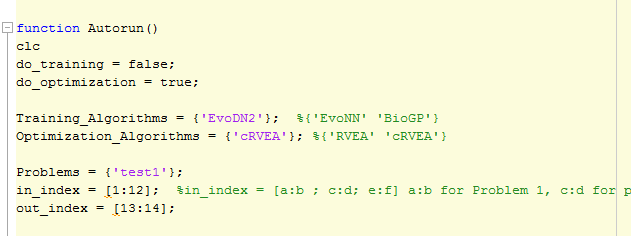
**Rules:**

**Training**

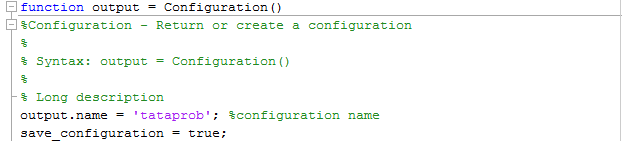
1. Save file in .xlsx format.

2. Open Autorun.m

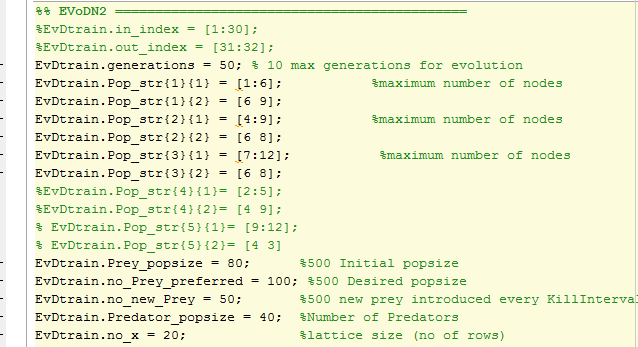


1. change do\_training to **true** for training and do\_optimization to **true** for optimization. (if only training is required, keep do\_training as true and do\_optimization as false and vice-versa).
2. In Training\_Algorithms put EvoDN2 as shown
3. In Optimization\_Algorithms put cRVEA as shown for EvoDN2 trained models.
4. Suppose the datasheet is 'test1.xlsx', put only 'test1' in Problems as shown.
5. in\_index = [input variable coloumn no.s]
6. out\_index = [output variable coloumn no.s]

3. Open Configuration.m



put output.name as suites you. In the example, a folder named as tataprob will be created in output parent folder and all results will be inside that folder.



Go to the EvoDN2 section (In EvoDN2, the variables can be distributed into subsets and each subset is trained seperately).

1. In the example, the dataset is divided into three subsets.

Each subset is named as EvDtrain.Pop\_str{subset no.}{1}. Here put the variable coloumn nos you choose to put in the subset.

EvDtrain.Pop\_str{subset no.}{1} = [variable coloumn no.s]

Similarly, do it for as many subsets you want. But each variable coloumn should be used at least once.

EvDtrain.Pop\_str{subset no.}{2} denotes [ no. of variable coloumns used; no. of nodes in the subsets]

For each subset, there will be a EvDtrain.Pop\_str{subset no.}{1} and EvDtrain.Pop\_str{subset no.}{2}.

e.g:

EvDtrain.Pop\_str{2}{1} = [4:9];

EvDtrain.Pop\_str{2}{2} = [6 (no. of variables in subset\_2 is (9-4+1)= 6) 8 (no. of nodes as per my choice: more the nodes better the fitting but avoid overfitting)];

Rest are same as in EvoNN parameters.

**After doing these changes run Autorun.m.**

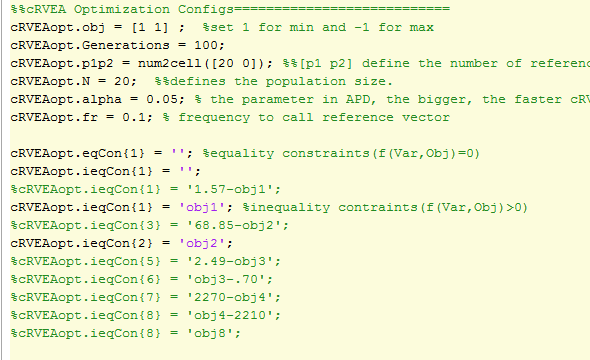
The output results and svr results will be obtained and saved in the same folder by its own.

**Optimization**

Once the training is done, the Y.mat files will be created in the output folder.

Open Configuration.m

Go to cRVEA section.



put cRVEAopt.obj = [ 1 1 -1] if we want to minimize f1 and f2 and maximize f3.

increase Generations for better convergence and keep others constant.

If no constraints are required put,

cRVEAopt.eqCon{1} = '';

cRVEAopt.ieqCon{1} = '';

if f(Obj)= 0 constraints are there,

then use

cRVEAopt.eqCon{1} = 'a\*obj1+b\*obj2+c\*obj3...';

if f(Obj)> 0 constraints are there,

then use

cRVEAopt.ieqCon{1} = 'a\*obj1+b\*obj2+c\*obj3...';

Put as many constraints as required as shown in figure.

Open Autorun.m

Check if do\_optimization = true in **Autorun.m** and run

A cRVEAopt.mat file will contain the datapoints and similarly cRVEA plot will be there.