Travelling salesman problem solution Mark McIlroy, 24 April 2010 V. 1.4

Given a set of cities, determine the shortest path to travel to visit all cities, without visiting any city twice.

For each point, determine the distance to each other point

Select a starting city

Take the path that’s shortest to an other city

Remove the starting city from the list

Repeat this process until all cities have been visited

This generates one solution

Repeat the process, selecting each possible city as the starting point

Select the shortest path from the solutions generated

Proof

Start with the set of all line segments connecting individual cities

Select a starting city

Select the shortest segment

This will have the effect of removing all other line segments from this city from the solution set

Continue to the next point

Repeat this process

The final solution set will contain the set of the shortest line segments connecting each of the cities

This path should then be the shortest path to travel to visit each of the cities

Simulation Results

Shortest Path 16956.8788943284

Longest Path 58396.6781378445

Longest Path Starting City 5

Shortest Path Starting City 5

TS solution Path 16956.8788943284

TS solution Path Starting City 5

TS solution Path Cities 7, 3, 5, 2, 1, 6, 4, 8

Total Paths 322560

Location x Location y

City 1 1955 3262

City 2 4132 1527

City 3 6198 994

City 4 2050 6925

City 5 9914 4714

City 6 5045 1831

City 7 40 4258

City 8 2887 7525

Applications

For the shortest path through a network, select the next city as the next possible one that is the closest to the destination point.

The distance between two points may be geographical, or a cost function such as energy lost.Appendix 1

Simulation Source Code

Const num\_cities As Integer = 8

Dim x\_values(100) As Double

Dim y\_values(100) As Double

Dim shortest\_path As Double

Dim longest\_path As Double

Dim lp\_starting\_city As Integer

Dim sp\_starting\_city As Integer

Dim sp\_path As String

Dim ts\_dist As Double

Dim ts\_starting\_city As Integer

Dim ts\_path As String

Dim total\_paths As Double

Function ts()

Dim i As Integer

shortest\_path = 10000000000#

ts\_dist = 10000000000#

' example set

If 1 = 1 Then

x\_values(0) = 1955

x\_values(1) = 4132

x\_values(2) = 6198

x\_values(3) = 2050

x\_values(4) = 5045

x\_values(5) = 40

x\_values(6) = 9914

x\_values(7) = 2887

y\_values(0) = 3262

y\_values(1) = 1527

y\_values(2) = 994

y\_values(3) = 6925

y\_values(4) = 1831

y\_values(5) = 4258

y\_values(6) = 4714

y\_values(7) = 7525

Else

For i = 0 To 99

x\_values(i) = Round(Rnd \* 10000, 0)

y\_values(i) = Round(Rnd \* 10000, 0)

Next

End If

For i = 0 To num\_cities - 1

proc\_city i, i, CLng(0), CDbl(0), False, ""

Next

For i = 0 To num\_cities - 1

proc\_city i, i, CLng(0), CDbl(0), True, ""

Next

Open "c:\solutions.txt" For Output As #1

Print #1, shortest\_path

Print #1, longest\_path

Print #1, lp\_starting\_city

Print #1, sp\_starting\_city

Print #1, ts\_dist

Print #1, ts\_starting\_city

Print #1, ts\_path

Print #1, total\_paths

Print #1, ""

For i = 0 To num\_cities

Print #1, x\_values(i), y\_values(i)

Next

Close #1

MsgBox "x"

End Function

Function proc\_city(start\_city As Integer, city As Integer, cities As Long, total\_dist As Double, ts As Boolean, ByVal path As String)

Dim i As Integer

Dim unv As Integer

Dim new\_val As Long

Dim min\_city As Integer

Dim min\_dist As Double

unv = False

If ts Then

min\_dist = 100000000

For i = 0 To num\_cities - 1

If ((Not cities) And 2 ^ i) <> 0 Then

unv = True

dist = Sqr((x\_values(city) - x\_values(i)) ^ 2 + (y\_values(city) - y\_values(i)) ^ 2)

If dist < min\_dist Then

min\_dist = dist

min\_city = i

End If

End If

Next

If unv Then

new\_val = cities + 2 ^ min\_city

dist = Sqr((x\_values(city) - x\_values(min\_city)) ^ 2 + (y\_values(city) - y\_values(min\_city)) ^ 2)

proc\_city start\_city, min\_city, new\_val, (total\_dist + dist), True, path & ", " & min\_city + 1

End If

Else

For i = 0 To num\_cities - 1

If ((Not cities) And 2 ^ i) <> 0 Then

unv = True

new\_val = cities + 2 ^ i

dist = Sqr((x\_values(city) - x\_values(i)) ^ 2 + (y\_values(city) - y\_values(i)) ^ 2)

proc\_city start\_city, i, new\_val, total\_dist + dist, False, path & ", " & i + 1

End If

Next

End If

If Not unv Then

If ts Then

If total\_dist < ts\_dist Then

ts\_dist = total\_dist

ts\_path = path

ts\_starting\_city = start\_city

End If

Else

total\_paths = total\_paths + 1

If total\_dist > longest\_path Then

longest\_path = total\_dist

lp\_starting\_city = start\_city

End If

If total\_dist < shortest\_path Then

shortest\_path = total\_dist

ts\_path = path

sp\_starting\_city = start\_city

End If

End If

End If

End Function