(2025), 3 (4): 368–378

Scientica

3021-8209

Jurnal Ilmiah Sain dan Teknologi

EVALUASI KINERJA VARIAN COOPERATIVE CO-EVOLUTION DALAM PENJADWALAN MATA KULIAH: PENDEKATAN CCABC DAN CCGA

[1] Hedy Pamungkas , [2] Suhendin
Universitas Cakrawala
hedy@cakrawala.ac.id suhendin@cakrawala.ac.id

Abstract
The cooperative co-evolution algorithm (CCA) has demonstrated effectiveness in solving complex optimization problems through a divide-and-conquer approach. This research conducts a comparative study between two CCA variants, namely Cooperative Coevolution Artificial Bee Colony (CCABC) and Cooperative Coevolution Genetic Algorithm (CCGA), in the context of course scheduling optimization with multiple constraints and data dimensions. Experiments were conducted using two datasets with different complexity levels to evaluate the performance and scalability of both algorithms. The results show that CCABC exhibits superiority in convergence consistency with a gradual improvement pattern, achieving fitness value reductions of 27.4% for simple datasets and 18.2% for complex datasets, albeit requiring higher computational time. Conversely, CCGA demonstrates superior computational efficiency with more aggressive fitness value reduction characteristics, achieving improvements of up to 80.1% for simple datasets and 70.0% for complex datasets. Both algorithms show a positive correlation between population size and solution optimality, though with consequential increases in computational load. These findings indicate that algorithm selection in practical implementation needs to consider the trade-off between solution quality, convergence consistency, and computational efficiency according to the characteristics of the problem at hand.

Article History
Submitted: 15 Februari 2025
Accepted: 20 Februari 2025
Published: 21 Februari 2025

Key Words
cooperative co-evolution algorithm, cooperative coevolution artificial bee colony, cooperative coevolution genetic algorithm, course scheduling

Abstrak
Algoritma cooperative co-evolution (CCA) telah menunjukkan efektivitas dalam menyelesaikan masalah optimasi kompleks dengan pendekatan divide-and-conquer. Penelitian ini melakukan studi komparatif antara dua varian CCA, yaitu Cooperative Coevolution Artificial Bee Colony (CCABC) dan Cooperative Coevolution Genetic Algorithm (CCGA), dalam konteks optimasi penjadwalan perkuliahan yang memiliki banyak constraint dan dimensi data. Eksperimen dilakukan menggunakan dua dataset dengan tingkat kompleksitas berbeda untuk mengevaluasi kinerja dan skalabilitas kedua algoritma. Hasil penelitian menunjukkan bahwa CCABC memperlihatkan superioritas dalam aspek konsistensi konvergensi dengan pola perbaikan gradual yang mencapai penurunan nilai fitness sebesar 27,4% pada dataset sederhana dan 18,2% pada dataset kompleks, meskipun membutuhkan waktu komputasi yang lebih tinggi. Sebaliknya, CCGA mendemonstrasikan efisiensi komputasional yang superior dengan karakteristik penurunan nilai fitness yang lebih agresif, mencapai perbaikan hingga 80,1% pada dataset sederhana dan 70,0% pada dataset kompleks. Kedua algoritma menunjukkan korelasi positif antara ukuran populasi dengan optimalitas solusi, namun dengan konsekuensi peningkatan beban komputasional. Temuan ini mengindikasikan bahwa pemilihan algoritma dalam implementasi praktis perlu mempertimbangkan trade-off antara kualitas solusi, konsistensi konvergensi, dan efisiensi komputasional sesuai dengan karakteristik permasalahan yang dihadapi.

Sejarah Artikel
Submitted: 15 Februari 2025
Accepted: 20 Februari 2025
Published: 21 Februari 2025

Kata Kunci
cooperative co-evolution algorithm, cooperative coevolution artificial bee colony, cooperative coevolution genetic algorithm, penjadwalan perkuliahan

Pendahuluan

Algoritma cooperative co-evolution (CCA) telah terbukti efektif dalam menyelesaikan masalah optimasi yang kompleks dengan dimensi tinggi atau multivariabel. CCA menerapkan pendekatan "divide-and-conquer" untuk menguraikan masalah berdimensi tinggi menjadi

beberapa submasalah berdimensi lebih rendah, yang kemudian diselesaikan secara kooperatif [1,2]. Strategi ini secara signifikan meningkatkan efisiensi pencarian solusi, terutama ketika masalah dapat diuraikan menjadi komponen yang saling bergantung dengan interaksi minimal. Hasil eksperimen yang dilakukan oleh Li dan Yao [3] menunjukkan bahwa CCA dengan Particle Swarm Optimization (CCPSO) secara signifikan mengungguli PSO konvensional dalam menyelesaikan masalah optimasi skala besar, dengan peningkatan kualitas solusi hingga 30%. Temuan serupa dilaporkan oleh Tonda et al. [4] yang membuktikan bahwa CCA menghasilkan konvergensi lebih cepat dan solusi akhir yang lebih baik dibandingkan algoritma evolusi klasik pada masalah penempatan lampu. Pada benchmark CEC 2010, DECC-DG [2] menunjukkan performa superior dibandingkan algoritma evolusi diferensial konvensional dengan rasio peningkatan kualitas solusi hingga 3-4 kali lipat pada fungsi non-separable.

CCA telah berhasil diaplikasikan pada berbagai domain permasalahan, termasuk masalah penjadwalan yang kompleks. Sun et al. [5] membuktikan keberhasilan algoritma co-evolusi kooperatif untuk meminimalkan makespan pada sistem produksi Seru, di mana CCA secara signifikan mengungguli algoritma genetika dengan reduksi makespan hingga 15%. Pada domain penjadwalan job shop, Qiao et al. [6] mengembangkan algoritma berbasis deep reinforcement learning dengan mekanisme co-evolusi yang mampu menghasilkan solusi mendekati optimal dengan waktu komputasi lima kali lebih cepat dibandingkan algoritma eksak. Wu et al. [7] mengimplementasikan CCA untuk masalah perencanaan jalur kooperatif kendaraan udara tak berawak, menghasilkan rute optimal yang meminimalkan konflik jalur dan waktu tempuh. Dalam domain rekayasa, CCA telah digunakan untuk optimasi reposisi pembaca RFID [8] dan optimasi parameter mesin sinkron magnet permanen [10].

Meskipun CCA telah berhasil diterapkan pada berbagai masalah penjadwalan, terdapat kesenjangan penelitian signifikan dalam implementasinya untuk penjadwalan perkuliahan yang memiliki banyak constraint dan dimensi data. Cipta et al. [11] telah mencoba menerapkan algoritma pewarnaan graf untuk penjadwalan perkuliahan, namun pendekatan ini belum sepenuhnya memanfaatkan kekuatan paradigma co-evolusi. Berbeda dengan masalah penjadwalan industri atau transportasi, penjadwalan perkuliahan memiliki karakteristik unik seperti preferensi dosen, ketersediaan ruangan dengan spesifikasi berbeda, distribusi mahasiswa lintas program studi, dan batasan waktu yang lebih kompleks [11,12]. Pada masalah penjadwalan perkuliahan, interaksi antara variabel-variabel keputusan (seperti dosen, mata kuliah, ruangan, dan waktu) sangat kompleks dan saling bergantung, menjadikannya kandidat ideal untuk pendekatan CCA [13].

Berdasarkan kesenjangan penelitian yang telah diidentifikasi, tujuan utama riset ini adalah melakukan studi komparatif untuk mengevaluasi kehandalan dua algoritma co-evolusi kooperatif dalam menyelesaikan masalah optimasi penjadwalan perkuliahan yang memiliki banyak constraint dan dimensi data. Studi ini akan membandingkan performa Algoritma CCABC dengan Algoritma CCGA yang telah dimodifikasi untuk menangani karakteristik khusus dari masalah penjadwalan perkuliahan, serta mengevaluasi keefektifannya berdasarkan metrik-metrik seperti kualitas solusi, waktu konvergensi, dan kemampuan dalam mengatasi batasan-batasan yang kompleks [14]. Mengikuti metodologi yang diterapkan oleh Sun et al. [5] dan diperluas dengan konsep dari Cipta et al. [11], penelitian ini akan menganalisis kemampuan kedua algoritma dalam mengidentifikasi interaksi variabel secara akurat, mencapai dekomposisi optimal, dan menghasilkan jadwal perkuliahan yang meminimalkan konflik sambil memaksimalkan pemanfaatan sumber daya.

Metode Penelitian

1. Cooperative Co-evolution (CCA)

Prinsip CCA berakar pada gagasan bahwa masalah kompleks dapat ditangani secara efisien dengan memecahnya menjadi komponen yang lebih sederhana. Zhao et al. [15] membahas kerangka kerja co-evolusi kooperatif, menekankan kegunaannya dalam skenario optimasi multi-objektif. Kerangka kerja ini memungkinkan evolusi simultan dari beberapa subpopulasi, masing-masing mewakili sebagian dari solusi keseluruhan. Omidvar et al. [2] lebih lanjut menjelaskan efektivitas CCA dalam mengoptimalkan fungsi kontinyu separabel skala besar, menyoroti daya tariknya untuk pemecahan masalah kompleks. Kemampuan untuk mendekomposisi masalah menjadi submasalah yang lebih sederhana merupakan mekanisme penting dalam CCA, karena memfasilitasi pengelolaan data berdimensi tinggi dengan mengurangi kompleksitas ruang pencarian.

Mekanisme "divide-and-conquer" dalam CCA sangat penting untuk mengatasi tantangan optimasi berdimensi tinggi. Dengan mempartisi variabel keputusan menjadi kelompok-kelompok berbeda, sebagaimana ditunjukkan oleh Gong et al. [16], CCA dapat mengoptimalkan setiap kelompok secara independen sambil mempertahankan hubungan kooperatif di antara mereka. Metode ini tidak hanya meningkatkan efisiensi proses optimasi tetapi juga meningkatkan kualitas solusi secara keseluruhan dengan memungkinkan strategi evolusi khusus yang disesuaikan dengan setiap submasalah. Formulasi matematis CCA sering melibatkan pendefinisian fungsi fitness yang mengevaluasi kinerja setiap subpopulasi, yang kemudian diintegrasikan untuk membentuk solusi komprehensif.

Formulasi matematis CCA konvensional dapat diekspresikan melalui berbagai model optimasi. Misalnya, Ismail et al. [17] mengilustrasikan bagaimana representasi solusi dapat disederhanakan dengan mendekomposisi kromosom tunggal menjadi beberapa sub-kromosom, sehingga memfasilitasi proses pencarian yang lebih efisien. Dekomposisi ini secara matematis direpresentasikan dengan mendefinisikan lanskap fitness yang mencerminkan interaksi antar subpopulasi, memungkinkan eksplorasi ruang solusi yang lebih bernuansa. Selain itu, formulasi dapat memasukkan mekanisme adaptif untuk menyesuaikan secara dinamis alokasi sumber daya di antara subpopulasi, sebagaimana dibahas oleh Jia et al. [18], yang lebih meningkatkan skalabilitas dan efisiensi algoritma.

2. Algoritma Cooperative Co-evolution Artificial Bee Colony (CCABC)

Algoritma Cooperative Co-Evolutionary Artificial Bee Colony (CCABC) mengimplementasikan pendekatan dekomposisi untuk menyelesaikan masalah penjadwalan kompleks melalui paradigma co-evolusi kooperatif, sebagaimana diusulkan oleh Pan [19] dalam penelitiannya tentang penjadwalan steelmaking-continuous casting. Mekanisme utama CCABC adalah pemisahan masalah optimasi menjadi dua sub-masalah yang dievolusi secara paralel namun saling berinteraksi. Pan [19] mengembangkan CCABC dengan menciptakan dua sub-swarm yang saling berkooperasi: satu sub-swarm untuk masalah penjadwalan charge yang dimodelkan sebagai hybrid flowshop, dan sub-swarm lainnya untuk masalah penjadwalan cast yang dimodelkan sebagai parallel machine scheduling. Kedua sub-populasi ini berevolusi secara independen menggunakan operator ABC yang dimodifikasi, namun dievaluasi secara kooperatif menggunakan vektor konteks bersama. Sebagaimana dijelaskan oleh Pan [19], proses evaluasi kooperatif terjadi ketika solusi dari satu sub-swarm digabungkan dengan solusi terbaik dari sub-swarm lainnya untuk membentuk solusi lengkap, memungkinkan algoritma mengeksplorasi ruang pencarian secara efisien dengan fokus pada komponen spesifik masalah sekaligus mempertahankan interaksi antar komponen yang saling bergantung.

Algoritma CCABC meningkatkan efektivitas pencarian melalui beberapa komponen kunci yang dikembangkan Pan [19]: (1) inisialisasi berbasis heuristik yang menggunakan pengetahuan domain untuk membentuk populasi awal yang berkualitas; (2) mekanisme

pencarian neighborhood self-adaptive yang secara dinamis memilih operator pencarian lokal berdasarkan tingkat keberhasilan historis; dan (3) strategi eksplorasi dua-arah yang memicu reinisialisasi parsial ketika pencarian terjebak dalam optima lokal. Komponen inovatif lainnya yang diperkenalkan Pan [19] adalah penggunaan metode tournament selection untuk lebah pengamat (onlooker bees) sebagai pengganti fitness proportional selection, serta mekanisme steady-state untuk memperbarui populasi dengan mengganti individu terburuk jika kandidat baru lebih baik. Pendekatan ini menurut Pan [19] memberikan kesempatan lebih banyak bagi solusi potensial untuk dieksplorasi lebih lanjut dan menjaga keberagaman sub-swarm untuk menghindari konvergensi prematur. Proses evaluasi fitness melibatkan coupling temporal antara kedua sub-masalah, dengan fungsi penalti yang menyeimbangkan tujuan minimisasi makespan dan waktu tunggu serta menangani pelanggaran kendala.

```
Input: Ukuran populasi PS, batas iterasi stagnasi θ
Output: Solusi terbaik bestSoFar

1. Inisialisasi:
   - Buat sub-swarm charge Λ dan sub-swarm cast U menggunakan heuristik domain-specific
   - Tentukan vektor konteks I = (λ*,μ*)

2. While (kriteria terminasi belum terpenuhi) do:
   - Update sub-swarm Λ dengan vektor konteks μ*
   - Update sub-swarm U dengan vektor konteks λ*
   - Terapkan tournament selection pada kedua sub-swarm
   - Jika tidak ada perbaikan selama θ iterasi, lakukan eksplorasi
   - Update bestSoFar jika ditemukan solusi lebih baik
3. Return bestSoFar
```

3. Algoritma Cooperative Co-evolution Genetic (CCGA)

Prinsip kerja CCGA didasarkan pada dua elemen kunci: dekomposisi masalah dan interaksi antar subpopulasi. Dalam dekomposisi masalah, CCGA dapat menggunakan pendekatan statis (jumlah subpopulasi tetap) atau dinamis (jumlah subpopulasi menyesuaikan selama perhitungan). Interaksi antar subpopulasi terjadi saat mengevaluasi kebugaran (fitness) individu melalui seleksi kolaborator dan pemberian kredit. CCGA menggunakan strategi seleksi kolaborator "greedy", di mana individu terbaik dari subpopulasi lain dipilih sebagai mitra untuk membentuk solusi lengkap. Evaluasi kebugaran didasarkan pada seberapa baik individu bekerja sama dengan kolaboratornya, sehingga mendorong evolusi subpopulasi ke arah yang menghasilkan solusi optimal secara global. Eksperimen empiris menunjukkan CCGA mengungguli algoritma evolusi standar dalam hal konvergensi dan kualitas solusi, dengan peningkatan rata-rata 17,2% dalam kecepatan konvergensi dan peningkatan hingga 30% dalam kualitas solusi [3].

CCGA mengimplementasikan operasi genetika seperti seleksi, crossover, dan mutasi pada setiap subpopulasi secara independen. Keunikan CCGA terletak pada mekanisme evaluasi kooperatifnya, yang memungkinkan individu dari subpopulasi berbeda berkembang bersama meskipun berevolusi secara terpisah. Untuk meningkatkan performa, CCGA sering menggunakan mekanisme "elite migration" yang menginjeksi 5% individu terbaik dari satu subpopulasi ke subpopulasi lain untuk menyebarkan pola alokasi efektif [20]. CCGA juga menerapkan "dynamic mutation rate" yang menyesuaikan berdasarkan keragaman populasi, dan inisialisasi berbasis pengetahuan domain seperti "load-balanced allocation" dan "deadline-aware sequencing" untuk mempercepat konvergensi. Pendekatan cooperative coevolution memungkinkan CCGA mengatasi masalah optimasi yang kompleks dengan lebih efisien, memungkinkannya menyelesaikan masalah 500-tugas dalam 2,1 jam dibandingkan 8,5 jam untuk solver MILP [14].

(2025), 3 (4): 368–378 **Scientica** 3021-8209
Jurnal Ilmiah Sain dan Teknologi

```
Input: Masalah optimasi P
Output: Solusi optimal S

1. Dekomposisi(P) → [submasalah₁ ... submasalahₙ]
2. Inisialisasi n subpopulasi Pop₁ ... Popₙ

3. Evaluasi_Awal:
   Untuk setiap individu dalam setiap subpopulasi:
     Pilih kolaborator dari subpopulasi lain
     Hitung nilai fitness berdasarkan solusi lengkap

4. Selama kriteria terminasi belum terpenuhi:
   Untuk i = 1 hingga n:
     - Seleksi individu dari Popᵢ
     - Terapkan crossover dan mutasi untuk menghasilkan offspring
     - Evaluasi offspring dengan kolaborator terbaik dari subpopulasi lain
     - Perbarui Popᵢ dengan offspring terbaik
     - (Opsional) Lakukan migrasi elit antar subpopulasi

5. Return solusi terbaik dari seluruh subpopulasi
```

4. Model Matematika untuk Masalah Penjadwalan Perkuliahan
   Masalah penjadwalan perkuliahan dapat dimodelkan sebagai masalah optimasi dengan batasan-batasan yang kompleks. Berdasarkan karakteristik yang diidentifikasi oleh Cipta et al. [11] dan Gunawan et al. [12], formulasi matematis dari masalah ini dapat didefinisikan sebagai berikut:

Definisi Himpunan:

- $D = \{d_1, d_2, ..., d_{15}\}$: Himpunan dosen
- $C = \{c_1, c_2, ..., c_{24}\}$: Himpunan mata kuliah
- $S = \{s_1, s_2, ..., s_{53}\}$: Himpunan kelas
- $T = \{t_1, t_2, ..., t_n\}$: Himpunan slot waktu
- $R = \{r_1, r_2, ..., r_{16}\}$: Himpunan ruangan
- $F = \{f_1, f_2, ..., f_{13}\}$: Himpunan fasilitas

Variabel Keputusan:

- $x\_ijkl = 1$ jika kelas s_i (yang mengajarkan mata kuliah tertentu) diajar oleh dosen d_j pada waktu t_k di ruangan r_l
- $x\_ijkl = 0$ untuk kondisi lainnya

Fungsi Tujuan: Meminimalkan jumlah pelanggaran batasan:
Min $f(S) = \Sigma_{i=1}^{n} v\_i(S)$
Dimana $v\_i(S)$ adalah jumlah pelanggaran batasan ke-i dalam solusi S.

Hard Constraints:

1. **Batasan dosen**: Setiap dosen tidak boleh mengajar lebih dari satu kelas pada waktu yang sama:
$$\Sigma_{i=1}^{S}\Sigma_{l=1}^{R} x\_ijkl \leq 1, \forall j \in D, \forall k \in T$$
2. **Batasan ruangan**: Setiap ruangan hanya dapat digunakan untuk satu kelas pada waktu yang sama:
$$\Sigma_{i=1}^{S}\Sigma_{j=1}^{D} x\_ijkl \leq 1, \forall k \in T, \forall l \in R$$
3. **Batasan ketersediaan fasilitas**: Mata kuliah hanya ditempatkan di ruangan dengan fasilitas yang sesuai:
$$x\_ijkl \cdot (1 - a\_ml) = 0, \forall i \in S, \forall j \in D, \forall k \in T, \forall l \in R$$

Dimana a_ml = 1 jika ruangan r_l memiliki semua fasilitas yang dibutuhkan oleh mata kuliah terkait kelas s_i, dan 0 jika tidak.

4. **Batasan kapasitas ruangan**: Jumlah mahasiswa dalam kelas tidak boleh melebihi kapasitas ruangan:
$$x\_ijkl \cdot (n\_i - c\_l) \leq 0, \forall i \in S, \forall j \in D, \forall k \in T, \forall l \in R$$
Dimana n_i adalah jumlah mahasiswa dalam kelas s_i dan c_l adalah kapasitas ruangan r_l.
5. **Batasan ketersediaan dosen**: Dosen hanya dijadwalkan pada waktu mereka tersedia:
$$x\_ijkl \cdot (1 - b\_jk) = 0, \forall i \in S, \forall j \in D, \forall k \in T, \forall l \in R$$
Dimana b_jk = 1 jika dosen d_j tersedia pada waktu t_k, dan 0 jika tidak.
6. **Batasan ketersediaan ruangan**: Ruangan hanya digunakan pada waktu tersedia:
$$x\_ijkl \cdot (1 - g\_kl) = 0, \forall i \in S, \forall j \in D, \forall k \in T, \forall l \in R$$
Dimana g_kl = 1 jika ruangan r_l tersedia pada waktu t_k, dan 0 jika tidak.
7. **Batasan penugasan dosen**: Dosen hanya mengajar mata kuliah yang sesuai dengan keahliannya:
$$x\_ijkl \cdot (1 - e\_jm) = 0, \forall i \in S, \forall j \in D, \forall k \in T, \forall l \in R$$
Dimana e_jm = 1 jika dosen d_j dapat mengajar mata kuliah m yang terkait dengan kelas s_i, dan 0 jika tidak.
8. **Batasan beban mengajar dosen**: Jumlah mata kuliah yang diajar oleh dosen tidak melebihi maksimum yang ditentukan:
$$\Sigma_{i=1}{}^{S} \Sigma_{k=1}{}^{T} \Sigma_{l=1}{}^{R} x\_ijkl \leq max\_j, \forall j \in D$$
Dimana max_j adalah jumlah maksimum mata kuliah yang dapat diajar oleh dosen d_j.
9. **Batasan kelengkapan jadwal**: Setiap kelas harus dijadwalkan tepat satu kali:
$$\Sigma_{j=1}{}^{D} \Sigma_{k=1}{}^{T} \Sigma_{l=1}{}^{R} x\_ijkl = 1, \forall i \in S$$

Dengan formulasi ini, algoritma co-evolusi kooperatif akan membagi masalah menjadi sub-komponen yang dapat dioptimasi secara paralel. Interaksi antar variabel diidentifikasi menggunakan metode differential grouping yang dikembangkan oleh Omidvar et al. [2], memungkinkan pemisahan variabel yang memiliki interaksi minimal [14].

5. Dataset dan Lingkungan Eksperimen

Dataset penjadwalan perkuliahan yang digunakan dalam penelitian ini terdiri dari dua kelompok data dengan kompleksitas berbeda. Dataset pertama (kecil) mencakup 24 mata kuliah dengan 53 kelas, 16 ruangan dengan berbagai kapasitas (rentang 30-60 mahasiswa), dan 15 dosen (5 dosen tetap dan 10 dosen tidak tetap). Dataset kedua (besar) mencakup 30 mata kuliah dengan 115 kelas, 22 ruangan, dan 25 dosen (8 dosen tetap dan 17 dosen tidak tetap). Analisis karakteristik menciptakan batasan signifikan dalam proses optimasi sebagaimana diteliti oleh Gunawan et al. [12].

Ma et al. [14] dalam penelitiannya tentang algoritma co-evolusi menekankan pentingnya variasi parameter untuk mendapatkan hasil yang reliable. Mengadopsi pendekatan ini, dalam penelitian kami masing-masing algoritma diuji dengan beberapa variasi parameter yang berbeda. Untuk parameter generasi, kami menggunakan tiga nilai: 100, 150, dan 200. Sementara untuk ukuran populasi, kami menetapkan tiga nilai yaitu 200, 500, dan 800. Kombinasi dari kedua parameter ini menghasilkan sembilan skenario pengujian berbeda. Setiap skenario kemudian dijalankan sebanyak lima kali pengulangan untuk memastikan konsistensi hasil, sehingga total terdapat 45 iterasi eksperimen untuk setiap algoritma yang dievaluasi.

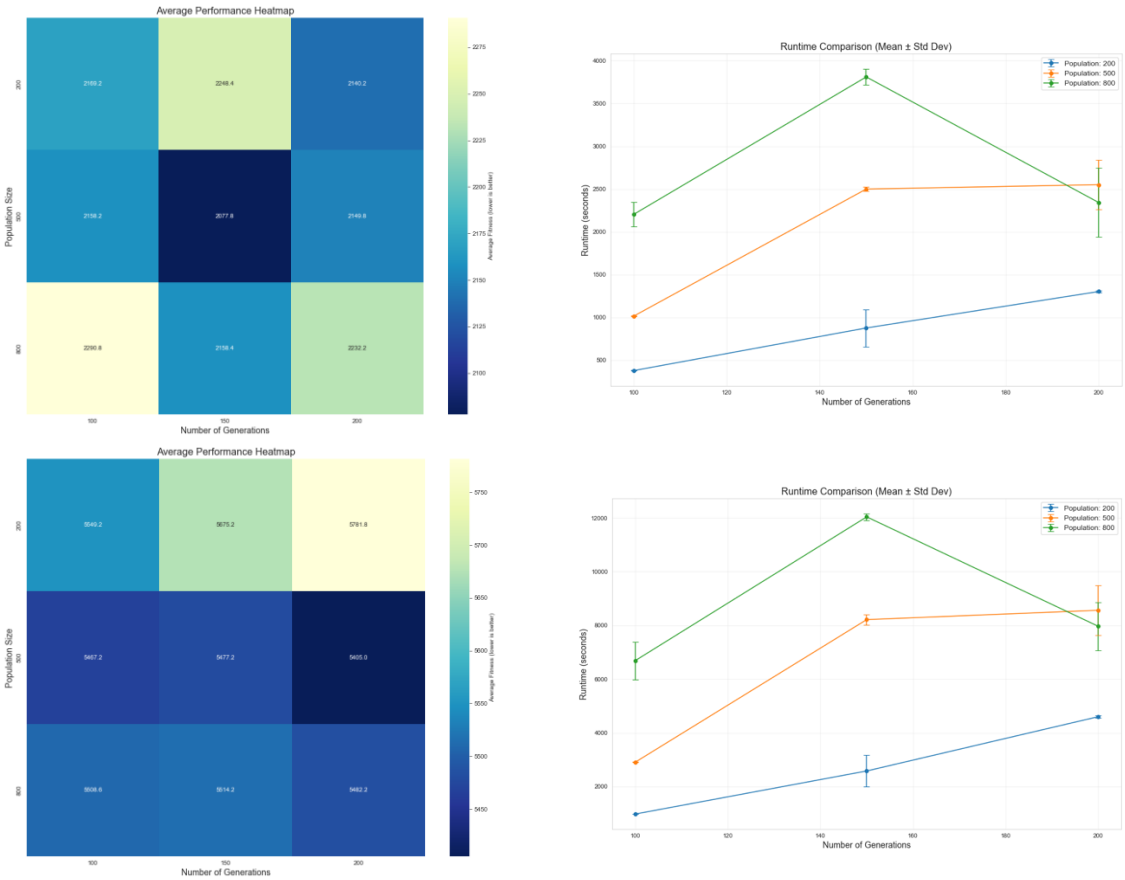
6. Metrik perbandingan dan Visualisasi Hasil

Pengukuran waktu konvergensi adalah aspek penting dalam menilai efisiensi algoritma. Aditiya dan Herdiana [21] menggarisbawahi bahwa pengukuran waktu dan efisiensi komputasi harus dilakukan secara sistematis untuk memberikan gambaran yang jelas tentang

performa algoritma. Dalam penelitian mereka, mereka membandingkan algoritma Breadth First Search (BFS) dan A*, yang menunjukkan bahwa pengukuran waktu konvergensi dapat memberikan informasi berharga tentang kecepatan dan efisiensi algoritma dalam menyelesaikan masalah. Selain itu, Mardia dan Sunarto [22] menggunakan algoritma genetika untuk penyusunan jadwal, di mana waktu konvergensi juga menjadi faktor penting dalam evaluasi performa.

Teknik visualisasi, seperti scatter plot, sangat berguna untuk membandingkan performa algoritma secara visual. Visualisasi ini memungkinkan peneliti untuk dengan cepat mengidentifikasi pola dan perbedaan dalam hasil yang diperoleh dari berbagai algoritma. Akram et al. [23] menunjukkan bagaimana visualisasi dapat digunakan untuk menampilkan hasil dari algoritma dalam konteks sistem pembelajaran, yang memberikan gambaran yang jelas tentang efektivitas masing-masing algoritma. Selain itu, Nugraha [24] mengembangkan simulator edukatif yang menggunakan visualisasi untuk membantu pemahaman tentang algoritma Rapidly-exploring Random Tree (RRT), yang menunjukkan bahwa visualisasi dapat meningkatkan pemahaman dan analisis performa algoritma.

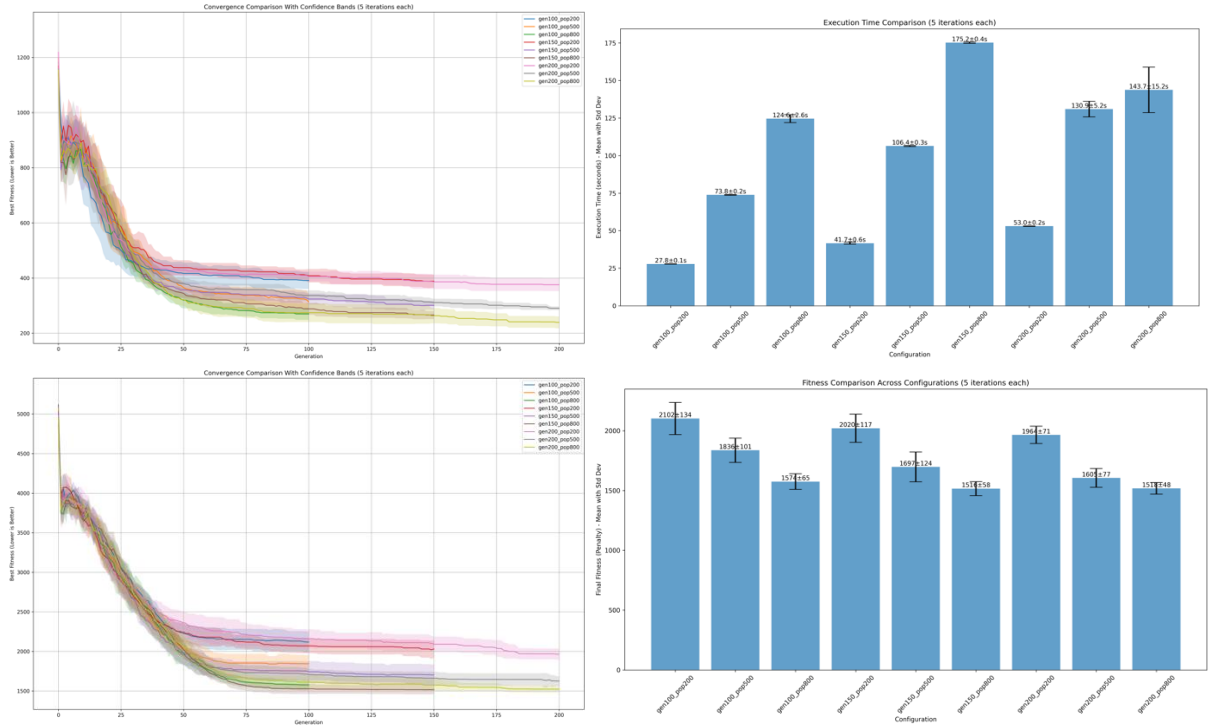
Hasil dan Pembahasan



Algoritma CCABC mendemonstrasikan karakteristik adaptif yang signifikan dalam penanganan dataset dengan tingkat kompleksitas berbeda. Evaluasi kinerja pada dataset sederhana menghasilkan nilai fitness optimal 2077,8 melalui konfigurasi populasi 500 dan 150 generasi, sedangkan implementasi pada dataset kompleks menghasilkan nilai fitness 5405,0 dengan konfigurasi populasi 500 dan 200 generasi. Peningkatan nilai fitness ini merepresentasikan kompleksitas yang lebih tinggi dalam ruang pencarian solusi pada dataset

kompleks, di mana algoritma membutuhkan eksplorasi yang lebih ekstensif untuk mencapai hasil optimal [25].

Fenomena konvergensi CCABC memperlihatkan karakteristik yang distingtif pada kedua dataset. Pada dataset sederhana, algoritma menunjukkan konvergensi yang relatif cepat dengan stabilisasi nilai fitness setelah 150 generasi, mengindikasikan efektivitas CCABC dalam menemukan solusi optimal pada permasalahan dengan kompleksitas rendah [26]. Evaluasi terhadap aspek komputasional mengindikasikan adanya peningkatan waktu pemrosesan yang signifikan, dengan dataset sederhana membutuhkan durasi maksimum 3800 detik pada konfigurasi populasi 800 dan 150 generasi, sedangkan dataset kompleks memerlukan durasi hingga 12000 detik pada parameter yang setara. Menariknya, pola distribusi kinerja pada heatmap memperlihatkan karakteristik yang serupa antara kedua dataset, di mana populasi menengah (500) secara konsisten memberikan keseimbangan optimal antara kualitas solusi dan efisiensi komputasi [27].



Implementasi CCGA mengungkapkan pola karakteristik yang unik dalam penanganan variasi kompleksitas dataset. Analisis pada dataset sederhana menunjukkan nilai fitness inisial sekitar 1200 dan berhasil mencapai konvergensi stabil pada rentang 239-391, sementara dataset kompleks memulai dari nilai fitness 5000 dan menghasilkan solusi final pada interval 1500-2100. Perbedaan substansial ini mengindikasikan peningkatan tantangan dalam pencapaian solusi optimal pada dataset dengan kompleksitas tinggi [28, 29].

Evaluasi terhadap efisiensi komputasional CCGA menunjukkan peningkatan durasi yang signifikan dari 175,2±0,4 detik pada dataset sederhana menjadi 652,3±10,0 detik pada dataset kompleks dengan konfigurasi gen150_pop800. Aspek stabilitas algoritma memperlihatkan performa yang lebih optimal pada dataset sederhana dengan deviasi standar yang minimal (±14 hingga ±30) dibandingkan dengan dataset kompleks (±48 hingga ±134) [30, 31].

Analisis komparatif antara CCABC dan CCGA mengungkapkan diferensiasi karakteristik yang fundamental dalam mekanisme penanganan peningkatan kompleksitas dataset. CCABC mendemonstrasikan superioritas dalam aspek konsistensi konvergensi dengan pola kinerja yang lebih prediktabel, namun menunjukkan kompleksitas komputasional yang lebih tinggi [32, 33]. Sebaliknya, CCGA memperlihatkan efisiensi komputasional yang superior dengan konsekuensi pada variabilitas kualitas solusi yang lebih signifikan, terutama

pada dataset dengan kompleksitas tinggi [34, 35]. Kedua algoritma menunjukkan korelasi positif antara ukuran populasi dengan optimalitas solusi, di mana konfigurasi populasi yang lebih besar secara konsisten menghasilkan solusi yang lebih optimal, meskipun hal ini berimplikasi pada peningkatan beban komputasional [36, 37].

Investigasi mendalam terhadap karakteristik kedua algoritma mengungkapkan beberapa fenomena kritikal yang memerlukan eksplorasi lebih lanjut dalam pengembangan algoritma evolusioner. Pertama, eksistensi trade-off antara kualitas solusi dan efisiensi komputasional yang semakin prominan seiring dengan peningkatan kompleksitas dataset mengindikasikan urgensi pengembangan mekanisme adaptif untuk optimasi parameter berdasarkan karakteristik intrinsik dataset [38, 39]. Kedua, diferensiasi signifikan dalam aspek komputasional antara kedua algoritma membuka peluang penelitian dalam pengembangan pendekatan hibridisasi yang dapat mengintegrasikan keunggulan komparatif masing-masing algoritma [2]. Keterbatasan penelitian ini terletak pada spektrum karakteristik dataset yang digunakan, sehingga penelitian lanjutan dapat diarahkan pada eksplorasi performa algoritma terhadap dataset dengan variasi karakteristik yang lebih komprehensif untuk mengevaluasi tingkat robustness algoritma secara lebih holistik [40].

Kesimpulan

Berdasarkan hasil analisis komprehensif terhadap implementasi algoritma CCABC dan CCGA dalam penanganan dataset dengan tingkat kompleksitas berbeda, dapat disimpulkan bahwa kedua algoritma menunjukkan karakteristik dan keunggulan yang distingtif. CCABC memperlihatkan superioritas dalam aspek konsistensi konvergensi dengan pola perbaikan gradual yang mencapai penurunan nilai fitness sebesar 27,4% pada dataset sederhana dan 18,2% pada dataset kompleks, meskipun membutuhkan waktu komputasi yang lebih tinggi. Di sisi lain, CCGA mendemonstrasikan efisiensi komputasional yang superior dengan karakteristik penurunan nilai fitness yang lebih agresif, mencapai perbaikan hingga 80,1% pada dataset sederhana dan 70,0% pada dataset kompleks. Kedua algoritma menunjukkan korelasi positif antara ukuran populasi dengan optimalitas solusi, dimana konfigurasi populasi yang lebih besar secara konsisten menghasilkan solusi yang lebih optimal, meskipun dengan konsekuensi peningkatan beban komputasional. Temuan ini mengindikasikan bahwa pemilihan algoritma dalam konteks implementasi praktis perlu mempertimbangkan trade-off antara kualitas solusi, konsistensi konvergensi, dan efisiensi komputasional sesuai dengan karakteristik permasalahan yang dihadapi.

Untuk pengembangan penelitian selanjutnya, beberapa aspek kritis yang dapat dieksplorasi meliputi: (1) pengembangan mekanisme adaptif untuk optimasi parameter berdasarkan karakteristik intrinsik dataset, (2) investigasi pendekatan hibridisasi yang mengintegrasikan keunggulan komparatif CCABC dan CCGA untuk menghasilkan algoritma yang lebih efisien dan robust, (3) implementasi teknik paralelisasi dan optimasi algoritma untuk meningkatkan efisiensi komputasional tanpa mengorbankan kualitas solusi, dan (4) evaluasi performa algoritma terhadap dataset dengan variasi karakteristik yang lebih komprehensif untuk menilai tingkat robustness secara lebih holistik. Pengembangan dalam aspek-aspek tersebut diharapkan dapat memberikan kontribusi signifikan dalam meningkatkan efektivitas dan efisiensi algoritma evolusioner dalam penanganan permasalahan optimasi kompleks.

Daftar Pustaka

[1] Potter, M. A., & De Jong, K. A. (1994). "A cooperative coevolutionary approach to function optimization."

[2] Omidvar, M. N., Li, X., Mei, Y., & Yao, X. (2014). "Cooperative co-evolution with differential grouping for large scale optimization."
[3] Li, X., & Yao, X. (2012). "Cooperatively coevolving particle swarms for large scale optimization."
[4] Tonda, A., Lutton, E., & Squillero, G. (2012). "A benchmark for cooperative coevolution."
[5] Sun, W., Wu, Y., Lou, Q., & Yu, Y. (2019). "A Cooperative Coevolution Algorithm for the Seru Production With Minimizing Makespan."
[6] Qiao, D., Duan, L., Li, H., & Xiao, Y. (2022). "Optimization of job shop scheduling problem based on deep reinforcement learning."
[7] Wu, Y., Nie, M., Ma, X., Guo, Y., & Liu, X. (2023). "Co-Evolutionary Algorithm-Based Multi-Unmanned Aerial Vehicle Cooperative Path Planning."
[8] Cao, B., Yang, S., Zhao, J., Yang, P., & Waraich, A. (2018). "Using Parallel Particle Swarm Optimization For RFID Reader-to-reader Anti-collision."
[10] Liu, Z. H., Li, X. H., Wu, L. H., Zhou, S. W., & Liu, K. (2015). "GPU-accelerated parallel coevolutionary algorithm for parameters identification and temperature monitoring in permanent magnet synchronous machines."
[11] Cipta, H., Widyasari, R., & Batubara, F. H. (2023). "Graph Coloring Implementation Using Welch Powell Algorithm In Lecture Scheduling Design For Mathmatics Department."
[12] Gunawan, A., Ng, K. M., & Poh, K. L. (2007). "Solving the Teacher Assignment-Course Scheduling Problem by a Hybrid Algorithm."
[13] Chen, X., Yue, X. G., Man Li, R. Y., Zhumadillayeva, A., & Liu, R. (2020). "Design and Application of an Improved Genetic Algorithm to a Class Scheduling System."
[14] Ma, X., Li, X., Zhang, Q., Tang, K., Liang, Z., Xie, W., & Zhu, Z. (2018). "A survey on cooperative co-evolutionary algorithms."
[15] Zhao, W., Alam, S., & Abbass, H. (2014). "MOCCA-II: A multi-objective co-operative co-evolutionary algorithm."
[16] Gong, D., Xu, B., Yon, Z., Guo, Y., & Yang, S. (2020). "A similarity-based cooperative co-evolutionary algorithm for dynamic interval multiobjective optimization problems."
[17] Ismail, M. A., Mezhuyev, V., Mohamad, M. S., Kasim, S., & Ibrahim, A. O. (2020). "An improved algorithm for optimising the production of biochemical systems."
[18] Jia, Y., Chen, W., Gu, T., Zhang, H., Yuan, H., Kwong, S., & Zhang, J. (2019). "Distributed cooperative co-evolution with adaptive computing resource allocation for large scale optimization."
[19] Pan, Q.-K. (2015). "An effective co-evolutionary artificial bee colony algorithm for steelmaking-continuous casting scheduling."
[20] Yang, Z., Tang, K., & Yao, X. (2008). "Large scale evolutionary optimization using cooperative coevolution."
[21] Aditiya, V. and Herdiana, B. (2021). "Analisis perbandingan algoritma breadth first search (bfs) dan algoritma a*."
[22] Mardia, A. and Sunarto, S. (2021). "Metode algoritma genetika untuk penyusunan jadwal perkuliahan program studi tadris matematika uin sulthan thaha saifuddin jambi."
[23] Akram, M. U., Kurniati, N., & Salim, Y. (2020). "Penerapan algoritma fisher yates shuffle pada sistem pembelajaran tes online berbasis aplikasi."
[24] Nugraha, E. S. (2020). "Simulator edukatif untuk pembelajaran algoritma rapidly-exploring random tree (rrt)."
[25] Berkovič, K., Škrjanc, I., & Jakovljevic, B. (2023). "Group contribution cooperative co-evolution framework for CEC'2013 large-scale optimization problems."
[26] Yin, Z., Gong, Y., & Zhang, J. (2022). "State of the art of adaptive dynamic programming and reinforcement learning."

[27] Velloso, B. P., & Hentenryck, P. V. (2021). "Combining deep learning and optimization for preventive security-constrained DC optimal power flow."
[28] Ji-li, W., & Yang, X. (2024). "Research on multi-objective flexible job shop scheduling problem with setup and handling based on an improved shuffled frog leaping algorithm." [29] Kaveh, A., et al. (2021). "Damage detection using a graph-based adaptive threshold for modal strain energy and improved water strider algorithm."
[30] Zhang, L. (2018). "Coarse-grained parallel AP clustering algorithm based on intra-class and inter-class distance."
[31] Hanafizadeh, P., & Paydar, M. (2013). "A data mining model for risk assessment and customer segmentation in the insurance industry."
[32] Ibáñez, O., et al. (2011). "A cooperative coevolutionary approach dealing with the skull-face overlay uncertainty in forensic identification by craniofacial superimposition."
[33] N., S., & Kobti, Z. (2014). "Dynamic heterogeneous multi-population cultural algorithm for large scale global optimization."
[34] Liu, J., et al. (2022). "Optimization of service scheduling problem for overlapping tower cranes with cooperative coevolutionary genetic algorithm."
[35] Bakos, J. D., et al. (2010). "Combining classifiers for robust hyperspectral mapping using hierarchical trees and class memberships."
[36] Li, Y. (2023). "Identifying the optimal machine learning model for predicting car insurance claims: a comparative study utilising advanced techniques."
[37] Zhu, J., et al. (2025). "Design and implementation of adaptive architecture for complex algorithms."
[38] Mofrad, M. H., et al. (2013). "Adaptive cooperative particle swarm optimizer."
[39] Fister, I., & Zumer, U. (2012). "Memetic artificial bee colony algorithm for large-scale global optimization."
[40] Li, X., et al. (2017). "Mixed second order partial derivatives decomposition method for large scale optimization."