

Оглавление

1	Введение	3
1.1	Построение математических моделей	3
2	Оптимизационные (экстремальные) задачи	7
2.1	Алгоритмы	8
2.1.1	Линейное программирование	8
2.2	Свойства оптимизационных задач	9

Глава 1

Введение

Организованные системы — системы, в которых решения принимаются «сознательно» (например, люди, промышленные предприятия, магазины).

Раньше (примерно до второй мировой войны) все решения принимались лишь на основе опыта и здравого смысла. Однако позже появились сложные системы, в которых ни опыт, ни здравый смысл не работают.

Идея: рассматриваем числовые характеристики систем для принятия решений. Для этого используем математические модели (упрощённое, но адекватное описание реальной жизни).

1.1 Построение математических моделей

Исходные данные — конкретная проблема в конкретной ситуации.

Алгоритм:

1. Нужно понять, что будем оптимизировать? По каким критериям будет оценивать решения?
2. Какие характеристики существенно влияют на оптимизируемую характеристику?
3. Формулировка задачи с точным указанием характеристик. Переменные — меняющиеся (x, y, z, \dots) , параметры — константы a, b, c, \dots (например, площадь, помещение и т.д.).

4. Выбор обозначений, запись с учётом обозначений, ограничения и требования для переменных.

Пример:

Где нужно причалить к берегу? Хочется: побыстрее попасть в А.

Характеристики рассматриваемой системы:

- $AB = 15$ км
- $v_{BA} = 5$ км/ч
- $v_{\text{лодки}} = 4$ км/ч
- Расстояние до берега = 9 км
- O — точка причаливания
- x — расстояние от В до точки O
- течения воды нет (то есть скорость течения = 0)
- цвет лодки не существен :D

Оптимизируем: t .

Составим уравнение для t :

$$t = \underbrace{\frac{\sqrt{(x^2 + 81)}}{4}}_{\text{движение по воде}} + \underbrace{\frac{15 - x}{5}}_{\text{движение по суше}} \rightarrow \min_x$$

Ограничения: $0 \leq x \leq 15$ — потому что иначе решение точно не оптимальное.

$$t' = \frac{2x}{8\sqrt{x^2 + 81}} - \frac{1}{5} = \frac{10x - 8\sqrt{x^2 + 81}}{40\sqrt{x^2 + 81}},$$

$$\frac{x}{4\sqrt{x^2 + 81}} = \frac{1}{5},$$

$$\frac{x^2}{16(x^2 + 81)} = \frac{1}{25},$$

$$25x^2 = 16x^2 + 16 \cdot 81,$$

$$9x^2 = 16 \cdot 81,$$

$$x^2 = 16 \cdot 9,$$

$$x_1 = -12, x_2 = 12.$$

Решение $x_1 = -12$ не подходит, так как ...

$x = 12$ является ответом (проверить — упражнение :D). Можно заметить, что при $x > 12$ $t' > 0$, а при $x < 12$ $t' < 0$, а значит $x = 12$ — действительно точка минимума.

Глава 2

Оптимизационные (экстремальные) задачи

Формулировка экстремальной задачи:

1. Есть $f(x)$
2. Есть набор ограничений $g_1(x) \leq b_1 \dots$
3. $x \in X$, X — множество всех решений

Нужно: $\min_x f(x)$ или $\max_x f(x)$.

Допустимые решения — все значения, которые удовлетворяют всем $\{g_i\}$.

Определение: Если $\exists x^* \forall x \in X : f(x^*) \geq f(x)$ и x^* — допустимое решение, то x^* — решение задачи.

Другая формулировка: x^* является решением, если $f(x^*) = \max_{x \in X} f(x)$. $g(x)$ — может быть всё что угодно!!

Пример. Пусть наши исходные данные это

$$f(x) = c_1x_1 + c_2x_2 \rightarrow \max$$

$$g(x) = ax_1 + bx_2 \leq d.$$

$$\forall i x_i \geq 0.$$

Нужно найти x_1 и x_2 .

Пусть $p \in \mathcal{P}$ (\mathcal{P} — общая задача), а p — конкретная задача, где c_1, c_2, a, b, d — конкретные числа.

Общая задача — множество конкретных задач.

Определение

Длина входа задачи — Это количество ячеек в памяти, которое занимает задача. Допущение: каждое число занимает в памяти одну ячейку.

2.1 Алгоритмы

Определение

Элементарные операции — арифметические операции и операции сравнения.

$T_A(p)$ — количество элементарных операций. Это трудоёмкость алгоритма A решения задачи $p \in \mathcal{P}$.

Чем больше $|p|$, тем больше $T_A(p)$, поэтому будем оценивать так:

$$T_A(p) \leq \underbrace{f_A(|p|)}_{\text{оценка трудоёмкости}}$$

Если $f_A(|p|) = C \cdot |p|^k$, то будем называть алгоритм «хорошим» (полиномиальным).

Примерами задач, для которых существуют «хорошие» алгоритмы, являются математические задачи (линейное и нелинейное программирование, где x — множество векторов) и задачи комбинаторики (например, перестановки).

2.1.1 Линейное программирование

$$\sum_{j=1}^n c_j x_j \rightarrow \max_{(x_j)},$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i,$$

$$x_j \geq 0, j = 1 \dots n$$

или же

$$x_j \in \{0, 1, 2, \dots\}$$

Частный случай: $x_j \in \{0, 1\}$.

2.2 Свойства оптимизационных задач

Утверждение

$$\max_x f(x) = \max(-1) \cdot (-f(x)) = \underbrace{-\min -f(x)}_{\text{новая задача}}.$$

Утверждение

Оценка сверху (релаксированные задачи)

$$\max_{x \in X} f(x) \leq \max_{x \in X'} f(x)$$

если $x \subseteq X'$.

Утверждение

Сведение к другой задаче

Пусть есть задача $\mathcal{P} \max_x f(x)$. Будем говорить, что она сводится к задаче $\mathcal{Q} \max_{y \in Y} g(y)$, если из оптимального решения y^* задачи \mathcal{Q} можно построить оптимальное решение x^* задачи \mathcal{P} с помощью эффективного алгоритма.

Остаётся вопрос: как понять, что x^* — оптимальное решение?

Утверждение

Если

1. $f(x^0) \geq g(y^*)$,
2. $\forall x \in X \exists y : f(x) \leq g(y)$

то x^0 — оптимальное решение задачи \mathcal{P} .

Доказательство. Нам бы хотелось доказать, что x^0 — оптимальное решение. Пусть задача \mathcal{P} имеет оптимальное решение x^* . По второму пункту для x^* существует некоторый y^0 такой, что $f(x) \leq g(y^0)$.

Однако по первому пункту $f(x^0) \geq g(y^*)$, где y^* — оптимальное решение для \mathcal{Q} .

Значит имеем:

$$f(x^0) \geq g(y^*) \geq g(y^0) \geq f(x^*)$$

То есть

$$f(x^0) \geq f(x^*)$$

, где x^* — оптимальное решение задачи \mathcal{P} , а x^0 — предполагаемое оптимальное решение задачи \mathcal{P} . Значит $x^0 = x^*$ — тоже оптимальное решение задачи \mathcal{P} .

Пример

(\mathcal{P})

$$\sum_{j=1}^n c_j x_j \rightarrow \max_{(x_j)}$$

$$\sum_{j=1}^n a_j x_j \leq b$$

$$x_1 + x_2 = d$$

$$x_j \geq 0, j = 1 \dots n$$

Заметим, что x_1 можно выразить через x_2 (и наоборот).

Рассмотрим другую задачу:

(\mathcal{Q})

$$c_1(d - y_2) + \sum_{j=2}^n c_j y_j \rightarrow \max$$

$$a_1(d - y_2) + \sum_{j=2}^n a_j y_j \leq b$$

И плюс какие-то ограничения на $y_j, j = 2 \dots n$.

Покажем, что задача \mathcal{P} сводится к задаче \mathcal{Q} .

Доказательство. Пусть $y^* = (y_1^*, \dots, y_n^*) = (y_j^*)$ — это оптимальное решение задачи \mathcal{Q} . Будем строить вектор x^0 (предполагаемое оптимальное решение задачи \mathcal{P}) следующим образом:

$$x_1^0 = d - y_2,$$

$$x_j^0 = y_j, j = 2 \dots n$$