

PENGEMBANGAN METODE ALGORITMA GENETIKA DAN DARWINIAN PARTICLE SWARM OPTIMIZATION UNTUK FUNGSI MULTIMODAL

Ariadi Retno Tri Hayati¹

Abstrak

Metode optimasi telah banyak diaplikasikan pada pengenalan pola yang bertujuan untuk optimasi parameter secara otomatis. Metode optimasi yang umum diterapkan adalah algoritma genetika atau particle swarm optimization. Tujuan dari penelitian ini mengembangkan dari penggabungan kedua metode tersebut agar mendapatkan parameter yang lebih optimal secara otomatis jika dibandingkan optimasi dengan salah satu algoritma. Berdasarkan hasil penelitian, maka diperoleh kesalahan yang lebih sedikit atau minimum dibandingkan salah satu metode optimasi.

Kata-kata kunci: algoritma genetika, particle swarm optimization, multimodal function

Abstract

Optimization methods have been widely applied to pattern recognition which aims to optimize parameters automatically. Optimization method is commonly applied genetic algorithms or particle swarm optimization. The purpose of this study developed from the merger of the two methods in order to obtain a more optimal parameters automatically when compared with one of the optimization algorithms. Based on the results of the study, then the comparison of the number of errors of method Particle Swarm Optimization, Genetic Algorithm using the method in this study that by using the method in this study will result in fewer errors or minimum than one of method.

Keywords: genetic algorithm, particle swarm optimization, multimodal function

¹Ariadi Retno TH. Program Studi Manajemen Informatika, Jurusan Teknik Elektro, Politeknik Negeri Malang

1. PENDAHULUAN

Algoritma genetika telah banyak diterapkan untuk mencari nilai optimum secara otomatis dengan cara mendapatkan nilai berdasarkan operasi mutasi dan *crossover* sehingga mendapatkan perubahan nilai yang bervariasi, sedangkan metode *Particle Swarm Optimization* telah banyak digunakan untuk mencari nilai secara otomatis dengan konsep sosialisasi ikan dimana secara vector mencari suatu nilai dengan berarah.

Dari kedua konsep ini, kedua metode memiliki kelebihan dan dengan menggabungkan kedua metode ini diharapkan mendapatkan nilai yang variatif dari pengaruh metode Algoritma Genetika dan terarah dari metode *Particle Swarm Optimization* sehingga mendapatkan kesalahan seminimum mungkin jika dibandingkan menggunakan salah satu metode.

2. KAJIAN PUSTAKA

2.1 Algoritma Genetika

Algoritma genetika berusaha menangani permasalahan dengan menganalisa setiap solusi pada masing-masing populasi. Tahapan pertama adalah penentuan kode pada setiap kromosom sesuai dengan permasalahan yang akan diselesaikan. Berikutnya adalah persiapan pembelajaran dengan operator pada Algoritma Genetika. Operator reproduksi pada Algoritma Genetika dengan mutasi dan *crossover*.

Operator pada Algoritma Genetika

Operator pada Algoritma genetika yang digunakan adalah *crossover* dan mutasi. Sebagaimana yang telah dijelaskan sebelumnya, proses pembelajaran pada Algoritma genetika dimulai dengan *crossover* dilanjutkan dengan mutasi. Tetapi sebelum *crossover* perlu adanya proses seleksi yang bertujuan mendapatkan dua *parent* yang terbaik yang diharapkan menghasilkan generasi yang lebih baik dibandingkan kedua *parent*. Pada penelitian ini proses seleksi yang digunakan berdasarkan pengurutan data berdasarkan *fitness*, yaitu dari nilai *fitness* terbaik hingga terburuk.

Crossover

Setelah dilakukan pengurutan data, dilakukan proses *crossover* sebagaimana algoritma sebagaimana berikut :

$$\begin{aligned} x'_i &= \text{Uniform}(0,1)x_i + (1 - \text{Uniform}(0,1))x_{i+1} & i = 1,2,\dots,N-1 \\ x'_i &= \text{Uniform}(0,1)x_i + (1 - \text{Uniform}(0,1))x_1 & i = N \end{aligned} \quad (1)$$

Mutasi

Setelah dilakukan proses *crossover*, maka dilakukan proses mutasi dengan mengubah satu atau seluruh gen berdasarkan probabilitas random. Jika probabilitas random sebesar 100%, maka seluruh data dan seluruh gen pada data dilakukan proses mutasi. Jika probabilitas mutasi 0%, maka populasi berikutnya sama dengan populasi sebelumnya. Pada penelitian ini metode mutasi yang digunakan berdasarkan random gaussian, sebagaimana berikut:

$$x'_k = x_k + \text{normrand}(0,1, [\text{jumlah gen}])$$

3. Darwinian Particle Swarm Optimization (PSO)

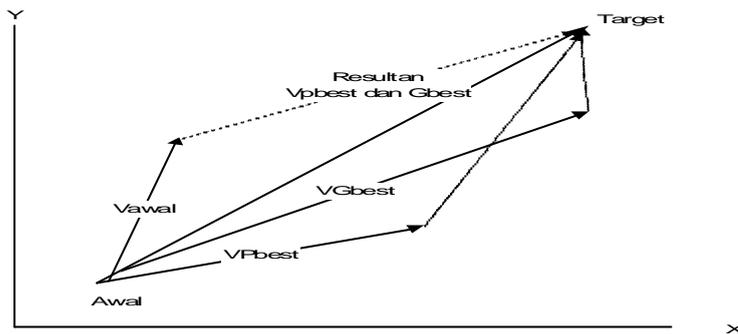
PSO adalah salah satu evolusi teknik komputasi untuk mendapatkan otomatisasi nilai variable yang lebih terarah sebagaimana konsep dasar *PSO* pada Gambar 2. Pada *PSO*, setiap particle menyimpan dua informasi penting yaitu:

- *Position vector* ($x_i(t)$)
- *Velocity vector* ($v_i(t)$)

Konsep dasar dari *PSO* yaitu mengembangkan simulasi dalam ruang dimensi XY (dua dimensi) berdasarkan informasi posisi dan kecepatan sebagaimana Gambar 2. Konsep dasar *PSO* adalah mengoptimisasi fungsi objektif tertentu. Setiap particle mengetahui nilai terbaiknya (P_{best}) dan posisinya (x). Selanjutnya, masing-masing particle mengetahui nilai terbaik didalam group (G_{best}). Masing-masing particle memodifikasi posisinya menggunakan kecepatan dan jarak dari P_{best} dan G_{best} . Modifikasi ini dapat diwakilkan dengan konsep velocity sebagaimana persamaan berikut:

$$V_{id}^{New} = w \times V_{id}^{old} + c_1 \times \text{rand} \times (p_{id} - x_{id}^{old}) + c_2 \times \text{rand} \times (p_{gd} - x_{id}^{old}). \quad (2)$$

Karakteristik dari metode *PSO* adalah adanya penyimpanan memory pada memory sebelumnya dan memory saat menganalisa saat ini. Hal ini berpengaruh dalam memperbarui nilai velocity dan nilai position. Penyimpanan memory ini untuk menentukan nilai yang terbaik yang digunakan (G_{best}) dan nilai P_{best} yang digunakan, untuk mendapatkan populasi baru.



Gambar 2 Konsep dasar PSO

Tujuan dari persamaan diatas adalah memperbaiki nilai velocity dengan nilai w adalah nilai *weight* yang mengontrol eksplorasi dari *search space*. Nilai $c1$ dan $c2$ adalah nilai konstan, pada umumnya kedua nilai ini bernilai 2. Nilai $rand$ adalah nilai random yang berkisar antara 0 dan 1. Variabel V_{id}^{old} adalah nilai velocity sebelumnya pada setiap bagian partikel (id), sedangkan nilai dari variabel V_{id}^{New} adalah nilai dari *velocity* yang baru dan akan digunakan pada iterasi berikutnya pada setiap bagian partikel (id). Setelah memperbaiki nilai velocity maka dilakukan memperbaiki nilai position sebagaimana persamaan berikut:

$$x_{id}^{New} = x_{id}^{old} + V_{id}^{New} \quad (3)$$

Nilai position dipengaruhi oleh nilai velocity dengan menambahkan nilai *velocity* yang telah terupdate dengan nilai position sebelumnya (x_{id}^{old}) sehingga didapatkan nilai position baru x_{id}^{New} yang akan digunakan sebagai *search point* pada iterasi berikutnya.

Pengembangan pada penelitian ini adalah menggunakan konsep Darwinian PSO dimana pada setiap iterasi akan mengubah dua menjadi dua nilai terbaik.

3. METODE

Dalam mencari *search point*, Algoritma Genetika memiliki karakteristik mencari pola baru dari seluruh kromosom sedangkan pada *Particle Swarm Optimization (PSO)*, dalam mencari *search point* berdasarkan *resultan* posisi *point* saat ini dengan nilai

fitness yang terbaik. Penggabungan kedua metode ini untuk menggabungkan karakteristik GA ataupun PSO. Terdapat pengembangan pada penelitian ini dengan pengurangan jumlah populasi yang sebelumnya 4M menjadi N sehingga lebih efektif pada komputasinya, dimana N/2 populasi pada Algoritma Genetika dan (N/2)+1 populasi pada Particle Swarm Optimization dalam penelitian ini dengan konsep Darwinian PSO (DPSO).

Berdasarkan gambar, maka algoritma dari penggabungan metode algoritma genetika dan PSO sebagaimana berikut :

- 1) Inisialisasi → random populasi sebanyak M data
- 2) Evaluasi dan ranking → evaluasi *fitness* pada setiap M individu, dimana fungsi *fitness* pada penelitian ini berdasarkan mean square error, sebagaimana persamaan berikut :

$$\text{MeanSquareError}(MSE) = \frac{1}{N} \sum_1^N (y - f(x))^2$$

Dimana y adalah nilai target yang diinginkan, $f(x)$ adalah hasil *actual output* dari pengklasifikasian data pembelajaran N .

- 3) Algoritma genetika dengan menggunakan real code operator GA, dari M/2 data terbaik.
 - 3.1. Data GA adalah dari M/2 data dengan *fitness* terbaik yaitu M/2 nilai *fitness* terkecil dari M.
 - 3.2. Crossover M/2 data dengan operasi sebagaimana berikut:

$$x'_i = \alpha * x_i + (1 - \alpha) x_{i-1} \quad i = 1, 2, \dots, (M / 2) - 1$$

$$x'_i = \alpha * x_i + (1 - \alpha) x_1 \quad i = M / 2,$$
 - 3.3. Mutasi data
- 4) PSO pada (M/2)+1 data
Update nilai velocities dan position
- 5) Jika belum memenuhi *stopping criteria*, maka data-data hasil pembelajaran PSO dan GA digabungkan dan dievaluasi dengan kembali pada langkah ke-2.

Cara penentuan P_{best} yang digunakan pada GAPSO, dengan cara membandingkan setiap data pada GA dan PSO berdasarkan urutannya. Nilai *fitness* antara kedua data ini yang digunakan sebagai P_{best} . Setelah mendapatkan P_{best} pada iterasi saat ini (t), maka dilanjutkan membandingkan P_{best} (t) dengan P_{best} ($t-1$)

berdasarkan urutan data pembelajaran PSO dan digunakan nilai yang terbaik.

Penentuan G_{best} yang merupakan nilai yang terbaik dari seluruh data pada setiap generasinya, ditentukan dengan cara membandingkan nilai fitness terbaik pada generasi saat ini dengan sebelumnya. Jika nilai G_{best} generasi sebelumnya lebih baik dibandingkan generasi saat ini, maka yang digunakan sebagai G_{best} adalah G_{best} generasi sebelumnya. Sebaliknya jika yang terbaik adalah nilai G_{best} saat ini, maka yang digunakan G_{best} adalah generasi saat ini.

Setelah pembelajaran dengan metode GA dilanjutkan dengan metode PSO, maka dilakukan analisa fitness dalam hal ini *Mean Square Error* untuk mengetahui error yang telah dicapai oleh pembelajaran GAPSO.

Sebelum memasuki optimasi pada PSO, maka perlu adanya konversi nilai dari kromosom dengan nilai fitness terbaik yang akan digunakan sebagai G_{best} sebagaimana persamaan berikut

$$x = r_b + (r_a - r_b) g,$$

dimana r_b adalah batas bawah dan r_a adalah batas atas dengan g adalah gen dari kromosom.



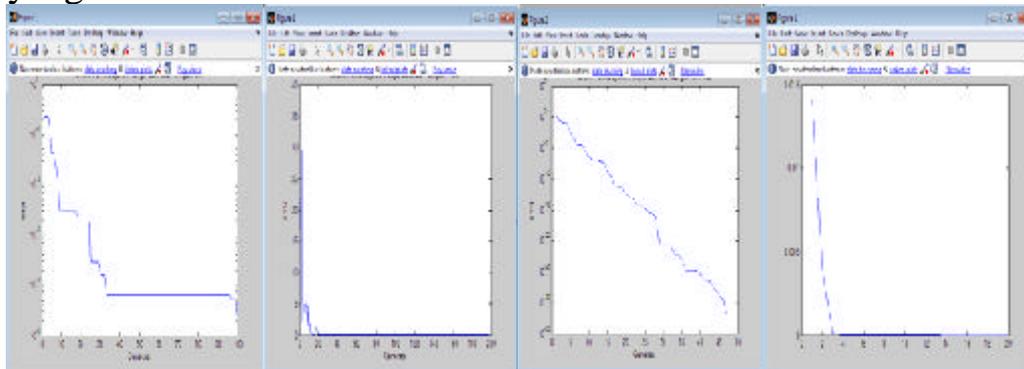
Gambar 3 Cara Penentuan P_{best} pada GAPSO

Pada penelitian ini untuk mempercepat komputasi, maka data inisialisasi yang digunakan sejumlah M data.

4. HASIL DAN PEMBAHASAN

Uji coba pada beberapa fungsi, dimana fungsi Rassenbrock akan ditampilkan beserta analisa iterasi untuk membandingkan

Sedangkan dengan metode GADPSO dapat menghasilkan kesalahan yang minimum diantara dibandingkan ke-3 metode yang lain.



a. GA b. PSO c. GAPSO d. DPSO

Gambar 5. Hasil uji coba fungsi Zakharov

Tabel 2. Analisa Kesalahan Iterasi Fungsi Zakharov dari ke-3 Metode

iter	Algoritma Gen	PSO	GAPSO	DGAPSO
1	0.2294	16.7549	1.9503	0.0140
2	0.2294	37.2976	0.3609	0.0039
3	0.2294	6.6525	0.1147	1.1287e-004
4	0.1360	0.6967	0.1055	1.1853e-005
5	0.0435	4.5899	0.0038	1.5872e-007
6	0.0435	4.9302	0.0003	1.1550e-008
7	0.0282	4.2017	2.8083e-005	1.5669e-009
8	0.0163	1.2408	2.7638e-005	5.5568e-011
9	0.0031	4.7482	4.8497e-006	2.8654e-014
10	0.0031	0.5326	4.1363e-007	2.8654e-014
11	0.0031	2.0412	1.0004e-007	4.0101e-015
12	0.0031	0.1941	9.8216e-008	2.0015e-016
13	0.0031	0.0246	8.4250e-008	1.9019e-017
14	0.0031	0.4537	4.5516e-008	1.0280e-018
15	0.0031	0.1154	1.3900e-009	1.0474e-019
16	0.0031	0.0172	2.4092e-010	7.5039e-021
17	0.0031	0.3612	7.9322e-012	4.4253e-022
18	0.0025	1.4970	5.6490e-012	2.2141e-023
19	0.0025	0.0827	4.1957e-012	2.2871e-025
20	0.0025	0.0192	5.4112e-013	1.1968e-025

Berdasarkan hasil uji coba dan analisa iterasi, maka hasil pembelajaran GADPSO memiliki tingkat kesalahan minimum dibandingkan salah satu metode GA ataupun PSO ataupun GAPSO.

5. PENUTUP

Berdasarkan hasil uji coba, metode pengembangan Algoritma Genetika dan Darwinian Particle Swarm Optimization pada penelitian ini menghasilkan kesalahan yang minimum dibandingkan metode GA, PSO, dan GAPSO. Dimana tujuan mencari kesalahan minimum pada penelitian ini adalah mencari nilai variabel secara otomatis dengan kesalahan minimum. Semakin sedikit kesalahan pada hasil uji coba maka nilai variabel lebih tepat.

5. DAFTAR PUSTAKA

- Bishop, CM. 2006. *“Pattern Classification and Machine Learning”*, Springer.
- Duda, R.O., Hart, P.E. dan Stork, D.G. 2001. *“Pattern Classification”*, Wiley, New York.
- Grodzevich, O. dan Wolkowicz, H. 2005. *“Regularization Using a Parameterized Trust Region Subproblem”*, The Natural Sciences and Engineering Research Council of Canada.
- Sivanandam, S.N. dan Deepa, S.N. 2008. *“Introduction to Genetic Algorithms”*, Springer-Verlag Berlin Heidelberg.
- Kao, Y.T. dan Zahara .2008. *“A hybrid genetic algorithm and particle swarm optimization for multimodal functions”*, Applied Soft Computing 8 849–857.
- Tillet, Jasson, Rao, Ferat Sahin. 2003. *“Darwinian Particle Swarm Optimization”*.
- Ariadi Retno Tri Hayati Ririd, Optimasi Metode Discriminatively Regularized Least Square Dengan Algoritma Genetika Dan Particle Swarm Optimization Untuk