



Assignment 5  
Computational Intelligence  
Dr. S. Hajipour

Mohammad Hossein Shafizadegan  
99104781

January 6, 2024

## Contents

1	Question 1	2
2	Question 2	2
3	Question 3	3
4	Question 4	3
5	Question 5	4
6	Question 6	5
7	Question 7	5

## 1 Question 1

we need to define the following components:

- **Particle Representation**

Each particle in the swarm represents a potential solution and consists of the locations of all sprinklers and and their irrigation time. A schematic of a particle can be seen below

x1	y1	T1	x2	y2	T2	x3	y3	T3
----	----	----	----	----	----	----	----	----

Where  $(x_i, y_i)$  are real numbers representing the location of the sprinkle and  $T_i$  represents the irrigation time in hours and is a real number .

$$0 \leq x_i \leq 100 \quad , \quad 0 \leq y_i \leq 100 \quad , \quad 0 \leq T_i \leq 24 \text{ hour}$$

- **Fitness function**

We aim to maximize the following fitness function:

$$f = -\frac{1}{N} \sum_{i=1}^N |H_i - \hat{H}_i|$$

Where  $H_I$  denotes the desired measured humidity of the  $i$ th sensor and  $\hat{H}_i$  is value of humidity of the sensor measured at each (current) iteration.

- Each particle keeps track of its best solution (position) and the global best solution among the entire swarm. The particles move based on their current position, velocity, and the influence of their best position and the global best position.

## 2 Question 2

**a**

In this problem, each particle is an array containing 9 elements, each representing the length of the wood stick. The first element corresponds to the bottom position and the last element indicates the wood placed at the top (part 9). This defines the position vector. The formula for updating the position and velocity vector can be seen below:

$$v_i(t+1) = v_i(t) + \beta_1 \left( x_i^{(local)}(t) - x_i(t) \right) + \beta_2 \left( x_i^{(global)}(t) - x_i(t) \right) \quad , \quad x_i(t+1) = x_i(t) + v_i(t)$$

Value of each element in the position vector, ranges from 0 to  $L$ , considering the fact that summation of all the elements must be equal to  $L$ .

$$\forall i : 0 \leq L_i \leq L \quad , \quad \sum_{i=1}^9 L_i = L$$

**b**

We will consider a penalty term for the case that summation of all pieces exceed  $L$ . The fitness function can be defined as follows:

$$f = -c \times \text{sign} \left( \sum_{i=1}^9 L_i > L \right) - \left| S - \delta \sum_{i=1}^9 L_i \right|$$

Where  $S$  is the area of the desired shape.

### 3 Question 3

**a**

The position and velocity vectors are arrays with 5 elements. Each element is a real number and can vary in the range of 0 to 10. We can update the velocity and position using the following formulas:

$$v_i(t+1) = v_i(t) + \beta_1 \left( x_i^{(local)}(t) - x_i(t) \right) + \beta_2 \left( x_i^{(global)}(t) - x_i(t) \right)$$

$$x_i(t+1) = \max \left( \min \left( x_i(t) + v_i(t), \begin{bmatrix} 10 \\ \vdots \\ 10 \end{bmatrix} \right), \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \right)$$

We don't have any constraints for the velocity vector but we should be careful about the range of the position vector.

**b**

The procedure is the same except that only the position values (not velocity) are discrete. Therefore, we have to change the formula for position correspondingly.

$$v_i(t+1) = v_i(t) + \beta_1 \left( x_i^{(local)}(t) - x_i(t) \right) + \beta_2 \left( x_i^{(global)}(t) - x_i(t) \right)$$

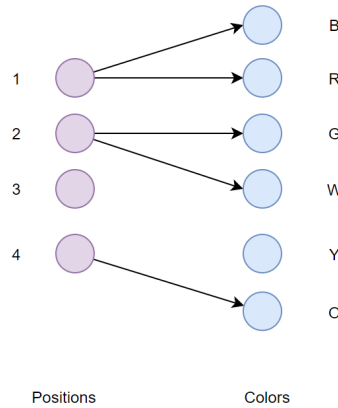
$$x_i(t+1) = \text{Round} \left( \max \left( \min \left( x_i(t) + v_i(t), \begin{bmatrix} 10 \\ \vdots \\ 10 \end{bmatrix} \right), \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \right) \right)$$

### 4 Question 4

In the ACO algorithm we have to define and clarify the following concepts:

- **Graph nodes and edges**

We consider a fully connected graph consisting of two sets of nodes. One including 4 nodes for each position. The other set consists of 6 nodes denoting the colors. The edges of the graph are places between these two sets of nodes. A schematic of this graph can be seen below:



- **A candidate for the solution**

We aim to choose 4 edge from all of the 24 possible edges. We note that these 4 edges, must connect each node of the left side to one of the nodes on the right side. The optimum and best 4 edges will be chosen based on the pheromone of the edges. In the process, whenever an edge is selected, all the edges connected to that starting node (on the left side) will be removed.

- **Constrains**

As stated before, each starting node on the left side must have one and only one edge. This must be considered in the process of selecting the edges.

- **The pheromone**

By choosing 4 edges, a candidate will be chosen and the computer can calculate the score based on the described method for similarity of the candidate and the final answer. This score can be used as the pheromone for the selecting process.

- **Useful info.**

It is stated that the probability of selecting two similar colors in a row (two adjacent position) is half of the probability of choosing different colors. This can be useful when choosing the edges in a way that we can consider the edge of the adjacent position.

## 5 Question 5

### Graph nodes and edges

We consider a fully connected graph consisting of two sets of nodes. One including N nodes for each task. The other set consists of M nodes indicating the workers. The edges of the graph are places between these two sets of nodes and each one assign a work a task.

### A candidate for the solution

First we consider a node in the tasks set. Based on the probability of choosing an edge which depends on the value of pheromone, task time (T) and cost (V), we choose one edge. This will indicate the worker assigned to this task. Then with a probability we decide whether we assign this task to another worker or not. If so, we probabilistically choose another edge after removing the previous one which is already assigned to a worker. We continue this process once again as it is stated that a task can be assigned to three workers. This procedure we be done again for other tasks.

### Constrains

We note that each task can not be assigned to more than 3 workers. This issue is addressed in the method we construct a candidate.

Also the total cost in this problem must not exceed  $V_{max}$ . This is addressed by utilizing a penalty term.

### The pheromone

By selecting and forming a candidate solution, we can calculate the total required time and cost considering the fact that some tasks are executed in parallel and some in series. Indeed there must be a simple mechanism that define this phenomena and calculate the total time. For updating the pheromones, we will add a proportion of the inverse of this total time to the pheromones of the edges selected in this solution. As of the penalty term, we decrease the pheromones if the total cost exceeds  $V_{max}$ .

$$pheromone(t+1) = pheromone(t) + \frac{1}{T_{total}} - c \times sign(V_{total} > V_{max})$$

**Useful info.**

We tend to minimize the required time to finish all the task and we want the cost to be as low as possible. Therefore, the inverse of time ( $\frac{1}{T_{nm}}$ ) and inverse of cost ( $\frac{1}{V_{nm}}$ ) can be also useful in the process of selecting the edges alongside considering the pheromones.

## 6 Question 6

**Graph nodes and edges**

Consider a graph including  $N$  sub graphs representing the provinces. Each sub graph consists of  $I_n$  nodes, indicating the cities of the province. The edges are only connected between sub graphs and the nodes of a sub graph are not connected to each other as it is stated that we want to travel into only one city of each province. The value of the edges represent the distance between the two corresponding cities.

**A candidate for the solution**

First we randomly choose one node from one of the sub graphs. Based on the probability of choosing an edge which depends on the value of pheromone, we choose one edge. Meanwhile, all the edges connected to the nodes of initial sub graph should be removed. We continue this procedure to reach the last city and then we choose the edge that bring us back to where we start our journey (the initial node).

**Constrains**

We shouldn't forget that we travel to only one city in a province. As mentioned before, we should remove all the edges connected to the other nodes of the province we start our journey.

**The pheromone**

By selecting and forming a candidate solution (a series of nodes), we can calculate the total distance traveled since the edges represent the distance of the cities. For updating the pheromones, we will add a proportion of the inverse of this total distance to the pheromones of the edges selected in this solution (path).

**Useful info.**

We tend to find the path with shortest total distance. Therefore, the inverse of value of edges can be also useful in the process of selecting the edges alongside considering the pheromones.

## 7 Question 7

**a**

The examiner tends to reduce the number of similar questions between individuals. Since cheating is some thing double sided (between two student), we can use  $L_{ij}^2$ . As the number similar questions increases, the chance of cheating will also increase but not linearly.

**b**

In the ACO algorithm we have to define and clarify the following concepts:

- Graph nodes and edges**

We consider a fully connected graph consisting of two sets of nodes. One including 50 nodes, each node stands for a student. The other set consists of 8 subsets representing the type of questions. Each subset contains  $M_i$  nodes representing the questions of the  $i$ -th type.

- **A candidate for the solution**

We begin with the first student. probabilistically based on the pheromones, we choose a edge among the first subset on the questions side and then remove all the edges connected to the nodes of this subset. We continue this process so that each student is assigned with 8 questions from different types of questions.

- **Constrains**

Only one question must be selected from each subset (type) of questions.

- **The pheromone**

Based on the selected edges, we calculate  $L_{ij}$ . Then we define the pheromone of these edges using the following formula:

$$\phi = \frac{c}{\sum_{i,j, i \neq j} L_{ij}^2}$$

- **Useful info.**

In the selection procedure, we can consider the history of selected questions for previous students. In this way, we choose the questions previously chosen for previous students with less probability.