

## PENERAPAN ALGORITMA CUCKOO SEARCH PADA TRAVELLING SALESMAN PROBLEM

Hardy <sup>1)</sup>, Ng Poi Wong <sup>2)</sup>, Dedy Suwandi <sup>3)</sup>

<sup>1,2,3</sup> Teknik Informatika, STMIK Mikroskil

Jl Thamin No. 140, Medan, 20212

Telp : (061) 4573767, Fax : (061) 4567789

E-mail : [hardy@mikroskil.ac.id](mailto:hardy@mikroskil.ac.id)<sup>1)</sup>, [poiwong@mikroskil.ac.id](mailto:poiwong@mikroskil.ac.id)<sup>2)</sup>, [081111317@students.mikroskil.ac.id](mailto:081111317@students.mikroskil.ac.id)<sup>3)</sup>

---

### Abstrak

Permasalahan NP-Complete (Nondeterministic Polynomial-Complete) merupakan masalah yang tidak bisa diselesaikan dengan algoritma konvensional. Hal ini disebabkan karena variasi parameter dan ruang pencarian yang sangat besar. Untuk menyelesaikannya diperlukan algoritma heuristik seperti contohnya algoritma Cuckoo Search. Algoritma cuckoo search telah menunjukkan performansi pencarian yang lebih bagus dari algoritma heuristik lain seperti algoritma Particle Swarm Optimization (PSO) maupun algoritma genetika. Keunggulan dari Cuckoo Search terletak pada random stepnya yang menggunakan Lévy Flight yang memungkinkan pencarian nilai optimal dalam waktu yang lebih singkat.

Penelitian ini meninjau performansi algoritma Cuckoo Search dengan menggunakan masalah NP-Complete klasik yaitu Travelling Salesman Problem. Hasil penelitian akan dibandingkan dengan algoritma genetika. Hasil pengujian menunjukkan algoritma Cuckoo Search memiliki efisiensi yang lebih besar bila dibandingkan dengan algoritma genetika. Dari pencarian hasil optimal 180 kota, algoritma Genetika memerlukan waktu kurang lebih 43 menit sedangkan algoritma Cuckoo Search hanya memerlukan waktu 53 detik.

**Kata kunci:** travelling salesman problem, cuckoo search, Lévy Flights, Genetic Algorithm

### Abstract

NP-Complete problem is a problem which can't be solved by using conventional algorithm. This is due to the numerous parameters and huge search space. To solve it, heuristic algorithm like Cuckoo Search algorithm is needed. Cuckoo Search has shown better performance than other heuristic algorithms such as Particle Swarm Optimization and Genetics Algorithm. Its advantageous point comes in its random step which uses Lévy Flights. Using Lévy Flights one can find optimum value in a shorter time.

This research review Cuckoo Search algorithm on NP-Complete classic problem, The Travelling Salesman Problem. The results will be compared with Genetic Algorithm results run on the same problem. Through the experiment, shows that Cuckoo Search algorithm have higher efficiency compared to Genetic Algorithm. The optimum value finding of 180 cities, Genetic Algorithm takes approximately 43 minutes while Cuckoo Search algorithm only takes 43 seconds.

### 1. PENDAHULUAN

Traveling Salesman Problem (TSP) merupakan permasalahan matematika klasik tentang seorang penjual yang melakukan tour tertutup terhadap kota-kota yang telah didaftarkan dalam jadwalnya untuk dikunjungi, dengan syarat tiap kota hanya dapat dikunjungi sekali dan total jarak perjalanan yang ditempuh harus seminimum mungkin. TSP memiliki cabang aplikasi yang luas dalam kehidupan sehari-hari, mencakup masalah jaringan komputer, transportasi maupun robotik. Banyak algoritma yang digunakan untuk menyelesaikan TSP mengalami penurunan performansi dalam proses pencarian ketika jumlah kota diperbesar, contohnya antara lain adalah algoritma genetika (Goldberg dan Holland, 1988) dan algoritma Particle Swarm Optimisation (PSO) (Eberhart dan Kennedy, 1995). Dalam hal ini, ukuran dari performansi yang dimaksud adalah waktu yang dibutuhkan untuk menyelesaikan pencarian dan hasil akhir yang didapatkan.

Cuckoo Search merupakan algoritma heuristik baru yang berdasarkan hasil studi awalnya dengan menggunakan dua belas fungsi stokastik berbeda menunjukkan performansi pencarian yang lebih bagus dari algoritma heuristik lain seperti algoritma PSO maupun Algoritma Genetik (Yang, 2010). Keunggulan dari Cuckoo Search ini sendiri terletak pada random stepnya yang menggunakan Lévy Flights (konsep pencarian makanan secara random pada hewan ketika sedang terbang), yang memungkinkan pencarian nilai optimal dalam waktu yang

lebih singkat (Yang, 2010). Cuckoo Search sebagai algoritma heuristik modern akan diterapkan ke TSP, dengan tujuan untuk mengukur performa Cuckoo Search tersebut dalam menyelesaikan masalah TSP.

## 2. TINJAUAN PUSTAKA

Cuckoo Search (Yang dan Deb, 2009) merupakan algoritma heuristik modern yang didasarkan pada perilaku burung cuckoo. Burung cuckoo memiliki strategi reproduksi yang unik, dimana mereka akan menaruh telur mereka di sarang burung jenis lain. Setiap burung cuckoo hanya menaruh satu butir telurnya di sarang burung lain yang dipilih secara acak. Sarang burung yang menghasilkan generasi cuckoo terbaik akan melanjutkan proses ke generasi berikutnya. Setiap pergantian generasi, jumlah dari pemilik sarang burung asli akan diatur kembali dan pemilik asli sarang ini mempunyai peluang untuk mengenali telur burung cuckoo yang ditaruh di sarangnya dengan probabilitas 0 sampai 1 ( $P_a$ ). Dalam hal ini, bila pemilik asli sarang menemukan telur burung cuckoo, maka pemilik itu dapat memilih untuk membuang telur tersebut ataupun meninggalkan sarangnya.

Ketika sedang menghasilkan generasi baru  $x^{(t+1)}$ , maka akan digunakan proses pengacakan langkah dengan Lévy Flights yang dapat dilihat sebagai berikut :

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus \text{Lévy}(\lambda) \quad (1)$$

di mana  $\alpha > 0$  merupakan ukuran langkah yang dikaitkan dengan tingkatan masalah yang dikerjakan. Sedangkan  $\text{Lévy}(\lambda)$  menyatakan fungsi persamaan posisi dari Lévy Flights, yang bentuk persamaannya adalah sebagai berikut.

$$\text{Lévy} \sim u = t^{-\lambda}, (1 < \lambda \leq 3) \quad (2)$$

Algoritma dari Cuckoo Search bisa dilihat di Algoritma 1.

Berdasarkan hasil percobaan perbandingan yang dilakukan antara algoritma Cuckoo Search, PSO, dan Genetika, dengan menggunakan beberapa fungsi pengetesan berbeda dan uji coba sebanyak 100 kali, didapatkan hasil analisis statistika yang mengejutkan.

---

### Algoritma 1 Algoritma Cuckoo Search (Yang dan Deb, 2009).

---

- 1: Fungsi Objektif  $f(x)$ ,  $x = (x_1, \dots, x_d)^T$ ;
  - 2: Inisialisasi populasi dari  $n$  sarang burung target  $X_i$  ( $i = 1, 2, \dots, n$ );
  - 3: **While** ( $t < \text{generasiTotal}$ ) atau (kriteria lain untuk berhenti);
  - 4: Evaluasi nilai kualitas dari masing-masing burung cuckoo
  - 5: Pilih dari burung cuckoo secara acak dan lakukan random walk
  - 6: **Jika** ( $F_i > F_j$ )
  - 7: Gantikan burung cuckoo  $j$  dengan burung cuckoo  $i$
  - 8: **End If**
  - 9: Reset ulang sarang-sarang dengan kondisi terburuk ( $P_a$ )
  - 10: Simpan sarang-sarang yang berhasil lolos
  - 11: Urutkan solusi dan cari yang terbaik
  - 12: **End While**
- 

Di Tabel 1 diperlihatkan hasil analisis statistika, dengan format penulisan rata-rata nilai evaluasi  $\pm$  standar deviasi (peluang menemukan global optima). Dari data tersebut dapat dilihat, Cuckoo Search memiliki persentase peluang untuk menemukan global optima lebih besar daripada algoritma Particle Swarm Optimization dan Algoritma Genetika.

Tabel 1. Perbandingan hasil antara algoritma Genetik, PSO dan CS pada dua belas fungsi stokastik berbeda. (Yang dan Deb, 2009)

Fungsi	GA	PSO	CS
Michaelwicz	89325 $\pm$ 7914 (95%)	6922 $\pm$ 537 (98%)	3221 $\pm$ 519 (100%)
Rosenbrock	55723 $\pm$ 8901 (90%)	32756 $\pm$ 5325 (98%)	5923 $\pm$ 1937 (100%)
De Jong	15232 $\pm$ 1270 (100%)	10079 $\pm$ 970 (100%)	3015 $\pm$ 540 (100%)
Schwefel	23790 $\pm$ 6523 (95%)	92411 $\pm$ 1163 (97%)	4710 $\pm$ 592 (100%)
Ackley	32720 $\pm$ 3327 (90%)	23407 $\pm$ 4325 (92%)	4936 $\pm$ 903 (100%)
Rastrigin	110523 $\pm$ 5199 (77%)	79491 $\pm$ 3715 (90%)	10354 $\pm$ 3755 (100%)
Easom	19239 $\pm$ 3307 (92%)	17273 $\pm$ 2929 (90%)	6751 $\pm$ 1902 (100%)
Griewangk	70925 $\pm$ 7652 (90%)	55970 $\pm$ 4223 (92%)	10912 $\pm$ 4050 (100%)

Stokastik pertama Yang	79025 ± 6312 (49%)	34056 ± 4470 (90%)	11254 ± 2733 (99%)
Stokastik kedua Yang	35072 ± 3730 (54%)	22360 ± 2649 (92%)	8559 ± 3480 (98%)
Robsenrockdengan komponen stokastik	63268 ± 5091 (40%)	49152 ± 6505 (89%)	10564 ± 4297 (99%)
Stokasik De Jong	24164 ± 4923 (68%)	11780 ± 4912 (94%)	7723 ± 2504 (100%)

### 3. METODOLOGI

Dalam subseksi berikutnya akan diberikan tahapan-tahapan yang dilakukan dalam penelitian ini.

#### 3.1. Inisialisasi

Pada permulaan proses, matriks fuzzy untuk setiap cuckoo akan diinisialisasi secara random, dan dapat dituliskan dalam bentuk sebagai berikut.

$$R = (r_{ij})_{m \times n} = \begin{bmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & r_{nn} \end{bmatrix}, r_{ij} \in [0, 1] \quad (3)$$

di mana, kondisi awal dari matriks yang diinisialisasi tersebut memenuhi syarat  $R_{ij} \in [0, 1]$  dan jika  $i = j$ ,  $R_{ij} = 0$ . Setelah matriks selesai diinisialisasi, matriks kemudian dikonversi menjadi sebuah jalur kota menggunakan cara yang telah dijelaskan sebelumnya yaitu defuzzifikasi dengan metode angka terbesar (*Max Number Method*).

Setelah proses defuzzifikasi selesai untuk semua cuckoo, proses dilanjutkan dengan menginisialisasi sarang yang terbaik, dengan melakukan seleksi berdasarkan nilai fitness diantara semua sarang cuckoo yang telah tersedia. Semakin kecil nilai fitness, maka semakin bagus sarang tersebut. Nilai fitness didapatkan dengan cara mencari jalur total yang menghubungkan antar tiap kota.

#### 3.2. Pencarian

Proses pencarian terbagi ke dalam dua tahapan yaitu proses pencarian sarang dan proses penggantian sarang terburuk. Proses pencarian sarang merupakan tahapan di mana akan dilakukan pemilihan secara acak dari sejumlah cuckoo yang tersedia, di mana pada sarang yang ditinggalinya akan dilakukan random step dengan menggunakan Lévy flights sebagai acuan step, sedangkan proses penggantian sarang terburuk merupakan proses penggantian sejumlah sarang cuckoo yang dianggap buruk dengan persentase pergantian berdasarkan input (PA) yang diterima.

##### 3.2.1. Pencarian Sarang

Pada proses pencarian sarang, sejumlah sarang dari burung cuckoo yang ada akan dipilih secara random, di mana sarang yang terpilih tidak boleh sarang yang terbaik, hal ini disebabkan karena sarang yang terbaik akan digunakan dalam proses perhitungan. Pada sarang yang terpilih ini kemudian akan dilakukan perhitungan terhadap matriks fuzzy yang terdapat di dalamnya. Proses perhitungan dapat dijabarkan sebagai berikut :

$$A1 = A + \text{stepsize} \quad (4)$$

dimana  $A$  menyatakan matriks fuzzy lama untuk sarang yang terpilih,  $A1$  menyatakan matriks fuzzy baru untuk sarang yang terpilih dan  $\text{stepsize}$  menyatakan besar *randomstep* yang menggunakan formula sebagai berikut.

$$\text{stepsize} = (L * \frac{\text{jarak rata-rata kota}}{\text{jarak kota m \& kota n}} * |A - B|) \quad (5)$$

dimana  $L$  menyatakan hasil perhitungan Lévy Flights dengan algoritma mantegna.  $A$  menyatakan matriks fuzzy lama untuk sarang yang terpilih dan  $B$  menyatakan matriks fuzzy dari sarang yang terbaik. Untuk proses perhitungan lévy flights dengan algoritma mantegna dapat dijabarkan sebagai berikut.

1. Dilakukan perhitungan terhadap besar langkah  $s$  dengan menggunakan rumus berikut ini :

$$s = \frac{u}{|v|^{1/\beta}} \quad (6)$$

di mana  $u$  dan  $v$  merupakan nilai yang diambil dari distribusi normal, yaitu  $u \sim N(0, \sigma_u^2)$ ,  $v \sim N(0, \sigma_v^2)$

2. Masukkan nilai  $\sigma_u = \left\{ \frac{\tau(1+\beta) \sin(\frac{\pi\beta}{2})}{\tau[\frac{1+\beta}{2}]^{\beta} \cdot 2^{(\beta-1)/2}} \right\}^{1/\beta}$  dan  $\sigma_v = 1$

dimana  $\tau$  menyatakan fungsi distribusi gamma (menggunakan fungsi dari library matematika bawaan) dan  $\beta$  menyatakan nilai konstanta perhitungan, dalam hal ini nilai yang digunakan adalah  $\frac{3}{2}$

Dalam proses komputasi pada program, hasil perhitungan lévy flights dikonversi menjadi bentuk berikut.

$$L = \frac{\text{randDouble}() * \sigma u}{|\text{randDouble}()|^{\frac{1}{\beta}}} \quad (7)$$

dengan  $\text{randDouble}()$  menyatakan nilai random antara 0 sampai 1 dan nilai  $\sigma u$  dimasukkan sesuai dengan rumus yang telah dijabarkan sebelumnya.

Setelah proses perhitungan selesai dilakukan, akan dilakukan proses defuzzifikasi dari matriks. Apabila jalur yang dihasilkan memiliki nilai fitness yang lebih buruk, matriks fuzzy dari sarang tersebut akan dikembalikan ke semula. Dan sebaliknya, apabila matriks fuzzy dari sarang tersebut memiliki nilai fitness yang lebih bagus maka matriks tersebut akan digantikan matriks fuzzy hasil kalkulasi. Secara ringkas, langkah pengerjaan untuk proses pencarian sarang dapat dilihat sebagai berikut.

a. Lakukan pengurangan antara matriks fuzzy sarang terbaik dan sarang yang terpilih seperti di Gambar 1.

Sarang Terbaik					Sarang Terpilih					Hasil Pengurangan				
Kota	1	2	3	4						Kota	1	2	3	4
1	0	0.24	0.12	0.16						1	0	0.11	0.39	0.15
2	0.24	0	0.51	0.73						2	0.11	0	0.18	0.09
3	0.12	0.51	0	0.85						3	0.39	0.18	0	0.59
4	0.16	0.73	0.85	0						4	0.15	0.09	0.59	0

Gambar 1. Pengurangan Matriks Fuzzy Sarang Terbaik dan Sarang Terpilih

b. Lakukan perhitungan matriks rata-rata jarak kota dibagi jarak kota seperti di Tabel 2.

Tabel 2. Matriks rata-rata jarak kota dibagi jarak kota

Kota	1	2	3	4
1	0	$\frac{\text{Rerata jarak}}{\text{Jarak kota (2,1)}}$	$\frac{\text{Rerata jarak}}{\text{Jarak kota (3,1)}}$	$\frac{\text{Rerata jarak}}{\text{Jarak kota (4,1)}}$
2	$\frac{\text{Rerata jarak}}{\text{Jarak kota (1,2)}}$	0	$\frac{\text{Rerata jarak}}{\text{Jarak kota (3,2)}}$	$\frac{\text{Rerata jarak}}{\text{Jarak kota (4,2)}}$
3	$\frac{\text{Rerata jarak}}{\text{Jarak kota (1,3)}}$	$\frac{\text{Rerata jarak}}{\text{Jarak kota (2,3)}}$	0	$\frac{\text{Rerata jarak}}{\text{Jarak kota (4,3)}}$
4	$\frac{\text{Rerata jarak}}{\text{Jarak kota (1,4)}}$	$\frac{\text{Rerata jarak}}{\text{Jarak kota (2,4)}}$	$\frac{\text{Rerata jarak}}{\text{Jarak kota (3,3)}}$	0

- Lakukan perkalian antara hasil pengurangan yang didapatkan pada langkah a dengan matriks pembagian pada langkah b.
- Lakukan perhitungan terhadap random step Lévy Flights dengan algoritma Mantegna untuk tiap baris dan kolom dengan menggunakan rumus akhir yang telah diberikan.
- Lakukan penjumlahan antara matriks random step dengan matriks yang didapatkan pada langkah c

### 3.2.2. Penggantian Sarang Terburuk

Pada proses penggantian sarang terburuk, sejumlah sarang selain sarang terbaik akan dipilih berdasarkan input parameter (PA) bila dibandingkan dengan suatu nilai acak. Apabila nilai acak yang didapatkan lebih besar dari nilai PA, maka sarang tersebut akan dimasukkan ke dalam list sarang yang akan digantikan. Sebaliknya apabila lebih kecil maka sarang tersebut tidak akan digantikan.

Proses penggantian dilakukan dengan melakukan persilangan edge antara sarang yang terpilih dengan sarang yang terbaik. Setelah proses persilangan selesai dilakukan, akan dilakukan swap nilai dari probabilitas yang terdapat pada matriks fuzzy sarang tersebut.

## 4. HASIL DAN PEMBAHASAN

Berikut ditampilkan hasil perbandingan hasil dan lama pencarian yang didapatkan antara program TSP yang menggunakan algoritma genetika, dengan program TSP yang menggunakan algoritma *cuckoo search* dengan jumlah kota yang dibandingkan adalah 10, 20, 40, 50, 80, 120, 150 dan 180 kota, Acuan perbandingan performansi untuk algoritma *Cuckoo Search* didasarkan pada hasil optimal yang sanggup didapatkan bila dibandingkan dengan algoritma genetika dalam waktu yang relative sama. Dalam hal ini, algoritma genetika akan dilakukan pencarian hasil optimalnya terlebih dahulu dengan maksimal iterasi 10.000.000. Hasil dan lama waktu yang didapatkan akan digunakan sebagai acuan *cuckoo search* untuk melakukan pencarian hasil optimal. Hasil pengujian yang didapatkan dari kedua algoritma untuk pengetesan 10, 20, 50, 40, 80 dan 120 kota bisa dilihat di Tabel 3.

Tabel 3. Hasil Optimal yang sanggup dicapai oleh tiap algoritma

No.	Banyak Kota	Algoritma Genetik		Algoritma Cuckoo Search	
		Hasil(px)	Waktu	Hasil(px)	Waktu
1	10 Kota	1053.50	00:00:00.03	1053.50	00:00:00.01
2	20 Kota	1727.18	00:00:02.19	1763.55	00:00:00.36
3	40 Kota	2085.47	00:01:55.00	2155.42	00:01:20.05
4	50 Kota	3218.03	00:00:50.60	3169.62	00:00:00.99
5	80 Kota	4117.52	00:05:34.00	4176.82	00:00:05.57
6	120 Kota	5469.57	00:40:46.00	5259.01	00:00:10.05
7	150 Kota	5279.21	00:27:40.54	5056.48	00:00:30.12
8	180 Kota	6087.49	00:43:53.04	5987.54	00:00:53.49

Dari hasil perbandingan di Tabel 3, didapatkan kalau performansi pencarian dengan algoritma *cuckoo search* memiliki efisiensi yang lebih besar bila dibandingkan dengan algoritma genetika. Hal ini dapat dibuktikan dengan melihat perbandingan antara lama pencarian dan hasil pencarian yang didapatkan.

## 5. SIMPULAN DAN SARAN

Berikut adalah simpulan dan saran dari penelitian ini.

### 5.1 Simpulan

Dari pembahasan yang telah dilakukan sebelumnya, maka dapat ditarik beberapa kesimpulan sebagai berikut :

1. Semakin besar nilai PA maka semakin cepat proses pencarian dan sebaliknya sedangkan semakin besar jumlah cuckoo yang digunakan maka semakin lama proses pencarian dan sebaliknya
2. Semakin besar nilai dari PA atau Total Cuckoo, semakin besar pula kemungkinan untuk mendapatkan hasil yang lebih bagus
3. Algoritma *cuckoo search* menunjukkan performansi pencarian yang lebih bagus untuk jumlah kota yang semakin besar bila dibandingkan dengan algoritma genetika dalam penyelesaian masalah TSP

### 5.2 Saran

Berikut saran-saran yang diberikan untuk kemajuan pada penerapan algoritma *cuckoo search* berikutnya :

1. Perlu dilakukan analisa lebih lanjut mengenai random step *Lévy Flights* yang lebih cocok untuk penyelesaian masalah diskrit seperti TSP.
2. Dapat dilakukan penelitian lebih lanjut mengenai sistem representasi TSP untuk algoritma *cuckoo search* selain menggunakan *Fuzzy Matrix*.

## 6. DAFTAR RUJUKAN

- [1] Barthelemy, P., Bertolotti, J., & Wiersma, D. S., 2008. *A Lévy flight for light*. Nature, 453(7194), 495-498.
- [2] Eberhart, R., & Kennedy, J., 1995, October. *A new optimizer using particle swarm theory*. In Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on (pp. 39-43). IEEE.
- [3] Geem, Z. W., Kim, J. H., & Loganathan, G. V., 2001. *A new heuristic optimization algorithm: harmony search*. Simulation, 76(2), 60-68
- [4] Goldberg, D. E., & Holland, J. H., 1988. *Genetic algorithms and machine learning*. Machine learning, 3(2), 95-99.

- [5] Yang, X. S., & Deb, S., 2009. *Cuckoo search via Lévy flights*. In Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on (pp. 210-214). IEEE.
- [6] Yang, X.S., 2010. *Nature Inspired Metaheuristic Algorithm Second Edition*, Luniver Press, United Kingdom