

```
1: import global_random_search
2: import lib
3: import numpy as np
4: import sg
5: import matplotlib.pyplot as plt
6: from matplotlib.lines import Line2D
7: import pandas as pd
8:
9: f = {
10:     "function": lib.f_real,
11:     "gradient": lib.f_grad,
12: }
13:
14: g = {
15:     "function": lib.g_real,
16:     "gradient": lib.g_grad,
17: }
18:
19:
20: def gradient_descent_constant(step_size=0.0065, start=[0, 0], funcs=f, max_iter=10000, exp="exp/aii-gd-constant.csv"):
21:     start = np.array(start)
22:     g = sg.StochasticGradientDescent()
23:     g.max_iter(max_iter)
24:     g.step_size(step_size)
25:     g.start(start)
26:     def function_generator():
27:         while True:
28:             yield funcs["function"], funcs["gradient"]
29:     g.function_generator(function_generator())
30:     g.debug(True)
31:     g.alg("constant")
32:     for i in range(max_iter):
33:         g.step()
34:         yield {
35:             "x": g._x_value,
36:         }
37:
38: if __name__ == "__main__":
39:     res = list(gradient_descent_constant(max_iter=1000))
40:     res = pd.DataFrame(res)
41:     res["f(x)"] = res["x"].apply(f["function"])
42:     print(res)
43:
44:     ps = [{"min": 0, "max": 10}, {"min": 0, "max": 18}]
45:     grs = global_random_search.a(costf=f["function"], parameters=ps, N=1000)
46:
47:     plt.figure()
48:
49:     print(res["f(x)"], len(res["f(x)"]))
50:     plt.plot(list(range(len(res["f(x)"]))), res["f(x)"], label="gradient descent", color="black")
51:     costs = grs['stats']['it_best_costs']
52:     plt.plot(list(range(len(costs))), costs, label="global random search", color="orange")
53:     plt.title("Global Random Search vs Gradient Descent on $f(x)$")
54:     custom_lines = [
55:         Line2D([0], [0], color='black', lw=2),
56:         Line2D([0], [0], color='orange', lw=2),
57:     ]
58:     custom_labels = ['gradient descent', 'rnd search a']
59:     plt.legend(custom_lines, custom_labels)
60:     plt.yscale('log')
61:     plt.xlabel('iteration')
62:     plt.tight_layout()
63:     plt.savefig('fig/aii-iterations-f.pdf')
64:
65:     res = list(gradient_descent_constant(max_iter=1000, step_size=0.003, funcs=g))
66:     res = pd.DataFrame(res)
67:     res["f(x)"] = res["x"].apply(g["function"])
68:     print(res)
69:
70:     ps = [{"min": 0, "max": 10}, {"min": 0, "max": 18}]
71:     grs = global_random_search.a(costf=g["function"], parameters=ps, N=1000)
72:
73:     plt.figure()
74:
75:     print(res["f(x)"], len(res["f(x)"]))
76:     plt.plot(list(range(len(res["f(x)"]))), res["f(x)"], label="gradient descent", color="black")
77:     costs = grs['stats']['it_best_costs']
78:     plt.plot(list(range(len(costs))), costs, label="global random search", color="orange")
79:     plt.title("Global Random Search vs Gradient Descent on $g(x)$")
80:     plt.legend()
81:     plt.yscale('log')
82:     plt.xlabel('iteration')
83:     plt.tight_layout()
84:     plt.savefig('fig/aii-iterations-g.pdf')
```