POLYTECHNIC SCHOOL
DEPARTMENT OF COMPUTER AND INFORMATION TECHNOLOGY ENGINEERING

COMPUTATIONAL INTELLIGENCE

Teachers: S. Lykothanasis, D. Koutsomitropoulos

Academic Year 2022-2023

Laboratory Exercise
Part B

B. Implementation of body postures by a humanoid robot using GA.

In this paper we will use **GENETIC Algorithm** to determine the optimal sensor value in order for a humanoid robot to perform a certain posture. For this purpose you will utilize the measurements from the PUC-Rio dataset used in part A as well.
The goal of the algorithm is to determine the optimal value of the sensors that the robot's accelerometers should have in order to realize the *sitting* state.

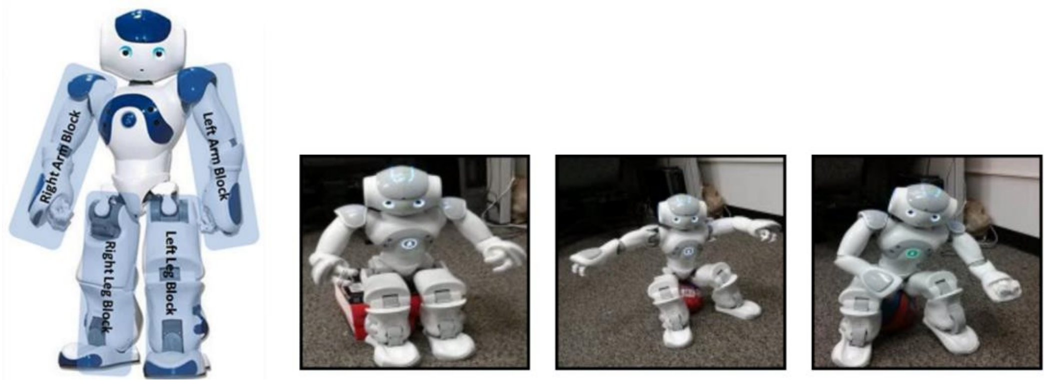

Figure 1: The NAO robot and sitting posture implementation.[1]

We assume that the robot has sensors in the same positions as human users (waist, thigh, ankle, arm). We also assume that the optimal values are derived from the M.O. of the human measurements for this condition. At the same time, these optimal values should be far from the M.O. of the other states to allow the robot to implement the movement/stance as discretely and independently of the other body postures as possible.

B1. GA design *[30 credits]*

α) CODING: propose a coding for individuals in the population. Consider the following:

i. An atom represents a vector corresponding to the values of the 4 sensors carried by the robot (12 values in total).

[1] Lakaemper, R. (2014). sitting pose generation using genetic algorithm for nao humanoid robots. in 2014 IEEE International Workshop on Advanced Robotics and its Social Impacts (pp. 137-142). iEEE.

ii. The range of these values may affect the similarity metrics and the fitness function you will use below. It is suggested to use preprocessing/normalization as you did in A.

b) Surplus prices: Depending on the coding applied in (a) it is possible that redundant values may occur, for example, values outside the range of sensor values or the normalization range. Describe how you will deal with this problem. Consider whether you can avoid redundant values, based on the coding you proposed in (a).

c) INITIAL population: describe a process for creating an initial population of individuals. The atoms in the population are potential sensor value vectors for the robot to perform the sitting stop.

(d) SIMILARITY CALCULATION: You need to calculate the distance of an atom from the M.O. of the dataset vectors corresponding to the sitting state. Also, you need to calculate the distance of an atom from each M.O. corresponding to each of the other states. Several metrics can be used to calculate the distance, such as *Euclidean distance*, *Manhattan distance, cosine* and *Pearson correlation*. Use cosine similarity and comment on its suitability, relative to the others, for this particular case.

(e) Suitability function: A person is more suitable than others if:

1. It is closer to the sensor readings for the specific situation/state.
2. It is further away from the sensor readings corresponding to the other states.

Therefore the fitness function should combine these two criteria with an emphasis on 1° of them. To calculate distance/similarity you can use averages and cosine similarity as mentioned in (d). A fitness function *F* could be:

$$F(v) = \frac{\cos(v,\ t_s) + c\,(1 - \frac{1}{4}\sum_{i \neq s} \cos\,(v, t_i))}{1 + c}$$

Where v is the vector of the individual in the population, t_i is the vector representing the M.O. of state *i*, *s* is the sitting state, and *c is an* appropriate constant. The function cos moves in the interval [-1, 1] or only in [0, 1] if there are no negative vectors.

i. What are the maximum and minimum values of F? Is it possible to obtain negative fitness values? If so, describe how to avoid them.
ii. Explain whether the above formula is a suitable choice for the fitness function.
iii. Specify a value for the constant c, so that minimizing similarity with other situations does not dominate the evaluation of an individual, but its effect is not negligible.

(f) GENETIC Operators: Based on your chosen coding, suggest the selection, crossover and mutation operators to use.

i. Especially for selection, evaluate the use of *roulette based on cost, based on rankings* and *tournaments.*
ii. Specifically for cross-validation, evaluate the suitability of the following operators: *Single-point crossing, multi-point crossing, uniform crossing*.
iii. Especially for mutation, evaluate the use of *elitism*.

B2. Implementation of GA *[30 credits]*
Write a program, in any environment or programming language, that implements the genetic algorithm you have designed.

B3. Evaluation AND Impact of Parameters *[40 credits]*

α) Run the algorithm for the parameter values shown in the table below and complete it. The algorithm will terminate when one or more of the *termination criteria* are met, i.e. when:

i. the best individual of each generation ceases to improve for a certain number of generations, or
ii. improves below a percentage (<1%) or
iii. a predefined number of generations (e.g. 1000) has been exceeded

A/N	POPULATION SIZE	LIKELIHOOD OF CROSSING	LIKELIHOOD OF MUTATION	AVERAGE PRICE BELTISTOU	MEDIUM NUMBER OF GENES
1	20	0.6	0.00		
2	20	0.6	0.01		
3	20	0.6	0.10		
4	20	0.9	0.01		
5	20	0.1	0.01		
6	200	0.6	0.00		
7	200	0.6	0.01		
8	200	0.6	0.10		
9	200	0.9	0.01		
10	200	0.1	0.01		

Caution: *because GAs are stochastic algorithms and therefore do not guarantee the same performance in each execution, you should run the algorithm at least ten times for each case. In the table, note the average performance of the best solution in each run.*

b) For each case in the above table, plot the evolution curve (performance/number of generations) of the best solution (the average value of this solution, in each generation, in each run).

c) Based on these curves and the results of the above table, state in detail your conclusions about the effect of each parameter (population size, crossover probability, mutation probability) on the convergence of the algorithm.

d) For each case in the above table, enter the optimal individual obtained as input to the optimal TND from part A and record the state with the highest probability of being obtained, as well as the probability of the sitting state. State your conclusions following (c).

Deliverables

Your report should contain a comprehensive commentary on your experiments and a full record of your results and conclusions, by sub-question. You should also include at the beginning of your report a link to the code you have used (in a file sharing service or code repo), so that it can be taken into account.

Don't forget to fill in your details at the top of the 1ης page.

Evaluation

The answer to questions A and B has a weight of 20% in the final grade of the course (the total of both parts of the paper has a weight of 40%). The Bonus grade (10%) is added to the above 40%.

Comments

1. The report, in electronic form, must be posted in e-class by Monday, 12/6/2023, at 23:59.
2. For any clarification/question you can use the relevant forum in the eclass of the course.