Экономико-математические методы и модели

Учебные вопросы

- Основы динамического программирования:
  – Решение задачи «О рюкзаке» методом динамического программирования

Имеется рюкзак с заданной вместимостью (под вместимостью понимается максимально возможная масса), и имеются предметы (n штук), причем каждый предмет характеризуется массой w и ценностью P.
w = {w1, w2, …, wn}
p={p1, p2, …, pn}
Требуется собрать рюкзак с максимальной ценностью и минимальным возможным весом, не превышающим Wmax.
1 способ: перебор (простой)
2 способ: метод ветвей и границ, который заключается в умном переборе. Могут быть случаи, когда перебираются все возможные варианты.
3 способ: использование «жадного» алгоритма (берется каждый текущий момент («лучший» элемент), ориентированный на их относительной точности). Решение будет получено достаточно быстро, но не факт, что оно будет оптимальным.

Математическая формулировка задачи

Имеется рюкзак с целочисленным значением «весова» W. Имеется n предметов, характеризующихся целочисленными показателями весов wi и ценностей pi. Требуется построить вектор бинарных величин B = {b1, b2, …, bn} (0 – не положили в рюкзак, 1– положили) так, чтобы при выполнении ограничения b1w + b2w2 + … + bnwn = ( )=<W b1p1 + b2p2 + … + bnpn = F(набора предметов) □max Вход: Wmax, bi, pi Выход: вес рюкзака, который получится W, pi, B и номера предметов.

Использование метода динамического программирования

Главной идеей метода динамического программирования является сохранение результата, достигнутого на предыдущих этапах, то есть, каждый раз, решая задачу о необходимости загрузить рассматриваемый предмет в рюкзак, пытаемся решить задачу, анализируя те результаты, которые были достигнуты ранее, то есть до того как мы начали рассматривать текущий k-й предмет, а именно, основываясь на том как был упакован рюкзак предметами с номерами 1,2, …,k-1, причем, необходимо еще учитывать минимальность веса, то есть рассматривать возможность веса рюкзака от 0 до w.

Метод решения

Для хранения результата предыдущих вычислений будем хранить все значения в
матрице **A(k,s).**
Все величины целочисленные.
k – номер текущего предмета, который может быть положен в рюкзак;
s- текущий рассматриваемый вес рюкзака.

Учитывая исходные данные, матрица будет (5х15). Элемент матрицы А будет иметь смысл максимальной возможной стоимости при весе меньшем или равном **s** в случае возможности разместить в рюкзаке **k**-первых предметов. Для удобства расчетов будем рассматривать нулевой столбец и нулевую строку, которые полностью заполнены нулями.

A(0,i) = 0; A(j,0) = 0; для любых i=0,15; j=0,5

Пример 1: **N=5, W max=15**

k	1	2	3	4	5
w	6	4	3	2	5
p	5	3	1	3	6

Расчетная формула:

$$A[k,s] \quad = \max(A[k-1,s], A[k-1, s - w[k]] + p[k])$$

Будем заполнять матрицу по строкам, то есть каждая строка соответствует анализу k-го предмета.

$$A[4,10] = \max(A[3,10], A[3,8] + 3) = \max (8,11)$$

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	5	5	5	5	5	5	5	5	5	5
2	0	0	0	0	3	3	5	5	5	5	8	8	8	8	8	8
3	0	0	0	1	3	3	5	5	5	6	8	8	8	9	9	9
4	0	0	3	3	3	4	6	6	8	8	8	9	11	11	11	12
5	0	0	3	3	3	6	6	9	9	9	10	12	12	14	14	14

Пример 2:
Wmax = 21
n = 6

k	1	2	3	4	5	6
w	3	5	7	2	4	8
p	2	6	5	3	5	7

k\s	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
1	**0**	**0**	**0**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
2	**0**	**0**	**0**	**2**	**2**	**6**	**6**	**6**	**6**	**6**	**8**	**8**	**8**	**8**	**8**	**8**	**8**	**8**	**8**	**8**	**8**	**8**
3	**0**	**0**	**0**	**2**	**2**	**6**	**6**	**6**	**8**	**8**	**8**	**8**	**1**	**1**	**1**	**13**	**13**	**13**	**13**	**13**	**13**	**13**
4	**0**	**0**	**3**	**3**	**3**	**6**	**6**	**9**	**9**	**9**	**11**	**11**	**11**	**11**	**14**	**14**	**14**	**16**	**16**	**16**	**16**	**16**
5	**0**	**0**	**3**	**3**	**5**	**6**	**8**	**9**	**9**	**11**	**11**	**14**	**14**	**14**	**16**	**16**	**16**	**16**	**19**	**19**	**19**	**21**
6	**0**	**0**	**3**	**3**	**5**	**6**	**8**	**9**	**9**	**11**	**11**	**14**	**14**	**14**	**16**	**16**	**16**	**18**	**19**	**21**	**21**	**21**