

TDK DOLGOZAT

Barcsák Csaba

Intézmény: Miskolci Egyetem

Kar: Gépészmérnöki és Informatikai Kar

Tanszék: Anyagmozgatási és Logisztikai Tanszék

Szerző: Barcsák Csaba

Témavezető: Prof. Dr. Jármái Károly, egyetemi tanár

Dolgozat címe:

PSO algoritmus bővítése véges differencia alapú gradiens becsléssel

Köszönetnyilvánítás

Köszönetemet fejezem ki témavezetőmnek, Prof. Dr. Jármai Károlynak, aki sokat segített az optimáló módszerek elméleti hátterének és gyakorlati alkalmazásainak megértésében, és figyelemmel kísérte munkámat.

A bemutatott TDK 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 Országos Tudományos Kutatási Alap OTKA T 75678 támogatásával.

Tartalomjegyzék

1. Bevezetés	1
2. PSO algoritmusok	2
2.1 Standard PSO algoritmus	2
2.2 PSO algoritmus bővítése gradiens becsléssel	3
2.3 Szimulációs eredmények	5
3. Gradiens alapú PSO módszer alkalmazása bordázott lemezek költségének optimalására	7
3.1 A bordázott lemezek matematikai modellje	7
3.2 A költségfüggvény	10
3.3 Az optimalás eredménye	11
4. Konklúzió	14
5. Függelék	16

1. fejezet

Bevezetés

Az optimálási 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 számításigényessé válnak. A heurisztikus optimálási módszerek nem rendelkeznek a deriváltat használó módszerek előbb említett hátrányaival, viszonylag könnyedén implementálhatóak, emiatt nagy népszerűsége tette szert az optimálással foglalkozók körében. Ezen algoritmusok közé tartozik a Hangya kolónia, az Ant colony algoritmus, mely hangyák viselkedését szimulálja, a Genetikus algoritmusok, melyek evolúciós folyamat modellezésével oldják meg a problémát, a Differenciális evolúció módszere, a Virtuális immunrendszer módszere, valamint ide tartozik a Részecskecsoport módszer, az ún. PSO (Particle Swarm Optimization) algoritmus is.

A dolgozatban új megközelítés kerül bemutatásra a PSO algoritmus gyorsítására, mely véges differencia alapú gradiens becslést használ, hatékonysága nem függ a rendszer kezdőállapotától, és nem igényel több függvénykiértékelést, mint az eredeti algoritmus. Az új módszer segítségével több optimálási tesztprobléma is megoldásra került, a tesztek során láthatóvá vált, hogy az új technika megnöveli az algoritmus konvergencia sebességét, ezáltal gyorsabban találja meg az optimális megoldást. A módszer segítségével megoldásra került továbbá egy műszaki optimálási feladat, melynek során bordázott lemezek kerültek optimálásra előállítási költség szempontjából. Az új algoritmus ezen probléma megoldásánál is jobban teljesített, mint az eredeti algoritmus.

A kutatói munka során kifejlesztésre került egy Java alkalmazás, mely összefoglalja az elért eredményeket. Az alkalmazás képes a PSO algoritmus változatainak összehasonlítására, egyes iterációs lépéseinek interaktív grafikus megjelenítésére kétdimenziós tesztproblémák esetén, lehetővé téve ezzel a PSO változat futásának vizualizációját, továbbá grafikus felületen történő paraméterezését.

2. fejezet

PSO algoritmusok

Kennedy és Eberhart 1995-ben mutatta be a PSO (Particle Swarm Optimization) algoritmust[1, 2], mellyel az eredeti céljuk az volt, hogy madarak csapaton belüli szociális viselkedését szimulálják és vizualizálják. A kutatásaik során felfedezték, hogy ez a módszer optimálási feladatok megoldására is hatékonyan alkalmazható. A PSO első változata csak folytonos nemlineáris optimálási feladatokat volt képes megoldani. Az évek során az algoritmusnak nagyon sok változata jelent meg melyek optimálási problémák széles skáláját képesek megoldani. Ezen algoritmusok az egyszerűségük és hatékonyságuk miatt váltak széles körben elterjedté a mérnöki gyakorlatban[3, 4, 5].

2.1. Standard PSO algoritmus

A következőkben a standard algoritmus kerül ismertetésre. Az algoritmus első lépésben ún. részecskéket generál. Minden részecske rendelkezik egy x pozíció, 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.

$$v_i^{k+1} = v_i^k + c_1 r_1 (pbest_i - x_i^k) + c_2 r_2 (gbest_i - x_i^k) \quad (2.1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} t \quad (2.2)$$

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 2.1. ábra mutatja:



2.1. ábra. Standard PSO algoritmus folyamatábrája

A 2.1. ábrán látható folyamatban a részecskék inicializálása a helyvektoraik 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 optimáló eljárásokba, kevésbé érzékeny célfüggvényre, beállításához kevesebb paraméter szükséges, mint más heurisztikus eljárásoknál. A hátránya, hogy nincs mögötte mély matematikai háttér.

2.2. PSO algoritmus bővítése gradiens becsléssel

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. Ez a megoldás véletlenszerűen kiválaszt részecskéket, és ezen részecskék sebességét nem a 2.1. 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 értékin 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 differenciálható függvény Taylor-sorából indul ki, amely egy egydimenziós függvény $f(x)$ esetében a következőképp írható fel:

$$f(x_0 + h) \approx f(x_0) + \frac{f'(x_0)}{1!}h + \frac{f''(x_0)}{2!}h^2 + \dots = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}h^n \quad (2.3)$$

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

$$f(x_0 + h) \approx f(x_0) + \frac{f'(x_0)}{1!}h \quad (2.4)$$

Kifejezve a deriváltat a következő formula adódik:

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h} \quad (2.5)$$

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

$$f'(x_0) \approx \frac{f(x_0) - f(x_0 - h)}{h} \quad (2.6)$$

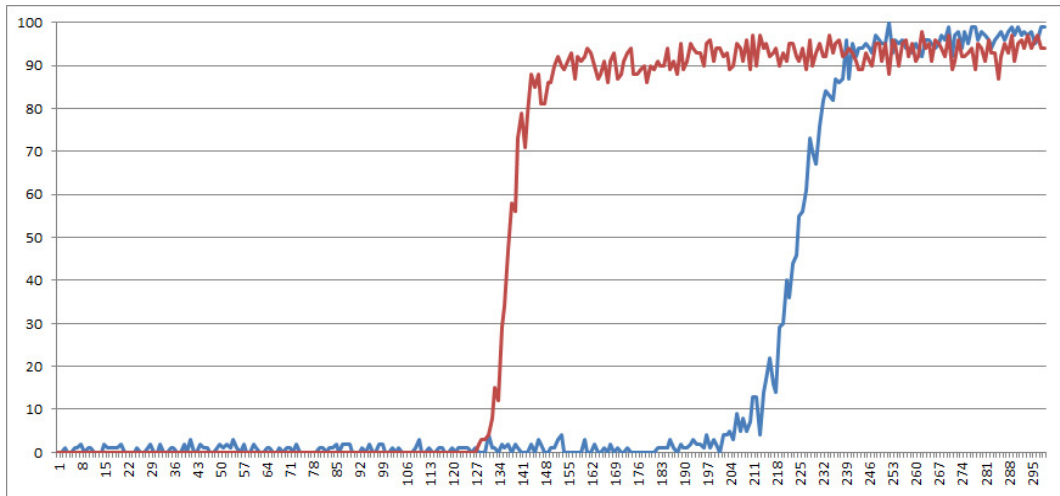
Ezt a képletet az irodalom hátravett differencia (backward difference) becslés néven említi. Ezek a megoldások egyszerűek, gyorsan kiszámolhatóak, hátrányuk viszont az, hogy kevésbé pontosak. Léteznek összetettebb gradienst becslő megoldások az irodalomban, de azok számításigényesebbek az előbbieken 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 gradiens, 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 szélsőértékig. A módszer eredményességét a következő fejezetben bemutatott szimulációk igazolják.

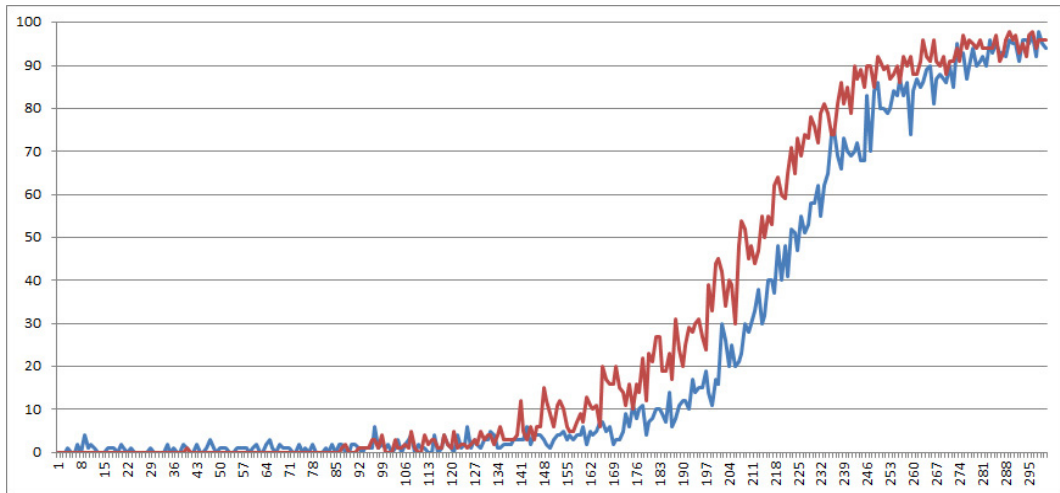
2.3. Szimulációs eredmények

A szimulációs eredmények előállításához egy Java alkalmazás került kifejlesztésre mely tartalmazza a standard valamint a gradienst használó megoldást, tizenkét feltétel nélküli[6] és három feltételes[7] kétdimenziós optimálási tesztfüggvényt. A tizenkét feltétel nélküli tesztfüggvényből öt „szélsőséges” eset. Ez azt jelenti, hogy sok a lokális szélsőérték, és ez nehezen megoldhatóvá teszi a problémát.

Mindkét algoritmust egy adott iterációs lépésig adott tesztfüggvényre százszor futtattunk. Az 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 kék, a gradiens számítással bővítettét pedig piros szín jelöli.



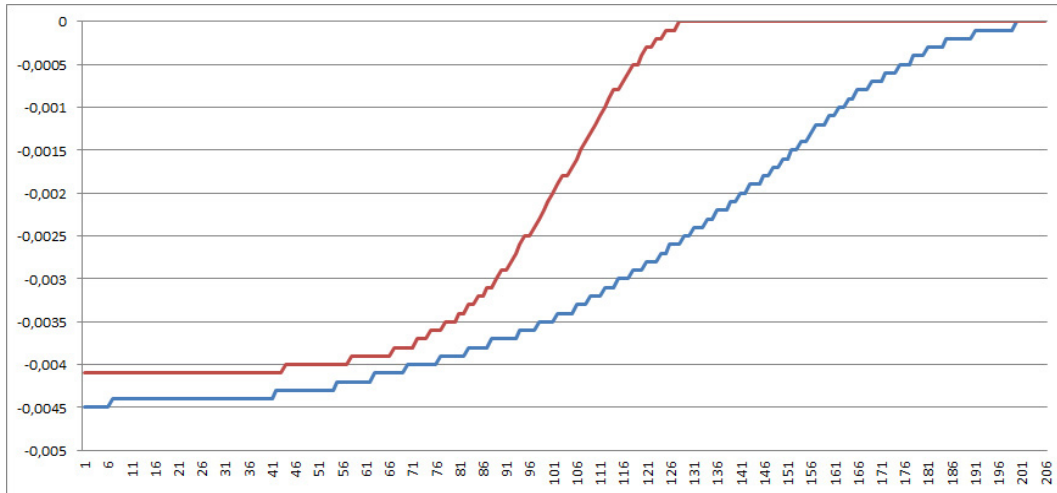
(a)



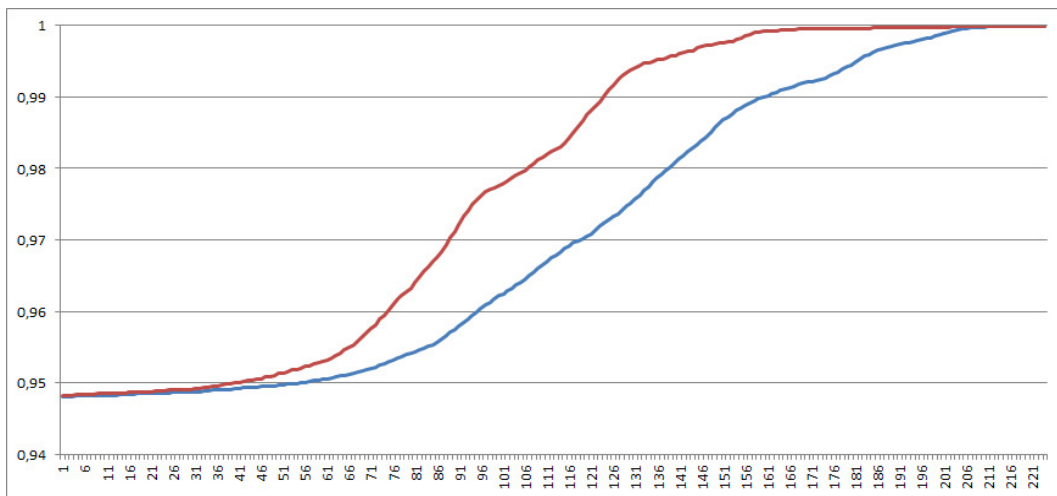
(b)

2.2. ábra. Az (a) képen a De Jong függvényről $f(x, y) = -(x^2 + y^2)$, a (b) képen a Drop Wave függvényről $f(x, y) = \frac{1 + \cos(12\sqrt{x^2 + y^2})}{\frac{1}{2}(x^2 + y^2) + 2}$ látható az előbbiekben említett teszt eredménye ezer részecske esetén. A függőleges tengely a talált megoldások számát, a vízszintes az iterációs számot jelöli. A képeken jól látható, hogy a gradiens alapú módszer ugyanazon iterációs szám mellett több esetben talált megoldást.

Az előbbieken említett tesztelési módszer eredménye függ az algoritmus kilépési feltételétől. Az irodalomban nincs egységes megállapodás arra, hogy milyen kilépési feltételt érdemes használni. A tesztelés során akkor állítottuk le az algoritmust, amikor a globális legjobb érték kétszáz iterációs lépésen keresztül nem változott meg. Egy másik lehetőség az algoritmus tesztelésére, ha a globális legjobb érték változását figyeljük meg az iterációk függvényében. Ekkor az eredmény már nem függ a kilépési feltételtől.



(a)



(b)

2.3. ábra. A képeken a globális legjobb érték változása figyelhető meg az iteráció függvényében. A függőleges tengely a globális legjobb értéket, a vízszintes az iterációs számot jelöli. Az (a) képen a De Jong, a (b) képen a Drop Wave függvényre teszteltük a módszereket. A diagramokon száz futás számtani átlagából kapott eredmény látható. Az átlagolásra amiatt volt szükség, hogy elnyomjuk a kiugró értékeket, mivel a folyamatban a véletlen is szerepet játszik.

A globális legjobb iteráció függvényében történő változása kiszámításra került a tizenkét feltétel nélküli tesztfüggvény esetén. A gradiens alapú módszer kilenc esetben jobb, két esetben hasonló, és csak egy esetben ért el rosszabb eredményt mint a standard algoritmus. Ezen tesztek eredményei a függelékben találhatók.

3. fejezet

Gradiens alapú PSO módszer alkalmazása bordázott lemezek költségének optimalására

Bordázott lemezek két alaplemezből, és az alaplemezek közé hegesztett merevítő rácsból állnak, melyeket felhasználnak épületek, hidak, hajók és más gépek megépítésénél. A bordázott lemezek előnye a csak az egyik oldalon merevített lemezekkel szemben a nagyobb merevség, rozsdá elleni jobb védelem. A merevítő rács fél I szelvényekből áll, és merevvé teszi a szerkezetet. Egytengelyű nyomás esetén a horpadás a Huber egyenletből került levezetésre. A költségfüggvény tartalmazza az anyagköltséget, az összeszerelés és hegesztés költségeit. Az ismeretlen változók az alaplemezek vastagságai, merevítők magassága, valamint a merevítők száma mindkét tengely mentén.

3.1. A bordázott lemezek matematikai modellje

A Huber egyenlet ortotróp lemezeknél a kihajlásra $w(x, y)$ a következőképp írható fel egytengelyű nyomás esetén:

$$B_x \frac{\partial^4 w}{\partial x^4} + 2H \frac{\partial^4 w}{\partial x^2 \partial y^2} + B_y \frac{\partial^4 w}{\partial y^4} + N_x \frac{\partial^2 w}{\partial x^2} = 0, \quad (3.1)$$

ahol

$$H = B_{xy} + B_{yx} + \frac{\nu}{2}(B_x + B_y) \quad (3.2)$$

a csavarási merevség, $\nu = 0.3$ a Poisson szám. A csavarási merevségek és a kihajlás a következők:

$$B_x = \frac{E_1 I_y}{a_y}; B_y = \frac{E_1 I_x}{a_x}; E_1 = \frac{E}{1 - \nu^2} \quad (3.3)$$

Ahol $E = 2.1 * 10^5 MPa$ az rugalmassági modulus.

$$B_{xy} = \frac{GI_y}{a_y}; B_{yx} = \frac{GI_x}{a_x}; G = \frac{E}{2(1 + \nu)} \quad (3.4)$$

Ahol G a nyírási modulus.

$$H = B_{xy} + B_{yx} + \frac{\nu}{2}(B_x + B_y) = \frac{E_1}{2} \left(\frac{I_y}{a_y} + \frac{I_x}{a_x} \right) \quad (3.5)$$

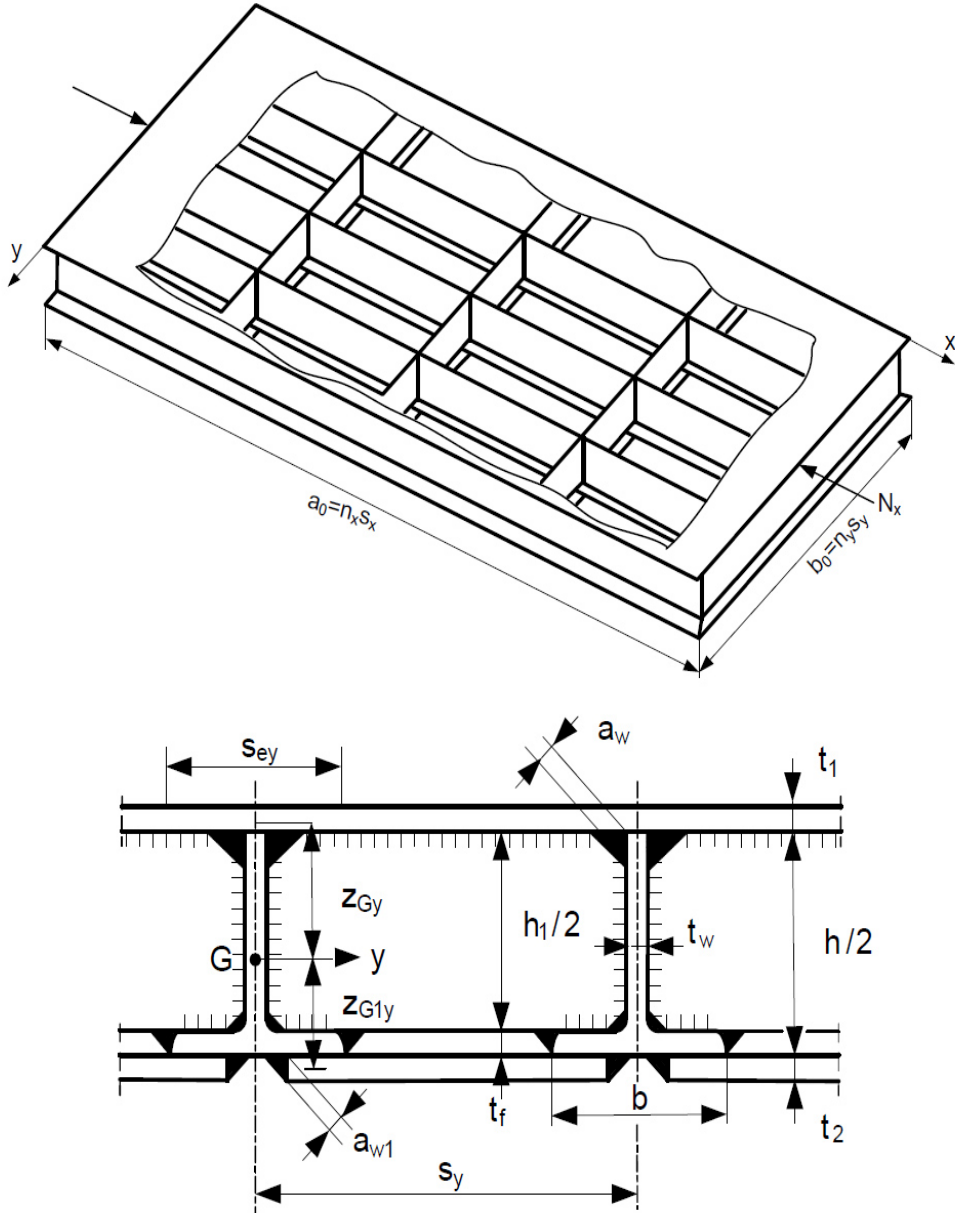
A 3.1 egyenlet megoldása[8]:

$$N_E = \frac{\pi^2}{b_0^2} \left[B_x \left(\frac{b_0}{a_0} \right)^2 + 2H + B_y \left(\frac{a_0}{b_0} \right)^2 \right] \quad (3.6)$$

A kihajlási feltétel a következőképp írható fel[9]:

$$\frac{N_x}{n_y A_{ey}} \leq \sigma_{cr} = \frac{f_y}{\sqrt{1 + \lambda^4}}, \lambda = \sqrt{\frac{f_y}{\sigma_E}}, \sigma_E = \frac{N_E s_y}{A_{ey}} \quad (3.7)$$

A klasszikus kritikus horpadási feszültség σ_E az előbb leírt összefüggéssel számolható



3.1. ábra. Ortogónálisan merevített bordázott lemez és keresztmetszete.

$$A_{ey} = \frac{h_1 t_w}{2} + b t_f + s_{ey1} t_1 + s_{ey2} t_2 \quad (3.8)$$

$$A_{ex} = \frac{h_1 t_w}{2} + b t_f + s_{ex1} t_1 + s_{ex2} t_2 \quad (3.9)$$

$$h_1 = h - 2t_f \quad (3.10)$$

$$s_y = \frac{b_0}{n_y}, s_x = \frac{a_0}{n_x} \quad (3.11)$$

Ahol n_x és n_y a merevítők közötti távolság x és y irányban, melyek a 3.1-es ábrán láthatók. A lemezrészek effektív szélessége [10] alapján a következőképp számítható ki:

$$s_{ey1} = \rho_{y1} s_y, s_{ey2} = \rho_{y2} s_y, s_{ex1} = \rho_{x1} s_x, s_{ex2} = \rho_{x2} s_x \quad (3.12)$$

ahol

$$\rho_{y1} = \frac{\lambda_{py1} - 0.22}{\lambda_{py1}^2} \quad ha \quad \lambda_{py1} = \frac{s_y}{56.8 \varepsilon t_1} \geq 0.673, \varepsilon = \sqrt{\frac{235}{f_y}} \quad (3.13)$$

$$\rho_{y1} = 1 \quad ha \quad \lambda_{py1} < 0.673 \quad (3.14)$$

$$\rho_{y2} = \frac{\lambda_{py2} - 0.22}{\lambda_{py2}^2} \quad ha \quad \lambda_{py2} = \frac{s_y}{56.8 \varepsilon t_1} \geq 0.673 \quad (3.15)$$

$$\rho_{y2} = 1 \quad ha \quad \lambda_{py2} < 0.673 \quad (3.16)$$

$$\rho_{x1} = \frac{\lambda_{px1} - 0.22}{\lambda_{px1}^2} \quad ha \quad \lambda_{px1} = \frac{s_x}{56.8 \varepsilon t_2} \geq 0.673 \quad (3.17)$$

$$\rho_{x1} = 1 \quad ha \quad \lambda_{px1} < 0.673 \quad (3.18)$$

$$\rho_{x2} = \frac{\lambda_{px2} - 0.22}{\lambda_{px2}^2} \quad ha \quad \lambda_{px2} = \frac{s_x}{56.8 \varepsilon t_2} \geq 0.673 \quad (3.19)$$

$$\rho_{x2} = 1 \quad ha \quad \lambda_{px2} < 0.673 \quad (3.20)$$

A súlypontok távolsága a következőképp írható fel:

$$z_{Gy} = \frac{1}{A_{ey}} \left[\frac{h_1 t_w}{2} \left(\frac{h_1}{4} + \frac{t_1}{2} \right) + b t_f \left(\frac{h_1 + t_f + t_1}{2} \right) + s_{ey2} t_2 \left(\frac{h_1 + t_f + t_1 + t_2}{2} \right) \right] \quad (3.21)$$

$$z_{Gx} = \frac{1}{A_{ex}} \left[\frac{h_1 t_w}{2} \left(\frac{h_1}{4} + \frac{t_1}{2} \right) + b t_f \left(\frac{h_1 + t_f + t_1}{2} \right) + s_{ex2} t_2 \left(\frac{h_1 + t_f + t_1 + t_2}{2} \right) \right] \quad (3.22)$$

Az inercianyomatékok a következők:

$$I_y = s_{ey1}t_1z_{Gy}^2 + \frac{h_1^3t_w}{96} + \frac{h_1t_w}{2} \left(\frac{h_1}{4} + \frac{t_1}{2} - z_{Gy} \right)^2 + I_{y1} \quad (3.23)$$

$$I_{y1} = bt_f \left(\frac{h_1 + t_f + t_1}{2} - z_{Gy} \right)^2 + s_{ey2}t_2 \left(\frac{h_1 + t_f + t_1 + t_2}{2} - z_{Gy} \right)^2 \quad (3.24)$$

$$I_x = s_{ex1}t_1z_{Gx}^2 + \frac{h_1^3t_w}{96} + \frac{h_1t_w}{2} \left(\frac{h_1}{4} + \frac{t_1}{2} - z_{Gx} \right)^2 + I_{x1} \quad (3.25)$$

$$I_{x1} = bt_f \left(\frac{h_1 + t_f + t_1}{2} - z_{Gx} \right)^2 + s_{ex2}t_2 \left(\frac{h_1 + t_f + t_1 + t_2}{2} - z_{Gx} \right)^2 \quad (3.26)$$

A következő gyártási feltétel teszi lehetővé a hegeszthetőséget:

$$s_{x,y} - b \geq 300mm \quad (3.27)$$

Az ismeretlenek: $x_1 = t_1$ a felső fedőlemez vastagsága, $x_2 = t_2$ az alsó fedőlemez vastagsága, $x_3 = h$ az I szelvény magassága, $x_4 = n_x$ a merevítők száma x irányba, és $x_5 = n_y$ a merevítők száma y irányba.

3.2. A költségfüggvény

Az ár tartalmazza az anyagköltséget K_M , és a hegesztési költséget K_W .

$$K_M = k_M \rho V \quad (3.28)$$

Ahol $k_M = 1.0\$/kg$, $\rho = 7.85 \times 10^{-6}kg/mm^3$, V a térfogat. Az általános összefüggés a hegesztés költségére a következő[11, 12]:

$$K_w = k_w \left(C_1 \Theta \sqrt{\kappa \rho V} + 1.3 \sum_i C_{wi} a_{wi}^n C_{pi} L_{wi} \right) \quad (3.29)$$

Ahol $k_w [$/min]$ a fajlagos hegesztési paraméter, C_1 az összeszerelési faktor amely általában $C_1 = 1 \text{ min}/kg^{0.5}$, Θ az összeszerelés bonyolultsági foka, az első tag az összeszerelési időt számítja ki, ahol κ az összeszerelni kívánt szerkezeti elemek száma, ρV pedig a struktúra tömege. A második tag a hegesztési időt becsüli, C_w és n konstansok melyeket a hegesztési technológia és típus határoz meg, C_p a hegesztési pozíció faktora, L_w a hegesztés hossza, az 1.3-as szorzó segítségével pedig a további hegesztési időt vesszük figyelembe (sorjázás, salakolás, elektróda csere).

A felső alaplemez hegesztéséhez fedőporos ívhegesztés (tompá varrattal) esetén a hegesztés hossza $L_{w1} = 3(a_0 + b_0)$, a struktúra térfogata $V_1 = a_0 b_0 t_1$, $\Theta_1 = 2$, az elemek száma $\kappa_1 = 16$, $k_W = 1\$/min$. Ha $t_1 \geq 15mm$ akkor $C_W a_W^n = 0.1346 \times 10^{-3} t_1^2$, ha $t_1 < 15mm$ akkor $C_W a_W^n = 0.1033 \times 10^{-3} t_1^{1.94}$.

$$K_{W1} = k_W \left(\Theta_1 \sqrt{\kappa_1 \rho V_1} + 1.3 C_W a_W^n L_{W1} \right) \quad (3.30)$$

A hosszirányú merevítők felső alaplemezhez történő hozzáhegesztése fedőporos ívhegesztéssel (sarokvarrattal) történik. A változók a következők:

$$\begin{aligned} L_{W2} &= 2a_0(n_y + 1), \kappa_2 = n_y + 2, V_2 = V_1 + a_0(bt_f + h_1t_w/2)(n_y + 1), \\ a_W &= 0.4t_w, \Theta_2 = 3 \end{aligned} \quad (3.31)$$

A költség:

$$K_{W2} = k_W \left(\Theta_2 \sqrt{\kappa_2 \rho V_2} + 1.3 \times 0.2349 \times 10^{-3} a_W^2 L_{W2} \right) \quad (3.32)$$

A keresztirányú merevítők felső alaplemezhez és longitudinális merevítőkhöz történő hegesztéséhez CO_2 védőgáz ívhegesztéssel (tompá varrattal):

$$\begin{aligned} V_3 &= V_2 + b_0(bt_f + h_1t_w/2)(n_x + 1), \kappa_3 = 1 + n_y(n_x + 1), \\ L_{W3} &= (n_x + 1)(2b_0 + n_y(h_1 + b)) \end{aligned} \quad (3.33)$$

$$K_{W3} = k_W \left(\Theta_2 \sqrt{\kappa_3 \rho V_3} + 1.3 \times 0.3394 \times 10^{-3} a_W^2 L_{W3} + 1.3 C_{Wf} t_f^n L_{Wf} \right) \quad (3.34)$$

Ha $t_f \geq 15mm$ akkor $C_{Wf} t_f^n = 0.1496 \times 10^{-3} t_f^{1.9029}$, ha $t_f < 15mm$ akkor $C_{Wf} t_f^n = 0.1939 \times 10^{-3} t_f^2$.

$$L_{Wf} = 2bn_y(n_x + 1) \quad (3.35)$$

Az alsó alaplemez a merevítők övlemezéhez hegesztéséhez fedőporos ívhegesztést (sarokvarrattal) alkalmazunk:

$$V_4 = V_3 + a_0 b_0 t_2, \kappa_4 = 1 + n_x n_y, L_{W4} = 2(a_0 n_y + b_0 n_x), a_{W1} = 0.7t_2 \quad (3.36)$$

$$K_{W4} = k_W \left(\Theta_1 \sqrt{\kappa_4 \rho V_4} + 1.3 \times 0.2349 \times 10^{-3} a_{W1}^2 L_{W4} \right) \quad (3.37)$$

A teljes költség:

$$K = K_M + K_{W1} + K_{W2} + K_{W3} + K_{W4} \quad (3.38)$$

3.3. Az optimálás eredménye

Az ismeretlen változók: h, t_1, t_2, n_x, n_y . A konstansok a következők: $b_0 = 8000mm$, $a_0 = 24000mm$, $N = 3 \times 10^7[N]$, $f_y = 355MPa$, $E = 2.5 \times 10^5 MPa$. A változók a következő intervallumokon vehetnek fel értékeket: $t = 4mm - 40mm$, $h = 152.4mm - 910.4mm$, n maximális értéke pedig a 3.27-es egyenletből származtatható.

$$n_{max} = \frac{b_0}{b + 300} \quad (3.39)$$

Az n_{max} értékeket a következő táblázat mutatja:

h	353.4	403.2	454.6	533.1	607.6	683.5	762.2	840.7	910.4	1008.1
b	126.0	142.2	152.9	209.3	228.2	253.7	266.7	292.4	304.1	302.1
n_{max}	18	18	17	15	15	14	14	13	13	13

3.1. táblázat. n_{max} értékek

Az UB profil méreteinek meghatározására közelítő formula került alkalmazásra[12], mely a magasság h függvénye. A peremvastagságot a következőképp számolhatjuk: $t_f(y = t_f; x = h)$, ahol

$$y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 \quad (3.40)$$

A hálóvastagságot a következőképp számolhatjuk: $t_w(y = t_w; x = h)$, ahol

$$y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 \quad (3.41)$$

A peremszélességet b , a következőképp számolhatjuk:

$$y = a + bx + c/x + dx^2 + e/x^2 + fx^3 + g/x^3 + hx^4 + i/x^4 + jx^5 + k/x^5 \quad (3.42)$$

A konstansok a következő táblázatban találhatók:

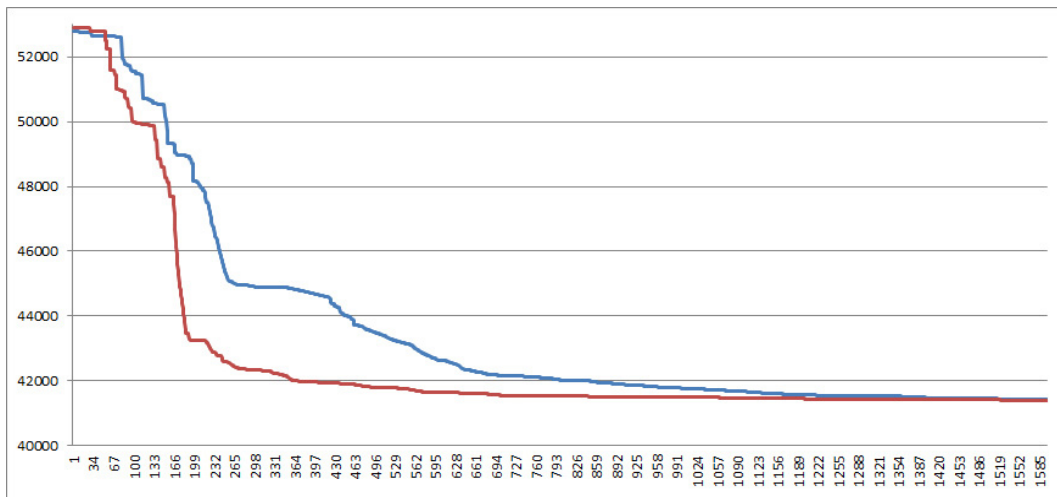
	t_f	t_w	b
a	-26.93815960004096	4.598131596507252	-1108926.658794802
b	0.7030053163805572	-0.1667245080692302	2054.96457373585
c	-0.00569333794408951	0.002662252638593643	394347552.4221416
d	$2.383106250400329 \times 10^{-5}$	$-1.662919423768273 \times 10^{-5}$	-2.475920494568994
e	$-5.605511588090933 \times 10^{-8}$	$5.42570607199179 \times 10^{-8}$	-91315532919.66857
f	$7.662794270183799 \times 10^{-11}$	$-1.003562930723944 \times 10^{-10}$	0.001858445891156483
g	$-5.902409057606285 \times 10^{-14}$	$1.063362616433473 \times 10^{-13}$	13189053888762.85
h	$2.267417890058806 \times 10^{-17}$	$-6.028516559742138 \times 10^{-17}$	$-7.856977790442618 \times 10^{-7}$
i	$-2.999371273581411 \times 10^{-21}$	$1.419727612597333 \times 10^{-20}$	-1073670362507492
j			$1.422535840934241 \times 10^{-10}$
k			$3.744384150518803 \times 10^{16}$

3.2. táblázat. konstansok értékei

A következő táblázat az optimalálás eredményét mutatja. A diszkrét értékek megke-
resése a folytonos eredmények megtalálása után történt.

módszer	$x_1 = t_1$	$x_2 = t_2$	$x_3 = h$	$x_4 = n_x$	$x_5 = n_y$	ár[\$]	részecskék száma
PSO	8	5	403.2	2	14	42308.188	1000
PSO	9	5	403.2	2	14	44364.36	10000
GPSO	5	5	454.6	2	15	41679.973	1000
GPSO	7	5	403.2	2	15	41442.72	10000

3.3. táblázat. Optimalálás eredménye. A PSO módszer a Standard PSO, míg a GPSO módszer a gradienssel bővített eljárást jelöli. Az algoritmusok leállási feltétele minden esetben az volt, hogy a global best ne változzon ötszáz iteráción át.



3.2. ábra. Global best változása az iteráció függvényében a grafikon tíz futás átlagából került kiszámításra. A képen látható, hogy a gradienst használó változat gyorsabban közelít a minimum felé, mint a standard eljárás.

4. fejezet

Konklúzió

Új megoldás került bemutatásra a PSO algoritmus gyorsítására, mely véges differencia alapú gradiens becslést használ a részecskék sebességének beállítására, hatékonysága nem függ a rendszer kezdőállapotától, és nem igényel több függvénykiértékelést, mint az eredeti algoritmus. Az új módszer segítségével több optimalizációs tesztprobléma is megoldásra került, a tesztek során láthatóvá vált, hogy az új technika megnöveli az algoritmus konvergencia sebességét, ezáltal gyorsabban találja meg az optimális megoldást. A módszer segítségével megoldásra került továbbá egy műszaki optimalizációs feladat, melynek során bordázott lemezek kerültek optimalizálásra előállítási költség szempontjából. Az új algoritmus ezen probléma megoldásánál is jobban teljesített, mint az eredeti algoritmus.

Az új módszer paramétereinek (milyen mértékben gyorsuljanak a részecskék, és mekkora egymást követő pozitív gradiensszám esetén) megfelelő beállítása további kutatásokat igényel. A cél az, hogy az algoritmus felhasználói beavatkozás nélkül határozza meg ezen paramétereket a rendelkezésére álló információkból. A továbbfejlesztés lehetséges irányai a következők: többcélűfüggvényes optimalizálás, diszkrét érték meghatározás, crazy bird aktivitás megnövelése konvergencia lassulása esetén.

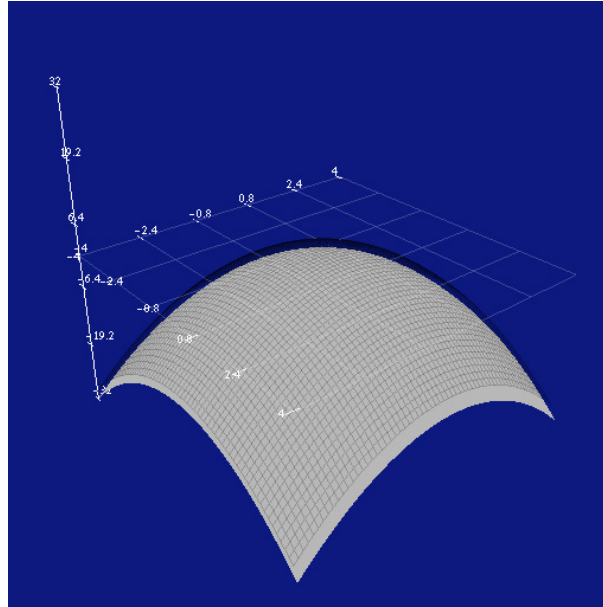
Irodalomjegyzék

- [1] J. Kennedy, R. Eberhart: *Particle swarm optimization*, IEEE International Conference on Neural Networks, Vol. 4, pp. 1942–1948, 1995.
- [2] J. Kennedy, R. Eberhart: *A new optimizer using particle swarm theory*, Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43, 1995.
- [3] S. He, E. Prempan, Q. H. Wu: *An improved particle swarm optimizer for mechanical design optimization problems*, Engineering Optimization, 36: 5, 585 — 605, 2004.
- [4] C.R. Suribabu, T.R. Neelakantan: *Design of water distribution networks using particle swarm optimization*, Urban Water Journal, 3: 2, 111 — 120, 2006.
- [5] S. Vakili, M. S. Gadala : *Effectiveness and Efficiency of Particle Swarm Optimization Technique in Inverse Heat Conduction Analysis*, Numerical Heat Transfer, Part B: Fundamentals, 56: 2, 119 — 141, 2009.
- [6] M. Mologa, C. Smutnicki: *Test functions for optimization needs*, 2005.
- [7] D. M. Himmelblau: *Applied Nonlinear Programming*, Mcgraw-Hill, 1972.
- [8] Farkas József: *Fémszerkezetek*, Tankönyvkiadó, Budapest, 1983.
- [9] Det Norske Veritas (DNV): *Buckling strength analysis*, Classification Notes No.30.1. Høvik, Norway, 1995
- [10] *Eurocode 3 Design of steel structures Part 1-5: Plated structural elements*, 2006.
- [11] József Farkas, Károly Jármái: *Optimum design and cost comparison of a welded plate stiffened on one side and a cellular plate both loaded by uniaxial compression*, Welding in the World, 50: No.3-4, 45-51, 2006.
- [12] József Farkas, Károly Jármái: *Design and optimization of metal structures*, Horwood, Chichester, UK, 2008.

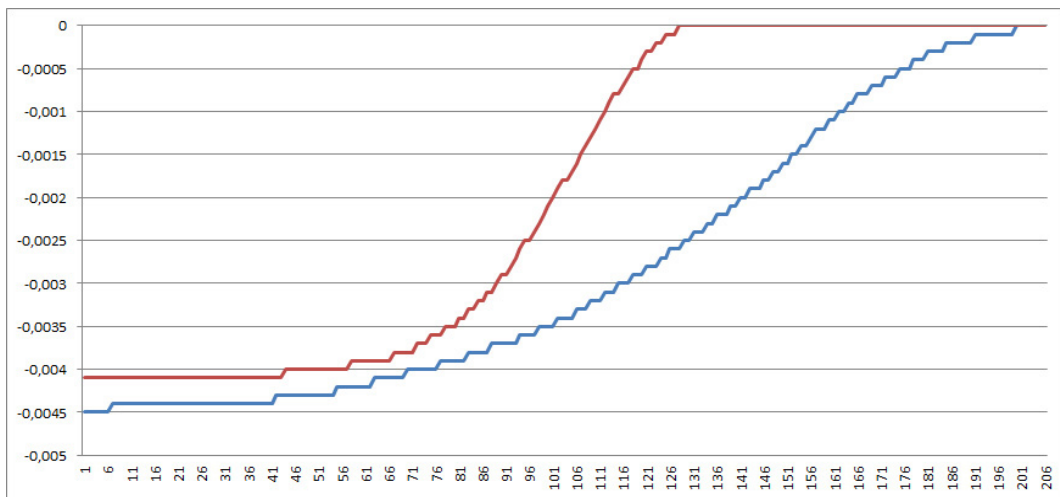
5. fejezet

Függelék

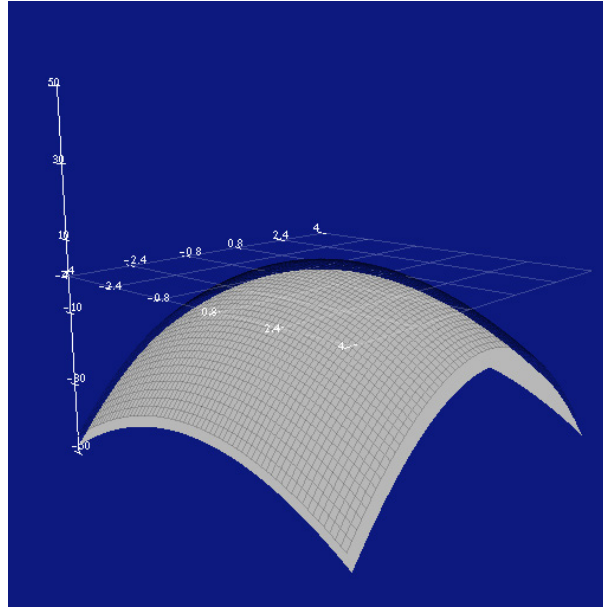
A következő ábrákon a függvényeket ábrázoló képek alatt a globális legjobb változását figyelhetjük meg az idő függvényében az adott függvény esetén. A standard algoritmus eredményét minden esetben kék, a gradienst használó algoritmus eredményét minden esetben piros szín jelöli. Az algoritmusok minden esetben ezer részecskével futottak, az ábrák elkészítéséhez száz futás átlaga lett felhasználva. A függvényekről látható képeket az elkészült Java alkalmazás állította elő. Mivel az alkalmazásban az algoritmus maximumkeresésre lett tervezve, emiatt a minimálási problémák esetében a függvények mínusz egyszeres szorzót kaptak, ez a formulák esetében is feltüntetésre került.



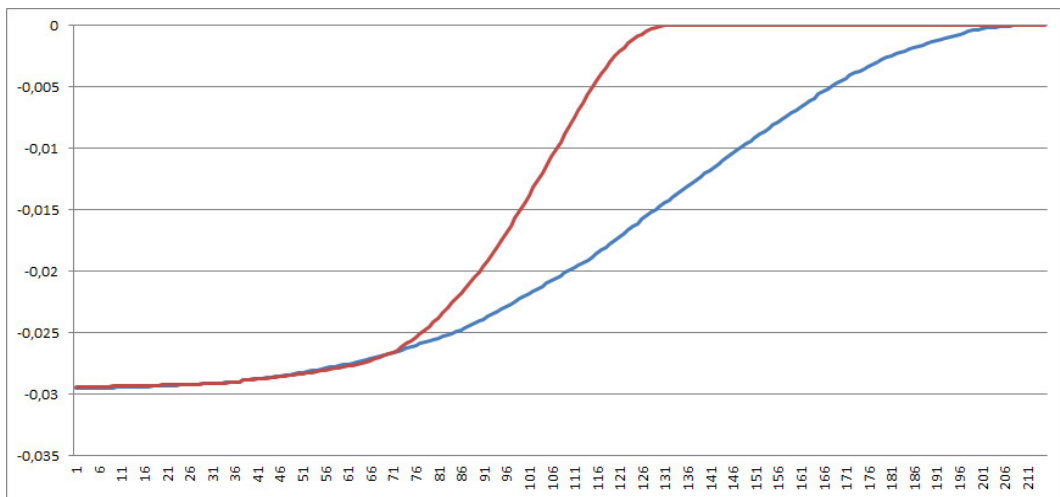
5.1. ábra. De Jong függvény: $f(x, y) = -(x^2 + y^2)$



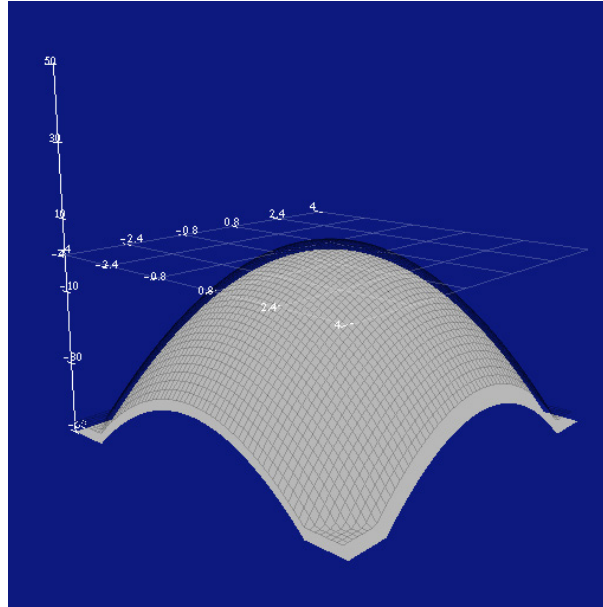
5.2. ábra. Global best változása az iteráció függvényében De Jong függvény esetén.



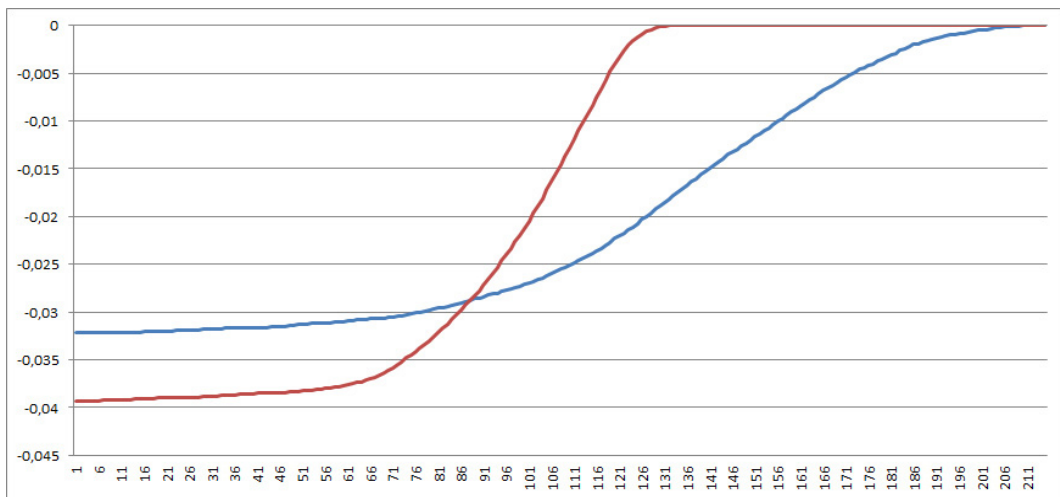
5.3. ábra. Ellipszoid függvény: $f(x, y) = -(x^2 + 2y^2)$



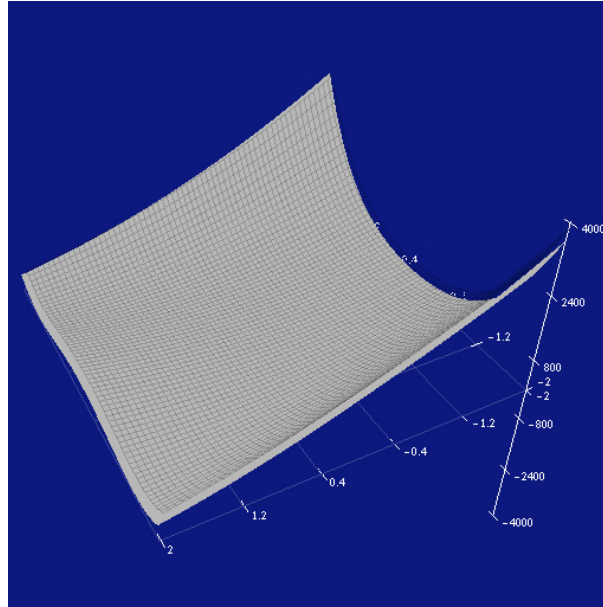
5.4. ábra. Global best változása az iteráció függvényében Ellipszoid függvény esetén.



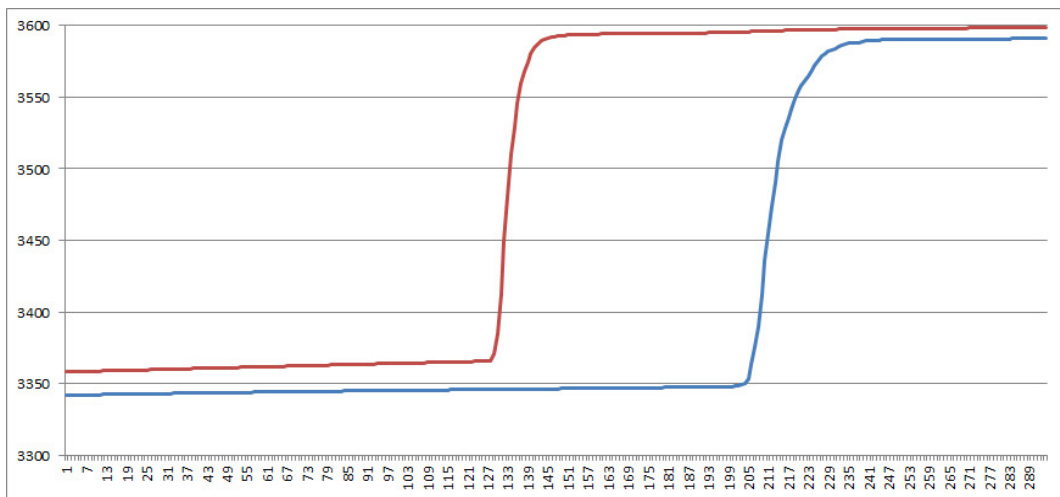
5.5. ábra. Elforgatott Ellipszoid függvény: $f(x, y) = -(2x^2 + y^2)$



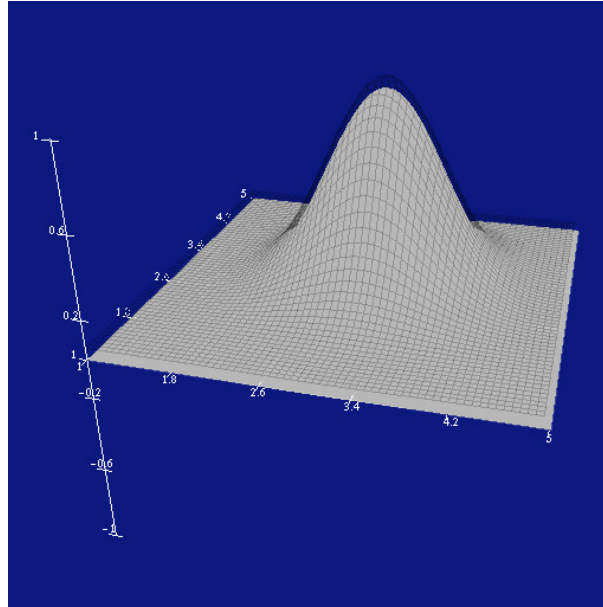
5.6. ábra. Global best változása az iteráció függvényében Elforgatott Ellipszoid függvény esetén.



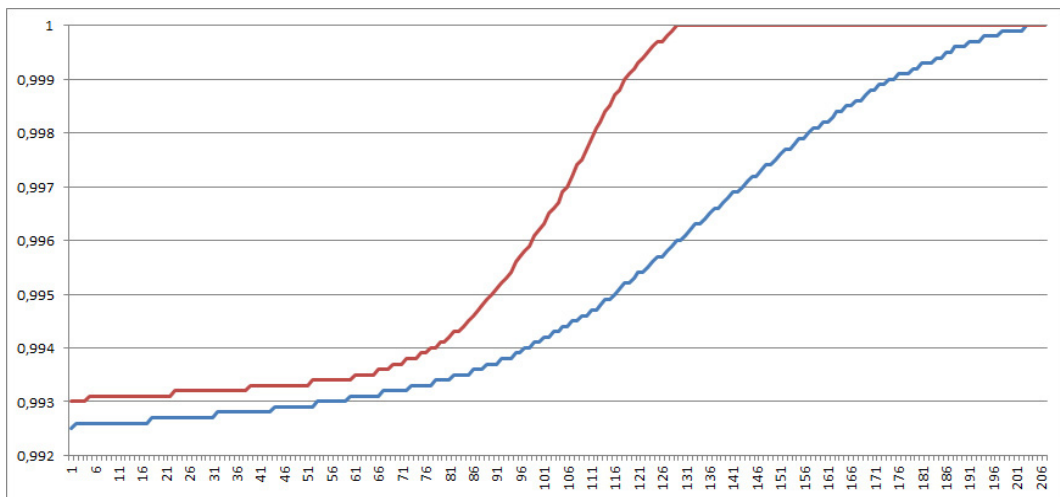
5.7. ábra. Rosenbrock függvény: $f(x, y) = 100(y - x^2)^2 + (1 - x)^2$



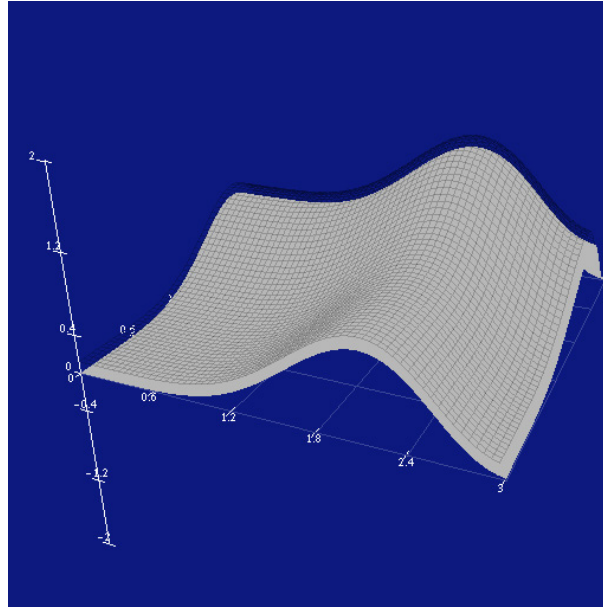
5.8. ábra. Global best változása az iteráció függvényében Rosenbrock függvény esetén.



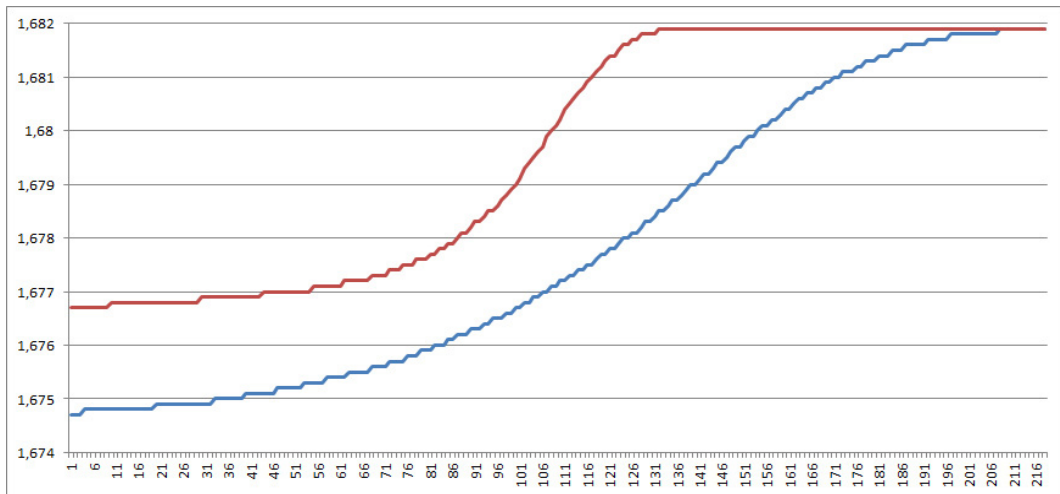
5.9. ábra. Easom függvény: $f(x, y) = \cos(x) \cos(y) e^{-(x-\pi)^2 - (y-\pi)^2}$



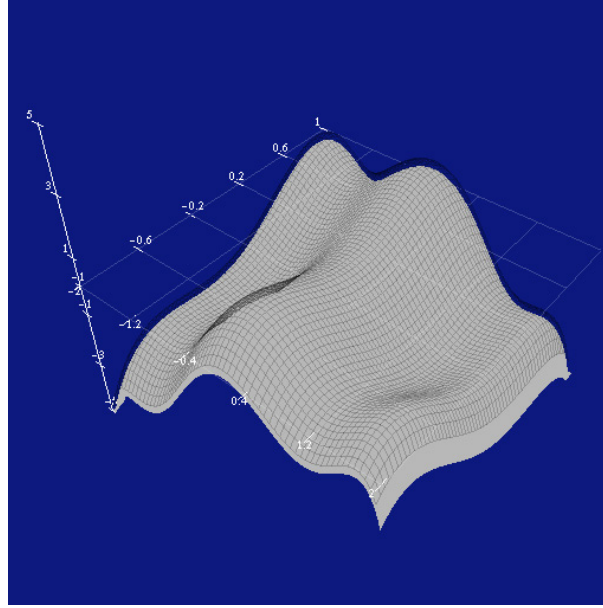
5.10. ábra. Global best változása az iteráció függvényében Easom függvény esetén.



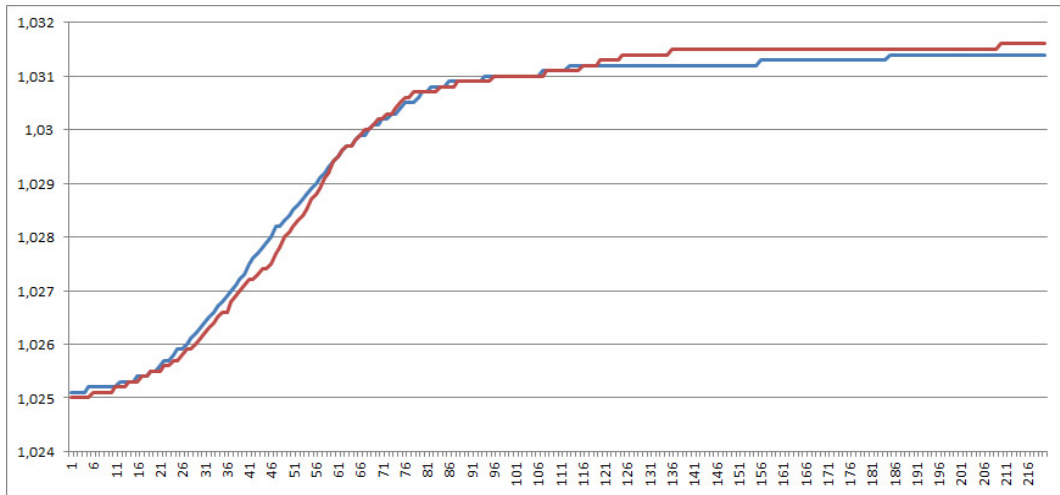
5.11. ábra. Michalewicz függvény: $f(x, y) = \sin(x) \sin\left(\frac{x^2}{\pi}\right)^2 + \sin(y) \sin\left(\frac{y^2}{\pi}\right)^2$



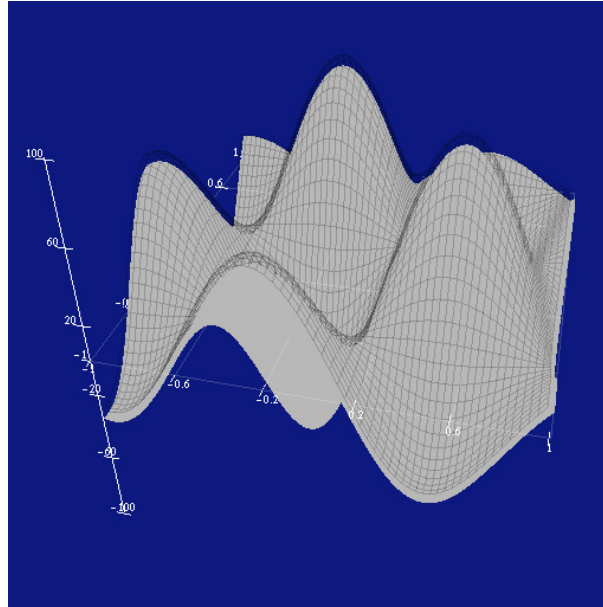
5.12. ábra. Global best változása az iteráció függvényében Michalewicz függvény esetén.



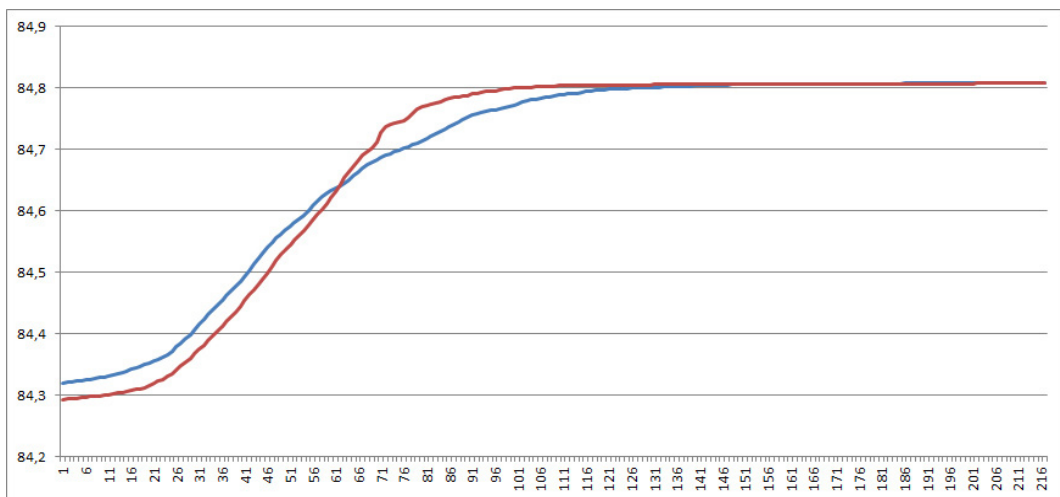
5.13. ábra. Six-hump camel back függvény: $f(x, y) = -\left(4 - 2.1x^2 + \frac{x^4}{3}\right)x^2 - xy - (-4 + 4y^2)y^2$



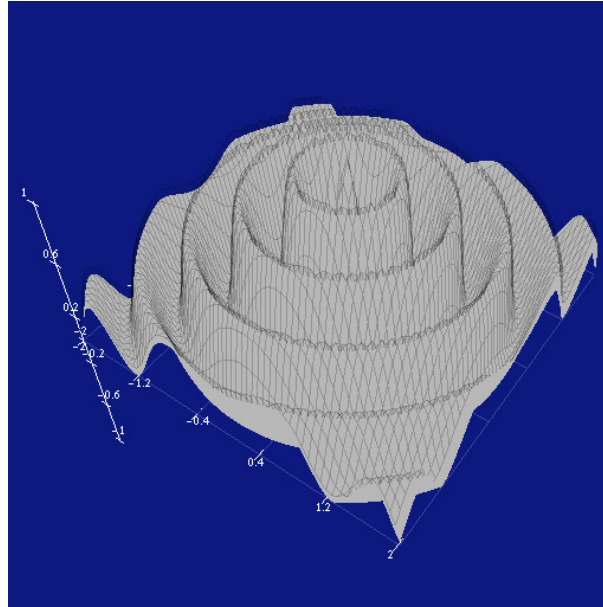
5.14. ábra. Global best változása az iteráció függvényében Six-hump camel back függvény esetén.



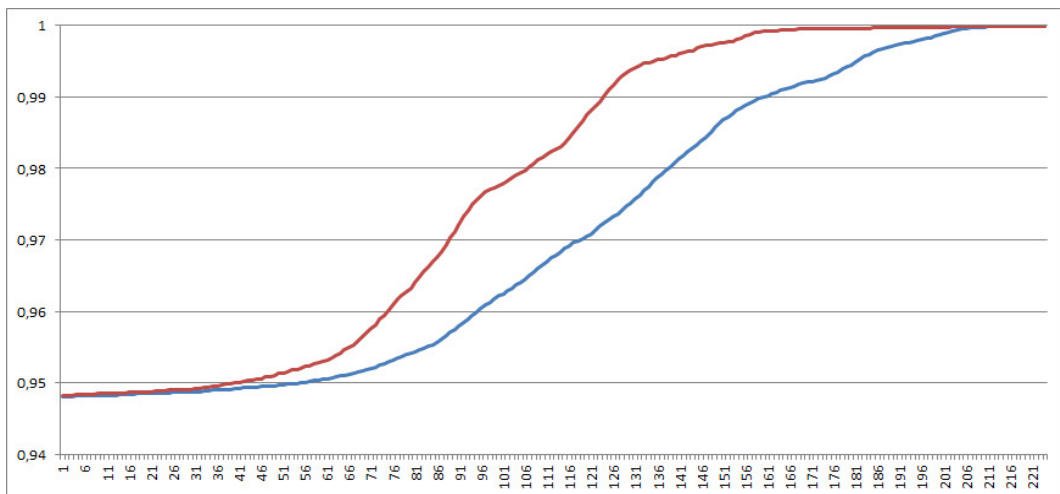
5.15. ábra. Shubert függvény: $f(x, y) = \sum_{i=1}^5 i \cos((i+1)x + 1) \sum_{i=1}^5 i \cos((i+1)y + 1)$



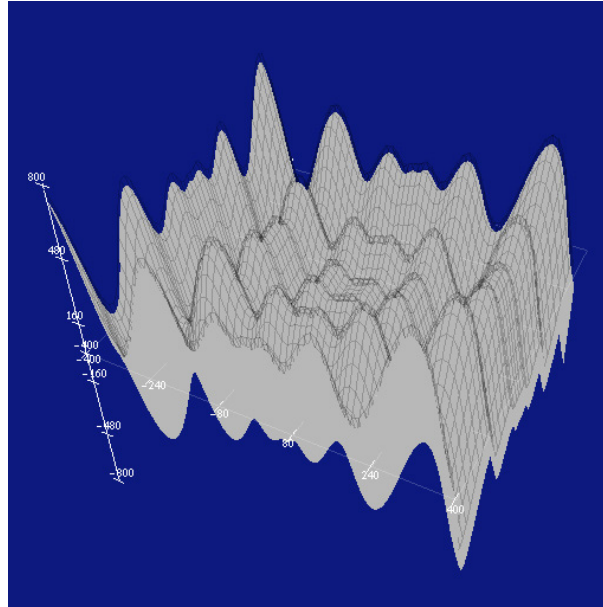
5.16. ábra. Global best változása az iteráció függvényében Shubert függvény esetén.



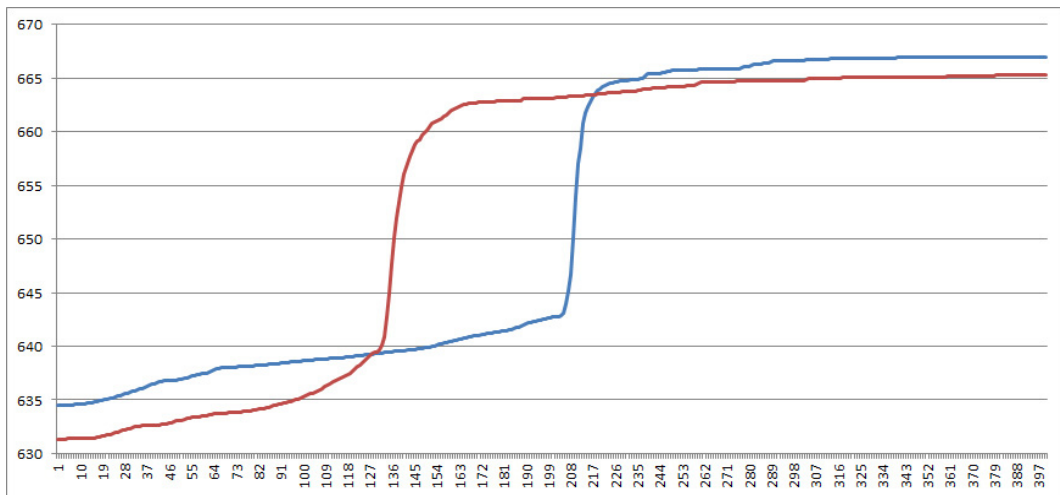
5.17. ábra. Drop Wave függvény: $f(x, y) = \frac{1 + \cos(12\sqrt{x^2 + y^2})}{\frac{1}{2}(x^2 + y^2) + 2}$



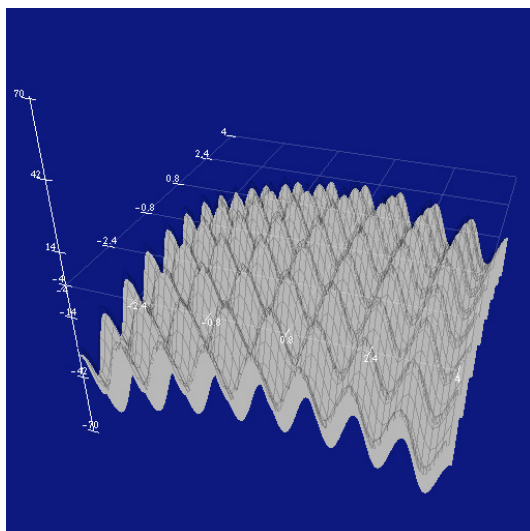
5.18. ábra. Global best változása az iteráció függvényében Drop Wave függvény esetén.



5.19. ábra. Schwefel függvény: $f(x, y) = -x \sin(\sqrt{|x|}) - y \sin(\sqrt{|y|})$

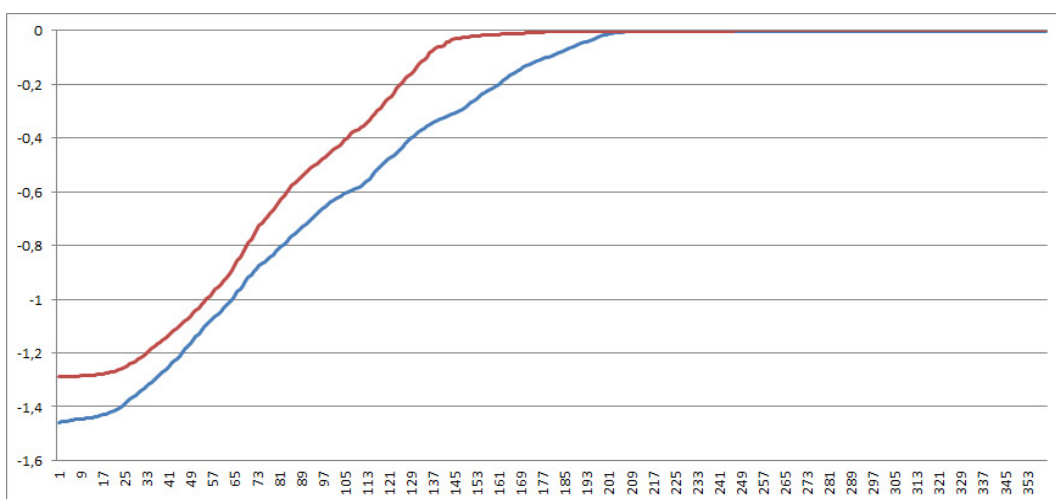


5.20. ábra. Global best változása az iteráció függvényében Schwefel függvény esetén.

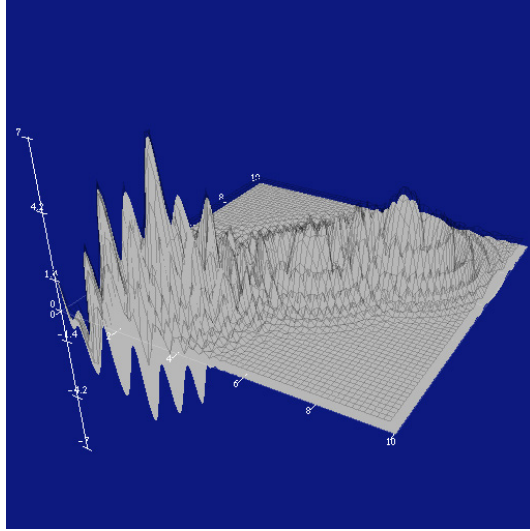


5.21. ábra. Rastrigin függvény:

$$f(x, y) = - \left(20 + [x^2 - 10 \cos(2\pi x)] + [y^2 - 10 \cos(2\pi y)] \right)$$



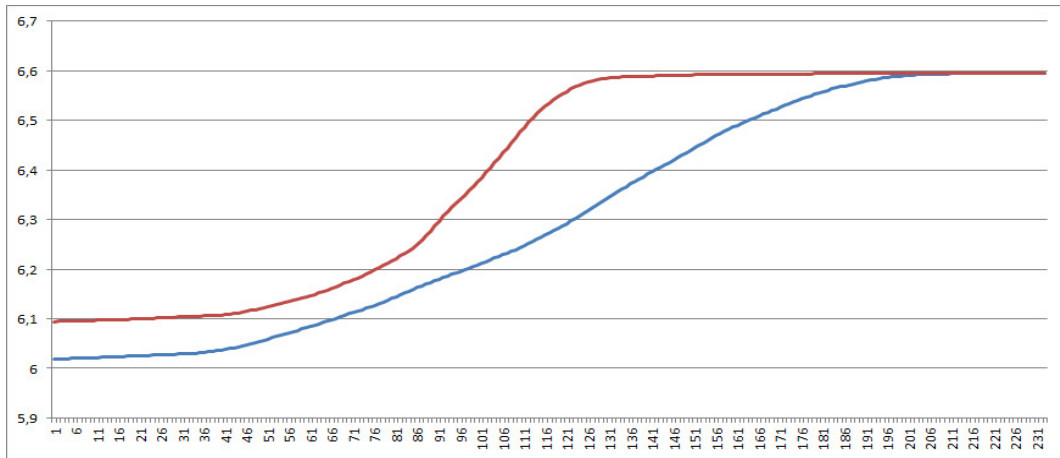
5.22. ábra. Global best változása az iteráció függvényében Rastrigin függvény esetén.



5.23. ábra. Langermann függvény:

$$f(x, y) = \sum_{i=1}^m c(i) e^{-\frac{1}{\pi} \Theta} \cos(\pi \Theta), \Theta = (x - a(i))^2 + (y - a(i))^2,$$

$$m = 5, a = [3, 5, 2, 1, 7], b = [5, 2, 1, 4, 9], c = [1, 2, 5, 2, 3]$$



5.24. ábra. Global best változása az iteráció függvényében Langermann függvény esetén.