

BAB III

METODOLOGI PENELITIAN

Bab ini berisi tentang identifikasi masalah optimisasi global fungsi, tahapan penelitian, algoritma DE, dan contoh implementasi algoritma DE untuk pencarian nilai optimal global dari fungsi *Stepint*.

3.1 Identifikasi Masalah

Permasalahan yang dibahas dalam penelitian ini adalah pencarian solusi optimal global dari beberapa fungsi *differentiable* dan fungsi *non-differentiable* menggunakan algoritma DE. Pencarian optimal global dari fungsi *differentiable* dapat memanfaatkan turunan pertama dan (atau) turunan kedua fungsi guna mengidentifikasi keberadaan nilai optimal lokal. Selanjutnya nilai optimal global diperoleh dengan membandingkan nilai terbesar (terkecil) dari nilai titik optimal lokal tersebut. Untuk beberapa fungsi, turunan dari fungsi sulit untuk dicari. Selain itu, untuk fungsi *non-differentiable* pencarian optimal global tidak bisa ditentukan dengan turunan fungsi tersebut. Oleh karena itu, diperlukan metode untuk mencari solusi optimal global yang dapat diterapkan pada pencarian optimal global untuk fungsi *differentiable* dan *non-differentiable*.

3.2 Tahapan Penelitian

Tahapan penelitian yang dilakukan adalah sebagai berikut.

1. Studi Pustaka

Pada tahap ini dilakukan pemahaman definisi, teori, dan informasi lain yang menunjang penelitian penulis dalam permasalahan optimisasi global fungsi dan algoritma DE. Studi pustaka dicari dari berbagai sumber seperti: jurnal, buku panduan, akhir, prosiding nasional dan internasional. Hasil dari studi pustaka ini adalah informasi mengenai beberapa studi kasus optimisasi global fungsi yang telah diselesaikan dengan metode metaheuristik tertentu.

2. Pengumpulan Data

Dalam masalah pencarian solusi optimal global fungsi menggunakan algoritma DE, diperlukan data penelitian berupa 16 fungsi uji (*benchmark*) yang terdiri dari 8 fungsi *differentiable* dan 8 fungsi *non-differentiable*. Data penelitian tersebut diperoleh dari jurnal: “*A Literature Survey of Benchmark Functions for Global Optimization Problems*” (Jamil dan Yang, 2013) dan “*A Collection of 30 Multidimensional Functions for Global Optimization Benchmarking*” (Plevris dan Solorzano, 2022).

3. Teknik Penyelesaian Algoritma DE

Masalah pencarian solusi optimal global fungsi *differentiable* dan *non-differentiable* diselesaikan menggunakan algoritma DE melalui beberapa tahapan. Langkah pertama yang dilakukan adalah penentuan parameter awal dan insialisasi vektor target. Kedua, lakukan mutasi untuk mendapatkan vektor mutan. Ketiga, lakukan *crossover*, yaitu penyilangan antara vektor target dan vektor mutan untuk memperoleh vektor percobaan dalam keragaman populasi. Keempat, lakukan seleksi antara vektor percobaan dan vektor target dengan membandingkan nilai *fitness* keduanya.

4. Implementasi DE dengan Matlab

Pada tahapan ini, algoritma DE diimplementasikan menggunakan *software* Matlab R2021a untuk mencari nilai global minimum dari 16 fungsi uji yang terdapat pada data penelitian. Selain itu, akan dilakukan juga uji sensitivitas parameter DE dengan tujuan untuk mengetahui tingkat signifikansi dari F, CR, NP, G, terhadap hasil optimal global fungsi.

5. Validasi

Pada tahapan ini, akan dilakukan validasi hasil dengan membandingkan nilai global minimum yang diperoleh dari pengujian dengan nilai global minimum dari jurnal yang ada di data penelitian. Selain itu, pada tahapan ini juga akan dihitung performa algoritma DE untuk melihat tingkat eror dan seberapa baik algoritma tersebut. Persamaan yang digunakan adalah:

$$\text{Performa DE} = |\text{Nilai Optimal Acuan} - \text{Nilai Optimal DE}| \quad (3.1)$$

6. Analisis Hasil

Analisis hasil akan dilakukan setelah implementasi algoritma DE mencapai iterasi maksimum dan diperoleh nilai optimal global (titik global minimum dan nilai *fitness* minimum) dari 16 fungsi uji dengan melihat grafik *line plot* yang menggambarkan nilai global minimum yang diperoleh dari tiap iterasi serta grafik 2D yang menunjukkan letak nilai global minimum fungsi. Selain itu, akan dilakukan juga analisis sensitivitas parameter DE berdasarkan ketentuan nilai yang telah ditetapkan pada tahap implementasi.

7. Penarikan Kesimpulan

Pada tahapan ini, penarikan kesimpulan dilakukan setelah dilakukan analisis hasil global minimum dan performa algoritma DE untuk 16 fungsi uji, serta hasil sensitivitas parameter DE.

3.3 Algoritma *Differential Evolution* (DE)

Algoritma DE merupakan algoritma optimisasi yang termasuk ke dalam metode metaheuristik dan memiliki skema evolusi (mutasi, rekombinasi, dan seleksi) di dalam tahapannya. Dalam mencari nilai optimal global fungsi, algoritma DE membutuhkan batas (atas dan bawah) fungsi serta parameter-parameter yang harus ditentukan di awal sebelum pemrosesan tahapan-tahapannya. Parameter DE terdiri dari:

1. Jumlah variabel atau dimensi fungsi ($N \in \mathbb{N}$),
2. Banyaknya vektor populasi ($NP \in \mathbb{N}$),
3. Banyaknya generasi atau iterasi ($G \in \mathbb{N}$),
4. Faktor skala mutasi ($F \in [0,2]$), dan
5. Nilai probabilitas rekombinasi atau *rate crossover* ($CR \in [0.1]$).

Setelah menentukan parameter DE, maka dapat dilakukan pemrosesan tahapan algoritma DE yang terdiri dari tahap inisialisasi (*initialization*), mutasi (*mutation*), rekombinasi (*crossover*), dan seleksi (*selection*). Berikut ini adalah tahapan-tahapan algoritma DE.

1. Inisialisasi (*Initialization*)

Inisialisasi bertujuan untuk menentukan vektor target dalam populasi awal. Vektor target dapat ditulis sebagai:

$$X_{m,G} = (X_{m,G}^1, \dots, X_{m,G}^N), m = (1, 2, 3 \dots, NP) \quad (3.1)$$

Sebelum dilakukan inisialisasi, perlu ditentukan batas atas (X_{max}) dan batas bawah (X_{min}) ruang pencarian. Batas atas dan batas bawah dapat ditentukan dari informasi fungsi yang terdapat di data penelitian. Selanjutnya, inisialisasi vektor target melalui persamaan:

$$X_{m,G} = X_{min} + rand(0,1) \cdot (X_{max} - X_{min}) \quad (3.2)$$

dengan $rand(0,1)$ adalah nilai *real* acak dari 0 sampai 1 yang dibangkitkan lewat distribusi uniform. Selanjutnya, hitung nilai *fitness* dari tiap vektor target. Nilai *fitness* didapat dengan cara memasukkan nilai kromosom vektor target (x_1, x_2) ke dalam persamaan fungsi yang diuji.

2. Mutasi (*Mutation*)

Mutasi bertujuan untuk membentuk vektor mutan atau vektor donor, yaitu:

$$V_{m,G+1} = (V_{m,G+1}^1, \dots, V_{m,G+1}^N), m = (1, 2, 3 \dots, NP) \quad (3.3)$$

Mutasi dilakukan dengan memilih 3 vektor secara acak dalam populasi, yaitu: $X_{r_1,G}$, $X_{r_2,G}$, dan $X_{r_3,G}$, dengan $r_1, r_2, r_3 \in \{1, 2, 3, \dots, NP\}$ dan $r_1 \neq r_2 \neq r_3$. Vektor $X_{r_2,G}$ dan $X_{r_3,G}$ dipilih untuk mendapatkan nilai *vector difference*, kemudian nilainya akan dikalikan dengan faktor skala mutasi $F \in [0,2]$, sedangkan $X_{r_1,G}$ dipilih sebagai vektor basis. Kemudian, dilakukan mutasi melalui persamaan:

$$V_{m,G+1} = X_{r_1,G} + F \cdot (X_{r_2,G} - X_{r_3,G}) \quad (3.4)$$

3. Rekombinasi (*Crossover*)

Rekombinasi bertujuan untuk menambah keragaman vektor dalam populasi dengan membentuk vektor percobaan atau *trial vector*. Pada tahap rekombinasi, dilakukan penyilangan antara vektor target $X_{m,G+1}$ dengan vektor mutan $V_{m,G+1}$.

Trial vektor didefinisikan sebagai:

$$U_{m,G+1}^j = (U_{m,G+1}^1, \dots, U_{m,G+1}^N), m = (1, 2, 3 \dots, NP) \quad (3.5)$$

di mana:

$$U_{m,G+1}^j = \begin{cases} V_{m,G+1}^j & \text{jika } (rand(0,1) \leq CR \text{ atau } j = \delta) \\ X_{m,G+1}^j & \text{jika } (rand(0,1) > CR \text{ dan } j \neq \delta) \end{cases} \quad (3.6)$$

dengan CR menyatakan nilai probabilitas rekombinasi (*rate crossover*) dan $CR \in 0.1$, δ menyatakan kromosom yang dipilih secara acak untuk memastikan minimal ada satu vektor donor di dalam vektor percobaan, $\delta = \{1, 2, \dots, N\}$, $rand(0,1)$ menyatakan nilai *real* random di antara 0 sampai dengan 1, $U_{m,G+1}^j$ menyatakan variabel ke- j dari vektor percobaan, $V_{m,G+1}^j$ menyatakan variabel ke- j dari vektor mutan, $X_{m,G+1}^j =$ variabel ke- j dari vektor target.

Selanjutnya, cek apakah nilai kromosom vektor percobaan (x_1, x_2) berada di dalam batas fungsi atau tidak. Kemudian, hitung nilai *fitness* dari tiap vektor percobaan dengan cara memasukkan nilai (x_1, x_2) ke dalam persamaan fungsi.

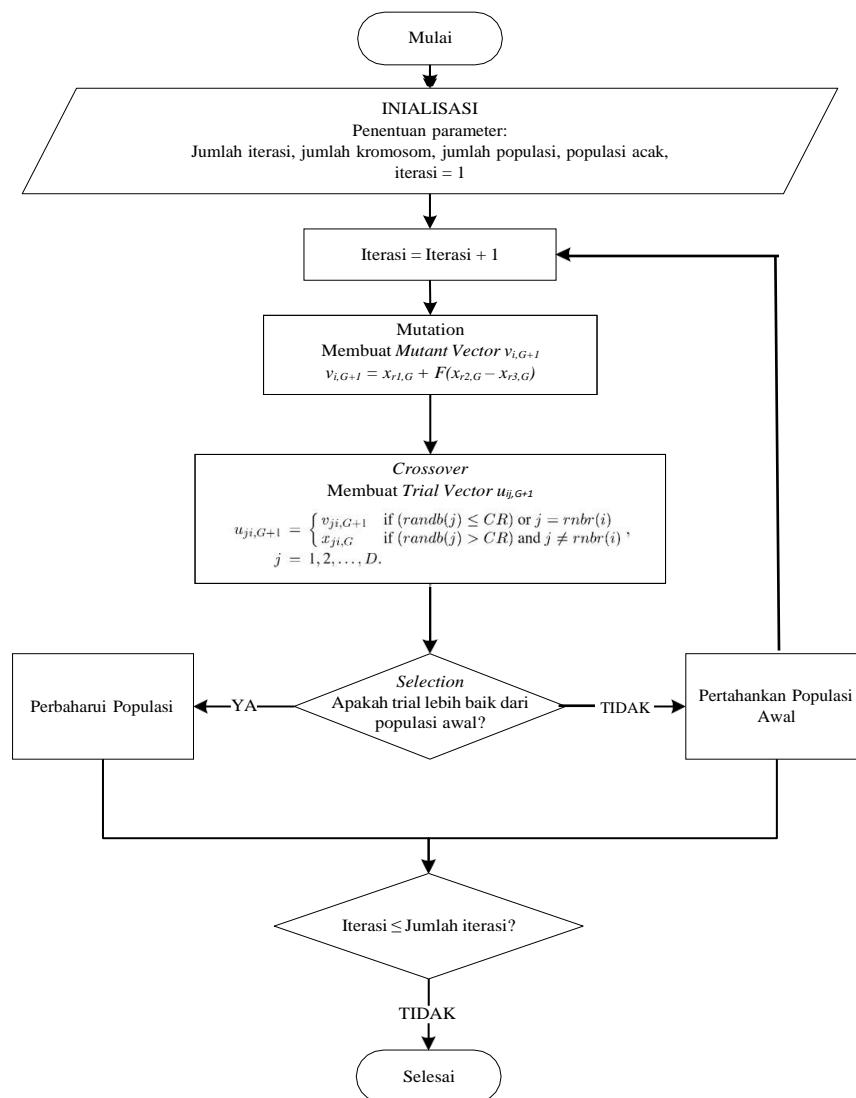
4. Seleksi (*Selection*)

Seleksi bertujuan untuk menentukan keberadaan vektor target di iterasi berikutnya. Skema seleksi dilakukan antara vektor percobaan $U_{m,G+1}$ dan vektor target $X_{m,G}$, dengan membandingkan nilai *fitness* keduanya dengan aturan:

$$\begin{aligned} (1) & \text{ Jika } f(U_{m,G+1}) < f(X_{m,G}), \text{ maka } X_{m,G} = U_{m,G+1}, f(X_{m,G}) = f(U_{m,G}) \\ (2) & \text{ Jika } f(U_{m,G+1}) \geq f(X_{m,G}), \text{ maka } X_{m,G} = X_{m,G}, f(X_{m,G}) = f(X_{m,G}) \end{aligned} \quad (3.7)$$

Proses mutasi, rekombinasi, dan seleksi diulang terus menerus dalam iterasi. Ketika proses DE telah mencapai iterasi maksimum dan sudah diperoleh nilai optimal global fungsi, maka proses dihentikan.

Berikut ini diagram alir implementasi algoritma DE dapat dilihat pada Gambar 3.1.



Gambar 3.1. Cara kerja Algoritma DE

3.4 Implementasi Algoritma DE untuk Optimasi Fungsi *Stepint*

Berikut diberikan contoh langkah-langkah algoritma DE dalam pencarian optimal global fungsi *Stepint*. *Stepint* adalah fungsi *non-differentiable* dengan karakteristik: *Discontinuous*, *Separable*, dan *Unimodal*. Rumus fungsi *Stepint*:

$$f_{15}(\mathbf{x}) = 25 + \sum_{i=1}^n (\lfloor x_i \rfloor) \quad (3.8)$$

dengan batas fungsi: $-5.12 \leq x_i \leq 5.12$.

Langkah 1: Misalkan diambil parameter DE sebagai berikut:

$$N = 2, NP = 10, G = 1, F = 0.2, \text{ dan } CR = 0.9.$$

Langkah 2: Inisialisasi vektor target ($X_{m,G}$) menggunakan Persamaan 3.2, yaitu:

$$X_{m,G} = -5.12 + rand(0,1) \cdot (5.12 - (-5.12))$$

Selanjutnya, hitung nilai *fitness* dari tiap vektor target dengan cara memasukkan nilai vektor target (x_1, x_2) ke dalam Persamaan 3.8. Inisialisasi fungsi *Stepint* dapat dilihat pada Tabel 3.1.

Tabel 3.1. Inisialisasi Fungsi *Stepint*

$X_{m,g}$	x_1	x_2	$f(x_1, x_2)$
1	-1.89210	-5.12000	17
2	-2.05100	2.13680	24
3	-1.18980	-1.52640	21
4	0.07820	4.73360	29
5	-1.19860	-3.13920	19
6	-0.58590	4.11810	28
7	5.12000	0.21380	30
8	-1.49560	-4.25480	18

Langkah 3: Kemudian, lakukan Mutasi untuk membentuk vektor mutan ($V_{m,G+1}$) dengan cara memilih 3 vektor target dari populasi secara acak dalam populasi, yaitu: $X_{r_1,G}, X_{r_2,G}, X_{r_3,G}$, dengan $r_1, r_2, r_3 \in \{1, 2, \dots, NP\}$, $r_1 \neq r_2 \neq r_3$, kemudian operasikan Persamaan 3.4, yaitu:

$$V_{m,G+1} = X_{r_1,G} + 0.2 \cdot (X_{r_2,G} - X_{r_3,G})$$

Mutasi vektor mutan dapat dilihat pada Tabel 3.2.

Tabel 3.2. Mutasi Fungsi *Stepint*

$V_{m,G+1}$	x_1	x_2	$f(x_1, x_2)$	r_1	r_2	r_3
1	-2.06434	-4.38736	18	1	2	3
2	5.29224	-0.51884	29	7	3	2
3	0.21690	5.12976	30	4	5	1
4	-2.11040	1.91368	24	2	8	5
5	-1.62842	-4.37790	18	8	4	6
6	-0.90066	3.21414	27	6	7	4
7	-1.27790	-3.31224	20	5	1	8
8	-2.62400	-1.14180	21	3	2	7

Langkah 5: Selanjutnya, pilih $\delta = 1$ dan lakukan Rekombinasi dengan menyilangkan vektor target dengan vektor mutan sesuai aturan pada Persamaan 3.6 untuk mendapatkan vektor percobaan dalam populasi. Kemudian, cek apakah nilai dari kromosom vektor berada di dalam batas fungsi atau tidak. Jika berada di luar batas fungsi maka ganti nilai kromosom vektor percobaan dengan nilai batas fungsi terdekat. Lalu, hitung nilai *fitness* dari tiap vektor percobaan. Tahap rekombinasi dapat dilihat pada Tabel 3.3.

Tabel 3.3. Rekombinasi Fungsi *Stepint*

j	X	V	r	$\delta = 1$	$r \geq CR$	$\delta = j$	U	Cek Batas
1	-1.89210	-2.06434	0.97469	-1.89210	TRUE	TRUE	-1.89210	-1.89210
	-5.12000	-4.38736	0.87514	-5.12000	FALSE	TRUE	-4.38736	-4.38736
2	-2.0510	5.29224	0.14652	-1.89210	FALSE	FALSE	5.29224	5.12000
	2.13680	-0.51884	0.53714	-5.12000	FALSE	FALSE	-0.51884	-0.51884
3	-1.18980	0.21690	0.07635	-1.89210	FALSE	FALSE	0.21690	0.21690
	-1.52640	5.12976	0.12778	-5.12000	FALSE	FALSE	5.12976	5.12000
4	0.07820	-2.11040	0.97635	-1.89210	TRUE	FALSE	-2.11040	-2.11040
	4.73360	1.91368	0.82778	-5.12000	FALSE	FALSE	1.91368	1.91368
5	-1.19860	-1.62842	0.28387	-1.89210	FALSE	FALSE	-1.62842	-1.62842
	-3.13920	-4.37790	0.53909	-5.12000	FALSE	FALSE	-4.37790	-4.37790

j	X	V	r	$\text{delta} = 1$	$r \geq CR$	$\text{delta} = j$	U	Cek Batas
6	-0.58590	-0.90066	0.28090	-1.89210	FALSE	FALSE	-0.90066	-0.90066
	4.11810	3.21414	0.93644	-5.12000	TRUE	FALSE	3.21414	3.21414
7	5.12000	-1.27790	0.61552	-1.89210	FALSE	FALSE	-1.27790	-1.27790
	0.21380	-3.31224	0.06426	-5.12000	FALSE	FALSE	-3.31224	-3.31224
8	-1.49560	-2.62400	0.99024	-1.89210	TRUE	FALSE	-2.62400	-2.62400
	-4.25480	-1.14180	0.29538	-5.12000	FALSE	FALSE	-1.14180	-1.14180

Langkah 6: Setelah proses rekombinasi selesai, maka dilanjut ke tahap Seleksi sesuai dengan aturan pada Persamaan 3.7. Tahap Seleksi dapat dilihat pada Tabel 3.4.

Tabel 3.4. Seleksi Fungsi *Stepint*

X			U			X		
x_1	x_2	$f(x_1, x_2)$	x_1	x_2	$f(x_1, x_2)$	x_1^*	x_2^*	$f(x_1^*, x_2^*)$
-1.89210	-5.12000	17	-1.89210	-4.38736	18	-1.89210	-5.12000	17
-2.05100	2.13680	24	5.12000	-0.51884	29	-2.05100	2.13680	24
-1.18980	-1.52640	21	0.21690	1.91368	26	-1.18980	-1.52640	21
0.07820	4.73360	29	-2.1104	1.91368	23	-2.1104	1.91368	23
-1.19860	-3.13920	19	-1.62842	-4.37790	18	-1.62842	-4.37790	18
-0.58590	4.11810	28	-0.90066	3.21414	27	-0.90066	3.21414	27
5.12000	0.21380	30	-1.27790	-3.31224	19	-1.27790	-3.31224	19
-1.49560	-4.25480	18	-2.62400	-1.14180	20	-1.49560	-4.25480	18

Berdasarkan Tabel 3.4, dapat dilihat bahwa fungsi *Stepint* memiliki titik global minimum $(x_1^*, x_2^*) = (-1.8921, -5.12)$ dan nilai *fitness* minimum $f(x_1^*, x_2^*) = 17$.