1

Вовед

1.1 Задача на математичко програмирање

Нека е дадена функцијата $f : X \to \mathbb{R}$ дефинирана на множеството $X \subseteq \mathbb{R}^n$. **Задачата на математичко програмирање** се состои од одредување на екстремни вредности (максимум или минимум) на функцијата f на множеството X, како и оптимални точки $x = (x_1, ..., x_n) \in X$ во кои се достигнуваат екстремните вредности. Задачата на математичко програмирање се запишува како

$$\min f(x) \quad \text{при услов} \quad x \in X. \tag{1.1}$$

Множеството X се нарекува **допустлива област**, а точките од тоа множество ($x \in X$) се **допустливи точки**. Ако $X = \mathbb{R}^n$, тогаш задачата (1.1) е **задача на безусловна минимизација**, ако $X \subset \mathbb{R}^n$, тогаш задачата (1.1) е **задача на условна минимизација**. Точката $x^* \in X$ (ако постои) за која важи

$$f(x^*) \leq f(x), \; \forall x \in X,$$

се нарекува **точка на глобален минимум**. Ако постои околина $U(x^*)$ на $x^* \in X$ и при тоа

$$f(x^*) \leq f(x), \; \forall x \in X \cap U(x^*),$$

тогаш x^* е **точка на локален минимум**.

Во случај да се бара максимум на функцијата f, користејќи го равенството $\max f(x) = -\min(-f(x))$, кое важи за секој x, задачата на максимизација се сведува на задача на минимизација.

Најчесто допустливата област X го има обликот

$$X = \{x \mid c_i(x) = 0, \; i \in \mathcal{E}, \; c_j(x) \leq 0, \; j \in \mathcal{I}\}, \tag{1.2}$$

каде $\mathcal{E}$ е множеството од индекси на **равенствените ограничувања** и $\mathcal{I}$ е множеството од индекси на **неравенствените ограничувања**.

Задачите на математичко програмирање се класифицираат во зависност од природата на **функцијата на целта** $f(x)$ и **ограничувањата** $c_i(x)$. Ако функциите f, c_i, $i \in \mathcal{E} \cup \mathcal{I}$ се линеарни, тогаш задачата (1.1) е **задача на линеарно програмирање**, во спротивно, кога некоја од функциите f, c_i, $i \in \mathcal{E} \cup \mathcal{I}$ не е линеарна, станува збор за **задача на нелинеарно програмирање**. Во нелинеарното програмирање се издвојува **задачата на конвексно програмирање**, кога се минимизира конвексна функција (функцијата f треба да е конвексна) над конвексна област (допустливата област X треба да е конвексно множество, а тоа е случај ако c_i, $i \in \mathcal{I}$ се конвексни функции).

Други типови на задачи на математичко програмирање, кои нема да бидат предмет на разгледување во оваа книга, се **задача на целобројно програмирање** (кога на променливите x им се додава и условот за целобројност), **задача на стохастичко програмирање** (кога се среќаваат некои случајности, неодредености во функцијата на целта и ограничувањата, од типот на случајни коефициенти, или веројатносни ограничувања).

1.2 Примери на задачи на математичко програмирање

Во овој дел ќе изложиме некои математички модели кои може да се сведат на задача на математичко програмирање.

1.2.1 Транспортна задача

Транспортниот проблем (Transportation Problem) е една од првите примени на линеарното програмирање. Математичкото изучување на транспортниот проблем се смета дека потекнува од 1920-тите години, за прв кој го разработувал овој проблем се смета дека е А. Н. Толстој.[1] Транспортата задача се состои во следното: Еден производ го произведуваат k производители и треба да го достават до m корисници, при тоа p_i е количина на производот со кој располага i-тиот производител, $i = 1, ..., k$, потоа q_j е количина на производот потребна на j-тиот корисник, $j = 1, ..., m$ и c_{ij} е цената на превоз (транспорт) на единица од производот од i-тиот производител до j-тиот корисник, $i = 1, ..., k$, $j = 1, ..., m$. Основната претпоставка (кај наједноставниот транспортен модел)

[1] Во 1930 год. А. Н. Толстој го објавува трудот "Methods of finding the minimal kilometrage in cargo-transportation in space" во зборникот "Transportation planning Volume I" за Националниот комесаријат за транспорт на Советскиот Сојуз

е дека вкупната побарувачка е еднаква на вкупната понуда, односно

$$\sum_{j=1}^{m} q_j = \sum_{i=1}^{k} p_i. \tag{1.3}$$

Треба да се одредат количините $x_{ij} \geq 0$ од производот кои треба да се транспортираат од i-тиот производител до j-тиот корисник, $i = 1, ..., k,\ j = 1, ..., m$. При тоа, вкупната цена на транспортот да е минимална.

Математичкиот модел на дадениот транспортен проблем може да се запише на следниот начин:

$$\begin{aligned}
\min \quad & \sum_{i=1}^{k}\sum_{j=1}^{m} c_{ij}x_{ij} \\
\text{при услов} \quad & \sum_{j=1}^{m} x_{ij} = p_i,\ i = 1, ..., k \\
& \sum_{i=1}^{k} x_{ij} = q_j,\ j = 1, ..., m \\
& x_{ij} \geq 0,\ i = 1, ..., k,\ j = 1, ..., m
\end{aligned} \tag{1.4}$$

Имено, врз основа на дадените услови, цената за транспорт на количината x_{ij} од i-тиот производител до j-тиот корисник е $c_{ij}x_{ij},\ i = 1, ..., k,\ j = 1, ..., m$, па вкупната цена која треба да се минимизира е дадена со изразот $\sum_{i=1}^{k}\sum_{j=1}^{m} c_{ij}x_{ij}$. Првите ограничувања $\sum_{j=1}^{m} x_{ij} = p_i,\ i = 1, ..., k$ се дека вкупната количина на транспотиран производ од i-тиот производител кон m-те корисници е еднаква на количината p_i со која располага производителот, $i = 1, ..., k$. Исто така, ограничувањата $\sum_{i=1}^{k} x_{ij} = q_j,\ j = 1, ..., m$ се однесуваат на тоа дека вкупната количина производ примен од страна на j-тиот корисник е еднаква на неговата побарувачка $q_j,\ j = 1, ..., m$. Последни се ограничувањата за ненегативност на одлучувачките променливи, количините на производот.

Горе изложениот модел (1.4) на транспортен проблем претставува упростена верзија, заради претпоставката (1.3). Во случај да не важи (1.3) ќе треба да се воведе фиктивен производител или фиктивен корисник, во зависност од тоа дали побарувачката или понудата се поголеми. Понатаму, кај реалните транспортни проблеми се јавуваат и дополнителни ограничувања, како, попусната моќ на сообраќајницата од i-тиот производител до j-тиот корисник, можноста за избор на превозно средство по различна цена, итн.

Иако транспортниот проблем се класифицира како задача на линерано програмирање, покрај методите за решавање на задачи на линеарно програмирање кои го решаваат овој проблем, заради неговата структура, постојат и специјални поефикасни методи за решавање на транспортните задачи.

1.2.2 Доделување на задолженија

Задачата на доделување на задолженија (Assignment Problem) може да се смета за специјален случај на транспортната задача во која имаме еднаков број на производители и корисници ($k = m$) и сите понуди и побарувачки се еднакви на еден ($p_i = q_i = 1$, $i = 1, ..., k$). Понатаму, доколку трошоците c_{ij} ги третираме како придобивки, ќе треба да се реши задачата на максимизација, на местото од задачата на минимизација.

Друга интерпретација на задачата на задолженија е следната: Еден раководител треба да распредели k работни задачи на k работници, така што секоја работна задача може да ја извршува само еден од работниците и секој работник може да извршува само една работна задача (на пример, кошаркарот i да игра на позиција j). При тоа, количината на добивка од работењето на i-тиот работник на j-тата работна задача, ја означуваме со c_{ij}. Раководителот треба да ги распредели работните задачи така што да ја максимизира вкупната добивка. За да го формираме матемтичкиот модел, ги означуваме со x_{ij} одлучувачките променливи, и при тоа x_{ij} прима вредност 1 ако i-тиот работник ја работи j-тата работна задача, и вредност 0 во сите останати случаи. Па, математичкиот модел на дадената задача на доделување на задолженија може да се запише на следниот начин:

$$\begin{aligned} \max \quad & \sum_{i=1}^{k}\sum_{j=1}^{k} c_{ij}x_{ij} \\ \text{при услов} \quad & \sum_{j=1}^{k} x_{ij} = 1,\ i = 1, ..., k \\ & \sum_{i=1}^{k} x_{ij} = 1,\ j = 1, ..., k \\ & x_{ij} \in \{0, 1\},\ i = 1, ..., k,\ j = 1, ..., k \end{aligned} \tag{1.5}$$

Моделот на доделување на задолженија има широк спектар на примена. Може да се искористи при наоѓање на "perfect match" меѓу одредена група на мажи и жени. Варијации на моделот се користат при правење на распоред на часови, распределба на класови во училници.

Според горната формулација (1.5), задачата на доделување на задолженија се вбројува во задачи на бинарно целобројно програмирање (BIP), одлучувачките поменливи примаат само две вредности 0 или 1. Забележуваме дека задачата на доделување на задолженија, практично може да се реши со испитување на сите можности (за k задолженија, има вкупно $k!$ можности), но и за релативна мала вредност на k таквиот број на можности е прилично голем, па потребно е многу време да се испитаат сите.