

复杂物体表面最优喷涂模型的研究

摘要

科学技术的发展极大地推动了工业生产的自动化进程，其中喷涂机器人发挥了非常重要的作用。但是目前仍然存在喷漆不均匀，喷涂效率低下等难题。本文以座椅为例，建立了椭圆双 β 分布模型、路线规划评价模型、基于蚁群算法的片间路径优化模型和连续变倾角动态喷涂模型，运用黄金分割法、K_means聚类分析、蚁群算法、粒子群算法等对模型进行求解，以解决喷涂复杂物体表面的优化问题。

对于问题一，本文首先建立**曲面拟合一般方程**。使用1stOpt工具处理座椅的点云数据，经拟合，求得描述坐垫曲面的有理表达式 z ，拟合的均方根误差为0.0889，相关系数为0.9983。因此，该有理表达式拟合曲面效果较好。

对于问题二，本文根据平面和曲面特性分别建立喷涂模型：

针对平面喷涂策略，建立**椭圆双 β 分布模型**和**路线规划评价模型**。根据椭圆双 β 分布模型的喷涂厚度积累速率，使用黄金分割法计算实际喷涂厚度与理想厚度的偏差平方和，以评估平面是否喷涂均匀；根据喷涂方向改变次数、喷涂路径转折次数和路径最小分段长度三个影响路径规划的因素构建路径规划评价函数 G 。通过对平面图形仿真发现，当喷涂轨迹间距离 $w=131.5\text{ mm}$ 时，偏差平方和为0.015，且使用方案1的 G 较大，此时平面喷涂效果好；

针对曲面喷涂策略，分为三个步骤：**基于K_means聚类算法的分片处理**，子片内路径规划评价和**基于蚁群算法的片间路径优化**。首先使用K_means算法将曲率相似并且距离相近的离散点划分为同类，将曲面问题转化为平面问题，获得多个子片；在子片内按照平面路径规划评价模型进行优化，获取片内最优路径；最后将子片间的路径规划转化为旅行商问题，使用蚁群算法优化片间路径。仿真发现，通过对座椅坐垫曲面分片获得10个子片，且使用蚁群算法好地实现片间路径规划，曲面喷涂效果好。

对于问题三，本文建立**连续变倾角动态喷涂模型**。将喷涂全过程分为垂直喷涂、变倾角喷涂和恒倾角喷涂三个阶段，根据每个阶段的特点建立表面喷涂厚度函数，并使用粒子群算法求得理想厚度与犄角实际喷涂厚度的偏差平方和。为求解优化喷涂高度，将变倾角和恒倾角分别分为六个子阶段，求解搜索优化后的喷涂高度。本文以 90° 的外犄角为例仿真，可以看到每个 H 均对应一个优化喷涂高度，且偏差平方和为29.4667，犄角喷涂效果良好。

综上所述，本文结合1stOpt，MATLAB等软件，从不同情况对多类复杂物体的喷涂问题进行分析，合理地解决了工业生产中对复杂物体的喷涂要求，在机械制造、航空航天、家具生产等领域具有极大的应用价值。

关键词：机械臂喷涂，路径优化，K_means聚类算法，蚁群算法，变倾角动态喷涂

一 问题重述

1.1 问题背景

随着科技浪潮的推进，工业机器人在各个领域发挥着举足轻重的作用，它们大大提高了生产效率和自动化程度，代替人类完成众多高强度高风险的工作。喷涂机器人是工业机器人之一，它们可以利用装有视觉识别系统的机械臂，对各种复杂的物体进行自动喷涂作业，同时避免了人类直接接触化学涂料中的有毒物质，使得产品获得良好的外部涂层，增加了产品的价值，在航空航天、汽车制造、家具生产等领域有普遍应用。

喷涂机械臂作业时，其自带的视觉系统需要识别复杂物体，根据识别到的物体表面寻找提前规划好的喷涂路径，进行自动涂层。在喷涂的过程中，由于物体空间结构的多样性和涂料的差异性，可能会出现喷涂不均匀、作业时间长等问题，导致机械臂效率低下，使得工业生产上对喷涂机械臂的要求越来越高。因此，解决对复杂物体的准确识别和构建最优的喷涂路线规划问题具有十分重要的意义。

1.2 问题重述

根据以上背景，本文将建立复杂表面喷涂作业数学模型，解决以下问题：

问题一：不考虑视觉的识别过程，建立描述复杂物体的数学模型；

问题二：根据物体的描述模型，建立喷涂规则形状以及不规则形状物体的数学模型；

问题三：提出复杂形状——犄角的喷涂策略。

二 问题分析

2.1 对于问题一的分析

在通常情况下，工业生产的物体是多形状多材料的复杂体，直接对整个物体进行分析比较困难。本文知道，每一个立体物体都是由面组成的，因此要想建立描述复杂物体的数学模型，只需要找到其曲面数学表达即可。

基于以上分析，对于复杂物体描述的数学模型，本文拟选取一个复杂物体——座椅为例，将立体物体分解成面，根据.obj文件提取出其点云数据，使用1stOpt工具对离散点曲面图进行拟合，将多次拟合所获取的最优解作为复杂物体曲面的最终表达式。

2.2 对于问题二的分析

在工业生产中，客户对产品各个维度的要求越来越高，这意味着对喷涂机器人的要求也越来越高。实现高质量的喷涂是目前亟待解决的难题。如何实现复杂物体表面喷涂均匀，减少喷涂所需要的时间、机械臂的姿态调整次数和机械臂所走的路径长度，是每个工厂需要认真考虑和解决的问题。

基于以上分析，对于规则物体和不规则物体的喷涂问题，本文将采取不同的涂抹方法来提高机械臂的工作效率。

为了提高效率，减少喷涂时间，本文将根据目标喷涂物体分两种喷涂方案：规则物体的操作较为简单，由于规则物体主要由平面组成，因此首先拟建立平面喷涂模型，根据模型本身的特点，利用不完全重叠实现均匀喷涂，同时建立路径评价函数对喷涂路径进行优化；而不规则的物体喷涂，需要对其先进行分类，采用K_means算法进行聚类分析，将曲率相似并且距离相近的数据点分为一类，进行分片喷涂作业；为了使

喷涂更加均匀，需要根据子片区域的特征喷涂不同厚度，实现喷涂的不完全重叠；为了减少机械臂姿态调整次数，本文将在子片内进行路径规划，确定机械臂最优的路径；为了使得机械臂所走的路径尽可能少，本文将对子片间的喷涂进行顺序规划，转化为GTSP旅行商问题，采用蚁群算法获取最佳路径。

2.3 对于问题三的分析

工业生产中，复杂物体喷涂，尤其是犄角的喷涂问题是工业生产制造的难点之一，需要对有犄角的物体实现涂抹均匀较为困难，且极易浪费涂料，造成生产效率较低等问题，因此犄角的喷涂问题亟待解决。

基于以上分析，本文将以外犄角为例，使用连续变倾角喷涂模型，将犄角的喷涂全过程分为垂直喷涂、变倾角喷涂和恒倾角喷涂三个阶段。在机械臂达到犄角两个面片相交线之前，连续变倾角会使得喷涂更加均匀，且比垂直喷涂的效率高。采用问题二中验证喷涂是否均匀的方法，计算实际喷涂厚度与理想厚度的偏差平方和，并将后两个阶段进行分解，获取变化的优化喷涂高度，最后验证模型的优劣。

三 模型假设

- 1.假设机械臂一次喷涂的面积是一定的。
- 2.假设物体是悬挂在流水线且姿态不动的，不考虑外界因素对物体的干扰。
- 3.假设喷枪在喷涂过程中的速度是恒定的，不考虑外界因素对机械臂的干扰。
- 4.假设机械臂的高度和姿态可以随时进行调整。
- 5.假设喷枪形成的喷涂分布近似为圆形，喷涂到物体上的喷涂投影为椭圆形。

四 符号说明

为简化对问题的分析和对数字的处理，本文将以如下符号代表变量：

变量名称	含义
k_n	曲面上任意一点的法曲率
K	聚类算法中分配的区域个数
$d_{i,k}$	点之间的欧几里得距离
q	喷涂厚度积累速率
w	喷涂轨迹间距
v	喷涂速度
q_u	椭圆中心喷涂的最大厚度积累
G	路径评价函数
φ	喷涂张角
T_d	喷涂的理想厚度

五 模型的建立与求解

5.1对于问题一建立的模型

问题一需要建立一个描述复杂物体的数学模型。首先，本文选取在家具生产当中具有代表性的复杂物体——座椅作为研究对象，通过座椅的.obj文件获取复杂物体曲面的点云数据。使用1stOpt工具绘制曲面拟合图，将拟合的结果与离散点图进行比较分析，根据均方根误差、相关系数、确定系数等大小来判断拟合效果，最终选择最优解作为描述复杂物体的有理多项式。

具体的思路流程图如下：

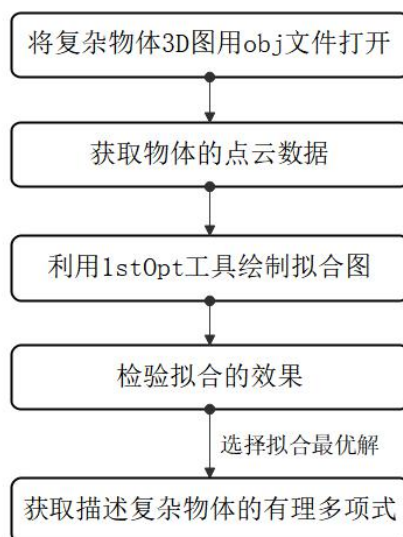


图1 问题一的思路流程图

5.1.1获取复杂物体的点云数据

如图2所示，该图是一个具有犄角、规则平面和不规则曲面的复杂物体，是复杂物体中的典型代表。

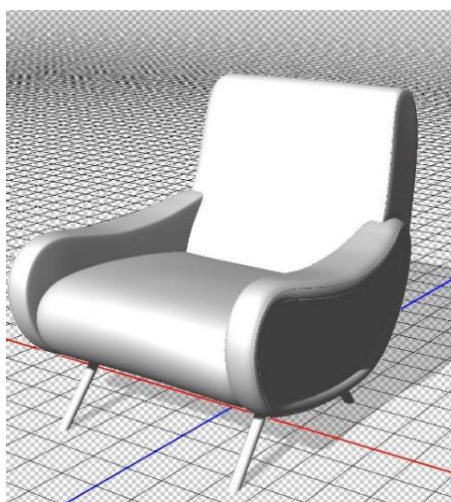


图2 复杂物体——座椅3D示意图

打开座椅的.obj文件，调用PCL中的TOOL，获取该座椅的点云数据，如图3所示。

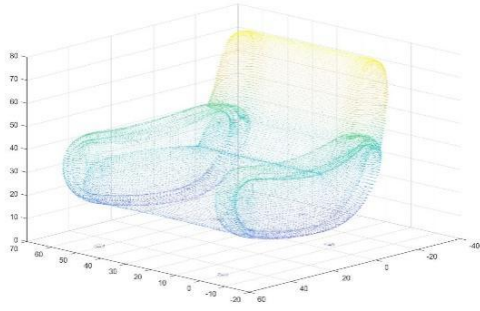


图3 座椅点云数据离散点图

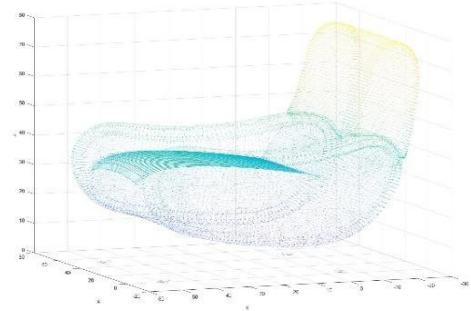


图4 座椅坐垫曲面拟合图

本文将以座椅坐垫曲面（即图4中浅蓝色区域）为代表，对该曲面进行拟合分析。

5.1.2 构建曲面的有理表达式

假设一般曲面方程为：

$$S(u,v) = \{x(u,v), y(u,v), z(u,v)\}$$

根据曲面理论^[1]可知曲面在任意一点处的法曲率是：

$$k_n = \frac{1}{\rho_n} = \frac{Ldu^2 + 2Mdudv + Ndv^2}{Edu^2 + 2Fdudv + Gdv^2}$$

分母为第一基本公式，分子为第二基本公式，E,F,G为第 I 类基本量，L,M,N为第 II 类基本量。高斯曲率（K）与平均曲率（H）的表达式为：

$$K = \frac{(LN - M^2)}{(EG - F^2)}$$

$$H = \frac{K_{\max} + K_{\min}}{2} = \frac{EN - 2FM + GL}{2(EG - F^2)}$$

将坐垫曲面的点云数据使用1stOpt工具进行处理，本文选用麦夸特法(Levenberg-Marquardt)和通用全局优化法，由离散点数据得到的拟合表达式为：

$$z = \frac{(P_1 + P_2 \cdot x + P_3 \cdot y + P_4 \cdot y^2 + P_5 \cdot y^3)}{1 + P_6 \cdot x + P_7 \cdot x^2 + P_8 \cdot y}$$

拟合结果与离散点之间的均方根误差（RMSE）为0.0889，相关系数（R）为0.9983，确定系数（R-Square）为0.9966，决定系数（DC）为0.9966，拟合表达式的其他数据均位于附录中。曲面表达式与离散点的对照图5所示，由此可见，相关系数R>0.8，相关性很强，拟合效果比较理想，则上式z可以较好地拟合出该曲面。

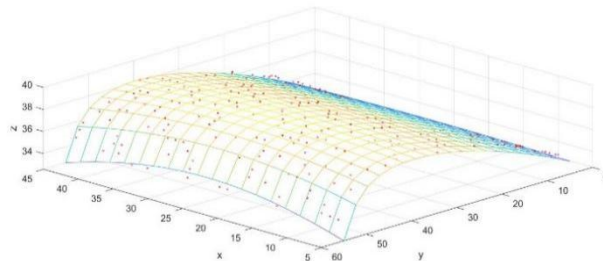


图5 座椅坐垫曲面表达式与离散点对照图

5.2 对于问题二建立的模型

5.2.1 规则物体的喷涂模型

1. 建立椭圆双 β 分布的平面喷涂模型

规则物体的面主要由平面组成。机械臂在实现喷涂时，喷枪形成的喷涂分布可以近似认为是圆形的，且喷涂区域中某点的喷涂厚度与该点到喷枪中心的距离有关^[2]。在喷涂的范围内，由于空气中涂料颗粒的相对浓度是一个 $(0,1)$ 的随机分布，满足 β 分布的条件，因此厚度积累可以近似认为满足 β 分布。在实际生产中，喷涂模型为圆锥体^[3]，并且具有一定的厚度，喷枪喷出的涂料在平面上，产生的喷涂投影为椭圆形，椭圆范围内的任意一点的喷涂厚度积累速率可以写为：

$$q(x,y) = q_u \left(1 - \frac{x^2}{a^2}\right)^{\beta_1-1} \left(1 - \frac{y^2}{b^2(1-x^2/a^2)}\right)^{\beta_2-1}$$

其中， $-a \leq x \leq a$ ， $-b(1-x^2/a^2)^{\frac{1}{2}} \leq y \leq b(1-x^2/a^2)^{\frac{1}{2}}$ ， a ， b 分别对应椭圆投影面的长轴长度和短轴长度， q_u 为椭圆中心点的喷涂厚度累积的最大值。上式由两个 β 分布函数组合而成，分别表示沿长轴和短轴方向上的不同厚度分布，因此构建椭圆双 β 分布模型^[4]，这是一种比较理想的喷涂模型，因此本文选用椭圆双 β 分布模型进行分析。

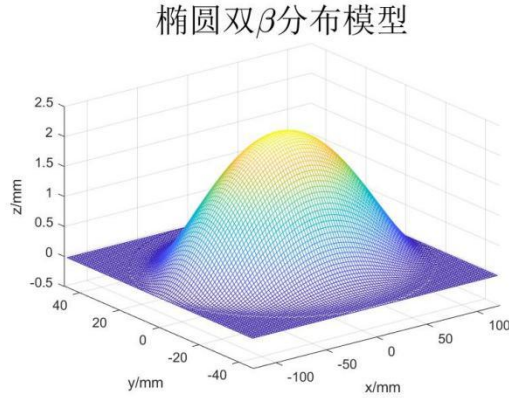


图6 喷枪椭圆双 β 分布模型

喷枪沿喷涂轨迹以恒定的速度进行喷涂，可以将 x 轴坐标为 x_s 的点的喷涂厚度通过对这条线上的点的厚度叠加得到^[5]。以长轴方向作为 x 轴，喷枪运动方向作为 y 轴，如图：

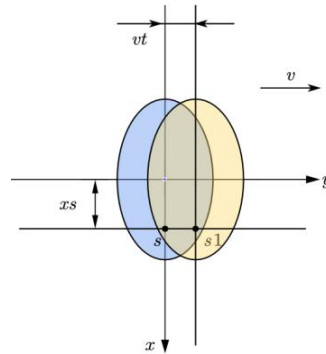


图7 沿直线喷涂示意图

假设喷涂区域中各点的涂层厚度累积正比于时间 t ，喷枪不动，点 s 以恒定速度 v 沿直线通过椭圆喷涂区域用的时间可以表示为 $t = 2b(1 - x_s^2/a^2)^{1/2}/v$ ，将 $x = x_s$ ， $y = y_s = b(1 - x_s^2/a^2)^{1/2} - vt$ 代入 $q(x, y)$ ，可以得到点 s 上的涂层厚度：

$$q_s(x_s) = \int_0^t q_u \left(1 - \frac{x_s^2}{a^2}\right)^{\beta_1-1} \left(1 - \frac{\left(b\left(1 - \frac{x_s^2}{a^2}\right)^{1/2} - vt\right)^2}{b^2\left(1 - \frac{x_s^2}{a^2}\right)}\right)^{\beta_2-1} dt$$

$$= q_u \left(1 - \frac{x_s^2}{a^2}\right)^{\beta_1-1} \frac{1}{v} \int_0^{t'} \left(1 - \frac{\left(b - \left(1 - \frac{x_s^2}{a^2}\right)^{1/2} - t'\right)^2}{b^2\left(1 - \frac{x_s^2}{a^2}\right)}\right)^{\beta_2-1} dt'$$

其中：

$-a \leq x \leq a$ ， $-b(1 - x^2/a^2)^{1/2} \leq y \leq b(1 - x^2/a^2)^{1/2}$ ， $t' = vt = 2b(1 - x_s^2/a^2)^{1/2}$ ，取 $v=100\text{mm/s}$ ，得到直线轨迹上一点的积累厚度模型，如下图所示：

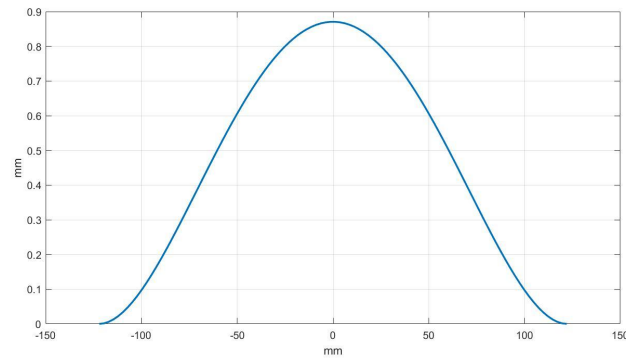


图8 匀速直线喷涂时厚度积累模型

2. 解决平面喷涂问题

1) 喷涂厚度均匀问题

由椭圆双 β 分布模型分析，喷枪沿椭圆短轴方向进行喷涂时可以得到较少的轨迹条数。因此，假设喷枪沿短轴行进，当对平面以直线路径进行喷涂时，需要给出相邻喷涂轨迹之间的喷涂间距与喷涂速度，示意图如下所示：

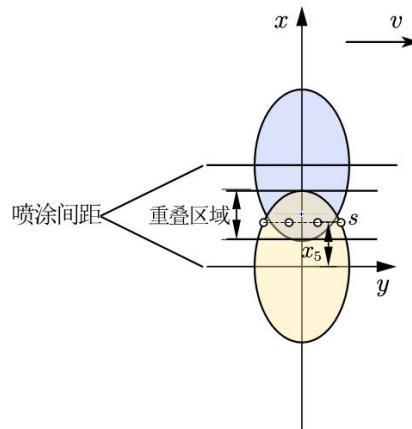


图9 平面的直线路径规划示意图

在两个相邻路径之间的喷涂厚度表达式为：

$$q(x, w, v) = \begin{cases} q_1(x, v) & 0 \leq x \leq w - a \\ q_1(x, v) + q_2(x, v, w)w & -a \leq x \leq a \\ q_2(x, v, w) & a \leq x \leq w \end{cases}$$

重叠区域中任一点的喷涂厚度可以由该点处垂直于长轴的两条喷涂轨迹厚度叠加得到，对于x轴坐标的 x_s 的点，可以再叠加区域处取离散点 s ，则距离喷枪中心为 x_s 处的喷涂厚度可以看作这些离散点处的厚度之和，因此有：

$$q_1(x, v) = \sum_{i=1}^{n+1} q_u \left(1 - \frac{x_s^2}{a^2}\right)^{\beta_1-1} \left(1 - \frac{\left(b\left(1 - \frac{x_s^2}{a^2}\right)^{1/2} - vt_{1i}\right)^2}{b^2\left(1 - \frac{x_s^2}{a^2}\right)}\right)^{\beta_2-1} \cdot \frac{T_1}{n}$$

式中 $T_1 = 2b(1 - x_s^2/a^2)^{1/2}/v$, $t_{1i} = \frac{T_1}{n}(i-1)$, $i = 1, 2, \dots, n+1$ 。

$$q_2(x, v, w) = \sum_{i=1}^{n+1} q_u \left(1 - \frac{(x_s - w)^2}{a^2}\right)^{\beta_1-1} \left(1 - \frac{\left(b\left(1 - \frac{(x_s - w)^2}{a^2}\right)^{1/2} - vt_{2i}\right)^2}{b^2\left(1 - \frac{(x_s - w)^2}{a^2}\right)}\right)^{\beta_2-1} \cdot \frac{T_2}{n}$$

式中 $T_2 = 2b(1 - (x_s - w)^2/a^2)^{1/2}/v$, $t_{2i} = \frac{T_2}{n}(i-1)$, $i = 1, 2, \dots, n+1$ 。

判断喷涂轨迹是否合理的一个重要指标是轨迹之间的喷涂厚度是否均匀，因此可以将实际喷涂厚度与理想厚度的偏差平方之和作为目标函数，用最小二乘法求出喷涂轨迹间距 w 以及喷涂速度 v ，目标函数可以用下式表达：

$$\min E(v, w) = \int_0^w (q(x, v, w) - q_d)^2 dx$$

式中 q_d 为理想涂层厚度，取 $q_d=0.9\text{mm}$ ，可得：

$$q_s(x_s) = \frac{1}{v} q_u \left(1 - \frac{x_s^2}{a^2}\right)^{\beta_1-1} \int_0^{t'} \left(1 - \frac{\left(b\left(1 - \frac{x_s^2}{a^2}\right)^{1/2} - t'\right)^2}{b^2\left(1 - \frac{x_s^2}{a^2}\right)}\right)^{\beta_2-1} dt'$$

其中 $t' = vt = 2b(1 - x_s^2/a^2)^{1/2}$ 。

设：

$$\rho(x, w) = q_u \left(1 - \frac{x_s^2}{a^2}\right)^{\beta_1-1} \int_0^{t'} \left(1 - \frac{\left(b\left(1 - \frac{x_s^2}{a^2}\right)^{1/2} - t'\right)^2}{b^2\left(1 - \frac{x_s^2}{a^2}\right)}\right)^{\beta_2-1} dt'$$

则有 $q(x, v, w) = \frac{1}{v} \rho(x, w)$ ，从而：

$$\frac{\partial E(v, w)}{\partial v} = 0$$

得到的速度 v 与间距 w 的关系：

$$v(w) = \frac{\int_0^w \rho^2(x, 2) dx}{q_d \int_0^w \rho(x, w) dx}$$

可以得到厚度偏差之和与喷枪的喷涂间距 w 之间的关系，并且只有一个自变量，因此 $E(w)$ 可以改写为：

$$E(w) = \int_0^w \left(\frac{1}{v} \rho(x, w) - q_d \right)^2 dx = \sum_{j=1}^{k+1} \left(\frac{1}{v} (x_j, w) - q_d \right)^2 \Delta x$$

其中 $j=1, 2, \dots, k+1$; $\Delta x = \frac{w}{k}$, $x_j = \Delta x(j-1)$ 。

以上求解转化为求一维最优解的问题，偏差极小值可以通过黄金分割法进行计算，喷涂间距由此得出。步骤如下：

STEP1: 设置初始间距范围，最小为 $\delta_1 = R$ ，最大为 $\delta_2 = 2R$ ，间距偏差阈值为 ε ；

STEP2: 计算间距为 δ_1, δ_2 时对应的喷涂速度 v_1, v_2 ，以及对应的偏差 E_1, E_2 ；

STEP3: 计算黄金分割点，其中 $\delta_3 = \delta_2 - 0.618(\delta_2 - \delta_1)$, $\delta_4 = \delta_1 + 0.618(\delta_2 - \delta_1)$ ；

STEP4: 计算间距为 δ_3, δ_4 时对应的喷涂速度 v_3, v_4 ，以及对应的偏差 E_3, E_4 ；

STEP5: 如果 $E_3 \geq E_4$ ，令 $\delta_1 = \delta_3$ ， $E_1 = E_3$ ；否则，令 $\delta_2 = \delta_4$ ， $E_2 = E_4$ ；

STEP6: 如果 $|\delta_1 - \delta_2| \leq \varepsilon$ ，算法停止， $\delta = (\delta_1 + \delta_2)/2$ ；否则，转到STEP3。

2) 路径规划优化问题

做好路径规划可以减少喷涂时间，使得机械臂的调整姿势次数和所走路径尽可能少。因此，基于以上考虑，本文拟建立一个路径评价体系模型来评价是否为最优路线。

通常情况下，喷涂路径的影响因素如下：

- 喷涂方向改变次数：在实际生产过程中，应尽量保证喷涂方向改变次数尽量少，这样有利于对喷漆机器人的运动控制，也可以保证良好的涂层均匀度。设 d_c 为喷涂方向改变次数， d_c 的数值可以通过观察喷涂路径获得。

- 喷涂路径转折点数：该因素近似于喷涂方向改变次数，若一个喷涂路径中转折点数过多则会影响喷涂效果。但是喷涂路径转折点增加不一定会导致喷涂方向改变次数的增加，故设喷涂路径转折点数 p_s ， p_s 的数值可以通过观察喷涂路径获得。

- 路径最小分段长度：在喷漆路径优化过程当中，复杂待喷表面由于曲率不同需要对每一段的喷涂速率进行优化，减少因停顿换向等影响因素，故最小分段长度越大，喷漆效果越好，喷涂所需要的时间越少，机械臂姿态调整次数越少。故设 l_m 为路径最小分段长度。

综上所述，影响路径规划的因素主要由三个，喷涂方向改变次数和喷涂路径转折点数越少，路径最小分段长度越长，选择的路径规划越好，越满足要求。基于上述三个因素建立的平面路径评价函数表达式为：

$$G = \omega_1 \left(\frac{1}{d_c} \right) + \omega_2 \left(\frac{1}{p_s} \right) + \omega_3 l_m$$

其中： $\omega_i (i=1, 2, 3)$ 为针对不同实验需求下对应不同参数的权值。显而易见，函数值 G 大的路径，喷漆结果好。

5.2.2 不规则物体的喷涂模型

1. 基于K_means聚类算法的曲面分片处理

不规则物体喷涂的主要难点是由于表面不光滑，在不同点位置的曲率存在明显差异。因此，需要首先对面进行分片处理，将曲率相似并且距离相近的离散点划分为同类。

基于以上的考虑，本文首先采用K_means聚类算法。K_means聚类算法可以将 n 个

数据点分配到K个区域当中，且K的值很大程度上决定了区域划分的效果。而K的值将通过枚举的方式，根据聚类效果决定K值。K_means聚类算法流程如下所示：

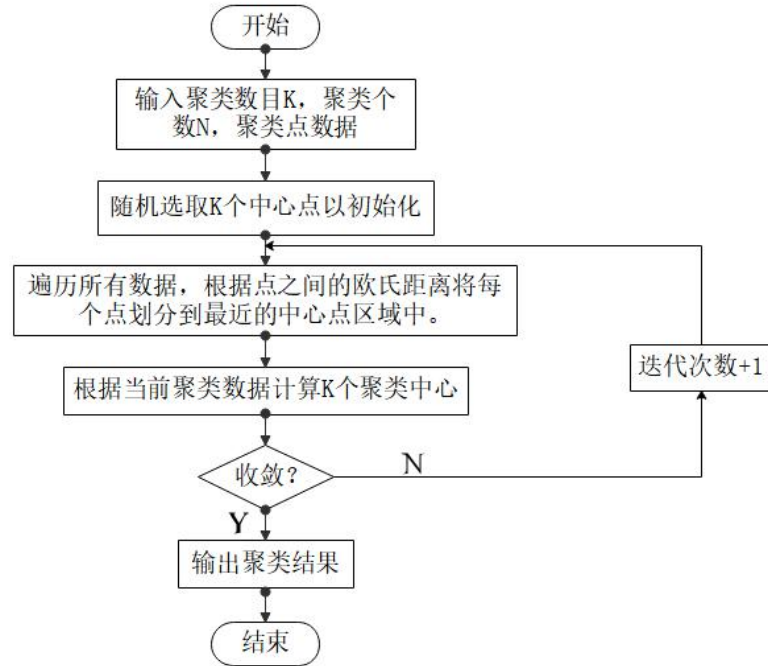


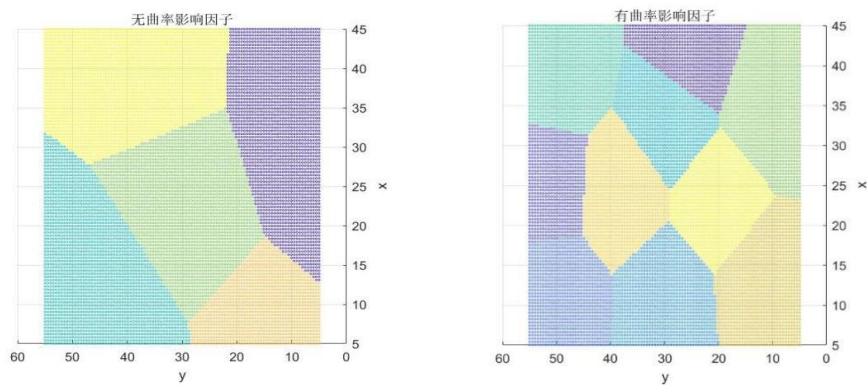
图10 K_means算法流程图

在该算法中加入权值能够改变聚类效果，因此引入曲率影响因子加以改造，曲率越大，曲率影响因子越大。在聚类中心计算时，将曲率影响因子与点之间的欧几里得距离相乘，曲率较大处的点与中心点之间的距离数值变大，使得曲率大小相差较多的点不容易被分到同一类中。

$$Q_i = 1 + A \cdot |K_i + H_i|$$

式中， $A \cdot |K_i + H_i|$ 表示曲率带来的影响，成为曲率影响因子，A为常数，可通过调节A的值来改变曲率的影响。“1”表示点之间的欧氏距离带来的影响，成为距离影响因子，此处采用默认值1，在计算聚类中心时用 $Q_i \cdot d_{i,k}$ 替换原来的欧氏距离 $d_{i,k}$ 进行计算，则能够将曲率与距离相近的点划分到同一区域。

将问题一中获取的座椅坐垫曲面点云数据进行聚类分析，结果如下：



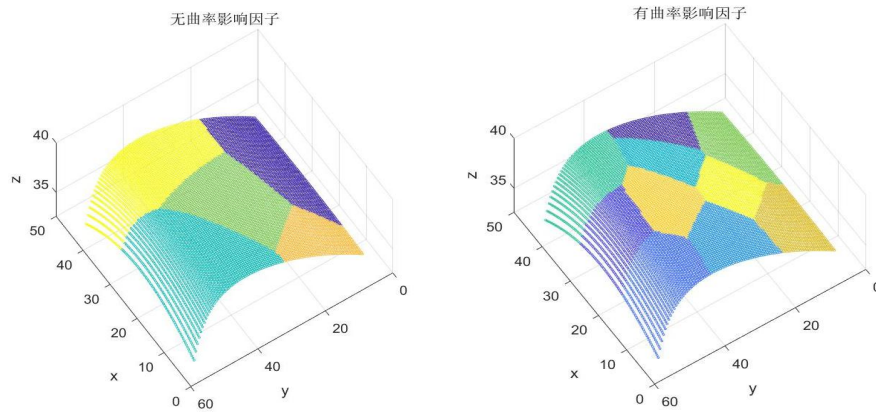


图11 聚类分析结果图

用MATLAB 编写的K_means聚类算法实现曲面的分片，经试验当 $K=10$ 的效果较好。根据上图可以发现，无曲率影响因子的聚类分析，分片只是将曲面以点之间的欧几里得距离作为唯一标准划分为 K 个部分，且根据这个单一的指标仅可以将曲面分解为五个部分。加入曲率影响因子之后，曲面可以根据曲率大小和点之间的欧几里得距离两个指标分为十个曲率变化不明显的部分。

以上曲面的子片可以确定十中心点，将中心点之间的连线取中垂线可以快速确定子片边界，生成最后的分片。经过分片步骤后，每个子片上的曲率变化都较小，因此在每个子片上都能够得到较为理想的喷涂效果。

2. 解决曲面子片内的喷涂问题

每个子片上的曲率变化较小，因此可以将每个子片近似看作一个平面，因此曲面的片内喷涂最优方案主要基于平面喷涂最优方案进行改进。

1) 喷涂厚度均匀问题

由本文5.2.1可知，为解决子片内的喷涂均匀问题，同样采用平面的不完全重叠优化路径。但是，平面和曲面的不同在于曲面与水平面存在一个夹角，如下图所示：

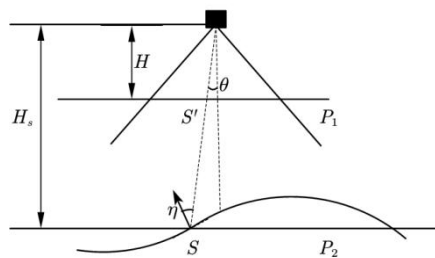


图12 曲面厚度分布

如图表示喷枪对曲面进行喷涂，曲面上任意一点 S ， η 表示点 S 的法向量和点 S 与喷枪之间的夹角， θ 为点 S 与喷枪的连线与喷枪中轴线之间的夹角，作为一个参考平面 P_1 ，平面 P_1 到喷枪的高度为 H ，点 S 与喷枪连线在 P_1 平面上的交点为 S' ，过点 S 做参考平面 P_2 ，平面 P_2 到喷枪的高度为 H_s 。

根据微分几何放大定理可知，若喷枪在同一时间内喷到平面 P_1 和平面 P_2 的涂料总量相同，则平面的面积越大，面上的涂层厚度越小，因此可以得到曲面上任意一点 S 的涂层积累速率模型为：

$$q_s = q_1 \left(\frac{H}{H_s} \right)^2 \frac{\cos \eta}{\cos \theta}$$

因此，要求曲面子片内的喷涂速率，只需要求得子片对应平面的喷涂路径规划，再根据上式求得即可。

2) 路径规划优化问题

曲面子片内路径规划的方法与平面路径规划的评价体系相同，但是不同之处在于需要把子片投影到XOY，XOZ，YOZ中投影面积较大的面上，再对投影面进行路径规划并给出路径评价体系模型的评估结果。

3. 基于蚁群算法的片间路径最优规划

当子片内路径规划结束后，应对子片间进行最优路径规划，即确定子片喷涂的先后顺序，使得子片间的路径最短。本文将该问题转换为GTSP问题，并采用蚁群算法解决，其思路流程图如下所示：

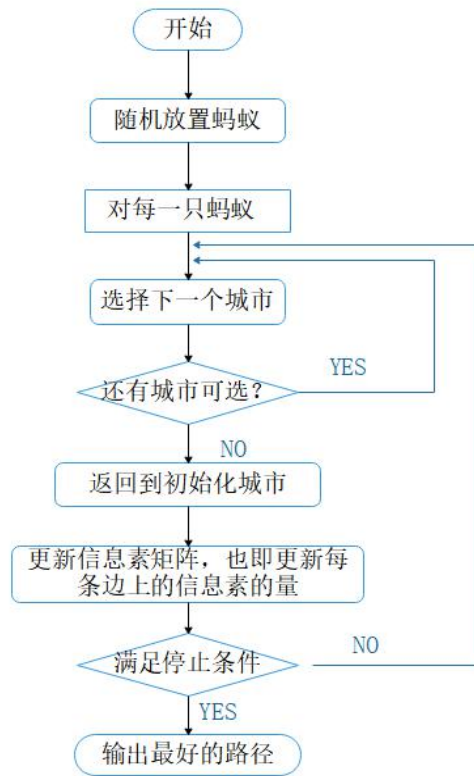


图13 蚁群算法流程图

根据以上的流程，算法中位于城市*i*的蚂蚁*k*，根据伪随机比例规则选择城市*j*作为下一个访问的城市，路径的选择公式为：

$$j = \begin{cases} \arg \max_{l \in a_k} \{ \tau_{il} [\eta_{il}]^\beta \} & q \leq q_0 \\ J & otherwise \end{cases}$$

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{s \in a_k} [\tau_{is}(t)]^\alpha [\eta_{is}(t)]^\beta} & j \in a_k \\ 0 & otherwise \end{cases}$$

式中，*q*为[0,1]中的一个随机变量，*q*₀为[0,1]中的一个参数，*J*为根据以上公式给出的概率分布产生的随机变量， τ_{ij} 为边(*i*,*j*)上的信息素， $\eta_{ij} = 1/d_{ij}$ 为城市*i*到城市*j*的启发

式因子， a_k 为蚂蚁k下一步被允许访问的城市的合集，如果 $q \leq q_0$ ，则按信息素浓度选择路径，否则按照 $p_{ij}^k(t)$ 进行概率搜索，全局信息素更新规则为：

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij}^{bs} \quad \forall (i,j) \in T^{bs}$$

$$\Delta\tau_{ij}^{bs} = \frac{1}{C^{bs}}$$

局部信息素更新规则为：蚂蚁每经过一条边 (i,j) ，都将立刻更新该边的信息素：

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \xi\tau_0$$

ξ 与 τ_0 为两个参数，其中 $0 < \xi < 1$ ， τ_0 为信息素量的初始值，局部更新使得蚂蚁每经过一边，该边的信息素 τ_{ij} 都会减少，使其它蚂蚁选中该边的概率减小，使得搜索能力变强。

根据10个子片的位置，每个子片能够确定喷涂开始与结束的两个点，即总共能够确定20个点，根据蚁群系统与GTSP的概念，GTSP问题中的城市即为子片中的20个点，蚂蚁则为喷枪，求解得出的路径顺序即为子片的喷涂顺序。所有点中位于同一子片上的两点在S中必须相邻，以此作为S向量的约束条件。约束公式为：

$$T(S,L) = T_1(L) + T_2(S,L)$$

$$J = \arg \min_s T(S,L)$$

$T_1(L)$ 为子片的喷涂路径长度，因为子片喷涂路径能够，所以为固定值； $T_2(S,L)$ 为子片间的移动路径长度，是蚁群算法优化的目标。

5.3 对于问题三建立的模型

5.3.1 连续变倾角喷涂轨迹生成

由于犄角属于较为复杂的物体，因此不考虑空气中颗粒等杂物的影响，将理想的垂直喷涂到物体上的喷涂分布近似为圆形。基于以上考虑，本文将问题2中建立的椭圆双 β 分布模型简化为抛物线涂层厚度模型，表示如下：

$$f(r) = \begin{cases} A(R^2 - r^2) & |r| \leq R \\ 0 & |r| > R \end{cases}$$

其中 R 是喷雾半径， A 是常数， r 是从平面上的一点 S 到喷枪中心沿喷射方向的投影点的距离。

基于以上分析，由文献^[6]可知：喷涂倾角 α 作为变量的倾角涂层生长速率的理论模型可以表示为：

$$f(x,y,\alpha) = A(R^2 - r_s^2) \left(\frac{H}{H_s} \right)^2 \cos \alpha$$

$$r_s = \frac{H \sin \beta \sqrt{x^2 + y^2}}{H + x \sin \alpha}$$

$$\sin \beta = \sqrt{\frac{x^2 \cos^2 \alpha + y^2}{x^2 + y^2}}$$

$$H_s = H + x \sin \alpha$$

当喷枪的轴线沿着逆时针方向倾斜时， α 是正值；当喷枪的轴线沿着顺时针方向倾斜时， α 是负值。当倾角 $\alpha \neq 0$ 时，由喷涂形成的涂层覆盖区域是椭圆形，点O将椭

圆的长轴分为 α_1 和 α_2 两个部分时，它们的长度可以表示如下：

$$\alpha_1(H, \alpha) = \frac{H \tan^2(\varphi/2) \sin \alpha - H \tan(\varphi/2) \cos \alpha}{\tan(\varphi/2) \sin(2\alpha) - 1 - \tan^2(\varphi/2) \sin^2 \alpha + \sin^2 \alpha}$$

$$\alpha_2(H, \alpha) = \frac{H \tan^2(\varphi/2) \sin \alpha + H \tan(\varphi/2) \cos \alpha}{\tan(\varphi/2) \sin(2\alpha) + 1 + \tan^2(\varphi/2) \sin^2 \alpha - \sin^2 \alpha}$$

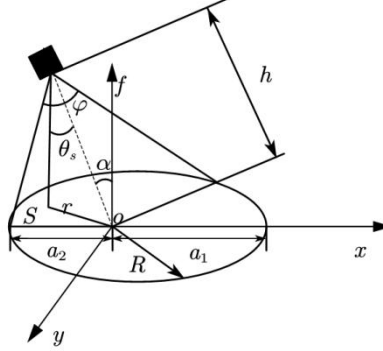


图14 倾角喷涂涂层分布

对于犄角喷涂问题，本文将采用连续变倾角喷涂轨迹来规划喷涂的方式：当喷枪靠近犄角处时则采用连续变倾角喷涂，喷枪由垂直喷涂状态以一定的角速度缓慢地倾斜，当喷枪距离两个面片相交线一定距离时，要求喷枪的倾斜轴线已经达到指定的倾角，此时喷枪倾斜轴线方向是两个面片之间角度的角平分线方向。相交线附近的喷涂轨迹区域可以分为三段：垂直喷涂区域，连续变倾角区域和恒定倾角喷涂区域，且两个面片的喷涂区域相对于相交线是对称分布的。如图所示为外犄角和内犄角喷涂的喷枪位姿直接过渡的情形：

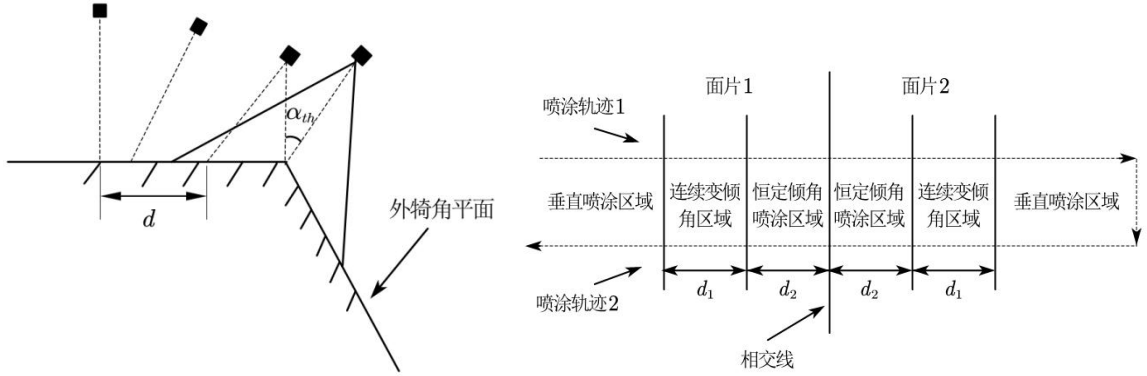


图15 连续变倾角喷涂喷枪位姿过渡示意图及喷涂区域分区

5.3.2 连续变倾角动态喷涂涂层厚度建模

为实现每个喷涂区域的均匀喷涂，需要优化轨迹的参数。这里需要优化连续变倾角喷涂轨迹段和恒定倾角喷雾轨迹端上的喷涂速度和相应的喷涂高度。在优化参数之前，需要建立每个喷涂区域的涂层厚度模型。

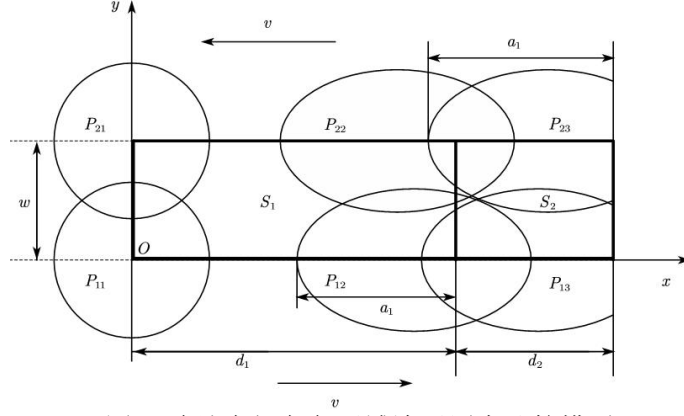


图16 连续变倾角各区域涂层厚度计算模型

如图所示， P_{11} (P_{21}), P_{12} (P_{22}), P_{13} (P_{23}) 分别代表喷涂轨迹段的垂直喷涂轨迹区域、连续变倾角轨迹区域和恒倾角喷涂轨迹区域。 P_{12} (P_{22}) 的长度为 d_1 ， P_{13} (P_{23}) 的长度为 d_2 ， S_1 所处的矩形区域为连续变倾角喷涂区域， S_1 所处的矩形区域为等倾角喷涂区域。

在 P_{11} (P_{21}) 区域为垂直喷涂轨迹段，喷涂速度为 v_0 ，当喷枪完成该段轨迹喷涂时，点 S_1 和 S_2 的涂层厚度可以表示为：

$$T_{P_{11}, P_{21}}(x, y) = \frac{1}{v_0} \int_{-H_0 \tan(\varphi/2)}^0 f(\alpha = 0, \gamma_{S_{1,2}}) dz$$

$$\gamma_{S_{1,2}} = \sqrt{(y + y_0)^2 + (z + x + H_0 \tan(\varphi/2))^2}$$

在 P_{12} (P_{22}) 处为连续变倾角喷涂轨迹段，采用离散的方法将该轨迹段分割为 i 段，假设每段中喷涂倾角，喷涂速度和高度都保持不变，当喷枪完成该段轨迹喷涂时，点 S_1 和 S_2 的涂层厚度可以表示为：

$$T_{P_{12}, P_{22}}(x, y, j) = \frac{1}{v_j} \int_{\frac{j-1}{i} d_1}^{\frac{j}{i} d_1} f\left(\frac{j}{i} \alpha_{th}, \gamma_{S_{1,2}}\right) dz$$

$$\gamma_{S_{1,2}} = \frac{H_j \sin \beta_j \sqrt{(y + y_0)^2 + (x - z)^2}}{H_j + (x - z) \sin\left(\frac{j}{i} \alpha_{th}\right)}, j \in [1, i]$$

其中， H_j 为连续变倾角轨迹第 j 段的喷涂高度， $(j/i) \alpha_{th}$ 为轨迹第 j 段相应的喷涂倾角。

P_{13} , P_{23} 段为等倾角喷涂轨迹段，喷枪倾角为 α_{th} ， H_{i+1} 保持恒定，采用离散的方法将该轨迹段分割为 n 段，假设在每个轨迹段中喷涂速度保持不变，当喷枪完成该段轨迹喷涂时，点 S_1 和 S_2 的涂层厚度可以表示为：

$$T_{P_{13}, P_{23}}(x, y) = \frac{1}{v_{i+m}} \int_{d_1 + \frac{m-1}{n} d_2}^{d_1 + \frac{m}{n} d_2} f(\alpha_{th}, \gamma_{S_{1,2}}) dz$$

$$\gamma_{S_{1,2}} = \frac{H_m \sin \beta_{th} \sqrt{(y + y_0)^2 + (x - z)^2}}{H_m + (x - z) \sin \alpha_{th}}, m \in [1, n]$$

喷涂区域中的点 S_1 和 S_2 的涂层厚度通过两个相邻轨迹喷涂完成之后涂层厚度叠加形成，根据 d_1 和 d_2 的不同值，涂层厚度模型如下所示：

$$T_{S_1}(x,y) = \sum_{j=1}^i T_{P_{12},P_{22}}(x,y,j) + \sum_{m=1}^n T_{P_{13},P_{23}}(x,y) + T_{P_{11},P_{21}}(x,y)$$

$$T_{S_2}(x,y) = \begin{cases} \sum_{j=1}^i T_{P_{12},P_{22}}(x,y,j) + \sum_{m=1}^n T_{P_{13},P_{23}}(x,y) + T_{P_{11},P_{21}}(x,y) & 0 < d_1 < H_0 \tan(\varphi/2) \\ \sum_{j=1}^i T_{P_{12},P_{22}}(x,y,j) + \sum_{m=1}^n T_{P_{13},P_{23}}(x,y) & d_1 > H_0 \tan(\varphi/2) \end{cases}$$

5.3.3 连续变倾角喷涂轨迹参数优化

根据建立的涂料厚度分布模型，以喷涂区域中任意一点的涂层厚度与理想涂层厚度方差最小建立优化目标函数，优化主要的喷涂工艺参数。目标方程为：

$$\begin{cases} E1(H_1, H_2, \dots, H_{i+1}) = \int_0^w \int_0^{d_1} [Ts_1(x,y) - T_d]^2 dx dy \\ E2(H_1, H_2, \dots, H_{i+1}) = \int_0^w \int_{d_1}^{d_1+d_2} [Ts_2(x,y) - T_d]^2 dx dy \\ \min E = E_1 + E_2 \\ H \in [H_{\min}, H_{\max}] \end{cases}$$

H_{\min} 和 H_{\max} 分别代表喷涂高度最小值和最大值， T_d 为理想的涂层厚度。

六 模型结果的分析 and 检验

6.1 对于椭圆双 β 分布模型的分析与检验

为检验椭圆双 β 分布模型和不完全重叠思路的合理性，使用MATLAB进行仿真。为符合实际生产过程中的喷涂需求，本文选取以下参数^[7]：

$q_u = 2.232mm$, $a = 121.99mm$, $b = 46.57mm$, $h = 200mm$, $\beta_1 = 2.47$, $\beta_2 = 4.72$, $q_d = 0.8mm$ ，优化前后两条相邻轨迹之间的喷涂厚度分布如下所示：

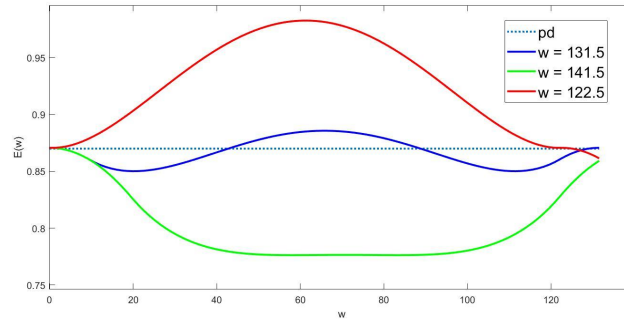


图17 平面模型优化前后路径间的涂层厚度

可以看到，当 $w = 131.5mm$ 时，涂层厚度与理想厚度之间的偏差较小，因此选择 $w = 131.5mm$ 作为最优的轨迹间距，此时偏差平方和为0.015。由图可以看到，优化前后的两条相邻喷涂轨迹之间的喷涂厚度分布波动较小，喷涂厚度更加接近理想厚度值。因此，该模型可以较好地反映喷涂厚度积累速率的变化，且根据黄金分割法可以喷涂理想厚度，并且获取该情况下的涂抹轨迹间距，解决平面喷涂均匀的问题。

6.2 对于路径规划评价模型的分析与检验

为检验路径规划评价函数模型的合理性，本文选择如下不规则的平面图形进行验证：

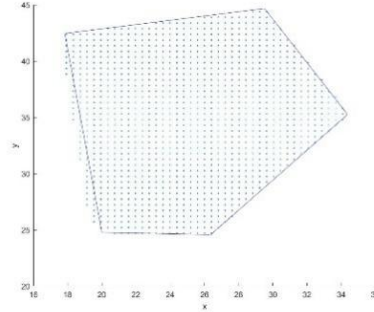


图18 平面图形

将路径规划分为两种方案，如下图所示：

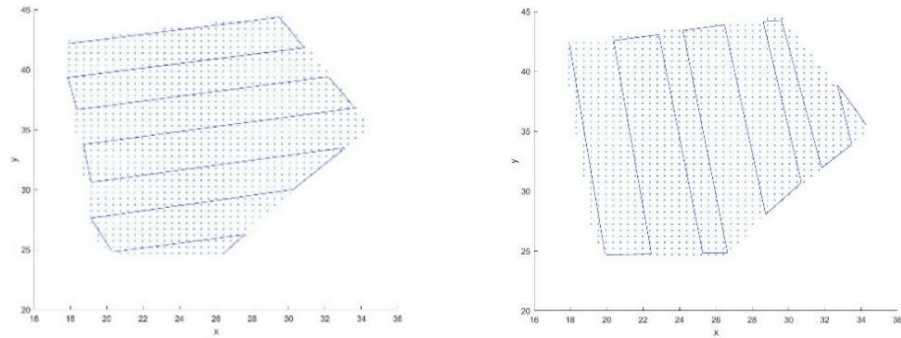


图19 平面图形的两种路径规划方案

根据以上的两种路径规划图，记录每个方案的数据。由于 d_c ， p_s ， l_m 三者的量纲无法统一，因此需要将其进行归一化。取 $\omega_1 = \omega_2 = \omega_3 = 1$ ，将获取的数据代入

$G = \omega_1 \left(\frac{1}{d_c} \right) + \omega_2 \left(\frac{1}{p_s} \right) + \omega_3 l_m$ 中，处理后获取如下信息：

表1 三种路径规划的归一化数据

	$1/d_c$	$1/p_s$	l_m	G
方案1	0.5357	0.5385	0.5461	1.6203
方案2	0.4623	0.4615	0.4539	1.3777

由表格分析可知，方案1的G要大于方案2，因此方案1的路径规划评价较高，本文选择方案1作为最优的路径规划。可以看到， $G = \omega_1 \left(\frac{1}{d_c} \right) + \omega_2 \left(\frac{1}{p_s} \right) + \omega_3 l_m$ 的路径评价模型可以较好地反映不同方案之间的三个变量的关系，并可以根据实际情况赋权重，且G越大，路径规划的效果越好，解决减少喷涂时间和机械臂的姿态调整次数的问题。

6.3 对于片间最优路径规划模型的分析与检验

为检验基于蚁群算法的片间最优路径规划模型，本文根据图11中的座椅坐垫曲面

图进行仿真，将根据子片内的路径规划寻找到的起始点和结束点，来确定的喷涂开始与结束的20个点数据，并将数据代入蚁群算法中，仿真获得如下喷涂路径：

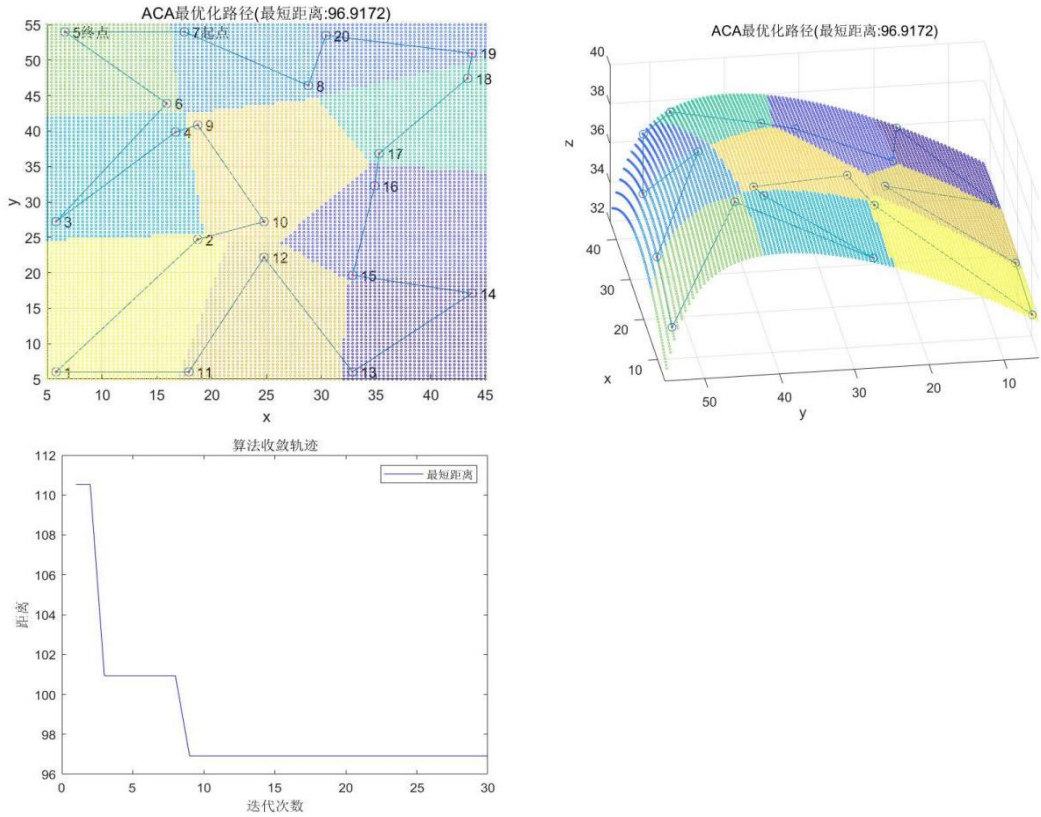


图20 子片连接顺序与优化过程图

根据仿真结果本文知道，最短路径为7、8、20、19、18、17、16、15、14、13、12、11、1、2、10、9、4、3、6、5、7，迭代13次后结果趋于稳定，且最短距离为96.9172。可以看到，基于蚁群算法的路径可以获取最短的片间路径长度，解决减少机械臂所走路径和喷涂所需时间的问题。

6.4 对于变倾角动态喷涂模型的分析与检验

为检验变倾角动态喷涂模型的合理性，本文选择90°外犄角为例进行仿真和检验。设 $T_d = 8\mu m$ ，喷枪的嘴角喷涂高度 H_0 及喷涂张角 φ 分别为200mm和28°。喷枪垂直喷涂时，进行垂直喷涂实验，拟合平面涂层生长速率函数（单位： $\mu m/s$ ）为： $f(r) = 0.1[(H_0 \tan(\varphi/2))^2 - r^2]$ ，垂直喷涂时优化后的喷枪行走速率与相邻轨迹间距分别为323.2mm/s和60.8mm，喷枪允许的喷涂高度范围为[150mm, 250mm]。

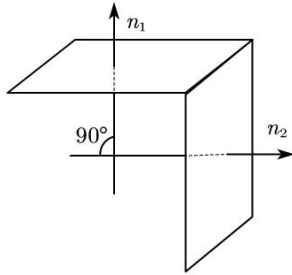


图21 外犄角示意图

如上图所示，以 90° 的外犄角面片组合为例，对应的喷涂倾角 α_{th} 为 45° ，获取 $a_{1\max} = 117.9\text{mm}$ ， $a_{2\max} = 70.7\text{mm}$ 。

这里以 $d_1 = 200\text{mm}$ ， $d_2 = 170\text{mm}$ 为例建立连续变倾角和定倾角喷涂区域内的动态喷涂涂层厚度模型。此时由于 $d_1 \geq H_0 \tan(\varphi/2) = 50\text{mm}$ ，则 T_{s_2} 选择下式进行计算。另设 $i = n = 6$ ，建立连续变倾角和定倾角喷涂区域内任意一点的涂层厚度叠加模型，使用MATLAB中结合模式搜索工具箱编写算法求解，获取优化后的高度参数如下所示：

表2 优化后的喷涂高度参数

H (mm)					
H ₁	H ₂	H ₃	H ₄	H ₅	H ₆
192.15	202.41	187.90	217.35	249.96	249.99
H ₇	H ₈	H ₉	H ₁₀	H ₁₁	H ₁₂
150.90	170.66	193.42	216.15	226.57	239.67

由上表可知，本文将变倾角过程分为六个阶段，每个阶段对应的优化喷涂高度分别为 $H_1 - H_6$ ；同理，恒倾角喷涂阶段也分为六个，每个阶段对应的优化喷涂高度分别为 $H_7 - H_{12}$ ，则该喷涂高度的方案下，对于 90° 的外犄角的喷涂效果最好。

在上述情况中，采用粒子群算法求得犄角喷涂的实际厚度与理想喷涂厚度偏差平方和为29.4667。可以看到，由于犄角本身的特殊性，无法完全做到像平面喷涂一样的高效无误差喷涂。但是在连续变倾角动态喷涂模型下，犄角已经在按照优化后的喷涂高度运作的机械臂下实现了良好的喷涂效果，一定程度上解决了犄角的喷涂问题。

七 模型的评价

7.1 模型优点

- 1.有理多项式表达模型拟合效果极好，拟合获得的曲面与离散点图的相关系数高达0.9983，可以用该有理多项式几乎完美地对曲面进行描述和表达。
- 2.K_means聚类算法将曲率的因素考虑进去，避免了单一欧几里得距离的影响，考虑较为全面，将曲面分成了十个近似平面的子片，且聚类效果较好。
- 3.模型一采用点云数据拟合曲面，获取到的点云数据为后续的分析提供了较大的帮助。

7.2 模型缺点

- 1.本模型对于复杂物体的分析局限在曲面。实际上，复杂物体的表面还有柱面、球面、凹凸面等，它们不便于使用有理多项式对其直接进行数学描述，只能使用微元法将其分片。没有对以上几种情况进行验证，存在一定的应用缺陷。
- 2.对于曲面子片内的路径规划仅局限于对现有的路径规划进行评价，选择最优路径，无法自行根据子片本身来自动生成最优的路径；由于给出的路径规划方案较少，选择出的方案只能是两者之间较优的方案，代表性不高。
- 3.对于犄角的模型建立中，本文忽略了在喷涂面片1时对面片2造成的影响。但是在喷涂犄角的实际操作中，在一个面片喷涂必定会对另一个面片产生造成影响。

7.3 模型推广

本文主要针对复杂物体中的平面、曲面和犄角三种情况提出对应的喷涂模型，但是实际的工业生产当中还会面临更多的困难与挑战，例如柱面、球面、凹凸面等。若将该模型进一步分析研究，提出更多的路径规划方案，则本文将具有极大的应用价值，在航空航天、医疗器械、家具喷刷等方面发挥着重要的作用。

八 参考文献

- [1] 朱心雄.自由曲线曲面造型技术[M].北京：科学出版社，2000.
- [2] 孙义东. 基于曲面分割的喷漆轨迹规划方法研究[D]. 黑龙江:哈尔滨工业大学,2018.
DOI:10.7666/d.D01587891.
- [3] 程昶运,熊瑞平,王波,等. 壳体曲面的机械臂喷涂路径规划方法研究[J]. 组合机床与自动化加工技术,2020(3):49-54. DOI:10.13462/j.cnki.mmtamt.2020.03.012.
- [4] 张永贵,黄玉美,高峰,等. 喷漆机器人空气喷枪的新模型[J]. 机械工程学报,2006,42(11):226-233.
DOI:10.3321/j.issn:0577-6686.2006.11.037.
- [5] 张淑珍. 基于活动标架的喷涂机器人喷枪轨迹规划研究[D]. 甘肃:兰州理工大学,2013.
DOI:10.7666/d.Y2567194.
- [6] 刘冬明. 犄角特征曲面倾角喷涂轨迹规划及仿真研究[D]. 江苏:江苏大学,2019.
- [7] Gasparetto A, Vidoni R, Saccavini E. Optimal Path Planning For Painting Robots. Proceedings of the ASME 2010 10th Biennial Conference on Engineering Systems Design and Analysis. Istanbul, Turkey, 2010: 1-8.

附录

附录1 点云数据的读取代码

```
% 点云数据的读取与输出
a = read_wobj("sofa1.obj");
x = a.vertices(:,1);
z = a.vertices(:,2);
y = a.vertices(:,3);
scatter3(x, y, z, 1,z);
xlabel('x');
ylabel('y');
zlabel('z');
xyz = [x,y,z];
```

附录二 拟合函数的高斯曲率的求解

```
% 求高斯曲率
n = 30;
x = linspace(5, 45, n);
y = linspace(5, 55, n);
[x, y] = meshgrid(x, y);
z = zeros(n);
for i = 1:n
    for j = 1:n
        z(i,j) = ni_he_fun(x(i,j), y(i,j));
    end
end

mc = mcurvature(x, y, z);
gc = gcurvature(x, y, z);
```

附录三 1stOpt拟合表达式结果

Function: $z = (p1+p2*x+p3*y+p4*y^2+p5*y^3)/(1+p6*x+p7*x^2+p8*y)$
Algorithms: 麦夸特法(Levenberg-Marquardt) + 通用全局优化法
Root of Mean Square Error (RMSE): 0.0889013331517269
Sum of Square Error (SSE): 12.18711532975
Correlation Coef. (R): 0.998304960282925
R-Square: 0.996612793725493
Determination Coef. (DC): 0.996612793322826
Parameters Name Parameter Value

p1	32.5397935824239
p2	-1.74018452670656E-5
p3	-0.314596830093201
p4	-0.00564951611306279
p5	2.74497493502093E-5
p6	-0.000675850249664152
p7	1.35073771075182E-5
p8	-0.01661124127415

附录四 椭圆双 β 分布模型

```
function q = beta_fun(qu, a, b, x, y)
    beta1 = 2.47;
    beta2 = 4.72;
    ymax = b*(1-x.^2/a.^2).^(0.5);
    if y > -ymax & y < ymax
        q = qu*(1-(x.^2)/(a.^2)).^(beta1-1)*(1-(y).^2/(b.^2*(1-(x.^2)/(a.^2)))).^(beta2-1);
    else
        q = -0.01;
    end
end
```

附录五 使用黄金分割法求偏差平方和的最小值

```
R = 130;
interval_left=R; % 初始搜索区间左边界点
interval_right=2*R; % 初始搜索区间右边界点
cycle_times=100; % 迭代次数
precision = 0.001; % 算法精度参数
right_interval_flag=0; % 右侧区间标志，1=选右侧区间，0=选左侧区间

interval_length = interval_right - interval_left; % 计算搜索区间长度
insert_point_left = interval_left + 0.382*interval_length; % 计算黄金分割法的左插入点
insert_point_right = interval_left + 0.618*interval_length; % 计算黄金分割法的右插入点

% 开始迭代
while interval_length>precision
    value_left = lossfun(insert_point_left); % 计算左插入点的函数值
    value_right = lossfun(insert_point_right); % 计算右插入点的函数值
    if value_left>value_right
        right_interval_flag=1; % 选择右侧区间
        interval_left = insert_point_left; % 压缩区间后，更新左边界点
        insert_point_left = insert_point_right; % 下次迭代运算左插入点
    else
        right_interval_flag=0; % 选择左侧区间
        interval_right = insert_point_right; % 压缩区间后，更新右边界点
        insert_point_right = insert_point_left; % 下次迭代运算右插入点
    end
    interval_length = interval_right - interval_left; % 计算区间长度
    cycle_times=cycle_times+1; % 迭代计数

    if right_interval_flag==1
        % 右区间，则计算下次迭代运算的右插入点
        insert_point_right = interval_left + 0.618*interval_length;
    else
        % 左区间，则计算下次迭代运算的左插入点
        insert_point_left = interval_left + 0.382*interval_length;
```

```

    end
end
% 迭代结束

xx = 0.5 * (interval_right + interval_left);
yy = lossfun(x);

%lossfun.m
function sum_loss = lossfun(w)
    a = 121.99;
    qd = 0.87;
    x = linspace(0, w, 100);
    q = zeros(length(x), 1);
    loss = zeros(length(x), 1);
    for i = 1:length(x)
        q(i) = cdfun(x(i), w);
        loss(i) = (q(i) - qd)^2;
    end
    sum_loss = sum(loss);
end
end

```

附录六 K_means 算法

```

%K-Means算法
function [ k_means_result,cls_vec ] = k_means( data_set,weight,K,plot2pos )
%% 初始化各项参数
[N,dim] = size(data_set);
if isempty(weight)
    weight = ones(N,1);
else
    weight = weight(:);
end
if isempty(plot2pos)
    isdisplay = 0;
else
    isdisplay = 0;
end

% 初始化点集
group_center = mean(data_set);
group_range = range(data_set);
centers = (randn(K,dim).*repmat(group_range,K,1)./3+repmat(group_center,K,1));

%各个变量的初始化
dist_vec = zeros(N,K);
rep_tms = 0;
delay = 1e-3;
dumper = 0.1;
p2pc = zeros(K,2);

```

```

%J = 0;
logJ = [];
minJ = -1;

%%为画图设置各项Legend
if isdisplay == 1
    Style = {'o'};
    Color = {'r','g','b','m','k','c','y'};
    Legends = {};
    for i = 1:length(Style)
        for j = 1:length(Color)
            Legends = [Legends,[Style{i},Color{j}]];
        end
    end
end

%% 迭代过程
while(1)

    for i = 1:K
        %求到每一个中心的距离向量，且不用张量求更加节约空间
        dist_vec(:,i) = sqrt(sum(...
            (data_set - repmat(centers(i,:),N,1)).^2,2)...
        ));
    end
    %求出每一个点离哪一个中心更近。
    [~,cls_vec] = min(dist_vec,[],2);
    lst_cnt = centers;
    for i = 1:K
        %根据权重重新计算中心
        cls_idx = find(cls_vec==i);
        if isempty(cls_idx)
            centers(i,:) = (randn(1,dim).*group_range./3+group_center);
        else
            centers(i,:) = sum(data_set(cls_idx,:).*...
                tan(repmat(weight(cls_idx),1,dim))...
                ./sum(weight(cls_idx)));
        end
    end

    % 计算代价函数
    CMat = (data_set-lst_cnt(cls_vec,:)).^2;
    J = sum(CMat(:));

    % 根据阻尼比更新
    centers = (1-dumper).*centers + dumper.*lst_cnt;

    if minJ<0 || J<minJ
        minJ=J;
    end
end

```



```

if J>=minJ
    rep_tms = rep_tms+1;
else
    rep_tms = 0;
end

if rep_tms>=5
    break;
end

if isdisplay == 1
    % 绘制代价函数及其差分
    figure(2)
    logJ = [logJ J]; %#ok<AGROW>
    subplot(2,1,1)
    plot(logJ)
    axis tight
    subplot(2,1,2)
    plot(diff(logJ))

    % 绘制分布图
    figure(1)
    cla
    hold on
    for i = 1:K
        cls_idx = find(cls_vec==i);
        if isempty(cls_idx)
            p2pc(i,:) = mean(plot2pos);
        else
            p2pc(i,:) = sum(...
                plot2pos(cls_idx,:).*...
                repmat(weight(cls_idx),1,2)...
            )./sum(weight(cls_idx));
        end
        plot(p2pc(i,1),p2pc(i,2),Legends{1+mod(i,length(Legends))});
        scatter(plot2pos(cls_idx,1),plot2pos(cls_idx,2),...坐标
            weight(cls_idx)./max(weight).*10,...大小
            Color{1+mod(i,length(Color))},'filled');%颜色
    end

    hold off
    pause(delay)
end

end

%% 输出数据集
not_empty_cls = unique(cls_vec);
k_means_result = cell(length(not_empty_cls),1);

```

```

for i = 1:length(not_empty_cls)
    cls = not_empty_cls(i);
    sub_res = cell(3,1);
    cls_idx = find(cls_vec==cls);
    sub_res{1} = centers(cls,:);
    sub_res{3} = cls_idx;
    sub_res{2} = data_set(cls_idx,:);
    k_means_result{i} = sub_res;
end
end

```

附录七 蚁群算法计算子片间最优顺序

```

% 程序运行计时开始
t0 = clock;

% 导入数据
load('xyz20.mat')
citys = xyz20;

%% 计算城市间距离
n = 20;
D = zeros(n,n);
for i = 1:n
    for j = 1:n
        if i ~= j
            if (i - j == 1 & mod(i,2) == 0) | (i - j == -1 & mod(j,2) == 0)
                D(i,j) = 1e-4;
            else
                D(i,j) = sqrt(sum( ( citys(i,:) - citys(j,:) ).^2 ));
            end
        else
            D(i,j) = 1e-4;      %设定的对角矩阵修正值
        end
    end
end
end

%% 初始化参数
m = 12;                % 蚂蚁数量
alpha = 1;             % 信息素重要程度因子
beta = 5;              % 启发函数重要程度因子
vol = 0.2;             % 信息素挥发(volatilization)因子
Q = 10;               % 常系数
Heu_F = 1./D;          % 启发函数(heuristic function)
Tau = ones(n,n);       % 信息素矩阵
Table = zeros(n,n);    % 路径记录表
iter = 1;              % 迭代次数初值
iter_max = 30;         % 最大迭代次数

```

```

Route_best = zeros(iter_max,n); % 各代最佳路径
Length_best = zeros(iter_max,1); % 各代最佳路径的长度
Length_ave = zeros(iter_max,1); % 各代路径的平均长度
Limit_iter = 0; % 程序收敛时迭代次数

%% 迭代寻找最佳路径
while iter <= iter_max
    % 随机产生各个蚂蚁的起点城市
    start = zeros(m,1);
    for i = 1:m
        temp = randperm(n);
        start = temp(1);
    end
    Table(:,1) = start;
    % 构建解空间
    citys_index = 1:n;
    % 逐个蚂蚁路径选择
    for i = 1:m
        % 逐个城市路径选择
        for j = 2:n
            tabu = Table(i,1:(j - 1)); % 已访问的城市集合（禁忌表）
            allow_index = ~ismember(citys_index,tabu); % 参数说明1（程序底部）
            allow = citys_index(allow_index); % 待访问的城市集合
            P = allow;
            % 计算城市间转移概率
            for k = 1:length(allow)
                P(k) = Tau(tabu(end),allow(k))^alpha * Heu_F(tabu(end),allow(k))^beta;
            end
            P = P / sum(P);
            % 轮盘赌法选择下一个访问城市
            Pc = cumsum(P); % 参数说明2（程序底部）
            target_index = find(Pc >= rand);
            target = allow(target_index(1));
            Table(i,j) = target;
        end
    end
    % 计算各个蚂蚁的路径距离
    Length = zeros(m,1);
    for i = 1:m
        Route = Table(i,:);
        for j = 1:(n - 1)
            Length(i) = Length(i) + D(Route(j),Route(j + 1));
        end
        Length(i) = Length(i) + D(Route(n),Route(1));
    end
End

```

```

% 计算最短路径距离及平均距离
if iter == 1
    [min_Length, min_index] = min(Length);
    Length_best(iter) = min_Length;
    Length_ave(iter) = mean(Length);
    Route_best(iter,:) = Table(min_index,:);
    Limit_iter = 1;
else
    [min_Length,min_index] = min(Length);
    Length_best(iter) = min(Length_best(iter - 1),min_Length);
    Length_ave(iter) = mean(Length);
    if Length_best(iter) == min_Length
        Route_best(iter,:) = Table(min_index,:);
        Limit_iter = iter;
    else
        Route_best(iter,:) = Route_best((iter - 1),:);
    end
end

% 更新信息素
Delta_Tau = zeros(n,n);
% 逐个蚂蚁计算
for i = 1:m
    % 逐个城市计算
    for j = 1:(n - 1)
        Delta_Tau(Table(i,j),Table(i,j+1)) = Delta_Tau(Table(i,j),Table(i,j+1)) + Q/Length(i);
    end
    Delta_Tau(Table(i,n),Table(i,1)) = Delta_Tau(Table(i,n),Table(i,1)) + Q/Length(i);
end
Tau = (1 - vol) * Tau + Delta_Tau;
% 迭代次数加1,清空路径记录表
iter = iter + 1;
Table = zeros(m,n);
end

%% 结果显示
[Shortest_Length,index] = min(Length_best);
Shortest_Route = Route_best(index,:);
Time_Cost = etime(clock,t0);
disp(['最短距离:' num2str(Shortest_Length)]);
disp(['最短路径:' num2str( [Shortest_Route Shortest_Route(1)] )]);
disp(['收敛迭代次数:' num2str(Limit_iter)]);
disp(['程序执行时间:' num2str(Time_Cost),'秒']);

%% 绘图
figure(1)
plot3([
    citys(Shortest_Route,1);citys(Shortest_Route(1),1)
    [ citys(Shortest_Route,2);citys(Shortest_Route(1),2) ],[ citys(Shortest_Route,3);citys(Shortest_Route(1),3) ], 'o-');

```

```

grid on;
for i = 1:size(citys,1)
    text(citys(i,1),citys(i,2),[' ' num2str(i)]);
end
text(citys(Shortest_Route(1),1),citys(Shortest_Route(1),2),' 起点');
text(citys(Shortest_Route(end),1),citys(Shortest_Route(end),2),' 终点')
xlabel('x');
ylabel('y');
title(['ACA最优化路径(最短距离:' num2str(Shortest_Length) ')']);
figure(2);
plot(1:iter_max,Length_best,'b');
legend('最短距离');
xlabel('迭代次数');
ylabel('距离');
title('算法收敛轨迹');

%% 程序解释或说明
% 1.ismember函数判断一个变量中的元素是否在另一个变量中出现,返回0-1矩阵;
% 2. cumsum函数用于求变量中累加元素的和,如A=[1,2,3,4,5],那么
cumsum(A)=[1,3,6,10,15]。

```

附录八 连续变倾角动态喷涂建模代码

```

% f.m
function ff = f(rs, a)
    A = 0.001;
    H0 = 200;      % 喷枪高度
    fai = 28/180*pi; % 喷涂张角
    if abs(rs) >= 50
        ff = 0;
    else
        ff = A*((H0*tan(fai/2)).^2-rs.^2)*cos(a); % -50 < rs < 50
    end
end

% a1.m
function aa1 = a1(H, a)
    fai = 1;
    aa1 = (H*tan(fai/2).^2*sin(a) - H*tan(fai/2)*cos(a))/(tan(fai/2)*sin(2*a)-1-
tan(fai/2).^2*sin(a).^2+sin(a).^2);
end

% a2.m
function aa2 = a2(H, a)
    fai = 1;
    aa2 = (H*tan(fai/2).^2*sin(a) +
H*tan(fai/2)*cos(a))/(tan(fai/2)*sin(2*a)+1+tan(fai/2).^2*sin(a).^2-sin(a).^2);
end

```

```

% t11.m
function t11 = t11(x,y)
    H0 = 200;      % 喷枪高度
    fai = 28/180*pi; % 喷涂张角
    y0 = 0;
    v = 323.2;    % 优化后的行走速度
    delta = 60.8; % 优化后的相邻轨迹间距

    fun = @(z) f(((y+y0).^2+(x+H0*tan(fai/2)+z).^2).^0.5,0);
    t11 = 1/v * integral(fun, -H0*tan(fai/2)+0.1, H0*tan(fai/2)+0.1,'ArrayValued',true);
    fun = @(z) f(((delta - y+y0).^2+(x+H0*tan(fai/2)+z).^2).^0.5,0);
    t11 = t11 + 1/v * integral(fun, -H0*tan(fai/2)+0.1,
    H0*tan(fai/2)+0.1,'ArrayValued',true);
end

% t12.m
function t12 = t12(x,y,Hjj)
    d1 = 200;      % s2 区域长度
    delta = 60.8;  % 优化后的相邻轨迹间距
    v = 363.2;    % 优化后的行走速度
    i = 6;         % 分段个数
    y0=0;
    ath = pi/4;    % j/i*ath 表示第j段相应的倾角
    % Hjj=ones(i,1);
    t12 = 0;
    for j = 1:i
        Hj = Hjj(j);
        sb = ((x.^2*cos(j/i*ath).^2+y.^2)./(x.^2+y.^2)).^(0.5);
        fun = @(z) f((Hj*sb*((y+y0).^2+(x-z).^2).^(0.5))./(Hj+(x-z)*sin(j/i*ath)),j/i*ath);
        t12 = t12 + 1/v * integral(fun, (j-1)/i*d1, j/i*d1,'ArrayValued',true);

        sb = ((x.^2*cos(j/i*ath).^2+(delta-y).^2)./(x.^2+(delta-y).^2)).^(0.5);
        fun = @(z) f((Hj*sb*(((delta-y)+y0).^2+(x-z).^2).^(0.5))./(Hj+(x-
z)*sin(j/i*ath)),j/i*ath);
        t12 = t12 + 1/v * integral(fun, (j-1)/i*d1, j/i*d1,'ArrayValued',true);
    end
end

% 粒子群算法
% Default options

options.CognitiveAttraction = 0.5 ;
options.ConstrBoundary = 'penalize' ;
options.AccelerationFcn = @psoiterate ;
options.DemoMode = 'off' ;
options.Display = 'final' ;
options.FitnessLimit = -inf ;
options.Generations = 50 ;
options.HybridFcn = [] ;

```

```

options.InitialPopulation = [] ;
options.InitialVelocities = [] ;
options.KnownMin = [] ;
options.OutputFcns = {} ;
options.PlotFcns = {} ;
options.PlotInterval = 1 ;
options.PopInitRange = [0;1] ;
options.PopulationSize = 40 ;
options.PopulationType = 'doubleVector' ;
options.SocialAttraction = 1.25 ;
options.StallGenLimit = 50 ;
options.StallTimeLimit = Inf ;
options.TimeLimit = Inf ;
options.TolCon = 1e-6 ;
options.TolFun = 1e-6 ;
options.UseParallel = 'never' ;
options.Vectorized = 'off' ;
options.VelocityLimit = [] ;

```

```

hmin = 150;
hmax = 250;
LB = ones(1,12)*hmin;
UB = ones(1,12)*hmax;
A = [];
b = [];
Aineq = [];
bineq = [];
Aeq = [];
beq = [];
fitnessfcn = @E;
nvars = 12;
x0 = ones(12, 1)*200;
nonlcon = [];

```

```

[xOpt,fval,exitflag,output,population,scores]
pso(fitnessfcn,nvars,Aineq,bineq,Aeq,beq,LB,UB,nonlcon,options);

```

=