

## ANALISIS ALGORITMA SEMUT UNTUK PEMECAHAN MASALAH PENUGASAN

Zainudin Zukhri

Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia  
Kampus Terpadu UII Jl Kaliurang Km 14.5 Yogyakarta  
E-mail: zainudin@fti.uui.ac.id

### Abstrak

Algoritma semut sangat tepat digunakan untuk pemecahan masalah-masalah berbasis *Traveling Salseman Problem*. Karena MP bukan masalah yang berbasis *Traveling Salseman Problem*, maka penerapan algoritma semut untuk pemecahan *Traveling Salseman Problem* membutuhkan beberapa penyesuaian yang menyangkut konsep kota, rute, jarak antar kota dan intensitas jejak antar kota. Penelitian sebelumnya menunjukkan bahwa kinerja algoritma semut dalam pemecahan MP masih belum begitu bagus. Hal ini kemungkinan besar disebabkan oleh tingginya kompleksitas waktu algoritma semut. Untuk itu penelitian ini ditujukan untuk menganalisis algoritma semut dalam pemecahan masalah penugasan, sehingga bisa ditemukan penyebab rendahnya kinerja algoritma semut pada penelitian sebelumnya secara tepat. Analisis menunjukkan bahwa tingginya kompleksitas algoritma semut dalam pemecahan MP, disebabkan adanya penyesaian yang menyangkut konsep antar kota, baik yang menyangkut parameter jarak maupun yang menyangkut parameter intensitas jejak. Diharapkan temuan ini bisa dijadikan bahan pertimbangan bagi penelitian untuk menyederhanakan tingkat kompleksitas algoritma semut dalam pemecahan MP.

**Kata kunci:** algoritma semut, analisis algoritma, kompleksitas, masalah penugasan

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Beberapa algoritma sudah dikembangkan untuk pemecahan masalah penugasan(MP), diantaranya adalah metode Hongaria[2], algoritma *simulated annealing*[5], *tabu search*[1] dan algoritma genetika[4]. Penelitian mengenai penerapan algoritma genetika dengan pendekatan yang berbeda juga dilakukan oleh Zukhri [9], yang juga pernah mencoba menerapkan algoritma semut (AS) untuk pemecahan masalah MP ini[8].

Walaupun penerapan AS pada penelitian sebelumnya telah menunjukkan kemampuannya untuk pemecahan MP dengan ruang pencarian yang cukup sempit, tetapi kalau dilihat dari segi beban komputasi ternyata masih cukup berat. Hal ini berarti kompleksitas waktu AS dalam pemecahan MP masih cukup tinggi.

### 1.2 Tujuan

Penelitian ini bertujuan untuk menganalisis AS dalam pemecahan MP, sehingga bisa ditemukan penyebab rendahnya kinerja AS pada penelitian sebelumnya secara tepat.

## 2. METODOLOGI PENELITIAN

Tahapan yang dilakukan dalam penelitian ini adalah:

1. Studi pustaka penelitian AS untuk pemecahan MP.
2. Analisis kompleksitas waktu AS untuk pemecahan MP.
3. Analisis ruang pencarian pada MP.

## 3. DASAR TEORI

### 3.1 Algoritma Semut

Secara umum langkah-langkah dalam AS adalah [3][8]:

- Langkah 1: a. Inisialisasi harga parameter-parameter algoritma.  
b. Inisialisasi kota pertama setiap semut.
- Langkah 2: Pengisian kota pertama ke dalam *tabu list*.
- Langkah 3: Penyusunan rute kunjungan setiap semut ke setiap kota.
- Langkah 4: a. Perhitungan panjang rute setiap semut.  
b. Pencarian rute terpendek.  
c. Perhitungan perubahan *harga intensitas jejak kaki semut antar kota*.
- Langkah 5: a. Perhitungan *harga intensitas jejak kaki semut antar kota* untuk siklus berikutnya.  
b. *Reset harga perubahan intensitas jejak kaki semut antar kota*.
- Langkah 6: Pengosongan *tabu list*, dan ulangi langkah 2 jika diperlukan.

### 3.2 Masalah Penugasan

MP adalah penjadwalan sumberdaya dan aktifitas[7]. Dari sudut pandang masalah optimasi, pemecahan MP bertujuan untuk menyusun pasangan sumberdaya dan aktifitas berdasarkan penugasan satu-ke-satu, sedemikian sehingga dapat menghemat total biaya yang dibutuhkan pada kasus minimasi atau mendapatkan keuntungan yang paling besar pada kasus maksimasi, sehingga secara manual untuk menemukan solusi harus diperiksa kombinasi

sebanyak faktorial dari banyak sumberdaya atau faktorial dari banyak aktifitas. Dalam MP, harus dilakukan penjadwalan terhadap  $n$  sumberdaya untuk menangani  $n$  aktifitas, sedemikian sehingga biaya yang dibutuhkan untuk semua aktifitas adalah minimum.

#### 4. ANALISIS AS UNTUK PEMECAHAN MP

##### 4.1 Modifikasi AS untuk Pemecahan MP

MP bukan merupakan masalah berbasis *Traveling Salesman Problem* (TSP), sehingga beberapa konsep dalam AS harus disesuaikan [8]. Solusi yang dicari dalam MP adalah pasangan sumberdaya dan aktifitas, sedangkan masalah dalam AS adalah total panjang rute tertutup yang dilakukan oleh semut.

Dalam konsep AS, koloni semut diperalat untuk menemukan rute terpendek pada sejumlah kota. Setiap semut secara acak memilih sebuah kota sebagai kota pertama. Kemudian berdasarkan probabilitas kunjungannya, setiap semut akan memilih kota lainnya sebagai kota kedua. Pemilihan kota selanjutnya dilakukan berdasarkan probabilitas kunjungannya, dan tidak boleh memilih kota yang pernah dikunjungi. Probabilitas untuk memilih kota berikutnya ini ditentukan berdasarkan sisa jejak semut pada jalur menuju kota-kota yang mungkin untuk dikunjungi, atau secara matematis berdasarkan persamaan (1). Demikian untuk seterusnya sampai semua kota dikunjungi sekali. Untuk setiap semut  $k$ , terdapat daftar kota yang sudah dikunjungi yang disebut dengan daftar  $tabu_k$ . Setiap semut, akan mempunyai panjang rute tertutup masing-masing, yang merupakan jumlah jarak kota pertama ke kota kedua, jarak kota kedua ke kota ketiga dan seterusnya sampai kota terakhir ditambah dengan jarak kota terakhir ke kota pertama. Solusi yang ditemukan adalah panjang rute tertutup paling pendek di antara panjang rute yang ditempuh semua semut.

Dalam menyelesaikan MP, 'perjalanan' semut harus disesuaikan. MP harus dimodelkan sedemikian sehingga menjadi masalah pencarian rute. Hal ini bisa dilakukan dengan menganggap koloni semut harus melakukan perjalanan dari sebuah kota menuju kota lainnya, menggunakan biro jasa transportasi. Setiap kota harus dikunjungi, dan tidak perlu kembali ke kota asal. Setiap semut tidak boleh menggunakan biro jasa yang sama untuk kota yang berbeda. Banyak kota dan banyak biro jasa transportasi ditentukan oleh banyak aktifitas dan banyak sumberdaya dalam matriks biaya. Banyak kelompok kota ditentukan berdasarkan mana yang lebih kecil antara banyak aktifitas dan sumberdaya. Yang lebih kecil akan menjadi banyak kelompok kota, sedangkan yang lebih besar akan menentukan banyak biro jasa.

Apabila dalam masalah TSP, kota dibentuk dengan nomor indeks kota-kota yang akan dikunjungi, maka dalam MP, kota dibentuk dengan pasangan nomor indeks kota dan nomor indeks biro

jasa perjalanan yang dipakai untuk mengunjungi kota tersebut.

Konsep jarak antar kota dalam AS juga harus disesuaikan. Jika dalam masalah TSP, jarak yang dimaksud adalah jarak riil antara kota yang satu dengan kota lainnya, maka dalam MP, pengertian jarak ini dapat dianalogikan dengan tarif biro jasa transportasi yang harus 'dibayar' oleh semut. Ongkos transportasi ini tidak bergantung pada jarak riil ke kota tersebut, tetapi justru hanya tergantung pada kota yang akan dituju, dari manapun berangkatnya. Jika dalam masalah TSP, total jarak yang dimaksud adalah jarak total yang harus ditempuh semut, maka dalam pemecahan MP pengertian jarak ini berubah menjadi total biaya transportasi yang harus dikeluarkan oleh semut.

Adanya keterlibatan biro jasa dalam pendekatan AS untuk MP ini, dengan sendirinya juga mengubah konsep jejak antar kota. Jika dalam masalah TSP jejak antar kota membentuk suatu bidang, sementara jejak antar kota dalam MP akan membentuk suatu ruang. Dengan sendirinya hal ini akan mempengaruhi kompleksitas AS. Pembahasan mengenai hal ini, membutuhkan suatu penelitian tersendiri.

##### 4.2 Kompleksitas AS untuk Pemecahan MP

Untuk pemecahan MP, masukan AS adalah [C] (matriks biaya) termasuk  $n_R$  (banyak sumberdaya), dan  $n_A$  (banyak aktifitas). Beberapa parameter algoritma, yaitu  $Q$  (tetapan siklus-semut),  $\alpha$  (tetapan pengendali intensitas jejak semut),  $\beta$  (tetapan pengendali visibilitas),  $\eta_{ij}$  (visibilitas antar kota= $1/d_{ij}$ ),  $m$  (banyak semut),  $\rho$  (tetapan penguapan jejak semut) dan  $NC_{max}$  (jumlah siklus maksimum) bersifat tetap selama algoritma dijalankan, sedangkan  $\tau_{ij}$  (intensitas jejak semut antar kota) akan selalu diperbaharui harganya pada setiap siklus algoritma mulai dari siklus pertama ( $NC=1$ ) sampai tercapai siklus maksimum ( $NC=NC_{max}$ ) atau sampai terjadi konvergensi. Intensitas jejak semut antar kota perlu diinisialisasi dengan harga tertentu. Biasanya harga yang dipakai adalah bilangan positif kecil dengan perubahan harga intensitas jejak semut antar kota sama dengan nol [8].

Kompleksitas langkah pertama AS untuk pemecahan MP dinyatakan sebagai  $O((n_R \cdot n_A)^2 + m)$ , hal ini dapat dilihat pada algoritma dalam langkah ini adalah:

```
t:=0;
NC:=0;
for i:=1 to nR do
  for j:=1 to nA do
    for i':=1 to nR do
      for j':=1 to nA do
        begin
           $\tau_{i'j',ij}(t) := c;$ 
           $\Delta\tau_{i'j',ij} := 0;$ 
        end;
      Tempatkan m semut pada (nRx nA) 'kota'
```

Kompleksitas langkah kedua AS untuk pemecahan MP dinyatakan sebagai  $O(m)$ , hal ini dapat dilihat pada algoritma dalam langkah ini adalah:

```
s:=1;
for k:=1 to m do
  Masukkan 'kota' awal semut ke-k dalam
  daftar tabuk(s);
```

Kompleksitas langkah ketiga AS untuk pemecahan MP dinyatakan sebagai  $O(\min(n_R, n_A) \cdot m)$ , hal ini dapat dilihat pada algoritma dalam langkah ini adalah:

```
repeat
  s:=s+1;
  for k:=1 to m do
    begin
      Pilih kota i'j' dengan
      probabilitas berdasarkan
      persamaan(1);
      Pindahkan semut ke-k menuju kota
      i'j';
      Masukkan kota i'j' dalam daftar
      tabuk(s);
    end;
  until daftar tabu penuh;
```

Kompleksitas langkah keempat AS untuk pemecahan MP dinyatakan sebagai  $O((n_R \cdot n_A)^2 \cdot m)$ , hal ini dapat dilihat pada algoritma dalam langkah ini adalah:

```
for k:=1 to m do
  Pindahkan semut ke-k dari tabuk(s)
  ke tabuk(1);
  Hitung panjang 'rute' yang
  ditempuh semut ke-k;
  Update 'Rute' Terpendek yang
  ditemukan;
  for i:=1 to nR do
    for j:=1 to nA do
      for i':=1 to nR do
        for j':=1 to nA do
          for k:=1 to m do
            begin
              Hitung  $\Delta\tau_{ij,i'j'}^k$  berdasarkan
              persamaan (2);
               $\Delta\tau_{ij,i'j'} := \Delta\tau_{ij,i'j'} + \Delta\tau_{ij,i'j'}^k$ ;
            end;
```

Kompleksitas langkah kelima AS untuk pemecahan MP dinyatakan sebagai  $O((n_R \cdot n_A)^2)$ , hal ini dapat dilihat pada algoritma dalam langkah ini adalah:

```
for i:=1 to nR do
  for j:=1 to nA do
    for i':=1 to nR do
      for j':=1 to nA do
         $\tau_{ij,i'j'}(t+n) = \rho \cdot \tau_{ij,i'j'}(t) + \Delta\tau_{ij,i'j'}$ ;
t:=t+ nR;
NC:=NC+1;
for i:=1 to nR do
  for j:=1 to nA do
    for i':=1 to nR do
      for j':=1 to nA do
         $\Delta\tau_{ij,i'j'} := 0$ ;
```

Kompleksitas langkah keenam AS untuk pemecahan MP dinyatakan sebagai  $O(\min(n_R, n_A) \cdot m)$ , hal ini dapat dilihat pada algoritma dalam langkah ini adalah:

```
if (NC < NCMAX) and (not(kondisi stagnasi))
then
begin
  Kosongkan semua daftar tabu;
  (* ada m tabu, setiap tabu berisi 'kota'
  yang sebanyak sumberdaya atau banyak
  aktifitas (mana yang lebih kecil) *)
  goto langkah 2;
end else
begin
  Cetak rute terpendek;
  Selesai;
end;
```

Probabilitas untuk mengunjungi kota i'j' dari kota ij dihitung berdasarkan persamaan berikut:

$$p_{ij,i'j'}^k = \frac{[\tau_{ij,i'j'}]^\alpha \cdot [\eta_{ij,i'j'}]^\beta}{\sum_{k' \in \{MIN(N_R, N_A) - \text{tabu}_k\}} [\tau_{ij,k'}]^\alpha \cdot [\eta_{ij,k'}]^\beta}$$

untuk  $i'j' \in \{MIN(N_R, N_A) - \text{tabu}_k\}$   
dan ... (1)

$p_{ij,i'j'}^k = 0$ , untuk  $i'j'$  lainnya

dengan  $\eta_{ij,i'j'} = \frac{1}{c_{i'j'}}$  pada kasus minimasi atau

$\eta_{ij,i'j'} = c_{i'j'}$  pada kasus maksimasi.

Sedangkan perubahan harga intensitas jejak kaki semut antar kota dihitung berdasarkan persamaan perubahan ini adalah:

$$\Delta\tau_{ij,i'j'} = \sum_{k=1}^m \Delta\tau_{ij,i'j'}^k$$

dengan  $\Delta\tau_{ij,i'j'}^k$  adalah perubahan harga intensitas jejak kaki semut antar kota setiap semut.

$$\Delta\tau_{ij,i'j'}^k = \frac{Q}{L_k},$$

untuk  $(ij, i'j') \in$  kota asal dan kota tujuan dalam  $\text{tabu}_k$   
dan ... (2)

$\Delta\tau_{ij,i'j'}^k = 0$ , untuk  $(ij, i'j')$  lainnya

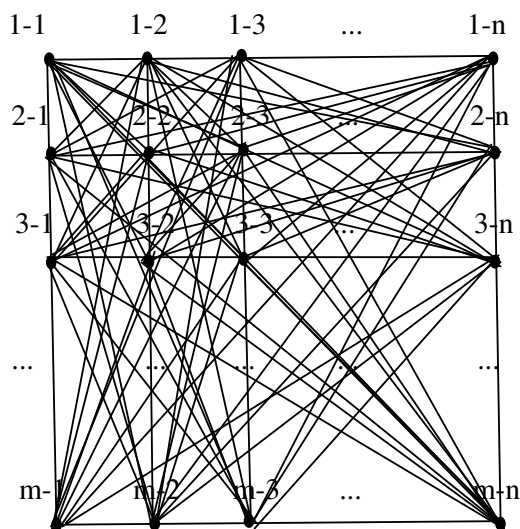
dengan  $L_k$  adalah panjang 'rute' yang dibentuk oleh semut ke-k atau pada pemecahan MP ini sama dengan total biaya yang dikeluarkan pada kasus minimasi atau total keuntungan yang didapat pada kasus maksimasi yang dibentuk oleh semut ke-k.

Keenam langkah AS untuk pemecahan MP tersebut paling banyak akan diulang sebanyak  $NC_{max}$  kali, sehingga kompleksitas AS secara keseluruhan dinyatakan sebagai  $O(NC_{max} \cdot (n_R \cdot n_A)^2 \cdot m)$ . Dengan

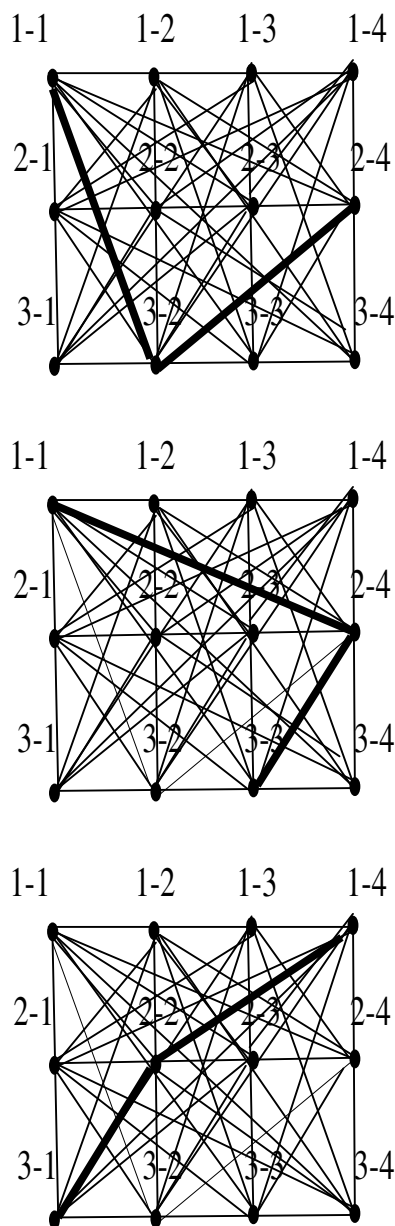
demikian dapat diketahui bahwa beban komputasi AS untuk pemecahan MP ini dipengaruhi oleh batas pengulangan maksimum yang diinginkan, banyak semut dan ukuran matriks biaya.

### 4.3 Ruang pencarian

Ruang pencarian solusi pada MP dapat direpresentasikan dalam bentuk jaringan terhubung penuh sebagaimana Gambar 1. Jaringan ini dibentuk dari matriks biaya  $[C]$  pada MP. Setiap elemen matriks biaya  $c_{ij}$  sekaligus membentuk titik-titik dalam jaringan beserta busur-busur penghubungnya. Banyak elemen menentukan banyak titik, sedangkan harga elemen menentukan nilai busur ke titik tersebut. Setiap titik diberi nama dengan kombinasi indeks elemen sumber daya dan indeks elemen aktifitas. Dengan representasi ruang pencarian sebagai jaringan terhubung penuh ini, jelaslah bahwa pencarian solusi pada MP adalah pencarian busur-busur yang menghubungkan titik-titik tertentu dalam jaringan, sedemikian sehingga indeks elemen sumber daya dan indeks elemen aktifitas pembentuk titik-titik tersebut, masing-masing hanya boleh dipakai sekali.



**Gambar 1.** Jaringan terhubung penuh sebagai representasi ruang pencarian pada MP



**Gambar 2.** Contoh solusi MP dalam bentuk jaringan untuk ukuran matriks biaya 4x3 berturut-turut dengan solusi (1-1, 3-2, 2-4), (1-1, 2-4, 3-3), dan (3-1, 2-2, 1-4)

Berdasarkan representasi ruang pencarian sebagaimana Gambar 1 dapat diketahui hanya sebagian kecil busur saja dalam jaringan yang akan membentuk solusi. Hal ini dapat sebagaimana Gambar 2. Dengan kata lain banyak busur solusi jauh lebih kecil dari pada banyak busur pada jaringan. Inilah penjelasan mengapa kompleksitas waktu AS sebanding dengan kuadrat dari ukuran matriksnya sebagaimana telah dibahas pada bagian sebelumnya. Wajarlah jika pada saat pengujian,

untuk menemukan solusi pada matriks biaya dengan ukuran  $15 \times 10$  AS membutuhkan waktu selama 10 detik. Beban komputasi akan terasa jika ukuran matriks biaya diperbesar. Dengan demikian untuk memperkecil ruang pencarian, pendekatan AS dalam pemecahan MP masih perlu ditinjau kembali.

## 5. KESIMPULAN

- a. Kompleksitas AS untuk pemecahan MP adalah  $O(NC_{max} \cdot (n_R \cdot n_A)^2 \cdot m)$ . Hal ini menunjukkan bahwa beban komputasi AS untuk pemecahan MP dipengaruhi oleh batas pengulangan maksimum yang diinginkan, banyak semut dan ukuran matriks biaya.
- b. Pengujian algoritma dengan ukuran matriks biaya  $15 \times 10$  yang telah dilakukan memakan waktu yang relatif cukup lama sangat wajar mengingat besarnya kompleksitas AS untuk pemecahan MP cukup tinggi.
- c. Perlu dilakukan penelitian lebih lanjut agar ruang pencarian dalam penerapan AS untuk pemecahan MP menjadi lebih sempit sehingga mengurangi beban komputasi.

## Daftar Pustaka

- [1] Boyce, J.F., Dimitripoulos, C.H.D., dan Taylor, J.G., (1995). Genet and Tabu search for combinatorial optimization problems, in Word conf. On Neural Network, WCCN'95, Washington.
- [2] Bronson, R., (1982). *Theory and Problems of Operations Research*, New York: McGraw Hill.
- [3] Dorigo, M., Maniezzo, V., dan Colorni, A., (1996). The Ant System: Optimization by a colony of cooperating agents, *IEEE Transaction on System, Man, and Cybernetics-Komponen B*, 26(1), 1-13.
- [4] Gen, M., dan Cheng, R., (1997). *Genetic Algorithms and Engineering Design*. New York: John Wiley & Sons, Inc.
- [5] Kapsalis, A., Smith, G.D., dan Rayward-Smith, V.J., (1994). *A unified approach to Tabu search, simulated annealing and genetic algorithms, in Application of Modern Heuristic Methods*, Mc.Graw Hill Editor.
- [6] Turban, E., (1995). *Decision Support And Expert Systems*, New Jersey: Prentice-Hall, Inc.
- [7] Taha, H.A., (1997). *Operations Research: An Introduction*, New York: MacMillan Publishing Company.
- [8] Zukhri, Z., (2003). Algoritma Semut untuk Pemecahan Masalah Penugasan, *Jurnal Media Informatika, Jurusan Teknik Informatika Fakultas Teknologi Industri UII*, 1(2), 1-10.
- [9] Zukhri, Z., (2004). Pemecahan Masalah Penugasan dengan Algoritma Genetika, *Prosiding Seminar Nasional Aplikasi Teknologi Informasi (SNATI 2005) - Jurusan Teknik Informatika Fakultas Teknologi Industri UII*, j51.