Penentuan Akar-Akar Sistem Persamaan Tak Linier dengan Kombinasi Differential Evolution dan Clustering

Jamaliatul Badriyah
Jurusan Matematika, Universitas Negeri Malang
Email: jamailatul.badriyah.mat@um.ac.id

Abstrak

Pada artikel ini akan dibahas tentang modifikasi dari Differential Evolution untuk menentukan akar-akar dari suatu sistem persamaan tak linier. Algoritma Differential Evolution ini nantinya akan dikombinasikan dengan teknik clustering. Algoritma Differential Evolution dengan clustering ini mampu untuk menentukan akar-akar dari system persamaan tak linier dalam satu kali running.

Kata Kunci: Sistem Persamaan tak Linier, Differential Evolution, Clustering

Abstract

This paper deal with finding all roots of systems of nonlinear equations using combination between Differential Evolution ad Clustering Technique. The result shows that Differential Evolution with clustering can be a promising algorithm for finding all roots of systems of nonlinear equations.

Keywords: Systems of Nonlinear Equations, Differential Evolution, Clustering

Pendahuluan

Banyak masalah dalam Matematika dimodelkan dalam suatu persamaan nonlinier dan dalam menyelesaikannya mengharuskan kita untuk menyelesaikan persamaan nonlinier tersebut dengan mencari semua akar-akar yang ada pada persamaan tersebut. Suatu masalah nonlinear dapat diformulasikan sebagai berikut.

$$f(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{pmatrix}$$

Dimana $x \in S = [a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_n, b_n] \subset \mathbb{R}^n$ dan setidaknya satu dari $f_1, f_2, \ldots, f_n$ merupakan fungsi tak linier. Sehingga suatu titik $x \in S$ dikatakan sebagai akar dari suatu persamaan tak linier $f(x)$ jika memenuhi

$$\begin{gathered} f_1(x) = 0 \\ f_2(x) = 0 \\ \vdots \\ f_n(x) = 0 \end{gathered}$$

Banyak metode yang sudah diajukan oleh para peneliti untuk mencari semua akar-akar pada suatu persamaan nonlinier. Metode yang sering digunakan adalah metode Newton ataupun modifikasinya seperti Metode quasi-Newton. Akan tetapi, metode ini mempunyai kelemahan yaitu keakuratan dari hasil ditentukan oleh keakuratan titik awal yang diberikan[1]. Untuk mengatasi kelemahan tersebut, pada saat ini banyak berkembang penggunaan algoritma metaheuristic sebagai salah satu alternatif untuk menyelesaikan persamaan nonlinier. Beberapa algoritma metaheuristic yang sudah berhasil dibangun untuk menyelesaikan masalah persamaan nonlinier tersebut antara lain, Metode Multistart dan Minfinder global optimization oleh Tsoulos [2], Hybrid Metaheuristic dengan Fuzzy Clustering Means oleh Sacco [3], dan Algoritma Dynamics Spiral dengan clustering oleh Kania[1].

Pada artikel ini, akan dibahas tentang penentuan semua akar-akar real dari suatu persamaan nonlinier dengan menggunakan algoritma Differential Evolution yang sudah dimodifikasi dengan clustering. Pada algoritma ini, akan didapatkan semua akar real dari suatu persamaan nonlinier dalam satu kali running. Kemudian algoritma Differential Evolution dengan clustering yang sudah dibangun akan diuji cobakan pada beberapa masalah benchmark.

Metode

1. Differential Evolution

Differential Evolution atau yang selanjutnya akan disebut DE merupakan suatu algoritma Metaheuristic yang dikembangkan oleh Storne and Price [4]. Secara umum, ada 4 tahapan yang harus dikerjakan pada algoritma DE, yaitu inisiasi, mutasi, crossover, dan seleksi. Tahapan pada DE bisa dituliskan seperti berikut.

Input:

$F \in (0,1)$

$Cr \in [0,1]$

D dimensi permasalahan

$NP = 10.D$

$Gmax$ Generasi Maksimum

BB Batas Bawah

BA Batas Atas

Proses

Langkah 1: Inisiasi

Untuk setiap $j = \{1,2,\dots,D\}$ dan $i = \{1,\dots,NP\}$, bangkitkan populasi awal secara acak dengan rumus:

$$x_{j,i,0} = BB + (BA - BB)rand_j[0,1)$$

Langkah 2: Mutasi

Pilih secara acak $r_0, r_1, r_2, i \in \{1,\dots,NP\}$ dengan $r_0 \neq r_1 \neq r_2 \neq i$

Akan dibentuk suatu vektor mutan $V_{i,g+1} = \{v_{1,i,g+1}, v_{2,i,g+1}, \dots, v_{D,i,g+1}\}$ dengan rumus:

$$V_{i,g+1} = X_{r_0,g} + F\left(X_{r_1,g} - X_{r_2,g}\right)$$

Langkah 3: Crossover

Akan dibentuk vektor trial $U_{i,g+1} = \{u_{1,i,g+1}, u_{2,i,g+1}, \dots, u_{D,i,g+1}\}$dengan rumus:

$$u_{j,i,g+1} = \begin{cases} v_{j,i,g+1}\ jika\ rand_j(0,1) \leq Cr\ atau\ j = j_{rand} \\ x_{j,i,g}\ jika\ rand_j(0,1) > Cr\ atau\ j \neq j_{rand} \end{cases}$$

Langkah 4: Seleksi

$$X_{i,g+1} = \begin{cases} U_{i,g+1}\ jika\ f\left(U_{i,g+1}\right) \leq f\left(X_{i,g}\right) \\ X_{i,g}\ \ jika\ yang\ lainnya \end{cases}$$

Kembali ke langkah 2 dan seterusnya sampai $g = Gmax$

2. Teknik Clustering

Kania [1] memanfaatkan teknik clustering untuk mencari akar-akar dari suatu system persamaan non linier dengan menambahkan teknik clustering pada algoritma Dinamic Spiral. Hasilnya, teknik clustering ini bisa menemukan semua akar yang mungkin dari suatu system persamaan tak linier dalam satu kali running.

Teknik clustering ini dimulai dengan mengevaluasi $F(X) = \frac{1}{1+\sum_{i=1}^{n} f_i(x_i)}$ dari suatu populasi awal. X yang mempunyai nilai $F(X) = 1$ berarti berada di sekitar akar dari fungsi f. Misal diberikan suatu nilai $0 < \gamma < 1$, jika dua titik x_i dan x_j mempunyai $F(x_i)$ dan $F(x_j)$ yang lebih besar dari γ maka kedua titik tersebut dimasukkan ke dalam kandidat. Kemudian misalkan $x_t = \frac{x_i+x_j}{2}$ merupakan titik tengah dari x_i dan x_j, lakukan evaluasi terhadap $F(x_t)$.

- Jika $F(x_i)$ dan $F(x_j)$ lebih besar dari $F(x_t)$, maka x_i dan x_j merupakan kandidat yang potensial untuk menjadi akar fungsi f. Kemudian kita bisa membentuk dua cluster yang berbeda dengan masing-masing x_i dan x_j sebagai pusatnya
- Jika $F(x_t)$ berada di antara $F(x_i)$ dan $F(x_j)$, maka kandidat yang potensial untuk menjadi akar adalah titik yang mempunyai nilai F lebih besar dari $F(x_t)$
- Jika $F(x_t)$ lebih besar dari $F(x_i)$ dan $F(x_j)$, maka ketiga titik x_t, x_i dan x_j merupakan kandidat yang potensial untuk menjadi akar

Kemudian dilakukan evaluasi terhadap nilai F dari masing-masing titik yang menjadi kandidat potensial untuk menjadi akar. Kandidat potensial yang dipertahankan adalah kandidat x yang mempunyai nilai $1 - F(x) < \delta$ untuk suatu nilai $0 < \delta < 1$. Misal terdapat n kandidat akar yang dipertahankan, lakukan evaluasi terhadap masing-masing kandidat sehingga tidak ada kandidat yang sama dengan aturan untuk setiap $i, j = 1, .., n$,

- Pilih x_i dan x_j sebagai akar jika $\|x_i - x_j\| > \delta$
- Jika $\|x_i - x_j\| \leq \delta$ pilih salah satu yang mempunyai nilai F lebih besar.

Hasil dan Diskusi

1. Differential Evolution dengan Teknik Clustering

Pada penelitian ini, akan diperkenalkan suatu modifikasi dari algoritma Differential Evolution dengan menambahkan teknik clustering didalamnya untuk menentukan akar-akar dari suatu system persamaan tak linier. Algoritma ini dimulai dengan pembangkitan populasi awal dengan aturan inisiasi pada DE. Kemudian teknik clustering dilaksanakan pada populasi awal yang telah dibangkitkan tersebut. Proses berlanjut pada mutasi, crossover, dan seleksi seperti pada tahapan DE. Kemudian individu baru yang didapat dari hasil seleksi dikenakan teknik clustering dan proses berlanjut sampai ditemukan akar-akar yang menjadi solusi dari sistem persamaan tersebut. Secara rinci, Differential Evolution dengan Clustering dapat ditulis sebagai berikut.

Input

Input pada DE: $F, Cr, D, NP, Gmax, BB, BA$
$m(> 1)$ banyaknya cluster
$\gamma(0 < \gamma < 1)$ cut-off dari F
$\varepsilon(0 < \varepsilon < 1)$ parameter untuk diterima sebagai akar
$\delta(0 < \delta < 1)$ parameter untuk menentukan dua akar sama atau tidak
$k(> 1)$ maximum iterasi untuk setiap cluster

Proses

Langkah 1: Inisiasi pada DE

Langkah 2: Clustering

- Evaluasi nilai F untuk populasi awal

- bentuk $x' = x_{i_g}(0)$ dimana $i_g = \arg maxF(x_i(0))$, dimana $i = 1,..,m$
- simpan x' sebagai pusat dari cluster pertama dengan jari-jarinya $\frac{1}{2}(min_t|b_t - a_t|)$ dengan $t = 1, \dots, n$
- lakukan uji jika $F(x_i) > \gamma$ maka x_i bukanlah pusat dari cluster tersebut, kemudian lakukan: *fungsi cluster*(input y)
    - ➢ temukan cluster yang titik pusatnya paling dekat dengan y
    - ➢ missal cluster tersebut adalah C dengan titik pusatnya x_c
    - ➢ buat x_t titik tengah antara y dan x_c
    - ➢ evaluasi nilai F untuk masing-masing x_t, x_c, y dengan aturan:
        - ✓ jika $F(y), F(x_c) > F(x_t)$ maka buat cluster baru dengan pusat y dan jari-jari sama dengan jarak antara y dan x_t
        - ✓ jika $F(y), F(x_c) < F(x_t)$ maka buat cluster baru dengan pusat y dan jari-jari sama dengan jarak antara y dan x_t dan perbaharui *fungsi cluster*(input x_t)
        - ✓ selain itu, jadikan y sebagai titik pusat C
    - ➢ Ubah jari-jari C menjadi jarak antara y dan x_t
- Ulangi sampai k

Langkah 3: Mutasi DE

Langkah 4: Crossover DE

Langkah 5: Seleksi DE

Langkah 6: Uji masing-masing kandidat akar

- jika $\|x_i - x_j\| > \delta$ pilih x_i dan x_j sebagai akar
- $\|x_i - x_j\| \le \delta$, evaluasi $F(x_i)$ dan $F(x_j)$, pilih yang mempunyai nilai F lebih besar

2. Uji coba dan Hasil

Kemudian algoritma DE dengan teknik clustering yang telah dibuat diuji cobakan pada beberapa fungsi benchmark sebagai berikut.

<u>*Masalah 1*</u>

Masalah ini diambil dari Tsoulos [2] yang diformulasikan sebagai berikut.

$$g(x,y) = \begin{pmatrix} \cos(2x) - \cos(2y) - 0.4 \\ 2(y-x) + \sin(2y) - \sin(2x) - 1.2 \end{pmatrix} = 0$$

$$D(x,y) = -10 \le x \le 10, -10 \le y \le 10$$

Pada literature disebutkan bahwa banyaknya akar dari system persamaan ini tidak diketahui.

Dengan menggunakan parameter:

$$NP = 200, Cm = 0.5, Cr = 0.5, G\max = 200$$

$$m_{grouping} = 150, k_{grouping} = 100, \delta 1 = 0.2, \delta 2 = 1e-5, \varepsilon = 1e-4$$

dari masalah 1 di atas, ditemukan 13 akar berbeda dalam satu kali running seperti berikut.

Tabel 1: Akar-akar dari Masalah 1

No	*x*	*y*
1	-9.268258290324827	-8.931402203088391
2	9.581298243431895	9.918154762762995
3	-6.126665237496450	-5.789808932956341
4	6.439705376862722	6.776561681402832
5	6.963421107108507	8.543176048596331
6	3.821828453518714	5.401583395006538
7	-5.602949507250666	-4.023194565762841

8	-8.744542160840460	-7.164787219352634
9	-2.461356853660873	-0.881601912173048
10	3.298112723272929	3.634969027813038
11	0.156520069683136	0.493376374223245
12	0.680235799928920	2.259990741416745
13	-2.985072583906657	-2.648216279366548

Masalah 2

Masalah ini didapatkan dari Tsoulos [2], Kania [1], dan Sacco [3] yang dapat diformulasikan sebagai berikut.

$$g(x,y)=\begin{pmatrix} 0.5\sin(xy)-0.25\dfrac{y}{\pi}-0.5x \\ (1-\dfrac{0.25}{\pi})(e^{2x}-e)+e\dfrac{y}{\pi}-2ex \end{pmatrix}=0$$

$$D(x,y)=-1\le x\le 3,-17\le y\le 4$$

Dengan menggunakan parameter:

$$NP=200, Cm=0.5, Cr=0.5, G\max=200$$

$$m_{grouping}=200, k_{grouping}=100, \delta 1=0.1, \delta 2=1e-5, \varepsilon=1e-4$$

Dari masalah 2 di atas, ditemukan 12 akar yang berbeda dalam sekali running seperti berikut.

Tabel 2: Akar-akar dari Masalah 2

No	x	y
1	0.299448692490926	2.836927770458940
2	1.337425611989260	-4.140438646827949
3	1.433949329930748	-6.820765266341005
4	-0.260599290022477	0.622530896613911
5	1.294360459920630	-3.137219791192911
6	1.530505323720723	-10.202247948959245
7	1.481319568131123	-8.383612685619593
8	0.500000000000000	3.141592653589793
9	1.578225399213536	-12.176689850705655
10	1.654582718764350	-15.819188232171317
11	1.663421981330833	-16.282790650132462
12	1.604570546849489	-13.362901677998673

Masalah 3

Masalah ini didapatkan dari Kania [1] yang diformulasikan sebagai berikut.

$$g(x_1,x_2,x_3,x_4,x_5,x_6)=\begin{pmatrix} x_1+\dfrac{x_2{}^2x_4x_6}{4}+0.75 \\ x_2+0.405e^{1+x_1x_2}-1.405 \\ x_3-\dfrac{x_4x_6}{2}+1.5 \\ x_4-0.605e^{1-x_3{}^2}-0.395 \\ x_5-\dfrac{x_2x_6}{2}+1.5 \\ x_6-x_1x_5 \end{pmatrix}$$

$$-10\le x_i\le 10, i=1,2,\ldots,6$$

Dengan menggunakan parameter:

$$NP = 600, Cm = 0.5, Cr = 0.5, G\max = 300$$

$$m_{grouping} = 1000, k_{grouping} = 1500, \delta 1 = 0.1, \delta 2 = 1e-3, \varepsilon = 1e-3$$

Didapatkan dua akar yang berbeda dalam satu kali running seperti berikut:

Tabel 3: Akar-akar dari Masalah 3

	Posisi 1	Posisi 2
x_1	-0.998592652657473	-1.043074558956918
x_2	0.999381546189543	-0.550852402057143
x_3	-1.001525429403092	0.431740182809616
x_4	0.998137847225442	1.759892170861573
x_5	-1.000678104437737	-2.104639236859035
x_6	0.999298327408938	2.195292978339812

Masalah 4

$$g(x,y) = \begin{pmatrix} 0.1\sin^2 x + 0.1\sin^2 y + 0.3\sin x \sin y - 0.1\sin x + 0.1\sin y - 0.126466 \\ 0.1\sin^2 x - 0.1\sin^2 y + 0.2\sin x \sin y - 0.1\sin x - 0.1\sin y - 0.147451 \end{pmatrix}$$

$$D(x,y) = -5 \le x \le 8, -10 \le y \le 0$$

Dengan menggunakan parameter:

$$NP = 200, Cm = 0.5, Cr = 0.5, G\max = 200$$

$$m_{grouping} = 200, k_{grouping} = 100, \delta 1 = 0.1, \delta 2 = 1e-5, \varepsilon = 1e-10$$

Didapatkan 32 akar yang berbeda dalam satu kali running seperti berikut.

Tabel 4: Akar-akar dari Masalah 4

No	x	y
1	-2.467660808654514	-2.911338277741531
2	5.798209854717229	-8.795372663786758
3	5.609253462244308	-0.230254375848263
4	3.815524498525071	-9.194523584921114
5	3.626568106052148	-2.512187356607159
6	3.815524498525075	-6.513439683027846
7	-0.673931844935277	-9.194523584921114
8	-2.467660808654513	-6.513439683027848
9	-0.673931844935278	-2.911338277741529
10	3.815524509318402	-2.911338292117136
11	5.798209857532731	-0.629405303638844
12	5.798209854717232	-6.912590604162221
13	-0.484975452462356	-2.512187356607164
14	-2.467660812378100	-9.194523577972893
15	5.609253462244312	-6.513439683027857
16	-0.673931844830798	-6.513439683181455
17	-2.467660808653997	-0.230254375847343
18	-2.656617201127438	-6.912590604162219
19	5.609253462244306	-2.911338277741532
20	-0.484975491805072	-6.912590530945325

21	5.609253462244306	-9.194523584921120
22	5.798209854717232	-2.512187356607158
23	-2.656609591113897	-2.512208518622052
24	3.626568106052148	-0.629405296982633
25	-0.484975452444530	-8.795372663743020
26	3.626568106052148	-8.795372663786747
27	3.626568106052147	-6.912590604162221
28	3.815524498525071	-0.230254375848264
29	-0.673931844935224	-0.230254375848328
30	-0.484975454113733	-0.629405288529086
31	-2.656617201127438	-8.795372663786747
32	-2.656617201127439	-0.629405296982634

Kesimpulan

Kombinasi dari algoritma Differential Evolution dan teknik Clustering dapat digunakan sebagai alternative untuk menyelesaikan masalah pencarian akar-akar pada system persamaan tak linier. Algoritma ini bisa menghasilkan semua akar pada domain yang ditentukan dengan satu kali running.

Referensi

[1] Kania, Adhe, Sidharto, K.A, "Finding All Solutions of Nonlinear Equations Using Spiral Dynamics Inspired Optimization with Clustering", Journal of Advanced Computational Intelligence and Intelligent Informatics, 2015.

[2] Tsoulos, I.G., Stavrakoudis, Athanassios, "On Locating All Roots of Systems of Nonlinear Equations Inside Bounded Domain Using Global Optimization Method", Nonlinear Analysis: Real World Applications 11 (2010)2465-2471, 2010

[3] Sacco, W.F, Henderson N, "Finding All Solutions of Nonlinear systems Using Hybrid Metaheuristic with Fuzzy Clustering Means", Applied Soft Computing 11(2011)5424-5432, 2011

[4] Storne, R., Price, K., and Lampinen, J., "Differential Evolution: A Practical Approach to Global Optimization", Berlin: Springer-Velberg, 2004.