

七星瓢虫优化算法及其在多学科协同优化中的应用

王 鹏¹ 李 洋¹ 王昆仑²

(西北工业大学航海学院 西安 710072)¹ (中国兵器工业试验测试研究院 华阴 714200)²

摘 要 基于对七星瓢虫觅食行为的研究,提出一种新的生物学启发式算法——七星瓢虫优化算法(Seven-spot Ladybird Optimization, SLO)。通过 6 个标准测试函数将其与现有的两种启发式算法(GA、PSO)进行比较,探知其寻优能力。在此基础上,针对多学科设计优化协同优化算法求解效率不高、鲁棒性差等缺陷,将 SLO 引入协同优化算法系统级优化器,通过工程实例验证了七星瓢虫优化算法良好的寻优能力。

关键词 七星瓢虫优化算法,多学科设计优化,协同优化算法,寻优能力

中图分类号 TP301.6 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.11.054

Seven-spot Ladybird Optimization and its Application in Multidisciplinary Collaborative Optimization

WANG Peng¹ LI Yang¹ WANG Kun-lun²

(School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China)¹

(Norinco Group Test and Measuring Academy, Huayin 714200, China)²

Abstract Based on the life habits of seven-spotted ladybird, we gave a new heuristic algorithm called seven-spot ladybird optimization(SLO). Compared with two existing heuristic algorithms, GA and PSO, its optimization ability was presented. In view of the defects of multidisciplinary design optimization (MDO), such as low solving efficiency, poor robustness, we took SLO to the system level optimization of collaborative optimization (CO) and verified the searching capability of SLO algorithm through a project example.

Keywords Seven-spot ladybird optimization, MDO, CO, Searching capability

1 前言

多学科设计优化方法(Multidisciplinary Design Optimization, MDO)是一种通过探索和利用工程系统中的相互作用的协同机制来实现复杂设计的方法论,用来解决复杂的多学科系统设计问题^[1]。其主要思想是在复杂设计过程中利用分布的计算机网络来集成多学科的知识,采用有效的设计优化策略,组织优化设计过程,从而能在综合考虑系统整体的情况下,获得设计最优解^[2]。

多学科设计优化以提高产品性能、缩短设计周期和降低研制成本为目的,可适应更大规模的工程产品系统设计发展的需要,是在传统设计方法和优化方法基础上的一个质的的发展,标志着国家产品创新技术与理论的制高点之一,具有重大的战略前沿意义^[3]。而优化算法作为 MDO 的核心部分,是最活跃的研究领域,其中协同优化算法是主要的研究内容之一。但标准的协同优化算法存在求解效率不高、鲁棒性差等缺点,为了解决这些问题,把启发式算法加入系统级优化器成为一种研究趋势^[4]。目前,已有多种算法(如遗传算法(GA)^[5]、粒子群算法(PSO)^[6]、蚁群算法(ACO)^[7]等)被加入到协同优化算法系统级优化器进行优化计算,取得了较好的效果。研究新的更适用于多学科协同优化的启发式算法已成

为一种必然的发展趋势^[8]。

本文根据七星瓢虫的捕食习性,提出了一种新的启发式算法:七星瓢虫优化算法。本文通过 6 个标准测试函数探知了该算法的寻优能力,同时用一个工程实例验证了该算法在低维时能通过较少的个体数快速找到最优解。

2 七星瓢虫优化算法

2.1 算法原理

七星瓢虫是一种常见的昆虫,因其两鞘翅上的 7 个黑斑点而得名,被人们称为“活农药”。而其因良好的捕食能力已经吸引了大量昆虫专家的兴趣^[9]。

通过对七星瓢虫捕食过程的观察发现,其在茎秆或叶片上寻找猎物时,总是用下颚须与下唇须不停地触碰茎秆与叶片,并沿着枝条与叶片往返爬行反复寻访,速度时快时慢,头部左右转动,触角与附着物呈平行前伸状态。而其侧向的振动可以扩大猎物可能存在的区域。如图 1 所示,捕食时瓢虫先通过相对线性而快速的广域搜索对猎物进行定位,猎食后即在该区域进行缓慢而迂回的局域搜索,以判断该区域是否为猎物聚集区,若是,则在该区域进行缓慢而迂回的集中搜索,反之,飞离该区域寻找新的目标。当瓢虫在区域的捕食率低于一个临界值,或者特定时间内依旧劳无所获,瓢虫就会飞

到稿日期:2014-11-27 返修日期:2015-03-08 本文受国家自然科学基金项目(51375389)资助。

王 鹏(1978—),男,博士,副教授,主要研究领域为水下飞行器总体设计、多学科设计、流体力学等, E-mail: wangpeng305@nwpu.edu.cn;

李 洋(1991—),女,硕士,主要研究领域为多学科优化设计、流体分析;王昆仑(1990—),男,工程师,主要研究领域为机械设计、武器工程。

离所在区域而前往别的区域,即迁徙。这样的搜索方式可以大大提高瓢虫的捕食率^[10]。七星瓢虫优化算法就是根据瓢虫的这种捕食习性产生的。

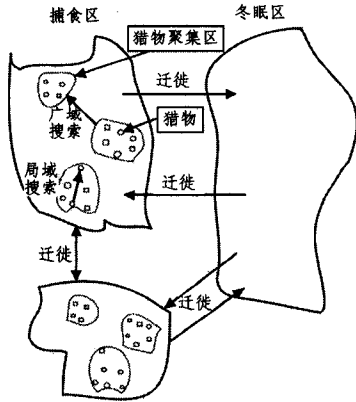


图1 七星瓢虫捕食过程

2.2 算法流程

Step1 分区:假设搜索空间(环境)是 D 维的。第 i 维空间被分为 n_i 个子空间,总的空间数即为 $n = \prod n_i$ 。

Step2 初始化种群:假设每只七星瓢虫都是 D 维空间中的一个点,第 i 只瓢虫被记作 $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$,其中 X_i 是问题潜在的一个解。

七星瓢虫的种群数量为 $N, N = m \times n$,其中 m 是每个子空间中七星瓢虫的数量。

Step3 计算适应度,表明个体或解的优劣性。

Step4 选择最优瓢虫:将空间中所有瓢虫的当前最佳位置分别与子空间历史最优($lbest$)、全局最优($gbest$)、个体历史最优($sbest$)进行比较。如果当前值优于历史最优值,则进行替换,否则保持不变。

Step5 迁徙:若一定循环后瓢虫的位置仍不改变,该位置即被舍弃,取而代之,根据式(1)在全局最优解的附近产生一个新的位置,这个新的位置能共享到全局最优信息。

$$x'_{i,j} = x_{gbest,j} + \phi\omega \quad (1)$$

其中, ω 为全局最佳解 $gbest$ 的邻域, ϕ 为 $[-1, 1]$ 间的随机数。

Step6 位置更新:瓢虫的位置随着之前的运动更新。如果瓢虫上次循环末进行了广域搜索,本次循环瓢虫将先进行局域搜索,位置更新公式如下:

$$V = c * r_1 * (S_i(t) - X_i(t)) + \epsilon_1 \quad (2)$$

$$X_i(t+1) = X_i(t) + V, |V| \leq V_{max} \quad (3)$$

反之,瓢虫先进行广域搜索,位置更新公式如下:

$$V = c * r_2 * (L_i(t) - X_i(t)) + \epsilon_2 \quad (4)$$

$$X_i(t+1) = X_i(t) + V, |V| \leq V_{max} \quad (5)$$

$S_i(t), L_i(t), X_i(t)$ 分别为个体最优位置、子空间最优位置、当前最优位置; r_1 和 r_2 为 0 到 1 范围内的两个随机数; ϵ_1 和 ϵ_2 通常设置为相对小的随机数; c 为常数,用于调整每次迭代的搜寻步数和搜寻方向。

一般情况下, V_{max} 设置如下:

$$V_{max} = 0.2(ub - lb) \quad (6)$$

其中, ub 和 lb 分别是搜寻区域的上下限。

在式(3)和式(5)中,每一维中瓢虫的速度都必须小于 V_{max} ,这决定了瓢虫在结果域中的搜寻精度。如果 V_{max} 太大,

瓢虫就会飞离最优解;如果 V_{max} 太小,瓢虫又会陷入局部搜索空间而无法继续全局搜索。

根据以上公式,可以得出速度更新由 3 部分组成:(1)局域搜索,由瓢虫缓慢而有条不紊的进行;(2)广域搜索,瓢虫沿着相对线性路径迅速的进行;(3)效仿瓢虫的振动性能,以扩大潜在结果的搜索区域。

Step7 检验终止条件:如果终止条件满足,即 SLO 达到了最大迭代数,那么 SLO 就会终止;否则,跳转至 Step3。

具体流程图如图 2 所示。

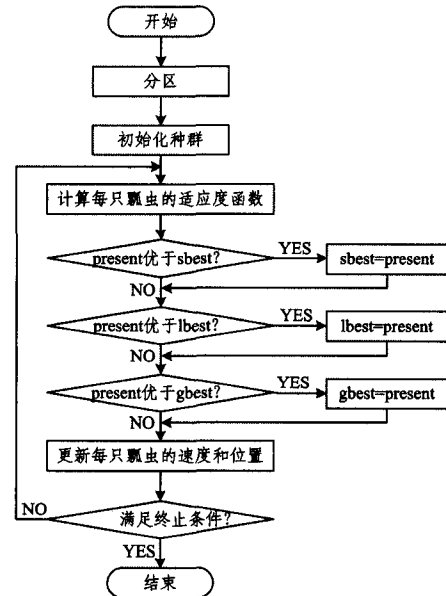


图2 七星瓢虫算法流程

2.3 算法论述

七星瓢虫优化算法是根据七星瓢虫的捕食习性产生的用于全局优化的仿生算法,相比较于根据鸟群捕食行为研究的粒子群算法,两者同为基于概率计算的随机搜索进化算法,在结构、研究内容、方法以及步骤上有一定的相似性。但 SLO 具有分区和广域与局域搜索相结合的特点,使其有不同于其他算法的特别之处。

类似于 PSO, SLO 也是通过迭代找到最优。不同的是, PSO 的每次迭代中,粒子通过跟踪“个体历史最优($sbest$)”和“全局最优($gbest$)”来更新自己的位置;而 SLO 在初始化种群前先分区,每个区域都存在“子空间历史最优($lbest$)”,个体要通过自身当前位置与 $sbest, lbest$ 和 $gbest$ 的先后比较来确定下一步的位置更新,减少了不必要的搜寻工作。相对线性而快速的广域搜索和缓慢而迂回的局域搜索的有效结合,可以确保搜寻结果的快速性和准确性,减少搜寻次数。

这种分区搜索的方式看似复杂,实则计算简单,不仅可以对搜索空间进行充分探索,还可以快速锁定最优解存在的区域,通过细致而针对性的搜索来精确找到目标,提高了寻优效率和寻优精度。相较于一些成熟的优化算法,如 PSO、GA, SLO 需要相对较少的初始个体数和迭代次数就能得到满意的结果。

3 算法性能测试

如表 1 所列,选择 6 个基准测试函数对 SLO 算法进行性能分析与评价。这些函数具有使优化过程非常困难的特性,

如多峰性、欺骗性、维度灾难等,可以有效地测试算法寻优精度和执行性能。在维数为 5、10、30 时,将函数在 SLO、PSO、GA 算法上的寻优结果进行比较,可以初步探知七星瓢虫优化算法的寻优能力。

表 1 测试函数信息表

函数名称	表达式	取值范围	极值
Rastrigin	$f(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$	0
Griewank	$f(x) = \sum_{i=1}^N \frac{x_i^2}{4000} - \prod_{i=1}^N \cos(x_i/\sqrt{i}) + 1$	$[-600, 600]$	0
Ackley	$f(x) = -20e^{(-0.2\sqrt{1/D \sum_{i=1}^{D-1} x_i^2})} - e^{(1/D \sum_{i=1}^{D-1} \cos(2\pi x_i))} + 20 + e$	$[-32, 32]$	0
Sphere	$f(x) = \sum_{i=1}^D x_i^2$	$[-100, 100]$	0
Rosenbrock	$f(x) = \sum_{i=1}^D [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]$	0
Schwefel	$f(x) = D * 418.9829 + \sum_{i=1}^D -x_i \sin(\sqrt{ x_i })$	$[-500, 500]$	0

SLO 参数设置如表 2 所列,其他 2 种算法随机种群数为 50,不同维数下的迭代次数和重复次数与 SLO 保持一致。具体测试结果见表 3—表 8。

表 2 SLO 参数设置

维数	随机种群数	分区数	迭代次数	重复次数
5	20	2	750	30
10	20	3	1000	30
30	20	5	1500	30

表 3 Rastrigin 函数求解结果

算法	维数	平均值	最优值	方差
SLO	5	0.0000E+00	0.0000E+00	0.0000E+00
	10	2.5555E+01	1.4961E+01	3.1011E+01
	30	3.3542E+02	2.2711E+02	8.9180E+02
PSO	5	9.9720E-02	0.0000E+00	8.8629E-02
	10	1.4075E+00	0.0000E+00	1.0174E+00
	30	2.8344E+01	1.8911E+01	1.6821E+01
GA	5	5.4896E-01	2.9504E-10	5.1084E-01
	10	1.8558E+00	1.8936E-07	2.2209E+00
	30	5.5024E+00	2.0088E+00	4.7370E+00

表 4 Griewank 函数求解结果

算法	维数	平均值	最优值	方差
SLO	5	1.6036E-01	3.7000E-02	4.4400E-03
	10	4.2589E-01	8.2500E-02	4.6778E-03
	30	1.4023E+00	9.5100E-01	5.1337E-01
PSO	5	2.2110E-02	7.4000E-03	1.3300E-04
	10	8.7503E-02	3.2000E-02	1.0001E-03
	30	1.8219E-02	6.6429E-04	2.0776E-04
GA	5	1.2240E+01	1.2240E+01	0.0000E+00
	10	1.4319E+01	5.8630E+00	2.0263E+01
	30	1.5553E-02	1.2152E-04	2.0429E-04

表 5 Ackley 函数求解结果

算法	维数	平均值	最优值	方差
SLO	5	-8.8818E-16	-8.8818E-16	0.0000E+00
	10	1.5811E-06	0.0000E+00	6.9612E-11
	30	1.5927E-07	8.4573E-10	1.2999E-13
PSO	5	1.9540E-15	-8.8818E-16	2.7686E-30
	10	4.3224E-15	2.6645E-15	3.0401E-30
	30	1.8166E-04	1.8786E-05	5.44066E-08
GA	5	2.4256E-05	1.0644E-05	3.7938E-10
	10	6.1032E-05	1.9088E-05	5.7124E-10
	30	6.2545E-03	1.1000E-03	2.7986E-04

表 6 Sphere 函数求解结果

算法	维数	平均值	最优值	方差
SLO	5	7.4322E-14	6.3046E-19	3.3010E-26
	10	3.1187E-06	3.3181E-12	2.7230E-10
	30	6.7848E+02	3.0870E+02	8.3183E+04
PSO	5	8.7700E-03	0.0000E+00	4.8500E-04
	10	3.0342E-13	8.8199E-17	5.7529E-25
	30	1.7059E+01	2.2696E+00	1.6419E+02
GA	5	1.7019E-10	3.0910E-12	4.5910E-20
	10	1.8790E-09	1.4700E-10	1.7167E-18
	30	1.2739E-02	1.7168E-03	1.2811E-04

表 7 Rosenbrock 函数求解结果

算法	维数	平均值	最优值	方差
SLO	5	1.3047E+00	4.7700E-01	1.71946E+00
	10	1.2993E+01	6.1600E-02	3.0678E+01
	30	3.5754E+04	1.0591E+04	5.0569E+04
PSO	5	2.4214E-01	7.4530E-04	7.0911E-02
	10	2.0050E+00	5.6000E-03	1.8968E+00
	30	4.3202E+01	4.8091E+00	6.4895E+01
GA	5	8.3220E-02	6.1000E-03	1.1840E-03
	10	7.1685E-01	7.7000E-02	4.4133E-01
	30	2.3103E+01	4.5300E-02	4.1803E+01

表 8 Schwefel 函数求解结果

算法	维数	平均值	最优值	方差
SLO	5	3.2512E+02	6.6309E-05	1.7288E+02
	10	1.2532E+03	7.5071E+02	2.7907E+02
	30	5.8721E+03	4.7153E+03	7.6123E+02
PSO	5	2.9647E+02	6.3689E-05	1.3131E+02
	10	2.6723E+02	1.2755E-04	1.6781E+02
	30	2.8631E+03	3.8228E-04	4.0473E+02
GA	5	2.0812E+03	2.0811E+03	5.2391E-11
	10	4.1515E+03	4.1516E+03	9.4903E-11
	30	1.2539E+04	1.2573E+04	7.3220E+10

通过对测试结果的观察分析不难发现:

(1)在维数较低(5 维)时,相比较其他两种算法,SLO 得到的结果更接近目标值。同时,从方差可以看出,30 次重复求解 SLO 得到的结果更加稳定,尤其是对典型的具有大量局部最优点的复杂多峰函数 Rastrigin 的求解。而对于 Ackley 函数,值得一提的是,其各维下的求解结果均优于其他两种算法。另外几个测试函数的结果也显示,3 种算法的寻优能力不分伯仲。可见,SLO 和 PSO、GA 一样具有良好的寻优能力。

(2)由于 SLO 选择的随机种群数为 20,其他 2 种算法随机种群数均为 50,因此通过比较 3 种算法选取的随机种群数差别可以发现,SLO 算法能用相对较小的种群数量找到全局最优解。

4 工程实例验证

本文选取的工程实例是 NASA 评估 MDO 优化过程算法性能的 10 个标准算例之一^[1]。该算例是一个齿轮减速箱的优化问题,其设计目标是使减速箱的体积(或重量)最小,并满足齿的弯曲应力和接触应力及轴的弯转变形等要求。减速器的结构如图 3 所示。

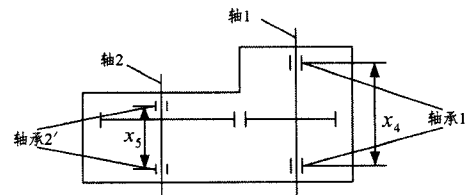


图 3 减速器结构图

对于该减速器优化问题,具体的数学模型和优化思想可描述如下。

(1)设计变量及取值:

齿宽 $x_1: 2.6 \leq x_1 \leq 3.6$

齿轮模数 $x_2: 0.7 \leq x_2 \leq 0.8$

小齿轮齿数 $x_3: 17 \leq x_3 \leq 28$

轴承 1 间距离: $7.3 \leq x_4 \leq 8.3$

轴承 2 间距离: $7.3 \leq x_5 \leq 8.3$

轴 1 直径: $2.9 \leq x_6 \leq 3.9$

轴 2 直径: $5.0 \leq x_7 \leq 5.5$

(2)目标函数:

该问题的设计目标是使减速箱的体积(或重量)最小,即:

$$\text{Min } f(x) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.477(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \quad (7)$$

(3)约束条件

齿轮减速箱的设计需要满足齿的弯曲应力 g_1 和接触应力 g_2 的设计要求;轴的扭转变形、应力约束以及由经验得到的约束($g_3 - g_8$);同时,还应满足几何空间约束($g_9 - g_{11}$)。

$$\begin{aligned} \text{s. t. } g_1 &= 27/(x_1x_2^2x_3) - 1 \leq 0 \\ g_2 &= 397.5/(x_1x_2^2x_3^2) - 1 \leq 0 \\ g_3 &= 1.93x_3^4/(x_2x_3x_4^3) - 1 \leq 0 \\ g_4 &= 1.93x_3^3/(x_2x_3x_4^2) - 1 \leq 0 \\ g_5 &= 10x_6^{-3}\sqrt{(745x_2^{-1}x_3^{-1}x_4)^2 + 1.69 \times 10^7} - 1100 \leq 0 \\ g_6 &= 10x_7^{-3}\sqrt{(745x_2^{-1}x_3^{-1}x_5)^2 + 1.575 \times 10^8} - 850 \leq 0 \\ g_7 &= x_2x_3 - 40 \leq 0 \\ g_8 &= x_1/x_2 - 12 \leq 0 \\ g_9 &= 5 - x_1/x_2 \leq 0 \\ g_{10} &= (1.5x_6 + 1.9)/x_4 - 1 \leq 0 \\ g_{11} &= (1.1x_7 + 1.9)/x_5 - 1 \leq 0 \end{aligned} \quad (8)$$

该算例在 $X = (3.5, 0.7, 17, 7.3, 7.71, 3.35, 5.29)$ 有最优解 $f(x)_{\min} = 2994^{[11]}$ 。

采用协同优化的思想,将该问题分解为 1 个系统级和并行的 3 个学科级。

系统级(sys):

$$\text{Min } \text{sys} = \text{sub1} + \text{sub2} + \text{sub3}$$

$$\begin{aligned} \text{s. t. } \text{sub1}_f &= (\text{sys}_x1 - \text{sub1}_x1)^2 + (\text{sys}_x2 - \text{sub1}_x2)^2 \\ &\quad + (\text{sys}_x3 - \text{sub1}_x3)^2 = 0 \\ \text{sub2}_f &= (\text{sys}_x1 - \text{sub2}_x1)^2 + (\text{sys}_x2 - \text{sub2}_x2)^2 \\ &\quad + (\text{sys}_x3 - \text{sub2}_x3)^2 = 0 \\ \text{sub3}_f &= (\text{sys}_x1 - \text{sub3}_x1)^2 + (\text{sys}_x2 - \text{sub3}_x2)^2 \\ &\quad + (\text{sys}_x3 - \text{sub3}_x3)^2 = 0 \end{aligned} \quad (9)$$

学科级(sub1):

$$\begin{aligned} \text{Min } \text{sub1}_f &= (\text{sys}_x1 - \text{sub1}_x1)^2 + (\text{sys}_x2 - \text{sub1}_x2)^2 + (\text{sys}_x3 - \text{sub1}_x3)^2 \\ \text{s. t. } \text{sub1} &= 0.7854 \times \text{sub1}_x1 \times (\text{sub1}_x2)^2 \times \\ &\quad (3.3333 \times \text{sub1}_x3)^2 + 14.9334 \times \\ &\quad \text{sub1}_x3 - 43.0934 \\ g_j, j &= 1, 2, 7, 8, 9 \end{aligned} \quad (10)$$

学科级(sub2):

$$\begin{aligned} \text{Min } \text{sub2}_f &= (\text{sys}_x1 - \text{sub2}_x1)^2 + (\text{sys}_x2 - \text{sub2}_x2)^2 + (\text{sys}_x3 - \text{sub2}_x3)^2 \\ \text{s. t. } \text{sub2} &= -1.508 \times \text{sub2}_x1 \times (\text{sub2}_x6)^2 + \\ &\quad 7.477 \times (\text{sub2}_x6)^3 + 0.7854 \times \\ &\quad \text{sub2}_x4 \times (\text{sub2}_x6)^2 \\ g_j, j &= 1, 2, 4, 6, 7, 8, 9, 11 \end{aligned} \quad (11)$$

学科级(sub3):

$$\begin{aligned} \text{Min } \text{sub3}_f &= (\text{sys}_x1 - \text{sub3}_x1)^2 + (\text{sys}_x2 - \text{sub3}_x2)^2 + (\text{sys}_x3 - \text{sub3}_x3)^2 \\ \text{s. t. } \text{sub3} &= -1.508 \times \text{sub3}_x1 \times (\text{sub3}_x7)^2 + \\ &\quad 7.477 \times (\text{sub3}_x7)^3 + 0.7854 \times \\ &\quad \text{sub3}_x5 \times (\text{sub3}_x7)^2 \\ g_j, j &= 1, 2, 3, 5, 7, 8, 9, 10 \end{aligned} \quad (12)$$

通过 Matlab 编程实现优化过程,其中系统级分别采用 SLO、SQP、PSO 和 GA 算法,学科级都用 SQP 算法,取表 9 中 4 组不同的初始点,结果对比如表 10 所列。

表 9 工程实例起始设计点

起始点	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
可行域内的设计点	X1	2.80	0.71	25.0	7.90	7.599	3.00	5.099
	X2	2.65	0.63	18.0	6.80	6.400	3.00	5.099
	X3	3.50	0.70	17.0	7.30	7.715	3.35	5.287
可行域外的设计点	X4	2.00	0.40	9.0	4.00	4.500	1.50	3.000

表 10 工程实例结果对比

SLO		SQP		PSO		GA		
初始值	最优解	函数值	最优解	函数值	最优解	函数值	最优解	函数值
(2.8, 0.71, 25, 7.9, 7.599, 3.0, 5.099)	(3.43, 0.71, 17.0, 7.5, 7.73, 3.35, 5.31)	2881.6	无可行解	—	(3.41, 0.68, 17.0, 7.9, 7.79, 3.35, 5.36)	2884.7	(3.45, 0.69, 17.0, 7.7, 7.73, 3.35, 5.29)	2890.1
(2.65, 0.63, 18, 6.8, 6.400, 3.0, 5.099)	(3.48, 0.71, 17.0, 7.5, 7.71, 3.35, 5.31)	2890.3	(3.34, 0.67, 18.0, 6.9, 7.72, 3.35, 5.29)	2882.2	(3.41, 0.68, 17.0, 7.9, 7.71, 3.35, 5.29)	2888.0	(3.43, 0.69, 17.0, 7.8, 7.73, 3.35, 5.29)	2889.3
(3.5, 0.7, 17, 7.3, 7.715, 3.35, 5.287)	(3.46, 0.70, 17.0, 7.4, 7.71, 3.35, 5.30)	2895.8	(3.49, 0.69, 16.9, 7.3, 7.72, 3.35, 5.29)	2882.3	(3.41, 0.68, 17.0, 8.1, 7.72, 3.35, 5.29)	2892.2	(3.45, 0.68, 17.0, 7.5, 7.72, 3.35, 5.30)	2888.7
(2.0, 0.4, 9.0, 4.0, 4.5, 1.5, 3.0)	(3.46, 0.70, 17.0, 7.4, 7.71, 3.35, 5.30)	2893.4	无可行解	—	(3.51, 0.70, 17.0, 7.3, 7.72, 3.39, 5.29)	2879.6	(3.45, 0.70, 17.0, 7.4, 7.73, 3.35, 5.32)	2877.5

从工程实例的结果对比可以看出,SLO、PSO 和 GA 均能找到函数的最小值和最小值点,SQP 能不能找到最优解由初始值决定。对于第一个初始点,SLO 的寻优结果不如 PSO 和

GA,但其他初始点所找到的最优解却更接近目标函数值。总的来说,把 SLO 作为系统级建立多学科协同优化模型能得到

(下转第 304 页)

- Tang Min, Chen Feng. Comparison of microscopy image denoising effects based on contourlet, curvelet and wavelet transform [J]. *Journal of Clinical Rehabilitative Tissue Engineering Research*, 2011, 15(22): 4094-4097
- [17] Do M N, Vetterli M. The contourlet transform: an efficient directional multiresolution image representation[J]. *IEEE Transactions on Image Processing*, 2005, 14(12): 2091-2106
- [18] 焦李成, 侯彪, 王爽, 等. 图像多尺度几何分析理论与应用: 后小波分析理论与应用[M]. 西安: 西安电子科技大学出版社, 2008: 280-289
- Jiao Li-cheng, Hou Biao, Wang Shuang, et al. Image multiscale geometric analysis: theory and applications beyond wavelets[M]. Xi'an: Xi'an University of Electronic Science and Technology Press, 2008: 280-289
- [19] Lu Yue, Do M N. A new contourlet transform with sharp frequency localization[C]// 2006 IEEE International Conference on Image Processing. IEEE, 2006: 1629-1632
- [20] 蒋媛. 基于多尺度几何分析的 SAR 图像去噪和融合[D]. 成都: 电子科技大学, 2010
- Jiang Yuan. SAR Image denoising and fusing based on multiscale geometric analysis[D]. Chengdu: University of Electronic Science and Technology of China, 2010
- [21] Zhou Jian-ping, Cunha A L, Do M N. Nonsubsampled contourlet transform: construction and application in enhancement [C]// IEEE International Conference on Image Processing, 2005 (ICIP 2005). IEEE, 2005: 469-472
- [22] 张宗念, 李金徽, 黄仁泰. 迭代硬阈值压缩感知重构算法——IIHT[J]. *计算机应用*, 2011, 31(8): 2123-2125
- Zhang Zong-nian, Li Jin-hui, Huang Ren-tai. IIHT: New improved iterative hard thresholding algorithm for compressive sensing[J]. *Journal of Computer Applications*, 2011, 31(8): 2123-2125
- [23] Herrity K K, Gilbert A C, Tropp J A. Sparse approximation via iterative thresholding[C]// ICASSP 2006 Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. IEEE, 2006: 624-627
- [24] 吴巧玲, 倪林, 何德龙. 基于非下采样 Contourlet 变换的压缩感知图像重建[J]. *中国科学技术大学学报*, 2012, 42(2): 87-91
- Wu Qiao-ling, Ni Lin, He De-long. Compressed sensing based on nonsubsampled contourlet transform for image reconstruction [J]. *Journal of University of Science and Technology of China*, 2012, 42(2): 87-91

(上接第 269 页)

相对较好的结果, 以后可将其用于解决与本算例类似的一些工程实例。

结束语 本文根据七星瓢虫的捕食习性提出了一种新的启发式算法: 七星瓢虫优化算法, 并对其算法原理与具体流程做了详细介绍。为了测试该算法的性能, 选择了 6 个标准测试函数进行优化计算, 对比结果发现该算法较于其他两种常用的启发式算法 GA 和 PSO 具有良好的寻优能力。同时, 针对传统多学科协同优化方法中存在的缺陷, 将该算法引入系统级优化器, 通过工程实例验证了该算法良好的寻优能力。比较发现, 该算法不仅在低维时具有相对更准确的寻优能力, 而且能用相对较小的种群数量找到全局最优解, 具有较高的计算效率和计算精度。但该算法的计算过程相对复杂, 这是今后需要研究和改进的方向。鉴于该算法的这些特点, 以后可以将其应用于维数相对较低、初始种群数相对较少的相关优化问题。

参 考 文 献

- [1] Ning A, Kroo I. Multidisciplinary considerations in the design of wings and wing tip devices[J]. *Journal of Aircraft*, 2010, 47(2): 534-543
- [2] Roshanian J, Ebrahimi M, Taheri E, et al. Multidisciplinary design optimization of space transportation control system using genetic algorithm[J]. *Proceedings of the Institution of Mechanical Engineers, Part G, Journal of Aerospace Engineering*, 2014, 228(4): 518-529
- [3] Simpson T W, Martins J R R A. Multidisciplinary design optimization for complex engineered systems: report from a national science foundation workshop[J]. *Journal of Mechanical Design*, 2011, 133(10): 1490-1495
- [4] Li Y, Jiang Z, Meng P, et al. A Collaborative Optimization Method for Solving Multi-objective Programming Problem[J]. *International Journal of Advancements in Computing Technology*, 2013, 5(1): 809
- [5] 王平, 郑松林, 吴光强. 基于协同优化和多目标遗传算法的车身结构多学科优化设计[J]. *机械工程学报*, 2011, 47(2): 102-108
- Wang P, Zheng S L, Wu G Q. Multidisciplinary design optimization of vehicle body structure based on collaborative optimization and multi-objective genetic algorithm[J]. *Journal of Mechanical Engineering*, 2011, 47(2): 102-108
- [6] Li Ying, Wang Jing-sheng, Wei Li-xin. Collaborative optimization based on particle swarm optimization and chaos searching [C]// Control Conference (CCC), 2012 31st Chinese. IEEE, 2012: 2427-2431
- [7] 薛锋, 王慈光, 牟峰. 基于信息熵和混沌理论的遗传-蚁群协同优化算法[J]. *控制与决策*, 2011, 26(1): 44-48
- Xue F, Wang C G, Mu F. Genetic and ant colony collaborative optimization algorithm based on information entropy and chaos theory[J]. *Control and Decision*, 2011, 26(1): 44-48
- [8] 周奇, 张立丽, 许辉, 等. 改进的多学科协同优化算法及其应用[J]. *计算机与数字工程*, 2014, 42(1): 65-68
- Zhou Q, Zhang L L, Xu H, et al. An improved collaborative optimization algorithm and its application[J]. *Computer & Digital Engineering*, 2014, 42(1): 65-68
- [9] Vantaux A, Roux O, Magro A, et al. Evolutionary perspectives on myrmecophily in ladybirds[J]. *Psyche: A Journal of Entomology*, 2012, 2012: 1-7
- [10] Pettersson J, Ninkovic V, Glinwood R, et al. Foraging in a complex environment-semiochemicals support searching behaviour of the seven spot ladybird[J]. *European Journal of Entomology*, 2005, 102(3): 365-370
- [11] Padula S L, Alexandrov N, Green L L. MDO test suite at NASA Langley Research Center [C]// Sixth AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization. 1996: 410-420