

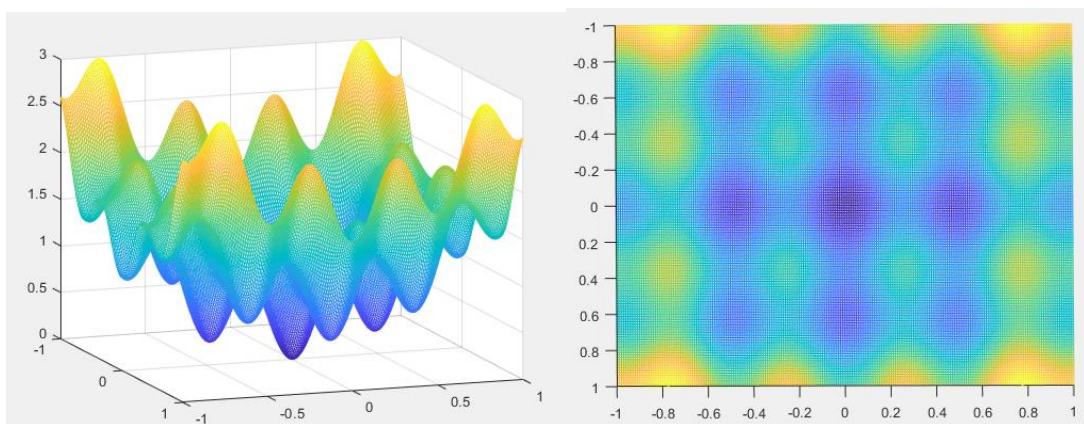
第二次大作业总结报告

无间

对非线性目标函数，求其最小值。

$$f(x_1, x_2) = x_1^2 + x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$$

，其中 $x_1 \in [-1, 1]$ ， $x_2 \in [-1, 1]$ 。



立体图

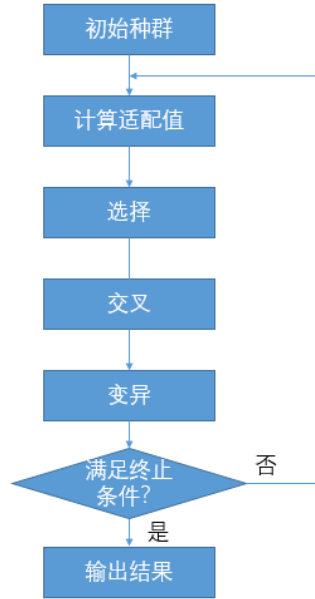
俯视图

如图，从该非线性函数的立体图和俯视图可以看出，该函数存在多个局部极值。最优解为 0，最优解对应的位置为[0 0]

一、GA 算法求解

➤ 理论部分

程序采用基础的 GA 算法。基本算法的流程为：



首先建立二进制字符串与自变量之间的对应关系。 m 表示二进制数的位数。 b 为该二进制串对应的十进制数。

$$x = x_{min} + b \frac{x_{max} - x_{min}}{2^m - 1}$$

选择操作采用排序方法，先对 n 个样本目标函数值进行从小到大排序。 $cost_1 < cost_2 < \dots < cost_n$, 相应的自变量二进制串也跟随排序，自变量与目标函数之间一直保持对应关系。由于此处求解的是一个最小值问题，故取适配值：

$$f_i = \frac{k(n - r_i)}{n}$$

再根据适配值确定每个样本的选择概率。

再对选择后的 n 个样本，建立两两对应关系。进入样本对循环，当随机数 $\text{rand} < \text{交叉概率}$ 时，再产生一个介于 $1-2n-1$ 的整数随机数 k ，将此对样本二进制串从第 k 位起进行交换。

完成交叉操作后，进行最后一步变异操作，进入样本循环，

当随机数 $\text{rand} < \text{变异概率}$ 时，再产生一个介于 $1-2n$ 的整数随机数 k ，将此样本的二进制串的第 k 位由 0 变为 1 或由 1 变为 0。

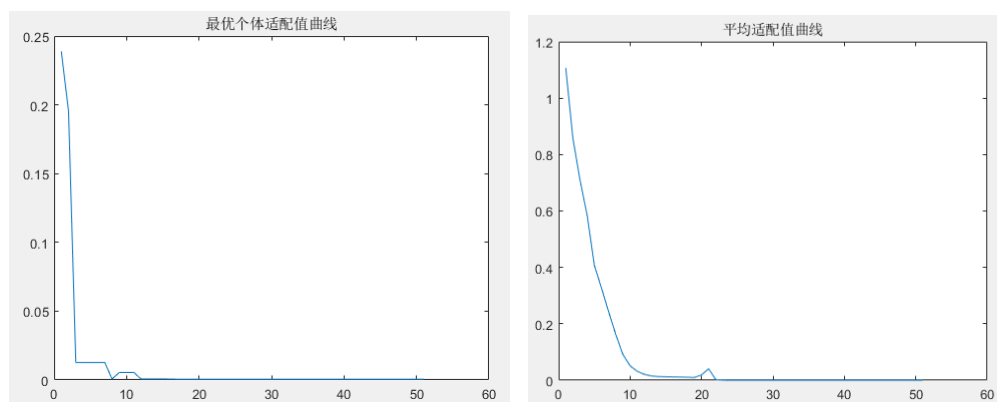
迭代搜索的终止条件为平均适配值变化 $< 1e-8$ 或者最大迭代次数超过 2000。

➤ 超参数讨论

下面进行超参数讨论，在讨论单个参数时，其他参数保持不变。基本超参数取

```
parameter.m = 100; %二进制串位数
parameter.k = 1.5; %控制适配值差异的常数
num_part = 100; %种群数目
itermax = 2000; %最大迭代数
crossover_probability = 0.5; %交叉概率
mutation_probability = 0.001; %变异概率
```

此时的最优适配值曲线和平均适配值曲线如下。

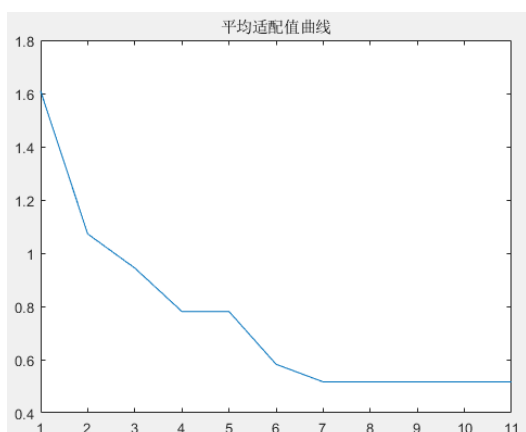


平均适配值：5.4835e-04

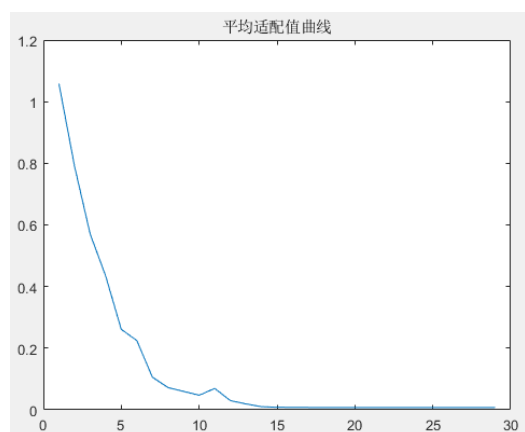
全局最优适配值：5.4835e-04

全局最优位置向量为[0.0054 -0.0020]

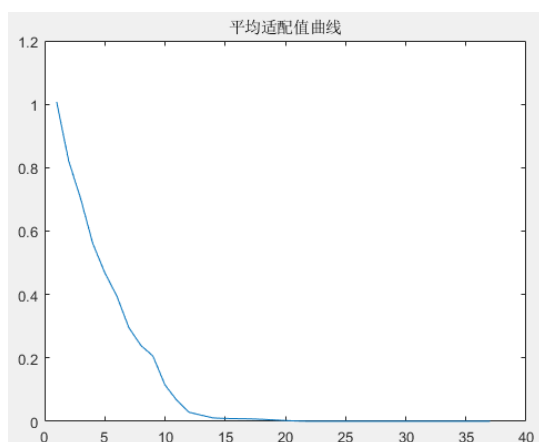
1. 种群数量 num_part 的影响



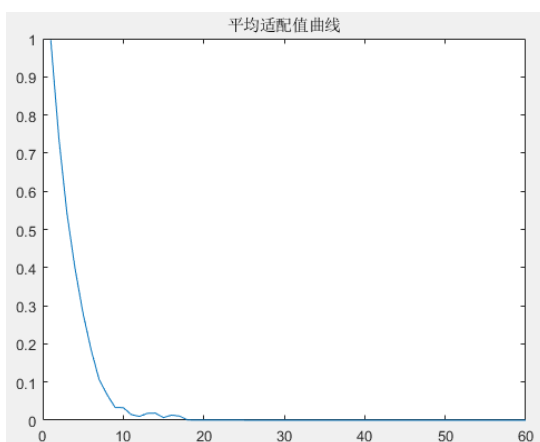
num_part = 10



num_part = 50



num_part = 100



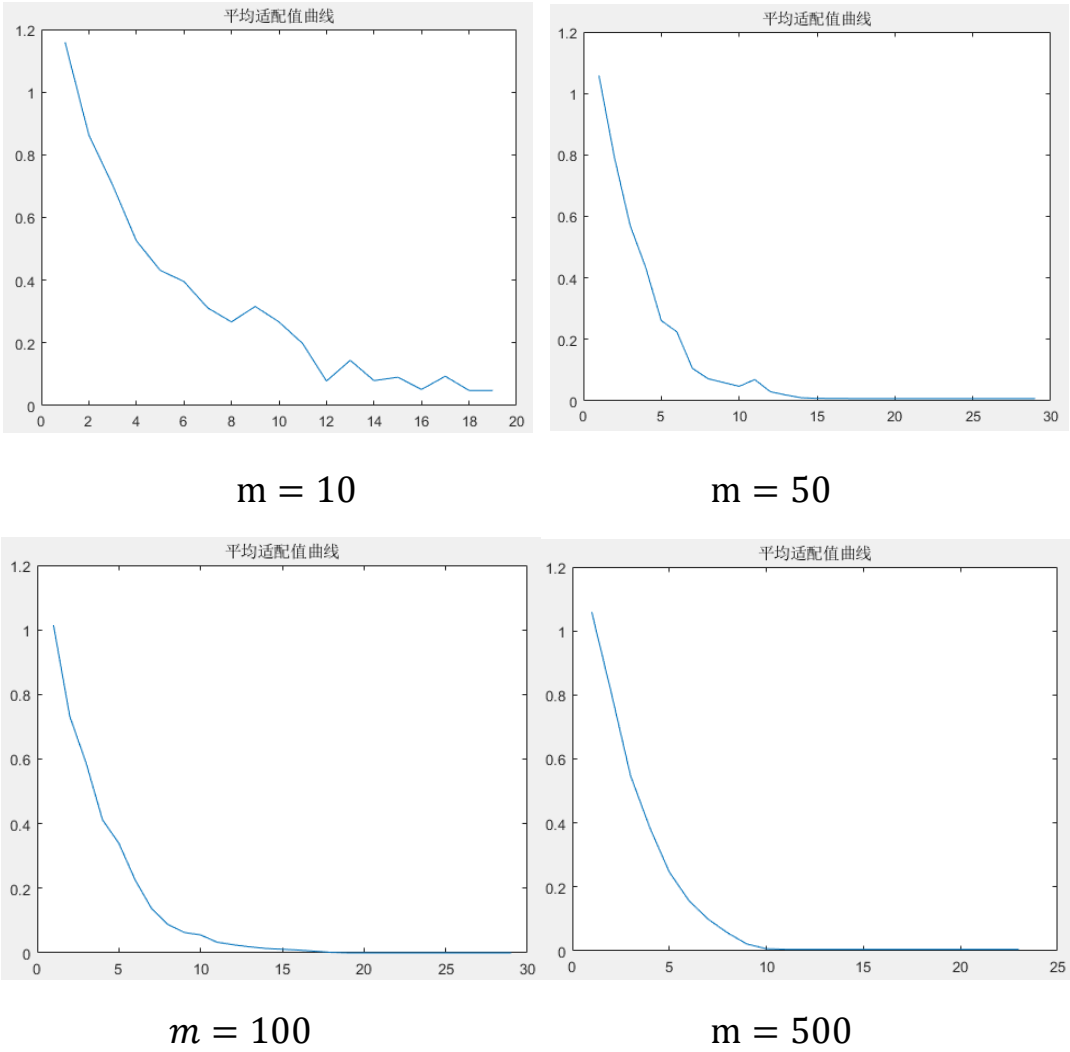
num_part = 500

num_part	iteration	meancost	globalcost
10	7	0.0939	0.0939
50	17	0.0072	0.0072
100	37	5.4835e-04	5.4835e-04
500	49	4.8470e-08	4.8470e-08

小结：当种群数量很小时由于没有比较优秀的父辈群体，很难收敛到最优解。随着群体数目的增加，迭代次数增大。每次迭代中子循

环的最大迭代值增加，总时间成本也在剧烈增加，但最优值也越来越接近理论值。可见群体数目会增加计算时间和计算精度。

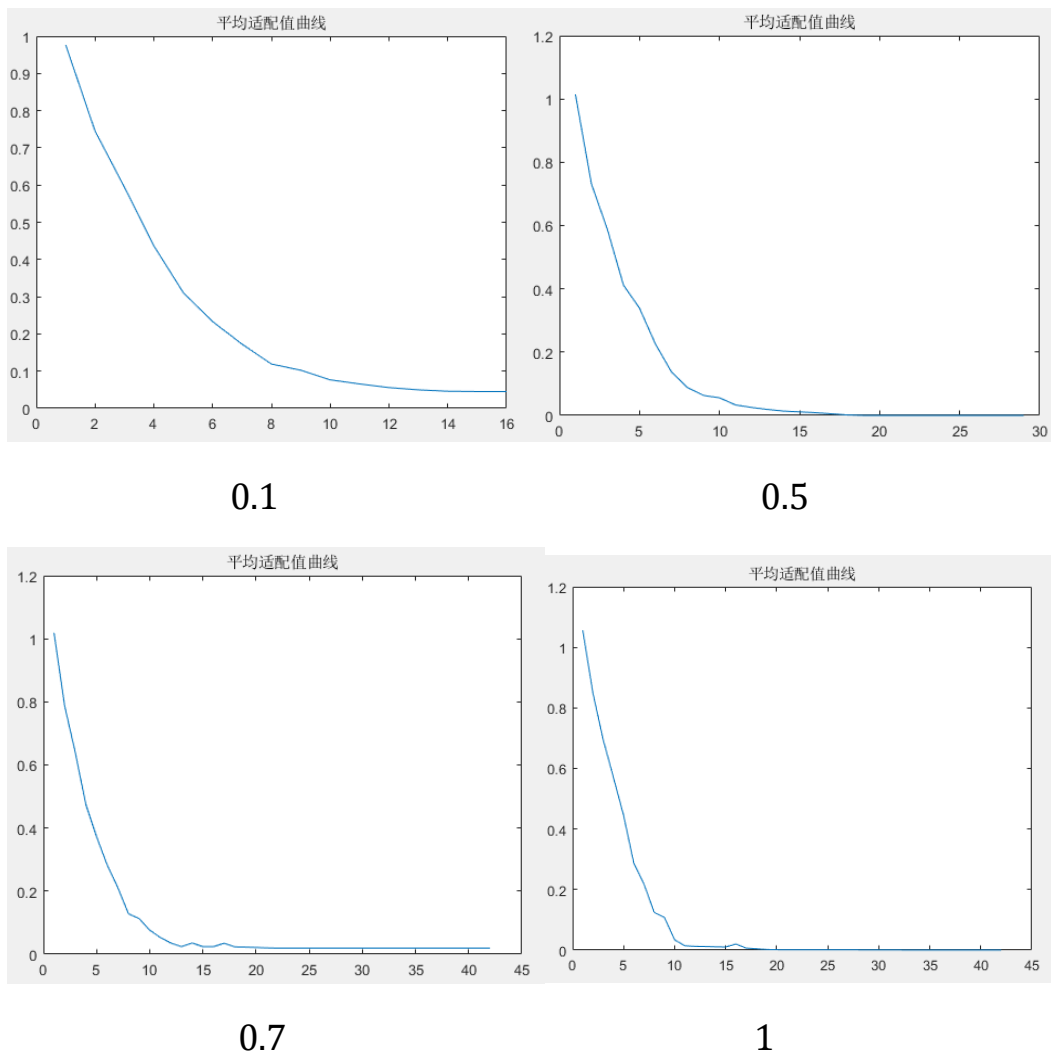
2. 二进制串位数 m 的影响



m	iteration	meancost	globalcost
m = 10	34	0.0029	0.0029
m = 50	30	5.5820e-04	5.5820e-04
m = 100	29	1.2520e-04	1.2520e-04
m = 500	23	0.0050	0.0050

小结：当 m 比较小时，优化曲线震荡下降。达不到最优解。随着 m 的增加，迭代数也有所下降，同时最优值也越来越接近理论值。但是太大的二进制串位数反而效果不太好。可见 m 的长短虽然等效于一般优化算法中的步长， m 大，则步长小。但是遗传算法并不是像一般优化算法更新自变量。所以“步长”也不会像一般优化算法中步长小则迭代步数增加，精度更好。

3. 交叉概率的影响



<i>crossover_probability</i>	iteration	meancost	globalcost
0.1	16	0.0453	0.0453
0.5	29	1.2520e-04	1.2520e-04
0.7	38	0.0191	0.0191
1	51	0.0013	0.0013

小结：可以看到交叉概率很小时，迭代步数少，收敛快，但是最优解精度不高。当交叉概率为 0.5 时，收敛速度和最优解相对折中。可以视为理想参数。随着交叉概率继续增加，收敛速度减慢，最优解精度也下降。

二、PSO 算法求解

➤ 理论部分

程序采用具有收缩系数的 PSO 算法。基本算法可以总结为：

第 i 个粒子的位置更新公式：取 $\Delta t = 1$

$$\mathbf{x}^i(k+1) = \mathbf{x}^i(k) + \mathbf{v}^i(k+1)$$

此处其中 $\mathbf{v}^i(k+1)$ 计算采用具有收缩系数的 PSO 算法：

$$\begin{aligned} v_j^i(k+1) = & K \left\{ v_j^i(k) + \varphi_1 \text{rand}(0, a_1) [p_j^i(k) - x_j^i(k)] \right. \\ & \left. + \varphi_2 \text{rand}(0, a_2) [p_j^g(k) - x_j^i(k)] \right\} \end{aligned}$$

其中： $\varphi = \varphi_1 + \varphi_2$

$$K = \begin{cases} 1 & \varphi \leq 4 \\ \frac{2}{\left| \varphi - 2 + \sqrt{\varphi^2 - 4\varphi} \right|} & \varphi > 4 \end{cases}$$

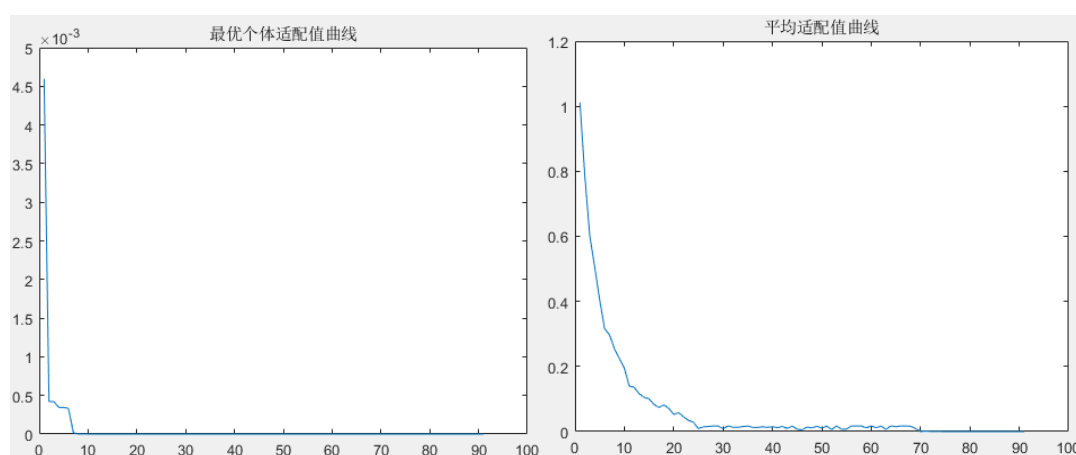
$\mathbf{p}^i(k)$ 是第 i 个粒子的历史最优位置， $\mathbf{p}^g(k)$ 是粒子群的迄今最优

位置。角标 j 表示 n 维向量的第 j 个分量。 φ_1, φ_2 分别是个体认知分量 $[p_j^i(k) - x_j^i(k)]$ 和整个粒子群认知分量 $[p_j^g(k) - x_j^i(k)]$ 的学习率。 $rand(0, a_1)$ $rand(0, a_2)$ 的引入可以增加搜索方向的随机性和多样性。使得 PSO 算法具有更好的全局搜索能力。

迭代搜索的终止条件为平均适配值变化 $< 1e-10$ 或者最大迭代次数超过 2000.

➤ 超参数讨论

下面进行超参数讨论，在讨论单个参数时，其他参数保持不变。基本超参数取 $\varphi_1 = 2.5$; $\varphi_2 = 2.5$; $a_1 = 1$; $a_2 = 1$; $num_part = 50$ 。此时的最优适配值曲线和平均适配值曲线如下。

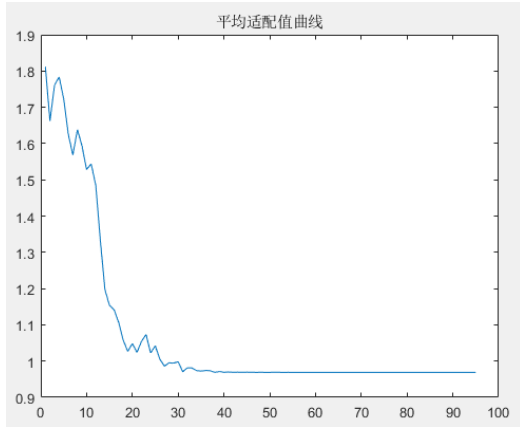


平均适配值: $1.4289e-10$

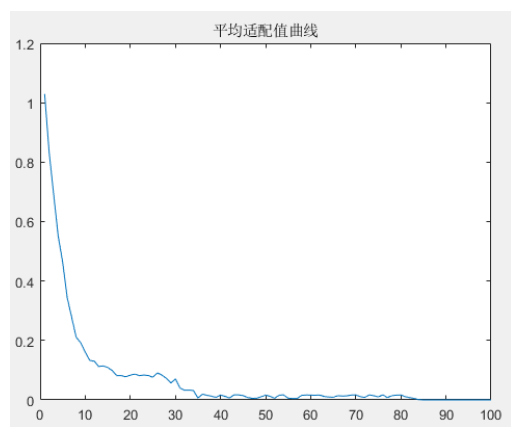
全局最优适配值: 0

全局最优位置向量为 $[-2.407e-09 \quad -1.684e-10]$

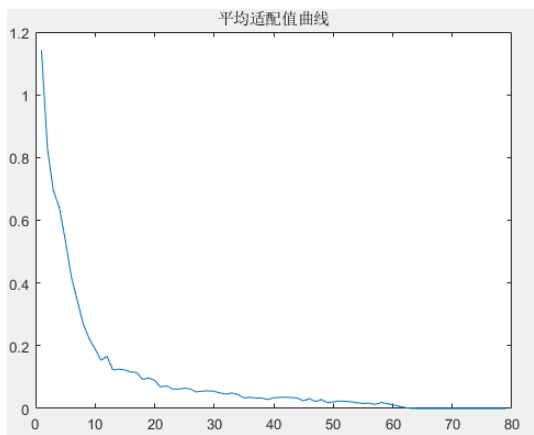
1. 粒子群数量 num_part 的影响



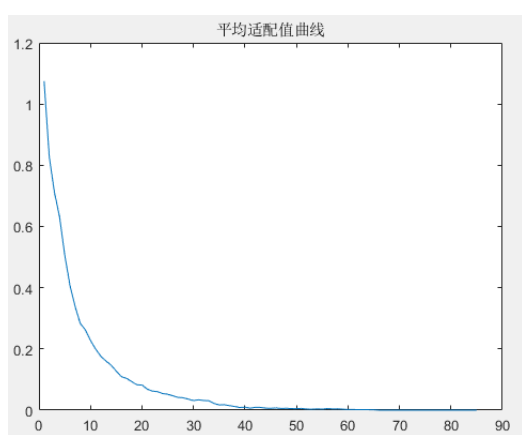
num_part = 5



num_part = 50



num_part = 100



num_part = 500

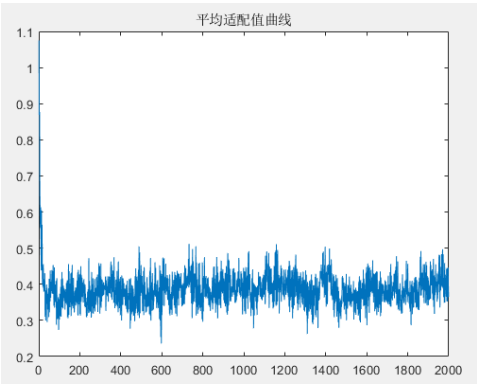
num_part	iteration	meancost	globalcost
5	88	0.0033	0.0033
50	94	4.4822e-12	0
100	79	7.4716e-11	0
500	85	4.1609e-11	0

小结：当粒子群数量很小时平均适配值曲线抖动很大，最终没有收敛到理论最优值。随着群体数目的增加，迭代次数变化不大。但是

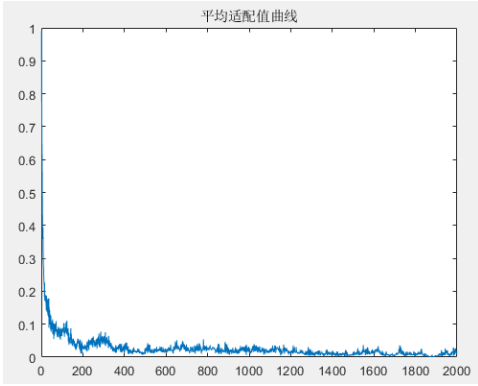
每次迭代的时间成本在剧烈增加，而最优值也越来越接近理论值。

可见群体数目与计算时间和计算精度相关。需要折中选取。

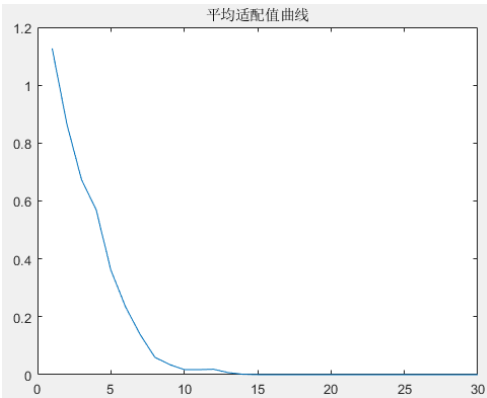
2. 学习率 φ_1 φ_2 影响



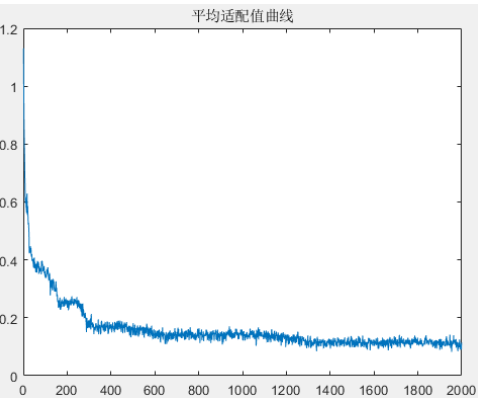
$$\varphi_1 = 1; \varphi_2 = 1$$



$$\varphi_1 = 2.01; \varphi_2 = 2.01$$



$$\varphi_1 = 2.1; \varphi_2 = 5$$

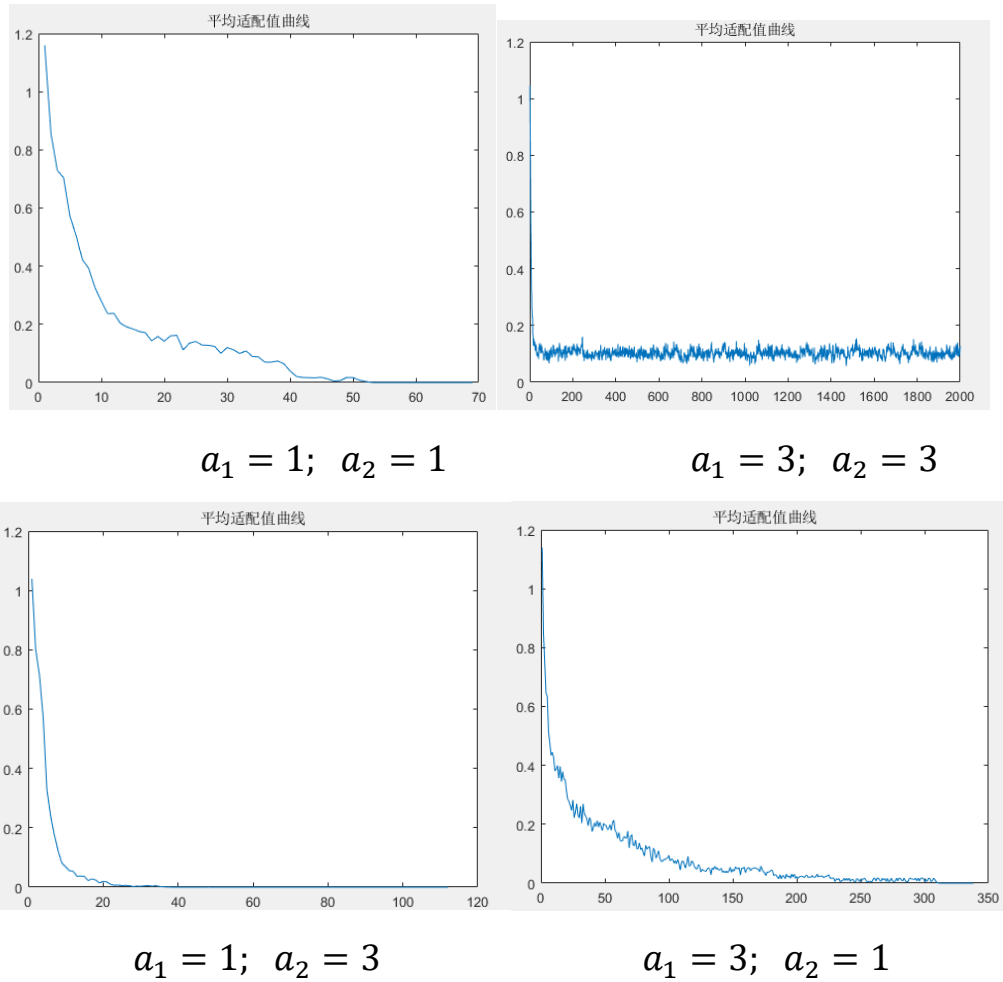


$$\varphi_1 = 5; \varphi_2 = 2.1$$

φ_1 φ_2	iteration	meancost	globalcost
$\varphi_1 = 1; \varphi_2 = 1$	2000	0.4110	1.3014e-05
$\varphi_1 = 2.01; \varphi_2 = 2.01$	2000	0.0049	0
$\varphi_1 = 2.1; \varphi_2 = 5$	30	4.5381e-11	2.2204e-16
$\varphi_1 = 5; \varphi_2 = 2.1$	2000	0.1234	0
$\varphi_1 = 2.5; \varphi_2 = 2.5$	84	1.0075e-09	0

小结：当 $\varphi = \varphi_1 + \varphi_2 < 4$ 时，迭代曲线严重震荡。最终达到最大迭代值，最优解也不理想。当 $\varphi = \varphi_1 + \varphi_2 > 4$ 但 φ 很接近4时，迭代曲线震荡缓解，但依然难以收敛。当 $\varphi_1 = 2.5$; $\varphi_2 = 2.5$ 时，收敛效果理想。当 $\varphi_1 > \varphi_2$ 时，个体认知占主导收敛曲线很难收敛。当 $\varphi_1 < \varphi_2$ 时，群体认知占主导，收敛曲线很快收敛。但并不是群体认知占主导就是最理想的超参数。因为这样减少了个体搜索方向的随机性和多样性，使得算法容易陷入局部极值。

3. a_1 ; a_2 影响



$a_1; a_2$	iteration	meancost	globalcost
$a_1 = 1; a_2 = 1$	69	8.3734e-13	0

$a_1 = 3; a_2 = 3$	2000	0.1103	1.0058e-10
$a_1 = 1; a_2 = 3$	88	3.9444e-10	0
$a_1 = 3; a_2 = 1$	352	1.1306e-10	0

小结：从速度更新公式的可以看出 a_1 a_2 的作用与学习率 φ_1 φ_2 相似，所以 $a_1 > a_2$ 和 $a_1 < a_2$ 时，收敛情况与 $\varphi_1 > \varphi_2$ 和 $\varphi_1 < \varphi_2$ 时相似。但是 a_1 a_2 控制随机量，适当的值可以增加搜索方向的多样性，而过大或者过小的值都会引起收敛曲线的剧烈震荡，甚至不收敛。

总结：两个算法相比。由于遗传算法单个样本要和别的样本进行交叉互换部分串位，所以单个遗传算法的演化过程比较离散。而且优化精度依赖种群规模，如果种群规模不大时，又较为依赖初始值，若有部分初始样本就在最优解附近，往往最终也能收敛到最优解附近。而 PSO 算法演化是基于惯性项、个体认知项和群体认知项的叠加。演化过程比较连续。从第二部分 PSO 算法中粒子群数量 num_part 的影响部分，可知达到一定规模后，pso 对种群规模不再敏感。针对此次研究的非线性函数的求解来说，PSO 算法更适合，求解精度更好。