Application network programming method – knapsack problem and it's modification

Irina Burkova[1], *Boris* Titarenko[2], and *Yulia* Zheglova[2*]

[1]Institute of Control Sciences of Russian Academy of Science, Leninskiy14, Moscow, Russia
[2]Moscow State University of Civil Engineering, Yaroslavskoye shosse, 26, 129337, Moscow, Russia

Abstract. New method of multi-extremal optimization has been developed - a network programming method, which is based on the possibility to represent a complex function as a superposition of simpler functions. The paper deals with network (dichotomous) programming method and its application for solving multi-extremal and discrete optimization problems in Project Management. The main goal of the paper is to develop a new approach to solving knapsack problem and to solve a number of optimization problems of project building management. The optimal structure of the dichotomous representation is determined. On the basis dichotomous programming method new algorithm to solve a optimization problem is proposed

1 Introduction

An essential part of the models and mechanisms of building project management are optimization problems, which are complex and multi-extremal. Therefore, the problem of developing effective methods for solving applied problems of multi-extremal optimization in the field of project management is relevant.

The main goal of the work is to develop a new approach to solving optimization problem called knapsack problem and to solve a number of optimization problems of project management based on the network programming method.

On the basis of the network programming method and its special case - the dichotomous programming method, new algorithms for solving a number of optimization problems in project management are proposed.

2 Literature review

A new method for solving problems of multi-extremal optimization has been developed - a network programming method, which is based on the possibility of representing a complex

[*] Corresponding author: uliagermanovna@yandex.ru

function as a superposition of simpler functions (such a representation is called a network representation).

Based on the network programming method and its special case - the dichotomous programming method, new algorithms for solving a number of optimization problems in project management are proposed.

Last years network programming theory has received wide recognition and has found a diverse range of applications [5,6,7,8]

The network programming method is applied to solve the following project management tasks:

• Scheduling problem taking into account the time of resource movement (traveling salesman problem) [2,5, 8],

• The problem of maximizing the amount of work performed by the project (the problem of maximum flow) [10,14];

• The problem of uniform use of the resource (the problem of stones) [14,15].

Solving knapsack problem the optimal structure of the network representation was done [1,3,6,9]. It was shown that the optimal structure is a combination of the maximally symmetric and Bellman structures [7,8 ,11,12,13].

Many problems of construction project management (formation of target programs, selection of a portfolio of projects, placement of service points, etc.) can be reduced to the knapsack problem and its modifications [16,17].

3 Problem formulation

Consider the formulation of the "one-dimensional knapsack" problem. There are n items, each item is characterized by weight α_i and value c_i (it is assumed that α_i, c_i are positive integers). There is also a knapsack (backpack) with a capacity R. It is required to load the knapsack with items so that the total value of the items placed in the knapsack is maximum, provided that the total weight does not exceed R. We denote $x_i = 1$ if the i-th item is placed in the knapsack, $x_i = 0$ otherwise. The mathematical formulation of the problem is:

$$f(x) = \sum_j c_j x_j \to \max ; \tag{1}$$

$$\varphi(x) = \sum_j a_j x_j \le R ; \tag{2}$$

$$x_j \in \{0;1\} ; \; j = \overline{1,n} \tag{3}$$

The structure of the network representation of the problem is a tree, so the method of network or dichotomous programming gives the optimal solution.

Example 1.

$$f(x) = 5x_1 + 7x_2 + 6x_3 + 3x_4 \to \max ;$$
$$\varphi(x) = 2x_1 + 3x_2 + 5x_3 + 7x_4 \le 9 .$$

Let's take the structure of the network view (Figure 1).

1 step. We solve a simple problem at the nodey_1 with the first two objects. To do this, we build Table 1.

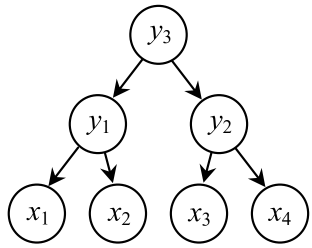

Fig. 1. Network view structure.

Table 1. Values and weights of options.

1	7, 3	12, 5
0	0, 0	5, 2
x_2 / x_1	0	1

Two numbers are written in the cells at the intersection of the columns corresponding to the values $x_1 = 0$ or 1 and the rows corresponding to the values $x_2 = 0$ or 1. The first is equal to the total value of the corresponding option, and the second is the total weight. We write the results in a table of options, in which weights and values are arranged in ascending order.

Table 2.Table of variants

N	1	2	3	4
Value(y_1)	0	5	7	12
Weight (z_1)	0	2	3	5

2 step. We solve a simple problem at the node y_2 with the last two objects (Table 3):

Table 3. Solution at node y_2.

1	3, 7	–
0	0, 0	6, 5
x_4 / x_3	0	1

The results are summarized in Table 4.

Table 4. Calculation results

N	1	-
Value(y_2)	0	6
Weight (z_2)	0	5

Option (3, 7) is excluded(it is dominated by option (6, 5)). It has a lower value, but has more weight)..

3 step. We solve the problem at the nodey_3. The solution is in Table 5.

Table 5. Solution at node y_3.

2	6, 5	11, 7	13, 8	–
1	0, 0	5, 2	7, 3	12, 5
y_2 / y_1	1	2	3	4

The results are summarized in Table 6.

Table 6. Calculation results

N	1	2	3	4
f	5	7	12	13
φ	2	3	5	8

To determine the optimal solution, we apply the "reverse move" method. Pare $(f, \varphi) = (13, 8)$ Table 6 corresponds to the cell (3, 2) of the Table 5, i.e. the third option from the Table 2 and the second option from Table 4.

The second version of Table 4 corresponds to the cell (1, 0) of the Table 3, i.e. $x_3 = 1$, $x_4 = 0$. In turn, the third variant of Table 2 corresponds to the cell (0, 1) of the Table 1, i.e. $x_1 = 0$, $x_2 = 1$. Finally, we obtain the optimal solution:

$$x_1 = 0,\ x_2 = 1,\ x_3 = 1,\ x_4 = 0,\ f_{\max} = 13.$$

4 Modification of problem

Consider generalizations of the problem that preserve the structure of the network representation in the form of a tree.

Let the set of all objects be divided into m disjoint subsets Q_j, $j = 1, 2, \ldots, m$. In each subset, one and only one item is allowed. For a formal statement of the problem, we introduce variables $\{x_{ij}\}$, $i = \overline{1, n_j}$, $j = \overline{1, m}$ taking the values 0 or 1. We take $x_{ij} = 1$ if the object $i \in Q_j$ is taken to the knapsack and $x_{ij} = 0$ otherwise. The problem is to find $\{x_{ij}\}$ that maximize

$$\sum_{i,j} c_{ij} x_{ij} \tag{4}$$

under restrictions

$$\sum_{i,j} a_{ij} x_{ij} \le R, \tag{5}$$

$$\sum_{i \in Q_j} x_{ij} = 1,\ j = \overline{1, m}, \tag{6}$$

where c_{ij}, a_{ij} are the value and weight of the item $i \in Q_j$, respectively.

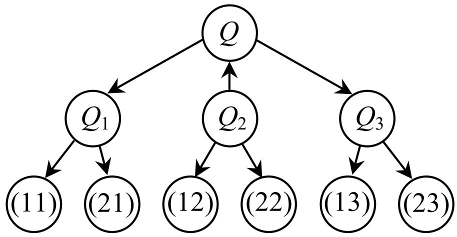

Fig. 2. Network view structure.

Another interpretation of this problem is that there are m items, but each item has n_j modifications that differ in value and weight. You must select one modification of each item. The structure of the network view is still a tree.

Example 2. There are three groups of two subjects each, details of which are given below.

Table 7. Values and weights of options

Groups	1		2		3	
Items	(11)	(21)	(12)	(22)	(13)	(23)
c_{ij}	12	9	4	6	0	7
a_{ij}	6	5	2	4	3	2

Let's take $R = 10$.

1 step. Consider groups 1 and 2. The solution is given in Table 8.

Table 8. The solution of the problem

6, 4	15, 9	18, 10
4, 2	13, 7	16, 8
2 / 1	9, 5	12, 6

The result is summarized in Table 9.

Table 9. Result

Option	1	2	3
y_1	13	16	18
z_1	7	8	10

Option (15, 9) is excluded because it is dominated by option (16, 8).

2 step. Considering the generalized group (1, 2) and group 3 we have the solution (see Table 10).

Table 10. Solution of problem

9, 3	22, 10	–	–
7, 2	20, 9	23, 10	–
3 / (1, 2)	13, 7	16, 8	18, 10

Cell (23, 10) corresponds to the optimal solution, i.e. $x_{31} = 1$, $x_{32} = 0$, $(y_1, z_1) = (16; 8)$. From Table 9 we see that (16, 8) is the second option. From Table 8 we get: $x_{11} = 0$, $x_{21} = 1$, $x_{12} = 1$, $x_{22} = 0$.

Fina Solution:

$$x_{11} = 0,\ x_{21} = 1,\ x_{12} = 1,\ x_{22} = 0,\ x_{13} = 1,\ x_{23} = 0,\ C_{\max} = 23.$$

Let us consider generalization of the problem, when the structure of the network representation has the form of a tree. Let restrictions be imposed on the total weight of objects of the j-th group, namely

$$\sum_{i \in Q_j} a_{ij} x_{ij} \le B_j, \quad j = \overline{1, m}. \tag{7}$$

We have problem to maximize (4) under constraints (5) and (7). Our assumption is $\sum_j B_j > R$; otherwise, the problem splits into m unrelated knapsack problems.

The structure of the network representation also has the form of a tree (Figure 2).

5 Algorithm

Description of the algorithm

1 step. We solve m knapsack problems :

$$\sum_{i \in Q_j} c_{ij} x_{ij} \to \max \tag{8}$$

$$\sum_{i \in Q_j} a_{ij} x_{ij} \le B_j \tag{9}$$

2 step. We solve the problem of choosing one option from each group.

$\sum_{i,j} c_{ij} x_{ij}$ is maximized under constraints (5), (6).

6 Discussion

The network programming method can be applied to solve the following construction project management problems:

• Scheduling problem taking into account the time of resource movement (traveling salesman problem);

• The problem of maximizing the amount of work performed by the project (the problem of the maximum flow);

• The problem of uniform use of the resource (the problem of stones);

For the knapsack problem, the optimal structure of the dichotomous representation is determined by the criterion of minimizing the maximum total number of cells in the matrices. The optimal structure is a combination of the maximally symmetric and Bellman structures.

Many tasks of construction project management (formation of target programs, selection of a portfolio of projects, placement of service points, etc.) are reduced to the knapsack problem and its modifications.

References

1. G. Aggarwal and J. Hartline, Knapsack auctions. In SODA, 1083–1092, 2006.
2. S. Blackburn, International Journal of Project Management **20**, 3 (2002)
3. K. Bretthauer, B. Shetty and S. Syam, A branch and bound algorithm for integer quadratic knapsack problems, Manuscript, Texas A&M University (1993), to appear in ORSA J. Computing.
4. P. Brucker, An algorithm for quadratic Knapsack problems, Oper. Res. Lett. **3**, 163-166 (1984).
5. P. Brucker, Scheduling Algorithms (Springer-Verlag, 2001)
6. V. Burkov, I. Burkova, Dichotomous programming method in optimization problems (Central Economic and Mathematical Institute of the Russian Academy of Sciences, Moscow, Russia, 2003)
7. I. Burkova, B. Titarenko, A. Hasnaoui, R. Titarenko, Serbian Journal of Management **16**, 1 (2021)
8. V. Burkov, I. Burkova, Dichotomous programming method. Theory of active systems (2003)
9. O. Ibarra and C.E. Kim, Fast approximation algorithms for the Knapsack and sum of subset problems, J. Assoc. Comput. Mach. **22**, 463-468 (1975).
10. W. Gutjahr, OR Spectrum **33** (2011)
11. E. Lawler, Fast approximation algorithms for Knapsack problems, Math. Oper. Res. **4**, 339-356 (1979).
12. S. Martello and P. Toth, Knapsack Problems: Algorithms and Computer Implementations, Wiley, New York, 1990.
13. T. Morin and R. Marsten, An algorithm for nonlinear Knapsack problems, Management Sci. **22**, 1147-1158 (1976).
14. G. Marcondes, R. Leme, M. Silva, The Engineering Economist **62**, 1 (2017)
15. M. Sharifghazvini, Economic computation and Economic Cybernetics Studies and Research **52**, 1 (2018)
16. M. Tavano, K. Khalili, A.-R. Abtahi, Annals of Operations Research **20** (2013)
17. B. Titarenko, A. Hasnaoui, R. Titarenko, L. Buzuk, *Robust data analysis in innovation project portfolio management,* MATEC Web of Conferences 170, SPbWOSCE-2017 (2018)