

# RANDOM SEARCH

- It explores decision variable space (parameter space) of an objective function sequentially in a seemingly random fashion to find optimal point that optimizes the objective function.

- It is based on finding the minimum point using random numbers.

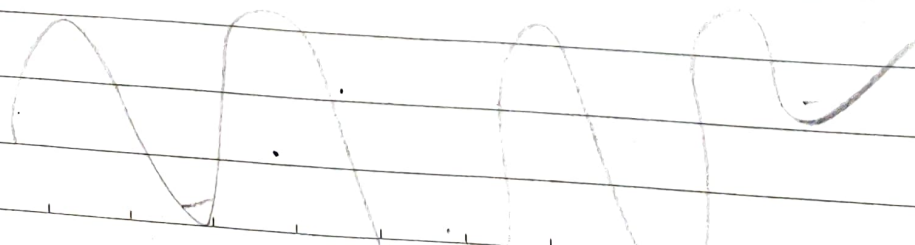
- Strengths

- 1) Simple, easy to understand.
- 2) Can be customized for specific application.

- 3) Can converge to global optimum solution if search space is compact.

- Weaknesses

Practically, for complex problems finding globally optimum solution can take indefinite amount of time.



# PRIMITIVE RANDOM SEARCH (For multivar problem)

Let  $f(\vec{x})$  be an objective function to be minimized and  $\vec{x}$  be the point under consideration.

- Primitive algorithm iterates over following steps.

Step 1: Choose a start point  $\vec{x}$  as the current point.

Step 2: Add a random vector  $\vec{d}_x$  to the current point  $\vec{x}$  in the decision variable space & evaluate the objective function at the new point  $\vec{x} + \vec{d}_x$ .

Step 3: If  $f(\vec{x} + \vec{d}_x) < f(\vec{x})$ ,  
Set current point,

$$\vec{x} = \vec{x} + \vec{d}_x$$

Step 4: Stop if maximum number of function evaluations is reached.

Otherwise, go back to step 2 to find a new point.

Stochastic approach is truly a random method because search directions are purely guided by a random number generator.

WAYS TO IMPROVE ?

Based on following observations:

Observation 1: If search in a direction results in a higher objective function, the opposite direction can often lead to a lower objective function.

Observation 2: Successive successful searches in a certain direction should bias subsequent searching toward this direction.

On the other hand, successive failures in a certain direction should discourage subsequent searching along this direction.



## MODIFIED RANDOM SEARCH

- First observation leads to reverse step in the original method.
- Second observation motivated the use of bias term.

### Algorithm:

Step 1: Choose a start point  $\vec{x}$  as the current point. Set initial bias  $\vec{B}$  equals to a zero vector.

Step 2: Add a bias term  $\vec{B}$  and a random vector  $d_x$  to the current point  $\vec{x}$  in the input space & evaluate the objective function at the new point at  $\vec{x} + \vec{B} + d_x$ .

Step 3: If  $f(\vec{x} + \vec{B} + d_x) < f(\vec{x})$ , set current point as,

$$\vec{x} = \vec{x} + \vec{B} + d_x \quad \text{and}$$

Step 4: if  $\vec{B} = 0.2\vec{B} + 0.4d_x$  and

goto step 6, otherwise go to next step

step 4 : If  $f(\vec{x} + \vec{B} - \vec{dx}) < f(\vec{x})$ , set  
current point as,

$$\vec{x} = \vec{x} + \vec{B} - \vec{dx} \quad \text{and}$$

$$\vec{B} = \vec{B} - 0.4 \vec{dx} \quad \text{goto step 6.}$$

otherwise, goto next step.

step 5 : Set the bias,  
 $\vec{B} = 0.5 \vec{B}$  and goto step 6.

Step 6 : Stop if the maximum no. of function  
evaluations is reached. Otherwise goto  
step 2. to find a new point.