Kvadratično i sekvencijalno kvadratično programiranje

Marko Đurasević

Sadržaj

1 Kvadratično programiranje

Optimizacijski problem kod kojeg je funkcija cilja kvadratna i ograničenja su linearna, nazivamo kvadratičnim programom [5]. Kvadratično programiranje može se općenito zapisati kao:

$$\min_{x} \frac{1}{2}x^T Gx + x^T d + c \tag{1.1a}$$

$$uz\ ograničenja : a_i^T x = b_i,\ i \in \mathcal{I} \tag{1.1b}$$

$$a_i^T x \geq b_i,\ i \in \mathcal{E}. \tag{1.1c}$$

Navedeni izraz često zapisujemo u kompaktnijem obliku kao:

$$\min_{x} f(x) \tag{1.2a}$$

$$uz\ ograničenja : h_i(x) = 0,\ i \in \mathcal{I} \tag{1.2b}$$

$$g_i(x) \geq 0,\ i \in \mathcal{E}. \tag{1.2c}$$

Iz izraza se može vidjeti kako je cilj kvadratičnog programiranja pronaći točku x koja minimizira ciljnu funkciju $f(x)$ i ujedno zadovoljava sva ograničenja $g(x)$ i $h(x)$. Sve točke koje zadovoljavaju sva ograničenja (no nisu nužno optimumi) nazivamo **regularnim točkama**. U ovisnosti o zadanoj funkciji ovisit će hoće li zadani problem biti jednostavan za riješiti ili ne. Ako je funkcija $f(x)$ konveksna onda problem kvadratičnog programiranja ima jedno jedinstveno rješenje i često je lako rješivo [5]. Za funkciju $f(x)$ kažemo da je konveksna ako vrijedi da je njezina Hesseova matrica pozitivno semidefinitna, odnosno da vrijedi:

$$x^T Gx \geq 0, \tag{1.3}$$

za svaki x koji nije nul-vektor. S druge strane ako funkcija $f(x)$ nije konveksna (odnosno njezina Hesseova matrica nije pozitivno semidefinitna) onda je pronalazak minimuma puno teži jer funkcija može imati nekoliko lokalnih minimuma i stacionarnih točaka [5].

U nastavku poglavlja ograničit ćemo se isključivo na problem kada je funkcija $f(x)$ konveksna te ćemo se upoznati s metodom koja služi za rješavanje kvadratičnog programa. No najprije ćemo pogledati kako se može riješiti kvadratično programiranje koje se sastoji od isključivo ograničenja jednakosti.

1.1 Kvadratično programiranje s ograničenjima jednakosti

U ovom poglavlju pozabavit ćemo se problemom kvadratičnog programiranja koje se sastoji samo od ograničenja jednakosti. Ovaj problem nam je veoma važan jer će se metoda za rješavanje općenitog problema kvadratičnog programiranja (s ograničenjima jednakosti i nejednakosti) u svakoj iteraciji oslanjati upravo na rješavanje ovog jednostavnijeg problema. Kvadratični program koji se sastoji samo od ograničenja jednakosti zadan je kao:

$$\min_x \frac{1}{2}x^T Gx + x^T d + c \tag{1.4a}$$

$$uz\ ograničenja : Ax = b. \tag{1.4b}$$

Želimo pronaći točku x^* koja minimizira zadani kvadratični program. Navedeni problem možemo riješiti korištenjem Lagrangeovih multiplikatora.

Postupak Lagrangeovih multiplikatora zahtijeva da uvedemo novu funkciju koju nazivamo Lagrangeovom funkcijom i definiramo ju kao:

$$L(x, \lambda) = f(x) - \lambda^T h(x) = \frac{1}{2}x^T Gx + x^T d + c - \lambda^T (Ax - b) \tag{1.5}$$

Ako je x^* točka optimuma funkcije 1.4a uz zadana ograničenja 1.4b onda vrijedi:

$$\left(\frac{\partial L}{\partial x}\right)(x^*, \lambda^*) = (\nabla f)(x^*) - (\nabla h)(x^*)\lambda^* = Gx^* + d - A\lambda^* = 0 \tag{1.6a}$$

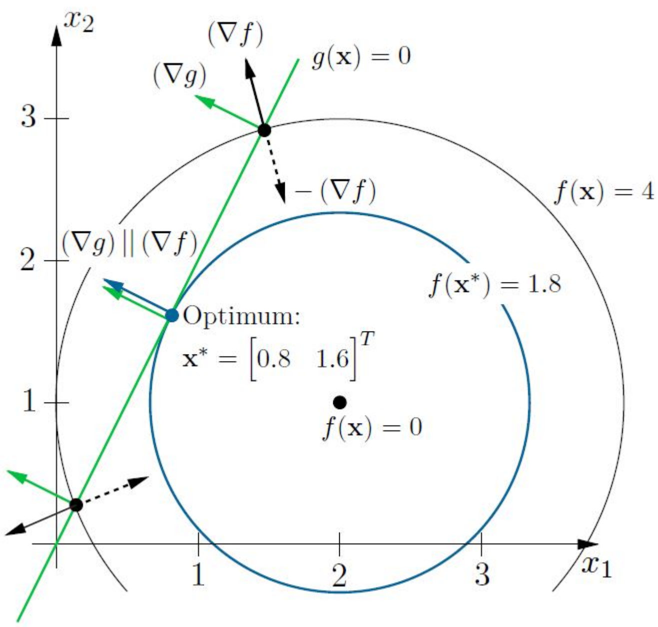
$$uz\ ograničenja : h(x^*) = Ax^* - b = 0. \tag{1.6b}$$

Izraz 1.6a zapravo označava da se u točki minimuma gradijent ciljne funkcije koja se minimizira mora moći izraziti kao linearna kombinacija gradijenata ograničenja. Brojeve λ nazivamo Lagrangeovim multiplikatorima i oni skaliraju iznose gradijenata ograničenja tako da se ponište s iznosom gradijenta ciljne funkcije. S druge strane, izraz 1.6b zahtijeva da točka x^* zadovoljava postavljena ograničenja. Zanimljivo je još pogledati koje značenje imaju vrijednosti Lagrangeovih multiplikatora. Naime, vrijednost Lagrangeovih multiplikatora nam govori koliko će se iznos ciljne funkcije promijeniti ako malo perturbiramo ograničenje. Pogledajmo sada primjer 1 koji ilustrira odnose među gradijentima ciljne funkcije i ograničenja u raznim točkama.

Primjer 1. Neka je zadan optimizacijski problem [3]:

$$\min_x \ f(x) = (x_1 - 2)^2 + (x_2 - 1)^2 \tag{1.7a}$$

$$uz\ ograničenja : g(x) = x_2 - 2x_1 = 0. \tag{1.7b}$$



Slika 1.1: Primjer kvadratičnog programa [3]

Slika 1.1 nam prikazuje skicu zadanog optimizacijskog problema. Na slici su prikazane konture ciljne funkcije i zadano ograničenje (označeno zelenom linijom). Ako pogledamo točku optimuma $x^* = (0.8, 1.6)$ onda možemo vidjeti da je gradijent ciljne funkcije (∇f) paralelan s gradijentom ograničenja (∇g).

Izračunamo li gradijent od ciljne funkcije dobit ćemo:

$$(\nabla f)(x) = \begin{bmatrix} 2(x_1 - 2) \\ 2(x_2 - 1) \end{bmatrix}. \tag{1.8a}$$

Uvrstimo li točku optimuma u izraz 1.8a dobivamo:

$$(\nabla f)(x) = \begin{bmatrix} -2.4 \\ 1.2 \end{bmatrix}. \tag{1.9a}$$

Izračunamo li gradijent ograničenja (kako je ograničenje linearno, gradijent će biti konstantan)

$$(\nabla g)(x) = \begin{bmatrix} -2 \\ 1 \end{bmatrix}, \tag{1.10a}$$

možemo vidjeti kako vrijedi jednakost $(\nabla f)(x) = 1.2(\nabla g)(x)$, pri čemu je 1.2 upravo iznos našeg Lagrangeovog multiplikatora za ovaj problem.

Pogledamo li druge dvije označene točke, možemo vidjeti kako u njima gradijenti ciljne funkcije i ograničenja nisu paralelni, iz čega proizlazi da navedene točke ne mogu biti minimumi kvadratičnog problema, iz razloga što se gradijent ciljne funkcije ne može prikazati kao linearna kombinacija gradijenata ograničenja (kako imamo samo jedno ograničenje onda gradijenti obavezno moraju biti međusobno paralelni).

Pogledajmo sada kako možemo odrediti rješenje ovako postavljenog kvadratičnog programa. Ako malo bolje promotrimo uvjete 1.6a i 1.6b možemo uočiti kako oni zapravo čine sustav linearnih jednadžbi. Rješavanjem navedenog sustava možemo dobiti rješenje našeg kvadratičnog programa. Navedeni sustav možemo zapisati u matričnom obliku kao:

$$\begin{bmatrix} G & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \lambda^* \end{bmatrix} = \begin{bmatrix} -d \\ b \end{bmatrix}. \tag{1.11}$$

Rješavanjem tog sustava dobit ćemo točku optimuma x^* i pripadne Lagrangeove multiplikatore λ^*. Navedeni sustav možemo riješiti nekom od metoda za rješavanje linearnih sustava (primjerice LUP dekompozicijom [4][8]). Ponekad će nam više odgovarati da izraz 1.11 zapišemo na malo drugačiji način kako bi umjesto optimalnog rješenja dobili pomak koji je potrebno napraviti od početne točke do točke koja predstavlja optimalno rješenje (kasnije će se to pokazati korisnim u metodi aktivnih ograničenja). Iz tog razloga izraz 1.11 možemo zapisati kao:

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -p \\ \lambda^* \end{bmatrix} = \begin{bmatrix} g \\ c \end{bmatrix}, \tag{1.12}$$

gdje su

$$c = Ax - b, \quad g = d + Gx, \quad p = x^* - x. \tag{1.13}$$

Pogledajmo sada kratko jedan primjer u kojemu ćemo pokazati kako korištenjem prethodnih zaključaka možemo izravno izračunati rješenje kvadratičnog programa s ograničenjima jednakosti.

Primjer 2. Neka je zadan optimizacijski problem:

$$\min_x \; f(x) = (x_1 - 2)^2 + (x_2 - 1)^2 \tag{1.14a}$$

$$uz\ ograničenja : h(x) = x_2 - 2x_1 = 0. \tag{1.14b}$$

Iz tog optimizacijskog problema lako možemo odrediti potrebne elemente koje je potrebno uvrstiti u (1.11):

$$G = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \quad A = \begin{bmatrix} -2 & 1 \end{bmatrix}, \quad d = \begin{bmatrix} -4 \\ -2 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \end{bmatrix}. \tag{1.15}$$

Uvršavanjem u 1.11 dobivamo

$$\begin{bmatrix} 2 & 0 & 2 \\ 0 & 2 & -1 \\ -2 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1^* \\ x_2^* \\ \lambda^* \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \\ 0 \end{bmatrix}. \tag{1.16}$$

Navedeni sustav sada možemo riješiti primjerice LUP dekompozicijom i dobivamo rješenje:

$$\begin{bmatrix} x_1^* \\ x_2^* \\ \lambda^* \end{bmatrix} = \begin{bmatrix} 0.8 \\ 1.6 \\ 1.2 \end{bmatrix}. \tag{1.17}$$

Prethodni primjer ćemo iskoristiti kako bismo ilustrirali značenje iznosa Lagrangeovih multiplikatora. Pogledajmo primjer 3.

Primjer 3. Neka je zadan optimizacijski problem:

$$\min_x \ f(x) = (x_1 - 2)^2 + (x_2 - 1)^2 \tag{1.18a}$$

$$uz\ ograničenja : g(x) = x_2 - 2x_1 = 0. \tag{1.18b}$$

Rješenje zadanog optimizacijskog problema smo dobili iz prošlog primjera i ono iznosi $x = (0.8, 1.6)$ i vrijednost ciljne funkcije u toj točki je $f(x) = 1.8$. Pripadni Lagrangeov multiplikator je $\lambda = 1.2$. Ako malo perturbiramo ograničenje tako da ono postane

$$g(x) = x_2 - 2x_1 = 0.1 \tag{1.19}$$

i riješimo kvadratični program s tim ograničenjem kao sustav

$$\begin{bmatrix} 2 & 0 & 2 \\ 0 & 2 & -1 \\ -2 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1^* \\ x_2^* \\ \lambda^* \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \\ 0.1 \end{bmatrix}. \tag{1.20}$$

dobivamo rješenje $x' = (0.76, 1.62)$ i Lagrangeov multiplikator $\lambda' = 1.24$. Vrijednost ciljne funkcije u ovoj točki je $f(x') = 1.922$.

Možemo vidjeti kako vrijedi $f(x') \approx f(x) + \Delta b * \lambda$ gdje je $\Delta b = 0.1 - 0$. Izračunamo li desnu stranu dobit ćemo $f(x) + \Delta b * \lambda = 1.92$, pa možemo vidjeti da smo dobili dosta dobru aproksimaciju promjene ciljne funkcije u ovisnosti o promjeni ograničenja. No bitno je za naglasiti da ta promjena mora biti mala (Lagrangeov multiplikator dobiven za ovako perturbirani sustav mora se malo razlikovati od originalnog Lagrangeovog multiplikatora), inače aproksimacija neće biti dobra. Iz primjera možemo vidjeti kako nam iznos Lagrangeovog multiplikatora govori kako će se iznos funkcije mijenjati s obzirom na promjenu ograničenja. Što je Lagrangeov multiplikator veći, to će i sama promjena biti značajnija.

Ovaj rezultat bit će nam važan i u idućem potpoglavlju gdje ćemo ukloniti ograničenja s najnegativnijim Lagrangeovim multiplikatorom. Tu odluku temeljimo upravo na prethodnoj analizi, jer ćemo se nadati da uklanjanjem ograničenja s najnegativnijem Lagrangeovim multiplikatorom možemo postići najveće smanjenje iznosa funkcije cilja.

1.2 Kvadratično programiranje s ograničenjima jednakosti i nejednakosti

U ovom potpoglavlju osvrnut ćemo se na kvadratične programe koji se sastoje od ograničenja jednakosti i nejednakosti. Kod ovog problema tražit ćemo točku koja zadovoljava sva ograničenja, bilo da su to ograničenja jednakosti ili nejednakosti.

U svakoj regularnoj točki, ograničenja nejednakosti mogu biti aktivna ili neaktivna (no u oba slučaja ona moraju biti zadovoljena). Ako neka točka leži na rubu ograničenja nejednakosti, onda kažemo da je to ograničenje aktivno. S druge strane ako se točka nalazi unutar ograničenja onda kažemo da to ograničenje nije aktivno za tu točku. Kod kvadratičnog programiranja s ograničenjima nejednakosti veoma nam je bitno pronaći skup aktivnih ograničenja u točki optimuma.

Definirajmo najprije nužne uvjete da neka točka predstavlja lokalni minimum kvadratičnog programa. Ti uvjeti se nazivaju Karush-Kuhn-Tuckerovi uvjeti optimalnosti.

Teorem 1. Karush-Kuhn-Tuckerovi (KKT) uvjeti optimalnosti. Ako je točka x^* lokalni minimum kvadratičnog problema

$$\min_x f(x) \tag{1.21a}$$

$$uz\ ograničenja : h_i(x) = 0,\ i = 1, \ldots, p \tag{1.21b}$$

$$g_i(x) \geq 0,\ i = 1, \ldots, q, \tag{1.21c}$$

tada postoje jedinstveni Lagrangeovi multiplikatori $\lambda^* \in \mathbb{R}^p$ i $\mu^* \in \mathbb{R}^q$ takvi da zadovoljavaju ograničenja:

$$(\nabla f)(x^*) - (\nabla h)(x^*)\lambda^* - (\nabla g)(x^*)\mu^* = 0 \tag{1.22a}$$

$$h(x^*) = 0 \tag{1.22b}$$

$$g(x^*) \geq 0 \tag{1.22c}$$

$$\mu^* \geq 0 \tag{1.22d}$$

$$g^T(x^*)\mu^* = 0 \tag{1.22e}$$

Pogledamo li izraze iz teorema 1 možemo vidjeti kako je opet cilj zapisati gradijent ciljne funkcije kao linearnu kombinaciju gradijenata ograničenja. U ovom slučaju uvodimo dodatne Lagrangeove multiplikatore za ograničenja nejednakosti, koje označavamo s μ. Za razliku od Lagrangeovih multiplikatora za ograničenja jednakosti, Lagrangeovi multiplikatori za ograničenja nejednakosti moraju obavezno biti pozitivni (što vidimo iz izraza 1.22d). Izraz 1.22c zahtijeva da sva ograničenja nejednakosti budu ispunjena, dok izraz 1.22b zahtijeva da sva ograničenja jednakosti budu ispunjena. Izraz 1.22e naziva se komplementarnim ograničenjem (engl. *complementary slackness*) i na prvu ruku možda nije intuitivan, no ovaj izraz zapravo govori da ako je neko ograničenje nejednakosti aktivno, da je onda pripadni Lagrangeov multiplikator veći od nula i obrnuto, ako je neko od ograničenja nejednakosti neaktivno, da je onda pripadni Lagrangeov multiplikator jednak nuli.

Potrebno je naglasiti da Karush-Kuhn-Tuckerovi uvjeti predstavljaju nužne, ali ne i dovoljne uvjete da točka x^* predstavlja minimum optimizacijskog problema. No ako je ciljna funkcija konveksna i ako su ograničenja linearna, onda KKT uvjeti predstavljaju i dovoljne uvjete.

Sada se postavlja pitanje kako možemo riješiti kvadratični program s ograničenjima jednakosti i nejednakosti. Ideja je zapravo pronaći skup aktivnih ograničenja nejednakosti u točki optimuma i onda ta ograničenja tretirati kao ograničenja jednakosti (neaktivna ograničenja nejednakosti se ne uzimaju u obzir) i riješiti pripadajući problem kvadratičnog programiranja (koje se sada sastoji samo od ograničenja jednakosti) kao što je prikazano u primjeru 4. Postavlja se pitanje kako možemo znati koja su ograničenja aktivna u točki optimuma? Nažalost, to ne možemo unaprijed znati, tako da nam je potreban postupak kojim ćemo na neki način pronaći aktivna ograničenja. Najjednostavnija mogućnost jest da se isprobaju sve kombinacije ograničenja i da se riješe pripadni kvadratični programi. Nažalost, ako je broj ograničenja velik, jednostavno nije izvedivo da se isprobaju sve kombinacije u nekom razumnom vremenu. Iz tog razloga potrebni su postupci koji neće isprobati sve kombinacije nego će na "pametan" način odredi skup aktivnih ograničenja. U idućem potpoglavlju bit će prikazana jedna metoda koja se koristi upravo za rješavanje kvadratičnog programa s ograničenjima jednakosti i nejednakosti. Ta metoda nipošto nije jedina metoda, već postoji široki spektar metoda od kojih svaka ima svoje prednosti i nedostatke. Neke druge metode koje se mogu primijeniti za rješavanje kvadratičnog programa su primjerice metoda projekcije gradijenta (engl. *gradient-projection method*) i metoda unutarnje točke (engl. *interior-point method*) [5][3].

Primjer 4. Neka je zadan optimizacijski problem:

$$\min_x \ f(x) = \frac{1}{2}(x_1^2 + x_2^2 + x_3^2) \tag{1.23a}$$

$$uz\ ograničenja : g_1(x) = x_1 + x_2 + x_3 - 3 \leq 0 \tag{1.23b}$$

$$g_2(x) = x_1 \leq 0 \tag{1.23c}$$

Potrebno je odrediti minimum zadanog optimizacijskog problema.

Kako ne znamo aktivni skup ograničenja u točki optimuma x^*, ne preostaje nam ništa drugo nego isprobati sve kombinacije i vidjeti za koju točku su KKT uvjeti zadovoljeni. Srećom, kako imamo samo dva ograničenja nejednakosti, ukupni broj mogućnosti je koji moramo isprobati je četiri.

Izračunajmo prvo pripadne matrice i vektore kvadratičnog programa.

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad d = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{1.24a}$$

$$\nabla g_1 = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} \quad b_1 = 3 \tag{1.24b}$$

$$\nabla g_2 = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \quad b_2 = 0 \tag{1.24c}$$

Prva mogućnost koju moramo isprobati jest da niti jedno ograničenje nije aktivno. Tada moramo riješiti sustav

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{1.25}$$

Očito da je rješenje ovog sustava točka $x = (0,0,0)$. No odmah možemo uočiti da točka ne zadovoljava ograničenje g_1 i zaključujemo da ona ne može biti točka optimuma.

Druga mogućnost je da je aktivno ograničenje g_1. Tada će sustav koji trebamo riješiti biti jednak

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ -1 & -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 3 \end{bmatrix} \tag{1.26}$$

Rješavanjem ovog sustava dobit ćemo rješenje $x = (-1,-1,-1)$i Lagrangeov multiplikator $\lambda = 1$. Vidimo kako ova točka zadovoljava sva ograničenja i sve KKT uvjete te ona u ovom slučaju ujedno i točka optimuma. No bez obzira na to ispitat ćemo i ostala dva slučaja.

Treći slučaj je ograničenje g_2 aktivno. Tada dobivamo sustav

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{1.27}$$

Rješavanjem ovog sustava dobivamo opet rješenje $x = (0,0,0)$ i Lagrangeov multiplikator $\lambda = 0$. Već smo vidjeli da ova točka ne zadovoljava ograničenje g_1 te ju odbacujemo.

Zadnji slučaj koji moramo razmotriti jest da su oba ograničenja aktivna. Tada dobivamo sustav

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ -1 & -1 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 3 \\ 0 \end{bmatrix} \tag{1.28}$$

Rješavanjem ovog sustava dobivamo rješenje $x = (0,-1.5,-1.5)$ i Lagrangeove multiplikatore $\lambda = (1.5,-1.5)$. Očito je da točka zadovoljava sva ograničenja, no vidimo da ne zadovoljava sve KKT uvjete. Naime Lagrangeov multiplikator ograničenja g_2 je negativan, što po uvjetima ne bi smio biti i iz tog razloga odbacujemo i ovu točku.

Konačno zaključujemo da je točka $x = (-1,-1,-1)$ točka optimuma ovog kvadratičnog programa.

1.3 Metoda aktivnih ograničenja

U ovom poglavlju pozabavit ćemo se metodom za pronalaženje rješenja kvadratičnog programa. Ako je funkcija cilja konveksna funkcija, ovaj postupak pronaći će rješenje koje je ujedno i globalni minimum kvadratičnog programa. U ovom potpoglavlju ograničit ćemo se isključivo na razmatranje slučaja kada je ciljna funkcija konveksna, odnosno kada je njezina Hesseova matrica pozitivno semidefinitna. Metoda aktivnih ograničenja pokazala se je najefikasnijom za rješavanje manjih i srednje velikih problema [5].

Kao što je u prošlom potpoglavlju istaknuto, kada bismo znali skup aktivnih ograničenja u točki optimuma x^* mogli bismo pronaći rješenje problema na način kako smo to napravili za kvadratični program s ograničenjima jednakosti. Naravno, najčešće nemamo nikakve informacije o skupu aktivnih ograničenja u točki minimuma i upravo određivanje tog aktivnog skupa će biti glavni izazov.

Metoda aktivnih ograničenja obično započinje s nekom regularnom točkom i osigurava da su sve točke u sljedećim iteracijama regularne. Ova metoda pronalazi pomak iz trenutne točke u sljedeću točku na način da riješi kvadratični program koji se sastoji od svih ograničenja jednakosti i nekih ograničenja nejednakosti koja se tretiraju kao ograničenja jednakosti (odnosno, kažemo da su ta ograničenja aktivna u toj točki). Navedeni podskup naziva se radnim skupom i označavaju sa W_k gdje k predstavlja trenutnu iteraciju. Bitno je uočiti da se radni skup uvijek sastoji od svih ograničenja jednakosti, a ograničenja nejednakosti se tijekom postupka aktivnih ograničenja dodaju, odnosno izbacuju iz tog skupa.

Za trenutnu točku x_k i pripadajući radni skup W_k prvo provjeravamo da li točka x_k minimizira pripadni kvadratični program. Ako ga ne minimizira, onda računamo pomak p_k pomoću podproblema kvadratičnog programiranja

$$\min_x \frac{1}{2}x^T G x + x^T d + c \tag{1.29a}$$

$$uz\ ograničenja : a_i^T x = b_i,\ i \in \mathcal{W} \tag{1.29b}$$

$$\tag{1.29c}$$

koristeći izraz 1.12. Ako je pomak različit od nula, potrebno je odrediti koliko ćemo se pomaknuti u tom smjeru. Vrijedi li da su za $x_k + p_k$ zadovoljena sva ograničenja nejednakosti, onda novu točku postavljamo na $x_{k+1} = x_k + p_k$. U suprotnom, ako bi pomak za iznos p_k narušio neka od ograničenja nejednakosti, onda je potrebno korak prilagoditi tako da sva ograničenja budu zadovoljena. Novu točku ćemo onda izračunati pomoću

$$x_{k+1} = x_x + \alpha_k p_k, \tag{1.30}$$

gdje je α_k najveća duljina koraka iz intervala $[0, 1]$ takva da x_{k+1} zadovoljava sva ograničenja. Cilj nam je na temelju svih zadanih ograničenja odrediti koliku duljinu koraka smijemo napraviti. Olakotna okolnost je ta što ne moramo ispitivati ograničenja iz trenutnog radnog skupa W_k, jer će ta ograničenja uvijek biti zadovoljena bez obzira na veličinu pomaka. Dakle moramo ispitati samo ona ograničenja koja nisu u tom skupu. Označimo naša ograničenja nejednakosti kao $a_i^T x \geq b_i$. Ako vrijedi $a_i^T p_k \geq 0$ za neki $i \notin W_k$, onda će za svaki $\alpha_k \geq 0$ vrijediti da će ograničenje biti zadovoljeno, odnosno $a_i^T(x_k + \alpha_k p_k) \geq a_i^T x_k \geq b_i$. S druge strane ako je $a_i^T p_k \leq 0$ za neki $i \notin W_k$, onda će $a_i^T(x_k + \alpha_k p_k) \geq b_i$ vrijedi samo za α_k takav da

$$\alpha_k \leq \frac{b_i - a_i^T x_k}{a_i^T p_k}. \tag{1.31}$$

Kako želimo da nam duljina koraka buda najveća moguća iz intervala $[0, 1]$ i da nova točka ujedno zadovoljava sva ograničenja, onda ćemo duljinu koraka izračunati preko izraza:

$$\alpha_k = min(1, \min_{i \notin W_k, a_i^T p_k < 0} \frac{b_i - a_i^T x_k}{a_i^T p_k}). \tag{1.32}$$

Ograničenja za koja je ostvaren minimum u izrazu 1.32 nazivamo **blokirajućim ograničenjima** (engl. *blocking constraints*). Treba uočiti da je lako moguće da α_k bude jednak nuli ako postoji ograničenje koje je aktivno u točki x_k ali nije u trenutnom radnom skupu W_k. Ako je $\alpha_k < 1$, odnosno, ako smo prilikom pomicanja u smjeru p_k naišli na ograničenja koja nisu u trenutnom radnom skupu W_k, onda konstruiramo novi radni skup tako da jedno od tih blokirajućih ograničenja dodamo u trenutni radni skup W_k.

Postupak nastavlja iterirati, dodajući ograničenja u radni skup sve dok ne dođe do točke $\hat{x}$ koja minimizira ciljnu funkciju s obzirom na ograničenja koja se nalaze u trenutnom radnom skupu. Ovaj slučaj se može prepoznati po tome što će iznos pomaka p u toj točci biti jednak nuli.

Rješavanjem sustava s trenutnim radnim skupom dobit ćemo iznos točke $\hat{x}$, kao i vrijednosti Lagrangeovih multiplikatora za ograničenja koje se nalaze u trenutnom radnom skupu (Lagrangeovi multiplikatori za ograničenja koja se ne nalaze u trenutnom radnom skupu biti će jednaki nula). Ograničenjem duljine koraka kod pomaka u novu točku osigurali smo da će u novoj točki sva ograničenja biti zadovoljena te jedino još moramo provjeriti da li je zadovoljen uvjet da su svi Lagrangeovi multiplikatori (od ograničenja nejednakosti) veći ili jednaki nula. Ako vrijedi da su svi Lagrangeovi multiplikatori (od ograničenja nejednakosti) veći ili jednaki nula onda točka $\hat{x}$ zadovoljava sve KKT uvjete i zaključujemo da ona predstavlja minimum kvadratičnog programa.

S druge strane ako je neki od Lagrangeovih multiplikatora negativan, to označava da pripadno ograničenje možemo odbaciti i na taj način dodatno smanjiti vrijednost ciljne funkcije. Nakon što odbacimo to ograničenje iz radnog skupa ograničenja, riješimo kvadratični program s novim radnim skupom. Postupak nastavlja iterirati dok ne pronađe točku koja zadovoljava sve KKT uvjete.

Potrebno je još spomenuti da postoji mogućnost da nekoliko Lagrangeovih multiplikatora bude manje od nula. Tada se obično iz radnog skupa izbacuje ono ograničenje koje ima najnegativniji Lagrangeov multiplikator. Ova odluka je temeljena na analizi koja je provedena u primjeru 3. Naime, da ponovimo, odbacivanjem ograničenja koje ima najnegativniji Lagrangeov multiplikator želimo postići najveće smanjenje u iznosu ciljne funkcije. Također moguće je u jednoj iteraciji odstraniti i nekoliko ograničenja s negativnim Lagrangeovim multiplikatorima, no u nastavku nećemo posvetiti posebnu pažnju ovakvom postupku.

Pogledajmo sada jedan primjer kako metodu aktivnih ograničenja možemo iskoristiti za rješavanje kvadratičnog programa.

Primjer 5. Neka je zadan optimizacijski problem [5]:

$$\min_x \; f(x) = (x_1 - 1)^2 + (x_2 - 2.5)^2 \tag{1.33a}$$

$$uz\ ograničenja: x_1 - 2x_2 + 2 \geq 0 \quad (1) \tag{1.33b}$$

$$-x_1 - 2x_2 + 6 \geq 0 \quad (2) \tag{1.33c}$$

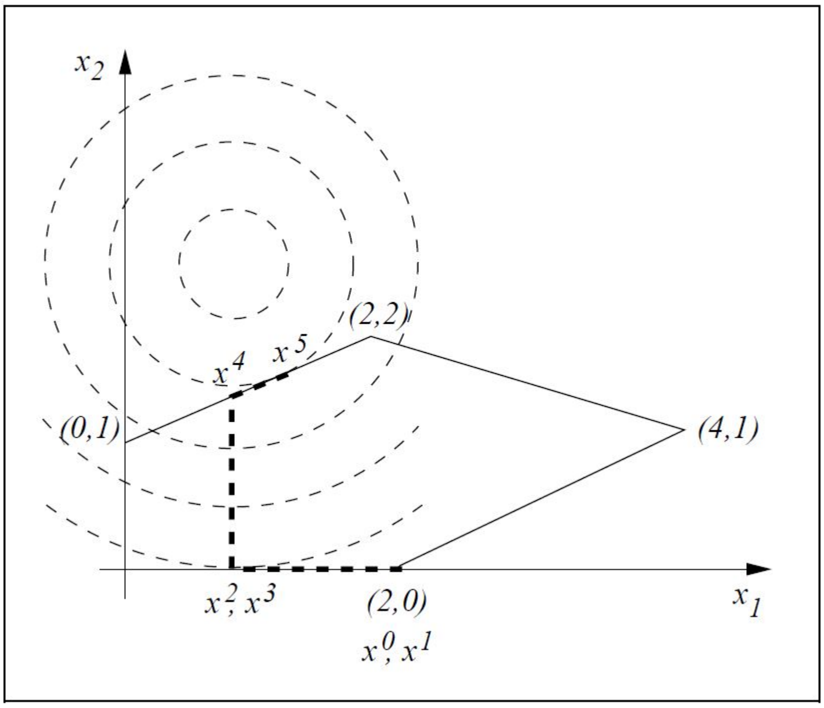
$$-x_1 + 2x_2 + 2 \geq 0 \quad (3) \tag{1.33d}$$

$$x_1 \geq 0 \quad (4) \tag{1.33e}$$

$$x_2 \geq 0 \quad (5) \tag{1.33f}$$

s početnom točkom $x_0 = (2, 0)$. Zbog jednostavnosti primjera, ograničenja su označena indeksima od jedan do pet.

Slika 1.2 prikazuje pojedine korake metode aktivnih ograničenja na gore spomenuti kvadratični problem.



Slika 1.2: Koraci metode aktivnih ograničenja [5]

Za početnu točku smo odabrali $x_0 = (2, 0)$. Iz slike 1.2 možemo vidjeti kako su u toj točki aktivna ograničenja s indeksima 3 i 5. Iz tog razloga početni radni skup ograničenja postavit ćemo na $W_0 = \{3, 5\}$. Treba naglasiti da smo početni radni skup jednako tako mogli postaviti na $W_0 = \{3\}$, $W_0 = \{5\}$ ili $W_0 = \emptyset$, no da ćemo onda dobiti ponešto drugačije ponašanje algoritma.

Odredimo najprije potrebne matrice za rješavanje kvadratičnog problema prema izrazu 1.12. Matrica G i vektor d će uvijek biti jednaki i iznose

$$G = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad d = \begin{bmatrix} -2 \\ -5 \end{bmatrix} \tag{1.34}$$

Matrice koje dobijemo za ograničenja će se mijenjati iz iteracije u iteraciju pa će tako za trenutni radni skup iznositi:

$$A_0 = \begin{bmatrix} -1 & 2 \\ 0 & 1 \end{bmatrix} \quad b_0 = \begin{bmatrix} -2 \\ 0 \end{bmatrix} \tag{1.35}$$

Iz njih izračunamo pomoćne vektore c i g koristeći izraz 1.13

$$c_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad g_0 = \begin{bmatrix} 2 \\ -5 \end{bmatrix} \tag{1.36}$$

Korištenjem tih matrica, možemo izgraditi sustav (korištenjem izraza 1.12) čijim rješavanjem možemo dobiti traženi pomak i pripadne Lagrangeove multiplikatore.

$$\begin{bmatrix} 2 & 0 & -1 & 0 \\ 0 & 2 & 2 & 1 \\ -1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 2 \\ -5 \\ 0 \\ 0 \end{bmatrix} \tag{1.37}$$

Rješavanjem ovog sustava dobit ćemo iznos pomaka $p_0 = (0, 0)$ i Lagrangeove multiplikatore $\lambda_0 = (-2, -1)$. Kao što možemo vidjeti pomak je jednak 0 (dakle točka x_1 je jednaka točki x_0), što znači da s trenutnim radnim skupom ne možemo dalje minimizirati kvadratični program. Ako pogledamo sliku 1.2 možemo i vidjeti zašto je pomak jednak 0. Naime iz trenutne točke nije se moguće pomaknuti u niti jednu drugu točku a da oba ograničenja ostanu aktivna. Također možemo uočiti da su predznaci Lagrangeovih multiplikatora negativni, što znači da kvadratični program možemo dalje minimizirati ako izbacimo neko ograničenje iz trenutnog radnog skupa. Izbacujemo ograničenje s najnegativnijim Lagrangeovim multiplikatorom, dakle treće ograničenje te je novi radni skup jednak $W_1 = \{5\}$. Sada rješavamo kvadratični program s novim radnim skupom

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \lambda_1 \end{bmatrix} = \begin{bmatrix} 2 \\ -5 \\ 0 \end{bmatrix} \tag{1.38}$$

Rješavanjem ovog sustava dobit ćemo iznos pomaka $p_1 = (-1, 0)$ i Lagrangeov multiplikator $\lambda_1 = (-5)$. Kao što možemo vidjeti iznos pomaka sada nije jednak 0 što znači da pomicanjem u zadanom smjeru možemo dodatno smanjiti vrijednost funkcije cilja. Naravno, prije nego što se pomaknemo u taj smjer potrebno je provjeriti hoće li neko od ograničenja biti narušeno. U ovom slučaju to se neće desiti, pa se možemo pomaknuti u novu točku $x_2 = (1, 0)$. U toj točki sada ponovo rješavamo kvadratični problem (skup aktivnih ograničenja ostaje isti $W_2 = W_1 = \{5\}$)

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \lambda_1 \end{bmatrix} = \begin{bmatrix} 0 \\ -5 \\ 0 \end{bmatrix} \tag{1.39}$$

te dobivamo pomak $p_2 = (0, 0)$ i Lagrangeov multiplikator $\lambda_2 = (-5)$. Možemo vidjeti kako je pomak opet jednak nula (dakle vrijedi $x_3 = x_2$) te da je Lagrangeov multiplikator negativan, što znači da odbacivanjem petog ograničenja možemo dodatno smanjiti iznos ciljne funkcije. Novi radni skup je onda $W_3 = \emptyset$ te rješavamo kvadratični problem koji nema ograničenja

$$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -5 \end{bmatrix}. \tag{1.40}$$

Rješavanjem ovog sustava dobit ćemo pomak $p_3 = (0, 2.5)$. Nažalost ne možemo se pomaknuti za cijeli iznos pomaka jer bi time narušili jedno od ograničenja (točnije prvo ograničenje). To se veoma dobro može vidjeti iz slike 1.2. Kada bi se pomaknuli za cijeli iznos pomaka došli bi do pravog minimuma ciljne funkcije. Nažalost, navedenim minimum ne zadovoljava prvo ograničenje i iz tog razloga ta točka ne predstavlja i minimum kvadratičnog programa. Iz tog razloga moram izračunati maksimalnu dopuštenu duljinu koraka koja osigurava da će to ograničenje biti zadovoljeno. Uvrštavanjem u izraz 1.31 možemo izračunati kolika je dozvoljena maksimalna duljina koraka (kako je samo prvo ograničenje blokirajuće, duljinu koraka će se izračunati samo za to ograničenje, dok ćemo račun ostalih preskočiti)

$$\alpha_3 \leq \frac{-2 - \begin{bmatrix} 1 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}}{\begin{bmatrix} 1 & -2 \end{bmatrix} \begin{bmatrix} 0 \\ 2.5 \end{bmatrix}} \leq 0.6 \tag{1.41}$$

Za iznos duljine koraka uzimamo najveću dozvoljenu vrijednost $\alpha_3 = 0.6$. Nova točka je onda jednaka $x_4 = \alpha_3 * p_3 + x_3 = (1, 1.5)$. Također, trenutni radni skup se proširuje s trenutnim blokirajućim ograničenjem, tako da je novi radni skup jednak $W_4 = \{1\}$. Rješavamo kvadratični program zadan novim radnim skupom

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \lambda_1 \end{bmatrix} = \begin{bmatrix} 0 \\ -2 \\ 0 \end{bmatrix}. \tag{1.42}$$

Rješavanjem ovog sustava dobit će se iznos pomaka $p_4 = (0.4, 0.2)$. Kako niti jedno ograničenje neće biti narušeno ovim pomakom, možemo se slobodno pomaknuti u novu točku $x_5 = (1.4, 1.7)$. Novi radni skup će biti jednak prethodnom, dakle $W_5 = \{1\}$ te se rješava kvadratični problem zadan tim skupom

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \lambda_1 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 1.6 \\ 0 \end{bmatrix}. \tag{1.43}$$

Dobiveni pomak je $p_5 = (0, 0)$ i Lagrangeov multiplikator $\lambda_5 = (0.8)$. Kako je pomak jednak nula i Lagrangeov multiplikator nije negativan, zaključujemo da smo pronašli optimalno rješenje našeg kvadratičnog problema $x^* = (1.4, 1.7)$.

Pogledajmo još ukratko što bi se desilo kada bi krenuli s nekim drugim od ranije spomenutih mogućih početnih radnih skupova ograničenja. Ako bi kao radni skup odabrali $W_0 = \{5\}$ onda bi se postupak odmah u prvoj iteraciji prebacio u točku $x_1 = (1, 0)$, bez potrebe da se prvo iz radnog skupa odstrani treće ograničenje. Ako bi početni radni skup bio $W_0 = \{3\}$, onda bi nova točka bila jednaka $x_1 = (2.2, 0.1)$ i možemo vidjeti kako bi se postupak krenuo gibati u suprotnom smjeru. Konačno, ako bi početni aktivni skup bio jednak $W_0 = \emptyset$, onda bi nova točka bila jednaka $x_1 = (\frac{4}{3}, \frac{5}{3})$ i novi radni skup bi bio $W_1 = \{1\}$. Možemo vidjeti kako bi u idućoj iteraciji dobili optimalno rješenje.

Kao što je spomenuto, u jednoj iteraciji potpuno je ispravno odstraniti i nekoliko ograničenja koja imaju negativne Lagrangeove multiplikatore. To znači da smo u ovom primjeru odmah u prvom koraku mogli iz radnog skupa odstraniti oba ograničenja i onda bi se postupak dalje ponašao kao u slučaju da je početni radni skup bio $W_0 = \emptyset$.

Iz prethodnog razmatranja vidimo da broj iteracija potrebnih da algoritam pronađe optimalno rješenje uvelike ovisi o odabranom početnom aktivnom skupu ograničenja.

Za kraj ovog poglavlja prikazat ćemo cijelu metodu aktivnih ograničenja u obliku pseudokoda prikazanom u algoritmu 1.

Algoritam 1 Metoda aktivnih ograničenja

Izračunati regularnu početnu točku x_0;
Postaviti W_0 na podskup aktivnih ograničenja u točki x_0;
while $k = 0, 1, 2, \dots$ **do**
    Odrediti p_k rješavanjem kvadratičnog programa 1.29 korištenjem izraza 1.12;
    if $p_k = 0$ **then**
        if $\lambda_i \geq 0$ za svaki $i \in W_k \cap \mathcal{I}$ **then**
            return rješenje $x^* = x_k$;
        else
            $j \leftarrow \mathrm{argmin}_{j \in W_k \cap \mathcal{I}} \lambda_j$;
            $x_{k+1} \leftarrow x_k$;
            $W_k \leftarrow W_k \setminus j$
        end if
    else
        Izračunaj α_k iz izraza 1.32
        $x_{k+1} \leftarrow x_k + \alpha_k p_k$;
        if postoje blokirajuća ograničenja **then**
            Odredi W_{k+1} tako da se u W_k doda jedno od blokirajućih ograničenja
        else
            $W_{k+1} = W_k$
        end if
    end if
end while

2 Sekvencijalno kvadratično programiranje

U prošlom poglavlju opisana je metoda kojom možemo riješiti kvadratični program gdje je ciljna funkcija kvadratna, a ograničenja su linearna. U ovom poglavlju opisat će se metoda koja može raditi s proizvoljno nelinearnim funkcijama cilja i s proizvoljno nelinearnim ograničenjima. Naravno, od takve metode ne treba očekivati da će ona biti u stanju uvijek pronaći globalni optimum optimizacijskog problema, jer će to uvelike ovisiti i o funkciji i o samim ograničenjima.

$$\min_x f(x) \tag{2.1a}$$
$$uz\ uvjete : h_i(x) = 0,\ i \in \mathcal{I} \tag{2.1b}$$
$$g_i(x) \geq 0,\ i \in \mathcal{E}. \tag{2.1c}$$

Metoda koja će biti opisana u ovom poglavlju naziva se sekvencijalno kvadratično programiranje (SQP - engl. *Sequential Quadratic programming*). Ideja ove metode je zapravo veoma jednostavna. Cilj je u svakoj iteraciji nelinearni optimizacijski problem aproksimirati kvadratičnim programom i riješiti ga primjerice metodom aktivnih ograničenja.

2.1 Definicija sekvencijalnog kvadratičnog programiranja

Kao što je već napomenuto, ideja sekvencijalnog kvadratičnog programiranja jest modelirati nelinearni problem s trenutnom točkom x_k kao kvadratični program koji se minimizira kako bi se pronašla točka x_{k+1} za sljedeću iteraciju.

Kako kvadratično programiranje radi isključivo s kvadratnim funkcijama cilja i linearnim ograničenjima, to znači da ćemo morati nelinearnu funkciju cilja aproksimirati kvadratnom funkcijom, a ograničenja linearnim funkcijama.

Krenimo prvo s aproksimacijom ograničenja pošto je ono jednostavnije. Ako je x_k trenutna točka našeg problema, onda možemo napraviti Taylorov razvoj oko te točke

$$g(x_k + p) = \nabla g(x_k)^T p + g(x_k). \tag{2.2}$$

Jednako tako mogli bismo htjeti aproksimirati i funkciju cilja pomoću Taylorovog reda, pa bismo dobili izraz

$$f(x_k + p) = f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T \nabla^2 f(x_k) p. \tag{2.3}$$

Nažalost pokazalo se je kako je ova aproksimacija može biti veoma loša u određenim problemima. Primjer 6 nam pokazuje što se dešava ako koristimo ovakvu aproksimaciju [1].

Primjer 6. Neka je zadan nelinearni optimizacijski problem

$$\min f(x) = (x_2 - 2)^2 - x_1^2 \tag{2.4a}$$

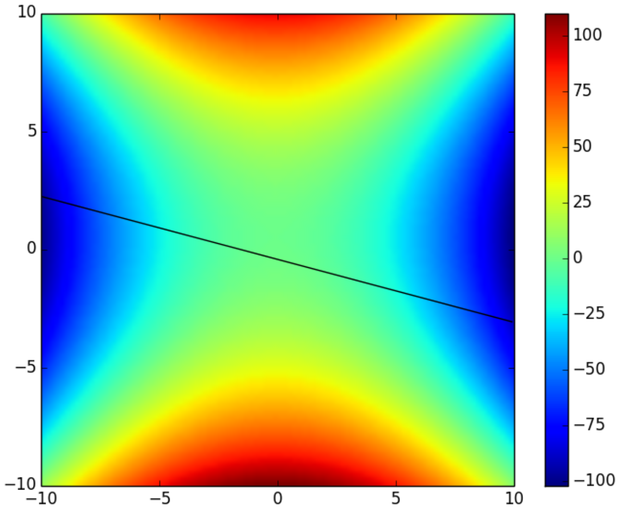
$$uz\ ograničenja : h(x) = 4x_1^2 + x_2^2 - 1 \tag{2.4b}$$

s početnom točkom $x_0 = (\frac{1}{10}, \frac{3}{2})$. Pretvorimo li ovaj nelinearni problem u kvadratični program korištenjem izraza 2.2 i 2.3 dobit ćemo

$$\min -p_1^2 + p_2^2 - \frac{1}{5} p_1 - p_2 + \frac{6}{25} \tag{2.5a}$$

$$uz\ ograničenja : \frac{4}{5} p_1 + 3p_2 + \frac{129}{100} = 0. \tag{2.5b}$$

Iz samih izraza 2.5a i 2.5b nije odmah uočljiv problem koji se je pojavio, no ako se kvadratični problem grafički prikaže lako je vidjeti što je pošlo po zlu. Slika 2.1 prikazuje konture ciljne funkcije i ograničenje (označeno crnom linijom). Sada se lako može vidjeti koji se je problem pojavio. Naime, kvadratični program nije ograničen i rješenje leži u beskonačnosti (iako originalni problem ima jedinstveno rješenje).



Slika 2.1: Konture kvadratičnog programa 2.5a

Problem koji se je pojavio u primjeru 6 je nastao zbog toga što se je izgubila informacija o nelinearnosti originalnih ograničenja. Očito je da na neki način moramo očuvati informaciju o nelinearnosti originalnih ograničenja. Ideja koja se nameće jest da se umjesto aproksimiranja originalne ciljne funkcije aproksimira Lagrangeova funkcija koja se sastoji od ciljne funkcije i ograničenja

$$L(x_k + p; \lambda_k) = \frac{1}{2} p^T \nabla^2 L(x_k; \lambda_k) p + \nabla L(x_k; \lambda_k)^T p + L(x_k, \lambda_k) \tag{2.6}$$

Primjer 7 prikazuje što će se dogoditi u slučaju ako aproksimiramo Lagrangeovu funkciju.

Primjer 7. Neka je zadan nelinearni optimizacijski problem

$$\min f(x) = (x_2 - 2)^2 - x_1^2 \tag{2.7a}$$

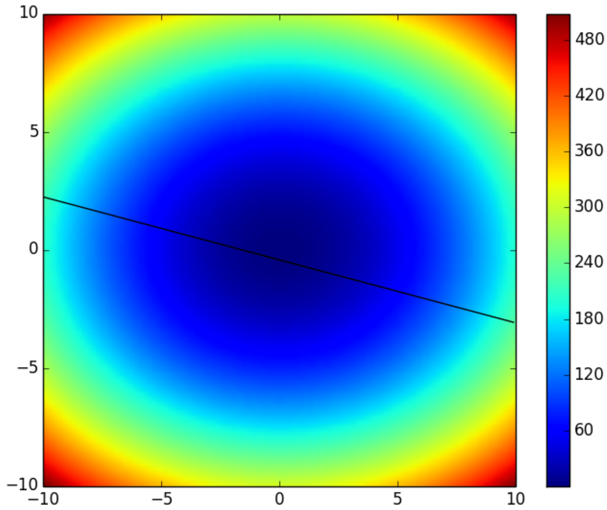
$$uz\ ograničenja : h(x) = 4x_1^2 + x_2^2 - 1 \tag{2.7b}$$

s početnom točkom $x_0 = (\frac{1}{10}, \frac{3}{2})$. U ovom primjeru ćemo zadani nelinearni problem pretvoriti u kvadratični program korištenjem izraza 2.2 i 2.6 dobit ćemo

$$\min\ 2p_1^2 + 3p_2^2 - \frac{1}{5}p_1 + \frac{1}{2}p_2 + \frac{177}{200} \tag{2.8a}$$

$$uz\ ograničenja : \frac{4}{5}p_1 + 3p_2 + \frac{129}{100} = 0. \tag{2.8b}$$

Kao što možemo vidjeti izraz 2.8a se ponešto razlikuje od izraza 2.5a dobivenog u prošlom primjeru. Ako sada grafički prikažemo kvadratični program iz ovog primjera vidjet ćemo kako je zapravo dobivena iznimno značajna razlika u usporedbi s prošlim primjerom. Slika 2.2 prikazuje konture ciljne funkcije 2.8a i ograničenje 2.8b (označeno crnom linijom). Možemo vidjeti kako ciljna funkcija ima jedinstveni minimum i sada možemo riješiti ovako postavljeni kvadratični program.



Slika 2.2: Konture kvadratičnog programa 2.8a

Prethodni primjer nam je dočarao koliko je važno prilikom definicije potproblema kvadratičnog programa uzeti u obzir informacije o nelinearnosti ograničenja.

Za kraj bitno je još napomenuti kako se umjesto izraza 2.6 često koristi sljedeći izraz:

$$L(x_k + p; \lambda_k) = \frac{1}{2} p^T \nabla^2 L(x_k; \lambda_k) p + \nabla f(x_k)^T p + f(x_k). \tag{2.9}$$

Može se pokazati kako su izrazi 2.6 i 2.9 zapravo ekvivalentni do na konstantu, tako da će dati isto rješenje (jedino će se razlikovati u iznosima Lagrangeovih multiplikatora).

Dakle, ako je zadan neki nelinearni problem

$$\min_x f(x) \tag{2.10a}$$

$$uz\ ograničenja : h_i(x) = 0,\ i \in \mathcal{I} \tag{2.10b}$$

$$g_i(x) \geq 0,\ i \in \mathcal{E}. \tag{2.10c}$$

njega ćemo pretvoriti u potproblem kvadratičnog programiranja zadanog kao

$$\min_{p} \; \frac{1}{2}p^T \nabla^2 L(x_k; \lambda_k)p + \nabla f(x_k)^T p + f(x_k) \tag{2.11a}$$

$$uz\ ograničenja : \nabla h_i(x_k)^T p + h_i(x_k) = 0, \; i \in \mathcal{I} \tag{2.11b}$$

$$\nabla g_i(x_k)^T p + g_i(x_k) \geq 0, \; i \in \mathcal{E}. \tag{2.11c}$$

Ako koristimo aproksimaciju Lagrangeove funkcije, osim početne točke potrebni su nam još i početni Lagrangeovi multiplikatori λ_0. Ako je točka x_0 blizu optimalne točke x^*, onda dobru početnu vrijednost za Lagrangeove multiplikatore možemo dobiti izrazom [2]

$$\lambda_0 = -[\nabla h(x_0)^T \nabla h(x_0)]^{-1} \nabla h(x_0)^T \nabla f(x_0) \tag{2.12}$$

2.2 Rješavanje sekvencijalnog kvadratičnog programiranja

U prošlom potpoglavlju pokazali smo kako se iz originalnog nelinearnog problema može dobiti potproblem kvadratičnog programiranja kojeg onda možemo riješiti, primjerice metodom aktivnih ograničenja. Pogledajmo kako će u ovom slučaju izgledati sustav koji će biti potrebno riješiti. Ako želimo pronaći minimum potproblema kvadratičnog programa zadanog kao

$$\min_{p} \; \frac{1}{2}p^T \nabla^2 L(x_k; \lambda_k)p + \nabla f(x_k)^T p + f(x_k) \tag{2.13a}$$

$$uz\ ograničenja : A_k p + c_k = 0 \tag{2.13b}$$

trebamo riješiti sustav koji je zadan kao

$$\begin{bmatrix} \nabla^2 L_k & -A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla f_k \\ -c_k \end{bmatrix}. \tag{2.14}$$

Rješavanjem tog sustava dobiti ćemo pomak p_k i Lagrangeov multiplikator za sljedeću iteraciju λ_{k+1}. Novu točku za sljedeću iteraciju ćemo onda izračunati kao $x_{k+1} = x_k + p_k$. Pogledajmo sada jedan jednostavan primjer koji će nam ilustrirati postupak sekvencijalnog kvadratičnog programiranja [7].

Primjer 8. Neka je zadan optimizacijski problem s ograničenjima

$$\min_x x_1^4 - 2x_1^2 x_2 + x_1^2 + x_1 x_2^2 - 2x_1 + 4 \tag{2.15a}$$

$$uz\ ograni\check{c}enja:\ x_1^2 + x_2^2 - 2 = 0 \tag{2.15b}$$

$$0.25x_1^2 + 0.75x_2^2 - 1 \leq 0 \tag{2.15c}$$

s početnom točkom $x_1 = (3, 2)$. Rješenje zadanog optimizacijskog problema je točka $x^* = (1, 1)$. U ovom primjeru pokazat ćemo nekoliko iteracija sekvencijalnog kvadratičnog programiranja primijenjenog na zadani optimizacijski problem.

Izračunajmo prvo Hesseovu matricu i sve ostale potrebne vektore.

$$H = \begin{bmatrix} 12x_1^2 - 4x_2 + 2 & -4x_1 + 2x_2 \\ -4x_1 + 2x_2 & 2x_1 \end{bmatrix} \tag{2.16a}$$

$$\nabla f = \begin{bmatrix} 4x_1^3 - 4x_1 x_2 + 2x_1 + x_2^2 - 2 \\ -2x_1^2 + 2x_1 x_2 \end{bmatrix} \tag{2.16b}$$

$$\nabla h = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix} \nabla g = \begin{bmatrix} -0.5x_1 \\ -1.5x_2 \end{bmatrix} \tag{2.16c}$$

U svakoj iteraciji pretpostavit ćemo da je radni skup ograničenja W_k prazan, pa nakon što dobijemo rješenje ćemo provjeriti da li ono zadovoljava ograničenje nejednakosti ili ne.

U prvoj iteraciji točka od koje krećemo biti će jednaka početnoj točki $x_1 = (3, 2)$ i nju ćemo iskoristiti kako bismo izradili aproksimaciju kvadratičnog programa za ovu iteraciju. Uvrštavanjem te točke u gore navedene izraze dobit ćemo sljedeće vrijednosti

$$H = \begin{bmatrix} 102 & -8 \\ -8 & 6 \end{bmatrix}, \nabla f = \begin{bmatrix} 92 \\ -6 \end{bmatrix}, f = 64 \tag{2.17a}$$

$$\nabla h = \begin{bmatrix} 6 \\ 4 \end{bmatrix}, h = 11 \tag{2.17b}$$

$$\nabla g = \begin{bmatrix} -1.5 \\ -3 \end{bmatrix}, g = -4.25 \tag{2.17c}$$

Sustav koji ćemo morati riješiti u ovoj iteraciji je

$$\begin{bmatrix} 102 & -8 & -6 \\ -8 & 6 & -4 \\ 6 & 4 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \lambda_1 \end{bmatrix} = \begin{bmatrix} -92 \\ 6 \\ -11 \end{bmatrix} \tag{2.18a}$$

Rješavanjem ovog sustava dobit ćemo $p_1 = (-1.0591, -1.1613)$ i $\lambda_1 = -1.124$. Trebamo još provjeriti da li točka p_1 zadovoljava ograničenje nejednakosti. To možemo lako izračunati

$$\begin{bmatrix} -1.0591 & -1.1613 \end{bmatrix} \begin{bmatrix} -1.5 \\ -3 \end{bmatrix} - 4.25 \geq 0 \tag{2.19a}$$

$$0.8225 \geq 0. \tag{2.19b}$$

Kao što vidimo, dobivena točka zadovoljava postavljeno ograničenje nejednakosti i prihvaćamo pomak p_1. Korištenjem tog pomaka možemo izračunati novu točku $x_2 = x_1 + p_1 = (1.9409, 0.8387)$. Ovu točku koristimo u drugoj iteraciji kako bismo definirali novi potproblem kvadratičnog programiranja

$$H = \begin{bmatrix} 43.85 & -6.0862 \\ -6.0862 & 3.8818 \end{bmatrix}, \nabla f = \begin{bmatrix} 25.32 \\ -4.2785 \end{bmatrix}, f = 13.123 \tag{2.20a}$$

$$\nabla h = \begin{bmatrix} 3.8818 \\ 1.6774 \end{bmatrix}, h = 2.4705 \tag{2.20b}$$

$$\nabla g = \begin{bmatrix} -0.9704 \\ -1.2580 \end{bmatrix}, g = -0.4693 \tag{2.20c}$$

U ovoj iteraciji potrebno je riješiti sljedeći sustav

$$\begin{bmatrix} 43.85 & -6.0862 & -3.8818 \\ -6.0862 & 3.8818 & -1.6774 \\ 3.8818 & 1.6774 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} -25.32 \\ 4.2785 \\ -2.4705 \end{bmatrix} \tag{2.21a}$$

Kada riješimo ovaj sustav dobit ćemo pomak $p_2 = (-0.269, 0.204)$ i $\lambda_2 = 0.091$. Možemo provjeriti kako pomak p_2 zadovoljava postavljeno ograničenje nejednakosti i dobivamo novu točku $x_3 = x_2 + p_2 = (1.3222, 0.7975)$. Prelazimo u iduću iteraciju te računamo potrebne matrice i vektore za novi potproblem kvadratičnog programiranja

$$H = \begin{bmatrix} 19.789 & -3.6938 \\ -3.6938 & 2.6444 \end{bmatrix}, \nabla f = \begin{bmatrix} 6.3087 \\ -1.3875 \end{bmatrix}, f = 4.2126 \tag{2.22a}$$

$$\nabla h = \begin{bmatrix} 2.6444 \\ 1.5951 \end{bmatrix}, h = 0.3843 \tag{2.22b}$$

$$\nabla g = \begin{bmatrix} -0.6611 \\ -1.1963 \end{bmatrix}, g = -0.0858 \tag{2.22c}$$

U ovoj iteraciji potrebno je riješiti sljedeći sustav

$$H = \begin{bmatrix} 19.789 & -3.6938 & -2.6444 \\ -3.6938 & 2.6444 & -1.5951 \\ 2.6444 & 1.5951 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} -6.3087 \\ 1.3875 \\ -0.3843 \end{bmatrix} \tag{2.23a}$$

Kada riješimo ovaj sustav dobit ćemo pomak $p_2 = (-0.619, 0.041)$ i $\lambda_2 = -0.401$. Možemo provjeriti kako pomak p_2 zadovoljava postavljeno ograničenje nejednakosti i dobivamo novu točku $x_3 = x_2 + p_2 = (1.3222, 0.7975)$. Prelazimo u iduću iteraciju te računamo potrebne matrice i vektore za novi potproblem kvadratičnog programiranja

$$H = \begin{bmatrix} 11.316 & -2.2112 \\ -2.2112 & 2.1074 \end{bmatrix}, \nabla f = \begin{bmatrix} 1.5682 \\ -0.10937 \end{bmatrix}, f = 3.0685 \tag{2.24a}$$

$$\nabla h = \begin{bmatrix} 2.1074 \\ 2.0034 \end{bmatrix}, h = 0.1138 \tag{2.24b}$$

$$\nabla g = \begin{bmatrix} -0.5268 \\ -1.5027 \end{bmatrix}, g = -0.0302 \tag{2.24c}$$

U ovoj iteraciji potrebno je riješiti sljedeći sustav

$$H = \begin{bmatrix} 11.316 & -2.2112 & -2.1074 \\ -2.2112 & 2.1074 & -2.0034 \\ 2.1074 & 2.0034 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} -1.5682 \\ 0.10937 \\ -0.1138 \end{bmatrix} \tag{2.25a}$$

Rješavanjem ovog sustava dobit ćemo pomak $p_5 = (-0.106, 0.054)$ i Lagrangeov multiplikator $\lambda_5 = 0.119$. Provjerimo zadovoljava li navedena točka ograničenje nejednakosti.

$$\begin{bmatrix}-0.106 & 0.054\end{bmatrix}\begin{bmatrix}-0.5268\\-1.5027\end{bmatrix} - 0.0302 \geq 0 \tag{2.26a}$$

$$-0.0555 \geq 0. \tag{2.26b}$$

Kao što vidimo, ograničenje nejednakosti nije zadovoljeno. Iz tog razloga to ograničenje dodajmo u sustav i rješavamo ga ponovo (alternativno mogli smo računati duljinu pomaka α i onda se pomaknuti za najveću dozvoljenu duljinu pomaka).

$$\begin{bmatrix}11.316 & -2.2112 & -2.1074 & 0.5268\\-2.2112 & 2.1074 & -2.0034 & 1.5027\\2.1074 & 2.0034 & 0 & 0\\-0.5268 & -1.5027 & 0 & 0\end{bmatrix}\begin{bmatrix}p_1\\p_2\\\lambda\\\mu\end{bmatrix} = \begin{bmatrix}-1.5682\\0.10937\\-0.1138\\0.0302\end{bmatrix} \tag{2.27a}$$

Rješavanjem ovog novog sustava dobit ćemo iznos pomaka $p_5 = (-0.0523, -0.0017)$ i Lagrangeovu multiplikatore $\lambda = 697$ i $\mu = 0.9275$. Ovo rješenje sigurno zadovoljava ograničenje nejednakosti, pa nije potrebno ponovo provjeravati. Nova točka iznosi $x_5 = x_4 + p_5 = (1.0014, 1)$. Možemo zaključiti kako je postupak sekvencijalnog kvadratičnog programiranja konvergirao i da nema smisla nastavljati s daljnjim iteracijama.

Za kraj dat ćemo veoma jednostavan pseudokod sekvencijalnog kvadratičnog programiranja. Algoritam 2 prikazuje pseudokod jednostavnog postupka sekvencijalnog kvadratičnog programiranja koji je opisan u ovom poglavlju.

Algoritam 2 Pseudokod skvencijalnog kvadratičnog programiranja

1: Odabrati inicijalni par (x_0, λ_0);
2: Postavi $k \leftarrow 0$;
3: **repeat**
4: Izračunaj $f_k, \nabla f_k, \nabla^2 L, c_k, A_k$
5: Riješi sustav 2.14 i izračunaj p_k i λ_{k+1}
6: $x_{k+1} = x_k + p_k$;
7: **until** do konvergencije

2.3 Naprednije teme

Postupak koji je objašnjen u prethodnom poglavlju može se primijeniti na rješavanje nelinearnih problema, no treba svakako napomenuti da postoje dodatna poboljšanja ali i neki problemi kojih bi trebali biti svjesni. U ovom poglavlju kratko će se dati osvrt na neke od njih, no bez ulaženja duboko u detalje.

2.3.1 Nekonzistentna linearizacija ograničenja

Jedna moguća poteškoća koja se može pojaviti u sekvencijalnom kvadratičnom programiranju je da linearizacija ograničenja dovede do pojave nerješivih problema. Problem koji će ovdje biti opisan razlikuje se od problema koji je bio uzrokovan time što smo aproksimirali ciljnu funkciju, a ne Lagrangeovu funkciju i ne može se riješiti na taj način. Primjer 9 nam ilustrira problem koji se može dogoditi.

Primjer 9. Neka je zadan optimizacijski problem koji se sastoji od sljedeća dva ograničenja (konkretna ciljna funkcija nam u ovom slučaju nije bitna)

$$\min f(x) \tag{2.28a}$$

$$uz\ ograničenja : x \leq 1 \tag{2.28b}$$

$$x \geq 0 \tag{2.28c}$$

s početnom točkom $x_0 = 1$. Ako lineariziramo ograničenja, dobit ćemo sljedeća ograničenja za potproblem kvadratičnog programiranja

$$-p \geq 0 \tag{2.29a}$$

$$2p - 3 \geq 0. \tag{2.29b}$$

Možemo vidjeti kako ne postoji niti jedna točka koja zadovoljava oba ograničenja, dakle možemo zaključiti da je problem u ovom slučaju loše postavljen, iako to nije bio slušaj za originalni problem.

Iz primjera smo vidjeli kako može doći do problema prilikom definiranja potproblema kvadratičnog programiranja. To se može spriječiti tako da preoblikujemo nelinearni problem

$$\min_{x,v,w,t} f(x) + \mu \sum_{i\in\mathcal{E}} (v_i + w_i) + \mu \sum_{i\in\mathcal{I}} t_i \tag{2.30a}$$

$$uz\ uvjete : h_i(x) = v_i + w_i,\ i \in \mathcal{E} \tag{2.30b}$$

$$g_i(x) \geq -t_i,\ i \in \mathcal{I}. \tag{2.30c}$$

$$v, w, t \geq 0. \tag{2.30d}$$

Parametar μ nazivamo parametrom kazne i mora biti pozitivan. Ako nelinearni problem ima rješenje, onda će i ovakva definicija kvadratičnog programa pronaći optimalno rješenje za dovoljno velik iznos parametra μ. S druge strane, ako ne postoji rješenje nelinearnog programa (kao u gornjem primjeru), onda će pomoćni problem definiran kao 2.30 pronaći dovoljno "dobru" točku koja ne zadovoljava ograničenja. Svakako se preporučuje prvo pokušati riješiti nelinearni problem na standardni način kako je to ranije opisano i ako to ne uspije, onda pokušati problem riješiti tako da se nelinearni program prilagodi korištenjem izraza 2.30. Osim ovog postupka postoje naravno i mnogi drugi postupci koji se mogu iskoristiti kako bi se izbjegao ovaj problem.

2.3.2 Aproksimacija Hesseove matrice

Kao što je pokazano, sekvencijalno kvadratično programiranje zahtijeva računanje Hesseove matrice od Lagrangeove funkcije $L(x_k, \lambda_k)$ u svakoj iteraciji algoritma. Ponekad neće biti moguće odrediti Hesseovu matricu, tako da je korisno tu matricu zamijeniti aproksimacijom. U tu svrhu se primjerice može iskoristiti BFGS metoda koja se je pokazala uspješnom u optimizaciji bez ograničenja.

Pogledajmo ukratko kako BFGS aproksimacija radi. Prvo definirajmo dva vektora

$$s_k = x_{k+1} - x_k \tag{2.31a}$$

$$y_k = \nabla L(x_{k+1}, \lambda_{k+1}) - \nabla L(x_k, \lambda_k). \tag{2.31b}$$

Ako je B_k aproksimacija Hesseove matrice u trenutnom koraku, onda će aproksimacija u idućem koraku biti jednaka

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \tag{2.32}$$

Ako je Hesseova matrica Lagrangeove funkcije pozitivno definitna u području u kojem se obavlja minimizacija, onda će aproksimacija pomoću BFGS metode konvergirati brzo i robusno prema minimumu. Ako to nije slučaj, odnosno ako Hesseova matrica nije pozitivno definitna, može doći do određenih problema prilikom aproksimacije

Hesseove matrice pomoću pozitivno definitne matrice. Iz tog razloga preporučuje se korištenje takozvane prigušene BFGS metode koja se ponaša ponešto bolje, ali se svejedno može loše ponašati na težim problemima.

2.3.3 Globalna konvergencija

Kako bi se osiguralo da sekvencijalno kvadratično programiranje konvergira s udaljenih točaka, često se koristi takozvana funkcija doprinosa (engl. *merit function*) ϕ. Funkciju doprinosa koristit ćemo kako bismo odredili duljinu pomaka p_k kod sekvencijalnog kvadratičnog programiranja (do sada smo uvijek uzimali cijeli iznos pomaka). Veoma je bitno da funkcija doprinosa ne ometa "dobre" korake, odnosno one korake koji se približavaju rješenju.

Jedna od mogućih funkcija doprinosa je takozvana l_1 funkcija doprinosa definirana kao

$$\phi_1(x;\mu) = f(x) + \mu||c(x)||_1 \tag{2.33}$$

pri čemu parametar μ nazivamo parametrom kazne. Ideja funkcije doprinosa je zapravo da se ograničenja i ciljna funkcija spoje u jednu novu funkciju o čijoj će vrijednosti onda ovisiti duljina pomaka. Naime, ne mora vrijediti da ćemo, ako se pomaknemo za puni iznos pomaka p_k, dospjeti do "najbolje" točke, dapače, može se dogoditi da s manjim pomakom možemo doći do puno bolje točke. Naravno, u tom slučaju nije dovoljno uzeti u obzir samo minimizaciju ciljne funkcije, već i koja točka najmanje narušava postavljena ograničenja i upravo je iz tog razloga definirana funkcija doprinosa.

U linijskom pretraživanju, korak $\alpha_k p_k$ biti će prihvaćen ako je sljedeće ograničenje zadovoljeno

$$\phi_1(x_k + \alpha_k p_k; \mu_k) \leq \phi_1(x_k; \mu_k) + \eta\alpha_k D(\phi_1(x_k;\mu); p_k), \qquad \eta \in (0,1), \tag{2.34}$$

gdje je $D(\phi_1(x_k;\mu); p_k)$ usmjerena derivacija funkcije doprinosa ϕ_1 u smjeru p_k.

Potrebno je još objasniti kako odrediti vrijednost parametra kazne. Postoji nekoliko mogućnosti odabira parametra kazne μ, ali najčešće se koristi sljedeći izraz:

$$\mu \geq \frac{\nabla f_k^T p_k + (\sigma/2) p_k^T \nabla^2 L(x_k, \lambda_k) p_k}{(1-\rho)||c_k||_1}, \qquad \rho \in (0,1), \tag{2.35}$$

pri čemu je parametar σ jednak

$$\sigma = \begin{cases} 1 & ako\ p_k^T \nabla^2 L(x_k, \lambda_k) p_k > 0 \\ 0 & ina\check{c}e \end{cases} \tag{2.36}$$

2.3.4 Sekvencijalno kvadratično programiranje s linijskom pretragom

U ovom potpoglavlju prikazat će se pseudokod malo složenijeg postupka sekvencijalnog kvadratičnog programiranja od onog prikazanog u algoritmu 2. U ovoj inačici sekvencijalnog kvadratičnog programiranja iskoristit će se neke stvari opisane u ovom poglavlju, kao što su linijska pretraga i aproksimacija Hesseove matrice. Algoritam 3 prikazuje pseudokod navedene inačice sekvencijalnog kvadratičnog programiranja. Naravno navedeni postupak se dalje može proširiti i nekim drugim unapređenjima (primjerice u ovom postupku se ne provjerava da li je pozproblem kvadratičnog programa rješiv). Još je bitno za napomenuti kako se pretpostavlja da je potproblem kvadratičnog programa konveksan, tako da ga možemo riješiti metodom aktivnih ograničenja.

Algoritam 3 Pseudokod sekvencijalnog kvadratičnog programiranja s linijskom pretragom

Odaberi parametre $\eta \in (0, 0.5)$ i $\tau \in (0, 1)$;
Odredi inicijalni par (x_0, λ_0);
Izračunaj $f_0, \nabla f_0, c_0, A_0$;
Izračunaj $\nabla^2 L$ ili inicijalnu aproksimaciju B_0;
repeat
    Izračunaj p_k i $\hat{\lambda}$ rješavanjem izraza 2.11;
    Postavi $p_\lambda = \hat{\lambda} - \lambda_k$;
    Odaberi μ_k koji zadovoljava 2.36 uz $\sigma = 1$;
    $\alpha_k \leftarrow 1$
    while $\phi_1(x_k + \alpha_k p_k; \mu_k) > \phi(x_k; \mu_k) + \eta \alpha_k D_1(\phi(x_k; \mu_k); p_k)$ **do**
        $\alpha_k \leftarrow \tau_\alpha$ za neki $\tau_\alpha \in (0, \tau]$;
    end while
    $x_{k+1} \leftarrow x_k + \alpha_k p_k$;
    $\lambda_{k+1} \leftarrow \lambda_k + \alpha_k p_\lambda$;
    Izračunaj $f_{k+1}, \nabla f_{k+1}, c_{k+1}, A_{k+1}$;
    Izračunaj $\nabla^2 L_{k+1}$ ili aproksimaciju B_{k+1};
until do konvergencije

Još jedno moguće unaprjeđenje su takozvane *warm-start* procedure, koje mogu ubrzati ranije opisani algoritam. Primjerice, aktivni skup potproblema kvadratičnog programiranja možemo inicijalizirati s konačnim aktivnim skupom iz prošle iteracije sekvencijalnog kvadratičnog programiranja.

Bibliografija

[1]Mark S. Gockenbach. *Introduction to sequential quadratic programming* (cit. on p. 18).

[2]Ronald H.W. Hoppe. *Optimization I*. 2006 (cit. on p. 21).

[3]Andreas Kugi. *Optimierung*. 2011 (cit. on pp. 2, 3, 8).

[4]Josipa Pina Milišić and Ana Žgaljić Keko. *Uvod u numeričku matematiku za inženjere*. Element, 2013 (cit. on p. 4).

[5]Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 1999 (cit. on pp. 1, 8, 10, 12, 13).

[6]Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2006.

[7]P. Venkataraman. *Applied Optimization with MATLAB Programming*. Wiley, 2009 (cit. on p. 21).

[8]William T. Vetterling William H. Press Saul A. Teukolsky and Brian P. Flannery. *Numerical Recipes*. Cambridge University Press, 2007 (cit. on p. 4).