

doi:10.11835/j.issn.1000-582X.2020.226

混合果蝇算法及其在组合优化中的应用

李彦苍^a, 韩沐轩^b

(河北工程大学 a.水利水电学院; b.土木工程学院, 河北 邯郸 056038)

摘要:为改善基本果蝇算法易陷入局部最优及早熟的缺陷,利用一种改进的果蝇算法来进行优化,利用免疫算法自我-非自我的抗原识别机制及免疫系统学习-记忆-遗忘的知识处理机制提高算法的搜索能力及算法精度。改进算法将在果蝇算法执行后期引入免疫反应,通过产生不同抗体来增强种群多样性,跳出局部最优。通过数值仿真及实际案例的对比结果表明,改进算法的寻优表现更加良好,为算法优化提供一种有效可行的方法和思路。

关键词:果蝇算法;改进;免疫反应;混合算法;组合优化

中图分类号:TP18

文献标志码:A

文章编号:1000-582X(2020)09-118-13

Mixed fruit fly algorithm and its application in combinatorial optimization

LI Yancang^a, HAN Muxuan^b

(a. College of Water Conservancy and Hydro-Electric Power; b. College of Civil Engineering, Hebei University of Engineering, Handan 056038, Hebei, P. R. China)

Abstract: To overcome the defects of local optimum and precocity in basic fruit fly algorithm, an improved drosophila algorithm was presented in this paper for optimization. The basic idea is to improve the searching ability and accuracy of the algorithm by using the self-non-self antigen recognition mechanism of immune algorithm and the knowledge processing mechanism of learning-memory-forgetting in immune system. The immune response is introduced at the later stage of fruit fly algorithm implementation and the population diversity is enhanced by producing different antibodies to jump out of the local optimum. The results of numerical simulation and practical cases show that the improved algorithm performs better, and it provides an effective and feasible method and idea for algorithm optimization.

Keywords: fruit fly algorithm; improvement; immune response; hybrid algorithm; combinatorial optimization

受果蝇觅食过程的启发,台湾学者 Pan^[1]于 2012 年提出了较为高效的群智能优化算法——果蝇算法(FOA, fruit fly optimization algorithm)。由于该算法原理清晰、参数较少、操作简单,受到广泛关注。学者通过深入研究将其运用到预测、物流、调度、结构设计等领域,得到了较高的认可度。该算法同样存在不足^[2-3]:1)标准果蝇算法无法求解最优值为负的问题;2)解决复杂、高维和非线性优化问题的能力较弱;3)易陷入局部极值点等。

针对上述典型问题,不少学者对标准果蝇算法进行了改进。在候选解产生方面:Pan^[4]在味道浓度判定

收稿日期:2019-11-19 **网络出版时间:**2019-11-27

基金项目:国家自然科学基金资助项目(11202062);河北省高等学校科学技术研究资助项目(ZD2019114)。

Supported by National Natural Science Foundation of China (11202062) and Project of Scientific Research Program of Colleges and Universities in Hebei Province (ZD 2019114).

作者简介:李彦苍(1974—),男,博士,教授,主要从事计算智能及其应用研究,(E-mail) liyancang@hebeu.edu.cn。

值中加入可以为负的逃逸参数,使得候选解能够取到负值。在搜索半径方面:Pan 等^[5]引入自适应参数来调节果蝇的搜索半径,使之能够较好地平衡全局搜索能力及局部搜索能力。在飞行策略方面:Wang^[6]、Li^[7]、张前图等^[8]加入了群体协作和随机摄动的操作,进而解决算法的早熟问题。在种群多样性方面:刘志雄等^[9]将果蝇群体分成多个相同规模的子种群、韩俊英等^[10]提出了一种动态双子群协同进化 FOA,提升搜索精度。

研究通过设计混合算法:融合免疫反应^[11-13]的混合果蝇算法(IAFOA, fruit fly optimization algorithm based on immune algorithm)对标准果蝇算法进行性能改进。充分利用免疫算法来改善果蝇算法后期易陷入局部极值的不足,即当进化停滞步数 t 大于进化停滞步数阈值 T 时,执行免疫操作^[14-15]来克服基本果蝇算法的缺陷。通过标准函数及求解 0~1 背包问题的测试验证改进算法具有较好的鲁棒性及智能性^[16]。最后,将其应用于桁架结构优化问题,通过对比其他算法验证 IAFOA 的可行性。

1 基本果蝇算法

果蝇算法(FOA, fruit fly optimization algorithm)是一种新兴启发式算法,模拟自然界果蝇的觅食活动寻求目标函数的最优解,果蝇觅食示意图,如图 1 所示。其基本步骤如下^[17-19]

步骤 1:初始化参数。设置种群规模 $S_{Sizepop}$ 和最大迭代次数 M_{Maxgen} ,初始化种群位置

$$(X_{axis}, Y_{axis})。 \tag{1}$$

步骤 2:赋予果蝇个体觅食的随机方向及距离

$$\begin{cases} X_i = X_{axis} + R_{value}, \\ Y_i = Y_{axis} + R_{value} \end{cases} \tag{2}$$

步骤 3:起初不能获知食物源位置,先计算果蝇个体与原点之间的距离 D_i ,再计算味道浓度判定值 S_i ,

$$D_i = \sqrt{X_i^2 + Y_i^2}, \tag{3}$$

$$S_i = 1/D_i。 \tag{4}$$

步骤 4:将 S_i 带入味道浓度判定函数(fitness function)求出果蝇个体位置的味道浓度 T_i ,

$$T_i = \text{Fitness}(S_i)。 \tag{5}$$

步骤 5:找出果蝇群体中味道浓度最高的个体为

$$[B_{BestTaste}, B_{BestIndex},] = \min(T_i)。 \tag{6}$$

步骤 6:保留最佳味道浓度值与坐标,种群中其他个体向该位置飞去,

$$T_{Best} = B_{BestTaste}, \tag{7}$$

$$\begin{cases} X_{axis} = X_{bestIndex}, \\ Y_{axis} = Y_{bestIndex} \end{cases} \tag{8}$$

步骤 7:终止条件,判断最佳位置浓度是否优于前一代,并且达到最大迭代次数,否则跳转步骤 2 进入迭代寻优。

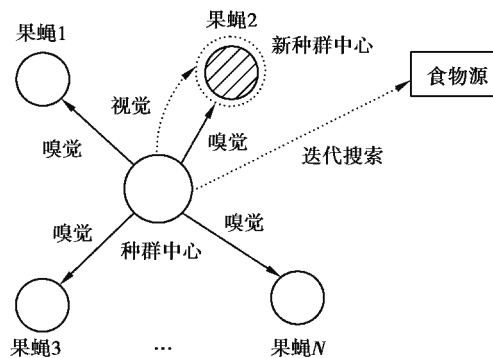


图 1 果蝇觅食示意图

Fig. 1 Schematic diagram of fruit fly foraging

2 融合免疫反应的混合果蝇算法

果蝇具有较发达的嗅觉和视觉系统,在觅食过程中先通过嗅觉进行大范围搜索,并向周围的果蝇发出食物气味信息。当果蝇通过比较发现具有较高浓度的位置时,果蝇个体将依靠视觉功能飞向该位置。正是由于这样的种群特性,降低了种群多样性,使模拟其特性的果蝇算法同其他仿生智能算法一样具有容易陷入局部最优的缺陷,带来早熟问题。

免疫算法从体细胞理论和网络理论中得到启发,实现了与免疫系统类似的自我调节和生成不同抗体的功能。该算法具有很强的局部搜索能力,将其引入果蝇算法执行过程后期,提出融合免疫反应的混合果蝇算法 IAFOA,新算法 IAFOA 可以用来平衡果蝇算法易陷入局部最优的不足,提高搜索效率。2 种标准算法对应关系如表 1 所示。

表 1 IA 和 FOA 在求解组合优化问题时的对应关系

算法	自变量	目标函数	优化解	解与目标函数的匹配程度
IA	免疫因子	抗原	抗体	抗原和抗体之间的亲和性
FOA	果蝇	食物源	果蝇位置	果蝇位置与食物源的接近程度

2.1 免疫算法

免疫算法(IA, immune algorithm)是一种仿生优化类算法,该研究最初是从 20 世纪 80 年代中期的免疫学研究发展而来。1990 年,Bersini^[20]首次使用免疫算法来解决问题。IA 算法通过模拟生物免疫系统功能进行抗原(目标函数)的识别,对免疫系统中的存储记忆原理的模拟,抗原和抗体(优化解)的结合,以及免疫系统自身多样性的模仿,实现了类似于免疫系统的抗原识别、细胞分化、记忆和自我调节的功能^[21]。其基本步骤如下:

步骤 1:抗原识别。输入目标函数和约束条件作为免疫算法的抗原。

步骤 2:产生初始抗体。在解空间范围内随机产生初始抗体。

步骤 3:计算亲和性(适应度评价)。按给定的适应度评价函数判定各个抗体与抗原的亲和性及抗体和抗体之间的亲和性。

抗原与抗体 V 之间的亲和性 A_v 定义为

$$A_v = 1/[1 + O_{ptv}], \quad (9)$$

式中: O_{ptv} 表示抗原和抗体的匹配程度; A_v 的值介于 0~1 之间。当 $O_{ptv} = 0$ 时, $A_v = 1$,这表明抗体和抗原非常匹配,即该抗体为最优的解。

步骤 4:记忆单元的更新。将步骤 3 计算出的与抗原亲和性最高的抗体加入记忆单元,并取代原有抗体。

步骤 5:促进和抑制节的产生。计算抗体 i 的期望值 E_{xi} ,期望值低的抗体将被抑制。

$$E_{xi} = A_i/C_i, \quad (10)$$

式中: A_i 是抗原与抗体 i 的亲和性; C_i 是抗体 i 的数目。

步骤 6:产生新抗体。父代通过遗传、变异、交叉产生下一代抗体。

步骤 7:是否满足终止条件。是,终止算法;否,跳转步骤 3。

2.2 IAFOA 种群多样性改进

免疫系统是通过阻拦细菌入侵维持生物正常代谢的基本防御系统。该系统通过识别基因类型以产生不同的抗体,通过调节机制促进新个体的出现、抑制个体的过多产生,从而实现生物的多样性^[22]。

设免疫系统有 N 个抗体,每个抗体有 M 个基因,如图 2 所示。第 j 个基因的信息熵 $H_j(N)$ 为

$$H_j(N) = \sum_{i=1}^N (-P_{ij} \times \log P_{ij}). \quad (11)$$

如果在位置 j 上所有抗体的等位基因都相同,那么 $H_j(N)$ 等于 0。
故,系统的平均信息熵 $H(N)$ 为

$$H(N) = \sum_{j=1}^M H_j(N)/M. \tag{12}$$

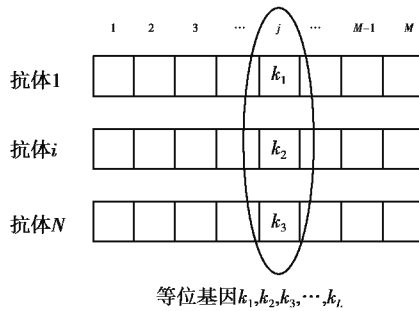


图 2 基因的信息熵

Fig. 2 Information entropy of genes

2.3 IAFOA 算法的执行

算法以标准果蝇算法为基本框架,调用 FOA 算法,构造各次迭代的可行解集,然后采用进化停滞步数 t 作为触发条件,当 $t > T$ (进化停滞步数阈值) 时,调用 IA 搜索过程,将免疫算法中免疫因子(对应果蝇个体)寻求抗原(对应果蝇算法中的食物源)产生抗体(优化解)的过程用于扩大搜索的空间。对经过 IA 搜索后获得的可行解集执行精英保留策略,得到的优化解与原可行解集共同用于更新搜索空间中气味信息的浓度,以指导其他果蝇的路径寻找机制。

其中,进化停滞步数阈值 T 为进入 IA 算法的指标,过早引入 IA 算法,不利于 IA 的搜索能力,甚至无法收敛可行解;过晚进入 IA 算法,不利于提高计算效率。通过多次独立试验取值,研究采用 $T=3$ 作为进入 IA 算法的触发条件值。

如满足触发条件,则以某一固定概率 p 将果蝇重新分配到寻优空间中,并对不同个体根据其适应度值赋予不同的自适应免疫概率 $P(i)$,通过 $P(i)$ 选出需进行免疫操作的果蝇个体,执行免疫操作,进而迭代搜索得优化解放入原优化解集进行比较。

$$P(i) = \frac{(T_{\text{Best}} - T_i)}{(T_{\text{Best}} - T_{\text{Worst}})p}, \tag{13}$$

其中,为避免算法陷入局部最优解,给定 IA 算法中免疫因子一定的初始概率 $p=0.25$ 使其随机分配,增加解的多样性。

IAFOA 的流程如下

步骤 1:初始化参数。设置种群规模 S_{Sizepop} 和最大迭代次数 M_{Maxgen} ,初始化种群位置 X_{axis} 和 Y_{axis} ,进化停滞步数 $t=0$ 。

步骤 2:根据式(3)随机产生果蝇群体。

步骤 3:采用式(4)~式(7)对种群进行操作。

步骤 4:根据式(14)记录并保留最佳味道浓度值,更新进化迭代步数 t 。

$$\left\{ \begin{array}{l} \text{if}(B_{\text{BestTaste}} < T_{\text{Best}}) \\ \quad T_{\text{Best}} = B_{\text{BestTaste1}} \\ \text{else}(t = t + 1) \\ \quad T_{\text{Best}} = B_{\text{BestTaste2}} \\ \quad \text{end} \end{array} \right. \tag{14}$$

步骤 5:判断 $t < T$ 是否成立,若是直接转步骤 7;否则,根据式(13)计算果蝇个体自适应免疫概率,按照免疫算法进行免疫操作,未执行免疫操作的个体转步骤 7。

步骤 6: 对执行免疫操作所得的新种群重复执行步骤 2~步骤 4 进行迭代搜索。

步骤 7: 令 $g_{\text{gen}} = g_{\text{gen}} + 1$, 若 $g_{\text{gen}} < M_{\text{Maxgen}}$ 则转步骤 2, 否则终止迭代。

IAFOA 流程图, 如图 3 所示。

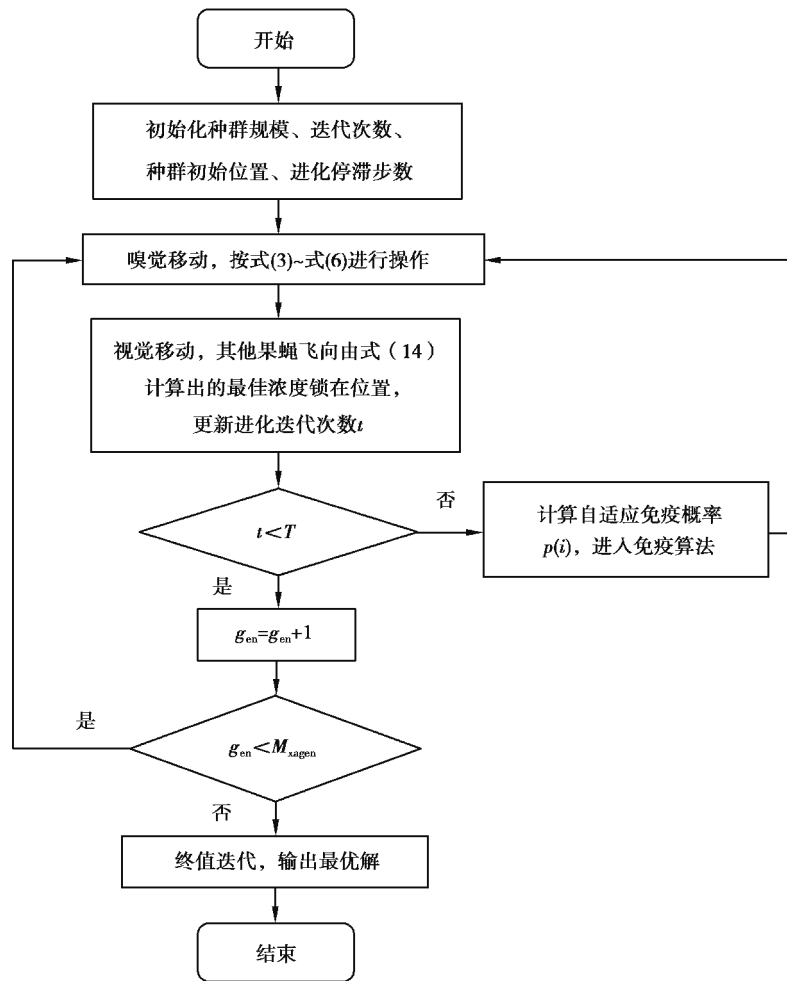


图 3 改进果蝇算法迭代图

Fig. 3 Iteration diagram of IAFOA algorithm

3 仿真实验

为验证文中算法的有效性, 选取 4 个标准测试函数进行数值仿真试验来测试 IAFOA 算法的性能^[23]。另外, 选取 0~1 背包问题进行仿真实验, 将结果与其他算法进行比较。初始化参数: 种群规模 $S_{\text{Sizepop}} = 200$, 最大迭代次数 $M_{\text{Maxgen}} = 1000$, 维度 $D = 30$, 初始概率 $p = 0.25$, 进化停滞步数阈值 $T = 3$ 。

3.1 标准函数测试

第 1 个标准函数: Rosenbrock 函数

$$f_1(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2].$$

第 2 个标准函数: Ackley 函数

$$f_2(x) = -a \exp\left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(cx_i)\right) + a + \exp(1).$$

第 3 个标准函数: Griewank 函数

$$f_3(x) = \frac{1}{4\ 000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1。$$

第 4 个标准函数:Schaffer 函数

$$f_4(x) = \sin\left(\frac{\sqrt{\sum_{i=1}^D (x_i^2)} - 0.5}{1 + 0.001 * \left(\sum_{i=1}^D (x_i^2)\right)^2}\right) + 0.5,$$

式中:Rosenbrock 函数为单峰值函数,主要用来测试文中改进算法在运算过程中的收敛性能;Ackley 函数和 Griewank 函数均为较复杂的多峰值函数,易使算法陷入局部最优,从而得不到真正的最优值,用来测试改进算法处理易陷入“早熟”的能力;Schaffer 函数具有复杂的空间性,用来测试改进算法的计算精度、收敛稳定性和算法时间复杂度。4 种标准函数如图 4 所示。

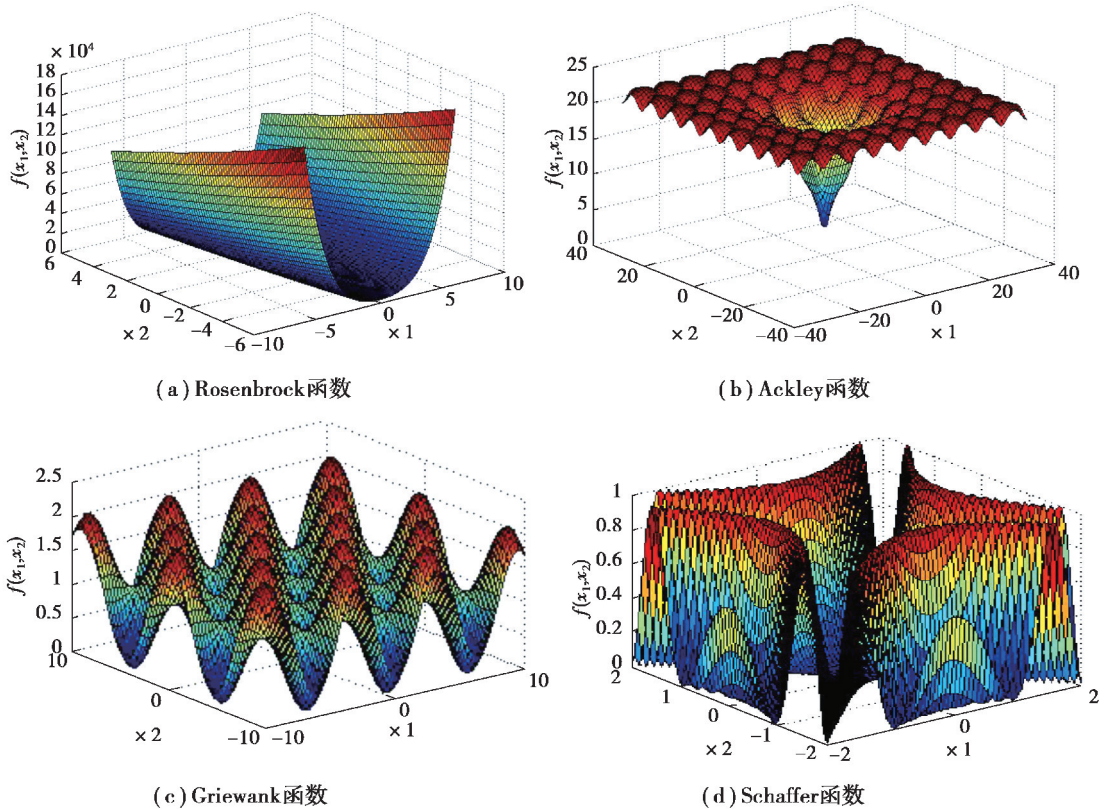


图 4 标准函数图像

Fig. 4 Standard function image

通过运用 4 种不同标准函数分别对文中改进算法的收敛性能、运行效率、处理“局部最优”以及“早熟”问题进行测试,寻优测试中的最优值、平均值,反映出解的质量。标准差反映了算法的鲁棒性。运行时间则反映算法收敛速度及精度。由表 2 可以看出,融合免疫反应的混合果蝇算法在标准函数上的表现优于标准果蝇算法。该改进算法是可行且有效的。

为了更加直观地看出改进算法性能的有效性,图 5 是免疫算法(IA, immune algorithm)、粒子群算法(PSO,particle swarm optimization)、标准果蝇算法(FOA, fruit fly optimization algorithm)及融合免疫反应的混合果蝇算法(IAFOA, fruit fly optimization algorithm based on immune algorithm)在上述 4 种标准函数下的收敛曲线图。采用多次运行结果的平均值作为最终结果以避免偶然误差。

表 2 标准函数优化性能对比

Table 2 Comparison of optimization performance for benchmark functions

函数	维数	算法	最优值	平均值	标准差	运行时间/s
f_1	30	IA	37.154 3	61.369 8	33.165 9	5.532 9
		PSO	16.599 8	34.672 9	29.526 7	3.953 2
		FOA	15.286 4	20.934 1	19.558 2	1.029 9
		IAFOA	14.095 8	17.957 4	6.469 3	1.362 1
f_2	30	IA	2.276 3	3.529 7	0.003 061	3.514 6
		PSO	2.005 1	2.119 3	0.002 796	4.228 7
		FOA	1.684 9	1.685 0	0.000 391	1.965 2
		IAFOA	0.253 9	1.005 2	0.005 983	0.953 7
f_3	30	IA	0.020 1	0.0452 1	0.306 92	4.795 8
		PSO	0.019 7	0.210 3	0.276 10	3.556 9
		FOA	0.188 9	0.175 4	0.111 75	1.259 3
		IAFOA	0.000 0	0.000 0	0.000 0	0.687 5
f_4	30	IA	0.000 196 3	0.000 234 7	0.600 210	3.638 0
		PSO	0.014 700 0	0.025 600 0	0.502 100	4.356 0
		FOA	0.020 900 0	0.136 810 0	0.038 672	0.993 8
		IAFOA	0.003 952 1	0.000 015 6	0.025 493	1.010 2

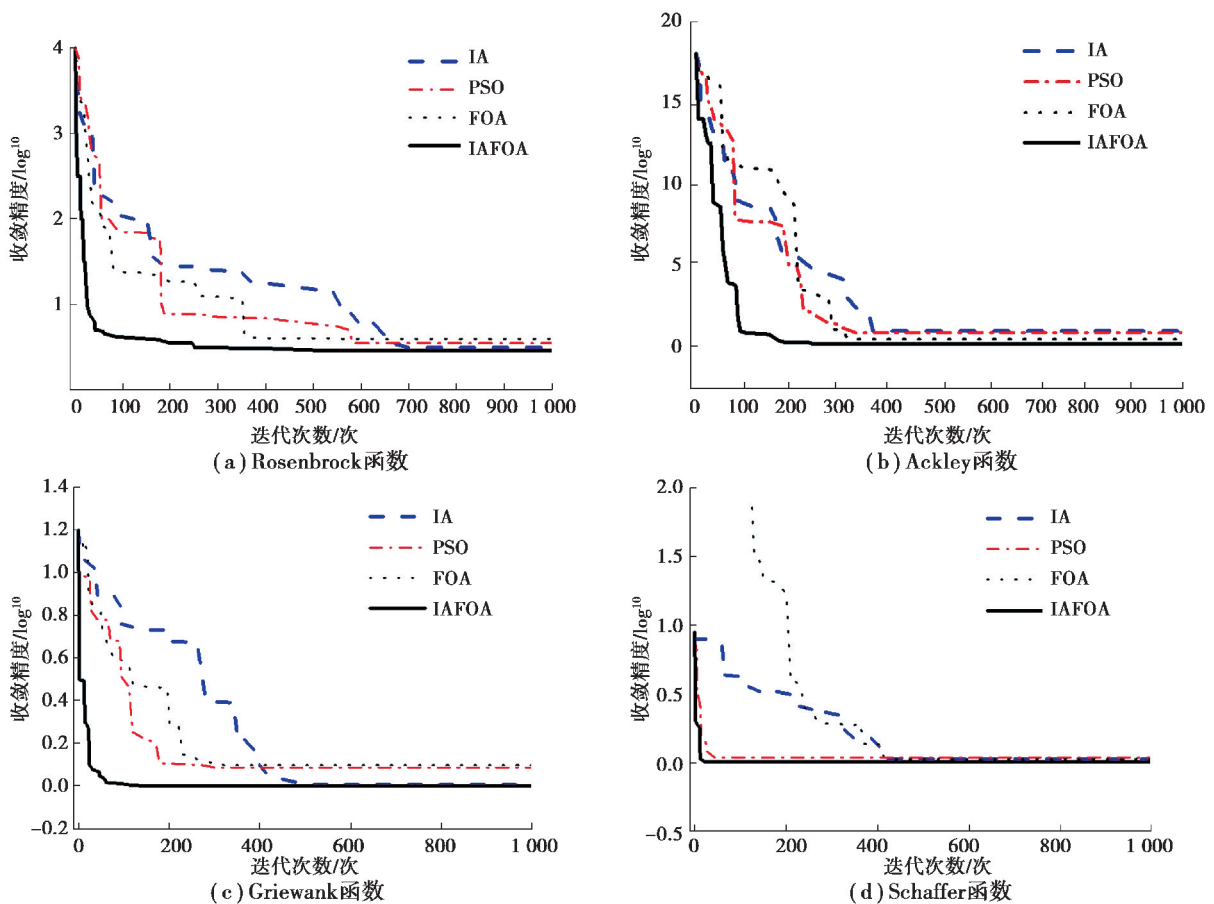


图 5 算法在 4 种函数下的迭代曲线

Fig. 5 The iterative curve of the algorithm under four functions

3.2 求解 0~1 背包问题

0~1 背包问题是组合优化领域常见问题之一,该问题的描述为:设物品总数为 D ,第 i 个物品的体积和价格分别为 w_i 和 p_i ,一个背包的最大容量为 V ,怎样选择物品使得背包的总价值最大,即为 0~1 背包问题。其数学模型为

$$\max f = \mathbf{R}\mathbf{X} = \sum_{i=1}^D R_i x_i, \tag{15}$$

$$\text{s.t. } \mathbf{W}\mathbf{X} = \sum_{i=1}^D w_i x_i \leq V, \tag{16}$$

式中: $\mathbf{R}=(R_1, R_2, \dots, R_D)$ 、 $\mathbf{W}=(w_1, w_2, \dots, w_D)$ 、 $\mathbf{X}=(x_1, x_2, \dots, x_D)$ 分别表示物品的价值向量、体积向量、解向量; x_i 为决策向量, $x_i=0$ 表示未放入包, $x_i=1$ 表示放入包。

算例的基本参数设置,如表 3 所示。

表 3 0~1 背包问题基本参数对照表

Table 3 Comparison table of basic parameters of 0~1 knapsack problem

参数	参数值
物品数量	$n=50$
背包容量	$V=1\ 000$
物品价值	$\mathbf{R}=[220, 208, 198, 192, 180, 65, 162, 160, 158, 155, 130, 125, 122, 120, 118, 115, 110, 105, 101, 100, 100, 98, 96, 95, 90, 88, 82, 80, 77, 75, 73, 72, 70, 69, 66, 65, 63, 60, 58, 56, 50, 30, 20, 15, 10, 8, 5, 3, 1]$
物品重量	$\mathbf{W}=[80, 82, 85, 70, 72, 72, 66, 50, 55, 25, 50, 55, 40, 48, 50, 32, 22, 60, 30, 32, 40, 38, 35, 32, 25, 28, 30, 22, 50, 30, 45, 30, 60, 50, 20, 65, 20, 25, 30, 10, 20, 25, 15, 10, 10, 10, 4, 4, 2, 1]$

改进算法及其他 3 种标准算法的优化对比,如表 4 和图 6 所示。结果表明,改进算法在求解 0~1 背包问题时的优化性能较优。

表 4 0~1 背包问题结果对照表

Table 4 Result comparison table of 0~1 knapsack problem

算法	最优值	最差值	平均值	方差
IA	279.5	260.1	2 865.9	5 684.100 0
PSO	299.9	248.2	2 877.2	1 2611.000 0
FOA	296.5	279.6	2 921.3	859.369 3
IAFOA	352.6	301.9	3 139.1	249.091 8

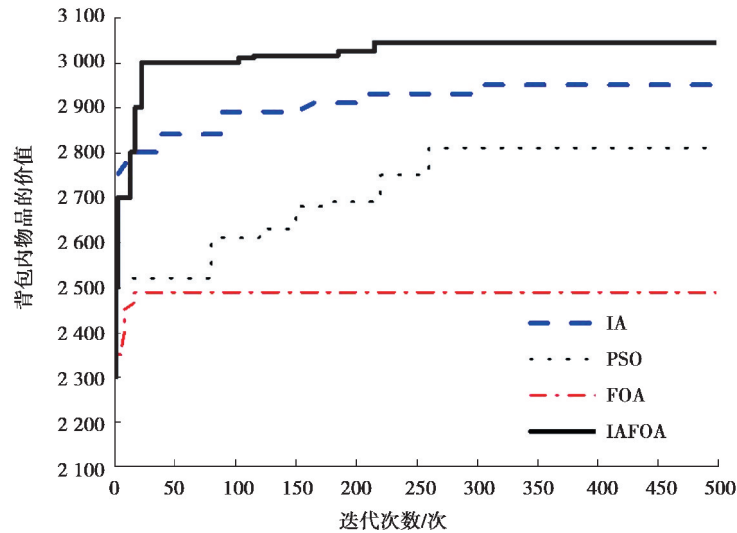


图 6 算例优化过程曲线

Fig. 6 Optimization process curve of the example

4 IAFOA 对桁架结构的优化

4.1 桁架结构优化模型

4.1.1 优化模型

以截面积为设计变量的桁架优化模型问题描述为

$$\min F = W(x), \quad (17)$$

$$\text{s.t. } g_i(x) \leq 0, i = 1, 2, \dots, m, \quad (18)$$

式中: $g_i(x)$ 为约束函数; m 为约束个数。

4.1.2 目标函数

$$W(A) = \sum_{i=1}^n \rho A_i L_i, \quad (19)$$

式中: $W(A)$ 为结构的重量; A_i 、 L_i 分别为第 i 根杆件的截面积和长度; ρ 为材料密度; n 为设计变量个数。

4.1.3 约束条件

结构中各个杆件需满足强度、刚度、稳定性及截面尺寸的要求

$$\frac{\sigma_i}{[\sigma]} - 1 \leq 0, \quad (20)$$

$$\frac{\mu_j}{\mu_{\max}} - 1 \leq 0, \quad (21)$$

$$A_{\min} \leq A_i \leq A_{\max}, \quad (22)$$

式中: σ_i 为第 i 根杆件的正应力; $[\sigma]$ 为材料的许用应力; μ_j 为节点 j 的位移; μ_{\max} 节点 j 的许用位移; A_{\min} 、 A_{\max} 分别为杆件截面的上、下限。

4.2 算例 1

建立 10 杆桁架结构模型, 如图 7 所示, 其包含节点数 6 个, 设计变量 10 个, 为求得结构的最小总重量, 现对其进行优化。 $E=68\,950$ MPa, $\rho=2\,768/\text{m}^3$, $\sigma=\pm 172.4$ MPa, 规定杆件截面面积的浮动范围为 $[0.645, 258]\text{cm}^2$, 该结构体系中图示节点受到大小为 $P=444.5$ kN 且方向向下的集中荷载, 优化结果如表 5 所示。

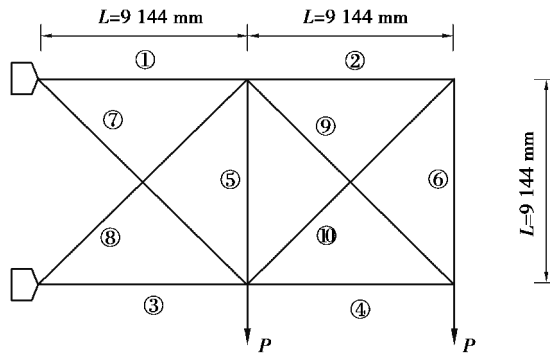


图 7 平面桁架结构图

Fig. 7 Plane truss structure diagram

表 5 平面桁架结构优化结果

Table 5 Results of plane truss structure optimization

杆件编号	杆件截面面积/mm ²			
	免疫算法 (IA)	粒子群算法 (PSO)	标准果蝇算法 (FOA)	融合免疫反应的混合果蝇算法 (IAFOA)
A ₁	210.56	196.52	198.31	195.30
A ₂	0.65	0.65	0.65	0.65
A ₃	149.50	146.96	150.91	146.23
A ₄	96.89	97.56	97.91	97.63
A ₅	0.64	0.64	0.64	0.64
A ₆	4.52	3.55	4.15	3.63
A ₇	47.66	48.07	46.96	43.95
A ₈	156.73	135.58	136.83	135.96
A ₉	139.44	149.18	148.11	145.78
A ₁₀	0.64	0.64	0.64	0.64
结构总重/kg	2 210.38	2 257.55	2 198.43	2 094.79

4.3 算例 2

选取 25 杆件空间桁架结构^[22],如图 8 所示。桁架结构的基本参数:杆件长度 $l=0.635$ m,弹性模量 $E=6.895 \times 10^4$ MPa,材料密度 $\rho=2.678 \times 10^3$ kg/m³,许用应力范围: $[-275.8, 275.8]$,结构受荷载情况如表 6 所示,节点 1 和节点 2 的最大竖向位移 $y=8.889$ mm。优化结果如表 7 所示。

表 6 25 杆桁架结构荷载工况

Table 6 Load conditions of 25-bar truss structure

节点编号	F_x	F_y	F_z
1	4.448	44.482	-22.241
2	0.000	44.482	-22.241
3	22.241	0.000	0.000
6	22.241	0.000	0.000

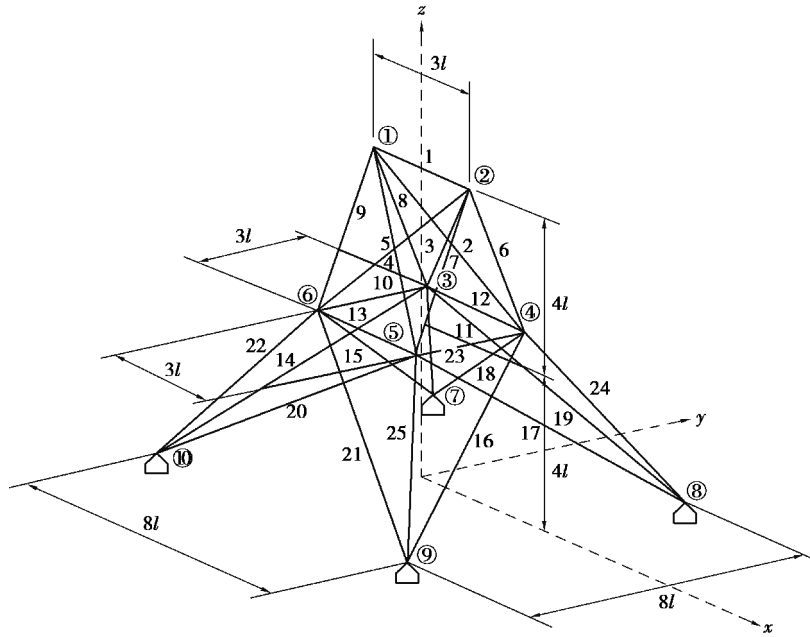


图 8 空间桁架结构图

Fig. 8 Space truss structure

表 7 25 杆桁架结构优化结果对比

Table 7 Comparison of optimization results of 25-bar truss structure

组别编号	杆件截面面积/mm ²			
	免疫算法 (IA)	粒子群算法 (PSO)	标准果蝇算法 (FOA)	融合免疫反应的 混合果蝇算法(IAFOA)
A ₁	65.70	64.516	64.900	64.10
A ₂ ~ A ₅	242.60	228.500	234.500	226.30
A ₆ ~ A ₉	2 287.50	2 237.600	2 230.100	2 240.40
A ₁₀ ~ A ₁₁	65.12	64.516	63.149	63.56
A ₁₂ ~ A ₁₃	1 245.80	1 227.900	1 226.100	1 223.50
A ₁₄ ~ A ₁₇	505.10	506.900	501.700	500.90
A ₁₈ ~ A ₂₁	92.10	83.900	89.500	85.70
A ₂₂ ~ A ₂₅	2 523.40	2 575.700	2 568.300	2 495.60
桁架结构总重量/kg	246.436	216.339	214.702	206.591

4.4 算法迭代曲线对比

通过该经典算例模型进行优化对比,由表 5 及表 7 优化结果可以看出,新算法 IAFOA 在求解速度和优化程度上的表现更良好,能够达到预想结果。通过图 9 迭代曲线可以看出,IAFOA 算法具有较好的全局搜索能力,在收敛速度及精度上比其他 3 种标准算法有较大的提高。

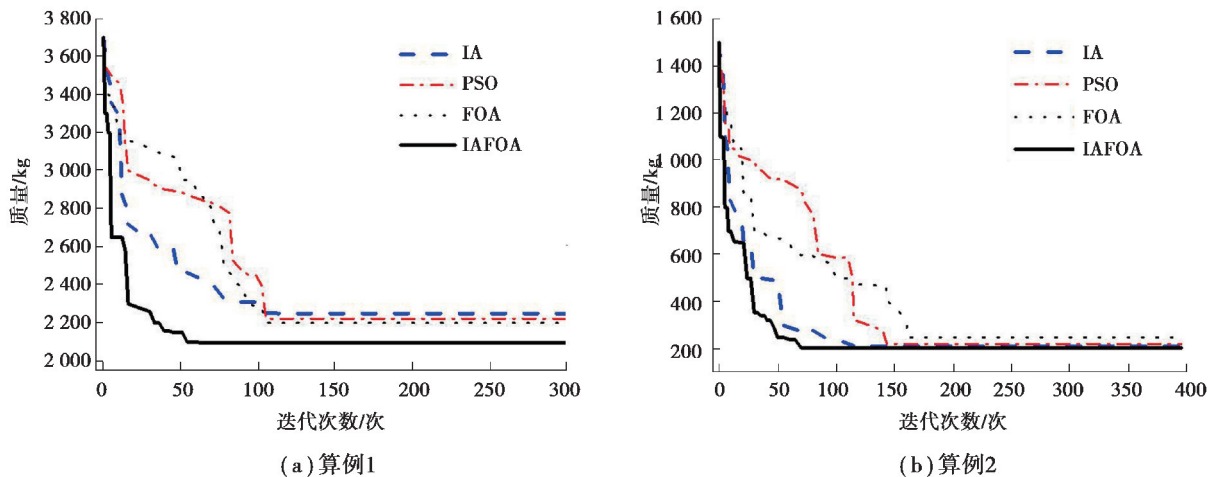


图9 2个算例的寻优迭代曲线

Fig. 9 The optimization iteration curve of two examples

5 结 论

研究将 IA 引入到 FOA 中构建 IAFOA 算法。IAFOA 同时结合了原有标准 FOA 算法的全局搜索能力和 IA 算法自身较强的局部搜索能力,在数值模拟、求解优化问题以及算例分析中 IAFOA 在收敛性、运行效率、跳出“局部最优”上均取得了较好的结果。文中虽然改进了算法的一些不足,但缺乏对参数停滞步数阈值 T 、固定概率 p 的调节,还需要更加详细的改进以求 IAFOA 算法收敛得更快更精准。另外,文中实例较为简单,算法应用于复杂结构优化中的相关研究仍需进一步深入。

参考文献:

- [1] Pan W T. A new fruit fly optimization algorithm: taking the financial distress model as an example[J]. Knowledge-Based Systems, 2012, 26: 69-74.
- [2] Shan D, Cao G H, Dong H J. LGMS-FOA: An improved fruit fly optimization algorithm for solving optimization problems[J]. Mathematical Problems in Engineering, 2013, 2013(7): 1256-1271.
- [3] Zhang Y, Cui G, Wu J, et al. A novel multi-scale cooperative mutation fruit fly optimization algorithm[J]. Knowledge-Based Systems, 2016, 114: 24-35.
- [4] Pan W T. Using modified fruit fly optimisation algorithm to perform the function test and case studies[J]. Connection Science, 2013, 25(2/3): 151-160.
- [5] Pan Q K, Sang H Y, Duan J H, et al. An improved fruit fly optimization algorithm for continuous function optimization problems[J]. Knowledge-Based Systems, 2014, 62: 69-83.
- [6] Wang L, Shi Y, Liu S. An improved fruit fly optimization algorithm and its application to joint replenishment problems [J]. Expert Systems with Applications, 2015, 42(9): 4310-4323.
- [7] Li J, Pan Q, Mao K, et al. Solving the steelmaking casting problem using an effective fruit fly optimization algorithm[J]. Knowledge-Based Systems, 2014, 72: 28-36.
- [8] 张前图, 房立清, 赵玉龙. 具有 Levy 飞行特征的双子群果蝇优化算法[J]. 计算机应用, 2015, 35(5): 1348-1352. ZHANG Qiantu, FANG Liqing, ZHAO Yulong. Gemini fly optimization algorithm with Levy flight characteristics [J]. Computer Applications, 2015, 35(5): 1348-1352. (in Chinese)
- [9] 刘志雄, 王雅芬, 张煜. 多种群果蝇优化算法求解自动化仓库拣选作业调度问题[J]. 武汉理工大学学报, 2014, 36(3): 71-77. LIU Zhixiong, WANG Yafen, ZHANG Yu. Multi-population fruit fly optimization algorithm for automated warehouse selection scheduling [J]. Journal of Wuhan University of Technology, 2014, 36(3): 71-77. (in Chinese)
- [10] 韩俊英, 刘成忠, 王联合国. 动态双子群协同进化果蝇优化算法[J]. 模式识别与人工智能, 2013, 26(11): 1057-1067.

- HAN Junying, LIU Chengzhong, WANG Lianguo. Dynamic twin group coevolution drosophila optimization algorithm [J]. Pattern Recognition and Artificial Intelligence, 2013, 26(11): 1057-1067. (in Chinese)
- [11] 蔡自兴, 龚涛. 免疫算法研究的进展[J]. 控制与决策, 2004, 19(8): 841-846.
CAI Zixing, GONG Tao. Progress of immune algorithm research [J]. Control and Decision, 2004, 19(8): 841-846. (in Chinese)
- [12] Chen D, Li S, Wang J, et al. A multi-objective trajectory planning method based on the improved immune clonal selection algorithm[J]. Robotics and Computer-Integrated Manufacturing, 2019, 59, 431-442.
- [13] Zhen Y, Ding Y, Hao K, et al. An adaptive immune algorithm for service-oriented agricultural internet of things[J]. Neurocomputing, 2019: 3-12.
- [14] Corus D, Oliveto P S, Yazdani D. When hypermutations and ageing enable artificial immune systems to outperform evolutionary algorithms[J]. Theoretical Computer Science, 2019, 3:1-34.
- [15] 胡东方, 雷若楠. 基于实例推理与人工免疫系统的工程产品设计方法[J]. 计算机集成制造系统, 2018, 24(1):146-153.
HU Dongfang, LEI Ruonan. Engineering product design method based on case reasoning and artificial immune system [J]. Computer Integrated Manufacturing System, 2018, 24(1):146-153. (in Chinese)
- [16] 徐小平, 师喜婷, 钱富才. 基于猴群算法求解 0-1 背包问题[J]. 计算机系统应用, 2018, 27(5):133-138.
XU Xiaoping, SHI Xiting, QIAN Fucui. Solving 0-1 knapsack problem based on monkey swarm algorithm [J]. Application of Computer System, 2018, 27(5):133-138. (in Chinese)
- [17] Hassan Z, Bahman N, Mohammad M. The economic lot scheduling problem in limited-buffer flexible flow shops: Mathematical models and a discrete fruit fly algorithm[J]. Applied Soft Computing Journal, 2019, 6:904-919.
- [18] 王凌, 郑晓龙. 果蝇优化算法研究进展[J]. 控制理论与应用, 2017, 34(5):557-563.
WANG Ling, ZHENG Xiaolong. Research progress of fruit fly optimization algorithm [J]. Control Theory and Application, 2017, 34(5):557-563. (in Chinese)
- [19] Hu J, Chen P, Yang Y, et al. The fruit fly optimization algorithms for patient-centered care based on interval trapezoidal type-2 fuzzy numbers[J]. International Journal of Fuzzy Systems, 2019, 5:1270-1287.
- [20] Bersini H, Varela F. Hints for adaptive problem solving gleaned from immune network[A]. Berlin Heidelberg: Springer-Verlag, 1991:343-354.
- [21] 张金梦, 刘慧君. 遗传算法优化 BP 神经网络的停车位数量预测[J]. 重庆大学学报, 2018, 41(3):76-81.
ZHANG Jinmeng, LIU Huijun. Prediction of parking space based on BP neural network optimized by genetic algorithm[J]. Journal of Chongqing University, 2018, 41(3):76-81. (in Chinese)
- [22] 谢开贵, 曾晓晖, 李春燕, 等. 免疫算法与其他随机优化算法的比较分析[J]. 重庆大学学报, 2003, 26(11): 43-47.
XIE Kaigui, ZENG Xiaohui, LIU Chunyan, et al. Comparative analysis of immune algorithm and other stochastic optimization algorithms [J]. Journal of Chongqing University, 2003, 26(11):43-47. (in Chinese)
- [23] 李彦苍, 王旭. 基于信息熵的改进海豚群算法及其桁架优化[J]. 重庆大学学报, 2019, 42(5):76-85.
LI Yancang, WANG Xu. Improved dolphin swarm algorithm based on information entropy and its truss optimization[J]. Journal of Chongqing University, 2019, 42(5):76-85. (in Chinese)

(编辑 詹燕平)