

基于一维热传导方程的炉温曲线机理模型

摘要

在电子产品生产中，精确控制回焊炉温度以实现自动焊接电子元件到印刷电路板上是至关重要的。本文的研究集中在回焊炉各温区温度和传送带速度对**炉温曲线**的影响，通过建立基于**热传导方程**与**牛顿冷却定律**的温度分布模型来探索这些关系。

针对问题一，本文首先深入研究了炉前、炉后以及间隙区域的温度分布。为了准确描述这些区域的温度变化，采用了热传导方程并制定了相应的**边界条件**。然而，通过实验验证发现，基于**牛顿冷却定律**的简化模型在相同时间段内能够更好地拟合实际数据，因此选择了这种简化模型作为进一步研究的基础。模型参数的识别过程则利用了实验数据，分析得出不同温区的动态变化规律，通过最小二乘法确定了各温区的**冷却系数**，确保模型的准确性和适用性。

针对问题二，本文建立了以最大过炉速度为目标的**单目标优化模型**。约束条件包括满足问题一中建立的微分方程模型和工艺要求的**制程界限**。为了有效求解优化模型，采用了**二分法**，最终得出了最大传输速度为 **76.2980 cm/min** 的结果。这表明通过优化回焊炉温度和传送带速度的组合，可以最大化生产效率，同时保证产品质量和制程稳定性。

针对问题三，本文进一步探索了衡量累计高温区域大小所对应的阴影面积最小化的目标。基于各温区的问题和**过炉速度**作为决策变量，建立了优化模型。为了找到最小阴影面积，采用了**变步长搜索法**，这种方法有效地在复杂的多变量空间中寻找到了最优解，进一步提升了生产过程中的效率和品质。

针对问题四，本文着眼于高温区炉温曲线的对称性及高温累计区域面积两个指标。首先定义了**高温区炉温曲线对称性指标**，并基于此建立了**双目标规划模型**，以同时考虑这两个目标。为了将多目标优化转化为单目标优化，采用了**动态综合加权法**，最终得出在尽量对称的情况下达到最小阴影面积的结果。

关键词：热传导方程、牛顿冷却定律、最小二乘法、变步长搜索法

一、 问题重述

1.1 问题背景

在集成电路板等电子产品的生产过程中，回焊炉的使用至关重要。回焊炉是一种通过加热将安装有各种电子元件的印刷电路板上的焊料熔化，使电子元件自动焊接到电路板上的设备。回焊炉的内部设置有多温区，这些温区从功能上分为四个大温区：预热区、恒温区、回流区和冷却区。每个大温区内部又划分为若干小温区。这些温区的温度控制对焊接过程的质量有显著影响，如下图所示：

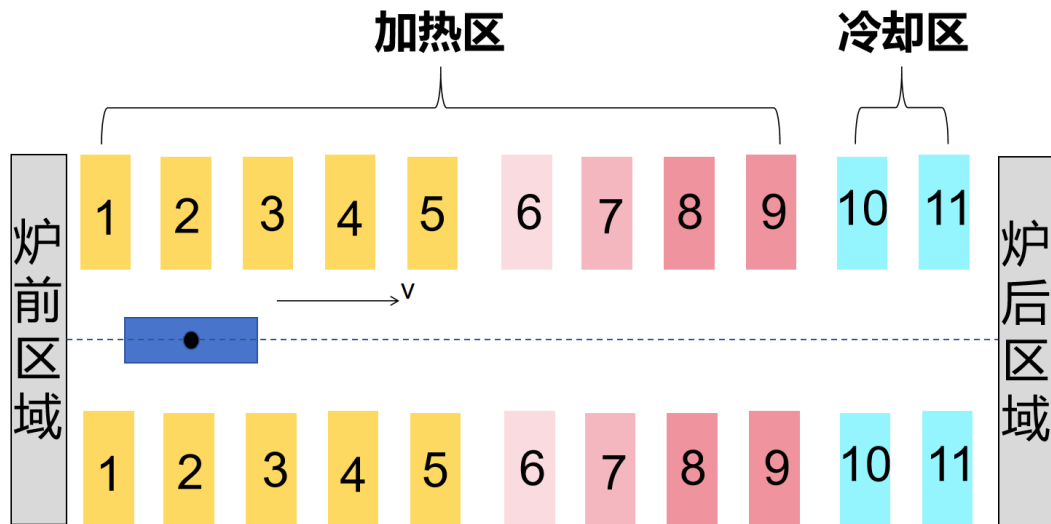


图1 回焊炉结构简图

生产过程中，温度传感器会监测焊接区域中心的温度变化，这些数据被称为炉温曲线。炉温曲线反映了焊接区域中心温度随时间的变化情况，对产品质量有重要影响。为了确保焊接质量，可以通过调整各温区的设定温度和传送带的过炉速度来控制产品质量。

在本题中，回焊炉内有11个小温区及炉前区域和炉后区域，每个小温区的长度为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的长度均为25 cm。设某次实验的温度分别为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）；传送带过炉速度为70 cm/min；焊接区域的厚度为0.15 mm。实际生产中，可以在此基础上，一定范围内对温区温度、传送带过炉速度等进行调整，以控制产品品质。同时，在回焊炉电路板焊接过程中，炉温曲线需满足一定的要求，称为制程界限，如表1。

表1 制程界限

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的时间	40	90	s
峰值温度	240	250	°C

1.2 问题提出

问题 1: 建立焊接区域温度变化的数学模型。在给定传送带速度和温区设定温度的条件下, 计算焊接区域中心温度的变化情况, 绘制相应的炉温曲线, 并记录特定时间间隔的温度数据。**问题要求:** 传送带过炉速度为 78 cm/min, 各温区温度的设定值分别为 173°C (小温区 1~5)、198°C (小温区 6)、230°C (小温区 7) 和 257°C (小温区 8~9)。

问题 2: 在设定各温区温度的情况下, 确定允许的最大传送带过炉速度, 使焊接过程满足制程界限的要求。**问题要求:** 各温区温度的设定值分别为 182°C (小温区 1~5)、203°C (小温区 6)、237°C (小温区 7)、254°C (小温区 8~9)。

问题 3: 优化炉温曲线, 使焊接区域中心温度超过 217°C 到峰值温度的时间面积最小。通过调整各温区的设定温度和传送带速度, 找到最优方案并计算该面积。

问题 4: 在满足问题 3 要求的基础上, 进一步优化炉温曲线, 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。结合对称性的要求, 确定最优的炉温曲线、各温区设定温度及传送带速度, 并给出相应的指标值。

二、问题分析

2.1 问题一的分析:

该问题的核心在于建立一个准确反映焊接区域温度变化规律的数学模型。当前大多数关于焊接区域温度变化的研究主要采用数据驱动的方法进行控制和调整, 例如通过 PID 控制原理来调节回焊炉温度。然而, 这些方法缺乏对焊接过程中物理机制的深入理解, 不利于全面了解电路板在回焊炉中的实际焊接过程。因此, 本题的难点在于建立一个机理模型, 该模型不仅需要反映出电路板焊接的物理过程, 还要与题目中给定的温度变化曲线相吻合。为此, 我们拟在热传导方程和牛顿冷却定律的基础上, 构建一个能够准确描述温度变化规律的数学模型。

2.2 问题二的分析:

本题在已确定的参数基础上, 给定各温区的温度, 要求找到满足题目所规定制程界限条件下的最大传送带速度。为了解决这个问题, 我们可以采用二分法来初步确定最大速度的范围, 然后在这个范围内进行详细搜索, 以找到符合要求的最大传送带速度。这个方法能够有效减少计算量, 并在合理时间内获得结果。

2.3 问题三的分析:

问题要求确定各温区的设定温度和传送带的过炉速度, 使得理想的炉温曲线超过 217°C 到峰值所覆盖的面积最小。本题难点在于变量个数增多, 解空间变大。但可通启发式算法进行求解。

2.4 问题四的分析:

在第三个问题的基础上, 进一步优化各温区温度和传送带速度, 使得炉温曲线在以峰值温度为中心线的两侧尽量对称, 同时确保超过 217°C 到峰值温度之间的面积最小化。这个问题的难点在于如何设定和衡量对称性指标, 以及如何在多目标优化中平衡不同目标。可以通过动态综合加权法, 将多目标优化问题转化为单目标优化问题, 在保证对称性的前提下, 找到面积最小化的最优解。

三、模型假设

- 1、焊接过程中电路板不会自产热
- 2、电路板进入焊接区域不会产生空气对流影响不同区域之间的温度
- 3、各电路板相互质地均匀
- 4、不同小温区之间若间隙较小则间隙温度呈线性变化，且不考虑不同小温区边界部分温度变化
- 5、炉前炉后区域温度较平稳的缓慢变化进而趋向室温

四、符号说明

本文中涉及到的主要变量符号说明：

符号	符号含义及说明
$u(x, t)$	电路板在 t 时刻 x 厚度上的温度
u_{air}	电路板进入回焊炉中外界温度
k	冷却系数
v	传送带过炉速度
u_{max}	峰值温度
T_1	温区 1-5 设置的温度
T_2	温区 6 设置的温度
T_3	温区 7 设置的温度
T_4	温区 8-9 设置的温度
S	炉温曲线超过 217℃ 到峰值温度覆盖面积
l	对称度

注：未申明的变量以其在符号出现处的具体说明为准。

五、模型的建立和求解

5.1 建模前的准备

5.1.1 实际问题处理

如图 2 所示，电路板在回焊炉内焊接时，实际上是三维立体温度扩散。本文不计空气对流，因此可以忽略平行板高度和宽度，将立体平行板简化为只与厚度有关的一维空间

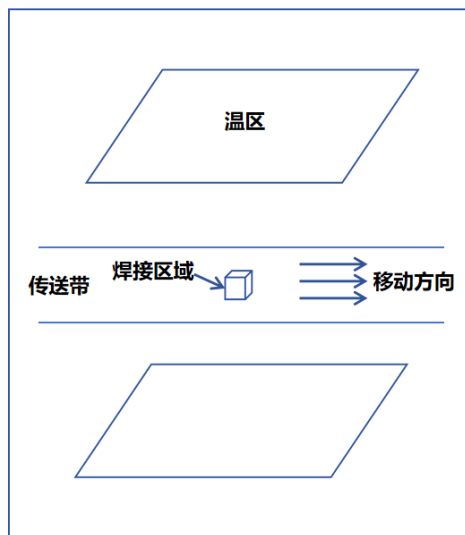


图 2 电路板在回焊炉传送示意图

5.1.2 环境温度分布

根据热传导方程：

$$\frac{\partial u(x,t)}{\partial t} = a^2 \frac{\partial^2 u(x,t)}{\partial x^2} \quad (1)$$

到达稳态后，温度不随时间变化，即：

$$\frac{\partial u(x,t)}{\partial t} = 0 \quad (2)$$

所以：

$$a^2 \frac{\partial^2 u(x,t)}{\partial x^2} = 0 \quad (3)$$

故：

$$u(x,t) = hx + b \quad (4)$$

由（4）可知两个小温区之间的温度随空间距离呈线性相关

5.1.3 牛顿冷却定律

牛顿指出系统温度损失速度与系统和环境的温差成正比，即牛顿冷却定律：

$$\frac{du}{dt} = -k(u - u_{air}(t)) \quad (5)$$

5.2 问题一模型建立

5.2.1 基于热传导方程的温度分布模型

在回焊炉内进行焊接时，电路板的温度变化遵循一维热传导方程，即：

$$\frac{\partial u(x,t)}{\partial t} = a^2 \frac{\partial^2 u(x,t)}{\partial x^2} \quad (6)$$

电路板进入回焊炉之前的初始温度为 25°C，因此初始条件为：

$$u(x, 0) = 25 \quad (7)$$

依据牛顿冷却定律，电路板在其两端的温度满足以下边界条件：

$$\begin{cases} -\frac{\partial u(x,t)}{\partial x} + \sigma u(t) = \sigma u_e, & x=0 \\ \frac{\partial u(x,t)}{\partial x} + \sigma u(t) = \sigma u_e, & x=l \end{cases} \quad (8)$$

5.2.2 基于牛顿冷却定律的温度分布模型

由于电路板的导热性能较好，且电路板的厚度仅为 0.15mm,当电路板受热时，其内部温度很快接近一致，即同一时刻，温度在电路板厚度上分布一致，故可将电路板视为一个整体，不需考虑电路板内部的传热情况，因此在上述基于热传导方程的温度分布模型中只有边界条件起作用，从而将模型简化为：

$$\frac{d\mu(t)}{dt} = -k(\mu(t) - \mu_{air(t)})$$

其中， $\mu(t)$ 为 t 时刻系统的温度， $\mu_{air(t)}$ 为 t 时刻系统外界温度， k 为冷却系数

5.3 问题一模型求解

5.3.1 求解前的准备工作

本文选择使用热传导方程模型对温度 u 随位置 x 与时间 t 的变化进行求解并通过牛顿冷却定律对模型进行优化，而在两模型中都存在未知参数，故需要先确定各模型中参数的最优值再进行求解。

● 热传导方程模型参数的求解

本文首先对热传导方程进行建立并进行初步分析，发现热传导方程属于抛物线型方程，边值条件复杂难以求得精确解，因而本文采用有限差分法，即将连续的定解区域用有限个离散点构成的网络进行替代，把定解区域上的连续函数用网络上选择的离散变量的函数进行近似替代，将原方程与定解条件中的微

商形式用差商形式近似替代，并通过迭代与最小二乘法对参数进行确定，并对电路板的温度进行预测。

● 定解区域的网格化剖析

设求解区域为 Ω ， $\Omega = \{(x, t) \mid 0 \leq x \leq L, 0 \leq t \leq T\}$ ，在空间维度上，将 $[0, L]$ 做 m 等分则有 $\Delta x = \frac{L}{m}$ ；在时间维度上将 $[0, T]$ 做 n 等分，则有 $\Delta t = \frac{T}{n}$ 。则可得到两组

平行直线簇 $\begin{cases} x = x_i, & 0 \leq i \leq m \\ t = t_j, & 0 \leq j \leq n \end{cases}$ 将 Ω 分割为矩形网格。如图 6

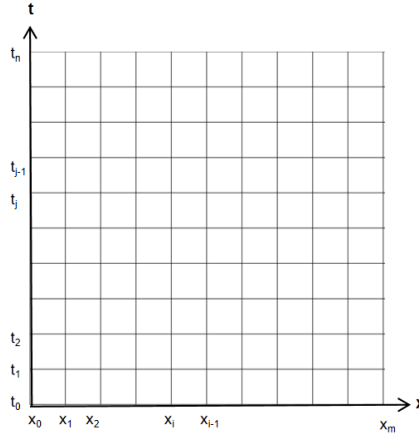


图 3 矩阵网格示意图

● 建立隐式差分格式

针对热传导方程中的微商形式进行离散化处理：

$$\left. \frac{\partial u}{\partial t} \right|_{(x_i, t_j)} = a^2 \left. \frac{\partial^2 u}{\partial x^2} \right|_{(x_i, t_j)} \quad (9)$$

其中等式的左半部分温度关于时间的微商形式本文采用欧拉折线法即：

$$u(x_i, t_{j-1}) = u(x_i, t_j) + \left. \frac{\partial u}{\partial t} \right|_{(x_i, t_j)} \Delta t \quad (10)$$

进行近似替代，其中 $f(x_n, t_n)$ 为热传导方程在 u_n 处对时间的微商，进而得到微商与差商之间的关系即：

$$\left. \frac{\partial u}{\partial t} \right|_{(x_i, t_j)} \approx \frac{u(x_i, t_j) - u(x_i, t_{j-1})}{\Delta t} \quad (11)$$

等式右边同样采用欧拉折线法进行近似替代，首先求得温度 u 在关于 x 在 $j-1$ 处的一阶偏导即：

$$\left. \frac{\partial u}{\partial x} \right|_{(x_i, t_{j-1})} = \frac{u(x_i, t_j) - u(x_i, t_{j-1})}{\Delta t} \quad (12)$$

其次求得关于 x 在 $j+1$ 处的一阶偏导即：

$$\left. \frac{\partial u}{\partial x} \right|_{(x_i, t_{j+1})} = \frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{\Delta t} \quad (13)$$

进而得到温度 u 关于 x 在 j 处的二阶偏导即：

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_{(x_i, t_j)} \approx \frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j)}{\Delta x^2} \quad (14)$$

进而对热传导方程的微分形式进行转换得到其差分形式：

$$\frac{u(x_i, t_j) - u(x_i, t_{j-1})}{\Delta t} \approx a^2 \frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j)}{\Delta x^2} \quad (15)$$

令其中 $r = a^2 \frac{\Delta t}{\Delta x^2}$ ，再对式子进行化简进一步得到离散化后的热传导方程：

$$-ru_{i-1}^j + (1 + 2r)u_i^j - ru_{i+1}^j = u_i^{j-1} \quad (16)$$

● 对于边界条件的离散化处理

对解微分方程进行定值求解需要引入定解条件，其中初值条件为常规数值条件，而边界条件含有微分形式若要求解则需对其进行离散化处理，本文已给出求解的边界条件为：

$$\begin{cases} \left. \frac{\partial u(x, t)}{\partial x} \right|_{x=0} = \sigma(u(t) - u_{\text{air}}(t)) \\ \left. \frac{\partial u(x, t)}{\partial x} \right|_{x=L} = -\sigma(u(t) - u_{\text{air}}(t)) \end{cases} \quad (17)$$

针对其中 $x=0$ 的边界条件进行同热传导方程相同的欧拉折现法变形转化为离散形式：

$$\frac{u_1^j - u_0^j}{\Delta x} = \sigma(u_0^j - u_{\text{air}}) \quad (18)$$

针对 $x=L$ 的边界条件进行相似的转化得到：

$$\frac{u_m^j - u_{m-1}^j}{\Delta x} = \sigma(u_{\text{air}} - u_m^j) \quad (19)$$

并对两式进行变形化简分别得到离散化的边界条件：

$$\begin{cases} (1 + \sigma \Delta x)u_0^j - u_1^j = u_{\text{air}} \sigma \Delta x \\ (1 + \sigma \Delta x)u_m^j - u_{m-1}^j = u_{\text{air}} \sigma \Delta x \end{cases} \quad (20)$$

● 建立隐格式对应方程组

分别对热传导方程与定解条件中的微商进行变化后建立隐格式方程对应如下：

$$AU^{(j)} = B^{(j-1)} \quad (21)$$

其中：

$$A = \begin{pmatrix} 1 + \sigma \Delta x & -1 & & & \\ -r & 1 + 2r & -r & & \\ & -r & 1 + 2r & -r & \\ & & \dots & \dots & \dots \\ & & & -r & 1 + 2r & -r \\ & & & & -1 & 1 + \sigma \Delta x \end{pmatrix}$$

$$U^{(j)} = \begin{pmatrix} u_0^j \\ u_1^j \\ u_2^j \\ \dots \\ u_{i-1}^j \\ u_i^j \\ u_{i+1}^j \\ \dots \\ u_{m-1}^j \\ u_m^j \end{pmatrix} \quad B^{(k-1)} = \begin{pmatrix} u_{air} \sigma \Delta x \\ u_1^{k-1} \\ u_2^{k-1} \\ \dots \\ u_{i-1}^{k-1} \\ u_i^{k-1} \\ u_{i+1}^{k-1} \\ \dots \\ u_{m-1}^{k-1} \\ u_{air} \sigma \Delta x \end{pmatrix} \quad (22)$$

5.3.2 模型求解：

● 求解最优参数流程图如图 4 所示：

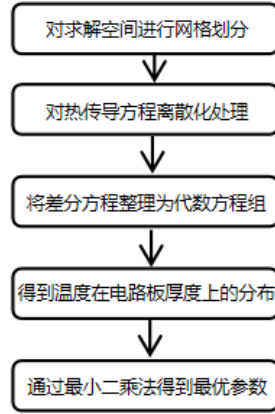


图 4 最优参数流程图

根据建立隐格式方程组通过 matlab 编程求解得到如下结果：

表 2 参数表

a_1	σ_1	a_2	σ_2
5.2000	3.2000	0.025	0.075

根据表中确定的最优参数并通过 matlab 编程求解得到 $u(x_i, t_j)$ 的函数曲线，同时求得不同时刻在电路板厚度上的温度分布如图 5 所示：

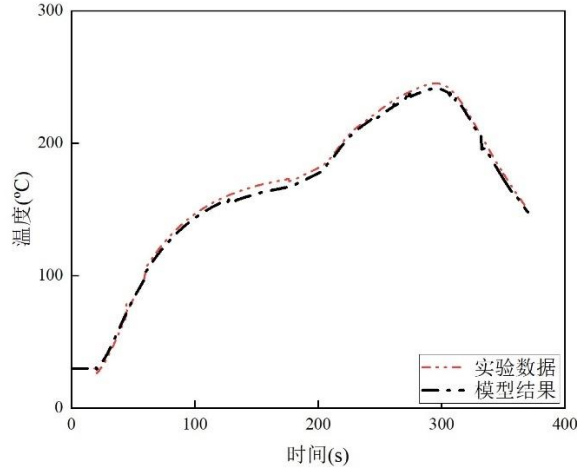


图 5 基于热传导模型结果

图 5 与实际值进行比较可较直观的看出其效果并不理想，由于所求参数为在其局部区域的最优解，但实际相互温区之间存在相互之间影响的联系，故上文参数实际是与时间 t 与 x 相关的函数，但若对其进行遍历求解所需要的时间过长，难以在较短时间对其求解且若调整则重新求解时间过长，较难在短时间内求得最优参数。

5.3.3 牛顿冷却定律模型建立与参数求解

● 冷却系数 k 的不确定证明

由于本文选用牛顿冷却定律含有微商形式：

$$\frac{du(t)}{dt} = -k(u(t) - u_{air}(t)) \quad (23)$$

因而先对其进行离散化处理得到：

$$\frac{T(t + \Delta t) - T(t)}{\Delta t} = -k(T(t) - T_{air}(t)) \quad (24)$$

并对其进行进一步的变形以得到 k 值与其相关确定：

$$k = -\frac{T(t + \Delta t) - T(t)}{\Delta t(T(t) - T_{air}(t))} \quad (25)$$

易知 k 是关于温度 T 的函数，而温度 T 是随时间 t 不断变化的函数，故 k 为随时间 t 不断变化的函数因而牛顿冷却定律应修正为：

$$\frac{du(t)}{dt} = -k(t) \cdot (u(t) - u_{air}(t)) \quad (26)$$

但 $k(t)$ 变化较小分段接近于常数，因而本文将 k 值分为：炉前区域、预热区（1~5 小温区）、恒温区（6~7 小温区）、冷却区（11~12 小温区）以及炉后区域，认为在相同区块中 k 值保持稳定，而在不同区块中 k 值有差异，并求得不同区块 k 参数的最优解。

● 冷却系数 k 的求解

冷却系数求解算法

- Step 1:** 将题目所给数据分段，并对每个分段分布求得最优参数 k 。
- Step 2:** 在每段中，设定 k 值，根据牛顿冷却定律的差分形式对电路板在此段的温度进行迭代，再通过与题目所给的实验数据比较并对 k 值进行修正。
- Step 3:** 通过最小二乘法确定最优参数 k 。

● 对牛顿冷却定律中系数 k 的结果求解

根据冷却系数 k 的求解方法并通过 matlab 编程求解得到各段 k 值如下表：

表 3 系数表

分段	炉前	1-5	6	7	8-9	10-炉后
k	0.0185	0.0177	0.0170	0.0247	0.0202	0.0290

再根据确定的冷却系数，与牛顿冷却定律的差分形式，通过 matlab 编程求解，可得到优化后的电路板炉温曲线，如图 6 所示：

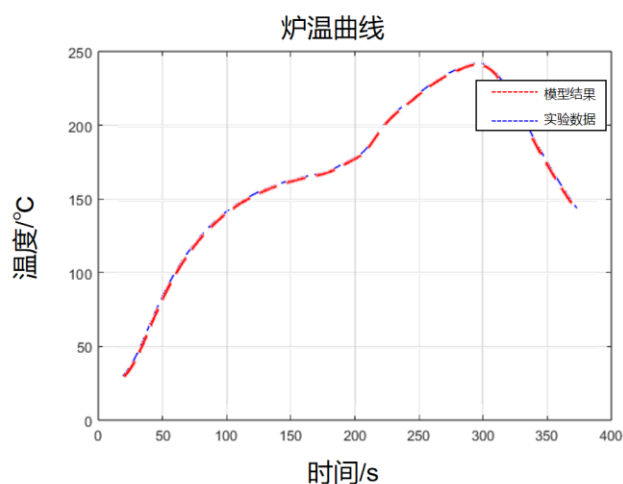


图 6 基于牛顿冷却定律模型结果

由图 10 可见，基于牛顿冷却定律模型优化后的热传导方程模型结果更加接近题目所给出的实验数据

3、改变各温区温度以及传送带速度后求解结果

改变各温区温度以及传送带速度后，小温区 3、6、7 中间与小温区 8 结束处焊接区域中心的温度如表所示：

表 4 各温区温度表

位置	小温区 3 中点	小温区 6 中点	小温区 7 中点	小温区 8 结束区
温度	129.7557℃	167.9916℃	189.6012℃	223.8698℃

炉温曲线如图 11 所示：

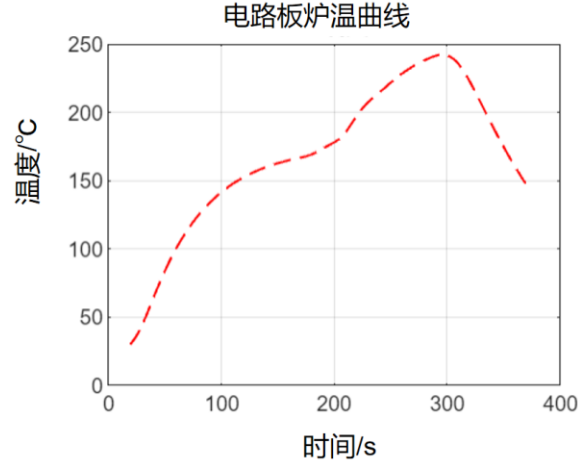


图 7 改变温区温度以及传送带速度后的炉温曲线

5.4 问题二的建模与求解

5.4.1 问题二模型建立

决策变量：传送带速度 v

目标函数：求最大传输速度 v 使得电路板在回焊炉中的炉温曲线满足制程界限，即

$$\max F = v \quad (27)$$

约束条件：传送带速度 v 必须确保电路板在回焊炉中的炉温曲线符合制程限制，即：

$$\left\{ \begin{array}{l} -3 \leq \frac{du}{dt} \leq 3 \\ 60 \leq t_{150 \leq u \leq 190} \leq 120 \\ 40 \leq t_{u \geq 270} \leq 90 \\ 240 \leq u_{\max} \leq 250 \\ 65 \leq v \leq 100 \\ \frac{du}{dt} = -k(u - u_{\text{air}}(t)) \end{array} \right. \quad (28)$$

其中， $t_{150 \leq u \leq 190}$ 是温度在 150 至 190°C 之间的时间； $t_{u \geq 270}$ 是温度超过 217°C 的时间； u_{\max} 是峰值温度； v 是过炉速度。

5.4.2 问题二模型求解

本题中决策变量较少，解空间较小，故直接遍历搜索即可得到最优解。求解步骤：

冷却系数求解算法

-
- Step 1:** 采用二分法求得最大速度所在的大致区间。
- Step 2:** 在 Step 1 中求得的大致区间中遍历搜索。
- Step 3:** 得到最大传输速度。
-

求解结果：

最大过炉速度为：

$$76.30\text{cm}/\text{min}$$

5.5 问题三的建模与求解

在焊接过程中，炉温曲线如图所示：

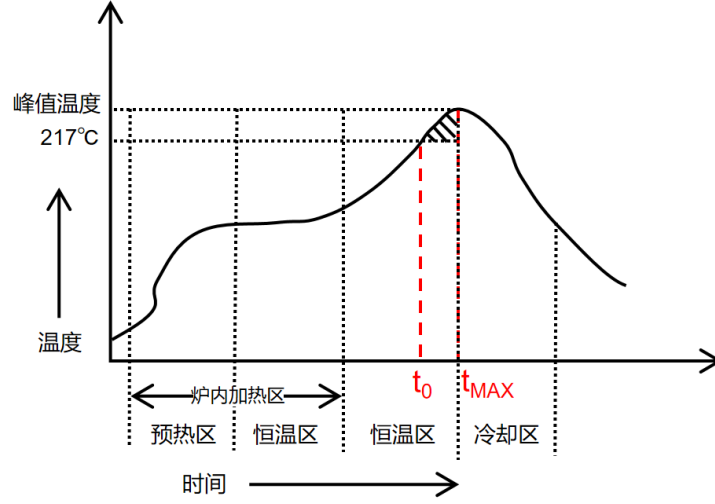


图 8 焊接过程的炉温曲线

图中阴影部分面积为：

$$S_1 = \int_{t_0}^{t_{\max}} (u(t) - 217) dt \quad (29)$$

上式中， t_0 为电路板第一次达到 217°C 所对应的时间， t_{\max} 为电路板达到峰值所对应的时间。

5.5.1 决策变量

传送带速度及各个温区的温度，即：

$$\{v, T_1, T_2, T_3, T_4\} \quad (30)$$

5.5.2 目标函数

速度温度应满足上下限，即：

$$\begin{cases} 65 \leq v \leq 100 \\ 165 \leq T_1 \leq 185 \\ 185 \leq T_2 \leq 205 \\ 225 \leq T_3 \leq 245 \\ 245 \leq T_4 \leq 265 \end{cases} \quad (31)$$

5.5.3 约束条件

炉温曲线应满足的制程界限，即：

$$\left\{ \begin{array}{l} -3 \leq u'(t) \leq 3 \\ 60 \leq t |_{150 \leq u \leq 190} \leq 120 \\ 40 \leq t |_{u \geq 270} \leq 90 \\ 240 \leq u_{\max} \leq 250 \\ \frac{du}{dt} = -k(u - u_{\text{air}}(t)) \end{array} \right. \quad (32)$$

综上，优化模型为：

$$\begin{aligned} \min S_1 &= \int_{t_0}^{t_{\max}} (u(t) - 217) dt \\ \left\{ \begin{array}{l} 65 \leq v \leq 100 \\ 165 \leq T_1 \leq 185 \\ 185 \leq T_2 \leq 205 \\ 225 \leq T_3 \leq 245 \\ 245 \leq T_4 \leq 265 \\ -3 \leq u'(t) \leq 3 \\ 60 \leq t |_{150 \leq u(t) \leq 190} \leq 120 \\ 40 \leq t |_{u(t) \geq 270} \leq 90 \\ 240 \leq u_{\max} \leq 250 \\ \frac{du}{dt} = -k(u - u_{\text{air}}) \end{array} \right. \end{aligned} \quad (33)$$

5.6 面积最小的最优炉温曲线求解：

本文中决策变量略多，且决策变量范围较大，故空间解较大，若采用经典定步长迭代法在优先时间内无法得到有效结果，因而选用遍布长迭代法对其实现求解：

5.6.1 变步长迭代法介绍

复化求积方法对提高积分精度是行之有效的，但必须事先给出恰当的步长 h 。如果步长太长，则难以保证精度，太小则增加计算量。在实际应用中，一般采用变步长积分法来解决该问题，即让步长不断减小，考察在不同步长条件下的计算结果的精度，一旦计算结果满足精度要求，则可以停止计算，给出正确答案。若决策变量较多，且决策变量范围较大则可采用逐次二分积分区间的方法得到不断减半的步长，再应用复化求积方法，可推导出某种递推公式，既可减少计算量，又可以较快地达到计算精度的要求。

5.6.2 利用变步长迭代法求解流程

首先建立满足制程界限条件下理想炉温曲线超过 217°C 到峰值温度所覆盖面积与划分四大温区与传送带速度之间的模型，并使用 `matlab` 软件进行求解在不同温度下，不同传送带速度的情况下面积的最优值

●面积最小的最优炉温曲线求解步骤

面积最小的最优炉温曲线求解

- Step1:** 首先定义一些所给定参数常量牛顿冷却方程系数 k ，热传导微分方程中的系数 σ 的具体数值等
- Step2:** 给出模型炉温曲线与各大温区温度 t_i 与传送带速度 v 之间的函数关系，并给出其各变量需满足的必要条件。
- Step3:** 建立一个用于计算理想炉温曲线超过 217°C 到峰值温度所覆盖面积的函数，引入符号计算库 `sympy`，构建面积与其他函数变量之间的联系。
- Step4:** 通过变步长迭代法，先粗略的计算最优取值所在的范围，再对其范围进一步进行迭代，从而起到缩短取得最优参数的迭代时间，并计算得到最优面积的情况下各大温区对应温度与传送带速度。

5.6.3 问题三求解展示

通过 `matlab` 编程，计算出的各温区温度 t_i (t_1, t_2, t_3, t_4) 的最优解部分结果如 3 所示。

表 3 各温区温度设定值

温区	1~5	6	7	8~9
温度	177.477°C	197.008°C	230.603°C	264.97°C

与传送带速度 v 的最优解：

过炉速度为 $v = 91.889\text{cm/min}$

炉温曲线超过 217°C 到峰值温度所覆盖面积最小为：

最小阴影面积为 $S_{1\min} = 447.983^\circ\text{C} \cdot \text{S}$

5.7 问题四的建模与求解

5.7.1 模型建立

设有两个函数 $f(x)$ 和 $g(x)$ ，函数 $f(x)$ 关于 x_0 对称，其对称函数为 $f(2x_0 - x)$ 。因此，函数 $g(x)$ 与 $f(x)$ 关于 x_0 对称的等价条件为：

$$\Delta S = \int_0^{x_0} [f(2x_0 - x) - g(x)] dx = 0 \quad (34)$$

由此可得，炉温曲线关于 $t=t_{\max}$ 对称的等价条件为：

$$S_2 = \int_{t_0}^{t_{\max}} [u(2t_{\max} - t) - u(t)] dt = 0 \quad (35)$$

其中， t_0 为电路板首次达到 217°C 时的时间， t_{\max} 为电路板达到峰值温度的时间。本题不仅要求炉温曲线在峰值温度 217°C 左右对称，还需要满足问题三中明胶部分的面积 S_1 最小，因此这是一个多目标优化问题。

对于多目标优化问题，通常将多个目标值标准化成指标，再进行加权平均，即：

$$S = \alpha_1 \frac{S_1}{S_{1,\max}} + \alpha_2 \frac{S_2}{S_{2,\max}} \quad (36)$$

其中, S_1 和 $S_{1,\max}$ 分别为 S_1 的实际值和最大值; α_1 和 α_2 为权重, 且满足 $\alpha_1 + \alpha_2 = 1$ 。

本模型采用灵活的动态余弦权函数。由于 S_1 面积明显较大, 先是缓慢增加, 中间有一个快速增长过程, 随后逐渐平缓增加直至最大值, 因此 α_1 可以设定为如下动态余弦函数:

$$\begin{cases} 0 & x \leq \alpha_1 \\ 1 - e^{0.1(x - \alpha_1)} & x > \alpha_1 \end{cases} \quad (37)$$

综上所述, 本题的多目标优化模型如下:

$$\min S = \alpha_1 \frac{S_1}{S_{1,\max}} + \alpha_2 \frac{S_2}{S_{2,\max}} \quad (38)$$

约束条件:

$$\begin{cases} 65 \leq v \leq 100 \\ 165 \leq T_1 \leq 185 \\ 185 \leq T_2 \leq 205 \\ 225 \leq T_3 \leq 245 \\ -3 \leq \frac{du}{dt} \leq 3 \\ 60 \leq t_{150 \leq u \leq 190} \leq 120 \\ 40 \leq t_{u \geq 270} \leq 90 \\ 240 \leq u_{\max} \leq 250 \\ \frac{du}{dt} = -k(u - u_{\text{air}}) \end{cases} \quad (39)$$

5.8 炉温曲线面积最小且曲线对称度最高的最优炉温曲线求解:

5.8.1 加权求和法介绍

本题任采用规划模型, 但规划模型数由单一炉温曲线面积最小变为同时考虑炉温曲线面积最小与对称度之间的多目标规划问题, 且两目标函数之间不存在明显的约束关系, 因而本文采用加权求和法将多目标规划问题转化为单目标问题进行求解, 对其进行动态调整。

5.8.2 利用权求和法与变步长迭代法求解流程

炉温曲线面积最小且曲线对称度最高的最优炉温曲线求解

Step1: 首先定义一些所给定参数常量牛顿冷却方程系数 k , 热传导微分方程中的系数 σ 的具体数值, 并通过其得到函数关系得到 $S_{1,\max}$ 与 $S_{2,\max}$ 的函数关系式

Step2: 将多目标规划函数加权转化为单目标规划函数, 即将 S_1 与 S_2 标准化, 再引入加权参数 ω_1 与 ω_2 得到单目标加权关系式并给出其各变量需满足的必要条件:

$$S = \omega_1 \frac{S_1}{S_{1,max}} + \omega_2 \frac{S_2}{S_{2,max}}$$

Step3: 以 $S = \omega_1 \frac{S_1}{S_{1,max}} + \omega_2 \frac{S_2}{S_{2,max}}$ 为目标函数，再用变步长迭代法，先粗略的计算最优取值所在的范围，再对其范围进一步进行迭代，从而起到缩短取得最优选参的迭代时间，并计算确定最优加权参数 ω_1 与 ω_2 。

Step4: 在确定最优加权参数 ω_1 与 ω_2 后，再次通过变步长迭代法，先粗略的计算最优取值所在的范围，再对其范围进一步进行迭代，从而起到缩短取得最优选参的迭代时间，并计算得到最优面积的情况下各大温区对应温度 t_i 与传送带速度 v 。

5.8.3 问题四求解展示

通过 matlab 编程，求得最优解展示如表所示：

表 5 传送带速度与各面积求解结果

传送带速度 v	$S_{1.min}$	$S_{2.min}$	S_{min}
88.782cm/min	449.17	35.872	0.3486

各大温区温度如 6 所示

表 6 各温区最优温度求解结果

温区	1~5	6	7	8~9
温度	169.733℃	186.657℃	231.844℃	264.999℃

六、模型评价

6.1 模型优点

6.1.1 热传导方程模型：使用热传导方程模型能够较好地反映出电路板在回焊炉中焊接的实际物理过程。该模型考虑了电路板在回焊炉中的温度分布，特别是在厚度方向上的分布，有助于对回焊过程进行更深入的研究和分析。

6.1.2 基于牛顿冷却定律模型：基于牛顿冷却定律的模型是在热传导方程的基础上简化条件后建立的。相比热传导方程，该模型不再考虑温度在电路板厚度上的分布，使得求解过程更加简单，效率更高。

6.2 模型缺点

6.2.1 热传导方程模型：使用热传导方程模型进行求解的过程较为复杂，难以在较短的时间内获得最优参数。特别是在实际应用中，可能会遇到求解时间过长的问题。且对于厚度较薄的电路板，温度在厚度方向上的分布接近一致，使用热传导方程模型实际效果未必显著。

6.2.2 基于牛顿冷却定律模型：基于牛顿冷却定律的模型未考虑到电路板在回焊炉中温度在厚度方向上的分布，对于较厚的电路板，模型可能过于简化，不能准确反映实际的温度变化。对于需要精确温度控制的应用场景，该模型的精度可能不足。

6.3 模型推广与改进








在本文的研究中，对于温度在两边温度不等的间隙中的分布，假设温度随空间位置线性变化。然而事实在间隙与温区边界上，温度连续但不可导。为了改进这一点，需对温区与间隙边界上的温度分布进行了修正，使得温度变化符合要求。这种改进有助于更加准确地模拟温度在间隙和温区边界上的变化情况，提高模型

的整体精度。此外，这种修正还能够为后续的优化和控制策略提供更加可靠的数据支持，从而进一步提升回焊过程的质量控制水平。

七、参考文献

[1]汪学军.多温区自动测控系统控制模型的建立与研究[D].中南大学,2007
[2]雷翔霄,唐春霞,徐立娟.基于 RBF-PID 的热风回流焊温度控制[J].邵阳学院学报(自然科学版),2020,17(04):31-38.
[3]杨晓生.多温区无铅回流焊炉控制系统的设计与实现[D].国防科学技术大学,2009.

八、附录

支撑材料目录				
名称	修改日期	类型	大小	
 problem1	2024/7/16 22:17	MATLAB Live Scr...	4 KB	
 problem2	2024/7/16 22:18	MATLAB Live Scr...	4 KB	
 problem3	2024/7/16 22:19	MATLAB Live Scr...	4 KB	
 problem4	2024/7/16 22:24	MATLAB Live Scr...	4 KB	
 result	2024/7/16 22:31	XLS 工作表	10 KB	
 图1	2024/7/16 21:35	PPTX 演示文稿	58 KB	
 图2	2024/7/14 21:42	JPG 图片文件	51 KB	
 图3, 图4	2024/7/16 21:33	PPTX 演示文稿	66 KB	
 图8	2024/7/16 21:34	PPTX 演示文稿	57 KB	

第一问代码

```
clc;
clear;
T15 = 173; % 1-5 区间温度
T6 = 198; % 6 区间温度
T7 = 230; % 7 区间温度
T89 = 257; % 8-9 区间温度
T1011 = 25; % 10-11 区间温度
load('C:\Users\HuangShaoze\Desktop\2020A');
dt = 0.5;
v = 78/60; % 传送带速度，单位为 m/s
t = 0:dt:435.5/v; % 根据传送带速度生成时间向量
Tair = T_air(t, v, T15, T6, T7, T89, T1011); % 空气温度
T = zeros(length(Tair), 1); % 初始化电路板温度向量
T(1) = 25; % 初始电路板温度
```

```

for i = 2:length(Tair)
    k = kk(v * (i - 1) * dt); % 假设 kk 是位置或时间的函数
    T(i) = -dt * (T(i - 1) - Tair(i)) * k + T(i - 1); % 数值积分（欧拉方法）
end
plot(t, T, '-r', 'linewidth', 1.6, 'markeredgecolor', 'k', 'markersize', 8);
xlabel('时间/s', 'Fontname', '宋体', 'fontweight', 'bold', 'fontsize', 12);
ylabel('温度/℃', 'Fontname', 'Times New Roman', 'fontweight', 'bold', 'fontsize', 12);
title('电路板炉温曲线', 'Fontname', '宋体', 'fontweight', 'bold', 'FontSize', 12);

```

第二问函数

```

function [tspan, u] = problem2(v, T15, T6, T7, T89, T1011)
    tspan = 0:0.1:100;
    u = T15 * exp(-v * tspan);
end

```

第二问代码

```

clc;
clear;
% 小温区温度
T15 = 182; % 1-5 区间温度
T6 = 203; % 6 区间温度
T7 = 237; % 7 区间温度
T89 = 254; % 8-9 区间温度
T1011 = 25; % 10-11 区间温度
V = [];
for v = 65/60:0.0001:100/60
    [tspan, u] = problem2(v, T15, T6, T7, T89, T1011);
    dt = tspan(2) - tspan(1);
    % 计算大于 150, 小于 190 的时间
    a = find(u < 190 & u > 150);
    t1 = max((length(a) - 2), 1) * dt;
    % 计算大于 217 的时间
    b = find(u > 217);
    t2 = max((length(b) - 1), 1) * dt;
    % 计算峰值温度
    Tmax = max(u);
    % 计算各一阶导数最大最小值

```

```

    uu = (u(2:end) - u(1:end-1)) / dt;
    uumin = min(uu);
    uumax = max(uu);
    % 判断条件
    if (uumin > -3) && (uumax < 3) && (t1 > 60) && (t1 < 120) && (t2 >
40) && (t2 < 90) && (Tmax > 240) && (Tmax < 250)
        V = [V; v];
    end
end
vmax = max(V);
vmax_minutes = vmax * 60; % 转换为分钟
[tspan1, u1] = problem2(vmax, T15, T6, T7, T89, T1011);
plot(tspan1, u1, '--r', 'linewidth', 1.6, 'markeredgecolor', 'k',
'markerfacecolor', 'b', 'markersize', 8);
xlabel('时间/s', 'Fontname', '宋体', 'fontweight', 'bold', 'fontsize',
12);
ylabel('温度/℃', 'Fontname', 'Times New Roman', 'fontweight', 'bold',
'fontsize', 12);
title('最大过炉速度时的电路板炉温曲线', 'Fontname', '宋体', 'fontweight',
'bold', 'FontSize', 12);

```

第三问代码

```

clear;
clc;
temperature = 10000;
cooling_rate = 0.94; % 冷却速率
previous_x = [90.4998096731545, 175.474315263950, 197.983991611696,
228.256966424909, 264.881809804930];
% 初始化前一解
[previous_tspan, previous_u] = problem2(previous_x(1)/60,
previous_x(2), previous_x(3), previous_x(4), previous_x(5), 25);
dt = previous_tspan(2) - previous_tspan(1);
previous_area = S(previous_u, dt); % 计算初始面积
i = 1;
pl(i) = previous_area; % 存储初始面积
temperature_iterations = 1;
while temperature > 2
    % 随机生成一个邻近解
    current_x = perturb(previous_x);
    [current_tspan, current_u] = problem2(current_x(1)/60, current_x(2),
current_x(3), current_x(4), current_x(5), 25);
    dt = current_tspan(2) - current_tspan(1);
    current_area = S(current_u, dt); % 计算当前解的面积

```

```

% 计算面积变化
diff = current_area - previous_area;

% Metropolis 算法
if diff < 0 || rand < exp(-diff / temperature)
    previous_x = current_x; % 接受新解
    previous_area = current_area;
    i = i + 1;
    pl(i) = previous_area; % 存储接受的面积
    temperature_iterations = temperature_iterations + 1;
end
% 每 100 次迭代降低温度
if temperature_iterations >= 100
    temperature = cooling_rate * temperature;
    temperature_iterations = 0;
end
end
% 绘制最终解
[tspan, uuu] = problem2(previous_x(1)/60, previous_x(2), previous_x(3),
previous_x(4), previous_x(5), 25);
figure(1)
plot(tspan, uuu, '-r', 'linewidth', 1.6, 'markeredgecolor', 'k',
'markerfacecolor', 'b', 'markersize', 8)
xlabel('时间/s', 'Fontname', '宋体', 'fontweight', 'bold', 'fontsize',
12)
ylabel('温度/℃', 'Fontname', '宋体', 'fontweight', 'bold', 'fontsize',
12)
title('最优炉温曲线', 'Fontname', '宋体', 'fontweight', 'bold',
'FontSize', 12)
hold on
plot(tspan, 217 * ones(length(tspan), 1), '--b', 'linewidth', 2,
'markeredgecolor', 'k', 'markerfacecolor', 'b', 'markersize', 8)
hold off
figure(2)
plot(pl, '-r', 'linewidth', 1.2)
xlabel('接受次数', 'Fontname', '宋体', 'fontweight', 'bold', 'fontsize',
12)
ylabel('内能', 'Fontname', '宋体', 'fontweight', 'bold', 'fontsize', 12)
title('内能变化曲线', 'Fontname', '宋体', 'fontweight', 'bold',
'FontSize', 12)

```

第四问代码

```

clc;
clear;
temperature = 1000; % 初始化温度
cooling_rate = 0.94; % 冷却速率
previous_x = [88.7818206515348, 169.733087638798, 186.657314640567,
231.844263600990, 264.999044246197];
% 目标函数，路线的总距离
[previous_tspan, previous_u] = problem2(previous_x(1)/60,
previous_x(2), previous_x(3), previous_x(4), previous_x(5), 25);
dt = previous_tspan(2) - previous_tspan(1);
previous_area = SDDD(previous_u, dt);
i = 1;
pl(i) = previous_area;
temperature_iterations = 1; % 用于控制每 100 次迭代后降温的标志
while 0.000080 < temperature
    % 随机生成相邻解
    current_x = perturb(previous_x);
    [current_tspan, current_u] = problem2(current_x(1)/60, current_x(2),
current_x(3), current_x(4), current_x(5), 25);
    dt = current_tspan(2) - current_tspan(1);
    current_area = SDDD(current_u, dt);
    % 计算面积变化
    diff = current_area - previous_area;
    % Metropolis 算法
    if diff < 0 || rand < exp(-diff / temperature)
        previous_x = current_x;
        % 接受新路线
        previous_area = current_area;
        i = i + 1;
        pl(i) = previous_area;
        % 更新迭代次数
        temperature_iterations = temperature_iterations + 1;
    end
    % 每 100 次迭代降温
    if temperature_iterations >= 100
        temperature = cooling_rate * temperature;
        temperature_iterations = 0;
    end
end
% 每 200 次迭代绘制当前路线
% 绘制最终解
[tspan, uuu] = problem2(previous_x(1)/60, previous_x(2), previous_x(3),
previous_x(4), previous_x(5), 25);
figure(1)

```

```

plot(tspan, uuu, '--r', 'linewidth', 1.6, 'markeredgecolor', 'k',
'markeredgecolor', 'b', 'markersize', 8)
xlabel('时间/s', 'Fontname', '宋体', 'fontweight', 'bold', 'fontsize',
12)
ylabel('温度/℃', 'Fontname', '宋体', 'fontweight', 'bold', 'fontsize',
12)
title('最优炉温曲线', 'Fontname', '宋体', 'fontweight', 'bold',
'FontSize', 12)
hold on
plot(tspan, 217 * ones(length(tspan), 1), '--b', 'linewidth', 2,
'markeredgecolor', 'k', 'markeredgecolor', 'b', 'markersize', 8);
figure(2)
p11 = [p1(floor(length(p1)/5*3):length(p1)),
p1(floor(length(p1)*19.5/20):length(p1)),
p1(floor(length(p1)*19.5/20):length(p1)),
p1(floor(length(p1)*19.5/20):length(p1))];
plot(p11, '-T', 'linewidth', 0.1)
xlabel('接受次数', 'Fontname', '宋体', 'fontweight', 'bold', 'fontsize',
12)
ylabel('内能', 'Fontname', '宋体', 'fontweight', 'bold', 'fontsize', 12)
title('内能变化曲线', 'Fontname', '宋体', 'fontweight', 'bold',
'FontSize', 12)
% ylim([7700, 14000])
dt = tspan(2) - tspan(1);
sff = SF(uuu, dt);
sss = S(uuu, dt);
sddd = previous_area;

```