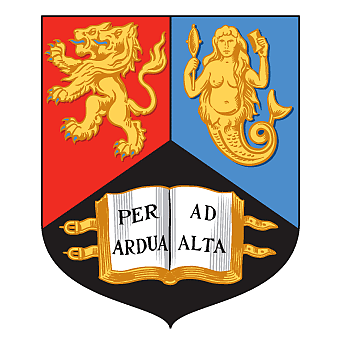
ELECTRONIC, ELECTRICAL AND COMPUTER ENGINEERING



University-of-Birmingham-logo

# *Object Oriented Programming Using C#*

**Assignment 2014-15**

**Name: Jing Wang**

**Student ID: 1483063**

Using Simple Greedy Solver and Basic ACO (Ant Colony Optimization) algorithms to solve Traveling Salesman Problem

1. Background

1.1 Basic conception

1.1.1 TSP

TSP (Traveling salesman problem), is given N cities and there have roads to connect any two cities. It is also known that the distance of two cities. The salesman should select one city to begin travel from N cities and visit every city once and only once. The TSP is the problem of finding the minimal length after traveling.

Traveling salesman problem is a typical combination optimization problem of operational research and is also typical NP problem. Although statement is simple, the solution of solving this problem is difficult. For TSP problem which has N cities, there are has (N-1)!/2 possible path. In theory, enumeration method can solve this problem, but when the cities N is large, the long time of solving this problem will make enumeration method lose its actual value. It is important that find a solver which meets the accuracy requirement and spend short time for real problem.

1.1.2 Basic ACO Algorithm

ACO (Ant Colony Optimization Algorithm, known as ACO), is came up by Dorigo·m, which is a new evolutionary algorithm. Its main feature: using feedback and distributed collaboration to looking for optimal path. This is a heuristic search algorithm based on swarm optimizing. Ant group can search the minimal path from nest to food by using pheromone between different individual and this process is similar to traveling salesman problem, there is a thought that using the ACO algorithm to solve the TSP problem and get the optimal solution which has NP-hard.

1.1.3 Greedy Algorithm

The greedy algorithm is a simpler and more rapid design technology for some problems which need find the best solutions. The feature of greedy algorithm is step by step, it is based on current situation and an optimal measure to make a optimal select, and it is not consider overall situation, it also saves time for find optimal solutions. Greedy algorithm uses top-down and iteration method to make a choice, making a choice to simplify the scale of the problem. An optimal solution can be got by making the choice each part. Although every step has to be guaranteed access to part optimization, the global solution sometimes is not the best result.

For TSP problem, one of the most straightforward algorithms is a brute-force method, considering all possible travel routes and choosing the best one. But the complexity of brute-force method to solve TSP problem is N!, and it is insoluble when N is extremely large. So we select the greedy algorithm, but we must clearly recognize that the greedy algorithm that still has its shortcomings (that is, when you are solving a problem, the best solution is based on current situation, not global situation. In other words, this solution is best in part). A greedy algorithm does not get the global optimal solution for all problems, but it can create a global optimal solution or approximation of global optimal solution for some problems.

1.2 Basic principle

1.2.1 ACO Algorithm

Use Ant Colony Optimal algorithm for N cities TSP problem. Assume  is the geometric distance between urban i and j, ＝. The number of Ant is m,  represents the amount of pheromone currently on the line between i and j. The initial amount of pheromone on each path ＝Ｃ(c is set to some small positive constant).  is the pheromone evaporation constant. Then after n moments later, the pheromone deposit and evaporation processes can be expressed by the following equation:

 (1)

 (2)

 is the pheromone of the path between city (i, j) by ant k left on this loop.

Q is a constant. is the tour (list of cities visited in sequence) by ant *k* at iteration *t* and is its length.

 is the pheromone of all ants visited the i and j in this loop .

We define ＝１／. Ant k (k=1,2,...,m) in the movement,  is ant k chooses

to go from city i to city :



 (3)

We use  to record the cities that ant k has visited, and remove these cities from all cities, and then ant k selects next city to visit. It is obviously that if P=1, the algorithm is then the deterministic. Typically is set to some small positive constant, , ,and are reported to give good results.

Using Ant Colony Optimal algorithm to solve TSP problem is a recursive process, when t=0, set ants in the cities and set the initial value of pheromone on each path =C. According to the probability of formula 3, every ant determines the next city.  means more closer to the city, there are more likely to be selected. β is used to control the influence on ants choice. After a loop, we need to update the pheromone on each path by using formula (1) (2). Finally we need to calculate the shortest path of this cycle. And this process will repeat until all the ants chose same path or reached  which is defaulted on the beginning of program.

1.2.2 Greedy Algorithm

Providing a total of N cities, the cities’ number are 1, 2, 3…n. Greedy algorithm start from city 1, calculating the distance between 1 and n (n=2,3,4,5…N), comparing these distances and go to the nearest city j (Assume j is 3). And then calculate the distance between the 3 and n (n=2,4,5…N). City 1 has been visited, so we use  to record the cities visited. This process will repeat until calculated the last city. And calculate the length of this tour.

2. Introduction

TSP Specification

The system use ACO algorithm or Greedy algorithm to solve the traveling salesman problem and show the shortest tours on the windows form. The user can see four options on the menu of form, the user clicks on the file and opens the location of TSP file, choosing one. Then the points of cities, the file name and the existed optimal length of TSP file will be shown on the right hand of form. And then there are three options on the menu: ACO algorithm, Greedy algorithm, Random algorithm, selecting one of them, and then the system will calculate the route, drawing the route of this algorithm and showing the length on the right hand of form. The Random algorithm use to highlight the significant of algorithm for solving a complex problem.

3. Use -Case Model

3.1 Use –Case Diagram

Figure 3.1 Use-case View Diagram

3.2 Scenario Description

Tsp: The user open the executable file, there are four options on the menu of form. The user opens a tsp file by clicking on the ‘File’ menu to reach the location of Tsp data file. One tsp file is selected and the form to show its cities’ points, name and the existed optimal length.

After user chooses the TSP file, then the user will choose a algorithm of the three algorithm (ACO, Greedy, Random), the current length and the current route will be shown on the form.

Note: If the user did not choose file and open it, the ACO, Greedy and Random can not be clicked.

Program：On the one hand, when the user clicks on ‘Greedy’, the associated event will be triggered. And this will link to greedy class to calculate the current length by using greedy algorithm. On the other hand: when the user click on ’ACO’, the associated event will be triggered. And this will link to ACO class to calculate the current length by using ACO algorithm. Obviously, the user can chooses ‘Random’ to compare the influence of using a algorithm. After the algorithm calculated the current length and store the current route, the result will be transfer to the tsp form to show the route on the picture box and show the current length on the right hand of form.

Operator Override: After the user chosen a tsp file and one algorithm, the calculation is in progress. At this moment, if the user chooses another file, the program will end. And if the calculation is in progress, the user can not choose another algorithm until this algorithm has completed.

3.3 Stereotypical classes

《Control》: Program

《Interface》: Tsp form

Noun identification

Tsp form: Tsp file, ACO algorithm, User, Greedy algorithm, Random algorithm, name, existed optimal length, city number

Program: current length, current route

Delete nouns that lack attributes and methods appropriate to a class and, perhaps, are attributes with values only. These are: name, existed optimal length, city number.

Conclusion: Re-considering the prompts from the list of Nouns and the stereotypical classes the following are identified as a minimal set of classes: Program, Tsp

Boundary: Tsp

Entity: Tsp, Program

Control: Program

Notes: The program includes six classes: ACO, Greedy, Random, Wj, Algorithm and Program. ACO, Greedy and randomAlg are the three algorithms for calculating the best route of this Tsp problem. The Wj class has a parameter and a function for the three algorithms to invoke it to calculate the distance. The Algorithm class is the parent class of the three algorithms for extending other algorithm. The program class is the main function. For statement convenience, I use the program to describe them.

3.4 Class diagram



Figure 3.4 Class Diagram (Stereotypes)

3.5 CRC Cards

CRC cards describe the responsibilities and collaborators of each class.

Table 3.5 Class Responsibility Collaboration Cards

|  |  |
| --- | --- |
| **Class: Tsp** |  |
| **Responsibilities** | **Collaborators** |
| The Tsp form class is responsible for show the initial form to user. User can click the ‘file’ to select the TSP data file. It reads the TSP file and stores the data of the file, showing the cities point, name, optimal length of this file.  It also transfers the data of this file to the program class. After one of the three algorithms calculates the data, the current route and the current length will be transferred to the tsp form, and then the class will draw the route and show on the form. | Program |
| **Class: Program** |  |
| **Responsibilities** | **Collaborators** |
| The program class is responsible for receiving the data from Tsp form class and calculating the data by using Greedy algorithm, ACO algorithm and Random algorithm. And then the current length and current route are transferred to the tsp form to show the route and length on the form. | Tsp |

3.6 Interaction Diagram



Figure 3.6 Interaction Diagram: Sequence Diagram

3.7 Statechart Diagram

Figure 3.7 Statechart diagram

4. Analysis Model

From Tsp Class

Potential attributes: Nc, cityn, xmax, xmin, ymax, ymin, XY, datapath, x ,y, bestL, graphic, imagetemp

Potential methods: Filepath\_Click(), ReadFile(), DrawRoute(), DrawPoint(), ShowLength(), ACO\_Click(), ACOThread(), AcoCal(), Random\_Click(), RandomThread(), RanCal(), Greedy\_Click(), GreedyThread(), GreedyCal(), Tsp\_FormClosing(),

From Program Class

Potential attributes: M, N, inittao, tao, detatao, distance, x, y, NcMax, Tk, NTk, route, Lk, solution, bestSolution, beta, rou, Q, ppij,

Potential methods: ACO class: ResultRoute(), AllRoute()

Greedy class: Calculate(), ResultRoute(), AllRoute()

randomAlg class: ResultRoute(), AllRoute()

4.1 Attributes (all private)

Table 4.1 Attributes

|  |  |  |
| --- | --- | --- |
| Class | Attributes | Comment |
| Tsp | Nc | Current iteration |
| cityn | City number |
| xmax | The max value of x of city |
| xmin | The min value of x of city |
| ymin | The min value of y of city |
| ymax | The max value of y of city |
| dataPath | The file path |
| graphic | For draw picture |
| imagetemp | For draw picture |
| XY | The parameter for draw |
| x | Save x point of city |
| y | Save y point of city |
| bestL | Store the best route |
| Program | M | The ant number |
| N | The city size |
| inittao | The initial pheromone |
| tao | Pheromone matrix |
| detatao | Pheromone increase matrix |
| x | X coordinate of the city |
| y | Y coordinate of the city |
| NcMax | Maximum of iteration |
| Tk | Store the cities visited |
| NTk | Choose cities that not visited |
| route | Store all cities visited |
| Lk | Store the current route |
| solution | Store the shortest route |
| bestsolution | Set a very large number |
| Beta,rou,Q | The parameter of algorithm |
| ppij | The probability of the ant moving |
| Nc | Current iteration |

4.2 Method

Table 4.2 Methods

|  |  |  |
| --- | --- | --- |
| Class | Method | Comment |
| Tsp | FilePath\_Click() | Open the location of file |
| ReadFile() | Read file |
| DrawRoute() | Draw the current route |
| DrawPoint() | Draw the point of city |
| ShowLength(), | Display the current length |
| ACO\_Click(), | Invoke the ACO thread |
| ACOThread(), | Invoke the AcoCal |
| AcoCal(), | Invoke ACO class |
| Random\_Click(), | Invoke the Random thread |
| RandomThread(), | Invoke the RanCal |
| RanCal(), | Invoke randomAlg class |
| Greedy\_Click(), | Invoke Greedy thread |
| GreedyThread(), | Invoke GreedyCal |
| GreedyCal(), | Invoke Greedy class |
| Tsp\_FormClosing(), | Exit to the system |
| Program | ResultRoute() | Start move by Greedy |
| AllRoute() | Calculate the current length |
| ResultRoute() | Start move by ACO |
| AllRoute() | Calculate the current length |
| ResultRoute() | Start move by randomAlg |
| AllRoute() | Calculate the current length |

4.3 International Diagram



Figure 4.3 Analysis Interaction Diagram

4.4 Class Diagram



Figure 4.4 Analysis Class Diagram

4.5 Statechart Diagram



Figure 4.5 Analysis Statechart Diagram

5. Design Modeling

5.1 Revisit Use-Case Model

The design meets the requirements.

5.2 Sequence Diagram

The principal qualifiers have already been added.

5.3 Textual Description of Object to Object Interaction

This is a simple system and the various diagrams give a clear description. The following description is intended for illustration.

The Tsp class handles the interaction between the user and the program, the user has some actor like as click the menu to select tsp file and select the algorithm. The Tsp class is responsibility for reading file and show the point of city and show the route of algorithm.

The program class responds to calculate the problem by using algorithm and return the best route and the current shortest length to the tsp class to show.

5.4 Subsystem

No subsystem is required as this is a simple design.

5.5 Implementation of Non-functional Requirements

No Non-functional Requirements.

5.6 Deployment Model

The program is implemented on a single processor.

5.7 Legacy Issues

No legacy issues.

5.8 Reconsider the Attributes

Incorporated in the Class Diagram

5.9 Reconsider the Associations

The use of Aggregation remains appropriate

5.10 StateChart

No further revisions required

5.11 Class Diagram Showing Visibility



Figure 5.11 Design Model Class Diagram showing visibility

6. Implementation and Testing

6.1 The form of this program

6.1.1 Four menus on the menustrip: File, ACO, Greedy, Random

6.1.2 Picturebox for show the point of cities and the route.

6.1.3 Using five labels to show the tsp file information and the current length

6.1.4 I set the beta=7.0 (), rou=0.5 (), Nc=400 (iteration), Q=100, inittao=1()

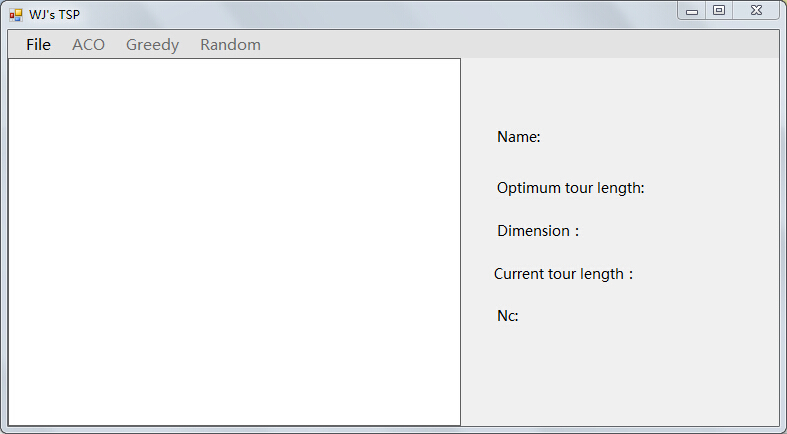


Figure 6.1 Tsp Interface Diagram

6.2 Click ‘File’ to choose a tsp file

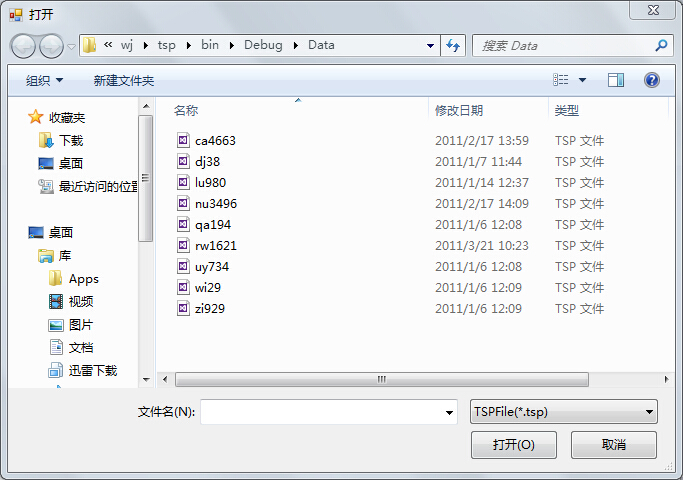


Figure 6.2 Choose tsp file

6.3 Display the point of cities and the information of this tsp file (I choose dj38)

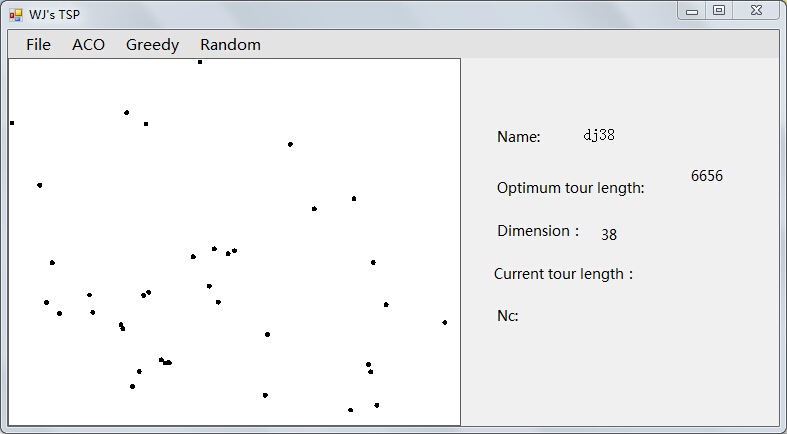


Figure 6.3 Show point of cities

6.4 Random algorithm

6.4.1 Click ’Random’ to calculate

6.4.2 Display the route and the current tour length and the iteration

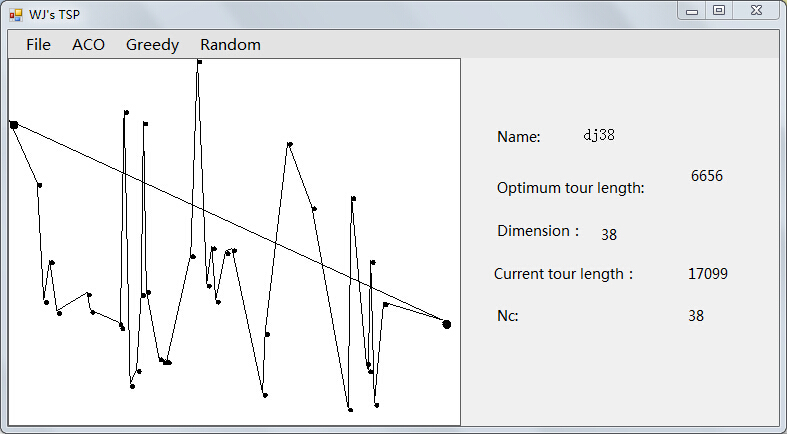


Figure 6.4 Result of Random

6.5 Greedy algorithm

6.5.1 Click ’Greedy’ to calculate

6.5.2 Display the route and the current tour length and the iteration

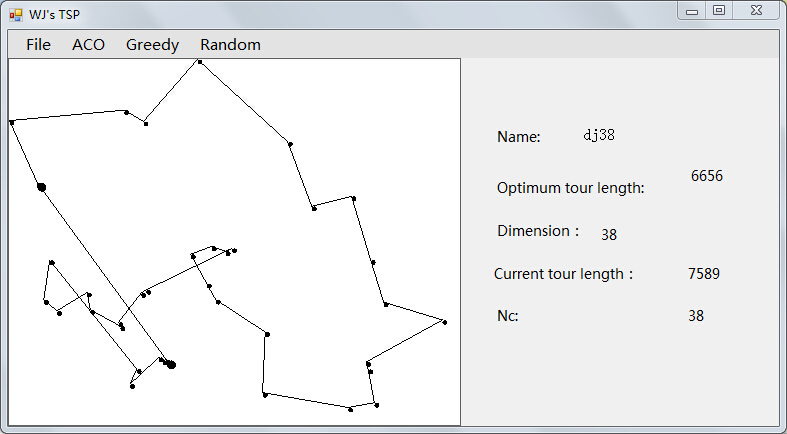


Figure 6.5 Result of Greedy

6.6 ACO algorithm

6.6.1 Click ’ACO’ to calculate

6.6.2 Display the route and the current tour length and the iteration

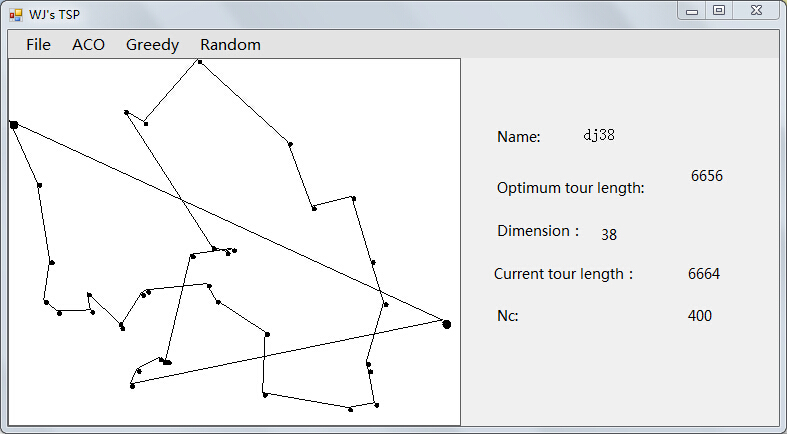


Figure 6.6 Result of ACO

Note: If the program is in progress, the user chooses another file, the program will stop and have some problem. If the iteration and city number are large, the rate of running is very slow.

6.8 Compare the three algorithms

6.8.1 Random algorithm

From the result: We can see the current length is 17099 and the optimal length is 6656. There has a very large gap. Actually, the random algorithm is the salesman travel randomly. For emphasizing the significant of algorithm, I use the random method to compare.

6.8.2 Greedy algorithm

From the result: We can see the current length is 7589. Compared whit the random method, there are a great progress. But there also has a gap between the optimal length and current length.

6.8.3 ACO algorithm

From the result: We can see the current length is 6664. Compared with the Greedy algorithm, the ACO algorithm is better, and the gap is small between optimal length and the current length.

7. Conclusion

This program is using algorithm to solve the Travelling Salesman Problem. I choose add a random method to emphasize the important of using algorithm to solve the problem.

From the result of this program, it is obviously that the ACO algorithm is better than Greedy algorithm. Compared to the ACO algorithm, the Greedy algorithm just use the distance of cities as a judge and to choose the next destination. It is easier than ACO and the result is worse than ACO.

8. References

1. Craig Larman, *Applying UML and Patterns An Introduction to Object-Oriented Analysis and Design and the Unified Process,* China Machine Press, 2005

2. Grady Booch, Ivar Jacobson and James Rumbaugh, *UML distilled: a brief guide to the standard object modeling language/Martin Fowler with Kendall Scott-2nd el.* ISBN 0-201-65783-X

3. D.Pycock, *Lift Control System,*2014-12

4.D.Pycock, *UML: Unified Modelling Language Issue 2.2,*2014-15(2007-8)

5. Haibin Duan, *Ant Colony Algorithms: Theory and Applications，*Science Press, 2005.12 ISBN 7-03-016204-8

5.段海滨，蚁群算法原理及其应用，科学出版社，2005.12 ISBN 7-03-016204-8