

Matematički praktikum

Predavanje 11 – *

prof. dr. sc. Rudolf Scitovski
prof. dr. sc. Kristian Sabo
doc. dr. sc. Danijel Grahovac
dr. sc. Matea Ugrica[†]

12. siječnja 2021.

Sadržaj

1	Višedimenzionalna globalna optimizacija	2
2	Transformacija problema na hiperkocku	3
3	Optimizacijski algoritam DIRECT za funkciju dviju varijabli	4
3.1	Dijeljenje kvadrata $[0, 1]^2$	5
3.2	Nekoliko primjera	7
4	Optimizacijski algoritam DIRECT za funkciju više varijabli	10
5	Optimizacijski algoritam DIRECT za simetričnu funkciju	11
6	Neke druge mogućnosti	15
6.1	Kratki povijesni pregled	15
6.2	Metoda grananja i ograđivanja	16

*Matematički praktikum obavezni je predmet u zimskom semestru druge godine sveučilišnog Diplomskog studija matematike na smjerovima Financijska matematika i statistika i Računarstvo, te na petoj godini sveučilišnog Nastavničkog studija matematike i informatike (30 sati predavanja i 30 sati vježbi, 6 ECTS bodova)

[†]scitowsk@mathos.hr, ksabo@mathos.hr, dgrahova@mathos.hr, mugrica@mathos.hr

1 Višedimenzionalna globalna optimizacija

Kao što smo spomenuli ranije u uvodu, problem globalne optimizacije pojavljuje se u različitim primjenama u gotovo svim područjima znanosti. Pri tome funkcija cilja može biti linearna, nelinearna, konveksna, nekonveksna, diferencijabilna, nediferencijabilna, može imati jednu ili više nezavisnih varijabli. Područje na kojemu je definirana funkcija može biti čitav skup \mathbb{R}^n (tada govorimo o problemu globalne optimizacije bez ograničenja), a može biti neki podskup u \mathbb{R}^n (tada govorimo o problemu globalne optimizacije s ograničenjima). Općenito, problem globalne optimizacije definirat ćemo na sljedeći način:

Treba odrediti točku $u^* \in \mathcal{D} \subseteq \mathbb{R}^n$ u kojoj funkcija $g: \mathcal{D} \rightarrow \mathbb{R}$ postiže globalni minimum $v^* = g(u^*) \in \mathbb{R}$, odnosno, treba odrediti točku $u^* \in \mathcal{D} \subseteq \mathbb{R}^n$ takvu da je

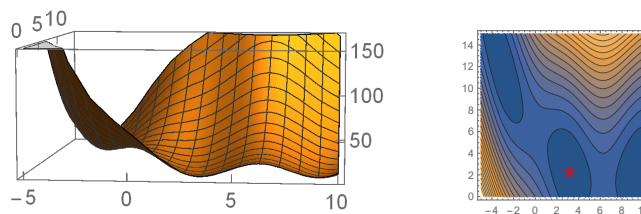
$$u^* \in \underset{u \in \mathcal{D}}{\operatorname{argmin}} g(u), \quad v^* = g(u^*). \quad (1)$$

Može se dogoditi da je skup $\underset{u \in \mathcal{D}}{\operatorname{argmin}} g(u)$ jednočlan (kao u Primjeru ??, str.??) ili višečlan (kao u Primjeru 1).

Primjer 1. Poznata Branin funkcija (koja se kao test primjer razmatra u brojnim radovima [4, 5, 11]) $g: [-5, 10] \times [0, 15] \rightarrow \mathbb{R}$,

$$g(x_1, x_2) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$$

postiže globalni minimum $g_{\min} = 0.398$ u tri različite točke. Jedna od njih je $u_1^* = (\pi, 2.275)$ (vidi Sliku 1).



Slika 1: Branin test funkcija [11]

U nastavku ćemo pretpostaviti da je domena funkcije g koju treba minimizirati hiperpravokutnik $\mathcal{D} = [a_1, b_1] \times \cdots \times [a_n, b_n]$ za neke $a_i < b_i$, $i = 1, \dots, n$. Prije pokretanja nekog optimizacijskog algoritma funkciju g najprije je korisno transformirati na jediničnu hiperkocku $[0, 1]^n$. U Primjeru 1 to znači da treba odrediti preslikavanje koje pravokutnik $[-5, 10] \times [0, 15]$ prevodi u kvadrat $[0, 1] \times [0, 1]$.

2 Transformacija problema na hiperkocku

U cilju transformacije optimizacijskog problema (1) na standardnu formu za funkciju $f: [0, 1]^n \rightarrow \mathbb{R}$

$$u^* \in \underset{u \in [0, 1]^n}{\operatorname{argmin}} f(u), \quad v^* = f(u^*), \quad (2)$$

potrebno je napraviti preslikavanje hiperpravokutnika $\mathcal{D} = \prod_{i=1}^n [a_i, b_i]$ na hiperkocku $[0, 1]^n$. Zbog jednostavnosti, u sljedećoj lemi pokazujemo kako se to može napraviti kada je $\mathcal{D} \subset \mathbb{R}^2$. Općeniti rezultat dobiva se analogno (vidi Zadatak 2).

Lema 1. *Neka su $a, b, c, d \in \mathbb{R}$ takvi da je $b > a$ i $d > c$. Jedinstvena linearna bijekcija T koja pravokutnik $[a, b] \times [c, d]$ preslikava u jedinični kvadrat $[0, 1]^2$ i pri tome vrijedi: $T(a, c)^T = (0, 0)^T$, $T(a, d)^T = (0, 1)^T$, $T(b, c)^T = (1, 0)^T$, $T(b, d)^T = (1, 1)^T$ zadana je s*

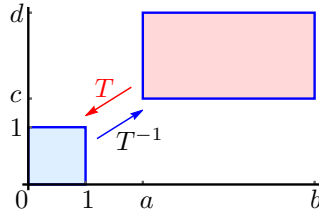
$$T(x) = A(x - u), \quad A = \begin{bmatrix} \frac{1}{b-a} & 0 \\ 0 & \frac{1}{d-c} \end{bmatrix}, \quad u = \begin{bmatrix} a \\ c \end{bmatrix}, \quad (3)$$

a inverzno preslikavanje $T^{-1}: [0, 1]^2 \rightarrow [a, b] \times [c, d]$ s

$$T^{-1}(x) = A^{-1}x + u. \quad (4)$$

Dokaz. Potražimo preslikavanje T u obliku $T(x) = Ax + \omega$, gdje je

$$A = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}, \quad \omega = \begin{bmatrix} \xi \\ \eta \end{bmatrix}.$$



Slika 2: Preslikavanje pravokutnika $[a, b] \times [c, d]$ na jedinični kvadrat $[0, 1]^2$

Po uvjetu treba biti

$$\begin{aligned} T(a, c)^T = (0, 0)^T &\Rightarrow \begin{cases} a\alpha + c\beta + \xi = 0 \\ a\gamma + c\delta + \eta = 0 \end{cases} \\ T(b, c)^T = (1, 0)^T &\Rightarrow \begin{cases} b\alpha + c\beta + \xi = 1 \\ b\gamma + c\delta + \eta = 0 \end{cases} \\ T(a, d)^T = (0, 1)^T &\Rightarrow \begin{cases} a\alpha + d\beta + \xi = 0 \\ a\gamma + d\delta + \eta = 1 \end{cases} \\ T(b, d)^T = (1, 1)^T &\Rightarrow \begin{cases} b\alpha + d\beta + \xi = 1 \\ b\gamma + d\delta + \eta = 1, \end{cases} \end{aligned}$$

iz čega slijedi

$$\alpha = \frac{1}{b-a}, \quad \beta = 0, \quad \gamma = 0, \quad \delta = \frac{1}{d-c}, \quad \xi = \frac{-a}{b-a}, \quad \eta = \frac{-c}{d-c},$$

odnosno

$$A = \begin{bmatrix} \frac{1}{b-a} & 0 \\ 0 & \frac{1}{d-c} \end{bmatrix}, \quad \omega = \begin{bmatrix} \frac{-a}{b-a} \\ \frac{-c}{d-c} \end{bmatrix}.$$

Kako je $\omega = -A \begin{bmatrix} a \\ c \end{bmatrix}$, onda uz oznaku $u := \begin{bmatrix} a \\ c \end{bmatrix}$ slijedi (3).

Formula (4) dokazuje se direktnom provjerom. \square

Transformacija Branin funkcije iz Primjera 1 na kvadrat $[0, 1]^2$ daje funkciju $f: [0, 1]^2 \rightarrow \mathbb{R}$, gdje je $f = g \circ T^{-1}$, $T^{-1}(x) = A^{-1}x + u$, $A^{-1} = \begin{bmatrix} 15 & 0 \\ 0 & 15 \end{bmatrix}$, $u = [-5, 0]^T$.

Zadatak 1. Neka je $g: [a, b] \times [c, d] \rightarrow \mathbb{R}$ Lipschitz-neprekidna funkcija s konstantom $L > 0$ i $T: [a, b] \times [c, d] \rightarrow [0, 1]^2$, $T(x) = A(x - u)$, $A = \begin{bmatrix} \frac{1}{b-a} & 0 \\ 0 & \frac{1}{d-c} \end{bmatrix}$, $u = \begin{bmatrix} a \\ c \end{bmatrix}$. Pokažite da je i $f = g \circ T^{-1}: [0, 1]^2 \rightarrow \mathbb{R}$ također Lipschitz neprekidna funkcija i odredite odgovarajuću Lipschitzovu konstantu.

Uputa: $|f(x) - f(y)| = |g(T^{-1}(x)) - g(T^{-1}(y))|$.

Zadatak 2. Neka su a_i, b_i , $i = 1, \dots, n$, ($n \geq 2$), realni brojevi takvi da je $b_i > a_i$, $\forall i = 1, \dots, n$. Definirajte preslikavanje $T: \mathbb{R}^n \rightarrow \mathbb{R}^n$, koje hiperpravokutnik $\prod_{i=1}^n [a_i, b_i]$ preslikava u jedinični hiperkvadrat $[0, 1]^n$.

Primjedba 1. Primijetite da se hiperpravokutnik u \mathbb{R}^n može zapisati na sljedeći način. Za dvije različite točke $a, b \in \mathbb{R}^n$ hiperpravokutnik $[a, b] \subset \mathbb{R}^n$ može se zapisati s $[a, b] = \{x \in \mathbb{R}^n: a_i \leq x_i \leq b_i, i = 1, \dots, n\}$ ili jednostavnije s $[a, b] = \{x \in \mathbb{R}^n: a \leq x \leq b\}$.

3 Optimizacijski algoritam *DIRECT* za funkciju dviju varijabli

Zadana je Lipschitz neprekidna funkcija $g: [a, b] \times [c, d] \rightarrow \mathbb{R}$. Primjenom transformacije T zadane s (3), umjesto rješavanja optimizacijskog problema za funkciju g rješavat ćemo optimizacijski problem za funkciju

$$f = g \circ T^{-1}: [0, 1]^2 \rightarrow \mathbb{R}, \quad \text{gdje je} \quad T^{-1}(x) = A^{-1}x + u. \quad (5)$$

Uočite da će i tako definirana funkcija f biti Lipschitz neprekidna (vidi Zadatak ?? i Zadatak ?? (e)).

Ako je $\hat{x} \in \operatorname{argmin}_{x \in [0, 1]^2} f(x)$ točka globalnog minimuma funkcije f dobivena *DIRECT* algoritmom, onda je $x^* = T^{-1}(\hat{x}) \in \operatorname{argmin}_{x \in [a, b] \times [c, d]} g(x)$ i $g(x^*) = f(T(x^*)) = f(\hat{x})$.

3.1 Dijeljenje kvadrata $[0, 1]^2$

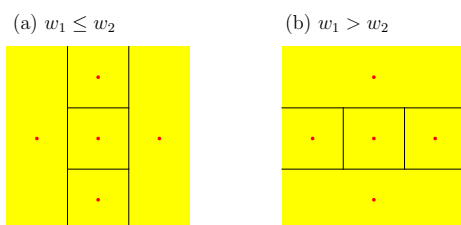
Slično kao što smo u t.??, str.?? definirali strategiju dijeljenja intervala, u ovom slučaju potrebno je razraditi strategiju dijeljenja kvadrata. Jedinični kvadrat $[0, 1]^2$ iterativnim postupkom dijelit ćemo na subpravokutnike prema niže navedenom postupku i pravilima.

Postupak počinje dijeljenjem jediničnog kvadrata $[0, 1]^2$ s centrom u točki $c = \left(\frac{1}{2}, \frac{1}{2}\right)$ na način opisan na Slici 3a ako je

$$w_1 := \min\{f(c - \delta e_1), f(c + \delta e_1)\} \leq \min\{f(c - \delta e_2), f(c + \delta e_2)\} =: w_2, \quad (6)$$

gdje je $\delta = \frac{1}{3}$, a e_1 i e_2 standardni bazni vektori u \mathbb{R}^2 , odnosno na način opisan na Slici 3b ako je

$$w_1 > w_2. \quad (7)$$



Slika 3: Dijeljenje kvadrata $[0, 1] \times [0, 1]$

Tako dobivamo skup od pet subpravokutnika $\mathcal{P} = \{P_1, \dots, P_5\}$ s centrima

$$\begin{aligned} c_1 &= \left(\frac{1}{2}, \frac{1}{2}\right), & c_2 &= \left(\frac{1}{2} - \delta, \frac{1}{2}\right), & c_3 &= \left(\frac{1}{2} + \delta, \frac{1}{2}\right), \\ c_4 &= \left(\frac{1}{2}, \frac{1}{2} - \delta\right), & c_5 &= \left(\frac{1}{2}, \frac{1}{2} + \delta\right). \end{aligned}$$

Također, svakom subpravokutniku pridružujemo duljine njegovih stranica u slučaju (6) (Slika 3a)

$$h^{(1)} = (\delta, \delta), \quad h^{(2)} = (\delta, 1), \quad h^{(3)} = (\delta, 1), \quad h^{(4)} = (\delta, \delta), \quad h^{(5)} = (\delta, \delta),$$

odnosno duljine njegovih stranica u slučaju (7) (Slika 3b)

$$h^{(1)} = (\delta, \delta), \quad h^{(2)} = (\delta, \delta), \quad h^{(3)} = (\delta, \delta), \quad h^{(4)} = (1, \delta), \quad h^{(5)} = (1, \delta).$$

Ako je $L > 0$ Lipschitzova konstanta funkcije f , \mathcal{B} -vrijednost pravokutnika P_i s centrom c_i i veličinom d_i definiramo s $\mathcal{B}_i = \mathcal{B}(c_i, d_i) = f(c_i) - Ld_i$, gdje d_i predstavlja neku mjeru veličine pravokutnika P_i . Budući da je cilj pronaći i dalje dijeliti subpravokutnik s najmanjom \mathcal{B} -vrijednosti, kao mjeru veličine subpravokutnika P_i sa stranicama $h_1^{(i)}, h_2^{(i)}$ treba uzeti

$$d_i = \max\{h_1^{(i)}, h_2^{(i)}\}, \quad i = 1, \dots, 5. \quad (8)$$

Naime, ako je primjerice $h_1^{(i)} > h_2^{(i)}$, onda se manja \mathcal{B} -vrijednost dobiva korištenjem dulje stranice

$$\mathcal{B}_1^{(i)} = f(c) - Lh_1^{(i)} < f(c) - Lh_2^{(i)} = \mathcal{B}_2^{(i)}.$$

Tako u slučaju (6) (Slika 3a) imamo

$$d_1 = \delta, \quad d_2 = 1, \quad d_3 = 1, \quad d_4 = \delta, \quad d_5 = \delta,$$

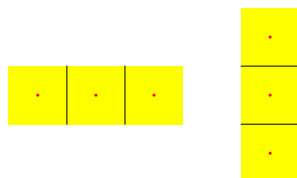
a u slučaju (7) (Slika 3b) imamo

$$d_1 = \delta, \quad d_2 = \delta, \quad d_3 = \delta, \quad d_4 = 1, \quad d_5 = 1.$$

Primjedba 2. Primijetite da je \mathcal{B} -vrijednost svakog pravokutnika P_i određena vrijednošću funkcije u centru c_i i duljinom d_i njegove veće stranice. Također je važno primijetiti da smo za mjeru veličine pravokutnika mogli uzeti i nešto drugo (primjerice, udaljenost centra do vrha, tj. duljinu poludijagonale) – to nam dozvoljavaju zapisi uvjeta u Lemi ??!. Ova je činjenica važna ako područje dijelimo na trokute [16]. Tada bismo za d_i uzimali maksimalnu udaljenost od centra (težišta) trokuta do vrha.

Nadalje, svaki subkvadrat (jednake stranice) dijelit ćemo također prema prethodno opisanom postupku, a svaki subpravokutnik (različite stranice) na tri jednaka dijela po većoj stranici kao na Slici 4.

Na taj način u nekom trenutku raspolažemo skupom \mathcal{P} od m subpravokutnika. Za svaki od njih odredit ćemo njegov centar c_i i veličinu d_i kao poluduljinu veće stranice.



Slika 4: Dijeljenje subpravokutnika

Trenutnu aproksimaciju \hat{c} točke globalnog minimuma odredit ćemo kao

$$\hat{c} = \underset{i=1, \dots, m}{\operatorname{argmin}} f(c_i), \quad f_{\min} = f(\hat{c}). \quad (9)$$

Uočimo da problemu traženja subpravokutnika minimalne \mathcal{B} -vrijednosti možemo pristupiti jednako kao u t.??, str.?? i promatrati potencijalno optimalne subpravokutnike definirane analogno Definiciji ??, str.?. Korištenjem rezultata Teorema ??, str.?? možemo na taj način iz skupa \mathcal{P} izdvojiti *potencijalno optimalne subpravokutnike* i samo na njima provesti daljnji postupak dijeljenja. Time smo izbjegli korištenje Lipschitzove konstante funkcije f u algoritmu.

Tijekom iterativnog postupka skup \mathcal{P} proširuje se novim subpravokutnicima, a minimum f_{\min} korigira se uvijek samo na centrima novih subpravokutnika.

Problem zaustavljanja iterativnog postupka u literaturi nije egzaktno riješen. Jedna mogućnost za zaustavljanje iterativnog postupka je kontroliranje trenutne veličine d subpravokutnika na kojemu se postiže trenutni minimum.

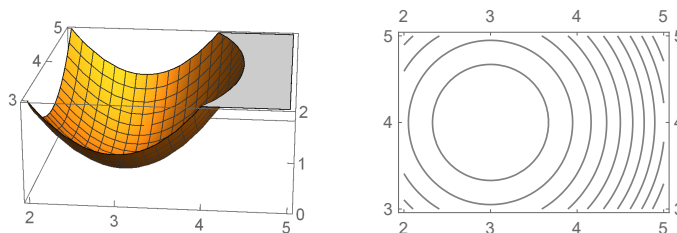
Nakon što odredimo prihvatljivu aproksimaciju $\{\hat{c}, f_{min}\}$ globalnog minimuma funkcije f , primjenom inverznog preslikavanja T^{-1} iz Leme 1, str.3, aproksimaciju globalnog minimuma polazne funkcije g određujemo kao $T^{-1}(\hat{c})$.

Opisani algoritam formaliziran je u Algoritmu 1, str.11 za funkciju n nezavisnih varijabli.

3.2 Nekoliko primjera

Korištenjem programskog sustava *Mathematica* s nekoliko primjera ilustrirat ćemo *Algoritam DIRECT* za traženje globalnog minimuma funkcije dviju varijabli uz odgovarajući postupak dijeljenja jediničnog kvadrata $[0, 1]^2$.

Primjer 2. Za početak promatrajmo sasvim jednostavnu kvadratnu funkciju $g: [2, 5] \times [3, 5] \rightarrow \mathbb{R}$, $g(x_1, x_2) = (x_1 - 3)^2 + (x_2 - 4)^2$, koja svoj globalni minimum postiže u točki $u^* = (3, 4)$, $g(u^*) = 0$.

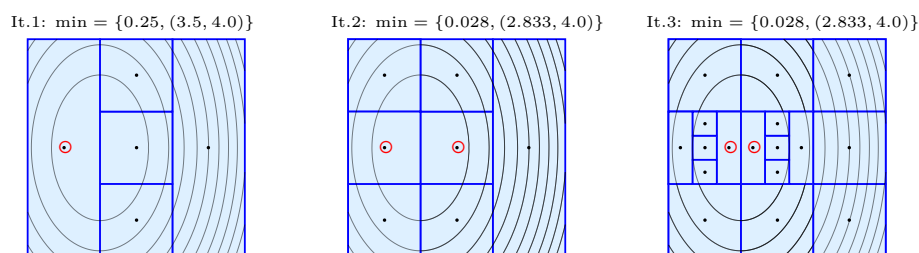


Slika 5: Kvadratna funkcija $g: [2, 5] \times [3, 5] \rightarrow \mathbb{R}$, $g(x_1, x_2) = (x_1 - 3)^2 + (x_2 - 4)^2$

Ploha i odgovarajući `ContourPlot` ove funkcije prikazani su na Slici 5. Primjenom preslikavanja $T(x) = A(x - u)$, gdje je $A = \begin{bmatrix} 1/3 & 0 \\ 0 & 1/2 \end{bmatrix}$, $u = [2, 3]^T$, $T^{-1}(x) = A^{-1}(x) + u$, dobivamo funkciju $f: [0, 1]^2 \rightarrow \mathbb{R}$, $f(x) = (g \circ T^{-1})(x) = g(3x_1 + 2, 2x_2 + 3) = (3x_1 - 1)^2 + (2x_2 - 1)^2$, koja svoj globalni minimum postiže u točki $\hat{u} = (\frac{1}{3}, \frac{1}{2})$, $f(\hat{u}) = 0$.

Globalni minimum originalne funkcije g postiže se u točki $u^* = T^{-1}(\hat{u}) = [3, 4]^T$, a vrijednost funkcije g u toj točki je $g(u^*) = f(T(u^*)) = f(\hat{u}) = 0$.

Prve tri iteracije dijeljenja jediničnog kvadrata $[0, 1]^2$ za ovu funkciju te odgovarajući trenutni minimum prikazani su na Slici 6. Crvenim kružićima naznačeni su centri potencijalno optimalnih subpravokutnika.

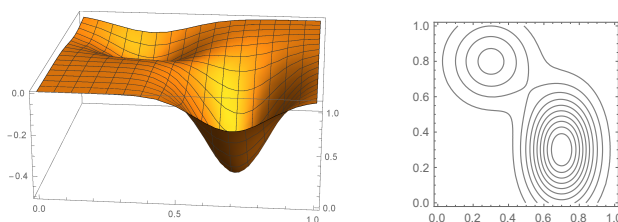


Slika 6: Dijeljenje kvadrata $[0, 1]^2 \subset \mathbb{R}^2$ u slučaju funkcije iz Primjera 2. Crvenim kružićima označeni su centri potencijalno optimalnih pravokutnika

Primjer 3. *Konstruirajmo funkciju $f: [0, 1]^2 \rightarrow \mathbb{R}$,*

$$f(x_1, x_2) = -\frac{1}{2}e^{-10\|A[x_1, x_2]^T - [7, 3]^T\|^2} - \frac{1}{5}e^{-20\|[x_1, x_2]^T - [0.3, 0.8]^T\|^2}, \quad (10)$$

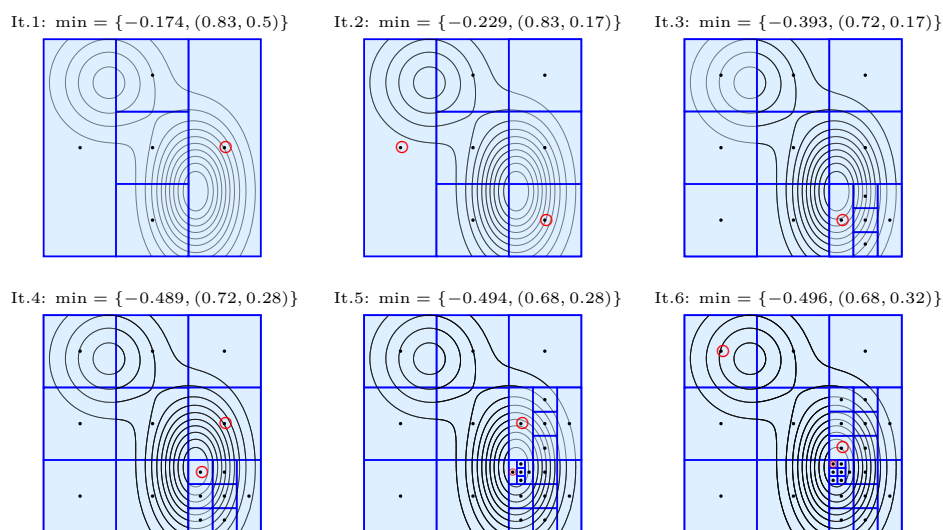
gdje je $A = .8 \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$, koja će na $[0, 1]^2$ postići lokalni minimum u točki $u_1^* = (0.3, 0.8)$, $f(u_1^*) = -0.2$ i globalni minimum u točki $u_2^* = (0.7, 0.3)$, $f(u_2^*) = -0.5$.



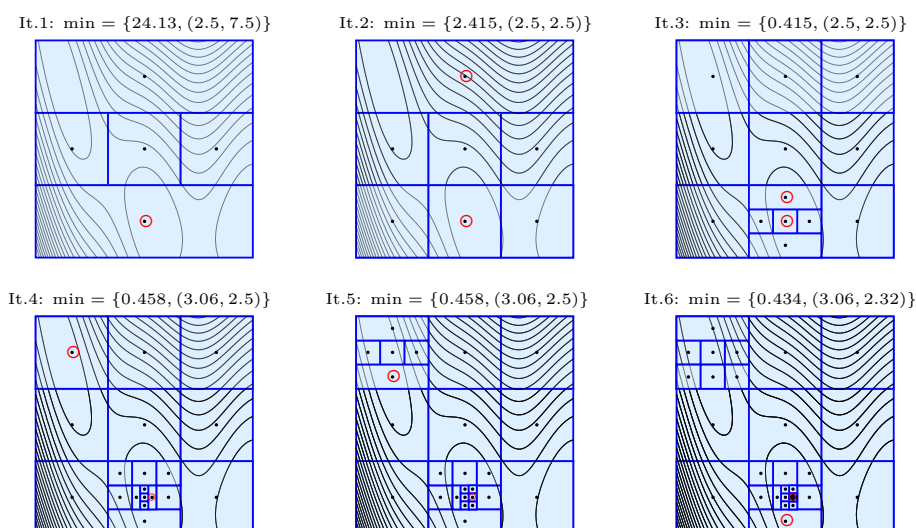
Slika 7: $f(x_1, x_2) = -\frac{1}{2}e^{-10\|A[x_1, x_2]^T - [7, 3]^T\|^2} - \frac{1}{5}e^{-20\|[x_1, x_2]^T - [0.3, 0.8]^T\|^2}$

Ploha i odgovarajući *ContourPlot* prikazani su na Slici 7. Prve tri iteracije dijeljenja jediničnog kvadrata $[0, 1]^2$ za ovu funkciju te odgovarajući trenutni minimum prikazani su na Slici 8.

Zadatak 3. *Pokažite da funkcija f zadana s (10) ima dvije stacionarne točke $u_1^* = (0.3, 0.8)$ i $u_2^* = (0.7, 0.3)$ i da je $f(u_1^*) = -0.2$, $f(u_2^*) = -0.5$.*



Slika 8: Dijeljenje kvadrata $[0, 1]^2 \subset \mathbb{R}^2$ u slučaju funkcije (10) iz Primjera 3. Crvenim kružićima označeni su centri potencijalno optimalnih pravokutnika



Slika 9: Dijeljenje kvadrata $[0, 1]^2 \subset \mathbb{R}^2$ u slučaju Branin funkcije iz Primjera 1. Crvenim kružićima označeni su centri potencijalno optimalnih pravokutnika

Primjer 4. Pokažimo još primjenu optimizacijskog Algoritma *DIRECT* za Branin funkciju iz Primjera 1, koja svoj globalni minimum $g_{min} = 0.398$ postiže u trima različitim točkama. Jedna od njih je $u_1^* = (\pi, 2.275)$ (vidi Sliku 1).

Prvih šest iteracija dijeljenja jediničnog kvadrata $[0, 1]^2$ za ovu funkciju te odgovarajući trenutni minimum prikazani su na Slici 9.

Zadatak 4. Zadana je funkcija $g: [-1, 1] \times [-1, 1] \rightarrow \mathbb{R}$,

$$g(x_1, x_2) = \max \{|2x_1 - x_2 - 1|, |x_1 + x_2|\}.$$

- (a) Odredite globalni minimizator funkcije g .
- (b) Odredite funkciju f dobivenu transformacijom funkcije g na jedinični kvadrat. Što je globalni minimizator tako dobivene funkcije?
- (c) Napravite dvije iteracije Algoritma *DIRECT* te skicirajte kako bi se dalje dijelili pravokutnici.
- (d) Koja je aproksimacija globalnog minimizatora polazne funkcije g ?

4 Optimizacijski algoritam *DIRECT* za funkciju više varijabli

Zadana je Lipschitz neprekidna funkcija $g: \mathcal{D} \rightarrow \mathbb{R}$, gdje je $\mathcal{D} \subset \mathbb{R}^n$ hiperpravokutnik u \mathbb{R}^n . Primjenom transformacije T (Zadatak 2, str.4), umjesto rješavanja optimizacijskog problema za funkciju g rješavat ćemo optimizacijski problem za funkciju

$$f = g \circ T^{-1}: [0, 1]^n \rightarrow \mathbb{R}, \quad \text{gdje je } T^{-1}(x) = A^{-1}x + u. \quad (11)$$

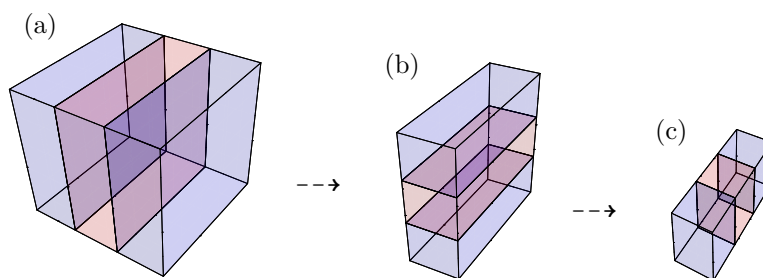
Slično kao što smo u t.??, str.?? definirali strategiju dijeljenja intervala, u ovom slučaju potrebno je razraditi strategiju dijeljenja hiperkocke. Jediničnu hiperkocku $[0, 1]^n$ iterativnim ćemo postupkom dijeliti na subhiperpravokutnike prema niže navedenom postupku i pravilima.

Primjedba 3. Primijetite da u slučaju $n = 3$ kocku dijelimo na $2n + 1$ subparalelepipeda na sljedeći način (vidi [11], Slika 8, str. 169):

- po prvoj izabranoj koordinati (izbor ovisi o brojevima w_1, w_2, w_3 koji su definirani analogno (6)) kocku dijelimo na 3 jednaka subparalelepipeda (sloja) (vidi Sliku 10a);
- po drugoj izabranoj koordinati centralni subparalelepiped (sloj) dijelimo na 3 jednaka subparalelepipeda (sloja) (vidi Sliku 10b);
- po trećoj koordinati centralni subparalelepiped (sloj) dijelimo na 3 jednake subkocke (vidi Sliku 10c).

Nadalje, uvijek samo kocku dijelimo na ovaj način. Subparalelepipede, kakvi god bili, uvijek dijelimo na tri jednaka dijela po najdužoj stranici.

Niže navodimo pseudokod *Algoritma DIRECT*, a odgovarajući *Mathematica*-modul može se preuzeti na adresi <http://www.mathos.unios.hr/GOP-knjiga/GOPmoduli.rar>

Slika 10: Dijeljenje kocke u \mathbb{R}^3 **Algorithm 1 (Algoritam DIRECT)****Require:** $f: \mathbb{R}^n \rightarrow \mathbb{R}$;**Require:** c_{min} ; $d = 0.5$; $0 < \eta < d$; $f_{min} = f(c_{min})$;

- 1: Neka je \mathcal{H} trenutni skup hiperpravokutnika;
 - 2: Grupiraj sve hiperpravokutnike iz \mathcal{H} prema njihovoj veličini, a u svakoj grupi sortiraj ih prema vrijednosti funkcije f u njihovom centru;
 - 3: Hiperpravokutnici iz svake grupe s najmanjom funkcijskom vrijednosti čine prošireni skup potencijalno optimalnih hiperpravokutnika \mathcal{E} ;
 - 4: Primjenom Teorema ?? iz skupa \mathcal{E} , sagradi skup potencijalno optimalnih hiperpravokutnika \mathcal{P} ;
 - 5: **for** $R \in \mathcal{P}$ **do**
 - 6: Podijeli R na subhiperpravokutnike r_1, \dots, r_s ;
 - 7: Izbriši hiperpravokutnik R
 - 8: **for** $i = 1, \dots, s$; **do**
 - 9: Odredi veličinu d_i , centar c_i i funkcijsku vrijednost $f(c_i)$;
 - 10: Korigiraj minimalnu veličinu d i $(c_{min}, f(c_{min}))$
 - 11: **end for**
 - 12: **end for**
 - 13: **if** $d > \eta$, **go to** Step 2; **else STOP**
- Ensure:** $(c_{min}, f(c_{min}))$

U Primjeru ??, str.?? već smo spomenuli primjenu algoritma DIRECT kod segmentacije crno-bijele slike uz primjenu inkrementalnog algoritma. Brojne druge primjere problema globalne optimizacije i test funkcije za globalnu optimizaciju možemo pronaći u [4–6, 10, 17, 19].

5 Optimizacijski algoritam DIRECT za simetričnu funkciju

Funkcija $f: [a, b]^n \rightarrow \mathbb{R}$ je simetrična ako proizvoljna permutacija njenih nezavisnih varijabli ne mijenja vrijednost funkcije. Simetrične funkcije često se javljaju u različitim primijenjenim istraživanjima. Primjerice, funkcija (??) simetrična je funkcija jer proizvoljna permutacija nezavisnih varijabli c_1, \dots, c_n znači samo drugačiji poredak centara

klastera, ali ne mijenja vrijednost funkcije.

Primjer 5. Funkcija $f: \mathbb{R}^2 \rightarrow \mathbb{R}$, $f(x_1, x_2) = x_1^2 - x_1x_2 + x_2^2$ simetrična je funkcija, ali funkcija $g: \mathbb{R}^2 \rightarrow \mathbb{R}$, $g(x_1, x_2) = x_1 - x_1x_2 + x_2^2$ nije.

Zadatak 5. Primjenom programskog sustava Mathematica nacrtajte graf (plohu) i `ContourPlot` funkcije [6]

$$g: [0, 11]^2 \rightarrow \mathbb{R}, \quad g(x, y) = -.2(x^2 + y^2) + 2xy \cos x \cos y.$$

Je li ovo simetrična funkcija?

Primjer 6. Neka je $\mathcal{A} \subset \mathbb{R}$ skup dobiven Mathematica-naredbom:

```
In[1]:= m = 20; SeedRandom[13];
        A = RandomReal[{0, 1}, m]
```

```
Out[1]= {0.456535, 0.868230, 0.704274, 0.795001, 0.040520, 0.957827, 0.008372,
         0.251257, 0.014313, 0.743946, 0.066294, 0.783009, 0.907372, 0.081007,
         0.486618, 0.824774, 0.684515, 0.063848, 0.086283, 0.658425}
```

Elemente skupa \mathcal{A} označit ćemo s a^i , $i = 1, \dots, m$ i grupirati ih u dva klastera minimizacijom funkcije (??).

Područje definicije minimizirajuće funkcije (??) je jedinični kvadrat $[0, 1]^2$, a za traženje globalnog minimuma direktno se može primijeniti Algoritam 1 ili Mathematica-modul `NMinimize`. Dobivamo

```
In[1]:= Clear[x, y];
        F[x_, y_] := Sum[Min[Norm[x - A[[i]]]^2, Norm[y - A[[i]]]^2], {i, m}]
        NMinimize[{F[x, y], 0 <= x <= 1, 0 <= y <= 1}, {x, y}]
```

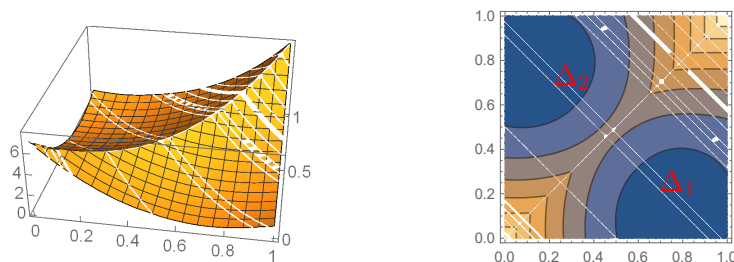
```
Out[2]= {0.300262, {x -> 0.739211, y -> 0.0764867}}
```

Graf (ploha) i `ContourPlot` odgovarajuće funkcije (??) prikazani su na Slici 11. Primijetite da je područje definicije simetrično podijeljeno na dva trokuta

$$\Delta_1 = \{(x_1, x_2) \in [0, 1]^2: 1 \geq x_1 \geq x_2 \geq 0\},$$

$$\Delta_2 = \{(x_1, x_2) \in [0, 1]^2: 0 \leq x_1 \leq x_2 \leq 1\},$$

a Mathematica-modul `NMinimize` pronašao je samo točku globalnog minimuma $u^* = (0.73921, 0.07648)$ u trokutu Δ_1 . Jasno je da postoji i druga točka globalnog minimuma $v^* = (0.07648, 0.73921) \in \Delta_2$. Kako treba korigirati program?



Slika 11: $F(c_1, c_2) = \sum_{i=1}^m \min\{\|c_1 - a^i\|_2^2, \|c_2 - a^i\|_2^2\}$

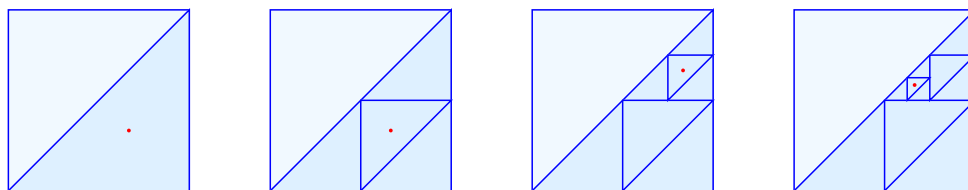
Situacija koja se pojavila u prethodnom primjeru prirodno se generalizira i u slučaju simetrične Lipschitz-neprekidne funkcije više varijabli. Ako simetrična Lipschitz-neprekidna funkcija s n nezavisnih varijabli ima globalni minimum, tada se on može postići u barem $n!$ različitih točaka. Primjerice, ako bismo skup $\mathcal{A} \subset \mathbb{R}$ grupirali u 5 klastera, funkcija (??) postigla bi svoj globalni minimum u barem 120 različitih točaka, čije se koordinate razlikuju do na permutaciju. Pet koordinata bilo koje od ovih točaka predstavljaju centre pet klastera skupa \mathcal{A} . Algoritam *DIRECT* pronalazi svih 120 točaka i zbog toga je vrijeme izvođenja (CPU time) izuzetno veliko.

Zato se u literaturi [6] pojavila specijalna metoda i algoritam *SymDIRECT* kojim se efikasno rješava globalni optimizacijski problem (GOP) za simetričnu Lipschitz-neprekidnu funkciju. Metoda *SymDIRECT* jedna je modifikacija standardnog Algoritma *DIRECT*, a najkraće rečeno, umjesto traženja točke globalnog minimuma na hiperkocki $[0, 1]^n$, on se traži samo na jednom od $n!$ jednakih hipertetraedara. Zbog jednostavnosti izabran je hipertetraedar

$$\Delta = \{(x_1, \dots, x_n) \in [0, 1]^n : 1 \geq x_1 \geq \dots \geq x_n \geq 0\}.$$

U postupku dijeljenja hiperkocke $[0, 1]^n$ Algoritam *SymDIRECT* zadržava samo one subhiperpravokutnike koji su potpuno ili djelomično sadržani u hipertetraedru Δ . Metoda pokazuje visoku efikasnost kod rješavanja GOP za simetričnu Lipschitz-neprekidnu funkciju.

Godinu dana kasnije pojavila se nova, još efikasnija metoda i algoritam *DISIMPL* [16]. Ideja ove metode sastoji se u tome da umjesto dijeljenja hiperkocke $[0, 1]^n$ posebnim postupkom dijelimo hipertetraedar Δ . U najjednostavnijem slučaju ($n = 2$), hipertetraedar Δ donji je trokut jediničnog kvadrata $[0, 1]^2$ (Slika 12a), koji se u prvom koraku dijeli na 4 jednaka (sukladna) subtrokuta od kojih središnji ima centar (težište) u centru polaznog trokuta, a njegovi vrhovi podudaraju se s polovištima stranica polaznog trokuta. Nakon određivanja \mathcal{B} -vrijednosti svih subtrokuta, dalje se dijeli onaj s najmanjom \mathcal{B} -vrijednosti. Naravno, i u ovom slučaju koriste se potencijalno optimalni subtrokuti. Jedno moguće dijeljenje trokuta Δ prikazano je na Slici 12.

Slika 12: Dijeljenje trokuta $\Delta = \{(x_1, x_2) \in [0, 1]^2 : 1 \geq x_1 \geq x_2 \geq 0\}$

Primjer 7. Neka je $\mathcal{A} = \{a^i \in [0, 1] : i = 1, \dots, m\} \subset \mathbb{R}$ skup konstruiran u Primjeru 6. Elemente skupa \mathcal{A} grupirat ćemo u tri klastera minimizacijom funkcije (??).

Primjenom *Mathematica*-modula `NMinimize` dobivamo

```
In[1]:= Clear[x, y, z];
        NMinimize[{F[x,y,z], 0<=x<=1, 0<=y<=1, 0<=z<=1}, {x,y,z}]

Out[2]= {0.125808, {x -> 0.792737, y -> 0.0515195, z -> 0.398137}}
```

Na ovaj GOP primijenit ćemo i *Algoritam SymDIRECT*. Područje definicije minimizirajuće funkcije (??) u ovom je slučaju jedinična kocka $[0, 1]^3$ koja je prirodno podijeljena na 6 jednakih tetraedara (vidi Sliku 13¹). Sukladno *Metodi SymDIRECT* dovoljno je promatrati samo jedan od 6 tetraedara i na njemu pronaći točke minimuma. Promatrajmo tetraedar

$$\Delta_1 = \{(x_1, x_2, x_3) \in [0, 1]^3 : 1 \geq x_1 \geq x_2 \geq x_3 \geq 0\}.$$

Slika 13: Jedinična kocka $[0, 1]^3$ prirodno je podijeljena na 6 jednakih tetraedara

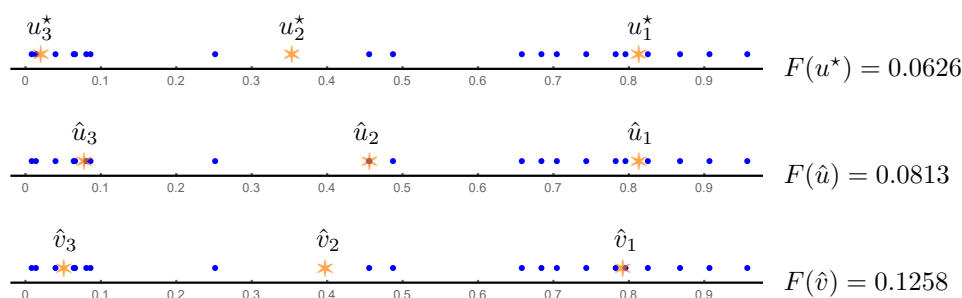
Algoritam SymDIRECT daje globalni minimum (vidi Sliku 14)

$$u^* = (0.81385, 0.35390, 0.02106) \in \Delta_1, \quad F(u^*) = 0.06258.$$

Mathematica-modul `NMinimize` pronašao je jednu od dvije točke lokalnog minimuma (vidi također Sliku 14).

$$\begin{aligned} \hat{u} &= (0.81386, 0.45654, 0.07861) \in \Delta_1, & F(\hat{u}) &= 0.08126, \\ \hat{v} &= (0.79274, 0.05152, 0.39814) \in \Delta_1, & F(\hat{v}) &= 0.12581. \end{aligned}$$

¹Maketu su izradili Lidija Radan i Jan Valenta, Odjel za matematiku, Sveučilište u Osijeku.

Slika 14: Jedan globalni $(u^*, F(u^*))$ i dva lokalna minimuma $(\hat{u}, F(\hat{u}))$, $(\hat{v}, F(\hat{v}))$ iz Primjera 6

Zadatak 6. Neka je $\mathcal{A} \subset \mathbb{R}$. Konstruirajte algoritam po uzoru na Algoritam *DIRECT* koji će tražiti točku globalnog minimuma simetrične Lipschitz-neprekidne funkcije definirane na trokutu $\Delta = \{(x_1, x_2) \in [0, 1]^2 : 1 \geq x_1 \geq x_2 \geq 0\}$ primjenom prethodno spomenutog postupka dijeljenja trokuta.

Još jedna nova metoda [19] za traženje točke globalnog minimuma simetrične Lipschitz-neprekidne funkcije $g: [a, b]^n \rightarrow \mathbb{R}$ nakon transformacije $f = g \circ T^{-1}$ također pretražuje hipertetraedar Δ . Primjenom nove transformacije \mathcal{T} (vidi Lemu 2) problem traženja točke globalnog minimuma funkcije f na hipertetraedru Δ prevodi se na traženje točke globalnog minimuma nove funkcije $\Phi: [0, 1]^n \rightarrow \mathbb{R}$, $\Phi(\zeta) = f(\mathcal{T}(\zeta))$ na hiperkocki $[0, 1]^n$, a tu je moguće primijeniti standardni *DIRECT* ili neki drugi optimizacijski algoritam (vidi t.6.2, str.16).

Lema 2. Neka je $\mathcal{T}: \mathbb{R}^n \rightarrow \mathbb{R}^n$ linearni operator zadan s $\mathcal{T}(e_i) = \sum_{j=1}^i e_j$, gdje je $\{e_1, \dots, e_n\}$ standardna ortonormirana baza u \mathbb{R}^n . Tada vrijedi

$$c \in \Delta \Leftrightarrow c \in [0, 1]^n \quad i \quad \mathcal{T}^{-1}(c) \in [0, 1]^n \quad .$$

Zadatak 7. Dokažite Lemu 2.

6 Neke druge mogućnosti

6.1 Kratki povijesni pregled

Neka je $f: \mathcal{D} \rightarrow \mathbb{R}$, $\mathcal{D} \subseteq \mathbb{R}^n$, proizvoljna funkcija koja u točki $x^* = \operatorname{argmin}_{x \in \mathcal{D}} f(x)$ postiže globalni minimum $f^* = f(x^*)$. Kao što je već i ranije navedeno, problem određivanja takve točke $x^* \in \mathcal{D}$ zovemo *problem globalne optimizacije*. Metode za rješavanje problema globalne optimizacije tijekom povijesti² najviše su se razvijale pod utjecajem razvoja inženjerskih tehnologija. Prvi poznati znanstveni rad u literaturi povezan s globalnom optimizacijom potječe iz 1797. godine³. U tom je radu prvi puta sustavno analitički

²Podaci o kratkom povijesnom pregledu metoda globalne optimizacije preuzeti su uglavnom iz [13].

³Iz povijesnih razloga navodimo originalnu referencu:

J.L. Lagrange, *Théorie des fonctions analytiques*, Impr. de la République, Paris, 1797.

obrađen problem globalne optimizacije. Analitičke metode za rješavanje problema globalne optimizacije općenito su vrlo kompleksne, posebno u slučaju velikog broja varijabli te zbog toga prije pojave suvremenih računala njihova efikasna implementacija bila je gotovo nemoguća. U kasnijim 50.-tim godinama 20. stoljeća razvijene su prve numeričke metode za rješavanje problema globalne optimizacije koje su se zasnivale na tzv. načelu *podijeli-pa-vladaj*. Ugrubo govoreći, ovo se načelo sastoji od podjele domene \mathcal{D} na manje poddomene na kojima se mogla primijeniti neka lokalna metoda. Na taj način generira se iterativni postupak koji konvergira prema točki globalnog minimuma. Tipičan primjer ovakve metode je *Metoda grananja i ograđivanja* (eng. *Branch and Bound - (BB)*) [15], koja se u svom originalu koristila za rješavanje specifičnih problema globalne optimizacije povezane s linearnim odnosno cjelobrojnim programiranjem. *Metoda grananja i ograđivanja* primijenjena na općenitu funkciju prvi se puta koristi u radu [3]. U razdoblju od 1970. do 1980. najvećim su se dijelom razvijale metode za rješavanje mješovitih cjelobrojnih i nelinearnih problema [13]. U tom je razdoblju također razvijena metoda ovog tipa koja se temelji na intervalnoj aritmetici [7, 8]. Metoda se nije pokazala efikasnom zbog svoje općenito vrlo spore konvergencije.

Do kraja 80.-ih godina razvijene su različite metode globalne optimizacije koje su koristile specijalne strukture problema, pa i *stohastička optimizacija* (vidi primjerice [20]). Početkom 90.-ih godina, broj znanstvenih publikacija u području globalne optimizacije značajno se povećao. Također se razvija veliki broj softverskih paketa za rješavanje ovog problema. U tom razdoblju izdana je prva knjiga vezana uz globalnu optimizaciju [9].

U radu [1] *Metoda grananja i ograđivanja* pod nazivom αBB primijenjena je na općeniti nelinearni i nekonveksni problem uvjetne optimizacije. Nakon toga uslijedio je niz znanstvenih radova u kojima se razmatra ova metoda, a temeljna pretpostavka je da je funkcija cilja dva puta neprekidno diferencijabilna. Nešto kasnije razvijaju se stohastičke metode, kao što su *Simulated and Nested Annealing* [18], *Tabu Search* [2], *Multi-Level Single Linkage* [12], *Variable Neighbourhood Search* [14], *Differential Evolution*, itd.

6.2 Metoda grananja i ograđivanja

U nastavku ćemo ukratko proanalizirati prethodno spomenutu *Metoda grananja i ograđivanja*. Temeljna ideja ove metode sastoji se od definiranja liste \mathcal{L} konveksnih podskupova domene \mathcal{D} , pri čemu pretpostavljamo da barem jedan od tih podskupova sadrži točku globalnog minimuma funkcije f . Iz liste \mathcal{L} odabiremo podskup R , kojega obrišemo iz liste, te na tom podskupu određujemo gornju $\Phi_{UB}(R)$ i donju $\Phi_{LB}(R)$ među optimalne vrijednosti funkcije f^* tako da vrijedi

$$\Phi_{LB}(R) \leq \min_{x \in R} f(x) \leq \Phi_{UB}(R).$$

Ovaj postupak zovemo *ograđivanje*.

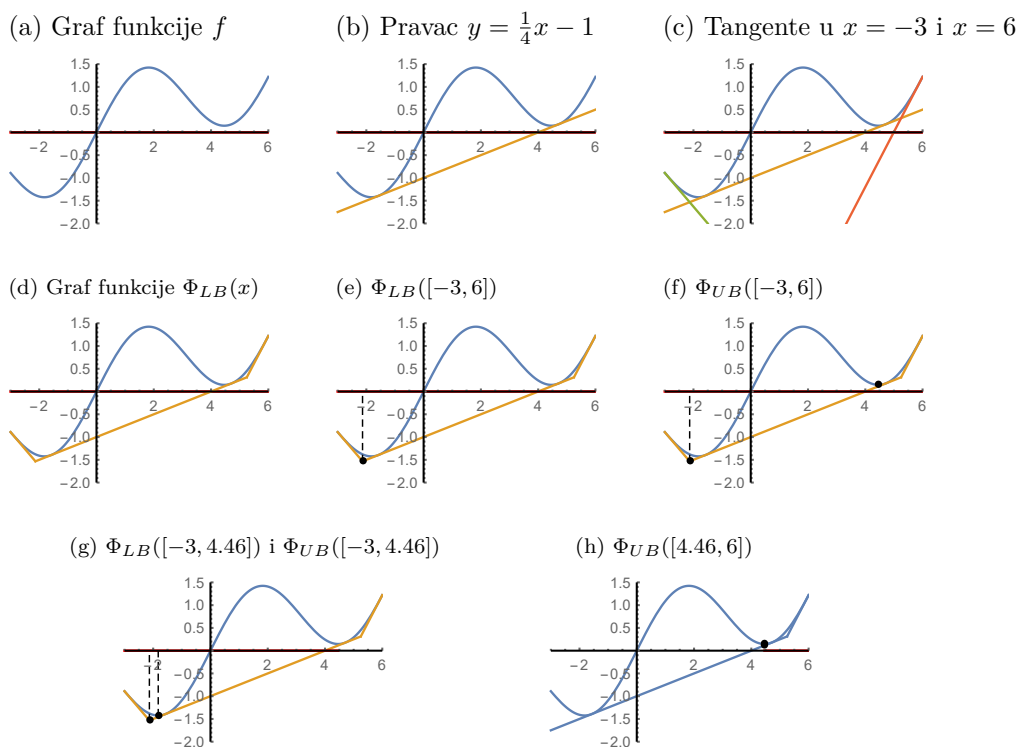
U cilju osiguravanja brzine konvergencije, gornju i donju među važno je odrediti na što je moguće jednostavniji način. Ako je pritom $\Phi_{UB}(R) - \Phi_{LB}(R) < \varepsilon$, gdje je $\varepsilon > 0$ unaprijed zadana točnost, podskup R zadržavamo te odgovarajuću vrijednost gornje međe $\Phi_{UB}(R)$ smatramo trenutnom vrijednošću globalnog minimuma. Također, iz liste \mathcal{L} uklonimo sve one podskupove kojima je vrijednost donje međe veća od $\Phi_{UB}(R)$.

U suprotnom, ako je $\Phi_{UB}(R) - \Phi_{LB}(R) > \varepsilon$, podskup R dalje dijelimo u disjunktne podskupove koje uvrštavamo u listu \mathcal{L} . U listi \mathcal{L} zadržavamo samo one podskupove kojima je vrijednost gornje međe manja od trenutne vrijednosti globalnog minimuma. Postupak daljnjeg dijeljenja nazivamo *grananje*.

U ovisnosti o načinu konstrukcije gornje i donje međe (ograđivanje), kao i načinu daljnjeg dijeljenja podskupa (grananja) mogu se konstruirati različite metode koje su zasnovane na ovoj ideji. Tipični primjer ove grupe metoda je već ranije spomenuta *αBB metoda*, kod koje se u svrhu definiranja gornje međe koristi pretpostavka da je funkcija f dva puta neprekidno diferencijabilna [1].

U nastavku kao ilustraciju navodimo primjer iz [13], u kojemu rješavamo problem jednodimenzionalne globalne optimizacije primjenom načela grananja i ograđivanja.

Primjer 8. Zadan je interval $Q = [-3, 6]$ te funkcija $f: Q \rightarrow \mathbb{R}$ formulom $f(x) = \frac{1}{4}x + \sin x$ (Slika 15a). Primjenom načela grananja i ograđivanja treba pronaći točku u kojoj se postiže globalni minimum funkcije f na intervalu Q s točnošću $\varepsilon = 0.15$



Slika 15: Metoda grananja i ograđivanja za funkciju $f: [-3, 6] \rightarrow \mathbb{R}$, $f(x) = \frac{1}{4}x + \sin x$

Na početku metode lista \mathcal{L} sastoji se samo od intervala $Q = [-3, 6]$, odnosno $\mathcal{L} = \{[-3, 6]\}$. Odredimo donju $\Phi_{LB}([-3, 6])$ i gornju među $\Phi_{UB}([-3, 6])$. Kako je $-1 \leq \sin x \leq 1$, vrijedi (vidi Slika 15b)

$$f(x) \geq \frac{1}{4}x - 1.$$

U cilju profinjenja donje međe, u rubnim točkama $x = -3$ te $x = 6$, konstruiramo tangente na graf funkcije f . U točki $x = -3$, tangenta glasi: $y = -3.11 - 0.74x$, dok u točki $x = 6$, tangenta glasi: $y = -6.04 + 1.21x$ (vidi Sliku 15c). Sada je očigledno (Slika 15d)

$$f(x) \geq \Phi_{LB}(x) := \max\{\frac{1}{4}x - 1, -3.11 - 0.74x, -6.04 + 1.21x\},$$

odakle je (Slika 15e)

$$\Phi_{LB}([-3, 6]) = \min_{x \in [-3, 6]} \Phi_{LB}(x) = -1.53,$$

a postiže se u točki $\bar{x} = -2.13$.

Nadalje određujemo gornju među. Jedna je mogućnost primijeniti neku lokalnu minimizacijsku metodu te za gornju među odabrati vrijednost funkcije kojoj je ta metoda konvergirala. Kako je funkcija f dva puta neprekidno derivabilna, u tu svrhu primjenjujemo *Newtonovu metodu*, pri čemu za početnu aproksimaciju uzimamo $x_0 = 6$. *Newtonova metoda* konvergirala je prema $\tilde{x} = 4.46$ te je prema tome gornja međa (Slika 15f)

$$\Phi_{UB}([-3, 6]) = f(4.46) = 0.147.$$

Razlika gornje i donje međe iznosi

$$\Phi_{UB}([-3, 6]) - \Phi_{LB}([-3, 6]) = 1.67 > \varepsilon = 0.15,$$

što znači da interval $[-3, 6]$ treba podijeliti (granati) u dva podintervala. Pretpostavimo da interval dijelimo na podintevala $Q_1 = [-3, 4.46]$, $Q_2 = [4.46, 6]$. Točku $\tilde{x} = 4.46$ zovemo *točka grananja*. Lista \mathcal{L} sada glasi

$$\mathcal{L} = \{[-3, 4.46], [4.46, 6]\}.$$

Odabiremo interval $[-3, 4.46]$ iz liste, obrišemo ga iz liste te za njega određujemo donju i gornju među. Za donju među koristimo istu funkciju kao i za interval $[-3, 6]$ pa je

$$\Phi_{LB}([-3, 4.46]) = \min_{x \in [-3, 4.46]} \Phi_{LB}(x) = -1.53,$$

a postiže se u točki $\bar{x} = -2.13$. U svrhu određivanja gornje međe na skupu $[-3, 4.46]$, primjenjujemo *Newtonovu metodu* s početnom aproksimacijom $x_0 = 3$. *Newtonova metoda* konvergirala je prema $\tilde{x} = -1.82$ te je prema tome gornja međa (Slika 15g)

$$\Phi_{UB}([-3, 4.46]) = f(-1.82) = -1.42.$$

Razlika gornje i donje međe iznosi

$$\Phi_{UB}([-3, 4.46]) - \Phi_{LB}([-3, 4.46]) = 0.11 < \varepsilon = 0.15.$$

Ovaj interval ne dijelimo dalje te za aproksimaciju točke u kojoj se postiže globalni minimum uzimamo $\tilde{x} = -1.82$, dok je odgovarajuća vrijednost globalnog minimuma $U := -1.42$.

Nadalje, promatramo interval $[4.46, 6]$. Za konstrukciju donje međe koristimo funkciju $\Phi_{LB}(x)$ te je

$$\Phi_{LB}([4.46, 6]) = f(4.46) = 0.115.$$

U svrhu određivanja gornje međe na skupu $[4.46, 6]$, primjenjujemo *Newtonovu metodu* s početnom aproksimacijom $x_0 = 6$. *Newtonova metoda* konvergirala je prema $\tilde{x} = 4.46$, te je pri tome gornja međa (Slika 15h)

$$\Phi_{UB}([4.46, 6]) = f(4.46) = 0.147.$$

Razlika gornje i donje međe iznosi

$$\Phi_{UB}([4.46, 6]) - \Phi_{LB}([4.46, 6]) = 0.04 < \varepsilon = 0.15.$$

Ovaj interval ne dijelimo dalje. Kako za trenutnu aproksimaciju globalnog minimuma $U = -1.42$ vrijedi $U < \Phi_{LB}([4.46, 6]) = 0.147$, ovaj interval ne sadrži točku globalnog minimuma.

Budući da lista \mathcal{L} ne sadrži više podskupova, zaključujemo da je funkcija f postigla globalni minimum funkcije u točki $\tilde{x} = -1.82$ ta da on iznosi $U = -1.42$.

Literatura

- [1] I. P. ANDROULAKIS, C. D. MARANAS, C. A. FLOUDAS, *αBB : A global optimization method for general constrained nonconvex problems*, Journal of Global Optimization, **7**(1995) 337–363.
- [2] R. CHELOUAH, P. SIARRY, *Tabu search applied to global optimization*, European Journal of Operational Research, **123**(2000) 256–270.
- [3] J. E. FALK, R. M. SOLAND, *An algorithm for separable nonconvex programming problems*, Management Science, **15**(1969) 55–569.
- [4] D. E. FINKEL, *DIRECT Optimization Algorithm User Guide*, Center for Research in Scientific Computation. North Carolina State University, 2003, <http://www4.ncsu.edu/definkel/research/index.html>.
- [5] J. M. GABLONSKY, *Direct version 2.0*, Technical report, Center for Research in Scientific Computation. North Carolina State University, 2001.
- [6] R. GRBIĆ, E. K. NYARKO, R. SCITOVSKI, *A modification of the DIRECT method for Lipschitz global optimization for a symmetric function*, Journal of Global Optimization, **57**(2013) 1193–1212.
- [7] E. R. HANSEN, *Global optimization using interval analysis: The one-dimensional case*, Journal of Optimization Theory and Applications, **29**(1979) 331–344.
- [8] E. R. HANSEN, G. W. WALSTER, *Global Optimization Using Interval Analysis*, Marcel Dekker, INC., New York-Basel, 1992.
- [9] R. HORST, H. TUY, *Global Optimization: Deterministic Approaches*, Springer, Berlin, third edition, 1996.

- [10] M. JAMIL, X.-S. YANG, *A literature survey of benchmark functions for global optimization problems*, International Journal of Mathematical Modelling and Numerical Optimisation, **4**(2013) 150–194.
- [11] D. R. JONES, C. D. PERTTUNEN, B. E. STUCKMAN, *Lipschitzian optimization without the Lipschitz constant*, Journal of Optimization Theory and Applications, **79**(1993) 157–181.
- [12] S. KUCHERENKO, Y. SYTSKO, *Application of deterministic low-discrepancy sequences in global optimization*, Computational Optimization and Applications, **30**(2004) 297–318.
- [13] L. LIBERTI, *Introduction to Global Optimization*, LIX, École Polytechnique, 2008.
- [14] N. MLADENOVIĆ, J. PETROVIĆ, V. KOVAČEVIĆ-VUJČIĆ, M. ČANGALOVIĆ, *Solving a spreadpectrum radar polyphase code design problem by tabu search and variable neighbourhood search*, European Journal of Operations Research, **15**(2003) 389–399.
- [15] C. H. PAPADIMITRIOU, K. STEIGLITZ, *Combinatorial Optimization: Algorithms and Complexity*, Dover, New York, 1998.
- [16] R. PAULAVIČIUS, J. ŽILINSKAS, *Simplicial Global Optimization*, volume X of *Series: Springer Briefs in Optimization*, Springer-Verlag, Berlin, 2014.
- [17] J. D. PINTÉR, *Global Optimization in Action (Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications)*, Kluwer Academic Publishers, Dordrecht, 1996.
- [18] S. RAJASEKARAN, *On simulated annealing and nested annealing*, Journal of Global Optimization, **16**(2000) 43–56.
- [19] R. SCITOVSKI, *A new global optimization method for a symmetric lipschitz continuous function and application to searching for a globally optimal partition of a one-dimensional set*, Journal of Global Optimization, (2017), DOI: 10.1007/s10898-017-0510-4.
- [20] Y. YAO, *Dynamic tunnelling algorithm for global optimization*, IEEE Transactions On Systems Man and Cybernetics, **19**(1989) 1222–1230.