

BAB II

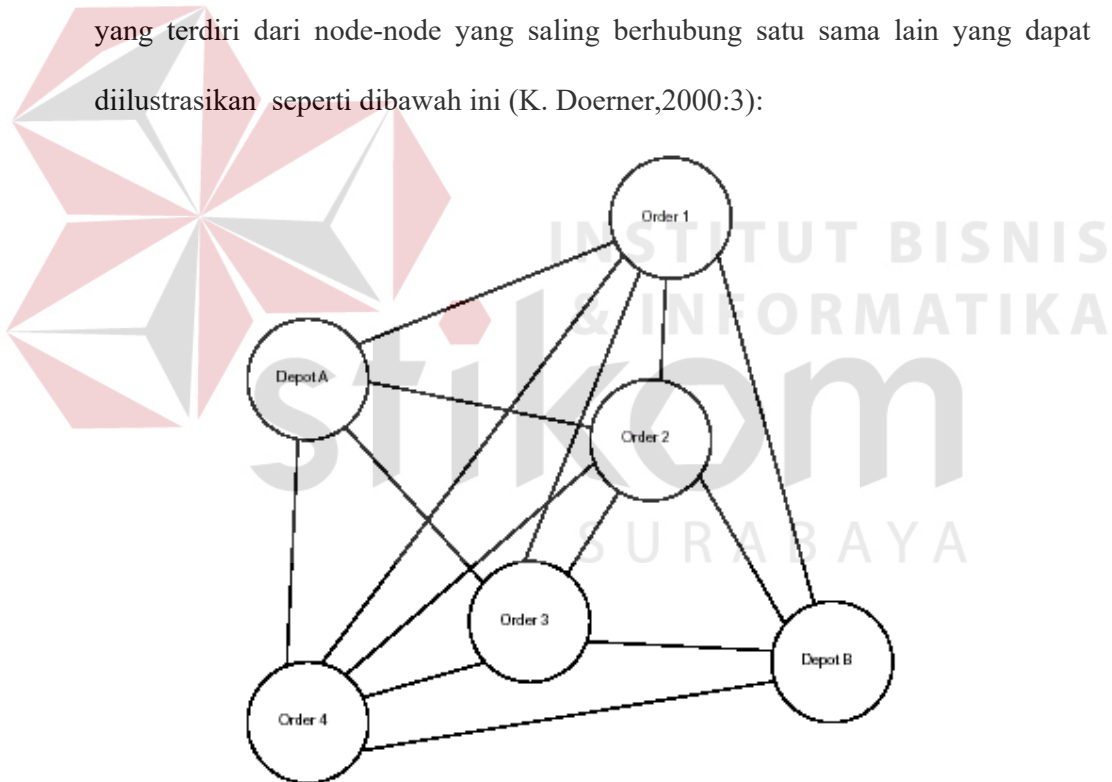
LANDASAN TEORI

2.1 Landasan Teori Tentang Permasalahan

Merupakan landasan teori yang terdapat pada permasalahan yang dibahas.

2.1.1 Masalah transportasi

Masalah transportasi dalam dunia distribusi pada perusahaan yang menyangkut pendistribusian barang kepada customer yang berupa barang pesanan ataupun bahan baku yang akan dijadikan menjadi bahan jadi. Sistem transportasi yang terdiri dari node-node yang saling berhubung satu sama lain yang dapat diilustrasikan seperti dibawah ini (K. Doerner,2000:3):



Gambar 2.1. Grafik dengan 2 gudang dan 4 pesanan

$J = \{1, \dots, n\}$ menunjukkan rangkaian pesanan, dan $D = \{1, \dots, m\}$ menunjukkan rangkaian pusat distribusi, dimana gudang dari truk. Kemudian, masalah pada TA ini digambarkan pada gambar diatas bahwa grafik $G = \{V, A, d\}$, dimana $V =$

$\{v_1, v_2, v_3, \dots, v_n, v_{n+1}, \dots, v_{n+m}\}$ adalah rangkaian vektor. Vertek v_1 sampai v_n menunjukkan pesanan 1 sampai n (kita arahkan bagian dari V sebagai V_o), vertek v_{n+1} sampai v_{n+m} menunjukkan gudang 1 sampai m (kita arahkan bagian dari V sebagai V_d). $A = \{(v_i, v_j); i \neq j, V \times V \setminus V_d \times V_d\}$ adalah rangkaian dari *arcs*. *Arcs* (v_i, v_j) menggambarkan perpindahan kendaraan kosong yang dibutuhkan diantara pesanan i dan j , dan besarnya non-negatif jarak d_{ij} .

Berdasarkan grafik, untuk semua constraint bahwa semua node pesanan $v_i \in V_o$ adalah tepat yang akan didatangi dan tiap putaran mengikutkan tepat satu node gudang $v_j \in V_d$.

Ciri-ciri khusus masalah transportasi adalah (Tjutju Tarliah Dimiyati, 2002:129):

1. Terdapat sejumlah sumber dan sejumlah tujuan tertentu.
2. Kuantitas komoditas atau barang yang didistribusikan dari setiap sumber dan yang diminta oleh setiap tujuan, besarnya tertentu.
3. Komoditas yang dikirim atau diangkut dari suatu sumber ke suatu tujuan, besarnya sesuai dengan permintaan dan atau kapasitas sumber.
4. Ongkos pengangkutan komoditas dari suatu sumber ke suatu tujuan besarnya tertentu.

2.1.2 Teori Ant Colony

Menurut K. Doerner dalam (2000:3) menyebutkan bahwa optimasi ant colony (Ant Colony Optimization-ACO) merupakan sistem cerdas yang diinspirasi oleh perilaku semut dan koloninya, yang digunakan untuk menyelesaikan permasalahan optimasi diskrit. Sistem ACO pertama kali diperkenalkan oleh

Marco Dorigo (1992), dan dinamakan sistem semut (Ant System-AS) yang pertama kali digunakan untuk menyelesaikan masalah perjalanan salesman (*The Travelling Salesman Problem-TSP*). AS sebagai hasil penelitian dalam bidang komputasi cerdas yang digunakan untuk mendekati permasalahan optimasi kombinator.

2.1.3 Sistem Semut (Marco Dorigo,1999:3)

Sebuah analogi terhadap bagaimana koloni semut berfungsi memberikan sebuah definisi baru model perhitungan, yang dinamakan Sistem Semut (*Ant System-AS*). Sistem Semut dikemukakan sebagai sebuah pendekatan baru untuk optimasi kombinatorial. Karakteristik utama dari model ini adalah feedback positif, perhitungan distribusi dan menggunakan konsepsi greedy heuristic. Feedback positif untuk menghitung penemuan solusi yang baik secara cepat, perhitungan distribusi untuk menghindari pertemuan sebelum waktunya dan greedy heuristic untuk membantu menemukan solusi yang cocok dalam mengatur sejak awal proses pencarian.

Karakteristik algoritma heuristic yang dapat digunakan untuk memecahkan perbedaan masalah optimasi kombinatorial adalah sebagai berikut:

1. Serbaguna, yaitu dapat digunakan untuk versi serupa dari masalah yang sama; sebagai contoh, sebuah perluasan pendekatan masalah dari masalah perjalanan salesman untuk masalah perjalanan salesman asymmetric.
2. Handal, maksudnya dapat digunakan hanya dengan perubahan minimal untuk masalah optimasi kombinatorial yang lain seperti quadratic assignment problem (QAP) dan masalah job-shop scheduling.

3. Pendekatan berdasarkan jumlah pertumbuhan, Ini menarik karena membolehkan eksploitasi feedback positif sebagai sebuah mekanisme pencarian dan juga membuat sistem menerima untuk implementasi paralel.

Sifat ini mengimbangi dengan fakta bahwa, untuk beberapa aplikasi, sistem semut dapat menjadi outperformed khususnya pada algoritma. Ini adalah sebuah masalah kerjasama dengan pendekatan populer yang lain seperti Simulated Annealing (SA) dan Tabu Search (TS), dibandingkan dengan sistem semut.

Aktivitas pencarian semut adalah sebagai agen yang kemampuannya sangat mendasar sekali dalam meniru perilaku semut sesungguhnya. Penelitian perilaku semut sesungguhnya sangat mengilhami kerja manusia. Salah satu masalah yang dipelajari oleh ethologist adalah memahami bagaimana binatang yang hampir buta seperti semut dapat mengatur untuk menentukan jalur rute terpendek ke sumber makanan dan jalan kembalinya. Telah ditemukan bahwa media yang digunakan untuk komunikasi informasi diantara individu-individunya berkaitan dengan jalur, dan digunakan untuk menentukan kemana dia pergi adalah jejak feromon. Semut yang berjalan meninggalkan beberapa feromon dalam jumlah yang berbeda ditanah kemudian jalur ditandai dengan substansi ini. Sementara semut yang terisolasi bergerak secara random, semut yang menjumpai jalur yang ditandai sebelumnya dapat menemukannya dan memutuskan dengan probabilitas yang tinggi untuk mengikuti jalur tersebut, kemudian meningkatkan jalur dengan feromonnya sendiri. Perilaku kolektif yang muncul adalah bentuk perilaku autocatalytic dimana lebih banyak semut yang mengikuti sebuah jalur, semakin menarik jalur tersebut dan menjadikannya semakin diikuti. Prosesnya kemudian digolongkan sebagai positive-feedback loop, dimana probabilitas dengan jalur

yang dipilih oleh semut meningkat dengan sejumlah semut yang sebelumnya telah memilih jalur yang sama.

2.1.4 Perilaku Semut

Untuk dapat membayangkan algoritma koloni semut bekerja, ada baiknya melihat bagaimana secara nyata semut dan koloninya menjalani kehidupannya. Jika kita amati, perjalanan semut dari sarang ke tempat makanan dan sebaliknya dari tempat makanan kembali ke sarang, cenderung membentuk iring-iringan.

Rute dari iring-iringan tersebut menyerupai sebuah garis, seperti diperlihatkan pada Gambar 2.1. Cara utama yang digunakan oleh para semut untuk membentuk dan memelihara garis ini yaitu dengan apa yang dikenal dengan istilah jejak feromon (*pheromone trail*). Feromon adalah zat yang dikeluarkan oleh semut untuk mendeteksi dan merespon keberadaan dari semut lain. Pada saat berjalan, feromon yang tercium lebih kuat secara probabilistik akan lebih menarik semut lain untuk mengikutinya, daripada yang tercium lemah. Dengan demikian semut-semut itu cenderung mengikuti teman terdekatnya, dan membentuk iring-iringan semut.



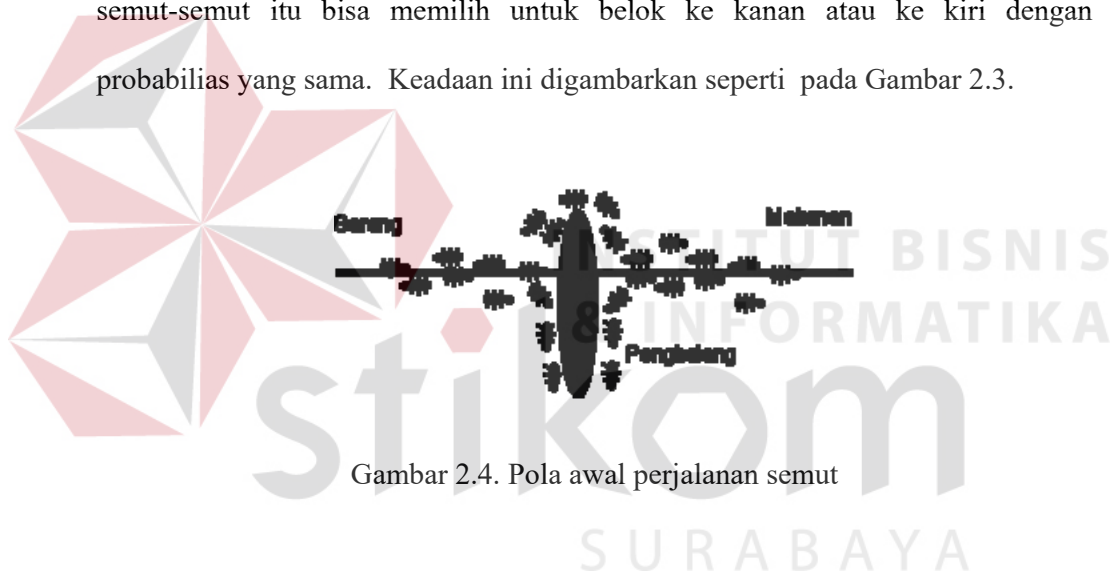
Gambar 2.2. perilaku semut

Dasar perilaku semut diatas dapat digunakan untuk menjelaskan bagaimana mereka dapat menemukan rute terpendek dan menyambung kembali rute yang terputus akibat adanya penghalang. Iring-iringan semut ketika diberi penghalang, dapat dilihat pada Gambar 2.2.



Gambar 2.3. Iring-iringan semut diberi penghalang

Ketika ada penghalang, maka semut yang berada di depan penghalang tidak dapat melanjutkan dan mengikuti jejak feromon. Yang terjadi kemudian, semut-semut itu bisa memilih untuk belok ke kanan atau ke kiri dengan probabilitas yang sama. Keadaan ini digambarkan seperti pada Gambar 2.3.



Gambar 2.4. Pola awal perjalanan semut

Selanjutnya menarik untuk dicatat, bahwa semut yang kebetulan memilih sisi lintasan yang lebih pendek segera menemukan dan menyambung kembali jejak feromon yang telah terputus dibandingkan dengan mereka yang memilih sisi lintasan yang lebih panjang. Dengan kejadian ini, pada sisi lintasan yang lebih pendek akan terbentuk sejumlah feromon yang lebih banyak akibat banyaknya semut yang memilih lintasan tersebut. Proses ini diberi istilah *autocatalytic process* dengan umpanbalik positif. Segera setelah itu semua semut mengikuti

lintasan tersebut yang ternyata sebagai lintasan terpendek. Seperti terlihat pada gambar 2.5.



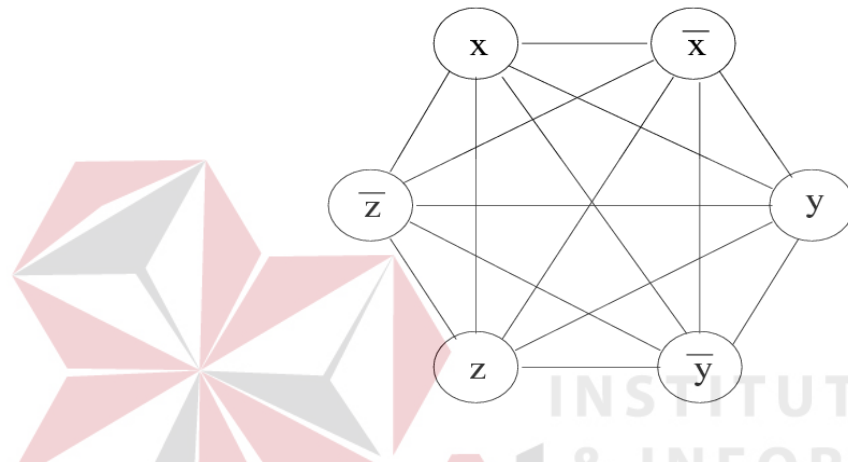
Gambar 2.5. Pola akhir perjalanan semut

Aspek yang sangat menarik dari proses autocatalytic ini adalah penemuan rute terpendek di sekitar penghalang merupakan kejadian yang muncul oleh adanya interaksi antara bentuk penghalang dan perilaku distribusi semut. Dengan asumsi semua semut bergerak dengan kecepatan yang hampir sama dan tumpukan jejak feromon rata-rata sama, maka waktu yang dibutuhkan akan lebih cepat jika dipilih rute terpendek, jika dibandingkan apabila mereka memilih lintasan yang lebih jauh.

2.1.5. Peranan Feromon

CSP dinyatakan dengan grafik lintasan $G = (C, \mathcal{L})$ yang didefinisikan sedemikian hingga simpul-simpul (komponen) merupakan pasangan (variabel, nilai) dan persambungan antar simpul yang secara penuh menghubungkan simpul-simpul (lihat Gambar 2.6). Solusinya adalah berupa rangkaian atau urutan n simpul yang dijadikan sejumlah n variabel. Semut-semut menyusun solusi secara probabilitas dengan melakukan pemilihan simpul dari tetangganya yang paling mungkin. Tetangga yang paling mungkin dari semut k didefinisikan sebagai

kumpulan pasangan (variabel, nilai) sedemikian hingga variabel tersebut belum memberikan nilai. Tetangga yang dipilih menerapkan kendala yaitu simpul yang memiliki variabel yang sama bukanlah pada solusi yang sama. Titik kunci dari algoritma ini adalah pemilihan elemen-elemen grafik (komponen dan persambungan) dari feromon.



Gambar 2.6. Grafik lintasan CSP dengan variabel biner

a. Feromon di komponen

Hal pertama yang mungkin adalah meletakkan feromon pada komponen. Dalam kasus ini sejumlah feromon secara proporsional merupakan keinginan dari pencarian solusi. Dengan pemilihan ini, aturan transisi keadaan ACS digunakan oleh semut k (dinamakan aturan *pseudo-random-proportional*) yang dinyatakan dalam :

$$s = \begin{cases} \arg \max_{u \in J_k} \{\tau(u)\}, & \text{jika } q \leq q_0 \\ z, & \text{lainnya} \end{cases} \quad p_k(z) = \begin{cases} \frac{\tau(z)}{\sum_{u \in J_k} \tau(u)}, & \text{jika } z \in J_k \\ 0, & \text{lainnya} \end{cases} \quad (2.1)$$

dimana,

s adalah simpul berikutnya, $\tau(u)$ adalah nilai feromon pada simpul u , J_k adalah kumpulan simpul dari tetangga dari semut k yang paling mungkin, q adalah variabel random yang terdistribusi serbasama pada $[0,1]$, q_0 ($0 \leq q_0 \leq 1$) berupa parameter dari algoritma dan z dipilih secara probabilitas p_k .

Aturan pembaharuan feromon dinyatakan dengan persambungan berikut :

1. Pembaharuan feromon tahap demi tahap :

$$\forall \text{ semut } k, \forall s \in S_k : \tau(s) \leftarrow (1 - \rho) \cdot \tau(s) + \rho \cdot \tau_0,$$

2. Pembaharuan feromon offline : $\forall s \in S_{opt} : \tau(s) \leftarrow (1 - \alpha) \cdot \tau(s) + \alpha \cdot g(S_{opt})$,

dimana S_k adalah solusi yang dibangun oleh semut k , S_{opt} sebagai solusi terbaik yang diperoleh, $\rho, \alpha, \in [0,1]$, $\tau_0 \in \mathbb{R}^+$ adalah parameter-parameter algoritma dan $0 < g(S_{opt}) < +\infty$ adalah fungsi monotonik yang tidak menghasilkan penurunan dari kualitas solusi S_{opt} . Untuk keadaan seperti terlihat pada Gambar 2.1, yang bersesuaian antar simpul dan penempatannya adalah :

$$(x,1) \leftrightarrow x, (x,0) \leftrightarrow \bar{x}, (y,1) \leftrightarrow y, (y,0) \leftrightarrow \bar{y}, (z,1) \leftrightarrow z, (z,0) \leftrightarrow \bar{z}.$$

b. Feromon di persambungan

Alternatif lain dari penempatan feromon adalah bukan pada komponennya tetapi pada persambungannya. Jadi jumlah feromon yang terletak pada persambungan dua simpul secara proporsional merupakan keuntungan dari pemilihan dua alternatif pencarian solusi. Untuk semut k yang bergerak dari simpul r ke simpul s , aturan (2.1) menjadi :

$$s = \begin{cases} \arg \max_{u \in J_k} \{\tau(r, u)\}, & \text{jika } q \leq q_0 \\ z, & \text{lainnya} \end{cases}$$

$$p_k(r, z) = \begin{cases} \frac{\tau(r, z)}{\sum_{u \in J_k} \tau(r, u)}, & \text{jika } z \in J_k \\ 0, & \text{lainnya} \end{cases} \quad (2.2)$$

dimana,

$\tau(r, u)$ adalah level komponen pada persambungan $(r; u)$. Aturan pembaharuan feromon merupakan pengembangan dari aturan sebelumnya.

c. Feromon di persambungan (dengan Jumlah)

Kemungkinan terakhir adalah dengan cara melakukan perhitungan ketergantungan dari apa yang telah dilakukan sebelumnya. Untuk melakukan hal ini, Persamaan (2.1) diubah menjadi sebagai berikut :

$$s = \begin{cases} \arg \max_{u \in J_k} \left\{ \sum_{w \in S_k} \tau(w, u) \right\}, & \text{jika } q \leq q_0 \\ z, & \text{lainnya} \end{cases}$$

$$p_k(r, z) = \begin{cases} \frac{\sum_{w \in S_k} \tau(w, z)}{\sum_{u \in J_k} \sum_{w \in S_k} \tau(w, u)}, & \text{jika } z \in J_k \\ 0, & \text{lainnya} \end{cases} \quad (2.3)$$

dimana,

S_k menyatakan kumpulan simpul dari bagian lintasan yang sedang dilalui.

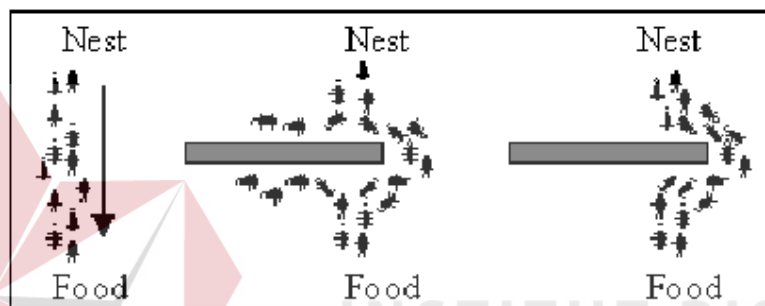
Persamaan (2.3) menyatakan jumlah feromon dari persambungan antar simpul. Aturan pembaharuan feromon diubah agar persambungan tiap pasangan simpul dari solusi terbaru (ter-update).

Dengan demikian, terdapat tiga penempatan yang penting dari feromon sehingga dihasilkan grafik lintasan untuk pencarian solusi terbaik berdasarkan kendala yang diberikan. Yaitu feromon diletakkan di komponen, di persambungan dan di persambungan dalam jumlah feromon tertentu. Solusi terbaik inilah yang diharapkan sebagai penyelesaian dari kasus pencarian rute terpendek.

2.1.6 Artificial Ant Colony System

Karakteristik utama dalam ant system (K. Doerner,2000:3):

- Tidak memiliki control pusat.
- Memiliki kemampuan auto-organization.
- Terdapat keadaan biasa yang objektif bagi semua individu/agen dalam sistem.
- Berdasarkan pada tugas suatu divisi.

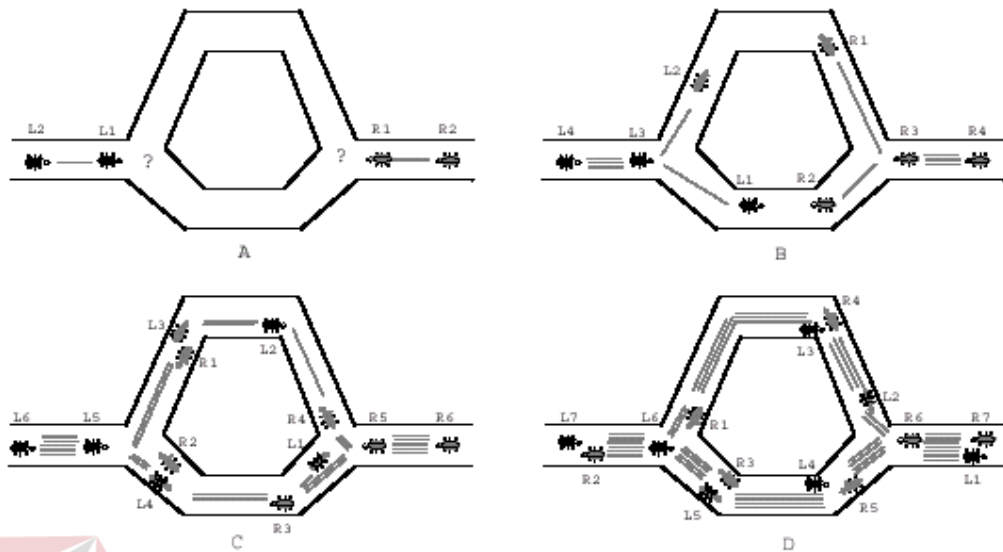


Gambar 2.7 semut mencari jalan terpendek

Pemisalan alami dimana algoritma semut didasarkan adalah koloni semut. Semut asli mampu menemukan rute terpendek dari sumber makanan ke sarang mereka, tanpa menggunakan isyarat visual dengan memanfaatkan informasi feromon. Ketika berjalan semut menempatkan feromon di tanah dan mengikuti feromon yang sebelumnya telah diletakkan oleh semut lain menurut probabilitasnya. Cara semut memanfaatkan feromon untuk menemukan rute terpendek antara dua titik diperlihatkan pada Gambar 2.8.

Pada Gambar 2.8.a. : Semut-semut datang pada titik keputusan dimana mereka harus memutuskan hendak berbelok ke kiri atau ke kanan. Karena mereka tidak memiliki petunjuk mana pilihan yang terbaik, mereka memilih secara random. Berdasarkan hal ini bisa diperkirakan bahwa secara rata – rata separuh

dari semut-semut tersebut memutuskan untuk berbelok ke kiri dan separuh yang lain berbelok ke kanan. Hal ini terjadi juga kepada semut yang bergerak dari kiri ke kanan (mereka yang memiliki nama depan L) dan mereka yang bergerak dari kanan ke kiri (yang bernama depan R). Gambar 2.8.b. dan 2.8.c. menunjukkan apa yang terjadi berikut : Misalkan semua semut berjalan dengan kecepatan yang kurang lebih sama. Jumlah garis tanda kurang lebih sebanding dengan jumlah feromon yang telah ditempatkan semut di tanah. Karena rute paling bawah adalah lebih pendek daripada yang paling atas, secara rata – rata lebih banyak semut yang akan mendatangnya, dan oleh karenanya feromon terkumpul dengan cepat. Setelah periode lintasan pendek perbedaan jumlah feromon pada dua rute cukup besar sehingga dapat mempengaruhi keputusan semut yang baru datang ke dalam sistem (hal ini diperlihatkan pada Gambar 2.8.d.). Sejak saat ini, semut akan lebih memilih secara probabilitas rute yang paling bawah, karena pada titik keputusan mereka merasakan jumlah feromon yang sangat besar pada rute paling bawah. Hal ini memicu peningkatan, dengan pengaruh positive feedback, jumlah semut yang memilih rute yang paling bawah dan rute yang terpendek. Dalam waktu singkat semua semut akan menggunakan rute terpendek.



Gambar 2.8. Cara semut menggunakan feromon untuk menemukan rute terpendek diantara 2 titik

Keterangan Gambar 2.8 :

- Semut datang di titik keputusan
- Beberapa semut memilih rute teratas dan beberapa semut lainnya memilih rute terbawah. Pemilihan dilakukan secara random.
- Karena semut bergerak dengan kecepatan konstan, semut yang memilih rute terbawah dan terpendek, mencapai titik keputusan yang berlawanan lebih cepat dibandingkan dengan yang memilih rute yang teratas dan terjauh.
- Feromon terkumpul dengan nilai yang lebih tinggi pada rute terpendek. Jumlah garis tanda kurang lebih sebanding dengan jumlah feromon yang diletakkan oleh semut.

Perilaku semut asli diatas terinspirasi *ant system*, sebuah algoritma dimana sekumpulan semut buatan bekerjasama untuk menyelesaikan masalah

dengan saling bertukar informasi lewat feromon yang ditempatkan pada busur grafik. Ide dasar Ant Colony System (ACS) adalah memiliki sekumpulan agen, disebut dengan *semut*, mencari secara sejajar penyelesaian yang baik untuk rute terpendek dan bekerja sama melalui media tak langsung yaitu feromon dan komunikasi global. Memory mengambil bentuk feromon yang ditempatkan semut pada busur, sementara informasi heuristic diberikan oleh panjang busur.

Bagaimana ACS bekerja dapat digambarkan sebagai berikut: awalnya m semut ditempatkan pada n kota yang dipilih sesuai dengan beberapa initialization rule (misalnya secara random). Masing – masing semut membangun sebuah tur dengan berulang – ulang menggunakan sebuah stochastic greedy rule (state transition rule). Sementara membangun tur mereka, semut juga mengubah sejumlah feromon pada busur yang dikunjungi dengan menggunakan local updating rule. Sekali semua semut mengakhiri tur mereka, sejumlah feromon pada busur diubah kembali (dengan menggunakan global updating rule). Semut dikendalikan, dalam membangun tur mereka, dengan informasi heuristic (mereka lebih memilih busur yang pendek) dan dengan informasi feromon: sebuah busur dengan jumlah feromon yang tinggi merupakan pilihan yang lebih diinginkan. Pheromon updating rule didesain sehingga mereka cenderung untuk memberikan feromon lebih untuk busur yang harus dikunjungi oleh semut.

2.1.7 Ant Colony Algorithm

Diketahui satu set n kota, rute terpendek dapat ditetapkan sebagai masalah untuk menemukan jarak terdekat. Misalkan d_{ij} adalah jarak dari jalur

antara kota i dan j , dalam kasus Euclidian TSP, d_{ij} adalah jarak antara i dan j (dengan nilai $d_{ij} = [(x_i - x_j)^2 + (y_i - y_j)^2]^{1/2}$).

Sebagai contoh diberikan oleh sebuah grafik (N, E) , dimana N = set dari kota, E = set simpul antar kota.

Misalkan $b_i(t) (i=1, 2, \dots, n)$ adalah jumlah semut pada kota i pada saat t dan $m = \sum_{i=1}^n b_i(t)$ adalah jumlah total semut. Tiap semut adalah agen sederhana dengan karakteristik sebagai berikut :

1. Jika memilih kota tujuan dengan suatu probabilitas itu adalah sebuah fungsi dari jarak kota dan jumlah dari tampilan jejak pada busur penghubung.
2. Untuk memaksa semut untuk melakukan legal tur, transisi ke kota yang telah dikunjungi tidak diperbolehkan sampai tur selesai (yang dikontrol oleh tabu list).
3. Ketika tur selesai, semut meninggalkan substansi yang disebut *trail* (jalur) pada setiap busur (i, j) yang dikunjungi.

Misalkan $\tau_{ij}(t)$ adalah intensitas jalur pada busur (i, j) pada waktu t . Masing – masing semut pada waktu t memilih kota berikutnya, dimana itu terjadi pada waktu $t+1$. Oleh karena itu jika memanggil sebuah iterasi dari algoritma AS, m gerakan dibawa oleh m semut dalam interval $(t, t+1)$ kemudian setiap n iterasi dari algoritma (dimana disebut sebagai satu cycle) setiap semut harus sudah menyelesaikan sebuah tur. Dalam hal ini intensitas jalur diperbarui menurut persamaan berikut:

$$\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + \Delta \tau_{ij} \quad (2.4)$$

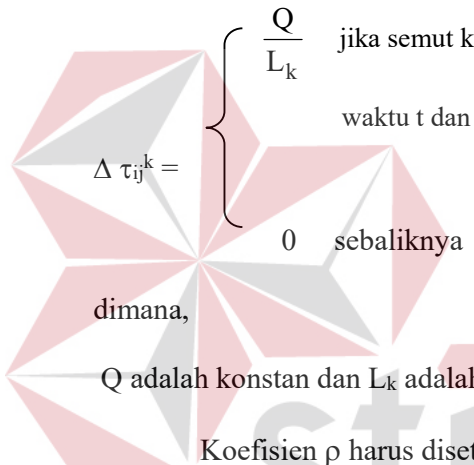
dimana,

ρ adalah koefisien seperti $(1-\rho)$ mewakili penguapan (evaporation) dari jalur antara waktu t dan $t+n$,

$$\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k \quad (2.5)$$

dimana,

$\Delta \tau_{ij}^k$ adalah jumlah per unit dari panjang substansi jalur (peromon pada semut sebenarnya) yang diletakkan pada busur (i,j) oleh semut ke- k antara waktu t dan $t+n$, yang diberikan oleh :



$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{jika semut ke-}k \text{ menggunakan busur } (i,j) \text{ dalam tournya (diantara} \\ & \text{waktu } t \text{ dan } t+n) \\ 0 & \text{sebaliknya} \end{cases} \quad (2.6)$$

dimana,

Q adalah konstan dan L_k adalah panjang tour dari semut ke- k .

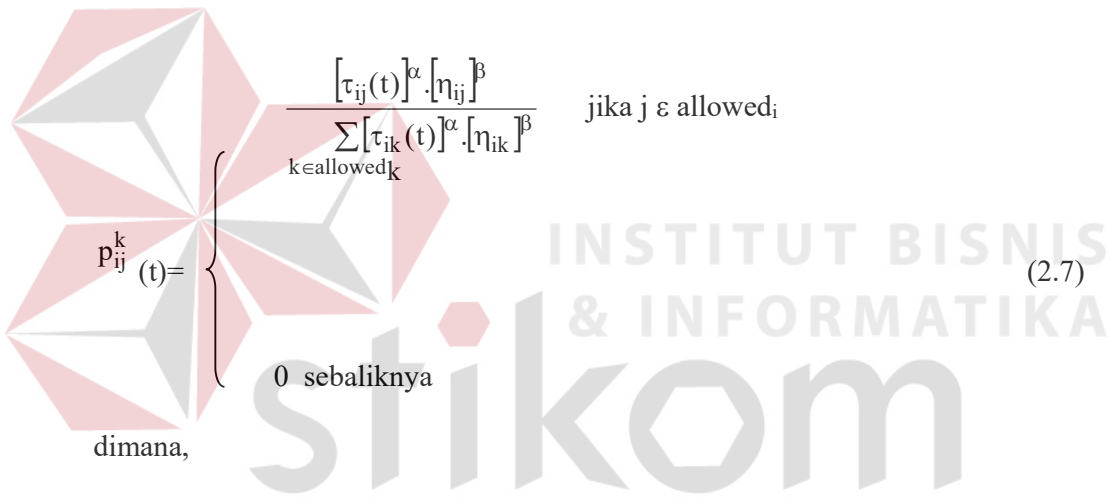
Koefisien ρ harus diset dengan nilai <1 untuk menghindari penimbunan jalur yang tak terbatas. Diset intensitas jalur pada waktu 0, $\tau_{ij}(0)$, ke nilai konstan positif c .

Dengan tujuan memenuhi batasan dimana semut mengunjungi semua n kota yang berbeda, dihubungkan dengan masing – masing semut sebuah struktur data bernama *tabu list*, yang menyimpan kota yang sudah dikunjungi sampai waktu t dan melarang semut untuk mengunjunginya lagi sebelum n iterasi (satu tur) telah diselesaikan. Ketika satu tur telah selesai, *tabu list* digunakan untuk menghitung hasil semut saat ini (yaitu jarak dari jalur yang diikuti semut). *Tabu list* lalu dikosongkan dan semut bebas untuk memilih lagi. Definisi $tabu_k$ adalah vektor yang tumbuh secara dinamik dimana mengandung *tabu list* dari semut ke-

k , kumpulan tabu_k dihasilkan dari elemen-elemen tabu_k , dan $\text{tabu}_k(s)$ adalah elemen ke- s dari daftar (yaitu kota ke- s yang telah dikunjungi oleh semut ke- k pada tur saat ini).

Disebutkan bahwa visibility η_{ij} adalah sejumlah $1/d_{ij}$. Jumlah ini tidak berubah selama menjalankan AS, sebagai kebalikan dari jalur dimana malah berubah sesuai dengan persamaan sebelumnya (Persamaan (2.4)).

Didefinisikan probabilitas transisi dari kota i ke kota j untuk semut ke- k sebagai :



$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{k \in \text{allowed}_k} [\tau_{ik}(t)]^\alpha \cdot [\eta_{ik}]^\beta} & \text{jika } j \in \text{allowed}_i \\ 0 & \text{sebaliknya} \end{cases} \quad (2.7)$$

dimana,

$\text{allowed}_k = \{N - \text{tabu}_k\}$ dan dimana α dan β adalah parameter yang mengontrol kepentingan relatif jalur melawan visibility. Oleh karena itu probabilitas transisi adalah sebuah pertukaran antara visibility (yang mana dikatakan bahwa kota terdekat harus dipilih dengan probabilitas tinggi, hingga diterapkan sebuah *greedy constructive heuristic*) dan intensitas jalur pada waktu t (yang disebutkan bahwa jika pada busur (i,j) terdapat lalu lintas yang ramai maka busur tersebut sangat diinginkan, hingga diterapkan proses autocatalytic).

Algoritma ant cycle ditetapkan sebagai berikut :

Pada saat $t=0$, fase inisialisasi dilakukan pada waktu semut diposisikan pada kota-kota yang berbeda dan inisial values $\tau_{ij}(0)$ untuk intensitas jalur diset pada busur-busur. Elemen pertama dari tabu list masing – masing semut diset sama dengan kota awalnya. Kemudian setiap semut bergerak dari kota i ke kota j memilih kota tujuan dengan probabilitas yang berupa fungsi (dengan parameter α dan β , lihat Persamaan (2.7.)) dari 2 perhitungan yang diinginkan. Yang pertama, jalur $\tau_{ij}(t)$, memberikan informasi tentang berapa banyak semut sebelumnya telah memilih busur (i,j) yang sama; yang kedua, visibility η_{ij} , mengatakan bahwa semakin dekat sebuah kota maka semakin diinginkan untuk dipilih. Jelasnya, diset nilai $\alpha=0$, level jalur tidak diperhitungkan, dan algoritma stochastic greedy dengan titik awal yang dikalikan diukur.

Setelah n iterasi semua semut telah menyelesaikan tur, dan tabu list-nya akan penuh. Pada keadaan ini untuk setiap semut k nilai L_k akan dihitung dan nilai dari $\Delta \tau_{ij}^k$ diperbarui sesuai dengan Persamaan (2.6). Dan rute terpendek yang ditemukan oleh semut (yaitu $\min L_k, k=1, \dots, m$) disimpan dan semua tabu list dikosongkan. Proses ini teriterasi sampai counter tur mencapai maksimum (user-defined) jumlah cycles NC_{MAX} , atau semua semut membuat tur yang sama yang disebut sebagai *stagnation behaviour* karena ia merupakan suatu situasi dimana algoritma berhenti mencari solusi alternatif.

2.1.8 Algoritma ant cycle

Step 1 : Inisialisasi

Set $t:=0$ {t adalah time counter}

Set $n:c=0$ {NC adalah Cycles counter}

Untuk setiap busur (i,j) set sebuah initial value $\tau_{ij}(t)=c$ untuk intensitas jalur dan $\Delta \tau_{ij}=0$

Letakkan m semut pada simpul n

Step 2 : Set $s:=1$ {s adalah tabu list index}

For $k:=1$ to m do

Letakkan kota awal dari semut ke- k dalam $\text{tabu}_k(s)$

Step 3 : Repeat until tabu list penuh {langkah ini akan diulang $(n-1)$ kali}

Set $s:=s+1$

For $k:=1$ to m do

Pilih kota j untuk bergerak, dengan probabilitas $p_{ij}^k(t)$ diberikan oleh pers(4) {pada saat t semut ke- k berada pada kota $i=\text{tabu}_k(s-1)$ }

Pindahkan semut ke- k ke kota j

Sisipkan kota j dalam $\text{tabu}_k(s)$

Step 4 : For $k:=1$ to m do

Pindahkan semut ke- k dari $\text{tabu}_k(n)$ ke $\text{tabu}_k(1)$

Hitung jarak L_k dari tur yang dibuat oleh semut ke- k

Perbarui rute terpendek yang sudah ada

Untuk setiap busur (i,j)

For $k:=1$ to m do

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{jika } (i,j) \in \text{tur di buat oleh } \text{tabu}_k \\ 0 & \text{sebaliknya} \end{cases}$$

$$\Delta \tau_{ij} = \Delta \tau_{ij} + \Delta \tau_{ij}^k ;$$

Step 5 : Untuk setiap busur (i,j) hitung $\tau_{ij}(t+n)$ sesuai dengan persamaan

$$\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + \Delta \tau_{ij}$$

Set $t:=t+n$

Set $NC:=NC+1$

Untuk setiap busur (i,j) set $\Delta \tau_{ij} := 0$

Step 6 : If ($NC < NC_{MAX}$) and (not stagnation behaviour)
 then
 Kosongkan semua tabu list
 Goto step 2
 else
 Print rute terpendek
 Stop

Kompleksitas dari algoritma ant cycle adalah $O(NC \cdot n^2 \cdot m)$ jika algoritma dihentikan setelah NC cycles. Pada faktanya step 1 adalah $O(n^2 + m)$, step 2 adalah $O(m)$, step 3 adalah $O(n^2 \cdot m)$, step 4 adalah $O(n^2 \cdot m)$, step 5 adalah $O(n^2)$, step 6 adalah $O(n \cdot m)$. Sejak ditemukan relasi linear antara jumlah kota dan jumlah semut terbaik, kompleksitas algoritma adalah $O(NC \cdot n^3)$.

Terdapat dua algoritma AS yang lain yang disebut sebagai algoritma *ant-density* dan *ant-quantity*. Mereka berbeda dalam hal memperbarui jalur. Dalam dua model ini masing – masing semut meletakkan jalurnya pada setiap step, tanpa menunggu akhir dari tur. Dalam model *ant density* sebuah kuantitas Q dari jalur ditinggalkan pada busur(i,j) setiap kali semut berangkat dari i ke j ; dalam model *ant-quantity* seekor semut berangkat dari i ke j meninggalkan sebuah kuantitas Q/d_{ij} dari jalur pada busur(i,j) setiap kali ia berangkat dari i ke j . Oleh karena itu, dalam model *ant-density* terdapat:

$$\Delta \tau_{ij}^k = \begin{cases} Q & \text{jika semut ke-}k \text{ berangkat dari } i \text{ ke } j \text{ antara waktu } t \text{ dan } t+1 \\ 0 & \text{sebaliknya} \end{cases} \quad (2.8)$$

dan dalam model *ant-quantity* kita mempunyai :

$$\Delta \tau_{ij}^k = \begin{cases} Q/d_{ij} & \text{jika semut ke-k berangkat dari } i \text{ ke } j \text{ antara waktu } t \text{ dan } t+1 \\ 0 & \text{sebaliknya} \end{cases} \quad (2.9)$$

Dari definisi ini maka jelas bahwa peningkatan dalam jalur pada busur (i,j) ketika seekor semut berangkat dari i ke j adalah berdiri sendiri dari d_{ij} dalam model *ant-density*, sementara ia berbanding terbalik dengan d_{ij} dalam model *ant-quantity* (yaitu busur terdekat dibuat lebih diinginkan oleh semut dalam model *ant-quantity*).

