

## 炉温曲线的机理建模与优化设计

### 摘 要

本文主要研究回焊炉各温区温度和传送带速度对炉温曲线的影响,建立了基于热传导方程与牛顿冷却定律的温度分布模型。

对问题一,首先研究了炉前、炉后与间隙的温度分布,建立了电路板温度所满足的热传导方程和边界条件,又将其简化为基于牛顿冷却定律的温度分布模型,后者较前者在相同的时间内得到了更优的拟合效果(其均方误差 RMSE 为  $2.81^{\circ}\text{C}$ , 平均绝对误差为  $1.73^{\circ}\text{C}$ ),故选择基于牛顿冷却定律的温度分布模型进行识别模型参数;其次,根据实验数据可以分析得到不同温区的模型参数是动态变化的,我们在四大温区中分别设定不同的冷却系数,段内相同、段间不同,并通过最小二乘法确定了上述参数;最后,利用有限差分法对温度分布模型进行了求解,得到小温区 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:

位置	小温区 3 中点	小温区 6 中点	小温区 7 中点	小温区 8 结束处
温度	129.7557 $^{\circ}\text{C}$	167.9916 $^{\circ}\text{C}$	189.6012 $^{\circ}\text{C}$	223.8698 $^{\circ}\text{C}$

对于问题二,建立了以最大过炉速度为目标的单目标优化模型,约束条件为满足问题一所建立的微分方程模型和工艺所要求的制程界限,并借助二分法进行求解得到最大传输速度为 76.2980 cm/min。

对于问题三,以衡量累计高温区域大小所对应的阴影面积最小为目标,建立了以各温区问题和过炉速度为决策变量的优化模型,其约束条件包括制程界限、温度分布所满足的微分方程模型,采用了模拟退火算法,求得最小阴影面积为  $447.983^{\circ}\text{C}\cdot\text{s}$ ,此时过炉速度为 91.889cm/min,各温区温度如下表所示:

温区	1~5	6	7	8~9
温度	177.477 $^{\circ}\text{C}$	197.008 $^{\circ}\text{C}$	230.603 $^{\circ}\text{C}$	264.967 $^{\circ}\text{C}$

对于问题四,首先给出了高温区炉温曲线对称性指标——两函数之差的范数;其次,建立了以高温区炉温曲线对称性和高温累计区域面积为目标的双目标规划模型,将两目标规范化并采用动态综合加权法将多目标优化转变为单目标优化;最后,通过模拟退火算法求得最小阴影面积为  $449.17^{\circ}\text{C}\cdot\text{s}$ ,最小范数为  $55.87^{\circ}\text{C}\cdot\text{s}$ ,此时传送带过炉速度 88.78cm/min,各温区温度设定如下表:

温区	1~5	6	7	8~9
温度	169.733 $^{\circ}\text{C}$	186.657 $^{\circ}\text{C}$	231.844 $^{\circ}\text{C}$	264.999 $^{\circ}\text{C}$

最后给出了建模过程的误差分析和结果检验,本文结论正确合理,具有一定的应用价值和普适性。

关键词:热传导方程;牛顿冷却定律;有限元差分法;最小二乘法;优化模型;模拟退火算法;动态综合加权法

## 1.问题重述

### 1.1 问题的背景

在过去的几十年里,电子产品行业取得了飞速的发展。随着电子产品的竞争日趋加剧,生产的不确定性不断加大,为此对电子设备的贴装、焊接提出了更高的要求。在集成电路板等电子产品生产中,需要将安装有各种电子元件的印刷电路板放置在回焊炉中,通过加热,将电子元件自动焊接到电路板上。在这个生产过程中,让回焊炉的各部分保持工艺要求的温度,对产品质量至关重要。回流焊温度控制技术和焊接工艺是回流焊最关键的部分<sup>[1]</sup>。

### 1.2 问题的重述

问题一:根据焊接区域的温度变化规律建立数学模型,在建立的数学模型基础上,给定传送带过炉速度和各温区温度的情况下,给出焊接区域中心的温度变化情况,列出小温区 3、6、7 中点及小温区 8 结束处焊接区域中心的温度,画出相应的炉温曲线。

问题二:在给定的各温区温度情况下,同时考虑到制程界限约束,确定最大的传送带过炉速度。

问题三:为了最优化炉温曲线,要求超过 217°C 到峰值温度所覆盖的面积最小,确定各温区设定温度和传送带过炉速度,并给出覆盖的面积。

问题四:结合问题三,要求以峰值温度为中心线的两侧超过 217°C 的炉温曲线应尽量对称以完善最优炉温曲线。给出炉温曲线,以及各温区设定的温度及传送带过炉速度,并给出相应的指标值。

## 2.问题分析

问题一:根据温度变化规律建立数学模型,目前,有关焊接区域的温度变化规律的研究大多是基于数据驱动的控制和调整,文献[2-3]是基于 PID 控制原理控制回焊炉中的温度,而缺乏相关机理模型,不利于了解到电路板在回焊炉内焊接的物理过程。所以本题难点在于需要建立机理模型,模型不仅要能够反映出电路板焊接的物理过程,同时又要与题目中所给的焊接区域温度变化曲线吻合。针对此问题,拟在热传导方程以及牛顿冷却定律的基础上建立数学模型。

问题二:在第一问已求得参数的基础上,又给定了各温区温度,要求确定电路板炉温曲线满足题目所要求的制程界限时电路板的最大过炉速度。本题可用先通过二分法求得最大速度大致区间,再在大致区间中遍历搜索,最后得到最大传输速度。

问题三: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。

问题四:在问题三的基础上,确定各温区的温度以及传送带过炉速度,使

得以峰值温度为中心线的两侧超过 217℃的炉温曲线应尽量对称，同时还要使得理想的炉温曲线超过到峰值所覆盖的面积最小。本题难点在于如何给定对称性指标，且该问题属于多目标优化问题，可通过动态综合加权法，将多目标函数转化为单目标函数。

### 3.模型假设

- (1) 不考虑空气对流。
- (2) 假设电路板材料均匀，各项同性。
- (3) 当小温区之间的温度差较小时，间隙温度在两边温度不等的间隙中随位移线性变化且不考虑小温区边界附近的温度变化。
- (4) 当小温区之间的温度较大时，考虑小温区温度之间的影响。
- (5) 炉前以及炉后区域中环境温度随空间距离平缓变化最终趋近于室温。

### 4.符号说明

符号	含义	单位
$u(x,t)$	电路板在 $t$ 时刻 $x$ 厚度上的温度	℃
$u_{av}$	电路板进入回焊炉中外界温度	℃
$k$	牛顿冷却方程中的冷却系数	$W/m^2 \cdot K$
$t _{150 \leq u \leq 190}$	温度上升过程中在 150~190℃ 的时间	$s$
$40 \leq t_{\text{total}} \leq 90$	温度大于 217℃ 的时间	$s$
$v$	传送带过炉速度	$cm/min$
$u_{\text{max}}$	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	℃

## 5.模型的建立和求解

### 5.1 建模前的准备

#### 5.1.1 实际问题的数字化处理

如图1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。

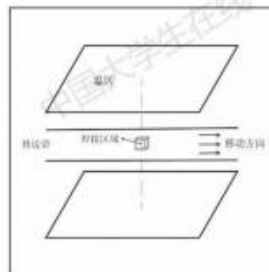


图1 电路板在回焊炉传送示意图

#### 5.1.2 环境温度分布

##### (1) 间隙温度分布

##### 1) 两边温度不等的间隙

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:

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

故可得:

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

进而

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

由式(4)可知两个小温区之间的温度随空间距离成线性变化

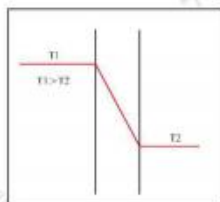


图2 间隙两边温度不等

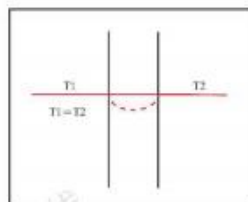


图3 间隙两边温度相等

## 2) 两边温度相等的间隙

两边恒热源不断向间隙内供热,直到间隙内温度处处与两端温度相等,如图3所示。

### (2) 其他区域温度分布

根据假设(4),小温区9与小温区10温度相差较大,故考虑小温区9对小温区10的影响,本文对温区间的影响做简化处理,认为在小温区10中环境温度随空间距离线性变化,即

$$u(x, t) = k_1 x + b_1 \quad (5)$$

又根据假设(5),炉前炉后区域随空间距离平缓变化最终趋于室温,故可设炉前炉后温度的变化为

$$u(x, t) = ce^{-\lambda x} + h \quad (6)$$

其中,  $\lambda > 0$ ,  $h$  为室温。

### (3) 总体温度分布

根据以上分析,我们可得大致的温度分布,如图4所示

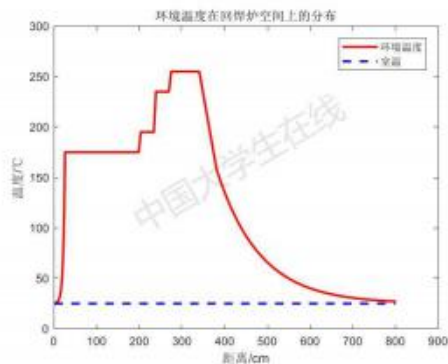


图4 环境温度随空间距离分布情况

由图可见,回焊炉两端的环境温度与空间距离呈指数变化,最后趋近于室温,部分区域根据假设将温度线性化处理。

### 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:

$$\frac{du}{dt} = -k(u - u_{av}(t)) \quad (7)$$

式中,  $u$  为系统的温度,  $u_{av}(t)$  为系统外界温度,  $k$  为冷却系数。

## 5.2 问题一的建模与求解

### 5.2.1 一维热传导方程的建立

在物体中任取一块闭曲面区域  $S$ , 设它所包含的任一区域为  $V$ , 如图 5 所示。假设在  $t$  时刻, 区域  $V$  内任一点  $M(x, y, z)$  处的温度为  $u(x, y, z, t)$ 。

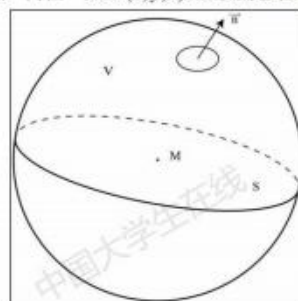


图 5 微元法示意图

根据 Fourier 实验的结果可知: 单位时间  $dt$  内, 流过一块无穷小区域  $ds$  的热量  $dQ$  正比例于时间  $dt$ , 曲面区域的面积  $dS$  和区域内的温度沿曲面法线分量

的方向导数  $\frac{\partial u}{\partial n}$ , 即

$$dQ = -k_1 \frac{\partial u}{\partial n} dt dS = -k_1 \text{grad } u \cdot d\vec{S} dt \quad (8)$$

其中,  $k_1 = k_1(x, y, z)$  是整个系统的导热率, 当物体内部介质均匀分布且各向同性时为常数。

由上述公式 (6) 我们可以得到: 当时间从  $t_1$  变化到  $t_2$  时, 曲面  $S$  内流向区域  $V$  的热量  $Q_1$  为

$$Q_1 = \int_{t_1}^{t_2} \left( \iint_S k_1 \text{grad } u \cdot d\vec{S} \right) dt \quad (9)$$



流入  $V$  的热量将带动这块区域的温度上升，由此可以得到温度在时间间隔  $t_1 \sim t_2$  之间从  $u(x, y, z, t_1)$  变化到  $u(x, y, z, t_2)$  所需要的热量为

$$Q_2 = \iiint_V c\rho[u(x, y, z, t_2) - u(x, y, z, t_1)]dV \quad (10)$$

其中， $c$  是物体的比热容； $\rho$  是物体的密度。

根据能量守恒定律， $Q_1$  等于  $Q_2$ ，即

$$\int_{t_1}^{t_2} \left( \iint_S k_i \text{grad } u \cdot d\vec{S} \right) dt = \iiint_V c\rho[u(x, y, z, t_2) - u(x, y, z, t_1)]dV \quad (11)$$

利用 Gauss 公式将上式左端化为三重积分，经过化简可得：

$$\int_{t_1}^{t_2} \left( \iiint_V k_i \nabla^2 u dV \right) dt = \int_{t_1}^{t_2} \left( \iiint_V c\rho \frac{\partial u}{\partial t} dV \right) dt \quad (12)$$

因被积函数连续，上式的被积函数相等即可满足等式恒成立，如下式所示

$$\frac{\partial u}{\partial t} = a^2 \nabla^2 u = a^2 \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \quad (13)$$

其中  $a^2 = \frac{k_i}{c\rho}$  为已知常量。

方程 (13) 称为三维热传导方程。本题已简化为一维传热问题，可列出方程如下：

$$\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2} \quad (14)$$

### 5.2.2 基于热传导方程的温度分布模型

电路板在回焊炉内焊接过程满足一维热传导方程即：

$$\frac{\partial u(x, t)}{\partial t} = a^2 \frac{\partial^2 u(x, t)}{\partial x^2} \quad (15)$$

电路板在进入回焊炉之前整体温度为  $25^\circ\text{C}$ ，即满足初始条件：

$$u(x, 0) = 25 \quad (16)$$

根据牛顿冷却定律可知，电路板两端需满足边界条件：

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

### 5.2.3 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  $0.15\text{mm}$ ，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（在

5.24 中可证明这一点)。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:

$$\frac{du(t)}{dt} = -k(u(t) - u_{av}(t)) \quad (18)$$

其中,  $u(t)$  为  $t$  时刻系统的温度,  $u_{av}(t)$  为  $t$  时刻系统外界温度,  $k$  为冷却系数。

#### 5.2.4 问题一模型求解

由于热传导方程模型和牛顿冷却定律模型中都存在未知参数,故需要先确定最优参数,再进行求解。

##### (1) 热传导方程模型参数求解

##### ① 求解热传导方程模型中最优参数方法

由于建立的热传导方程属于抛物型方程,边值条件复杂难以求得解析解。采用有限差分法,将连续的定解区域用有限个离散点构成的网络来代替,把定解区域上的连续函数用网格上定义的离散变量的函数来近似,把原方程和定解条件中的微商用差商来近似。通过迭代求得参数下的电路板温度,最后通过最小二乘法确定最优参数。

求解最优参数步骤为:

**Step 1:** 对求解区域进行网格剖析。设求解区域为  $\Omega$ ,

$\Omega = \{(x, t) | 0 \leq x \leq L, 0 \leq t \leq T\}$ , 在空间维度上,将  $[0, L]$  做  $m$  等份,  $\Delta x = \frac{L}{m}$ ; 在时

间维度上,将  $[0, T]$  做  $n$  等份,  $\Delta t = \frac{T}{n}$ 。将两组平行直线簇  $\begin{cases} x = x_i, & 0 \leq i \leq m \\ t = t_j, & 0 \leq j \leq n \end{cases}$  将

$\Omega$  分割成矩形网络,如图 6 所示。

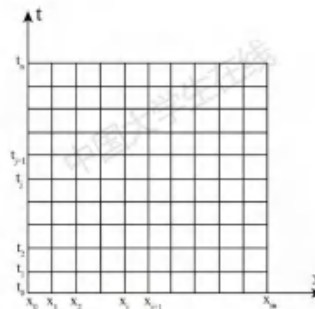


图 6 矩阵网络示意图



**Step 2:** 建立隐式差分格式，分为以下两步：

1) 将热传导方程离散化处理

$$\left. \frac{\partial u}{\partial t} \right|_{(x_i, t_j)} = a^2 \left. \frac{\partial^2 u}{\partial x^2} \right|_{(x_i, t_j)} \quad (19)$$

$$\frac{u(x_i, t_j) - u(x_i, t_{j-1})}{\Delta t} \approx a^2 \frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j)}{\Delta x^2} \quad (20)$$

故：

$$\frac{u_i^j - u_i^{j-1}}{\Delta t} = a^2 \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{\Delta x^2} \quad (21)$$

令  $r = a^2 \frac{\Delta t}{\Delta x^2}$ ，则

$$-ru_{i-1}^j + (1+2r)u_i^j - ru_{i+1}^j = u_i^{j-1} \quad (22)$$

2) 边界条件离散

$$\left. \frac{\partial u}{\partial x} \right|_{x=0} = \sigma(u - u_{av}) \quad (23)$$

$$\frac{u_1^j - u_0^j}{\Delta x} = \sigma(u_0^j - u_{av}^j) \quad (24)$$

故：

$$(1 + \sigma \Delta x)u_0^j - u_1^j = u_{av}^j \sigma \Delta x \quad (25)$$

$$\left. \frac{\partial u}{\partial x} \right|_{x=L} = \sigma(u_{av} - u) \quad (26)$$

$$\frac{u_n^j - u_{n-1}^j}{\Delta x} = \sigma(u_{av}^j - u_n^j) \quad (27)$$

故：

$$(1 + \sigma \Delta x)u_n^j - u_{n-1}^j = u_{av}^j \sigma \Delta x \quad (27)$$

**Step 3:** 建立隐格式对应的方程组如下

$$AU^{(j)} = B^{(j-1)} \quad (28)$$

其中：

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

$$U^{(j)} = \begin{pmatrix} u_0^j \\ u_1^j \\ u_2^j \\ \dots \\ u_{i-1}^j \\ u_i^j \\ u_{i+1}^j \\ \dots \\ u_{m-1}^j \\ u_m^j \end{pmatrix} \quad B^{(k-1)} = \begin{pmatrix} u_{m0}\sigma\Delta x \\ u_1^{k-1} \\ u_2^{k-1} \\ \dots \\ u_{i-1}^{k-1} \\ u_i^{k-1} \\ u_{i+1}^{k-1} \\ \dots \\ u_{m-1}^{k-1} \\ u_{m0}\sigma\Delta x \end{pmatrix}$$

Step 4: 利用最小二乘法求得参数  $a, \sigma$ 。

求解最优参数流程如图 7 所示:

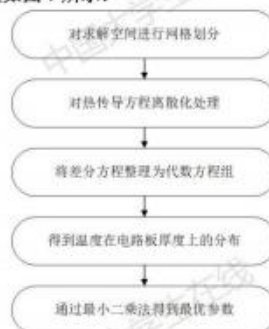


图 7 热传导方程求解流程图

## ②热传导方程模型最优参数求解结果

根据①中的求解方法,通过 *matlab* 编程求解,可得到结果如下表所示:

表 1 热传导方程最优参数

$a_1$	$\sigma_1$	$a_2$	$\sigma_2$
5.2000	3.2000	0.0250	0.0075

根据表中确定的最优参数，通过 *matlab* 编程求解，求得电路板的炉温曲线如图 8 所示，同时可以求得某个时刻温度在电路板厚度上的分布，如图 9 所示：

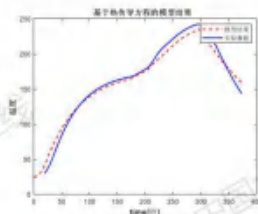


图 8 基于热传导模型结果

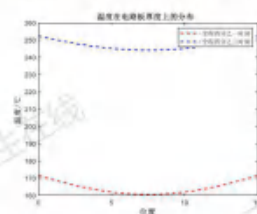


图 9 温度在电路板厚度上的分布

图 8 所示效果不理想，该参数为仅为在有限时间内求得的局部最优解。由于热传导方程模型求解程序较为复杂，需要的时间过长（32000 次遍历大约需要 83 分钟），故在较短的时间内较难求得最优参数。

由图 9 可见，电路板进入回焊炉中，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接近一致，故可将电路板视为一个整体与外界交换热量，因此电路板在回焊炉中温度变化满足牛顿冷却定律模型。接下来对牛顿冷却模型进行求解。

## (2) 牛顿冷却定律模型参数求解

### ① 冷却系数 $k$ 的不定性说明

将式 (13) 离散化处理得：

$$\frac{T(t+\Delta t)-T(t)}{\Delta t} = -k(T(t)-T_{av}(t)) \quad (29)$$

从而可得

$$k = -\frac{T(t+\Delta t)-T(t)}{\Delta t(T(t)-T_{av}(t))} \quad (30)$$

根据题目所给实验数据可以看出， $k$  值随时间不断变化，故  $k$  为时间  $t$  的函数。

故方程 (29) 应修正为

$$\frac{du(t)}{dt} = -k(t) \cdot (u(t) - u_{av}(t)) \quad (31)$$

但  $k(t)$  又分段接近于常数，故本文认为  $k$  值在炉前区域、预热区（1~5 小温区）、恒温区（6~7 温区）、回流区（8~9 温区）、冷却区（11~12 温区）以及炉后区域中相同，而不同段  $k$  值有差异，故分别求出各段最优参数  $k$ 。

### ② 冷却系数 $k$ 的确定方法

确定冷却系数  $k$  的步骤如下：

**Step 1:** 将题目所给数据分段，每段分别求得最优参数  $k$ 。

**Step 2:** 在每段中，设定一个  $k$  值，根据式 (27) 差分形式，迭代求得电路板在此段的温度，再通过与题目所给的实验数据比较来修正  $k$  值。

**Step 3:** 最终通过最小二乘法得到每段最优参数  $k$ 。

### ③ 冷却系数 $k$ 的求解结果

根据冷却系数的确定方法，通过 *matlab* 编程求得各段  $k$  值如下表所示：

表 2 各段冷却系数  $k$  值

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

再根据表中确定的冷却系数，以及式 (27) 牛顿冷却定律的差分形式，通过 *matlab* 编程求解，可求得电路板炉温曲线，如图 10 所示：

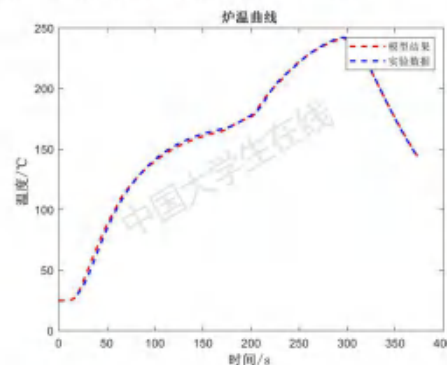


图 10 基于牛顿冷却定律模型的炉温曲线

由图 10 可见，使用基于牛顿冷却定律的模型效果优于基于热传导方程的模型，模型结果更接近题目所给出的实验数据。

### (3) 改变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 中点及小温区 8 结束处焊接区域中心的温度如表所示：

表 3 电路板在回焊炉中部分位置焊接区域的中心温度

位置	小温区 3 中点	小温区 6 中点	小温区 7 中点	小温区 8 结束处
温度	129.7557°C	167.9916°C	189.6012°C	223.8698°C

电路板炉温曲线如图 11 所示：

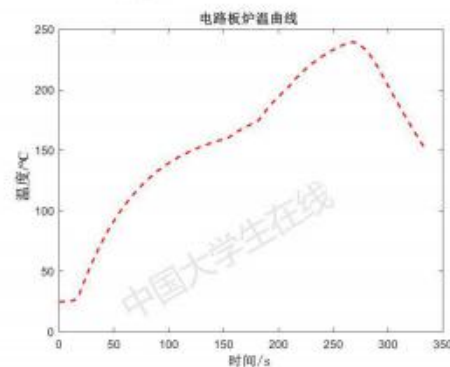


图 11 改变温区温度以及传送带速度后的炉温曲线

### 5.3 问题二的建模与求解

#### 5.3.1 问题二模型建立

决策变量：传送带速度  $v$

目标函数：目标为求使得电路板在回焊炉中炉温曲线满足制程界限的最大传输速度，即

$$\max F = v \quad (32)$$

约束条件：运行速度  $v$  需要使得电路板在回焊炉中的炉温曲线满足制程界限，即

$$\begin{cases} -3 \leq \frac{du}{dt} \leq 3 \\ 60 \leq t_{150 \leq u \leq 190} \leq 120 \\ 40 \leq t_{u \geq 217} \leq 90 \\ 240 \leq u_{\max} \leq 250 \\ 65 \leq v \leq 100 \\ \frac{du}{dt} = -k(u - u_{av}(t)) \end{cases} \quad (33)$$

其中， $t_{150 \leq u \leq 190}$  为温度上升过程中在 150–190°C 的时间；

$t_{u \geq 217}$  为温度大于 217°C 的时间；

$u_{\max}$  为峰值温度；

$v$  为过炉速度。

#### 5.3.2 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：

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

最大过炉速度时炉温曲线如图 12 所示:

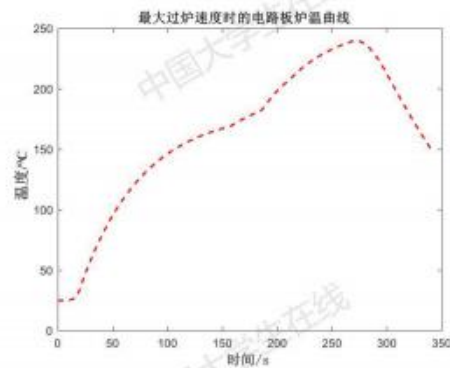


图 12 最大过炉速度下的炉温曲线

#### 5.4 问题三的建模与求解

##### 5.4.1 问题三模型建立

在焊接过程中, 炉温曲线如图 13 所示

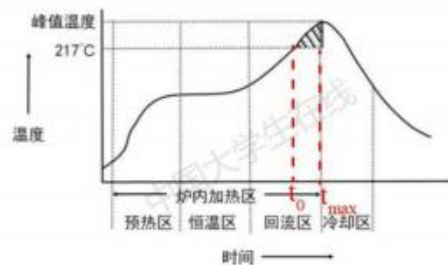


图 13 焊接过程的炉温曲线

图中阴影部分面积为:



$$S_1 = \int_{t_0}^{t_{\max}} (u(t) - 217) dt \quad (34)$$

式中,  $t_0$  为电路板第一次到达 217℃ 时对应的时间,  $t_{\max}$  为电路板到达峰值时对应的时间。

在本题中各小温区设定温度可以进行  $\pm 10^\circ\text{C}$  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℃。传送带的过炉速度调节范围为 65~100 cm/min。

同时, 理想的炉温曲线需满足题目所要求的制程界限。

因此, 可建立如下单目标优化模型:

#### 决策变量

传送带速度及各个温区的温度, 即

$$\{v, T_1, T_2, T_3, T_4\} \quad (35)$$

#### 目标函数

目标为阴影面积最小, 即

$$\min S_1(v, T_1, T_2, T_3, T_4) = \int_{t_0}^{t_{\max}} (u(t) - 217) dt \quad (36)$$

#### 约束条件

速度温度应满足的上下限, 即

$$\begin{cases} 65 \leq v \leq 100 \\ 165 \leq T_1 \leq 185 \\ 185 \leq T_2 \leq 205 \\ 225 \leq T_3 \leq 245 \\ 245 \leq T_4 \leq 265 \end{cases} \quad (37)$$

炉温曲线应满足的制程界限, 即

$$\begin{cases} -3 \leq u'(t) \leq 3 \\ 60 \leq t_{150 \leq u \leq 190} \leq 120 \\ 40 \leq t_{u \geq 270} \leq 90 \\ 240 \leq u_{\max} \leq 250 \\ \frac{du}{dt} = -k(u - u_{\infty}(t)) \end{cases} \quad (38)$$

综上所述, 优化模型为:

$$\begin{aligned}
\min S_1 &= \int_{t_0}^{t_{\max}} (u(t) - 217) dt \\
&\begin{cases} 65 \leq v \leq 100 \\ 165 \leq T_1 \leq 185 \\ 185 \leq T_2 \leq 205 \\ 225 \leq T_3 \leq 245 \\ 245 \leq T_4 \leq 265 \end{cases} \\
s.t. &\begin{cases} -3 \leq u'(t) \leq 3 \\ 60 \leq t \Big|_{\substack{150 \leq u(t) \leq 190}} \leq 120 \\ 40 \leq t \Big|_{\substack{u(t) \geq 270}} \leq 90 \\ 240 \leq u_{\max} \leq 250 \\ \frac{du}{dt} = -k(u - u_{\text{env}}) \end{cases}
\end{aligned} \quad (39)$$

#### 5.4.2 问题三模型求解

本题中决策变量略多，且决策变量变化范围大，故解空间较大，若采用遍历搜索则在有限时间内不能搜索到有效结果，故采用模拟退火算法来实现对其求解。下面先对模拟退火算法进行简单的介绍。

##### (1) 模拟退火算法简要介绍

模拟退火算法最早由 Metropolis 提出，1983 年由 Kirkpatrick 等人将模拟退火的思想成功引入组合优化领域。目前，模拟退火算法已经应用到各门学科中以解决非线性系统的优化问题。

理论上已经证明，模拟退火算法是一个全局最优算法，而且概率 1 接近最优值，并且克服了对初值的依赖。算法的基础源于对固体退火过程的模拟，采用 Metropolis 准则，用冷却进度表控制算法进程，最终得到一个近似最优解。固体退火是指将固体加热到足够高的温度后，使分子呈随机排列状态，然后逐步降温使之冷却，最后分子以低能状态排列，固体达到某种稳定状态。

模拟退火算法的基本思想：在一定的温度下，搜索从一个状态随机地变化到另一个状态，随着温度的不断下降直到最低温度，搜索过程以概率 1 停留在最优解。根据玻尔兹曼概率分布可以得出：

1. 在同一个温度，分子停留在能量较小状态的概率大于停留在能量较大状态的概率。
2. 温度越高，不同能量状态对应的概率相差越小；温度足够高时，各状态对应的概率基本相同。
3. 随着温度的下降，能量最低状态对应的概率越来越大；温度趋于 0 时，其状态趋于 1。

##### (2) 利用模拟退火算法的求解流程

本题采用模拟退火算法的基本步骤如下：

**Step1:** 设置初始解及其控制参数。

**Step2:** 判断是否满足终止条件。

**Step3:** 随机给出一种方案，计算电路板在回焊炉中的炉温曲线，判断是否满足约束条件，若满足，则是对当前解进行一次随机扰动，即产生新的方案；若不满足，返回 Step3。

**Step4:** 比较新旧方案，利用 Metropolis 准则，更新方案。

**Step5:** 判断是否达到降温停止条件，若满足则输出方案，否则，返回 Step3。

模拟退火算法在本题中的具体流程如图 14 所示

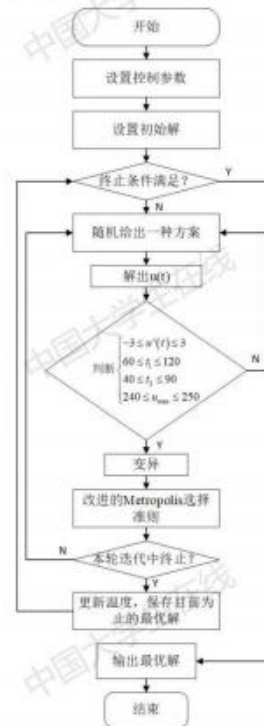


图 14 模拟退火算法的流程示意图

#### 5.4.3 问题三求解结果

##### (1) 结果

初始温度设置为 5000°C，冷却系数为 0.94，步长为 100，截至温度为 0.1°C。

通过使用 *matlab* 编程求解, 可得:

过炉速度为  $v = 91.889 \text{ cm/min}$

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

各温区温度如下表所示:

表 4 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	264.967°C

同时在该求解结果下, 电路板炉温曲线如图 15 所示

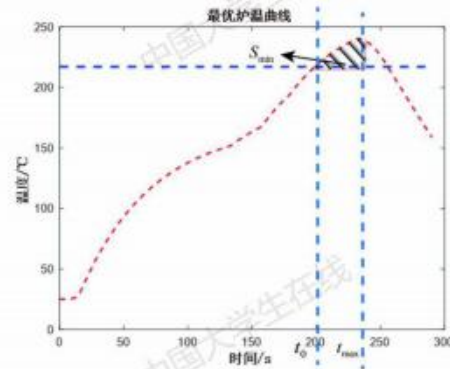


图 15 阴影面积最小时的炉温曲线

## (2) 结果的置信度说明

模拟退火算法内能曲线如图 16 所示

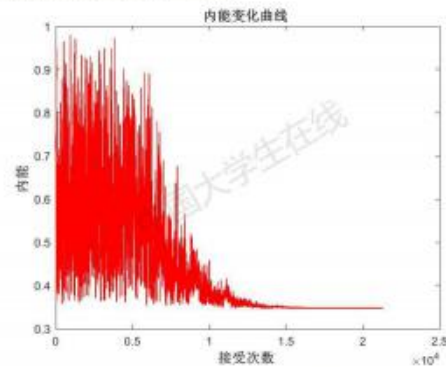


图 16 模拟退火算法收敛曲线

由图可见，内能变化最终收敛，故上述结果较优。

## 5.5 问题四的建模与求解

### 5.5.1 模型建立

设有两函数  $f(x), g(x)$  如图 17 所示，某炉温曲线如图 18 所示：

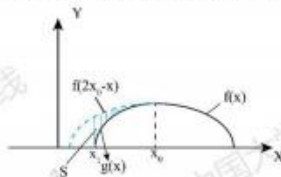


图 17 示意图

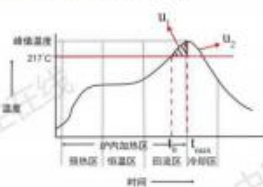


图 18 炉温曲线图

$f(x)$  关于  $x_0$  对称的函数为  $f(2x_0 - x)$ ，则  $g(x)$  与  $f(x)$  关于  $x_0$  对称的等价条件为：

$$\Delta S = \int_{x_0}^{x_0} |f(2x_0 - x) - g(x)| dx = 0 \quad (40)$$

由此可得，炉温曲线  $u_1, u_2$  关于  $t = t_{\max}$  对称的等价条件为：

$$S_2 = \int_{t_0}^{t_{\max}} |u(2t_{\max} - t) - u(t)| dt = 0 \quad (41)$$

其中， $t_0$  为电路板第一次到达 217°C 时对应的时间， $t_{\max}$  为电路板到达峰值时对应的时间。

本题不仅要求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，同时还要满足问题三中阴影部分面积  $S_1$  最小，故为多目标优化问题。

对于多目标优化问题通常解法是将多个指标标准化，再进行加权平均，即

$$S = \omega_1 \frac{S_1}{S_{1,\max}} + \omega_2 \frac{S_2}{S_{2,\max}} \quad (42)$$

其中， $S_{1,\max}, S_{2,\max}$  分别为  $S_1, S_2$  的最大值； $\omega_1, \omega_2$  为权重，且满足  $\omega_1 + \omega_2 = 1$ 。

本模型采用动态综合加权法，其中阴影面积  $S_1$  先是缓慢增加，中间有一个快速增长的过程，随后平缓增加趋于最大，故  $\omega_1$  可以设定为偏大型正态分布函数，即

$$\omega_1(t) = \begin{cases} 0 & x \leq \alpha_1 \\ 1 - e^{-\frac{(x-\alpha_1)^2}{\sigma_1^2}} & x > \alpha_1 \end{cases} \quad (43)$$

其图像如图 19 所示

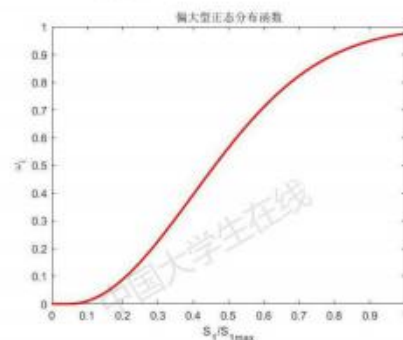


图 19 偏大型正态分布

综上所述，本题的多目标优化模型如下

$$\begin{aligned} \min S &= \omega_1 \frac{S_1}{S_{1,\max}} + \omega_2 \frac{S_2}{S_{2,\max}} \\ \text{s.t. } &\begin{cases} 65 \leq v \leq 100 \\ 165 \leq T_1 \leq 185 \\ 185 \leq T_2 \leq 205 \\ 225 \leq T_3 \leq 245 \\ 245 \leq T_4 \leq 265 \\ -3 \leq u'(t) \leq 3 \\ 60 \leq t \begin{cases} 0.50 \sin(t) \leq 90 \\ \ln(t) \geq 270 \end{cases} \leq 120 \\ 40 \leq t \begin{cases} \ln(t) \geq 270 \end{cases} \leq 90 \\ 240 \leq u_{\max} \leq 250 \\ \frac{du}{dt} = -k(u - u_{\text{avr}}) \end{cases} \end{aligned} \quad (44)$$

### 5.5.2 模型求解

#### (1) 求解方法

本题仍然采用模拟退火算法进行求解。

#### (2) 基本步骤

**Step 1:** 先通过模拟退火算法分别算出  $S_{1,\max}$  和  $S_{2,\max}$ 。

**Step 2:** 将多目标函数转变为单目标函数，即将  $S_1$ 、 $S_2$  标准化，再引入  $\omega_1$ 、

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



**Step 3:** 以  $\min S = \omega_1 \frac{S_1}{S_{1,\max}} + \omega_2 \frac{S_2}{S_{2,\max}}$  为目标函数，再通过模拟退火算法求得各个温区温度以及传送带速度。

流程如图 20 所示：

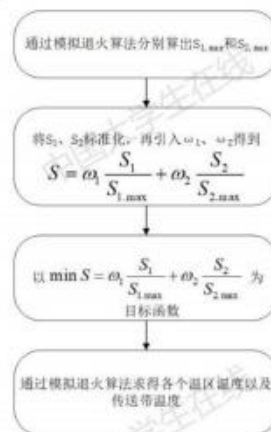


图 20 问题四求解流程图

### (3)求解结果

初始温度设置为 5000°C，冷却系数为 0.94，步长为 100，截至温度为 0.0008°C。通过 *matlab* 编程，求得：

$$v = 88.782 \text{ cm/min}, S_{1,\min} = 449.17, S_{2,\min} = 35.872, S_{\min} = 0.3486$$

各温区温度如表 5 所示

表 5 各温区温度设定值

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

同时在该求解结果下，电路板炉温曲线如图 21 所示

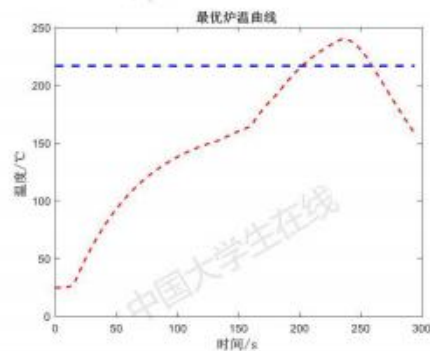


图 21 最优解下的炉温曲线

## (2) 结果的置信度说明

模拟退火算法内能曲线如图 22 所示

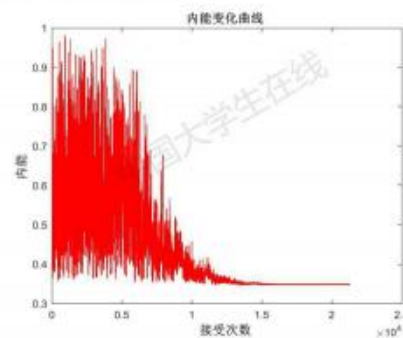


图 22 模拟退火算法内能变化曲线

由图可见，内能变化最终收敛，故上述结果较优。

## 6 结果检验与误差分析

### 6.1 问题二结果检验

在问题二中，对传送带最大过炉速度进一步验证，当对最大过炉速度进行适当的升高时，可以发现电路板的炉温曲线不再满足题中要求的制程界限；对最大过炉速度进行适当的降低，可以发现电路板的炉温曲线满足题中要求的制程界限，因此可以验证问题二最大过炉速度解的正确性。

### 6.2 问题三、四结果检验

固定已经得到的结果  $T_1, T_2, T_3$  的值, 任意改变 10 次  $v, T_4$  的值, 发现总体的目标函数值都有所上升, 这初步验证了结果的较优性。

第四问中, 范数  $S_2$  的最大值为 199.50, 而第四问中范数仅为 35.87, 并从图形中可以观察到图形的对称性, 这再次验证了结果的合理性。

### 6.3 分段合理性分析

根据式 (30), 冷却系数  $k$  值实际上是随时间不断变化的, 但本文做了简化处理, 按照预热区、恒温区、回流区、冷却区进行分段,  $k$  值在这些段中相同, 段间  $k$  值不同。最后求得的电路板焊接中心温度数据与试验数据比较如图 23 所示:

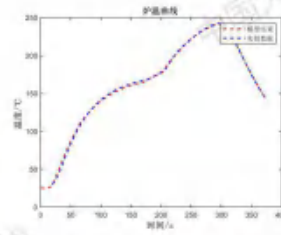


图 23  $k$  值分段时模型结果

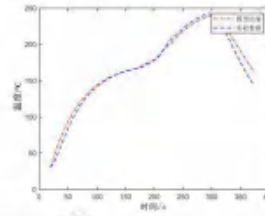


图 24  $k$  值取值时模型结果

模型结果与原始实验数据的均方误差为:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (x_i^{model} - x_i^{test})^2}{n}} \quad (45)$$

模型结果与原始实验数据的平均绝对误差为:

$$MAE = \frac{\sum_{i=1}^n |x_i^{model} - x_i^{test}|}{n} \quad (46)$$

$k$  值分段求得时模型结果与原始实验数据根据公式 (45) 与公式 (46) 求得均方误差为  $2.81^{\circ}\text{C}$ , 平均绝对误差为  $1.73^{\circ}\text{C}$ ;  $k$  取定值 0.193 时模型结果与原始实验数据的均方误差为  $6.9475^{\circ}\text{C}$ , 平均绝对误差为  $4.8568^{\circ}\text{C}$ ,  $k$  值分段时模型结果的误差要小于  $k$  值不分段时的误差, 验证了本题分段求取  $k$  值的合理性。

### 6.4 对终止温度的分析

在问题四中, 已将目标函数做了标准化处理, 从而导致对终止温度的要求较高, 终止温度对结果的收敛性起到至关重要的作用, 图 (26) 为终止温度为  $0.1^{\circ}\text{C}$  时收敛曲线图。由图 (26) 可见, 在终止时结果未收敛, 为了进一步使曲线收敛, 我们选择了二次退火, 并大幅度降低终止温度, 当终止温度  $0.0008^{\circ}\text{C}$  时, 收敛效果较好, 如图 (27) 所示。

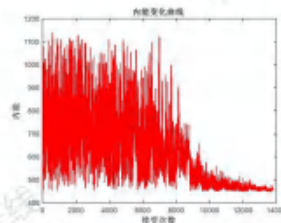


图 26 0.1℃时内能变化曲线

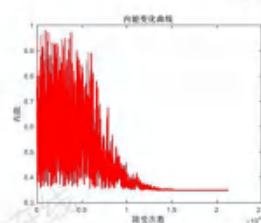


图 27 0.0008℃时的内能变化曲线

## 7 模型评价

### 模型优点:

#### (1) 热传导方程模型:

使用热传导方程模型能够较好的反应出电路板在回焊炉中焊接的物理过程, 考虑到了电路板在回焊炉中, 温度在电路板厚度上的分布, 有助于对回焊炉过程进行更加深入的研究。

#### (2) 基于牛顿冷却定律模型:

基于牛顿冷却定律模型是在热传导方程基础上简化条件后建立的, 相比如热传导方程, 基于牛顿冷却定律模型不再考虑温度在电路板厚度上的分布, 求解简单, 求解效率大大提高, 有利于在此基础上进行其他优化问题的研究。

### 模型缺点:

#### (1) 热传导方程模型:

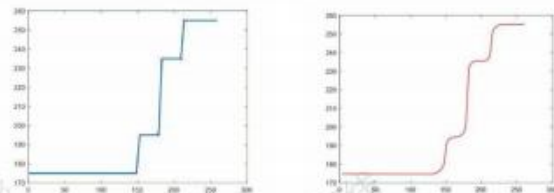
使用热传导方程模型求解过程复杂, 在较短的时间内难以求得最优的参数, 并且对于厚度较薄的电路板, 温度在厚度上的分布接近一致。

#### (2) 基于牛顿冷却定律模型:

基于牛顿冷却定律模型未考虑到电路板在回焊炉中温度在电路板厚度上的分布, 对于厚度较厚的电路板在回焊炉中, 模型过于简化。

## 8 模型推广与改进

本文在考虑温度在两边温度不等的间隙中的分布时认为, 温度随空间位置线性变化, 如图所示, 但在间隙与温区边界上, 温度连续但不可导。故对温区与间隙边界上的温度分布进行修正, 如图所示, 此时, 温度在间隙与温区边界上既连续又可导。



## 9 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.

## 10 程序附录

```

问题一: main
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('example')
dt=0.5;
v=78/60;%传送带速度
t=0;dt;435.5/v;%求解区间
Tair=T_air(t,v,T15,T6,T7,T89,T1011);%空气温度
T=zeros(length(Tair),1);%电路板温度
T(1)=25;%电路板初始温度
for i=2:length(Tair)
    k=kk(v*(i-1)*dt);
    T(i)=-dt*(T(i-1)-Tair(i))*k+T(i-1);%差分求解
end
plot(t,T,'-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','Times New Roman','fontweight','bold','fontsize',12)

```

```
title('电路板升温曲线','Fontname','宋体','fontweight','bold','FontSize',12)
```

#### 问题二: main2

```
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);
    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*60
[tspan1,u1]=problem2(vmax,T15,T6,T7,T89,T1011);
plot(tspan1,u1,'-r')
% hold on
% plot(u)
```



```

plot(tspan1,u1,'-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','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)

```

### 问题三: main3

```

clear;clc;

temperature = 10000;           % Initialize the temperature.
cooling_rate = 0.94;           % cooling rate

previous_x=[90.4998096731545,175.474315263950,197.983991611696,228.256966424
909,264.881809804930];
% This is objective function, the total distance for the routes.
[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);
i=1;
pl(i)=previous_area;
% This is a flag used to cool the current temperature after 100 iterations
temperature_iterations = 1;

while 2 < temperature
    % generate randomly a neighbouring solution
    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);
    % compute change of area
    diff = current_area - previous_area;

    % Metropolis Algorithm
    if (diff < 0) || (rand < exp(-diff/(temperature)))

```

```

previous_x = current_x; %accept new route
previous_area = current_area;
i=i+1;
pl(i)=previous_area;
% update iterations
temperature_iterations = temperature_iterations + 1;

end

% reduce the temperature every 100 iterations
if temperature_iterations >= 100
    temperature = cooling_rate*temperature;
    temperature_iterations = 0;
end

% plot the current route every 200 iterations
temperature

end

% plot the final solution
[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)
plot(pl,'-r','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])
sss=previous_area

```

#### 问题四: main4

```
clear;clc;

temperature = 1000;           % Initialize the temperature.
cooling_rate = 0.94;          % cooling rate

previous_x=[88.7818206515348,169.733087638798,186.657314640567,231.844263600
990,264.999044246197];

% This is objective function, the total distance for the routes.
[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;
% This is a flag used to cool the current temperature after 100 iterations
temperature_iterations = 1;

while 0.000080 < temperature
    % generate randomly a neighbouring solution
    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);
    % compute change of area
    diff = (current_area - previous_area);

    % Metropolis Algorithm
    if (diff < 0) || (rand < exp(-diff/(temperature)))
        previous_x = current_x;      %accept new route
        previous_area = current_area;
        i=i+1;
        pl(i)=previous_area;
```

```

% update iterations
temperature_iterations = temperature_iterations + 1;

end

% reduce the temperature every 100 iterations
if temperature_iterations >= 100
    temperature = cooling_rate*temperature;
    temperature_iterations = 0;
end

% plot the current route every 200 iterations
temperature
end

% plot the final solution
[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)
pll=[pl((length(pl)/5*3):length(pl)) pl((length(pl)*19.5/20):length(pl))
    pl((length(pl)*19.5/20):length(pl)) pl((length(pl)*19.5/20):length(pl))];
plot(pll, 'r', '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

```

kk

```
function k=kk(x)
if x>=0&&x<=25
    k=0.0185;
elseif 25<x&&x<=202.5
    k=0.0177;
elseif 202.5<x&&x<=238
    k=0.017;
elseif 238<x&&x<=273.5
    k=0.0247;
elseif 273.5<x&&x<=339.5
    k=0.0202;
else
    k=0.029;
end
end
```

#### T\_air

```
function y= T_air(t,v,T15,T6,T7,T89,T1011)
y=ceil((heaviside(v*t)-heaviside(v*t-25)).*((T15-25)*exp(0.251*(v*t-25)))+25)+...
floor(heaviside(v*t-25)-heaviside(v*t-197.5)).*T15+...
ceil(heaviside(v*t-197.5)-heaviside(v*t-202.5)).*(((T6-T15)/5)*(v*t-197.5)+T15)+...
floor(heaviside(v*t-202.5)-heaviside(v*t-233)).*T6+...
ceil(heaviside(v*t-233)-heaviside(v*t-238)).*(((T7-T6)/5)*(v*t-233)+T6)+...
floor(heaviside(v*t-238)-heaviside(v*t-268.5)).*T7+...
ceil(heaviside(v*t-268.5)-heaviside(v*t-273.5)).*(((T89-T7)/5)*(v*t-268.5)+T7)+...
floor(heaviside(v*t-273.5)-heaviside(v*t-339.5)).*T89+...
ceil(heaviside(v*t-339.5)-heaviside(v*t-380)).*(((25-T89)/96)*(v*t-339.5)+T89)+...
floor(heaviside(v*t-380)).*((((((25-T89)/96)*(380-339.5)+T89)-25)*exp(-0.00991*(v*t-380)))+25);%+...
% ceil(heaviside(v*t-410.5)-heaviside(v*t-435.5)).*(((25-T1011)/25)*(v*t-410.5)+T1011);

end
```

#### problem2

```
function [tspan,T]=problem2(v,T15,T6,T7,T89,T1011)
dt=0.2;
```

```

tspan=0:dt:435.5/v;

Tair=T_air(tspan,v,T15,T6,T7,T89,T1011);
T=zeros(length(Tair),1);
T(1)=25;%电路板初始温度

for i=2:length(Tair)
    k=kk(v*(i-1)*dt);
    T(i)=-dt*(T(i-1)-Tair(i))*k+T(i-1);%差分求解
end
end

```

#### perturb

```

function y=perturb(x)
flag=0;
while flag==0
    x_new=x;
    local=ceil(rand*5);
    switch local
        case 1
            x_new(1)=rand*35+65;
        case 2
            x_new(2)=rand*20+165;
        case 3
            x_new(3)=rand*20+185;
        case 4
            x_new(4)=rand*20+225;
        case 5
            x_new(5)=rand*20+245;
    end
    [tspan,u]=problem2(x_new(1)/60,x_new(2),x_new(3),x_new(4),x_new(5),25);
    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);
if (uumin>-3)&&(uumax<3)&&...
    (t1>60)&&(t1<120)&&...
    (t2>40)&&(t2<90)&&...
    (Tmax>240)&&(Tmax<250)
    y=x_new;
    flag=1;
end
end
end

```

#### S

```

function s=S(u,dt)
a=find(u>217);
b=find(u==max(u));
s=sum((u(a(1):b-1)-217)*dt);
end

```

#### SDDD

```

function s=SDDD(u,dt)
s1=S(u,dt)/1151.4;
s2=SF(u,dt)/199.5;
sgm=0.494;
if s1<0.05
    s=s2;
else
    w=1-exp(-(s1-0.05)/sgm)^2);
    s=(1-w)*s1+(w)*s2;
end
end

```

#### SF

```

function s=SF(u,dt)
a=find(u>217);
b=find(u==max(u));
if (b-a(1))<=(a(end)-b)
    u1=u(a(1):b-1);

```

```
u2=u((2*b-a(1)):-1:b+1);  
  
else  
    u1=u((b+1):a(end));  
    u2=u((b-1):-1:(2*b-a(end)));  
end  
s=sum(abs(u1-u2)*dt);  
end
```