

PSO ALGORITMUS BŐVÍTÉSE OPTIMÁLÁSHOZ, VÉGES DIFFERENCIA ALAPÚ GRADIENS BECSLÉSSEL

IMPROVING PSO ALGORITHM WITH FINITE DIFFERENCE BASED GRADIENT ESTIMATION FOR OPTIMIZATION NEEDS

Barcsák Csaba, BSc hallgató;

Prof. Dr. Jármai Károly, DSc, egyetemi tanár, Miskolci Egyetem

ABSTRACT

A new method is proposed which estimates the gradient in the sample points using a finite difference based technique. This method doesn't need more function evaluations than the standard algorithm, and it's efficiency doesn't depend on the initial state of the process. The efficiency of the method depends on how, and when we change the particle's speed knowing the gradient information in the previous sample points. These parameters can be different in every objective function. We have applied the method at several test problems, the result of two test examples are shown. Further research is necessary to apply the technique for real structural optimization.

1. BEVEZETÉS

Az optimalizációs problémák a tudomány sok területén megtalálhatóak. Ezen problémák nagyon összetettek is lehetnek a célfüggvényekből és a feltételekből adódóan, melyeket analitikus módszerekkel nem, vagy csak nagyon nehezen lehet megoldani, emiatt az évek során sok különböző algoritmus született a megoldásukra. A deriváltat használó technikák sok esetben hatékonyak, a hátrányuk viszont az, hogy könnyen elakadnak lokális szélsőértéknél, valamint összetett célfüggvény esetén számításiigényessé válnak. A heurisztikus optimalizációs módszerek nem rendelkeznek a deriváltat használó módszerek előbb említett hátrányaival, viszonylag könnyedén implementálhatóak, emiatt nagy népszerűsége tettek szert az optimalizálással foglalkozók körében. Ezen algoritmusok közé tartozik az Ant colony algoritmus

[1], mely hangyák viselkedését szimulálja, a genetikus algoritmusok [6], melyek evolúciós folyamat modellezésével oldják meg a problémát, valamint ide tartozik az ún. PSO algoritmus is.

2. PSO ALGORITMUS

Kennedy és Eberhart 1995-ben mutatta be a PSO (Particle Swarm Optimization) algoritmust [4,7], mellyel az eredeti céljuk az volt, hogy madarak csapaton belüli szociális viselkedését szimulálják és vizualizálják. A kutatásaik során felfedezték, hogy ez a módszer optimalizációs feladatok megoldására is hatékonyan alkalmazható. A PSO első változata csak folytonos nemlineáris optimalizációs feladatokat volt képes megoldani. Az évek során az algoritmusnak nagyon sok változata jelent meg melyek optimalizációs problémák széles skáláját képesek megoldani. Ezen algoritmusok az egyszerűségük és hatékonyságuk miatt váltak széles körben elterjedtté a mérnöki gyakorlatban [3,8,9]. Lehet költségre is optimalizálni. Ebben az esetben a célfüggvény összetettebb [10,11]. A következőkben a standard algoritmus kerül ismertetésre.

Az algoritmus első lépésben ún. részecskéket generál. Minden részecske rendelkezik egy x pozícióval, és egy v sebességvektorral. Ezen vektorok elemszáma megegyezik a célfüggvény változóinak számával. A helyvektorok generálása a célfüggvény előre definiált tartományán egyenletes eloszlás szerint történik. A részecskék a megadott tartományon mozognak, és keresik az optimális megoldást. Minden részecske tárolja a mozgása során talált legjobb megoldást és annak pozícióját, ezeket lokális legjobb néven említi

az irodalom. Külön tárolásra kerül a lokális legjobbak közül a legjobb. Ezt nevezzük globális legjobbnak. A részecskék minden iterációs lépésben újabb mintát vesznek a célfüggvényből, valamint változtatják a pozíciójukat és sebességüket a következő egyenletek szerint.

$$v_i^{k+1} = v_i^k + c_1 r_1 (pbest_i - x_i^k) + c_2 r_2 (gbest_i - x_i^k) \quad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} t \quad (2)$$

ahol v_i a sebességvektor i -edik eleme, x_i a pozícióvektor i -edik eleme, c_1 és c_2 pozitív konstansok, r_1 és r_2 két egyenletes eloszlás szerint generált véletlen szám a $[0,1]$ intervallumon, $pbest_i$ az adott részecske lokális legjobb pozíciójának i -edik eleme $gbest_i$ pedig a globális legjobb pozíció i -edik eleme, a k index az adott iterációt jelöli, t az egységnyi időintervallum. Az algoritmus lépéseit az 1. ábra mutatja:



1. ábra

Standard PSO algoritmus folyamatábrája

Az 1. ábrán látható folyamatban a részecskék inicializálása a helyvektorok inicializálásaként értendő. Ezen vektorok egyenletes eloszlás szerinti véletlen értékeket kapnak, ügyelve arra, hogy a helyvektorok ne legyenek az előre definiált tartományon kívül.

A PSO algoritmus amiatt lett népszerű, mert működése könnyen megérthető, egyszerűen implementálható, könnyen integrálható más optimális eljárásokba, kevésbé érzékeny célfüggvényre,

beállításához kevesebb paraméter szükséges, mint más heurisztikus eljárásoknál. A hátránya, hogy nincs mögötte mély matematikai háttér.

3. PSO ALGORITMUS BŐVÍTÉSE GRADIENS BECSLÉSEL

Az előbbieken említésre került, hogy a PSO algoritmusnak sokféle változata jelent meg, hatékonyságának javítására többféle technikát találhatunk az irodalomban.

Az egyik ismert megoldás, hogy egyidejűleg több részecskecsoporttal dolgozunk az egy helyett, ekkor tároljuk részecskénként a lokális legjobb eredményt, az egyes csoportokhoz tartozó legjobb eredményt, és a részecskecsoportok legjobbjai közül a legjobbat.

Ekkor nemcsak az egyes részecskék között, hanem a részecskecsoportok között is értelmezett a kommunikáció, tehát az egyes részecskéknek a sebesség és pozíció változtatásánál, a lokális legjobb, a csoport legjobb, és a csoportok összességének legjobb eredményeit is figyelembe vesszük.

Egy másik megoldás az ún. crazy bird, az őrült madár. Ez a megoldás véletlenszerűen kiválaszt részecskéket, és ezen részecskék sebességét nem az (1) formula alapján változtatja, hanem véletlenszerű irányba, ezáltal kiszakít részecskéket a csoportból melyek nem a $gbest$ irányába tartanak, remélve ezzel azt, hogy más irányba jobb eredményt találunk mint a jelenlegi $gbest$.

Az előbbieken említett eljárások hatékonysága minden esetben függ a véletlentől. Nem tudhatjuk azt, hogy a részecskék több csoportra bontása, vagy véletlen irányba küldése által biztosan jobb eredményt érünk-e el, mint a standard algoritmus használata esetén.

A standard algoritmus a célfüggvény kiszámított értékein kívül más információkkal nem rendelkezik a függvényről, pedig több esetben hasznos lenne a függvény egyes lokális tulajdonságainak ismerete, mivel ezen információk nem kötődnek a véletlenhez és általuk hatékonyabbá tehetnénk az eljárást.

Az egyik ilyen lokális tulajdonság a gradiens, melyet, mivel csak diszkrét pontokban rendelkezünk mintákkal, becsülnünk kell.

A véges differencia alapú megoldások gyors és hatékony megoldást nyújtanak a gradiens becslésére diszkrét adatok esetén. Mindegyik véges differencia alapú séma a differenciálandó függvény Taylor-sorából indul ki, amely egy egydimenziós függvény $f(x)$ esetében a következőképp írható fel:

$$f(x_0 + h) \approx f(x_0) + \frac{f'(x_0)}{1!}h + \frac{f''(x_0)}{2!}h^2 + \dots = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}h^n \quad (3)$$

Abban az esetben, ha a sor kifejtését abbahagyjuk a második tagnál a formula a következő:

$$f(x_0 + h) \approx f(x_0) + \frac{f'(x_0)}{1!}h \quad (4)$$

Kifejezve a deriváltat a következő formula adódik:

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h} \quad (5)$$

Ezt a formulát nevezi az irodalom forward difference (előrevett differenciák) becslésnek. Akkor, ha $f(x_0+h)$ helyett $f(x_0-h)$ esetben fejtjük ki a sort az előbbiekben használt gondolatmenettel a következő eredményt kapjuk:

$$f'(x_0) \approx \frac{f(x_0) - f(x_0 - h)}{h} \quad (6)$$

Ezt a képletet az irodalom backward difference (hátravett differenciák) becslés néven említi. Ezek a megoldások egyszerűek, gyorsan kiszámolhatóak, hátrányuk viszont az, hogy kevésbé pontosak. Léteznek összetettebb gradienst becsülő megoldások az irodalomban, de azok számításgényesebbek az előbbiekben leírt eljárásoknál, és kettőnél több mintavételi pontot igényelnek.

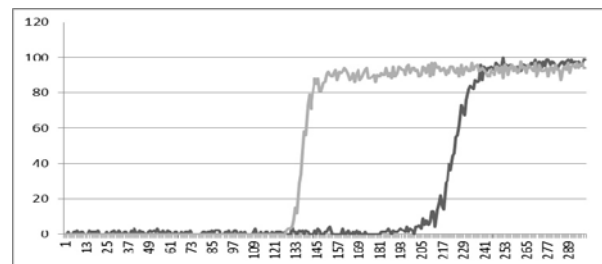
Egy adott részecske mozgása során adott pillanatig érintett pontokhoz tartozó függvény értékeket felhasználhatjuk az adott pontokban vett gradiensek becslésére. Az általunk implementált algoritmusban a backward difference módszer került beépítésre, mivel egyszerűbb volt implementálni. A gradienseket az elkészült algoritmus a részecskék sebességének beállítására használja, ezáltal a részecskék a függvény értelmezési tartományának egyes intervallumaiban gyorsabban, még más intervallumokban lassabban mozognak. Minden egyes részecske a pozíció és sebesség adatok mellett tárolja, hogy a már érintett pontokban hány egymás utáni esetben talált pozitív előjelű gradienst. Abban az esetben, ha ez túllép egy előre definiált konstans értéket, akkor az adott részecske sebességét növeljük, ha negatív előjelű gradienst találunk, vagy nem értelmezett a gradienst, akkor a sebesség visszaáll az alapértékre. Ha egy adott

részecske esetén nagy az egymás utáni mintavett pontokban egymást követő pozitív előjelű gradiensek száma, abból az következik, hogy a részecske ezen idő alatt nem haladt át lokális szélsőértéken, emiatt vagy a globális szélsőérték felé tart vagy egy olyan lokális szélsőértékhez, amely a részecske környezetében egy nagyobb intervallumon vett szélsőérték, emiatt gyorsíthatunk a sebességen, hogy a részecske kevesebb iterációs lépés alatt elérjen a szélsőértékg. A módszer eredményességét a következő fejezetben bemutatott szimulációk igazolják.

4. SZIMULÁCIÓS EREDMÉNYEK

A szimulációs eredmények előállításához egy Java alkalmazás került kifejlesztésre mely tartalmazza a standard valamint a gradienst használó megoldást, 12 feltétel nélküli [5] és 3 feltételes [2] kétdimenziós optimalizációs tesztfüggvényt. A 12 feltétel nélküli tesztfüggvényből 5 „szélsőséges” eset. Ez azt jelenti, hogy sok a lokális szélsőérték, és ez nehezen megoldhatóvá teszi a problémát.

Az eljárások úgy kerültek összehasonlításra, hogy mindkét algoritmust egy adott iterációs lépésig adott tesztfüggvényre százszor futtattunk. A következő ábrák vízszintes tengelyén az iterációs számok láthatóak, a függőleges tengelyen pedig az, hogy adott iterációs lépésnél a száz futtatásból hány esetben találta meg az adott algoritmus a megoldást. A standard algoritmus eredményét minden esetben sötétszürke, a gradienst számításal bővítettét pedig világosszürke szín jelöli.

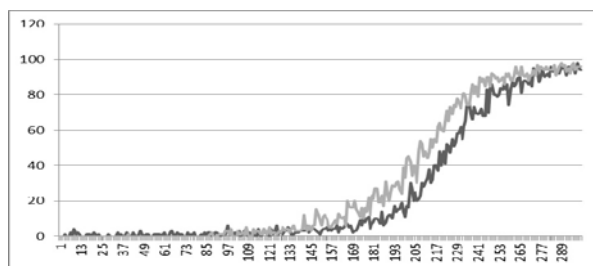


2. ábra

Standard és gradienst számításal kibővített eljárások összehasonlítása a De-Jong függvényen 1000 részecske esetén $f(x,y) = x^2 + y^2$.

Az algoritmusok mindegyik szimuláció esetén azonos paraméterekkel futottak. A 2 és 3. ábrán jól látható, hogy a gradienst használó algoritmus ugyanakkora iterációs szám mellett több esetben talál megoldást, mint a standard algoritmus. A 2. ábra előállításánál használt függvény esetében nincsenek lokális

szélsőértékek, csak egy globális szélsőérték, a 3. ábránál használt függvény



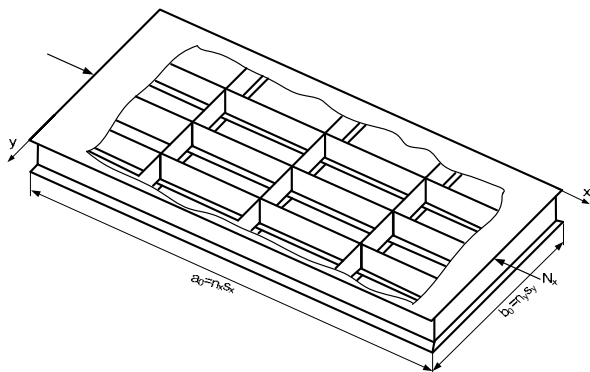
3. ábra Standard és gradiens számítással kibővített eljárások összehasonlítása az Drop Wave függvényen 1000 részecske esetén.

$$f(x,y) = -(1 + \cos(12(x^2 + y^2)^{0.5})) / (0.5(x^2 + y^2) + 2)$$

pedig „szélsőséges eset”, nagyon sok lokális szélsőértékkel rendelkezik. Az ábrákon jól látható, hogy a gradienst használó algoritmus eredményessége a sok lokális szélsőértékkel rendelkező függvény esetében gyengébb, mint azon függvényeknél ahol csak globális szélsőérték létezik.

5. CELLALEMEZ OPTIMÁLIS MÉRETEZÉSE

Elvégeztük egy hegesztett cellalemez optimális méretezését is, ahol a változók a felső és alsó fedőlemez vastagság $x_1=t_1$, $x_2=t_2$, a merevítő magassága $x_3=h$, a merevítők száma x - és y -irányban $x_4=n_x$, $x_5=n_y$. A méretezési feltételek a feszültségre és a stabilitásra vonatkoznak. A célfüggvény, pedig a szerkezet anyag- és gyártási költsége. Az eredmények nagyon biztatóak, összehasonlítva más számításokkal.



4. ábra A nyomásnak kitett cellalemez bontott ábrája

6. ÖSSZEFOGLALÁS

Kidolgozásra került egy új megközelítés PSO algoritmus gyorsítására, mely a mintavételi pontokban véges differencia alapú becsléssel határozza meg a gradienst és ezt használja fel a részecskék

sebességének megváltoztatására, a módszer nem igényel több mintavételezést mint az eredeti eljárás és nem függ a rendszer kezdőállapotától. A módszer hatékonysága függ attól, hogy mekkora egymást követő pozitív gradiens szám esetén változtatjuk a részecske sebességét, valamint a sebességet milyen mértékben változtatjuk. Minden célfüggvény esetén más és más beállítások lehetnek hatékonyak.

7. KÖSZÖNETNYILVÁNÍTÁS

A bemutatott kutató munka a TÁMOP-4.2.1.B-10/2/KONV-2010-0001 jelű projekt részeként az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg, valamint az OTKA T 75678 projekt keretében, illetve támogatásával.

8. IRODALOM

- [1] OSTFELD,A.: Ant colony optimization methods and applications, *InTech Publishers* 2011. ISBN 978-953-307-157-2
- [2] HIMMELBLAU,D.M.: Applied Nonlinear Programming, *McGraw-Hill*, 1972.
- [3] HE,S., PREMPAIN,E., WU,Q.H.: An improved particle swarm optimizer for mechanical design optimization problems, *Engineering Optimization*, Vol. 36: No. 5, pp. 585-605, 2004.
- [4] KENNEDY,J., EBERHART,R.: Particle swarm optimization, *IEEE International Conference on Neural Networks*, Vol. 4, pp. 1942–1948, 1995.
- [5] MOLOGA,M., SMUTNICKI,C.: Test functions for optimization needs, 2005. pp. 1-10. www.bioinformaticslaboratory.nl
- [6] MITCHELL,M.: An introduction to genetic algorithms, *The MIT Press*, 1998.
- [7] EBERHART,R., KENNEDY,J.: A new optimizer using particle swarm theory, *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43, 1995.
- [8] SURIBABU,C.R., NEELAKANTAN,T.R.: Design of water distribution networks using particle swarm optimization, *Urban Water Journal*, Vol. 3: No. 2, pp. 111-120, 2006.
- [9] VAKILIS,S., GADALA,M.S.: Effectiveness and Efficiency of Particle Swarm Optimization Technique in Inverse Heat Conduction Analysis, *Numerical Heat Transfer, Part B: Fundamentals*, Vol. 56: No. 2, pp. 119-141, 2009.
- [10] TÍMÁR,I., HORVÁTH,P., BORBÉLY,T.: Optimierung von profilierten Sandwichbalken, *Stahlbau*, Vol. 72. No. 2. Febr. 2003, pp. 109-113.
- [11] ORBÁN,F.: Minimum cost design of horizontal vessels on saddles. *International Conference on Metal Structures*, University of Miskolc, Hungary, Proceedings pp. 217-221, 2003. Millpress Publishers. Rotterdam