

MEGTERMÉKENYÍTÉSI OPTIMÁLÓ ALGORITMUS TESZTELÉSE A CEC2015 KÖRNYEZETBEN ÉS ALKALMAZÁSA NAGYMÉRETŰ PROBLÉMÁKRA

TESTING A FERTILIZATION OPTIMIZATION ALGORITHM IN THE CEC2015 ENVIRONMENT AND APPLYING IT TO LARGE-SCALE PROBLEMS

Hazim Nasir Ghafil*, Shaymaa Alsamia** és Dr. Jármái Károly***

ABSTRACT

This paper presents the fertilization optimization algorithm, a new optimization method based on Levy's flight and random search in a search space. Based on the fertilization of the egg in mammalian reproduction, this algorithm is biologically inspired. In CEC2015, the algorithm's performance was compared to that of other optimization algorithms using time-consuming benchmarks and large-scale optimization problems. The fertilization optimization algorithm outperformed other optimization algorithms in many cases, and the study and comparison results are encouraging for the use of the fertilization optimization algorithm in other potential applications.

1. BEVEZETÉS

Története során az optimáló algoritmusokat ihlette a természet vagy az ember által létrehozott jelenségek, mely által inspirált matematikai formulák kerültek bevezetésre, amelyekkel különböző területek optimálási problémáit lehetett megoldani a tudományok területén. Konkrétabban, az optimáló algoritmusokat arra használták, hogy megtalálják a maximumát vagy minimumát egy célfüggvénynek. Széles körű alkalmazásuk van az iparban [1] és a mérnöki tudományokban, a robotikában [2] és a szerkezeteknél [3]. Az optimáló módszer fejlesztőket jobban érdeklik a jelenségek, amelyek olyan új módszer kifejlesztésére inspirálhatják őket, melyekkel megbízhatóbbak, gyorsabbak lesznek, vagy képesek megtalálni a meglévő problémákra a legjobb megoldásokat.

Az egyik inspirációs motor az állatraj, madárcsapat, halraj és rovarok, amelyek a rajintelligencia [4, 5] módszerek kifejlesztéséhez vezettek. Ez a kifejezés lehet az egyedek egy csoportja között felhalmozódó és megosztott tudásként definiálható, és ez a fajta intelligencia nem érhető el egyetlen egyed által. Példák

a rajintelligenciára Particle Swarm Optimization (PSO) [6], Artificial Bee Colony (ABC) [7], és Grey Wolf Optimálás (GWO) [8]. Nem minden biológiailag inspirált algoritmus rajintelligencia; a baktériumok és az invazív gyomok optimálása nem követi a raj szabályait. Ebben a cikkben egy biológiailag ihletett algoritmus a megtermékenyítési folyamatból a szaporodási traktusban az emlősök vonatkozásában származik. Az új algoritmus neve Megtermékenyítési Optimáló (Fertilization Optimization FO) algoritmusnak nevezik. A számításgépes benchmarkok a CEC2015-ben található [9] a kísérletek során. Ezekben a matematikai optimálási problémákon keresztül az FO-t más metaheurisztikákkal hasonlítottuk össze. Figyelemre méltó, hogy a FO nagy teljesítményt mutatott és számos esetben megelőzött számos más algoritmust. A matematikai optimálási problémák sokfélesége és nehézsége, amelyeken a FO sikeresen át tudott jutni, bizonyította a megtermékenyítő algoritmus megbízhatóságát a matematikai optimálásban. Röviden, a FO algoritmus a következőképpen írható le.

Minden megoldásnak van egy pozíciója (X) és sebessége (v) a keresési térben. Minden egyes iterációnál a sebesség csökkent egy bizonyos delta értékkel.

$$v^{t+1} = \delta v^t, \quad 0 < \delta < 1, \quad (1)$$

$$V_i^{t+1} = V_i^t e^{-\frac{1}{v^{t+1}}}, \quad (2)$$

ahol t az optimálási folyamat iterációinak száma. A megoldások a keresési térben a Lévy-féle repülés segítségével mozognak L és a megoldás frissítése a következő egyenlet szerint történik:

$$X_i^{t+1} = L(X_i^t - V_i^t), \quad (i = 1, 2, \dots, n), \quad (3)$$

ahol i a megoldáskomponensek indexe, és n az össz változószám (3). Az átlagos érték a legjobb X_{first}^t első,

* doktorandusz, Miskolci Egyetem Energetikai és Vegyipari Gépészeti Intézet, Gépészmérnöki Tanszék, Mérnöki Kar, Kufai Egyetem, Irak

** adjunktus, Szerkezeti és Vizsgadálkodási Tanszék, Mérnöki Kar, Kufai Egyetem, Irak

*** professzor, Miskolci Egyetem Energetikai és Vegyipari Gépészeti Intézet

a közepesen legjobb X_{middle}^t középső és legrosszabb megoldás X_{end}^t vége szintén hatással lehet a megoldás frissítési folyamatára:

$$X_i^{t+1} = \frac{X_{first}^t + X_{middle}^t + X_{end}^t}{3} \quad (4)$$

$$X_{ij}^{t+1} = X_{ij}^t - V_{ij}^t e^{\frac{-1}{v^{t+1}}} + L(X_i^t - V_{ij}^t) - \frac{X_{first}^t + X_{middle}^t + X_{end}^t}{3}, \quad (j = 1, 2, \dots, m), \quad (5)$$

ahol m a változók száma a javasolt megoldásnál, a pszeudokód pedig az 1. kódlistán látható.

Az (1)-(4) egyenletek kombinációja adja a keresést, az FO algoritmus keresőmotorját:

1. kódlista Az FO algoritmus pszeudokódja

A probléma paramétereinek meghatározása (változók száma, célfüggvény, feltételek)
 Az algoritmus paramétereinek meghatározása (populáció mérete, maximális iteráció, sebességcsökkentési együttható, csillapítás).
 A populáció véletlenszerű pozícióinak és sebességeinek inicializálása
 A legjobb költség inicializálása
 Ismétlés 1-től a maximális iterációszámig
 Új megoldás meghatározása
 Ismétlés 1-től a populáció számáig
 Az (5) egyenlet segítségével számítsuk ki az új megoldás új pozícióját.
 Megállás, ha a maximális populációszámot elértük
 A régi megoldás egyesítése az új megoldással
 A megoldások rendezése
 Válassza ki a populáció első megoldását
 Válassza ki a populáció közepén lévő megoldást
 Válassza ki a populáció utolsó megoldását
 A rendezett csoport első megoldása a legjobb megoldás
 A legjobb megoldás költsége a legjobb költség.
 A legjobb költség frissítése
 Leállítás az iterációk maximális számának elérésekor

A FO algoritmus forráskódját a MATLAB 2016 használatával fejlesztették ki, és a következő helyről tölthető le: https://www.mathworks.com/matlabcentral/fileexchange/103970-fertilization-optimization-algorithm-fo/?s_tid=mlc_lp_leaf

Az alkalmazott tesztfüggvények a következők:

Magasan kondicionált elliptikus függvény

$$f_1(\mathbf{x}) = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} x_i^2$$

Szivar függvény

$$f_2(\mathbf{x}) = x_1^2 + 10^6 \sum_{i=2}^D x_i^2$$

Diszkosz függvény

$$f_3(\mathbf{x}) = 10^6 x_1^2 + \sum_{i=2}^D x_i^2$$

Rosenbrock-féle függvény

$$f_4(\mathbf{x}) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$$

Ackley-féle függvény

$$f_5(\mathbf{x}) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e$$

Weierstrass függvény

$$f_6(\mathbf{x}) = \sum_{i=1}^D \left(\sum_{k=0}^{kmax} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - D \sum_{k=0}^{kmax} [a^k \cos(2\pi b^k \cdot 0.5)]$$

$a = 0.5, b = 3, kmax = 20$

Griewank-féle függvény

$$f_7(\mathbf{x}) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

Rastrigin-féle függvény

$$f_8(\mathbf{x}) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$$

Módosított Schwefel-féle függvény

$$f_9(\mathbf{x}) = 418.9829 \times D - \sum_{i=1}^D g(z_i), z_i = x_i + 4.209687462275036e + 002$$

$$g(z_i) = \begin{cases} z_i \sin(|z_i|^{1/2}) & \text{if } |z_i| \leq 500 \\ (500 - \text{mod}(z_i, 500)) \sin\left(\sqrt{|500 - \text{mod}(z_i, 500)|}\right) - \frac{(z_i - 500)^2}{10000D} & \text{if } z_i > 500 \\ (\text{mod}(|z_i|, 500) - 500) \sin\left(\sqrt{|\text{mod}(|z_i|, 500) - 500|}\right) - \frac{(z_i + 500)^2}{10000D} & \text{if } z_i < -500 \end{cases}$$

Katsuura függvény

$$f_{10}(\mathbf{x}) = \frac{10}{D^2} \prod_{i=1}^D \left(1 + i \sum_{j=1}^{32} \frac{|2^j x_i - \text{round}(2^j x_i)|}{2^j} \right)^{\frac{10}{D^{1.2}}} - \frac{10}{D^2}$$

HappyCat függvény

$$f_{11}(\mathbf{x}) = \left| \sum_{i=1}^D x_i^2 - D \right|^{1/4} + \left(0.5 \sum_{i=1}^D x_i^2 + \sum_{i=1}^D x_i \right) / D + 0.5$$

HGBatFunction

$$f_{12}(\mathbf{x}) = \left| \left(\sum_{i=1}^D x_i^2 \right)^2 - \left(\sum_{i=1}^D x_i \right)^2 \right|^{1/2} + \left(0.5 \sum_{i=1}^D x_i^2 + \sum_{i=1}^D x_i \right) / D + 0.5$$

Bővített Griewank-féle plusz Rosenbrock-féle függvény

$$f_{13}(\mathbf{x}) = f_7(f_4(x_1, x_2)) + f_7(f_4(x_2, x_3)) + \dots + f_7(f_4(x_{D-1}, x_D)) + f_7(f_4(x_D, x_1))$$

Bővített Scaffer-féle F6 függvény

$$g(x, y) = 0.5 + \frac{(\sin^2(\sqrt{x^2 + y^2}) - 0.5)}{(1 + 0.001(x^2 + y^2))^2}$$

$$f_{14}(\mathbf{x}) = g(x_1, x_2) + g(x_2, x_3) + \dots + g(x_{D-1}, x_D) + g(x_D, x_1)$$

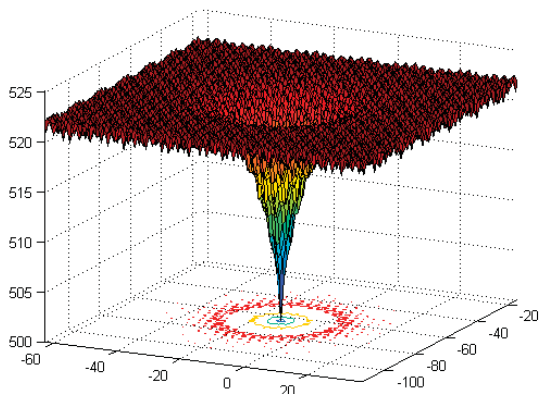
M_i: forgásmátrix: Minden függvényhez és minden alapfüggvényhez különböző forgatási mátrix van hozzárendelve.

Eltolt és elfordított Ackley-féle függvény

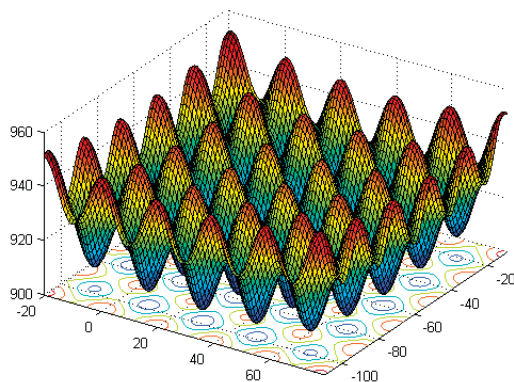
$$F_3(\mathbf{x}) = f_5(\mathbf{M}_3(\mathbf{x} - \mathbf{o}_3)) + F_3^*$$

Eltolt és elfordított Rastrigin-féle függvény

$$F_4(\mathbf{x}) = f_8\left(\mathbf{M}_4\left(\frac{5.12(\mathbf{x} - \mathbf{o}_4)}{100}\right)\right) + F_4^*$$



1. ábra Az eltolt és elfordított Ackley-féle függvény, ábrája



2. ábra. Az eltolt és elfordított Rastrigin-féle függvény ábrája

1. táblázat A CEC2015 időigényes benchmark problémái F1-F9

CEC2015	Típuszám	Leírás	f_{\min}
Unimodális függvények	F1	Forgatott hajlított szivarfüggvény	100
	F2	Forgatott korongfüggvény	200
Egyszerű multimodális függvények	F3	Eltolt és elforgatott Weierstrass-függvény	300
	F4	Eltolt és elforgatott Schwefel-függvény	400
	F5	Eltolt és elforgatott Katsuura-függvény	500
	F6	Eltolt és elforgatott HappyCat függvény	600
	F7	Eltolt és elforgatott HGBat függvény	700
	F8	Eltolt és elforgatott, kibővített Griewank plusz Rosenbrock-függvény	800
	F9	Eltolt és elforgatott, kibővített Scaffer F6 függvény	900

2. EREDMÉNYEK ÉS ÖSSZEHASONLÍTÁS

Azokat a CEC2015 benchmark függvényeket, amelyeket a 1. és 2. táblázatok tartalmaznak, ebben a tanulmányban a teljesítmény vizsgálatára használtuk fel az FO algoritmus teljesítményének vizsgálatához. A futtatási feltételek a CEC2015 kísérletben a következők: változószám 10, populáció méret 10, maximális iterációk száma 1000 és 20 független futtatás minden esetben. Először a FO algoritmust hasonlítottuk össze a

Hibrid részecske raj optimalizáló algoritmus és a FireFly algoritmussal (*HPSOFF*) [10], valamint a Hybrid Firefly és Particle Optimization (*HFPSO*) algoritmussal [11]. A 3. és 4. táblázat mutatja az összehasonlítás eredményeit az átlagos megoldások és a standard *FO*, *HPSOFF* és *HFPSO* közötti összehasonlításban.

A 3. táblázatban az átlagos megoldások és a standard *FO*, *HPSOFF* és *HFPSO* közötti eltérés kerül megadásra.

2. táblázat A CEC2015 időigényes benchmark problémái F10-F15

Típus	No.	Leírás	f_{\min}
Hibrid függvények	F10	Hibrid függvény 1 ($N = 3$)	1000
	F11	Hibrid függvény 2 ($N = 4$)	1100
	F12	Hibrid függvény 3 ($N = 5$)	1200
Összetett függvények	F13	Összetett függvény 1 ($N = 5$)	1300
	F14	Összetett függvény 2 ($N = 3$)	1400
	F15	Összetett függvény 3 ($N = 5$)	1500

3. táblázat Az FO algoritmus standard szórásai értékei a CEC2015 függvényekre a HPSOFF és a HFPSO algoritmusokkal összehasonlítva

No.	HPSOFF	HFPSO	FO
F1	3.4292E+07	6.6375E+06	0E+00
F2	1.2383E+04	1.5696E+04	1.9569E-06
F3	1.5636E+00	1.4189E+00	7.0195E-02
F4	3.0718E+02	3.9950E+02	1.8913E+00
F5	8.2275E-01	5.7466E-01	2.0303E+02
F6	1.4097E-01	1.4584E-01	7.1663E-10
F7	9.3694E-01	2.5433E-01	6.1138E+00
F8	2.8927E+00	4.0866E+00	4.7333E+04
F9	2.3372E-01	2.6387E-01	4.6656E-13
F10	2.9730E+05	3.3036E+05	8.8424E+04
F11	1.9652E+00	2.6814E+00	0E+00
F12	9.5565E+01	1.0221E+02	2.9857E-01
F13	2.5959E+01	2.8341E+01	3.3013E+01
F14	5.0554E+00	5.8221E+00	2.8957E+02
F15	1.8650E+02	1.0398E+02	6.8567E+00

4. táblázat Az FO algoritmus átlagértékei a CEC2015 függvényekre a HPSOFF és a HFPSO algoritmusokkal összehasonlítva

No.	HPSOFF	HFPSO	FO
F1	4.8387E+07	1.3768E+07	7.0974E+07
F2	3.8331E+04	3.8542E+04	1.1254E+10
F3	3.0845E+02	3.0671E+02	3.2049E+02
F4	1.7084E+03	1.3159E+03	4.8685E+02
F5	5.0273E+02	5.0250E+02	2.1652E+03
F6	6.0063E+02	6.0054E+02	1.6116E+06
F7	7.0087E+02	7.0060E+02	7.5666E+02
F8	8.0740E+02	8.0773E+02	1.6292E+05
F9	9.0388E+02	9.0393E+02	1.0413E+03
F10	3.5402E+05	3.3099E+05	6.8481E+04
F11	1.1067E+03	1.1074E+03	1.4195E+03
F12	1.4517E+03	1.3983E+03	1.3391E+03
F13	1.6333E+03	1.6452E+03	1.3908E+03
F14	1.6053E+03	1.6021E+03	1.5594E+04
F15	1.8365E+03	1.9233E+03	2.0528E+03

5. táblázat Az FO algoritmus átlag szórása a CEC2015 függvényekre a PSO, FF és az FFPSO alg. összehasonlítva

No.	PSO	FF	FFPSO	FO
F1	1.3549E+08	2.8945E+08	4.9786E+09	0E+00
F2	1.5114E+04	9.7404E+03	4.6261E+08	1.9569E-06
F3	1.3259E+00	1.2487E+00	1.6124E+00	7.0195E-02
F4	3.5521E+02	3.2112E+02	2.6203E+02	1.8913E+00
F5	6.3611E-01	5.9796E-01	9.3430E-01	2.0303E+02
F6	2.8490E-01	5.8361E-01	1.3580E+00	7.1663E-10
F7	1.8947E+00	5.8077E+00	3.3329E+01	6.1138E+00
F8	2.7690E+01	1.7256E+02	2.7423E+05	4.7333E+04
F9	3.2749E-01	2.3842E-01	1.7883E-01	4.6656E-13
F10	1.9786E+05	6.5054E+05	7.7896E+07	8.8424E+04
F11	2.9153E+00	2.4020E+00	6.7179E+01	0E+00
F12	1.1574E+02	9.1615E+01	4.3894E+02	2.9857E-01
F13	1.9141E+01	2.9519E+01	9.8971E+02	3.3013E+01
F14	4.5254E+00	3.5980E+00	4.2292E+01	2.8957E+02
F15	1.4570E+02	7.4514E+01	1.0567E+02	6.8567E+00

- F16: Rastrigin: $f(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)]$, Range = [-5.12, 5.12], $F_{min} = 0$;
- F17: $f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$, Range = [-2.048, 2.048], $F_{min} = 0$;
- F18: $f(x) = -a \exp\left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d \cos(cx_i)}\right) + a + \exp(1)$, Range = [-32.768, 32.768], $F_{min} = 0$;
- F19: $f(x) = \sum_{i=1}^d \frac{x_i^2}{4000} - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$, Range = [-600, 600], $F_{min} = 0$.

6. táblázat Az FO algoritmus átlagértékei a CEC2015 függvényekre a PSO, FF és az FFPSO algoritmusokkal összehasonlítva

No.	PSO	FF	FFPSO	FO
F1	2.4553E+08	4.3059E+08	1.6287E+10	7.0974E+07
F2	3.8112E+04	3.3304E+04	1.4957E+08	1.1254E+10
F3	3.0779E+02	3.0773E+02	3.1455E+02	3.2049E+02
F4	2.2534E+03	1.5473E+03	3.1120E+03	4.8685E+02
F5	5.0277E+02	5.0293E+02	5.0350E+02	2.1652E+03
F6	6.0089E+02	6.0092E+02	6.0673E+02	1.6116E+06
F7	7.0193E+02	7.0586E+02	8.0586E+02	7.5666E+02
F8	8.1583E+02	8.6344E+02	2.7632E+05	1.6292E+05
F9	9.0391E+02	9.0395E+02	9.0451E+02	1.0413E+03
F10	2.9540E+05	5.3162E+05	5.1186E+07	6.8481E+04
F11	1.1088E+03	1.1080E+03	1.2198E+03	1.4195E+03
F12	1.4620E+03	1.3995E+03	2.1953E+03	1.3391E+03
F13	1.6415E+03	1.6437E+03	3.0005E+03	1.3908E+03
F14	1.6076E+03	1.6111E+03	1.6770E+03	1.5594E+04
F15	1.9149E+03	1.9269E+03	2.1840E+03	2.0528E+03

Az 5. táblázat a szórás összehasonlítását mutatja a FO, a PSO, az FFPSO algoritmus [12] és a FireFly (FF) algoritmus között, míg a 6. táblázat az összehasonlítást mutatja, az átlagos megoldási eredmények összehasonlítását ugyanezen algoritmusok között.

A 7. és 8. táblázat nagyméretű optimálási feladatok számítási eredményeit mutatják az F16 és F17, valamint az F18 és F19 feladatokra.

A FO algoritmus kevésbé hatékony a magas fokú multimodális benchmarkokon, és ez a viselkedés látható a következő példákban is a statisztikai eredményeken. A kísérleti eredmények azt mutatják, hogy a FO algoritmus hatékonyabb a nagy léptékű optimálásnál mint kis léptékben. A viselkedés nagy és kis léptékű problémákon való viselkedés külön tanulmányt igényel, amely irányt mutathat a jövőbeni munkára. Röviden, a FO algoritmus stabil és gyors lehet konvergencia, unimodális optimálási problémákon, valamint a hatékonysága jelentős a nagy léptékű problémáknál.

3. ÖSSZEFOGLALÁS

A megtermékenyítési optimálási algoritmus egy erőteljes biológiailag ihletésű algoritmus, amelyet matematikai optimálási problémáknál vizsgáltunk. Utánozza a kölcsönhatást a spermiumok és a méh között a petesejt megtermékenyítése során. A statisztikai eredmények 19 tesztfüggvényen; CEC2015-ös időigényes tesztfeladat gyűjteményen, benchmarkon került bemutatásra, valamint egymódusú, többmódusú, kis léptékű, és nagy léptékű problémákon mutattuk be a javasolt algoritmus hatékonyságát számos optimáló algoritmussal összehasonlítva. A FO algoritmus vizsgálataiban során az FO algoritmus teljesítménye nagy léptékű problémákon jobbnak bizonyult, mint a kis léptékű problémákon nyújtott teljesítménye. A statisztikai eredmények azt mutatják, hogy a FO algoritmus stabil, kisebb szórással, STD-vel és adott esetben a legjobb megoldással rendelkezik, mint nyolc más versenyképes algoritmus. A FO algoritmus bizonyította, hogy erőteljes az unimodális függvényekre, és ígéretes alkalmazásokat kínál a

7. táblázat Nagyméretű optimalizációs feladat eredményei az F16 és F17 feladatokra

Algoritmus	Függvény	F16	F17
ALO	Best	1.2463E+04	9.6145E+04
	Worst	1.5088E+04	2.6251E+05
	Mean	1.3352E+04	1.2072E+05
	STD	5.7343E+02	2.5922E+04
BOA	Best	0.0000E+00	9.9873E+02
	Worst	1.1460E-10	9.9894E+02
	Mean	3.3526E-12	9.9885E+02
	STD	1.5967E-11	4.6350E-02
GWO	Best	6.3429E+03	3.3616E+03
	Worst	7.4800E+03	7.2573E+03
	Mean	6.9049E+03	4.8800E+03
	STD	2.6947E+02	8.9692E+02
PSO	Best	1.4774E+04	4.2109E+05
	Worst	1.7302E+04	4.6887E+05
	Mean	1.6192E+04	4.5065E+05
	STD	6.4158E+02	1.0330E+04
SCA	Best	5.1110E+02	1.2307E+05
	Worst	4.6834E+03	3.0827E+05
	Mean	1.7431E+03	2.3237E+05
	STD	9.1819E+02	3.8524E+04
DDAO	Best	2.2573E-01	9.9897E+02
	Worst	6.4277E+03	1.4620E+03
	Mean	5.7423E+02	1.0382E+03
	STD	1.0834E+03	8.9853E+01
BA	Best	1.2667E+04	5.4067E+04
	Worst	1.7947E+04	4.3425E+05
	Mean	1.4585E+04	1.8689E+05
	STD	1.3142E+03	7.7892E+04
TSA	Best	6.3653E+03	1.2015E+04
	Worst	1.4767E+04	5.2552E+04
	Mean	1.0217E+04	2.7948E+04
	STD	2.1618E+03	9.2913E+03
FO	Best	0.0000E+00	9.9890E+02
	Worst	0.0000E+00	9.9899E+02
	Mean	0.0000E+00	9.9896E+02
	STD	0.0000E+00	2.2892E-02

8. táblázat Nagyméretű optimalizációs feladat eredményei az F18 és F19 feladatokra

Algoritmus	Függvény	F18	F19
ALO	Best	1.9469E+01	9.8149E+03
	Worst	2.0307E+01	1.4948E+04
	Mean	1.9750E+01	1.1345E+04
	STD	2.4107E-01	1.4212E+03
BOA	Best	1.2957E-08	2.0416E-06
	Worst	6.1917E-07	5.9821E-04
	Mean	5.9824E-08	5.5767E-05
	STD	9.7536E-08	1.2535E-04
GWO	Best	9.1437E+00	6.8521E+02
	Worst	1.2941E+01	1.4847E+03
	Mean	1.0128E+01	9.7278E+02
	STD	9.8562E-01	1.8400E+02
PSO	Best	1.7001E+01	8.1418E+02
	Worst	1.7812E+01	1.0013E+03
	Mean	1.7356E+01	9.2068E+02
	STD	1.6680E-01	4.4017E+01
SCA	Best	7.0434E+00	2.0794E+03
	Worst	1.9671E+01	1.2949E+04
	Mean	1.6601E+01	8.3566E+03
	STD	3.2924E+00	2.5246E+03
DDAO	Best	2.4380E-02	1.3322E+00
	Worst	9.5690E+00	6.0214E+02
	Mean	3.3104E+00	8.4837E+01
	STD	2.1546E+00	1.5768E+02
BA	Best	1.9362E+01	1.0246E+04
	Worst	2.1148E+01	2.8739E+04
	Mean	2.0312E+01	1.7242E+04
	STD	4.7706E-01	4.9752E+03
TSA	Best	6.4128E+00	3.1366E+02
	Worst	1.2601E+01	3.3672E+03
	Mean	9.0431E+00	1.0870E+03
	STD	1.5954E+00	5.1949E+02
FO	Best	8.9617E-13	0.0000E+00
	Worst	2.4986E-07	7.2283E-07
	Mean	5.0366E-09	1.4257E-08
	STD	3.4971E-08	1.0120E-07

következő területeken: folytonosan differenciálható célfüggvények és nagyméretű optimalizálás. A FO algoritmus gyors és egyszerű, és képes hatékonyan átugorni a keresési tér lokális pontjait, és továbbléphet a globális ponthoz.

4. KÖSZÖNETNYILVÁNÍTÁS

A bemutatott kutató munka részben a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal (NKFIH) támogatásával a K 134358 nyilvántartási számú projekt keretében valósul meg.

5. IRODALOM

- [1] H. N. Ghafil and K. Jármái, "Research and application of industrial robot manipulators in vehicle and automotive engineering, a survey," in *Vehicle and Automotive Engineering 2, Lecture Notes in Mechanical Engineering*, K. Jármái and B. Bolló, Eds, Springer, 2018, pp. 611–623.
- [2] H. N. Ghafil and A. H. Mohammed, "A virtual reality environment for 5-DOF robot manipulator based on XNA framework," *Int. J. Computer Appl.*, vol. 113, no. 3, pp. 33–37, 2015.
- [3] H. N. Ghafil and K. Jármái, "Kinematic-based structural optimization of robots," *Pollack Period.*, vol. 14, no. 3, pp. 213–222, 2019.

- [4] E. Figueiredo, M. Macedo, H. V. Siqueira, C. J. Santana, Jr, A. Gokhale, and C. J. A. Bastos-Filho, "Swarm intelligence for clustering- A systematic review with new perspectives on data mining," *Eng. Appl. Artif. Intell.*, vol. 82, pp. 313–329, 2019.
- [5] P. Sulek and T. Kinczer, "Expert control system of shipping operation on the Gabcikovo project," *Pollack Period.*, vol. 14, no. 1, pp. 139–150, 2019.
- [6] S. Alsamia, D. S. Ibrahim, and H. N. Ghafil, "Optimization of drilling performance using various metaheuristics," *Pollack Period.*, vol. 16, no. 2, pp. 80–85, 2021.
- [7] H. Ghafil and K. Jármai, "Comparative study of particles warm optimization and artificial bee colony algorithms," in *Multiscience XXXII. MicroCAD International Multidisciplinary Scientific Conference*, Miskolc-Egyetemváros, Hungary, Sep. 5–6, 2018, pp. 1–6.
- [8] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014.
- [9] J. J. Liang, B. Y. Qu, P. N. Suganthan, and Q. Chen, "Problem definitions and evaluation criteria for the CEC2015 competition on learning-based real-parameters in gle objective optimization," Tech. Rep. 201411A, Comput. Intell. Lab. Zhengzhou Univ. Zhengzhou China Nanyang Technol. Univ. Singapore, vol. 29, pp. 625–640, 2014.
- [10] S. Arunachalam, T. A. Bhomila, and M. R. Babu, "Hybrid particles warm optimization algorithm and firefly algorithm based combined economic and emission dispatch including valve point effect," in *International Conference on Swarm, Evolutionary, and Memetic Computing, Swarm, volutionary, and Memetic Computing. SEMCCO 2014. Lecture Notes in Computer Science*, vol. 8947, B. Panigrahi, P. Suganthan, and S. Das, Eds, Springer, 2014, pp. 647–660.
- [11] P. Kora and K. S. R. Krishna, "Hybrid firefly and particle swarm optimization algorithm for the detection of bundle branch block," *Int. J. Cardiovasc. Acad.*, vol. 2, no. 1, pp. 44–48, 2016.
- [12] I. B. Aydilek, "A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems," *Appl. Soft Comput.*, vol. 66, pp. 232–249, 2018.
- [13] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, 2015.
- [14] M. Pelikan, D. E. Goldberg, and E. Cant_u-Paz, "BOA: The Bayesian optimization algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference*, vol. 1, Orlando Florida, Jul. 13, 1999, pp. 525–532.
- [15] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-Based Syst.*, vol. 96, pp. 120–133, 2016.
- [16] H. N. Ghafil and K. Jármai, "Dynamic differential annealed optimization: New metaheuristic optimization algorithm for engineering applications," *Appl. Soft Comput.*, 2020, Paper no. 106392.
- [17] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization Studies in Computational Intelligence*, vol. 284, J. R. Gonz_alez, D. A. Pelta, C. Cruz, G. Terrazas, and N. Krasnogor, Eds, Berlin, Heidelberg: Springer, 2010, pp. 65–74.
- [18] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, "Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization," *Eng. Appl. Artif. Intell.*, vol. 90, 2020, Paper no. 103541.