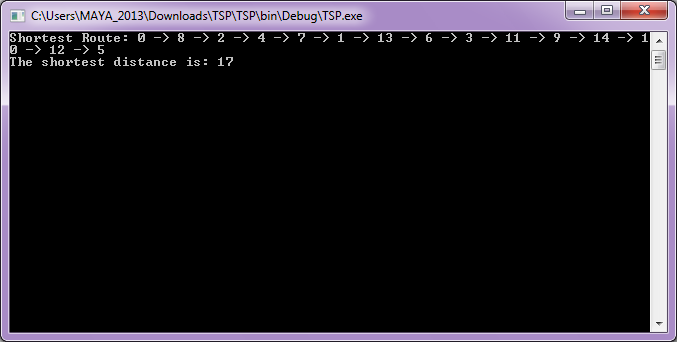
**ANALISIS TRAVELLING SALESMAN PROBLEM (TSP)**

**MENGGUNAKAN ALGORITMA SIMULATED ANNEALING**

Travelling Salesman Problem (TSP) adalah problem untuk mengoptimasi dan menemukan perjalanan (tour) yang paling terpendek. TSP adalah problem untuk menentukan urutan dari sejumlah kota yang harus dilalui oleh salesman, setiap kota hanya boleh dilalui satu kali dalam perjalanannya, dan perjalanan tersebut harus berakhir pada kota keberangkatannya dimana salesman tersebut memulai perjalananya, dengan jarak antara setiap kota satu dengan kota lainnya sudah diketahui.

Misalnya seorang salesman ingin melakukan perjalanan ke N buah kota (ia harus mellewati masing-masing kota). Bagaimana caranya mengurutkan kota sehingga perjalanan salesman tersebut merupakan yang terpendek? Fungsi tujuan untuk meminimalkan di sini adalah panjang perjalanan (jumlah dari jarak semua kota dalam urutan tertentu). Permasalah tersebut dapat dipecahkan dengan cara:

1. Pengaturan konfigurasi : adalah permutasi dari kota ke-1 sampai kota ke-N, diberikan untuk semua urutan. Pilih yang paling optimal diantara permutasi tersebut.
2. Strategi penataan : yaitu mengganti bagian jalan (path) dan menggantinya dengan salah satu yang lain secara acak untuk menguji apakah yang dimodifikasi tersebut optimal atau tidak.
3. Fungsi tujuan (yan merupakan tujuan minimalisasi): adalah jumlah jarak antara semua kota untuk urutan tertentu.



Pada travelling salesman problem algoritma simulated annealing digunakan untuk mencari fungsi tujuan secara acak (yang terutama mencirikan masalah optimasi kombinatorial).

Keuntungan simulated annealing dibandingkan dengan metode lain adalah kemampuannya untuk menghindarkan terjebak dalam minimum lokal. Di sini, berarti algoritma tidak selalu menolak perubahan yang menurunkan fungsi tujuan tetapi juga perubahan yang meningkatkan fungsi tujuan sesuai dengan fungsi probabilitas:

P = exp (-∆f/T)

Dimana T adalah parameter control (analogi suhu) dan ∆f adalah variasi dalam fungsi tujuan. Fungsi probabilitas merupakan turunan dari fungsi distribusi probabilitasBoltzmann.

Dibawah ini merupakan algoritma simulated annealing untuk aplikasi di atas:

/// <span class="code-SummaryComment"><summary></span>

/// Annealing Process

/// <span class="code-SummaryComment"></summary></span>

public void Anneal()

{

int iteration = -1;

double temperature = 10000.0;

double deltaDistance = 0;

double coolingRate = 0.9999;

double absoluteTemperature = 0.00001;

LoadCities();

double distance = GetTotalDistance(currentOrder);

while (temperature > absoluteTemperature)

{

nextOrder = GetNextArrangement(currentOrder);

deltaDistance = GetTotalDistance(nextOrder) - distance;

//if the new order has a smaller distance

//or if the new order has a larger distance but

//satisfies Boltzman condition then accept the arrangement

if ((deltaDistance < 0) || (distance > 0 &&

Math.Exp(-deltaDistance / temperature) > random.NextDouble()))

{

for (int i = 0; i < nextOrder.Count; i++)

currentOrder[i] = nextOrder[i];

distance = deltaDistance + distance;

}

//cool down the temperature

temperature \*= coolingRate;

iteration++;

}

shortestDistance = distance;

}