

Opposition-Based Dynamic Grey Wolf Optimizer untuk Eksplorasi dan Eksploitasi

I Gede Abhijana Prayata Wistara^{a1}, I Gusti Agung Gede Arya Kadyanan^{a2}

^aProgram Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana

Jalan Raya Kampus Udayana, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia

¹wistara.2308561142@student.unud.ac.id (Corresponding Author)

²gungde@unud.ac.id

Abstract

Grey Wolf Optimizer (GWO) is a prominent swarm intelligence algorithm, but it exhibits significant drawbacks, including a strong search bias towards the origin and premature convergence on complex multimodal landscapes. To address these limitations, this paper proposes a novel hybrid algorithm, the Opposition-Based Dynamic Grey Wolf Optimizer (OB-DGWO). The proposed method integrates a dynamic prey estimation strategy to mitigate search bias with an Opposition-Based Learning (OBL) mechanism to enhance population diversity and global exploration capabilities. The performance of OB-DGWO was rigorously evaluated against the conventional GWO, GWO with OBL, and the dynamic GWO (DGWO) using standard unimodal and multimodal benchmark functions. Experimental results demonstrate that the proposed OB-DGWO exhibits superior robustness. It successfully overcomes the failure of DGWO on problems with optima at the origin, while demonstrating improved accuracy and consistency on complex multimodal functions where the standard GWO fails. The findings indicate that OB-DGWO provides a more balanced and reliable approach for solving diverse optimization problems.

Keywords: Metaheuristic, Swarm Intelligence, Grey Wolf Optimizer, Opposition-Based Learning, Hybrid Algorithm, Global Optimization

1. Pendahuluan

Metaheuristics adalah sekumpulan teknik berbasis strategi yang mampu menjelajahi *search space* pada suatu masalah optimisasi secara cerdas untuk menemukan solusi yang mendekati optimal. Algoritma berbasis metaheuristik ini dapat dikategorikan menjadi dua jenis utama: algoritma berbasis *local search* dan algoritma berbasis populasi, di mana *Swarm Intelligence* adalah salah satunya [1].

Swarm Intelligence merupakan teknik yang sangat efisien untuk menyelesaikan masalah optimisasi numerik, terutama dalam menangani masalah yang bersifat *non-linear*, *non-convex*, atau diskontinu. Banyak algoritma metaheuristik modern dikembangkan dengan meniru perilaku sosial berbagai makhluk hidup, seperti kawanan burung, koloni semut, lebah, laba-laba, dan sebagainya [2].

Dalam domain ini, sebuah algoritma baru bernama *Grey Wolf Optimizer (GWO)* dikembangkan dengan meniru hierarki kepemimpinan dan mekanisme berburu serigala abu-abu di alam liar. Mekanisme ini meliputi proses pencarian, pelacakan, pengepungan, hingga penyerangan mangsa. Berbeda dari algoritma metaheuristik cerdas lainnya, GWO secara unik memasukkan aspek hierarki kepemimpinan dalam modelnya [3].

GWO adalah algoritma *Swarm Intelligence (SI)* yang perkembangannya paling pesat dan telah terbukti menjadi *optimizer* yang andal. Keunggulannya terletak pada sifatnya yang adaptif, kebutuhan komputasi yang ringan, dan memiliki sedikit parameter untuk diatur. Pada tahap awal pencarian, GWO berfokus secara intensif pada fase *exploration*. Namun, seiring berjalannya iterasi,

fokusnya bergeser ke fase *exploitation* melalui perubahan posisi bertahap dari tiga serigala pemimpin terbaiknya [4].

Algoritma GWO telah berhasil diterapkan untuk berbagai masalah di dunia nyata. Pada [5], Zhang et al. menggunakan GWO konvensional untuk menyelesaikan masalah perencanaan jalur pada *unmanned combat aerial vehicle* (UCAV). Dalam penelitian lain [6], algoritma GWO konvensional dimanfaatkan untuk mengoptimalkan parameter *kernel extreme learning machine* guna memprediksi kebangkrutan perusahaan. GWO mampu mengatasi beragam masalah optimisasi yang mencakup berbagai bidang penelitian, mulai dari rekayasa, jaringan dan komunikasi, pemrosesan citra, robotika, matematika, bioinformatika, hingga biomedis.

Meskipun GWO menunjukkan performa yang unggul pada berbagai aplikasi nyata dibandingkan algoritma berbasis populasi lainnya, ia tidak luput dari tantangan. Sebagai contoh, saat digunakan untuk menyelesaikan fungsi *multi-modal*, algoritma GWO konvensional rentan terjebak dalam *local optima*. Akibatnya, *laju konvergensi* (*convergence rate*) dapat menurun drastis pada tahap akhir proses evolusi [7].

Karena itu, sejumlah varian GWO telah dikembangkan untuk mengatasi kedua kelemahan tersebut. Luo [8] mengusulkan GWO dinamis yang memungkinkan posisi serigala berubah secara lebih fleksibel. Dalam versi ini, serigala pemimpin terus-menerus memperkirakan lokasi mangsa, sehingga memberikan kemampuan pelacakan yang lebih baik. Ada pula *random opposition-based GWO* yang diusulkan oleh Long et al. [7] untuk mencegah GWO terjebak di *local minima*, sekaligus mencapai keseimbangan yang lebih baik antara fase *exploration* dan *exploitation*.

Teorema *No-free-Lunch* (NFL) [9] menyatakan secara logis bahwa tidak ada satu pun teknik optimisasi berbasis populasi yang ideal untuk menyelesaikan semua jenis masalah. Artinya, sebuah algoritma mungkin memberikan hasil yang sangat memuaskan untuk sekelompok masalah, tetapi bisa jadi menunjukkan performa yang buruk pada kelompok masalah yang berbeda. Atas dasar itulah, penulis mengusulkan *enhanced Opposition-Based Dynamic GWO* (OB-DGWO), sebuah algoritma hasil penyempurnaan yang bertujuan untuk meningkatkan performa pencarian dari algoritma GWO konvensional.

2. Metode Penelitian

Pada penelitian ini, diusulkan sebuah algoritma hibrida baru bernama *enhanced Opposition-Based Dynamic Grey Wolf Optimizer* (OB-DGWO) untuk meningkatkan performa optimisasi. Algoritma ini dirancang untuk mengatasi kelemahan yang melekat pada GWO konvensional, yakni kecenderungan untuk stagnan di *local optima* dan adanya bias pencarian (*search bias*).

Untuk mencapai tujuan tersebut, metodologi penelitian ini disusun melalui beberapa tahapan berikut:

- a. Implementasi algoritma GWO konvensional sebagai acuan dasar (*baseline*).
- b. Implementasi *Enhanced / Dynamic Grey Wolf Optimizer* (EGWO / DGWO) berdasarkan kerangka kerja Luo (2019) untuk mengatasi *search bias*.
- c. Implementasi *Opposition-Based Learning* (OBL) sebagai mekanisme untuk meningkatkan kualitas populasi awal dan kemampuan eksplorasi, yang terinspirasi dari penelitian Long et al.
- d. Pengembangan algoritma hibrida OB-DGWO dengan mengintegrasikan DGWO dan OBL.
- e. Perancangan skenario pengujian komprehensif menggunakan fungsi benchmark standar untuk mengevaluasi dan membandingkan performa setiap algoritma.
- f. Analisis statistik terhadap hasil pengujian untuk memvalidasi signifikansi peningkatan performa yang dicapai oleh algoritma yang diusulkan.

2.1. GWO

GWO adalah algoritma metaheuristik yang terinspirasi dari hierarki sosial dan perilaku berburu kolektif serigala abu-abu (*Canis lupus*). Dalam GWO, solusi terbaik dalam populasi dianggap sebagai serigala *alpha* (α), diikuti oleh solusi terbaik kedua dan ketiga, yaitu *beta* (β) dan *delta* (δ).

Sementara itu, sisa populasi lainnya, yang disebut serigala *omega* (ω), akan mengikuti panduan dari ketiga pemimpin tersebut. Proses perburuan dimodelkan melalui tiga tahapan utama: mencari, mengepung, dan menyerang mangsa. Posisi setiap serigala omega diperbarui dengan meniru proses pengepungan mangsa yang dipandu oleh ketiga pemimpin. Jarak terhadap masing-masing pemimpin dihitung, dan posisi baru ditentukan sebagai rata-rata dari tiga vektor gerakan potensial. Vektor gerakan (X_1, X_2, X_3) menuju alpha, beta, dan delta dihitung sebagai berikut:

$$D_\alpha = |C_1 \cdot X_\alpha - X(t)| \quad (1)$$

$$X_1 = X_\alpha - A_1 \cdot D_\alpha \quad (2)$$

$$D_\beta = |C_2 \cdot X_\beta - X(t)| \quad (3)$$

$$X_2 = X_\beta - A_2 \cdot D_\beta \quad (4)$$

$$D_\delta = |C_3 \cdot X_\delta - X(t)| \quad (5)$$

$$X_3 = X_\delta - A_3 \cdot D_\delta \quad (6)$$

Dalam GWO, solusi terbaik dalam populasi dianggap sebagai serigala *alpha* (α), di mana $X(t)$ adalah posisi serigala saat ini, sedangkan $X_\alpha, X_\beta, X_\delta$ adalah posisi pemimpin. Vektor A dan C adalah koefisien acak yang mengontrol keseimbangan antara eksplorasi dan eksploitasi. Posisi baru serigala kemudian diperbarui menggunakan persamaan:

$$X(t + 1) = (X_1 + X_2 + X_3)/3 \quad (7)$$

2.2. DGWO / EGWO

Untuk mengatasi kelemahan GWO konvensional, penelitian ini mengadopsi kerangka kerja DGWO yang diusulkan oleh Luo. DGWO terlebih dahulu memperkirakan lokasi mangsa secara dinamis, kemudian mengarahkan seluruh kawanan serigala menuju satu titik estimasi tersebut.

- a. Posisi mangsa (X'_p) diestimasi pada setiap iterasi menggunakan pembobotan dari posisi pemimpin, yang dihitung sebagai berikut:

$$X'_p(t) = w_\alpha X_\alpha(t) + w_\beta X_\beta(t) + w_\delta X_\delta(t) + \epsilon(t) \quad (8)$$

di mana $w_\alpha, w_\beta, w_\delta$ adalah bobot yang ditentukan secara acak untuk mencerminkan hierarki sosial, dan $\epsilon(t)$ adalah vektor galat acak dari distribusi Normal $N(0, \sigma(t))$ yang mensimulasikan ketidakpastian dalam estimasi. Standar deviasi $\sigma(t)$ menurun secara eksponensial seiring berjalannya iterasi untuk memastikan pergeseran dari eksplorasi ke eksploitasi.

- b. Setiap serigala kemudian bergerak langsung menuju posisi mangsa yang telah diestimasi menggunakan persamaan:

$$X_i(t + 1) = X'_p(t) - r \cdot |X'_p(t) - X_i(t)| \quad (9)$$

di mana r adalah vektor acak dalam rentang $[-2, 2]$ yang mengontrol langkah serangan atau pencarian, yang terbukti lebih efektif dalam menghindari bias pencarian dan meningkatkan laju konvergensi.

2.3. OB-GWO / GWO-OBL

OBL adalah teknik yang digunakan untuk meningkatkan kemampuan eksplorasi dan kualitas solusi awal dari sebuah algoritma optimisasi. Prinsip dasarnya adalah bahwa solusi yang berada di

arah berlawanan dari solusi saat ini memiliki kemungkinan yang cukup besar untuk menjadi kandidat yang lebih baik. Untuk sebuah solusi X dalam ruang pencarian terbatas $[lb, ub]$, solusi oposisinya (X_{opp}) didefinisikan sebagai:

$$X_{opp} = lb + ub - X \tag{10}$$

Pada awal proses, sebuah populasi awal sejumlah N serigala dibuat secara acak. Kemudian, populasi oposisinya juga dibuat. Dari gabungan $2N$ serigala tersebut, N serigala dengan nilai *fitness* terbaik dipilih sebagai populasi awal.

Selama iterasi, mekanisme lompatan (*jumping*) berbasis OBL diterapkan dengan probabilitas rendah (Jr). Jika terpicu, seluruh populasi saat ini akan digabungkan dengan populasi oposisinya, dan N serigala terbaik akan dipilih untuk melanjutkan ke generasi berikutnya. Ini berfungsi sebagai mekanisme yang kuat untuk keluar dari optima lokal.

2.4. OB-DGWO / EGWO-OBL

OB-DGWO adalah algoritma hibrida yang diusulkan dalam penelitian ini. Algoritma ini mengintegrasikan DGWO untuk pembaruan posisi dengan mekanisme OBL untuk peningkatan eksplorasi. Alur kerja OB-DGWO adalah sebagai berikut.

- a. Populasi awal dibuat menggunakan mekanisme inisialisasi berbasis OBL untuk memastikan kualitas dan keragaman yang tinggi sejak awal.
- b. Nilai *fitness* dari setiap serigala dihitung, dan tiga serigala terbaik ditetapkan sebagai pemimpin α, β, δ .
- c. Posisi mangsa diestimasi secara dinamis menggunakan persamaan DGWO.
- d. Setiap serigala omega memperbaiki posisinya dengan bergerak langsung menuju estimasi posisi mangsa.
- e. Operator lompatan berbasis OBL diterapkan secara probabilistik untuk membantu kawanan keluar dari optima lokal.
- f. Posisi dan nilai *fitness* dari setiap serigala dievaluasi ulang, dan posisi pemimpin α, β, δ diperbarui jika ditemukan solusi yang lebih baik.
- g. Implementasi OBL sebagai mekanisme untuk meningkatkan kualitas populasi awal dan kemampuan eksplorasi.
- h. Proses diulang hingga kriteria berhenti (jumlah iterasi maksimum) terpenuhi. Solusi terbaik yang ditemukan, yaitu posisi serigala alpha, dikembalikan sebagai hasil akhir.

Untuk memvalidasi performa algoritma OB-DGWO, sebuah skenario pengujian dirancang. Pengujian dilakukan menggunakan dua fungsi *benchmark* yang umum digunakan dalam literatur optimisasi. Satu unimodal dan satu multimodal

Tabel 1. Fungsi Benchmark yang digunakan.

Nama Fungsi	Dimensi (n)	Batas (<i>bounds</i>)	Tipe
Sphere	50	[-10, 10]	Unimodal
Schwefel	50	[-500, 500]	Multimodal

Tabel 2. Konfigurasi Parameter Eksperimen.

Parameter	Nilai
Serigala	60
Iterasi	500
Percobaan	50

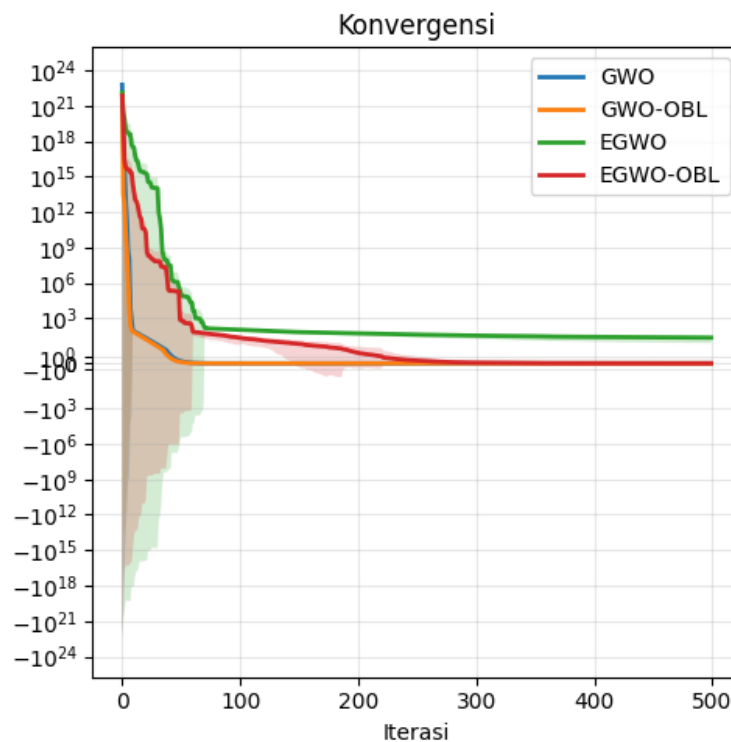
3. Hasil dan Diskusi

Bagian ini menyajikan evaluasi kuantitatif terhadap performa dari empat varian algoritma yang diuji: GWO, GWO-OBL, DGWO / EGWO, dan algoritma hibrida yang kami usulkan, OB-DGWO (direpresentasikan sebagai EGWO-OBL agar konsisten dengan penggunaan terminologi EGWO pada penelitian-penelitian GWO lainnya. Namun terminologi OB-DGWO lebih akurat dan representatif). Analisis difokuskan pada dua fungsi *benchmark*, yaitu Sphere dan Schwefel, yang dipilih untuk mewakili dua kelas masalah optimisasi yang berbeda, yakni unimodal sederhana dan multimodal yang lebih kompleks.

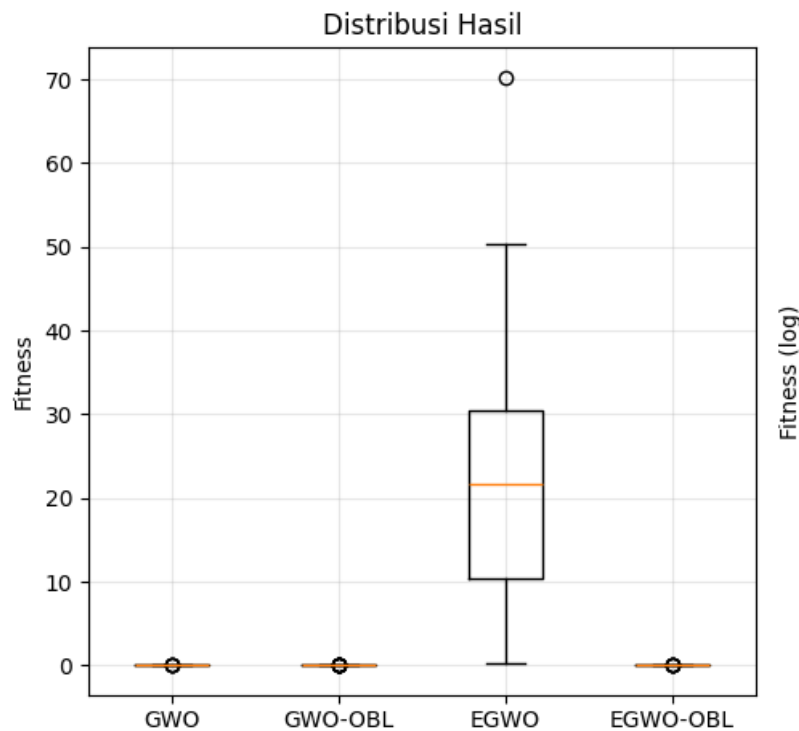
3.1. Analisis Performa pada Fungsi Unimodal (*Sphere*)

Tabel 3. Hasil Statistik Konvergensi Algoritma pada Fungsi *Sphere*.

Algoritma	Mean	Std Dev	Min	Max
GWO	0.000000	0.000000	0.000000	0.000000
GWO-OBL	0.000000	0.000000	0.000000	0.000000
EGWO	22.766707	14.974542	0.307568	70.273696
EGWO-OBL	0.000678	0.001181	0.000004	0.006289



Gambar 1. Konvergensi pada Fungsi *Sphere*



Gambar 2. Distribusi Hasil pada Fungsi *Sphere*

Fungsi *Sphere* digunakan sebagai kasus uji untuk mengevaluasi performa algoritma pada lanskap pencarian unimodal dengan titik optima global yang berlokasi di origin ($f_{min} = 0$). Skenario ini secara efektif mengukur dampak dari bias pencarian (*search bias*) yang mungkin dimiliki oleh sebuah algoritma.

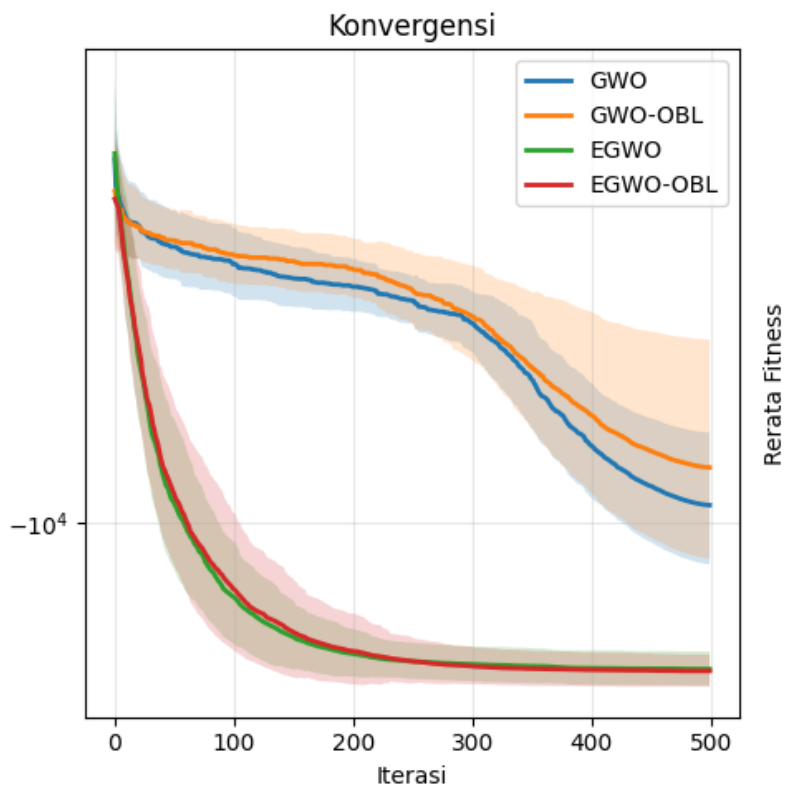
Hasil statistik menunjukkan adanya perbedaan performa yang signifikan antar algoritma. Sebagaimana dapat diamati pada **Gambar 1**, algoritma GWO konvensional dan GWO-OBL menunjukkan efektivitas tertinggi, keduanya mampu mencapai nilai *fitness* rata-rata mendekati nol secara konsisten. Fenomena ini mengindikasikan bahwa bias pencarian inheren pada GWO, yang cenderung mengarahkan agen pencarian menuju titik origin, justru berfungsi sebagai mekanisme konvergensi yang sangat efisien ketika solusi optimal memang berada di lokasi tersebut.

Sebaliknya, EGWO, yang secara spesifik didesain untuk menghilangkan bias ini, menunjukkan performa paling tidak kompetitif dengan rata-rata *fitness* 22.766707. Mekanisme estimasi mangsa dinamis pada EGWO, dalam skenario ini, secara kontra-produktif menjauhkan proses pencarian dari solusi optimal. Algoritma usulan, OB-DGWO, menunjukkan kemampuannya dalam memitigasi kelemahan ini. Dengan rata-rata *fitness* 0.000678, OB-DGWO berhasil menunjukkan perbaikan signifikan dibandingkan EGWO. Integrasi mekanisme *Opposition-Based Learning* (OBL) memberikan tekanan eksplorasi yang efektif untuk mengimbangi bias negatif dari DGWO dalam konteks ini, sehingga mampu mengarahkan kembali pencarian ke area di sekitar titik optimal. Hasil ini menunjukkan bahwa OB-DGWO mempertahankan kapabilitas DGWO sambil menambahkan mekanisme korektif yang membuatnya lebih *robust*.

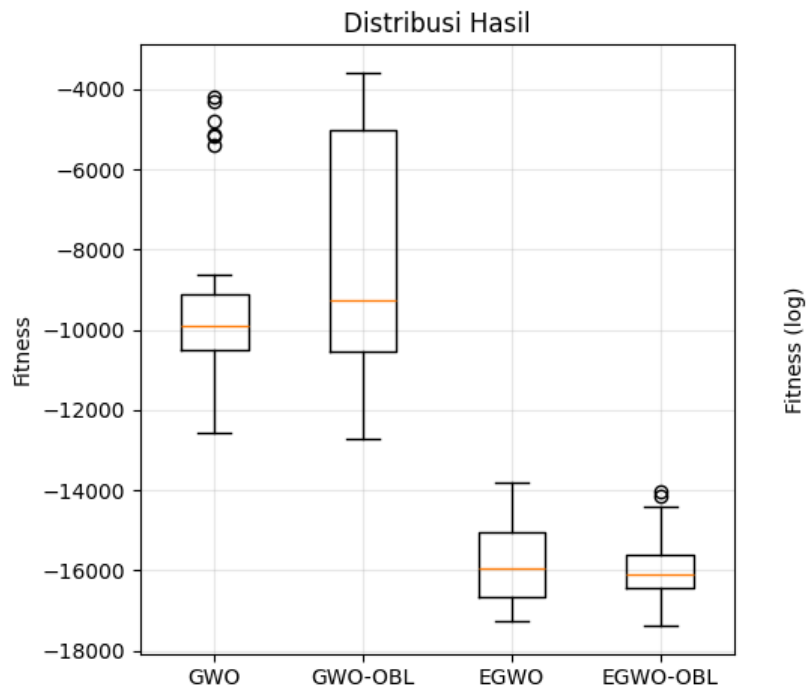
3.2. Analisis Performa pada Fungsi Multimodal (*Schwefel*)

Tabel 4. Hasil Statistik Konvergensi Algoritma pada Fungsi *Schwefel*.

Algoritma	Mean	Std Dev	Min	Max
GWO	-9476.949201	1935.629532	-12559.758292	-4191.724839
GWO-OBL	-8417.213457	2789.127365	-12712.757804	-3579.420752
EGWO	-15879.861795	851.666328	-17255.695513	-13804.917606
EGWO-OBL	-15978.309013	805.743611	-17383.715816	-14038.957651



Gambar 3. Konvergensi pada Fungsi *Schwefel*



Gambar 4. Distribusi Hasil pada Fungsi *Schwefel*

Fungsi *Schwefel* mewakili tantangan optimisasi yang lebih kompleks, dengan karakteristik multi-modal, banyak *local optima*, dan titik minimum global bernilai negatif yang lokasinya jauh dari titik pusat ($f_{min} = -16759.316$). Skenario pengujian ini dirancang untuk mengevaluasi kemampuan eksplorasi global dan kemampuan algoritma dalam menghindari stagnasi.

Hasil pengujian (**Tabel 4** dan **Gambar 3**) menunjukkan tren performa yang berkebalikan dibandingkan dengan fungsi *Sphere*. GWO konvensional dan GWO-OBL menunjukkan performa yang sangat terbatas. *Search bias* yang sebelumnya menjadi keuntungan, kini berubah menjadi penghambat utama yang membatasi kemampuan algoritma untuk menjelajahi area solusi di luar sekitar titik pusat.

Di sisi lain, EGWO menunjukkan peningkatan performa yang substansial, dengan rata-rata *fitness* -15879.86. Kemampuannya untuk melepaskan diri dari bias titik pusat terbukti krusial dalam menavigasi lanskap pencarian yang rumit. Algoritma usulan, OB-DGWO, berhasil mencapai hasil terbaik dengan rata-rata *fitness* -15978.31. Peningkatan ini, meskipun terlihat marginal dibandingkan EGWO, disertai dengan standar deviasi yang lebih rendah (805.74 vs 851.67). Hal ini mengindikasikan bahwa penambahan mekanisme OBL tidak hanya berkontribusi pada penemuan solusi yang lebih baik, tetapi juga meningkatkan konsistensi dan keandalan (reliabilitas) dari proses optimisasinya.

4. Kesimpulan

Penelitian ini telah mengusulkan dan mengevaluasi sebuah algoritma hibrida, OB-DGWO, yang mengintegrasikan mekanisme estimasi mangsa dinamis dari DGWO dengan teknik *Opposition-Based Learning*. Hasil evaluasi pada serangkaian fungsi *benchmark* menunjukkan bahwa OB-DGWO berhasil meningkatkan performa optimisasi, terutama dalam hal ketangguhan (*robustness*) dan konsistensi. Algoritma ini terbukti mampu mengatasi keterbatasan yang dimiliki oleh GWO, OB-GWO, dan DGWO ketika dihadapkan pada kelas masalah yang berbeda, baik unimodal maupun multimodal.

Daftar Pustaka

- [1] S. N. Makhadmeh *et al.*, “Recent Advances in Grey Wolf Optimizer, its Versions and Applications: Review,” *IEEE Access*, vol. 12, pp. 22991–23028, 2024, doi: 10.1109/ACCESS.2023.3304889.
- [2] A. Slowik and H. Kwasnicka, “Nature Inspired Methods and Their Industry Applications—Swarm Intelligence Algorithms,” *IEEE Trans. Ind. Inform.*, vol. 14, no. 3, pp. 1004–1015, Mar. 2018, doi: 10.1109/TII.2017.2786782.
- [3] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.
- [4] S. Gupta and K. Deep, “A memory-based Grey Wolf Optimizer for global optimization tasks,” *Appl. Soft Comput.*, vol. 93, p. 106367, Aug. 2020, doi: 10.1016/j.asoc.2020.106367.
- [5] S. Zhang, Y. Zhou, Z. Li, and W. Pan, “Grey wolf optimizer for unmanned combat aerial vehicle path planning,” *Adv. Eng. Softw.*, vol. 99, pp. 121–136, Sep. 2016, doi: 10.1016/j.advengsoft.2016.05.015.
- [6] M. Wang *et al.*, “Grey wolf optimization evolving kernel extreme learning machine: Application to bankruptcy prediction,” *Eng. Appl. Artif. Intell.*, vol. 63, pp. 54–68, Aug. 2017, doi: 10.1016/j.engappai.2017.05.003.
- [7] W. Long, J. Jiao, X. Liang, and M. Tang, “Inspired grey wolf optimizer for solving large-scale function optimization problems,” *Appl. Math. Model.*, vol. 60, pp. 112–126, Aug. 2018, doi: 10.1016/j.apm.2018.03.005.
- [8] K. Luo, “Enhanced grey wolf optimizer with a model for dynamically estimating the location of the prey,” *Appl. Soft Comput.*, vol. 77, pp. 225–235, Apr. 2019, doi: 10.1016/j.asoc.2019.01.025.
- [9] S. P. Adam, S.-A. N. Alexandropoulos, P. M. Pardalos, and M. N. Vrahatis, “No Free Lunch Theorem: A Review,” in *Approximation and Optimization*, vol. 145, I. C. Demetriou and P. M. Pardalos, Eds., in Springer Optimization and Its Applications, vol. 145. , Cham: Springer International Publishing, 2019, pp. 57–82. doi: 10.1007/978-3-030-12767-1_5.

Halaman ini sengaja dibiarkan kosong