一个 Bottleneck 问题及其算法*

罗　宗　俊
（贵州民族学院数学系）

A BOTTLENECK PROBLEM AND ITS ALGORITHM

Luo Zong-jun
(Department of Mathematics, Quzhou National Minorities College)

Abstract

In this paper we discuss the following mathematical model I:

Find an $X=(x_1, x_2, \cdots, x_n)$ satisfying the constraints

$$\begin{cases}\sum_{j=1}^{n} x_j = m \ (m \geqslant n \text{ integer}) \\ x_j \geqslant 1 \text{ integer}, \ j=1, 2, \cdots, n\end{cases}$$

such that the objective function

$$y = \min_{1 \leqslant j \leqslant n} \{C_j x_j\}$$

achieves the maximum, where C_j $(j=1, 2, \cdots, n)$ are positive constants. Without loss of generality, we may assume that $c_1 \leqslant c_2 \leqslant \cdots \leqslant c_n$.

The main result is:

Theorem 1. For the model I there exists necessarily an optimal solution $X=(x_1, x_2, \cdots, x_n)$ satisfying the following condition

$$c_k(x_k - 1) \leqslant \min_{1 \leqslant j \leqslant n} \{C_j x_j\} \quad k=1, 2, \cdots, n. \tag{1}$$

Moreover, every feasible solution satisfying condition (1) is necessarily an optimal solution.

A procedure of the quasi-polynomial algorithm is established for finding an optimal solution to the model I.

在文[1]中，提出了下面的数学模型：

模型 I. 求一 $X=(x_1, x_2, \cdots, x_n)$ 满足下列约束条件

$$\begin{cases}\sum_{j=1}^{n} x_j = m \ (m \geqslant n \text{ 且为整数}), \\ x_j \geqslant 1 \text{ 且为整数}, \ j=1, 2, \cdots, n,\end{cases}$$

* 1983 年 4 月 22 日收到.

使目标函数 $y=\min\limits_{1\leqslant i\leqslant n}\{c_ix_i\}$ 取最大值，其中 $c_i(i=1,2,\cdots,n)$ 均为正常数．不失一般性，不妨设 $c_1\leqslant c_2\leqslant\cdots\leqslant c_n$．

对上述模型，[1]是在不考虑 $x_i\ (i=1,2,\cdots,n)$ 为整数的情况下给出了一种求解方法，然后按四舍五入取近似整数值．因此，从理论上说，[1] 并未解决自己所提出的问题．

本文对上述模型进行了理论探讨，给出了最优解的判别条件和求最优解的拟多项式算法程序，并把类似的程序用于解模型 II．

§ 1. 最优解的判别条件

对于模型 I 我们有如下结果：

定理 1. 模型 I 必存在一个最优解 $X=(x_1,x_2,\cdots,x_n)$ 满足如下条件：

$$c_k(x_k-1)\leqslant\min_{1\leqslant i\leqslant n}\{c_ix_i\},\quad k=1,2,\cdots,n. \tag{1}$$

反之，凡满足条件(1)的可行解一定是模型 I 的最优解．

证明．记 $f(X)\triangleq\min\limits_{1\leqslant i\leqslant n}\{c_ix_i\}$，并记使 c_ix_i 取极小的下标为 i_0．

设 $X=(x_1,x_2,\cdots,x_n)$ 是模型 I 的最优解且不满足条件 (1)，即存在 $k\in\{1,2,\cdots,n\}$ 使得

$$c_k(x_k-1)>f(X)=c_{i_0}x_{i_0}\qquad(k\neq i_0),$$

而

$$c_{i_0}(x_{i_0}+1)>c_{i_0}x_{i_0}.$$

于是以 $X'=(x_1',x_2',\cdots,x_n')$ 代替 X 必有

$$f(X')\geqslant f(X),$$

其中

$$x_j'=\begin{cases}x_j-1, & \text{当 } j=k,\\ x_j+1, & \text{当 } j=i_0,\\ x_j, & \text{当 } j\neq k、i_0.\end{cases} \tag{2}$$

既然 X 是最优解，故必 $f(X')=f(X)$，即 X' 也是最优解．对 X' 而言，若指标 k 处的分量 x_k' 仍不满足条件(1)，即

$$c_k(x_k'-1)>f(X')=c_{i_1}x_{i_1}'\qquad(k\neq i_1),$$

那末沿用上述作法，又可找到一个最优解 X'' 且 $f(X'')=f(X')=f(X)$．对 X'' 而言，不满足(1)的指标 k 处的分量比 x_k' 又减少了 1．如此下去经有限次手续，必可找到一个最优解 $X^{(1)}$，$f(X^{(1)})=f(X)$ 且指标 k 处的分量满足条件(1)．这样一来，对 $X^{(1)}$ 而言，不满足(1)的指标便减少了一个．

重复以上讨论，又可找到一个最优解 $X^{(2)}$，$f(X^{(2)})=f(X^{(1)})=f(X)$，且它的不满足条件(1)的指标比 $X^{(1)}$ 又减少了一个．这里引起注意的是，根据(2)式的叠代方法，每次都对不满足 (1) 的指标处的分量减少 1，使 c_ix_i 取极小的指标 i 处的分量增加 1 而其余

分量不变，又目标函数值 $f(X^{(i)})$ 不随 i 而变. 因此，原来满足条件(1)的分量，经一次叠代后，仍满足条件(1). 于是，继续如此下去，经有限次重复，必可找到一个满足条件(1)的最优解.

现证定理的第二部分. 设满足条件(1)的可行解 $X=(x_1, x_2, \cdots, x_n)$ 不是最优解，则一定存在一个最优解 $X'=(x'_1, x'_2, \cdots, x'_n)$ 使 $f(X')>f(X)$，即

$$c_{j_0}x_{j_0}=f(X)<f(X')\leqslant c_{j_0}x'_{j_0},$$

亦即

$$x_{j_0}<x'_{j_0}.$$

又 $\sum_i x_i=\sum_i x'_i=m$，故存在指标 $j_1\in\{1, 2, \cdots, j_0-1, j_0+1, \cdots, n\}$ 使得

$$x_{j_1}>x'_{j_1} \text{ 即 } x'_{j_1}\leqslant x_{j_1}-1.$$

于是由 X 满足条件(1)可得

$$f(X')\leqslant c_{j_1}x'_{j_1}\leqslant c_{j_1}(x_{j_1}-1)\leqslant f(X),$$

即 $f(X')\leqslant f(X)$，此与假设矛盾.

§2. 模型 I 的算法程序

下面我们建立一个求模型 I 最优解的算法程序.

模型 I 最优解算法程序:

(一) 若 $m=n$，则最优解必为 $X=(1, 1, \cdots, 1)$.

(二) 若 $m>n$，则

1° 先取初始解 $X^1=(2, 1, 1, \cdots, 1)$;

2° 设第 i 步的解为 $X^i=(x_1, x_2, \cdots, x_n)$，且

$$\sum_{j=1}^{n} x_j<m.$$

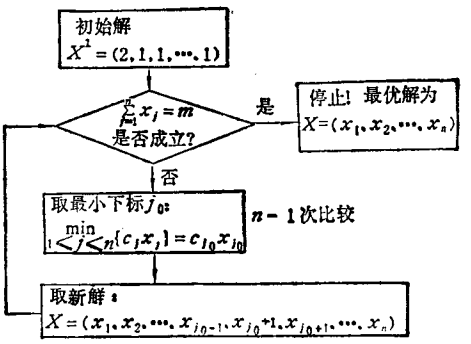
若 $\min\limits_{1\leqslant j\leqslant n}\{c_jx_j\}=c_{j_0}x_{j_0}$（如果适合此式的 j_0 不止一个，则取所有适合此式下标中最小的一个为 j_0），则第 $i+1$ 步的解取作

$$X^{i+1}=(x_1, x_2, \cdots, x_{j_0-1}, x_{j_0}+1, x_{j_0+1}, \cdots, x_n);$$

3° 若 X^{i+1} 是模型 I 的可行解，则停止；否则继续 2°，直到得出可行解便停止.

我们的算法是一个多项式算法. 事实上，我们最多只须 $m-n$ 步就可达到最优解. 除第一步外，每一步需要 $n-1$ 次比较，一次加法和最多 n 次乘法，因而不超过 $(m-n)\cdot[(n-1)+1+n]=2n(m-n)$ 次计算. 当 $m=O(n^\alpha)$（α 为常数），我们的算法是 $O(n^{\alpha+1})$ 阶的. 在一般情形，若 $m\fallingdotseq O(n^\alpha)$，则我们的算法是 $O(n(m-n))$ 阶的，因而是拟多项式的.

算法框图如下:



下面我们证明算法的正确性.

定理 2. 上述算法程序所给出的可行解必是模型 I 的最优解.

证明. 设 $X=(x_1,x_2,\cdots,x_n)$ 是由上述算法程序所给出的一个可行解. 欲证它是最优解,只须证明 X 满足定理 1 的条件(1)就可以了.

倘若 X 不满足条件(1),即存在 $k\in\{1,2,\cdots,n\}$,使

$$c_k(x_k-1)>f(X)=\min_{1\leqslant i\leqslant n}\{c_ix_i\}.$$

假定在某一步，第 k 个分量被赋值 x_k-1，而每一步的目标函数值均不大于 $f(X)$，因此按照我们的程序 2°,第 k 个分量永远也不可能赋值 x_k，亦即按照上述算法就不可能得到可行解 X，此与假设矛盾.

§3. 算　例

设某工厂生产某种产品,共由 10 个部件组成. 现有 100 台设备可供加工. 若每台设备加工第 i 个部件的日效率为 c_i (个/天),列表如下:

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
3	8	8	16	18	24	48	56	64	72

问怎样安排每台设备的加工任务,才能使该厂的日产量达到最大?

解. 该问题所指日产量显然应理解成最大配套数. 设 x_i 是加工第 i 个部件的设备台数,则数学模型可归结为

求一 $X=(x_1,x_2,\cdots,x_{10})$ 满足下列约束条件

$$\begin{cases}\sum_{i=1}^{10} x_i=100\\ x_i\geqslant 1 \text{ 且为整数},\ i=1,2,\cdots,10\end{cases}$$

使目标函数 $y=\min_{1\leqslant i\leqslant 10}\{c_ix_i\}$ 取最大值.

我们按 §2 中的算法程序编成 BAISC 程序:

```
10 DIM C%(10), X%(10)
```

```
20 FOR J=1 TO 10
30 READ C%(J), X%(J)
40 NEXT J
50 S%=0
60 FOR J=1 TO 10
70 S%=S%+X%(J)
80 NEXT J
90 IF S%=m THEN 180
100 M%=10000
110 FOR J=10 TO 1 STEP -1
120 IF M%<C%(J)*X%(J) THEN 140
130 M%=C%(J)*X%(J): B%=J
140 NEXT J
150 X%(B%)=X%(B%)+1
160 S%=S%+1
170 IF S%<m THEN 100
180 FOR J=1 TO 10
190 LPRINT X%(J);
200 NEXT J
210 DATA 3, 2, 8, 1, 8, 1, 16, 1, 18, 1, 24, 1, 48, 1, 56, 1, 64, 1, 72, 1
```

按此上机程序在 TRS-80 微机上进行计算，花时 21″，其打印结果（即最优解）为

$$(40, 15, 15, 8, 7, 5, 3, 3, 2, 2).$$

另外，我们改变设备台数 m 的值，仍按上述编制的 BASIC 程序试算．我们发现，计算时间基本上是随设备台数的增加而成比例地增加的．

§4. 附　记

我们先看一个实际例子．

某工厂给零件加工车间下达一项紧急生产任务，要在尽可能短的时间内完成 m 个零件的加工任务．已知第 i 台设备加工一个零件所需时间为 c_i（小时），现共有 n 台设备，问如何合理分派各台设备的加工任务，才能使下达的任务完成得最快？

设 x_i 为第 i 台设备加工零件的个数，则上述问题显然可归结为下列数学模型：

模型 II. 求一 $X=(x_1, x_2, \cdots, x_n)$ 满足下列约束条件：

$$\begin{cases} \sum_{j=1}^{n} x_j = m \ (m \geqslant n \text{ 且为整数}), \\ x_j \geqslant 0 \text{ 且为整数}, \ j=1, 2, \cdots, n. \end{cases}$$

使目标函数 $y=\max\limits_{1\leqslant j\leqslant n}\{c_j x_j\}$ 取最小值．其中 $c_i (i=1, 2, \cdots, n)$ 均为正常数．不失一

般性，不妨设 $c_1 \leqslant c_2 \leqslant \cdots \leqslant c_n$.

不难发现此模型与模型 I 十分对称，因此处理模型 I 的一套方法可类似地用来处理模型II. 对于模型 II，我们有

定理 3. 模型 II 必存在一个最优解 $X = (x_1, x_2, \cdots, x_n)$ 满足如下条件

$$c_k(x_k + 1) \geqslant \max_{1 \leqslant j \leqslant n} \{c_j x_j\}, \quad k = 1, 2, \cdots, n. \tag{3}$$

反之，凡满足条件(3)的可行解一定是模型 II 的最优解.

证明可仿定理 1 进行.

模型 II 最优解的算法程序如下：

1° 任取满足条件 $x_1^0 \geqslant x_2^0 \geqslant \cdots \geqslant x_n^0$ 的初始可行解 $X^0 = (x_1^0, x_2^0, \cdots, x_n^0)$;

2° 设第 i 步的解为 $X^{i-1} = (x_1, x_2, \cdots, x_n)$ 且

$$\max_{1 \leqslant j \leqslant n} \{c_j x_j\} = c_u x_u$$

(若适合此式的 u 不止一个，则取最大的一个作为 u). 如果存在 $k \in \{1, 2, \cdots, n\}$ 使

$$c_k(x_k + 1) < c_u x_u$$

(若适合此式的 k 不止一个，则取最小的一个作为 k)，则取第 $i + 1$ 步的解为

$$X^i = (x_1', x_2', \cdots x_n'),$$

其中

$$x_j' = \begin{cases} x_j, & \text{当 } j \neq k、u, \\ x_k + 1, & \text{当 } j = k, \\ x_u - 1, & \text{当 } j = u. \end{cases}$$

3° 若 X^i 满足定理 3 的条件(3)，则停止；否则继续 2°，直到找到适合条件(3)的可行解便停止.

容易证明此算法是收敛的且是一拟多项式算法.

本文初稿承郑州大学副教授林诒勋同志仔细审阅并提出宝贵修改意见，谨此致谢！

参　考　文　献

[1] 汪德营、么子皋，生产系统劳力的平衡配套问题，曲阜师院学报，第 3 期 1982 年(或见：全国最优化理论及应用学术交流资料，1982，10 武汉).