

0-1整数规划在科技项目管理中的应用

王洪国^{1,2} 陈火旺³ 马绍汉⁴

(山东省科技厅 济南250011)¹ (山东中创软件工程有限公司 济南250014)²
(国防科技大学计算机学院 长沙410073)³ (山东大学计算机学院 济南250100)⁴

Application of 0-1 Integral Linear Programming in Science and Technology Project Management

WANG Hong-Guo CHEN Huo-Wang MA Shao-Han

(Shandong Provincial Science and Technology Bureau)¹ (National Defense University of Science and Technology)³
(Computer Science College of Shandong University)⁴

Abstract This paper founds a 0-1 integral programming model for science and technology project management problem by applying mathematical programming method, researching the object function and constraint conditions, and gives a responsible polynomial approximation algorithm. The software package for decision support is developed, it can available solve the realistic problem of planning project.

Keywords Knapsack problem, Planning project, 0-1 integral programming

1 前言

电子政务建设中涉及到宏观规划、财政预算、项目立项、政府采购等诸多智能决策支持问题,这些决策问题都带有若干制约条件,比背包问题要复杂得多,但其最终目的都是要寻求一种好的方案,以较少的投资产生最大的效益。本文以政府决策中的科技计划项目立项决策为例,探讨这类问题的一种可行的实用解决方案。

2 计划项目立项过程中存在的问题

通常情况下,政府5年或年度用于科技计划项目的经费数额是一定的,但可供选择的项目会有许多,每个项目都有其立项的意义和理由。由于每个项目的投入产出比、产生的效益绝对值、产业关联度及其对产业的影响千差万别,如何选择一批项目,使有限的资金发挥最大的效益是政府决策部门十分关注的重要问题。现行的方法是人工凭经验来判定。如果要立项的项目数量少、技术领域单一,人工凭经验判定的方法是能够做到的。如果要立项的项目数量多,且只有资金约束,可以利用0-1背包问题的实用快速算法得到其近似解。但若这些项目数量多,且涉及多个技术领域,涉及到经济社会生活的方方面面,存在许多约束条件,依靠人工凭经验判定和背包问题的算法不可能选择出一组最优的项目。因此,迫切需要建立一套科学有效、简便易行的计划项目立项解决方案,确保政府行政科学、合理和高效。

3 计划项目立项的数学模型

计划项目立项是一个典型的整数规划问题,设 $1, 2, \dots, i, \dots, n$ 表示 n 个项目。

3.1 决策变量

设 x_i 表示第 i 个项目的决策变量。由于项目具有不可分性, x_i 的取值只能是0或1:

$$x_i = \begin{cases} 1 & \text{表示第 } i \text{ 个项目要立项} \\ 0 & \text{表示第 } i \text{ 个项目不立项} \end{cases} \quad i=1, 2, \dots, n$$

只要确定了 x_i 的值,就能够得到一个最优的计划项目立项方案。用 $X=\{x_i | i=1, 2, \dots, n\}$ 表示解集。

3.2 约束条件

(1)资金约束 项目的投资主体是多元化的,往往涉及到多个部门或个人,多个投资者以不同的形式,如股份制、合作经营等形式共同投资。政府的投资应该也只能是项目总投资的一部分,而且只占相当小的比重,是引导性的资金,其作用和意义在于政策的激励、引导和促进产业结构的调整,通过政府的支持加强投资者的信心和决心,促使技术水平高的项目尽快实现产业化。设政府用于计划项目的经费总额度为 T ,用 a_j 表示第 j 个项目投产所需的资金,用 b_j 表示申请政府资助的资金,则有约束条件:

$$\sum_{j=1}^n b_j x_j \leq T \quad (1)$$

(2)项目总量约束 由于资金有限,不可能所有的项目都能得到政府的资助。因此,项目间就存在互相排斥的关系,一个项目的入选就意味着其它项目入选的可能性减小。有些情况下,决策部门会提出一个立项项目总量控制指标。设 k 表示项目立项的上限, k 为正整数,则有约束条件:

$$\sum_{j=1}^n x_j \leq k \quad (2)$$

(3)项目组的上下界约束 为平衡各方面的关系,计划项目立项过程中往往会对每个区域或技术领域立项项目数制定一个上下界限。不妨设 n 个项目可以分成 m 个不同的项目组,每个项目组的上下限分别用 k_q, k'_q 表示, $X_{1,q} \subset X$ 表示含有上下界约束的第 q 个项目组的集合, $q=1, 2, \dots, m, m < n$,则有约束条件:

$$k'_q \leq \sum_{x_i \in X_{1,q}} x_i \leq k_q, q=1, 2, \dots, m, m < n \quad (3)$$

显然,若 $1 \leq j \leq m, 1 \leq k \leq m, j \neq k$,则 $X_{1,j} \cap X_{1,k} = \emptyset$

(4)项目间的互相依赖约束 由于一些项目能够组成一个相互关联的产业群,分别承担一个整体的不同部分。这些项目要么都上,要么都下。当然,这些项目可以合并为一个项目来处理,但为便于处理项目间的交叉关系,我们还是把这些项

目看作一个项目组,用 $X_{2,q} \subset X$ 表示含有互相依赖约束的第 q 个项目组的集合, $q=1,2,\dots,m, m < n$, 则有约束条件:

$$x_i = x_j \quad \text{若 } x_i, x_j \in X_{2,q} \quad q=1,2,\dots,m \quad m < n \quad (4)$$

同样,若 $1 \leq j \leq m, 1 \leq k \leq m, j \neq k$, 则 $X_{2,j} \cap X_{2,k} = \emptyset$

(5)项目间的互相排斥约束 有时一些项目的技术水平、投资和产出规模等基本相同,为避免重复上雷同的项目,只能在这些项目中选择一个。如果我们用人工的方式从这些项目选择一个项目,去除其他项目,不便于作为一个整体来处理这批项目。因为互相排斥的项目可能分属于不同的具有互相依赖关系的项目组中。因此,我们还是把这些项目看作一个项目组,用 $X_{3,q} \subset X$ 表示含有互相排斥约束的第 q 个项目组的集合, $q=1,2,\dots,m, m < n$, 则有约束条件:

$$\sum_{x_i \in X_{3,q}} x_i = 1 \quad q=1,2,\dots,m \quad m < n \quad (5)$$

同样,若 $1 \leq j \leq m, 1 \leq k \leq m, j \neq k$, 则 $X_{3,j} \cap X_{3,k} = \emptyset$

(6)项目的指定约束 由于某种特殊的原因,有些项目可能被直接指定为必上项目,必须立项给予支持。而有些项目尽管项目本身不错,但由于项目承担人的信誉或能力等方面的原因,可能被直接指定为不立项支持。我们用 $X_{4,1} \subset X$ 表示指定必上项目组,用 $X_{4,0} \subset X$ 表示指定不上项目组,则有约束条件:

$$x_i = 1 \quad x_i \in X_{4,1}; x_i = 0 \quad x_i \in X_{4,0} \quad (6)$$

3.3 目标函数

(1)效益最大 由于反映项目效益的指标有很多,不同投资主体关注的重点也不尽相同。政府作为投资者,不但要分析项目的投资收益率、净现值、净年值、净年值率、内部收益率等项目的经济效益和赢利水平,而且要看项目投产后能实现多大的产值、利税,解决多大的社会就业和带动多大的产业群等社会总体指标。因此,在计划项目立项中,考核一个项目的综合效益需要一套十分复杂的评价体系,在此不做具体的研究。我们仅以 c_i 表示第 i 个项目立项后单位资金(万元或千元)所实现的效益, c_i 代表的具体内容以后再做专门的研究。因此,若求政府资金产生的效益最大,则有目标函数:

$$\max z = \sum_{i=1}^n c_i b_i x_i \quad i=1,2,\dots,n \quad (7)$$

若求项目投入的总资金产生的效益最大,有目标函数:

$$\max z = \sum_{i=1}^n c_i a_i x_i \quad i=1,2,\dots,n \quad (8)$$

表示一组项目投产后实现的效益最大。

(2)带动资金最大 由(1)式可以看出,政府投资的引导作用越大越好,因此存在目标函数:

$$\max z = \sum_{i=1}^n (a_i - b_i) x_i \quad i=1,2,\dots,n \quad (9)$$

(3)总投资最少 在政府决策中,有时会考虑利用政府有限的资金,在实现(7)的前提下,所需的总投资越少越好。因此,存在目标函数:

$$\min z = \sum_{i=1}^n a_i x_i \quad i=1,2,\dots,n \quad (10)$$

3.4 数学模型

通过上述分析不难看出,计划项目立项的决策过程属于0-1整数规划问题。由于6种约束条件和4个目标函数可以构成若干个不同的问题,因而也产生不同的数学模型。一般地,我们选取目标函数(7)与约束条件(1)~(6)的并集作为计划项目立项的数学模型,其他情况的问题可以稍做变化得出。因此,计划项目立项的数学模型为:

$$\max z = \sum_{i=1}^n c_i b_i x_i$$

$$\text{s. t. } \begin{cases} \sum_{i=1}^n b_i x_i \leq T \\ \sum_{i=1}^n x_i \leq k \\ k'_q \leq \sum_{x_i \in X_{1,q}} x_i \leq k_q \\ x_i = x_j \text{ 当 } x_i, x_j \in X_{2,q} \\ \sum_{x_i \in X_{3,q}} x_i = 1 \\ x_i = 1 \text{ 当 } x_i \in X_{4,1} \\ x_i = 0 \text{ 当 } x_i \in X_{4,0} \\ T > 0, n, k, q, k_q, k'_q \text{ 为正整数, } q < n \end{cases} \quad (\text{LP})$$

这里,约束条件(3)、(4)、(5)中的 q 代表的值不一定相同。

4 计算机求解方法

4.1 约束条件的形成

由于计算机不能够自动识别项目间的约束关系,因此对项目约束的判定只能由工作人员根据项目本身的特性来确定。具体操作时,可以针对6种不同的约束条件分别建立约束表来简化工作人员的判定和输入。约束条件的输入采取人机交互方式,提高了工作人员的主动性和灵活性,便于检查约束条件的正确性。

4.2 求解思路

0-1整数规划可以用传统的线性规划的单纯形方法求解,但要先松弛整数约束,求解后再进行还原,复杂性大且准确性差不是计算机处理的首选方案。0-1整数规划的求解也可以用二叉树与二元向量之间的对应关系,通过遍历二叉树求出最优解,常用的方法有加法算法和隐枚举法。但这种方法运算量比较大,算法的复杂性通常为 $O(2^n)$,计算机实现不太容易。我们从计划项目立项的实际出发,通过对各类约束条件的优先级分析,确定如下解决方案:

①项目的资金约束和总量约束贯穿于整个方案之中;

②总的来讲,项目的指定性约束优先级最高,其次是项目组的下界约束,然后是上界约束;

③具体来说,项目的指定约束优先级最高,其次是依赖约束,然后是排斥约束;

④单个项目的优先级低于项目组。

我们按照这个方案给出计划项目立项的计算机解决方案。

4.3 算法

先定义几个过程:

INX1(x_i) 判定项目 x_i 是否属于约束条件(3)的某个项目组。若是,输出结果1和 q 的值;若不是,输出结果0(注意,这里的结果0有两种可能:即不存在约束条件(3)或存在约束条件(3),但不属于任何一个项目组。下同。)

INX2(x_i) 判定项目 x_i 是否属于约束条件(4)的某个项目组。若是,输出结果1和 q 的值;若不是,输出结果0。

INX3(x_i) 判定项目 x_i 是否属于约束条件(5)的某个项目组。若是,输出结果1和 q 的值;若不是,输出结果0。

INX4(x_i) 判定项目 x_i 是否属于约束条件(6)的指定立项项目组。若是,输出结果1;若不是,输出结果0。

$INX40(x_i)$ 判定项目 x_i 是否属于约束条件(6)的指定不立项项目组。若是,输出结果1;若不是,输出结果0。

$N(X_{i,q})$ 计算项目 $X_{i,q}$ 组中的项目数量, $i=1,2,3,4$ 。

$P(x_i)$ 计算项目 x_i 所在的约束条件(2)的项目组中已确定立项的项目数。

算法:

设 n 个项目已按 c_i 由大到小进行排序,且令所有的 $x_i=2$;

输入: c_i, b_i, T, k, n 的值和约束条件;

```

step 1: 令  $B=0, R=0$ 
step 2: for  $i=1$  to  $n$  do {指定约束}
    if  $INX40(x_i)=1$ , then
         $x_i=0$ ;
    if  $INX2(x_i)=1$  then  $x_i=0$  for all  $x_i \in X_{2,q}$  endif;
endif;
if  $INX41(x_i)=1$ , then
     $x_i=1, R=R+1, B=B+b_i$ ;
if  $INX2(x_i)=1$  then
     $R=R+N(X_{2,q}), B=B+\sum_{x_i \in X_{2,q}} b_i$ ;
    for all  $x_i \in X_{2,q}$  do
         $x_i=1$ 
        if  $INX3(x_i)=1$  then
            for all  $x_j \in X_{3,q}-\{x_i\}$  do
                if  $x_j=2$  then  $x_j=0$  else goto lab endif;
            enddo;
        endif;
    enddo;
endif;
enddo;
step 3: if  $B>T$  or  $R \geq k$  then goto lab;
step 4: for  $i=1$  to  $n$  do {下界约束}
    if  $R+1 \leq k$  then
        if  $x_i=2$  and  $B+b_i \leq T$  then
            if  $INX1(x_i)=1$  and  $P(x_i) < k_q$  then
                if  $INX2(x_i)=0$  and  $INX3(x_i)=0$  then
                     $x_i=1, B=B+b_i, R=R+1$ 
                else
                    if  $INX2(x_i)=1$  then
                        if  $\sum_{x_i \in X_{2,q}} x_i = 2 \times N(X_{2,q})$  and  $R+N(X_{2,q}) \leq k$  and  $B+\sum_{x_i \in X_{2,q}} b_i \leq T$  then
                            for all  $x_j \in X_{2,q}$  do  $x_j=1$ ;
                            for all  $x_j \in X_{2,q}$  do
                                if  $P(x_j) > k_q$  then
                                    for all  $x_g \in X_{2,q}$  do  $x_g=0$ ;
                                    goto lab1;
                                endif;
                            enddo;
                             $R=R+N(X_{2,q}), B=B+\sum_{x_i \in X_{2,q}} b_i$ ;
                            for all  $x_j \in X_{2,q}$  do
                                if  $INX3(x_j)=1$  then
                                    for all  $x_g \in X_{3,q}-\{x_i\}$  do  $x_g=0$ ;
                                endif;
                            enddo;
                            goto lab2;
                        endif;
                    endif;
                else
                    if  $INX3(x_i)=1$  then
                         $x_i=1, B=B+b_i, R=R+1$ 
                        for all  $x_j \in X_{3,q}-\{x_i\}$  do  $x_j=0$ ;
                    endif;
                endif; (lab2)
            endif;
        endif;
    endif;
endif;
enddo;
step 7: there are errors in subject condition; (lab)
step 8: output the result  $x_i, i=1,2,\dots,n$ 

```

```

endif;
enddo;
step 5: if  $B>T$  or  $R \geq k$  then goto lab;
step 6: for  $i=1$  to  $n$  do {上界约束}
    if  $R+1 \leq k$  then
        if  $x_i=2$  and  $B+b_i \leq T$  then
            if  $INX1(x_i)=0$  then
                 $x_i=1, B=B+b_i, R=R+1$ 
            else
                if  $INX1(x_i)=1$  and  $P(x_i) < k_q$  then
                    if  $INX2(x_i)=0$  and  $INX3(x_i)=0$  then
                         $x_i=1, B=B+b_i, R=R+1$ 
                    else
                        if  $INX2(x_i)=1$  then
                            if  $\sum_{x_i \in X_{2,q}} x_i = 2 \times N(X_{2,q})$  and  $R+N(X_{2,q}) \leq k$  and  $B+\sum_{x_i \in X_{2,q}} b_i \leq T$  then
                                for all  $x_j \in X_{2,q}$  do  $x_j=1$ ;
                                for all  $x_j \in X_{2,q}$  do
                                    if  $P(x_j) > k_q$  then
                                        for all  $x_g \in X_{2,q}$  do  $x_g=0$ ;
                                        goto lab2;
                                    endif;
                                enddo;
                                 $R=R+N(X_{2,q}), B=B+\sum_{x_i \in X_{2,q}} b_i$ ;
                                for all  $x_j \in X_{2,q}$  do
                                    if  $INX3(x_j)=1$  then
                                        for all  $x_g \in X_{3,q}-\{x_i\}$  do  $x_g=0$ ;
                                    endif;
                                enddo;
                                goto lab2;
                            endif;
                        else
                            if  $INX3(x_i)=1$  then
                                 $x_i=1, B=B+b_i, R=R+1$ 
                                for all  $x_j \in X_{3,q}-\{x_i\}$  do  $x_j=0$ ;
                            endif;
                        endif; (lab2)
                    endif;
                else
                    if  $INX3(x_i)=1$  then
                         $x_i=1, B=B+b_i, R=R+1$ 
                        for all  $x_j \in X_{3,q}-\{x_i\}$  do  $x_j=0$ ;
                    endif;
                endif;
            endif;
        endif;
    endif;
endif;
enddo;
step 7: there are errors in subject condition; (lab)
step 8: output the result  $x_i, i=1,2,\dots,n$ 

```

可以看出,该算法总体上是遵循背包问题的近似求解思路。由于背包问题是一个典型的 NP-hard 问题^[1],在多项式时间内不可能得到最优解,只能得到近似解。但这种方法简单有效,在实际应用中具有广泛的推广价值。本文给出的算法时间复杂性为 $O(n^2)$ 。

参 考 文 献

- 1 Garey M R, Johnson D S. Computers and Intractability. A Guide to the theory of NP-Completeness, 1979
- 2 Ibarra O H, Kim C E. Fast Approximations Algorithms for the Knapsack and Sum of Subset Problem. JACM 1975, 22: 463~468
- 3 Lawler E L. Fast Approximation Algorithms for Knapsack Problem. In: 18th Annual Symposium on Foundations of Computer Science, 1977
- 4 郑汉鼎,刁在筠. 数学规划. 山东教育出版社, 1997
- 5 官梅谷,郑汉鼎. 线性规划. 山东科学技术出版社, 1987
- 6 马绍汉. 一类背包问题的可解性. 山东大学学报, 1985(4): 37~42