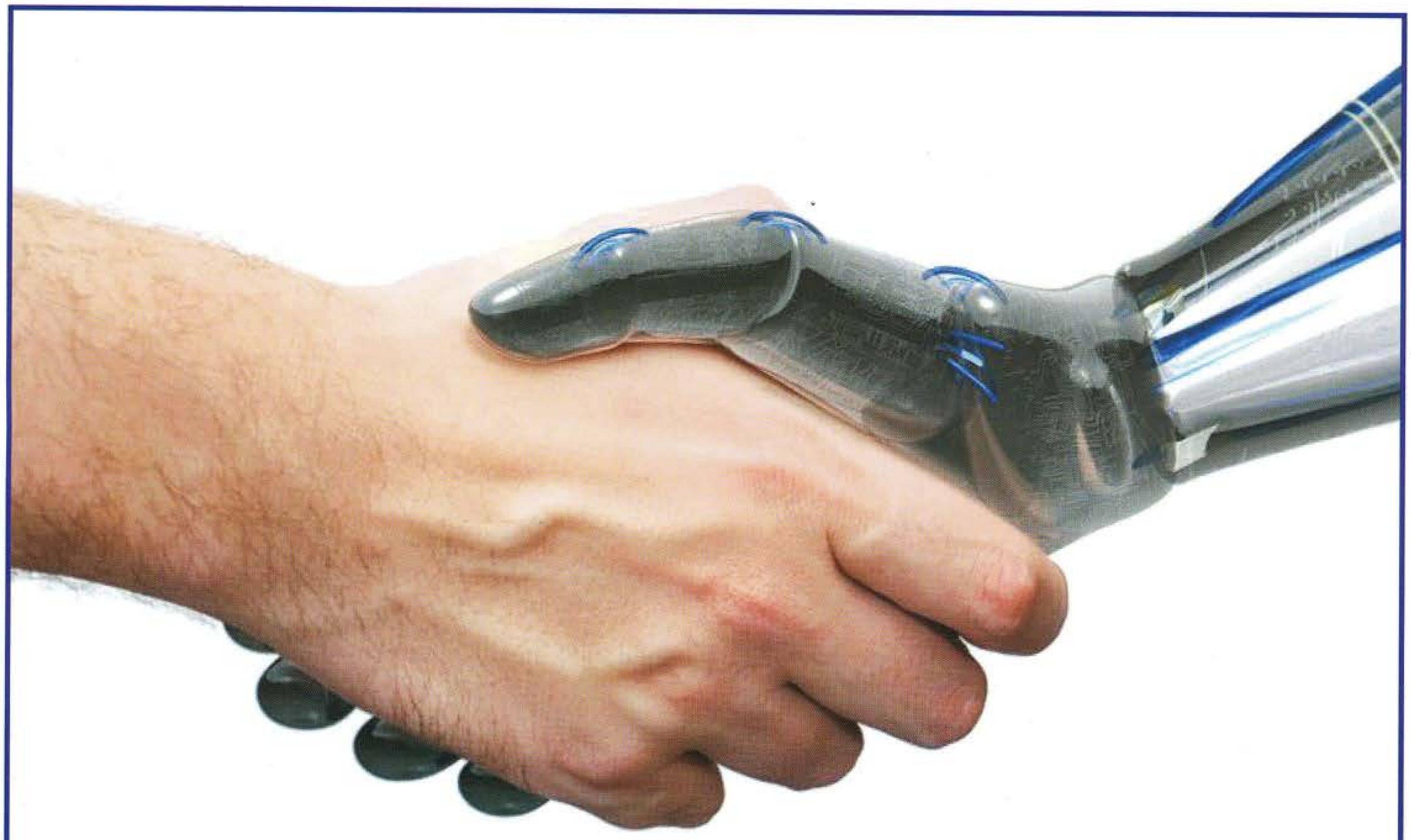
GÉP

A GÉPIPARI TUDOMÁNYOS EGYESÜLET MŰSZAKI FOLYÓIRATA



2014/3.

nka
Nemzeti Kulturális Alap

36 oldal
LXV. évfolyam

TARTALOM

A tanulmány részlegesen kapcsolódó rúdelemekből felépített vékonyfalú zárt szelvényű rudak egyenletes (Saint-Venant féle) csavarási feladatával foglalkozik. Az egyes rúdelemek kapcsolata nem tökéletes, tengelyirányú elcsúszást is megengedhet. Az elcsúszást gátló erő, arányos az elcsúszás mértékével.

A közlemény összefoglalja a Bánki-turbina esetében az ütközésmentes belépést és a kilépés perdület mentességét biztosító üzemi- és járókerék méret adatok meghatározására szolgáló eljárást. A paraméterek kölcsönös kapcsolatát a lapátkongruens súrlódásmentes áramlás feltételezésével, a sebességi háromszögek geometriáján alapuló összefüggésekkel írták le. Az így adódó komplex összefüggések nem tekinthetőek át, ezért a gyakorlat számára diagramokban feldolgozva is közölték az eredményeket.

Túlterhelés esetén a munkavezetékben keletkező hő függ a vezető villamos ellenállásától és a felületének hőleadó képességétől. Ismert, hogy a felületi oxidáció miatt a régebben felszerelt munkavezetékek jobban ellenállnak a túlterhelésnek az oxidált felület jobb hőleadó képessége miatt. A felület jobb hőleadó képésségének másik előnye, hogy a munkavezetékre fagyott kemény zúzmara kisebb terhelés mellett leolvad a munkavezetékről.

Ebben a cikkben a szerzők részletezik az eddig elért eredményeiket, külön kiemelve az új evolúciós és rajintelligencia eljárásokat. A heurisztikus algoritmus gyűjtemény jelenleg 15 eljárást tartalmaz. Ezen algoritmusok képesek nagyon komplex függvények optimálására.

Szerkezeti acélok hegesztésekor az anyagba bevitt hő mennyisége döntő tényező. Emiatt ezen alapanyagok hegesztése eddig nem volt gazdaságosan kivitelezhető impulzusívvel. Az impulzusív általi magas hőbevitel miatt jelentősen csökkenteni kellett a teljesítményt. Ez azonban döntő hatással volt a hegesztési sebesség csökkenésére, így szerkezeti acélok esetén, mindeddig egyértelműen a hagyományos szórtívet részesítettük előnyben.

HEURISZTIKUS ALGORITMUSOK HATÉKONYSÁG-VIZSGÁLATA ÁLTALUNK LÉTREHOZOTT TESZTFÜGGVÉNYEK SEGÍTSÉGÉVEL

BENCHMARKING HEURISTIC ALGORITHMS WITH TEST FUNCTIONS CREATED BY US

*Marcsák Gábor Zoltán**, *Prof. Dr. Jármai Károly***

ABSTRACT

The process of optimization is finding the best solution to a given problem, when the amount of available resources (time, computational power) are often restricted. Despite the rapid development of computer science, most optimization problems can't be solved by evaluating all feasible solutions, simply because the search space is enormously huge, and would require exponential computation time to be fully explored. We encounter optimization problems almost everywhere in life, for example engineering, information technology, medicine, and many other areas of science. In our previous article, we examined the most frequently used lifting machinery in the modern industry, the bridge crane, from an aspect of structural optimization. We minimized the weight of the crane's main beam, in order to reduce production and operating costs. For the optimization process, we used a heuristic algorithm collection implemented by us, which is under continuous development. In this paper, we would like to summarize our latest results regarding the heuristic algorithm collection, particularly the novel evolutionary and swarm intelligence methods we use.

1. BEVEZETÉS

Az optimálás során a cél a legjobb megoldás megtalálása egy adott problémára úgy, hogy a rendelkezésre álló erőforrások (idő, számítási teljesítmény) gyakran korlátozottak. A számítógépek rohamos fejlődése ellenére még mindig sok olyan feladat ismert, ami nem oldható meg pusztán a számítási teljesítményre alapozva. Az élet számos területén találkozhatunk optimálási problémákkal, legyen szó mérnöki, informatikai, orvosi, vagy bármilyen egyéb tudományterületről. Korábbi cikkünkben [1] a modern ipar egyik leggyakrabban használt emelő szerkezetét, a futódarut vizsgáltuk szerkezetoptimálási szempontból. Az egyik legfontosabb szerkezeti elem, a főtartó súlyának minimalizálását végeztük, az üzemeltetési és gyártási költségek csökkentése érdekében.

Az optimáláshoz egy általunk megvalósított heurisztikus algoritmus gyűjteményt használtunk, amit folyamatosan fejlesztünk. Ebben a cikkben az említett fejlesztéseket szeretnénk részletezni, külön kiemelve az új evolúciós és rajintelligencia eljárásokat. A heurisztikus algoritmus gyűjtemény jelenleg 15 eljárást tartalmaz, legfőbb előnyét pedig éppen az jelenti, hogy a különböző optimálási problémákat nem egy algoritmussal oldja meg, hanem mindegyik eljárás eredményt szolgáltat. Előfordulhat ugyanis, hogy adott típusú feladat esetében valamelyik algoritmus rosszabbul működik mint a többi, azonban egy másik típusú feladatnál ennek épp az ellenkezője igaz.

A heurisztikus algoritmusok hatalmas előnye, hogy nagyon bonyolult problémák esetén is képesek viszonylag rövid idő alatt, kevés számítás árán eredményt szolgáltatni. Hátrányuk azonban, hogy nem garantálható teljes bizonyossággal az optimális megoldás megtalálása. Minél több egymástól független algoritmussal próbálunk megoldani egy problémát, annál inkább bízhatunk az optimum, vagy optimum-közeli megoldás megtalálásában.

2. OPTIMÁLÁSI PROBLÉMÁK LEÍRÁSA

A problémák megoldásához szükséges azok matematikai megfogalmazása. Az optimálás során a különböző értékek két csoportba sorolhatók: előre megadott (bemenő) paraméterek, illetve döntési változók. Az alapvető különbséget az jelenti közöttük, hogy a bemenő paraméterek értéke rögzített (konstans), ezzel szemben a döntési változók értéke az optimálás során változik. Annak függvényében, hogy a döntési változók milyen értékeket vehetnek fel, léteznek diszkrét és folytonos változók. A döntési változók értékét különböző feltételek definiálásával befolyásolhatjuk. Ha definiálunk feltételt, akkor

* *logisztikai mérnök MSc hallgató, Miskolci Egyetem*
** *egyetemi tanár, rektorhelyettes, Miskolci Egyetem*

feltételes, egyébként feltétel nélküli optimálást végzünk. A feltételek matematikailag lehetnek egyenlőségek vagy egyenlőtlenségek. Az optimálás célját, a megoldási alternatívák vizsgálatát célfüggvény segítségével határozzuk meg. Abban az esetben, ha csupán egyetlen célfüggvény van, egycélfüggvényes, ellenkező esetben többcélfüggvényes optimálásról beszélünk. Egycélfüggvényes optimálás lehet például egy rácsos tartó súlyminimumának meghatározása, mely egyetlen végeredményt, legtöbbször egy skalár értéket ad [2]. A többcélfüggvényes eset bonyolultabb, mivel az egyes, általában egymással konfliktusban lévő célfüggvények minimumának és maximumának egyidejű meghatározása szükséges. Vizsgáljunk például egy egyszerű kéttámaszú tartót, ahol az egyik minimálandó célfüggvény a tartó súlya, a másik célfüggvény pedig a maximális merevség. Nyilvánvalóan a két célfüggvény egymással konfliktusban van. A többcélfüggvényes optimálási feladat megoldását Pareto fogalmazta meg, ezért szokás Pareto optimumnak nevezni. A definíció szerint akkor beszélünk optimumról, ha egyik célfüggvény értéke sem javítható úgy, hogy legalább egy másik célfüggvény értéke ne romlana. Az optimum tehát nem olyan egyértelmű, mint egycélfüggvényes optimálás esetén, mert alternatív megoldások egész halmazát (Pareto halmaz) jelenti. A végső megoldás csak további kritériumok, feltételek segítségével határozható meg. Összefoglalásképpen tehát az optimálási problémák leírásához szükséges a változók, feltételek és célfüggvények definiálása.

3. HEURISZTIKUS ALGORITMUSOK ELMÉLETI HÁTTERE

Az optimáláshoz heurisztikus algoritmusokat használtunk, akár nagyon bonyolult, sok bemenő paraméterrel, döntési változóval és feltétellel leírható, többcélfüggvényes optimálási feladatok is eredményesen oldhatók meg segítségükkel. A hagyományos kereső algoritmusokkal szemben a heurisztikus algoritmusok próbálgatással, a korábban megszerzett tapasztalatok felhasználásával jutnak eredményre. Szokás ezért informált kereső eljárásoknak is nevezni őket.

A heurisztika kifejezés a görög „heureszisz" szóból származik, melynek jelentése rátalálás. A heurisztikus algoritmusok nem vizsgálják az összes lehetséges kimenetelt (sok esetben ez egyébként is fizikai képtelenség), csupán a problématér egy adott részletét.

A heurisztikus algoritmusok hatalmas előnye, hogy nagy bonyolultságú problémák esetében is képesek viszonylag rövid idő alatt, kevés számítás árán eredményt szolgáltatni. Hátrányuk azonban, hogy nem garantálható teljes bizonyossággal az optimális megoldás megtalálása. Használatukkal tehát sebességet kapunk, azonban cserébe pontatlansággal fizetünk.

A heurisztikus algoritmusok egyik jellemző tulajdonsága, hogy működésüket gyakran sztochasztikus jellemzők befolyásolják. Hatékonyságvizsgálatuk ezért bonyolult feladat, mivel kis túlzással nincs két egyforma futás. Másik jellemző tulajdonságuk, hogy működésük gyakran valamilyen természeti jelenségen alapul.

4. HEURISZTIKUS ALGORITMUSOK GYŰJTEMÉNYE

A heurisztikus algoritmus gyűjtemény C# programozási nyelven lett megvalósítva, jelenleg 15 eljárást tartalmaz. A futódaru főtartó optimálásával foglalkozó cikkünkben [1] 10 algoritmust már részletesen ismertettünk (Bacterial Foraging Algorithm [BFOA], Bees Algorithm [BA], Cultural Algorithm [CA], Differential Evolution [DE], Harmony Search [HS], Memetic Algorithm [MA], Nelder-Mead Algorithm [NM], Particle Swarm Optimization [PSO], Random Search [RS], Simulated Annealing [SA]). Az alábbiakban a nemrégiben beépített evolúciós (Cross-Entropy Method [CEM]) és négy rajintelligencia (Bat Algorithm [BATA], Cuckoo Search [CS], Firefly Algorithm [FF], Multi-swarm Optimization [MSO]) ismertetjük.

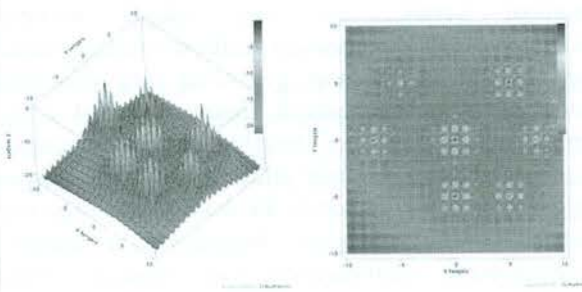
4.1. Bat Algorithm (BATA)

Egy Swarm Intelligence (Rajintelligencia) elven működő eljárás. A rajintelligencia (kollektív intelligencia) módszerek közös tulajdonsága, hogy nagyszámú homogén egyed viselkedésmintáin alapulnak. Az alapelv szerint lehetséges, hogy egy individuális egyed nem képes megoldani adott feladatot, azonban ha nagyszámú egyed csoportot alkot, akkor a csoport kollektív intelligenciája már elég lehet a feladat sikeres megoldásához.

A Bat, vagy magyarul denevér algoritmust Xin-She Yang dolgozta ki 2010-ben különböző mérnöki problémák megoldására. Működési elve szerint a Particle Swarm algoritmushoz hasonlóan rajintelligencia eljárás, a denevérek visszhang alapján való tájékozódását másolja. A denevérek teljes sötétségben is képesek elejteni zsákmányukat, az általuk kibocsátott hang környezetről való visszaverődése alapján [3].

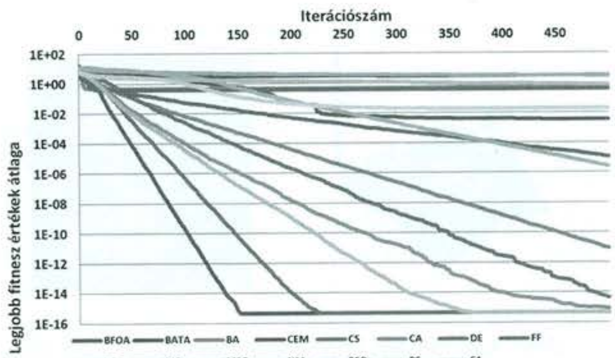
4.2. Cross-Entropy Method (CEM)

Az algoritmus az evolúciós algoritmusok osztályába tartozik, mely eljárások közös tulajdonsága, hogy Darwin evolúciós elméletén alapul működésük. Ennek megfelelően központi eleme a természetes kiválasztódás, tehát a problémára jobb megoldást adó



4. ábra: Komplex függvény háromdimenziós ábrázolása és színtérképe Gábor-függvényen alapuló súlyozással (F4)

6.AZ EREDMÉNYEK VIZSGÁLATA

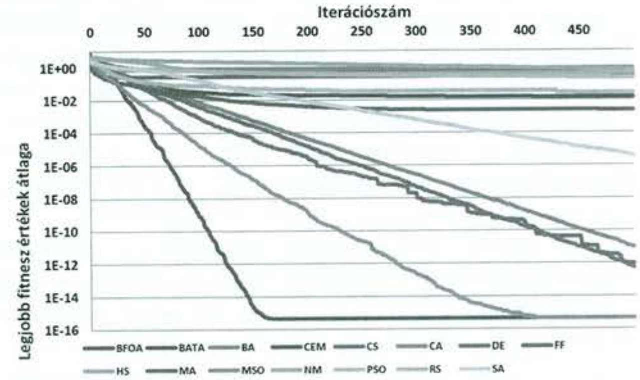


5. ábra: Legjobb fitnesz értékek átlaga 100 Monte Carlo futás adatai alapján (F1)

1. táblázat: Fitnesz értékek alakulása 500 iteráció után, 100 Monte Carlo futás adatai alapján (F1)

	Átlag fitnesz (Mean)	Legjobb fitnesz	Legrosszabb fitnesz	Fitnesz szórás (St. Dev.)	Átlag futási idő [ms]
BFOA	0,003284	0,000541	0,007118	0,001593	28,88
BATA	3,04138	0,000254	12,55774	2,833844	15,34
BA	8,84E-12	1,61E-12	2,24E-11	4,3E-12	24,81
CEM	**4,44E-16**	**4,44E-16**	**4,44E-16**	**0**	**143,95**
CS	4,21E-15	4,44E-16	1,25E-13	1,32E-14	63,29
CA	0,500321	4,44E-16	6,88414	1,159808	48,54
DE	**4,44E-16**	**4,44E-16**	**4,44E-16**	**0**	**17,35**
FF	1,08E-05	7,58E-07	2,83E-05	6,15E-06	491,88
HS	2,791707	0,006343	6,884312	1,665716	3,42
MA	0,37983	0,002013	3,512875	0,882168	1954,37
MSO	8,35E-16	4,44E-16	7,55E-15	1,23E-15	52,64
NM	2,52E-06	6,85E-07	6,53E-06	1,21E-06	2,03
PSO	**4,44E-16**	**4,44E-16**	**4,44E-16**	**0**	**19,57**
RS	0,768881	0,015493	2,272974	0,526217	36,65
SA	0,019018	1,11E-06	1,900706	0,190069	45,29

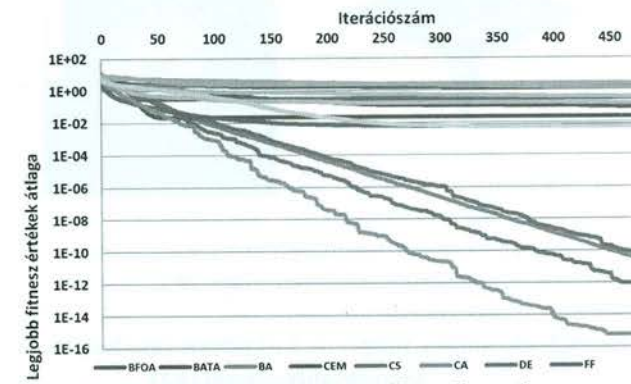
A tesztfüggvények esetében legfeljebb 10^{-16} pontosság volt elérhető, mivel a .NET keretrendszer néhány általam használt matematikai függvénye (pl. Math.Sin(), Math.Cos(), Math.Exp(), stb.) ennyit tesz lehetővé. Az Ackley's function (F1) esetében a CEM, DE és PSO eljárások bizonyultak a legjobbnak. A lokális optimumok számának növekedésével (F2) A CEM és MSO kiemelkedtek a többi algoritmus közül, a BA, CS és DE nagyjából tudták velük tartani a lépést, a többi eljárás azonban eléggé lemaradt. Ha a tesztfüggvényt súlyozó függvénnyel zajosítottuk (F3, F4), akkor egyértelműen kiderült az MSO eljárás létjogosultsága. A PSO-hoz képest több raj használata egyértelmű előnyt jelentett a globális optimum felderítése szempontjából.



6. ábra: Legjobb fitnesz értékek átlaga 100 Monte Carlo futás adatai alapján (F2)

2. táblázat: Fitnesz értékek alakulása 500 iteráció után, 100 Monte Carlo futás adatai alapján (F2)

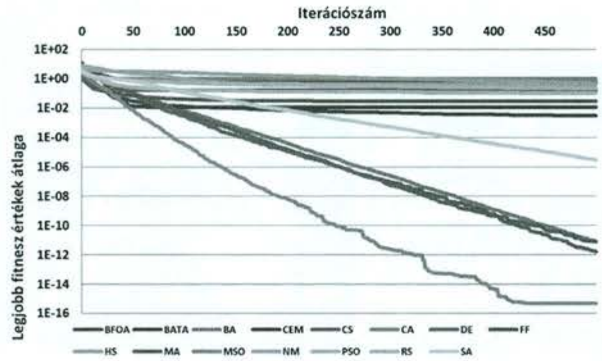
	Átlag fitnesz (Mean)	Legjobb fitnesz	Legrosszabb fitnesz	Fitnesz szórás (St. Dev.)	Átlag futási idő [ms]
BFOA	0,013692	0,000257	0,505462	0,070438	227,22
BATA	0,559007	0,000186	3,803036	0,788892	85,23
BA	9,29E-12	1,54E-13	2,17E-11	4,55E-12	117,99
CEM	**4,44E-16**	**4,44E-16**	**4,44E-16**	**0**	**224,69**
CS	5,46E-13	4,44E-16	8,59E-12	1,18E-12	203,49
CA	0,735248	4,44E-16	3,135372	0,887617	135,97
DE	9,71E-13	4,44E-16	7,03E-11	7,43E-12	105,5
FF	0,002287	3,1E-07	0,079827	0,011588	2145,3
HS	0,612692	0,002125	2,87193	0,696436	6,65
MA	0,270109	0,00069	2,175912	0,443973	2904,53
MSO	**4,44E-16**	**4,44E-16**	**4,44E-16**	**0**	**327,76**
NM	0,4	1,16E-07	2	0,460566	7,17
PSO	0,025001	4,44E-16	1	0,130558	105,45
RS	0,2049	0,014181	0,556557	0,135328	219,55
SA	3,61E-06	2,61E-07	7,96E-06	1,88E-06	237,76



7. ábra: Legjobb fitnesz értékek átlaga 100 Monte Carlo futás adatai alapján (F3)

3. táblázat: Fitnesz értékek alakulása 500 iteráció után, 100 Monte Carlo futás adatai alapján (F3)

	Átlag fitnesz (Mean)	Legjobb fitnesz	Legrosszabb fitnesz	Fitnesz szórás (St. Dev.)	Átlag futási idő [ms]
BFOA	0,076815	0,000231	0,63933	0,168965	284,33
BATA	1,40407	0,000163	7,330941	1,816438	112,68
BA	8,51E-12	7,86E-14	2,5E-11	4,42E-12	159,82
CEM	0,02101	4,44E-16	0,5	0,098784	291,42
CS	1,2E-11	4E-15	2,55E-10	3,1E-11	263,77
CA	2,088154	4,44E-16	6,441469	1,969068	180,58
DE	1,65E-13	4,44E-16	9,02E-12	9,72E-13	144,14
FF	0,005606	3,83E-07	0,197582	0,024592	2809,4
HS	1,158764	0,003844	5,473186	1,447596	7,06
MA	0,306188	0,000615	4,689873	0,658486	3502,67
MSO	**5,15E-16**	**4,44E-16**	**4E-15**	**5E-16**	**401,75**
NM	2,504176	5,07E-08	7,330936	2,418446	9,69
PSO	0,145003	4,44E-16	1	0,343002	133,55
RS	0,35094	0,024371	0,882359	0,237366	274,84
SA	0,005006	3,09E-07	0,500007	0,05	278,69



8. ábra: Legjobb fitnesz értékek átlaga 100 Monte Carlo futás adatai alapján (F4)

4. táblázat: Fitnesz értékek alakulása 500 iteráció után, 100 Monte Carlo futás adatai alapján (F4)

	Átlag fitnesz (Mean)	Legjobb fitnesz	Legrosszabb fitnesz	Fitnesz szórás (St. Dev.)	Átlag futási idő [ms]
BFOA	0,002931	0,000344	0,0091	0,001666	246,05
BATA	0,595731	9,35E-05	4,500406	0,734037	93,23
BA	8,69E-12	7,18E-13	2,26E-11	4,17E-12	123,02
CEM	0,011858	4,44E-16	0,5	0,071759	243,87
CS	1,64E-12	4,44E-16	3,76E-11	5,59E-12	211,28
CA	1,00155	4,44E-16	4,47836	0,999758	140,32
DE	8,02E-12	4,44E-16	2,3E-10	3,2E-11	116,4
FF	0,029957	3,25E-07	0,256238	0,054979	2344,59
HS	0,495442	0,002075	3,615511	0,652229	6,16
MA	0,165668	0,000612	1,174854	0,330159	3184,65
MSO	**4,8E-16**	**4,44E-16**	**4E-15**	**3,55E-16**	**352,5**
NM	0,43	1,82E-07	2	0,466125	7,5
PSO	0,100064	4,44E-16	1	0,30149	113,78
RS	0,223209	0,022458	0,598175	0,130766	224,06
SA	2,92E-06	2,09E-07	9,32E-06	1,69E-06	233,56

7. ÖSSZEFOGLALÁS

A cikkben szakirodalomból vett és saját matematikai tesztfüggvények szélsőértékét kerestük. Az optimáláshoz egy általunk megvalósított heurisztikus algoritmus gyűjteményt használtunk, amit folyamatosan fejlesztünk. A heurisztikus algoritmus gyűjtemény jelenleg 15 eljárást tartalmaz, amiből öt algoritmust nemrégiben implementáltunk. A hatékonyságvizsgálat eredményei alapján a CEM, CS, FF és legfőképpen az MSO eljárások teljes mértékben beváltották a hozzájuk fűzött reményeket. A pozitív tapasztalatok alapján a jövőben folyamatosan szeretnénk bővíteni az elérhető heurisztikus algoritmusokat számát.

8. KÖSZÖNETNYILVÁNÍTÁS

A kutatás az Európai Unió és Magyarország támogatásával, az Európai Szociális Alap társfinanszírozásával a TÁMOP 4.2.4.A/2-11-1-2012-0001 azonosító számú „Nemzeti Kiválóság Program – Hazai hallgatói, illetve kutatói személyi támogatást biztosító rendszer kidolgozása és működtetése konvergencia program" című kiemelt projekt keretei között valósult meg. A kutató munka részben a Miskolci Egyetem stratégiai kutatási területén működő Innovációs Gépészeti Tervezés és Technológiák Kiválósági Központ keretében valósult meg, valamint az OTKA T 109860 projekt támogatásával.

9. IRODALOM

[1] MARCSÁK G. Z. és JÁRMAI K.: Futódaru főtartó szerkezetoptimálása heurisztikus algoritmusok segítségével, GÉP: A Gépipari Tudományos Egyesület Műszaki Folyóirata, ISSN 0016-8572, 2014/1. pp. 39-44

[2] MICHELL, A. G. M.: The limits of economy of material in frame-structures, Philosophical Magazine, Vol. 8(47), 1904. pp. 589-597

[3] YANG X.-S.: A New Metaheuristic Bat-Inspired Algorithm, Nature Inspired Cooperative Strategies for Optimization (NISCO 2010), (Eds. J. R. Gonzalez et al.), Studies in Computational Intelligence, Springer Berlin, Springer, 2010. pp. 65-74

[4] RUBINSTEIN R. Y.: Optimization of computer simulation models with rare events. European Journal of Operations Research, 99, 1997. pp. 89–112.

[5] YANG X. S., DEB S.: Cuckoo search via L´evy flights, Proc. of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), December 2009, India. IEEE Publications, 2009. pp. 210-214.

[6] YANG X. S.: Firefly algorithms for multimodal optimization. Stochastic Algorithms: Foundations and Applications, SAGA 2009. Lecture Notes in Computer Sciences 5792. pp. 169-178.

[7] Zhao S. Z., Liang J. J., Suganthan P. N., és Tasgetiren M. F.: Dynamic Multi-Swarm Particle Swarm Optimizer with Local Search for Large Scale Global Optimization, in Proceedings IEEE Congress on Evolutionary Computation, 2008. pp. 3845–3852.

[8] MOLGA M., SMUTNICKI C.: Test functions for optimization needs, 2005, http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf

[9] LIANG J., SUGANTHAN N., DEB K.: Novel composition test functions for numerical global optimization, Swarm Intelligence Symposium, SIS 2005. Proceedings IEEE, 2005. pp. 68-75

[10] BARCSÁK CS., JÁRMAI K.: Benchmark for testing evolutionary algorithms, 10th World Congress on Structural and Multidisciplinary Optimization, May 19 -24, 2013, Orlando, Florida, USA

Internetes hivatkozások ellenőrizve: 2014. 05. 16.