

# Penyelesaian Masalah *Travelling Salesman Problem* Menggunakan *Ant Colony Optimization* (ACO)

Anna Maria<sup>1</sup>, Elfira Yolanda Sinaga<sup>2</sup>, Maria Helena Iwo<sup>3</sup>

*Laboratorium Ilmu dan Rekayasa Komputasi  
Departemen Teknik Informatika, Institut Teknologi Bandung  
Jl. Ganesha 10, Bandung*

E-mail : [if13075@students.if.itb.ac.id](mailto:if13075@students.if.itb.ac.id)<sup>1</sup>, [if13087@students.if.itb.ac.id](mailto:if13087@students.if.itb.ac.id)<sup>2</sup>,  
[if13088@students.if.itb.ac.id](mailto:if13088@students.if.itb.ac.id)<sup>3</sup>

## Abstrak

Masalah TSP merupakan salah satu masalah yang populer di bidang keinformatikaan. Masalah tersebut telah mengundang berbagai macam solusi untuk menyelesaikannya. Solusi yang diharapkan adalah solusi yang mempunyai kompleksitas algoritma terkecil (paling mangkus). Makalah ini membahas solusi pencarian jalur terpendek menggunakan ACO. ACO adalah salah satu *Swarm Intelligent*. *Swarm intelligence* adalah suatu metode pemecahan masalah dengan menggunakan sifat koloni hewan.

Kata Kunci : TSP, *Ant Colony System*, *Swarm Intelligent*.

## 1. Pendahuluan

TSP mempermasalahkan pencarian jalur terdekat dari satu titik ke satu titik itu lagi. Penyelesaian persoalan ini melibatkan banyak lintasan dengan jarak yang berbeda. Solusi merupakan salah satu dari sekian banyak alternatif jalur yang diambil. Pencarian jalur terdekat (TSP) dapat diaplikasikan dalam banyak hal, karena itu sampai saat ini penyelesaian yang paling baik masih terus dicari. Berbagai metode sudah ada dan dapat digunakan untuk menyelesaikan persoalan ini. Metode-metode tersebut bervariasi kompleksitas algoritmanya. Makalah ini khusus membahas satu metode untuk menyelesaikan persoalan TSP, yakni ACO. Makalah ini mengangkat pemahaman mengenai ACO dan bagaimana metode ini menyelesaikan persoalan TSP.

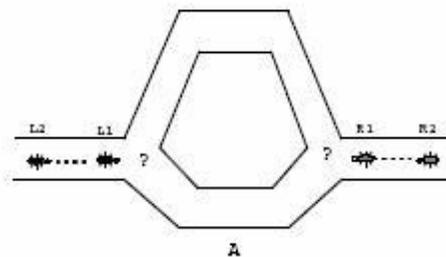
## 2. Pola Tingkah Laku Semut dalam Dunia Nyata

Pada kehidupan sebenarnya, semut-semut meninggalkan sarang untuk mencari makanan dan harus mencari kembali sarang mereka. Misalkan ada segerombolan semut yang mencari makanan., maka semut yang berada di depan harus memilih lintasan tertentu untuk dilewati. Pada saat semut pertama bejalan, semut tersebut meninggalkan hormon feromon yang dapat dicium oleh semut berikutnya,

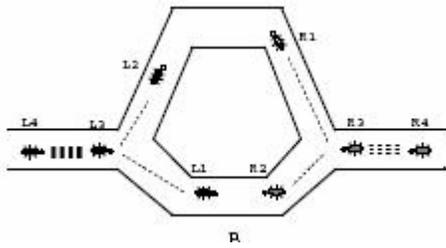
sehingga semut-semut berikutnya tahu apakah tempat tersebut sudah dilewati atau belum.

Semut yang melewati lintasan yang lebih pendek akan meninggalkan aroma feromon yang lebih tajam daripada semut yang menempuh lintasan yang lebih panjang. Hal ini terjadi karena feromon yang ditinggalkan dapat menguap. Saat semut-semut yang di belakang mengikuti semut-semut yang di depannya, semut-semut tersebut akan memilih lintasan berdasarkan kuatnya aroma feromon dan jarak lintasan. Semakin banyak semut yang menempuh suatu lintasan tertentu, maka aroma feromon pada lintasan tersebut akan semakin kuat sehingga semut-semut berikutnya akan mengikuti lintasan tersebut.

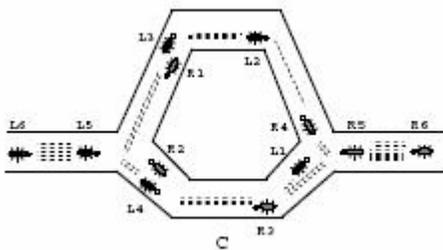
*Ant Colony System* adalah salah satu algoritma yang digunakan untuk memecahkan TSP dengan menggunakan sifat koloni semut untuk Contoh dalam gambar :



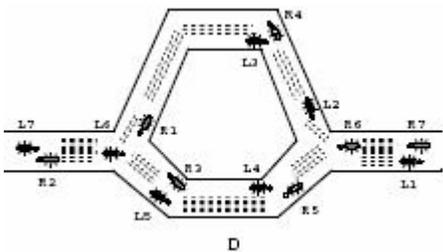
Gambar 1 Semut-semut sampai pada persimpangan



**Gambar 2 . Sebagian semut memilih lintasan yang diatas sebagian lagi memilih lintasan yang di bawah. Kedua lintasan ditempuh dengan kecepatan yang sama.**



**Gambar 3 Karena kecepatan menempuh lintasannya sama, maka semut-semut yang dibawah lebih cepat sampai ke tujuan**



**Gambar 4 Semut-semut yang ada di belakang akan mengikuti lintasan yang terpendek karena aroma feromon tercium lebih tajam**

### 3. Pemetaan Pola Tingkah Laku Kawan Semut ke bidang Intelejensia Buatan

Pola tingkah laku semut di atas menghasilkan inspirasi untuk memecahkan masalah TSP. Strategi ACO untuk memilih kota berikutnya yang akan dikunjungi memperhitungkan dua hal, yaitu jarak antara dua buah kota dan jumlah feromon yang ditinggalkan. Karena dalam TSP kota – kota hanya boleh dikunjungi sebanyak satu kali, maka kawan semut dalam dunia intelejensia buatan dibekali dengan memori untuk menyimpan kota apa saja yang telah dikunjungi..

Untuk mempermudah pemecahan masalah TSP, maka harus dimodelkan ke dalam Graf. Secara umum, strategi ACO yang digunakan untuk masalah TSP adalah sebagai berikut. Pada tahap awal, tempatkan semut secara acak pada setiap simpul dan berikan feromon yang sama banyak setiap sisi dalam graf. Pada setiap langkah, setiap semut harus memilih simpul berikutnya yang akan dikunjungi. Simpul yang akan dikunjungi tersebut harus merupakan simpul yang belum ada dalam memori semut itu. Fungsi untuk memilih simpul berikutnya ini dinyatakan dalam **Aturan Kemungkinan Transisi (Probabilistic Transition Rule)**. Jumlah feromon pada sisi yang dilalui oleh semut akan diperbaharui jumlahnya, yakni dikurangi dan ditambah. Fungsi untuk mengurangi dan menambahkan feromon yang ditinggalkan oleh semut dinyatakan dalam **Aturan Pembaharuan Feromon Lokal (Local Pheromon Updating Rule)**. Kemudian, jika setiap semut telah mengunjungi semua simpul dan kembali lagi ke simpul asal, maka akan diperoleh lebih dari satu kemungkinan solusi jarak tempuh karena ada lebih dari satu semut yang melintasi setiap sisi dalam graf. Keadaan dimana setiap semut telah mengunjungi setiap simpul dalam graf dan kembali lagi ke simpul asal dinamakan satu iterasi. Pada strategi ACO ini, digunakan lebih dari satu kali iterasi. Pada akhir setiap iterasi akan dihasilkan lebih dari satu kemungkinan solusi. Langkah selanjutnya adalah memilih solusi yang terbaik sejauh ini, yaitu solusi dengan jarak tempuh minimum. Kemudian, tambahkan feromon pada lintasan terpendek yang dihasilkan dari solusi tersebut. Fungsi untuk menambahkan feromon ini dinyatakan dalam **Aturan Pembaharuan Feromon Global (Global Pheromon Updating Rule)**. Tujuan penambahan feromon global ini adalah untuk mendorong semut – semut pada iterasi berikutnya agar memilih lintasan terpendek yang telah dicapai sejauh ini. Selanjutnya, jalankan metode di atas untuk setiap iterasi. Setelah iterasi yang terakhir selesai dijalankan, pilihlah jarak tempuh minimum dari berbagai kemungkinan solusi yang dihasilkan pada setiap iterasi. Jarak tempuh minimum itulah yang merupakan solusi atas masalah TSP.

#### Fungsi dan Parameter

Berikut akan dijelaskan fungsi – fungsi dan parameter yang digunakan dalam Strategi ACO ini.

##### 3.1. Aturan Kemungkinan Solusi

Kemungkinan simpul  $s$  akan dipilih oleh semut  $k$  yang sedang berada pada simpul  $r$  dinyatakan dalam perhitungan matematika secara heuristik sebagai berikut:

$$s = \begin{cases} \arg \max_{u \in M_k} \{ [\tau(r,u)] \cdot [\eta(r,u)]^\beta \} & \text{jika } q < q_0 \text{ (Eksplorasi)} \\ S & \text{jika } q \geq q_0 \text{ (Eksplorasi)} \end{cases}$$

### Persamaan 1

Keterangan :

$\tau(r,u)$  : jumlah feromon pada sisi dari simpul r ke simpul s.

$\eta(r,u)$  : (panjang sisi dari simpul r ke simpul s)<sup>-1</sup>

$\beta$  : parameter perbandingan jumlah feromon relatif terhadap jarak (merupakan parameter yang telah ditentukan sebelumnya)

$M_k$  : himpunan yang berisi simpul – simpul yang telah dikunjungi oleh semut

$U$  : simpul yang berada dalam  $M_k$

$q$  : bilangan random

$q_0$  : parameter perbandingan eksploitasi terhadap eksplorasi

$S$  : simpul berikutnya yang dipilih berdasarkan persamaan (2).

Eksplorasi : semut mengeksplorasi simpul yang telah dikunjungi sebelumnya

Eksplorasi : semut mengeksplorasi simpul yang belum pernah dikunjungi sebelumnya.

Persamaan (2) diambil jika semut harus mengeksplorasi simpul yang belum pernah dikunjungi sama sekali.

Kemungkinan simpul s akan dipilih oleh semut k yang sedang berada pada simpul r dinyatakan dalam perhitungan matematika secara heuristik sebagai berikut :

$$p_k(r,s) = \begin{cases} \frac{[\tau(r,s)] \cdot [\eta(r,s)]^\beta}{\sum_{u \in M_k} [\tau(r,u)] \cdot [\eta(r,u)]^\beta} & \text{jika } s \in M_k \\ 0 & \text{jika } s \notin M_k \end{cases}$$

### Persamaan 2

Keterangan :

$\tau(r,s)$  : jumlah feromon pada sisi dari simpul r ke simpul s.

$\eta(r,s)$  : (panjang sisi dari simpul r ke simpul s)<sup>-1</sup>

$\beta$  : parameter perbandingan jumlah feromon relatif terhadap jarak (merupakan parameter yang telah ditentukan sebelumnya)

$\tau(r,u)$  : jumlah feromon pada sisi dari simpul r ke simpul s.

$\eta(r,u)$  : (panjang sisi dari simpul r ke simpul s)<sup>-1</sup>

$\beta$  : parameter perbandingan jumlah feromon relatif terhadap jarak (merupakan parameter yang telah ditentukan sebelumnya)

$M_k$  : himpunan yang berisi simpul – simpul yang telah dikunjungi oleh semut

$u$  : simpul yang berada dalam  $M_k$

### 3.2. Aturan Pembaharuan Lokal

Jumlah feromon  $\tau(r,s)$  akan berubah sesuai dengan aturan dibawah ini :

$$\tau(r,s) \leftarrow (1 - \rho) \times \tau(r,s) + \rho \times \Delta\tau(r,s)$$

Keterangan :

$\tau(r,s)$  : jumlah feromon pada sisi dari simpul r ke simpul s.

$\rho$  : parameter lokal feromon yang hilang

$\Delta\tau(r,s)$  : jumlah total feromon pada sisi dari simpul r ke s

### 3.3. Aturan Pembaharuan Feromon Global

Jumlah feromon  $\tau(r,s)$  akan berubah sesuai dengan aturan dibawah ini :

$$\tau(r,s) \leftarrow (1 - \alpha) * \tau(r,s) + \alpha * \Delta\tau(r,s)$$

### Persamaan 3

Keterangan :

$\tau(r,s)$  : jumlah feromon pada sisi dari simpul r ke simpul s.

$\alpha$  : parameter global feromon yang hilang

$\Delta\tau(r,s)$  : 1 / jarak tempuh lintasan terpendek sejauh ini (*best solution so far*) jika simpul r dan simpul s ada dalam lintasan tersebut.

## 4. Algoritma

*/\* fase inisialisasi \*/*

For tiap pasangan (r,s) do

• (r,s) := •<sub>0</sub>

End-for

For k:=1 to m do

set rk1 menjadi simpul awal untuk semut k

$Jk(rk1) := \{1, \dots, n\} - rk1$   
*/\* Jk(rk1) adalah himpunan titik yang harus dikunjungi semut k \*/*

rk:= rk1

*/\* rk adalah simpul tempat semut berada \*/*

End-for

*/\* fase pemilihan jalur (tur).*

```

    Jalur yang dipilih disimpan
dalam Tourk */
For i:=1 to n do
  If i<n then
    For k:=1 to m do
      Pilih simpul selanjutnya
      berdasarkan Eq. (3) dan Eq. (1)
      Jk(sk) := Jk(rk) - sk
      Tourk(i) := (rk ,sk)
    End-for
  Else
    For k:=1 to m do
      /* dalam loop ini semua semut
      kembali ke titik awal*/
      sk := rk1
      Tourk(i) := (rk ,sk)
    End-for
  End-if

  /* fase pembaharuan secara local,
  jumlah pheromone diperbaharui
  menggunakan Eq. (5)*/
  For k:=1 to m do
    •(rk ,sk) := (1-•)•(rk ,sk) + ••0
    rk := sk /* New city for ant k
  */
  End-for
End-for

/* fase pembaharuan secara global,
jumlah pheromone diperbaharui*/
For k:=1 to m do
  Hitung Lk
  /* Lk adalah panjang tur yang
  sudah dilakukan oleh semut k*/
End-for
Hitung Lbest

/* Perbaharui simpul-simpul yang ada
pada LBest menggunakan Eq. (4)*/
For tiap simpul(r,s)
   $\tau(rk ,sk) := (1-\alpha)\tau(rk ,sk) + \alpha$ 
  (Lbest)-1
End-for

If (End_kondisi = True)
then Cetak jalur terpendek dari Lk
else kembali ke fase pemilihan
jalur(tur)

```

### 5. Analisis Strategi ACO

Pada saat pertama kali seekor semut memilih simpul yang akan dilintasi pada iterasi pertama, sebenarnya strategi ACO memanfaatkan pendekatan dengan strategi *greedy*. Hal ini dikarenakan pada status awal, harga feromon sama untuk setiap sisi sehingga dengan sendirinya pemilihan simpul tergantung dari panjang sisi ke simpul tersebut.

Parameter  $\alpha, \beta, \rho$ , sebaiknya diisi dengan :

$$\alpha : 0 < \alpha < 1$$

$$\beta : 1 < \beta < 5$$

$$\rho = \pm 0.5$$

Parameter feromon lokal senilai  $0 < \alpha < 1$  dimaksudkan untuk menghindari akumulasi feromon yang tidak terbatas pada sisi tersebut.

Pada tahap awal, jumlah feromon tidak boleh diinisialisasi dengan 1 karena akan menonaktifkan fungsi feromon pada strategy ACO ini.

Selanjutnya, pemilihan jumlah semut sebaiknya kurang dari atau sama dengan jumlah simpul. Dan sebaiknya penempatan awal semut adalah satu semut untuk satu simpul. Hal ini untuk menjamin adanya lintasan awal dari simpul tersebut pada iterasi pertama.

Untuk permasalahan dengan jumlah simpul yang banyak, maka sebaiknya digunakan pendekatan heuristik lain yang bertujuan memilih simpul – simpul apa saja yang akan harus dikunjungi lebih dulu. Simpul – simpul pilihan tersebut disimpan dalam sebuah list. Setelah selesai mengunjungi simpul – simpul tersebut, barulah mengunjungi simpul – simpul lainnya dengan pendekatan ACO.

### 6. Kesimpulan

1. Algoritma ini tidak bisa memperkirakan jumlah iterasi yang terbaik. Karena jika jumlah iterasi terlalu kecil, maka hasil yang didapat akan tidak akurat, sebaliknya jika jumlah iterasi terlalu besar, maka nilai lintasan terbaik semakin lama semakin besar dan tidak semakin kecil.
2. Algoritma ini cocok untuk jumlah simpul yg sangat banyak, karena algoritma ini menempatkan semut pada setiap simpul, sehingga eksplorasi semakin cepat.
3. Algoritma ini biasanya menempatkan satu semut pada satu kota karena lebih optimal untuk setiap iterasi(jumlah iterasi akan dikalikan dengan jumlah semut).
4. Kompleksitas algoritma ini adalah  $O(NC.n^2.m)$  dimana  
 NC : jumlah iterasi (*Number Of Loops*).  
 n: jumlah simpul.  
 m: jumlah semut

### 7. Daftar Pustaka

1. Dorigo Marco and Gambardella Luca Maria, *Ant Colony System : A Cooperative Learning Approach to the Travelling Salesman*

*Problem*, Université Libre de Bruxelles, 1996.

2. Kennedy James, Eberhart Russell C., and Shi Yuhui, *Swarm Intelligence*, S. ???, Morgan Kaufmann Publishers, 2001.
3. Tarasewich Peter and McMullen Patrick R., *Swarm Intelligence*, on Communications of

the ACM vol 45, No. 8, S. 62-67, ACM Publications Office, 2002.