

STUDI KOMPARATIF ALGORITMA ANT DAN ALGORITMA GENETIK PADA TRAVELLING SALESMAN PROBLEM

Bambang Siswoyo¹⁾, Andrianto²⁾

¹ Sekolah Tinggi Teknologi Informatika Sony Sugema, Bandung

² Fakultas Teknik dan Ilmu Komputer, Jurusan Teknik Informatika, UNIKOM

Abstract

Travelling Salesman Problem (TSP) is a problem in which a trader must make a circuit or a route through several cities, where each city is visited only once and to minimize the total distance traveled . The limit is that every city that passed only once visited, and the merchant must return to the city of origin at the end of his journey . The program searches the shortest distance made, based on two algorithms namely genetic algorithms and Ant Colony System (ACS) .

Genetic algorithm is an algorithm that works based on the mechanism of natural selection and natural genetics. The main idea behind this algorithm is to choose the best individuals from a population of individuals and between individuals recombination to generate new individuals are expected to be better than the previous individual.

Ant Colony System is an algorithm that uses the workings of the ant colony to solve TSP problem. At ACS, there are a set of ants working together to determine the best solution TSP, ants working together through indirect communication using pheromone trail deposited on the sides of the graph TSP.

From the difference of this approach, will be studied and compared the ability of each approach in solving the TSP .

Keywords: *Traveling Salesman Problem , Genetic Algorithms , Ant Colony System*

Abstrak

*Travelling Salesman Problem (TSP) yaitu sebuah masalah dimana seorang pedagang harus membuat suatu sirkuit atau rute melalui beberapa kota, dimana setiap kota hanya dikunjungi sekali dan dengan meminimalkan total jarak yang dilalui. Batasannya adalah bahwa setiap kota yang dilalui hanya sekali dikunjungi, dan pedagang tersebut harus kembali ke kota asal pada akhir perjalanannya. Program pencarian jarak terpendek yang dibuat, didasarkan pada dua algoritma yaitu algoritma genetik dan algoritma *Ant Colony System* (ACS).*

Algoritma genetik adalah algoritma yang bekerja berdasarkan mekanisme seleksi alam dan genetika alam. Ide utama dibalik algoritma ini adalah memilih individu-individu

terbaik dari sebuah populasi individu dan melakukan rekombinasi antar individu untuk membangkitkan individu baru yang diharapkan lebih baik dari individu sebelumnya.

Ant Colony System adalah algoritma yang menggunakan cara kerja koloni semut untuk memecahkan masalah TSP. Pada ACS ini terdapat sekumpulan semut yang bekerja sama untuk menentukan solusi TSP yang paling baik, semut-semut bekerja sama melalui komunikasi tidak langsung dengan menggunakan jejak *pheromone* yang disimpan pada sisi-sisi dari graph TSP.

Dari perbedaan pendekatan ini, akan dipelajari dan diperbandingkan kemampuan masing-masing pendekatan dalam menyelesaikan TSP.

Kata kunci: *Travelling Salesman Problem, Algoritma Genetik, Ant Colony System*

1. Latar Belakang Masalah

Travelling Salesman Problem (TSP) adalah sebuah masalah dimana seorang salesman mempunyai tugas untuk mengirimkan pesanan barang ke rumah klien yang berada di sejumlah tempat yang berbeda di sebuah kota. Salesman tersebut mempunyai masalah dalam hal menentukan tempat mana yang terlebih dahulu dikunjungi sedemikian sehingga total jarak dan waktu bepergian diperkecil dan setiap kota hanya boleh dilalui sekali dalam satu perjalanan, dan perjalanan berakhir pada kota awal dimana seorang sales memulai perjalanannya. Permasalahan yang timbul yaitu seorang sales harus bisa menentukan rute terpendek dengan waktu yang lebih efektif sehingga suatu tempat tidak terlewat lebih dari satu kali.

Algoritma genetik adalah algoritma pencarian *heuristic* yang bekerja berdasarkan mekanisme seleksi alam dan genetika alam. Ide utama dibalik algoritma ini adalah memilih individu-individu terbaik dari sebuah populasi individu dan melakukan rekombinasi antar individu untuk membangkitkan individu baru yang diharapkan lebih baik dari individu sebelumnya. Sedangkan, Algoritma Ant Colony System (ACS) juga merupakan algoritma pencarian *heuristic* dalam penyelesaian kasus kombinatorial. Dalam pencarian solusi kasus kombinatorial, Pada ACS ini terdapat sekumpulan semut yang bekerja sama untuk menentukan solusi TSP yang paling baik, semut-semut bekerja sama melalui komunikasi tidak langsung dengan menggunakan jejak *pheromone* yang disimpan pada sisi-sisi dari graph TSP.

2. Identifikasi Masalah

Berdasarkan latar belakang diatas, masalah yang akan dibahas dalam tugas akhir ini adalah:

1. Bagaimana algoritma genetik dan algoritma ACS dapat menyelesaikan masalah TSP sehingga dapat menghasilkan jalur terpendek.
2. Bagaimana membuat perangkat lunak dari Algoritma genetik dan algoritma ACS.

3. Batasan Masalah

Batasan masalah yang menjadi tolak ukur pembuatan tugas akhir ini, yaitu

1. Algoritma genetik dan algoritma ACS diterapkan hanya untuk pencarian jarak terpendek pada masalah TSP.
2. Jalur transportasi yang digunakan yaitu jalur transportasi kota di Pulau Sumatera dengan maksimal node 40 kota.
3. Jalur TSP yang digunakan adalah simetri yaitu jarak pergi dan jarak pulang langsung antara dua kota besarnya sama.
4. Tidak ada prioritas kota mana yang akan dilalui terlebih dahulu.
5. Graph yang digunakan adalah graph lengkap berbobot.
6. Jarak yang digunakan berdasarkan jarak pada peta.

4. Maksud dan Tujuan

Maksud dari penulisan skripsi ini adalah menerapkan algoritma genetik dan algoritma ACS sebagai metode untuk mencari jarak terpendek dalam penyelesaian *Travelling Salesman Problem*.

Tujuan dari penelitian adalah :

1. Untuk mengetahui cara kerja algoritma genetik dan algoritma ACS dalam memecahkan masalah *Travelling Salesman Problem*.
2. Membuat Aplikasi atau perangkat lunak untuk mensimulasikan kedua algoritma dalam menyelesaikan masalah TSP.
3. Melakukan analisis hasil eksekusi dari perangkat lunak yang dibangun terhadap beberapa contoh kasus

dalam menyelesaikan *Travelling salesman problem*. Analisis bertujuan untuk mengetahui perbandingan performansi kedua algoritma tersebut.

5. Landasan Teori

5.1 Travelling Salesman Problem

Travelling salesman problem termasuk ke dalam persoalan yang sangat terkenal dalam teori graf. Nama persoalan ini diilhami oleh masalah seorang pedagang yang akan mengunjungi sejumlah kota. Uraian persoalannya adalah sebagai berikut: diberikan sejumlah kota dan jarak antar kota. Tentukan sirkuit terpendek yang harus dilalui oleh seorang pedagang bila pedagang itu berangkat dari sebuah kota asal dan menyinggahi setiap kota tepat satu kali dan kembali lagi ke kota asal kebererangkatan. Kota dapat dinyatakan sebagai simpul graf, sedangkan sisi menyatakan jalan yang menghubungkan antar dua buah kota. Bobot pada sisi menyatakan jarak antara dua buah daerah. Persoalan pedagang tidak lain adalah menentukan sirkuit Hamilton yang memiliki bobot minimum pada sebuah graf terhubung.

5.2 Algoritma Genetik

Pada algoritma ini, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin yang dikenal dengan istilah populasi. Individu yang terdapat dalam suatu populasi disebut dengan kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan dengan istilah generasi. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang dengan fungsi *fitness*. Nilai *fitness* suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut. Generasi berikutnya

dikenal dengan istilah anak (*offspring*) yang terbentuk dari gabungan dua kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilangan (*crossover*). Selain operator penyilangan, suatu kromosom dapat juga dimodifikasi dengan menggunakan operator mutasi. Populasi generasi yang baru dibentuk dengan cara menyeleksi nilai *fitness* dari kromosom induk (*parent*) dan nilai *fitness* dari kromosom anak (*offspring*), serta menolak kromosom-kromosom yang lainnya sehingga ukuran populasi (jumlah kromosom dalam suatu populasi) konstan. Setelah melalui beberapa generasi, maka algoritma ini akan konvergen ke kromosom terbaik.

5.2.1 Operator Genetika

Pada algoritma genetika terdapat tiga operator genetika, yaitu :

- Seleksi
- Pindah Silang (*crossover*)
- Mutasi

5.3 Algoritma Ant Colony System (ACS)

Dalam algoritma ACS akan dilakukan proses iterasi (pengulangan dalam mencari jalur terpendek dan jarak perjalanan yang terpendek). Dalam sebuah proses iterasi, semua semut akan melakukan perjalanan sampai semua kota dikunjungi, ketika semut melakukan perjalanannya, semut akan melakukan *state transition* untuk memilih kota yang akan dituju dan aturan *local updating* untuk memodifikasi jumlah *pheromone* pada sisi yang telah dikunjungi oleh semut. Setelah semua kota dikunjungi oleh semut, maka digunakan aturan *global updating* untuk mengurangi jumlah *pheromone* pada semua sisi dan menambah jumlah *pheromone* pada sisi-sisi yang termasuk dalam perjalanan dengan jarak terpendek. Cara kerja ACS

secara umum dapat dilihat pada algoritma berikut :

<p>Inisialisasi semua parameter dan variable yang akan digunakan</p> <p>Ulangi (pengulangan pada bagian ini disebut iterasi)</p> <p>Masing-masing semut diletakan secara acak pada sebuah kota yang diperoleh dari hasil inisialisasi</p> <p>Ulangi</p> <p>Masing-masing semut menerapkan aturan <i>state transition</i> dan <i>local updating</i> sampai semua semut selesai mengunjungi semua kota</p> <p>Cari panjang perjalanan untuk semua semut</p> <p>Cari semut yang memiliki jarak perjalanan terpendek</p> <p>Terapkan aturan <i>global updating</i></p> <p>Sampai jumlah iterasi yang dikehendaki</p> <p>Tampilkan jalur dan jarak yang terpendek</p>

Ada tiga buah aturan yang akan digunakan dalam ACS yaitu :

1. Aturan State Transition.

Aturan ini digunakan oleh semut untuk memutuskan ke kota mana ia akan pergi, dengan menerapkan aturan ini, semut dapat memilih untuk pergi kesisi yang baru (sisi yang belum dilewati oleh semut) atau sisi yang terbaik (sisi yang memiliki jumlah *pheromone* terbanyak dan jarak terpendek) secara probabilitas.

$$s = \begin{cases} \arg \max_{i \in Jk(i)} \{ \tau(i,l) \cdot [\eta(i,l)]^\beta \} \\ \text{Pilih menurut persamaan berikutnya} \end{cases}$$

dimana :

q_0 = parameter yang menunjukkan hubungan antara *exploitasi* dan *eksplorasi* ($0 < q_0 < 1$), q = parameter bernilai acak antara 0 sampai 1 ($0 < q < 1$), β = Mengatur bobot intensitas relative jejak *pheromone*. Jika $\beta = 0$ jarak antar kota tidak berpengaruh, $\tau(i,l)$ = jumlah *pheromone* pada sisi yang menghubungkan kota i dengan kota l, $\eta(i,l)$ = Untuk memilih

kota terdekat yang mempunyai probabilitas tinggi dan merupakan fungsi heuristic yang digunakan sebagai invers jarak $d(i,l) = 1/dil$, $Jk(i)$ = Daftar sekumpulan kota yang masih harus dikunjungi oleh semut k ketika semut masih di kota i, lengkap dengan daftar semua kota yang akan dikunjungi, $Jk(i)$ disebut juga sebagai tempat penyimpanan kota-kota yang akan dikunjungi semut

Jika $q \leq q_0$ (*exploitasi*) yaitu semut akan memilih jalur yang terbaik menurut nilai *pheromone* dan dari heuristic η .

Jika $q \geq q_0$ (*eksplorasi*) yaitu semut akan memilih kota berikutnya menurut persamaan probabilitas berikut ini :

$$Pk(i,j) = \begin{cases} \frac{[\tau(i,j)] \cdot [\eta(i,j)]^p}{\sum_{j \in Jk(i)} [\tau(i,l)] \cdot [\eta(r,l)]^p} \\ 0 \end{cases}$$

dimana :

$P(i,j)$ = Probabilitas semut k memilih untuk berpindah dari kotai ke kota j

2. Aturan Local Updating

Setelah semut melalui sisi tertentu, maka jumlah *pheromone* pada sisi jika $J \in Jk(i)$ tersebut akan berkurang. Jumlah *pheromone* pada sisi yang baru saja dikunjungi oleh semut dapat diperoleh dengan menerapkan *local updating* jika $J \notin Jk(i)$

$$\tau(i,j) \leftarrow (1 - \rho) \cdot \tau(i,j) + \rho \cdot \tau_0$$

dimana :

ρ = parameter yang merepresentasikan *evaporation pheromone local* yang bernilai antara 0 sampai 1 ($0 < \rho < 1$)

$$\Delta \tau(i,j) = \tau_0 = \frac{1}{\text{jarakawal} * \text{jumlahkota}}$$

3. Aturan Global Updating

Aturan ini digunakan setelah semua semut membentuk jalur perjalanan. Pada

aturan *global updating* akan dilakukan pengurangan jumlah *pheromone* pada semua sisi, kemudian dilakukan penambahan jumlah *pheromone* pada sisi-sisi yang termasuk dalam perjalanan dengan jarak yang terpendek.

$$\tau(i, j) \leftarrow (1 - \rho) \cdot \tau(i, j) + \rho \cdot \frac{1}{L_{gb}}$$

dimana :

ρ = Parameter yang merepresentasikan *evaporation pheromone* global yang bernilai antara 0 dan 1 ($0 < \rho < 1$)

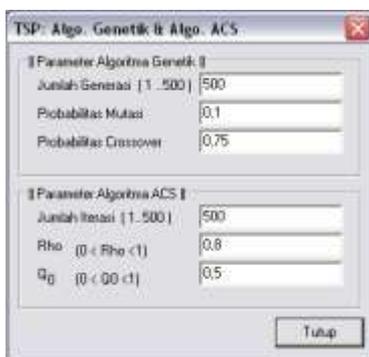
L_{gb} = Jarak tempuh terpendek yang di peroleh sejak awal iterasi sampai dengan iterasi yang sedang berlangsung.

6. Implementasi

Tampilan program jendela utama



Tampilan jendela input parameter



Tampilan jendela analisis perbandingan

No	Kota	Jumlah
1	Pangkalant	0.25 13.100.61 9T
2	Pangkepangan	0.64 13.103.21 9T
3	Mataram	4.33 13.104.31 9T
4	Korlekan	4.07 13.103.36 9T
5	Sugabahan	0.88 13.101.50 9T
6	Tombokan	0.62 13.103.84 9T
7	Jambi	4.01 13.103.64 9T
8	Pua	0.21 13.103.85 9T
9	Bergulu	2.01 13.102.25 9T
10	Muarasin	2.67 13.103.79 9T
11	Lakengen	4.54 13.106.46 9T
12	Palingi	0.97 13.100.31 9T
13	Medan	2.54 13.100.45 9T
14	Rantakapat	2.08 13.100.82 9T
15	Siboga	4.72 13.100.77 9T
16	Muarabuk	4.03 13.104.89 9T

7. Kesimpulan dan Saran

7.1 Kesimpulan

Berdasarkan uraian dan penjelasan yang telah dikemukakan diatas, maka penulis dapat mengambil kesimpulan sebagai berikut :

1. Algoritma genetik dan Algoritma ACS dapat dijadikan sebagai alternatif untuk memecahkan masalah Travelling Salesman Problem meskipun tidak selalu memberikan solusi yang tepat, dikarenakan cara kerja Algoritma genetik dan Algoritma ACS yang bersifat probabilitas.
2. Jika jumlah kota relatif banyak, waktu pencarian algoritma genetik lebih cepat dibanding algoritma ACS sedangkan untuk jumlah kota relatif kecil waktu pencarian algoritma ACS lebih cepat.
3. Susunan rute kota yang dihasilkan untuk menghasilkan rute terpendek setiap algoritma berbeda.
4. Jika jumlah kota sedikit, jarak antar kota sama.
5. Pada Algoritma ACS iterasi sangat berpengaruh pada waktu komputasi, walaupun susunan rute yang terlewat relatif sama.
6. Semakin banyak generasi dan jumlah individu yang dimunculkan dalam menjalankan algoritma genetik semakin membuka peluang untuk mendapatkan *fitness* terbaik.

7.2 Saran

Adapun beberapa hal yang penulis sarankan untuk pihak yang akan melakukan penelitian lebih lanjut diharapkan dapat memperbaiki kekurangan dari skripsi ini. Untuk mencapai tujuan tersebut diberikan saran sebagai berikut :

1. Program yang telah berhasil dibuat masih terdapat banyak kekurangan, diharapkan dapat dikembangkan lagi dengan menambah jumlah kota yang lebih banyak (lebih dari 40 kota)
2. Program ini dapat dikembangkan lebih lanjut untuk memecahkan masalah TSP asimetris
3. Dari hasil penelitian masih banyak ditemui kekurangan sebagai akibat keterbatasan pengetahuan penulis serta ilmu yang dimiliki.
4. Dilakukan Perbandingan dengan Metode – metode yang lain.

Daftar Pustaka

Deo, Narsingh. *Graph Theory (With applications To Engineering And Computer Science)*

Dorigo, Marco, Ant Colony System : *A Cooperative Learning Approach to the traveling Salesman Problem*, IEEE Transaction on Evolutionary Computation,10,1997,1-24

Dorigo, Marco, *Ant Colonies The Travelling Salesman Problem*, Biosystems,1, 1997,1-9

Harahap, Fitriana, S.T (2005). *“Penerapan Algoritma Ant Colony System (ACS) Dan Algoritma Generate And Test Dalam Pemecahan Solusi Travelling Salesman Problem (TSP)”*, Tugas Akhir Teknik Informatika UNIKOM, Bandung.

Kusumadewi, Sri. *Artificial Intelligence (Teknik dan Aplikasinya)*, Yogyakarta: Graha Ilmu.2003.

Kusumadewi, Sri,. Purnomo, Hari,. *Penyelesaian Masalah Optimasi dengan Teknik – teknik Heuristik*, Yogyakarta: Graha Ilmu.2005.

Marco Dorigo dan Gianni Di Caro, *The Ant Colony Optimization Meta – Heuristic* ¹. Belgium, 1996