Задача о рюкзаке

Дано:

- W - грузоподъемность рюкзака.
- n - количество типов деталей.
- Для каждого типа детали i заданы вес w_i, стоимость c_i.

Задача: найти количество x_i для каждого типа предметов так, чтобы

$$\begin{cases} \sum_{i=1}^{n} c_i x_i \to max \ (1) \\ \sum_{i=1}^{n} w_i x_i \le W (2) \\ x_i \ge 0, \ x_i \in \mathbb{Z}. \ (3) \end{cases}$$

Задача о линейном раскрое

Задача целочисленного линейного программирования.
Требуется раскроить целостный объект, согласно исходным параметрам "детали".

- Максимизация суммарного дохода от раскроенных деталей.
- Минимизации отходов - остатков от целого куска материала.

Задача о линейном раскрое (максимизация)

Дано:

- l_0 - общая длина.
- n - количество типов деталей.
- Для каждого типа детали i заданы длина l_i, стоимость c_i.

Задача: найти количество x_i для каждого типа детали так, чтобы

$$\begin{cases} \sum_{i=1}^{n} c_i x_i \to max \ (1) \\ \sum_{i=1}^{n} l_i x_i \leq (2) \\ x_i \geq 0, \ x_i \in \mathbb{Z}. \ (3) \end{cases}$$

Динамическое программирование

1. Разбиение на подзадачи меньшего размера.
2. Нахождение оптимального решения рекурсивно, проделывая аналогичный алгоритм.
3. Использование полученного решения подзадач для конструирования решения исходной задачи. Оптимальное решение одной подзадачи используется в качестве исходных данных для следующей.

Линейный раскрой - динамическое программирование

Дано:

- l_0 - общая длина.
- n - количество типов деталей.
- Для каждого типа детали i заданы длина l_i, стоимость c_i.

Задача: найти количество x_i для каждого типа детали так, чтобы

$$\begin{cases} \sum_{i=1}^{n} c_i x_i \to max \ (1) \\ \sum_{i=1}^{n} l_i x_i \leq (2) \\ x_i \geq 0, \ x_i \in \mathbb{Z}. \ (3) \end{cases}$$

$$v[j] = max\{c[u] + v[j - l(u)]\},$$

где u - деталь выбранная на шаге $j \in 1 : n$.
Сортировка деталей по убыванию $\frac{c}{l}$.

Линейный раскрой - динамическое программирование

УНИВЕРСИТЕТ ИТМО

i	v[i]	U
0	0	-
1	0	-
2	0	-
3	4	{2}
4	7	{1}
5	7	{1}
6	8	{2}{2}
7	11	{1}{2}
8		
9		
10		
11		
12		
13		
14		

c	7	4	6
l	4	3	5

v[6] = max{7 + v[2]; 4 + v[3]; 6 + v[1]} = max{7; 8; 6} = 8
v[7] = max{7 + v[3]; 4 + v[4]; 6 + v[2]} = max{11; 11; 6} = 11

ITsMOre than a UNIVERSITY

Линейный раскрой - динамическое программирование

i	v[i]	U
0	0	-
1	0	-
2	0	-
3	4	{2}
4	7	{1}
5	7	{1}
6	8	{2}{2}
7	11	{1}{2}
8	14	{1}{1}
9	14	{1}{1}
10	15	{1}{2}{2}
11	18	{1}{1}{2}
12	21	{1}{1}{1}
13	21	{1}{1}{1}
14	22	{1}{1}{2}{2}

c	7	4	6
l	4	3	5

v[6] = max{7 + v[2]; 4 + v[3]; 6 + v[1]} = max{7; 8; 6} = 8
v[7] = max{7 + v[3]; 4 + v[4]; 6 + v[2]} = max{11; 11; 6} = 11

Ответ: x* = (2, 2, 0); f(x*) = 22

Метод ветвей и границ

Алгоритм:

1. Поиск частичного решения. Строится дерево перебора, где каждая ветвь является некоторым подмножеством решения. Выбирается обход дерева в глубину. Первое найденное допустимое решение становится нижней (или верхней для задачи с *max*) оценкой - *Rec*.
2. Оценка частичного решения x^k. Для задачи нахождения *min*:
   - Если $est(x^k) < Rec$, то шаг вперед (по дереву обхода).
   - Если $est(x^k) \geq Rec$, то шаг назад (по дереву обхода).
3. Каждое новое локальное решение обновляет оценку *Rec*.
4. При каждом шаге вперед проверяется допустимость решения.
5. Оптимальное решение - последняя оценка *Rec*, которая осталась после отсечения всех допустимых решений.

Линейный раскрой - м. ветвей и границ

1. Сортировка деталей по убыванию $\frac{c}{l}$.
2. Выбор самой выгодной детали.
3. Оценка оставшейся свободной части:

$$est[u] = \sum_{i=1}^{k} c_i x_i + (l_0 - \sum_{i=1}^{k} l_i x_i)\frac{c_j}{l_j},$$

где k - порядковый номер последней выбранной детали, j - деталь наиболее выгодная для оставшейся части.

Линейный раскрой - м. ветвей и границ

i	U	f_0	l	est	Rec?
0	(. . .)	0	14	24,5	-
1	(3, . .)	21	2	21	Да
2	(2, . .)	14	6	22	-
3	(2, 2, .)	22	0	22	Да, новый
4					
5					
6					
7					

c	7	4	6
l	4	3	5

est[0] = 14 * 7 / 4 = 24,5
est[1] = 7 * 3 + 2 * 0 = 21 /Rec/
est[2] = 2 * 7 + 6 * 4 / 3 = 22
est[3] = 2 * 7 + 2 * 4 = 22 /новый Rec/

Линейный раскрой - м. ветвей и границ

i	U	f_0	l	est	Rec?
0	(. . .)	0	14	24,5	-
1	(3, . .)	21	2	21	Да
2	(2, . .)	14	6	22	-
3	(2, 2, .)	22	0	22	Да, новый
4	(2, 1, .)	18	33	18	18 < 22 (шаг назад)
5	(2, 0, .)	14	6	21,2	21,2 < 22 (шаг назад)
6	(1, . .)	7	10	{2}{2}	20,3 < 22 (шаг назад)
7	(0, . .)	0	14	{1}{2}	18,7 < 22 (конец)

c	7	4	6
l	4	3	5

est[0] = 14 * 7 / 4 = 24,5
est[1] = 7 * 3 + 2 * 0 = 21 /Rec/
est[2] = 2 * 7 + 6 * 4 / 3 = 22
est[3] = 2 * 7 + 2 * 4 = 22 /новый Rec/

Ответ: x* = (2, 2, 0); f(x*) = 22