**PERBANDINGAN CHAOTIC ELEPHANT HERDING OPTIMIZATION DAN ELEPHANT SWARM WATER SEARCH ALGORITHM**

**Ai Nurhayati**

Program Studi Teknik Industri, Sekolah Tinggi Teknologi Bandung

Jln. Soekarno-Hatta No.378 Bandung 40235

[ai.nurhayati@sttbandung.ac.id](mailto:ai.nurhayati@sttbandung.ac.idcom)

**Abstract**

The purpose of this study was to compare the effectiveness level of two algorithms developed by Elephant Herd Optimization (EHO), namely Chaotic Elephant Herding Optimization (CEHO) and Elephant Swarm Water Search (ESWS). The method used to measure the performance of the two algorithms is the testing method through the 2018a version of the MATLAB software which is carried out on six mathematical equations as a comparative function to measure the level of performance effectiveness of the two algorithms, namely the Rosenbrock, Griewank, Ackley, Shwefel, Elliptic and Rastrigin function equations. The results show that the ESWS algorithm is more effective than the CEHO algorithm for finding the global minimum point in the optimization method problem. The ESWS algorithm appears to excel at reaching the minimum for the Elliptic, Schwefel, Ackley, Rosenbrock, Griewank and Rastrigin functions.

**Keywords** : *chaotic elephant herding optimization* *(CEHO),* *elephant swarm water search (ESWS)*

**Abstrak**

Tujuan dari penelitian ini adalah untuk membandingkan tingkat efektivitas dari dua algoritma hasil pengembangan *Elephant Herd Optimization* (EHO) yaitu *Chaotic Elephant Herding Optimization* (CEHO) dan *Elephant Swarm Water Search* (ESWS). Metode yang digunakan untuk mengukur kinerja dari kedua algoritma adalah metode pengujian melalui *software* MATLAB versi 2018a yang dilakukan terhadap enam persamaan matematika sebagai fungsi pembanding untuk mengukur tingkat efektivitas kinerja dari kedua algoritma yaitu persamaan fungsi *Rosenbrock, Griewank, Ackley, Shwefel, Elliptic* dan *Rastrigin*. Hasil penelitian menunjukkan bahwa algoritma ESWS lebih efektif daripada algoritma CEHO untuk mencari titik minimum global dalam persoalan metode optimasi. Algoritma ESWS tampak unggul dalam mencapai titik minimum untuk fungsi *Elliptic, Schwefel, Ackley, Rosenbrock, Griewank* dan *Rastrigin*.

**Kata kunci** : *chaotic elephant herding optimization* *(CEHO),* *elephant swarm water search (ESWS)*

1. **PENDAHULUAN**

Pada zaman sekarang ini ada banyak macam algoritma yang secara luas dikembangkan oleh para peneliti [1]. Beberapa algoritma sangat menunjang dalam pengembangan dan pencarian solusi untuk metode optimisasi [2]. Kemajuan yang luar biasa dibidang teknik optimisasi telah menuntun pada perkembangan pesat dalam penggunaannya pada bidang perencanaan industri, penjadwalan, membuat keputusan dan masalah kompleks lainnya [3].

*Swarm intelligence metaheuristics* merupakan simulasi kelompok dari organisme-organisme yang ada dialam seperti *flock of birds and fish, colony of bees and ants, groups of bats and cuckoo birds*, dan sebagainya [4].

Algoritma yang terinspirasi dari alam terutama *swarm intelligence algorithms* telah terbukti sangat efisien [5].

Pada penelitian ini akan lebih fokus pada algoritma *Elephant Herding Optimization* (EHO) atau algoritma optimisasi kawanan gajah.

Tujuan dari penelitian ini adalah untuk membandingkan tingkat efektivitas dari dua algoritma hasil pengembangan EHO yaitu *Chaotic Elephant Herding Optimization* (CEHO) dan *Elephant Swarm Water Search* (ESWS).

1. **TINJAUAN PUSTAKA DAN TEORI**
2. **Tinjauan Pustaka**

Gajah Asia dan Afrika adalah dua spesies yang secara tradisional disadari merupakan salah satu dari jenis mamalia yang terbesar di planet bumi [6].

Gajah Afrika memiliki tinggi tiga sampai empat meter dan berat 4000-7000 kilogram. Gajah Asia memiliki tinggi dua sampai tiga setengah meter dan berat 3000-5000 kilogram [6].

Di habitat alaminya, gajah-gajah liar hidup dalam kawanan yang terdiri atas 3 sampai 35 gajah. Jumlah gajah dalam suatu kawanan dapat bervariasi bergantung pada musim dan ketersediaan makanan, air dan sebagainya [6].

Gajah adalah makhluk sosial yang mana gajah betina dan anak-anak gajah hidup bersama dalam suatu kawanan gajah. Gajah-gajah yang berbeda klan ada didalam kelompok dimana gajah betina yang paling tua memimpinnya. Sebuah klan tersusun dari satu atau lebih gajah betina bersama anak-anaknya. Gajah-gajah betina hidup dalam kelompok-kelompok gajah. Gajah-gajah jantan biasanya hidup secara terpisah dari kelompok tersebut. Gajah-gajah jantan biasanya secara naluri dan secara alamiah akan meninggalkan anggota keluarganya masing-masing setelah mereka tumbuh menjadi gajah tinggi, besar dan cukup dewasa dalam kelompoknya. Walaupun terpisah tempatnya, melalui getaran frekuensi rendah, gajah-gajah jantan tetap kontak dengan gajah-gajah betina dalam klannya. Sifat kelakuan gajah tersebut dapat ditiru untuk formasi metode optimisasi global, dua operator yang dihasilkan dari sifat kawanan gajah [7].

Algoritma *Elephant Herding Optimization* (EHO) merupakan salah satu dari algoritma *swarm intelligence* yang terbaru yang diusulkan pada tahun 2016.

Untuk meniru dan mengimplementasikan sifat kelakuan gajah yang sesuai dengan masalah optimisasi, prinsip dari *Elephant Herding Optimization* (EHO) secara sederhana dibagi ke dalam tiga aturan utama, yaitu [3]:

1. Populasi memiliki jumlah beberapa klan dan setiap klan memiliki jumlah tetap dari gajah jantan dan gajah betina.
2. Jumlah tetap dari gajah jantan hidup terpisah dengan meninggalkan klan mereka.
3. Seekor gajah betina pemimpin adalah pemimpin dari klan-klan.

Gambar 1 State Of The Art

Pada paragraph akhir bagian ini memuat *state of the art* (kebaharuan) dari penelitian ini.

1. **Algoritma *Elephant Herd Optimization* (EHO)**

Populasi gajah dibagi ke dalam beberapa klan, misalnya k klan. Awalnya, solusi dimensi-D dihasilkan secara acak di dalam ruang pencarian dengan batas bawah xmin dan batas atas xmax yang dituliskan dalam rumus persamaan (1).

(1)

dengan r adalah bilangan acak yang terdistribusi seragam di dalam rentang [0, 1].

Setiap klan gajah selalu dipimpin oleh gajah betina. Setiap anggota j dari klan i akan bergerak dengan mempertimbangkan matriarki ci yang dapat dituliskan dalam bentuk rumus Persamaan (2).

(2)

Keterangan:

= posisi baru dari gajah j dalam klan i

= posisi lamanya

= solusi terbaik dari klan ci

= parameter algoritma, suatu faktor skala yang menentukan pengaruh dari matriarki

= suatu bilangan acak yang digunakan secara potensial meningkatkan keragaman populasi dalam tahap akhir algoritma.

Posisi dari matriarki misalnya gajah terbaik dalam suatu klan yang diperbaharui dengan rumus Persamaan (3).

(3)

= parameter algoritma kedua yang mengendalikan pengaruh dari

didefinisikan dalam rumus Persamaan (4).

(4)

dengan adalah dimensi ke-d dimana D adalah dimensi total dari ruang dan adalah jumlah gajah di dalam klan i.

Di dalam setiap *generation* dan di dalam setiap klan jumlah gajah yang pindah untuk hidup jauh dari klannya. Gajah dengan nilai kelayakan terburuk dipilih pindah dan posisi baru mereka dihitung oleh Persamaan (5).

(5)

= batas bawah dari ruang pencarian

= batas atas dari ruang pencarian

= distribusi seragam bentuk bilangan acak [8].

1. **Algoritma *Chaotic Elephant Herding Optimization* (CEHO)**

Algoritma *swarm intelligence* sudah terbukti sangat ampuh untuk menemukan solusi terdepan yang relatif baik untuk bermacam-macam masalah optimisasi yang sulit [9]. Algoritma *Elephant Herd Optimization* diperbaiki dengan mengganti nilai acak oleh peta *chaotic* sehingga disebut algoritma *Chaotic Elephant Herding Optimization* (CEHO). Pada faktanya, peta *chaotic* menghasilkan bilangan yang tak berulang dan ergodik, sehingga diharapkan dapat mencari titik solusi yang lebih baik.

Dalam algoritma CEHO dipertimbangkan dua peta *chaotic* dimensi satu yang berbeda yaitu peta lingkaran (*circle*) dan peta sinusoidal. Untuk selanjutnya dalam penelitian ini algoritma CEHO dengan peta lingkaran (*circle*) akan disebut algoritma CEHO-C, sedangkan algoritma CEHO dengan peta sinusoidal akan disebut CEHO-S.

Peta lingkaran (*circle*) didefinisikan oleh Persamaan (6).

(6)

Keterangan:  
a = 0,5

b = 0,2

deret *chaotic* yang dihasilkan ada di dalam (0, 1).

Peta sinusoidal didefinisikan dalam rumus Persamaan (7).

(7)

Keterangan:

a = 2,3

Rumus yang disederhanakan ditulis pada Persamaan (8).

(8)

Peta *chaotic* digunakan untuk menghasilkan deret bilangan *chaos*. Angka acak yang disimbolkan r di dalam Persamaan (1), Persamaan (2) dan Persamaan (5) diganti dengan angka dari deret bilangan *chaos* [9].

1. **Algoritma *Elephant Swarm Water Search* (ESWS)**

Algoritma *Elephant Swarm Water Search* (ESWS) terinspirasi oleh sifat kelakuan gajah dalam mencari air. Ada empat aturan sederhana gajah dalam mencari air yang ditiru dalam algoritma ESWS, yaitu [6]:

1. Gajah berkeliaran untuk mencari air selama kekeringan dalam beberapa kelompok, yang disebut kawanan gajah. Setiap kelompok terdiri dari sejumlah gajah dan semua kelompok (yaitu, kawanan gajah) bekerja sama untuk menemukan air. Pemimpin masing-masing kelompok (gajah tertua) bertanggung jawab untuk mengambil keputusan tentang pergerakan kelompok untuk mencari sumber daya air terbaik. Untuk masalah pengoptimalan (optimisasi), setiap kelompok gajah diidentifikasi oleh kecepatan tertentu dan posisi tertentu sedangkan setiap kelompok gajah dari kawanan mirip dengan solusi dari masalah optimasi.
2. Setiap kali kelompok gajah menemukan beberapa sumber daya air, pemimpin berkomunikasi (melalui komunikasi seismik, akustik, kimia, visual dan taktil) dengan kelompok lain dari kawanan gajah tentang kuantitas dan kualitas air. Untuk masalah maksimisasi, nilai *fitness* dan fungsi objektif berbanding lurus dengan kuantitas dan kualitas sumber daya air. Ketinggian air yang lebih baik menunjukkan solusi yang lebih baik.
3. Gajah memiliki memori yang sangat tajam. Setiap kelompok gajah dapat mengingat lokasi pasokan air terbaik yang ditemukan oleh kelompoknya sendiri (solusi terbaik lokal) sejauh ini, dan lokasi sumber air terbaik sejauh ini (solusi terbaik global), yang ditemukan oleh seluruh kawanan atau semua kelompok. Berdasarkan ingatan (solusi) ini, kelompok gajah dapat berpindah dari satu titik ke titik lain yaitu kecepatan dan posisi setiap kelompok gajah diperbarui secara bertahap selama proses pencarian sesuai dengan beberapa aturan (aturan untuk pencarian global, pencarian lokal dan pembaruan posisi dijelaskan kemudian). Teknik komunikasi jarak jauh dan jarak pendek gajah masing-masing dominan untuk pencarian global dan lokal.
4. Pencarian air di daerah lokal dan global dikendalikan oleh konstanta probabilistik yang disebut *switching probability* *p* ∈ [0, 1]. Pemimpin kelompok mengambil keputusan probabilistik untuk beralih antara pencarian lokal dan pencarian global selama pencarian air. Karena kedekatan fisik dan faktor-faktor lain seperti pelemahan sinyal dari jarak jauh, pencarian air lokal dapat memiliki fraksi *p* yang signifikan dalam kegiatan pencarian secara keseluruhan.

Untuk masalah optimisasi dimensi-d, posisi kelompok gajah ke-i dari segerombolan (terdiri dari partikel N, yaitu, jumlah kelompok gajah) pada iterasi ke-t ditulis dalam Persamaan (9).

(9)

dan kecepatan ditulis dalam Persamaan (10).

(10)

Solusi terbaik secara lokal oleh kelompok gajah ke-i pada iterasi saat ini ditulis dalam Persamaan (11).

(11)

dan solusi terbaik global ditulis dalam Persamaan (12).

(12)

Awalnya, posisi dan kecepatan kelompok-kelompok gajah ditempatkan secara acak di seluruh ruang pencarian. Ketika iterasi berlanjut, kecepatan dan posisi gajah diperbarui sesuai dengan beberapa aturan.

Sebagian besar aktivitas pencarian air dapat terjadi diskala global. Dalam praktiknya, pencarian air yang berdekatan di lingkungan yang tidak terlalu jauh lebih mungkin dipotong oleh kelompok daripada yang jauh. Untuk ini, konstanta yang dikenal sebagai *switching probability p* digunakan untuk beralih antara pencarian air global dan lokal.

Diasumsikan bahwa jika nilai variabel acak lebih besar dari p, pencarian air global umum akan dilakukan, pencarian air lokal yang intens lainnya akan dieksekusi. Kondisi acak ini membantu mengurangi kemungkinan terjebak di optima lokal.

Solusi terbaik global dan lokal diperbarui setelah setiap iterasi. Ketika iterasi berlanjut, kecepatan dari partikel diperbarui dengan cara yang berbeda untuk pencarian global dan lokal sesuai dengan Persamaan (13) dan Persamaan (14) tergantung pada nilai parameter *p*.

(13)

Jika r > p (untuk pencarian titik *global*)

(14)

Jika r ≤ p (untuk pencarian lokal)

Dengan r (1, d) menghasilkan suatu baris dimensi-d yang tersusun dari angka-angka acak dalam rentang [0,1].

⨀ menunjukkan perkalian elemen.

adalah bobot inersia pada iterasi saat ini untuk keseimbangan antara eksplorasi dan eksploitasi yang ditulis dalam Persamaan (15).

(15)

Posisi dari suatu kelompok gajah dimodifikasi sesuai dengan Persamaan (16).

(16)

tmax adalah nilai dari jumlah iterasi maksimum.

Xmax adalah batas atas dari posisi.

Xmin adalah batas bawah dari posisi.

Setelah menyelesaikan semua iterasi, gajah secara bertahap memperbarui posisi mereka dan mencapai posisi sumber daya air terbaik, yaitu, solusi terbaik dari masalah optimisasi [6].

1. **METODOLOGI PENELITIAN**
2. **Skema Alur Penelitian**

Tahapan dari penelitian ini dapat dilihat pada Gambar 2.

Gambar 2 Bagan Alir Penelitian

Algoritma *Elephant Swarm Water Search* (ESWS), *Chaotic Elephant Herding Optimization-Circle* (CEHO-C) dan *Chaotic Elephant Herding Optimization-Sinusoidal* (CEHO-S) merupakan algoritma hasil pengembangan hibrida atau hasil modifikasi dari algoritma *Elephant Herd Optimization* (EHO).

Tahapan dari algoritma *Elephant Herd Optimization* (EHO) terlihat pada Tabel I [8]:

TABEL I. ALGORITMA EHO

|  |  |
| --- | --- |
| Algoritma EHO | |
| 1 | ***Initialization*** |
| 2 | Atur *generation counter* t = 1 |
|  | Atur *maximum generation* MaxGen |
| 3 | Inisialisasi populasi |
| 4 | ***Repeat*** |
| 5 | Sortir semua gajah sesuai dengan kelayakannya |
| 6 | ***for all*** klan ci di dalam populasi ***do*** |
| 7 | ***for all*** gajah j di dalam klan ci ***do*** |
| 8 | Perbaharui dan *generate* dengan persamaan (2) |
| 9 | ***if***  ***then*** |
| 10 | Perbaharui dan *generate* dengan persamaan (3) |
| 11 | ***end if*** |
| 12 | ***end for*** |
| 13 | ***end for*** |
| 14 | ***for all*** klan ci di dalam populasi ***do*** |
| 15 | Ganti gajah terburuk di dalam klan ci oleh persamaan (5) |
| 16 | ***end for*** |
| 17 | Evaluasi populasi dengan posisi terbaharui |
| 18 | ***until*** t < *MaxGen* |
| 19 | ***return*** solusi terbaik ditemukan |

Tahapan dari algoritma ESWS terlihat pada Tabel II [6].

TABEL II. ALGORITMA ESWS

|  |  |
| --- | --- |
| Start ESWS |  |
| Definisikan N, d, tmax, Xmax, Xmin, p dan fungsi objektif f; | input |
| ***for*** i = 1 ***to*** N | inisialisasi |
| *Initialize* dan ; |  |
| ; |  |
| ***end***; |  |
| Evaluasi nilai *fitness* untuk seluruh posisi N (kelompok gajah) | Evaluasi dan cari yang terbaik |
| ; |  |
| Menentukan nilai sesuai dengan aturan pembaruan bobot Persamaan (15) | Penentuan |
| ***for*** t = 1 ***to*** tmax | Mulai iterasi |
| ***for*** i = 1 ***to*** N |  |
| If rand > p |  |
| Cari air global atau perbarui kecepatan gajah memakai Persamaan (13) | Pencarian titik *global* |
| ***else*** |  |
| Cari air lokal atau perbarui kecepatan gajah memakai Persamaan (14) | Pencarian titik lokal |
| ***end if***; |  |
| Perbarui posisi memakai Persamaan (16) | Perbarui posisi |
| Evaluasi nilai fitness for |  |
| ***if*** | Perbarui terbaik saat ini |
|  |  |
| ***end if*** |  |
| ***if*** | Perbarui *global* terbaik |
|  |  |
| ***end if*** |  |
| ***end for*** |  |
|  |  |
| ***end for*** | Akhir iterasi |
| ***return*** and |  |
| ***end*** ESWS | *output* |

1. **Pengumpulan Data**

Persamaan dari rumus fungsi yang dijadikan sebagai alat ukur untuk membandingkan kinerja dari algoritma ESWS, CEHO-C dan CEHO-S tampak dalam Tabel III.

TABEL III. RUMUS FUNGSI PENGUKUR KINERJA

|  |  |
| --- | --- |
| Fungsi | Rumus |
| *Rosenbrock* [10] |  |
| *Rotating hyper-ellipsoid* [11] |  |
| *Rotated high conditioned elliptic* [12] |  |
| *Schwefel* [13] |  |
| *Ackley* [14] |  |
| *Griewank* [15] |  |
| *Rastrigin* [16] |  |

Nilai setiap titik minimum global dari fungsi alat ukur terdapat pada Tabel IV.

TABEL IV. TITIK MINIMUM FUNGSI

|  |  |  |
| --- | --- | --- |
| No | Fungsi | Titik minimum global |
| 1 | *Rosenbrock* | 0 |
| 2 | *Rotating hyper-ellipsoid* | 0 |
| 3 | *Schwefel* | 0 |
| 4 | *Ackley* | 0 |
| 5 | *Griewank* | 0 |
| 6 | *Rastrigin* | 0 |

Ukuran populasi N = 50. Jumlah klan = 5. Parameter α = 0,5. Parameter β = 0,1. Iterasi tmax = 5000.

1. **Analisa Data**

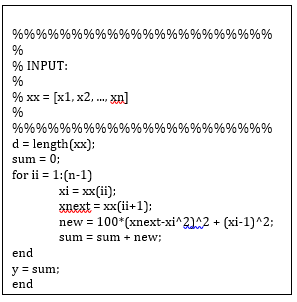
Semua fungsi pengukur ditulis dalam MATLAB versi 2018a. Contoh penulisan fungsi *Rosenbrock* ditulis dalam MATLAB terlihat pada Gambar 3.

Ukuran populasi N = 50. Iterasi maksimum tmax = 5000.

1. **Tahapan penelitian**

Tahapan atau langkah-langkah dari penelitian ini adalah tahapan implementasi dan tahapan pengujian. Implementasi dilaksanakan dengan *setting* ukuran populasi, jumlah iterasi maksimum dan nilai p atau *switching probability*.

Tahap pengujian dilakukan terhadap enam fungsi pengukur yaitu fungsi *Rosenbrock, Ackley, Schwefel, Elliptic, Griewank* dan *Rastrigin* untuk menguji efektivitas dari algoritma ESWS, CEHO-C dan CEHO-S.



Gambar 3 Fungsi Rosenbrock Dalam MATLAB

Pencarian titik minimum global dilakukan dengan alat *software* MATLAB versi 2018a dengan laptop i7.

1. **HASIL DAN PEMBAHASAN**

Dalam penelitian terdahulu perbandingan algoritma dilakukan antara algoritma yang berbeda jenis seperti perbandingan algoritma gajah dengan algoritma ABC, PSO, GA dengan hasil ESA(*Elephant Search Algorithm*) lebih efektif dan efisien daripada algoritma ABC(*Artificial Bee Colony*), GA(*Genetic Algorithm*), PSO(*Particle Swarm optimization*) dan FSA (*Firefly Search Algorithm*) [17].

Ukuran populasi N = 50. Iterasi maksimum tmax = 5000. p = 0,6.

TABEL V. TITIK MINIMUM ESWS

|  |  |  |
| --- | --- | --- |
| Fungsi | Titik minimum global fungsi | Titik minimum terbaik ESWS |
| *Rosenbrock* | 0 | 3,77x10-4 |
| *Rotating hyper-ellipsoid* | 0 | 5,77x10-91 |
| *Schwefel P2.22* | 0 | 2,32x10-39 |
| *Ackley* | 0 | 4,44x10-15 |
| *Griewank* | 0 | 1,48x10-2 |
| *Rastrigin* | 0 | 9,95x10-1 |

Pada Tabel V terlihat pencarian titik minimum global terbaik yang dilakukan oleh algoritma ESWS mendekati nilai titik minimum fungsi yang sebenarnya dengan hasil yang cukup akurat.

TABEL VI. TITIK MINIMUM CEHO-C

|  |  |  |
| --- | --- | --- |
| Fungsi | Titik optimal fungsi | Titik minimum terbaik CEHO-C |
| *Rotated Rosenbrock* | -1400 | -8,974 x 102 |
| *Rotated high conditioned elliptic* | -1300 | 6,063 x 103 |
| *Schwefel* | -100 | 1,677 x 102 |
| *Rotated Ackley* | -700 | -6,8 x 102 |
| *Rotated Griewank* | -500 | -5 x 102 |
| *Rastrigin* | -400 | -3,87 x 102 |

Dari Tabel VI dapat dilihat bahwa hasil pencarian titik minimum terbaik dari algoritma CEHO-C kurang mendekati nilai titik minimum fungsi pengukur yang sebenarnya.

TABEL V. TITIK MINIMUM CEHO-S

|  |  |  |
| --- | --- | --- |
| Fungsi | Titik optimal fungsi | Titik minimum terbaik CEHO-S |
| *Rotated Rosenbrock* | -1400 | -8,996 x 102 |
| *Rotated high conditioned elliptic* | -1300 | 2,963 x 103 |
| *Schwefel* | -100 | 9,702 x 102 |
| *Rotated Ackley* | -700 | -6,799 x 102 |
| *Rotated Griewank* | -500 | -4,999 x 102 |
| *Rastrigin* | -400 | -3,781 x 102 |

Pada Tabel VII terlihat bahwa hasil pencarian titik minimum terbaik dari algoritma CEHO-S kurang mendekati nilai sebenarnya dari titik minimum fungsi pengukur.

TABEL VIII. PERBANDINGAN TITIK MINIMUM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Fungsi | Titik minimum terbaik ESWS | Titik minimum terbaik CEHO-C | Titik minimum terbaik  CEHO-S |
| 1 | *Rosenbrock* | 3 |  | -8 |
| 2 | *Schwefel* | 2 | 1 | 9 |
| 3 | *Ackley* | 4 |  | -6 |
| 4 | *Griewank* | 1 |  |  |
| 5 | *Rastrigin* | 9 | -3 |  |

Dilihat dari Tabel VIII, fungsi *rotated high conditioned elliptic* memiliki nilai titik minimum terbaik diperoleh dengan algoritma ESWS yaitu .

Titik minimum terbaik pada fungsi *Schwefel* diperoleh dengan algoritma ESWS yaitu .

Fungsi *Rastrigin* mencapai titik minimum terbaik dengan algoritma ESWS yaitu 9.

Dari tabel di atas terlihat secara keseluruhan nilai minimum terbaik diperoleh oleh algoritma ESWS. Kelebihan dari algoritma ESWS yaitu dapat mencari titik minimum global terbaik pada fungsi *Schwefel* tanpa terjebak pada titik minimum lokal.

Kelemahan dari penelitian ini belum diujicoba pada beberapa jenis fungsi pengukur lainnya. Dari hasil penelitian terlihat bahwa nilai titik minimum yang diperoleh melalui algoritma ESWS untuk enam fungsi sesuai dengan titik target nilai titik minimum fungsi yaitu nilai nol, sedangkan algoritma CEHO-C dan CEHO-S tidak terlalu dekat dengan titik minimum target dari enam fungsi pengukur.

Berdasarkan hasil pengujian kinerja dari ketiga algoritma ini yang sudah diujicobakan untuk mencari nilai titik minimum global dari enam fungsi maka dapat disimpulkan bahwa algoritma ESWS lebih efektif daripada algoritma CEHO-C dan CEHO-S.

1. **Kesimpulan Dan Saran**

Berdasarkan hasil dari penelitian maka dapat diperoleh kesimpulan bahwa algoritma ESWS telah mencapai titik minimum terbaik jika dibandingkan dengan algoritma CEHO-C dan CEHO-S ketika diujicoba dengan enam fungsi pengukur yaitu *Rosenbrock, Elliptic, Schwefel, Ackley, Griewank* dan *Rastrigin*.

Berdasarkan hasil penelitian ini dapat disimpulkan bahwa algoritma ESWS terbukti cukup efektif dan efisien untuk mencari nilai titik minimum global terbaik dalam persoalan metode optimasi.

Algoritma ESWS tampak sangat unggul dalam mencapai titik minimum untuk fungsi *Elliptic, Schwefel, Ackley, Rosenbrock, Griewank* dan *Rastrigin*.

Kelemahan dari penelitian ini adalah belum uji coba pada fungsi lainnya oleh karena keterbatasan peralatan yang ada.

Penelitian selanjutnya dapat diujicoba beberapa hibrida algoritma gajah pada beberapa fungsi pengukur lainnya seperti fungsi *Fletcher-Powell*, fungsi *Penalty* #1, fungsi *Penalty* #2, fungsi *Quartic with noise*, fungsi *Sphere* dan fungsi *Step*.

1. **UCAPAN TERIMA KASIH**

Penulis mengucapkan terima kasih banyak pada Sekolah Tinggi Teknologi Bandung yang memberikan dukungan penuh atas penelitian ini.

**Daftar Pustaka:**

[1] N. Giarsyani, A. F. Hidayatullah, and R. Rahmadi, “i ISSN . 2620-6900 ( Online ) 2620-6897 ( Cetak ),” *JIRE*, vol. 3, no. 1, pp. 48–57, 2020.

[2] S. Fadli and K. Imtihan, “Penerapan Multi-Objective Optimization on the Basis of Ratio Analysis ( Moora ) Method,” *JIRE*, vol. 2, no. 2, pp. 10–19, 2019.

[3] M. A. Elhosseini, R. A. El Sehiemy, Y. I. Rashwan, and X. Z. Gao, “On the performance improvement of elephant herding optimization algorithm,” *Knowledge-Based Syst.*, vol. 166, pp. 58–70, 2019, doi: 10.1016/j.knosys.2018.12.012.

[4] I. Strumberger, M. Beko, M. Tuba, M. Minovic, and N. Bacanin, “Elephant herding optimization algorithm for wireless sensor network localization problem,” *IFIP Adv. Inf. Commun. Technol.*, vol. 521, no. July, pp. 175–184, 2018, doi: 10.1007/978-3-319-78574-5\_17.

[5] E. Tuba, D. Dolicanin-Djekic, R. Jovanovic, D. Simian, and M. Tuba, “Combined elephant herding optimization algorithm with k-means for data clustering,” *Smart Innov. Syst. Technol.*, vol. 107, no. December 2018, pp. 665–673, 2019, doi: 10.1007/978-981-13-1747-7\_65.

[6] S. Mandal, “Elephant swarm water search algorithm for global optimization,” *Sadhana - Acad. Proc. Eng. Sci.*, vol. 43, no. 1, pp. 1–21, 2018, doi: 10.1007/s12046-017-0780-z.

[7] C. H. Prasad, K. Subbaramaiah, and P. Sujatha, “Cost–benefit analysis for optimal DG placement in distribution systems by using elephant herding optimization algorithm,” *Renewables Wind. Water, Sol.*, vol. 6, no. 1, 2019, doi: 10.1186/s40807-019-0056-9.

[8] E. Tuba, I. Ribic, R. Capor-Hrosik, and M. Tuba, “Support Vector Machine Optimized by Elephant Herding Algorithm for Erythemato-Squamous Diseases Detection,” *Procedia Comput. Sci.*, vol. 122, pp. 916–923, 2017, doi: 10.1016/j.procs.2017.11.455.

[9] E. Tuba, R. Capor-Hrosik, A. Alihodzic, R. Jovanovic, and M. Tuba, “Chaotic elephant herding optimization algorithm,” *SAMI 2018 - IEEE 16th World Symp. Appl. Mach. Intell. Informatics Dedic. to Mem. Pioneer Robot. Antal K. Bejczy, Proc.*, vol. 2018-Febru, no. September, pp. 213–216, 2018, doi: 10.1109/SAMI.2018.8324842.

[10] A. Raß, “High precision particle swarm optimization algorithm (HiPPSO),” *J. Open Res. Softw.*, vol. 8, no. 1, pp. 1–7, 2020, doi: 10.5334/JORS.282.

[11] E. Kaya, “A Novel Neural Network Training Algorithm for the Identification of Nonlinear Static Systems : Artificial Bee Colony Algorithm Based on Effective Scout Bee Stage,” *Symmetry (Basel).*, vol. 13, p. 419, 2021.

[12] I. Damaj, M. Elshafei, M. El-Abd, and M. E. Aydin, “An analytical framework for high-speed hardware particle swarm optimization,” *Microprocess. Microsyst.*, vol. 72, no. 102949, pp. 1–16, 2020, doi: 10.1016/j.micpro.2019.102949.

[13] A. Nurhayati and A. G. Darmoyono, “A Comparison of Hybrid Methods of the Krill Herd Algorithm,” 2018, doi: 10.1109/INCAE.2018.8579364.

[14] S. Agrawal and R. Kumar, “SELF ORGANIZING MIGRATING ALGORITHM BASED ON VARIOUS CHAOTIC MAPS,” *Adv. Appl. Math. Sci.*, vol. 19, no. 12, pp. 1323–1335, 2020.

[15] A. Nurhayati, “Perbandingan algoritma cuckoo search krill herd (cskh) dan levy-flight krill herd (lkh),” 2018, no. Selisik, pp. 448–453.

[16] S. Weiser, H. Wulf, and J. Ihlemann, “Characterization of the objective function landscape using a modified Dijkstra algorithm,” in *Proceedings in Applied Mathematics & Mechanics published*, 2021, vol. 20, no. 1, pp. 1–3, doi: 10.1002/pamm.202000072.

[17] M. Panda, “Elephant search optimization combined with deep neural network for microarray data analysis,” *J. King Saud Univ. - Comput. Inf. Sci.*, pp. 1–9, 2017, doi: 10.1016/j.jksuci.2017.12.002.