

# 基于莱维搜索的寄生算法在气动优化中的应用

夏露, 李朗, 王亮

(西北工业大学 航空学院, 陕西 西安 710072)

**摘要:**为了平衡智能优化算法的全局与局部搜索能力,对布谷鸟算法和寄生算法进行了探讨与分析,将布谷鸟算法中的莱维搜索加入到改进的寄生算法中,在增加收敛速度的同时保证个体多样性,进而发展得到了基于莱维搜索的寄生算法。通过函数测试及气动算例验证了新算法优越的寻优性能。将基于莱维搜索的寄生算法应用于翼型的气动优化设计中,取得了良好的效果,从而表明提出的算法准确有效,具有良好的实用性。

**关键词:**莱维搜索;寄生算法;翼型气动优化

**中图分类号:**V211.4

**文献标志码:**A

**文章编号:**1000-2758(2016)05-0767-07

随着计算机技术和计算流体力学(CFD)的快速发展,采用数值优化方法进行飞行器的气动外形设计,可以大大提高优化设计效率并减少设计成本<sup>[1]</sup>。近年来,粒子群算法(PSO)、布谷鸟算法(CS)、寄生算法(SP)等现代启发式智能优化算法相继被提出,相较于传统的优化算法,这些算法表现出了较强的全局搜索能力,往往可以更快速准确地寻找到优化的期望值<sup>[2]</sup>。CS算法通过莱维搜索可以保证收敛性,而且稳定性较好,SP算法则可以有效继承前代的优势信息。然而,与更多优秀的智能优化算法相比,CS算法收敛速度依旧较慢,而SP算法在寻优过程中则容易陷入局部最优。

为了获得寻优性能更佳的优化算法,本文对CS算法、SP算法进行了探讨与分析,将CS算法中的核心技术莱维搜索引入到寄生算法中,提出了一种基于莱维搜索的寄生算法(LSP),使得改进的算法具有良好的全局/局部搜索平衡能力,有效地提高了算法的寻优性能。将该算法应用到翼型的气动优化设计中,取得了良好的结果。

## 1 布谷鸟搜索算法

布谷鸟搜索算法<sup>[3-4]</sup>是杨新社等人在2009年提

出的一种智能优化算法,其生物模型为布谷鸟的鸟巢寄生繁殖行为,具体模型如下所述:布谷鸟把鸟蛋置于其他鸟类的巢中,靠其孵化以及提供食物,不过同时也存在被寄主发现的可能,这时寄主会把布谷鸟蛋弃于巢外或直接将巢抛弃;另一方面,布谷鸟为了提高生存力,布谷鸟先孵化后会破坏寄主的蛋,以获得更多的生存资源。基于3个假设便得到了简化的算法:①布谷鸟1次只产1个蛋,并放在随机选取的巢中;②质量最好的鸟巢将会被保留到下一代;③鸟巢的数量是固定的,布谷鸟蛋被发现的概率为 $P_a \in [0, 1]$ 。理论证明布谷鸟搜索满足全局收敛要求,从而保证了全局收敛特性。

算法以昆虫的觅食飞行模式——莱维飞行(Levy flights)作为其全局搜索方式<sup>[4]</sup>。搜索步长 $s$ 取自莱维分布

$$\text{Levy} \sim u = t^{-\lambda} (1 < \lambda < 3) \quad (1)$$

式中, $t$ 为进化代数。这种分布的方差和平均值均为无穷大。莱维飞行的特性在许多动物和昆虫的飞行行为中得到体现,应用在优化搜索中时显示出了可观的潜力。

算法运用了局部搜索(local random walk)与全局搜索(global random walk)的结合,由参数 $P_a$ 控制其转换。

局部搜索的方式为

$$x_i^{t+1} = x_i^t + \alpha s \otimes H(p_a - \varepsilon) \otimes (x_j^t - x_k^t) \quad (2)$$

式中,  $x_i^t$  代表第  $i$  个个体的第  $t$  代位置,  $\alpha$  为调整步长的参数,  $\otimes$  为点对点乘法,  $H(u)$  是 Heaviside 函数 (0 for  $x < 0$ , 1 for  $x > 0$ , and 0.5 for  $x = 0$ ), 变量  $\varepsilon$  是取自均匀分布的一个随机数,  $s$  为步长。

全局搜索的方式为

$$x_i^{t+1} = x_i^t + \alpha L(s, \lambda) \quad (3)$$

式中

$$L(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin(\pi \lambda)}{\pi} \frac{1}{s^{1+\lambda}}; \quad (s \gg s_0 > 0) \quad (4)$$

图 1 为布谷鸟搜索算法的算法流程, 图 2 显示了 Levy flights 选取的随机数。

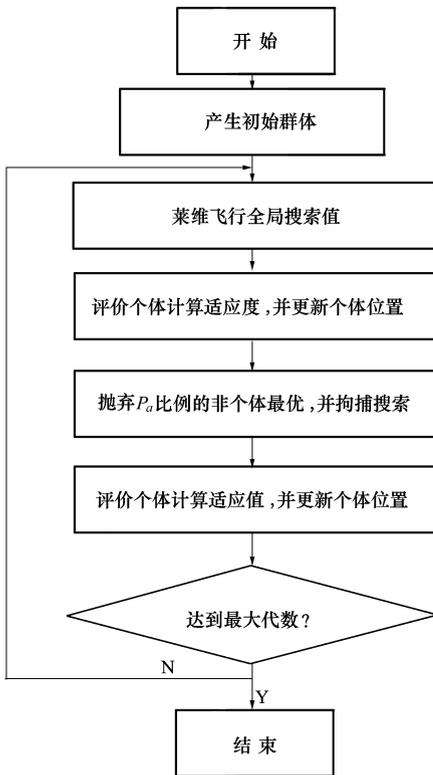


图 1 Cuckoo Search 算法流程

由图 2 可以看出, Levy flights 选出的随机数具有较大的方差, 即波动性比较大, 这样能较大程度上保证样本的多样性, 不会轻易漏掉潜在的全局最优解。同时, 还可以通过调节系数来对随机数进行缩放, 使变量尽量落在限制的区间内。

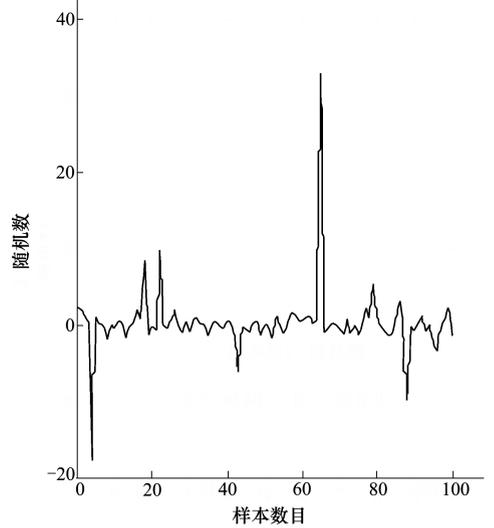


图 2 Levy flights 选取的随机数

## 2 寄生算法

### 2.1 基本寄生算法

自然界中的大部分寄生物需要获取宿主体内的养分才能生存, 并在合适的部位产下后代, 越是营养丰富的部位越能吸引寄生物产生下一代<sup>[5]</sup>。在宿主体内发育的下一代个体只有能获得足够养分的个体才能发育为成熟个体, 而其他个体则被淘汰掉, 这里假设被淘汰掉的个体将由发育成熟的个体所吞噬。将寄生模式以整个寻优空间为宿主, 养分视为适应值, 寄生物群在整个寻优空间寻找最优解, 寄生物群落最后应该聚集在最优解周围, 这样便得到了下面的寄生算法 (SP)。

每一代保留一定数量的成熟个体, 其产生后代的数量由下式确定:

幼体数目为

$$N = k \frac{P_{\text{bestFit}}}{G_{\text{bestFit}}} N_0 \quad (5)$$

式中, 参数  $k$  为繁殖系数,  $P_{\text{bestFit}}$  为当前个体最优适应度值,  $G_{\text{bestFit}}$  为群体最优适应度值,  $N_0$  为繁殖基准数。

拉丁超立方抽样的范围:  $R = R_0/q$ , 其中,  $q$  为拥挤度因子, 其值由下面的公式给出; 其中,  $N_{\text{iter}}$  为当前迭代次数,  $R_0$  为初始变量区间。

$$q = \begin{cases} N_{\text{iter}}^2, & N_{\text{iter}} < 10 \\ 350 \exp((G_{\text{bestFit}}/P_{\text{bestFit}})^2), & N_{\text{iter}} \geq 10 \end{cases} \quad (6)$$

如果成熟个体适应度值劣于全局最优值, 其后

代将均匀分布于该个体周围、整个设计空间内和全局最优解周围。分布于整个设计空间的后代是通过拉丁超立方抽样产生。具体操作方式如下

$$x = \begin{cases} rand * q/5 + P_{best} \\ random \\ rand/(200 * q^2) + G_{best} \end{cases}, q = P_{bestFit}/G_{bestFit} \quad (7)$$

上式中  $rand$  为  $0 \sim 1$  之间的随机数,  $random$  为随机产生的个体位置,  $P_{best}$  为当前个体最优位置,  $G_{best}$  为群体最优位置。

如果成熟个体适应度值优于全局最优值,则后代将围绕在该个体周围产生

$$x = \frac{rand}{35 \exp(G_{bestFit}/P_{bestFit})^2} + P_{best} \quad (8)$$

如此得到每一代幼体并计算其适应值,适应值差的那些幼体将被淘汰掉,最后存活下来的  $M$  个幼体将发育为成熟个体,继续产生下一代。

SP 算法的流程结构如图 3 所示:

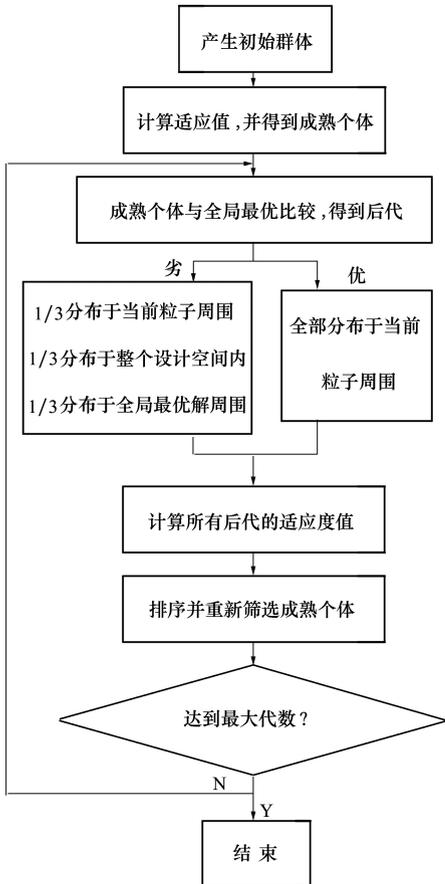


图 3 SP 算法流程

## 2.2 寄生算法的改进

寄生算法虽然强化了局部搜索,但是还存在一些不足。首先,寄生算法中成熟个体产生的后代数目过多,以  $N=60, M=5$  为例,每一代产生后代的数量则为 300,因此为了减少计算量,将产生后代的模式调整为  $15 \times 4 = 60$  个。其次,为了加快收敛速度,将产生后代的模型,改为每个成熟个体只在自身周围产生后代,抛弃随机生成的部分。除此之外,算法收敛得好需要后期搜索的精细化,也就是搜索范围要逐渐缩小。因此,对 SP 算法产生后代公式中的参数做出调整,使搜索半径随着优化代数的增加而递减。如此便得到了改进的寄生模型。

对于劣于全局最优的成熟个体

$$x = rand/(q * N_{iter}^{0.5}) + P_{best} \quad (9)$$

式中,  $q = P_{bestFit}/G_{bestFit}$

对于优于最优的成熟个体

$$x = \frac{rand}{35 \exp(G_{bestFit}/P_{bestFit})^2} + P_{best} \quad (10)$$

为了检验改进的算法收敛速度,下面将对算法进行函数测试。本文用到的函数如下所示:

1) Sphere 函数

$$f(x) = \sum_{i=1}^n x_i^2$$

搜索范围:  $-100 \leq x_i \leq 100$

全局最小值:  $\min(f_1) = f_1(0, \dots, 0) = 0$

此函数为非线性的对称单峰函数,大多数算法都能轻松达到优化效果,其主要用于测试算法的寻优精度。

2) Rosenbrock 函数

$$f_2(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$$

搜索范围:  $-30 \leq x_i \leq 30$

全局最优值为:  $\min(f_2(x)) = f_2(1, 1, \dots, 1) = 0$

该函数是一个单峰函数,靠近最优点的区域为香蕉状,变量之间有很强的相关性,且梯度信息经常误导算法的搜索方向。

3) Ackley 函数

$$f(x) = -20 \exp \left( -0.2 \sqrt{\frac{1}{30} \sum_{i=1}^n x_i^2} \right) - \exp \left( \frac{1}{30} \sum_{i=1}^{30} \cos 2\pi x_i \right) + 20 + e$$

搜索范围:  $-50 \leq x_i \leq 50$

全局最优值为:  $\min(f_2) = f_2(0, \dots, 0) = 0$

该函数是一个连续、旋转、不可分离的多峰函数,存在大量的局部最优点。

下面选取 Rosenbrock 和 Sphere 这 2 个函数对 SP 算法及改进 SP 算法进行测试,最大优化代数设为 100,取 50 次测试的平均值作为性能指标。函数测试结果如表 1 所示。

表 1 算法函数测试对比

算法	Rosenbrock		Sphere	
	平均	最小	平均	最小
SP 算法	68.490 2	6.582 2	0.013 694	0.003 881 2
改进 SP 算法	65.425 5	5.284 5	0.001 906	$2.069 \times 10^{-5}$

通过以上测试可以看出,改进的寄生模型在大大减小计算量的基础上,而且算法的寻优性能有所加强,后期仍然可以提供较大的种群多样性,可以有效地避免陷于局部最优。

### 3 基于莱维搜索的寄生算法

CS 算法通过莱维搜索可以保证收敛性,而且稳定性较好,但是收敛速度相对较慢,往往需要上千次的优化代数才能得到较理想的优化结果,限制了算法的应用范围。SP 算法可以有效继承前代的优势信息,但是其搜索机制限制了收敛性,过大的收缩半径会导致搜索效率的降低,而过小的搜索半径则有可能导致陷入局部最优。为了保证算法收敛性的同时,加以优势信息的指导以期增加算法的收敛速度,考虑将 CS 算法中的核心技术莱维搜索方法与改进的寄生算法相结合。

在寄生算法中,每个成熟个体及其得到的后代都可以看做是一个群落,通过自然选择不断进行繁衍和进化,使得有优势的个体得以生存,淘汰掉没有竞争力的个体。基于上述原理,本节将 CS 算法中的莱维搜索加入到改进的寄生算法中,这样便得到了基于莱维搜索的寄生算法(Levy simulated parasitism, LSP)。如此一来,后代在保留了最优解的潜在设计空间的条件下,还能够通过莱维搜索继续有效寻优搜索,增大了寻优全局最优解的概率。

LSP 算法仍然以寄生算法为框架,在种群中选择较优的  $n$  个个体作为成熟个体,然后由每个成熟个体产生  $m$  个后代,得到  $n * m$  个后代。在成熟个体产生后代的过程中,成熟个体与最优个体比较,如

果成熟个体的适应值不优于当前最优,则后代在最优个体附近得到后代,方式如下所示

$$x_i = G_{best} + \alpha L(s, \lambda) \tag{11}$$

否则后代在成熟个体附近产生,方式如下所示

$$x_i = P_{best} + \alpha L(s, \lambda) \tag{12}$$

式中,  $L(s, \lambda)$  为莱维搜索,如前文所描述;系数  $\alpha$  可以调整搜索的步长,采用线性递减的策略,如下式所示

$$\alpha = \alpha_{start} - \frac{\alpha_{start} - \alpha_{end}}{\maxgen} * k \tag{13}$$

式中,  $\alpha_{start}$ 、 $\alpha_{end}$  分别为开始及结束时的步长系数,  $\maxgen$  为最大迭代代数,  $k$  为当前优化代数。

算法的流程如图 4 所示:

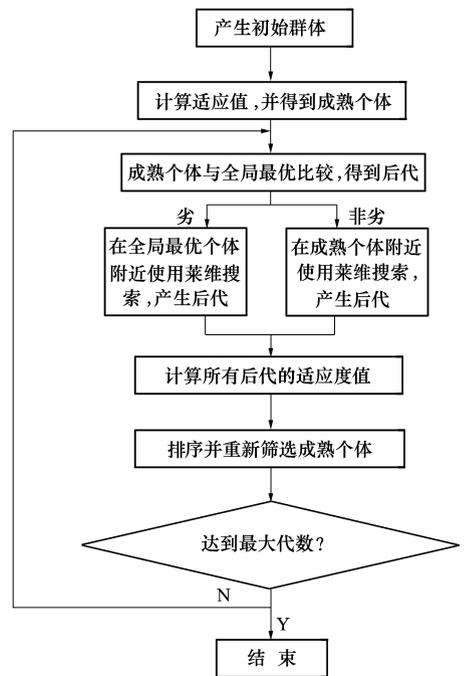


图 4 LSP 算法流程

### 4 算法函数测试

为了比较 CS 算法、改进的 SP 算法和 LSP 算法的性能,并且验证 LSP 算法是否具有良好的寻优能力,分别用这 3 种算法进行函数测试,并与粒子群算法(PSO)进行对比。PSO 算法是一种模拟鸟群飞行觅食行为的智能优化算法,该算法简单易实现,无复杂的参数调整,具有一定的全局/局部搜索平衡能力,被广泛应用于各种设计领域。

这里选用的测试函数是 Rosenbrock 函数以及 Ackley 函数,为了保证各优化算法在优化过程中调

用测试函数的次数尽可能相同,从而保证相当的计算量,各优化算法参数设置如表2所示,函数测试结果如表3所示。

表2 算法设置

算法	种群数目	优化代数
CS 算法	20	60
改进的 SP 算法	15	70
PSO 算法	40	50
LSP 算法	15	10

表3 算法函数测试

算法	Rosenbrock		Ackley	
	平均	最小	平均	最小
CS	165.534 8	62.481 4	2.271 1	1.699 6
改进的 SP	216.594 1	15.319 0	5.851 7	4.886 7
PSO	21.990 7	10.465 6	2.489 8	0.868 6
LSP	17.307 3	9.842 0	0.794 3	0.093 4

通过函数测试可以看出,不管是多峰的 Ackley 函数还是单峰的 Rosenbrock 函数,多次函数测试的平均值以及最小值,在相同计算量的前提下,新发展的 LSP 算法可以收敛到更优的最优值,从而得到更好的优化效果。下面通过气动算例来加以验证。

## 5 翼型优化设计

优化设计模型包括3个基本要素:目标函数,约束条件和设计变量。对于翼型气动优化设计而言,一般确定了设计点后可选取表示升阻特性或力矩特性的气动系数作为目标函数,对翼型厚度和面积等进行限制作为约束,将翼型的几何形状作为设计变量<sup>[6]</sup>。为了检验 LSP 算法对翼型的优化性能,下面我们将分别使用 CS 算法和 LSP 算法对 RAE2822 翼型进行单点减阻优化设计,并与 PSO 算法的优化结果进行对比。

设计状态为

$$Ma = 0.74, \alpha = 2.3^\circ, Re = 5.59 \times 10^6$$

$$\text{s.t.} : \{g_1(x) : c_m \geq c_{m0}\}$$

$$g_2(x) : t_{\max} \geq t_{\max0}$$

$$g_3(x) : s \geq s_0$$

$$g_4(x) : 98\%c_{l0} \leq c_l \leq 102\%c_{l0}$$

优化目标为最小化翼型阻力系数,约束条件为翼型的面积和最大厚度不减小,翼型的力矩系数不减小且保持升力系数基本不变(上下浮动2%)。翼型几何参数化采用改进的 Hicks-Henne 参数化方法<sup>[7-8]</sup>,选取14个设计变量来确定翼型,其中7个设计变量表示翼型厚度分布,另外7个设计变量表示翼型弯度分布。气动特性的分析计算采用 N-S 雷诺平均方程计算绕翼型流场,湍流模型为 S-A,网格采用非结构网格。

本节算例采用了标准 CS 算法、标准 PSO 算法以及 LSP 这3种优化算法。对于 LSP 算法,分别采用步长系数为 0.000 05 和 0.000 5 的算法策略,以检验不同的步长系数对 LSP 算法的优化性能影响。为了区分上述2种 LSP 算法,分别以 LSP-1 与 LSP-2 表示。此外,由于算法原理各有差异,为了保证调用气动计算的次数尽可能相同,从而保证相当的计算量,各算法参数设置如下所述:

标准 CS 算法,种群数量为 20,步长系数采用非线性递减,  $\alpha = 0.5 - 0.45/\maxgen^2 \times k^2$ ,优化代数设为 40 代;标准 PSO 算法,初始种群为 40,算法优化 39 代;LSP-1 算法,成熟个体的数量 4 个,每个成熟个体产生 25 个后代,优化代数设为 4 代,PSO 算法优化 10 代,步长系数为 0.000 05;LSP-2 算法,步长系数取 0.000 5,其他算法设置同 LSP-1;优化后翼型的气动特性比较如表4所示,算法优化的迭代收敛过程如图5所示,翼型优化几何形状如图6所示,翼型优化压力分布如图7所示。

表4 翼型气动特性对比

参数	初始	PSO	CS	LSP-1	LSP-2
$C_d$	0.017 6	0.013 93	0.013 84	0.013 88	0.013 7
$C_l$	0.686	0.686	0.685	0.685	0.686
$C_m$	-0.092 6	-0.089 5	-0.082 1	-0.084 3	-0.079 8
最大厚度	0.121	0.121	0.122	0.121	0.121
翼型面积	0.077 8	0.078 1	0.078 2	0.0784	0.077 9
CFD 调用次数		1 600	1 625	16 25	1 625

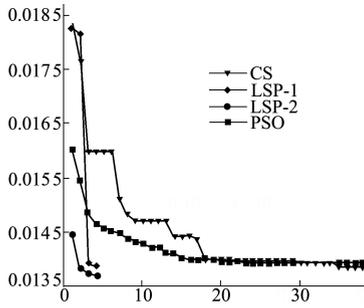


图5 翼型优化收敛曲线

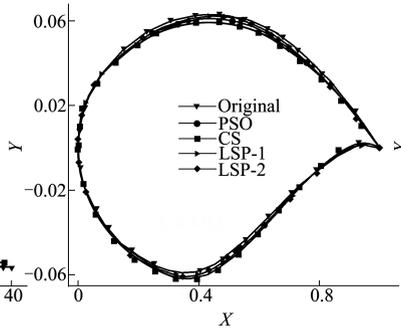


图6 翼型优化几何形状

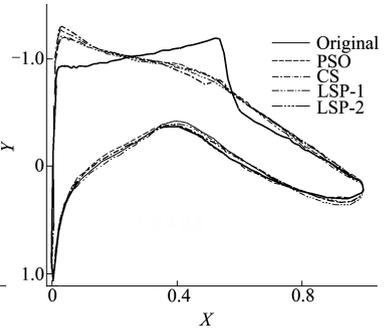


图7 翼型优化压力分

从表可以看出,步长系数为 0.000 5 的 LSP-2 算法可以收敛到比较接近全局最优的位置,优化后阻力系数降低了 22.16%;而与之相比,PSO、CS 以及步长系数为 0.000 05 的 LSP-1 算法优化后升阻比则分别降低了 20.85%、21.36%、21.13%。另外,图 5 则显示,相对于 PSO 算法以及 CS 算法,发展的 LSP 算法在较短的代数内就能够基本收敛,而且能够寻优到比较好的解位置处;图 6 显示优化后翼型的上表面顶点后移,最大厚度位置也稍微往后移动了一些;图 7 表明翼型的前加载增强,同时上边面的激波得以抹去。证明了发展的 LSP 算法在选取了合适的步长系数后,可以达到很好的优化效果。

## 6 结论

在本文的 CS 算法中,莱维飞行(Levy flights)搜索方式可以保证算法的收敛性,而且稳定性较好。但是,CS 算法收敛速度相对较慢,往往需要上千次

的优化代数才能得到较理想的优化结果,限制了算法的应用范围。SP 算法可以有效继承前代的优势信息,但是其搜索机制限制了收敛性,过大的收缩半径会导致搜索效率的降低,而过小的搜索半径则有可能导致陷入局部最优。本节将 CS 算法中的核心技术莱维搜索加入到改进的寄生算法中,便发展得到了 LSP 算法。

通过函数测试以及气动优化算例可以发现,发展的 LSP 算法通过将 CS 算法中的莱维搜索机制引入到寄生算法中,在保留成熟个体优势信息的同时,还能通过莱维搜索补充群体所需要的多样性,取得了比 CS、PSO、SP 算法更好的优化效果,达到了优化的目的。

在优化过程中,莱维搜索方式能较大程度上保证样本的多样性,不会轻易漏掉潜在的全局最优解。在最优化问题中可以考虑将莱维搜索与其他算法进行有机结合,从而有可能为算法提供更好的智能驱动力,提升算法的寻优性能。

## 参考文献:

- [1] 李丁. 智能优化算法及其在气动优化设计中的应用研究[D]. 西安:西北工业大学,2011  
Li Ding. Aerodynamic Optimization Design Based on Intelligent Algorithms[D]. Xi'an, Northwestern Polytechnical University, 2011 (in Chinese)
- [2] 王亮. 基于智能算法的气动外形优化研究[D]. 西安:西北工业大学,2015  
Wang Liang. Research on Aerodynamic Optimization Based on Swarm Intelligence Algorithm[D]. Xi'an; Northwestern Polytechnical University, 2015 (in Chinese)
- [3] Wang F, He X S, Wang Y, Yang S M. Markov Model and Convergence Analysis Based on Cuckoo Search Algorithm[J]. Comput Eng, 2012, 38(11): 180-185
- [4] Yang Xinshe, Deb S. Engineering Optimisation by Cuckoo Search[J]. International Journal of Mathematical Modeling and Numerical Optimization, 2010, 1(4): 330-343
- [5] 孙腾腾. 基于代理模型的群智能算法在气动优化设计中的应用[D]. 西安:西北工业大学,2014

Sun Tengting. Swarm Intelligence Algorithms Based on Surrogate Model for Aerodynamic Optimization Design [D]. Xi'an, Northwestern Polytechnical University, 2014 (in Chinese)

- [6] Ray T. Swarm Algorithm for Single-And Multiobjective Airfoil Design Optimization[J]. AIAA Journal, 2004, 42(2): 366-373
- [7] Hicks R M, Henne P A. Wing Design by Numerical Optimization[J]. Journal of Aircraft, 1978, 15(7): 407-412
- [8] 王建军,高正红. HicksHenne翼型参数化方法分析及改进[J]. 航空计算技术, 2010, 40(4): 46-49
- Wang Jianjun, Gao Zhenghong. Analysis and Improvement of Hicks Henne Airfoil Parameterization Method[J]. Aeronautical Computing Technique, 2010, 40(4): 46-49 (in Chinese)

## Applying the Levy Simulated Parasitism Algorithm to Airfoil Aerodynamic Optimization

Xia Lu, Li Lang, Wang Liang

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

**Abstract:** In order to balance the global and local search ability of intelligent optimization algorithm, the cuckoo search algorithm and simulated parasitic algorithm is discussed and analyzed. The mechanism of simulated parasitism model and cuckoo search is much alike, so the Levy flight search strategy of cuckoo search is introduced into the former model. So we get a optimization algorithm which is named Levy simulated parasitism algorithm. Function tests and aerodynamics problem prove that improved algorithm has better searching ability. Applying the Levy simulated parasitism algorithm to the airfoil aerodynamic optimization design, we achieve good results, which show that the proposed algorithm is effective and practical.

**Keywords:** the Levy flight search; the simulated parasitic algorithm; airfoil aerodynamic optimization design