

汕头大学工学院

三级项目报告

学年学期：2020-2021 学年秋季学期

课程名称：人工智能优化理论及应用 (EEG3236A)

题目名称：基于遗传算法的回流焊接问题的求解研究

仿真软件：Matlab

团队成员(5~10 人)

姓名	学号	分工
姜志成（组长）	2018633007	查找资料，编写代码，调试代码
张桦灿（组员）	2016141074	调试代码，制作 PPT
梁绍文（组员）	2018633010	查找资料，编写代码
韵皓宇（组员）	2018631060	撰写报告，调试代码

完成时间：2020 年 11 月 1 日至 11 月 30 日

评阅人：

目录

- 一. 研究背景..... 3
 - 1.1 求解问题背景..... 3
 - 1.2 求解问题介绍..... 4
- 二. 算法原理介绍..... 5
 - 2.1 简介..... 5
 - 2.2 术语..... 5
 - 2.3 原理及流程..... 5
- 三. 炉温曲线问题的求解应用仿真分析..... 8
 - 3.1 求解问题及优化目标: 8
 - 3.2 优化变量: 8
 - 3.3 变量上下限约束:..... 8
 - 3.4 制程约束（机器、工艺上的约束） 9
 - 3.5 仿真过程 9
 - 3.5.1 初始化遗传算子等参数 9
 - 3.5.2 产生初始种群..... 9
 - 3.5.3 判断是否满足终止条件..... 10
 - 3.5.4 仿真结果..... 10
 - 3.5.5 算法优化: 多种群遗传算法 (Multi-populationGenetic Algorithm, MGA) 11
- 四. 总结..... 13

一．研究背景

1.1 求解问题背景

回流焊接技术是指借由回焊炉，通过表面黏着技术使得电子元器件管脚与 PCB 板间的焊料经过高温融化后再冷却固定，从而完成集成电路焊接的工艺。传统的回流焊接分为预热、恒温、回流、冷却 4 个阶段，分别对应由若干小温区组合而成的如图所示 4 大功能温区。由于该过程的关键在于对内部温室的温度和传送带速度的控制，合理设定各温区温度和传送带的过炉速度可以得到更高质量的焊接（电路板两侧搭在传送带上匀速进入炉内进行加热焊接）。

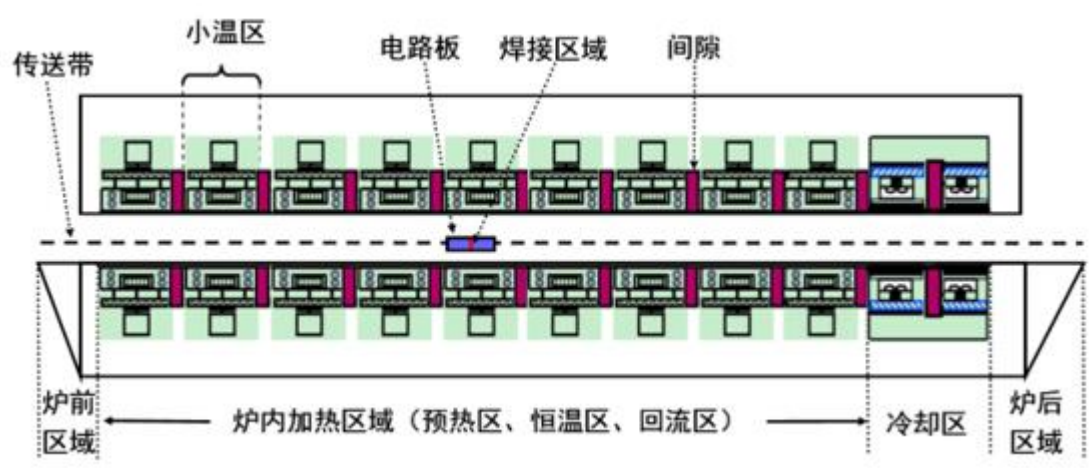


图 1 回焊炉内部构造图

某回焊炉内有 11 个小温区及炉前区域和炉后区域（如图 1），每个小温区长度为 30.5 cm，相邻小温区之间有 5 cm 的间隙，炉前区域和炉后区域长度均为 25 cm。

在设定各温区的温度和传送带的过炉速度后，可以通过温度传感器测试某些位置上焊接区域中心的温度，称之为炉温曲线(即焊接区域中心温度曲线)。各温区设定的温度分别为 175°C(小温区 1~5)、195°C（小温区 6）、235°C（小温区 7）、255°C（小温区 8~9）及 25°C（小温区 10~11）；传送带的过炉速度为 70 cm/min；焊接区域的厚度为 0.15 mm。温度传感器在焊接区域中心的温度达到 30°C 时开始工作，电路板进入回焊炉开始计时。

在实际生产时，可以通过调节各温区的设定温度和传送带的过炉速度来控制产品质量。在上述实验设定的基础上在上述实验设定温度的基础上，各小温区设定温度可以进行±10°C 范围内的调整。调整时要求小温区 1~5 中的温度保持一致，小温区 8~9 中的温度保持一致，小温区 10~11 中的温度保持 25°C。传送带的过炉速度调节范围为 65~100 cm/min。则变量约束的上下限如表 1 所示：

变量名称	数值	单位
小温区 1-5	165-185	℃
小温区 6	185-205	℃
小温区 7	225-245	℃

小温区 8-9	245-265	℃
小温区 10	25	℃
过炉速度	65-100	cm/min

表 1 变量上下限约束表

为了更好地求解和优化回流焊接过程，炉温曲线的各参数应满足表 2 所示的制程界限：

界限名称	最低值	最高值	单位
温度上升斜率	0	3	℃/s
温度下降斜率	-3	0	℃/s
温度上升过程中在 150℃-190℃的时间	60	120	s
温度大于 217℃的时间	40	90	s
峰温度	240	250	℃

表 2 制程界限表

1.2 求解问题介绍

在焊接过程中，焊接区域中心的温度超过 217℃ 的时间不宜过长，峰值温度也不宜过高。理想的炉温曲线应使超过 217℃ 到峰值温度所覆盖的面积（图 2 中阴影部分）最小。求解问题为确定在此要求下的最优炉温曲线，以及区的设定温度和传送带的过炉速度，并给出相应的面积。

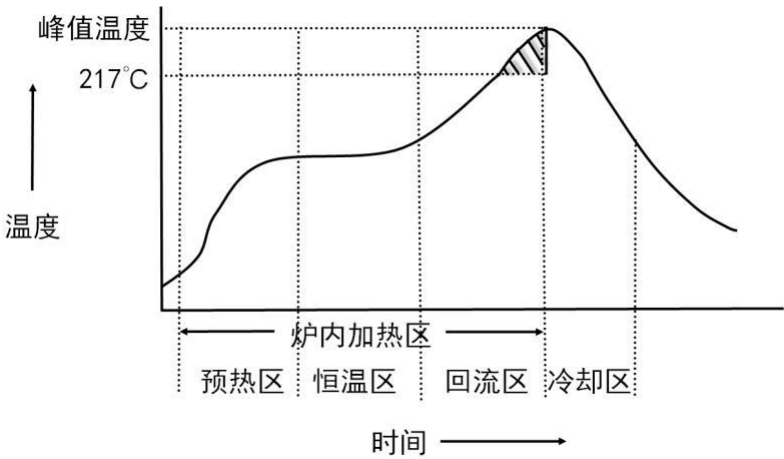


图 2 炉温曲线示意图

根据表 1 和表 2 的约束条件，若用多重循环嵌套搜索的传统算法时间复杂度过大，不宜使用。因此，鉴于遗传算法的并行性能够加快求解速度，为简化计算量文本，报告基于标准遗传算法建立单目标优化模型，用其简单的编码技术和算法机制模拟复杂的优化过程，从而搜索最优的回流焊接控制方案，并提出多种群遗传算法来优化进化策略。

二. 算法原理介绍

2.1 简介

遗传算法是一种随机的进化算法，起源于达尔文的进化论。通过多次地将高质量的个体进行繁衍，从而生成更加优质的个体。生物遗传和进化的规律有：

- (1) 生物的所有遗传信息都包含在染色体当中, 染色体决定了生物的形状
生物的繁殖过程是由其基因的复制过程来完成的. 同源的染色体的交叉或变异会使生物产生新的性状.
- (2) 对环境适应能力强的个体有更多的机会繁殖下一代, 并将优秀的基因遗传到下一代

2.2 术语

- (1) 群体：个体（多少个解）
- (2) 个体：每个个体有 N 条基因定义这个个体
- (3) 适应度：通过适应函数计算出个体的适应度，适应度越高说明个体越优。
- (4) 繁殖池：只有在这个池子的物种能够繁殖
- (5) 父母：繁殖池里面的个体，这些个体能够繁殖后代
- (6) 选择：选择繁殖池内的个体进行繁殖
- (7) 交叉：父母各随机取一定比例的基因给后代
- (8) 变异：繁殖的时候染色体上的基因的随机变化从而有一定几率能够克服父母的缺点。

2.3 原理及流程

染色体的表达形式

基因型：用基因表达的染色体：数值，位置

表达型：染色体的实际物理表达

表达方式：二进制数字，数值，排列

二进制染色体例子

0	1	1	1
---	---	---	---

上图中每一个格子都是一个基因 0111 是一个基因型的表达, 7 则是表达型的表达如果适应度函数为 $y=3x+2$ ，则这条染色体的适应度 $y=3*7+2=23$

下面这个例子展示如何怎样通过遗传算法提高种群适应度

题目：

有四个美人鱼分别有对应的基因值，繁衍两代人每个美人鱼由 4 种基因组成，分别为 g1, g2, g3, g4, 每条基因的范围是-2 到 2 之间的整数每一个美人鱼的适应度公式= $g1-2*g2+3*g3+4*g4$

适应度最高的两个美人鱼能够进行繁殖两个美人鱼繁殖两个孩子：

第一个孩子：g1, g2 来自父亲，g3, g4 来自母亲

第二个孩子：g1, g2 来自母亲，g3, g4 来自父亲

新孩子诞生后，第一条数值最高的基因会突变成-5 到 5 之间的整数
新生的孩子及其父母会留在繁衍池内，其他会被淘汰

第一代

	g1	g2	g3	g4	适应度
美人鱼 1	-2	2	1	2	5
美人鱼 2	1	-1	1	-1	2
美人鱼 3	-1	-1	0	2	9
美人鱼 4	1	1	0	2	7

选择适应度最高的两条美人鱼（3，4）进入繁衍池，成为下一代父母按顺序选择美人鱼 3 为父亲，美人鱼 4 为母亲经过基因交叉，第一个孩子继承父亲的 g1, g2, 母亲的 g3，g4；第二个孩子继承母亲的 g1, g2, 父亲的 g3, g4.

	g1	g2	g3	g4	适应度
美人鱼 3	-1	-1	0	2	9
美人鱼 4	1	1	0	2	7
小美人鱼 1	-1	-1	0	2	9
小美人鱼 2	1	1	0	2	7

根据突变规则，数值最高的基因分别突变为-5 和 4

	g1	g2	g3	g4	适应度
美人鱼 3	-1	-1	0	2	9
美人鱼 4	1	1	0	2	7
小美人鱼 1	-1	-1	0	-5	-19
小美人鱼 2	1	1	0	4	15

选择适应度最高的美人鱼 3 和小美人鱼 2 作为下一代的父母，和上面过程一样，经过基因交叉和基因突变，繁衍出第三代

	g1	g2	g3	g4	适应度
美人鱼 3	-1	-1	0	2	9
小美人鱼 2	1	1	0	4	15
小小美人鱼 1	-1	-1	0	5	21
小小美人鱼 2	1	1	0	2	7

可见经过两代繁殖，总群的最优个体的适应度由 9 上升到 21

上述过程可以用伪代码表示如下

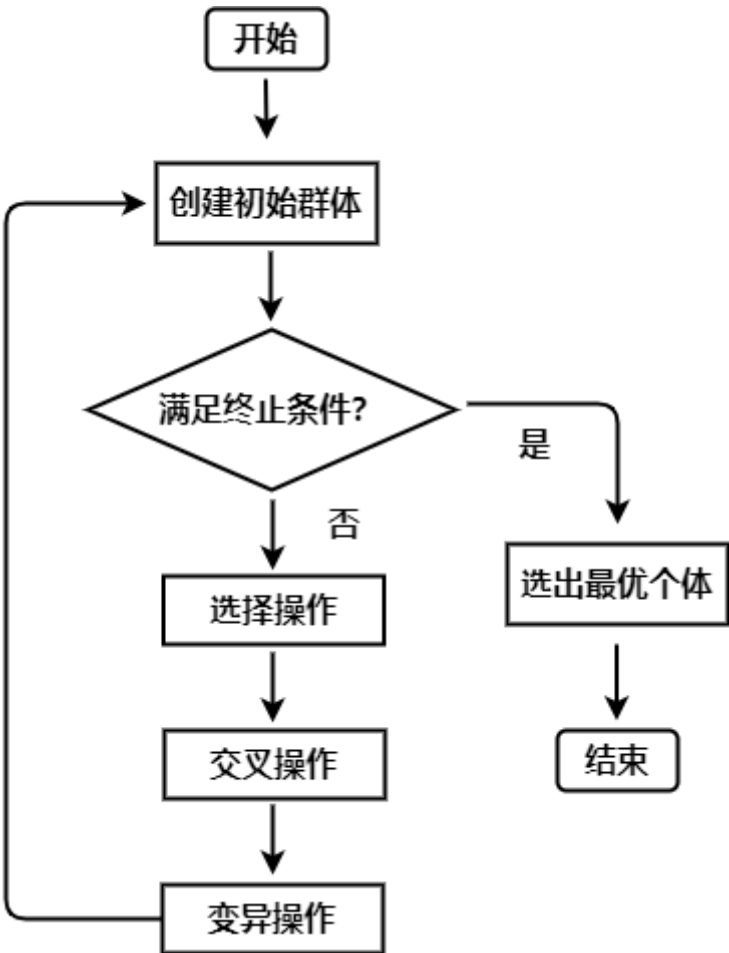


图 1 遗传算法的运算流程

三. 炉温曲线问题的求解应用仿真分析

3.1 求解问题及优化目标:

该问题的优化目标为在满足制程约束（工艺要求）的情况下，炉温曲线应使超过 217°C 到峰值温度所覆盖的面积最小

首先在遗传算法中对每个个体中传送带过炉速度和温区温度进行搜索时，根据传热方程 $T = f(t)$ 拟合出对应的炉温曲线，借由微元法的思想在单位时间内求得函数值与时间的乘积，再进行面积积分，因此在遗传算法中设定适应函数为面积公式：

$$Fitness(T_{1-5}, T_6, T_7, T_{8-9}, T_{10}, v_{machine}, t) = -\min \sum_{t_{rise} > 217} T(t) \Delta t$$

3.2 优化变量:

各温区设定的温度分别为 175°C（小温区 1~5）、195°C（小温区 6）、235°C（小温区 7）、255°C（小温区 8~9）及 25°C（小温区 10~11）传送带的过炉速度调节范围为 65-100 cm/min。；小温区 1-5 中的温度保持一致。小温区 8-9 中的温度保持一致，小温区 10-11 中的温度保持 25°C。小温区（除了 10-11 小温区）设定温度可以进行 ±10°C 范围内的调整

小温区 1-5 中的温度保持一致，变量范围是 165-185°C，初始值为 175°C

小温区 6 中的温度变量范围是 185-205°C，初始值为 195°C

小温区 7 中的温度变量范围是 225-245°C，初始值为 235°C

小温区 8-9 中的温度保持一致，变量范围是 245-265°C，初始值为 255°C

小温区 10-11 中的温度保持一致，数值是 25°C

3.3 变量上下限约束:

$$s. t. \begin{cases} 165 < T_{1-5} < 185 & (^\circ C) \\ 185 < T_6 < 205 & (^\circ C) \\ 225 < T_7 < 245 & (^\circ C) \\ 245 < T_{8-9} < 265 & (^\circ C) \\ T_{10-11} = 25 & (^\circ C) \\ 65 < v_{machine} < 100 & (cm/min) \end{cases}$$

3.4 制程约束（机器、工艺上的约束）

$$\begin{aligned} & 0 < k_{rise} < 3 \\ & -3 < k_{fall} < 0 \\ s.t. \{ & 60 < t_{T_{150-190}} < 120 \\ & 40 < t_{T_{>217}} < 90 \\ & 240 < t_{peak} < 250 \end{aligned}$$

3.5 仿真过程

3.5.1 初始化遗传算子等参数

1) 群体规模 NP

初始化种群数目 $NP = 40$

2) 选择算子

在遗传算法中选择算子用于选取群体中的优良个体，并淘汰劣质个体。基于题目要求取得最小面积的背景，本文选取排序选择法作为选择算子，根据每个个体的适应值大小进行排序后计算其生存概率，大适应值对应较高的存活率，以实现优良个体的延续。

3) 交叉算子 P_c

本文以 $P_c = 0.7$ 的交叉概率，将群体中的个体成对搭配，每队个体在交叉概率下进行随机选择染色体部分交换，即对换交叉位置的二进制串。新个体组合了上一代个体的特性，可以跳出局部最优搜索全局最优解。

4) 变异算子

模拟自然界中生物体进化时染色体发生的基因突变，使得染色体按照变异概率 $P_m = 0.01$ 进行变异，随机对某一位置产生变异基因，即将个体的某个二进制编码位进行取反操作的变异过程。变异算子使得丰富了种群的多样性，有机会产生更加优质的个体。

5) 代沟

$GGAP = 0.95$

6) 染色体二进制编码长度 PRECI

$PRECI = 7$

7) 最大进化次数

$MAXGEN = 50$

3.5.2 产生初始种群

将二进制编码转换成十进制，计算个体适应度值，并进行归一化，采用基于轮盘赌的选择操作，基于概率的交叉和变异操作，产生新的种群，并把历代的最优个体保存在新种群中，进行下一步遗传

操作

3.5.3 判断是否满足终止条件

经过遗传算子处理过后的群体 P_t 根据适应度函数大小排序后筛选出优良种群，再不断进化迭代 P_{t+1} 。当迭代的代数 t 达到最大迭代次数 $MAXGEN = 50$ 时，退出迭代，否则迭代继续。

3.5.4 仿真结果

优化结束后，其适应度进化曲线如 1-6 所示，各变量的进化曲线分别如 1-1，1-2，1-3，1-4，1-5 所示。优化结果 $T = [165.15\ 190.98\ 232.55\ 264.21\ 86.49]$ 算得最小面积为：473.38

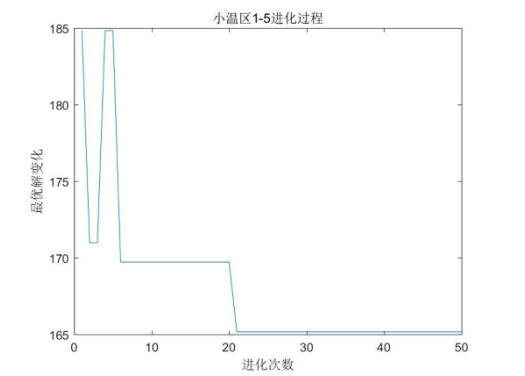


图 1-1

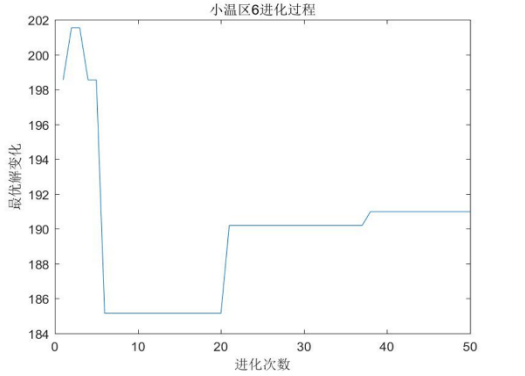


图 1-2

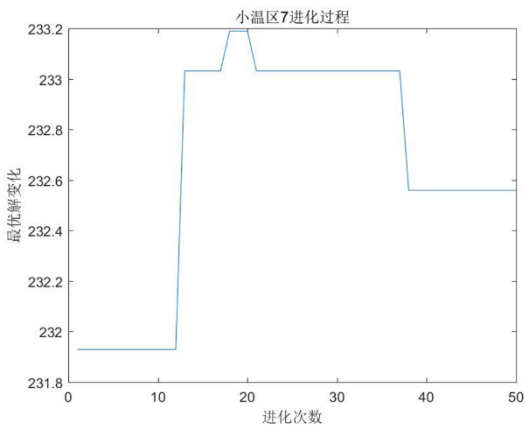


图 1-3

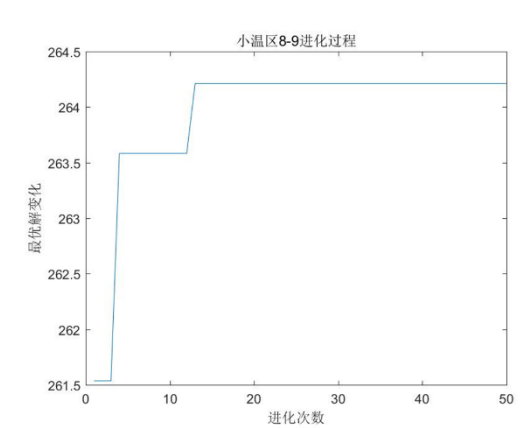


图 1-4

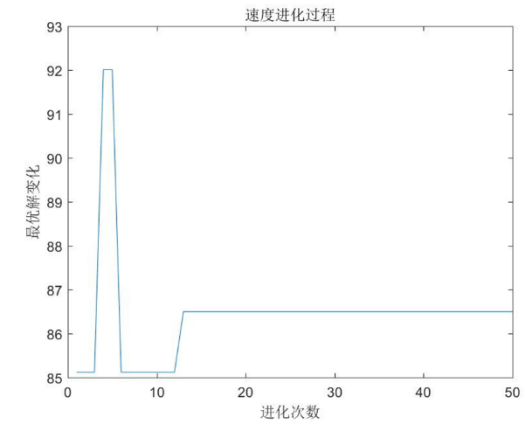


图 1-5

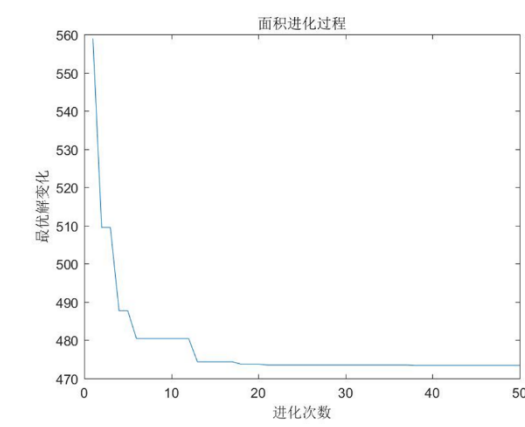


图 1-6

3.5.5 算法优化：多种群遗传算法 (Multi-population Genetic Algorithm, MPGA)

标准 GA 在求解组合优化等问题上具有独特的优势，但是极易早熟收敛。为了克服这一缺点，多种群遗传算法 (Multi-population Genetic Algorithm, MGA) 随之出现，并获得了广泛的关注和应用^[2-4]。MPGA 将标准 GA 的单种群划分为多个子群，保证了种群的多样性，每个子群内的个体按照标准 GA 进化，精英个体在子群间迁徙传播优势基因，从而避免早熟收敛。如文^[2]利用 MPGA 实现非线性动力学模型参数的辨识，预测橡胶波形发生器产生的冲击脉冲。文^[3]采用 MPGA 对常用的多孔吸声结构参数进行优化。

本文在解决的问题时，把变量的取值范围取到了小数点后两位，也就是每个变量有 $20 * 100 = 2000$ 种寻优范围，极易产生早熟现象。

本文提出改进多种群遗传算法的炉温最优曲线目标规划方法。根据目标函数和约束规则建立模型，利用多种群遗传算法对模型进行求解，采用移民算子种群在多种群之间关联及更新种群；移民算子种群保留每一次进化中所有种群中的部分最优炉温曲线，然后和每次进化中所有种群进行对比置换，保留每代进化中的最优炉温曲线，从而针对电子器件过炉能够有效地规划出目标函数较优的温度方案。具体过程如下：

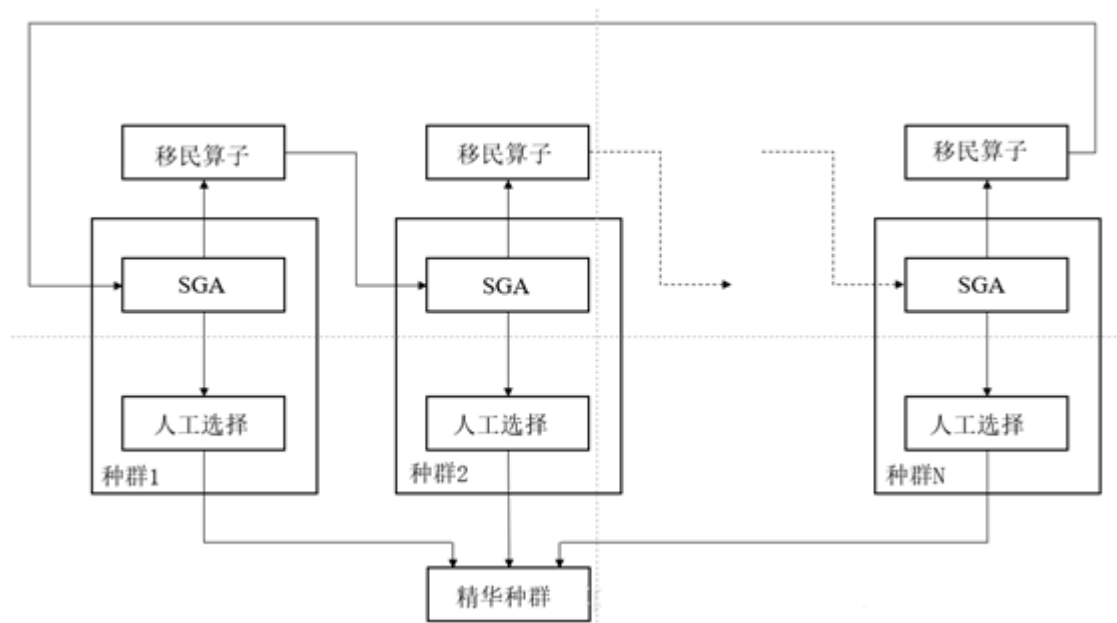
多种群遗传算法可以解决传统遗传算法存在的早熟收敛问题，多种群遗传算法 MPGA 在标准遗传算法 SGA 的基础上主要引入了以下概念：

(1) 突破 SGA 仅靠单个群体进行遗传进化的框架，引入多个种群同时进行优化搜索；不同的种群赋以不同的控制参数，实现不同的搜索目的。

(2) 各个种群之间通过移民算子进行联系，实现多种群的协同进化；最优解的获取是多个种群协同进化的综合结果。

(3) 通过人工选择算子保存各种群每个进化代中的最有个体，并作为判断算法收敛的依据。

MPGA 的算法结构示意图：



首先，我们以一个具体问题为例，使用英国谢菲尔德大学的遗传算法工具箱求解。
求复杂二元函数的最值：

$$\begin{aligned} \max f(x, y) &= 21.5 + x \sin(4\pi x) + y \sin(20\pi y) \\ &\begin{cases} -3 \leq x \leq 12.1 \\ 4.1 \leq y \leq 5.8 \end{cases} \end{aligned}$$

求解结果如图：

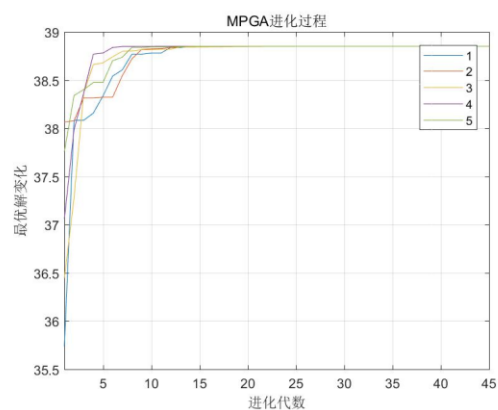
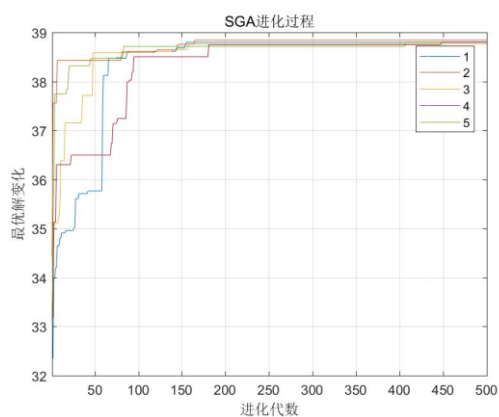


图 2 多种群遗传算法运行 5 次的进化过程图

从图 1 可见，5 次得到的优化结果均不同，标准遗传算法很不稳定，而且在接近 500 代的时候任然未稳定下来，说明最优解还有上升的可能。对于这种复杂的函数优化，使用标准的遗传算法已经很难得到最优解了。

从图 2 可见，多种群遗传算法运行 5 次的结果完全一致，说明多种群遗传算法稳定性很好，而且稳定性很好，而且使用的遗传代数很小，最大不超过 60 代，可见，多种群遗传算法的收敛速度快，适合复杂问题的优化。

- 代入我们解决的问题中，得到进化过程图

由于训练一次时间过长，所以并没有进化 500 次那么多，只是迭代了 200 次。但是仍然可以看到标准遗传算法很不稳定，在接近 200 代的时候仍然未稳定下来，而多种群遗传算法的收敛速度快，适合复杂问题的优化。

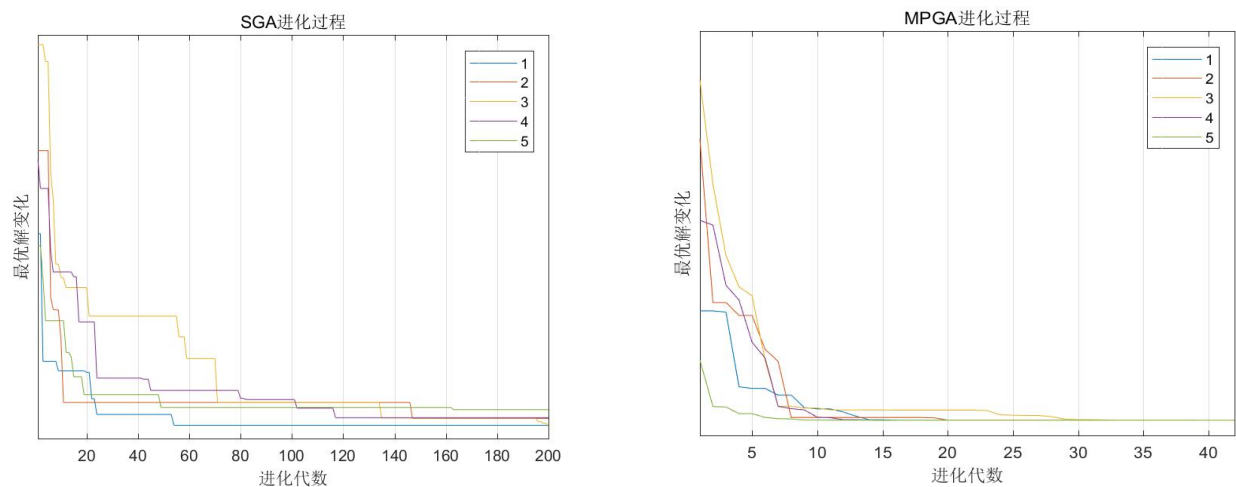


图 3 多种群遗传算法运行 5 次的进化过程图

四．总结

张桦灿：

就团队合作而言，我们小组成员之间可以进行有效的沟通，大家一起通力合作，共同攻克难关。虽然这其中也有一些波折，但我们最终还是都完成了。这也对于我在大学甚至之后的工作中解决相似问题是提供了蓝本。

而就项目本而言。我们小组的目标是利用算法来优化炉温曲线。在这一过程中，我明白了如何将理论与实践联系起来，同时，我对于整个流程有了更加深刻地认识。在解决几个重要的问题上积累了一定的经验，从而促进了我在平时生产或是学习中运用这一知识的能力。也使我对于这一领域有了更多的兴趣以及继续探索的热情。

韵皓宇：

这次仿真三级项目我们的主题是优化炉温曲线通过本次实践，在这次实践中，我深刻地了解了人工智能优化的理论的相关知识，并将其与实践相连。尤其是对于其中的遗传算法的部分以及其相关应用有了更深的体会。同时通过实践，我也能够了解到包括调参等在内的一系列操作的重要意义以及具体的实践方法，与此同时，我们也更好的理解了如何去分析收敛过程，从而使得自己对于专业知识的运用得到了更好的锻炼。

除此之外，这次的理论实践，还增强了我的团队合作意识，增进我和其他人一起解决问题的能力。这对于我在生活中，在学习上，甚至将来在工作上应对同其他人的交往也产生了积极的，有意义的影响。

姜志成：

在此次仿真项目之前我已经通过某次竞赛接触过人工智能优化算法的应用，借助这次三级项目，我们小组决定将未解决的问题进行完善。通过优化炉温曲线这一问题，不仅巩固了我对遗传算法的认知，同时我也学会了对标准遗传算法进行优化。从开始对项目内容的理解认识到项目计划的讨论和确定，从对项目的整体把握到实验创新点的寻找，并制定详细的项目方案和进程，以及项目当中重要的实践环节，整个项目过程中我不仅学到了许多我所感兴趣的、觉得有用的东西，更重要的是自己的思维能力、团队协作能力、实践能力都得到了提高，而且也学到了善于思考、积极总结的可贵精神。

梁绍文：

这次优化炉温曲线的三级项目，让我有机会将遗传算法的理论知识应用于实践当中用，使我对遗传算法以及相关应用有了进一步的认识。同时我也懂得了将理论应用于实践是学习知识最有效的途径。此外，我也认识到团队协作在团体项目的重要性。通过实践，大家都锻炼了自己行动实施的能力，力求做到最好。虽然项目进展不是一帆风顺的，但小组成员都以积极向上的心态去应对，正是通过努力使我们收货颇深。

参考文献

- [1] ABDULLAH S F, FITRIADHY A. Application of genetic algorithm for optimum hydrodynamic performance of twin pontoon floating breakwater[J]. Journal of Waterway Port Coastal and Ocean Engineering, 2020, 146(2): 04019040.
- [2] WEN J J, LIU C W, YAO H P, et al. A nonlinear dynamic model and parameters identification method for predicting the shock pulse of rubber waveform generator[J]. International Journal of Impact Engineering, 2018, 120:1-15.
- [3] WANG Y H, LIU S F, WU H Q, et al. On-demand optimize design of sound-absorbing porous material based on multi-population genetic algorithm[J]. e-Polymers, 2020, 20(1):122-132.
- [4] MA H P, FEI M R, JIANG Z H, et al. A multipopulation-based multiobjective evolutionary algorithm[J]. IEEE Transactions on Cybernetics, 2019, 50(2):689-702.
- [5] SHI X Q, LONG W, LI Y Y, et al. Different performances of different intelligent algorithms for solving FJSP: a perspective of structure[J]. Computational Intelligence and Neuroscience, 2018: 1-14.
- [6] GONG Y J, CHEN W N, ZHAN Z H, et al. Distributed evolutionary algorithms and their models: A survey of the state-of-the-art[J]. Applied Soft Computing, 2015, 34: 286-300.
- [7] 姜启源, 谢金星, 叶俊. 数学模型第五版 [M]. 北京: 高等教育出版社, 2018.
- [8] 杨世铭, 陶文铨. 传热学第四版 [M]. 北京: 高等教育出版社, 2006. 22

[9] 高金刚. 表面贴装工艺生产线上回流焊曲线的优化与控制 [D]. 上海交通大学,2007.

[10] 石宇强, 田永政, 张雨琦, 石小秋. 运用含复杂网络结构的多种群遗传算法求解 FJSP[J]. 计算机工程与应用, , : 1-

代码:

主训练函数 (GA) :

```
1. %% 定义遗传算法参数
2. %      NIND;          %个体数目
3. %      MAXGEN;        %最大遗传代数
4. %      PRECI;         %变量的二进制位数
5. %      GGAP;          %代沟
6. %      pc;            %交叉概率
7. %      pm;            %变异概率
8.
9. cor_matrix = [
10.     40 25 7 0.95 0.7 0.05
11.     50 25 7 0.95 0.7 0.01
12.     40 50 7 0.95 0.7 0.01
13.     40 100 7 0.95 0.7 0.01
14.     40 200 7 0.95 0.7 0.01
15.     40 50 10 0.95 0.7 0.05
16.     40 25 10 0.95 0.6 0.05
17.     40 30 10 0.90 0.7 0.01
18.     40 30 7 0.90 0.6 0.01
19.     40 25 7 0.90 0.6 0.01
20. ]
21.
22.%% 进化过程
23.for i = 1:length(cor_matrix)
24.    trace = cal_trace(cor_matrix(i,1), cor_matrix(i,2),...
25.        cor_matrix(i,3), cor_matrix(i,4), cor_matrix(i,5), cor_matrix(i,6), i)
26.end
27.
28.
29.function trace = cal_trace(NIND, MAXGEN, PRECI, GGAP, pc, pm, order)
30.T_part5 = 175;
31.T_part6 = 195;
32.T_part7 = 235;
33.T_part9 = 255;
34.T_part10 = 25;
35.
36.%% 画出函数图
37.lbx=T_part5 - 10;ubx=T_part5 + 10; %函数自变量 x 范围 175
38.lby=T_part6 - 10;uby=T_part6 + 10; %函数自变量 y 范围 195
39.lbz=T_part7 - 10;ubz=T_part7 + 10; %函数自变量 z 范围 235
40.lba=T_part9 - 10;uba=T_part9 + 10;%函数自变量 a 范围 255
41.lbb=65;      ubb=100; %函数自变量 b 的范围 65-100
42.trace=zeros(6,MAXGEN); %寻优结果的初始值
43.FieldD=[PRECI PRECI PRECI PRECI PRECI
44.    lbx lby lbz lba lbb
45.    ubx uby ubz uba ubb
```



```

46.     1 1 1 1 1
47.     0 0 0 0 0
48.     1 1 1 1 1
49.     1 1 1 1 1
50. ];                                     %区域描述器
51. Chrom=crtbp(NIND,PRECI*5);           %初始种群
52. %% 优化
53. gen=0;                               %代计数器
54. XY=bs2rv(Chrom,FieldD);              %计算初始种群的十进制转换
55. X=XY(:,1);Y=XY(:,2);Z=XY(:,3);A=XY(:,4);B=XY(:,5);
56. ObjV = zeros(20,1);
57. a = zeros(20,1);
58. for i = 1:length(X)
59.     speed = B(i);
60.     temp_stove = [X(i) Y(i) Z(i) A(i)];
61.     S = fit(speed, temp_stove);
62.     ObjV(i) = S;
63. end
64.
65. while gen<MAXGEN
66.     ObjVSel = [];
67.     FitnV=ranking(ObjV);               %分配适应度值
68.     SelCh=select('rws',Chrom,FitnV,GGAP); %选择
69.     SelCh=recombin('xovsp',SelCh,pc);   %重组
70.     SelCh=mut(SelCh,pm);                %变异
71.     XY=bs2rv(SelCh,FieldD);             %子代个体的十进制转换
72.     X=XY(:,1);Y=XY(:,2);Z=XY(:,3);A=XY(:,4);B=XY(:,5);
73.     for i = 1:length(X)
74.         speed = B(i);
75.         temp_stove = [X(i) Y(i) Z(i) A(i)];
76.         S = fit(speed, temp_stove);
77.         ObjVSel = [ObjVSel; S];
78.     end
79.     %计算子代的目标函数值
80.     [Chrom,ObjV]=reins(Chrom,SelCh,1,1,ObjV,ObjVSel); %重插入子代到父代，得到新种群
81.     XY=bs2rv(Chrom,FieldD);
82.     gen=gen+1;                           %代计数器增加
83.     %获取每代的最优解及其序号，Y 为最优解,I 为个体的序号
84.     [ans,I]=min(ObjV);
85.     trace(1:5,gen)=XY(I,:);              %记下每代的最优值
86.     trace(6,gen)=ans;                    %记下每代的最优值
87. end
88. %% 保存训练数据
89. filename = "simulate_5D_GA_" + string(order);
90. save(filename)
91. end

```

适应度函数：

```
1. function [area] = fit(speed,temp_stove)
2.
3. % %输入速度
4. % speed = 78
5. % %输入温度
6. % temp_stove = [173,198,230,257]
7.
8. speed_que1 = speed/60
9. distance = 50+35.5*11-5
10. whole_time = distance/speed_que1
11. Temp_pcb = [25]
12. pcb_w = 1
13. delta_Time = 0.5
14. kesi = 44
15. time_line =[0:delta_Time:whole_time]
16.
17.
18. for i = time_line
19.     location = i*speed_que1
20.     Temp_machine = Temp_S(temp_stove(1),temp_stove(2),temp_stove(3),temp_stove(4),25
        ,location)
21.     delta_Temp = (Temp_machine-Temp_pcb(pcb_w))*delta_Time/kesi
22.     Temp_pcb = [Temp_pcb Temp_pcb(pcb_w)+delta_Temp]
23.     pcb_w = pcb_w+1
24.
25.
26.     %斜率 max 值,min 值
27.     if delta_Temp/delta_Time > 3 || delta_Temp/delta_Time<-3
28.         area =999999
29.         return ;
30.     end
31.
32.
33. end
34.
35.
36. %温度上升过程中在 150?C~190?C 的时间
37. Temp_more150 = Temp_pcb(Temp_pcb>150)
38. freq_150_190 = Temp_more150(Temp_more150<190)
39. time_150_190 = length(freq_150_190)*delta_Time
40. if time_150_190<60 || time_150_190>120
41.     area = 999999
42.     return ;
43. end
44. %温度大于 217?C 的时间
45. freq_217 = Temp_pcb(Temp_pcb>217)
46. time_217 = length(freq_217)*delta_Time
```

```

47.     if time_217<40 ||time_217>90
48.         area = 999999
49.         return ;
50.     end
51.     %峰值温度
52.     temp_max = max(Temp_pcb)
53.     if temp_max<240 ||temp_max>250
54.         area = 999999
55.         return ;
56.     end
57.
58.
59.time_and_temp = [time_line;Temp_pcb(1:length(Temp_pcb)-1)]'
60.%大于 217 的面积
61.area217_t_T = time_and_temp(time_and_temp(:,2)>216.9999,:)
62.area = 0
63.for i = [1:length(area217_t_T)-1]
64.    area = area + ((area217_t_T(i,2)+area217_t_T(i+1,2))/2-217)*delta_Time
65.end
66.
67.end

```

热力学传热方程：

```

1. function [T] = Temp_S(temp1_5,temp_6,temp_7,temp8_9,temp10_11,s)
2. % 1 之前
3. s=s-10
4. kk=30
5. if s < 40.25+kk
6.     x = 40.25+kk
7.     a = inv([10,-10,1;x^2,x,1;2*x,1,0])*[0;temp1_5-25;0]
8.     T = a(1)*s^2+a(2)*s+25+a(3)
9. end
10.% 1-5
11.if s>=40.25+kk && s<182.25
12.    T = temp1_5
13.end
14.% 5-6 前半
15.if s>=182.25 && s<200
16.    a = (temp_6-temp1_5)/2/(17.75^2)
17.    T = a*(s-182.25)^2+temp1_5
18.end
19.% 5-6 后半
20.if s>=200 && s<217.75
21.    x = 2.5+15.25
22.    a = [x^2 x;2*x 1]\[temp_6-(temp_6+temp1_5)/2;0]
23.    T = a(1)*(s-200)^2+a(2)*(s-200)+(temp_6+temp1_5)/2
24.end
25.% 6-7 前半

```

```

26. if s >= 217.75 && s < 235.5
27.     a = (temp_7 - temp_6) / 2 / (17.75^2)
28.     T = a * (s - 217.75)^2 + temp_6
29. end
30. % 6-7 后半
31. if s >= 235.5 && s < 253.25
32.     x = 2.5 + 15.25
33.     a = inv([x^2 x; 2*x 1]) * [temp_7 - (temp_6 + temp_7) / 2; 0]
34.     T = a(1) * (s - 235.5)^2 + a(2) * (s - 235.5) + (temp_6 + temp_7) / 2
35. end
36.
37.
38. % 7-8 前半
39. if s >= 253.25 && s < 271
40.     a = (temp8_9 - temp_7) / 2 / (17.75^2)
41.     T = a * (s - 253.25)^2 + temp_7
42. end
43. % 7-8 后半
44. if s >= 271 && s < 288.75
45.     x = 2.5 + 15.25
46.     a = inv([x^2 x; 2*x 1]) * [(temp8_9 - (temp8_9 + temp_7) / 2) / 2; 0]
47.     T = a(1) * (s - 271)^2 + a(2) * (s - 271) + (temp8_9 + temp_7) / 2
48. end
49. % % 8-9
50. if s >= 288.75 && s < 324.25
51.     T = temp8_9
52. end
53.
54. % % >>>9 完美拟合
55. k3 = 200
56. k_c = 340
57. if s >= 324.25
58.     a = (temp10_11 - temp8_9) / (50^2)
59.     T = a * (s - 324.25)^2 + temp8_9
60.     if s > k_c
61.         b = 2 * a * (k_c - 324.25)
62.         a2 = (25 - b * k3) / k3^2
63.         T = a2 * (s - k_c)^2 + b * (s - k_c) + a * (k_c - 324.25)^2 + temp8_9
64.
65. end
66.
67.
68.
69. end

```

```

1. %% 标准遗传算法 SGA
2. function [Xx Yy] = SGA()
3. pc=0.7;
4. pm=0.05;
5. %定义遗传算法参数
6. NIND=40;           %个体数目
7. MAXGEN=200;        %最大遗传代数
8. NVAR=2;            %变量的维数
9. PRECI=20;          %变量的二进制位数
10. GGAP=0.9;         %代沟
11. trace=zeros(MAXGEN,1);
12. FieldD=[rep(PRECI,[1,NVAR]);[-3,4.1;12.1,5.8];rep([1;0;1;1],[1,NVAR])];%建立区域描述
    器
13. Chrom=crtbp(NIND, NVAR*PRECI);           %创建初始种群
14. gen=0;                                   %代计数器
15. maxY=0; %最优值
16. ObjV=ObjectFunction(bs2rv(Chrom, FieldD));%计算初始种群个体的目标函数值
17. while gen<MAXGEN                         %迭代
18.     FitnV=ranking(-ObjV);                 %分配适应度值
        (Assign fitness values)
19.     SelCh=select('sus', Chrom, FitnV, GGAP); %选择
20.     SelCh=recombin('xovsp', SelCh, pc);    %重组
21.     SelCh=mut(SelCh,pm);                  %变异
22.     ObjVSel=ObjectFunction(bs2rv(SelCh, FieldD)); %计算子代目标函数值
23.     [Chrom ObjV]=reins(Chrom, SelCh, 1, 1, ObjV, ObjVSel); %重插入
24.     gen=gen+1;                            %代计数器增加
25.     if maxY<max(ObjV)
26.         maxY=max(ObjV);
27.     end
28.     trace(gen,1)=maxY;
29. end
30.
31. %% 进化过程图
32. % plot(1:gen,trace(:,1));
33. Xx = gen;
34. Yy = trace(:,1);
35. %% 输出最优解
36. [Y,I]=max(ObjV);
37. X=bs2rv(Chrom, FieldD);
38. % disp(['最优值为: ',num2str(Y)])
39. % disp(['对应的自变量取值: ',num2str(X(I,:))])

```

多种群遗传算法

```

1. %% 多种群遗传算法
2. function [Xx Yy] = MPGA()
3. % clear;

```

```

4. % clc
5. % close all
6.
7. NIND=40;           %个体数目
8. NVAR=2;           %变量的维数
9. PRECI=20;         %变量的二进制位数
10. GGAP=0.9;        %代沟
11. MP=10;           %种群数目
12. FieldD=[rep(PRECI,[1,NVAR]);[-3,4.1;12.1,5.8];rep([1;0;1;1],[1,NVAR])]; %译码矩阵
13. for i=1:MP
14.     Chrom{i}=crtbp(NIND, NVAR*PRECI);           %创建初始种群
15. end
16. pc=0.7+(0.9-0.7)*rand(MP,1);    %在【0.7,0.9】范围内随机产生交叉概率
17. pm=0.001+(0.05-0.001)*rand(MP,1); %在【0.001,0.05】范围内随机产生变异概率
18. gen=0; %初始遗传代数
19. gen0=0; %初始保持代数
20. MAXGEN=10; %最优个体最少保持代数
21. maxY=0; %最优值
22. for i=1:MP
23.     ObjV{i}=ObjectFunction(bs2rv(Chrom{i}, FieldD));%计算各初始种群个体的目标函数值
24. end
25. MaxObjV=zeros(MP,1);           %记录精华种群
26. MaxChrom=zeros(MP,PRECI*NVAR); %记录精华种群的编码
27.
28. while gen0<=MAXGEN
29.     gen=gen+1;           %遗传代数加 1
30.     for i=1:MP
31.         FitnV{i}=ranking(-ObjV{i});           % 各种群的适应度
32.         SelCh{i}=select('sus', Chrom{i}, FitnV{i},GGAP); % 选择操作
33.         SelCh{i}=recombin('xovsp',SelCh{i}, pc(i));      % 交叉操作
34.         SelCh{i}=mut(SelCh{i},pm(i));           % 变异操作
35.         ObjVSel=ObjectFunction(bs2rv(SelCh{i}, FieldD)); % 计算子代目标函数值
36.         [Chrom{i},ObjV{i}]=reins(Chrom{i},SelCh{i},1,1,ObjV{i},ObjVSel); %重插入操作
37.     end
38.     [Chrom,ObjV]=immigrant(Chrom,ObjV); % 移民操作
39.     [MaxObjV,MaxChrom]=EliteInduvidual(Chrom,ObjV,MaxObjV,MaxChrom); % 人工选择精华种群
40.     YY(gen)=max(MaxObjV); %找出精华种群中最优的个体
41.     if YY(gen)>maxY %判断当前优化值是否与前一次优化值相同
42.         maxY=YY(gen); %更新最优值
43.         gen0=0;
44.     else
45.         gen0=gen0+1; %最优值保持次数加 1
46.     end
47. end
48. %% 进化过程图
49. Xx = gen;
50. Yy = YY;

```

```
51.%% 输出最优解
52. [Y,I]=max(MaxObjV);    %找出精华种群中最优的个体
53. X=(bs2rv(MaxChrom(I,:), FieldD));    %最优个体的解码解
54.% disp(['最优值为: ',num2str(Y)])
55.% disp(['对应的自变量取值: ',num2str(X)])
```