

基于 Alpha-stable 的粒子群算法变异策略研究及气动优化应用

樊华羽, 李典, 郝海兵, 梁益华

(中国航空工业集团西安航空计算技术研究所, 陕西 西安 710065)

摘要:提出了一种基于 Alpha stable 分布的新型变异方法。针对粒子群算法容易陷入局部最优的缺点,通过对比分析确定了一种调整 Alpha stable 分布的稳态系数动态变异策略,使粒子群算法能够在搜索初始阶段具有更强的种群多样性以及算法探索能力,减少陷入局部最优的可能;在算法末期增强粒子群优化算法的局部搜索能力,提高解的精度。将基于 Alpha stable 变异的粒子群优化算法(Alpha stable particle swarm optimization, ASPSO)与多种改进型粒子群优化算法以及差分进化算法(differential evolution algorithm, DE)进行了比较,基准测试函数结果表明新建立的 ASPSO 算法极大地提高了算法的收敛速度和精度。将其应用到 RAE2822 翼型的单点跨声速减阻优化中,在保持种群规模等参数相同的情形下,ASPSO 算法的优化效果和效率都远高于传统 PSO 算法,最终得到的翼型也比 PSO 优化的翼型具有更高的升阻比,翼面波阻有明显降低。

关键词:粒子群优化算法;Alpha-stable 分布;动态变异;气动优化

中图分类号:V211 **文献标志码:**A **文章编号:**1000-2758(2022)06-1385-09

航空科学技术的快速发展对飞行器气动力优化设计技术提出了更高的要求。借助各种优化算法,可以实现翼型、翼身组合体甚至全机减阻降噪设计与分析,为各种新型的飞行器研制提供支撑。

气动力优化设计^[1]的关键技术之一就是优化算法。根据优化机制和行为的不同,这些优化算法可以分为两类:经典优化算法和智能优化算法。相比之下,以粒子群算法(particle swarm optimization, PSO)^[2]和差分演化算法(differential evolution algorithm, DE)^[3]为代表的智能优化算法由于良好的并行效率、全局性以及鲁棒性,得到了更多的关注。其中,PSO 算法简洁易行、参数少、优化能力强,是气动优化领域广泛使用的算法之一。然而,该算法依然存在固有的缺陷,在进化后期,算法侧重于开采能力,容易陷入局部极值点,收敛速度慢,精度较差。为此,研究人员进行了一系列的研究工作并相继提出了多个改进的 PSO 算法。文献[4]针对 PSO 算

法容易陷入早熟的缺点,对 PSO 算法引入田口算法思想,进行多目标优化,取得较好效果,但其是对单波束和半功率宽度进行优化,而没有用在多波束赋形方向,且该算法优化目标数较少。魏法等^[5]从自适应操作粒子结构以及采用反梯度加权的方式平衡粒子群优化算法的全局搜索与局部搜索能力,通过对阵列相位、幅度优化实现低旁瓣、窄波束宽度和高增益效果。明振兴等^[6]通过粒子群算法中的惯性权重线性化,使用余弦函数改进算法中的学习因子,使算法能够兼顾全局与局部寻优性能,并将改进后的粒子群算法应用于 LED 光源阵列的优化。周飞红^[7]对粒子群算法引入爆炸因子,让粒子群能够通过爆炸的方式增强搜索能力,有利于算法跳出局部最优解,提高算法的搜索能力。李真等^[8]将粒子群优化算法结合灰狼算法,根据自然界中的优胜劣汰法则,对每代种群中的最差粒子进行进化,提高粒子群算法的优化性能。综上所述,无论何种改进的

收稿日期:2022-03-06

作者简介:樊华羽(1985—),中国航空工业集团西安航空计算技术研究所工程师,主要从事飞行器总体设计、飞行器气动与隐身多学科优化研究。e-mail:kevin.fan@163.com

PSO 算法,出发点都是如何在提高算法收敛速度的同时提高全局收敛性,避免算法陷入局部最优,但是这些算法的调参方法和应用能力各有优劣。

本文针对飞行器气动优化设计中广泛使用的 PSO 算法的不足之处开展改进研究。在综合分析各种改进算法的基础上,提出了一种基于 Alpha-stable 动态变异的新型粒子群优化算法,使用多个测试函数对该算法的低、中、高维设计变量下的寻优能力进行了分析,最终将其应用到实际的气动优化领域,分析研究了 RAE2822 二维翼型的减阻设计。

1 基于 Alpha-stable 变异的粒子群算法

在粒子群优化算法中,粒子通过(1)~(2)式来更新自己的位置与速度。

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

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

式中: $i = 1, 2, \dots, N; j = 1, 2, \dots, n; x$ 表示粒子所在位置; v 表示粒子速度; ω 为惯性权重因子,是一个在 $[0.4, 0.9]$ 之间动态变化的参数,它决定了粒子继承当前速度的程度,常见的动态变化策略主要有线性递减与线性微分等; c_1, c_2 为学习因子,主要用来调节粒子群步长的参数; r_1, r_2 为 $(0, 1)$ 之间的随机数;下标 i 和 j 表示第 i 个粒子和第 j 维;上标 t 为进化代数。

1.1 Alpha-stable 分布的概率密度函数

概率密度函数是常用且极为重要的表示函数分布的工具,图像中一般表现为尾端趋近于幂律分布,头端偏离幂律分布。由于 Alpha-stable 分布的概率密度函数是存在且连续的,所以可以用 Alpha-stable 分布的特征函数的连续傅里叶变换来定义

$$f(x; \alpha, \beta, \gamma, \sigma) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \exp \cdot [it\sigma - |\gamma t|^\alpha (1 - i\beta \text{sign}(t)) \phi] e^{-itx} dt \quad (3)$$

式中: $\text{sign}(t)$ 是 t 的符号; ϕ 表示为

$$\phi = \begin{cases} \tan(\pi\alpha/2), & \alpha \neq 1 \\ -(2/\pi) \ln |t|, & \alpha = 1 \end{cases} \quad (4)$$

在 Alpha stable 分布中有 3 个特例:

1) 当 $\alpha = 2, \beta = 0$ 时,表示高斯分布,其概率密度函数为

$$f(x) = \frac{1}{2\sqrt{\pi}\gamma} e^{-\frac{(x-\sigma)^2}{4\gamma^2}}$$

2) 当 $\alpha = 1, \beta = 0$ 时表示柯西分布,其概率密度函数为

$$f(x) = \frac{\gamma}{\pi [(x - \sigma)^2 + \gamma^2]}$$

3) 当 $\alpha = 0.5, \beta = 1$ 时表示 Levy 分布,其概率密度函数为

$$f(x) = \left(\frac{\gamma}{2\pi}\right)^{\frac{1}{2}} \frac{1}{(x - \sigma)^{\frac{3}{2}}} e^{\frac{\gamma}{2(x-\sigma)}}$$

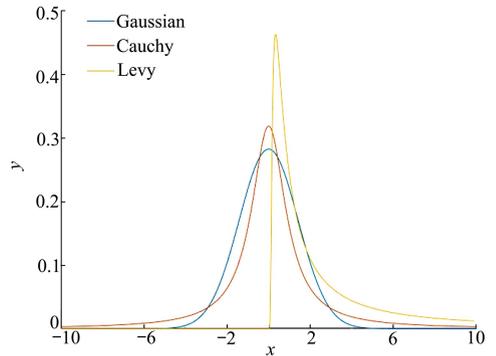
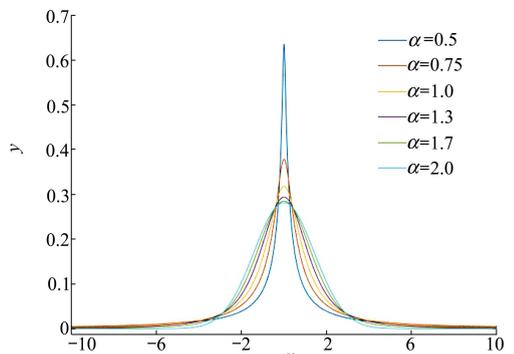


图 1 高斯分布、柯西分布和 Levy 分布的概率密度函数图

除了高斯分布、柯西分布分布 Levy 分布以外,一般的稳定密度函数没有已知的封闭表达式,且其他任何稳定分布的密度函数都不太可能有封闭表达式。Zolotarev 的研究表明^[9],仅在极少数情况下,可以用某些特殊函数表示。

将 Alpha-stable 分布 $X \in S_\alpha(0, 1, 0)$ 在 $\alpha \in (0, 2]$ 内选择 6 个不同的 α 得到如图 2 所示的变化图。此时 $\beta = 0$, 密度函数呈现关于 0 的对称分布。

在 Alpha-stable 分布中,图 2a) 呈现出钟型的对称稳定分布,可以看出当 α 减小时,函数图形会呈现 3 种变化:峰值变高、峰值两侧的区域变低、尾部变重。在图 2b) 中可以了解当 α 减小时尾部概率是如何增加的。通过改变稳定性系数 α 的大小,就可以改变成不同的 Alpha-stable 分布。



a) Alpha-stable 分布的概率密度函数图

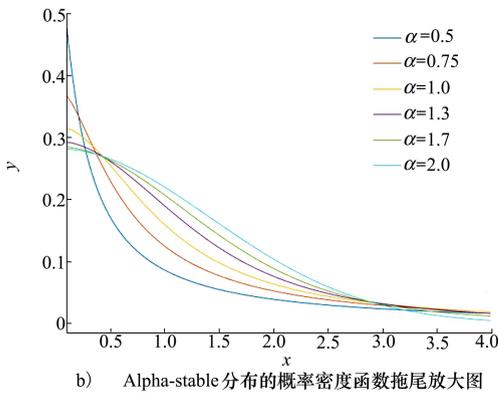


图 2 函数图

1.2 稳定性系数 α 的变化策略与分析

由于粒子群算法容易陷入局部最优,通常在粒子群算法中加入一个确定的变异因子以避免过早陷入局部最优。本文选择通过 Alpha-stable 分布产生的随机数对粒子群算法中的个体进行变异操作。不同的稳定性系数 α 的变化过程决定不同的变异策略,本节通过测试函数来验证 3 种不同的变异策略下 ASPSO 优化算法的寻优能力。设置 1 为前期加速的上凸二阶函数型变异策略,设置 2 为无加速一

阶函数型变异策略,设置 3 为后期加速的下凹二阶函数型变异策略,3 种不同的变异策略趋势见图 3。

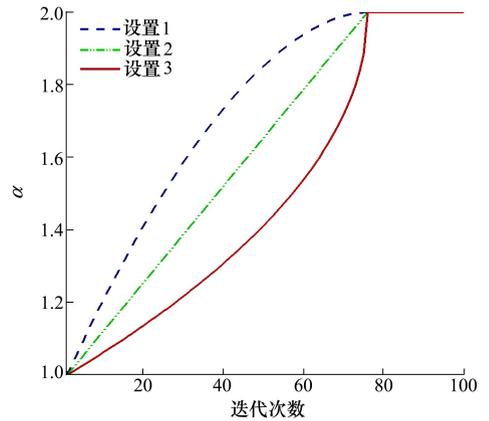


图 3 不同设置的 Alpha 变化策略

通过 7 种标准测试函数在 30 维下的函数测试结果来比较哪种变异策略更具有优势。算法的种群规模均取 100。惯性权重在 $[0.4, 0.9]$ 之间随代数线性递减。稳定性系数 α 的取值从 1 增加到 2。所有算法的加速常数 $c_1 = c_2 = 15$ 。分别用 3 种算法搜索运行 30 次,每次运行迭代 1 000 次,统计 30 次优化的平均目标函数值和搜索到的最小目标函数值。

表 1 不同变化策略下的测试函数结果对比

测试函数	设置 1		设置 2		设置 3	
	均值	标准差	均值	标准差	均值	标准差
Sphere	0	0	0	0	0	0
Schwefel	0	0	0	0	0	0
Elliptic	0	0	0	0	0	0
Rosenbrock	0.527 07	3.582 8	0.268 29	2.573 9	0.0148 05	0.018 947
Ackley	$7.247\ 5 \times 10^{-15}$	$1.454\ 3 \times 10^{-15}$	$7.425\ 2 \times 10^{-15}$	1.309×10^{-15}	$7.354\ 1 \times 10^{-15}$	$1.371\ 8 \times 10^{-15}$
Rastrigin	$7.157\ 4 \times 10^{-12}$	$7.157\ 4 \times 10^{-11}$	0	0	0	0
Griewank	$1.229\ 9 \times 10^{-3}$	$8.980\ 9 \times 10^{-3}$	$1.174\ 1 \times 10^{-3}$	$8.517\ 3 \times 10^{-3}$	$1.178\ 4 \times 10^{-7}$	1.107×10^{-6}

通过表 1 可以看出 3 种设置下的 Alpha 变化策略对粒子群优化算法的影响。一开始都是使用较小的 α 值,此时的变异方式为柯西变异,此种变异方式能够帮助提高粒子群优化算法的全局搜索能力,随着 α 值逐渐增大到 2,变异方式也从柯西变异变化为传统的高斯变异,此种变异方式能够提高算法末期的搜索精度。那么表 1 中可以明确看出在设置 1 与设置 2 下,由于 α 值的增速过大,Alpha-stable 变异过快地从柯西变异过渡为高斯变异,即过快地从

寻找全局最优转换为局部寻优,不能较为优秀地帮助粒子群完成优化任务。总的来说,3 种不同的设置方式下都能够得到较优秀的测试结果。其中测试结果差异较大的函数为 Rosenbrock 函数和 Griewank 函数。Rosenbrock 函数是病态的,在极值周围梯度极小。Griewank 函数是典型的非线性多模态函数,具有大量的局部极小值点。正是因为这 2 个函数本身的特点,对较为激进的变异策略较敏感,使得设置 1 和设置 2 情况下的测试结果没有设置 3 的数值优

秀。设置 3 的变异策略在开始阶段使用较小的 α 值,采用变异能力较强的柯西变异增强了粒子群算法在初始阶段的的全局搜索能力,避免陷入局部最优;随着算法进程 α 值逐渐增大,最终增大为 2,此时的变异方式为传统的高斯变异,提升算法末期的局部搜索能力,增强算法的搜索精度。通过动态调整 Alpha-stable 函数的稳定性系数 α ,完成变异范围以及幅度的变化,从而使优化算法能在搜索最优解时具有更好的精度以及全局搜索能力。最终选择设置 3 作为本文优化算法中的变异策略,稳定性系数 α 的变化过程见图 4,本文命名在此变异策略下的粒子群算法为基于 Alpha-stable 分布的粒子群算法 (ASPSO)。

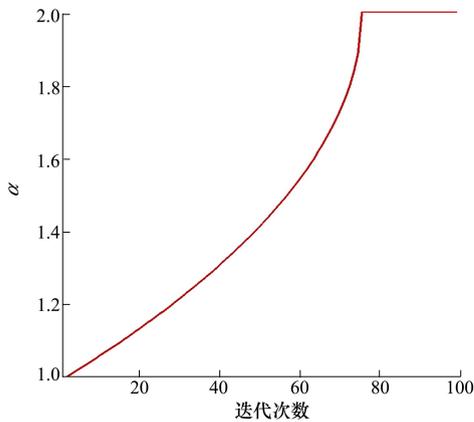


图 4 α 值的变化过程

2 算法验证

2.1 低维优化问题寻优能力检验

在低维优化问题的寻优能力检验中,3 个优化算法的参数设定为:设计变量为 30 维,种群规模均取为 100。基本 PSO 算法的惯性权重在 $[0.4, 0.9]$ 之间随代数线性递减。ASPSO 算法的稳定性系数 α 的取值从 1 增加到 2,变异策略为 1.2 节中所验证的设置 3 下凹型曲线。所有算法的加速常数 $c_1 = c_2 = 1.5$ 。分别用 3 种算法搜索运行 30 次,每次运行迭代 1 000 次,统计 30 次优化的平均目标函数值和搜索到的最小目标函数值。

表 2 显示了在低维优化问题上 7 个测试函数的检验结果。可以看出,不论是这 7 个测试结果的均值还是标准差,本文建立的 ASPSO 算法都是要优于其他 2 种算法。为了更突出地显示 ASPSO 优化算法的优化性能,在图 5 中,显示了 3 种优化算法在 30 维下的 Rastrigin 函数全局最优值收敛曲线。可以很明显地看出 ASPSO 算法甚至能在大量正弦拐点排列很深的局部最优前提下直接搜索到最优解 0,显示了其具有的优化性能。综合来说,在低维优化问题中,ASPSO 优化算法优于 DE 算法以及 PSO 优化算法。

表 2 低维优化问题计算效果对比

测试函数	DE 算法		基本 PSO 算法		ASPSO 算法	
	均值	标准差	均值	标准差	均值	标准差
Sphere	7.79×10^{-12}	2.91×10^{-12}	1.52×10^{-40}	4.11×10^{-40}	0	0
Schwefel	2 288.15	3 100.639 2	$1.564 5 \times 10^{-3}$	1.99×10^{-3}	0	0
Elliptic	1.27×10^{-8}	4.11×10^{-9}	2.93×10^{-37}	6.31×10^{-37}	0	0
Rosenbrock	65.731 3	22.993 6	30.081 5	24.882 5	$3.219 4 \times 10^{-2}$	$4.264 9 \times 10^{-2}$
Ackley	6.97×10^{-7}	1.21×10^{-7}	$3.104 3 \times 10^{-2}$	0.170 03	7.40×10^{-15}	1.35×10^{-15}
Rastrigin	59.726 5	4.713 2	38.604 4	10.556 8	0	0
Griewank	2.06×10^{-10}	4.45×10^{-10}	1.165×10^{-2}	0.012 338	6.34×10^{-9}	6.43×10^{-9}

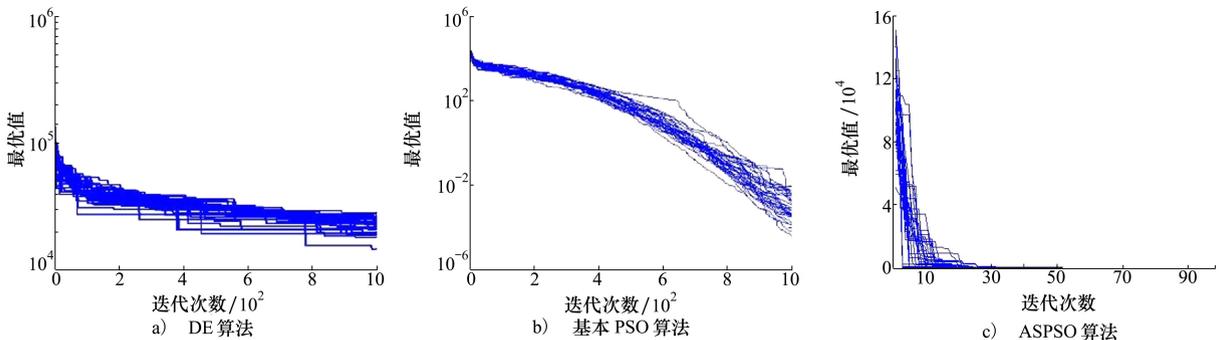


图 5 3 种算法在 Rastrigin 函数的低维优化全局收敛图

2.2 中、高维优化问题寻优能力检验

将设计变量分别扩展到 60 和 100 维,其他参数不变,测试 3 种算法在中、高维问题中的寻优能力。具体参数设置跟低维优化问题时相同,每种算法进行 30 次优化,每次进行 1 000 次迭代,最后统计得到这 30 次优化结果的平均目标函数值与最小函数值。

Rastrigin 函数 60 维设计变量优化问题的全局收敛曲线见图 6。7 个函数的均值和标准差结果见

表 3。Rosenbrock 函数的 100 维高维问题的收敛曲线见图 7,7 个函数的均值和标准差值统计结果见表 4。可以看出随着维数的增加,3 种算法的寻优精度都不同程度地降低,尤其是在高维优化问题中,寻优精度下降得很厉害。但是相比来看,在调用目标函数次数相同的情况下,ASPSO 在平均值和标准差上都反映出了更高的精度,收敛效果更好,表明了其对于中、高维问题的较好适应能力。

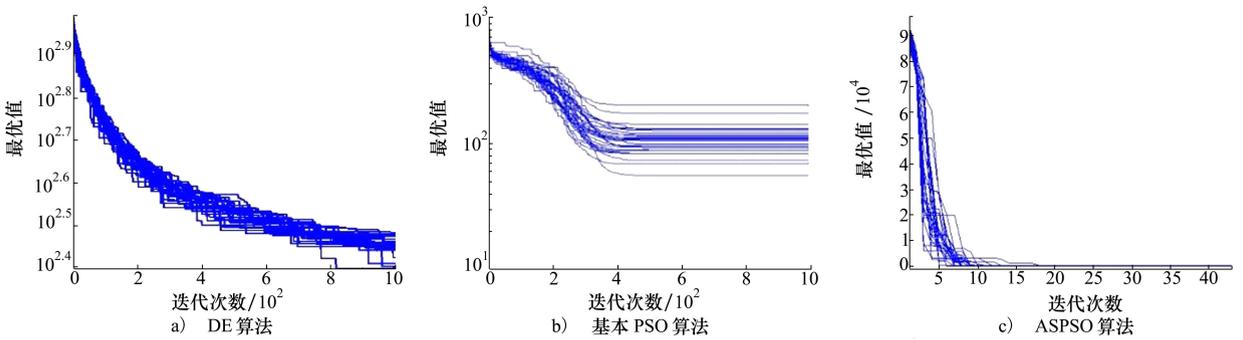


图 6 3 种算法在 Rastrigin 函数的中维优化全局收敛图

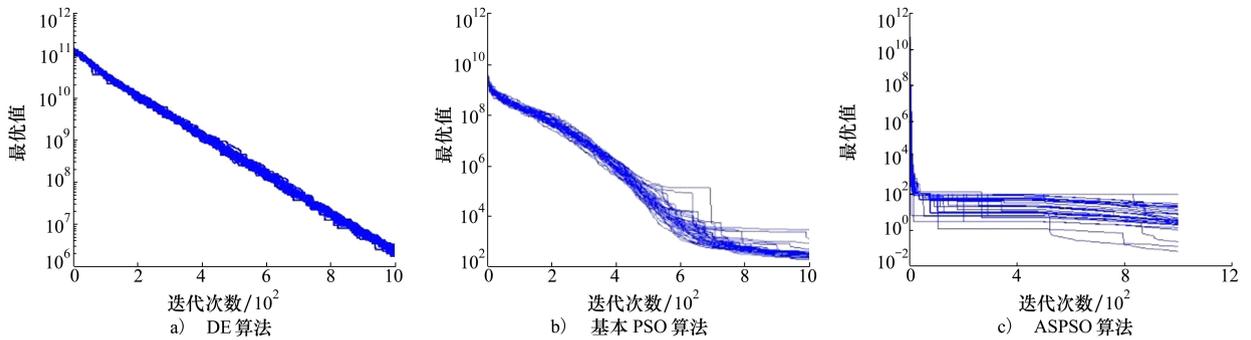


图 7 3 种算法在 Rosenbrock 函数的高维优化全局收敛图

表 3 中维优化问题计算结果对比

测试函数	DE 算法		基本 PSO 算法		ASPSO 算法	
	均值	标准差	均值	标准差	均值	标准差
Sphere	4.27×10^{-3}	6.58×10^{-3}	2.12×10^{-13}	6.91×10^{-13}	8.97×10^{-8}	4.91×10^{-17}
Schwefel	132 544.01	8 833.384 6	385.783 7	263.321 4	0.036 105	0.078 582
Elliptic	3.864 7	0.592 78	2.02×10^{-9}	7.01×10^{-9}	0	0
Rosenbrock	1 442.369 1	253.937 8	82.345 5	39.048 4	10.655 5	20.687 5
Ackley	0.013 124	0.001 29	1.456 2	0.602 74	9.98×10^{-6}	5.47×10^{-5}
Rastrigin	282.983	13.205 8	107.687 5	30.037 5	2.06×10^{-6}	8.43×10^{-6}
Griewank	6.01×10^{-3}	1.88×10^{-3}	1.19×10^{-2}	2.007 3	1.24×10^{-5}	1.03×10^{-5}

表 4 高维优化问题计算效果对比

测试函数	DE 算法		基本 PSO 算法		ASPSO 算法	
	均值	标准差	均值	标准差	均值	标准差
Sphere	32.060 8	2.963 1	0.015 059	0.042 626	1.40×10^{-5}	7.02×10^{-5}
Schwefel	368 656.08	30 912.335	7 254.025 9	2 231.435	7.565 9	14.984 6
Elliptic	18 683.689	1 925.892 3	202.486 8	760.372 9	4.29×10^{-4}	2.02×10^{-3}
Rosenbrock	2 436 751.7	432 694.94	472.273 2	535.462 6	13.554 9	23.762 2
Ackley	2.358 8	0.103 1	2.587 5	0.529 53	1.52×10^{-4}	5.44×10^{-4}
Rastrigin	691.203 1	20.463 7	180.055 8	29.796 1	1.53×10^{-4}	2.52×10^{-4}
Griewank	1.275 4	0.033 412	0.092 413	0.118 71	8.28×10^{-3}	0.033 773

综上所述,无论是对低维还是中高维问题,相比基本 PSO 算法和 DE 算法,ASPSO 算法都能够以更快的收敛速度、稳定性和鲁棒性获得单峰及多峰函数更高精度的最优解。同时还能够发现,以较为难以得到结果的 Rosenbrock 函数为例,在处理同一个函数不同维度的最优问题时,DE 算法的性能相对较差,都无法寻优。而对比 PSO 与 ASPSO 算法,可以看出,中高维寻优中 PSO 算法有明显的性能下降,而 ASPSO 则体现了更为优秀的鲁棒性。所以相比其他 2 种算法,ASPSO 是一种更加稳定且高效的全局优化算法。

2.3 多种 PSO 算法性能对比

为了能够更为全面地验证 ASPSO 优化算法的性能与优势,本节将选择多个先进 PSO 改进算法与 ASPSO 优化算法进行对比实验。本节选取的算法有:PSO-DAC(PSO based on dynamic acceleration coefficients)^[10]、NDPSO(a normal distribution decay inertial weight PSO)^[11]、EXPPSO(exponential decay weight PSO)^[12]、SCVPSO(self-conclusion and self-adaptive variation PSD)^[13]。PSO-DAC 算法是基于

动态加速因子的改进粒子群优化算法,通过引入动态的加速因子,改善粒子群算法的收敛速度及精度。NDPSO 算法以正态分布曲线作为惯性权重的衰减策略曲线,引入控制因子对粒子的位置进行改善,使 NDPSO 算法能够在优化过程中很好地平衡全局搜索与局部搜索能力。EXPPSO 算法在速度更新方程中引入约束因子,并对惯性系数采用指数衰减模式,使算法能够兼顾优秀的全局以及局部搜索能力。SCVPSO 算法采用动态惯性权重、反向搜索以及引入新的参数 SCR(self-conclusion rate)等方法综合性地提高粒子群的优化性能。本验证环节的算法参数设置为:设计变量为 30 维,种群规模均取为 100。非特殊算法的惯性权重在[0.4,0.9]之间随代数线性递减。其余的采用其特有的惯性权重递减方式。所有算法的加速常数及 ASPSO 算法中的稳定性系数变化策略保持不变。为了全面验证算法的性能,验证算法选择单峰测试函数与多峰测试函数各 3 个。测试结果见表 5。可以看出相比其他改进型 PSO 算法,ASPSO 算法具有更为优秀的性能。

表 5 多种改进 PSO 算法性能对比

算法	Sphere	Schwefel	Rosenbrock	Rastrigin	Ackley	Griewank
ASPSO	0	0	7.93	0	8.88×10^{-16}	4.55×10^{-4}
PSO-DAC	1.22×10^{-12}	1.06	55.9	40.1	1.99×10^{-1}	1.56×10^{-2}
NDPSO	1.43×10^{-19}	3.56	4.42×10^2	59.1	1.20	4.12×10^{-2}
EXPPSO	4.31×10^{-7}	3.68	53.2	49.9	8.59×10^{-1}	1.61×10^{-1}
SCVPSO	1.43×10^{-52}	2.47×10^{-9}	2.63	64.2	1.59×10^{-1}	5.90×10^{-3}

3 优化算例分析

3.1 优化模型及参数化方法

为了更好地分析 ASPSO 算法在实际应用中的优化效果,选择对 RAE2822 翼型进行工程应用减阻优化。以其在 $Ma=0.73, \alpha=2.79^\circ, Re=6.5 \times 10^6$ 时的阻力最小为目标进行气动外形优化设计^[14],约束条件是优化后的翼型升力系数不小于原始翼型的升力系数值,并且翼型的最大厚度不减小,同时俯仰力矩不恶化。综上,可得到优化设计的数学模型:

$$\begin{cases} \min f_1(\mathbf{X}) = C_d \\ \text{s.t. } g_1(\mathbf{X}) = C_l \geq C_l^* \\ g_2(\mathbf{X}) = c_{\max} \geq c_{\max}^* \\ g_3(\mathbf{X}) = C_m \leq C_m^* \end{cases}$$

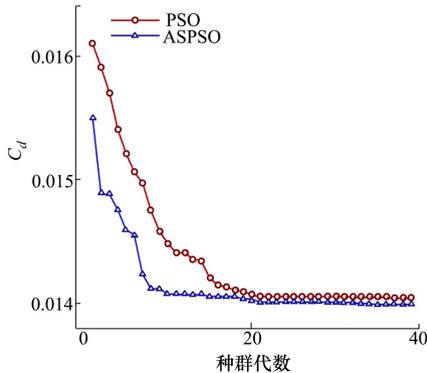


图 8 减阻优化过程

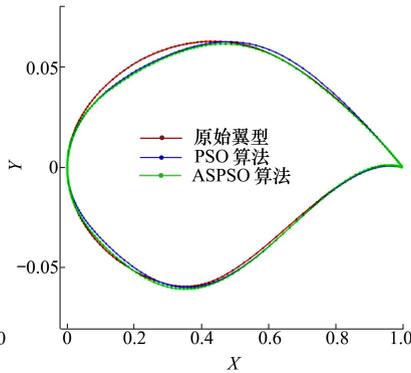


图 9 优化前后翼型剖面形状对比

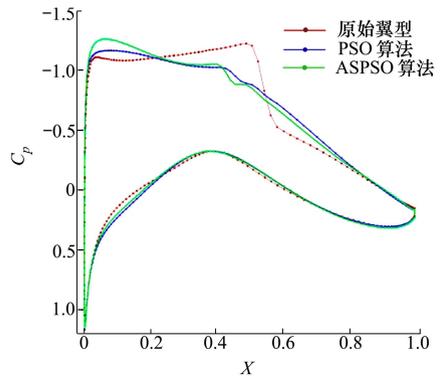


图 10 优化前后压力分布对比

图 8 显示了 2 种优化算法的减阻优化过程,可以很明显看出 ASPSO 的优化程度与效率都优于基本 PSO 算法,ASPSO 算法在第 10 次迭代时就能接近最终优化结果附近。在优化结果上来看,相比 PSO 算法,ASPSO 算法能在保证优化程度上拥有更快的收敛速度。表 6 展示了 2 种优化算法的优化结果与原始翼型的气动参数对比,结果表明 2 种优化算法已满足基本的优化需求。图 9 与图 10 为优化前后的翼型的剖面形状以及压力分布。结合图 9 与图 10 可知,机翼的最大厚度后移,同时上表面翼面顶点的移动也使得优化后的翼型上翼面更加饱满,翼面激波有明显减缓,优化后的上翼面压力分布明显比原始翼型更加柔和。综合来看,针对 RAE2822 跨声速翼型的减阻优化,其结果显示波阻显著降低,升阻比有明显提高。相比于基本 PSO 优化算法,ASPSO 算法有更优秀的适应性。在同样的种群数

式中: \mathbf{X} 为设计变量; C_d 表示翼型的阻力系数; C_m 表示翼型的力矩系数; C_l 表示翼型的升力系数; c_{\max} 代表翼型的最大厚度,带上标 * 的符号表示原始翼型的气动数据。

针对 RAE2822 翼型,本节选用的是结合六阶 Bernstein 多项式型函数的 CST 方法作为翼型参数化方法。对翼型上下表面各取 6 个(共 12 个)设计参数作为本次优化过程中的设计变量。

3.2 优化结果及分析

采用基本 PSO 算法与 ASPSO 算法 2 种方法对单点跨声速优化设计进行优化,比较 2 种方法的优化效果。2 种优化算法的种群大小均为 25,迭代步数为 39,惯性权重 w 在 $[0.4, 0.9]$ 之间随代数线性递减。加速系数 $c_1 = c_2 = 15$ 。稳定性系数 α 变化范围为 1.0 至 2.0。

量和优化代数下,ASPSO 算法具有更快的收敛速度以及更强的全局优化能力,优化结果也更理想。

表 6 优化前后的气动参数对比

数据来源	C_l	C_d	C_m	c_{\max}
原始翼型	0.754 5	0.017 57	-0.085 8	0.120 8
PSO	0.754 7	0.014 04	-0.084 5	0.120 0
ASPSO	0.754 5	0.013 99	-0.081 2	0.120 0

4 结论

本文针对粒子群算法容易陷入局部最优等缺点,通过 Alpha stable 分布产生的随机数对其进行变异操作,以实现粒子群优化算法优化性能的提高。通过测试函数验证了 3 种不同的 α 值变异策略对于

粒子群优化算法的影响。选择了其中较优的变异策略参与进粒子群优化算法当中,使得算法在优化初始阶段能够更好地摆脱局部最优,在算法末期提高算法精度。

在多个测试函数验证环节中,选择了 DE、PSO 以及 ASPSO 3 种优化算法在低维以及中、高维优化问题中进行对比验证,结果表明 ASPSO 算法在单目标优化中优于 DE 及 PSO 算法。为了更进一步验证算法性能,横向对比多种改进策略下的 PSO 优化算法性能,选取了多种先进改进型 PSO 优化算法参与

标准测试函数验证过程,结果表明 ASPSO 优化算法仍具有相当的优越性。综上所述,函数测试结果表明运用 Alpha stable 分布对 PSO 算法进行的变异过程的 ASPSO 优化算法具有优秀的性能。

最后,通过对 RAE2822 翼型进行工程应用减阻优化来验证 ASPSO 算法在工程应用上优于传统 PSO 算法。相比 PSO 算法,ASPSO 算法的优化结果具有更好的力学特性,翼面波阻明显降低,ASPSO 算法的优化结果拥有更好的气动性能。

参考文献:

- [1] HICKS R M, HENNE P A. Wing design by numerical optimization[J]. Journal of Aircraft, 1979, 15(7): 407-413
- [2] POLI R. Analysis of the publications on the applications of particle swarm optimization[J]. Journal of Artificial Evolution and Applications, 2008, 3: 1-10
- [3] STORN R, PRICE K. Differential evolution—a simple and efficient Heuristic for global optimization over continuous spaces[J]. Journal Global Optimization, 1997, 11: 341-359
- [4] YIGIT M E, GUNEL T. Pattern synthesis of linear antenna array using hybrid particle swarm-taguchi optimization[C]//Proceeding of the 10th International Conference on Electrical and Electronics Engineering, 2017
- [5] 魏法, 杨明磊, 何小静, 等. 基于改进粒子群算法的平面阵同时多波束赋形方法[J]. 系统工程与电子技术, 2022, 44(6): 1789-1797
WEI Fa, YANG Minglei, HE Xiaojing, et al. Simultaneous multi-beam forming method for planar array based on improved particle swarm algorithm[J]. Systems Engineering and Electronics, 2022, 44(6): 1789-1797 (in Chinese)
- [6] 明振兴, 吕清花, 明月, 等. 基于改进粒子群算法的 LED 光源阵列优化[J]. 应用光学, 2022, 43(3): 524-531
MING Zhenxing, LYU Qinghua, MING Yue, et al. Optimization of LED light source array based on improved particle swarm algorithm[J]. Journal of Applied Optics, 2022, 43(3): 524-531 (in Chinese)
- [7] 周飞红. 一种改进的粒子群优化算法[J]. 集成电路应用, 2021, 38(11): 74-75
ZHOU Feihong. An improved particle swarm optimization algorithm[J]. Application of IC, 2021, 38(11): 74-75 (in Chinese)
- [8] 李真, 王帆, 王冉. 一种结合灰狼算法的粒子群优化算法[J]. 计算机测量与控制, 2021, 29(10): 217-222
LI Zhen, WANG Fan, WANG Ran. An improved particle swarm optimization algorithm[J]. Computer Measurement & Control, 2021, 29(10): 217-222 (in Chinese)
- [9] ZOLOTAREV V M. One-dimensional stable distributions[M]. Providence: American Mathematical Society, 1986
- [10] 滕志军, 吕金珍, 郭力文, 等. 基于动态加速因子的粒子群优化算法研究[J]. 微电子学与计算机, 2017, 34(12): 125-129
TENG Zhijun, LYU Jinzhen, GUO Liwen, et al. Research on particle swarm optimization based on dynamic acceleration coefficients[J]. Microelectronics & Computer, 2017, 34(12): 125-129 (in Chinese)
- [11] 徐造天, 季伟东, 孙小晴, 等. 基于正态分布衰减惯性权重的粒子群优化算法[J]. 深圳大学学报, 2020, 37(2): 208-213
XU Zao Tian, JI Weidong, SUN Xiaoqing, et al. A PSO algorithm with inertia weight decay by normal distribution[J]. Journal of Shenzhen University, 2020, 37(2): 208-213 (in Chinese)
- [12] YAN C M, LU G Y, LIU Y T, et al. A modified PSO-algorithm with exponential decay weight[C]//13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, 2017
- [13] 陈博文, 邹海. 总结性自适应变异的粒子群算法[J]. 计算机工程与应用, 2022, 58(8): 67-75
CHEN Bowen, ZOU Hai. Self-conclusion and self-adaptive variation particle swarm optimization[J]. Computer Engineering and Applications, 2022, 58(8): 67-75 (in Chinese)

[14] VAVALLE A, QIN N. Iterative response surface based optimization scheme for transonic airfoil design[J]. Journal of Aircraft, 2012, 44(2): 365-376

A fast PSO algorithm based on Alpha-stable mutation and its application in aerodynamic optimization

FAN Huayu, LI Dian, HAO Haibing, LIANG Yihua

(AVIC Xi'an Aeronautics Computing Technique Research Institute, Xi'an 710065, China)

Abstract: In this paper, a novel mutation method based on Alpha stable distribution is proposed. Aiming at the shortcoming that the particle swarm optimization (PSO) is easy to fall into local optimum, a dynamic mutation strategy of Alpha stable distribution is determined through comparative analysis. This mutation strategy can make the particle swarm optimization algorithm based on Alpha-stable (ASPSO) have stronger population diversity and exploration ability in the initial stage of search, and make the algorithm avoid falling into local optimum. At the end of the algorithm, it can also enhance the local search ability of the particle swarm optimization algorithm and improve the accuracy of the solution. The ASPSO algorithm is compared with several improved particle swarm optimization algorithms and differential evolution algorithm. The benchmark function results show that the new ASPSO algorithm greatly improves the convergence speed and accuracy of the algorithm. Finally, both PSO and ASPSO algorithms were applied to a minimal drag optimization design of the RAE2822 airfoil and compared. The comparisons show that the ASPSO algorithm achieves a lower drag in a faster rate which lifts and pitching moment is well constrained.

Keywords: particle swarm optimization; Alpha-stable distribution; dynamic mutation; aerodynamic optimization

引用格式: 樊华羽, 李典, 郝海兵, 等. 基于 Alpha-stable 的粒子群算法变异策略研究及气动优化应用[J]. 西北工业大学学报, 2022, 40(6): 1385-1393

FAN Huayu, LI Dian, HAO Haibing, et al. A fast PSO algorithm based on Alpha-stable mutation and its application in aerodynamic optimization[J]. Journal of Northwestern Polytechnical University, 2022, 40(6): 1385-1393 (in Chinese)