Linear Programming

Jie Wang

University of Massachusetts Lowell
Department of Computer Science

- **Linear function**:

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} a_i x_i,$$

where $a_1, a_2, \ldots, a_n$ are real numbers and $x_1, x_2, \ldots, x_n$ are valuables.

- **Linear equality**: $f(x_1, x_2, \ldots, x_n) = b$.
- **Linear inequalities**:
  - $f(x_1, x_2, \ldots, x_n) > b$.
  - $f(x_1, x_2, \ldots, x_n) \geq b$.
  - $f(x_1, x_2, \ldots, x_n) < b$.
  - $f(x_1, x_2, \ldots, x_n) \leq b$.
- Linear inequalities and linear equalities are often referred to as **linear constraints**.
- LP problems are to maximize (or minimize) a linear objective function with linear constraints.

LP Standard Form

Standard form: Maximization + linear inequalities.

$$
\begin{array}{lrcrcr}
\text{maximize} & x_1 & + & x_2 & & \\
\text{subject to} & 4x_1 & - & x_2 & \leq & 8 \\
 & 2x_1 & + & x_2 & \leq & 10 \\
 & 5x_1 & - & 2x_2 & \geq & -2 \\
 & & & x_1 & \geq & 0 \\
 & & & x_2 & \geq & 0.
\end{array}
$$

- A minimization problem can be converted to an equivalent maximization problem, and vice versa.
- For example, maximizing $f(x_1, x_2, \ldots, x_n)$ under a set of constraints is equivalent to minimizing $-f(x_1, x_2, \ldots, x_n)$ with the same set of constraints.

LP Slack Form

Maximization + linear equalities with slack variables.

$$\begin{array}{llllllll}
\text{maximize} & x_1 & + & x_2 & & & & \\
\text{subject to} & x_3 & = & 8 & - & 4x_1 & + & x_2 \\
 & x_4 & = & 10 & - & 2x_1 & - & x_2 \\
 & x_5 & = & -2 & - & 5x_1 & + & 2x_2 \\
 & & & & & x_i & \geq & 0 \\
 & & & & & i & = & 1, 2, 3, 4, 5.
\end{array}$$

- Variables on the left-hand side of the equalities are **basic variables** and on the right-hand side **nonbasic variables**.
- Initially, basic variables are slack variables, and nonbasic variables are the original variables. These can be changed during the simplex procedure.
- The slack form turns an LP problem in a lower dimension with a set of inequality constraints to an equivalent LP problem in a higher dimension with a set of equality constraints. Equalities are much easier to handle than inequalities.

LP Applications

- An LP formulation for single-pair shortest path of (s, t) .

$$\begin{array}{rll} \text{maximize} & d_t & \\ \text{subject to} & d_v & \leq d_u + w(u,v) \text{ for each edge } u \to v \in E, \\ & d_s & = 0. \end{array}$$

  Note: it's maximization, not minimization.

- An LP formulation for maximum flow.

$$\begin{array}{lrcll} \text{maximize} & \sum_{v \in V} f_{sv} & - & \sum_{v \in V} f_{vs} & \\ \text{subject to} & f_{uv} & \leq & c(u,v) & \text{for each } u, v \in V \\ & \sum_{v \in V} f_{uv} & = & \sum_{v \in V} f_{uv} & \text{for each } u \in V - \{s, t\} \\ & f_{uv} & \geq & 0 & \text{for each } u, v \in V. \end{array}$$

Minimum-Cost Flow

- Each edge $u \to v$ in a flow network has a capacity $c(u, v)$ and a cost $a(u, v)$.
- Want to send d units of flow from s to t while minimizing the total cost.

$$
\begin{array}{lllcl}
\text{minimize} & \sum_{u \to v \in E} a(u,v) f_{uv} & & & \\
\text{subject to} & \text{for each } u, v \in V: & f_{uv} & \leq & c(u,v) \\
 & \text{for each } u \in V - \{s, t\}: & \sum_{v \in V} f_{uv} & = & \sum_{v \in V} f_{uv} \\
 & \sum_{v \in V} f_{sv} - \sum_{v \in V} f_{vs} & & = & d \\
 & \text{for each } u, v \in V: & f_{uv} & \geq & 0.
\end{array}
$$

Multicommodity Flow

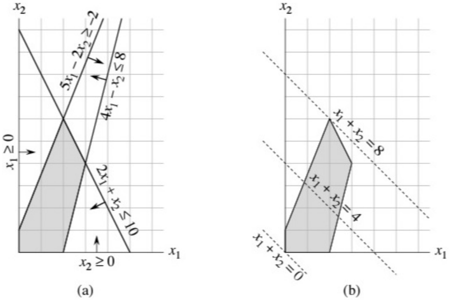
- Given a digraph $G = (V, E)$ with k different commodities $K_1, K_2, \ldots, K_k$, where
  - Each edge $u \to v \in E$ has a capacity $c(u, v) \geq 0$.
  - G has no anti-parallel edges.
  - $K_i = (s_i, t_i, d_i)$, representing the source, sink, and demand of commodity i.
- Let f_i denote the flow for commodity i, where f_{iuv} is the flow of commodity i from u to v.
- Let $f_{uv} = \sum_{i=1}^{k} f_{iuv}$ denote the **aggregate flow**.
- Want to determine if such a flow exists. That is, there is no objective function to optimize.

Multicommodity Flow Continued

$$
\begin{array}{lrcl}
\text{optimize} & 0 & & \\
\text{subject to} & \sum_{i=1}^{k} f_{iuv} & \leq & c(u,v) \text{ for each } u, v \in V \\
& \sum_{v \in V} f_{iuv} - \sum_{v \in V} f_{iuv} & = & 0 \text{ for each } u \in V - \{s_i, t_i\} \\
& \sum_{v \in V} f_{i,s_i,v} - \sum_{v \in V} f_{i,v,s_i} & = & d_i \\
& f_{iuv} & \geq & 0 \text{ for each } u, v \in V \text{ and} \\
& i & = & 1, 2, \ldots, k.
\end{array}
$$

Feasible Solutions and Feasible Regions

- **Feasible solutions** are values of the variables that satisfy the constraints.
- **Feasible region** (a.k.a. **simplex**) is the set of feasible solutions, which is a convex polytope.



A convex polytope is a geometric shape in d-dimensional space without any dents.

LP Algorithms

- The optimal solution occurs at the boundary of the simplex.
    - Could be at exactly one vertex. In this case there is only one optimal solution.
    - Could be at a line segment. In this case there are multiple optimal solutions.
- The simplex algorithm, while having exponential runtime in the worst case, is **very** efficient in practice.
- Other algorithms include (1) the ellipsoid method, the first-known polynomial-time algorithm; and (2) the Interior-Point method, which walks through the interior of the simplex instead of the vertices.

The Simplex Algorithm

The simplex algorithm finds a basic solution from the slack form iteratively as follows:

1. Set each nonbasic variable to zero.
2. Compute the values of the basic variables from the equality constraints.
    - If a nonbasic variable in a constraint causes a basic variable in the constraint to become 0, then the constraint is **tight**.
3. The goal of an iteration is to reformulate the linear program so that the basic solution (when the nonbasic variables are 0) has a greater objective value.
    - This is done by (performing a **pivot**): Select a nonbasic variable x_e (a.k.a. the **entering variable**); maximize it such that all constraints are satisfied.
    - The variable x_e becomes basic and another variable x_l (a.k.a. the **leaving variable**) becomes nonbasic.
4. The simplex algorithm terminates when all of the coefficients appearing in the objective function become negative.

An Example

$$
\begin{array}{lccccccc}
\text{maximize} & 3x_1 & + & x_2 & + & 2x_3 & & \\
\text{subject to} & x_1 & + & x_2 & + & 3x_3 & \leq & 30 \\
 & 2x_1 & + & 2x_2 & + & 5x_3 & \leq & 24 \\
 & 4x_1 & + & x_2 & + & 2x_3 & \leq & 36 \\
 & & & & & x_1, x_2, x_3 & \geq & 0.
\end{array}
$$

Slack form:

$$
\begin{aligned}
\text{maximize } & 3x_1 + x_2 + 2x_3 \\
\text{subject to } & x_4 = 30 - x_1 - x_2 - 3x_3 \\
& x_5 = 24 - 2x_1 - 2x_2 - 5x_3 \\
& x_6 = 36 - 4x_1 - x_2 - 2x_3 \\
& x_1, x_2, x_3, x_4, x_5, x_6 \geq 0.
\end{aligned}
$$

Simplex Algorithm Example Continued

- Set $x_1 = 0, x_2 = 0, x_3 = 0$ and get $x_4 = 30$, $x_5 = 24$, and $x_6 = 36$.
- Select entering variable x_1 and leaving variable x_6.
- Solve the corresponding equation for x_1 to get

$$x_1 = 9 - \frac{x_2}{4} - \frac{2x_3}{4} - \frac{x_6}{4}.$$

- Maximize it to get $x_1 = 9$.
- Substitute the right-hand side of x_1 into each remaining equation:

$$\begin{aligned} z &= 27 + \frac{x_2}{4} + \frac{x_3}{2} - \frac{3x_6}{4} \\ x_1 &= 9 - \frac{x_2}{4} - \frac{x_3}{4} - \frac{x_6}{4} \\ x_4 &= 21 - \frac{3x_2}{4} - \frac{5x_3}{2} + \frac{x_6}{4} \\ x_5 &= 6 - \frac{3x_2}{2} - 4x_3 + \frac{x_6}{2}. \end{aligned}$$

Simplex Algorithm Example Continued

- Set x_2, x_3, and x_6 to 0 to get $z = 27, x_1 = 9, x_4 = 21$, and $x_5 = 6$.
- Select entering variable x_3 and leaving variable x_5. Note that x_6 cannot be chosen for it will decrease the value of z.
- Solve for x_3 to get

$$x_3 = \frac{3}{2} - \frac{3x_2}{8} + \frac{x_6}{8} - \frac{x_5}{4}.$$

- The maximum value for x_3 is $\frac{3}{2}$.
- Substitute x_3's left-hand side into the rest of the equations to obtain

$$\begin{aligned} z &= \frac{111}{4} + \frac{x_2}{16} - \frac{x_5}{8} - \frac{11x_6}{16} \\ x_1 &= \frac{33}{4} - \frac{x_2}{16} + \frac{x_5}{8} - \frac{5x_6}{16} \\ x_3 &= \frac{3}{2} - \frac{3x_2}{8} + \frac{x_6}{8} - \frac{x_5}{4} \\ x_4 &= \frac{69}{4} + \frac{3x_2}{16} + \frac{5x_5}{8} - \frac{x_6}{16} \end{aligned}$$

Simplex Algorithm Example Continued

- Set x_2, x_5, and x_6 to 0 to get $z = \frac{111}{4}, x_1 = \frac{33}{4}, x_3 = \frac{3}{2}$, and $x_4 = \frac{69}{4}$.
- Select entering variable x_2; maximize it to get $x_2 = 4$.
- The leaving variable is x_3:

$$x_2 = 4 - \frac{8x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3}.$$

- Substitute x_2's left-hand side into the rest of the equations to obtain:

$$\begin{aligned}
z &= 28 - \frac{x_3}{6} - \frac{x_5}{6} - \frac{2x_6}{3} \\
x_1 &= 8 + \frac{x_3}{6} + \frac{x_5}{6} - \frac{x_6}{3} \\
x_2 &= 4 - \frac{8x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3} \\
x_4 &= 18 - \frac{x_3}{2} + \frac{x_5}{2}
\end{aligned}$$

Simplex Algorithm Example Continued

- Now all the coefficients are negative, meaning that there is no more room for improvement.
- The algorithm terminates with $z = 28$ by setting x_3, x_5, x_6 to 0.
- The basic solution is

$$\begin{aligned} x_1 &= 8, \;\; x_4 = 18, \\ x_2 &= 4, \;\; x_5 = 0, \\ x_3 &= 0, \;\; x_6 = 0. \end{aligned}$$

- Note: When ties are present, choose the variable with the smallest index (Bland's Rule).

Pivoting

- Pivoting takes as input a slack form.
- Let N denote the set of indices of nonbasic variables and B the set of indices of basic variable in the slack form.
- Let the slack form be given by (N, B, A, b, c, v), where v is the objective value.
- Let l denote the index of the leaving variable x_l and e the entering variable x_e.
- Pivoting returns $(\hat{N}, \hat{B}, \hat{A}, \hat{b}, \hat{c}, \hat{v})$ for the new slack form.

```
PIVOT(N, B, A, b, c, v, l, e)
// Compute the coefficients of the equation for new basic variable x_e.
let Â be a new m × n matrix
b̂_e = b_l / a_le
for each j ∈ N − {e}
    â_ej = a_lj / a_le
â_el = 1 / a_le
```

Pivoting Continued

// Compute the coefficients of the remaining constraints.

for each $i \in B - \{l\}$

    $\hat{b}_i = b_i - a_{ie}\hat{b}_e$

    for each $j \in N - \{e\}$

        $\hat{a}_{ij} = a_{ij} - a_{ie}\hat{a}_{ej}$

    $\hat{a}_{il} = -a_{ie}\hat{a}_{el}$

// Compute the objective function.

$\hat{v} = v + c_e\hat{b}_e$

for each $j \in N - \{e\}$

    $\hat{c}_j = c_j - c_e\hat{a}_{ej}$

$\hat{c}_l = -c_e\hat{a}_{el}$

// Compute new sets of basic and nonbasic variables.

$\hat{N} = N - \{e\} \cup \{l\}$

$\hat{B} = B - \{l\} \cup \{e\}$

return $(\hat{N}, \hat{B}, \hat{A}, \hat{b}, \hat{c}, \hat{v})$

How Will Variables Change in Pivoting?

Lemma 29.1 In a call to $\textsc{Pivot}(N, B, A, b, c, v, l, e)$ with $a_{le} \neq 0$, let $\overline{x}$ denote the basic solution after this call. Then

1. $\overline{x}_j = 0$ for each $j \in \hat{N}$.
2. $\overline{x}_e = b_l / a_{le}$.
3. $\overline{x}_i = b_i - a_{ie}\hat{b}_e$ for each $i \in \hat{B} - \{e\}$.

Proof. The first statement is true because the basic solutions always sets nonbasic variables to 0. For these variables we have $\overline{x}_i = \hat{b}_i$ for $i \in \hat{B}$, since $x_i = \hat{b}_i - \sum_{j \in \hat{N}} \hat{a}_{ij} x_j$.
Since $e \in \hat{B}$, line 3 of $\textsc{Pivot}$ gives

$$\overline{x}_e = \hat{b}_e = b_l / a_{le}.$$

For $i \in \hat{B} - \{e\}$, line 9 gives

$$\overline{x}_i = \hat{b}_i = b_i - a_{ie}\hat{b}_e.$$

This completes the proof.

Simplex Code

```
SIMPLEX(A, b, c)
(N, B, A, b, c, v) = INITIALIZE-SIMPLEX(A, b, c)
let Δ be a new vector of length m
while some index j ∈ N has c_j > 0
    choose an index e ∈ N for which c_e > 0
    for each index i ∈ B
        if a_ie > 0
            Δ_i = b_i / a_ie
        else Δ_i = ∞
    choose an index l ∈ B that minimizes Δ_l
    if Δ_l == ∞
        return "unbounded"
    else (N, B, A, b, c, v) = PIVOT(N, B, A, b, c, v, l, e)
for i = 1 to n
    if i ∈ B
        x̄_i = b_i
    else x̄_i = 0
return (x̄_1, x̄_2, ..., x̄_n)
```

Degeneracy

- Each iteration of the simplex algorithm either increases the objective value or leaves the objective value unchanged. The latter phenomenon is **degeneracy**.
- Example of degeneracy:

$$\begin{array}{rclcccccc} z & = & & & x_1 & + & x_2 & + & x_3 \\ x_4 & = & 8 & - & x_1 & - & x_2 & & \\ x_5 & = & & & & & x_2 & - & x_3 . \end{array}$$

- Suppose that x_1 is chosen to be the entering variable and x_4 the leaving variable. After pivoting:

$$\begin{array}{rclcccccc} z & = & 8 & & & + & x_3 & - & x_4 \\ x_1 & = & 8 & - & x_2 & & & - & x_4 \\ x_5 & = & & & x_2 & - & x_3 & & . \end{array}$$

Degeneracy Continued

- Now x_3 is the only option for being the entering variable and x_5 leaving. After pivoting:

$$\begin{array}{rcrcrcrcr} z & = & 8 & + & x_2 & - & x_4 & - & x_5 \\ x_1 & = & 8 & - & x_2 & - & x_4 & & \\ x_3 & = & & & x_2 & & & - & x_5. \end{array}$$

- The objective value of 8 has not changed, and degeneracy occurs.
- Fortunately, x_2 can now be chosen to be entering and x_1 leaving. After pivoting:

$$\begin{array}{rcrcrcrcrcr} z & = & 16 & - & x_1 & & & - & 2x_4 & - & x_5 \\ x_2 & = & 8 & - & x_1 & & & - & x_4 & & \\ x_3 & = & & & & & x_2 & & & - & x_5. \end{array}$$

Cycling

- Degeneracy may lead to **cycling**: The slack forms at two different iterations are identical.
- If we instead choose x_2 entering and x_3 leaving, then after pivoting:

$$
\begin{array}{rcrcrcrcrcr}
z & = & 8 & & & + & x_3 & - & x_4 & & \\
x_1 & = & 8 & - & x_2 & & & - & x_4 & & \\
x_2 & = & & & & & x_3 & & & + & x_5 .
\end{array}
$$

- This is a cycling.

Simplex Runtime

- Cycling is the only reason that the simplex algorithm might not terminate.
- By choosing the variable with the smallest index, the simplex algorithm always terminate.
- The simplex algorithm terminates in at most $\binom{n+m}{m}$ iterations. Proof. This is because $|B| = m$ and there are $n + m$ variables, implying that this are at most $\binom{n+m}{m}$ ways to choose B.
- In practice, the simplex algorithm runs extremely fast.

Duality

The **dual** of a maximization LP problem (a.k.a. **primal**) is a minimization LP, and vice versa.

Primal:

$$
\begin{array}{llcll}
\text{maximize} & \sum_{j=1}^{n} c_j x_j & & & \\
\text{subject to} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \text{for } i = 1, 2, \ldots, m \\
 & x_j & \geq & 0 & \text{for } j = 1, 2, \ldots, n.
\end{array}
$$

Dual:

$$
\begin{array}{llcll}
\text{maximize} & \sum_{i=1}^{m} b_i y_i & & & \\
\text{subject to} & \sum_{i=1}^{m} a_{ij} y_i & \geq & c_i & \text{for } j = 1, 2, \ldots, n \\
 & y_i & \geq & 0 & \text{for } i = 1, 2, \ldots, m.
\end{array}
$$

Primal-Dual Example

Primal:

$$
\begin{array}{lrcrcrcl}
\text{maximize} & 3x_1 & + & x_2 & + & 2x_3 & & \\
\text{subject to} & x_1 & + & x_2 & + & 3x_3 & \leq & 30 \\
 & 2x_1 & + & 2x_2 & + & 5x_3 & \leq & 24 \\
 & 4x_1 & + & x_2 & + & 2x_3 & \leq & 36 \\
 & x_1, x_2, x_3 & \geq & 0. & & & &
\end{array}
$$

Dual:

$$
\begin{array}{lrcrcrcl}
\text{minimize} & 30y_1 & + & 24y_2 & + & 36y_3 & & \\
\text{subject to} & y_1 & + & 2y_2 & + & 4y_3 & \geq & 3 \\
 & y_1 & + & 2y_2 & + & y_3 & \geq & 1 \\
 & 3y_1 & + & 5y_2 & + & 2y_3 & \geq & 2 \\
 & y_1, y_2, y_3 & \geq & 0. & & & &
\end{array}
$$

Weak LP Lemma

Lemma 29.8 Let $\overline{x}$ be any feasible solution to the primal LP and $\overline{y}$ be any feasible solution to the dual LP. Then

$$\sum_{j=1}^{n} c_j \overline{x}_j \leq \sum_{i=1}^{m} b_i \overline{y}_j.$$

Proof. We have

$$\begin{aligned}
\sum_{j=1}^{n} c_j \overline{x}_j &\leq \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} \overline{y}_i \right) \overline{x}_j && \text{(Definition of } c_j \text{ from dual.)} \\
&= \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} \overline{x}_j \right) \overline{y}_i && \text{(Definition of } b_i \text{ from primal.)} \\
&\leq \sum_{i=1}^{m} b_i \overline{y}_i.
\end{aligned}$$

Equality of Primal and Dual

Corollary 29.9. Let $\overline{x}$ be a feasible solution to a primal linear program $(\mathbf{A}, \mathbf{b}, \mathbf{c})$, and let $\overline{y}$ be a feasible solution to the corresponding dual linear program. If

$$\sum_{j=1}^{n} c_j \overline{x}_j = \sum_{i=1}^{m} b_i \overline{y}_i,$$

then $\overline{x}$ and $\overline{y}$ are optimal solutions to the primal LP and dual LP, respectively.

Proof. It follows from the Weak LP Lemma.

Theorem 29.10. Suppose that the simplex algorithm returns values $\bar{x} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ for the primal LP $(\mathbf{A}, \mathbf{b}, \mathbf{c})$. Let N and B denotes the nonbasic and basic variables for the final slack form, let $\mathbf{c}'$ denote the coefficients in the final slack form, and let $\bar{y} = (\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_m)$ be defined as

$$\bar{y}_i = \begin{cases} -c'_{n+i} & \text{if } (n+i) \in N, \\ 0 & \text{otherwise.} \end{cases}$$

Then $\bar{x}$ is an optimal solution to the primal LP, $\bar{y}$ is an optimal solution to the dual LP, and

$$\sum_{j=1}^{n} c_j \bar{x}_j = \sum_{i=1}^{m} b_i \bar{y}_i.$$

Proof

- It suffices to show that

$$\sum_{j=1}^{n} c_j \overline{x}_j = \sum_{i=1}^{m} b_i \overline{y}_i .$$

- Suppose the simplex algorithm terminates with a final slack form and the objective function

$$z = v' + \sum_{j \in N} c'_j x_j ,$$

  where $c'_j \leq 0$ for all $j \in N$.
- For the basic variables in the final slack form we can set their coefficients to 0. Namely, let $c'_j = 0$ for all $j \in B$.

Proof Continued

- We have

$$
\begin{aligned}
z &= v' + \textstyle\sum_{j \in N} c'_j x_j \\
&= v' + \textstyle\sum_{j \in N} c'_j x_j + \sum_{j \in B} c'_j x_j \quad \text{(since } c'_j = 0 \text{ for } j \in B) \\
&= v' + \textstyle\sum_{j=1}^{n+m} c'_j x_j \quad \text{(since } N \cup B = \{1, \ldots, n+m\}).
\end{aligned}
$$

- Since for $j \in N$ we have $\overline{x}_j = 0$, we have $z = v'$.
- Since each step in the simplex algorithm is an elementary operation, we know that the original objective function on $\overline{x}$ must be the same to that of the final slack form. Namely,

$$
\begin{aligned}
\sum_{j=1}^{n} c_j \overline{x}_j &= v' + \sum_{j=1}^{n+m} c'_j \overline{x}_j \\
&= v' + \sum_{j \in N} c'_j \overline{x}_j + \sum_{j \in B} c'_j \overline{x}_j \\
&= v'.
\end{aligned}
$$

Proof Continued

- We now show that $\overline{y}$ is a feasible solution to the dual LP and

$$\sum_{i=1}^{m} b_i \overline{y}_i = \sum_{j=1}^{n} c_j \overline{x}_j.$$

$$\begin{aligned}
\sum_{j=1}^{n} c_j \overline{x}_j &= v' + \sum_{j=1}^{n+m} c'_j \overline{x}_j \\
&= v' + \sum_{j=1}^{n} c'_j \overline{x}_j + \sum_{j=n+1}^{n+m} c'_j \overline{x}_j \\
&= v' + \sum_{j=1}^{n} c'_j \overline{x}_j + \sum_{i=1}^{m} c'_{n+i} \overline{x}_{n+i} \\
&= v' + \sum_{j=1}^{n} c'_j \overline{x}_j + \sum_{i=1}^{m} (-\overline{y}_i) \overline{x}_{n+i} \qquad \text{(by definition of } \overline{y}_i\text{)}
\end{aligned}$$

Proof Continued

$$= v' + \sum_{j=1}^{n} c'_j \overline{x}_j + \sum_{i=1}^{m} (-\overline{y}_i)\left(b_i - \sum_{j=1}^{n} a_{ij}\overline{x}_j\right) \qquad \text{(by slack form)}$$

$$= v' + \sum_{j=1}^{n} c'_j \overline{x}_j - \sum_{i=1}^{m} b_i \overline{y}_i + \sum_{i=1}^{m}\sum_{j=1}^{n} (a_{ij}\overline{x}_j)\overline{y}_i$$

$$= v' + \sum_{j=1}^{n} c'_j \overline{x}_j - \sum_{i=1}^{m} b_i \overline{y}_i + \sum_{j=1}^{n}\sum_{i=1}^{m} (a_{ij}\overline{y}_i)\overline{x}_j$$

$$= \left(v' - \sum_{i=1}^{m} b_i \overline{y}_i\right) + \sum_{j=1}^{n}\left(c'_j + \sum_{i=1}^{m} a_{ij}\overline{y}_i\right)\overline{x}_j$$

Proof Continued

- This implies that

$$
\begin{aligned}
v' - \sum_{i=1}^{m} b_i \overline{y}_i &= 0, \\
c'_j + \sum_{i=1}^{m} a_{ij} \overline{y}_i &= c_j \qquad\qquad \text{for } j = 1, 2, \ldots, n.
\end{aligned}
$$

- Thus, $\sum_{i=1}^{m} b_i \overline{y}_i = v' = z$.
- Now we show that $\overline{y}$ is feasible. Note that $c'_j \le 0$, we have

$$
\begin{aligned}
c_j &= c'_j + \sum_{i=1}^{m} a_{ij} \overline{y}_i \\
&\le \sum_{i=1}^{m} a_{ij} \overline{y}_i .
\end{aligned}
$$

This completes the proof.

Revisit: LP Formulation for Maximum Flow

- We have seen an LP formulation for maximum flow, where the objective function $\sum_{(s,u)} f_{su}$ has a number of variables f_{su} for $(s,u) \in E$.
- In the formulation, variable f_{uv} is the value of the flow from u to v for $(u,v) \in E$.
- Recall that the flow network does not have an edge from t to s (the residual network may change this, but we don't deal with residual networks in the LP formulation).
- To make an LP dual simpler, we reformulate LP for maximum flow with one variable in the objective function. To do so, we create an edge (t,s) to E with capacity $c(t,s) = +\infty$.

LP Formulation for Maximum Flow

- The following LP models the maximum flow problem (Note: we use indexing i and j instead of u and v to place ourselves in our comfort zone):

$$
\begin{array}{lcl}
\text{Maximize} & f_{ts} & \text{(only one variable)} \\
\text{Subject to} & \forall (i,j) \in E - \{(t,s)\} : f_{ij} \leq c(i,j) & \text{(capacity constraint)} \\
& \forall i \in V : \sum_{(i,j)\in E} f_{ij} - \sum_{(j,i)\in E} f_{ji} = 0 & \text{(flow conservation)} \\
& \forall (i,j) \in E : f_{ij} \geq 0 &
\end{array}
$$

- Note that there is no need to include constraint $f_{ts} \leq c(t,s)$.

Formulation of the Dual

- Each primal constraint corresponds to a dual variable.
  - Let y_{ij} be the variable for constraint

$$f_{ij} \leq c(i,j), \text{ for } (i,j) \in E - \{(t,s)\}.$$

  - Let y_i be the variable for constraint

$$\sum_{(i,j)\in E} f_{ij} - \sum_{(j,i)\in E} f_{ji} = 0 \text{ for } i \in V.$$

- For the objective function we look at the constraints in the primal with nonzero right-hand side and identify dual variables for these constraints. The following is the objective of the dual:

$$\text{Minimize} \sum_{(i,j)\in E-\{(t,s)\}} c(i,j)y_{ij}.$$

Formulation of the Dual Continued

- Each primal variable x must correspond to a dual constraint. To form a dual constraint, follow the steps below:
  1. Identify all the primal constraints containing x.
  2. For each primal constraint containing x, let a be the coefficient of x and y be the dual variable for this constraint.
  3. Sum up all ay's to become the left-hand side of the dual constraint for x.
  4. The value of the right-hand side of the dual constraint is determined by the coefficient of x.

Formulation of the Dual Continued

- For the primal variable f_{ij} with $i \neq t$ and $j \neq s$, we observe that (1) the coefficient in the primal objective is 0; and (2) f_{ij} appears only in the following three primal constraints:

$$\begin{aligned} f_{ij} &\leq c(i,j) && (y_{ij}) \\ \sum_{j:(i,j)\in E} f_{ij} - \sum_{j:(i,j)\in E} f_{ji} &= 0 && (y_i) \\ \sum_{i:(j,i)\in E} f_{ji} - \sum_{i:(i,j)\in E} f_{ij} &= 0 && (y_j) \end{aligned}$$

- Thus, for the primal variable f_{ij} with $i \neq t$ and $j \neq s$, we have the following dual constraint:

$$y_{ij} + y_i - y_j \geq 0.$$

The "$\geq$" sign is used because the variable appears in one constraint with the "$\leq$" sign and the rest with the "$=$" sign.

Formulation of the Dual Continued

- For the primal variable f_{ts}, we note that (1) its coefficient in the primal objective is 1 and (2) it appears in the following two constraints:

$$\begin{aligned} \sum_{j:(t,j)\in E} f_{tj} - \sum_{j:(j,t)\in E} f_{jt} &= 0 \quad (y_t) \\ \sum_{j:(s,j)\in E} f_{sj} - \sum_{j:(j,s)\in E} f_{js} &= 0. \quad (y_s) \end{aligned}$$

Thus, the dual constraint w.r.t. variable f_{ts} is

$$y_t - y_s = 1.$$

The "=" sign is used because the primal constraints involving variable f_{ts} are equations.

The Dual and the Minimum Cut

- Finally, the following is the dual for the maximum-flow LP:

$$
\begin{array}{lll}
\text{Minimize} & \sum_{(i,j)\in E-\{(t,s)\}} c(i,j)y_{ij} & \\
\text{Subject to} & \forall (i,j) \in E - \{(t,s)\} : y_{ij} + y_i - y_j & \geq 0 \\
 & y_t - y_s & = 1 \\
 & y_{ij}, y_i & \geq 0
\end{array}
$$

- We know that the maximum flow is equal to the summation of the capacity of minimum cut, and any feasible solution to the primal is bounded above by a feasible solution to the dual. Thus, the optimal solution to the dual is the minimum cut.

Exercise

- As an exercise let's consider the previous dual as primal and obtain its dual.
- Let x_{ij} be the dual variable for the following primal constraint:

$$y_{ij} + y_i - y_j \geq 0 \text{ for } (i, j) \in E - \{(t, s)\}$$

- Let x_{ts} denote the dual variable for the following primal constraint:

$$y_t - y_s = 1.$$

- To form the dual objective, we note that only primal constraint $y_t - y_s = 1$ has nonzero right-hand side. Thus, the objective for the dual is

$$\text{Maximize } x_{ts}.$$

- To form the dual constraint, we note that each primal varialbe y_{ij} with $(i,j) \neq (t,s)$ appears in the following constraints:

$$y_{ij} + y_i - y_j \leq 0$$

and it has coeficient $c(i,j)$ in the primal objective function. Thus, for this variable we have a dual constraint

$$x_{ij} \leq c(i,j) \text{ for } (i,j) \neq (t,s).$$

- Each primal variable y_i has coeffieint 0 in the primal objective function. For $i \notin \{s,t\}$, y_i appears in the following constraints:

$$\begin{aligned} y_{ij} + y_i - y_j &\geq 0 && \text{for } (i,j) \in E - \{(t,s)\} \\ y_{ji} + y_j - y_i &\geq 0 && \text{for } (j,i) \in E - \{(t,s)\} \end{aligned}$$

Exercise Continued

- Thus, the dual constraint for the primal variable y_i with $i \notin \{s, t\}$ is

$$\sum_{j:(i,j)\in E-\{(t,s)\}} x_{ij} - \sum_{j:(j,i)\in E-\{(t,s)\}} x_{ji} \leq 0.$$

- The primal variable y_s appears in the following constraints:

$$\begin{aligned} y_{sj} + y_s - y_j &\geq 0 && \text{for } (s,j) \in E \\ y_{ts} + y_t - y_s &\geq 0. \end{aligned}$$

Thus, the dual constraint for y_s is

$$\sum_{j:(s,j)\in E} x_{sj} - x_{ts} \leq 0.$$

- Likewise, the dual constraint for y_t is

$$x_{ts} - \sum_{j:(j,t)\in E} x_{jt} \leq 0.$$

Exercise Continued

- View x_{ij} as flow from node i to node j. Because for each $i \in V$, the total flow entering node i is at least the total flow going out of i and there is an edge from t to s, all the inequaliies in the dual constrains must be equalities, namely, for each $i \in V$,

$$\sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = 0.$$

- Finally, the dual problem is

$$\begin{array}{rl} \text{Maximize} & x_{ts} \\ \text{Subject to} & \forall (i,j) \in E - \{(t,s)\} : x_{ij} \leq c(i,j) \\ & \forall i \in V : \sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} \quad = 0 \\ & \forall (i,j) \in E : x_{ij} \quad \geq 0. \end{array}$$

This is exactly the same as the maximum flow LP formulation.

Sample Final Quesions

Unlike problems in quizes, the final exam consists of only multiple-choice (including true/false) quesions. Example:

1. Which is the correct dual for the following primal LP:

$$\begin{array}{lll} \text{Minimize} & -3x_1 + 8x_2 + 4x_3 & \\ \text{Subject to} & x_1 + 5x_2 - 3x_4 & \geq 7 \\ & -3x_1 + 10x_3 + 5x_4 & \geq 1 \\ & x_1, x_2, x_3, x_4 & \geq 0. \end{array}$$

(a)

$$\begin{array}{lll} \text{Maximize} & 7y_1 + y_2 & \\ \text{Subject to} & y_1 - 3y_2 & \leq -3 \\ & 5y_1 & \leq 8 \\ & 10y_2 & \leq 4 \\ & -3y_1 + 5y_2 & \leq 0 \\ & y_1, y_2, y_3, y_4 & \geq 0. \end{array}$$

(b)

$$\begin{aligned}
\text{Maximize} \quad & 7y_1 + y_2 \\
\text{Subject to} \quad & y_1 - 3y_2 && \leq -3 \\
& -3y_1 + 5y_2 && \leq 0 \\
& y_1, y_2, y_3, y_4 && \geq 0.
\end{aligned}$$

(c)

$$\begin{aligned}
\text{Maximize} \quad & 3y_1 + 8y_2 - 4y_3 \\
\text{Subject to} \quad & y_1 - 3y_2 && \leq -3 \\
& 5y_1 && \leq 8 \\
& 10y_2 && \leq 4 \\
& -3y_1 + 5y_2 && \leq 0 \\
& y_1, y_2, y_3, y_4 && \geq 0.
\end{aligned}$$

(d) None of the above.

Announcement

- The final exam will also contain one bonus problem for the purpose of replacing one of your quiz scores of your choice.