

具有自适应调整策略的混沌灰狼优化算法

张悦 孙惠香 魏政磊 韩博

(空军工程大学航空航天工程学院 西安 710038)

摘要 灰狼优化算法(Grey Wolf Optimization, GWO)是新型启元优化算法,相比于其他群体智能优化算法,该算法同样存在收敛速度较慢、不稳定、易陷入局部最优等问题。针对上述问题,根据GWO算法的结构特点,提出了一种自适应调整策略的混沌灰狼优化算法(Chaotic Local Search GWO),利用自适应调整策略来提高GWO算法的收敛速度,通过混沌局部搜索策略增加种群的多样性,使搜索过程避免陷入局部最优。最后利用6个测试函数对算法进行仿真验证,并结合其他4种算法进行了横向比较。实验结果证明,所提出的改进算法在收敛速度、精度以及稳定性方面具有明显的优势。

关键词 灰狼算法, 自适应, 混沌, 收敛速度, 局部最优

中图分类号 TP301 **文献标识码** A

Chaotic Gray Wolf Optimization Algorithm with Adaptive Adjustment Strategy

ZHANG Yue SUN Hui-xiang WEI Zheng-lei HAN Bo

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

Abstract Gray wolf optimization is a new optimization algorithm. Compared with other groups, it has the problems of low convergence speed, unstable and easy to fall into local optimum. According to the structural characteristics of the GWO algorithm, a chaotic gray wolf optimization algorithm based on the adaptive adjustment strategy was proposed. The adaptive adjustment strategy is used to improve the convergence rate of the GWO algorithm. The chaotic local search strategy is used to increase the diversity of the population and avoid search process falling into the local optimal. Finally, six algorithms are used to simulate the algorithm, and the other four algorithms are compared. The experimental results show that the proposed algorithm has obvious advantages in terms of convergence speed, accuracy and stability.

Keywords Gray wolf optimization, Adaptive, Chaos, Convergence rate, Local optimum

1 引言

灰狼优化算法(GWO)是由 Seyedali, Seyed 和 Andrew 于 2014 年提出的一种新型仿生群体智能算法^[1],该算法具有调置参数少、概念易理解、编程难度小,以及寻优指向性好等特点,不仅在求解约束问题^[2]、SVM 分类^[3]和模型求解等优化问题中得到了较好的应用,而且在图像分割^[4]、三维堆叠 SoC 测试规划^[5]、任务决策^[6]等领域等得到了广泛的应用。但灰狼算法也与 PSO^[7]、ACO^[8]、ABC^[9]等算法一样,存在后期收敛速度慢、易陷入局部最优、优化质量不稳定等缺点。

针对此不足,文献^[10]提出了一种自适应惯性权重的粒子群优化算法,惯性权重随着迭代次数的不断增大而非线性降低,使算法在后期通过调整权重增强局部搜索能力,最终达到最优。文献^[11]将混沌技术引入到蚁群算法(Ant Colony Optimization, ACO)中,扩大了算法后期的搜索空间,在求解 Benchmark 函数时取得了很好的结果。文献^[12]通过对人工蜂群算法(Artificial Bee Algorithm, ABC)构造自适应变异

策略提高了算法的收敛速度和精度。文献^[13]通过对布谷鸟算法(Cuckoo Search, CK)引入动态适应策略构建了 DACS,经验证其性能明显优于原始算法和决定性 CS 算法。

混沌现象具有随机性、遍历性和规律性的特点,对于给定的优化问题,通过把混沌状态引入到优化变量,利用混沌运动的特点,可得到混沌优化方法^[14]。混沌搜索算法(Chaotic Search)能够有效跳出局部最优点,但是不利于搜索算法的收敛。文献^[15]利用混沌算法具有避免陷入局部最优的能力,同时结合模式搜索算法(Pattern Search Method, PSM)^[16]较强的局部搜索能力,提出了一种混合混沌模式搜索方法。文献^[17]利用混沌序列初始化蜂群蜜源,通过从大量初始蜜源中选择较优蜜源来提高蜂群算法的收敛速度。

本文从上述文献中得到启发,提出了一种自适应调整策略的混沌灰狼优化算法。首先通过结合适应度构造了自适应控制参数调整策略和位置更新策略,提高算法的更新速度;另一方面,将混沌思想引入到算法后期搜索中,当算法陷入到最优解时,重新构造一个较小的搜索空间,并进一步评估每一个解,然后继续进行寻优,最终获得最优解。

本文受国家自然科学基金资助项目(51208506)资助。

张悦(1992—),女,硕士生,主要研究方向为智能优化算法, E-mail: 610534845@qq.com; 孙惠香(1975—),女,博士,副教授,主要研究方向为智能优化算法在结构工程中的应用, E-mail: sunhx7504@sina.com; 魏政磊(1991—),男,博士,主要研究方向为智能优化算法, E-mail: happyhanbo@163.com; 韩博(1992—),男,博士,主要研究方向为智能优化算法, E-mail: 295629469@qq.com。

2 灰狼优化算法的原理

灰狼优化算法实质上是一个模拟生物特性的算法。狼是一种群体性生物,一般以狼群为单位进行狩猎,而其内部等级结构非常森严。狼群的首领是 α ,它的责任是对狼群进行管理,组织其他狼进行捕食和决定栖息地。处于狼群第2级的灰狼 β 的责任是辅助首领 α 进行决策;第3级是 δ ,它的责任是进行目标侦查、警戒护卫和打围;最低级的是灰狼 ω ,它的责任是服从其他高层级灰狼的领导。

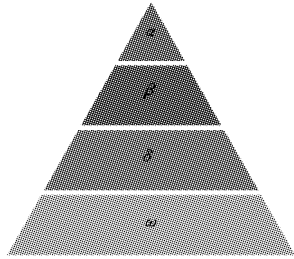


图1 灰狼社会等级制度

狼群进行狩猎时的步骤为接近、包围、攻击,灰狼算法模仿这个狩猎的过程建立了一个数学模型。狼群中每一个个体都是一个潜在的解,而位于狼群最高的3个层级的灰狼(α, β, δ)从高至低也是位置最好的3个解,猎物的位置由这3个解进行判定,其余的灰狼 ω 以所判定的位置为依据,计算与猎物之间的距离,实现对猎物全方位的靠近、包围、攻击等。

猎物的位置由 α, β, δ 3个位置决定,狼群追踪猎物的示意图如图2所示。

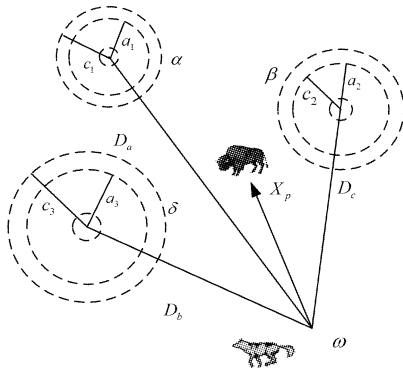


图2 灰狼算法示意图

其中灰狼个体与猎物的距离为:

$$D = |CX_p(t) - X(t)| \quad (1)$$

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

灰狼位置更新为:

$$X(t+1) = X_p(t) - AD \quad (3)$$

$$A = 2a r_2 - a \quad (4)$$

$$a = 2 - 2(t/\max_iter), a \in [0, 2] \quad (5)$$

其中, t 为迭代的次数, $X_p(t)$ 代表第 t 代时猎物的位置, $X(t)$ 表示第 t 代时个体灰狼的位置, D 为猎物与灰狼之间的距离。 A 为收敛因子,当 $|A| > 1$ 时进行全局搜索,当 $|A| < 1$ 时进行局部搜索。 a 的取值为 $[0, 2]$, r_1 和 r_2 为 $[0, 1]$ 内的随机数。

狼群跟踪猎物的数学模型表达如下。

$$\begin{cases} D_\alpha = |C_1 X_\alpha(t) - X(t)| \\ D_\beta = |C_1 X_\beta(t) - X(t)| \\ D_\delta = |C_1 X_\delta(t) - X(t)| \end{cases} \quad (6)$$

$$\begin{cases} X_1 = X_\alpha - A_1 D_\alpha \\ X_2 = X_\beta - A_2 D_\beta \\ X_3 = X_\delta - A_3 D_\delta \end{cases} \quad (7)$$

式(6)和式(7)为狼群内个体 ω 与 α, β, δ 的距离关系。

$$X_p(t+1) = \frac{X_1 + X_2 + X_3}{3} \quad (8)$$

式(8)根据 α, β, δ 的位置判断出灰狼个体 ω 与猎物的位置关系。

从式(4)中可以得知,通过控制参数 a ,可实现全局与局部最优搜索的切换。当 a 较大时,搜索步长较大,有利于扩大搜索范围,利于全局寻优,而随着迭代次数的不断增加, a 呈现一种线性递减的趋势,当 a 较小时,个体步长减小,搜索区域变小,有利于算法全局最优值的确定。

3 自适应调整策略的灰狼优化算法

通过对GWO算法进行分析可知,灰狼算法是通过最优解、次优解以及次优解定位猎物的位置。这种方法一方面使算法具有更明确的指向性,在迭代初期展现出快速的收敛速度,但另一方面也导致了在优化后期,种群数量过少,优化效果不稳定,易陷入局部最优。同时由于算法的控制参数与位置更新策略都是以线性方式进行变化的,没有考虑迭代过程中种群实时的变化情况,在一定程度上减缓了算法的收敛速度。针对上述问题,本文通过结合适应度值,提出了自适应调整策略,加快了算法的收敛速度,同时引入了混沌局部搜索策略来增加种群的多样性,使搜索过程避免陷入局部最优。

3.1 自适应调整策略

灰狼算法中适应度值是反映个体与全局最优值之间的重要参数。在灰狼进行寻优的过程中, α, β, δ 的位置对于寻找最优解至关重要,但在进行更新位置时,没有体现出 α, β, δ 之间的真实位置的相对关系,只是进行了简单的求平均值,导致扩大了搜索的范围,减缓了收敛速度。

为了进一步提高GWO算法的收敛速度,同时也为了平衡GWO算法的全局搜索能力和局部改良能力,控制参数值 a 采用自适应调整策略。在该策略中,将当前个体的适应度值 J_i 与灰狼群的平均适应度值 J_{avg} 进行比较,如果 J_i 优于平均适应度值 J_{avg} ,其相应灰狼个体的控制参数 a 较小,便于使其向最好的位置靠拢;如果 J_i 次于 J_{avg} ,那么其相应灰狼个体的控制参数值 a 会较大,使其到更广阔的区域搜索,向较好的搜索区域靠拢。

本文采用的控制参数自适应调整策略表达式如下:

$$a_i = \begin{cases} a_{\min} + \frac{(a_{\max} - a_{\min}) \cdot (J_i - J_{\min})}{J_{avg} - J_{\min}}, & J_i \leq J_{avg} \\ a_{\max}, & J_i > J_{avg} \end{cases} \quad (9)$$

其中, $a_{\min}, a_{\max}$ 分别为控制参数的最小值和最大值, a_i 为灰狼个体当前的控制参数值, J_i 表示灰狼个体当前的适应度值, J_{avg} 和 $J_{\min}$ 分别表示当前灰狼群体的平均适应度值和最小适应度值。在上式中,控制参数值会随着灰狼个体的适应度值而自动改变。

本文同时提出了一种基于适应度值的位置更新策略,通过 α, β, δ 相互之间的适应值比值对3个灰狼的位置赋予权重,然后再对猎取的位置进行求解。其表达式为:

$$x(t+1) = \frac{(J_\alpha/J_\delta)x_1 + (J_\beta/J_\delta)x_2 + x_3}{3} \quad (10)$$

其中, J_α 为灰狼 α 的适应度值, J_β 为灰狼 β 的适应度值, J_δ 为灰狼 δ 的适应度值。

3.2 混沌局部搜索策略

混沌优化算法是根据混沌现象的随机性、遍历性和规律性,通过把混沌状态引入到优化变量,利用混沌运动的遍历性使算法具有避免陷入局部最优的能力,因此可以利用混沌算法改进灰狼优化算法,以提高其收敛质量。Logistic 映射就是一个典型混沌系统。

$$x_{n+1} = \mu x_n (1 - x_n) \quad (11)$$

其中, μ 为混沌控制参数, μ 的取值直接影响着 Logistic 方程的混沌程度,当 μ 值越大,混沌程度越高, μ 一般在 $[0, 4]$ 之间取值, $x_n \in [0, 1]$ 。

初值对混沌的影响非常大,因此给式(11)赋 i 个微小差异的初值,得到 i 个混沌变量。

混沌灰狼优化算法要涉及到混沌局部搜索算法 (Chaotic Local Search, CLS), 因此 CLS 算法步骤如下。

Step1 当 $k=0$ 时,将决策变量 x_j^k 按式(12)映射为 0 到 1 之间的混沌变量 s_j^k :

$$s_j^k = \frac{x_j^k - x_{\min,j}}{x_j^k - x_{\max,j}} \quad (12)$$

其中, $x_{\min,j}$ 和 $x_{\max,j}$ 分别为决策变量 X 的第 j 维变量的搜索上下界。

Step2 根据式(13),计算混沌局部搜索算法内部下一步迭代的混沌变量 s_j^{k+1} :

$$s_j^{k+1} = 4s_j^k (1 - s_j^k) \quad (13)$$

Step3 根据式(14),将混沌变量 s_j^{k+1} 转换为决策变量 x_j^{k+1} :

$$x_j^{k+1} = x_{\min,j} + s_j^{k+1} (x_{\max,j} - x_{\min,j}) \quad (14)$$

Step4 根据决策变量 x_j^{k+1} ,对新解进行评价,若新解优于初始解 $X^{(0)} = [x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}]$ 或者混沌搜索已经达到最大迭代步数,则将新解作为 CLS 的搜索结果,否则转到 Step2 继续搜索。

3.3 混沌灰狼优化算法流程

利用混沌搜索算法具有陷入避免局部极小的能力,结合改进 GWO 算法的快速收敛能力,提出了一种混沌灰狼优化算法。在该算法中,改进 GWO 算法主要用于全局搜索,而 CLS 算法根据 GWO 算法结果进行局部搜索,而且为了保持种群的多样性,加强搜索的分散性,在保留一半数量优秀灰狼个体的同时,根据种群最好位置动态收缩搜索区域,并在收缩搜索区域内随机产生灰狼个体来代替性能较差的灰狼个体。

混沌灰狼优化算法的流程如下:

Step1 初始化灰狼种群,即随机产生 n 个智能个体的位置;初始化 a , A 和 C ;初始化 X_α , X_β 和 X_δ 的值。

Step2 处理越界智能个体,计算每个智能个体的适应度值,同时计算灰狼群的平均适应度值 J_{avg} 和最小适应度值 J_{min} 。

Step3 比较智能个体的适应度值 X_α 和 X_δ ,确定当前迭代期间的最优解 X_α , 优解 X_β 和次优解 X_δ 。

Step4 对于每个智能个体,根据控制参数自适应地调整策略,比较当前智能个体适应度值 J_i 与 J_{avg} ,通过式(9)计算控制参数 a 的值,然后计算当前智能个体相对于各个优解的 A 和 C 。

Step5 根据式(6)、式(7)和式(10)更新当前智能个体的位置。

Step6 计算更新位置后的每个智能个体的适应度值,然后选取灰狼群一半较好的灰狼个体(智能个体)。

Step7 对群体中较好的智能个体进行混沌局部搜索,并更新最优解 X_α 、优解 X_β 和次优解 X_δ 。

Step8 如果达到结束条件(最大迭代次数或者预定的运算精度),则结束,输出最优解,否则转到 Step2。

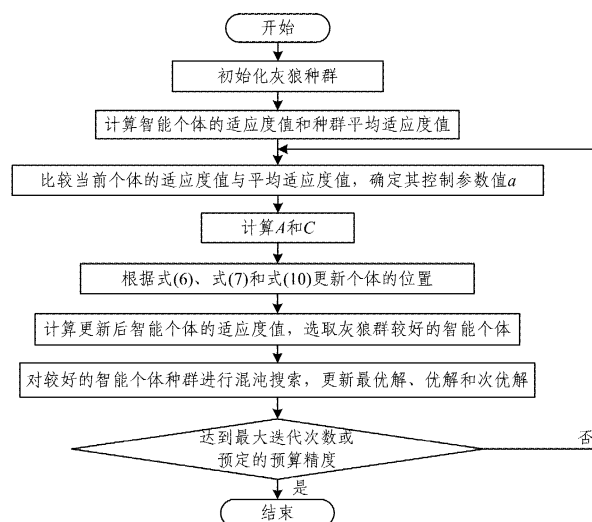


图 3 混沌灰狼优化算法的流程图

4 实验仿真与结果分析

4.1 实验设置

为了验证本文提出的 CLS-GWO 算法的优化性能,实验选取了文献[18]中 6 种经典测试函数进行验证,与标准 GWO、ACO 算法、PSO 算法、ABC 算法比较。表 1 列出了 6 种基准函数的基本描述。 F_1 , F_2 和 F_3 是单峰标准测试函数, F_4 , F_5 和 F_6 是多峰标准测试函数。

表 1 6 种典型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	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	30	$[-5.12, 5.12]$	0
F_5	Ackley	30	$[-32, 32]$	0
F_6	Griewank	30	$[-600, 600]$	0

在仿真验证中,灰狼种群 $N=30$,初始值 $a=2$,迭代次数 $T=500$,混沌控制参数 $\mu=4$,各算法独立循环 50 次。PSO 算法设置参照文献[7],ABC 参数设置参照文献[8],ACO 参数设置参照文献[9],当算法达到最大迭代次数或者优化结果连续 6 次结果相差为 $1.00E-3$ 时停止循环,输出结果。

4.2 实验结果与分析

实验采用 Matlab R2013a 进行仿真,运行环境为 Intel (R) Core(TM) i5-3470 处理器, 3.53GB 内存。为了达到准确的实验结果,针对 6 种测试函数计算它们独立运行 50 次的统计平均值和方差值。

表 2 和表 3 列出了 5 种算法在 6 种测试函数上独立运行 50 次的实验结果。从测试结果中可以得到,相比于另外 4 种

算法,CLS-GWO算法在 F_1 — F_6 测试函数中的收敛精度和稳定性都有显著的优势。

表 2 5 种算法之间的平均值对比

算法	F_1	F_2	F_3	F_4	F_5	F_6
	Mean	Mean	Mean	Mean	Mean	Mean
ACO	2.3145e-02	6.1248e-08	6.1478e-01	8.1456e+01	5.1457e-04	1.8749e-02
PSO	1.4875e+00	6.1458e+00	5.4785e+00	2.4574e+02	3.1452e+00	1.4785e-01
ABC	1.1458e-16	5.1426e-07	2.1345e-02	3.1452e+01	5.1245e-03	3.1547e+01
GWO	2.1458e-27	8.1245e-17	1.5874e-03	3.1425e+00	1.5464e-13	3.2654e-03
CLS-GWO	4.2564e-37	1.2365e-22	6.1425e-04	0.0000e+00	8.1475e-15	0.0000e+00

表 3 5 种算法之间的方差对比

算法	F_1	F_2	F_3	F_4	F_5	F_6
	Std	Std	Std	Std	Std	Std
ACO	3.1425e-06	6.1487e-03	4.1254e+00	7.1245e+04	6.1234e-07	2.1456e-01
PSO	1.1245e+00	2.1456e+00	3.2144e+01	2.1245e+03	3.3654e-01	3.2659e-03
ABC	5.1246e-33	3.1456e-15	3.3641e-03	3.1245e+01	9.1254e-04	3.2145e+01
GWO	3.6548e-54	5.1245e-33	1.2548e-07	1.2548e+01	8.1236e-28	4.5687e-05
CLS-GWO	8.4576e-72	6.1457e-44	9.1256e-09	0.0000e+00	2.3654e-28	0.0000e+00

为了更加直观地体现 CLS-GWO 算法的优越性能,图 4—图 9 仿真了 5 种优化算法的收敛曲线。从图 4—图 9 中可以看出,相比于 GWO,ACO,PSO 和 ABC 算法,本文提出的自适应 CLS-GWO 算法表现出了更快的收敛速度、更高的收敛精度。通过实验仿真验证,自适应 CLS-GWO 算法的性能优于另外 4 种算法。

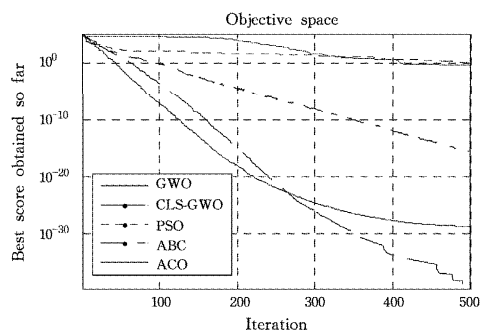


图 4 F_1 测试函数

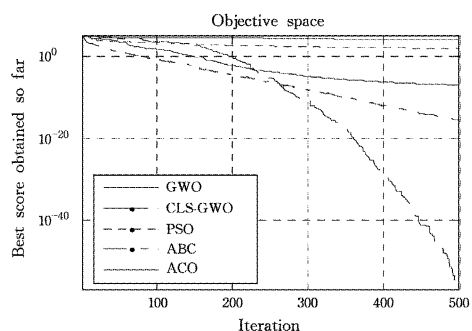


图 5 F_2 测试函数

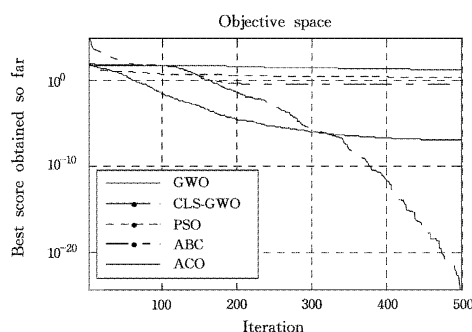


图 6 F_3 测试函数

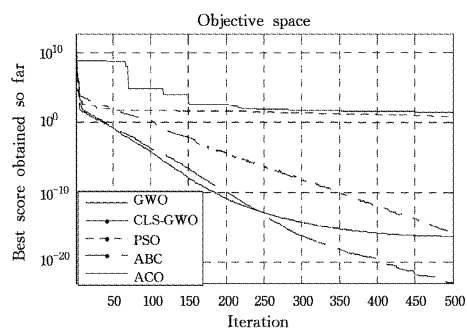


图 7 F_4 测试函数

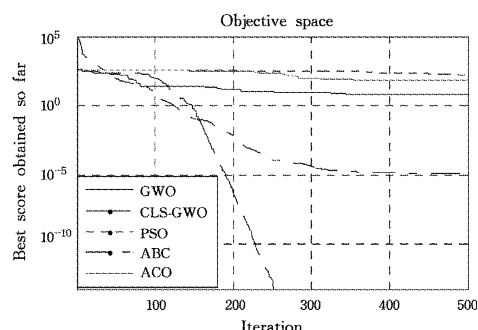


图 8 F_5 测试函数

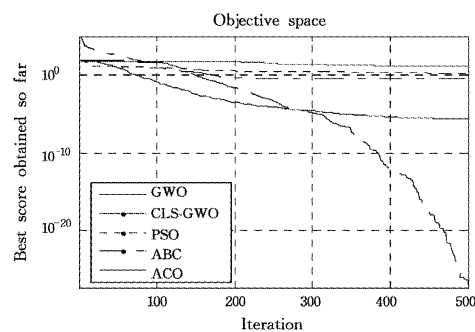


图 9 F_6 测试函数

结束语 本文针对 GWO 算法收敛速度慢、易陷入局部最优等缺点,结合适应度值构建了自适应调整策略,使位置控制参数和位置更新策略能根据算法中灰狼个体的相对位置关系进行调整,加快了算法的收敛速度;同时将混沌搜索策略引

(下转第 159 页)

- Speed Tracking with Kernelized Correlation Filters[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(3): 583-596.
- [12] KALMAN R E. A New Approach to Linear Filtering and Prediction Problems[J]. Journal of Basic Engineering, 1960, 82(1): 35-45.
- [13] REN S, HE K, GIRSHICK R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[C]// Advances in Neural Information Processing Systems, 2015: 91-99.
- [14] YANG F, CHOI W, LIN Y. Exploit All the Layers: Fast and Accurate CNN Object Detector with Scale Dependent Pooling and Cascaded Rejection Classifiers[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 2129-2137.
- [15] RIFKIN R, YEO G, POGGIO T. Regularized Least-Squares Classification[J]. Nato Science Series Sub Series III Computer and Systems Sciences, 2003, 190: 131-154.
- [16] YU F, LI W, LI Q, et al. POI: Multiple Object Tracking with High Performance Detection and Appearance Feature[C]// European Conference on Computer Vision. Springer International Publishing, 2016: 36-42.
- [17] LEAL-TAIXÉ L, MILAN A, REID I, et al. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking[J/OL]. <https://arXiv.org/abs/1504.01942>.
- [18] GEIGER A, LAUER M, WOJEK C, et al. 3D Traffic Scene Understanding from Movable Platforms[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(5): 1012-1025.
- [19] PIRSAVASH H, RAMANAN D, FLOWKES C C. Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects[C]// 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2011: 1201-1208.
- [20] DCILE C, CAMPS O I, SZNAIER M. The Way They Move: Tracking Multiple Targets with Similar Appearance[C]// Proceedings of the IEEE International Conference on Computer Vision, 2013: 2304-2311.
- [21] MILAN A, ROTH S, SCHINDLER K. Continuous Energy Minimization for Multitarget Tracking[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(1): 58-72.
- [22] YOON J H, YANG M H, LIM J, et al. Bayesian Multi-object Tracking using Motion Context from Multiple Objects[C]// 2015 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2015: 33-40.

(上接第 122 页)

人到算法中,通过利用混沌现象的特点在局部进行搜索,一方面增加了种群的多样性,另一方面扩大了搜索区域,避免了灰狼个体因种群多样性的不足而陷入局部最优。通过 6 种测试函数的统计结果,CLS-GWO 算法在收敛速度和精度方面得到了明显的提高,并解决了算法在多峰值函数中陷入局部最优的问题。如何在航迹规划中利用 CLS-GWO 的特点进行应用是下一步研究的内容。

参 考 文 献

- [1] SEYEDAIL M, SEYED M M, LEWIS A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69: 46-61.
- [2] 罗佳,唐斌. 新型灰狼优化算法在函数优化中的应用[J]. 兰州理工大学学报, 2016, 42(3): 96-101.
- [3] 徐达宇,丁帅. 改进 GWO 优化 SVM 的云计算资源负载短期预测研究[J]. 计算机工程与应用, 2017, 53(7): 68-93.
- [4] 张新明,涂强,康强,等. 双模狩猎的灰狼优化算法在多阈值图像分割中应用[J]. 山西大学学报(自然科学版), 2016, 39(3): 378-385.
- [5] ZHU A J, XU C P. Hybridizing grey wolf optimization with differential evolution for global optimization and test scheduling for 3D stacked SoC[J]. Journal of Systems Engineering and Electronics, 2015, 26(2): 317-328.
- [6] 魏政磊,赵辉,韩邦杰,等. 基于自适应 GWO 的多 UCAV 协同攻击目标决策[J]. 计算机工程与应用, 2016, 52(18): 257-261.
- [7] KENNEDY J, EBERHART R. Particle swarm optimization[C]// IEEE Int Conf on Neural Networks. Piscataway: IEEE, 1995: 1942-1948.
- [8] DORIGO M, STUTZLE T. Ant colony optimization[M]. Cambridge, MA: MIT Press, 2004.
- [9] KARABOGA D. An idea based on bee swarm for numerical optimization[R]. Turkey: Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- [10] JIAO B, LIAN Z G, GU X S. A dynamic inertia weight particle swarm optimization algorithm[J]. Chaos, Solitons & Fractals, 2008, 37(3): 698-705.
- [11] QIAN S, CHEN D. Discrete Gabor transform[J]. IEEE Transactions on Signal Processing, 1993, 41(7): 2429-2438.
- [12] 何鹏,阎兴顺,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.
- [13] 张永韡,江韬,吴启迪. 动态布谷鸟搜索算法[J]. 控制与决策, 2014, 29(4): 617-622.
- [14] 李兵,蒋慰孙. 混沌优化方法及其应用[J]. 控制理论与应用, 1997, 14(4): 613-615.
- [15] 柳贺,黄猛,柳桂国,等. 基于混沌搜索和模式搜索的混合优化方法[J]. 华东理工大学学报(自然科学版), 2008, 34(1): 126-130.
- [16] 薛毅. 最优化原理与方法[M]. 北京:北京工业大学出版社, 2001.
- [17] 曹璐,贾银平,张安. 基于改进人工蜂群算法的多无人机协同航迹规划[J]. 计算机应用, 2013, 33(12): 3596-3599.
- [18] DIGALAKIS J, SMUTNICKI C. On benchmarking functions for genetic algorithm[J]. International Journal of Computer Mathematic, 2001, 77: 841-506.