

МЕТОД СЕТЕВОГО ПРОГРАММИРОВАНИЯ

В. Н. Бурков, И. В. Буркова, М. В. Попок, Т. И. Овчинникова

Институт проблем управления им. В.А. Трапезникова, г. Москва

Предложен новый подход к задачам дискретной оптимизации, названный методом сетевого программирования, в основу которого положена возможность представления функции многих переменных в виде суперпозиции более простых функций. Структура такой суперпозиции представляется в виде сети, входы которой соответствуют переменным, а выходы — функции. Показано, что если сеть является деревом, то решение задачи сводится к последовательному решению более простых задач. В общем случае предложено преобразовать сеть в дерево путем разделения вершин сети. Доказано, что решение задачи для преобразованной структуры дает нижнюю оценку для целевой функции исходной задачи (если решается задача минимизации). Метод проиллюстрирован на примере известной задачи о камнях.

ВВЕДЕНИЕ

Многие задачи дискретной оптимизации сводятся к следующей постановке: определить вектор $x = \{x_i\}$ с дискретными компонентами, минимизирующий аддитивную функцию

$$\varphi(x) = \sum_{i=1}^n \varphi_i(x_i) \quad (1)$$

при ограничении

$$f(x) \geq b. \quad (2)$$

Любая функция дискретных переменных допускает сетевое представление, такое, что вычисление значений функции сводится к последовательному вычислению значений более простых функций. В частности, любая функция дискретных переменных допускает дихотомическое представление, когда вычисление значения функции сводится к последовательному вычислению значений функций двух переменных. Так, функция $f(x) = f_0[f_1(x_1, x_2), f_2(x_2, x_3)]$ допускает дихотомическое представление (рис. 1). При этом функции f_0, f_1 и f_2 удобно представлять в матричном виде (рис. 2). Такое представление широко используется в методах комплексного оценивания программ развития предприятий, регионов, результатов деятельности подразделений, уровня безопасности объектов и др.

В работах [1, 2] доказаны теоремы о представлении непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных (в частности, двух переменных). Так, например, любая непрерывная функция трех переменных

представима в виде [2] $f(x_1, x_2, x_3) = h^1(x_1, \varphi_1(x_2, x_3)) + h^2(x_1, \varphi_2(x_2, x_3)) + h^3(x_1, \varphi_3(x_2, x_3))$. Ее сетевое представление приведено на рис. 3.

В сетевом виде можно представить и систему неравенств. Рассмотрим, например, систему неравенств

$$f_j(x) \leq b_j, \quad j = \overline{1, m}. \quad (3)$$

Без ограничения общности можно принять, что b_j — положительные и одинаковые числа, $b_j = b > 0$. В этом случае систему неравенств (3) можно заменить одним неравенством $f(x) \leq b$, где $f(x) = \max_j f_j(x)$.

Очевидно, что функция $f(x)$ допускает сетевое представление, если все функции f_j допускают такое представление.

В настоящей работе описывается новый метод решения задач дискретной оптимизации, использующий сетевое представление функции $f(x)$. Его естественно на-

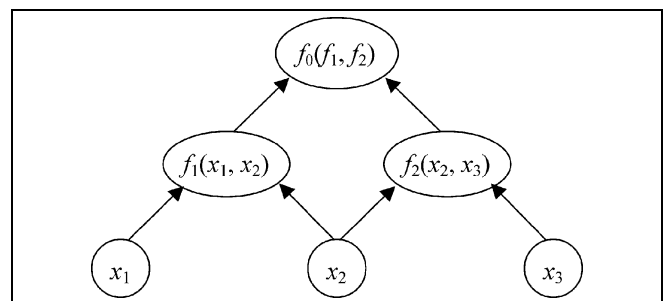


Рис. 1. Дихотомическое представление функции дискретных переменных

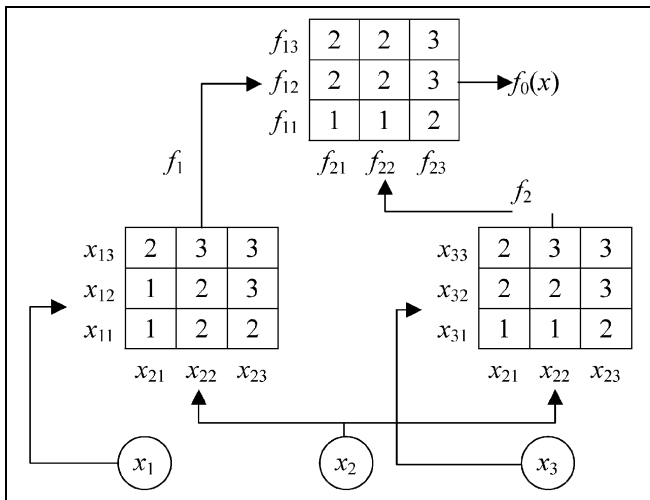
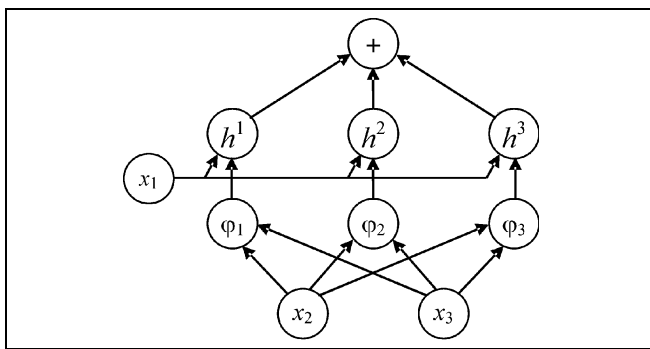

 Рис. 2. Представление функций f_0 , f_1 и f_2 в матричном виде


Рис. 3. Сетевое представление непрерывных функций трех переменных

звать методом сетевого программирования (в частном случае дихотомического представления получаем метод дихотомического программирования [3]).

1. СЕТЕВОЕ ПРЕДСТАВЛЕНИЕ ТИПА ДЕРЕВА

Рассмотрим случай, когда функция $f(x)$ допускает сетевое представление в виде дерева. Дадим описание метода сетевого программирования для задачи (1), (2). На рис. 4 приведен пример функции трех переменных, имеющей вид

$$f(x_1, x_2, x_3) = \varphi_0[f_1(x_1, x_2), x_3] = \varphi_0(y, x_3).$$

Значения функций $\varphi_i(x_i)$ даны в нижних половинах квадратов, соответствующих переменным x_1 , x_2 и x_3 . Идея метода состоит в следующем. Сначала решается задача минимизации функции двух переменных

$$\varphi_1(x_1) + \varphi_2(x_2)$$

при ограничении

$$f_1(x_1, x_2) \geq y,$$

соответствующая начальной вершине сетевого представления. Обозначим через $z(y)$ решение этой задачи в

зависимости от y . Далее решаем задачу минимизации функции тоже двух переменных

$$z(y) + \varphi_3(x_3)$$

при ограничении

$$\varphi_0(y, x_3) \geq b,$$

соответствующую конечной вершине сетевого представления. Решение этой задачи определяет оптимальное решение исходной задачи.

Проиллюстрируем метод на примере рис. 4.

1 шаг. Рассматриваем нижнюю матрицу и для каждого элемента этой матрицы записываем в нижней половине соответствующей клетки сумму функций $\varphi_1(x_1)$ и $\varphi_2(x_2)$ для соответствующих значений x_1 и x_2 . Так, например, клетке $(x_1, x_2) = (3, 2)$ соответствует сумма

$$\varphi_1(3) + \varphi_2(2) = 20 + 10 = 30.$$

Далее будем называть эту сумму затратами на достижение соответствующего состояния.

2 шаг. Из всех элементов матрицы, имеющих одно и то же значение $y = f_1(x_1, x_2)$, выбираем элемент с минимальной суммой $\varphi_1(x_1) + \varphi_2(x_2)$. Минимальную сумму записываем в нижнюю половину клетки, соответствующей этому значению y в верхней матрице. Так, например, значению $y = 3$ соответствуют 5 элементов нижней матрицы: (3; 2), (4; 2), (3; 3), (4; 3) и (2; 4). Из них элемент (3; 2) имеет минимальную сумму 30 (это число записано в нижней половине соответствующей клетки).

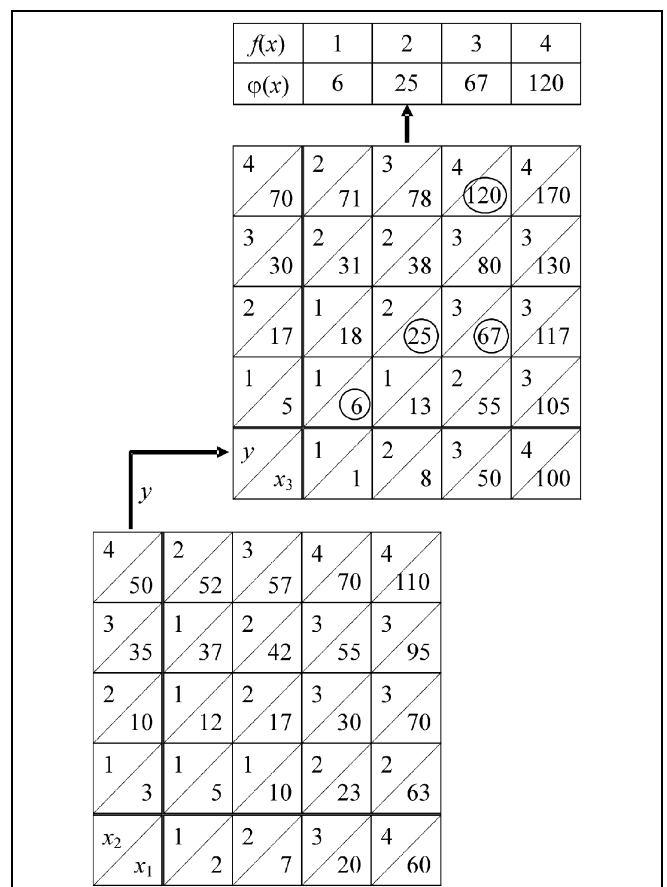


Рис. 4. Иллюстрация метода сетевого программирования

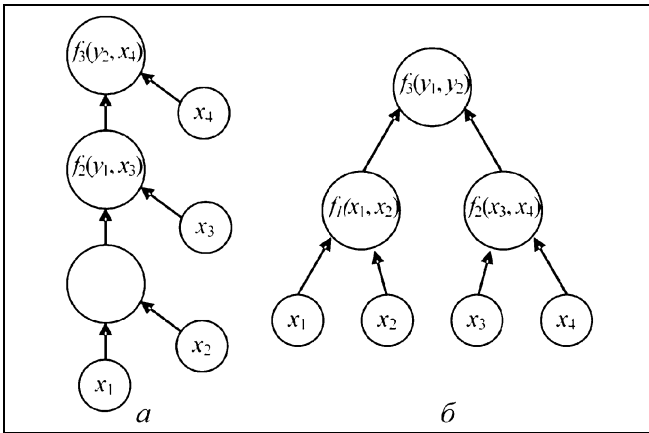


Рис. 5. Иллюстрация методов программирования:
a — динамического (ветвь дерева); *б* — дихотомического (произвольное дерево)

Поэтому в верхней матрице значению $y = 3$ соответствует число 30, записанное в нижней половине соответствующей клетки.

Далее шаги 1 и 2 повторяются для верхней матрицы. В результате для каждого значения $f(x)$ мы получаем минимальное значение $\varphi(x)$. На рис. 4 кружками выделены минимальные затраты.

Несложно обобщить описанный метод на случай произвольного сетевого представления функции $f(x)$ в виде дерева. Главное, чтобы задачи, соответствующие каждой вершине сетевого представления, имели эффективные методы решения. В случае дихотомического представления это всегда имеет место.

Заметим, что дихотомическое представление (см. рис. 4) имеет структуру в виде ветви дерева. В этом случае метод дихотомического программирования переходит в метод динамического программирования. Таким образом, метод дихотомического программирования в случае дихотомического представления в виде дерева является обобщением метода динамического программирования, расширяя круг задач, решаемых на основе данного подхода (рис. 5).

Если в методе динамического программирования решением задачи является путь в некоторой специально построенной сети, то в методе дихотомического программирования решением задачи является частичное дерево в некотором специально построенном дереве. Соответственно, принцип оптимальности в методе дихотомического программирования можно сформулировать следующим образом: любое поддерево оптимального дерева должно быть оптимальным. Формально этот принцип оптимальности можно записать следующим образом: $\varphi_k(y) = \min_{(i,j) \in p(y)} [\varphi_i(y_i) + \varphi_j(y_j)]$, где $p(y)$ — множество пар (i, j) таких, что $f_k(y_i, y_j) = y$.

2. ОБЩИЙ СЛУЧАЙ

Рассмотрим произвольное сетевое представление функции $f(x)$, задаваемое сетью, выходом которой является вершина, соответствующая функции $f(x)$, а входами — вершины, соответствующие переменным $x_i, i = \overline{1, n}$.

Рассмотрим множество начальных вершин, которые не являются висячими, т. е. их степени исхода больше единицы. Разделим произвольным образом затраты $\varphi_i(x_i)$ на k_i частей, где k_i — число исходящих дуг. Фактически мы как бы разделили вершину i на k_i висячих вершин с соответствующей частью затрат. Далее применяем описанный выше алгоритм. При этом каждый раз, когда встречается вершина со степенью исхода больше единицы, мы делим затраты на соответствующее число частей. В результате применения алгоритма мы получим оптимальное решение для модифицированной сети. Однако это решение может не быть решением исходной задачи. Тем не менее, имеет место следующая

Теорема. *Полученное с помощью вышеописанного алгоритма решение дает нижнюю оценку оптимального решения исходной задачи.*

Доказательство. Заметим, что множество решений модифицированной сети содержит все решения исходной задачи. Эти решения имеют следующий вид. Если из вершины, соответствующей переменной x_{ik} , исходит хотя бы одна дуга полученного решения, то все дуги, исходящие из этой вершины, также принадлежат полученному решению. Отсюда следует, что полученное оптимальное решение модифицированной задачи дает нижнюю оценку для оптимального решения исходной задачи.

Пример. Рассмотрим сеть, представленную на рис. 1 и 2. На рис. 6 приведено решение соответствующей задачи. Затраты $\varphi_2(x_2)$ разделены на две части, поскольку переменная x_2 используется при вычислении функций и f_1 , и f_2 . В данном случае общие затраты, равные 8, 12 и 20 при значениях переменной x_2 , равных 1, 2 и 3, соответственно, поделены пополам.

В каждой матрице кружками выделены клетки, соответствующие минимальным затратам на получение того или иного значения функций f_1, f_2 и f_0 . В результате по-

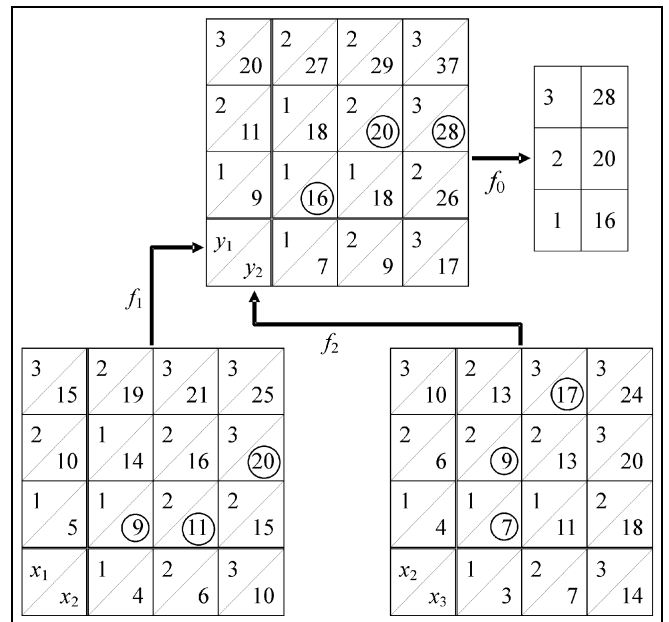


Рис. 6. Решение оценочной задачи методом сетевого программирования

лучены минимальные затраты $\varphi(f_0)$, требуемые для получения значений функции f_0 . Если $f_0 = 1$, то $\varphi(1) = 16$, если $f_0 = 2$, то $\varphi(2) = 20$, если $f_0 = 3$, то $\varphi(3) = 28$.

Рассмотрим случай $f_0 = 2$. Ему соответствует оптимальное решение модифицированной задачи: $x_1 = 1$, $x_{21} = 2$, $x_{22} = 2$, $x_3 = 1$.

Здесь x_{21} соответствует значению x_2 в левой нижней матрице, а x_{22} — в правой нижней матрице. Поскольку оба значения $x_{21} = 2$, $x_{22} = 2$ вошли в оптимальное решение модифицированной задачи, то полученное решение является допустимым для исходной задачи, а значит, мы получили оптимальное решение исходной задачи.

Другая ситуация возникает в случае $f_0 = 3$. Оптимальное решение модифицированной задачи имеет вид: $x_1 = 1$, $x_{21} = 2$, $x_{22} = 3$, $x_3 = 2$ со значением затрат $\varphi_0 = 28$. Это решение не является допустимым для исходной задачи, поэтому значение $\varphi_0 = 28$ является нижней оценкой минимальных затрат для исходной задачи. Здесь возможны два варианта действий. Первый заключается в попытке улучшить нижнюю оценку, изменяя разбиение затрат $c_2 = \varphi_2(x_2)$ на две части — c_{21} и c_{22} . Очевидно, что для улучшения оценки следует часть c_{21} увеличить, а часть c_{22} уменьшить. Возьмем, например, $c_{21} = 10$, а $c_{22} = 2$. В этом случае оптимальное решение модифицированной задачи будет иметь вид: $x_1 = 1$, $x_{21} = 2$, $x_{22} = 2$, $x_3 = 3$ с затратами $\varphi_0 = 31$. Это решение является допустимым для исходной задачи, а значит, оптимальным. Однако изменение разбиения затрат на части может и не привести к получению допустимого решения для исходной задачи.

Второй вариант состоит в применении метода ветвей и границ. Разобьем множество всех решений исходной задачи на два подмножества: в первом из них $x_2 \leq 2$, а во втором $x_2 = 3$, и применим описанный выше алгоритм. Получим оценку снизу для первого подмножества. Получаем следующее решение: $x_1 = 1$, $x_{21} = 2$, $x_{22} = 2$, $x_3 = 3$ с затратами $\varphi_0 = 31$.

Получим оценку снизу для второго подмножества. Оптимальное решение модифицированной задачи имеет вид: $x_1 = 1$, $x_2 = 3$, $x_3 = 2$ со значением затрат $\varphi_0 = 32$.

Выбираем первое подмножество с минимальной оценкой. Поскольку полученное решение модифицированной задачи является допустимым для исходной задачи, то оно оптимально.

Рассмотрим на ряде задач построение оценочной задачи и метод ветвей и границ на основе полученной оценки.

3. ЗАДАЧА ЦЕЛОЧИСЛЕННОГО ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Рассмотрим следующую постановку задачи целочисленного линейного программирования. Определить целочисленный неотрицательный вектор $x = \{x_1, x_2, \dots, x_n\}$, максимизирующий функцию

$$\varphi(x) = \sum_{i=1}^n c_i x_i \quad (4)$$

при ограничениях

$$\sum_i a_{ij} x_i \leq b_j, \quad j = \overline{1, m}. \quad (5)$$

Для построения оценочной задачи разделим c_i на m частей s_{ij} так, что

$$\sum_j s_{ij} = c_i, \quad i = \overline{1, n}, \quad (6)$$

и рассмотрим m задач целочисленного линейного программирования следующего вида: определить целочисленный вектор x , максимизирующий функцию

$$s_j(x) = \sum_i s_{ij} x_i \quad (7)$$

при ограничениях

$$\sum_i a_{ij} x_i \leq b_j. \quad (8)$$

Обозначим через $\Phi_j(s_j)$ оптимальное решение j -й задачи ($s_j = \{s_{ij}\}$). Согласно теореме, величина

$$\Phi(s) = \sum_{j=1}^m \Phi_j(s_j) \quad (9)$$

является оценкой сверху оптимального решения исходной задачи. Окончательно получаем следующую формулировку оценочной задачи: определить значения $\{s_{ij}\}$, $i = \overline{1, n}$, $j = \overline{1, m}$, максимизирующие величину $\Phi(s)$ при ограничениях (6). Оценочную задачу назовем двойственной к исходной задаче целочисленного линейного программирования. Обоснованием этого названия служит следующая интересная связь. Рассмотрим обычную задачу линейного программирования (4), (5) (без требования целочисленности). Для упрощения выводов примем, что все параметры системы ограничений — положительные числа. Заметим, что если не требовать целочисленности решений, то задача (7), (8) легко решается. Ее оптимальное решение:

$$x_{ij} = \begin{cases} b_j/a_{kj}, & \text{если } s_{kj}/a_{kj} = \max_q (s_{qj}/a_{qj}), \\ 0, & \text{если } i \neq k. \end{cases}$$

Оптимальное значение величины $\Phi_j(s_j) = b_j \max_q (s_{qj}/a_{qj})$.

Обозначим $y_j = \max_q s_{qj}/a_{qj}$, $j = \overline{1, m}$. Заметим, что $y_j \geq s_{qj}/a_{qj}$ для всех q . Увеличим s_{qj} так, чтобы $s_{qj} = y_j a_{qj}$. Тогда оценочная задача (6), (9) запишется в следующем виде: определить значения $y_j \geq 0$, $j = \overline{1, m}$, минимизирующие функцию

$$B(y) = \sum_j b_j y_j$$

при ограничениях

$$\sum_j a_{ij} y_j \geq c_i, \quad i = \overline{1, n}.$$

Таким образом, в непрерывном случае оценочная задача становится двойственной задачей линейного программирования.

Рассмотрим на примере применение метода ветвей и границ для решения задачи целочисленного линейного программирования.



Пример. Определить значения $x_i = \{0; 1\}$, $i = \overline{1, n}$, максимизирующие функцию

$$\Phi(x) = 10x_1 + 8x_2 + 6x_3 + 7x_4$$

при ограничениях

$$\begin{aligned} 6x_1 + 3x_2 + 2x_3 + 5x_4 &\leq 11, \\ 3x_1 + 5x_2 + 6x_3 + 3x_4 &\leq 11. \end{aligned}$$

Для построения оценочных задач возьмем $s_{i1} = a_{i1}$, $s_{i2} = c_i - a_{i1}$, $i = \overline{1, 4}$. Получаем две задачи о ранце.

Задача 1.

$$\begin{aligned} \varphi_1 &= \max(6x_1 + 3x_2 + 2x_3 + 5x_4), \\ 6x_1 + 3x_2 + 2x_3 + 5x_4 &\leq 11. \end{aligned}$$

Она имеет два решения: $x_1 = x_4 = 1$, $x_2 = x_3 = 0$ и $x_1 = x_2 = x_3 = 1$, $x_4 = 0$. В обоих случаях $\varphi_1 = 11$.

Задача 2.

$$\begin{aligned} \varphi_2 &= \max(4x_1 + 5x_2 + 4x_3 + 2x_4), \\ 3x_1 + 5x_2 + 6x_3 + 3x_4 &\leq 11. \end{aligned}$$

Ее решение: $x_1 = x_2 = x_4 = 1$, $x_3 = 0$, $\varphi_2 = 11$. Оценка сверху исходной задачи $\varphi_0 = \varphi_1 + \varphi_2 = 22$.

Для улучшения оценки увеличим s_{21} и s_{41} на единицу, уменьшив на единицу s_{22} и s_{42} . Получаем две новые оценочные задачи.

Задача 1.

$$\begin{aligned} \varphi_1 &= \max(6x_1 + 4x_2 + 2x_3 + 6x_4), \\ 6x_1 + 3x_2 + 2x_3 + 5x_4 &\leq 11. \end{aligned} \quad (10)$$

Ее решения: $x_1 = x_4 = 1$, $x_2 = x_3 = 0$; $x_1 = x_2 = x_3 = 1$, $x_4 = 0$; $x_2 = x_3 = x_4 = 1$, $x_1 = 0$; $\varphi_1 = 12$.

Задача 2.

$$\begin{aligned} \varphi_2 &= \max(4x_1 + 4x_2 + 4x_3 + 1x_4), \\ 3x_1 + 5x_2 + 6x_3 + 3x_4 &\leq 11. \end{aligned} \quad (11)$$

Ее решение: $x_1 = x_2 = x_4 = 1$, $x_3 = 0$; $\varphi_2 = 9$.

Оценка сверху исходной задачи уменьшилась на единицу: $\varphi_0 = \varphi_1 + \varphi_2 = 21$.

Применим метод ветвей и границ. Разобьем множество всех решений на два подмножества. В первом подмножестве $x_1 = 1$, а во втором $x_1 = 0$.

Оценим первое подмножество. Положив в ограничениях (10) и (11) $x_1 = 1$, получим следующие две задачи:

Задача 1.

$$\begin{aligned} \varphi_1 &= \max(4x_2 + 2x_3 + 6x_4), \\ 3x_2 + 2x_3 + 5x_4 &\leq 5. \end{aligned}$$

Ее решения: $x_4 = 1$, $x_2 = x_3 = 0$; $x_2 = x_3 = 1$, $x_4 = 0$; $\varphi_1 = 6$.

Задача 2.

$$\begin{aligned} \varphi_2 &= \max(4x_2 + 4x_3 + 1x_4), \\ 5x_2 + 6x_3 + 3x_4 &\leq 8. \end{aligned}$$

Ее решение: $x_2 = x_4 = 1$, $x_3 = 0$; $\varphi_2 = 5$.

Оценка сверху первого подмножества: $\varphi_0 = \varphi_1 + \varphi_2 + c_1 = 21$.

Оценим второе подмножество ($x_1 = 0$). Заметим, что при $x_1 = 0$ любое решение является допустимым для первой оценочной задачи. Поэтому достаточно решить вторую задачу, положив $s_{i2} = c_i$, $i = 2, 3, 4$. Ее решение

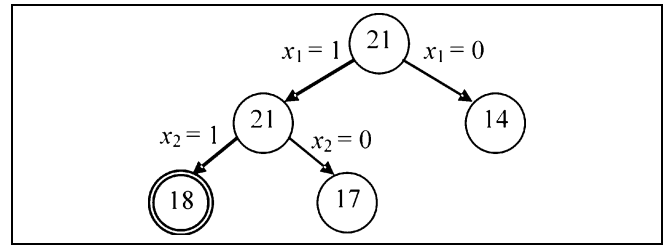


Рис. 7. Дерево ветвлений в задаче целочисленного программирования

$x_2 = x_3 = 1$, $x_4 = 0$ является оптимальным во втором подмножестве со значением целевой функции $\varphi_0 = 14$. Выбираем первое подмножество, имеющее большую оценку. Разбиваем первое подмножество на два. В одном из них $x_2 = 1$, а в другом $x_2 = 0$.

Оценим первое подмножество ($x_2 = 1$). Рассматривая два ограничения

$$\begin{aligned} 2x_3 + 5x_4 &\leq 2, \\ 6x_3 + 3x_4 &\leq 3, \end{aligned}$$

видим, что единственное решение $x_3 = x_4 = 0$, следовательно, оно оптимально в данном подмножестве со значением целевой функции $\varphi_0 = 18$.

Оценим второе подмножество ($x_2 = 0$). В данном случае достаточно сравнить два варианта: $x_3 = 1$, $x_4 = 0$; $\varphi_1 = 6$ и $x_3 = 0$, $x_4 = 1$; $\varphi_1 = 7$.

Оценка второго подмножества: $\varphi_0 = 10 + 7 = 17$. Ей соответствует оптимальное решение в этом подмножестве $x_1 = x_4 = 1$, $x_2 = x_3 = 0$ со значением целевой функции $\varphi_0 = 17$.

Выбираем первое подмножество, а следовательно, и оптимальное решение $x_1 = x_2 = 1$, $x_3 = x_4 = 0$; $\varphi_0 = 18$. Дерево ветвлений приведено на рис. 7.

Интересно сравнить описанный способ получения оценок для задач целочисленного линейного программирования с известными способами. В основном применяются два способа. В соответствии с первым из них решаются m задач о ранце с каждым ограничением отдельно. Очевидно, что наихудшее решение (по значению целевой функции) определяет оценку сверху исходной задачи. Для нашего примера имеем две задачи о ранце.

Задача 1.

$$\begin{aligned} \max(10x_1 + 8x_2 + 6x_3 + 7x_4), \\ 6x_1 + 3x_2 = 2x_3 + 5x_4 \leq 11. \end{aligned}$$

Ее решение $x_1 = x_2 = x_3 = 1$, $x_4 = 0$; $\varphi_1 = 24$.

Задача 2.

$$\begin{aligned} \max(10x_1 + 8x_2 + 6x_3 + 7x_4), \\ 3x_1 + 5x_2 = 6x_3 + x_4 \leq 11. \end{aligned}$$

Ее решение $x_1 = x_2 = x_4 = 1$, $x_3 = 0$; $\varphi_1 = 25$.

Таким образом, оценка сверху оптимального решения исходной задачи в данном случае равна 24, что больше чем оценка 21, полученная методом сетевого программирования. Более того, поскольку обе рассмотренные задачи о ранце получаются в методе сетевого программирования (первая — при $s_{i1} = c_i$, $s_{i2} = 0$, а вто-

рая, наоборот, при $s_{12} = c_i, s_{i1} = 0$, то можно утверждать, что применение метода сетевого программирования даст лучшие (или такие же) оценки. Согласно второму способу решается задача линейного программирования без требования целочисленности.

Рассмотрим простой

Пример. Определить значения $x_i = \{0; 1\}, i = 1, 2$, максимизирующие функцию

$$\varphi(x) = 42x_1 + 14x_2$$

при ограничениях

$$\begin{aligned} 3x_1 + 4x_2 &\leq 6, \\ 5x_1 + 2x_2 &\leq 6. \end{aligned}$$

Получим оценку методом сетевого программирования. Для этого достаточно положить $s_{11} = 42, s_{21} = 14, s_{12} = s_{22} = 0$. Решение первой оценочной задачи $x_1 = 1, x_2 = 0, \varphi_1 = 42$ является допустимым для второй задачи и, следовательно, оптимально.

Получим оценку, решая нецелочисленную задачу линейного программирования. Ее решение $x_1 = 6/7; x_2 = 6/7; \varphi_0 = 48 > 42$. Как видим, оценка существенно хуже.

4. РЕШЕНИЕ ЗАДАЧИ О КАМНЯХ МЕТОДОМ СЕТЕВОГО ПРОГРАММИРОВАНИЯ

Рассмотрим постановку задачи о камнях. Имеется n камней разного веса. Требуется разбить их на m групп (куч) так, чтобы максимальный вес камней в группе был минимален. Задача о камнях имеет многочисленные варианты применения (равномерное распределение работ между исполнителями, функций по подразделениям организационной структуры и др.). Дадим ее формальную постановку.

Задача 1. Обозначим через a_i вес i -го камня, $x_{ij} = 1$, если камень i попал в j -ю кучку, $x_{ij} = 0$ — в противном случае. Суммарный вес камней в j -й группе $T_j = \sum_i a_i x_{ij}$.

Максимальный вес группы

$$T = \max_j \sum_i a_i x_{ij} \rightarrow \min. \quad (12)$$

Поскольку каждый камень должен быть помещен только в одну группу, имеем ограничения:

$$\sum_j x_{ij} = 1, \quad i = \overline{1, n}. \quad (13)$$

Задача заключается в минимизации функции (12) при ограничениях (13). Мы будем рассматривать вспомогательную задачу следующего вида.

Задача 2. Фиксируем допустимый вес каждой группы T и сформулируем следующую задачу: максимизировать сумму весов размещенных в ящики вместимостью T камней:

$$\Phi = \sum_{i,j} a_i x_{ij} \rightarrow \max \quad (14)$$

при ограничениях (13) и

$$\sum_i a_i x_{ij} \leq T, \quad j = \overline{1, m}. \quad (15)$$

Связь между задачами (12), (13) и (14), (15) очевидна. Минимальное значение T , при котором в оптимальном

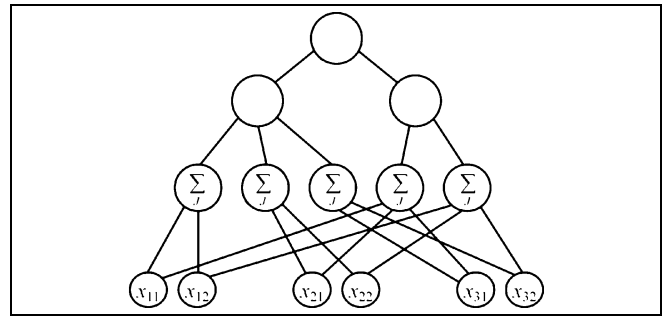


Рис. 8. Сетевое представление ограничений задачи о камнях

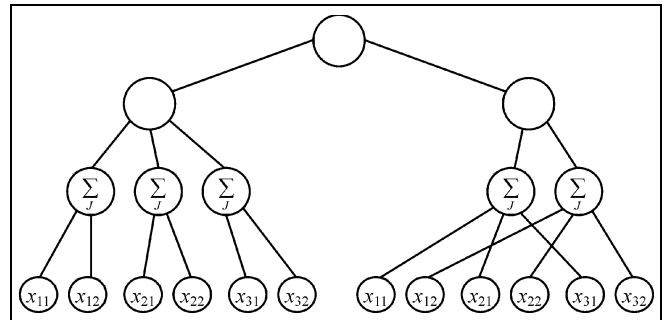


Рис. 9. Сетевое представление оценочной задачи

решении задачи 2 размещены все камни, определяет оптимальное решение задачи 1.

Сначала получим сетевое представление задачи 2. Оно в агрегированном виде представлено на рис. 8 для случая $n = 3, m = 2$.

Поскольку структура сетевого представления имеет вид сети, а не дерева, то для построения оценочной задачи разделяем каждую вершину нижнего уровня на две вершины. Преобразованная структура приведена на рис. 9. Все a_i также делим на две части u_{ij} и v_{ij} для каждой вершины нижнего уровня так, что

$$u_{ij} + v_{ij} = a_i \text{ для всех } i, j. \quad (16)$$

Рассмотрим следующие две задачи.

Задача 1. Определить значения x_{ij} , максимизирующие функцию $\sum_{i,j} u_{ij} x_{ij}$ при ограничениях (13).

Задача 2. Максимизировать функцию $\sum_{i,j} v_{ij} x_{ij}$ при ограничениях (15).

Обозначим через $S_m(u)$ и $L_m(v)$ оптимальные решения первой и второй задач при заданных величинах u_{ij} и v_{ij} . Оценочная задача заключается в определении значений $\{u_{ij}\}$ и $\{v_{ij}\}$, минимизирующих функцию

$$F(u, v) = S_m(u) + L_m(v) \quad (17)$$

при ограничениях (16).

Заметим, во-первых, что в оптимальных решениях первой и второй задач можно принять $u_{ij} = y_i, v_{ij} = a_i - y_i, j = \overline{1, m}$. Во-вторых, решение задачи 1 очевидно: $S_m(y) = \sum_i y_i$. В третьих, решение m задач 2 при заданных вели-



чинах $\{y_i\}$ сводится к решению одной задачи о ранце: оп-
ределить значения $x_i = \overline{0, 1}$, максимизирующие функцию

$$\sum_i x_i(a_i - y_i) \tag{18}$$

при ограничении

$$\sum_i x_i a_i \leq T. \tag{19}$$

Решим задачу (18), (19) при $y_i = 0, i = \overline{1, n}$.

Обозначим через $Q = \{Q_j\}$ множество векторов x ,
удовлетворяющих ограничению (19) и упорядоченных
по убыванию величин $M_j = \sum_{i \in Q_j} a_i, Y_j = \sum_{i \in Q_j} y_j$, а

$$Z = \max_j (M_j - Y_j).$$

Заметим, что при заданных $\{y_i\}$ величина Z опреде-
ляет оптимальное решение каждой из m задач 2. Оценка
(17) при этом имеет вид

$$F(y) = mZ + \sum_i y_i, \tag{20}$$

где $y_i \geq 0$ удовлетворяют неравенствам

$$\sum_{i \in Q_j} y_i + Z \geq M_j, \quad j = \overline{1, N}, \tag{21}$$

где N — число различных решений неравенства (19). Та-
ким образом, оценочная задача свелась к определению
значений $0 \leq y_i \leq a_i, i = \overline{1, n}$ и $0 \leq Z \leq M_j$, максими-
зирующих функцию (20) при ограничениях (21). Это
обычная задача линейного программирования.

Фиксируем величину Z и определяем максимальный
номер k такой, что $Z < M_k$. Рассматриваем следующую
задачу линейного программирования: определить значе-
ния $0 \leq y_i \leq a_i, i = \overline{1, n}$, минимизирующие функцию

$$Y(Z) = \sum_i y_i,$$

при ограничениях (21), где $j = \overline{1, k}$. Двойственная задача
имеет вид: определить значения $u_j \geq 0, j = \overline{1, k}$, макси-
мизирующие функцию

$$\sum_{j=1}^k (M_j - Z)u_j$$

при ограничениях

$$\sum_{j \in R_i} u_j \leq 1, \quad i = \overline{1, n},$$

где R_i — множество групп j , содержащих камень i .

Обозначим через $Y_0(Z)$ минимальное значение $Y(Z)$.
Оценочная задача сводится к минимизации функции
одной переменной $Y_0(Z) + mZ \rightarrow \min$.

Берем $T_0 = A/m$, где $A = \sum_i a_i$, и решаем задачу 2. Если
 $\Phi_{\max}(T_0) < A$, то увеличиваем T_0 до T_1 так, чтобы появил-
ся хотя бы один новый вектор Q_j . Если $\Phi_{\max}(T_1) < A$, то
продолжаем увеличивать T до тех пор, пока не получим
величину T_k такую, что $\Phi_{\max}(T_k) \geq A$. Величина T_k явля-
ется нижней оценкой для задачи 1. Далее можно при-
менить метод ветвей и границ на основе полученной
оценки.

Пример. Пусть $m = 3$ и имеется 7 камней весов a_i :

i	1	2	3	4	5	6	7
a_i	10	12	13	14	18	19	22

1 шаг. Имеем $A = 108, T_0 = 36$. Имеется только одно
решение: $Q = (4, 7)$ со значением $M = 36$.

2 шаг. Увеличиваем T_0 до $T_1 = 37$. Имеются следую-
щие решения: $Q_1 = (5, 6), Q_2 = (4, 7), Q_3 = (1, 2, 3),$
 $Q_4 = (3, 7)$. Соответственно, $M_1 = 37, M_2 = 36, M_3 = 35,$
 $M_4 = 35$. Выпишем систему неравенств:

$$\begin{matrix} & & & & y_5 + & y_6 + & & z \geq 37 \\ & & & y_4 + & & & y_7 + & z \geq 36 \\ y_1 + & y_2 + & y_3 + & & & & & z \geq 35 \\ & & y_3 + & & & & y_7 + & z \geq 35 \end{matrix}$$

Имеем:

$$\begin{aligned} Z_0 &= 37, \quad y_i = 0, \quad Y_0(37) = 111, \\ Z_1 &= 36, \quad y_5 = 1, \quad Y_0(36) = 109, \\ Z_2 &= 35, \quad y_5 = 2, \quad y_4 = 1, \quad Y_0(35) = 108. \end{aligned}$$

Нетрудно показать, что дальнейшее уменьшение ве-
личины Z не приводит к уменьшению оценки. Поэтому
оптимальное значение $Z_0 = Z_2 = 35$.

Берем $x_{51} = x_{61} = 1$, т. е. помещаем камни 5 и 6 в пер-
вую группу. Исключая эти камни, рассматриваем зада-
чу меньшей размерности. Имеем для нее также $Y_0(35) =$
 $= 71 = A - 37$. Получаем оптимальное решение: $x_{12} =$
 $= x_{22} = x_{32} = 1, x_{43} = x_{73} = 1$ со значением $T_{\min} = 37$.

ЗАКЛЮЧЕНИЕ

Предложенный подход к решению задач дискретной
оптимизации применим к любой задаче данного класса,
поскольку любая функция дискретных переменных до-
пускает сетевое представление. Если целевая функция
не аддитивная, то вводом новой переменной всегда
можно перейти к задаче с аддитивной целевой функци-
ей, как это показано на примере задачи о камнях. Воп-
рос об эффективности применения метода сетевого
программирования к решению конкретных задач диск-
ретной оптимизации, конечно, требует дальнейших ис-
следований. Заметим, что метод можно применить и к
непрерывным задачам.

ЛИТЕРАТУРА

1. Арнольд В. И. О функциях трех переменных // Доклады АН СССР. — 1957. — № 2.
2. Колмогоров А. Н. О представлении непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных // Доклады АН СССР. — 1956. — № 2.
3. Бурков В. Н., Буркова И. В. Задачи дихотомической оптимизации // Материалы междунар. науч.-техн. конф. «Системные проблемы качества, математического моделирования, информационных и электронных технологий». — М., 2003. — С. 23–28.

☎ (0742) 334-90-51

E-mail: irbur27@mail.ru

