Evolutionary Algorithms :TP1

Find the Maximum value of the function using simulated annealing algorithm

Maximize $\boldsymbol{f(x) = (x^2 + x)\cos(2x) + x^2}$

$\boldsymbol{-11 < x < 11}$

You need to:

- Randomly create the first population of real values between -20 and 20
- The population size is 100
- In every iteration,select one individual to mutate from the parents list using roulette wheel selection operator.
- After selecting the parent the child is formed by adding a random noise(between -0.51 and 0.51),(gaussian mutation)
- No crossover operator is needed
- Maximum permissible number of iteration is n = 100

Evolutionary Algorithms :TP2

In this TP we will use a genetic algorithm to optimize the parameters of an artificial neural network.

In the genetic algorithm the individuals in the population are composed of activation, solver, the number of neurons of the first hidden layer and the number of neurons of the second hidden layers.

You need to:

- Using this two lists

  activation = ['identity','logistic', 'tanh', 'relu']

  solver = ['lbfgs','sgd', 'adam']

- Each individual in the population is a list with 4 elements
  - Random activation from the activation list
  - Random solver from the solver list
  - Random number between 2 and 100 of neurons for the first hidden layer
  - Random number between 2 and 100 of neurons for the second hidden layer
- The size of the population is 10
- Sort the population from best to worst in term of fitness function
- Using single point crossover create children by combining the parents from the sorted list.
- The fitness function is calculated from the accuracy of the neural network, The objective of the genetic algorithm is to maximize the accuracy of the neural network.
- The mutation has a probability of 30%, done by adding a random value between 1 and 10 to either the number of first or second neurons of the hidden layers
- Maximum number of iterations is 10.
- Return the ANN parameters that gives the best results
- Use any dataset of your choice (example: iris dataset)

Evolutionary Algorithms :TP3

In this TP you will learn how to code a self adaptive genetic algorithm to find the best parameters for a hard support vector machine

The genetic algorithm must have a self adaptive :

- probability of mutation pm,
- probability of crossover pc,
- population size Np...

You need to:

- Use an adaptive population size, initial population size is n= 100
- Use random selection
- Use one point crossover with an adaptive rate, initial rate is rc=0.9
- Use gaussian mutation with an adaptive rate,initial rate is rm=0.1
- Maximum number of iteration is 100
- Return the parameters of SVM that give the best results
- Use any dataset of your choice

Evolutionary Algorithms :TP4

In this TP we want to learn how to use a multiobjective genetic algorithm to find the best parameters for a soft support vector machine which gives the best classification results.

You should use the NSGA2 algorithm for this optimization problem.

The NSGA2 algorithm should be coded from scratch.

You need to:

- Randomly create the first population
- The population size is 100
- Use roulette wheel selection operator
- Set your 2 objective functions
- Use one point crossover with a rate rc = 0.9
- Use gaussian mutation with a rate rm = 0.1
- Maximum number of iterations is 100
- Return the parameters of SVM that give the best results
- Use any dataset of your choice

Artificial Immune Systems :TP1

Use CLONALG (The CLONal selection ALGorithm) to find the maximum value of the function

$$maximize\ f(x,y) = x.sin(4\pi x) - y.sin(4\pi y + \pi) + 1$$

$$\mathbf{-1 < x < 2}$$

$$\mathbf{-1 < y < 2}$$

You need to:

- Randomly create the first population of real values
- The population size is 200
- Use Gaussian mutation with a probability of 5%
- Clonal operator with a rate of rc = 0.5
- Maximum permissible number of iteration is n = 100

Artificial Immune Systems :TP2

Negative selection algorithm is mostly used for classification and anomaly/ fault detection

Use the algorithm to classify any data of your choice

You can use for example the negative selection algorithm to filter out spam emails by analyzing the content of incoming emails.

You need to:

- Create the code for the detector generation procedure for negative selection algorithm
- Create the appropriate code for the detector application procedure for the negative selection algorithm

Artificial Immune Systems :TP3

In this TP we want to learn how to use a multiobjective immune systems to find the best parameters for a soft support vector machine which gives the best classification results.

You should use a clonal selection based multiobjective algorithm.

You need to:

- Randomly create the first population
- The population size is 100
- Set your 2 objective functions
- Use roulette wheel selection operator
- Use Gaussian mutation with rate of rm=0.1
- Clonal operator with a rate of rc = 0.5
- Maximum permissible number of iteration is n = 100
- Return the parameters of SVM that give the best results
- Use any dataset of your choice