

EGY PÉNZÜGYI PROBLÉMA VIZSGÁLATA KVADRATIKUS PROGRAMOZÁSSAL

Szakedolgozat

Írta: Szluka Szilvia Irén

Matematika BSc szak, Matematikai elemző szakirány

Témavezető:

Nagy Marianna, egyetemi tanársegéd
Operációkutatási Tanszék

Eötvös Loránd Tudományegyetem, Természettudományi Kar



Eötvös Loránd Tudományegyetem
Természettudományi Kar

2009

Tartalomjegyzék

1. Bevezetés	4
2. A portfólió kiválasztási probléma és modellezése	5
2.1. A portfólió	5
2.2. A portfólió kiválasztási probléma	5
2.3. A probléma modellezése	6
2.3.1. Alapfogalmak	6
2.3.2. A hasznossági függvény	8
2.3.3. A modell	9
2.3.4. Programozási modellek	11
3. Kvadratikus programozás	14
3.1. Az általános nemlineáris optimalizálási feladat	14
3.2. A kvadratikus programozási feladat általános alakja	15
3.3. Optimalitási feltételek	16
3.3.1. A Karush-Kuhn-Tucker tétel	18
3.4. A lineáris komplementaritási feladat	19
3.4.1. A feladat ismertetése	19
3.4.2. A kvadratikus programozási feladat felírása lineáris komplementaritási feladatként	20
3.5. Alkalmazások	21
4. Algoritmusok	23
4.1. A keretalgoritmus	23
4.2. A konjugált gradiens módszer	25
4.3. A kvázi-Newton módszer	27
4.4. A nulltér módszer	28
4.5. Az aktív halmaz módszer	30
4.6. Lemke féle pivotalgorithmus	32

5. A portfólió kiválasztási probléma modelljeinek megvalósítása MATLAB környezetben	36
5.1. A quadprog parancs	37
5.2. A Lemke-féle pivotalgorithmus megvalósítása MATLAB-ban	38
5.2.1. A lemke függvény	38
5.2.2. A lemkequad függvény	39
5.3. A felhasznált adathalmaz	39
5.4. Az első modell megoldása	42
5.5. A második modell megoldása	43
6. Összefoglalás	47
7. Köszönetnyilvánítás	48
A. Függelék	49
A.1. A Lemke féle pivotalgorithmus kódja	49

1. fejezet

Bevezetés

A portfólió kiválasztási probléma, mint a portfólió elmélet fontos fejezete a pénzügyi problémák között jelentős helyet foglal el. A probléma megoldása egy befektetőnek arra a kérdésre adja meg a választ, hogy adott korlátozott tőkéjét a lehető legnagyobb haszon elérése érdekében, mely pénzügyi eszközökbe és milyen arányban fektesse be. A legtöbb befektető ugyan megelégedne azzal, ha valaki elvárásai ismeretében egyszerűen csak megmondaná neki a választ, de dolgozatomban Harry Markowitz nyomain elindulva betekintést nyújtok e meghatározási folyamat lépéseibe, a probléma matematikai modellezése által.

Az első rész témája a probléma részletes ismertetése, a modellek felépítése. A második illetve harmadik részben a felépített kvadratikus modellel a szemünk előtt áttekintjük a kvadratikus programozáshoz kapcsolódó legfontosabb elméleti eredményeket, fogalmakat, és a problémát megoldó fontosabb algoritmusokat. Végül a negyedik részben MATLAB programozási környezetben megvalósítjuk a modelleket, és két módszerrel is megoldva őket az eredményeket grafikusán interpretáljuk és elemezzük.

2. fejezet

A portfólió kiválasztási probléma és modellezése

Ezen fejezet forrása a [16] honlapon található esettanulmány.

2.1. A portfólió

Először tisztáznunk kell, pontosan mit értünk a portfólió fogalma alatt. Különböző pénzügyi eszközök egy halmazára kell gondolnunk, dolgozatomban - és általában a szakirodalomban - azonban az ezek részhalmazát alkotó értékpapírok (részvények, kötvények, különböző hitelek, stb.) összességét tekintjük portfóliónak.

2.2. A portfólió kiválasztási probléma

Harry Markowitz 1952-ben publikált cikke [14] volt az első portfólióelméleti munka. Markowitz a portfólió kiválasztásának folyamatát két fő lépésre bontotta.

Elsőként a rendelkezésre álló pénzügyi eszközök jövőbeli viselkedését (elsődlegesen hozamát és szórását) kell valamiképpen előrejelezni.

Egy egyszerű vagy annuitásos kötvény esetében ez a feladat nem nehéz, hiszen a névleges kamatláb alapján pontosan tudja a befektető, hogy a kamatfizetési periódusok és a lejárat időpontjában mekkora pénzáramlásokra számíthat, és ez mekkora megtérülést jelent az egységnyi befektetett tőkéjére nézve.

Részvények esetében azonban már nincs ilyen biztos tudásunk a jövőbeli hozamokról. Az árfolyamváltozás minden t időpillanatban egy $X(t)$ valószínűségi változóval, az idősorral jellemezhető. 1953-ban Maurice Kendall brit statisztikus megmutatta [6], hogy ezek az ármegmozgások véletlen bolyongást végeznek, azaz a holnapi árfolyamváltozás független a maiktól.

A gyakorlatban elterjedt statisztikai módszer a jövőbeli viselkedés leírására a historikus, azaz múltbeli adatok alapján történő becslés, majd ezen eredmények esetleges utólagos korrigálása.

A második lépésben ezen adatok, és a befektető preferenciáitól függő feltételek alapján történik meg a portfólió kiválasztása.

Az alapvető megválaszolandó kérdés a következő: adott értékpapír halmaz és adott tőke mennyiség mellett a befektetőnek mely értékpapírokba, és milyen arányban kell elhelyeznie a pénzét, hogy adott kockázat mellett a lehetséges legmagasabb várható hozamot; adott elvárt hozam mellett a lehetséges legkisebb kockázatot biztosítsa az így létrehozott portfólióval. A cél az ún. hatékony portfóliók meghatározása (Markowitz az efficiens határ kifejezést használja), melyet a következőképpen definiálunk:

2.2.1. Definíció. Legyen P portfólió, elvárt hozama r , kockázata σ . P hatékony, ha a következők teljesülnek:

- nem létezik P' portfólió: $r' \geq r$, de $\sigma' < \sigma$,
- nem létezik P'' portfólió: $\sigma'' \leq \sigma$, de $r'' > r$.

Ha az értékpapír halmazban minden elemről egyértelműen tudjuk, hogy mekkora hozamot biztosít, akkor a legnagyobb hozamúakból előállított portfóliók lesznek a hatékonyak.

A portfólió várható hozamát úgy kapjuk meg, hogy a beválasztott értékpapírok várható hozamának súlyozott átlagát számítjuk ki.

A portfólió kockázatát a hozamok szórásnégyzetével, *variánciájával* mérjük.

Az értékpapírpiacot vizsgálva beszélhetünk arról, hogy két eszköz hozamai milyen mértékben mozognak együtt, ellentétesen vagy esetleg egymástól függetlenül. Ennek mérőszáma lesz a *korreláció*, illetve a *kovariancia*. A portfóliókiválasztás lényege azon elven alapszik, hogy a kockázat csökkentése érdekében érdemes ellentétesen mozgó eszközöket beválasztani portfóliónkba. (Ezt nevezzük diverzifikációnak.)

A következő részben definiáljuk mindezen fogalmakat.

2.3. A probléma modellezése

2.3.1. Alapfogalmak

Szükségünk lesz néhány algebrából és valószínűségszámításból ismert definícióra.

2.3.1. Definíció. Egy $Q \in \mathbb{R}^{n \times n}$ mátrix szimmetrikus, ha $Q^T = Q$,

pozitív (negatív) definit, ha $x^T Q x > 0$ (< 0) $\forall 0 \neq x \in \mathbb{R}^n$,
 pozitív (negatív) szemidefinit, ha $x^T Q x \geq 0$ (≤ 0) $\forall x \in \mathbb{R}^n$,
 indefinit, ha pozitív és negatív értékeket is felvesz.

Egy szimmetrikus mátrix pontosan akkor pozitív szemidefinit, ha a sajátértékei nemnegatívak, illetve pontosan akkor pozitív definit, ha a sajátértékei pozitívak. Továbbá egy szimmetrikus Q mátrix pontosan akkor pozitív szemidefinit, ha felírható $Q = C^T C$ alakban, ahol $C \in \mathbb{R}^{n \times n}$. Ez a felbontás megvalósítható például Cholesky-faktorizáció segítségével.

2.3.2. Definíció. Legyen X diszkrét valószínűségi változó, $P(X = x_i) = p_i$, $i = 1, \dots, n$.

Ekkor X várható értéke:

$$E(X) = \sum_{i=1}^n p_i x_i,$$

X varianciája:

$$\sigma^2(X) = E(X - E(X))^2 = E(X^2) - (E(X))^2.$$

$\sigma(X)$ az X szórása.

2.3.3. Definíció. Legyenek X és Y diszkrét valószínűségi változók, melyeknek létezik a várható értéke. X és Y kovarianciája:

$$\text{cov}(X, Y) = E[(X - E(X))(Y - E(Y))].$$

Látható, hogy $\text{cov}(X, X) = \sigma^2(X)$, vagyis a valószínűségi változó önmagával vett kovarianciája a varianciája.

Két valószínűségi változó közötti függőség mérőszáma a *korrelációs együttható*.

2.3.4. Definíció. Legyenek X és Y diszkrét valószínűségi változók, melyeknek létezik a várható értéke, szórása. Ekkor az X és Y közötti korrelációs együttható:

$$\rho(X, Y) = \frac{\text{cov}(X, Y)}{\sigma(X)\sigma(Y)},$$

ha $\sigma(X) \neq 0$ és $\sigma(Y) \neq 0$. Ha az utóbbiak közül valamelyik 0, akkor $\rho(X, Y)$ definíció szerint 0.

A korrelációs együttható segítségével a kovarianciát a következő alakban is felírhatjuk:

$$\text{cov}(X, Y) = \rho(X, Y)\sigma(X)\sigma(Y).$$

2.3.5. Definíció. Legyen

$$\mathbf{X} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{pmatrix}$$

ahol X_i , $i = 1, \dots, n$ diszkrét valószínűségi változók. Az \mathbf{X} által generált kovariancia mátrix az a $\mathbf{Q} \in \mathbb{R}^{n \times n}$, melyre

$$Q_{i,j} = E[(X_i - E(X_i))(X_j - E(X_j))].$$

A korrelációmátrixot a fenti definíció alapján hasonlóan definiálhatjuk: az a $\mathbf{C} \in \mathbb{R}^{n \times n}$, melyre

$$C_{i,j} = \frac{\text{cov}(X_i, X_j)}{\sigma(X_i)\sigma(X_j)}.$$

Nyilvánvalóan \mathbf{C} és \mathbf{Q} szimmetrikus mátrixok. Belátható továbbá, hogy \mathbf{C} és \mathbf{Q} pozitív szemidefinit is.

2.3.2. A hasznossági függvény

A közgazdaságtanban a hasznossági függvény, általános célja, hogy a gazdaság egy szereplőjének - vagy bizonyos esetekben a társadalom egészének - meghatározott javakhoz kapcsolódó preferenciáit matematikai eszközökkel modellezze. Az itt felvázolt esetben a gazdasági szereplő a *befektető*. A hasznossági függvény, $U(x)$ a *kockázatra* vonatkozó preferenciáit írja le.

A *kockázatkerülő* befektetőt a konkáv hasznossági függvény jellemzi, vagyona növekedésével csökken a marginális hasznosság. Ha nő a vagyona, nő a hasznossága is, de egyre csökkenő mértékben. A hasznossági függvényre fennállnak a következők: $U'(x) > 0, U''(x) < 0$. Ennek a befektetőnek bizonyos mennyiségű megszerzett pénz után, egy következő egység már alig vagy egyáltalán nem nyújt hasznosságot.

A *kockázatsemleges* befektető hasznossági függvénye lineáris, minden következő egység megszerzett pénz után ugyanakkora hasznosságot realizál, mint az előző egységnél, tehát a marginális haszon állandó. Ezt azt jelenti, hogy $U''(x) = 0$. A kockázatsemleges befektető érzéketlen a kockázatra, azaz két azonos várható hozamú befektetésnél a választása közömbös, független azok kockázatától.

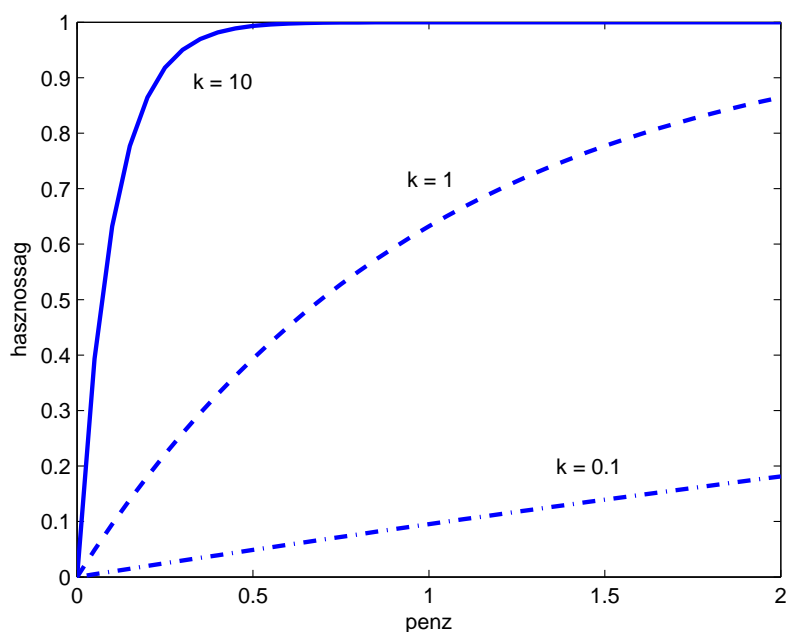
A *kockázatkedvelő* befektetőre a konvex hasznossági függvény és a növekvő marginális hasznosság jellemző. $U'(x) > 0, U''(x) > 0$.

A portfólió kiválasztási probléma modelljében a következő függvényt használjuk majd a befektető magatartásának leírására:

$$U(x) = 1 - e^{-kx},$$

ahol $k > 0$ a kockázatkerülési konstans, x a pénz (vagy vagyon) mennyisége. Minél közelebb van a nullához k értéke, a befektető annál inkább kedveli a kockázatot. A befektetők általában kockázatkerülők.

Az alábbi ábrán $U(x)$ látható különböző k értékekre.



Léteznek más alakú hasznossági függvények is a kockázattal szembeni magatartás leírására, két példa:

$$U(x) = x^a, U(x) = \log(x).$$

A feladat minden esetben a befektető hasznosságának, esetünkben várható hasznosságának maximalizálása.

2.3.3. A modell

Legyen $S = \{s_1, s_2, \dots, s_N\}$ az N darab értékpapírt tartalmazó halmaz, P az ebből képzett portfóliónk. Legyenek

$$\mathbf{R} = \begin{pmatrix} R_1 \\ R_2 \\ \vdots \\ R_N \end{pmatrix}, \quad \mathbf{w} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{pmatrix},$$

ahol R_i diszkrét valószínűségi változó az i . értékpapír hozama, $w_i \in \mathbb{R}$ az i . értékpapír súlya a portfólióban. Ekkor nyilvánvalóan:

$$\sum_{i=1}^N w_i = 1.$$

Feltesszük, hogy minden i indexre $w_i \geq 0$, azaz nem engedjük meg a rövidre eladást, az ún. shortolást.

Legyen

$$\mathbf{r} = E(\mathbf{R}) = (E(R_1), E(R_2), \dots, E(R_N))^T.$$

Ahogy azt korábban már láttuk, a *portfólió várható hozama* az őt alkotó értékpapírok elvárt hozamának súlyozott átlaga:

$$\mathbf{r}_P = E(R_P) = \sum_{i=1}^N E(R_i)w_i = \mathbf{r}^T \mathbf{w}.$$

Legyenek $\sigma_1, \sigma_2, \dots, \sigma_N$ rendre az egyes értékpapírok hozamának szórása, azaz az értékpapír kockázata:

$$\sigma_i = E(R_i - E(R_i))^2, \quad i = 1, \dots, N$$

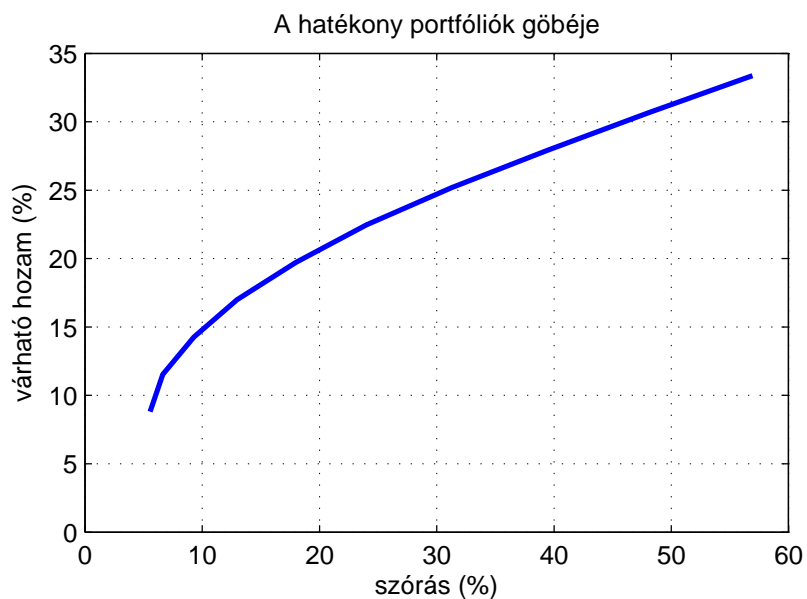
Jelölje ρ_{ij} az s_i és s_j értékpapírok hozamának korrelációját. Legyen $\mathbf{Q} \in \mathbb{R}^{N \times N}$ az értékpapírok hozamainak kovarianciamátrixa, azaz $Q_{i,j} = \text{cov}(R_i, R_j)$.

Ekkor a *portfólió kockázata*:

$$\sigma_P^2 = \sum_{i=1}^N \sum_{j=1}^N w_i w_j \rho_{ij} \sigma_i \sigma_j = \sum_{i=1}^N \sum_{j=1}^N w_i w_j Q_{i,j} = \mathbf{w}^T \mathbf{Q} \mathbf{w}.$$

Az egyes portfóliókat két paraméterük, szórásuk és várható hozamuk szerint a síkon ábrázolhatjuk. Jelölje a vízszintes tengely a kockázatot (σ), míg a függőleges tengely a várható hozamot (\mathbf{r}).

A hatékony portfóliók halmaza ekkor az alábbi ábrán látható.



A hatékony portfóliókénál bővebb halmazt alkotnak a *határportfóliók*, melyek az összes lehetséges portfólió burkolóját jelentik. Ezen portfóliókra rögzített elvárt hozam mellett minimális kockázat jellemző.

2.3.4. Programozási modellek

A befektető által elvárt minimális hozam legyen \mathbf{r}^* , az elfogadott maximális kockázat pedig $(\sigma^*)^2$.

Az alábbiakban először a következő két modellt tekintjük.

1. modell

A kockázat minimalizálása az elvárt minimális hozamra vonatkozó feltétel mellett.

$$\min \mathbf{w}^T \mathbf{Q} \mathbf{w}$$

$$\mathbf{r}^T \mathbf{w} \geq \mathbf{r}^*$$

$$\sum_{i=1}^N w_i = 1$$

$$\mathbf{w} \geq 0$$

2. modell

A várható hozam maximalizálása a maximális kockázatra vonatkozó korlát mellett.

$$\begin{aligned} \max \mathbf{r}^T \mathbf{w} \\ \mathbf{w}^T \mathbf{Q} \mathbf{w} &\leq (\sigma^*)^2 \\ \sum_{i=1}^N w_i &= 1 \\ \mathbf{w} &\geq 0 \end{aligned}$$

Korábban láttuk, hogy a \mathbf{Q} kovariancia mátrix mindig pozitív szemidefinit. Abban az esetben, ha \mathbf{Q} nem pozitív definit (tehát a szigorú egyenlőtlenség nem áll fenn) a fenti programozási feladatok megoldása nem feltétlenül eleme a hatékony portfóliók halmazának. Az első modell esetében például előfordulhat, hogy létezik egy azonos kockázatú, de magasabb várható hozamú portfólió. Ekkor a modell által adott portfólió nyilván nem hatékony.

A két modell egyesítése

Legyen a befektető hasznossági függvénye $U(x) = 1 - e^{-kx}$. Továbbra is N darab értékpapír áll rendelkezésünkre a kívánt portfólió összeállításához, melyeket a fentebb említett S halmaz reprezentál. Tegyük fel, hogy \mathbf{R} , azaz a hozamvektor N dimenziós normális eloszlású valószínűségi változó, melynek várható értéke \mathbf{r} , szórása a \mathbf{Q} kovariancia mátrix. Ekkor $\mathbf{Z} = \mathbf{R}^T \mathbf{w}$ is N dimenziós normális eloszlású, $\mathbf{r}^T \mathbf{w}$ várható értékkel és $\sigma^2 = \mathbf{w}^T \mathbf{Q} \mathbf{w}$ szórással.

A várható hasznosság ekkor a következő számítás útján adódik:

$$\begin{aligned} E(U(x)) &= \int_{-\infty}^{\infty} (1 - e^{-kx}) \phi(x) dx = \\ &= 1 - \frac{1}{\sigma \sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-kx - \frac{1}{2} \left(\frac{x-\mathbf{z}}{\sigma}\right)^2} dx = \\ &= 1 - e^{-k\mathbf{z} + \frac{1}{2} k^2 \sigma^2}. \end{aligned}$$

Mivel az $f(x) = 1 - e^{-kx}$ függvény szigorúan monoton növekedő, ezért $E(U(x))$ akkor és csak akkor maximális, ha $-k\mathbf{z} + \frac{1}{2} k^2 \sigma^2$ minimális, amely ekvivalens $\mathbf{z} - \frac{1}{2} k \sigma^2$ maximalizálásával. Mindezekből már fel tudjuk írni a kvadratikus programozási feladatot. Ha adott a \mathbf{Q} kovariancia mátrix, az \mathbf{r} várható hozam vektor, valamint a befektető k kockázatkerülési konstansa, akkor a várható hasznosságát maximalizáló kvadratikus programozási feladat:

$$\begin{aligned} \max \mathbf{r}^T \mathbf{w} - \frac{k}{2} \mathbf{w}^T \mathbf{Q} \mathbf{w} \\ \sum_{i=1}^N w_i = 1 \\ \mathbf{w} \geq 0 \end{aligned}$$

Összefoglalva a következőket mondhatjuk el az egyesített modellről. A befektető célja portfóliójának összeállításakor várható hasznosságának maximalizálása. Egy portfólió akkor lehet maximális várható hasznosságú egy befektető számára, ha a neki leginkább megfelelő egyensúly érvényesül annak várható hozama és szórása között. Ezt az egyensúlyi állapotot a kockázati preferenciáját leíró k nemnegatív konstans határozza meg. A célfüggvényt tehát felfoghatjuk úgy is, hogy leírja, milyen mértékben hajlandó a befektető várható nagyobb hozam reményében kockáztatni. A hasznossági függvény bevezetésekor pontosan ez volt a célunk. Az ötödik fejezet főként ezen végső modell megvalósítására épül.

3. fejezet

Kvadratikus programozás

Ebben a fejezetben röviden áttekintjük a nemlineáris programozással azon belül elsősorban a kvadratikus feladatosztállyal kapcsolatos legfontosabb tételeket, definíciókat, majd rátérünk az optimalitási feltételek vizsgálatára, kiemelten tárgyalva a Karush-Kuhn-Tucker tételt. A fejezet végén a kvadratikus programozási feladat egy alternatív interpretációjáról lesz szó, amikor a problémát lineáris komplementaritási feladatként írjuk fel. A bizonyításokat nem részletezzük, ezeket megtalálhatjuk a [1, 7, 17] könyvekben, melyek ezen fejezet forrásául szolgáltak.

3.1. Az általános nemlineáris optimalizálási feladat

A feladat általános alakban a következőképpen írható fel:

$$\begin{aligned} \min f(x) \\ h_i(x) = 0, \quad i \in I = \{1, \dots, p\} \\ g_j(x) \leq 0, \quad j \in J = \{1, \dots, m\} \\ x \in \mathcal{C}, \end{aligned} \tag{3.1}$$

ahol $x \in \mathbb{R}^n, \mathcal{C} \subseteq \mathbb{R}^n$ adott halmaz, $f, g_1, \dots, g_m, h_1, \dots, h_p$ függvények értelmezési tartománya pedig valamely $\mathcal{X} \supseteq \mathcal{C}$ nyílt halmaz.

3.1.1. Definíció. Az $\mathcal{F} = \{x \in \mathcal{C} : h_i(x) = 0, g_j(x) \leq 0, \forall i \in I, j \in J\}$ halmazt megengedett halmaznak nevezzük. Az x pont megengedett megoldás, ha $x \in \mathcal{F}$.

A (3.1) feladatot az f függvény illetve az egyenlőség és egyenlőtlenség feltételekben szereplő függvények alakja alapján csoportosíthatjuk. Egy igen fontos osztályt képeznek azok a feladatok, melyekben a célfüggvény konvex.

3.1.2. Definíció. Egy $X \subseteq \mathbb{R}^n$ konvex halmazon értelmezett $f : X \rightarrow \mathbb{R}$ függvény (szigorúan) konvex, ha $\forall x_1, x_2 \in X$ és $0 \leq \lambda \leq 1$ -re:

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2).$$

Konvex nemlineáris optimalizálási feladatról tehát akkor beszélünk, ha f konvex.

3.2. A kvadratikus programozási feladat általános alakja

A nemlineáris optimalizálási feladatok speciális osztályát képezik a kvadratikus programozási feladatok. Ebben az esetben $f(x)$ kvadratikus, azaz másodfokú függvény, az egyenlőségekkel és egyenlőtlenségekkel adott feltételekben pedig a h_i és g_j függvények lineárisak vagy más szóval affin függvények. A feladat a következőképpen írható fel:

$$\min \frac{1}{2} x^T Q x + c^T x$$

$$a_i^T x = b_i, \quad i \in I = \{1, \dots, p\}$$

$$a_j^T x \leq b_j, \quad j \in J = \{p + 1, \dots, p + m\},$$

(3.2)

ahol $Q \in \mathbb{R}^{n \times n}$ szimmetrikus mátrix, $c \in \mathbb{R}^n$ vektor, I és J pedig az egyenlőtlenség és egyenlőség feltételeknek megfelelő indexhalmazok.

Konvex kvadratikus programozási feladatról akkor beszélünk, ha a célfüggvény konvex. Belátható, hogy ez akkor és csak akkor teljesül, ha Q pozitív szemidefinit.

Minden további esetben, azaz ha Q negatív (szemi)definit vagy indefinit a feladat **nemkonvex**. Ilyenkor nem létezik olyan algoritmus, mely polinomidőben globális minimumot találna, azaz a probléma *NP*-nehéz. A célfüggvény ugyanis több lokális minimumhellyel is rendelkezhet, a globális minimum megtalálása igen nehéz feladat. Tekintsünk egy szélsőséges példát. Legyen $Q = -I$, ahol I az egységmátrix és legyen a minimalizálási feladat az alábbi:

$$\min -x^T x$$

$$-1 \leq x_i \leq 1, \quad i = 1, \dots, n$$

A Q mátrix negatív definit, hiszen -1 n -szeres sajátértéke. Látható, hogy a célfüggvénynek minimuma van az $|x_i| = 1$, $i = 1, \dots, n$ alakú pontokban, azaz összesen 2^n lokális minimumhelyet kellene megvizsgálnunk.

3.3. Optimalitási feltételek

Az optimális megoldás létezéséhez, azaz az $f(x)$ függvény minimumának létezéséhez teljesülniük kell bizonyos *optimalitási feltételek*nek. A következőekben ezeket tekintjük át röviden, speciálisan a kvadratikus optimalizálási feladatra nézve. Néhány definícióra szükségünk lesz:

3.3.1. Definíció. Az $f : \mathbb{R}^n \rightarrow \mathbb{R}$ differenciálható függvény gradiensvektora, $\nabla f(x)$ az az n dimenziós vektor, melyre

$$(\nabla f(x))_i = \frac{\partial f(x)}{\partial x_i}.$$

3.3.2. Definíció. Az $f : \mathbb{R}^n \rightarrow \mathbb{R}$ differenciálható függvény $x \in \mathbb{R}^n$ pontbeli $s \in \mathbb{R}^n$ irányú iránymenti deriváltja a következő:

$$\delta f(x, d) = \lim_{\lambda \rightarrow 0} \frac{f(x + \lambda d) - f(x)}{\lambda},$$

ha a határtérték létezik.

Ha az f függvény folytonosan differenciálható, akkor minden $d \in \mathbb{R}^n$ esetén

$$\delta f(x, d) = \nabla f(x)^T d.$$

3.3.3. Definíció. Az $f : \mathbb{R}^n \rightarrow \mathbb{R}$ differenciálható függvény Hesse-mátrixa az a $\nabla^2 f(x) \in \mathbb{R}^{n \times n}$ mátrix, melyre

$$(\nabla^2 f(x))_{i,j} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j},$$

azaz $\nabla^2 f(x)$ f másodrendű parciális deriváltjait tartalmazó mátrix.

A feltétel nélküli kvadratikus programozási feladat

Tekintsük a következő problémát:

$$\min q(x) = \frac{1}{2} x^T Q x + c^T x, \quad (3.3)$$

ahol $Q \in \mathbb{R}^{n \times n}$ szimmetrikus mátrix, $c \in \mathbb{R}^n$ vektor. A $q(x)$ függvényre a következő megfigyeléseket tehetjük:

a gradiensvektor:

$$\nabla q(x) = Qx + c,$$

a Hesse-mátrix:

$$\nabla^2 q(x) = Q,$$

q kétszer folytonosan differenciálható, ezért az iránymenti derivált:

$$\delta q(x, s) = (Qx + c)^T s.$$

Analízisből ismert a többváltozós függvény (szigorú) lokális illetve globális minimumának/maximumának definíciója. Nyilván teljesül, hogy a globális optimum egyben lokális optimum is. A (3.3) problémában, ha Q pozitív szemidefinit mátrix, azaz a q függvény konvex, akkor többet is állíthatunk: ha $b \in \mathbb{R}^n$ pont lokális minimum, akkor egyben globális minimum is.

Az optimalitás *szükséges feltételei* a (3.3) feladatra:

elsőrendű feltétel: ha a $b \in \mathbb{R}^n$ pont q lokális minimuma, akkor $\nabla q(b) = (Qb + c) = 0$, másodrendű feltétel: ha a $b \in \mathbb{R}^n$ pont q lokális minimuma, akkor $\nabla q(b) = (Qb + c) = 0$ és Q pozitív szemidefinit.

Az optimalitás *elégséges feltétele* a (3.3) feladatra:

ha a $b \in \mathbb{R}^n$ pontra teljesül, hogy $Qb + c = 0$ és Q pozitív definit, akkor a b szigorú lokális minimuma q -nak.

Az egyenlőtlenség feltételekkel adott kvadratikus programozási feladat

Tekintsük a következő problémát:

$$\begin{aligned} \min q(x) \\ a_i^T x \leq b_i, \quad i \in I = \{1, \dots, p\} \\ x \in \mathcal{C} \end{aligned}$$

ahol $\mathcal{C} \subseteq \mathbb{R}^n$, Q szimmetrikus mátrix. A megengedett megoldások halmaza a korábban bevezetett jelölésnek megfelelően legyen \mathcal{F} .

Definiálni szeretnénk, hogy egy adott x megengedett pontban, melyek azok a vektorok, amelyek irányában létezik megengedett megoldás, azaz ezen vektor irányába elindulva létezik egy határ, ameddig biztosan nem sértünk meg egyetlen egyenlőtlenséget sem.

3.3.4. Definíció. *A $d \in \mathbb{R}^n$ vektort az $x \in \mathcal{F}$ ponthoz tartozó megengedett iránynak hívjuk, ha létezik olyan $\lambda_0 > 0$, amelyre minden $0 \leq \lambda \leq \lambda_0$ esetén $x + \lambda d \in \mathcal{F}$. Az $x \in \mathcal{F}$ megengedett pontban a megengedett irányok halmazát jelölje $\mathcal{FD}(x)$, tehát*

$$\mathcal{FD} = \{d : 0 \neq d \in \mathbb{R}^n, x + \lambda d \in \mathcal{F}\}.$$

3.3.1. Állítás. *Tegyük fel, hogy Q pozitív szemidefinit, és tekintsük a (3.3) problémát. Az $x^* \in \mathcal{F}$ megengedett megoldás akkor és csak akkor optimális megoldása a feladatnak, ha minden $d \in \mathcal{FD}(x^*)$ esetén $(Qx^* + c)^T d \geq 0$.*

Másképpen fogalmazva az x^* pont akkor és csak akkor optimális megoldása (3.3) feladatnak, ha onnan indulva bármely megengedett irányban az iránymenti derivált nemnegatív, azaz a célfüggvény értéke bármely irányban legalább akkora, mint az x^* pontban.

3.3.1. A Karush-Kuhn-Tucker tétel

Karush valamint Kuhn és Tucker, 1939-ben és 1951-ben egymástól függetlenül bizonyították az alább szereplő tételt (ld. [5, 11]), amely a többváltozós függvények egyenlőség feltételek mellett történő optimalizálására (minimalizálására illetve maximalizálására) kifejlesztett Lagrange-féle multiplikátor módszer általánosításaként fogható fel. A tételt kimondjuk általános nemlineáris optimalizálási feladatra, majd megvizsgáljuk mit jelent ez a (3.2) problémára.

3.3.1. Tétel. *Legyen x^* a (3.1) feladat megengedett megoldása, és legyen $A = \{j : g_j(x^*) = 0\}$. Tegyük fel, hogy $f, h_i \in I = \{1, \dots, p\}$ és $g_j, j \in J$ folytonosan differenciálhatóak az x^* pontban. Továbbá tegyük fel, hogy $\nabla g_j(x^*)_{j \in A}$ és $\nabla h_i(x^*)_{i \in I}$ lineárisan függetlenek. Ha x^* a (3.1) probléma lokális minimuma, akkor léteznek $\lambda^*_i \ i \in I$ és $\mu^*_j \ j \in J$ Lagrange-szorozók, hogy*

$$\nabla f(x^*) + \sum_{j=1}^m \mu^*_j \nabla g_j(x^*) + \sum_{i=1}^p \lambda^*_i \nabla h_i(x^*) = 0 \quad (3.4)$$

$$\mu^*_j g_j(x) = 0 \quad j \in J \quad (3.5)$$

$$\mu^*_j \geq 0 \quad j \in J \quad (3.6)$$

$$(3.7)$$

A (3.7) szükséges feltételekre *KKT feltételekként* szokás hivatkozni.

A tétel a kvadratikus programozási feladatra:

Ha az $x^* \in \mathbb{R}^n$ megengedett megoldás lokális minimuma a (3.2) feladatnak, akkor léteznek $\lambda^* \in \mathbb{R}^p$ és $\mu^* \in \mathbb{R}^m$ vektorok, hogy:

$$Qx^* + c + A_1^T \lambda^* + A_2^T \mu^* = 0 \quad (3.8)$$

$$A_2 x^* \mu^* = 0 \quad (3.9)$$

$$\mu^* \geq 0 \quad (3.10)$$

ahol $A_1 = (a_1, \dots, a_p)^T \in \mathbb{R}^{p \times n}$ és $A_2 = (a_{p+1}, \dots, a_{p+m})^T \in \mathbb{R}^{m \times n}$.

Karush Kuhn és Tucker elégséges feltételeket is adtak a (3.1) feladat x^* optimális megoldására, melyek további tulajdonságok teljesülését követelik meg az f, h_i és g_j függvényekkel szemben. (Ezekre itt részletesen nem térünk ki, csak megjegyezzük, hogy a kvázikonvexitás és pszeudokonvexitás fogalmára lenne szükségünk a tétel kimondásához.)

A KKT feltételek az algoritmusok, a számítógépes megvalósítás szempontjából is lényegesek, hiszen megállási kritériumként szolgálnak. A következő fejezetben a Lemke féle pivotalgorithmus, illetve a nulltér módszer esetében használjuk fel a KKT feltételeket a megoldás direkt úton történő kiszámítására.

3.4. A lineáris komplementaritási feladat

Ebben a részben ismertetjük a lineáris komplementaritási feladatot, az ezt megoldó Lemke-féle pivotalgorithmus leírása az Algoritmusok című fejezetben található. A feladat igen fontos szerepet tölt be a mérnöki számítások, a játékelmélet illetve a közgazdasági alkalmazások terén. Azonban amiért itt elsődlegesen vizsgáljuk e feladattípust, az annak köszönhető, hogy szoros kapcsolatban áll a kvadratikus programozási problémával. Az alfejezetben látni fogjuk, hogyan írhatóak fel a kvadratikus programozási feladat KKT feltételei lineáris komplementaritási feladatként. A problémáról további áttekintést olvashatunk az [1, 2, 3, 4, 7, 10] könyvekben.

3.4.1. A feladat ismertetése

Legyen $M \in \mathbb{R}^{p \times p}$ és $q \in \mathbb{R}^p$. A lineáris komplementaritási feladat:

$$w - Mz = q \quad (3.11)$$

$$w_j \geq 0, \quad z_j \geq 0, \quad j = 1, \dots, p \quad (3.12)$$

$$w_j z_j = 0, \quad j = 1, \dots, p \quad (3.13)$$

ahol w, z ismeretlen p dimenziós vektorok. A cél ezek meghatározása, vagy annak megmutatása, hogy nem létezik megoldása a feladatnak. Látható, hogy a feltételeket teljesítő minden (w_j, z_j) pár esetén valamelyik változó nulla, ezeket *komplementáris változóknak* nevezzük, a feladat egy (w, z) megoldása *komplementáris megengedett megoldás*. Továbbá (w, z) *komplementáris megengedett bázismegoldás*, ha (w, z) megengedett bázismegoldása a (3.11)-(3.12) feladatnak, méghozzá úgy, hogy minden (w_j, z_j) párra pontosan az egyik változó bázisváltozó. Megmutatható, hogy a (w, z) pár pontosan akkor komplementáris megengedett megoldása a (3.11)-(3.13) feladatnak, ha a következő speciális *bináris bilineáris programozási feladat* optimális megoldásának része.

$$\begin{aligned} \min \sum_{j=1}^p y_j w_j + (1 - y_j) z_j \\ w - Mz = q \\ w \geq 0, \quad z \geq 0, \quad y \in \{0, 1\}^p. \end{aligned}$$

Mivel w és z nemnegatívak, ezért az optimális célfüggvényérték legalább nulla és $y_j w_j = (1 - y_j) z_j = 0$ teljesül minden j indexre, amiből következik, hogy $w_j z_j = 0$ minden j indexre. Vagyis a fenti feladat minden optimális megoldása megoldása a lineáris komplementaritási feladatnak is. Ez a feladat megoldható bilineáris, konkáv és egészértékű lineáris programozási módszerekkel.

3.4.2. A kvadratikus programozási feladat felírása lineáris komplementaritási feladatként

Tekintsük a kvadratikus programozási problémát a következő alakban:

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Qx + c^T x \\ & Ax \leq b \\ & x \geq 0, \end{aligned} \tag{3.14}$$

ahol $Q \in \mathbb{R}^{n \times n}$ szimmetrikus mátrix, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ és $b \in \mathbb{R}^m$. Ha x nemnegativitását megköveteljük, akkor a (3.2) feladatban az $a_i^T x = b_i$ egyenlőség feltétel helyett az $a_i^T x \leq b_i$ és a $-a_i^T x \leq -b_i$ $i = 1, \dots, p$ egyenlőtlenségeket véve, a feladat felírható a (3.14) problémával azonos formában. Ekkor a (3.14) feladat egy x lokális minimumára a KKT feltételek miatt igaz, hogy léteznek $\lambda \in \mathbb{R}^m$ és $\mu \in \mathbb{R}^n$ Lagrange-szorozók, hogy

$$\begin{aligned} Qx + c + A^T \lambda - \mu &= 0 \\ Ax - b &\leq 0 \\ \lambda^T (Ax - b) &= 0 \\ \mu^T x &= 0 \\ x, \lambda, \mu &\geq 0. \end{aligned} \tag{3.15}$$

Bevezetve a nemnegatív $y \in \mathbb{R}^m$ slack változót az alábbi rendszert kapjuk:

$$\begin{aligned} Qx + c + A^T \lambda - \mu &= 0 \\ Ax - b + y &= 0 \\ \lambda^T y &= 0 \\ \mu^T x &= 0 \\ x, y, \lambda, \mu &\geq 0. \end{aligned} \tag{3.16}$$

Egyszerűbb alakban felírva

$$\begin{pmatrix} y \\ \mu \end{pmatrix} - \begin{pmatrix} 0 & -A \\ A^T & Q \end{pmatrix} \cdot \begin{pmatrix} \lambda \\ x \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix}$$

$$\mu^T x = 0, \lambda^T y = 0$$

$$x, y, \lambda, \mu \geq 0,$$

és a következő jelöléseket bevezetve

$$M = \begin{pmatrix} 0 & -A \\ A^T & Q \end{pmatrix} \quad q = \begin{pmatrix} b \\ c \end{pmatrix} \quad w = \begin{pmatrix} y \\ \mu \end{pmatrix} \quad z = \begin{pmatrix} \lambda \\ x \end{pmatrix}$$

a (3.15) rendszer láthatóan megegyezik a (3.11)-(3.13) lineáris komplementaritási feladattal. Azaz a kvadratikus programozási feladat ekvivalens módon felírható lineáris komplementaritási feladatként.

3.5. Alkalmazások

Dolgozatom tárgya a kvadratikus optimalizálási feladat portfólióelméleti alkalmazásának bemutatása, de itt említést teszek egyéb alkalmazási lehetőségekről is, melyek főként a statisztika, a közgazdaságtan és a mérnöki számítások területéhez kötődnek.

Közülük talán legismertebb és leggyakrabban előforduló alkalmazási terület a statisztikai adatfeldolgozásnál használt **legkisebb négyzetek módszere**. Adottak az $x_1 < x_2 < \dots < x_n \in \mathbb{R}$ pontokhoz tartozó $y_1, y_2, \dots, y_n \in \mathbb{R}$ megfigyelések. Szeretnénk egy görbét illeszteni az y_i pontokra, méghozzá oly módon, hogy a valós és a becült adatok közti eltérés minimális legyen, ami számszerűen azt jelenti, hogy az eltérések négyzetösszege legyen a lehető legkisebb. Keressük tehát azt a (konvex) $y(x)$ függvényt, melyre teljesül, hogy $y_i = y(x_i) + \epsilon_i$, $i = 1, \dots, n$, ahol ϵ_i $i = 1, \dots, n$ független, normális eloszlású, nulla várható értékű valószínűségi változók. A **konvex regressziós feladat**:

$$\min_{y(x) \text{ konvex}} \sum_{i=1}^n (y_i - y(x_i))^2.$$

Mivel csak a megfigyelt y_i értékeket vizsgáljuk a feladatban, feltehetjük, hogy az (x_i, x_{i+1}) , $i = 1, 2, \dots$ intervallumokon $y(x)$ lineáris. A feladat, ekkor a következő kvadratikus programozási probléma:

$$\min \sum_{i=1}^n (y_i - y(x_i))^2$$

$$\frac{y_i - y_{i-1}}{x_i - x_{i-1}} \leq \frac{y_{i+1} - y_i}{x_{i+1} - x_i}, \quad i = 2, \dots, n-1.$$

Egy másik lehetséges alkalmazás az ún. **betonkeverési feladat**. A cél az ideális sóderösszetételű betonhoz nagyon hasonló keverék előállításának minimális cementfelhasználás mellett. Adott n db kavicsméret-kategória. Adott továbbá a $\mathbf{c} = (c_1, c_2, \dots, c_n)^T$ vektor, mely az ideális keveréket reprezentálja, azaz $0 < c_i < 1$ megmutatja, hogy

az i . kategóriájú kavics mekkora részét teszi ki a keveréknek. Mindebből következik, hogy $\sum_{i=1}^n c_i = 1$. Adott m db különböző bányá, ahonnan a sóder származik, az egyes bányákra jellemző sóderösszetételt pedig az $\mathbf{A}_j = (a_{1j}, \dots, a_{mj})^T$ vektor írja le. Az \mathbf{A}_j vektorok elemeire teljesül, hogy $0 \leq a_{ij} \leq 1$ valamint $\sum_{i=1}^n a_{ij} = 1$. Legyen $\mathbf{x} = (x_1, \dots, x_m)^T$ az a vektor, melyre x_i az összes mennyiségnek az i . bányából felhasznált hányadát jelenti. Ekkor teljesül, hogy $0 \leq x_i$ és $\sum_{i=1}^m x_i = 1$. $\mathbf{A} = (\mathbf{A}_1, \dots, \mathbf{A}_m)^T$ jelöléssel a feladatot ekkor a következőképpen írhatjuk fel:

$$\min(\mathbf{Ax} - \mathbf{c})^T(\mathbf{Ax} - \mathbf{c})$$

$$e^T \mathbf{x} = 1$$

$$\mathbf{x} \geq 0,$$

ahol e a csupa egyes vektort jelöli.

4. fejezet

Algoritmusok

Elsőként a feltétel nélküli optimalizálás algoritmusait vizsgáljuk meg, kezdve az alapot jelentő keretalgortmustól. A gradiens módszer és a Newton módszer továbbfejlesztett változait írjuk le részletesebben. A feltételes kvadratikus programozási feladat megoldására mutatjuk be a nulltér, valamint az aktív halmaz módszert, mely lényegi eleme a fejezetnek, hiszen a MATLAB beépített `quadprog` parancsa is ezzel az algoritmussal dolgozik. A fejezet végén a lineáris komplementaritási feladatot megoldó Lemke-féle pivotálgoritmusról lesz szó. A fejezet eredményei a [1, 7, 10] könyvekből illetve a [16] honlapról származnak. További eljárások találhatóak az [1] munkában.

4.1. A keretalgoritmus

A nemlineáris optimalizálási problémát megoldó algoritmusok mindegyike az alábbiakban ismertetett vázra épül.

A feltétel nélküli optimalizálási feladat általános *keretalgortmusa*:

Bemenet: $\varepsilon > 0$ pontossági paraméter és egy x^0 megengedett megoldás.

0. $k = 0$

A következő lépéseket ismételjük:

1. Az x^k pontból indulva egy s^k **keresési irányt** kell találnunk, amelyre $\delta f(x^k, s^k) < 0$, azaz a célfüggvény az s^k irányban csökken.

Ha találtunk megfelelő s^k irányt, akkor az **egyenes menti keresés** következik.

2. Legyen $\alpha^k = \arg \min_{\alpha} f(x^k + \alpha s^k)$.

3. Legyen $x^{k+1} = x^k + \alpha^k s^k$, $k = k + 1$.

4. Ha az x^k pont teljesíti a **megállási feltételt**, akkor az algoritmus leáll. Egyébként $k = k + 1$.

A kiemelt lépések az algoritmus legfőbb lépései, ezek különböző megvalósításai különböztetik meg egymástól a programkódokat.

A **keresési irány** meghatározásának egyik módja a **gradiens-módszer**. Ez az eljárás a legmeredekebb csökkenési iránynak is nevezett negatív gradienst, azaz $-\nabla f(x^k)$ -t választja keresési iránynak. Ez a választás azon alapul, hogy a lenormált iránymenti deriváltat a negatív gradiens minimalizálja. A gradiens-módszer hátrányai közé tartozik, hogy konvergenciája csak lineáris, ezért sok esetben nem túl hatékony. Kvadratikus függvényekre nem alkalmazható, már a konvex esetben sem garantálható ugyanis a minimalizálási folyamat végessége. A konvergencia gyorsítására fejlesztették ki a **konjugált gradiens-módszert**, melyről a következő részben lesz szó.

A másik eljárás a **Newton-módszer**, mely a célfüggvény másodrendű Taylor polinomját minimalizálja. Tegyük fel, hogy f kétszer folytonosan differenciálható, szigorúan konvex függvény, és tekintsük f másodrendű Taylor-polinomját:

$$T_2(x) := f(x^k) + \nabla f(x^k)^T(x - x^k) + \frac{1}{2}(x - x^k)^T \nabla^2 f(x^k)(x - x^k).$$

Belátható, hogy mivel f szigorúan konvex függvény, így $\nabla^2 f(x^k)$ pozitív definit, amiből következik, hogy T_2 is szigorúan konvex függvény. $T_2(x)$ ott veszi fel a minimumát, ahol gradiense a nullvektor:

$$\nabla T_2(x) = \nabla f(x^k) + \nabla^2 f(x^k)(x - x^k) = 0,$$

azaz az

$$x^{k+1} = x^k - (\nabla^2 f(x^k))^{-1} \nabla f(x^k)$$

pontban. A Hesse-mátrix kiszámítása valamint invertálása sokszor igen költségigényes lehet, gyakran több műveletet igényel, mint a gradiens meghatározása. Az ún. **kvázi-Newton módszert** ezen anomáliák kiküszöbölésére fejlesztették ki. Erről részletesen a fejezet későbbi részében lesz szó. A másodfokú polinomfüggvény Taylor-polinomja önmaga, ezért látható, hogy a feltétel nélküli kvadratikus programozási feladat esetében egy lépésben lezajlik a minimalizálás.

Az **egyenes mentén történő keresés** tulajdonképpen egy egydimenziós optimalizálási feladat, melyben a $\Phi(\alpha) = f(x^k + \alpha s^k)$ függvény minimumát kell megtalálnunk. Ez történhet például intervallumfelezéssel vagy a fent ismertetett Newton-módszerrel.

A **megállási feltétel** lehet például annak ellenőrzése, hogy valamilyen eltérés (távolság), amely az optimális megoldás esetében nulla az ε pontossági paraméter értékénél kisebb-e. Megszabhatunk egy felső korlátot az iterációk számára vonatkozóan, illetve ellenőrizhetjük a Karush-Kuhn-Tucker feltételek teljesülését is az aktuális (az iteráció által meghatározott) pontban.

4.2. A konjugált gradiens módszer

A előző részben említést tettünk róla, hogy a gradiens módszer nem véges konvex kvadratikus függvények esetében. Az eljárás, melyet a következőkben fogunk ismertetni igen hatékonyan kezeli ezt a problémát, ugyanis legfeljebb n lépésben megtalálja az optimumot, ahol n a feladat dimenziója, azaz a Hesse mátrix mérete. Elsőként bevezetjük a *konjugált* fogalmát.

4.2.1. Definíció. Legyen $A \in \mathbb{R}^{n \times n}$ szimmetrikus, pozitív definit mátrix, továbbá $x, y \in \mathbb{R}^n$ vektorok. Azt mondjuk, hogy x A -konjugált y -hoz, ha

$$x^T A y = 0.$$

A konjugált tulajdonság szimmetrikus reláció, azaz ha x A -konjugált y -hoz, akkor y A -konjugált x -hez.

4.2.2. Definíció. Legyen $A \in \mathbb{R}^{n \times n}$ szimmetrikus, pozitív definit mátrix. A $d_1, d_2, \dots, d_n \in \mathbb{R}^n$ vektorok A -konjugáltak, ha lineárisan függetlenek és $d_i^T A d_j = 0$ minden $i \neq j \in \{1, 2, \dots, n\}$.

A konjugált rendszer vektorai lineárisan függetlenek, így az n dimenziós tér egy bázisát alkotják. Legyen

$Q \in \mathbb{R}^{n \times n}$ szimmetrikus pozitív definit mátrix, $c \in \mathbb{R}^n$ vektor és tekintsük a következő problémát:

$$\min \frac{1}{2} x^T Q x + c^T x. \quad (4.1)$$

Jelöljük x^* -gal (4.1) optimális megoldását.

A konjugált gradiens módszer direkt változata

Az optimalitás elsőrendű feltételét felírva a feladatra az x^* pontra fennáll, hogy $Qx^* + c = 0$. Mivel Q pozitív definit, ezért a feladat a

$$Qx^* = -c \quad (4.2)$$

lineáris egyenletrendszer megoldásának problémájára vezethető vissza. Legyen d_1, d_2, \dots, d_n egy Q -konjugált rendszer (ld. [19]), továbbá legyenek $\alpha_1, \alpha_2, \dots, \alpha_n$ az x^* optimum koordinátái ebben a bázisban, tehát:

$$x^* = \sum_{i=1}^n \alpha_i d_i.$$

Ekkor

$$c = -Qx^* = -\sum_{i=1}^n \alpha_i Q d_i,$$

$$d_k^T c = -d_k^T Q x^* = -\sum_{i=1}^n \alpha_i d_k^T Q d_i,$$

amiből felírva az α_k koordinátát:

$$\alpha_k = -\frac{d_k^T c}{d_k^T Q d_k}.$$

Az x^* megoldást az α_k együtthatókból nyerjük, tehát összesen n iterációs lépés megtétele szükséges.

A konjugált gradiens módszer iteratív változata

Előfordulhat, hogy n nagy, így a direkt eljárás túl sok számítást igényel. Ekkor alkalmazhatjuk az iteratív eljárást, mely abban különbözik az előzőtől, hogy a konjugált vektort (irányt) mindig az adott iterációban határozzuk meg, így ügyesen megválasztva n -nél kevesebb lépés is elég a minimum eléréséhez. Legyen x^0 a kiinduló vektor. Az a célunk, hogy minden iterációs lépéssel egyre közelebb kerüljünk az optimumhoz. A "közelebb" itt azt jelenti, hogy ha x^0, x^1, \dots az egyes iterációkban a vizsgált vektorok, akkor valamely normára az $\|x^i - x^*\|$, $i = 0, 1, \dots$ számok monoton csökkenő sorozatot alkotnak. Ez a norma lehet például a hagyományos euklideszi norma.

Vezessük be a *reziduális* fogalmát. A k . iterációban:

$$r_k = -c - Qx^k.$$

Tehát a k . iterációban a reziduális értéke a célfüggvény negatív gradiense az x^k helyen. Minden iterációs lépésben találunk kell egy megfelelő d_k keresési irányt, ennek megválasztásánál használjuk ki majd a konjugáltságot. Legyen

$$d_0 = Qx^0 + c,$$

tehát a célfüggvény gradiense az x^0 helyen, a k . iterációban pedig:

$$d_{k+1} = r_k - \sum_{i \leq k} \frac{d_i^T Q r_k}{d_i^T Q d_i} d_i.$$

Egyszerű számolással belátható, hogy d_k és d_{k+1} Q -konjugáltak.

Megadhatunk egy $\varepsilon > 0$ pontossági paramétert és minden iterációban ellenőrizhetjük, hogy elég közel vagyunk-e a megoldáshoz. Ha valamely k -ra

$$\|r_k\| = \|-c - Qx^k\| < \varepsilon,$$

akkor az algoritmus leáll, a kimenet az x^* optimumot ε pontossággal közelítő x^k pont.

KONJUGÁLT GRADIENS MÓDSZER

Bemenet: $\epsilon > 0$ pontossági paraméter és egy x^0 induló megoldás.

0.

$$r_0 = -c - Qx^0$$

$$d_0 = -r_0$$

$$k = 0$$

A következő lépéseket ismételjük:

1. $\alpha_k = \frac{r_k^T r_k}{d_k^T Q d_k}$

2. $x^{k+1} = x^k + \alpha_k d_k$

3. $r_{k+1} = r_k - \alpha_k Q d_k$

4. Ha $|r_{k+1}| < \epsilon$, akkor leállunk.

5. $\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$

6. $d_{k+1} = r_{k+1} + \beta_k d_k$

7. $k = k + 1$

A megoldás az x_{k+1} vektor.

4.3. A kvázi-Newton módszer

A kvázi-Newton módszer lényege, hogy nem számítjuk ki közvetlenül a Hesse-mátrixot és inverzét, hanem különböző formulák segítségével iteratív úton közelítjük azokat. Előnye tehát a Newton-módszerrel szemben a lecsökkent számításigény.

Legyen a feladat

$$\min f(x),$$

továbbá legyen $T_2(x)$ az $f(x)$ függvény 4.1 részben definiált másodfokú Taylor-polinomja. Legyen x^0 egy megengedett induló megoldás. Hasonlóan a Newton-módszerhez az f függvény helyett annak x^k pontbeli másodrendű közelítését vesszük, azaz $\min f(x)$ helyett a

$$\min f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{1}{2} (x - x^k)^T \nabla^2 f(x^k) (x - x^k)$$

problémát szeretnénk megoldani. Vezessük be a következő jelöléseket: $b := f(x^k)$, $c := \nabla f(x^k)$, $H := \nabla^2 f(x^k)$, valamint legyen $\Delta x = x - x^k$. A feladat új formája:

$$\min \frac{1}{2} \Delta x^T H \Delta x + c^T \Delta x + b \tag{4.3}$$

Ekkor (4.3) x^* optimális megoldására igaz, hogy ott a gradiensvektor nullával egyenlő, azaz

$$Hx^* + c = 0,$$

amiből

$$x^* = -H^{-1}c.$$

A kvadratikus programozási feladat esetében természetesen nincs szükség a Hesse-mátrix közelítésére, hiszen az a feladatban adott, ezekkel a formulákkal itt nem foglalkozunk.

A Hesse-mátrix inverzét közelítő formulák

Jelölje A_k H^{-1} k . közelítését. Legyen

$$s_k = x^{k+1} - x^k,$$

$$q_k = \nabla f(x^{k+1}) - \nabla f(x^k).$$

A legelső kvázi-Newton formulát Davidon, amerikai fizikus fejlesztette ki 1959-ben, mely 1963-ban DFP-formula néven vált ismertté (Davidon, Fletcher és Powell után).

A DFP-formula:

$$A_{k+1} = A_k + \frac{s_k s_k^T}{q_k^T s_k} - \frac{A_k^T q_k^T q_k A_k}{q_k^T A_k q_k}.$$

A legtöbbet alkalmazott és leginkább hatékony formula Broyden, Fletcher, Godfarb és Shanno nevéhez fűződik. **A BFGS formula:**

$$A_{k+1} = \left(I - \frac{q_k s_k^T}{q_k^T s_k} \right)^T A_k \left(I - \frac{q_k s_k^T}{q_k^T s_k} \right) + \frac{s_k s_k^T}{q_k^T s_k}.$$

Érdekességképpen megemlítjük még az *SR1* (Symmetric Rank 1) formulát is:

$$A_{k+1} = A_k + \frac{(s_k - A_k q_k)(s_k - A_k q_k)^T}{(s_k - A_k q_k)^T q_k}.$$

A formula onnan kapta a nevét, hogy az iterációkban egy szimmetrikus, egy rangú mátrixszal javítunk.

4.4. A nulltér módszer

Tekintsük az *egyenlőség feltételekkel* adott kvadratikus programozási problémát:

$$\min \frac{1}{2} x^T Q x + c^T x \quad Ax = b, \quad (4.4)$$

ahol $Q \in \mathbb{R}^{n \times n}$, $A \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$.

4.4.1. Definíció. Egy $A \in \mathbb{R}^{m \times n}$ mátrix nulltere azon vektorok halmaza, melyeket A a 0-ba képez, azaz $\text{Null}(A) = \{x \in \mathbb{R}^n : Ax = 0\}$

A nulltér módszer először megkeresi azt a $Z \in \mathbb{R}^{n \times p}$ mátrixot, mely kifeszíti A nullterét, vagyis amelyre $\text{Null}(A) = \{y \in \mathbb{R}^n : y = Zw, w \in \mathbb{R}^p\}$. Z ortogonális felbontási módszerekkel számítható ki, egyszerű struktúrájú A mátrix esetén például (sok 0 eleme van) egy almátrixának LU felbontásából.

Legyen x^0 egy megengedett megoldása a (4.4) feladatnak, ekkor a megengedett megoldások halmaza a következőképpen írható fel:

$$\mathcal{F} = \{x \in \mathbb{R}^n : x = x^0 + Zw, w \in \mathbb{R}^p\}.$$

Helyettesítsünk be a célfüggvénybe:

$$\begin{aligned} \frac{1}{2}x^T Qx + c^T x &= \frac{1}{2}(x_0 + Zw)^T Q(x_0 + Zw) + c^T(x_0 + Zw) = \\ &= \frac{1}{2}w^T(Z^T QZ)w + (x_0^T Q + c^T)(x_0 + Zw) - \frac{1}{2}x_0^T Qx_0 - \\ &\quad - \frac{1}{2}x_0^T QZw + \frac{1}{2}w^T Z^T Qx_0 = \\ \frac{1}{2}w^T(Z^T QZ)w + (Qx_0 + c)^T Zw + x_0^T Qx_0 + c^T x_0 - \frac{1}{2}x_0^T Qx_0 &= \\ \frac{1}{2}w^T(Z^T QZ)w + (Qx_0 + c)^T Zw + \frac{1}{2}x_0^T Qx_0 + c^T x_0. \end{aligned}$$

Mindebből látható, hogy a probléma a következő feltétel nélküli kvadratikus programozási feladatra vezethető vissza:

$$\begin{aligned} \min \frac{1}{2}w^T(Z^T QZ)w + (Qx_0 + c)^T Zw \\ w \in \mathbb{R}^p \end{aligned}$$

Tegyük fel, hogy a redukált Hesse-mátrix, $Z^T QZ$ pozitív definit. Ekkor az optimalitás elégséges feltételei alapján, ha w^* kielégíti a

$$\nabla \left(\frac{1}{2}w^T(Z^T QZ)w + (Qx_0 + c)^T Zw \right) = 0,$$

egyenletet, akkor w^* szigorú lokális optimum. A fenti kifejezés az alábbi lineáris egyenletrendszerrel ekvivalens:

$$(Z^T QZ)w = -Z^T(Qx_0 + c).$$

Ezt megoldva (például a konjugált gradiens módszerrel) kapjuk a feladat w^* optimális megoldását, melyből az egyenlőségi feltételekkel adott probléma x^* optimális megoldása: $x^* = x_0 + Zw^*$.

A Karush-Kuhn-Tucker tételben megfogalmazott elsőrendű optimalitási feltételt felírva kiszámíthatjuk a Lagrange-szorzókat az x^* pontban. A feltétel azt mondja ki, hogy létezik egy λ^* vektor, melyre

$$Qx^* + c + A^T\lambda^* = 0 \quad (4.5)$$

Tegyük fel, hogy A teljes sorrangú mátrix, azaz létezik p darab lineárisan független sora. Ekkor az AA^T mátrix invertálható, így a (4.5) egyenletet balról beszorozva A -val, majd rendezve a következőt kapjuk:

$$\lambda^* = -(AA^T)^{-1}A(Qx^* + c).$$

Röviden említést teszünk itt az ún. **képtér módszerről**, mely Q egyszerű struktúráját használja ki. Tegyük fel, hogy Q pozitív definit és egyszerűen invertálható, például diagonális vagy blokkdiagonális mátrix. Ekkor a 4.5 egyenletet Q^{-1} -el beszorozva, majd rendezve, a megoldást és a Lagrange-szorzókat a következőképpen számíthatjuk ki:

$$\begin{aligned} x^* &= -Q^{-1}(c + A^T\lambda^*), \\ \lambda^* &= -(A^{-1}A^T)^{-1}(b + AQ^{-1}c). \end{aligned}$$

4.5. Az aktív halmaz módszer

Az aktív halmaz módszer az általános kvadratikus optimalizálási feladatot oldja meg, mégpedig oly módon, hogy visszavezeti azt egyenlőség feltételekkel adott problémák egy véges sorozatára. A feladat tehát:

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Qx + c^T x \\ & a_i^T x = b_i, \quad i \in I = \{1, \dots, p\} \\ & a_j^T x \leq b_j, \quad j \in J = \{p+1, \dots, p+m\}. \end{aligned} \quad (4.6)$$

4.5.1. Definíció. Legyen $x^k \in \mathbb{R}^n$ megengedett megoldása a 4.6 feladatnak. Az $a_j^T x \leq b_j$ ($j \in \{1, \dots, p+m\}$) feltétel aktív az x^k pontban, ha $a_j^T x^k = b_j$ teljesül, inaktív, ha $a_j^T x^k < b_j$.

Aktív halmaznak nevezzük azon feltételek indexhalmazát, melyek aktívak az x^k pontban. Jelölés: $\mathcal{A}(x^k) = I \cup \{j \in \{p, \dots, p+m\} : a_j^T x^k = b_j\}$.

A definícióból egyértelmű, hogy az egyenlőség feltétel mindig aktív.

Az algoritmus minden egyes iterációjában meghatározunk egy munkahalmazt, \mathcal{W}_k -t, melyre mindig teljesül, hogy

$$\mathcal{W}_k \subseteq \mathcal{A}(x^k).$$

Majd látni fogjuk, hogy az x^k megengedett pontban \mathcal{W}_k vagy megegyezik $\mathcal{A}(x^k)$ -val vagy eggyel kevesebb indexet tartalmaz nála.

Legyen x^* a feladat optimális megoldása, \mathcal{A}^* pedig az x^* pontban aktív feltételek halmaza.

A továbbiakban tegyük fel, hogy Q pozitív definit mátrix.

Az algoritmus lépései

AKTÍV HALMAZ MÓDSZER

Bemenet: egy x^0 megengedett megoldás és a $\mathcal{W}_0 = \mathcal{A}(x^0)$ kezdeti munkahalmaz.

0. $k = 0$

A következő lépéseket hajtjuk végre:

1. Legyen $q(x) = \frac{1}{2}x^T Qx + c^T x$.

Tekintsük az alábbi egyenlőség feltételekkel adott kvadratikus programozási feladatot:

$$\begin{aligned} \min q(x^k + d^k) \\ a_i^T(x^k + d^k) = b_i, \\ i \in \mathcal{W}_k \end{aligned} \tag{4.7}$$

Keressük meg ennek $(d^k)^*$ optimális megoldását, a *keresési irányt*. A nulltér módszerénél bemutatott módon a feladatot ekkor visszavezetjük egy feltétel nélküli problémára. Mivel Q pozitív definit, ezért $Z_k^T Q Z_k$ is pozitív definit.

Ha $(d_k)^* = 0$, akkor az x^k pont minimalizálja a q célfüggvényt a \mathcal{W}_k munkahalmaz által meghatározott feltételekre nézve. Ugorjunk a **3.** lépésre.

Ha $(d_k)^* \neq 0$, akkor ugorjunk a **2.** lépésre.

2. Meg kell határoznunk azt a legnagyobb α_k lépést, hogy az x^k pontból α_k hosszal lépve a $(d_k)^*$ irányba még megengedett pontba érkezzünk, vagyis egyetlen feltételt se sértsünk meg. (Ez felel meg a keretalgoritmusban az egyenes menti keresés lépésének.) Legyen

$$\begin{aligned} \mu_k = \max \left\{ \frac{b_i - a_i^T x^k}{a_i^T (d^k)^*} : a_i^T (d^k)^* > 0, i \notin \mathcal{W}_k \right\}. \\ \alpha_k = \min\{1, \mu_k\} \\ x^{k+1} = x^k + \alpha_k (d^k)^*. \end{aligned}$$

Ha $\alpha_k = 1$, akkor látható, hogy a (4.7) feladat optimumhelyét kaptuk. Mivel minden \mathcal{W}_k -beli feltétel teljesül az x_{k+1} pontban is, így az új munkahalmaz:

$$\mathcal{W}_{k+1} = \mathcal{A}(x^{k+1}).$$

Legyen $k = k + 1$ és ugorjunk az **1.** lépésre.

3. Az elsőrendű optimalitási feltétel vizsgálata következik. A Karush-Kuhn-Tucker rendszert felírva a (4.7) feladatra kiszámoljuk a Lagrange szorzókat.

$$Qx^k + c + \sum_{i \in \mathcal{W}_k} \lambda_i^k a_i = 0.$$

Ugorjunk a **4.** lépésre.

4. Ha $\lambda_i^k \geq 0$ minden $i \in \mathcal{W}_k$ -ra teljesül, akkor x^k a (4.6) feladat lokális minimuma. Megállunk.

Egyébként legyen \mathcal{N} azon i indexek halmaza, melyekre $\lambda_i^k < 0$, és legyen j egy tetszőleges elem \mathcal{N} -ből. Az új munkahalmaz:

$$\mathcal{W}_{k+1} = \mathcal{W}_k - \{j\}.$$

Legyen $k = k + 1$ és ugorjunk az **1.** lépésre.

Számos árazó modell létezik a megfelelő j index kiválasztására.

Ha a $Z_k^T Q Z_k$ Hesse-mátrix indefinit volt, akkor a (4.7) feladat alulról nem korlátos. Ekkor az algoritmus **1.** és **2.** lépése a következőképpen módosul:

- Keresünk egy d^k irányt, melyre $q(x^k + d^k)$ alulról nem korlátos. (Ezt a $Z_k^T Q Z_k$ Hesse-mátrix megfelelő felbontása révén tehetjük meg. (ld. []))
- kiszámoljuk μ_k -t:

$$\mu_k = \max \left\{ \frac{b_i - a_i^T x^k}{a_i^T d^k} : a_i^T d^k > 0, i \notin \mathcal{W}_k \right\}.$$

- az új megengedett vektor:

$$x^{k+1} = x^k + \mu_k d^k.$$

- az új munkahalmaz:

$$\mathcal{W}_{k+1} = \mathcal{W}_k \cup \mathcal{A}(x^{k+1}).$$

4.6. Lemke féle pivotalgorithmus

A lineáris komplementaritási feladat megoldására számos módszert dolgoztak ki. A pivotmódszerek közé soroljuk a legkisebb index criss-cross módszert, mellyel a konvex kvadratikus programozási feladatot oldhatjuk meg [7] és a Lemke-féle pivotalgorithmust [12], melyet a következőkben mutatunk be. Léteznek belsőpontos

algoritmusok is a probléma megoldására, ezzel kapcsolatban érdemes megnézni a [8, 18] könyveket. Tekintsük tehát a lineáris komplementaritási problémát:

$$w - Mz = q$$

$$w_j \geq 0, z_j \geq 0, j = 1, \dots, p \quad (4.8)$$

$$w_j z_j = 0, j = 1, \dots, p,$$

ahol $M \in \mathbb{R}^{p \times p}$ és $q \in \mathbb{R}^p$. Ha q nemnegatív, akkor a feladatra azonnal adódik a $w = q, z = 0$ megoldás. A legtöbb esetben azonban nincs ilyen szerencsénk, ezért bevezetjük a z_0 mesterséges változót, és a következő feladatot írjuk fel:

$$w - Mz - ez_0 = q \quad (4.9)$$

$$w_j \geq 0, z_j \geq 0, z_0 \geq 0, j = 1, \dots, p \quad (4.10)$$

$$w_j z_j = 0, j = 1, \dots, p, \quad (4.11)$$

ahol e a csupa egyest tartalmazó p dimenziós vektort jelöli. Feltéve, hogy $z_0 = \max\{-q_i : 1 \leq i \leq p\}$ ezen probléma egy megengedett induló megoldását kapjuk a $z = 0$ és $w = q + ez_0$ választással. Az algoritmusban a szimplex módszernél megismert pivotálást, azaz báziscsere műveletet alkalmazzuk. Az a célunk, hogy báziscserék egy meghatározott sorozatának végrehajtása révén a z_0 változó értékét nullává tegyük, ami által a (4.8) feladat egy komplementáris bázismegoldásához jutunk.

4.6.1. Definíció. Tekintsük a (4.9) - (4.11) problémát. A rendszer egy (w, z, z_0) megengedett megoldását majdnem komplementáris megengedett bázismegoldásnak nevezzük, ha az alábbiak teljesülnek rá:

1. (w, z, z_0) megengedett bázismegoldása a (4.9)- (4.10) rendszernek,
2. létezik olyan s index, hogy mind w_s mind z_s nem-bázisváltozók,
3. z_0 bázisváltozó és minden komplementáris (w_j, z_j) párra teljesül, hogy közülük pontosan egy bázisváltozó, ahol $j \neq s$.

A Lemke-féle pivotalgorithmus majdnem komplementáris megengedett bázismegoldások sorozatán halad végig. Mindig a bázist éppen elhagyó változó komplementáris párja fog belépni a bázisba, a bázisból kilépő változót pedig (a szimplex módszernél megismert generálóelem-választási szabály szerint) hányadostesztel határozzuk meg. Belátható, hogy a mesterséges változó, z_0 értéke az iterációk során monoton csökken. Értéke kétféleképpen válhat nullává: vagy úgy, hogy elhagyja a bázist, vagy úgy, hogy a bázisban maradván a nulla értéket veszi fel.

LEMKE-FÉLE PIVOTALGORITMUS

Inicializálás Ha $q \geq 0$, akkor megállunk, $(w, z) = (q, 0)$ komplementáris bázismegoldása a feladatnak. Egyébként tekintsük a (4.9)-(4.10) rendszert a következő bázistábla formájában:

	w_1	\dots	w_p	z_1	\dots	z_p	z_0	RHS
w_1	I_p			$-M$			-1	q
\vdots							\vdots	
w_p							-1	

ahol I_p a p dimenziós egységmátrix. Az első függőleges oszlop mindig azon változókat jelenti, melyek aktuálisan a bázisban vannak. Legyen $-q_s = \max\{-q_i : 1 \leq i \leq p\}$ és hajtsuk végre a báziscserét az s . sorban a z_0 oszlopban. A bázistábla eszerint frissül, a bázisváltozók tehát most a nemnegatív z_0 és w_i , $i = 1, \dots, p$, $i \neq s$ lesznek, azaz egy majdnem komplementáris megengedett bázismegoldást kaptunk. Legyen $y_s = z_s$.

Fő lépések

1. Legyen d_s a bázistábla y_s változónak megfelelő oszlopa. Ha $d_s \leq 0$, akkor ugorjunk a **4.** lépésre. Legyen q' az aktuális bázisváltozók értékeit tartalmazó jobboldali oszlop a táblában. A hányadoseszt elvégzésével határozzuk meg az r indexet:

$$\frac{q'_r}{d_{rs}} = \min \left\{ \frac{q'_i}{d_{is}} : d_{is} > 0 \right\}.$$

Ha az r . sorban a bázisváltozó z_0 , ugorjunk a **3.** lépésre, egyébként ugorjunk a **2.** lépésre.

2. Az r . sorban a bázisváltozó w_l vagy z_l valamely $l \neq s$ indexre. Az y_s változó belép a bázisba és a táblát frissítjük az r . sorban y_s oszlopban történő báziscsere eredménye szerint. Ha az éppen kilépő bázisváltozó:

- w_l , akkor $y_s = z_l$,
- z_l , akkor $y_s = w_l$.

Térjünk vissza az **1.** lépésre.

3. A z_0 változó elhagyja a bázist, és az y_s változó belép a bázisba. Hajtsuk végre a

báziscserét a z_0 sorban y_s oszlopban, ezáltal a feladat egy *komplementáris megengedett bázismegoldását* kapjuk. Megállunk.

4. Az

$$\mathbf{R} = \left\{ \begin{pmatrix} w \\ z \\ z_0 \end{pmatrix} + \lambda d : \lambda \geq 0 \right\}$$

halmaz pontjai kielégítik a (4.9)-(4.11) rendszert, ahol $(w, z, z_0)^T$ a legutóbbi bázistábla által megadott majdnem komplementáris bázismegoldás, $d \in \mathbb{R}^{2p+1}$ vektor, melyre:

$$d_i = \begin{cases} 1 & \text{ha } y_s \text{ az } i. \text{ változó} \\ -d_s & \text{ha } i \text{ bázisváltozó} \\ 0 & \text{egyébként} \end{cases}$$

Megállunk.

Az algoritmus konvergenciájára és végességére vonatkozó tételt csak a kvadratikus programozási feladatra alkalmazva mutatjuk be.

4.6.1. Tétel. *Tekintsük a (3.14) problémát. Tegyük fel, hogy a megengedett megoldások halmaza nemüres, és tegyük fel, hogy a pivotálgoritmust alkalmazzuk a feladat KKT feltételeinek felírásából származó (3.15) probléma megoldására. Ha a feladat nem degenerált, akkor a következő feltételek bármelyikének fennállása esetén a Lemke-féle pivotálgoritmus véges számú iteráció után leáll.*

1. Q pozitív szemidefinit és $c = 0$.

2. Q pozitív definit.

3. Q minden eleme nemnegatív és a főátlójában lévő elemek pozitívak.

Továbbá Q pozitív szemidefinitése esetén, ha az algoritmus a 4. lépésben áll le, abból következik, hogy az optimális megoldás alulról nem korlátos.

A lineáris komplementaritási probléma degeneráltságáról a [15] cikkben olvashatunk részletesebben.

A második fejezetben bevezetett két konvex kvadratikus programozási feladat esetében a Q mátrix pozitív szemidefinit, de előfordulhat, hogy egy kisebb dimenziós részfeladatot (Q almátrixát) véve pozitív definit esettel is találkozunk. A következő fejezetben a Lemke-féle pivotálgoritmust beprogramozzuk MATLAB-ban, és megvizsgáljuk, milyen eredményeket kapunk a portfólióelméleti modellekre alkalmazva.

5. fejezet

A portfólió kiválasztási probléma modelljeinek megvalósítása MATLAB környezetben

Az utóbbi két fejezet ismereteivel felvértezve visszatérünk a második fejezet végén bevezetett portfólió kiválasztási modellekre és megnézzük, hogyan alkalmazhatóak a gyakorlatban. A korábban tárgyalt modellek közül a második - amely a várható hozam maximalizálását célozta meg a kockázat korlátozása mellett - nemlineáris feltételt tartalmaz, tehát nem sorolható a kvadratikus programozási feladatok osztályába, így a továbbiakban az alábbi két modellel fogunk dolgozni:

(1)

$$\begin{aligned} \min \mathbf{w}^T \mathbf{Q} \mathbf{w} \\ \mathbf{r}^T \mathbf{w} &\geq \mathbf{r}^* \\ \sum_{i=1}^N w_i &= 1 \\ \mathbf{w} &\geq 0, \end{aligned}$$

(2)

$$\begin{aligned} \max \mathbf{r}^T \mathbf{w} - \frac{k}{2} \mathbf{w}^T \mathbf{Q} \mathbf{w} \\ \sum_{i=1}^N w_i &= 1 \\ \mathbf{w} &\geq 0. \end{aligned}$$

Mindkét feladat egy konvex kvadratikus programozási probléma, hiszen a Q kovariancia mátrix pozitív szemidefinit.

A MATLAB beépített `quadprog` parancsával, illetve az általunk implementált Lemke-féle pivotálgoritlussal is megoldjuk (1)-et és (2)-t, majd ezeket elemezzük elsősorban alkalmazhatóság és futási eredmények szempontjából. Konkrét példákon keresztül grafikusán is szemléltetjük az eredményeket.

5.1. A `quadprog` parancs

A MATLAB optimalizálási eszköztárához tartozik a `quadprog` parancs, mely a nemlineáris programozási feladatok kvadratikus feladatosztályára alkalmazható. Először ezt fogjuk használni a portfólió kiválasztási probléma megoldására, ezért most röviden áttekintjük a szintaktikáját, és a mögötte álló megoldó módszerről is esik néhány szó.

A `quadprog` parancs a következő alakú feladatot oldja meg:

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Hx + f^T x \\ & A \cdot x \leq b \\ & Aeq \cdot x = beq \\ & lb \leq x \leq ub. \end{aligned}$$

A parancs paraméterei:

```
[x,fval,exitflag,output,lambda] = quadprog(H,f,A,b,Aeq,beq,lb,ub,...
x0,options).
```

A bemenő paraméterek:

- `H,f,A,b,Aeq,beq,lb,ub`: a fentieknek megfelelő oszlopvektorok, illetve mátrixok,
- `x0`: induló megoldás,
- az `options`: beállításait az `optimset` paramétereinek módosításával érhetjük el.

A kimenő paraméterek:

- `x`: az optimális megoldásvektor,
- `fval`: a célfüggvény értéke az optimumban,
- `exitflag`: értéke egy egész szám, mely az algoritmus leállítását előidéző ok azonosítására szolgál,
- `output`: az optimalizálási folyamattal kapcsolatos információkat tartalmaz, például a használt algoritmusról, az elvégzett iterációk számáról,
- `lambda`: a Lagrange szorzók értéke az optimumpontban.

Az

```
options = optimset('LargeScale','on|off');
```

beírásával megadhatjuk, hogy `large-scale (on)` vagy `medium-scale (off)` módszerrel kívánjuk megoldani a feladatot. A `large-scale` módszert - mely az alapértelmezett beállítás - akkor ajánlatos használni, ha a feladatban vagy nincsenek lineáris korlátozó feltételek, vagy csak egyenlőség feltételek szerepelnek alsó és felső korlátok nélkül (azaz az x értékészlete a teljes n -dimenziós tér) és ezek száma nem haladja meg az n dimenziószámot. Minden egyéb esetben a `medium-scale` módszert kell alkalmazni, a beállítás elmulasztása esetén figyelmeztető üzenetet kapunk. A `medium-scale` módszer a feltételes nemlineáris optimalizálási problémát az ún. *szekvenciális kvadratikus programozási* eljárással oldja meg, ami speciálisan a kvadratikus programozási probléma esetében az aktív halmaz módszert jelenti. Az (1) és (2) modelleket a `quadprog` paranccsal megoldva tehát az aktív halmaz módszert fogjuk alkalmazni.

5.2. A Lemke-féle pivotalgorithmus megvalósítása MATLAB-ban

5.2.1. A lemke függvény

A függelékben megtaláljuk a Lemke-féle pivotalgorithmus kódját, mely megoldja a (4.8) problémát és lépései a 4.6 fejezetben leírtakkal azonosak. A pivotálás műveletét és a hányadostesztet egy-egy külön függvényben (`pivot` és `hányadosteszt`) írtuk meg, melyeket a program a megfelelő helyeken meghív. A programkódok a mellékelt cd-lemezen is megtalálhatóak. A `lemke` parancs paraméterei:

```
[wz,z0,d,it_szam,iteraltak] = lemke(M,q,max_iter).
```

A bemenő paraméterek:

- `M,q` a lineáris komplementaritási problémát meghatározó négyzetes mátrix és oszlopvektor,
- `max_iter` a maximális iterációk száma (opcionális).

A kimenő paraméterek:

- `wz`: a w és z megoldásvektorok,
- `z0`: a mesterséges változó értéke az algoritmus leállásakor,
- `d`: az algoritmus 4. lépésében meghatározott irány,
- `it_szam`: a végrehajtott iterációk száma,
- `iteraltak`: az egyes iterációk során kapott `wz` vektorok.

5.2.2. A lemkequad függvény

A

$$\begin{aligned} \min & \frac{1}{2}x^T Hx + f^T x \\ & A \cdot x \leq b \\ & Aeq \cdot x = beq \\ & x \geq 0 \end{aligned}$$

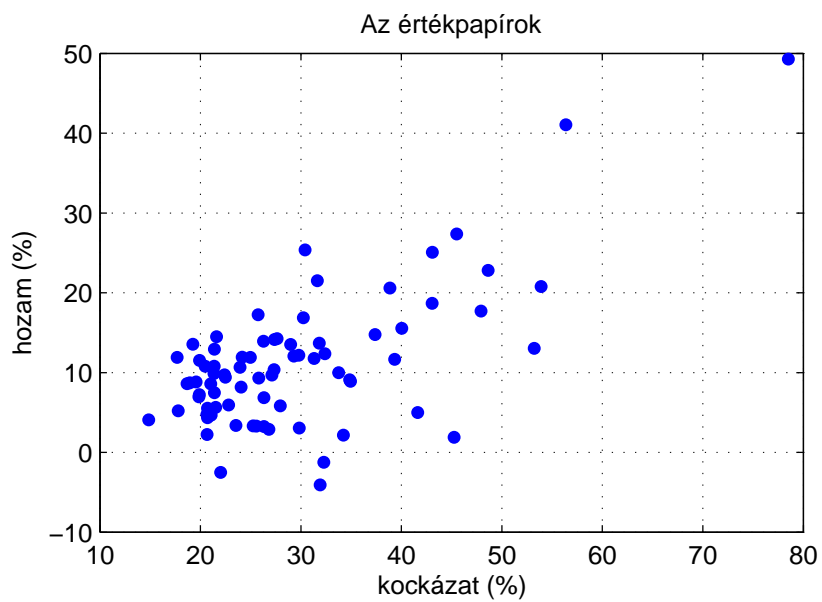
formában adott kvadratikus programozási feladatot a Lemke-féle pivotálgoritmussal a `lemkequad` függvény meghívásával oldhatjuk meg egyszerűen. Ennek kódját szintén megtalálhatjuk a függelékben. A parancs paraméterei:

```
[x,z0,d,it_szama,iteraltak] = lemkequad(H,f,A,b,Aeq,beq,max_iter)
```

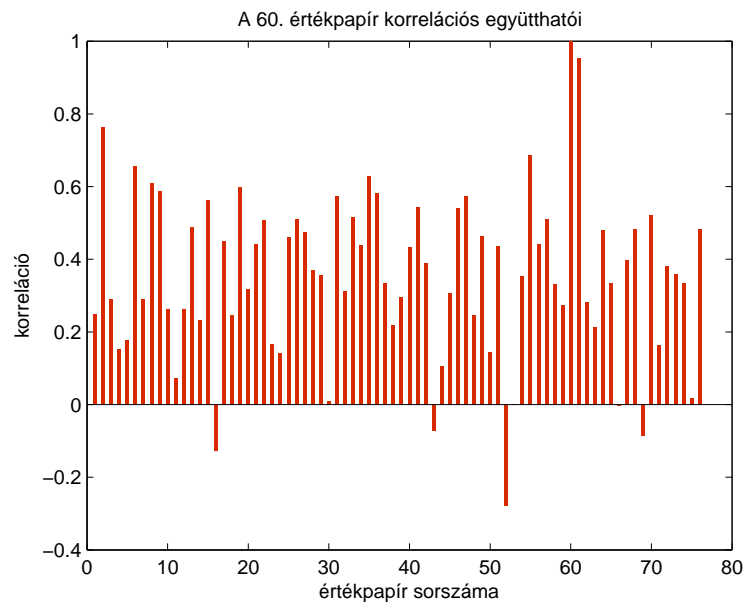
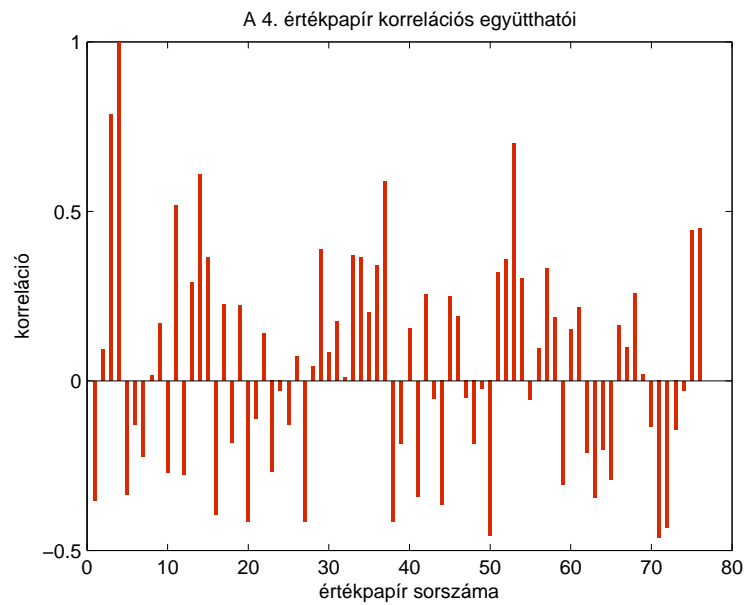
A bemenő és kimenő paraméterek az eddigiekhez hasonlóan értelmezendők.

5.3. A felhasznált adathalmaz

Az algoritmusok teszteléséhez és az elemzések elvégzéséhez kezdetben egy 76 darab értékpapír árfolyamadatait tartalmazó táblázat állt rendelkezésünkre, mely Fábíán Csabától származik. A táblázat oszlopaiban az azonosító nélküli értékpapírok 1992.12.01 és 2005.06.01. közötti havi árfolyamadatai szerepeltek. A táblázatot MATLAB-ba beolvasva elvégeztük az éves hozamok számítását, amely a gyakorlatban úgy valósult meg, hogy minden értékpapírra sorra kiszámoltuk az $i + 12.$ és $i.$ havi árfolyam hányadosát kivontunk belőle egyet, majd szoroztuk 100-al. A kapott mátrixra a `mean`, `cov` és `corr` parancsok segítségével normális eloszlást illesztettünk, megkapva így az éves várható hozamokat, valamint a kovariancia és korreláció mátrixot. A várható hozamok és a kockázat tehát mindig százalékban értendők. Az így nyert 76 darab eszközt tartalmazó értékpapírhalmoz az alábbi ábrán látható.



A kovariancia mátrix sajátértékeit vizsgálva mind pozitívnak adódnak, így a megoldandó feladatokban a Hesse-mátrix pozitív definit. Az adatokat alapvetően a pozitív gyenge korreláció jellemzi, a legkisebb korrelációs együttható -0.4634 volt, melyet a 4. értékpapírnál tapasztaltunk, míg a legnagyobb 0.9549 , mely a 60. sorszámú értékpapírt jellemezte. A következő oldalon található két ábrán ezen értékpapírok többi értékpapírral vett korrelációs együtthatóit figyelhetjük meg:



Az értékpapírokat jellemző néhány fontos mérőszám:

minimális hozam -4.0743%	maximális hozam 49.2837%	átlagos hozam 11.1061%
minimális kockázat 14.8481%	maximális kockázat 78.5005%	átlagos kockázat 29.4275%

Az átlagos hozamú portfólió pontosan az, mely az összes eszközt azonos súllyal véve tartalmazza. Kiszámítva azt kapjuk, hogy a hozzá tartozó kockázat 15.1906%, tehát ezen portfólió megvalósítása igen kockázatos befektetés lenne. A továbbiakban

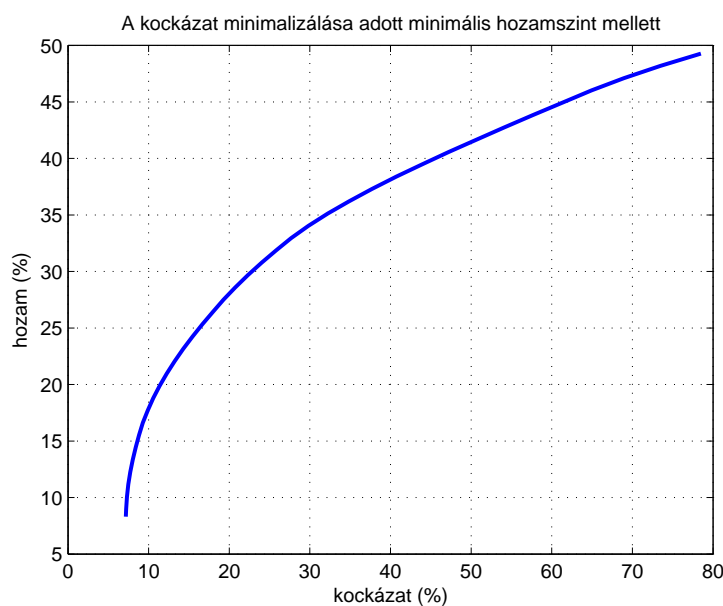
megnézzük, hogyan és mennyit tudunk javítani ezeken az értékeken a megfelelő tőkeallokáció kialakításával, azaz a kvadratikus programozási modell megoldásával.

5.4. Az első modell megoldása

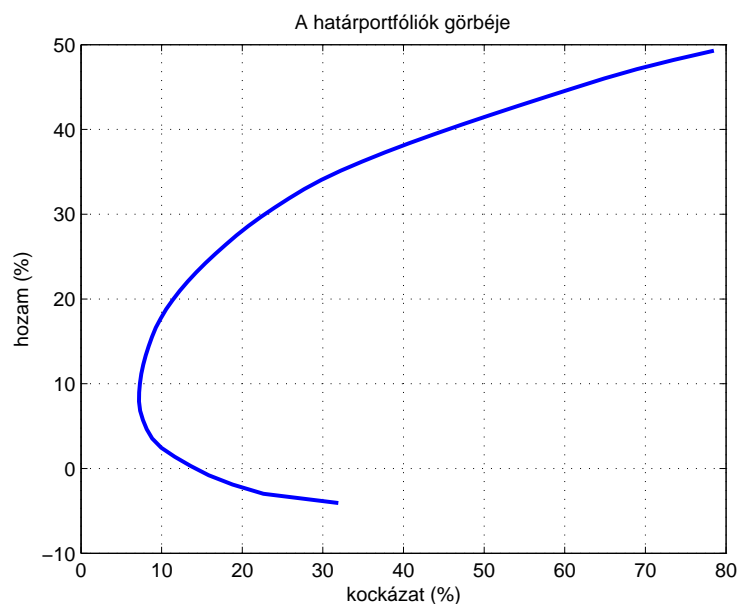
A mellékelt cd lemezen is megtalálható modell1.m nevű MATLAB fájlt először a teljes értékpapírhalmasra futtattuk le. A programban a kiválasztott algoritmus (quadprog vagy lemke) összesen 50 különböző r^* értékre (a várható hozamra vonatkozó alsó korlát) fut le, ahol r^* értéke az aktuális legalacsonyabb és legmagasabb várható hozam között van. Az így nyert 50 db portfólió grafikus megjelenítése a parancs kimenetének része. Ezenkívül hasonló módon 50-szer fut le az (1) modell módosított változata, amelyben a várható minimális hozamra vonatkozó egyenlőtlenség feltétel helyett egyenlőség feltétel szerepel, azaz a parancs a határportfóliók görbéjét is kiszámolja, valamint grafikusán megjeleníti. További kimenő paraméter a két modell futási ideje. A parancsot Windows XP operációs rendszerben, 2.80 GHz-es Intel Pentium D CPU processzorú számítógépen futtatva az alábbi futási időket (sec) kaptuk.

	quadprog	lemke
(1) modell (hatékony portfóliók)	16.3313	2.0773
(1) modell módosítása (határportfóliók)	11.2381	1.5158

A második fejezetben említettük, hogy az (1) modell abban az esetben felel meg a hatékony portfóliók halmazának, ha a Hesse-mátrix pozitív definit. Mivel esetünkben ez teljesül, az (1) modell által meghatározott portfóliók hatékonyak:



A határportfóliók görbéje:



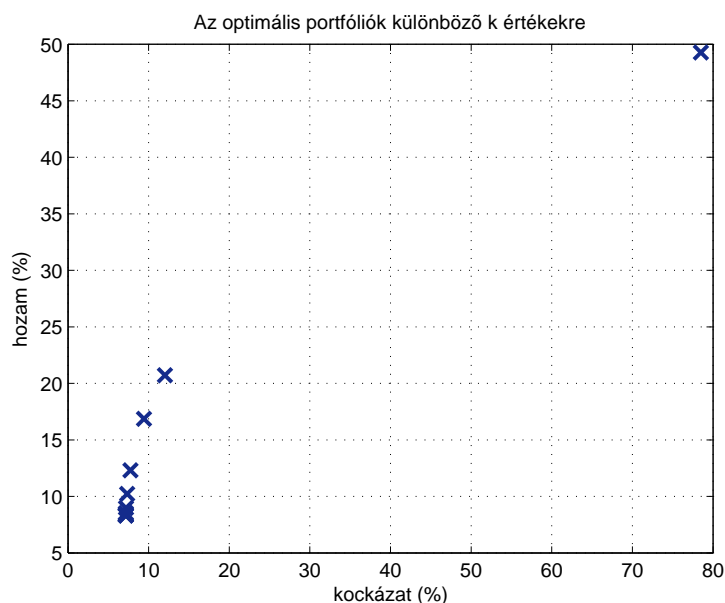
Vegyük észre, hogy a határportfóliók görbéje tartalmazza a hatékony portfóliók görbéjét.

5.5. A második modell megoldása

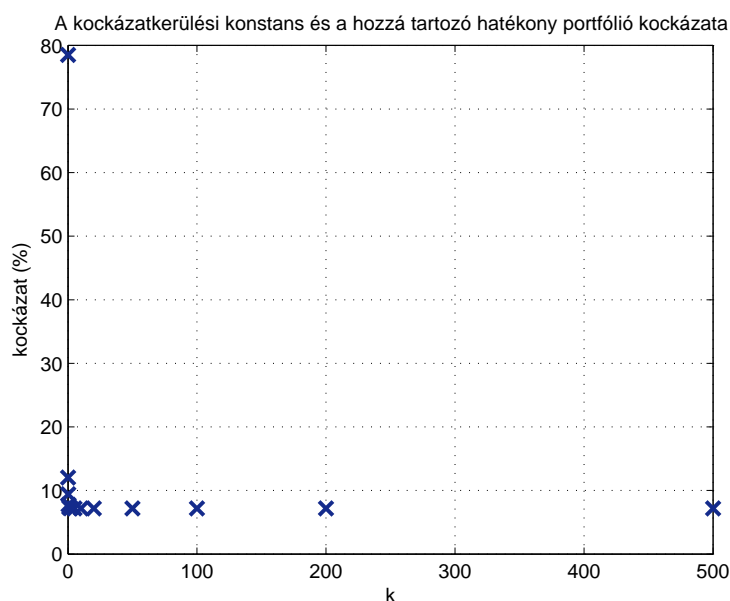
A `modell2.m` fájl futtatásával a (2) modellt oldhatjuk meg az értékpapírhalma-
zunk egy tetszőleges részhalmazára. Az alábbi táblázat tartalmazza a teljes értékpapír-
halmazon kapott eredményeket. A negyedik és ötödik oszlopban szereplő futási
idők másodpercben értendők.

k	optimális hozam (%)	optimális kockázat (%)	quadprog	lemke
0	49.2837	78.5005	0.1595	0.0058
0.1	20.7175	12.0300	0.2080	0.0256
0.2	16.8650	9.4164	0.2072	0.0257
0.5	12.3024	7.7418	0.2044	0.0356
1	10.2201	7.3315	0.2187	0.0439
2	8.9762	7.2031	0.2166	0.0443
5	8.5250	7.1793	0.2030	0.0454
10	8.4165	7.1770	0.1988	0.0448
20	8.3623	7.1765	0.2145	0.0464
50	8.3297	7.1763	0.1989	0.0452
100	8.3189	7.1763	0.2002	0.0449
200	8.3135	7.1763	0.2034	0.0452
500	8.3102	7.1763	0.1991	0.0484

Abban az esetben, amikor k értéke nulla, a modell egy lineáris programozási feladat, a MATLAB ezért automatikusan meghívta a `linprog` megoldót. Látható továbbá, hogy ebben az esetben az optimális hozam a legnagyobb hozamú értékpapír hozama, a hozzá tartozó optimális kockázat pedig a legmagasabb kockázatú értékpapír hozama. Mindebből következik, hogy esetünkben a legnagyobb hozamú értékpapír egyben a legnagyobb kockázatú értékpapír is. Az egyes k értékek esetében kapott hatékony portfóliókat az alábbi ábrán figyelhetjük meg.



A kockázatkerülési konstans függvényében ábrázolva a optimális kockázatot az alábbi ábrát nyerjük:

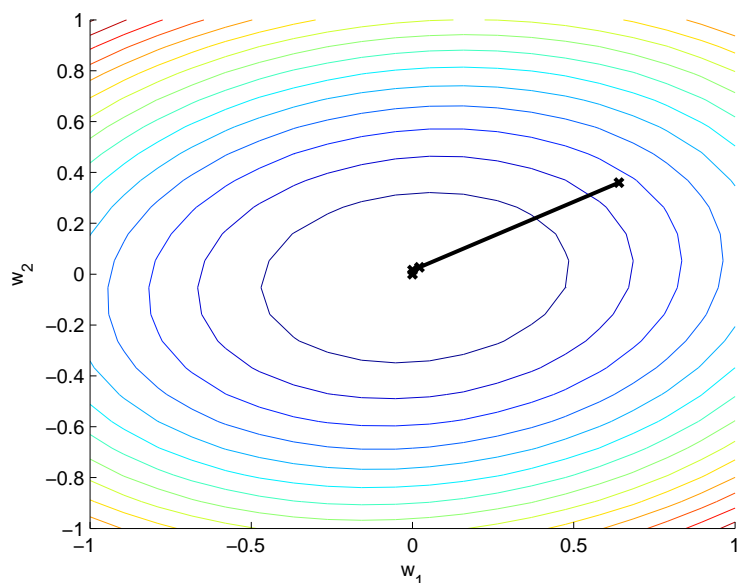


A táblázatból és az utóbbi ábrából jól látható, hogy k értékét növelve az optimális portfólió várható hozama 8.31%-hoz, míg kockázata 7.176%-hoz közelít. (A kockázatot, melyet nem tudunk kiküszöbölni a pénzügytanban *piaci* vagy *nem diversifikálható kockázatnak* nevezik.) Itt említjük meg, hogy a pénzügyekben általában feltesszük, hogy lehetőség van a piacon egy adott kamatláb mellett kockázatmentesen befektetni. Tehát rendelkezésünkre áll még egy értékpapír (például államkincstárjegy), melynek kockázata nulla, korrelációs együtthatója minden értékpapírral nulla, várható hozama pedig a fix kamatláb. Egy újabb optimalizálási folyamatot elvégezve, melyben immáron csak a befektető kockázatkerülésének leginkább megfelelő hatékony portfólió és a kockázatmentes eszköz szerepelnek, megkapjuk a ténylegesen optimális tőkeallokációt. Ez a lépés jelentősen csökkentheti a kockázatot, esetlegesen a várható hozam csökkenésével egyidejűleg is, ám mint tudjuk egy kockázatkerülő befektetőnél a biztosabb alacsonyabb hozam elsőbbséget élvez egy kevéssel magasabb, de bizonytalanabb hozammal szemben. Lássunk egy példát erre, tegyük fel, hogy a kockázatmentes kamatláb 6% a befektető kockázatkerülési konstansa pedig 0.5. Újra optimalizálva a fenti táblázat negyedik sorában olvasható hatékony portfólió és a kockázatmentes eszköz segítségével a következő eredményeket kapjuk:

$$\begin{aligned} \text{várható hozam: } & 7.3254\% \\ \text{kockázat: } & 1.6281\% \end{aligned}$$

Ennek eléréséhez a befektetőnek tőkéje 78.97%-át a kockázatmentes eszközbe, míg 21.03%-át a hatékony portfólióba kell fektetnie.

Vizsgáljunk most egy olyan esetet, amikor az aktuális értékpapírhalmoz kételemű. Ebben az esetben a célfüggvényünk kétváltozós, így lehetőségünk nyílik a szintvonalak megjelenítésére. Az alábbi ábrán a 10 és 11 sorszámú értékpapírok, $k = 0.5$ kockázatkerülési konstans által meghatározott célfüggvény szintvonalai láthatóak, valamint a Lemke algoritmus egyes iterációiban nyert pontok (súlyok).



A fentiek után látható, hogy a Lemke algoritmust alkalmazva lényegesen gyorsabb volt a megoldási folyamat, mint az aktív halmaz módszer esetében. Meg kell azonban jegyeznünk, hogy az általunk megoldott problémák kisméretű feladatok, a nagy eltérések a két algoritmus futási ideje között legfőképp ebből adódtak. A nemnegativitási feltételt nem számítva az (1) modell esetében 2, míg a (2) modell esetében 1 darab feltételünk volt. Ha a feltételek száma n nagyságrendű és ráadásul egyenlőség feltételek is szerepelnek a feladatban (ami két egyenlőtlenségnek felel meg), akkor a Lemke-féle pivotálgoritmusban felépített táblázat, ennél fogva a pivotálás és hányadoseszt elvégzése minden egyes iterációban lényegesen több műveletet igényel. A lineáris komplementaritási probléma megoldására léteznek nagyméretű feladatokat is hatékonyan kezelő eljárások, ún. belsőpontos módszerek. Ezekkel kapcsolatban érdemes megnézni a [8, 18] könyveket. Egy másik specialitás, amely miatt hatékonyan tudtuk alkalmazni a Lemke féle pivotálgoritmust az az, hogy a Hesse-mátrix pozitív definit volt. Az (1) modell esetében ez nem jelent megkötést, hiszen ott a c vektor nulla volt (a célfüggvényben csak másodfokú tag szerepelt), így a 4.6.1. Tétel értelmében a pivotálgoritmus véges sok lépésben megtalálja az optimumot. A (2) modellt tekintve azonban - a fenti tételre hivatkozva ismét - nem garantálható, hogy pozitív szemidefinit mátrix esetén a kvadratikusan optimális megoldásával áll le az algoritmus.

Összefoglalva azt mondhatjuk tehát, hogy a portfóliókiválasztási problémát jól leíró kvadratikusan programozási modellek kis dimenziójuk révén hatékonyan megoldhatóak a beprogramozott Lemke féle pivotálgitmussal. A fentiekben azt is láttuk, hogy az optimalizálási folyamat ezzel az algoritmussal lényegesen gyorsabb volt, mint az aktív halmaz módszer esetében.

6. fejezet

Összefoglalás

A szakdolgozat célja a portfólió kiválasztási probléma ismertetése, és lehetséges matematikai modelljei közül a kvadratikus programozási modell vizsgálata volt. Áttekintettük a probléma tárgyalásához elengedhetetlenül szükséges matematikai alapokat: a valószínűségszámítás, az algebra, az analízis néhány fontosabb tételét, definícióját; valamint a nemlineáris programozás, majd azon belül a kvadratikus programozás legfontosabb elméleti eredményeit, az optimalitási feltételeket, a lineáris komplementaritási feladattal való kapcsolatot és az algoritmusokat. A bemutatott algoritmusok közül kettőt használtunk a portfólió kiválasztási probléma megoldására. A saját kóddal futtatott Lemke-féle pivotalgorithmussal, és a MATLAB beépített megoldójával tesztelt aktív halmaz módszerrel egy konkrét adatsoron kapott futási eredményeket a dolgozat utolsó fejezetében hasonlítottuk össze.

7. fejezet

Köszönetnyilvánítás

Szeretnék köszönetet mondani témavezetőmnek Nagy Mariannak a készülő dolgozat formájával és tartalmával kapcsolatos hasznos észrevételeiért, tanácsaiért, valamint a források kiválasztásában nyújtott segítségéért. Köszönöm továbbá Fábián Csabának a rendelkezésemre bocsátott adatokat.

A. Függelék

Függelék

A.1. A Lemke féle pivotalgorithmus kódja

A programkód lépései a 4.6 szakaszban leírtakkal azonosak.

hanyadosteszt.m

```
function [i]=hanyadosteszt(B,c)

n = size(B,1);
rhs = B(:,2*n+2);
oszlop = B(:,c);
a = 1:n;

k = a(oszlop' >0);
[mi, index] = min(rhs(k)./oszlop(k));
i = k(index);
```

pivot.m

```
function [B]=pivot(B,r,c)

akt_r = B(r,:);
B(r,:) = B(r,:)/B(r,c);

p = size(B,1);
for i=1:p
    if(i~=r)
```

```

        B(i,:)=(B(i,:).*akt_r(c) - (akt_r.*B(i,c)))/akt_r(c);
    end
end

```

megoldas.m

```
function [wz,z0]=megoldas(tablazat,bazisvaltozok)
```

```
n = size(tablazat,1);
```

```
megoldas = zeros(2*n+1,1);
```

```
for i=1:n
```

```
    x = bazisvaltozok(i);
```

```
    megoldas(x) = tablazat(i,2*n+2);
```

```
end
```

```
wz = [megoldas([1:n]),megoldas([n+1:2*n])];
```

```
z0 = megoldas(2*n+1);
```

lemke.m

```
function [wz,z0,d,it_szam,iteraltak] = lemke(M,q,max_iter)
```

```
% Ez a parancs megoldja a
```

```
%
```

```
%  $w - Mz = q$ 
```

```
%  $w_j * z_j = 0, \quad j=1..p$ 
```

```
%  $w \geq 0, \quad z \geq 0$ 
```

```
%
```

```
% linearis komplementaritasi feladatot, ahol M egy p x p valos matrix,
```

```
% q egy p dimenzios oszlopvektor.
```

```
if (nargin < 2)
```

```
    error('Legalabb ket parametert kell megadni.');
```

```
end
```

```
if (size(M,1)~=length(q))
```

```
    error('M es q nem megfelelo meretuek.');
```

```
end
```

```

if(nargin==2)
    max_iter = Inf;
end

p = length(q);

%::::::::::::::::::::::::::::::::::::INICIALIZALAS::::::::::::::::::::::::::::::::::::
d=[];
null = zeros(p,1);
it_szam = 0;
iteraltak = [q,null];

if (q >= null)
    disp('A q vektor nemnegativ. A megoldas (w,z)=(q,0).');
    wz = [q,null];
else

%tablazat konstrualasa
W = eye(p);
Z = -M;
Z0 = -ones(p,1);
RHS = q;

tablazat = [W,Z,Z0,RHS];
bazisvaltozok = [1:p];

%bazisvektor keresese
for i=1:p
    if(-q(i)== max(-q))
        s = i;
        break
    end
end

y_s = 2*p+1;

tablazat = pivot(tablazat,s,y_s);
bazisvaltozok(s) = 2*p+1;
akt_megoldas = megoldas(tablazat,bazisvaltozok);

```

```

iteraltak = [iteraltak,akt_megoldas];
y_s = p+s;

end
%-----

lepes = 1;

while(lepes~=0)
    if(it_szam==max_iter)

        disp('Az iteraciok szama elerte a max_iter erteket, ezert...
              az algoritmus leallt.');
```

[wz,z0] = megoldas(tablazat,bazisvaltozok);

```

        lepes = 0;
    end
    it_szam = it_szam + 1;

%:::::::::::1. LEPES:::::::::::
    if(lepes==1)
        d_s = tablazat(:,y_s);

        if (d_s <= null)
            lepes = 4;
        else

            r = hanyadosteszt(tablazat,y_s);

            if(bazisvaltozok(r)==2*p+1)
                lepes = 3;
            else
                lepes = 2;
            end
        end
    end

end

%:::::::::::2. LEPES:::::::::::

    if(lepes==2)
        regi_bazisvaltozo = bazisvaltozok(r);
    end
end

```

```

if (d_s <= null)
    lepes = 4;
else

tablazat = pivot(tablazat,r,y_s);
bazisvaltozok(r) = y_s;
akt_megoldas = megoldas(tablazat,bazisvaltozok);
iteraltak = [iteraltak,akt_megoldas];

if(regi_bazisvaltozo <= p)
    y_s = regi_bazisvaltozo + p;
else
    y_s = regi_bazisvaltozo-p;
end

lepes=1;
end
end

```

```

%:.....:3. LEPES:.....

```

```

if(lepes==3)

    tablazat = pivot(tablazat,r,y_s);
    bazisvaltozok(r)=y_s;
    disp('Az algoritmus komplementaris bazismegoldast talalt,...
        ezert leallt.');
```

```

    [wz,z0] = megoldas(tablazat,bazisvaltozok);
    iteraltak = [iteraltak,wz];

    lepes=0;
end

```

```

%:.....:4. LEPES:.....

```

```

if(lepes==4)
    disp('Az algoritmus nem talalt kilepo bazisvaltozot,...
        ezert leallt.');
```

```

    [wz,z0] = megoldas(tablazat,bazisvaltozok);

```

```
    iteraltak = [iteraltak,wz];  
    d = zeros(1,2*p+1);  
    d(y_s) = 1;  
    for i=1:p  
        h = bazisvaltozok(i);  
        d(h) = -d_s(i);  
    end  
    lepes=0;  
end  
  
end
```

Ezen programok és a felhasznált további kódok megtalálhatóak a dolgozathoz mellékelt cd lemezen.

Irodalomjegyzék

- [1] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear programming: Theory and Algorithms*. John Wiley and Sons, New York, 1993.
- [2] R. W. Cottle, G. B. Dantzig. *Complementary pivot theory of mathematical programming*. *Linear Algebra and its Applications*, 1:103-125, 1968.
- [3] R. W. Cottle et al. *The linear complementarity problem*. Boston, Mass. Academic Press, 1992.
- [4] R. W. Cottle. *Linear complementarity since 1978*. *Nonconvex Optimization and Its Applications - Variational Analysis and Applications*, Part 2: 239-257, Springer US, 2005.
- [5] W. Karush. *Minima of Functions of Several Variables with Inequalities as Side Constraints*. M.Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago, Chicago, Illinois, 1939.
- [6] M. G. Kendall. *The Analysis of Economic Time-Series-Part I: Prices*. *Journal of the Royal Statistical Society. A (General)* 116 1: 11-34, 1953.
- [7] E. de Klerk, C. Roos, Terlaky Tamás. *Nemlineáris optimalizálás*. Budapesti Közgazdaságtudományi és Államigazgatási Egyetem, Aula kiadó, Budapest, 2004.
- [8] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise. *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems.*, Volume 538 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1991.
- [9] Kovács Margit. *A nemlineáris programozás elmélete*. Typotex Kiadó, Budapest, 1997.
- [10] Kovács Margit. *Operációkutatás II. egyetemi jegyzet*. Operációkutatási Tanszék, Eötvös Loránd Tudományegyetem, 2005. <http://www.cs.elte.hu/margo/jegyzet/opkut/progmat-opkut-pdf.pdf>

- [11] H. W. Kuhn, A. W. Tucker. *Nonlinear programming*. Proceedings of 2nd Berkeley Symposium: 481-492, Berkeley: University of California Press, 1951.
- [12] C. E. Lemke. *On Complementary Pivot Theory*. Mathematics of the Decision Sciences, 1968.
- [13] Mádi Nagy Gergely. *Kvadrátikus programozás alkalmazása a portfólió problémára*. Szakdolgozat. Eötvös Loránd Tudományegyetem, 1997.
- [14] H. M. Markowitz. *Portfolio Selection*. The Journal of Finance 7 1: 77-91, March 1952.
- [15] S. R. Mohan. *Degeneracy in linear complementarity problems: a survey*. Annals of Operations Research, Volume 46-47, Number 1: 179-194, March, 1993.
- [16] <http://neos.mcs.anl.gov/CaseStudies/port/index.html>
- [17] Rapcsák Tamás. *Nemlineáris optimalizálás*. Budapesti Közgazdaságtudományi és Államigazgatási Egyetem, Aula kiadó, Budapest, 2006.
- [18] C. Roos, T. Terlaky, and J.-Ph Vial. *Theory and Algorithms for Linear Optimization, An Interior Point Approach*. Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley and Sons, New York, 1997. Second edition: *Interior Point Methods for Linear Optimization*. Springer, New York, 2006.
- [19] A. R. Conn, N. I. M. Gould, P. L. Toint. *Trust-region methods*. Society for Industrial and Applied Mathematics, Cambridge University Press, Part 5: 75-82, 1987.