

回流焊接建模仿真设计

摘要

本文通过热辐射与热对流模拟元件在回流焊接器中受热过程，仿真其温度变化。

我们首先查阅文献并进行分析，在给定各个小温区的温度后，整个系统达到热稳定，我们使用两种方法进行了模拟，并使用 Matlab 中 `pdemodeler` 进行二次验证。因为工件焊点面积非常小，随后我们在考虑整体环境对焊点的热对流和热辐射的情况下，列出差分方程。在附件给出的曲线条件（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）），使用退火算法寻找和附件曲线各点差平方和最小的系数值，并带入之后的模拟。

问题一我们使用之前求的的相关系数，对 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）条件下的温度分布进行仿真，随后模拟工件通过这一流程。小温区 3、6、7 中点及小温区 8 结束处的温度为法一 [140.8685°C 178.2353°C 197.7191°C 229.8128°C] 法二 [128.4420°C 185.8256°C 205.2437°C 232.7520°C]。炉温曲线数据见 `result.csv`，只列出法一的数据。

问题二首先模拟 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）情况下回流焊炉温度分布。再通过遍历传送带速率，寻找到符合五项制程要求的最大传送带速度，为法一：80.63cm/min，法二 79.13cm/min。

问题三有五项决策变量，使用退火算法解五维的优化问题，使超过 217°C 到峰值的面积最小。通过计算，各温区设定为 183.3°C 小温区 1-5）、203.6°C（小温区 6）、228.6°C（小温区 7）和 265°C（小温区 8-9），过炉速度为 99.87 cm/min，求得最小面积为 467.6122°Cs

问题四为多目标最优化问题，在考虑面积最小化同时，考虑峰值两侧函数值差平方和最小话，我们使用遗传算法求解。得到一组 rank 为 1 的解，相关值请详见内容，在此不予全部列出。

关键字： 回流焊接 热辐射 热对流 非线性优化模型

一、 问题重述

1.1 问题背景

由于电子元器件的不断小型化，需要使用回流焊接的方式来焊接，通过加热，将元器件自动接到电路板上。整个工艺流程对温度控制有较高要求。

1.2 问题重述

通过热量转递机理建立模型，使得对回流焊接有较好的温度控制。回流焊接机有 11 个小温区以及炉前区域和炉后区域，各小温区长 30.5cm，相邻小温区之间有 5cm 的间隙，炉前区域和炉后区域长度均为 25cm。室温视为 25°C。

要求 1-5 温区温度保持一致，8-9 温区温度保持一致，10-11 温区保持 25°C。解决下列问题：

(1) 请对焊接区域的温度变化规律建立数学模型。假设传送带过炉速度为 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），请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，并将每隔 0.5 s 焊接区域中心的温度存放在提供的 result.csv 中。

(2) 假设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），请确定允许的最大传送带过炉速度。

(3) 最小化超过 217°C 到峰值温度所覆盖的面积。请确定在此要求下的最优炉温曲线，以及各温区的设定温度和传送带的过炉速度，并给出相应的面积。

(4) 在（3）基础上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。请结合问题 3，进一步给出最优炉温曲线，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。

二、 问题分析

2.1 问题一分析

题目给出了各个温区的温度分布和传送带速度，要求我们给出焊接区域中心温度曲线。

首先考虑整个回流焊炉内温度分布，因为空气的热吸收系数基本可以忽略，视为透明体，所以在炉内达到问题的过程中热传导行为占据主导。参考顾老师《数理方法》[1]，即为解给定边界条件下的拉普拉斯方程。但由于边界条件比较复杂，解析解难以获得，应当选择数值解法。

在此问题中，根据参考文献 [2] 由于焊片的面积较小，热传导可以忽略，此过程中只考虑热辐射与热对流。可以列出热量传递方程。在此给定条件下，根据热交换方程，问题得解。

2.2 问题二分析

问题二给出小温区的设定温度，根据热传导方程可以建立温度场分布。随后考虑工件吸热过程，可以得到炉温曲线。对相关范围中的传送带速度进行遍历，对每一条传送带速度得到一条炉温曲线，对每条曲线进行分析，判断其是否满足五条制程要求。寻找在符合制程要求下的最大传送带速度。

2.3 问题三分析

在给定温区温度和传送带速度后，可以得到唯一炉温曲线。对曲线大于 217°C 到峰值的部分求积分，也即由五个量确定了一个面积值，随后使用退火算法解五维的最优化问题即可。

2.4 问题四分析

在问题三基础上，将面积和对称度都作为目标，使用遗传算法解多目标的优化问题。对称度可以将右侧按峰值对称到左边，随后逐点求差平方和。

三、模型的假设

- 温度场瞬时建立，工件进入温度场已经达到稳态
- 工件在回流焊炉中对空气没有扰动，空气始终保持稳定
- 工件在回流焊炉中忽略热传导
- 空气视为透明物质，不考虑热辐射对其影响
- 测量器和焊接区域看作质点，热对流只考虑中轴线的空气
- 简化问题为二维问题，不考虑加热器的纵深(宽度)
- 由于热辐射是电磁波光速传播，不考虑不同位置热源辐射到达时间
- 温度分布只与空间有关，与时间无关
- 焊接部件的运动不会影响温度场，温度场始终恒定

- 小温区当作条形、厚度为零的恒定热源

四、符号说明

符号	意义
c	工件比热容 (J/°C)
m	工件质量 (kg)
d	回焊炉焊接区高度 (m)
$Source_T$	小温区温度分布 (°C)
Air_T	空气温度 (°C)
α	高于 217°C 对称度
$T1$	小温区 1-5 温度 (°C)
$T2$	小温区 6 温度 (°C)
$T3$	小温区 7 温度 (°C)
$T4$	小温区 8-9 温度 (°C)
L	回流焊炉长度 (L)

五、模型预备

5.1 加热因子定义

炉温曲线大于 217°C 到峰值的区域定义为加热因子。加热因子对产品质量影响很大，是一个重要的衡量炉温曲线的值。

5.2 给定温度温度场模拟

回焊炉内部的温度分布可以看作是一个包含复杂边界条件或非齐次项的温度场问题，通常用热传导方程来描述：

$$\frac{\partial u}{\partial t} = \alpha \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (1)$$

对于稳态过程：

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

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad (3)$$

由此可以看出，稳态的情况下，温度场问题是一个二维的拉普拉斯方程求解问题。对于回流焊而言，上述问题还需要根据题目要求考虑边界条件的设置。为了简化计算的时间复杂度，本文中采用了两种边界条件的设置方法，并根据此温度场分布计算了炉温曲线，分析了不同设置方法的优缺点。

5.2.1 温度场建立方法一

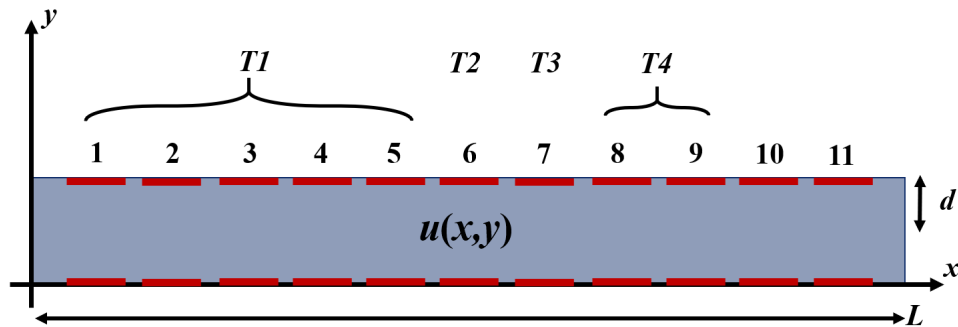


图 1 回流焊结构

当固定某一维度时，温度场对另一维度的二阶导数为零，所以温度场在此维度上为线性分布，因此第一种温度场建立方法里，我们将空气间隙设置为线性连接两端温区，小温区的温度控制在设定温度，且只考虑上图中灰蓝色区域内的温度场，由此可以得到加热区顶部和底部的边界条件：

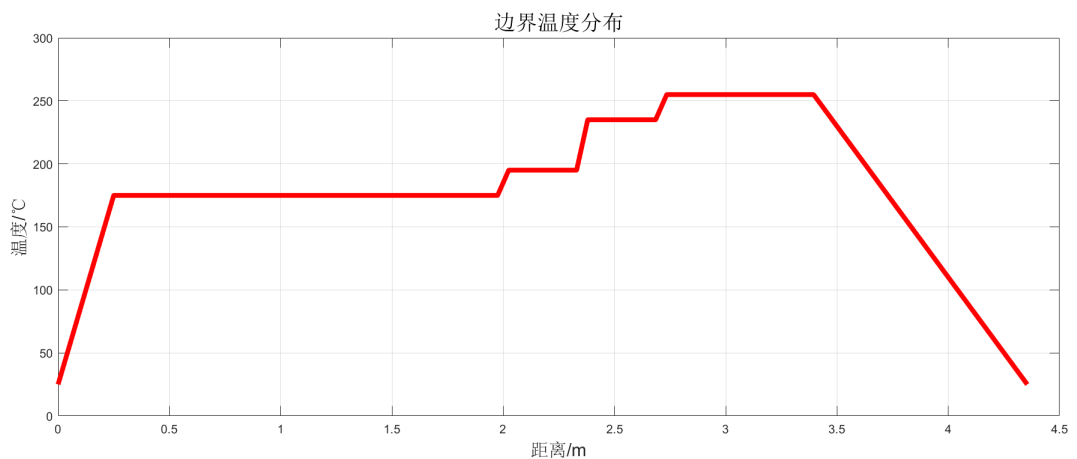


图 2 方法二回流焊温度分布

温度场左右两端均设置为室温。

二维有界拉普拉斯方程可以通过差分法数值得到, 将待求解区域划分为 $N1 \times N2$ 的矩形网格, 对于网格内每一个格点的温度值可以由以下差分结果得到:

$$u_{ij} = \frac{u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}}{4}, i, j \in [1, N2] \times [1, N1] \quad (4)$$

令 b_1, b_2, b_3, b_4 分别为顶端底端左端右端的边界条件

$$u_{1,j} = b_1, u_{N_1,j} = b_2, u_{i,1} = b_3, u_{i,N_2} = b_4 \quad (5)$$

上式可以矩阵化, 通过共轭梯度法求解得到收敛的温度分布:

$$\begin{pmatrix} u_{1,1}(k+1) \\ \vdots \\ u_{1,N_2}(k+1) \\ \vdots \\ u_{N_1,1}(k+1) \\ \vdots \\ u_{N_1,N_2}(k+1) \end{pmatrix} = A \begin{pmatrix} u_{1,1}(k) \\ \vdots \\ u_{1,N_2}(k) \\ \vdots \\ u_{N_1,1}(k) \\ \vdots \\ u_{N_1,N_2}(k) \end{pmatrix} + b \quad (6)$$

矩阵 A 的细节和边界 B 的细节详见附件代码, 迭代上述表达式得到收敛的状态:

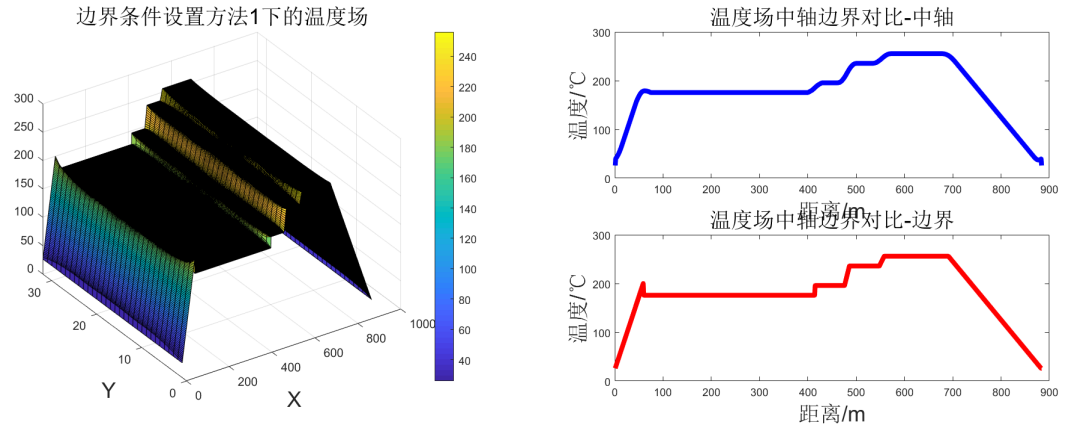


图 3 方法一回流焊温度场分布与中轴边界温度分布对比

图中右边两幅对比了温度场中轴和边界的温度变化, 两者基本重合, 差异非常小, 所以在这种边界条件设置下, 可以认为温度场在垂直焊接物运动方向上分布均匀, 在焊接物运动方向上温度分布和边界一致。

5.2.2 温度场建立方法二

在方法二中, 我们假象了一个无限大的空气介质内有二十二个条形热源, 设热源分布函数为 $f(x, y), T(x)$ 为热源温度的 x 方向分布, 定解问题变成:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad (7)$$

$$f(x, y) = \delta(y)T(x) + \delta(y - 2d)T(x); \quad (8)$$

参考数理方程，对此类问题的求解可以通过傅里叶变换和逆变换求得 $u(x, y)$ 的卷积表达形式：

$$u(x, y) = \frac{1}{\pi} \left[\int_{-\infty}^{+\infty} T(\xi) \frac{y}{(x - \xi)^2 + y^2} d\xi + \int_{-\infty}^{+\infty} T(\xi) \frac{2d - y}{(x - \xi)^2 + (y - 2d)^2} d\xi \right] \quad (9)$$

将上式离散化，数值求解可以得到温度场，由于我们只关注条形热源内部的温度场分布，可忽略外部的求解，下图展示的是全局温度场分布和中轴线温度分布：

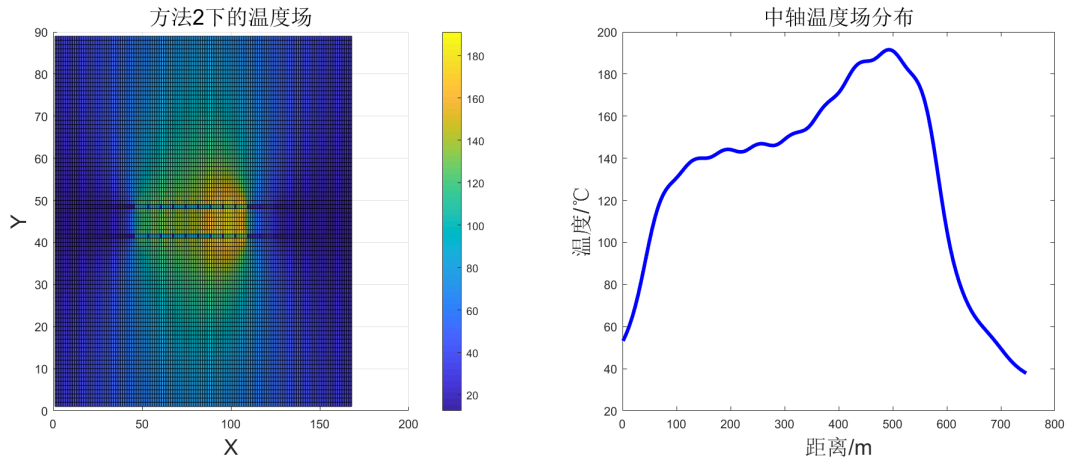


图 4 方法二回流焊温度场分布及中轴温度分布

5.2.3 方法一二对比及优缺点

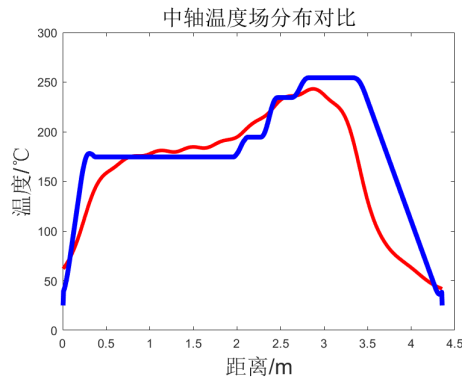


图 5 两种方法中轴温度场分布对比

1. 方法一适合加热区域厚度比较薄的情况，对于计算量比较大的情况，该近似可以省略大量的计算，节省计算成本，但其物理过程并不完全正确，尤其在左右两端，线性行偏离较大；

2. 方法二的物理图像更准确，可以看出左右两端类似指数分布，但计算量较大，在后期优化高维变量向量的适合，会带来大量计算成本；

3. 实际上的温度场外围有铁皮，应当做绝热条件处理，但一方面题目中没有明确的参数条件，一方面计算起来不方便，但可以计算一种参量设置情况下温场与
后文将基于不同的温度分布分别计算各个题目。

5.3 炉温曲线相关系数测定

由于原题中没有给出相关材料与机器宽的数据，导致我们在计算热辐射与热对流时有很多参量欠定，并且整个问题被约化到了“假”二维问题，可以写出以下热量吸收式：

$$\phi_{total} = \phi_{convection} + \phi_{radiate} \quad (10)$$

热对流 指流体中质点发生相对位移而引起的热量传递过程。[3] 我们在此问题中将工件所处环境温度视为对流温度。忽略空气环境的流动，仅考虑工件运动过设定的空气环境。

热辐射 热辐射，物体由于具有温度而辐射电磁波的现象。热量传递的 3 种方式之一。一切温度高于绝对零度的物体都能产生热辐射，温度愈高，辐射出的总能量就愈大。[4]。空气主要由 N_2 和 O_2 组成，根据热学知识，对于双原子对称分子，吸收系数基本可以忽略。所以我们考虑热辐射时忽略空气吸收和辐射，仅考虑各个小温区的热辐射。

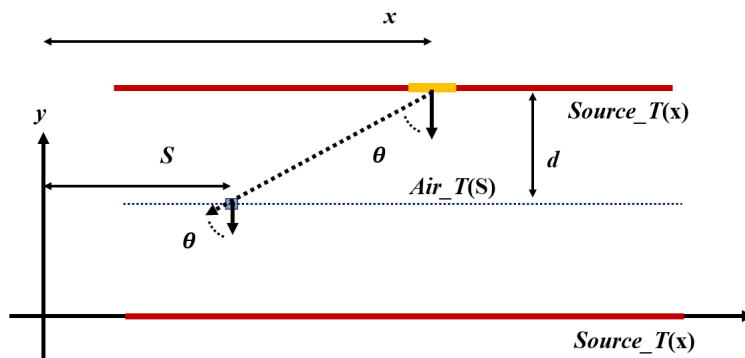


图 6 工件受热示意图

考虑探测器周围小质元吸收热量的变化率为： $cm \frac{\Delta u}{\Delta t}$ ，其大小受到空气热对流 $\phi_{convection}$ 和热辐射 $\phi_{radiate}$ 的影响

$$cm \frac{\Delta u}{\Delta t} = \phi_{convection} + \phi_{radiate} \quad (11)$$

$$\phi_{convection} = A(Air_T(S) - u) \quad (12)$$

$$\phi_{radiate} = \int_0^L [B \times Source_T(x)^4] \frac{d}{d^2 + (x - S)^2} dx - D \times u^4 \quad (13)$$

$$S = vt \quad (14)$$

$$cm \frac{\Delta u}{\Delta t} = A(Air_T(S) - u) + \int_0^L B \times Source_T(x)^4 \frac{d}{d^2 + (x - S)^2} dx - D \times u^4 \quad (15)$$

将上式两端同时除以 cm ，吸收进 A 、 B 和 D ，重新定义参数 C_1 、 C_2 、 C_3

$$\Delta u = \Delta t [C_1(Air_T(S) - u) + \int_0^L C_2[Source_T(x)^4] \frac{d}{d^2 + (x - S)^2} dx] - C_3 \times u^4 \quad (16)$$

将上式离散化：

$$u(t+1) = u(t) + \Delta t [C_1(Air_T(v \cdot t) - u) + \sum C_2 Source_T(x)^4 \frac{d}{d^2 + (x - vt)^2} \Delta x] - C_3 \times u^4 \quad (17)$$

上式中系数 C_1, C_2, C_3 由题目给定的曲线拟合得到，这里我们首先观察一下前两个系数单独存在的作用，请见图 7：

系数 C_1 刻画了与热空气交换热量的过程， C_1 越大，炉温曲线与空气温度场中轴线分布越接近。而系数 C_2 刻画的是热源的辐射影响，从上图右可以看出在前期热源辐射导致的加热作用比较明显，但进入冷却区域后，辐射项加热作用转为散热作用。所以与冷空气的对流和热辐射散热作用让焊接区域在冷却区有了一个比较可观的温度下降过程。

5.3.1 系数确定

Air_T 即我们在温度场建立环节得到的空气温度场中轴分布，我们通过退火算法寻找建模数据和实验数据差的平方和的最小值对应的系数 C_1, C_2, C_3 ，两种方法得到了两套系数，并且计算了两套系数确定的曲线与给定曲线的 R 方，如下图 8、9 所示：

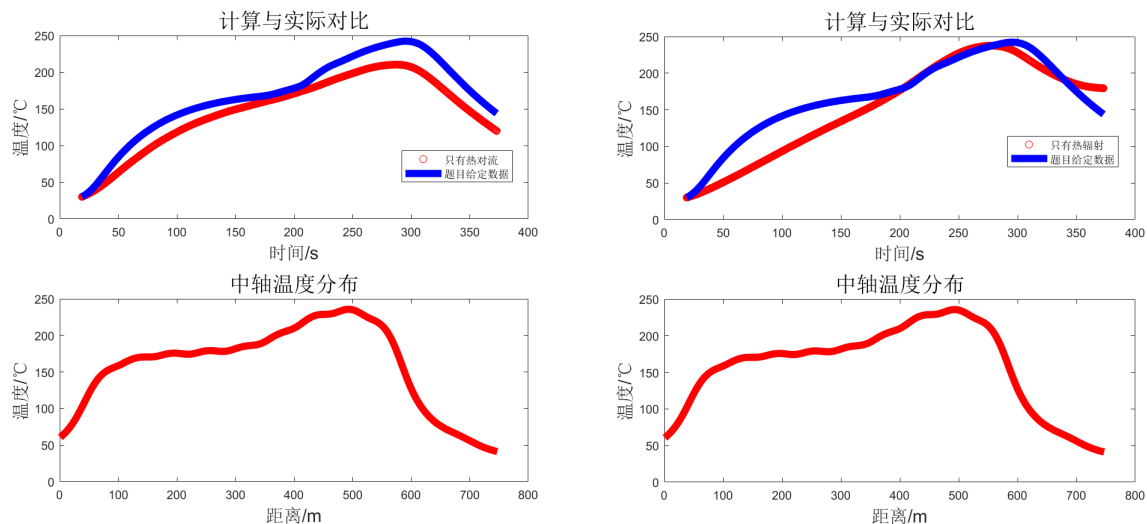


图 7 计算值与附件值对照

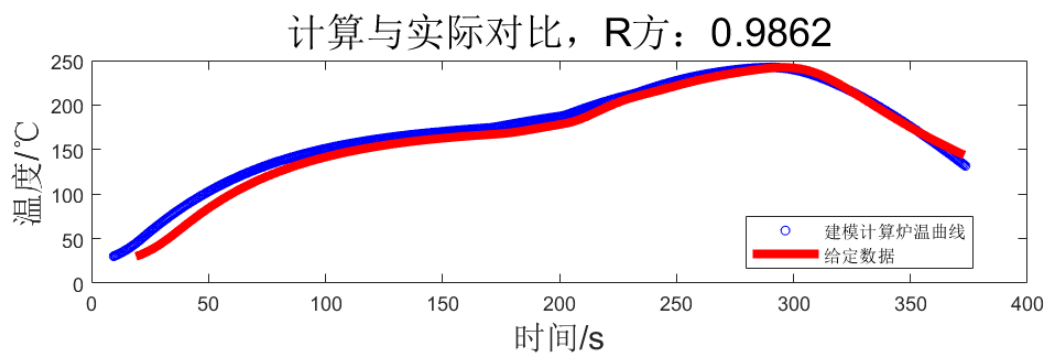


图 8 方法一拟合

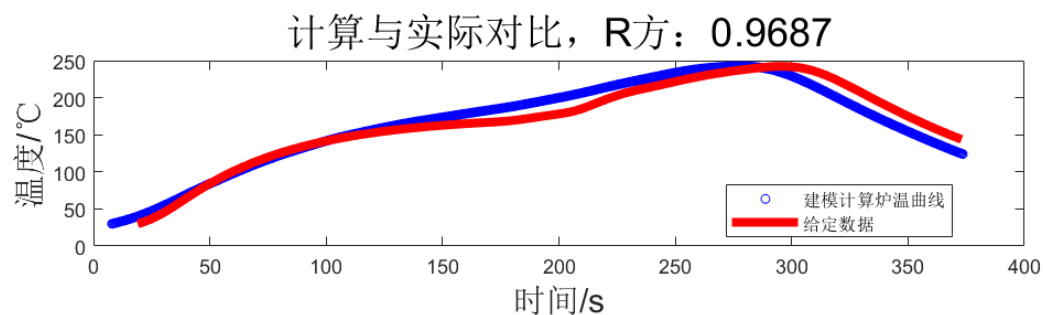


图 9 方法二拟合

六、模型建立与求解

温度场建立后，根据17，计算工件在一定位置的温度，可以得到其炉温曲线，可以解决以下问题。

表 1 方法一系数表

C_1	C_2	C_3
0.0163	2.86e-11	6e-12

表 2 方法二系数表

C_1	C_2	C_3
0.0148	1.36e-11	3e-13

6.1 问题一

基于上述模型，将四个温度和移动速度代入程序得到：

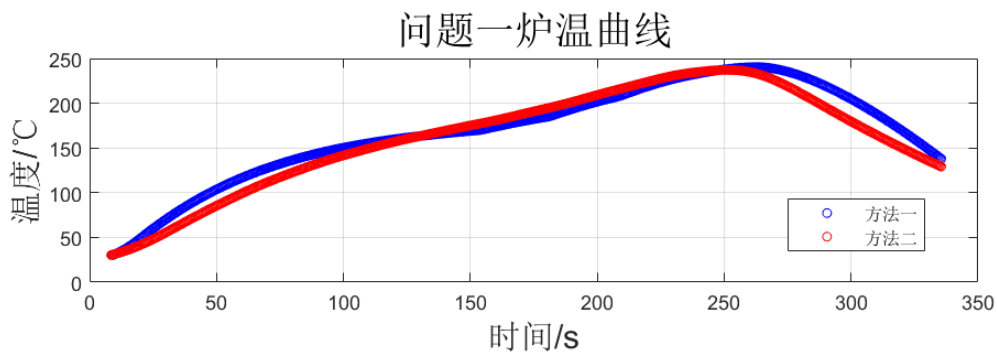


图 10 炉温曲线

表 3 问题一结果

	温区 3 中部	温区 6 中部	温区 7 中部	温区 8 结束
方法一	140.8685°C	178.2353°C	197.7191°C	229.8128°C
方法二	128.4420°C	185.8256°C	205.2437°C	232.7520°C

6.2 问题二

问题二中，定性的极限理解，如果速度过慢，会突破低温区的上限，高温区的上限，但温度上升和下降斜率会很慢，同时峰值可能会很高，因为热对流和热辐射导致升热时

间更久。但若速度过快，温度上升和下降斜率就会过快，同时在温区的时间不满足要求，峰值不达要求，所以温度应当在一个中间区域。我们通过遍历的方法寻找适合的最大速度。

表 4 问题一结果

传送带最大速度	
方法一	80.63(cm/min)
方法二	79.13(cm/min)

6.3 问题三

与问题一，问题二相同，我们可以通过直接带入 5 个参数（4 个小温区的设定温度，传送带的过炉速度）得到一组每隔 0.5s 焊接区域中心的温度数据，可将其用数组的形式表达。通过对该数组的处理，得到超过 217°C 至峰值温度的温度数据。根据温度数据计算出炉温曲线加热因子的面积，并由题目中的制程界限得到约束条件，建立单目标优化模型进行求解。采用退火算法求解该模型。

6.3.1 面积模型

根据得到的炉温曲线超过 217°C 至峰值温度数据，采用梯形法执行数值积分运算，通过将一个区域分为包含多个更容易计算的区域梯形，该方法对区间内的积分计算近似值。

如下图所示，因为时间间隔为 0.5s，因此两点间间隔为 0.5，根据 $f(x_1), f(x_2)$ 可以得出某一个梯形的面积。将所有梯形面积累加即可得近似面积。

对于 N+1 个均匀分布的点，可得近似面积为

$$\begin{aligned}
 S &\approx \frac{1}{4} \sum_{n=1}^N (f(x_n) + f(x_{n+1})) \\
 &= \frac{1}{4} [f(x_1) + 2f(x_2) + \dots + 2f(x_N) + f(x_{N+1})]
 \end{aligned} \tag{18}$$

6.3.2 约束条件

根据题目中制程界限可得一系列约束条件。

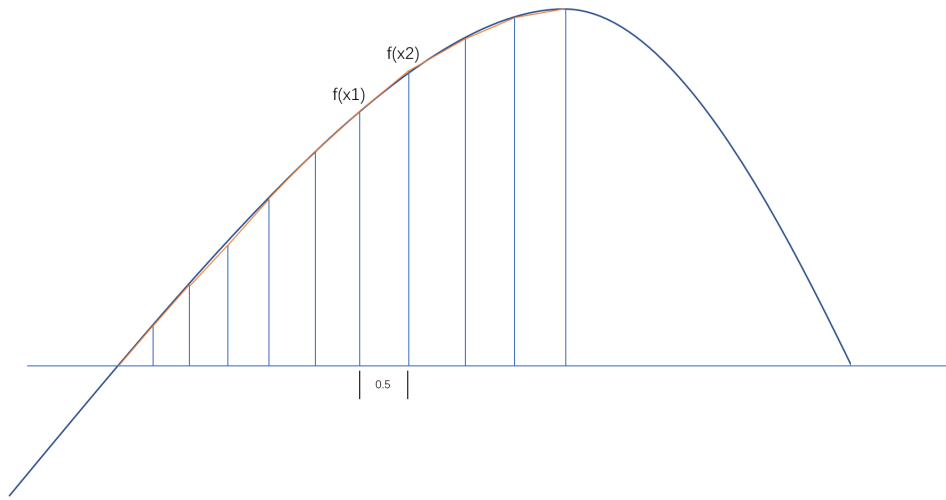


图 11 加热因子面积划分

- 温度斜率 k 温度斜率应该满足最高值不超过 3，最低值不低于-3。因此应满足 $-3 < k < 3$ 。而斜率 k 可通过下式表示

$$k_i = \frac{x_{i+1} - x_i}{0.5}, x_i \in U \quad (19)$$

- 温度上升过程中在 150°C 190°C 的时间 t_{150190} 温度上升过程中在 150°C 190°C 的时间最高值不超过 120s，最低值不低于 60s。因此应满足 $60 < t_{150190} < 120$ 。
- 温度大于 217°C 的时间 t_{217} 温度大于 217°C 的时间最高值不超过 90s，最低值不低于 40s。因此应满足 $30 < t_{217} < 90$ 。
- 峰值温度 T_{max} 峰值温度最高值不超过 250°C，最低值不低于 240°C。因此应满足 $240 < T_{max} < 250$ 。

以上所有约束条件都可通过遍历法查找数据数组得到。

6.3.3 模型建立

通过以上分析可以建立优化模型，如下

$$\min S \quad (20)$$

$$\text{s.t.} \quad -3 < \mathbf{k} < 3 \quad (21)$$

$$60 < t_{150190} < 120 \quad (22)$$

$$30 < t_{217} < 90 \quad (23)$$

$$240 < T_{max} < 250 \quad (24)$$

6.3.4 模型求解

由于数据数组会随着 5 个输入参数的变化实时改变，因此无法直接通过输入参数建立与面积之间的表达式。因此我们使用智能优化算法解决该问题。智能优化算法主要有：模拟退火算法，蚁群算法，遗传算法等等。我们使用模拟退火算法求解模型。

模拟退火算法的主要特点为：若新生成的解更优则直接选择，否则以一定概率接受新解。这样保证了能够在解空间中找到目标函数的全局最优解。

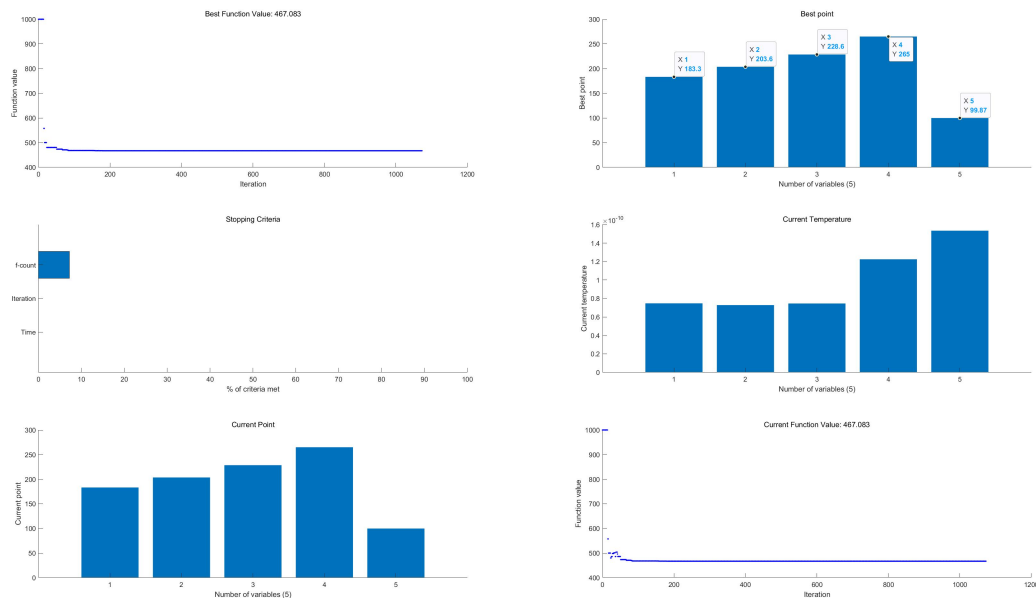


图 12 问题三退火算法

经过代码计算，方法一、二结果如下：

表 5 问题三各温区温度及传送带速度（方法一）

T1	T2	T3	T4	传送带速度	加热因子面积
183.3°C	203.6°C	228.6°C	265°C	99.87 cm/min	467.6122°C s

表 6 问题三各制程值（方法一）

time _{150,190}	time ₂₁₇	max _T	slope _{min}	slope _{max}
62.5000s	58s	240.0112°C	-2.4581°C/s	2.5003°C/s

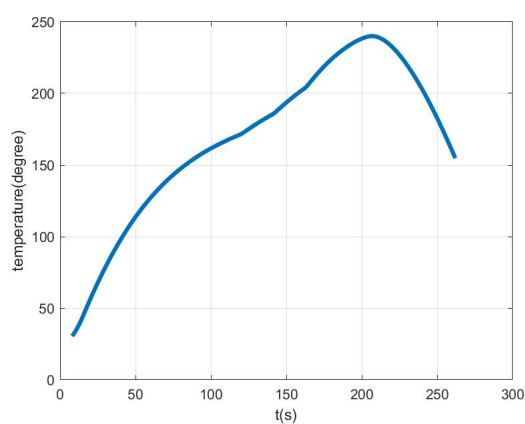


图 13 问题三炉温曲线（方法一）

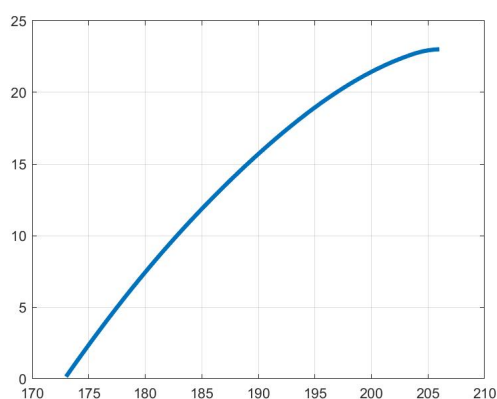


图 14 问题三加热因子区域曲线（方法一）

6.3.5 对称度 α 定义

题目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。

表 7 问题三各温区温度及传送带速度（方法二）

T1	T2	T3	T4	传送带速度	加热因子面积
179.6°C	187.2°C	225.6°C	264.9°C	80.12 cm/min	570.504°C s

表 8 问题三各制程值（方法二）

$time_{150,90}$	$time_{217}$	max_T	$slope_{min}$	$slope_{max}$
65.0000s	65.0s	240.0000°C	-1.6538°C/s	1.6437°C/s

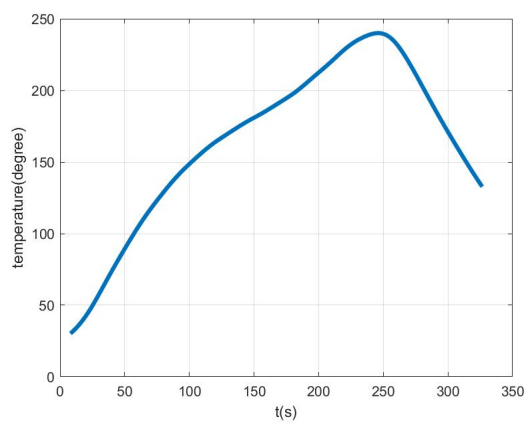


图 15 问题三炉温曲线（方法二）

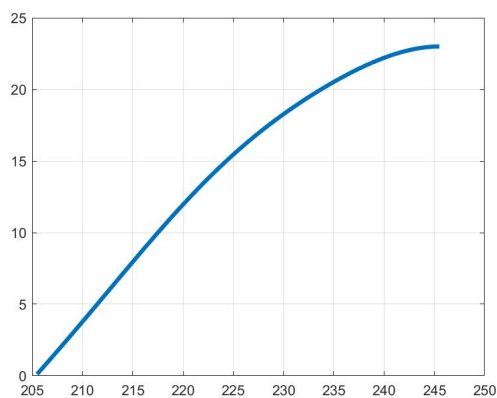


图 16 问题三加热因子区域曲线（方法二）

step 1 取出超过 217°C 的炉温曲线数据，并找到峰值数据。

step 2 以峰值为中心，依次对左右两侧相距中心间隔相同的数据点做差。若一侧有值而另一次没有，就将没有值的一侧设置为 0。

step 3 将差平方和，作为衡量对称度的指标

6.4 问题四

问题四在考虑面积的基础上，还需考虑对称度，因此该问题为多目标优化问题，目标函数为面积和对称度。我们建立多目标优化模型，在制程界限的约束条件下，使用多目标遗传算法求解该模型。称度的指标。 α 越高对称度越差， α 越低对称度越好。

6.4.1 模型建立

由于该问题有两个目标函数，因此我们需要同时考虑两个目标函数建立优化模型，具体如下：

$$\min_{S, \alpha} \mathbf{f} \quad (25)$$

$$\text{s.t. } \mathbf{f} = \{S, \alpha\} \quad (26)$$

$$-3 < \mathbf{k} < 3 \quad (27)$$

$$60 < t_{150190} < 120 \quad (28)$$

$$30 < t_{217} < 90 \quad (29)$$

$$240 < T_{max} < 250 \quad (30)$$

6.4.2 模型求解

此时我们对 α 与加热因子应用 NSGA-II 多目标函数优化方法求最优解。NSGA-II 算法在遗传算法的基础上引入非劣解的概念，能够快速寻找出同时满足多目标函数最优的参量。首先介绍一下非劣解的概念，非劣解是指在种群中无法被任何其他个体支配 (dominate) 的个体。数学定义如下：

$$\forall i \in m, f_i(x_1) \leq f_i(x_2) \quad (31)$$

$$\text{and } j \in m, f_j(x_1) < f_j(x_2) \quad (32)$$

$$\text{and } \mathbf{F} = \{f_1x, f_2x, f_3x, ; f_mx\} \quad (33)$$

若满足上述数学条件，则称 x_1 支配 (dominate) x_2 。如果在所有个体中没有其它个体支配 x_1 ，则称 x_1 为非劣解，并且可以并入 Rank1 (第一梯队)。在每次迭代中选取

Rank 最低的种群个体进行交叉变异生成下一代种群。反复迭代能够找到较优的非劣解。

我们使用 NSGA-II 算法对方法一进行求解。得到以下结果，

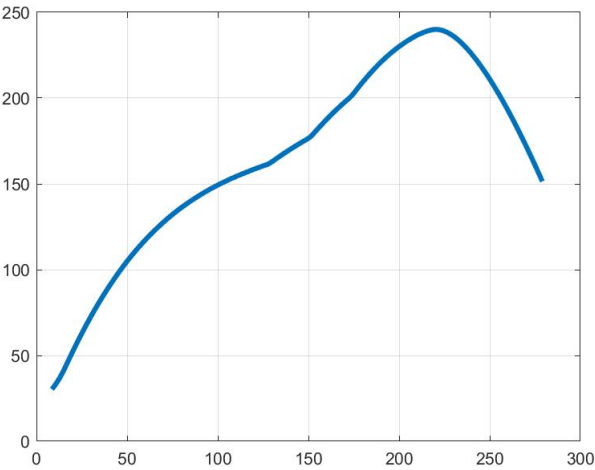


图 17 问题四炉温曲线（方法一）

T1/ (摄氏度)	T2/ (摄氏度)	T3/ (摄氏度)	T4/ (摄氏度)	传送带/ (cm/min)
170.150983454879	194.965302391165	232.956094101733	264.800643257023	93.714369844364
170.150983446760	194.965376065798	232.956049986898	264.800643150965	93.714369838513
170.150983454879	194.965302391165	232.956094101733	264.800643257023	93.714369844364
170.150983446760	194.965376065798	232.956049986898	264.800643150965	93.714369838513
170.150983359418	194.966278546090	232.955503768213	264.800643053879	93.714369833182
170.150983359420	194.966278545723	232.955503768207	264.800643062493	93.714369833187
170.150983359351	194.966278545401	232.955503768270	264.800643086679	93.714369833181
170.150983359377	194.966278545681	232.955503768277	264.800643056614	93.714369833182
170.150983359601	194.966278546671	232.955503768350	264.800643048038	93.714369833197
170.150983359351	194.966278545401	232.955503768270	264.800643086679	93.714369833181
170.150983359478	194.966278546113	232.955503768406	264.800643050011	93.714369833183
170.150983359414	194.966278545544	232.955503768424	264.800643070087	93.714369833170
170.150983359373	194.966278545498	232.955503768302	264.800643076849	93.714369833182
170.150983361214	194.966278547423	232.955503768646	264.800643041972	93.714369833297
170.150983359438	194.966278545556	232.955503768285	264.800643062832	93.714369833171
170.150983361844	194.966278547445	232.955503769036	264.800643041308	93.714369833376
170.150983360454	194.966278546210	232.955503769623	264.800643044484	93.714369833228
170.150983359414	194.966278545544	232.955503768424	264.800643070087	93.714369833170

图 18 求出的 rank1 梯队个体输入参量

以上是对方法一求解得到的结果，最优的加热因子的面积稳定在 477.011 附近，最优的对称度稳定在 29.74 附近。我们由第三问对方法一的求解可以得知最优加热因子面积为 467.08，对该情况下的输入参数得到的炉温曲线可以计算出对称度为 30.83。我们可以发现这是符合物理规律的，因为该问题同时考虑了两个目标函数，因此在其中一个目标函数尽可能更有的情况下，另一个目标函数可能要变为次优来维持两者的均衡，使两个目标函数都尽可能更靠近最优。

做出大于 217°C 的曲线区域，我们可以发现曲线较为对称且光滑。

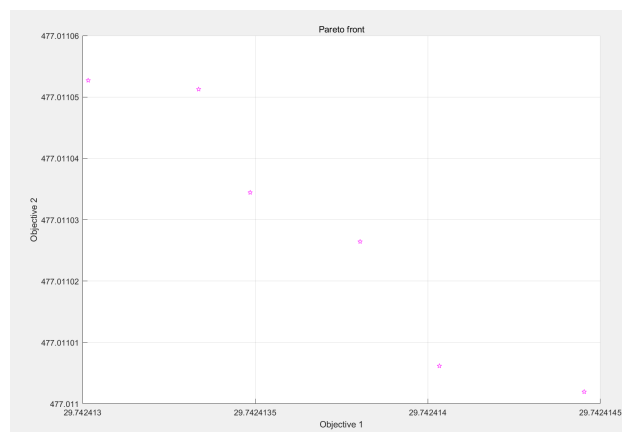


图 19 rank1 梯队个体最优面积- α 对称度

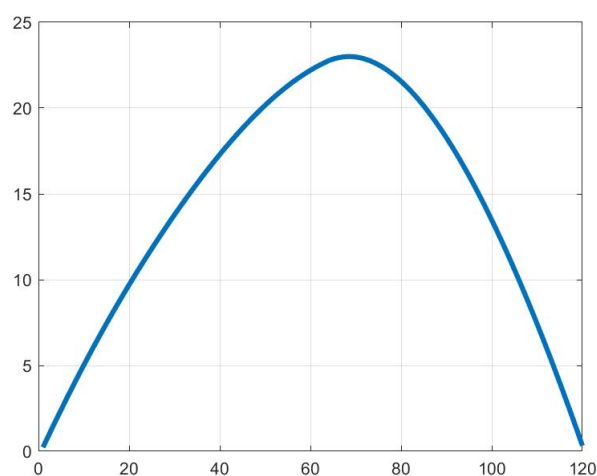


图 20 rank1 梯队某个体大于 217 度的炉温曲线

6.4.3 模型求解的缺点

1. 由于多目标函数的遗传算法处理问题的时候每次迭代都需要一定时间，因此找到最优解往往需要较长时间，我们这里仅对方法一进行了求解。

2. NSGA-II 算法最后会得到一组 Rank 较低的个体，在这一组个体中很难选择出唯一一个最优，如果个体之间的拥挤度不够大，则会导致决策难度上升。但也可最后结合线性加权组合法或主要目标函数法做出判断。

七、模型评价与展望

我们查阅了国内相关数值模拟回焊炉的文献，发现并没有太多可以得到的文献。根据高金刚文中介绍 [5]，数值计算在回流焊工艺中需要考虑的变量过多，效用不大，仍需要工程师通过多次实际调整得到最佳炉温曲线。所以对于加工过程中炉温曲线与加热

因子等要素的方针与优化很重要也有很好的发展前景。

7.1 模型优点

1. 本文通过有限的条件估计出相关系数，曲线拟合较好。
2. 使用退火算法与遗传算法，工作效率较高
3. 使用多种分布相互印证，模型比较符合物理规律
4. 进行了有效的取舍，忽略相关不重要参量

7.2 模型缺点与展望

由于真实情况较为复杂，模型做了很多简化，比如忽略热传导，将小温区三维化简到一维，这使得整个模型并不完全符合物理规律。若有可能，我们希望带入真实的三维模型，使用真实的相关参数进行计算。

两个温度场分布各有优劣，法一因使用线性方法在前部分拟合较好，但在冷却区中，线性的假设并不符合物理规律，导致炉温曲线在最后趋势上和给定的数据有差异；法二在后半部分因考虑真实物理模型拟合较好，但在 8，9 区之前的温度渐变过程让炉温曲线在中部与给定数据差距较大，且计算复杂度更高。希望后期改进可以更加优化。

系数拟合过程中的退火算法表现出比较强的起点依赖特性，由于单次炉温曲线的计算需要 0.5 秒左右的时间，当多个变量大范围改变时，所需成本较高，可能需要更好的全局搜索算法，同时提高单次计算的效率。

在第四题遇到多目标优化函数的时候，遗传算法表现出了不稳定性，由于 MATLAB 自带的算法工具包有迭代次数限制，往往最优解需要比较长的计算时间才出现，更高效的算法值得进一步探索和讨论。

参考文献

- [1] 顾樵, 数学物理方法. 科学出版社, 2012.
- [2] 姚焕, 基于 *Ansys Icepak* 的板级回流焊接建模与仿真. PhD thesis, 北华大学, 2018.
- [3] “热对流.” <https://baike.baidu.com/item//6319974?fr=aladdin>. Accessed on 2020-09-11.
- [4] “热辐射.” <https://baike.baidu.com/item//120950?fr=aladdin>. Accessed on 2020-09-11.

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

附录 A 温度场生成和拟合主程序

1.1 方法一

```
%%第一题方法一代码
tic
%炉温曲线计算
data = xlsread('附件');
sensor = data(:,2);
time = data(:,1);

%温区可变参数设置
T1 = 175;
T2 = 195;
T3 = 235;
T4 = 255;
v = 0.7/60; %单位: m/s

%温区不变参数设置
d = 0.2;
dt = 0.5;
L = 4.355;

%待定参数设置 方法一
C1 = 0.0163;
C2 = 2.86e-11;
C3 = 6e-12;
%焊接位置温度初始化
uu = 25;
u = [uu];
for t = dt:dt:L/v
    S = v*t;
    sum = 0;
    for x = 0:0.005:4.355
        if SourceT(x,T1,T2,T3,T4) ~= 0
            sum =
```

```

        sum+C2*0.005*((SourceT(x,T1,T2,T3,T4)+273.16)^4)*d^2/((x-S)^2+d^2);

    end

end

uu = uu + dt*(C1*(boundaryT(S,T1,T2,T3,T4)-uu)+sum)-C3*(uu+273.16)^4; %法一
u = [u uu];
end
ttt = [];
uuu = [];
for i=1:length(u)
    if u(i)>30
        uuu=[uuu,u(i)];
        ttt = [ttt, dt*i];
    end
end

% 拟合指标计算
min=(uuu(1:709)-sensor(:,1)')*(uuu(1:709)-sensor(:,1)')');
a = (sensor' - uuu(1:709))*(sensor' - uuu(1:709))';
b = (sensor' - mean(uuu(1:709))*ones(1,709))*(sensor' -
    mean(uuu(1:709))*ones(1,709))';
R = 1 - a/b;

% 炉温曲线绘制
plot(ttt,uuu,'bo','MarkerSize',4)
hold on
plot(time,sensor,'r','linewidth',4);
legend('建模计算炉温曲线','给定数据')
xlabel('时间/s','FontSize',15);
ylabel('温度/','FontSize',15);
title('计算与实际对比, R方: 0.9862','FontSize',18);

toc

%源分布函数
function center = axis(v,T1,T2,T3,T4)
dt =0.5;
d = 0.2
delta = v*dt;
source1 = [];

```

```

for i =0:delta:4.355
    source1 = [source1 SourceT(i,T1-25,T2-25,T3-25,T4-25)];
end
center = [];
for j =1:length(0:delta:4.355)
    sum = 0;
    x = 0+j*delta;
    y= (floor(d/delta)+1)*delta;
    for m = 1:length(source1)
        sigma1 = 0+m*delta;
        sum = sum+1/pi*source1(m)*y/((x-sigma1)^2+(y)^2)*delta;
    end
    center = [center sum];
end
center = center + 25*ones(1,length(center));
end

function T = SourceT(x,T1,T2,T3,T4)
    if x<0
        T =0;
    end
    if x>=0 && x<0.25
        T = 0;
    end
    if x>=0.25 && x<0.555
        T = T1;
    end
    if x>=0.555 && x<0.605
        T = 0;
    end
    if x>=0.605 && x<0.91
        T = T1;
    end
    if x>=0.91 && x<0.96
        T =0;
    end
    if x>=0.96 && x<1.265

```



```

    T = T1;
end
    if x>=1.265 && x<1.315
        T = 0;
    end
    if x>=1.315 && x<1.62
        T = T1;
    end
    if x>=1.62 && x<1.67
        T = 0;
    end
    if x>=1.67 && x<1.975
        T = T1;
    end
    if x>=1.975 && x<2.025
        T = 0;
    end
    if x>=2.025 && x<2.33
        T = T2;
    end
    if x>=2.33 && x<2.38
        T = 0;
    end
    if x>=2.38 && x<2.685
        T = T3;
    end
    if x>=2.685 && x<2.735
        T = 0;
    end
    if x>=2.735 && x<3.04
        T = T4;
    end
    if x>=3.04 && x<3.09
        T = 0;
    end
    if x>=3.09 && x<3.395
        T = T4;
    end

```

```

end
if x>=3.395 && x<3.445
    T = 0;
end
if x>=3.445 && x<3.75
    T = 25;
end
if x>=3.75 && x<3.8
    T = 0;
end
if x>=3.8 && x<4.105
    T = 25;
end
if x>=4.105 && x<4.355
    T = 0;
end
if x>=4.355
    T = 0;
end
end

function T = boundaryT(x,T1,T2,T3,T4)
    if x<0
        T =25;
    end
    if x>=0 && x<0.25
        T = 25+(T1-25)/0.25*x;
    end
    if x>=0.25 && x<1.975
        T = T1;
    end
    if x>=1.975 && x<2.025
        T = T1+(T2-T1)/0.05*(x-1.975);
    end
    if x>=2.025 && x<2.33
        T = T2;
    end
end

```

```

if x>=2.33 && x<2.38
    T = T2+(T3-T2)/0.05*(x-2.33);
end
if x>=2.38 && x<2.685
    T = T3;
end
if x>=2.685 && x<2.735
    T = T3+(T4-T3)/0.05*(x-2.685);
end
if x>=2.735 && x<3.395
    T = T4;
end
if x>=3.395 && x<4.355
    % T = 0.27*200*cos(2*pi/0.96*(x+3.395))+T4+(25-T4)/0.71*(x-3.395);
    T = T4+(25-T4)/0.96*(x-3.395);
end
if x>=4.355
    T = 25;
end
end
end

```

1.2 方法二

```

%% 第一题方法二代码
tic
%炉温曲线计算
data = xlsread('附件');
sensor = data(:,2);
time = data(:,1);

%温区可变参数设置
T1 = 175;
T2 = 195;
T3 = 235;
T4 = 255;
v = 0.7/60; %单位: m/s

```

```
%温区不变参数设置
```

```
d = 0.2;
```

```
dt =0.5;
```

```
L =4.355;
```

```
%待定参数设置 方法二
```

```
C1 = 0.0148;
```

```
C2 = 1.36e-11;%5.8696e-10;
```

```
C3 = 3e-13;
```

```
%迭代
```

```
uu = 25;
```

```
u = [uu]; %焊接位置温度初始化
```

```
center = axis(v,T1,T2,T3,T4); %边界初始化
```

```
for t = dt:dt:L/v
```

```
    S = v*t;
```

```
    sum = 0;
```

```
    for x = 0:0.025:4.355
```

```
        if SourceT(x,T1,T2,T3,T4) ~= 0
```

```
            sum =
```

```
                sum+C2*0.025*((SourceT(x,T1,T2,T3,T4)+273.16)^4)*d^2/((x-S)^2+d^2);
```

```
        end
```

```
    end
```

```
    uu = uu + dt*(C1*(center(floor(t/dt))-uu)+sum-C3*(uu+273.16)^4);
```

```
    u = [u uu];
```

```
end
```

```
ttt2 = [];
```

```
uuu2 = [];
```

```
for i=1:length(u)
```

```
    if u(i)>30
```

```
        uuu2=[uuu2,u(i)];
```

```
        ttt2 = [ttt2, dt*i];
```

```
    end
```

```
end
```

%优化控制参量和衡量参量

```
min=(uuu2(1:709)-sensor(:,1))'*(uuu2(1:709)-sensor(:,1))';  
a = (sensor' - uuu2(1:709))*(sensor' - uuu2(1:709))';  
b = (sensor' - mean(uuu2(1:709))*ones(1,709))*(sensor' -  
    mean(uuu2(1:709))*ones(1,709))';  
R = 1 - a/b;
```

% 炉温曲线绘制

```
plot(ttt2,uuu2,'bo','MarkerSize',4)  
hold on  
plot(time,sensor,'r','linewidth',4);  
    legend('建模计算炉温曲线','给定数据')  
xlabel('时间/s','FontSize',15);  
ylabel('温度/','FontSize',15);  
title('计算与实际对比, R方: 0.9687','FontSize',18);
```

%源分布函数

```
function center = axis(v,T1,T2,T3,T4)  
dt =0.5;  
d = 0.2  
delta = v*dt;  
source1 = [];  
for i =0:delta:4.355  
    source1 = [source1 SourceT(i,T1-25,T2-25,T3-25,T4-25)];  
end  
center = [];  
for j =1:length(0:delta:4.355)  
    sum = 0;  
    x = 0+j*delta;  
    y= (floor(d/delta)+1)*delta;  
    for m = 1:length(source1)  
        sigma1 = 0+m*delta;  
        sum = sum+1.315*1/pi*source1(m)*y/((x-sigma1)^2+(y)^2)*delta;  
    end  
    center = [center sum];  
end
```

```
center = center + 25*ones(1,length(center));  
end
```

```
function T = SourceT(x,T1,T2,T3,T4)
```

```
    if x<0
```

```
        T =0;
```

```
    end
```

```
    if x>=0 && x<0.25
```

```
        T = 0;
```

```
    end
```

```
    if x>=0.25 && x<0.555
```

```
        T = T1;
```

```
    end
```

```
    if x>=0.555 && x<0.605
```

```
        T = 0;
```

```
    end
```

```
    if x>=0.605 && x<0.91
```

```
        T = T1;
```

```
    end
```

```
    if x>=0.91 && x<0.96
```

```
        T =0;
```

```
    end
```

```
    if x>=0.96 && x<1.265
```

```
        T = T1;
```

```
    end
```

```
    if x>=1.265 && x<1.315
```

```
        T = 0;
```

```
    end
```

```
    if x>=1.315 && x<1.62
```

```
        T = T1;
```

```
    end
```

```
    if x>=1.62 && x<1.67
```

```
        T =0;
```

```
    end
```

```
    if x>=1.67 && x<1.975
```

```
        T = T1;
```

```
    end
```

```

if x>=1.975 && x<2.025
    T = 0;
end
if x>=2.025 && x<2.33
    T = T2;
end
if x>=2.33 && x<2.38
    T = 0;
end
if x>=2.38 && x<2.685
    T = T3;
end
if x>=2.685 && x<2.735
    T = 0;
end
if x>=2.735 && x<3.04
    T = T4;
end
    if x>=3.04 && x<3.09
        T = 0;
    end
    if x>=3.09 && x<3.395
        T = T4;
    end
if x>=3.395 && x<3.445
    T = 0;
end
if x>=3.445 && x<3.75
    T = 25;
end
if x>=3.75 && x<3.8
    T = 0;
end
if x>=3.8 && x<4.105
    T = 25;
end
if x>=4.105 && x<4.355

```

```

        T = 0;
    end
    if x>=4.355
        T = 0;
    end
end

function T = boundaryT(x,T1,T2,T3,T4)
    if x<0
        T =25;
    end
    if x>=0 && x<0.25
        T = 25%+(T1-25)/0.25*x;
    end
    if x>=0.25 && x<1.975
        T = T1;
    end
    if x>=1.975 && x<2.025
        T = T1+(T2-T1)/0.05*(x-1.975);
    end
    if x>=2.025 && x<2.33
        T = T2;
    end
    if x>=2.33 && x<2.38
        T = T2+(T3-T2)/0.05*(x-2.33);
    end
    if x>=2.38 && x<2.685
        T = T3;
    end
    if x>=2.685 && x<2.735
        T = T3+(T4-T3)/0.05*(x-2.685);
    end
    if x>=2.735 && x<3.395
        T = T4;
    end
    if x>=3.395 && x<4.355
        T = T4+(25-T4)/0.96*(x-3.395);
    end
end

```



```

    end
    if x>=4.355
        T = 25;
    end
end
end

```

附录 B 问题二主程序

```

%%第二题代码

%%初始化
maxi1 = 0;
maxi2 = 0;

%%遍历
for i = 65:0.01:100
    if checkfunc_method1(i) == 1
        maxi1 = i;
    end
end
for i = 65:0.01:100
    if checkfunc_method2(i) == 1
        maxi2 = i;
    end
end

%%判断函数
function ret = checkfunc_method1(x)

%% 设定初始值
v = x/100/60; %单位: m/s
%焊接位置温度初始化
d = 0.2; %传送带半宽
%温区参数设置
T1 = 182;

```

```

T2 = 203;
T3 = 237;
T4 = 254;

%迭代
dt =0.5;
L =4.355;
%待定参数设置
C1 = 0.0163;
C2 = 2.86e-11;%5.8696e-10;
C3 = 6e-12

slope_min = 0;
slope_max = 0;
maxi = 0;
time_150_190 = 0;
time_217 = 0;

%% 计算曲线数据
uu = 25;
u = [uu]; %焊接位置温度初始化
for t = dt:dt:L/v
    S = v*t;
    sum = 0;
    for x = 0:0.005:4.355
        if SourceT(x,T1,T2,T3,T4) ~= 0
            sum =
                sum+C2*0.005*((SourceT(x,T1,T2,T3,T4)+273.16)^4)*d^2/((x-S)^2+d^2);
        end
    end
    uu = uu + dt*(C1*(boundaryT(S,T1,T2,T3,T4)-uu)+sum)-C3*(uu+273.16)^4; %法一
    u = [u uu];
end

%% 检查制程界限
diff = zeros(length(u)-1,1);

```

```

slope = zeros(length(u)-1,1);

for i = 1: length(u) - 1
    diff(i) = u(i+1) - u(i);
    slope(i) = diff(i) / dt;
    if slope(i) > slope_max
        slope_max = slope(i);
    end
    if slope(i) < slope_min
        slope_min = slope(i);
    end
end

start_150 = 0;
end_190 = 0;
start_217 = 0;
end_217 = 0;
for i = 1:length(u)
    if i == 1 || i == length(u)
        continue;
    end

    if u(i) >= maxi
        maxi = u(i);
    end
    if u(i) >= 217 && u(i - 1) < 217
        start_217 = i;
    end
    if u(i) >= 217 && u(i + 1) < 217
        end_217 = i;
    end
    if u(i) <= 150 && u(i + 1) > 150
        start_150 = i + 1;
    end
    if u(i) <= 190 && u(i + 1) > 190
        end_190 = i;
    end
end

```

```

        end

    end

    time_150_190 = (end_190 - start_150) / 2
    time_217 = (end_217 - start_217) / 2
    maxi = maxi
    slope_min = slope_min;
    slope_max = slope_max;
    if slope_min < -3 || slope_max > 3 || maxi > 250 || maxi < 240 ||
        time_217 < 40 || time_217 > 90 || time_150_190 < 60 || time_150_190
        > 120
        ret = 0;
        disp('fail');
    else
        ret = 1;
    end
end

function ret = checkfunc_method2(x)

%% 设定初始值
v = x/100/60; %单位: m/s
%焊接位置温度初始化
d = 0.2; %传送带半宽
%温区参数设置
T1 = 182;
T2 = 203;
T3 = 237;
T4 = 254;

%迭代
dt =0.5;
L =4.355;
%待定参数设置
C1 = 0.0148;
C2 = 1.36e-11;%5.8696e-10;

```

```

C3 = 3e-13

slope_min = 0;
slope_max = 0;
maxi = 0;
time_150_190 = 0;
time_217 = 0;

%% 计算曲线数据
uu = 25;
u = [uu]; %焊接位置温度初始化
center = axis(v,T1,T2,T3,T4); %边界初始化
for t = dt:dt:L/v
    S = v*t;
    sum = 0;
    for x = 0:0.025:4.355
        if SourceT(x,T1,T2,T3,T4) ~= 0
            sum =
                sum+C2*0.025*((SourceT(x,T1,T2,T3,T4)+273.16)^4)*d^2/((x-S)^2+d^2);
        end
    end
    uu = uu + dt*(C1*(center(floor(t/dt))-uu)+sum-C3*(uu+273.16)^4);
    u = [u uu];
end
uuu = []
for i=1:length(u)
    if u(i)>30
        uuu=[uuu,u(i)];
    end
end

%% 检查制程界限
diff = zeros(length(u)-1,1);
slope = zeros(length(u)-1,1);

for i = 1: length(u) - 1

```

```

    diff(i) = u(i+1) - u(i);
    slope(i) = diff(i) / dt;
    if slope(i) > slope_max
        slope_max = slope(i);
    end
    if slope(i) < slope_min
        slope_min = slope(i);
    end
end

start_150 = 0;
end_190 = 0;
start_217 = 0;
end_217 = 0;
for i = 1:length(u)
    if i == 1 || i == length(u)
        continue;
    end

    if u(i) >= maxi
        maxi = u(i);
    end
    if u(i) >= 217 && u(i - 1) < 217
        start_217 = i;
    end
    if u(i) >= 217 && u(i + 1) < 217
        end_217 = i;
    end
    if u(i) <= 150 && u(i + 1) > 150
        start_150 = i + 1;
    end
    if u(i) <= 190 && u(i + 1) > 190
        end_190 = i;
    end
end
end

```

```

time_150_190 = (end_190 - start_150) / 2
time_217 = (end_217 - start_217) / 2
maxi = maxi
slope_min = slope_min
slope_max = slope_max
if slope_min < -3 || slope_max > 3 || maxi > 250 || maxi < 240 ||
    time_217 < 40 || time_217 > 90 || time_150_190 < 60 || time_150_190
    > 120
    ret = 0;
    disp('fail');
else
    ret = 1;
end
end

%% 边界温度分布函数
function center = axis(v,T1,T2,T3,T4)
dt =0.5;
d = 0.2
delta = v*dt;
source1 = [];
for i =0:delta:4.355
    source1 = [source1 SourceT(i,T1-25,T2-25,T3-25,T4-25)];
end
center = [];
for j =1:length(0:delta:4.355)
    sum = 0;
    x = 0+j*delta;
    y= (floor(d/delta)+1)*delta;
    for m = 1:length(source1)
        sigma1 = 0+m*delta;
        sum = sum+1.315*1/pi*source1(m)*y/((x-sigma1)^2+(y)^2)*delta;
    end
    center = [center sum];
end
center = center + 25*ones(1,length(center));
end

```

```
function T = SourceT(x,T1,T2,T3,T4)
```

```
    if x<0
```

```
        T =0;
```

```
    end
```

```
    if x>=0 && x<0.25
```

```
        T = 0;
```

```
    end
```

```
    if x>=0.25 && x<0.555
```

```
        T = T1;
```

```
    end
```

```
        if x>=0.555 && x<0.605
```

```
            T = 0;
```

```
        end
```

```
        if x>=0.605 && x<0.91
```

```
            T = T1;
```

```
        end
```

```
        if x>=0.91 && x<0.96
```

```
            T =0;
```

```
        end
```

```
        if x>=0.96 && x<1.265
```

```
            T = T1;
```

```
        end
```

```
        if x>=1.265 && x<1.315
```

```
            T = 0;
```

```
        end
```

```
        if x>=1.315 && x<1.62
```

```
            T = T1;
```

```
        end
```

```
        if x>=1.62 && x<1.67
```

```
            T =0;
```

```
        end
```

```
        if x>=1.67 && x<1.975
```

```
            T = T1;
```

```
        end
```

```
    if x>=1.975 && x<2.025
```

```
        T = 0;
```

```
    end
```



```

if x>=2.025 && x<2.33
    T = T2;
end
if x>=2.33 && x<2.38
    T = 0;
end
if x>=2.38 && x<2.685
    T = T3;
end
if x>=2.685 && x<2.735
    T = 0;
end
if x>=2.735 && x<3.04
    T = T4;
end
    if x>=3.04 && x<3.09
        T = 0;
    end
    if x>=3.09 && x<3.395
        T = T4;
    end
if x>=3.395 && x<3.445
    T = 0;
end
if x>=3.445 && x<3.75
    T = 25;
end
if x>=3.75 && x<3.8
    T = 0;
end
if x>=3.8 && x<4.105
    T = 25;
end
if x>=4.105 && x<4.355
    T = 0;
end
if x>=4.355

```

```

        T = 0;
    end
end
function T = boundaryT(x,T1,T2,T3,T4)
    if x<0
        T =25;
    end
    if x>=0 && x<0.25
        T = 25%+(T1-25)/0.25*x;
    end
    if x>=0.25 && x<1.975
        T = T1;
    end
    if x>=1.975 && x<2.025
        T = T1+(T2-T1)/0.05*(x-1.975);
    end
    if x>=2.025 && x<2.33
        T = T2;
    end
    if x>=2.33 && x<2.38
        T = T2+(T3-T2)/0.05*(x-2.33);
    end
    if x>=2.38 && x<2.685
        T = T3;
    end
    if x>=2.685 && x<2.735
        T = T3+(T4-T3)/0.05*(x-2.685);
    end
    if x>=2.735 && x<3.395
        T = T4;
    end
    if x>=3.395 && x<4.355
        T = T4+(25-T4)/0.96*(x-3.395);
    end
    if x>=4.355
        T = 25;
    end
end

```

end

附录 C 问题三主程序

```
function fitnessVal = fitness_three_methodone( x )
%% 设定初始值
%   x = [184.6 189.1 226.4 260 96.9]
%   lb = [165 185 225 245 65]
%   ub = [185 205 245 265 100]
v = x(5)/100/60; %单位: m/s
%焊接位置温度初始化

d = 0.2; %传送带半宽
%温区参数设置
T1 = x(1);
T2 = x(2);
T3 = x(3);
T4 = x(4);

%迭代
dt =0.5;
L = 4.355;
%待定参数设置
C1 = 0.01249;
C2 = 5.8696e-10;

slope_min = 0;
slope_max = 0;
maxi = 0;
time_150_190 = 0;
time_217 = 0;

%% 求解炉温曲线数据
uu = 25;
```

```

u = [uu]; %焊接位置温度初始化
for t = dt:dt:L/v
    S = v*t;
    sum = 0;
    for x = 0:0.005:4.355
        if SourceT(x,T1,T2,T3,T4) ~= 0 && SourceT(x,T1,T2,T3,T4) ~= 25
            sum =
                sum+C2*0.005*(SourceT(x,T1,T2,T3,T4)^4-uu^4)*d^2/((x-S)^2+d^2);
        end
    end
    uu = uu + dt*(C1*(boundaryT(S,T1,T2,T3,T4)-uu)+sum); %法一
    u = [u uu];
end

%% 绘制炉温曲线
%     hold on;
%     plot(0:dt:L/v,u);
%     data = xlsread('data.xlsx');
%     u = data(:,2);
%     plot(data(:,1),data(:,2));
%     hold off

%% 求解目标函数
% 暂时的目标函数为曲线最大值
%     fitnessVal = u(1);
%     for i = 1:length(u)
%         if u(i) > fitnessVal
%             fitnessVal = u(i);
%         end
%     end

% 曲线大于217度的面积
row = 0:dt:L/v;
col = u;
maxx = 0;
for i = 1:length(row)
    if col(i) >= maxx

```

```

        maxx = col(i);
        t_max = i;
    end
    if col(i) >= 217 && col(i - 1) < 217
        start_2177 = i;
    end
end

k = t_max - start_2177 + 1;
rownew = zeros(k,1);
colnew = zeros(k,1);
for i = 1:k
    rownew(i) = row(start_2177 + i - 1);
    colnew(i) = col(start_2177 + i - 1) - 217;
end

fitnessVal = trapz(rownew,colnew);

%         plot(rownew,colnew);

%% 检查约束条件
diff = zeros(length(u)-1,1);
slope = zeros(length(u)-1,1);

for i = 1: length(u) - 1
    diff(i) = u(i+1) - u(i);
    slope(i) = diff(i) / dt;
    if slope(i) > slope_max
        slope_max = slope(i);
    end
    if slope(i) < slope_min
        slope_min = slope(i);
    end
end
end

```

```

start_150 = 0;
end_190 = 0;
start_217 = 0;
end_217 = 0;
for i = 1:length(u)
    if i == 1 || i == length(u)
        continue;
    end

    if u(i) >= maxi
        maxi = u(i);
    end

    if u(i) >= 217 && u(i - 1) < 217
        start_217 = i;
    end

    if u(i) >= 217 && u(i + 1) < 217
        end_217 = i;
    end

    if u(i) <= 150 && u(i + 1) > 150
        start_150 = i + 1;
    end

    if u(i) <= 190 && u(i + 1) > 190
        end_190 = i;
    end

end

time_150_190 = (end_190 - start_150) / 2 ;
time_217 = (end_217 - start_217) / 2;
maxi = maxi;
slope_min = slope_min;
slope_max = slope_max;
if slope_min < -3 || slope_max > 3 || maxi > 250 || maxi < 240 ||
    time_217 < 40 || time_217 > 90 || time_150_190 < 60 || time_150_190
    > 120
    fitnessVal = 1000;%失败设参
    disp('fail');
end

```

```

        else
            disp('succ')
        end

end

%%
%源分布函数
function center = axis(v,T1,T2,T3,T4)
dt =0.5;
d = 0.2
delta = v*dt;
source1 = [];
for i =0:delta:4.355
    source1 = [source1 SourceT(i,T1-25,T2-25,T3-25,T4-25)];
end
center = [];
for j =1:length(0:delta:4.355)
    sum = 0;
    x = 0+j*delta;
    y= (floor(d/delta)+1)*delta;
    for m = 1:length(source1)
        sigma1 = 0+m*delta;
        sum = sum+1/pi*source1(m)*y/((x-sigma1)^2+(y)^2)*delta;
    end
    center = [center sum];
end
center = center + 25*ones(1,length(center));
end
%源分布函数
function T = SourceT(x,T1,T2,T3,T4)
    if x<0
        T =0;

    elseif x>=0 && x<0.25
        T = 0;

    elseif x>=0.25 && x<0.555

```

```

    T = T1;

elseif x>=0.555 && x<0.605
    T = 0;

elseif x>=0.605 && x<0.91
    T = T1;

elseif x>=0.91 && x<0.96
    T =0;

elseif x>=0.96 && x<1.265
    T = T1;

elseif x>=1.265 && x<1.315
    T = 0;

elseif x>=1.315 && x<1.62
    T = T1;

elseif x>=1.62 && x<1.67
    T =0;

elseif x>=1.67 && x<1.975
    T = T1;

elseif x>=1.975 && x<2.025
    T = 0;

elseif x>=2.025 && x<2.33
    T = T2;

elseif x>=2.33 && x<2.38
    T = 0;

elseif x>=2.38 && x<2.685
    T = T3;

```



```

elseif x>=2.685 && x<2.735
    T = 0;

elseif x>=2.735 && x<3.04
    T = T4;

elseif x>=3.04 && x<3.09
    T = 0;

elseif x>=3.09 && x<3.395
    T = T4;

elseif x>=3.395 && x<3.445
    T = 0;

elseif x>=3.445 && x<3.75
    T = 25;

elseif x>=3.75 && x<3.8
    T = 0;

elseif x>=3.8 && x<4.105
    T = 25;

elseif x>=4.105 && x<4.355
    T = 0;

elseif x>=4.355
    T = 0;
end
end

function T = boundaryT(x,T1,T2,T3,T4)
    if x<0
        T =25;

```

```

elseif x>=0 && x<0.25
    T = 25+(T1-25)/0.25*x;

elseif x>=0.25 && x<1.975
    T = T1;

elseif x>=1.975 && x<2.025
    T = T1+(T2-T1)/0.05*(x-1.975);

elseif x>=2.025 && x<2.33
    T = T2;

elseif x>=2.33 && x<2.38
    T = T2+(T3-T2)/0.05*(x-2.33);

elseif x>=2.38 && x<2.685
    T = T3;

elseif x>=2.685 && x<2.735
    T = T3+(T4-T3)/0.05*(x-2.685);

elseif x>=2.735 && x<3.395
    T = T4;

elseif x>=3.395 && x<4.355
    T = 0.27*200*cos(2*pi/0.96*(x+3.395))+T4+(25-T4)/0.71*(x-3.395);

elseif x>=4.355
    T = 25;
end
end

```

附录 D 问题四主程序

```

%%第四题优化函数(以方法1为例)(第三题去掉目标函数一即可)

```

```

function fitnessVal = fitness_four_methodone( x )
%% 设定初始值
%   x = [178.8 185.1 228.9 259 99.86];
v = x(5)/100/60; %单位: m/s
%焊接位置温度初始化

d = 0.2; %传送带半宽
%温区参数设置
T1 = x(1);
T2 = x(2);
T3 = x(3);
T4 = x(4);

%迭代
dt =0.5;
L =4.355;
%待定参数设置
C1 = 0.0163;
C2 = 2.86e-11;%5.8696e-10;
C3 = 6e-12;

slope_min = 0;
slope_max = 0;
max = 0;
time_150_190 = 0;
time_217 = 0;

%% 求解炉温曲线数据(未定)
uu = 25;
u = [uu]; %焊接位置温度初始化
for t = dt:dt:L/v
    S = v*t;
    sum = 0;
    for x = 0:0.005:4.355
        if SourceT(x,T1,T2,T3,T4) ~= 0
            sum =

```

```

        sum+C2*0.005*((SourceT(x,T1,T2,T3,T4)+273.16)^4)*d^2/((x-S)^2+d^2);

    end

end

uu = uu + dt*(C1*(boundaryT(S,T1,T2,T3,T4)-uu)+sum)-C3*(uu+273.16)^4;

    %法一
    u = [u uu];

end

ttt = [];
uuu = [];
for i=1:length(u)
    if u(i)>30
        uuu=[uuu,u(i)];
        ttt = [ttt, dt*i];
    end
end

% plot(0:dt:L/v,u);
%% 从曲线处理出大于217度的曲线数据colnew
start_21778 = 0;
end_2177 = 0;
row = 0:dt:L/v;
col = u;
maxx = 0;
for i = 1:length(row)
    if col(i) >= maxx
        maxx = col(i);
        t_max = i;
    end

    if col(i) >= 217 && col(i - 1) < 217
        start_21778 = i;
    end

    if col(i) >= 217 && col(i + 1) < 217
        end_2177 = i;
    end
end

k = end_2177 - start_21778 + 1;
rownew = zeros(k,1);

```

```

colnew = zeros(k,1);
for i = 1:k
    rownew(i) = row(start_21778 + i -1);
    colnew(i) = col(start_21778 + i -1) - 217;
end
left_k = t_max - start_21778;
right_k = end_2177 - t_max;
if left_k <= right_k
    half_k = right_k;
    temp = zeros(right_k - left_k,1);
    colneww = [temp' colnew']';
else
    half_k = left_k;
    temp = zeros(left_k - right_k,1);
    colneww = [colnew' temp']';
end
% plot(colneww);
%% 计算对称的差平方--目标函数一
residue = 0;
for i = 1:half_k
    residue = residue + (colneww(half_k + 1 + i) - colneww(half_k + 1 -
        i))^2;
end
fitnessVal(1) = sqrt(residue)

%% 计算面积--目标函数二
k = t_max - start_21778 + 1;
rownew = zeros(k,1);
colnew = zeros(k,1);
for i = 1:k
    rownew(i) = row(start_21778 + i -1);
    colnew(i) = col(start_21778 + i -1) - 217;
end

fitnessVal(2) = trapz(rownew,colnew)

```

%% 检查约束条件

```
diff = zeros(length(u)-1,1);  
slope = zeros(length(u)-1,1);
```

```
for i = 1: length(u) - 1  
    diff(i) = u(i+1) - u(i);  
    slope(i) = diff(i) / dt;  
    if slope(i) > slope_max  
        slope_max = slope(i);  
    end  
    if slope(i) < slope_min  
        slope_min = slope(i);  
    end  
end
```

```
start_150 = 0;
```

```
end_190 = 0;
```

```
start_217 = 0;
```

```
end_217 = 0;
```

```
for i = 1:length(u)  
    if i == 1 || i == length(u)  
        continue;  
    end  
  
    if u(i) >= max  
        max = u(i);  
    end  
    if u(i) >= 217 && u(i - 1) < 217  
        start_217 = i;  
    end  
    if u(i) >= 217 && u(i + 1) < 217  
        end_217 = i;  
    end  
    if u(i) <= 150 && u(i + 1) > 150  
        start_150 = i + 1;
```

```

        end

        if u(i) <= 190 && u(i + 1) > 190
            end_190 = i;
        end

    end

    time_150_190 = (end_190 - start_150) / 2 ;
    time_217 = (end_217 - start_217) / 2;
    max = max;
    slope_min = slope_min;
    slope_max = slope_max;
    if slope_min < -3 || slope_max > 3 || max > 250 || max < 240 || time_217 <
        40 || time_217 > 90 || time_150_190 < 60 || time_150_190 > 120
        fitnessVal = [3000,3000];%失败设参
        disp('fail');
    end

end

%% 边界温度分布函数
function center = axis(v,T1,T2,T3,T4)
dt =0.5;
d = 0.2
delta = v*dt;
source1 = [];
for i =0:delta:4.355
    source1 = [source1 SourceT(i,T1-25,T2-25,T3-25,T4-25)];
end
center = [];
for j =1:length(0:delta:4.355)
    sum = 0;
    x = 0+j*delta;
    y= (floor(d/delta)+1)*delta;
    for m = 1:length(source1)
        sigma1 = 0+m*delta;
        sum = sum+1/pi*source1(m)*y/((x-sigma1)^2+(y)^2)*delta;
    end
end

```

```

    center = [center sum];
end
center = center + 25*ones(1,length(center));
end
%源分布函数
function T = SourceT(x,T1,T2,T3,T4)
    if x<0
        T =0;

    elseif x>=0 && x<0.25
        T = 0;

    elseif x>=0.25 && x<0.555
        T = T1;

    elseif x>=0.555 && x<0.605
        T = 0;

    elseif x>=0.605 && x<0.91
        T = T1;

    elseif x>=0.91 && x<0.96
        T =0;

    elseif x>=0.96 && x<1.265
        T = T1;

    elseif x>=1.265 && x<1.315
        T = 0;

    elseif x>=1.315 && x<1.62
        T = T1;

    elseif x>=1.62 && x<1.67
        T =0;

    elseif x>=1.67 && x<1.975

```



```

    T = T1;

elseif x>=1.975 && x<2.025
    T = 0;

elseif x>=2.025 && x<2.33
    T = T2;

elseif x>=2.33 && x<2.38
    T = 0;

elseif x>=2.38 && x<2.685
    T = T3;

elseif x>=2.685 && x<2.735
    T = 0;

elseif x>=2.735 && x<3.04
    T = T4;

elseif x>=3.04 && x<3.09
    T = 0;

elseif x>=3.09 && x<3.395
    T = T4;

elseif x>=3.395 && x<3.445
    T = 0;

elseif x>=3.445 && x<3.75
    T = 25;

elseif x>=3.75 && x<3.8
    T = 0;

elseif x>=3.8 && x<4.105
    T = 25;

```

```

elseif x>=4.105 && x<4.355
    T = 0;

elseif x>=4.355
    T = 0;
end
end

function T = boundaryT(x,T1,T2,T3,T4)
    if x<0
        T =25;

elseif x>=0 && x<0.25
    T = 25+(T1-25)/0.25*x;

elseif x>=0.25 && x<1.975
    T = T1;

elseif x>=1.975 && x<2.025
    T = T1+(T2-T1)/0.05*(x-1.975);

elseif x>=2.025 && x<2.33
    T = T2;

elseif x>=2.33 && x<2.38
    T = T2+(T3-T2)/0.05*(x-2.33);

elseif x>=2.38 && x<2.685
    T = T3;

elseif x>=2.685 && x<2.735
    T = T3+(T4-T3)/0.05*(x-2.685);

elseif x>=2.735 && x<3.395
    T = T4;

```

```

elseif x>=3.395 && x<4.355
%   T = 0.27*200*cos(2*pi/0.96*(x+3.395))+T4+(25-T4)/0.71*(x-3.395);
    T = T4+(25-T4)/0.96*(x-3.395);

elseif x>=4.355
    T = 25;
end
end

fitnessfcn = @fitness_four_methodtwo;
nvars = 5;
lb = [165 185 225 245 65];
ub = [185 205 245 265 100];
A = []; b = [];
Aeq = []; beq = [];
options = gaoptimset('PlotFcns',@gaplotpareto);
[x, fval] = gamultiobj(fitnessfcn,nvars, A,b,Aeq,beq,lb,ub,options);
axis([0 50 400 700]);

```

附录 E 固定边界下的拉普拉斯数值解

```

% 方法一固定边界下的拉普拉斯数值解
results1=Laplace_Solver(882,30);
results1 = results1 + 25*ones(size(results1));
subplot(2,2,[1,3])
surf(results1);
xlabel('X','FontSize',18);ylabel('Y','FontSize',18);title('方法1下的温度场','FontSize',18);
subplot(2,2,2)
plot(results1(17,:), 'b', 'linewidth',4)
xlabel('距离/m','FontSize',18);ylabel('温度/','FontSize',18);title('温度场中轴边界对比-中轴',
subplot(2,2,4)
plot(results1(1,:), 'r', 'linewidth',4)
xlabel('距离/m','FontSize',18);ylabel('温度/','FontSize',18);title('温度场中轴边界对比-边界',

function results1 = Laplace_Solver(N1,N2)
% Input:

```

```

% N: size of lattice
% f_00,f_01,f_10,f_11:values on four points
% Output:
% results1: a N*N matrix contains each lattice's unit's value with CGM
% results2: a N*N matrix contains each lattice's unit's value with SGM
%Initial boundary setting
left=[];right=[];
for i=1:N2+2
    left=[left, 0];
    right=[right, 0];
end
top=[];
top = [0:(175-25)/round(25/0.5):175,top];
for i = 1:5
    top = [top 150*ones(1,round(30.5/0.5)),150*ones(1,round(5/0.5))];
end
top = [top 170*ones(1,round(30.5/0.5)) 170:(210-170)/round(5/0.5):210];
top = [top 210*ones(1,round(30.5/0.5)) 210:(230-210)/round(5/0.5):230];
top = [top 230*ones(1,round(66/0.5))];
top = [top 230:(0-230)/round((5+30.5+5+30.5+25)/0.5):0 0];
bottom = top;
%%%Comstruct matrix
%initialize
matrix = 4*eye(N1*N2);
for i=1:N2*N1
    for j =1:N1*N2
        if i-j ==1 || i-j==-1
            matrix(i,j)=-1;
        end
        if i==j && (j+N1)<(N1*N2)
            matrix(i,j+N1) = -1;
        end
        if i==j && (j-N1)>0
            matrix(i,j-N1) = -1;
        end
    end
end
end

```

```

b =
    [top(2:length(top)-1),zeros(1,N1*N2-length(top)-length(bottom)+4),bottom(2:length(top)-1

%Solver
y1 = CGM(matrix,b');
% y2 = SGM(matrix,b');
% Conjugate Gradient Method Solver
function x = CGM (A,b)
[~,n]=size(A);
x = zeros(n,1);
r = b - A * x;
d = r;
for k=1:30
    if r == 0
        break
    end
    alpha = r'*r/(d'*A*d);
    x = x + alpha * d;
    R = r;
    r = b - A * x;
    belta = r'*r/(R'*R);
    d = r + belta * d;
end
end

% % Steepest Gradient Method
% function x = SGM (A,b)
% [~,n]=size(A);
% x = zeros(n,1);
% r = b - A * x;
% for k=1:1000
%     if r == 0
%         break
%     end
%     alpha = r'*r/(r'*A*r);
%     x = x + alpha * r;
%     r = b - A * x;

```

```

% end
% end
%
% Results re-organization
y1=y1';
final=[top];
for j=1:N2
    final=[final;[left(j),y1((j-1)*N1+1:j*N1),right(j)]];
end
results1=[final;bottom];
% y2=y2';
% final=[left];
% for j=1:N
%     final=[final;[bottom(j),y2((j-1)*N+1:j*N),top(j)]];
% end
% results2=[final;right];
end

```