



Daftar isi tersedia di ScienceDirect

## Komputasi Lunak Terapan

jurnal: [www.elsevier.com/locate/asoc](http://www.elsevier.com/locate/asoc)Algoritma Komodo Mlipir<sup>☆</sup>Suyanto Suyanto<sup>\*</sup>, Alifya Aisyah Ariyanto, Alifya Fatimah Ariyanto

Fakultas Ilmu Komputer, Universitas Telkom, Bandung, Indonesia

## artikel info

## Riwayat artikel:

Diterima 15 Maret 2021

Diterima dalam bentuk revisi 18 Oktober 2021

Diterima 4 November 2021

Tersedia secara online pada 17 November 2021

## Kata kunci:

Algoritma Komodo *mlipir*

Optimasi metaheuristik Optimasi

adaptasi mandiri populasi

Keseimbangan eksploitasi-eksplorasi

Dapat diskalakan hingga ribuan dimensi

## abstract

Makalah ini mengusulkan Algoritma Komodo Mlipir (KMA) sebagai pengoptimal metaheuristik baru. Algoritma ini terinspirasi dari dua fenomena: perilaku komodo yang hidup di Nusa Tenggara Timur, Indonesia, dan gaya berjalan orang Jawa yang bernama *mlipir*. Mengadopsi perilaku mencari makan dan reproduksi komodo, populasi beberapa individu komodo (kandidat solusi) di KMA dibagi menjadi tiga kelompok berdasarkan kualitasnya: jantan besar, betina, dan jantan kecil. Pertama, jantan besar berkualitas tinggi melakukan gerakan baru yang disebut eksploitasi tinggi eksplorasi rendah untuk menghasilkan solusi yang lebih baik. Selanjutnya, betina berkualitas menengah menghasilkan solusi yang lebih baik dengan mengawini jantan besar berkualitas tinggi (eksploitasi) atau melakukan partenogenesis (eksplorasi). Terakhir, jantan kecil berkualitas rendah melakukan diversifikasi kandidat solusi dengan menggunakan gerakan baru yang disebut *mlipir* (istilah Jawa yang berarti berjalan di pinggir jalan untuk mencapai tujuan tertentu dengan aman), yang diimplementasikan dengan mengikuti jantan besar di sebagian dimensinya. Adaptasi diri dari populasi juga diusulkan untuk mengendalikan keseimbangan eksploitasi-eksplorasi. Pemeriksaan menggunakan dua puluh tiga fungsi benchmark yang terdokumentasi dengan baik menunjukkan bahwa KMA mengungguli algoritma metaheuristik yang ada saat ini. Selain itu, KMA menyediakan skalabilitas yang tinggi untuk mengoptimalkan fungsi-fungsi ribuan dimensi. Kode sumber KMA tersedia untuk umum di: <https://suyanto.staff.telkomuniversity.ac.id/komodo-mlipir-algorithm> dan <https://www.mathworks.com/matlabcentral/fileexchange/102514-komodo-mlipir-algorithm>.

© 2021 Para Penulis. Diterbitkan oleh Elsevier B.V. Ini adalah artikel akses terbuka di bawah lisensi CC BY-NC-ND lisensi (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

## 1. Pendahuluan

Teknik optimasi matematika dapat dikategorikan menjadi deterministik dan stokastik. Yang pertama bergantung pada informasi gradien dalam masalah yang diberikan untuk mendapatkan solusi; oleh karena itu, teknik ini biasanya bekerja dengan baik untuk masalah unimodal tetapi tidak untuk masalah multi-modal. Sebaliknya, yang terakhir dapat menangani kedua jenis masalah. Selain itu, metode ini sederhana, fleksibel, bebas gradien, dan juga tidak bergantung pada masalah. Oleh karena itu, metode ini lebih populer untuk menangani banyak masalah dunia nyata dengan ruang pencarian non-linear dan non-konveks.

Metode stokastik, yang juga dikenal sebagai algoritma optimasi metaheuristik berbasis populasi, umumnya terinspirasi dari alam, seperti konsep evolusi, genetika, perilaku hewan, fenomena di planet, budaya, masyarakat, fenomenafenomena fisik, atau alat musik. Banyak peneliti telah mengusulkan ratusan algoritma dan variannya, yang dapat dikelompokkan ke dalam tiga kategori: (1) algoritma yang paling banyak dipelajari, seperti Algoritma Genetika (GA) [1,2], Evolusi Diferensial (DE) [3],

Pemrograman Genetika (Genetic Programming/GP) [4,5], Particle Swarm Optimization (PSO) [6], dan Sistem Semut (Ant System/Ant System) [7]; (2) algoritma yang baru saja dikembangkan, seperti Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [8], Success-History Based Parameter Adaptation Differential Evolution (SHADE) [9], SHADE dengan linear population size reduction (LSHADE) [10], LSHADE dengan Ensemble Sinusoidal Differential Covariance Matrix Adaptation with Euclidean Neighborhood (LSHADE-cnEpSin) [11], Firefly Algorithm (FA) [12, 13], Cuckoo Search (CS) [14,15], Bat Algorithm (BA) [16,17], Bacterial Foraging Optimization (BFO) [18,19], Gravitational Search Algorithm (GSA) [20], Salp Swarm Algorithm (SSA) [21], Teaching-Based Optimization (TLBO) [22], Gray Wolf Optimizer (GWO) [23,24], Dragonfly Algorithm (DA) [25,26], Ant Lion Optimizer (ALO) [27,28], Moth-Flame Optimization (MFO) [29,30], dan Artificial Algae Algorithm (AAA) [31,32]; dan (3) algoritma yang baru-baru ini dibuat, seperti Atomic Search Optimization (ASO) [33], Rao Algorithms (RA) [34], Evolutionary Rao Algorithm (ERA) [35], Equilibrium Optimizer (EO) [36], Marine Predators Algorithm (MPA) [37,38], dan Slime Mold Algorithm (SMA) [39,40].

Semua algoritma metaheuristik terlihat sama. Mereka bergantung pada dua konsep yang sama: pergerakan sebuah individu (solusi kandidat) dan interaksi antar individu dalam populasi. Pada umumnya, pergerakan ini menggunakan operator evolusi (crossover dan mutasi) atau teknik random walk dengan distribusi tertentu, seperti uniform, normal (Gaussian), Lévy,

<sup>☆</sup> Penelitian ini didanai oleh Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi, Indonesia, berdasarkan Hibah Matching Fund Keadireka dengan nomor kontrak: 3541/E3/SK.09/KL/2021.

<sup>\*</sup> Penulis yang berkorespondensi.

Alamat e-mail: [suyanto@telkomuniversity.ac.id](mailto:suyanto@telkomuniversity.ac.id) (S. Suyanto), [alifyaaisyah@student.telkomuniversity.ac.id](mailto:alifyaaisyah@student.telkomuniversity.ac.id) (A.A. Ariyanto), [alifyafatimah@student.telkomuniversity.ac.id](mailto:alifyafatimah@student.telkomuniversity.ac.id) (A.F. Ariyanto).

atau Brownian. Sementara itu, interaksi biasanya mengadopsi teori evolusi, fenomena alam, atau perilaku hewan untuk menciptakan strategi pencarian yang menghasilkan keseimbangan eksploitasi-eksplorasi. Eksplorasi secara luas mencari ruang untuk menghindari optima lokal. Sebaliknya, eksploitasi secara lokal mencari solusi yang menjanjikan.

Banyak algoritme terbaru yang dikembangkan berdasarkan ide serupa yang digunakan pada algoritme sebelumnya dengan memperkenalkan beberapa perbaikan. Dengan kata lain, algoritma ini adalah versi perbaikan dari metode sebelumnya. Sebagai contoh, DE adalah versi perbaikan dari GA. DE menggunakan individu-individu dengan bilangan real, bukan bilangan biner, dan memperkenalkan dua operator evolusi: crossover binomial dan mutasi diferensial dengan faktor penskalaan untuk mengontrol besarnya mutasi. Dibandingkan dengan GA, DE secara umum memberikan kinerja yang lebih tinggi dalam menyelesaikan masalah optimasi kontinu. Namun, seperti algoritma metaheuristik lainnya, kinerjanya bergantung pada pengaturan parameter kontrol, yang bergantung pada masalah. Oleh karena itu, banyak varian diusulkan dengan menyediakan adaptasi mandiri untuk menyeting parameter kontrol secara adaptif selama evolusi. SHADE, LSHADE, LSHADE-cnEpSin, jDE, dan IMODE adalah beberapa varian DE mutakhir yang memberikan performa tinggi untuk fungsi benchmark klasik [37] dan rangkaian tes CEC [11, 41,42]. Sayangnya, semua algoritme tersebut tidak dapat diskalakan dan tidak stabil. Mereka membutuhkan populasi yang lebih tinggi dan lebih banyak evaluasi fungsi (jumlah individu harus dibangkitkan dan dievaluasi selama evolusi) untuk memecahkan masalah dimensi yang lebih tinggi. Sebagai contoh, ukuran populasi LSHADE-cnEpSin harus diatur menjadi 18 kali dimensi dari probabilitas yang diberikan [11]. Jika tidak, kinerjanya akan menurun seiring dengan bertambahnya dimensi. Masalah lainnya adalah, untuk banyak kasus, mereka tidak stabil untuk beberapa percobaan independen dengan benih acak yang berbeda dari populasi awal. Dengan kata lain, mereka tidak menjamin untuk mendapatkan solusi optimum global.

Oleh karena itu, beberapa algoritme terbaru dirancang untuk dapat diskalakan dan stabil, seperti EO, MPA, dan SMA. EO terinspirasi dari model keseimbangan massa untuk memperkirakan keadaan dinamis dan ekuilibrium [36]. Sebuah partikel dalam EO merepresentasikan sebuah kandidat solusi (agen pencari). Berdasarkan kualitas atau nilai fitness, populasi beberapa partikel dalam EO dibagi menjadi dua kelompok: (1) lima kandidat keseimbangan yang terdiri dari empat partikel yang paling cocok dan rata-ratanya, yang mengeksplorasi area yang menjanjikan di sekitar global optimum; dan (2) partikel lainnya yang melakukan eksplorasi. Sebuah evaluasi dengan menggunakan 58 fungsi benchmark dan tiga masalah dunia nyata menunjukkan bahwa EO secara signifikan mengungguli enam teori: GA, GSA, PSO, SSA, GWO, dan CMA-ES, namun sedikit dikalahkan oleh SHADE dan LSHADE-SPACMA. Namun, yang mengesankan, EO memberikan skalabilitas dari 10 hingga 200 dimensi, meskipun untuk beberapa fungsi tolak ukur.

KKL dimotivasi oleh taktik mencari makan yang luas pada predator lautan bersama dengan kebijakan tingkat kontak yang optimal dalam interaksi predator-mangsa [37]. Strategi ini dikembangkan dengan menggunakan gerakan Lévy dan Brown. Evaluasi menggunakan 59 fungsi tolak ukur menunjukkan bahwa MPA kompetitif dengan LSHADE-cnEpSin. Strategi ini secara signifikan mengungguli GA, GSA, PSO, SSA, CS, dan CMA-ES. Secara statistik, MPA dapat dikategorikan sebagai algoritma pengoptimalan berkinerja tinggi yang mencapai kinerja yang sama dengan SHADE dan LSHADE-cnEpSin. Selain itu, MPA juga menyediakan skalabilitas dari 100 hingga 500 dimensi untuk beberapa fungsi benchmark.

Sementara itu, SMA meniru mode osilasi jamur lendir di alam [39]. Metode ini menggunakan bobot adaptif untuk meniru umpan balik biososial (negatif dan positif) selama mencari makan untuk mendapatkan keseimbangan eksploitasi-eksplorasi. Metode ini memiliki tiga langkah evolusi: distribusi acak (menemukan makanan), eksplorasi dan transformasi adaptif (mendekati makanan), dan eksploitasi (membungkus makanan). Sebuah evaluasi dengan menggunakan 33 fungsi tolak ukur menunjukkan bahwa, untuk beberapa fungsi tolak ukur tersebut,

SMA stabil dan menjamin optimalisasi global. Seperti halnya varian DE, EO, dan MPA terbaru, SMA tidak dapat diskalakan dan tidak stabil untuk sebagian besar fungsi benchmark.

Hingga saat ini, tidak ada satupun pengoptimal metaheuristik yang menjamin untuk mendapatkan global optima untuk semua masalah dan ukuran dimensi. Hal ini dibuktikan secara logis dengan teorema tidak ada makan siang gratis [43]. Oleh karena itu, banyak peneliti yang terus menerus didorong untuk membuat algoritma baru untuk menyelesaikan masalah sebanyak mungkin, se-scalable mungkin, se-stabil mungkin dalam memberikan tingkat jaminan yang paling tinggi. Termotivasi oleh fakta-fakta tersebut, sebuah algoritma baru yang terukur dan stabil yang diberi nama Algoritma Komodo Mlipir (KMA) diusulkan di sini. Meniru proses mencari makan dan reproduksi komodo, populasi beberapa individu (kandidat solusi) dalam KMA dibagi menjadi tiga kelompok: jantan besar berkualitas tinggi, satu betina berkualitas menengah, dan jantan kecil berkualitas rendah. KMA memperkenalkan dua gerakan baru, yaitu eksploitasi rendah eksplorasi tinggi (HILE) untuk mengintensifkan solusi berkualitas tinggi dan gerakan *mlipir* yang menciptakan eksploitasi tinggi eksplorasi rendah (LIHE) untuk menghindari optimalisasi lokal dan mengatasi area datar yang luas (dataran tinggi).

Makalah ini ditulis dengan struktur sebagai berikut. Bagian awal Penjelasan tentang KMA diberikan di Bagian 2. Selanjutnya, Bagian 3 membahas beberapa hasil eksperimen untuk 23 fungsi benchmark yang dikomparasi menjadi enam algoritma: GA, SHADE, LSHADE-cnEpSin, EO, MPA, dan SMA. Selanjutnya, analisis konvergensi, skalabilitas, dan stabilitas untuk fungsi-fungsi berdimensi lebih tinggi kemudian diberikan untuk KMA yang diusulkan dan kelima algoritma tersebut. Diskusi komprehensif kemudian diberikan pada Bagian 4. Kesimpulan akhirnya diberikan pada Bagian 5.

## 2. Usulan KMA

Bagian ini membahas dua inspirasi dari KMA yang diusulkan: perilaku Komodo dan gaya berjalan khas Jawa "*mlipir*". Konsep dan model matematis kemudian diberikan secara rinci dengan ilustrasi dan formulasi yang jelas. Terakhir, pseudocode dari KMA diberikan.

### 2.1. Komodo di alam liar

Komodo (*Varanus komodoensis*) adalah biawak dari keluarga Varanidae [44] yang hidup di Pulau Komodo dan beberapa pulau tetangga di Kepulauan Sunda Kecil di dekat Flores, Nusa Tenggara Timur, Indonesia. Biawak ini dapat tumbuh hingga 3 meter dengan berat sekitar 135 kg (300 pon) [45]. Komodo memiliki perilaku yang unik dalam mencari makan dan bereproduksi.

Komodo dewasa yang berukuran besar memakan mangsa berukuran besar, seperti rusa, kambing, babi hutan, dan bangkai. Selain itu, mereka juga memangsa anak-anak yang lebih kecil dan terkadang bahkan komodo dewasa lainnya [46]. Ketika seekor komodo besar memakan mangsa, beberapa komodo lain tertarik untuk bergabung [47]. Namun, Komodo besar bisa bersedia atau menolak untuk berbagi mangsa.

Komodo betina dapat menghasilkan keturunan dengan mengawini jantan besar dewasa atau terkadang melalui partenogenesis (reproduksi aseksual) [44,48]. Beberapa komodo jantan besar sering melakukan pertarungan berdarah untuk memperebutkan mangsa atau betina. Pemenangnya dapat menyimpan mangsa atau mengawini betina (jika dia mau) untuk reproduksi seksual [47]. Misalkan si betina tidak ingin kawin dengan pemenang. Dalam hal ini, betina melakukan partenogenesis [48].

Sementara itu, pejantan kecil mencari makan dengan mencari sisa-sisa makanan yang ditinggalkan oleh pejantan besar. Namun, mereka harus menjaga jarak atau akan di kanibal oleh pejantan besar. Oleh karena itu, mereka menyingkir dari pejantan besar. Begitu ada kesempatan, mereka mendekati sisa makanan.

## 2.2. Mlipir

"Mlipir" adalah sebuah kata dalam bahasa Jawa yang dapat diartikan sebagai "berjalan di pinggir jalan untuk menghindari bahaya" atau "berjalan di pinggir jalan dengan hati-hati, tanpa diketahui orang lain, dan berhasil mencapai tujuan tertentu dengan selamat". Ini adalah gaya berjalan khas Jawa untuk mencapai tujuan dengan selamat.

Seperti yang dijelaskan pada Bagian 2.1, jantan kecil menggunakan strategi pergerakan seperti *mlipir* saat mencari makan. Mereka mencari sisa makanan yang ditinggalkan oleh komodo jantan besar dan menghindari kanibalisme dari komodo jantan besar. Oleh karena itu, mereka mengendap-endap di sekitar komodo jantan besar. Begitu ada kesempatan, mereka mengambil sisa makanan dengan aman.

## 2.3. Konsep KMA

Konsep KMA secara sederhana dapat diilustrasikan pada Gambar 1. Diberikan sebuah optimasi masalah sederhana dari fungsi dua dimensi  $f(x_1, x_2) = x_1^2 + x_2^2$ , di mana  $x_1$  dan  $x_2$  adalah horisontal dan vertikal

dan solusi optimum global (target) adalah  $f = 0$  pada  $x_1 = 0$  dan  $x_2 = 0$ . Pertama, biarkan sebuah populasi awal yang terdiri dari enam individu Komodo (kandidat solusi):  $k_1, k_2, \dots, k_6$  dibangkitkan secara acak, di mana setiap Komodo hanya merepresentasikan sebuah vektor posisi (tanpa kecepatan) dari sebuah kandidat solusi di dalam ruang permasalahan. Selanjutnya, populasi dibagi menjadi tiga kelompok: jantan besar berkualitas tinggi, betina berkualitas menengah, dan jantan kecil berkualitas rendah. Di sini, jantan besar berkualitas tinggi meniru Komodo terkuat yang menangkap satu-satunya mangsa, yang mewakili solusi terbaik sejauh ini.

Evolusi kemudian dilakukan dalam urutan tertentu. Pertama, besar berinteraksi satu sama lain untuk melakukan eksplorasi rendah eksploitasi tinggi (HILE). Selanjutnya, dengan menggunakan probabilitas terdistribusi normal sebesar 0,5, betina akan kawin dengan jantan besar terbaik atau partenogenesis (reproduksi asexual), melakukan eksploitasi sedang atau eksplorasi sedang (MIME). Terakhir, jantan kecil melakukan gerakan *mlipir* yang baru untuk mengelilingi ruang pencarian untuk melakukan eksploitasi rendah atau eksplorasi tinggi (LIHE).

Sebagai jantan besar, dua komodo  $k_1$  dan  $k_2$  bertarung satu sama lain. Hal ini diadopsi dalam model ini sebagai daya tarik dan daya tolak dengan menggunakan aturan tertentu. Jantan besar yang berkualitas lebih rendah harus bergerak maju ke arah jantan besar yang berkualitas lebih tinggi. Sebaliknya, pejection besar yang berkualitas lebih tinggi dapat bergerak maju atau mundur ke pejection yang berkualitas lebih rendah dengan probabilitas 0,5. Oleh karena itu, beberapa pejection besar melakukan eksploitasi tinggi (intensifikasi) di area sempit di sekitar global optimum sementara yang lain melakukan eksplorasi rendah (diversifikasi) di area yang lebih luas. Seperti yang diilustrasikan pada Gambar 1,  $k_2$  tertarik pada  $k_1$  (ditunjukkan oleh  $w_{21}$ ) yang membuatnya bergerak menuju optimum global (eksplorasi). Pergerakan ini meniru perilaku komodo untuk bertarung merebut mangsa yang disimpan oleh jantan besar lainnya. Sebaliknya,  $k_1$  terdistraksi dari  $k_2$  (ditunjukkan oleh  $w_{12}$ ) secara acak dengan probabilitas 0.5 dan melakukan eksplorasi. Gerakan ini meniru perilaku Komodo untuk menjaga mangsanya sendiri. Selain itu, seperti kebanyakan algoritma metaheuristik, KMA dirancang untuk mempertahankan pejection besar dengan kualitas terbaik.

Selanjutnya, betina akan mengawini individu komodo pemenang (dengan kebugaran tertinggi) untuk menghasilkan dua keturunan (ditunjukkan oleh  $g_1$  dan  $g_2$ ) atau melakukan partenogenesis dengan gerakan acak (ditunjukkan oleh  $h$ ). Oleh karena itu, sebagai kualitas menengah, betina mencari solusi dengan cara eksploratif-medium-eksploratif.

Akhirnya, sebagai jantan kecil yang berkualitas rendah, tiga ekor komodo  $k_4, k_5$ , dan  $k_6$  melakukan *mlipir* untuk menyisihkan jantan besar dengan cara memilih secara acak sebagian dari dimensi mereka. Pemilihan dimensi dilakukan dengan menggunakan distribusi normal dengan probabilitas tetap sebesar 0,5. Pergerakan ini merupakan metode baru dalam bidang optimasi metaheuristik. Di sini, pergerakan ini dinamakan "*mlipir*" dan probabilitas porsi dimensi dinamakan "*laju mlipir*". Pada 1,

"*laju mlipir*" adalah 0.5, yang berarti pejection kecil mengikuti pejection besar hanya pada separuh dimensi. Sebagai contoh,  $k_4$  bergerak mengesampingkan  $k_1$  dan  $k_2$  dengan hanya menggunakan dimensi pertama  $x_1$  (horisontal), yang ditunjukkan oleh  $w_{41}$  dan  $w_{42}$ . Oleh karena itu, penjumlahan kedua vektor  $w$  menciptakan pergerakan yang membuat  $k_4$  berjalan ke samping dan mendapatkan posisi baru yang mendekati  $k_{(2)}$ . Sebaliknya,  $k_5$  bergerak ke samping  $k_1$  dan  $k_2$  dengan hanya menggunakan dimensi kedua  $x_2$  (vertikal), yang ditunjukkan oleh  $w_{51}$  dan  $w_{52}$ . Penjumlahan ketiga vektor  $w$  akan memindahkan  $k_5$  ke atas, mengelilingi ruang pencarian. Akhirnya,  $k_6$  secara vertikal mengikuti  $k_1$  dan secara horisontal mengikuti  $k_2$  (ditunjukkan oleh  $w_{61}$  dan  $w_{62}$ ), yang secara diagonal menggerakkan  $k_6$  ke depan menuju titik optimal global. Menariknya, gerakan *mlipir* ini tidak hanya melakukan eksplorasi yang tinggi tetapi juga dapat melakukan eksploitasi yang rendah. Oleh karena itu, gerakan *mlipir* yang baru memainkan peran penting dalam KMA yang diusulkan. Karena populasi pada generasi awal memiliki keragaman yang tinggi, gerakan *mlipir* ini membuat pejection kecil melakukan eksplorasi yang tinggi untuk mencakup seluruh ruang pencarian. Pada iterasi terakhir, mereka melakukan eksplorasi yang rendah.

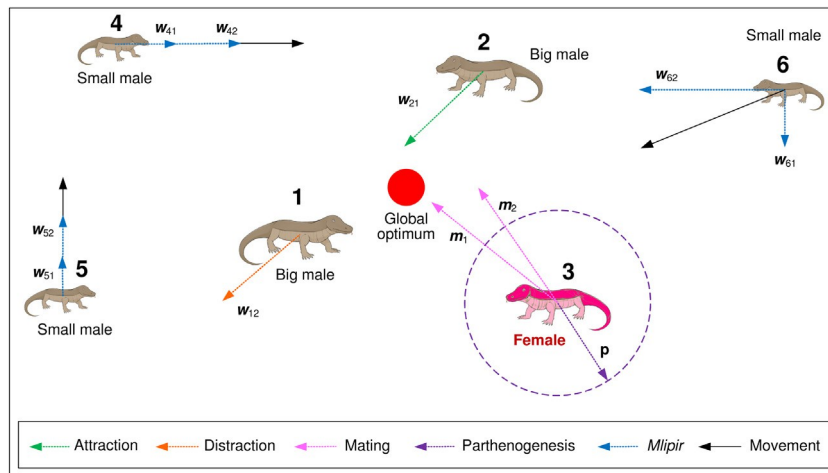
Ketiga gerakan jantan besar, betina, dan kecil

jantan, dapat diperluas ke ruang dimensi tinggi, seperti m-dimensi. Pejection besar saling menarik dan mengalihkan perhatian satu sama lain di seluruh m-dimensi sementara pejection kecil melakukan gerakan *mlipir* ke pejection besar hanya di dimensi  $dm$ , di mana  $d$  adalah laju *mlipir* yang direpresentasikan sebagai bilangan riil dalam interval (0, 1) untuk memilih bagian dari dimensi: 1, 2, ...,  $(m - 1)$ . Selain itu, semua posisi baru dari pejection kecil disimpan tanpa seleksi survivor karena pejection besar telah tampil.

## 2.4. Pseudocode dari KMA

Konsep KMA dapat dijelaskan dengan menggunakan pseudocode pada Algoritma 1. Pertama, tiga parameter yaitu  $n$ ,  $p$ , dan  $d$  ditetapkan masing-masing sebagai jumlah Komodo, porsi jantan besar, dan laju *mlipir*. Selanjutnya,  $n$  Komodo dibuat secara acak. Setelah itu, evolusi dilakukan hingga kriteria berhenti tercapai. Pada setiap generasi, semua Komodo dievaluasi berdasarkan tujuan dari masalah yang diberikan. Kualitas kemudian diurutkan untuk menentukan peringkat mereka. Berdasarkan peringkat, populasi dibagi menjadi tiga kelompok:  $q$  jantan besar berkualitas tinggi, satu betina berkualitas menengah, dan  $s$  jantan kecil berkualitas rendah berdasarkan Persamaan (1) dan (2). Setiap jantan besar dipindahkan menggunakan Persamaan (4), dan posisi  $q$  terbaik dipertahankan untuk bertahan hidup di generasi berikutnya. Kemudian, betina melakukan reproduksi seksual atau asexual berdasarkan Persamaan (5) dan (7). Selanjutnya, setiap jantan kecil dipindahkan menggunakan Persamaan (9), dan semua posisi baru mereka dipertahankan untuk bertahan hidup di generasi berikutnya (tanpa seleksi yang selamat). Terakhir, populasi diperbarui secara adaptif menggunakan Persamaan (10). Setelah kriteria penghentian tercapai, Komodo  $k_{\text{terbaik}}$  dengan kualitas  $k_{\text{terbaik}}$  akan dikembalikan sebagai solusi optimal global.

Pseudocode KMA diimplementasikan dalam dua fase untuk menangani fungsi unimodal dan multimodal secara efektif dan efisien. Pada fase pertama, KMA menggunakan ukuran populasi rendah dengan setengah porsi jantan besar dan laju *mlipir* maksimum. Fase ini dijalankan selama 1000 generasi untuk memeriksa kompleksitas masalah yang diberikan. Jika masalah yang diberikan memiliki kompleksitas yang rendah (unimodal), KMA akan mendapatkan global optimum dengan cepat. Setelah optimum global tercapai, KMA mengembalikan solusi dan menghentikan evolusi. Jika tidak, fase kedua diterapkan hingga evaluasi fungsi maksimum tercapai atau KMA mencapai optimal global. Pada fase kedua, KMA menggunakan populasi yang lebih tinggi dengan prosedur adaptasi diri untuk secara otomatis menyesuaikan ukuran populasi selama proses evolusi dengan setengah porsi jantan besar dan setengah laju *mlipir*. Pada fase ini, KMA diharapkan dapat menyelesaikan benchmark yang kompleks (multimodal).



Gbr. 1. Konsep Algoritma Komodo Mlipir.

**Algoritma 1:** Algoritma Komodo Mlipir

**Hasil:**  $K$  (*terbaik*) sebagai solusi optimum global

Tetapkan  $n$ ,  $p$ , dan  $d$  masing-masing sebagai jumlah individu komodo, porsi jantan besar, dan laju *mlipir*; Inisialisasi  $n$  individu dengan  $m$  dimensi;

**while** *StoppingCriterion* = *false* **do**

**untuk** setiap Komodo, menghitung kualitasnya, dan kemudian memberi peringkat;

    Berdasarkan peringkat mereka dan porsi  $p$ , bagi populasi menjadi tiga kelompok:  $q$  jantan besar berkualitas tinggi, 1 betina berkualitas menengah, dan  $s$  jantan kecil berkualitas rendah dengan menggunakan Persamaan (1) dan (2);

**untuk** setiap jantan besar, pindahkan dengan menggunakan Persamaan (4), dan pertahankan  $q$  jantan besar dengan kualitas terbaik (posisi terbaik) untuk bertahan hidup di generasi berikutnya;

    Perbarui betina dengan mengawinkan jantan besar pemenang menggunakan Persamaan (5) atau melakukan partenogenesis menggunakan Persamaan (7); **untuk** setiap jantan kecil, pindahkan menggunakan Persamaan (9), dan pertahankan semua posisi barunya untuk bertahan hidup di generasi berikutnya; Perbarui ukuran populasi  $n$  menggunakan Persamaan (10);

    Pilih Komodo dengan kualitas tertinggi dari ketiga kelompok sebagai solusi terbaik sejauh ini  $K_{\text{terbaik}}$

**akhir**

## 2.5. Representasi individu Komodo

Dalam KMA, individu Komodo dinyatakan sebagai vektor bernilai real  $K$  dengan  $m$  dimensi, yang menentukan posisi dalam ruang masalah. Pada 1, populasi terdiri dari enam Komodo,  $K_1, K_2, \dots, K_6$  dengan dimensi  $m = 2$ . Seperti kunang-kunang di FA, setiap Komodo di KMA didesain hanya memiliki posisi tanpa kecepatan. Komodo bergerak lebih dinamis dalam ruang pencarian karena kelembaman diabaikan dalam pergerakannya.

## 2.6. Tiga kelompok individu

Populasi  $n$  individu Komodo dibagi menjadi tiga kelompok: jantan besar berkualitas tinggi, betina berkualitas menengah, dan betina berkualitas rendah.

jantan kecil berkualitas dengan porsi  $p$ , yang berada dalam interval  $(0, 1)$  dan secara umum dapat disetel sebagai 0,5 untuk masalah tertentu. Dengan menggunakan porsi  $p$ , populasi  $n$  Komodo dibagi menjadi  $q$  besar, satu betina, dan  $s$  jantan kecil dengan menggunakan rumus berikut

$$q = \lfloor (p - 1)n \rfloor \quad (1)$$

$$s = n - q, \quad (2)$$

Namun, porsi  $p$  yang terlalu kecil atau terlalu besar dapat menghasilkan nilai nol untuk  $q$  atau  $s$ . Oleh karena itu, prosedur sederhana diterapkan untuk memberlakukan jantan besar dan jantan kecil dengan setidaknya dua interaksi Komodo.

## 2.7. Pergerakan jantan besar

Jantan besar berinteraksi dengan ketertarikan atau gangguan berdasarkan aturan sederhana yang diperkenalkan dalam penelitian ini. Jantan besar berkualitas rendah akan tertarik dengan jantan berkualitas tinggi. Sebaliknya, jantan besar berkualitas tinggi dapat tertarik atau terganggu oleh jantan yang berkualitas lebih rendah secara acak dengan probabilitas 0,5 untuk mendapatkan posisi baru. Skema ini memastikan peluang eksploitasi selalu lebih tinggi daripada eksplorasi. Oleh karena itu, skema pergerakan baru ini dinamakan eksploitasi tinggi eksplorasi rendah (HILE).

Pergerakan seekor jantan besar  $K_i$  untuk menghasilkan posisi baru  $K'_i$  dapat didefinisikan dengan menggunakan dua rumus berikut

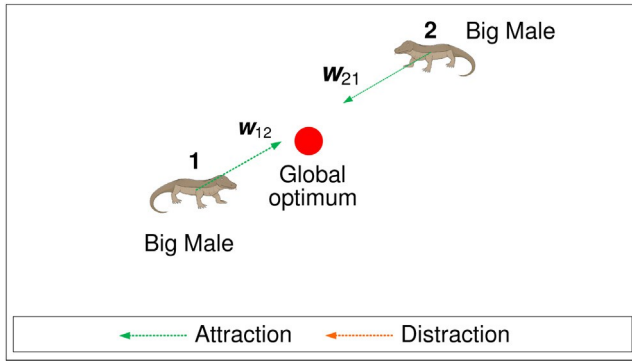
$$w = \begin{cases} r_{(1)}(K_j - K_i), & \text{jika } f(K_j) < f(K_i) \text{ atau } r_2 < 0.5 \\ r_1(K_j - K_i), & \text{jika tidak,} \end{cases} \quad (3)$$

$$K'_i = K_i + \sum_{j=1}^q w_{ij}, \text{ di mana } j \neq i, \quad (4)$$

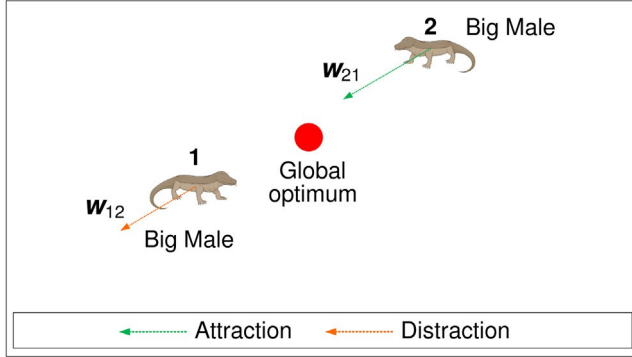
di mana  $f(K_i)$  dan  $f(K_j)$  masing-masing adalah kebugaran (atau kualitas) dari jantan besar ke- $i$  dan ke- $j$ ,  $K_i$  dan  $K_j$  adalah jantan besar ke- $i$  dan ke- $j$  (di mana  $j \neq i$ ), masing-masing,  $r_{(1)}$  dan  $r_{(2)}$  adalah angka acak dalam interval  $[0, 1]$  pada distribusi normal, dan  $q$  adalah jumlah jantan besar.

Jika ada dua jantan besar, maka jantan yang berkualitas rendah akan melakukan eksploitasi, dan jantan yang berkualitas tinggi dapat melakukan eksploitasi atau eksplorasi secara acak dengan probabilitas 0,5. Oleh karena itu, ada dua kemungkinan pergerakan jantan besar tersebut seperti yang diilustrasikan pada Gbr. 2. Pada kemungkinan pergerakan 1, kedua jantan besar melakukan eksploitasi





(a) Possible movement 1 of two big males



(b) Possible movement 2 of two big males

Gbr. 2. Dua kemungkinan gerakan dari dua jantan besar.

mencari di area-area yang dekat dengan solusi global. Sementara itu, pada gerakan possible 2, Big Male 2 yang berkualitas rendah melakukan eksploitasi dengan menarik Big Male 1 yang berkualitas lebih tinggi. Sebaliknya, Big Male 1 melakukan eksplorasi dengan mengalihkan perhatian Big Male 2. Oleh karena itu, dalam kasus ini, probabilitas eksploitasi lebih tinggi daripada eksplorasi.

Jika populasi terdiri dari Gbr. tiga jantan besar, terdapat delapan kemungkinan pergerakan dari tiga besar, seperti yang diilustrasikan pada 3. Pada kemungkinan pergerakan 1, ketiga pejantan besar melakukan eksploitasi mendekati optimal global. Sementara itu, pada kemungkinan pergerakan 2, 3, 4, dan 5, dua jantan besar melakukan eksploitasi, dan sisanya melakukan eksplorasi. Terakhir, pada kemungkinan pergerakan 6, 7, dan 8, hanya satu pejantan besar yang melakukan eksploitasi, dan sisanya melakukan eksplorasi. Oleh karena itu, pada semua kemungkinan gerakan, setidaknya satu jantan besar melakukan eksploitasi. Tidak ada kemungkinan ketiga pejantan besar melakukan eksplorasi. Oleh karena itu, dapat dijamin bahwa probabilitas eksploitasi harus lebih tinggi daripada eksplorasi.

Namun, harus ada jumlah jantan besar yang optimal untuk melakukan interaksi. Secara hipotesis, dua atau tiga pejantan besar akan memberikan interaksi optimal untuk menyeimbangkan eksploitasi dan eksplorasi. Untuk menyelesaikan fungsi unimodal, dua jantan besar sudah cukup. Namun demikian, untuk menyelesaikan fungsi multimodal, tiga jantan besar akan lebih baik. Hipotesis ini terbukti secara empiris berdasarkan beberapa percobaan awal. Konsep interaksi ini berbeda dengan EO [36] yang menggunakan lima partikel sebagai kandidat kesetimbangan untuk memberikan performa yang optimal.

## 2.8. Reproduksi betina

Sebagai individu berkualitas menengah, seekor betina dirancang untuk melakukan eksploitasi atau eksplorasi dalam kesempatan yang sama. Hal ini diimplementasikan dengan menggunakan probabilitas tetap sebesar 0,5. Jika eksploitasi yang dipilih, betina akan mengintensifkan area potensial di sekitar

solusi optimal global dengan mengawinkan pejantan besar pemenang (kualitas tertinggi) untuk menghasilkan dua keturunan (posisi baru). Di sisi lain, algoritma ini mendiversifikasi solusi di area yang luas dari ruang pencarian dengan melakukan partenogenesis. Pada akhirnya, sang betina diperbarui oleh keturunan yang dihasilkan terbaik. Selanjutnya, prosedur perkawinan (reproduksi seksual) dari dua ekor komodo dirumuskan sebagai

$$\begin{aligned} k'_{il} &= r_l - k_{il} + (1 - r_l) - k_{jl} \\ k'_{jl} &= r_l - k_{jl} + (1 - r_l) - k_{il} \end{aligned} \quad (5)$$

di mana  $k_{il}$  dan  $k_{jl}$  adalah dimensi ke- $i$  dari ke- $j$  Komodo sebagai pemenang besar pria dan wanita, masing-masing,  $k'$  adalah dimensi ke- $k$  dari dua keturunan yang dihasilkan oleh proses perkawinan, dan  $r_k$  adalah sebuah bilangan acak dalam interval  $[0, 1]$  dalam distribusi normal untuk dimensi ke- $l$ . Karena perkawinan dilakukan secara acak pada setiap dimensi, maka banyak kemungkinan keturunan yang dapat dihasilkan. Sebagai contoh, dua kemungkinan keturunan dihasilkan seperti yang diilustrasikan pada Gambar 4(a). Selanjutnya, betina diperbarui oleh keturunan terbaik seperti yang ditunjukkan pada Gbr. 4(b).

Sementara itu, prosedur partenogenesis (reproduksi aseksual) diimplementasikan dengan menambahkan nilai kecil pada setiap dimensi betina. Angka kecil tersebut dibangkitkan secara acak menggunakan distribusi normal simetris, yang didefinisikan sebagai

$$(k_{i1}, k_{i2}, \dots, k_{im}) \rightarrow (k'_{i1}, k'_{i2}, \dots, k'_{im}) \quad (6)$$

$$k'_{ij} = k_{ij} + (2r - 1)\alpha |ub_j - lb_j|. \quad (7)$$

dimana  $k_{i1}, k_{i2}, \dots, k_{im} \in [lb_j, ub_j]$  adalah elemen vektor dari  $m$  dimensi dari individu Komodo  $k$ ,  $lb_j$  dan  $ub_j$  adalah batas bawah dan batas atas dari dimensi ke- $j$ ,  $r$  adalah nilai acak dalam distribusi normal, dan  $\alpha$  adalah radius partenogenesis yang ditetapkan dengan nilai tetap 0,1, yang berarti solusi baru dapat dihasilkan dalam radius 10% dari ruang pencarian. Gambar 5(a) mengilustrasikan dua dari sekian banyak kemungkinan keturunan yang dihasilkan:  $p_1$  dan  $p_2$  dalam radius 0,1. Selanjutnya, biarkan  $p_1$  dipilih, dan betina kemudian diperbarui oleh  $p_1$ , seperti yang ditunjukkan pada Gbr. 5(b).

## 2.9. Pergerakan jantan kecil

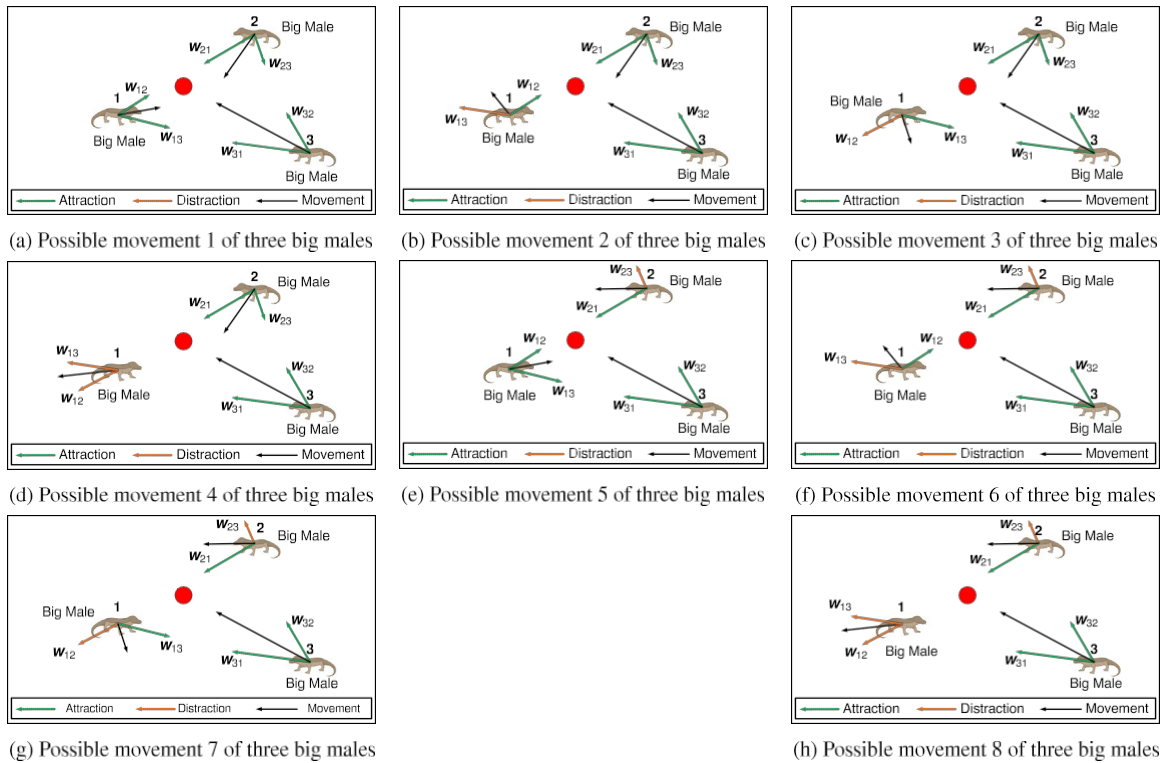
Di alam liar, komodo jantan kecil memiliki dua tujuan: (1) mencari sisa makanan yang ditinggalkan oleh komodo jantan besar, dan (2) menghindari kanibalisme oleh komodo jantan besar dengan cara menyingkir (*mlipir*). Perilaku ini diadopsi di KMA dengan menciptakan gerakan baru yang disebut *mlipir*. Sebagai individu berkualitas rendah, jantan kecil didesain untuk mencari solusi di area yang luas namun terkadang melakukan sedikit eksploitasi. Setiap pejantan kecil bergerak *menyingkir (mlipir)* dari pejantan besar, yang diimplementasikan dengan memilih secara acak bagian dari dimensi mereka dengan probabilitas tertentu (*laju mlipir*). Dengan *laju mlipir*  $d$  dalam interval  $(0, 1)$ , pergerakan *mlipir* komodo ke- $i$  mengikuti komodo ke- $j$  dirumuskan sebagai

$$w_{ij} = \begin{cases} r_1(k_{jl} - k_{il}), & \text{jika } r_2 < d \\ 0, & \text{jika tidak} \end{cases} \quad (8)$$

$$k' = k_i + \sum_{j=1}^q w_{ij}, \text{ di mana } j \neq i, \quad (9)$$

di mana  $r_1$  dan  $r_2$  adalah angka yang dibangkitkan secara acak dalam  $[0, 1]$  menggunakan distribusi normal:  $r_1$  mendefinisikan kecepatan pergerakan sementara  $r_2$  memilih dimensi yang akan diikuti,  $m$  adalah dimensi,  $k_{il}$  dan  $k_{jl}$  adalah dimensi ke- $l$  dari jantan kecil ke- $i$  dan jantan besar ke- $j$ ,  $l$  adalah dimensi yang dipilih secara acak, yang berdasarkan distribusi normal dengan probabilitas yang sama dengan tingkat *mlipir*  $d$  untuk memilih 1, 2, ...,  $(m - 1)$  dimensi jantan besar, dan  $q$  adalah jumlah jantan besar.

Jika ada dua jantan kecil dan dua jantan besar, maka ada sembilan kemungkinan kombinasi gerakan jantan kecil, seperti



Gbr. 3. Delapan kemungkinan gerakan dari tiga jantan besar.

diilustrasikan pada Gbr. 6. Pada empat kemungkinan gerakan 1, 2, 4, dan 5, kedua jantan kecil melakukan pencarian eksploratif dengan bergerak secara horizontal dan vertikal untuk mengelilingi jantan besar: di atas, di bawah, kiri, dan kanan. Sementara itu, pada kemungkinan gerakan 3, 6, 7, 8, dan 9, setidaknya satu jantan kecil melakukan eksploitasi dengan bergerak secara diagonal mendekati jantan besar. Jika pergerakan kedua jantan kecil dihitung secara terpisah, maka 12 dari 18 (66,67%) kemungkinan pergerakan adalah eksplorasi, dan sisanya adalah eksploitasi. Oleh karena itu, dalam pergerakan *mlipir* ini, probabilitas eksplorasi dijamin lebih tinggi daripada eksploitasi.

## 2.10. Skema adaptasi populasi

Seperti yang telah dijelaskan di atas, KMA memiliki tiga parameter: ukuran populasi  $n$ , porsi  $p$ , dan laju *mlipir*  $d$ . Secara hipotesis,  $n$  lebih sensitif daripada  $p$  dan  $d$  karena  $n$  sangat mengontrol strategi eksplorasi dan eksploitasi, sementara  $p$  dan  $d$  dapat didefinisikan sebagai nilai tetap 0,5 karena karakteristik dari masalah yang diberikan tidak diketahui. Oleh karena itu, sebuah skema adaptasi diusulkan untuk menyatel  $n$  secara adaptif selama evolusi. Jika dua kali peningkatan fitness terbaik secara berurutan, ukuran populasi  $n$  akan dikurangi dengan menghapus lima individu. Sebaliknya, jika mereka menunjukkan stagnasi,  $n$  meningkat dengan menghasilkan lima individu baru; di sini, individu baru dibuat dari jantan besar terbaik sejauh ini yang dipindahkan secara acak. Ukuran populasi baru  $n'$  dihitung sebagai

$$n^{(i)} = \begin{cases} -a, & \text{jika } \delta f_1 > 0 \text{ dan } \delta f_2 > 0 \\ n+a, & \text{jika } \delta f_1 = 0 \text{ dan } \delta f_2 = 0 \end{cases} \quad (10)$$

di mana  $a$  adalah jumlah individu yang akan dihapus atau dibangkitkan,  $\delta f_1 = \frac{f_1 - f_2}{f_1}$  dan  $\delta f_2 = \frac{f_2 - f_3}{f_2}$  adalah perbedaan kebugaran dari dua sukses generasi  $ke-i$  dan  $(i-1)$  generasi  $ke-i$ . Di sini, nilai  $a$  ditetapkan menjadi lima, yang mewakili ukuran populasi kecil yang digunakan pada fase pertama. Selain itu, nilai awal, minimum, dan maksimum dari  $n$  harus ditentukan secara konsekuen. Beberapa pengamatan awal menginformasikan bahwa nilai optimumnya adalah 200, 20, dan 200.

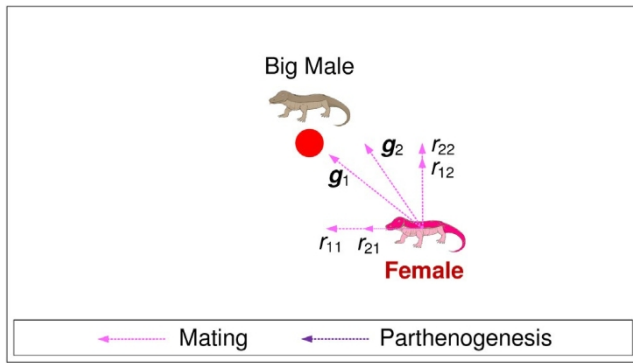
## 2.11. Perbedaan KMA dan algoritma lainnya

KMA yang diusulkan berbeda dengan algoritma metaheuristik yang sudah ada karena menggabungkan tiga strategi pencarian: HILE, MIME, dan LIHE untuk menemukan solusi optimal global dengan cepat dan menghindari solusi lokal. Pergerakan HILE dan LIHE sama sekali berbeda dengan yang digunakan pada algoritma yang sudah ada, sedangkan MIME sama dengan crossover dan mutasi pada GA.

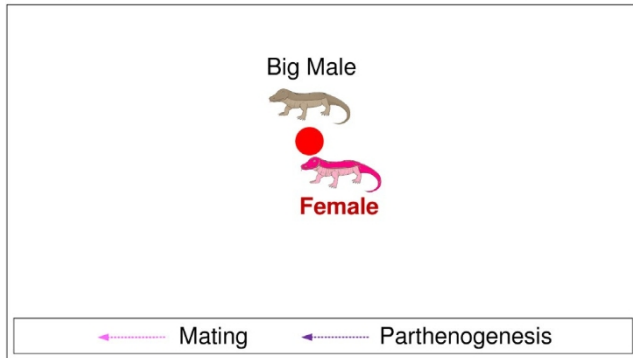
HILE dirancang dengan atraksi dan distraksi antara individu berkualitas tinggi dengan probabilitas 0,5, yang berbeda dengan atraksi dan distraksi pada DFA [25] yang selalu (dengan probabilitas 1,0) dilakukan berdasarkan dua aturan: capung tertarik pada individu yang lebih baik, dan mengalihkan perhatian pada individu yang lebih buruk. Sementara itu, LIHE melakukan diversifikasi tinggi dengan intensitas rendah menggunakan gerakan *mlipir* dari individu berkualitas rendah untuk menghindari *local optima*. Tidak seperti mutasi diferensial dalam DE yang dihitung dari vektor tidak langsung dari dua hingga empat individu terbaik atau individu yang dipilih secara acak [49], gerakan *mlipir* terdiri dari vektor daya tarik langsung dari dimensi parsial dari satu hingga tiga individu berkualitas lebih tinggi yang dipilih secara acak. Seperti yang diilustrasikan pada Gbr. 6, gerakan *mlipir* ini secara logis memberikan strategi eksploitasi yang lebih tinggi dibandingkan dengan mutasi diferensial pada DE.

## 2.12. Biaya komputasi KMA

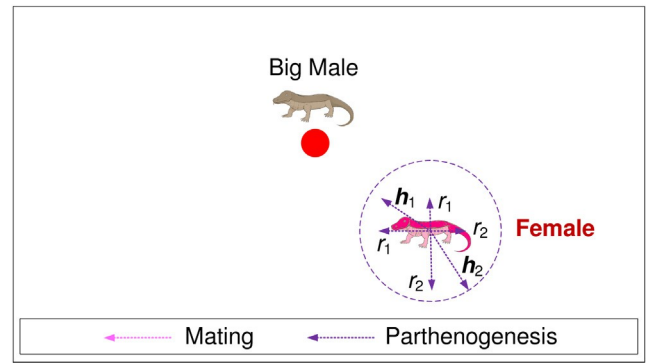
Biaya komputasi KMA dapat dengan mudah diperkirakan berdasarkan analisis kompleksitas. Kompleksitas waktu KMA untuk satu generasi (iterasi) adalah  $O(nmnc + \log n)$ , di mana  $n$ ,  $m$ ,  $c$ , dan  $\log n$  berturut-turut adalah jumlah individu dalam populasi, dimensi, perhitungan fungsi objektif, dan proses pengurutan nilai fitness. Dibandingkan dengan GA, EO, MPA, dan SMA, KMA yang diusulkan sedikit lebih rumit. KMA memiliki kompleksitas tambahan sebesar  $\log p$  yang berasal dari proses pengurutan. Selain itu, KMA juga memiliki kompleksitas lain dari skema adaptasi populasi, meskipun cukup rendah.



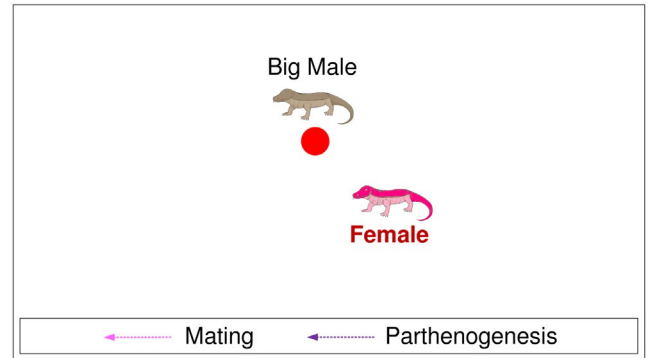
(a) Possible matings of female and the winning big male



(b) New female offspring after mating

**Gbr. 4.** Dengan probabilitas terdistribusi normal sebesar 0,5, betina mengawini jantan besar pemenang, kemudian betina diperbarui oleh keturunan terbaik (jika lebih baik dari betina).

(a) Possible parthenogenesis



(b) New female offspring after doing parthenogenesis

**Gbr. 5.** Dengan probabilitas terdistribusi normal sebesar 0,5, betina melakukan partenogenesis untuk menghasilkan keturunan, kemudian betina diperbarui jika keturunannya lebih baik dari dirinya sendiri.

### 3. Hasil percobaan

23 fungsi tolok ukur klasik yang umum digunakan: 7 unimodal dimensi tinggi (HDU), 6 multimodal dimensi tinggi (HDM), dan 10 multimodal dimensi tetap (FDM) seperti yang dijelaskan dalam [34] digunakan di sini untuk menguji tiga kemampuan KMA: eksploitasi, eksplorasi, dan skalabilitas. Penjelasan rinci dari 23 fungsi benchmark klasik dan plotnya dalam dua dimensi digambarkan pada Tabel 6 dan Gambar . 8 di Lampiran Tabel 6. Tujuh fungsi tolok ukur unimodal: F1, F2, . . . F7 (dengan hanya satu nilai optimum) menguji strategi eksploitasi. Sebaliknya, fungsi-fungsi multimodal: F8, F9, . . . F13 (dengan banyak nilai optimum lokal) akan memeriksa eksplorasi. Sepuluh fungsi multimodal berdimensi rendah yang tetap: F14, F15, . . . F23 mengevaluasi kemampuan eksplorasi dimensi rendah. Selain itu, fungsi F1 hingga F13 akan ditingkatkan menjadi 50, 100, 500, dan 1000 dimensi untuk menilai skalabilitas.

#### 3.1. Pengaturan parameter

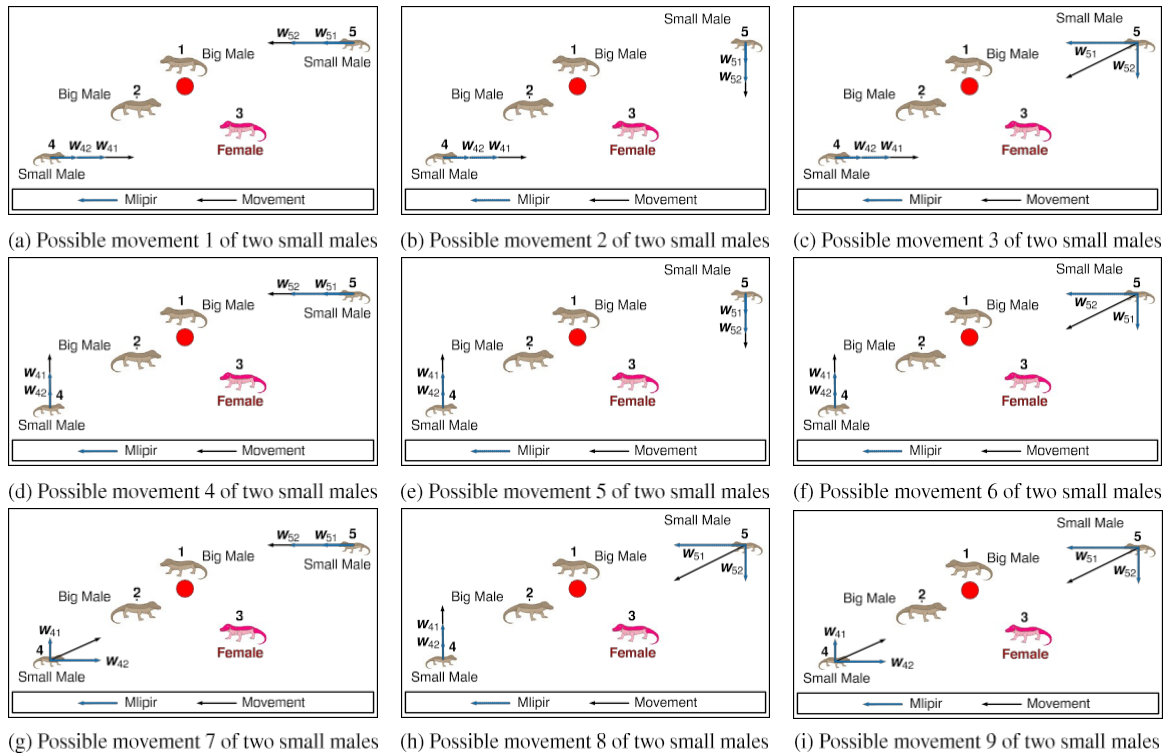
KMA dirancang untuk memiliki tiga parameter yang mudah disesuaikan: ukuran populasi  $n$ , porsi  $p$ , dan laju *mlipir*  $d$ . Pada fase pertama evolusi, KMA menggunakan populasi yang rendah yaitu lima individu ( $n_1 = 5$ ), setengah porsi jantan besar ( $p_1 = 0.5$ ), dan laju *mlipir* maksimum ( $d_1 = \frac{(m-1)}{2}$ ). Fase pertama ini dijalankan selama 1000 generasi (atau 5000 evaluasi) untuk mendapatkan solusi optimum global dengan cepat untuk masalah sederhana (unimodal). Pada fase kedua, KMA disetel untuk menggunakan populasi yang besar yaitu dua ratus individu ( $n_2 = 200$ ) dengan prosedur adaptasi diri untuk secara otomatis menyesuaikan ukuran populasi dalam interval antara 20 dan 200 selama

proses evolusi ( $n_{2,min} = 20$ ,  $n_{2,max} = 200$ ), setengah bagian dari *jantan* besar ( $p_2 = 0.5$ ), dan setengah laju *mlipir* ( $d_2 = 0.5$ ). Pada tahap kedua, KMA diharapkan dapat menyelesaikan fungsi benchmark yang lebih kompleks (multimodal), yang mengandung banyak optima lokal dan daerah datar. Semua pengaturan parameter tersebut didasarkan pada beberapa percobaan awal.

Parameter optimum untuk KMA dan kompetitor dirangkum dalam Tabel 1. Enam pengaturan pertama diadopsi dari penelitian sebelumnya di [9,11,36,37,37], dan [39]. Di sini, semua pengaturan parameter ditetapkan untuk semua fungsi benchmark untuk menguji skalabilitas dengan jumlah evaluasi fungsi yang terbatas, yaitu 25.000. GA dirancang untuk menggunakan populasi yang tinggi  $n = 100$  kromosom pengkodean nyata, seleksi parsial berbasis turnamen dengan ukuran turnamen 10, crossover aritmatika keseluruhan dengan probabilitas  $p_c = 0.8$ , dan mutasi merayap dengan probabilitas  $p_m = 0.05$  agar sederhana untuk ribuan dimensi. SHADE juga didesain untuk mengeksploitasi populasi yang tinggi  $n = 100$ , dengan probabilitas untuk menggunakan individu terbaik sebesar 0.1 dan tingkat arsip sebesar 2 (yang berarti ukuran maksimum arsip adalah  $2n$ ). Seperti yang dijelaskan dalam [11], LSHADE-cnEpSin direkomendasikan untuk menggunakan ukuran populasi yang bervariasi sebesar  $18 \times m$ . Namun, ukuran populasinya ditetapkan pada  $18 \times 5 = 90$  karena ukuran populasi yang besar memiliki performa yang lebih buruk untuk evaluasi fungsi yang terbatas.

#### 3.2. Perbandingan dengan pengoptimal lain

Selanjutnya, KMA dibandingkan dengan enam algoritma: GA, SHADE, LSHADE- cnEpSin, EO, MPA, dan SMA dengan menggunakan 23 fungsi benchmark yang diilustrasikan pada Gbr. . 8, di mana deskripsi rinci mereka dapat ditemukan di [36] Di sini, jumlah evaluasi yang diperbolehkan (yang dihasilkan



Gbr. 6. Sembilan kemungkinan gerakan dari dua jantan kecil.

**Tabel 1**  
Pengaturan parameter KMA dan pesaing.

Algoritma	Parameter
GA [37]	$n = 100$ , ukuran seleksi turnamen = 10, <i>crossover aritmatika</i> $u_{th}$ $p_c = 0.8$ , mutasi merayap $p_m = 0.05$
TEDUH [9]	$n = 100$ , $P_{best} = 0.1$ , Arc rate = 2
LSHADE-cnEpSin [11]	$n = 90$ , $n_{min} = 4$ , $P_{best}$ rate = 0.11, Arc rate = 1.4, Memori = 5, $p_r = 0.5$ , $p_c = 0.4$
EO [36]	$n = 30$ , $a_1 = 2$ , $a_2 = 1$ , GP = 0.5
KAWASAN KONSERVASI PERAIRAN (KKP) [37]	$n = 50$ , $runpon = 0.2$ , $P = 0.5$
SMA [39]	$n = 30$ , $z = 0.03$
KMA	$n_1 = 5$ , $n_2 = 200$ , $n_{2,min} = 20$ , $n_{2(c),max} = 200$ , $p_1 = 0.5$ , $p_2 = 0.5$ , $d_1 = \frac{(m-1)}{m}$ , $d_2 = 0.5$

individu) untuk setiap fungsi benchmark dibatasi hingga 25.000 dengan 30 percobaan independen untuk memberikan dua hasil statistik yang berarti: solusi rata-rata (Avg) dan deviasi standar (Std).

Tabel 2 menunjukkan temuan-temuan dari evaluasi dengan menggunakan kedua metrik tersebut (Met): Angka yang dicetak tebal menunjukkan hasil terbaik di antara semua kompetitor, angka yang digarisbawahi menginformasikan bahwa solusi optimum global selalu dicapai oleh algoritma dengan Std sama dengan nol atau kurang dari angka desimal solusi global, dan angka yang dicetak tebal dan digarisbawahi menunjukkan bahwa algoritma tersebut mencapai solusi terbaik dan juga solusi optimum global.

Berdasarkan Avg dan Std, KMA diusulkan menjamin solusi optimum global untuk sebagian besar (17 dari 23) fungsi: lima unimodal, dua multimodal berdimensi tinggi, dan kesepuluh multimodal berdimensi rendah yang telah ditetapkan. Hampir mencapai optima global untuk F7, F8, F10, F12, dan F13. Yang mengesankan, meskipun F7 memiliki tingkat noise yang tinggi, KMA dapat mencapai optimum yang hampir global. KMA memberikan solusi yang secara signifikan berbeda dari optimal global hanya untuk F5. Fungsi F5 cukup sulit untuk diselesaikan oleh KMA dan semua kompetitornya karena memiliki daerah datar yang luas (dataran tinggi) yang membuat semua algoritma terjebak dalam stagnasi.

Sementara itu, MPA menjamin untuk mendapatkan solusi optimum global untuk tiga belas fungsi: satu fungsi unimodal, dua fungsi multimodal berdimensi tinggi, dan kesepuluh fungsi multimodal berdimensi rendah yang telah ditetapkan. Metode ini menghasilkan solusi yang mendekati global optima untuk delapan fungsi dan solusi yang jauh dari global optima untuk dua fungsi (F5 dan F8). SMA menjamin untuk menemukan global optima untuk sepuluh fungsi. Ini

memberikan solusi yang mendekati global optima untuk dua belas fungsi dan solusi yang lebih buruk hanya untuk F5. Selanjutnya, EO menjamin untuk mencapai global optima hanya untuk enam fungsi. Algoritma ini menghasilkan solusi yang mendekati global optima untuk dua belas fungsi dan solusi yang jauh lebih buruk untuk lima fungsi. Tiga algoritma lainnya, SHADE, LSHADE-cnEpSin, dan GA menghasilkan hasil yang jauh lebih buruk.

Berdasarkan ukuran statistik bernama Friedman Mean Rank (FMR), GA memberikan FMR terburuk sebesar 5,13 diikuti oleh LSHADE- cnEpSin dan SHADE yang masing-masing mencapai 4,57 dan 3,43. EO, MPA, dan SMA menghasilkan peringkat yang lebih baik masing-masing 3,17, 2,35, dan 2,13. Akhirnya, KMA mencapai peringkat terbaik yaitu 1,52, seperti yang diilustrasikan pada Tabel 2. Secara umum, KMA mencapai hasil yang lebih baik (atau sama) daripada yang lainnya untuk 18 dari 23 fungsi. KMA menghasilkan solusi yang sedikit lebih buruk daripada SMA hanya untuk F7 dan F8 dan jauh lebih buruk daripada kompetitor hanya untuk F5, F12, dan F13. Hasil ini menunjukkan keunggulan KMA.

Dalam hal jaminan optimal global (Gua), yang didefinisikan sebagai kepastian menemukan optimal global dengan Std sama dengan nol atau lebih dapat diabaikan daripada digit desimal dari optimal global, KMA juga jauh lebih baik daripada para pesaing. KMA menjamin untuk mencapai solusi optimum global untuk 17 dari 23 (73,91%) fungsi benchmark, yang jauh lebih tinggi GA, SHADE, LSHADE-cnEpSin, EO, MPA, dan SMA yang memberikan jaminan 5 dari 23 (21,74%), 9 dari 23 (39,13%), 5 dari 23 (21,74%), 7 dari 23 (30,43%), 13 dari 23 (56,52%), dan 10 dari 23 (43,48%).



Tabel 2

Perbandingan KMA dan para pesaing untuk tiga belas fungsi benchmark 50 dimensi (F1 hingga F13) dan sepuluh fungsi benchmark tetap berdimensi rendah (F14 hingga F23) untuk 30 kali running, masing-masing 25.000 evaluasi fungsi. Kedua simbol  $-$  dan  $+$  (dalam tanda kurung) menginformasikan bahwa hasil saat ini secara signifikan lebih buruk dan lebih baik daripada hasil KMA berdasarkan uji jumlah peringkat Wilcoxon (Wil) dengan tingkat signifikansi 0,05, sedangkan simbol  $\approx$  menunjukkan hasil yang tidak signifikan.

Fungsi	Bertem p	GA	TEDUH	LSHADE-cnEpSin	EO	MPA	SMA	KMA
F1	$\lambda$ ata-rata	4.380E+ 01 ( )	5.484E- 05 ( )	3.136E- 06 ( )	2.334E- 86 ( )	4.138E- 21 ( )	0 ( $\approx$ )	0
	std	5.257E+ 00	2.497E- 05	3.016E- 06	7.436E- 86	4.086E- 21	0	0
F2	$\lambda$ ata-rata	3.063E+ 00 ( )	1.263E- 02 ( )	1.314E- 02 ( )	7.094E- 49 ( )	3.256E- 12 ( )	5.945E- 174 ( )	0
	std	2.347E- 01	3.780E- 03	1.202E- 02	1.583E- 48	2.494E- 12	0.000E+ 00	0
F3	$\lambda$ ata-rata	4.114E+ 03 ( )	4.717E+ 02 ( )	6.163E+ 01 ( )	6.547E- 13 ( )	4.149E- 02 ( )	0 ( $\approx$ )	0
	std	8.527E+ 02	1.674E+ 02	2.887E+ 01	3.408E- 12	8.361E- 02	0	0
F4	$\lambda$ ata-rata	2.708E+ 00 ( )	1.066E+ 00 ( )	4.544E+ 00 ( )	2.193E- 24 ( )	2.787E- 08 ( )	6.137E- 178 ( )	0
	std	3.018E- 01	3.224E- 01	1.151E+ 00	2.657E- 24	1.409E- 08	0.000E+ 00	0
F5	$\lambda$ ata-rata	6.347E+ 02 ( )	5.217E+ 01 ( )	6.054E+ 01 ( )	4.480E+ 01 ( + )	4.532E+ 01 ( + )	<b>7.075E+ 00 ( + )</b>	4.831E+ 01
	std	1.300E+ 02	1.958E+ 01	2.401E+ 01	2.024E- 01	4.179E- 01	1.401E+ 01	1.820E- 01
F6	$\lambda$ ata-rata	4.690E+ 01 ( )	0 ( $\approx$ )	1.230E+ 01 ( )	0 ( $\approx$ )	0 ( $\approx$ )	0 ( $\approx$ )	0
	std	9.159E+ 00	0	7.666E+ 00	0	0	0	0
F7	$\lambda$ ata-rata	8.233E- 02 ( )	2.189E- 02 ( )	1.903E- 02 ( )	1.135E- 03 ( )	1.377E- 03 ( )	<b>1.456E- 04 ( + )</b>	1.715E- 04
	std	3.496E- 02	4.702E- 03	7.689E- 03	4.641E- 04	7.427E- 04	1.583E- 04	1.188E- 04
F8	$\lambda$ ata-rata	- 1.538E+ 04 ( )	- 1.059E+ 04 ( )	- 1.877E+ 04 ( + )	- 1.491E+ 04 ( )	- 1.455E+ 04 ( )	- 2.095E+ 04 ( + )	- 1.701E+ 04
	std	7.506E+ 02	5.241E+ 02	3.637E+ 02	8.352E+ 02	6.191E+ 02	6.814E- 01	2.453E+ 03
F9	$\lambda$ ata-rata	1.213E+ 02 ( )	2.617E+ 02 ( )	2.686E+ 01 ( )	0 ( $\approx$ )	0 ( $\approx$ )	0 ( $\approx$ )	0
	std	1.509E+ 01	1.230E+ 01	5.255E+ 00	0	0	0	0
F10	$\lambda$ ata-rata	2.588E+ 00 ( )	1.605E- 03 ( )	1.499E+ 00 ( )	4.441E- 15 ( )	1.060E- 11 ( )	<b>8.882E- 16 ( )</b> $\approx$	<b>8.882E- 16</b>
	std	1.595E- 01	5.147E- 04	4.012E- 01	0.000E+ 00	5.925E- 12	0.000E+ 00	0
F11	$\lambda$ ata-rata	1.391E+ 00 ( )	3.601E- 04 ( )	5.434E- 03 ( )	0 ( $\approx$ )	0 ( $\approx$ )	0 ( $\approx$ )	0
	std	7.608E- 02	1.366E- 03	7.676E- 03	0	0	0	0
F12	$\lambda$ ata-rata	1.057E- 01 ( )	2.077E- 03 ( + )	2.281E- 02 ( )	<b>2.083E- 07 ( + )</b>	9.452E- 04 ( + )	4.413E- 03 ( )	2.799E- 03
	std	3.319E- 02	1.136E- 02	5.030E- 02	2.463E- 07	1.241E- 03	6.600E- 03	1.815E- 03
F13	$\lambda$ ata-rata	2.253E+ 00 ( )	<b>4.633E- 05 ( + )</b>	2.558E- 02 ( + )	6.054E- 02 ( )	7.316E- 02 ( )	5.307E- 03 ( + )	5.270E- 02
	std	3.612E- 01	2.131E- 05	4.703E- 02	5.858E- 02	6.265E- 02	4.593E- 03	4.689E- 02
F14	$\lambda$ ata-rata	<u>9.980E- 01 ( <math>\approx</math> )</u>	<u>9.980E- 01 ( <math>\approx</math> )</u>	3.925E+ 00 ( )	1.064E+ 00 ( )	<u>9.980E- 01 ( <math>\approx</math> )</u>	<u>9.980E- 01 ( <math>\approx</math> )</u>	<u>9.980E- 01</u>
	std	3.011E- 05	4.123E- 17	4.659E+ 00	3.622E- 01	1.515E- 16	3.680E- 13	4.382E- 16
F15	$\lambda$ ata-rata	1.815E- 03 ( )	<u>3.075E- 04 ( <math>\approx</math> )</u>	5.947E- 04 ( )	1.076E- 03 ( )	<u>3.075E- 04 ( <math>\approx</math> )</u>	5.400E- 04 ( )	<u>3.075E- 04</u>
	std	2.472E- 03	1.135E- 15	1.859E- 04	3.650E- 03	5.890E- 15	3.337E- 04	5.878E- 15
F16	$\lambda$ ata-rata	<u>- 1.032E+ 00 ( <math>\approx</math> )</u>	<u>- 1.032E+ 00 ( <math>\approx</math> )</u>	- 1.004E+ 00 ( )	<u>- 1.032E+ 00 ( <math>\approx</math> )</u>	<u>- 1.032E+ 00 ( <math>\approx</math> )</u>	<u>- 1.032E+ 00 ( <math>\approx</math> )</u>	<u>- 1.032E+ 00</u>
	std	8.681E- 06	8.192E- 06	1.490E- 01	9.888E- 06	9.302E- 06	9.342E- 06	9.779E- 06
F17	$\lambda$ ata-rata	<u>3.979E- 01 ( <math>\approx</math> )</u>	<u>3.979E- 01 ( <math>\approx</math> )</u>	<u>3.979E- 01 ( <math>\approx</math> )</u>	<u>3.979E- 01 ( <math>\approx</math> )</u>	<u>3.979E- 01 ( <math>\approx</math> )</u>	<u>3.979E- 01 ( <math>\approx</math> )</u>	<u>3.979E- 01</u>
	std	2.863E- 05	3.269E- 05	2.967E- 05	3.401E- 05	3.393E- 05	3.024E- 05	3.514E- 05
F18	$\lambda$ ata-rata	<u>3.000E+ 00 ( <math>\approx</math> )</u>	<u>3.000E+ 00 ( <math>\approx</math> )</u>	3.989E+ 00 ( )	<u>3.000E+ 00 ( <math>\approx</math> )</u>	<u>3.000E+ 00 ( <math>\approx</math> )</u>	<u>3.000E+ 00 ( <math>\approx</math> )</u>	<u>3.000E+ 00</u>
	std	4.252E- 05	1.327E- 14	4.937E+ 00	1.473E- 14	1.714E- 14	1.601E- 11	4.103E- 10
F19	$\lambda$ ata-rata	<u>- 3.861E+ 00 ( <math>\approx</math> )</u>	<u>- 3.861E+ 00 ( <math>\approx</math> )</u>	<u>- 3.861E+ 00 ( <math>\approx</math> )</u>	<u>- 3.861E+ 00 ( <math>\approx</math> )</u>	<u>- 3.861E+ 00 ( <math>\approx</math> )</u>	<u>- 3.861E+ 00 ( <math>\approx</math> )</u>	<u>- 3.861E+ 00</u>
	std	7.195E- 04	6.576E- 04	6.727E- 04	1.273E- 03	6.650E- 04	7.448E- 04	8.201E- 04
F20	$\lambda$ ata-rata	- 3.289E+ 00 ( )	- 3.317E+ 00 ( + )	<u>- 3.320E+ 00 ( <math>\approx</math> )</u>	- 3.271E+ 00 ( )	<u>- 3.320E+ 00 ( <math>\approx</math> )</u>	- 3.238E+ 00 ( )	<u>- 3.320E+ 00</u>
	std	5.283E- 02	2.145E- 02	3.996E- 04	6.234E- 02	3.105E- 04	5.478E- 02	3.165E- 04
F21	$\lambda$ ata-rata	- 7.326E+ 00 ( )	- 9.9848 ( )	<u>- 10.1532 ( <math>\approx</math> )</u>	- 9.9618 ( )	<u>- 10.1532 ( <math>\approx</math> )</u>	- 10.1530 ( )	<u>- 10.1532</u>
	std	3.569E+ 00	9.224E- 01	6.224E- 15	9.236E- 01	4.687E- 11	1.528E- 04	6.452E- 15
F22	$\lambda$ ata-rata	- 8.507E+ 00 ( )	<u>- 10.4029 ( <math>\approx</math> )</u>	<u>- 10.4029 ( <math>\approx</math> )</u>	- 9.3141 ( )	<u>- 10.4029 ( <math>\approx</math> )</u>	- 10.4027 ( )	<u>- 10.4029</u>
	std	3.233E+ 00	8.105E- 06	1.117E- 05	2.219E+ 00	9.030E- 06	1.298E- 04	1.014E- 05
F23	$\lambda$ ata-rata	- 8.165E+ 00 ( )	<u>- 10.5364 ( <math>\approx</math> )</u>	- 10.0985 ( )	- 10.5294 ( )	<u>- 10.5364 ( <math>\approx</math> )</u>	- 10.5362 ( )	<u>- 10.5364</u>
	std	3.460E+ 00	2.130E- 06	1.694E+ 00	3.847E- 02	2.172E- 06	1.321E- 04	2.538E- 06
Friedman		5.13	3.43	4.57	3.17	2.35	2.13	<b>1.52</b>
Peringkat		7	5	6	4	3	2	<b>1</b>
Wil ( - )		18	11	16	14	8	8	
Wil ( $\approx$ )		5	9	5	7	13	11	
Wil ( + )		0	3	2	2	2	4	
Gua (%)		21.74	39.13	21.74	30.43	56.52	43.48	<b>73.91</b>

### 3.3. Analisis kurva konvergensi

Evaluasi terperinci kemudian dilakukan pada kurva konvergensi. Pertama, kurva konvergensi dianalisis terhadap fungsi-fungsi modal tunggal (F1 hingga F7). **Gambar 7(a)** menunjukkan kurva konvergensi dari ketujuh algoritma: GA, SHADE, LSHADE-cnEpSin, EO, MPA, SMA, dan KMA untuk fungsi F1. Sumbu vertikal dan horizontal masing-masing mewakili  $\log$  dari solusi rata-rata dan generasi (iterasi). Karena algoritma-algoritma tersebut memiliki ukuran populasi yang berbeda-beda, mereka menggunakan ukuran langkah yang berbeda, masing-masing 9, 9, 10, 30, 18, 30, dan 180, pada sumbu horizontal untuk mendapatkan keadilan. Dapat dilihat bahwa KMA konvergen paling cepat pada generasi awal. Hasil ini juga terjadi pada lima fungsi unimodal lainnya: F2, F3, F4, F6, dan F7. KMA juga konvergen dengan cepat untuk F5 pada generasi awal. Hal ini menunjukkan bahwa tiga gerakan yang diusulkan bekerja dengan sangat baik pada generasi awal.

fase evolusi. Sayangnya, ia mengalami stagnasi hingga akhir evolusi dan memperoleh optimum lokal yang lebih buruk di sekitar 1,68 ( $\log$  48,31), seperti yang diilustrasikan pada **Gbr. . 7(b)**Fungsi F5 menjadi tantangan karena area yang datar, membuat KMA (dan semua algoritma lainnya) terjebak dalam stagnasi. Kurva ini menunjukkan bahwa populasi yang tinggi pada fase kedua evolusi KMA tidak dapat melakukan koordinasi yang tepat di lanskap fitness yang datar. Hal ini terjadi karena sebagian besar (atau bahkan semua) individu memiliki nilai (kualitas) fitness yang sama, membuat KMA kesulitan untuk membagi populasi menjadi jantan besar, betina, dan jantan kecil berdasarkan kualitas mereka.

Kurva konvergensi kemudian dianalisis untuk fungsi multimodal (F8 hingga F13). **Gambar 7(c)** mengilustrasikan kurva konvergensi dari semua algoritma untuk F8. Mirip dengan SMA, KMA yang diusulkan konvergen dengan cepat dibandingkan dengan yang lain pada generasi awal. Sayangnya, algoritma ini mengalami stagnasi dan mencapai solusi yang lebih buruk daripada SMA yang mendapatkan solusi terbaik. Sebaliknya, algoritma

algoritma lain konvergen jauh lebih lambat dan memberikan solusi yang jauh lebih buruk. Sementara itu, [Gbr. 7\(d\)](#) menunjukkan kurva konvergensi untuk F9. Dalam kasus ini, KMA konvergen paling cepat dibandingkan dengan para pesaingnya. KMA mencapai solusi minimum 0 (global optimum) pada generasi awal, dengan evaluasi fungsi rata-rata terendah sebesar 145.33, dimana garis putus-putus merah tidak muncul karena  $\log(0)$  tidak terbatas. Sebaliknya, EO, MPA, dan SMA konvergen lebih lambat sementara GA, SHADE, LSHADE-cnEpSin tidak dapat mencapai global optimum. Kurva konvergensi ini juga terjadi pada F11.

Analisis konvergensi akhirnya dilakukan untuk sepuluh fungsi multimodal berdimensi tetap. [Gbr. 7\(e\)](#) memplot kurva kekonvergenan untuk F17. Semua algoritma dapat konvergen ke solusi global. Dalam kasus ini, KMA konvergen lebih lambat pada generasi awal karena hanya menggunakan lima individu pada fase pertama evolusi. Kelima individu tersebut mengalami kesulitan untuk berkoordinasi di beberapa daerah datar di F17. Setelah mengubah ukuran populasi menjadi 200 individu pada fase kedua evolusi, KMA mencapai titik optimal global dengan cepat. Hasil yang sama juga terjadi pada F15, F16, dan F18. Sementara itu, [Gambar 7\(f\)](#) menunjukkan kurva konvergensi untuk F19. Sama halnya dengan kurva F17, KMA juga mengalami konvergensi yang lebih lambat pada generasi awal karena pada fase pertama evolusi, KMA hanya menggunakan lima individu yang mengalami kesulitan untuk berkoordinasi pada lanskap fitness yang luas pada F19. Hasil ini juga berlaku untuk F14, F20, F21, F22, dan F23.

Oleh karena itu, dapat digarisbawahi bahwa semua kurva konvergensi tersebut mengindikasikan bahwa KMA merupakan algoritma metaheuristik yang paling efisien di antara para pesaingnya. Hasil ini membuktikan bahwa ketiga gerakan (strategi HILE, MIME, dan LIHE) dan skema adaptasi populasi yang diusulkan dalam KMA dapat memberikan keseimbangan eksploitasi-eksplorasi dan sangat menjamin solusi optimal global untuk berbagai fungsi benchmark.

### 3.4. Analisis skalabilitas

Skalabilitas dianalisis dengan memeriksa kinerja KMA dibandingkan dengan para pesaing untuk mengoptimalkan versi dimensi tinggi dari ketiga belas fungsi: F1 hingga F13, di mana dimensinya (Dim) ditingkatkan dari 50 menjadi 100, 500, dan 1000. Jumlah individu yang dibangkitkan (evaluasi fungsi) juga dibatasi hingga 25.000 dengan 30 kali percobaan independen untuk mendapatkan rata-rata (Avg), deviasi standar (Std), dan rata-rata evaluasi fungsi (MFE). [Tabel 3](#), [4](#), dan [5](#) mengilustrasikan hasilnya. Angka yang dicetak tebal menunjukkan hasil terbaik di antara para pesaing. Angka yang digarisbawahi menunjukkan bahwa solusi optimum global selalu tercapai dengan Std sama dengan nol atau kurang dari angka desimal optimum global. Angka yang digarisbawahi tebal menginformasikan bahwa angka tersebut mencapai solusi terbaik dan juga solusi optimal global.

Pada semua dimensi tinggi (100, 500, dan 1000), KMA menjamin untuk mendapatkan global optimum untuk 7 dari 13 (53,85%) fungsi benchmark, yang jauh lebih tinggi daripada SMA, EO, MPA, SHADE, LSHADE-cnEpSin, dan GA yang masing-masing memberikan jaminan 38,46%, 23,08%, 23,08%, 0%, 0%, dan 0%. Selain itu, KMA secara mengesankan membutuhkan MFE yang kecil. Misalnya, ia hanya membutuhkan 55,83, 62,33, dan 65,17 MFE untuk mendapatkan solusi optimum global untuk fungsi unimodal F6 100, 500, dan 1000 dimensi, yang jauh lebih sedikit daripada EO, MPA, dan SMA yang membutuhkan lebih dari 1000 MFE.

Untuk 5 dari 7 fungsi unimodal, KMA stabil dan memberikan jaminan solusi optimal global (dengan Std = 0). mengesankan, KMA stabil dengan MFE yang rendah (dari 55,83 hingga 4221,67) untuk semua dimensi, jauh lebih rendah daripada semua kompetitor. Untuk F7, KMA mencapai solusi yang hampir global dengan Std yang rendah, yang jauh lebih baik daripada GA, SHADE, LSHADE-cnEpSin, EO, dan MPA. Ini kompetitif dengan SMA. Sedikit tidak stabil hanya untuk F5. Untungnya, hasil ini lebih baik daripada SHADE dan LSHADE-cnEpSin dan kompetitif dengan EO dan MPA. Oleh karena itu, hasil ini menyiratkan bahwa KMA jauh lebih baik

dibandingkan para pesaing dalam hal kemampuan eksploitasi dan stabilitas dalam mengoptimalkan fungsi unimodal berdimensi tinggi.

Sementara itu, untuk dua fungsi multimodal: F9 dan F11, KMA menjamin optimal global dengan Std (nol) minimum untuk semua dimensi. Selain itu, KMA membutuhkan MFE yang sangat rendah (dari 150,50 hingga 183,00). Untuk fungsi F10, KMA juga relatif stabil untuk menemukan solusi yang mendekati global dengan Std yang rendah, yang lebih baik daripada (atau sama dengan) pesaing. Untuk F12 dan F13, meskipun stabil untuk semua dimensi tinggi, KMA sedikit lebih buruk daripada SMA.

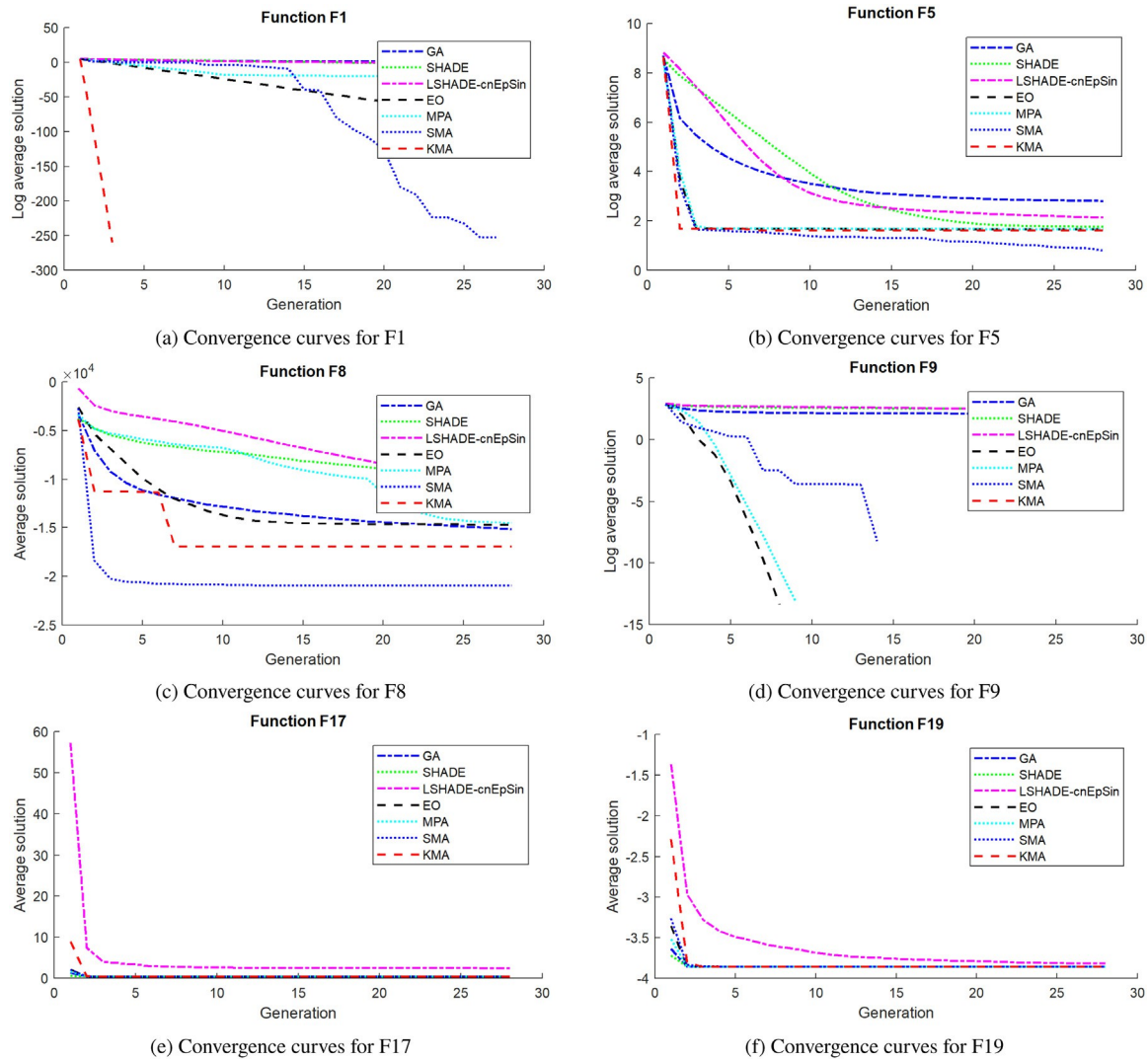
Secara statistik, untuk 500 dan 1000 dimensi, KMA memberikan Friedman Mean Rank (FMR) terbaik, seperti yang diilustrasikan pada [Tabel 4](#) dan

5. Sementara itu, untuk 100 dimensi, KMA memberikan FMR yang lebih rendah daripada SMA, seperti yang diilustrasikan pada [Tabel 3](#). Menariknya, dapat digarisbawahi bahwa dalam hal jaminan optimum global, KMA jauh lebih baik daripada semua kompetitor. Untuk dimensi 50, 100, 500, dan 1000, KMA stabil untuk memberikan jaminan solusi optimum global untuk 7 dari 13 (53,85%) fungsi benchmark, yang jauh lebih tinggi daripada SMA, EO, dan MPA yang tidak stabil: jaminannya menurun dari 3846% menjadi 23,08%, dari 23,08% menjadi 15,38%, dan dari 23,08% menjadi 15,38%, dan jauh lebih baik daripada GA, SHADE, dan LSHADE-cnEpSin yang tidak memberikan jaminan (0%). Hasil ini secara empiris membuktikan bahwa KMA memiliki skalabilitas dan ketahanan yang tinggi dalam keseimbangan eksploitasi-eksplorasi.

## 4. Diskusi

Seperti algoritma berbasis swarm lainnya, sulit untuk memberikan bukti matematis untuk menunjukkan bahwa KMA mampu menyelesaikan masalah unimodal dan multimodal secara efektif dan efisien karena menggunakan individu yang dikodekan dengan bilangan real. Hingga saat ini, hanya GA dengan individu dengan pengkodean biner yang dapat dijelaskan secara matematis menggunakan teorema skema Holland, dan hipotesis blok bangunan [50]. Namun, pengkodean biner tidak dapat diskalakan untuk mengoptimalkan masalah ribuan dimensi yang membutuhkan ketelitian tinggi karena beberapa alasan: mengelola kromosom biner yang panjang terlalu rumit dalam ruang, decoding kromosom yang panjang dan kemudian mengevaluasi kebugaran memakan waktu, mengawinkan pasangan kromosom orang tua yang panjang berdasarkan persilangan satu titik hanya memberikan eksploitasi yang rendah, namun membuat persilangan multi-titik yang lebih baik atau skema lain terlalu sulit, dan mutasi beberapa gen biner akan menghasilkan eksplorasi yang rendah. Oleh karena itu, representasi individu terbaik untuk GA adalah pengkodean bilangan real, seperti yang digunakan dalam KMA dan algoritma lainnya. Akibatnya, kinerja GA, KMA, dan yang lainnya hanya dapat dijelaskan dalam makalah ini dengan menggunakan analisis kualitatif (subyektif) tentang keseimbangan eksploitasi dan eksplorasi.

Pertama, analisis kualitatif dilakukan untuk GA yang menghasilkan kinerja terendah. Seleksi turnamen menciptakan seleksi proporsional orang tua berdasarkan nilai kebugaran mereka, tanpa memperhatikan nilai negatif, nol, atau positif. Eksperimen kami menunjukkan bahwa ukuran turnamen 10 individu, yang merupakan 10% dari ukuran populasi 100, memberikan pemilihan induk yang optimal. Selanjutnya, karena GA menggunakan kromosom pengkode bilangan real, seluruh crossover aritmatika dan mutasi merayap dapat dipertimbangkan untuk memberikan eksplorasi medium eksploitasi. Hasil eksperimen kami menunjukkan bahwa keduanya lebih baik daripada crossover satu titik dan mutasi pada pengkodean bilangan biner atau bilangan bulat. Lebih lanjut, [Tabel 2](#) menunjukkan bahwa GA dapat menjamin global optima untuk 5 dari 10 (50%) fungsi benchmark berdimensi rendah yang telah ditetapkan (F14 dan F16 hingga F19) dan mencapai global optima untuk lima fungsi yang lain. Namun, GA gagal mengoptimalkan tiga belas fungsi dimensi tinggi, bahkan untuk 50 dimensi. Hasil ini menunjukkan bahwa seluruh crossover aritmatika tidak dapat menciptakan eksploitasi yang tinggi untuk menyelesaikan tujuh fungsi unimodal (F1 sampai F7). Demikian juga, mutasi merayap tidak dapat memberikan eksplorasi yang tinggi untuk mengoptimalkan enam fungsi multimodal (F8 hingga F13). Dengan kata lain, semua operator tersebut tidak dapat memberikan skalabilitas pada GA.



Gbr. 7. Kurva konvergen dari KMA dan kompetitor untuk enam fungsi representatif untuk 30 kali percobaan independen, masing-masing 25.000 fungsi evaluasi.

Lebih jauh lagi, SHADE menjamin global optima untuk 8 dari 10 (80%) fungsi benchmark dimensi rendah yang tetap dan mencapai hampir global optima untuk fungsi-fungsi lainnya, sementara LSHADE-cnEpSin hanya mencapai 50%, seperti yang ditunjukkan pada Tabel 2. Namun, baik SHADE maupun LSHADE-cnEpSin juga tidak dapat memberikan skalabilitas. Pada benchmark 50 dimensi, SHADE menjamin optimal global hanya untuk F6 sedangkan LSHADE-cnEpSin tidak memberikan jaminan apa pun, seperti yang ditunjukkan pada Tabel 2. Pada dimensi yang lebih tinggi yaitu 100, 500, dan 1000, kedua algoritma tidak memberikan jaminan apapun, seperti yang digambarkan pada Tabel 3, 4, dan 5. Fakta-fakta ini menunjukkan bahwa mutasi diferensial pada DE, model dasar yang digunakan pada kedua algoritma, tidak dapat memberikan eksploitasi dan eksplorasi yang tinggi. Mereka dihitung dari vektor tidak langsung dari dua hingga empat individu terbaik atau yang dipilih secara acak [49] sehingga individu berkualitas rendah tidak dapat secara langsung mengikuti individu terbaik.

Sementara itu, EO memperoleh FMR yang lebih rendah daripada GA, SHADE, dan

LSHADE-cnEpSin. Meskipun menjamin optimalisasi global hanya untuk 40% dari fungsi dimensi rendah tetap (F16 hingga F19), namun secara menarik memberikan skalabilitas untuk 2 dari 13 (15,38%) tolok ukur dimensi tinggi (F6 dan F9), seperti yang dapat dilihat pada Tabel 5. Hasil ini dapat dijelaskan secara subyektif sebagai berikut. EO membagi populasi tiga puluh partikel (kandidat solusi) menjadi dua kelompok dengan strategi pencarian yang berbeda. Pertama, empat partikel berkualitas tinggi (sebagai kandidat solusi) dengan rata-rata aritmatikanya melakukan eksploitasi yang tinggi. Kedua, dua puluh enam partikel berkualitas rendah melakukan eksploitasi yang tinggi.

eksplorasi. Untuk fungsi unimodal, kandidat ekuilibrium dengan cepat menemukan titik optimum global F6, bahkan untuk 1000 dimensi, tetapi gagal untuk enam dimensi lainnya. Sementara itu, untuk tolok ukur multimodal, partikel berkualitas rendah dapat mengeksplorasi lanskap fitness, dan kandidat ekuilibrium dapat dengan cepat menemukan optimal global F9 untuk semua dimensi. Namun, untuk lima fungsi multimodal lainnya, mereka gagal. Masalah ini muncul karena EO tidak memiliki strategi adaptif untuk secara otomatis menyesuaikan jumlah kandidat ekuilibrium. Seperti yang dijelaskan dalam [36], tiga kandidat ekuilibrium membuat EO lebih eksploitatif untuk menemukan global optima fungsi unimodal dengan cepat. Sebaliknya, lima atau lebih kandidat ekuilibrium membuat EO lebih eksploratif untuk menemukan global optima dari fungsi multimodal dengan cepat. Sayangnya, jenis fungsi yang diberikan tidak diketahui, dan EO tidak memiliki prosedur untuk secara otomatis mendeteksi jenis fungsi tersebut. Menciptakan prosedur seperti itu cukup menantang. Masalah lainnya mungkin berasal dari rata-rata aritmatika yang dihitung dari empat kandidat ekuilibrium. Hal ini dapat membuat EO konvergen terlalu dini (prematur) dan terjebak dalam optima lokal dalam kasus benchmark multimodal.

Selanjutnya, MPA mencapai kinerja yang lebih baik daripada EO. Ini menjamin global optima untuk semua fungsi dimensi rendah tetap (F14 hingga F23) dan juga menyediakan skalabilitas untuk 2 dari 13 (15,38%) tolok ukur dimensi tinggi (F6 dan F9). Pencapaian ini dapat dijelaskan secara subyektif sebagai berikut. Pertama, KKL mempekerjakan 50 individu: 25 pemangsa berkualitas tinggi dan 25 mangsa berkualitas rendah dengan

**Tabel 3**

Perbandingan enam algoritma untuk tiga belas fungsi benchmark 100 dimensi selama 30 kali proses, masing-masing 25.000 evaluasi fungsi. Kedua simbol— dan+ (dalam tanda kurung) menunjukkan bahwa hasil saat ini secara signifikan lebih buruk dan lebih baik daripada hasil KMA berdasarkan uji jumlah peringkat Wilcoxon dengan tingkat signifikansi 0,05, sedangkan≈ menunjukkan hasil yang tidak signifikan.

Fungsi	Bertemu	GA	TEDUH	LSHADE-cnEpSin	EO	MPA	SMA	KMA
F1	Rata-rata	7.638E+ 02 (—)	2.432E— 01 (—)	2.697E+ 00 (—)	1.166E— 74 (—)	4.929E— 19 (—)	0 (≈)	0
	Std	6.477E+ 01	8.656E— 02	1.040E+ 00	1.535E— 74	3.687E— 19	0	0
	MFE	25000	25000	25001	25020	25000	21696	2087.17
F2	Rata-rata	1.950E+ 01 (—)	8.981E— 01 (—)	3.658E+ 00 (—)	7.151E— 43 (—)	2.030E— 11 (—)	4.393E— 178 (—)	0
	Std	1.131E+ 00	3.456E— 01	8.951E— 01	8.133E— 43	1.244E— 11	0.000E+ 00	0
	MFE	25000	25000	25001	25020	25000	25007	3922.5
F3	Rata-rata	2.453E+ 04 (—)	1.257E+ 04 (—)	2.629E+ 03 (—)	6.080E— 05 (—)	4.333E+ 00 (—)	0 (≈)	0
	Std	5.323E+ 03	2.378E+ 03	6.532E+ 02	2.100E— 04	5.581E+ 00	0	0
	MFE	25000	25000	25001	25020	25000	22382	2464.33
F4	Rata-rata	8.272E+ 00 (—)	1.052E+ 01 (—)	1.242E+ 01 (—)	3.241E— 07 (—)	2.702E— 07 (—)	9.698E— 160 (—)	0
	Std	5.292E— 01	1.221E+ 00	1.539E+ 00	1.774E— 06	1.468E— 07	5.312E— 159	0
	MFE	25000	25000	25001	25020	25000	25015	4144.17
F5	Rata-rata	2.932E+ 04 (—)	3.233E+ 02 (—)	6.299E+ 02 (—)	9.531E+ 01 (≈)	9.626E+ 01 (—)	<b>9.883E+ 00 (+)</b>	9.530E+ 01
	Std	6.765E+ 03	8.492E+ 01	2.270E+ 02	6.363E— 01	7.919E— 01	1.878E+ 01	1.467E+ 01
	MFE	25000	25000	25001	25020	25000	25020	25011.33
F6	Rata-rata	7.714E+ 02 (—)	4.300E+ 00 (—)	8.920E+ 01 (—)	0 (≈)	0 (≈)	0 (≈)	0
	Std	6.764E+ 01	2.718E+ 00	2.830E+ 01	0	0	0	0
	MFE	25000	24910	25001	1785	2060	1222	55.83
F7	Rata-rata	2.641E— 01 (—)	1.005E— 01 (—)	1.037E— 01 (—)	1.675E— 03 (—)	1.483E— 03 (—)	<b>2.025E— 04 (+)</b>	2.122E— 04
	Std	9.003E— 02	2.357E— 02	2.673E— 02	7.879E— 04	6.133E— 04	1.722E— 04	1.393E— 04
	MFE	25000	25000	25001	25020	25000	25020	25149.67
F8	Rata-rata	—2.413E+ 04 (—)	—1.334E+ 04 (—)	—2.321E+ 04 (—)	—2.891E+ 04 (—)	—2.615E+ 04 (—)	—4.189E+ 04 (+)	—3.259E+ 04
	Std	1.268E+ 03	6.594E+ 02	6.562E+ 02	1.592E+ 03	9.166E+ 02	7.066E+ 00	4.765E+ 03
	MFE	25000	25000	25001	25020	25000	25020	25045.83
F9	Rata-rata	5.109E+ 02 (—)	7.234E+ 02 (—)	3.496E+ 02 (—)	0 (≈)	0 (≈)	0 (≈)	0
	Std	3.177E+ 01	1.963E+ 01	2.985E+ 01	0	0	0	0
	MFE	25000	25000	25001	5895	6891.67	5210	150.5
F10	Rata-rata	4.960E+ 00 (—)	7.185E— 01 (—)	2.672E+ 00 (—)	5.033E— 15 (—)	6.508E— 11 (—)	<b>8.882E— 16 (≈)</b>	<b>8.882E— 16</b>
	Std	1.284E— 01	4.472E— 01	3.237E— 01	1.347E— 15	2.833E— 11	0.000E+ 00	0.000E+ 00
	MFE	25000	25000	25001	25020	25000	25020	25085
F11	Rata-rata	8.017E+ 00 (—)	1.672E— 01 (—)	7.733E— 01 (—)	0 (≈)	0 (≈)	0 (≈)	0
	Std	4.267E— 01	4.487E— 02	1.330E— 01	0	0	0	0
	MFE	25000	25000	25001	6611	7825	6700	169.83
F12	Rata-rata	2.651E+ 00 (—)	3.780E— 01 (—)	8.041E— 01 (—)	5.759E— 03 (—)	2.800E— 02 (—)	<b>1.216E— 03 (+)</b>	2.407E— 03
	Std	3.679E— 01	2.931E— 01	3.086E— 01	5.622E— 03	7.451E— 03	2.333E— 03	2.832E— 03
	MFE	25000	25000	25001	25020	25000	25020	25012.33
F13	Rata-rata	5.324E+ 01 (—)	8.230E— 01 (—)	1.171E+ 01 (—)	3.255E+ 00 (—)	5.419E+ 00 (—)	<b>8.191E— 02 (+)</b>	1.741E— 01
	Std	5.268E+ 00	7.507E— 01	8.421E+ 00	1.135E+ 00	2.842E+ 00	2.222E— 01	3.133E— 01
	MFE	25000	25000	25001	25020	25000	25020	25009.17
Friedman		6.61	5.31	5.92	2.77	3.23	1.15	<b>1.38</b>
Peringkat		7	5	6	3	4	1	<b>2</b>
Wil (—)	13	13	13	9	10	2		
Wil (≈)	0	0	0	4	3	6		
Wil (+)	0	0	0	0	0	5		
Gua (%)	0	0	0	23.08	23.08	38.46		<b>53.85</b>

strategi pergerakan yang berbeda untuk eksploitasi dan eksplorasi. Skema ini menggunakan probabilitas efek Fish Aggregating De- vices (FAD) sebesar 0,2, membuat predator melakukan lompatan (eksplorasi) yang lebih lama di dimensi yang berbeda secara acak dalam 20% dari jumlah iterasi. Skema ini dapat menghindari terperangkapnya predator dalam optima lokal. Namun, karena MPA membagi evolusi secara merata menjadi tiga fase: eksplorasi tinggi, eksplorasi-eksplorasi yang setara, dan eksploitasi tinggi, MPA tidak dapat menjamin global optima untuk fungsi yang paling unimodal. Selain itu, MPA rentan terjebak dalam optima lokal untuk tolok ukur dimensi yang lebih tinggi.

Sementara itu, SMA memperoleh FMR yang lebih rendah daripada MPA. Meskipun SMA menjamin global optima hanya untuk 5 dari 10 (50%) fungsi dimensi rendah yang tetap (seperti yang ditunjukkan pada Tabel 2), namun SMA memberikan kemampuan skalabilitas untuk 3 dari 13 (23,08%) tolok ukur dimensi tinggi (F6, F9, dan F11), seperti yang ditunjukkan pada Tabel 5. Hasil ini dapat dicapai karena SMA menggunakan bobot adaptif  $W$  untuk mengatur tingkat gangguan tertentu sambil melakukan konvergensi dengan cepat. Oleh karena itu, metode ini dapat menghindari optima lokal selama konvergensi yang cepat. Sayangnya, metode ini tidak dapat diskalakan untuk beberapa tolok ukur karena menggunakan parameter getaran untuk mengeksplorasi secara efisien pada generasi awal dan sangat mengeksplorasi iterasi selanjutnya.

Akhirnya, KMA secara signifikan mengungguli semua pesaing. Ini menjamin optimalisasi global untuk semua (100%) dimensi rendah tetap

fungsi, seperti yang ditunjukkan pada Tabel 2. Selain itu, KMA menjamin optimalisasi global dan juga skalabilitas untuk 7 dari 13 (53,85%) tolok ukur dimensi tinggi, seperti yang digambarkan pada Tabel 5. Hasil ini dapat dicapai karena KMA menggunakan tiga fitur berikut.

Pertama, berbeda dengan EO, MPA, dan SMA yang menggunakan eksploitasi tinggi dan eksploitasi rendah pada generasi awal (tahap pertama) dan strategi yang berlawanan pada generasi akhir (tahap kedua), KMA menggunakan tiga strategi pencarian: HILE, MIME, dan LIHE dalam keseluruhan evolusi (tahap pertama dan kedua). HILE dirancang berdasarkan skema yang jauh berbeda dengan yang digunakan dalam EO terkait pergerakan mula, probabilitas, dan lebih sedikit (satu, dua, atau tiga) individu yang paling cocok. HILE dapat melakukan eksploitasi yang lebih tinggi dan menyatu ke global op- tima dengan lebih cepat untuk fungsi unimodal. Sementara itu, MIME diimplementasikan dengan menggunakan skema yang mirip dengan operator yang digunakan dalam GA. MIME menggunakan crossover aritmatika keseluruhan yang sama GA, namun dengan mutasi yang sedikit berbeda, yaitu dengan memperkenalkan radius mutasi sebesar 0.5 untuk membuat lompatan yang lebih panjang. Sebaliknya, LIHE diimplementasikan dengan menggunakan gerakan *mlipir*, yang terdiri dari vektor daya tarik langsung dengan dimensi parsial dari satu hingga tiga individu berkualitas tinggi yang dipilih secara acak. Seperti yang diilustrasikan pada Gbr. .6, gerakan *mlipir* ini secara logis memberikan strategi eksploitasi yang lebih tinggi daripada mutasi diferensial pada DE



**Tabel 4**

Perbandingan enam algoritma untuk tiga belas fungsi patokan 500 dimensi selama 30 kali proses, masing-masing 25.000 evaluasi fungsi. Kedua simbol— dan+ (dalam tanda kurung) menunjukkan bahwa hasil saat ini secara signifikan lebih buruk dan lebih baik daripada hasil KMA berdasarkan uji jumlah peringkat Wilcoxon dengan tingkat signifikansi 0,05, sedangkan≈ menunjukkan hasil yang tidak signifikan.

Fungsi	Bertemu	GA	TEDUH	LSHADE-cnEpSin	EO	MPA	SMA	KMA
F1	Rata-rata	3.057E+ 04 (—)	1.498E+ 04 (—)	2.072E+ 04 (—)	1.587E— 62 (—)	3.273E— 16 (—)	0 (≈)	0
	Std	1.267E+ 03	1.718E+ 03	2.374E+ 03	3.301E— 62	2.702E— 16	0	0
	MFE	25000	25000	25000	25020	25000	22423	2139
F2	Rata-rata	3.069E+ 02 (—)	2.112E+ 02 (—)	3.253E+ 02 (—)	5.498E— 37 (—)	7.210E— 10 (—)	5.355E— 01 (—)	0
	Std	7.687E+ 00	1.482E+ 01	6.976E+ 01	4.340E— 37	6.922E— 10	2.104E+ 00	0
	MFE	25000	25000	25000	25020	25000	25020	3982.83
F3	Rata-rata	9.780E+ 05 (—)	1.448E+ 06 (—)	1.508E+ 05 (—)	5.591E+ 04 (—)	5.933E+ 03 (—)	3.812E— 303 (—)	0
	Std	1.362E+ 05	1.682E+ 05	2.820E+ 04	7.208E+ 04	3.310E+ 03	0.000E+ 00	0
	MFE	25000	25000	25000	25020	25000	23750	2539.83
F4	Rata-rata	2.977E+ 01 (—)	3.982E+ 01 (—)	1.847E+ 01 (—)	9.501E+ 01 (—)	4.364E— 05 (—)	6.143E— 141 (—)	0
	Std	7.773E— 01	1.579E+ 00	2.791E+ 00	6.427E+ 00	4.672E— 05	2.412E— 140	0
	MFE	25000	25000	25000	25020	25000	25020	4191.67
F5	Rata-rata	6.899E+ 06 (—)	2.730E+ 06 (—)	2.238E+ 06 (—)	4.965E+ 02 (—)	4.965E+ 02 (—)	<b>9.652E+ 01 (+)</b>	4.715E+ 02
	Std	6.357E+ 05	5.643E+ 05	9.571E+ 05	5.158E— 01	3.831E— 01	1.261E+ 02	7.375E+ 01
	MFE	25000	25000	25000	25020	25000	25020	25045.33
F6	Rata-rata	3.069E+ 04 (—)	1.475E+ 04 (—)	2.005E+ 04 (—)	0 (≈)	0 (≈)	0 (≈)	0
	Std	1.397E+ 03	1.405E+ 03	2.008E+ 03	0	0	0	0
	MFE	25000	25000	25000	2437	2451.67	1312	62.33
F7	Rata-rata	5.312E+ 01 (—)	1.950E+ 01 (—)	2.130E+ 01 (—)	2.747E— 03 (—)	1.931E— 03 (—)	2.725E— 04 (—)	<b>2.643E— 04</b>
	Std	5.177E+ 00	3.283E+ 00	2.040E+ 01	1.042E— 03	6.748E— 04	2.671E— 04	1.781E— 04
	MFE	25000	25000	25000	25020	25000	25020	25103.33
F8	Rata-rata	—6.522E+ 04 (—)	—2.556E+ 04 (—)	—3.126E+ 04 (—)	—9.022E+ 04 (—)	—8.763E+ 04 (—)	—2.094E+ 05 (+)	—1.664E+ 05
	Std	4.664E+ 03	1.780E+ 03	1.991E+ 03	4.873E+ 03	3.222E+ 03	1.806E+ 02	2.339E+ 04
	MFE	25000	25000	25000	25020	25000	25020	25057
F9	Rata-rata	4.469E+ 03 (—)	4.773E+ 03 (—)	4.404E+ 03 (—)	0 (≈)	0 (≈)	0 (≈)	0
	Std	1.133E+ 02	4.481E+ 01	1.162E+ 02	0	0	0	0
	MFE	25000	25000	25000	6792	7216.67	7546	157.67
F10	Rata-rata	9.622E+ 00 (—)	7.980E+ 00 (—)	8.523E+ 00 (—)	7.046E— 15 (—)	6.442E— 10 (—)	<b>8.882E— 16 (≈)</b>	<b>8.882E— 16</b>
	Std	1.319E— 01	2.402E— 01	3.125E— 01	1.598E— 15	1.770E— 10	0.000E+ 00	0.000E+ 00
	MFE	25000	25000	25000	25020	25000	25020	25085
F11	Rata-rata	2.763E+ 02 (—)	1.297E+ 02 (—)	1.829E+ 02 (—)	0 (≈)	0 (≈)	0 (≈)	0
	Std	1.158E+ 01	1.468E+ 01	2.420E+ 01	0	0	0	0
	MFE	25000	25000	25000	7996	10531.67	8790	179.5
F12	Rata-rata	5.889E+ 04 (—)	3.252E+ 02 (—)	1.103E+ 02 (—)	3.622E— 01 (—)	3.359E— 01 (—)	<b>3.424E— 03 (+)</b>	1.197E— 02
	Std	3.413E+ 04	6.120E+ 02	5.415E+ 02	1.788E— 02	2.141E— 02	5.155E— 03	9.172E— 03
	MFE	25000	25000	25000	25020	25000	25020	25010.67
F13	Rata-rata	3.395E+ 06 (—)	5.172E+ 05 (—)	1.168E+ 05 (—)	4.885E+ 01 (—)	4.824E+ 01 (—)	<b>4.239E— 01 (+)</b>	2.263E+ 00
	Std	5.043E+ 05	2.135E+ 05	2.922E+ 05	3.293E— 01	3.656E— 01	7.088E— 01	9.714E— 01
	MFE	25000	25000	25000	25020	25000	25020	25011.67
Friedman		6.46	5.77	5.54	3.15	2.77	1.46	1.31
Peringkat		7	6	5	4	3	2	1
Wil (—)	13	13	13	13	10	10	4	
Wil (≈)	0	0	0	0	3	3	5	
Wil (+)	0	0	0	0	0	0	4	
Gua (%)	0	0	0	0	23.08	23.08	30.77	<b>53.85</b>

Kedua, KMA menggunakan prosedur untuk mendeteksi jenis fungsi menggunakan populasi kecil yang terdiri dari 5 individu pada tahap pertama evolusi. Prosedur ini dilakukan hanya untuk 100 generasi (500 evaluasi). Jika tingkat peningkatan kebugaran lebih dari 0.5 (fungsi sederhana terdeteksi), ia melanjutkan proses evolusi hingga maksimum 1000 generasi (5000 evaluasi). Hasil ini menunjukkan bahwa HILE dan MIME dapat secara efektif mengeksplorasi lanskap seperti mangkuk dan dengan cepat mencapai titik optimal global. Yang menarik, hasil ini juga menunjukkan bahwa LIHE dapat membantu HILE dan MIME untuk mengeksplorasi lanskap multimodal secara maksimal (terdapat pada F9 dan F11) sehingga KMA dapat mencapai titik optimal global dalam waktu kurang dari 40 generasi (200 evaluasi). Oleh karena itu, dapat dikatakan bahwa ketiga strategi pencarian tersebut bekerja dengan sangat baik pada tahap ini untuk menjamin optimalisasi global dan skalabilitas untuk 53,85% dari tolok ukur dimensi tinggi secara efisien dalam waktu kurang dari 1000 generasi (5000 evaluasi). Sebaliknya, jika tingkat peningkatan fitness kurang dari 0.5 (fungsi yang kompleks terdeteksi), maka akan beralih ke tahap kedua.

Ketiga, pada tahap kedua evolusi, KMA memanfaatkan adaptasi diri dari populasi besar (20 hingga 200 individu) berdasarkan kebugaran mereka untuk memberikan keseimbangan eksploitasi-eksplorasi. Menambahkan lima individu, yang dihasilkan secara acak dari yang terbaik sejauh ini

solusi, pada populasi meningkatkan tingkat eksplorasi untuk menghindari optima lokal atau keluar dari lanskap kebugaran yang datar. Sebaliknya, menghilangkan lima individu akan meningkatkan tingkat eksploitasi untuk menemukan global optima dengan cepat. Oleh karena itu, tahap kedua ini dapat menjamin optima global untuk semua (100%) tolok ukur dimensi rendah yang tetap.

Namun, KMA memiliki keterbatasan untuk masalah dengan area datar yang luas, seperti fungsi F5. Hal ini disebabkan oleh dua parameter yang tetap pada tahap pertama evolusi, yaitu porsi jantan setengah besar  $p_1 = 0.5$  dan laju *mlipir* yang tinggi  $d_1 = \frac{(m-1)}{m}$ . Area datar yang luas membuat KMA kesulitan membagi lima individu pada tahap pertama menjadi jantan besar, betina, dan jantan kecil karena mereka mungkin memiliki kebugaran yang sangat mirip (bahkan sama). Oleh karena itu, HILE tidak dapat secara efektif melakukan eksploitasi yang tinggi. Sebaliknya, HILE bekerja secara acak dan memilih jantan besar sebagai pemenang. Akibatnya, dengan pemenang yang dipilih secara acak, betina tidak dapat melakukan strategi MIME. Sebaliknya, pejantan kecil tidak terpengaruh oleh situasi ini karena mereka melakukan LIHE dengan memilih pejantan besar secara acak untuk *mlipir*. Demikian juga pada tahap kedua, ketiga strategi tersebut juga tidak dapat bekerja dengan baik karena adanya dua parameter yang tetap: porsi jantan setengah besar  $p_2 = 0.5$  dan tingkat *mlipir* yang tinggi  $d_2 = 0.5$  selama evolusi, yang membuat KMA tidak dapat melakukan eksplorasi yang lebih tinggi.

**Tabel 5**

Perbandingan enam algoritma untuk tiga belas fungsi benchmark 1000 dimensi selama 30 kali proses, masing-masing 25.000 evaluasi fungsi. Kedua simbol  $-$  dan  $+$  (dalam tanda kurung) menunjukkan bahwa hasil saat ini secara signifikan lebih buruk dan lebih baik daripada hasil KMA berdasarkan uji jumlah peringkat Wilcoxon dengan tingkat signifikansi 0,05, masing-masing, sedangkan  $\approx$  menunjukkan hasil yang tidak signifikan.

Fungsi	Bertemu	GA	TEDUH	LSHADE-cnEpSin	EO	MPA	SMA	KMA
F1	Rata-rata	1.104E+ 05 (—)	9.927E+ 04 (—)	5.959E+ 04 (—)	2.879E— 59 (—)	1.718E— 15 (—)	3.413E— 292 (—)	0
	Std	4.031E+ 03	5.047E+ 03	9.258E+ 03	6.916E— 59	1.340E— 15	0.000E+ 00	0
	MFE	25000	25000	25001	25020	25000	23518	2171.17
F2	Rata-rata	8.338E+ 02 (—)	6.554E+ 04 (—)	6.554E+ 04 (—)	1.644E— 35 (—)	1.136E+ 03 (—)	1.223E+ 00 (—)	0
	Std	1.458E+ 01	6.554E+ 04	6.554E+ 04	1.931E— 35	1.490E+ 02	4.522E+ 00	0
	MFE	25000	25000	25001	25020	25000	25020	4021
F3	Rata-rata	4.055E+ 06 (—)	6.989E+ 06 (—)	1.431E+ 06 (—)	1.001E+ 05 (—)	3.510E+ 04 (—)	2.115E— 240 (—)	0
	Std	6.387E+ 05	8.208E+ 05	3.607E+ 05	1.306E+ 05	1.980E+ 04	0.000E+ 00	0
	MFE	25000	25000	25001	25020	25000	2.398E+ 04	2650.5
F4	Rata-rata	4.335E+ 01 (—)	4.903E+ 01 (—)	3.560E+ 01 (—)	9.943E+ 01 (—)	3.442E— 04 (—)	7.211E— 147 (—)	0
	Std	1.047E+ 00	2.192E+ 00	1.165E+ 01	2.702E— 01	4.182E— 04	3.950E— 146	0
	MFE	25000	25000	25001	25020	25000	25020	4221.67
F5	Rata-rata	4.296E+ 07 (—)	3.468E+ 07 (—)	1.271E+ 07 (—)	9.965E+ 02 (—)	9.962E+ 02 (—)	<b>1.374E+ 02 (+)</b>	9.617E+ 02
	Std	2.898E+ 06	4.123E+ 06	1.760E+ 07	1.888E— 01	2.049E— 01	2.225E+ 02	8.468E+ 01
	MFE	25000	25000	25001	25020	25000	25020	25040.67
F6	Rata-rata	1.087E+ 05 (—)	1.046E+ 05 (—)	5.698E+ 04 (—)	0 ( $\approx$ )	0 ( $\approx$ )	0 ( $\approx$ )	0
	Std	3.581E+ 03	7.069E+ 03	8.936E+ 03	0	0	0	0
	MFE	25000	25000	25001	2673	2626.67	1919	65.17
F7	Rata-rata	6.923E+ 02 (—)	4.583E+ 02 (—)	2.687E+ 02 (—)	3.268E— 03 (—)	1.962E— 03 (—)	4.483E— 04 (—)	<b>2.731E— 04</b>
	Std	5.411E+ 01	7.016E+ 01	4.823E+ 02	1.420E— 03	9.409E— 04	3.325E— 04	1.411E— 04
	MFE	25000	25000	25001	25020	25000	25020	25123.33
F8	Rata-rata	—9.853E+ 04 (—)	—3.526E+ 04 (—)	—3.573E+ 04 (—)	—1.366E+ 05 (—)	—1.368E+ 05 (—)	—4.189E+ 05 (+)	—3.279E+ 05
	Std	7.907E+ 03	1.707E+ 03	2.873E+ 03	7.654E+ 03	5.031E+ 03	1.680E+ 02	4.762E+ 04
	MFE	25000	25000	25001	25020	25000	25020	25059.67
F9	Rata-rata	9.707E+ 03 (—)	9.976E+ 03 (—)	9.485E+ 03 (—)	0 ( $\approx$ )	0 ( $\approx$ )	0 ( $\approx$ )	0
	Std	1.339E+ 02	9.388E+ 01	1.358E+ 02	0	0	0	0
	MFE	25000	25000	25001	7169	7386.67	8253	157.83
F10	Rata-rata	1.155E+ 01 (—)	1.152E+ 01 (—)	9.241E+ 00 (—)	7.638E— 15 (—)	1.232E— 09 (—)	<b>8.882E— 16 (<math>\approx</math>)</b>	<b>8.882E— 16</b>
	Std	1.381E— 01	2.372E— 01	3.882E— 01	1.084E— 15	4.100E— 10	0.000E+ 00	0.000E+ 00
	MFE	25000	25000	25001	25020	25000	25020	25085
F11	Rata-rata	9.932E+ 02 (—)	9.272E+ 02 (—)	5.022E+ 02 (—)	3.701E— 18 (—)	9.992E— 17 (—)	0 ( $\approx$ )	0
	Std	3.161E+ 01	7.810E+ 01	8.310E+ 01	2.027E— 17	3.388E— 17	0	0
	MFE	25000	25000	25001	10828	24193.33	7146	183
F12	Rata-rata	5.097E+ 06 (—)	4.692E+ 05 (—)	2.014E+ 05 (—)	6.439E— 01 (—)	5.707E— 01 (—)	<b>5.707E— 03 (+)</b>	1.333E— 02
	Std	8.851E+ 05	2.195E+ 05	8.138E+ 05	2.257E— 02	2.200E— 02	1.080E— 02	1.293E— 02
	MFE	25000	25000	25001	25020	25000	25020	25010.67
F13	Rata-rata	5.130E+ 07 (—)	2.395E+ 07 (—)	4.123E+ 06 (—)	9.914E+ 01 (—)	9.776E+ 01 (—)	<b>2.472E+ 00 (+)</b>	4.218E+ 00
	Std	6.384E+ 06	5.036E+ 06	1.086E+ 07	2.294E— 01	4.164E— 01	6.227E+ 00	2.215E+ 00
	MFE	25000	25000	25001	25020	25000	25020	25013
Friedman		6.31	6.23	5.08	3.38	3.08	1.46	<b>1.31</b>
Peringkat		7	6	5	4	3	2	<b>1</b>
Wil (—)	13	13	13	11	11	5		
Wil ( $\approx$ )	0	0	0	2	2	4		
Wil (+)	0	0	0	0	0	4		
Gua (%)	0	0	0	15.38	15.38	23.08		<b>53.85</b>

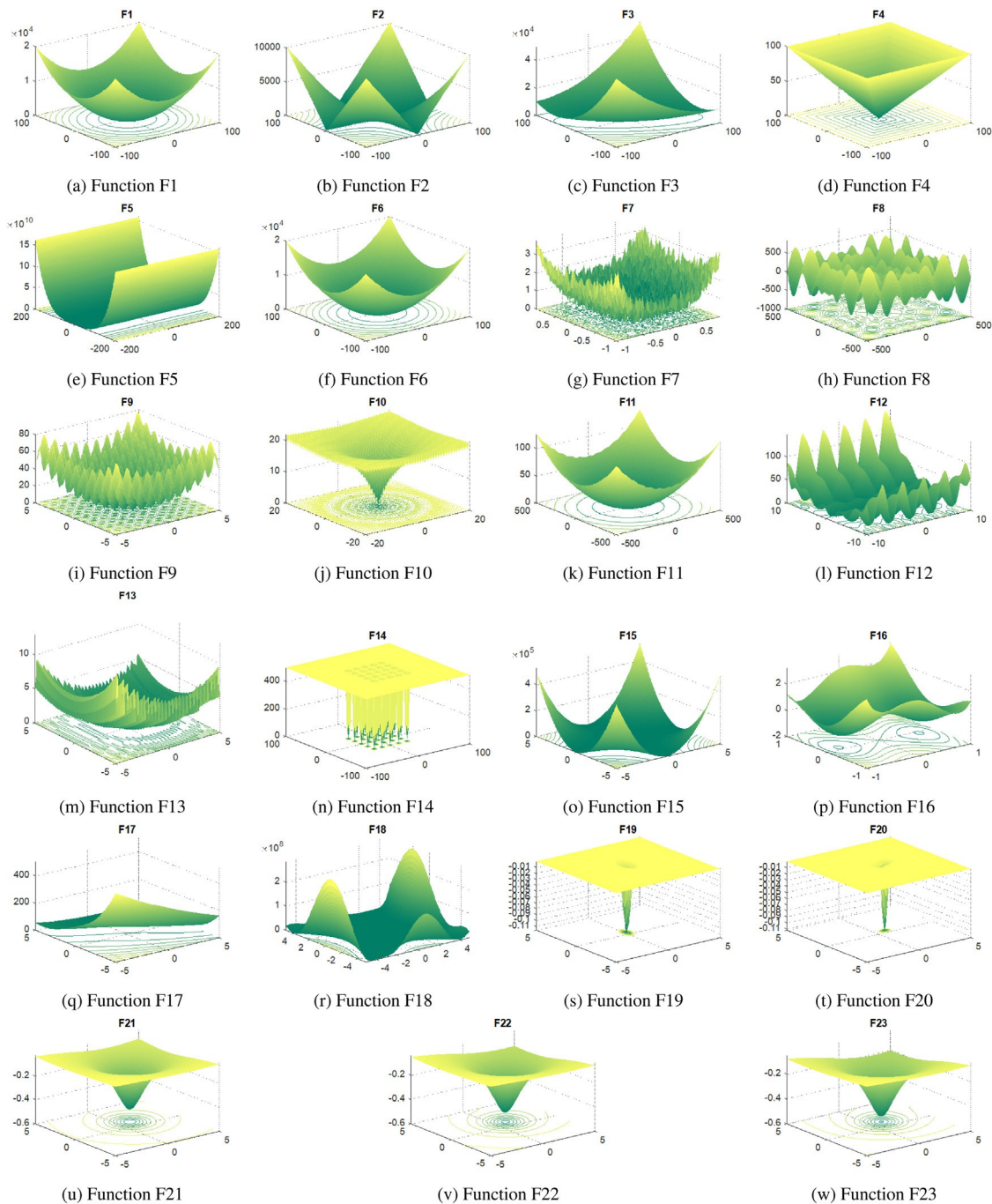
Akhirnya, dapat disimpulkan bahwa KMA dapat memberikan jaminan dan skalabilitas untuk dengan cepat menemukan global optima untuk fungsi unimodal (atau multimodal seperti mangkuk) selama tahap pertama evolusi karena menggunakan tiga strategi yang berbeda: HILE, MIME, dan LIHE dengan populasi yang kecil dan tingkat *mlipir* yang tinggi. KMA juga dapat menjamin global optima untuk berbagai fungsi multimodal berdimensi rendah yang kompleks dan tetap, karena pada tahap kedua evolusi, KMA menggunakan populasi besar yang disetel secara adaptif untuk melakukan tiga strategi yang berbeda dengan tingkat *mlipir* setengah. Analisis kualitatif ini telah divisualisasikan dengan jelas dalam enam kurva konvergensi dari fungsi tolak ukur yang representatif di Bagian 3.3.

## 5. Kesimpulan

Dalam penelitian ini, sebuah pengoptimal metaheuristik baru yang dinamakan Ko- modo Mlipir Algorithm (KMA) telah berhasil dikembangkan untuk memberikan jaminan yang lebih tinggi dalam mencapai optimalisasi global dan kemampuan yang lebih besar hingga seribu dimensi. Dibandingkan dengan algoritma yang ada saat ini (sebagai kompetitor dalam penelitian ini), KMA memiliki tiga keunggulan. Pertama, selama dua tahap evolusi, algoritma ini menggunakan tiga kelompok individu: jantan besar berkualitas tinggi, betina berkualitas menengah, dan jantan kecil berkualitas rendah yang bekerja dengan cara yang berbeda (HILE,

MIME, dan LIHE), memberikan adaptasi yang hebat untuk berbagai jenis fungsi. Kedua, pada tahap pertama evolusi, ia menggunakan populasi kecil yang tetap yang terdiri dari lima individu (dua jantan besar, seekor betina, dan dua jantan kecil), dengan tingkat *mlipir* yang tinggi. Karakteristik ini memberikan jaminan dan skalabilitas untuk mencapai global optima dari fungsi unimodal dan multimodal seperti mangkuk sederhana. Ketiga, pada tahap kedua evolusi, KMA menggunakan populasi besar yang dapat beradaptasi sendiri yang terdiri dari 20 hingga 200 individu dan laju *mlipir* setengah untuk menjamin optima global dari fungsi multimodal berdimensi rendah yang kompleks dan tetap. Namun, KMA memiliki satu kelemahan. Karena KMA menggunakan dua parameter tetap yaitu porsi jantan besar dan laju *mlipir* selama dua tahap evolusi, maka KMA tidak dapat menjamin optima global dari fungsi yang mengandung area datar yang luas.

Hasil eksperimen menunjukkan bahwa KMA secara signifikan mengungguli para pesaing: GA, SHADE, LSHADE-cnEpSin, EO, MPA, dan SMA dalam hal solusi rata-rata, evaluasi fungsi rata-rata, tingkat jaminan, dan skalabilitas. Evaluasi terhadap 23 fungsi benchmark menunjukkan bahwa semua skema yang diusulkan memberikan KMA keseimbangan eksploitasi-eksplorasi yang menjamin global optima untuk sebagian besar fungsi dengan cepat. KMA dapat menjamin global optima untuk 17 dari 23 (73,91%) fungsi, yang lebih tinggi daripada MPA (56,52%), SMA (43,48%), SHADE (39,13%), dan kompetitor lainnya. A



Gbr. 8. Dua puluh tiga fungsi tolok ukur F1 hingga F23.

Evaluasi skalabilitas menginformasikan bahwa ia dapat ditingkatkan hingga ribuan dimensi. Algoritma ini menjamin optimalisasi global untuk 7 dari 13 (53,85%) fungsi secara efisien dalam waktu kurang dari 5000 evaluasi, yang jauh lebih tinggi daripada SMA (23,08%), MPA (15,38%), EO (15,38%), dan algoritma lainnya (0%). Algoritma ini sedikit tidak stabil hanya untuk F5 (mengandung area datar yang luas), tetapi masih lebih baik daripada GA, SHADE, dan LSHADE-cnEpSin serta kompetitif dengan EO dan MPA. Oleh karena itu, KMA jauh lebih baik daripada semua pesaingnya, menjadikannya pengoptimal berbasis swarm metaheuristik yang potensial.

Di masa depan, skema adaptasi mandiri tingkat lanjut dapat dikembangkan untuk memperbarui dua parameter secara dinamis: jantan besar

porsi dan laju *mlipir*, memberikan KMA rasio eksploitasi yang lebih tinggi dan adaptif untuk mengatasi benchmark dengan area datar yang luas. Selain itu, penelitian ini memiliki satu keterbatasan. KMA hanya dievaluasi dengan menggunakan satu set kecil dari 13 tolok ukur dimensi tinggi dan sepuluh tolok ukur dimensi rendah yang sederhana dan difokuskan pada beberapa metrik evaluasi. Oleh karena itu, kinerjanya harus diselidiki secara komprehensif dengan menggunakan fungsi tolok ukur yang lebih menantang dan aplikasi dunia nyata.

**Tabel 6**

Dua puluh tiga fungsi benchmark: tiga belas dimensi tinggi dan sepuluh dimensi rendah tetap.

Fungsi	Nama	Jenis	Dimensi	Jangkauan	$f_{min}$	$x_i^*$
F1	Sphere	HDU	{50, 100, 500, 1000}	$[-100, 100]$	0	(0, 0, ...)
F2	Schwefel 2.22	HDU	{50, 100, 500, 1000}	$[-100, 100]$	0	(0, 0, ...)
F3	Schwefel 1.2	HDU	{50, 100, 500, 1000}	$[-100, 100]$	0	(0, 0, ...)
F4	Schwefel 2.21	HDU	{50, 100, 500, 1000}	$[-100, 100]$	0	(0, 0, ...)
F5	Rosenbrock	HDU	{50, 100, 500, 1000}	$[-30, 30]$	0	(1, 1, ...)
F6	Langkah	HDU	{50, 100, 500, 1000}	$[-100, 100]$	0	(0, 0, ...)
F7	Quartic	HDU	{50, 100, 500, 1000}	$[-1.28, 1.28]$	0	(0, 0, ...)
F8	Schwefel	HDM	{50, 100, 500, 1000}	$[-500, 500]$	$-418.9829 \times \text{Dim}$	(420.9687, 420.9687, ...)
F9	Rastrigin	HDM	{50, 100, 500, 1000}	$[-5.12, 5.12]$	0	(0, 0, ...)
F10	Ackley	HDM	{50, 100, 500, 1000}	$[-32, 32]$	0	(0, 0, ...)
F11	Griewank	HDM	{50, 100, 500, 1000}	$[-600, 600]$	0	(0, 0, ...)
F12	Dihukum	HDM	{50, 100, 500, 1000}	$[-50, 50]$	0	(-1, -1, ...)
F13	Dihukum2	HDM	{50, 100, 500, 1000}	$[-50, 50]$	0	(1, 1, ...)
F14	Lubang rubah	FDM	2	$[-65, 65]$	0.998	(-31.9783, -31.9783)
F15	Kowalik	FDM	4	$[-5, 5]$	0.0003	(0.1928, .1908, 0.1231, 0.1358)
F16	Unta Enam Punuk	FDM	2	$[-5, 5]$	-1.0316	(0.0898, -0.7126), (-0.0898, 0.7126)
F17	Branin	FDM	2	$[-5, 5]$	0.398	(- $\pi$ , 12.275), ( $\pi$ , 2.275), (9.425, 2.275)
F18	Harga EmasStein	FDM	2	$[-2, 2]$	3	(0, -1)
F19	Hartman 3	FDM	3	[0, 1]	-3.86278	(0.1146, .5557, 0.8526)
F20	Hartman 6	FDM	6	[0, 1]	-3.32	(0.202, .150, 0.477, 0.275, 0.312, 0.657)
F21	Shekel 5	FDM	4	[0, 10]	-10.1532	(4, 4, 4, 4)
F22	Shekel 7	FDM	4	[0, 10]	-10.4029	(4, 4, 4, 4)
F23	Shekel 10	FDM	4	[0, 10]	-10.5364	(4, 4, 4, 4)

**Pernyataan kontribusi kepengarangan CRediT**

**Suyanto Suyanto:** Peneliti utama, Konseptualisasi, Metodologi, Kurasi data, Supervisi, Penulisan - draf awal. **Alifya Aisyah Ariyanto:** Investigasi, Penulisan - tinjauan & penyuntingan. **Alifya Fatimah Ariyanto:** Validasi, Penulisan - tinjauan & penyuntingan.

**Deklarasi kepentingan yang bersaing**

Para penulis menyatakan bahwa mereka tidak memiliki kepentingan keuangan atau hubungan pribadi yang dapat mempengaruhi pekerjaan yang dilaporkan dalam makalah ini.

**Ucapan terima kasih**

Sebagai penulis pertama, saya ingin mengucapkan terima kasih kepada kedua orang tua saya, Saekan dan Musripah, yang telah menginspirasi gerakan *mlipir* yang diadopsi dalam Algoritma Mlipir Komodo dan juga putri saya yang masih remaja, Alima Tasnim Ariyanto, yang telah menciptakan kartun Komodo.

**Lampiran. Spesifikasi 23 fungsi benchmark dan tampilan dua dimensinya**

Lihat Tabel 6 dan Gbr. 8.

**Referensi**

- J.H. Holland, Algoritma genetika, Sci. Am. 267 (1) (1992) 66-72, <http://dx.doi.org/10.1038/scientificamerican0792-66>, URL <https://www.scientificamerican.com/article/genetic-algorithms/>.
- Suyanto, Algoritma Genetika Terinformasi untuk Masalah Penjadwalan Mata Kuliah dan Jadwal Kuliah Mahasiswa, vol. 6114 LNAI, (PART 2) 2010, [http://dx.doi.org/10.1007/978-3-642-13232-2\\_28](http://dx.doi.org/10.1007/978-3-642-13232-2_28).
- J. Cheng, Z. Pan, H. Liang, Z. Gao, J. Gao, Algoritma evolusi diferensial dengan operator mutasi berbasis peringkat kebugaran dan keragaman, Swarm Evol. Comput. 61 (Mei 2020) 100816, <http://dx.doi.org/10.1016/j.swevo.2020.100816>, URL <https://doi.org/10.1016/j.swevo.2020.100816>.
- M.A. Bourouis, A. Zadjoui, A. Djedid, Kontribusi dua teknik in-telligence buatan dalam memprediksi indeks kompresi sekunder tanah berbutir halus, Innov. Infrastr. Solut. 5 (3) (2020) <http://dx.doi.org/10.1007/s41062-020-00348-1>.
- X. Yin, Z. Niu, Z. He, Z. Li, D. Lee, Skema optimasi parameter operasi berbasis teknik kecerdasan komputasi terintegrasi untuk sistem manufaktur berorientasi peningkatan kualitas, Comput. Ind. Eng. 140 (2020) <http://dx.doi.org/10.1016/j.cie.2020.106284>.
- V. Mp, B. Anand, Mikroprosesor dan Mikrosistem Teknik op- timisasi kawanan partikel untuk inverter bertingkat pada sistem jaringan mikro pemanen tenaga surya, Mikroproses. Mikrosist. 79 (Agustus) (2020) 103288, <http://dx.doi.org/10.1016/j.micpro.2020.103288>.
- S. Palacios, S. Chiriboga, W. Montalvo, Pengontrol kecepatan untuk PID-2DOF-ACO motor DC pada platform ARM [Controlador de velocidad PID-2DOF-ACO untuk motor DC pada platform ARM], RISTI - Revista Iberica de Sistemas E Tecnologias de Informacao 2020 (E30) (2020) 217-228.
- N. Hansen, S.D. Müller, P. Koumoutsakos, Mengurangi kompleksitas waktu dari strategi evolusi teracak dengan adaptasi matriks kovarian (CMA-ES), Genet. Evol. comput. Ser. 11 (1) (2003) 1-18, <http://dx.doi.org/10.1162/10636560321828970>.
- R. Tanabe, A. Fukunaga, Adaptasi parameter berbasis sejarah keberhasilan untuk Evolusi Diferensial, dalam: Kongres IEEE 2013 tentang Komputasi Evolusioner, (ISSN: 1941-0026) 2013, hal. 71-78, <http://dx.doi.org/10.1109/CEC.2013.6557555>.
- R. Tanabe, A.S. Fukunaga, Meningkatkan performa pencarian SHADE menggunakan pengurangan ukuran populasi linear, in: IEEE Congress on Evolutionary Computation (CEC), 2014.
- N.H. Awad, M.Z. Ali, P.N. Suganthan, Ensemble sinusoidal differential covariance matrix adaptation with euclidean neighborhood for solving CEC2017 benchmark problems, in: 2017 IEEE Congress on Evolutionary Computation (CEC), 2017, pp. 372-379, <http://dx.doi.org/10.1109/CEC.2017.7969336>.
- X.-S. Yang, Algoritma Firefly, fungsi uji stokastik dan desain, Int. J. Bio-Inspired Comput. 2 (2) (2010) 78-84, <http://dx.doi.org/10.1504/IJBIC.2010.032124>, URL <https://www.inderscienceonline.com/doi/pdf/10.1504/IJBIC.2010.032124>.
- V. Kumar, D. Kumar, Tinjauan sistematis pada algoritma firefly: Masa lalu, sekarang, dan masa depan, Arch. Comput. Methods Eng. (2020) <http://dx.doi.org/10.1007/s11831-020-09498-y>, URL <https://www.x-mol.com/paper/1311351373626052608>.
- X.-S. Yang, S. Deb, Pencarian kukuk melalui penerbangan pungutan, 2010, arXiv:1003.1594.
- H. Ma, S. Li, E. Zhang, Z. Lv, J. Hu, X. Wei, Pengemudian otonom kooperatif yang berorientasi pada MEC-Bantuan 5G-V2X: Desain sistem prototipe, uji lapangan, dan alat pengoptimalan berbasis AI, IEEE Access 8 (2020) 54288-54302, <http://dx.doi.org/10.1109/ACCESS.2020.2981463>.
- X.-s. Yang, Algoritma metaheuristik baru yang terinspirasi dari kelelawar, dalam: Nature Inspired Cooperative Strategies for Optimization (NISCO), 2010, hal. 65-74, arXiv: arXiv:1004.4170v1, URL [https://link.springer.com/chapter/10.1007/978-3-642-12538-6\\_6](https://link.springer.com/chapter/10.1007/978-3-642-12538-6_6).
- A.A. Fadhil, R.G.H. Alsarraj, A.M. Altaie, Estimasi biaya perangkat lunak berdasarkan algoritma lumba-lumba, IEEE Access 8 (2020) 75279-75287, <http://dx.doi.org/10.1109/ACCESS.2020.2988867>.
- K.M. Passino, Biomimikri bakteri untuk pencarian makanan untuk optimasi dan kontrol terdistribusi, IEEE Control Syst. 22 (3) (2002) 52-67, <http://dx.doi.org/10.1109/MCS.2002.1004010>, URL <https://ieeexplore.ieee.org/abstract/document/1004010>.
- X. Gan, B.X. B. Algoritma Optimasi Pencarian Makan Bakteri yang Disempurnakan dengan Kawanan Komprehensif, Penerbitan Internasional Springer, 2020, hal. 325- 334, <http://dx.doi.org/10.1007/978-3-030-53956-6>, URL [http://dx.doi.org/10.1007/978-3-030-53956-6\\_29](http://dx.doi.org/10.1007/978-3-030-53956-6_29).



- [20] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: Algoritma pencarian gravitasi, *Inform. Sci.* 179 (13) (2009) 2232-2248, <http://dx.doi.org/10.1016/j.ins.2009.03.004>, URL <http://www.sciencedirect.com/science/article/pii/S0020025509001200>, Bagian Khusus tentang Himpunan Fuzzy Orde Tinggi.
- [21] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Algoritma Salp Swarm: Pengoptimal yang terinspirasi oleh bio untuk masalah desain teknik, *Adv. Eng. Softw.* 114 (2017) 163-191, <http://dx.doi.org/10.1016/j.advengsoft.2017.07.002>, URL <http://www.sciencedirect.com/science/article/pii/S0965997816307736>.
- [22] R. Rao, V. Savsani, D. Vakharia, Optimasi berbasis belajar-mengajar: Metode baru untuk masalah optimasi desain mekanik yang dibatasi, *Comput. Aided Des.* 43 (3) (2011) 303-315, <http://dx.doi.org/10.1016/j.cad.2010.12.015>, URL <http://www.sciencedirect.com/science/article/pii/S0010448510002484>.
- [23] S. Mirjalili, S.M. Mirjalili, A. Lewis, serigala abu-abu, *Adv. Eng. Softw.* 69 (2014) 46-61, <http://dx.doi.org/10.1016/j.advengsoft.2013.12.007>.
- [24] S. Gupta, K. Deep, Pengoptimal serigala abu-abu yang terinspirasi oleh kepemimpinan yang disempurnakan untuk masalah optimasi global, *Eng. Comput.* 36 (4) (2020) 1777-1800, <http://dx.doi.org/10.1007/s00366-019-00795-0>.
- [25] S. Mirjalili, Algoritma capung: teknik optimasi meta-heuristik baru untuk menyelesaikan masalah dengan tujuan tunggal, diskrit, dan multi-tujuan, *Neural Comput. Appl.* 27 (4) (2016) 1053-1073, <http://dx.doi.org/10.1007/s00521-015-1920-1>, URL <https://doi.org/10.1007/s00521-015-1920-1>.
- [26] A.I. Hammouri, M. Mafarja, M.A. Al-Betar, M.A. Awadallah, I. Abu-Doush, Algoritma Capung yang lebih baik untuk pemilihan fitur, *Knowl.* 203 (2020) 106131, <http://dx.doi.org/10.1016/j.knosys.2020.106131>, URL <http://www.sciencedirect.com/science/article/pii/S0950705120303889>.
- [27] S. Mirjalili, Kemajuan dalam perangkat lunak rekayasa pengoptimal semut singa, *Adv. Eng. Softw.* 83 (2015) 80-98, <http://dx.doi.org/10.1016/j.advengsoft.2015.01.010>, URL <http://dx.doi.org/10.1016/j.advengsoft.2015.01.010>.
- [28] A.S. Assiri, A.G. Hussien, M. Amin, Optimasi semut singa: Varian, hibrida, dan aplikasi, *IEEE Access* 8 (2020) 77746-77764, <http://dx.doi.org/10.1109/ACCESS.2020.2990338>, URL <https://ieeexplore.ieee.org/abstract/document/9078091>.
- [29] S. Mirjalili, Algoritma optimasi api ngengat: Paradigma heuristik baru yang terinspirasi dari alam, *Knowl.* 89 (2015) 228-249, <http://dx.doi.org/10.1016/j.knosys.2015.07.006>, URL <https://www.sciencedirect.com/science/article/abs/pii/S0950705115002580>.
- [30] R. Zhang, Z. Qiu, Mengoptimalkan hiper-parameter jaringan saraf dengan kecerdasan kawanan: Kerangka kerja baru untuk penilaian kredit, *PLoS ONE* 15 (6) (2020) <http://dx.doi.org/10.1371/journal.pone.0234254>.
- [31] S.A. Uymaz, G. Tezel, E. Yel, Algoritma ganggang buatan (AAA) untuk optimasi global nonlinier, *Appl. Soft Comput.* 31 (2015) 153-171, <http://dx.doi.org/10.1016/j.asoc.2015.03.003>, URL <https://www.sciencedirect.com/science/article/pii/S1568494615001465>.
- [32] S. Korkmaz, M.S. Kiran, Algoritma ganggang buatan dengan perilaku stigmatisasi untuk optimasi biner, *Appl. Soft Comput.* 64 (2018) 627-640, <http://dx.doi.org/10.1016/j.asoc.2018.01.001>, URL <https://www.sciencedirect.com/science/article/pii/S1568494618300061>.
- [33] W. Zhao, L. Wang, Z. Zhang, Optimasi pencarian atom baru untuk estimasi koefisien dispersi dalam air tanah, *Future Gener. Comput. Syst.* (2018) <http://dx.doi.org/10.1016/j.future.2018.05.037>.
- [34] R.V. Rao, Algoritma Rao: Tiga algoritma sederhana tanpa metafora untuk memecahkan masalah optimasi, *Int. J. Ind. Eng. Comput.* 11 (1) (2020) 107-130, <http://dx.doi.org/10.5267/j.ijec.2019.6.002>.
- [35] S. Suyanto, A.T. Wibowo, S.A. Faraby, S. Saadah, R. Rismala, Algoritma Rao Evolusioner, *J. Comput. Sci.* 53 (Maret) (2021) 101368, <http://dx.doi.org/10.1016/j.jocs.2021.101368>.
- [36] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, keseimbangan: Algoritma optimasi baru, *Sistem Berbasis Pengetahuan*. 191 (2020) <http://dx.doi.org/10.1016/j.knosys.2019.105190>, URL <https://www.sciencedirect.com/science/article/abs/pii/S0950705119305295?via%3Dihub>.
- [37] A. Faramarzi, M. Heidarinejad, S. Mirjalili, A.H. Gandomi, Algoritma Pemangsa Laut: Metaheuristik yang terinspirasi dari alam, *Expert Syst. Appl.* 152 (2020) <http://dx.doi.org/10.1016/j.eswa.2020.113377>.
- [38] M.A. Elaziz, A.A. Ewees, D. Yousri, H.S.N. Alwerfali, Q.A. Awad, S. Lu, M.A.A. Al-Qaness, Algoritma pemangsa laut yang lebih baik dengan entropi kabur untuk penguraian multi level: Contoh dunia nyata dari segmentasi citra CT COVID-19, *IEEE Access* 8 (2020) 125306-125330, <http://dx.doi.org/10.1109/ACCESS.2020.3007928>.
- [39] S. Li, H. Chen, M. Wang, A.A. Heidari, S. Mirjalili, Algoritma cetakan lendir: Metode baru untuk optimasi stokastik, *Future Gener. Comput. Syst.* 111 (2020) 300-323, <http://dx.doi.org/10.1016/j.future.2020.03.055>, URL <https://www.sciencedirect.com/science/article/pii/S0167739X19320941>.
- [40] R.-E. Precup, R.-C. David, R.-C. Roman, E.M. Petriu, A.-I. Szedlak-Stinean, Penyetelan berbasis algoritma cetakan lendir dari pengendali fuzzy yang hemat biaya untuk sistem servo, *Int. J. Comput. Intell. Syst.* 14 (1) (2021) 1042-1052, <http://dx.doi.org/10.2991/ijcis.d.210309.001>.
- [41] J. Brest, M.S. Maucec, Tantangan 100 digit: Algoritma jDE100, in: 2019 IEEE Congress on Evolutionary Computation (CEC), 2019, <http://dx.doi.org/10.1109/CEC.2019.8789904>.
- [42] K.M. Sallam, S.M. Elsayed, R.K. Chakraborty, M.J. Ryan, Algoritma evolusi diferensial multi-operator yang lebih baik untuk menyelesaikan masalah yang hemat biaya untuk sistem servo, *Kongres IEEE 2020 tentang Komputasi Evolusioner (CEC)*, 2020, pp. 1-8, <http://dx.doi.org/10.1109/CEC48606.2020.9185577>.
- [43] D. Wolpert, W. Macready, Tidak ada teorema makan siang gratis untuk optimasi, *IEEE Trans. Evol. Comput.* 1 (1) (1997) 67-82, <http://dx.doi.org/10.1109/4235.585893>, URL <https://www.scopus.com/inward/record.uri?eid=2-s2.0-0031118203&doi=10.1109%2F4235.585893&partnerID=40&md5=fac8c56be911367d556066800e863066>, dikutip oleh 5993.
- [44] T.E. dari , Komodo, dalam: *Britannica*, 2020, URL <https://www.britannica.com/animal/Komodo-dragon>.
- [45] J. Sartore, Komodo, dalam: *National Geographic*, 2021, URL <https://www.nationalgeographic.com/animals/reptiles/k/komodo-dragon/>.
- [46] A.L. Lind, Y.Y. Lai, Y. Mostovoy, A.K. Holloway, A. Iannucci, A.C. Mak, M. Fondi, V. Orlandini, W.L. Eckalbar, M. Milan, dkk., Genom Komodo mengungkapkan adaptasi dalam kardiovaskular dan kemiosensorisistem biawak, *Nat. Ecol. Evol.* 3 (8) (2019) 1241-1252.
- [47] C. C. Ciofi, , *Sci. Am.* 280 (3) (1999) 84-91.
- [48] P.C. Watts, K.R. Buley, S. Sanderson, W. Boardman, C. Ciofi, R. Gibson, Partenogenesis pada , *Nature* 444 (7122) (2006) 1021-1022.
- [49] K. Opara, J. Arabas, Perbandingan strategi mutasi dalam evolusi diferensial - perspektif probabilistik, *Swarm Evol. Comput.* 39 (2018) 53-69, <http://dx.doi.org/10.1016/j.swevo.2017.12.007>, URL <https://www.sciencedirect.com/science/article/pii/S2210650217303310>.
- [50] C. Stephens, H. Waelbroeck, Evolusi skemata dan blok bangunan, *Genet. Evol. comput. Ser.* 7 (2) (1999) 109-124, <http://dx.doi.org/10.1162/evco.1999.7.2.109>.