

# 基于一维热传导的炉温曲线模拟与优化

## 摘要

炉温曲线的控制决定了电子元件焊接的产品质量,本文基于一维热传导方程和牛顿冷却定律建立了炉温曲线变化规律的机理模型。

对于问题一,首先基于热传导方程,推导出温区间隙之间温度线性过渡,建立回焊炉内部的温度分布模型。再通过热传导方程与牛顿冷却定律,建立焊接区域的温度分布模型,边界条件为第三类边界条件,建立微分控制方程组(包含6个待求参数,5个 $\alpha$ ,1个 $\beta$ )。利用有限差分法对方程组逐层求解,基于最小二乘原理,拟合实测温度,遍历得到最优的参数组合,其中最优 $\alpha$ 为 $[3.78E-07, 4.95E-07, 5.71E-07, 4.74E-07, 2.10E-07]cm^2/s$ ,最优 $\beta$ 为 $82197cm^{-1}$ 。拟合数据与实际数据的 $R^2$ 为0.9997,拟合效果较优。将求得的未知参数及问题一相关参数代入方程求解,得到炉温曲线,其中小温区3、6、7中点及小温区8结束处焊接区域中心的温度分别为 $129.2787^\circ C$ ,  $168.2546^\circ C$ ,  $189.1527^\circ C$ ,  $231.5747^\circ C$ ,每隔0.5s焊接区域中心的温度存放在result.csv中。

对于问题二,建立以最大过炉速度为目标的单目标优化模型。以制程界限为约束条件。对约束条件进行差分,得出差分后的约束条件。应用问题一中得出的热传递模型,并利用变范围搜索的方法,搜索该温度分布条件下最大速度为 $78.95cm/min$ 。

对于问题三,根据问题三的要求与制程界限约束,将该问题转换为关于温区温度和过炉速度的,以面积最小化为目标的单目标多变量最优化问题,并使用遗传算法求解。本文进行了多次独立求解,在多次结果中取最优的结果。最终确定的最佳变量值为 $182.1140^\circ C$ (小温区1-5), $189.8592^\circ C$ (小温区6), $230.2581^\circ C$ (小温区7), $264.9846^\circ C$ (小温区8-9),过炉速度为 $91.5834cm/min$ ,阴影部分面积为 $488.2102(^\circ C \cdot s)$ 。

对于问题四,同样根据多种约束,建立以面积最小化和对称性指标最小化为目标的多目标多变量最优化问题,利用遗传算法,结合分层序列法的思想,先优化面积指标,再优化对称性指标,多次求解得到多组求解结果,并取最优结果,最佳变量值为 $172.3431^\circ C$ (小温区1-5), $185.0034^\circ C$ (小温区6), $230.637^\circ C$ (小温区7), $265.00^\circ C$ (小温区8-9),过炉速度为 $88.7126cm/min$ ,其中面积指标为 $487.2066(^\circ C \cdot s)$ ,对称性指标为 $9.3831(^\circ C)$ 。

最后,本文对相关参数进行了灵敏性分析,并对模型作出了评价。

**关键字:** 一维热传导 参数拟合 有限差分法 变范围搜索 分层序列法 遗传算法

# 一、问题重述

## 1.1 问题背景

为了生产集成电路板，将安装有电子元件的电路板置于回焊炉中，通过高温空气加热后元件被焊接在电路板上。回焊炉内有几个小温区，以功能上划分为 4 大温区：预热区、恒温区、回流区、冷却区。电路板搭在传送带上匀速进入炉内进行焊接。

某回焊炉内有炉前区域和炉后区域和 11 个小温区。小温区长为 30.5 cm，相邻小温区间隔 2cm，炉前和炉后区域长为 25 cm。启动回焊炉后，短时间内炉内空气温度达到稳态。生产车间的温度保持在 25°C。

通过装载在电路板中心的热电偶等装置，可以测试焊接区域中心的温度，即炉温曲线。附件给出了某次实验数据，在本次实验中，各温区设定的温度分别为 175°C、195°C、235°C、255°C 及 25°C，传送带速度为 70cm/min，焊接区域厚为 0.15 mm。传感器在温度大于 30°C 时开始工作，电路板进入炉前区域时，开始计时。

通过改变各温区的温度以及传送带速度，可以控制产品质量。以本次实验温度为基础，各温区的温度能够在  $\pm 10^\circ\text{C}$  范围内进行调整，并且要求温区 1-5 的温度相同，温区 8-9 中温度相同，温区 10-11 温度保持 25°C。传送带速度范围为 65-100 cm/min。同时在电路板焊接生产过程中，炉温曲线应满足制程界限，例如上升下降斜率不能大于 3 等等。

## 1.2 问题要求

**问题 1** 首先建立炉温曲线的模型。在此基础上，传送带速度变为 78 cm/min，各温区温度相应地改变为 173°C、198°C、230°C 和 257°C，要求求出该情况下的炉温曲线，列出小温区 3、6、7 中点及小温区 8 结束炉温曲线温度。还需要将每隔 0.5 s 焊接区域中心的温度存放在 result.csv 中。

**问题 2** 各温区温度变成为 182°C、203°C、237°C、254°C，求解允许的最大过炉速度。

**问题 3** 炉温曲线超过 217°C 的时间不能过长，峰值温度也不能过高，超过 217°C 到峰值温度所覆盖的面积应该最小。求解该要求下的最优炉温曲线，以及各温区温度和过炉速度，并计算相应的面积指标。

**问题 4** 在高温焊接过程中，除了需要满足制程界限，还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。在问题 3 的基础上，给出最优炉温曲线以及各温区温度及过炉速度，同时给出相应的面积指标值和对称性指标值。

## 二、问题分析

### 2.1 问题一分析

对于问题一，首先基于热传导方程，推导出温区间隙之间温度线性过渡，建立回焊炉内部的温度分布模型。再通过热传导方程与牛顿冷却定律，建立焊接区域的温度分布模型，边界条件为第三类边界条件，建立微分控制方程组。基于最小二乘法，利用附件数据进行参数拟合，得到误差最小时最优参数。带入问题一中给出的回焊炉各温区的温度数据，得出焊接处中心温度随时间变化的情况。

### 2.2 问题二分析

对于问题二，在题目给定的温度条件和制程界限的限制下，求解最大化传送带速度的单目标规划模型。基于问题一中得出的炉温曲线模型，利用变范围搜索方法，对传送带的可能速度进行多次遍历，以得出精度较高的解。取  $1\text{cm}/\text{min}$  的速度步长，从  $60\text{cm}/\text{min}$  往上遍历后选取出第一个满足制程界限的解，即为最大解。

### 2.3 问题三分析

对于问题三，根据问题要求与制程界限约束，将该问题转换为关于温区温度和过炉速度的单目标多变量最优化问题，求出最优的温区设定温度和传送带的过炉速度，使得在满足制程界限的同时，炉温曲线超过  $217^{\circ}\text{C}$  到峰值温度所覆盖的面积也达到最小。由于参数过多，遍历搜索的难度极大，可以考虑采用遗传算法等智能算法求解。

### 2.4 问题四分析

问题四是在问题三基础上的多目标优化问题，需要满足制程界限的同时，在问题三炉温曲线超过  $217^{\circ}\text{C}$  到峰值温度所覆盖的面积达到最小的基础上，做到以峰值温度中线的两侧超过  $217^{\circ}\text{C}$  的炉温曲线尽量对称的目标。因此建立以面积最小化和对称性指标最小化为目标的多目标多变量最优化问题，利用遗传算法，结合分层序列法的思想，先优化面积指标，再优化对称性指标，最终求得最佳参数。

## 三、模型假设

- 元件在传送带上运行时与小温区始终保持垂直。
- 回焊炉内温度稳定，只在沿传送方向上变化，垂直于传送方向的平面温度稳定。
- 假设元件为材质处处均匀的带厚度无限大平板，内部不含热源
- 仅考虑热传导和热对流对元件焊接区域中心温度的影响

## 四、符号说明

符号	说明	单位
$l$	小温区长度	$cm$
$a$	小温区间隔	$cm$
$b$	炉前炉后区域长度	$cm$
$d$	电路板厚度	$cm$
$t$	时间	$s$
$u(x, t)$	电路板各处温度分布	-
$u(t)$	电路板中心温度随时间的变化 (炉温曲线)	-
$u_s(t)$	空气温度随时间的变化	-
$T_{15}$	小温区 1-5 设定温度	$^{\circ}C$
$T_{06}$	小温区 6 设定温度	$^{\circ}C$
$T_{07}$	小温区 7 设定温度	$^{\circ}C$
$T_{89}$	小温区 8-9 设定温度	$^{\circ}C$
$T_{1011}$	小温区 10-11 设定温度	$^{\circ}C$
$v$	传送带过炉速度	$cm/s$
$S$	问题 3、4 的面积指标	$(^{\circ}C \cdot s)$
$D$	问题 4 对称性指标	$(^{\circ}C)$

## 五、模型的建立和求解

### 5.1 炉内温度随位置的变化

问题一中，由题可知，炉内空气温度会在短时间内达到稳定，因此首先要建立稳态时回焊炉中空气温度随位置分布的规律。对于空气温度随  $u_s(x)$ ，在热导率一定时，根据热导微分方程式，有

$$\frac{\partial u_s(x)}{\partial t} = \alpha \frac{\partial^2 u_s(x)}{\partial x^2} \quad (1)$$

对于稳态温度场，有  $\frac{\partial u_s(x)}{\partial t} = 0$ ，因此  $\alpha \frac{\partial^2 u_s(x)}{\partial x^2} = 0$ ，小温区内空气温度等于设置温度，对于小温区间隙中的温度，对式 (1) 积分，有

$$u_s(x) = k_1 x + b_1 \quad (2)$$

即小温区间隙中的温度随着空间均匀变化，线性地由上一台阶温度过渡到下一台阶温度。

设炉前炉后距离为  $b$ ，各小温区的间距为  $a$ ，各小温区的长度为  $l$ ，令各分割点的位置为

$$\begin{cases} len0 = b \\ len1 = b + 5l + 4a \\ len2 = b + 5l + 5a \\ len3 = b + 6l + 4a \\ len4 = b + 6l + 6a \end{cases} \begin{cases} len5 = b + 7l + 6a \\ len6 = b + 7l + 7a \\ len7 = b + 9l + 8a \\ len8 = b + 9l + 9a \\ len9 = b + 11l + 10a + b \end{cases} \quad (3)$$

则空气温度  $U_s(x)$  关于距离  $x$  的分段函数为：

$$U_s(x) = \begin{cases} \frac{T_{15}-25}{len0}x + 25 & 0 < x \leq len0 \\ T_{15} & len0 < x \leq len1 \\ \frac{T_{06}-T_{15}}{len2-len1}(x - len0) + T_{15} & len1 < x \leq len2 \\ T_{15} & len2 < x \leq len3 \\ \frac{T_{07}-T_{06}}{len4-len3}(x - len3) + T_{06} & len3 < x \leq len4 \\ T_{07} & len4 < x \leq len5 \\ \frac{T_{89}-T_{07}}{len6-len5}(x - len5) + T_{07} & len5 < x \leq len6 \\ T_{89} & len6 < x \leq len7 \\ \frac{T_{1011}-T_{89}}{len8-len7}(x - len7) + T_{15} & len7 < x \leq len8 \\ T_{1011} & len8 < x \leq len9 \end{cases} \quad (4)$$

其中  $T_{15}$  是小温区 1-5 的温度， $T_{06}$  是小温区 6 的温度， $T_{07}$  是小温区 7 的温度， $T_{89}$  是小温区 8-9 的温度， $T_{1011}$  是小温区 10-11 的温度。

考虑到传送带是匀速运动的，电路板的位移  $x = vt$ ，模型中讨论的电路板周围空气温度可以看作是随时间变化的量，空气温度随时间的变化函数  $u_s(t)$  可以写作  $U_s(vt)$ 。

$$u_s(t) = U_s(x) = U_s(vt) \quad (5)$$

传送带运动速度为  $70\text{cm/s}$ ，在此速度下回焊炉中的温度随时间 (位置) 变化如图示

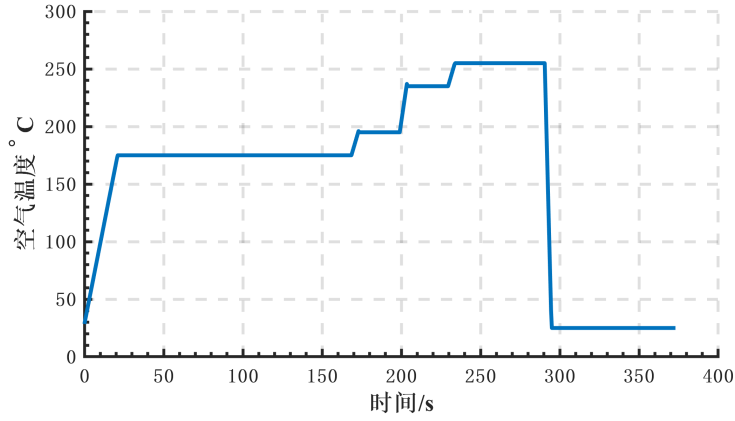


图 1 回焊炉温度随时间（位置）的变化

## 5.2 电路板上一维热传导模型的建立

对于电路板上的温度  $u(x, t)$ ，以电路板下端中心为原点建立坐标轴，方向向上为正：

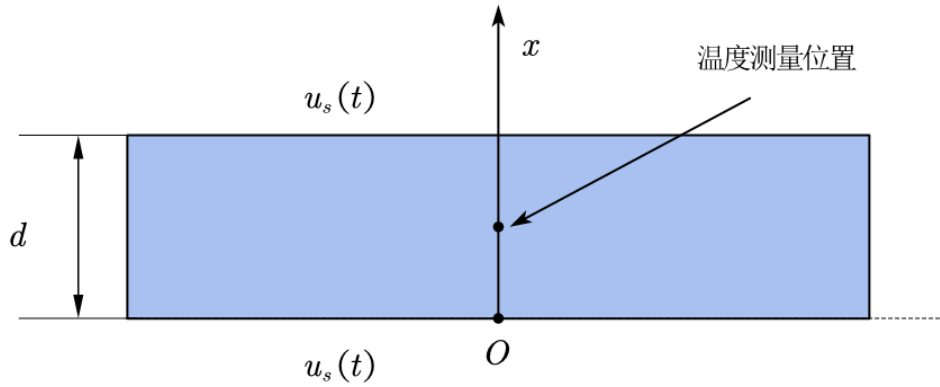


图 2 电路板示意图

同样根据热导微分方程式，得到控制方程

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

其中  $\lambda$  为热导率， $c, \rho$  分别为比热容与密度。在电路板的上下边界上，满足第三类边界条件，根据牛顿冷却公式，物体边界与流体对流换热量可以写为  $q = h(u - u_s(t))$ ，其中  $h$  为边界与空气的表面传热系数， $u_s(t)$  为空气温度。则对于电路板的上下边界有

$$\begin{cases} \lambda \frac{\partial u}{\partial x}|_{x=d} = h(u_s(t) - u(d, t)) \\ -\lambda \frac{\partial u}{\partial x}|_{x=0} = h(u_s(t) - u(0, t)) \end{cases} \quad (7)$$

上述公式中，令  $\alpha = \frac{\lambda}{c\rho}$ ， $\beta = \frac{h}{\lambda}$ ，得到

$$\begin{cases} \frac{\partial u(x, t)}{\partial t} = \alpha \frac{\partial^2 u(x, t)}{\partial x^2} \\ \frac{\partial u}{\partial x}|_{x=d} = \beta(u_s(t) - u(d, t)) \\ -\frac{\partial u}{\partial x}|_{x=0} = -\beta(u_s(t) - u(0, t)) \end{cases} \quad (8)$$

### 5.3 有限差分法求解热方程

对式 (8) 离散化, 其中边界条件离散为

$$\begin{aligned} -\frac{u_{2,j+1} - u_{1,j+1}}{\Delta x} &= \beta(u_s(j+1) - u_{1,j+1}) \\ \frac{u_{m,j+1} - u_{m-1,j+1}}{\Delta x} &= \beta(u_s(j+1) - u_{m,j+1}) \end{aligned} \quad (9)$$

控制方程离散为

$$\begin{aligned} \frac{u_{i,j+1} - u_{i,j}}{\Delta t} &= \alpha \frac{u_{i-1,j} + u_{i+1,j} - 2u_{i,j}}{(\Delta x)^2} \\ \frac{u_{i,j+1} - u_{i,j}}{\Delta t} &= \alpha \frac{u_{i-1,j+1} + u_{i+1,j+1} - 2u_{i,j+1}}{(\Delta x)^2} \end{aligned} \quad (10)$$

上式取平均得

$$\frac{u_{i,j+1} - u_{i,j}}{\Delta t} = \frac{\alpha}{2} \frac{u_{i-1,j} + u_{i+1,j} - 2u_{i,j} + u_{i-1,j+1} + u_{i+1,j+1} - 2u_{i,j+1}}{(\Delta x)^2} \quad (11)$$

令  $r = \frac{\alpha \Delta t}{(\Delta x)^2}$ , 对公式 (9)(11) 化简得到

$$\begin{aligned} -ru_{i-1,j+1} + 2(1+r)u_{i,j+1} - ru_{i+1,j+1} &= u_{i-1,j} + 2(1-r)u_{i,j} + ru_{i+1,j} \\ (1 + \beta \Delta x)u_{1,j+1} - u_{2,j+1} &= \beta \Delta x u_s(j+1) \\ (1 + \beta \Delta x)u_{m,j+1} - u_{m-1,j+1} &= \beta \Delta x u_s(j+1) \end{aligned} \quad (12)$$

式 (12) 写为矩阵形式:

$$\begin{aligned} &\begin{bmatrix} 1 + \beta \Delta x & -1 & & & \\ & -r & 2(1+r) & -r & \\ & & \ddots & & \\ & & & -r & 2(1+r) & -r \\ & & & & -1 & 1 + \beta \Delta x \end{bmatrix} \begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ \vdots \\ u_{m-1,j+1} \\ u_{m,j+1} \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & & & \\ r & 2(1-r) & r & & \\ & & \ddots & & \\ & & & r & 2(1-r) & r \\ & & & & 0 & 0 \end{bmatrix} \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ \vdots \\ u_{m-1,j} \\ u_{m,j} \end{bmatrix} + \begin{bmatrix} \beta \Delta x u_s(j+1) \\ 0 \\ \vdots \\ 0 \\ \beta \Delta x u_s(j+1) \end{bmatrix} \end{aligned} \quad (13)$$

式 (13) 可以写为以下形式:

$$Au(j+1) = Bu(j) + C \quad (14)$$

其中  $A, B, C$  分别对应 3 个大矩阵。则下一时刻的温度的递推公式为:

$$u(j+1) = A/(Bu(j) + C) \quad (15)$$

初始温度  $u(1) = 25^\circ C$ , 按式 (15) 可以递推得到以后各个时刻的温度。

## 5.4 模型未知参数的确定

$\alpha, \beta$  均为未知参数, 需要通过附件的已知数据进行拟合确定。假设五个大温区的  $\alpha$  不同,  $\beta$  相同, 温度传感器得到的温度由该模型确定  $u$  在位置  $x = d/2$  处随  $t$  的变化, 即  $u(t)|_{x=d/2}$ , 通过对五个区域的  $\alpha$  和  $\beta$  进行遍历, 得到温度曲线  $u(t)|_{x=d/2}$ 。根据最小二乘法原理:

$$F = \sum (\hat{u} - u)^2 \quad (16)$$

为确定 5 个  $\alpha$  和  $\beta$ , 使  $F$  最小, 本文使用了变范围搜索的方法:

**Step1:** 根据文献确定各个  $\alpha$  和  $\beta$  的大致范围, 其中  $\alpha$  的数量级为  $10^{-7} - 10^{-6}$ ,  $\beta$  的数量级为  $10^3 - 10^5$ ;

**Step2:** 先根据上述范围以较大步长搜索最小的误差对应的各个参数值;

**Step3:** 在上一步中找到的最佳参数值周围以较小步长搜索最小的误差对应的各个最佳参数值, 并重复多次, 直到满足精度要求。

最终确定的各个  $\alpha$  和  $\beta$  值如表1, 其中  $\alpha$  的单位为  $cm^2/s$ ,  $\beta$  的单位为  $cm^{-1}$ : 以上

表 1 5 个  $\alpha$  和  $\beta$  参数的值

$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$	$\beta$
3.78E-07	4.95E-07	5.71E-07	4.74E-07	2.10E-07	82197

述参数代入模型, 求解炉温曲线, 与实验数据比较, 求得的误差  $F = 1.2307e + 03$ , 相关系数  $R^2 = 0.9997$ , 可知拟合效果很好。拟合效果如图3。

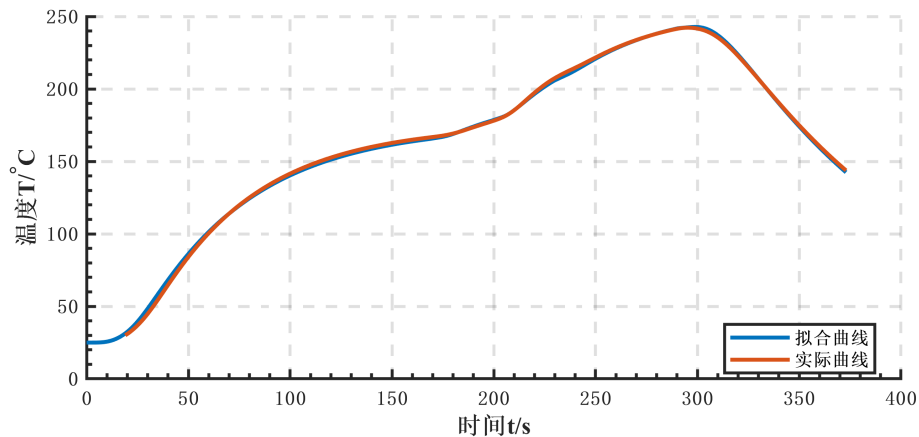


图 3 拟合炉温曲线与实际炉温曲线

## 5.5 问题一的求解

传送带过炉速度为  $78cm/min$ , 各温区的温度分别为  $173^\circ C$  (小温区 1 5)、 $198^\circ C$  (小温区 6)、 $230^\circ C$  (小温区 7) 和  $257^\circ C$  (小温区 8 9), 将上述参数和求得的  $\alpha, \beta$  参数代



入模型求解，得到炉温曲线如图4

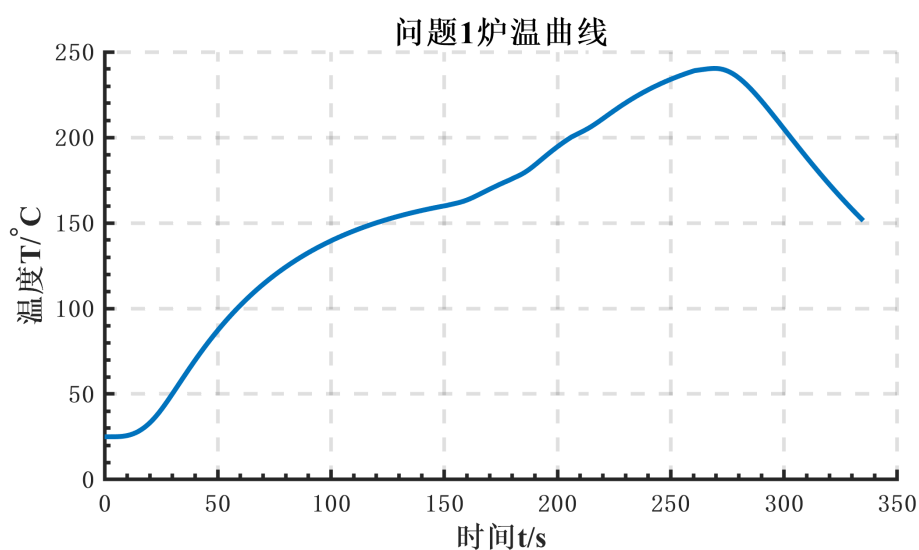


图 4 问题 1 炉温曲线

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表2，每隔 0.5 s 焊接区域中心的温度已经存放在 `result.csv` 中。

表 2 各点温度

小温区 3 中点 ( $^{\circ}\text{C}$ )	小温区 6 中点 ( $^{\circ}\text{C}$ )	小温区 7 中点 ( $^{\circ}\text{C}$ )	小温区 8 结束处 ( $^{\circ}\text{C}$ )
129.2787	168.2546	189.1527	231.5747

## 5.6 问题二模型的建立

问题二给出了各温区温度的设定值，其中小温区 1 5 为  $182^{\circ}\text{C}$ ，小温区 6 为  $203^{\circ}\text{C}$ ，小温区 7 为  $237^{\circ}\text{C}$ 、小温区 8-9 为  $254^{\circ}\text{C}$ ，需要求解在满足制程界限的前提下，允许的最大传送带过炉速度。以制程界限、热传导方程、空气温度随时间的变化函数、温区温度设定值为约束条件，以最大化传送带过炉速度为目标，建立单目标规划模型：

$$\begin{aligned}
& \max v \\
& \left\{ \begin{array}{l} \text{制程界限} \left\{ \begin{array}{l} \max(|\dot{u}(t)|) \leq 3 \\ 60 \leq t | (150 \leq u(t) \leq 190 \& \dot{u}(t) > 0) \leq 120 \\ 40 \leq t | (217 \leq u(t)) \leq 90 \\ 240 \leq \max(u(t)) \leq 250 \\ 65 < v < 100 \end{array} \right. \\ \\ \text{控制方程} \left\{ \begin{array}{l} u(t) = u(\frac{d}{2}, t) \\ \frac{\partial u(x,t)}{\partial t} = \alpha \frac{\partial^2 u(x,t)}{\partial x^2} \\ \frac{\partial u}{\partial x} \Big|_{x=d} = \beta (u_s(t) - u(d, t)) \\ - \frac{\partial u}{\partial x} \Big|_{x=0} = -\beta (u_s(t) - u(0, t)) \end{array} \right. \\ \\ \text{空气温度 } u_s(t) = U_s = U_s(vt) \\ \\ \text{温区温度} \left\{ \begin{array}{l} T_{15} = 182 \\ T_{06} = 203 \\ T_{07} = 237 \\ T_{89} = 254 \\ T_{1011} = 25 \end{array} \right. \end{array} \right. \quad (17)
\end{aligned}$$

### 5.7 变范围搜索策略

首先以  $65\text{cm/min}$  为初值,  $1\text{cm/min}$  为步长,  $100\text{cm/min}$  为终值, 将各个参数代入模型求解炉温曲线, 作出各制程的变化情况如图5。

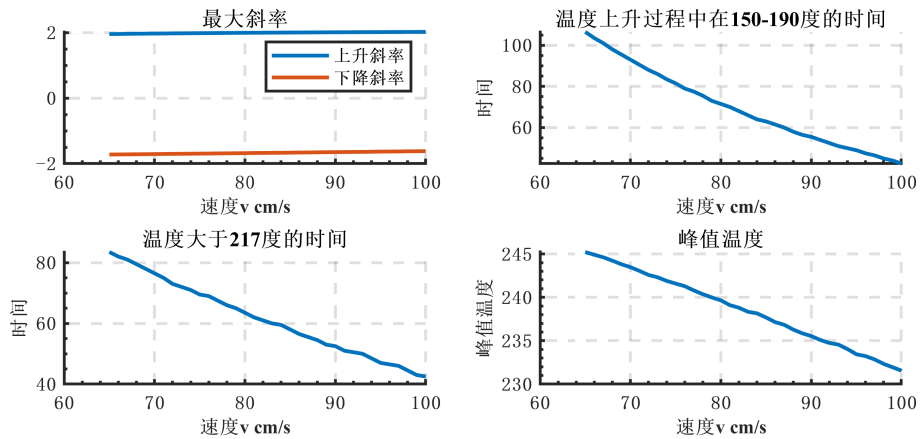


图 5 问题 2 各个指标随速度的变化

由图5可以看出, 速度在  $65 - 100\text{cm}/\text{min}$  时, 斜率要求和温度大于  $217$  度的要求都满足, 速度上升的过程中, 温度上升过程中在  $150-190$  度的时间要求和峰值温度要求逐渐不满足, 是速度的单调函数。由满足跳变到不满足时的速度为  $78\text{cm}/\text{min}$ , 因此, 缩小步长, 以  $0.01\text{cm}/\text{min}$  的步长遍历  $77 - 79\text{cm}/\text{min}$  的速度, 最终求得最大速度为  $78.95\text{cm}/\text{min}$ 。

### 5.8 问题 3 模型的建立

问题 3 需要求解  $[T_{15}, T_{06}, T_{07}, T_{89}, v]$ , 使得超过  $217^\circ\text{C}$  到峰值温度所覆盖的面积最小, 并且满足制程及温度变化约束 (上下波动  $10^\circ\text{C}$ )。面积可以通过积分求得, 因此建立以面积最小化为目标的单目标规划模型:

$$\begin{aligned}
 & [\hat{T}_{15}, \hat{T}_{06}, \hat{T}_{07}, \hat{T}_{89}, \hat{v}] = \arg \min_{T_{15}, T_{06}, T_{07}, T_{89}, v} S = \int_{t'}^{t''} [u(t) - 217] dt \\
 & \left\{ \begin{array}{l}
 t' = \min \{t \mid u(t) \geq 217\} \\
 t'' = \arg \max_t u(t) \\
 \text{制程界限} \left\{ \begin{array}{l}
 \max(|\dot{u}(t)|) \leq 3 \\
 60 \leq t \mid (150 \leq u(t) \leq 190 \& \dot{u}(t) > 0) \leq 120 \\
 40 \leq t \mid (217 \leq u(t)) \leq 90 \\
 240 \leq \max(u(t)) \leq 250 \\
 65 < v < 100
 \end{array} \right. \\
 \text{控制方程} \left\{ \begin{array}{l}
 u(t) = u\left(\frac{d}{2}, t\right) \\
 \frac{\partial u(x, t)}{\partial t} = \alpha \frac{\partial^2 u(x, t)}{\partial x^2} \\
 \frac{\partial u}{\partial x} \Big|_{x=d} = \beta (u_s(t) - u(d, t)) \\
 -\frac{\partial u}{\partial x} \Big|_{x=0} = -\beta (u_s(t) - u(0, t))
 \end{array} \right. \\
 \text{空气温度: } u_s(t) = U_s = U_s(vt) \\
 \text{速度界限: } 65\text{cm}/\text{min} \leq v \leq 100\text{cm}/\text{min} \\
 \text{温区温度界限} \left\{ \begin{array}{l}
 |T_{15} - 175| \leq 10 \\
 |T_{06} - 195| \leq 10 \\
 |T_{07} - 235| \leq 10 \\
 |T_{89} - 255| \leq 10 \\
 T_{1011} = 25
 \end{array} \right.
 \end{array} \right. \quad (18)
 \end{aligned}$$

在求解面积  $S$  时，可以通过梯形法求解：

$$S = \sum_{i=i'}^{i''-1} \frac{\Delta t}{2} [u((i-1)\Delta t) + u(i\Delta t) - 2u(i')] \quad (19)$$

其中

$$i' = \left\lfloor \frac{t'}{\Delta t} \right\rfloor + 1, \quad i'' = \left\lfloor \frac{t''}{\Delta t} \right\rfloor + 1 \quad (20)$$

### 5.9 遗传算法求解问题 3 单目标规划

问题 3 是一个复杂的非线性规划问题，搜索变量有 5 个，分别是  $[T_{15}, T_{06}, T_{07}, T_{89}, v]$ ，如果使用遍历的方法，花费的时间十分巨大。因此本文考虑使用遗传算法，以面积作为遗传算法的适应度函数，并使用罚函数法，将不满足温区温度界限的各温区温度、不满足速度界限的过炉速度、以及不满足制程界限的参数组合的适应度设置为一个很大的数（例如 10000）。

为减少遗传算法的不确定性，增加说服力，多次使用遗传算法求解问题 3，得到多组结果，如表3. 其中某一次的算法收敛过程如图6.

表 3 问题 3 多组遗传算法求解结果

组别	$T_{15}(^{\circ}C)$	$T_{06}(^{\circ}C)$	$T_{07}(^{\circ}C)$	$T_{89}(^{\circ}C)$	$v(cm/min)$	面积 $S(^{\circ}C \cdot s)$	峰值温度 $(^{\circ}C)$
1	184.1404	194.4357	240.5300	262.6221	94.1413	509.0461	240.0
2	182.1140	189.8592	230.2581	264.9846	91.5834	488.2102	240.0
3	177.9146	186.0785	230.9235	260.8567	84.2747	533.7085	240.0
4	183.1857	195.0097	233.7218	265.0000	94.7943	488.7792	240.0
5	179.8122	185.5750	239.1417	264.7966	94.0406	490.6426	240.0
6	174.3615	189.2047	236.8863	261.4947	87.7588	524.3180	240.0
7	170.3522	188.8334	226.1385	264.7179	86.5511	490.2605	240.0
8	173.7903	190.1923	229.6501	264.8884	89.3426	488.8328	240.0
9	181.8678	195.3838	231.2184	265.0000	92.8019	488.4086	240.0
10	174.0087	186.5502	232.0147	264.9075	89.8245	488.6447	240.0

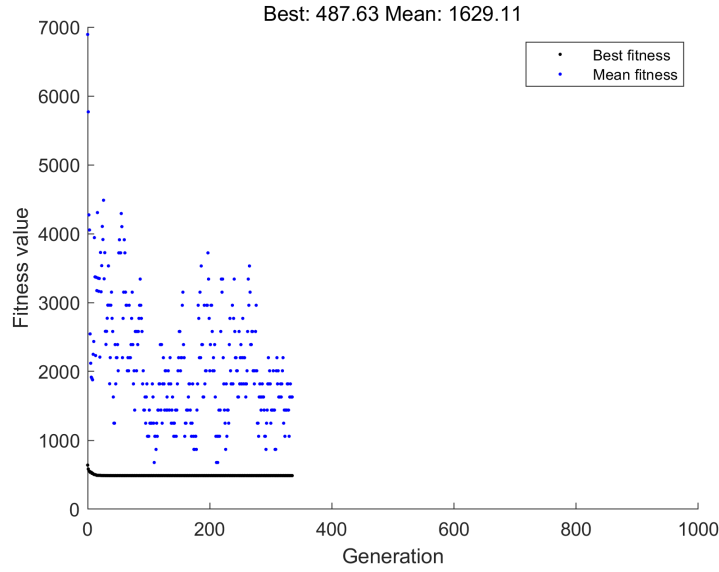


图 6 遗传算法过程图

可以看到，10 次运行结果的峰值温度都是  $240^{\circ}\text{C}$ ，即最小允许的峰值温度。因为峰值温度越大，面积相应地就会越大，为了得到最小的面积，因此需要峰值温度最小，模型的结果和我们的判断相符合。取面积最小的一次结果，即第 2 组的各个参数，作为问题 3 求解的最终参数，此时的面积指标为  $488.2102(^{\circ}\text{C} \cdot \text{s})$ 。

表 4 问题 3 最佳参数

$T_{15}(^{\circ}\text{C})$	$T_{06}(^{\circ}\text{C})$	$T_{07}(^{\circ}\text{C})$	$T_{89}(^{\circ}\text{C})$	$v(\text{cm}/\text{min})$
182.1140	189.8592	230.2581	264.9846	91.5834

将最佳参数代入模型求解炉温曲线，如图7。

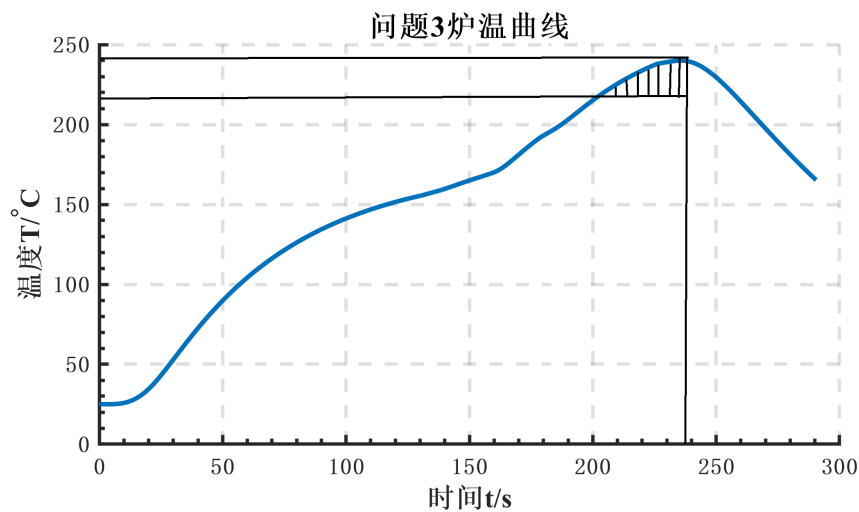


图 7 问题 3 炉温曲线

### 5.10 问题 4 模型的建立

$t' = \min \{t \mid u(t) \geq 217\}$  为第一次到达 217 度的时刻,  $t'' = \arg \max_t u(t)$  为峰值温度对应的时刻

定义  $N = \frac{t''-t'}{\Delta t}$  为求和个数, 对于大于  $t'$  小于  $t''$  的时刻点  $t' + i\Delta t$ , 关于  $t''$  的对称点为  $2t'' - i\Delta t$ , 两者作差并求和, 得到关于对称性的指标  $D$ :

$$D = \sqrt{\frac{1}{N} \sum_{i=0}^N |u(2t'' - (t' + i\Delta t)) - u(t' + i\Delta t)|^2} \quad (21)$$

$D$  越小, 对称性越大。因此以面积  $D$  最小化和对称性指标  $D$  最小化为目标建立多目标规划模型:

$$\begin{aligned} [\hat{T}_{15}, \hat{T}_{06}, \hat{T}_{07}, \hat{T}_{89}, \hat{v}] &= \arg \min_{T_{15}, T_{06}, T_{07}, T_{89}, v} S = \int_{t'}^{t''} [u(t) - 217] dt \\ [\hat{T}_{15}, \hat{T}_{06}, \hat{T}_{07}, \hat{T}_{89}, \hat{v}] &= \arg \min_{T_{15}, T_{06}, T_{07}, T_{89}, v} D = \sqrt{\frac{1}{N} \sum_{i=0}^N |u(2t'' - (t' + i\Delta t)) - u(t' + i\Delta t)|^2} \\ \left\{ \begin{array}{l} t' = \min \{t \mid u(t) \geq 217\} \\ t'' = \arg \max_t u(t) \\ N = \frac{t''-t'}{\Delta t} \\ \text{制程界限} \left\{ \begin{array}{l} \max(|\dot{u}(t)|) \leq 3 \\ 60 \leq t \mid (150 \leq Nu(t) \leq 190 \& \dot{u}(t) > 0) \leq 120 \\ 40 \leq t \mid (217 \leq u(t)) \leq 90 \\ 240 \leq \max(u(t)) \leq 250, 65 < v < 100 \end{array} \right. \\ \text{控制方程} \left\{ \begin{array}{l} u(t) = u\left(\frac{d}{2}, t\right) \\ \frac{\partial u(x,t)}{\partial t} = \alpha \frac{\partial^2 u(x,t)}{\partial x^2} \\ \frac{\partial u}{\partial x} \Big|_{x=d} = \beta(u_s(t) - u(d,t)) \\ -\frac{\partial u}{\partial x} \Big|_{x=0} = -\beta(u_s(t) - u(0,t)) \end{array} \right. \\ \text{空气温度: } u_s(t) = U_s = U_s(vt) \\ \text{速度界限: } 65 \text{cm/min} \leq v \leq 100 \text{cm/min} \\ \text{温区温度界限} \left\{ \begin{array}{l} |T_{15} - 175| \leq 10, |T_{06} - 195| \leq 10 \\ |T_{07} - 235| \leq 10, |T_{89} - 255| \leq 10 \\ T_{1011} = 25 \end{array} \right. \end{array} \right. \quad (22)$$

### 5.11 分层序列法

本文题中的模型有多个目标, 由于同时处理多个目标较为麻烦, 因此采用运筹学中的经典方法——分层序列法进行决策.

分层序列法的主体思想是根据各目标的重要程度依次排序, 分为最重要目标、次要目标等, 对每个目标的最优解, 并以最优解的解集作为下一级目标的可行域, 再进行下一级目标的最优化, 直到所有目标优化完毕. 具体的算法过程如下:

设多个优化目标的优先级序列为:  $f_1(x), f_2(x), \dots, f_m(x)$  首先对第一优先级目标  $f_1(x)$  求最优解, 并在该最优解集合  $R_1$  中求解第二优先级目标  $f_2(x)$  的最优解, 以此类方式逐个求解, 直至最后求  $f_m(x)$  最优解结束, 即为多目标问题的最优解集合  $R_m$ . 其有解的前提是  $R_0, R_1, \dots, R_{m-1}$  非空, 且  $R_0, R_1, \dots, R_{m-2}$  至少有两个元素. 相对于常规的赋权优化方法, 分层序列法性能优越, 且每步都有较为合适的现实含义与决策背景, 便于解决问题时与实际结合, 是一种有效的决策方法.

### 5.12 使用结合分层序列法的遗传算法求解问题 4

对于问题 4, 可以使用结合分层序列法思想的遗传算法. 步骤如下:

**Step1:** 同问题 3 的求解过程, 使用面积指标  $S$  作为遗传算法的适应度函数, 对不满足条件的参数赋以较大的适应度, 求解若干代后保存最后的种群;

**Step2:** 以步骤一中保存的种群为初始种群, 使用对称性指标  $D$  作为遗传算法的适应度函数, 使用罚函数法对不满足条件的参数赋以较大的适应度, 同时求解面积指标  $S$ . 在问题 3 中, 求得的最小面积为  $488.2102(^{\circ}C \cdot s)$ , 这里放宽限制, 对  $S > 505^{\circ}C \cdot s$ , 本文认为其面积过大, 同样使用罚函数法将这样的参数组合赋以较大的适应度, 使其在进化的过程中被抛弃, 对于面积  $S < 505^{\circ}C \cdot s$  的参数组合, 则留下参与进化.

### 5.13 问题 4 求解结果

同问题 3, 为减少遗传算法的不确定性, 增加说服力, 多次使用遗传算法求解问题 4, 得到多组结果, 如表 5. 各组参数的两指标散点图如图 8.

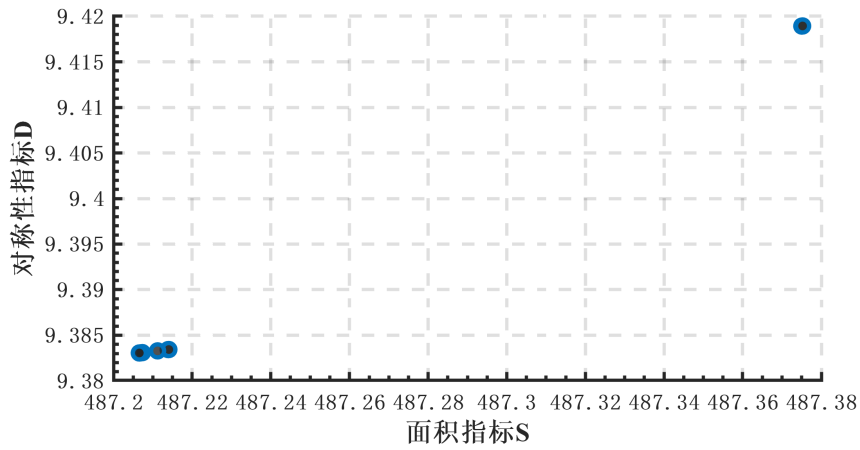


图 8 问题 4 两指标散点图

表 5 问题 4 遗传算法多次求解结果

组别	$T_{15}(^{\circ}C)$	$T_{06}(^{\circ}C)$	$T_{07}(^{\circ}C)$	$T_{89}(^{\circ}C)$	$v(cm/min)$	面积指标 $S(^{\circ}C \cdot s)$	对称性指 标 ( $^{\circ}C$ )
1	172.9719	185.2241	230.2076	265	88.7127	487.2139	9.3834
2	172.3227	185.1239	230.591	265	88.7126	487.2074	9.3831
3	172.3448	185.027	230.6251	265	88.7125	487.2068	9.3831
4	172.3717	185.0462	230.6021	265	88.7125	487.2072	9.3831
5	172.8574	185.0523	231.9734	265	88.9720	487.3750	9.4189
6	172.8416	185.0936	231.9621	265	88.9720	487.3752	9.4189
7	172.3051	185.0758	230.6227	265	88.7127	487.2068	9.3831
8	172.5251	185.748	230.1928	265	88.7125	487.2141	9.3834
9	172.5383	185.3709	230.3632	265	88.7125	487.2112	9.3833
10	172.3431	185.0034	230.637	265	88.7126	487.2066	9.3831

由于两个目标函数都是最小化目标函数，因此可以取散点图中最靠近原点的参数组合，即两个目标函数值都最小的参数组合，显然，第 10 组参数组合两目标函数都是最小，因此，取第 10 组参数作为最佳参数，代入模型求解炉温曲线，并与实验数据对比，如图9, 显然，本文模型求解的炉温曲线与实验数据相比，具有很好的对称性。

综上，问题 4 最佳参数组合如下，其中面积为  $487.2066S(^{\circ}C \cdot s)$ ，对称性指标为  $9.3831(^{\circ}C)$ 。



$T_{15}(^{\circ}C)$	$T_{06}(^{\circ}C)$	$T_{07}(^{\circ}C)$	$T_{89}(^{\circ}C)$	$v(cm/min)$
172.3431	185.0034	230.637	265	88.7126

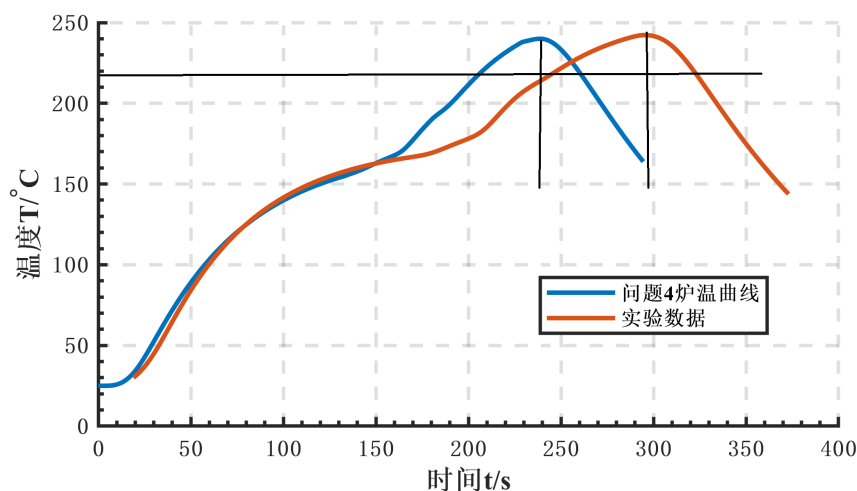


图 9 问题 4 炉温曲线对称性与实验数据对比

## 六、灵敏性分析

对于问题 3，本文求得了最佳温区温度和最佳速度，考虑到实际需要控制的温度十分严格，精度需要达到  $0.1^{\circ}C$ ，因此，对小温区 1-5 的温度  $T_{15}$ ，小温区 6 的温度  $T_{06}$ ，小温区 7 的温度  $T_{07}$ ，小温区 8、9 的温度  $T_{89}$  在最佳温度附近施加以  $0.1^{\circ}C$  为步长的正负变化，并计算相应的面积  $S$  的改变量，如图10

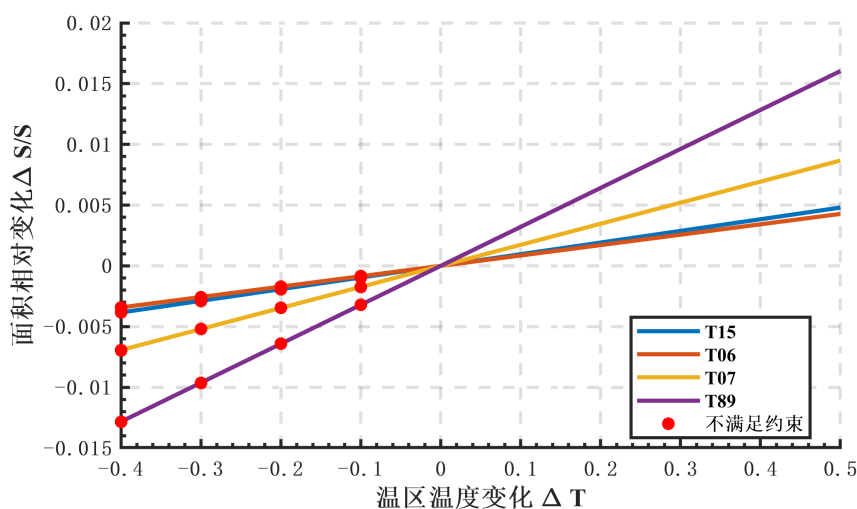


图 10 灵敏度分析

可以看到，改变各温区温度值，面积  $S$  的改变量在  $-1.5\% - 2\%$  之间。当温度上升时，面积变大，当温度下降时，面积变小但是不满足制程约束。可见面积对各温区温度

较敏感，敏感程度由大到小依次是  $T_{89}, T_{07}, T_{15}, T_{06}$ 。在实际生产过程中，应该严格控制温度。

## 七、模型的评价

### 7.1 模型的优点

- 对回流焊炉的热参数  $\alpha$  分段讨论，能够较准确地拟合实验数据，对炉温曲线的预测也更加精准；
- 对模型热参数  $\alpha$  和  $\beta$  的确定以及问题 2 的求解过程中，使用了变范围搜索的方法，减少了计算量，更快速地得到结果；
- 对问题 3、4 使用遗传算法，并结合分层序列法的思想，避免陷入局部最优，较好地解决了问题。

### 7.2 模型的缺点

- 未考虑热回收等对炉温曲线的影响；
- 问题 3、4 的求解时间较长。

## 参考文献

- [1] 冯志刚, 郁鼎文, 朱云鹤. 回流焊工艺参数对温度曲线的影响[J]. 电子工艺技术, 2004(06):243-246+251.
- [2] 姜海峡. 关于回流焊接温度曲线设置的研究[J]. 新技术新工艺, 2019(08):64-67.
- [3] SMT 回流焊炉的炉温工艺曲线设置[J]. 电子报, 2018:009.
- [4] 万学斌. 回流焊接温度曲线的拟合研究[J]. 河南科技, 2018(32):53-54.

## 附录 A 代码文件列表

文件名	功能描述
Data.mat	附件数据
getTs.m	获取空气温度分布向量
getTt.m	获取炉温曲线向量
find_alpha_beta_ga.m	寻找最佳参数值
problem1.m	问题 1 求解
problem2.m	问题 2 求解
problem3.m	问题 3 求解
problem4.m	问题 4 求解
condition.m	根据温度向量判断是否满足制程约束
evaluateS.m	计算面积指标
evaluateD.m	计算对称性指标
linminxin.m	灵敏性分析
alpha_beta1.mat	最佳热传导模型参数
problem3_Data.mat	问题 3 求解结果
problem4_Data.mat	问题 4 求解结果

## 附录 B 代码

getTs.m

```

1 function ts = getTs()
2 global T15 T06 T07 T89 T1011 v deltat l a b
3 l=30.5;a=5;b=25;
4 %t=0:deltat:(11*l+10*a)/v;
5 len0=floor((b)/(v*deltat));
6 len1=floor((b+5*l+4*a)/(v*deltat));
7 len2=floor((b+5*l+5*a)/(v*deltat));
8 len3=floor((b+6*l+5*a)/(v*deltat));
9 len4=floor((b+6*l+6*a)/(v*deltat));
10 len5=floor((b+7*l+6*a)/(v*deltat));
11 len6=floor((b+7*l+7*a)/(v*deltat));
12 len7=floor((b+9*l+8*a)/(v*deltat));
13 len8=floor((b+9*l+9*a)/(v*deltat));
14 len9=floor((b+11*l+10*a+25)/(v*deltat));
15 ts(1:len0)=(T15-25)/(25/v)*deltat*[1:len0]+25;
16 ts(len0+1:len1)=T15;

```

```

17 ts(len1+1:len2)=(T06-T15)/(a/v)*deltat*[1:(len2-len1)]+T15;%429
18 ts(len2+1:len3)=T06;
19 ts(len3+1:len4)=(T07-T06)/(a/v)*deltat*[1:(len4-len3)]+T06;
20 ts(len4+1:len5)=T07;
21 ts(len5+1:len6)=(T89-T07)/(a/v)*deltat*[1:(len6-len5)]+T07;
22 ts(len6+1:len7)=T89;
23 ts(len7+1:len8)=(T1011-T89)/(a/v)*deltat*[1:(len8-len7)]+T89;
24 ts(len8+1:len9+1)=T1011;

```

## getTt.m

```

1 function T = getTt(alpha,beta)
2 global T15 T06 T07 T89 T1011 v deltax
3 global ts l a b d deltax
4 global T_real index
5 global Data
6 global len0 len1 len2 len3 len4 len5
7 size_x=floor(d/deltax);
8 size_t=len5;
9 Tt=zeros(size_x,size_t);
10 Tt(:,1)=25;
11 A=cell(1,5);B=cell(1,5);C=cell(1,5);
12 i=1;
13 for r=alpha*deltat/(deltax^2)
14     A{i}=diag([1+beta*deltax,2*ones(1,size_x-2)*(1+r),1+beta*deltax]);
15     A{i}=A{i}+diag([-1,-r*ones(1,size_x-2)],1);
16     A{i}=A{i}+diag([-r*ones(1,size_x-2),-1],-1);
17
18     B{i}=diag([0,2*ones(1,size_x-2)*(1-r),0]);
19     B{i}=B{i}+diag([0,r*ones(1,size_x-2)],1);
20     B{i}=B{i}+diag([r*ones(1,size_x-2),0],-1);
21     C{i}=A{i}\B{i};
22     i=i+1;
23 end
24 for i=1:len1-1
25     c=zeros(size_x,1); c(1)=beta*deltax*ts(i); c(size_x)=beta*deltax*ts(i); c=A{1}\
26     c;
27     Tt(:,i+1)=C{1}*Tt(:,i)+c;
28 end
29 for i=len1:len2-1
30     c=zeros(size_x,1); c(1)=beta*deltax*ts(i); c(size_x)=beta*deltax*ts(i); c=A{2}\
31     c;
32     Tt(:,i+1)=C{2}*Tt(:,i)+c;
33 end
34 for i=len2:len3-1
35     c=zeros(size_x,1); c(1)=beta*deltax*ts(i); c(size_x)=beta*deltax*ts(i); c=A{3}\
36     c;
37     Tt(:,i+1)=C{3}*Tt(:,i)+c;

```

```

35 end
36 for i=len3:len4-1
37     c=zeros(size_x,1); c(1)=beta*deltax*ts(i); c(size_x)=beta*deltax*ts(i); c=A{4}\
    c;
38     Tt(:,i+1)=C{4}*Tt(:,i)+c;
39 end
40 for i=len4:len5
41     c=zeros(size_x,1); c(1)=beta*deltax*ts(i); c(size_x)=beta*deltax*ts(i); c=A{5}\
    c;
42     Tt(:,i+1)=C{5}*Tt(:,i)+c;
43 end
44 T=Tt(floor(size_x/2),:);

```

### find\_alpha\_beta\_ga.m

```

1  clc,clear,close all
2  global T15 T06 T07 T89 T1011 v deltat
3  global ts l a b d deltax
4  global T_real index
5  global Data
6  global len0 len1 len2 len3 len4 len5
7  load Data
8  T15=175;T06=195;T07=235;T89=255;T1011=25;v=70/60;l=30.5;a=5;b=25;deltat=0.5;d
    =0.015;deltax=0.001;
9  ts=getTs();
10 t=0:deltat:(b+11*l+10*a+25)/v;
11 figure('Position',[483.4,390.6,560,299])
12 plot(t,ts),xlabel('时间/s'),ylabel('空气温度{ }^{\circ} C')
13 beautiplot
14 exportgraphics(gcf,'img\空气温度趋势图.png','Resolution',400)
15 %%
16 index=Data(:,1)/deltat+1;
17 T_real = Data(:,2);
18 len0=floor((b)/(v*deltat));
19 len1=floor((b+5*l+4*a)/(v*deltat));
20 len2=floor((b+6*l+5*a)/(v*deltat));
21 len3=floor((b+7*l+6*a)/(v*deltat));
22 len4=floor((b+9*l+8*a)/(v*deltat));
23 len5=floor((b+11*l+10*a+25)/(v*deltat));
24 options = optimoptions(@ga,...
25     'Display','off','Generations', 150, 'StallGenLimit', 200, 'PlotFcns', '
    gaplotbestf');%,'Migrationfraction', 0.3);
26 number=6;
27 [alpha_beta,objvalue]=ga(@problem0_fitness ,number,[],[],[],[],[zeros(1,number-1),1
    e+4],[ones(1,number-1)*(1e-6),1e+5],[[],options]);
28 %%
29 load alpha_beta1.mat
30 t=0:deltat:(b+11*l+10*a+25)/v;

```

```

31 Tt_M=getTt(alpha_beta(1:5),alpha_beta(6));
32 figure('Position',[276.2,356.2,709,307.2])
33 Dt1=problem0_fitness(alpha_beta)
34 plot(t,Tt_M);hold on
35 plot(Data(:,1),Data(:,2));xlabel('时间t/s');ylabel('温度T/{\circ}C')
36 legend('拟合曲线','实际曲线','Location','best');beautiplot
37 exportgraphics(gcf,'img\拟合.png','Resolution',400)
38 [R,P]=corrcoef(T_real,Tt_M(index));
39 R2=R(1,2)
40
41
42 t=0:deltat:(b+11*l+10*a+25)/v;
43 % Tt_M1=getTt([4 5 7 4 2]*(1e-7),14.458e-4/1.67e-8);
44 Tt_M1=getTt([3.9 5.1 6.5 4.7 2.1]*(1e-7),9000);
45 Dt2=problem0_fitness([3.9 5.1 6.5 4.7 2.1]*(1e-7),9000)
46 % Tt_M=getTt([4.437 5.621 7.449 4.997 2.401]*(1e-7),14.458e-4/1.67e-8);
47 figure('Position',[276.2,356.2,709,307.2])
48 plot(t,Tt_M1);hold on
49 plot(Data(:,1),Data(:,2));xlabel('时间t/s');ylabel('温度T/{\circ}C')
50 legend('拟合曲线1','实际曲线','Location','best');beautiplot

```

#### problem1.m

```

1 clc,clear,close all
2 global T15 T06 T07 T89 T1011 v deltat
3 global ts l a b d deltax
4 global T_real index
5 global Data
6 global len0 len1 len2 len3 len4 len5
7 load Data
8 T15=173;T06=198;T07=230;T89=257;T1011=25;v=78/60;l=30.5;a=5;b=25;deltat=0.5;d
   =0.015;deltax=0.001;
9 ts=getTs();
10 index=Data(:,1)/deltat+1;
11 T_real = Data(:,2);
12 len0=floor((b)/(v*deltat));
13 len1=floor((b+5*l+4*a)/(v*deltat));
14 len2=floor((b+6*l+5*a)/(v*deltat));
15 len3=floor((b+7*l+6*a)/(v*deltat));
16 len4=floor((b+9*l+8*a)/(v*deltat));
17 len5=floor((b+11*l+10*a+25)/(v*deltat));
18 load alpha_beta1.mat
19 t=0:deltat:(b+11*l+10*a+25)/v;
20 T_problem1=getTt(alpha_beta(1:5),alpha_beta(6));
21 figure('Position',[483.4,390.6,560,299])
22 plot(t,T_problem1);xlabel('时间t/s');ylabel('温度T/{\circ}C');title('问题1炉温曲线
   ');beautiplot
23 exportgraphics(gcf,'img\问题1炉温曲线.png','Resolution',400)

```

```

24 t_wenqv3=(b+2*l+2*a+1/2)/v;
25 t_wenqv6=(b+5*l+5*a+1/2)/v;
26 t_wenqv7=(b+6*l+6*a+1/2)/v;
27 t_wenqv8=(b+8*l+7*a+1/2)/v;
28 T_wenqv3=(T_problem1(floor(t_wenqv3/deltat)+1)+T_problem1(floor(t_wenqv3/deltat)+2)
    )/2;
29 T_wenqv6=(T_problem1(floor(t_wenqv6/deltat)+1)+T_problem1(floor(t_wenqv6/deltat)+2)
    )/2;
30 T_wenqv7=(T_problem1(floor(t_wenqv7/deltat)+1)+T_problem1(floor(t_wenqv7/deltat)+2)
    )/2;
31 T_wenqv8=(T_problem1(floor(t_wenqv8/deltat)+1)+T_problem1(floor(t_wenqv8/deltat)+2)
    )/2;
32 t_T_problem1_wenqv=table([T_wenqv3,T_wenqv6,T_wenqv7,T_wenqv8]);
33 writetable(t_T_problem1_wenqv,'温区点温度.csv')
34 t_T_problem1=table([t',T_problem1']);
35 writetable(t_T_problem1,'result.csv')
36 %%
37 figure('Position',[483.4,390.6,560,299])
38 plot()

```

## problem2.m

```

1  clc,clear,close all
2  global T15 T06 T07 T89 T1011 v deltat
3  global ts l a b d deltax
4  global T_real index
5  global Data
6  global len0 len1 len2 len3 len4 len5
7  load Data
8  T15=173;T06=198;T07=230;T89=257;T1011=25;l=30.5;a=5;b=25;deltat=0.5;d=0.015;deltax
    =0.001;
9  index=Data(:,1)/deltat+1;
10 T_real = Data(:,2);
11 max_up_xielv=zeros(1,length(65:1:100));
12 max_down_xielv=zeros(1,length(65:1:100));
13 t_between150_190_all=zeros(1,length(65:1:100));
14 t_up217_all=zeros(1,length(65:1:100));
15 maxT_all=zeros(1,length(65:1:100));
16 for vi=65:1:100
17     v=vi/60;
18     ts=getTs();
19     len0=floor((b)/(v*deltat));
20     len1=floor((b+5*l+4*a)/(v*deltat));
21     len2=floor((b+6*l+5*a)/(v*deltat));
22     len3=floor((b+7*l+6*a)/(v*deltat));
23     len4=floor((b+9*l+8*a)/(v*deltat));
24     len5=floor((b+11*l+10*a+25)/(v*deltat));
25     load alpha_beta1.mat

```

```

26     t=0:deltat:(b+11*l+10*a+25)/v;
27     T_problem2=getTt(alpha_beta(1:5),alpha_beta(6));
28     [isOk,up_xielv,down_xielv,t_between150_190,t_up217,maxT]=condition(T_problem2);
29     max_up_xielv(vi-64)=max(up_xielv);
30     max_down_xielv(vi-64)=min(down_xielv);
31     t_between150_190_all(vi-64)=t_between150_190;
32     t_up217_all(vi-64)=t_up217;
33     maxT_all(vi-64)=maxT;
34     if isOK
35         vmax=v*60;%vmax=78
36     end
37 end
38 vi=65:1:100;
39 figure('Position',[97.8,342.6,762.4,327.2])
40 subplot(221)
41 plot(vi,max_up_xielv,vi,max_down_xielv),xlabel('速度v cm/s'),title('最大斜率')
42 legend('上升斜率','下降斜率','Orientation','vertical')
43 subplot(222)
44 plot(vi,t_between150_190_all),xlabel('速度v cm/s'),ylabel('时间'),title('温度上升过程中在150-190度的时间')
45 subplot(223)
46 plot(vi,t_up217_all),xlabel('速度v cm/s'),ylabel('时间'),title('温度大于217度的时间')
47 subplot(224)
48 plot(vi,maxT_all),xlabel('速度v cm/s'),ylabel('峰值温度');title('峰值温度'),
    beautiplot('small')
49 exportgraphics(gcf,'img\问题2各个指标随速度的变化.png','Resolution',400)
50 %%
51 for vi=77:0.01:79
52     v=vi/60;
53     ts=getTs();
54     len0=floor((b)/(v*deltat));
55     len1=floor((b+5*l+4*a)/(v*deltat));
56     len2=floor((b+6*l+5*a)/(v*deltat));
57     len3=floor((b+7*l+6*a)/(v*deltat));
58     len4=floor((b+9*l+8*a)/(v*deltat));
59     len5=floor((b+11*l+10*a+25)/(v*deltat));
60     load alpha_beta1.mat
61     t=0:deltat:(b+11*l+10*a+25)/v;
62     T_problem2=getTt(alpha_beta(1:5),alpha_beta(6));
63     [isOk,up_xielv,down_xielv,t_between150_190,t_up217,maxT]=condition(T_problem2);
64     if isOK
65         vmax=v*60;%vmax=78.95
66     end
67 end
68 vmax

```



### problem3.m

```

1  clc,clear, close all
2  global T15 T06 T07 T89 T1011 v deltat
3  global ts l a b d deltax
4  global len0 len1 len2 len3 len4 len5
5  global alpha_beta
6  load alpha_beta1.mat
7  T1011=25;l=30.5;a=5;b=25;deltat=0.5;d=0.015;deltax=0.001;
8  lb=[165,185,225,245,65];
9  ub=[185,205,245,265,100];
10 options = optimoptions(@ga,...
11     'Display','off','Generations', 1000, 'StallGenLimit', 300,'Migrationfraction',
12     0.3);%'PlotFcns', 'gaplotbestf'
13 number=5;
14 problem3_Data=[];
15 for i=1:10
16     [T_all,Fit]=ga(@evaluates ,number,[],[],[],[],lb,ub,[],options);
17     T_all
18     T15=T_all(1);T06=T_all(2);T07=T_all(3);T89=T_all(4);v=T_all(5)/60;
19     ts=getTs();
20     len0=floor((b)/(v*deltat));
21     len1=floor((b+5*l+4*a)/(v*deltat));
22     len2=floor((b+6*l+5*a)/(v*deltat));
23     len3=floor((b+7*l+6*a)/(v*deltat));
24     len4=floor((b+9*l+8*a)/(v*deltat));
25     len5=floor((b+11*l+10*a+25)/(v*deltat));
26     S_problem3=Fit;
27     T_problem3=getTt(alpha_beta(1:5),alpha_beta(6));
28     T_max_problem3=max(T_problem3)
29
30     tmp=[T_all,S_problem3,T_max_problem3];
31     problem3_Data=[problem3_Data;tmp];
32 end
33 save problem3_Data
34 %%
35 t=0:deltat:(b+11*l+10*a+25)/v;
36 figure('Position',[483.4,390.6,560,299])
37 plot(t,T_problem3)
38 xlabel('时间t/s');ylabel('温度T/{\circ}C');title('问题3炉温曲线');beautiplot
exportgraphics(gcf,'img\问题3炉温曲线.png','Resolution',400)

```

### problem4.m

```

1  clc,clear,close all
2  global T15 T06 T07 T89 T1011 v deltat
3  global ts l a b d deltax
4  global len0 len1 len2 len3 len4 len5
5  global alpha_beta

```

```

6 load alpha_beta1.mat
7 T1011=25;l=30.5;a=5;b=25;deltat=0.5;d=0.015;deltax=0.001;
8 lb=[165,185,225,245,65];
9 ub=[185,205,245,265,100];
10 options = optimoptions(@ga,...
11     'Display','off','Generations', 200, 'StallGenLimit', 100, 'PlotFcns', '
    gaplotbestf');
12 number=5;
13 problem4_Data=[];
14 for i=1:10
15     clear pop_S
16     [~,~,~,~,pop_S]=ga(@evaluateS ,number,[],[],[],[],lb,ub,[],options);
17     options = optimoptions(@ga,...
18         'Display','off','Generations', 200, 'StallGenLimit', 100,'InitialPopulation
        ',pop_S, 'PlotFcns', 'gaplotbestf');
19
20     [T_all,FitD,~,~,pop_D]=ga(@evaluateD ,number,[],[],[],[],lb,ub,[],options);
21     T15=T_all(1);T06=T_all(2);T07=T_all(3);T89=T_all(4);v=T_all(5)/60;
22     ts=getTs();
23     len0=floor((b)/(v*deltat));
24     len1=floor((b+5*l+4*a)/(v*deltat));
25     len2=floor((b+6*l+5*a)/(v*deltat));
26     len3=floor((b+7*l+6*a)/(v*deltat));
27     len4=floor((b+9*l+8*a)/(v*deltat));
28     len5=floor((b+11*l+10*a+25)/(v*deltat));
29     S_problem4=evaluateS(T_all);
30     D_problem4=FitD;
31     T_problem4=getTt(alpha_beta(1:5),alpha_beta(6));
32     tmp=[T_all,S_problem4,D_problem4];
33     problem4_Data=[problem4_Data;tmp];
34 end
35 save problem4_Data
36
37 %%
38 figure('Position',[407.4,348.2,560,285.6])
39 scatter(problem4_Data(:,6),problem4_Data(:,7));xlabel('面积指标S');ylabel('对称性指
    标D')
40 beautiplot
41 exportgraphics(gcf,'img\问题4两指标散点图.png','Resolution',400)
42 %%
43 t=0:deltat:(b+11*l+10*a+25)/v;
44 figure('Position',[483.4,390.6,560,299])
45 plot(t,T_problem4),hold on
46 xlabel('时间t/s');ylabel('温度T/{\circ}C');
47 load Data.mat
48 plot(Data(:,1),Data(:,2));legend('问题4炉温曲线','实验数据')
49 beautiplot

```

```
50 exportgraphics(gcf,'img\问题4炉温曲线.png','Resolution',400)
```

#### condition.m

```
1 function [isOk,up_xielv,down_xielv,t_between150_190,t_up217,maxT]=condition(T)
2 global deltat
3 DT=[0,(T(2:end)-T(1:end-1))/deltat];
4 up_xielv=DT(DT>0);
5 down_xielv=DT(DT<0);
6 t_between150_190=(length(T(DT>0&T>150&T<190))-1)*deltat;
7 t_up217=(length(T(T>217))-1)*deltat;
8 maxT=max(T);
9 isOK=1;
10 if ~isempty(up_xielv)&&~isempty(down_xielv)
11     if max(up_xielv)>3 || max(down_xielv)<-3
12         isOK=0;
13     end
14 else
15     isOK=0;
16 end
17 if t_between150_190<60 || t_between150_190>120
18     isOK=0;
19 end
20 if t_up217<40 || t_up217>90
21     isOK=0;
22 end
23 if maxT<240 || maxT>250
24     isOK=0;
25 end
```

#### evaluateS.m

```
1 function Fitness=evaluateS(x)
2 global T15 T06 T07 T89 T1011 v deltat
3 global ts l a b d deltax
4 global len0 len1 len2 len3 len4 len5
5 global alpha_beta
6 T15=x(1);T06=x(2);T07=x(3);T89=x(4);v=x(5)/60;
7 ts=getTs();
8 len0=floor((b)/(v*deltat));
9 len1=floor((b+5*l+4*a)/(v*deltat));
10 len2=floor((b+6*l+5*a)/(v*deltat));
11 len3=floor((b+7*l+6*a)/(v*deltat));
12 len4=floor((b+9*l+8*a)/(v*deltat));
13 len5=floor((b+11*l+10*a+25)/(v*deltat));
14
15 if abs(T15-175)>10 || abs(T06-195)>10 || abs(T07-235)>10 || abs(T89-255)>10
16     Fitness=10000;
17 else
```

```

18 T_Model=getTt(alpha_beta(1:5),alpha_beta(6));
19 [~,maxT_index]=max(T_Model);
20 up217_index=find(T_Model>217);
21 up217_T=T_Model(up217_index(1):maxT_index)-217;
22 up217_S=sum((up217_T(1:end-1)+up217_T(2:end))*deltat/2);
23 [isOK,~,~,~,~]=condition(T_Model);
24 if isOK
25     Fitness=up217_S;
26 else
27     Fitness=10000;
28 end
29 end

```

## evaluateD.m

```

1 function Fitness=evaluateD(x)
2 %对称性评估
3 global T15 T06 T07 T89 T1011 v deltat
4 global ts l a b d deltax
5 global len0 len1 len2 len3 len4 len5
6 global alpha_beta
7 T15=x(1);T06=x(2);T07=x(3);T89=x(4);v=x(5)/60;
8 ts=getTs();
9 len0=floor((b)/(v*deltat));
10 len1=floor((b+5*l+4*a)/(v*deltat));
11 len2=floor((b+6*l+5*a)/(v*deltat));
12 len3=floor((b+7*l+6*a)/(v*deltat));
13 len4=floor((b+9*l+8*a)/(v*deltat));
14 len5=floor((b+11*l+10*a+25)/(v*deltat));
15
16 if abs(T15-175)>10 || abs(T06-195)>10 || abs(T07-235)>10 || abs(T89-255)>10
17     Fitness=10000;
18 else
19     T_Model=getTt(alpha_beta(1:5),alpha_beta(6));
20     [~,maxT_index]=max(T_Model);
21     up217_index=find(T_Model>217);
22     up217_T=T_Model(up217_index(1):maxT_index)-217;
23     up217_S=sum((up217_T(1:end-1)+up217_T(2:end))*deltat/2);
24     [isOK,~,~,~,~]=condition(T_Model);
25     chazhi=T_Model(up217_index(1):maxT_index)-T_Model((2*maxT_index-up217_index(1))
26     :-1:maxT_index);
27     D=sqrt(sum(chazhi.^2)/length(chazhi));
28     if isOK && up217_S<505
29         Fitness=D;
30     else
31         Fitness=10000;
32     end
33 end

```

## linminxin.m

```

1  %灵敏性分析
2  global T15 T06 T07 T89 T1011 v deltat
3  global ts l a b d deltax
4  global len0 len1 len2 len3 len4 len5
5  global alpha_beta
6  load alpha_beta1.mat
7  T1011=25;l=30.5;a=5;b=25;deltat=0.5;d=0.015;deltax=0.001;
8  x=[182.113972660114 189.859222123262 230.258078795877 264.984602768553
    91.5833943649100 488.210219430565];
9  isOK=zeros(4,10);S=zeros(4,10);
10 for i=1:4
11     for j=1:10
12         y=x;
13         y(i)=y(i)+(j-5)/10;
14         [isOK(i,j),S(i,j)]=evaluateS2(y(1:5));
15     end
16 end
17 D_S=(S-x(6))/x(6);
18 D_T=(1:10)-5)/10;
19 figure('Position',[454.6,349.8,591.2,335.9])
20 for i=1:4
21     plot(D_T,D_S(i,:),'-'),hold on
22 end
23 for i=1:4
24     for j=1:10
25         if isOK(i,j)==0
26             scatter(D_T(j),D_S(i,j),'r*')
27         end
28     end
29 end
30 legend('T15','T06','T07','T89','不满足约束'),xlabel('温区温度变化 \Delta T'),ylabel
    ('面积相对变化\Delta S/S')
31 beautiplot
32 exportgraphics(gcf,'img\灵敏度分析.png','Resolution',400)
33
34
35 function [isOK,Fitness]=evaluateS2(x)
36 global T15 T06 T07 T89 T1011 v deltat
37 global ts l a b d deltax
38 global len0 len1 len2 len3 len4 len5
39 global alpha_beta
40 T15=x(1);T06=x(2);T07=x(3);T89=x(4);v=x(5)/60;
41 ts=getTs();
42 len0=floor((b)/(v*deltat));
43 len1=floor((b+5*l+4*a)/(v*deltat));

```

```

44 len2=floor((b+6*l+5*a)/(v*deltat));
45 len3=floor((b+7*l+6*a)/(v*deltat));
46 len4=floor((b+9*l+8*a)/(v*deltat));
47 len5=floor((b+11*l+10*a+25)/(v*deltat));
48
49 if abs(T15-175)>10 || abs(T06-195)>10 || abs(T07-235)>10 || abs(T89-255)>10
50     isOK=0;
51 end
52
53 T_Model=getTt(alpha_beta(1:5),alpha_beta(6));
54 [~,maxT_index]=max(T_Model);
55 up217_index=find(T_Model>217);
56 up217_T=T_Model(up217_index(1):maxT_index)-217;
57 up217_S=sum((up217_T(1:end-1)+up217_T(2:end))*deltat/2);
58 [isOK,~,~,~,~,~]=condition(T_Model);
59 Fitness=up217_S;
60 end

```