

ANALISIS DAN IMPLEMENTASI BAT ALGORITHM UNTUK CONTINUOUS OPTIMIZATION TASK

Putri Aditya R¹, Suyanto^{2, 3}

¹Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

Bat Algorithm (BA) adalah algoritma metaheuristik yang tergolong baru, diciptakan oleh Xin-She Yang berdasarkan perilaku **echolocation** kelelawar. Sebagai algoritma yang dikembangkan dengan mengkombinasikan keuntungan dari Particle Swarm Optimization (PSO), genetic algorithm dan juga Harmony Search, Bat Algorithm dianggap lebih unggul dibandingkan algoritma-algoritma tersebut.

Pada Tugas Akhir ini, Bat Algorithm diimplementasikan untuk menyelesaikan Continuous Optimization Task, yang merupakan permasalahan optimasi fungsi kontinu. Permasalahan-permasalahan tersebut sering dipergunakan sebagai benchmark atas performansi dari algoritma metaheuristik, sehingga performansi BA dapat diteliti. Fungsi yang dipergunakan adalah Ackley, Griewank, Rastrigin dan Rosenbrock. Sebagai perbandingan performansi, hasil dari BA dibandingkan dengan hasil dari PSO.

Pengujian menunjukkan bahwa BA dapat diimplementasikan untuk Continuous Optimization Task dengan akurasi 100% dan BA lebih baik 1% dari PSO.

Kata Kunci : bat algorithm, continuous optimization task

Abstract

Bat Algorithm (BA) is a new metaheuristic algorithm, invented by Xin She Yang based from bat's echolocation behaviour as an algorithm that developed by combining the advantage of Particle Swarm Optimization (PSO), genetic algorithm and Harmony Search. Bat Algorithm is stated more superior than its predecessor.

In this Final Project, Bat Algorithm is implemented for resolving Continuous Optimization Task. These Tasks usually be used as benchmark function for the performance of metaheuristic algorithm, so BA's performance can be inspected. For performance comparison, BA's result is compared by PSO's result.

Testing finds out that BA can be implemented to Continuous Optimization Task with 100% accurate and BA is 1% better than PSO.

Keywords : bat algorithm, continuous optimization task

1. Pendahuluan

1.1 Latar belakang

Continuous Optimization Task adalah suatu permasalahan optimasi fungsi kontinu yang mempunyai batasan untuk mempersempit ruang pencarian. Beberapa contoh dari permasalahan ini adalah *Schwefel*, *Ackley*, *Rosenbrock*, *Shekel*, dan *Griewank*. Permasalahan-permasalahan tersebut sering dipergunakan sebagai *benchmark* atas performansi dari algoritma metaheuristik, karena memiliki dimensi yang luas serta lebih dari 100 variabel penentu, yang sering menjadi faktor mengurangnya efisiensi dari algoritma metaheuristik^[4]. Contoh algoritma metaheuristik adalah *Firefly Algorithm*, *Ant Colony Optimization*, *Particle Swarm Optimization*, dan *Bat Algorithm*.

Particle Swarm Optimization (PSO), adalah teknik optimasi berbasis populasi yang diciptakan oleh James Kennedy dan Russ Eberhart, terinspirasi dari tingkah laku sosial pada kawanan burung yang terbang berduyun-duyun atau ikan yang berenang berkelompok. Baik sekumpulan ikan maupun sekawanan burung, jika bergerak bersamaan dengan kecepatan tinggi tidak pernah terjadi kasus tabrakan, meskipun jarak antar masing-masing sangat dekat. Kecerdasan inilah yang diadopsi oleh Kennedy dan Eberhart.

PSO memiliki konsep yang sederhana dan mudah diimplementasikan dengan hanya beberapa baris kode, sehingga banyak yang mengambil keuntungan tersebut. Dibandingkan dengan algoritma evolusioner lain, keuntungan utama dari *PSO* adalah kehandalannya dalam mengontrol parameter dan efisiensi komputasinya yang tinggi.^[5]

Sedangkan *Bat Algorithm (BA)* adalah algoritma metaheuristik yang tergolong baru, diciptakan oleh Xin-She Yang berdasarkan perilaku *echolocation* kelelawar. Menurut beliau, kemampuan *echolocation* dari *microbat* sangat mengagumkan karena kelelawar-kelelawar tersebut dapat menemukan mangsa mereka dan dapat membedakan beberapa jenis serangga yang berbeda, bahkan dalam keadaan gelap total. Formulasi algoritma ini dibuat dengan mengidealkan perilaku *echolocation* dari kelelawar. Sebagai algoritma yang dikembangkan dengan mengkombinasikan keuntungan dari *PSO*, *genetic algorithm* dan juga *Harmony Search*, *Bat Algorithm* dianggap lebih unggul dibandingkan algoritma-algoritma tersebut.

Bat Algorithm diciptakan untuk menyelesaikan permasalahan optimasi fungsi kontinu, namun masih sedikit sekali riset yang menguji performansi *Bat Algorithm* dalam permasalahan tersebut jika dibandingkan dengan algoritma yang sudah ada sebelumnya.

1.2 Perumusan masalah

Permasalahan yang ada pada Tugas Akhir ini adalah:

1. Bagaimana mengimplementasikan *Bat Algorithm* untuk mengerjakan *Continuous Optimization Task*?

2. Apakah pengaruh dari perbedaan nilai parameter *input Bat Algorithm* pada *Continuous Optimization Task*?
3. Apakah jika dibandingkan dengan PSO, *output* dari sistem *Bat Algorithm* lebih optimal?

1.3 Batasan Masalah

Batasan masalah pada Tugas Akhir ini adalah :

1. Algoritma yang digunakan sebagai pembanding adalah algoritma *Particle Swarm Optimization* (PSO).
2. Permasalahan yang akan dipergunakan adalah permasalahan fungsi optimasi yang memiliki bidang permukaan nilai objektif yang termasuk dalam kelompok *Deceptive*, *Neutral*, *needle-in-a-haystack* dan/atau *Nightmare*.

1.4 Tujuan

Adapun tujuan dalam pengerjaan Tugas Akhir ini adalah :

1. Mengimplementasikan *Bat Algorithm* untuk mengerjakan *Continuous Optimization Task*,
2. Menganalisis pengaruh nilai parameter yang berbeda untuk parameter *input* terhadap hasil akhir program.
3. Menganalisis hasil *output BA* jika dibandingkan dengan *PSO*.

1.5 Hipotesa

Hipotesa :

Bat Algorithm dapat diterapkan pada masalah *Continuous Optimization Task* dengan performansi lebih baik dari algoritma *Particle Swarm Optimization*.

1.6 Metodologi Penyelesaian Masalah

Metodologi yang digunakan dalam menyelesaikan masalah-masalah di atas adalah sebagai berikut :

1. Studi Literatur
Mencari referensi-referensi yang relevan yang berkaitan dengan *Bat Algorithm* dan *Continuous Optimization Task*.
2. Pembangunan Aplikasi
 - a. Analisis dan Perancangan Sistem
Menganalisa dan merancang sistem *Bat Algorithm* dan *Continuous Optimization Task*.
 - b. Implementasi
Pembangunan sistem berdasarkan hasil perancangan sebelumnya. Implementasi dilakukan dengan *software* Matlab.
 - c. Pengujian dan Analisis Hasil Implementasi

Sistem diuji coba dengan input dan output dianalisis sehingga didapatkan kesimpulan mengenai perubahan nilai parameter input terhadap hasil output sistem.

3. Penyusunan laporan Tugas Akhir
Laporan dibuat berdasarkan metode yang dilakukan dan hasil yang didapat.

1.7 Sistematika Penulisan

Tugas Akhir ini disusun dalam beberapa bagian sebagai berikut :

BAB I PENDAHULUAN

Berisi latar belakang, perumusan masalah, tujuan, metodologi penyelesaian masalah dan sistematika penulisan.

BAB II DASAR TEORI

Bab ini membahas mengenai dasar teori yang mendukung penerapan algoritma yang digunakan untuk membangun sistem ini, yaitu *Continuous Optimization Task* dan *BA*.

BAB III ANALISIS DAN PERANCANGAN SISTEM

Berisi rincian mengenai perancangan sistem yang akan dibangun, penjelasan mengenai *BA*, serta implementasinya dalam memproses *Continuous Optimization Task*.

BAB IV ANALISIS DAN PENGUJIAN SISTEM

Berisi rincian mengenai pengujian yang dilakukan terhadap sistem yang dikembangkan, skenario pengujiannya, beserta analisis terhadap pengujian.

BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan yang diambil berkaitan dengan sistem yang dikembangkan, serta saran-saran untuk pengembangan lebih lanjut.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan hasil implementasi, pengujian dan analisis *BA* untuk *continuous optimization task*, maka diambil beberapa kesimpulan sebagai berikut :

1. *Bat Algorithm* bisa diimplementasikan untuk menyelesaikan *Continuous Optimization Task*.
2. Parameter populasi menentukan keluaran akhir dari sistem untuk fungsi *Ackley*, *Rastrigin*, dan *Rosenbrock*. Semakin banyak populasi, semakin baik hasil yang didapat karena kelelawar tersebar dalam area pengujian sehingga lebih memungkinkan area solusi tercakupi.
3. Parameter *loudness* berpengaruh untuk menentukan solusi sesuai dengan area pengujian fungsi tersebut. Perlu dilakukan *trial* untuk *range* nilai yang tepat untuk masing-masing fungsi, karena jika terlalu rendah ataupun tinggi, *BA* akan lebih mungkin terjebak dalam solusi lokal atau menghasilkan solusi yang tidak optimal.
4. Parameter *pulse* berpengaruh pada kondisi *random walk*. Karena tidak ada mekanisme pengecekan untuk solusi yang dibuat di sekitar *best* saat itu, ada kemungkinan *BA* membuat solusi baru disekitar solusi lokal sehingga hasil yang didapat tidak optimal. Sehingga semakin tinggi nilai batas atas *pulse*, semakin baik hasil yang didapat.
5. Generasi menentukan hasil keluaran untuk *Ackley* dan *Griewank*, karena *Ackley* memerlukan waktu yang lebih banyak untuk menghasilkan solusi yang lebih optimal dan rata-rata generasi yang diperlukan *Griewank* melebihi 100. Untuk *Rastrigin* dan *Rosenbrock*, batas 100 generasi sudah lebih dari cukup.
6. *BA* lebih baik dari *PSO* dalam hal mencari solusi optimal, dan generasi untuk fungsi *Griewank*, *Rastrigin*, dan *Rosenbrock*. Untuk *Ackley*, *PSO* lebih baik jika ingin mendapatkan hasil yang cukup mendekati dengan jumlah generasi yang tidak terlalu banyak. Dalam hal performansi, *BA* lebih baik 1% dari *PSO*.

5.2 Saran

Saran untuk pengembangan selanjutnya topik ini adalah :

1. Perlu diadakan uji coba untuk fungsi-fungsi lainnya sehingga efek dari parameter-parameter bisa lebih digeneralisir.
2. Perlu diujicobakan berbagai nilai parameter dengan kombinasi yang lebih banyak sehingga bisa didapat nilai parameter yang tepat untuk solusi yang lebih optimal.

Referensi

- [1] Yang, Xin-She. *A New Metaheuristic Bat-Inspired Algorithm*. Nature Inspired Cooperative Strategies for Optimization (2010).
- [2] Lukasik, Szymon ; Zak, Slawomir. *Firefly Algorithm for Continuous Constrained Optimization Task*. System Research Institute, Polish Academy of Science.
- [3] Suyanto. *Algoritma Optimasi: Deterministik atau Probabilistik*. Graha Ilmu (2010)
- [4] Tang, Ke; Li, Xiaodong dkk. *Benchmark Functions for the CEC'2010 Special Session and Competition on Large-Scale Global Optimization*. (2009).
- [5] Chen, Po-Hung. *Particle Swarm Optimization fo Power Dispatch with Pumped Hydro*. Particle Swarm Optimization.
- [6] Weise, Thomas. *Global Optimization Algorithms – Theory and Application*. E-book (2009).
- [7] Molga, Marcin; Smutnicki, Czeslaw. *Test Functions for Optimization Needs*. (2005)
- [8] Suyanto. *Evolutionary Computation : Komputasi berbasis Evolusi dan Genetika*. Penerbit Informatika (2008)
- [9] Suganthan, P,N; Hansen, N; Liang, J,J dkk. *Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization*. Technical Report, Nanyang Technological University and KanGAL Report Number 2005005, Kanpur Genetic Algorithms Laboratory (2005)
- [10] Yang, Xin-She. *Nature-Inspired Metaheuristic Algorithms*. Luniver Press (2008)
- [11] Pohlheim, Hartmut. *GEATbx Examples : Example of Objective Functions*. E-book (2006).
- [12] Hedar, Abdel-Rahman. *Global Optimization Methods and Codes*.