



The 0-1 Programming Solution for the Problem of Resource Allocation

Guoliang Luo & Xingang Zhao

School of Business and Administration

North China Electric Power University

Beijing 102206, China

Tel: 86-10-8079-8529 E-mail: LGL1965@126.com

E-mail: rainman319@ncepu.edu.cn

The research is financed by National Nature Fund of China (No. 70671042). (Sponsoring information)

Abstract

In this article, we first translated the problem of disperse resource allocation into the 0-1 programming problem, and put forward the special solution based on the relative difference quotient aiming at the particularity of resource allocation 0-1 programming model. And the example proved that the new algorithm was effective and fast.

Keywords: Resource allocation, 0-1 programming, Relative difference quotient, Feasible increasing direction

1. Introduction of the problem

In actual problems, we always need to allocate a sort of or many sorts of resources with certain quantity to many departments or units, and make the benefit achieve optimization. The mathematical model of this sort of problem is not complex for one sort of resource, and if suppose the gross of resource is a which should be allocated to n departments, the allocation quantum of the i 'th department is x_i , and the corresponding benefit function is $g_i(x_i)$, so the model can be denoted as

$$\max z = \sum_{i=1}^n g_i(x_i)$$

$$s.t. \quad \sum_{i=1}^n x_i \leq a$$

$$x_i \geq 0, i = 1, 2, \dots, n$$

In fact, because of differences of original base, geographic position, market orientation and using attention for various departments, even if we offer resources with same quantity to various departments, the benefits of various departments are not same, i.e. the benefit functions of various departments are different, and their function relationships are not certainly resolutions, and these characters of non-resolution and dispersion of benefit function make the resolution computation become more difficult. General operational research teaching materials all adopt the method of dynamic programming to solve the problem, and comparing with general method of exhaustion, the computation process could be simplified, but the time process is long and the computation quantity is large, especially for one of n , a is big at least. On the other hand, once the data of benefit sheet change, the former and the latter allocation projects are different, which couldn't be adjusted in time. Aiming at these problems, how to design more simple and effective algorithm?

2. 0-1 programming model for the problem of resource allocation

To the problem of resource allocation, we can establish possible allocation project in advance, and according to relative investigation and statistics, the benefit value c_{ij} brought when resources are allocated to the j 'th department d_j , and they are seen in Table 1.

Introduce 0-1 variables,

When $x_{ij}=1$, allocate resources to the j 'th department according to the i 'th mode.

When $x_{ij}=0$, don't allocate resources to the j 'th department according to the i 'th mode.

Because every department only can be allocated according to one sort of project, so we can establish the following mathematical model of 0-1 programming.

$$\begin{aligned} \max z &= \sum_{j=1}^n \sum_{i=0}^m c_{ij} x_{ij} \\ s.t. &\begin{cases} \sum_{i=0}^m x_{ij} = 1, & j = 1, 2, \dots, n \\ \sum_{j=1}^n \sum_{i=0}^m d_i x_{ij} \leq a \\ x_{ij} = 0 \text{ or } 1, & j = 1, 2, \dots, n, i = 0, 1, 2, \dots, m \end{cases} \end{aligned}$$

It is familiar 0-1 programming model. When the resource quantity is not large, and the allocation projects and the departments participating in allocation are few, we can apply the programming solution function in Excel software to solve the problem fast and conveniently.

3. Special solution for the resource allocation 0-1 programming model

When the resource quantity is large, and the allocation projects and the departments participating in allocation are more, the programming solution function in Excel software is inefficient to solve the problem. To the 0-1 programming problem, we usually use the filtration concealed enumeration method to solve it, and the method is to find a feasible solution first, and then obtain a filtration condition by using it, and don't test the feasibility for the point which doesn't fulfill the filtration condition. Though the operation quantity of the filtration concealed enumeration method is less than the enumeration method, but in actual application, it has two deficiencies.

(1) This method should first seek one feasible solution, but it doesn't give the method to seek the first feasible solution. When the limitation equations are more, and we could not distinguish which variable combination is feasible solution, we must solve the problem by the tentative method. This sort of method is blind, and sometimes we must try several variable combinations to find the first feasible solution.

(2) When the objective function value of the first feasible solution is different with the optimal value, the filtration condition confirmed by the objective function value is bad, and even if we continually improve the filtration conditions in the computation process, the computation quantity is still large.

In Huang, Huiqing's article (Huang, 2002, p.1052-1053), the author put forward a sort of new method to solve 0-1 integer programming. The new method first seeks corresponding objective function value Z to various variable combinations $(x_1, x_2, \dots, x_n)^T$, then sorts all objective function values by size, and tests the feasibilities of various solutions and confirms the optimal solution. If the optimal solution exists in the 0-1 programming, we will find the optimal solution through 2^n times at best. For the 0-1 programming model of resource allocation, because there are $m \times n$ 0-1 variables and the possible combination mode quantity of various variables is 2^{mn} , so the method still has too large computation quantity. For the large scale problem, it is not feasible.

For the 0-1 programming model of resource allocation

$$\begin{aligned} \max z &= \sum_{j=1}^n \sum_{i=0}^m c_{ij} x_{ij} \\ s.t. &\begin{cases} \sum_{i=0}^m x_{0j} = 1, & j = 1, 2, \dots, n \\ \sum_{j=1}^n \sum_{i=0}^m d_i x_{ij} \leq a \\ x_{ij} = 0 \text{ or } 1, & j = 1, 2, \dots, n, i = 0, 1, 2, \dots, m \end{cases} \end{aligned}$$

, if we don't consider the 0-1 limitation of the problem, we add the slack variable into the inequality limitation, and the coefficient matrix of its limitation condition is

$$A = \begin{pmatrix} 1 & 1 & \cdots & 1 & & & & & & 0 \\ & & & & 1 & 1 & \cdots & 1 & & & 0 \\ & & & & & & & \cdots & & & \cdots \\ & & & & & & & & 1 & 1 & \cdots & 1 & 0 \\ d_1 & d_2 & \cdots & d_m & d_1 & d_2 & \cdots & d_m & \cdots & d_1 & d_2 & \cdots & d_m & 1 \end{pmatrix}$$

, and the item of constant is $(1 \ 1 \ \cdots \ 1 \ a)^T$.

The problem must have the optimal solution, and to any $j \in \{1, 2, \dots, n\}$, the factor with $\{x_{ij} \mid i = 0, 1, 2, \dots, m\}$ in the base variable of feasible base exists and its quantity is only one. Aiming at this special problem, we put forward following special solution.

(1) Confirmation of initial project (feasible solution)

For the confirmation of initial project (feasible solution), we adopt the maximum proportion principle, i.e. the position which the relative efficiency of asset is maximum is first selected. The concrete computation block diagram is seen in Figure 1.

(2) Optimal adjustment

In this article, the allocation project of resource we consider is in the situation with equal step length, and suppose that any one department obtains more resources, it will produce more benefits. The 0-1 programming model of the problem is

$$\begin{aligned} \max z &= \sum_{j=1}^n \sum_{i=1}^K c_{ij} x_{ij} \\ \text{s.t.} &\begin{cases} \sum_{i=1}^K x_{ij} = 1, & j = 1, 2, \dots, n \\ \sum_{j=1}^n \sum_{i=1}^K d_i x_{ij} = a \\ x_{ij} = 0 \text{ or } 1, & j = 1, 2, \dots, n, i = 1, 2, \dots, K \end{cases} \end{aligned}$$

, denote the feasible field of the problem is Ω .

To conveniently describe the problem, we first introduce several concepts.

Definition 1. Denote $\xi(i, j, k) = \frac{c_{i(j+k)} - c_{ij}}{k}$ is the k 'th step relative difference quotient of the variable x_{ij} . When $k > 0$, it is called as the forward k 'th step relative difference quotient, and when $k < 0$, it is called as the backward $|k|$ step relative difference quotient.

Suppose

$$\begin{aligned} x &= (x_{10}, x_{11}, \dots, x_{1m}, x_{20}, x_{21}, \dots, x_{2m}, \dots, x_{n0}, x_{n1}, \dots, x_{nm})^T \in \Omega \\ y &= (y_{10}, y_{11}, \dots, y_{1m}, y_{20}, y_{21}, \dots, y_{2m}, \dots, y_{n0}, y_{n1}, \dots, y_{nm})^T \in \Omega \\ \text{where, } x_{k,i} &= \begin{cases} 1, & k_i = i_k^* \\ 0, & k_i \neq i_k^* \end{cases}, x_{k,i,j} = \begin{cases} 1, & k_i = i_k^* \\ 0, & k_i \neq i_k^* \end{cases}, y_{k,i} = \begin{cases} 1, & k_i = j_k^* \\ 0, & k_i \neq j_k^* \end{cases} \\ \text{so } \sum_{k=1}^n i_k^* h &= a, \sum_{k=1}^n j_k^* h = a, \text{ and } \sum_{k=1}^n (i_k^* - j_k^*) = 0. \end{aligned}$$

Definition 2. For the integer vector $m = (m_1, m_2, \dots, m_n)^T$, if $\sum_{j=1}^n m_j = 0$, so the integer vector is feasible.

Definition 3. For $x = (x_{10}, x_{11}, \dots, x_{1m}, x_{20}, x_{21}, \dots, x_{2m}, \dots, x_{n0}, x_{n1}, \dots, x_{nm})^T \in \Omega$ and the integer vector $m = (m_1, m_2, \dots, m_n)^T$, where $x_{k,i} = \begin{cases} 1, & k_i = i_k^* \\ 0, & k_i \neq i_k^* \end{cases}$, if $\sum_{k=1}^n \xi(k, i_k^*, m_k) \cdot m_k > 0$, so m is the increasing direction of x .

So it is obvious that the theorem.

Theorem. For the feasible solution of the resource allocation 0-1 programming model

$$x = (x_{10}, x_{11}, \dots, x_{1m}, x_{20}, x_{21}, \dots, x_{2m}, \dots, x_{n0}, x_{n1}, \dots, x_{nm})^T \in \Omega,$$

if the feasible increasing direction of x doesn't exist, so x is the optimal solution of the problem.

(3) Computation block diagram of special solution of 0-1 programming model for the problem of resource allocation

The computation block diagram is seen in Figure 2.

4. Concrete example

Suppose the benefit values of different resource allocation quantities in different departments are seen in Table 2, and the total quantity of resource is 14.

From the confirmation method of the initial project, we can obtain the initial project.

$$x_{21} = 1, x_{32} = 1, x_{53} = 1, x_{24} = 1, x_{25} = 1, x_{46} = 1, x_{37} = 1$$

The relative difference quotients of various step lengths of the feasible solution are seen in Table 3.

The feasible increasing direction is $(0, 0, 0, 2, 0, -2, 0)$, so the new feasible solution is

$$x_{21} = 1, x_{32} = 1, x_{53} = 1, x_{44} = 1, x_{25} = 1, x_{26} = 1, x_{37} = 1$$

, and the relative difference quotients of various step lengths of this feasible solution are seen in Table 4.

And the feasible increasing direction is $(0, 0, 0, 2, 0, -1, 0)$, so the new feasible solution is

$$x_{21} = 1, x_{32} = 1, x_{53} = 1, x_{44} = 1, x_{25} = 1, x_{16} = 1, x_{47} = 1$$

, and the relative difference quotients of various step lengths of this feasible solution are seen in Table 5.

Because the increasing direction doesn't exist in this feasible solution, so the optimal solution is

$$x_{21} = 1, x_{32} = 1, x_{53} = 1, x_{44} = 1, x_{25} = 1, x_{16} = 1, x_{47} = 1, \max z = 88.$$

5. Conclusions

In this article, we translate the problem of disperse resource allocation into the 0-1 programming problem. Aiming at the particularity of the 0-1 programming model of resource allocation, we put forward the special solution, i.e. first confirm the initial project by the principle first selecting the position which relative efficiency of asset is maximum, judge where more excellent feasible project exist by the forward relative difference quotient and the backward relative difference quotient. We also give the computation block diagram to confirm the initial feasible solution and the whole computation process, and the example and numerical experiment indicate the algorithm is fast and effective.

References

- Editing Team of Operational Research Teaching Material. (2005). *Operational Research (3rd Edition)*. Beijing: Tsinghua University Press. p.213-216.
- Huang, Huiqing. (2002). A New Algorithm for Zero-One Type Integer Programming. *Mathematics in Practice and Theory*. No.32(6). p.1052-1053.
- Huang, Zhenglong. (2007). Conversion of Optimized Models of Resources Allocation and Its Algorithms. *Journal of Central South University of Forestry & Technology*. No.27(2). p.122-125.

Table 1. Benefit values produced by different resource allocation quantities in different departments

Quantity of resource	Benefit values produced in various departments			
	1	2	...	n
0	c_{01}	c_{02}	...	c_{0n}
1	c_{11}	c_{12}	...	c_{1n}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
m	c_{m1}	c_{m2}	...	c_{mn}

Table 2. Benefit values produced by different resource allocation quantities in different departments (example)

Quantity of resource	No. of department						
	1	2	3	4	5	6	7
	Available profits						
0	0	0	0	0	0	0	0
1	9	1	1	9	6	4	6
2	11	10	4	15	9	8	15
3	13	14	12	23	10	12	20
4	13	17	21	23	11	16	24
5	13	20	23	24	12	25	25
6	19	23	24	32	16	25	31
7	20	26	32	36	24	27	38
8	28	31	34	44	28	27	41
9	29	32	40	51	36	33	41

Table 3. Relative difference quotients of different steps for the initial project feasible solution

step length	forward relative difference quotient						
1	2	4	2	6	3	4	5
2	2	3.5	1.5	7	2	6.5	4.5
3	1.333333	3.333333	3.666667	4.666667	1.666667	4.333333	3.333333
4	1	3.25	3.25	3.75	1.5	3.75	4
5	2	3.2	3.8	4.6	2	3	4.6
6	1.833333	3.5	0	4.5	3	3.5	4.333333
7	2.714286	3.142857	0	5	3.142857	0	3.714286
8	2.5	0	0	5.25	3.75	0	0
step length	forward relative difference quotient						
1	9	9	9	9	6	4	9
2	1000	5	8.5	1000	1000	4	7.5
3	1000	1000	6.666667	1000	1000	4	1000
4	1000	1000	5.25	1000	1000	1000	1000

Table 4. Relative difference quotients of different steps for new feasible solution

step length	forward relative difference quotient						
1	2	4	2	0	3	4	5
2	2	3.5	1.5	0.5	2	4	4.5
3	1.333333	3.333333	3.666667	3	1.666667	4	3.333333
4	1	3.25	3.25	3.25	1.5	5.25	4
5	2	3.2	3.8	4.2	2	4.2	4.6
6	1.833333	3.5	0	4.666667	3	3.833333	4.333333
7	2.714286	3.142857	0	0	3.142857	3.285714	3.714286
8	2.5	0	0	0	3.75	3.625	0
step length	forward relative difference quotient						
1	9	9	9	8	6	4	9
2	1000	5	8.5	7	1000	1000	7.5
3	1000	1000	6.666667	7.666667	1000	1000	1000
4	1000	1000	5.25	1000	1000	1000	1000

Table 5. Relative difference quotients of different steps for new feasible solution

step length	forward relative difference quotient						
1	2	4	2	0	3	4	4
2	2	3.5	1.5	0.5	2	4	2.5
3	1.333333	3.333333	3.666667	3	1.666667	4	3.666667
4	1	3.25	3.25	3.25	1.5	4	4.5
5	2	3.2	3.8	4.2	2	5	4.2
6	1.833333	3.5	0	4.666667	3	4.166667	3.5
7	2.714286	3.142857	0	0	3.142857	3.857143	0
8	2.5	0	0	0	3.75	3.375	0
9	0	0	0	0	0	3.666667	0
step length	forward relative difference quotient						
1	9	9	9	8	6	1000	5
2	1000	5	8.5	7	1000	1000	7
3	1000	1000	6.666667	7.666667	1000	1000	6.666667
4	1000	1000	5.25	1000	1000	1000	1000

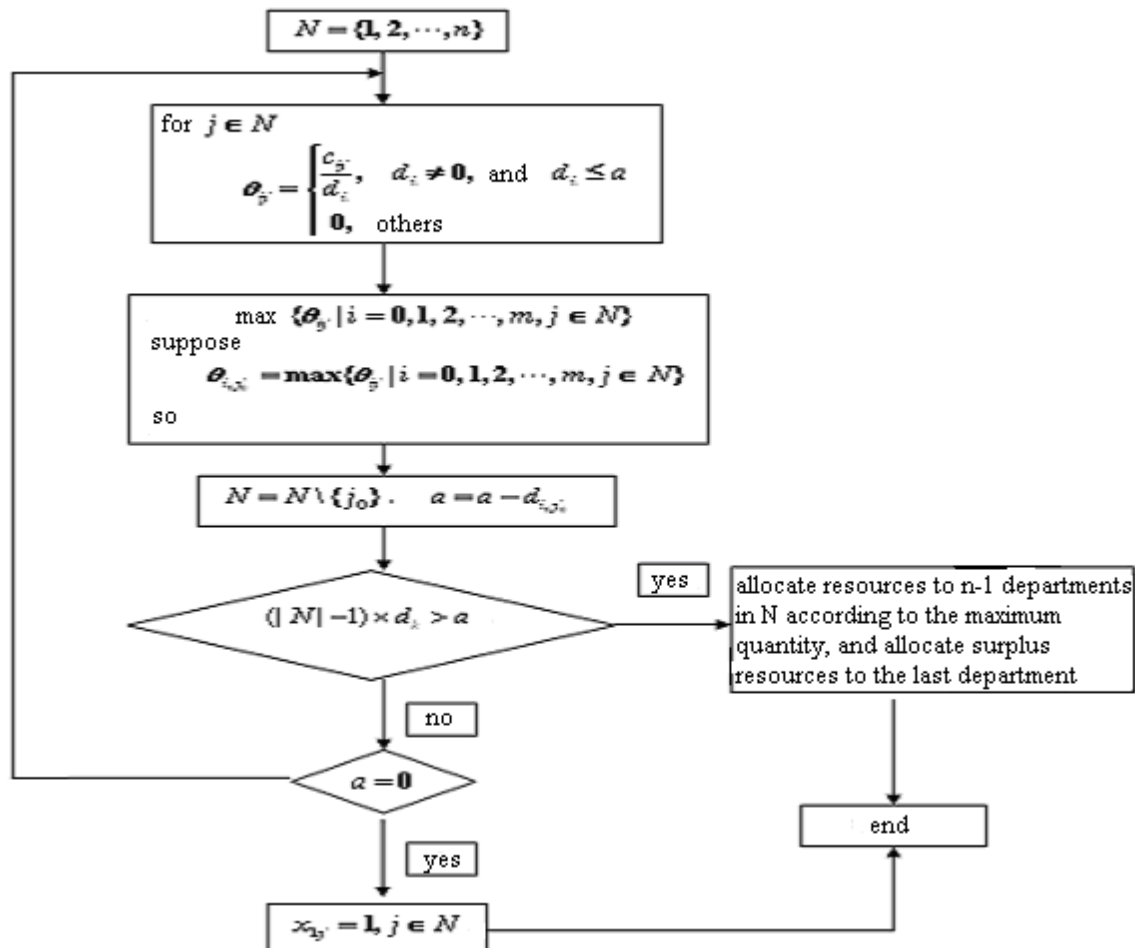


Figure 1. Computation Block Diagram of Initial Project

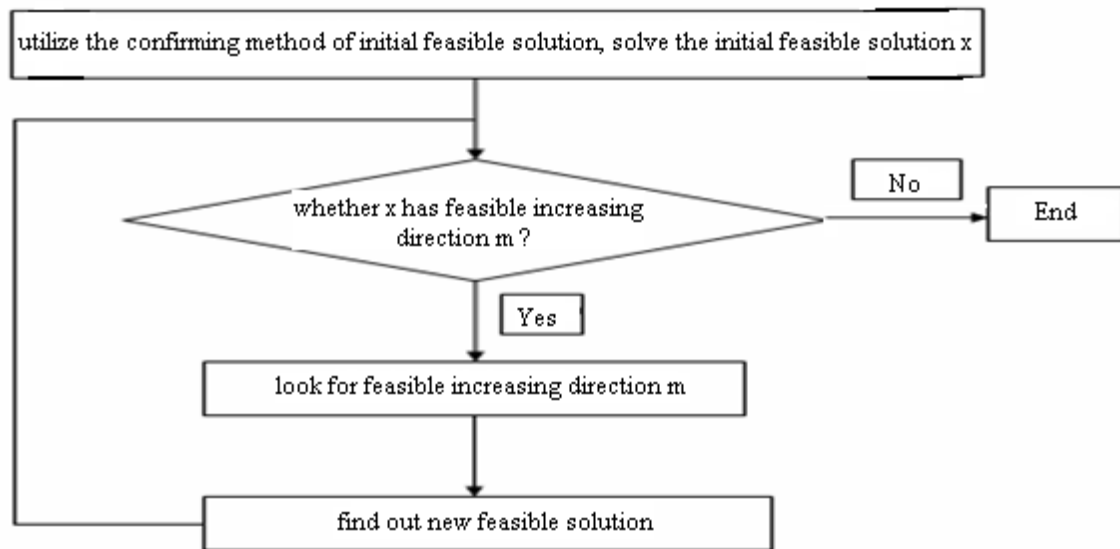


Figure 2. Computation Block Diagram of Special Solution