Уральский федеральный университет имени первого Президента России Б.Н.Ельцина

Министерство образования и науки Российской Федерации.
Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Уральский федеральный университет имени первого Президента России Б.Н.Ельцина» (УрФУ)

Уральский федеральный университет

Г.М. Черногородова

Динамическое программирование

Методические указания к домашней работе по дисциплине «Методы оптимизации» для студентов дневной формы обучения по направлению 09.04.01 «Информатика и вычислительная техника»

Екатеринбург

2015

1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ

1.1. Постановка задачи о загрузке самолета

Решить методом динамического программирования целочисленную задачу нелинейного программирования на примере задачи о загрузке самолёта. Самолет может быть заполнен различными по весу и ценности грузами.

Введём параметры задачи. Обозначим:

j - тип груза, $j = \overline{(1,4)}$,

x_j - количество груза j -го типа,

c_j - стоимость единицы груза j -го типа,

a_j - вес единицы груза j -го типа,

b - общая грузоподъемность самолета.

Математическая модель задачи

Пусть целевая функция имеет вид:

$$Z = c_2 x_2^2 + \sum_{j=1,3,4} c_j x_j \to \max$$

при ограничениях:

$$\sum_{j=1}^{4} a_j x_j \le b, \quad x \ge 0, \quad x_1 \ge 2, \quad x_2 \le 3, \quad x \text{ - целое.}$$

Необходимо выбрать такой набор грузов x, при котором общая ценность груза Z будет максимальной.

Модель задачи принадлежит к классу целочисленной задачи нелинейного программирования. Рассмотрим особенности модели задачи.

Отметим, что целевая функция $f(x)$ является аддитивной. Задача допускает интерпретацию как n- шаговый процесс принятия решения. На каждом шаге загружаем груз одного типа. Структура задачи не зависит от числа шагов. В качестве текущего состояния системы на j-м шаге примем грузоподъёмность самолёта перед загрузкой груза j -го типа. Решение, принимаемое на j -м шаге, это выбор x_j - количества груза j -го типа, которое следует загрузить на борт самолёта.

Условия задачи позволяют применить для её решения метод динамического программирования, основанный на принципе оптимальности Беллмана, который утверждает, что решения $x_{k+1}^0,...,x_n^0$ должны быть оптимальными по отношению к состоянию, получающемуся в результате применения предыдущих решений $x_1^0,...,x_k^0$. Иначе говоря, оптимальное управление системой на каждом шаге не зависит от того, как система пришла в текущее состояние, а определяется только самим этим состоянием.

Изложение теории по динамическому программированию представлено: модуль Лекция 6 в ЭУМК; учеб. пособия из списка литературы раздела «Справочный материал» данного модуля.

Варианты заданий (параметры модели) приведены в табл. 1.

Таблица 1

№ Вар.	***b***	a_j				c_j			
1	19	1	2	2	3	1	2	2,8	3
2	24	1	3	2	1	2	7	4,5	1
3	23	2	2	1	3	3	4	1,5	2
4	21	1	3	2	4	2	7	2	8,6
5	19	4	1	2	2	3	2	3	1,8
6	23	3	1	1	2	2	3	2,7	1
7	27	1	2	1	2	1	8	3,7	2
8	22	2	3	1	2	2	6	1,5	3
9	19	1	2	3	4	1	3	4	1,5
10	21	4	1	2	3	4	3	3	1,6
11	24	4	3	1	1	1	7	2	1,3
12	19	2	1	2	3	3	2	4	1,7
13	30	3	1	1	2	1	3	2,5	3,7
14	29	3	2	4	1	3	3	2	1,5
15	24	2	1	2	3	2	2	1	3,5

Справочный материал

Метод динамического программирования

Динамическое программирование – это вычислительный метод для решения задач определенной структуры, в модели которых предполагается, что целевая функция $f(x)$ является аддитивной, если $f(x)=\sum_{i=1}^{n} f_i(x_i)$, или мультипликативной, если $f(x)=\prod_{i=1}^{n} f_i(x_i)$.

Пусть задан многошаговый процесс принятия решений

$p_i \to \omega(p_i, u_i) \to p_{i+1}$,

где p_i - состояние системы на i-м шаге,

u_i - решение, принимаемое на i-м шаге,

$\omega(p_i, u_i)$ - функция эффективности принятого решения на i-м шаге.

Пусть p_1 - начальное состояние и p_{n+1} - конечное состояние системы, тогда процесс можно представить в виде

$$P_1 \xrightarrow[\omega(p_1,u_1)]{U_1} P_2 \xrightarrow[\omega(p_2,u_2)]{U_2} P_3 \to \cdots \to P_n \xrightarrow[\omega(p_n,u_n)]{U_n} P_{n+1}.$$

Введем некоторые понятия. Совокупность принятых решений на n шагах $S_n=(u_1,...u_n)$ назовем ***стратегией.***

$C=\sum_{i=1}^{n}\omega(p_i,u_i)=C(p_1,S_n)$ назовем ***суммарной эффективностью***,

$S_n^0 : C(p_1,S_n^0)=\max_{S_n} C(p_1,S_n)$ назовем ***оптимальной стратегией***, при которой суммарная эффективность максимальна.

Принцип оптимальности Беллмана

Вычислим суммарную эффективность при оптимальной стратегии.

$$C(p_1,S_n^0)= f_n(p_1)=\max_{S_n=(u_1,\dots,u_n)}\sum_{i=1}^{n}\omega(p_i,u_i)=\max_{u_1,\dots,u_k}\left(\max_{u_{k+1},\dots,u_n}\left(\sum_{i=1}^{k}\omega(p_i,u_i)+\sum_{i=k+1}^{n}\omega(p_i,u_i)\right)\right)=$$

$$= \max_{u_1,\ldots,u_k}\left(\sum_{i=1}^{k}\omega(p_i,u_i)+\max_{u_{k+1},\ldots,u_n}\sum_{i=k+1}^{n}\omega(p_i,u_i)\right).$$

Учитывая, что второе слагаемое в скобках равно $f_{n-k}(p_{k+1})$, получаем

$$f_n(p_1)=\max_{u_1,\ldots,u_k}\left(\sum_{i=1}^{k}\omega(p_i,u_i)+f_{n-k}(p_{k+1})\right), \qquad (1)$$

т.е. одна сложная задача сведена к двум задачам меньшей размерности.

При $k=1$ получим ***уравнение Беллмана***:

$$\boldsymbol{f_n(p_1)=\max_{u_1}(\omega(p_1,u_1)+f_{n-1}(p_2))}. \qquad (2)$$

Уравнение (2) называется ***основной рекуррентной формулой*** для метода динамического программирования. Рассмотрим оптимальную стратегию $S_n^0=(u_1^0,\ldots,u_n^0)$.

По определению имеем

$$f_n(p_1)=\sum_{i=1}^{n}\omega(p_i^0,u_i^0)=\sum_{i=1}^{k}\omega(p_i^0,u_i^0)+f_{n-k}(p_{k+1}^0).$$

Отсюда получаем

$$f_{n-k}(p_{k+1}^0)=\sum_{i=k+1}^{n}\omega(p_i^0,u_i^0), \qquad (3)$$

т.е. принцип оптимальности Беллмана утверждает, что решения $u_{k+1}^0,\ldots,u_n^0$ должны быть оптимальными по отношению к состоянию, получающемуся в результате применения предыдущих решений $u_1^0,\ldots,u_k^0$. Иначе говоря, оптимальное управление системой на каждом шаге не зависит от предыстории процесса, т.е. от того, как система пришла в текущее состояние, а определяется только самим этим состоянием.

Сущность динамического подхода состоит в замене решения данной n- шаговой задачи последовательностью задач: одношаговой, двухшаговой и т.д.

Отметим основные свойства задач, к которым применим данный подход.

- Задача должна допускать интерпретацию как n-шаговый процесс принятия решения.

- Задача должна иметь структуру, не зависящую от числа шагов (числом шагов можно регулировать точность вычислений).
- Формулировка задачи предполагает, что состояние системы должно описываться параметрически.
- Выбор управления на k-м шаге не должен оказывать влияния на предыдущие решения, кроме необходимого пересчета переменных.

Список литературы

1. Аттетков А.В. Методы оптимизации: учеб. для вузов / А.В. Аттетков, С.В. Галкин, В.С. Зарубин. М.: Изд-во МГТУ им. Н.Э. Баумана, 2003. 440 с.
2. Васильев Ф.П. Методы оптимизации. Книга 1. Учебник для вузов. М.: МЦМНО, **2011**.- 620 с.
3. Катулев А.Н., Северцев Н.А. Математические методы в системах поддержки принятия решений. М: Высшая школа. 2005. 311 с.
4. Сухарев А.Г. Курс методов оптимизации: учебное пособие / А.Г. Сухарев, А.В. Тимохов, В.В. Фёдоров. М.: Физматлит, 2005. 368 с.
5. Черногородова Г.М. Теория принятия решений: учебное пособие/ Г.М. Черногородова. Екатеринбург: УГТУ - УПИ, 2006. 183 с.