X. FIATAL MŰSZAKIAK TUDOMÁNYOS ÜLÉSSZAKA

Kolozsvár, 2005. március 18-19.

OPTIMÁLÓ ALGORITMUSOK ÖSSZEHAISONLÍTÁSA

Lipták Gábor

Abstract

Optimization means to reach the best solution. By the optimizer methods to solve a technical problem takes much less time. Someone do not need to examine every variation, could use some type of optimization algorithm instead. In this article one specific example is described, which is optimizing plates stiffened on one side. One cost function will be shown, which we need to minimize to get the optimum. Some conditions have to be fulfilled by this optimum. Then three optimizing algorithm will be introduced. The complex algorithm, the differential evolution algorithm, and the particle swarm algorithm. All these are non-derivative, direct optimization methods, which can be used to solve problems having single, continuous objective. We will see the results of them for this particular example, and some conclusion will be drawn.

A példafeladat

1. ábra Bordázott lemez

A példa, melyen az optimáló algoritmusokat tesztelni fogjuk egy egyik oldalán megerősített lemez (1. ábra). A költségfüggvény ill. feltételek levezetése, megértetése nem célja ennek a cikknek, mivel ez a cikk optimáló algoritmusokat hasonlít össze. Így a szükséges ábra, ill. a képletek egy forrásműből származnak [1]. A lemezt az ábrán látható módon megerősítjük azáltal, hogy egymástól s távolságra $(n-1)$ darab hosszirányban félbevágott „I” tartót hegesztünk rá (n az osztások száma). Az I szelvény minden mérete kiszámolható a h magassági értékből (lásd [1] 20-23-as képlet). A szerkezettel kapcsolatban két követelmény van, melyek alkotják az optimálás két feltételét. Az egyik az, hogy a bordák száma nem haladhat meg egy bizonyos értéket, mivel ha a bordák közti távolság minimum 300mm kell (1) hogy legyen a hegeszthetőség miatt.

$$s - b \geq 300$$

(1)

A második feltétel az, hogy a szerkezetben a nyomóerő hatására fellépő kihajlási feszültségnek (σ) a kritikus feszültség alatt kell maradnia (2), vagyis nem szabad maradandó alakváltozást szenvednie.

A képletekben használt konstansok: $b_0 = 8000mm$, $a_0 = 24000mm$, $N = 3 \cdot 10^7 N$, $f_y = 355MPa$, $E = 2,1 \cdot 10^5 MPa$,

$$\sigma = \frac{N}{nA_e} \leq \sigma_{cr} = \frac{f_{y1}}{\sqrt{1+\lambda^4}}; \quad f_{y1} = \frac{f_i}{1.1}; \quad A_e = \frac{h_1 t_w}{2} + b t_f + s_e t; \quad s_e = \min \left(s, 1.9t \sqrt{\frac{E}{f_y}} \right)$$

$$\lambda = \sqrt{\frac{f_{y1}}{\sigma_E}}; \quad \sigma_E = \frac{N_E S}{A_E}; \quad z_G = \frac{1}{A_E} \left[\frac{h_1 t_w}{2} \left(\frac{h_1}{4} + \frac{t}{2} \right) + b t_f \left(\frac{h+t-t_f}{2} \right) \right] \quad (2)$$

$$N_E = \frac{\pi^2 B_x}{a_0^2}; \quad B_x = \frac{EI_y}{s}; \quad I_y = s_e t z_G^2 + \frac{h_1^3 t_w}{96} + \frac{h_1 t_w}{2} \left(\frac{h_1}{4} + \frac{t}{2} + z_G \right)^2 + b t_f \left(\frac{h+t-t_f}{2} - z_G \right)^2$$

A célfüggvény pedig egy költségfüggvény (3), melynek 2 fő komponense az anyagköltség (4) és a hegesztési költség (5). Konstansok: $\theta = 2$, $k_w = 1\$/min$, $a_w = 0.4t_w$, $C = 0.3394 \cdot 10^{-3} min/mm^3$,

$$k_M = 1 \frac{\$}{kg},$$

$$K = K_M + K_W \quad (3)$$

$$K_M = k_M \rho V; \quad V = a_0 b_0 t + (n-1) a_0 \left(\frac{h_1 t_w}{2} + b t_f \right) \quad (4)$$

$$K_W = k_w \left[\theta \sqrt{n \rho V} + 1.3 C a_w^2 (n-1) 2 a_0 \right] \quad (5)$$

Ismeretlenek tehát az n osztásközök száma, $4mm \leq t \leq 40mm$ a lemez vastagsága, és $152.4mm \leq h \leq 910.4mm$ az I tartó magassága.

Közös tulajdonságok

Mindhárom eljárásban szükség van arra, hogy véletlenszerűen generáljunk hozzájuk kezdőpontokat, melyek koordinátái az optimálási változók. A kiinduló pontok kordinátái egyszerű véletlenszámok lesznek a változók alsó és felső korlátja között. A módszerek iterációkból állnak, melyek futásának a *kilépési feltétel* teljesülése, vagy egy *iterációszám korlát* vethet véget. Az általam alkalmazott kilépési feltétel *relatív* [2] volt. Most pedig röviden tekintsük át a három módszert!

Komplex algoritmus

A módszer részletes leírása megtalálható a [2] forrásműben. A komplex eljárás iterációkból áll. A véletlenszerű kezdőpontok között kell lennie minimum egynek, amely minden feltételnek megfelel. Ezután minden iterációban megkeressük a legrosszabb pontot. Kiszámítjuk hozzá az ún. centroidot,

melyet úgy kapunk meg, hogy minden dimenziója a többi pont dimenzió értékeinek átlaga. Tükrözzük a legrosszabb pontot erre a centroidra úgy, hogy meg is nyújtjuk kissé távolságukat. Ezáltal kapunk egy új pontot. Ha ez javított a ponton, akkor elfogadjuk, mint javulást. Ha nem, akkor tologatjuk a centroid felé felezve kettőjük távolságát egészen addig, amíg javulást nem tapasztalunk. Majd haladunk tovább a következő legrosszabb ponttal. Ez az eredeti algoritmus lényege.

Differenciális evolúció algoritmus

A módszer részletes leírása található a [3] forrásműben. Evolúciós algoritmus, melynek lényege, hogy egy iteráció során egy M pontot, vagyis egyedet tartalmazó generációból egy újabb generációt hoz létre. Egy iteráció a következőképpen alakul. Végigmegyünk az összes egyeden (a képletben minden egyed egy X sormátrix), és képzünk mindegyikhez egy új egyedet a következőképpen:

$$x_{i,G+1} = x_{r_1G} + F(x_{r_2G} - x_{r_3G}) \quad i = 1, 2, \dots, M \quad (6)$$

Az első index itt is a pont sorszámát jelöli. A G , illetve $G+1$ index pedig arra utal, hogy a $G+1$ -dik iteráció elemét a G -dik iteráció egyedei alapján lehet előállítani. Az r_1, r_2, r_3 index pedig arra utal, hogy az $1 \dots M$ indexű egyedek közül találmra kiválasztunk 3 különbözőt úgy, hogy az i -dik elem ne legyen azok között. Megjelenik a módszer egyik paramétere az F mutációs konstans. Értéke általában 0 és 1 közé esik. Amikor egy az új egyedek közül megvan, akkor keresztezzük (egyes változó értékét véletlenszerűen az egyik egyedből vesszük át, másokét a másiktól) az előző generáció azonos sorszámú egyedével, és ha az így kapott egyed jobb, mint az előző generáció neves képviselője, akkor az új generáció tagja lehet. Ha nem akkor marad a régi egyed. Az iterációk addig folytatódnak, amíg a kilépési feltétel nem teljesül.

Particle swarm algoritmus

A módszer részletes leírása található a [4] forrásműben. Evolúciós algoritmus, melynek ideológiája az egymással kommunikáló, saját eddigi útjukra bizonyos mértékig emlékező madarakon alapszik. Van egy adott számú madár (vagy pont) a tervezési térben. Mindegyik „emlékszik” az eddigi legjobb pozíciójára, valamint mindegyik tudja, hogy az egész csapatnak melyik volt az eddigi legjobb pozíciója. Minden egyes iteráció során a madarak sebességét, majd a pozícióját kiszámítjuk a következő képlet alapján (7) (V_{id} sebesség; w, c_1, c_2 súly konstansok; x_{id} pozíció; p_{id} madár eddigi legjobb helye; p_{gd} csapat eddigi legjobb helye):

$$V_{id} = wV_{id} + c_1 \cdot rand() \cdot (p_{id} - x_{id}) + c_2 \cdot rand() \cdot (p_{gd} - x_{id}); \quad x_{id} = x_{id} + V_{id} \quad (7)$$

Minden iterációban egyszerűen kiszámoljuk a madarak új helyét addig, amíg a kilépési feltétel nem teljesül.

Teszteredmények

A módszerek tesztelése során mindhárom algoritmus egy-egy saját implementációját használtam. Jellegzetességük, hogy JAVA nyelven készültek, így bár kissé lassabbak, mint a natív társaik, viszont gyakorlatilag bármilyen platformon futtathatók. A tesztek során az $n=8$ érték megadásával az ismeretlenek számát 2-re csökkentettem az egyszerűség érdekében. Legtöbbször mindhárom eljárás a $h=910.4mm; t=17,37$ optimumot adta, ahol a költség 52933\$ volt. Mindhárom módszernél 30 pontból állt a kereső halmaz, és a táblázat 10-10 futás eredményének átlagából áll. A következő táblázat néhány adatot közöl a tapasztalataim alapján (Biztonság azt jelenti, hogy a 10 futásból hányszor jutott el az optimumba a módszer):

	Iterációszám	Futási idő	Célfüggvény kiértékelések	Biztonság
Komplex	2671	240 msec	2729	70%
Diff. evolúció	1405	493 msec	35653	100%
Madár csapat	400	168 msec	12061	100%

A táblázat értékeiből látható, hogy a madár csapat módszer reális kompromisszum a célfüggvénykiértékelések száma és a biztonság között. A komplex algoritmus használata akkor kifizetődő, ha nagyon bonyolult a célfüggvény. A differenciális evolúció előnye pedig az egyszerű implementálhatóság, használhatóság.

Irodalomjegyzék

- [1] Farkas J., Jármái K. : **Optimum design and cost comparison of a welded plate stiffened on one side and a cellural plate both loaded by uniaxial compression** (kiadás alatt álló cikk)
- [2] Dr. Jármái K., Dr. Iványi M. : **Gazdaságos fémszerkezetek analízise és tervezése**. Műegyetemi kiadó, 2001. ISBN 963 420 674 3
- [3] Rainer Storn: **On the Usage of Differential Evolution for Function Optimization**. 1996 Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS 1996), Berkeley, pp. 519–523. IEEE, New York, NY, USA.
- [4] Xiaohui Hu and Russell Eberhart. **Solving Constrained Nonlinear Optimization Problems with Particle Swarm Optimization**. In *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002)*, volume 5. Orlando, USA, IIS, July 2002.

Köszönetnyilvánítás: A kutatás az OTKA 38058 projekt támogatásával történt.

Lipták Gábor, doktorandusz

IND Kft. (www.indweb.hu). 3530 Miskolc Széchenyi u. 70., Magyarország

Tel: +36-20-5199146, **E-mail:** gabor.liptak@indweb.hu