**LECTURE NOTES**

**Selected Topics in Computational Intelligence I**

**Session 3**

**Ant Colony Optimization**

**LEARNING OUTCOMES**

**Tujuan Instruksional Umum :**

1. *Mahasiswa mampu menjelaskan tentang metode Ant Colony Optimization (ACO)*

**Tujuan Instruksional Khusus :**

1. *Mahasiswa dapat menyebutkan konsep feromon dan ACO*
2. *Mahasiswa mampu menjelaskan tentang algoritma ACO*
3. *Mahasiswa mampu membuat aplikasi sederhana yang menerapkan konsep ACO.*

**3.1 Pendahuluan**

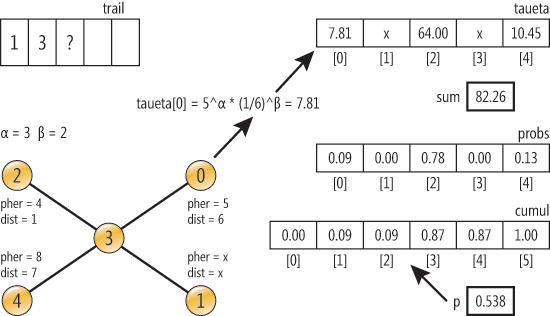
Algoritma Ant Colony Optimization (ACO) adalah teknik kecerdasan buatan didasarkan pada perilaku feromon- peletakan semut(pheromone-laying behavior of ants); dapat digunakan untuk mencari solusi untuk masalah yang sangat rumit yang mencari jalur optimal melalui graph.

**3.2 Penerapan ACO**

Program memecahkan sebuah instance dari TSP (Traveling Salesman Program) dengan tujuan untuk menemukan jalur terpendek yang dilihat setiap 60 kota tepat satu kali. Deemo menggunakan empat semut; setiap semut merupakan solusi potensial. ACO membutuhkan spesifikasi beberapa parameter seperti faktor pengaruh feromon (alpha) dan koefisien feromon penguapan (rho), yang saya akan menjelaskannya nanti. Keempat semut telah siap untuk melakukan jalan acak melalui 60 kota; setelah inisialisasi, semut terbaik memiliki panjang jalan terpendek dari 245,0 unit. Ide kunci dari ACO adalah penggunaan feromon simulasi, yang menarik semut untuk jalan yang lebih baik melalui grafik. Utama pengolahan lingkaran bergantian antara memperbarui jalan semut berdasarkan nilai feromon saat ini dan memperbarui feromon berdasarkan jalan semut baru. Setelah jumlah kali maksimum (1.000) melalui loop pengolahan utama, program ini menampilkan jejak terbaik yang ditemukan dan panjang yang sesuai (61.0 unit).

Kunci untuk algoritma ACO adalah proses yang update setiap semut dan jejak dengan membangun trail berdasarkan feromon dan jarak informasi. Lihatlah Gambar di bawah Misalkan kita memiliki grafik kecil dengan hanya lima kota. Dalam Gambar di bawah jejak baru seekor semut sedang dibangun. Jejak dimulai pada kota 1, kemudian pergi ke kota 3, dan algoritma update menentukan kota berikutnya.

Sekarang anggaplah feromon dan jarak informasi seperti yang ditunjukkan pada gambar. Langkah pertama dalam menentukan kota berikutnya adalah membangun sebuah array bernama "taueta" (karena makalah penelitian asli yang digunakan huruf Yunani tau dan eta). Nilai taueta adalah nilai feromon di tepi dipangkat alpha, kali salah satu dari nilai jarak pangkat beta. Ingat bahwa alpha dan beta adalah konstanta global yang harus ditentukan. Di sini dianggap alpha yaitu 3 dan beta 2 Nilai taueta untuk kota 1 dan kota 3 tidak dihitung karena mereka sudah lalui saat ini. Perhatikan nilai-nilai yang lebih besar dari feromon meningkatkan taueta, tetapi jarak yang lebih besar menurunkan taueta.



**Gambar 3.1 Update informasi Ant**

Setelah semua nilai taueta telah dihitung, langkah berikutnya adalah untuk mengkonversi nilai-nilai untuk probabilitas dan menempatkan mereka dalam sebuah array berlabel probs. Algoritma merangkum nilai-nilai taueta, mendapatkan 82,26 dalam contoh ini, dan kemudian membagi setiap nilai taueta dengan jumlah. Pada titik ini, kota 0 memiliki probabilitas 0.09 untuk dipilih dan sebagainya. Selanjutnya, algoritma harus memilih kota berikutnya berdasarkan probabilitas yang telah dihitung. Seperti yang saya sebutkan sebelumnya, algoritma ACO menggunakan teknik yang disebut seleksi roulette wheel. Sebuah array augmented disebut cumul, yang meyimpan probabilitas kumulatif. Ukuran array ditambah satu lebih besar dari array probs, dan sel [0] diunggulkan dengan 0.0. Setiap sel di cumul adalah jumlah kumulatif probabilitas. Setelah array cumul telah dibangun, nomor acak p antara 0,0 dan 1,0 dihasilkan. Misalkan p = 0,538 seperti yang ditunjukkan. Itu nilai p jatuh antara nilai-nilai di [2] dan [3] dalam array cumul, yang berarti bahwa [2], atau kota 2, terpilih sebagai kota berikutnya.

**AntColony.cs:**

using System;

// Demo of Ant Colony Optimization (ACO) solving a Traveling Salesman Problem (TSP).

// There are many variations of ACO; this is just one approach.

// The problem to solve has a program defined number of cities. We assume that every

// city is connected to every other city. The distance between cities is artificially

// set so that the distance between any two cities is a random value between 1 and 8

// Cities wrap, so if there are 20 cities then D(0,19) = D(19,0).

// Free parameters are alpha, beta, rho, and Q. Hard-coded constants limit min and max

// values of pheromones.

namespace AntColony {

**class AntColonyProgram {**

static Random random = new Random(0);

static int alpha = 3; // influence of pheromone on direction

static int beta = 2; // influence of adjacent node distance

static double rho = 0.01; // pheromone decrease factor

static double Q = 2.0; // pheromone increase factor

static void Main(string[] args) {

try

{

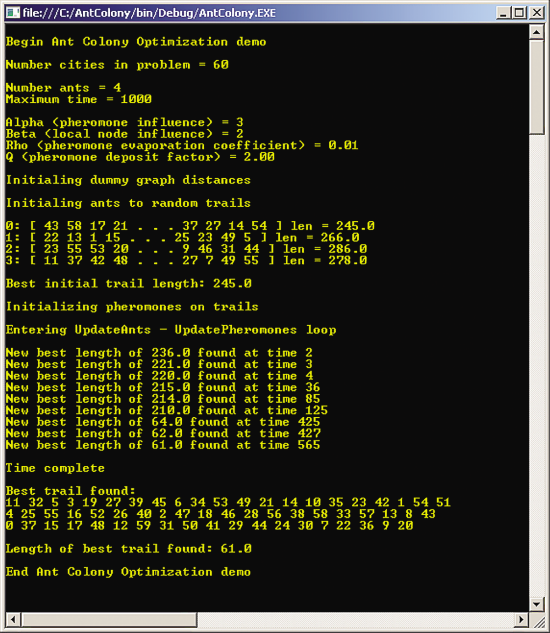
Console.WriteLine("\nBegin Ant Colony Optimization demo\n");

int numCities = 60;

int numAnts = 4;

int maxTime = 1000;

…



**Gambar 3.2 Hasil Ant Collony**

**Latihan:**

1. Buatlah program Ant Collony Optimization untuk Pencarian jarak Terpendek pada traveling salesman problem (TSP)

**SIMPULAN**

*Algoritma Ant Colony Optmization (ACO) adalah teknik kecerdasan buatan didasarkan pada perilaku feromon- peletakan semut(pheromone-laying behavior of ants); dapat digunakan untuk mencari solusi untuk masalah yang sangat rumit yang mencari jalur optimal melalui graph.*

**DAFTAR PUSTAKA**

* Adries P. Engelbrect. (2007), ***Computational Intelligence An Introduction***. 2nd ed. John Wiley & Sons. USA.
* **http://msdn.microsoft.com/en-us/magazine/hh781027.aspx**