Assignment 3: GA and ACO
(Due Date: July 20, 2020- 11:59PM)

Question 1 [50 marks]: Genetic Algorithms and PID Controller Design

GA can be used to design a PID (Proportional-Integral-Differential) controller for closed loop plant control systems. A common configuration is shown in the following figure 1.

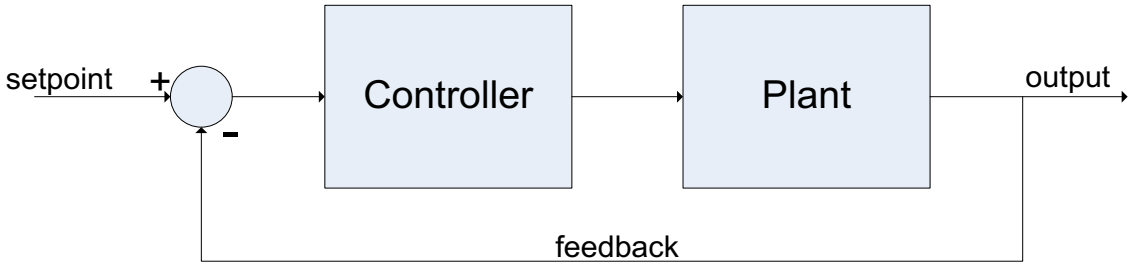

Figure 1. Closed-loop control system

Notice that the output of this system has an explicit impact on the input of the controller, which explains the "closed-loop" term. In this loop, the output of the system is what we would like to control to the set-point. So, the controller is being fed by the error.

The PID controller is specified by a transfer function of the form:

$$G_C(s) = K_p(1 + \frac{1}{T_I s} + T_D s)$$

Where K_p, T_I, and T_D are the controller parameters. The objective of the design is to obtain the values of these parameters that optimize the performance of the system. For a given plant transfer function, the performance of the system is typically evaluated by the step response of the system which is of the form shown in figure 2.

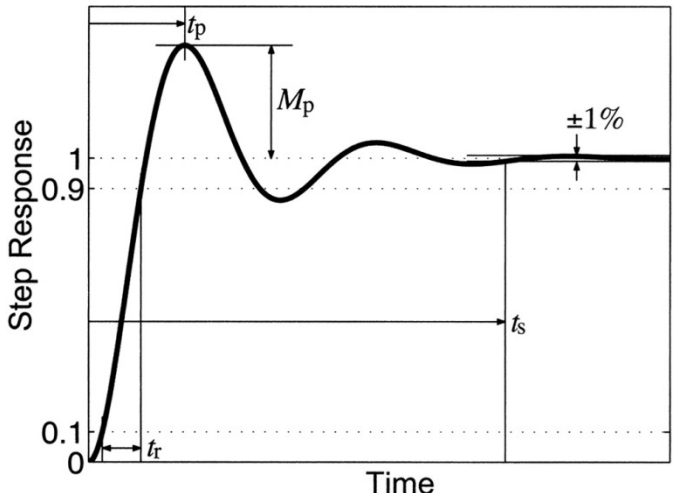

Figure 2. Step response parameters

The performance is measured in terms of integral squared error (ISE), $ISE = \int_0^T (e(t))^2 dt$ which is the integral of the square of the difference between the output and the steady state value (1 in this case), along with the values of the step response parameters: t_r, the rise-time; t_s, the settling time, and M_p, the maximum overshoot magnitude, which can be addressed as the percentage of the steady-state output. t_r and t_s are important, as they demonstrate how fast a system can work. M_p is of importance as well, as we may abide by the plant restrictions. So it is important to minimize all these performance measures.

In this problem you are provided by a function that gives you the performance measures if you provide it with values for the parameters K_P, T_I, and T_D. The function is [ISE,t_r,t_s,M_p] = Q1_perfFCN([K_p;T_i;T_d]. It is a Matlab code that uses other functions of the Matlab control tool box.
Assume that the values of the controller parameters are in the ranges:
$K_p \in (2,18), T_I \in (1.05,9.42), T_D \in (0.26,2.37)$

a) Develop a suitable representation for the solutions with precision of 2 decimal points.
b) Formulate a fitness function that you can use to evaluate a solution.
c) Implement GA algorithm to solve this problem, use a population of 50 individuals, number of generations of 150, crossover probability of 0.6 and mutation probability of 0.25. Use FPS parent selection strategy and an elitism survival selection strategy keeping the best two individuals across generations. Select a crossover and mutation operators and solve the problem.
d) In this part we would like to study the effect of the choice of the GA control parameters:
   i) Experiment with 3 different values for the number of generations and report the progression of the solutions as a function of the different number of generations.
   ii) Repeat i) for 3 different population sizes.
   iii) Experiment with different crossover and mutation probabilities (within the guidelines given in class) and report the results.

Deliverables: a report that contains:

a) [4 marks] For part (a) provide a summary on the representation for the solutions. Justify your choice.
b) [5 marks] For part (b) provide a summary on the fitness function that you used to evaluate a solution. Justify your choice.
c) [20 marks] For part (c) provide a Plot(s) of the fitness of the best solution in each generation throughout the generations. That is the x-axis is generation no., the y-axis is the fitness value of the best solution in that generation.
d) For part (d):
    i) [7 marks] Plot the progress of the solution fitness as a function of the 3 different values for the number of generations. Provide your insight to the behavior of the search as the number of generations is increased.
    ii) [7 marks] Same as in (i) but for the population experiment.
    iii) [7 marks] Same as in (ii) but for the crossover/mutation experiment.

The supplementary files and an article on PID controller design are posted in this assignment module.

Question 2 [50 mark]: Ant Foraging Behavior and TSP

a) **NetLogo**[1] is a high-level multi-agent modelling environment, very suitable for fast creation of agent-based models. NetLogo has a large library of sample models. The model "ANTS" demonstrates a colony of ants forages for food. Though each ant follows a set of simple rules, the colony as a whole acts in a sophisticated way. The first part of this question is to experiments on the NetLogo's ANTS model. The model provides control sliders to change the model parameters.

Run experiments with population (30, 50, 100), diffusion rate (40, 80), evaporation rate (10, 20) and different placements of the food sources. Examining the ant colony's food foraging and transporting behavior (finish time).

b) The target problem in this question is the 29-city TSP (known as Bays29 in literature). The distance between cities is symmetrical, and Euclidean distance measure is used. Each city must be visited once and only once. The objective is to minimize the sum of the Euclidean distances of a complete TSP tour. Ignore curvature of the earth. The 29 city coordinate matrix is listed below.

City #	X-coord.	Y-coord.
1	1150.0	1760.0
2	630.0	1660.0
3	40.0	2090.0
4	750.0	1100.0
5	750.0	2030.0
6	1030.0	2070.0
7	1650.0	650.0
8	1490.0	1630.0
9	790.0	2260.0
10	710.0	1310.0
11	840.0	550.0
12	1170.0	2300.0
13	970.0	1340.0
14	510.0	700.0
15	750.0	900.0
16	1280.0	1200.0
17	230.0	590.0
18	460.0	860.0
19	1040.0	950.0
20	590.0	1390.0
21	830.0	1770.0
22	490.0	500.0

[1] NetLogo is a multi-agent programmable modeling environment. It is used by many tens of thousands of students, teachers and researchers worldwide. It also powers HubNet participatory simulations. It is authored by Uri Wilensky and developed at the CCL. You can download it free of charge. You can also try it online through NetLogo Web.

23	1840.0	1240.0
24	1260.0	1500.0
25	1280.0	790.0
26	490.0	2130.0
27	1460.0	1420.0
28	1260.0	1910.0
29	360.0	1980.0

For this problem do the following:
1. Code a simple ACO algorithm to solve the problem. To do this you need to select a transition rule, select online and offline pheromone update rules, set a population size and select a stopping criterion.
2. Run your ACO algorithm.
3. Perform the following changes on your ACO code (one by one) and compare the results.
   a) Change the values of pheromone persistence constant 3 times
   b) Change the values of state transition control parameter 3 times
   c) Change the population size twice
   d) Turn off online pheromone update

Deliverables: a report that contains

- [10 marks] For part (a) report your observations on the behavior of the colonies for the different parameters.
- For part (b):
  - [5 marks] (2) Report the details of your ACO algorithm specifications.
  - (3)
    - [5 marks] (a) Plot the solution progression for each pheromone persistence constant.
    - [5 marks] (b) Same as in (3)(a) but for the state transition control parameter
    - [5 marks] (c) Same as in (3)(a) but for the population size parameter
    - [5 marks] (d) Same as in (3)(a) but for the case where the pheromone update is turned-off for each case of the above three scenarios.
- [15 marks] Summarize your observations and conclusions on the behavior of your ACO algorithm as you change its control parameters.