1 Problem matematičkog programiranja

Definicija 1 *Problem* ***matematičkog programiranja*** *je nalaženje $x^* \in X$ za koje je*

$$f(x^*) = \min_{x \in X} f(x), \tag{1.1}$$

gde je $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$ ***funkcija cilja****, $X \subseteq \mathbb{R}^n$ je* ***skup dopustivih vrednosti****.*

Drugim rečima, matematičko programiranje je minimizacija realne funkcije nad skupom dopustivih vrednosti. Problem je ekvivalentan maksimizaciji funkcije $-f$.

2 Optimitacija bez ograničenja

Ako je skup dopustivih vrednosti $X = \mathbb{R}^n$, govorimo o **optimizaciji bez ograničenja**. Vidi [**?**].

Problem (1.1) može imati više rešenja.

Definicija 2 *Ako postoji $x^* \in X$ takvo da za sve $x \in X$ važi $f(x^*) \leq f(x)$, kažemo da je x^** ***globalni minimum*** *funkcije f nad X.*

Definicija 3 *Ako postoji $x^* \in X$ i okolina $\mathcal{N}(x^*)$ takva da za sve $x \in \mathcal{N}(x^*) \cap X$ važi $f(x^*) \leq f(x)$, kažemo da je x^** ***lokalni minimum*** *za f.*

Algoritmi koje ćemo posmatrati su kreirani tako da nalaze stacionarnu tačku, koja je pod dodatnim uslovima lokalni minimum.

Definicija 4 *Kažemo da je x^** ***strogi lokalni minimum*** *funkcije f ako postoji okolina $\mathcal{N}(x^*)$ takva da za sve $x \in \mathcal{N}(x^*) \backslash \{x^*\} \cap X$ važi $f(x^*) < f(x)$.*

Glavna alatka koju ćemo koristiti u dokazima i ispitivanju ponašanja funkcije je **Tejlorova teorema**, koja daje razvoj funkcije f u okolini tačke x:

Teorema 1 (Taylor) *Neka je funkcija $f : \mathbb{R}^n \to \mathbb{R}$ neprekidno diferencijabilna i neka $p \in \mathbb{R}^n$. Tada, za neko $t \in (0, 1)$*

$$f(x + p) = f(x) + \nabla f(x + tp)\, p\,. \tag{2.1}$$

Ako je, pritom, f dva puta neprekidno diferencijabilna, onda

$$\nabla f(x + p) = \nabla f(x) + \int_0^1 \nabla^2 p^T\, f(x + tp)\, dt\,, \tag{2.2}$$

odnosno, za neko $t \in (0, 1)$

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

Koristeći Tejlorovu teoremu mogu se dokazati teoreme o potrebnim uslovima i teorema o dovoljnom uslovu za lokalni ekstrem funkcije.

Teorema 2 (Potreban uslov prvog reda) *Neka je x^* lokalni minimum funkcije f i neka je f neprekidno diferencijabilna u otvorenoj okolini od x^*. Onda je $\nabla f(x^*) = 0$.*

Definicija 5 *Za neprekidno diferencijabilnu funkciju $f : \mathbb{R}^n \to \mathbb{R}$ tačka $x^* \in X$ je **stacionarna tačka** ako $\nabla f(x^*) = 0$.*

Posledica teoreme 2 je da su za neprekidno diferencijabilnu funkciju svi lokalni minimumi stacionarne tačke.

Teorema 3 (Potreban uslov drugog reda) *Neka je x^* lokalni minimum funkcije f i neka je $\nabla^2 f$ neprekidno u otvorenoj okolini od x^*. Onda je $\nabla f(x^*) = 0$ i $\nabla^2 f(x^*)$ je pozitivno semidefinitno.*

Prvi deo ove teoreme je posledica teoreme 2, drugi deo sledi iz Tejlorove teoreme.

Teorema 4 (Dovoljan uslov drugog reda) *Neka je $\nabla^2 f$ neprekidno u otvorenoj okolini x^*, neka je $\nabla f(x^*) = 0$ i neka je $\nabla^2 f(x^*)$ pozitivno definitno. Onda je x^* strogi lokalni minimum od f.*

Teorema 5 *Za konveksnu funkciju f lokalni minimum je i globalni minimum. Ako je f diferencijabilna i konveksna, stacionarna tačka je i globalni minimum.*

Ova teorema omogućava da globalni optimum za diferencijabilnu konveksnu funkciju dobijemo rešavajući sistem jednačina $\nabla f(x) = 0$.

Sa druge strane, rešavanje **sistema jednačina** $r(x) = 0$, gde je r vektorska funkcija n promenljivih $r : \mathbb{R}^n \to \mathbb{R}^n$ se može postaviti kao problem minimizacije funkcije

$$\min_x \|r(x)\|^2 .$$

Traži se x^*, minimum funkcije $\|r(x^*)\|^2$, za koji važi $\|r(x^*)\|^2 = 0$.

U praksi se vrlo često pojavljuju problemi rešavanja sistema (nelinearnih) jednačina sa predefinisanim sistemom (kad imamo više jednačina od nepoznatih). U tom slučaju obično ne postoji rešenje. Ipak, tada se mogu naći vrednosti parametara koje "najbolje" odgovaraju datom modelu. Pritom najbolje mislimo u smislu sume kvadrata odstupanja.

Ako su date jednačine $r_j(x) = 0$, $j = 1, 2, \ldots, m$, gde su r_j glatke funkcije i $m \geq n$, **problem najmanjih kvadrata (least squares problem)** je minimizacija funkcije

$$f(x) = \frac{1}{2} \min_x \sum_{j=1}^{m} r_j^2(x) .$$

Problem najmanjih kvadrata se pojavljuje u puno oblasti, kad želimo da zadati model koji zavisi od n parametara najbolje prilagodimo datim vrednostima u m tačaka.

Sistem jednačina je specijalni slučaj problema najmanjih kvadrata, a oba su specijalni slučaj problema matematičkog programiranja.

3 Postupci za rešavanje problema optimizacije bez ograničenja

Posmatraćemo problem minimizacije glatke funkcije $f : \mathbb{R}^n \to \mathbb{R}$ bez ograničenja na vrednost argumenta:

$$\min_{x \in \mathbb{R}^n} f(x) \, . \tag{3.1}$$

Većina postupaka traži lokalni minimum, odnosno stacionarnu tačku koja je rešenje sistema jednačina

$$\nabla f(x) = 0 \, . \tag{3.2}$$

Za konveksnu funkciju problem traženja lokalnog i globalnog minimuma su ekvivalentni.

Postupci optimizacije bez ograničenja koje ćemo posmatrati nalaze lokalni minimum, obeležavaćemo ga x^*.

Postupke možemo podeliti u zavisnosti od toga koliko izvoda funkcije cilja koriste. Postupke koji koriste $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$ nazivamo **postupci sa drugim izvodom**; ako koriste $f(x)$, $\nabla f(x)$, to su **gradijentni postupci**; odnosno, ako koriste samo proceduru za računanje $f(x)$, to su **postupci bez izvoda**.

U zavisnosti od toga koliko podataka koriste, postupci zahtevaju manji ili veći stepen glatkosti funkcije cilja.

Postupci koje posmatramo su **iterativni postupci**. To znači da se zadaje početna tačka x_0, a algoritam generiše iterativni niz $\{x_k\}$ koji konvergira ka x^*.

Vrednosti funkcije, gradijenta i Hesijana u tekućoj tački iterativnog postupka ćemo obeležavati redom

$$f_k = f(x_k), \nabla f_k = \nabla f(x_k), \nabla^2 f_k = \nabla^2 f(x_k), k = 1, 2, \ldots \tag{3.3}$$

Obično je niz vrednosti funkcije $\{f_k\}$ koji odgovara generisanom iterativnom nizu **monotono opadajući**. Postoje postupci koji ne zahtevaju da se u svakom iterativnom koraku generiše tačka u kojoj je vrednost funkcije manja, mada se i kod njih posle nekog broja iteracija vrednost funkcije mora smanjiti. To su **nemonotoni postupci**.

Pri generisanju niza $\{x_k\}$ u iterativnom koraku algoritam daje pravilo kako se, koristeći dotad generisane tačke $x_1, x_2, \ldots, x_k$, generiše tačka x_{k+1}. U praksi se pri tome koriste dva glavna pristupa: **linijsko pretraživanje** i **oblast poverenja**.

3.1 Postupci linijskog pretraživanja

Glavna razlika između postupaka Oblast poverenja i Linijsko pretraživanje je u redosledu izbora: 1 - pravca od tekuće do sledeće iteracije i 2 - dužine koraka do sledeće iteracije. Neka su iteracije oblika:

$$x_{k+1} = x_k + \alpha_k p_k^T \, . \tag{3.4}$$

U postupku Oblast poverenja prvo se postavlja maksimalna dužina koraka (poluprečnik oblasti poverenja), a potom se određuje pravac u kome se ostvaruje najveće smanjenje funkcije cilja.

Postupak Linijsko pretraživanje radi obrnuto. Pretpostavljamo da je dat pravac p koji sa gradijentom zaklapa tup ugao, odnosno, za koji važi $\nabla f_k^T p_k < 0$. Kažemo da je to **opadajući pravac**.

Za zadati opadajući pravac p_k traži se dužina koraka α_k koja će dovesti do najvećeg smanjenja funkcije cilja po tom pravcu:

$$\min_{\alpha>0} f(x_k + \alpha p_k^T) \, . \tag{3.5}$$

Na osnovu Tejlorove teoreme sigurno postoji $\alpha > 0$ za koje je $f_{k+1} < f_k$.

Pri tome nije neophodno naći tačno rešenje problema (3.5), već je moguće da je računski efikasnije naći njegovo približno rešenje. Obično se u jednoj iteraciji generiše nekoliko dužina koraka dok se ne nađe dovoljno dobra aproksimacija minimuma iz (3.5).

Metode linijskog pretraživanja daju konvergenciju za proizvoljan izbor opadajućih pravaca.

Ipak, nekoliko izbora pravaca je više zastupljeno.

3.2 Pravac najbržeg opadanja

Najjednostavniji opadajući pravac se može dobiti aproksimacijom Tejlorovog razvoja prvog reda (2.1)

$$f(x_k + \alpha p) = f_k + \alpha \nabla f(x_k + tp) \; p \, ,$$

sa $t = 0$: $f_{k+1} \approx f_k + \alpha \nabla f_k \, p$. Ako tražimo jedinični pravac, $\|p\| = 1$, najveće smanjenje funkcije cilja se dobija kada se reši $\min\limits_p p^T \nabla f_k$.

Kako je $\nabla f_k \, p = \|\nabla f_k\| \, \|p\| \cos\phi$, gde je ϕ ugao između pravca p i gradijenta, vidimo da se najveće smanjenje postiže za

$$\cos\phi = -1 \, , \quad \text{što se dobija za } p_k = -\frac{1}{\|\nabla f_k\|} \nabla f_k^T \, .$$

U praksi se nakon određivanja pravca ovom metodom određuje dužina koraka koja skalira pravac, pa se zbog jednostavnosti za opadajući pravac uzima **pravac negativnog gradijenta**

$$p_k := -\nabla f_k^T \, . \tag{3.6}$$

Postupci Linijskog pretraživanja u kombinaciji sa ovim pravcem se nazivaju **postupci najbržeg opadanja**. Njihova prednost je što u implementaciji nisu potrebni proračuni za izvode drugog reda.

Ovi postupci su za neke probleme prespori.

3.3 Njutnov pravac

Ako u Tejlorovoj teoremi (2.3) stavimo $t = 0$, dobijamo aproksimaciju

$$f(x_k + p) \approx f_k + \nabla f_k \, p + \frac{1}{2} p^T \nabla^2 f_k p =: m_k(p) \, . \tag{3.7}$$

Ako pretpostavimo da je $\nabla^2 f_k$ pozitivno definitno, Njutnov pravac se dobija nalaženjem minimuma za $m_k(p)$. Izjednačavanjem izvoda od $m_k(p)$ po p sa nulom, dobija se sistem čije rešenje je

$$p_k^N = -\left(\nabla^2 f_k\right)^{-1} \nabla f_k^T \, . \tag{3.8}$$

Za dovoljno glatku funkciju aproksimacija Hesijana $\nabla^2 f(x_k + tp) \approx \nabla^2 f_k$ je dobra za male $\|p\|$, što daje brzu konvergenciju u blizini rešenja. Pravac (3.8) nazivamo **Njutnov pravac**.

Za pozitivno definitno $\nabla^2 f_k$ Njutnov pravac je opadajući pravac jer:

$$\nabla f_k p_k^N = -p_k^{N\,T} \nabla^2 f_k p_k^N \leq -\sigma_k \|p_k^N\|^2 , \tag{3.9}$$

za neko $\sigma_k > 0$.

Za Njutnov pravac dužina koraka iteracije (3.4) koji daje najveće smanjenje funkcije je $\alpha_k = 1$. Iterativni postupak sa Njutnovim pravcem i korakom $\alpha_k = 1$ nazivamo **Njutnov iterativni postupak**.

Ako $\nabla^2 f_k$ nije pozitivno definitno, moguće je da Njutnov pravac ne postoji, ili, ako postoji, da nije opadajući pravac.

U Njutnovom postupku se u svakoj iteraciji izračunava i faktoriše Hesijan. To je računski zahtevno, pored toga što zahteva poznavanje formule Hesijana.

Ako se pribegne računanju Hesijana pomoću konačnih razlika, nije potrebno poznavanje formule Hesijana, ali je postupak i dalje računski skup.

3.4 Kvazi Njutnovi postupci

Kvazi Njutnovi postupci namesto Hesijana $\nabla^2 f_k$ u formuli (3.8) koriste neku njegovu aproksimaciju B_k.

U kvazi Njutnovim metodama opadajući pravac se dobija po formuli

$$p_k = -B_k^{-1} \nabla f_k^T , \tag{3.10}$$

gde je B_k simetrična, nesingularna matrica koja aproksimira Hesijan. Obično se za matricu B_k u prvoj iteraciji uzima jedinična matrica, a potom se u svakoj iteraciji ažurira koristeći nove vrednosti funkcije i gradijenta.

Osnovna ideja kvazi Njutnovih postupaka je da se uopšti ideja sečice iz postupka za rešavanja jednodimenzionalnih jednačina. Aproksimacija Hesijana se bira tako da zadovoljava **jednačinu sečice**:

$$B_{k+1} s_k = y_k, \text{ gde je } s_k = x_{k+1} - x_k , \; y_k = \nabla f_{k+1} - \nabla f_k . \tag{3.11}$$

Matrica B_k treba da je simetrična (isto kao Hesijan).

Za aproksimaciju Hesijana često se koristi **formula SR1, odnosno Symmetric-Rank-one formula**:

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k} . \tag{3.12}$$

Ova formula daje simetrične matrice koje zadovoljavaju jednačinu sečice. Razlika između dve susedne iteracije je matrica ranga 1.

Još popularnija je BFGS formula koju su izumeli Broyden, Fletcher, Goldfarb i Shanno:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k^T}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} . \tag{3.13}$$

Ova formula takođe daje simetrične matrice koje zadovoljavaju jednačinu sečice. Razlika između dve susedne iteracije je matrica ranga 2.

Praktična implementacija BFGS postupka umesto aproksimacije B_k koristi aproksimaciju inverznog Hesijana $H_k = B_k^{-1}$.

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T \ , \ \text{gde je } \rho_k = \frac{1}{y_k^T s_k} \ . \qquad (3.14)$$

Koristeći ovu aproksimaciju inverznog Hesijana pravci iterativnog koraka su

$$p_k = -H_k \nabla f_k^T \ .$$

3.5 Dužina koraka linijskog pretraživanja

Posle određivanja pravca, u svakoj iteraciji (3.4) se linijskim pretraživanjem određuje dužina koraka.

Pravac najbržeg opadanja i Njutnov pravac su specijalni slučajevi formule (3.10), sa $B_k = I$ i $B_k = \nabla^2 f_k$ redom.

Pri određivanju dužine koraka se polazi od koraka $\alpha_k = 1$ koji bi odgovarao Njutnovom pravcu. Ako taj korak ne zadovoljava uslove koji garantuju konvergenciju, korak se skraćuje.

Za dati opadajući pravac, idealno je da se dužina koraka α_k odredi kao globalni minimum funkcije cilja duž tog pravca

$$\phi(\alpha) = f(x_k + \alpha p_k) \ . \qquad (3.15)$$

Određivanje tog minimuma ćemo zvati **tačno linijsko pretraživanje**. Tačno linijsko pretraživanje je računski skupo.

Zato se primenjuje **netačno linijsko pretraživanje** koje određuje dužinu koraka sa zadovoljavajućim smanjenjem funkcije (3.15) u odnosu na broj računanja vrednosti funkcije i gradijenta.

Da bi se obezbedila konvergencija postupka potrebno je postaviti uslove na dužinu koraka. Armijo uslov zahteva da dužina koraka α_k pruži **dovoljno umanjenje** funkcije cilja:

$$f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f_k p_k \ , \ \text{za neko } c_1 \in (0,1) \ . \qquad (3.16a)$$

Odnosno: dopuštene dužine koraka α su one za koje je $\phi(\alpha) \leq l(\alpha)$, gde je $l(\alpha)$, desna strana nejednakosti (3.16a). To su na slici 1 vrednosti za koje je grafik funkcije $\phi(\alpha)$ ispod prave $l(\alpha)$. U praksi se c_1 uzima malo, recimo $c_1 = 10^{-4}$.

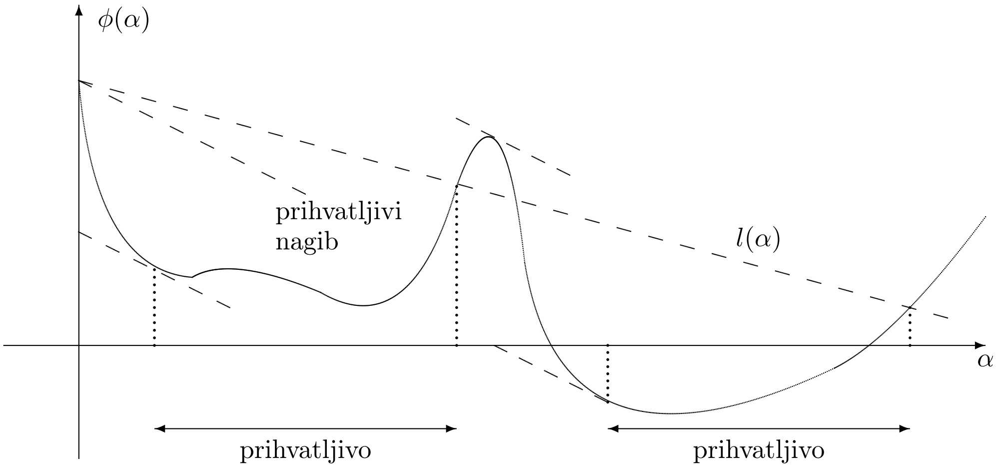
Da se ne bi za dužinu koraka uzimale jako male vrednosti koje ne dovode do napretka u iteracijama, postavlja se **uslov nagiba**

$$\nabla f(x_k + \alpha p_k) p_k \geq c_2 \nabla f_k p_k \ , \ \text{za neko } c_2 \in (c_1, 1) \ . \qquad (3.16b)$$

Leva strana nejednakosti (3.16b) je izvod $\phi'(\alpha)$, treba da je veća od umnoška $\phi'(0)$. To su na slici 1 apscise tačaka krive u kojima nagib nije strmiji od prihvatljivog nagiba. U ovom uslovu se traži da je $c_2 > c_1$, zato da bi uslov (3.16b) mogao biti ispunjen istovremeno sa uslovom (3.16a).

Zajedno se uslovi (3.16a), (3.16b) nazivaju **Volfovi (Wolfe) uslovi**.

U praktičnoj primeni, obično se uzima $c_2 = 0.1$ za gradijentni korak ili korak konjugovanog gradijenta; odnosno $c_2 = 0.9$ za Njutnov ili kvazi Njutnove korake.



Slika 1: Slika oblasti za α koje zadovoljavaju Wolfe uslove

Lema 1 *Neka je $f : \mathbb{R}^n \to \mathbb{R}$ neprekidno diferencijabilna. Neka je p_k opadajući pravac u x_k i neka je f ograničena odole na skupu $\{x_k + \alpha p_k | \alpha > 0\}$. Ako je $0 < c_1 < c_2 < 1$, onda postoji interval vrednosti α koje zadovoljavaju Volfove uslove* (3.16a), (3.16b).

Za dokaz konvergencije metode linijskog pretraživanja koristi se teorema Zoutendijka:

Teorema 6 (Zoutendijk) *Neka je iterativni niz $\{x_k\}$ definisan iteracijom* (3.4) *sa opadajućim pravcima p_k i neka α_k* zadovoljava *Volfove uslove* (3.16a), (3.16b).

Neka je f ograničena odole na $\mathbb{R}^n$ i neprekidno diferencijabilna na otvorenom skupu $\mathcal{N}$ koji sadrži skup $\mathcal{L} = \{x : f(x) \leq f(x_0)\}$, gde je x_0 data početna iteracija.

Neka je, dalje, gradijent ∇f Lipšic neprekidan nad $\mathcal{N}$, odnosno:

$$\forall x, \tilde{x} \in \mathcal{N}, \; \|\nabla f(x) - \nabla f(\tilde{x})\| \leq L\|x - \tilde{x}\| \,, \quad \text{za neko } \; L > 0. \tag{3.17}$$

Onda

$$\sum_{k \geq 0} \cos^2 \theta_k \; \|\nabla f(x_k)\|^2 < \infty \,, \tag{3.18}$$

gde je θ_k ugao koji zaklapaju p_k i $-\nabla f_k$. Njegov kosinus se dobija po formuli

$$\cos \theta_k = \frac{-\nabla f_k^T p_k}{\|\nabla f_k\| \; \|p_k\|} \,. \tag{3.19}$$

Pored uslova (3.16a), (3.16b), postoji verzija ove teoreme sa drugim skupom uslova:

Na primer, **jaki Volfovi uslovi**:

$$f(x_k + \alpha_k p_k) \leq f_k + c_1 \alpha_k \nabla f_k^T p_k \tag{3.20a}$$

$$|\nabla f(x_k + \alpha_k p_k)^T p_k| \leq c_2 |\nabla f_k^T p_k| \,, \tag{3.20b}$$

za neke $0 < c_1 < c_2 < 1$.

Ili, na primer, **Goldštajnovi (Goldstein) uslovi**:

$$f_k + (1-c)\alpha_k \nabla f_k^T p_k \le f(x_k + \alpha_k p_k) \le f_k + c\alpha_k \nabla f_k^T p_k \,, \tag{3.21}$$

za neko $c \in (0, \frac{1}{2})$.

Ista posledica: (3.18) koju nazivamo **Zoutendijkov uslov** sledi iz para uslova (3.20a), (3.20b), odnosno iz uslova (3.21).

Prosta posledica Zoutendijkovog uslova je

$$\lim_{k\to\infty} \cos^2\theta_k \, \|\nabla f_k\|^2 = 0 \,. \tag{3.22}$$

Da bi se obezbedila konvergencija niza $\{x_k\}$ ka stacionarnoj tački, potrebno je postaviti još uslov na pravce p_k. Niz uglova između pravaca p_k i gradijenata ∇f_k ne sme se nagomilavati kod pravog ugla:

$$\forall k \in \{0,1,2,\ldots\},\ \cos\theta_k \ge \delta \,, \quad \text{za neko} \quad \delta > 0. \tag{3.23}$$

Direktna posledica Zoutendijkovog uslova i uslova (3.23) je:

$$\lim_{k\to\infty} \|\nabla f(x_k)\| = 0 \,. \tag{3.24}$$

Da podsetimo: globalna konvergencija, odnosno uslov (3.24), znači da postupak konvergira ka x^*, stacionarnoj tački funkcije f. Tek ako se na f postave dodatni uslovi sledi da je x^* lokalni, odnosno globalni minimum.

Za postupke najbržeg opadanja (pravci negativnog gradijenta), važi $\cos\theta_k = 1$. Za te postupke je dovoljno linijskim pretraživanjem uzimati dužine koraka koje zadovoljavaju Volfove uslove (ili Jake Volfove uslove ili Goldštajnove uslove) i važiće globalna konvergencija.

Za kvazi Njutnove postupke, iz pozitivne definitnosti aproksimacije Hesijana B_k sledi da su im pravci opadajući. Za Njutnove pravce (koji su u neku ruku specijalni slučaj kvazi Njutnovih pravaca) to imamo izvedeno u (3.9).

Dodajmo pretpostavku da matrice B_k imaju **uniformno ograničen uslovni broj**. To znači da:

$$\forall k \in \{0,1,2,\ldots\},\ \|B_k\|\,\|B_k^{k-1}\| \le M \,, \quad \text{za neko} \quad M > 0. \tag{3.25}$$

Lako se može pokazati iz (3.19) da iz (3.25) sledi $\cos\theta_k \ge \dfrac{1}{M}$.

Dakle: U iterativnom nizu (3.4), Njutnov ili kvazi Njutnov postupak sa pravcima (3.10), uz uslov (3.25) čije dužine koraka daju Zoutendijkov uslov jesu globalno konvergentni.

Teorema 6 daje i više: Ako neki algoritam generiše iterativni niz (3.4) koji u svakoj iteraciji daje smanjenje funkcije cilja i svakih m koraka uzima pravac najbržeg opadanja $p_k = -\nabla f_k$ sa dužinom koraka koja zadovoljava Volfove ili Goldštajnove uslove, može se dokazati

$$\liminf_{k\to\infty} \|\nabla f(x_k)\| = 0 \,. \tag{3.26}$$

3.6 Bektreking linijsko pretraživanje

Kada su pravci linijskog pretraživanja Njutnovi ili kvazi Njutnovi, u želji da postupak ima kvadratnu brzinu konvergencije Njutnovog postupka, treba i korake α_k birati bliže Njutnovom koraku $\alpha_k = 1$. Zato u Linijskom pretraživanju krećemo sa korakom $\bar{\alpha} = 1$.

Jedan praktičan algoritam za određivanje dužine koraka koji može zadovoljiti prvi Volfov uslov (3.16a), a da koraci istovremeno ne budu previše kratki, zove se **bektreking linijsko pretraživanje**, algoritam Backtracking Line Search.

Algoritam 1 Backtracking Line Search

Require: $\bar{\alpha} > 0, 0 < \rho_{lo} < \rho_{hi} < 1, c \in (0, \frac{1}{2})$
Ensure: α koje zadovoljava (3.16a)
  $\alpha := \bar{\alpha}$;
  while $f(x_k + \alpha p_k) > f(x_k) + c\alpha\nabla f_k p_k$ **do**
    choose $\rho \in [\rho_{lo}, \rho_{hi}]$;
    $\alpha := \rho\alpha$;
  end while
  return

Ova jednostavna procedura za nalaženje dužine koraka α_k duž opadajućeg pravca p_k se poziva sa početnom vrednošću koraka $\bar{\alpha} = 1$ za Njutnovu i kvazi Njutnove metode. Može se odrediti i manja početna vrednost za drugi izbor opadajućih pravaca. Kad se procedura bektreking linijskog pretraživanja zaustavi sa $\alpha_k = \alpha$, dužina koraka će biti takva da korak α_k/ρ ne zadovoljava uslov (3.16a), a korak α_k zadovoljava.

Vrednost koeficijenta c se bira tako da i malo smanjenje funkcije cilja može da zaustavi petlju traženja koraka α, recimo, $c = 10^{-4}$.

Faktor kontrakcije ρ može da se menja tokom iteracija unutar linijskog pretraživanja. Jedino treba da se sačuva $\rho \in [\rho_{lo}, \rho_{hi}]$, za unapred odabrane konstante $0 < \rho_{lo} < \rho_{hi} < 1$.

3.7 Algoritmi linijskog pretraživanja

Linijsko pretraživanje je postupak nalaženja minimuma jednodimenzionalne funkcije (3.15) uz zadovoljenje uslova (3.16a), (3.16b) ili nekih drugih uslova koji garantuju konvergenciju.

Za konveksnu kvadratnu funkciju $f(x) = \frac{1}{2}x^T Qx - b^T x$ tačan minimum duž pravca $\{x_k + \alpha p_k, \alpha > 0\}$ imamo dat formulom

$$\alpha_k = \frac{\nabla f_t^T p_k}{p_k^T Q p_k} \,. \tag{3.27}$$

Uglavnom se minimum ne može egzaktno odrediti, a iterativno nalaženje minimuma je računski skupo. Ipak, dovoljna je i aproksimacija minimuma koja zadovoljava uslove date u glavi 3.5 da bi postupak konvergirao.

Da bismo dobili dužine koraka koje daju brzu konvergenciju izmenićemo algoritam Backtracking Line Search. Izbor koeficijenta smanjenja koraka ρ u iteracijama (linija choose $\rho \in [\rho_{lo}, \rho_{hi}]$) se vrši minimizacijom modela funkcije (3.15) kvadratnim ili kubnim polinomom. Algoritam Line Search u nastavku to radi.

Ova procedura vraća zadovoljenost jednog od tri izlazna kriterijuma kao vrednost *retcode*:

$retcode = 0$, ako α koje vraća zadovoljava uslov (3.16a),

$retcode = 1$, ako postupak dobija za λ vrednosti manje od $\lambda_{\min}$,

$retcode = 2$, ako je potrošena kvota računanje funkcije f,

$retcode = 3$, negativna diskriminanta za minimum kubne interpolacije.

U slučajevima $retcode = 1$, $retcode = 2$, $retcode = 3$ vrednost iteracije x_1 i vrednost funkcije f_1 koje Line Search vraća su nepouzdane, a za $retcode = 0$ se vraćaju da se ne bi ponovo računale u iterativnom postupku.

Ova procedura kreće od vrednosti $\alpha = 1$ koju smanjuje korakom između 0.1 i 0.5, sve dok se ne zadovolji jedan od izlaznih kriterijuma. Korak smanjenja se nalazi kao minimum modela funkcije (3.15) kubnim, a u prvom koraku kvadratnim polinomom.

Nekad iterativni proces blizu rešenja bude usporen zbog malih dužina koraka koje daje ovaj algoritam. Štaviše, u blizini rešenja greška zaokruživanja može da dovede do zaglavljivanja iterativnog procesa (*stall*). Zato se kao ulazni parametar ove procedure daje $\alpha_{\min}$. U slučaju kad algoritam 2 smanji korak α ispod $\alpha_{\min}$ iterativni proces je verovatno došao do rešenja, treba da stane, ali treba i proveriti da li je rešenje traženi minimum. Tada se vraća poruka $retcode = 1$.

Korenovanje negativnog broja može se pojaviti u proceduri Line search kada na vrednost funkcije f utiče šum ili kad se pojavi greška zaokruživanja u determinističkom slučaju. Mi ćemo, kada se pojavi korenovanje negativnog broja, vraćati $retcode = 3$. Obrada ovog povratnog koda će biti ista kao kada iterativni proces zaglavljuje zbog grešaka zaokruživanja blizu rešenja.

Brojanje računanja funkcije se vodi u lokalnoj promenljivi $fcalcls$. Ovu mogućnost nude mnoge implementacije linijskog pretraživanja, na primer u Matlabu. Vrednost $fcalcls$ je jedan od izlaznih parametara ovog algoritma, da bi glavni iterativni program mogao da vodi računa o ukupnoj kvoti računanja funkcije cilja.

Algoritam 2 Line Search

Require: $x_0 \in \mathbb{R}^n$, $f_0 \in \mathbb{R}(=\hat{f}(x_0))$, $f(\cdot) : \mathbb{R}^n \to \mathbb{R}$, $g_0 \in \mathbb{R}^n(=\hat{g}(x_0))$, $p \in \mathbb{R}^n$, $\lambda_{\min} \in \mathbb{R}^+$, $maxfcalcls \in \mathbb{Z}^+$

Ensure: x_1, f_1, $retcode \in \mathbb{Z}^+$, $\lambda \in \mathbb{R}^+$, $fcalcls \in \mathbb{Z}^+$

```
c := 10^{-4};
initslope := g_0^T p;
retcode := 1;
α := 1;
fcalc := 0;
while retcode > 0 do
    x_1 := x_0 + α p;
    f_1 := f(x_1); fcalc := fcalc + 1;
    if f_1 ≤ f_0 + c α initslope then                ▷ Nađeno α
        retcode := 0;
    else if α < α_min then                           ▷ Premalo α zaglavljuje
        x_1 := x_0; f_1 := f_0; retcode := 1;
    else if fcalc ≥ maxfcalc then                    ▷ maksimalni broj računanja f
        x_1 := x_0; f_1 := f_0; retcode := 2;
    else                                             ▷ Smanjujemo α
        if α = 1 then                                ▷ Prvi put kvadratni model
            α_temp := − initslope / (2(f_1 − f_0 − initslope));
        else                                         ▷ Inače kubni model
            [A; B] := 1/(α − α_prev) [1/α², −1/α_prev²; −α_prev/α², α/α_prev²] [f_1 − f_0 − α initslope; f_prev − f_0 − α_prev initslope];
            disc := B² − 3 A initslope;
            if disc < 0 then                         ▷ Šum ili zaokruživanje
                x_1 := x_0; f_1 := f_0; retcode := 3;
            end if
            if A = 0 then                            ▷ Treba pravi kubni model
                α_temp := − initslope / (2B);
            else
                α_temp := (−b + √disc) / (3A);
            end if
            if α_temp > 0.5α then                    ▷ Ne želimo premalo smanjenje
                α_temp := 0.5α;
            end if
        end if
        α_prev := α;
        f_prev := f_1;
        if α_temp ≤ 0.1α then                        ▷ Ni preveliko smanjenje
            α := 0.1α;
        else
            α := α_temp;
        end if
    end if
end while
return
```

4 Njutnov postupak za rešavanje sistema nelinearnih jednačina

Posmatramo problem

$$g(x) = 0, \tag{4.1}$$

gde je $g : \mathbb{R}^n \to \mathbb{R}^n$ vektorska funkcija,

$$g(x) = \left[g_1(x), g_2(x), \ldots, g_n(x)\right]^T,$$

i gde su g_k, $k = 1, 2, \ldots, n$ glatke funkcije.

Vektor x^* koji zadovoljava (4.1) zove se **rešenje** ili koren.

U glavi 1 je data veza sistema linearnih jednačina sa problemom matematičkog programiranja.

Daćemo Njutnov algoritam za sistem linearnih jednačina.

Neka je $g'(x)$ Jakobijan vektorske funkcije $g(x)$.

Algoritam 3 Njutnov algoritam za sistem nelinearnih jednačina

---

Korak 1 Odaberi x^0.

Korak 2 For $k = 0, 1, 2, \ldots$

Korak 3 Reši s^k iz sistema linearnih jednačina $g'(x^k)\, s^k = -g(x^k)$.

Korak 4 Postavi $x^{k+1} = x^k + s^k$.

Korak 5 End For

Da smo primenili algoritam 3 na gradijent $g = \nabla f$ funkcije f iz problema (1.1), dobili bismo Njutnov metod iz glave 3.3. Pri tome je $g' = \nabla^2 f$.

Daćemo bez dokaza teoremu o konvergenciji Njutnovog postupka.

Teorema 7 *Neka je g neprekidno diferencijabilna u konveksnom otvorenom skupu $\mathcal{D} \subseteq \mathbb{R}^n$. Neka je $x^* \in \mathcal{D}$ nedegenerisano rešenje* (4.1) *i neka je $\{x^k\}$ niz generisan algoritmom 3.*

Ako je $x^k \in \mathcal{D}$ dovoljno blizu x^*, onda $\{x^k\}$ superlinearno konvergira ka x^*.

Ako je g Lipšic neprekidno diferencijabilna u okolini x^*, onda je konvergencija kvadratna.

5 Spisak problema za stohastičku optimizaciju

Problemi koje ćemo ovde navesti su klasični problemi za testiranje algoritama za optimizaciju. Neki od njih su veštački kreirani da bi onemogućili nedovoljno robustne algoritme da ih reše.

Osim n i m vrednosti koje su gore opisane, za svaki problem ćemo navesti početnu iteraciju x^0 i optimum x^* ako je poznat.

5.1 Spall's polynomial, [?]

$n = 4$

$$f(x) = x^T A^T Ax + 0.1 \sum_{i=1}^{n} (Ax)_i^3 + 0.01 \sum_{i=1}^{n} (Ax)_i^4$$

gde je A gornja trougaona matrica popunjena jedinicama.
Start: $x^0 = 0.2(1, 1, \dots, 1)$
Optimum: $x^* = (0, 0, \dots 0),\ f(x^*) = 0$

5.2 QuadAppr, [?]

$n = 2$

$$f(x) = 200\, \phi_b(x_1, x_2)/\phi_b(10, 10)$$

gde je

$$\phi_b(x_1, x_2) = \exp\left(0.1\sqrt{x_1^2 + bx_2^2}\right) + \exp\left(-0.1\sqrt{x_1^2 + bx_2^2}\right) - 2.$$

Start: $x^0 = 0.5(1, 1)$
Optimum: $x^* = (0, 0),\ f(x^*) = 0 \ \ (b = 1)$

5.3 Fletcher-Powell helical valley function, [?, ?]

$n = 3$

$$m = 3$$
$$f_1(x) = 10[x_3 - 10\,\theta(x_1, x_2)]$$
$$f_2(x) = 10(\sqrt{x_1^2 + x_2^2} - 1)$$
$$f_3(x) = x_3$$

gde je

$$\theta(x_1, x_2) = \begin{cases} \frac{1}{2\pi} \arctan \frac{x_2}{x_1}, & x_1 > 0 \\ \frac{1}{2} + \frac{1}{2\pi} \arctan \frac{x_2}{x_1}, & x_1 < 0. \end{cases}$$

Start: $x^0 = (-1, 0, 0)$
Optimal: $x^* = (1, 0, 0),\ f(x^*) = 0\ (b = 1).$

5.4 Biggs EXP6 function, [?, ?]

$n = 6$

$$m = 13\ (m \geq n)$$
$$f_i(x) = x_3 \exp(-t_i x_1) - x_4 \exp(-t_i x_2) + x_6 \exp(-t_i x_5) - y_i,$$

gde je

$$t_i = i/10,\ y_i = \exp(-t_i) - 5\exp(-10t_i) + 3\exp(-4t_i),$$

za $i = 1, 2, \ldots, m$.

Start: $x^0 = (1, 2, 1, 1, 1, 1)$

Optimum:

lokalni: $f(x^*) = 5.65565 \cdot 10^{-3}$

globalni: $x^* = (1, 10, 1, 5, 4, 3),\ f(x^*) = 0$.

5.5 Box three-dimensional function, [?, ?]

$n = 3$

$$m = 10\ (m \geq n)$$

$$f_i(x) = \exp(-t_i x_1) - \exp(-t_i x_2) - x_3 \left(\exp(-t_i) - \exp(-10 t_i)\right),$$

gde je

$$t_i = i/10$$

za sve $i = 1, 2, \ldots, m$.

Start: $x^0 = (0, 10, 5)$

Optimum: $f(x^*) = 0$ za $(1, 10, 1)$, $(10, 1, -1)$ i za ($x_1 = x_2$ i $x_3 = 0$).

5.6 The Shekel SQRN5 function [?]

$n = 4$

$$f(x) = -\sum_{i=1}^{m} \left(c_i + \sum_{j=1}^{n} (x_j - a_{i,j})^2 \right)^{-1},$$

gde je

$$m = 5$$

$$[a_{i,j}] = \begin{bmatrix} 4.0 & 4.0 & 4.0 & 4.0 \\ 1.0 & 1.0 & 1.0 & 1.0 \\ 8.0 & 8.0 & 8.0 & 8.0 \\ 6.0 & 6.0 & 6.0 & 6.0 \\ 3.0 & 7.0 & 3.0 & 7.0 \end{bmatrix}, c = [0.1, 0.2, 0.2, 0.4, 0.6]^T.$$

Start: $x^* = (1, 3, 5, 6)$.

Optimum: $x^0 = (4, 4, 4, 4),\ f(x^*) = -10.1527$.

Literatura