УНИВЕРЗИТЕТ “Св. КИРИЛ И МЕТОДИЈ” - СКОПЈЕ

ФАКУЛТЕТ ЗА ЕЛЕКТРОТЕХНИКА И ИНФОРМАЦИСКИ ТЕХНОЛОГИИ

-МАГИСТЕРСКА РАБОТА -

Тема

Математичко моделирање на некои проблеми од теоријата на оптимизација и примена

Ментор:
Проф.д-р.Боро Пиперевски

Изработил:
Елена Гелова, индекс бр. 236/08

Скопје, 2011 година

Членови на комисија :

Датум на одбрана: ________________________________

Датум на промоција: _____________________________

Научна област: Математика

Математичко моделирање на некои проблеми од теоријата на оптимизација и примена

Апстракт

Целта на оваа работа е да се запознаеме со оптимизацијата, со основните теоретски резултати а пред се со одбрани нумерички методи во нелинеарното програмирање. Условот на оптималност е тоа што ги спојува различните подрачја опфатени со оваа работа.

На почеток ќе биде даден преглед на елементарните математички поими и резултати кои се потребни за изучување на нелинеарното програмирање. Понатаму во класичната оптимизација ќе биде даден преглед на најважните класични резултати во теоријата на оптимизација на функција од неколку променливи, одредување на оптималните решенија со или без присуство на ограничувања.

Исто така се разгледуваат и неколку нумерички методи во нелинеарното програмирање, како на пример: методите на споредување на вредности, методата на парабола и кубната метода кои се сметаат за брзи и сигурни методи за функции кои не се премногу комплицирани. Објаснети се и теоремата на Кун и Такер, градиентните методи, методата на секачки рамнини, методите на линеаризација и алгоритамот за одредување на почетен план.

Разработено е и конвексното програмирање кое е наједноставната и најдобро обработена област од нелинеарното програмирање. Многу својства од линеарните програми се пренесуваат на конвексните програми. Најмногу користени услови во конвексното програмирање, со чија помош се одредува дали некое дозволено решение е оптимално, се Кун-Такеровите услови.

Исто така се разработени и методите на конвергенција и градиент методата, која може да се користи за решавање на програми за конвексни непрекинати диференцијабилни целни функции.

На крај е разгледана примената на нелинеарното програмирање.

Клучни зборови: нелинеарно програмирање, конвексно програмирање, оптимизација, методи на оптимизација, целна функција, Кун-Такерови услови, градиент метода.

Mathematical models of same problems in optimization theory and its application

Abstract

The objective of this work is to become familiar with optimization, with the basic theoretical results, and above all with specified numerical methods in the nonlinear programming. The optimality condition is what connects the different areas covered in this work.

Initially, there is a review of the elementary mathematical terms and results needed for studying nonlinear programming. Furthermore, in the classic optimization, a review will be given of the most important classic results in the optimization theory of the function with several variables, defining optimal results with or without limits.

Several numerical methods in the nonlinear programming are also being considered, for instance: value comparison methods, the parabola method, the cube method, which are considered to be fast and secure methods for not too complicated functions. The Kuhn-Tucker theorem, the gradient methods, the intersecting planes method, the linearization methods, and algorithm for defining an initial plan are also explained.

Convex programming as the most simple and most elaborated area of nonlinear programming is as well analyzed. Many features of the linear programs are transmitted to the convex programs. There is a frequent use of the Kuhn-Tucker conditions in the convex programming, which help to define the optimality of a certain solution.

The convergence method and the gradient method, which can be used for solving programs for convex continuous differentiated objective functions, are also being elaborated.

At the end the application of nonlinear programming is considered.

Key words: nonlinear programming, convex programming, optimization, optimization methods, objective function, Kuhn-Tucker conditions, gradient method.

Содржина:

ВОВЕД

Проблемите на оптимизација и оптималните процеси ги наоѓаме во многу области на природните, општествените и техничките науки. Всушност, овие проблеми секогаш биле и се неразделно поврзани со развојот на човештвото. Практичните проблеми поврзани со самоодржувањето и подобрувањето на условите за живот ги наведувале и наведуваат луѓето во планирањето на своите активности. Старите грчки филозофи, како Сократ, барале некој општ увид кој би покажал или сугерирал дека сите математички работи (вклучувајќи ја и улогата на човекот во општеството) биле, се или може да се прилагодат на „најдобриот начин“. Кон крајот на триесеттите и почетокот на четириесеттите години на минатиот век станало јасно дека многу економски проблеми, проблемите во администрацијата и бизнис организациите можат да се опишат со помош на едноставен математички модел. Во овие модели се оптимизира линеарната „целна функција“ (на пр. цена) на множеството „дозволени решенија“ (на пр. можните планови на производство), кое е исто така зададено со помош на линеарни релации. Овие модели се нарекуваат „линеарни програми“. Областа на оптимизација која се занимава со линеарните програми се нарекува „линеарно програмирање“ (или линеарно планирање). Важна карактеристика на линеарните програми е линеарните релации кои го одредуваат множеството на дозволени решенија, да не се само равенки, туку и неравенки. До сознанието дека економските модели типично се опишуваат со помош на неравенки, изгледа дека прв дошол Канторович во Ленинград во 1938 година. Неговите сознанија биле револуционерни, бидејќи во тоа време единственото подрачје на оптимизација биле варијациските пресметки кај кои сите ограничувања биле дадени со помош на равенки. Единствен исклучок била магистерската тема на Каруш во 1939 година на универзитетот во Чикаго. Меѓутоа, Каруш и неговите ментори не го виделе потенцијалот на оваа теза. Во исто време се појавила методологијата „операциони истражувања“ и на југословенското тло од страна на Ивановиќ во 1940 година. Ивановиќ ги решава проблемите на пресметување на минималниот број на возила кои се потребни да се транспортира одредена количина на материјал за дадени ограничувања. Мислите на Канторовиќ како и Ивановиќ во тоа време биле опструирани од војната. По формулацијата на Данцинговата симплекс метода и брзиот развој на компјутерите, што овозможи решавање на големи програми, линеарното програмирање почнало рутински да се користи во ситуации во кои треба да се одреди оптималниот план на производство. Теоријата на линеарното програмирање била комплетирана со Чарнсоновата работа во 1952 година каде била покажана конвергенцијата на симплекс методата и во случај на така наречени „дегенерирани решенија“. Во истиот број на списанието *Econometrica* се појавило и истражувањето на Чарнс, Купер и Мелон за „оптималното“ мешање на авионскиот бензин. Нивната работа била една од првите во која била забележана важноста на линеарното програмирање во индустриската примена.

На почетокот на педесеттите години станало јасно дека многу проблеми на оптимизација се нелинеарни, т.е. најчесто крајната функција или барем едно од ограничувањата е нелинеарна функција. Затоа било потребно да се развие теорија и нумерички методи за ова т.н. „нелинеарна програма“. Еден од првите трудови во кој исклучиво се учат нелинеарните програми, бил трудот на Кун и Такер во 1951 година. Нивните критериуми на оптимизација, за произволно но фиксно дозволено решение, станаа основа во теоријата на нелинеарното програмирање. Теоријата и нумеричките методи се значително посложени за нелинеарното програмирање во многу области на научните истражувања (на пр. нумерички методи, повеќекратно програмирање, стабилност на модел) и се доста значајни за развојот на науката.

1. Математичка припрема

“ Математиката е над целото село......”

Сангома од Свазија

Во ова поглавје даден е преглед на елементарните математички поими и резултати кои се потребни за изучување на оптимизацијата.

1.1 Квадратна форма на матрица

Матриците (квадратна и правоаголна) ги означуваме со големи букви од англиската азбука $A, B, C, \ldots$, векторите (колоните) ги означуваме со малите букви $x, y, z, \ldots$ Скаларниот производ на векторите $x = (x_i)$ и $y = (y_i)$ со n компоненти го означуваме со (x, y) т.е.

$$(x, y) = \sum_{i=1}^{n} x_i \cdot y_i.$$

Квадратната форма на матрицата претставува наједноставна нелинеарна функција. Тоа е функција од обликот :

$$f(x) = \sum_{i,j} a_{ij} x_i x_j = (x, Ax)$$

каде $(a_{ij}) = A$ претставува некоја $n \, x \, n$ матрица која се нарекува *матрица на квадратната форма* и

$$x = (x_i)$$

претставува вектор со n компоненти. $f(x)$ може да се прикаже и на овој начин:

$$f\,(x) = x^T(Ax).$$

каде x^T е транспонирано x

$$= \sum_{i=1}^{n} x_i (Ax)$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j \qquad .$$

Ако матрицата $(a_{ij}) = A$ е *симетрична*, за матрицата A велиме дека е симетрична ако $A = \mathrm{A}^{\mathrm{T}}$, важи

$$f(x) = \sum_{i=1}^{n} a_{ii}x_i^2 + 2\sum_{1\le i<j\le n} a_{ij}x_i x_j = (x, Ax)$$

Пример 1.1 Функцијата на квадратна форма на матрицата

$$\begin{bmatrix} 1 & 8 \\ -4 & 5 \end{bmatrix}$$

е следната:

$$f(x_1, x_2) = x_1^2 + 4x_1x_2 + 5x_2^2 \ .$$

Квадратната форма ја нарекуваме *позитивно определена (дефинитна)* ако

$$f(x_1, x_2, \dots \dots, x_n) > 0$$

секогаш кога меѓу променливите барем една е различна од нула.

Квадратната форма ја нарекуваме *негативно определена (дефинитна)* ако

$$f(x_1, x_2, \dots \dots, x_n) < 0$$

секогаш кога меѓу променливите барем една е различна од нула.

Формите за кои е $(x, Ax) < 0$ и $(y, Ay) > 0$, за некои x и y се нарекуваат недефинирани форми.

Во општ случај, не е едноставно да се определи во која група од зададените форми припаѓа квадратната форма. Помошта обично се наоѓа во математичкото решавање.

Прво *за* симетричната матрица A, ќе речеме дека е *позитивно семидефинитна* ако и само ако, нејзината квадратна форма е ненегативна; A *е позитивно дефинитна* ако и само ако, квадратната форма е позитивна; A *е негативно семидефинитна* ако и само ако, квадратната форма е непозитивна; A *е негативно дефинитна* ако и само ако, квадратната форма е негативна. Конечно матрицата A *е недефинитна* ако и само ако, квадратната форма е недефинитна. Едноставен критериум за одредување дали некоја матрица (или квадратна форма) е позитивно дефинитна или не, дал *Силвестер* во средината на деветнаесетиот век.

Критериум на Силвестер

-Една квадратна форма е позитивно определена ако и само ако главните минори на матрицата од таа форма се позитивни.
-Една квадратна форма е негативно определена ако и само ако главните минори од парен ред на нејзината матрица се позитивни, а од непарен ред се негативни.
Притоа главни минори на матрицата A се викаат минорите

$$a_{11},\ det\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix},\ det\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \dots, detA.$$

Пример 1.2 Дали квадратната форма

$$f(x_1, x_2) = x_1^2 + 4x_1x_2 + 5x_2^2$$

е позитивна?

Бидејќи f е генерирана со помош на симетричната матрица

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix}$$

треба да провериме дали е A позитивно дефинитна. Главните минори се:

$$a_{11} = 1, detA = 5 - 4 = 1;$$

позитивни. Користејќи ја Силвестеровата лема заклучуваме дека A е позитивно дефинитна. Квадратната форма f е позитивна т.е:

$$f(x_1, x_2) = x_1^2 + 4x_1x_2 + 5x_2^2 > 0$$

за секое x_1 и x_2, освен за $x_1 = x_2 = 0$.

Пример 1.3 Дали е квадратната форма

$$f(x_1, x_2, x_3) = 4x_1^2 + 6x_2^2 + 6x_3^2 + 6x_1x_2 - 4x_1x_3 + 4x_2x_3$$

позитивна?

Овде f е генерирано со помош на симетричната матрица

$$A = \begin{bmatrix} 4 & 3 & -2 \\ 3 & 6 & 2 \\ -2 & 2 & 6 \end{bmatrix}.$$

За да се најде одговорот потребно е да се испита дали A е позитивно дефинитна. Главните минори се:

$$a_{11} = 4,\ det\begin{bmatrix} 4 & 3 \\ 3 & 6 \end{bmatrix} = 15,\ detA = 26;$$

и сите се позитивни. Поради тоа, A е позитивно дефинитна матрица и f е позитивна форма.

Забелешка: После Силвестеровата лема, се добива впечаток дека симетричната матрица е позитивно семидефинитна ако и само ако, сите главни минори на матрицата се ненегативни (т.е. позитивни или еднакви на нула). Ова претпоставка не е точна, кај матрицата

$$B = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}$$

сите главни минори се ненегативни (поточно еднакви на нула) додека B не е позитивно семидефинитна матрица.

Критериумот за позитивна семидефинитност на матрицата значително е посложен. Прво се одредуваат т.н. “главни минори” т.е детерминанта на подматрицата која се наоѓа на пресеците на нејзините произволни редици и колони. Симетричната матрица A е позитивно семидефинитна ако и само ако, се сите дијагонални минори на матрицата A ненегативни.

Секоја позитивно дефинитна матрица очигледно е позитивно семидефинитна. Симетричната матрица A е негативно дефинитна ако и само ако, е $-A$ позитивно дефинитна. A е негативно семидефинитна ако и само ако е $-A$ позитивно семидефинитна.

Забелешка: Дефинитноста на матрицата може да се окарактеризира и со помош на т.н. *сопствени вредности* на матрицата. Секој број λ кој го задоволува равенството

$$Ax = \lambda x, x \neq 0$$

се нарекува сопствена вредност на матрицата. Квадратната матрица од n – ти ред има точно n сопствени вредности. Може да се покаже дека симетричната матрица A е позитивно дефинитна ако и само ако, сите сопствени вредности на матрицата A се позитивни, симетричната матрица A е позитивно семидефинитна ако и само ако сите сопствени вредности на матрицата A се ненегативни итн. Матрицата A е недефинитна ако има барем една позитивна и барем една негативна сопствена вредност. Карактеризацијата на квадратната форма која се пресметува со помош на сопствените вредности не е практична. Во одредувањето на позитивна (или негативна) дефинитност на квадратната форма, Силвестеровата лема е незаменлива.

1.2 Извод на функција: Градиент, Јакобиева и Хесеова матрица

Изводот на функција, заради повеќе причини, важен е во оптимизацијата. Како што ќе видиме наскоро, изводот го опишува однесувањето на функцијата, и на тој начин функциите се класифицираат на линеарни, конвексни, конкавни итн. Изводот ни е потребен за формулирање на условите за оптималност т.е за проверка дали некое допуштено решение е оптимално или не. Ако допуштеното решение не е оптимално, изводот се користи за формулирање на нумерички методи кои служат за наоѓање на подобри допуштени решенија и евентуално оптимално решение.

Означуваме со $f\colon R^n \to R^m$ векторска функција дефинирана на R^n (Евклидов простор од n-торки) со вредности во R^m, т.е.

$$f(x) = \begin{bmatrix} f^1(x) \\ f^2(x) \\ . \\ . \\ f^m(x) \end{bmatrix}$$

каде $x = (x_1, x_2, \dots \dots, x_n)$.

Извод на функција f (доколку постои) во произволна точка x претставува матрица со m редици и n колони:

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f^1(x)}{\partial x_1} & \frac{\partial f^1(x)}{\partial x_2} & . & . & \frac{\partial f^1(x)}{\partial x_n} \\ \frac{\partial f^2(x)}{\partial x_1} & \frac{\partial f^2(x)}{\partial x_2} & . & . & \frac{\partial f^2(x)}{\partial x_n} \\ . & . & . & . & . \\ . & . & . & . & . \\ \frac{\partial f^m(x)}{\partial x_1} & \frac{\partial f^m(x)}{\partial x_2} & . & . & \frac{\partial f^m(x)}{\partial x_n} \end{bmatrix}$$

Оваа матрица се нарекува *Јакобијева матрица*.

Во случајот кога $f: R^n \to R$, т.е кога f е скаларна функција зависна од n променливи, нејзиниот извод во точка $x = (x_1, x_2, \dots \dots, x_n)$ е прв случај на Јакобијева матрица, т.е

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} & \frac{\partial f(x)}{\partial x_2} \dots \dots \dots & \frac{\partial f(x)}{\partial x_n} \end{bmatrix}$$

Оваа матрица (односно вектор редица) се нарекува ***градиент***.

Пример 1.4 Извод на векторската функција

$$f(x_1, x_2, x_3, x_4) = \begin{bmatrix} x_1^2 + x_2 + x_1x_3 + 1 \\ x_1 - x_2^2 - x_1x_2^3 \\ x_2 + x_3^5 + x_4^6 \end{bmatrix}$$

во точката x е Јакобијева матрица:

$$\nabla f(x) = \begin{bmatrix} 2x_1 + x_3 & 1 & x_1 & 0 \\ 1 - x_2^3 & -2x_2 - 3x_1x_2^2 & 0 & 0 \\ 0 & 1 & 5x_3^4 & 6x_4^5 \end{bmatrix}$$

Специјално, во точката $x^* = (0,1,2,-1)^T$, извод на функцијата f е

$$\nabla f(x^*) = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 1 & 80 & -6 \end{bmatrix}.$$

Пример 1.5 Изводот на скаларната функција

$$f(x_1, x_2, x_3, x_4) = x_1^2 + x_2 + x_1x_3 + 1$$

во точката x е градиентот :

$$\nabla f(x) = [2x_1 + x_3 \quad 1 \quad x_1 \quad 0].$$

Специјално, во точката $x^* = (1,1,1,1)^T$, изводот е:

$$\nabla f(x^*) = [3 \quad 1 \quad 1 \quad 0].$$

Во многу случаеви, се користи и *вториот извод на скаларната функција* f во точката x, кој доколку постои има облик

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & . & . & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & . & . & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ . & . & . & . & . \\ . & . & . & . & . \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & . & . & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}.$$

каде $x = (x_1, x_2, \dots \dots, x_n)$.

Вториот извод на скаларната функција f во точка x е познат како ***Хесеова матрица***. (Вториот извод на векторската функција е покомплициран математички облик.).

Пример 1.6 Втор извод на функцијата

$$f(x_1, x_2) = x_1^2 + 4x_1x_2 + 5x_2^2$$

е:

$$\nabla^2 f(x) = 2\begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix}.$$

Пример 1.7 Вториот извод на функцијата

$$f(x_1, x_2, x_3) = 4x_1^2 + 6x_2^2 + 6x_3^2 + 6x_1x_2 - 4x_1x_3 + 4x_2x_3$$

е:

$$\nabla^2 f(x) = 2\begin{bmatrix} 4 & 3 & -2 \\ 3 & 6 & 2 \\ -2 & 2 & 6 \end{bmatrix}.$$

Во општ случај, вториот извод зависи од точката во која се пресметува, тоа ќе го покажеме со следниот пример:

Пример 1.8 Вториот извод на функцијата

$$f(x_1, x_2) = x_1^4 + x_1x_2^2$$

е:

$$\nabla^2 f(x) = \begin{bmatrix} 12x_1^2 & 2x_2 \\ 2x_2 & 2x_1 \end{bmatrix}.$$

Во точката $x^* = (1, -1)^T$,

$$\nabla^2 f(x^*) = \begin{bmatrix} 6 & -1 \\ -1 & 1 \end{bmatrix}$$

додека во точката $y^* = (0,0)^T$,

$$\nabla^2 f(y^*) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Познавањето на првиот, посебно на вториот извод на функцијата во проблемите на нелинеарното програмирање по правило значително го олеснува третирањето на проблемите. Ако е една функција недиференцијабилна, програмот станува теоретски и нумерички многу посложен.

Во случај кога функцијата зависи само од една променлива, изводот на функцијата f се означува со f' и f'', место ∇f и $\nabla^2 f$.

1.3 Конвексни и конкавни функции

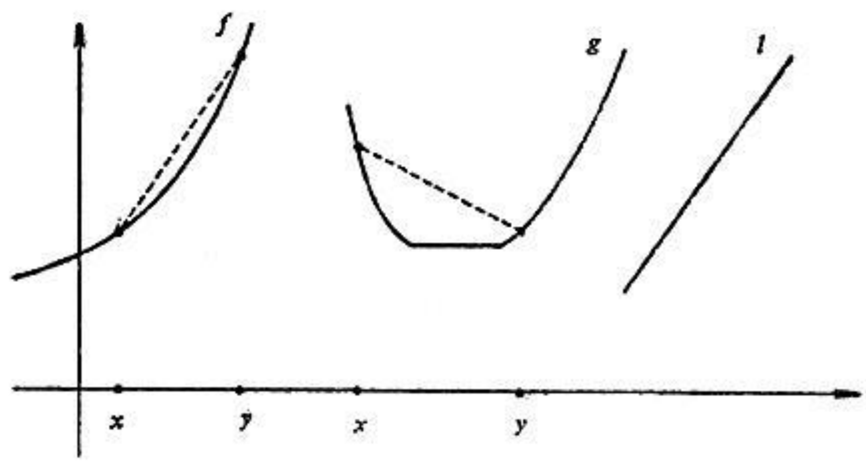
Конвексните функции можат да се сфатат како воопштување на линеарните функции. Тоа се скаларни функции од n променливи за кои ќе претпоставиме дека се дефинирани на целиот простор R^n. Се дефинираат на следниот начин:

Функцијата f е конвексна ако важи

$$f[\lambda x + (1-\lambda)y] \leq \lambda f(x) + (1-\lambda) f(y) \tag{1.1}$$

за сите вектори x и y и за сите броеви $0 \leq \lambda \leq 1$.

Конвексните функции можат многу едноставно геометриски да се окарактеризираат. Тоа се функции со следните својства: помеѓу било кои две точки x и y, функцијата f добива вредности кои се помали или еднакви од вредностите на праволинискиот сегмент кој ги спојува $f(x)$ и $f(y)$. (Види цртеж 1.1)



цртеж 1.1 Конвексни функции

Можеме да се увериме дека дефиницијата за конвексна функција не е секогаш соодветна за да се утврди дали е некоја зададена функција конвексна или не. Меѓутоа, ако функцијата е диференцијабилна, тогаш (1.1) може да се замени со многу попрактична релација во која го нема параметарот λ:

Диференцијабилната функција f е конвексна ако и само ако, важи

$$f(y) \geq f(x) + \nabla f(x)(y - x) \tag{1.2}$$

за сите вектори x и y.

Ако функцијата f е непрекината и диференцијабилна од втор ред (т.е ако f има втор извод и ако $x \to y$ повлекува $\nabla^2 f(x) \to \nabla^2 f(y)$, тогаш конвексната функција може да се окарактеризира со :

Непрекината диференцијабилна функција f од втор ред е конвексна ако и само ако Хесеовата матрица

$$\nabla^2 f(x)$$

е позитивно семидефинитна за секое x.

Пример 1.9 Дали функцијата

$$f(x_1, x_2, x_3) = (x_1 + x_2 - x_3)^2 + x_3^2$$

е конвексна?

Земајќи дека

$$\nabla^2 f(x) = 2\begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

е позитивно семидефинитна за секое x (односно $\nabla^2 f$ овде не зависи од x), заклучуваме дека f е конвексна функција.

Пример 1.10 Функцијата од една променлива $f(\xi) = e^{\xi}$ очигледно е конвексна. Меѓутоа функцијата од две променливи

$$f(x_1, x_2) = e^{x_1 x_2}$$

не е конвексна. Хесеовата матрица на функцијата $f(x_1, x_2)$ изгледа вака

$$\nabla^2 f(x_1, x_2) = e^{x_1 x_2}\begin{bmatrix} x_2^2 & 1 + x_1 x_2 \\ 1 + x_1 x_2 & x_1^2 \end{bmatrix}.$$

ова матрица е позитивно семидефинитна само во случај кога е

$$2x_1 x_2 \leq -1.$$

Можеме да забележиме дека ова неравенство го одредува подрачјето во кое е $f(x_1, x_2)$ конвексна функција. Графикот на функцијата

$$f(x_1, x_2) = \mathrm{e}^{\mathrm{x}_1 \mathrm{x}_2}$$

прикажан е на цртеж 1.2.

Меѓу конвексните функции посебно место земаат строго конвексните функции:

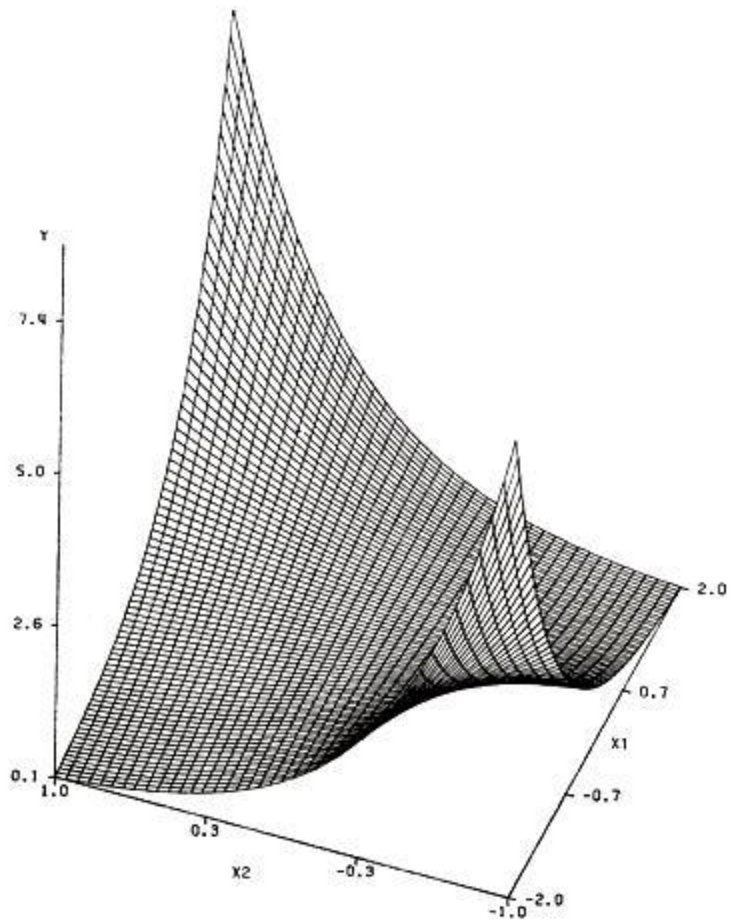
Функцијата f е строго конвексна ако важи

$$f[\lambda x + (1-\lambda)y] < \lambda f(x) + (1-\lambda)f(y)$$

за сите вектори x и y, $x \neq y$ и за сите броеви $0 < \lambda < 1$.

Строго конвексната функција f има својство помеѓу било кои две различни точки x и y, функцијата f да добива вредности кои се секогаш помали од вредноста на праволинискиот сегмент кој ги спојува $f(x)$ и $f(y)$. На цртеж 1.1, од три конвексни функции f, g и l единствено функцијата f е строго конвексна. Во случај на диференцијабилна функција f, f е строго конвексна ако и само ако, важи

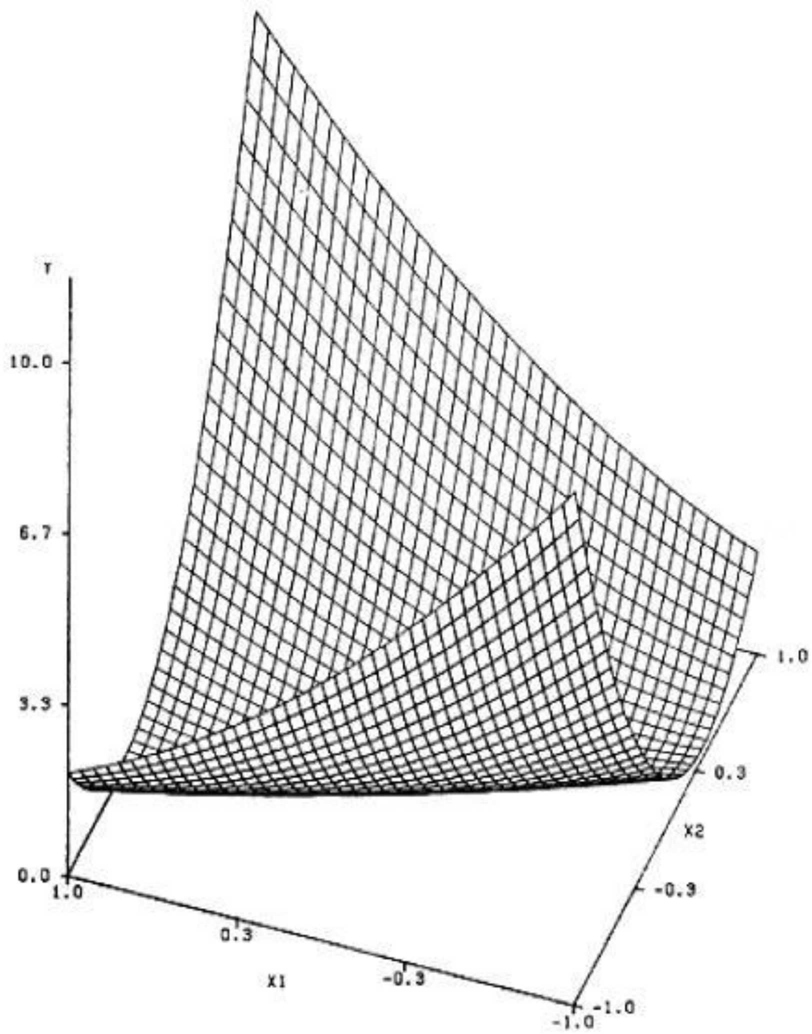
$$f(y) > f(x) + \nabla f(x)(y-x) \tag{1.3}$$



цртеж 1.2 График на неконвексна функција

Забелешка: Непрекината диференцијабилна функција f од втор ред е строго конвексна ако е Хесеовата матрица $\nabla^2 f(x)$ позитивно дефинитна за секое x.

Меѓутоа, f може да биде строго конвексна и во случаевите кога $\nabla^2 f(x)$ не е позитивно дефинитна за секое x. На пример $f(t) = t^4$ е строго конвексна функција, но Хесеовата матрица (т.е вториот извод) $f''(t) = 12t^2$ не е позитивно дефинитна за секое t. (За $t = 0$ следува $f''(0) = 0$).



цртеж 1.3 График на конвексна функција.

Пример 1.11 Квадратните форми

$$f(x_1, x_2) = x_1^2 + 5x_2^2 + 4x_1x_2$$

и

$$f(x_1, x_2, x_3) = 4x_1^2 + 6x_2^2 + 6x_3^2 + 6x_1x_2 - 4x_1x_3 + 4x_2x_3$$

се строго конвексни. Причина: нивните Хесеови матрици се позитивно дефинитни. Првата функција е прикажана на цртеж 1.3.

Од дефиницијата за конвексни и строго конвексни функции следи дека збирот од две конвексни функции е исто така конвексна функција, збир од конвексна и строго конвексна функција е строго конвексна функција, како и збир од две стого конвексни функции. Така на пример, $1 + t + e^t$ е строго конвексна функција.

Конвексните функции поседуваат важно својство: ако изводот на конвексната функција во некоја точка x е еднаква на 0 (т.е нула-вектор), тогаш во x точката, функцијата има најмала вредност т.е x ја минимизира функцијата f. Тоа следува од неравенството (1.2). Ако, е $\nabla f(x) = 0$ во (1.2), тогаш

$$f(x) \leq f(y)$$

за секое y. Кога имаме строго конвексна функција, $\nabla f(x) = 0$ значи дека x е единствената точка која ја минимизира f. За позитивна f, точката x во која е $\nabla f(x) = 0$ се нарекува *стационарна точка*.

Пример 1.12 За да се одреди точката за минимумот на функцијата

$$f(x_1, x_2) = x_1^2 + 5x_2^2 + 4x_1x_2$$

потребно е градиентот на функцијата да се изедначи на нула. Земајќи дека

$$\nabla f(x) = 2[x_1 + 2x_2 \quad 2x_1 + 5x_2]$$

точката на минимумот е решение на системот

$$x_1 + 2x_2 = 0, 2x_1 + 5x_2 = 0$$

т.е.

$$x_1 = 0, x_2 = 0.$$

Во случај на произволни (неконвексни) функции, $\nabla f(x) = 0$ е потребен но не и доволен услов за минимална точка.

Кога се работи за конвексни функции, секој локален минимум е глобален минимум. Во општ случај тие два минимуми мораат да се разликуваат и затоа сега ќе ги дефинираме. Користејќи ја следната нотација

$$\|x\| = (x^T x)^{1/2} = \left(\sum_{i=1}^{n} x_i^2\right)^{1/2}$$

за должина на векторот $x = (x_i)$ во R^n, под *околина* $N(x^*)$ на некоја фиксна точка x^* ги подразбираме сите точки кои се во внатрешноста на оддалеченост $\varepsilon > 0$ од x^* т.е

$$N(x^*) = \{x \in R^n : \|x - x^*\| \leq \varepsilon\}.$$

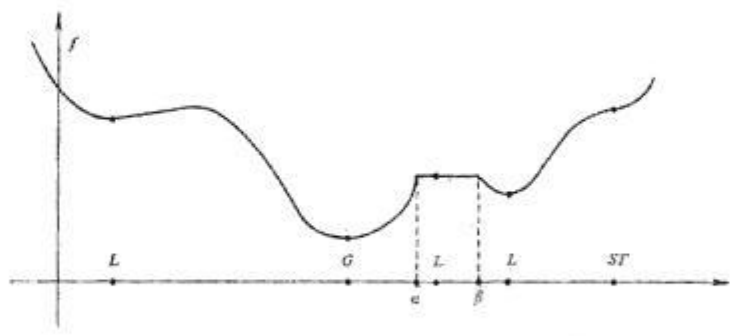
(Векторите често ги нарекуваме “точки” кога сакаме да ја потенцираме нивната геометриска смисла.) Точката x^* е минимум (или глобален минимум) на функцијата f ако е

$$f(x^*) \leq f(x)$$

за секоја x. Ако претходното неравенство важи само за некоја околина $N(x^*)$ т.е ако е

$$f(x^*) \leq f(x)$$

за секое $x \in N(x^*)$, тогаш велиме дека x^*, е ***локален минимум*** на функцијата. Секој ***глобален минимум*** истовремено е и локален. Обратното не важи.



Цртеж 1.4 Глобален и локален минимум на функцијата

На цртеж 1.4 функцијата f има единствен глобален минимум (означен со G) и бесконечно многу локални минимуми (означени со L). Секоја точка помеѓу α и β е локален минимум, надвор од овој интервал постојат само три локални минимуми. Точката ST е фиксирана точка. Во неа изводот се анулира, иако самата точка не е локален минимум. Точките (α и β не се локални минимуми).

Проблемот за одредување на глобален минимум, на неконвексните функции, е многу потежок од одредувањето на локалниот минимум.

Паралелно со конвексните функции се изучуваат т.н. ***конкавни функции***. Функцијата f е конкавна, ако нејзината негативна функција $-f$е конвексна, т.е ако важи

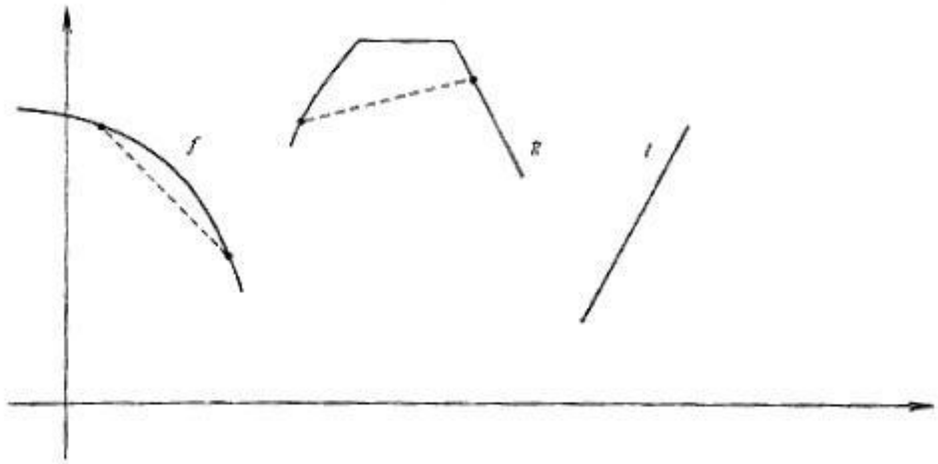
$$f[\lambda x + (1-\lambda)y] \geq \lambda f(x) + (1-\lambda) f(y) \tag{1.4}$$

за сите вектори x и y и за сите броеви $0 \leq \lambda \leq 1$. На цртеж 1.5 се прикажани три конкавни функции. Линеарната функција

$$l(x) = (\alpha, x) + \beta$$

каде (.,.) означува скаларен производ, е конвексна и конкавна функција (затоа се наоѓа на двата цртежи 1.1 и 1.5). Ова може лесно да се докаже:

$$\begin{aligned} l[\lambda x + (1-\lambda)y] &= (a, \lambda x + (1-\lambda)y) + \beta \\ &= \lambda[(a,x) + \beta] + (1-\lambda)[(a,y) + \beta] \\ &= \lambda l(x) + (1-\lambda) l(y) \end{aligned}$$



Цртеж 1.5 Конкавни функции

Од тука следуваат две неравенства

$$l[\lambda x + (1 - \lambda y)] \leq \lambda l(x) + (1 - \lambda) l(y)$$

и

$$l[\lambda x + (1 - \lambda) y] \geq \lambda l(x) + (1 - \lambda) l(y).$$

Значи l е истовремено конвексно и конкавно.

Исто така, јасно е дека диференцијабилната функција f е конкавна ако и само ако е (ако споредиме со (1.2))

$$f(y) \leq f(x) + \nabla f(x)(y - x)$$

за сите вектори x и y. Непрекината диференцијабилна функција f од втор ред е конкавна ако и само ако, Хесеовата матрица е $\nabla^2 f(x)$ семидефинитна за секое x. Функцијата f е строго конкавна ако (1.4) важи за строго нееднакво “>” за секое x и y, $x \neq y$ и за секое $0 < \lambda < 1$. Диференцијабилната функција f е строго конкавна ако и само ако (1.3) важи за “<”, за секое x и y, $x \neq y$. Непрекината диференцијабилна функција од втор ред е строго конкавна ако $\nabla^2 f(x)$ е негативно дефинитна за секое x. Секоја точка што е локален максимум на конкавната функција е и глобален максимум. Ако строго конкавна функција има максимална точка, тогаш таквата точка е единствена. Конвексните функции обично се минимизираат, додека конкавните функции се максимизираат (врз некоја листа од точки).

1.4 Конвексни множества

Множеството M се нарекува *конвексно множество* ако содржи праволиниски сегмент кој ги спојува секој пар на точки во M:

$$x \in M, y \in M, 0 \leq \lambda \leq 1 \text{ повлекува } \lambda x + (1-\lambda) y \in M.$$

Примерите со конвексните множества во применета математика се многу чести и разновидни. Тука ќе се запознаеме со најважните и најчести конвексни множества кои се користат во линеарното и нелинеарното програмирање.

Пример 1.13 Множеството на сите дозволени решенија F во *линеарното програмирање* е конвексно множество. Претпоставуваме дека множеството на дозволени решенија е зададено во облик

$$F = \{x: Ax = b, x \geq 0\},$$

каде A е некоја $m \times n$ матрица, b вектор во R^m и $x = (x_i)$ векторска променлива во R^n. (Со „$x \geq 0$" означено е „$x_i \geq 0$ за секои $i = 1,2, \dots, n$"). Дека F е конвексно множество се докажува на следниот начин: Земаме произволна вредност за $x \in F, y \in F$ и $0 \leq \lambda \leq 1$. Треба да се докаже дека и конвексната комбинација $z = \lambda x + (1-\lambda) y$ е во множеството F. Јасно е дека $z \geq 0$ (бидејќи $x \geq 0, y \geq 0$ и $\lambda \geq 0,\ 1-\lambda \geq 0$).

Понатака

$$\begin{aligned} Az &= \lambda \mathrm{A}x + (1-\lambda) Ay, \text{ за линеарната матрица } A \\ &= \lambda b + (1-\lambda) b \text{ , бидејќи } Ax = b \text{ и } Ay = b \\ &= b. \end{aligned}$$

Со тоа докажавме дека $z \in F$, па следува F е конвексно множество.

Пример 1.14 Множеството на дозволените решенија во така нареченото конвексно програмирање може да се изрази со помош на неравенките:

$$\begin{gathered} f^1(x_1, \dots, x_n) \leq 0 \\ f^2(x_1, \dots, x_n) \leq 0 \\ \dots\dots\dots\dots\dots\dots \\ f^p(x_1, \dots, x_n) \leq 0 \end{gathered}$$

каде $f^1, f^2, \dots, f^p$ се конвексни функции. Поради едноставност, овие неравенки можат да се напишат во обликот

$$\begin{gathered} f^1(x) \leq 0 \\ f^2(x) \leq 0 \\ \dots\dots\dots\dots \\ f^p(x) \leq 0 \end{gathered}$$

каде $x = (x_1, \dots, x_n)^T$, или уште поедноставно

$$f^i(x) \le 0, i = 1, \ldots, p.$$

Тврдиме дека множеството на дозволените решенија за конвексните функции $f^i, i = 1, \ldots, p$,

$$F = \{x: f^i(x) \le 0, i = 1, \ldots, p\}$$

е конвексно множество.

За да го докажеме тоа тврдење, земаме произволни $x \in F, y \in F$ и $0 \le \lambda \le 1$. Треба да се докаже дека $z = \lambda x + (1 - \lambda)y$ припаѓа во множеството F. За секое $i = 1, \ldots, p$ ја имаме оваа ситуација:

$$f^i(z) = f^i(\lambda x + (1 - \lambda)y) \le \lambda f^i(x) + (1 - \lambda)f^i(y)$$

бидејќи f^i е конвексно ≤ 0, и $f^i(x) \le 0, f^i(y) \le 0$,

(знаеме дека $x \in F, y \in F$), $\lambda \ge 0,\ 1 - \lambda \ge 0$.

Затоа $f^i(z) \le 0,\ i = 1, \ldots, p$ што значи дека $z \in F$ т.е. F е конвексно множество.

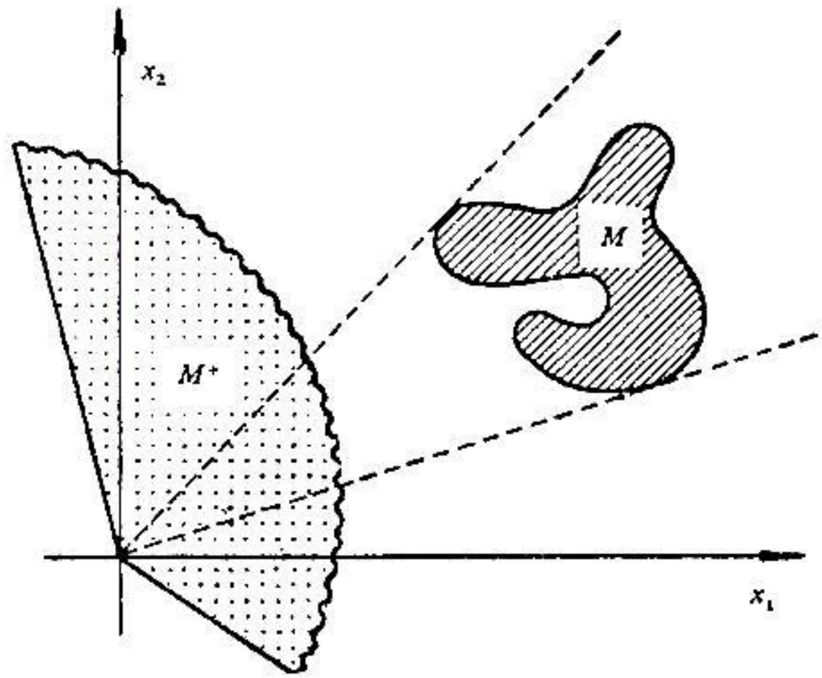
Нумерички пример е на пример :

$$f^1 = -5x_1 - 2x_2 + 2x_2^2 \le 0$$

$$f^2 = \qquad - 4x_2 + \ x_x^x \le 0.$$

Бидејќи f^1 и f^2 се конвексни функции, од горната дискусија следи дека множеството на сите точки (x_1, x_2), кои ги задоволуваат овие две неравенки е конвексно множество.

Сега ќе се запознаеме со нешто по комплицирани комплексни множества. За произволно множество M, поларното множество на M се означува со M^+ и се дефинира со $M^+ = \{u: ux \ge 0 \text{ за секое } x \in M\}$. Ако е M целиот простор R^n, поларноста е нулти вектор и обратно. На цртеж 1.6 конструирано е поларно множество во R^2.



цртеж 1.6 Поларно множество M

Пример 1.15 Поларното произволно множество M е конвексно множество.

Доказ: Земаме произволни $u^1 \in M^+$ и $u^2 \in M^+$ и $0 \leq \lambda \leq 1.$ Тогаш

$$[\lambda u^1 + (1-\lambda)u^2]x = \lambda(u^1 \cdot x) + (1-\lambda)(u^2 \cdot x) \geq 0$$

За секое $x \in M$, ако $u^1 \cdot x \geq 0,\ u^2 \cdot x \geq 0,\ \lambda \geq 0$ и $1-\lambda \geq 0.$

Затоа $\lambda u^1 + (1-\lambda)u^2 \in M^+$. Бидејќи сите конвексни комбинации се наоѓаат во множеството M^+, констатираме дека M^+ е конвексно множество.

Поларитетот на секое множество M^+ е всушност *конвексен конус*, т.е. конвексно множество кое е исто така и конус. Велиме да множеството K *е конус* ако

$$x \in K, \lambda \geq 0 \text{ повлекува } \lambda x \in K.$$

Освен во тривијалниот случај, кога K е нула-вектор, конусот е неограничено множество. Примери за конвексни конуси се исто така целиот простор R^n, ненегативниот орт

$$R^n_+ = \{x \in R^n : x \geq 0\}$$

и нула-векторот. Пример за неконвексен конус се координатните оски во R^2, т.е. сите точки (x_1, x_2) за кои производот $x_1 x_2 = 0$.

Еден од најважните резултати во конвексните множества во нелинеарното програмирање е т.н. „теорема на сепарација на хиперрамнини“. Таа едноставно го кажува следново:

Две непразни конвексни множества, кои немаат ни една заедничка точка, можат да се раздвојат со хиперрамнини.

Во математичка терминологија овој резултат изгледа вака:

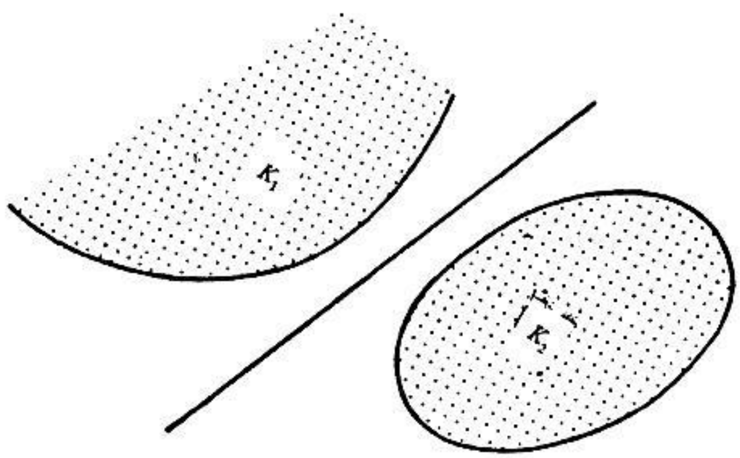
Нека K_1 и K_2 се две непразни конвексни множества во R^n така да пресекот им е празно множество, т.е. $К_1 \cap К_2 = \emptyset$. Тогаш постои вектор $u \in R^n$ различен од нула и број a за кој важи

$(u, x) \leq a$ за секое $x \in K_1$

и

$(u, x) \geq a$ за секое $x \in K_2$.

Ова тврдење е прикажано на цртеж 1.7.



цртеж 1.7 Раздвојување на конвексни множества со хиперрамнини

(За даден вектор u и број a „*хиперрамнина*“ е множество на сите точки x кои ја задоволуваат равенката $(x, u) = a$. На пример ако е $u = (1, -2,3,5)^T$ и $a = 1$, соодветната хиперрамнина во R^4 е

$$x_1 - 2x_2 + 3x_3 + 5x_4 = 1.$$

Во случај на три променливи, хиперрамнината се нарекува „рамнина“, а во случај на две променливи хиперрамнината е „права“ како на цртежот 1.7. Овој резултат ќе се користи во формулирање на условите за оптималност со помош на Лагранжовата функција. Компонентите на векторот u ќе се препознаваат како Лагранжови множители.

1.5 Решение на системот на равенки и неравенки

Секој критериум на оптималност во нелинеарното програмирање може да се формулира на два начини: на примерен и дуален начин. Примарниот критериум на оптималност тврди дека некои равенки и неравенки во оптималната точка немаат решение (т.е. не е конзистентен), дуалниот критериум пак тврди дека во оптималната точка некој друг систем („дуален“ на првиот систем) е решлив. Врската помеѓу такви два системи ја даваат т.н. *теореми на алтернативност*. Во овие теореми се разгледуваат два системи од кои само еден има решение. Со други зборови, еден систем (било кој) е решлив ако и само ако другиот систем нема решение. Во ова поглавие ќе дадеме четири такви теореми.

Првата е Фаркашова теорема, која прв пат се појавила на унгарски јазик во 1895 година.

***Фаркашова теорема*.** На два системи:

(1) $Ax = b \qquad x \geq 0$

(2) $A^T y \geq 0, \qquad (b, y) < 0$

само еден има решение.

Фаркашовата теорема може и на овој начин да се изрази: Системот

$$Ax = b, \qquad x \geq 0$$

има решение ако и само ако,

$$A^T y \geq 0, \quad (b, y) < 0.$$

Теоремата на алтернативност за хомоген систем ја дал Жордан уште во 1873 година.

***Жорданова теорема*.** На два системи:

(1) $Ax = 0 \quad x \geq 0, \ x \neq 0$

(2) $A^T y > 0$

има само едно решение.

Овде, како и во иднина, ќе се користи нотација $u = (u_i) > 0$ која значи дека секоја компонента на векторот u е позитивна, т.е. $u_i > 0, i = 1, \ldots, n$. Следниот резултат е важно решение на Жордановата теорема.

***Жорданова теорема за конвексни конуси*.** Нека C е конвексен конус со непразна внатрешност во R^n и нека K е произволен конвексен конус во R^n. Од двата системи:

(1) $uA^T + p = 0, \ u \in C^+, \ u \neq 0, \ p \in K^+$

(2) $A^T y \in intC, \ \ y \in K$

само еден има решение.

Овде „int C“ ја означува *внатрешноста на множеството* C. Ако спецификуваме $C = R^n$ (ненегативен орт во R^n) и $K = R^m$,

тогаш соодветната поларност $C^+ =$ ненегативен орт во просторот на редот со n компоненти и $K^+ =$ нула ред со m компоненти. Сега, горното постанува

(1) $uA^T = 0, u \geq 0, u \neq 0$ (u е ред)

(2) $A^T y > 0$.

После транспонирањето на системот (1) и воведувањето на ознаката $x = u^T$, се гледа да Жордановата теоремата за конвексни конуси навистина се сведува на класичната Жорданова теорема.

Најелегантна и најопшта (за наши потреби) е теоремата на алтернатива која ја дале Дубовички и Милутин во 1963 год. Прво ни треба концепт на „отворен конус“. Велиме дека множеството C е *отворен конус* ако е C конус и истовремено отворено множество. (*Множеството е отворено* ако содржи некоја околина на секоја точка која припаѓа на тоа множество.)

Теорема на Дубовички и Милутин. Нека $C_1, \dots, C_m$ се отворени конвексни конуси и нека C_{m+1} е произволен конвексен конус. Тогаш пресекот на тие конуси е празно множество т.е.

$$\bigcap_{i=1}^{m+1} C_i = \emptyset$$

ако и само ако постојат векторите $y^i \in C_i^+, i = 1, \dots, m+1$, од кои барем еден е различен од нула така да

$$y^1 + \cdots + y^m + y^{m+1} = 0. \tag{1.5}$$

Во апстракното формулирање на проблемите на нелинеарното програмирање, (1.5) одговара на т.н. Ојлер-Лагранжов услов.

Теоремата на алтернативност е корисна не само во формулацијата на критериумите на оптималност, туку и во многу други случаи. Во следниот пример ќе испитаме дали множеството на дозволените решенија на линеарното програмирање е празно, со примена на Фаркашовата теорема.

Пример 1.16 Дали системот

$$\begin{aligned} x_1 + x_2 - x_3 + x_4 &= 1 \\ -x_1 \qquad - 2x_3 + x_4 &= 3 \\ -x_1 + x_2 + x_3 + 3x_4 &= 1 \\ x_i \geq 0, i &= 1,2,3,4 \end{aligned} \tag{1.6}$$

има решение?

Одговор може да се добие ако прво се реши системот на равенки и после тоа се испита дали некое од решенијата е ненегативно.

Другиот начин е да се испита решливоста на соодветниот „дуален“ систем (2) во Фаркашовата теорема. Овој систем има облик

$$
\begin{aligned}
y_1 - y_2 - y_3 &\geq 0 \\
y_1 \qquad + y_3 &\geq 0 \\
-y_1 - 2y_2 + y_3 &\geq 0 \\
y_1 + y_2 + 3y_3 &\geq 0 \\
y_1 + 3y_2 + y_3 &< 0.
\end{aligned}
\tag{1.7}
$$

Се работи за решлив систем бидејќи на пр. $y_1 = 1,\ y_2 = -1, y_3 = 1$ ги задоволува сите неравенки.

Заклучок: Бидејќи системот (1.7) има решение, почетниот систем (1.6) нема решение.

2. Вовед во оптимизација

Теоријата на оптимизација се занимава со развој на модели и методи со кои се определуваат оптимални решенија на математички дефинирани проблеми. Било кое решение на математичкиот проблем x, во општ случај може да биде n – торка на решение во облик $x = (x_1, x_{2,\ldots\ldots,} x_n)$. Компонентите на решението $x_i, i = 1, \ldots, n$ се нарекуваат управувачки параметри. Вообичаено оптималното решение на математички дефиниран проблем се означува со x^*.

За да може да констатираме дека некое решение е оптимално, потребно е да постои мерка со која се одредува неговиот квалитет и ќе овозможува негово споредување со други можни решенија. Во математичкиот модел мора да постои некој функција со која на секое решение се придружува некој вредност која претставува мерка за квалитет. Таквата функција вообичаено се нарекува целна функција и најчесто математички се означува со $F(X)$. Задачата на оптимизација е одредување на решение кое дава оптимална (минимална или максимална) вредност на $F(X)$.

Променливите кои што треба да се определат обично се условени со меѓусебни релации и ограничувања. Секое решение кое ги задоволува постоечките ограничувања се нарекува можно решение.

Вредноста на функцијата $F^* = F(x^*)$ која одговара на оптималното решение се нарекува оптимална вредност или **оптимум.**

Неопходни претпоставки кои треба да постојат за да може да се оствари една задача на оптимизација се :

1. Објект на оптимизација. Може да биде произволен процес,апарат,итн.
2. Критериум на оптималност кој се нарекува и целна функција.
3. Управување на објектот на оптимизација. За да може да се изврши процесот на оптимизација потребно е објектот на оптимизација да може да се управува. За да се осигура можноста за управување на објектот на оптимизација неопходно е тој да има управуваки параметри кој ќе можат да се сменуваат независно еден од друг.
4. Метод на оптимзација. За даден управувачки објект и целна функција неопходно е да се одбере метод за определување на оптимум. Не постои некој универзален метод за решавање на сите задачи на оптимзација. Изборот на метод обично се прави врз основа на целната функцијата и избраниот објект.

Задачите на оптимизација можеме да ги поделеме како задачи на статичка оптимизација и задачи на динамичка оптимизација. Каи задачите на статичката оптимизација објектот се разгледува во единствена не променлива состојба. Во задачите на динамичката оптимизација целната функција зависи од параметри кои што претставуваат променливи.

Методите на оптимизација можеме да ги поделеме на следниот начин :

а) Аналитички методи кои се засноваат на аналитичка анализа на изводот на целната функција. Во овие методи екстремната вредност на функцијата $F(X)$ се добива со наоѓање на такви вредности на X за кои $F'(X) = 0$. За поголем број нелинеарни проблеми аналитичките методи не се од посебно значење.

б) Нумерички (итеративни) методи се засноваат на дефинирана нумеричка итерација за приближна апроксимација на решението. Овие методи се најпогодни за програмирање. Се делат во две групи:

-Градиентни методи кои користат извод на целната функција

-Неградиентни методи кои не користат извод на целната функција

в) Графички методи кај кои имаме графичко претставување на целната функција и ограничувањата. Екстремните вредности на целната функција се добиваат од графикот преку пребарување. Овие методи можат да се применат само на целна функција од еден или два управувачки параметри.

г) Експериментални методи каи кој се добива екстрем врз основа на серија извршени експерименти. Експерименталните методи се користат само во случаи кога математичкиот модел на објетот се покажува неадекватен.

Основни особини кои треба да ги задоволува алгоритмот на оптимизација се:

-Конвергенција (добивање нумеричко решение со конечен број на чекори)

-Брза конвергенција (добивање решение за што пократко време и за што помал број изрaчунати вредности на целната функција).

-Универзалност. Пожелно е алгоритамот да биде применлив на што повеќе класи на задачи. Меѓутоа универзален метод за решавање на сите типови оптимизациони задачи не постои.

2.1 Класична Оптимизаци

Овде ќе дадеме преглед на најважните класични резултати во теоријата на оптимизација на функција од неколку променливи. Покрај резултатите кои го опишуваат решението, ќе прикажеме и неколку класични методи за одредување на оптимални решенија со или без присуство на ограничувања. Овие методи се од теоретска природа и поради тоа можат да се користат успешно само во едноставни случаи.

2.1.1 Стационарни точки

Нека функцијата $f(x)$ со n променливи $x = (x_1, \dots, x_n)^T$ е дефинирана и диференцијабилна на целиот простор R^n.

Ако x^* е локален минимум или локални максимум на функцијата f , тогаш x^* мора да биде нејзина стационарна точка. Но, ако x^* е стационарна точка, тоа не значи дека x^* мора да биде и локален оптимум.

Пример 2.1 Функцијата $f(x) = x_1^2 + 4x_1x_2 + 5x_2^2$ има стационарна точка

$$x_1^* = 0, \qquad x_2^* = 0$$

која е нејзин локален минимум. (Бидејќи f е конвексна фунција, ова точка е всушност глобален минимум.) Меѓутоа, резултатот е стационарна точка и за функцијата $f = x_1x_2$, иако резултатот не е нејзин локален оптимум.

Информацијата само за првиот извод не е во општ случај доволна за да се заклучи дали е некоја точка локален минимум или не. Затоа ни е потребна информација и за вториот извод. x^* е изолирана точка на оптимум, ако е таа точка во некоја околина единствен оптимум. На пример, ако строго конвексна функција има точка на минимум, тогаш се работи за единствен, изолиран минимум. Точките на минимумот на произволна конвексна функција не мораат да бидат изолирани .

Сега ќе дадеме доволен услов да некоја точка x^* е изолиран оптимум.

> Нека е f два пати непрекинато деријабилна функција. Ако во некоја точка x^* важи
>
> $$\nabla f(x^*) = 0$$
>
> и ако Хесеовата матрица
>
> $$\nabla^2 f(x^*)$$
>
> е позитивно дефинитна, тогаш f има локален минимум во x^*. Ако $\nabla f(x^*) = 0$ и ако $\nabla^2 f(x^*)$ е негативно дефиниран, тогаш x^* е локален максимум. Така да, оптималните точки се изолирани.

Овој резултат е очигледен: Кога $\nabla^2 f(x^*)$ е позитивно дефиниран, тогаш поради непрекинатоста и диференцијабилноста, f е строго конвексна во некоја околина на точката x^*. Изразувањето на градиентот сега значи дека x^* ја минимизира f. Згора на тоа, поради строгата конвексност на функцијата f, точката на минимум е изолирана. Кога $\nabla^2 f(x^*)$ е негативно дефинитна, f е строго конкавна во околината на точката x^* и следи заклучокот.

Условите за оптималност на произволна (т.е. изолирана или неизолирана) точка на оптимум гласат:

Нека f е два пати непрекинато диференцијабилна функција. Ако x^* е локален минимум на функцијата f тогаш

$$\nabla f(x^*) = 0 \tag{2.1}$$

и Хесеовата матрица

$$\nabla^2 f(x^*)$$

е позитивно семидефинитна. Од друга страна, ако во некоја точка x^* важи (2.1) и ако е Хесеовата матрица

$$\nabla^2 f(x^*)$$

позитивно семидефинитна за секое x во некоја околина $N(x^*)$, тогаш x^* е локален минимум на функцијата f. Ако зборот „позитивно" се замени со зборовите „негативно", резултатите важат за локалниот максимум.

Пример 2.2 Стационарната точка на (неконвексна) функција

$$f(x) = x_1^4 + x_1^3 - x_1 + x_2^4 - x_2^2 + x_2 + x_3^2 - x_3 + x_1x_2x_3$$

е решение на системот (2.1) т.е.

$$\begin{aligned} 4x_1^3 + 3x_1^2 + x_2x_3 &= 1 \\ 4x_2^3 - 2x_2 + x_1x_3 &= -1 \\ 2x_3 + x_1x_2 &= 1. \end{aligned}$$

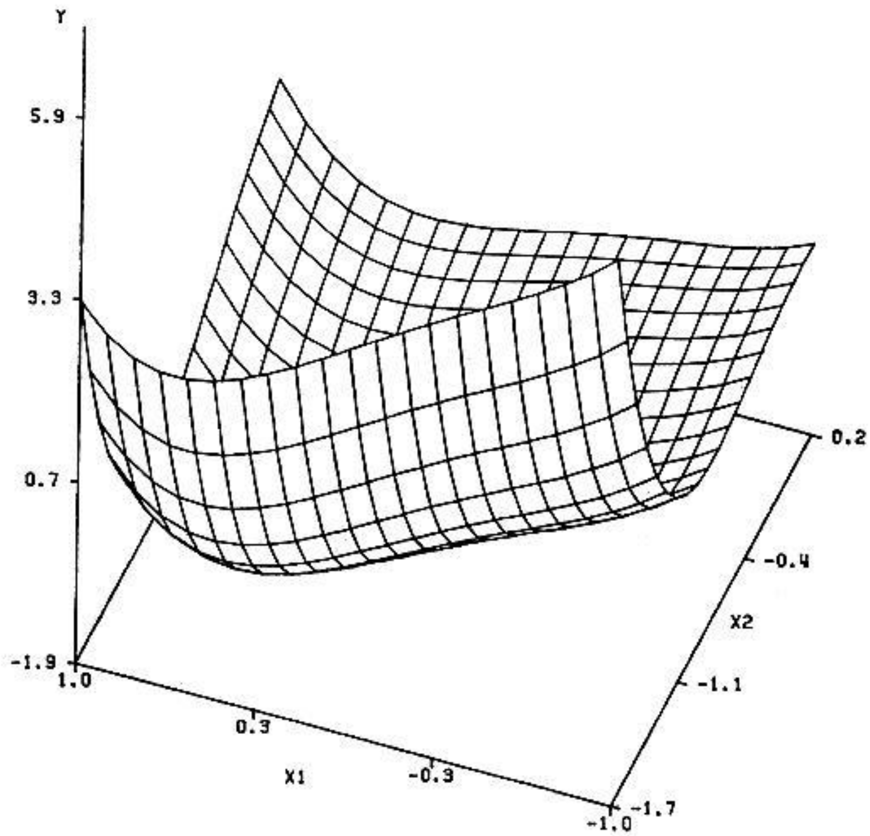
Едно такво нумеричко решение е

$$x_1^* = 0{,}571;\;\; x_2^* = -0{,}940;\;\; x_3^* = 0{,}768. \tag{2.2}$$

Бидејќи Хесеовата матрица е

$$\nabla^2 f(x) = \begin{bmatrix} 12x_1^2 + 6x_1 & x_3 & x_2 \\ x_3 & 12x_2^2 - 2 & x_1 \\ x_2 & x_1 & 2 \end{bmatrix}$$

позитивно дефинитна во точката (2.2) (ова може да се покаже со помош на Силвестеровата лема), се заклучува дека оваа точка е изолиран локален минимум. Функцијата од овој пример е прикажана на цртеж 2.1.



цртеж 2.1 Функција од пример 2.2 за фиксно $x_3 = 0,77$

Пример 2.3 Функцијата

$$f = (x_1 + x_2 - x_3)^2 + x_3^2$$

има бесконечно многу стационарни точки. Тоа се сите точки од облик

$$x_1^* = \alpha, x_2^* = -\alpha, x_3^* = 0 \tag{2.3}$$

каде α е произволен параметар. (На пример $x_1^* = 1, x_2^* = -1, x_3^* = 0$ или $x_1^* = 7, x_2^* = -7, x_3^* = 0$ итн, се стационарни точки.) Хесеовата матрица на функцијата f е позитивно семидефинитна за секое x. Затоа f е конвексна и (2.3) ги дава сите глобални минимуми на функцијата f.

3. Методи на Оптимизација

Постојат повеќе методи на оптимизација. Ние ќе споменеме повеќе од тие методи, но подетално ќе разгледаме само некои од нив.

Методите на споредување на вредности се типично еднодимензионални методи, т.е. овие методи неможат да се генерализираат на едноставен начин за проблеми со n променливи. За доволно „глатка“ функција (т.е. функции со неколку непрекинати изводи) се користи Њутновата метода. Оваа метода се одликува со брза конвергенција (кога конвергира). Поедноставување на Њутновата метода е Секантата методата. Многу популарни методи за наоѓање минимум се т.н. Методи на апроксимација на полином. Една од нив е Методата на парабола, која е формулирана за функции кои не мора да се диференцијабилни. Оваа метода како и Кубната метода, се сметаат за брзи и сигурни методи кога функцијата не е премногу комплицирана.

Кога се зборува за ефикасноста на нумеричките методи во нелинеарното програмирање за пресметување на оптималното решение x^* пожелно е за секоја метода да се знае нејзиниот. степен на конвергенција. Претпоставуваме (за функција која зависи од n променливи) дека методата е дадена преку векторот x^k кој тежнее кон оптималното решение x^*. Означуваме за секое $x = (x_i)$ во R^n:

$$\|x\| = \sqrt{\sum_{i=1}^{n} x_i^2}.$$

Ова е т.н. Евклидова норма (или должина) на векторот x. Ако постои број p и некое $a \neq 0$ така да

$$\lim_{k\to\infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|^p} = a$$

тогаш p ќе го наречеме степен на конвергенција на низата x^k, додека $\|x^k - x^*\|$ е грешка на k-та апроксимација. Ако $p = 1$, велиме дека степенот на конвергенција е *линеарен*; ако $p > 1$ велиме дека е *суперлинеарен.* Специјален случај на суперлинеарната конвергенција е $p = 2$; тогаш велиме дека степенот на конвергенција е *квадратен*. За некои методи и класи на функции степенот на конвергенција ќе биде секогаш линеарен, за други суперлинеарен. Се разбира, колку е степенот на конвергенција поголем, методата е поефикасна (побрза).

Во методите на споредување на вредностите важна улога играат и Фибоначиевите броеви. Тоа се броевите кои можат да се пишуваат како збир од претходните два броја, т.е.

$$F_1 = 1, F_2 = 1, \dots, F_i = F_{i-1} + F_{i-2}, i = 3,4, \dots$$

Првите неколку Фибоначиеви броеви се: 1,1,2,3,5,8,13,21,...

3.1 Метода на споредување на вредностите на фунцијата

Претпоставуваме дека f е реална функција со една променлива x. Нека x^* е (непознат глобален) минимум на функцијата f, за кој се знае дека лежи во некој интервал $[a_{k-1}, b_{k-1}]$, т. е. $a_{k-1} \leq x^* \leq b_{k-1}$ (види цртеж 3.1) Вредностите на функцијата f ќе ги пресметаме во точките y_k и z_k каде бараме да:

(1) y_k и z_k бидат еднакво оддалечени од крајните точки т.е.

$$y_k - a_{k-1} = b_{k-1} - z_k$$

(2) y_k биде во лева и z_k во десна половина на интервалот, т.е.

$$y_k = a_{k-1} + c_{k-1}(b_{k-1} - a_{k-1})$$

$$z_k = b_{k-1} - c_{k-1}(b_{k-1} - a_{k-1})$$

$$= b_{k-1} - a_{k-1} + a_{k-1} - c_{k-1}(b_{k-1} - a_{k-1}), \qquad (3.1)$$

по додавањето и одземањето на бројот a_{k-1}

$$= a_{k-1} + (1 - c_{k-1})(b_{k-1} - a_{k-1}) \qquad (3.2)$$

за некое $0 < c_{k-1} < \frac{1}{2}$.

Интервалот $[a_{k-1}, b_{k-1}]$ ќе го намалиме по споредувањето на вредностите на функцијата во точките y_k и z_k по следното правило:

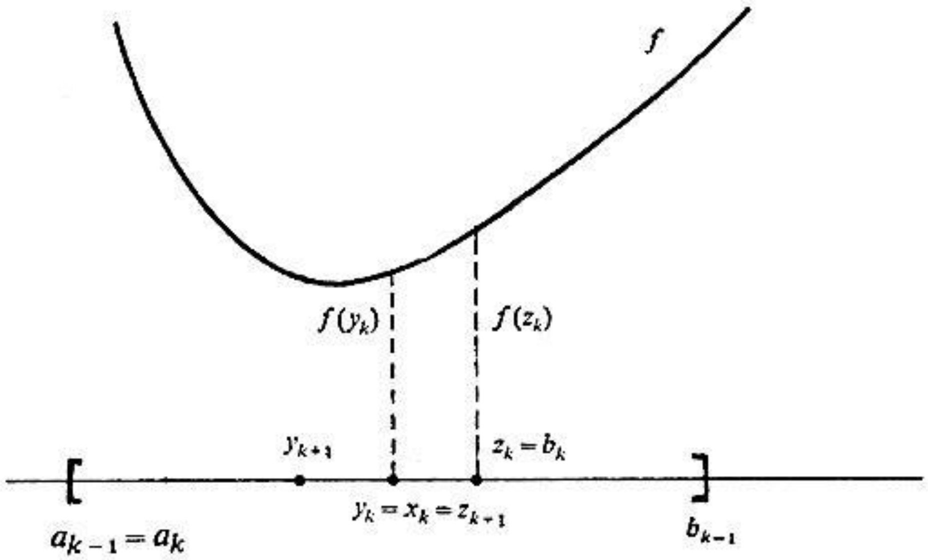
(3) Ако $f(y_k) \leq f(z_k)$, тогаш x^* не може да биде во интервалот $(z_k, b_{k-1}]$, т.е. тогаш мора да биде $a_{k-1} \leq x^* \leq z_k$. Во овој случај се одбира

$$a_k = a_{k-1},\ b_k = z_k,$$

и за апроксимација на решението x^* се зема $x_k = y_k$. (види цртеж 3.1) Меѓутоа, ако $f(y_k) > f(z_k)$, тогаш $y_k \leq x^* \leq b_{k-1}$ и понатаму се одбира

$$a_k = y_k,\ b_k = b_{k-1}$$

така да за апроксимацијата на решението x^* се зема $x_k = z_k$.



цртеж 3.1 Споредување на вредностите на функцијата f за $f(y_k) < f(z_k)$.

3.2 Метода на апроксимација на полином

Идејата на оваа метода се состои во следното: За да се одреди точката на минимум x^* на функцијата f, прво функцијата се апроксимира со некој едноставен полином $y(x)$ на интервалот I кој ја содржи точката x^*. Потоа се одредува точката на минимум x на функцијата $y(x)$. Бидејќи $y(x)$ ја апроксимира $f(x)$, точката x ја апроксимира x^*. Интервалот I сега се намалува, функцијата f се апроксимира со новиот полином и постапката се повторува додека не се постигне посакуваната точност на апроксимацијата. Функцијата f обично се апроксимира со полиноми од втор или трет степен кога зборуваме за Метода на парабола или Кубна метода (метода од трет степен).

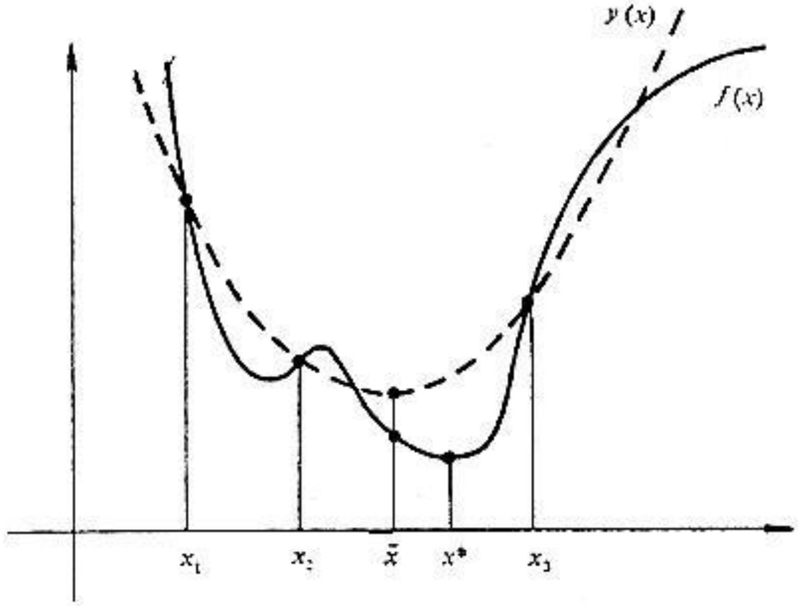
Прво ќе ја објасниме ***Методата на парабола***. Во првиот чекор треба да се пронајдат три точки

$$x_1 < x_2 < x_3$$

со особини

$$f(x_1) > f(x_2) \text{ и } f(x_2) < f(x_3). \tag{3.3}$$

(Постојат разни стратегии за наоѓање на овие точки. Една стратегија се состои од фиксирање на $x = x_0$ и пресметување на вредностите на функцијата f во точките $x_0 + h, x_0 + 2h, x_0 + 3h$ итн. за некое h.) Кога овие сите три точки се пронајдени, тогаш локалниот минимум x^* на функцијата f се наоѓа негде на интервалот помеѓу x_1 и x_3, т.е. $x_1 \leq x^* \leq x_3$. (види цртеж 3.2)



цртеж 3.2 Метода на парабола

Трите точки x_1, x_2, x_3 со своите вредности на функциите $f(x_1), f(x_2), f(x_3)$ ја одредуваат квадратната апроксимација (т.е. параболата)

$$y(x) = a + bx + cx^2. \quad (3.4)$$

Коефициентите a, b, c се добиваат со решавање на системот на линеарни алгебарски равенки

$$\begin{aligned} a + x_1 b + x_1^2 c &= f(x_1) \\ a + x_2 b + x_2^2 c &= f(x_2) \\ a + x_3 b + x_3^2 c &= f(x_3). \end{aligned} \quad (3.5)$$

Минимумот на функцијата $y(x)$ следи од

$$y'(x) = b + 2cx = 0$$

и се наоѓа во точката

$$\tilde{x} = -\frac{b}{2c}.$$

Точката $\tilde{x}$ е апроксимација на оптималната точка x^*. Сега постојат четири точки: „старите“ точки x_1, x_2, x_3 и „новата“ точка $\tilde{x}$. Меѓутоа, за да се одреди нова квадратна апроксимација ќе се користат само три точки, што значи дека една точка е вишок. Од горните четири точки во следната итерација со x_2 ќе ја означиме точката во која функцијата f има најмала вредност (x_2 може да биде само $\tilde{x}$ или „старата“ точка x_2). Соседната точка која се наоѓа лево од „новата“ точка x_2 ќе ја означиме со x_1, а соседната точка десно од „новата“ точка x_2 со x_3. За вака означените нови три точки повторно важи релацијата (3.3) и постапката се поновува, т.е. одредуваме нова парабола (3.4), нејзина точка на минимум $\tilde{x}$ итн. (Во проблемот која го опишува цртеж 3.2, после првата итерација точката x_1 е вишок, x_2 постанува x_1, $\tilde{x}$ постанува x_2, додека x_3 си останува x_3.)

Забелешка:

1. Теоретски е можно произволно одбрана точка x_2 на почетокот на процедурата на барање на оптимумот да е точката на минимумот на параболата $y(x)$. Во тој случај Методата на парабола влегува во „бесконечна јамка“, т.е. секогаш го дава истото $\tilde{x} = x_2$. За да се излези од „јамката“, треба да се промени вредноста на точката x_2 (на пример треба да се замени x_2 со $x_2 + \varepsilon$, каде ε е некој мал позитивен број, на пример $\varepsilon = 10^{-5}$).

2. Во методата на парабола (за разлика од Њутновата метода) не користиме извод на функцијата f. Затоа методата може да се користи за наоѓање на оптималните точки на недиференцијабилни функции.

3. Постапката се прекинува кога две последователни апроксимации x се доволно близу една до друга или кога е $|f(\tilde{x}) - y(\tilde{x})|$ доволно мало.

3.3 Алгоритам (метода на парабола)

1. Се специфицира правилото на стопирање, т.е. се одбира позитивен број ε во однос на кој алгоритмот ќе се прекине кога $|f(\tilde{x}) - y(\tilde{x})| \leq \varepsilon$. (Точката $\tilde{x}$ тогаш е прифатена за апроксимација на оптималната точка x^*.)

2. Се одредуваат произволни точки $x_1 < x_2 < x_3$ за кои ќе бидат исполнети условите $f(x_1) > f(x_2)$ и $f(x_2) < f(x_3)$.

3. Се пишува и решава системот (3.5), и се означуваат со a, b, c неговите решенија.

4. Се пресметува

$$\tilde{x} = -\frac{b}{2c}$$

и $y(\tilde{x}) = a + b\tilde{x} + c\tilde{x}^2$ и $f(\tilde{x})$.

5. Ако $|f(\tilde{x}) - y(\tilde{x})| \leq \varepsilon$ е процес на застанување, $\tilde{x}$ е „добра“ апроксимација на локалниот минимум x^*. Ако $|f(\tilde{x}) - y(\tilde{x})| > \varepsilon$ го означиме со x_2 она од четирите точки во која функцијата f има најмала вредност, за x_1 соседната точка лево, а x_3 за соседна точка десно од „новата“ точка x_2. Со овие „нови“ точки x_1, x_2, x_3 се враќаме во чекор 3. и го повторуваме алгоритамот.

Алгоритамот 3.3 ќе го прикажеме преку пример.

Пример 3.1 Користејќи ја методата на парабола потребно е да се одреди оптималната точка на функцијата

$$f(x) = 2x^4 - 3x.$$

1. Нека е прифатена точноста одредена со $\varepsilon = 10^{-7}$.
2. Во точките $x_1 = 0, x_2 = 1, x_3 = 2$ вредностите на функцијата f се $f(0) = 0$, $f(1) = -1$, $f(2) = 26$ што значи дека важи

$$f(x_1) > f(x_2) \text{ и } f(x_2) < f(x_3)$$

како што се бара на почетокот на методата.

3. Системот (3.5) за овие три точки е

$$a = 0$$
$$b + c = -1$$
$$b + 2c = 13.$$

Решението е

$$a = 0, b = -15, c = 14.$$

(Значи, функцијата f е апроксимирана со параболата $y = 14x^2 - 15x$).

4. Оптималната точка на параболата, нејзината оптимална вредност и вредноста на функцијата f во таа точка, се:

$$\tilde{x} = -\frac{b}{2c} = \frac{15}{28} \approx 0{,}53571$$
$$y(\tilde{x}) = -4{,}01786, f(\tilde{x}) = -1{,}44242.$$

5. Бидејќи $|f(\tilde{x}) - y(\tilde{x})| > 10^{-7}$, апроксимацијата $\tilde{x} = 0{,}53571$ не е прифатена како решение на проблемот. Наоѓаме да $f(\tilde{x}) < f(x_i),\ i = 1,2,3,$ затоа земаме $x_2 = \tilde{x}$, што одредува

$$x_1 = 0, x_2 = \tilde{x} = 0{,}53571, x_3 = 1.$$

Со овие точки се враќаме во чекор 3. и продолжуваме со пресметките.

3. Системот (3.5) за новите три точки сега е

$a = 0$

$0{,}53571b + 0{,}28699c = -1{,}4424$

$b + c = -1;$

неговото решение е

$$a = 0, b = -4{,}64541, c = 3{,}64541.$$

(Значи, новата парабола е $y = -4{,}64541x + 3{,}64541x^2$.)

4. $\tilde{x} = -\frac{b}{2c} = 0{,}63816$

$y(\tilde{x}) = -1{,}47993, f(\tilde{x}) = -1{,}58185.$

Бидејќи сеуште важи $|f(\tilde{x}) - y(\tilde{x})| > 10^{-7}$, апроксимацијата $\tilde{x} = 0{,}63716$ не е прифатена и треба да се изврши замена

$$x_1 = 0{,}53571, x_2 = \tilde{x} = 0{,}63716, x_3 = 1.$$

Повторно се враќаме на чекор 3. и продолжуваме со пресметувањето.

Резултатите од пресметките се прикажани во **табела 3.1.**

Табела 3.1.

Iteracija	x_1	x_2	x_3	$\tilde{x}$	$\lvert f(\tilde{x})-y(\tilde{x})\rvert$
1	0	1	2	0,53571	2,57544
2	0	0,53571	1	0,63716	0,10192
3	0,53571	0,63716	1	0,69358	0,01564
4	0,63716	0,69358	1	0,70903	0,00196
5	0,69358	0,70903	1	0,71680	0,00032
6	0,70903	0,71680	1	0,71932	0,00005
7	0,71680	0,71932	1	0,72045	$7{,}3 \cdot 10^{-6}$
8	0,71932	0,72045	1	0,72085	$1{,}1 \cdot 10^{-6}$
9	0,72045	0,72085	1	0,72102	$2{,}0 \cdot 10^{-7}$
10	0,72085	0,72102	1	0,72108	$2{,}6 \cdot 10^{-7}$

Бидејќи после десетата итерација важи $|f(\tilde{x}) - y(\tilde{x})| < 10^{-7}$, постапката ја прекинуваме. Апроксимацијата на локалниум минимум е $\tilde{x} = 0{,}72108$.

Сега ќе ја опишеме Кубната метода (методата од трет степен). Функцијата f се апроксимира во оваа метода со помош на полином од трет степен

$$y(x) = a + bx + cx^2 + dx^3. \tag{3.6}$$

Коефициентите a, b, c, d се одредуваат од системот со четири линеарни равенки. За разлика од Методата на парабола, тука претпоставуваме дека функцијата f е диференцијабилна. Наместо $y(x)$ да поминува низ четири точки, бараме $y(x)$ да поминува низ само две точки x_1, x_2, но во овие точки поставуваме барање во однос на изводот $y'(x)$:

$$\begin{aligned} f(x_1) &= a + bx_1 + cx_1^2 + dx_1^3 \\ f(x_2) &= a + bx_2 + cx_2^2 + dx_2^3 \\ f'(x_1) &= b + 2cx_1 + 3dx_1^2 \\ f'(x_2) &= b + 2cx_2 + 3dx_2^2. \end{aligned} \tag{3.7}$$

Првите две точки x_1 и x_2, каде е $x_1 < x_2$, нека го задоволуваат следниот услов:

$$f'(x_1) < 0, f'(x_2) > 0. \tag{3.8}$$

Ова гарантира дека точката на минимум x^* на функцијата f се наоѓа помеѓу x_1 и x_2.

Точките x_1 и x_2 со горните својства можат да се најдат со истата постапка како и кај Методата на парабола. Во точките x_1 и x_2, за кои важи (3.8), се формира и решава системот (3.7). Решението a, b, c, d се заменува во (3.6), и како такво постанува апроксимација на функцијата f. После тоа се одредува минимална точка x на функцијата $y(x)$ на интервалот $[x_1, x_2]$. Оваа точка лесно се пресметува од квадратната равенка $y'(x) = b + 2cx + 3dx^2 = 0$. Точката $\tilde{x}$ ја апроксимира x^*. Сега во точката $\tilde{x}$ се пресметува $f(\tilde{x})$ и $f'(\tilde{x})$ после што се намалува интервалот кој ја содржи x^* на овој начин: Ако е $f'(\tilde{x}) < 0$, „старата“ точка x_1 се отфрла и се користи како „нова“ точка $x_1 = \tilde{x}$ и x_2. Од друга страна, ако е $f(\tilde{x}) > 0$, тогаш „старата“ точка x_2 се отфрла и се користи како „нова“ точка x_1 и $x_2 = \tilde{x}$. Во „новите“ точки x_1 и x_2 се изразува и решението на системот (3.7),и се добива нов полином (3.6) чија е точката на минимум $\tilde{x}$ и нова апроксимација на точката x^*, итн. Критериумот на стопирање на итерационата процедура е ист како и кај методата на парабола.

3.4 Алгоритам (Кубна метода)

1. Се специфицира правилото за стопирање на алгоритамот, т.е. се избира доволно мал произволен број $\varepsilon > 0$.

2. Се одредуваат две произволни точки x_1 и x_2 кои ги задоволуваат условите

$$x_1 < x_2,\ f'(x_1) < 0, f'(x_2) > 0$$

(или $f(x_1) < f(x_2)$ наместо $f'(x_2) > 0$).

3. Се пишува решението на системот (3.7), и се означува со a, b, c, d тоа решение.

4. Се пресметува точката на минимум $\tilde{x}$ на полиномот

$$y(x) = a + bx + cx^2 + dx^3$$

на интервалот $[x_1, x_2]$, а потоа и $y(\tilde{x})$ и $f(\tilde{x})$.

5. Ако е $|f(\tilde{x}) - y(\tilde{x})| \leq \varepsilon$ процесот се прекинува, $\tilde{x}$ е прифатлива апроксимација на локалниот минимум x^*. Ако е $|f(\tilde{x}) - y(\tilde{x})| > \varepsilon$, алгоритамот продолжува. Во тој случај се пресметува $f'(\tilde{x})$. Ако $f'(\tilde{x}) < 0$, означуваме со $x_1 = \tilde{x}$ и за точките x_1 и „старата“ точка x_2 се враќаме на чекор 3. Ако е $f'(\tilde{x}) > 0$, означуваме со $x_2 = \tilde{x}$ и за „старата“ точка x_1 и „новата“ точка x_2 се враќаме на чекор 3.

Забелешка:

1. Во кубната метода, оптималната точка x^* на функцијата f секогаш се наоѓа помеѓу тековните точки x_1 и x_2, т.е. во секоја итерација имаме

$$x_1 \leq x^* \leq x_2.$$

2. Кубната метода обично конвергира побрзо од методата на парабола, но бара повеќе операции за пресметување.

4. Методи за безусловна оптимизација на диференцијабилни функции

Проблематиката која се разгледува во овој дел се однесува на наоѓање на точка на локален оптимум на дадена целна функција при што не се зададени ограничувања.

Методите на безусловна оптимизација на функција која зависи од n променливи кои не користат извод на целната функција во принцип не се брзи методи, но нашле примена во оптимизација на недиференцијабилни функции, како и кај функции кои што не се непрекинати во целата своја област на дефинираност.

Методите за безусловна оптимизација на диференцијабилните функции f со n променливи, можат грубо да се поделат на две класи. Во првата спаѓаат методите кои го користат само првиот извод на f и се нарекуваат ***Методи од прв ред***. Типична и најстара метода од прв ред е ***Кошиевата метода за најстрмно опаѓање*** која потекнува од 1847 година. Методата има линеарен степен на конвергенција и се одликува со добар напредок кон точката на минимумот x^*, но има спора конвергенција во близина на оптималната точка. Во втората класа спаѓаат методите кои го користат покрај првиот и вториот извод (или некоја нивна апроксимација) и се нарекуваат ***Методи од втор ред*** кои што се побрзи методи. Њутновата метода е класичен претставник на овие методи. Њутновата метода се карактеризира со суперлинеарна конвергенција. Кога конвергира, таа е побрза од Кошиевата метода. Но, Кошиевата метода иако е поспора сепак е посигурна, додека Њутновата метода е побрза но е понесигурна. Нормално, тоа многу зависи од функцијата f, начинот на кои векторите се нормализираат и стратегијата која се применува во методата.

4.1. Кошиева метода за најстрмно опаѓање

Кошиевата метода, го користи следниот добро познат факт за диференцијабилна функција f

> Градиентот $\nabla f(x^0)$ на функцијата f во произволна точка x^0 ги анулира сите вектори во множеството $S = \{x: f(x) = f(x^0)\}$ во точката x^0. $\nabla f(x^0)$ ја дава насоката на најголемиот пораст на функцијата f во околина на точката x^0.

Како сите нумерички методи за нелинеарна оптимизација и Кошиевата метода е итеративна. Прво се специфицира некоја произволна апроксимација x^0, потоа во точката x^0 се одредува градиентот $\nabla f(x^0)$. Градиентот ја покажува насоката на најбрзиот пораст на функцијата f во околина на точката x^0. Доколку сакаме да го одредиме минимумот (а не максимумот) на функцијата f, важна ни е насоката за најбрзо опаѓање на функцијата f. Ова се одредува со негативен градиент $-\nabla f(x^0)$. Се решава проблем од еднодимензионална оптимизација

$$Min\, f(x^0 - \sigma[\nabla f(x^0)]^T) \qquad (4.1)$$

Јасно е дека $f(x^0 - \sigma[\nabla f(x^0)]^T)$ е функција од една променлива σ која можеме да ја обележиме како $F(\sigma)$, т.е

$$F(\sigma) = f(x^0 - \sigma[\nabla f(x^0)]^T).$$

Затоа програмот (4.1) може да се напише како

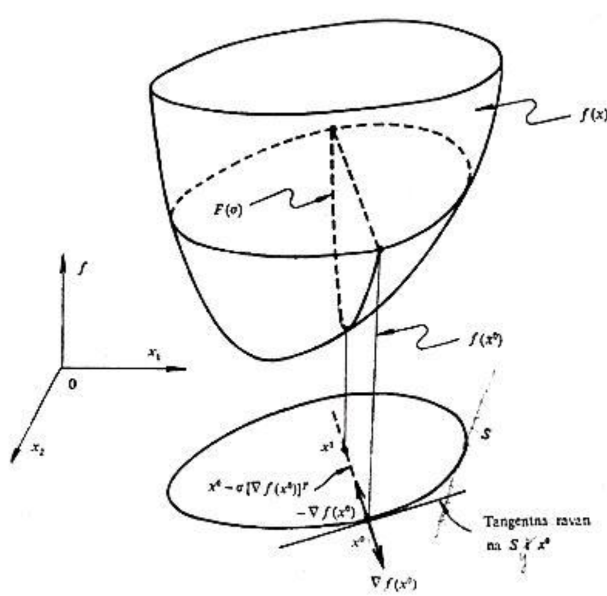
$$Min\, F(\sigma).$$

$$p.o.$$

$$\sigma > 0.$$

Функцијата $F(\sigma)$ се одредува со вредностите кои f ги добива во насока на $f(x^0 - \sigma[\nabla f(x^0)]^T)$. Со други зборови, $F(\sigma)$ е пресек на “површината” f со рамнината која е вертикална на рамнината (x_1, x_2) и која поминува низ

$$x^0 - \sigma[\nabla f(x^0)]^T).$$



цртеж 4.1. Идеа за Кошиева метода.

Претпоставувајќи дека градиентот ∇f е редица (според дефиницијата во поглавје 1.2), додека останатите вектори се колони, мораме градиентот да го транспонираме за да аргументот на функцијата f во (4.1) има смисла. Претпоставуваме дека проблемот (4.1) сме го решиле со некоја од методите за еднодимензионална оптимизација и дека сме добиле некое локално решение $\sigma = \sigma_0$. Бројот σ_0 е должина на почетниот "чекор" во насока $-\nabla f(x^0)$. Сега точката

$$x^1 = x^0 - \sigma[\nabla f(x^0)]^T)$$

ја земаме како нова апроксимација и го повторуваме истиот начин но, сега за x^1.

Забелешка: На неколку работи треба да се посвети внимание: Прво, насоката на градиентот не се менува ако градиентот се "нормализира", т.е. ако наместо векторот $[\nabla f(x^k)]^T$ ги користиме векторите

$$u^k = \frac{[\nabla f(x^k)]^T}{\|[\nabla f(x^k)]^\mathrm{T}\|}, k = 0,1,2, \dots$$

Можеме да видиме дека u^k има еднодимензионална должина, т.е. $\|u^k\| = 1,\ k = 0,1,2 \dots$. Нормализацијата обично допринесува за стабилноста на методот. Второ, во решавање на еднодимензионалните проблеми (4.1) во k-та итерација, може да се случи функцијата $F(\sigma)$ да има неколку локални минимуми. Во тој случај треба да се специфицира кои од нив треба да се задржат. За да се задржи контролата над нумеричките операции, во тој случај по договор може да се земе локалниот минимум кој е најблиску до решението. Трето, Кошиевата метода, обично започнува со многу "скокови" и со добро напредување кон оптималното решение, но, брзо се успорува. Ова успорување може да се открие со помош на некој критериум, како следниот

$$\|\sigma_k u^k\| \le \|K\sigma_0 u^0\| \tag{4.2}$$

т.е

$$\sigma_k \le K\sigma_0. \tag{4.3}$$

(Векторите u^k се нормализирани, броевите σ_k се позитивни), затоа (4.2) и (4.3) се една иста релација. Бројот K е константа, $0 < K < 1$, која е специфицирана пред почетокот на решавање. Критериумот (4.3), може да послужи и како правило за застанување, т.е методата престанува кога должината на чекорите ќе стане помала од некој однапред зададен број (на пример една стотинка) од почетниот чекор. Нормално, должината на почетниот чекор σ_0 не можеме (и не треба) да ја одредиме. Како второ правило за застанување можеме да користиме "должината" на градиентот да биде помала од некој однапред специфициран број $\varepsilon > 0$, т.е

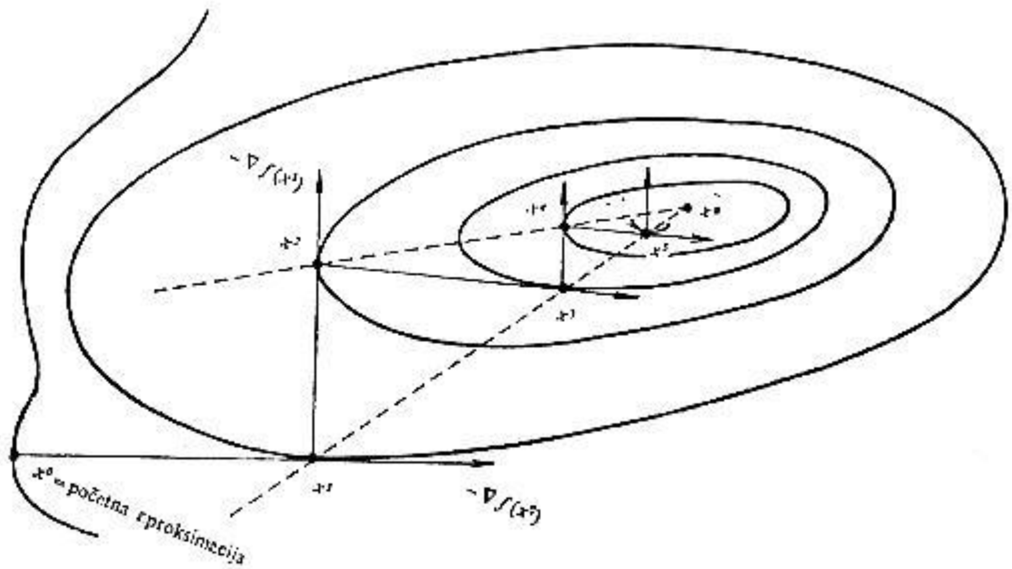
$$\|[\nabla f(x^k)]^\mathrm{T}\| < \varepsilon. \tag{4.4}$$

Ако Кошиевата метода почне да го покажува знакот на успорување додека должината на градиентот уште е голема, можеме пресметувањето да го продолжиме со некоја друга метода (да речиме Њутновата) метода.

Ортогоналност на градиентот. Ова својство претставува, битно својство за методот на најстрмно опаѓање, пресметано во две произволни последователни апроксимации, меѓусебно вертикални, т.е

$$\nabla f(x^{k+1})[\nabla f(x^k)]^T = 0, k = 0,1,2, \dots \tag{4.5}$$

Типична конвергенција на Кошиевата метода за функцијата f со две променливи прикажана е на цртеж 4.2



цртеж 4.2 Конвергенција на Кошиевата метода.

4.2. .1 Алгоритам (Кошиева метода)

1. Се специфицира почетната апроксимација x^0 и правилото за застанување, на пример бројот $0 < K < 1$, т.ш $\sigma_k \leq K\sigma_0$, каде σ_0 е должина на почетниот чекор, а σ_k е должината на наредните чекори за $k = 1,2,........$.
2. Се пресметува $f(x^k)$ во точката x^k.
3. Се нормализира градиентот, т.е се пресметува

$$u^k = \frac{[\nabla f(x^k)]^T}{\|[\nabla f(x^k)]^{\mathrm{T}}\|}$$

4. Се решава проблемот за одредување на функцијата f од точката x^k во насока на $-u^k$ т.е се пресметува.

$$\min_{\sigma>0} f(x^k - \sigma u^k).$$

Се означува оптималното решение со σ_k.

5. Се одредува новата точка $x^{k+1} = x^k - \sigma_k u^k$.

6. Ако е $\sigma_k \leq K\sigma_0$, процесот застанува, методата го дава приближното оптимално решение $x^* = x^{k+1}$. Ако е $\sigma_k > K\sigma_0$, продолжуваме со пресметување и се враќаме на чекорот 2, за x^{k+1}.

Пример 4.1 Ја применуваме Кошиевата метода над функцијата

$$f(x) = x_1^4 + x_1^3 - x_1 + x_2^4 - x_2^2 + x_2 + x_3^2 - x_3 + x_1 x_2 x_3.$$

Пресметката ќе ја завршиме кога ќе биде исполнето правилото за застанување со вредностите $K = 0{,}1$, т.е.

$$\sigma_k \leq 0{,}1\sigma_0$$

Во нормализирањето на векторите u^k се користи *Чебишевата должина*. (Чебишевата должина, за векторот $z = (z_i)$, е број $\|z\| = \max\{|z_1|, \dots, |z_n|\}$). Земајќи дека $x^0 = (0{,}0{,}0)^T$ и $\nabla f(x^0) = [-1{,}1, -1]$, гледаме дека $\|[\nabla f(x^0)]^{\mathrm{T}}\| = 1$ и затоа е

$$u^0 = \frac{[\nabla f(x^0)]^T}{\|[\nabla f(x)]^T\|} = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}.$$

Функцијата f ќе ја одредуваме во следната насока

$$x^0 - \sigma u^0 = \begin{bmatrix} \sigma \\ -\sigma \\ \sigma \end{bmatrix},\ \sigma > 0$$

земајќи дека

$$F(\sigma) = f(x^0 - \sigma u^0) = f(\sigma, -\sigma, \sigma) = 2\sigma^4 - 3\sigma$$

минимумот на оваа функција (пресметан со Методот на парабола) изнесува

$$\sigma_0 = 0{,}71308$$

Пресметаната прва апроксимација изнесува

$$x^1 = x^0 - \sigma u^0 = \begin{bmatrix} 0{,}71308 \\ -0{,}71308 \\ 0{,}71308 \end{bmatrix}.$$

Вредностите на функцијата f во точките x^0 и x^1 се $f(x^0) = 0$ и $f(x^1) = -1{,}62213$. Значи се работи за “голем скок”. Пресметката сега почнува од x^1. За да се исполни правилото на застанување потребно е да се пресметаат 13 итерации. Реаултатите се прикажани во табела 4.1.

Табела 4.1

k	x_1^k	x_2^k	x_3^k	$f(x^k)$
1	0,71308	−0,71308	0,71308	−1,62213
2	0,52536	−0,90927	0,72361	−1,90097
3	0,56270	−0,93966	0,72774	−1,91020
4	0,56916	−0,93058	0,74999	−1,91122
5	0,56759	−0,93946	0,75407	−1,91158
6	0,57072	−0,93658	0,76153	−1,91170
7	0,56969	−0,93925	0,76310	−1,91175
8	0,57080	−0,93848	0,76578	−1,91176
9	0,57044	−0,93954	0,76634	−1,91177
10	0,57083	−0,93917	0,76731	−1,91177
11	0,57070	−0,93955	0,75751	−1,91177
12	0,57085	−0,93942	0,76786	−1,91177
13	0,57080	−0,93955	0,76794	−1,91177

Чебишевата должина на градиентот во точката x^{13} е помала од 0,001. Ова значи дека x^{13} е стационарна точка (во која градиентот мода да биде еднаков на нултиот вектор и затоа неговата должина е еднаква на нула). Земајќи дека f е строго конвексна функција во околината на точката x^{13} заклучуваме дека x^{13} апроксимира локален минимум.

Кошиевата метода конвергира во стационарната точка на функцијата f под многу ниски услови:

Ако f е непрекинато диференцијабилна функција, и ако

$$M = \{x: f(x) \leq y\} \text{ за некој број } y$$

е ограничено, и почетното x^0 е земено во множеството M, тогаш Кошиевиот метод е со најстрмно опаѓање

$$\lim_{k\to\infty} \nabla f(x^k) = 0.$$

***Физичка интерпретација*.** Кошиевиот метод може лесно да се даде со едноставна физичка интерпретација. Патувањето на честичките со маса m во потенцијално поле $f(x)$ опишано е (како функција од времето t) со следниот систем од линеарни диференцијални равенки:

$$m\frac{d^2x}{dt^2} + r\frac{dx}{dt} + [\nabla f(x)]^T = 0, x = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}$$

Каде r е коефициент на триење. После апроксимацијата изводот е опишан со следниот израз

$$m\frac{d^2x}{dt^2} + r\frac{dx}{dt} + [\nabla f(x)]^T = 0.$$

Во точката x^k можеме да напишеме

$$\Delta x = x^{k+1} - x^k$$
$$\Delta^2 x = x^{k+1} - 2x^k + x^{k-1}$$

доколку $\Delta t = 1$, се добива:

$$m\left(x^{k+1} - 2x^k + x^{k-1}\right) + r\left(x^{k+1} - x^k\right) + [\nabla f\left(x^k\right)]^T = 0.$$

Кога масата е занемарливо мало m, имаме

$$x^{k+1} = x^k - \frac{1}{r}[\nabla f\left(x^k\right)]^T.$$

Заклучок: Честичката со занемарливо мала маса, патува низ точката x^k во правец на најстрмното опаѓање на потенцијалното поле $f(x)$.

5. Нелинеарно програмирање

Нелинеарното програмирање (НП) спаѓа во група на динамички методи за решавање на една голема класа статичко управувачки задачи. Секоја управувачка задача кај која целната функција $F(X)$ и множество ограничувања се дефинирани со нелинеарни зависности (целната функција со нелинеарна функција, а множеството ограничувања со нелинеарни алгебарски равенства или неравенства), се сведува на задача од НП, чие оптимално решение се пронаоѓа со некоја од погодните методи, која е најадекватна за пронаоѓање на конкретното решение.

Задачите на НП, за разлика од задачите на ЛП (линеарното програмирање), не може да се решаваат со примена на некои универзални методи, како што е симплекс методата за задачите на ЛП, туку за секој конкретен случај, во зависност од неговиот математички модел, димензијата и карактерот на нелинеарноста, потребна е нова метода или прилагодување на некоја од веќе постоечките. Во поголемиот број случаи дури и да не постои соодветна метода врз база на која ќе се пронајде оптималното решение на формулираната задача на НП, што значи дека постојат голем број на нерешливи или тешко решливи задачи на НП. Заради тоа во оваа област на управувачки задачи како во подрачјето на методологијата, така и во подрачјето на поставените соодветни математички модели на задачите на НП постоеле обемни истражувања, кои имале за цел, ако ништо друго, само да се употреби методологијата, систематизацијата и селекцијата на решливите задачи да се создадат услови за поширока примена на методата на оптимизација во решавањето на оние управувачки задачи чии математички модели се нелинеарни.

Сега веќе постојат повеќе методи на оптимизација со чија помош може да се решат некои задачи на НП. Сите тие методи се развиени во текот на последниве 15 години и се специјализирани за различни типови на задачи на НП, кои формално се разликуваат по обликот на математичкиот модел т.е. по обликот и димензијата на целната функција и множеството ограничувања. Така, на пример, постојат специјални методи за линеарни ограничувања, за квадратен облик на целна функција, за целобројни вредности на променливите итн. Оттука потекнуваат и некои посебни имиња за таквите специфични задачи на НП, како што се: квадратно програмирање, целобројно програмирање, итн.

5.1 Поставување на задачата

Општата формулација на задача на НП може да се искаже на следниов начин: Најди ја вредноста на $n-$димензионалниот вектор

$$X=(x_1, x_2, \ldots, x_n),$$

за кој целната функција

$$F(X), \tag{5.1}$$

добива максимална (минимална) вредност, а при тоа да бидат задоволени ограничувања

$$G(X) \leq 0, \tag{5.2}$$
$$X \geq 0,$$

каде $G(X)$ е $m-$ димензионален вектор на функцијата чии компоненти се $g_1(X), g_2(X),..., g_m(X)$.

Математичкиот модел на општата задача на НП (5.1)-(5.2), која е напишана во најопшт облик, подразбира одредување на оние вредности

$x_1, x_2,..., x_n$ за кои целната функција

$$F(x_1, x_2,..., x_n),$$

добива максимална (минимална) вредност при ограничувањата

$$g_i(x_1, x_2,..., x_n) \leq 0, \qquad (i = 1,2,..., m),$$
$$x_j \geq 0, \qquad (j = 1,2,..., n).$$

Често, вообичаено е равенствата во изразите (5.2) да се нарекуваат услови, а неравенствата ограничувања. Индексите m и n меѓусебно се независни т.е. m може да биде помало, еднакво или поголемо од n. Функциите $F(X)$ и $g_i(X)$, $(i = 1,2,..., m)$, во општ случај се нелинеарни функции, па оттука и името нелинеарно програмирање.

Посебни случаи на задачи на НП се јавуваат кога истовремено F и G се нелинеарни функции, т.е. кога е само F или G нелинеарна функција. Врз таа основа, врзувајќи се за обликот на функцијата $F(X)$ и $G(X)$, се врши формална класификација на задачите на НП, со цел да се избегне повисокото ниво на апстракција.

Ќе се најават некои од тие класификации, кои во прв план ги истакнуваат решливите задачи на НП и нивната општа формулација.

а) НП со линеарно множество ограничувања

Во оваа класа задачи на НП множеството ограничувања се задава со функциите $g_i \ (i = 1,2,\ldots,m)$ кои се линеарни , т.е.

$$g_i: \sum_{i=1}^{n} a_{ij} x_j \leq b_i, \quad (i = 1,2,\ldots,m),$$

$$x_j \geq 0, \qquad (j = 1,2,\ldots,n).$$

б) *НП со сепарабилна целна функција* $F(X)$

Карактеристика на оваа класа задачи на НП се состои во тоа што, нивната целна функција се дефинира со збирот на $n-$функции, од кои секоја зависи само од една променлива, т.е.

$$F(X) = \sum_{j=1}^{n} f_j(x_j)$$

в) *Квадратно програмирање*

Оваа класа задачи на НП се карактеризира со квадратна форма на целната функција $F(X)$, која се искажува во обликот

$$F(X) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_i x_j.$$

Специјална подкласа на задачите на квадратното програмирање ја сочинуваат задачите кај кои множеството ограничувања е линеарно. Задачите од овој вид, кај кои целната функција е зададена во квадратна форма, каде множеството ограничувања е зададено со линеарни равенства и неравенства, спаѓаат во групата на релативно лесно решливи задачи.

г) *Целобројно програмирање*

Задачите на НП, кои покрај задоволувањето на условите (5.1) и (5.2) мораат да задоволат посебни услови за сите променливи да може да ги земаат само целобројните вредности, спаѓаат во класата на задачи кои се изучуваат под името целобројно програмирање.

Како специјален случај на оваа класа на задачи се јавуваат оние задачи, кај кои променливите може да примаат само две вредности: нула или еден. Тоа се задачи кои се изучуваат во рамки на посебно име: **0-1 програмирање**. Со оглед на тоа дека овие задачи од оваа врста често се среќаваат во праксата, на нив ќе посветиме посебно внимание.

Оттука следува еден од општите заклучоци, дека за секоја класа задачи на НП се дефинираат услови кои го одредуваат карактерот на функциите $F(X)$ и $G(X)$, што е направено за претходните четири случаи.

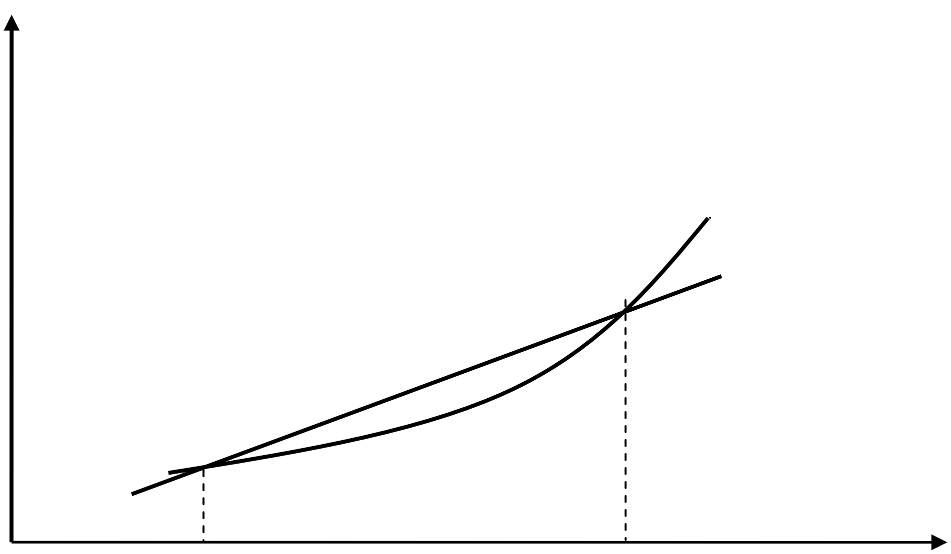
До сега, според останатото што е напоменато, развиени се ефикасни алгоритми за максимизација (минимизација) на конкавни (конвексни) функции $F(X)$ во конкавна (конвексна) област (D), која е ограничена со множество ограничувања од обликот $g_i(X) \geq 0$, $(i = 1,2,...,m)$, каде $g_i(X)$ се исто така конкавни (конвексни) функции.

Максимизација (минимизација) на целната функција $F(X)$, при множество ограничувања $g_i(X) \geq 0$, $(i = 1,2,...,m)$, ја претпоставува егзистентијата на почетниот план

$$X^{(0)} = \left(x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)}\right),$$

за кој е задоволено множеството ограничувања $g_i\left(X^{(0)}\right) \geq 0$,

$(i = 1,2,...,m)$.



Цртеж 5.1.Графичка интерпретација на конвексна функција $F(X)$

За конвексни функции задачата на НП се формулира во облик (5.1) - (5.2), каде се бара минимумот на целната функција $F(X)$. Меѓутоа, за конкавна целна функција и множество ограничувања, задачата на НП се формулира со посредство на максимизација на целната функција $F(X)$ при множество ограничувања $g_i(X) \geq 0$, $(i = 1,2,...,m)$. Сведувањето на задачата од (5.1) - (5.2) на задача за максимизација на целната функција

$$F(X), \qquad (5.1')$$

при множеството ограничувања

$$X \geq 0, \qquad (5.2')$$

како и сведување на задачата од (5.1')-(5.2') на задачата од (5.1)-(5.2), не претставува тешкотија, со оглед на тоа дека и во едниот и во другиот случај потребно е да се изврши промена на знакот на функцијата, т.е. наместо $F(X)$ потребно е да стави $-F(X)$, односно наместо $G(X)$ да се стави $-G(X)$ и да се промени знакот во неравенството на множеството ограничувања. За случај со обратни барања се постапува на ист начин.

Спрема тоа, сеедно е дали зборуваме за задачата (5.1) - (5.2) или за задачата (5.1') - (5.2'). Таа особина често ќе ја користиме, и при тоа ќе имаме во предвид дека кога се зборува за задачата (5.1) - (5.2), истовремено зборуваме за задачата (5.1') - (5.2'), и обратно. Заради тоа за овие две задачи, кои се формално различни, ќе се зборува како за една иста задача.

Методите кои се развиени за решавање на овие задачи, овозможуваат наоѓање само на локални екстреми.

Имајќи во предвид на укажаните напомени, потребно е да се истакне дека во праксата постојат неколку типични задачи на НП, како што се: задачи за распределба на еднородни ресурси, некои транспортни задачи, задачи врзани за управување со залихи, итн. За некои од тие задачи ќе стане збор во наредниот дел.

5.2 Методи на решавање

Од поставување на задачата јасно се гледа дека кај решавањето на задачите на НП се работи за наоѓање на екстремни вредности на нелинеарните алгебарски функции $F(X)$, кои зависат од повеќе променливи, при што тие променливи мораат да ги задоволуваат посебните услови дефинирани со множеството ограничувања од обликот (5.2). Класичниот математички апарат за теорија на максимум и минимум на функција со повеќе променливи не може да обезбеди решение за задачите на НП од повеќе причини, меѓу кои доминанто место заземаат:

а) класичен математички апарат кој го претпоставува не само познавањето на аналитичкиот облик на функциите $F(X)$ и $g_i(X)$, $(i=1,2,...,m)$, туку и нивната непрекинатост, како и егзистенцијата на нивните парцијални изводи од втор ред, што во повеќето конкретни задачи на НП не е исполнето, или дури и ако е исполнето, долга е постапката за таа проверка.

б) класичниот математички апарат со варијациски пресметувања станува немоќен, ако се работи за наоѓање на екстремни вредности (максимум или минимум) на функции кои зависат од повеќе променливи (на пример: $n \geq 5$). Меѓутоа, реалните задачи имаат неколку десетици или дури и стотина променливи, што е една од основните препреки во примената на класичниот математички апарат.

Освен овие две наведени причини постојат и други причини кои влијаеле на создавањето на посебни методологии за нумеричко решавање на задачите на НП, со примена на современи сметачи. Еден од фундаменталните резултати во оваа област е даден во резултатите на Кун и Такер кои се однесуваат на задачите на НП, чија целна функција и множество ограничувања ги задоволуваат условите за конвексност.

5.3 Теорема на Кун и Такер

Теоремата на Кун и Такер во литературата често се споменува под името – *теорема за седлеста точка.* Зазема централно место во теоријата на конвексното програмирање и претставува воопштување на класичната метода на Лагранжовите множители.

Како што е познато, методата на Лагранжовите множители (мултипликатори) обезбедува наоѓање на екстремни вредности на функции кои зависат од повеќе променливи, при множество ограничувања кои се зададени со равенства. Меѓутоа, теоремата на Кун и Такер ја воопштува методата на Лагранжовите множители, проширувајки ја за наоѓање на екстремни вредности на функција која зависи од повеќе променливи, но при множество ограничувања кои не се зададени само со равенства туку и со неравенства. Теоремата на Кун и Такер дава потребен и доволен услов кој мора да го исполнува векторот $X = X^*$, кој претставува решение на задачата (5.1) - (5.2). Критериумите на исполнување на потребниот и доволен услов се формираат и се проверуваат врз база на *воопштена Лагранжова функција* $\Phi(X,\Lambda)$. За формирање на оваа функција воведуваме m-нови променливи кои се нарекуваат Лагранжови множители или мултипликатори, а кои ќе ги означуваме со $\lambda_1, \lambda_2, ..., \lambda_m$. Со други зборови, Лагранжовите множители ги сочинуваат компонентите на m - димензионалниот вектор Λ од кој зависи воопштената Лагранжова функција Φ која ја претставува функцијата зависна од $n+m$ променливи (X,Λ), која се формира на следниов начин

$$\Phi(X,\Lambda)=F(X)+\sum_{i=1}^{m}\lambda_i g_i(X). \tag{5.4}$$

Дури сега може да се даде прецизна дефиниција на теоремата на Кун и Такер која гласи:

Векторот $X=X^*$ претставува решение на задачата на НП, дефиниран за наоѓање на минимум на функцијата (5.1) при ограничувањата (5.2), тогаш и само тогаш кога постои вектор $\Lambda=\Lambda^*$ таков да

$$X^*\geq 0, \qquad \Lambda^*\geq 0, \tag{5.5}$$

$$\Phi(X^*,\Lambda)\leq\Phi(X^*,\Lambda^*)\leq\Phi(X,\Lambda^*) \tag{5.6}$$

за сите вредности на $X \geq 0$, $\Lambda\geq 0$.

Тогаш функцијата Φ во точката (X^*,Λ^*) мора да има глобален минимум во областа $X^*\geq 0$ во однос на X и глобален максимум во областа $\Lambda^*\geq 0$, во однос на Λ, или со други зборови: (X^*,Λ^*) претставува ненегативна седлеста точка за функцијата Φ.

Заради тоа оваа теорема често се нарекува *теорема за седлеста точка*, со оглед на тоа дека задачата на минимизација на $F(X)$ одговара на задачата за одредување на седлеста точка за функцијата Φ кај која од сите ограничувања сочувани се само ограничувањата во однос на знакот. Решението X^* на минимаксната задача претставува истовремено решение на минимизација на функцијата $F(X)$ и обратно.

Најпрвин ќе покажеме дека условите (5.5) и (5.6) се *доволни*. Нека (X^*,Λ^*) е седлеста точка на функцијата Φ во смисла на дефиницијата (5.6). Воведувајќи замена за вредноста на Φ преку изразот (5.4) и (5.6) се добива

$$F(X^*)+\sum_{i=1}^{m}\lambda_i g_i(X^*)\leq F(X^*)+\sum_{i=1}^{m}\lambda_i^* g_i(X^*)\leq F(X)+\sum_{i=1}^{m}\lambda_i^* g_i(X),$$

за сите вредности на

$$X\geq 0, \quad \Lambda\geq 0.$$

Како левото неравенство во претходниот израз мора да биде исполнето за секое Λ, тогаш е

$$g_i(X^*)\leq 0, \qquad (i=1,2,\ldots,m),$$

т.е. X^* претставува допуштен план, бидејќи припаѓа на областа (D) која е дефинирана со множеството ограничувања и условите

$$\sum_{i=1}^{m} \lambda_i^* g_i\left(X^*\right) = 0.$$

Десното неравенство со самото тоа добива облик

$$F\left(X^*\right) \leq F(X) + \sum_{i=1}^{m} \lambda_i^* g_i(X),$$

за сите вредности на $X \geq 0,$ од каде имајќи предвид за условот $\Lambda^* \geq 0,$ следи дека

$$F\left(X^*\right) \leq F(X),$$

за сите вредности на $X \geq 0$ кои ги задоволуваат условите

$$g_i(X) \leq 0, \qquad (i = 1,2,...,m),$$

заради што вредноста на X^* е решение на поставената задача (5.1) - (5.2).

За доказот дека условите (5.5) и (5.6) се *потребни* ќе ја искористиме претпоставката за регуларност врз основа на која, во краен случај, постои барем една точка $X = \overline{X}$ (допуштен план) таква да

$$g_i\left(\overline{x}\right) < 0, \qquad (i = 1,2,...,m), \tag{5.7}$$

Посебно е потребно да се истакне дека ***воведувањето на претпоставката за регуларност*** е непотребна за сите функции $g_i(X)$ кои се јавуваат како линеарни функции. Деталите на тој доказ овде нема да се наведуваат. Во секој случај, кога множеството ограничувања е линеарно, теоремата на Кун и Такер нема никакви ограничувања.

Кун и Такер првобитно ја докажале својата теорема само за случај на диференцијабилни функции $F(X)$ и $g_i(X)$ и при неколку различни претпоставки за регуларноста. Воопштувањето на оваа теорема на конвексни функции и условите (5.7) дал Слатер.

Кога функциите $F(X)$ и $g_i(X)$ се диференцијабилни, условите (5.5) и (5.6) се еквивалентни со следниве ‘локални’ услови на Кун - Такер:

$$\left.\begin{aligned}&\left(\frac{\partial \Phi}{\partial x_j}\right)_{X^*,\Lambda^*} \geq 0,\\&x_j^*\left(\frac{\partial \Phi}{\partial x_j}\right)_{X^*,\Lambda^*} = 0,\\&x_j^* \geq 0,\end{aligned}\right\} \qquad (j = 1,2,...,n), \tag{5.8}$$

$$\left.\begin{aligned}&\left(\frac{\partial \Phi}{\partial \lambda_i}\right)_{X^*,\Lambda^*} \leq 0,\\&\lambda_i\left(\frac{\partial \Phi}{\partial \lambda_i}\right)_{X^*,\Lambda^*} = 0,\\&\lambda_i^* \geq 0.\end{aligned}\right\} \qquad (i = 1,2,...,m), \tag{5.9}$$

Условите на Кун - Такер остануваат во важност и при некои измени во поставувањето на задачата (5.1) - (5.2). Така, на пример, може да се случи да при поставување на задачата не се присутни ограничувањата $x_j \geq 0,\ (j = 1,2,...,n)$. Во тој случај трите услови (5.8) се заменуваат само со еден услов

$$\left(\frac{\partial \Phi}{\partial x_j}\right)_{X^*,\Lambda^*} = 0. \tag{5.10}$$

За случај кога функциите $g_i(X)$ се линеарни, условите (5.9) се заменуваат со условот

$$\left(\frac{\partial \Phi}{\partial \lambda_i}\right)_{X^*,\Lambda^*} = 0, \tag{5.11}$$

кој претставува втор начин на запишување на условот

$$g_i(X) = 0, \qquad (i = 1,2,...,m).$$

Множителите $\lambda_i \ (i = 1,2,...,m)$ овде не се ограничуваат во однос на знакот.

На крајот, кога ограничувањата $g_i(X)$ се линеарни и дефинирани со равенството

$$g_i(X) = 0,$$

условите (5.8) и (5.9) се сведуваат на условите (5.10) и (5.11), кои претставуваат класичен случај на методата на Лагранжови множители (мултипликатори).

5.4 Градиентни методи

За решавање на задачите на НП постојат голем број на методи врз основа на кои се развиени алгоритмите кои одговараат. Од посебно значење имаат градиентните методи, заради кои подетално ќе зборуваме.

Градиентните методи претставуваат методи на систематско пребарување и наоѓање на решение. Сите тие во принцип имаат ист приод во наоѓање решение. Имено, со одбирањето на почетниот план $X^{(0)}=(x_1^{(0)}, x_2^{(0)},..., x_n^{(0)})$ се врши негова промена во смисла на дефинираната целна функција, тежнеејќи во еден чекор да се направат што повеќе ефикасни промени врз база на кои се одредува $X^{(1)}=(x_1^{(1)}, x_2^{(1)},..., x_n^{(1)})$. За тоа да се постигне потребно е да се пресмета најдобрата насока на промена која кај градиентните методи се изразува во однос на насоката на градиентот (спротивна насока во процесот на барање минимум, а директна насока во процесот на барање на максимум), по што методите и го добиле името.

Разликите кај поедини градиентни методи се базираат во однос на начинот на пресметување на градиентот, како и во однос на големината на поместувања кон бараното решение долж градиентната насока. Заради тоа и овде, како и кај другите итеративни процеси, присутен е проблемот на конвергенција на процесот како и проблемот на брзината на конвергенција на итерациониот процес за наоѓање решение. За сега, математичките библиотеки за современите сметачи располагаат со готови програми за примена на градиентните методи во решавањето на задачите на НП. Со оглед на тоа дека градиентните методи не се универзални, тоа е случај и со готовите програми за примена на сметачите, така да потребно е за секоја конкретна задача да се проверува дали исполнува одредени претпоставки во однос на примената на готовите програми.

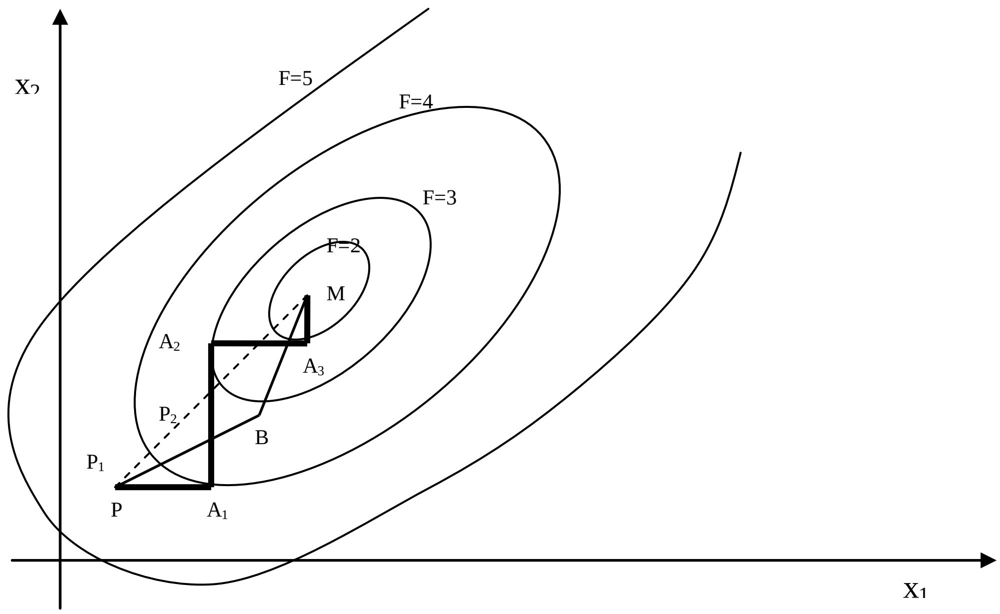
5.5 Графичка интерпретација на градиентните методи

Со цел за полесно разбирање, концептот на градиентните методи ќе биде изложен во примерот на одредување на минимумот на функцијата со две независни променливи, при што ќе се споредуваат и соодветните графички толкувања. За да биде процедурата што поедноставна, се претпоставува дека минимизацијата на функцијата $F(X) = F(x_1, x_2)$ се врши во отсуство на множеството ограничувања.

Нека е зададена функцијата

$$F(X) = F(x_1, x_2),$$

прикажана на цртеж 5.2., во рамнина (x_1, x_2) со помош на нивовски линии. Минимумот на функцијата $F(x_1, x_2)$ ќе го одредиме со некоја врста на пребарување по доменот на нејзината дефинираност во рамнината (x_1, x_2).



Цртеж 5.2 Графичка интерпретација на градиентни методи за функцијата $F = F(x_1, x_2)$

Ако се знае дека функцијата $F(x_1, x_2)$ и нејзините парцијални изводи по x_1 и x_2 се непрекинати функции, поставениот проблем може да се реши со пребарување на решението долж некоја линија, в долж која тоталниот прираст на функцијата е постојано негативен. На цртеж 5.2 се нацртани три такви линии P-A$_1$-A$_2$-A$_3$-M, P-B-M и P-M.

Алгоритмот за пребарување на решението долж линијата P-A$_1$-A$_2$-A$_3$-M се состои во следново:

Најпрво во почетната точка P се одредува парцијалниот извод на функцијата $F(x_1, x_2)$ по променливата x_1 и кога ќе се констатира дека е негативен, тогаш пребарувањето (континуално или дискретно) се врши долж правата P-A$_1$, при што x_2=const. , а x_1 расте се додека функцијата $F(x_1, x_{2P})$ опаѓа, т.е. до точката A$_1$. Потоа во точката A$_1$ се одредува парцијалниот извод на функцијата $F(x_1, x_2)$ по x_2, па бидејќи е негативен, пребарувањето продолжува долж правата A$_1$-A$_2$ се додека функцијата $F(x_{1A1}, x_2)$ опаѓа, т.е. до точката A$_2$, итн. Ова пребарување долж правата паралелна на x_1 оската, а потоа долж правата паралелна на x_2 оската се продолжува наизменично, се додека не се одредат координатите на минимумот на функцијата $F(x_1, x_2)$ со произволна точност.

Другата можност се состои во пребарување на минимум (максимум) на функцијата $F(x_1, x_2)$ долж линијата со најголем пад (раст). За разгледуваниот пример на минимизација на функцијата $F(x_1, x_2)$, линијата на најголем пад P-M е прикажана на цртеж 5.2. Алгоритам за такво пребарување доаѓа од одредување на насоката на најголемиот пад на функцијата $F(x_1, x_2)$ во точката P која е спротивна на правецот на градиентот на функцијата, а потоа се врши елементарно поместување во таа насока до соседната точка P_1, во која исто така се одредува насоката на најголемиот пад како во точката P.

Понатамошната постапка продолжува вршејќи елементарно поместување од точката P_1 во P_2, итн., се додека не се добие минимумот на функцијата $F(x_1, x_2)$. Со други зборови, оваа постапка се повторува се додека функцијата $F(x_1, x_2)$ не престане да опаѓа, т.е. до точката во која функцијата $F(x_1, x_2)$ добива минимална вредност. Во таа точка градиентот на функцијата станува еднаков на нула и во тоа се состои дескриптивно графичката интерпретација на градиентните методи. За аналитичката интерпретација на оваа метода ќе биде посебно кажано.

Пресметувањето на градиентната функција $F(X)$ претставува обемна и заморна работа, па од сметачки аспект поедноставно е тоа пресметување да се врши повремено, што ефикасно се користи во модифицираните методи на градиентот. Пребарувањето на вредностите на функцијата $F(x_1, x_2)$ користејќи ја оваа метода е прикажано на цртеж 5.2 со линијата P-B-M. Во тој случај алгоритмот се состои во следново:

Прво се одредува насоката на најголемиот пад во почетната точка P, а потоа се пребаруваат вредностите долж правата P-B која е колинеарна со правецот на градиентот на функцијата во точката P и тоа се додека функцијата $F(x_1, x_2)$ опаѓа, т.е. до точката B. После тоа се одредува насоката на најголемиот пад на функцијата во точката B и пребарувањето се продолжува долж правата која почнува од точката B, а колинеарна е со правецот на градиентот на функцијата $F(x_1, x_2)$ во точката B. Процесот понатаму продолжува на ист начин. Сето ова може да се воопшти и на функциите $F(X)$, кои зависат од $n-$независни променливи.

5.6 Аналитичка интерпретација на градиентните методи

За разлика од графичката интерпретација на градиентните методи, која е илустрирана во примерот за функцијата која зависи од две независни променливи, овде аналитичката интерпретација ќе се прикаже за случај кога функцијата $F(X)$ зависи од $n-$независни променливи т.е.

$$F(X) = F(x_1, x_2, ..., x_n).$$

Ако во $n-$димензионалниот простор се одбере насоката s, тогаш изводот на функцијата $F(X)$ во точката $X = (x_1, x_2, ..., x_n)$, во насока s е даден со изразот

$$\frac{dF}{ds} = \sum_{j=1}^{n} \frac{\partial F}{\partial x_j} \frac{dx_j}{ds}, \tag{5.12}$$

каде:

$\frac{\partial F}{\partial x_j}$ – парцијални изводи на функцијата $F(X)$,

$\frac{dx_j}{ds}$ – косинуси од агли меѓу насоката s и координатните оски,

$\frac{dF}{ds}$ – извод на функцијата $F(X)$ во насоката s.

Воведувајќи ознака

$$z_j = \frac{dx_j}{ds},$$

Изразот (5.12) може да се напише во обликот

$$\frac{dF}{ds} = \sum_{j=1}^{n} \frac{\partial F}{\partial x_j} z_j. \tag{5.13}$$

За случај кога функцијата $F(X)$ се минимизира (максимизира), потребно е да се одреди онаа насока, долж која изводот на функцијата има најмала(најголема) вредност. Меѓутоа, променливите z_j не се меѓусебно независни, бидејќи постои следнава релација

$$ds = \left(\sum_{j=1}^{n} dx_j^{\,2} \right)^{1/2},$$

која очигледно може да се преуреди и напише во обликот

$$1 - \sum_{j=1}^{n} z_j^{\,2} = 0. \tag{5.14}$$

На тој начин е дефиниран проблемот на минимизација на функцијата $\frac{dF}{ds}$ дефинирана со равенството (5.13) со присуство на условот (5.14). Со оглед на тоа дека постои само едно ограничување (услов) од обликот (5.14), воопштената Лагранжова функција која одговара може да се напише во обликот

$$\Phi = \Phi(X,\Lambda) = \sum \frac{\partial F}{\partial x_j} z_j + \lambda_0 \left(1 - \sum_{j=1}^{n} z_j^2 \right),$$

каде λ_0 – Лагранжов множител. Наоѓањето на парцијалните изводи со воопштената Лагранжова функција Φ по z_j и нивните изедначувања со нула се добива

$$\frac{\partial F}{\partial x_j} - 2\lambda_0 z_j = 0, \qquad (j = 1,2,...,n),$$

откаде следи

$$z_j = \frac{1}{2\lambda_0} \frac{\partial F}{\partial x_j}, \ (j = 1,2,...,n). \tag{5.15}$$

Со замена на вредноста за z_j од изразот (5.15) во изразот (5.14) се добива дека

$$\lambda_0 = \pm \frac{1}{2} \left[\sum_{j=1}^{n} \left(\frac{\partial F}{\partial x_j} \right)^2 \right]^{1/2}. \tag{5.16}$$

Кога вредноста за λ_0 од изразот (5.16) се воведе во изразот (5.15), се добива следнава вредност за z_j за насоката на најголемиот пад

$$z_j = -\frac{\partial F}{\partial x_j} \left[\sum_{j=1}^{n} \left(\frac{\partial F}{\partial x_j} \right)^2 \right]^{-1/2}, \qquad (j = 1,2,...,n),$$

односно

$$z_j = \frac{\partial F}{\partial x_j} \left[\sum_{j=1}^{n} \left(\frac{\partial F}{\partial x_j} \right)^2 \right]^{-1/2}, \qquad (j = 1,2,...,n),$$

за насоката на најголемиот раст. Конечно, ако вака добиените вредности за z_j се вметнат во изразот (5.13), се добива дека изводот на функцијата $F(X)$ во насока на најголемиот пад е еднаков

$$\frac{dF}{ds} = -\sum_{j=1}^{n} \left(\frac{\partial F}{\partial x_j} \right)^2 \left[\sum_{j=1}^{n} \left(\frac{\partial F}{\partial x_j} \right)^2 \right]^{-\frac{1}{2}} = -\left[\sum_{j=1}^{n} \left(\frac{\partial F}{\partial x_j} \right)^2 \right]^{-\frac{1}{2}} \left[\sum_{j=1}^{n} \left(\frac{\partial F}{\partial x_j} \right)^2 \right] = -\left[\sum_{j=1}^{n} \left(\frac{\partial F}{\partial x_j} \right)^2 \right]^{\frac{1}{2}},$$

односно за насока на најголемиот раст

$$\frac{dF}{ds}=\left[\sum_{j=1}^{n}\left(\frac{\partial F}{\partial x_j}\right)^2\right]^{\frac{1}{2}}.$$

Ако за брзината на работната точка се одбере вредност која е пропорционална на градиентот на функцијата, т.е.

$$\frac{ds}{dt}=k\left[\sum_{j=1}^{n}\left(\frac{\partial F}{\partial x_j}\right)^2\right]^{\frac{1}{2}}$$

каде $k-$коефициент на пропорционалност, се добива дека

$$\frac{dx_j}{dt}=-k\left(\frac{\partial F}{\partial x_j}\right),\quad (j=1,2,...,n),\tag{5.17}$$

Системот диференцијални равенки (5.17) опишува континуален процес на градиентните методи, при што почетните услови кои одговараат на почетната точка P се:

$$x_j=(t_0=0)=x_j^{(0)},\qquad (j=1,2,...,n),\tag{5.18}$$

За системот (5.17) и почетните услови (5.18), кога $t\to\infty$ работната точка се приближува кон минимумот на функцијата $F(X)$ во која и

$$grad\ \ F(X)=0\ .$$

Во процесот на барање на максимум на функцијата $F(X)$, во системот диференцијални равенки (5.17), наместо минус се зема знакот плус.

Вака формулираната задача на максимизација (минимизација) на функцијата

$$F(X)=F(x_1,x_2,...,x_n),$$

но без множеството ограничувања (5.2’), односно (5.2), може да се прикаже и во кондензиран облик. Претставувајќи го $grad\ \ F(X)$ како

$\nabla\ F(X)$, каде

$$\nabla F(X) = \begin{Vmatrix} \frac{\partial F}{\partial x_1} \\ . \\ . \\ . \\ \frac{\partial F}{\partial x_n} \end{Vmatrix},$$

системот диференцијални равенки кој се користи за максимизација на функцијата $F(X)$ може да се напише со употреба на воведените ознаки како

$$\frac{dX}{dt} = k\nabla F(X),$$

$$X(t_0 = 0) = X^{(0)}, \qquad (5.19)$$

каде $X = (x_1, x_2, ..., x_n)$, а

$$X^{(0)} = (x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)}).$$

Во повеќето случаи се избира вредност за коефициентот на пропорционалноста $k = 1$, што воедно овозможува процесот, опишан со системот диференцијални равенки (5.17), односно (5.19), да може многу едноставно да се преведе во дискретен процес.

Системот диференцијални равенки (5.19) го опишува најкраткиот пат од точката $X^{(0)}$ кон точката во која функцијата $F(X)$ постигнува максимална вредност. Заради тоа максимумот (минимумот) на функцијата може да се одреди како решение на системот (5.19), кога $t \to \infty$, за било кој почетен услов $X(t_0 = 0) = X^{(0)}$.

За оваа метода да одговара на условите од формулираната задача (5.1') - (5.2'), односно (5.1) - (5.2) треба на системот диференцијални равенки (5.19) за $k = 1$, односно $k = -1$, да му се додадат условите кои го задржуваат X во областа дефинирана со множеството ограничувања (5.2'), односно (5.2), при што тие услови треба да влијаат на решението на системот (5.19) само во случај ако решението отпаѓа од доменот на ограничувањата.

За таа цел се користи систем диференцијални равенки

$$\frac{dX}{dt} = \nabla F(X) - \sum_{i=1}^{m} \delta_i(X) \nabla g_i(X), \qquad (5.20)$$

каде

$$\delta_i(X) = \begin{cases} 0, & \text{za} \quad g_i(X) \geq 0 \\ \max\limits_{i} \ \max\limits_{X} \dfrac{|\nabla F(X)|}{|\nabla g_i(X)|} + \varepsilon, & \text{za} \quad g_i(X) < 0 \end{cases}$$

и ε > 0.

Јасно може да се види тоа што е претходно кажано, дека веднаш штом се наруши нееднаквоста $g_i(X) \geq 0,$ системот равенки (5.20) не се дегенерира бидејќи $\delta_i(X)$ не е еднакво на нула.

Континуалното решавање на формулираната задача на НП на овој начин е сведено на решавање на системот диференцијални равенки (5.20), чие решение, освен за некои попрости примери, тешко може да се добие во затворен облик, заради што е неопходна примена на нумеричките методи. Освен тоа, испитувањето на конвергенцијата на решението како и итерациониот процес кој се користи за негово добивање не претставува, во најголем број случаи, едноставен проблем, што е еден од основните недостатоци на оваа метода кога е во прашање максимизација (минимизација) на функцијата во присуство на множеството ограничувања.

5.7 Диференцијална градиентна метода

Оваа метода е развиена врз база на теоремата на Кун и Такер, изложена во поглавјето 5.3. Имено, тие покажале дека $n-$димензионалниот вектор

$$X = (x_1, x_2, ..., x_n),$$

чии компоненти претставуваат координати на седлеста точка на функцијата

$$\Phi(X, \Lambda) = F(X) + \sum_{i=1}^{m} \lambda_i g_i(X),$$

при условите $\lambda_i \geq 0$, $(i = 1,2,...,m),$ претставува истовремено и решение на задачата (5.1’) -(5.2’), односно (5.1) - (5.2).

Одредувањето на седлеста точка со Лагранжовите функции $\Phi(X,\Lambda)$ се сведува на решавање на системот диференцијални равенки

$$\frac{dX}{dt} = \nabla F(X) + \sum_{i=1}^{m} \lambda_i \nabla g_i(X), \tag{5.21}$$

$$\frac{d\lambda_i}{dt} = \begin{cases} -g_i(X), & \text{ако е } \lambda_i > 0 \text{ ili } g_i(X) < 0, \\ 0, & \text{во спротивен случај,} \end{cases}$$

кога $t \to \infty$ при што се зема дека почетните услови кога $t = 0$ се зададени како вредности

$$X(0) = X^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}) .$$

$$\Lambda(0) = \Lambda^{(0)} = (\lambda_1^{(0)}, \lambda_2^{(0)}, \ldots, \lambda_n^{(0)}) > 0,$$

Врз основа на резулатите на Кун и Такер може да се покаже дека онаа вредност

$$X = X^* = \left(x_1^*, x_2^*, \ldots, x_n^*\right),$$

која претставува решение на системот диференцијални равенки (5.21) за било кој почетни услови

$$X(0) = X^{(0)} \geq 0 ,$$

$$\Lambda(0) = \Lambda^{(0)} \geq 0 ,$$

кога $t \to \infty$, претставува истовремено и точка во која функцијата (5.1’) при множеството ограничувања (5.2’) достигнува максимална вредност. Аналогните математички модели, како што е зададениот математички модел со системот диференцијални равенки (5.21), со посредство на кој што се решава задачата на НП (5.1’) - (5.2’), може на сличен начин да се конструираат и за задачата на НП дефиниран со изразите (5.1) - (5.2), што не претставува никаква посебна тешкотија.

Со други зборови, диференцијалната градиентна метода овозможува задачите на НП и нивното решавање да се сведе на решавање на задачи со систем диференцијални равенки од обликот (5.21). Тешко е и да се помисли, изоставувајќи некои поедноставни случаи на задачи на НП дека решението на системот диференцијални равенки (5.21) може да се добие во затворен облик. Заради тоа излезот во процесот на барење на неговото решение се наоѓа исклучиво во примената на нумеричките методи за пресметување, односно било која метода со конечни разлики, каде се земаат во предвид критериумите за нивна примена, како што се: точност на методите, брзина на конвергенција на решението, број на сметачки операции, итн.

Заради илустрација на методата на конечни разлики, која многу често се користи за решавање на систем диференцијални равенки од обликот (5.21), ќе се послужиме со следниот пример во рамки на кој е потребно да се одреди максимумот на функцијата

$$F(X) = F(x_1, x_2) = -x_1^2 - x_2^2 + 4x_1,$$

при облик на ограничувањата

$$x_1 \geq 0,$$

$$x_2 \geq 0,$$

$$-x_1^2 - 2x_2^2 + x_2 + 1 \geq 0,$$

$$-0{,}2x_1^2 - 12x_2 + 1 \geq 0,$$

Ќе претпоставиме дека е потребно да се одреди решение со точност 10^{-2}.

Тргнувајќи од воопштената Лагранжова функција $\Phi(X, \Lambda)$, врз основа на која се добива систем диференцијални равенки од обликот (5.21), за дадениот пример одговара системот диференцијални равенки кој има општ облик

$$\frac{dx_j}{dt} = \frac{\partial F(x_1, x_2)}{\partial x_j} + \sum_{i=1}^{4} \lambda_i \frac{\partial g_i(x_1, x_2)}{\partial x_j}, \quad (j = 1,2)$$

$$\frac{d\lambda_i}{dt} = \begin{cases} -g_i(X), & ако\ е\ \lambda_i > 0 \ \text{ili} \ g_i(X) < 0, \\ 0, & \text{во спротивен случај}, \end{cases}$$

при што индексите се $(i = 1,2,3,4)$.

Овој систем диференцијални равенки непосредно произлегува од системот (5.21), кога имаме дефиниран конкретен пример. Овие равенки уште попрецизно може да се конкретизираат во однос на дадениот пример, така да од првото равенство за $(j = 1,2)$ следува:

$$\frac{dx_1}{dt} = -2x_1 + 4 + \lambda_1 - 2\lambda_3 x_1 - 0{,}4\lambda_4 x_1,$$

$$\frac{dx_2}{dt} = -2x_2 + \lambda_2 + (1 - 4x_2)\lambda_3 - 1{,}2\lambda_4,$$

додека врз основа на второто равенство за $(i = 1,2,3,4)$ се добива

$$\frac{d\lambda_1}{dt} = \begin{cases} -x_1, & \text{ако е } \lambda_1 > 0, \text{ ili } x_1 < 0, \\ 0, & \text{во спротивен случај,} \end{cases}$$

$$\frac{d\lambda_2}{dt} = \begin{cases} -x_2, \ \textit{ако е} & \lambda_2 > 0, \text{ ili } x_2 < 0, \\ 0, & \text{во спротивен случај,} \end{cases}$$

$$\frac{d\lambda_3}{dt} = \begin{cases} x_1^2 + 2x_2^2 - x_2 - 1, & \text{ако е } \lambda_3 > 0, \text{ или } \quad -x_1^2 - 2x_2^2 + x_2 + 1 < 0, \\ 0, & \text{во спротивен случај,} \end{cases}$$

$$\frac{d\lambda_4}{dt} = \begin{cases} 0{,}2x_1^2 + 1{,}2x_2 - 1, & \textit{ако е } \lambda_4 > 0 \text{ или } \quad -0{,}2x_1^2 - 1{,}2x_2 + 1 < 0, \\ 0, & \text{во спротивен случај,} \end{cases}$$

Избираме за вака дефиниран систем диференцијални равенки, со посредство на кој може да се најде решение на дадениот пример, дека сите почетни услови

$$x_1^{(0)}, \ x_2^{(0)}; \quad \lambda_1^{(0)}, \ \lambda_2^{(0)}, \lambda_3^{(0)}, \ \lambda_4^{(0)}$$

се еднакви на нула. Заради тоа однапред формулираниот систем диференцијални равенки за интервал [0, t_1] ќе има облик

$$\frac{dx_1}{dt} = -2x_1 + 4$$

$$\frac{dx_2}{dt} = -2x_2$$

$$\frac{d\lambda_i}{dt} = 0, \qquad (i = 1,2,3,4),$$

чие решение може да се добие во затворен облик, така да:

$$x_1 = 2 - 2e^{-2\tau},$$
$$x_2 = 0,$$
$$\lambda_i = 0, \quad (i = 1,2,3,4).$$

Вредноста за t_1 се одредува како целна точка, во која истовремено се исполнети следниве три услови:

$$x_1 \geq 0,$$
$$-x_1^2 + 1 \geq 0,$$
$$-0{,}2\, x_1^2 + 1 \geq 0.$$

Овие услови се образувани врз основа на множеството ограничувања на дадениот пример. Оттука следи дека

$$t_1 = \frac{\ln 2}{2} = 0{,}35.$$

Сега се преминува на систем диференцијални равенки

$$\frac{dx_1}{dt} = -2x_1 - 2\lambda_3 x_1 + 4,$$
$$\frac{dx_2}{dt} = -2x_2 - 4\lambda_3 x_2 + \lambda_3,$$
$$\frac{d\lambda_3}{dt} = x_1^2 + 2x_2^2 - x_2 - 1,$$
$$\frac{d\lambda_i}{dt} = 0, \qquad (i = 1,2,4).$$

кој се образува од системот диференцијални равенки кој служи за решавање на дадениот пример. Решението на овој систем равенки потребно е да се најде за интервалот [t_1, t_2], при што почетните услови се

$$x_1(t_1) = 1,$$
$$x_2(t_1) = 0,$$
$$\lambda_i(t_1) = 0, \qquad (i = 1,2,3,4).$$

Слично како за претходниот случај кај кого е извршено одредување на t_1 и овде се одредува вредноста на t_2 како целна точка во која се исполнети условите:

$$x_1 \geq 0,$$
$$x_2 \geq 0,$$
$$-0{,}2\, x_1^2 - 1{,}2\, x_2 + 1 < 0, \text{ и}$$
$$\begin{cases} -x_1^2 - 2x_2^2 + x_1 + 1 < 0, \\ \text{ili} \\ \lambda_3 > 0. \end{cases}$$

Понатамошната постапка се состои во интеграција на претходно наведениот систем диференцијални равенки за дадените почетни услови, во формирањето на итерационите равенки

$$x_1^{(k+1)} = x_1^{(k)} + h\,(-2\, x_1^{(k)} - 2\, \lambda_3^{(k)}\, x_1^{(k)} + 4),$$

$$x_2^{(k+1)} = x_2^{(k)} + h\,(-2\, x_2^{(k)} - 4\, \lambda_3^{(k)}\, x_2^{(k)} + \lambda_3^{(k)}),$$

$$\lambda_3^{(k+1)} = \lambda_3^{(k)} + h\,(x_1^{(k)^2} + 2\, x_2^{(k)^2} - x_2^{(k)} - 1).$$

Овој систем на итерациони равенки, во смисла на конвергенција на решението, се врзува за конвергенцијата на итерациониот процес. Ако во почетокот се избере чекор $h_1 = 0{,}1$ и ако, после неколку итерации, не се добие точката на смирување за која се постигнува вредност t_2, тогаш се усвојува нов чекор $h_2 = 0{,}01$ и со таа вредност h_2 се спроведува одново итерационата процедура, при што почетокот на овој итерационен процес е резултат на последната итерација за чекор h_1 и $\lambda_i = 0$, $(i = 1,2,4)$.

Со примена на опишаната постапка за одредени вредности на h_1 и h_2 се добиваат следниве резултати на 14-тата и 15-тата итерација:

$$x_1^{(14)} = 1{,}1; \qquad x_1^{(15)} = 1{,}1;$$
$$x_2^{(14)} = 0{,}15; \qquad x_2^{(15)} = 0{,}15;$$
$$\lambda_3^{(14)} = 0{,}9; \quad \lambda_3^{(15)} = 0{,}9;$$

кога е $h = h_1 = 0{,}1$, додека конечните резултати се добиени за чекор $h = h_2 = 0{,}01$ и изнесуваат:

$$x_1^* = 1{,}05; \qquad x_2^* = 0{,}15; \qquad \lambda_3 = 0{,}88.$$

Со други зборови, овие резултати претставуваат координати на седлеста точка на воопштената Лагранжова функција за задачата на НП која се решава, со што истовремено е одредено неговото решение за кое целната функција $F(x_1, x_2)$ ја добива својата максимална вредност, т.е.

$$\max F(x_1, x_2) = F(x_1^*, x_2^*) = 3{,}05 .$$

Изложената процедура за наоѓање на оптималното решение на задачата на НП на бројниот пример овозможува подобро да се сфати суштината на диференцијалните градиентни методи, како и редоследот и сложеноста на нумеричките процедури кои треба да се сработат за да се добие оптимално решение на поедини задачи на НП.

5.8 Метода на секачки рамнини

Оваа метода се базира на мислење дека областа на ограничување (D) може да биде претставена со два полупростори (две полуобласти), (D_1) и (D_2) кои се содржани во (D). Областите (D_1) и (D_2) се добиваат со сечење на областа (D) со рамнини, односно хипер - рамнини, со оглед на тоа дека се работи за повеќедимензионален простор.

Со воведувањето на дополнителна променлива x_{n+1}, задачата на НП дефинирана со изразите (5.1') - (5.2') може да биде преформулирана во задача која се состои, наместо во наоѓање на максимум на нелинеарна целна функција $F(X)$, во наоѓање на максимум во линеарна форма, што од математичка страна претставува поедноставен проблем.

За таа цел, со воведување на следниве ознаки

$$\overline{g_i}\,(x_1, x_2, \dots, x_{n+1}) = g_i(x_1, x_2, \dots, x_n), \qquad (i = 1,2,\dots,m),$$

$$\overline{g_{m+1}}\,(x_1, x_2, \dots, x_{n+1}) = -x_{n+1} + F(x_1, x_2, \dots, x_n),$$

може да се формулира математички модел од обликот

$$\overline{g_i}\,(x_1, x_2, \dots, x_{n+1}) \ge 0, \quad (i = 1,2,\dots,m+1)$$

$$x_{n+1} = \max \qquad (5.23)$$

кој го дава решението на задачата (5.1') - (5.2').

Вака формулираниот математички модел од обликот (5.23) дава приближна вредност за оптималниот план на задачата (5.1') - (5.2'), што претставува многу погоден пристап за одредување на почетниот план на задачата на НП, кој се подобрува со некоја друга попрецизна метода.

Не е тешко да се заклучи, со оглед на тоа дека $\overline{g_i}\ (x_1, x_2, ..., x_{n+1})$ се конкавни функции, дека неравенката

$$\overline{g_i}\left(X^t\right)+\nabla\overline{g_i}\left(X^t\right)\left(X-X^t\right)\geq 0\ ,$$

ќе биде задоволена за било која вредност на векторот X кој го задоволува множеството ограничувања $\overline{g_i}(X) \geq 0$.

Навистина имаме дека

$$\overline{g_i}\left(X^t\right)+\nabla\overline{g_i}\left(X^t\right)\left(X-X^t\right)\geq\overline{g_i}(X)\geq 0\ ,$$

заради кој математичкиот модел (5.23) може да се сведе на следниов модел

$$\overline{g_i}\left(X^t\right)+\nabla\overline{g_i}\left(X^t\right)\left(X-X^t\right)\geq 0\ , \qquad (i=1,2,...,m+1),$$

$$(t=1,2,...,T),$$

$$x_{n+1}=\max. \qquad (5.24)$$

5.9 Методи на линеаризација

Нека земеме дека $X^1, X^2, ..., X^T$ е множество од $n-$димензионални вектори. Тогаш точката X која припаѓа на конвексната контура на тоа множество може да биде прикажана на следниов начин

$$X=\sum_{t=1}^{T}u^t X^t,$$

каде

$$u^t\geq 0,\quad (t=1,2,...,T), \qquad (5.25)$$

$$\sum_{t=1}^{T}u^t=1.$$

Разгледувајќи ја следнава линеаризација на функцијата $\eta(X)$:

$$\eta(X)=\sum_{t=1}^{T}u^t\eta\left(X^t\right),$$

може да се забележи дека изборот u^t не може да се изврши единствено врз основа на изразот (5.25). Заради тоа задачата ќе се разгледа во однос на u^t кој се дефинира со следниве изрази:

$$\sum_{t=1}^{T}u^t g_i\left(X^t\right)\geq 0, \qquad (i=1,2,...,m),$$

$$\sum_{t=1}^{T}u^t=1, \quad u^t\geq 0, \tag{5.26}$$

$$\sum_{t=1}^{T}u^t F\left(X^t\right)=\max.$$

Под претпоставка дека u_0^1, u_0^2, ... , u_0^T е решение на задачата (5.26), лесно може да се одреди векторот

$$X^{(0)}=\sum_{t=1}^{T}u_0^t X^t,$$

кој ги задоволува ограничувањата

$$g_i(X)\geq 0, \qquad (i=1,2,...,m),$$

ако ги задоволуваат X^1, X^2, ... , X^T.

Решението на задачата (5.26) се зема како приближно решение на задачата (5.1') - (5.2'). Наоѓање на вакво приближно решение пропратено е со одредени тешкотии, кои се состојат во изборот на системот вектори

$$X^1,\ X^2,\ ...,\ X^T$$

кој ќе обезбедат доволно блиско решение за кое се добива екстрем на целната функција на задачата на НП, дефинирана со изразите (5.1') - (5.2'). Од друга страна, методата на линеаризација има ограничена примена која зависи од димензијата на математичкиот модел на задачата на НП, со оглед на тоа дека зголемувањето на нејзината димензија доведува до голем број променливи u^T. Инаку, методата на линеаризација е ефикасна во случаите кога функциите $g_i(X)$и целната функција $F(X)$ во внатрешноста на областа на ограничување (D) се доволно блиски на линеарните функции, кои зависат од мал број величини X^T.

За сите случаи на задачата на НП, кај кои методата на линеаризација ефикасно е применлива, не е потребно да се земаат во предвид големиот број точки за линеаризација на задачата (5.1') - (5.2'). За таа цел се тргнува од следнава задача

$$g_i(X) \equiv \sum_{j=1}^{n} g_{ij}(x_j) \geq 0, \quad (i = 1,2,...,m),$$

$$F(X) \equiv \sum_{j=1}^{n} F_j(x_j) = \max,$$

при што за $F(X)$ и $g_i(X)$ се исполнети сите услови од задачата (5.1')-(5.2').

Како што може да се забележи, овде линеаризацијата се спроведува одвоено за секоја променлива

$$x_j = \sum_{t=1}^{T} u_j{}^t x_j{}^t.$$

Врз основа на ова се формулира нова задача

$$\sum_{j=1}^{n} \sum_{t=1}^{T_j} u_j^t \, g_{ij} \left(x_j^t\right) \geq 0;$$

$$\lambda_j^t \geq 0, \qquad \sum_{t=1}^{T_j} u_j^t = 1; \qquad (j = 1,2,...,n);$$

$$\sum_{j=1}^{n} \sum_{t=1}^{T_j} u_j^t F_j \left(x_j^t\right) = \max,$$

која ако има конвергентно решение, конвергентно решение има и задачата (5.27).

5.10 Алгоритам за одредување на почетен план

Кај сите методи за решавање на задачите на НП, како и кај задачите на ЛП, се поставува едно многу важно прашање кое се однесува на проблемот на одредување на почетниот план на задачата, што во повеќето случаи не е ни малку едноставно.

Еден од најуспешните алгоритми од оваа врста кој се однесува на одредување на почетен план на задача на НП, врзан е за одредување на апсолутен максимум на функцијата

$$F(X) = -\sum_{i=1}^{m} \left(\max_{X} \left[0, -g_i(X)\right] \right)^2. \tag{5.28}$$

Лесно може да се види дека

$$\max_{X} F(X) = 0,$$

во точката $\overline{X}$, при што во истата таа точка $X = \overline{X}$ имаме дека сите ограничувања се

$$g_i(\overline{X}) \geq 0 ,$$

т.е. $\overline{X}$ е допуштениот план, ако воопшто допуштениот план егзистира.

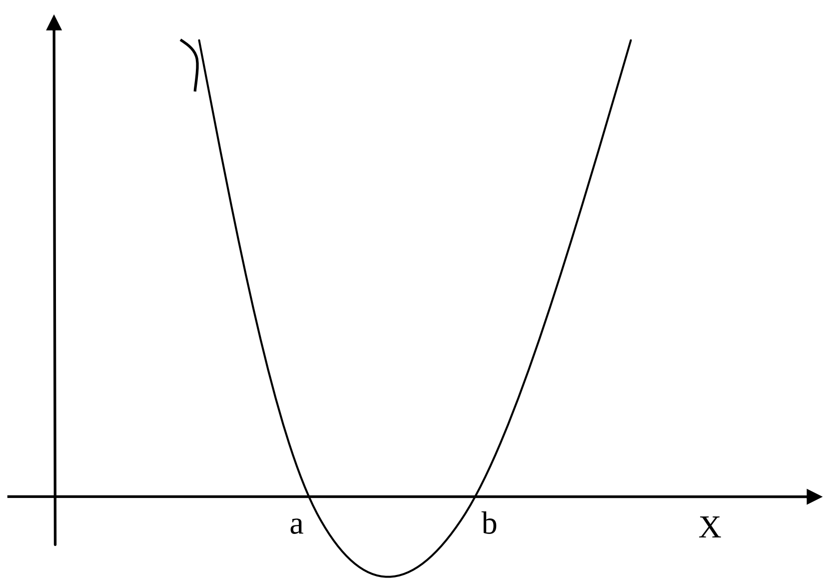
Може да се забележи дека, ако функциите $g_i(X)$ се конвексни, тогаш и функциите

$$\eta_i(X) = \left(\max_{X}\left[0, -g_i(X)\right]\right)^2$$

се исто така конвексни. На цртеж 3. дадена е парабола која претставува график на функцијата $-g_i(X)$, додека со подебела црта е даден графикот на функцијата.

$$\max_{X} [0, -g_i(X)].$$

$$-g_i(X)$$



Цртеж 5.3 Графичка функција на $-g_i(X)$ **и** $\max_{X} [0, -g_i(X)]$

Јасно се гледа дека ако функциите $g_i(X)$ се конвексни и диференцијабилни функции, тогаш и функциите $\eta_i(X)$ се исто така конвексни и диференцијабилни функции. Во точката a изводот на функцијата $\eta_i(X)$ од левата страна е еднаков на

$$\eta_i^{'}(a-0)=2g_i(a-0)g_i^{'}(a-0)=0.$$

На сличен начин може да се покаже дека десниот извод на функцијата $\eta_i(X)$ во точката a исто така е еднаков на нула. За точката b имаме потполна аналогија.

За одредување на апсолутниот максимум на конвексната диференцијабилна функција $F(X)$ дадена со изразот (5.28), може да се користи било која претходно наведена метода и тоа за било кои почетни услови. Задачата на НП кај кои множеството ограничувања е линеарно не се водат под оваа процедура во целта на одредување на почетниот план.

5.11 Други методи за решавање на задачите на нелинеарно програмирање

Според однапред изложените методи, постојат уште низа други методи кои може да се користат за решавањето на поедини задачи на НП. Некои од тие методи имаат многу кратки и ограничени можности за примена. Нешто подетално ќе се укаже на две методи, чија примена може да дојде до израз во решавањето на една широка класа на задачи на НП, за случај кога ограничувањата се зададени како равенства, така и за неравенства. Една од тие методи ќе се сретне под името *метода на ограничен градиент*, а друга како *метода на проширени крајни функции* со ограничувања.

За случај кога множеството ограничувања за одредена задача на НП е дефинирано со равенства од обликот:

$$g_i(x_1, x_2, ..., x_n)=0, \qquad (i=1,2,...,m), \tag{5.29}$$

каде $m<n$ при што е потребно да се одреди екстремната вредност на целната функција

$$F(X)=F(x_1, x_2, ..., x_n), \tag{5.30}$$

примената на методата на ограничен градиент за наоѓање на насоката на најголем пад и минимумот на функцијата (5.30), при ограничувања (5.29), се сведува на следнава процедура:

Најпрвин е потребно да се изврши парцијално диференцирање на равенството (5.29), по s, при што се добива :

$$\sum_{j=1}^{m}\left(\frac{\partial g_i}{\partial x_i}\right)z_j = 0\,, \qquad (i = 1,2,...,m), \tag{5.31}$$

каде $z_j = \frac{dx_j}{ds}$, а потоа ограничувањата за променливата z_j се воведуваат во воопштена функција со помош на Лангранжовите множители $\lambda_0, \lambda_1, \lambda_2, ..., \lambda_m$ која има облик

$$\Phi = \sum_{j=1}^{n}\left(\frac{\partial F}{\partial x_i}\right)z_j + \lambda_0\left(1 - \sum_{j=1}^{n} z_j^2\right) + \sum_{i=1}^{m}\lambda_i \sum_{j=1}^{n}\left(\frac{\partial g_i}{\partial x_i}\right)z_j. \tag{5.32}$$

Со изедначување на парцијалните изводи на функција (5.32) по z_j со нула се добива

$$z_j = \frac{1}{2\lambda_0}\left(\frac{\partial F}{\partial x_j}\right) + \sum_{i=1}^{m}\lambda_i\left(\frac{\partial g_i}{\partial x_i}\right), \qquad (j = 1,2,...,m) \tag{5.33}$$

при што ако од (5.33) вредностите за z_j се воведат во равенката (5.31), ќе се добие систем линеарни равенки по λ_i од обликот:

$$\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i\left(\frac{\partial g_i}{\partial x_i}\right)\left(\frac{\partial g_k}{\partial x_j}\right) = -\sum_{i=1}^{n}\left(\frac{\partial g_k}{\partial x_j}\right)\left(\frac{\partial F}{\partial x_j}\right) \tag{5.34}$$

$$(k = 1,2,...,m),$$

кој има решение под условот детерминантата на системот да е различна од нула или, кажано со други зборови, ако векторите на градиентот g_i се различни од нула и ако не се колинеарни.

Процедурата на методата на ограничен градиент може да се искаже со следнава рекурентна релација

$$x_j^{(K+1)} = x_j^{(K)} - \left(\frac{\partial F}{\partial x_j} + \sum_{i=1}^{m}\lambda_i\frac{\partial g_i}{\partial x_j}\right)\Delta, \qquad (j = 1,2,...,n),$$

каде Δ е чекор кој мора внимателно да се избере, додека λ_i се Лагранжови множители кои се одредуваат од системот линеарни равенки (5.34).

Методата на ограничен градиент може успешно да се користи за решавање на оние задачи на НП, кај кои множеството ограничувања е линеарно или приближно линеарно, бидејќи за случај со исклучиво нелинеарни ограничувања чекорот Δ мора да се избере да биде исклучиво мала вредност, што, од друга страна, води кон нерамномерно зголемување на бројот на сметачки операции. На тој начин самата метода би ја изгубила својата практична смисла.

Друг приод кај решавањето на задача на НП од обликот (5.29) - (5.30) се состои во придодавање на функции на целната функција (5.30), а како последица се јавува проширен критериум од обликот

$$F+\frac{1}{2}\sum_{i=1}^{m}\alpha_i g_i^2, \tag{5.35}$$

каде вредностите α_i се позитивни константи. Минимизација на проширените функции (5.35) тежи кон решението на оригиналниот проблем (5.29) - (5.30). На вака формулираната задача лесно се применува градиентната метода. Константите α_i се корегираат на крајот на процесот на минимизација врз основа на отстапување на ограничувањата g_i од однапред одбраните толеранции.

Во случај кога множеството ограничувања за претходно формулираната задача на НП е зададено со неравенствата од обликот

$$g_i(x_1, x_2, \ldots, x_n) \le 0, \qquad (i=1,2,\ldots,m), \tag{5.36}$$

проблемот може да се реши со комбинирање на градиентната метода и методата на ограничен градиент.

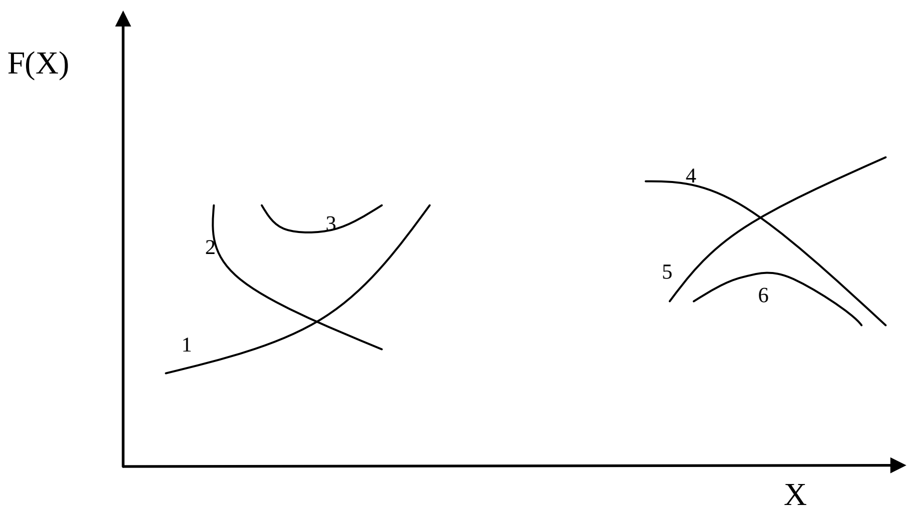
5.12 Некои специјални случаи на нелинеарно програмирање

Како што е истакнато, може да се констатира дека покрај низа линеарни зависности кои се јавуваат во математичките модели на поедини техно–економски задачи, може да бидат присутни и нелинеарните зависности. Во некои случаи на тие нелинеарни зависности можно е да се линеаризираат, но постојат примери од праксата каде што тоа не е можно да се направи на никаков начин.

Во множеството нелинеарни зависности кај поедини математички модели на НП, посебно значење за праксата, со оглед на можностите за наоѓање на решение, имаат задачите кои може да се прикажат со конвексни и конкавни функции, како и некои други специјални модели кои се лесно решливи. Овде ќе се укаже на некои од тие модели.

5.13 Квадратно програмирање

Методите за решавање на задачите на НП потполно се развиени за една специјална класа задачи, кои се карактеризираат со квадратна целна функција и линеарно множество ограничувања. Задачите на НП од оваа врста се изучуваат под името квадратно програмирање и во пракса најчесто се јавуваат кај анализи со корисен ефект (производство, добивки, итн.) што може да се прикаже со конкавна функција која е слична со кривата 5 на цртеж 5.4, или кога се во прашање анализите на трошоците, целната функција се прикажува со конвексна крива од обликот 1, 2, или 3 (види на цртеж 5.4).



Цртеж 5.4 Конкавни и конвексни функции: 1,2,3 се конвексни, а 4,5,6 се конкавни

Формулираниот соодветен математички модел на задачата на квадратно програмирање може да се искаже на следниов начин:

Потребно е да се одреди минимум или максимум на квадратната целна функција

$$F(X) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} x_j x_i, \qquad (5.40)$$

ако непознатите $x_1, x_2, ..., x_n$ го задоволуваат множеството ограничувања

$$g_i(x_1, x_2, ..., x_n) \leq 0, \qquad (i = 1,2,...,n),$$

кое е линеарно.

Така, на пример, целната функција

$$F(X)=c_{11}x_1^2+c_{12}x_1x_2+c_{22}x_2^2$$

претставува квадратна функција со две променливи x_1 и x_2.

Некои од коефициентите во целната функција (5.40) може да бидат еднакви на нула. Квадратната целна функција (5.40) е конвексна, ако коефициентите c_{ij} се такви да за било кој пар на ненегативни вредности на променливите x_i, x_j ги имаме исполнето условите

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_i x_j \geq 0. \qquad (5.41)$$

Во спротивен случај, кога знакот нееднаквост во условите (5.41) е обратен, функцијата $F(X)$ е конкавна, односно функцијата $-F(X)$ е конвексна.

Задачите на НП кои се однесуваат на максимизација на одреден корисен ефект или на минимизација на трошоци, во најголем број случаи имаат целна функција која е конвексна или конкавна. Во таа класа задача спаѓаат и задачите на квадратното програмирање, така да за нивно решавање може широко да се користи однапред изложената метода. Ако во математичкиот модел се вклучени променливите x_j $(j=1,2,...,n)$, кои претставуваат, на пример, количини на различни артикли на производство, тогаш целната функција која го одржува производството во најголем број случаи не е конвексна туку конкавна. Исто така во доволно голем временски интервал, порастот на расходот на производство почнува да добива пригушен карактер, што укажува на соодветно претставување на овие појави на квадратната функција која има негативна вредност на коефициентот со втор степен на независните променливи, така да

$$F=a+bx-cx^2.$$

Сето тоа укажува на широка можност на примена на квадратното програмирање во процесот на оптимизација на различните техничко - економски задачи, во рамките на кои се сака да се изнајдат оптималните услови за искористување на расположливите ресурси.

5.14 Сепарабилно програмирање

Основна карактеристика на сепарабилното програмирање, како една посебна класа на задачи на НП, се состои во тоа што секогаш е можно за таквите задачи целната функција да се прикаже со збирот на n функции кои во општ случај се нелинеарни, при што секоја од тие функции зависи само од една променлива ,т.е.

$$F(X) = \sum_{j=1}^{n} f_j(x_j) \qquad (5.42)$$

Спрема тоа, задачата на сепарабилно програмирање може да се формулира на следниов начин:

Потребно е да се одредат негативните вредности на $x_{ij} \geq 0$, $(j = 1,2,...,n)$ за кои целната функција (5.42) добива максимална (минимална) вредност, а при тоа да биде задоволено множеството ограничувања од обликот

$$g_i(X) = \sum_{j=1}^{n} g_{ij}(x_j) \geq 0, \qquad (i = 1,2,...,m), \qquad (5.43)$$

каде е $X = (x_1, x_2,..., x_n)$.

Очигледно е дека и множеството ограничувања од обликот (5.43), кое фигурира кај задачите на сепарабилното програмирање, мора да биде дефинирано со сепарабилните функции $g_i(X)$, кои се прикажуваат со збирот на функциите $g_i(x_j)$, од кои секоја зависи само од една променлива x_j .

Еквивалентен математички модел на задачата (5.42) - (5.43) може да се сведе на системот равенки

$$\left.\begin{aligned} \frac{\partial^2 F(X)}{\partial x_k \partial x_j} = 0, \\ \frac{\partial^2 g_i(X)}{\partial x_k \partial x_j} = 0, \end{aligned}\right\} \text{ за сите вредности на } k \neq j$$

$$(i = 1,2,...,m), \qquad (j = 1,2,...,n).$$

Јасно е дека сите задачи на линеарното програмирање спаѓаат во класата задачи на сепарабилното програмирање. Освен тоа, еден голем број задачи на квадратното програмирање може да се вклучи под сепарабилно програмирање, како и сите други задачи кои ги исполнуваат условите (5.42)-(5.43), односно (5.44).

5.15 Целобројно програмирање

Ако во општата задача на НП од обликот (5.1)-(5.2), односно (5.1')-(5.2') за променливите x_j $(j=1,2,...,n)$ може да се земаат само целобројни вредности, тогаш во прашање е една пониска класа на задачи на НП, позната под името целобројно програмирање. Посебна подкласа на задачите на целобројното програмирање ја прават задачите во кои променливите x_j $(j=1,2,...,n)$ земаат само две вредности нула или еден (0-1, или не - да). Тоа се задачи на 0-1 програмирање кои во праксата многу често се среќаваат, заради тоа ќе им посветиме посебно внимание.

5.15.1 0 - 1 програмирање

Кај решавањето на задачите на 0-1 програмирањето, заради конечниот број можни алтернативи, не се користат континуални методи за кои веќе зборувавме. За нивно решавање развиени се посебни методи, од кои некои се базираат на комбинаториката. Заради таквиот пристап на процесот за барање на решение на задачите на 0-1 програмирањето, често наместо ова вообичаено име се користи и името комбинаторно програмирање.

Постојат повеќе методи за решавање на задачите на 0-1 програмирањето. Една од најстарите методи се базира на испитување на сите можни варијанти на допуштениот план и избирање на оние варијанти кои претставуваат оптимален план. Тргнувајќи од претпоставката дека математичкиот модел на одредена задача на 0-1 програмирањето содржи $n-$променливи x_j $(j=1,2,...,n)$, од кои секоја променлива може да земе вредност 0 или 1, лесно се доаѓа до заклучокот дека бројот на можни варијанти на решението е еднаков на 2. Со оглед на тоа дека бројот на променливи x_j $(j=1,2,...,n)$ кои се појавуваат во реалните математички модели е многу голем, оваа метода за одбирање на решение, низ пребарување на сите можни варијанти, станува неупотреблива во пракса.

Другите методи за решавање на задачите на 0-1 програмирањето се состојат во исфрлање на некои варијанти од можните решенија, за кои се знае дека не може да претставуваат оптимален план. За да може попрецизно да се зборува за тие методи, најпрво потребно е да се даде математичка формулација на задачата на 0-1 програмирањето, која се состои во следново:

Потребно е да се одреди вредноста на x_j $(j=1,2,...,n)$ за која целната функција

$$F = F(x_1, x_2, ..., x_n) \tag{5.45}$$

достигнува максимална (минимална) вредност, а при тоа да биде задоволено множеството ограничувања

$$g_i(x_1, x_2, ..., x_n) \leq 0, \qquad (i = 1,2,...,m), \tag{5.46}$$

при што сите променливи x_j земаат вредности

$$x_j = \begin{cases} 0 \\ 1 \end{cases} \quad (i = 1,2,...,m), \tag{5.47}$$

Вака формулираниот модел (5.45) - (5.47) претставува општ математички модел на 0-1 програмирањето. Во овој модел условите (5.47) може да се напишат и на еден од следниве начини

$$x_j = x_j^{\,2}, \text{ или}$$

$$0 \leq x_j \leq 1,$$

при што x_j е цел број, $(j = 1,2,...,n)$.

Секое X = $(j = 1,2,...,n)$ кое ги задоволува условите (5.46) и (5.47) се нарекува допуштен план, а она $X = X^* = \left(x_1^*, x_2^*, ..., x_n^*\right)$ за кое, според задоволувањето на условите (5.46) и (5.47), целната функција (5.45) достигнува максимум (минимум), претставува оптимално решение.

Може да се забележи дака вака формулираниот математички модел се разликува од општиот математички модел на НП кој е прикажан со изразите (5.1) - (5.2), односно (5.1') - (5.2'), само во ограничувањата со облик (5.47) со кои се лимитираат променливите x_j $(j = 1,2,...,n)$.

6. Посебен случај на условна оптимизација
Конвексно програмирање

Конвексното програмирање е наједноставната и најдобро обработена област од нелинеарното програмирање. Многу својства од линеарните програми се пренесуваат на конвексните програми, како што ќе видиме во наредното поглавие.

6.1 Својства на конвексните програми

Проблемот во кој треба да најдеме точка на минимум на некоја функција $f^0(x)$ во некое зададено множество F, се нарекува програм (или математички програм). Кога f^0 е конвексна функција и F е конвексно множество, тогаш се работи за конвексен програм. Множеството F во конвексниот програм обично е зададен имплицитно на овој начин:

$$F = \{x \in R^n : f^1(x) \leq 0, f^2(x) \leq 0, \dots, f^p(x) \leq 0\}$$

каде $f^1, f^2, \dots, f^p$ се некои конвексни функции. Во овој случај конвексниот програм може да се формулира вака:

(К)

$$\begin{gathered} Min\, f^0(x) \\ p.o. \\ f^1(x) \leq 0 \\ f^2(x) \leq 0 \\ \dots\dots\dots \\ f^p(x) \leq 0. \end{gathered} \tag{6.1}$$

Функцијата f^0 се нарекува целна функција, додека $f^1, f^2, \dots, f^p$ се функции на ограничување. („Min“ значи дека се бара точка на минимум на функцијата f^0, додека „р.о.“ е скратено од зборовите „при ограничување“.) Множеството на дозволени решенија може да се напише и како

$$F = \{x \in R^n : f^i(x) \leq 0, i = 1, \dots, p\}$$

во кој случај програмот (К) го добива обликот

(К)

$$Min\, f^0(x)$$

$$p.o.$$

$$f^i(x) \leq 0, i = 1, \ldots, p.$$

Велиме дека некоја точка x е дозволено решение на програмот (К) ако x се наоѓа во F. Овде ќе бидат прикажани две важни својства на конвексните програми.

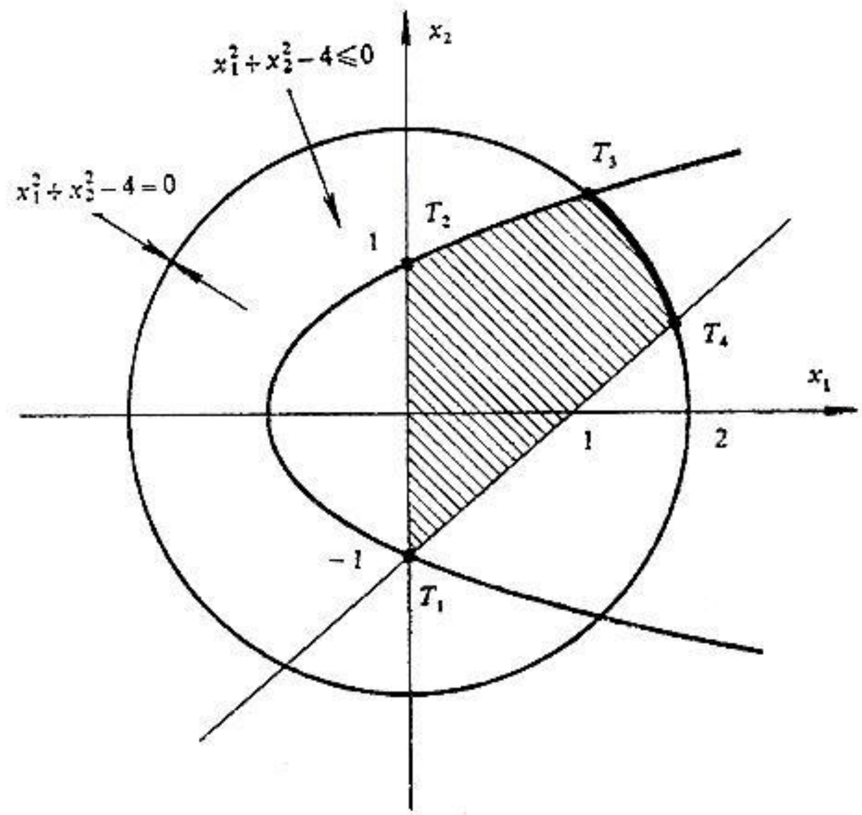
Ако се $f^i, i = 1, \ldots, p$ конвексни функции, тогаш $F = \{x \in R^n : f^i(x) \leq 0, i = 1, \ldots, p\}$ е конвексно множество.

Пример 6.1 Нека множеството на дозволени решенија F е зададено со помош на овие четири неравенки:

$$\begin{aligned} f^1 &= x_1^2 + x_2^2 - 4 \quad \leq 0 \\ f^2 &= -x_1 + x_2^2 - 1 \leq 0 \\ f^3 &= x_1 - x_2 - 1 \quad \leq 0 \\ f^4 &= -x_1 \qquad\qquad \leq 0. \end{aligned}$$

Бидејќи f^1, f^2, f^3, f^4 се конвексни функции, заклучуваме од горното својство дека множеството F е конвексно. Овој заклучок го потврдува и цртеж 6.1.

Дека навистина се работи во конвексни функции $f^i, i = 1,2,3,4$ можеме да заклучиме на неколку начини. Доволно е на пример да се провери дека сите Хесеови матрици $\nabla^2 f^i, i = 1,2,3,4$ се позитивно полу-дефинирани. Заклучокот овде уште побрзо следи ако користиме резултат кај кој збирот на конвексните функции е исто така конвексна функција. На пример f^1 е конвексна функција бидејќи f^1 е алгебарски збир на функциите x_1^2, x_2^2 и -4.



Цртеж 6.1 Множество на дозволени решенија на конвексниот програм

Во случај на произволна конвексна функција f секој локален минимум воедно е и глобален минимум. Со други зборови, не постојат две точки на локалниот минимум x и y такви да едната е подобра од другата на пример $f(x) < f(y)$. Ова својство на конвексните функции се пренесува и на конвексните програми. Нека, за некоја точка x^* ќе ја означиме нејзината околина со $N(x^*)$, т.е.

$$N(x^*) = \{x \in R^n : \|x - x^*\| \leq \varepsilon\}$$

каде $\varepsilon > 0$ е некој фиксиран (мал) број. Велиме дека некое дозволено решение x^* на програмот (К) е оптимално решение (или глобално решение) ако

$$f^0(x^*) \leq f^0(x) \text{ за секое } x \in F,$$

т.е. ако не постои дозволена точка која е „подобра" од x^*. Соодветната вредност $f^0(x^*)$ е оптимална вредност на програмот (К). Дозволеното решение x^* на програмот (К) е локално оптимално решение (или локално решение) ако е

$$f^0(x^*) \leq f^0(x) \text{ за секое } x \in F \cap N(x^*)$$

т.е. ако не постои „подобра" точка во некоја произволна „дозволена околина" на точката x^*.

Локалните решенија можат да се разликуваат од глобалните решенија само во случај на неконвексните програми. Во случај на конвексни програми нема разлика. Тоа е другото важно својство на конвексните програми:

> Секое локално оптимално решение на конвексната програма (К) воедно е и негово (глобално) оптимално решение.

Фактот дека множеството на дозволени решенија е конвексно и дека е доволно да се најде локално оптимално решение за да би се решил конвексниот програм, многу го поедноставува изучувањето на конвексните програми. Лесно е да се докаже дека множеството на сите оптимални решенија на конвексната програма е исто така конвексно множество.

Забелешка: Секој линеарен програм може да се напише во облик (6.1) каде $f^0, f^1, \dots, f^p$ се некои конвексни функции. Поради тоа сите резултати кои важат за конвексните програми, важат и за линеарните програми. (Меѓу другото, добро е познато да во линеарното програмирање множеството на дозволени решенија е конвексно, исто така и во линеарното програмирање секое локално решение е исто и глобално решение. Множеството на сите оптимални решенија.) во линеарното програмирање е исто така конвексно множество

Забелешка: Во конвексниот програм (К) од значајна важност е да сите функции се конвексни и да ограничувањата се задаени со помош на неравенки од типот „≤ 0“ (т.е. помали или еднакви на нула). Програмот од облик (К) во кој сите функции се конвексни, но со барем една неравенка во спротивна насока, т.е. од типот „$\geq$“ не е во опш случај конвексен програм. Истиот заклучок важи и за случај кога „≤ 0“ се замени со равенката „=0“. Ова тврдење ќе го прикажеме во следниот пример.

Пример 6.2 Да се вратиме пак на множеството на дозволени решенија F опишано во пример 6.1. Ако првото ограничување „$\leq$“ се замени со „=0“, тогаш новото множество на дозволени решенија, опишано со помош на релациите

$$f^1 = x_1^2 + x_2^2 - 4 \quad = 0$$
$$f^2 = -x_1 + x_2^2 - 1 \leq 0$$
$$f^3 = x_1 - x_2 - 1 \quad \leq 0$$
$$f^4 = -x_1 \qquad\qquad \leq 0$$

одговара на цртеж 6.1 само на лакот на кругот помеѓу точките T_3 и T_4; ова множество очигледно не е конвексно.

Конкавен програм. Ако f е конвексна функција, тогаш $-f$ е конкавна функција. Поради тоа конвексниот програм (К) може да се напише и вака:

$$Max\ f^0(x)$$
$$p.o.$$
$$f^i(x) \geq 0, i = 1, \dots, p$$

каде сега функциите $f^0, f^1, \dots, f^p$ се конкавни. Ова формулација на конвексниот програм, поради очигледна причина, се нарекува конкавен програм. Постои и трета формулација на конвексниот програм:

$$Min\, f^0(x)$$
$$p.o.$$
$$f^i(x) \geq 0, i = 1, \dots, p$$

каде функцијата f^0 е конвексна, додека сите функции на ограничување $f^1, \dots, f^p$ се конкавни. Се разбира, овде се работи само за различни формулации на еден ист конвексен програм. Затоа својствата кои ги има конвексниот програм важат и за „конкавниот" програм во погорната „модификација". (Важно е да се запамти да покрај конвексните функции на ограничување секогаш доаѓа „≤ 0", а кај конкавните „≥ 0", во принцип се минимизира конвексната а максимизира конкавната функција. Доколку ова „правило" се наруши, повеќе не се работи за конвексен програм и проблемот станува премногу комплициран.

6.2 Кун-Такерови услови на диференцијабилни функции

Најповеќе користени услови во конвексното програмирање, со чија помош се одредува дали некое дозволено решение е оптимално се Кун-Такерови услови. Овие услови овде можат да се изразат на два елементарини начини: „примарен" и „дуален", па затоа зборуваме за примарен и дуален услов на оптималност. Врската помеѓу овие две формулации ја воспоставуваат теоремата на алтернатива. Примарните услови за оптималност се изразуваат со помош на т.н. „дозволени насоки" во просторот на векторската променлива x. Овие услови се од конструктивна природа и затоа се користени во формулација на најстарата метода на конвексна оптимизација. Дуалните услови на оптималност се изразуваат со помош на Лагранжовите множители и се соодветни за економска интерпретација и анализа на стабилност на моделот. Во последно време дуалните услови на оптималност успешно се користат и во фурмулација на нумерички методи.

Кун-Такеровите услови на оптималност ќе ги формулираме за конвексен програм

(К)

$$Minf^0(x)$$
$$p.o.$$
$$f^1(x) \leq 0$$
$$f^2(x) \leq 0$$
$$\dots\dots\dots$$
$$f^p(x) \leq 0$$

каде сите функции се конвексни и диференцијабилни. Прво ќе ја поедноставиме нотацијата. Наместо да пишеме за сите p ограничувања, едно по едно, како погоре, ќе запишеме вака

$$f^i(x) \leq 0, i = 1, \dots, p$$

или ако го означиме множество $\{1,2,\dots,p\}$ со $\mathcal{P}$, како

$$f^i(x) \leq 0, i \in \mathcal{P}.$$

Конвексниот програм (К) сега ќе изгледа

$$Minf^0(x)$$

$$p.o.$$

$$f^i(x) \leq 0, i \in \mathcal{P}.$$

Множеството на дозволени решенија во оваа нотација е

$$F = \{x: f^i(x) \leq 0, i \in \mathcal{P}\}.$$

Пред изучувањето на самите услови на оптималност да обрнеме внимание на следново својство на програмот (К):

Ако ограничувањето во програмот (К) го задоволува Слатеровиот услов, тогаш во близина на секоја дозволена точка x^* се наоѓа Слатерова точка $\hat{z}$, т.е. дозволена точка $\hat{z}$ со својство да $f^i(\hat{z}) < 0, i \in \mathcal{P}$.

Примарен Кун-Такеров услов на оптималност. Со $\mathcal{P}(x^*)$ го означуваме множеството на активни ограничувања на дозволената точка x^*, т.е.

$$\mathcal{P}(x^*) = \{i \in \mathcal{P}: f^i(x^*) = 0\}.$$

Примарниот услов за оптималност на програмот (К) може да се дефинира на следниот начин:

Претпоставуваме дека ограничувањата во конвексниот програм (К) го задоволуваат Слатеровиот услов. Тогаш дозволеното решение x^* е оптимално ако и само ако, системот на линеарните неравенки

$$\nabla f^0(x^*)d < 0$$

$$\nabla f^i(x^*)d \leq 0, i \in \mathcal{P}(x^*) \qquad (6.2)$$

нема решение d.

Примарниот Кун-Такеров услов ќе го прикажеме на програмот од примерот 6.2 за две дозволени точки.

Пример 6.3 Треба да се одреди дали $x_1^* = 0, x_2^* = 0$ е оптимално решение на програмот

$$Minf^0 = x_1^2 - 8x_1 + x_2$$

$$p.o.$$

$$f^1 = x_1^2 + x_2^2 - 1 \leq 0$$

$$f^2 = -x_1 + x_2^2 - 1 \leq 0$$
$$f^3 = x_1 - x_2 - 1 \quad \leq 0$$
$$f^4 = -x_1 \qquad\qquad \leq 0.$$

Единствено активно ограничување за $x^* = (0,0)^T$ е четврто ограничување, т.е.

$$\mathcal{P}(x^*) = \{4\}.$$

Пресметуваме

$$\nabla f^0(x^*) = [2x_1^* - 8 \quad 1] = [-8 \quad 1]$$
$$\nabla f^4(x^*) = [-1 \quad 0].$$

Бидејќи Слатеровиот услов е задоволен, точката x^* е оптимална ако и само ако, системот (1.2) нема решение. Тој систем во нашиот пример дава

$$\nabla f^0(x^*)d = [-8 \quad 1]\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = -8d_1 + d_2 < 0$$

$$\nabla f^4(x^*)d = [-1 \quad 0]\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = -d_1 \leq 0.$$

Меѓутоа, гледаме дека постојат d_1 и d_2 такви да системот е задоволен (на пример $d_1 = 0, d_2 = -1$), што значи да $x_1^* = 0, x_2^* = 0$, не е оптимално решение.

Сега провеpуваме дали $z_1^* = \frac{1}{2}\left(1 + \sqrt{7}\right)$, $z_2^* = \frac{1}{2}(-1 + \sqrt{7})$ е оптимално.

Активни ограничувања во точката z^* се еден и три, т.е.

$$\mathcal{P}(z^*) = \{1,3\}$$

и системот (6.2) е

$$\nabla f^0(z^*)d = [2z_1^* - 8 \quad 1]\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = (\sqrt{7} - 7)d_1 + d_2 < 0$$

$$\nabla f^1(z^*)d = [2z_1^* \quad 2z_2^*]\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \left(1 - \sqrt{7}\right)d_1 + (-1 + \sqrt{7})d_2 \leq 0$$

$$\nabla f^3(z^*)d = [1 \quad -1]\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = d_1 - d_2 \leq 0.$$

Тврдиме дека овој систем не е решлив! Тврдењето може вака да се покажи: Кога би постоеле d_1 и d_2 такви да горните три неравенки важат, тогаш би имале

$$d_1 \leq d_2 < (7 - \sqrt{7})d_1 \qquad (6.3)$$

према првата и третата неравенка. Ова повлекува, после одземањето на -d_1, да е

$$0 < \left(7 - \sqrt{7}\right)d_1 - d_1 = \left(6 - \sqrt{7}\right)d_1.$$

Бидејќи $6 - \sqrt{7} > 0$, заклучуваме дека d_1 мора да е позитивно:

$$d_1 > 0. \qquad (6.4)$$

Меѓутоа, према левата неравенка во (6.3), сега заклучуваме дека и d_2 мора да биде позитивно:

$$d_2 > 0. \tag{6.5}$$

Ова дава контрадикција! Не може да важи (6.4) и (6.5) и да имаме

$$\nabla f^1(z^*)d = \left(1+\sqrt{7}\right)d_1 + (-1+\sqrt{7})d_2 \leq 0.$$

Значи, системот (6.2) овде навистина не е решлив, што значи да $z_1^* = \frac{1}{2}\left(1+\sqrt{7}\right)$

$z_2^* = \frac{1}{2}\left(-1+\sqrt{7}\right)$ е оптимално решение.

Забелешка: Во случај кога некое дозволено решение x^* не е оптимално, тогаш секој вектор d кој го решава системот (6.2) дава насока $x^* + ad, a > 0$ во која се наоѓаат подобри дозволени решенија. Овој факт ќе го користиме за формулација на нумерички методи за решавање на проблемот (К).

Дуален Кун-Такеров услов на оптималност. Дуалните услови на оптималност се добиваат од примарните услови на оптималност со примена на соодветна теорема од теоремите на алтернатива. Во случај на Кун-Такеровиот услов ќе се применува Фаркашовата теорема.

Примарниот услов вели дека ако претпоставката Слатеровиот услов да е задоволен е исполнета, некое дозволено x^* е оптимално ако и само ако, системот (6.2) нема решение. Без загуба на севкупноста (и да го избегнеме внесувањето на двојни индекси) можеме да претпоставиме дека првите r ограничувања се активни во точката x^*. Така да системот (6.2) може да се прикаже во обликот

$$A^T d \geq 0, \;\; (b,d) < 0 \tag{6.6}$$

каде имаме означено

$$A^T = \begin{bmatrix} -\nabla f^1(x^*) \\ -\nabla f^2(x^*) \\ \cdot \\ \cdot \\ \cdot \\ -\nabla f^r(x^*) \end{bmatrix}, \;\; b^T = \nabla f^0(x^*).$$

(Овде A^T означува транспонирана матрица на матрицата A, b^T е транспониран вектор на векторот b, $A^T d \geq 0$ значи дека секоја компонента на векторот $A^T b$ е поголема или еднаква на нула, додека $(b,d) = b^T d$ е скаларен производ на векторите b и d.) Спрема Фаркашовата теорема, системот (6.6) нема решение ако и само ако, системот

$$A\lambda = b, \;\; \lambda \geq 0 \tag{6.7}$$

има решение. Во нотацијата на градиентот, системот (6.7) е

$$(-[\nabla f^1(x^*)]^T, \; -[\nabla f^2(x^*)]^T, \dots, \; -[\nabla f^r(x^*)]^T)\lambda = [\nabla f^0(x^*)]^T, \; \lambda \geq 0$$

или, после множењето со векторот $\lambda = (\lambda_i)$ и средувањето се добива

$$[\nabla f^0(x^*)]^T + \lambda_1[\nabla f^1(x^*)]^T + \lambda_2[\nabla f^2(x^*)]^T + \cdots + \lambda_r[\nabla f^r(x^*)]^T = 0$$

$$\lambda_1 \geq 0, \lambda_2 \geq 0, \ldots, \lambda_r \geq 0.$$

Поради естетски причини можеме горната равенка да ја транспонираме, и добиваме

$$\nabla f^0(x^*) + \lambda_1 \nabla f^1(x^*) + \lambda_2 \nabla f^2(x^*) + \cdots + \lambda_r \nabla f^r(x^*) = 0$$

$$\lambda_1 \geq 0, \lambda_2 \geq 0, \ldots, \lambda_r \geq 0.$$

Следи, ако изразот го напишеме со користење на симболот за сума имаме:

$$\nabla f^0(x^*) + \sum_{i=1}^{r} \lambda_i \nabla f^i(x^*) = 0$$

$$\lambda_i \geq 0, i = 1, \ldots, r.$$

Ако, претпоставката дека првите r ограничувања активни во точката x^* може да се замени во доказот со било кој редослед на активните ограничувања, горниот систем може во општ случај да се прикажи во облик

(К-Т)

$$\nabla f^0(x^*) + \sum_{i \in \mathcal{P}(x^*)} \lambda_i \nabla f^i(x^*) = 0$$

$$\lambda_i \geq 0,\ i \in \mathcal{P}(x^*).$$

Овде симболот $\sum_{i \in \mathcal{P}(x^*)}$ значи да се сумира преку сите индекси активни на аграничувањето $\mathcal{P}(x^*)$. Така го докажавме Кун-Такеровиот услов на оптималност: Претпоставуваме да ограничувањата во конвексниот програм (К), со диференцијабилните функции, го задоволуваат Слатеровиот услов. Тогаш дозволеното решение x^* е оптимално ако и само ако, системот (К-Т) е решлив т.е. ако и само ако, постојат ненегативни броеви $\lambda_i \geq 0, i \in \mathcal{P}(x^*)$ такви да

$$\nabla f^0(x^*) + \sum_{i \in \mathcal{P}(x^*)} \lambda_i \nabla f^i(x^*) = 0.$$

Забелешка:

1. Во литературата на нелинеарното програмирање, под Кун-Такеров услов, за проблем (К) обично се подразбира дуалниот Кун-Такеров услов. Причината е во тоа што дуалниот услов е поважен за економска интерпретација од примарниот услов.
2. Системот (К-Т) може да се замени со еквивалентниот систем, кој вклучува и неактивни ограничувања:

$$\nabla f^0(x^*) + \sum_{i \in \mathcal{P}} \lambda_i \nabla f^i(x^*) = 0$$

$$\lambda_i f^i(x^*) = 0, \lambda_i \geq 0, i \in \mathcal{P}. \tag{6.8}$$

Бидејќи за неактивно ограничување во точката x^* мора да биде $f^i(x^*) < 0$ јасно е да од $\lambda_i f^i(x^*) = 0$ следи дека $\lambda_i = 0$. Затоа двата системи се навистина еквивалентни. Релациите

$$\lambda_i f^i(x^*) = 0, i \in \mathcal{P}$$

Во (6.8) се т.н. „комплементарни услови“.

3. Во случај на конвексен програм во кој се бара да променливите $x_i, i = 1, \dots, n$ бидат ненегативни:

$$\begin{gathered} Minf^0(x) \\ p.o. \\ f^i(x) \le 0, i = 1, \dots, p \\ x_1 \ge 0, \dots, x_n \ge 0 \end{gathered} \tag{6.9}$$

секое $x_i \ge 0, i = 1, \dots, n$ треба да се третира како додатно ограничување. Во тој случај конвексниот програм изгледа вака:

$$\begin{gathered} Minf^0(x) \\ p.o. \\ f^i(x) \le 0, i = 1, \dots, p \\ f^{p+i}(x) = -x_i \le 0, i = 1, \dots, n. \end{gathered}$$

Се работи за конвексен програм од облик (К) со $p + n$ ограничувања. Бидејќи последните n ограничувања се едноставни функции, Кун-Такеровиот услов (К-Т) за програмот (6.9) има едноставна структура.

4. Доколку дуалниот Кун-Такеров услов е задоволен во некоја дозволена точка x^*, тогаш x^* е оптимална, без разлика дали Слатеровиот услов е задоволен. Ова значи, дека Слатеровиот услов е важен само во докажување на потребите на Кун-Такеровиот услов.

6.3 Методи за решавање на конвексни програми

Градиент метода

Градиент методата е модификација на Кошиевата метода со најстрмен пад за програмите со ограничувања. Поради едноставност, претпоставуваме дека треба да решиме конвексен програм со линеарни ограничувања

$$\begin{gathered} Minf(x) \\ p.o. \\ Ax \le b \\ x \ge 0 \end{gathered} \tag{6.10}$$

каде f е некоја конвексна непрекината диференцијабилна функција. Го означуваме со F множеството на дозволени решенија

$$F = \{x: Ax \leq b, x \geq 0\}.$$

Методата е итеративна, т.е. оптималното решение x^* на програмот (6.10) е гранична точка на низата со апроксимација x^k, $к = 0,1,2, \ldots$ Секоја апроксимација x^k се добива по решавањето на линеарната програма и еднодимензионалното пребарување.

Алгоритам

1. Се одбира почетна дозволена апроксимација $x^0 \in F$. Се пресметува градиентот $\nabla f(x^0)$ и се специфицира правилото назастанување, на пример „доволно мал“ број $\varepsilon > 0$ со својство да алгоритамот застане кога е

$$\|x^{k+1} - x^k\| < \varepsilon. \tag{6.11}$$

За $к = 0,1,2, \ldots$:

2. Се решава линеарниот програм

$$\begin{gathered} Min \nabla f(x^k) \\ p.o. \\ Ax \leq b \\ x \geq 0 \end{gathered} \tag{6.12}$$

Се означете со $\bar{x}^k$ неговото оптимално решение.

3. Се решава еднодимензионалниот програм

$$\begin{gathered} Min f(x^k + \lambda(\bar{x}^k - x^k)) \\ p.o. \\ 0 \leq \lambda \leq 1. \end{gathered}$$

(Овде f се пребарува на линискиот сегмент кој ги спојува x^k и $\bar{x}^k$.)
Се означува со λ_k неговото оптимално решение.

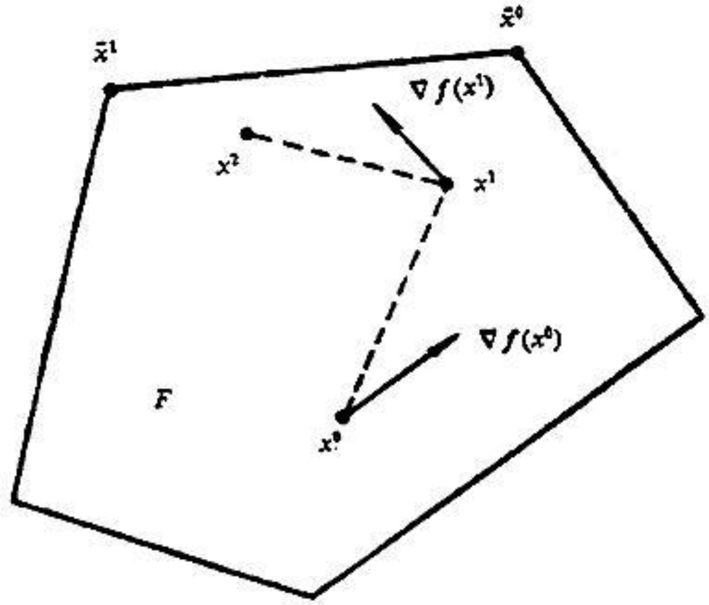
4. Се пресметува новата апроксимација

$$x^{k+1} = x^k + \lambda_k(\bar{x}^k - x^k).$$

5. Ако правилото на застанување е задоволено, процесот се стопира, x^{k+1} е прифатлива апроксимација на оптималното решение. Во спротивно, се

враќаме на чекор 2, го пресметуваме $\nabla f(x^{k+1})$ и продолжуваме со алгоритамот за ново x^{k+1}.

Методата графички е опишана на цртеж 6.2. Бидејќи (6.12) е линеарен програм, неговото решение x^k може да се земи како екстремна точка на множеството дозволени решенија F.



цртеж 6.2 Градиент метода

6.4 Метода на конвергенција

Доказот да некоја метода вистински конвергира на своето решение не е обично едноставен. Ова тврдење ќе го демонстрираме на Градиент методата, чиј доказ на конвергенција е типичен но и едноставен во однос на другите методи. Сакаме да го докажеме следниот резултат:

Ако $f(x)$ е конвексна непрекината диференцијабилна функција и ако е множеството дозволени решенија F ограничено, тогаш секоја гранична точка x^* на низата $\{x^k\}$ на пресметаната Градиент метода е глобален минимум на програмот (6.1).

Бидејќи x^* е гранична точка на низата $\{x^k\}$, а множеството F е ограничено (и затворено), постои подниза $\{x^{k_l}\}$ на низата $\{x^k\}$ со својство

$$\lim_{l\to\infty} x^{k_l} = x^*.$$

За секое x^{k_l}, методата одредува $\bar{x}^{k_l}$ како решение на линеарниот програм (6.12). Без загуба на едноставноста можеме да претпоставиме дека низите $\{\bar{x}^{k_l}\}$ и $\{\bar{x}^{k_l+1}\}$ се конвергетни. Ги означуваме нивните гранични точки со $\bar{x}^*$ и x^{**}. Бидејќи $\bar{x}^{k_l+1}$ ја минимизира f на интервалот $[x^{k_l}, \bar{x}^{k_l}]$, поради непрекинатоста на функцијата f следи да x^{**} ја минимизира f на интервалот $[x^*, \bar{x}^*]$. Значи

$$f(x^*) \le f(x) \text{ за секое } x = x^* + \lambda(\bar{x}^* - x^*), 0 \le \lambda \le 1. \quad (6.13)$$

Специјално, $f(x^{**}) \le f(x^*)$. Меѓутоа $f(x^{**}) < f(x^*)$ е невозможно бидејќи низ $\{f(x^{k_l})\}$ може само да опаѓа. Така да $f(x^{**}) = f(x^*)$ и (6.13) постанува

$$f(x^*) \le f(x) \text{ за секое } x = x^* + \lambda(\bar{x}^* - x^*), 0 \le \lambda \le 1. \quad (6.14)$$

Сега ќе ја напишеме добро познатата Тајлорова формула за точката x:

$$f(x) = f(x^*) + \lambda\nabla f(x^*)(\bar{x}^* - x^*) + O(\lambda) \quad (6.15)$$

каде $O(\lambda)$ е „функција на остаток“ со својство

$$\lim_{\lambda\to 0} \frac{O(\lambda)}{\lambda} = 0. \quad (6.16)$$

Оваа формула, поради (6.14) повлекува

$$\lambda\nabla f(x^*)(\bar{x}^* - x^*) + O(\lambda) \ge 0.$$

По делењето со $\lambda > 0$ и $\lambda \to 0$ имаме

$$\nabla f(x^*)(\bar{x}^* - x^*) \ge 0. \quad (6.17)$$

Неравенката (6.17) користи само две специфични точки $\bar{x}^*$ и x^*. За да би се приклучиле на неравенката дозволените произволни решенија, прво треба да примениме дека

$$\nabla f(x^{k_l})(x - x^{k_l}) \ge \nabla f(x^{k_l})(\bar{x}^{k_l} - x^{k_l}) \text{ за секое } x \in F.$$

(Ова е вистина бидејќи $\bar{x}^{k_l}$ е минимално решение на програмот (6.12).) Бидејќи f е непрекината диференцијабилна функција, во граничен случај кога $l \to \infty$, горниот израз дава за секое $x \in F$:

$$\nabla f(x^*)(x - x^*) \ge \nabla f(x^*)(\bar{x}^* - x^*)$$

и конечно, поради тоа што

$$\nabla f(x^*)(x - x^*) \ge 0. \quad (6.18)$$

оптималноста на точката x^* сега е неминовна: За секое $x \in F$ имаме

$$f(x) \ge f(x^*) + \nabla f(x^*)(x - x^*)$$

поради конвексноста на функцијата f и овде

$$f(x) \ge f(x^*)$$

поради (6.18). Значи x^* е глобален минимум.

Пример 6.4 Треба да се реши програмот

$$Min f(x) = (x_1 - 2)^2 + (x_2 - \tfrac{1}{2})^2 + x_1^2 x_2^2$$

$$p.o.$$
$$0 \leq x_1 \leq 1.$$
$$0 \leq x_2 \leq 1.$$

Прво ќе покажеме дека целната функција $f(x)$ е конвексна на множеството дозволени решенија F.

Хесеовата матрица на функцијата е

$$\nabla^2 f(x) = 2\begin{bmatrix} 1+x_1^2 & 2x_1x_2 \\ 2x_1x_2 & 1+x_2^2 \end{bmatrix}.$$

Дијагоналните минори на оваа матрица се ненегативни на F, бидејќи е

$$1 + x_1^2 > 0$$
$$1 + x_2^2 > 0$$
$$(1+x_1^2)(1+x_2^2) - 4x_1^2x_2^2 = 1 + x_1^2 + x_2^2 - 3x_1^2x_2^2 \geq 0.$$

Матрицата $\nabla^2 f(x)$ е позитивно семидефинитна на F.

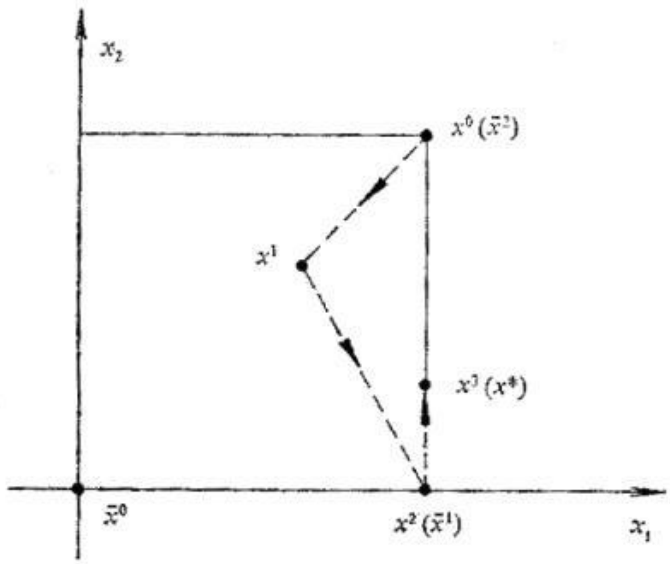
Ова повлекува да f е конвексна функција на F.

Ако се почне итерацијата од на пример $x^0 = (1,1)^T$, оптималното решение го наоѓаме после само три итерации што е прикажано на **табела 6.1.**

Табела 6.1

k	x_1^k	x_2^k	λ_k	$f(x^k)$
0	1,0000	1,0000	0,3379	2,2500
1	0,6621	0,6621	1,0000	2,0084
2	1,0000	0,0000	0,2500	1,2500
3	1,0000	0,2500	0,0000	1,1250

Апроксимациите $x^0, x^1, x^2, x^3 = x^*$ прикажани се на цртеж 6.3**.**



цртеж 6.3

Оптималноста на точката $x^* = \begin{bmatrix} 1 \\ \frac{1}{4} \end{bmatrix}$ можеме лесно да ја провериме со помош на Кун-Такеровиот услов. Прво, ограничувањата ги запишуваме во облик кој одговара на програмот (К):

$$f^1(x) = x_1 - 1 \leq 0$$
$$f^2(x) = -x_1 \quad \leq 0$$
$$f^3(x) = x_2 - 1 \leq 0$$
$$f^4(x) = -x_2 \quad \leq 0.$$

Во точката $x = x^*$ само првото ограничување е активно, т.е. $\mathcal{P}(x^*) = \{1\}$. Затоа дуалниот Кун-Такеров систем има облик

$$\nabla f(x^*) + \lambda_1 \nabla f^1(x^*) = 0$$
$$\lambda_1 \geq 0. \tag{6.19}$$

Бидејќи е

$$\nabla f(x^*) = \left[2(x_1 - 2) + 2x_1x_2^2,\ 2\left(x_2 - \frac{1}{2}\right) + 2x_1^2x_2\right] | x = x^*$$

$$= \left[-\frac{15}{8},\ 0\right]$$

додека е

$$\nabla f^1(x^*) = [1, 0]$$

Кун-Такеровиот систем (6.19), после транспонирањето на векторот, ќе изгледа вака:

$$\begin{bmatrix} -\frac{15}{8} \\ 0 \end{bmatrix} + \lambda_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\lambda_1 \geq 0.$$

Очигледно дека овој систем има решение $\left(\lambda_1 = \frac{15}{8}\right)$. Заклучок: $x_1^* = 1, x_2^* = \frac{1}{4}$ е навистина оптимално решение.

Забелешка: Градиент методата може да се користи и за решавање на програми за конвексни непрекинати диференцијабилни целни функции и произволно конвексно множество на дозволени решенија F. Конвексноста на множеството F не смее да се игнорира, бидејќи во спротивно можеме да налетаме на апроксимација x^k која не е во множеството F. Важно својство на Градиент методата е да сите апроксимации на оптималното решение се наоѓаат во множеството на дозволени решенија. Методата има слични својства како и Кошиевата метода на најстрмен пад, т.е. се работи за поспора но во главно по сигурна метода.

7. Примена на нелинеарното програмирање

Во овој дел ќе бидат разгледани проблеми поврзани со примената на оптимизацијата.

7.1 Нелинеарен транспортен проблем

Транспортните проблеми во најголем број на случаи се врзани за избор на најповолна варијанта на транспорт која обезбедува трошоците на транспорт да бидат минимални во однос на определена сообраќајна мрежа и транспорни средства. Сепак денес под тој поим се подразбираат и задачите за оптимално разместување на машини, помошни служби, енергетски објекти и друго со цел да се постигне поголема економичност на работата и времето.

Класичниот транспортен проблем се состои во наоѓање на најекономичен план на превоз на производ од еден вид од местото на негово производство до местото на потрошување. Ако во m-складишта има $a_1, a_2, ..., a_m$ количини, а во $n-$ продавници се бараат $b_1, b_2, ..., b_n$ количини од еден производ и цената на транспортот е $c_{ij}, (i = 1,2,...,m)$ и $(j = 1,2,...,n)$, за единица производ од i-тото складиште до j-тата продавница, проблемот за наоѓање економичен план на превоз се сведува на решавање на задачата

$$\min F(x_{ij}) = \min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

со ограничувања

$$\sum_{j=1}^{n} x_{ij} = a_i, \qquad \sum_{i=1}^{m} x_{ij} = b_j,$$

каде x_{ij} е количината на производот која треба да се транспортира од i-тото складиште до j-тата продавница. При тоа $x_{ij} \geq 0$. Вака поставениот проблем е проблем на линеарното програмирање во кој трошоците на транспорт $\varphi_{ij}(x)$ се пропорционални на соодветната количина x_{ij} која се транспортира т.е. $\varphi_{ij}(x) = c_{ij} \; x_{ij}$.

Таа претпоставка доведува соодветната целна функција да добие линеарна зависност

$$F = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}.$$

Меѓутоа, во низата практични задачи кои се однесуваат на решавањето на транспортните проблеми, за претпоставката за која станува збор претходно, бидејќи се работи за трошоци кои се нелинеарни во однос на количината на производи кои се транспортираат. Така, на пример, трошоците на транспортот на маршутата $(i-j)$ може да се менуваат по квадратен закон

$$\varphi_{ij}(x) = a_{ij}\, x_{ij}^{\,2} + b_{ij} x_{ij} \,,$$

при што целната функција може да се напише во обликот

$$F = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(a_{ij}\, x_{ij}^{\,2} + b_{ij} x_{ij} \right).$$

Целната функција, во зависност од природата на транспортниот проблем, може да биде и од некој друг облик на нелинеарна функција. Во одредени случаи, таа може да биде зададена со множеството дискретни вредности, при што за одредени вредности на x имаме зададени вредности на $\varphi_{ij}(x)$. Во тој случај аналитичката зависност на $\varphi_{ij}(x)$ се одредува со помош на интерполација.

Независно од тоа за каква нелинеарна целна функција се работи, задачата за наоѓање на оптималниот план на транспортот се сведува на минимизација зададена со целната функција со облик

$$F = \sum_{i=1}^{m} \sum_{j=1}^{n} \varphi_{ij}(x_{ij}),$$

каде $\varphi_{ij}(x_{ij})$ се зададени нелинеарни функции кои зависат од еден аргумент x_{ij} (т.н. сепарабилни функции), при што непознатите x_{ij} мораат да го задоволуваат множеството ограничувања со облик

$$\sum_{j=1}^{n} x_{ij} = a_i, \qquad (i = 1,2,\ldots,m),$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \qquad (j = 1,2,\ldots,n),$$

$$x_{ij} \geq 0 \,.$$

Вака формулираната задача спаѓа во класата задачи на НП и се нарекува нелинеарна транспортна задача.

7.2 Нелинеарни задачи за распределба на еднородни ресурси

Задачите на распределба на различните категории на ресурси по својата природа се многу блиски со транспортните проблеми, со оглед на тоа дека и во едниот и во другиот случај соодветните математички модели се слични или потполно изедначени.

Во поедноставните случаи на распределбата на ресурсите се тргнува од претпоставката дека ресурсите се еднородни. Задачата станува доста посложена кога е потребно да се одреди оптималната распределба на разнородните ресурси.

За случај кога е потребно да се најде оптимална распределба на еднородните ресурси, соодветниот математички модел може лесно да се формулира ако се воведат следниве ознаки:

a - вкупна количина на еднороден ресурс кој треба да се распредели;

x_j - количина на ресурсот кој се распоредува на j-тата гранка (активност, објект), $(j=1,2,...,n)$;

$\varphi_j(x_j)$ - функција која го прикажува ефектот за искористување на ресурсот (средства) на j-тата гранка.

Спрема тоа, вкупниот ефект на искористување на целокупната количина на еднородниот ресурс е еднаков на збирот на парцијалните ефекти кои се постигнуваат на било која j-та гранка (активности, објекти), така да целната функција може да се напише во обликот на сепарабилните функции

$$F(X)=\sum_{j=1}^{n}\varphi_j(x_j).$$

Задачата се состои во наоѓање на оптимален план $X=(x_1,x_2,...,x_n)$, за кој дефинираната целна функција ќе добие максимална вредност, а при тоа да бидат задоволени условите

$$\sum_{j=1}^{n}x_j=a,$$

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

Во зависност од природата на конкретната задача, вредностите на x_j може да бидат произволни или целобројни ненегативни вредности (случај кога ресурсот кој се распределува е неделив - цел број).

Во случај кога функциите $\varphi_j(x_j)$ кои фигурираат во целната функција $F(X)$ се линеарни функции од обликот $\varphi_j(x_j) = a_j x_j + b_j$, тогаш решението на поставената задача е очигледно, бидејќи е јасно дека сите средства (ресурси) треба да се вложат само во онаа j-та гранка (работност, објект) каде ефикасноста на вложување е најголема, т.е. каде што е најголемо a_j. Меѓутоа, тоа не е случај кога функциите $\varphi_j(x_j)$ се нелинеарни.

Заради илустрација на изложениот математички модел ќе се послужиме со примерот од теоријата на доверливост (задача на оптимално резервирање). Имено, нека сложен технички уред се карактеризира со множеството недоволно доверливи елементи кои ги има вкупно n. Понатаму, нека доверливоста (веројатност за безотказна работа) на j-тиот елемент (блок) е еднаква на p_j. За зголемувањето на доверливоста на уредите кои се составени од релативно недоверливи елементи (блокови), во краен случај, неопходно е по пат на резервирање (дуплирање, триплирање, итн.) на поедини елементи (блокови) да се постигне сакананата доверливост на уредите (системите), поготово ако технолошките и другите можности се исцрпени. Нека x_j е бројот на j-тите резервирани елементи (блокови), тогаш доверливоста на j-тиот елемент (блок) е зголемена за вредност

$$1-\left(1-p_j\right)^{x_j},$$

додека веројатноста на отказите q_j после резервирање на j-тиот елемент за x_j-пати е еднаков на

$$q_j=\left(1-p_j\right)^{x_j}.$$

Под претпоставка дека вкупниот број на расположливи еднородни елементи (блокови) кои треба да послужат за резервирање е еднаков на $a>n$, се поставува прашањето на нивната распределба на n-блокови на еден уред. Земајќи во предвид дека трошоците на ремонтот на било кој j-ти блок се c_j $\left(j=1,2,\ldots,n\right)$, математичкото очекување на вкупните трошоци на ремонтот ќе биде зададено со функцијата

$$F\left(X\right)=\sum_{j=1}^{n}c_j\left(1-p_j\right)^{x_j},$$

која треба да се минимизира при ограничувањата со облик

$$\sum_{j=1}^{n}x_j=a,$$

$$x_j\geq 0,\quad \left(j=1,2,\ldots,n\right).$$

Барањето на минимум на дадената целна функција при зададеното множество ограничувања е идентично со барањето на максимумот на функцијата

$$\Phi\left(X\right)=\sum_{j=1}^{n}c_j\left[1-\left(1-p_j\right)^{x_j}\right]$$

при истите ограничувања.

Очигледно е дека вака формулираната задача на оптималното резервирање се јавува како посебен случај на задача за распределба на еднороден ресурс, каде сепарабилните функции $\varphi_j(x_j)$ се дефинирани со изразите

$$\varphi_j(x_j) = c_j\left[1-\left(1-p_j\right)^{x_j}\right].$$

Вредностите x_j за вака формулирана задача мораат да бидат целобројни. Меѓутоа, ако ресурсот кој се распределува би имал значење на пари, електрична енергија, итн. тогаш никогаш не се поставува барање вредностите на x_j дека мораат да бидат целобројни.

7.3 Нелинеарни задачи за распределба на нееднородни ресурси

Задачата за распределба на нееднородни ресурси претставува , во однос на задачата за распределба на еднородни ресурси, далеку посложена задача.

Ќе претпоставиме дека постојат m категории на различни ресурси чии количини се b_i $(i=1,2,...,m)$. Нека x_{ij} е количина на i-тите категории на ресурси кој се распоредуваат на j-тата гранка (активност, објект). Очигледно е дека непознатите величини x_{ij} мора да ги задоволуваат условите

$$\sum_{j=1}^{n} x_{ij} = b_i, \qquad (i=1,2,...,m),$$

$$x_{ij} \geq 0 \,,$$

со кои во конкретните случаи на распределба на одредените категории на ресурси може да се придодадат и други услови како, на пример, дека непознатите x_{ij} мора да бидат целобројни, итн.

Со воведување на ознаката a_{ij} - која покажува колку единици на i-тиот ресурс заменуваат една единица од реалниот ресурс на j-тиот објект (на пример, ако се работи за нееднороден ресурс кој претставува употреба на различни категории горива), може да се констатира дека на j-тиот објект ќе му биде доделено

$$\sum_{i=1}^{n} a_{ij} x_{ij}$$

единици на реалниот ресурс. На тој начин, како и кај распределбата на еднородниот ресурс, ефектот кој се добива на j-тиот објект, зависи од еквивалентните величини на реалниот ресурс кој на него се распределува, така да целната функција може да се напише во обликот

$$F = \sum_{j=1}^{n} \varphi_j \left(\sum_{i=1}^{m} a_{ij} x_{ij} \right),$$

кој треба да се максимизира, земајќи ги во предвид однапред формулираните ограничувања.

За случај кога функциите $\varphi_j(x_j)$ се линеарни, т.е. $\varphi_j(x_j) = a_j\ x + \beta_j$, соодветната целна функција може да се напише во обликот

$$F = \sum_{j=1}^{n} \sum_{i=1}^{m} \alpha_j a_{ij} x_{ij} + const.$$

Овде производот на константите $\alpha_j\ a_{ij}$ покажува каков е ефектот на распределба на единици од i-тиот ресурс на j-тиот објект (активност,процес, гранка, итн.)

7.4 Примена на нелинеарното програмирање во изборот на асортиманот на производство

Со пораст на обемот на производство, по правило, цената на производот треба да опаѓа, што може да се изрази со релацијата

$$c_j = c_j^{(0)} - a_j (x_j - b_j),$$

каде

b_j - почетен обем на производство на j-тиот артикл,

c_j $^{(0)}$ - почетно ниво на цена кое одговара на b_j,

c_j - ниво на цена кое одговара на реализација на производство x_j,

a_j - коефициент на зависност помеѓу цената на j-тиот артикл и неговата побарувачка.

Обемот на доходот или добивката од производството на одреден асортиман на артиклот, или во склоп на извршувањето на одредени услуги, може да се изрази со нелинеарна (квадратна) целна функција

$$F(X) = \sum_{i=1}^{n} c_j x_j = \sum_{j=1}^{n} \left(c_j^{(0)} + a_j b_j \right) x_j - \sum_{j=1}^{n} a_j x_j^{\ 2} \ ,$$

која треба да се максимизира, односно да се најде онаа вредност на
n-димензионалниот вектор $X = (x_1, x_2, ..., x_n)$, за која функцијата $F(X)$ постигнува максимална вредност, а притоа да бидат задоволени ограничувањата

$$\sum_{j=1}^{n} b_{ij} x_j \le d_i, \quad (i = 1,2,...,m).$$

Во множеството дефинирани ограничувања воведените ознаки ги имаат следниве значења:

b_{ij} - количество на ресурс на i-тиот тип кој се троши на производство на единици на j-тиот артикл,

d_i - ограничени ресурси на i-тиот тип.

Без оглед на тоа што во рамките на овој модел ограничувањата се линеарни, заради нелинеарниот облик целната функција $F(X)$, дефинираната задача за одбирање на оптималниот асортиман на производство спаѓа во класата задачи на НП.

7.5 Примена на нелинеарното програмирање во оптимизација на производство, увоз и извоз

Ќе претпоставиме дека се во прашање $n-$ различни производи за кои е потребно да се решава прашањето за сопствено производство, увоз и извоз, така што потребите во количини на поедините производи, со уважување на одредени ограничувања, би биле задоволени на оптимален начин.

За таа цел нека е:

x_j - количина на j-тиот производ кој треба да се обезбеди со сопствено производство, $(j = 1,2,...,n)$;

y_j - количина на j-тиот производ кој треба да се увезе,

z_j - количина на j-тиот производ кој треба да се извезе,

Коефициентите и параметрите кои се појавуваат во соодветниот математички модел ги имаат следниве значења:

a_{ij} - коефициент на i-тата категорија на трошоци на j-тиот производ (односно, трошокот на i-тата категорија на средствата на производство на единици на j-тиот производ),

b_j - трошоци за работа кои настануваат во склоп на производството на единици на j-тиот производ,

c_j - вложен капитал со цел обезбедување на производство на една единица на j-тиот производ,

d_j - увозна цена на единица на j-тиот производ,

e_j - извозна цена на единица на j-тиот производ,

A_j - вкупна потребна количина на j-тиот производ,

B - вкупна расположлива работа,

C - максимален допуштен дефицит во трговијата.

Природно, сите параметри на соодветниот математички модел мора да бидат ненегативни.

Математичката формулација на поставената задача се состои во наоѓање на минимално вложување, со посредство на минимизација на целната функција

$$F(X)=\sum_{j=1}^{n} c_j x_j, \tag{7.1}$$

за следниве ограничувања:

$$x_j \geq 0, \quad y_j \geq 0, \quad z_j \geq 0, \quad (j=1,2,...,n), \tag{7.2}$$

$$x_j + y_j - z_j - \sum_{k=1}^{n} a_{jk} x_k \geq A_j, \quad (j=1,2,...,n) \tag{7.3}$$

$$C + \sum_{j=1}^{n} e_j z_j - \sum_{j=1}^{n} d_j y_j \geq 0, \tag{7.4}$$

$$B - \sum_{j=1}^{n} b_j x_j \geq 0, \tag{7.5}$$

$$a_{ij} \geq 0, \quad (i,j=1,2,...,n); \quad \sum_{i=1}^{n} a_{ij} < 1, \quad (j=1,2,...,n), \tag{7.6}$$

$$e_j = \alpha_j + \beta_j z_j; \quad \alpha_j > 0, \beta_j > 0, \quad (j=1,2,...,n), \tag{7.7}$$

Ограничувањата (7.2) укажуваат на ненегативност на непознатите, додека ограничувањата (7.3) укажуваат на услови кои мора да го задоволат производството, увозот и извозот. Со други зборови, ограничувањата (7.3) покажуваат дека вкупната количина на произведените x_j и увезените y_j производи, т.е. ($x_j + y_j$), намалена за количината z_j - на извезените и количината $\sum_{k=1}^{n} a_{jk} x_k$ - на производи за сопствени потреби, мора да биде поголема или во краен случај еднаква на A_j - вкупна потребна количина на j-тиот производ.

Ограничувањето (7.4) укажува на билансот на надворешната трговија, со оглед на тоа дека

$$\sum_{j=1}^{n} e_j z_j.$$

има значење на добивка од извозот, а

$$\sum_{j=1}^{n} d_j y_j$$

трошоци на увозот, додека C - максимален допуштен дефицит.

Ограничувањето (7.5) се дефинира како понудена работа. Линеарната форма на изразот (7.7) претставува еден од најчестите случаи на зависност.

Вака дефинираниот математички модел спаѓа во класата задачи на НП затоа што ограничувањата му се нелинеарни. Лесно може да се забележи дека единствената нелинеарност се содржи во (7.7) и (7.4).

Функцијата (1.1) која се максимизира претставува конвексна функција по x_j $(j=1,2,...,n)$, додека множеството дефинирано со тројката (X,Y,Z), која го задоволува множеството ограничувања (1.2) - (1.7), е исто така конвексно.

8. Заклучок

Како во повеќе области, така и во областа на управувачките методи (статичко управувачки задачи) постојат повеќе методи на оптимизација со чија помош може да се решат задачите на НП.

Голем број методи врз чија основа се развиени многу алгоритми имаат големо значење за задачите на НП. Една од тие е градиентната метода која претставува метода на систематско пребарување и наоѓање на решение. Се користи за одредување на минимум на функција со две независни променливи, но исто така и за функција со $n-$независни променливи.

Останатите методи кој овде се споменати се: диференцијалната градиентна метода, методата на секачки рамнини, методите на линеаризација, како и методата на ограничен градиент и методата на проширени целни функции со ограничувања на функциите.

Задачите на НП покриваат доста широко подрачје од управувачките задачи и се поразновидни од задачите кои се сведуваат на примена на ЛП. Многу од нив се уште не се решливи од таа причина што не постојат развиени алгоритми чија примена би можела да даде одредени ефекти. Применливоста на поедини алгоритми се проценува врз основа на бројот на сметачките операции кои треба да се сработат во процесот на наоѓање на решението. Некои алгоритми во одредени задачи на НП, дури и кога се има во предвид примената на современите сметачи, не се секогаш применливи. Таа примена зависи од повеќе фактори, посебно од карактерот и димензијата на математичкиот модел, заради кој повеќето од задачите на НП во извесна смисла се третираат како истражувачки, а не како рутински задачи.

Примената на НП е изложена кај некои задачи од транспортен тип, како што се: трошоците во однос на количините на производите кои се транспортираат, кај нелинеарните задачи за распределба на еднородни ресурси, кај нелинеарните задачи за распределба на нееднородни ресурси кои се далеку посложени.

Нелинеарното програмирање наоѓа примена и во изборот на асортиманот на производство, во оптимизација на производство, увоз и извоз како потреба од количините на поедини производи со одредени ограничувања.

Се применува и во воени цели како што е на пример распоред на оружје на противничките цели(за добивање на нивни максимални губитоци), за нанесување на одредени штети кај противникот, под услов на минимални трошоци.

Проблемите на оптимизација и оптималните процеси ги наоѓаме во многу области на природните, општествените и техничките науки. Всушност, овие проблеми секогаш биле и се неразделно поврзани со развојот на човештвото.

Литература:

[1] Kuhn H.Tucker A., Non-Linear Programming, Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probality, University of California Press, Berkeley, California, 1950

[2] Slater M. Lagrange Multipliers Revisited, A Contribution to Nonlinear Programming, Cowles Commission Discussion Paper in Mathematics, No. 403, 1950

[3] Dantzing G.B., Linear programming and extensions, Princenton University Press, Princenton, New Jersey, 1963

[4] Dr. Jovan J.Petrić , Operaciona istraźivanja, Naučna knjiga, Beograd 1989

[5] Sanjo Zlobec, Jovan Petrić, Nelinearno programiranje, Naučna knjiga, Beograd, 1989

[6] P.Venkataraman, Applied Optimization with Mat lab Programming, John Wiley and Sons,inc. 2002

[7] Е.Гелова, Б.Пиперевски, Е.Аспровска, М.Николовска, За еден варијационен проблем од електротехника и неговиот соодветен контурен проблем.