

# 基于并行交换的增强粒子群优化算法在气动优化中的应用

王培君, 夏露, 周文硕, 栾伟达

(西北工业大学 航空学院, 陕西 西安 710072)

**摘要:**粒子群优化(PSO)算法易于实现,对优化问题可以获得质量较高的解,被广泛应用在如气动优化这种非线性高难度问题中,但是面对多峰问题容易陷入局部最优,存在鲁棒性较差的问题,为了提高PSO的鲁棒性,提出了基于并行交换的增强粒子群优化算法(EP SOBPE)。该算法通过布谷鸟搜索算法(CSA)和PSO种群并行进化,分层交换操作和增强学习策略来增强算法寻优能力与鲁棒性。该算法兼具了CSA的全局搜索能力和PSO的局部能力,使得新算法具有极强的鲁棒性。函数测试表明,新算法相较于其他智能优化算法有更强的鲁棒性,对不同问题的适应能力更强。将EP SOBPE算法应用到RAE2822翼型和M6机翼的气动优化设计中,相较于其他算法可以得到更好的效果,从而表明新算法有鲁棒性,同时兼具了更好的寻优能力。

**关键词:**粒子群优化算法;布谷鸟搜索算法;气动优化设计;全局优化

**中图分类号:**V221.3

**文献标志码:**A

**文章编号:**1000-2758(2022)03-0493-11

随着计算流体力学(CFD)技术的不断发展,气动设计中CFD的占比逐渐提升,数值优化方法已成为飞行器气动外形设计中的一个重要组成部分。在各类优化方法中,启发式算法表现出了十分强大的寻优能力。目前较常用的有以粒子群优化(particle swarm optimization, PSO)算法<sup>[1]</sup>、蚁群算法(ant colony optimization, ACO)、布谷鸟搜索算法(cuckoo search algorithm, CSA)<sup>[2]</sup>为代表的群集智能算法,和以差分进化算法(differential evolution, DE)<sup>[3]</sup>、遗传算法(genetic algorithms, GA)为代表的进化算法。其中PSO算法简单方便、收敛速度较快,对许多问题都有较好的优化效果,但是其更新公式会导致粒子被吸引在当前全局最优解附近,搜索空间被大幅压缩,分散度下降从而导致算法陷入局部最优<sup>[4]</sup>。CSA算法是一种全局寻优能力较强的算法,较强的搜索随机性极大提升了该算法的全局搜索能力,但是CSA也存在收敛速度慢的问题。DE算法主要通过差分变异操作来生成新的变异个体,再通过交叉

和选择得到新的种群个体,贪婪选择策略使其有较强的局部搜索能力,但也容易陷入局部最优。由此可见,有的算法不擅长全局搜索,容易陷入局部最优;有的算法不擅长局部搜索,搜索效率低,在单峰问题中表现差。所以如何补足短板,使算法适应更多形式问题,增强算法的鲁棒性是一个难题。

为了提高启发式算法的鲁棒性,已经有了很多改进措施,比如借鉴其他算法优秀思想对算法更新公式进行改进或在一个种群内通过一定策略使用多种更新公式;对种群拓扑结构进行改进<sup>[5]</sup>,改善种群多样性;还有通过多个种群并行,再利用某种信息交换机制来提高全局搜索能力。其中对算法更新公式进行改进和单一种群内使用多个更新公式的方法改善了原算法搜索模式,但是其搜索模式仍有很大相似度,原算法的短板无法得到有效提升。种群拓扑方法提高了算法全局性能,但是由于算法更新公式并未改变,所以鲁棒性提高不显著。而多种群并行中的分层交换<sup>[6]</sup>的方法将2种不同算法有机结合

收稿日期:2021-06-18

基金项目:翼型、叶栅空气动力学重点实验室基金(6142201200106)资助

作者简介:王培君(1998—),西北工业大学硕士研究生,主要从事飞行器设计研究。

通信作者:夏露(1977—),女,西北工业大学副教授,主要从事飞行器设计研究。e-mail: xialu@nwpu.edu.cn

起来,共享最优信息,互相引导,取长补短,可以较好改进全局搜索能力。李丁等<sup>[6]</sup>通过分层交换操作将 PSO 和 DE 这 2 种容易陷入局部最优的算法结合起来得到了 RADPSO 算法,新算法鲁棒性得到了较大提升,体现了分层交换操作的有效性。但是限于其使用的 PSO 和 DE 算法在很多情况下无法得到较优解,所以该算法在有些多峰问题中仍会陷入局部最优,其鲁棒性仍有较大提升空间。

为了提高 PSO 算法鲁棒性,本文选择将 PSO 种群与其他算法种群进行并行,然后进行分层交互操作,其中选择什么算法种群进行并行至关重要。科研人员通常借鉴全局性较好的算法对粒子群优化算法进行改进。石季英等<sup>[7]</sup>构建了 PSO-CS 混合算法,其核心是将 PSO 算法每轮迭代后个体历史最优位置进行 CSA 算法位置更新,新算法全局搜索能力虽提升有限,但是说明了通过 CSA 算法来提升 PSO 算法全局寻优能力是可行的。鉴于此,如果通过使用分层交换操作来引入 CSA 算法,那就可以通过改善算法全局寻优能力来达到高鲁棒性。故本文提出了基于并行交换的增强粒子群优化算法(enhanced particle swarm optimization based on parallel exchange, EPSOBPE)。该算法首先借鉴差分进化的变异行为对粒子群优化算法进行增强,在不减少全局收敛性的基础上加强快速收敛能力。然后通过让增强后的粒子群与布谷鸟种群进行并行运行,每次迭代后进行分层交换,使二者种群可以有机结合起来,共享最优信息的同时增加种群多样性,以达到增强算法鲁棒性的目的。

## 1 优化算法

### 1.1 粒子群优化算法

粒子群优化算法<sup>[1]</sup>是通过群体中的个体互相协助和群体最优信息共享来探索最优解的。粒子群中每个粒子都有一个初始速度和适应度值,个体在迭代过程中根据自身经历过的最佳位置以及种群得到的最优解的位置来生成速度,再根据目前位置和生成的速度来得到新的在空间中的位置<sup>[1]</sup>。粒子群优化算法的数学描述如(1)式所示

$$v_{ij}^{t+1} = \omega v_{ij}^t + c_1 r_1 (p_{ij}^t - x_{ij}^t) + c_2 r_2 (g_j^t - x_{ij}^t) \quad (1)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (2)$$

式中,下标  $i$  和  $j$  分别代表的是第  $i$  个粒子与维度的

第  $j$  维,上标  $t$  代表迭代代数。 $\omega$  为惯性权重因子; $c_1, c_2$  为学习因子; $r_1$  和  $r_2$  是介于  $[0, 1]$  之间的随机数。

粒子群算法综合能力较强,但因为更新公式导致粒子被限制在当前全局最优解附近,在很多情况下搜索空间会被压缩,使得搜索过早停滞,最终导致算法在大部分情况下无法获得较好的结果。而布谷鸟算法凭借莱维飞行这种长短步相间的操作使其不会出现搜索空间被压缩的情况,有较强的全局搜索能力。

### 1.2 布谷鸟搜索算法

布谷鸟搜索算法(CSA)<sup>[2]</sup>中有 2 种更新位置的方式,一种是布谷鸟寻找鸟窝下蛋的寻找路径采用了莱维飞行。莱维飞行的主要特点是长短步相间,采用莱维飞行更新位置的公式如下

$$X^{t+1} = X^t + \alpha \otimes \text{Levy}(\beta) \quad (3)$$

式中: $\alpha$  是步长缩放因子; $\text{Levy}(\beta)$  是莱维随机路径; $\otimes$  是点乘运算。其中生成服从莱维随机步长的公式使用 Mantegna 方法,公式如(4)式所示

$$\text{Levy}(\beta) = \frac{u}{|v|^{\frac{1}{\beta}}} \quad (4)$$

式中,  $u \sim N(0, \sigma^2)$ ,  $v \sim N(0, 1)$ , 其中  $\sigma$  如下所示

$$\sigma = \left\{ \frac{\Gamma(1 + \beta) \sin\left(\frac{\pi\beta}{2}\right)}{\beta \Gamma\left(\frac{1 + \beta}{2}\right) 2^{\frac{\beta-1}{2}}} \right\}^{\frac{1}{\beta}} \quad (5)$$

式中: $\beta \in [1, 3]$ , 本文  $\beta$  取 1.5; $\alpha$  是步长缩放因子,在布谷鸟算法中针对  $\alpha$  有不同的变种,本文中  $\alpha$  取值如(6)式所示

$$\alpha = \frac{(X_i - X_j)}{\sqrt{D}} \quad (6)$$

式中: $X_i$  与  $X_j$  是任意不同鸟窝的位置; $D$  为问题自变量维数。

另一种位置更新方式是宿主鸟以一定概率  $P_a$  发现外来鸟后重新建窝的位置路径,这个路径可以采用莱维飞行或偏好随机游动方式,本文使用偏好随机游动方式,更新位置公式如(7)式所示

$$X^{t+1} = X^t + r \otimes \text{Heaviside}(P_a - \delta) \otimes (X_i - X_j) \quad (7)$$

式中: $r, \delta$  是服从均匀分布的随机数; $\text{Heaviside}$  是跳跃函数( $x > 0, y = 1; x < 0, y = 0$ )。

布谷鸟搜索算法通过莱维飞行可以较大程度上

保证样本的多样性,但是也导致其局部搜索能力相对于 PSO 和 DE 较差,收敛速度慢,鲁棒性也较差。

### 1.3 基于并行交换的增强粒子群优化算法

PSO 算法前期收敛速度快,后期种群多样性的消失使其容易陷入局部最优,而 CSA 算法的莱维飞行使其全局搜索能力较强,但收敛速度慢,缺少活力。二者鲁棒性均较差,但是各有优点,为了获得一个鲁棒性较强并兼具二者优点的算法,提出了基于并行交换的增强粒子群优化算法(EP SOBPE)。

在 EP SOBPE 中,由于 CSA 种群分走了一部分粒子数量,需要对粒子种群进行增强,在保证全局搜索能力的前提下加速收敛,所以利用差分进化中的突变操作对粒子群全局最优解进行学习。

$$x_i^{t+1} = g^t + F(t)(x_{r_1}^t + x_{r_2}^t - x_{r_3}^t - x_{r_4}^t) \quad (8)$$

式中,  $r_1, r_2, r_3, r_4 \in [1, N]$  的整数,  $N$  为粒子数量,且  $r_1 \neq r_2 \neq r_3 \neq r_4, g$  为粒子群全局最优解,  $F(t)$  为比例因子。

为了使算法在前期具有一定的全局搜索能力避免快速收敛陷入局部最优,所以前期选择开关  $T$  值应稍小,  $F$  应稍大,使得早期使用增强学习的粒子数量较少,使用增强学习的粒子得到的扰动更大,后期使用增强学习策略的粒子较多,使用增强学习的粒子得到的扰动较小,加速收敛,得到最优解。  $T$  与  $F$  更新公式如(9)~(10)式所示

$$T(t) = T_{\min} + \frac{(T_{\max} - T_{\min})(t - 1)}{G_{\max} - 1} \quad (9)$$

$$F(t) = F_{\max} - \frac{(F_{\max} - F_{\min})(t - 1)}{G_{\max} - 1} \quad (10)$$

式中,  $G_{\max}$  为最大迭代轮数。

采用增强学习策略,既可保证算法收敛,又可防止算法早熟。优化前期,粒子之间距离较远,差分变化较大,引入的扰动更强,粒子不会过早地聚集在  $g$  周边,粒子分散度大,探索能力强,对空间搜索更为充分;优化后期,粒子相互靠近,差分变化减小,粒子向  $g$  聚集,加速收敛。增强学习策略流程图如图 1 所示。  $R$  表示均匀分布在  $[0, 1]$  区间的随机数。

由于在气动优化问题中,最优解一般位于部分维度的边界上,所以有必要对边界处理进行改进,故在边界处理中采用了减幅的处理方法,即当粒子位置超出了边界后,粒子的位置仍然重置于边界  $x$  上,但  $x$  方向的速度以一个随机大小相反方向返回到边界内,如图 2 所示,  $r$  表示均匀分布在  $[0, 1]$  区间的随机数。

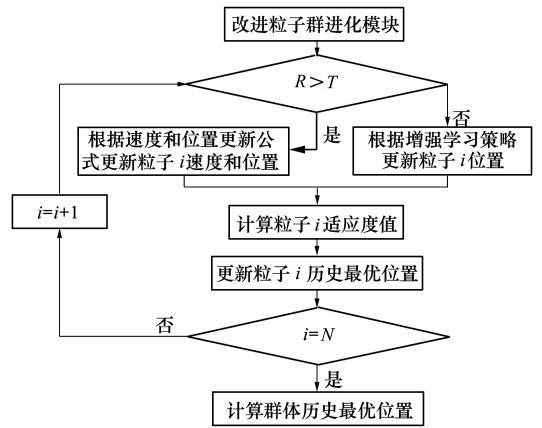


图 1 增强学习策略流程图

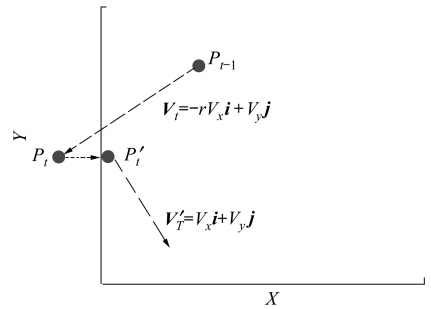


图 2 速度减幅边界处理

EP SOBPE 算法策略是一种种群并行策略,2 种算法分别使用改进后的 PSO 和 CSA 算法。改进后 PSO 算法虽然通过增强学习策略增强了局部搜索能力和收敛速度,但是在很多情况下仍不可避免陷入局部最优。CSA 算法有着较强的全局搜索能力,但是收敛能力较差,所以通过分层交换策略将 2 种算法取长补短,共享二者优势信息使算法具有较强鲁棒性,而且既可以快速收敛,也可以保证其全局搜索能力。新算法没有通过只改进更新公式或改进种群拓扑结构来增强算法鲁棒性,而是使用多种群并行与分层交换机制。这点与其他改进算法有较大差别,有效提高了新算法的鲁棒性,使新算法在面对不同类型问题时可以获得更好的解。

EP SOBPE 算法的具体步骤如下:

- 1) 对算法进行初始化,给定相关参数。
- 2) 将整个大种群等分成 2 个种群,分别运行改进 PSO 算法和 CSA 算法。
- 3) 在一轮后得到 2 个种群的函数最优值后进行比较,选出最优者。
- 4) 更新种群的最优值并进行替换,用最优者替换改进 PSO 种群中的最优者,或 CSA 种群中任意一

个个体。

5) 对改进 PSO 种群进行一次分层,即求出改进 PSO 群体中所有个体的平均适应度值  $F_{PSO1}$ ;对 CSA 种群进行一次分层,即求出 CSA 群体中所有个体的平均适应度值  $F_{CSA1}$ 。

6) 对改进 PSO 种群进行二次分层,即求出改进 PSO 群体中适应度值优于  $F_{PSO1}$  的所有个体的平均适应度值  $F_{PSO2}$ ;对 CSA 种群进行二次分层,即求出 CSA 群体中适应度值优于  $F_{CSA1}$  的所有个体的平均适应度值  $F_{CSA2}$ 。

7) 判断  $F_{PSO2}$  是否优于  $F_{CSA2}$ , 若满足,用 PSO 群体中适应度值优于  $F_{PSO2}$  的个体替换 CSA 群体中适应度值较差的那些个体;若不满足,用 CSA 群体中适应度值优于  $F_{CSA2}$  的个体替换 PSO 群体中适应度值较差的那些个体。

8) 判断是否满足停止准则,若满足,则结束,若不满足,则转至步骤 2)。

本文在 PSO 算法的基础上首先进行了增强行为,增强了 PSO 算法的局部搜索能力;同时修改了边界处理方法。然后通过分层交换策略引入 CSA 算法,来增强 PSO 算法的全局寻优能力,最终获得了具有强鲁棒性的 EPSOBPE 算法。

## 2 函数测试

为了测试 EPSOBPE 算法性能,本文对 PSO 算法,DE 算法,CSA 算法,EPSOBPE 算法和 RADPSO 算法<sup>[6]</sup>进行比较,测试函数<sup>[8]</sup>如表 1 所示,函数具体设置可参考文献<sup>[8]</sup>。

表 1 测试函数

函数类型	函数表达式
单峰函数	$f_1(x) = \sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)^2$
单峰函数	$f_2(x) = \max\{ x_i , 1 \leq i \leq n\}$
单峰函数	$f_3(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
多峰函数	$f_4(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$
多峰函数	$f_5(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
多峰函数	$f_6(x) = \left( \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$
多峰函数	$f_7(x) = - \sum_{j=1}^4 c_j \exp\left(- \sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2\right)$
多峰函数	$f_8(x) = - \sum_{j=1}^5 [(X - a_j)(X - a_j)^T + c_j]^{-1}$

由于测试函数维度不同,故不同维度算法种群个数也不同,具体设置如表 2 所示。

表 2 算法设置

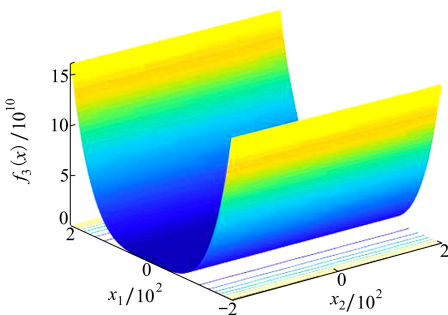
自变量维度	优化算法种群个体数 N
2	30
3	30
4	40
30	60

在函数测试中,每个算法计算 30 次,结果取 30 次平均值。PSO 算法中  $C_1 = C_2 = 2$ ,惯性权重因子  $\omega$  随进化代数由 0.9 线性递减到 0.4<sup>[9]</sup>。DE 算法交叉因子  $C_R = 0.7$ ,缩放因子  $F = 0.5$ 。RADPSO 算法参数取上述 PSO 算法和 DE 算法的参数。CSA 算法中  $P_a = 0.25$ 。EPSOBPE 算法的设置同上述 PSO 算法和 CSA 算法具有相同设置,缩放因子  $F$  从 0.7 线性递减至 0.3,  $T$  从 0.3 线性递增至 0.7。每次寻优中算法收敛条件为调用函数次数达到  $100 \times D \times N$ 。表 3 展示了不同算法 30 次运行结果的均值和标准差。

表3 函数测试的算法性能

测试函数	结果	PSO	DE	CSA	RADPSO	EPSOBPE	函数最小值
$f_1(x)$	Mean	1.17	$2.21 \times 10^3$	$2.53 \times 10^2$	$3.14 \times 10^{-1}$	<b><math>1.06 \times 10^{-1}</math></b>	0
	Std	$8.02 \times 10^{-1}$	$1.19 \times 10^3$	$6.11 \times 10^1$	$3.85 \times 10^{-1}$	<b><math>7.55 \times 10^{-2}</math></b>	
$f_2(x)$	Mean	$2.37 \times 10^{-1}$	$1.74 \times 10^{-2}$	$9.93 \times 10^{-1}$	$6.89 \times 10^{-1}$	<b><math>1.50 \times 10^{-2}</math></b>	0
	Std	$1.06 \times 10^{-1}$	$9.27 \times 10^{-2}$	$1.16 \times 10^{-1}$	$3.08 \times 10^{-1}$	<b><math>6.48 \times 10^{-3}</math></b>	
$f_3(x)$	Mean	$3.58 \times 10^1$	<b><math>1.35 \times 10^1</math></b>	$2.67 \times 10^1$	$3.18 \times 10^1$	$1.84 \times 10^1$	0
	Std	$2.63 \times 10^1$	2.34	1.34	$1.93 \times 10^1$	<b><math>9.98 \times 10^{-1}</math></b>	
$f_4(x)$	Mean	$-6.64 \times 10^3$	$-7.08 \times 10^3$	$-8.71 \times 10^3$	$-9.76 \times 10^3$	<b><math>-1.05 \times 10^4</math></b>	-12 509.7
	Std	$6.37 \times 10^2$	$4.46 \times 10^2$	<b><math>2.66 \times 10^2</math></b>	$7.17 \times 10^2$	$8.52 \times 10^2$	
$f_5(x)$	Mean	$1.33 \times 10^{-2}$	$4.93 \times 10^{-4}$	$1.19 \times 10^{-3}$	$2.38 \times 10^{-3}$	<b><math>1.39 \times 10^{-12}</math></b>	0
	Std	$1.10 \times 10^{-2}$	$1.88 \times 10^{-3}$	$1.43 \times 10^{-3}$	$5.62 \times 10^{-3}$	<b><math>4.72 \times 10^{-12}</math></b>	
$f_6(x)$	Mean	3.30	<b><math>9.98 \times 10^{-1}</math></b>	$9.98 \times 10^{-1}$	1.99	<b><math>9.98 \times 10^{-1}</math></b>	1
	Std	2.73	<b>0</b>	$5.09 \times 10^{-4}$	1.78	$2.66 \times 10^{-14}$	
$f_7(x)$	Mean	-3.862 8	-3.862 8	-3.862 8	-3.862 8	<b>-3.860 0</b>	-3.86
	Std	$2.45 \times 10^{-15}$	$2.71 \times 10^{-15}$	$9.79 \times 10^{-10}$	$2.64 \times 10^{-15}$	<b><math>2.32 \times 10^{-15}</math></b>	
$f_8(x)$	Mean	-5.897	<b>-10.153</b>	-10.151	-6.569	<b>-10.153</b>	-10.153 2
	Std	3.42	<b><math>6.90 \times 10^{-15}</math></b>	$3.53 \times 10^{-3}$	3.51	$1.48 \times 10^{-10}$	

由表3可见, EPSOBPE算法在3个单峰函数中的寻优效果并没有因为执行粒子群优化算法的个体数减少而变差,在单峰函数中得到的结果都较好,体现了对PSO进行增强是十分有效的。但是对于测试函数 $f_3(x)$ , EPSOBPE算法并没有获得最好的寻优结果,这主要是因为测试函数 $f_3(x)$ 具有一定的峡谷特征,如图3所示优化算法很容易在峡谷中陷入停滞。

图3  $2-Df_3(x)$  函数示意图

同时该函数的峡谷特征正好与坐标轴相匹配,此时DE算法的变异操作可以更好地进行搜索, EPSOBPE算法由于加入了增强学习策略,在一定程度上

上增强了此类函数的寻优能力,但是由于并不是所有粒子都会使用增强学习策略,所以EPSOBPE算法在此测试函数的表现上仅次于DE算法。

在之后5个多峰函数中EPSOBPE算法由于使用了增强PSO与CSA并行并进行分层交换,体现出了极强的全局寻优能力,同时,得到的标准差也较低,具有很强的稳定性。由于EPSOBPE算法在更新公式上与PSO和CSA算法较为相似,分层交换操作并不复杂,所以在计算时间方面与其余算法相似。EPSOBPE算法在不同维度不同种类的函数测试均表现良好,这体现出了EPSOBPE算法相较于其他算法的鲁棒性更强,能够适应不同种类的函数,同时具有更好的寻优能力。

### 3 翼型气动优化设计

#### 3.1 翼型CFD算例验证

本文CFD二维算例验证采用经典算例RAE 2822跨音速流场数值模拟,该算例被广泛应用于二维CFD计算校核。该算例选取设计状态为自由来流马赫数 $Ma=0.73$ ,雷诺数 $Re=6.5 \times 10^6$ ,迎角 $\alpha=$

2.85°,设计升力系数  $C_L = 0.8$ 。在 CFD 计算后与试验数据<sup>[10]</sup> 结果进行对比。二维算例的网格量为3万,远场距离翼型表面40倍弦长。本文二维翼型计算时湍流模型选用  $k-\omega$  SST 湍流模型。

图4展示了数值模拟结果与试验值的压力分布对比。从图4可以看出,数值模拟所得压力分布与试验结果基本吻合,能够较为准确地捕捉激波的位置和强度。这说明本文采用的 CFD 数值计算方法在求解二维翼型的气动特性中具有较高的数值模拟精度。

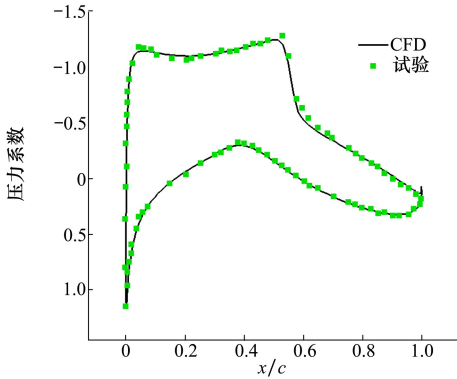


图4 翼型压力分布对比

### 3.2 翼型优化模型

考虑到在工程中,翼型均有后缘厚度,故翼型气动优化设计的初始翼型为修改成钝后缘的RAE2822翼型,设计状态为  $M_\infty = 0.73$ ,  $Re = 6.5 \times 10^6$ 。巡航状态下翼型应有优秀的低阻力系数  $C_D$ 、小低头力矩系数  $C_M$  特性,且升力系数  $C_L$  应基本保持不变,同时还应考虑结构布置和机翼油箱容积等问题。故而优化目标及约束如下

$$\begin{aligned} \min: & F(x) = C_D \\ g_1(x): & C_L = 0.8 \\ g_2(x): & C_M - C_{M0} \geq 0 \\ g_3(x): & \frac{A}{A_0} \geq 1 \\ g_4(x): & \frac{t_{\max}}{t_{\max 0}} \geq 1 \end{aligned} \quad (11)$$

式中:  $A$  为翼型围成的面积;  $t_{\max}$  为翼型最大厚度; 角标 0 为初始翼型相关参数。

翼型参数化使用扰动 CST 方法<sup>[11-12]</sup>,设计变量为12个,翼型上下表面各6个。分别采用 CSA 算法、DE 算法、EPSOBPE 算法、PSO 算法和 RADPSO 算法进行翼型优化设计,各算法种群个体数均为

24,其余参数与函数测试中参数设置相同。

CFD 设置与算例验证相同,采用  $k-\omega$  SST 湍流模型,网格数量为3万。二维翼型优化用计算机 CPU 为 Intel(R) Core(TM) i7-10700F CPU,内存为32 GB。由于算法调用 CFD 求解器次数过多所以使用基于 Kriging 代理模型<sup>[13]</sup> 的优化方法。

本文使用的基于 Kriging 代理模型的优化方法是首先使用拉丁超立方抽样方法(Latin hypercube sampling, LHS)生成少量的样本点构建初始代理模型,然后通过使用优化算法来求解加点带来的优化子问题得到新的样本坐标,将新样本加入样本库后再次构建代理模型,进行代理模型的更新,直到满足收敛条件。二维翼型优化问题中初始样本点数为240,使用混合加点方式<sup>[14]</sup> 加点200个,共440次 CFD 计算。

混合加点方式是先使用最大改善期望(expected improvement, EI) 加点准则,当  $E[I(x)] < 0.01 \min(F(x))$  时使用最小预测值(minimize prediction, MP) 加点准则。

EI 加点准则选取改善期望最大的设计点作为加点位置。设计点  $x$  处的改善值表示为  $I(x) = y_{\min} - y(x)$ ,改善期望为

$$\begin{aligned} E[I(x)] &= \int_0^{+\infty} I(x) \text{pdf}(y) dI(x) = \\ &= \int_{-\infty}^{y_{\min}} (y_{\min} - y) \text{pdf}(y) dy = \\ &= \int_{-\infty}^{y_{\min}} (y_{\min} - y) \frac{1}{\sigma} \varphi\left(\frac{y - \hat{y}}{\sigma}\right) dy \end{aligned} \quad (12)$$

式中,  $\varphi$  为标准正态分布的概率密度函数。对(12)式使用分部积分可得

$$E[I(x)] = \sigma(x) [u\Phi(u) + \varphi(u)] \quad (13)$$

式中,  $u = (y_{\min} - \hat{y})/\sigma$ ,  $\Phi$  为标准正态分布的累积分布函数。

MP 方法直接使用代理模型预测值作为目标值进行寻优,把寻优得到的具有最小预测值的设计点作为加点位置。

### 3.3 优化结果及其分析

图5和图6给出了优化迭代记录及优化前后几何形状与压力系数分布。

表4给出了优化后翼型的性能参数比较。

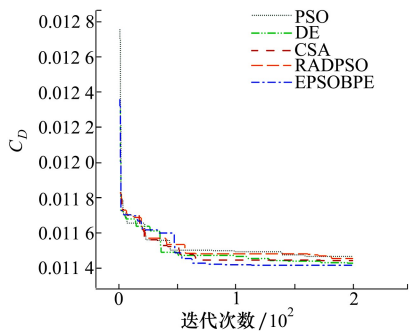
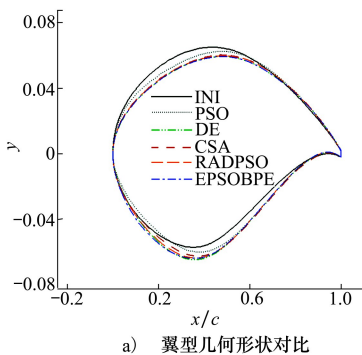
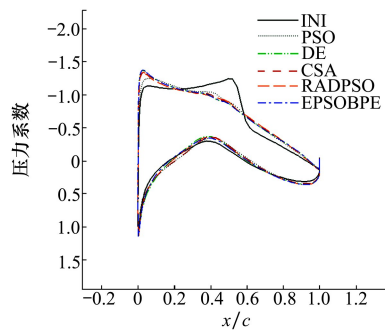


图 5 翼型优化迭代记录



a) 翼型几何形状对比



b) 翼型压力系数对比

表 4 翼型优化结果比较

性能	$C_D$	$C_M$	面积	厚度
初始翼型	0.016 306	-0.102 650	0.077 801	0.120 930
PSO	0.011 465	-0.099 134	0.077 802	0.120 993
DE	0.011 429	-0.097 615	0.077 805	0.121 838
CSA	0.011 442	-0.096 064	0.077 812	0.120 981
RADPSO	0.011 455	-0.100 896	0.077 802	0.121 325
EPSOBPE	0.011 415	-0.100 084	0.077 802	0.120 950

由图 5 可知,几种优化算法的收敛速度类似,但是接近收敛时,EPSOBPE 算法的结果更好,体现了 EPSOBPE 算法有着更好的鲁棒性,所以可以更好地找到每次代理模型中获得最大改善的点或阻力系数最小的点。由图 6 可见,优化后的翼型上表面顶点后移,最高点有所下移,下表面顶点后移,最低点下移量与上表面最高点下移量相似。压力系数方面,优化后的前缘吸力峰峰值有所增大,为了保持升力,翼型上表面前部和后部的压力系数有所减少,中部消除了激波,压力系数过渡更为平滑,但也导致压力系数绝对值减少。翼型下表面压力系数与初始值相似。在优化后的翼型中,经过 PSO 优化的翼型与其他算法优化的翼型相差较大,其阻力系数也在优化后的翼型中最大,说明其陷入了局部最优,体现了 PSO 算法较差的鲁棒性。在其余算法中 EPSOBPE 算法得到的结果在厚度和面积上更接近初始翼型的值,且结果的阻力系数是最小的,说明 EPSOBPE 优化后得到的结果更接近最优解。EPSOBPE 算法正如上文分析的一样,具有极强的鲁棒性,面对不断变换的代理模型所带来的子优化问题,可以获得更好的解。

## 4 机翼气动优化

### 4.1 机翼 CFD 算例验证

本文三维算例采用的是经典三维验证算例, ONERA M6 机翼<sup>[15]</sup>流场的数值模拟算例。该算例被广泛应用于三维 CFD 计算校核。该算例的气动设计状态为  $Ma = 0.839 5$ ,迎角  $\alpha = 3.06^\circ$ ,雷诺数  $Re = 1.172 \times 10^7$ 。该算例的计算网格量为 108 万。本文三维机翼计算时湍流模型选用  $k-\omega$  SST 湍流模型。

图 7 展示了 0.44 倍和 0.9 倍展向位置处截面下的数值模拟结果与试验值的压力分布对比,其中,  $y$  表示截面处位置距离机翼根部的长度,  $b$  表示机翼展长。结果表明数值模拟所得压力分布与试验结果基本吻合,并能够较为准确地捕捉激波的位置和强度。这说明本文采用的 CFD 数值计算方法在求解三维机翼的气动特性中具有较高的数值模拟精度。

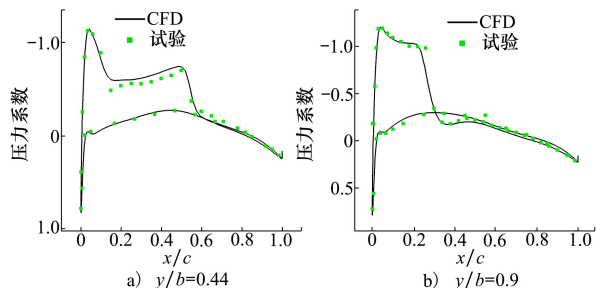


图 7  $y/b=0.44$  与  $y/b=0.9$  截面处的压力分布

### 4.2 机翼优化模型

以 M6 机翼为初始机翼,分别采用 CSA 算法、DE 算法、EPSOBPE 算法、PSO 算法和 RADPSO 算法进行机翼优化设计,各算法种群个体数均为 58,

其余参数与函数测试中参数设置相同。CFD 计算参数设置与算例验证相同,网格与算例验证时相同均为 108 万。三维机翼优化用计算机 CPU 为 2 个 Intel(R) Xeon(R) Gold 6248R,内存为 256 GB。由于三维流场计算时间较长所以使用基于 Kriging 代理模型的优化方法。优化方法流程与二维翼型优化流程相同。三维机翼优化初始样本点数量为 300,采用混合加点方式,加点数量为 300。参数化方法使用自由变形方法(free-form deformation, FFD)<sup>[16]</sup>,本文通过改变翼根与翼梢 2 个控制截面处的控制框点在 Z 方向上的位置实现对外形的改变,每个控制截面翼型上下表面各 7 个控制点,再对翼尖在 XOY 平面方向内设置 1 个扭转角的控制变量,综上,该算例模型共使用 29 个设计变量。

机翼优化设计状态为  $M_\infty = 0.839 5, Re = 11.72 \times 10^6$ 。优化目标为阻力系数  $C_D$  最小化,约束为保持升力系数  $C_L$  不变以及翼根厚度  $t_r$  与翼尖厚度  $t_t$  不减小。可得公式如下

$$\begin{aligned} \min : F(x) &= C_D \\ g_1(x) : C_L &= C_{L0} \\ g_2(x) : \frac{t_r}{t_{r0}} &\geq 1 \\ g_3(x) : \frac{t_t}{t_{t0}} &\geq 1 \end{aligned} \quad (14)$$

### 4.3 优化结果及其分析

图 8 展示了优化过程中样本中最小阻力系数的收敛过程,表 5 给出了 M6 机翼最终的优化结果。

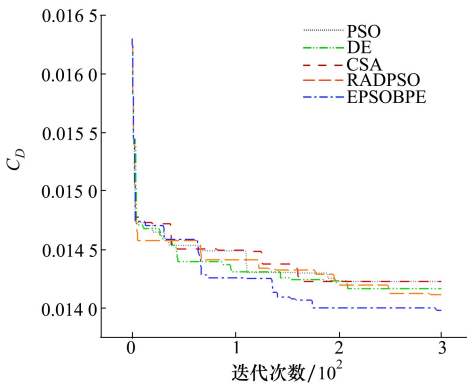


图 8 机翼优化迭代记录

表 5 机翼优化结果对比

性能	阻力系数	翼根厚度	翼尖厚度
初始机翼	0.017 654	0.078 8	0.044 3
PSO	0.014 223	0.079 7	0.044 7
DE	0.014 161	0.079 9	0.044 8
CSA	0.014 227	0.081 1	0.044 9
RADPSO	0.014 112	0.080 0	0.044 8
EPSOBPE	0.013 979	0.079 0	0.045 4

由图 8 可知, EPSOBPE 算法在较高维度的优化中可以适应不断变化的子优化问题,提高了优化收敛速度,体现了 EPSOBPE 算法的鲁棒性较强,最终可以得到与其他算法相比更好的优化结果。由表 5 可知,相同计算量下通过使用 EPSOBPE 算法,阻力系数从 0.017 654 降低到 0.013 979,结果比其他算法中最小值仍少 1.35 个框。EPSOBPE 优化后翼根厚度为 0.079 0,与初始机翼翼根厚度较为相近,说明 EPSOBPE 算法得到的结果更接近全局最优解。

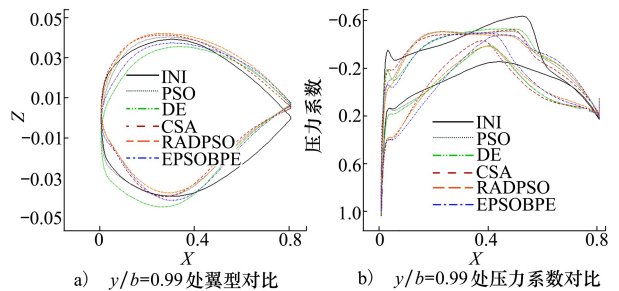


图 9  $y/b=0.99$  处翼型与压力系数对比图

通过图 9 可以看出 EPSOBPE 得到的翼根处翼型的头部前缘半径更小,翼型靠近后缘部分变薄并有向下弯曲的趋势。上表面相较于初始机翼的翼型更加平坦,为了补偿上表面厚度的减少,下表面厚度有所增加。通过图 9 展示的压力系数对比图可以看出各算法的优化结果均减弱了前缘的吸力峰,同时减弱了上表面的激波,其中 EPSOBPE 优化后的上表面压力系数过渡更为平缓,阻力更小。下表面靠近前缘处 PSO 算法和 DE 算法得到的压力系数与初始翼型压力系数相近,其余 3 种算法结果不同程度提高了压力系数。在下表面中部所有算法都有较初始翼型更低的压力系数,其中 EPSOBPE 算法压力系数更小,十分接近上表面同位置的的压力系数。



从图 10 中可以看出不同优化算法得到的翼尖翼型都有不同程度的低头扭转,新的截面形状与原截面形状区别较大。在所有算法中 DE 算法扭转最大,RADPSO 算法扭转最小,优化结果最好的 EPSOBPE 算法扭转程度位于中间位置。

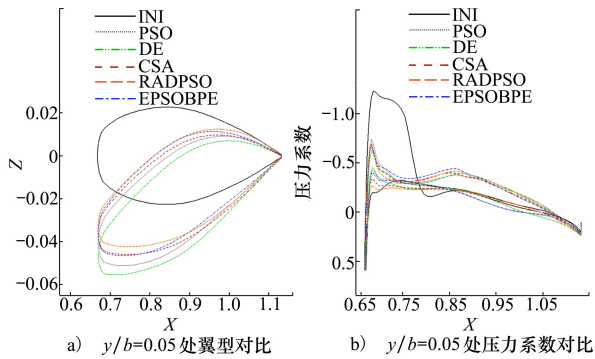


图 10  $y/b=0.05$  处翼型与压力系数对比图

图 10 展示了翼尖截面的压力系数情况,可以发现各算法结果都不同程度地减小了翼尖的载荷,使得原机翼翼尖的激波得到了有效减弱。翼尖截面优化后前缘部分上下截面的压力差变小,截面中后部的压力系数由初始的几乎无压力差变为有压力差,提高了气动效率。

图 11~16 展示了优化前后机翼上表面压力系数云图,可以发现经过优化后各算法都减弱了机翼中部激波,DE 和 EPSOBPE 算法结果前缘吸力的激波较其他算法结果更弱,上表面压力分布更为均匀,其他算法机翼上表面压力分布均形成了 2 个吸力团。DE 算法上表面压力分布虽均匀,但是其机翼中部吸力峰后的压力过渡较快,故其阻力较大。EPSOBPE 的上表面压力分布均匀,机翼中部吸力峰后的压力过渡较 DE 更为平缓,故有更低的阻力。

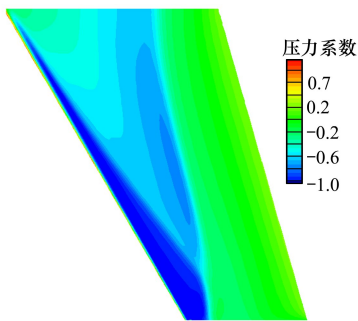


图 11 初始机翼上表面云图

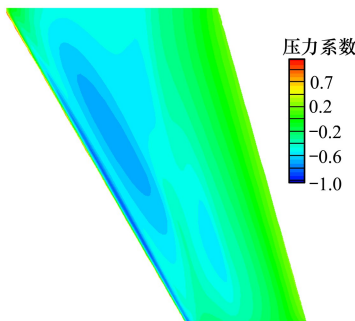


图 12 PSO 优化后机翼上表面压力系数云图

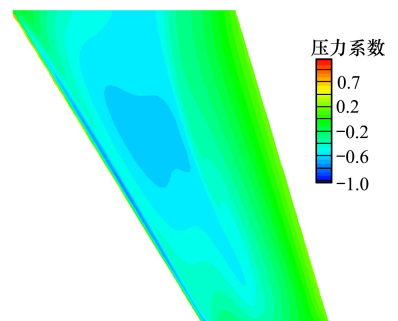


图 13 DE 优化后机翼上表面压力系数云图

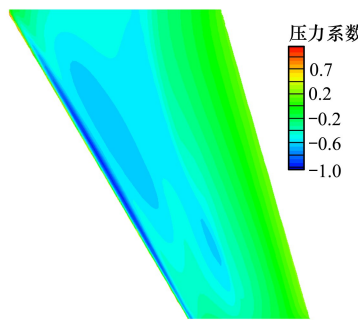


图 14 CSA 优化后机翼上表面压力系数云图

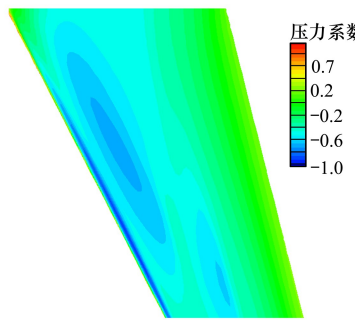


图 15 RADPSO 优化后机翼上表面压力系数云图

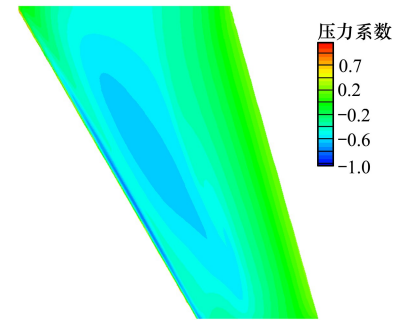


图 16 EPSOBPE 优化后机翼上表面压力系数云图

## 5 结 论

本文针对粒子群优化算法面对多峰问题容易陷入局部最优,存在鲁棒性较差的问题提出了一种基

于并行交换的增强粒子群优化算法。新算法首先使用增强学习策略对粒子群算法进行了改进来增强其局部寻优能力,同时考虑到气动优化的特点,对边界处理进行了改进,得到增强后的粒子群优化算法;然后使增强后的粒子群优化算法和布谷鸟搜索算法并

行,每次迭代后通过分层交换操作来交换各种群信息,有机结合了增强后粒子群优化算法的局部寻优能力和布谷鸟搜索算法的全局寻优能力,得到了EPSOBPE算法。新算法同时兼具了强局部寻优能力和强全局寻优能力,具有了极强的鲁棒性。在函数测试中,EPSOBPE算法与PSO算法等共5种优化算法在8种测试函数上进行试验对比。结果表明EPSOBPE算法在处理低维和高维、单峰和多峰优化测试函数都有突出表现,验证了新算法的优越性和鲁棒性。在二维气动优化算例中各算法在优化前期表现较为相近,但是在优化后期EPSOBPE算法逐

渐与其他算法拉开差距,最终优化效果更好,展现了对气动优化问题有更强适应能力。在三维气动优化算例中,EPSOBPE算法展现出了更强的寻优能力,能够在不断变化的子问题中不断找到全局最优点,最终使机翼阻力系数优化效果明显好于其他算法,进一步验证了EPSOBPE算法的强鲁棒性,可在不同维度和不同种类问题中得到更优解。

目前提升算法鲁棒性、彻底解决算法搜索模式单一的方法除种群并行外仍缺乏其他方法。继续使用种群并行机制,然后发展其他信息交换机制或提出其他解决方法将是今后重要的研究方向。

## 参考文献:

- [1] KENNEDY J, EBERHART R. Particle swarm optimization[C]//Proceedings of International Conference on Neural Networks, 1995: 1942-1948
- [2] YANG X S, DEB S. Cuckoo search via Lévy flights[C]//2009 World Congress on Nature & Biologically Inspired Computing, 2009: 210-214
- [3] STORN R, PRICE K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces[J]. Journal of Global Optimization, 1997, 11(4): 341-359
- [4] CLERC M, KENNEDY J. The particle swarm—explosion, stability, and convergence in a multidimensional complex space[J]. IEEE Trans on Evolutionary Computation, 2002, 6(1): 58-73
- [5] MENDES R, KENNEDY J, NEVES J. The fully informed particle swarm: simpler, maybe better[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 204-210
- [6] 李丁, 夏露. 一种混合粒子群优化算法在翼型设计中的应用[J]. 航空计算技术, 2010, 40(6): 66-71  
LI Ding, XIA Lu. Application of a hybrid particle swarm optimization to airfoil design[J]. Aeronautical Computing Technique, 2010, 40(6): 66-71 (in Chinese)
- [7] 石季英, 巩莹, 薛飞, 等. 计及电动汽车的 PSO-CS 算法配网重构[J]. 电力系统及其自动化学报, 2018, 30(2): 66-70  
SHI Jiying, GONG Ying, XUE Fei, et al. Distribution network reconfiguration with PSO-CS algorithm considering electric vehicles[J]. Proceedings of the CSU-EPSA, 2018, 30(2): 66-70 (in Chinese)
- [8] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69: 46-61
- [9] 胡建秀, 曾建潮. 微粒群算法中惯性权重的调整策略[J]. 计算机工程, 2007(11): 193-195  
HU Jianxiu, ZENG Jianchao. Selection on inertia weight of particle swarm optimization[J]. Computer Engineering, 2007(11): 193-195 (in Chinese)
- [10] COOK P, MCDONALD M, FIRMIN M. Aerofoil RAE-2822: pressure distributions, and boundary layer and wake measurements [R]. AGARD AR-138-A6, 1979
- [11] KULFAN B M. Universal parametric geometry representation method[J]. Journal of aircraft, 2008, 45(1): 142-158
- [12] 卜月鹏, 宋文萍, 韩忠华, 等. 基于 CST 参数化方法的翼型气动优化设计[J]. 西北工业大学学报, 2013, 31(5): 829-836  
BU Yuepeng, SONG Wenping, HAN Zhonghua, et al. Aerodynamic optimization design of airfoil based on CST parameterization method[J]. Journal of Northwestern Polytechnical University, 2013, 31(5): 829-836 (in Chinese)
- [13] SACKS J, WELCH W J, MITCHELL T J, et al. Design and analysis of computer experiments[J]. Statistical Science, 1989, 4(4): 409-423
- [14] 王超, 高正红, 张伟, 等. 自适应设计空间扩展的高效代理模型气动优化设计方法[J]. 航空学报, 2018, 39(7): 45-63  
WANG Chao, GAO Zhenghong, ZHANG Wei, et al. Efficient surrogate-based aerodynamic design optimization method with adaptive design space expansion[J]. Acta Aeronautica et Astronautica Sinica, 2018, 39(7): 45-63 (in Chinese)
- [15] SCHMITT V, CHARPIN F. Pressure distributions on the ONERA-M6-Wing at transonic mach numbers[R]. AGARD AR-138-

B1, 1979

- [16] SEDERBERG T W, PARRY S R. Free-form deformation of solid geometric models[C]//Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, 1986: 151-160

## Applying to aerodynamic optimization an enhanced particle swarm optimization algorithm based on parallel exchange

WANG Peijun, XIA Lu, ZHOU Wenshuo, LUAN Weida

(School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China)

**Abstract:** The particle swarm optimization (PSO) algorithm is easy to implement and can obtain high-quality solutions to optimization problems. It is widely applied to nonlinear and difficult problems such as aerodynamic optimization. However, to solve multi-modal problems, it easily falls into locally optimal values, showing that its robustness is poor. In order to improve the robustness of the PSO algorithm, an enhanced particle swarm optimization algorithm based on parallel exchange (EPSOBPE) is proposed. The algorithm enhances the optimization capability and its robustness through the parallel evolution of the cuckoo search algorithm (CSA), PSO population, hierarchical exchange operation and reinforcement learning strategy. Therefore, the algorithm has both the global search capability of the CSA and the local capability of the PSO algorithm, thus making the EPSOBPE very robust. Functional test results show that the EPSOBPE has stronger robustness and adaptability to different problems than other intelligent optimization algorithms. Moreover, the EPSOBPE is applied to the aerodynamic optimization design of the RAE2822 airfoil and the M6 wing. Compared with other algorithms, the EPSOBPE is more robust, and its optimization capability is better.

**Keywords:** particle swarm optimization algorithm; cuckoo search algorithm; aerodynamic optimization design; global optimization

**引用格式:**王培君,夏露,周文硕,等.基于并行交换的增强粒子群优化算法在气动优化中的应用[J].西北工业大学学报,2022,40(3):493-503

WANG Peijun, XIA Lu, ZHOU Wenshuo, et al. Applying to aerodynamic optimization an enhanced particle swarm optimization algorithm based on parallel exchange[J]. Journal of Northwestern Polytechnical University, 2022, 40(3): 493-503 (in Chinese)