2.3. Дискретное программирование

В общем случае задача дискретного программирования имеет следующую постановку [3, 4, 9, 24, 34, 52].

Найти максимум (минимум) целевой функции $f(x_1, x_2, ..., x_n)$ при заданных условиях

$$\left.\begin{array}{l} g_1(x_1, x_2, ..., x_n) \le b_1, \\ g_2(x_1, x_2, ..., x_n) \le b_2, \\ ... \\ g_m(x_1, x_2, ..., x_n) \le b_m, \\ X = \{x_1, x_2, ..., x_n\} \subseteq D, \end{array}\right\} \quad (2.22)$$

где D – конечное или счётное множество.

В этом случае говорят о ***дискретном программировании***. Если X ограничено множеством целых чисел, то задачу назначают задачей ***линейного целочисленного программирования (ЛЦП)***.

Известны следующие классы задач дискретного программирования.

- Задачи с неделимостями (задачи о рюкзаке), обусловлены физическими свойствами объектов. Это задача размещение массивов информации на внешних устройствах ЭВМ при ограничениях на объём, скорость вращения, стоимостные рамки и др.
- Экстремальные комбинаторные задачи (выбора, о назначениях, коммивояжёра, о покрытиях).
- Задачи на несвязных и невыпуклых областях.
- Задачи с разрывной целевой функцией.
- Транспортная задача. Когда задано ограничение целочисленности, то есть при целых значениях массивов поставок, потребления и стоимостей или ограничениях на пропускные способности коммуникаций, является задачей линейного целочисленного программирования.

Для решения задач дискретного программирования применяются как строго обоснованные, так и неформальные метода поиска решений.

1. Метод отсечений, отсекающих плоскостей, он же метод ГОмори (или ГомОри).
2. Метод ветвей и границ.
3. Методы, учитывающие особенности задачи.
4. Методы случайного поиска (эвристические).

2.3.1. Решение задачи ЛЦП методом Гомори

Данный метод носит название метода ***отсекающих плоскостей*** или метода ***целочисленных форм***, но чаще именуется по имени Автора. В основании метода положены следующие теоретические положения [8].

Любое уравнение или неравенство линейной системы ограничений представимо линейной комбинацией базисных векторов и в канонической форме записывается так:

$$\sum_{j=1}^{n} d_{ij} \cdot x_j = p_i, \ i = \overline{1,m}. \quad (2.23)$$

Обозначим заключением в квадратные скобки $[d]$ целую часть d:

[2,5] = 2; [10/3] = 3; [4] = 4; [–2,5] = –3; [–10/3] = –4; [–4] = –4.

По сути дела, операция […] представляет собой округление по недостатку (в меньшую сторону).

Если считать переменные $X = (x_1, x_2, ... , x_n)$ целыми числами, то можно перейти к более слабому, по сравнению с (2.23), выражению

$$\sum_{j=1}^{n} \left[d_{ij}\right] \cdot x_j \le p_i, \ i = \overline{1,m}, \quad (2.24)$$

а учитывая, что сумма в (2.20) целочисленна, то справедливо и

$$\sum_{j=1}^{n} \left[d_{i,j}\right] \cdot x_j \le [p_i], \ i = \overline{1,m}. \quad (2.25)$$

Введя свободную целочисленную переменную x_{n+t}, канонизируем (2.25):

$$\sum_{j=1}^{n} \left[d_{ij}\right] \cdot x_j + x_{n+t} = [p], \ i = \overline{1,m}. \quad (2.26)$$

Очевидно, что добавление последнего равенства к исходной канонизированной системе ***не противоречит*** исходной системе ограничений.

Чтобы получить (2.26) из (2.23), необходимо «отсечь» от (2.26) дробную часть. С этой целью формируется ***отсечение*** для приведения произвольного уравнения в целочисленную форму. Указанное отсечение представляет собою уравнение, которое, будучи прибавленным к исходному уравнению, делает его целочисленным:

$$\sum_{j=1}^{n}\{-d_{ij}\}\cdot x_j + x_{n+t} = \{-p\},\ i=\overline{1,m}. \qquad (2.27)$$

В (2.27) обозначена символами {…} операция нахождения дробной части. Отсюда просматривается простой вычислительный алгоритм.

Алгоритм формирования отсечения Гомори

1. Для выбранного канонизированного уравнения (2.23) сформировать желаемую целочисленную форму вида (2.26).

2. Из целочисленной формы (2.26) вычесть исходное уравнение (2.23), получится уравнение отсекающей плоскости (2.27).

Дадим и нотацию этого алгоритма в виде формулы:

<Отсечение> = <Целочисленная_форма> – <Исходная_строка>. (2.28)

Алгоритм решения задачи ЛЦП методом Гомори

Процедура получения решения структурно состоит из

- предварительного этапа,
- проверки условия окончания и
- так называемой "большой итерации", которая включает операцию формирования отсечения и несколько шагов итеративной части двойственного симплекс-метода.

1. ***Предварительный этап***. В ходе его получают оптимальное решение ЗЛП без учёта целочисленности. Решение выполняется любым удобным методом, кроме, разумеется, графического метода.

2. ***Условие окончания*** расчётов. Если в текущем решении в***се компоненты базисного столбца*** A_0, соответствующие основным переменным, ***являются целыми*** числами, то найдено оптимальное решение задачи ЛЦП. В противном случае, выполняется большая итерация.

3. ***Большая итерация***.

3.1. ***Отсечение Гомори*** формируется

- ***для тех строк*** симплекс-таблицы, ***в которых*** компоненты $a_{i,0}$, соответствующие основным переменным задачи, ***дробные числа***;
- ***на*** каждом ***шаге*** алгоритма отсечение Гомори формируется только ***по одной*** из строк;
- ***очерёдность*** формирования отсечений ***не регламентируется***.

3.2. В базисное решение вводится дополнительная переменная x_{r+t}, соответствующая канонизированному уравнению отсекающей плоскости, одновременно симплекс-таблица пополняется строкой и столбцом-ортом A_{n+t}.

3.3. Выполняется итерационная часть двойственного симплекс-метода. Направляющая строка соответствует вектору A_{n+t}, а направляющий столбец определяется по обычному условию этого метода:

$$\arg\min_{j}\left\{\frac{-\delta_j \ge 0}{a_{i^*,j} < 0}\right\} \Rightarrow j^*.$$

3.4. Если вектор A_{n+t}, ранее выведенный из базиса, в ходе расчётом снова в него вводится в процессе итераций, то строку и столбец симплекс-таблицы, соответствующие переменной x_{n+t} после пересчёта по методу Жордана-Гаусса вычёркивают (удаляют) из неё.

На этом циклическая часть алгоритма завершена, а цикл возобновляется с п. 2.

Замечания к методу Гомори [9].

1. Сходимость вычислений обеспечивается за конечное число итераций, что и обусловливает применение данного метода на практике

2. Метод особенно эффективен, когда ***большинство*** переменных в оптимальном нецелочисленном решении имеют целочисленные значения.

4. После выполнения нескольких больших итераций на шаге отсечения Гомори появляются многочисленные альтернативы. Это ведёт к зацикливанию, именуемому Г. Вагнером [9] "сплошной вырожденностью", когда решение возвращается к ранее бытовавшей позиции, вследствие неверного выбора строки для формирования отсечения. Вагнеру известны многочисленные примеры, когда при значениях n и m, не превышающих десяти, потребовались тысячи итераций, прежде чем оптимум был достигнут.

5. Затруднена сходимость при решении задач, в которых значения элементов $a_{i,j}$ и b_i велики.

6. Иногда, для достижения успеха, требуется видоизменить постановку задачи в сторону усиления, например, введя ограничения $x_1 \le 6$ и $x_2 \le 6$ в дополнение к уже существующему ограничению $x_1 + x_2 \le 6$.

Рассмотрим пример применения метода Гомори.

Поскольку операции с симплекс-таблицами нами ранее вельми подробно рассматривались, ограничим содержание нашего примера процессом формирования отсечений Гомори по результатам решения демонстрационной задачи о производстве изделий из картошки.

Оптимальное решение этой задачи без учёта ограничения целочисленности имеет вид.

		c_j	5	6	0	0	0
Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5
A_2	6	3	0	1	5	–5	0
A_1	5	4,5	1	0	–2,5	7,5	0
A_5	0	0,15	0	0	–0,75	–0,75	1
	δ_j	40,5	0	0	17,5	7,5	0

Видно, что основная переменная x_1 не целая, поэтому необходимы отсечения.

Сформируем отсечение Гомори по второй строке, которая соответсвует основной переменной x_1. Исходной строке соответствует уравнение

$$4{,}5 = 1x_1 + 0x_2 - 2{,}5x_3 + 7{,}5x_4 + 0x_5 .$$

Целочисленная форма для этой строки есть

$$4 = 1x_1 + 0x_2 - 3x_3 + 7x_4 + 0x_5 .$$

Поэтому отсечение будет

$$\begin{array}{l} 4 \;\; = 1x_1 + 0x_2 - \;\; 3x_3 + \;\;\; 7x_4 + 0x_5 \\ - \\ \underline{4{,}5 = 1x_1 + 0x_2 - 2{,}5x_3 + 7{,}5x_4 + 0x_5} \\ -0{,}5 = 0x_1 + 0x_2 - 0{,}5x_3 - 0{,}5x_4 + 0x_5 \end{array} .$$

Результат вычислений занесём в симплекс-таблицу в отдельную строку. Одновременно таблица пополнится дополнительным столбцом для вектора A_6 соответствующим переменной x_6.

			c_j	5	6	0	0	0	0
	Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5	A_6
	A_2	6	3	0	1	5	–5	0	0
	A_1	5	4,5	1	0	–2,5	7,5	0	0
	A_5	0	0,15	0	0	–0,75	–0,75	1	0
←	A_6	0	–0,5	0	0	–0,5	–0,5	0	1
		δ_j	40,5	0	0	17,5	7,5	0	0
							↑		

Далее выполняются несколько итерационных шагов двойственного симплекс-метода:

- выводимая строка определяется отрицательной компонентой столбца A_0;
- вектор, вводимый в базис определяет условие $\min\left\{\frac{-17{,}5}{-0{,}5};\frac{-7{,}5}{-0{,}5}\right\}$, это A_4;
- осуществляется пересчёт симплек-таблицы по методу Жордана-Гаусса до получения условия окончания итераций – положительности компонентов A_0.

2.3.2. Решение задачи ЛЦП методом ветвей и границ [4, 34, 40, 52]

Этот метод применяется для решения как полностью целочисленных, так и частично целочисленных задач дискретного программирования.

Пусть математическая модель имеет вид

$$\begin{aligned} &C^T X \to \max, \\ &AX \le B. \end{aligned} \tag{2.29}$$

Компоненты вектора X положительны и целочисленны. Допустим, что исходная задача линейного программирования имеет решение. В этом случае область ограничений замкнута.

Тогда каждая переменная x_j и в допустимом решении, и оптимуме ограничена диапазоном

$$L_j \le x_j \le U_j, \tag{2.30}$$

где L_j – нижний предел, а U_j – верхний предел (граница), которые определяются границами области допустимых решений задачи. Это следует из самого факта наличия непротиворечивых ограничений, образующих замкнутую область

Пусть I есть целое число, такое, что $L_j \le I \le U_j - 1$. Тогда оптимальное ***целочисленное*** значение x_j, удовлетворяющее решению (2.29) и лежащее в пределах (2.30), будет находиться либо между L_j и I, либо между $I + 1$ и U_j. Это приводит к тому, что возникают дополнительные условия, по отношению к исходным условиям (2.29), не противоречащие им:

$$\left.\begin{array}{l} x_j \le I, \\ x_j \ge I+1 \end{array}\right\}. \qquad (2.31)$$

На базе ограничений (2.31) основана систематическая схема применения метода.

Ограничения, приписываемые к исходной задаче, есть ***дополнительные границы***, благодаря чему мы имеем, на каждом шаге постановку ***пары задач*** на базе одной нецелочисленной.

Интерпретация хода решения в виде дерева определило второе название метода – ***ветвей.***

Алгоритм метода ветвей и границ

Композиционно алгоритм состоит из предварительного этапа, проверки условия целочисленности текущего решения, построения задач G_{i1} и G_{i2}, большой итерации, которая представляет собой несколько итерационных шагов двойственного симплекс-метода, и заключительной части, на которой выбирается наилучшее из всех, ранее полученных, целочисленных решений. Цифровой код i в индексации задач соответствует положению текущей "родительской" задачи на дереве решений

1. Предварительный этап. Задача (2.29) решается любым удобным методом до отыскания нецелочисленного оптимального решения, соответствующего точке X_0.

2. Этой точке X_0 ставится в соответствие решение G_0 и его оценка – текущее значение целевой функции $\xi = C^T \times X_0$. Если X_0 – целочисленное решение для основных переменных математической модели, то задача считается решённой.

В противном случае, если X_0 – нецелочисленное решение, то, используя систему неравенств (2.31), получаем множество из двух задач G_{01} и G_{02} (ветвей). Особо подчеркнём, что пара задач ***возникает для одной нецелочисленной переменной одновременно***. Каждая задача решается в отдельности, при этом находят их оценки $\xi(G_{01})$ и $\xi(G_{02})$.

3. В ходе решения на k–й итерации, в зависимости от текущих оценок $\xi(G_{i1})$ и $\xi(G_{i2})$, может произойти дальнейшее ветвление задач.

4. Вычислительный процесс осуществляется до "перерешивания" всех возникающих задач или до получения признаков их неразрешимость. Из полученных решений выбирается то, которое является наилучшим (в смысле оптимума) решением исходной задачи (2.29).

5. Для решения возникающих задач (2.31) используют двойственный симплекс-метод, который, как нам известно, допускает ввод новых ограничений в псевдоплан по ходу решения.

6. Ограничения вводятся только для одной из основных базисных нецелочисленных переменых. Правила формирования ограничений по неравенствам (2.31) суть следующие. Пусть в базисе находится вектор A_j, соответствующая переменная которого x_j – дробное число.

- Задача G_{i1} формируется по ограничению $x_j \le I$, где I – целая часть $[x_j]$, округленная по недостатку. Первоначально формируется ограничение A_{n+t}, которое соответствует канонической форме неравенства и представляется в виде уравнения $I = x_j + x_{n+t}$. В симплекс-таблицу помещается строка, которая получается в результате операции вычитания $A_{n+t} — A_j$.
- Задача G_{i2} формируется по ограничению $– x_j \le – (I + 1)$, которой соответствует каноническая форма $– (I + 1) = – x_j + x_{n+t}$. В симплекс таблицу помещается строка, равная сумме $A_{n+t} + A_j$.

Каждая из исходных симплекс-таблиц задач G_{i1} и G_{i2}, дополняется строкой симплекс-разностей, взятой из таблицы, содержащей нецелочисленное решение G_i.

Продемонстрируем работу алгоритма на известном примере.

Оптимальное решение без учёта целочисленности есть

		c_j	5	6	0	0	0
Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5
A_2	6	3	0	1	5	–5	0
A_1	5	4,5	1	0	–2,5	7,5	0
A_5	0	0,15	0	0	–0,75	–0,75	1
		δ_j 40,5	0	0	17,5	7,5	0

а его оценка $G_0[40,5] = 40$. Обе задачи формируются для переменной x_1 по 2-й строке таблицы A_2.

Задача G_{01}.

$$x_1 \le 4 \Rightarrow A_6\text{: } 4 = x_1 + x_6,\ \widetilde{A}_6 : A_6 – A_1 .$$

$$\begin{array}{llcllllll}
A_6 & 4 & = & 1x_1 & +\ 0x_2 & -\ 0x_3 & +\ 0x_4 & +\ 0x_5 & +\ 1x_6 \\
- & & & & & & & & \\
A_1 & 4{,}5 & = & 1x_1 & +\ 0x_2 & -\ 2{,}5x_3 & +\ 7{,}5x_4 & +\ 0x_5 & \ \ 0x_6 \\
\hline
\widetilde{A}_6 & -0{,}5 & = & 0x_1 & +\ 0x_2 & +\ 2{,}5x_3 & -\ 7{,}5x_4 & +\ 0x_5 & +\ 1x_6
\end{array}$$

Задача G_{02}.

$-x_1 \le -5 \Rightarrow A_6' : -5 = -x_1 + x_6'$, $\widetilde{A}_6' : A_6' + A_1$

$$
\begin{array}{llcccccccccccccc}
A_6' & -5 & = & -1x_1 & + & 0x_2 & - & 0x_3 & + & 0x_4 & + & 0x_5 & + & 1x_6' \\
- & & & & & & & & & & & & & \\
A_1 & 4{,}5 & = & 1x_1 & + & 0x_2 & - & 2{,}5x_3 & + & 7{,}5x_4 & + & 0x_5 & & 0x_6 \\
\hline
\widetilde{A}_6' & -0{,}5 & = & 0x_1 & + & 0x_2 & - & 2{,}5x_3 & + & 7{,}5x_4 & + & 0x_5 & + & 1x_6'
\end{array}
$$

Сформируем симплекс-таблицы для обеих задач.

Задача G_{01}.

			c_j	5	6	0	0	0	0
	Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5	A_6
	A_2	6	3	0	1	5	–5	0	0
	A_1	5	4,5	1	0	–2,5	7,5	0	0
	A_5	0	0,15	0	0	–0,75	–0,75	1	0
←	$\widetilde{A}_6$	0	-0,5	0	0	2,5	–0,75	0	1
		δ_j	40,5	0	0	17,5	7,5	0	0
							↑		

Задача G_{02}.

			c_j	5	6	0	0	0	0
	Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5	A_6'
	A_2	6	3	0	1	5	–5	0	0
	A_1	5	4,5	1	0	–2,5	7,5	0	0
	A_5	0	0,15	0	0	–0,75	–0,75	1	0
←	$\widetilde{A}_6'$	0	-0,5	0	0	–2,5	–0,75	0	1
		δ_j	40,5	0	0	17,5	7,5	0	0
						↑			

Вид дерева решений показан ниже, на рисунке 2.5.

$X^T(4{,}5;\ 3;0;\ 0;\ 0{,}15)$, $\xi(G_0) = 40{,}5$

$X^T = ?$, $\xi(G_{01}) = ?$ $\qquad$ $X^T = ?$, $\xi(G_{02}) = ?$

Рисунок 2.6 – Первоначальное дерево решений

Решение опускаем как нами освоенное в предыдущих разделах.

Метод ещё называют ***методом обрыва ветвей*** или ***методом возврата***: всё зависит от способа перемещения по дереву задач. Существует множество алгоритмов метода, адаптированных под разнообразные частные условия содержательной задачи.

2.3.3. Вопросы для самоконтроля

1. В чем сходства и различия терминов “дискретный” и “целочисленный”?
2. Как построить отсекающую плоскость Гомори?
3. Почему алгоритм ветвей и границ получил такое название, что является ветвями, а что — границами?
4. В чём идея сущность и неравенств (2.31)?
5. Почему в ходе решения ЛЦП используется двойственный симплекс-метод?
6. В каких случаях задача ЛЦП не будет иметь решения?
7. Как вы думаете, оптимальное решение ЛЦП будет единственным? Обоснуйте свои соображения по этому поводу.