align*}
\text{maximize} & \quad - \begin{bmatrix} b_0^T \\ 0 \end{bmatrix} y \\
\text{subject to} & \quad \begin{bmatrix} B^T & I & 0 \\ N^T & 0 & I \end{bmatrix} \begin{bmatrix} y \\ z_B \\ z_N \end{bmatrix} = \begin{bmatrix} c_B \\ c_N \end{bmatrix} \\
& \quad z \geq 0
\end{align*}
\]

Solve for \(y\):

\[
y = B^{-T} (c_B - z_B).\]

Eliminate from problem:

\[
\begin{align*}
\text{maximize} & \quad -(B^{-1}b_0)^T (c_B - z_B) \\
\text{subject to} & \quad \begin{bmatrix} -(B^{-1}N)^T & I \end{bmatrix} \begin{bmatrix} z_B \\ z_N \end{bmatrix} = c_N - (B^{-1}N)^T c_B \\
& \quad z \geq 0
\end{align*}
\]
```matlab
% a node/arc incidence matrix
A = [-1 0 0 0 0 1 0 0;...
    1 -1 1 0 0 0 1 0;...
    0 0 -1 -1 0 0 0 0;...
    0 0 0 1 1 0 0 0;...
    0 1 0 0 -1 0 0 1;...
    0 0 0 0 0 -1 -1 -1];

[m,n] = size(A);
A0 = A(1:end-1,:); % drop last row
B=A0(:,[1 2 4 5 7]); % these are the tree arcs
N=A0(:,[3 6 8]); % these are the nontree arcs
Binv = inv(B);
BinvN = Binv*N;
BN = [B N];
alg_dual = [-BinvN' eye(n-m+1)]
BinvNt = BinvN'

% node/arc incidence matrix for graphical dual
AA = [ 1 0 -1 1 -1 1 0 -1;...
      -1 0 0 0 0 -1 1 0;...
      0 -1 0 0 0 0 -1 1;...
      0 1 1 -1 1 0 0 0];
AA0 = AA(1:end-1,:); % drop last row
geom_dual = AA
BB = AA0(:,[3 6 8]);
NN = AA0(:,[1 2 4 5 7]);
BBinvNN = inv(BB)*NN
```
Matlab Output

alg_dual =
    0  1  -1  1  0  1  0  0  0
    1  0  0  0  -1  0  1  0
    0  -1  0  0  -1  0  0  1

BinvNt =
    0  -1  1  -1  0
    -1  0  0  0  1
    0  1  0  0  1

geom_dual =
    1  0  -1  1  -1  1  0  -1
    -1  0  0  0  0  -1  1  0
    0  -1  0  0  0  0  -1  1
    0  1  1  -1  1  0  0  0

BBinvNN =
    0  1  -1  1  0
    1  0  0  0  -1
    0  -1  0  0  -1

Notes:
• If the middle row of alg_dual is negated, then the matrix can be made into a node/arc incidence matrix by adding an appropriately chosen forth row. In general it is not obvious how to connect alg_dual to geom_dual.
• But, BinvNt = -BBinvNN.
Euler’s Formula

Dimension of primal problem with (one) redundant equation:

\[ m \times n \]

Dimension of primal problem without redundant equation:

\[ (m - 1) \times n \]

Dimension of dual problem with slacks, free variables, but without redundancy:

\[ n \times (m - 1 + n) \]

Dimension of dual problem without free variables and without redundancy:

\[ (n - (m - 1)) \times n \]

Number of faces equals number of dual nodes:

\[ n - m + 2 \]

**Euler’s formula:**

\[ \text{nodes} - \text{arcs} + \text{faces} = m - n + (n - m + 2) = 2 \]
Theorem. Assuming integer data, every basic feasible solution assigns integer flow to every arc.

Corollary. Assuming integer data, every basic optimal solution assigns integer flow to every arc.