SCT Experiment No: 8

Name: Abhishek S Waghchaure

PRN: 1032221714

Dept: FY M Tech DSA(2022-24)

Aim:

Demonstrate multi-objective optimization using tools and libraries in Matlab/python/R

Introduction:

Multi-objective optimization (also known as multi-objective programming, vector optimization, multicriteria optimization, multiattribute optimization or Pareto optimization) is an area of multiple criteria decision making that is concerned with mathematical optimization problems involving more than one objective function to be optimized simultaneously. Multi-objective optimization has been applied in many fields of science, including engineering, economics and logistics where optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. Minimizing cost while maximizing comfort while buying a car, and maximizing performance whilst minimizing fuel consumption and emission of pollutants of a vehicle are examples of multi-objective optimization problems involving two and three objectives, respectively. In practical problems, there can be more than three objectives.

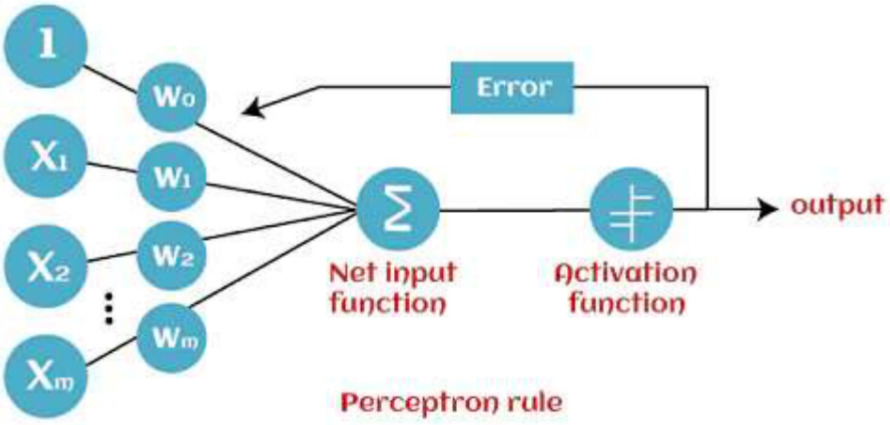
Example:

Minimize the costs and Emissions in power plants
Maximize the reliability while minimizing the costs
Minimizing the investment while maximizing the satisfaction

There are several approaches to address these types of problems such as weighted sum approach or Epsilon constrained method.

The problem with weighted sum approach is that it is applicable when the objective functions are of the same dimensions (costs, weight, time) and even if they re of similar dimensions if the magnitude of them are not very similar then weighted sum is not very efficient. Additionally, if the preference of the decision maker is changed, then the problem should be resolved with a new set of weighting factors.

Alternatively, the epsilon constraint will provide the decision maker with a set of Pareto optimal solutions were none of the solutions would dominate the other ones. It provides the Pareto optimal solutions.

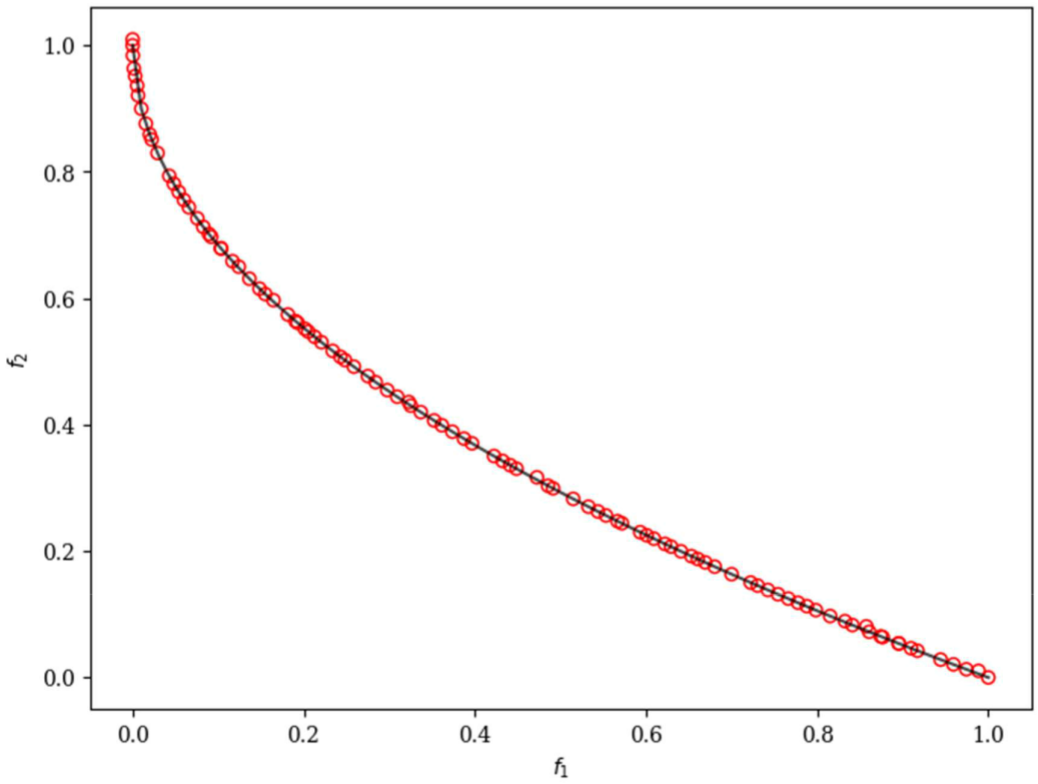

1
Wo
X1
W1
X2
W2
Xm
Wm
Error
output
Net input function
Activation function
Perceptron rule

Code:

```
from pymoo.algorithms.moo.nsga2 import NSGA2
from pymoo.problems import get_problem from
pymoo.optimize import minimize
from pymoo.visualization.scatter import Scatter

problem = get_problem("zdt1")

algorithm = NSGA2(pop_size=100)

res = minimize(problem,
               algorithm,
               ('n_gen', 200),
               seed=1,
               verbose=False)

plot = Scatter()
plot.add(problem.pareto_front(), plot_type="line", color="black", alpha=0. 7)

plot.add(res.F, facecolor="none", edgecolor="red")
plot.show()
```

Out[3]:

```
<pymoo.visualization.scatter.Scatter at 0x1fe366cfb50>
```

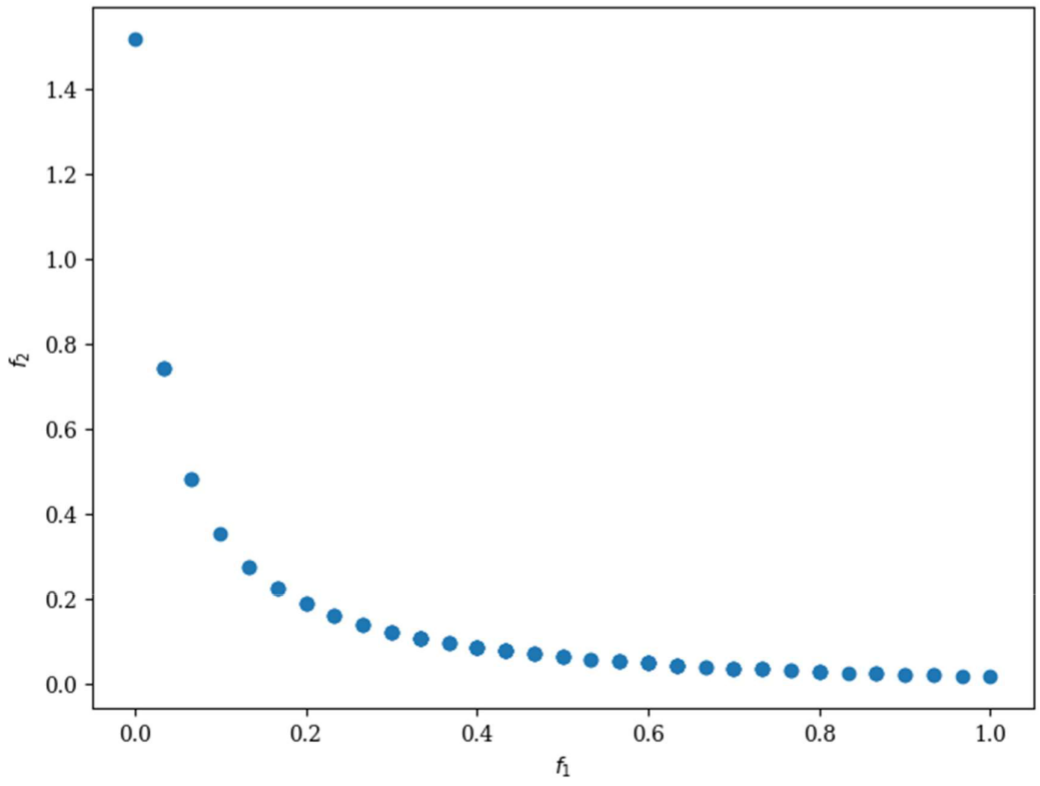


In [4]:

```
from pymoo.algorithms.moo.nsga2 import NSGA2
from pymoo.problems import get_problem
from pymoo.operators.crossover.pntx import TwoPointCrossover
from pymoo.operators.mutation.bitflip import BitflipMutation from
pymoo.operators.sampling.rnd import BinaryRandomSampling from
pymoo.optimize import minimize
from pymoo.visualization.scatter import Scatter

problem = get_problem("zdt5")

algorithm = NSGA2(pop_size=100,
                  sampling=BinaryRandomSampling(),
                  crossover=TwoPointCrossover(),
                  mutation=BitflipMutation(),
                  eliminate_duplicates=True)

res = minimize(problem,
               algorithm,
               ('n_gen', 500),
               seed=1,
               verbose=False)

Scatter().add(res.F).show()
```

Out[4]:

```
<pymoo.visualization.scatter.Scatter at 0x1fe366c0b50>
```



Conclusion:

The problems in MOO are numerous and can be found in various areas of human life. Along with this, there are also many methods in solving the MOO problem. In this MOO review, there are two conclusions. Firstly, there are two MOO methods that do not require complex mathematical equations so that the problem becomes simplified, namely the Pareto and scalarization methods. In the Pareto method, there are dominated solutions and non-dominated solutions that can be described in POF. The non-dominated solution is obtained through the continuously updated algorithm. The scalarization method creates multi-objective functions into a single solution, and the weights are first determined. The weights on the scalarization method are equal weights, ROC weights, and RS weights. Second, the solution from the Pareto method is a performance indicator that forms MOO a separate and produces a compromise solution and can be displayed in the form of POF. Meanwhile, the solution with the scalarization method is in the form of performance indicators that form the scalar function that is incorporated in the fitness function.

Reference:

https://in.mathworks.com/help/optim/ug/large-scale-problem-based-quadratic-programming.html

https://www.linkedin.com/pulse/multi-objective-optimization-pyomo-alireza-soroudi

https://codemonk.in/blog/a-gentle-introduction-to-multi-objective-optimization/

https://ieeexplore.ieee.org/document/996017