

## Optimalisasi *Execution Time Firefly Algorithm*

Dedy Abdianto Nggego

Prodi Elektronika, Akademi Teknologi Industri Dewantara Palopo,  
Jalan K.H. Ahmad Razak 2 No. 7, Kota Palopo, Indonesia

\*Email : [dedyabdianto@atidewantara.ac.id](mailto:dedyabdianto@atidewantara.ac.id)

---

### Abstrak

Penelitian ini bertujuan untuk mencari nilai parameter optimal algoritma kunang-kunang dilihat dari waktu eksekusi yang dibutuhkan dalam menemukan solusi optimal pada kasus *n-queens problem*. Metode yang digunakan yaitu metode eksperimen dengan melakukan 10 kali percobaan pada setiap parameter dengan populasi 15 dan 50 kunang-kunang pada dimensi 10x10. Parameter optimal yang diperoleh yaitu  $\alpha = 2.0$ ,  $\beta = 1.0$ , dan  $\gamma = 0.2$  dengan rata-rata waktu eksekusi 0.63776 detik pada populasi 15 kunang-kunang dan rata-rata waktu eksekusi 0.98321 detik pada populasi 50 kunang-kunang.

**Kata Kunci :** *execution time, algoritma kunang-kunang, optimalisasi*

---

### 1. Latar Belakang

Dalam dunia komputasi dikenal sebuah istilah masalah komputasi yaitu NP-Complete/NP-Hard. Umumnya masalah ini memiliki kompleksitas algoritma dengan waktu eksponensial seiring dengan ukuran input [1]. Untuk menyelesaikan masalah-masalah seperti NP-Hard, dibutuhkan sebuah metode optimasi, salah satunya metode metaheuristik. Menurut [2] metaheuristik merupakan sebuah kerangka kerja algoritma yang dirancang untuk menyelesaikan masalah optimasi yang kompleks. Untuk masalah optimasi, metode metaheuristik moderen cukup efisien untuk mencapai dan menemukan solusi global [3]. Salah satu algoritma yang termasuk dalam metode heuristik adalah firefly algorithm (FA) atau algoritma kunang-kunang [4]. Penelitian [5] mengusulkan algoritma kunang-kunang global untuk memecahkan kasus randomized time-varying knapsack problems dan berhasil menemukan performa yang unggul dibandingkan dengan beberapa algoritma yang jadi

pembanding. Kasus lain yang dipecahkan menggunakan pendekatan algoritma kunang-kunang adalah production-distribution network (PDN). [6] menggunakan selective firefly algorithm (SFA) untuk menyelesaikan masalah berukuran besar yang kompleks seperti PDN. Dalam metode SFA yang diusulkan, setiap kunang-kunang mengidentifikasi semua kunang-kunang yang mempunyai tingkat kecerahan yang lebih, lalu mengevaluasi perubahan kecerahannya sebelum bergerak atau berpindah ke posisi baru.

[7] mengatakan bahwa metaheuristik cukup sesuai untuk melakukan pekerjaan-pekerjaan optimasi. N-Queen Problem merupakan salah satu kasus yang termasuk dalam NP-Hard Problem yang diselesaikan menggunakan metode metaheuristik [8]. [9] memecahkan masalah n-queens problem menggunakan metode Distributed P System (dP System) dengan active membranes dengan komponennya adalah P dan membran aktif. Metode

tersebut dapat mengurangi hingga sepertiga sampai setengah waktu komputasi.

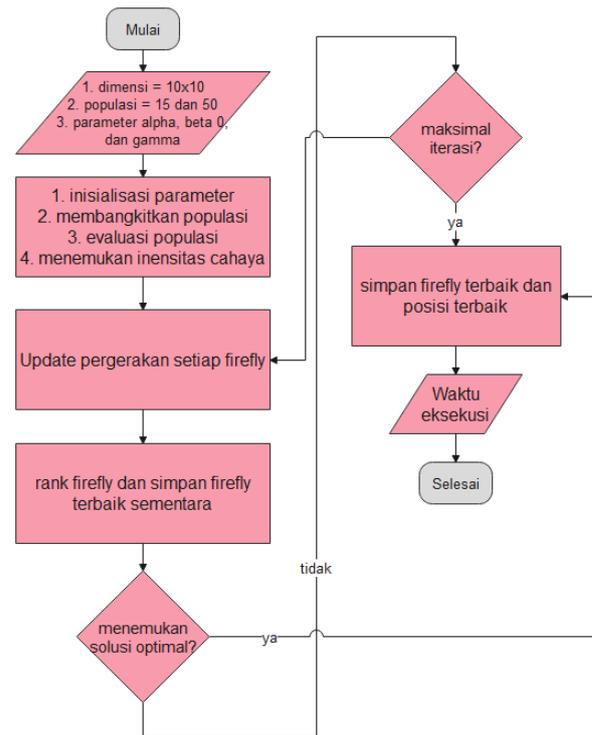
Secara umum n-queens problem merupakan sebuah permainan puzzle, yaitu masalah bagaimana meletakkan bidak ratu sebanyak n pada papan catur berukuran nxn sehingga tidak ada bidak ratu yang saling menangkap/memakan hanya dengan satu langkah. Penelitian [10] dan [11] menggunakan beberapa algoritma metaheuristik dalam melakukan perhitungan cost dan execution time pada n-queens problem. [12] melakukan serangkaian eksperimen untuk menguji worst-case execution time menggunakan teknik yang mereka usulkan yaitu evolutionary symbolic execution, dengan menghitung execution cost berdasarkan fitness function terburuk.

Penelitian ini berfokus pada pencarian nilai parameter optimum  $\alpha$ ,  $\beta_0$ , dan  $\gamma$  untuk menemukan waktu eksekusi yang paling cepat dalam menemukan solusi optimal pada kasus n-queens problem.

## 2. Metodologi

Penelitian ini merupakan penelitian eksperimen dalam menemukan solusi optimal dengan menggunakan populasi sebanyak 15 dan 50 kunang-kunang pada dimensi 10x10. Data yang digunakan adalah data rata-rata dan standar deviasi waktu eksekusi FA dalam menemukan solusi optimal. Data hasil penelitian diperoleh dari 10 kali percobaan *running algorithm* pada masing-masing parameter.

*Flowchart firefly algorithm* dapat dilihat pada gambar 1.



Gambar 1. *Flowchart FA*

## 3. Hasil dan Pembahasan

Pada penelitian ini dilakukan masing-masing 10 kali percobaan pada setiap parameter yang berbeda-beda untuk mengetahui rata-rata waktu eksekusi dari FA dalam menemukan solusi optimal. Pada kasus ini, populasi yang digunakan yaitu sebanyak 15 dan 50 kunang-kunang dengan dimensi yaitu 10x10. Pada beberapa kasus, seperti [13] dan [14] digunakan populasi antara 15-50 kunang-kunang. Dalam FA, semua kunang-kunang adalah unisex, artinya daya tarik kunang-kunang bergantung pada kecerahan cahaya. Kunang-kunang yang kurang cerah/terang akan terbang menuju kunang-kunang yang lebih cerah. Menurut [15][16] tingkat kecerahan kunang-kunang ditentukan oleh fungsi dari masalah yang diberikan. Untuk masalah maksimisasi, intensitas cahaya sebanding dengan dengan nilai fungsi tujuan. *Pseudocode FA* dapat dilihat pada tabel 1.

Tabel 1. *Pseudocode FA*

```

Procedure FA
Begin
  input data;
  inisialisasi parameter;
  bangkitkan populasi awal firefly;
  hitung fungsi fitness;
  hitung intensitas cahaya setiap firefly;
  while (t < maksimal_iter)
    For i <- 1 to banyak firefly
      For j <- 1 to banyak firefly
        If (I(xi) < I(xj))
          Movement;
          Hitung fungsi tujuan;
          Update intensitas cahaya;
        End If
      End For j
    End For i
    Tentukan firefly terbaik sementara;
  End While
  Tentukan firefly terbaik dan posisi terbaik;
End

```

Persamaan yang dirumuskan oleh [17] pada perilaku pergerakan kunang-kunang sebagai berikut.

$$\beta = \beta_0 e^{-\gamma r^2} \quad (1)$$

Dimana  $\beta$  adalah daya tarik pada  $r = 0$  saat tidak ada jarak antar kunang-kunang.  $\gamma$  adalah koefisien penyerapan udara.

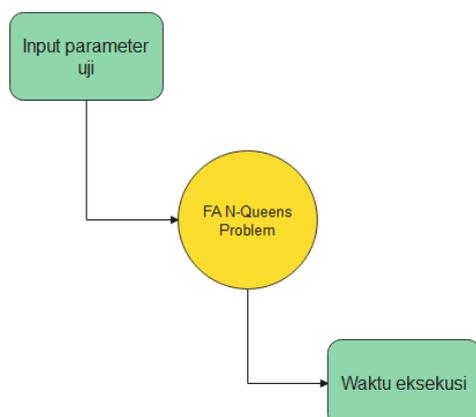
$$r_{ij} = |x_i - x_j| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (2)$$

$r_{ij}$  adalah jarak antara kedua kunang-kunang  $i$  dan  $j$ .

$$x_i = x_i + \beta_{ij}(x_j - x_i) + \alpha \left( \text{rand} - \frac{1}{2} \right) \quad (3)$$

Dimana  $\alpha$  adalah koefisien parameter acak dan  $\text{rand}$  adalah bilangan riil acak pada interval 0-1.

Skenario pengujian parameter dapat dilihat pada gambar 2.



Gambar 2. Skenario pengujian parameter

Adapun parameter yang diujikan adalah nilai  $\alpha$ ,  $\beta_0$ , dan  $\gamma$  dengan diberikan masukan nilai yang berbeda-beda. Pada beberapa penelitian digunakan parameter  $\beta_0$ ,  $\alpha$ , dan  $\gamma$  yang berbeda-beda, disesuaikan dengan kasus yang dipecahkan. Pada penelitian [13] digunakan masing-masing 1, 0.2, dan 1 untuk parameter  $\beta_0$ ,  $\alpha$ , dan  $\gamma$ , sedangkan pada penelitian [18] digunakan masing-masing 1, 2, dan 1. Penelitian [19] digunakan nilai  $\alpha$ ,  $\beta_0$ , dan  $\gamma$  yaitu 0.5, 1.0, dan 1.0. Nilai parameter yang diujikan disajikan pada tabel 2.

Tabel 2. Parameter uji coba

No	$\alpha$	$\beta_0$	$\gamma$	Populasi
1	0.2 - 3	0.1 - 2	0.1 - 3	15 dan 50

Hasil eksperimen yang dilakukan masing-masing 10 kali menghasilkan nilai rata-rata dan standar deviasi dari waktu eksekusi algoritma. Pada tabel 3 disajikan seluruh hasil eksperimen. Waktu eksekusi terbaik yang diperoleh ditandai dengan huruf tebal.

Tabel 3. Waktu eksekusi FA dengan populasi 15

Parameter	Mean	Std	Fitness
$\beta_0 = 0.1$			
$\alpha = 2.0$	4,83859	2,73271	2
$\gamma = 1.0$			
$\beta_0 = 0.2$			
$\alpha = 2.0$	2,51794	1,50617	1
$\gamma = 1.0$			
$\beta_0 = 0.5$			
$\alpha = 2.0$	<b>1,74668</b>	<b>1,30109</b>	<b>0</b>
$\gamma = 1.0$			
$\beta_0 = 1$			
$\alpha = 1.5$	2,04024	1,55782	3
$\gamma = 1.0$			
$\beta_0 = 1$			
$\alpha = 1.8$	<b>0,33997</b>	<b>0,21511</b>	<b>1</b>
$\gamma = 1.0$			
$\beta_0 = 1$			
$\alpha = 2.0$	0,48078	0,32294	1
$\gamma = 1.0$			
$\beta_0 = 1$	1,04198	0,87358	1

Parameter	Mean	Std	Fitness
$\alpha = 2.0$ $\gamma = 0.1$			
$\beta_0 = 1$ $\alpha = 2.0$ $\gamma = 0.2$	<b>0,63776</b>	<b>0,45522</b>	<b>0</b>
$\beta_0 = 1$ $\alpha = 2.0$ $\gamma = 0.3$	0,72233	0,42485	1
$\beta_0 = 1$ $\alpha = 2.0$ $\gamma = 0.4$	1,08921	0,87457	2
$\beta_0 = 1$ $\alpha = 2.0$ $\gamma = 0.5$	<b>0,67231</b>	<b>0,71064</b>	<b>1</b>
$\beta_0 = 1$ $\alpha = 2.0$ $\gamma = 0.6$	<b>1,26841</b>	<b>0,78053</b>	<b>0</b>
$\beta_0 = 1$ $\alpha = 2.0$ $\gamma = 0.7$	1,88326	1,46626	1
$\beta_0 = 1$ $\alpha = 2.0$ $\gamma = 0.8$	2,58474	2,21004	3
$\beta_0 = 1$ $\alpha = 2.0$ $\gamma = 0.9$	1,81584	1,16677	4
$\beta_0 = 1$ $\alpha = 2.0$ $\gamma = 1.0$	1,90989	1,91724	2
$\beta_0 = 1$ $\alpha = 2.0$ $\gamma = 1.5$	2,21996	1,59533	1
$\beta_0 = 1$ $\alpha = 2.0$ $\gamma = 2.0$	1,27423	2,13117	2
$\beta_0 = 1$ $\alpha = 2.0$ $\gamma = 2.5$	0,79374	0,65636	2
$\beta_0 = 1$ $\alpha = 2.0$ $\gamma = 3.0$	0,79911	0,73273	2
$\beta_0 = 1$ $\alpha = 3.0$ $\gamma = 1.0$	0,81951	0,56794	2

Seperti yang disajikan pada tabel 3, dapat dilihat bahwa rata-rata waktu eksekusi paling cepat adalah 0,33997 detik

dengan standar deviasi 0,21511. Waktu tersebut sudah cukup baik, namun pada kasus ini, salah satu hasil yang penting untuk diperhatikan yaitu nilai fitness. Semakin kecil nilai fitness, maka solusi optimal yang diperoleh semakin baik. Dapat dilihat bahwa nilai fitness pada waktu eksekusi tercepat tersebut adalah 1, yang artinya masih terdapat 1 posisi pada papan catur yang solusinya tidak ditemukan. Parameter dengan waktu eksekusi tercepat dan fitness terbaik pada percobaan ini yaitu  $\beta_0 = 1$ ,  $\alpha = 2$ , dan  $\gamma = 0.2$  dengan waktu eksekusi 0.63776 detik dan standar deviasi 0.45522. Selanjutnya dilakukan uji coba dengan menambahkan jumlah populasi yaitu 50 kunang-kunang terhadap beberapa parameter dengan waktu eksekusi terbaik yang didapatkan pada populasi 15. Hasil yang diperoleh disajikan pada tabel 4.

**Tabel 4.** Waktu eksekusi FA dengan populasi 50

Parameter	Mean	Std	Fitness
$\beta_0 = 0.5$ $\alpha = 2$ $\gamma = 1.0$	1,05368	0,75766	0
$\beta_0 = 1$ $\alpha = 1.8$ $\gamma = 1.0$	2,86526	4,51075	0
$\beta_0 = 1$ $\alpha = 2.0$ $\gamma = 0.2$	<b>0,98321</b>	<b>1,47955</b>	<b>0</b>
$\beta_0 = 1$ $\alpha = 2.0$ $\gamma = 0.5$	1,34592	1,38376	0

Waktu eksekusi tercepat pada populasi 50 kunang-kunang didapatkan pada parameter  $\alpha$ ,  $\beta_0$ , dan  $\gamma$  dengan nilai masing-masing 2.0, 1, dan 0.5 dengan waktu eksekusi 0.98321 detik dan standar deviasi 1.47955. Penambahan populasi dari 15 menjadi 50 berdampak baik pada nilai fitness untuk semua kombinasi parameter yaitu dengan nilai 0. Pada populasi 50 kunang-kunang, waktu eksekusi yang didapatkan lebih lambat

dibanding dengan populasi 15 kunang-kunang. Hal ini dikarenakan terjadinya evaluasi setiap kunang-kunang pada masing-masing posisi dalam setiap iterasi.

#### 4. Kesimpulan

Waktu eksekusi dan penemuan solusi optimal pada kasus ini dipengaruhi oleh penentuan inialisasi parameter  $\alpha$ ,  $\beta_0$ , dan  $\gamma$ . Parameter optimal pada populasi 15 dan 50 kunang-kunang pada kasus ini yaitu  $\alpha = 2.0$ ,  $\beta_0 = 1.0$ , dan  $\gamma = 0.2$ . Semakin besar populasi kunang-kunang, semakin besar juga peluang untuk memperoleh solusi optimal dengan nilai fitness 0, namun hal ini berdampak pada kecepatan waktu eksekusi algoritma.

#### Daftar Pustaka

- [1] Y. Huang, "Computing quantum discord is NP-complete Computing quantum discord is NP-complete," 2014, doi: 10.1088/1367-2630/16/3/033027.
- [2] L. Bianchi, M. Dorigo, L. M. Gambardella, and W. J. Gutjahr, "A survey on metaheuristics for stochastic combinatorial optimization," *Nat. Comput.*, vol. 8, no. 2, pp. 239–287, 2009, doi: 10.1007/s11047-008-9098-4.
- [3] V. N. Rajput, K. S. Pandya, and K. Joshi, "Optimal coordination of Directional Overcurrent Relays using hybrid CSA-FFA method," *ECTI-CON 2015 - 2015 12th Int. Conf. Electr. Eng. Comput. Telecommun. Inf. Technol.*, 2015, doi: 10.1109/ECTICon.2015.7207044.
- [4] E.-G. Talbi, *METAHEURISTICS - From Design To Implementation*, vol. 66. 2012.
- [5] Y. Feng, G. G. Wang, and L. Wang, "Solving randomized time-varying knapsack problems by a novel global firefly algorithm," *Eng. Comput.*, vol. 34, no. 3, pp. 621–635, 2018, doi: 10.1007/s00366-017-0562-6.
- [6] S. Khalifehzadeh and M. B. Fakhrzad, "A modified firefly algorithm for optimizing a multi stage supply chain network with stochastic demand and fuzzy production capacity," *Comput. Ind. Eng.*, vol. 133, no. May, pp. 42–56, 2019, doi: 10.1016/j.cie.2019.04.048.
- [7] S. Desale, A. Rasool, S. Andhale, and P. Rane, "Heuristic and Meta-Heuristic Algorithms and Their Relevance to the Real World: A Survey," *Int. J. Comput. Eng. Res. Trends*, vol. 351, no. 5, pp. 2349–7084, 2015, [Online]. Available: <http://www.ijcert.org>.
- [8] S. Sathyapriya, R. Stephen, and V. S. J. Irudayaraj, "Survey on N-Queen Problem with Genetic Algorithm," no. 2, pp. 54–59, 2018.
- [9] K. C. Buño, F. G. C. Cabarle, M. D. Calabia, and H. N. Adorna, "Solving the N-Queens problem using dP systems with active membranes," *Theor. Comput. Sci.*, vol. 736, pp. 1–14, 2018, doi: 10.1016/j.tcs.2017.12.013.
- [10] A. Habiboghli and T. Jalali, "A Solution to the N-Queens Problem Using Biogeography-Based Optimization," *Int. J. Interact. Multimed. Artif. Intell.*, vol. 4, no. 4, p. 20, 2017, doi: 10.9781/ijimai.2017.443.
- [11] S. Mukherjee, S. Datta, P. B. Chanda, and P. Pathak, "Comparative Study of Different Algorithms to Solve N Queens Problem," *Int. J. Found. Comput. Sci. Technol.*, vol. 5, no. 2, pp. 15–27, 2015, doi: 10.5121/ijfcst.2015.5202.
- [12] A. Aquino, G. Denaro, and P. Salza, "Worst-Case Execution Time Testing via Evolutionary Symbolic Execution," *Proc. - Int. Symp. Softw. Reliab. Eng. ISSRE*, vol. 2018-October, pp. 76–87, 2018, doi: 10.1007/s00366-017-0562-6.

- 10.1109/ISSRE.2018.00019.
- [13] X. Qi, S. Zhu, and H. Zhang, "A hybrid firefly algorithm," *Proc. 2017 IEEE 2nd Adv. Inf. Technol. Electron. Autom. Control Conf. IAEAC 2017*, pp. 287–291, 2017, doi: 10.1109/IAEAC.2017.8054023.
- [14] X. S. Yang, "Firefly algorithms for multimodal optimization," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5792 LNCS, pp. 169–178, 2009, doi: 10.1007/978-3-642-04944-6\_14.
- [15] M. A. Bramer and R. Ellis, "Programme chairs' introduction," *Res. Dev. Intell. Syst. XXVI Inc. Appl. Innov. Intell. Syst. XVII*, 2010, doi: 10.1007/978-1-84882-983-1.
- [16] X. S. Yang, "Firefly algorithm, stochastic test functions and design optimization," *Int. J. Bio-Inspired Comput.*, vol. 2, no. 2, pp. 78–84, 2010, doi: 10.1504/IJBIC.2010.032124.
- [17] X. Yang, "Stochastic Algorithms: Foundations and Applications," vol. 5792, no. May, 2009, doi: 10.1007/978-3-642-04944-6.
- [18] D. A. Nggego, A. Setyanto, and Sukoco, "Implementasi Algoritma Firefly pada Kasus N-Queens Problem," vol. 1, no. 10, pp. 219–227, 2020.
- [19] H. Wang *et al.*, "Firefly algorithm with neighborhood attraction," *Inf. Sci. (Ny.)*, vol. 382–383, pp. 374–387, Mar. 2017, doi: 10.1016/j.ins.2016.12.024.