CS 573: Algorithms, Fall 2014

Linear Programming II

Lecture 17

October 23, 2014

Simplex algorithm

Simplex($\hat{L}$ a LP)
  Transform $\hat{L}$ into slack form.
  Let L be the resulting slack form.
  $L' \leftarrow$ **Feasible**(L)
  $x \leftarrow$ **LPStartSolution**(L')
  $x' \leftarrow$ **SimplexInner**(L', x) (*)
  $z \leftarrow$ objective function value of x'
  if $z > 0$ **then**
    return "No solution"
  $x'' \leftarrow$ **SimplexInner**(L, x')
  return x''

Simplex algorithm...

- **SimplexInner**: solves a LP if the trivial solution of assigning zero to all the nonbasic variables is feasible.
- $L' =$ **Feasible**(L) returns a new LP with feasible solution.
- Done by adding new variable x_0 to each equality.
- Set target function in L' to $\min x_0$.
- original LP L feasible $\iff$ LP L' has feasible solution with $x_0 = 0$.
- Apply **SimplexInner** to L' and solution computed (for L') by **LPStartSolution**(L').
- If $x_0 = 0$ then have a feasible solution to L.
- Use solution in **SimplexInner** on L.
- need to describe **SimplexInner**: solve LP in slack form given a feasible solution (all nonbasic vars assigned value 0).

Simplex algorithm...

1. **SimplexInner**: solves a LP if the trivial solution of assigning zero to all the nonbasic variables is feasible.
2. $L' = \textbf{Feasible}(L)$ returns a new LP with feasible solution.
3. Done by adding new variable x_0 to each equality.
4. Set target function in L' to $\min x_0$.
5. original LP L feasible $\iff$ LP L' has feasible solution with $x_0 = 0$.
6. Apply **SimplexInner** to L' and solution computed (for L') by **LPStartSolution**(L').
7. If $x_0 = 0$ then have a feasible solution to L.
8. Use solution in **SimplexInner** on L.
9. need to describe **SimplexInner**: solve LP in slack form given a feasible solution (all nonbasic vars assigned value 0).

Simplex algorithm...

- **SimplexInner**: solves a LP if the trivial solution of assigning zero to all the nonbasic variables is feasible.
- $L' =$ **Feasible**(L) returns a new LP with feasible solution.
- Done by adding new variable x_0 to each equality.
- Set target function in L' to $\min x_0$.
- original LP L feasible $\iff$ LP L' has feasible solution with $x_0 = 0$.
- Apply **SimplexInner** to L' and solution computed (for L') by **LPStartSolution**(L').
- If $x_0 = 0$ then have a feasible solution to L.
- Use solution in **SimplexInner** on L.
- need to describe **SimplexInner**: solve LP in slack form given a feasible solution (all nonbasic vars assigned value 0).

Simplex algorithm...

1. **SimplexInner**: solves a LP if the trivial solution of assigning zero to all the nonbasic variables is feasible.
2. $L' = \textbf{Feasible}(L)$ returns a new LP with feasible solution.
3. Done by adding new variable x_0 to each equality.
4. Set target function in L' to $\min x_0$.
5. original LP L feasible $\iff$ LP L' has feasible solution with $x_0 = 0$.
6. Apply **SimplexInner** to L' and solution computed (for L') by **LPStartSolution**(L').
7. If $x_0 = 0$ then have a feasible solution to L.
8. Use solution in **SimplexInner** on L.
9. need to describe **SimplexInner**: solve LP in slack form given a feasible solution (all nonbasic vars assigned value 0).

Simplex algorithm...

1. **SimplexInner**: solves a LP if the trivial solution of assigning zero to all the nonbasic variables is feasible.
2. $L' = \textbf{Feasible}(L)$ returns a new LP with feasible solution.
3. Done by adding new variable x_0 to each equality.
4. Set target function in L' to $\min x_0$.
5. original LP L feasible $\iff$ LP L' has feasible solution with $x_0 = 0$.
6. Apply **SimplexInner** to L' and solution computed (for L') by **LPStartSolution**(L').
7. If $x_0 = 0$ then have a feasible solution to L.
8. Use solution in **SimplexInner** on L.
9. need to describe **SimplexInner**: solve LP in slack form given a feasible solution (all nonbasic vars assigned value 0).

Simplex algorithm...

1. **SimplexInner**: solves a LP if the trivial solution of assigning zero to all the nonbasic variables is feasible.
2. $L' = \textbf{Feasible}(L)$ returns a new LP with feasible solution.
3. Done by adding new variable x_0 to each equality.
4. Set target function in L' to $\min x_0$.
5. original LP L feasible $\iff$ LP L' has feasible solution with $x_0 = 0$.
6. Apply **SimplexInner** to L' and solution computed (for L') by **LPStartSolution**(L').
7. If $x_0 = 0$ then have a feasible solution to L.
8. Use solution in **SimplexInner** on L.
9. need to describe **SimplexInner**: solve LP in slack form given a feasible solution (all nonbasic vars assigned value 0).

Simplex algorithm...

1. **SimplexInner**: solves a LP if the trivial solution of assigning zero to all the nonbasic variables is feasible.
2. $L' = \textbf{Feasible}(L)$ returns a new LP with feasible solution.
3. Done by adding new variable x_0 to each equality.
4. Set target function in L' to $\min x_0$.
5. original LP L feasible $\iff$ LP L' has feasible solution with $x_0 = 0$.
6. Apply **SimplexInner** to L' and solution computed (for L') by **LPStartSolution**(L').
7. If $x_0 = 0$ then have a feasible solution to L.
8. Use solution in **SimplexInner** on L.
9. need to describe **SimplexInner**: solve LP in slack form given a feasible solution (all nonbasic vars assigned value 0).

Notations

B - Set of indices of basic variables
N - Set of indices of nonbasic variables
$n = |N|$ - number of original variables
b, c - two vectors of constants
$m = |B|$ - number of basic variables (i.e., number of inequalities)
$A = \{a_{ij}\}$ - The matrix of coefficients
$N \cup B = \{1, \ldots, n + m\}$
v - objective function constant.
LP in slack form is specified by a tuple (N, B, A, b, c, v).

The corresponding LP

$$\begin{aligned} \max \quad & z = v + \sum_{j \in N} c_j x_j, \\ \text{s.t.} \quad & x_i = b_i - \sum_{j \in N} a_{ij} x_j \text{ for } i \in B, \\ & x_i \geq 0, \qquad \forall i = 1, \ldots, n+m. \end{aligned}$$

Reminder - basic/nonbasic

Nonbasic variables

$$
\begin{aligned}
\max \quad z &= 29 - \frac{1}{9}x_3 - \frac{1}{9}x_5 - \frac{2}{9}x_6 \\
x_1 &= 8 + \frac{1}{6}x_3 + \frac{1}{6}x_5 - \frac{1}{3}x_6 \\
x_2 &= 4 - \frac{8}{3}x_3 - \frac{2}{3}x_5 + \frac{1}{3}x_6 \\
x_4 &= 18 - \frac{1}{2}x_3 + \frac{1}{2}x_5
\end{aligned}
$$

Basic variables

The SimplexInner Algorithm

Description **SimplexInner** algorithm:

- LP is in slack form.
- Trivial solution $x = \tau$ (i.e., all nonbasic variables zero), is feasible.
- objective value for this solution is v.
- Reminder: Objective function is $z = v + \sum_{j \in N} c_j x_j$.
- x_e: nonbasic variable with positive coefficient in objective function.
- Formally: e is one of the indices of $\left\{ j \;\middle|\; c_j > 0, j \in N \right\}$.
- x_e is the ***entering variable*** (enters set of basic variables).
- If increase value x_e (from current value of 0 in τ)...
- ... one of basic variables is going to vanish (i.e., become zero).

The SimplexInner Algorithm

Description **SimplexInner** algorithm:

1. LP is in slack form.
2. Trivial solution $x = \tau$ (i.e., all nonbasic variables zero), is feasible.
3. objective value for this solution is v.
4. Reminder: Objective function is $z = v + \sum_{j \in N} c_j x_j$.
5. x_e: nonbasic variable with positive coefficient in objective function.
6. Formally: e is one of the indices of $\left\{ j \mid c_j > 0, j \in N \right\}$.
7. x_e is the ***entering variable*** (enters set of basic variables).
8. If increase value x_e (from current value of 0 in τ)...
9. ... one of basic variables is going to vanish (i.e., become zero).

The SimplexInner Algorithm

Description **SimplexInner** algorithm:

1. LP is in slack form.
2. Trivial solution $x = \tau$ (i.e., all nonbasic variables zero), is feasible.
3. objective value for this solution is v.
4. Reminder: Objective function is $z = v + \sum_{j \in N} c_j x_j$.
5. x_e: nonbasic variable with positive coefficient in objective function.
6. Formally: e is one of the indices of $\left\{ j \;\middle|\; c_j > 0, j \in N \right\}$.
7. x_e is the ***entering variable*** (enters set of basic variables).
8. If increase value x_e (from current value of 0 in τ)...
9. ... one of basic variables is going to vanish (i.e., become zero).

The SimplexInner Algorithm

Description **SimplexInner** algorithm:

1. LP is in slack form.
2. Trivial solution $x = \tau$ (i.e., all nonbasic variables zero), is feasible.
3. objective value for this solution is v.
4. Reminder: Objective function is $z = v + \sum_{j \in N} c_j x_j$.
5. x_e: nonbasic variable with positive coefficient in objective function.
6. Formally: e is one of the indices of $\left\{ j \mid c_j > 0, j \in N \right\}$.
7. x_e is the ***entering variable*** (enters set of basic variables).
8. If increase value x_e (from current value of 0 in τ)...
9. ... one of basic variables is going to vanish (i.e., become zero).

The SimplexInner Algorithm

Description **SimplexInner** algorithm:

1. LP is in slack form.
2. Trivial solution $x = \tau$ (i.e., all nonbasic variables zero), is feasible.
3. objective value for this solution is v.
4. Reminder: Objective function is $z = v + \sum_{j \in N} c_j x_j$.
5. x_e: nonbasic variable with positive coefficient in objective function.
6. Formally: e is one of the indices of $\left\{ j \;\middle|\; c_j > 0, j \in N \right\}$.
7. x_e is the ***entering variable*** (enters set of basic variables).
8. If increase value x_e (from current value of 0 in τ)...
9. ... one of basic variables is going to vanish (i.e., become zero).

The SimplexInner Algorithm

Description **SimplexInner** algorithm:

1. LP is in slack form.
2. Trivial solution $x = \tau$ (i.e., all nonbasic variables zero), is feasible.
3. objective value for this solution is v.
4. Reminder: Objective function is $z = v + \sum_{j \in N} c_j x_j$.
5. x_e: nonbasic variable with positive coefficient in objective function.
6. Formally: e is one of the indices of $\left\{ j \,\middle|\, c_j > 0, j \in N \right\}$.
7. x_e is the ***entering variable*** (enters set of basic variables).
8. If increase value x_e (from current value of 0 in τ)...
9. ... one of basic variables is going to vanish (i.e., become zero).

Choosing the leaving variable

1. x_e: ***entering variable***
2. x_l: ***leaving*** variable – vanishing basic variable.
3. increase value of x_e till x_l becomes zero.
4. How do we now which variable is x_l?
5. set all nonbasic to 0 zero, except x_e
6. $x_i = b_i - a_{ie} x_e$, for all $i \in B$.
7. Require: $\forall i \in B \quad x_i = b_i - a_{ie} x_e \geq 0$.
8. $\implies x_e \leq (b_i / a_{ie})$
9. $l = \arg\min_i b_i / a_{ie}$
10. If more than one achieves $\min_i b_i / a_{ie}$, just pick one.

Choosing the leaving variable

1. x_e: ***entering variable***
2. x_l: ***leaving*** variable – vanishing basic variable.
3. increase value of x_e till x_l becomes zero.
4. How do we now which variable is x_l?
5. set all nonbasic to $\mathbf{0}$ zero, except x_e
6. $x_i = b_i - a_{ie}x_e$, for all $i \in B$.
7. Require: $\forall i \in B \quad x_i = b_i - a_{ie}x_e \geq 0$.
8. $\implies \quad x_e \leq (b_i / a_{ie})$
9. $l = \arg\min_i b_i / a_{ie}$
10. If more than one achieves $\min_i b_i / a_{ie}$, just pick one.

Choosing the leaving variable

1. x_e: ***entering variable***
2. x_l: ***leaving*** variable – vanishing basic variable.
3. increase value of x_e till x_l becomes zero.
4. How do we now which variable is x_l?
5. set all nonbasic to 0 zero, except x_e
6. $x_i = b_i - a_{ie}x_e$, for all $i \in B$.
7. Require: $\forall i \in B \quad x_i = b_i - a_{ie}x_e \geq 0$.
8. $\implies x_e \leq (b_i / a_{ie})$
9. $l = \arg\min_i b_i / a_{ie}$
10. If more than one achieves $\min_i b_i / a_{ie}$, just pick one.

Choosing the leaving variable

1. x_e: ***entering variable***
2. x_l: ***leaving*** variable – vanishing basic variable.
3. increase value of x_e till x_l becomes zero.
4. How do we now which variable is x_l?
5. set all nonbasic to 0 zero, except x_e
6. $x_i = b_i - a_{ie} x_e$, for all $i \in B$.
7. Require: $\forall i \in B \quad x_i = b_i - a_{ie} x_e \geq 0$.
8. $\implies \quad x_e \leq (b_i / a_{ie})$
9. $l = \arg\min_i b_i / a_{ie}$
10. If more than one achieves $\min_i b_i / a_{ie}$, just pick one.

Choosing the leaving variable

1. x_e: ***entering variable***
2. x_l: ***leaving*** variable – vanishing basic variable.
3. increase value of x_e till x_l becomes zero.
4. How do we now which variable is x_l?
5. set all nonbasic to 0 zero, except x_e
6. $x_i = b_i - a_{ie}x_e$, for all $i \in B$.
7. Require: $\forall i \in B \quad x_i = b_i - a_{ie}x_e \geq 0$.
8. $\implies x_e \leq (b_i/a_{ie})$
9. $l = \arg\min_i b_i/a_{ie}$
10. If more than one achieves $\min_i b_i/a_{ie}$, just pick one.

Choosing the leaving variable

1. x_e: ***entering variable***
2. x_l: ***leaving*** variable – vanishing basic variable.
3. increase value of x_e till x_l becomes zero.
4. How do we now which variable is x_l?
5. set all nonbasic to 0 zero, except x_e
6. $x_i = b_i - a_{ie}x_e$, for all $i \in B$.
7. Require: $\forall i \in B \quad x_i = b_i - a_{ie}x_e \geq 0$.
8. $\implies x_e \leq (b_i / a_{ie})$
9. $l = \arg\min_i b_i / a_{ie}$
10. If more than one achieves $\min_i b_i / a_{ie}$, just pick one.

Choosing the leaving variable

1. x_e: ***entering variable***
2. x_l: ***leaving*** variable – vanishing basic variable.
3. increase value of x_e till x_l becomes zero.
4. How do we now which variable is x_l?
5. set all nonbasic to 0 zero, except x_e
6. $x_i = b_i - a_{ie}x_e$, for all $i \in B$.
7. Require: $\forall i \in B \quad x_i = b_i - a_{ie}x_e \geq 0$.
8. $\implies x_e \leq (b_i / a_{ie})$
9. $l = \arg\min_i b_i / a_{ie}$
10. If more than one achieves $\min_i b_i / a_{ie}$, just pick one.

Choosing the leaving variable

1. x_e: ***entering variable***
2. x_l: ***leaving*** variable – vanishing basic variable.
3. increase value of x_e till x_l becomes zero.
4. How do we now which variable is x_l?
5. set all nonbasic to 0 zero, except x_e
6. $x_i = b_i - a_{ie} x_e$, for all $i \in B$.
7. Require: $\forall i \in B \quad x_i = b_i - a_{ie} x_e \geq 0$.
8. $\implies x_e \leq (b_i / a_{ie})$
9. $l = \arg\min_i b_i / a_{ie}$
10. If more than one achieves $\min_i b_i / a_{ie}$, just pick one.

Choosing the leaving variable

1. x_e: ***entering variable***
2. x_l: ***leaving*** variable – vanishing basic variable.
3. increase value of x_e till x_l becomes zero.
4. How do we now which variable is x_l?
5. set all nonbasic to 0 zero, except x_e
6. $x_i = b_i - a_{ie}x_e$, for all $i \in B$.
7. Require: $\forall i \in B \quad x_i = b_i - a_{ie}x_e \geq 0$.
8. $\implies x_e \leq (b_i / a_{ie})$
9. $l = \arg\min_i b_i / a_{ie}$
10. If more than one achieves $\min_i b_i / a_{ie}$, just pick one.

Choosing the leaving variable

1. x_e: ***entering variable***
2. x_l: ***leaving*** variable – vanishing basic variable.
3. increase value of x_e till x_l becomes zero.
4. How do we now which variable is x_l?
5. set all nonbasic to 0 zero, except x_e
6. $x_i = b_i - a_{ie} x_e$, for all $i \in B$.
7. Require: $\forall i \in B \quad x_i = b_i - a_{ie} x_e \geq 0$.
8. $\implies x_e \leq (b_i / a_{ie})$
9. $l = \arg\min_i b_i / a_{ie}$
10. If more than one achieves $\min_i b_i / a_{ie}$, just pick one.

Pivoting on $\mathbf{x_e}$...

1. Determined x_e and x_l.
2. Rewrite equation for x_l in LP.
   1. (Every basic variable has an equation in the LP!)
   2. $x_l = b_l - \sum_{j \in N} a_{lj} x_j$

$$\implies \qquad x_e = \frac{b_l}{a_{le}} - \sum_{j \in N \cup \{l\}} \frac{a_{lj}}{a_{le}} x_j, \qquad \text{where } a_{ll} = 1.$$

3. Cleanup: remove all appearances (on right) in LP of x_e.
4. Substituting x_e into the other equalities, using above.
5. Alternatively, do Gaussian elimination remove any appearance of x_e on right side LP (including objective). Transfer x_l on the left side, to the right side.

Pivoting on $\mathbf{x_e}$...

1. Determined x_e and x_l.
2. Rewrite equation for x_l in LP.
   1. (Every basic variable has an equation in the LP!)
   2. $x_l = b_l - \sum_{j \in N} a_{lj} x_j$

$$\implies \quad x_e = \frac{b_l}{a_{le}} - \sum_{j \in N \cup \{l\}} \frac{a_{lj}}{a_{le}} x_j, \qquad \text{where } a_{ll} = 1.$$

3. Cleanup: remove all appearances (on right) in LP of x_e.
4. Substituting x_e into the other equalities, using above.
5. Alternatively, do Gaussian elimination remove any appearance of x_e on right side LP (including objective). Transfer x_l on the left side, to the right side.

Pivoting on $\mathbf{x_e}$...

1. Determined x_e and x_l.
2. Rewrite equation for x_l in LP.
    1. (Every basic variable has an equation in the LP!)
    2. $x_l = b_l - \sum_{j \in N} a_{lj} x_j$

$$\implies \quad x_e = \frac{b_l}{a_{le}} - \sum_{j \in N \cup \{l\}} \frac{a_{lj}}{a_{le}} x_j, \qquad \text{where } a_{ll} = 1.$$

3. Cleanup: remove all appearances (on right) in LP of x_e.
4. Substituting x_e into the other equalities, using above.
5. Alternatively, do Gaussian elimination remove any appearance of x_e on right side LP (including objective). Transfer x_l on the left side, to the right side.

Pivoting on $\mathbf{x_e}$...

1. Determined x_e and x_l.
2. Rewrite equation for x_l in LP.
    1. (Every basic variable has an equation in the LP!)
    2. $x_l = b_l - \sum_{j \in N} a_{lj} x_j$

$$\implies \qquad x_e = \frac{b_l}{a_{le}} - \sum_{j \in N \cup \{l\}} \frac{a_{lj}}{a_{le}} x_j, \qquad \text{where } a_{ll} = 1.$$

3. Cleanup: remove all appearances (on right) in LP of x_e.
4. Substituting x_e into the other equalities, using above.
5. Alternatively, do Gaussian elimination remove any appearance of x_e on right side LP (including objective). Transfer x_l on the left side, to the right side.

Pivoting on $\mathbf{x_e}$...

1. Determined x_e and x_l.
2. Rewrite equation for x_l in LP.
   1. (Every basic variable has an equation in the LP!)
   2. $x_l = b_l - \sum_{j \in N} a_{lj} x_j$

$$\implies \qquad x_e = \frac{b_l}{a_{le}} - \sum_{j \in N \cup \{l\}} \frac{a_{lj}}{a_{le}} x_j, \qquad \text{where } a_{ll} = 1.$$

3. Cleanup: remove all appearances (on right) in LP of x_e.
4. Substituting x_e into the other equalities, using above.
5. Alternatively, do Gaussian elimination remove any appearance of x_e on right side LP (including objective). Transfer x_l on the left side, to the right side.

Pivoting on $\mathbf{x_e}$...

1. Determined x_e and x_l.
2. Rewrite equation for x_l in LP.
   1. (Every basic variable has an equation in the LP!)
   2. $x_l = b_l - \sum_{j \in N} a_{lj} x_j$

$$\Longrightarrow \qquad x_e = \frac{b_l}{a_{le}} - \sum_{j \in N \cup \{l\}} \frac{a_{lj}}{a_{le}} x_j, \qquad \text{where } a_{ll} = 1.$$

3. Cleanup: remove all appearances (on right) in LP of x_e.
4. Substituting x_e into the other equalities, using above.
5. Alternatively, do Gaussian elimination remove any appearance of x_e on right side LP (including objective). Transfer x_l on the left side, to the right side.

Pivoting on $\mathbf{x_e}$...

1. Determined x_e and x_l.
2. Rewrite equation for x_l in LP.
   1. (Every basic variable has an equation in the LP!)
   2. $x_l = b_l - \sum_{j \in N} a_{lj} x_j$

$$\implies \qquad x_e = \frac{b_l}{a_{le}} - \sum_{j \in N \cup \{l\}} \frac{a_{lj}}{a_{le}} x_j, \qquad \text{where } a_{ll} = 1.$$

3. Cleanup: remove all appearances (on right) in LP of x_e.
4. Substituting x_e into the other equalities, using above.
5. Alternatively, do Gaussian elimination remove any appearance of x_e on right side LP (including objective). Transfer x_l on the left side, to the right side.

Pivoting on $\mathbf{x_e}$...

1. Determined x_e and x_l.
2. Rewrite equation for x_l in LP.
   1. (Every basic variable has an equation in the LP!)
   2. $x_l = b_l - \sum_{j \in N} a_{lj} x_j$

$$\implies \qquad x_e = \frac{b_l}{a_{le}} - \sum_{j \in N \cup \{l\}} \frac{a_{lj}}{a_{le}} x_j, \qquad \text{where } a_{ll} = 1.$$

3. Cleanup: remove all appearances (on right) in LP of x_e.
4. Substituting x_e into the other equalities, using above.
5. Alternatively, do Gaussian elimination remove any appearance of x_e on right side LP (including objective). Transfer x_l on the left side, to the right side.

Pivoting continued...

1. End of this process: have new *equivalent* LP.
2. basic variables: $B' = (B \setminus \{l\}) \cup \{e\}$
3. non-basic variables: $N' = (N \setminus \{e\}) \cup \{l\}$.
4. End of this ***pivoting*** stage:
   LP objective function value increased.
5. Made progress.
6. LP is completely defined by which variables are basic, and which are non-basic.
7. Pivoting never returns to a combination (of basic/non-basic variable) already visited.
8. ...because improve objective in each pivoting step.
9. Can do at most $\binom{n+m}{n} \leq \left(\frac{n+m}{n} \cdot e\right)^n$.
10. examples where 2^n pivoting steps are needed.

Pivoting continued...

1. End of this process: have new *equivalent* LP.
2. basic variables: $B' = (B \setminus \{l\}) \cup \{e\}$
3. non-basic variables: $N' = (N \setminus \{e\}) \cup \{l\}$.
4. End of this ***pivoting*** stage:
   LP objective function value increased.
5. Made progress.
6. LP is completely defined by which variables are basic, and which are non-basic.
7. Pivoting never returns to a combination (of basic/non-basic variable) already visited.
8. ...because improve objective in each pivoting step.
9. Can do at most $\binom{n+m}{n} \leq \left(\frac{n+m}{n} \cdot e\right)^n$.
10. examples where 2^n pivoting steps are needed.

Pivoting continued...

1. End of this process: have new *equivalent* LP.
2. basic variables: $B' = (B \setminus \{l\}) \cup \{e\}$
3. non-basic variables: $N' = (N \setminus \{e\}) \cup \{l\}$.
4. End of this ***pivoting*** stage:
   LP objective function value increased.
5. Made progress.
6. LP is completely defined by which variables are basic, and which are non-basic.
7. Pivoting never returns to a combination (of basic/non-basic variable) already visited.
8. ...because improve objective in each pivoting step.
9. Can do at most $\binom{n+m}{n} \leq \left(\frac{n+m}{n} \cdot e\right)^n$.
10. examples where 2^n pivoting steps are needed.

Pivoting continued...

1. End of this process: have new *equivalent* LP.
2. basic variables: $B' = (B \setminus \{l\}) \cup \{e\}$
3. non-basic variables: $N' = (N \setminus \{e\}) \cup \{l\}$.
4. End of this ***pivoting*** stage:
   LP objective function value increased.
5. Made progress.
6. LP is completely defined by which variables are basic, and which are non-basic.
7. Pivoting never returns to a combination (of basic/non-basic variable) already visited.
8. ...because improve objective in each pivoting step.
9. Can do at most $\binom{n+m}{n} \leq \left(\frac{n+m}{n} \cdot e\right)^n$.
10. examples where 2^n pivoting steps are needed.

Pivoting continued...

1. End of this process: have new *equivalent* LP.
2. basic variables: $B' = (B \setminus \{l\}) \cup \{e\}$
3. non-basic variables: $N' = (N \setminus \{e\}) \cup \{l\}$.
4. End of this ***pivoting*** stage:
   LP objective function value increased.
5. Made progress.
6. LP is completely defined by which variables are basic, and which are non-basic.
7. Pivoting never returns to a combination (of basic/non-basic variable) already visited.
8. ...because improve objective in each pivoting step.
9. Can do at most $\binom{n+m}{n} \leq \left(\frac{n+m}{n} \cdot e\right)^n$.
10. examples where 2^n pivoting steps are needed.

Pivoting continued...

1. End of this process: have new *equivalent* LP.
2. basic variables: $B' = (B \setminus \{l\}) \cup \{e\}$
3. non-basic variables: $N' = (N \setminus \{e\}) \cup \{l\}$.
4. End of this ***pivoting*** stage:
   LP objective function value increased.
5. Made progress.
6. LP is completely defined by which variables are basic, and which are non-basic.
7. Pivoting never returns to a combination (of basic/non-basic variable) already visited.
8. ...because improve objective in each pivoting step.
9. Can do at most $\binom{n+m}{n} \leq \left(\frac{n+m}{n} \cdot e\right)^n$.
10. examples where 2^n pivoting steps are needed.

Pivoting continued...

1. End of this process: have new *equivalent* LP.
2. basic variables: $B' = (B \setminus \{l\}) \cup \{e\}$
3. non-basic variables: $N' = (N \setminus \{e\}) \cup \{l\}$.
4. End of this ***pivoting*** stage:
   LP objective function value increased.
5. Made progress.
6. LP is completely defined by which variables are basic, and which are non-basic.
7. Pivoting never returns to a combination (of basic/non-basic variable) already visited.
8. ...because improve objective in each pivoting step.
9. Can do at most $\binom{n+m}{n} \leq \left(\frac{n+m}{n} \cdot e\right)^n$.
10. examples where 2^n pivoting steps are needed.

Pivoting continued...

1. End of this process: have new *equivalent* LP.
2. basic variables: $B' = (B \setminus \{l\}) \cup \{e\}$
3. non-basic variables: $N' = (N \setminus \{e\}) \cup \{l\}$.
4. End of this ***pivoting*** stage:
   LP objective function value increased.
5. Made progress.
6. LP is completely defined by which variables are basic, and which are non-basic.
7. Pivoting never returns to a combination (of basic/non-basic variable) already visited.
8. ...because improve objective in each pivoting step.
9. Can do at most $\binom{n+m}{n} \le \left(\frac{n+m}{n} \cdot e\right)^n$.
10. examples where 2^n pivoting steps are needed.

Pivoting continued...

1. End of this process: have new *equivalent* LP.
2. basic variables: $B' = (B \setminus \{l\}) \cup \{e\}$
3. non-basic variables: $N' = (N \setminus \{e\}) \cup \{l\}$.
4. End of this ***pivoting*** stage:
   LP objective function value increased.
5. Made progress.
6. LP is completely defined by which variables are basic, and which are non-basic.
7. Pivoting never returns to a combination (of basic/non-basic variable) already visited.
8. ...because improve objective in each pivoting step.
9. Can do at most $\binom{n+m}{n} \leq \left(\frac{n+m}{n} \cdot e\right)^n$.
10. examples where 2^n pivoting steps are needed.

Pivoting continued...

1. End of this process: have new *equivalent* LP.
2. basic variables: $B' = (B \setminus \{l\}) \cup \{e\}$
3. non-basic variables: $N' = (N \setminus \{e\}) \cup \{l\}$.
4. End of this ***pivoting*** stage:
   LP objective function value increased.
5. Made progress.
6. LP is completely defined by which variables are basic, and which are non-basic.
7. Pivoting never returns to a combination (of basic/non-basic variable) already visited.
8. ...because improve objective in each pivoting step.
9. Can do at most $\binom{n+m}{n} \leq \left(\frac{n+m}{n} \cdot e\right)^n$.
10. examples where 2^n pivoting steps are needed.

Simplex algorithm summary...

1. Each pivoting step takes polynomial time in n and m.
2. Running time of **Simplex** is exponential in the worst case.
3. In practice, **Simplex** is extremely fast.

Simplex algorithm summary...

1. Each pivoting step takes polynomial time in n and m.
2. Running time of **Simplex** is exponential in the worst case.
3. In practice, **Simplex** is extremely fast.

Simplex algorithm summary...

1. Each pivoting step takes polynomial time in n and m.
2. Running time of **Simplex** is exponential in the worst case.
3. In practice, **Simplex** is extremely fast.

Degeneracies

1. **Simplex** might get stuck if one of the b_is is zero.
2. More than $> m$ hyperplanes (i.e., equalities) passes through the same point.
3. Result: might not be able to make any progress at all in a pivoting step.
4. Solution I: add tiny random noise to each coefficient. Can be done symbolically.
   Intuitively, the degeneracy, being a local phenomena on the polytope disappears with high probability.

Degeneracies

1. **Simplex** might get stuck if one of the b_is is zero.
2. More than $> m$ hyperplanes (i.e., equalities) passes through the same point.
3. Result: might not be able to make any progress at all in a pivoting step.
4. Solution I: add tiny random noise to each coefficient. Can be done symbolically.
   Intuitively, the degeneracy, being a local phenomena on the polytope disappears with high probability.

Degeneracies

1. **Simplex** might get stuck if one of the b_is is zero.
2. More than $> m$ hyperplanes (i.e., equalities) passes through the same point.
3. Result: might not be able to make any progress at all in a pivoting step.
4. Solution I: add tiny random noise to each coefficient. Can be done symbolically.
   Intuitively, the degeneracy, being a local phenomena on the polytope disappears with high probability.

Degeneracies

1. **Simplex** might get stuck if one of the b_is is zero.
2. More than $> m$ hyperplanes (i.e., equalities) passes through the same point.
3. Result: might not be able to make any progress at all in a pivoting step.
4. Solution I: add tiny random noise to each coefficient.
   Can be done symbolically.
   Intuitively, the degeneracy, being a local phenomena on the polytope disappears with high probability.

Degeneracies

1. **Simplex** might get stuck if one of the b_is is zero.
2. More than $> m$ hyperplanes (i.e., equalities) passes through the same point.
3. Result: might not be able to make any progress at all in a pivoting step.
4. Solution I: add tiny random noise to each coefficient. Can be done symbolically.
   Intuitively, the degeneracy, being a local phenomena on the polytope disappears with high probability.

Degeneracies

1. **Simplex** might get stuck if one of the b_is is zero.
2. More than $> m$ hyperplanes (i.e., equalities) passes through the same point.
3. Result: might not be able to make any progress at all in a pivoting step.
4. Solution I: add tiny random noise to each coefficient.
   Can be done symbolically.
   Intuitively, the degeneracy, being a local phenomena on the polytope disappears with high probability.

Degeneracies – cycling

1. Might get into cycling: a sequence of pivoting operations that do not improve the objective function, and the bases you get are cyclic (i.e., infinite loop).
2. Solution II: ***Bland's rule***.
   Always choose the lowest index variable for entering and leaving out of the possible candidates.
   (Not prove why this work - but it does.)

Degeneracies – cycling

1. Might get into cycling: a sequence of pivoting operations that do not improve the objective function, and the bases you get are cyclic (i.e., infinite loop).
2. Solution II: ***Bland's rule***.
   Always choose the lowest index variable for entering and leaving out of the possible candidates.
   (Not prove why this work - but it does.)

Correctness of LP

Definition

A solution to an LP is a ***basic solution*** if it the result of setting all the nonbasic variables to zero.

Simplex algorithm deals only with basic solutions.

Theorem

For an arbitrary linear program, the following statements are true:

(A) If there is no optimal solution, the problem is either infeasible or unbounded.

(B) If a feasible solution exists, then a basic feasible solution exists.

(C) If an optimal solution exists, then a basic optimal solution exists.

Proof: is constructive by running the simplex algorithm.

Correctness of LP

Definition

A solution to an LP is a ***basic solution*** if it the result of setting all the nonbasic variables to zero.

Simplex algorithm deals only with basic solutions.

Theorem

For an arbitrary linear program, the following statements are true:

(A) If there is no optimal solution, the problem is either infeasible or unbounded.

(B) If a feasible solution exists, then a basic feasible solution exists.

(C) If an optimal solution exists, then a basic optimal solution exists.

Proof: is constructive by running the simplex algorithm.

Correctness of LP

Definition

A solution to an LP is a ***basic solution*** if it the result of setting all the nonbasic variables to zero.

Simplex algorithm deals only with basic solutions.

Theorem

For an arbitrary linear program, the following statements are true:

(A) If there is no optimal solution, the problem is either infeasible or unbounded.

(B) If a feasible solution exists, then a basic feasible solution exists.

(C) If an optimal solution exists, then a basic optimal solution exists.

Proof: is constructive by running the simplex algorithm.

Correctness of LP

Definition

A solution to an LP is a ***basic solution*** if it the result of setting all the nonbasic variables to zero.

Simplex algorithm deals only with basic solutions.

Theorem

For an arbitrary linear program, the following statements are true:

(A) If there is no optimal solution, the problem is either infeasible or unbounded.

(B) If a feasible solution exists, then a basic feasible solution exists.

(C) If an optimal solution exists, then a basic optimal solution exists.

Proof: is constructive by running the simplex algorithm.

On the ellipsoid method and interior point methods

1. **Simplex** has exponential running time in the worst case.
2. ***ellipsoid method*** is *weakly* polynomial.
   It is polynomial in the number of bits of the input.
3. Khachian in 1979 came up with it. Useless in practice.
4. In 1984, Karmakar came up with a different method, called the *interior-point method*.
5. Also weakly polynomial. Quite useful in practice.
6. Result in arm race between the interior-point method and the simplex method.
7. BIG OPEN QUESTION: Is there *strongly* polynomial time algorithm for linear programming?

On the ellipsoid method and interior point methods

1. **Simplex** has exponential running time in the worst case.
2. ***ellipsoid method*** is *weakly* polynomial.
   It is polynomial in the number of bits of the input.
3. Khachian in 1979 came up with it. Useless in practice.
4. In 1984, Karmakar came up with a different method, called the *interior-point method*.
5. Also weakly polynomial. Quite useful in practice.
6. Result in arm race between the interior-point method and the simplex method.
7. BIG OPEN QUESTION: Is there *strongly* polynomial time algorithm for linear programming?

On the ellipsoid method and interior point methods

1. **Simplex** has exponential running time in the worst case.
2. ***ellipsoid method*** is *weakly* polynomial.
   It is polynomial in the number of bits of the input.
3. Khachian in 1979 came up with it. Useless in practice.
4. In 1984, Karmakar came up with a different method, called the *interior-point method*.
5. Also weakly polynomial. Quite useful in practice.
6. Result in arm race between the interior-point method and the simplex method.
7. BIG OPEN QUESTION: Is there *strongly* polynomial time algorithm for linear programming?

On the ellipsoid method and interior point methods

1. **Simplex** has exponential running time in the worst case.
2. ***ellipsoid method*** is *weakly* polynomial.
   It is polynomial in the number of bits of the input.
3. Khachian in 1979 came up with it. Useless in practice.
4. In 1984, Karmakar came up with a different method, called the *interior-point method*.
5. Also weakly polynomial. Quite useful in practice.
6. Result in arm race between the interior-point method and the simplex method.
7. BIG OPEN QUESTION: Is there *strongly* polynomial time algorithm for linear programming?

On the ellipsoid method and interior point methods

1. **Simplex** has exponential running time in the worst case.
2. ***ellipsoid method*** is *weakly* polynomial.
   It is polynomial in the number of bits of the input.
3. Khachian in 1979 came up with it. Useless in practice.
4. In 1984, Karmakar came up with a different method, called the *interior-point method*.
5. Also weakly polynomial. Quite useful in practice.
6. Result in arm race between the interior-point method and the simplex method.
7. BIG OPEN QUESTION: Is there *strongly* polynomial time algorithm for linear programming?

On the ellipsoid method and interior point methods

1. **Simplex** has exponential running time in the worst case.
2. ***ellipsoid method*** is *weakly* polynomial.
   It is polynomial in the number of bits of the input.
3. Khachian in 1979 came up with it. Useless in practice.
4. In 1984, Karmakar came up with a different method, called the *interior-point method*.
5. Also weakly polynomial. Quite useful in practice.
6. Result in arm race between the interior-point method and the simplex method.
7. BIG OPEN QUESTION: Is there *strongly* polynomial time algorithm for linear programming?

On the ellipsoid method and interior point methods

1. **Simplex** has exponential running time in the worst case.
2. ***ellipsoid method*** is *weakly* polynomial.
   It is polynomial in the number of bits of the input.
3. Khachian in 1979 came up with it. Useless in practice.
4. In 1984, Karmakar came up with a different method, called the *interior-point method*.
5. Also weakly polynomial. Quite useful in practice.
6. Result in arm race between the interior-point method and the simplex method.
7. BIG OPEN QUESTION: Is there *strongly* polynomial time algorithm for linear programming?

On the ellipsoid method and interior point methods

1. **Simplex** has exponential running time in the worst case.
2. ***ellipsoid method*** is *weakly* polynomial.
   It is polynomial in the number of bits of the input.
3. Khachian in 1979 came up with it. Useless in practice.
4. In 1984, Karmakar came up with a different method, called the *interior-point method*.
5. Also weakly polynomial. Quite useful in practice.
6. Result in arm race between the interior-point method and the simplex method.
7. BIG OPEN QUESTION: Is there *strongly* polynomial time algorithm for linear programming?

Duality...

1. Every linear program L has a ***dual linear program*** L'.
2. Solving the dual problem is essentially equivalent to solving the ***primal linear program*** original LP.
3. Lets look an example..

Duality by Example

$$\begin{aligned} \max \quad & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 + 4x_2 \leq 1 \\ & 3x_1 - x_2 + x_3 \leq 3 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

- η: maximal possible value of target function.
- Any feasible solution $\Rightarrow$ a lower bound on η.
- In above: $x_1 = 1, x_2 = x_3 = 0$ is feasible, and implies $z = 4$ and thus $\eta \geq 4$.
- $x_1 = x_2 = 0,\ x_3 = 3$ is feasible $\implies \eta \geq z = 9$.
- How close this solution is to opt? (i.e., η)
- If very close to optimal – might be good enough. Maybe stop?

Duality by Example

$$\begin{aligned} \max \quad & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 + 4x_2 \le 1 \\ & 3x_1 - x_2 + x_3 \le 3 \\ & x_1, x_2, x_3 \ge 0 \end{aligned}$$

1. η: maximal possible value of target function.
2. Any feasible solution $\Rightarrow$ a lower bound on η.
3. In above: $x_1 = 1, x_2 = x_3 = 0$ is feasible, and implies $z = 4$ and thus $\eta \ge 4$.
4. $x_1 = x_2 = 0,\ x_3 = 3$ is feasible $\implies \eta \ge z = 9$.
5. How close this solution is to opt? (i.e., η)
6. If very close to optimal – might be good enough. Maybe stop?

Duality by Example

$$\begin{aligned} \max \quad & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 + 4x_2 \leq 1 \\ & 3x_1 - x_2 + x_3 \leq 3 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

1. η: maximal possible value of target function.
2. Any feasible solution $\Rightarrow$ a lower bound on η.
3. In above: $x_1 = 1, x_2 = x_3 = 0$ is feasible, and implies $z = 4$ and thus $\eta \geq 4$.
4. $x_1 = x_2 = 0,\ x_3 = 3$ is feasible $\implies \eta \geq z = 9$.
5. How close this solution is to opt? (i.e., η)
6. If very close to optimal – might be good enough. Maybe stop?

Duality by Example

$$\begin{aligned} \max \quad & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 + 4x_2 \le 1 \\ & 3x_1 - x_2 + x_3 \le 3 \\ & x_1, x_2, x_3 \ge 0 \end{aligned}$$

1. η: maximal possible value of target function.
2. Any feasible solution $\Rightarrow$ a lower bound on η.
3. In above: $x_1 = 1, x_2 = x_3 = 0$ is feasible, and implies $z = 4$ and thus $\eta \ge 4$.
4. $x_1 = x_2 = 0,\ x_3 = 3$ is feasible $\implies \eta \ge z = 9$.
5. How close this solution is to opt? (i.e., η)
6. If very close to optimal – might be good enough. Maybe stop?

Duality by Example

$$\begin{aligned} \max \quad & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 + 4x_2 \leq 1 \\ & 3x_1 - x_2 + x_3 \leq 3 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

1. η: maximal possible value of target function.
2. Any feasible solution $\Rightarrow$ a lower bound on η.
3. In above: $x_1 = 1, x_2 = x_3 = 0$ is feasible, and implies $z = 4$ and thus $\eta \geq 4$.
4. $x_1 = x_2 = 0,\ x_3 = 3$ is feasible $\implies \eta \geq z = 9$.
5. How close this solution is to opt? (i.e., η)
6. If very close to optimal – might be good enough. Maybe stop?

Duality by Example

$$\begin{aligned} \max \quad & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 + 4x_2 \leq 1 \\ & 3x_1 - x_2 + x_3 \leq 3 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

1. η: maximal possible value of target function.
2. Any feasible solution $\Rightarrow$ a lower bound on η.
3. In above: $x_1 = 1, x_2 = x_3 = 0$ is feasible, and implies $z = 4$ and thus $\eta \geq 4$.
4. $x_1 = x_2 = 0,\ x_3 = 3$ is feasible $\implies \eta \geq z = 9$.
5. How close this solution is to opt? (i.e., η)
6. If very close to optimal – might be good enough. Maybe stop?

Duality by Example: II

$$\begin{aligned} \max \quad & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 + 4x_2 \le 1 \\ & 3x_1 - x_2 + x_3 \le 3 \\ & x_1, x_2, x_3 \ge 0 \end{aligned}$$

- Add the first inequality (multiplied by 2) to the second inequality (multiplied by 3):

$$\begin{aligned} 2(\ x_1 + 4x_2 \qquad\ ) &\le 2(1) \\ +3(3x_1 - x_2 + x_3) &\le 3(3). \end{aligned}$$

- The resulting inequality is

$$11x_1 + 5x_2 + 3x_3 \le 11. \tag{1}$$

Duality by Example: II

$$
\begin{array}{rl}
\max & z = 4x_1 + x_2 + 3x_3 \\
\text{s.t.} & x_1 + 4x_2 \leq 1 \\
 & 3x_1 - x_2 + x_3 \leq 3 \\
 & x_1, x_2, x_3 \geq 0
\end{array}
$$

1. Add the first inequality (multiplied by 2) to the second inequality (multiplied by 3):

$$
\begin{array}{r}
2(\ x_1 + 4x_2 \qquad\ ) \leq 2(1) \\
+3(3x_1 - x_2 + x_3) \leq 3(3).
\end{array}
$$

2. The resulting inequality is

$$11x_1 + 5x_2 + 3x_3 \leq 11. \tag{1}$$

Duality by Example: II

$$\begin{aligned} \max \quad & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 + 4x_2 \le 1 \\ & 3x_1 - x_2 + x_3 \le 3 \\ & x_1, x_2, x_3 \ge 0 \end{aligned}$$

1. got $11x_1 + 5x_2 + 3x_3 \le 11$.
2. inequality must hold for any feasible solution of L.
3. Objective: $z = 4x_1 + x_2 + 3x_3$ and x_1, x_2 and x_3 are all non-negative.
4. Inequality above has larger coefficients than objective (for corresponding variables)
5. For any feasible solution: $z = 4x_1 + x_2 + 3x_3 \le 11x_1 + 5x_2 + 3x_3 \le 11,$

Duality by Example: II

$$\begin{aligned} \max \quad & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 + 4x_2 \le 1 \\ & 3x_1 - x_2 + x_3 \le 3 \\ & x_1, x_2, x_3 \ge 0 \end{aligned}$$

1. got $11x_1 + 5x_2 + 3x_3 \le 11$.
2. inequality must hold for any feasible solution of L.
3. Objective: $z = 4x_1 + x_2 + 3x_3$ and x_1, x_2 and x_3 are all non-negative.
4. Inequality above has larger coefficients than objective (for corresponding variables)
5. For any feasible solution: $z = 4x_1 + x_2 + 3x_3 \le 11x_1 + 5x_2 + 3x_3 \le 11$,

Duality by Example: II

$$\begin{array}{rl} \max & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} & x_1 + 4x_2 \leq 1 \\ & 3x_1 - x_2 + x_3 \leq 3 \\ & x_1, x_2, x_3 \geq 0 \end{array}$$

1. got $11x_1 + 5x_2 + 3x_3 \leq 11$.
2. inequality must hold for any feasible solution of L.
3. Objective: $z = 4x_1 + x_2 + 3x_3$ and x_1, x_2 and x_3 are all non-negative.
4. Inequality above has larger coefficients than objective (for corresponding variables)
5. For any feasible solution: $z = 4x_1 + x_2 + 3x_3 \leq 11x_1 + 5x_2 + 3x_3 \leq 11$,

Duality by Example: II

$$\begin{aligned} \max \quad & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 + 4x_2 \leq 1 \\ & 3x_1 - x_2 + x_3 \leq 3 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

1. got $11x_1 + 5x_2 + 3x_3 \leq 11$.
2. inequality must hold for any feasible solution of L.
3. Objective: $z = 4x_1 + x_2 + 3x_3$ and x_1, x_2 and x_3 are all non-negative.
4. Inequality above has larger coefficients than objective (for corresponding variables)
5. For any feasible solution: $z = 4x_1 + x_2 + 3x_3 \leq 11x_1 + 5x_2 + 3x_3 \leq 11$,

Duality by Example: II

$$\begin{aligned} \max \quad & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 + 4x_2 \leq 1 \\ & 3x_1 - x_2 + x_3 \leq 3 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

1. got $11x_1 + 5x_2 + 3x_3 \leq 11$.
2. inequality must hold for any feasible solution of L.
3. Objective: $z = 4x_1 + x_2 + 3x_3$ and $x_{1,}x_2$ and x_3 are all non-negative.
4. Inequality above has larger coefficients than objective (for corresponding variables)
5. For any feasible solution: $z = 4x_1 + x_2 + 3x_3 \leq 11x_1 + 5x_2 + 3x_3 \leq 11$,

Duality by Example: III

$$\begin{aligned} \max \quad & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 + 4x_2 \le 1 \\ & 3x_1 - x_2 + x_3 \le 3 \\ & x_1, x_2, x_3 \ge 0 \end{aligned}$$

- For any feasible solution:
  $z = 4x_1 + x_2 + 3x_3 \le 11x_1 + 5x_2 + 3x_3 \le 11,$
- Opt solution is LP L is somewhere between 9 and 11.
- Multiply first inequality by y_1, second inequality by y_2 and add them up:

$$\begin{array}{rcrcrcl} y_1(x_1 & + & 4x_2 & & &) \le & y_1(1) \\ + \; y_2(3x_1 & - & x_2 & + & x_3 &) \le & y_2(3) \\ \hline (y_1 + 3y_2)x_1 & + & (4y_1 - y_2)x_2 & + & y_2 x_3 & \le & y_1 + 3y_2. \end{array}$$

Duality by Example: III

$$\begin{array}{ll} \max & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} & x_1 + 4x_2 \leq 1 \\ & 3x_1 - x_2 + x_3 \leq 3 \\ & x_1, x_2, x_3 \geq 0 \end{array}$$

1. For any feasible solution:
   $z = 4x_1 + x_2 + 3x_3 \leq 11x_1 + 5x_2 + 3x_3 \leq 11,$
2. Opt solution is LP L is somewhere between 9 and 11.
3. Multiply first inequality by y_1, second inequality by y_2 and add them up:

$$\begin{array}{lllllll} & y_1(x_1 & + & 4x_2 & &) \leq & y_1(1) \\ + & y_2(3x_1 & - & x_2 & + \; x_3 &) \leq & y_2(3) \\ \hline & (y_1 + 3y_2)x_1 & + & (4y_1 - y_2)x_2 & + \; y_2x_3 & \leq & y_1 + 3y_2. \end{array}$$

Duality by Example: III

$$\begin{array}{rl} \max & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} & x_1 + 4x_2 \ \leq 1 \\ & 3x_1 - x_2 + x_3 \leq 3 \\ & x_1, x_2, x_3 \geq 0 \end{array}$$

1. For any feasible solution:
   $z = 4x_1 + x_2 + 3x_3 \leq 11x_1 + 5x_2 + 3x_3 \leq 11,$
2. Opt solution is LP L is somewhere between 9 and 11.
3. Multiply first inequality by y_1, second inequality by y_2 and add them up:

$$\begin{array}{ccccccc} y_1(x_1 & + & 4x_2 & & &) \leq & y_1(1) \\ +\ y_2(3x_1 & - & x_2 & + & x_3 &) \leq & y_2(3) \\ \hline (y_1 + 3y_2)x_1 & + & (4y_1 - y_2)x_2 & + & y_2x_3 & \leq & y_1 + 3y_2. \end{array}$$

Duality by Example: IV

$$\begin{aligned} \max \quad & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 + 4x_2 \leq 1 \\ & 3x_1 - x_2 + x_3 \leq 3 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

- $(y_1 + 3y_2)x_1 + (4y_1 - y_2)x_2 + y_2 x_3 \leq y_1 + 3y_2$.
  - Compare to target function – require expression bigger than target function in each variable.

$\implies z = 4x_1 + x_2 + 3x_3 \leq (y_1 + 3y_2)x_1 + (4y_1 - y_2)x_2 + y_2 x_3 \leq y_1 + 3y_2$.

Duality by Example: IV

$$\begin{aligned} \max \quad & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 + 4x_2 \leq 1 \\ & 3x_1 - x_2 + x_3 \leq 3 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

- $(y_1 + 3y_2)x_1 + (4y_1 - y_2)x_2 + y_2 x_3 \leq y_1 + 3y_2$.
  - Compare to target function – require expression bigger than target function in each variable.

$\Longrightarrow \quad z = 4x_1 + x_2 + 3x_3 \leq (y_1 + 3y_2)x_1 + (4y_1 - y_2)x_2 + y_2 x_3 \leq y_1 + 3y_2$.

Duality by Example: IV

$$\begin{aligned} \max \quad & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 + 4x_2 \le 1 \\ & 3x_1 - x_2 + x_3 \le 3 \\ & x_1, x_2, x_3 \ge 0 \end{aligned}$$

- $(y_1 + 3y_2)x_1 + (4y_1 - y_2)x_2 + y_2x_3 \le y_1 + 3y_2.$

$$\begin{aligned} 4 &\le y_1 + 3y_2 \\ 1 &\le 4y_1 - y_2 \\ 3 &\le y_2, \end{aligned}$$

- Compare to target function – require expression bigger than target function in each variable.

$\Longrightarrow \quad z = 4x_1 + x_2 + 3x_3 \le (y_1 + 3y_2)x_1 + (4y_1 - y_2)x_2 + y_2x_3 \le y_1 + 3y_2.$

Duality by Example: IV

$$\begin{aligned} \max \quad & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 + 4x_2 \leq 1 \\ & 3x_1 - x_2 + x_3 \leq 3 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

- $(y_1 + 3y_2)x_1 + (4y_1 - y_2)x_2 + y_2 x_3 \leq y_1 + 3y_2.$

$$\begin{aligned} 4 &\leq y_1 + 3y_2 \\ 1 &\leq 4y_1 - y_2 \\ 3 &\leq y_2, \end{aligned}$$

- Compare to target function – require expression bigger than target function in each variable.

$$\implies z = 4x_1 + x_2 + 3x_3 \leq (y_1 + 3y_2)x_1 + (4y_1 - y_2)x_2 + y_2 x_3 \leq y_1 + 3y_2.$$

Duality by Example: IV

$$\begin{aligned} \max \quad & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 + 4x_2 \leq 1 \\ & 3x_1 - x_2 + x_3 \leq 3 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

- $(y_1 + 3y_2)x_1 + (4y_1 - y_2)x_2 + y_2 x_3 \leq y_1 + 3y_2$.

$$\begin{aligned} 4 &\leq y_1 + 3y_2 \\ 1 &\leq 4y_1 - y_2 \\ 3 &\leq y_2, \end{aligned}$$

- Compare to target function – require expression bigger than target function in each variable.

$$\Longrightarrow \quad z = 4x_1 + x_2 + 3x_3 \leq (y_1 + 3y_2)x_1 + (4y_1 - y_2)x_2 + y_2 x_3 \leq y_1 + 3y_2.$$

Duality by Example: IV

Primal LP:

$$\begin{aligned} \max \quad & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} \quad & x_1 + 4x_2 \leq 1 \\ & 3x_1 - x_2 + x_3 \leq 3 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

Dual LP: $\widehat{L}$

$$\begin{aligned} \min \quad & y_1 + 3y_2 \\ \text{s.t.} \quad & y_1 + 3y_2 \geq 4 \\ & 4y_1 - y_2 \geq 1 \\ & y_2 \geq 3 \\ & y_1, y_2 \geq 0. \end{aligned}$$

- Best upper bound on η (max value of z) then solve the LP $\widehat{L}$.
- $\widehat{L}$: Dual program to L.
- opt. solution of $\widehat{L}$ is an upper bound on optimal solution for L.

Duality by Example: IV

Primal LP:

$$\begin{array}{ll} \max & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} & x_1 + 4x_2 \leq 1 \\ & 3x_1 - x_2 + x_3 \leq 3 \\ & x_1, x_2, x_3 \geq 0 \end{array}$$

Dual LP: $\widehat{L}$

$$\begin{array}{ll} \min & y_1 + 3y_2 \\ \text{s.t.} & y_1 + 3y_2 \geq 4 \\ & 4y_1 - y_2 \geq 1 \\ & y_2 \geq 3 \\ & y_1, y_2 \geq 0. \end{array}$$

- Best upper bound on η (max value of z) then solve the LP $\widehat{L}$.
- $\widehat{L}$: Dual program to L.
- opt. solution of $\widehat{L}$ is an upper bound on optimal solution for L.

Duality by Example: IV

Primal LP:

$$\begin{array}{ll} \max & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} & x_1 + 4x_2 \leq 1 \\ & 3x_1 - x_2 + x_3 \leq 3 \\ & x_1, x_2, x_3 \geq 0 \end{array}$$

Dual LP: $\widehat{L}$

$$\begin{array}{ll} \min & y_1 + 3y_2 \\ \text{s.t.} & y_1 + 3y_2 \geq 4 \\ & 4y_1 - y_2 \geq 1 \\ & y_2 \geq 3 \\ & y_1, y_2 \geq 0. \end{array}$$

1. Best upper bound on η (max value of z) then solve the LP $\widehat{L}$.
2. $\widehat{L}$: Dual program to L.
3. opt. solution of $\widehat{L}$ is an upper bound on optimal solution for L.

Duality by Example: IV

Primal LP:

$$\begin{array}{ll} \max & z = 4x_1 + x_2 + 3x_3 \\ \text{s.t.} & x_1 + 4x_2 \leq 1 \\ & 3x_1 - x_2 + x_3 \leq 3 \\ & x_1, x_2, x_3 \geq 0 \end{array}$$

Dual LP: $\widehat{L}$

$$\begin{array}{ll} \min & y_1 + 3y_2 \\ \text{s.t.} & y_1 + 3y_2 \geq 4 \\ & 4y_1 - y_2 \geq 1 \\ & y_2 \geq 3 \\ & y_1, y_2 \geq 0. \end{array}$$

1. Best upper bound on η (max value of z) then solve the LP $\widehat{L}$.
2. $\widehat{L}$: Dual program to L.
3. opt. solution of $\widehat{L}$ is an upper bound on optimal solution for L.

Primal program/Dual program

$$\begin{aligned} \max \quad & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \le b_i, \\ & \qquad \text{for } i = 1, \ldots, m, \\ & x_j \ge 0, \\ & \qquad \text{for } j = 1, \ldots, n. \end{aligned}$$

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} b_i y_i \\ \text{s.t.} \quad & \sum_{i=1}^{m} a_{ij} y_i \ge c_j, \\ & \qquad \text{for } j = 1, \ldots, n, \\ & y_i \ge 0, \\ & \qquad \text{for } i = 1, \ldots, m. \end{aligned}$$

Primal program/Dual program

Dual variables \ Primal variables	$x_1 \geqq 0$	$x_2 \geqq 0$	$x_3 \geqq 0$	$\cdots$	$x_n \geqq 0$	Primal relation	Min v
$y_1 \geqq 0$	a_{11}	a_{12}	a_{13}	$\cdots$	a_{1n}	$\leqq$	b_1
$y_2 \geqq 0$	a_{21}	a_{22}	a_{23}	$\cdots$	a_{2n}	$\leqq$	b_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$
$y_m \geqq 0$	a_{m1}	a_{m2}	a_{m3}	$\cdots$	a_{mn}	$\leqq$	b_m
Dual Relation	$\geqq$	$\geqq$	$\geqq$		$\geqq$		
Max z	c_1	c_2	c_3	$\cdots$	c_n		

$$\begin{aligned} \max \quad & c^T x \\ \text{s. t.} \quad & Ax \leq b. \\ & x \geq 0. \end{aligned}$$

$$\begin{aligned} \min \quad & y^T b \\ \text{s. t.} \quad & y^T A \geq c^T. \\ & y \geq 0. \end{aligned}$$

Primal program/Dual program

What happens when you take the dual of the dual?

$$\begin{aligned} \max \quad & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \\ & \qquad \text{for } i = 1, \ldots, m, \\ & x_j \geq 0, \\ & \qquad \text{for } j = 1, \ldots, n. \end{aligned}$$

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} b_i y_i \\ \text{s.t.} \quad & \sum_{i=1}^{m} a_{ij} y_i \geq c_j, \\ & \qquad \text{for } j = 1, \ldots, n, \\ & y_i \geq 0, \\ & \qquad \text{for } i = 1, \ldots, m. \end{aligned}$$

Primal program / Dual program in standard form

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{n} c_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \le b_i, \\
& \qquad \text{for } i = 1, \ldots, m, \\
& x_j \ge 0, \\
& \qquad \text{for } j = 1, \ldots, n.
\end{aligned}
$$

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{m} (-b_i) y_i \\
\text{s.t.} \quad & \sum_{i=1}^{m} (-a_{ij}) y_i \le -c_j, \\
& \qquad \text{for } j = 1, \ldots, n, \\
& y_i \ge 0, \\
& \qquad \text{for } i = 1, \ldots, m.
\end{aligned}
$$

Dual program in standard form

Dual of a dual program

$$\begin{aligned}
\max \quad & \sum_{i=1}^{m} (-b_i) y_i \\
\text{s.t.} \quad & \sum_{i=1}^{m} (-a_{ij}) y_i \le -c_j, \\
& \qquad \text{for } j = 1, \ldots, n, \\
& y_i \ge 0, \\
& \qquad \text{for } i = 1, \ldots, m.
\end{aligned}$$

$$\begin{aligned}
\min \quad & \sum_{j=1}^{n} -c_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{n} (-a_{ij}) x_j \ge -b_i, \\
& \qquad \text{for } i = 1, \ldots, m, \\
& x_j \ge 0, \\
& \qquad \text{for } j = 1, \ldots, n.
\end{aligned}$$

Dual of dual program

Dual of a dual program written in standard form

$$\min \sum_{j=1}^{n} -c_j x_j$$
$$\text{s.t.} \sum_{j=1}^{n} (-a_{ij}) x_j \geq -b_i, \quad \text{for } i = 1, \ldots, m,$$
$$x_j \geq 0, \quad \text{for } j = 1, \ldots, n.$$

$$\max \sum_{j=1}^{n} c_j x_j$$
$$\text{s.t.} \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad \text{for } i = 1, \ldots, m,$$
$$x_j \geq 0, \quad \text{for } j = 1, \ldots, n.$$

$\Longrightarrow$ Dual of the dual LP is the primal LP!

Dual of dual program

Dual of a dual program written in standard form

$$\begin{aligned} \min \sum_{j=1}^{n} -c_j x_j & \\ \text{s.t.} \sum_{j=1}^{n} (-a_{ij}) x_j \geq -b_i, & \\ \text{for } i = 1, \ldots, m, & \\ x_j \geq 0, & \\ \text{for } j = 1, \ldots, n. & \end{aligned}$$

$$\begin{aligned} \max \sum_{j=1}^{n} c_j x_j & \\ \text{s.t.} \sum_{j=1}^{n} a_{ij} x_j \leq b_i, & \\ \text{for } i = 1, \ldots, m, & \\ x_j \geq 0, & \\ \text{for } j = 1, \ldots, n. & \end{aligned}$$

$\implies$ Dual of the dual LP is the primal LP!

Result

Proved the following:

Lemma

Let $\boldsymbol{L}$ be an LP, and let $\boldsymbol{L}'$ be its dual. Let $\boldsymbol{L}''$ be the dual to $\boldsymbol{L}'$. Then $\boldsymbol{L}$ and $\boldsymbol{L}''$ are the same LP.

Weak duality theorem

Theorem

If $(x_1, x_2, \ldots, x_n)$ *is feasible for the primal* LP *and* $(y_1, y_2, \ldots, y_m)$ *is feasible for the dual* LP, *then*

$$\sum_j c_j x_j \leq \sum_i b_i y_i.$$

Namely, all the feasible solutions of the dual bound all the feasible solutions of the primal.

Weak duality theorem – proof

Proof.

By substitution from the dual form, and since the two solutions are feasible, we know that

$$\sum_j c_j x_j \le \sum_j \left(\sum_{i=1}^{m} y_i a_{ij} \right) x_j \le \sum_i \left(\sum_j a_{ij} x_j \right) y_i \le \sum_i b_i y_i \,.$$

□

- y being dual feasible implies $c^T \le y^T A$
- x being primal feasible implies $Ax \le b$
- $\Rightarrow c^T x \le (y^T A)x \le y^T (Ax) \le y^T b$

Weak duality theorem – proof

Proof.

By substitution from the dual form, and since the two solutions are feasible, we know that

$$\sum_j c_j x_j \leq \sum_j \left(\sum_{i=1}^{m} y_i a_{ij} \right) x_j \leq \sum_i \left(\sum_j a_{ij} x_j \right) y_i \leq \sum_i b_i y_i \, .$$

□

1. y being dual feasible implies $c^T \leq y^T A$
2. x being primal feasible implies $Ax \leq b$
3. $\Rightarrow c^T x \leq (y^T A)x \leq y^T(Ax) \leq y^T b$

Weak duality is weak...

1. If apply the weak duality theorem on the dual program,
2. $$\implies \sum_{i=1}^{m}(-b_i)y_i \leq \sum_{j=1}^{n} -c_j x_j,$$
3. which is the original inequality in the weak duality theorem.
4. Weak duality theorem does not imply the strong duality theorem which will be discussed next.

Weak duality is weak...

1. If apply the weak duality theorem on the dual program,
2. $$\Longrightarrow \sum_{i=1}^{m}(-b_i)y_i \leq \sum_{j=1}^{n} -c_jx_j,$$
3. which is the original inequality in the weak duality theorem.
4. Weak duality theorem does not imply the strong duality theorem which will be discussed next.

Weak duality is weak...

- If apply the weak duality theorem on the dual program,
- $\implies \sum_{i=1}^{m}(-b_i)y_i \leq \sum_{j=1}^{n} -c_j x_j$,
- which is the original inequality in the weak duality theorem.
- Weak duality theorem does not imply the strong duality theorem which will be discussed next.

Weak duality is weak...

1. If apply the weak duality theorem on the dual program,
2. $\Longrightarrow \sum_{i=1}^{m}(-b_i)y_i \leq \sum_{j=1}^{n} -c_jx_j,$
3. which is the original inequality in the weak duality theorem.
4. Weak duality theorem does not imply the strong duality theorem which will be discussed next.

The strong duality theorem

Theorem (Strong duality theorem.)

If the primal LP *problem has an optimal solution* $x^* = \left(x_1^*, \ldots, x_n^*\right)$ *then the dual also has an optimal solution,* $y^* = \left(y_1^*, \ldots, y_m^*\right)$*, such that*

$$\sum_j c_j x_j^* = \sum_i b_i y_i^*.$$

Proof is tedious and omitted.

Shortest path

- $\mathbf{G} = (\mathbf{V}, \mathbf{E})$: graph. $\mathbf{s}$: source , $\mathbf{t}$: target
- $\forall\, (u \to v) \in \mathbf{E}$: weight $\omega(u, v)$ on edge.
- Q: Comp. shortest $\mathbf{s}$-$\mathbf{t}$ path.
- No edges into $\mathbf{s}$/out of $\mathbf{t}$.
- d_x: var=dist. $\mathbf{s}$ to x, $\forall x \in \mathbf{V}$.
- $\forall\, (u \to v) \in \mathbf{E}$: $d_u + \omega(u, v) \geq d_v$.
- Also $d_{\mathbf{s}} = 0$.
- Trivial solution: all variables 0.
- Target: find assignment max

Shortest path

1. $\mathbf{G} = (\mathbf{V}, \mathbf{E})$: graph. **s**: source , **t**: target
2. $\forall (u \to v) \in \mathbf{E}$: weight $\omega(u, v)$ on edge.
3. Q: Comp. shortest **s**-**t** path.
4. No edges into **s**/out of **t**.
5. d_x: var=dist. **s** to x, $\forall x \in \mathbf{V}$.
6. $\forall (u \to v) \in \mathbf{E}$: $d_u + \omega(u, v) \geq d_v$.
7. Also $d_{\mathbf{s}} = 0$.
8. Trivial solution: all variables 0.
9. Target: find assignment max

Shortest path

1. $\mathbf{G} = (\mathbf{V}, \mathbf{E})$: graph. $\mathbf{s}$: source , $\mathbf{t}$: target
2. $\forall (u \to v) \in \mathbf{E}$: weight $\omega(u, v)$ on edge.
3. Q: Comp. shortest $\mathbf{s}$-$\mathbf{t}$ path.
4. No edges into $\mathbf{s}$/out of $\mathbf{t}$.
5. d_x: var=dist. $\mathbf{s}$ to x, $\forall x \in \mathbf{V}$.
6. $\forall (u \to v) \in \mathbf{E}$: $d_u + \omega(u, v) \geq d_v$.
7. Also $d_{\mathbf{s}} = 0$.
8. Trivial solution: all variables 0.
9. Target: find assignment max d...

Shortest path

1. $\mathbf{G} = (\mathbf{V}, \mathbf{E})$: graph. **s**: source , **t**: target
2. $\forall\, (u \rightarrow v) \in \mathbf{E}$: weight $\omega(u, v)$ on edge.
3. Q: Comp. shortest **s**-**t** path.
4. No edges into **s**/out of **t**.
5. d_x: var=dist. **s** to x, $\forall x \in \mathbf{V}$.
6. $\forall\, (u \rightarrow v) \in \mathbf{E}$: $d_u + \omega(u, v) \geq d_v$.
7. Also $d_{\mathsf{s}} = 0$.
8. Trivial solution: all variables 0.
9. Target: find assignment max

Shortest path

1. $\mathbf{G} = (\mathbf{V}, \mathbf{E})$: graph. $\mathbf{s}$: source , $\mathbf{t}$: target
2. $\forall\, (u \to v) \in \mathbf{E}$: weight $\omega(u, v)$ on edge.
3. Q: Comp. shortest $\mathbf{s}$-$\mathbf{t}$ path.
4. No edges into $\mathbf{s}$/out of $\mathbf{t}$.
5. d_x: var=dist. $\mathbf{s}$ to x, $\forall x \in \mathbf{V}$.
6. $\forall\, (u \to v) \in \mathbf{E}$: $d_u + \omega(u, v) \geq d_v$.
7. Also $d_{\mathbf{s}} = 0$.
8. Trivial solution: all variables 0.
9. Target: find assignment max d

Shortest path

1. $\mathsf{G} = (\mathsf{V}, \mathsf{E})$: graph. s: source , t: target
2. $\forall\,(u \to v) \in \mathsf{E}$: weight $\omega(u, v)$ on edge.
3. Q: Comp. shortest s-t path.
4. No edges into s/out of t.
5. d_x: var=dist. s to x, $\forall x \in \mathsf{V}$.
6. $\forall\,(u \to v) \in \mathsf{E}$: $d_u + \omega(u, v) \geq d_v$.
7. Also $d_{\mathsf{s}} = 0$.
8. Trivial solution: all variables 0.
9. Target: find assignment max d

Shortest path

- $\mathbf{G} = (\mathbf{V}, \mathbf{E})$: graph. $\mathbf{s}$: source , $\mathbf{t}$: target
- $\forall (u \rightarrow v) \in \mathbf{E}$: weight $\omega(u, v)$ on edge.
- Q: Comp. shortest $\mathbf{s}$-$\mathbf{t}$ path.
- No edges into $\mathbf{s}$/out of $\mathbf{t}$.
- d_x: var=dist. $\mathbf{s}$ to x, $\forall x \in \mathbf{V}$.
- $\forall (u \rightarrow v) \in \mathbf{E}$: $d_u + \omega(u, v) \geq d_v$.
- Also $d_{\mathbf{s}} = 0$.
- Trivial solution: all variables 0.
- Target: find assignment max d

Shortest path

1. $\mathsf{G} = (\mathsf{V}, \mathsf{E})$: graph. **s**: source , **t**: target
2. $\forall (u \rightarrow v) \in \mathsf{E}$: weight $\omega(u, v)$ on edge.
3. Q: Comp. shortest **s**-**t** path.
4. No edges into **s**/out of **t**.
5. d_x: var=dist. **s** to x, $\forall x \in \mathsf{V}$.
6. $\forall (u \rightarrow v) \in \mathsf{E}$: $d_u + \omega(u, v) \geq d_v$.
7. Also $d_{\mathsf{s}} = 0$.
8. Trivial solution: all variables 0.
9. Target: find assignment max $d_{\ldots}$

Shortest path

1. $\mathsf{G} = (\mathsf{V}, \mathsf{E})$: graph. s: source , t: target
2. $\forall (u \to v) \in \mathsf{E}$: weight $\omega(u, v)$ on edge.
3. Q: Comp. shortest s-t path.
4. No edges into s/out of t.
5. d_x: var=dist. s to x, $\forall x \in \mathsf{V}$.
6. $\forall (u \to v) \in \mathsf{E}$: $d_u + \omega(u, v) \geq d_v$.
7. Also $d_{\mathsf{s}} = 0$.
8. Trivial solution: all variables 0.
9. Target: find assignment max d

Shortest path

1. $\mathbf{G} = (\mathbf{V}, \mathbf{E})$: graph. $\mathbf{s}$: source , $\mathbf{t}$: target
2. $\forall (u \to v) \in \mathbf{E}$: weight $\omega(u, v)$ on edge.
3. Q: Comp. shortest $\mathbf{s}$-$\mathbf{t}$ path.
4. No edges into $\mathbf{s}$/out of $\mathbf{t}$.
5. d_x: var=dist. $\mathbf{s}$ to x, $\forall x \in \mathbf{V}$.
6. $\forall (u \to v) \in \mathbf{E}$: $d_u + \omega(u, v) \geq d_v$.
7. Also $d_{\mathbf{s}} = 0$.
8. Trivial solution: all variables 0.
9. Target: find assignment max d

Shortest path

$$\begin{aligned} \max \quad & d_{\mathsf{t}} \\ \text{s.t.} \quad & d_{\mathsf{s}} \leq 0 \\ & d_u + \omega(u,v) \geq d_v \quad \forall (u \to v) \in \mathbf{E}, \\ & d_x \geq 0 \quad \forall x \in \mathbf{V}. \end{aligned}$$

1. $\mathbf{G} = (\mathbf{V}, \mathbf{E})$: graph. **s**: source , **t**: target
2. $\forall (u \to v) \in \mathbf{E}$: weight $\omega(u,v)$ on edge.
3. Q: Comp. shortest **s**-**t** path.
4. No edges into **s**/out of **t**.
5. d_x: var=dist. **s** to x, $\forall x \in \mathbf{V}$.
6. $\forall (u \to v) \in \mathbf{E}$: $d_u + \omega(u,v) \geq d_v$.
7. Also $d_{\mathsf{s}} = 0$.
8. Trivial solution: all variables 0.
9. Target: find assignment max d...

Shortest path

$$
\begin{aligned}
\max \quad & d_t \\
\text{s.t.} \quad & d_s \leq 0 \\
& d_u + \omega(u, v) \geq d_v \\
& \qquad \forall (u \to v) \in \mathbf{E} \\
& d_x \geq 0 \quad \forall x \in \mathbf{V}.
\end{aligned}
$$

Equivalently:

$$
\begin{aligned}
\max \quad & d_t \\
\text{s.t.} \quad & d_s \leq 0 \\
& d_v - d_u \leq \omega(u, v) \\
& \qquad \forall (u \to v) \in \mathbf{E}, \\
& d_x \geq 0 \qquad \forall x \in \mathbf{V}.
\end{aligned}
$$

1. $\mathbf{G} = (\mathbf{V}, \mathbf{E})$: graph. **s**: source , **t**: target
2. $\forall (u \to v) \in \mathbf{E}$: weight $\omega(u, v)$ on edge.
3. Q: Comp. shortest **s**-**t** path.
4. No edges into **s**/out of **t**.
5. d_x: var=dist. **s** to x, $\forall x \in \mathbf{V}$.
6. $\forall (u \to v) \in \mathbf{E}$: $d_u + \omega(u, v) \geq d_v$.
7. Also $d_s = 0$.
8. Trivial solution: all variables 0.
9. Target: find assignment max d

The dual

$$\min \sum_{(u \to v) \in \mathsf{E}} y_{uv} \omega(u, v)$$

$$\text{s.t.} \quad y_{\mathsf{s}} - \sum_{(\mathsf{s} \to u) \in \mathsf{E}} y_{\mathsf{s}u} \geq 0 \qquad ($$

$$\sum_{(u \to x) \in \mathsf{E}} y_{ux} - \sum_{(x \to v) \in \mathsf{E}} y_{xv} \geq 0$$

$$\forall x \in \mathsf{V} \setminus \{\mathsf{s}, \mathsf{t}\} \qquad (*$$

$$\sum_{(u \to \mathsf{t}) \in \mathsf{E}} y_{u\mathsf{t}} \geq 1 \qquad (**$$

$$y_{uv} \geq 0, \quad \forall (u \to v) \in \mathsf{E},$$

$$y_{\mathsf{s}} \geq 0.$$

$$\max \quad d_{\mathsf{t}}$$

$$\text{s.t.} \quad d_{\mathsf{s}} \leq 0$$

$$d_v - d_u \leq \omega(u, v) \quad \forall (u \to v) \in \mathsf{E},$$

$$d_x \geq 0 \quad \forall x \in \mathsf{V}.$$

The dual – details

1. y_{uv}: dual variable for the edge $(u \rightarrow v)$.
2. y_s: dual variable for $d_s \leq 0$
3. Think about the y_{uv} as a flow on the edge y_{uv}.
4. Assume that weights are positive.
5. LP is min cost flow of sending 1 unit flow from source **s** to **t**.
6. Indeed... (**) can be assumed to be hold with equality in the optimal solution...
7. conservation of flow.
8. Equation (***) implies that one unit of flow arrives to the sink **t**.
9. (*) implies that at least y_s units of flow leaves the source.
10. Remaining of LP implies that $y_s \geq 1$.

The dual – details

1. y_{uv}: dual variable for the edge $(u \to v)$.
2. y_s: dual variable for $d_s \leq 0$
3. Think about the y_{uv} as a flow on the edge y_{uv}.
4. Assume that weights are positive.
5. LP is min cost flow of sending 1 unit flow from source **s** to **t**.
6. Indeed... (**) can be assumed to be hold with equality in the optimal solution...
7. conservation of flow.
8. Equation (***) implies that one unit of flow arrives to the sink **t**.
9. (*) implies that at least y_s units of flow leaves the source.
10. Remaining of LP implies that $y_s \geq 1$.

The dual – details

1. y_{uv}: dual variable for the edge $(u \rightarrow v)$.
2. y_s: dual variable for $d_\mathsf{s} \leq 0$
3. Think about the y_{uv} as a flow on the edge y_{uv}.
4. Assume that weights are positive.
5. LP is min cost flow of sending 1 unit flow from source s to t.
6. Indeed... (**) can be assumed to be hold with equality in the optimal solution...
7. conservation of flow.
8. Equation (***) implies that one unit of flow arrives to the sink t.
9. (*) implies that at least y_s units of flow leaves the source.
10. Remaining of LP implies that $y_\mathsf{s} \geq 1$.

The dual – details

1. y_{uv}: dual variable for the edge $(u \to v)$.
2. y_s: dual variable for $d_s \leq 0$
3. Think about the y_{uv} as a flow on the edge y_{uv}.
4. Assume that weights are positive.
5. LP is min cost flow of sending 1 unit flow from source **s** to **t**.
6. Indeed... (**) can be assumed to be hold with equality in the optimal solution...
7. conservation of flow.
8. Equation (***) implies that one unit of flow arrives to the sink **t**.
9. (*) implies that at least y_s units of flow leaves the source.
10. Remaining of LP implies that $y_s \geq 1$.

The dual – details

1. y_{uv}: dual variable for the edge $(u \to v)$.
2. y_s: dual variable for $d_s \leq 0$
3. Think about the y_{uv} as a flow on the edge y_{uv}.
4. Assume that weights are positive.
5. LP is min cost flow of sending 1 unit flow from source **s** to **t**.
6. Indeed... (**) can be assumed to be hold with equality in the optimal solution...
7. conservation of flow.
8. Equation (***) implies that one unit of flow arrives to the sink **t**.
9. (*) implies that at least y_s units of flow leaves the source.
10. Remaining of LP implies that $y_s \geq 1$.

The dual – details

1. y_{uv}: dual variable for the edge $(u \to v)$.
2. y_s: dual variable for $d_s \leq 0$
3. Think about the y_{uv} as a flow on the edge y_{uv}.
4. Assume that weights are positive.
5. LP is min cost flow of sending 1 unit flow from source **s** to **t**.
6. Indeed... (**) can be assumed to be hold with equality in the optimal solution...
7. conservation of flow.
8. Equation (***) implies that one unit of flow arrives to the sink **t**.
9. (*) implies that at least y_s units of flow leaves the source.
10. Remaining of LP implies that $y_s \geq 1$.

The dual – details

1. y_{uv}: dual variable for the edge $(u \to v)$.
2. y_{s}: dual variable for $d_{\mathsf{s}} \leq 0$
3. Think about the y_{uv} as a flow on the edge y_{uv}.
4. Assume that weights are positive.
5. LP is min cost flow of sending 1 unit flow from source **s** to **t**.
6. Indeed... (**) can be assumed to be hold with equality in the optimal solution...
7. conservation of flow.
8. Equation (***) implies that one unit of flow arrives to the sink **t**.
9. (*) implies that at least y_{s} units of flow leaves the source.
10. Remaining of LP implies that $y_{\mathsf{s}} \geq 1$.

Integrality

1. In the previous example there is always an optimal solution with integral values.
2. This is not an obvious statement.
3. This is not true in general.
4. If it were true we could solve **NPC** problems with LP.

Integrality

1. In the previous example there is always an optimal solution with integral values.
2. This is not an obvious statement.
3. This is not true in general.
4. If it were true we could solve **NPC** problems with LP.

Integrality

1. In the previous example there is always an optimal solution with integral values.
2. This is not an obvious statement.
3. This is not true in general.
4. If it were true we could solve **NPC** problems with LP.

Integrality

1. In the previous example there is always an optimal solution with integral values.
2. This is not an obvious statement.
3. This is not true in general.
4. If it were true we could solve **NPC** problems with LP.

Set cover...

Details in notes...

Set cover LP:

$$\begin{aligned} \min \quad & \sum_{F_j \in \mathfrak{F}} x_j \\ \text{s.t.} \quad & \sum_{\substack{F_j \in \mathfrak{F}, \\ u_i \in F_j}} x_j \geq 1 && \forall u_i \in \mathsf{S}, \\ & x_j \geq 0 && \forall F_j \in \mathfrak{F}. \end{aligned}$$

Set cover dual is a packing LP...

Details in notes...

$$\begin{aligned} \max \quad & \sum_{u_i \in \mathsf{S}} y_i & \\ \text{s.t.} \quad & \sum_{u_i \in F_j} y_i \leq 1 & \forall F_j \in \mathfrak{F}, \\ & y_i \geq 0 & \forall u_i \in \mathsf{S}. \end{aligned}$$

Network flow

$$
\begin{aligned}
\max \quad & \sum_{(\mathsf{s}\to v)\in \mathsf{E}} x_{\mathsf{s}\to v} \\
& x_{u\to v} \le \mathsf{c}(u \to v) && \forall\, (u \to v) \in \mathsf{E} \\
& \sum_{(u\to v)\in \mathsf{E}} x_{u\to v} - \sum_{(v\to w)\in \mathsf{E}} x_{v\to w} \le 0 && \forall v \in \mathsf{V} \setminus \{\mathsf{s},\mathsf{t}\} \\
& -\sum_{(u\to v)\in \mathsf{E}} x_{u\to v} + \sum_{(v\to w)\in \mathsf{E}} x_{v\to w} \le 0 && \forall v \in \mathsf{V} \setminus \{\mathsf{s},\mathsf{t}\} \\
& 0 \le x_{u\to v} && \forall\, (u \to v) \in \mathsf{E}
\end{aligned}
$$

Dual of network flow...

$$
\begin{aligned}
\min \sum_{(u\to v)\in \mathsf{E}} \mathsf{c}(u \to v)\, y_{u\to v} & \\
d_u - d_v \leq y_{u\to v} & \qquad \forall\,(u \to v) \in \mathsf{E} \\
y_{u\to v} \geq 0 & \qquad \forall\,(u \to v) \in \mathsf{E} \\
d_{\mathsf{s}} = 1, \qquad d_{\mathsf{t}} = 0. &
\end{aligned}
$$

Under right interpretation: shortest path (see notes).

Duality and min-cut max-flow

Details in class notes

Lemma

The Min-Cut Max-Flow Theorem follows from the strong duality Theorem for Linear Programming.

Solving LPs without ever getting into a loop - symbolic perturbations

Details in the class notes.

Notes

Notes

Notes

Notes