

2012 高教社杯全国大学生数学建模竞赛

承 诺 书

我们仔细阅读了中国大学生数学建模竞赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们参赛选择的题号是（从 A/B/C/D 中选择一项填写）：_____

我们的参赛报名号为（如果赛区设置报名号的话）：_____

所属学校（请填写完整的全名）：_____

参赛队员（打印并签名）：1. _____

2. _____

3. _____

指导教师或指导教师组负责人（打印并签名）：_____

日期：_____年____月____日

赛区评阅编号（由赛区组委会评阅前进行编号）：

2012 高教社杯全国大学生数学建模竞赛

编 号 专 用 页

赛区评阅编号（由赛区组委会评阅前进行编号）：

赛区评阅记录（可供赛区评阅时使用）：

| | | | | | | | | | | |
|-------------|--|--|--|--|--|--|--|--|--|--|
| 评 阅 人 | | | | | | | | | | |
| 评 分 | | | | | | | | | | |
| 备 注 | | | | | | | | | | |

全国统一编号（由赛区组委会送交全国前编号）：

全国评阅编号（由全国组委会评阅前进行编号）：

基于牛顿冷却定律的炉温曲线优化模型

摘要

本文通过建立回焊炉炉内温度与电路板速度控制的数学模型,解决了以下问题:首先,在问题一中,模拟并分析了焊接区域中心温度随时间的变化曲线,得到了各小温区温度设定条件下焊接区域中心在小温区 3、6、7 中点以及在小温区 8 结束处的温度,并绘制了相应的炉温曲线。其次,在问题二中,基于温区温度设定值和制程界限,计算出允许的最大传送带过炉速度。然后,在问题三中,通过调节温度设定值,使炉温曲线超过 217°C 到峰值温度所覆盖的面积最小,确定了各温区的设定温度和传送带过炉速度,并绘制了最优炉温曲线。本文在分析与解决这些问题时,结合了傅里叶热传导方程和牛顿冷却定律,利用差分方程和粒子群算法进行了数值模拟和优化,提出了科学、实用且创新的解决方案。

针对问题一:问题一旨在模拟焊接区域中心温度随时间的变化。为此,我们首先确定了各小温区的温度设定条件,然后建立了数学模型,通过傅里叶热传导方程和牛顿冷却定律,计算出在小温区 3、6、7 中点及小温区 8 结束处的温度分别为 124.6422°C 165.4861°C 188.1697°C 224.7746°C 。接着,我们绘制了温区 3、6、7 中点及温区 8 结束处的温度曲线,分析了焊接过程中的温度变化。最后,我们利用数值模拟和曲线拟合的方法,验证了模型的准确性,并将结果存储在 result.csv 文件中。

针对问题二:问题二在问题一的基础上,通过设定各温区的温度条件,推导出制程界限下允许的最大传送带过炉速度。我们利用牛顿冷却定律和差分方程,计算了不同速度下焊接区域中心的温度变化,并判断其是否符合制程要求。通过逐步调整传送带速度,找出了满足所有约束条件的最大速度值为 $78.0459\text{cm}/\text{min}$ 。此方法不仅提高了焊接效率,还确保了产品质量的稳定性和一致性,为实际生产提供了重要的参考数据。

针对问题三:问题三的目标是通过调节各温区的温度设定值,使炉温曲线超过 217°C 到峰值温度的面积最小。我们建立了单目标多变量的优化模型,以传送带速度和各温区温度为决策变量,通过积分计算目标面积。采用粒子群算法进行优化,搜索最优解。最终,我们确定了最优传送带速度为 $93.6582\text{cm}/\text{min}$,各温区温度设定值 ($180.854, 194.833, 232.121, 264.994$) 和最小面积值 395.9465 ,绘制了最优炉温曲线。这一优化过程不仅提高了焊接质量,还减少了能耗,具有重要的实际应用价值

关键词: 热传导方程 牛顿冷却定律 数据拟合 粒子群算法

一、问题重述

1.1 问题背景

随着电子元器件的不断发展，表面贴装技术逐渐兴起。表面贴装技术是一种将元器件直接粘贴在电路板的表面上的焊接方式，相比传统的插针焊接方式，表面贴装技术可以大大提高电路板的集成度和可靠性。为了适应表面贴装技术的需求，人们开发了各种新型的焊接设备和焊接工艺，如回流焊接、热风焊接等。

1.2 问题重述

问题一：在已知条件下对焊接区域中心的温度变化建立数学模型，更改温区的温度和传送带过炉速度，依据已建立的模型给出更改条件后焊接区域中心在小温区 3、6、7 中点以及在小温区 8 结束处的温度，画出相应的炉温曲线，并将每隔 0.5s 焊接区域中心的温度存放在提供的 `result.csv` 中。

问题二：在各温区温度设定值的条件下，依据制程界限推出允许的最大传送带过炉速度。

问题三：调节温度，使炉温曲线超过 217°C 到峰值温度所覆盖的面积最小，在此条件下，确定各温区的设定温度和传送带的过炉速度，画出最优炉温曲线，并给出相应的面积。

问题四：在满足制程界限的前提下，使以峰值温度为中心线的两侧超过 217°C 的炉温曲线尽量对称，确定最优炉温曲线，以及各温区设定的温度和传送带过炉速度，并给出相应指标。

二、问题分析

该问题在满足制程界限的前提下，通过调节炉内各温区的设定温度和传送带的过炉速度，建立相应的数学模型，绘制最优炉温曲线。结合题目中所列出的条件，绘制出思维导图如下：

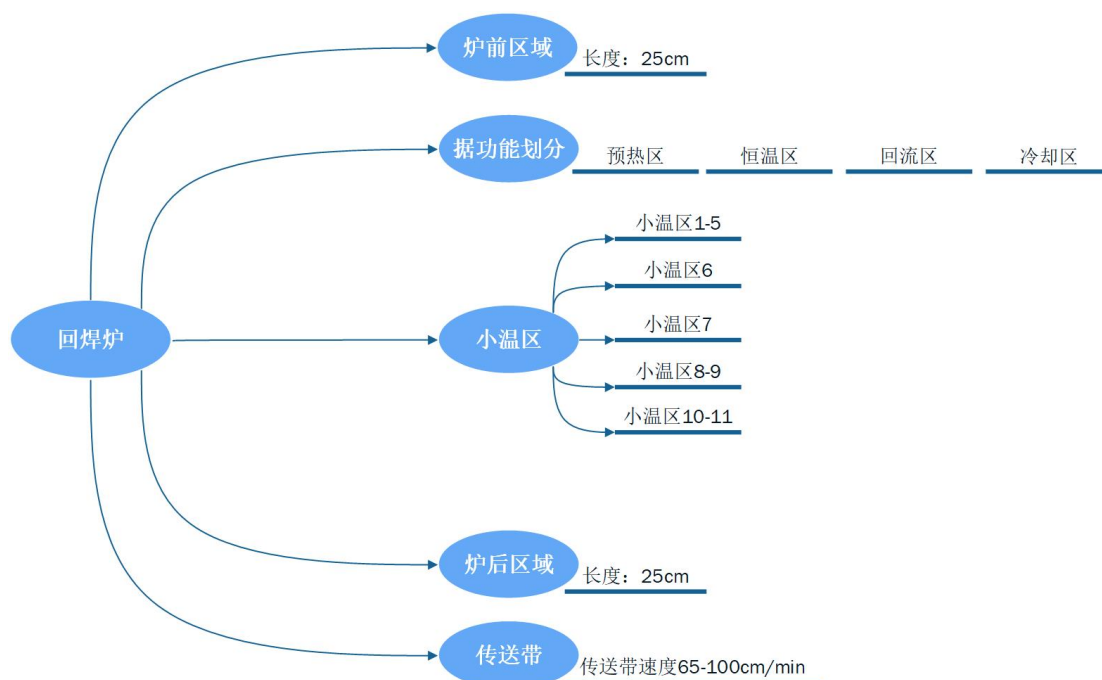


图 1 思维导图

2. 1 问题一的分析

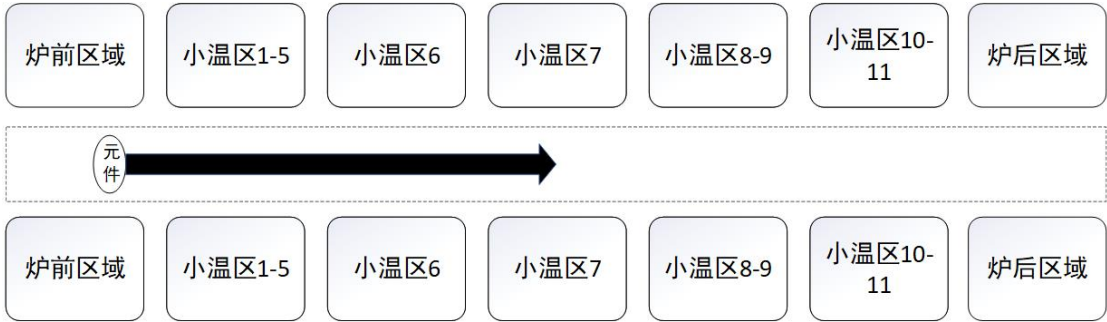


图 2 传送带工作流程图

传送带的过炉速度一定，各温区温度有设定值，电路板只在水平方向上随时间做匀速直线运动，本题是将炉温变化曲线与实际测得的数据比较，重在不断进行测试数据来调整和控制，要求建立可以反映温度变化的机理模型，可以类比分析用已知的与温度变化有关的物理定律，其中与探究温度变化相关的有基于热力学基本原理和傅里叶热学实验定律的热传导方程，牛顿冷却定律，从而建立微分方程模型。

表 1 温区设定温度表

| 温区号 | 温区 1~5 | 温区 6 | 温区 7 | 温区 8~9 | 温区 10~11 |
|------|--------|-------|-------|--------|----------|
| 设定温度 | 173°C | 198°C | 230°C | 257°C | 25°C |

2. 2 问题二的分析

表 2 约束条件表

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

问题二是在问题一的基础上，已知牛顿冷却定律的 k 值，各温区的温度设定值一定，以表 1 的制成界限为约束，求解最大过炉速度。经初步分析，允许通过的速度为一段连续的区间，可以从速度的最大值向下枚举。

表 3 问题二设定温度表

| 温区号 | 温区 1~5 | 温区 6 | 温区 7 | 温区 8~9 | 温区 10~11 |
|------|--------|-------|-------|--------|----------|
| 设定温度 | 182°C | 203°C | 237°C | 254°C | 25°C |

2. 3 问题三分析

问题三是单目标非线性的优化模型，其中决策变量相对于问题二来说多了各温区的温度，目标函数为使炉温曲线中超过 217°C 部分的面积最小，可以将目标函数与约束条件离散化，用遍历法求得结果，但是缺乏机理支撑，可以结合牛顿冷却定律或者热传导方程，用粒子群算法对模型进行优化，从而求得最优解

三、模型假设及符号说明

3. 1 模型假设

- (1) 各小温区的实际温度即为设定温度
- (2) 每个温区只影响该温区前后 5 厘米的温度

(3) 不考虑电路板的厚度

3.2 符号说明

表 4 符号说明表

| 符号 | 说明 | 单位 |
|-----|----------|-------------|
| v | 传送带过炉速度 | cm/min |
| t | 电路板移动时间 | s |
| x | 电路板位移 | cm |
| u | 焊接区域中心温度 | $^{\circ}C$ |
| k | 冷却系数 | |

四、模型的建立与求解

4.1.1 问题一模型的建立

焊接区域的中心温度受周围环境温度的影响,所以首先要确定电路板周围环境的温度,已知小温区温度,炉前区域、炉后区域以及小温区之间间隙的温度与其相邻的温度有关,这种变化通常遵循热传导的基本规律,根据傅里叶定律和傅里叶热传导方程,热传导问题可以用以下的偏微分方程来描述:

$$\partial u / \partial t = \alpha (\partial^2 u / \partial x^2 + \partial^2 u / \partial y^2 + \partial^2 u / \partial z^2) + f(x, y, z) \quad (1)$$

其中, $u(x, y, z, t)$ 表示温度分布, $f(x, y, z, t)$ 表示源项(可能是热源或热损失), α 为导热系数, t 为时间, x, y, z 为空间坐标。

本题中电路板只在水平方向上移动,所以可以简化为一维偏微分方程:

$$\partial u / \partial t = \alpha (\partial^2 u / \partial x^2) \quad (2)$$

该方程为抛物型偏微分方程,已知边界条件和初始条件,当达到热稳态时,焊接中心温度不再随时间变化,关于时间的偏导为零。分离变量并将其转化为常微分方程组,最终可得出这些区域的环境温度大致呈一次函数关系。

根据附录温度变化的数据使用工具箱初次进行曲线拟合,已知时间与位移的关系为:

$$x = vt \quad (3)$$

不同路段的环境温度不相同,因此根据周围环境温度的不同划分出每一段的长度,并分段拟合出温度随位移变化的函数关系。共分为六段,分别是炉前区域、小温区(1~5)、小温区 6、小温区 7、小温区(8~9)、以及最后一段(冷却区和炉后区域),其次根据附录进行数据拟合,但这种拟合没有考虑到改变温区温度时参数的变化,并且缺乏机理模型的支撑,可以用牛顿冷却定律对其进行改进。

牛顿冷却定律是描述温度高于周围环境的物体向周围媒质传递热量逐渐冷却时所遵循的规律。牛顿冷却定律的微分方程形式为:

$$du / dt = -k(u - u_{env}) \quad (4)$$

其中, u 是物体的温度, u_{env} 是周围环境的温度, k 是常数。这个方程描述了物体温度随时间的变化率与物体和周围环境温度差之间的关系。但本题 k 值未知,具体来说,当物体温度随时间变化时,其温度下降的速度与物体和环境之间的温度差成正比。这个比例系数在牛顿冷却定律中表示为 k 。通过实验或测量,可以得到不同时间点的温度数据,进而通过微分方程求解出 k 的值, k 值是动态变化的,有限差分法是一种求解微分方程数值解的近似方法,其主要原理是对微分方程中的微分项进行直接差分近似,从而将微分方程转化为代数方程组求解。

微分方程与代数方程最大的不同就是其包含微分项,这也是求解微分方程最

难处理的地方。差分方程又称递推关系式，是含有未知函数及其差分，但不含有导数的方程。满足该方程的函数称为差分方程的解。差分方程是微分方程的离散化。有限差分法的基本原理即使用近似方法处理微分方程中的微分项。为了得到微分的近似，我们最容易想到的即导数定义：

$$u'(t) = \lim_{t \rightarrow 0} \frac{u(t + \Delta t) - u(t)}{\Delta t} \approx \frac{u(t + \Delta t) - u(t)}{\Delta t} \quad (5)$$

联立(4) (5)两式，则差分方程可以表示为：

$$\frac{u(t + \Delta t) - u(t)}{\Delta t} = -k(u(t) - u_{env}(t)) \quad (6)$$

进一步化简得到：

$$u(t + \Delta t) = (1 - k\Delta t)u(t) + k\Delta t u_{env}(t) \quad (7)$$

这也是一阶差分代替导数的方程，此时可以令 Δt 等于一个确定的值，例如为 1，将收集到的附录数据依旧按分段处理，这些数据将用于差分拟合，构建一个最小二乘法问题，求解出差分方程中的未知参数 k ，随后可以通过与实际生产的炉温曲线比较进行误差检验，最后可以通过假设中的周围环境温度预设在此情况下新的炉温曲线数据。

4.1.2 问题一模型的求解

首先绘制出在不同位置处电路板周围环境的温度图形：

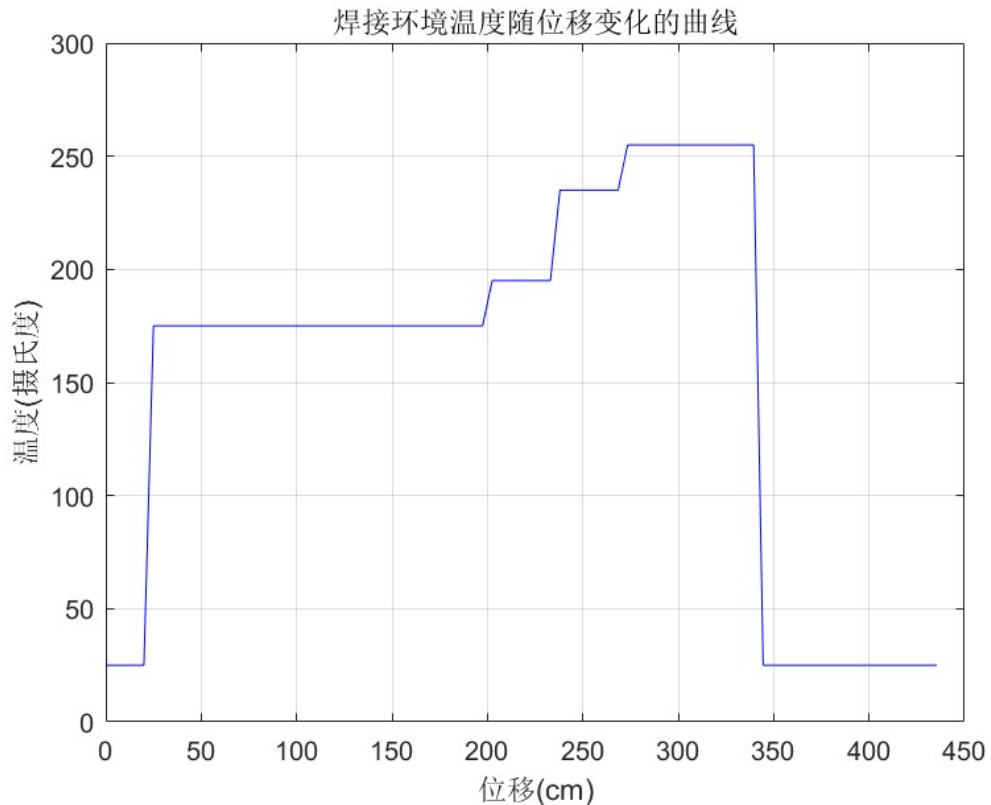


图 3 焊接环境温度随位移变化曲线

绘制出附录数据中温度随时间和位移变化曲线，分别为图 4 图 5：

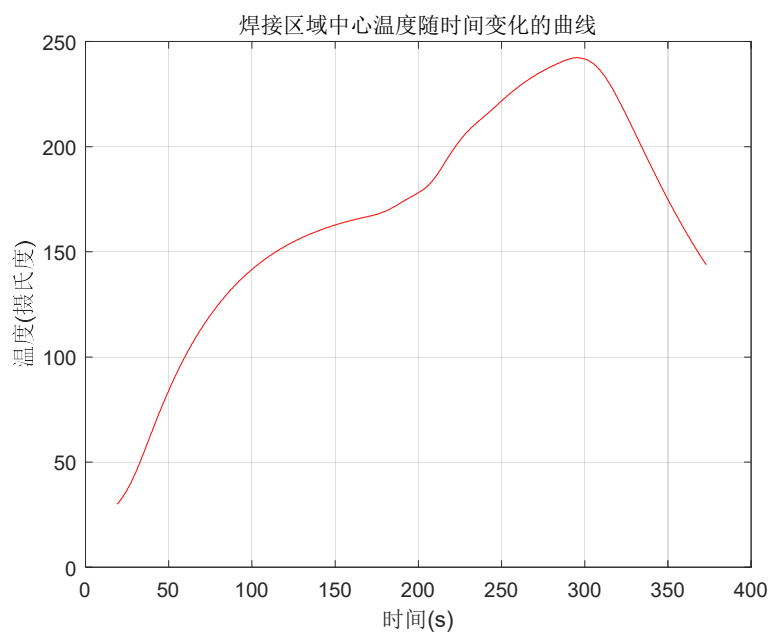


图 4 焊接区域中心温度随时间变化的曲线

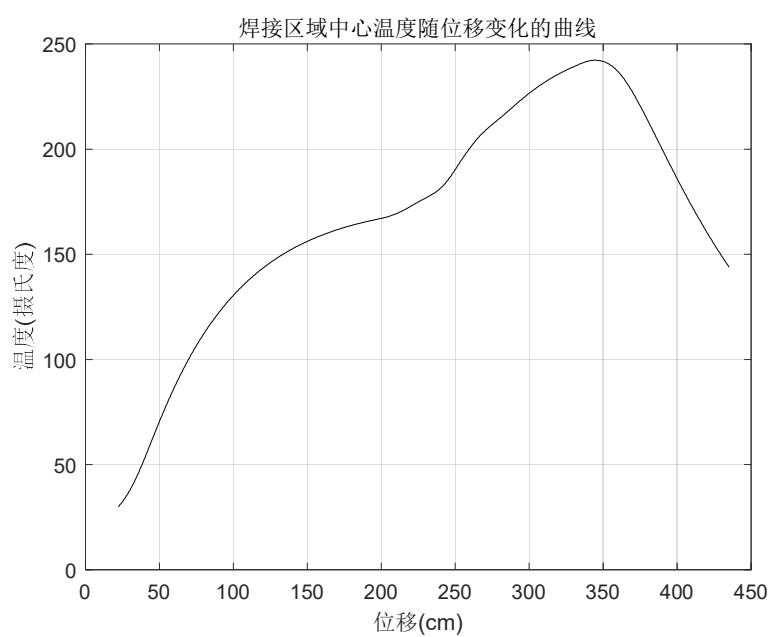


图 5 焊接区域中心温度随位移变化的曲线

对附录数据用 matlab 工具箱进行分段曲线拟合并与图 5 曲线对比得到炉温曲线如下：

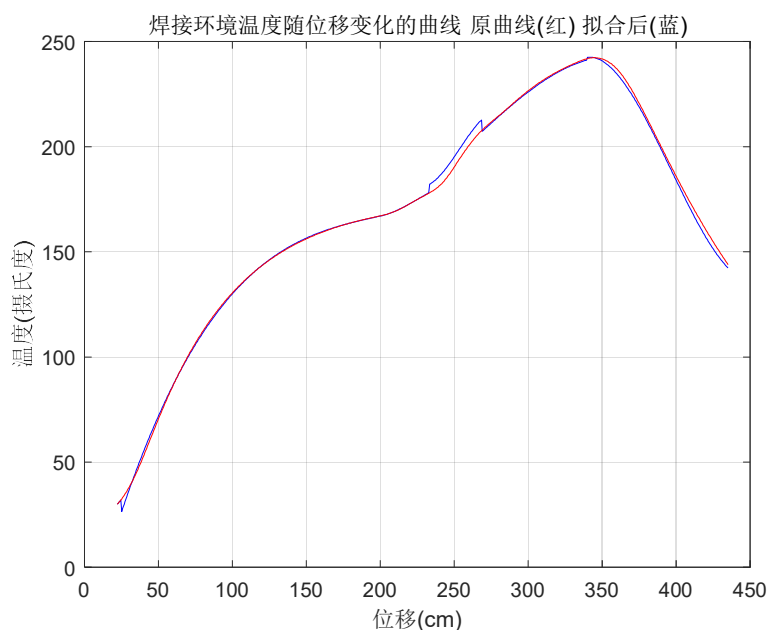


图 6 焊接环境温度随位移变化的曲线

通过分段拟合可以得到如下函数关系：

$$\begin{cases} u_1 = 2.056 \cdot 10^{-15} x^3 + 0.02939 x^2 - 0.5486 x + 27.75 & 0 < x \leq 25 \\ u_2 = 2.061 \cdot 10^{-15} x^3 - 0.01249 x^2 + 2.674 x - 33.09 & 25 < x \leq 202.5 \\ u_3 = -0.0001127 x^3 + 0.07699 x^2 - 17.12 x + 1413 & 202.5 < x \leq 238 \\ u_4 = -0.0005473 x^3 + 0.4163 x^2 - 104.5 x + 8853 & 238 < x \leq 273.5 \\ u_5 = -2.378 \cdot 10^{-5} x^3 + 0.01846 x^2 - 4.12 x + 442.7 & 273.5 < x \leq 344.5 \\ u_6 = -0.000162 x^3 - 0.1926 x^2 + 74.87 x - 9316 & 344.5 < x \leq 410.5 \end{cases} \quad (8)$$

在该段表达式中， $u_1 \sim u_6$ 分别表示在炉前区域、小温区 1-5、小温区 6、小温区 7、小温区 8-9 以及冷却区往后电路板的温度。

小温区 3、6、7 中点及小温区 8 结束处焊接区域中心的温度如下表：

表 5 焊接区域中心温度表

| 温区号 | 温区 3（中心） | 温区 6（中心） | 温区 7（中心） | 温区 8（结束） |
|-----------|------------|------------|------------|------------|
| 温区温度（摄氏度） | 138.1870°C | 172.0199°C | 198.5818°C | 230.6030°C |

将上述数据和附件对比，可以得出结果是可行的。

4. 1. 3 模型的改进

用曲线拟合更加侧重于数据驱动模型构建，并且没有考虑周围环境温度对参数的影响，为加深对系统内部作用机制的理解，需要用牛顿冷却定律来构建微分方程模型。

用牛顿冷却定律求得的焊接区域中心温度曲线与附录的原始数据曲线对比图如图 5：

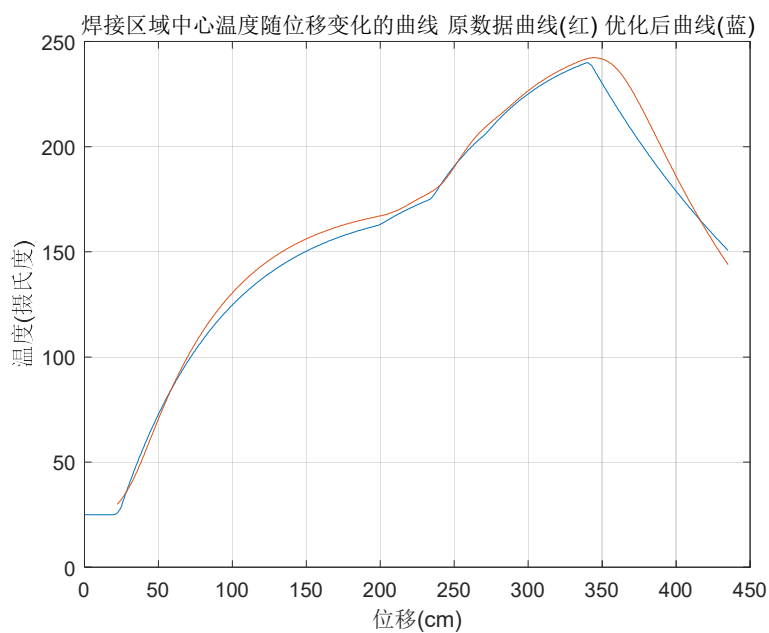


图 7 焊接区域中心温度随位移变化的曲线

分段求出的 k 值如下表：

表 6 冷却系数表

| 温区号 | 炉前区域 | 1-5 号温区 | 6 号温区 | 7 号温区 | 8-9 号温区 | 10-11 号温区 |
|-----|--------|---------|--------|--------|---------|-----------|
| k | 0.0108 | 0.0165 | 0.0162 | 0.0231 | 0.0200 | 0.0067 |

已求得 k 值，代入差分方程（7）并改变各温区温度的设定值为假设的设定值，求出新的焊接中心温度曲线图如下图：

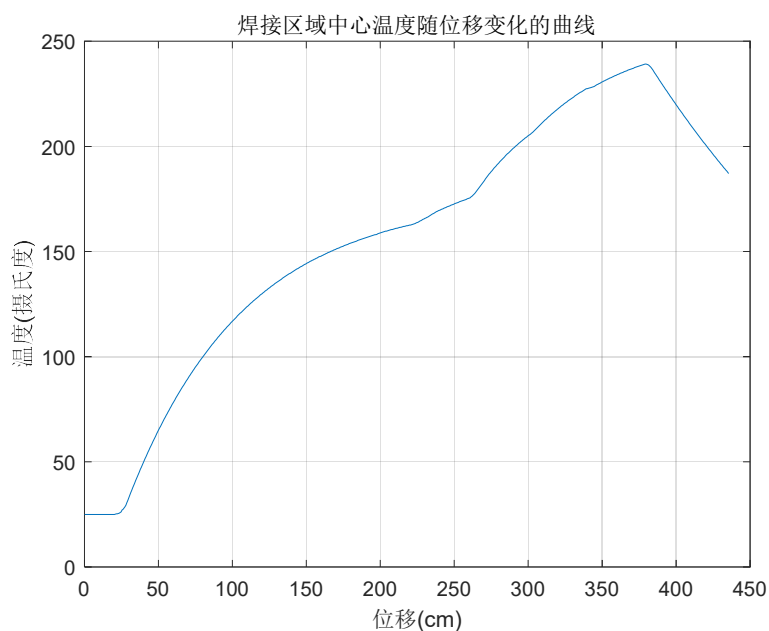


图 8 焊接区域中心温度随位移变化的曲线

小温区 3、6、7 中点及小温区 8 结束处焊接区域中心的温度如下表：

表 7 焊接区域中心温度表

| 温区号 | 温区 3 (中心) | 温区 3 (中心) | 温区 3 (中心) | 温区 3 (中心) |
|------------|------------|------------|------------|------------|
| 温区温度 (摄氏度) | 124.6422°C | 165.4861°C | 188.1697°C | 224.7746°C |

每隔 0.5s 焊接区域中心的温度已存放在提供的 result.csv 中。

4. 1. 4 误差分析

5.1.2 在解决问题时采用的是根据附件数据进行数据拟合的办法，未考虑热传导和牛顿冷却定律带来的影响，因此会产生误差。在 5.1.3 中对模型基于牛顿冷却定律进行改进，但未考虑在温差较大情况下冷却系数 k 的变化，因此会产生微小误差，但几乎可以忽略，得到的数据具有一定的可信度。

4. 2. 1 问题二模型的建立

问题二在问题一的基础上，可以将题目给定的速度区间内的值代入差分方程 (7) 中，求出该速度下焊接中心温度曲线，然后判断该焊接中心温度曲线是否符合题目中所给出的约束条件，题目中所给出的速度区间为：

$$65 \leq v \leq 100 \quad (9)$$

约束条件一：根据题目中给出的约束条件为温度上升时和温度下降时的斜率的绝对值小于等于三，在这里我们可以选择一个 Δt ，将 (10) 式近似为斜率，进行判断。

$$-3 \leq \frac{u_{t+\Delta t} - u_t}{\Delta t} \leq 3 \quad (10)$$

约束条件二：温度上升过程中在 150°C~190°C 的时间要大于等于 60 秒小于等于 120 秒。

$$60 \leq t_{190^\circ\text{C}} - t_{150^\circ\text{C}} \leq 120 \quad (11)$$

约束条件三：温度大于 217°C 的时间要大于等于 40 秒，小于等于 90 秒。

$$40 \leq t'_{217^\circ\text{C}} - t_{217^\circ\text{C}} \leq 90 \quad (12)$$

约束条件四：峰值温度要大于等于 240°C 小于等于 250°C

$$240 \leq u_{\max} \leq 250 \quad (13)$$

可以将速度代入差分方程 (7) 中，求出焊接中心温度曲线，然后依次判断该温度曲线是否符合这四个约束条件。

4. 2. 2 问题二模型的求解

首先以过炉速度 v 作为变量，使 v 从速度区间最大值 100cm/min 开始，每次使速度减小 1cm/min，并求出速度为 v 时的焊接中心温度曲线，然后判断该温度曲线是否符合约束条件，直到找出第一个符合约束条件的 v 值，然后从该 v 值每次使 v 增加 0.0001cm/min，并求出速度为 v 时的焊接中心温度曲线，然后判断该温度曲线是否符合约束条件直到找出最后一个符合约束条件的 v 值，即为问题二的答案。使用该策略可以使程序耗时较少，并求出较为精确的答案。根据每次搜索的速度值，画出的搜索曲线如下：

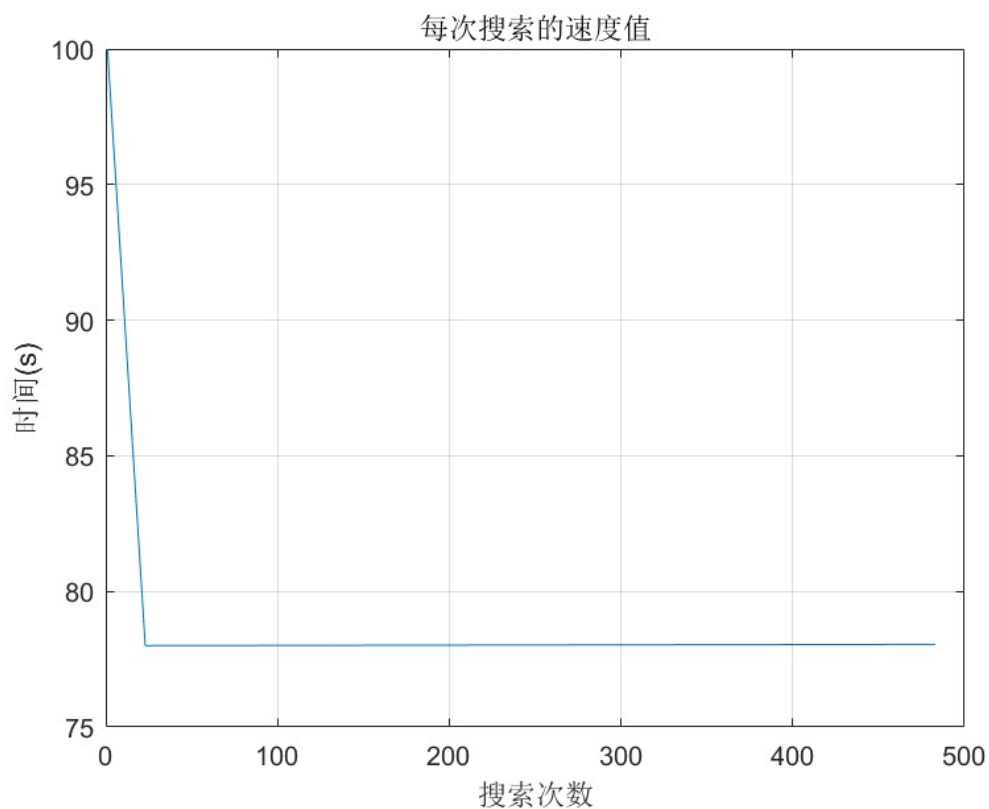


图 9 搜索曲线

最终通过该策略下求出的在约束条件下的允许的最大传送带过炉速度为 $v_{\max} = 78.0459 \text{ cm/min}$ 。

4. 2. 3 结果分析

绘制在最大传送带过炉速度为 $v_{\max} = 78.0459 \text{ cm/min}$ 速度下, 焊接区域的温度与电路板在传送带上运动时间拟合图如下:

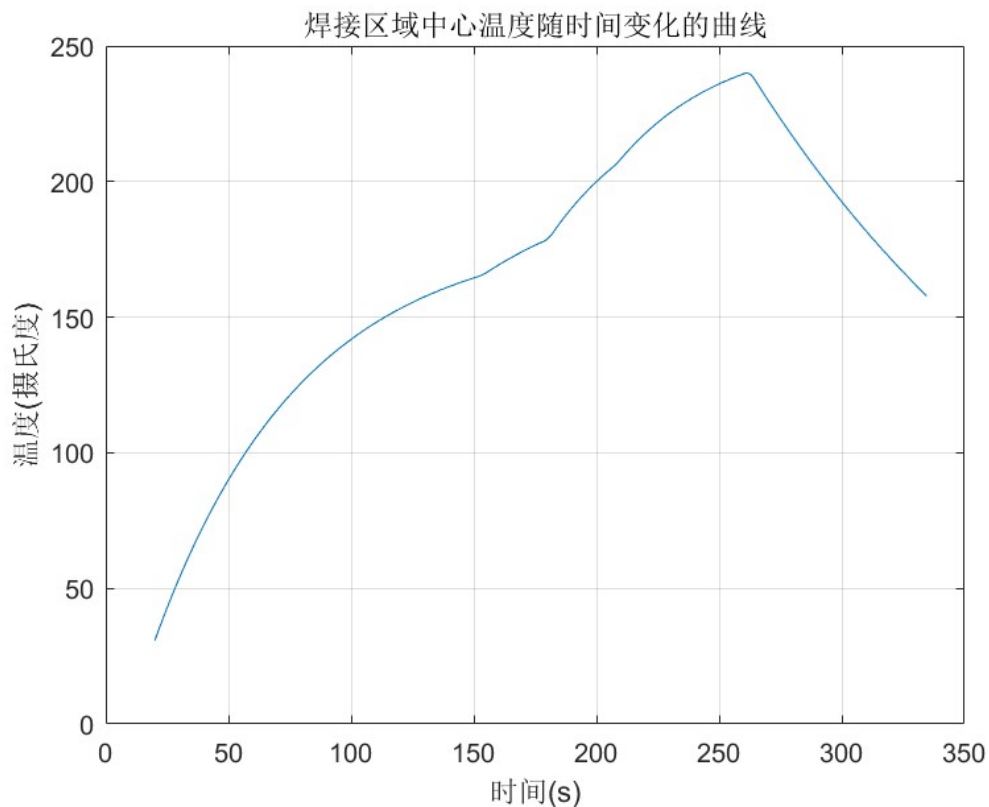


图 10 焊接区域中心温度随时间变化的曲线

由图像分析得出的数据显示该速度下的焊接中心温度曲线符合题目中的四个约束条件。

4.3.1 问题三模型的建立

该问题是单目标多变量的优化模型，目标函数为理想炉温曲线中温度超过 217°C 所覆盖的面积最小，决策变量为速度以及各温区的温度，约束条件为速度的限制，表 1 的制程限制以及各温区温度的限制。

关于决策变量：

决策变量为传送带的过炉速度 v ，小温区 1-5, 6, 7, 8-9 的温度分别记为 u_{15}, u_6, u_7, u_{89} 。

关于目标函数：

在炉温关于时间的曲线图中，可以对区域进行积分求面积为：

$$S = \min \int_{t_1}^{t_2} (u(t) - 217) dt \quad (14)$$

其中 t_1, t_2 分别为温度超过 217°C 的最小值对应的时间，最大值对应的时间。

积分可以看作是无限多个小区间上函数值的总和的极限，所以可以将区间 $[t_1, t_2]$ 等分成 n 个等宽的小区间，每个小区间的宽度记为 Δt ，在每个小区间的右端点上计算函数值 $u(t_i)$ ，其中 t_i 是第 i 个小区间的右端点。将这些函数值乘以小区间的宽度，并将所有这些乘积相加，就得到了积分的近似值。所以 S 的表达式可以用数学求和表达式表示为：

$$S = \min \sum_{i=t_1}^{t_2} u(t_i) \Delta t \quad (15)$$

关于约束条件：

关于速度的变化范围:

$$\frac{65}{60} < v < \frac{100}{60} \quad (16)$$

每个温区的变化范围为:

$$\begin{cases} 165 \leq u_{15} \leq 185 \\ 185 \leq u_6 \leq 205 \\ 225 \leq u_7 \leq 245 \\ 245 \leq u_{89} \leq 265 \end{cases} \quad (17)$$

温度斜率变化范围为:

$$-3 \leq \frac{du}{dt} = -k(u - u_{env}) \leq 3 \quad (18)$$

离散化为:

$$-3 \leq \frac{u(t_k) - u(t_k - \Delta t)}{\Delta t} \leq 3 \quad (19)$$

温度上升过程在 150℃ 到 190℃ 的时间限制为温度关于时间函数的反函数对应值范围:

$$60 \leq u^{-1} \leq 120 \quad (20)$$

根据前两问的炉温曲线图可以判断出在进入小温区 8-9 的位移时会出现温度大于 217℃ 的情况, 根据公式 (8) 可以得出时间 t_{89} 对应范围为:

$$\frac{273.5}{v} < t_{89} < \frac{344.4}{v} \quad (21)$$

峰值温度限制:

$$240 \leq \max u(t) \leq 250 \quad (22)$$

上述约束条件离散化后用 matlab 遍历求解。

4. 3. 2 问题三模型的求解

用粒子群算法对该目标优化:

粒子群算法的步骤:

1. 初始化: 随机生成一群粒子, 每个粒子具有位置和速度属性, 并计算每个粒子的适应度值。

2. 迭代搜索: 每个粒子根据自身的经验(个体极值)和群体的经验(全局极值)来更新自己的速度和位置。

3. 更新极值: 在每次迭代中, 如果粒子找到了更好的位置, 它会更新自己的个体极值。同时, 如果某个粒子的适应度值优于当前的全局极值, 它将成为新的全局极值。

4. 终止条件: 当满足一定的终止条件(如达到最大迭代次数或找到满足要求的解)时, 算法停止。

4. 3. 3 问题三结果分析

当 $t=0$ 时电路板焊接区域温度为 25℃, 最终计算得出各温区温度大小如下表:

表 8 温区温度表

| 温区号 | 1-5 温区 | 6 温区 | 7 温区 | 8-9 温区 |
|---------|---------|---------|---------|---------|
| 温度 (°C) | 180.854 | 194.833 | 232.121 | 264.944 |

传送带过炉速度为 78.0460cm/s, 在该温区与速度设定下的最小面积为 395.9465, 最优炉

温曲线如下图：

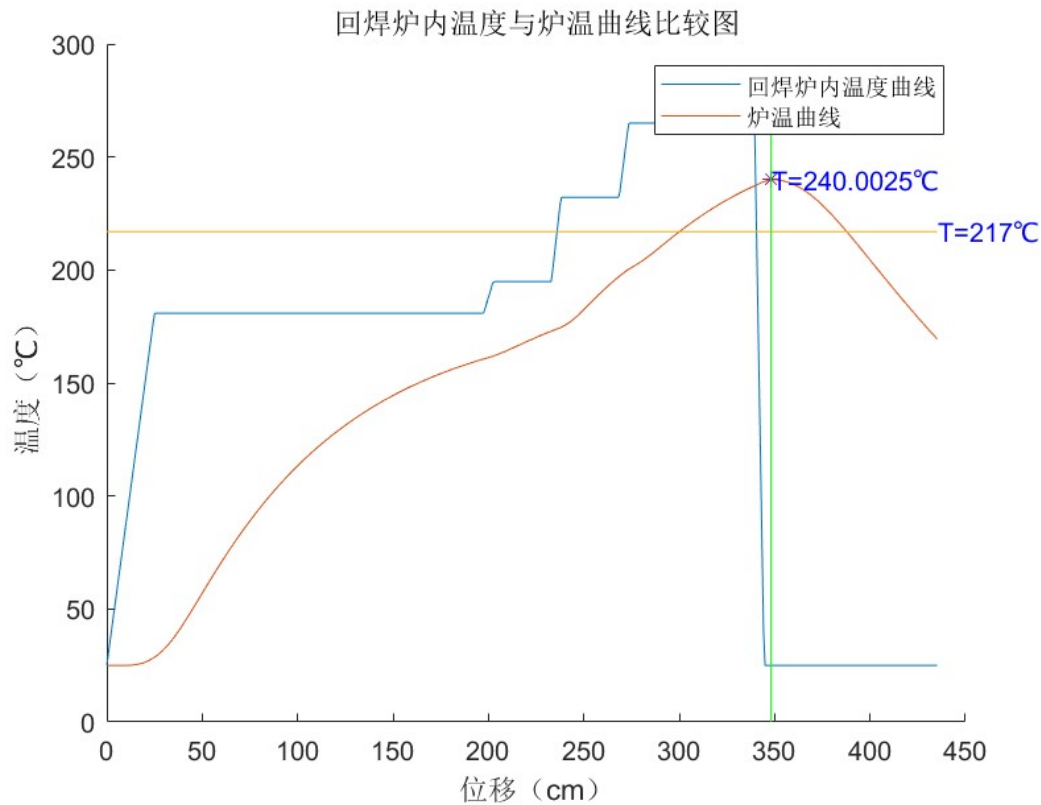


图 11 回焊炉内温度与炉温曲线比较图

结果分析：该问题结合牛顿冷却定律与粒子群算法，没有用到热力学定律，存在一定的误差，但相对于优化前，优化后的模型有深层理论支撑，更准确可靠。

五、模型的评价与推广

5.1.1 模型优点

科学性：模型基于热力学基本原理和傅里叶热学实验定律，通过微分方程和差分方程的构建，确保了模型的科学性和准确性。

实用性：通过调整各温区的设定温度和传送带的过炉速度，我们的模型能够有效地预测焊接区域中心的温度变化，为实际生产提供了有价值的参考。

创新性：在模型求解过程中，我们采用了数据拟合与机理模型相结合的方式，既利用了数据驱动的优势，又加深了对系统内部作用机制的理解。

5.1.2 模型缺点

误差来源：虽然我们对模型进行了改进，但在温差较大情况下，冷却系数 k 的变化可能未完全考虑，导致存在一定的误差。

模型简化：为了降低计算复杂度，我们对模型进行了一定的简化，如忽略电路板的厚度等，这可能会对结果产生一定影响。

5.1.3 模型的推广

我们的模型在表面贴装技术中的焊接工艺优化方面具有广泛的应用前景。通过调整炉内各温区的设定温度和传送带的过炉速度，可以实现对焊接区域中心温度的精确控制，从而提高焊接质量和生产效率。此外，该模型还可以进一步扩展到其他类似的热处理工艺中，如金属材料的退火、淬火等过程，为这些工艺的优化提供理论支持。

随着计算机技术和数值计算方法的不断发展，我们可以进一步优化模型，考

虑更多的影响因素，如电路板的材质、厚度、形状等，以提高模型的准确性和实用性。同时，结合机器学习和人工智能技术，可以实现对焊接过程的实时监控和智能调整，进一步提高生产效率和产品质量。

参考文献

[1] 许飞, 杨扬, 张江伟, 等. 牛顿冷却定律及其适用范围的探究[J]. 大学物理实验, 2022, 35 (02) :11-13. DOI:10. 14139/j. cnki. cn22-1228. 2022. 02. 003.

[2] 郭连红. 热传导方程解的稳定性以及 Matlab 数值模拟[J]. 高等数学研究, 2024, 27 (01) :123-126.

[3] 莫愿斌, 最优控制问题的粒子群算法求解研究. 广西壮族自治区, 广西民族大学, 2014-01-16.

[4] 罗仁泽, 一种基于最小二乘法优化系数的傅里叶有限差分偏移方法. 四川省, 西南石油大学, 2012-01-18.

附录

| 附录 1 |
|---|
| 介绍：支撑材料的文件列表 |
| result.csv 问题一代码.rar 问题二代码.rar 问题三代码.rar |

| 附录 2 |
|---|
| 介绍：问题一主程序代码 |
| <pre>%将附件中数据导入分别存在 s 和 t 中 num=xlsread('附件.xlsx','Sheet1'); % disp(s); s=num(:,1); t=num(:,2); %绘制出附件中数据的炉温曲线 plot(s,t,'r'); title('焊接区域中心温度随时间变化的曲线'); xlabel('时间(s)'); ylabel('温度(摄氏度)'); grid on; figure; x = s.*70./60; plot(x,t,'k'); title('焊接区域中心温度随位移变化的曲线'); xlabel('位移(cm)'); ylabel('温度(摄氏度)'); grid on; figure;</pre> |


```

groundx = [0,20,25,197.5,202.5,233,238,268.5,273.5,339.5,344.5,410.5,435.5];
groundt = [25,25,175,175,195,195,235,235,255,255,25,25,25];
plot(groundx,groundt,'B');
title('焊接环境温度随位移变化的曲线');
xlabel('位移(cm)');
ylabel('温度(摄氏度)');
grid on;
a = [];
b = [];
%分段拟合各温区的炉温曲线
for i = 1 : 5
    a = [a,x(i)];
    b = [b,t(i)];
end
plot(a,b);
for i = 6 : 310
    a = [a,x(i)];
    b = [b,t(i)];
end
plot(a,b);
for i = 311 : 371
    a = [a,x(i)];
    b = [b,t(i)];
end
plot(a,b);
for i = 372 : 431
    a = [a,x(i)];
    b = [b,t(i)];
end
plot(a,b);
for i = 432 : 553
    a = [a,x(i)];
    b = [b,t(i)];
end
plot(a,b);
for i = 554 : 709
    a = [a,x(i)];
    b = [b,t(i)];
end
plot(a,b);
y = [];
for i = 1:length(x)
    y = [y,Count(x(i))];
end

```

```

plot(x,y,'b',x,t,'r');
title('焊接环境温度随位移变化的曲线 原曲线(红) 拟合后(蓝)');
xlabel('位移(cm)');
ylabel('温度(摄氏度)');
grid on;
% 绘制拟合后的炉温曲线
res=[];
xx=[111.25,217.75,253.25,309];
for i=1:4
    res=[res,Count(xx(i))];
end
disp(res);
y = [];
for i = 1:length(x)
    y=[y,Count(x(i))];
end
plot(x,y,'b',x,t,'r');
T=[];
TX=[];
TT=[];
T0=[];
T1=[];
k0=0.5;
for i=1:length(t)
    if x(i)>339.5
        break
    elseif x(i)>=268.5
        TT=[TT,t(i)];
        TX=[TX,x(i)];
    end
end
for i=1:length(TT)-1
    T0=[T0,huihanlu(175,195,235,255,TX(i))];
    disp(x(i));
end
for i=1:length(TT)-1
    T=[T,TT(i)]
end
for i=2:length(TT)
    T1=[T1,TT(i)]
end
%使用最小二乘法分段拟合 k 值
fun1 = @(k,xdata) fun(k, T, T0);
options = optimoptions('lsqcurvefit', 'Algorithm', 'levenberg-marquardt');

```

```

[k,res]=lsqcurvefit(fun1,k0,T,T1,0,1,options);
disp(k);

xi=[];
Ti=[];
xi(1)=0;
Ti(1)=25;
dt=0.5;
k=[0.0108,0.0165,0.0162,0.0231,0.0200,0.0067];
%拟合出 k 值后将 k 值代入算出炉温曲线
for i=1:435.5/(70/60)*2
    kk=0;
    if i*0.5*70/60<=25
        kk=k(1);
    elseif i*0.5*70/60<=197.5
        kk=k(2);
    elseif i*0.5*70/60<=233
        kk=k(3);
    elseif i*0.5*70/60<=268.5
        kk=k(4);
    elseif i*0.5*70/60<=339.5
        kk=k(5);
    elseif i*0.5*70/60<=435.5
        kk=k(6);
    elseif i*0.5*70/60>435.5
        break;
    end
    cnt=(1-kk*dt)*Ti(i)+kk*dt*huihanlu(175,195,235,255,i*0.5*70/60);
    Ti=[Ti,cnt];
    xi=[xi,i*0.5*70/60];
end
% 将算出的炉温曲线与附件中的炉温曲线进行比对
plot(xi,Ti,x,t);
title('焊接区域中心温度随位移变化的曲线 原数据曲线(红) 优化后曲线(蓝)');
xlabel('位移(cm)');
ylabel('温度(摄氏度)');
grid on;
figure;

T1 = [];
x1 = [];
T1(1)=25;
x1(1)=0;

```

```

% 算出问题一条件下的炉温曲线
for i=1:1:435.5/(78/60)*2
    kk=0;
    if i*0.5*78/60<=25
        kk=k(1);
    elseif i*0.5*78/60<=197.5
        kk=k(2);
    elseif i*0.5*78/60<=233
        kk=k(3);
    elseif i*0.5*78/60<=268.5
        kk=k(4);
    elseif i*0.5*78/60<=339.5
        kk=k(5);
    elseif i*0.5*78/60<=435.5
        kk=k(6);
    elseif i*0.5*78/60>435.5
        break;
    end
    cnt=(1-kk*dt)*T1(i)+kk*dt*huihanlu(173,198,230,257,i*0.5*78/60);
    T1=[T1,cnt];
    x1=[x1,i*0.5*78/60];
end
plot(x1,T1);

% 将数据存入到 result.csv 文件中
t1=x1./(78/60);
res=[];
for i=1:length(t1)
    if T1(i)>30
        res=[res;t1(i),T1(i)];
    end
end
csvwrite('result.csv', res);
title('焊接区域中心温度随位移变化的曲线');
xlabel('位移(cm)');
ylabel('温度(摄氏度)');
grid on;

function res = Count(x)
%求拟合后的曲线
if x<=25
    res=2.056e-15*x*x*x+0.02939*x*x-0.5486*x+27.75;
elseif x<=197.5
    res=2.061e-05*x*x*x-0.01249*x*x+2.674*x-33.09;

```

```

elseif x<=233
    res=-0.0001127*x*x*x+0.07699*x*x-17.12*x+1413;
elseif x<=268.5
    res=-0.0005473*x*x*x+0.4163*x*x-104.5*x+8853;
elseif x<=339.5
    res=-2.378e-05*x*x*x+0.01846*x*x-4.12*x+442.7;
elseif x<=435.5
    res=0.000162*x*x*x-0.1926*x*x+74.87*x-9316;
elseif x>435.5
    end
end

```

```

function res = fun(k,T,T0)
% 最小二乘法拟合函数
dt=0.5;
res=(1-dt*k).*T+dt*k.*T0;
end

```

```

function T0 = huihanlu(t1,t2,t3,t4,x)
%得到回焊炉内各个位置的温度
if x<=20
    T0=25;
elseif x<=25
    T0=25+(t1-25).*(x-20)./5;
elseif x<=197.5
    T0=t1;
elseif x<=202.5
    T0=t1+(t2-t1).*(x-197.5)./5;
elseif x<=233
    T0=t2;
elseif x<=238
    T0=t2+(t3-t2).*(x-233)./5;
elseif x<=268.5
    T0=t3;
elseif x<=273.5
    T0=t3+(t4-t3).*(x-268.5)./5;
elseif x<=339.5
    T0=t4;
elseif x<=344.5
    T0=25+(t4-25).*(5-x+339.5)./5;
else
    T0=25;
end
end

```

```

function res= Get(t1,t2,t3,t4,v,x)
%用于得到问题一四个点的值
T1 = [25];
x1 = [0];
dt=0.5;
k=[0.0108,0.0165,0.0162,0.0231,0.0200,0.0067];
for i=1:435.5/(v/60)*2
    kk=0;
    if i*0.5*v/60<=25
        kk=k(1);
    elseif i*0.5*v/60<=197.5
        kk=k(2);
    elseif i*0.5*v/60<=233
        kk=k(3);
    elseif i*0.5*v/60<=268.5
        kk=k(4);
    elseif i*0.5*v/60<=339.5
        kk=k(5);
    elseif i*0.5*v/60<=435.5
        kk=k(6);
    elseif i*0.5*v/60>435.5
        break;
    end
    cnt=(1-kk*dt)*T1(i)+kk*dt*huihanlu(t1,t2,t3,t4,i*0.5*v/60);
    T1=[T1,cnt];
    x1=[x1,i*0.5*v/60];
end

t = x/(v/60)*2;
cnt = int16(t);
res = (T1(cnt)+T1(cnt+1))/2;
end

```

附录 3

介绍：问题二主程序代码

```

v=0;
vi=[];
% 从 100 到 65 依次递减 1 求出首个满足的速度
for i=100:-1:65
    res=check(i);
    vi=[vi,i];
%    disp(res);
    if res>0
        v=i;

```

```

        break
    end
end
% 从求出的速度 v 依次递增 0.0001 求出最后一个满足的速度
for i=v:0.0001:100
    if(check(i))
        v=i;
    else
        break;
    end
    vi=[vi,i];
end
disp(v);
x=1:length(vi);
plot(x,vi);
title('每次搜索的速度值');
xlabel('搜索次数');
ylabel('时间(s)');
grid on;
function flag = check(v)
flag=1;
t1=182;
t2=203;
t3=237;
t4=254;
T1=[];
x1=[];
T1(1)=25;
x1(1)=0;
dt=0.5;
%最小二乘法拟合出的常数 k
k=[0.0108,0.0165,0.0162,0.0231,0.0200,0.0067];
%求出炉温曲线
for i=1:435.5/(v/60)*2
    kk=0;
    if i*0.5*v/60<=25
        kk=k(1);
    elseif i*0.5*v/60<=197.5
        kk=k(2);
    elseif i*0.5*v/60<=233
        kk=k(3);
    elseif i*0.5*v/60<=268.5
        kk=k(4);
    elseif i*0.5*v/60<=339.5

```

```

        kk=k(4);
    elseif i*0.5*v/60<=435.5
        kk=k(6);
    elseif i*0.5*v/60>435.5
        break;
    end
    cnt=(1-kk*dt)*T1(i)+kk*dt*huihanlu(t1,t2,t3,t4,i*0.5*v/60);
    T1=[T1,cnt];
    x1=[x1,i*0.5*v/60];
end
t=x1./(v./60);
% plot(t,T1);
% title('焊接区域中心温度随时间变化的曲线');
% xlabel('时间(s)');
% ylabel('温度(摄氏度)');
% grid on;
Tmax=0;
T=[];
tt=[];
for i=1:length(T1)
    if(T1(i)>=30)
        T=[T,T1(i)];
        tt=[tt,t(i)];
    end
end
T1=T;
t=tt;
% plot(t,T1);
for i=2:length(T1)
    cnt=(T1(i)-T1(i-1))/dt;
    % disp(T1(i));
    % disp(T1(i-1));
    % disp(cnt);
    %判断是否符合斜率约束条件
    if cnt>3||cnt<-3
        flag=0;
    end
    if T1(i)>Tmax
        Tmax=T1(i);
    end
end
% disp(Tmax);
%判断是否符合峰值温度约束条件
if Tmax<240||Tmax>250

```



```

        flag=0;
    end
    smax1=0;
    smin1=0;
    % smax2=0;
    % smin2=0;
    smax3=0;
    smin3=0;
    for i=1:length(t)
        if T1(i)<150
            smin1=t(i+1);
        elseif T1(i)>190
            smax1=t(i-1);
            break;
        end
    end
    end
    % for i=length(T1):-1:2
    %     if T1(i)<150
    %         smax2=t(i-1);
    %     elseif T1(i)>190
    %         smin2=t(i+1);
    %         break;
    %     end
    % end
    % disp(smax1);
    % disp(smin1);

    cnt=smax1-smin1;
    % disp('*');
    % disp(cnt);
    %判断 150~190 的时间是否符合约束条件
    if cnt<60||cnt>120
        flag=0;
    end
    for i=1:length(T1)
        if T1(i)<=217
            smin3=t(i+1);
        else
            break;
        end
    end
    end
    for i=length(T1):-1:2
        if T1(i)<=217
            smax3=t(i-1);

```

```

        else
            break;
        end
    end
    end
    cnt=smax3-smin3;
    % disp(smax3);
    % disp(smin3);
    % disp("*");
    % disp(cnt);
    %判段大于 217 的时间是否符合约束条件
    if cnt<40||cnt>90
        flag=0;
    end
end
end

```

附录 4

介绍：支撑材料的文件列表

```

clc; clear;
global xm L u0
% 计算总长度，包含若干段长度
L=50+30.5*11+5*10;
% 参数数组，包含优化算法中需要用到的常量
xm=[6.683140781153043e-04;2.498727636628837e+04;8.076275611754757e-
04;1.431525270900793e+03;9.743913118484225e-04;8.279199794106737e+02;8.4
92734595661612e-04;6.547702252291799e+02;5.286841991732021e-04;1.337603
876051590e+03];
% 初始解
x=[1.792186138356053e+02;1.961138441942145e+02;2.287700888258109e+
02;2.649997068223830e+02;1.539745823162197]; %fv=3.957753927514420e+02
tic
% 调用粒子群优化算法
[x, fv]=SelPSO(@miaji, 60, 0.8, 2,
2,[185,205,245,265,100/6],[165,185,225,245,65/60],500,5);
% 60 为粒子数量。
% 0.8 为惯性权重
% 2 和 2 为学习因子
% 500 为最大迭代次数
toc
F=[25,x(1),x(2),x(3),x(4),25]; v=x(5);
T=L/v;
S=miaji(x);
t=0:0.5:T;

```

```

k=find(u0==max(u0));
figure
hold on
plot(t*v,f(v*t,F),t*v,u0,t*v,217*ones(length(t),1));
xlabel('位移 (cm) ');ylabel('温度 (°C) ');
title('回焊炉内温度与炉温曲线比较图');

text(L,217,'T=217°C','color','b');
plot(t(k)*v,u0(k),'*');
text(t(k)*v,u0(k),['T=',num2str(u0(k)),'°C'],'color','b');
line([t(k)*v,t(k)*v],[0,270],'color','g');
text(t(k)*v,270,['x=',num2str(t(k)*v),'cm'],'color','b');
legend('回焊炉内温度曲线','炉温曲线');

% 计算不同位置的温度分布
function y=f(x,F)
l=5; L=30.5; s=25;
x1=0; x2=25;
x3=x2+5*L+4*l; x4=x3+l;
x5=x4+L;x6=x5+l;
x7=x6+L;x8=x7+l;
x9=x8+2*L+1;x10=x9+l;
y=((F(2)-F(1))/s.*(x-x1)+F(1)).*(x<=x2)+F(2).*(x>x2).*(x<=x3)+((F(3)-F(2))/l
.*(x-x3)+F(2)).*(x>x3).*(x<=x4)+F(3).*(x>x4).*(x<=x5)...
+((F(4)-F(3))/l.*(x-x5)+F(3)).*(x>x5).*(x<=x6)+F(4).*(x>x6).*(x<=x7)...
+((F(5)-F(4))/l.*(x-x7)+F(4)).*(x>x7).*(x<=x8)+F(5).*(x>x8).*(x<=x9)...
+((F(6)-F(5))/l.*(x-x9)+F(5)).*(x>x9).*(x<=x10) + F(6).*(x>x10);
end

function t=h(t1,t2,t3,t4,v)
global xm L
F=[25,t1,t2,t3,t4,25]; T=L/v;
L1=25+5*30.5+5*5;
L2=L1+30.5+5;
L3=L2+30.5+5;
t1=L1/v;t2=L2/v;t3=L3/v;
dt=0.5;
m1=floor(t1/dt)+1;
m2=floor(t2/dt)+1;
m3=floor(t3/dt)+1;

l=0.015;
x=1e-4;
r1=xm(1)^2*dt/(x^2);

```

```

r2=xm(3)^2*dt/(x^2);
r3=xm(5)^2*dt/(x^2);
r4=xm(7)^2*dt/(x^2);
r5=xm(9)^2*dt/(x^2);
h1=xm(2);h2=xm(4);h3=xm(6);h4=xm(8);h5=xm(10);
n=ceil(l/x)+1; m=floor(T/dt)+1;
u=zeros(n,m);t=ones(m,1)*25;
u(:,1)=25;
u0=f(v*(0:floor(T/dt))*dt,F);
k=ceil(l/2/x);
A1=diag([1+h1*x,2*ones(1,n-2)*(1+r1),1+h1*x]);
A1=A1+diag([-1,-r1*ones(1,n-2)],1);
A1=A1+diag([-r1*ones(1,n-2),-1],-1);
B1=diag([0,2*ones(1,n-2)*(1-r1),0]);
B1=B1+diag([0,r1*ones(1,n-2)],1);
B1=B1+diag([r1*ones(1,n-2),0],-1);
C1=A1\B1;
c=zeros(n,m); c(1,:)=h1*u0*x;
c(n,:)=c(1,:);
c=A1\c;
for j=1:m1-1
    u(:,j+1)=C1*u(:,j)+c(:,j+1);
    t(j+1)=u(k,j+1);
end

A2=diag([1+h2*x,2*ones(1,n-2)*(1+r2),1+h2*x]);
A2=A2+diag([-1,-r2*ones(1,n-2)],1);
A2=A2+diag([-r2*ones(1,n-2),-1],-1);
B2=diag([0,2*ones(1,n-2)*(1-r2),0]);
B2=B2+diag([0,r2*ones(1,n-2)],1);
B2=B2+diag([r2*ones(1,n-2),0],-1);
C2=A2\B2;
c=zeros(n,m); c(1,:)=h2*u0*x;
c(n,:)=c(1,:);
c=A2\c;
for j=m1:m2-1
    u(:,j+1)=C2*u(:,j)+c(:,j+1);
    t(j+1)=u(k,j+1);
end

A3=diag([1+h3*x,2*ones(1,n-2)*(1+r3),1+h3*x]);
A3=A3+diag([-1,-r3*ones(1,n-2)],1);
A3=A3+diag([-r3*ones(1,n-2),-1],-1);
B3=diag([0,2*ones(1,n-2)*(1-r3),0]);

```

```

B3=B3+diag([0,r3*ones(1,n-2)],1);
B3=B3+diag([r3*ones(1,n-2),0],-1);
C3=A3\B3;
c=zeros(n,m); c(1,:)=h3*u0*x;
c(n,:)=c(1,:);
c=A3\c;
for j=m2:m3-1
    u(:,j+1)=C3*u(:,j)+c(:,j+1);
    t(j+1)=u(k,j+1);
end

```

```

A4=diag([1+h4*x,2*ones(1,n-2)*(1+r4),1+h4*x]);
A4=A4+diag([-1,-r4*ones(1,n-2)],1);
A4=A4+diag([-r4*ones(1,n-2),-1],-1);
B4=diag([0,2*ones(1,n-2)*(1-r4),0]);
B4=B4+diag([0,r4*ones(1,n-2)],1);
B4=B4+diag([r4*ones(1,n-2),0],-1);
C4=A4\B4;
c4(1,:)=h4*u0*x;
c4(n,:)=c4(1,:);
c4=A4\c4;

```

```

A5=diag([1+h5*x,2*ones(1,n-2)*(1+r5),1+h5*x]);
A5=A5+diag([-1,-r5*ones(1,n-2)],1);
A5=A5+diag([-r5*ones(1,n-2),-1],-1);
B5=diag([0,2*ones(1,n-2)*(1-r5),0]);
B5=B5+diag([0,r5*ones(1,n-2)],1);
B5=B5+diag([r5*ones(1,n-2),0],-1);
C5=A5\B5;
c5(1,:)=h5*u0*x;
c5(n,:)=c5(1,:);
c5=A5\c5;
for j=m3:m-1
    if t(j)>=t(j-1)
        u(:,j+1)=C4*u(:,j)+c4(:,j+1);
    else
        u(:,j+1)=C5*u(:,j)+c5(:,j+1);
    end
    t(j+1)=u(k,j+1);
end

```

```
end
```

```
function a=miaji(x)
```

```

global u0
u0=h(x(1),x(2),x(3),x(4),x(5));
u=u0(u0>=30);
tmp2=u(u>=217)-217;
t2=length(tmp2)*0.5;
umax=max(u);
tmp3=abs(u(2:length(u))-u(1:length(u)-1))/0.5;
kmax=max(tmp3);
s=u(u(2:length(u))>=u(1:length(u)-1));
s=s(s>=150);
s=s(s<=190);
t1=length(s)*0.5;

y=logical((umax<=250)*(umax>=240)*(t2>=40)*(t2<=90)*(kmax<=3)*(t1<=120)*
(t1>=60));
if y
    k=find(tmp2==max(tmp2));
    if k>2
        a=(sum(tmp2(1:k))-(tmp2(1)+tmp2(k))/2)*0.5;
    else
        a=(tmp2(1)+tmp2(k))*0.5/2;
    end
else
    a=inf;
end
end

function [xm, fv]=SelPSO(fitness, N , w,c1, c2,xmax,xmin,M,D)
%基于自然选择的粒子群算法
% fitness:待优化的目标函数
% N:粒子数目
% w: 惯性权重
% c1: 学习因子 1
% c2: 学习因子 2
% 自变量搜索域的最大值: xmax, 行向量
% 自变量搜索域的最小值: xmin 行向量
% M: 最大迭代次数
% D:自变量个数
% xm: 目标函数取最小值时的自变量值
% fv: 目标函数最小值
format long
Vm=0.2*(xmax-xmin);
x=zeros(N,D);v=x;
for i=1:N
    for j=1:D

```

```

        x(i,j)=xmin(j)+rand*(xmax(j)-xmin(j));
        v(i,j)=Vm(j)*(-1+2*rand);
    end
end
p=zeros(N,1);y=zeros(N,D);
for i=1:N
    p(i)=fitness(x(i,:));
    y(i,:)=x(i,:);
end
px=x(N,:); pg=p(N);
for i=1:(N-1)
    if p(i)<pg
        pg=p(i);
        px=x(i,:);
    end
end
f=zeros(N,1);
for t=1:M
    for i=1:N
        v(i,:)=w*v(i,:)+c1*(y(i,:)-x(i,:))*rand+c2*rand*(px-x(i,:));
        for j=1:D
            if v(i,j)>Vm(j)
                v(i,j)=Vm(j);
            end
            if v(i,j)<-Vm(j)
                v(i,j)=-Vm(j);
            end
        end
        x(i,:)=x(i,:)+v(i,:);
        tp1=x(i,:)<=xmax;
        tp2=x(i,:)>=xmin;
        if (sum(tp1)+sum(tp2))==2*D
            f(i)=fitness(x(i,:));
        else
            f(i)=inf;
        end
        if f(i)<p(i)
            p(i)=f(i);
            y(i,:)=x(i,:);
        end
        if p(i)<pg
            pg=p(i);
            px=y(i,:);
        end
    end
end

```

```
end
[~,sortx]=sort(f);
exindex=round((N-1)/2);
x(sortx((N-exindex+1):N))=x(sortx(1:exindex));
v(sortx((N-exindex+1):N))=v(sortx(1:exindex));
end
xm=px';
fv=pg;
```