

BAB III

ALGORITMA *BEE COLONY OPTIMIZATION* UNTUK MENYELESAIKAN MASALAH PENDISTRIBUSIAN BARANG

Bab ini membahas tentang masalah pendistribusian barang dan model optimisasinya. Selanjutnya, akan dibahas pula tentang cara Algoritma *Bee Colony Optimization* dalam menyelesaikan masalah pendistribusian barang.

3.1. Masalah Pendistribusian Barang

Masalah pendistribusian barang adalah salah satu masalah yang banyak ditemui dalam kehidupan sehari-hari. Dalam pendistribusian barang, distributor harus memasarkan barangnya dengan cara mengunjungi setiap konsumen untuk mendistribusikan barangnya dan memenuhi permintaan konsumen. Tujuan penyelesaian masalah distribusi adalah meminimumkan biaya pendistribusian dan jarak tempuh. Dalam hal ini, perlu untuk mengetahui rute terpendek yang bisa dilalui agar biaya pendistribusian yang harus dikeluarkan dapat diminimumkan. Dalam penelitian ini, rute tersebut haruslah bisa melewati setiap konsumen tepat satu kali dan kembali ke tempat semula sehingga jarak tempuhnya minimum.

Masalah pendistribusian barang dapat dipandang sebagai masalah *Travelling Salesman Problem* dimana distributor dianggap sebagai *salesman* yang harus berkeliling dari tempat asalnya, mengunjungi beberapa kota, lalu kembali ke tempat asalnya. Pada bagian selanjutnya dari bab ini akan dibahas tentang penurunan model optimisasi dari masalah pendistribusian barang.

3.2. Model Optimisasi Masalah Pendistribusian Barang

Masalah pendistribusian barang dapat dimodelkan sebagai model optimisasi dengan sebuah fungsi tujuan dan kendala. Asumsi yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Jarak tempuh dari lokasi i ke j sama dengan jarak tempuh dari lokasi j ke i
2. Hanya terdapat satu distributor

Untuk memodelkan masalah pendistribusian barang, terlebih dahulu didefinisikan himpunan, parameter, dan variabel keputusan. Misalkan I adalah himpunan kota dan d_{ij} adalah jarak tempuh dari kota i ke kota j . Variabel keputusan x_{ij} menentukan apakah distributor melakukan perjalanan dari kota i ke kota j atau tidak. Variabel keputusan tersebut didefinisikan sebagai berikut. Misal

$$x_{ij} = \begin{cases} 1, & \text{jika terdapat perjalanan dari } i \text{ ke } j, \\ 0, & \text{lainnya.} \end{cases}$$

Misal z adalah fungsi tujuan dari model optimisasi pendistribusian barang yang bertujuan untuk meminimumkan jarak tempuh dalam pendistribusian. Fungsi tujuan dari model optimisasi masalah pendistribusian barang adalah

Meminimumkan:

$$z = \sum_{i \in I} \sum_{j \in I} d_{ij} x_{ij}$$

Kendala dari model adalah batasan yang harus dipenuhi dalam menentukan rute yang dilalui. Kendala tersebut menentukan bahwa setiap lokasi hanya dapat dikunjungi tepat satu kali. Kendala yang pertama adalah kendala yang menjamin bahwa hanya ada satu jalur menuju setiap kota. Kendala ini diekspresikan dengan

$$\sum_{i \in I} x_{ij} = 1, \quad \forall j \in I$$

Kendala kedua menjamin bahwa jalur yang keluar dari setiap kota juga hanya ada satu. Kendala ini diekspresikan dengan

$$\sum_{j \in I} x_{ij} = 1, \quad \forall i \in I$$

Selengkapnya, model optimisasi pendistribusian barang adalah sebagai berikut:

Meminimumkan:

$$z = \sum_{i \in I} \sum_{j \in I} d_{ij} x_{ij}$$

dengan kendala:

$$\sum_{i \in I} x_{ij} = 1, \quad j \in I$$

$$\sum_{j \in I} x_{ij} = 1, \quad i \in I$$

$$x_{ij} \in \{0,1\}, \quad \forall i, j \in I$$

3.3. Algoritma *Bee Colony Optimization* untuk Menyelesaikan Masalah Pendistribusian Barang

Masalah pendistribusian barang termasuk dalam permasalahan *Nondeterministic Polynomial-Hard (NP-Hard)*. Hal ini berarti penyelesaian secara eksak tidak efisien untuk diterapkan karena akan membutuhkan waktu komputasi yang sangat lama. Pada penelitian ini, model pendistribusian barang akan diselesaikan menggunakan Algoritma *Bee Colony Optimization*. Algoritma ini dibuat berdasarkan perilaku koloni lebah dalam menemukan sumber makanan dan menyebarkan informasi mengenai sumber makanan kepada lebah lainnya yang berada di sarang. Dalam Algoritma *Bee Colony Optimization*, sumber makanan merepresentasikan kemungkinan solusi dari masalah. Nilai *fitness* merepresentasikan kualitas dari sebuah sumber makanan. Jumlah makanan yang ada di sumber makanan i direpresentasikan oleh P_{f_i} .

Berdasarkan tugasnya, lebah dapat dikelompokkan kedalam tiga kelompok yaitu *employed bees* (lebah pekerja), *onlooker bees* (lebah pencari), dan *scouts* (lebah pengintai). Lebah pekerja adalah lebah yang sudah mempekerjakan dirinya pada suatu sumber makanan. Mereka mengambil makanan dari sumber makanan dan menyimpannya di sarang lalu menari untuk mengajak lebah lainnya untuk ikut mengambil makanan di sumber makanannya. Tarian tersebut berisi informasi mengenai posisi dan kualitas sumber makanannya. Lebah pencari menyaksikan tarian dari lebah pekerja di dalam sarang. Mereka akan memilih lebah pekerja yang akan mereka ikuti dan berubah menjadi lebah pekerja. Setelah makanan di

sumber makanan tersebut habis, lebah pekerja bisa memilih untuk menjadi lebah lebah pencari atau lebah pengintai. Lebah pengintai adalah lebah bebas yang bertugas untuk mencari sumber makanan baru yang belum pernah dikunjungi sebelumnya. Setelah menemukan sumber makanan, mereka akan berubah menjadi lebah pekerja.

Algoritma *Bee Colony Optimization* mempunyai beberapa parameter, yaitu jumlah kota (N), peluang sebuah kota dikunjungi seekor lebah (λ) yang bernilai $[0,1]$, variabel biner yang menentukan pengaruh *arc fitness* (α), faktor skalar pengendali jarak antar sumber makanan (β) yang merupakan bilangan real $[0,\infty]$, dan faktor skalar pengendali panjang durasi tarian lebah (K) yang bernilai positif.

Secara lengkap, langkah kerja dari Algoritma *Bee Colony Optimization* adalah sebagai berikut:

1. Pembentukan rute awal

Pembentukan rute awal ini menggunakan metode *Nearest Neighbor*. Algoritma *Nearest Neighbor* dimulai pada suatu lokasi dan secara berulang mengunjungi lokasi terdekat dari lokasi saat ini hingga semua lokasi dikunjungi tepat satu kali. Sebagai contoh, akan dicari jarak tempuh terpendek dari 10 lokasi menggunakan metode *Nearest Neighbor* dari lokasi awal 1, mengunjungi setiap lokasi, lalu kembali ke lokasi asal. Jarak tempuh antar kota dapat dilihat dari tabel berikut:

Tabel 3.1 Jarak Tempuh Antara 10 Lokasi (dalam km)

	1	2	3	4	5	6	7	8	9	10
1	0	7	3	12	15	8	20	18	10	11
2	7	0	4	5	8	14	12	10	13	9
3	3	4	0	2	9	12	11	5	15	8
4	12	5	2	0	7	10	15	8	3	4
5	15	8	9	7	0	3	5	2	10	6
6	8	14	12	10	3	0	7	6	9	5
7	20	12	11	15	5	7	0	4	6	10
8	18	10	5	8	2	6	4	0	7	12
9	10	13	15	3	10	9	6	7	0	12
10	11	9	8	4	6	5	10	12	2	0

Penentuan jarak tempuh terpendek dengan metode *Nearest Neighbor* adalah sebagai berikut:

Dari lokasi 1, lokasi terdekat yang dapat dikunjungi adalah 3 dengan jarak tempuh 3 km.

1	3
---	---

Dari lokasi 3, lokasi terdekat yang dapat dikunjungi adalah 4 dengan jarak tempuh 2 km.

1	3	4
---	---	---

Dari lokasi 4, lokasi terdekat yang dapat dikunjungi adalah 9 dengan jarak tempuh 3 km.

1	3	4	9
---	---	---	---

Dari lokasi 9, lokasi terdekat yang dapat dikunjungi adalah 10 dengan jarak tempuh 2 km.

1	3	4	9	10
---	---	---	---	----

Dari lokasi 10, lokasi terdekat yang dapat dikunjungi adalah 6 dengan jarak tempuh 5 km.

1	3	4	9	10	6
---	---	---	---	----	---

Dari lokasi 6, lokasi terdekat yang dapat dikunjungi adalah 5 dengan jarak tempuh 3 km.

1	3	4	9	10	6	5
---	---	---	---	----	---	---

Dari lokasi 5, lokasi terdekat yang dapat dikunjungi adalah 8 dengan jarak tempuh 2 km.

1	3	4	9	10	6	5	8
---	---	---	---	----	---	---	---

Dari lokasi 8, lokasi terdekat yang dapat dikunjungi adalah 7 dengan jarak tempuh 4 km.

1	3	4	9	10	6	5	8	7
---	---	---	---	----	---	---	---	---

Dari lokasi 7, lokasi terdekat yang dapat dikunjungi adalah 2 dengan jarak tempuh 12 km.

1	3	4	9	10	6	5	8	7	2
---	---	---	---	----	---	---	---	---	---

Karena setiap lokasi sudah dikunjungi, maka dari lokasi 2 kembali lagi ke lokasi 1 dengan jarak tempuh 7 km.

1	3	4	9	10	6	5	8	7	2	1
---	---	---	---	----	---	---	---	---	---	---

Jadi, urutan lokasi yang dikunjungi dari rute tersebut adalah 1-3-4-9-10-6-5-8-7-2-1 dengan jarak tempuh $30 + 20 + 30 + 20 + 50 + 30 + 20 + 40 + 120 + 70 = 43$ km.

Selanjutnya, rute yang terbentuk dipandang sebagai kumpulan sumber makanan. Solusi dari tahap ini akan dioptimalkan di tahap berikutnya.

2. Tahap *Forage*

Pada tahap ini, lebah akan memilih sumber makanan mana yang akan mereka pilih. Pemilihan tersebut bergantung pada *arc fitness* dan jarak antar sumber makanan. *Arc fitness* dihitung untuk setiap kemungkinan lebah bergerak dari sumber makanan i ke sumber makanan j pada transisi (iterasi) ke- n , yaitu

$$\rho_{ij,n} = \begin{cases} \lambda, & j \in F_{i,n}, |A_{i,n}| > 1 \\ \frac{1 - \lambda |A_{i,n} \cap F_{i,n}|}{|A_{i,n} - F_{i,n}|}, & j \notin F_{i,n}, |A_{i,n}| > 1 \\ 1, & |A_{i,n}| = 1 \end{cases}$$

dengan:

λ : peluang dari sebuah kota yang dikunjungi seekor lebah

$A_{i,n}$: deretan sumber makanan yang belum dikunjungi yang dapat dicapai dari i pada transisi (iterasi) ke- n

$F_{i,n}$: himpunan sumber makanan yang dapat dikunjungi berdasarkan rekomendasi *preferred path* (0 atau 1).

Preferred path adalah rute yang sudah didapatkan pada transisi sebelumnya. Untuk transisi pertama, *preferred path* yang digunakan adalah rute yang dihasilkan dari metode *nearest neighbor*. Untuk transisi kedua dan selanjutnya, *preferred path* yang digunakan adalah rute yang dihasilkan pada transisi $n-1$.

Untuk memilih sumber makanan berikutnya, lebah harus mengikuti aturan. Besar kemungkinan lebah bergerak dari sumber makanan i ke sumber makanan j setelah transisi ke- n dihitung melalui persamaan berikut:

$$P_{ij,n} = \frac{[\rho_{ij,n}]^\alpha \left[\frac{1}{d_{ij}}\right]^\beta}{\sum_{j \in A_{i,n}} [\rho_{ij,n}]^\alpha \left[\frac{1}{d_{ij}}\right]^\beta}$$

dengan:

$\rho_{ij,n}$: *arc fitness* dari sumber makanan i ke sumber makanan j setelah transisi ke- n

d_{ij} : jarak antara sumber makanan i dengan sumber makanan j

$A_{i,n}$: deretan sumber makanan yang belum dikunjungi yang dapat dicapai dari i pada transisi ke- n

α : faktor skalar pengendali *arc fitness* (biner)

β : faktor skalar pengendali jarak antar sumber makanan

3. Tahap *Waggle Dance*

Setelah lebah mendapatkan rute lengkap (rute dari lokasi awal, mengunjungi setiap lokasi tepat satu kali, lalu kembali ke lokasi awal), lebah pekeja akan kembali ke sarang dan melakukan tarian untuk mengajak lebah pencari agar ikut mengambil makanan di sumber makanannya. Tarian lebah berlangsung dalam durasi tertentu. Durasi tarian lebah bergantung pada jumlah makanan yang ditemukan oleh lebah dari sumber makanan ke- i dan profitabilitas (kemampuan memperoleh keuntungan jika mengunjungi suatu sumber makanan) koloni lebah tersebut. Jumlah makanan dari suatu sumber makanan diekspresikan melalui persamaan berikut:

$$P_{f_i} = \frac{1}{L_i}$$

dengan L_i adalah panjang rute lengkap dari lebah i . P_{f_i} merepresentasikan kualitas makanan atau jumlah makanan yang dikumpulkan oleh lebah i . Semakin pendek jarak sumber makanan ke sarang, maka jumlah makanan yang dikumpulkan akan semakin banyak.

Profitabilitas koloni lebah diekspresikan melalui persamaan berikut:

$$P_{f_{colony}} = \frac{1}{N} \sum_{i=1}^N P_{f_i}$$

Lama durasi tarian lebah dinotasikan sebagai D_i dengan K adalah faktor skalar yang mengendalikan panjang durasi tarian. Lama durasi tarian lebah diekspresikan melalui persamaan berikut:

$$D_i = K \frac{P_{f_i}}{P_{f_{colony}}}$$

Setelah lebah pencari melihat tarian lebah pekerja, lebah pencari akan memilih sumber makanan mana yang akan dituju menggunakan probabilitas P_{follow} . Probabilitas P_{follow} disesuaikan mengikuti nilai profitabilitas lebah dan koloninya. Nilai P_{follow} dapat dilihat pada tabel berikut.

Tabel 3.2 Nilai P_{follow}

Nilai Probabilitas	P_{follow}
$P_{f_i} < 0.95 P_{f_{colony}}$	0.8
$0.95 P_{f_{colony}} \leq P_{f_i} < 0.975 P_{f_{colony}}$	0.2
$0.975 P_{f_{colony}} \leq P_{f_i} < 0.99 P_{f_{colony}}$	0.02
$0.99 P_{f_{colony}} \leq P_{f_i}$	0

Untuk lebih jelas, dapat dilihat contoh berikut. Seorang penjual buku harus mengunjungi 4 pelanggannya yang berada di kota yang berbeda-beda. Penjual buku berada di kota Basin (lokasi awal), pelanggannya berada di kota Mena, Kiln, Wald, dan Bon. Jarak tempuh antar kota dapat dilihat pada tabel berikut:

Tabel 3.3 Jarak Tempuh Antara 5 Kota (dalam km)

	Basin	Wald	Bon	Mena	Kiln
Basin	0	120	220	150	210
Wald	120	0	80	110	130
Bon	220	80	0	160	185
Mena	150	110	160	0	190
Kiln	210	130	185	190	0

Langkah pertama, akan dicari rute awal menggunakan metode *Nearest Neighbor*. Misal dengan menggunakan metode *Nearest Neighbor*, diperoleh solusi awal dengan rute Basin-Wald-Bon-Mena-Kiln-Basin dengan jarak tempuh sebesar $120 + 80 + 160 + 190 + 210 = 760$ km. Kemudian, rute awal akan dioptimalkan menggunakan Algoritma *Bee Colony Optimization*. Untuk langkah awal, misal ditetapkan parameter yang akan digunakan yaitu $N = 5$, $\lambda = 0.35$, $\alpha = 1$, $\beta = 1$, dan $K = 1$.

Hitung *Arc fitness* untuk $i = \text{Basin}$, $j = \text{Wald, Bon, Mena, dan Kiln}$ dan $n = 1$.

$$\rho_{\text{BasinWald},1} = \lambda = 0,35$$

$$\rho_{\text{BasinBon},1} = \frac{(1 - 0,35) \cdot 1}{3} = 0.2167$$

$$\rho_{\text{BasinMena},1} = \frac{(1 - 0,35) \cdot 1}{3} = 0.2167$$

$$\rho_{\text{BasinKiln},1} = \frac{(1 - 0,35) \cdot 1}{3} = 0.2167$$

Selanjutnya, hitung besar kemungkinan lebah bergerak dari sumber makanan i ke sumber makanan j setelah transisi (iterasi) ke- n .

$$P_{\text{BasinWald},1} = \frac{0.35^1 \cdot \left(\frac{1}{120}\right)^1}{0.0064} = \frac{0.00292}{0.00638} = 0.45728$$

$$P_{BasinBon,1} = \frac{0.2167^1 \cdot \left(\frac{1}{220}\right)^1}{0.0064} = \frac{0.00099}{0.00638} = 0.15443$$

$$P_{BasinMena,1} = \frac{0.2167^1 \cdot \left(\frac{1}{150}\right)^1}{0.0064} = \frac{0.00144}{0.00638} = 0.2265$$

$$P_{BasinKiln,1} = \frac{0.2167^1 \cdot \left(\frac{1}{210}\right)^1}{0.0064} = \frac{0.00103}{0.00638} = 0.16179$$

Karena $P_{BasinWald,1}$ memiliki nilai terbesar, maka dari Basin, penjual akan melakukan perjalanan ke Wald dengan jarak tempuh 120 km.

Hitung *Arc fitness* untuk $i = Wald, j = Bon, Mena, dan Kiln$ dan $n = 2$.

$$\rho_{WaldBon,2} = \lambda = 0,35$$

$$\rho_{WaldMena,2} = \frac{(1 - 0,35) \cdot 1}{2} = 0.325$$

$$\rho_{WaldKiln,2} = \frac{(1 - 0,35) \cdot 1}{2} = 0.325$$

Selanjutnya, hitung besar kemungkinan lebah bergerak dari sumber makanan i ke sumber makanan j setelah transisi ke- n .

$$P_{WaldBon,2} = \frac{0.35^1 \cdot \left(\frac{1}{80}\right)^1}{0.00983} = \frac{0.00438}{0.00983} = 0.44509$$

$$P_{WaldMena,2} = \frac{0.325^1 \cdot \left(\frac{1}{110}\right)^1}{0.00983} = \frac{0.00295}{0.00983} = 0.30058$$

$$P_{WaldKiln,2} = \frac{0.325^1 \cdot \left(\frac{1}{130}\right)^1}{0.00983} = \frac{0.0025}{0.00983} = 0.25434$$

Karena $P_{WaldBon,2}$ memiliki nilai terbesar, maka dari Wald, penjual akan melakukan perjalanan ke Bon dengan jarak tempuh 80 km.

Hitung *Arc fitness* untuk $i = Bon, j = Mena, dan Kiln$ dan $n = 3$.

$$\rho_{BonMena,3} = \lambda = 0,35$$

$$\rho_{BonKiln,3} = \frac{(1 - 0,35) \cdot 1}{1} = 0.65$$

Selanjutnya, hitung besar kemungkinan lebah bergerak dari sumber makanan i ke sumber makanan j setelah transisi ke- n .

$$P_{BonMena,3} = \frac{0.35^1 \cdot \left(\frac{1}{160}\right)^1}{0.00570} = \frac{0.00219}{0.00570} = 0.3837$$

$$P_{BonKiln,3} = \frac{0.65^1 \cdot \left(\frac{1}{185}\right)^1}{0.00570} = \frac{0.00351}{0.00570} = 0.6163$$

Karena $P_{BonKiln,3}$ memiliki nilai terbesar, maka dari Bon, penjual akan melakukan perjalanan ke Kiln dengan jarak tempuh 185 km.

Hitung *Arc fitness* untuk $i = Kiln$, $j = Mena$, dan $n = 4$.

$$P_{KilnMena,4} = 1$$

Selanjutnya, hitung besar kemungkinan lebah bergerak dari sumber makanan i ke sumber makanan j setelah transisi ke- n .

$$P_{KilnMena,4} = \frac{1^1 \cdot \left(\frac{1}{190}\right)^1}{0.00526} = \frac{0.00526}{0.00526} = 1$$

Karena $P_{KilnMena,4}$ memiliki nilai terbesar, maka dari Kiln, penjual akan melakukan perjalanan ke Mena dengan jarak tempuh 190 km.

Karena semua pembeli telah dikunjungi, maka penjual akan kembali ke Basin dengan jarak tempuh 150 km. Jadi, rute perjalanan yang terbentuk adalah Basin-Wald-Bon-Kiln-Mena-Basin dengan jarak tempuh $120 + 80 + 185 + 190 + 150 = 725$ km.