

这周主要学习了 PSO (Particle Swarm Optimization, PSO) 粒子群算法的基本思想和代码编写。因为粒子群和边缘计算模型中的 MD 设备移动寻找最佳 MEC 服务器很相似。

基本思想：通过不断的迭代，记录每个粒子的局部最优和粒子群的全局最优，进行粒子的速度以及位置的更新。

粒子的速度以及位置更新的方式如下：

$$v_i^d = wv_i^d + c_1r_1(p_i^d - x_i^d) + c_2r_2(p_g^d - x_i^d)$$
$$x_i^d = x_i^d + \alpha v_i^d$$

参数说明：

1: i 表示粒子 i, d 表示节点。

2: p_i, p_g 分别为局部和全局最优位置

3: w 是一个非负数，称为惯性因子，对算法的收敛起到很大的作用，其值越大，粒子飞跃的范围就越广，更容易找到全局最优，但是也会错失局部搜寻的能力。

4: 加速常数 c_1, c_2 是非负常数，是调整局部最优值和全局最优值权重的参数

5: r_1, r_2 是 0-1 之间的随机数，a 是约束因子，控制速度的权重。

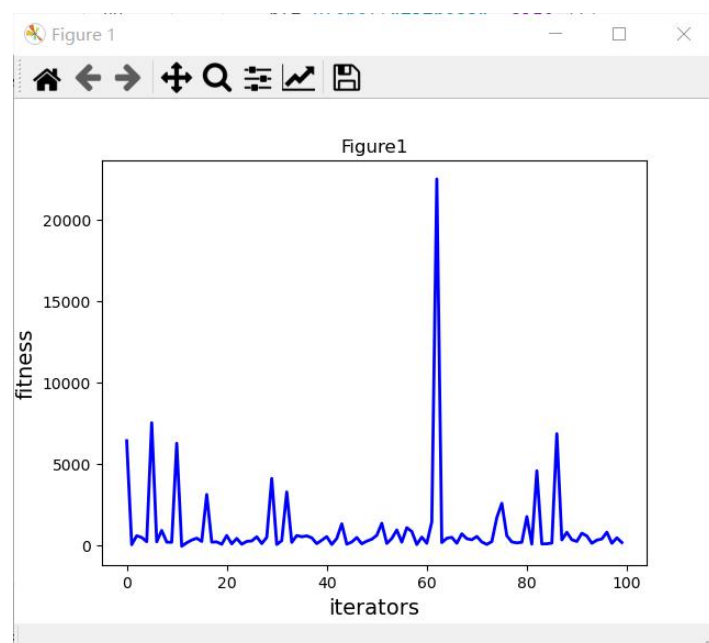
问题描述：寻找 $y = x_1 + 6 \cdot \cos(x_2) + 9 \cdot \sin(x_3) + 10 \cdot \tan(x_4)$ 在 x_1, x_2, x_3, x_4 在 (1, 30) 范围内的最大值。

代码：

`__init__()` 方法初始化参数，包括自变量的上下限，种群大小，迭代代数；

`fitness()` 计算 y 的值；`update_operator()` 根据位置和速度更新粒子下一时刻的位置，挑选出当前代该粒子和种群中所有粒子的最好位置作为历史记录；`main()` 实现整个算法的迭代过程。

我选择使用两百次迭代得到最后的结果如图所示：



可以得到最好的结果为

```
##### Generation 100 #####  
bestplace: [25.01534862 15.22526562 22.15512888 26.70309231]  
g_best: 22477.615486426384
```

但是这个结果与实际值存在较大的差距。

PSO 算法的缺点也是显而易见的：

1: 算法局部搜索能力较差，搜索精度不够高。

2: 算法不能绝对保证搜索到全局最优解，主要有两方面的原因：

① 有时粒子群在俯冲过程中会错失全局最优解。

② 应用 PSO 算法处理高维复杂问题时，算法可能会早熟收敛

3: 算法搜索性能对参数具有一定的依赖性。对于特定的优化问题，如果用户经验不足，参数调整的确是个棘手的问题。参数值的大小直接影响到算法是否收敛以及求解结果的精度。

4: PSO 算法是一种概率算法，算法理论不完善，缺乏独特性，理论成果偏少。