

高温作业专用服装设计

摘要

在高温环境下工作时，人们需要穿着专用服装以避免灼伤。专用服装通常由三层织物材料构成，记为I、II、III层，其中I层与外界环境接触，III层与皮肤之间还存在空隙，将此空隙记为IV层。

第一问中，我们首先从传热学的角度出发，考虑热传递的三种方式：热传导、热辐射和热对流，对已经抽象化为五层的热防护服-空气-假人模型，考虑每层的不同情况，分析并简化每个层的三种热传递方式的具体情况，提出合理且有效的热传递方程模型，并利用有限差分法写出其数值算法。通过题目中已有的数据，结合粒子群算将模型中待确定的参数确定出来。

在第二问，我们首先提出人体圆柱模型，对人真实的穿衣情况进行模拟，考虑到轻便性是热防护服同样也非常重要的参数，于是我们选取质量最轻为优化目标。通过与粒子群算法结合，我们得出第二问的最优厚度为 8.69 mm ，

在第三问中，通过前面对人体圆柱模型的分析，我们得出结论：虽然不必考虑空气层的质量和价格，但是空气层的厚度会影响其它层材料的耗费量。于是，我们仍然采用质量最轻作为优化目标。通过粒子群算法我们得出：第三问的第II层和第III层的最优厚度分别为 11.23 mm 与 6.92 mm 。

最后在模型优化中，我们重新考虑了推翻假设：温度不影响影响热传导率，通过提出有效温度传导率和有效比热容来替代本来模型里的热传导率、比热容，更贴近真实应用的需求。

关键词：多介质热传导模型、有限差分法、目标优化、粒子群算法

一、 问题重述

在高温环境下工作时，人们需要穿着专用服装以避免灼伤。专用服装通常由三层织物材料构成，记为I、II、III层，其中I层与外界环境接触，III层与皮肤之间还存在空隙，将此空隙记为I₅层。

为设计专用服装，将体内温度控制在 37°C 的假人放置在实验室的高温环境中，测量假人皮肤外侧的温度。为了降低研发成本、缩短研发周期，请你们利用数学模型来确定假人皮肤外侧的温度变化情况，并解决以下问题：

(1) 专用服装材料的某些参数值由附件1给出，对环境温度为 75°C 、II层厚度为 $6\text{ }\mu\text{m}$ 、I₅层厚度为 $5\text{ }\mu\text{m}$ 、工作时间为90分钟的情形开展实验，测量得到假人皮肤外侧的温度（见附件2）。建立数学模型，计算温度分布，并生成温度分布的Excel文件（文件名为 $\pi\rho\beta\lambda\epsilon\mu 1.\xi\lambda\sigma\xi$ ）。

(2) 当环境温度为 65°C 、I₅层的厚度为 $5.5\text{ }\mu\text{m}$ 时，确定II层的最优厚度，确保工作60分钟时，假人皮肤外侧温度不超过 47°C ，且超过 44°C 的时间不超过5分钟。

(3) 当环境温度为 80°C 时，确定II层和I₅层的最优厚度，确保工作30分钟时，假人皮肤外侧温度不超过 47°C ，且超过 44°C 的时间不超过5分钟。

附件1. 专用服装材料的参数值

附件2. 假人皮肤外侧的测量温度

二、 问题的分析

第一问，要求我们建立有关假人皮肤外侧的温度的数学模型，并提供了一组某条件下的实验数据。这显然是一个模型的反问题：通过提供的数据与建立的模型相配合，进行参数优化得出模型中欠缺的参数的真实值

第二、三问，要求我们对某一特定要求进行热防护服的设计，也就是有关材料的最佳厚度。这是两道典型的规划问题：而最重要的是目标函数怎么求。这需要我们联系人体圆柱模型进行分析。

三、 模型假设

- (1) 热传递沿着垂直于防护服，因此可以视为一维系统。
- (2) 三层织物中的每一层都是性质相同且均一的，具有稳定的理化参数。
- (3) 热源的能量在到达第 I 层时，包含热对流和热辐射。但第 I 层会阻挡大部分的辐射，因此第 I 层和第 I 层的热传递时忽略热辐射。
- (4) 由于 $L_{IV} \leq 6.4$ ，在该空气层厚度的条件下热对流影响较小，故不考虑热对流。
- (5) 织物层之间、织物与空气层之间、空气层与假人皮肤之间的温度分布是连续变化的，但是温度梯度是跳跃的。
- (6) 假人是一个均质物体，具有稳定的理化参数。
- (7) $37^{\circ}\text{C} - 85^{\circ}\text{C}$ 之间的温差变化不算超高温度变化，不足以影响热传导率和显热容。
- (8) 在没有实验之前，这个系统保持一致均一稳定温度。

四、 符号与约定

序号	参数	意义
1	C_p^A	第 p 层的显热容，p 可为 I, II, III, IV, V
2	F_L	第一层向左侧的热辐射量
3	F_R	第一层向右侧的热辐射量
4	t_{exp}	热源下的暴露时间
5	T_p	第 p 层的温度
6	k_p	第 p 层的传热系数，p 可为 I, II, III, IV, V
7	x	水平坐标
8	β_p	第 p 层的辐射吸收常数，p 可为 I, II, III, IV, V
9	σ	斯蒂芬-玻尔兹曼常数
10	L_{fab}	热防护层的厚度
11	ξ_1	第 I 层与环境热源接触面的辐射率
12	ξ_2	第 I 层与第 II 层接触面的辐射率

13	q_{air}	第 III 层和空气层接触面的热辐射密度
14	κ	空气层的吸收系数
15	ε_{III}	第 III 层与空气层接触面的放射率
16	ε_V	第 V 层与空气层接触面的放射率
17	q_{conv}	热源到到第 I 层的热对流密度
18	q_{rad}	热源到到第 I 层的热辐射密度
19	T_g	外界的温度
20	$h_{c,I}$	环境热源与第 I 层外表面之间的热对流系数
21	Nu	努塞尔特数

五、模型的建立和求解

5.1 问题一模型的建立与求解

5.1.1 问题的分析和规划模型的提出

对于第一个问题，我们应该先建立起关于热防护服-空气-假人的体系的热力学模型，通过提供的实验参数获得模型中需要估计的参数量。在估计参数量的时候可以被看成是一个规划模型：

$$\min \quad \frac{1}{2} [T_{theory} - T_{experiment}]^2$$

其中 T_{theory} 是我们模型的理论值， $T_{experiment}$ 是题目附件中提供的实验值。

在我们求取该优化问题的时候，采用粒子群算法进行优化。

5.1.2 热防护服-空气-假人的体系的建立及传热学分析

我们将题目中描述的热防护服-空气-假人的体系抽象为图中的一维系统。如图1所示，整个系统被分为五层：前四层分别是附件中提到的三层热防护服材料和空气层，最后一层是假人的橡胶层皮肤，在这里我们假设它具有和人体皮肤有一样的理化性质。

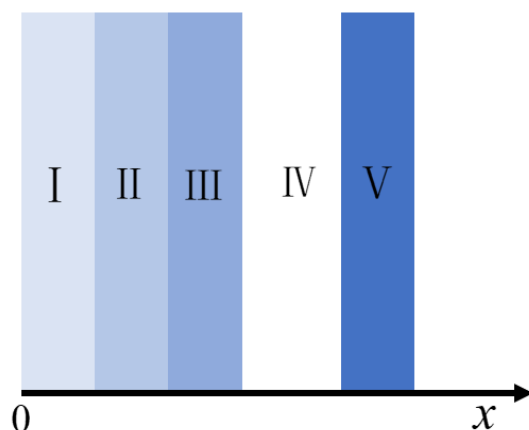


图1 热防护服-空气-假人系统

根据附件的材料，热防护服-空气-假人系统需满足以下条件：

$$\begin{aligned} L_I &= 0.6, \quad 0.6 \leq L_{II} \leq 25, \quad L_{III} = 3.6, \quad 0.4 \leq L_{IV} \leq 6.4; \\ k_I &= 0.082, \quad k_{II} = 0.037, \quad k_{III} = 0.045, \quad k_{IV} = 0.028; \\ c_I &= 1377, \quad c_{II} = 2100, \quad c_{III} = 1726, \quad c_{IV} = 1005; \\ \rho_I &= 300, \quad \rho_{II} = 862, \quad \rho_{III} = 74.2, \quad \rho_{IV} = 1.18; \end{aligned}$$

其中， L_p 、 k_p 、 c_p 、 ρ_p 分别是第 p 层的厚度 (mm)、热传导率 ($W/(m \cdot K)$)、比热 ($J/(kg \cdot K)$)、密度 (kg/m^3)， $p \in \{I, II, III, IV, V\}$ 。

在热防护服-空气-假人系统中，我们进行如下热力学分析与假设的提出：

(1) 热传递沿着垂直于防护服，也即图中的 x 轴进行，因此可以视为一维系统。

(2) 三层织物中的每一层都是性质相同且均一的，具有稳定的理化参数。

(3) 热源的能量在到达第 I 层时，包含热对流和热辐射。但第 I 层会阻挡大部分的辐射，因此第 I 层和第 I 层的热传递时忽略热辐射。

(4) 由于 $L_{IV} \leq 6.4$ ，在该空气层厚度的条件下热对流影响较小，故不考虑热对流。

(5) 织物层之间、织物与空气层之间、空气层与假人皮肤之间的温度分布是连续变化的，但是温度梯度是跳跃的。

(6) 假人是一个均质物体，具有稳定的理化参数。

(7) $37^{\circ}\text{C}-85^{\circ}\text{C}$ 之间的温差变化不算超高温度变化，不足以影响热传导率和显热容。

(8) 在没有实验之前，这个系统保持一致均一稳定温度。

5.1.3 热防护服-空气-假人的系统的传热学模型和方程的建立

据上面的八个假设，对于热防护服，我们提出如下的热传递模型^[2]：

$$C_I^A \frac{\partial T_I}{\partial t} = k_I \frac{\partial^2 T_I}{\partial x^2} + \frac{\partial F_L}{\partial x} - \frac{\partial F_R}{\partial x} \quad (x, t) \in \Omega_I \bullet (0, t_{\text{exp}}).$$

$$C_{II}^A \frac{\partial T_{II}}{\partial t} = k_{II} \frac{\partial^2 T_{II}}{\partial x^2} \quad (x, t) \in \Omega_{II} \bullet (0, t_{\text{exp}}).$$

$$C_{III}^A \frac{\partial T_{III}}{\partial t} = k_{III} \frac{\partial^2 T_{III}}{\partial x^2} \quad (x, t) \in \Omega_{III} \bullet (0, t_{\text{exp}}).$$

其中， C_p^A 是第 p 层的显热容 ($\text{kJ} / (\text{m}^3 \cdot \text{K})$)，其满足如下公式：

$$C_p^A = \rho_p \cdot c_p;$$

$T_p(x, t)$ 是第 p 层的温度 (K) 函数； t 为时间 (s)； x 为水平坐标； F_L 和 F_R 分别是第 I 层向左和向右的辐射量 ($\text{kJ} / \text{m}^2 \cdot \text{s}$)； t_{exp} 是暴露在热源下的时间 (s)； Ω_p 表示第 p 层在水平坐标 x 上的取值范围， $\Omega_I = (0, L_I)$ ， $\Omega_{II} = (L_I, L_I + L_{II})$ ， $\Omega_{III} = (L_I + L_{II}, L_{fab})$ ； L_{fab} 是热防护的厚度 (mm)， $L_{fab} = L_I + L_{II} + L_{III}$ 。

上述方程中的辐射量满足：

$$\frac{\partial F_L}{\partial x} = \beta_I F_L - \beta_I \sigma T^4 \quad (x, t) \in \Omega_I \bullet (0, t_{\text{exp}}).$$

$$\frac{\partial F_R}{\partial x} = -\beta_I F_R + \beta_I \sigma T^4 \quad (x, t) \in \Omega_I \bullet (0, t_{\text{exp}}).$$

$$(1 - \xi_I) F_L(0, t) + \xi_I \sigma T^4 = F_R(0, t)$$

$$(1 - \xi_2) F_L(L_I, t) + \xi_2 \sigma T^4(L_I, t) = F_R(0, t)$$

其中， β_I 是第 I 层的辐射吸收常数 (K^{-1})； σ 为斯蒂芬-玻尔兹曼常数， $\sigma = 5.670 \times 10^{-8} \text{W} / (\text{m}^2 \cdot \text{K}^4)$ ； ξ_I 是第 I 层与环境热源接触面的辐射率 ($\text{W} / (\text{s} \cdot \text{m}^2)$)； ξ_2 是第 II 层与第 III 层接触面的辐射率 ($\text{W} / (\text{s} \cdot \text{m}^2)$)。

对于第 III 层和第 IV 层（空气层）之间，我们提出如下空气层热传递模型：

$$C_{IV}^A \frac{\partial T_{IV}}{\partial t} = k_{IV} \frac{\partial^2 T_{IV}}{\partial x^2} - \frac{\partial q_{air}}{\partial x} \quad (x, t) \in \Omega_{IV} \bullet (0, t_{exp}).$$

其中， q_{air} 是第 III 层和空气层接触面的热辐射密度 (W / m^2) ； $\Omega_{IV} = (L_{fab}, L_{fab} + L_{IV})$.

关于空气层热传递模型我们可以做出如下简化：考虑将空气层作为不发射辐射的辐射介质，么 q_{air} 可以由辐射热传递方程 (RTE) 来计算，即 *Beer* 定律：假设空气层吸收热辐射呈指数衰减，其计算方程如下所示^[1]：

$$q_{air} = q_{air}|_{x=L_{fab}} \cdot e^{-\kappa x} ..$$

其中， κ 是空气层的吸收系数； $q_{air}|_{x=L_{fab}}$ 可由以下公式计算：

$$q_{air}|_{x=L_{fab}} = \frac{\sigma(T_{III}^4(L_{fab}, t) - T_V^4(L_{fab} + L_{IV}, t))}{\frac{1}{\varepsilon_{III}} + \frac{1}{\varepsilon_V} - 1}.$$

公式中， ε_{III} 和 ε_V 分别是指第 III 层与空气层接触面的放射率 ($w / sr^1 \cdot m^2$) 、第 V 层与空气层接触面的放射率 ($w / sr \cdot m^2$) 。

对于第 IV 层（空气层）和第 V 层之间，我们提出如下热传递模型：

$$C_V^A \frac{\partial T_V}{\partial t} = k_V \frac{\partial^2 T_V}{\partial x^2} \quad (x, t) \in \Omega_V \bullet (0, t_{exp}).$$

接下来我们开始讨论模型的边界条件问题。讨论之前，我们先申明以下初值问题：

$$T(x, 0) \quad x \in (0, L_{fab} + L_{IV} + L_V).$$

对于第 I 层与环境热源的接触面，需满足以下边界条件：

$$-k_I \frac{\partial T_I}{\partial x} \Big|_{x=0} = (q_{conv} + q_{rad})|_{x=0} ..$$

其中， q_{conv} 是热源到第 I 层的热对流密度 (W / m^2) ； q_{rad} 是热源到第 I 层的热辐射密度 (W / m^2) 。

从热源到第 I 层的辐射和对流量可以被描述为：

$$(q_{conv} + q_{rad})|_{x=0} = h_{c,I} (T_g - T(0, t)) ,$$

其中， T_g 为外界的温度 (K) ； $h_{c,I}$ 为环境热源与第 I 层外表面之间的热对流系数 ($W / (m^2 \cdot K)$) ，计算方式如下：

$$h_{c,1} = \text{Nu} \frac{k_1}{L_1}, \quad \text{Nu 是努塞尔特数.}$$

结合两处，可有新的限制条件：

$$-k_1 \frac{\partial T_1}{\partial x} \Big|_{x=0} = (q_{conv} + q_{rad}) \Big|_{x=0}.$$

对于第 I 层和第 II 层的接触面，我们提出以下边界条件需要被满足：

$$T_1(L_1, t) = T_{II}(L_1, t).$$

$$-k_1 \frac{\partial T_1}{\partial x} \Big|_{x=L_1} = -k_{II} \frac{\partial T_{II}}{\partial x} \Big|_{x=L_1}.$$

对于第 II 层和第 III 层的接触面，我们提出以下边界条件需要被满足：

$$T_{II}(L_1 + L_{II}, t) = T_{III}(L_1 + L_{II}, t)$$

$$-k_{II} \frac{\partial T_{II}}{\partial x} \Big|_{x=L_1 + L_{II}} = -k_{III} \frac{\partial T_{III}}{\partial x} \Big|_{x=L_1 + L_{II}}.$$

对于第 III 层和第 IV 层的接触面，我们提出以下边界条件需要被满足：

$$T_{III}(L_{fab}, t) = T_{IV}(L_{fab}, t),$$

$$-k_{III} \frac{\partial T_{III}}{\partial x} \Big|_{x=L_{fab}} = (q_{air} - k_{IV} \frac{\partial T_{IV}}{\partial x}) \Big|_{x=L_{fab}}.$$

对于第 IV 层和第 V 层的接触面，我们提出以下边界条件需要被满足：

$$T_{IV}(L_{fab} + L_{IV}, t) = T_V(L_{fab} + L_{IV}, t),$$

$$(q_{air} - k_{IV} \frac{\partial T_{IV}}{\partial x}) \Big|_{x=L_{fab} + L_{IV}} = -k_V \frac{\partial T_V}{\partial x} \Big|_{x=L_{fab} + L_{IV}}.$$

以上式子均服从热力原理与我们对于热防护服-空气-假人系统的假设。

结合上述分析，我们获得如下关于热防护服-空气-假人系统的热传递偏微分方程组：

$$\left\{ \begin{array}{ll} C_I^A \frac{\partial T_I}{\partial t} = k_I \frac{\partial^2 T_I}{\partial x^2} + \frac{\partial F_L}{\partial x} - \frac{\partial F_R}{\partial x} & (x, t) \in \Omega_I \bullet (0, t_{\text{exp}}) \\ \frac{\partial F_L}{\partial x} = \beta_I F_L - \beta_I \sigma T_I^4 & (x, t) \in \Omega_I \bullet (0, t_{\text{exp}}) \\ \frac{\partial F_R}{\partial x} = -\beta_I F_R + \beta_I \sigma T_I^4 & (x, t) \in \Omega_I \bullet (0, t_{\text{exp}}) \\ C_{II}^A \frac{\partial T_{II}}{\partial t} = k_{II} \frac{\partial^2 T_{II}}{\partial x^2} & (x, t) \in \Omega_{II} \bullet (0, t_{\text{exp}}) \\ C_{III}^A \frac{\partial T_{III}}{\partial t} = k_{III} \frac{\partial^2 T_{III}}{\partial x^2} & (x, t) \in \Omega_{III} \bullet (0, t_{\text{exp}}) \\ C_{IV}^A \frac{\partial T_{IV}}{\partial t} = k_{IV} \frac{\partial^2 T_{IV}}{\partial x^2} - \frac{\partial q_{\text{air}}}{\partial x} & (x, t) \in \Omega_{IV} \bullet (0, t_{\text{exp}}) \\ q_{\text{air}} = q_{\text{air}}|_{x=L_{fab}} \cdot e^{-\kappa x} & (x, t) \in \Omega_{IV} \bullet (0, t_{\text{exp}}) \\ C_V^A \frac{\partial T_V}{\partial t} = k_V \frac{\partial^2 T_V}{\partial x^2} & (x, t) \in \Omega_V \bullet (0, t_{\text{exp}}) \end{array} \right.$$

满足以下边界条件:

$$\left\{ \begin{array}{l} T(x, 0) \quad x \in (0, L_{fab} + L_{IV} + L_V) \\ -k_I \frac{\partial T_I}{\partial x} \Big|_{x=0} = (q_{\text{conv}} + q_{\text{rad}}) \Big|_{x=0} = h_{c,l} (T_g - T_I(0, t)) \\ (1 - \xi_1) F_L(0, t) + \xi_1 \sigma T_I^4 = F_R(0, t) \\ (1 - \xi_2) F_L(L_I, t) + \xi_2 \sigma T_I^4(L_I, t) = F_R(0, t) \\ T_I(L_I, t) = T_{II}(L_I, t) \\ -k_I \frac{\partial T_I}{\partial x} \Big|_{x=L_I} = -k_{II} \frac{\partial T_{II}}{\partial x} \Big|_{x=L_I} \\ T_{II}(L_I + L_{II}, t) = T_{III}(L_I + L_{II}, t) \\ -k_{II} \frac{\partial T_{II}}{\partial x} \Big|_{x=L_I + L_{II}} = -k_{III} \frac{\partial T_{III}}{\partial x} \Big|_{x=L_I + L_{II}} \\ T_{III}(L_{fab}, t) = T_{IV}(L_{fab}, t) \\ -k_{III} \frac{\partial T_{III}}{\partial x} \Big|_{x=L_{fab}} = (q_{\text{air}} - k_{IV} \frac{\partial T_{IV}}{\partial x}) \Big|_{x=L_{fab}} \\ q_{\text{air}}|_{x=L_{fab}} = \frac{\sigma(T_{III}^4(L_{fab}, t) - T_V^4(L_{fab} + L_{IV}, t))}{\frac{1}{\varepsilon_{III}} + \frac{1}{\varepsilon_V} - 1} \\ T_{IV}(L_{fab} + L_{IV}, t) = T_V(L_{fab} + L_{IV}, t) \\ (q_{\text{air}} - k_{IV} \frac{\partial T_{IV}}{\partial x}) \Big|_{x=L_{fab} + L_{IV}} = -k_V \frac{\partial T_V}{\partial x} \Big|_{x=L_{fab} + L_{IV}} \end{array} \right.$$

5.1.4 传热学方程模型的数值算法

对于一个复杂的偏微分方程系统，我们考虑用有限差分的方法来解决这一问题^[1]：将整个系统的时间离散成 $N+1$ 个点，分别为 $0=t_0 < t_1 < \dots < t_N = t_{\text{exp}}$ ，令 $t_j = j\tau$ ， $j=0, \dots, N$ ，

$\tau = \frac{t_{\text{exp}}}{N}$ ；将第 p 层分割的空间离散为 M_p+1 个点，分别为

$$x_{M_{p-1}} = \sum_{j=1}^{p-1} L_j = x_{0_p} < x_{1_p} < \dots < x_{M_p} = \sum_{j=1}^p L_j, \text{ 令 } x_{i_p} = i_p \cdot h_p, i_p = 0_p, 1_p, \dots, M_p, h_p = \frac{L_p}{M_p}。$$

对于第 I 层及其左边界，我们有偏微分方程组如下：

$$\left\{ \begin{array}{l} C_1^A \frac{\partial T_1}{\partial t} = k_1 \frac{\partial^2 T_1}{\partial x^2} + \frac{\partial F_L}{\partial x} - \frac{\partial F_R}{\partial x} \quad (x, t) \in \Omega_1 \bullet (0, t_{\text{exp}}) \\ T(x, 0) \quad x \in (0, L_{fab} + L_{IV} + L_V) \\ \frac{\partial F_L}{\partial x} = \beta_1 F_L - \beta_1 \sigma T_1^4 \quad (x, t) \in \Omega_1 \bullet (0, t_{\text{exp}}) \\ \frac{\partial F_R}{\partial x} = -\beta_1 F_R + \beta_1 \sigma T_1^4 \quad (x, t) \in \Omega_1 \bullet (0, t_{\text{exp}}) \\ -k_1 \frac{\partial T_1}{\partial x} \Big|_{x=0} = (q_{\text{conv}} + q_{\text{rad}}) \Big|_{x=0} = h_{c,1} (T_g - T(0, t)) \\ (1 - \xi_1) F_L(0, t) + \xi_1 \sigma T_1^4 = F_R(0, t) \\ (1 - \xi_2) F_L(L_1, t) + \xi_2 \sigma T_1^4(L_1, t) = F_R(0, t) \end{array} \right.$$

由假设可有： $T(x, 0) = 310.15K$ 。

令 $\Theta = \frac{\partial F_L}{\partial x} - \frac{\partial F_R}{\partial x}$ ，结合边界条件中的两个初值有：

$$\Theta(T) = -\beta_1^2 \sigma e^{\beta_1 x} \left[\int_0^x e^{-\beta_1 y} T_1(y, t) dy_1 + c_2 \right] + \beta_1^2 \sigma e^{-\beta_1 x} \left[\int_0^x e^{\beta_1 y} T_1(y, t) dy_1 + c_1 \right] - 2\beta_1 \sigma T_1^4(x_1, t)$$

其中，

$$c_1 = \frac{\xi_1}{\beta_1} T_1^4 - (1 - \xi_1) c_2,$$

$$c_2 = \frac{1}{(1-\xi_2)\beta_1(1-\xi_1)e^{-\beta_1 L} - \beta_1 e^{-\beta_1 L}}.$$

$$[(1-\xi_2)\beta_1 e^{-\beta_1 L} \int_0^L e^{\beta_1 x} T_1(x, t) dx + \beta_1 e^{\beta_1 L} \cdot \int_0^L e^{\beta_1 x} T_1(x, t) dx + (1-\xi_2)\xi_1 e^{-\beta_1 L} T_1^4(0, t) + \xi_2 T(L, y)]$$

$$L \in [0, L_1].$$

对上述体系进行离散化，我们有以下隐式迭代形式：

$$C_1^A \frac{T_1(i_1, j+1) - T_1(i_1, j)}{\tau} = k_1 \cdot \frac{T_1(i_1+1, j-1) - 2T_1(i_1, j+1) + T_1(i_1+1, j+1)}{h_1^2} + \Theta(i_1, j+1)$$

其中， $i_1 = 1, 2, \dots, M_1 - 1$, $j = 0, 1, \dots, N - 1$.

$$\text{令 } s = \frac{\tau \cdot k_1}{h_1^2 \cdot C_1^A}, \quad r = \frac{\tau}{C_1^A}, \quad \text{于是有:}$$

$$-sT_1(i_1-1, j+1) + (1+2s)T_1(i_1, j+1) - sT_1(i_1+1, j+1) = T_1(i_1, j) + r\Theta(i_1, j+1)$$

为了方便计算，用 $\Theta(i_1, j)$ 代替 $\Theta(i_1, j+1)$ ，对于 $\Theta(i_1, j)$ 可由辛普森插值型数值积分公式计算：

$$\Theta(i_1, j) = \beta_1^2 \sigma e^{-\beta_1 x_i} [F_1(i_1, j) + c_2(i_1, j)] - \beta_1^2 \sigma e^{\beta_1 x_i} [F_2(i_1, j) + c_1(i_1, j)]$$

$$\left\{ \begin{array}{l} F_1(i_1, j) = \frac{h_1}{3} [f(0, j) + f_1(i_1, j) + 4 \sum_{k=1_1}^{\frac{i_1+1}{2}} f(2k-1, j) + 2 \sum_{k=1_1}^{\frac{i_1-1}{2}} f(2k, j)] \\ F_2(i_1, j) = \frac{h_1}{3} [g(0, j) + g_1(i_1, j) + 4 \sum_{k=1_1}^{\frac{i_1+1}{2}} g(2k-1, j) + 2 \sum_{k=1_1}^{\frac{i_1-1}{2}} g(2k, j)] \end{array} \right.$$

其中： $f(0, j) = e^{-\beta_1 x_0} T_1^4(0, j)$, $f(i_1, j) = e^{-\beta_1 x_i} T_1^4(i_1, j)$,

$$g(0, j) = e^{\beta_1 x_0} T_1^4(0, j) , \quad g(i_1, j) = e^{\beta_1 x_i} T_1^4(i_1, j) ,$$

$$c_1(i_1, j) = \frac{\xi_1}{\beta_1} T_1^4(0, j) - (1-\xi_1)c_2(i_1, j) ,$$

$$c_2 = \frac{1}{(1-\xi_2)\beta_1(1-\xi_1)e^{-\beta_1 L} - \beta_1 e^{-\beta_1 L}}.$$

$$[(1-\xi_2)\beta_1 e^{-\beta_1 L} F + \beta_1 e^{\beta_1 L} \cdot F_1(M_1 + M_2, j) + (1-\xi_2)\xi_1 e^{-\beta_1 L} T_1^4(0, j) + \xi_2 T^4(M_1 + M_2, j)]$$

第I层的左边界离散为：

$$-k_1 \frac{T_1(1_1, j+1) - T_1(0_1, j+1)}{h_1} = h_{c,1} (T_g - T_1(0_1, j+1))$$

同理：

对于第 II 层及其左边界，我们有偏微分方程：

$$\begin{cases} C_{II}^A \frac{\partial T_{II}}{\partial t} = k_{II} \frac{\partial^2 T_{II}}{\partial x^2} & (x, t) \in \Omega_{II} \bullet (0, t_{\text{exp}}) \\ T_I(L_I, t) = T_{II}(L_I, t) \\ -k_I \frac{\partial T_I}{\partial x} \Big|_{x=L_I} = -k_{II} \frac{\partial T_{II}}{\partial x} \Big|_{x=L_I} \end{cases}$$

可离散化为：

$$C_{II}^A \frac{T_{II}(i_{II}, j+1) - T_{II}(i_{II}, j)}{\tau} = k_{II} \frac{T_{II}(i_{II}+1, j-1) - 2T_{II}(i_{II}, j+1) + T_{II}(i_{II}+1, j+1)}{h_{II}^2}$$

$$T_I(M_I+1, j+1) = T_{II}(0_{II}, j+1)$$

$$-k_I \frac{T_I(M_I+1, j+1) - T_I(M_I, j+1)}{h_I} = -k_{II} \frac{T_{II}(1_{II}, j+1) - T_{II}(0_{II}, j+1)}{h_{II}}$$

对于第 III 层及其左边界，我们有偏微分方程：

$$\begin{cases} C_{III}^A \frac{\partial T_{III}}{\partial t} = k_{III} \frac{\partial^2 T_{III}}{\partial x^2} & (x, t) \in \Omega_{III} \bullet (0, t_{\text{exp}}) \\ T_{II}(L_I + L_{II}, t) = T_{III}(L_I + L_{II}, t) \\ -k_{II} \frac{\partial T_{II}}{\partial x} \Big|_{x=L_I+L_{II}} = -k_{III} \frac{\partial T_{III}}{\partial x} \Big|_{x=L_I+L_{II}} \end{cases}$$

可离散化为：

$$C_{III}^A \frac{T_{III}(i_{III}, j+1) - T_{III}(i_{III}, j)}{\tau} = k_{III} \frac{T_{III}(i_{III}+1, j-1) - 2T_{III}(i_{III}, j+1) + T_{III}(i_{III}+1, j+1)}{h_{III}^2}$$

$$T_{II}(M_{II}+1, j+1) = T_{III}(0_{III}, j+1)$$

$$-k_{II} \frac{T_{II}(M_{II}+1, j+1) - T_{II}(M_{II}, j+1)}{h_{II}} = -k_{III} \frac{T_{III}(1_{III}, j+1) - T_{III}(0_{III}, j+1)}{h_{III}}$$

对于第 IV 层及其左边界，我们有偏微分方程：

$$\left\{ \begin{array}{l} C_{IV}^A \frac{\partial T_{IV}}{\partial t} = k_{IV} \frac{\partial^2 T_{IV}}{\partial x^2} - \frac{\partial q_{air}}{\partial x} \quad (x, t) \in \Omega_{IV} \bullet (0, t_{\exp}) \\ q_{air} = q_{air}|_{x=L_{fab}} \cdot e^{-\kappa x} \quad (x, t) \in \Omega_{IV} \bullet (0, t_{\exp}) \\ T_{III}(L_{fab}, t) = T_{IV}(L_{fab}, t) \\ -k_{III} \frac{\partial T_{III}}{\partial x} \Big|_{x=L_{fab}} = (q_{air} - k_{IV} \frac{\partial T_{IV}}{\partial x}) \Big|_{x=L_{fab}} \\ q_{air}|_{x=L_{fab}} = \frac{\sigma(T_{III}^4(L_{fab}, t) - T_V^4(L_{fab} + L_{IV}, t))}{\frac{1}{\varepsilon_{III}} + \frac{1}{\varepsilon_V} - 1} \end{array} \right.$$

可离散化为：

$$C_{IV}^A \frac{T_{IV}(i_{IV}, j+1) - T_{IV}(i_{IV}, j)}{\tau} = k_{IV} \frac{T_{IV}(i_{IV} + 1, j-1) - 2T_{IV}(i_{IV}, j+1) + T_{IV}(i_{IV} + 1, j+1)}{h_{IV}^2} + \theta(i, j+1)$$

$$\begin{aligned} T_{III}(M_{III} + 1, j+1) &= T_{IV}(0_{IV}, j+1) \\ -k_{III} \frac{T_{III}(M_{III} + 1, j+1) - T_{III}(M_{III}, j+1)}{h_{III}} &= \frac{\sigma(T_{III}^4(L_{fab}, t) - T_V^4(L_{fab} + L_{IV}, t))}{\frac{1}{\varepsilon_{III}} + \frac{1}{\varepsilon_V} - 1} \\ &\quad - k_{IV} \frac{T_{IV}(1_{IV}, j+1) - T_{IV}(0_{IV}, j+1)}{h_{IV}} \end{aligned}$$

其中：

$$\theta(i, j+1) = \frac{\left[\frac{\sigma(T_{III}^4(M_{III}, j+1) - T_V^4(0_{III}, j+1))}{\frac{1}{\varepsilon_{III}} + \frac{1}{\varepsilon_V} - 1} e^{-\kappa x_{i_{IV}+1}} - \frac{\sigma(T_{III}^4(M_{III}, j+1) - T_V^4(0_{III}, j+1))}{\frac{1}{\varepsilon_{III}} + \frac{1}{\varepsilon_V} - 1} e^{-\kappa x_{i_{IV}}} \right]}{h_{IV}}$$

对于第 V 层及其左边界，我们有偏微分方程：

$$\left\{ \begin{array}{l} C_V^A \frac{\partial T_V}{\partial t} = k_V \frac{\partial^2 T_V}{\partial x^2} \quad (x, t) \in \Omega_V \bullet (0, t_{\exp}) \\ T_{IV}(L_{fab} + L_{IV}, t) = T_V(L_{fab} + L_{IV}, t) \\ (q_{air} - k_{IV} \frac{\partial T_{IV}}{\partial x}) \Big|_{x=L_{fab} + L_{IV}} = -k_V \frac{\partial T_V}{\partial x} \Big|_{x=L_{fab} + L_{IV}} \end{array} \right.$$

可离散化为:

$$C_V^A \frac{T_V(i_V, j+1) - T_V(i_V, j)}{\tau} = k_V \frac{T_V(i_V + 1, j-1) - 2T_V(i_V, j+1) + T_V(i_V + 1, j+1)}{h_V^2}$$

$$T_{IV}(M_{IV} + 1, j+1) = T_V(0_V, j+1)$$

$$\frac{\sigma(T_{III}^4(L_{fab}, t) - T_V^4(L_{fab} + L_{IV}, t))}{\frac{1}{\varepsilon_{III}} + \frac{1}{\varepsilon_V} - 1} e^{-\kappa L_{IV}} - k_{IV} \frac{T_{IV}(1_{IV}, j+1) - T_{IV}(0_{IV}, j+1)}{h_{IV}} = -k_V \frac{T_V(1_V, j+1) - T_V(0_V, j+1)}{h_V}$$

至此，系统的数值离散处理完毕。

5.1.5 粒子群（PSO）优化算法

粒子群是模拟群鸟觅食的模型，它在寻找全局最优的应用上具有十分广泛的作用。在群鸟觅食的模型中。我们把每一只鸟儿都看作一个粒子，那么整个鸟群可以被看成一个粒子群。设在某一个 D 维的目标搜索空间中，有 m 个粒子所组成的粒子群，其中第 i ($i=1, 2, \dots, m$) 个粒

子正处于位置 $x_i = (x_i^1, x_i^2, \dots, x_i^D)$ ，这个位置可能正好是一个最优解，将 x_i 带入目标函数就可以得出其适应值，通过适应值的大小可以判断出这个位置的优劣。假设个体粒子经过的最好位置是 $P_i = (p_i^1, p_i^2, \dots, p_i^D)$ ，整体粒子群所经过最好的位置是 $P_g = (p_g^1, p_g^2, \dots, p_g^D)$ 。第

i 粒子的速度记为 $S_i = (s_i^1, s_i^2, \dots, s_i^D)$ 。

在粒子群算法运行中，用以下公式对粒子的位置进行不断的更新：

$$s_i^d = \omega s_i^d + c_1 r_1 (p_i^d - x_i^d) + c_2 r_2 (p_g^d - x_i^d)$$

$$x_i^d = x_i^d + \alpha s_i^d$$

其中， $i=1, 2, \dots, m$; $d=1, 2, \dots, D$; 惯性因子 $\omega \geq 0$ ；加速常数 $c_1, c_2 \geq 0$ ；

r_1, r_2 是在 $[0,1]$ 内变化的随机数；约束因子 α 控制速度的权重； $s_i^d \in [-s_{\max}^d, s_{\max}^d]$ ，表明粒子的速度被最大速度限制，这个最大速度可以人为设定。最后，粒子群位置的改变，也即迭代次数可以被人为设定。

5.1.6 结果

在结合粒子群算法与有限差分法，我们得到了整个系统的温度分布图，如下图所示。

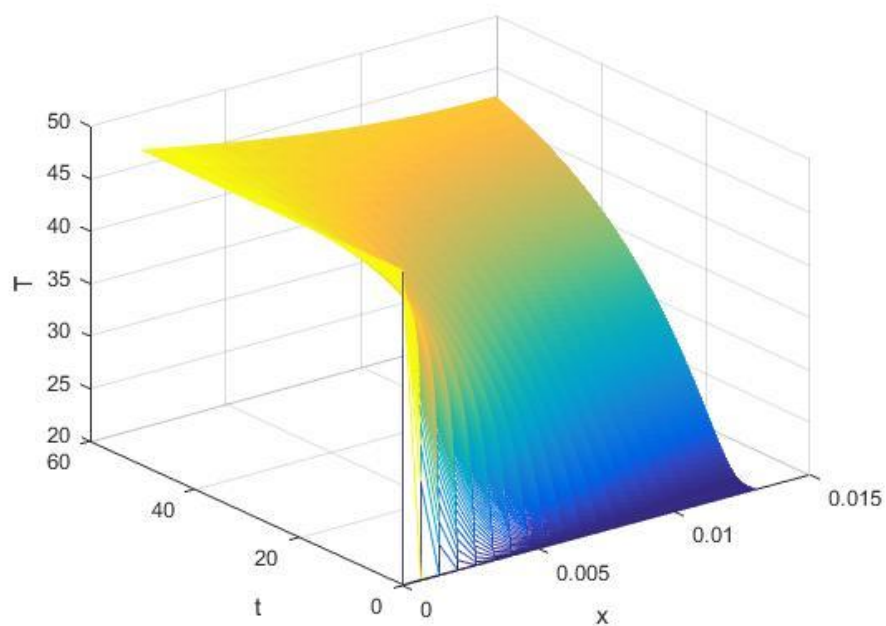


图 2 温度分布图

还有一些存在于系统中未知不知道的参数，如下表所示：

参数	真实值
L_{fab}	0.982
ξ_1	3.25
ξ_2	4.25
q_{air}	5.52
ε_{III}	7.5
ε_V	0.25
q_{conv}	0.35
q_{rad}	0.52
$h_{c,l}$	0.45

5.2 问题二的模型建立与求解

5.2.1 人体圆柱模型的提出与问题二的规划提出

我们将人近似为一个圆柱体。那么热防护服就是包裹在圆柱体外侧的衣物，也即圆柱的侧表面。

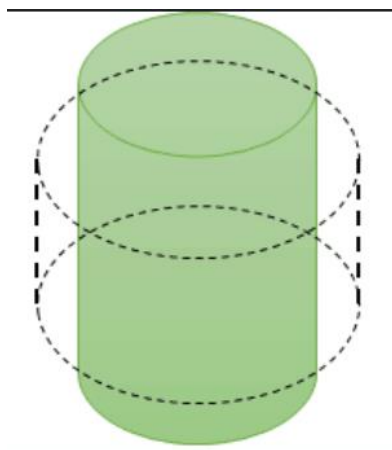


图 3 人体圆柱模型

圆柱体有一些基本参数如下：

H：圆柱体的高度，R：圆柱体的底面半径

这两个参数分别相当于人体的高度和人体的最大横截面半径。根据人体一般情况，我们假设两个参数的值分别为：1.7m和0.4m。

在第二问中我们考虑优化目标分别是：热防护服的质量最轻，因为对于热防护服来说，其轻便型十分重要。

该规划目标的函数应该为：

$$\begin{aligned} \min \quad m = & \rho_1 \cdot L_1 \cdot 2\pi(L + L_{IV} + L_{III} + L_{II} + L_I) \cdot H + \rho_{II} \cdot L_{II} \cdot 2\pi(L + L_{IV} + L_{III} + L_{II}) \cdot H \\ & + \rho_{III} \cdot L_{III} \cdot 2\pi(L + L_{IV} + L_{III}) \cdot H \end{aligned}$$

由于第二问空气层的厚度被确定了，所以该问题也被等价于：

$$\min \quad L_{II}$$

其满足限制条件为：

$$T(L_I + L_{II} + L_{III} + L_{IV}, 60 \text{ min}) \leq 47^\circ\text{C}$$

$$T(L_I + L_{II} + L_{III} + L_{IV}, 55 \text{ min}) \leq 44^\circ\text{C}$$

5.2.2 求解结果

结合粒子群算法，最优厚度为 8.69mm.

5.3 问题三的模型建立与求解

5.3.1 问题的分析和规划模型的提出

通过第二问对于人圆柱模型的分析，我们得出以下结果：空气层的厚度会影响到其他各层的材料体积。在第三问中，我们仍然考虑质量最低这一规划目标，有上面分析可知其规划目标函数为：

$$\begin{aligned} \min \quad m = & \rho_I \cdot L_I \cdot 2\pi(L + L_{IV} + L_{III} + L_{II} + L_I) \cdot H + \rho_{II} \cdot L_{II} \cdot 2\pi(L + L_{IV} + L_{III} + L_{II}) \cdot H \\ & + \rho_{III} \cdot L_{III} \cdot 2\pi(L + L_{IV} + L_{III}) \cdot H \end{aligned}$$

需满足的条件是：

$$\begin{aligned} T(L_I + L_{II} + L_{III} + L_{IV}, 30 \text{ min}) & \leq 47^\circ\text{C} \\ T(L_I + L_{II} + L_{III} + L_{IV}, 25 \text{ min}) & \leq 44^\circ\text{C} \end{aligned}$$

5.3.2 求解结果

结合粒子群算法，第三问的第 II 层和第 IV 层的最优厚度为 11.23 mm 与 6.92 mm。

六、模型评价与改进

7.1 模型的优点

- (1) 清楚地考虑了热传递过程中的三种方式。
- (2) 较为精确的验证了外环境温度变化不大的情况下的模型预测。

7.2 模型的缺点

- (1) 假设较多，有一些不符合实际运用：例如外环境温度变化不大。这点在模型改进中得到修改。
- (2) 皮肤模型处理较精简，没有考虑人体生物结构。

7.3 模型的改进

7.3.1 考虑了高温导致热传导率和显热容变化的模型

考虑到模型的假设基础建立在温度变化不会影响到不足以影响热传导率和显热容的情况下，这是由于题目中给定的条件没有太高的温度值。但是实际应用中热防护服的外界问题一般在 2000K 左右，因此我们在模型改进中考虑了温度对于热传导率和显热容的影响，提出有效温度传导率和有效比热容来替代本来模型立的热传导率和比热容：

$$\begin{aligned} c_p^{eff} &= 1300 + 1.6(T - 300) \\ c_{air}^{eff} &= \begin{cases} 0.026 + 0.000068(T - 300) & T < 700\text{K} \\ 0.053 + 0.000054(T - 700) & T > 700\text{K} \end{cases} \end{aligned}$$

以求做到对温度更为精准的预测，也更加方便真是情况下的使用。

参考文献

- [1] 潘斌. 热防护服装热传递数学建模及参数决定反问题[D]. 浙江理工大学, 2017.
- [2] 卢琳珍, 徐定华, 徐映红. 应用三层热防护服热传递改进模型的皮肤烧伤度预测[J]. 纺织学报, 2018(1):111-118.

附录

附录 1 PDEsolver based on Python

```
import numpy as np
import matplotlib.pyplot as plt
import copy
import sopt
import math

def pdesolver():
    maxIter = 1000
    lenX = 160
    lenY = 5400
    xdelta = 1
    ydelta = 1
    Ttop = 37
    Tbot = 37
    Tleft = 37
    Tright = 37
    Tguess=37
    X, Y = np.meshgrid(np.arange(0, lenX), np.arange(0, lenY))

    T = np.empty((lenX, lenY))
    T.fill(Tguess)

    T[(lenY-1):, :] = Ttop
    T[:, 0] = Tbot
    T[:, (lenX-1):] = Tright
    T[:, :1] = Tleft

    Tlast = copy.deepcopy(T)
```

```

for iteration in range(0, maxIter):
    for i in range(1, lenX-1, xdelta):
        for j in range(1, lenY-1, ydelta):
            if (i/10) in range(0, 0.6):
                T[1][j+1]=(hc1/k1)*(Tg-T[0][j+1])+T[0][j+1]
                F0j=e**(-beta*i)*((T[0][j])**4)
                Fij=e**(-beta*i)*((T[i][j])**4)
                G0j=e**(beta*i)*((T[0][j])**4)
                G0j=e**(beta*i)*((T[i][j])**4)
                F1=F0j+Fij
                F2=G0j+Gij
                for m1 in range(1,math.ceil((i+1)/2)):
                    F1=F1+4*e**(-beta*i)*((T[2m-1][j])**4)
                for m2 in range(1,math.ceil((i-1)/2)):
                    F1=F1++2*e**(-beta*i)*((T[2m2-1][j])**4)
                for m3 in range(1,math.ceil((i+1)/2)):
                    F2=F1+4*e**(beta*i)*((T[2m-1][j])**4)
                for m4 in range(1,math.ceil((i-1)/2)):
                    F2=F1+2*e**(beta*i)*((T[2m2-1][j])**4)
                F1=F1/3
                F2=F2/3
                F3=
                F4=
                for m1 in range(1,66):
                    F3=F1+4*e**(-beta*i)*((T[2m-1][j])**4)
                for m2 in range(1,66):
                    F3=F1++2*e**(-beta*i)*((T[2m2-1][j])**4)
                for m3 in range(1,66):
                    F4=F1+4*e**(beta*i)*((T[2m-1][j])**4)
                for m4 in range(1,66):
                    F4=F1+2*e**(beta*i)*((T[2m2-1][j])**4)
                C2=(1/((1-explo2)*beta*(1-explo1)-beta*e**(-
beta*i))*e**(-beta*i)))*((1-explo2)*beta*e**(-
beta*i)*(F4)+beta*e**(-beta*i)*(F3)+exploe2*(-beta*i)*((T[66][j])**4))
                C1=(explo1/beta)*((T[0][j])**4)-(1-explo1)*C2
                H=beta*beta*kesei*e**(-beta*i) (F1+C2)-
beta*beta*kesei*e**(-beta*i) [F2+C1]
                T[i][j]=T[i][j+1]-(k2/C2A)*(T[i+1][j-1]-
2T[i][j+1]+T[i+1][j+1))+H
            elif (i/10) in range(0.6,6.6):
                T[7][j]=(k1/K2)*(T[6][j+1]-T[5][j+1])-T[6][j+1]
                T[i][j]=T[i][j+1]-(k2/C2A)*(T[i+1][j-1]-
2T[i][j+1]+T[i+1][j+1]))
            elif (i/10) in range(6.6,10.2):

```

```

T[67][j]=(k2/K3)*(T[66][j+1]-T[65][j+1])-
T[66][j+1]
T[i][j]=T[i][j+1]-(k3/C3A)*(T[i+1][j-1]-
2T[i][j+1]+T[i+1][j+1])
elif (i/10) in range(10.2, 15.2):
    M((((kesei)*T[102][j]**4)-(T[102][j]**4))*e**(-
k*i)-(((kesei)*T[102][j]**4)-(T[102][j]**4))*e**(-
k*(i+1))))/((1/explo3)+(1/explo4)-1)
    T[i][j]=T[i][j+1]-(k2/C2A)*(T[i+1][j-1]-
2T[i][j+1]+T[i+1][j+1])+M
else:

T[153][j+1]T[152][j+1]+(1/k5)*((((kesei)*T[102][j]**4)-
(T[102][j]**4))*e**(-k*3.6*(10**(-9))))-(k4)*((T[153][j+1]-
T[152][j+1]))
T[i][j]=T[i][j+1]-(k5/C5A)*(T[i+1][j-1]-
2T[i][j+1]+T[i+1][j+1])

Tlast = copy.deepcopy(T)

return T[152][:]

```

附录2 PSO modular based on Python

```

# -*- coding: utf-8 -*-
from time import time
from sopt.util.functions import *
from sopt.util.pso_config import *
from sopt.PSO.PSO import PSO
from sopt.util.constraints import *
from application.app.folder.file import pde.py
class TestPSO:
    def fun, c1 (T[152;:]):
        return (T[152;:]-T0[欽 ]^2)
    def __init__(self):
        self.func = func1
        self.func_type = 欽樅in欽?
        self.variables_num = ?
        self.lower_bound = quadratic11_lower_bound
        self.upper_bound = quadratic11_upper_bound
        self.c1 = basic_config.c1
        self.c2 = basic_config.c2
        self.generations = 200

```

```

        self.population_size = 100
        self.vmax = 1
        self.vmin = -1
        self.w = 1
        self.w_start = 0.9
        self.w_end = 0.4
        self.w_method = pso_w_method.linear_decrease
        #self.complex_constraints = [constraints1, constraints2, constraints3]
        self.complex_constraints = None
        self.complex_constraints_method = complex_constraints_method.loop
        self.PSO = PSO(**self.__dict__)

    def test(self):
        start_time = time()
        self.PSO.run()
        self.PSO.save_plot()
        self.PSO.show_result()

if __name__ == '__main__':
    TestPSO().test()

```

附录三 第二问：

```

# -*- coding: utf-8 -*-
from time import time
# -*- coding: utf-8 -*-
from sopt.util.functions import *
from sopt.util.pso_config import *
from sopt.PSO.PSO import PSO
from sopt.util.constraints import *
from application.app.folder.file import pde.py
class TestPSO:
    def func2(l):
        return (T[152,3600]-47>1exp**-7)&&(44-
T[152,3300]>1exp**-7)
    def __init__(self):
        self.func = func2
        self.func_type = 'min'
        self.variables_num = 1
        self.lower_bound = 0.6
        self.upper_bound = 25
        self.c1 = basic_config.c1
        self.c2 = basic_config.c2

```

```

        self.generations = 200
        self.population_size = 100
        self.vmax = 1
        self.vmin = -1
        self.w = 1
        self.w_start = 0.9
        self.w_end = 0.4
        self.w_method = pso_w_method.linear_decrease
        #self.complex_constraints =
[constraints1,constraints2,constraints3]
        self.complex_constraints = None
        self.complex_constraints_method =
complex_constraints_method.loop
        self.PSO = PSO(**self.__dict__)

    def test(self):
        start_time = time()
        self.PSO.run()
        self.PSO.save_plot()
        self.PSO.show_result()

if __name__ == '__main__':
    TestPSO().test()

```

附录四 第三问：

```

# -*- coding: utf-8 -*-
from sopt.util.functions import *
from sopt.util.pso_config import *
from sopt.PSO.PSO import PSO
from sopt.util.constraints import *
from application.app.folder.file import pde.py
class TestPSO:
    def func3(l2,l4):
        return (T[152,1800]-47>1exp**-7)&&(44-T[152,1500]>1exp**-7)
    def __init__(self):
        self.func = func3
        self.func_type = 欽櫟in欽?
        self.variables_num = 2
        self.lower_bound = 0.6,0.6
        self.upper_bound = 25,6.4
        self.c1 = basic_config.c1
        self.c2 = basic_config.c2

```

```

self.generations = 400
self.population_size = 100
self.vmax = 1
self.vmin = -1
self.w = 1
self.w_start = 0.9
self.w_end = 0.4
self.w_method = pso_w_method.linear_decrease
#self.complex_constraints = [constraints1, constraints2, constraints3]
self.complex_constraints = None
self.complex_constraints_method = complex_constraints_method.loop
self.PSO = PSO(**self.__dict__)

def test(self):
    start_time = time()
    self.PSO.run()
    self.PSO.save_plot()
    self.PSO.show_result()

if __name__ == '__main__':
    TestPSO().test()

```