

# Globális optimalizálási algoritmusok intervallum korlátos feladatokra

Doktori értekezés tézisei

Pál László

Témavezető:  
Dr. Csendes Tibor  
egyetemi tanár

Szegedi Tudományegyetem  
Informatika Doktori Iskola  
Szeged, 2010

# 1. Bevezetés

A globális optimalizálás egy multidiszciplináris tudományterület, az alkalmazott matematikának egy olyan ága, amelyben a cél megtalálni egy feladat lehető legjobb megoldását, eleget téve bizonyos feltételeknek. Tehát a cél jellemezni, és megtalálni egy általában nem konvex, több szélsőértékkel rendelkező célfüggvény globális minimumát. Egy ilyen feladat globális optimumának megkeresése komoly erőfeszítéseket igényel és kiszámítása általában NP-nehéz probléma. Másfelől, sok valós gyakorlati feladat globális optimalizálási feladatként fogalmazható meg, ezért egy ilyen feladat globális optimumának a megkeresése igazi kihívást jelent. A sok gyakorlati alkalmazásnak köszönhetően az utóbbi időben egyre nagyobb az érdeklődés a kutatók részéről, hatékony módszerek kidolgozására.

A disszertációban a folytonos globális optimalizálás két fontos témakörével, a sztochasztikus- valamint az intervallum aritmetikán alapuló módszerrel foglalkozunk. Célunk volt olyan hatékony algoritmusok kidolgozása és tanulmányozása, amelyek segítségével kezelni tudjuk az intervallum korlátos globális optimalizálási feladatokat.

Az intervallum korlátos globális optimalizálási feladat a következő formában írható:

$$\min_{x \in X} f(x), \quad (1)$$

ahol  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  egy skalár értékű függvény,  $X = \{a_i \leq x_i \leq b_i, i = 1, 2, \dots, n\}$  a lehetséges megoldások halmaza. Általában az  $f$  célfüggvényről feltételezzük, hogy kétszer folytonosan differenciálható, habár ez nem mindig szükséges, ugyanis a tárgyalt módszerek nem sima függvényekre is jól működnek.

## 2. Sztochasztikus globális optimalizálás

A globális optimalizálási feladatok megoldására nem létezik olyan algoritmus, amely véges időben pontos eredményt ad, ezért a sztochasztikus módszerek fontos szerepet játszanak a globális optimalizálásban. A sztochasztikus módszerek lényege, hogy véletlenszerűen pontokat generálnak a keresési tartományban. A kapott pontokban kiértékelik a célfüggvényt, majd ez alapján próbálnak következtetni a globális minimumra. Ezek a módszerek nem garantálják a globális optimum megtalálását, viszont a mintapontok számának növelésével egy valószínűséggel aszimptotikusan konvergálnak a globális optimumhoz.

## 2.1. A GLOBAL optimalizálási módszer

A GLOBAL (Csendes [3]) egy sztochasztikus optimalizálási algoritmus, amely alapjául Boender és társszerzői [1] algoritmusát szolgálta. Korábbi összehasonlító tanulmányok alapján (Mongeau és társszerzői [15]; Moles és társszerzői [14]), a módszer hatékony és megbízható, valamint sok esetben a legjobb megoldást adta.

A módszer célja az összes helyi minimum megtalálása, ezért helyi kereséseket indít egyenletes eloszlás alapján generált induló pontokból. A helyi keresés általában nagyon időigényes művelet, ezért ezek számának csökkentésére, valamint a helyi minimum pontok vonzáskörzetének az azonosítására klaszterezési módszereket alkalmaz. A módszer lépéseit az alábbiakban adjuk meg.

---

### Algorithm 1. A GLOBAL módszer

---

```
function GLOBAL( $f, X$ )
   $k := 0; X^* := \emptyset; X^{(1)} := \emptyset$ 
  repeat
     $k := k + 1$ 
    Generálunk  $N$  pontot  $(x_{(k-1)N+1}, \dots, x_{kN})$  egyenletes
    eloszlással  $X$ -en
    Az  $x_1, \dots, x_{kN}$  halmazból kiválasztjuk a  $\gamma kN$  darab legjobb
    pontot (ez lesz a redukált mintahalmaz)
    Klaszterezük az  $X^*$  és  $X^{(1)}$  halmazokat
    while az összes pont a redukált mintahalmazból nincs
    hozzárendelve egy klaszterhez do
      Legyen  $x^{(1)}$  pont a legkisebb célfüggvényértékű
       $x^* := \text{Loc}(x^{(1)})$ 
      if  $x^* \notin X^*$  then
         $X^* := X^* \cup \{x^*\}$ 
        Legyen  $x_s := x^*$  a következő mag pont
      else
         $X^{(1)} := X^{(1)} \cup \{x^{(1)}\}$ 
        Legyen  $x_s := x^{(1)}$  a következő mag pont
      end
      Minden klaszterezetlen pontot hozzáadunk az  $x_s$  mag
      pont által inicializált klaszterhez, amely pont  $r_k$ 
      távolságon belül van a klaszter egy tetszőleges pontjától
    end
  until Valamilyen megállási feltétel teljesül
  return A legkisebb függvényértékű pont
```

---

A GLOBAL algoritmus több módosítást és javítást tartalmaz a Boender és társai módszeréhez képest. Ezek az alábbiak:

- A Single Linkage klaszterezési módszert használja.
- A klaszter távolság kiszámításában nem használja a Hesse mátrixot.
- Nem használja a gradiens feltételt.
- Nem teszi meg a legmeredekebb lejtőnek megfelelő lépést a kiinduló minta transzformálásához.
- Nem számol konfidencia intervallumot a globális minimum értékére.
- Az eredeti problémát skálázza a jobb numerikus stabilitás érdekében.

A fenti módosításokon túl a GLOBAL tartalmaz két új helyi kereső eljárást: egy kvázi-Newton eljárást a DFP (Davidon-Fletcher-Powell) formulával, illetve egy véletlen sétát használó közvetlen keresési módszert, az UNIRANDI (Järvi [11]) eljárást. Ezt akkor használhatjuk, ha a problémából adódóan nem használhatjuk ki annak kvadratikus alakját.

A GLOBAL módszert az 1980-as években közölték és alkalmazták globális optimalizálási feladatokra. Az azóta eltelt idő alatt a számítógépes környezet sokat változott. Ezért az volt a célunk, hogy a régi módszert újragondolva és módosítva alkalmazzuk az új számítógépes környezetben, javítva annak hatékonyságát és megbízhatóságát. Az elvégzett munka nagy része olyan kísérletezés és javítás volt, amelyek eredményeképpen a GLOBAL hatékonyabb, illetve megbízhatóbb lett.

A régi algoritmust számos ponton sikerült javítani, amelyek közül a fontosabbak:

- MATLAB környezetben van kódolva, használva ennek tömbösített műveleteit, így növelve a módszer hatékonyságát.
- A DFP helyi kereső helyett a korszerűbb BFGS (Broyden-Fletcher-Goldfarb-Shanno) (Broyden [2]) módszert használjuk.
- Hatékonyabb egyenletes- és normál eloszlású véletlen számgenerátort használunk.
- Az UNIRANDI helyi keresőt sikerült úgy javítani, hogy segítségével magasabb dimenziójú feladatokat is meg tudunk oldani.

Az implementálás során kihasználtuk a MATLAB előnyeit, ezáltal hatékony kódot kaptunk. A hatékonyságot a MATLAB tömbműveleteinek segítségével értük el, amelyek teljes mértékben kihasználják a processzor csővezetékét (pipeline), ezáltal akár nagyobb tömbökkel is gyorsan tudunk dolgozni. Tehát a tömbműveletek segítségével sikerült a GLOBAL hatékonyságán javítani (a CPU idő szempontjából). Az új módszer magasabb dimenziójú feladatok megoldására is képes a korábbihoz hasonló megbízhatósággal.

A BFGS helyi kereső módszer hasonlóan működik, mint a DFP algoritmus. A lényeges különbség a kettő között az, hogy az első módszer egy másik frissítési formulát alkalmaz. Az összehasonlító eredmények (Powell [22]) alapján a BFGS alapú kvázi-Newton módszer hatékonyabb, mint a DFP alapú.

Az UNIRANDI helyi kereső két fő részből áll: véletlen irányválasztás és egyenesmenti keresés. A véletlen irányválasztás azt jelenti, hogy egy véletlenül meghatározott irányú egyenest fektetünk a kezdőpontra. Ha a ponttól adott távolságra lévő két pont közül egyik is jobb, mint a kezdőpont, akkor egyenes menti keresést indítunk ezek közül a jobb irányába. A keresési irányokat egyenletes eloszlás alapján generáltuk a  $[-0.5, 0.5]^n$  intervallumban, viszont csak akkor fogadtuk el, ha az irány normája kisebb vagy egyenlő volt, mint 0.5. Ez a feltétel azt jelenti, hogy eldobtuk azon pontokat, amelyek a 0.5 sugarú hipergömbön kívül estek, ezáltal biztosítva az egyenletes eloszlást a hipergömbön belül. A dimenzió növekedésével persze egyre nehezebb olyan pontokat találni, amelyek megfelelnek az előbbi feltételnek, és 15 dimenzió felett már gyakorlatilag nem teljesíthető a feltétel.

A fenti probléma megoldására módosítottuk az UNIRANDI helyi keresőt úgy, hogy a keresési irányokat normális eloszlás ( $\mathcal{N}(0, 1)$ ) alapján generáltuk és a megfelelő vektorokat normalizáltuk.

## 2.2. Numerikus eredmények

Két különböző numerikus tesztet végeztünk az új, javított GLOBAL módszerrel. Az elsőnek az volt a célja, hogy összehasonlítsuk az új módszer és a régi eljárás (Csendes [3]) hatékonyságát és megbízhatóságát. A második esetben összehasonlítottuk a C-GRASP nevű módszerrel (Hirsch és társszerzői [10]), amely a GLOBAL-hoz hasonlóan kétfázisú eljárás, és egy korábbi módszer (Feo és Resende [9]) kiterjesztése a folytonos esetre.

A GLOBAL algoritmus esetén a következő hat paraméter beállítására van lehetőség: egy iterációban dobott mintapontok száma, a redukált

mintahalmazból kiválasztott legjobb pontok száma, a helyi kereső megállításhoz szükséges érték, a függvénykiértékelések maximális száma, a végrehajtott lokális keresések maximális száma, valamint az alkalmazott lokális kereső algoritmus. Valamennyi paraméter rendelkezik alapértelmezett értékkel, és általában elég, ha az első hármát változtatjuk a tesztelesek során.

Az első összehasonlító tesztben, a régi GLOBAL esetén is használt függvényeket vizsgáltuk. A futási idő mérésére a standard időegységet használtuk (a Shekel-5 függvény kiértékelése az  $x^T = (4.0, 4.0, 4.0, 4.0)^T$  pontban 1000-szer). Minden feladatra 100 független futást (korábban csak 10 volt) hajtottunk végre, és meghatároztuk az átlagos függvényhívások számát, valamint az átlagos CPU időt standard időegységben mérve. Az új algoritmus paramétereit úgy állítottuk be, hogy az minden esetben megtalálta a globális optimumot.

Az összehasonlítások eredményeként elmondható, hogy a vizsgált standard tesztfüggvény halmazon az új GLOBAL hatékonysága legalább annyira jó, mint a régi algoritmusé, míg a megbízhatóságot sikerült jelentősen javítani. Az új kvázi-Newton keresőnek köszönhetően az új algoritmus sokkal jobban teljesít sima függvényeken a szükséges függvényhívások száma tekintetében.

A C-GRAPS egy deriváltmentes globális optimalizáló módszer, ezért a második összehasonlító tesztben a GLOBAL eljárást az UNIRANDI helyi keresővel használtuk. A tesztfüggvények halmaza a C-GRASP esetén használt 14 függvényből állt, amelyekre ismert volt a globális optimum értéke. Mindkét módszer esetén ugyanazt a megállási feltételt alkalmaztunk. Tehát mindkét algoritmus akkor áll le, ha a célfüggvény értéke megfelelő pontossággal közelíti a globális optimum értékét (azaz  $|f^* - f| \leq 10^{-4}|f^*| + 10^{-6}$ ). A GLOBAL ezen kívül akkor is leállhat, ha az utolsó iterációban nem talál új lokális minimumot.

A teszt során minden feladatra 100 független futást hajtottunk végre. Vizsgáltuk a sikeres futások százalékos számát, a szükséges CPU időt, valamint a függvényhívások számát. Ebben az esetben is úgy állítottuk be az algoritmus paramétereit, hogy valamennyi esetben megtalálja a globális optimumot.

Összegezve az eredményeket, elmondható, hogy az új GLOBAL módszer kihasználja a MATLAB nyújtotta előnyöket és a módosításoknak köszönhetően magasabb dimenziós feladatokat is tudunk kezelni. Az algoritmus megbízhatósága és hatékonysága szintén javult, és a részletes összehasonlítások alapján jobban teljesít mint a régi változat, valamint a C-GRASP módszer. A teszteredmény és az összehasonlítás megtalálható

a Csendes és társszerzői [6] cikkben.

A GLOBAL módszer teljesítményét vizsgáltuk a BBOB (Black-Box Optimization Benchmarking) 2009 zajnélküli tesztfüggvény halmazon is. Valamennyi feladat úgy volt megtervezve, hogy tükrözze a valós feladatokban előforduló nehézségeket. A teszteredmények azt mutatják, hogy a GLOBAL algoritmus jól teljesít kis függvényhívás számig olyan függvényeken, amelyek nem túl sok lokális minimummal rendelkeznek. A GLOBAL a legjobb algoritmusok között szerepelt  $500n$  függvényhívás számig, viszont magasabb függvényhívás szám esetén már gyengébben teljesített. Az eredmények megtalálhatók a (Pál és társszerzői [20]; Pošík és társszerzői [21]) cikkekben.

### 2.3. Alkalmazás

A vizsgált feladat: optimális járadékfüggvény tervezése rugalmas nyugdíjrendszerre (Eső és Simonovits [7, 8]). Feltételezzük, hogy az egyéneknek információjuk van saját várható élettartamukról. A kormányzat célja: egy olyan nyugdíjmechanizmus tervezése, amely maximalizál egy társadalmi jóléti függvényt, és kielégít egy társadalmi költségvetési korlátot.

A probléma matematikai modellje:

$$\max_{(b_t, R_t)_t} \sum_{t=S}^T \psi(v_t) f_t, \quad (2)$$

amelyre

$$v_t = [\bar{u} - w(b_t)]R_t + w(b_t)t, \quad t = S, \dots, T, \quad (3)$$

$$\sum_{t=S}^T [(\tau + b_t)R_t - tb_t] f_t = 0, \quad (4)$$

$$v_{t+1} = v_t + w(b_t), \quad t = S, \dots, T-1. \quad (5)$$

A fenti modellben  $t$  egy egyén típusa, amely a várható élettartamot jelenti,  $f_t$  a  $t$  típus gyakorisága,  $\tau$  a járulékulcs,  $b_t$  az éves életjáradék, amelyet  $R_t$  évesen kap az egyén,  $v_t$  az életpálya-hasznosságfüggvény, amely a dolgozói és nyugdíjas szakasz összege. A feladatban megköveteljük, hogy a népesség átlagos várható egyenlege nulla legyen ((4)-es egyenlet) és bevezetjük a szomszédos érdekeltégi feltételeket ((5)-ös egyenlet), amely szerint a  $t$  típusnak nem érdemes  $t+1$  típusúnak mondania magát. A cél egy optimális  $b(R)$  nyugdíjjáradék-szolgálati idő séma tervezése, amely maximalizál egy konkáv társadalmi jóléti függvényt ((2)-es függvény).

A modell nem egy egyszerű korlátos optimalizálási feladat, ugyanis tartalmaz egyenlőség típusú feltételeket. Ezért a megoldás során a büntetőfüggvények módszerét alkalmaztuk ezek kezelésére. A numerikus tesztek alapján elmondható, hogy a GLOBAL jó közelítő megoldást talált az optimum helyre, valamint a megfelelő futási idő is elfogadható. A részletes eredmények megtalálhatók a Pál és Csendes [16] cikkben.

### 3. Intervallumos globális optimalizálás

Az intervallum aritmetikát használó globális optimalizálási módszerek megbízható módon találják meg a globális optimumot. Ezek a módszerek többnyire a korlátozás és szétválasztás elvét használják, azaz a keresési teret felosztják részintervallumokra, a részintervallumokon alsó és felső korlátot adva a célfüggvény lehetséges értékeire, majd eldobják azokat, amelyek nem kapnak jobb megoldást, mint az eddig ismert legjobb. Ezt az eljárást kombinálhatjuk az intervallum aritmetikai eszköztárával, amely természetes módon szolgáltatja a megfelelő alsó és felső korlátokat az egyes részfeladatokra. Az alap intervallumos korlátozás és szétválasztás módszer lépései a 2. Algoritmusban vannak megadva.

Célunk volt egy könnyen használható, megbízható optimalizálási módszer kidolgozása és implementálása MATLAB környezetben, használva az INTLAB csomagot, amellyel hatékonyan meg tudjuk oldani az intervallum korlátos globális optimalizálási feladatot.

#### 3.1. Javasolt algoritmusok

Az alaplómódszerre támaszkodva két másik intervallum alapú algoritmust dolgoztunk ki, amelyek csak egy, a célfüggvény kiszámítására szolgáló szubrutinra támaszkodnak. Más információt nem használnak a globális optimalizálási problémára vonatkozóan.

Az első algoritmus egy egyszerű változat, amely nem támaszkodik a célfüggvény deriváltjára, valamint Hesse mátrixára, habár ezeket tudjuk számolni automatikus differenciálással. Tehát ebben az esetben olyan algoritmust tanulmányoztunk, amely nem feltételezi a célfüggvény deriválhatóságát, azaz nem sima függvényekre is tudjuk alkalmazni. A befoglaló függvény kiszámítására természetes befoglalást használunk, valamint az intervallumok felosztására sima kettéosztást alkalmazunk a legszelesebb komponensre merőlegesen, eltekintve a szeleteléstől. A felosztási irány megválasztása az A típusú szabály alapján történik.



---

**Algorithm 2.** Az alap intervallumos módszer

---

```
function IntervalBranchAndBound( $f, X$ )
   $L_{res} := \emptyset$ ;  $L_{work} := \{X\}$ 
  while  $L_{work} \neq \emptyset$  do
    Válasszuk ki az  $X$  intervallumot az  $L_{work}$  listáról
    Számítsuk ki az  $f(X)$  egy befoglalását
    if  $X$ -et nem lehet eldobni then
      osszuk fel  $X$ -et  $X^i$ ,  $i = 1, \dots, p$  részintervallumokra
      if  $X^i$  kielégíti a megállási feltételt then
        tároljuk  $X^i$ -t az  $L_{res}$  listában
      else
        tároljuk  $X^i$ -t az  $L_{work}$  listában
      end
    end
  end
  return  $L_{res}$ 
```

---

A második algoritmus egy fejlettebb változat, amely a következő gyorsító tesztek tartalmazza: középponti teszt, konkavitási teszt, monotonitási teszt és intervallumos Newton lépés. A természetes befoglaláson kívül a középponti formulákat is alkalmazzuk, mint befoglaló függvényt. Ha a gradiens befoglalása ismert, akkor az előbbi két befoglaló függvény metszete általában jó megközelítést ad a függvény értékkészletére.

A keresési tartományok felosztására szeletelést, illetve fejlett felosztási irányt választó szabályokat (Kearfott [12]) használunk, eltekintve a  $pf^*$  heurisztikán (Csendes [4]) alapuló változattól. A szeletelés azt jelenti, hogy az intervallumot felosztjuk három másik intervallumra a két legmegfelelőbb irányt használva. A felosztási irány megválasztása a C típusú szabály (Csendes [4]; Kearfott [12]) alapján történik.

Mindkét algoritmus implementálása során követtük a Markót Mihály Csaba által készített C-XSC alapú program (Markót és társszerzői [13]) szerkezetét. Természetesen ahol lehetett, használtuk a MATLAB vektor struktúráit.

Összegezve a numerikus eredményeket, a MATLAB alapú implementációk esetén a hatékonysági mutatók (egy, a CPU idő kivételével) megegyeznek, vagy hasonlóak a C-XSC alapú program hasonló mutatóival. A CPU idő esetén egy-két nagyságrendbeli különbség figyelhető meg, amely a MATLAB interpreter módban való működésének tudható be. Az összehasonlítás eredményeképpen elmondható, hogy MATLAB kör-

nyezetben az INTLAB alapú algoritmus egyszerűen használható, viszont hátránya a sebesség visszaesése. Ennek ellenére az új globális optimalizálási módszer hasznos modellező eszköz lehet optimalizálási feladatok kezdeti tanulmányozására. Különösen igaz ez olyan feladatok esetén, amelyekre a CPU idő a növekedés ellenére is elfogadható mértékű (legfeljebb pár perc). Az eredmények megtalálhatók a (Csendes és Pál [5]; Pál és Csendes [17]) cikkekben.

### 3.2. A módosított Newton lépés

A Newton lépés az egyik legfontosabb gyorsító teszt, amelyet algoritmusunkban használunk. Tulajdonképpen egy intervallumos Newton Gauss-Seidel lépést alkalmazunk a célfüggvény gradiensére, ezáltal megpróbálunk közeli korlátokat adni a minimum helyekre. A teljes Newton algoritmust nem futtatjuk le, mert az túl költséges lenne, ezért alkalmazunk csak iterációnként egy-egy lépést.

A Newton lépés alkalmazása minden egyes részintervallumra szintén költséges lenne, ezért célszerű valamilyen feltételt használni ennek bekapcsolására. Egy korábbi cikk (Markót és társszerzői [13]) alapján algoritmusunkban (Pál és Csendes [17]) a Newton lépést csak abban az esetben használtuk, ha a felosztás során keletkezett három részintervallumból a többi gyorsító lépés végrehajtása után csak egy részintervallum maradt.

Célunk volt egy új bekapcsolási feltétel bevezetése. Ennek az a lényege, hogy minden olyan részintervallumra alkalmazzuk a Newton lépést, amelynek szélessége kisebb, mint egy előre megadott érték. Az indok az, hogy a Newton lépés igazából csak akkor hatékony, ha az adott argumentum intervallum már elég kicsi. Az új feltételben 0.1 értéket használtunk az intervallum szélessége küszöbértékének.

Implementáltunk egy új MATLAB/INTLAB algoritmus változatot, amely tartalmazza az új feltételt és összehasonlítottuk a régi feltételt tartalmazó algoritmus változattal. Az új algoritmus hasonló a korábbi módszerhez (Pál és Csendes [17]). A fontosabb különbségek: a lista rendezésére a  $pf^*$  heurisztikus paramétert használjuk, valamint az algoritmus leáll az első olyan intervallumnál, amely teljesíti a megállási feltételt. Tehát az algoritmus nem keresi meg az összes globális minimum pontot, hanem az elsőnél leáll, ha az azt tartalmazó intervallum befoglalásának szélessége kisebb, mint egy adott érték.

Az összehasonlítás (Pál és Csendes [18]) eredményeként elmondható, hogy az új feltétel segítségével sikerült jelentősen csökkenteni a teljes futásidőt, valamint a Hesse mátrix kiértékelések számát.

### 3.3. A Newton lépés bekapcsolásának vizsgálata

Az előbbi vizsgálat alapján látható, hogy a Newton lépés bekapcsolási feltételében egy jól megválasztott küszöbérték esetén sikerült javítani bizonyos hatékonysági mutatókat. Vannak azonban esetek, amikor az előbbi feltételt alkalmazva a Newton lépés nem eredményes abban az értelemben, hogy vagy nem is csökken az intervallum mérete, vagy sok darabra osztja fel az aktuális intervallumot. A gyakorlatban többnyire az utóbbi eset szokott előfordulni, amely azért nem előnyös, mert hasonló eredményt érhetünk el egyszerű kettéosztással is, de jóval kisebb költséggel.

A Newton lépés bekapcsolására használt új feltétel elméleti hátterét is megvizsgáltuk, a másodrendű derivált egy előre megadott befoglalófüggvény osztályán. Az eredmények (Pál és Csenedes [18]) alapján elmondható, hogy sikerült jellemezni azokat az eseteket, amelyekre a Newton lépés nem eredményes a fenti értelemben.

### 3.4. Alkalmazás

Az intervallumos technikákat két valós feladat esetén alkalmaztuk (Pál és Csenedes [18, 19]). A disszertációban többnyire a szenzorhálózatok lokalizálásával foglalkoztunk, amelyben bizonyos csomópontok földrajzi helyzetét próbáltuk megbecsülni, kis számú csomópont ismert pozíciója, valamint a szomszédos csomópontok közötti zajos mérések alapján. A feladat megfogalmazható globális optimalizálási problémaként is, amelyben célunk minimalizálni a szomszédos csomópontok közötti távolságok összegének hibáját.

A fenti célfüggvény minimumának megkeresése az INTLAB alapú optimalizálóval nagyon időigényes művelet, főleg több száz csomópont esetén. Ezért egy közelítő megoldás megtalálására az ívmetszés technikáját alkalmaztuk, amelynek lényege, hogy egy ismeretlen csomópont pozíciója meghatározható három ismert helyzetű szomszédja segítségével.

Az iteratív ívmetszést alkalmazva kezdeti közelítő megoldásokat találtunk a szenzorok pozíciójára. Az eredmények alapján (Pál és Csenedes [18]), a lokalizálási hiba csökkenthető, ha az ismert helyzetű csomópontok számát, vagy a hatókör sugarát növeljük.

A publikált cikkek, valamint a GLOBAL és az INTLAB alapú módszerek MATLAB implementációi az alábbi címen érhetőek el

<http://www.emte.siculorum.ro/~pallaszlo/Disszertacio>

## Hivatkozások

- [1] C.G.E. Boender, A.H.G. Rinnooy Kan, G.T. Timmer, and L. Stougie. A stochastic method for global optimization. *Mathematical Programming*, 22:125–140, 1982.
- [2] C.G. Broyden. The convergence of a class of double-rank minimization algorithms. *Journal of the Institute of Mathematics and Its Applications*, 6:76–90, 1970.
- [3] T. Csendes. Nonlinear parameter estimation by global optimization – efficiency and reliability. *Acta Cybernetica*, 8:361–370, 1988.
- [4] T. Csendes. New subinterval selection criteria for interval global optimization. *Journal of Global Optimization*, 19:307–327, 2001.
- [5] T. Csendes and L. Pál. A basic interval global optimization procedure for Matlab/INTLAB. In *Proceedings of International Symposium on Nonlinear Theory and its Applications (NOLTA2008)*, pages 592–595, Budapest, Hungary, 2008.
- [6] T. Csendes, L. Pál, J.O.H. Sendín, and J.R. Banga. The GLOBAL optimization method revisited. *Optimization Letters*, 2:445–454, 2008.
- [7] P. Eső and A. Simonovits. Designing Optimal Benefit Rules for Flexible Retirement. Technical Report CMS-EMS 1353, Northwestern University, Evanston, 2002.
- [8] P. Eső and A. Simonovits. Optimális járadékfüggvény tervezése rugalmas nyugdíjrendszerre. *Közgazdasági Szemle*, 50:99–111, 2003.
- [9] T.A. Feo and M.G.C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.
- [10] M.J. Hirsch, C.N. Meneses, P.M. Pardalos, and M.G.C. Resende. Global optimization by continuous GRASP. *Optimization Letters*, 1: 201–212, 2007.
- [11] T. Järvi. A random search optimizer with an application to a max-min problem. *Publications of the Institute for Applied Mathematics*, (3), University of Turku, 1973.
- [12] R.B. Kearfott. *Rigorous global search: continuous problems*. Kluwer Academic Publishers, Dordrecht, 1996.

- [13] M.C. Markót, J. Fernandez, L.G. Casado, and T. Csendes. New interval methods for constrained global optimization. *Mathematical Programming*, 106:287–318, 2006.
- [14] C.G. Moles, G. Gutierrez, A.A. Alonso, and J.R. Banga. Integrated Process Design and Control Via Global Optimization – A Wastewater Treatment Plant Case Study. *Chemical Engineering Research and Design*, 81:507–517, 2003.
- [15] M. Mongeau, H. Karsenty, V. Rouzé, and J.B. Hiriart-Urruty. Comparison of public-domain software for black box global optimization. *Optimization Methods and Software*, 13:203–226, 2000.
- [16] L. Pál and T. Csendes. Improvements on the GLOBAL Optimization Algorithm with Numerical Tests. In *Proceedings of the 8th International Conference on Applied Informatics (ICAI2007)*, pages 101–109, Eger, Hungary, 2007.
- [17] L. Pál and T. Csendes. INTLAB implementation of an interval global optimization algorithm. *Optimization Methods and Software*, 24:749–759, 2009.
- [18] L. Pál and T. Csendes. Egy intervallum alapú globális optimalizálási módszer és alkalmazása szenzor lokalizálási feladatra. Közlésre beküldve, 2010.
- [19] L. Pál and T. Csendes. Efficient estimation of loads in service networks. Accepted for publication, 2010.
- [20] L. Pál, T. Csendes, M.C. Markót, and A. Neumaier. Black-box optimization benchmarking of the global method. Submitted for publication, 2010.
- [21] P. Pošík, W. Huyer, and L. Pál. Benchmarking Several Global-Optimization Algorithms. Submitted for publication, 2010.
- [22] M.J.D. Powell. How bad are the BFGS and DFP methods when the objective function is quadratic? *Mathematical Programming*, 34: 34–47, 1986.