

一. 遗传算法框架

算法框架由四部分构成，第一部分是个体类 `GA/Individual.java`，每个个体有一段基因，用 `int` 的数组表示，本 Lab 中即为 0,1 构成的数组。有一个适应度，不同问题的适应度函数在不同问题中分别实现，这里只调用各问题提供的 `calFitness` 方法。

种群类 `GA/Population.java`，该部分包括了种群大小 `individualNum`，种群的个体数组 `individuals`，种群的交叉概率和变异概率。

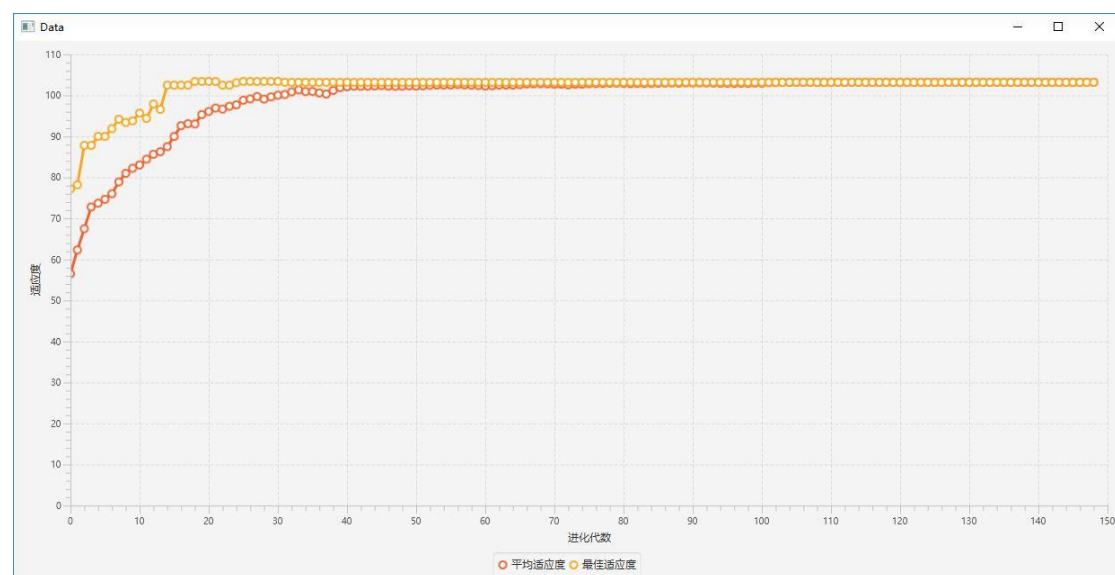
遗传算法核心类 `GA/GA.java`，这里有进化、交叉、变异和自然选择的方法。默认的进化算法采用均匀交叉，变异算法采用单点变异，自然选择方法采用轮盘赌选择。在每个问题中，有单独的类继承 `GA` 类，实现不同的算法。

适应度计算类 `GA/Fitness.java`，每个问题的函数单独在问题类中实现。

二. 背包问题

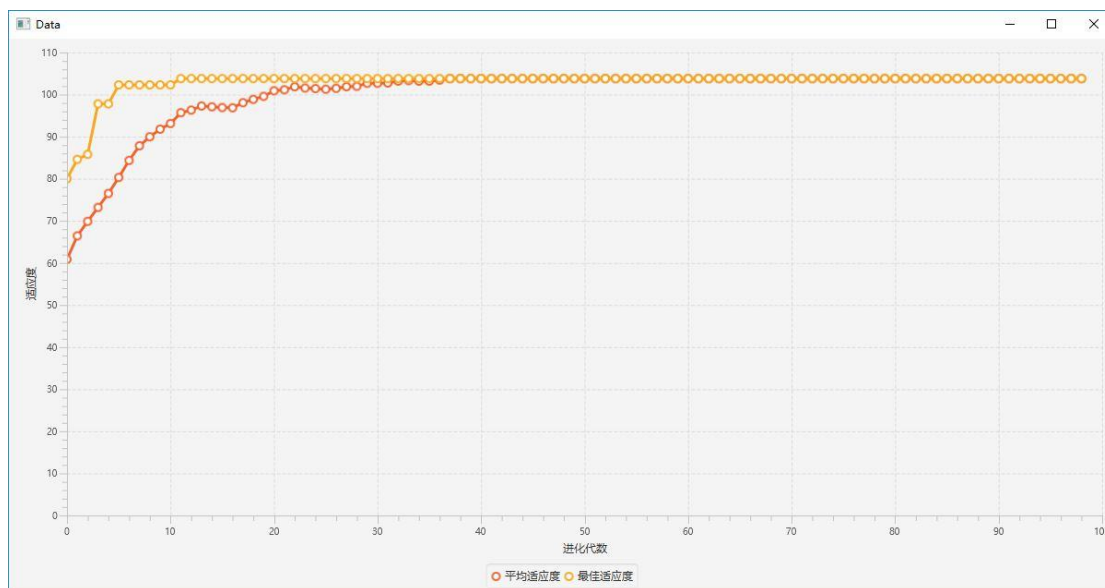
背包问题在遗传算法框架上的拓展主要是，适应度的计算，这里需要累加放入物品的价值。另外在进行种群初始化、交换、变异的过程中，需要计算总的重量，如果超过的背包的容量，不能进行该部分的操作。

运用事例训练集进行实验。选择种群的个体数量为 50，交换概率为 0.8，变异概率为 0.005。其中交叉算法为均匀交叉，生成与基因串长度相同的屏蔽串，每一位为任意 0 或 1，当为 1 时表示该位置进行交叉。突变算法为单点交叉，选择基因串中任意一位进行突变。设置停止条件为第 200 代时。实验结果为在第 105 代后收敛，平均适应度为 103.2，最大适应度为 103.2，但是在 20 代左右有出现 103.4 的最大适应度。并且在进化过程前 40 代中，最大适应度的变化很大，考虑到均匀交叉的随机性较大，有可能将种群中优秀的基因破坏掉了。多次实验发现进化过程中平均适应度和最高适应度都不稳定。并且收敛速度很慢，基本是在 100 代以后才收敛。



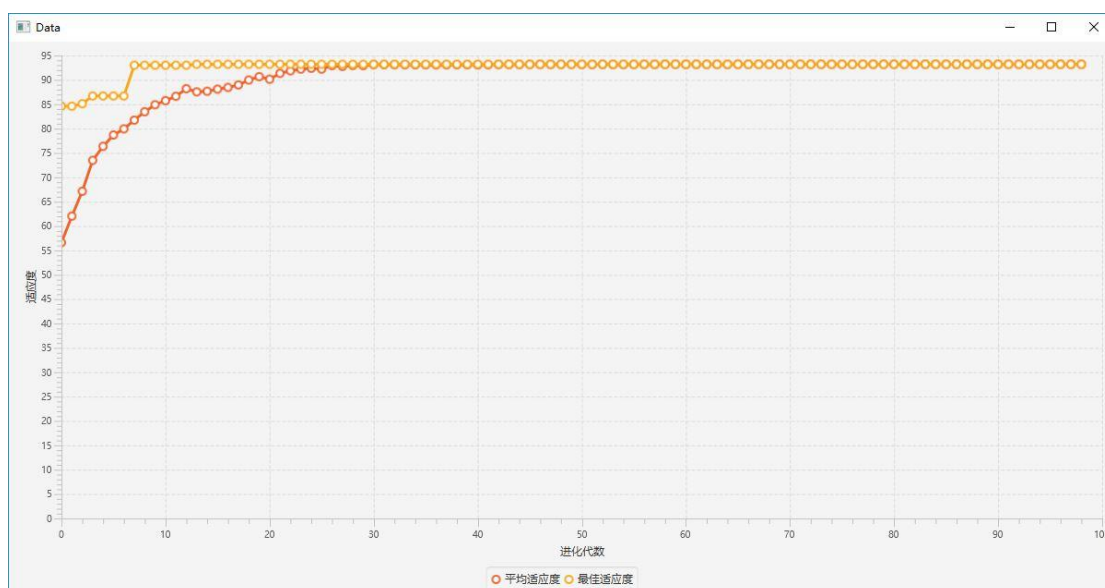
初始

于是采用了“精英策略”，在繁衍下一代的过程中，将上一代最优秀的个体完全保留下来，该个体不参与到后序的交叉和变异中以保留优秀的基因，防止被破坏。承接上一次的实验参数，实验效果较好，没有出现最优适应度波动的情况，收敛速度也迅速加快，在第 38 代收敛，平均适应度为 103.80，最高适应度为 103.80。可见“精英策略”在加快收敛速度和保留优秀基因上有很大作用。



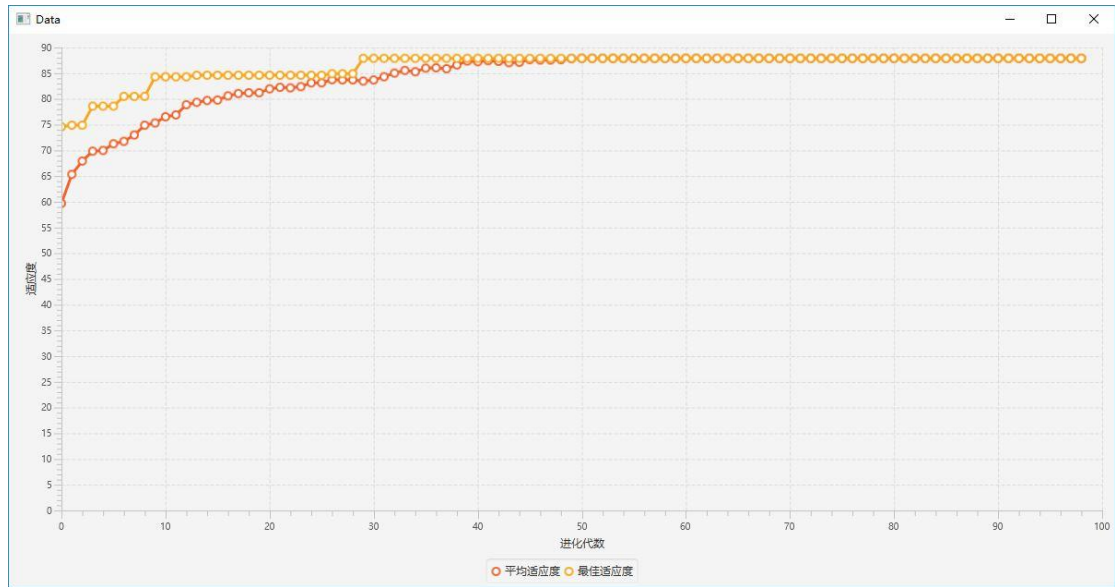
“精英策略”——均匀交叉

更改交叉算法为**单点交叉**，即选择基因串长度的任意一点，将该点以后的基因交换，而之前的保持不变。多次实验后，发现最高适应度在 95 以下，平均在 70 代后收敛。下面图中实验收敛后最优适应度和平均适应度为 93.20。我认为在背包问题中，使用单点交叉不容易找到优秀解，一方面是基因重组的能力较差，另一方面是交换操作中，交叉点后的组合在满足容量前提下，受限于交叉点前的基因形式，难以跳出前序基因的限制。



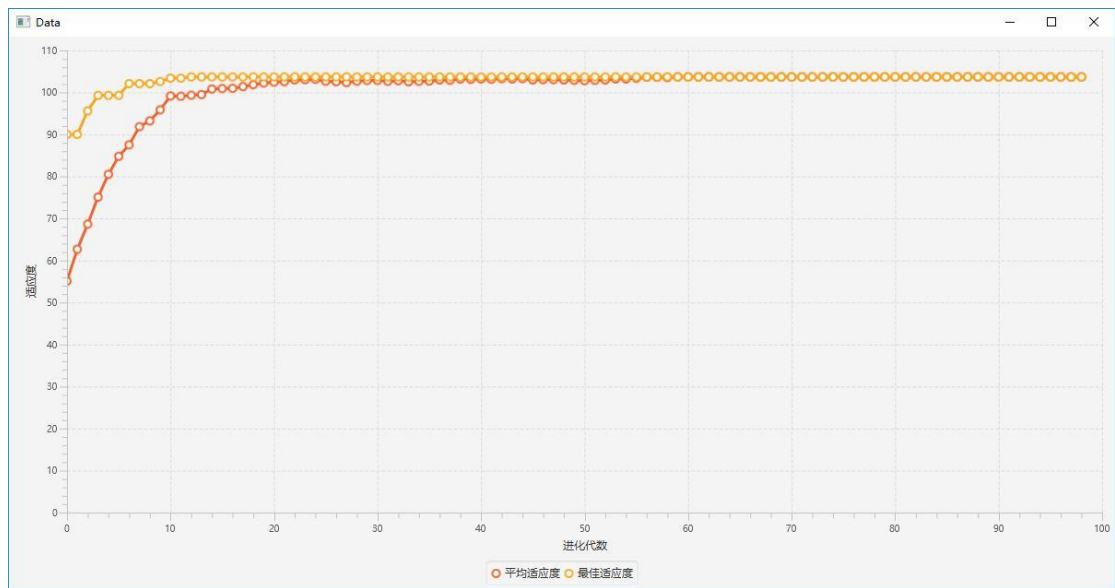
单点交叉

更改交叉算法为**两点交叉**，即选择基因串中任意两点，将两点间的基因进行交换，而两旁的保持不变。在其他参数不变的情况下，多次实验，最高适应度均低于 90，平均在 55 代以后收敛。下图实验中，在第 51 代收敛，最大适应度和平均适应度为 87.9。两点交叉与单点交叉有相同的问题，两点交叉由于进行交换的基因数量更少，所以跳出两端的基因模式进行重组的能力更差，更容易陷入局部最优解。



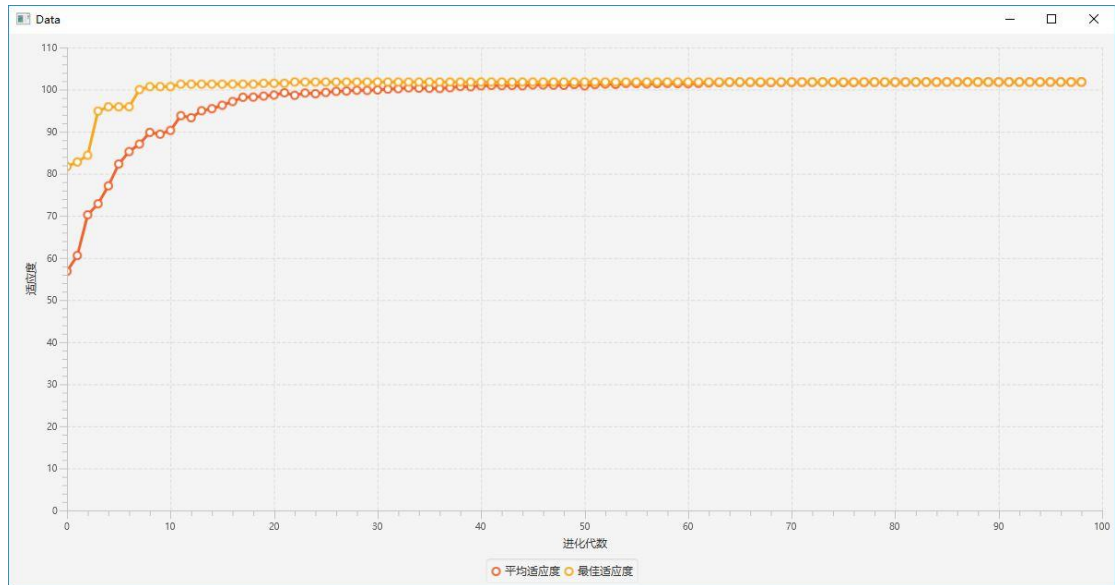
两点交叉

综上，不再对交叉算法进行调整，采用均匀交叉。更改变异算法为**两点变异**，即随机选择两个基因位置进行变异。实验结果与单点变异相差不大。可能因为变异点增加导致适应度改变概率较大，收敛的速度变慢了。重复实验，收敛后的适应度稳定在 100—104 之间，结果比单点变异要好。



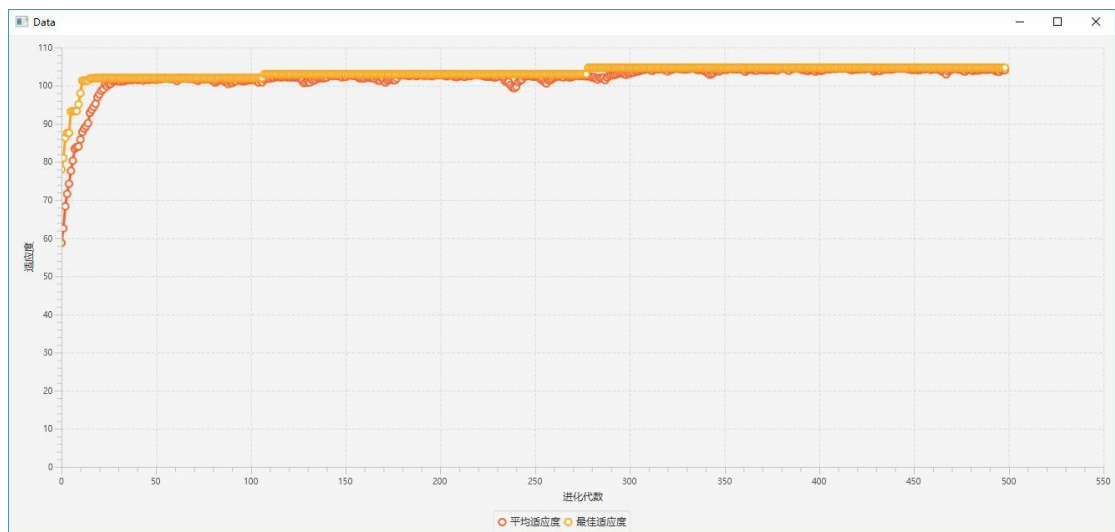
两点变异

改变异算法为**根据适应度启发式算法**，即如果变异的个体比种群平均适应度高，则进行单点交叉或两点变异，否则实现均匀变异。多次实验后结果与单纯两点交叉比，没有明显的优化。反而有时收敛后的适应度下降到了 100 以下。故仍然采用单点交叉。



根据适应度启发式突变

增加交叉率为 0.9，重复实验十次，收敛后的适应度保持在 103—106 之间，已经有优化的比较好，故不再调整交叉率。增加变异率到 0.05，200 代内无法收敛，增加进化代数到 500，一次实验中 500 代的结果如下图所示，可见因为较大的变异率，平均适应度始终无法收敛，最佳适应度在 500 代过程中有三次明显的提升，说明增大变异率可以增加找到最优个体的概率。但是由于进化代数过多，最后调整为进化 300 代，保持 0.05 的变异率。



500 代

增加种群数目到 100 个，运行了 5 次，每次实验后的最佳适应度都是 106.3。所以不再修改超参数。

最终的超参数为：

种群大小：100；进化代数：300；交叉概率：0.9；突变概率 0.05

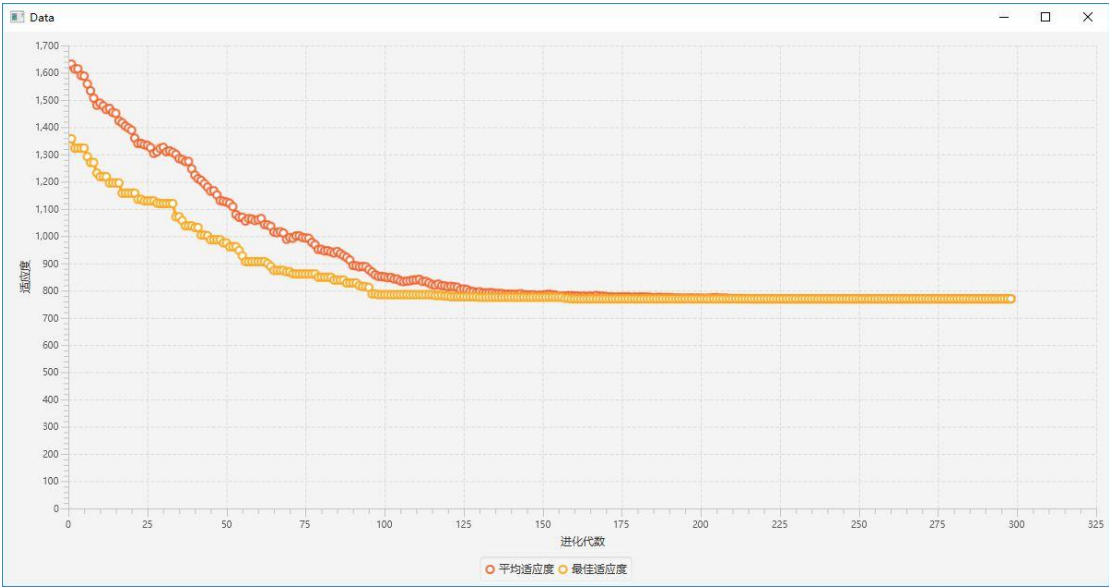
自然选择算法：轮盘赌；交叉算法：均匀交叉；突变算法：单点变异

三. TSP

为了和背包问题用同一个框架，这里将适应度函数设为距离和的倒数。

设置背包问题的超参数为初始的超参数。交叉算法为**顺序交叉法**，即选择随机一个基因片段，子代 a 和 b 分别复制父代 A 和 B 的该片段基因。然后对于子代 a 中剩余的位置，依

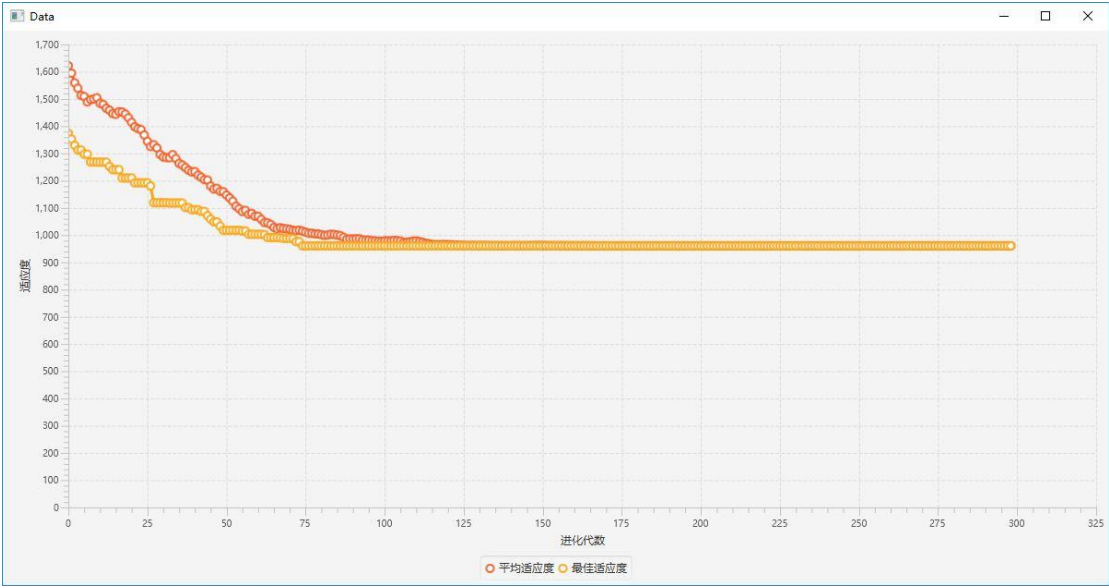
次从父代 B 中顺序选取（若 a 中已有该基因则跳过避免重复）。b 同理。变异算法为**单点随机位置变异**，即选择基因段中任意一位，将其与基因段中任意另外一位交换。实验结果如下图，在 225 代后收敛，最佳适应度和平均适应度为 769.79，



顺序交叉法& 单点随机位置变异

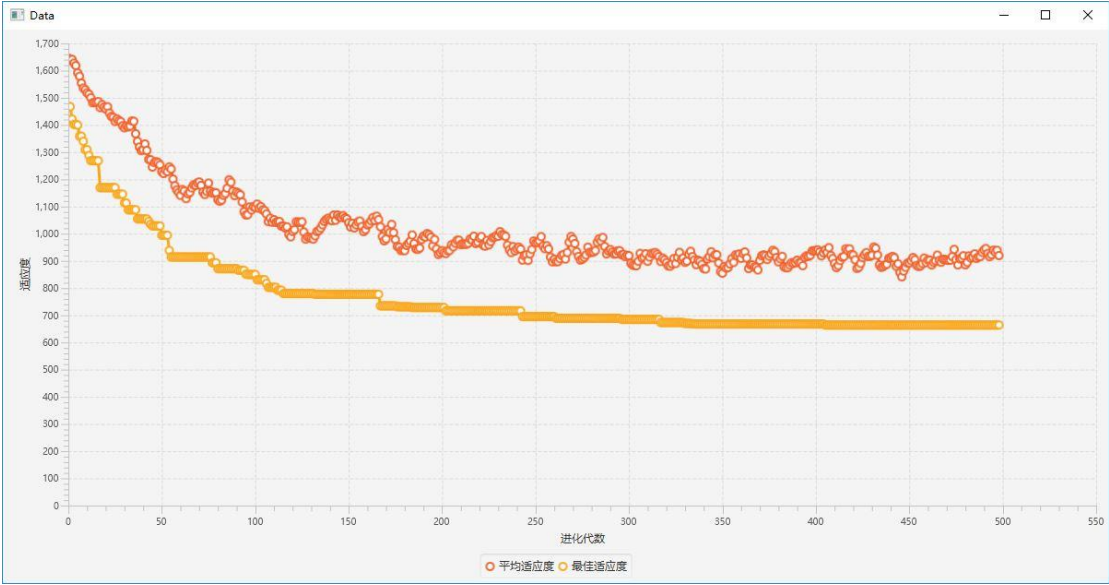
将突变算法改为**单点相邻变异**，即随机选取一位与旁边的进行交换。多次实验后发现最佳适应度始终高于 800。所以仍然改回单点随机位置变异。

更改交换算法为**部分匹配交叉**，即随机选取基因段中两个位置，父代 A, B 两个位置中的基因分别替换，其他位置保持不变。对其他位置产生冲突的基因，按照更换区域的对应顺序进行替换。保持其他参数不变。实验多次发现效果不佳。下图是选取的最优的一次结果。在第 170 代收敛，适应度为 960.64。其他实验结果的适应度均高于 1000。于是将交换算法换回到顺序交叉。



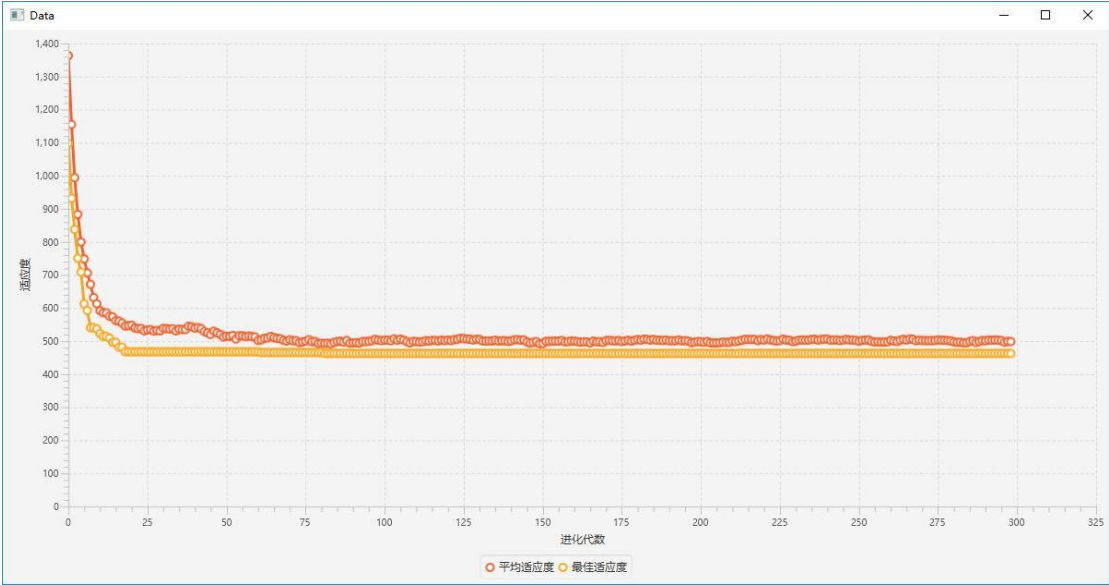
部分匹配交叉

修改交换算法为顺序交换，但是不同于之前，现在补全交换区域以外的位，是从交换位置开始向后找，找到基因串末尾后再从头开始。进化 500 代，结果显著好于前两种算法。最后的最优适应度可以达到了 664.12。如果持续进化，尝试过在 10000 代后可以降到 480，但是显然该算法存在问题。由于该算法的交换实在太大，所以平均适应度的波动很大，也不能收敛到与最优适应度相同。



顺序交换（后序）

修改交叉算法为**两交换启发交叉(HGA)**，该算法原理是每次选择两个个体，如
A=4 1 3 6 7 8 2 5 B=1 3 4 2 6 5 8 7
现在任意选取一位，比如我选择 2 作为开始城市，然后修改 A，B 的基因序列为
A=2 5 4 1 3 6 7 8 B=2 6 5 8 7 1 3 4
下一步是判断(2,5)与(2,6)间的距离谁更近，如果(2,5)更近，则将 2 添加到输出，A 和 B 变为
A=5 4 1 3 6 7 8 B=5 8 7 1 3 4 6
依次对整个基因序列进行操作。该算法的效果很好，在其他参数不变的情况下，依次实验结果如下。可见收敛很快，300 代后最佳适应度为 462.72，平均适应度为 493.68。

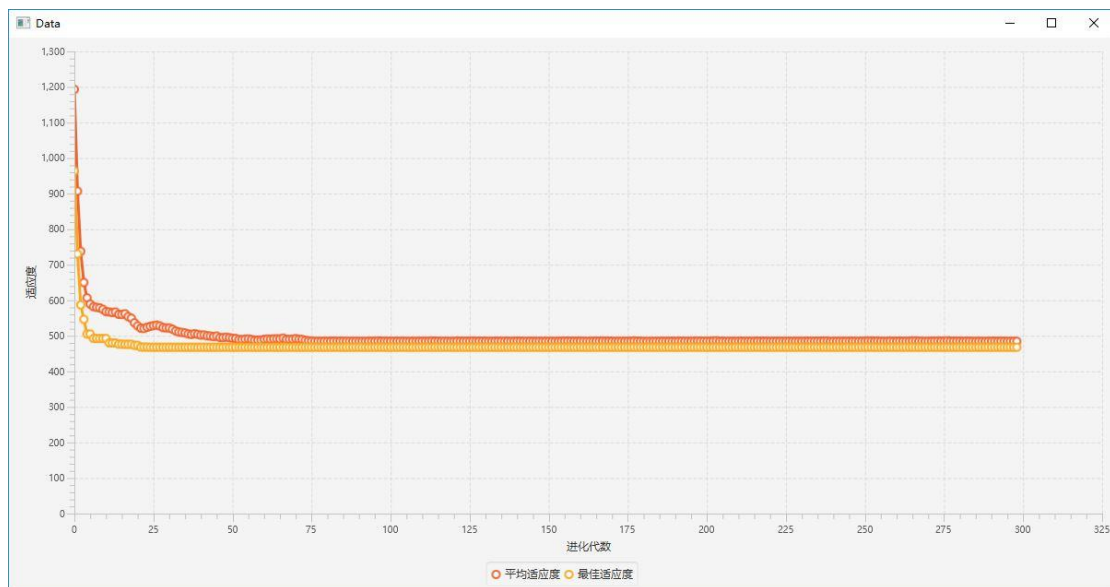


两交换启发交叉(HGA)

重复多次实验，发现最优值在 430—480 间波动。在每一次交换算法执行完成前，做了一次优化。该优化是将输入的两个个体的基因序列进行了“旋转”，也就是本来是“12345”，旋转后为“23451”，这样总的路程不变，路径不变，但是起点变了。这样的改变有利于找到交叉后的不同解，有利于找到更优解。

调整种群大小为 200 个，实验结果没有较大变化。继续增加到 300 个，仍然没有较大变化。增加变异率为 0.1 后多次实验仍然没有较大变化。修改变异算法为**翻转交叉**，即任选一个基因段，将其中心旋转。如原来是“1 2 3 4 5”，可能变为“1 4 3 2 5”。修改后算法仍然在原区间波动。

继续调整交换算法为**三交换启发式算法(THGA)**，原理与之前两交换相似，算法收敛速度进一步加快，但是最优值仍然不稳定。一次实验结果如下，300 代后最优适应度为 467.85，平均适应度为 484.75。最优值在 25 代后收敛。



三交换启发式算法(THGA)

由于平时适应度仍然较高，我修改了变异算法，使其进行多次翻转，避免陷入局部最小值，也能发现更优秀的基因。使翻转次数从 3 增加到 20，发现增加翻转确实能增大发现更优基因的概率。最后选择了翻转 10 次。并增加近代代数到 500，一次实验后结果如下。最终最优适应度为 449.96，平均适应度为 540.49。



最终的超参数为：

种群大小：300；进化代数：400；交叉概率：0.9；突变概率 0.1

自然选择算法：轮盘赌；交叉算法：三交换启发式算法；突变算法：多翻转