

控制与决策

Control and Decision

求解高维复杂函数的遗传-灰狼混合算法

顾清华, 李学现, 卢才武, 阮顺领

引用本文:

顾清华, 李学现, 卢才武, 等. 求解高维复杂函数的遗传-灰狼混合算法[J]. 控制与决策, 2020, 35(5): 1191-1198.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2018.1253>

您可能感兴趣的其他文章

Articles you may be interested in

基于自适应差分进化的常压塔轻质油产量多目标优化

Multi-objective optimization of light oil production in atmospheric distillation column based on self-adaptive differential evolution

控制与决策. 2020, 35(3): 604-612 <https://doi.org/10.13195/j.kzyjc.2018.0794>

基于精英混沌搜索策略的交替正弦余弦算法

Alternating sine cosine algorithm based on elite chaotic search strategy

控制与决策. 2019, 34(8): 1654-1662 <https://doi.org/10.13195/j.kzyjc.2018.0006>

基于拓扑结构与粒子变异改进的粒子群优化算法

Modified particle swarm optimization algorithms based on topology and particle mutation

控制与决策. 2019, 34(2): 419-428 <https://doi.org/10.13195/j.kzyjc.2017.1063>

基于改进烟花-蚁群混合算法的智能移动体避障路径规划

Obstacle avoidance path planning of intelligent mobile based on improved fireworks-ant colony hybrid algorithm

控制与决策. 2019, 34(2): 335-343 <https://doi.org/10.13195/j.kzyjc.2017.0870>

基于频繁覆盖策略的随机漂移粒子群优化算法

Random drift particle swarm optimization with frequent coverage strategy

控制与决策. 2017, 32(12): 2127-2136 <https://doi.org/10.13195/j.kzyjc.2016.1402>

基于最优高斯随机游走和个体筛选策略的差分进化算法

Differential evolution based on optimal Gaussian random walk and individual selection strategies

控制与决策. 2016, 31(8): 1379-1386 <https://doi.org/10.13195/j.kzyjc.2015.0779>

一种狼群智能算法及收敛性分析

A smart wolf pack algorithm and its convergence analysis

控制与决策. 2016, 31(12): 2131-2139 <https://doi.org/10.13195/j.kzyjc.2015.1202>

基于分治策略的改进人工蜂群算法

Modified artificial bee colony algorithm based on divide-and-conquer strategy

控制与决策. 2015(2): 316-320 <https://doi.org/10.13195/j.kzyjc.2013.1442>

求解高维复杂函数的遗传-灰狼混合算法

顾清华[†], 李学现, 卢才武, 阮顺领

(西安建筑科技大学 管理学院, 西安 710055)

摘要: 高维函数优化一般是指维数超过 100 维的函数优化问题, 由于“维数灾难”的存在, 求解起来十分困难. 针对灰狼算法迭代后期收敛速度慢, 求解高维函数易陷入局部最优的缺点, 在基本灰狼算法中引入 3 种遗传算子, 提出一种遗传-灰狼混合算法 (hybrid genetic grey wolf algorithm, HGGWA). 混合算法能够充分发挥两种算法各自的优势, 提高算法的全局收敛性, 针对精英个体的变异操作有效防止算法陷入局部最优值. 通过 13 个标准测试函数和 10 个高维测试函数验证算法的性能, 并将优化结果与 PSO、GSA、GWO 三种基本算法以及 9 种改进算法进行比较. 仿真结果表明, 所提算法在收敛精度方面得到了极大改进, 验证了 HGGWA 算法求解高维函数的有效性.

关键词: 高维复杂函数优化; 灰狼优化算法; 遗传算子; 反向学习; 种群划分

中图分类号: TP301.6

文献标志码: A

Hybrid genetic grey wolf algorithm for high dimensional complex function optimization

GU Qing-hua[†], LI Xue-xian, LU Cai-wu, RUAN Shun-ling

(School of Management, Xi'an University of Architecture and Technology, Xi'an 710055, China)

Abstract: High-dimensional function optimization usually refers to the function optimization problem with dimension over 100, which is difficult to be solved for the existence of “dimension disaster”. In this paper, three genetic operators are embedded into the basic grey wolf algorithm, and a hybrid genetic-grey wolf algorithm (HGGWA) is proposed. The global convergence of the HGGWA is greatly improved by combining the advantages of the GWO and GA. The current three optimal individuals are disturbed by the diversity mutation operator in the process of the search so as to avoid the possibility of falling into local optimum. The performance of the algorithm is verified using 13 standard benchmark functions and 10 high dimensional functions, and the optimization results are compared with the PSO, GSA, GWO and 9 improved algorithms. Simulation results show that the HGGWA is greatly improved in convergence accuracy, which verify the effectiveness of the HGGWA in solving high-dimensional functions.

Keywords: high dimensional function optimization algorithm; grey wolf optimizer; genetic operator; opposition-based learning; population partition

0 引 言

高维复杂函数优化在实际工程问题当中具有十分广泛的应用, 如大规模作业车间调度问题^[1]、大规模电力系统无功优化^[2]等. 高维函数一般是指维数超过 100 维的函数, 具有非线性、高复杂度等特点, 且存在着大量的局部最优解. 随着维数的不断增加, 问题的复杂度会呈指数级的速度增长^[3], 因此求解起来十分困难.

目前, 针对高维函数的优化主要从两个方面来解决: 一是通过降低问题的规模, 采用多种群并行算法进行寻优. 徐东方等^[4]提出了一种多种群子空间学

习粒子群算法, 采用多种群进化模式加快了算法的收敛速度. 拓守恒等^[5]提出了一种并行差分进化算法, 利用种群中各维变量间的相关性进行维度划分, 从而实现降维. 针对大规模非线性函数优化问题, 刘剑英^[6]将协同进化的思想引入差分进化算法, 提出了一种基于 GPU 的协同差分进化算法. 另外, 为了有效地对问题进行分解, Omidvar 等^[7]提出了一种自动分解的策略对大规模优化问题进行求解. 虽然以上改进算法取得了一定的效果, 但是多种群并行算法会随着函数维度的增加, 对高维问题的求解精度有所下降.

在改进算法的研究方面, 通过改善算法的全局

收稿日期: 2018-09-14; 修回日期: 2018-11-20.

基金项目: 国家自然科学基金项目 (51774228, 51404182); 陕西省自然科学基金项目 (2017JM5043); 陕西省教育厅专项科研计划项目 (17JK0425).

责任编委: 陈家伟.

[†]通讯作者. E-mail: qinghuagu@xauat.edu.cn.

收敛性以提高算法对高维函数的寻优能力. 针对粒子群算法求解高维函数容易早熟收敛的问题, 胡成玉等^[8]提出了一种基于动态维度交叉的改进粒子群算法. 另外, 田瑾^[9]在分析量子行为粒子群优化算法 QPSO 机理的基础上, 对 QPSO 算法进行改进, 大大提高了粒子群算法的收敛速度和收敛精度. 针对混合算法的研究, 王忠贵等^[10]将模拟退火算法与几种经典算法相结合, 提出了一种针对高维复杂函数的混合模拟退火全局优化策略. 于万霞等^[11]将遗传算法和粒子群算法相结合, 提出了一种主从结构算法, 很好地发挥了算法的全局搜索和局部搜索的能力. 以上改进算法在全局收敛性和求解精度方面还有待提高, 同时算法中参数的选择对其寻优性能有一定的影响.

灰狼优化算法 (grey wolf optimizer, GWO) 是 Mirjalili 等^[12]提出的一种群体智能优化算法, 具有控制参数少、收敛速度快等优点. 灰狼算法自提出以来, 不同学者对其进行了相关改进研究. 例如, 利用佳点集策略^[13]、混沌序列策略^[14]生成初始种群, 控制参数的非线性调整^[15]等. GWO 算法虽然得到了许多改进, 但对于高维复杂函数问题的研究仍不多见. 因此, 为解决高维复杂函数优化问题, 本文将选择、交叉、变异 3 种遗传算子与 GWO 算法相结合, 提出一种遗传-灰狼混合算法 (hybrid genetic grey wolf algorithm, HGGWA). 算法的主要改进策略如下:

- 1) 采用反向学习策略初始化种群, 利用基本灰狼算法平衡全局搜索和局部搜索;
- 2) 通过精英保留策略与赌轮盘相结合的方式对种群进行选择操作;
- 3) 基于降维和种群划分的方法, 将整个种群划分为多个子种群进行交叉操作, 以增加种群的多样性;
- 4) 对种群中的精英个体进行变异操作, 防止算法陷入局部最优.

经过测试, 改进算法具有较高的收敛精度, 能有效地解决高维函数优化问题.

1 遗传-灰狼混合算法

1.1 基本灰狼算法原理

在 GWO 算法中, α 狼的位置为最优解, β 、 δ 狼的位置为次优解和第 3 优解, ω 狼的位置是剩余的候选解. 在狼群搜寻猎物的整个过程中, ω 狼是追随 α 、 β 、 δ 狼的最佳位置情况, 逐步更新自己的位置, 以此来逐渐接近猎物, 即搜寻最优解的过程. 灰狼的位置更新公式如下:

$$D = |C \times X_p(t) - X(t)|, \quad (1)$$

$$X(t+1) = X_p(t) - A \times D. \quad (2)$$

其中: t 是迭代次数; A 、 C 是系数向量, A 是收敛因子, 用来平衡全局搜索和局部搜索, C 是用来模拟自然界的影响作用; X_p 是猎物位置向量, X 是狼群位置向量. 系数向量 A 和 C 按如下公式计算:

$$A = 2a \times r_1 - a, \quad (3)$$

$$C = 2r_2. \quad (4)$$

其中: r_1 、 $r_2 \in [0, 1]$; a 随着整个迭代过程的进行, 从 2 到 0 逐渐线性减少. 如此, 根据式 (1) 和 (2), 灰狼随机地在靠近猎物的空间内做出移动, 以逐渐接近最优解.

1.2 混合算法的改进策略

1.2.1 基于反向学习的初始化种群策略

对于种群迭代的群智能优化算法而言, 初始化种群的多样性奠定了算法搜索效率的基础, 多样性较好的种群会减少算法的计算时间, 同时提高算法的全局收敛性^[16]. 反向学习是由 Tizhoosh^[17]提出的一种提高算法搜索效率的策略, 利用已知个体位置的对立点生成新的个体位置, 从而增加搜索种群的多样性. 目前, 反向学习策略初始化种群已经成功地应用到多种群体智能优化算法中^[18-19]. 反向点的定义如下: 假设在种群 P 中存在个体 $X(x_1, x_2, \dots, x_D)$, 则每一维度上的元素 $x_i (i = 1, 2, \dots, D)$ 反向点的计算公式为

$$x'_i = l_i + u_i - x_i,$$

其中 l_i 和 u_i 分别是第 i 维度的下界和上界.

根据上述定义, HGGWA 算法采用反向学习策略初始化种群的步骤如下:

- 1) 随机初始化种群 P , 计算每个个体位置 x_i 的反向点 x'_i , 所有反向点构成反向种群 P' ;
- 2) 计算初始种群 P 和反向种群 P' 中每个个体的适应度, 并按照降序进行排列;
- 3) 选取适应度最高的前 N 个灰狼个体作为最终的初始种群.

1.2.2 基于最佳保留的选择操作

对于种群进化的智能优化算法而言, 每一代种群进化的优劣直接影响到算法的寻优效果. 为了将父代种群中优良的个体遗传到下一代, 并且不遭到破坏, 需要将优良个体直接保存下来. 最佳保留选择策略是遗传算法中保存优良个体的一种有效方式, 本文将融入 GWO 算法中, 以提高算法的求解效率. 假设当前种群为 $P(X_1, X_2, \dots, X_D)$, 个体 X_i 的适应度为 F_i , 具体的操作如下:

- 1) 计算个体的适应度值 $F_i (i = 1, 2, \dots, D)$, 并按照降序进行排列.

2) 选取适应度最高的个体直接复制到下一代种群中.

3) 计算剩余个体的适应度总和 S , 以及每个个体被选取的概率 p_i , 即

$$S = \sum_{i=1}^{D-1} F_i, \quad i = 1, 2, \dots, D-1; \quad (5)$$

$$p_i = F_i/S, \quad i = 1, 2, \dots, D-1. \quad (6)$$

4) 计算每个个体的累积适应度值 s_i , 并按照赌轮盘的方式进行选择操作, 直到子代种群中的个体数目与父代种群保持一致.

$$s_i = \sum_{i=1}^i F_i/S, \quad i = 1, 2, \dots, D-1. \quad (7)$$

1.2.3 基于多种群降维的交叉操作

协同进化^[20]是求解高维函数优化问题的一种策略, 此方法采用了“分而治之”的思想, 通过将高维优化问题划分为多个相互独立的低维子问题, 然后利用多个种群分别对子问题进行求解. 在 HGGWA 算法中, 通过借鉴多种群协同进化的思想, 对应于高维优化问题的降维策略, 将整个种群 P 划分为 $v \times \eta$ 个子种群 $P_{i,j}$ ($i = 1, 2, \dots, v, j = 1, 2, \dots, \eta$), 分别对每个子种群的个体进行交叉操作. 这种基于多种群降维的交叉操作很好地保持了种群的多样性, 使得 HGGWA 算法能有效地在解空间内搜索最优解. 经过测试, 每个子种群 $P_{i,j}$ 的最佳规模为 5×5 . 具体的划分方法如下所示.

初始种群 P 为

$$\begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,d} \\ x_{2,1} & x_{2,2} & \dots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{P,1} & x_{P,2} & \dots & x_{P,d} \end{bmatrix}; \quad (8)$$

子种群 $P_{i,j}$ 为

$$\begin{bmatrix} \begin{bmatrix} x_{1,1} & \dots & x_{1,5} \\ \vdots & \ddots & \vdots \\ x_{5,1} & \dots & x_{5,5} \end{bmatrix} & \dots & \begin{bmatrix} x_{1,(d-4)} & \dots & x_{1,d} \\ \vdots & \ddots & \vdots \\ x_{5,(d-4)} & \dots & x_{5,d} \end{bmatrix} \\ \vdots & \ddots & \vdots \\ \begin{bmatrix} x_{(P-4),1} & \dots & x_{(P-4),5} \\ \vdots & \ddots & \vdots \\ x_{P,1} & \dots & x_{P,5} \end{bmatrix} & \dots & \begin{bmatrix} x_{(P-4),(d-4)} & \dots & x_{(P-4),d} \\ \vdots & \ddots & \vdots \\ x_{P,(d-4)} & \dots & x_{P,d} \end{bmatrix} \end{bmatrix} \quad (9)$$

在遗传算法中, 交叉操作具有十分重要的作用, 是产生新个体的主要方式. 在本文的改进算法中, 采用线性交叉的方式对子种群中的每个个体进行交叉操作. 即将子种群中的每个个体 x_i 生成对应的随机数 $r_i \in (0, 1)$, 当 $r_i <$ 交叉概率 P_c 时, 将对应的个体 x_i 两两配对进行交叉操作. 示例如下.

Crossover (p_1, p_2):

1) 随机选择 $\lambda \in (0, 1)$.

2) 两个子代 $c^1 = (c_1^1, \dots, c_D^1)$, $c^2 = (c_1^2, \dots, c_D^2)$

由父代 $p^1 = (p_1^1, \dots, p_D^1)$, $p^2 = (p_1^2, \dots, p_D^2)$ 生成, 公式为

$$c_i^1 = \lambda p_i^1 + (1 - \lambda) p_i^2, \quad i = 1, 2, \dots, D; \quad (10)$$

$$c_i^2 = \lambda p_i^2 + (1 - \lambda) p_i^1, \quad i = 1, 2, \dots, D. \quad (11)$$

1.2.4 针对精英个体的变异操作

由于 HGGWA 算法加入了最优保存策略的选择操作, 在算法的迭代后期, 整个种群中的灰狼个体都聚集在一个较小的最优区域内, 从而导致种群多样性的丧失. 特别是在求解高维多峰函数时, 若当前最优个体为局部最优解, 则算法很容易陷入局部最优. 为此, 本文在 HGGWA 算法中引入了变异算子对种群中的精英个体进行变异操作. 具体操作如下.

假设最优个体为 $x_i = (x_1, x_2, \dots, x_d)$, 以概率 P_m 对当前得到的最优个体进行变异操作, 即从最优个体中以概率 P_m 选择一个基因 x_k , 然后随机生成 $[l, u]$ 范围之间的一个实数替代此基因, 从而生成一个新个体 $x'_i = (x'_1, x'_2, \dots, x'_d)$. 多样性变异算子为

$$x'_i = \begin{cases} l + \lambda(u - l), & i = k; \\ x_i, & i \neq k. \end{cases} \quad (12)$$

其中: λ 是 $[0, 1]$ 之间的随机数, l 和 u 分别是个体 x_i 的下界和上界.

综上所述, 本文所提出的 HGGWA 算法的基本流程如下.

step 1: 初始化参数, 包括种群规模 N , 最大迭代次数 Max_{iter} , 交叉概率 P_c , 变异概率 P_m ;

step 2: 反向学习策略生成初始种群, 计算个体的适应度, 将前 3 个最优个体的位置依次保存为 X_α 、 X_β 和 X_δ ;

step 3: 根据式 (1) 和 (2) 更新灰狼位置信息, 更新参数 a 、 A 、 C 的值;

step 4: 对灰狼种群进行选择、交叉、变异 3 种遗传操作, 从而得到下一代种群, 计算新种群中个体的适应度, 更新 X_α 、 X_β 和 X_δ ;

step 5: 重复 step 3 和 step 4, 直到求得最优解或达到最大迭代次数, 算法终止.

2 HGGWA 算法的全局收敛性分析

在高维复杂优化问题中, 变量之间的支配关系为进化算法的求解过程提供了理论基础, 从进化算法收敛性的角度来说, 如果种群中变量之间的支配关系没有改变, 则算法会保持收敛的鲁棒性^[21]. 对于某一求最大值的函数优化问题 $f(x)$, 假设 s_k^* 和 s_{k+1}^*

分别是第 k 代和第 $k + 1$ 代种群 S_k 和 S_{k+1} 中的最优个体, $f(s_k^*)$ 和 $f(s_{k+1}^*)$ 为其相应的适应度. 即存在 $\forall s_k \in S_k$, 都有 $f(s_k) \leq f(s_k^*)$, 这说明最优个体在整个种群中对其他个体存在支配关系. 在 HGGWA 算法中, 基于最佳保留的选择操作维持了最优个体的支配地位, 即在下一代种群 S_{k+1} 中仍然存在 $f(s_{k+1}) \leq f(s_{k+1}^*)$ 的支配关系. 随着遗传代数 k 的不断增长, 每代最优个体的适应度所组成的数列 $\{f(s_1^*), f(s_2^*), \dots\}$ 为单调不减数列. 通过以下定理来证明 HGGWA 算法的全局收敛性.

定理 1 假设问题 $f(x)$ 的全局最优解为 X^* , 包含全局最优解 X^* 的种群或状态的集合为 E , HGGWA 算法具有全局收敛性的充分条件为: 在搜索过程中, HGGWA 算法能发现全局最优解 X^* , 或经历包含全局最优值的状态 E .

证明 根据最佳保留策略, 若 HGGWA 算法在

第 n 代种群中发现最优解 X^* , 则个体之间的支配关系将一直保持下去, 该最优解将一直保留而不会丢失. 即, 当进化代数 $k > n$ 时, 会一直满足 $f(s_k^*) = f(X^*)$ 的关系, 则存在 $\lim_{k \rightarrow \infty} p(f(s_k^*) = f(X^*)) = 1$, 因此 HGGWA 算法是全局收敛的. \square

3 数值实验及分析

3.1 测试函数

为了测试本文所提出的 HGGWA 算法的性能, 首先选取文献 [12] 中的 13 个标准测试函数 (30 维) 对比测试 GWO 和 HGGWA 两种算法的寻优性能, 以验证改进策略的有效性. 另外, 为了进一步验证 HGGWA 算法求解高维复杂函数的能力, 选取 10 个高维测试函数进行寻优测试, 并将测试结果与 PSO、GSA、GWO 三种算法进行比较. 10 个标准测试函数的具体特征见表 1 所示.

表 1 10 个高维标准测试函数

函数名	函数表达式	搜索范围	f_{\min}	收敛精度
Sphere	$f_1(x) = \sum_{i=1}^d x_i^2$	$[-100, 100]$	0	1×10^{-8}
Schwefel's 2.22	$f_2(x) = \sum_{i=1}^d x_i + \prod_{i=1}^d x_i $	$[-10, 10]$	0	1×10^{-8}
Schwefel's 2.21	$f_3(x) = \max_i \{ x_i , 1 \leq i \leq d\}$	$[-100, 100]$	0	1×10^{-2}
Rosenbrock	$f_4(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]$	0	1×10^0
Schwefel's 1.2	$f_5(x) = \sum_{i=1}^d \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]$	0	1×10^{-3}
Quartic	$f_6(x) = \sum_{i=1}^d ix_i^4 + \text{random}[0, 1)$	$[-1.28, 1.28]$	0	1×10^{-4}
Rastrigin	$f_7(x) = \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$	0	1×10^{-8}
Ackley	$f_8(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i)\right) + 20$	$[-32, 32]$	0	1×10^{-5}
Griewank	$f_9(x) = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$	0	1×10^{-5}
Penalized 1	$f_{10}(x) = \frac{\pi}{d} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{d-1} (y_i - 1)^2 [1 + \sin^2(\pi y_{i+1})] + (y_d - 1)^2 \right\} + \sum_{i=1}^d u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-50, 50]$	0	1×10^{-2}

为了更清晰地表达算法的寻优效果, 本文选取两个指标评价算法的性能^[22]. 一是求解精度 (accuracy, AC), 该指标反映了算法寻优的结果与理论最优值之间的差距. 在一次实验中, 如果算法最终的收敛结果小于求解精度 AC, 则认为此次寻优成功. 假设某次寻

优的最优值为 X_{opt} , 理论最优值为 X_{min} , 则 AC 的计算方式如下:

$$AC = |X_{\text{opt}} - X_{\text{min}}|. \quad (13)$$

二是寻优成功率 (successful ratio, SR), 该指标反映了算法在求解精度一定的条件下, 寻优成功的次数占总

寻优次数的比重. 假设在算法的 T 次迭代中, 寻优成功的次数为 t 次, 则 SR 的计算方式如下:

$$SR = \left\lfloor \frac{t}{T} \times 100\% \right\rfloor. \quad (14)$$

3.2 数值结果及分析

为了保证比较结果的公平性, 几种算法均采用相同的参数设置, 其中种群规模 $N = 50$, 最大迭代次数 $Max_{iter} = 1000$. 另外, 在 PSO 中, $c_1 = c_2 = 2$, $\omega = 0.7294$; 在 GSA 中, $G_0 = 100$, $\beta = 20$; 在 GWO 中, 参数 a 的初始值为 2; HGGWA 中交叉概率 $P_c = 0.8$, 变异概率 $P_m = 0.01$. 每种算法独立运行 30 次, 记录每种算法寻优的最优值、最差值、平均值以及标准差, 并计算寻优成功率. 所有测试均在 Intel Core I5-6300HQ、8 G 内存、2.30 GHz 的计算机上进行, 编写程序在 Matlab 2017b 上实现.

表 2 给出了 GWO、HGGWA 两种算法对 13 个测试函数的寻优结果对比. 通过分析得知, HGGWA 算法在所有 13 个测试函数上均取得了比 GWO 算法更

好的寻优结果. 其中对于函数 f_1 、 f_2 、 f_3 、 f_4 、 f_{10} 、 f_{11} 的收敛精度大大提高, 特别是对于函数 f_9 和 f_{11} , 算法的收敛结果达到了理论最优值 0. 仿真结果表明了 HGGWA 算法改进策略的有效性.

表 2 GWO 算法和 HGGWA 算法对 13 个标准测试函数的寻优结果比较

函数	GWO 算法		HGGWA 算法	
	平均值	标准差	平均值	标准差
f_1	2.66e-43	5.69e-43	9.56e-118	9.46e-118
f_2	2.51e-47	7.13e-47	2.07e-64	3.49e-64
f_3	1.50e-26	2.78e-26	2.63e-30	3.88e-30
f_4	3.04e-24	2.24e-24	2.54e-35	4.85e-35
f_5	2.88e+01	4.23e-01	2.52e+01	3.35e-01
f_6	3.41e+00	1.54e-01	3.19e-04	4.16e-04
f_7	2.09e-03	2.16e-03	7.86e-05	8.25e-05
f_8	-6.03e+03	5.39e+02	-3.12e+03	3.50e+02
f_9	1.19e+02	4.13e+01	0.00e+00	0.00e+00
f_{10}	3.85e+00	8.79e+00	7.99e-15	2.33e-15
f_{11}	0.00e+00	0.00e+00	0.00e+00	0.00e+00
f_{12}	1.47e-01	1.79e-02	4.57e-04	5.77e-04
f_{13}	1.84e+00	2.61e-01	1.15e-02	2.17e-02

表 3 PSO、GSA、GWO、HGGWA 四种算法对 10 个高维测试函数的寻优结果比较

函数	维数	PSO 算法			GSA 算法			GWO 算法			HGGWA 算法		
		平均值	标准差	SR/%	平均值	标准差	SR/%	平均值	标准差	SR/%	平均值	标准差	SR/%
f_1	100	2.55e+02	2.75e+01	0	5.53e+01	4.69e+01	0	4.67e-12	8.09e-12	100	4.85e-53	2.41e-52	100
	500	2.14e+04	8.59e+02	0	1.75e+04	1.81e+03	0	2.22e-07	4.69e-07	0	8.35e-25	8.86e-25	100
	1000	7.86e+04	6.20e+03	0	4.35e+04	3.43e+03	0	6.14e-03	2.58e-03	0	1.75e-17	2.73e-17	100
f_2	100	6.57e+01	1.32e+01	0	2.01e-05	3.14e-05	0	2.46e-07	6.52e-07	20	4.01e-33	4.34e-33	100
	500	4.97e+02	1.36e+02	0	4.65e-01	5.31e-01	0	4.61e-03	5.25e-03	0	9.58e-18	2.36e-18	100
	1000	5.19e+03	1.61e+03	0	3.77e+03	2.98e+03	0	4.46e+00	4.60e+00	0	2.50e-11	1.66e-11	100
f_3	100	2.23e+00	2.34e-01	0	1.22e+01	2.94e+00	0	1.28e+00	2.37e-01	0	4.55e+01	3.69e+01	0
	500	1.05e+01	3.16e+00	0	1.73e+01	3.61e+00	0	4.87e+01	5.22e+00	0	7.22e+01	5.01e+00	0
	1000	1.73e+01	2.78e+00	0	2.19e+01	5.16e-01	0	6.97e+01	1.29e+00	0	8.31e+01	4.36e+00	0
f_4	100	2.14e+03	4.61e+02	0	1.77e+03	2.73e+02	0	9.72e+01	8.99e+00	80	6.82e+01	1.25e-01	100
	500	6.71e+06	1.05e+06	0	5.43e+06	4.69e+05	0	4.97e+02	4.47e+01	0	3.67e+02	8.23e-01	0
	1000	9.35e+07	8.60e+06	0	8.43e+06	2.18e+05	0	4.08e+03	3.61e+03	0	9.64e+02	3.74e+01	0
f_5	100	2.46e-01	2.37e-01	0	3.69e+02	2.16e+02	0	7.00e+00	6.64e-01	0	3.45e-05	2.77e-05	100
	500	3.47e+03	4.96e+02	0	2.77e+04	3.68e+03	0	8.83e+01	3.54e+00	0	2.67e-02	4.38e-03	40
	1000	5.21e+04	8.51e+02	0	6.43e+04	5.77e+03	0	2.02e+02	9.14e+00	0	8.33e+00	5.12e+00	0
f_6	100	6.02e+03	5.34e+02	0	5.34e-01	2.44e-01	0	1.40e-03	2.33e-03	40	2.34e-09	3.17e-10	100
	500	2.43e+04	3.22e+03	0	1.24e+02	5.34e+01	0	6.49e-03	3.64e-03	30	5.73e-05	4.33e-06	100
	1000	5.04e+05	4.91e+04	0	5.39e+03	6.22e+02	0	1.44e-02	5.23e-03	0	1.08e-03	1.62e-03	40
f_7	100	3.63e+02	6.38e+01	0	2.74e+01	4.93e+01	0	3.27e-13	1.81e-13	100	0.00e+00	0.00e+00	100
	500	4.39e+03	3.24e+02	0	3.93e+03	8.13e+02	0	2.62e-07	1.66e-07	0	0.00e+00	0.00e+00	100
	1000	2.77e+04	4.08e+02	0	2.33e+03	7.24e+02	0	7.89e+01	1.64e+01	0	5.57e-11	3.15e-12	100
f_8	100	2.36e+00	4.08e-01	0	1.64e-01	2.03e-01	0	1.67e-06	4.82e-07	100	2.15e-15	3.48e-15	100
	500	7.34e+00	3.71e-01	0	5.36e+00	4.71e-01	0	3.80e-04	2.83e-05	30	8.74e-12	5.39e-12	100
	1000	2.47e+01	4.23e-01	0	2.37e+00	1.53e+00	0	7.79e-02	2.03e-02	0	1.59e-07	1.63e-07	100
f_9	100	2.44e+02	9.57e+02	0	3.12e-01	4.33e+00	0	2.41e-09	3.68e-09	100	0.00e+00	0.00e+00	100
	500	1.37e+02	2.04e+02	0	2.82e+03	8.13e+01	0	1.65e-05	2.13e-06	80	1.84e-16	8.44e-17	100
	1000	2.84e+03	1.93e+02	0	1.24e+04	4.61e+02	0	2.17e-02	2.09e-03	0	2.31e-13	3.29e-14	100
f_{10}	100	8.43e-01	7.93e+00	0	5.34e+00	5.21e-01	0	1.78e-01	2.63e-02	10	2.73e-07	3.11e-07	100
	500	6.22e+03	5.77e+03	0	2.49e+01	3.42e+00	0	7.25e-01	5.18e-02	10	8.12e-05	4.98e-06	100
	1000	2.57e+06	6.34e+05	0	3.46e+01	2.18e+00	0	8.43e-01	6.14e-02	0	6.33e-04	3.77e-03	100

表3给出了PSO、GSA、GWO以及HGGWA四种算法对表1中10个高维测试函数,分别在100维、500维、1000维的情况下,30次运行结果的平均值、标准差和寻优成功率SR的对比分析。

通过分析表3可以得知,对于大部分测试函数,HGGWA算法的收敛精度和寻优成功率明显高于其他3种算法.其中在100维的情况下,除了函数 f_3 ,HGGWA算法对于其他9个测试函数的寻优成功率均达到了100%.特别地,对于函数 f_7 和 f_9 的寻优结果达到了理论最优值0.在500维的情况下,HGGWA算法对于函数 f_1 、 f_2 、 $f_6 \sim f_{10}$ 的寻优成功率达到了100%,对于函数 f_7 的寻优结果达到了理论最优值0.在1000维的情况下,HGGWA算法对函数 f_1 、 f_2 、

$f_7 \sim f_{10}$ 的寻优成功率达到100%.随着函数维度的增加,几种算法的收敛精度均有所下降,但HGGWA算法的寻优结果明显优于PSO、GSA、GWO三种算法,且在标准差方面具有优势.总体而言,在100维、500维和1000维的测试函数上,HGGWA算法表现出比PSO、GSA、GWO三种算法更好的寻优效果,验证了HGGWA算法求解高维复杂函数的有效性。

为了更清晰地比较4种算法的收敛性能,将Sphere、Schwefel's 2.22、Rastrigin、Rosenbrock、Ackley、Griewank六种函数的适应度进化曲线图进行对比.从图1可以很明显看出,不管是求解单峰函数还是多峰函数,HGGWA算法都具有更快的收敛速度和更高的收敛精度。

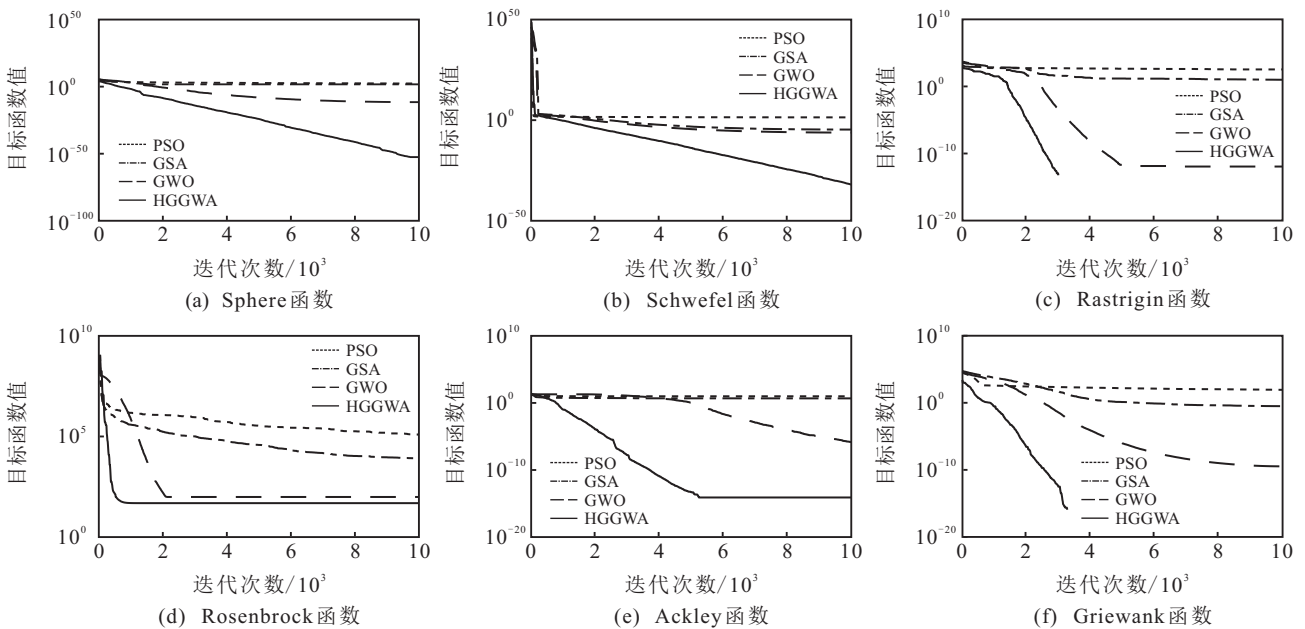


图1 PSO、GSA、GWO、HGGWA四种算法的寻优曲线对比

3.3 3种改进策略的有效性分析

为了进一步验证HGGWA算法中3种改进策略的有效性,本小节对3种只加入某一种策略的改进算法与GWO和HGGWA算法进行寻优效果对比分

析.其中,分别只加入选择、交叉、变异操作的算法分别记作HGGWA-1、HGGWA-2、HGGWA-3.10个测试函数以及算法的参数设置与上文相同,每种算法独立运行30次.具体的测试结果见表4.

表4 GWO与4种改进算法对10个测试函数(100维)的寻优结果比较

函数	GWO 算法		HGGWA-1 算法		HGGWA-2 算法		HGGWA-3 算法		HGGWA 算法	
	平均值	标准差	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	4.67e-12	8.09e-12	4.34e-41	3.16e-41	2.31e-51	2.46e-50	3.05e-41	3.16e-41	4.85e-53	2.41e-52
f_2	2.46e-07	6.52e-07	8.13e-21	5.73e-20	2.24e-31	4.31e-31	2.97e-23	3.48e-23	4.01e-33	4.34e-33
f_3	1.28e+00	2.37e-01	2.02e-06	1.78e-06	9.05e-10	5.73e-10	1.67e+01	2.37e+01	4.55e+01	3.69e+01
f_4	9.72e+01	8.99e+00	9.83e+01	5.14e+00	9.82e+01	6.34e+01	9.73e+01	8.14e+01	6.82e+01	1.25e-01
f_5	7.00e+00	6.64e-01	5.28e-01	4.43e-01	5.53e-01	3.19e-01	2.42e+01	1.62e-00	3.45e-05	2.77e-05
f_6	1.40e-03	2.33e-03	4.99e-05	3.28e-05	8.01e-05	7.56e-06	5.59e-03	3.17e-03	2.34e-09	3.17e-10
f_7	3.27e-13	1.81e-13	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
f_8	1.67e-06	4.82e-07	2.93e-14	1.35e-14	7.99e-15	1.28e-14	2.93e-14	3.74e-14	2.15e-15	3.48e-15
f_9	2.41e-09	3.68e-09	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
f_{10}	1.78e-01	2.63e-02	4.79e-02	2.34e-02	1.45e-02	3.27e-02	3.82e-02	5.63e-02	2.73e-07	3.11e-07

由表4可以得知,除了函数 f_3 , HGGWA 算法比 HGGWA-1、HGGWA-2、HGGWA-3 三种算法获得了更好的全局最优解和标准差. 特别地, 对于函数 f_7 、 f_9 , 3种改进策略对算法性能的改善作用十分明显. 具体而言, 策略1和策略2对于函数 f_1 、 f_2 、 f_3 、 f_7 、 f_8 、 f_9 的改进效果显著, 大幅度提高了算法的收敛精度; 策略3对于函数 f_1 、 f_2 、 f_7 、 f_8 、 f_9 的改进效果显著, 但是对于函数 f_3 和 f_5 却使得收敛结果变差, 这说明精英个体的变异操作对于求解函数 f_3 和 f_5 起了相

反的效果.

3.4 与其他改进算法的比较分析

为了进一步验证 HGGWA 算法的有效性, 将其与文献[23]中的9种改进算法进行寻优结果对比分析. 这9种改进算法依次为: CLPSO、OLPSO、HPSO、SaDE、JADE、jDE、GABC、CABC、OCABC. 在表1中选取单峰函数 f_1 、 f_2 、 f_6 和多峰函数 f_7 、 f_8 、 f_9 , 在30维的情况下测试算法的性能, 9种改进算法的寻优结果直接来源于文献[23]. 具体的比较结果见表5.

表5 HGGWA算法与9种改进算法的寻优结果比较

函数	HPSO	CLPSO	OLPSO	GABC	CABC	OCABC	SaDE	JADE	jDE	HGGWA
f_1	2.83e-33	1.58e-12	4.12e-54	1.92e-22	5.41e-35	4.32e-43	3.28e-20	2.69e-56	1.46e-28	9.56e-118
f_2	9.03e-20	2.51e-08	3.51e-25	3.23e-12	1.43e-18	1.17e-22	3.51e-25	3.18e-25	9.02e-24	2.07e-64
f_6	9.82e-02	5.85e-03	1.16e-02	8.94e-02	4.98e-02	4.39e-03	4.86e-03	6.14e-04	3.35e-03	7.86e-05
f_7	9.43e+00	9.09e-05	1.07e+00	1.15e-15	0.00e+00	0.00e+00	2.43e+00	1.33e-01	3.32e-04	0.00e+00
f_8	7.29e-14	3.66e-07	7.98e-15	2.15e-11	3.07e-14	5.32e-15	3.81e-06	3.35e-09	2.37e-04	7.99e-15
f_9	9.75e-03	9.02e-09	4.83e-03	2.08e-03	4.93e-04	0.00e+00	2.52e-09	1.57e-08	7.29e-06	0.00e+00

从表5可以看出,除了函数 f_8 , HGGWA 算法对其他5个测试函数均取得了最好的寻优结果. 特别是对函数 f_7 、 f_9 , HGGWA 算法的全局收敛结果达到了理论最优值0, 这体现了 HGGWA 强大的全局搜索能力, 验证了 HGGWA 算法求解高维复杂函数的优势.

4 结论

针对灰狼优化算法求解高维复杂函数易陷入局部最优的缺点, 本文对基本灰狼优化算法进行了改进, 将3种遗传算子融入到算法中, 提出了一种遗传-灰狼混合算法(HGGWA). 改进算法采用反向学习策略初始化种群, 为提高算法的搜索效率奠定了基础, 通过种群划分的方式降低了问题的规模, 针对精英个体的变异操作有效地防止了算法陷入局部最优值. 通过对10个高维测试函数的仿真模拟, 与PSO、GSA 和GWO 三种基本算法的结果相比较, 可以看出 HGGWA 算法在收敛精度方面得到了极大改进, 验证了 HGGWA 算法求解高维函数的有效性. 最后, 通过与9种改进算法的寻优结果相比较, 表明了 HGGWA 算法不管是求解单峰函数还是多峰函数, 都表现出比其他9种算法更好的全局收敛性.

参考文献(References)

[1] 翟颖妮, 孙树栋, 杨宏安, 等. 大规模作业车间多瓶颈调度算法[J]. 计算机集成制造系统, 2011, 17(7): 1486-1494.
(Zhai Y N, Sun S D, Yang H A, et al. Multi-bottleneck scheduling algorithm for large-scale job shop[J]. Computer Integrated Manufacturing Systems, 2011, 17(7): 1486-1494.)

[2] 尚文洁, 吉兴全, 郑耀东, 等. 大规模电力系统无功优化的一种分解协调算法[J]. 华北电力大学学报: 自然科学版, 2011, 38(6): 17-22.
(Shang W J, Ji X Q, Zheng Y D, et al. A decomposition and coordination algorithm for reactive power optimization in large-scale power systems[J]. Journal of North China Electric Power University: Natural Science Edition, 2011, 38(6): 17-22.)

[3] Rahnamayan S, Wang G G. Solving large scale optimization problems by opposition-based differential evolution (ODE)[J]. WSEAS Transactions on Computers, 2008, 7(10): 1792-1804.

[4] 徐东方, 郭战伟. 求解高维复合体函数的智能优化算法[J]. 数学的实践与认识, 2016, 46(19): 205-211.
(Xu D F, Guo Z W. Intelligent optimization algorithm for high-dimensional and complex functions[J]. Mathematics in Practice and Theory, 2016, 46(19): 205-211.)

[5] 拓守恒, 周涛. 基于维度分组交叉的自适应差分进化算法[J]. 计算机工程与设计, 2011, 32(9): 3174-3177.
(Tuo S H, Zhou T. Self-adaptive differential evolution algorithm based on dimensionality group cross[J]. Computer Engineering and Design, 2011, 32(9): 3174-3177.)

[6] 刘剑英. 基于GPU的并行协同差分进化算法研究[J]. 计算机工程与应用, 2012, 48(7): 48-50.
(Liu J Y. Research of parallel cooperation differential evolution algorithm based on GPU[J]. Computer Engineering and Applications, 2012, 48(7): 48-50.)

[7] Omidvar M N, Li X D, Mei Y, et al. Cooperative co-evolution with differential grouping for large scale optimization[J]. IEEE Transactions on Evolutionary Computation, 2014, 18(3): 378-393.

[8] 胡成玉, 王博. 基于动态维度交叉的粒子群高维函数优化[J]. 计算技术与自动化, 2009, 28(1): 92-95.

- (Hu C Y, Wang B. Particle swarm optimization with dynamic dimension crossover for high dimensional problems[J]. Computing Technology and Automation, 2009, 28(1): 92-95.)
- [9] 田瑾. 高维多峰函数的量子行为粒子群优化算法改进研究[J]. 控制与决策, 2016, 31(11): 1967-1972.
(Tian J. Improvement of quantum-behaved particle swarm optimization algorithm for high-dimensional and multi-modal functions[J]. Control and Decision, 2016, 31(11): 1967-1972.)
- [10] 王忠贵, 罗亚中. 高维复杂函数的混合模拟退火全局优化策略[J]. 计算机工程与应用, 2004, 40(23): 36-39.
(Wang Z G, Luo Y Z. SA-based hybrid global optimization approach for complex functions with high-dimension[J]. Computer Engineering and Applications, 2004, 40(23): 36-39.)
- [11] 于万霞, 张维存, 郑宏兴. 基于遗传粒子群算法的高维复杂函数优化方法[J]. 计算机工程与应用, 2007, 43(36): 31-33.
(Yu W X, Zhang W C, Zheng H X. Genetic and particle swarm algorithm-based optimization solution for high-dimension complex functions[J]. Computer Engineering and Applications, 2007, 43(36): 31-33.)
- [12] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69(3): 46-61.
- [13] 龙文, 伍铁斌. 协调探索和开发能力的改进灰狼优化算法[J]. 控制与决策, 2017, 32(10): 1749-1757.
(Long W, Wu T B. Improved grey wolf optimization algorithm coordinating the ability of exploration and exploitation[J]. Control and Decision, 2017, 32(10): 1749-1757.)
- [14] 龙文, 蔡绍洪, 焦建军, 等. 求解高维优化问题的混合灰狼优化算法[J]. 控制与决策, 2016, 31(11): 1991-1997.
(Long W, Cai S H, Jiao J J, et al. Hybrid grey wolf optimization algorithm for high-dimensional optimization[J]. Control and Decision, 2016, 31(11): 1991-1997.)
- [15] Rodríguez L, Castillo O, Soria J. Grey wolf optimizer with dynamic adaptation of parameters using fuzzy logic[C]. Evolutionary Computation. IEEE, 2016: 3116-3123.
- [16] Anderson-Cook R B C M. Practical genetic algorithms[J]. Journal of the American Statistical Association, 2005, 100(471): 1099.
- [17] Tizhoosh H R. Opposition-based learning: A new scheme for machine intelligence[C]. International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06). Vienna: IEEE, 2005: 695-701.
- [18] 夏学文, 刘经南, 高柯夫, 等. 具备反向学习和局部学习能力的粒子群算法[J]. 计算机学报, 2015(7): 1397-1407.
(Xia X W, Liu J N, Gao K F, et al. Particle swarm optimization algorithm with reverse-learning and local-learning behavior[J]. Chinese Journal of Computers, 2015(7): 1397-1407.)
- [19] 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演化算法[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]. Journal of Wuhan University: Natural Science Edition, 2013, 59(2): 111-116.)
- [20] Potter M A, Jong K A D. A cooperative coevolutionary approach to function optimization[C]. International Conference on Parallel Problem Solving from Nature. Berlin: Springer, 1994: 249-257.
- [21] Hao G S, Lim M H, Ong Y S, et al. Domination landscape in evolutionary algorithms and its applications[J]. Soft Computing, 2018(4): 1-8.
- [22] Engelbrecht A P. Fundamentals of computational swarm intelligence[M]. New York: John Wiley & Sons, 2006: 35-36.
- [23] Gao W F, Liu S Y, Huang L L. A novel artificial bee colony algorithm based on modified search equation and orthogonal learning[J]. IEEE Transactions on Cybernetics, 2013, 43(3): 1011-1024.

作者简介

顾清华(1981—), 男, 副教授, 博士, 从事系统工程理论与应用、运输车辆调度及优化技术、复杂系统建模及仿真等研究, E-mail: qinghuagu@xauat.edu.cn;

李学现(1993—), 男, 硕士生, 从事智能优化算法的研究, E-mail: 18366184992@163.com;

卢才武(1965—), 男, 教授, 博士生导师, 从事智能矿山、矿业系统工程等研究, E-mail: 1114238783@qq.com;

阮顺领(1981—), 男, 副教授, 博士, 从事机器学习在图像处理方向的研究, E-mail: ruanshunling@xauat.edu.cn.

(责任编辑: 齐 霖)