线性规划概述

在给定有限的资源和竞争约束情况下，很多问题都可以表达为最大化或最小化某个目标。如果可以把目标指定为某些变量的一个线性函数，而且如果可以将资源的约束指定为这些变量的等式或不等式，则得到一个**线性规划问题（Linear-Programming Problem）**。

在求解线性规划时又两种有用的格式：**标准型**和**松弛型**。在标准型中所有的约束都是不等式，而在松弛型中所有的约束都是等式。

下面给出一个将实际问题转换为线性规划形式的例子。

> 有m种不同的食物$F_1,...,F_m$，这些食物能够提供n种营养$N_1,...,N_n$，营养N_j每天的最低需求量是c_j，b_i是F_i的单位价格。a_{ij}代表食物F_i单位体积所含的营养N_j。问题是求在满足营养需求下的最小花费。
>
> 假设每种食物的数量为x_i，则使用线性规划的形式可表示为：
>
> $$min\sum_i b_ix_i$$
> $$s.t.\sum_i a_{ij}x_i \geq c_j$$
>
> 这个问题的目标是求满足营养需求的条件下最小化价格，接下来我们会看到，其实它的对偶问题就是最大化营养的需求量。

单纯形算法

解决线性规划问题主要用三种算法：

- **单纯形算法**：指数时间的复杂度，但是在实际中应用广泛，当它被精心实现时，通常能够快速地解决一般的线性规划问题。
- **椭圆算法**：第一个指数时间算法，但是在实际中运行缓慢。
- **内点法**：在理论和实际中都能比较有效率地解决线性规划问题。

本章我们主要讨论在实际问题中应用广泛的单纯形算法。

首先从一个例子开始，考虑下列松弛型的线性规划并将等式重写后可得到一个tableau：

- Consider the following LP

$$
\begin{aligned}
\max \quad & x_1 + x_2 \\
\text{s.t.} \quad & -x_1 + x_2 + x_3 = 1 \\
& x_1 + x_4 = 3 \\
& x_2 + x_5 = 2 \\
& x_1, \dots, x_5 \geq 0
\end{aligned}
$$

- Rewrite equalities as follows. (A tableau.)

$$
\begin{aligned}
x_3 &= 1 + x_1 - x_2 \\
x_4 &= 3 - x_1 \\
x_5 &= 2 - x_2 \\
z &= x_1 + x_2
\end{aligned}
$$

现在，x_3, x_4, x_5是基本解，令$x_1 = x_2 = 0$，可得$x_3 = 1, x_4 = 3, x_5 = 2$，且$z = 0$。

我们当然希望改善z的值，很明显需要增加x_1或者x_2。令$x_1 = 0$，由于$x_1, \dots, x_5 \geq 0$，x_2最大可取至1，此时$x_3 = 0$。现在基本解变为x_2, x_4, x_5，重写tableau：

$$
\begin{aligned}
x_2 &= 1 + x_1 - x_3 \\
x_4 &= 3 - x_1 \\
x_5 &= 1 - x_1 + x_3 \\
z &= 1 + 2x_1 - x_3
\end{aligned}
$$

重复上面的过程，为了增加z的值，我们可以增加x_1（由于x_3的系数是负数，因此增加x_3是无效的）。令$x_3 = 0$，x_1最大可取至1，此时$x_5 = 0$。这时基本解变为x_1, x_2, x_4，重写tableau：

$$\begin{aligned} x_1 &= 1 + x_3 - x_5 \\ x_2 &= 2 - x_5 \\ x_4 &= 2 - x_3 + x_5 \\ z &= 3 + x_3 - 2x_5 \end{aligned}$$

重复上面的过程，为了增加z的值，我们可以增加x_3（由于x_5的系数是负数，因此增加x_5是无效的）。令$x_5 = 0$，x_3最大可取至2，此时x_4变为0。这时基本解变为x_1, x_2, x_3，重写tableau：

$$\begin{aligned} x_1 &= 3 - x_4 \\ x_2 &= 2 - x_5 \\ x_3 &= 2 - x_4 + x_5 \\ z &= 5 - x_4 - x_5 \end{aligned}$$

此时可看出z的取值已达最优，因此解是$x_1 = 3, x_2 = 2, x_3 = 2, x_4 = 0, x_5 = 0$，目标值$z = 5$。

对偶性

对偶性是个非常重要的性质。在一个最优化问题中，一个对偶问题的识别几乎总是伴随着一个多项式时间算法的发现。

在线性规划的形式下，对偶问题可互相转化，具体如下图所示：

- Primal
  $\max\ c^Tx$
  s.t. $Ax \le b$
  $x \ge 0$
- $\max\ c^Tx$
  s.t. $Ax = b$
  $x \ge 0$

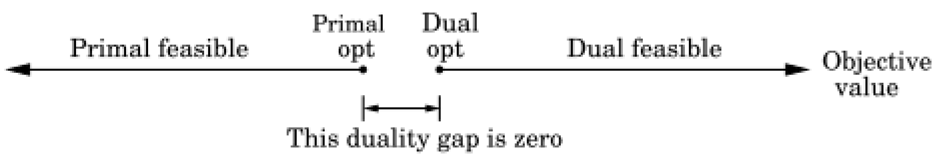
- Dual
  $\min\ b^Ty$
  s.t. $A^Ty \ge c$
  $y \ge 0$
- $\min\ b^Ty$
  s.t. $A^Ty \ge c$

下面给出一个实际例子，照着原问题的线性规划形式，我们即可写出对偶问题的线性规划形式。

> **原问题有多少个未知数，对偶问题就有多少个式子；原问题有多少个式子，对偶问题就有多少个未知数。**

- Primal
  - $\max\ x_1 + 6x_2$
  - s.t. $x_1 \le 200 \quad (1)$
    $x_2 \le 300 \quad (2)$
    $x_1 + x_2 \le 400 \quad (3)$
    $x_1, x_2 \ge 0$
- Dual
  - $\min\ 200y_1 + 300y_2 + 400y_3$
  - s.t. $y_1 + y_3 \ge 1 \quad (1)$
    $y_2 + y_3 \ge 6 \quad (2)$
    $y_1, y_2, y_3 \ge 0$

如下图所示，原问题给出了对偶问题的可行解的下界，对偶问题给出了原问题的可行解的上界。



总结几个经典的对偶问题：

- 最大流的对偶问题是最小割
- 最大匹配的对偶问题是最小顶点覆盖
- 最优匹配的对偶问题是最小定标和

最大流与最小割的线性规划表示

最大流满足两个性质：**反对称性**、**容量限制**。最大流是满足这两个约束和最大化流量值的流，其中流量值是从源流出的总流量。因此，流满足线性约束，且流的值是一个线性函数。可以将最大流问题表示为线性规划并作如下的转换：

LP

$$\max \ \sum_{v:(s,v)\in E} f_{s,v}$$

$$\begin{aligned}
\text{s.t.}\quad & \sum_{u:(u,v)\in E} f_{u,v} = \sum_{u:(v,u)\in E} f_{v,u} && \forall\, v \in V, v \neq s, t && \text{(conservation)}\\
& f_e \leq c_e && \forall\, e \in E && \text{(capacity)}\\
& f_e \geq 0 && \forall\, e \in E &&
\end{aligned}$$

Add an edge (t,s) with infinity capacity.

$$\max \ f_{t,s}$$

$$\begin{aligned}
\text{s.t.}\quad & \sum_{u:(u,v)\in E} f_{u,v} - \sum_{u:(v,u)\in E} f_{v,u} = 0 && \forall\, v \in V && \text{(conservation)}\\
& f_e \leq c_e && \forall\, e \in E && \text{(capacity)}\\
& f_e \geq 0 && \forall\, e \in E &&
\end{aligned}$$

其对偶问题的实际意义是最小割：

Dual:

(define variable l_e and d_u for each edge e and vertex u respectively)

$$\min \ \sum l_e c_e$$

$$\begin{aligned}
\text{s.t.}\quad & d_t - d_s + l_{ts} \geq 1\\
& d_u - d_v + l_{uv} \geq 0 \quad \text{for either } u \text{ or } v \text{ not } s \text{ or } t.
\end{aligned}$$