Tartalomjegyzék

Bevezető

A globális optimalizálás korunk egyik rohamosan fejlődő tudományterülete, hiszen az alkalmazott tudományágak, mint például a fizika, kémia, biológia, közgazdaságtan, stb. mind-mind globális optimalizálási feladat(ok) megoldására támaszkodva válaszolják meg egyes kutatási kérdéseiket. Ez nagy motiváció a globális optimalizálási módszerek kutatóinak, és az elmúlt évtizedekben a számítógépek rohamos fejlődésének is köszönhetően igen sok szép eredmény született mind az elméleti mind a gyakorlati oldalon. Ez a rohamos fejlődés sok újat hozott tudományos cikkek terén, viszont nem született olyan leírás, ami bemutatná az alapvető elméleti hátteret a gyakorlati módszerekkel együtt olyan formában, hogy se túl elméleti, se túl technikai ne legyen.

A jelen dokumentum célja bevezetni az olvasót a globális optimalizálás témakörébe eleinte elméleti szempontból, majd mindinkább rátérve a gyakorlatban megvalósítható és használható módszerek tárgyalására. Az első fejezetekben tárgyaljuk az általános globális optimalizálási feladatok nehézségét, hogy megérthessük miért olyan fontos a megoldandó feladatok tulajdonságainak ismerete. Az ezt követő fejezetekben olyan módszerekről olvashatunk, amelyek a feladat adott tulajdonságai mellett képesek megtalálni a globális optimum egy akármilyen jó közelítését véges időben. Az utolsó fejezetekben olyan módszerekről ejtünk szót, amik ugyan nem követelnek meg kvázi semmit a feladattól, de ennek megfelelően az eredmény pontosságáról nincs információnk. Az így bemutatott elméleti és gyakorlati ismeretek birtokában az olvasó könnyen eligazodik a globális optimalizálási módszerek sokaságában, és megtalálhatja egy adott feladathoz a hatékony megoldó módszert.

Nem támaszkodunk az alapvető matematikai ismereteken kívül másra, és az esetleg hiányzó alapfogalmakat az olvasó segítésére a függelékben szedtük össze. Így ez az összefoglaló hasznos lehet nem csak matematikus, és informatikus hallgatók, kutatók számára, hanem egyaránt bármely, az alkalmazási oldalról érkező érdeklődő olvasónak. Ahol lehetséges volt, példák segítségével próbáltuk segíteni a megértést, illetve sok helyen feladatokkal látjuk el az olvasót hogy próbára tegye a megszerzett tudását. A módszerek megértését segítik a pszeudokódok és Matlabban megírt algoritmusok, amiket így ki is próbálhatunk néhány tesztfeladaton.

Kívánom, hogy ez az összefoglaló a globális optimalizálás iránt érdeklődők segítségére legyen és olvasóim haszonnal forgassák.

G.-Tóth Boglárka

1. fejezet

Optimalizálás

Az optimalizálás az alkalmazott kutatási területek mindegyikén megjelenik, legyen az fizika, kémia, közgazdaságtan, vagy akár biológia. A valós életben még gyakrabban találkozunk optimalizálási problémákkal, de ezeket legtöbbször még észre sem vesszük: Mit együnk ebédre? Melyik boltba menjünk? Milyen útvonalon jutunk el valahová? Ezek gyakorlatilag mind-mind optimalizálási feladatok. Ezért ez egy erősen alkalmazás-orientált terület, így érthetően az alkalmazott matematika része.

Az általános optimalizálási probléma megfogalmazása lehet a következő: Lehetséges alternatívák közül válasszuk ki a „legjobbat". Ez persze nagyon tág megfogalmazás, mégis helyénvaló. Az 1.1 ábrán láthatjuk egy függvény lokális és globális optimumait az X tartományon. Mint a példa is mutatja sokszor találhatunk lokális optimumokat, amik nem globálisak, és ezek bármilyen messze lehetnek a globális optimumtól.

A következő alfejezetben megpróbáljuk kisebb osztályokba sorolni az optimalizálás főbb részterületeit, ezzel segítve a globális optimalizálás elhelyezését. A fejezet végén definiáljuk az általános globális optimalizálási feladatot, bevezetünk néhány ehhez kapcsolódó alapvető fogalmat, és ezeket egy példán keresztül be is mutatjuk.

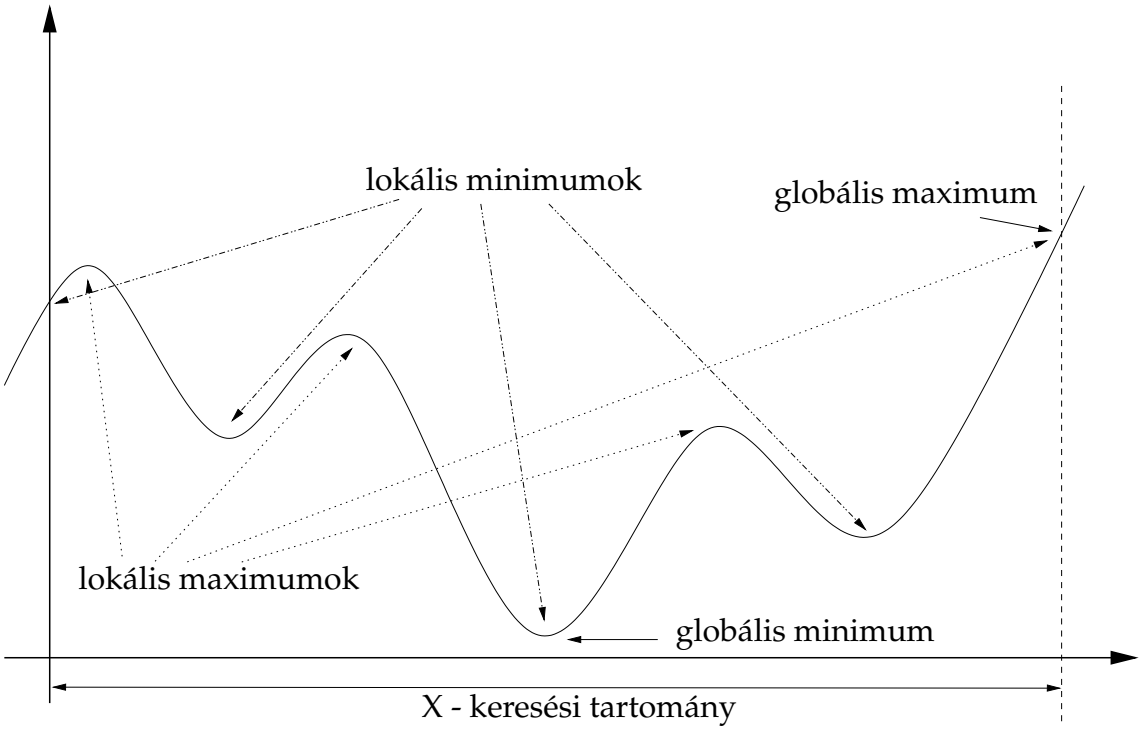
1.1. Főbb részterületek

Konvex programozás: A célfüggvény és a megszorítások is konvexek. Minden lokális optimum globális is egyben.

Lineáris programozás: Tekinthető a konvex optimalizálás részének, itt mind a célfüggvény, mind a feltételek lineárisak.

Konkáv programozás: A megszorítások konvexek, a célfüggvény konkáv. Az elnevezést néhol konkáv maximalizálásra értik, vagyis konvex programozásra, mivel a Kaursh-Kuhn-Tucker (KKT) feltételek először konkáv függvény maximalizálására lettek kimondva.

Kvadratikus programozás: A célfüggvény kvadratikus, a feltételek lineárisak. Ha a célfüggvény konvex, polinom idejű algoritmussal megoldható, egyébként



1.1. ábra. Lokális és globális optimumok

NP-nehéz.

Egészértékű programozás: Egyes változók csak egész értékeket vehetnek fel. Az egyik legelterjedtebben használt módszer a Korlátozás és szétválasztás, ahol a részfeladatok lineáris programozási problémák.

Kombinatorikus optimalizálás: A lehetséges alternatívák száma véges, vagy végesre redukálható. Általában a problémák diszkrét struktúrákon (gráfokon, matroidokon) definiáltak.

Nemlineáris programozás: Nemlineáris célfüggvény, feltételek. Lokális optimumot (KKT pontot) keres általában iteratív módszerekkel. Ezek a módszerek alkalmazhatók a globális optimalizálásban mint lokális keresők, ezért néhány módszert bemutatunk a 10. és 11. fejezetben.

Sztochasztikus programozás: Bizonytalansággal rendelkező problémák megoldása a célja. Egyes paraméterek valószínűségi változók, tudjuk az eloszlásukat. Általában egy függvény várható értékét kell optimalizálni. Nem összekeverendő a sztochasztikus optimalizálással, ahol az eljárás sztochasztikus, nem a feladat!

Robusztus programozás: A bizonytalan paraméterekről ez esetben csak annyit tudunk, hogy milyen intervallumon belül mozoghatnak.

Kényszerek kielégítése: Nincsen célfüggvény, csak a kényszerfeltételek adottak. Ha a célfüggvényt konstansnak tekintjük optimalizálási problémához jutunk, de vannak módszerek amelyekhez ez nem szükséges.

Többcélú optimalizálás: A probléma nem csak egy, hanem több célfüggvénnyel rendelkezik, amelyek nem vonhatóak össze. Nem egyértelmű, hogy mit nevezünk megoldásnak.

Globális optimalizálás: Több lokális optimum van. A nemlineáris programozással ellentétben itt a globális optimumot keressük.

1.2. Globális optimalizálás

Az általános globális optimalizálási feladatot a következőképpen írhatjuk fel.

$$\begin{aligned} &\min_{x\in A} / \max_{x\in A} f(x) \\ &\text{f.h. } g_i(x) \leq 0 \quad i \in I \quad \text{egyenlőtlenség feltételek} \\ &\qquad g_j(x) = 0 \quad j \in J \quad \text{egyenlőség feltételek} \end{aligned} \tag{1.1}$$

ahol

- $f(x) : A \to \mathbb{R}$ a célfüggvény: az a függvény, amelynek a minimumát vagy maximumát keressük,
- $A = \{x \in \mathbb{R}^n \mid a \leq x \leq b, a, b \in \mathbb{R}^n\}$ intervallum korlát (bound constraint),
- $\forall i \in I, j \in J$ $g_i, g_j : A \to \mathbb{R}$ korlátozó feltételek, g_i egyenlőtlenség, g_j egyenlőség feltétel,
- $X = \{x \in A \mid g_i(x) \leq 0 \; \forall i \in I, g_j(x) = 0 \; \forall j \in J\}$ a megengedett megoldások halmaza.

A továbbiakban minimalizálási problémákra szorítkozunk, hiszen a

$$\max f(x) = -\min -f(x)$$

konverzióval bármely maximalizálási feladatot visszavezethetünk minimalizálásra. Az adott feltételektől függően az alábbi módon hívhatjuk az optimalizálási feladatot:

- ha $X = \mathbb{R}^n$ akkor feltétel nélküli az optimalizálási feladatunk,
- ha $I \cup J = \emptyset$ de $X = A \subset \mathbb{R}^n$ akkor intervallum korlátozott feladatunk van,
- és ha $I \cup J \neq \emptyset$ akkor feltételes vagy korlátozott optimalizálási feladatról beszélünk.

Egy feladat megoldásának tekinthetjük a globális (esetleg lokális) optimum értékét, helyét, az összes globális (esetleg lokális) optimum halmazát, vagy ezek valamely kombinációját.

Használjuk a következő fogalmakat, jelölést:

- f_i^* egy lokális minimum érték, vagyis létezik $x_i^* \in X$ és $\varepsilon > 0$ hogy $f_i^* = f(x_i^*) \leq f(x)$ minden $||x - x_i^*|| < \varepsilon, x \in X$ esetén. Legyenek nagyság szerint rendezettek, azaz, $f_1^* < f_2^* < \ldots < f_i^* < f_{i+1}^* < \ldots$.
- $X_i^* = \{x_i^* \in X \mid f(x_i^*) = f_i^*\}$ az f_i^* lokális minimumhoz tartozó lokális minimumhelyek halmaza.
- $f^* = f_1^*$ a globális minimum, $f^* \leq f(x)\ \forall x \in X$.
- $X^* = \{x^* \in X \mid f(x) = f^*\} = \{x^* \in X \mid f(x^*) \leq f(y)\ \forall y \in X\}$ a globális minimum pontok halmaza.
- $L_f(y) = \{x \in X \mid f(x) \leq y\}$ f egy szinthalmaza.

1.1. Példa. Egy hajózható csatornát keresünk a tenger egy téglalap alakú területén belül. Legyen d a hajózhatósághoz szükséges mélység. A következő feladatok megoldása lehet szükséges.

a) Határozzuk meg a legkisebb távolságot a felszín és a fenék között, ha f a mélységet leíró függvény és X a megadott terület, például

$$X = \left\{ (x_1, x_2) \;\middle|\; \begin{array}{l} a \leq x_1 \leq b \\ c \leq x_2 \leq d \end{array} \right\},$$

akkor becsüljük a globális minimumot, f^*-ot.

b) Határozzuk meg azokat a lokális minimumokat, ahol a hajó fennakadhat, azaz határozzuk meg

$$f_i^* < d, \quad f^* = f_1^* < f_2^* < \cdots < f_l^* < d < f_{l+1}^* < \ldots$$

és az ezekhez tartozó

$$X_i^* = \{x \in X \mid f(x) = f_i^*\} \quad \text{lokális optimum pontok halmazait.}$$

c) Határozzuk meg az $L_f(d)$ szinthalmazt.

A megoldhatóság mindig kérdéses az általános esetben. A megoldhatóság esélye apriori információk segítségével növelhető. Ilyen apriori információ pl:

- $f(x), \nabla f(x), \nabla^2 f(x), \ldots$
- Simaság, közelítő lokális minimum, alsó és felső korlátok a globális minimumra, stb.

A globális optimalizáló eljárások elvi felépítése:

- **Globális fázis:** A teljes keresési régiót (X) elemzi
- **Lokális fázis:** Az ígéretes (várhatóan optimumhoz közeli) részeket behatóbban vizsgálja.
- **Adaptáció:** Az ígéretes részeken sűrűbben mintavételezünk, ez egy köztes lépés a lokális és a globális fázis között.

2. fejezet

A globális optimum karakterizációja

2.1. Feltétel nélküli feladat lokális optimumának szükséges és elégséges feltétele

Szükséges feltétel: $\nabla f = 0$
Elégséges feltételek:

$$f''(x) > 0 \;\; (H_f(x) \text{ pozitív definit}) \quad \textbf{esetén minimum}$$
$$f''(x) < 0 \;\; (H_f(x) \text{ negatív definit}) \quad \textbf{esetén maximum}$$

Ha minden lokális optimumot megtalálunk

$$f_1^* < f_2^* < \ldots; \qquad X_i^* = \{x \in X | f(x) = f_i^*\}$$

akkor a globális optimum elégséges feltétele

$$f(x^*) \leq f(x) \quad \forall x \in \bigcup X_i^*$$

2.2. Feltételes optimalizálás lokális optimumának szükséges és elégséges feltétele

A feltételek megfogalmazásához idézzük fel az általános globális optimalizálási feladat, (1.1) felírását, itt már szorítkozva minimalizálásra.

$$\begin{aligned} &\min_{x \in A} f(x) \\ &\text{f.h. } g_i(x) \leq 0 \quad i \in I \\ &\qquad g_j(x) = 0 \quad j \in J \end{aligned} \tag{2.1}$$

A fenti feladat Lagrange-függvényének nevezzük az

$$L(x, u) = f(x) + \sum_{i \in I \cup J} u_i g_i(x)$$

függvényt, ahol u_i az i. feltétel Langrange multiplikátora ($i = 1, \dots, m$, ahol $m = |I \cup \cup J|$). Jelöljük $\nabla_x L(x, u)$-val a csak x változók szerint vett gradiens vektort. Ekkor a lokális optimalitás szükséges feltételét a következő tétel mondja ki.

2.1. Tétel (Karush-Kuhn-Tucker tétele). *Ha az x pont a (2.1) feladat lokális optimuma (és a feltételek kielégítenek bizonyos regularitási feltételeket), akkor létezik $u \in \mathbb{R}^m$ amire teljesül, hogy*

i) $\nabla_x L(x, u) = 0$,

ii) $u_i g_i(x) = 0$ *minden* $i \in I$,

iii) $u_i \geq 0$ *minden* $i \in I$,

vagyis i)-iii) az optimalitás szükséges feltételei.

Az ii) feltételek alapján egy egyenlőtlenségfeltétel vagy aktív, azaz $g_i(x) = 0$, vagy a hozzá tartozó u_i Lagrange multiplikátor nulla. Vegyük észre hogy az első esetben a feltételt mint egyenlőségfeltételt kezeljük, a második esetben pedig mintha ez a feltétel nem is létezne.

A tételben szereplő regularitási feltétel lehet a lineárisan független feltételrendszer (LICQ – linear independent constraint qualification), ahol megköveteljük, hogy a lokális optimumban az aktív feltételek gradiense legyen lineárisan független.

2.3. Konvertálás egydimenziós feladatra

Minden globális optimalizálási feladatot elméletileg konvertálhatunk egy egydimenziós feladatra. Ehhez definiáljuk egy szinthalmaz hatásos tartományát.

2.2. Definíció. Legyen $L_f(y)$ az f egy szinthalmaza. Ekkor a

$$G_f = \{f(y) \mid y \in \mathbb{R}^n, L_f(y) \neq \emptyset\}$$

kifejezést $L_f(y)$ *hatásos tartomány*ának nevezzük.

Ekkor $f^* = \min_{x \in G_f} x$, az eredeti globális optimalizálási feladat egy egydimenziós felírása. A G_f halmaz definíciója miatt viszont ez nem egy G.O. probléma.

2.4. Hogyan döntsük el egy minimumról, hogy lokális vagy globális?

Ha létezne egy olyan gyakorlatilag használható módszer, amellyel egy lokális optimumról könnyen eldönthetnénk, hogy globális-e, akkor lokális keresőket használva is lenne esélyünk úgy megállni, hogy tudjuk, a globális optimumot találtuk meg. Most két ilyen próbálkozásról fogunk beszélni.

2.3. Definíció. Az $y \mapsto L_f(y)$ pont-halmaz–leképezés *alulról félig folytonos* az $\bar{y} \in G_f$ pontban, ha

$$(\forall x \in L_f(\bar{y}))(\forall \{y^i \in G_f\}_i \to \bar{y})$$

$$(\exists K \in \mathbb{N})(\exists \{x^i \in X\}_i)(\forall i \geq K)(\{x^i\} \to x, x^i \in L_f(y^i))$$

Azaz ha egy $\bar{y}$ szintvonalhoz szintvonalak sorozatával tartunk, akkor megadható ezen sorozat mentén az $\bar{y}$ szintvonal bármely pontjához egy konvergens X-beli pontsorozat.

2.4. Tétel. *Legyen $f : X \subset \mathbb{R}^n \to \mathbb{R}$ valós értékű függvény. Egy $\bar{y}$ értékre $y \mapsto L_f(\bar{y})$ alulról félig folytonos, ha $\forall x \in X$ amire $f(x) = \bar{y}$, akkor x vagy egy globális optimumhely, vagy nem optimumhely.*

Bizonyítás (vázlat). A bizonyítás alapja hogy $\bar{y}$ lokális (de nem globális) optimum esetén, ha az y^i szintvonalsorozatra $\forall y^i < \bar{y}$, akkor belátható, hogy nincs a lokális optimumhelyhez tartozó konvergens X-beli sorozat y^i értékekkel, hiszen lokális optimum pont kis környezetében bármely pont függvényértéke legalább akkora mint a lokális optimum. ■

Ez a megközelítés teljesen teoretikus, gyakorlatban használhatatlan. A következő próbálkozás valamivel jobb ennél.

Tegyük fel, hogy találtunk egy jelöltet f^*-ra, amely a z pontban vétetik fel. Ha affin transzformációval (negatívval való szorzás nélkül) tudunk generálni egy g függvényt, amelyre

$$g(z) = -1 \text{ és } g(x) \leq 0 \quad \forall x \in X,$$

akkor az alábbi tétel karakterizálja a globális optimumot.

2.5. Tétel. *A konvex*

$$G(t) = \int_X |g(x)|^t \, dx$$

függvény $\lim_{t\to\infty} G(t)$ határértéke ∞, ha z nem globális minimalizáló pont.

Azért ez sem igazán praktikus, numerikus integrálás szükséges és a konverzió sem egyértelmű.

2.5. Létezik globálissal ekvivalens lokális probléma?

Tegyük fel, hogy f folytonos és X kompakt. Legyen $\tilde{X}$ az X legkisebb konvex burka. Legyen u olyan függvény, amely

a) konvex $\tilde{X}$-en,

b) $u(x) \leq f(x), \forall x \in X$, és

c) $u(x) \geq g(x), \forall x \in X$, ahol g teljesíti a)-t és b)-t.

Ekkor u minimalizáló pontjainak halmaza tartalmazza f globális minimumhelyét.

Ez sem praktikus. u-t nehéz meghatározni és ha több globális minimum van, akkor u-nak végtelen sok minimumhelye van.

3. fejezet

Globális optimalizálási problémák megoldhatósága

Egy feladat megoldhatósága nagyban függ a probléma tulajdonságaitól, illetve hogy mit keresünk mint megoldást. Ha a globális minimum értékét, f^*-ot akarjuk közelíteni, akkor egy általános folytonos függvényre nem garantált, hogy $f^* + \varepsilon$ véges számú lépésben megtalálható.

Ha a globális minimumhelyek egyikét, vagy mindegyikét, X^*-ot, akkor a probléma még csak nem is korrekt kitűzésű, mivel $\exists f : f(x) \leq f^* + \varepsilon$, de $\|x - x^*\| >> \delta$. Ezek az általános állítások sokban javíthatók, ha a probléma valamilyen jó tulajdonsággal rendelkezik.

3.1. A globális optimum létezése

Weierstrass tétele kimondja, hogy $D \subseteq \mathbb{R}^n$ nemüres kompakt halmazon folytonos függvény felveszi abszolút minimumát és maximumát is. Az abszolút minimum és maximum természetesen ugyanaz, mint a globális minimum, ill. maximum. Tulajdonképpen azért szokás az egyenlőtlenség feltételeket mindig $g_i(x) \leq 0$-ként megadni, azaz sosem szigorú egyenlőtlenségként, hogy a megengedett megoldások halmaza kompakt legyen, és ezáltal létezzen az optimum.

3.2. Megoldható problémák

- Ismert f^*, vagy alsó és felső korlátok generálhatóak valamely módszerrel
- f Lipschitz-folytonos:

$$|f(x) - f(y)| \leq L \cdot \|x - y\| \quad \forall x, y \in X$$

- Folytonossági modulus:

$$w(t) = \max_{d(x,y) \leq t} |f(x) - f(y)|$$

Ha kiértékelünk N pontot $x_1, \dots, x_N \in X$, akkor

$$\tilde{f}_N := \min_{i \in \{1,\dots,N\}} f(x_i) \qquad \text{a közelítő megoldásunk}$$

Ekkor a megoldásunk hibája becsülhető:

$$\left|\tilde{f}_N - f^*\right| \leq w(d_N)$$

ahol d_N a mintapontoktól vett maximális távolság:

$$d_N = \max_{x \in X} \min_{i \in \{1,\dots,N\}} d(x, x_i).$$

3.3. Minimumok tulajdonságai

A lokális minimum definíciója miatt minden nem elszeparált f_i^* minimumhoz létezik egy olyan környezet

$$M_{i\varepsilon} = \{x \in X \mid f_i^* \leq f(x), \|x - x_i^*\| < \varepsilon, \text{ ahol } x_i^* \in X_i^*\}$$

hogy $\mu(M_{i\varepsilon}) > 0$, azaz valamely $\mathbb{R}^n$-beli μ mértékre $M_{i\varepsilon}$ mérhető. Ez jobb esetben azt jelenti, hogy ha megtaláljuk ezt a környezetet, akkor megtaláljuk a minimumot is. Sajnos ez nem minden esetben igaz. A minimum megtalálásához az úgynevezett vonzási tartomány megtalálása szükséges, ami ennél lehet tágabb, vagy szűkebb halmaz is.

3.1. Definíció. Legyen f_i^* lokális minimum érték. Ekkor az f_i^* lokális minimum vonzási tartománya $\text{attr}(f_i^*) \subseteq X$ azon $x \in X$ pontok legnagyobb halmaza, amelyekből a legmeredekebb lejtő módszere végtelenül kicsi lépésközzel f_i^*-hoz konvergál $\forall x \in$ $\in \text{attr}(f_i^*)$ esetén.

3.2. Megjegyzés. Képletesen, kétdimenziós esetben ha a függvényt mint domborzatot tekintjük, a vonzási tartomány a völgy olyan pontjainak halmaza, ahonnan az esővíz a minimumba folyik. $\text{attr}(f_i^*)$ nem feltétlenül összefüggő. Ha a minimum pontok száma > 1, akkor lehet nem összefüggő halmaz. A vonzási tartomány definiálható minimumhelyre is, vagyis $\text{attr}(f_i^*, x_i^*)$, ekkor a konvergencia x_i^* minimumhelyhez kell, így $\text{attr}(f_i^*, x_i^*)$ összefüggő.

3.3. Definíció. Egy f_i^* lokális minimum elérhető, ha

$$\mu(\text{attr}(f_i^*)) > 0$$

ahol μ egy $\mathbb{R}^n$-beli mérték.

3.4. Megjegyzés. A legmeredekebb lejtő módszere minden lépésben a negatív gradiens irányába halad valamilyen lépésközzel, azaz

$$x_{k+1} = x_k + \lambda \cdot d$$

ahol $d = -\nabla f(x_k)$. Az optimális lépésköz $\lambda^* = \arg\min_{\nu>0} f(x_k + \nu \cdot d)$. A végtelenül kicsi lépésköz ahhoz kell, hogy a lejtő irányában lévő legközelebbi lokális optimumot találjuk meg.

3.5. Definíció (Medence). Egy minimum medencéje a vonzási tartomány azon része, amelynek egy x belső pontjából csak a függvényérték $f(x)$ fölé növelésével juthatunk ki bármilyen irányt is választva. Formalizálva,

$$\mathrm{bas}(f_i^*, x_i^*) = \left\{ x \in \mathrm{attr}(f_i^*, x_i^*) | f(x) < \hat{f}_i, \hat{f}_i = \min_{x \in \partial\, \mathrm{attr}(f_i^*, x_i^*)} f(x) \right\}$$

ahol ∂A egy A tartomány határát jelöli. Képletesen, kétdimenziós esetben a medence a vonzási tartomány azon része, amiből nem folyik ki a víz.

Annak a valószínűsége, hogy lokális keresővel f_i^*-ot véletlen kiindulási pontból megtaláljuk X-ben:

$$p_i = \frac{\mu(\mathrm{attr}(f_i^*))}{\mu(X)}.$$

A p_1 valószínűség kitüntetett, mert ez a globális optimum megtalálásának valószínűsége.
Ha $p_1 = 1$, akkor bármely $x \in X$-ből indulva az egyetlen (globális) minimumot találjuk meg.
Ha $p_1 = 0$, akkor a globális minimum nem elérhető.
Ha $p_1 = \max_i p_i$, akkor jó esélyünk van megtalálni a globális optimumot.

3.4. Probabilisztikusan megoldható problémák

Egy G.O. probléma probabilisztikusan megoldható, ha egy eljárás 1 valószínűséggel megtalálja a globális optimumot, ha végtelen sokáig futhat. (gyenge konvergencia)

Ha csak elérhető globális minimum érdekel minket, ($p_1 > 0$), akkor könnyedén konstruálhatunk ilyen eljárást:

Tiszta véletlen keresés: Véletlenül generált mintapontokból lokális keresés, majd a legjobb lokális optimum kiválasztása.

Rács menti keresés: A keresési tartományon egy rács rácspontjaiból indítjuk a lokális keresést. A rács egyre sűrűbb lesz.

3.6. Feladat. Tekintsük a következő optimalizálási problémát:

$$\min_{x\in X}\left(x_2^2+\min\left\{(x_1-2)^2,(x_1+2)^2-0.1\right\}\right)\quad X=([-4,4],[-2,2]).$$

a) Mi f^* vonzási tartománya? ($f^*=-0.1$)

b) Mi az $Lf(1)$ szinthalmaz?

c) Mennyi p_1 közelítőleg?

d) Mi az f_2^* medencéje?

4. fejezet

Globális optimalizálási problémák és eljárások osztályozása

Ha egy adott probléma megoldását keressük, a probléma ismeretében választhatunk megoldó módszert. Például egydimenziós feladatokra sokkal több egyszerű módszer létezik, mint magasabb dimenziós problémákra. A feladat más tulajdonságai is sokat segíthetnek, ezért vegyük sorra a lehetséges szempontokat, osztályozásokat.

4.1. Problémák osztályozása

Szempontok:

- Dimenzionalitás: egydimenziós, többdimenziós
- Függvény tulajdonságok: konvex, konkáv, kvadratikus, Lipschitzes, stb.
- Korlátok tulajdonságai: feltétel nélküli($x \in \mathbb{R}^n$), intervallum korlátok($X = \{x \in \mathbb{R}^n | a \leq x \leq b\}$), egyenlőség/egyenlőtlenség feltételes, feltételek tulajdonságai: lineáris, kvadratikus, konvex, stb.
- Egyéb információk: analitikus függvény megadás, gradiens, Hesse–mátrix számíthatósága, lokális optimumok száma (megállási feltételt ad), f értékkészlete X-ben.

4.1. Megjegyzés. Feketedoboz-probléma: Nincs információnk a függvényről, csak egy kiértékelő „dobozunk” (eljárásunk).

4.2. Mintaproblémák

4.2.1. Csendes EX2 feladata

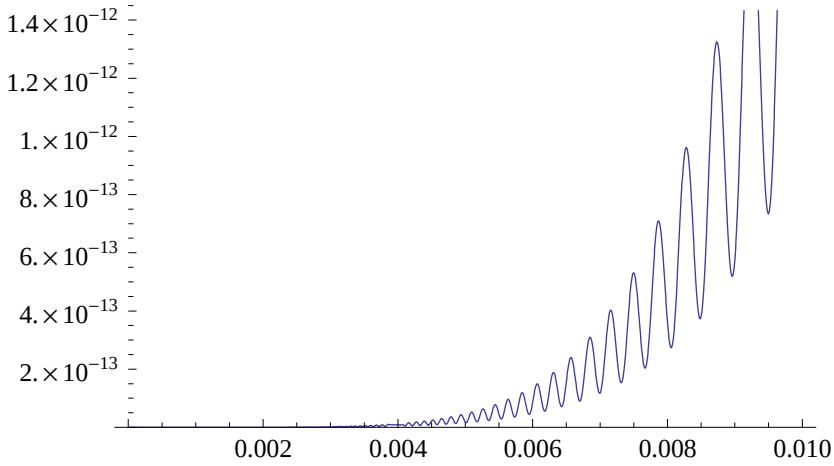
Ezt a feladatot bármely $n \in \mathbb{N}^+$ dimenzióban definiálhatjuk:

$$f(x) = \begin{cases} \sum_{i=1}^{n} x_i^6 \left(\sin \frac{1}{x_i} + 2\right) & \forall i : x_i \neq 0 \\ 0 & \text{egyébként} \end{cases}$$

Ha $n = 1$, akkor

$$f(x) = \begin{cases} x^6 \left(\sin \frac{1}{x} + 2\right) & x \neq 0 \\ 0 & \text{egyébként} \end{cases}$$

A globális minimum az $x = 0$ helyen van, a minimumérték 0. A $\sin \frac{1}{x}$ oszcillál a 0



4.1. ábra. A Csendes EX2 függvény oszcillál a nulla közelében

közelében. A 4.1. ábrán az $f(x)$ függvény $]0, 0.1]$ intervallumon felvett képe látható.

Ebben az esetben $p_1 = 0$, tehát az $f^* = 0$ minimum nem elérhető minimuma a függvénynek, holott folytonos a 0 közelében. Ez szemlélteti, hogy a folytonosság nem elégséges feltétele az elérhető minimumnak.

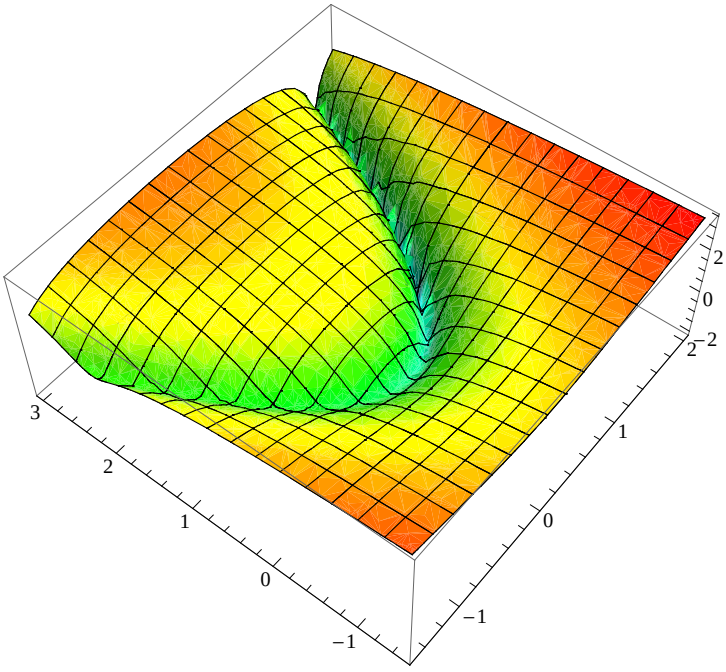
4.2.2. Rosenbrock függvény

A Rosenbrock, másnéven banán–függvény egy elterjedt tesztfüggvény nemlineáris optimalizáló eljárások teljesítményének mérésére.

$$f(x,y) := (1-x)^2 + 100(y-x^2)^2$$

A függvény négyzetek összege, könnyű látni, hogy $f(1,1) = 0$. Tehát az $(1,1)$ pont globális minimumhelye a függvénynek. Más lokális optimum nem létezik. A minimum egy parabolikus „völgyben" helyezkedik el, ahogy azt a 4.2. ábrán is láthatjuk.

Egy gradiens alapú optimalizáló eljárás általában gyorsan eljut a völgybe, azonban ott oszcilláció alakul ki a két meredek fal között, ami miatt az optimumhoz való konvergencia lassú lehet.



4.2. ábra. A Rosenbrock „banán"-függvény a logaritmustérben

4.2.3. Six-Hump-Camel-Back (SHCB) probléma

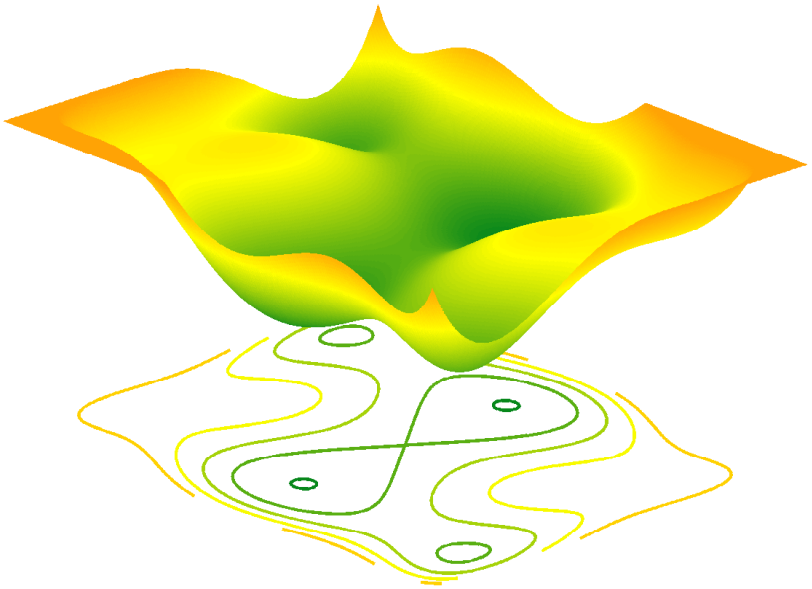
A Six-Hump-Camel-Back, azaz a hatpupú teve probléma kifejezetten globális optimalizáló módszerek tesztelésére lett kifejlesztve, hiszen 6 lokális minimummal rendelkezik, amiből kettő globális.

$$f(x,y) = (4 - 2.1x^2 + x^{4/3})x^2 + xy + (-4 + 4y^2)y^2$$

A 4.3. ábrán a függvény grafikonja és alatta kontúrjai is láthatók a $[-2{,}2]^2$ intervallumon.

4.3. Globális optimalizáló eljárások osztályozása

Az eljárások egy egyszerű osztályozása, ha azokat determinisztikus vagy probabilisztikus (sztochasztikus) csoportra bontjuk. Ez az osztályozás könnyű, mégsem szerencsés, hiszen nem árul el semmit az eljárással kapott eredményről. Egy részben hibás, de elterjedt nézet, hogy a determinisztikus módszerek egzaktak, míg a probabilisztikusak nem. Természetesen találhatunk ilyet is olyat is bármelyik osztályban.



4.3. ábra. A Six-Hump-Camel-Back függvény kontúrokkal

Heurisztikus eljárás: Minden olyan eljárás, amely képes egy közelítő megoldást adni, de nem bizonyítható sem valamely korlát a közelítés hibájára, sem az eljárás helyessége.

4.2. Megjegyzés. Heurisztika $\neq$ probabilisztikus eljárás. A raszter keresés például egy determinisztikus heurisztika általános esetben. Ez példa nem egzakt determinisztikus módszerre is.

A módszerekkel elérhető megoldás minősége alapján a következő osztályozást adhatjuk:

Nemteljes eljárás: Lehetséges, hogy nem a globális optimumhoz konvergál.

Aszimptotikusan teljes eljárás: Olyan eljárás, mely biztosan, vagy 1 valószínűséggel eléri a globális optimumot, ha végtelen ideig futhat, de soha nem tudhatjuk, hogy az aktuális megoldásunk milyen messze van a globális optimumtól.

Teljes eljárás: Olyan eljárás, mely egzakt aritmetikát feltételezve biztosan eljut a globális optimumhoz, ha végtelen ideig futhat, valamint véges sok lépést követően tudja, hogy egy közelítő megoldást talált a globális minimumra (előre megadott tűrésen belül).

Megbízható eljárás: Olyan teljes eljárás, amely kerekítési hibák mellett is teljes.

4.3. Megjegyzés. A megbízható, illetve teljes eljárásokat összefoglaló néven egzakt eljárásoknak nevezzük. Ilyen eljárásokhoz mindig szükséges valamilyen plusz in-

formáció a feladatról, teljesen általános esetre nem létezik algoritmus ezekben az osztályokban.

4.4. Példa. Nemteljes eljárások: Newton módszer, Genetikus algoritmus

5. fejezet

Lipschitz-optimalizálás

Az említett teljes eljárásokhoz tartoznak a Lipschitz tulajdonságon alapuló módszerek. Nézzük meg először is, hogy mi ez a tulajdonság.

5.1. Definíció (Lipschitz–folytonosság). Azt mondjuk, hogy egy $f : D \subset \mathbb{R}^n \to \mathbb{R}$ valós függvény minden változójában kielégíti a Lipschitz-feltételt (azaz Lipschitz-folytonos), ha

$$|f(x_1) - f(x_2)| \leq L \|x_1 - x_2\| \quad \forall x_1, x_2 \in D$$

valamely $L > 0$ konstansra.

5.2. Megjegyzés. Ha egy függvény Lipschitz valamely L konstansra, akkor bármely $L_1 > L$ konstansra is az.

5.3. Definíció. Egy optimalizálási feladat Lipschitz, ha a célfüggvény és a feltételekben szereplő függvények mindegyike Lipschitz-folytonos.

5.4. Tétel. *Legyen f folytonosan differenciálható függvény egy D nyílt halmazon és legyen X egy kompakt részhalmaza D-nek. Ekkor f Lipschitzes X-en.*

Bizonyítás. A Lagrange középértéktételből:

$$f(x) = f(\hat{x}) + \nabla f(\tilde{x})(x - \hat{x}) \quad \text{valamely } \tilde{x} \in [x, \hat{x}] \text{ pontra.}$$

Így

$$|f(x) - f(\hat{x})| = |\nabla f(\tilde{x})(x - \hat{x})| \leq \|\nabla f(\tilde{x})\| \cdot \|x - \hat{x}\|$$

Nyílt halmazon folytonos függvénynek létezik kompakt halmazon a maximuma, így

$$L := \max_{x \in X} \|\nabla f(x)\| .$$

5.5. Állítás. *Minden Lipschitz-folytonos függvény folytonos, de nem minden folytonos függvény Lipschitz.*

5.1. Lipschitz függvények tere

5.6. Tétel. *Legyen X egy kompakt halmaz, $\{f_j\}_{j\in\{1,\dots,n\}}$ Lipschitz függvények X-en, valamint h legyen egy egyváltozós Lipschitzes függvény f_j értékkészletén. Ekkor a következő függvények szintén Lipschitz folytonosak:*

$$\sum_{j=1}^{n} c_j f_j(x) \quad c_j \in \mathbb{R}$$
$$\min_{j\in\{1,\dots,n\}} f_j(x); \qquad \max_{j\in\{1,\dots,n\}} f_j(x);$$
$$f_i(x) f_j(x) \quad \forall i,j \in \{1,\dots,n\}$$
$$h(f_i(x)) \quad \forall i \in \{1,\dots,n\}$$

Bizonyítás. ($\max f_j(x)$ esetére)

$$\left|\max_j f_j(x_1) - \max_j f_j(x_2)\right| \le \max_j \left|f_j(x_1) - f_j(x_2)\right| \le \max_j L_j \|x_1 - x_2\|$$

ahol L_j az f_j függvényhez tartozó Lipschitz-konstans. ∎

5.7. Megjegyzés. Legegyszerűbb teljes eljárás Lipschitz függvényekre: A rács menti kereséssel δ maximális távolsággal

$$\tilde{f} = \min_{i<K} f(x_i)$$

közelítése az f^* optimumnak. $\delta < \frac{\varepsilon}{L}$ választással

$$\left|\tilde{f} - f^*\right| < \varepsilon.$$

5.2. Pyavskii-Schubert algoritmus

Ez az algoritmus egydimenziós intervallum korlátos Lipschitz feladatok megoldására készült. Az iterációk során egy fűrészfog függvényt közelít alulról a célfüggvényhez. A Lipschitz tulajdonság miatt egy pontot használva az

$$F(x) = f(\hat{x}) - L\|x - \hat{x}\| \quad \hat{x} \in X$$

függvény alsó becslést ad $f(x)$-re X-en. Az algoritmus intervallum korlátokra adott, legyen például $X = [a,b]$. Az algoritmus a következő lépéseket teszi:

$$x_1 := \frac{a+b}{2}, X = [a,b]$$
$$F_1(x) = f(x_1) - L|x - x_1|$$
$$x_2 = \arg\min_{x\in X} F_1(x) \quad x_2 \in \{a,b\}$$
$$F_2(x) = \max_{i\in\{1,2\}} \{f(x_i) - L|x - x_i|\}$$

$$x_{k+1} = \arg\min_{x\in X} F_k(x) \qquad F_k(x) = \max_{i\in\{1,\dots,k\}} \{f(x_i) - L\,|x - x_i|\}$$

5.8. Megjegyzés. Ha $\tilde{f}$ az f^* globális optimum egy felső korlátja, például $\tilde{f} = \min_{i\in\{1,\dots,n\}} f(x_i)$, akkor az

$$A_k = \left\{x \in X \mid F_k(x) \le \tilde{f}\right\}$$

halmaz tartalmazza a globális optimumhelyet.

5.9. Példa. Legyen

$$f(x) = \sin x + \sin 3x + \ln x \qquad X \in [3{,}7].$$

Minimalizálni akarunk a Pyavskii-Schubert algoritmussal. Először is adjunk becslést az L Lipschitz konstansra.

Megoldás: Definíció szerint

$$|f(x_1) - f(x_2)| \le L\,\|x_2 - x_1\| \qquad \forall x_1, x_2 \in X.$$

Minden folytonosan differenciálható függvényhez $\exists L$ Lipschitz konstans egy kompakt tartományon. Nem szükséges feltétlenül a legkisebb Lipschitz konstanst megadni, ezért tagonként becslünk.

$$L = \max_{x\in X} \|\nabla f\|$$

$$f'(x) = \cos x + 3\cos 3x + \frac{1}{x}$$

$$\left|f'(x)\right| \le |\cos x| + |3\cos 3x| + \left|\frac{1}{x}\right| \le 1 + 3 + \frac{1}{3}$$

Tehát $L = \frac{13}{3}$ egy jó becslés.

5.10. Példa. Legyen f Lipschitz függvény $[a,b]$-n, egy Lipschitz konstans pedig L. Mutassuk meg, hogy ha $\exists x_1, x_2 \in [a,b]$ amire

$$f(x_1) = f(x_2) - L(x_2 - x_1) \quad x_2 > x_1$$

akkor

$$f(x) = f(x_2) - L(x_2 - x) \quad \forall x \in [x_1, x_2]$$

Megoldás: Legyen $x \in (x_1, x_2]$. Ekkor a Lipschitz tulajdonság, és a feltételünk alapján

$$\begin{aligned} |f(x_1) - f(x)| &\le L\,|x - x_1| \\ -L(x - x_1) &\le f(x_1) - f(x) \\ -L(x - x_1) &\le f(x_2) - L(x_2 - x_1) - f(x) \\ L(x_2 - x) &\le f(x_2) - f(x) \end{aligned}$$

Másrészt, $f(x_2) - f(x) \leq L(x_2 - x)$ szintén a Lipschitz tulajdonság miatt, amiből

$$f(x) = f(x_2) - L(x_2 - x) \quad \forall x \in [x_1, x_2].$$

5.11. Példa. Tekintsük az

$$f(x) = x^4 - 12x^3 + 47x^2 - 60x$$

függvényt. Adjunk becslést az L Lipschitz konstansra a $[0,6]$ intervallumon, illetve számítsuk ki x_5-öt a Pyavskii-Schubert algoritmussal!

Megoldás: Kiszámítjuk a deriváltfüggvényt, és azt becsüljük.

$$\begin{aligned} f'(x) &= 4x^3 - 36x^2 + 94x - 60 \\ L &= \max_{x \in X} \left| f'(x) \right| \leq \max_{x \in X} \left| 4x^3 \right| + \max \left| -36x^2 \right| + \max |94x| + |-60| = \\ &= 4 \cdot 6^3 + 36 \cdot 6^2 + 94 \cdot 6 + 60 = 5808 \end{aligned}$$

de ez nagyon nagy. Mivel a derivált abszolút értékét becsüljük, ezt megtehetjük úgy is, hogy a két szélsőértéket vizsgáljuk.

$$\begin{aligned} \max_{x \in X} \left| f'(x) \right| &= \max\{\max f'(x), -\min f'(x)\} \leq \max\{\max_{x \in X}(4x^3) + \max(-36x^2) + \\ &\quad + \max(94x) - 60, -\min(4x^3) - \min(-36x^2) - \min(94x) + 60\} = \\ &= \max\{4 \cdot 6^3 + 0 + 94 \cdot 6 - 60, 0 + 36 \cdot 6^2 - 0 + 60\} = \max\{1368, 1356\} \end{aligned}$$

de ez még mindig túl nagy. A második derivált vizsgálatával kiderül, hogy a $[0,6]$ intervallumon a derivált az intervallum végpontjában veszi fel a maximumát, így $L = 72$ már jó konstans. Vegyük $L = 100$-at. Számítsuk ki x_5-öt:

$$\begin{aligned} x_1 &= 3 \qquad f(x_1) = 0 \\ x_2 &= 0 \qquad f(x_2) = 0 \\ x_3 &= 6 \qquad f(x_3) = 36 \end{aligned}$$

Mivel $f(x_1) = f(x_2)$,

$$\begin{aligned} \min_{x \in [x_2, x_1]} F_4(x) &= f(x_1) - \frac{|x_2 - x_1|}{2} L = 0 - \frac{3}{2} \cdot 100 = -150 \\ \min_{x \in [x_1, x_3]} F_4(x) &: \quad f(x_1) - L(x - x_1) = f(x_3) - L(x_3 - x) \\ f(x_1) - f(x_3) &= 2Lx - Lx_1 - Lx_3 = 2Lx - L(x_1 + x_3) \\ x &= \frac{f(x_1) - f(x_3)}{2L} + \frac{x_1 + x_3}{2} \qquad x_1 < x_3 \\ \min_{x \in [x_1, x_3]} F_4(x) &= f(x_1) - L\frac{f(x_1) - f(x_3)}{2L} - L\frac{x_1 + x_3}{2} + Lx_1 = \end{aligned}$$

$$= \frac{f(x_1)+f(x_3)}{2} + L\frac{x_1-x_3}{2} = 18 - 150 = -132, \quad x_1 < x_3$$
$$\Rightarrow x_4 \in [x_1, x_2], \quad x_4 = \frac{x_1+x_2}{2} = \frac{3}{2}, \quad f(x_4) = -19.6875.$$

A fenti képletekkel számítva $\min_{x\in[x_2,x_4]} F_5(x) = \frac{f(x_2)+f(x_4)}{2} - L\frac{|x_2-x_4|}{2} = -9.84375 - -100\cdot\frac{3}{4} = -84.84375$, illetve $\min_{x\in[x_4,x_1]} F_5(x) = \frac{f(x_4)+f(x_1)}{2} - L\frac{|x_4-x_1|}{2} = -9.84375 - -100\cdot\frac{3}{4} = -84.84375$. Vagyis $x_5 \in [x_1, x_3]$, és ahogy azt korábban felírtuk $x_5 = \frac{f(x_1)-f(x_3)}{2L} + \frac{x_1+x_3}{2} = -18/200 + 4.5 = 4.41$, ahol $f(x_5) = -1.50$.

A Pyavskii-Schubert algoritmus pszeukódját a 5.2.1 algoritmus írja le. Legyen $\mathcal{L}$ egy (F_{ij}, x_i, x_j) 3-asokat tartalmazó rendezett lista, ahol x_i, x_j két egymást követő pont, és $F_{ij} = \min_{x\in[x_i,x_j]} F_k(x)$ alsókorlát ezen pontok között. Pozitív ε-ra az algoritmus megállásakor $\tilde{f}$ maximum ε-nal tér el a globális optimumtól, az $\mathcal{L}$ lista olyan pontpárokat tartalmaz, amik között megtalálható az összes globális optimumhely, és tartalmazzák az $L_f(\tilde{f})$ szinthalmazt is.

5.2.1. algoritmus Pyavskii-Schubert

Bemenet: $f, [a, b]$

$x_1 = a, x_2 = \frac{a+b}{2}, x_3 = b, k = 3$

$F_{12} = \dfrac{f(x_1)+f(x_2)}{2} - L\dfrac{|x_1-x_2|}{2}$

$F_{23} = \dfrac{f(x_2)+f(x_3)}{2} - L\dfrac{|x_2-x_3|}{2}$

$\mathcal{L} = \{(F_{12}, x_1, x_2), (F_{23}, x_2, x_3)\}$ *$\mathcal{L}$ alsókorlátok szerint növekvő lista*

$\tilde{f} = \min\{f(x_1), f(x_2), f(x_3)\}$

$\tilde{x} = \arg\min\{f(x_1), f(x_2), f(x_3)\}$

while $\tilde{f} - \min\limits_{F_{ij}\in\mathcal{L}} F_{ij} > \varepsilon$ **do**

$k = k + 1$

Vegyük le az első (F_{ij}, x_i, x_j) 3-ast az $\mathcal{L}$ listáról

$x_k = \dfrac{x_i+x_j}{2} + \dfrac{f(x_i)-f(x_j)}{2L}$

if $f(x_k) < \tilde{f}$

$\tilde{f} = f(x_k);\ \tilde{x} = x_k$ *Frissítjük a felsőkorlátot*

Töröljük $\mathcal{L}$-ről az $\tilde{f}$-nál nagyobb alsókorlátú 3-asokat

Értékeljük ki az F_{ik}, F_{kj} alsókorlátokat

Szúrjuk be $\mathcal{L}$-be (F_{ik}, x_i, x_k)-t, (F_{kj}, x_k, x_j)-t az alsókorlát szerint

6. fejezet

D.C. Programozás

D.C - konvex függvények különbsége (Difference of Convexes)

6.1. Definíció. Egy $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$ (D konvex) függvényt d.c. függvénynek nevezünk, ha léteznek olyan p és q konvex függvények D-n, hogy

$$f(x) = p(x) - q(x) \quad \forall x \in D.$$

6.2. Definíció (D.C. optimalizálási feladat). Keressük min $f(x)$-et, ahol

$$x \in C, g_j(x) \leq 0 \quad \forall j = 1, \ldots, m$$

és C zárt, konvex halmaz, f, g_j pedig mind d.c. függvények.

6.3. Megjegyzés. Mire lehet jó egy d.c. felbontás? Például számíthatunk alsó és/vagy felső korlátokat egy d.c. függvényre konvex halmazon, poliéderen. Legyen $f(x) = p(x) - q(x)$ egy d.c. felbontás, ekkor bármely x pontban

$$\mathrm{Lb}(f(x)) = \mathrm{Lb}(p(x)) - \mathrm{Ub}(q(x))$$

ahol Lb jelöli az alsó, Ub a felső korlátot. Legyen v_i, v_j két pont, ekkor az összekötő szakaszra $\mathrm{Ub}(q(x)) = \lambda q(v_i) + (1-\lambda) q(v_j)$, hiszen q konvex. Egy $v_1, \ldots, v_m$ csúcspontú konvex poliéderre $\mathrm{Ub}(q(x)) = \sum_i \lambda_i q(v_i)$, ahol $x = \sum_i \lambda_i v_i$. Vagyis $Ub(q(x)) = \max_i q(v_i)$. Az alsó korlátot adhatja egy érintősík valamely x' pontra számítva $\mathrm{Lb}(p(x)) = p(x') + \nabla p(x')^T (x - x')$.

6.1. D.C. függvények tere

6.4. Állítás. *Legyenek $f, f_i (i = 1, \ldots, m)$ d.c. függvények. Ekkor az alábbi függvények szintén d.c.-beliek:*

$$\sum_{i=1}^{m} \lambda_i f_i(x), \qquad \lambda_i \in \mathbb{R},$$

$$\max_{i=1,\dots,m} f_i(x), \qquad \min_{i=1,\dots,m} f_i(x),$$
$$\prod f_i(x), \qquad |f(x)|, \qquad f^+(x), \qquad f^-(x).$$

Bizonyítás. Például a $\max_i f_i(x)$ függvényre

$$\max_i f_i(x) = \max_i (p_i(x) - q_i(x)) = \max_i \left(p_i(x) + \sum_{i \neq j} q_j(x) \right) - \sum_j q_j(x),$$

mert konvex függvények összege illetve maximuma konvex. ■

6.5. Definíció. Egy függvény lokálisan d.c., ha $\forall x_0 \in \mathbb{R}^n$ ponthoz létezik $M_\varepsilon(x_0)$ környezet, hogy f d.c. $M_\varepsilon(x_0)$-on.

6.6. Tétel. *Minden lokálisan d.c. függvény d.c.*

6.7. Következmény. *Minden $f : \mathbb{R}^n \to \mathbb{R} \in C^2$ függvény d.c.*

Bizonyítás. A $\nabla^2 f$ Hesse–mátrix minden eleme korlátos, bármely $M_\varepsilon(x_0)$ környezeten. Ezért egy kellően nagy $\mu > 0$ esetén $f(x) + \mu \|x\|^2$ konvex $M_\varepsilon(x_0)$-on, mivel a $\nabla^2 f + 2\mu \cdot I$ pozitív szemidefinit ha μ elég nagy (I az egységmátrix). Így az

$$f(x) = \left(f(x) + \mu \|x\|^2 \right) - \mu \|x\|^2$$

az f egy d.c. felbontása $M_\varepsilon(x_0)$-on, és a fenti tétel miatt bárhol, ha μ-t elég nagyra választjuk. ■

6.8. Megjegyzés. Egy $g \circ f$ kompozíció d.c., ha f d.c. és g konvex függvény.

6.9. Példa. Legyen $f(x_1, x_2) = x_1 x_2$. Adjuk meg f d.c. dekompozícióját.

$$p(x) = \frac{(x_1 + x_2)^2}{2} \qquad q(x) = \frac{x_1^2}{2} + \frac{x_2^2}{2}$$

Legyen $f(x) = p(x) - q(x)$ az f d.c. felbontása. Adjunk alsó becslést $f(x)$-re az $X = ([-1,2],[1,3])$ intervallumon.

$$\begin{aligned} \mathrm{Lb}(f(x)) &= \mathrm{Lb}(p(x)) - \mathrm{Ub}(q(x)) \\ \mathrm{Ub}(q(x)) &= \max q(v_i) = \frac{2^2}{2} + \frac{3^2}{2} = 6.5 \end{aligned}$$

Az $\mathrm{Lb}(p(x))$ alsó becsléshez:

$$\nabla p(x) = \begin{pmatrix} x_1 + x_2 \\ x_1 + x_2 \end{pmatrix}, \qquad \nabla p(0,1) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$p(0,1) + \nabla p(0,1)^T \left(x - \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right) = \frac{1}{2} + \begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 - 1 \end{pmatrix} = x_1 + x_2 - \frac{1}{2}$$

A most felírt hipersík alsó korlátot ad a függvényre, így a $-\frac{1}{2}$ jó alsó becslés. Ezzel

$$\mathrm{Lb}(f) = -0.5 - 6.5 = -7$$

(Jelen esetben ez persze nem éles, nyilvánvaló, hogy a $p(x)$-re a nulla egy természetesen adódó alsó korlát, de ez nem mindig ilyen triviális.

6.2. Kanonikus D.C. programozás

6.10. Definíció. Kanonikus d.c. programozási feladatnak nevezzük a

$$\begin{aligned} \min_{x \in C} \quad & c^T x \\ \text{f.h.} \quad & g(x) \geq 0 \end{aligned}$$

alakú feladatot, ahol $c \in \mathbb{R}^n$, C zárt, konvex halmaz és $g : \mathbb{R}^n \to \mathbb{R}$ konvex függvény. A $g(x) \geq 0$ feltételt fordított konvex feltételnek is nevezik a nagyobb-egyenlőség miatt.

6.11. Állítás. *Minden d.c. program felírható kanonikus alakban.*

Bizonyítás. Legyen a feladatunk a

$$\begin{aligned} \min_{x \in D} \quad & f(x) \\ \text{f.h.} \quad & g_j(x) \leq 0 \ (j = 1, \ldots, m) \end{aligned}$$

ahol f, g_j mind d.c., D zárt, konvex halmaz. Ez ekvivalens az alábbi feladattal:

$$\begin{aligned} \min_{x \in D} \quad & t \\ \text{f.h.} \quad & g_0(x) = f(x) - t \leq 0 \\ & g_j(x) \leq 0 \ (j = 1, \ldots, m) \end{aligned}$$

Ekkor t minimalizálásával egyben $f(x)$-et is minimalizáljuk.

$$g_j(x) \leq 0 \ (j = 0, \ldots, m) \iff \max_{j=0,\ldots m} g_j(x) \leq 0$$

Mivel f és minden g_j d.c., és d.c. függvények maximuma is d.c. függvény, létezik a $\max_{j=0,\ldots m} g_j(x) = p(x) - q(x)$ felbontás. A kapott $p(x) - q(x) \leq 0$ feltétel ekvivalens a

$$\begin{aligned} \varphi(x,s) &= p(x) - s \leq 0 \\ \psi(x,s) &= q(x) - s \geq 0 \end{aligned}$$

konvex feltételekkel, mivel $p(x) \leq s,\ q(x) \geq s \Rightarrow p(x) \leq q(x)$. Innen

$$C = \{x \in D : \varphi(x,s) \leq 0\} \text{ konvex halmaz}$$

és így az új kanonikus alakú feladatunk felírható

$$\begin{array}{ll} \min\limits_{y \in C} & c^T y, \quad c = (0,\ldots,0,1,0) \\ \text{f.h.} & \psi(y) \geq 0 \\ & C = \{y = (x, x_{n+1}, x_{n+2}) \mid x \in D, \varphi(x, x_{n+2}) \leq 0\} \end{array}$$

■

A kanonikus d.c. feladatok optimális megoldása karakterizálható, ha néhány gyenge feltétel teljesül.

A_1 *feltétel:* A $g(x) \geq 0$ feltétel lényeges, azaz létezik $x^0 \in C$, hogy $g(x^0) < 0$ és $c^T x^0 < \min_{x \in C, g(x) \geq 0} c^T x$.

A_2 *feltétel:* Az $F = \{x \in C \mid g(x) \geq 0\}$ megengedett megoldások halmazára cl(int(F))=F, azaz legyen robusztus halmaz. Ez tulajdonképpen azt jelenti, hogy F teljes dimenziós belsejében nem üres halmaz.

Az A_1 feltétel nem megszorító, hiszen ha nem teljesül akkor a feladatból a fordított konvex feltétel elhagyható, vagyis egyszerűbb feladatot kapunk.

Az A_2 feltétel azokat az eseteket akarja kizárni, amikor a $G = \{x \mid g(x) \geq 0\}$ és a C halmaz metszete C határának a része.

6.12. Tétel. *Tegyük fel egy kanonikus d.c. problémára, hogy C korlátos, F nemüres és a fordított konvex feltétel lényeges. Ekkor a feladatnak létezik optimális megoldása a C és G halmazok határának metszetén.*

6.13. Feladat. Bizonyítsuk be a 6.12 tételt!

6.3. Egy élkövető algoritmus

A következő kanonikus alakú feladat megoldását keressük:

$$\begin{array}{ll} \min\limits_{x \in C} & c^T x \\ \text{f.h.} & g(x) \geq 0 \end{array}$$

ahol C egy politóp, amelynek belseje nem üres, illetve feltesszük hogy az $A_1 - A_2$ feltételek teljesülnek. A 6.12 tétel alapján az optimális megoldás C egy élének és G határának metszetén van.

A módszert röviden a 6.3.1 algoritmus írja le. Lényegében az F halmazt közelítjük egy politóppal, amíg meg nem találjuk az optimális megoldást.

6.3.1. algoritmus Élkövető algoritmus

Inicializáció: Oldjuk meg a $\min_{x\in C} c^T x$ LP-t. Legyen x^0 a megoldás.
Legyen $S^0 \supseteq C$ egy n-szimplex, V^0 a csúcshalmaza, hogy $x^0 \in V^0$.
$z = \operatorname{argmax}_{v\in V^0} g(v)$
if $g(z) = 0$
    STOP, $F = \emptyset$
$y := x^0$,
for $k = 0,1,2,\ldots$ **do**
    if $g(z) < 0$
        STOP, $F = \emptyset$
    if $g(z) = 0$
        STOP, y optimális
    A szimplex algoritmussal z-ből indulva keressünk olyan $[u,v]$ élét S^k-nak, hogy $g(u) \geq 0, g(v) \leq 0$, valamint $c^T v < c^T u$.
    Határozzuk meg $s \in [u,v]$, amire $g(s) = 0$.
    if $s \in C$
        $y = s, S^{k+1} = S^k \cap \{x : c^T x \leq c^T y\}$
    else
        $S^{k+1} = S^k \cap \{x : l(x) \leq 0\}$, ahol $l(x) \leq 0$ meghatározza a C halmazt, és $l(s) > 0$
    Kiszámítjuk a V^k csúcshalmazt, és a $z = \operatorname{argmax}_{v\in V^k} g(v)$ pontot.

6.14. Példa. Oldjuk meg az élkövető algoritmussal a következő kanonikus feladatot

$$\begin{aligned} \min -\frac{1}{2}x_1 - \frac{1}{6}x_2 \\ x_2 - x_1 &\leq 2 \\ \frac{1}{3}x_1 + \frac{1}{2}x_2 &\geq 1 \\ 0 \leq x_1 &\leq 6 \\ 0 \leq x_2 &\leq 4 \\ x_1^2 + x_2^2 - 8x_1 - 6x_2 + 16 &\geq 0 \end{aligned}$$

A fordított konvex feltételt átírjuk: $(x_1-4)^2 + (x_2-3)^2 \geq 3^2$. A kezdeti LP feladat grafikusan megoldva a $x^0 = (6,4)$ megoldást kapjunk. Legyen a kezdő szimplex a $V^0 = \{(6,4),(6,-2),(-3,4)\}$ csúcshalmazzal adott. Ekkor $z = (-3,4)$, hiszen

$$\max_{v\in V^0} g(v) = \max\{g(6,4) = -4, g(-3,4) = 41, g(6,2) = 20\}.$$

Grafikusan z-ből indulva $(u,v) = (z,x^0)$, ahonnan $s = (1.172,4)$. Mivel $s \notin C$, $S^1 = S^0 \cap \{x \mid x_2 - x_1 - 2 \leq 0\}$ (olyan feltételt kell választani, ami meghatározza C-t, és kizárja s-et. Az új csúcshalmaz

$$V^1 = \{(6,4),(6,-2),(0,2),(2,4)\},$$

ahonnan az új $z = (6, -2)$. Újra grafikusan nézve $u = z^1, v = x^0$, így $s = (6,0.764)$. Most $s \in C$ így $y = s, S^2 = S^1 \cap \{x : -\frac{1}{2}x_1 - \frac{1}{6}x_2 \leq -3.127\}$. Az új csúcshalmaz $v^2 = \{(6,4), (6,0.764), (4.92, 4)\}$ amire $z = (6,0.764)$ és $g(z) = 0$, vagyis megállunk, (6,0.764) a megoldás.

6.15. Példa. Legyen $f : \mathbb{R} \to \mathbb{R}$ konkáv a $(-\infty, a)$ intervallumon, és konvex az $[a, \infty)$ intervallumon. Adjunk meg f d.c. felbontását az f' derivált segítségével egy adott pontban.

$$
\begin{aligned}
f(x) &\approx f(a) + f'(a)(x-a) \\
p(x) &= \begin{cases} f(a) + f'(a)(x-a) & \text{ha } x < a \\ f(x) & \text{ha } x \geq a \end{cases} \\
f(x) &= p(x) - q(x) \\
q(x) &= p(x) - f(x) \\
q(x) &= \begin{cases} -f(x) + f(a) + f'(a)(x-a) & \text{ha } x < a \\ 0 & \text{ha } x \geq a \end{cases}
\end{aligned}
$$

Másik megoldás:

$$
\begin{aligned}
p_2(x) &= \begin{cases} \frac{1}{2}\left(f(a) + f'(a)(x-a)\right) & x < a \\ f(x) - \frac{1}{2}\left(f(a) + f'(a)(x-a)\right) & x \geq a \end{cases} \\
q_2(x) &= \begin{cases} -f(x) + \frac{1}{2}\left(f(a) + f'(a)(x-a)\right) & x < a \\ \frac{1}{2}\left(f(a) + f'(a)(x-a)\right) & x \geq a \end{cases}
\end{aligned}
$$

7. fejezet

Korlátozás és szétválasztás módszere

Szétválasztás: A feladatot részfeladatokra osztjuk.

Korlátozás: A feldolgozás során alsó korlátokat állapítunk meg a globális optimumra, ami révén az optimumot biztosan nem tartalmazó részfeladatok kiküszöbölhetjük.

7.1. Prototipus algoritmus

A korlátozás és szétválasztás módszer általános algoritmikus leírását a 7.1.1 algoritmusban láthatjuk. Itt a következő jelöléseket használjuk:

$\mathcal{L}$: a részfeladatok listája

$\tilde{f}$: az aktuális felső korlát a globális minimumra

$\mathrm{Lb}(Y)$: alsó korlát f-re az Y halmazon

$\mathrm{w}(Y)$: az Y halmaz szélessége, átmérője

7.1.1. algoritmus Korlátozás és szétválasztás módszere

Inicializálás: $\mathcal{L} = \{X\}$, $\tilde{f} = \infty$
while $\mathcal{L} \neq \emptyset$ **do**
  Kiválasztjuk és levesszük Y-t $\mathcal{L}$-ről — *kiválasztási szabály*
  Kiértékeljük $f(v)$-t valamely $v \in Y$ pontra
  $\tilde{f} = \min\{\tilde{f}, f(v)\}$
  Y-t felosztjuk $Y_1, \ldots, Y_p$ részhalmazokra. — *felosztási szabály*
  for $i = 1, \ldots, p$ **do**
    $\mathrm{Lb}(Y_i)$ meghatározása — *korlátozási szabály*
    if $\mathrm{Lb}(Y_i) \leq \tilde{f}$ — *kivágási szabály*
      Y_i-t hozzávesszük $\mathcal{L}$-hez.

A 7.1.1 algoritmusban megfogalmazott szabályok leggyakrabban használt megvalósításait a következő felsorolásokban gyűjtöttük össze.

Kiválasztási szabály:

a) Legkisebb alsó korlát alapján: $\operatorname{argmin}_{Y\in\mathcal{L}} \mathrm{Lb}(Y)$

b) Legnagyobb szélesség alapján: $\operatorname{argmax}_{Y\in\mathcal{L}} \mathrm{w}(Y)$

c) Legrégebben vizsgált (FIFO – First In First Out lista)

d) Véletlen kiválasztás

Felosztási szabály: Általában megköveteljük, hogy

$$Y = \bigcup_{i=1}^{p} Y_i \qquad Y_i \cap Y_j = \partial Y_i \cap \partial Y_j, \quad \forall i \neq j$$

a) felezés (általában a legnagyobb oldalnál/kiterjedésnél felezünk)

b) darabolás (több egyforma méretű darabra vágás, ez lehet egy vagy több irány szerint is)

Jó esetben Y olyan halmaz, amit önmagához hasonló halmazokra oszthatunk, mint például a szimplex, hypertégla, vagy végtelen kúp.

Korlátozási szabály: Az aktuális részfeladaton korlátokat számítunk a célfüggvény értékére. Ezt mindig az aktuális algoritmus határozza meg, nem általánosítható.

Kivágási szabály: Egy mindig alkalmazható kivágási szabály, hogy eltávolítunk $\forall Y \in \mathcal{L}$-t, amire

$$\mathrm{Lb}(Y) > \tilde{f}.$$

Ha feltételekkel korlátozott problémánk van, akkor a nem megengedett megoldásokat is eltávolítjuk, illetve megfogalmazhatunk más információra támaszkodó kivágási teszteket is. Például Lipschitz probléma esetén a 5.8 megjegyzés alapján.

7.1. Megjegyzés. Bizonyos esetekben a b) és c) kiválasztási szabály megegyezik. Találjunk erre példát!

Legyen az $\tilde{f}, Y, v, \mathcal{L}$ négyes $\tilde{f}_k, Y_k, v_k, \mathcal{L}_k$ a k. iteráció után.

7.2. Tétel. *Ha minden végtelen* $\{Y_{k_q}\} : Y_{k_q} \supset Y_{k_{q+1}} \quad (q = 1,2,\dots)$ *finomodó partíció-halmazokra a korlátok kielégítik a*

$$\lim_{q\to\infty} (\tilde{f}_{k_q} - \mathrm{Lb}(Y_{kq})) = 0$$

feltételt, akkor

$$\lim_{k\to\infty} \mathrm{Lb}(\mathcal{L}_k) = \lim_{k\to\infty} f(v_k) = \lim_{k\to\infty} \tilde{f}_k$$

és $\{v_k\}$ *minden* v^* *torlódási pontja globális optimuma a* $\min_{x\in X} f(x)$ *problémának, ahol* $\mathrm{Lb}(\mathcal{L}_k) = \min_{Y\in\mathcal{L}_k} \mathrm{Lb}(Y)$.

Bizonyítás. Létezik $\{v_k\}$-nak torlódási pontja, mert X kompakt. Legyen v^* a $\{v_k\}$ sorozat torlódási pontja, ekkor létezik olyan $\{v_{k_q}\}$ részsorozat, ami v^*-hoz konvergál, hogy

$$Y_{k_q} \supset Y_{k_{q-1}} \qquad v_{k_q} \in Y_{k_q} (q = 1,2,\dots)$$

Így f folytonossága miatt

$$\lim_{q\to\infty} f(v_{k_q}) = f(v^*)$$

Az $\mathrm{Lb}(\mathcal{L}_k)$ sorozat monoton növekvő és korlátos az f^* globális minimum által. Ezért a $k \to \infty$ határértéke létezik. Mindemellett $\tilde{f}_k$ monoton csökkenő és felülről korlátolt f^* által, így van határértéke. Az alsó korlát szerinti kiválasztást használva, és a

$$\lim_{q\to\infty} \left(\tilde{f}_{k_q} - \mathrm{Lb}(Y_{k_q})\right) = 0$$

feltétel miatt

$$\lim_{k\to\infty} \mathrm{Lb}(\mathcal{L}_k) = f^* = \lim_{k\to\infty} f(v_k) = f(v^*) = \lim_{k\to\infty} \tilde{f}_k.$$

7.3. Megjegyzés. Ugyanez belátható a méret és élettartam szerinti kiválasztásra is. Sőt, az is igaz, hogy ekkor X^* megegyezik a torlódási pontok halmazával.

Tegyük fel, hogy $\mathrm{Lb}(Y_1) = f^*$, de $\mathrm{Lb}(Y_k) < f^*$, minden $k = 2,3,\dots$ esetén. Ekkor $\mathrm{Lb}(Y_k) \to f^*$, de sosem lesz egyenlő. Mindig Y_k-t felezem, Y_1-et sosem választom ki. Ha viszont méret, vagy kor szerint választok, akkor mindegyik globális optimumot megtaláljuk.

7.1.1. A Lipschitz-optimalizálás korlátozási szabálya

Legyen az f Lipschitz függvény és az Y tartomány adott, szükségünk van $\mathrm{Lb}(Y)$-ra. Legyen Y középpontja c, és $\mathrm{w}(Y)$ az Y tartomány átmérője. Ha f Lipschitz-folytonos, akkor bármely $x, y \in Y$ pontra

$$|f(x) - f(y)| \le L \|x - y\|,$$

és így a tartomány középpontjában véve

$$\mathrm{Lb}(Y) = f(c) - L\frac{\mathrm{w}(Y)}{2}.$$

Ha az Y tartomány túl nagy, akkor a fenti korlát semmitmondó lehet. Ilyenkor vegyünk $v_1, \dots, v_l \in Y$ mintapontokat, és számítsunk korlátot a következőképpen:

$$\mathrm{Lb}(Y) = \min_{i=1,\dots,l} (f(v_i) - L \max_{x\in Y} \|v_i - x\|).$$

Könnyen belátható, hogy még a középpontra számított alsókorlátnál is egy Y_{k_q} egymásba ágyazott halmazsorozatra ($c(Y_{k_q})$ a középpont), amire $\lim_{q\to\infty} \mathrm{w}(Y_{k_q}) = 0$,

$$\lim_{q\to\infty} \left(\tilde{f}_{k_q} - \mathrm{Lb}(Y_{k_q})\right) = 0,$$

hiszen $\max_{x \in Y_{k_q}} \left\| c(Y_{k_q}) - x \right\| \to 0$, és így $\mathrm{Lb}(Y_{k_q}) \to f(c(Y_{k_q})) \geq \tilde{f}_{k_q}$. Vagyis a Lipschitz konstanssal számított alsókorlát eleget tesz a 7.2 Tétel feltételének.

7.4. Definíció. Egy felosztás kimerítő, ha minden beágyazott $\{Y_{k_q}\}$ végtelen sorozatra

$$\mathrm{w}(Y_{k_q}) \to 0.$$

7.5. Definíció. Tekintsünk egy tetszőleges Y n-szimplexet, $V(Y) = \{v_0, \ldots, v_n\}$ csúcshalmazzal. Ekkor az adott szimplex radiális felosztásán adott $w \in Y, w \notin V(Y)$ pontra az

$$Y_i = \mathrm{conv}\{v_0, \ldots, v_{i-1}, w, v_{i+1}, \ldots, v_n\}$$

n-szimplexeket értjük, amelyekre nem $Y_i \subset \partial Y$. Vagyis ha $w \in \partial Y$, azaz Y határán van, csak azokra az i-kre kapunk új n-szimplexet, amelyekre a

$$w = \sum_{i=0}^{n} \lambda_i v_i, \qquad \lambda_i \geq 0 \quad \forall i \text{ és } \sum_{i=0}^{n} \lambda_i = 1$$

reprezentációban $\lambda_i > 0$.

7.6. Példa. Vegyünk egy szabályos 3-szimplexet, azaz egy szabályos háromszöget. Legyen ez Y $\{v_1, v_2, v_3\}$ csúcshalmazzal. Lássuk be, hogy amennyiben radiális felosztásnál w-t mindig a háromszög belsejéből választjuk, akkor a felosztás nem lesz kimerítő. Az is belátható, hogy ha w-t mindig „ugyanarról" az élről választjuk, akkor a felosztás szintén nem lesz kimerítő.

Kimerítő felosztáshoz vezet viszont, ha w-t mindig, vagy legalábbis végtelen sokszor, a felosztandó háromszög leghosszabb oldalának középpontjaként választjuk.

8. fejezet

Intervallum analízis

A legfontosabb tulajdonsága, hogy automatikusan korlátot szolgáltat egy szubrutin által szolgáltatott függvényre intervallum bemenettel.

8.1. Aritmetikai műveletek intervallumokon

Legyen A és B véges és zárt intervallum, és $\circ \in \{+, -, \cdot, /\}$. Az intervallumos aritmetikai műveletektől természetesen megkövetelük, hogy

$$A \circ B = \{a \circ b \mid a \in A, b \in B\},$$

ahol $0 \notin B$ ha $\circ = /$.

Adott $[a,b]$ és $[c,d]$ intervallumra formalizálhatók az alapműveletek képletei a következőképpen ($0 \notin [c,d]$ osztás esetén):

$$\begin{aligned}
[a,b] + [c,d] &= [a+c, b+d], \\
[a,b] - [c,d] &= [a-d, b-c], \\
[a,b] \cdot [c,d] &= [\min\{ac,ad,bc,bd\}, \max\{ac,ad,bc,bd\}], \\
[a,b]/[c,d] &= [a,b] \cdot \left[\frac{1}{d}, \frac{1}{c}\right].
\end{aligned}$$

A halmazelméleti definíció ekvivalens az aritmetikai definícióval.

8.1.1. Algebrai tulajdonságok

Létezik zérus- és egységelem, a $[0,0]$ a zérus-, $[1,1]$ az egységelem. Viszont a kivonás és az osztás nem az összeadás és a szorzás inverze: például $[0,1] - [0,1] = [-1,1]$.

A szorzás nem disztributív az összeadásra nézve, viszont van szubdisztributivitási szabály:

$$A(B+C) \subseteq AB + AC$$

8.1. Példa. Az $X = [0,1]$ intervallumra

$$X(X-1) = [0,1]([0,1]-1) = [0,1][-1,0] = [-1,0],$$
$$X^2 - X = [0,1]^2 - [0,1] = [0,1] - [0,1] = [-1,1].$$

Az összeadás és a szorzás asszociativitása és kommutativitása az intervallumos műveletekre is igaz. Minden elemi valós függvényre megadható annak intervallumos megfelelője. Ahogy korábban is, itt is megköveteljük egy ϕ elemi függvény intervallumos kiterjesztésére, hogy

$$\Phi(x) \supseteq \{\phi(x) \mid x \in X\}.$$

A legtöbb elemi művelet monoton, így az intervallumos megfelelője könnyen számítható. Például, ha $X = [\underline{x}, \overline{x}]$, akkor

$$\exp(X) = [e^{\underline{x}}, e^{\overline{x}}]$$
$$\sqrt{X} = [\sqrt{\underline{x}}, \sqrt{\overline{x}}]$$

8.2. Definíció. Egy $f : \mathbb{R}^n \to \mathbb{R}$ valós függvényhez definiált intervallumos függvényt, $F : \mathbb{I}^n \to \mathbb{I}$ az f befoglalófüggvényének nevezzük, ha teljesíti, hogy minden $X \in \mathbb{I}$

$$x \in X \Rightarrow f(x) \in F(X), \text{ azaz } f(X) = \{f(x) \mid x \in X\} \subseteq F(X).$$

Ha f elemi függvények aritmetikai műveletekkel képzett függvénye, akkor ha minden elemi függvényt és aritmetikai műveletet azok intervallumos megfelelőjével helyettesítjük, egy befoglalófüggvényt kapunk. Az így kapott befoglalófüggvényt naív befoglalásnak, vagy természetes befoglalásnak nevezzük.

Tegyük fel, hogy ki tudjuk értékelni a függvény gradiensének egy befoglalófüggvényét, bármely intervallumon. Használjuk erre a $F'(X)$, illetve $\nabla F(X)$ jelölést. Ekkor az

$$F_C(X) = f(\hat{x}) + \nabla F(X)(X - \hat{x}), \qquad \hat{x} \in X$$

centrális vagy másképp középponti alaknak nevezett befoglalófüggvény is számítható. Nyilválvalóan $F_C(X) \supseteq f(X)$, hiszen a Lagrange középértéktétel szerint

$$f(x) = f(\hat{x}) + \nabla f(\xi)(x - \hat{x}) \quad \xi \in [\hat{x}, x]$$
$$f(x) \in f(\hat{x}) + \nabla F(X)(X - \hat{x})$$
$$f(X) \subseteq f(\hat{x}) + \nabla F(X)(X - \hat{x})$$

Baumann–pont: Adott X-re, leyen b az a pont, amelyik a legjobb alsó korlátot adja.

$$F_b(x) = f(b) + F'(x)(x - b)$$
$$f(b) + L(\underline{x} - b) = f(b) + U(\overline{x} - b)$$

$$b_- = \frac{U\underline{x} - L\overline{x}}{U - L}$$
$$c - b_+ = b_- - c$$
$$b_+ = \frac{U\overline{x} - L\underline{x}}{U - L}$$

8.3. Példa.

$$f(x,y) = xy + e^- \sqrt{y}$$
$$X = [-1,2]; \qquad Y = [1,3]$$

Naív tartalmazás:

$$F(X,Y) = [-1,2][1,3] + [e^{-1}, e^2] - [0, \sqrt{3}] = [-3 + \frac{1}{e} - \sqrt{2}, 5 + e^2]$$

$$\nabla f(x,y) = \begin{pmatrix} y + e^x \\ x - \frac{1}{2\sqrt{y}} \end{pmatrix}$$

$$\nabla F(X,Y) = \begin{pmatrix} [1,3] + [e^{-1}, e^2] \\ [-1,2] - [\frac{1}{2\sqrt{3}}, \frac{1}{2}] \end{pmatrix} = \begin{pmatrix} [1 + e^{-1}, 3 + e^2] \\ [-1.5, 2 - \frac{1}{2\sqrt{3}}] \end{pmatrix}$$

$$(\hat{x}, \hat{y}) == 0,1) \qquad f(\hat{x}, \hat{y}) = 0$$

$$F(X,Y) = 0 + \begin{pmatrix} [1 + \frac{1}{e}, 3 + e^2] \\ [-1.5, 2 - \frac{1}{2\sqrt{3}}] \end{pmatrix} ([-1,2], [0,2]) = \cdots = [-6 + e^2, 10 - \frac{1}{\sqrt{3}} + 2e^2]$$

8.4. Példa.

$$\sin(\pi x^3 y)(y - 1) \quad X = [0,1/2]; Y = [0,1]$$

$$\begin{aligned} F(X) &= \sin(\pi [0,0.5]^3 [0,1]) \cdot ([0,1] - [1,1]) = \\ &= \sin(\pi [0, \frac{1}{8}][0,1])[-1,0] = \\ &= \sin([0,\pi])[-1,0] = [0,0.39][-1,0] = [-0.39,0] \end{aligned}$$

8.2. Automatikus differenciálás

Az $f'(x)$ vagy $\nabla f(x)$ kiszámítására használjuk adott x esetén. Alternatívaként említhetjük meg a következőket:

- numerikus deriválás: gyors eljárás, de csak közelítést ad.
- szimbolikus deriválás: pontos értéket ad, de lassú, bonyolult eljárás.

Az automatikus differenciálás használja az úgynevezett láncszabályt:

$$f(x) = g(h(x)) \Rightarrow f'(x) = \frac{\mathrm{d}g}{\mathrm{d}h}\frac{\mathrm{d}h}{\mathrm{d}x}$$

A láncszabály alkalmazása alapján két típust különböztetünk meg.

Előrehaladó mód: Először $\frac{\mathrm{d}h}{\mathrm{d}x}$-et, majd később $\frac{\mathrm{d}g}{\mathrm{d}h}$-t számítjuk ki.

Visszafejtő mód: Pont fordítva, azaz először $\frac{\mathrm{d}g}{\mathrm{d}h}$-et, majd később $\frac{\mathrm{d}h}{\mathrm{d}x}$-t számítjuk ki.

8.5. Példa.

$$f(x_1, x_2) = x_1 x_2 + \sin x_1$$

Haladó mód:

$$\begin{aligned} w_4' &= \cos(w_1) \cdot w_1' \\ w_3' &= w_1' w_2 + w_2' w_1 \\ w_5' &= w_4' + w_3' w \end{aligned}$$

Legyen $(x_1, x_2) = (0,2)$, ekkor $w_1 = 0, w_1' = 1, w_2 = 2, w_2' = 0$.

$$w_1' = 1; \qquad w_3' = 2 + 0 = 2; \qquad w_5' = 3$$

Fordított mód:

$$\begin{aligned} v_5' &= \frac{\partial f}{\partial v_5} = 1 \\ v_3' &= v_5' \cdot \frac{\partial v_5}{\partial v_3} = v_5' \cdot 1 = 1 \\ v_4' &= v_5' \cdot \frac{\partial v_5}{\partial v_4} = 1 \\ v_2' &= v_3' \frac{\partial v_3}{\partial v_1} = v_3' \cdot v_1 = v_1 \quad x_2' = x_1 \end{aligned}$$

$$\left.\begin{aligned} v_1^{a)} &= v_4' \cdot \tfrac{\partial v_4}{\partial v_1} = v_1 \\ v_1^{b)} &= v_3' \cdot v_2 = v_2 \end{aligned}\right\} \Rightarrow \begin{aligned} v_1' &= v_1^{a)} + v_1^{b)} = \cos v_1 + v_2 \\ x_1' &= \cos(x_1) + x_2 \end{aligned}$$

1. kör: Számítsuk ki a formulákat a v_i'-kre és számítsuk ki az értékeket a csúcsoknál.
2. kör: Számítsuk ki a deriváltakat

Adott (u, u') párra definiálhatjuk a következő műveleteket:

$$\begin{aligned} (u, u') \pm (v, v') &= (u \pm v, u' \pm v') \\ (u, u')(v, v') &= (uv, u'v + uv') \\ \sin(u, u') &= (\sin(u), \cos(u)u') \\ (u, u')^n &= (u^n, nu^{n-1}u') \end{aligned}$$

8.6. Példa.

$$\begin{aligned}
f(x) = \sin(\pi x^3), \quad X = [0{,}0.5], \quad f'(X) = ? \\
(u, u') &= ([0{,}0.5], 1), \\
(u, u')^3 &= ([0{,}0.5], 1)^3 = ([0{,}0.125], 3[0{,}0.5]^2) = ([0{,}0.125], [0{,}0.75]), \\
(\pi, \pi') &= ([\pi, \pi], 0), \\
(\pi, \pi')(u, u')^3 &= ([\pi, \pi], 0)([0{,}0.125], [0{,}0.75]) = ([0{,}0.125\pi], [0{,}0.75\pi]), \\
\sin([0{,}0.125\pi], [0{,}0.75\pi]) &= ([0, \sin(0.125\pi)] \cos([0{,}0.125\pi])[0{,}0.75\pi]) = \\
&= ([0{,}0.35], [0{,}2.36]) \supseteq (f(X), f'(X))
\end{aligned}$$

8.3. Intervallumos Newton módszer

A hagyományos Newton-módszerrel alapvetően egy függvény zérushelyét szoktuk keresni. Az iterációs lépése a következőképpen vezethető le.

$$f(x) = 0; \qquad 0 = f(x) \approx f(x_0) + f'(x_0)(x - {}_0)$$

$$x \approx x_0 - \frac{f(x_0)}{f'(x_0)}$$

Vagyis a hagyományos Newton-módszer adott x_0 kezdőpontból az alábbi iterációs képlettel számítható

$$x_{k+1} := x_k - \frac{f(x_k)}{f'(x_k)}.$$

8.7. Megjegyzés. Amikor minimumot keresünk, akkor $f'(x) = 0$ kell, vagyis

$$x_{k+1} := x_k - \frac{f'(x_k)}{f''(x_k)}$$

Intervallumokra:

$$X_{k+1} = \left(m(X_k) - \frac{f'(m(X_k))}{F''(X_k)} \right) \cap X_k$$

$$\text{ahol } m(X) = \frac{\underline{x} + \overline{x}}{2}$$

$$f'(x) \in f'(x_0) + F''(x_0)(x - x_0)$$

$F''(x_k)$ lehet $(-\infty, \infty)$ is, ilyenkor X_{k+1} a metszetképzés nélkül nem véges, ilyenkor a módszer nem használható.

8.8. Állítás. *Az intervallumos Newton-módszerre teljesülnek a következők.*

i) Ha $x \in X$, amire $f'(x) = 0$, akkor $x \in X_{k+1}$.

ii) Ha $X_{k+1} \subset X_k$, akkor X_k-ban pontosan egy stacionárius pont van.

iii) Ha $X_{k+1} = \emptyset$, akkor $\nexists x \in X_k$, hogy $f'(x) = 0$.

8.9. Példa.

$$f(x) = x^2 - x$$
$$X_0 = [0,1]$$
$$f'(x) = 2x - 1 \qquad f''(x) = 2$$
$$X_1 = (0.5 - \frac{0}{[2,2]}) \cap [0,1] = [0.5,0.5]$$

A minimum $[0,1]$-en: 0.5

$$X_0 = [1,2]$$
$$X_1 = (1.5 - \frac{2}{[2,2]}) \cap [1,2] = [0.5,0.5] \cap [1,2] = \emptyset$$

Nincs minimum $[1,2]$-ben.

8.10. Példa.

$$f(x) = x^4 - 2x^3 + x^2$$
$$f'(x) = 4x^3 - 6x^2 + 2x$$
$$f''(x) = 12x^2 - 12x + 2 = 12x(x-1) + 2$$

$$X = [0,2]$$
$$F''(X) = 12[0,2][-1,1] + 2 = 12[-2,2] + 2 = [-22,26]$$
$$X_1 = (1 - \frac{0}{[-22,26]}) \cap [0,2] = (-\infty,\infty) \cap [0,2] = [0,2]$$

$$X = [0.9,1.1]$$
$$F''(X) = [-1.48,6.02]$$
$$X_0 = [0.99,1.01] \qquad F''(X) = [1.64,2.37]$$
$$X_1 = [0.9989,1.0021]$$

9. fejezet

DIRECT (DIviding RECTangles)

A heurisztikákra gyakran gondolunk úgy, mint olyan algoritmusokra, melyek adnak egy közelítő megoldást, mindenféle garancia nélkül, hogy az közel van az optimumhoz. A globális optimalizálásban heurisztikák gyakran párosulnak véletlen keresési technikákkal. Egy könnyen érthető példa a rács menti keresés. Hatékonysága könnyen elemezhető, a kiértékelt pontok száma exponenciálisan nő a dimenzióval.
Ami általános a heurisztikákban, nyerni próbálunk a lokális és globális keresésen egy megengedett területen.

A DIviding RECTangles (felosztó téglalapok) algoritmus előre meghatározott N számú mintapontot generál egy tégla által határolt megengedett területen elhelyezkedő rács felett a skálázott

$$x_1 = \frac{1}{2}(1,1,\ldots,1)^T$$

középpontból indulva. Ezt követően az x_k pont finomítása az x_k egy környezetében való újabb mintavételezést jelent.

Hogy eldöntsük mik az érdekes területek, minden mindtaponthoz egy (akár változó) u_k sugárvektort tárolunk, hogy leírja az $(x_k - u_k, x_k + u_k)$ téglát. Az u_k hossza és az $f(x_k)$ függvényérték határozza meg, hogy x_k jelölt-e finomításra, vagy sem. Egy α paraméter szabályozza a lokális kontra globális nyereséget. Az algoritmust három döntés jellemzi

- Hogyan válasszuk meg a finomításra szánt pontokat
- Hogyan mintavételezzünk a kiválasztott pont körül
- Hogyan frissítsük az u_k információt.

9.1. Kiválasztás és finomítás

A kiválasztás módja megad minden iterációban egy M számú listát a már kiszámolt u_i vektorok méreteivel. Ezt tárolhatjuk rendezve. Minden pontban a hozzárendelt u_k vektor valamely s_j által meghatározott S_j méretosztályba esik. Az alapötlet az, hogy új

pontok generálásával az aktuális mintapontok kisebb méretekhez kerülnek. Az előforduló méretek nem ekvidisztánsak, de u_k frissítésének módja miatt egy meghatározott mintázatot követnek. Minket a viszonylag alacsony pontok ($f(x_k)$ kicsi) érdekelnek és azok is a viszonylag feltáratlan részeken ($\|u_k\|$ nagy). Egy Pareto-féle módon minden nemdominált pontot kiválasztunk finomításra. Első körben ez minden olyan pont kiválasztását jelenti, ahol $m_j = \min_{k \in S_j} f(x_k)$. Itt az α paraméter arra szolgál, hogy ne legyen túl lokális a mintavételezés. Az $f^U - \alpha \left|f^U\right|$ értéket jegyezzük meg, $f^U = \min_k f(x_k)$. Ezt a pontot az u.n. nemdominált pontokhoz vesszük. Egy egyenest képzeljünk ebből a pontból felfeé úgy, hogy a kapott görbe konvex maradjon. Lássuk hogyan kivitelezhető ez. A legnagyobb méretosztályból S_1 indulunk, kiválasztjuk az m_1 minimumnak megfelelő pontot és végigmegyünk a j osztályon egészen

$$m_j \geq f^U - \alpha \left|f^U\right| + \frac{s_j}{s_{j-1}}(m_{j-1} - f^U + \alpha \left|f^U\right|)$$

esetig hogy m_j az egyenes fölött legyen az $f^U - \alpha \left|f^U\right|$ és m_{j-1}. Ez azt jelenti hogy az utolsó legalacsonyabb pont nem feltétlenül lesz finomítva, mert a körülötte lévő tér nem elég üres. Ahogy már szerepelt, ez az α csak az paraméterrel szabályozható.

9.1.1. algoritmus select($f_1, \ldots, f_k, \|u_1\|, \ldots, \|u_k\|, \alpha$

Határozzuk meg $f^U = \min_l f_l$-t.
Rendezzük az $\|u_1\|, \ldots, \|u_k\|$ listát és adjuk meg az $S_1, \ldots, S_M$ osztályokat $s_1 > > s_2 > \cdots > s_M$
$m_1 = \min_{k \in_1} f_k, \quad j = 1$
repeat
    kiválasztjuk $\arg\min_{k \in S_j} f_k$-t
    $j = j + 1, \quad m_j = \min_{k \in S_j} f_k$
until ($m_j \geq f^U - \alpha \left|f^U\right| + \frac{s_j}{s_j - 1}(m_{j-1} - f^U + \alpha \left|f^U\right|)$)

9.2. Finomítás és négyzetek frissítése

Egy pont finomításán az $(x - u, x + u)$ hipertéglából való további mintavételezést értünk. Továbbá a régi x mintapont a további új mintapontokkal együtt kisebb sugárvektort kap, mint u.

9.2.1. algoritmus refine(x, u,**global** k,N)

Határozzuk meg $I = \arg\max_i u_i$
for ($i \in I$) **do**
    Értékeljük ki az $f(x - \frac{2}{3}u_i e_i)$ és $f(x + \frac{2}{3}u_i e_i)$
    $w_i = \min\{f(x - \frac{2}{3}u_i e_i), f(x + \frac{2}{3}u_i e_i)\}$
    $k = k + 2$
    if ($k = k + 2$)
        STOP
for ($i \in I$) **do**
    $v_i = u$
repeat
    kiválasztjuk $\eta = \arg\max_{i \in I} w_i$-t
    for ($i \in I$) **do**
        $v_{i\eta} = \frac{1}{3}u_\eta$
        $u_\eta = \frac{1}{3}u_\eta$
    Tároljuk $x_{k-1} = x - \frac{2}{3}u_\eta e_\eta, u_{k-1} = v_\eta$
    Tároljuk $x_k = x + \frac{2}{3}u_\eta e_\eta, u_k = v_\eta$
until $(I = \emptyset)$

9.2.2. algoritmus DIRECT(f, α, N)

$k = 1, x_1 = \frac{1}{2}(1,1,\ldots,1)^T, u_1 = \frac{1}{2}(1,1,\ldots,1)^T, f_1 = f(x_1)$
repeat
    $J = \text{select}(f_1, \ldots, f_k, \|u_1\|, \ldots, \|u_k\|, \alpha)$
    for ($j \in J$) **do**
        refine(x_j, u_j, k, N)
until (STOP)

10. fejezet

Deriválást nem igénylő eljárások

Tegyük fel, hogy egy $f : \mathbb{R}^n \to \mathbb{R}$ függvény optimumát keressük. Előfordulhat, hogy a deriváltra vonatkozóan nem áll rendelkezésre információ, ez a rész néhány ebben az esetben használható eljárást ismertet.

10.1. Nelder–Mead-algoritmus

A most bemutatott eljárás robusztus, barátságos geometriai leírással bír, így meglehetősen népszerű.

A $P = \{p_0, \ldots, p_n\}$ iteratív halmazt szimplexnek nevezzük, mert $n+1$ pontot tartalmaz egy n-dimenziós térben. Legyen x_0 egy tetszőleges kezdőpont, ekkor az induló P ponthalmaz:

$$P := \{x_0, x_0 + \delta e_1, \ldots, x_0 + \delta e_n\}$$

ahol δ egy skálázási faktor, e_i pedig az i. egységvektor. Az alábbi összetevők lesznek fontosak az algoritmus futása során:

- A P ponthalmaz két „legrosszabb" pontja:

  $$p^{(n)} = \arg\max_{p \in P} f(p); \qquad p^{(n-1)} = \arg\max_{p \in P \setminus p^{(n)}} f(p)$$

  valamint a „legjobb" pontja:

  $$p^{(0)} = \arg\min_{p \in P} f(p)$$

- A „legmagasabb" pontot kivéve az összes pontra vett:

  $$c = \frac{1}{n} \sum_{i \neq (n)} p_i$$

  centroid.

- Egy u.n. *reflexiós* lépés során generált próbapont:

  $$x^{(r)} = c + (c - p^{(n)})$$

- Sikeres reflexiós lépést követően egy u.n. *nyújtási* lépés során generált próbapont:
$$x^{(e)} = c + (1+\lambda_1)(c - p^{(n)}), \;\; \lambda_1 \in (0,1)$$
- Sikertelen reflexiós lépést követően egy u.n. *zsugorítási* lépés során generált próbapont:
$$x^{(e)} = c + \lambda_2(c - p^{(n)}), \;\; \lambda_2 \in (0,1)$$
- Ha a fenti próbák nem hoznak jó eredményt, a szimplexet egy u.n. *többszörös összehúzás* során a minimális $p^{(0)}$ pont felé húzzuk össze:
$$p_i := \frac{1}{2}(p_i + p^{(0)}), i \in \{0,\dots,n\}.$$

10.1. Példa. Tekintsük az $f(x) = 2x_1^2 + x_2^2 - 2x_1x_2 + |x_1 - 3| + |x_2 - 2|$ függvényt. Legyen a kiinduló szimplex a $p_0 = (1,2)^T, p_1 = (1,0)^T, p_2 = (2,1)^T$ pontokkal adva. Az algoritmus először egy reflexiós lépést tesz, az új pont $p^{(1)}$. Azonban a következő iterációban a reflexiós pont se az $f_{(0)} < f(x^{(r)} < f_{(n-1)}$ sem az $f(x^{(r)}) < f_{(0)}$ feltételeket nem elégíti ki, így a zsugorított pontot számítjuk ezután. Mivel ennél a függvényérték jobb mint $f(x^{(r)})$, így $p^{(n)}$-t kicseréljük erre a pontra. Azt is láthatjuk, hogy $f(x^{(c)}) < f_{(n-1)}$. Megfigyelhető, hogy ha az optimum a politópon belül van, akkor annak mérete csökken, ezzel közelítve a kilépési feltétel teljesüléséhez.

10.2. Powell módszer

Ebben az eljárásban irányok egy $(d_1,\dots,d_n)$ halmazát bővítjük iteratívan, annak érdekében hogy az x^* felé mutató irányt közelítsük. Egy x_0 kezdőpontból indulunk, amit ezúttal jelöljünk $x_1^{(1)}$-gyel. Minden iterációban n lépést teszünk az n irányt felhasználva. Minden lépésben $x_{i+1}^{(k)} = x_i^{(k)} + \lambda d_i$, ahol k az iteráció sorszáma. A λ lépésközről feltesszük hogy optimális, azaz
$$\lambda = \arg\min_\mu f(x_i^{(k)} + \mu d_i).$$
Az irányok halmazát a koordinátairányokkal inicializáljuk, azaz $d_i = e_i$ kezdetben. Az irányokat az alábbi módon frissítjük: $d = x_{n+1}^{(k)} - x_1^{(k)}$ a végső irány a k. iterációban. Legyen a következő kezdőpontunk ebben az irányban:
$$x_1^{(k+1)} = x_{n+1}^{(k)} + \lambda d$$
ahol λ az optimális lépésköz. A régebbi irányokat eltoljuk: $d_i = d_{i+1}, i \in 1,\dots,n-1$. Az utolsó irány a legújabb lesz: $d_n = d$. Az iterációt folytatjuk, egészen addig amíg $\left| f(x_1^{(k+1)} - f(x_1^{(k)})) \right| < \varepsilon$

10.1.1. algoritmus Nelder–Mead

$k = 0$; $P = \{p_0, \dots, p_n\}$ ahol $p_0 = x_0$, $p_i = x_0 + \delta e_i$, $i \in 1, \dots, n$
Értékeljük ki f-et a $p_1, \dots, p_n$ pontokban.
Határozzuk meg a $p^{(n)}$, $p^{(n-1)}$ és $p^{(0)}$ pontokat P-ben. (rendezés)
while $(f_{(n)} - f_{(0)} > \varepsilon)$ **do**
  $c = \frac{1}{n} \sum_{i \neq (n)} p_i$
  $x^{(r)} := c + (c - p^{(n)})$, kiértékeljük $f(x^{(r)})$-t.
  if $(f_{(0)} < f(x^(r)) < f_{(n-1)})$
    $P = P \setminus \{p^{(n)}\} \cup \{x^{(r)}\}$
  if $(f(x^(r)) < f_{(0)})$
    $x^{(e)} = c + 1.5(c - p^{(n)})$, kiértékeljük $f(x^{(e)})$
    $P = P \setminus \{p^{(n)}\} \cup \{\arg\min\{f(x^{(e)}), f(x^{(r)})\}\}$
  if $(f(x^{(r)}) \geq f_{(n-1)})$
    $x^{(c)} = c + 0.5(c - p^{(n)})$, kiértékeljük $f(x^{(c)})$
    if $(f(x^{(c)}) < f(x^{(r)})$ && $f(x^{(c)}) < f_{p^(n)})$
      $P = P \setminus \{p^{(n)}\} \cup \{x^{(c)}\}$
    else
      if $(f(x^{(c)}) > f(x^{(r)})$ && $f(x^{(r)}) < f(p^{(n)}))$
        $P = P \setminus \{p^{(n)}\} \cup \{x^{(r)}\}$
      else
        if $(f(x^{(c)}) < f(p^{(n)}))$
          $P = P \setminus \{p^{(n)}\} \cup \{x^{(c)}\}$
        else
          $p_i = \frac{1}{2}(p_i + p^{(0)})$, $i = 0, \dots, n$
          Kiértékeljük $f(p_i)$ $i = 1, \dots, n$
          $P = \{p_0, \dots, p_n\}$
  $k = k + 1$

10.2.1. algoritmus Powell

$k = 0$, $(d_0, \dots, d_n) = (e_0, \dots, e_n)$ és $x_1^{(1)}$
repeat
  $k = k + 1$
  for $(i \in \{1, \dots, n\})$ **do**
    $\lambda = \arg\min_\mu f(x_i^{(k)} + \mu d_i)$
    $x_{i+1}^{(k)} = x_i^{(k)} + \lambda d_i$
  $d = x_{n+1}^{(k)} - x_1^{(k)}$
  $x_1^{(k+1)} = x_{n+1}^{(k)} + \lambda d$, ahol $\lambda = \arg\min_\mu f(x_{n+1}^{(k)} + \mu d)$
  $d_i = d_{i+1}$, $i = 1, \dots, n$, $d_n = d$
until $(\left| f(x_1^{(k+1)}) - f(x_1^{(k)}) \right| < \varepsilon)$

11. fejezet

Eljárások korlátozott feladatok optimalizálásra

Egy általános nemlineáris programozási feladatot felírhatunk az

$$\begin{aligned} &\min f(x) && \\ &g_i(x) \leq 0 && i = 1,\ldots,p \text{egyenlőtlenség kényszerek} \\ &g_i(x) = 0 && i = p+1,\ldots,\text{egyenlőségi kényszerek} \end{aligned}$$

Amikor egy ilyen korlátozott feladatot akarunk megoldani, akkor két lehetőségünk van. Vagy beépítjük a kényszereket a célfüggvénybe, ezáltal nemkorlátozott feladattá átalakítva az eredetit (amely bár nem ekvivalens azzal, de jó paraméterekkel a megoldás tart az eredetihez), vagy közvetlenül korlátozzuk a keresést a megengedett területre.

Ebben a részben a korlátok célfüggvénybe építésére látunk példát, majd a Gradiens vetítés módszerével ismerkedünk meg, amely példa a második útra.

11.1. Bűntetőfüggvény módszer

A módszer alapötlete a nem megengedett területek bűntetése egy

$$p_\mu(x) = \mu\left(\sum_{i=1}^{p}\max\{g_i(x),0\} + \sum_{i=p+1}^{m}|g_i(x)|\right)$$

vagy

$$p_\mu(x) = \mu\left(\sum_{i=1}^{p}\max\{g_i(x),0\}^2 + \sum_{i=p+1}^{m}\left|g_i^2(x)\right|\right)$$

u.n. bűntető függvény segítségével. Ezek a függvények 0-k a megengedett részeken, azonban pozitívak a tiltott részeken. A bűntetőfüggvény céllfüggvényhez adásával minden μ-re egy nemkorlátos feladatot kapunk, amelynek minimalizáló pontja egy közelítése lesz az eredeti megoldásnak, ha μ kellően nagy. A módszerrel több probléma is van

- Nem tudjuk előre, mekkora μ fog kelleni, azonban
- ha μ túl nagy, a feladat rosszul kondíciónált lesz.

A probléma kiküszöbölésére μ értékét fokozatosan növeljük, és a következő minimumkeresést az előző eredményéből indítjuk. Ezáltal a konvergencia gyorsabb lesz, és a rosszul kondíciónáltságot is elkerüljük. Kilépünk, ha az aktuális közelítésre $p_\mu(x^*(\mu)) \leq \varepsilon$, ekkor $x^*(\mu)$-t elfogadjuk egy közelítő megoldásra.

11.1.1. algoritmus BűntetőEljárás($f, g, p, \mu_0, \beta, \varepsilon$)

$k = 1$
$x_k = \arg\min P_\mu(x)$
while $(p_\mu(x_k) > \varepsilon)$ **do**
    $k = k + 1$
    $\mu_k = \beta \cdot \mu_{k-1}$
    $x_k = \arg\min_{x=X} P_\mu(x)$

11.2. Korlátozófüggvény-módszer

Minden igyekezet ellenére a bűntetőfüggvény-módszer olykor nem megengedett megoldást ad. Így ha az alkalmazás szigorúan megköveteli a megengedettséget, akkor más módszerre van szükségünk. Az ötlet: Adjunk meg egy olyan függvényt, amely egy „gátat" szab a korlátoknál, így x_k csak a megengedett tartományban lehet. (Emiatt ez a módszer csak egyenlőtlenségi korlátok esetén használható).

Például

$$b_\mu(x) = -\mu \sum_{i=1}^{p} \frac{1}{g_i(x)}$$

és

$$b_\mu(x) = -\mu \sum_{i=1}^{p} \ln(-g_i(x))$$

pozitív értéket ad a szigorúan megengedett pontokra, végtelent ha valamely korlát éles. A korlátozófüggvényt a nem megengedett pontokban nem szükséges definiálnunk. A $B_\mu(x) = f(x) + b_\mu(x)$ új célfüggvényt minimalizálva kapjuk a közelítést. Az algoritmus alapvetően ugyanaz mint a bűntető-függvényeknél, azonban μ-t csökkentjük a növelés helyett (a határon továbbra is nagy lesz). Joggal vehetjük észre, hogy ezzel a módszerrel továbbra is figyelembe kell vennünk bizonyos kényszereket, de az új feladatra ezek egyike sem éles, így bármilyen nemkorlátos eljárást használhatunk, bizonyos óvintézkedések megtétele után.

11.2.1. algoritmus KorlátozóEljárás($f, g, b, \mu_0, \beta, \varepsilon$)

$k = 1$
$x_k = \arg\min B_\mu(x)$
while $(b_\mu(x_k) > \varepsilon)$ **do**
    $k = k + 1$
    $\mu_k = \frac{\mu_{k-1}}{\beta}$
    $x_k = \arg\min_{x \in X} P_\mu(x)$

11.3. Gradiens vetítés módszer

Ez az eljárás a legmeredekebb ereszkedés módszerének korlátozott problémák megoldására szolgáló módosítása. Minden lépésben, az új irányt úgy módosítjuk, hogy még a megengedett régióban maradjunk, oly módon, hogy a gradienst az aktív korlátokra vetítjük.

A vetítést egy P projekcióval végezzük, $r = -P\nabla f$ módon. Ha M az aktív korlátok Jacobi-mátrixa (azaz oszlopai $\nabla g_i(x)$, azokra a g_i-kre, melyekre $g_i(x) = 0$), akkor

$$P = I - M(M^T M)^{-1} M^T$$

ugyanis tudjuk, hogy minden aktív korlátra r merőleges a korlát gradiensére, azaz $\nabla g_i^T r = 0$. Így

$$M^T r = 0.$$

A legmeredekebb ereszkedés irányát a

$$\begin{aligned} \min & r^T \nabla f \\ & M^T r = 0 \\ & \|r\|_2 = 1 \end{aligned}$$

probléma megoldásával kapjuk meg. A Lagrange–függvényt használva

$$L(r, u, v) = r^T \nabla f + r^T M u + v r^T r$$

ahol $u \in \mathbb{R}^n, v \in \mathbb{R}$, $\|r\|_2^2 = r^T r$. Az optimalitás szükséges feltétele

$$\frac{\partial L}{\partial r} = \nabla f + Mu + 2vr = 0$$

Beszorozva M^T-al és figyelembe véve hogy $M^T r = 0$

$$M^T \nabla f + M^T M u + 2v M^T r = M^T \nabla f + M^T M u = 0$$

amiből

$$u = -(M^T M)^{-1} M^T \nabla f$$

Ezt visszaírva az eredeti egyenletbe megkapjuk az

$$r = -\frac{1}{2v}(E - M(M^TM)^{-1}M^T)\nabla f$$

Az $\frac{1}{2v}$ szorzótól eltekinthetünk, r egy iránynak felel meg.

Ha $r = 0$ és $u \geq 0$, akkor a Karush-Kuhn-Tucker-feltételek állnak, így KKT pontot találtunk. Ha valamelyik Lagrange-multiplikátor negatív, akkor továbbra is találhatunk csökkenő irányt bizonyos negatív u_i-vel bíró korlátok elhagyásával. A negatív u_i jelentése ugyanis, hogy a megfelelő korlát nem éles az ereszkedési irányra. Általában a legkisebb u_i-vel rendelkező korlátot hagyjuk el. Ha $r \neq 0$, akkor megtaláltuk az ereszkedési irányt. Egyébként még több korlátot hagyhatunk el. Ha már nincs több korlát, de $r = 0$, akkor megállhatunk, elértünk egy KKT pontot.

Miután az r megengedett irányt megtaláltuk, meghatározzuk az optimális lépésközt

$$\lambda = \arg\min_{\mu>0} f(x_k + \mu r) \leq 0$$

úgy, hogy a következő iteráció kielégítse a nem éles feltételeket, azaz $g_i(x_k + \lambda r) \leq 0$. Valójában az első olyan korlát, amely éles lesz az r irány mentén, határozza meg a maximális lépésközt $\lambda_{\max}$-ot. Speciálisan egy $a_i^T x - b_i \leq 0$ lineáris korlát esetén λ-nak ki kell elégítenie az $a^T(x_k + \lambda r) - b_i \leq 0$ egyenlőtlenéget, úgy hogy $\lambda_{\max} \leq \dfrac{b_i - a_i^T x_k}{a_i^T r}$

11.3.1. algoritmus GradProj(f, g, x_0, ε)

```
k = 0
repeat
    r = −(I − M(M^T M)^{−1} M^T)∇f
    while (r = 0) do
        u = −(M^T M)^{−1} M^T ∇f
        if (min_i u_i < 0)
            g_i-t eltávolítjuk az aktív korlátok közül.
        else
            return x_k (egy KKT pont)
    λ = arg min_μ f(x_k + μr)
    if (∃i g_i(x_k + λr) < 0)
        Határozzuk meg λ_max-ot.
        λ = λ_max
    x_{k+1} = x_k + λr
    k = k + 1
until (|x_k − x_{k−1}| < ε)
```

A λ kiszámítására nemlineáris korlátok esetén azok helyett számolhatunk azok lineáris közelítésével. nemlineáris korlátoknál arra is szükség lehet, hogy egy visszaállítási lépésben gondoskodjunk róla, hogy az új pont nem sérti meg az éles korlátokat.

A legmeredekebb ereszkedés vetítése általánosítható más ereszkedési irányt használó eljárásokra is. Annyit kell tennünk, hogy ∇f helyett a használni kívánt iránnyal implementáljuk az algoritmust.

11.4. Pontatlan vonalmenti keresés

Majdnem minden ereszkedési irányt használó eljárásban szükséges minden lépésben egy vonalmenti keresés. Eddig csak optimális lépésközt használtunk, így pontos keresést tételeztünk fel. Egyéb esetben egydimenziós optimalizáló eljárást használhatunk. Ha a minimumtól még távol vagyunk, az optimális lépésköz „túl jó közelítése" általában nem hatékony. A kérdés, honnan tudhatjuk, hogy mennyire messze vagyunk, és a közelítésünk már elegendő? Pontos válasz nincs a kérdésre, de néhány szabályt alkalmazhatunk. Például gyaníthatjuk, hogy $\|\nabla f(x)\| \to 0$, ahogy $x \to x^*$. Hogy elkerüljük a túl nagy, vagy túl kicsi lépésközt, a célfüggvény megfelelő csökkentésére van szükség. Egy kicsi $0 < \alpha < 1$-re

$$f_k + (1-\alpha)\lambda \nabla f_k^T r_k < f(x_k + \lambda r_k)$$
$$f(x_k + \lambda r_k) < f_k + \alpha\lambda \nabla f_k^T r_k$$

teljesülése szükséges. A $\varphi_{r_k}(\lambda) := f(x_k + \lambda r_k)$ jelölést használva, a két egyenlőtlenséget együtt a

$$\varphi_{r_k}(0) + (1-\alpha)\varphi'_{r_k}(0)\lambda < \varphi_{r_k}(\lambda) < \varphi_{r_k}(0) + \alpha\varphi'_{r_k}(0)\lambda$$

alakban írhatjuk. Ezt Goldstein feltételnek nevezzük. A második egyenlőtlenséget, ha magában szerepel Armijo-egyenlőtlenségnek hívjuk. A két egyenlőtlenség ad egy alsó $\underline{\lambda}$ és egy felső $\bar{\lambda}$ korlátot λ-ra. Ez jelenthet több nem összefüggő intervallumot és az optimális megoldás akár kívül is eshet ezeken. Ezt elkerülendő további feltételre van szükségünk: A gradiens az új pontban kisebb legyen, mint a régiben. Az elõzõ jelölésekkel tehát

$$\varphi_{r_k}(\lambda) \leq \sigma\varphi'_{r_k}(0)$$

vagy másképp

$$\nabla f(x_k + \lambda r_k)^T r_k < \sigma \nabla f(x_k)^T r_k$$

Ezt a kritériumot Wolfe-feltételnek nevezzük. Az Armijo- és Wolfe-feltételt együtt szokás Wolfe-feltételeknek nevezni.

A gyakorlatban általában egy visszakövető vonalkeresést végzünk, amíg a kiválasztott feltételek nem teljesülnek. Ennek koncepciója az alábbi: Adott (akár nagy) kezdő lépésköz λ_0, csökkentsük arányosan egy $0 < \beta < 1$ faktorral, amíg a kiválasztott feltételek nem teljesülnek

$k = 1$
while (feltételek nem teljesülnek) **do**
    $\lambda_k = \beta\lambda_{k-1}$
    $k = k + 1$

12. fejezet

Sztochasztikus módszerek

Sztochasztikus eljárások alatt az olyan algoritmusokat értjük, melyek (ál)véletlen számok segítségével generálnak új próbapontokat. Először két alapvető megközelítést vizsgálunk meg: Pure Random Search (tiszta véletlen keresés), valamint Multistart. Ezt a szimulált hűtés heurisztika egy klasszikus változata követi.

12.1. Pure Random Search (PRS)

Egyenletes eloszlásban generáljuk a pontokat a tartományon, és a legkisebb értéket adót eltároljuk mint a globális minimum pont egy közelítését. Kedvelt referenciaalgoritmus, mivel könnyű elemezni. Megmutatható, hogy néhány könnyed feltevés után az

$$y_{(1)} = \min\{f(x_1), \ldots, f(x_N)\}$$

valószínűségi változó eloszlásfüggvénye

$$F_{(1)}(y) = 1 - (1 - \mu(y))^N$$

ahol $\mu(y)$ az y-hoz tartozó nívóhalmaz mértéke, $\mu(y) = P(f(x) \leq y)$.

Egy korai megfigyelés szerint annak valószínűsége hogy az $N+1$. húzással jobb pontot találunk ha már N húzás volt, $\frac{1}{N+1}$, és K újabb pontra ez a valószínűség $\frac{K}{N+K}$-ra nő.

A sztochasztikus folyamatokra igaz, hogy ha $N \to \infty$, akkor a végén a globális optimumot megtaláljuk. Ez persze csak elméletileg nyugtathat meg minket, valójában célunk elérésének valószínűsége nagyban függ a sikerterület méretétől. Egy klasszikus megközelítés ennek a növelésére lokális keresés használata. Ezt az eljárást *multistart*nak nevezzük.

12.2. Multistart

Legyen $\mathrm{LS}(x) : X \to X$ olyan lokális optimalizáló rutin, amely egy megadott kezdőpontból visszaad egy pontot a tartományban, ami közelítése egy lokális minimumnak.

A multistart véletlen generált kezdőpontokból LS-sel generált pontok torlódási pontjait határozza meg.

Általában problémát jelent a megállás kérdése, azaz hogy mikor találtuk már meg az összes optimumot. A megállási szabályt tekintve egy 1987-es eredmény alapján egy nem túl sok feltevést igénylő eredmény született:

Ha N lokális keresést végeztünk, és megtaláltunk w optimum pontot, akkor az optimum pontok becsült $\hat{w}$ száma:

$$\hat{w} = \frac{w(N-1)}{N-w-2}$$

Az alapötlet az, hogy megállunk, amikor w már közeli $\hat{w}$-hez.

12.2.1. algoritmus Multistart$(X, f, \mathrm{LS}, N$

$f^U = \infty$
for $(k = 1, \ldots, N)$ **do**
  Generálunk x pontot X-ben egyenletesen
  $x_k = \mathrm{LS}(x)$
  if $(f(x_k) < f^U)$
    $f^U = f(x_k), \quad x^U = x_k$

12.3. Klaszterezés a lokális keresések számának csökkentésére

Az alapötlet az, hogy értelmetlen számítási teljesítményt fektetni olyan lokális optimalizálásba, amelyben a kiindulási pont valamely már megtalált lokális optimum vonzási tartományában van. Kis függvényértékeket adó pontok medencékbe koncentrálódnak, amik érintkezhetnek lokális minimalizáló eljárások vonzási tartományával. Sima függvények esetén az ilyen tartományok elliptikus karakterisztikával bírnak, amit a Hesse-mátrix ír le az optimum pontokban. Sok klaszterező algoritmus változatot terveztek és nagy előrelépések történtek a 70-es és 80-as években a klaszterek meghatározásához felhasznált információt illetően. Numerikus eredményeket analitikusak váltotak fel.

Most az u.n. MLSL (Multi-Level Single Linkage) algoritmust mutatjuk be. Ez nem közvetlenül formák klaszterekt, de az ötlet, hogy nem indítunk lokális keresést olyan pontból, melyet impliciten már egy másik megtalált optimumhoz rendeltünk. A megtalált optimumokat a Λ halmazban tároljuk. A tűrési távolságot az aktuális k iterációban a

$$r_k = \sqrt{\pi}\left(\sigma V(X)\Gamma(1+\frac{n}{2})\frac{\log k}{k}\right)^{\frac{1}{n}}$$

12.3.1. algoritmus Multi-level Single Linkage($X, f, N, \mathrm{LS}, \gamma, \sigma$)

Válasszunk egyenletesen N pontot X-en és értékeljünk ki őket. $\Lambda = \emptyset$
Válasszuk ki a $k = \gamma N$ legalacsonyabb pontot.
for ($i = 1, \ldots, k$) **do**
  if ($\nexists j < i, (f(_j) < f(x_i)$ÉS $\left\|x_j - x_i\right\| < r_k$)
    Lokális keresés $\mathrm{LS}(x_i)$
    if ($\mathrm{LS}(x_i) \notin \Lambda$)
      Tároljuk $\mathrm{LS}(x_i)$-t Λ-ban.
while ($|\Lambda| - \hat{w} < 0.5$) **do**
  $k = k + 1$, veszünk egy x_k mintapontot X-ből.
  if ($\nexists j < k, (f(x_j) < f(x_k)$ ÉS $\left\|x_j - x_k\right\| < r_k)$)
    Lokális keresés $\mathrm{LS}(x_k)$
    if ($\mathrm{LS}(x_k) \notin \Lambda$)
      Tároljuk $\mathrm{LS}(x_k)$-t.

12.4. Alagutazás és feltöltött függvények

Míg a klaszterezés ötletét a 70-es években kutatták, egy másik ötlet is megjelent a 80-as években: Nem érdekel minket az összes lokális optimum, csak le akarunk lépdelni a lokális optimumokon keresztül a globálisig. Számos gyakran hivatkozott cikk jelent meg a témában és további kutatásokat eredményezett. Az egyik ilyen ötlet:

Tegyük fel, hogy egy véges sok minimum ponttal rendelkező sima függvény optimumát keressük. Miután megtaláltunk egy lokális minimumot, a függvényt áttranszformáljuk, hogy egy új kezdőpontot találjunk egy jobb lokális minimumhoz tartozó vonzási tartományban. Ezt megtehetjük egy u.n. alagutazással, vagy feltöltéssel.

Az alagutazás: Miután megtaláltunk egy x_1^* lokális minimumot egy x_0 kezdőpontból, az algoritmus iteratívan keresi a

$$T_k(x) = \frac{f(x) - f(x_k^*)}{(x - x_k^*)^\alpha} = 0$$

egyenlet megoldását, $\alpha > 0$. Az $x_k \neq x_k^*$ megoldásban a függvényérték azonos $f(x_k) = f(x_k^*)$, így ezt kezdőpontnak használhatjuk egy lokális keresésben hogy elérjük x_{k+1}^*-ot amiben $f(x_{k+1}^*) < f(x_k^*)$, amire aztán ugyanezt végrehajtjuk. Az ötlet csábító, a kérdés csak az alagutazás végrehajtásának hatékonysága.

Egy másik eljárás az úgynevezett feltöltés. A cél ugyanaz mint az alagutazásnál. El kívánjuk érni egy eddig találtaknál jobb minimum vonzási tartományát. Az ötlet azonban itt az, hogy megszűntetjük a stacionárius pontokat a már megtalált optimumok vonzási tartományában. Ehhez az x_k^*-hoz tartozó vonzási tartományt „feltöltjük". Például az alábbi

$$ff(x) := -\frac{1}{r + f(x)} \exp\left(\frac{\left\|x - x_k^*\right\|^2}{\rho}\right)$$

feltöltött függvényt.

12.4.1. algoritmus Feltöltött függvény multistart

$k = 1, \quad x_1^* = \mathrm{LS}\, f(x_0)$
repeat
Adaptáljuk a ρ és r paramétereket.
Válasszuk meg ξ-t
$x_k = \mathrm{LS}\, ff(x_k^* + \xi)$
$x_{k+1} = \mathrm{LS}\, f(x_k)$
$k = k + 1$
until $(f(x_k^*) \geq f(x_{k-1}^*))$

12.5. P-algoritmus

Legyen $f^U = \min_i y_i$ a legjobb eddig talált függvényérték, és δ_k egyféle pozitív aspirációs szint a javulásra a k iterációban. A P-algoritmus a következő pontnak azt választja, amelynél az $f^U - \delta_k$ javulás valószínűsége maximális:

$$x_{k+1} = \arg\max_x P(\xi_k(x) < f^U - \delta_k)$$

A fenti megoldása nagyban függ a sztochasztikus modell konstrukciójától. Például ha ξ_k Gauss-eloszlású, akkor a modellt leírja az $m_k(x)$ és $s_k^2(x)$ várható érték és szórás, ekkor a fentivel ekvivalens:

$$x_{k+1} = \arg\max_x \frac{f^U - \delta_k - m_k(x)}{s_k(x)}$$

Vegyük észre hogy $s_k(p_i) = 0$.

12.6. Radiális alapfüggvény

Az alapötlet, hogy interpoláljuk f-et az x pontban az y_i és p_i értékek ismeretében úgy, hogy a közeli pontok jobban számítanak. Ezt az ú.n. radiális alapfüggvénnyel érjük el. Pl. $\Theta(r) = \exp(-r^2)$. Legyen most

$$\varphi_k(x) = \sum_i w_i \Theta(\|x - p_i\|)$$

ahol a w_k súlyok meghatározhatók a $\varphi_k(p_j) = y_j$ egyenletekből ($w = \Theta(p)^{-1} y^T$ egyenletrendszer)

12.7. Vezérelt véletlen keresés

12.7.1. algoritmus Vezérelt véletlen keresés

$k = N$
Generáljunk és értékeljünk ki egy P halmazt N egyenletesen választott X-beli ponttal
$y_k = f(\text{pmax}_k) = \max_{p \in P} f(p)$
while $(y_k - \min_{p \in P} f(p) > \alpha)$ **do**
    $k = k + 1$
    Válasszunk egy $\{p_1, \ldots, p_{n+1}\}$ véletlen részhalmazt P-ből
    $x_k := \frac{2}{n} \sum_{i=1}^{n} p_i - p_{n+1}$
    if ($x_k \in X$ ÉS $f(x_k) < y_{k-1}$)
        cseréljük ki pmax_k-t x_k-ra P-ben.
    $y_k = f(\text{pmax}_k) = \max_{p \in P} f(p)$

12.8. UPCR - Málna-algoritmus

Az eljárást elsősorban azért fejlesztették, hogy képesek legyenek egy $S(f^* + \delta)$ szintvonalat lefedni, ami egy konfidenciatartományt reprezentálna nemlineáris paraméterbecslésnél. Az ötlet az, hogy az $S(f^* + \delta)$ fedését keressük P-beli mintapontokkal mintha azok egyenletes eloszlásból lennének, vagy egy u.n. Málna-halmazból $R = \{x \in X | \exists p \in P, \|x - p\| \leq r\}$ ahol r egy kicsi sugár.

12.8.1. algoritmus UCPR$(f, X, N, c, f^* + \delta)$

$k = N$
Generálunk és kiértékelünk N véletlenül egyenletesen választott X-beli pontot (P).
$y_k = f(\text{pmax}_k) = \max_{p \in P} f(p)$
while $(y_k > f^* + \delta)$ **do**
    $k = k + 1$
    Határozzuk meg az átlagos pontközi távolságot (r_k) P-ben.
    Málna halmaz: $R_k = \{x \in X | \exists p \in P, \|x - p_i\| \leq c \cdot r_k\}$
    Generáljuk x_k-t egyenletesen R_k-ban.
    if ($x_k \in X$ ÉS $f(x_k) < y_{k-1}$)
        Cseréljük pmax_k-t $x_k \in P$-re.
    $y_k = f(\text{pmax}_k) = \max_{p \in P} f(p)$

12.9. Szimulált hűtés

A fenti két algoritmusnak bőséges irodalma van. Jóval nehezebben elemezhetőek, de alkalmazásokban nagyon népszerű módszerek az u.n. meta-heurisztikák. Ide tartozik a Szimulált hűtés, evolúciós algoritmus, genetikai algoritmus, tabu keresés.
Fontos kérdés, hogy vajon tényleg jobbak tudnak-e lenni ezek a módszerek, mint a klasszikus módszerek valamilyen ötvözete? Most bemutatjuk az u.n. szimulált

lágyítást. Ez egy olyan mintavételezési folyamatot ír le a döntési térben, ahol az új mintapontokat a jelenlegi iteráció egy u.n. szomszédságából választjuk. Az új mintapontokat mindig elfogadjunk, ha jobbak, és p valószínűséggel ha rosszabbak. A valószínűség az u.n. hőmérséklettől függ, ami csökken az iterációk során.

Az algoritmus tartalmazza a CR (cooling rate) paramétert, ami a hőmérséklet csökkenését szabályozza. A kezdőhőmérsékletet itt fix 1000-nek választjuk. Az algoritmus egy rossz pontot attól függően fogad el, hogy az mennyire rossz, illetve mennyire előrehaladott az algoritmus. Ez egy általános koncepció a szimulált lágyításban. Számos út kínálkozik a szomszédságból való mintavételezésre. Egy dimenzióban intuitívan az $[x_k - \varepsilon, x_k + \varepsilon]$ választás a kézenfekvő. Mivel ezek a heurisztikák kezdetben nem folytonos, hanem egészértékű problémákra irányultak, az egyik első megközelítés szerint a folytonos változókat bitláncként kódolhatjuk. Szemléltetésül: Az $[3,7]$ minden pontját egy $(B_1, \dots, B_9) \in \{0,1\}^9$ bitlánc reprezentálja, ahol

$$x = 3 + 4\frac{\sum_{i=1}^{9} B_i 2^{i-1}}{511}$$

Ez egy rácsot ír le, ahol minden bitláncnak egy rácspont felel meg, így a rasztertávolság $\frac{4}{511}$. A szomszédságból való mintavételezésen az egyik B_i átbillentését értjük. Az így kapott új pontokat nem feltétlenül tekintenénk a folytonos értelemben szomszédságnak.

12.9.1. algoritmus SA$(X, f, \mathrm{CR}, N$

$f^U = \infty, \quad T_1 = 1000$
Legyen x_1 egyenletes eloszlás szerint generált pont X-ből.
for $(k = 1, \dots, N)$ **do**
    Generáljuk x-et x_k egy környezetéből
    if $(f(x) < f(x_k))$
        $x_{k+1} = x$
        if $(f(x) < f^U)$
            $f^U = f(x)$ és $x^U = x$
    else
        $e^{\frac{f(x_k)-f(x)}{T_k}}$ valószínűséggel $x_{k+1} = x$
    $T_{k+1} = \mathrm{CR} \cdot T_k$

A függelék

Alapfogalmak

A.1. Definíció. Egy $D \subseteq \mathbb{R}^n$ halmaz *kompakt*, ha korlátos és zárt.

A.2. Definíció. Egy halmaz *konvex*, ha bármely két pontját összekötő szakasz a halmaz része.

A.3. Definíció. Egy A $n \times n$-es szimmetrikus mátrix *pozitív definit*, ha bármely $x \in \mathbb{R}^n$ nem nulla vektor esetén

$$x^T A x > 0.$$

Másrészről A pontosan akkor pozitív definit, ha minden főminora pozitív, illetve ha minden sajátértéke pozitív.

A.4. Definíció. Egy A $n \times n$-es szimmetrikus mátrix *negatív definit*, ha bármely $x \in \mathbb{R}^n$ nem nulla vektor esetén

$$x^T A x < 0.$$

Másrészről A pontosan akkor *negatív definit*, ha minden $k \times k$-s főminora pozitív páros k-kra és negatív minden páratlan k-ra. Ezzel szintén ekvivalens, hogy minden sajátértéke negatív.

A.5. Definíció. Egy A $n \times n$-es szimmetrikus mátrix *pozitív szemidefinit*, ha bármely $x \in \mathbb{R}^n$ nem nulla vektor esetén

$$x^T A x \geq 0.$$

Másrészről A pontosan akkor pozitív szemidefinit, ha minden főminora nemnegatív, illetve ha minden sajátértéke nemnegatív.