



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Lecture with Computer Exercises:
Modelling and Simulating Social Systems with MATLAB

Project Report

**Solving the Travelling Salesman Problem
by Using an Artificial Ant Colony**

Raphaela Wagner & Giandrin Barandun

Zurich
May 2014

Agreement for free-download

We hereby agree to make our source code for this project freely available for download from the web pages of the SOMS chair. Furthermore, we assure that all source code is written by ourselves and is not violating any copyright restrictions.

Raphaela Wagner

Giandrin Barandun

Contents

1	Abstract	4
2	Individual contributions	4
2.1	Raphaela Wagner	4
2.2	Giandrin Barandun	4
3	Introduction and Motivations	5
4	Description of the Model	5
5	Implementation	5
5.1	Main program	5
5.2	Choose the Next City	5
6	Simulation Results and Discussion	6
7	Summary and Outlook	6
8	References	6

1 Abstract

2 Individual contributions

2.1 Raphaela Wagner

With the aim of achieving a good model for solving the travelling salesman problem by the use of artificial ants Raphaela helped the group understanding the underlying paper and the included model. She contributed a great amount of explanations and ideas how to approach the whole project.

In a second step she took care of how to implement the raw data from the TSP-library into MATLAB and transform it to a usable form. Further more she coded the functions "eta.m", "global_pheromene_update.m", "test_funktionen.m" and "update.m" and helped improving and correcting the main program.

After the code was written she did a lot of testing with different problem sets and compared the solution of the program to known solutions.

When the group got stuck and did not see a way out of a specific problem she was the one to bring along a hot chocolate and cheer the group up again.

2.2 Giandrin Barandun

The paper which is thought to be reconstructed on the following pages was selected and suggested to the group by Giandrin and during the whole process he tried to have some influence on the project with his wide technical understanding of the problem.

The codes for the functions "prob_dist.m", "calc_Lnn.m", "choose_city.m", "main_initialize_system.m", "coordinates.m" and "calc_dist.m" are his contributions as well as the collaboration on the main program. He searched the internet for known TS-problems and their solutions and put all data in a readable form. A lion's share for the program working at the end was his bug fixing in all the functions and programs and combining them to the running model.

3 Introduction and Motivations

4 Description of the Model

5 Implementation

5.1 Main program

In the previous section the theoretical understanding of the model was tried to be imparted to the reader. Following in this section is an overview of how the model was implemented in MATLAB.

To start with a main program (*main_initialize_system.m*) was written which reads the data of a specific TSP and puts it into an upper triangle matrix form which contains the distances between the cities as elements. Further more all parameters like number of rounds and number of contributing ants (agents) can be adjusted in this main program. At the end it activates the function *main_main.m* with the purpose to find the shortest route.

In the main function there are two for-loops, one for the number of rounds and one for the number of ants (agents). For every agent we have a memory of the visited cities (M_k) and a trajectory vector which saves the sequence of how the cities were visited. In the first step every ant chooses the next city with help of the function *choose_city.m* which is described below. The city is added to the memory of the ant and the trajectory vector and the pheromone on the edge is reduced according to the formula on page 75 in the paper. As long as there are unvisited cities this procedure is repeated for the agents and at the end the way to the start city is updated (pheromone, trajectory) and the memory reset. Then the second agent starts his tour.

After every agent has finished his route the shortest path is detected and the edges along this route are rewarded with pheromone with help of the trajectory vector and the function *global_pheromone_update.m*. At the end of this step the trajectory vector is reset for all ants and the second round can be initialized. After every round the shortest route of the current round is compared to the overall shortest route found until now and then kept or rejected depending on the result of the comparison.

When all rounds have been calculated the function gives the shortest path found in any of the rounds.

5.2 Choose the Next City

To choose which city the ant will go next two different methods are implemented according to the paper and one of them is selected based on a certain probability.

The first way to decide which city to go next is to optimize a function which depends on the amount of pheromone on the edge between the current city and the chosen one and its length. This method chooses the edge with the highest amount of pheromone and the shortest length whereas there are parameters to weight these two variables relatively to each other.

On the other hand a probability was assigned to every unvisited city again depending on the amount of pheromone and the length of the edge between the cities. Then a city was randomly chosen and accepted or rejected with its assigned probability.

At the end the function *choose_city.m* gives the number of the next city to visit back to the main function.

6 Simulation Results and Discussion

7 Summary and Outlook

8 References