Исследование операций

НГУ Механико-математический факультет

4 курс, 1 семестр

Лектор: **Кочетов Юрий Андреевич**

http://www.math.nsc.ru/LBRT/k5/or_mmf.html

Исследование операций

Раздел индустриальной математики, прикладная наука, направленная на применение математических моделей и методов поиска оптимальных решений

Основные цели курса:

- научить студентов строить математические модели;
- разрабатывать эффективные методы решений оптимизационных задач;
- оценивать и сравнивать между собой алгоритмы по трудоемкости и требуемой памяти;
- понимать границы наших возможностей при разработке алгоритмов.

Схема исследования

Математическая модель

Математическая модель — объективная схематизация основных аспектов решаемой задачи или ее описание в математических терминах.

Математическая модель описывает исследуемую систему и позволяет выразить ее эффективность в виде ***целевой функции***

$$W = f(X, Y),$$

где $X = (x_1, \ldots, x_n)$ — управляемые переменные,

$Y = (y_1, \ldots, y_m)$ — неуправляемые переменные (исходные данные).

Связь между переменными X и исходными данными Y выражается с помощью ограничений

$$\varphi(X, Y) \leq 0.$$

Модели принятия решений

1. *Долгосрочное стратегическое планирование:*

задачи размещения производства, развитие нефтяной и газовой промышленности

2. *Среднесрочное планирование:*

транспортные задачи, задачи маршрутизации, задачи календарного планирования с ограниченными ресурсами

3. *Оперативное управление:*

задачи теории расписаний, задачи раскроя и упаковки

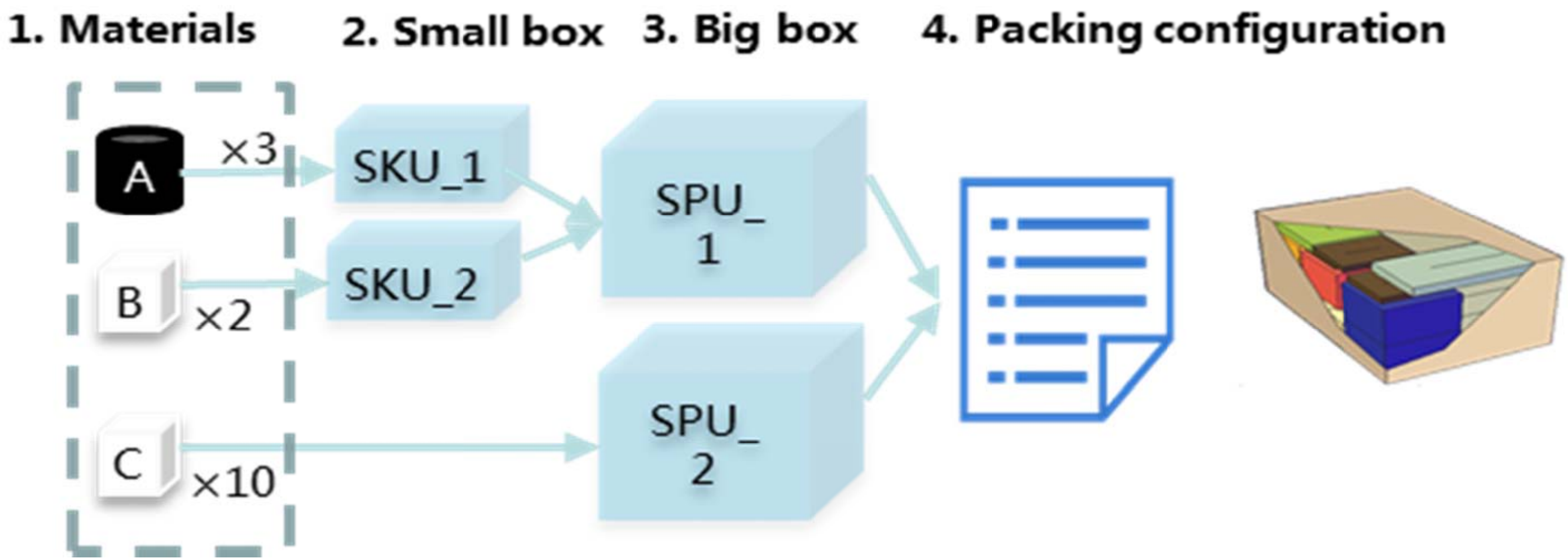
Управление складом

Оптимизационные задачи

- Оптимизация производственных помещений
- Расписание поставок
- Оптимизация простоев
- Планирование производства
- Расписание автоматических погрузчиков
- Расписание поставок продукции
- Упаковка в контейнеры (грузовики)
- ...

Задача упаковки

Описание задачи



Поиск стратегии упаковки на основе предыстории

Расписание в многономенклатурном производстве

более 40 предприятий
300 производственных линий
20 000 компонент
более 200 заказов

Целевые функции:

- Максимизация удовлетворенного спроса
- Минимизация суммарной стоимости
- Управление гибкими производственными системами в реальном времени

Распределительная задача

Имеем

n — число предприятий;

Y — количество единиц некоторого ресурса;

$f_k(x)$ — количество продукции, которое будет произведено на -м предприятии, если в него будет вложено *x* единиц ресурса (монотонно неубывающая функция).

Требуется: максимизировать объем продукции

$$f_1(x_1) + \cdots + f_n(x_n) \to max \quad (1)$$

$$x_1 + \cdots + x_n \leq Y \quad (2)$$

$$x_i \geq 0, \text{целые}, i = 1, \ldots n. \quad (3)$$

Идея динамического программирования (ДП)

Метод ДП (Р. Беллман, В.С. Михалевич, Н.З. Шор) можно трактовать как алгоритмическую версию рассуждений по индукции.

Пусть $s_k(y)$, $1 \le k \le n$, $0 \le y \le Y$, — оптимальное значение целевой функции задачи (1) – (3), где n заменено на k, Y заменено на y.

Требуется найти $s_n(Y)$ и набор переменных, на котором достигается это значение.

ТЕОРЕМА 1. Пусть $f_1, \dots, f_n$ — монотонно неубывающие функции. Тогда справедливы следующие ***рекуррентные соотношения:***

$$s_1(y) = f_1(y), \qquad 0 \le y \le Y; \tag{4}$$

$$s_k(y) = \max\{s_{k-1}(y-x) + f_k(x) | 0 \le x \le y\}, \qquad 2 \le k \le n, \qquad 0 \le y \le Y, \tag{5}$$

Доказательство: Соотношение (4) очевидно. По определению

$$s_k(y) \ge \max\{s_{k-1}(y-x) + f_k(x) | 0 \le x \le y\},$$

Пусть теперь $(x_1^*, \dots, x_k^*)$ — такой вектор, что $x_1^* + \dots + x_k^* \le y$ и

$$s_k(y) = f_1(x_1^*) + \dots + f_k(x_k^*).$$

Поскольку $s_{k-1}(y - x_k^*) \ge f_1(x_1^*) + \dots + f_{k-1}(x_{k-1}^*)$, имеем

$$s_k(y) = f_1(x_1^*) + \dots + f_k(x_k^*) \le s_{k-1}(y - x_k^*) + f_k(x_k^*). \blacksquare$$

Алгоритм ДП вычисляет множество $S_k = \{s_k(y) \mid 0 \leq y \leq Y\}, k = 1, \dots, n$ с помощью соотношений (4) и (5), где на каждом шаге оптимизируется ровно одна переменная.

Процесс вычисления $S_1, \dots, S_n$ называется ***прямым ходом*** алгоритма.

Число операций $\approx Y^2 n$

Память $\approx Yn$.

y	$S_1(y)$	$S_2(y)$	…	$S_n(y)$
0 1 2 ⋮ Y				 $S_n(Y)$

При ***обратном ходе*** алгоритма вычисляются значения $(x_n^*, \dots, x_1^*)$, с учетом того, что уже известны $S_k(y)$. Например, x_n^* определяется из уравнения $s_n(Y) = f_n(x_n^*) + s_{n-1}(Y - x_n^*)$ и так далее.

Число операций $\approx Yn$.   Память $\approx Yn$.

Характеристики алгоритмов

Для оценки качества алгоритмов будем использовать два параметра:

T_A— ***трудоемкость*** (число элементарных операций алгоритма *A*);

Π_A— требуемый ***объем памяти***.

Элементарная операция — одна из арифметических операций: сложение, вычитание, умножение, деление или логическая операция сравнение двух чисел.

Нас будет интересовать зависимость параметров алгоритма от длины записи исходных данных задачи с точностью до порядка величин.

Пример: При $T = \frac{3}{2}\,n^2$, будем писать $T = O(n^2)$ или $T \approx n^2$.

Полиномиальные алгоритмы

Определение. Алгоритм A называют ***полиномиальным***, если его трудоемкость T_A ограничена полиномом от длины записи исходных данных, то есть существует константа $c \geq 0$ и натуральное число k такие, что $T_A \leq c\,L^k$, где L — длина записи исходных данных.

Если трудоемкость алгоритма нельзя ограничить полиномом, то алгоритм называют ***экспоненциальным***.

Пример 1

Заданы n чисел $a_1, \dots, a_n$. Упорядочить их в порядке неубывания.

Алгоритм: выбираем наименьший элемент, затем наименьший из оставшихся и т.д.

Длина запись исходных данных:

$$L = \sum_{i=1}^{n} \log_2 a_i$$

$$T \sim O(n^2) \le O(L^2),$$

то есть алгоритм является полиномиальным.

Пример 2

Алгоритм динамического программирования для задачи (1)–(3). Пусть $f_i(x_i) = a_i x_i$, все коэффициенты a_i ограничены константой, $a_i \leq c$, но $Y = 2^n$.

Тогда длина записи

$$L = \sum_{i=1}^{n} \log a_i + \log Y \leq c'n, \text{ то есть } L = O(n),$$

но

$$T = O(Y^2 n) = O(2^{2L} L),$$

и алгоритм ДП не является полиномиальным.

Обобщим задачу (1)–(3):

$$f_1(x_1) + \dots + f_n(x_n) \to max \tag{1'}$$

$$h_1(x_1) + \cdots\ h_n(x_n) \le Y \tag{2'}$$

$$a_i \ge x_i \ge 0, \text{целые}, i = 1, \dots n. \tag{3'}$$

Если $h_i(x)$ — целочисленные монотонно неубывающие функции, то вместо (4)–(5) можно использовать следующие *рекуррентные соотношения:*

$$s_1(y) = f_1(x^*), \text{ где } x^* = \max\{x \le a_1\, |h_1(x) \le y\}, 0 \le y \le Y; \tag{4'}$$

$$s_k(y) = \max_{\{x \le a_k | h_k(x) \le y\}} \{f_k(x) + s_{k-1}\big(y - h_k(x)\big)\}, 2 \le k \le n,\ 0 \le y \le Y. \tag{5'}$$

Упражнение 1. Доказать справедливость соотношений (4′)–(5′).

Обратная задача — поиск наименьших затрат на получение заданного количества продукции:

$$h_1(x_1) + \dots + h_n(x_n) \to min \qquad (6)$$

$$f_1(x_1) + \dots + f_n(x_n) \geq D \qquad (7)$$

$$a_i \geq x_i \geq 0, \text{целые}, i = 1, \dots n. \qquad (8)$$

Если $f_k(x)$ — целочисленные монотонно неубывающие функции, то для решения задачи (6)–(8) можно использовать идеи динамического программирования.

Пусть $f_i^{-1}(d) = \min\{0 \le x \le a_i | f_i(x) \ge d\}$.

Для $1 \le k \le n$, $0 \le d \le D$ обозначим через $t_k(d)$ — оптимальное решение задачи (6)–(8), в которой n заменено на k, а D заменено на d.

Требуется найти $t_n(D)$.

Рекуррентные соотношения

$$t_1(d) = \begin{cases} \infty, & \text{если } f_1(a_1) < d, \\ h_1\left(f_1^{-1}(d)\right), & \text{если } f_1(a_1) \ge d, \end{cases} \quad 0 \le d \le D, \tag{9}$$

$$t_k(d) = \min\{t_{k-1}\left(d - f_k(x)\right) + h_k(x) | 0 \le x \le a_k, \ x \le f_k^{-1}(d)\} \tag{10}$$

$$k \ge 2, \ 0 \le d \le D$$

Упражнение 2. Доказать справедливость соотношений (9)–(10).

ТЕОРЕМА 2: Предположим, что D — наибольшее число, для которого оптимальное значение целевой функции задачи (6)–(8) не превосходит Y. Тогда оптимальное значение целевой функции задачи (1′)–(3′) равно D.

Доказательство: Пусть D удовлетворяет условию теоремы и $(x_1^*, \ldots, x_n^*)$ — соответствующее решение задачи (6)–(8).
Значит

$$f_1(x_1^*) + \ldots + f_n(x_n^*) \geq D \quad \text{и} \quad h_1(x_1^*) + \ldots + h_n(x_n^*) \leq Y.$$

Следовательно, D не превосходит оптимального решения D_1 задачи (1′)–(3′). Если бы D_1 было больше D, то решение задачи (6)–(8), в которой D заменено на D_1, тоже не превышало бы Y, что противоречит максимальности D. ■

Вопросы

- Полиномиальные алгоритмы — это такие алгоритмы,
  1. трудоемкость которых ограничена полиномом от длины записи исходных данных?
  2. память которых ограничена полиномом от длины записи исходных данных?
  3. трудоемкость и память которых ограничена полиномом от длины записи исходных данных?
- Полиномиальные алгоритмы всегда быстрее экспоненциальных?
- Алгоритмы динамического программирования
  1. всегда экспоненциальны, но полиномиальны по памяти?
  2. экспоненциальны, но быстрее полного перебора?
  3. полиномиальны по памяти и трудоемкости?