

求解高维优化问题的混合灰狼优化算法

龙文^{1a,1b}, 蔡绍洪^{1a}, 焦建军^{1b}, 张文专^{1a}, 唐明珠²

(1. 贵州财经大学 a. 贵州省经济系统仿真重点实验室, b. 数学与统计学院, 贵阳 550025; 2. 长沙理工大学 能源与动力工程学院, 长沙 410114)

摘要: 提出一种基于混沌和精英反向学习的混合灰狼优化算法以解决高维优化问题. 首先, 采用混沌序列产生初始种群为算法进行全局搜索奠定基础; 然后, 对当前精英个体分别执行精英反向学习策略以协调算法的勘探和开采能力; 最后, 在搜索过程中对决策层个体进行混沌扰动, 以避免算法陷入局部最优的可能性. 选取10个高维(100维、500维和1000维)标准测试函数进行数值实验, 结果表明, 混合灰狼优化算法在求解精度及收敛速度指标上均明显优于对比算法.

关键词: 灰狼优化算法; 混沌映射; 精英反向学习; 高维优化问题

中图分类号: TP273

文献标志码: A

Hybrid grey wolf optimization algorithm for high-dimensional optimization

LONG Wen^{1a,1b}, CAI Shao-hong^{1a}, JIAO Jian-jun^{1b}, ZHANG Wen-zhuan^{1a}, TANG Ming-zhu²

(1a. Guizhou Key Laboratory of Economics System Simulation, 1b. School of Mathematics and Statistics, Guizhou University of Finance and Economics, Guiyang 550025, China; 2. School of Energy and Power Engineering, Changsha University of Science and Engineering, Changsha 410114, China. Correspondent: LONG Wen, E-mail: lw6822@163.com)

Abstract: A hybrid grey wolf optimization(HGWO) algorithm combined chaotic mapping and elite opposition-based learning strategy is proposed for solving unconstrained high-dimensional function optimization problems. In the proposed HGWO algorithm, the chaotic sequence is used to initiate the individuals' position, which can strengthen the diversity of global searching. Then the elite opposition-based learning strategy is applied to the current elite individuals, which can coordinate the exploration and exploitation ability of the proposed HGWO algorithm. Finally, the first three best individuals are disturbed by chaotic mapping in the process of the search so as to avoid the possibility of falling into local optimum. Numerical experiments are conducted on the 10 high-dimensional(100, 500, and 1000 dimension) classical test functions. The simulation results show that the proposed HGWO algorithm has better performance in solution precision and convergence speed.

Keywords: grey wolf optimization algorithm; chaotic mapping; elite opposition-based learning; high-dimensional optimization problems

0 引言

高维优化问题在科学研究和工程应用中广泛存在, 如大规模无线传感器布局优化问题、航天器设计优化问题等. 高维优化问题一般是指维度超过100的函数优化问题. 随着搜索空间维数的增加, 这些问题的复杂度以指数级增长, 在求解时需要耗费大量的

计算代价, 从而出现“维数灾难”问题^[1-2], 因此, 传统优化方法难以对其进行有效地处理.

由于智能优化算法是一类基于群体迭代的优化技术, 较适宜用来解决高维优化问题, 利用智能优化算法求解高维优化问题已成为优化领域及进化计算领域的研究热点之一. 刘星宝等^[3]对随机克隆算子和

收稿日期: 2015-09-22; 修回日期: 2015-12-25.

基金项目: 国家自然科学基金项目(61463009, 61403046); 教育部人文社会科学研究规划基金项目(12XJA910001); 贵州省教育厅125重大科技专项项目(黔教重大专项2012011).

作者简介: 龙文(1977—), 男, 教授, 博士, 从事复杂工业生产建模与控制、智能优化算法等研究; 蔡绍洪(1958—), 男, 教授, 博士生导师, 从事复杂系统建模与仿真等研究.

超变异算子进行改进,提出了一种新型免疫进化算法用于求解高维优化问题,结果表明,改进算法能有效地提高整体寻优性能;葛方振等^[4]提出了一种扰动混沌蚁群算法用于求解高维优化问题,该算法通过改进贪婪规则和随机邻居选择方法来降低计算复杂度,引入自适应扰动策略以提高搜索精度;Rao等^[5]提出一种新型的教学优化算法用于求解连续非线性高维优化问题,获得了满意的结果;王旭等^[6]将状态观测器和随机分组策略组成的协同进化思想引入差分进化算法中,提出了一种解决高维优化问题的改进差分进化算法;Tang等^[7]提出了一种基于文化基因算法和记忆机制的改进量子粒子群优化算法用于求解高维优化问题。

灰狼优化(GWO)算法是由Mirjalili等^[8]于2014年提出的一种新型群体智能优化算法,它源于对自然界中灰狼种群的等级层次机制和捕食行为的模拟,通过狼群跟踪、包围、追捕、攻击猎物等过程实现优化的目的。GWO算法具有原理简单、参数设置少、较强的全局搜索能力等特点,在函数优化方面,已被证明在求解精度和收敛速度上均优于粒子群优化算法^[8]。自提出以来,GWO算法已被广泛地应用于函数优化^[9-10]、传感器网络训练^[11]、直流电机最优控制^[12]、经济调度指派问题^[13]等领域中。

尽管GWO算法在求解函数优化方面取得了成功,但是它们均只针对低维(小于100维)函数优化问题。目前,尚未见到将GWO算法用于解决高维函数优化问题的相关报道。因此,本文对GWO加以改进,将HGWO算法用于求解高维优化问题。10个标准高维(100, 500, 1 000维)测试函数的仿真结果表明,HGWO算法能有效地解决高维优化问题。

1 基本灰狼优化算法

1.1 灰狼种群等级和捕食行为

灰狼是顶级食肉动物,其生活方式大多以群居为主,其行为构成了灰狼种群等级金字塔,并具有严格的等级管理制度,如图1所示。

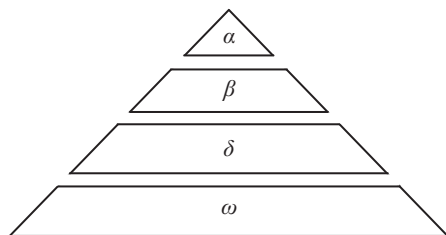


图1 灰狼种群等级金字塔示意

金字塔第1层为种群中的头狼,称为 α ,主要负责群体各项决策事务;金字塔第2层称为 β ,它协助做出管理决策;金字塔第3层为 δ ,主要负责侦察、放哨、

捕猎、看护等事务;金字塔最底层称为 ω ,主要负责平衡种群内部关系。

灰狼的种群等级在实现群体高效捕杀猎物的过程中发挥着至关重要的作用,捕食过程由 α 带领完成。首先,狼群以团队模式搜索、跟踪、靠近猎物;然后,从各个方位包围猎物,当包围圈足够小且完善时,狼群在 α 的指挥下由猎物最近的 β 、 δ 展开进攻,当猎物逃跑时,其余个体进行补给,实现群狼包围圈的跟随变换移动,从而对猎物不断实施各个方向的攻击;最终捕获猎物。

1.2 GWO算法描述

在GWO算法中,由 α 、 β 、 δ 执行追捕行为, ω 跟随前3者进行猎物跟踪围剿,最终完成捕食任务。利用GWO算法求解连续函数优化问题时,假设灰狼种群中的灰狼数目为 N ,搜索空间为 d 维,其中第 i 只灰狼在 d 维空间中的位置可表示为 $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$,种群中当前最优个体记为 α ,将适应度值排名第2及第3的对应个体记为 β 和 δ ,剩余个体记为 ω ,猎物的位置对应于优化问题的全局最优解。

在捕食过程中,灰狼群体根据下式对猎物进行包围:

$$D = |C \cdot X_p(t) - X(t)|. \quad (1)$$

其中: $X_p(t)$ 表示第 t 代时猎物的位置; $X(t)$ 表示第 t 代时灰狼个体的位置;常数 C 为摆动因子,由下式决定:

$$C = 2r_1, \quad (2)$$

r_1 为 $[0, 1]$ 之间的随机数。

灰狼位置更新公式为

$$X(t+1) = X_p(t) - A \cdot D, \quad (3)$$

其中 A 为收敛因子,由下式决定:

$$A = 2ar_2 - a. \quad (4)$$

这里: r_2 为 $[0, 1]$ 之间的随机数, a 随着迭代次数增加从2线性递减到0。

当灰狼判断出猎物所在位置时,将由头狼 α 带领 β 和 δ 发动追捕行为。在狼群中, α 、 β 、 ω 三者最靠近猎物,可以利用这3者的位置来判断猎物所在方位,其数学描述如下:

$$D_\alpha = |C_1 \cdot X_\alpha(t) - X(t)|, \quad (5)$$

$$D_\beta = |C_2 \cdot X_\beta(t) - X(t)|, \quad (6)$$

$$D_\delta = |C_3 \cdot X_\delta(t) - X(t)|, \quad (7)$$

$$X_1 = X_\alpha - A_1 \cdot D_\alpha, \quad (8)$$

$$X_2 = X_\beta - A_2 \cdot D_\beta, \quad (9)$$

$$X_3 = X_\delta - A_3 \cdot D_\delta, \quad (10)$$

$$X_p(t+1) = \frac{X_1 + X_2 + X_3}{3}. \quad (11)$$

由式(5)~(10)计算出群内个体与 α 、 β 、 ω 的距离,然后由式(11)即可综合判断出个体向猎物移动的方向.

2 混合灰狼优化算法

2.1 混沌序列初始化

文献[14]认为,多样性较好的初始种群对提高算法的搜索效率很有帮助.然而,基本GWO算法在求解优化问题时通常在搜索空间中随机初始化个体作为初始种群,这样可能导致初始种群多样性较差.另外,由于对优化问题的全局最优解没有任何先验知识,在初始化种群时应尽可能使个体均匀分布在搜索空间中.

混沌是一种非线性现象,具有随机性、遍历性及规律性等特点,利用其产生初始种群能具有较好的多样性.其基本思想是,通过映射规则将优化变量映射到混沌变量空间的取值区间内,然后将获得的优化解线性转化到优化空间.产生混沌序列的模型有很多,最常用的是Logistic映射模型,因此,本文采用Logistic映射模型产生混沌序列来进行种群初始化,其数学表达式为^[15]

$$y_i^{j+1} = 4y_i^j(1 - y_i^j). \quad (12)$$

其中: $y_i^j \in [0, 1]$ 为混沌变量; $i = 1, 2, \dots, n$ 表示混沌变量的序号; $j = 1, 2, \dots, N$ 表示种群序号; N 表示种群规模.

利用Logistic混沌映射模型对初始值敏感的特点,给式(12)赋 n 个微小差异的初始值,即可得到 n 个混沌变量 y_i^j ,依次取 $j = 1, 2, \dots, N$ 便可得到 N 个初始种群个体.然后利用下式将 n 个混沌变量 y_i^j 作逆映射得到相应优化空间中的变量 x_i^j :

$$x_i^j = l_i + (u_i - l_i) \cdot y_i^j, \quad (13)$$

其中 $[l_i, u_i]$ 为 x_i 的变化范围.

2.2 精英反向学习策略

反向学习策略是由学者Tizhoosh^[16]于2005年提出的一种智能计算方法,目前已广泛应用于遗传算法、粒子群算法、差分进化算法、人工蜂群算法、蚁群算法等群体智能优化算法中以提高算法的搜索性能.基于反向学习策略的有效性,Wang等^[17]引入一般反向因子,提出了一般反向学习策略的概念.实验结果表明,一般反向学习策略比反向学习策略具有更优越的性能^[17].

汪慎文等^[18]引入精英学习的思想,在一般反向学习策略的基础上提出了精英反向学习策略.实验结果表明,精英反向学习策略比一般反向学习策略具有更好的性能^[18].

定义 1 精英反向学习. 设 $x_i(k)$ 和 $x_i^*(k)$ 分别为第 k 代的当前解和反向解,设 $x_{i,j}(k)$ 和 $x_{i,j}^*(k)$ 分别为 $x_i(k)$ 和 $x_i^*(k)$ 的第 j 维上的值. $m(2 \leq m \leq N)$ 个精英个体表示为 $\{e_1(k), e_2(k), \dots, e_m(k)\} \subseteq \{x_1(k), x_2(k), \dots, x_N(k)\}$,则 $x_{i,j}^*(k)$ 定义为

$$x_{i,j}^*(k) = \lambda(a_j(k) + b_j(k)) - x_{i,j}(k). \quad (14)$$

其中: $\lambda \in (0, 1)$ 是随机数, $a_j(k) = \min(e_{1,j}(k), \dots, e_{m,j}(k))$, $b_j(k) = \max(e_{1,j}(k), \dots, e_{m,j}(k))$.

需要说明的是,若 $x_{i,j}^*(k) > b_j(k)$,则取 $x_{i,j}^*(k) = b_j(k)$;若 $x_{i,j}^*(k) < a_j(k)$,则取 $x_{i,j}^*(k) = a_j(k)$.

为了协调算法的勘探和开采能力,本文对当前种群中的前 m 个精英个体执行精英反向学习策略.当 $m = 0.1 \times N$ 时,精英反向学习策略表现出最佳性能^[18].由精英反向学习策略可知,一方面将反向种群与当前种群合并,选出优秀个体进入下一代群体中以增强种群的多样性,能降低算法陷入局部最优的概率;另一方面,它又充分吸收了当前种群中精英个体的有益搜索信息,可以加快算法的收敛速度^[18].由此可见,精英反向学习策略可平衡算法的勘探和开采能力.

2.3 混沌扰动策略

在基本GWO算法中, α 、 β 和 ω 作为种群的决策层,吸引整个狼群靠近并捕获目标.然而,与其他群智能优化算法类似,基本GWO算法在进化后期也存在易陷入局部最优、搜索精度不够高等缺点.为了克服这些缺点,本文利用混沌序列的随机性、规律性和遍历性对决策层中的3个个体进行混沌扰动,可使GWO算法跳出局部最优,从而增强GWO算法的全局搜索能力,进而提高GWO算法的求解精度.混沌扰动策略如下:

$$y'_k = (1 - \tau)y^* + \tau y_k. \quad (15)$$

其中: y'_k 为添加扰动后形成的混沌变量; y^* 为决策层中个体映射到 $[0, 1]$ 后形成的混沌变量; y_k 为迭代 k 次后的混沌变量; $\tau \in [0, 1]$ 为扰动的强度,即

$$\tau = 1 - \left(\frac{k-1}{k}\right)^N. \quad (16)$$

这里: k 为迭代次数, N 为种群规模.

2.4 HGWO算法步骤

Step 1: 设置算法参数,种群规模 N ,最大迭代次数,在搜索空间中随机生成 a 、 A 和 C 等参数.

Step 2: 在搜索空间中按2.1节所描述的方法产生初始种群,令 $k = 1$.

Step 3: 计算群体中每个灰狼个体的适应度值,并将所有适应度值进行排序,将适应度值排列前3位的灰狼个体位置分别记为 X_α 、 X_β 和 X_δ ,作为决策层.

Step 4: 由式(5)~(7)计算群体中其他灰狼个体分别与 X_α 、 X_β 和 X_δ 的距离, 并根据式(8)~(11)更新每个灰狼个体的位置.

Step 5: 利用 2.2 节所描述的方法对当前群体中的前 m 个精英个体执行精英反向学习策略, 并比较更新每个灰狼个体的位置.

Step 6: 对决策层中的个体 X_α 、 X_β 和 X_δ 执行 2.3 节所描述的混沌扰动策略以产生新的个体.

Step 7: 更新 a 、 A 和 C 等参数的值.

Step 8: 判断算法是否满足收敛条件. 若是, 则算法结束; 否则, 令 $k = k + 1$, 返回 Step 3.

3 数值实验及分析

3.1 测试函数

为验证 HGWO 算法的有效性, 本文从文献[8]中选取 10 个高维优化函数, 即 Sphere (f_1), Schwefel 2.22 (f_2), Schwefel 2.21 (f_3), Rosenbrock (f_4), Step (f_5), Quartic (f_6), Rastrigin (f_7), Ackley (f_8), Griewank (f_9)

和 Penalized 1 (f_{10}) 函数进行测试. 10 个测试函数的基本特征(函数表达式、搜索范围)详见文献[8], 其中: $f_1 \sim f_4$ 为单峰函数, f_5 为非连续函数, f_6 为带噪声函数, $f_7 \sim f_{10}$ 为多峰函数. 10 个测试函数的理论最优值均为 0.

3.2 实验结果及分析

利用 HGWO 算法对上述 10 个测试函数进行求解. 10 个函数的维数分别设置为 100 维、500 维和 1000 维, 并与基本 GWO 算法、万有引力搜索算法(GSA)和粒子群优化(PSO)进行比较. 经过反复多次数值实验, 确定 GWO 和 HGWO 算法最优参数设置如下: 种群规模均设置为 $N = 50$, 最大迭代次数均设置为 1000, 收敛精度设置为 $1e-05$; GSA 和 PSO 算法的参数直接来源于参考文献, 即在 GSA 中, $G_0 = 100$, $\beta = 20$; 在 PSO 中, $c_1 = c_2 = 2$, $w = 0.7294$. 采用 4 种算法分别对 10 个测试函数分别运行 30 次, 记录其平均结果和标准差, 如表 1 所示.

表 1 4 种算法对 10 个测试函数的寻优结果比较

函数	维数	GSA		PSO		GWO		HGWO	
		平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
f_1	100	7.27e+01	4.69e+01	4.63e-01	9.21e-01	4.67e-07	8.09e-07	2.66e-35	1.09e-34
	500	2.35e+04	1.81e+03	1.72e+03	1.51e+02	2.22e-03	4.69e-03	3.07e-14	1.51e-14
	1000	5.73e+04	3.43e+03	1.37e+04	9.31e+02	6.14e-01	2.58e-01	4.62e-10	2.73e-10
f_2	100	7.83e-01	3.76e-01	6.17e+00	1.60e+00	2.46e-05	6.52e-05	7.51e-21	6.05e-21
	500	1.56e+02	7.40e+00	9.80e+02	2.26e+02	4.61e-01	5.25e-01	5.61e-09	1.78e-09
	1000	4.36e+04	3.88e+03	1.15e+03	2.61e+01	4.46e+00	4.60e+00	1.81e-05	1.00e-05
f_3	100	1.02e+01	1.22e+00	7.28e+00	8.19e-01	1.28e+00	9.07e-01	3.54e-05	2.50e-05
	500	1.86e+01	1.08e+00	2.46e+01	8.71e-01	4.87e+00	4.49e+00	1.81e-03	2.09e-03
	1000	2.14e+01	9.34e-01	3.09e+01	1.96e+00	6.97e+01	2.29e+00	5.77e-01	7.03e-02
f_4	100	1.35e+03	5.95e+02	1.14e+03	4.61e+02	9.72e+01	8.99e+00	6.14e+00	7.00e-01
	500	1.79e+06	2.13e+05	6.71e+06	1.05e+06	4.97e+02	4.47e-01	9.25Ee+01	5.22e-01
	1000	6.02e+06	5.60e+05	9.35e+07	8.60e+06	4.08e+03	3.61e+03	9.96e+02	0.00e+00
f_5	100	3.73e+02	2.01e+02	7.46e-01	6.59e-01	7.00e+00	9.10e-01	1.03e-05	2.18e-06
	500	2.59e+04	1.34e+03	1.59e+03	1.36e+02	8.83e+01	1.20e+00	3.76e-02	5.87e-03
	1000	6.03e+04	2.61e+03	1.33e+04	6.54e+02	2.02e+02	2.30e+00	8.49e+00	2.75e+00
f_6	100	5.76e-01	2.78e-01	1.29e+03	3.01e+02	1.40e-03	1.06e-03	1.77e-08	2.33e-08
	500	1.45e+02	2.58e+01	5.58e+04	2.42e+03	6.49e-03	1.34e-03	4.21e-06	1.90e-06
	1000	1.32e+03	1.41e+02	2.43e+05	5.08e+03	1.44e-02	5.30e-03	6.17e-03	3.53e-03
f_7	100	7.86e+01	1.11e+01	3.77e+02	6.38e+01	3.27e-06	1.81e-06	1.13e-13	0.00e+00
	500	1.46e+03	3.91e+01	5.19e+03	3.24e+02	2.62e+00	1.66e+00	1.01e-10	1.18e-10
	1000	4.60e+03	1.19e+02	1.29e+04	4.08e+02	7.89e+01	1.64e+01	3.95e-07	2.52e-07
f_8	100	9.56e-01	4.64e-01	2.30e+00	4.08e-01	1.67e-06	4.82e-07	7.55e-14	0.00e+00
	500	7.04e+00	1.41e-01	8.58e+00	3.71e-01	3.80e-04	2.83e-05	7.62e-09	2.50e-10
	1000	8.42e+00	7.70e-02	1.37e+01	4.23e-01	7.79e-02	2.03e-02	5.45e-07	7.08e-08
f_9	100	4.85e+01	1.36e+01	2.04e-02	9.57e-03	2.41e-09	3.68e-09	0.00e+00	0.00e+00
	500	4.34e+03	5.50e+01	1.59e+00	2.04e-02	1.65e-05	2.13e-06	5.53e-15	1.92e-15
	1000	1.35e+04	2.77e+02	8.84e+00	1.93e-01	2.17e-02	2.09e-03	2.59e-11	1.16e-11
f_{10}	100	1.95e+00	2.87e-01	9.23e-01	1.03e+00	1.78e-01	2.98e-02	3.13e-06	3.37e-06
	500	1.30e+01	1.37e+00	6.98e+03	4.30e+03	7.00e-01	3.38e-02	3.98e-05	1.69e-05
	1000	2.00e+01	1.41e+00	1.43e+06	3.39e+05	8.48e-01	1.50e-02	2.83e-03	4.10e-03

由表 1 可知, 在函数维数设置为 100 维的实验中, 除了函数 f_4 , HGWO 算法在其他 9 个测试函数 30 次独立实验中均能一致地收敛到全局最优解(只要结果满足固定收敛精度 $1e-05$ 就认为收敛), 尤其是函数 f_9 , HGWO 算法能收敛到理论最优值 0; 另外, HGWO 在 9 个函数上的标准差值较小, 说明其鲁棒性较强. 由表 1 可知: 在函数维数设置为 500 维的实验中, HGWO 算法在 7 个函数($f_1, f_2, f_6, f_7, f_8, f_9$ 和 f_{10})上能收敛到全局最优解; 在函数维数设置为 1000 维的实验中, HGWO 算法在 5 个函数(f_1, f_2, f_7, f_8 和 f_9)上能收敛到全局最优解; 随着函数维数的增加, HGWO 算法的寻优性能略有下降.

由表 1 可知, 与 GSA、PSO 和 GWO 算法相比, 不管是 100 维, 500 维, 还是 1000 维, HGWO 算法在 10 个测试函数上均获得了较好的寻优结果. 对于 10 个高维(100, 500 和 1000 维)函数, GSA 和 PSO 均不能收敛到全局最优解, 说明 GSA 和 PSO 求解高维优化问题比较困难; GWO 则在 100 维的 5 个函数(f_1, f_2, f_7, f_8 和 f_9)上能收敛到全局最优解, 而在 500 维和 1000 维的函数中基本上收敛不到全局最优解, 这说明 GWO 算法求解高维优化问题的能力还有待改进.

利用 Wilcoxon 秩和检验来分析 HGWO 算法与对比算法性能差异的显著性. 表 1 中, 给定原假设为 $M_{HGWO} = M_{GSA}, M_{HGWO} = M_{PSO}, M_{HGWO} = M_{GWO}$, 备用假设为 $M_{HGWO} < M_{GSA}, M_{HGWO} < M_{PSO}$ 和 $M_{HGWO} < M_{GWO}$, 其中, $M_{HGWO}, M_{GSA}, M_{PSO}, M_{GWO}$ 分别是 4 种算法所测试的 10 个函数均值的中位数. 在选定 1% 的显著性水平下, 所得的 P-value 分别为 0.000 245 5、0.000 326 和 0.036 17, 说明前面两个检验是显著的, 即认为 HGWO 比 GSA 和 PSO 具有更高的收敛精度; 在 5% 的显著性水平下, HGWO 算法比 GSA、PSO、GWO 具有更高的收敛精度.

限于篇幅, 图 2 只给出了 5 个测试函数(Sphere, Schwefel2.22, Rastrigin, Ackley 和 Griewank 函数) 100 维时的进化收敛曲线. 从图 2 中可以清晰地看出, 与 GSA、PSO 和 GWO 算法相比, HGWO 算法具有更快的收敛速度和更高的收敛精度.

为了更进一步验证 HGWO 算法的有效性, 将其与 3 种求解高维优化问题的改进差分进化算法, 即 Self adaptive differential evolution(记为 SaDE)^[19]、Differential evolution with ensemble strategies(记为 DEES)^[20]和 Multi-population differential evolution with balanced ensemble strategies(记为 mDE-bES)^[20]进行比较, 比较结果如表 2 所示. 其中, 为了比较的公平性, 3 种算法的参数设置和实验结果直接来源于各自文献.

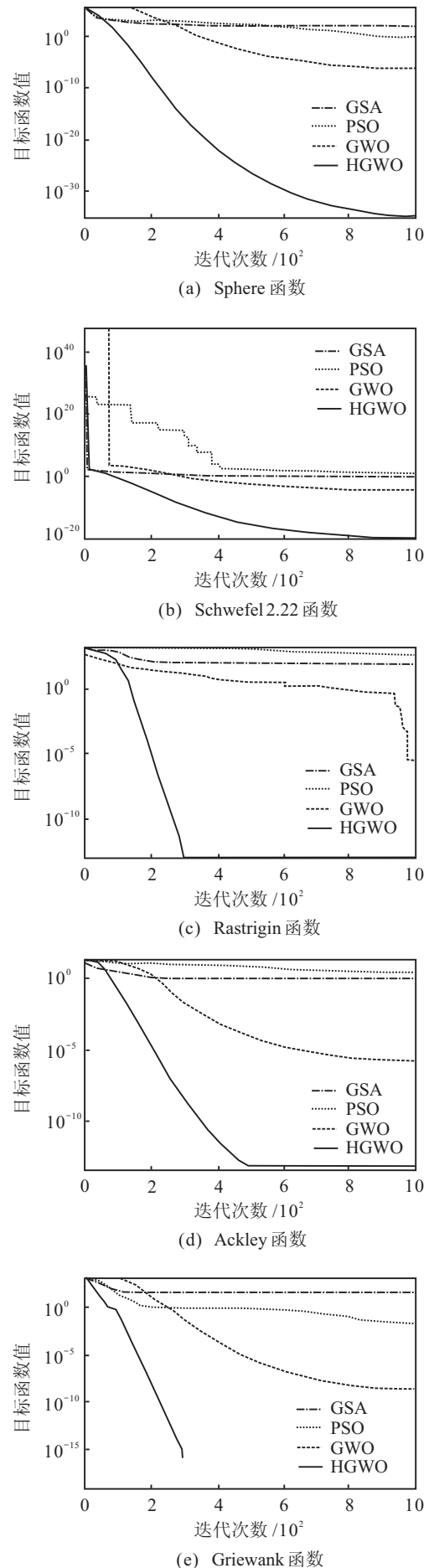


图 2 4 种算法对 5 个函数的收敛曲线

表2 4种算法对7个高维函数的平均值比较

函数	维数	SaDE	DEES	mDE-bES	HGWO
f_1	100	3.13e+01	1.41e-13	0.00e+00	2.66e-35
	500	1.34e+01	6.16e-13	3.92e-13	3.07e-14
	1000	3.49e+01	1.29e-12	8.24e-13	4.62e-10
f_2	100	0.00e+00	0.00e+00	0.00e+00	7.51e-21
	500	0.00e+00	0.00e+00	0.00e+00	5.61e-09
	1000	0.00e+00	8.77e+00	0.00e+00	1.81e-05
f_3	100	1.26e+02	3.92e+01	4.00e+01	3.54e-05
	500	9.23e+01	5.69e+01	4.56e+01	1.81e-02
	1000	1.43e+02	7.94e+01	5.97e+01	5.77e-01
f_4	100	1.11e+05	4.40e+01	4.90e-01	6.14e+00
	500	2.62e+04	5.11e+02	4.16e+02	9.25e+01
	1000	1.62e+05	9.65e+02	9.00e+02	9.96e+02
f_7	100	1.58e+01	1.49e+01	1.87e+01	1.13e-13
	500	1.31e+00	1.68e+01	1.91e-11	1.10e-10
	1000	3.21e+01	1.69e+01	4.03e+01	3.95e-07
f_8	100	8.32e-02	2.72e-01	1.44e-13	7.55e-14
	500	4.63e-01	8.81e-13	3.56e-14	7.62e-09
	1000	4.28e-01	1.85e-12	1.28e-12	5.42e-07
f_9	100	3.53e-01	1.38e-03	0.00e+00	0.00e+00
	500	7.48e-01	1.28e-03	1.83e-13	5.53e-15
	1000	6.33e+01	6.38e-13	0.00e+00	2.59e-11

由表2可知:与SaDE算法相比,HGWO在6个函数(f_1 、 f_3 、 f_4 、 f_7 、 f_8 和 f_9)上获得了较好的寻优结果,而SaDE算法则在函数 f_2 上得到了较好的结果;与DEES算法相比,HGWO算法在函数 f_3 、 f_4 、 f_7 和 f_9 上获得了较好的结果.对于测试函数 f_1 ,HGWO算法在100维和500维上取得了较好的结果,而在1000维上得到的结果要比DEES算法差;对于函数 f_2 ,DEES算法在100维和500维上获得了较好的结果,而HGWO算法则在1000维上取得了较优的结果;对于函数 f_8 ,DEES算法在500维和1000维上获得了较好的结果.与mDE-bES算法相比,对于函数 f_1 、 f_2 和 f_8 ,HGWO算法获得了稍差的结果;对于测试函数 f_3 、 f_4 和 f_7 ,HGWO算法取得了较好的结果;对于函数 f_9 ,两种算法获得了相似的结果.由以上比较结果可知,HGWO算法在求解大部分函数上具有一定的优势.

4 结 论

灰狼优化算法是最近提出的一种新型群体智能优化方法,其思路是基于对灰狼群体捕食行为的模拟.针对标准GWO算法求解高维函数优化问题存在的不足,本文提出了一种混合GWO算法,即HGWO算法.对10个高维函数进行仿真实验,结果表明,与GSA、PSO、GWO、SaDE、DEES和mDE-bES算法相

比,HGWO算法在大部分函数上获得了较高的寻优精度,能有效地处理高维函数优化问题.下一步研究内容是将HGWO算法应用于实际工程应用中.

参考文献(References)

- [1] Rahnamayan S, Wang G. Solving large scale optimization problems by opposition-based differential evolution[J]. WSEAS Transactions on Computers, 2008, 7(10): 1792-1804.
- [2] Wang Y, Dang C. An evolutionary algorithm for global optimization based on level-set evolution and Latin squares[J]. IEEE Trans on Evolutionary Computation, 2007, 11(5): 579-595.
- [3] 刘星宝,蔡自兴,王勇,等.应用于高维优化问题的免疫进化算法[J].控制与决策,2011,26(1):59-64.
(Liu X B, Cai Z X, Wang Y, et al. Novel immune evolutionary algorithm for global optimization[J]. Control and Decision, 2011, 26(1): 59-64.)
- [4] 葛方振,魏臻,田一鸣,等.求解高维优化问题的扰动混沌蚁群优化算法[J].计算机应用,2011,31(4):1084-1089.
(Ge F Z, Wei Z, Tian Y M, et al. High-dimensional optimization problems via disturbance chaotic ant swarm algorithm[J]. J of Computer Applications, 2011, 31(4): 1084-1089.)
- [5] Rao R, Savsani V, Vakharia D. Teaching-learning-based optimization: An optimization method for continuous nonlinear large scale problems[J]. Informance Sciences, 2012, 183(1): 1-15.
- [6] 王旭,赵曙光.解决高维优化问题的差分进化算法[J].计算机应用,2014,34(1):179-181.
(Wang X, Zhao S G. Differential evolution algorithm for high-dimensional optimization problem[J]. J of Computer Applications, 2014, 34(1): 179-181.)
- [7] Tang D, Cai Y, Zhao J, et al. A quantum-behaved particle swarm optimization with memetic algorithm and memory for continuous nonlinear large scale problems[J]. Informance Sciences, 2014, 289(36): 162-189.
- [8] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69(7): 46-61.
- [9] Saremi S, Mirjalili S Z, Mirjalili S M. Evolutionary population dynamics and grey wolf optimizer[J]. Neural Computation and Applications, 2015, 26(5): 983-989.
- [10] 龙文,赵东泉,徐松金.求解约束优化问题的改进灰狼优化算法[J].计算机应用,2015,35(9):2590-2595.
(Long W, Zhao D Q, Xu S J. Improved grey wolf optimization algorithm for constrained optimization problem[J]. J of Computer Applications, 2015, 35(9): 2590-2595.)

- [11] Mirjalili S. How effective is the grey wolf optimizer in training multilayer perceptrons[J]. *Applied Intelligence*, 2015, 42(4): 608-619.
- [12] Madadi A, Motlagh M. Optimal control of DC motor using grey wolf optimizer algorithm[J]. *Technical J of Engineering and Applied Science*, 2014, 4(4): 373-379.
- [13] Song H, Sulaiman M, Mohamed M. An application of grey wolf optimizer for solving combined economic emission dispatch problems[J]. *Int Review on Modeling and Simulation*, 2014, 7(5): 838-844.
- [14] Haupt R, Haupt S. *Practical genetic algorithm*[M]. New York: John Wiley and Sons, 2004: 38-39.
- [15] 杨景明, 马鸣鸣, 车海军, 等. 多目标自适应混沌粒子群优化算法[J]. *控制与决策*, 2015, 30(12): 2168-2174. (Yang J M, Ma M M, Che H J, et al. Multi-objective adaptive chaotic particle swarm optimization algorithm[J]. *Control and Decision*, 2015, 30(12): 2168-2174.)
- [16] Tizhoosh H. Opposition-based learn: A new scheme for machine intelligence[C]. *Proc of Int Conf on Computational Intelligence for Modeling Control and Automation*. Vienna: IEEE Press, 2005: 695-701.
- [17] Wang H, Wu Z, Rahamayan S, et al. Enhancing particle swarm optimization using generalized opposition-based learning[J]. *Informances Sciences*, 2011, 181(20): 4699-4714.
- [18] 汪慎文, 丁立新, 谢承旺, 等. 应用精英反向学习策略的混合差分进化算法[J]. *武汉大学学报: 理学版*, 2013, 59(2): 111-116. (Wang S W, Ding L X, Xie C W, et al. A hybrid differential evolution with elite opposition-based learning[J]. *J of Wuhan University: Natural Science Edition*, 2013, 59(2): 111-116.)
- [19] Qin K, Huang V, Suganthan P. Differential evolution algorithm with strategy adaptation for global numerical optimization[J]. *IEEE Trans on Evolutionary Computation*, 2009, 13(2): 398-417.
- [20] Ali M, Awad N, Suganthan P. Multi-population differential evolution with balanced ensemble of mutation strategies for large-scale global optimization[J]. *Applied Soft Computing*, 2015, 33(3): 304-327.

(责任编辑: 李君玲)