Задача линейного раскроя

В случае, когда параметры L_1, l_j – целые числа, j=1,..n, метод динамического программирования можно применить для решения задачи линейного раскроя.

Введем обозначения:

$$\varphi(y) = max\{\sum_{j=1}^{n} c_j x_j \mid \sum_{j=1}^{n} l_j x_j \leq y, x_j \geq 0, \text{целое}, j = 1,..,n\}$$

То есть, $\varphi(y)$- стоимость оптимального раскроя стрежня длиной y. Нас интересует $\varphi(L)$.

Покажем, как можно вычислить значения $\varphi(y)$ при известных $\varphi(y'), y' < y$.

Пусть l_0 равно минимальной длине детали:

$$l_0 = min\ l_j.$$

Очевидно, что $\varphi(y) = 0, y = 0,..,l_0 - 1$, так как стержни с длинами, меньшими длин любой из деталей, раскроить не удастся.

Для вычисления значений $\varphi(y)$ при $y \geq l_0$ можно воспользоваться следующим рекуррентным соотношением Беллмана:

$$\varphi(y) = max\{\varphi(y - l_j) + c_j | l_j \leq y\}, y = l_0,..,L.$$

Решение примера:

Дано: L = 23, n = 3, $l_1 = 3$, $l_2 = 7$, $l_3 = 12$, $c_1 = 5$, $c_2 = 12$, $c_3 = 16$.

Так как наименьшая длина детали = 3, то

$$\varphi[0] = \varphi[1] = \varphi[2] = 0$$
$$\psi[0] = \psi[1] = \psi[2] = 0$$

Рассмотрим для y=3..L. Для вычислений была написана программа на языке c++.

y	φ	ψ
3	5	1
4	5	1
5	5	1
6	10	1
7	12	2
8	12	2

9	15	1
10	17	1
11	17	1
12	20	1
13	22	1
14	24	2
15	25	1
16	27	1
17	29	1
18	30	1
19	32	1
20	34	1
21	36	2
22	37	1
23	39	1

Результаты вычислений:

Максимальная стоимость оптимального раскроя стержня $\varphi[23]$=39, так как $\psi[23] = 1$, то отрезаем от стержня деталь номер 1, получаем остаток, равный 20. $\psi[20] = 1$, то отрезаем от стержня деталь номер 1, получаем остаток, равный 17. $\psi[17] = 1$, то отрезаем от стержня деталь номер 1, получаем остаток, равный 14. $\psi[14] = 2$, то отрезаем от стержня деталь номер 2, получаем остаток, равный 7. $\psi[7] = 2$, то отрезаем от стержня деталь номер 2, получаем остаток, равный 0. Остатка стержня не остаётся. Первая деталь отрезалась 3 раза, а вторая 2 раза, третья ни разу, поэтому $x_1 = 3, x_2 = 2, x_3 = 0$.

Жадный алгоритм:

Пусть выполнено:

$$\frac{c_1}{l_1} \geq \frac{c_2}{l_2} \geq \ldots \geq \frac{c_n}{l_n}$$

То есть все детали упорядочены по стоимости единицы длины.

В качестве эвристики применим следующий жадный алгоритм. Раскроим стержень на максимальное количество деталей первого вида:

$$x_1 = [\frac{L}{l_1}], y_1 = L \bmod l_1$$

Где y1 – длина остатка материала. Остаток можно раскроить на детали другого вида.

$$x_j = [\frac{y_j - 1}{l_j}], y_j = y_{j-1} \bmod l_j, j = 2, ..., n$$

Решение примера:

Дано: L = 23, n = 3, $l_1 = 3$, $l_2 = 5$, $l_3 = 12$, $c_1 = 7$, $c_2 = 12$, $c_3 = 16$.

Для решения данного примера была также написана программа на языке с++. После вычислений программы были получены следующие результаты:

```
После упорядочивания по стоимости единицы длины:
Для элемента 0 стоимость единицы длины равна = 1.71429, где длина элемента = 7
Для элемента 1 стоимость единицы длины равна = 1.66667, где длина элемента = 3
Для элемента 2 стоимость единицы длины равна = 1.33333, где длина элемента = 12
Итоговое значение phi для стоимости: 36

Количество деталей по типу:
x1=3
x2=0
x3=0
```

Итоговое значение $\varphi[23]$=36, что на 3 меньше, чем у оптимального раскроя. После сортировки по стоимости единицы длины x_1 стал x_2, а x_3. Значит второй элемент самый эффективный, потом по эффективности идёт первый элемент и самый неэффективный третий. Значит следуя алгоритму отрежем все детали длины 7, их получается 3, остаток стержня 2, и на другие элементы нам не хватит стержня. Максимальная стоимость оптимального раскроя стержня жадным алгоритмом 36.

Первая деталь отрезалась 0 раз, вторая 3 раза, третья ни разу, поэтому $x_1 = 0, x_2 = 3, x_3 = 0$.

Результат вычисления:

```
------------------------------Линейный раскрой-------------------
Y=0
phi[0] = 0
psi[0] = 0

Y=1
phi[1] = 0
psi[1] = 0

Y=2
phi[2] = 0
psi[2] = 0

Y=3
phi[3] = 5
psi[3] = 1

Y=4
phi[4] = 5
psi[4] = 1

Y=5
phi[5] = 5
psi[5] = 1

Y=6
phi[6] = 10
psi[6] = 1

Y=7
phi[7] = 12
psi[7] = 2

Y=8
phi[8] = 12
psi[8] = 2

Y=9
phi[9] = 15
psi[9] = 1

Y=10
phi[10] = 17
psi[10] = 1

Y=11
phi[11] = 17
psi[11] = 1

Y=12
phi[12] = 20
psi[12] = 1

Y=13
phi[13] = 22
psi[13] = 1

Y=14
phi[14] = 24
psi[14] = 2

Y=15
phi[15] = 25
psi[15] = 1

Y=16
phi[16] = 27
psi[16] = 1

Y=17
phi[17] = 29
psi[17] = 1

Y=18
phi[18] = 30
psi[18] = 1

Y=19
phi[19] = 32
psi[19] = 1

Y=20
phi[20] = 34
psi[20] = 1

Y=21
phi[21] = 36
psi[21] = 2

Y=22
phi[22] = 37
psi[22] = 1

Y=23
phi[23] = 39
psi[23] = 1

Количество деталей по типу:
x1=3
x2=2
x3=0
```

```
------------------------------Жадный алгоритм------------------------------
После упорядочивания по стоимости единицы длины:
Для элемента 1 стоимость единицы длины равна = 1.71429, где длина элемента = 7
Для элемента 2 стоимость единицы длины равна = 1.66667, где длина элемента = 3
Для элемента 3 стоимость единицы длины равна = 1.33333, где длина элемента = 12
Итоговое значение phi для стоимости: 36

Количество деталей по типу:
x1=3
x2=0
x3=0
```

Вывод

Результаты, полученные методом динамического программирования, показали более оптимальные результаты чем результаты жадного алгоритма.