Elements Of Operation Research

1 Elementary Mathematic Analytics

1.1 Convex functions

Function f is convex if and only if for $\forall 0 \leq t \leq 1$, $x_1, x_2 \in X$,

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$$

Well-known examples of convex functions include $x^\alpha(\alpha \geq 1)$, e^{ax}, $ax+b$, $x \log x$. See below the properties of convex functions.

- A differentiable function f defined on a convex domain is convex if and only if

  $$f(x) \geq f(y) + \nabla f(y) \cdot (x-y)$$

  holds for all x, y in the domain.

- A twice differentiable function is convex on a convex set if and only if its Hessian matrix of second partial derivatives is positive semidefinite on the interior of the convex set.

- Any local minimum of a convex function is also a global minimum. A strictly convex function will have at most one global minimum.

- If X is a random variable taking values in the domain of convex function f, then

  $$E(f(X)) \geq f(E(X))$$

  (Jensen's inequality),where E denotes the mathematical expectation.

- $-f$ is concave if and only if f is convex.

- if $w_1, \ldots, w_n \geq 0$ and $f_1, \ldots, f_n$ are all convex, then so is $w_1 f_1 + \cdots + w_n f_n$. In particular, the sum of two convex functions is convex.

- let $\{f_i\}$be a collection of convex functions. Then $g(x) = \max f_i(x)$ is convex. If $f(x,y)$ is convex in x then $g(x) = \sup_{y \in C} f(x,y)$ is convex even if C is not a convex set . (Danskin's theorem).

- If f and g are convex functions and g is non-decreasing, then $h(x) = g(f(x))$ is convex. As an example, if f is convex, then so is $e^{f(x)}$.

- if f is convex, then so is $g(x) = f(Ax+b)$.

- If $f(x,y)$ is convex in (x,y) then $g(x) = \inf_{y \in C} f(x,y)$ is convex in x, provided that C is a convex set and that $g(x) \neq -\infty$.

2 矩阵代数基础知识

初等矩阵分为3种类型，分别对应着3种不同的行/列变换。

- 两行/列互换rearrangement ：$R_i \leftrightarrow R_j$X。
- 把某行/列乘以一非零常数：$kR_i \rightarrow R_i$，其中$k \neq 0$。
- 把第i行/列加上第j行/列的k倍：$R_i + kR_j \rightarrow R_i$。

初等矩阵即是将上述3种初等变换应用于单位矩阵的结果。设E是某个初等矩阵，那么EA就相当于对A做了对应的初行等变换，而AE就相当于对A做了对应的初列等变换，如下： 如果B可以由A经过

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 7 & 8 & 9 \\ 4 & 5 & 6 \end{pmatrix}$$

(a) 行交换

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 3 & 2 \\ 4 & 6 & 5 \\ 7 & 9 & 8 \end{pmatrix}$$

(b) 列交换

Figure 1: 初等变换示意图

一系列初等变换得到，则称矩阵A与B称为等价。更多内容见这里。

3 矩阵数值运算的误差与效率

4 线性规划基础知识

线性规划问题的目标函数和约束条件都是线性的，Standard Maximum Problem 定义为如下：

$$\begin{aligned} \max_{\boldsymbol{x}} \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t} \quad & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned} \tag{1}$$

其中，$\boldsymbol{x} \in R^n, \boldsymbol{A} \in R^{m \times n}$。注意，这里介绍几个将问题转化成Standard 形式的小技巧。第一个把大于小于混合变成全部小于：

$$\begin{aligned} & a_{11}x_1 + a_{12}x_2 \geq b_1 \\ & a_{21}x_1 + a_{22}x_2 \leq b_2 \\ \Rightarrow \quad & -a_{11}x_1 - a_{12}x_2 \leq -b_1 \qquad \text{两边乘以-1} \\ & a_{21}x_1 + a_{22}x_2 \leq b_2 \\ \Rightarrow \quad & a'_{11}x_1 + a'_{12}x_2 \leq b'_1 \qquad \text{令} a'_{11} = -a_{11}, a'_{12} = -a_{12}, b'_1 = -b_1 \\ & a_{21}x_1 + a_{22}x_2 \leq b_2 \end{aligned}$$

第二个把最小化问题变成最大化问题：

$$\max_{\boldsymbol{x}} \boldsymbol{c}^T \boldsymbol{x} = \min_{\boldsymbol{x}} \boldsymbol{c}^T(-\boldsymbol{x}) = \min_{\boldsymbol{y}} \boldsymbol{c}^T \boldsymbol{y} \quad \text{令} \boldsymbol{y} = -\boldsymbol{x}$$

第三个消除等式约束：

$$
\begin{array}{llll}
 & \max_{x_1,x_2,x_3} & c_1x_1+c_2x_2+c_3x_3 & \\
 & \text{s.t} & a_{11}x_1+a_{12}x_2+a_{13}x_3 & \ge b_1 \\
 & & a_{21}x_1+a_{22}x_2 & \le b_2 \\
 & & a_{32}x_2+a_{33}x_3 & = b_3 \\
 & \max_{x_1,x_2} & c_1x_1+c_2x_2+\frac{c_3b_3}{a_{33}}-\frac{c_3a_{32}x_2}{a_{33}} & \\
\Rightarrow & \text{s.t} & a_{11}x_1+a_{12}x_2+\frac{a_{13}b_3}{a_{33}}-\frac{a_{13}a_{32}x_2}{a_{33}} & \ge b_1 \\
 & & a_{21}x_1+a_{22}x_2 & \le b_2 \\
 & \max_{x_1,x_2} & c_1x_1+(c_2-\frac{c_3a_{32}}{a_{33}})x_2+\frac{c_3b_3}{a_{33}} & \\
\Rightarrow & \text{s.t} & a_{11}x_1+(a_{12}-\frac{a_{13}a_{32}}{a_{33}})x_2 & \ge b_1-\frac{a_{13}b_3}{a_{33}} \\
 & & a_{21}x_1+a_{22}x_2 & \le b_2 \\
 & \max_{x_1,x_2} & c_1x_1+c'_2x_2 & \\
\Rightarrow & \text{s.t} & a_{11}x_1+a'_{12}x_2 & \ge b'_1 \\
 & & a_{21}x_1+a_{22}x_2 & \le b_2
\end{array}
$$

由等式约束可得$x_3=\frac{b_3-a_{32}x_2}{a_{33}}$，把这个带入

目标函数中的常数项不影响解，去掉

其中，$c'_2=(c_2-\frac{c_3a_{32}}{a_{33}}),a'_{12}=(a_{12}-\frac{a_{13}a_{32}}{a_{33}}),b'_1=b_1-\frac{a_{13}b_3}{a_{33}}$。第四个为无非负约束的变量加入非负约束：如果$x_j$没有限制为非负，那么令$x_j=u_j-v_j,u_j\ge 0,v_j\ge 0$，这样做的根据是任意一个数都一定可以被两个非负的数相减得到。

问题1的对偶问题为：

$$
\begin{array}{ll}
\min_{\boldsymbol{y}} & \boldsymbol{b}^T\boldsymbol{y} \\
\text{s.t} & \boldsymbol{A}^T\boldsymbol{y}\ge\boldsymbol{c} \\
 & \boldsymbol{y}\ge\boldsymbol{0}
\end{array} \tag{2}
$$

同时问题2的对偶问题正是问题1。

Theorem 4.1 (Weak Duality Theorem) *如果$\boldsymbol{x}$是问题1的可行解，$\boldsymbol{y}$是问题2的可行解，那么有：*

$$\boldsymbol{c}^T\boldsymbol{x}\le\boldsymbol{b}^T\boldsymbol{y}$$

Proof 1

$$
\begin{array}{ll}
\boldsymbol{c}^T\boldsymbol{x} & \\
=\boldsymbol{x}^T\boldsymbol{c} & \\
\le\boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{y} & \text{带入问题2的约束} \\
\le\boldsymbol{b}^T\boldsymbol{y} & \text{带入问题1的约束}
\end{array}
$$

由此可以推出

Corollary 4.1 *如果$\boldsymbol{x}^*$是问题1的可行解，$\boldsymbol{y}^*$是问题2的可行解，且：*

$$\boldsymbol{c}^T\boldsymbol{x}^*=\boldsymbol{b}^T\boldsymbol{y}^*$$

那么$\boldsymbol{x}^*$是问题1的最优解，$\boldsymbol{y}^*$是问题2的最优解，

Proof 2 *对任意一个可行解$\boldsymbol{x}$，有*

$$
\begin{array}{l}
\boldsymbol{c}^T\boldsymbol{x} \\
\le\boldsymbol{b}^T\boldsymbol{y}^* \\
=\boldsymbol{c}^T\boldsymbol{x}^*
\end{array}
$$

从而，$\boldsymbol{x}^*$是最优解。

Theorem 4.2 (The Strong Duality Theorem) 下面三个结论相互等价：

- 问题*1*的解存在.
- 问题*2*的解存在.
- 问题*1*的最优值等于问题*2*的最优值。

其证明过程可以参考这里。

Proof 3

Theorem 4.3 (The Complementary Slackness Theorem) $\boldsymbol{x}$和$\boldsymbol{y}$分别是原问题和对偶问题的最优解，当且仅当：

$$\text{如果}j\text{满足：}\sum_{i=1}^{m} y_i a_{ij} > c_j\text{，那么，}x_j = 0 \tag{3}$$

$$\text{如果}i\text{满足：}\sum_{j=1}^{n} x_j a_{ij} < b_i\text{，那么，}y_i = 0 \tag{4}$$

Proof 4 如果$\boldsymbol{x}$和$\boldsymbol{y}$分别是原问题和对偶问题的最优解，那么根据*Theorem4.1* 的证明过程我们有：

$$\begin{aligned} & \boldsymbol{x}^T\boldsymbol{c} = \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{y} \\ \Rightarrow\ & 0 = \boldsymbol{x}^T(\boldsymbol{A}^T\boldsymbol{y} - \boldsymbol{c}) \\ \Rightarrow\ & 0 = \textstyle\sum_{j=1} x_j(\sum_{i=1} a_{ij}y_i - c_j) \end{aligned}$$

根据问题对偶问题的约束条件，$(\sum_{i=1} a_{ij}y_i - c_j) \geq 0$，而原问题的约束中有$x_j \geq 0$，所以要是的上式成立，则必有

$$x_j(c_j - \sum_{i=1} a_{ij}y_i) = 0 \tag{5}$$

，进而：

$$x_j = 0 \quad \text{或者} \quad \sum_{i=1} a_{ij}y_i - c_j = 0$$

类似地，我们可以得到：

$$y_i = 0 \quad \text{或者} \quad \sum_{j=1} a_{ij}x_j - b_i = 0$$

从而我们得证*3*和*4*。

如果*3*成立，那么就有：

$$\begin{aligned} & 0 = \textstyle\sum_{j=1} x_j(\sum_{i=1} a_{ij}y_i - c_j) \\ \Rightarrow\ & 0 = \boldsymbol{x}^T(\boldsymbol{A}^T\boldsymbol{y} - \boldsymbol{c}) \\ \Rightarrow\ & \boldsymbol{x}^T\boldsymbol{c} = \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{y} \end{aligned}$$

如果*4*成立，类似地有：

$$\begin{aligned} & 0 = \textstyle\sum_{i=1} y_i(\sum_{j=1} a_{ij}x_j - b_i) \\ \Rightarrow\ & 0 = \boldsymbol{y}^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) \\ \Rightarrow\ & \boldsymbol{y}^T\boldsymbol{b} = \boldsymbol{y}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{y} \end{aligned}$$

所以，我们有

$$\boldsymbol{b}^T\boldsymbol{y} = \boldsymbol{x}^T\boldsymbol{c} = \boldsymbol{c}^T\boldsymbol{x}$$

于是他们便是最优解。

5 Simplex Method

We'll start by explaining the "easy case" of the Simplex Method: when you start with a linear program in standard form where all the right-hand sides of the constraints are non-negative.

Roughly speaking, you turn the LP into a dictionary, and then repeatedly pivot to get new dictionaries until at some point the numbers in the dictionary indicate you are done. We'll illustrate the procedure with the following example:

$$
\begin{array}{ll}
\max & x_1 + 2x_2 - x_3 \\
\text{s.t} & 2x_1 + x_2 + x_3 \leq 14 \\
 & 4x_1 + 2x_2 + 3x_3 \leq 28 \\
 & 2x_1 + 5x_2 + 5x_3 \leq 30 \\
 & x_1, x_2, x_3 > 0
\end{array}
$$

The first step will be to introduce slack variables:

$$
\begin{array}{ll}
\max & x_1 + 2x_2 - x_3 \\
\text{s.t} & 2x_1 + x_2 + x_3 + x_4 = 14 \\
 & 4x_1 + 2x_2 + 3x_3 + x_5 = 28 \\
 & 2x_1 + 5x_2 + 5x_3 + x_6 = 30 \\
 & x_1, x_2, x_3, x_4, x_5, x_6 > 0
\end{array}
$$

Solving for the slack variables we get the initial dictionary for our problem:

$$
\begin{array}{llllll}
x_4 & = 14 & -2x_1 & -x_2 & -x_3 \\
x_5 & = 28 & -4x_1 & -2x_2 & -3x_3 \\
x_6 & = 30 & -2x_1 & -5x_2 & -5x_3 \\
z & = 0 & +x_1 & +2x_2 & -x_3
\end{array}
$$

The variables on the left x_4, x_5, x_6 are the basic variables for this dictionary; the set of these variables is called the basis. (In the initial dictionary the basic variables are the slack variables, that changes after pivoting.) Notice that at the bottom we've added a variable for the objective function.

The LP we want to solve is equivalent to "maximize z subject to the equations in the dictionary and to positivity constraints for all". To any dictionary there is an associated solution of the system of equations: just set all the non-basic variables to zero and compute the values of the basic variables. For the above dictionary the associated solution is

$$x_1 = 0, x_2 = 0, x_3 = 0, x_4 = 14, x_5 = 28, x_6 = 30$$

The solution associated to a dictionary certainly satisfies all of the constraints that are equations, but may fail to satisfy the positivity constraints. When it does also satisfy the positivity constraints it is said to be a feasible solution and the dictionary is called feasible too.

The main idea of the Simplex Method is to go from dictionary to dictionary by exchanging a basic variable for a non-basic one, in such a way that:

- The objective function z increases at each step.
- The dictionary is feasible at every step.

In the dictionary we have, the z-row is $z = 0 + x_1 + 2x_2 - x_3$. Since x_1 and x_2 have positive coefficients, increasing them would increase z (注意，他们二者还满足非负约束). So we will choose either x_1 or x_2 to enter the basis. Either one would be a valid choice and make some progress toward solving the LP, but we'll use a definite rule to pick one:among the non-basic variables that have a positive coefficient in the objective function, choose the one with the largest coefficient.

We've settled on putting x_2 in the basis and now we need to decide which basic variable to swap it with. The solution associated to our current dictionary has $x_2 = 0$. The idea now is to increase the value of x_2 as much as possible while keeping the resulting dictionary feasible. Let's focus on how the values of the basic variables depend on (keeping $x_1 = x_3 = 0$):

$$\begin{aligned} x_4 &= 14 \quad -x_5 \\ x_5 &= 28 \quad -2x_2 \\ x_6 &= 30 \quad -5x_2 \\ z &= 0 \quad\; +2x_2 \end{aligned}$$

We see that:

- To keep $x_4 \geq 0$, we can only increase x_2 up to 14 .
- To keep $x_5 \geq 0$, we can only increase x_2 up to 14.
- To keep $x_6 \geq 0$, we can only increase x_2 up to 6 .

So, to keep all variables non-negative, we can only go up to 6. Since x_6 had the strictest requirement on the entering variable , we pick x_6 as the exiting variable. 我们已经决定把x_2选入basis set，我们想让x_2尽可能地大从而让目标函数增长最多，但是同时还需要使得每个变量满足非负约束。我们于是看看之前的哪个basis 变量会在我们加大x_2时最先到达0，那么久逐出这个basis 变量。

Now we get the next dictionary by:

- Solving for x_2 in the equation for x_6. 也就是用新的非基变量x_6, x_1, x_3表示新的基变量x_2，得：

$$2x_1 + 5x_2 + 5x_3 + x_6 = 30 \Rightarrow x_2 = 6 - 2/5x_1 - x_3 - 1/5x_6$$

- Plugging that value for x_2 into the equations for the other basic variables and the equation for the objective function.

$$4x_1 + 2x_2 + 3x_3 + x_5 = 28 \Rightarrow x_5 = 16 - 16/5x_1 - x_3 + 2/5x_6 \quad \text{带入}x_2\text{的新表达式}$$

同样地，

$$\begin{aligned} &2x_1 + x_2 + x_3 + x_4 = 14 &&\Rightarrow x_4 = 8 - 8/5x_1 + 0 * x_3 + 1/5x_6 \\ &z = x_1 + 2x_2 - x_3 &&\Rightarrow z = 12 + 1/5x_1 - 3x_3 - 2/5x_6 \end{aligned}$$

The associated solution has $x_1 = x_3 = x_6 = 0, x_2 = 6, x_4 = 8, x_5 = 16$ and $z = 12$. This is a feasible solution so we learn that getting $z = 12$ is possible, and thus the maximum is 12 or larger.

The solution turned out feasible again by virtue of the way we picked the exiting variable. If we had, for example, decided to make x_2 enter but x_4 exit we would have gotten the following dictionary:

$$\begin{aligned} x_2 &= 14 \quad -2x_1 \quad -x_3 \quad -x_4 \\ x_5 &= 0 \qquad\qquad\quad -x_3 \quad +2x_4 \\ x_6 &= -40 \quad +8x_1 \qquad\quad +5x_4 \\ z &= 28 \quad -3x_1 \quad -3x_3 \quad -2x_4 \end{aligned}$$

The associated solution has $x_6 = 40$ violating the positivity constraint. If that ever happens, you'll know you made a mistake somewhere!从基变量中剔除一个，然后从非基变量选一个作为新的基变量，这个过程叫Pivoting。

When dictionary has all of the coefficients in the z-row negative, we can't pick a new entering variable. This means we are done with this problem.

为什么simplex method是正确的呢？我们首先需需要知道LP的可行域是一个凸集，并且最优点一定是凸集的某个顶点，我们每一次Pivoting就意味着我们从一个顶点换到另外一个顶点。但问题是我们怎么证明这样换就一定能增加目标函数值呢？而且怎么证明一次Pivoting就是一次顶点的切换呢？总共有多少个顶点呢？这些都需要搞明白。

simplex method 的表格法表达如下：

- step0. 首先选择一组基比如$x_1, x_2, \dots, x_m$(我们说这些变量对应的A中的列形成矩阵A_B为基矩阵)，他们一般为每个不等式的松弛变量，当然也可以是其他的，那么形成下面的初始表格(可能需要做一些初等变换将基矩阵单位化)：

$Basis$	x_1	x_2	$\dots$	$x_m,$	x_{m+1}	$\dots$	x_{m+n}	RHS	$ratio$
x_1	1	0	$\dots$	0	$a_{1,m+1}$	$\dots$	$a_{1,m+n}$	b_1	
x_2	0	1	$\dots$	0	$a_{2,m+1}$	$\dots$	$a_{2,m+n}$	b_2	
$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
x_m	0	0	$\dots$	1	$a_{m,m+1}$	$\dots$	$a_{m,m+n}$	b_m	
z	0	0	$\dots$	0	$c_{1,m+1}$	$\dots$	$c_{1,m+n}$	0	

- step1.如果最下面所有的$c_j \le 0$，则算法结束。否则选择$q = \arg\max_{j, c_j > 0} c_j$，对应列的$x_q$就是进入变量。

- step2.计算每行的比率:

$$\frac{b_i}{a_{iq}}$$

并计算这个比率自小值：

$$\frac{b_p}{a_{pq}} = \min_i \{\frac{b_i}{a_{iq}}, a_{iq} > 0\}$$

注意，如果没有i满足$a_{iq} > 0$，那么该问题无界。

$Basis$	x_1	x_2	$\dots$	x_p	$\dots$	x_m	x_{m+1}	$\dots$	x_q	$\dots$	x_{m+n}	RHS	$ratio$
x_1	1	0	$\dots$	0	$\dots$	0	$a_{1,m+1}$	$\dots$	a_{1q}	$\dots$	$a_{1,m+n}$	b_1	b_1/a_{1q}
x_2	0	1	$\dots$	0	$\dots$	0	$a_{2,m+1}$	$\dots$	a_{2q}	$\dots$	$a_{2,m+n}$	b_2	b_2/a_{2q}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_p	0	0	$\dots$	1	$\dots$	0	$a_{p,m+1}$	$\dots$	$\boxed{a_{pq}}$	$\dots$	$a_{p,m+n}$	b_p	$\boxed{b_p/a_{pq}}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_m	0	0	$\dots$	0	$\dots$	1	$a_{m,m+1}$	$\dots$	a_{mq}	$\dots$	$a_{m,m+n}$	b_m	b_m/a_{mq}
z	0	0	$\dots$	0	$\dots$	0	$c_{1,m+1}$	$\dots$	$\boxed{c_q}$	$\dots$	$c_{1,m+n}$	0	

- step3.把x_p逐出基变量，换成x_q:

$Basis$	x_1	x_2	$\dots$	$\boxed{x_q}$	$\dots$	x_m	x_{m+1}	$\dots$	$\boxed{x_p}$	$\dots$	x_{m+n}	RHS	$ratio$
x_1	1	0	$\dots$	$\boxed{a_{1q}}$	$\dots$	0	$a_{1,m+1}$	$\dots$	$\boxed{0}$	$\dots$	$a_{1,m+n}$	b_1	
x_2	0	1	$\dots$	$\boxed{a_{2q}}$	$\dots$	0	$a_{2,m+1}$	$\dots$	$\boxed{0}$	$\dots$	$a_{2,m+n}$	b_2	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
$\boxed{x_p}$	0	0	$\dots$	$\boxed{a_{pq}}$	$\dots$	0	$a_{p,m+1}$	$\dots$	$\boxed{1}$	$\dots$	$a_{p,m+n}$	b_p	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
x_m	0	0	$\dots$	$\boxed{a_{mq}}$	$\dots$	1	$a_{m,m+1}$	$\dots$	$\boxed{0}$	$\dots$	$a_{m,m+n}$	b_m	
z	0	0	$\dots$	$\boxed{c_q}$	$\dots$	0	$c_{1,m+1}$	$\dots$	$\boxed{0}$	$\dots$	$c_{1,m+n}$	0	

我们可以看出，此时矩阵正是所谓的eta矩阵，然后重新经过变换把基矩阵单位化(这个新的基矩阵只有一列不是单位向量)，最后goto step1。

6 Revised Simplex Method

现在我们看看Revised Simplex Method，参考课件在这里。

我们现在再用表格法看看单纯性算法。首先问题如下：

$$\begin{array}{lll} \max & \sum_j c_j x_j & \\ \text{s.t} & \sum_{j=1}^n a_{ij}x_j \le b_i, & i=1,\dots,m \\ & x_j \ge 0, & j=1,\dots,n \end{array} \xrightarrow{\text{为}m\text{个不等式加入}m\text{个松弛变量}} \begin{array}{ll} \max & c^T x \\ \text{s.t} & Ax=b \\ & x \ge 0 \end{array}$$

其中，$x \in R^{n+m}$,$A \in R^{m\times(n+m)}$，也就是有m个约束，m个松弛变量，n个原生变量。

首先我们选择m个初始基(因为有m个等式约束)，比如说我们初始基为$B=\{1,2,\dots,m\}$，那么初始非基便是$N=\{m+1,\dots,m+n\}$，A_B便是基对应的列形成的矩阵，$[A_B]_{ij}=a_{ij}, j\in B, i=1,\dots,m$，类似地定义$A_N, c_N, c_B$。初始表格如下：

$$\left[\begin{array}{c:c:c} A_N & A_B & b \\ \hdashline c_N^T & c_B^T & 0 \end{array}\right]$$

表格每个元素的意义为：

$$\left[\begin{array}{c:c:c} \text{非基的约束系数矩阵} & \text{基的约束系数矩阵} & \text{当前RHS(right-hand-side)} \\ \hdashline \text{非基的目标函数系数} & \text{基的目标函数系数} & \text{当前目前函数取值} \end{array}\right]$$

因为

$$\begin{aligned} & Ax = A_N x_N + A_B x_B = b \\ \Rightarrow\ & A_B x_B = b - A_N x_N \\ \Rightarrow\ & x_B = A_B^{-1}b - A_B^{-1}A_N x_N \\ \Rightarrow\ & A_B^{-1}A_N x_N + I x_B = A_B^{-1}b \end{aligned}$$

其中I是单位矩阵，注意A_B一定可逆，否则就说明我们选错了基。把x_B带入目标函数：

$$\begin{aligned} & c^T x = c_N^T x_N + c_B^T x_B \\ = \ & c_N^T x_N + c_B^T(A_B^{-1}b - A_B^{-1}A_N x_N) \\ = \ & (c_N^T - c_B^T A_B^{-1}A_N)x_N + c_B^T A_B^{-1}b \end{aligned}$$

我们令非基变量取值为0，那么$c_B^T A_B^{-1} b$就是我们当前的目标函数取值。此时我们看到，当我们求出基的解x_B后，约束等式和目标函数的变化为：

$$
\begin{array}{lll}
A_N x_N + A_B x_B = b & \Longrightarrow & A_B^{-1} A_N x_N + I x_B = A_B^{-1} b \\
c_N^T x_N + c_B^T x_B & \Longrightarrow & (c_N^T - c_B^T A_B^{-1} A_N) x_N + c_B^T A_B^{-1} b
\end{array}
$$

换个展现方式：

$$
\begin{array}{lll}
A_N & \Longrightarrow & A_B^{-1} A_N \\
A_B & \Longrightarrow & I \\
b & \Longrightarrow & A_B^{-1} b \\
c_N^T & \Longrightarrow & c_N^T - c_B^T A_B^{-1} A_N
\end{array}
$$

所以我们的表格变为：

$$
\left[\begin{array}{c|c|c}
A_B^{-1} A_N & I & A_B^{-1} b \\
\hline
c_N^T - c_B^T A_B^{-1} A_N & 0 & c_B^T A_B^{-1} b
\end{array}\right]
$$

我们举个例子:

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

我们初始选择x_1, x_2, x_4作为基，初始表格如下：

$$
\left[\begin{array}{c|c|c|c|c|c}
x_1 & x_2 & x_3 & x_4 & x_5 & b \\
\hline
-1 & 1 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 3 \\
0 & 1 & 0 & 0 & 1 & 2 \\
\hline
1 & 1 & 0 & 0 & 0 & 0
\end{array}\right]
$$

对这个表格做变换，使得基对应的系数矩阵你成为单位阵：

$$
\left[\begin{array}{c|c|c|c|c|c}
x_1 & x_2 & x_3 & x_4 & x_5 & b \\
\hline
-1 & 1 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 3 \\
0 & 1 & 0 & 0 & 1 & 2 \\
\hline
1 & 1 & 0 & 0 & 0 & 0
\end{array}\right]
$$

$$
\Rightarrow \left[\begin{array}{c|c|c|c|c|c}
x_1 & x_2 & x_3 & x_4 & x_5 & b \\
\hline
1 & -1 & -1 & 0 & 0 & -1 \\
0 & 1 & 1 & 1 & 0 & 4 \\
0 & 1 & 0 & 0 & 1 & 2 \\
\hline
0 & 2 & 1 & 0 & 0 & 1
\end{array}\right]
$$

先把x_1那一列单位向量化，第一行依次加到第二行和第四行，最后第一行乘以-1

$$
\Rightarrow \left[\begin{array}{c|c|c|c|c|c}
x_1 & x_2 & x_3 & x_4 & x_5 & b \\
\hline
1 & 0 & 0 & 1 & 0 & 3 \\
0 & 1 & 1 & 1 & 0 & 4 \\
0 & 0 & -1 & -1 & 1 & -2 \\
\hline
0 & 0 & -1 & -1 & -2 & -7
\end{array}\right]
$$

在把x_2那一列单位向量化，第二行加到第一行，乘以-1后加第三行，乘以-2后加到第4行

$$
\Rightarrow \left[\begin{array}{c|c|c|c|c|c}
x_1 & x_2 & x_3 & x_4 & x_5 & b \\
\hline
1 & 0 & -1 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 2 \\
0 & 0 & 1 & 1 & -1 & 2 \\
\hline
0 & 0 & 1 & 0 & -2 & 3
\end{array}\right]
$$

在把x_4那一列单位向量化，第三行加到第一行和第二行，乘以-1后加到第四行

现在我们从非基变量中选择一个进入基变量，我们看看哪个非基变量的目标函数系数最大就选择哪个那么它既是候选基变量，当前非基变量的目标函数系数向量为：$c_N^T - c_B^T A_B^{-1} A_N$，上面的例子中我们选择$x_3$，他现在目标函数系数为1。然后我们用$x_3$去表示当前基变量，其他非基变量置0，则有：

$$x_B = A_B^{-1}b - A_B^{-1}A_N x_N \Rightarrow x_B = A_B^{-1}b - A_B^{-1}A_N \left[\begin{array}{c} 0,\ldots,0,1,0,\ldots,0 \end{array}\right]^T$$

然后看看随着候选基变量(x_3)的增长哪个基变量最先达到0，那么就把它剔除基变量。然后新的A_B就是直接用入选基变量的系数列替换被逐出的变量的系数列即可。

我们再次复述一遍我们的操作流程：

- 选定基集合B。
- 选择进入的变量(使之进入基集合)，需要计算$c_N^T - c_B^T A_B^{-1} A_N$ ，而他的计算方法为：求解方程:

$$y^T A_B = c_B^T \tag{6}$$

  ，得到$y^T = c_B^T A_B^{-1}$，然后计算$c_N^T - y^T A_N$。

- 计算退出变量(使之从基集合中退出)，我们用选择的进入变量表达现在每个基变量，需要计算：$A_B^{-1}b - A_B^{-1}A_N x_N$，令$x_N = [0,\ldots,0,1,0,\ldots,0$，也就是让进入变量为1，其他变量为0，求解方程：

$$A_B d = a \tag{7}$$

  ，其中$a = A_N x_N$正是进入变量对应的列。然后计算$A_B^{-1}b - d$看看哪个变量最先达到零，就是我们的退出变量。。

- 更新B，进而更新A_B，仅仅需要把其中退出变量对应的列换成进入变量对应列即可。

我们看到这里有两个方程组需(6和7)要求解。一般这都需要计算A_B^{-1}，但是我们应该避免直接求逆矩阵，这样不但计算量太大，而且会存在数值计算精度问题。

我们现在用B代表A_B，B_k表示第k次pivoting 对应的A_B。方程(6和7)变为：

$$y^T B_k = c_B^T \tag{8}$$

$$B_k d = a \tag{9}$$

我们已经说过了B_k是将B_{k-1}中某一列替换得到的，那一列对应的正是被逐出的变量，换入的是新的基变量，所以：

$$B_k = B_{k-1}E_k$$

E_k是eta 矩阵，只有某一列有值，其他列都是单位向量：

$$\begin{bmatrix} 1 & 0 & \ldots & 0 & \eta_1 & 0 & \ldots 0 \\ 0 & 1 & \ldots & 0 & \eta_2 & 0 & \ldots 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & 0 & \ldots 0 \\ 0 & 0 & \vdots & \vdots & \eta_p & 0 & \ldots 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ldots \vdots \\ 0 & 0 & \ldots & \ldots & \eta_m & 0 & \ldots 1 \end{bmatrix}$$

我们有：

$$[BE]_{ij} = \begin{cases} \sum_k B_{ik}\eta_k & j = p \\ B_{ij} & j \neq p \end{cases}$$

可以看到右乘eta 矩阵只会改变相应的列。所以只要我们在B_{k-1}右乘合适的eta 矩阵就实现了换基(pivoting)。初始我们的基矩阵是B_0的话，那么：

$$B_k = B_0 E_1 E_2 \dots E_k$$

所以，方程8变为：

$$y^T B_k = c_B^T \Rightarrow y^T B_0 E_1 E_2 \dots E_k = c_B^T \Rightarrow (((y^T B_0 E_1) E_2) \dots E_k) = c_B^T \tag{10}$$

方程9变为：

$$B_k d = a \Rightarrow B_0 E_1 E_2 \dots E_k d = a \Rightarrow (B_0(E_1(E_2 \dots (E_k d)))) = a \tag{11}$$

我们现在看下LU分解，它将一个方阵分解为一个下三角矩阵和一个上三角矩阵的乘积。在来解线性方程组求逆矩阵等运算中都是一个关键的步骤。

$$A = LU$$

一个例子如下：

$$\begin{bmatrix} a_1 1 & a_1 2 & a_3 3 \\ a_2 1 & a_2 2 & a_2 3 \\ a_3 1 & a_3 2 & a_3 3 \end{bmatrix} = \begin{bmatrix} l_1 1 & 0 & 0 \\ l_2 1 & l_2 2 & 0 \\ l_3 1 & l_3 2 & l_3 3 \end{bmatrix} \begin{bmatrix} u_1 1 & u_1 2 & u_3 3 \\ 0 & u_2 2 & u_2 3 \\ 0 & 0 & u_3 3 \end{bmatrix}$$

LU分解在本质上是高斯消元法的一种表达形式。从下至上地对矩阵A做初等行变换，将对角线左下方的元素变成零，然后再证明这些行变换的效果等同于左乘一系列单位下三角矩阵，这一系列单位下三角矩阵的乘积的逆就是L矩阵，它也是一个单位下三角矩阵。令$A^{(0)} = [a_{ij}^{(0)}] = A$，在第$n$步对消去矩阵$A^{(n-1)}$的第$n$列主对角线下的元素：

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & \dots & \dots & \dots \\ 0 & a_{22} & a_{23} & \dots & \dots & \dots & \dots \\ 0 & 0 & a_{33} & \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \ddots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & a_{nn} & a_{n,n+1} & \dots \\ 0 & 0 & 0 & \dots & a_{n+1,n} & a_{n+1,n+1} & \dots \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \dots \\ 0 & 0 & 0 & \dots & a_{mn} & a_{m,n+1} & \dots \end{bmatrix}$$

可以看出，只要执行下面的操作即可：

$$R_i = -\frac{a_{in}}{a_{nn}} R_n + R_i, i = n+1, \dots, m$$

也就是行n乘以$-\frac{a_{in}}{a_{nn}}$后加到行i上。a_{nn}叫做pivot。而这个操作相当于$A^{(n-1)}$左乘一个所谓的单位下三角矩阵：

$$L_n = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & 0 & \dots 0 \\ 0 & 1 & \dots & 0 & 0 & 0 & \dots 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 & \dots 0 \\ 0 & 0 & \vdots & \vdots & 1 & 0 & \dots 0 \\ 0 & 0 & \vdots & \vdots & -\frac{a_{n+1,n}}{a_{nn}} & 0 & \dots 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \vdots \\ 0 & 0 & \dots & \dots & -\frac{a_{m,n}}{a_{nn}} & 0 & \dots 1 \end{bmatrix}$$

$$A^{(n)} = L_n A^{(n-1)}$$

可以看到L_n也是一个eta矩阵。经过$m-1$次这样的操作，最后我们就能得到一个上三角矩阵$A^{(m-1)}$。

$$\begin{aligned}
A &= L_1^{-1} L_1 A^{(0)} \\
&= L_1^{-1} A^{(1)} \\
&= L_1^{-1} L_2^{-1} L_2 A^{(1)} \\
&= L_1^{-1} L_2^{-1} A^{(2)} \\
&\vdots \\
&= L_1^{-1} L_2^{-1} \ldots L_{m-1}^{-1} A^{(m-1)}
\end{aligned}$$

令$L = L_1^{-1} L_2^{-1} \ldots L_{m-1}^{-1}, U = A^{(m-1)}$。下三角矩阵的逆依然是下三角矩阵，而且下三角矩阵的乘积仍是下三角矩阵，所以L是下三角矩阵。显然，要使得算法成立，在每步操作时必须有$a_{n,n} \neq 0$，如果这一条件不成立，就要将第n行和另一行交换，就相当于左乘对应的初等矩阵，就是一个所谓的permutation 矩阵P，例如，交换第一行和第二行对应的初等矩阵：

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

对于任何permutation 矩阵P，有：

$$P^{-1} = P^T$$

我们现在先把B_0变成上三角矩阵：

$$L_m P_m \ldots L_2 P_2 L_1 P_1 B_0 = U$$

我们这里左成P是为了应对$a_{nn} = 0$的情况，如果$a_{nn} \neq 0$，那么P就是单位矩阵。而上三角矩阵可以分解为一些了单位上三角矩阵的乘积：

$$U = U_m U_{m-1} \ldots U_1$$

因为我们一定可以把一个上三角阵通过行变换变成单位矩阵，而每个行变换就相等于左乘了一个eta矩阵，而eta矩阵的逆还是一个eta矩阵：

$$E_m \ldots E_2 E_1 U = I, \Rightarrow U = E_1^{-1} E_2^{-1} \ldots E_m^{-1}$$

例如：

$$\begin{bmatrix} 1/a_{11} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -a_{12}/a_{22} & 0 \\ 0 & 1/a_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -a_{13}/a_{33} \\ 0 & 1 & -a_{23}/a_{33} \\ 0 & 0 & 1/a_{33} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

那么：

$$\begin{aligned}
& B_k = B_0 E_1 E_2 \ldots E_k \\
\Rightarrow\ & L_m P_m \ldots L_2 P_2 L_1 P_1 B_k \\
& = L_m P_m \ldots L_2 P_2 L_1 P_1 B_0 E_1 E_2 \ldots E_k \\
& = U E_1 E_2 \ldots E_k \\
& = U_m U_{m-1} \ldots U_1 E_1 E_2 \ldots E_k \\
\Rightarrow\ & B_k = (L_m P_m \ldots L_2 P_2 L_1 P_1)^{-1} U_m U_{m-1} \ldots U_1 E_1 E_2 \ldots E_k \\
& = L^{-1} U
\end{aligned}$$

我们令$L_mP_m\dots L_2P_2L_1P_1 = L, U_mU_{m-1}\dots U_1E_1E_2\dots E_k = U$，所以我们可以通过一系列矩阵乘法实现求逆，而且这些矩阵都是简单的矩阵，他们大部分元素都是为0，可以被稀疏存储。方程8变为：

$$y^TB_k = c_B^T \Rightarrow y^TL^{-1}U = c_B^T$$

那么：

$$wU = c_B^T, w = y^TL^{-1}, y = wL$$

所以可以先求解w，然后运算wL就能得到8的解。求解方程8叫做backward transformation (BTRAN)，求解方程9叫做forward transformation (FTRAN)。

7 interior point method (IPM)

8 dual steepest-edge (DSE)

9 bound flipping ratio test (BFRT)

10 sequential dual simplex

$$\hat{x} = E^{-1}x \Rightarrow \hat{x}_i = \sum_j E_{ij}^{-1}x_j \Rightarrow \hat{x}_i = \begin{cases} x_j/\eta_p & i = p \\ x_i - \eta_i/\eta_p x_p & i \neq p \end{cases}$$

令$\eta = [\eta_1, \dots, \eta_{p-1}, 0, \eta_{p+1}, \dots, \eta_m]$

11 parallelization

12 多核编程概述

13 CLP源码要点

References