

基于 AIWCPSO 算法的三次样条气动参数插值方法

李军伟, 程咏梅, 陈克喆, 阮晓明, 宋春华

(西北工业大学 自动化学院, 西安 710072)

摘要: 针对飞行仿真建模过程中气动参数以矩阵的形式给出, 大都存在着非线性关系, 提出一种基于自适应惯性权重的混沌粒子群优化(AIWCPSO)算法的三次样条气动参数插值方法. 首先建立粒子与三次样条插值函数中系数的映射关系; 然后利用 AIWCPSO 算法对三次样条插值函数的系数进行寻优, 将获得的最优解近似看作三次样条插值函数的系数; 最后计算得到离散点的气动参数. 仿真实验结果表明, 所提出的方法能有效地解决飞行气动参数插值问题.

关键词: 气动参数; 三次样条插值; 混沌; 粒子群优化; 自适应惯性权重

中图分类号: TP391

文献标志码: A

Cubic spline interpolation method of aerodynamic parameters based on AIWCPSO algorithm

LI Jun-wei, CHENG Yong-mei, CHEN Ke-zhe, RUAN Xiao-ming, SONG Chun-hua

(College of Automation, Northwestern Polytechnical University, Xi'an 710072, China. Correspondent: LI Jun-wei, E-mail: lijunwei@mail.nwpu.edu.cn)

Abstract: Since the aerodynamic parameters for flight simulation modeling process are given in the form of a matrix, most of which are in the nonlinear relationship, a cubic spline interpolation method of aerodynamic parameters based on the adaptive inertia weight chaotic particle swarm optimization(AIWCPSO) algorithm is proposed. Firstly, the mapping relationship between the particle and the coefficients of the cubic spline interpolation function is established. Then the optimal value of the coefficient of the cubic spline interpolation function is searched by using AIWCPSO algorithm, which is taken as the coefficient of the cubic spline interpolation function. Finally, the aerodynamic parameters of discrete points are obtained by calculating functions. The simulation results show that the proposed method can be effectively used in flight dynamics modeling of the aerodynamic parameters.

Key words: aerodynamic parameter; cubic spline interpolation; chaos; particle swarm optimization(PSO); adaptive inertia weight

0 引言

随着计算机技术和仿真技术的迅速发展, 飞行仿真可以为飞行器研制过程中方案论证、设计分析等各个阶段进行全面的系统分析和评估^[1]. 飞行动力学仿真模型是飞行仿真的基础, 负责飞机各系统的功能和特性的模拟, 直接影响到飞行仿真系统的实时性和真实性^[2]. 通常情况下, 在计算飞行器气动力和气动矩时, 相关的气动参数可通过风洞实验或者空中试飞实验获得, 但气动参数作为因变量以表格或矩阵的形式给出迎角或侧滑角等作为自变量的函数时, 具有离散性和不完整性, 大都存在着非线性关系, 往往不

能获得飞机在整个飞行包线内各种状态下的气动参数^[3]. 因此, 在气动力和力矩建模时需要对气动参数的实验值进行曲线拟合或估算, 使之成为光滑的连续函数.

目前国内外学者已提出了一些数据插值方法, 主要包括线性插值法(LI)、拉格朗日插值法(LAI)和三次样条插值法(CSI)等. 当因变量与自变量之间呈现出明显的非线性关系时, LI的精确度将很难保证; LAI在靠近插值区间两端时可能会出现龙格现象; CSI具有很好的光滑性, 但涉及到矩阵的求逆, 离散点越多, 矩阵求逆计算量越大.

收稿日期: 2012-09-12; 修回日期: 2012-11-08.

基金项目: 航空科学基金项目(20100853010).

作者简介: 李军伟(1981—), 男, 博士生, 从事人工智能、飞行仿真的研究; 程咏梅(1960—), 女, 教授, 博士生导师, 从事多目标跟踪与识别、惯性导航、信息融合、人工智能等研究.

粒子群优化 (PSO) 算法^[4-5]是源于自然界鸟群和鱼群群体觅食行为启发下的基于迭代运算的全局智能优化算法。PSO 算法具有进化初期收敛速度快、运算简单、易于实现等优点, 已广泛应用于函数优化、神经网络训练及其他应用领域^[6-8]。但 PSO 算法存在着易陷入局部最优解、过早收敛等不足, 在后期进化中很难获得更精确的解。对此, 国内外学者提出了很多改进算法^[9-11], 这些算法在不同程度上改善了 PSO 算法的优化性能, 但在保持种群多样化和寻优收敛搜索等方面还有待于改进。而混沌的随机性、遍历性和规律性等特点能在一定范围内按其自身规律不重复地遍历所有状态, 将混沌特性与 PSO 算法结合可以提高初始粒子种群的质量。

为此, 本文将气动参数插值转化为三次样条曲线的参数优化问题, 借助于 AIWCPSO 算法并结合 CSI 解决气动参数的插值问题, 提出一种基于自适应惯性权重的混沌粒子群优化 (AIWCPSO) 算法的三次样条气动参数插值方法 (AIWCPSO-CSI)。仿真实验验证了 AIWCPSO 算法的有效性和可行性, 利用 AIWCPSO 算法对三次样条插值函数系数进行寻优, 可有效地满足飞行仿真中气动力建模的需求。

1 三次样条插值

三次样条插值^[12]: 设函数 $y = f(x)$ 是定义在区间 $[a, b]$ 的并有 $n + 1$ 个互异节点 $a = x_0 < x_1 < \dots < x_n = b$, 函数 $y = f(x)$ 在这些节点上的值分别为 $f(x_i) = y_i (i = 0, 1, \dots, n)$ 。三次样条插值函数要求分段表示的函数 $\phi(x)$ 符合以下条件:

1) $\phi(x)$ 在子区间 $[x_{i-1}, x_i] (i = 1, 2, \dots, n)$ 上的表达式 $\phi_i(x)$ 都是次数不超过 3 的多项式;

2) $\phi(x_i) = y_i, i = 0, 1, \dots, n$;

3) $\phi(x), \phi'(x), \phi''(x)$ 在区间 $[x_{i-1}, x_i]$ 上都连续。

假设 $\phi''(x_i) = m_i, \phi''(x_{i-1}) = m_{i-1}$, 由于在子区间 $[x_{i-1}, x_i]$ 上 $\phi(x_i) = y_i (i = 1, 2, \dots, n)$ 是不高于 3 次的多项式, 其二阶导数为线性函数或常数, 可得

$$\phi_i''(x) = m_{i-1} \frac{x_i - x}{h_i} + m_i \frac{x - x_{i-1}}{h_i}. \quad (1)$$

其中: $x \in [x_{i-1}, x_i], h_i = x_i - x_{i-1}$ 。根据插值条件 $\phi_i(x_{i-1}) = y_{i-1}, \phi_i(x_i) = y_i$, 连续对其进行两次积分可得

$$\begin{aligned} \phi_i(x) &= m_{i-1} \frac{(x_i - x)^3}{6h_i} + m_i \frac{(x - x_{i-1})^3}{6h_i} + \\ &\left(y_{i-1} - \frac{m_{i-1}h_i^2}{6} \right) \frac{x_i - x}{h_i} + \\ &\left(y_i - \frac{m_i h_i^2}{6} \right) \frac{x - x_{i-1}}{h_i}. \end{aligned} \quad (2)$$

利用 $\phi'(x)$ 在内节点 x_i 处连续的条件 $\phi_i'(x_i - 0) = \phi_i'(x_i + 0)$, 可得线性方程组

$$m_{i-1}m_i + 2m_i + \lambda_i m_{i+1} = f_i. \quad (3)$$

其中

$$m_i = \frac{h_i}{h_i + h_{i+1}}, \quad (4)$$

$$\lambda_i = \frac{h_{i+1}}{h_i + h_{i+1}}, \quad (5)$$

$$f_i = \frac{6}{h_i + h_{i+1}} \left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i} \right). \quad (6)$$

因此, 对于插值函数 $\phi(x)$ 需要确定 $n + 1$ 个未知参数 m_0, m_1, \dots, m_n 。加入自然边界条件 $\phi''(x_0) = 0, \phi''(x_n) = 0$, 将式 (3) 改为如下线性方程组:

$$\begin{bmatrix} 2 & \lambda_1 & & & & & \\ m_2 & 2 & \lambda_2 & & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & m_{n-2} & 2 & \lambda_{n-2} & \\ & & & & & m_{n-1} & 2 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_{n-2} \\ m_{n-1} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n-2} \\ f_{n-1} \end{bmatrix}. \quad (7)$$

由式 (7) 和边界条件求解出 m_0, m_1, \dots, m_n , 再将所得的结果代入式 (2), 便可得到所求解的三次样条函数的分段表达式。但是, 当离散样本点越多时, 三次样条函数的矩阵求逆将越繁琐。

2 基于 AIWCPSO 算法的三次样条气动参数插值方法

2.1 PSO 算法的基本原理

假设在 D 维搜索空间中, 有 N 个粒子组成的群体, 粒子群中第 i 个粒子 ($i = 1, 2, \dots, N$) 的位置向量表示为 $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$; 粒子经历过的历史最优位置为 $\mathbf{pbest}_i = (\text{pbest}_{i1}, \text{pbest}_{i2}, \dots, \text{pbest}_{iD})$, $\mathbf{gbest} = (\text{gbest}_1, \text{gbest}_2, \dots, \text{gbest}_D)$ 为 \mathbf{pbest}_i 中的最优; 第 i 个粒子的速度向量为 $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{iD})$, 每个粒子根据下式更新第 $d (d = 1, 2, \dots, D)$ 维速度和位置^[5]:

$$v_{id}(k+1) = v_{id}(k) + c_1 r_1 [\text{pbest}_{id}(k) - x_{id}(k)] + c_2 r_2 [\text{gbest}_d(k) - x_{id}(k)], \quad (8)$$

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1). \quad (9)$$

其中: 式 (8) 为速度更新, 式 (9) 为位置更新; k 为迭代次数; r_1 和 r_2 为区间 $[0, 1]$ 上均匀分布随机数; c_1 和 c_2 为学习因子, 且 $c_1 > 0, c_2 > 0$ 。

在式 (8) 中引入惯性权重 w_k , 便可形成标准的粒子群优化 (SPSO) 算法, 其速度更新公式为

$$v_{id}(k+1) = w_k v_{id}(k) + c_1 r_1 [\text{pbest}_{id}(k) - x_{id}(k)] + c_2 r_2 [\text{gbest}_d(k) - x_{id}(k)], \quad (10)$$

$$w_k = w_{\max} - \frac{w_{\max} - w_{\min}}{k_{\max}} k. \quad (11)$$

其中: w_{\max}, w_{\min} 分别为最大、最小惯性权重, k_{\max}

为最大迭代次数. w_k 具有调整全局和局部搜索能力的平衡作用, 当 w_k 较大时有利于全局搜索最优, 而 w_k 较小时适于最终结果的收敛.

2.2 混沌初始化策略

混沌一般是指由确定性方程得到的具有随机性的运动状态, 混沌状态是非线性系统中一种较为普遍的现象. 常用的混沌映射有 Logistic 映射、Tent 映射和立方映射等.

Logistic 映射生成混沌序列公式为

$$\begin{cases} x_{n+1} = \mu x_n(1 - x_n), \\ 0 < x_0 < 1, \end{cases} \quad (12)$$

其中 $\mu \in (2, 4]$ 为控制参数. 当 $\mu = 4$ 时, Logistic 映射将完全处于混沌状态, 此时产生的混沌变量具有较好的遍历性. 但是 Logistic 映射均匀性差^[13], 需要增加迭代次数才能达到映射点在空间中较广的分布; 否则其映射点在边缘处密度很高, 在中间区域密度较低.

Tent 映射又称帐篷映射, 是分段线性的一维映射, 具有均匀的分布密度、功率谱密度和理想的相关特性. 其数学表达式如下:

$$x_{n+1} = \begin{cases} 2x_n, & 0 \leq x_n \leq 0.5; \\ 2(1 - x_n), & 0.5 < x_n \leq 1; \end{cases} \quad (13)$$

$$x_{n+1} = (2x_n) \bmod 1. \quad (14)$$

Tent 映射的混沌序列结构简单, 迭代过程中适合计算机运行. 但对于 $[0, 1]$ 之间的浮点数进行 Tent 映射时, 其实是将小数部分的二进制数进行无符号左移. 由于计算机字长有限, 小数部分的二进制序列经过一定次数的无符号左移运算将趋向于零^[14-15], 并趋向于 Tent 映射 $\{0, 0.25, 0.5, 0.75\}$ 的不动点. 序列中还存在小周期, 如 4 周期 $\{0.2, 0.4, 0.6, 0.8\}$, 严重影响了混沌优化算法的寻优效率.

立方映射生成混沌序列公式如下:

$$\begin{cases} x_{n+1} = 4x_n^3 - 3x_n, \\ -1 < x_0 < 1. \end{cases} \quad (15)$$

立方映射遍历性分布区间较均匀^[16], 而且只要迭代初值不为零, 混沌序列就会产生, 保证了种群在解空间内的均匀分布, 从而提高了算法的寻优效率.

在 PSO 算法中, 种群的初始化通常采用随机方式选取, 初始值位置的分散程度以及在搜索空间中的分布是否均匀^[17], 将直接影响整个迭代搜索的收敛速度和算法的求解效率. 当解空间范围较大时, 有可能造成粒子初始化在整个解空间的不均匀分布. 将混沌映射和 PSO 算法结合起来, 既不改变粒子初始化具有的随机性, 又利用混沌特性提高了种群的多样性. 因此, 从混沌映射分布是否均匀的角度出发, AIWCPSO 算法中采用立方映射进行粒子群初始化.

2.3 算法基本思想及步骤

AIWCPSO-CSI 的基本思想是: 首先建立粒子群中粒子与三次样条插值函数系数的映射关系, 将粒子的位置对应于三次样条插值函数的系数; 然后利用混沌理论中的立方映射初始化粒子的位置和速度, 粒子通过个体极值和全局极值来更新粒子的速度和位置, 并将得到的最优解近似视为三次样条插值函数的系数; 最后确定三次样条插值函数, 进而获得离散点处的气动参数.

综上所述, AIWCPSO-CSI 的基本步骤如下.

Step 1 根据已知离散点数据 (x_i, y_i) ($i = 0, 1, \dots, n$) 建立粒子群中粒子与三次样条插值函数系数 m_0, m_1, \dots, m_n 的映射关系, 其中 $y_i = f(x_i)$. 将未知参数的一个可能解看成粒子群中的一个粒子, 并进行插值方法的参数初始化和混沌初始化.

Step 1.1 参数初始化.

初始化搜索粒子的个数 N , 搜索维数 D , 惯性权重最大值 w_{\max} 和最小值 w_{\min} , 学习因子 c_1, c_2 , 粒子的最大速度 v_{\max} 以及最大迭代次数 iter_{\max} 等参数.

Step 1.2 混沌初始化.

随机产生一个 D 维的且每个分量都在 $(-1, 1)$ 的随机数向量 $\mathbf{y}_1 = (y_{11}, y_{12}, \dots, y_{1D})$, 根据式 (15) 立方映射得到 $(M-1)$ 个混沌分量 $\mathbf{y}_2, \mathbf{y}_3, \dots, \mathbf{y}_M$, 其中 $M \gg N$. 将混沌分量 y_{id} 载入各优化变量的取值范围 $[a_i, b_i]$ 内, 则产生的第 i 个粒子在搜索空间第 d 维的坐标 x_{id} 表达式为

$$x_{id} = a_i + \frac{1}{2}(b_i - a_i)(1 + y_{id}). \quad (16)$$

计算每个 x_{id} 所对应的适应度函数, 并从中选取最优的 N 个适应度函数值所对应的 x_{jd} ($j = 1, 2, \dots, N$; $d = 1, 2, \dots, D$) 作为粒子位置的初始解. 其中, 适应度函数 f 采用已知数据点处的实际输出值 $\phi_i(x_i)$ 与期望输出值 y_i 之间的均方误差 (MSE) 的平方来表示, 即

$$f = \frac{1}{n} \sum_{i=1}^n ([\phi_i(x_{i-1}) - y_{i-1}]^2 + [\phi_i(x_i) - y_i]^2). \quad (17)$$

随机产生一个 D 维的且每个分量都在 $(-1, 1)$ 内的随机数向量 $\mathbf{z}_1 = (z_{11}, z_{12}, \dots, z_{1D})$, 根据式 (15) 立方映射得到 $(N-1)$ 个混沌分量 $\mathbf{z}_2, \mathbf{z}_3, \dots, \mathbf{z}_N$. 同理, 将混沌分量变换到速度变量的取值范围内, 得到新的 $N \times D$ 个向量作为粒子速度的初始解.

Step 2 计算更新后的 N 个粒子对应的适应度值, 并将当前迭代的粒子适应度值与个体极值 \mathbf{pbest}_i 以及全局极值 \mathbf{gbest} 比较, 对个体极值 \mathbf{pbest}_i 和全局极值 \mathbf{gbest} 进行更新.

Step 3 根据式 (18) 更新自适应惯性权重, 通过

式(9)和(10)更新下一步的位置和速度,将位置和速度限定在优化变量设置的允许范围内.

SPSO算法中的惯性权重采用线性递减策略,并不能反映实际的复杂非线性的寻优搜索过程^[9].为此,本文提出一种新的自适应惯性权重调整策略,将惯性权重 w_i ($i = 1, 2, \dots, N$)表示为迭代次数 $iter$ 、进化速度参数 esf_i 和聚集度参数 adf 的函数,表达式如下:

$$w_i = w_{\max} - (w_{\max} - w_{\min}) \frac{iter}{iter_{\max}} e^{-(\alpha esf_i - \beta adf)}. \quad (18)$$

其中: w_i 为第 i 个粒子的惯性权重, w_{\max} 为最大惯性权重, w_{\min} 为最小惯性权重; $iter$ 为当前迭代次数, $iter_{\max}$ 为最大迭代次数;与文献[10]不同, esf_i 为改进后的进化速度参数, adf 为聚集度参数,二者的表达式分别为

$$esf_i = \frac{\text{abs}[F(\text{pbest}_i^k) - F(\text{pbest}_i^{k-1})]}{\text{abs}F(\text{pbest}_i^k) + \text{abs}F(\text{pbest}_i^{k-1})}, \quad (19)$$

$$adf = \frac{\text{abs}[\min(F_{\text{best}}^k, F_{\text{avg}}^k)]}{\text{abs}[\max(F_{\text{best}}^k, F_{\text{avg}}^k)]}, \quad (20)$$

$F(\text{pbest}_i^k)$ 为粒子 i 的 k 次迭代时的最优适应度值, F_{best}^k 和 F_{avg}^k 分别为 k 次迭代时种群中所有粒子当前适应度值的最优值和平均值.

esf_i 考虑到粒子的算法运行状况,也反映了第 i 个粒子的进化速度, esf_i 越大,进化速度越快. adf 反映了种群粒子的聚集程度, adf 越大,粒子群的聚集程度越大.

根据PSO算法的收敛性和惯性权重等参数的关系^[18-19],只有满足如下关系式才能保证优化算法的收敛:

$$\begin{cases} w_i < 1, \\ c_1 r_1 + c_2 r_2 > 0, \\ 2w_i - (c_1 r_1 + c_2 r_2) + 2 > 0. \end{cases} \quad (21)$$

在AIWCPSO算法中选为 $c_1 = c_2 = 1.49618$,则惯性权重的取值范围选为 $[0.5, 0.9]$.

Step 4 根据下式计算当前粒子群体适应度方差 σ^2 ^[20]:

$$\sigma^2 = \sum_{i=1}^N \left(\frac{F(x_i^k) - F_{\text{avg}}^k}{\max[\max[\text{abs}[F(x_i^k) - F_{\text{avg}}^k]], 1]} \right)^2. \quad (22)$$

其中: N 为种群粒子的数目, $F(x_i^k)$ 为第 i 个粒子的当前适应度值, F_{avg}^k 为当前种群的平均适应度值. σ^2 反映了种群中粒子的收敛程度, σ^2 越小,越趋于收敛.若 σ^2 小于设定的阈值 ε 且理论上的最优适应度值优于当前种群的最好适应度值,则转Step 5,否则转Step 6.

Step 5 加入混沌扰动策略进行位置更新,使算法跳出局部最优.

利用立方映射产生混沌分量 $u_{id} = 4u_{(i-1)d}^2 -$

$3u_{(i-1)d}$ ($i = 1, 2, \dots, N; d = 1, 2, \dots, D$),其中 u_{0d} 是 $(-1, 1)$ 中的随机数.将各个分量载入混沌扰动范围内 $[-\beta_d, \beta_d]$ ($d = 1, 2, \dots, D$),混沌扰动量为 $\Delta x_i = (\beta_1 u_{i1}, \beta_2 u_{i2}, \dots, \beta_D u_{iD})$,加入扰动后位置更新为 $x_{id}^b(k+1) = x_{id}(k) + v_{id}(k+1) + \Delta x_{id}$.比较加入混沌扰动位置更新后的适应度值 $f(x_i^b(k+1))$ 和不含扰动位置更新后的适应度值 $f(x_i(k+1))$ 的优劣,如果 $f(x_i^b(k+1))$ 优于 $f(x_i(k+1))$,则将粒子的位置 $x_i(k+1)$ 更新为 $x_i^b(k+1)$.

Step 6 检验算法是否达到最大迭代次数或满足规定的误差等终止条件,若满足则停止迭代,输出求解的最优适应度值及对应的参数,并计算得到离散点之间的气动参数函数值,否则令 $k = k + 1$,转Step 2.

3 仿真分析与验证

为了验证本文提出的AIWCPSO-CSI的有效性,首先通过基准测试函数对AIWCPSO算法的性能进行测试,并同PSO算法和SPSO算法的性能进行比较,分析基准测试函数优化时的全局优化能力以及寻优稳定性;然后以某型飞机气动参数阻力系数与迎角之间的关系为样本,通过仿真实验来验证AIWCPSO-CSI可以解决气动参数插值的问题.

3.1 算法仿真测试

选取两个典型基准测试函数分别为Sphere函数(f_1)和Ackley函数(f_2).其中: f_1 为非线性的对称单峰函数,用来测试算法的精度; f_2 为大量局部最优点的连续、旋转、不可分离的多峰函数,用来检验算法跳出局部最优的能力. f_1 和 f_2 的理论最优值均为0,搜索维数均为30.

Sphere函数(f_1)表达式为

$$f_1 = \sum_{i=1}^{30} x_i^2, \quad x_i \in [-100, 100]; \quad (23)$$

Ackley函数(f_2)表达式为

$$f_2 = 20 + e - 20 \exp \left(-0.2 \sqrt{\frac{1}{30} \sum_{i=1}^{30} x_i^2} \right) - \exp \left(\frac{1}{30} \sum_{i=1}^{30} \cos(2\pi x_i) \right), \quad x_i \in [-32, 32]. \quad (24)$$

仿真实验中优化算法参数设置为:PSO算法和SPSO算法中的最大速度 $v_{\max} = 0.3$,SPSO算法中的最大惯性权重值 $w_{\max} = 0.9$,最小惯性权重值 $w_{\min} = 0.4$, $c_1 = c_2 = 2$.AIWCPSO算法中取 $w_{\min} = 0.5$, $w_{\max} = 0.9$,速度范围为 $v_{\max} = 0.15(x_{\max} - x_{\min})$,其中 x_{\max} 、 x_{\min} 为每一维参数的变化范围, $c_1 = c_2 = 1.49618$, $M = 30000$, $\alpha = 0.4$, $\beta = 0.05$,设定阈值 $\varepsilon = 0.05$,混沌扰动范围参数 $\beta_j = 0.1(b_i - a_i)$.

在 AIWCPSO 算法仿真性能测试中, 粒子个数分别选取 $N=20、30、50$, 对 f_1 和 f_2 寻优迭代 1000 次, 并对 f_1 和 f_2 寻优各进行 50 次独立实验, 将 50 次独立寻优实验中得到的最优适应度值的最好值、最差值、平均值以及方差分别记为 f_{bestg} 、 f_{bestb} 、 f_{avg} 、 Var . 3 种优化算法在 f_1 和 f_2 寻优的实验结果如表 1 和表 2 所示.

表 1 3 种算法在 f_1 上的寻优测试结果

优化算法	N	f_{bestg}	f_{bestb}	f_{avg}	Var
PSO	20	0.271 6	0.469 2	0.373 4	0.001 9
	30	0.241 0	0.413 4	0.344 1	0.001 9
	50	0.203 0	0.384 1	0.313 4	0.001 3
SPSO	20	3.431 7e-8	5.671 8	0.226 4	1.254 9
	30	9.049 2e-10	3.101 1e-5	7.384 2e-7	1.910 3e-11
	50	3.949 7e-10	5.183 8e-8	7.618 7e-9	8.270 5e-17
AIWCPSO	20	2.942 3e-10	0.164 0	0.003 5	5.367 3e-4
	30	5.395 9e-21	6.361 0e-12	2.213 1e-13	8.881 2e-25
	50	7.917 5e-24	2.854 3e-15	1.529 8e-16	2.427 3e-31

表 2 3 种算法在 f_2 上的寻优测试结果

优化算法	N	f_{bestg}	f_{bestb}	f_{avg}	Var
PSO	20	2.962 3	10.866 9	7.402 3	3.336 1
	30	0.807 1	9.903 9	4.914 1	4.460 7
	50	0.648 4	5.946 5	3.383 9	2.551 8
SPSO	20	5.790 1	11.080 1	8.469 7	1.588 6
	30	4.076 1	8.024 3	6.021 5	1.345 7
	50	2.316 8	6.670 3	4.553 3	1.036 9
AIWCPSO	20	2.048 4e-4	2.660 2	1.509 5	0.647 6
	30	1.281 2e-8	2.120 1	1.022 8	0.528 4
	50	1.024 1e-12	1.646 2	0.522 3	0.444 0

由表 1 和表 2 的测试结果可以看出, AIWCPSO 算法寻优的最优值、最差值、平均值以及方差均明显优于 PSO 算法和 SPSO 算法. 因此, AIWCPSO 算法具有较快的收敛速度和较高的收敛精度, 能有效地避免陷入早熟收敛的问题.

3.2 气动参数插值仿真测试

选取迎角值范围为 $[-8^\circ, 8^\circ]$ 上 17 个整数点的样本数据, 将数据按自变量取值的奇偶性分为 A、B 两组. 在仿真实验中将 B 组数据作为插值的原始数据, A、B 两组数据均为测试数据, 在 Matlab 环境下分别采用 LI、LAI、CSI 和 AIWCPSO-CSI 进行气动参数数据解算, 对每种插值方法各进行 1000 次独立实验. AIWCPSO-CSI 的参数设置为: 粒子个数 $N=50$, 搜索维数 $D=9$, $K_{max}=1000$, $w_{min}=0.5$, $w_{max}=0.9$, $c_1=c_2=1.49618$, $v_{max}=0.15(x_{max}-x_{min})$, $M=3000$, $\alpha=0.4$, $\beta=0.05$. 根据调试经验, 取三次样条函数中待优化参数的变化范围为 $[-50, 50]$, 阈值 $\varepsilon=0.05$, 混沌扰动范围参数 $\beta_j=0.1(b_i-a_i)$. 气动参数与迎角经过多种插值方法计算得到的关系曲线如图 1 所示.

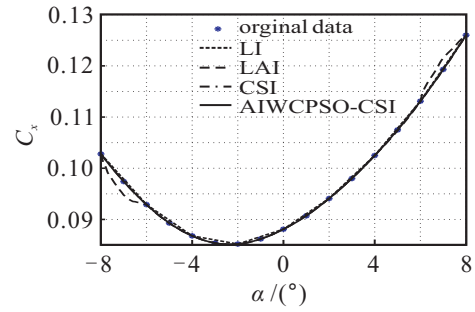


图 1 不同的气动参数插值方法比较曲线

由图 1 可以看出, 迎角值在 $[-6^\circ, 2^\circ]$ 之间时, 迎角 α 与阻力系数 C_x 之间有明显的非线性关系, 此时 LI 只利用相邻点的气动数据来计算未知点的气动数据, 计算精度难以保证, LAI 的数据不稳定, 在样本区间边界处进行插值计算时出现了龙格现象; CSI 和 AIWCPSO-CSI 的插值结果相对较好.

为了验证算法的实时性, 通过 1000 次独立实验计算结果的比较, 分析 4 种插值方法的实时性和鲁棒性, 其计算时间和插值精度如表 3 所示.

表 3 4 种插值算法的计算时间和插值精度对比

优化算法	平均计算时间/ms	平均绝对误差	最大绝对误差
LI	0.026	4.3e-4	6.0e-4
LAI	0.273	9.7e-4	2.7e-3
CSI	0.089	3.1e-5	7.5e-5
AIWCPSO-CSI	0.061	2.9e-5	6.3e-5

4 结 论

针对飞行仿真动力学气动建模过程中气动参数存在非线性关系, 本文采用优化算法的思想对三次样条插值函数的系数寻优, 从粒子群混沌初始化、自适应惯性权重、混沌扰动等方面对 PSO 算法进行改进, 在 AIWCPSO 算法和 CSI 的基础上, 提出了一种基于 AIWCPSO 算法的三次样条气动参数插值方法 (AIWCPSO-CSI). 仿真实验表明, AIWCPSO-CSI 利用 AIWCPSO 算法对三次样条函数系数进行寻优, 不仅克服了传统 PSO 算法易陷入局部极值的缺点, 而且避免了 LAI 的龙格现象和 CSI 求解矩阵的繁琐, 为气动参数插值计算提供了一种新的方法.

参考文献(References)

- [1] 董中翔, 王晓东. 飞行仿真技术的发展与展望[J]. 飞行力学, 2002, 20(3): 5-8.
(Tong Z X, Wang X D. Development and prospect of flight simulation technology[J]. Flight Dynamics, 2002, 20(3): 5-8.)
- [2] 李鹏. 轻型飞行模拟器飞行仿真建模研究[D]. 南京: 南京航空航天大学民航学院, 2008.
(Li P. Research on modeling of flight simulation about prototype of light weight flight simulator[D].

- Nanjing: College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, 2008.)
- [3] Brian L Stevens, Frank L Lewis. Aircraft control and simulation[M]. Hoboken: John Wiley and Sons, 2003: 61-80.
- [4] Kennedy J, Eberhart R C. Particle swarm optimization[C]. Proc of the IEEE Int Conf on Neural Networks. Perth, 1995: 1942-1948.
- [5] Shi Y, Eberhart R C. A modified particle swarm optimizer[C]. Proc of the IEEE Int Conf on Evolutionary Computation. Anchorage, 1998: 69-73.
- [6] Schutte J F, Koh B I, Reinbolt J A, et al. Evaluation of a particle swarm algorithm for biomechanical optimization[J]. J of Biomechanical Engineering, 2005, 127(3): 465-474.
- [7] Luitel B, Venayagamoorthy G K. Quantum inspired PSO for the optimization of simultaneous recurrent neural networks as MIMO learning systems[J]. Neural Networks, 2010, 23(5): 583-586.
- [8] Liu H B, Abraham A, Hassanien A E. Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm[J]. Future Generation Computer Systems, 2010, 26(8): 1336-1343.
- [9] 张选平, 杜玉平, 秦国强, 等. 一种动态改变惯性权的自适应粒子群算法[J]. 西安交通大学学报, 2005, 39(10): 1039-1042.
(Zhang X P, Du Y P, Qin G Q, et al. Adaptive particle swarm algorithm with dynamically changing inertia weight[J]. J of Xi'an Jiaotong University, 2005, 39(10): 1039-1042.)
- [10] Yang X M, Yuan J S, Yuan J Y, et al. A modified particle swarm optimizer with dynamic adaptation[J]. Applied Mathematics and Computation, 2007, 189(2): 1205-1213.
- [11] Nickabadi A, Ebadzadeh M M, Safabakhsh R. A novel particle swarm optimization algorithm with adaptive inertia weight[J]. Applied Soft Computing, 2011, 11(4): 3658-3670.
- [12] 李信真, 车刚明, 欧阳洁, 等. 计算方法[M]. 西安: 西北工业大学出版社, 2006: 116-127.
(Li X Z, Che G M, Ouyang J, et al. Calculate method[M]. Xi'an: Northwestern Polytechnical University Press, 2006: 116-127.)
- [13] 唐巍. 基于幂函数载波的混沌优化方法及其应用[J]. 控制与决策, 2005, 20(9): 1043-1046.
(Tang W. Chaotic optimization method based on power function carrier and its applications[J]. Control and Decision, 2005, 20(9): 1043-1046.)
- [14] 程志刚, 张立庆, 李小林, 等. 基于 Tent 映射的混沌混合粒子群优化算法[J]. 系统工程与电子技术, 2007, 29(1): 103-106.
(Cheng Z G, Zhang L Q, Li X L, et al. Chaotic hybrid particle swarm optimization algorithm based on Tent map[J]. Systems Engineering and Electronics, 2007, 29(1): 103-106.)
- [15] 张浩, 张铁男, 沈继红, 等. Tent 混沌粒子群算法及其在结构优化决策中的应用[J]. 控制与决策, 2008, 23(8): 857-862.
(Zhang H, Zhang T N, Shen J H, et al. Research on decision-makings of structure optimization based on improved Tent PSO[J]. Control and Decision, 2008, 23(8): 857-862.)
- [16] 桂志志. 混沌序列在优化理论中的应用[D]. 南京: 南京理工大学自动化系, 2006.
(Gui C Z. Application of chaotic sequences in optimization theory[D]. Nanjing: Department of Automation, Nanjing University of Science and Technology, 2006.)
- [17] Clerc M. Initialisations for particle swarm optimization [EB/OL]. (2008-12-24). <http://clerc.maurice.free.fr/psol/>.
- [18] Van Den Bergh F, Engelbrecht A P. A study of particle swarm optimization particle trajectories[J]. Information Sciences, 2006, 176(8): 937-971.
- [19] 刘洪波, 王秀坤, 谭国真. 粒子群优化算法的收敛性分析及其混沌改进算法[J]. 控制与决策, 2006, 21(6): 636-640.
(Liu H B, Wang X K, Tan G Z. Convergence analysis of particle swarm optimization and its improved algorithm based on chaos[J]. Control and Decision, 2006, 21(6): 636-640.)
- [20] 吕振肃, 侯志荣. 自适应变异的粒子群优化算法[J]. 电子学报, 2004, 32(3): 416-420.
(Lv Z S, Hou Z R. Particle swarm optimization with adaptive mutation[J]. Acta Electronica Sinica, 2004, 32(3): 416-420.)

(责任编辑: 李君玲)