A RÉSZECSKE CSOPORT ÉS A MESTERSÉGES MÉHCSALÁD MÓDSZEREK ÖSSZEHASONLÍTÓ VIZSGÁLATA

COMPARATIVE STUDY OF PARTICLE SWARM OPTIMIZATION AND ARTIFICIAL BEE COLONY ALGORITHMS

Hazim Nasir Ghafil, Dr. Jármai Károly***

ÖSSZEFOGLALÁS

Ebben a munkában a részecske csoport optimálás és a méhcsalád algoritmusai közötti összehasonlítást mutatjuk be különböző vizsgálati módszerekkel. Minden algoritmust részletesen ismertetünk, és bemutatjuk a matematikai modelljüket. Megállapítást nyert, hogy a részecske csoport optimálás jobb, mint a mesterséges méhcsalád, módszer és egy speciális tesztfüggvény esetében a mesterséges méhcsalád nem tudott megfelelő megoldást találni.

ABSTRACT

In this work greedy comparison between particle swarm optimization and artificial bee colony algorithms was made using different test functions. Each algorithm was explained in detail, and the mathematical model behind the algorithms has been presented. It is found that particle swarm optimization is better than artificial bee colony and for a specific test function, artificial bee had failed to find a feasible solution.

1. BEVEZETÉS

A PSO egy csoport intelligencián alapuló optimáló algoritmus; az optimáló algoritmusok egy osztályába tartozik, amelyet meta-heurisztikusnak neveznek. A PSO utánozza az állatok, mint a halak és a madarak csoport viselkedését, és ez egy egyszerű, hatékony optimáló módszerhez vezet. Sikeresen alkalmazták a tudomány különböző területein, mint például a gépi tanulásban, a képfeldolgozásban, az adatbányászatban, a robotikában és sok más területen. A PSO-t 1995-ben Russell Eberhart és James Kennedy vezette be [1]. Egy olyan modell kidolgozásával, amely leírja az állatok csoport viselkedését, mint a madarak és a halak raja. 1995 óta a PSO az egyik leghatékonyabb és legnépszerűbb algoritmus lett a különböző területek különböző optimálási problémáinak megoldására. Ennek a csoport intelligenciának a legfontosabb pontja az egyedek közötti együttműködés. Az egyéni intelligencia hatékonyabbá válik, ha együttműködik egy másik egyeddel [2]. 2005-ben [3] bevezetésre került a mesterséges méhcsalád ABC nevű csoport intelligencia optimáló algoritmus. Ez szintén egy metaurisztikus algoritmus, amellyel hatékonyan lehet megoldani a többdimenziós optimalizálási problémákat. A mézelő méh kolónia táplálkozási magatartását utánozza a [4] által javasolt modell alapján. Mesterséges méh kolónia úgy működik, hogy vannak méhek (egyedek), akik gazdag élelmiszerforrást keresnek (legjobb megoldás) a kaptár szomszédságában (keresési hely). Minden egyed egy lehetséges megoldás, és csak egy konkrét megoldáshoz kapcsolódik a keresési térben.

2. A PSO ALGORITMUS

Tekintsük az 1. ábrát, amely a PSO mögötti matematikai modellt mutatja. Hasonlóképpen hivatkozhatunk az csoport egyedeire és a csoport egészére, mivel minden részecske a megoldandó optimalizálási probléma egy potenciális megoldása. A keresési terület korlátozza az összes lehetséges megoldást a problémára, és a részecskéknek a legjobb helyzetbe kell kerülniük (a legjobb megoldás az optimálási problémára) a térben. Egy adott részecske pozícióját és sebességét a következők jelölik.

$$x_k(t) \in x$$
$$v_k(t) \in x$$

ahol k a részecske indexe a csoportban, és x a keresési tér, míg (t) egy diszkrét időszakasz, és az algoritmus iterációs számát mutatja. A sebesség- és pozícióvektorok ugyanabban a térben vannak, ugyanolyan dimenzióval. Tekintsük az 1. ábrán látható rendszert, amely egy egyszerű matematikai modellt mutat be, amely leírja a PSO-t. Ahol $x_k(t)$ a

* *PhD hallgató, Miskolci Egyetem, Miskolc, H-3515 Miskolc, Egyetemváros, vegyhnr@uni-miskolc.hu*
** *Professzor Dr., Miskolci Egyetem, Miskolc, H-3515 Miskolc, Egyetemváros*

részecske aktuális helyzete, és az új pozícióba kell eljutnia $x_k(t+1)$. Minden részecske rendelkezik saját tapasztalattal és saját memóriával a legjobb helyzetére vonatkozólag, ezt nevezzük a saját legjobb pozíciójának a *k*-adik részecskénél $p_k(t)$. A részecske elmozdul a jelenlegi helyzetéből $v_k(t)$ sebességgel és iránnyal. A részecskék nem egyedülállóak, egymással kommunikálnak és egymással kölcsönhatásba lépnek, és megosztják személyes tapasztalataikat, hogy megismerjék és eldöntsék, mi a legjobb pozíció a többi egyed tapasztalatai alapján, vagyis amit globális legjobbnak nevezünk és $G(t)$-vel jelölünk. Az alábbi mennyiségek azonosíthatók az 1. ábrán:
Az aktuális helyet és a személyes legjobbat összekötő vektor értéke:

$$p_x = p_k(t) - x_k(t) \qquad \text{.......(1)}$$

Továbbá, az aktuális helyet a globális legjobbhoz kapcsoló vektor értéke:

$$g_x = G(t) - x_k(t) \qquad (2)$$

A részecskének az új pozícióba történő mozgását a következő egyenletek adják meg:

$$x_k(t+1) = x_k(t) + v_k(t+1) \qquad (3)$$

$$v_k(t+1) = w * v_k(t) + C_1(p_k(t) - x_k(t)) + C_2(G(t) - x_k(t)) \qquad (4)$$

A (4) egyenletben a $v_k(t+1)$ vektor a három részmozgás m_1, m_2 és m_3 összegzése, mellyel a részecske mozog $v_k(t)$, a px és a gx vektorok irányában:

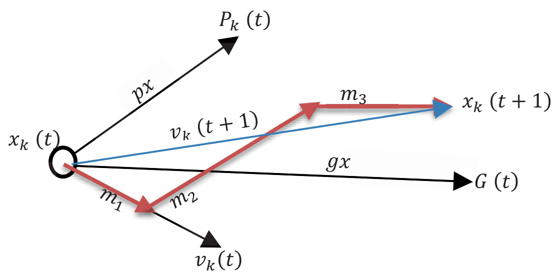
$$v_k(t+1) = m_1 + m_2 + m_3 \qquad (5)$$

ahol

$$m_1 = w * v_k(t) \qquad (6)$$

$$m_2 = C_1 px = C_1(p_k(t) - x_k(t)) \qquad (7)$$

$$m_3 = C_2 gx = C_2(G(t) - x_k(t)) \qquad (8)$$



1. ábra. A PSO matematikai modelljét ábrázoló vázlatos diagram

A (3) és (4) egyenletek teljesen leírják a PSO matematikai modelljét. Ezek az egyenletek azonban egyszerű matematikai modellek a standard PSO mögött, és néhány feltétel szükséges az egyenletek teljesítéséhez. A standard PSO a következő:

$$x_{kj}(t+1) = x_{kj}(t) + v_{kj}(t+1) \qquad (9)$$

$$v_{kj}(t+1) = w * v_{kj}(t) + r_1 C_1(p_{kj}(t) - x_{kj}(t)) + r_2 C_2(G_j(t) - x_{kj}(t)) \qquad (10)$$

$v_{kj}(t+1)$: jelöli a *k*-dik részecske *j*-irányú sebesség komponensét a (*t*+1) időlépésben,
r_1, r_2: véletlen számok, egyenletesen elosztva a 0 -1 intervallumon,
C_1, C_2: gyorsulási együtthatók,
$w * v_{kj}(t)$: inercia kifejezés,
w: inercia együttható,
$r_1 C_1(p_{kj}(t) - x_{kj}(t))$: kognitív komponens,
$r_2 C_2(G_j(t) - x_{kj}(t))$: szociális komponens.

A (9) és (10) egyenlet az a két szabály, amelyeket minden részecskének követnie kell a csoportban, és ez a csoportintelligencia pontos jelentése. Ezen szabályoknak a meghatározásával a PSO minden iterációja során, az egyes részecskék sebessége és helyzete ezen egyszerű mechanizmus szerint frissül.

3. MESTERSÉGES MÉHCSALÁD

Az algoritmus általános szerkezete a következő:

Cserkész méhek fázisa (inicializálás),
Ismétlés
    Foglalkoztatott méhek,
    Kereső méhek,
    Cserkész méhek,
    Tárolja a legjobb megoldást az aktuális nyomvonalban,
Amíg (konvergencia feltétel, a ciklusok maximális száma),

Az algoritmus minden egyes részének saját alacsony szintű szerkezete van, és ezek a globális szintet befolyásolják egymás közötti kölcsönhatásukkal. Kezdetben minden méh cserkész és véletlenszerűen új megoldásokat keres.
Tegyük fel, hogy x a véletlenszerű megoldások vektora, amelyet eredetileg a cserkész méhek adtak vissza.

$$x = (x_1, x_2, x_i ..., x_{n-1}, x_n) \tag{11}$$

ahol $n \in R^n$, i=1...n

3.1 Foglalkoztatott méhek szekció

A foglalkoztatott méhek kiaknázzák az élelmiszerforrást, és visszaadják az információt, és elhagyják a kimerült forrást. Az ABC-nél ezt véletlenszerűen kell tennünk. [5] a következő képletet javasolta:

$$v_i = x_i + \phi_i (x_i - x_k) \tag{12}$$

ahol v_i az új megoldás vektor, ϕ_i véletlen szám $[-1,1]$ között. k szintén véletlen szám, ami a megoldás vektor különböző véletlen sorrendjét képviseli.

3.2 Kereső méhek szekció

A böngésző méhek a valószínűséget használják, amely a fitnesz érték függvénye, a legjobb megoldás kiválasztásához. Rulett kerék kiválasztási módszert alkalmazza [6], amely fitnesz érték alapú kiválasztási technika. A megoldás valószínűsége (P_i) a következő értékű:

$$P_i = \frac{f_i}{\sum_{i=1}^{pop} f_i} \tag{13}$$

$$f_i = \left\{ \begin{array}{ll} \dfrac{1}{1+O_i} & O_i \geq 0 \\ 1 + abs(O_i) & O_i < 0 \end{array} \right\} \tag{14}$$

ahol f_i a fitnesz értéke az O_i célfüggvénynek.

3.3 Cserkész méhek szekció

Az algoritmus elején az összes méh cserkész, majd programfutás közben átalakul a foglalkoztatottra, vagy keresőre. A foglalkoztatott méheknek, akiknek a pozíciója (megoldása) nem változik egy adott időpont után, el kell hagynia pozícióját és át kell alakítania azt a cserkészeknek. Az elhagyási kritérium, amelyet határellenőrzésnek nevezünk, nagyon fontos a helyi minimumból való kiugráshoz, és továbbra is keresni kell az optimálási probléma globális minimumát.

1. táblázat. Teszt függvények az optimáló algoritmusokhoz

Teszt függvény	Képlete	Megengedett tartomány, globális optimum
Sphere függvény	$f(x) = \sum_{i=1}^{n} x_i^2$	$-5.12 \leq x_i \leq 5.12$ $x_i = 0, i = 1, \dots, n$ $f(x)$=0
Rosenbrock's valley függvény	$f(x) = \sum_{i=1}^{n-1} [100 \times (x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	$-2.048 \leq x_i \leq 2.048$ $x_i = 1, i = 1, \dots, n$ $f(x)$=0
Rastrigin's függvény	$f(x) = 10 \times n + \sum_{i=1}^{n} [x_i^2 - 10 \times \cos(2\pi x_i)]$	$-5.12 \leq x_i \leq 5.12$ $x_i = 0, i = 1, \dots, n$ $f(x)$=0
Schwefel's függvény	$f(x) = 0{,}5 + \frac{sin^2(x_1^2 - x_2^2) - 0{,}5}{[1 + 0{,}001 \times (x_1^2 + x_2^2)]^2}$	$-10 \leq x_i \leq 10$ $x_i = 0, i = 1, 2$ $f(x)$=0
Griewangk's függvény	$f(x) = \frac{1}{4000} \times \sum_{i=1}^{n} x_i^2 - \prod_{\mathrm{i}=1}^{\mathrm{n}} \cos\left(\frac{x_\mathrm{i}}{\sqrt{\mathrm{i}}}\right) + 1$	$-600 \leq x_i \leq 600$ $x_i = 0, i = 1, \dots, n$ $f(x)$=0

2. táblázat. PSO és ABC algoritmusok futási eredményei különböző teszt függvényeknél

Teszt függvény	Globális megoldás	Algoritmus	Legjobb megoldás	Fő érték	Szórás
Sphere függvény	0	PSO	0	0.0030	0.0365
		ABC	0	0.0112	0.0763
Rosenbrock's valley függvény	0	PSO	0	0.0166	0.0621
		ABC	0	0.0252	0.0453
Rastrigin függvény	0	PSO	0	0.1556	1.0164
		ABC	0	0.3091	1.5653
Schwefel függvény	-837.9658	PSO	-719.5274	-717.7633	12.9889
		ABC	$-7.86*10^{83}$	$-6.89*10^{81}$	$6.06*10^{82}$
Griewangk függvény	0	PSO	0.0049	0.0347	0.1426
		ABC	0.0025	0.0592	0.1973

4. TESZT FÜGGVÉNYEK

Különböző vizsgálati függvényeket alkalmaztunk a részecske csoport optimálás és a mesterséges méh kolónia összehasonlítására (1. táblázat). A 2. táblázat a két algoritmus összehasonlítását mutatja be az öt különböző tesztfüggvényen [7]. A populációszám költségei és a költségek szórása egyértelműen azt mutatja, hogy az előny a PSO algoritmusánál van. Azt is megállapítottuk, hogy az ABC algoritmus nem közelíti meg a megoldást a Schwefel függvény használatakor.

5. ÖSSZEFOGLALÁS

Az optimalizálási technikák fontos szerepet játszanak abban, hogy számos alternatívából megtalálják a legjobb megoldást. Számos optimálási technika áll rendelkezésre. Napjainkban a metaheurisztikus algoritmusok népszerűek. Kiválasztottuk a részecske csoport PSO és a Bee kolónia algoritmusokat. Mindkettő a csoport intelligenciát használja. Összehasonlításunkban azt találtuk, hogy a részecske csoport algoritmus jobb alkalmazhatósággal bír, minden esetben jó megoldást adott.

KÖSZÖNETNYILVÁNÍTÁS

A cikkben ismertetett kutató munka az EFOP-3.6.1-16-2016-00011 jelű „Fiatalodó és Megújuló Egyetem – Innovatív Tudásváros – a Miskolci Egyetem intelligens szakosodást szolgáló intézményi fejlesztése" projekt részeként – a Széchenyi 2020 keretében – az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

IRODALOM

[1] Eberhart R, Kennedy J (1995) A new optimizer ustiing particle swarm theory. In: Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on, IEEE, pp. 39-43.

[2] Beni G, Wang J (1993) Swarm intelligence in cellular robotic systems. In: Robots and Biological Systems: Towards a New Bionics? Springer, pp. 703-712.

[3] Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department,

[4] Tereshko V, Loengarov A (2005) Collective decision making in honey-bee foraging dynamics. Computing and Information Systems 9 (3):1

[5] Karaboga D (2010) Artificial bee colony algorithm. Scholarpedia 5: 6915

[6] Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. Machine learning 3 (2): 95-99.

[7] Marcin Molga, Czesław Smutnicki (2005) Test functions for optimization needs, 43 p. http://www.robertmarks.org/Classes/ENGR5358/Papers/functions.pdf (accessed 14 Aug.2018)