CS711008Z Algorithm Design and Analysis

Lecture 9. Algorithm design technique: Linear programming and duality

Dongbo Bu

Institute of Computing Technology
Chinese Academy of Sciences, Beijing, China

Outline

- The first example: the dual of DIET problem;
- Understanding duality: Lagrangian duality explanation;
- Conditions of optimal solution;
- Four properties of duality for linear program;
- Solving LP using duality: Dual simplex algorithm, PRIMAL_DUAL algorithm, and interior point method;
- Applications of duality: Farkas lemma, von Neumann' s MINIMAX theorem, Yao's MINIMAX theorem, Dual problem in SVM, and SHORTESTPATH problem.

Importance of duality

- When minimizing a function $f(x)$, it is invaluable to know a lower bound of $f(x)$ in advance. Calculation of lower bound is extremely important to the design of approximation algorithm and branch-and-bound method.
- Duality and relaxation (say integer relaxation, convex relaxation) are powerful techniques to obtain a reasonable lower bound.
- Linear programs come in primal/dual pairs. It turns out that every feasible solution for one of these two problems provides a bound for the objective value for the other problem.
- The dual problems are always convex even if the primal problems are not convex.

The first example: the dual of DIET problem.

Revisiting DIET problem

- A housewife wonders how much money she must spend on foods in order to get all the energy (2000 kcal), protein (55 g), and calcium (800 mg) that she needs every day.

Food	Energy	Protein	Calcium	Price	Quantity
Oatmeal	110	4	2	3	x_1
Whole milk	160	8	285	9	x_2
Cherry pie	420	4	22	20	x_3
Pork beans	260	14	80	19	x_4

- Linear program:

$$\begin{array}{rrcrcrcrll}
\min & 3x_1 & + & 9x_2 & + & 20x_3 & + & 19x_4 & & \text{money} \\
s.t. & 110x_1 & + & 160x_2 & + & 420x_3 & + & 260x_4 & \geq 2000 & \text{energy} \\
 & 4x_1 & + & 8x_2 & + & 4x_3 & + & 14x_4 & \geq 55 & \text{protein} \\
 & 2x_1 & + & 285x_2 & + & 22x_3 & + & 80x_4 & \geq 800 & \text{calcium} \\
 & x_1 & , & x_2 & , & x_3 & , & x_4 & \geq 0 &
\end{array}$$

Dual of DIET problem: PRICING problem

- Consider a company producing protein powder, energy bar, and calcium tablet as substitution to foods.
- The company wants to design a reasonable pricing strategy to earn money as much as possible.
- However, the price cannot be arbitrarily high due to the following considerations:
  1. If the prices are competitive with foods, one might consider choosing a combination of the ingredients rather than foods;
  2. Otherwise, one will choose to buy foods directly.

LP model of PRICING problem

Food	Energy	Protein	Calcium	Price (cents)
Oatmeal	110	4	2	3
Whole milk	160	8	285	9
Cherry pie	420	4	22	20
Pork with beans	260	14	80	19
Price	y_1	y_2	y_3	

- Linear program:

$$
\begin{array}{rrcrcrcrl}
\max & 2000y_1 & + & 55y_2 & + & 800y_3 & & & \text{money} \\
s.t. & 110y_1 & + & 4y_2 & + & 2y_3 & \leq & 3 & \text{oatmeal} \\
 & 160y_1 & + & 8y_2 & + & 285y_3 & \leq & 9 & \text{milk} \\
 & 420y_1 & + & 4y_2 & + & 22y_3 & \leq & 20 & \text{pie} \\
 & 260y_1 & + & 14y_2 & + & 80y_3 & \leq & 19 & \text{pork\&beans} \\
 & y_1 & , & y_2 & , & y_3 & \geq & 0 &
\end{array}
$$

PRIMAL problem and DUAL problem

$$
\begin{array}{ccccc}
c_1 & c_2 & \dots & c_n & \\
a_{11} & a_{12} & \dots & a_{1n} & b_1 \\
a_{21} & a_{22} & \dots & a_{2n} & b_2 \\
 & & \dots & & \\
a_{m1} & a_{m2} & \dots & a_{mn} & b_m
\end{array}
$$

- PRIMAL problem and DUAL problem are two points of view of the coefficient matrix $\mathbf{A}$:
  - Primal problem: row point of view
  - Dual problem: column point of view

Primal problem

$$
\begin{array}{rcccccccccl}
\min & c_1 x_1 & + & c_2 x_2 & + & \ldots & + & c_n x_n & & & \\
& a_{11} x_1 & + & a_{12} x_2 & + & \ldots & + & a_{1n} x_n & \geq & b_1 & \\
& a_{21} x_1 & + & a_{22} x_2 & + & \ldots & + & a_{2n} x_n & \geq & b_2 & \\
& & & & & \ldots & & & & & \\
& a_{m1} x_1 & + & a_{m2} x_2 & + & \ldots & + & a_{mn} x_n & \geq & b_m & \\
& & & & & & & x_i & \geq & 0 & \text{for each } i
\end{array}
$$

- Primal problem: row point of view (in red);

$$
\begin{array}{rll}
\min & \mathbf{c}^{\mathbf{T}}\mathbf{x} & \\
s.t. & \mathbf{A}\mathbf{x} & \geq \mathbf{b} \\
 & \mathbf{x} & \geq \mathbf{0}
\end{array}
$$

Dual problem

$$
\begin{array}{ccccccc}
c_1 & c_2 & \cdots & c_n & & & \\
\vee\!| & \vee\!| & & \vee\!| & \max & & \\
y_1 a_{11} & y_1 a_{12} & \cdots & y_1 a_{1n} & y_1 b_1 & & \\
+ & + & & + & + & & \\
y_2 a_{21} & y_2 a_{22} & \cdots & y_2 a_{2n} & y_2 b_2 & & \\
+ & + & & + & + & & \\
\vdots & \vdots & \cdots & \vdots & \vdots & & \\
+ & + & & + & + & & \\
y_m a_{m1} & y_m a_{m2} & \cdots & y_m a_{mn} & y_m b_m & & \\
 & & & y_j & \geq & 0 & \text{for each } j
\end{array}
$$

- Dual problem: column point of view (in blue).

$$
\begin{array}{rrl}
\max & \mathbf{b^T y} & \\
s.t. & \mathbf{y} & \geq \mathbf{0} \\
 & \mathbf{A^T y} & \leq \mathbf{c}
\end{array}
$$

How to write DUAL problem? Case 1

- For each **constraint** in the PRIMAL problem, a **variable** is set in the DUAL problem.
- If the PRIMAL problem has **inequality constraints**, the DUAL problem is written as follows.
- Primal problem:

$$\begin{array}{rrcl} \min & \mathbf{c^T x} & & \\ s.t. & \mathbf{Ax} & \geq & \mathbf{b} \\ & \mathbf{x} & \geq & \mathbf{0} \end{array}$$

- Dual problem:

$$\begin{array}{rrcl} \max & \mathbf{b^T y} & & \\ s.t. & \mathbf{y} & \geq & \mathbf{0} \\ & \mathbf{A^T y} & \leq & \mathbf{c} \end{array}$$

How to write Dual problem? Case 2

- For each **constraint** in the Primal problem, a **variable** is set in the Dual problem.
- If the Primal problem has **inequality constraints**, the Dual problem is written as follows.
- Primal problem:

$$\begin{array}{rrcl} \min & \mathbf{c^T x} & & \\ s.t. & \mathbf{Ax} & \leq & \mathbf{b} \\ & \mathbf{x} & \geq & \mathbf{0} \end{array}$$

- Dual problem:

$$\begin{array}{rrcl} \max & \mathbf{b^T y} & & \\ s.t. & \mathbf{y} & \leq & \mathbf{0} \\ & \mathbf{A^T y} & \leq & \mathbf{c} \end{array}$$

How to write DUAL problem? Case 3

- For each **constraint** in the PRIMAL problem, a **variable** is set in the DUAL problem.
- If the PRIMAL problem has **equality constraints**, the DUAL problem is as follows.
- Primal problem:

$$\begin{array}{rrcl} \min & \mathbf{c}^{\mathbf{T}}\mathbf{x} & & \\ s.t. & \mathbf{A}\mathbf{x} & = & \mathbf{b} \\ & \mathbf{x} & \geq & \mathbf{0} \end{array}$$

- Dual problem:

$$\begin{array}{rrcl} \max & \mathbf{b}^{\mathbf{T}}\mathbf{y} & & \\ s.t. & & & \\ & \mathbf{A}^{\mathbf{T}}\mathbf{y} & \leq & \mathbf{c} \end{array}$$

- Note: there is neither $\mathbf{y} \geq \mathbf{0}$ nor $\mathbf{y} \leq \mathbf{0}$ constraint in the dual problem.

Why can the DUAL problem be written as above?
— Understanding duality from the Lagrangian dual point of view

Standard form of constrained optimization problems

- Consider the following constrained optimization problem (might be non-convex).

$$\begin{array}{rlcll} \min & f_0(\mathbf{x}) & & & \\ s.t. & f_i(\mathbf{x}) & \leq & 0 & i = 1, ..., m \\ & h_i(\mathbf{x}) & = & 0 & i = 1, ..., p \end{array}$$

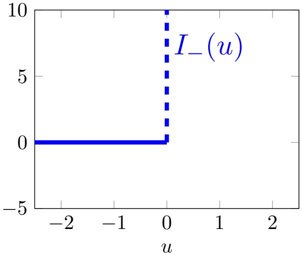
- Here the variables $\mathbf{x} \in \mathbb{R}^n$ and we use $\mathcal{D} = \bigcap_{i=0}^{m} \mathbf{dom}\ f_i \cap \bigcap_{i=1}^{p} \mathbf{dom}\ h_i$ to represent the domain of definition. We use p^* to represent the optimal value of the problem.

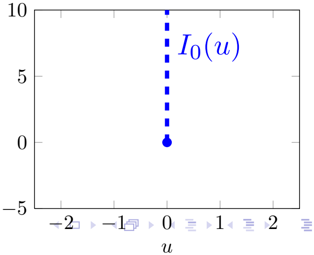
- We can transform this **constrained optimization** problem into an equivalent **unconstrained optimization** problem:

$$\min f_0(\mathbf{x}) + \sum_{i=1}^{m} I_-(f_i(\mathbf{x})) + \sum_{i=1}^{p} I_0(h_i(\mathbf{x}))$$

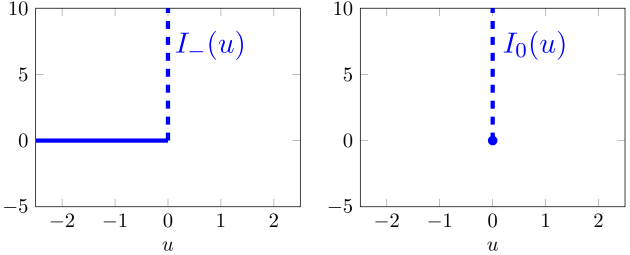
where $\mathbf{x} \in \mathcal{D}$, $I_-(u)$ and $I_0(u)$ are indicator functions for non-positive reals and the set $\{0\}$, respectively:

$$I_-(u) = \begin{cases} 0 & u \le 0 \\ \infty & u > 0 \end{cases} \qquad I_0(u) = \begin{cases} 0 & u = 0 \\ \infty & u \neq 0 \end{cases}$$





Difficulty in solving the optimization problem

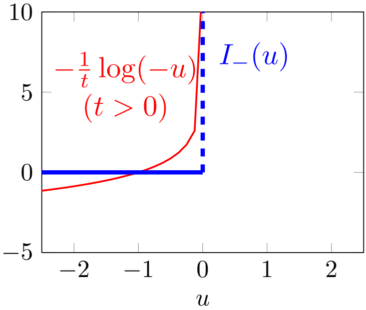


- Intuitively, $I_-(u)$ and $I_0(u)$ represent our "infinite dissatisfaction" with the violence of constraints.
- Both $I_0(u)$ and $I_-(u)$ are non-differentiable; thus, although the optimization problem

$$\min f_0(\mathbf{x}) + \sum_{i=1}^{m} I_-(f_i(\mathbf{x})) + \sum_{i=1}^{p} I_0(h_i(\mathbf{x}))$$

is unconstrained, it is not easy to solve the problem directly.
- Question: How to efficiently solve this optimization problem?

Approximating $I_-(u)$ using a differentiable function (1)

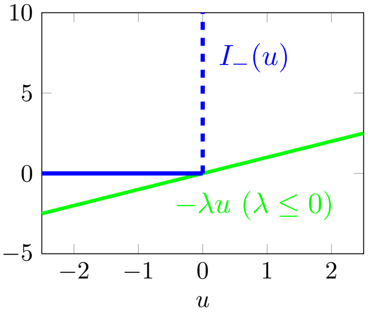


- An approximation to $I_-(u)$ is **logarithm barrier function**:

$$\hat{I}_-(u) = -\frac{1}{t}\log(-u) \qquad (t > 0)$$

- The difference between $\hat{I}_-(u)$ and $I_-(u)$ decreases as t increases. This approximation was used in interior point method.

Approximating $I_-(u)$ using a differentiable function (2)

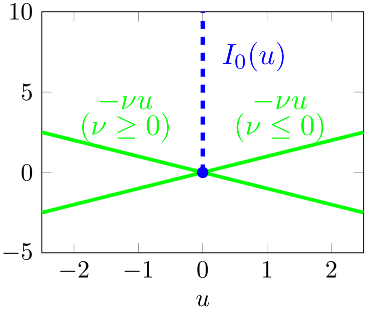


- Another approximation to $I_-(u)$ is a simple **linear function**:

$$\hat{I}_-(u) = -\lambda u \qquad (\lambda \leq 0)$$

- Although the difference between $\hat{I}_-(u)$ and $I_-(u)$ is large, $\hat{I}_-(u)$ still provides lower bound information of $I_-(u)$.

Approximating $I_0(u)$ using a differentiable function



- $I_0(u)$ can also be approximated using **linear function**:

$$\hat{I}_0(u) = -\nu u$$

- Although $\hat{I}_0(u)$ deviates considerably from $I_0(u)$, $\hat{I}_0(u)$ still provides lower bound information of $I_0(u)$.
- It is worthy pointed out that unlike $\hat{I}_-(u)$, $\hat{I}_0(u)$ has no restriction on ν.

- Consider the unconstrained optimization problem

$$\min f_0(\mathbf{x}) + \sum_{i=1}^{m} I_-(f_i(\mathbf{x})) + \sum_{i=1}^{p} I_0(h_i(\mathbf{x})).$$

- Now let's replace $I_-(u)$ with $-\lambda u$ $(\lambda \leq 0)$ and replace $I_0(u)$ with $-\nu u$. Then the objective function becomes **Lagrangian function**

$$L(\mathbf{x}, \lambda, \nu) = f_0(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) - \sum_{i=1}^{p} \nu_i h_i(\mathbf{x})$$

  which is a lower bound of $f_0(\mathbf{x})$ for any feasible solution $\mathbf{x} \in \mathcal{D}$ when $\lambda \leq 0$.

- Here we call λ_i **Lagrangian multiplier** for the i-th inequality constraint $f_i(\mathbf{x}) \leq 0$ and ν_i **Lagrangian multiplier** for the i-th equality constraint $h_i(\mathbf{x}) = 0$.

Lagrangian connecting primal and dual: An example

- Primal problem:

$$\begin{array}{llll} \min & x^2 - 2x & & \\ s.t. & -x & \leq & 0 \end{array}$$

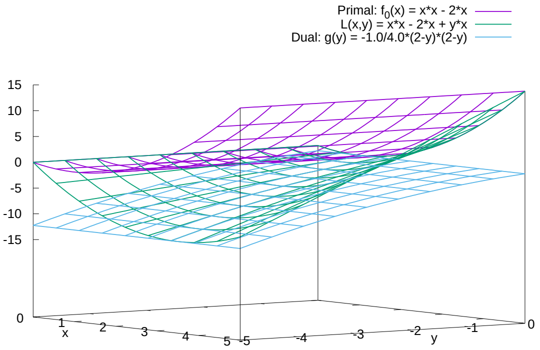
- Lagrangian:

$$L(x, \lambda) = x^2 - 2x + \lambda x$$

- Note that $L(x, \lambda)$ is a lower bound of the primal objective function x when $\lambda \leq 0$ and $x \geq 0$.
- Dual problem:

$$\begin{array}{ll} \max & -\frac{1}{4}(2 - \lambda)^2 \\ s.t. & \lambda \leq 0 \end{array}$$

Lagrangian connecting primal and dual



- Observation: PRIMAL objective function $\geq$ Lagrangian $\geq$ DUAL objective function in the feasible region.

Lagrangian dual function and Lagrangian dual problem

Lagrangian dual function

- Consider the following constrained optimization problem.

$$\begin{array}{llll} \min & f_0(\mathbf{x}) & & \\ s.t. & f_i(\mathbf{x}) & \leq & 0 \quad i = 1, ..., m \\ & h_i(\mathbf{x}) & = & 0 \quad i = 1, ..., p \end{array}$$

- **Lagrangian function:**

$$L(\mathbf{x}, \lambda, \nu) = f_0(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) - \sum_{i=1}^{p} \nu_i h_i(\mathbf{x})$$

  which is a lower bound of $f_0(\mathbf{x})$ for any feasible solution $\mathbf{x}$ when $\lambda \leq 0$.

- Now let's consider the inferior bound of Lagrangian function (called **Lagrangian dual function**):

$$g(\lambda, \nu) = \inf_{\mathbf{x} \in \mathcal{D}} L(\mathbf{x}, \lambda, \nu)$$

Lagrangian dual problem

- Lagrangian dual function provides lower bound of the primal objective function, i.e.

$$f_0(\mathbf{x}) \geq L(\mathbf{x}, \lambda, \nu) \geq g(\lambda, \nu)$$

for any feasible solution $\mathbf{x} \in \mathcal{D}$ when $\lambda \leq 0$.

- Now let's try to find **the tightest lower bound** of the primal objective function, which can be obtained by solving the following **Lagrangian dual problem**:

$$\begin{array}{rl} \max & g(\lambda, \nu) \\ s.t. & \lambda \geq \mathbf{0} \end{array}$$

Lagrangian function: An example I

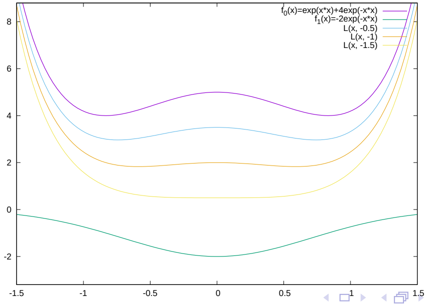
- Consider the following primal problem:

$$\begin{array}{llll} \min & e^{x^2} + 4e^{-x^2} & & \\ s.t. & -2e^{-x^2} + 1 & \leq & 0 \quad i = 1, ..., m \end{array}$$

- Lagrangian function:

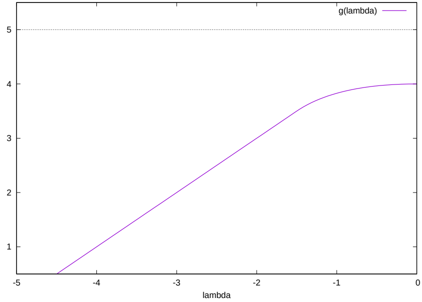
$$L(x, \lambda) \quad = \quad e^{x^2} + 4e^{-x^2} - \lambda(-2e^{-x^2} + 1)$$

where $\lambda \leq 0$.



- Lagrangian dual function:

$$
\begin{aligned}
g(\lambda) &= \inf_{x \in \mathbb{R}} L(x, \lambda) \\
&= \begin{cases} 5 + \lambda & \lambda \le -1.5 \\ 2\sqrt{4 + 2\lambda} - \lambda & -1.5 \le \lambda \le 0 \end{cases}
\end{aligned}
$$



Deriving dual problem of linear program in standard form

Dual problem of LP problem in standard form

- Consider a LP problem in standard form:

$$\begin{array}{rrcl} \min & \mathbf{c}^{\mathbf{T}}\mathbf{x} & & \\ s.t. & \mathbf{A}\mathbf{x} & \leq & \mathbf{b} \\ & \mathbf{x} & \geq & \mathbf{0} \end{array}$$

- Lagrangian function:

$$L(\mathbf{x}, \lambda, \nu) = \mathbf{c}^{\mathbf{T}}\mathbf{x} - \sum_{i=1}^{m} \lambda_i(a_{i1}x_1 + \ldots + a_{in}x_n - b_i) - \sum_{i=1}^{n} \nu_i x_i$$

where $\lambda \leq \mathbf{0}$, $\nu \geq \mathbf{0}$.

- Notice that for any feasible solution $\mathbf{x}$ and $\lambda \leq \mathbf{0}$, $\nu \geq \mathbf{0}$, **Lagrangian function is a lower bound of the primal objective function**, i.e. $\mathbf{c}^{\mathbf{T}}\mathbf{x} \geq L(\mathbf{x}, \lambda, \nu)$, and further

$$\mathbf{c}^{\mathbf{T}}\mathbf{x} \geq L(\mathbf{x}, \lambda) \geq \inf_{\mathbf{x} \in \mathcal{D}} L(\mathbf{x}, \lambda)$$

- Let's define **Lagrangian dual function** $g(\lambda, \nu) = \inf_{\mathbf{x} \in \mathcal{D}} L(\mathbf{x}, \lambda, \nu)$ and rewrite the above inequality as

$$\mathbf{c}^{\mathbf{T}}\mathbf{x} \geq L(\mathbf{x}, \lambda, \nu) \geq g(\lambda, \nu)$$

Lagrangian dual function

- What is the Lagrangian dual function $g(\lambda, \nu)$?

$$
\begin{aligned}
g(\lambda, \nu) &= \inf_{\mathbf{x} \in \mathcal{D}} L(\mathbf{x}, \lambda, \nu) \\
&= \inf_{\mathbf{x} \in \mathcal{D}} (\mathbf{c}^{\mathbf{T}}\mathbf{x} - \sum_{i=1}^{m} \lambda_i (a_{i1}x_1 + ... + a_{in}x_n - b_i) - \sum_{i=1}^{m} \nu_i x_i) \\
&= \inf_{\mathbf{x} \in \mathcal{D}} (\lambda^{\mathbf{T}}\mathbf{b} + (\mathbf{c}^{\mathbf{T}} - \lambda^{\mathbf{T}}\mathbf{A} - \nu^{\mathbf{T}})\mathbf{x}) \\
&= \begin{cases} \lambda^{\mathbf{T}}\mathbf{b} & \text{if } \mathbf{c}^{\mathbf{T}} = \lambda^{\mathbf{T}}\mathbf{A} + \nu^{\mathbf{T}} \\ -\infty & \text{otherwise} \end{cases}
\end{aligned}
$$

- Note that $\mathcal{D} = \mathbb{R}^n$. Thus $g(\lambda, \nu) = \lambda^{\mathbf{T}}\mathbf{b}$ if $\mathbf{c}^{\mathbf{T}} = \lambda^{\mathbf{T}}\mathbf{A} + \nu^{\mathbf{T}}$; otherwise, $g(\lambda, \nu) = -\infty$, which is a trivial lower bound for the primal objective function $\mathbf{c}^{\mathbf{T}}\mathbf{x}$.
- We usually denote the domain of $g(\lambda, \nu)$ as $\mathbf{dom}\ g = \{(\lambda, \nu) | g(\lambda, \nu) > -\infty\}$.

Lagrangian dual problem

- Now let's try to find **the tightest lower bound** of the primal objective function $\mathbf{c}^{\mathbf{T}}\mathbf{x}$, which can be calculated by solving the following **Lagrangian dual problem**:

$$\begin{array}{rrcl} \max & g(\lambda, \nu) & = & \begin{cases} \lambda^{\mathbf{T}}\mathbf{b} & \text{if } \mathbf{c}^{\mathbf{T}} = \lambda^{\mathbf{T}}\mathbf{A} + \nu^{\mathbf{T}} \\ -\infty & \text{otherwise} \end{cases} \\ s.t. & \lambda & \geq & \mathbf{0} \\ & \nu & \geq & \mathbf{0} \end{array}$$

, or explicitly representing constraints in $\mathbf{dom}\ g$:

$$\begin{array}{rrcl} \max & \lambda^{\mathbf{T}}\mathbf{b} & & \\ s.t. & \lambda^{\mathbf{T}}\mathbf{A} & \leq & \mathbf{c}^{\mathbf{T}} \\ & \lambda & \geq & \mathbf{0} \end{array}$$

- Note that this is actually the $\textsc{Dual}$ form of LP if replacing λ by $\mathbf{y}$; thus, we have another explanation of $\textsc{Dual}$ variables $\mathbf{y}$ — the Lagrangian multiplier.

An example

- Primal problem:

$$\begin{array}{lll} \min & x & \\ s.t. & x & \geq & 2 \\ & x & \geq & 0 \end{array}$$

- Lagrangian function:

$$L(x, \lambda, \nu) = x - \lambda(x - 2) - \nu x = \lambda + (1 - \lambda - \nu)x$$

- Note that when $\lambda \geq 0$, $\nu \geq 0$ and $x \geq 2$, $L(x, \lambda, \nu)$ is a lower bound of the primal objective function x.
- Lagrangian dual function:

$$g(\lambda, \nu) = \inf_{\mathbf{x} \in \mathcal{D}} L(x, \lambda, \nu) = \begin{cases} 2\lambda & \text{if } 1 - \lambda - \nu = 0 \\ -\infty & \text{otherwise} \end{cases}$$

- Dual problem:

$$\begin{array}{ll} \max & 2\lambda \\ s.t. & \lambda \leq 1 \\ & \lambda \geq 0 \end{array}$$

Deriving dual problem of linear program in slack form

Dual problem of LP problem in slack form

- Consider a LP problem in slack form:

$$\begin{array}{lrcl} \min & \mathbf{c}^{\mathbf{T}}\mathbf{x} & & \\ s.t. & \mathbf{A}\mathbf{x} & = & \mathbf{b} \\ & \mathbf{x} & \geq & \mathbf{0} \end{array}$$

- Lagrangian function:

$$L(\mathbf{x}, \lambda, \mathbf{s}) = \mathbf{c}^{\mathbf{T}}\mathbf{x} - \sum_{i=1}^{m} \lambda_i (a_{i1}x_1 + \ldots + a_{in}x_n - b_i) - \sum_{i=1}^{n} \nu_i x_i$$

- Notice that for any feasible solution $\mathbf{x}$ and $\nu \geq \mathbf{0}$, **Lagrangian function is a lower bound of the primal objective function**, i.e. $\mathbf{c}^{\mathbf{T}}\mathbf{x} \geq L(\mathbf{x}, \lambda, \nu)$, and further

$$\mathbf{c}^{\mathbf{T}}\mathbf{x} \geq L(\mathbf{x}, \lambda) \geq \inf_{\mathbf{x} \in \mathcal{D}} L(\mathbf{x}, \lambda)$$

- Let's define **Lagrangian dual function** $g(\lambda, \nu) = \inf_{\mathbf{x} \in \mathcal{D}} L(\mathbf{x}, \lambda, \nu)$ and rewrite the above inequality as

$$\mathbf{c}^{\mathbf{T}}\mathbf{x} \geq L(\mathbf{x}, \lambda, \nu) \geq g(\lambda, \nu)$$

Lagrangian dual function

- What is the Lagrangian dual function $g(\lambda, \nu)$?

$$
\begin{aligned}
g(\lambda,\nu) &= \inf_{\mathbf{x}\in\mathcal{D}} L(\mathbf{x},\lambda,\nu) \\
&= \inf_{\mathbf{x}\in\mathcal{D}} (\mathbf{c}^{\mathbf{T}}\mathbf{x} - \sum_{i=1}^{m} \lambda_i (a_{i1}x_1 + \ldots + a_{in}x_n - b_i) - \sum_{i=1}^{m} \nu_i x_i) \\
&= \inf_{\mathbf{x}\in\mathcal{D}} (\lambda^{\mathbf{T}}\mathbf{b} + (\mathbf{c}^{\mathbf{T}} - \lambda^{\mathbf{T}}\mathbf{A} - \nu^{\mathbf{T}})\mathbf{x}) \\
&= \begin{cases} \lambda^{\mathbf{T}}\mathbf{b} & \text{if } \mathbf{c}^{\mathbf{T}} = \lambda^{\mathbf{T}}\mathbf{A} + \nu^{\mathbf{T}} \\ -\infty & \text{otherwise} \end{cases}
\end{aligned}
$$

- Note that $\mathcal{D} = \mathbb{R}^n$. Thus $g(\lambda,\nu) = \lambda^{\mathbf{T}}\mathbf{b}$ if $\mathbf{c}^{\mathbf{T}} = \lambda^{\mathbf{T}}\mathbf{A} + \nu^{\mathbf{T}}$; otherwise, $g(\lambda,\nu) = -\infty$, which is a trivial lower bound for the primal objective function $\mathbf{c}^{\mathbf{T}}\mathbf{x}$.

Lagrangian dual problem

- Now let's try to find **the tightest lower bound** of the primal objective function $\mathbf{c}^\mathbf{T}\mathbf{x}$, which can be calculated by solving the following **Lagrangian dual problem**:

$$\begin{array}{rrcl} \max & g(\lambda, \nu) & = & \begin{cases} \lambda^\mathbf{T}\mathbf{b} & \text{if } \mathbf{c}^\mathbf{T} = \lambda^\mathbf{T}\mathbf{A} + \nu^\mathbf{T} \\ -\infty & \text{otherwise} \end{cases} \\ s.t. & \nu & \geq & \mathbf{0} \end{array}$$

, or explicitly representing constraints in $\mathbf{dom}\ g$:

$$\begin{array}{rrcl} \max & \lambda^\mathbf{T}\mathbf{b} & & \\ s.t. & \lambda^\mathbf{T}\mathbf{A} & \leq & \mathbf{c}^\mathbf{T} \end{array}$$

Two explanations of dual variables $\mathbf{y}$: L. Kantorovich vs. T. Koopmans

1. Price interpretation: constrained optimization plays an important role in economics. Dual variables are also called as **shadow price** (by T. Koopmans), i.e. the instantaneous change in the optimization objective function when constraints are relaxed, or **marginal cost** when strengthening constraints.
2. **Lagrangian multiplier**: the effect of constraints on the objective function (by L. Kantorovich). For example, when b_i increase to $b_i + \Delta b_i$, how much the objective function value will change. In fact, we have $\frac{\partial L(\mathbf{x},\lambda)}{\partial b_i} = \lambda_i$.

Explanation of dual variables y: using DIET as an example

- Optimal solution to primal problem with $b_1 = 2000, b_2 = 55, b_3 = 800$:
$$\mathbf{x} = (14.24, 2.70, 0, 0),$$
$$\mathbf{c^T x} = 67.096.$$
- Optimal solution to dual problem:
$$\mathbf{y} = (0.0269, 0, 0.0164),$$
$$\mathbf{y^T b} = 67.096.$$
- Let's make a slight change on $\mathbf{b}$, and watch the effect on $\min \mathbf{c^T x}$.
  1. $\mathbf{b} = (2001, 55, 800)$: $\min \mathbf{c^T x} = 67.123$ (Note that $\mathbf{y_1} = 0.0269 = 67.123 - 67.096$)
  2. $\mathbf{b} = (2000, 56, 800)$: $\min \mathbf{c^T x} = 67.096$ (Note that $\mathbf{y_2} = 0 = 67.096 - 67.096$)
  3. $\mathbf{b} = (2000, 55, 801)$: $\min \mathbf{c^T x} = 67.112$ (Note that $\mathbf{y_3} = 0.0164 = 67.112 - 67.096$)

Dual problem is always convex

- Note that the Lagrangian dual function $g(\lambda, \nu)$ is a point-wise minimum of affine functions over λ, ν.

$$
\begin{aligned}
g(\lambda, \nu) &= \inf_{\mathbf{x} \in \mathcal{D}} L(\mathbf{x}, \lambda, \nu) \\
&= \inf_{\mathbf{x} \in \mathcal{D}} \left(f_0(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) - \sum_{i=1}^{m} \nu_i h_i(\mathbf{x})\right)
\end{aligned}
$$

- Thus Lagrangian dual function $g(\lambda, \nu)$ is always concave and **the dual problem is always a convex programming problem** even if the primal problem is non-convex.

Note: The dual is not intrinsic

- It is worthy pointed out that the dual problem and its optimal objective value are not properties of the primal feasible set and primal objective function alone.
- They also depend on the specific constraints in the primal problem.
- Thus we can construct equivalent primal optimization problem with different duals through the following ways:
    - Replacing primal objective function $f_0(x)$ with $h(f_0(x))$ where $h(u)$ is monotonically increasing.
    - Introducing new variables.
    - Adding redundant constraints.

Property of Lagrangian dual problem: Weak duality

Weak duality

- Let's p^* and d^* denote the optimal objective value of a primal problem and its dual problem, respectively. We always have

$$d^* \leq p^*$$

  regardless of convexity of the primal problem. The difference $p^* - d^*$ is called **duality gap**.
- Weak duality holds even if $p^* = -\infty$, which means the infeasibility of the dual problem. Similarly, if $d^* = +\infty$, the primal problem is infeasible.
- As dual problems are always convex, it is relatively easy to calculate d^* efficiently, which provides a lower bound for p^*.

An example of non-zero duality gap

- Consider the following non-convex optimization problem.

$$\begin{array}{lrcl} \min & x_1 x_2 & & \\ s.t. & x_1 & \geq & 0 \\ & x_2 & \geq & 0 \\ & x_1^2 + x_2^2 & \leq & 1 \end{array}$$

- Lagrange dual function:

$$g(\lambda) = \inf_{\mathbf{x} \in \mathcal{D}} (x_1 x_2 - \lambda_1 x1 - \lambda_2 x_2 + \lambda_3 (x_1^2 + x_2^2 - 1))$$

- Dual problem:

$$\begin{array}{lrcl} \max & g(\lambda) & & \\ s.t. & \lambda_1 & \geq & 0 \\ & \lambda_2 & \geq & 0 \\ & \lambda_3 & \geq & \frac{1}{2} \end{array}$$

- Duality gap: $p* = 0 > d^* = -\frac{1}{2}$.

Property of Lagrangian dual problem: Strong duality

Strong duality

- Strong duality holds if $p^* = d^*$, i.e., the duality gap is 0.
- Strong duality doesn't necessarily hold for any optimization problem, but it almost always holds for convex ones, i.e.

$$\begin{array}{rrcll} \min & f_0(\mathbf{x}) & & & \\ s.t. & f_i(\mathbf{x}) & \leq & 0 & i = 1, 2, ..., m \\ & \mathbf{Ax} & = & \mathbf{b} & \end{array}$$

  where $f_i(\mathbf{x})$ $(i = 0, 1, ..., m)$ are convex functions.
- The conditions that guarantee strong duality are called **regularity conditions**, one of which is the Slater's condition.
- Slater's condition: Consider a convex optimization problem. The strong duality holds if there exists a vector $\mathbf{x} \in \textbf{relint } \mathcal{D}$ such that

$$f_i(\mathbf{x}) < 0, i = 1, ..., m, \quad \mathbf{Ax} = \mathbf{b}$$

Slater's condition

- Suppose the first k constraints are affine, then the Slater's conditions turns into: there exists a vector $\mathbf{x} \in \mathbf{relint}\mathcal{D}$ such that

$$f_i(\mathbf{x}) \leq 0, i = 1, ..., k, \quad f_i(\mathbf{x}) < 0, i = 1 + 1, ..., m, \quad \mathbf{A}\mathbf{x} = \mathbf{b}$$

- Specifically, if all constraints are affine, then the original constraints are themselves the Slater's conditions.

Conditions of optimal solution: KKT conditions

Three types of optimization problems

- It is relatively easy to optimize an objective function **without any constraint**, say:

$$\min \quad f_0(\mathbf{x})$$

- But how to optimize an objective function **with equality constraints**?

$$\begin{array}{llcll} \min & f_0(\mathbf{x}) & & & \\ s.t. & h_i(\mathbf{x}) & = & 0 & i = 1, 2, ..., p \end{array}$$

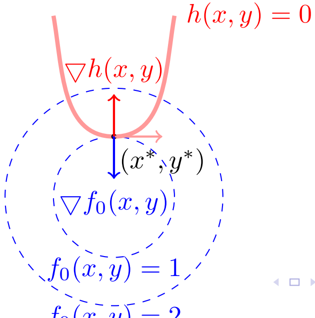
- And how to optimize an objective function **with inequality constraints**?

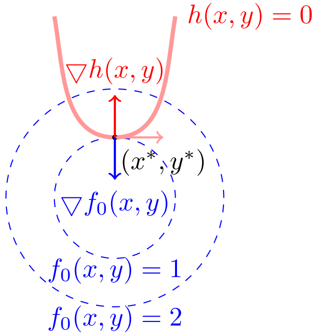
$$\begin{array}{llcll} \min & f_0(\mathbf{x}) & & & \\ s.t. & f_i(\mathbf{x}) & \leq & 0 & i = 1, 2, ..., m \\ & h_i(\mathbf{x}) & = & 0 & i = 1, 2, ..., p \end{array}$$

- Consider the following optimization problem:

$$\begin{array}{rrcl} \min & f_0(x,y) & & \\ s.t. & h(x,y) & = & 0 \end{array}$$

- Intuition: suppose (x^*, y^*) is the optimum point. Thus at (x^*, y^*), $f_0(x,y)$ does not change when walking along the curve $h(x,y) = 0$; otherwise, we can follow the curve to make $f_0(x,y)$ smaller, meaning that the starting point (x^*, y^*) is not optimum.





- So at (x^*, y^*), the red line tangentially touches a blue contour, i.e. there exists a real λ such that:
$$\bigtriangledown f_0(x,y) = \lambda \bigtriangledown h(x,y)$$
- Lagrange must have cleverly noticed that the equation above looks like partial derivatives of some larger scalar function:
$$L(x,y,\lambda) = f_0(x,y) - \lambda h(x,y)$$
- Necessary conditions of optimum point: If (x^*, y^*) is local optimum, then there exists a λ such that $\bigtriangledown L(x^*, y^*, \lambda) = 0$.

Understanding Lagrangian function

- Lagrangian function: a combination of the original optimization objective function and constraints:

$$L(x, y, \lambda) = f_0(x, y) - \lambda h(x, y)$$

- The critical point of Lagrangian $L(x, y, \lambda)$ occurs at saddle points rather than local minima (or maxima). To utilize numerical optimization techniques, we must first transform the problem such that the critical points lie at local minima. This is done by calculating the magnitude of the gradient of Lagrangian.

KKT conditions: under inequality constraints

- Consider the following optimization problem:

$$\begin{array}{lll} \min & f_0(x,y) & \\ s.t. & f_i(x,y) \leq 0 & i=1,2,...m \end{array}$$

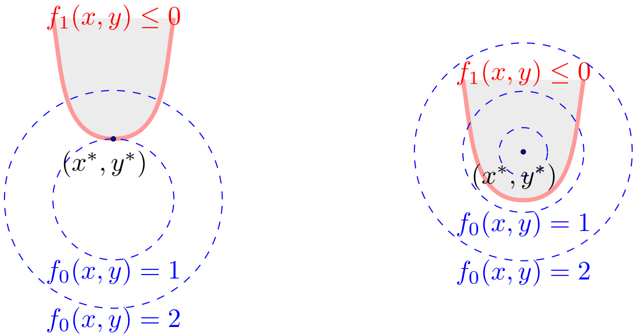


Figure: Case 1: the optimum point lies in the curve $f_1(x,y) = 0$. Thus Lagrangian condition $\bigtriangledown L(x,y,\lambda) = 0$ applies. Case 2: the optimum point lies within the region $f_1(x,y) \leq 0$; thus we have $\bigtriangledown f_0(x,y) = 0$ at (x^*, y^*)

Complementary slackness

- These two cases can be summarized as the following two conditions:
  - (Stationary point) $\bigtriangledown L(\mathbf{x}^*, \lambda) = \mathbf{0}$
  - (Complementary slackness) $\lambda_i f_i(\mathbf{x}^*) = 0$, $i = 1, 2, ..., m$
- Reason: In case 2, $f_i(\mathbf{x}^*) < 0 \Rightarrow \lambda = 0 \Rightarrow \bigtriangledown f_0(\mathbf{x}^*) = 0$.
- Complementary slackness, also called **orthogonality** by Gomory, essentially equals to the strong duality for convex optimization problems.
- A relaxation of this condition, i.e., $\lambda_i g_i(\mathbf{x}^*) = \mu$, is used in the interior point method to reflect duality gap.

KKT conditions

- Lagrangian:

$$L(\mathbf{x}, \lambda) = f_0(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x})$$

- Necessary conditions of optimum point:
  If $\mathbf{x}^*$ is local optimum and the primal problem satisfies some regularity conditions (say Slater's conditions), then there exists λ such that:
  1. (Stationary point) $\bigtriangledown L(\mathbf{x}^*, \lambda) = \mathbf{0}$
  2. (Primal feasibility) $f_i(\mathbf{x}^*) \leq 0$
  3. (Dual feasibility) $\lambda_i \leq 0$, $i = 1, 2, ..., m$
  4. (Complementary slackness) $\lambda_i f_i(\mathbf{x}^*) = 0$, $i = 1, 2, ..., m$
- KKT conditions are usually not solved directly in optimization; instead, iterative successive approximation is most often used to find the final results that satisfy KKT conditions.

1

[1] KKT conditions were named after William Karush, Harold W. Kuhn, and Albert W. Tucker.

KKT conditions for non-convex problems

- Let $\mathbf{x}^*$ and (λ, ν) denote the optimal solutions to the primal and dual problems, respectively. Suppose the strong duality holds.
- Since Lagrangian function reaches its minimum at $\mathbf{x}^*$, its gradient is 0 at $\mathbf{x}^*$, i.e.,

$$\bigtriangledown f_0(\mathbf{x}^*) - \sum_{i=1}^{m} \lambda_i \bigtriangledown f_i(\mathbf{x}^*) - \sum_{i=1}^{p} \nu_i \bigtriangledown h_i(\mathbf{x}^*) = 0$$

- Then $\mathbf{x}^*$ and (λ, ν) satisfy the following KKT conditions:
  1. (Stationary point) $\bigtriangledown L(\mathbf{x}^*, \lambda) = \mathbf{0}$
  2. (Primal feasibility) $f_i(\mathbf{x}^*) \leq 0$, $i = 1, 2, ..., m$
  3. (Dual feasibility) $\lambda_i \leq 0$, $i = 1, 2, ..., m$
  4. (Complementary slackness) $\lambda_i f_i(\mathbf{x}^*) = 0$, $i = 1, 2, ..., m$

KKT conditions for convex problems

- Consider a convex problem. If $\mathbf{x}^*$ and (λ, ν) satisfy the following KKT conditions:

  1. (Stationary point) $\bigtriangledown L(\mathbf{x}^*, \lambda) = \mathbf{0}$
  2. (Primal feasibility) $f_i(\mathbf{x}^*) \leq 0$, $i = 1, 2, ..., m$
  3. (Dual feasibility) $\lambda_i \leq 0$, $i = 1, 2, ..., m$
  4. (Complementary slackness) $\lambda_i f_i(\mathbf{x}^*) = 0$, $i = 1, 2, ..., m$

  then x^* and (λ, ν) are optimal solutions to the primal and dual problems, respectively.

Four properties of duality for linear program

Property 1: Primal is the dual of dual

Theorem

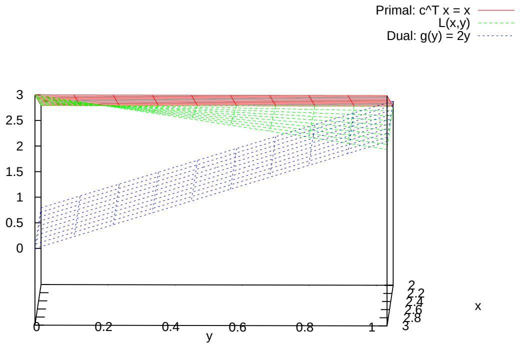
For linear program, primal problem is the dual of dual.

- For a general optimization problem, the dual of dual is not always the primal problem but a convex relaxation of the primal problem.

Property 2: Weak duality

Theorem

(Weak duality) The objective value of **any** *feasible solution to the dual problem is always a lower bound of the objective value of primal problem.*



An example: DIET problem and its dual problem

- Primal problem P:

$$
\begin{array}{rrcrcrcrll}
\min & 3x_1 & + & 9x_2 & + & 20x_3 & + & 19x_4 & & \text{money} \\
s.t. & 110x_1 & + & 160x_2 & + & 420x_3 & + & 260x_4 & \geq 2000 & \text{energy} \\
 & 4x_1 & + & 8x_2 & + & 4x_3 & + & 14x_4 & \geq 55 & \text{protein} \\
 & 2x_1 & + & 285x_2 & + & 22x_3 & + & 80x_4 & \geq 800 & \text{calcium} \\
 & x_1 & , & x_2 & , & x_3 & , & x_4 & \geq 0 &
\end{array}
$$

  Feasible solution $\mathbf{x}^{\mathbf{T}} = [0, 8, 2, 0]^T \Rightarrow \mathbf{c}^{\mathbf{T}}\mathbf{x} = 112$.

- Dual problem D:

$$
\begin{array}{rrcrcrll}
\max & 2000y_1 & + & 55y_2 & + & 800y_3 & & \text{money} \\
s.t. & 110y_1 & + & 4y_2 & + & 2y_3 & \leq 3 & \text{oatmeal} \\
 & 160y_1 & + & 8y_2 & + & 285y_3 & \leq 9 & \text{milk} \\
 & 420y_1 & + & 4y_2 & + & 22y_3 & \leq 20 & \text{pie} \\
 & 260y_1 & + & 14y_2 & + & 80y_3 & \leq 19 & \text{pork\&beans} \\
 & y_1 & , & y_2 & , & y_3 & \geq 0 &
\end{array}
$$

  Feasible solution $\mathbf{y}^{\mathbf{T}} = [0.0269, 0, 0.0164]^T \Rightarrow \mathbf{y}^{\mathbf{T}}\mathbf{b} = 67.096$

- The theorem states that $\mathbf{c}^{\mathbf{T}}\mathbf{x} \geq \mathbf{y}^{\mathbf{T}}\mathbf{b}$ for **any** feasible solutions $\mathbf{x}$ and $\mathbf{y}$.

Proof.

- Consider the following PRIMAL problem:

$$\begin{array}{lrl} \min & \mathbf{c^T x} & \\ s.t. & \mathbf{Ax} & \geq \mathbf{b} \\ & \mathbf{x} & \geq \mathbf{0} \end{array}$$

  and DUAL problem:

$$\begin{array}{lrl} \max & \mathbf{b^T y} & \\ s.t. & \mathbf{y} & \geq \mathbf{0} \\ & \mathbf{A^T y} & \leq \mathbf{c} \end{array}$$

- Let $\mathbf{x}$ and $\mathbf{y}$ denote a feasible solution to primal and dual problems, respectively.
- We have $\mathbf{c^T x} \geq \mathbf{y^T A x}$ (by the feasibility of dual problem, i.e., $\mathbf{y^T A} \leq \mathbf{c^T}$, and $\mathbf{x^T} \geq \mathbf{0}$)
- Therefore $\mathbf{c^T x} \geq \mathbf{y^T A x} \geq \mathbf{y^T b}$ (by the feasibility of primal problem, i.e., $\mathbf{Ax} \geq \mathbf{b}$, and $\mathbf{y} \geq \mathbf{0}$)

Property 3: Strong duality

Theorem

(Strong duality) Consider a linear program. If the primal problem has an optimal solution, then the dual problem also has an optimal solution with the same objective value.

Proof.

- Suppose $\mathbf{x}^* = \begin{bmatrix} \mathbf{B}^{-1}\mathbf{b} \\ 0 \end{bmatrix}$ be the optimal solution to the primal problem. We have $\mathbf{c}^\mathbf{T} - \mathbf{c}_\mathbf{B}^\mathbf{T}\mathbf{B}^{-1}\mathbf{A} \geq \mathbf{0}$.
- Define $\mathbf{y}^{*\mathbf{T}} = \mathbf{c_B}^\mathbf{T}\mathbf{B}^{-1}$. We will show that $\mathbf{y}^{*\mathbf{T}}$ is the optimal solution to the dual problem.
- In fact, we have $\mathbf{y}^{*\mathbf{T}}\mathbf{b} = \mathbf{c_B}^\mathbf{T}\mathbf{B}^{-1}\mathbf{b} = \mathbf{c}^\mathbf{T}\mathbf{x}^*$.
- That is, $\mathbf{y}^{*\mathbf{T}}\mathbf{b}$ reaches its upper bound. So $\mathbf{y}^{*\mathbf{T}}$ is an optimal solution to the dual problem.

$\square$

Property 4: Complementary slackness

Theorem

Let $\mathbf{x}$ and $\mathbf{y}$ denote feasible solutions to the primal and dual problems, respectively. Then $\mathbf{x}$ and $\mathbf{y}$ are optimal solutions iff $u_i = y_i(a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n - b_i) = 0$ *for any* $1 \leq i \leq m$, *and* $v_j = (c_j - a_{1j}y_1 - a_{2j}y_2 - ... - a_{mj}y_m)x_j = 0$ *for any* $1 \leq j \leq n$.

- Intuition: a constraint of primal problem is loosely restricted $\Rightarrow$ the corresponding dual variable is tight.
- An example: the optimal solutions to Diet and its dual are $\mathbf{x} = (14.244, 2.707, 0, 0)$ and $\mathbf{y} = (0.0269, 0, 0.0164)$.

$$\begin{array}{rcrcrcrl}
110x_1 & + & 160x_2 & + & 420x_3 & + & 260x_4 & = 2000 \\
4x_1 & + & 8x_2 & + & 4x_3 & + & 14x_4 & > 55 \qquad \Rightarrow y_2 = 0 \\
2x_1 & + & 285x_2 & + & 22x_3 & + & 80x_4 & = 800 \\
x_1 & , & x_2 & , & x_3 & , & x_4 & \geq 0
\end{array}$$

Proof.

$u_i = 0$ and $v_j = 0$ for any i and j
$\Leftrightarrow \sum_i u_i = 0$ and $\sum_j v_j = 0$ (since $u_i \geq 0, v_j \geq 0$)
$\Leftrightarrow \sum_i u_i + \sum_j v_j = 0$
$\Leftrightarrow (\mathbf{y^T A x} - \mathbf{y^T b}) + (\mathbf{c^T x} - \mathbf{y^T A x}) = \mathbf{0}$
$\Leftrightarrow \mathbf{y^T b} = \mathbf{c^T x}$
$\Leftrightarrow \mathbf{y}$ and $\mathbf{x}$ are optimal solutions (by strong duality property, i.e., both $\mathbf{y^T b}$ and $\mathbf{c^T x}$ reach its bound)

$\square$

Summary: 9 cases of primal and dual problems

Dual \ Primal	Bounded Optimal Objective Value	Unbounded Optimal Objective Value	Infeasible
Bounded Optimal Objective Value	Possible	Impossible	Impossible
Unbounded Optimal Objective Value	Impossible	Impossible	Possible
Infeasible	Impossible	Possible	Possible

Example 1: PRIMAL has unbounded objective value and DUAL is infeasible

PRIMAL:

$$\begin{array}{rrcrcr} \min & -2x_1 & - & 2x_2 & & \\ s.t. & x_1 & - & x_2 & \leq & 1 \\ & -x_1 & + & x_2 & \leq & 1 \\ & x_1 & & & \geq & 0 \\ & & & x_2 & \geq & 0 \end{array}$$

DUAL:

$$\begin{array}{rrcrcr} \max & y_1 & + & y_2 & & \\ s.t. & y_1 & & & \leq & 0 \\ & & & y_2 & \leq & 0 \\ & y_1 & - & y_2 & \leq & -2 \\ & -y_1 & + & y_2 & \leq & -2 \end{array}$$

Example 2: both PRIMAL and DUAL are infeasible

PRIMAL:

$$
\begin{array}{lrcrcr}
\min & x_1 & - & 2x_2 & & \\
s.t. & x_1 & - & x_2 & \geq & 2 \\
 & -x_1 & + & x_2 & \geq & -1 \\
 & x_1 & & & \geq & 0 \\
 & & & x_2 & \geq & 0
\end{array}
$$

DUAL:

$$
\begin{array}{lrcrcr}
\max & 2y_1 & - & y_2 & & \\
s.t. & y_1 & & & \geq & 0 \\
 & & & y_2 & \geq & 0 \\
 & y_1 & - & y_2 & \leq & 1 \\
 & -y_1 & + & y_2 & \leq & -2
\end{array}
$$

Solving linear program using duality

KKT conditions for linear program

- For a linear program in slack form, the KKT conditions turns into:
  $\mathbf{x}$ and $\mathbf{y}$ are optimal solutions to the primal and dual problems, respectively, if they satisfy the following three conditions:
  1. (Primal feasibility)
  $$\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$$
  2. (Dual feasibility)
  $$\mathbf{y}^{\mathbf{T}}\mathbf{A} \leq \mathbf{c}^{\mathbf{T}}$$
  3. (Complementary slackness)
  $$\mathbf{c}^{\mathbf{T}}\mathbf{x} = \mathbf{y}^{\mathbf{T}}\mathbf{b}$$
- Question: How to obtain $\mathbf{x}$ and $\mathbf{y}$ that satisfy these three conditions simultaneously?

Strategy 1

$\mathbf{x} = \mathbf{x_0}$; //Initialize $\mathbf{x}$ with a primal feasible solution
$\mathbf{y} = \mathbf{y_0}$; //Calcualte initial $\mathbf{y}$ according to complementary slackness
while TRUE **do**
  $\mathbf{x} = \text{IMPROVE}(\mathbf{x})$; //Improve $\mathbf{x}$ and update $\mathbf{y}$ accordingly
  if $\text{STOPPING}(\mathbf{x})$ **then**
    break;
    //If $\mathbf{y}$ is dual feasible then stop
  end if
end while
return $\mathbf{x}$;

- Example: primal simplex

Strategy 2

$\mathbf{y} = \mathbf{y_0}$; //Initialize $\mathbf{y}$ with a dual feasible solution
$\mathbf{x} = \mathbf{x_0}$; //Calcualte initial $\mathbf{x}$ according to complementary slackness
while TRUE **do**
    $\mathbf{y} = \textsc{Improve}(\mathbf{y})$; //Improve $\mathbf{y}$ and update $\mathbf{x}$ accordingly
    if $\textsc{stopping}(\mathbf{y})$ **then**
        break;
        //If $\mathbf{x}$ is primal feasible then stop
    end if
end while
return $\mathbf{y}$;

- Example: dual simplex, primal and dual

Strategy 3

$\mathbf{x} = \mathbf{x_0}$; //Initialize $\mathbf{x}$ with a primal feasible solution
$\mathbf{y} = \mathbf{y_0}$; //Initialize $\mathbf{y}$ with a dual feasible solution
while TRUE **do**
    $(\mathbf{x}, \mathbf{y}) = \text{IMPROVE}(\mathbf{x}, \mathbf{y})$; //Improve $\mathbf{x}$ and $\mathbf{y}$
    if $\text{STOPPING}(\mathbf{x}, \mathbf{y})$ **then**
        break;
        //If $\mathbf{x}$ and $\mathbf{y}$ satisfy complementary slackness then stop
    end if
end while
return $\mathbf{y}$;

- Example: interior point method

Dual simplex method

Revisiting PRIMAL SIMPLEX algorithm

- Consider the following PRIMAL problem **P**:

$$\begin{array}{rrcrcrcr} \min & x_1 & + & 14x_2 & + & 6x_3 & & \\ s.t. & x_1 & + & x_2 & + & x_3 & \leq & 4 \\ & x_1 & & & & & \leq & 2 \\ & & & & & x_3 & \leq & 3 \\ & & & 3x_2 & + & x_3 & \leq & 6 \\ & x_1 & , & x_2 & , & x_3 & \geq & 0 \end{array}$$

- PRIMAL simplex tabular:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
-z= 0	$\overline{c_1}$= 1	$\overline{c_2}$=14	$\overline{c_3}$=6	$\overline{c_4}$=0	$\overline{c_5}$=0	$\overline{c_6}$=0	$\overline{c_7}$=0
$\mathbf{x_{B1}} = b'_1$=4	1	1	1	1	0	0	0
$\mathbf{x_{B2}} = b'_2$=2	1	0	0	0	1	0	0
$\mathbf{x_{B3}} = b'_3$=3	0	0	1	0	0	1	0
$\mathbf{x_{B4}} = b'_4$=6	0	3	1	0	0	0	1

- Primal variables: $\mathbf{x}$; Feasible: $\mathbf{B^{-1}b} \geq \mathbf{0}$.
- A basis $\mathbf{B}$ is called **primal feasible** if all elements in $\mathbf{B^{-1}b}$ (the first column except for $-z$) are non-negative.

- Now let's consider the DUAL problem **D**:

$$\begin{array}{rccccccccc}
\max & 4y_1 & + & 2y_2 & + & 3y_3 & + & 6y_4 & & \\
s.t. & y_1 & + & y_2 & & & & & \leq & 1 \\
 & y_1 & & & & & + & 3y_4 & \leq & 14 \\
 & y_1 & & & + & y_3 & + & y_4 & \leq & 6 \\
 & y_1 & , & y_2 & , & y_3 & , & y_4 & \leq & 0
\end{array}$$

- PRIMAL simplex tabular:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
-z= 0	$\overline{c_1}$= 1	$\overline{c_2}$=14	$\overline{c_3}$=6	$\overline{c_4}$=0	$\overline{c_5}$=0	$\overline{c_6}$=0	$\overline{c_7}$=0
$\mathbf{x_{B1}} = b'_1$=4	1	1	1	1	0	0	0
$\mathbf{x_{B2}} = b'_2$=2	1	0	0	0	1	0	0
$\mathbf{x_{B3}} = b'_3$=3	0	0	1	0	0	1	0
$\mathbf{x_{B4}} = b'_4$=6	0	3	1	0	0	0	1

- Dual variables: $\mathbf{y^T} = \mathbf{c_B^T B^{-1}}$; Feasible: $\mathbf{y^T A} \leq \mathbf{c^T}$.
- A basis $\mathbf{B}$ is called **dual feasible** if all elements in $\overline{\mathbf{c^T}} = \mathbf{c} - \mathbf{c_B^T B^{-1} A} = \mathbf{c^T} - \mathbf{y^T A}$ (the first row except for $-z$) are non-negative.

Another view point of the PRIMAL SIMPLEX algorithm

- Thus another view point of the PRIMAL SIMPLEX algorithm can be described as:
  1. **Starting point:** The PRIMAL SIMPLEX algorithm starts with a primal feasible solution ($\mathbf{x_B} = \mathbf{B^{-1}b} \geq \mathbf{0}$);
  2. **Maintenance:** Throughout the process we maintain the primal feasibility and move towards the dual feasibility;
  3. **Stopping criteria:** $\mathbf{\bar{c}^T} = \mathbf{c^T} - \mathbf{c_B^T B^{-1} A} \geq \mathbf{0}$, i.e., $\mathbf{y^T A} \leq \mathbf{c^T}$. In other words, the iteration process ends when the basis is both primal feasible and dual feasible.

- DUAL SIMPLEX:

  1. **Starting point:** The DUAL SIMPLEX algorithm starts with a dual feasible solution ($\overline{\mathbf{c}}^{\mathbf{T}} \geq \mathbf{0}$);
  2. **Maintenance:** Throughout the process we maintain the dual feasibility and move towards the dual feasibility;
  3. **Stopping criteria:** $\mathbf{x_B} = \mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$. In other words, the iteration process ends when the basis is both primal feasible and dual feasible.

- Both PRIMAL SIMPLEX and DUAL SIMPLEX terminate at the same condition, i.e. the basis is both primal feasible and dual feasible.
- However, the final objective is achieved in totally opposite fashions— the PRIMAL SIMPLEX method keeps the primal feasibility while the DUAL SIMPLEX method keeps the dual feasibility during the pivoting process.
- The PRIMAL SIMPLEX algorithm *first selects an entering variable and then determines the leaving variable*.
- In contrast, the DUAL SIMPLEX method does the opposite; it *first selects a leaving variable and then determines an entering variable*.

DUAL SIMPLEX($B_I, z, \mathbf{A}, \mathbf{b}, \mathbf{c}$)

```
//DUAL SIMPLEX starts with a dual feasible basis. Here, B_I contains the
indices of the basic variables.
while TRUE do
    if there is no index l (1 ≤ l ≤ m) has b_l < 0 then
        x = CALCULATEX(B_I, A, b, c);
        return (x, z);
    end if;
    choose an index l having b_l < 0 according to a certain rule;
    for each index j (1 ≤ i ≤ n) do
        if a_lj < 0 then
            Δ_j = -c_j / a_lj;
        else
            Δ_j = ∞;
        end if
    end for
    choose an index e that minimizes Δ_j;
    if Δ_e = ∞ then
        return ''no feasible solution'';
    end if
    (B_I, A, b, c, z) = PIVOT(B_I, A, b, c, z, e, l);
end while
```

An example

- Standard form:

$$\begin{array}{rrcrcrcr} \min & 5x_1 & + & 35x_2 & + & 20x_3 & & \\ s.t. & x_1 & - & x_2 & - & x_3 & \leq & -2 \\ & -x_1 & - & 3x_2 & & & \leq & -3 \\ & x_1 & , & x_2 & , & x_3 & \geq & 0 \end{array}$$

- Slack form:

$$\begin{array}{rrcrcrcrcrcr} \min & 5x_1 & + & 35x_2 & + & 20x_3 & & & & & & \\ s.t. & x_1 & - & x_2 & - & x_3 & + & x_4 & & & = & -2 \\ & -x_1 & - & 3x_2 & & & & & + & x_5 & = & -3 \\ & x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & \geq & 0 \end{array}$$

Step 1

	x_1	x_2	x_3	x_4	x_5
-z= 0	$\overline{c_1}$= 5	$\overline{c_2}$=35	$\overline{c_3}$=20	$\overline{c_4}$=0	$\overline{c_5}$=0
$\mathbf{x_{B1}} = b_1'$=-2	1	-1	-1	1	0
$\mathbf{x_{B2}} = b_2'$=-3	-1	-3	0	0	1

- Basis (in blue): $\mathbf{B} = \{\mathbf{a_4}, \mathbf{a_5}\}$
- Solution: $\mathbf{x} = \begin{bmatrix} \mathbf{B^{-1}b} \\ \mathbf{0} \end{bmatrix} = [0, 0, 0, -2, -3]^T$.
- Pivoting: choose $\mathbf{a_5}$ to leave basis since $b_2' = -3 < 0$; choose $\mathbf{a_1}$ to enter basis since $\min_{j, a_{2j}<0} \frac{\bar{c}_j}{-a_{2j}} = \frac{\bar{c}_1}{-a_{21}}$.

Step 2

	x_1	x_2	x_3	x_4	x_5
-z= -15	$\overline{c_1}$= 0	$\overline{c_2}$=20	$\overline{c_3}$=20	$\overline{c_4}$=0	$\overline{c_5}$=5
$\mathbf{x_{B1}} = b_1'$=-5	0	-4	-1	1	1
$\mathbf{x_{B2}} = b_2'$=3	1	3	0	0	-1

- Basis (in blue): $\mathbf{B} = \{\mathbf{a_1}, \mathbf{a_4}\}$
- Solution: $\mathbf{x} = \begin{bmatrix} \mathbf{B^{-1}b} \\ \mathbf{0} \end{bmatrix} = [3, 0, 0, -5, 0]^T$.
- Pivoting: choose $\mathbf{a_4}$ to leave basis since $b_1' = -5 < 0$; choose $\mathbf{a_2}$ to enter basis since $\min_{j, a_{1j}<0} \frac{\bar{c}_j}{-a_{1j}} = \frac{\bar{c}_2}{-a_{12}}$.

Step 3

	x_1	x_2	x_3	x_4	x_5
-z= -40	$\overline{c_1}=0$	$\overline{c_2}=0$	$\overline{c_3}=15$	$\overline{c_4}=5$	$\overline{c_5}=10$
$\mathbf{x_{B1}} = b'_1 = \frac{5}{4}$	0	1	$\frac{1}{4}$	$-\frac{1}{4}$	$-\frac{1}{4}$
$\mathbf{x_{B2}} = b'_2 = -\frac{3}{4}$	1	0	$-\frac{3}{4}$	$\frac{3}{4}$	$-\frac{1}{4}$

- Basis (in blue): $\mathbf{B} = \{\mathbf{a_1}, \mathbf{a_2}\}$
- Solution: $\mathbf{x} = \begin{bmatrix} \mathbf{B^{-1}b} \\ \mathbf{0} \end{bmatrix} = [\frac{5}{4}, -\frac{3}{4}, 0, 0, 0]^T$.
- Pivoting: choose $\mathbf{a_1}$ to leave basis since $b'_2 = -\frac{3}{4} < 0$; choose $\mathbf{a_3}$ to enter basis since $\min_{j, a_{2j}<0} \frac{\bar{c}_j}{-a_{2j}} = \frac{\bar{c}_3}{-a_{23}}$.

Step 4

	x_1	x_2	x_3	x_4	x_5
-z= -55	$\overline{c_1}$= 20	$\overline{c_2}$=0	$\overline{c_3}$=0	$\overline{c_4}$=20	$\overline{c_5}$=5
$\mathbf{x_{B1}} = b'_1$=1	$\frac{1}{3}$	1	0	0	$-\frac{1}{3}$
$\mathbf{x_{B2}} = b'_2$=1	$-\frac{4}{3}$	0	1	-1	$\frac{1}{3}$

- Basis (in blue): $\mathbf{B} = \{\mathbf{a_2}, \mathbf{a_3}\}$
- Solution: $\mathbf{x} = \begin{bmatrix} \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{0} \end{bmatrix} = [0, 1, 1, 0, 0]^T$.
- Done!

When dual simplex method is useful?

1. The dual simplex algorithm is most suited for problems for which an **initial dual feasible solution** is easily available.
2. It is particularly useful for **reoptimizing** a problem after a constraint has been added or some parameters have been changed so that the previously optimal basis is no longer feasible.
3. Trying dual simplex is particularly useful if your LP appears to be highly degenerate, i.e. there are many vertices of the feasible region for which the associated basis is degenerate. We may find that a large number of iterations (moves between adjacent vertices) occur with little or no improvement.[2]

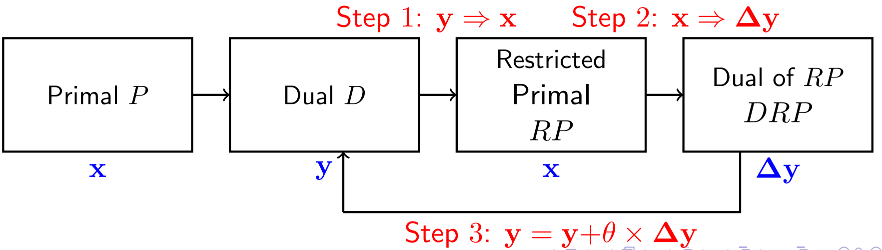
[2] References: Operations Research Models and Methods, Paul A. Jensen and Jonathan F. Bard; OR-Notes, J. E. Beasley

Primal_Dual: another IMPROVMENT approach

Primal_Dual method: a brief history

- In 1955, H. Kuhn proposed the Hungarian method for the MaxWeightedMatching problem. This method effectively explores the duality property of linear programming.
- In 1956, G. Dantzig, R. Ford, and D. Fulkerson extended this idea to solve linear programming problems.
- In 1957, R. Ford, and D. Fulkerson applied this idea to solve network-flow problem and Hitchcock problem.
- In 1957, J. Munkres applied this idea to solve the transportation problem.

- Primal_Dual method is a dual method, which exploits the lower bound information in subsequent linear programming operations.
- Advantages:
  1. Unlike dual simplex starting from a **dual basic feasible solution**, primal_dual method requires only a **dual feasible solution**.
  2. An optimal solution to DRP usually has combinatorial explanation, especially for graph-theory problems.



Basic idea of primal_dual method

- Primal P:

$$
\begin{array}{lccccccccccc}
\min & c_1x_1 & + & c_2x_2 & + & \ldots & + & c_nx_n & & & \\
s.t. & a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1n}x_n & = & b_1 & (y_1) \\
 & a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2n}x_n & = & b_2 & (y_2) \\
 & & & & & \ldots & & & & & \\
 & a_{m1}x_1 & + & a_{m2}x_2 & + & \ldots & + & a_{mn}x_n & = & b_m & (y_m) \\
 & x_1 & , & x_2 & , & \ldots & , & x_n & \geq & 0 &
\end{array}
$$

- Dual D:

$$
\begin{array}{lcccccccccc}
\max & b_1y_1 & + & b_2y_2 & + & \ldots & + & b_my_m & & \\
s.t. & a_{11}y_1 & + & a_{21}y_2 & + & \ldots & + & a_{m1}y_m & \leq & c_1 \\
 & a_{12}y_1 & + & a_{22}y_2 & + & \ldots & + & a_{m2}y_m & \leq & c_2 \\
 & & & & & \ldots & & & & \\
 & a_{1n}y_1 & + & a_{2n}y_2 & + & \ldots & + & a_{mn}y_m & \leq & c_n
\end{array}
$$

- Basic idea: Suppose we are given a dual feasible solution $\mathbf{y}$. Let's verify whether $\mathbf{y}$ is an optimal solution or not:
  1. If $\mathbf{y}$ is an optimal solution to the dual problem D, then the corresponding primal variables $\mathbf{x}$ should satisfy a restricted primal problem called RP;
  2. Furthermore, even if $\mathbf{y}$ is not optimal, the solution to the dual of RP (called DRP) still provide invaluable information — it can be used to improve $\mathbf{y}$.

Step 1: $\mathbf{y} \Rightarrow \mathbf{x}$ I

- Dual problem D:

$$\begin{array}{lccccccccccl}
\max & b_1y_1 & + & b_2y_2 & + & \ldots & + & b_my_m & & & \\
s.t. & a_{11}y_1 & + & a_{21}y_2 & + & \ldots & + & a_{m1}y_m & \leq & c_1 & \color{blue}{('=' \Rightarrow x_1 \geq 0)} \\
 & & & & & \ldots & & & & & \\
 & a_{1n}y_1 & + & a_{2n}y_2 & + & \ldots & + & a_{mn}y_m & \leq & c_n & \color{blue}{('<' \Rightarrow x_n = 0)}
\end{array}$$

- $\mathbf{y}$ provides information of the corresponding primal variables $\mathbf{x}$:
  1. Given a dual feasible solution $\mathbf{y}$. Let's check whether $\mathbf{y}$ is optimal solution or not.
  2. If $\mathbf{y}$ is optimal, we have the following restrictions on $\mathbf{x}$: $a_{1i}y_1 + a_{2i}y_2 + \ldots + a_{mi}y_m \color{red}{<} c_i \Rightarrow \color{red}{x_i = 0}$ (Reason: complement slackness. An optimal solution $\mathbf{y}$ satisfies $(a_{1i}y_1 + a_{2i}y_2 + \ldots + a_{mi}y_m - c_i) \times x_i = 0$)
  3. Let's use J to record the index of **tight constraints** where "$\color{red}{=}$" holds.

Step 1: $\mathbf{y} \Rightarrow \mathbf{x}$ II

$$
\begin{array}{ccccc}
 & & & & x_n = 0 \\
 & & & & \Uparrow \\
 & J & & & \\
c_1 & c_2 & \cdots & & c_n \\
\| & \| & & & \vee \\
y_1 a_{11} & y_1 a_{12} & \cdots & & y_1 a_{1n} \\
+ & + & & & + \\
y_2 a_{21} & y_2 a_{22} & \cdots & & y_2 a_{2n} \\
+ & + & & & + \\
\vdots & \vdots & \cdots & & \vdots \\
+ & + & & & + \\
y_m a_{m1} & y_m a_{m2} & \cdots & & y_m a_{mn}
\end{array}
$$

Step 1: $\mathbf{y} \Rightarrow \mathbf{x}$ III

4. Thus the corresponding primal solution $\mathbf{x}$ should satisfy the following restricted primal (RP):
5. RP:

$$
\begin{array}{rcrcccrcll}
a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1n}x_n & = & b_1 & \\
a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2n}x_n & = & b_2 & \\
 & & & & \ldots & & & & & \\
a_{m1}x_1 & + & a_{m2}x_2 & + & \ldots & + & a_{mn}x_n & = & b_m & \\
 & & & & & & x_i & = & 0 & i \notin J \\
 & & & & & & x_i & \geq & 0 & i \in J
\end{array}
$$

6. In other words, the optimality of $\mathbf{y}$ is determined via solving RP.

But how to solve RP? I

- RP:

$$
\begin{array}{rcrcccrcll}
a_{11}x_1 & + & a_{12}x_2 & + & ... & + & a_{1n}x_n & = & b_1 & \\
a_{21}x_1 & + & a_{22}x_2 & + & ... & + & a_{2n}x_n & = & b_2 & \\
 & & & & ... & & & & & \\
a_{m1}x_1 & + & a_{m2}x_2 & + & ... & + & a_{mn}x_n & = & b_m & \\
 & & & & & & x_i & = & 0 & i \notin J \\
 & & & & & & x_i & \geq & 0 & i \in J
\end{array}
$$

- How to solve RP? Recall that $\mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$ can be solved via solving an extended LP.

But how to solve RP? II

- RP (extended through introducing slack variables):

$$
\begin{array}{llllllllll}
\min & \epsilon = & s_1 & +s_2 & \ldots & +s_m & & & & \\
s.t. & & s_1 & & & & +a_{11}x_1 & \ldots & +a_{1n}x_n & = b_1 \\
 & & & s_2 & & & +a_{21}x_1 & \ldots & +a_{2n}x_n & = b_2 \\
 & & & & \ldots & & & \ldots & & \\
 & & & & & s_m & +a_{m1}x_1 & \ldots & +a_{mn}x_n & = b_m \\
 & & & & & & & & {\color{blue}x_i} & {\color{blue}= 0 \quad i} \\
 & & & & & & & & {\color{red}x_i} & {\color{red}\geq 0 \quad i} \\
 & & & & & & & & s_i & \geq 0 \quad \forall
\end{array}
$$

1. If $\epsilon_{OPT} = 0$, then we find a feasible solution to RP, implying that $\mathbf{y}$ is an optimal solution;
2. If $\epsilon_{OPT} > 0$, $\mathbf{y}$ is not an optimal solution.

- Alternatively, we can solve the dual of RP, called DRP:

$$
\begin{array}{rrrrrrrrrrrr}
\max & w & = & b_1y_1 & + & b_2y_2 & + & \ldots & + & b_my_m & & \\
s.t. & & & a_{11}y_1 & + & a_{21}y_2 & + & \ldots & + & a_{m1}y_m & \leq & 0 \\
 & & & a_{12}y_1 & + & a_{22}y_2 & + & \ldots & + & a_{m2}y_m & \leq & 0 \\
 & & & & & & & \ldots & & & & \\
 & & & a_{1|J|}y_1 & + & a_{2|J|}y_2 & + & \ldots & + & a_{m|J|}y_m & \leq & 0 \\
 & & & y_1 & , & y_2 & , & & , & y_m & \leq & 1
\end{array}
$$

1. If $w_{OPT} = 0$, $\mathbf{y}$ is an optimal solution
2. If $w_{OPT} > 0$, $\mathbf{y}$ is not an optimal solution. However, the optimal solution still provides useful information — the optimal solution to DRP can be used to improve $\mathbf{y}$.

The difference between DRP and D

- Dual problem D:

$$\begin{array}{rccccccccc}
\max & b_1y_1 & + & b_2y_2 & + & ... & + & b_my_m & & \\
s.t. & a_{11}y_1 & + & a_{21}y_2 & + & ... & + & a_{m1}y_m & \leq & c_1 \\
 & a_{12}y_1 & + & a_{22}y_2 & + & ... & + & a_{m2}y_m & \leq & c_2 \\
 & & & & & ... & & & & \\
 & a_{1n}y_1 & + & a_{2n}y_2 & + & ... & + & a_{mn}y_m & \leq & c_n
\end{array}$$

- DRP:

$$\begin{array}{rccccccccccc}
\max & w & = & b_1y_1 & + & b_2y_2 & + & ... & + & b_my_m & & \\
s.t. & & & a_{11}y_1 & + & a_{21}y_2 & + & ... & + & a_{m1}y_m & \leq & 0 \\
 & & & a_{12}y_1 & + & a_{22}y_2 & + & ... & + & a_{m2}y_m & \leq & 0 \\
 & & & & & & & ... & & & & \\
 & & & a_{1|J|}y_1 & + & a_{2|J|}y_2 & + & ... & + & a_{m|J|}y_m & \leq & 0 \\
 & & & y_1 & , & y_2 & , & & , & y_m & \leq & 1
\end{array}$$

- How to write DRP from D?
  - Replacing c_i with 0;
  - Only $|J|$ restrictions in DRP;
  - An additional restriction: $y_1, y_2, ..., y_m \leq 1$;

Why $\mathbf{\Delta y}$ can be used to improve $\mathbf{y}$? Consider an improved dual solution $\mathbf{y}' = \mathbf{y} + \theta \mathbf{\Delta y}, \theta > 0$. We have:

- **Objective function:** Since $\mathbf{\Delta y}^{\mathbf{T}}\mathbf{b} = w_{OPT} > 0$, $\mathbf{y}'^{\mathbf{T}}\mathbf{b} = \mathbf{y}^{\mathbf{T}}\mathbf{b} + \theta w_{OPT} > \mathbf{y}^{\mathbf{T}}\mathbf{b}$. In other words, $(\mathbf{y} + \theta \mathbf{\Delta y})$ is better than $\mathbf{y}$.
- **Constraints:** The dual feasibility requires that:
  - For any $j \in J$, $a_{1j}\Delta y_1 + a_{2j}\Delta y_2 + ... + a_{mj}\Delta y_m \leq 0$. Thus we have $\mathbf{y}'^{\mathbf{T}}\mathbf{a_j} = \mathbf{y}^{\mathbf{T}}\mathbf{a_j} + \theta \mathbf{\Delta y}^{\mathbf{T}}\mathbf{a_j} \leq \mathbf{c_j}$ for any $\theta > 0$.
  - For any $j \notin J$, there are two cases:

1. $\forall j \notin J, a_{1j}\Delta y_1 + a_{2j}\Delta y_2 + \ldots + a_{mj}\Delta y_m \leq 0$:
   Thus $\mathbf{y}'$ is feasible for any $\theta > 0$ since for $\forall 1 \leq j \leq n$,

$$
\begin{aligned}
& a_{1j}y'_1 + a_{2j}y'_2 + \ldots + a_{mj}y'_m && (1)\\
= \; & a_{1j}y_1 + a_{2j}y_2 + \ldots + a_{mj}y_m && (2)\\
+ \; & \theta(a_{1j}\Delta y_1 + a_{2j}\Delta y_2 + \ldots + a_{mj}\Delta y_m) && (3)\\
\leq \; & c_j && (4)
\end{aligned}
$$

   Hence dual problem D is unbounded and the primal problem P is infeasible.
2. $\exists j \notin J, a_{1j}\Delta y_1 + a_{2j}\Delta y_2 + \ldots + a_{mj}\Delta y_m > 0$:
   We can safely set $\theta \leq \frac{c_j - (a_{1j}y_1 + a_{2j}y_2 + \ldots + a_{mj}y_m)}{a_{1j}\Delta y_1 + a_{2j}\Delta y_2 + \ldots + a_{mj}\Delta y_m} = \frac{\mathbf{c_j} - \mathbf{y^T a_j}}{\mathbf{\Delta y^T a_j}}$
   to guarantee that $\mathbf{y'^T a_j} = \mathbf{y^T a_j} + \theta \mathbf{\Delta y^T a_j} \leq \mathbf{c_j}$.

Primal_Dual algorithm

```
Infeasible = "No"
Optimal = "No"
y = y0;        //y0 is a feasible solution to the dual problem D
while TRUE do
    Finding tight constraints index J, and set corresponding x_j = 0
    for j ∉ J.
    Thus we have a smaller RP.
    Solve DRP. Denote the solution as Δy.
    if DRP objective function w_OPT = 0 then
        Optimal="Yes"
        return  y;
    end if
    if Δy^T a_j ≤ 0 (for all j ∉ J) then
        Infeasible = "Yes";
        return  ;
    end if
    Set θ = min (c_j − y^T a_j)/(Δy^T a_j) for Δy^T a_j > 0, j ∉ J.
    Update y as y = y + θΔy;
end while
```

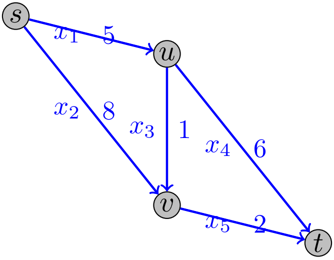
Advantages of Primal_Dual algorithm

- Some facts:
  - Primal_dual algorithm ends if using anti-cycling rule. (Reason: the objective value $\mathbf{y^T b}$ increases if there is no degeneracy.)
  - Both RP and DRP do not explicitly rely on $\mathbf{c}$. In fact, the information of $\mathbf{c}$ is represented in J.
  - This leads to another advantage of primal_dual technique, i.e., RP is usually a purely combinatorial problem. Take ShortestPath as an example. RP corresponds to a "connection" problem.
  - More and more constraints become tight in the primal_dual process.

(See Lecture 10 for a primal_dual algorithm for MaximumFlow problem.)

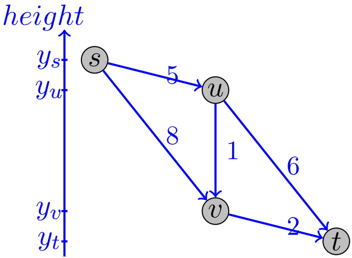
ShortestPath: Dijkstra's algorithm is essentially Primal_Dual algorithm

ShortestPath problem



- Primal problem: relax the 0/1 integer linear program into linear program by the **totally uni-modular** property.

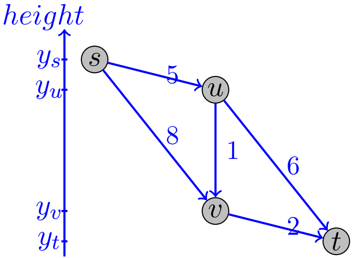
$$
\begin{array}{lrcrcrcrcrll}
\min & 5x_1 & + & 8x_2 & + & 1x_3 & + & 6x_4 & + & 2x_5 & & \\
s.t. & x_1 & + & x_2 & & & & & & & = 1 & \text{vertex } s \\
 & & & & & & - & x_4 & - & x_5 & = -1 & \text{vertex } t \\
 & -x_1 & & & + & x_3 & + & x_4 & & & = 0 & \text{vertex } u \\
 & & - & x_2 & - & x_3 & & & + & x_5 & = 0 & \text{vertex } v \\
 & x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & \geq 0 & \\
 & x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & \leq 1 &
\end{array}
$$



- Dual problem: set variables for **cities**. (Intuition: y_i means the height of city i; thus, $y_s - y_t$ denotes the height difference between s and t, providing a lower bound of the shortest path length.)

$$
\begin{array}{rrrrrrrrrll}
\max & y_s & - & y_t & & & & & & & \\
s.t. & y_s & & & - & y_u & & & \leq 5 & x_1 : \text{edge } (s,u) \\
 & y_s & & & & & - & y_v & \leq 8 & x_2 : \text{edge } (s,v) \\
 & & & & & y_u & - & y_v & \leq 1 & x_3 : \text{edge } (u,v) \\
 & & - & y_t & + & y_u & & & \leq 6 & x_4 : \text{edge } (u,t) \\
 & & - & y_t & & & + & y_v & \leq 2 & x_5 : \text{edge } (v,t)
\end{array}
$$

A simplified version



- Dual problem: simplify by setting $y_t = 0$ (and remove the 2nd constraint in the primal problem P, accordingly)

$$
\begin{array}{rlllll}
\max & y_s & & & & \\
s.t. & y_s & - & y_u & & \leq 5 \quad x_1 : \text{edge } (s,u) \\
 & y_s & & & - \; y_v & \leq 8 \quad x_2 : \text{edge } (s,v) \\
 & & & y_u & - \; y_v & \leq 1 \quad x_3 : \text{edge } (u,v) \\
 & & & y_u & & \leq 6 \quad x_4 : \text{edge } (u,t) \\
 & & & & y_v & \leq 2 \quad x_5 : \text{edge } (v,t)
\end{array}
$$

- Dual feasible solution: $\mathbf{y}^\mathbf{T} = (\mathbf{0}, \mathbf{0}, \mathbf{0})$. Let's check the constraints in D:

$$\begin{array}{ccccccccc}
y_s & - & y_u & & & < & 5 & \Rightarrow x_1 = 0 \\
y_s & & & - & y_v & < & 8 & \Rightarrow x_2 = 0 \\
& & y_u & - & y_v & < & 1 & \Rightarrow x_3 = 0 \\
& & y_u & & & < & 6 & \Rightarrow x_4 = 0 \\
& & & & y_v & < & 2 & \Rightarrow x_5 = 0
\end{array}$$

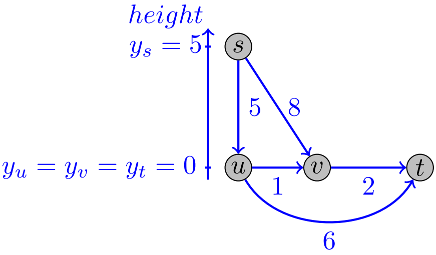
- Identifying tight constraints in D: $J = \Phi$, implying that $x_1, x_2, x_3, x_4, x_5 = 0$.
- RP:

$$\begin{array}{lcccccccccl}
\min & s_1 & +s_2 & +s_3 & & & & & & & \\
s.t. & s_1 & & & +x_1 & +x_2 & & & & =1 & \text{node } s \\
& & s_2 & & -x_1 & & +x_3 & +x_4 & & =0 & \text{node } u \\
& & & s_3 & & -x_2 & -x_3 & & +x_5 & =0 & \text{node } v \\
& s_1, & s_2, & s_3, & & & & & & \geq 0 & \\
& & & & x_1, & x_2, & x_3, & x_4, & x_5 & =0 &
\end{array}$$

- DRP:

$$\begin{array}{rllll} \max & y_s & & & \\ s.t. & y_s & & & \leq 1 \\ & & y_u & & \leq 1 \\ & & & y_v & \leq 1 \end{array}$$

- Solve DRP using combinatorial technique: optimal solution $\Delta\mathbf{y^T} = (1, 0, 0)$. *Note: the optimal solution is not unique*
- Step length θ: $\theta = \min\{\frac{\mathbf{c_1 - y^T a_1}}{\mathbf{\Delta y^T a_1}}, \frac{\mathbf{c_2 - y^T a_2}}{\mathbf{\Delta y^T a_2}}\} = \min\{5, 8\} = 5$
- Update $\mathbf{y}$: $\mathbf{y^T} = \mathbf{y^T} + \theta\Delta\mathbf{y^T} = (5, 0, 0)$.



- From the point of view of Dijkstra's algorithm:
  - Optimal solution to DRP is $\Delta\mathbf{y}^{\mathbf{T}} = (1, 0, 0)$: the explored vertex set $S = \{s\}$ in Dijkstra's algorithm. In fact, DRP is solved via identifying the nodes reachable from s.
  - Step length $\theta = \min\{\frac{\mathbf{c_1} - \mathbf{y}^{\mathbf{T}}\mathbf{a_1}}{\Delta\mathbf{y}^{\mathbf{T}}\mathbf{a_1}}, \frac{\mathbf{c_2} - \mathbf{y}^{\mathbf{T}}\mathbf{a_2}}{\Delta\mathbf{y}^{\mathbf{T}}\mathbf{a_2}}\} = \min\{5, 8\} = 5$: finding the closest vertex to the nodes in S via comparing all edges going out from S.

- Dual feasible solution: $\mathbf{y^T} = (\mathbf{5}, \mathbf{0}, \mathbf{0})$. Let's check the constraints in D:

$$
\begin{array}{ccccccccc}
y_s & - & y_u & & & = & 5 & & \\
y_s & & & - & y_v & < & 8 & \Rightarrow x_2 = 0 \\
& & y_u & - & y_v & < & 1 & \Rightarrow x_3 = 0 \\
& & y_u & & & < & 6 & \Rightarrow x_4 = 0 \\
& & & & y_v & < & 2 & \Rightarrow x_5 = 0
\end{array}
$$

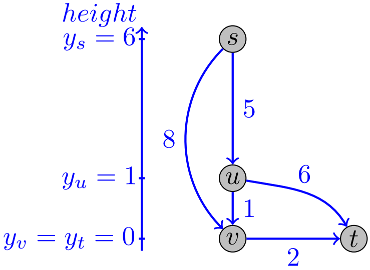
- Identifying tight constraints in D: $J = \{1\}$, implying that $x_2, x_3, x_4, x_5 = 0$.
- RP:

$$
\begin{array}{lccccccccll}
\min & s_1 & +s_2 & +s_3 & & & & & & & \\
s.t. & s_1 & & & +x_1 & +x_2 & & & & = 1 & \text{node } s \\
& & s_2 & & -x_1 & & +x_3 & +x_4 & & = 0 & \text{node } u \\
& & & s_3 & & -x_2 & -x_3 & & +x_5 & = 0 & \text{node } v \\
& s_1, & s_2, & s_3, & & & & & & \geq 0 & \\
& & & & & x_2, & x_3, & x_4, & x_5 & = 0 &
\end{array}
$$

- DRP:

$$\begin{array}{rccccccl} \max & y_s & & & & & & \\ s.t. & y_s & - & y_u & & & & \leq 0 \\ & y_s & , & y_u & , & y_v & & \leq 1 \end{array}$$

- Solve DRP using combinatorial technique: optimal solution $\Delta \mathbf{y}^{\mathbf{T}} = (1, 1, 0)$. *Note: the optimal solution is not unique*
- Step length θ:
$\theta = \min\{\frac{\mathbf{c_2} - \mathbf{y}^{\mathbf{T}}\mathbf{a_2}}{\Delta \mathbf{y}^{\mathbf{T}}\mathbf{a_2}}, \frac{\mathbf{c_3} - \mathbf{y}^{\mathbf{T}}\mathbf{a_3}}{\Delta \mathbf{y}^{\mathbf{T}}\mathbf{a_3}}, \frac{\mathbf{c_4} - \mathbf{y}^{\mathbf{T}}\mathbf{a_4}}{\Delta \mathbf{y}^{\mathbf{T}}\mathbf{a_4}}\} = \min\{3, 1, 6\} = 1$
- Update $\mathbf{y}$: $\mathbf{y}^{\mathbf{T}} = \mathbf{y}^{\mathbf{T}} + \theta \Delta \mathbf{y}^{\mathbf{T}} = (6, 1, 0)$.



- From the point of view of Dijkstra's algorithm:
  - Optimal solution to DRP is $\Delta\mathbf{y}^\mathbf{T} = (1, 1, 0)$: the explored vertex set $S = \{s, u\}$ in Dijkstra's algorithm. In fact, DRP is solved via identifying the nodes reachable from s.
  - Step length $\theta = \min\{\frac{\mathbf{c_2} - \mathbf{y}^\mathbf{T}\mathbf{a_2}}{\Delta\mathbf{y}^\mathbf{T}\mathbf{a_2}}, \frac{\mathbf{c_3} - \mathbf{y}^\mathbf{T}\mathbf{a_3}}{\Delta\mathbf{y}^\mathbf{T}\mathbf{a_3}}, \frac{\mathbf{c_4} - \mathbf{y}^\mathbf{T}\mathbf{a_4}}{\Delta\mathbf{y}^\mathbf{T}\mathbf{a_4}}\} = \min\{3, 1, 6\} = 1$: finding the closest vertex to the nodes in S via comparing all edges going out from S.

Iteration 3 I

- Dual feasible solution: $\mathbf{y^T} = (\mathbf{6}, \mathbf{1}, \mathbf{0})$. Let's check the constraints in D:

$$
\begin{array}{ccccccccl}
y_s & - & y_u & & & = & 5 & & \\
y_s & & & - & y_v & < & 8 & \Rightarrow x_2 = 0 \\
& & y_u & - & y_v & = & 1 & & \\
& & y_u & & & < & 6 & \Rightarrow x_4 = 0 \\
& & & & y_v & < & 2 & \Rightarrow x_5 = 0
\end{array}
$$

- Identifying tight constraints in D: $J = \{1, 3\}$, implying that $x_2, x_4, x_5 = 0$.
- RP:

$$
\begin{array}{llllllllll}
\min & s_1 & +s_2 & +s_3 & & & & & & \\
s.t. & s_1 & & & +x_1 & +x_2 & & & & = 1 \quad \text{node } s \\
& & s_2 & & -x_1 & & +x_3 & +x_4 & & = 0 \quad \text{node } u \\
& & & s_3 & & -x_2 & -x_3 & & +x_5 & = 0 \quad \text{node } v \\
& s_1, & s_2, & s_3, & & & & & & \geq 0 \\
& & & & & x_2, & & x_4, & x_5 & = 0
\end{array}
$$

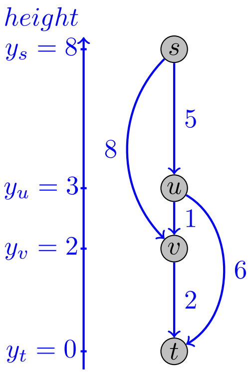
Iteration 3 II

- DRP:

$$\begin{array}{rccccccl} \max & y_s & & & & & & \\ s.t. & y_s & - & y_u & & & \leq 0 \\ & & & y_u & - & y_v & \leq 0 \\ & y_s & , & y_u & , & y_v & \leq 1 \end{array}$$

- Solve DRP using combinatorial technique: optimal solution $\Delta\mathbf{y}^{\mathbf{T}} = (1, 1, 1)$. *Note: the optimal solution is not unique*
- Step length θ: $\theta = \min\{\frac{\mathbf{c_4} - \mathbf{y}^{\mathbf{T}}\mathbf{a_4}}{\Delta\mathbf{y}^{\mathbf{T}}\mathbf{a_4}}, \frac{\mathbf{c_5} - \mathbf{y}^{\mathbf{T}}\mathbf{a_5}}{\Delta\mathbf{y}^{\mathbf{T}}\mathbf{a_5}}\} = \min\{5, 2\} = 2$
- Update $\mathbf{y}$: $\mathbf{y}^{\mathbf{T}} = \mathbf{y}^{\mathbf{T}} + \theta\Delta\mathbf{y}^{\mathbf{T}} = (8, 3, 2)$.

Iteration 3 III



- From the point of view of Dijkstra's algorithm:
  - Optimal solution to DRP is $\Delta \mathbf{y}^{\mathbf{T}} = (1, 1, 1)$: the explored vertex set $S = \{s, u, v\}$ in Dijkstra's algorithm. In fact, DRP is solved via identifying the nodes reachable from s.

Iteration 3 IV

- Step length $\theta = \min\{\frac{\mathbf{c_4 - y^T a_4}}{\mathbf{\Delta y^T a_4}}, \frac{\mathbf{c_5 - y^T a_5}}{\mathbf{\Delta y^T a_5}}\} = \min\{5, 2\} = 2$: finding the closest vertex to the nodes in S via comparing all edges going out from S.

- Dual feasible solution: $\mathbf{y^T} = (\mathbf{8}, \mathbf{3}, \mathbf{2})$. Let's check the constraints in D:

$$
\begin{array}{cccccccl}
y_s & - & y_u & & & = & 5 & \\
y_s & & & - & y_v & < & 8 & \Rightarrow x_2 = 0 \\
 & & y_u & - & y_v & = & 1 & \\
 & & y_u & & & < & 6 & \Rightarrow x_4 = 0 \\
 & & & & y_v & = & 2 &
\end{array}
$$

- Identifying tight constraints in D: $J = \{1, 3, 5\}$, implying that $x_2, x_4 = 0$.
- RP:

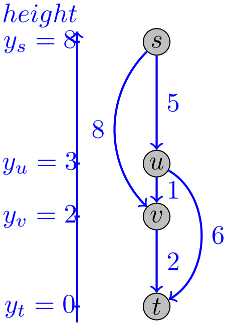
$$
\begin{array}{llllllllll}
\min & s_1 & +s_2 & +s_3 & & & & & & \\
s.t. & s_1 & & & +x_1 & +x_2 & & & & = 1 \quad \text{node } s \\
 & & s_2 & & -x_1 & & +x_3 & +x_4 & & = 0 \quad \text{node } u \\
 & & & s_3 & & -x_2 & -x_3 & & +x_5 & = 0 \quad \text{node } v \\
 & s_1, & s_2, & s_3, & & & & & & \geq 0 \\
 & & & & & x_2, & & x_4 & & = 0
\end{array}
$$

Iteration 4 II

- DRP:

$$\begin{array}{rccccccl} \max & y_s & & & & & & \\ s.t. & y_s & - & y_u & & & & \leq 0 \\ & & & y_u & - & y_v & & \leq 0 \\ & & & & & y_v & & \leq 0 \\ & y_s & , & y_u & , & y_v & & \leq 1 \end{array}$$

- Solve DRP using combinatorial technique: optimal solution $\Delta\mathbf{y}^{\mathbf{T}} = (0, 0, 0)$. Done!



- From the point of view of Dijkstra's algorithm:
  - Optimal solution to DRP is $\Delta\mathbf{y}^{\mathbf{T}} = (0, 0, 0)$: there is a path from s to t, forcing $y_s = 0$ (note y_t is fixed to be 0). This corresponds to the explored node set $S = \{s, u, v, t\}$ in Dijkstra's algorithm.
- Another intuitive explanation: the **tightest** rope when picking up s.

Application 1: A succinct proof of Farkas lemma [1894]

Theorem (Farkas lemma)

Given vectors $\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_m}, \mathbf{c} \in \mathbb{R}^n$. *Then either*

1. $\mathbf{c} \in \mathbf{C}(\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_m})$; *or*
2. *there is a vector* $\mathbf{y} \in \mathbb{R}^n$ *such that for all* i, $\mathbf{y^T a_i} \geq \mathbf{0}$ *but* $\mathbf{y^T c} < \mathbf{0}$.

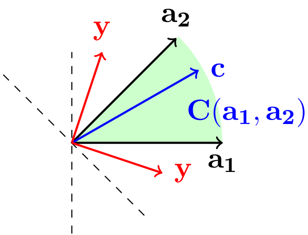


Figure: Case 1: $\mathbf{c} \in \mathbf{C}(\mathbf{a_1}, \mathbf{a_2})$

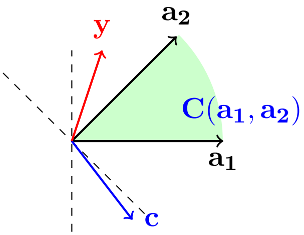


Figure: Case 2: $\mathbf{c} \notin \mathbf{C}(\mathbf{a_1}, \mathbf{a_2})$

- Here, $\mathbf{C}(\mathbf{a_1}, ..., \mathbf{a_m})$ denotes the cone spanned by $\mathbf{a_1}, ..., \mathbf{a_m}$, i.e. $\mathbf{C}(\mathbf{a_1}, ..., \mathbf{a_m}) = \{\mathbf{x} | \mathbf{x} = \sum_{i=1}^{m} \lambda_i \mathbf{a_i}, \lambda_i \geq 0\}$.

Proof.

- Suppose for any vector $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{y^T a_i} \geq \mathbf{0}$ $(i = 1, 2, ..., m)$, we always have $\mathbf{y^T c} \geq \mathbf{0}$. We will show that $\mathbf{c}$ should lie within the cone $\mathbf{C}(\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_m})$.
- Consider the following PRIMAL problem:

$$\begin{array}{llll} \min & \mathbf{c^T y} & & \\ s.t. & \mathbf{a_i^T y} & \geq & \mathbf{0} \quad i = 1, 2, ..., m \end{array}$$

- It is obvious that the PRIMAL problem has a feasible solution $\mathbf{y} = \mathbf{0}$, and is bounded since $\mathbf{c^T y} \geq \mathbf{0}$.
- Thus the DUAL problem also has a bounded optimal solution:

$$\begin{array}{rrcl} \max & 0 & & \\ s.t. & \mathbf{x^T A^T} & = & \mathbf{c^T} \\ & \mathbf{x} & \geq & \mathbf{0} \end{array}$$

- In other words, there exists a vector $\mathbf{x}$ such that $\mathbf{c} = \sum_{i=1}^{m} x_i \mathbf{a_i}$ and $x_i \geq 0$.

Variants of Farkas' lemma

Farkas' lemma lies at the core of linear optimization. Using Farkas' lemma, we can prove SEPARATION theorem, and MINIMAX theorem in the game theory.

Theorem

Let $\mathbf{A}$ be an $m \times n$ matrix, and $\mathbf{b} \in \mathbb{R}^m$. Then either

1. $\mathbf{Ax} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$ *has a feasible solution; or*
2. *there is a vector* $\mathbf{y} \in \mathbb{R}^m$ *such that* $\mathbf{y}^\mathbf{T}\mathbf{A} \geq \mathbf{0}$ *but* $\mathbf{y}^\mathbf{T}\mathbf{b} < \mathbf{0}$.

Variants of Farkas' lemma

Theorem

Let $\mathbf{A}$ *be an* $m \times n$ *matrix, and* $\mathbf{b} \in \mathbb{R}^m$*. Then either*

1. $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ *has a feasible solution; or*
2. *there is a vector* $\mathbf{y} \in \mathbb{R}^m$ *such that* $\mathbf{y} \geq \mathbf{0}$, $\mathbf{y}^{\mathbf{T}}\mathbf{A} \geq \mathbf{0}$ *but* $\mathbf{y}^{\mathbf{T}}\mathbf{b} < \mathbf{0}$.

Caratheodory's theorem

Theorem

Given vectors $\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_m} \in \mathbb{R}^n$*. If* $\mathbf{x} \in \mathbf{C}(\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_m})$*, then there is a linearly independent vector set of* $\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_m}$*, say* $\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_r}$*, such that* $\mathbf{x} \in \mathbf{C}(\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_r})$*.*

Theorem

Let $\mathbf{C} \subseteq \mathbb{R}^n$ *be a closed, convex set, and let* $\mathbf{x} \in \mathbb{R}^n$. *If* $\mathbf{x} \notin \mathbf{C}$, *then there exists a hyperplane separating* $\mathbf{x}$ *from* $\mathbf{C}$.

Application 2: von Neumann's MINIMAX theorem on game theory

Game theory

- Game theory studies competing and cooperative behaviours among intelligent and rational decision-makers.
- In 1928, John von Neumann proved the existence of mixed-strategy equilibria in two-person zero-sum games.
- In 1950, John Forbes Nash Jr. developed a criterion of mutual consistency of players' strategies, which applies to a wider range of games than that proposed by J. von Neumann. He proved the existence of Nash equilibrium in every n-player, non-zero-sum, non-cooperative game (not just 2-player, zero-sum games).
- Game theory was widely applied in mathematical economics, in biology (e.g., analysis of evolution and stability) and computer science (e.g., analysis of interactive computations and lower bound on the complexity of randomized algorithms, the equivalence between linear program and two-person zero-sum game).

Paper-rock-scissors: an example of two-player zero-sum game

- Paper-rock-scissors is a hand game usually played by two players, denoted as row player and column player: each player selects one of the three hand shapes, including "paper", "rock", and "scissors"; then the players show their selections simultaneously.
- It has two possible outcomes other than tie: one player wins and the other player loses, which can be formally described using the following payoff matrix.

	Paper	Rock	Scissors
Paper	0, 0	1, -1	-1, 1
Rock	-1, 1	0, 0	1, -1
Scissors	1, -1	-1, 1	0, 0

- Each player attempts to select appropriate action to maximize his gain.

Matching penny: another example of two-person zero-sum game

- Matching pennies is a game played by two players, namely, row player and column player. Each player has a penny and secretly turns it to head or tail. The players then reveal their selections simultaneously.
- If the pennies match, then row player keeps both pennies; otherwise, column player keeps both. The payoff matrix is as follows.

	Head	Tail
Head	1, -1	-1, 1
Tail	-1, 1	1, -1

- Each player tries to maximize his gain via making an appropriate selection.

Simultaneous games vs. sequential games

- **Simultaneous games** are games in which all players move **simultaneously**. Thus, no player have information of the others' selections in advance.
- **Sequential games** are games in which the later player has some information, although maybe imperfect, of previous actions by the other players. A complete plan of action for every stage of the game, regardless of whether the action actually arises in play, is denoted as **a (pure) strategy**.
- **Normal form** is used to describe simultaneous games while **extensive form** is used to describe sequential games.
- J. von Neumann proposed an approach to transform transform strategies in sequential games into actions in simultaneous games.
- Note that the transformation is one-way, i.e., multiple sequential games might correspond to the same simultaneous game, and it may result in an exponential blowup in the size of the representation.

Normal form

- A game Γ in normal form among m players contains the following items:
  - Each player k has a finite number of **pure strategies** $S_k = \{1, 2, ..., n_k\}$.
  - Each player k is associated with a **payoff function** $H_k : S_1 \times S_2 \times ... \times S_m \to \mathbb{R}$.
- To play the game, each player selects a strategy without information of others, and then reveals the selection **simultaneously**. The players' gain are calculated using corresponding payoff functions.
- Each player attempts to maximize his gain via selecting an appropriate strategy.

Two-person zero-sum game in normal form

- In a two-person zero-sum game game Γ, a player's gain or less is exactly balanced by the other player's loss or gain, i.e.,

$$H_1(s_1, s_2) + H_2(s_1, s_2) = 0.$$

- Thus we can define another function

$$H(s_1, s_2) = H_1(s_1, s_2) = -H_2(s_1, s_2)$$

and represent it using a **payoff matrix**.

	Head	Tail
Head	1	-1
Tail	-1	1

- Row player aims to maximize $H(s_1, s_2)$ by selecting an appropriate strategy s_1 while column player aims to minimize $H(s_1, s_2)$ by selecting an appropriate strategy s_2.

von Neumann's MiniMax theorem: motivation

- When analyzing a two-person zero-sum game Γ, von Neumann noticed that the difficulty comes from the difference between games and ordinary optimization problems: row player tries to maximize $H(s_1, s_2)$; however, he can control s_1 only as he has no information of the other player's selection s_2, and so does column player.
- Thus von Neumann suggested to investigate two auxiliary games without this difficulty, denoted as Γ_1 and Γ_2, before attacking the challenging game Γ.
  1. Γ_1: Row player selects a strategy s_1 first, and exposes his selection to column player before column player selects a strategy s_2.
  2. Γ_2: Column player selects a strategy s_2 first, and exposes his selection to row player before row player selects a strategy s_1.
- The two auxiliary games are much easier than the original game Γ, and more importantly, they provide upper and lower bounds for Γ.

- Let's consider column player first. As he knows row player's selection s_1, the objective function $H(s_1, s_2)$ becomes an ordinary optimization function over a single variable s_2, and column player can simply select a strategy s_2 with the minimum objective function value $\min_{s_2} H(s_1, s_2)$.

	Head	Tail	Row minimum
Head	-2	1	-2
Tail	-1	2	$v_1 = -1$

- Now consider row player. When he selects a strategy s_1, he can definitely predict the selection of column player. Since $\min_{s_2} H(s_1, s_2)$ is an ordinary function over a single s_1, it is easy for row player to select a strategy s_1 with the maximum objective function value

$$v_1 = \max_{s_1} \min_{s_2} H(s_1, s_2).$$

Auxiliary game Γ_2

- Let's consider row player first. As he knows column player's selection s_2, the objective function $H(s_1, s_2)$ becomes an ordinary optimization function over a single variable s_1, and row player can simply select a strategy s_1 with the maximum objective function value $\max_{s_1} H(s_1, s_2)$.

	Head	Tail
Head	-2	1
Tail	-1	2
Column maximum	$v_2 = -1$	2

- Now consider column player. When he selects a strategy s_2, he can definitely predict the selection of row player. Since $\max_{s_1} H(s_1, s_2)$ is an ordinary function over a single variable s_2, it is easy for column player to select a strategy s_2 with the minimum objective function value

$$v_2 = \min_{s_2} \max_{s_1} H(s_1, s_2).$$

Γ_1 and Γ_2 bound Γ

- For row player, it is clearly Γ_1 is disadvantageous to him as he should expose his selection s_1 to column player.
- On the contrary, Γ_2 is beneficial to row player as he knows column player's selection s_2 before making decision.

	Head	Tail	Row minimum
Head	-2	1	-2
Tail	-1	2	$v_1 = -1$
Column maximum	$v_2 = -1$	2	

- Thus these two auxiliary games provides lower and upper bounds:

$$v_1 \leq v \leq v_2$$

where v denote row player's gain in the original game Γ.

Case 1: $v_1 = v_2$

- For a game with the following payoff matrix, we have $v_1 = v = v_2$ and call this game **strictly determined**.

	Head	Tail	Row minimum
Head	-2	1	-2
Tail	-1	2	$v_1 = -1$
Column maximum	$v_2 = -1$	2	

- The saddle point of the payoff matrix $H(s_1, s_2)$ represents a **pure strategy equilibrium**. In this equilibrium, each player has nothing to gain by changing only his own strategy. In addition, knowing the opponent's selection will bring no gain.
- von Neumann proved the existence of the optimal strategy in a perfect information two-person zero-sum game, e.g., chess. L. S. Shapley further showed that a finite two-person zero-sum game has a pure strategy equilibrium if every 2×2 submatrix of the game has a pure strategy equilibrium [?].

Case 2: $v_1 < v_2$

- In contrast, matching penny does not have a pure strategy equilibrium as there is no saddle point in the payoff matrix. So does the paper-rock-scissors game.

	Head	Tail	Row minimum
Head	1	-1	-1
Tail	-1	1	$v_1 = -1$
Column maximum	$v_2 = 1$	1	

- This fact implies that knowing the opponent's selection might bring gain; however, it is impossible to know the opponent's selection as the players reveal their selections simultaneously. In this case, let's play a mixed strategy rather than a pure strategy.

From pure strategy to mixed strategy

- A **mixed strategy** is an assignment of probability to pure strategies, allowing a player to **randomly select a pure strategy**.
- Consider the payoff matrix as below. If the row player select strategy A with probability 1, he is said to play a pure strategy. If he tosses a coin and select strategy A if the coin lands head and B otherwise, then he is said to play a mixed strategy.

	A	B
A	1	-1
B	-1	1

Two types of interpretation of mixed strategy

- From a player's viewpoint: J. von Neumann described the motivation underlying the introduction of mixed strategy as follows: since it is impossible to exactly know opponent's selection, a player could switch to **protect himself by "randomly selecting his own strategy"**, making it difficult for the opponent to know the player's selection. However, this interpretation came under heavy fire for lacking of behaviour supports: Seldom do people make choices following a lottery.
- From opponent's viewpoint: Robert Aumann and Adam Brandenburger interpreted mixed strategy of a player as **opponent's "belief" of the player's selection**. Thus, Nash equilibrium is an equilibrium of "belief" rather than actions.

Existence of mixed strategy equilibrium

- Consider a mixed strategy game: row player has m strategies available and he selects a strategy s_1 according to a distribution $\mathbf{u}$, while column player has n strategies available and he selects a strategy s_2 according to a distribution $\mathbf{v}$, i.e.,

$$\Pr(s_1 = i) = u_i, i = 1, ..., n \quad \Pr(s_2 = j) = v_j, j = 1, ..., m$$

  Here, $\mathbf{u}$ and $\mathbf{v}$ are independent.
- Thus the expected gain of row player is:

$$\sum_{i=1}^{m} \sum_{j=1}^{n} u_i H_{ij} v_j = \mathbf{u^T H v}$$

- row player attempts to minimize $\mathbf{u^T H v}$ via selecting an appropriate $\mathbf{u}$, while column player attempts to maximize it via selecting an appropriate $\mathbf{v}$.
- Now let's consider the two auxiliary games Γ_1 and Γ_2 again and answer the following questions: what happens if row player exposes his mixed strategy to column player? And if we reverse the order of the players?

von Neumann's MiniMax theorem [1928]

- This question has been answered by the von Neumann's MiniMax theorem.

Theorem

$$\max_{\mathbf{u}} \min_{\mathbf{v}} \mathbf{u}^{\mathbf{T}}\mathbf{H}\mathbf{v} = \min_{\mathbf{v}} \max_{\mathbf{u}} \mathbf{u}^{\mathbf{T}}\mathbf{H}\mathbf{v}$$

- The theorem states that knowing the other player's strategy will bring no gain in a mixed-strategy zero-sum game, and the order doesn't change the value.
- A **mixed-strategy Nash equilibrium** exists for **any two-person zero-sum game** with a finite set of actions. A Nash equilibrium in a two-player game is a pair of strategies, each of which is a best response to the other; i.e., each gives the player using it the highest possible payoff, given the other players' strategy.

- Let's consider the auxiliary game Γ_1 first, in which the strategy of row player, i.e., $\mathbf{u}$, was exposed to column player. This is of course beneficial to column player since he can select the optimal strategy $\mathbf{v}$ to minimize $\mathbf{u}^{\mathbf{T}}\mathbf{H}\mathbf{v}$, which is

$$\inf\{\mathbf{u}^{\mathbf{T}}\mathbf{H}\mathbf{v}|\mathbf{v} \geq \mathbf{0}, \mathbf{1}^{\mathbf{T}}\mathbf{v} = 1\} = \min_{j=1,\ldots,n} (\mathbf{u}^{\mathbf{T}}\mathbf{H})_j$$

- Thus row player should select $\mathbf{u}$ to maximize the above value, which can be formulated as a linear program:

$$\begin{array}{lllll} \max & \min_{j=1,\ldots,n} (\mathbf{u}^{\mathbf{T}}\mathbf{H})_j & & & \\ s.t. & \mathbf{1}^{\mathbf{T}}\mathbf{u} & & = & 1 \\ & \mathbf{u} & & \geq & \mathbf{0} \end{array}$$

- The linear program can be rewritten as below.

$$\begin{array}{llll} \max & s & & \\ s.t. & \mathbf{u^T H} & \geq & s\mathbf{1^T} \\ & \mathbf{1^T u} & = & 1 \\ & \mathbf{u} & \geq & \mathbf{0} \end{array}$$

- Similarly we consider the auxiliary game Γ_2 and calculate the optimal strategy $\mathbf{v}$ by solving the following linear program.

$$\begin{array}{llll} \min & t & & \\ s.t. & \mathbf{Hv} & \leq & t\mathbf{1} \\ & \mathbf{1^T v} & = & 1 \\ & \mathbf{v} & \geq & \mathbf{0} \end{array}$$

- These two linear programs are both feasible and form Lagrangian dual. Thus they have the same optimal objective value according to the strong duality property.

An example: paper-rock-scissors game

- For the paper-rock-scissors game, we have the following two linear programs.
  - Linear program for Γ_1:

$$\begin{array}{lrcrcrl}
\max & s & & & & & \\
s.t. & 0u_1 & - & u_2 & + & u_3 & \geq s \\
 & u_1 & + & 0u_2 & - & u_3 & \geq s \\
 & -u_1 & + & u_2 & + & 0u_3 & \geq s \\
 & u_1 & + & u_2 & + & u_3 & = 1 \\
 & u_1, & & u_2, & & u_3 & \geq 0
\end{array}$$

  - Linear program for Γ_2:

$$\begin{array}{lrcrcrl}
\min & t & & & & & \\
s.t. & 0v_1 & + & v_2 & - & v_3 & \leq t \\
 & -v_1 & + & 0v_2 & + & v_3 & \leq t \\
 & v_1 & - & v_2 & + & 0v_3 & \leq t \\
 & v_1 & + & v_2 & + & v_3 & = 1 \\
 & v_1, & & v_2, & & v_3 & \geq 0
\end{array}$$

- The mixed strategy equilibrium is $\mathbf{u}^{\mathbf{T}} = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ and $\mathbf{u}^{\mathbf{T}} = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ with the game value 0.

Comments on the mixed strategy equilibrium by von Neumann

- Note that a mixed strategy equilibrium always exists no matter whether the payoff matrix $\mathbf{H}$ has a saddle point or not.
- Regardless of column player's selection, row player can select an appropriate strategy to guarantee his gain $v_1 \geq 0$.
- Regardless of row player's selection, column player can select an appropriate strategy to guarantee row player's gain $v_1 \leq 0$.
- Using the strategy $\mathbf{u}^{\mathbf{T}} = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$, row player can guarantee that he "won't lose", i.e., the probability of losing is less than the probability of winning.
- The strategy $\mathbf{u}^{\mathbf{T}} = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ is designed for "protecting himself" rather than "attacking his opponent", i.e., it cannot be used to benefit from opponent's fault.

Application 3: Yao's MiniMax principle [1977]

Yao's MINIMAX principle

- Consider a problem Π. Let $\mathcal{A} = \{A_1, A_2, ..., A_n\}$ be algorithms to Π, and $\mathcal{I} = \{I_1, I_2, ..., I_m\}$ be the inputs with a given size. Let $T(A_i, I_j)$ be the running time of algorithm A_i on the input I_j.

		Algorithms	
		A_1	A_2
Inputs	I_1	T_{11}	T_{12}
	I_2	T_{21}	T_{22}

- Thus $\max_{I_j \in \mathcal{I}} T(A_i, I_j)$ represents the worst-case time for the deterministic algorithm A_i.
- For a randomized algorithms, however, it is usually difficult to bound its expected running time on worst-case inputs.
- Yao's MINIMAX principle provides a technique to build lower bound for **the expected running time of any randomized algorithm on its worst-case input**.

Expected running time of a randomized algorithm A_q

- A "Las Vegas" randomized algorithm can be viewed as a distribution over all deterministic algorithms $\mathcal{A} = \{A_1, A_2, ..., A_n\}$.
- Specifically, let q be a distribution over $\mathcal{A}$, and A_q be a randomized algorithm chosen according to q, i.e., A_q refers to a deterministic algorithm A_i with probability q_i.
- Given a input I_j, the expected running time of A_q can be written as

$$E[T(A_q, I_j)] = \sum_{i=1}^{n} q_i T(A_i, I_j)$$

- Thus $\max_{I_j \in \mathcal{I}} E[T(A_q, I_j)]$ represents the expected running time of A_q on its worst-case input.

Expected running time of a deterministic algorithm A_i on random input

- Now consider a deterministic algorithm A_i running on random input.
- Let p be a distribution over $\mathcal{I}$, and I_p be a random input chosen from $\mathcal{I}$, i.e., I_p refers to I_j with probability p_j.
- Given a deterministic algorithm A_i, its expected running time on random input I_p can be written as

$$E[T(A_i, I_p)] = \sum_{j=1}^{m} p_j T(A_i, I_j)$$

- Thus $\min_{A_i \in \mathcal{A}} E[T(A_i, I_p)]$ represents the expected running time of the best deterministic algorithm on the random input I_p.

Yao's MINIMAX principle

Theorem

For any random input I_p and randomized algorithm A_q,

$$\min_{A_i \in \mathcal{A}} E[T(A_i, I_p)] \leq \max_{I_j \in \mathcal{I}} E[T(A_q, I_j)]$$

- To establish a lower bound for the expected running time of a randomized algorithm on its worst-case input, it suffices to find an appropriate distribution over inputs and prove that on this random input, no deterministic algorithm can do better than the randomized one.
- The power of this technique lies at the fact that one can choose any distribution over inputs and the lower bound is constructed based on deterministic algorithms.

Yao's MiniMax principle: proof

Proof.

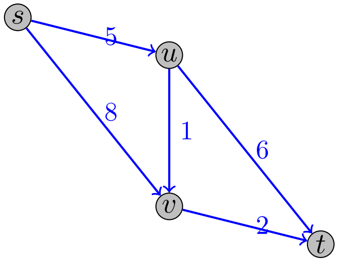
$$\begin{aligned}
\min_{A_i \in \mathcal{A}} E[T(A_i, I_p)] &\leq \max_{u \in \Delta_m} \min_{A_i \in \mathcal{A}} E[T(A_i, I_u)] && (5)\\
&= \max_{u \in \Delta_m} \min_{v \in \Delta_n} E[T(A_v, I_u)] && (6)\\
&= \min_{v \in \Delta_n} \max_{u \in \Delta_m} E[T(A_v, I_u)] && (7)\\
&= \min_{v \in \Delta_n} \max_{I_j \in \mathcal{I}} E[T(A_v, I_j)] && (8)\\
&\leq \max_{I_j \in \mathcal{I}} E[T(A_q, I_j)] && (9)
\end{aligned}$$

□

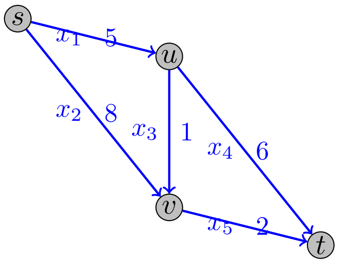
- Here, Δ_n denotes the set of n-dimensional probability vectors.
- Equation (3) follows by the von Neumann's MiniMax theorem.

Application 4: Revisiting SHORTESTPATH algorithm

INPUT: n cities, and a collection of roads. A road from city i to j has a distance $d(i, j)$. Two specific cities: s and t.
OUTPUT: the shortest path from city s to t.

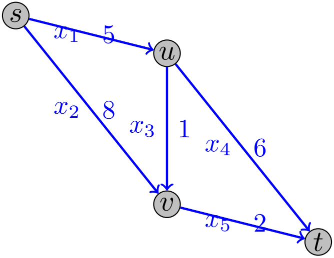


ShorestPath problem: Primal problem



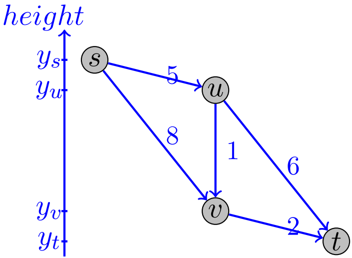
- Primal problem: set variables for **roads** (Intuition: $x_i = 0/1$ means whether edge i appears in the shortest path), and a constraint means that *"we enter a node through an edge and leaves it through another edge"*.

$$
\begin{array}{rrcrcrcrcrcrl}
\min & 5x_1 & + & 8x_2 & + & 1x_3 & + & 6x_4 & + & 2x_5 & & \\
s.t. & x_1 & + & x_2 & & & & & & & = 1 & \text{vertex } s \\
 & & & & & & - & x_4 & - & x_5 & = -1 & \text{vertex } t \\
 & -x_1 & & & + & x_3 & + & x_4 & & & = 0 & \text{vertex } u \\
 & & - & x_2 & - & x_3 & & & + & x_5 & = 0 & \text{vertex } v \\
 & x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & = 0/1 &
\end{array}
$$



- Primal problem: relax the 0/1 integer linear program into linear program by the **totally uni-modular** property.

$$
\begin{array}{lrcrcrcrcrll}
\min & 5x_1 & + & 8x_2 & + & 1x_3 & + & 6x_4 & + & 2x_5 & & \\
s.t. & x_1 & + & x_2 & & & & & & & = 1 & \text{vertex } s \\
 & & & & & & - & x_4 & - & x_5 & = -1 & \text{vertex } t \\
 & -x_1 & & & + & x_3 & + & x_4 & & & = 0 & \text{vertex } u \\
 & & - & x_2 & - & x_3 & & & + & x_5 & = 0 & \text{vertex } v \\
 & x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & \geq 0 & \\
 & x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & \leq 1 &
\end{array}
$$



- DUAL PROBLEM: set variables for **cities**. (Intuition: y_i means the height of city i; thus, $y_s - y_t$ denotes the height difference between s and t, providing a lower bound of the shortest path length.)

$$
\begin{array}{rrrrrrrrrl}
\max & y_s & - & y_t & & & & & & \\
s.t. & y_s & & & - & y_u & & & \leq 5 & x_1 : \text{edge } (s, u) \\
 & y_s & & & & & - & y_v & \leq 8 & x_2 : \text{edge } (s, v) \\
 & & & & & y_u & - & y_v & \leq 1 & x_3 : \text{edge } (u, v) \\
 & & - & y_t & + & y_u & & & \leq 6 & x_4 : \text{edge } (u, t) \\
 & & - & y_t & & & + & y_v & \leq 2 & x_5 : \text{edge } (v, t)
\end{array}
$$