Random Walk

October 31, 2019

[1]:
```
import numpy as np
import random
import matplotlib
import matplotlib.pyplot as plt

# random walk
def random_walk(gmax, F, M, D, f):
    M_low, M_hi = list(zip(*M))

    xg = np.random.uniform(low=M_low, high=M_hi, size=D)
    for g in range(gmax):
        vg = xg + F * np.random.normal(loc=0, scale=1.0)

        if f(vg) <= f(xg):
            xg = vg

    return xg


def sphere(x):
    return np.dot(x, x.T)
```

[2]:
```
#differential mutation
def diffrentital_mutation(gmax, F, M, D, f,N):
    M_low, M_hi = list(zip(*M))
    pop=[]
    for i in range(N):
        pop.append(np.random.uniform(low=M_low, high=M_hi, size=D))


    for g in range(gmax):
        for i in range(N):
            xg=pop[i]

            r1_index,r2_index=random.sample(range(1, N), 2)
            while(r1_index==i or r2_index==i):
                r1_index,r2_index=random.sample(range(1, N), 2)
```

```
            r1= pop[r1_index]
            r2= pop[r2_index]

            vg = xg + F * np.subtract(r1,r2)
            if f(vg) <= f(xg):
                xg = vg

    return xg
```

[3]:
```
maximum_generation = 15
mutation_scale = 0.1
search_space = [(-10, 10)]
number_dimensions = 5
N=10
best_random_walk = random_walk(maximum_generation,
                               mutation_scale,
                               search_space,
                               number_dimensions,
                               sphere)

best_differential_mutation = diffrentital_mutation(maximum_generation,
                               mutation_scale,
                               search_space,
                               number_dimensions,
                               sphere,N)
```

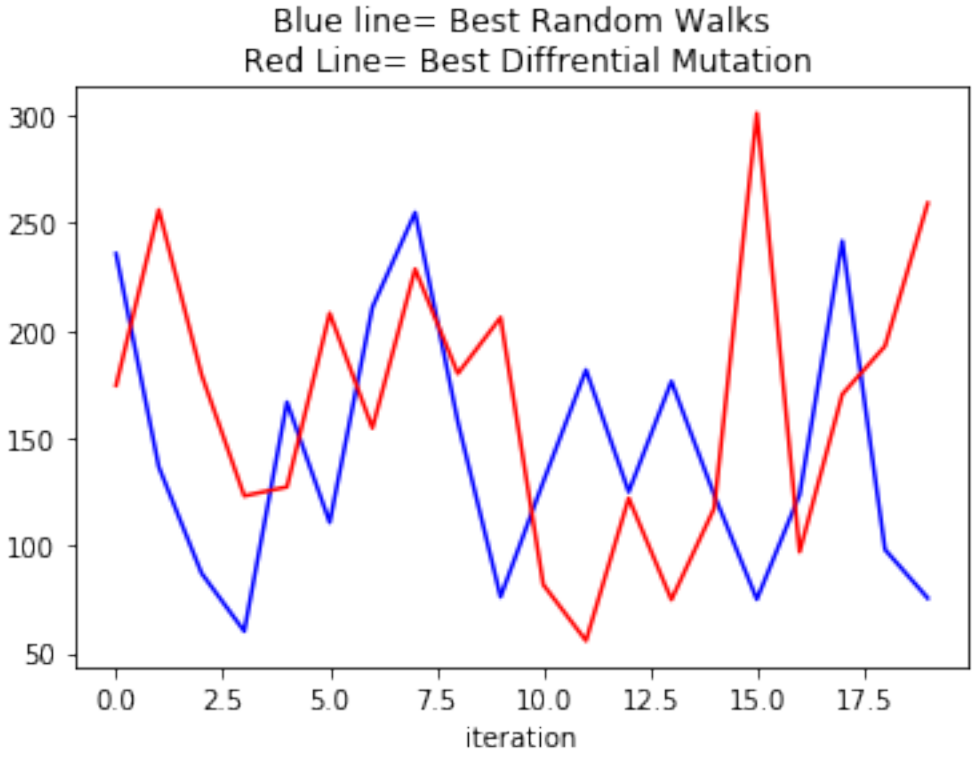
[4]:
```
print(sphere(best_random_walk))
print(sphere(best_differential_mutation))
```

```
89.3534890794522
110.66334535675172
```

[5]:
```
random_walk_val=[]
best_differential_mutation_val=[]
for i in range(20):
    best_random_walk = random_walk(maximum_generation,
                               mutation_scale,
                               search_space,
                               number_dimensions,
```

```
                                sphere)
    random_walk_val.append(sphere(best_random_walk))

    best_differential_mutation = diffrentital_mutation(maximum_generation,
                                mutation_scale,
                                search_space,
                                number_dimensions,
                                sphere,N)

    best_differential_mutation_val.append(sphere(best_differential_mutation))
```

[6]:
```
plt.plot(random_walk_val,'-b', label='Best Random Walks')

plt.plot(best_differential_mutation_val,'-r', label='Best Differential Mutation')

plt.xlabel('iteration')
plt.title('Blue line= Best Random Walks\n Red Line= Best Diffrential Mutation')


plt.show()
```



[ ]:

[ ]: