

具有自适应搜索策略的灰狼优化算法

魏政磊¹ 赵辉¹ 韩邦杰^{1,2} 孙楚¹ 李牧东¹

(空军工程大学航空航天工程学院 西安 710038)¹ (空军驻石家庄地区军事代表室 邯郸 056004)²

摘要 灰狼优化算法(Grey Wolf Optimization, GWO)是一种新型的群智能优化算法。与其他智能优化算法类似,该算法仍存在收敛速度慢、容易陷入局部极小点的缺点。针对这一问题,提出了具有自适应搜索策略的改进算法。为了提高算法的收敛速度和优化精度,通过适应度值控制智能个体位置,并引入了最优引导搜索方程;另一方面,为提高GWO的种群多样性,改进算法利用位置矢量差随机跳出局部最优。最后对10个标准测试函数进行了仿真实验,并与其他4种算法进行了比较,统计结果和Wilcoxon符号秩检验结果均表明,所提出的改进算法在收敛速度以及搜索精度方面具有明显优势。

关键词 灰狼优化算法, 自适应, 最优学习搜索方程, Wilcoxon符号秩检验, 函数优化

中图分类号 TP18 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.03.053

Grey Wolf Optimization Algorithm with Self-adaptive Searching Strategy

WEI Zheng-lei¹ ZHAO Hui¹ HAN Bang-jie^{1,2} SUN Chu¹ LI Mu-dong¹

(College of Aeronautics and Astronautics, Air Force Engineering University, Xi'an 710038, China)¹

(Military Delegate Office of Air Force, Handan 056004, China)²

Abstract The grey wolf optimization algorithm (GWO) is a new type intelligence optimization algorithm. However, the GWO has the problem of low convergence speed and easily falling into local optimum as the same as other optimization algorithms. Aiming at this problem, an improved GWO with the self-adaptive searching strategy was proposed. Firstly, the search direction is controlled by fitness value and the best learning equation is introduced to improve the convergence speed and the optimization precision. Secondly, in order to maintain the population diversity, the position vector difference is utilized to escape from local optimum. Finally, 10 benchmarks and other four popular algorithms were compared to illustrate the superiority of GWO with the self-adaptive searching strategy. The experimental results and the Wilcoxon signed ranks test results show that the improved GWO outperforms the other four algorithms in terms of convergence speed and precision.

Keywords GWO, Self-adaptive, Best learning equation, Wilcoxon signed ranks test, Function optimization

自20世纪90年代以来,元启发式优化算法以结构简单、求解效率高等特点得到了前所未有的发展,典型代表有遗传算法(Genetic Algorithm, GA)^[1]、蚁群算法(Ant Colony Optimization, ACO)^[2]、粒子群算法(PSO)^[3]、人工蜂群算法(ABC)^[4]等,并且已经成功运用在很多优化问题上。然而,多模态、高维、非线性优化问题的出现对全局优化算法提出了更大的挑战。对此,学者们在现有群体智能方法研究的基础上提出了大量的改进算法以及新的优化算法,以提高算法的优化效率。文献[5]利用邻域的大小随迭代次数的增加逐渐加大的特性,保证了更大的种群多样性和更快的收敛速度;文献[6]在ABC算法的基础上通过引入自适应变异策略提高了算法的性能;文献[7-9]通过模拟群居生物相互合作工作行为分

别提出了布谷鸟搜索(cuckoo search, CK)算法、引力搜索算法(Gravity Search Algorithm, GSA)以及共栖生物搜索(Symbiotic Organisms Search, SOS)算法等新型智能优化算法。

受狼群猎食行为的启发, Seyedali, Seyed 和 Andrew 提出了灰狼优化算法(GWO)^[10],它是一种模仿灰狼的社会等级制度和猎食行为的新型群智能优化算法。与其他进化算法相比,灰狼算法通过比较得到各个灰狼社会等级并控制搜索方向,更好地提高了算法的优化效率,同时该算法由于控制参数较少,因此实现简单。文献[10]通过对不同类型测试函数进行仿真实验,证明了该算法具有较好的优化性能。

但是,与其他搜索算法一样, GWO 算法同样存在着早

到稿日期:2016-01-13 返修日期:2016-07-26 本文受国家自然科学基金(71501184)资助。

魏政磊(1991-),男,硕士生,主要研究方向为最优化理论与方法、无人飞行器多目标决策, E-mail: 18792805060@163.com; 赵辉(1973-),男,教授,博士生导师,主要研究方向为武器系统与运用工程、最优化方法; 韩邦杰(1972-),男,硕士,主要研究方向为最优化理论与方法、无人飞行器多目标决策; 孙楚(1992-),男,硕士生,主要研究方向为最优化理论与方法、无人飞行器武器系统; 李牧东(1988-),男,博士,主要研究方向为最优化理论与方法。

熟、易陷入局部极小点的缺点;同时,由于没有体现个体社会等级与重要程度,解决函数优化问题时不易到达最优解。针对这些问题,本文提出了一种改进GWO。受文献[11-15]的启发,首先为了提高算法的收敛速度,利用适应度值控制智能个体位置,根据种群最优信息来控制算法的搜索方向,引入最优引导搜索方程;另一方面,为提高GWO的种群多样性,改进算法针对种群多样性变化特征,利用最优个体与当前个体之间的位置矢量差控制当前个体不在最优解的区域内。通过多组仿真实验验证了所提策略的有效性。

1 灰狼优化算法

灰狼优化算法模拟灰狼的社会等级制度和猎食行为,视灰狼为猎食者顶端,即处于食物链最上层。在寻找最优解过程中,根据灰狼猎食的3个重要步骤,即接近、包围、攻击猎物,GWO算法建立了一个模型:狼群中每一个灰狼代表了种群的一个潜在解,其中领导狼群的 α 狼位置是最好的解,处于狼群等级第二阶层的 β 狼位置和负责侦察、警戒、打围以及看守的 δ 狼位置分别为最优解和次优解,其他的候选解是阶层较低的 ω 狼位置。

(1) 种群初始化

由于GWO的性能受种群初始值影响较小,因此在该算法中采用随机产生种群的方法进行初始化,即

$$X_{i,j} \sim U(lb_j, ub_j) \quad (1)$$

其中, X 为灰狼种群, $i \in [1, 2, 3, \dots, N]$ 且 $j \in [1, 2, 3, \dots, D]$, N 是灰狼种群个数, D 是种群维数; lb 和 ub 分别为搜索区间的上界和下界; U 是随机均匀分布函数。

(2) 种群搜索

通过式(2)和式(3)搜索接近猎物:

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (2)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (3)$$

其中, \vec{D} 为猎物与灰狼之间的距离; t 为迭代的次数; \vec{C} 和 \vec{A} 为系数向量; \vec{X}_p 和 \vec{X} 为猎物位置向量和灰狼位置向量。

向量 \vec{X} 和 \vec{C} 的计算公式如下:

$$\vec{X} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (4)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (5)$$

其中, \vec{r}_1 和 \vec{r}_2 是在 $[0, 1]$ 范围内的随机数。一般情况,控制参数 a 在 $[0, 2]$ 范围内取值,且随着算法迭代次数的增加而线性递减。对应于式(4), $|\vec{A}| \geq 1$ 意味着灰狼进行全局搜索, $|\vec{A}| < 1$ 表示灰狼在附近搜索。

(3) 种群位置更新

通过计算灰狼的适应度值,得到最优解、优解和次优解,其他灰狼的位置由 α 、 β 和 δ 的位置共同决定。在产生新群体后,对种群中的元素进行边界控制,完成一次迭代。重复上述过程,直到满足循环终止条件,最后输出最优解。

$$\begin{cases} \vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \\ \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \\ \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \end{cases} \quad (6)$$

$$\begin{cases} \vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha) \\ \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta) \\ \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta) \end{cases} \quad (7)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (8)$$

2 自适应搜索策略及其相应算法

通过分析GWO的原理可知,该算法在循环初期具有较快的收敛速度,全局搜索能力强,然而随着迭代次数的增多,GWO容易陷入局部最优,导致算法收敛减慢,对于多峰值函数问题,存在达不到最优解的缺点。同时,GWO中仅仅通过最优解、优解和次优解更新种群的位置,种群多样性差,这也是导致其早熟、易陷入局部最优的原因。对此,本文提出了自适应调整策略,通过学习种群最优信息来控制算法的搜索方向;另一方面,考虑到保持GWO的种群多样性,改进算法利用最优个体与当前个体之间的位置矢量差跳出局部最优区域。

2.1 自适应调整策略

在GWO算法中,灰狼个体的适应度值是反映个体当前位置优劣的一个参数。在狼群接近目标的过程中,位置更新方程(7)表现出 α 、 β 和 δ 3个解的同等重要性,但是忽略了 α 、 β 和 δ 3个解的不同特征, α 灰狼的领导位置与最优解的所占比例没有体现在式(7)中;同时,对适应度值差的位置邻域进行搜索,降低了收敛效率^[16]。

为了进一步提高算法的收敛速度,采用自适应调整策略,将当前个体的适应度值 f_i 与灰狼群的平均适应度值 f_{avg} 进行比较,如果 f_i 优于 f_{avg} ,则继续使用原策略更新灰狼位置;如果 f_i 次于 f_{avg} ,则利用 α 、 β 和 δ 3个灰狼的适应度值变异灰狼位置。

在变异中,要更新个体的学习较优个体位置信息概率与适应度值成比例,对于原种群里较优个体中适应度值越小的个体,新个体要学习其位置信息的概率会越大。因此,本文对位置更新方程(7)进行改进,创建一个与适应度值成比例的概率分布,并通过该分布求解最优解、优解和次优解。

$$\vec{X}(t+1) = \begin{cases} \frac{(\frac{1}{f_\alpha})\vec{X}_1 + (\frac{1}{f_\beta})\vec{X}_2 + (\frac{1}{f_\delta})\vec{X}_3}{\frac{1}{f}}, & f_i \leq f_{avg} \\ \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}, & f_i > f_{avg} \end{cases} \quad (9)$$

$$\frac{1}{f} = \frac{1}{f_\alpha} + \frac{1}{f_\beta} + \frac{1}{f_\delta} \quad (10)$$

其中, f_α 、 f_β 和 f_δ 分别为 α 、 β 和 δ 的适应度值。

2.2 跳出局部最优策略

灰狼优化算法在循环初期具有较快的收敛速度,种群位置变化率很大,种群多样性和全局搜索能力较强,然而随着迭代次数的增多,种群位置变化率降低,种群多样性减小,GWO容易陷入局部最优,对于多峰值函数问题,存在达不到最优解的缺点。

针对种群多样性的变化特征,跳出局部最优策略将智能个体置于不包含当前最优解的区域内,以此来验证算法能够找到不在初始化区域内的最优解^[17],增加种群多样性。因此,本文对更新后的位置进行重置,通过最优个体与当前个体

之间的位置矢量差随机控制当前个体不在最优解的区域内:

$$\vec{X}_i'(t+1) = \vec{X}_\alpha(t) + |\vec{X}_\alpha(t) - \vec{X}_i(t)| \cdot \vec{r} \quad (11)$$

其中, $\vec{X}_i'(t+1)$ 为跳出局部最优的智能个体位置; $\vec{X}_i(t)$ 为位置更新后的智能个体位置; \vec{r} 为 $[-2, -1]$ 间和 $[1, 2]$ 间的随机数。

2.3 最优学习搜索方程

虽然标准的 GWO 算法具有初期快速收敛性、良好的全局搜索能力与局部搜索能力之间的平衡性和简单易行等优点,但是随着收敛次数的增加,该算法收敛速度变慢、搜索精度降低而且容易陷入局部最优。受文献[11-12,15]的启发,提出了最优学习搜索方程:

$$V_{i,j}(t+1) = \omega \cdot V_{i,j}(t) + \varphi_{i,j} \cdot (Y_{i,j}(t) - X'_{i,j}(t)) + \mu_{i,j} \cdot (X_{\alpha,j}(t) - X'_{i,j}(t)) \quad (12)$$

$$X''_{i,j}(t+1) = X'_{i,j}(t) + V_{i,j}(t+1) \quad (13)$$

其中, V_i 为第 i 灰狼的当前速度; Y_i 为第 i 灰狼的最优位置; X'_i 为第 i 灰狼的当前位置; X_α 为种群中的最优位置; X''_i 为学习后的位置; ω 为惯性权重,一般取 0.8; $\varphi_{i,j}$ 和 $\mu_{i,j}$ 分别为 $[-1, 1]$ 间和 $[0, 1.5]$ 间的随机数。

2.4 改进算法流程

由于 GWO 存在收敛速度慢、容易陷入局部最优的缺点,结合本文提出的自适应调整策略、最优学习搜索方程和跳出局部最优策略,提出了一种新的 GWO 改进算法。该算法首先通过智能个体的适应度值改进种群位置更新公式,对适应度值差的位置邻域进行搜索;随后对更新后的位置进行重置,通过最优解与当前解之间的位置矢量差控制当前个体不在当前最优解的区域内,避免陷入局部最优;最后通过重置后的种群采用最优学习搜索策略方程进行进一步的优化操作,使其快速跳出局部最优,提高其收敛速度。具体算法流程图如图 1 所示。

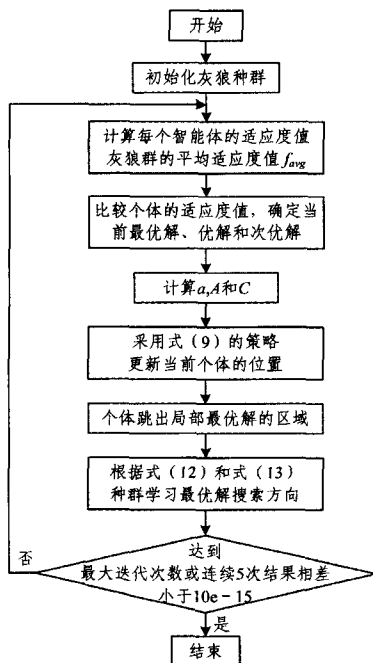


图 1 改进后灰狼优化算法流程图

下面给出了 GWO 改进算法的流程:

(1) 初始化灰狼种群: 随机产生 n 个智能个体的位置和速度; 初始化 a, A 和 C ; 初始化 ω ; 初始化 X_α, X_β 和 X_δ 的值。

(2) 处理越界智能个体, 计算每个智能体的适应度值和灰狼群的平均适应度值 f_{avg} 。

(3) 比较智能个体的适应度值以及 X_α, X_β 和 X_δ 的适应度值, 确定当前迭代期间的最优解 X_α 、优解 X_β 和次优解 X_δ 。

(4) 根据控制参数 a 随迭代次数增加而线性递减的特性计算各个优解的 A 和 C 。

(5) 比较当前智能个体的适应度值 f_i 与 f_{avg} , 采用式(9)的策略更新当前智能个体的位置。

(6) 利用式(11)使智能个体跳出局部最优解的区域, 保留优解 X_α, X_β 和 X_δ 。

(7) 根据式(12)和式(13), 种群学习最优解搜索方向。

(8) 输出最优解 X_α 和最优值。

(9) 如果达到结束条件(最大迭代次数或连续 5 次结果相差小于 $10e-15$), 则结束; 否则转到步骤(2)。

3 仿真实验

3.1 实验设置

为了验证本文提出的 GWO 改进算法(简称 SAGWO)的有效性,对文献[18]中的 10 个典型函数优化问题进行测试仿真,同时将其与标准 GWO、文献[5]提出的改进 PSO 算法、文献[8]提出的 GSA 算法以及文献[9]提出的 SOS 算法进行比较。表 1 列出这 10 个函数的基本信息,包括测试函数公式、函数维数、搜索空间以及最优值。 F_1, F_2 和 F_3 是单峰标准函数, F_4, F_5 和 F_6 是多峰标准函数, F_7, F_8, F_9 和 F_{10} 是固定维多峰标准函数。

表 1 10 个测试函数的基本描述

函数	名称	维数	变量定义域	最优值
F_1	Sphere Function	30	$[-100, 100]$	0
F_2	Schwefel's Problem 2.2.2	30	$[-10, 10]$	0
F_3	Quadratic with Noise	30	$[-1.28, 1.28]$	0
F_4	Rastrigin Function	30	$[-5.12, 5.12]$	0
F_5	Ackley Function	30	$[-32, 32]$	0
F_6	Griewank Function	30	$[-600, 600]$	0
F_7	Foxholes Function	2	$[-65, 65]$	1
F_8	Six Hump Camel Back	2	$[-5, 5]$	-1.0316
F_9	Branin RCOS Function	2	$[-5, 5]$	0.398
F_{10}	Goldstein Price Function	2	$[-2, 2]$	3

在仿真中,灰狼种群大小设置为 30, a 的初始值为 2, ω 设置为 0.8, 迭代次数为 500, 在达到最大迭代次数或者当连续 5 次结果相差小于 $10e-15$ 时输出结果; 其他两种比较算法按照文献[5, 19]的参数设置方法进行设置。

3.2 实验结果与分析

本文采用 Matlab 8 进行仿真, 运行环境为 Inter(R)Core (TM)i5-3470 处理器, 3.46GB 内存。仿真针对同一测试函数, 每种算法独立运行 30 次, 比较 5 种算法对 10 个测试函数的统计平均值和方差。另外, 为了更好地对实验结果进行分析, 本文引入了文献[19]采用的优化算法比较方法, 即 Wilcoxon 符号秩检验。该检验方法是一种成对比较方法, 能够有效检验出不同算法存在的明显差异。在文中通过将 SAGWO

分别与其他算法进行比较得出统计数据,其中,将30次独立运行的最优值作为算法间比较的样本数据,信息显著水平设置为0.05。

表2给出了5种算法在10个测试函数下30次独立运行的统计平均值(Mean)和方差(Std)。从表2中可以看出,SAGWO无论在收敛精度还是算法稳定性,相比于GWO,在

函数 F_8, F_9 和 F_{10} 外,均有明显提高。相比于IPSO算法,对于函数 F_1-F_7 ,SAGWO算法在收敛精度和算法稳定性方面均优于IPSO算法;对于函数 F_8 和 F_9 ,SAGWO算法在收敛精度方面与IPSO算法相当,但稳定性方面优于IPSO算法。相比于GSA和SOS算法,除了函数 F_9 和 F_{10} ,SAGWO均表现出了明显较优的优化性能。

表2 5种算法对10种测试函数的测试结果

算法	F_1	F_2	F_3	F_4	F_5
	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)
IPSO	2.3212e+00 (9.6703e-01)	4.0200e+00 (1.6001e+00)	5.0092e+00 (1.4095e+01)	1.5554e+02 (1.2872e+03)	2.5401e+00 (1.1278e-01)
GSA	2.0401e-16 (4.6106e-33)	1.0151e-07 (2.7509e-15)	9.3829e-02 (2.7860e-03)	2.9526e+01 (2.2610e+01)	4.8452e-03 (2.1100e-04)
SOS	9.8869e-02 (3.5214e-04)	1.0502e+00 (1.4466e+01)	5.2160e-02 (6.3541e-04)	1.4429e+02 (3.6220e+04)	3.0312e+01 (2.1147e+02)
GWO	1.3545e-27 (1.7303e-54)	7.9042e-17 (4.8536e-33)	1.6813e-03 (1.8163e-07)	2.9606e+00 (1.3812e+01)	1.1354e-13 (3.2627e-28)
SAGWO	4.5223e-37 (2.5414e-72)	1.1780e-22 (5.6766e-44)	1.1528e-04 (4.0056e-09)	0.0000e+00 (0.0000e+00)	9.7761e-15 (1.6219e-28)

算法	F_6	F_7	F_8	F_9	F_{10}
	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)	Mean(Std)
IPSO	1.4110e-01 (2.6460e-03)	3.5585e+00 (7.7750e+00)	-1.0316e+00 (4.9304e-32)	3.9989e-01 (3.0815e-33)	3.0000e+00 (0.0000e+00)
GSA	2.8499e+01 (6.5874e+01)	4.5223e+00 (1.1154e+01)	-1.0316e+00 (4.9303e-32)	3.9789e-01 (3.0815e-33)	3.0000e+00 (0.0000e+00)
SOS	1.1518e-02 (1.1338e-04)	9.9800e-01 (1.0152e-02)	-1.0316e+00 (4.9303e-32)	3.9789e-01 (3.0815e-33)	3.0000e+00 (0.0000e+00)
GWO	5.6966e-03 (8.8605e-05)	6.0869e+00 (1.8304e+01)	-1.0316e+00 (4.9304e-32)	3.9789e-01 (3.0815e-33)	3.0001e+01 (3.6000e-09)
SAGWO	0.0000e+00 (0.0000e+00)	1.3949e+00 (6.2986e-01)	-1.0316e+00 (4.9303e-32)	3.9790e-01 (1.0216e-33)	3.0000e+00 (0.0000e+00)

为了进一步说明SAGWO算法的有效性,图2示出了5种算法的收敛曲线。由于篇幅限制,本文列出了典型的6种测试函数的收敛曲线。从图2中可以看出,相比于GWO,IP-

SO,GSA和SOS算法,本文的改进算法具有较快的收敛速度,达到了更高的精度。总的来说,SAGWO算法的性能好于其他4种算法。

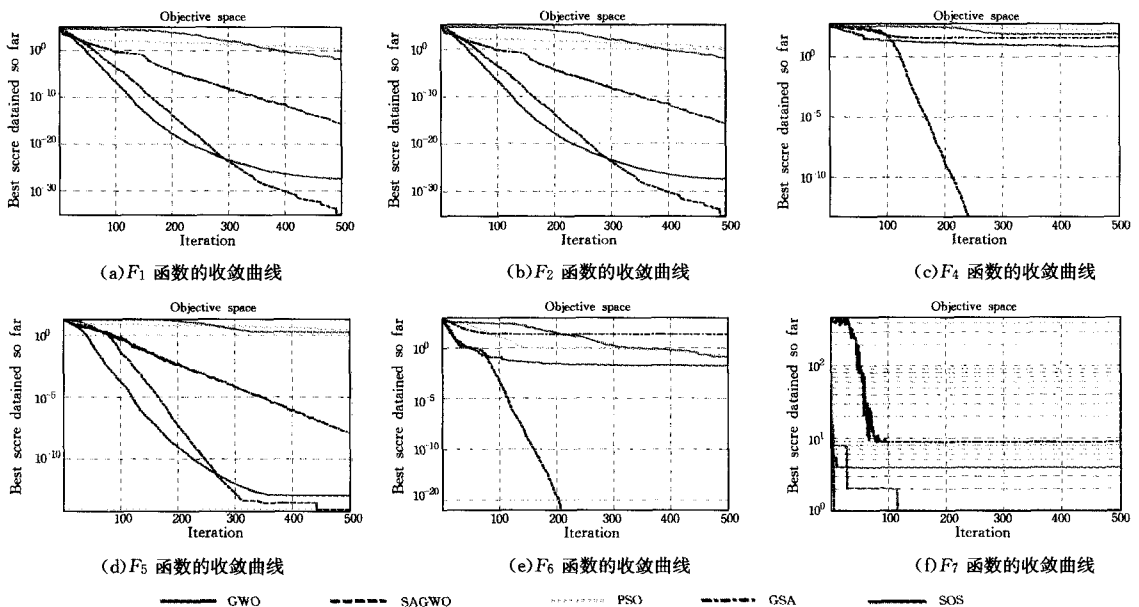


图2 5种不同算法对6个测试函数的平均最优适应度值曲线

为了更清楚地比较5种算法的优化性能,表3列出了Wilcoxon符号秩检验的检查结果。其中,“T+”表示SAGWO样本数据中优于其他算法的符号数;“T-”表示不如其

比较算法的符号数;“+”表示假设不成立(两种算法存在明显差异),SAGWO表现出更好的性能;“-”表示假设不成立(两种算法存在明显差异),SAGWO表现出较差的性能;“=”

表示假设成立,即两种算法没有较大差异。各统计值的具体计算原理可参见文献[19]。从表 3 可以看出,相比于 GWO 和 IPSO 算法,在函数 F_8, F_9 和 F_{10} 的优化中,本文算法与其并无明显差异,而在其他函数的优化方面,本文算法表现出了明显优于 GWO 和 IPSO 算法的优化性能;相比于 GSA 和

SOS 算法,在 F_9 和 F_{10} 的优化中,本文算法的优化性能与其相当,而在其他函数的优化方面,本文算法表现出明显优于 GSA 的性能。因此,相比于其他 4 种算法, SAGWO 对 10 个标准测试函数的实验结果可以说明其改进的有效性。

表 3 基于 30 次独立运行的 10 个测试函数最优值双边 Wilcoxon 符号秩检验结果

函数	GWO vs. SAGWO				PSO vs. SAGWO				GSA vs. SAGWO				SOS vs. SAGWO			
	p	T+	T-	winer	p	T+	T-	winer	p	T+	T-	winer	p	T+	T-	winer
F_1	4.64e-08	465	0	+	6.80e-08	465	0	+	6.80e-08	465	0	+	6.80e-08	465	0	+
F_2	6.80e-08	465	0	+	6.80e-08	465	0	+	6.80e-08	465	0	+	6.80e-08	465	0	+
F_3	1.83e-04	465	0	+	1.83e-04	465	0	+	1.83e-04	465	0	+	1.83e-04	465	0	+
F_4	6.29e-05	465	0	+	6.39e-05	465	0	+	6.25e-05	465	0	+	1.73e-06	465	0	+
F_5	1.60e-04	465	0	+	1.62e-04	465	0	+	1.62e-04	465	0	+	1.73e-06	465	0	+
F_6	3.16e-03	465	0	+	6.39e-05	465	0	+	6.39e-05	465	0	+	2.35e-06	465	0	+
F_7	1.70e-03	465	0	+	6.62e-03	465	0	+	1.20e-03	465	0	+	4.30e-06	465	0	+
F_8	1.00e+00	0	0	=	1.00e+00	0	0	=	7.79e-02	440	25	+	1.00e+00	0	0	=
F_9	1.00e+00	0	0	=	1.00e+00	0	0	=	1.00e+00	0	0	=	1.00e+00	0	0	=
F_{10}	1.00e+00	0	0	=	1.00e+00	0	0	=	1.00e+00	0	0	=	2.74e-01	167	268	-
+/-/=	7/0/3				7/0/3				8/0/2				7/1/2			

结束语 灰狼优化算法从灰狼群体的社会行为中得到启发,是一种新的群智能优化算法,实现简单,但是迭代后期容易陷入局部最优,导致算法收敛减慢。本文基于灰狼优化算法的特点,提出了具有最优学习搜索策略的灰狼优化算法,将最优引导搜索方程引入到该策略中,并通过智能个体的适应度值改进种群位置更新公式,最后对位置更新后的智能个体重新进行初始化。从对 10 个标准测试函数的统计结果和 Wilcoxon 符号秩检验结果可以看出,相比于其他 4 种优化算法,本文提出的 SAGWO 算法有效提高了算法搜索性能,克服了在多峰测试函数优化中易陷入局部最优的缺点。此外,如何利用改进的优化算法解决多目标优化问题以及动态优化问题是下一步需要研究的方向。

参考文献

[1] HOLLAND J. Adaptation in natural and artificial systems[M]. Cambridge, MA: MIT Press, 1992.

[2] DORIGO M, STUTZLE T. Ant colony optimization[M]. Cambridge, MA: MIT Press, 2004.

[3] KENNEDY J, EBERHART R. Particle swarm optimization [C]// IEEE Int Conf on Neural Networks, Piscataway: IEEE, 1995: 1942-1948.

[4] KARABOGA D. An idea based on bee swarm for numerical optimization[R]. Turkey: Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[5] SUGANTHAN P N. Particle swarm optimizer with neighborhood operator[C]// Proceedings of the IEEE Congress on Evolutionary Computation. IEEE Press, 1999: 1958-1962.

[6] HE P, YAN X D, SHI H B. A quick self-adaptive artificial bee colony Algorithm and its application[J]. Journal of East China University of Science of Technology(Natural Science Edition), 2013, 39(5): 588-595. (in Chinese)
何鹏, 阎兴岷, 何洪波. 一种快速自适应蜂群算法及其应用[J]. 华南理工大学学报(自然科学版), 2013, 39(5): 588-595.

[7] YANG X S, DEB S. Cuckoo search via levy flights[C]// Proceedings of the World Congress on Nature and Biologically Inspired Computing. 2009: 210-214.

[8] RASHEDI E, NEZAMABADI-POUR H, SARYAZDI S. GSA: a gravitation search algorithm[J]. Information Science, 2009, 179

(13): 2232-2248.

[9] CHENG M Y, PRAYOGO D. Symbiotic Organisms Search: A new metaheuristic optimization algorithm[J]. Computers and Structures, 2014, 139: 98-112.

[10] SEYEDAIL M, SEYED M M, LEWIS A. Grey wolf optimizer [J]. Advances in Engineering Software, 2014, 69: 46-61.

[11] MINHAZUL I S, DAS S, GHOSH S, et al. An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization[J]. IEEE Trans. on Systems, Man, and Cybernetics—Part B: Cybernetics, 2012, 42(2): 482-500.

[12] BACK T, FOGEL D B, MICHALEWIC Z. Evolutionary computation 2[M]// Advanced Algorithms and Operators. IOP Press, 2000.

[13] EBERHART R C, SHI Y. Tracking and optimizing dynamic systems with particle swarms[C]// Proceedings of the IEEE Congress on Evolutionary Computation, 2001: 94-100.

[14] KERSTIN W, AAN C, TOMAS M. Knowledge-guided genetic algorithm for input parameter optimization in environmental modeling[J]. Procedia Computer Science, 2012, 1: 1367-1375.

[15] LIANG J J, QIN A K, SUGANTHAN P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal function[J]. IEEE Trans. on Evolutionary Computation, 2006, 10(3): 281-295.

[16] WEI Z L, ZHAO H, LI M D, et al. Grey wolf optimization algorithm based on nonlinear adjustment strategy of control parameter[J]. Journal of Air Force Engineering University (Natural Science Edition), 2016, 17(3): 6-10. (in Chinese)
魏政磊, 赵辉, 李牧东, 等. 基于控制参数非线性调整策略的灰狼优化算法[J]. 空军工程大学学报(自然科学版), 2016, 17(3): 6-10.

[17] ANDRIES P E. Fundamentals of computational swarm intelligence[M]. Tsinghua University Press, 2009.

[18] DIGALAKIS J, SMUTNICKI C. On benchmarking functions for genetic algorithm[J]. International Journal of Computer Mathematic, 2001, 77: 841-506.

[19] DERRAC J, GARCIA S, MOLINA D, et al. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms [J]. Swarm Evolution Computation, 2011, 1(1): 3-18.