3. Задача оптимального раскроя

Постановка задачи

Выпускаемая в 3-м цехе продукция, представляющая собой полуфабрикат определенного типоразмера постоянного сечения и длиной 450 см, разрезается на заготовки длиной 450 см, 230 см, 90 см в комплектности, определяемой соотношением 2:3:4.

Требуется решить задачу оптимального раскроя в двух постановках и провести ее исследование:

1. спланировать раскрой полуфабриката, при котором число комплектов заготовок будет наибольшим;
2. спланировать раскрой полуфабриката при условии минимизации остатков и сравнить полученные результаты;
3. средствами параметрического исследования правых частей выяснить необходимое приращение количества поступивших полуфабрикатов для увеличения числа комплектов заготовок на 1 (или на 10), причем провести указанное исследование для разных значений исходного количества полуфабрикатов (проверка линейности).

Формализация задачи 1

Переменные

Пусть n — число комплектов заготовок. Прежде чем строить математическую модель, укажем все осмысленные варианты разреза полуфабриката на заготовки (табл. 1).

Таблица 1. Варианты раскроя

Длина заготовки, см	Вариант разреза заготовки			Количество заготовок, шт.
	1	2	3	
450	1	0	0	$2n$
230	0	1	0	$3n$
90	0	2	5	$4n$

Теперь мы можем ввести переменные следующим образом: x_j — количество полуфабрикатов, разрезанных по варианту j, $j = 1, 2, 3$.

Отметим, что в этой постановке задачи про потери материала мы не думаем.

Математическая модель

$$n \rightarrow max$$

$$x_1 = 2n \quad , (1)$$

$$x_2 = 3n \quad , (2)$$

$$2x_2 + 5x_3 \geq 4n \quad , (3)$$

$$n \geq 0, x_j \geq 0, j = 1, 2, 3 \quad . (4)$$

Очевидно, что третий вариант раскроя никогда не будет использоваться, поэтому ограничение (3) вырождается в $6n \geq 4n$ и может быть вычеркнуто.

Решение задачи 1

Подготовим математическую модель для решения задачи с помощью пакета CVXOPT (табл. 2).

Таблица 2. Подготовка матриц

	x1	x2	x3	n	Знак	Пр. часть
c	0	0	0	1	-	max **(-1)**
y1	1	0	0	-2	=	0
y2	0	1	0	-3	=	0
y3	1	0	0	0	>= **(-1)**	0
y4	0	1	0	0	>= **(-1)**	0
y5	0	0	1	0	>= **(-1)**	0
y6	0	0	0	1	>= **(-1)**	0
y7	1	1	1	0	<=	100

Ограничение y7 устанавливает верхнюю границу числа полуфабрикатов.

```python
from cvxopt import matrix, solvers

c = matrix([0, 0, 0, -1], tc='d')

G = matrix([[-1,  0,  0,  0],
            [ 0, -1,  0,  0],
            [ 0,  0, -1,  0],
            [ 0,  0,  0, -1],
            [ 1,  1,  1,  0]], tc='d')

h = matrix([0, 0, 0, 0, 100], tc='d')

A = matrix([[ 1,  0,  0, -2],
            [ 0,  1,  0, -3]], tc='d')

b = matrix([0, 0], tc='d')

solution = solvers.lp(c, G.T, h, A.T, b, solver='glpk')
print('Status', solution['status'])
print('x = \n', solution['x'])
print('Objective:', -solution['primal objective'])
```

```
Status optimal
x =
 [ 4.00e+01]
[ 6.00e+01]
[ 0.00e+00]
[ 2.00e+01]

Objective: 20.0
```

Рис. 1. Решение первой задачи

На рис. 1 приведено решение первой задачи. Результат: разрезав 40 полуфабрикатов первым способом и 60 — вторым, получим 20 комплектов.

Теперь рассмотрим эту же задачу с точки зрения минимизации потерь материала.

Формализация задачи 2

Переменные

В этой постановке задачи мы будем минимизировать потери, которые связаны не только с обрезками материала, но и с избыточным числом заготовок.

Добавим строку потерь материала в таблицу вариантов раскроя (табл. 3).

Таблица 3. Раскрой с учетом потерь

Длина заготовки, см	Вариант разреза заготовки			Количество заготовок, шт.
	1	2	3	
450	1	0	0	2*n*
230	0	1	0	3*n*
90	0	2	5	4*n*
Потери, см	0	40	0	

Введем переменные: x_j — количество полуфабрикатов, разрезанных по варианту j, j = 1, 2, 3; y_i — избыточное число заготовок типа i, i = 1, 2, 3.

Математическая модель

$$40x_2 + 450y_1 + 230y_2 + 90y_3 \rightarrow min$$

$$x_1 - y_1 = 40 \quad , (1)$$

$$x_2 - y_2 = 60 \quad , (2)$$

$$2x_2 + 5x_3 - y_3 = 80 \quad , (3)$$

$$x_j \geq 0, j = 1,2,3; y_i \geq 0, i = 1,2,3 \quad . (4)$$

Отметим, что правые части ограничений 1, 2, 3 такие, потому что мы хотим получить 20 комплектов заготовок. Так мы сможем сравнить решения задачи в первой и второй постановках.

Решение задачи 2

Подготовим математическую модель для решения задачи с помощью пакета CVXOPT (табл. 4).

Таблица 4. Подготовка матриц

	x1	x2	x3	y1	y2	y3	Знак	Пр. часть
c	0	40	0	450	230	90	-	min
y1	1	0	0	-1	0	0	=	40
y2	0	1	0	0	-1	0	=	60
y3	0	2	5	0	0	-1	=	80
y4	1	0	0	0	0	0	>= **(-1)**	0
y5	0	1	0	0	0	0	>= **(-1)**	0
y6	0	0	1	0	0	0	>= **(-1)**	0
y7	0	0	0	1	0	0	>= **(-1)**	0
y8	0	0	0	0	1	0	>= **(-1)**	0
y9	0	0	0	0	0	1	>= **(-1)**	0

```python
from cvxopt import matrix, solvers

c = matrix([0, 40, 0, 450, 230, 90], tc='d')

G = matrix([[-1,  0,  0,  0,  0,  0],
            [ 0, -1,  0,  0,  0,  0],
            [ 0,  0, -1,  0,  0,  0],
            [ 0,  0,  0, -1,  0,  0],
            [ 0,  0,  0,  0, -1,  0],
            [ 0,  0,  0,  0,  0, -1]], tc='d')

h = matrix([0, 0, 0, 0, 0, 0], tc='d')
A = matrix([[ 1,  0,  0, -1,  0,  0],
            [ 0,  1,  0,  0, -1,  0],
            [ 0,  2,  5,  0,  0, -1]], tc='d')

b = matrix([40, 60, 80], tc='d')
solution = solvers.lp(c, G.T, h, A.T, b, solver='glpk')
print('Status', solution['status'])
print('x = \n', solution['x'])
print('Objective:', solution['primal objective'])
```

```
Status optimal
x = 
 [ 4.00e+01]
[ 6.00e+01]
[ 0.00e+00]
[ 0.00e+00]
[ 0.00e+00]
[ 4.00e+01]

Objective: 6000.0
```

Рис. 2. Решение второй задачи

На рис. 2 приведено решение второй задачи. Результат тот же, только теперь мы видим общие потери материала, которые составляют $40 \cdot 60 + 40 \cdot 90 = 6000$ см.

Параметрическое исследование правых частей

Посмотрим на вектор теневых цен первой задачи (рис. 3).

```
print('z = \n', solution['z'])
```

```
z = 
 [-0.00e+00]
[-0.00e+00]
[ 2.00e-01]
[-0.00e+00]
[ 2.00e-01]
```

Рис. 3. Теневые цены

Видим, что, увеличив правую часть ограничения y7 на единицу, значение целевой функции вырастет на 0.2. Следовательно, чтобы произвести дополнительный комплект заготовок, мы должны увеличить число полуфабрикатов на 5. Убедимся, что это так (рис. 4).

```
h = matrix([0, 0, 0, 0, 105], tc='d')

A = matrix([[ 1,  0,  0, -2],
            [ 0,  1,  0, -3]], tc='d')

b = matrix([0, 0], tc='d')

solution = solvers.lp(c, G.T, h, A.T, b, solver='glpk')
print('Status', solution['status'])
print('x = \n', solution['x'])
print('Objective:', -solution['primal objective'])
```

```
Status optimal
x = 
 [ 4.20e+01]
[ 6.30e+01]
[ 0.00e+00]
[ 2.10e+01]

Objective: 21.0
```

Рис. 4. Увеличение числа комплектов на единицу (задача 1)

Убедимся, что задача во второй постановке дает то тоже решение, задав число комплектов заготовок равным 21 (рис. 5).

```python
b = matrix([42, 63, 84], tc='d')
solution = solvers.lp(c, G.T, h, A.T, b, solver='glpk')
print('Status', solution['status'])
print('x = \n', solution['x'])
print('Objective:', solution['primal objective'])
```

```
Status optimal
x =
 [ 4.20e+01]
[ 6.30e+01]
[-2.84e-15]
[ 0.00e+00]
[ 0.00e+00]
[ 4.20e+01]

Objective: 6300.0
```

Рис. 5. Увеличение числа комплектов на единицу (задача 2)

Получаем то же решение. При этом видим, что потери материала увеличились на $40\cdot 3+2\cdot 90=300$ см.

Проводить такое же исследование для другого числа полуфабрикатов не имеет смысла, поскольку, как было показано выше, выбор всегда делается в пользу первых двух вариантов раскроя. Таким образом, увеличение числа полуфабрикатов на 5 единиц всегда будет давать один дополнительный комплект заготовок.

Приложение — блокноты программы

Блокноты программы, решающей задачу оптимального раскроя, доступны по ссылке:
https://github.com/valery42/Decision-Theory/tree/main/src/course/cutting