

## MODIFIKASI BARU ALGORITMA KOLONI LEBAH BUATAN UNTUK MASALAH OPTIMASI GLOBAL

Nursyiva Irsalinda, Sugiyarto Surono

Program Studi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Ahmad Dahlan, Yogyakarta  
nursyiva.irsalinda@math.uad.ac.id

**ABSTRACT.** *Artificial Bee Colony (ABC) algorithm is one of metaheuristic optimization technique based on population. This algorithm mimicking honey bee swarm to find the best food source. ABC algorithm consist of four phases: initialization phase, employed bee phase, onlooker bee phase and scout bee phase. This study modify the onlooker bee phase in selection process to find the neighborhood food source. Not all food sources obtained are randomly sought the neighborhood as in ABC algorithm. Food sources are selected by comparing their objective function values. The food sources that have value lower than average value in that iteration will be chosen by onlooker bee to get the better food source. In this study the modification of this algorithm is called New Modification of Artificial Bee Colony Algorithm (MB-ABC). MB-ABC was applied to 4 Benchmark functions. The results show that MB-ABC algorithm better than ABC algorithm.*

**Keywords:** *optimization, artificial bee colony algorithm, selection mechanism, benchmark function.*

**ABSTRAK.** Algoritma koloni lebah buatan (*Artificial Bee Colony*) merupakan teknik optimasi metaheuristik berdasarkan sekawanan (populasi). Algoritma ini meniru perilaku sekawanan lebah madu dalam mencari sumber makanan terbaik. Algoritma ABC terdiri dari empat fase yaitu fase inisialisasi, lebah *employed*, lebah *onlooker* dan lebah *scout*. Kajian ini memodifikasi fase lebah *onlooker* dalam proses seleksi sumber makanan yang akan dicari persekitarannya. Tidak semua sumber makanan yang telah diperoleh lebah *employed* dicari persekitarannya secara random seperti pada algoritma ABC. Sumber makanan dipilih dengan membandingkan nilai fungsi objektifnya. Sumber makanan yang memiliki nilai fungsi objektif lebih buruk dari rata-rata nilai pada saat itu akan diperbaiki oleh lebah *onlooker* sedangkan yang lebih baik tidak akan diperbaiki lagi. Pada kajian ini modifikasi dari algoritma ini dinamakan Modifikasi Baru Algoritma Koloni Lebah Buatan (MB-ABC). MB-ABC diaplikasikan pada 4 fungsi Benchmark. Hasil yang diperoleh menunjukkan bahwa algoritma MB-ABC lebih baik dari algoritma ABC.

**Kata Kunci:** optimasi, algoritma koloni lebah buatan, mekanisme seleksi, fungsi *benchmark*.

## 1. PENDAHULUAN

Beberapa tahun terakhir, metode optimasi berdasarkan populasi telah banyak diteliti seperti Particle Swarm Optimization (PSO) sebuah metode yang mensimulasikan perilaku sekawanan burung atau ikan dalam mencari makan [1], Ant Colony Optimization (ACO) mensimulasikan perilaku sekawanan semut dan sumber makanannya [2] dan Artificial Bee Colony (ABC) sebuah metode yang dikembangkan oleh Karaboga pada tahun 2005 yang terinspirasi oleh perilaku lebah madu dalam mencari sumber makanan terbaik [3]. Performa dan akurasi algoritma ABC diujikan pada permasalahan numerik multidimensi dan telah dibandingkan dengan PSO, differential evolution (DE), evolutionary algorithm (EA) dan Genetic Algorithm (GA) [4-6].

Sejak ditemukan algoritma ABC pada tahun 2005, kajian tentang algoritma ini meningkat signifikan. Algoritma ABC telah digunakan untuk mendesain filter digital IRR [7], menyelesaikan masalah *leaf-constrained minimum tree* [8], sedangkan Rao dkk, membuat algoritma ABC untuk optimasi distribusi network konfigurasi untuk menghilangkan reduksi [9]. Algoritma ABC telah diimplementasikan untuk menyelesaikan masalah minimum spanning tree kuadratik oleh Sundar dan Sighn [10]

Berdasarkan pada aplikasi dan modifikasi algoritma ABC, algoritma ini menjadi lebih baik dalam menyelesaikan permasalahan optimasi. Tetapi ketika dianalisis, terlihat bahwa seleksi persekitaran seluruh sumber makanan yang ada pada algoritma ABC diselesaikan secara random. Singkatnya, proses algoritma ABC menjadi lama dan tidak efektif. Pada kajian ini, mekanisme seleksi sumber makanan atau penyelesaian yang akan diperbaiki atau dicari persekitarannya oleh lebah *onlooker* hanyalah sumber makanan yang memiliki nilai fitness yang lebih buruk dari nilai fitness rata-rata pada iterasi saat itu. Hal ini membuat algoritma ABC menjadi lebih cepat dan efektif.

Pada kajian ini, algoritma ABC dijelaskan pada Bagian 2, mekanisme seleksi ABC diuraikan pada Bagian 3, pada Bagian 4 diberikan hasil eksperimen yang telah dilakukan beserta pembahasan. Bagian 5 adalah kesimpulan dan kajian yang bisa dikembangkan selanjutnya.

## 2. ALGORITMA KOLONI LEBAH (ALGORITMA ABC)

### 2.1. Inisialisasi Permasalahan Optimasi dan Parameter Kontrol dalam Algoritma ABC

Secara umum, permasalahan optimasi diformulasikan sebagai berikut :

$$\min_{x \in X} f(x) \quad (2.1)$$

dengan  $f(x)$  fungsi obyektif yang diminimumkan.  $x$  merupakan sebuah  $SN$ -vektor dari variabel putusan yaitu  $x = [x_1, x_2, \dots, x_{SN}]^T \in X$ . Setiap vektor berdimensi  $D$ .  $x_i \in (lb, ub)$  dimana  $lb$  dan  $ub$  merupakan batas bawah dan batas atas masing-masing elemen vektor  $x$ .

Algoritma ABC terdiri dari dua parameter kontrol:

- Solution Number (SN)* merupakan banyaknya sumber makanan (anggota penyelesaian dalam populasi). Jumlah  $SN$  sama dengan jumlah lebah *employed* dan lebah *onlooker*.
- Maximum Cycle Number (MCN)* merupakan iterasi maksimum dari algoritma ABC.

Selain dua parameter kontrol tersebut terdapat suatu *limit* yang merupakan batas maksimum waktu tunggu suatu penyelesaian yang tidak dapat diperbaiki oleh lebah *employed* maupun lebah *onlooker*. Ketika sampai pada limit yang diberikan maka penyelesaian tersebut digantikan dengan penyelesaian lain yang secara random dihasilkan oleh lebah *scout*. Jumlah limit merupakan perkalian ukuran populasi dengan parameter individu ( $SN \times D$ ).

### 2.2. Inisialisasi Populasi Solusi

Populasi awal dari calon penyelesaian didefinisikan sebagai matriks berukuran  $SN \times D$ .

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1D} \\ x_{21} & x_{22} & \dots & x_{2D} \\ \dots & \dots & \ddots & \dots \\ x_{SN1} & x_{SN2} & \dots & x_{SND} \end{bmatrix} \quad (2.2)$$

dengan setiap elemen matriksnya dibangkitkan menggunakan persamaan berikut

$$x_{ij} = lb + r(ub - lb) \quad (2.3)$$

$$\forall i \in (1, 2, \dots, SN), \quad \forall j \in (1, 2, \dots, D)$$

$x_{ij}$  merupakan komponen vektor ke  $i$  pada dimensi  $j$ ,  $lb$  dan  $ub$  merupakan batas bawah dan batas atas ruang pencarian. Batasan ini diberikan agar penyelesaian tetap berada di dalam ruang pencarian. Sedangkan  $r$  membentuk bilangan random uniform antara 0 dan 1.

### 2.3. Tahap Lebah *Employed*

Pada tahap ini, setiap lebah *employed* ditempatkan pada calon penyelesaian yang telah dihasilkan pada tahap inisialisasi. Selanjutnya, masing-masing lebah *employed* mencari calon penyelesaian baru di persekitaran calon penyelesaian sebelumnya dengan menggunakan persamaan:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (2.4)$$

$\forall k \in (1, 2, \dots, SN), k \neq j$ ,  $\phi_{ij}$  merupakan faktor skala dan  $\phi_{ij} \sim (-1, 1)$ . Ketika penyelesaian yang baru ( $v_{ij}$ ) telah diperoleh, dilakukan seleksi antara calon penyelesaian sebelumnya dengan penyelesaian baru tersebut. Nilai fungsi obyektif dari  $v_i$  dibandingkan dengan  $x_i$ . Jika nilai fungsi obyektif  $v_i$  sama atau lebih baik dari  $x_i$ ,  $v_i$  menggantikan  $x_i$  dan menjadi anggota baru dari populasi, dengan kata lain  $x_i$  tidak digunakan.

### 2.4. Tahap Lebah *Onlooker*

Jumlah lebah *Onlooker* sama dengan jumlah penyelesaian yang telah diperoleh lebah *employed*. Pada tahap ini lebah *Onlooker* menghitung nilai probabilitas seleksi dari tiap penyelesaian yang diperoleh lebah *employed*. Penyelesaian terbaik diseleksi oleh *onlooker* menggunakan nilai probabilitas  $p_i$ .

$$p_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \quad (2.5)$$

dimana  $fit_i$  adalah nilai *fitness* dari solusi  $i$  dan  $SN$  merupakan banyaknya penyelesaian. Dalam menghitung nilai *fitness* untuk masalah meminimuman, digunakan persamaan:

$$fit_i = \begin{cases} \frac{1}{1 + f(x_i)} & \text{jika } f(x_i) \geq 0 \\ 1 + abs(f(x_i)) & \text{jika } f(x_i) < 0 \end{cases} \quad (2.6)$$

dimana  $f(x_i)$  adalah nilai fungsi obyektif dari  $X$ .

Dari persamaan (2.6) jelas bahwa jika nilai  $fit_i$  tinggi, maka semakin besar probabilitas penyelesaian ke- $i$  dipilih. Calon penyelesaian dari lebah *employed* yang memiliki probabilitas terbesar akan dipilih oleh *onlooker*. Lebah *Onlooker* akan menuju penyelesaian yang dipilih sembari mencari penyelesaian baru disekitar penyelesaian yang dipilihnya menggunakan persamaan (2.4). Akan tetapi  $x_{kj}$  pada tahap ini merupakan penyelesaian yang dipilih oleh lebah *onlooker*. Setelah itu, *onlooker* menyeleksi nilai fungsi obyektif antara penyelesaian sebelumnya ( $X$ ) dengan penyelesaian baru yang didapatkan ( $V$ ).

### 2.5. Tahap Lebah Scout

Calon penyelesaian  $x_i$  diasumsikan dapat ditinggalkan ketika calon penyelesaian tersebut tidak dapat diperbaiki lebih lanjut sampai banyaknya percobaan yang ditentukan pada tahap inisialisasi (limit). Pada tahap ini lebah *employed* berubah menjadi lebah *scout*. Lebah *Scout* menghasilkan calon penyelesaian baru secara random menggunakan persamaan (2.3).

## 3. MODIFIKASI BARU ALGORITMA KOLONI LEBAH BUATAN (MB-ABC)

Lebah *employed* pada algoritma ABC mengeksploitasi sumber makanan dan menginformasikannya kepada lebah *onlooker* yang berada di dalam sarang melalui tarian (*wagle dance*). Pada modifikasi ini, lebah *employed* tidak mencari sumber makanan lain disetiap sumber makanan yang telah dieksploitasinya. Lebah *employed* hanya akan mencari sumber makanan lain yang lebih baik terhadap sumber makanan yang kualitas nektarnya lebih buruk dari rata-rata

kualitas nektar pada tahapan tersebut. Sumber makanan baru dicari untuk menggantikan sumber makananan dibawah rata-rata. Dengan hal ini, maka tahapan lebah *employed* akan lebih baik dari tahapan lebah *employed* algoritma ABC

Pada tahap ini, setiap lebah *employed* ditempatkan pada calon penyelesaian yang telah dihasilkan pada tahap inisialisasi seperti pada algoritma ABC. Selanjutnya, lebah *employed* menghitung rata-rata kualitas sumber makanan (nilai fungsi obyektif daricalon penyelesaian) dari populasi tersebut dengan menggunakan persamaan

$$Avg^{pop} = \frac{1}{SN} \sum_i^{SN} f(x_i) \quad (3.1)$$

masing-masing lebah *employed* mencari calon penyelesaian baru dari penyelesaian yang nilai obyektifnya lebih buruk dari nilai fungsi obyektif rata-rata ( $Avg^{pop}$ ). Dalam kajian ini karena meminimumkan suatu fungsi maka dipilih penyelesaian yang nilai fungsi obyektifnya lebih besar dari  $Avg^{pop}$  dengan menggunakan persamaan:

$$v_{bj} = x_{bj} + \varphi_{bj}(x_{bj} - x_{gj}) \quad (3.2)$$

$\forall b \in$  indeks-indeks penyelesaian yang memiliki nilai lebih buruk dari  $Avg^{pop}$ ,  
 $\forall g \in$  indeks-indeks penyelesaian yang memiliki nilai lebih baik dari  $Avg^{pop}$ ,  
 $\varphi_{bj}$  merupakan faktor skala dan  $\varphi_{bj} \sim (0,1)$ . Ketika penyelesaian yang baru ( $v_{bj}$ ) telah diperoleh, dilakukan seleksi antara calon penyelesaian sebelumnya dengan penyelesaian baru tersebut. Nilai fungsi obyektif dari  $v_b$  dibandingkan dengan  $x_b$ . Jika nilai fungsi obyektif  $v_b$  sama atau lebih baik dari  $x_b$ ,  $v_b$  menggantikan  $x_b$  dan menjadi anggota baru dari populasi, dengan kata lain  $x_b$  tidak digunakan.

#### 4. EKSPERIMEN

Untuk melihat performa dari algoritma MB-ABC, algoritma tersebut diaplikasikan untuk menyelesaikan permasalahan optimasi global yaitu meminimumkan fungsi benchmark yang diberikan pada Tabel 4.1.

**Tabel 4.1.** Fungsi *Benchmark* yang digunakan dalam eksperimen

Fungsi	<i>D</i>	Rentan [ <i>lb, ub</i> ]	Formula
$f_1$ (Sphere)	40	[-100,100]	$f(x) = \sum_{i=1}^D x_i^2$
$f_2$ (Rastrigin)	40	[-5.12,5.12]	$f(x) = 10D + \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i)]$
$f_3$ (Schwefel)	40	[-500,500]	$f(x) = 418.9829D - \sum_{i=1}^D x_i \sin(\sqrt{ x_i })$
$f_4$ (Rosenbrock)	40	[-30,30]	$f(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$

Algoritma ABC dan algoritma BM-ABC digunakan untuk meminimumkan fungsi *benchmark* yang diberikan pada Tabel 4.1. Dalam rentan yang diberikan seluruh fungsi *Benchmark* memiliki nilai minimum global 0. MCN (iterasi) yang diberikan sebanyak 1000 dan algoritma diulang sebanyak 30 kali. Rata-rata (*mean*) dan standar deviasi yang diperoleh dari eksperimen ini digunakan untuk membandingkan performa kedua algoritma.

## 5. HASIL DAN PEMBAHASAN

**Tabel 5.1** Mean dan Standar Deviasi yang diperoleh algoritma ABC dan MB-ABC

No	Fungsi	Algoritma ABC		Algoritma MB-ABC	
		Mean	Std. Deviasi	Mean	Std. Deviasi
1	$f_1$	7.5276e-016	1.3629e-016	6.9016e-17	1.0940e-017
2	$f_2$	7.5791e-015	1.9653e-014	0	0
3	$f_3$	5.2995e-016	5.7007e-015	4.9205e-14	3.2536e-015
4	$f_4$	0,1377	0,2410	0,0228	0,0934

Dari Tabel 5.1 diperoleh mean dan standar deviasi dari algoritma ABC dan algoritma MB-ABC. Mean atau rata-rata nilai semua fungsi objektif yang diperoleh menggunakan algoritma MB-ABC lebih rendah dibanding dengan mean pada algoritma ABC hal ini menunjukkan bahwa hasil yang diperoleh semakin mendekati nilai minimum yaitu 0 dan memiliki standar deviasi yang lebih rendah. Hal ini menunjukkan bahwa modifikasi yang diberikan pada kajian ini lebih efektif dan efisien dibanding dengan Algoritma ABC

## 6. KESIMPULAN DAN SARAN

Modifikasi baru dari algoritma ABC untuk menyelesaikan permasalahan telah diuraikan. Modifikasi ini diberikan pada mekanisme seleksi persekitaran pada tahap lebah *employed* yaitu dengan membandingkan atau mengujikan masing-masing nilai fungsi obyektif yang dimiliki oleh lebah *employed* terhadap rata-rata nilai fungsi obyektif pada tahap lebah *employed* disetiap iterasi. Modifikasi algoritma telah diimplementasikan pada beberapa fungsi *Benchmark*. Dari eksperimen yang dilakukan menunjukkan bahwa algoritma yang diusulkan memperoleh hasil yang lebih baik bila dibandingkan dengan algoritma ABC. Dalam rangka meningkatkan kinerja dan kerja selanjutnya, disarankan untuk menerapkan modifikasi algoritma ini untuk menyelesaikan masalah fungsi diskrit.

## DAFTAR PUSTAKA

- [1] Dorigo, M. dan Stutzle, T. *Ant Colony Optimization*. MIT Press, Cambridge. 2004.
- [2] Eberhart, R.C. dan Kennedy, I., *A New Optimizer Using Particle swarm Theory*, Proceedings of the Sixth International Symposium on Micro Machine and Human Science 1995 in Nagoya, 39-43.
- [3] Karaboga, D., *An Idea Based on Honey Bee Swarm for Numerical Optimization*, Tehnical Report-TR06, Department of Computer Engineering, Engineering Faculty, Erciyes University, 2005.
- [4] Karaboga, D. dan Basturk, B., *On the Performance of Artificial Bee Colony (ABC) Algorithm.*, Applied Soft Computing, **8** (2008), 687-697.
- [5] Karaboga, D. dan Akay, B., *A Comparative Study of Artificial Bee Colony Algorithm Numerical.* Applied Mathematics and Computation, **214** (2009). 208-132.
- [6] Akay, B., *Performance Analysis of Artificial Bee Colony Algorithm on Numerical Optimization Problems*, Ph.D Thesis, Erciyes University, Graduate School of natural and Applied Sciences, Kayseri, 2009.



- [7] Karaboga, N., *A New Design Method Based on Artificial Bee Colony Algorithm for Digital IIR Filters*, Journal of the Franklin Institute, **346** (2009), 328-348.
- [8] Singh, A., *An Artificial Bee Colony Algorithm for the Leaf-Constrained Minimum Spanning Tree Problem*, Applied Soft Computing, **9** (2009), 625-631.
- [9] Rao, R.S., Narasimham, S., dan Ramalingaraju, M., *Optimization of Distribution Network Configuration for Loss Reduction Using Artificial Bee Colony Algorithm*, International Journal of Electrical Power and Energy Systems Engineering, **1** (2008), 116-122.
- [10] Sundar, S. dan Singh, A., *A Swarm Intelligence Approach to the Quadratic Minimum Spanning Tree Problem*, Information Sciences, **180** (2010), 3182-3191.
- [11] Akay, B. dan Karaboga, D., *A modified Artificial Bee Colony Algorithm for Real-Parameter Optimization*, Information Science, **192** (2012), 120-142.
- [12] Karaboga, D. dan Akay, B., *A Modified Artificial Bee Colony (ABC) Algorithm for Constrained Optimization Problems*, Applied Soft Computing, **11** (2011), 3021-3031.
- [13] Kiran, M.S. dan Babalik, A., *Improved Artificial Bee Colony Algorithm for Continuous Optimization Problems*, Journal of Computer and Communications, **2** (2014), 108-116.

