

其中, A 为收敛影响因子, a 为收敛因子, 随迭代次数的增加线性递减, r_2 为 $[0,1]$ 区间均匀分布的随机数, Max_iter 为最大迭代次数.

定义 3. 灰狼群体位置更新. 在灰狼群体中, α, β 和 δ 三头狼为距离猎物相对最近的狼, 假设这三头狼对猎物的位置具有更多的信息, 因此在捕食过程中由这三头狼带领狼群其他狼一起向猎物靠近, 其他狼每一步移动的位置由这三头狼共同决策. α, β 和 δ 需通过式 (6)–(8) 分别计算各自与猎物的位置, 然后通过式 (9)–(11) 更新各自下一步的位置, 则灰狼群体其他狼下一步移动的位置由式 (12) 决定.

$$D_\alpha = |C_1 \cdot X_\alpha - X| \quad (6)$$

$$D_\beta = |C_2 \cdot X_\beta - X| \quad (7)$$

$$D_\delta = |C_3 \cdot X_\delta - X| \quad (8)$$

$$X_1 = X_\alpha - A \cdot D_\alpha \quad (9)$$

$$X_2 = X_\beta - A \cdot D_\beta \quad (10)$$

$$X_3 = X_\delta - A \cdot D_\delta \quad (11)$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \quad (12)$$

2 FWGWO 算法

2.1 改进的收敛因子

对于启发式算法, 寻优的过程都涵盖了广度搜索和深度搜索两类操作. 一般, 前期缺少先验知识, 种群会在全局范围内进行搜索, 此时主要为广度搜索, 目的是尽可能探索更广度的区域, 发现较多的全局最优解; 深度搜索则主要是在算法后期, 在广度搜索基础上进行深度开发, 此阶段种群会根据前期搜索的先验知识, 逐渐靠近较优解, 尽可能靠近最优解以及获得更好的求解精度. 一个优秀的寻优算法不仅要求算法有较快的收敛速度, 还要求在前期有较强的探索能力, 在后期有较强的开发能力, 以获得更高的求解精度. 因此, 提高算法的寻优性能务必要协调好广度搜索和深度搜索的比例关系.

在 GWO 算法中, 变量 A 是主要影响到广度搜索和深度搜索的变量. 在 GWO 算法中定义了, 当 $|A| > 1$ 时, 灰狼群体将扩大包围圈, 此时, 算法为广度搜索, 当 $|A| < 1$ 时, 灰狼群体将缩小包围圈, 以完成对猎物的包围攻击, 此时, 算法为深度搜索, 如图 2 所示^[8].

由式 (4) 可知, A 的变化区间主要由收敛因子 a 决定, 当 a 从 2 递减为 0, A 的取值在 $[-a, a]$ 之间变化. 由此可知, A 的取值变化主要受收敛因子 a 影响, 因此, 通过调整收敛因子 a , 可以调整算法寻优过程中广度和深度搜索的比例.

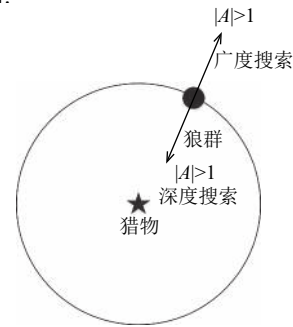


图 2 变量 A 与种群搜索的关系

在原 GWO 算法中, 广度搜索和深度搜索比例为 1:1. 为加快 GWO 算法的收敛速度, 避免原 GWO 算法容易陷入局部最优的问题, 本文提出了一种新的非线性收敛因子, 增加了前期广度搜索的比例, 一方面利于搜索更多的全局最优点, 避免陷入局部最优, 另一方面该收敛因子在算法后期的取值变化迅速, 能使得算法在后期获得更快的收敛速度, 从而获得更好的求解精度. 改进后的收敛因子由式 (13) 表示.

$$a = 2 \sqrt{1 - \left(\frac{t}{Max_iter} \right)^2} \quad (13)$$

2.2 基于权重决策的位置更新策略

在 GWO 算法中, 由式 (8) 可知, 种群下一次移动的位置由 α, β, δ 共同做决策, 且决策平均. 显然, 这种平均决策方法并没有考虑 α, β, δ 的个体特征, 没办法体现出 α 狼作为主要领导者和适应度最高的个体在决策时的重要性.

为实现自适应调整决策个体在不同时期的权重比例, 本文提出了一种基于模糊控制的权重决策种群位置更新策略, 在算法迭代的不同时期, 通过模糊控制器对 α, β 和 δ 赋予不同的权重比例, 使得种群更新的位置更加可靠, 改进的种群更新位置公式如式 (14) 所示.

$$X(t+1) = \frac{fw_\alpha}{fw_\alpha + fw_\beta + fw_\delta} X_1 + \frac{fw_\beta}{fw_\alpha + fw_\beta + fw_\delta} X_2 + \frac{fw_\delta}{fw_\alpha + fw_\beta + fw_\delta} X_3 \quad (14)$$

其中, f_{w_α} 、 f_{w_β} 、 f_{w_δ} 分别为 α 、 β 、 δ 的模糊权重系数, 由模糊控制器获得.

基本的模糊控制器一般包括输入量模糊化, 模糊逻辑和模糊判决, 其结构如图3表示^[22].

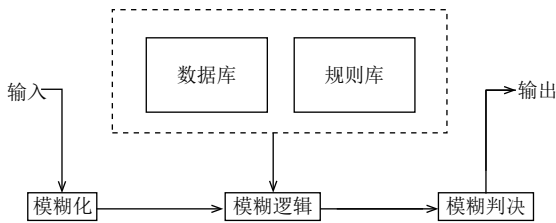


图3 基本模糊控制器

本文定义的模糊控制器包括一个输入和三个输出, 其中迭代次数作为输入, 模糊集包括“前”, “中前”, “中”, “中后”, “后”; α 、 β 、 δ 的模糊系数作为输出, 模糊集均用“低”, “较低”, “中”, “较高”, “高”来描述. 算法迭代次数的隶属度函数, 如图4所示, α 、 β 、 δ 的模糊系数的隶属度函数相同, 如图5所示.

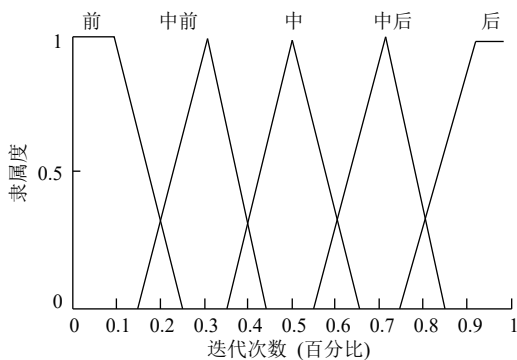


图4 模糊控制器输入—迭代次数的隶属度函数

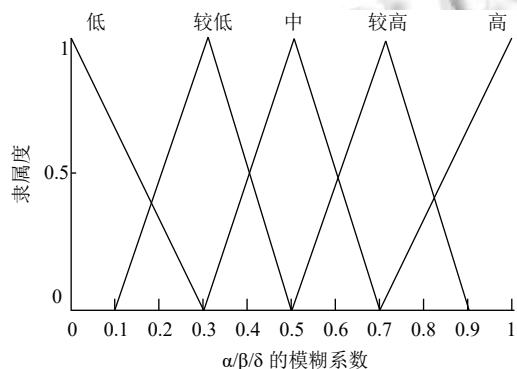


图5 模糊控制器输出— $f_{w_\alpha}/f_{w_\beta}/f_{w_\delta}$ 的隶属度函数

定义如下 IF-THEN 模糊规则:

规则 1. IF<迭代次数为“前”>, THEN< α 的模糊系

数 f_{w_α} 为“中”, β 的模糊系数 f_{w_β} 为“中”, δ 的模糊系数 f_{w_δ} 为“较低”>;

规则 2. IF<迭代次数为“中前”>, THEN< α 的模糊系数 f_{w_α} 为“较高”, β 的模糊系数 f_{w_β} 为“中”, δ 的模糊系数 f_{w_δ} 为“较低”>;

规则 3. IF<迭代次数为“中”>, THEN< α 的模糊系数 f_{w_α} 为“较高”, β 的模糊系数 f_{w_β} 为“中”, δ 的模糊系数 f_{w_δ} 为“低”>;

规则 4. IF<迭代次数为“中后”>, THEN< α 的模糊系数 f_{w_α} 为“较高”, β 的模糊系数 f_{w_β} 为“较低”, δ 的模糊系数 f_{w_δ} 为“低”>;

规则 5. IF<如果迭代次数为“后”>, THEN< α 的模糊系数 f_{w_α} 为“高”, β 的模糊系数 f_{w_β} 为“较低”, δ 的模糊系数 f_{w_δ} 为“低”>.

最后, 使用 Matlab 的模糊控制器工具箱完成模糊控制器的设计, 解模糊采取“重心法”.

2.3 FWGWO 算法描述

综上所述, FWGWO 的算法描述如下:

算法 1. FWGWO 算法

- 1) 初始化种群规模 N , 随机产生初始种群, 初始化 $t=0$, 初始化 a , A 和 C 等参数;
- 2) 计算种群中每个个体的适应度值, 将适应度排名前三的个体分别记为 X_α , X_β 和 X_δ ;
- 3) 由式 (6)–(8) 计算种群中其他个体与 X_α , X_β 和 X_δ 的距离, 并根据式 (9)–(11) 更新个体位置.
- 4) 由模糊控制器获得决策层 X_α , X_β 和 X_δ 的权重系数, 由式 (14) 完成目标定位.
- 5) 更新算法中 a , A 和 C 等参数
- 6) 判定算法是否满足收敛条件. 如果满足, 则算法结束; 否则, 令 $t=t+1$, 返回第 3) 步.

3 实验及分析

本文采用 Matlab R2012b 进行仿真, 运行环境为 Intel(R) Core(TM) i7-3770 处理器, 3.5 GHz 内存. 测试集为 CEC2005, 将 GWO 算法、粒子群算法 (Particle Swarm Optimization, PSO) 及 FWGWO 算法对测试集中的 23 个标准测试函数进行测试, 其中 F1–F7 为单峰函数, F8–F13 为多峰函数, F14–F23 为固定维数多峰函数, 篇幅原因, 测试函数的定义、维度及最小值等参数详见文献[1].

仿真实验设置种群数目为 30, 迭代次数为 500, 测

试函数的维度按文献[1]中设置. 几种算法的参数设置如下: PSO 算法中设置 $\omega_{\max}=0.9$, $\omega_{\min}=0.2$, $c_1=c_2=2$; GWO 算法和 FWGWO 算法设置 $a_{\max}=2$, $a_{\min}=0$. 针对同一测试函数, 将每种算法独立运行 50 次, 并统计均值和标准差.

表 1 给出了三种算法针对 13 个标准测试函数独立运行 50 次的均值和标准差, 其中 F1—F7 为单峰标准函数, F8—F13 为多峰标准函数.

表 2 给出了三种算法针对 10 个固定维数多峰函数独立运行 50 次的均值和标准差.

表 1 三种算法针对 13 个标准函数的测试结果

测试函数	PSO	GWO	FWGWO
	均值/标准差	均值/标准差	均值/标准差
F1	0.000 14/0.000 21	5.95E-28/6.85E-28	2.29E-45/8.48E-45
F2	0.042 14/0.045 42	7.95E-17/4.97E-17	4.09E-27/2.63E-27
F3	70.1324/22.1192	2.83E-05/0.000 11	1.36E-09/1.30E-08
F4	1.086 48/0.317 04	5.69E-07/5.55E-07	3.61E-12/3.22E-12
F5	96.718 32/60.1155	27.034 09/0.826 68	26.942 64/0.700 48
F6	0.000 11/8.28E-05	0.764 43/0.358 24	0.610 14/0.261 94
F7	0.122 85/0.044 96	0.001 72/0.000 75	0.001 18/0.000 72
F8	-4841.280 42/1152.841 21	-6084.025 25/1022.676 27	-5580.744 21/772.697 36
F9	46.704 23/11.629 38	3.229 59/4.163 19	0.165 75/0.907 86
F10	0.276 01/0.509 01	1.05E-13/2.39E-14	7.82E-15/7.89E-16
F11	0.009 21/0.007 74	0.005 14/0.009 98	7.16E-04/0.003 23
F12	0.006 92/0.011 91	0.059 94/0.097 85	0.041 66/0.019 52
F13	0.006 68/0.008 91	0.627 15/0.306 81	0.533 53/0.233 30

表 2 三种算法针对固定维数多峰测试函数的测试结果

测试函数	PSO	GWO	FWGWO
	均值/标准差	均值/标准差	均值/标准差
F14	3.627 17/2.560 81	5.086 61/4.347 26	4.092 58/4.080 47
F15	0.000 58/0.000 22	0.005 73/0.008 98	0.001 57/0.000 47
F16	-1.0316/6.25E-16	-1.0316/2.42E-08	-1.0316/0
F17	0.397 89/0	0.397 89/0	0.397 89/0
F18	3.182 40/1.33E-15	5.700 03/14.788 50	3.000 01/1.41E-05
F19	-3.862 78/2.58E-15	-3.861 68/0.002 17	-3.861 86/0.001 85
F20	-3.261 34/0.060 52	-3.230 36/0.084 33	-3.26321/0.06597
F21	-6.8651/3.019 64	-8.799 56/2.279 61	-9.466 88/1.749 44
F22	-8.456 53/3.087 09	-10.223 84/0.970 07	-10.311 94/0.967 92
F23	-8.952 91/1.782 75	-9.9050/1.964 50	-10.526 48/0.006 55

由表 2 可以看出, FWGWO 算法对于大部分函数均取得较好的寻优结果, 特别是对于单峰标准函数 F1—F4, FWGWO 算法性能明显优于对比算法. FWGWO 和 GWO 算法在 F1、F2、F3、F4、F10 上可判定为找到全局最优解, 而 FWGWO 算法相对于基本 GWO 算法具有更好的求解精度. 对于 F9、F11, 在 100 次独立测试中, FWGWO 算法寻优正确率分别是 91%、93%, 而 GWO 算法寻优正确率分别为 51%、72%. 由此可知, FWGWO 算法相较于 GWO 算法, 在大部分函数的寻优性能上有明显提升. PSO 算法在测试函数 F6、F12、F13 取得较好的寻优结果, 但整体看来, FWGWO 算法在大部分函数的寻优结果均优于 PSO 算法.

由表 3 可以看出, FWGWO 算法在测试函数 F16—F23 都能收敛到函数最小值, 对于测试函数 F14、F15 基本接近函数最小值. 相比于对比算法, FWGWO 算法在大部分函数具有更好的求解精度. 图 6 示意了三种算法针对不同测试函数的收敛曲线, 展示了算法针对两个单峰测试函数 F1、F3 和两个多峰测试函数 F9、F10 的收敛曲线.

由图 6 可以看出, FWGWO 算法相较于对比算法, 算法后期具有更快的收敛速度, 且能取得更好的寻优结果. 综合以上分析可知, 相较于 PSO 算法和 GWO 算法, FWGWO 算法具有更好的算法寻优性能和算法稳定性.

4 结论与展望

本文提出一种改进的灰狼优化算法,提出了一种新的收敛因子以及一种基于动态权重的种群位置更新策略,通过模糊控制器获得决策个体的权重系数,使决

策时更好地体现决策个体的差异性,从而使种群更新位置更准确.算法在23个标准测试函数上的实验结果表明,本文提出的模糊权重灰狼优化算法FWGWO与对比算法相比,具有更好的寻优性能和更好的稳定性.

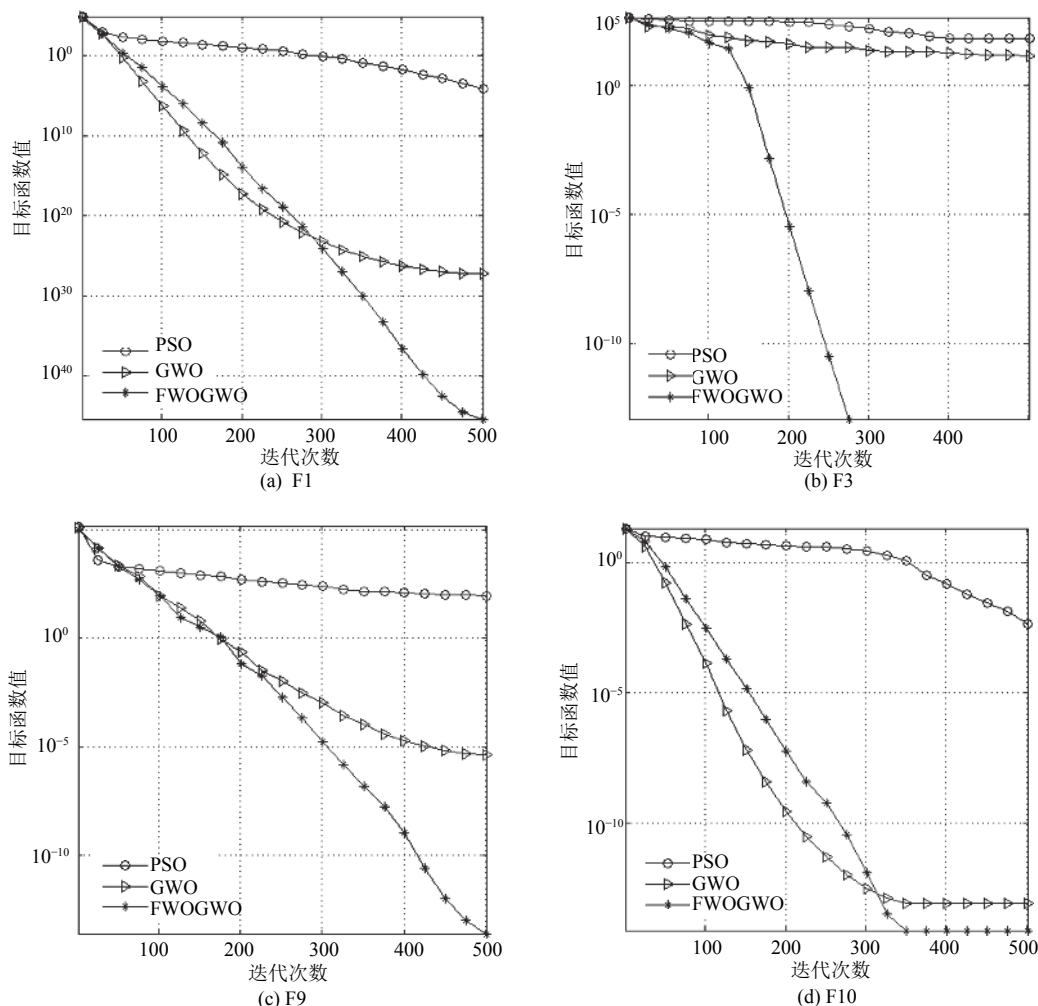


图6 算法针对不同函数的收敛曲线示意图

参考文献

- Goldberg DE. Genetic algorithms in search, optimization, and machine learning. Boston: Addison-Wesley Professional, 1989: 2104-2116.
- Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of ICNN'95-International Conference on Neural Networks. Perth, WA, Australia. 2002. 1942-1948. [doi: 10.1109/ICNN.1995.488968]
- Dorigo M, Birattari M, Stutzle T. Ant colony optimization. IEEE Computational Intelligence Magazine, 2006, 1(4): 28-39. [doi: 10.1109/MCI.2006.329691]
- Bertsimas D, Tsitsiklis J. Simulated Annealing. Statistical Science, 1993, 8(1): 10-15. [doi: 10.1214/ss/1177011077]
- Yang XS. A new metaheuristic bat-inspired algorithm. In: González JR, Pelta DA, Cruz C, et al, eds. Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Berlin: Springer, 2010. 65-74. [doi: 10.1007/978-3-642-12538-6_6]
- Yang XS. Firefly algorithms for multimodal optimization. In: Watanabe O, Zeugmann T, eds. Stochastic Algorithms: Foundations and Applications. Berlin: Springer. 2010. 169-178.

- 7 Yang XS, Deb S. Cuckoo search via levy flights. World 2009 Congress on Nature & Biologically Inspired Computing. Coimbatore, India. 2010. 210–214. [doi: [10.1109/NABIC.2009.5393690](https://doi.org/10.1109/NABIC.2009.5393690)]
- 8 Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. *Advances in Engineering Software*, 2014, 69: 46–61. [doi: [10.1016/j.advengsoft.2013.12.007](https://doi.org/10.1016/j.advengsoft.2013.12.007)]
- 9 Komaki GM, Kayvanfar V. Grey wolf optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time. *Journal of Computational Science*, 2015, 8: 109–120. [doi: [10.1016/j.jocs.2015.03.011](https://doi.org/10.1016/j.jocs.2015.03.011)]
- 10 Daniel E, Anitha J, Kamaleshwaran KK, *et al.* Optimum spectrum mask based medical image fusion using gray wolf optimization. *Biomedical Signal Processing and Control*, 2017, 34: 36–43. [doi: [10.1016/j.bspc.2017.01.003](https://doi.org/10.1016/j.bspc.2017.01.003)]
- 11 Shayeghi H, Asefi S, Younesi A. Tuning and comparing different power system stabilizers using different performance indices applying GWO algorithm. *International Comprehensive Competition Conference on Engineering Sciences*. Anzali, Iran. 2016.
- 12 Mirjalili S. How effective is the grey wolf optimizer in training multi-layer perceptrons. *Applied Intelligent*, 2015, 43(1): 150–161. [doi: [10.1007/s10489-014-0645-7](https://doi.org/10.1007/s10489-014-0645-7)]
- 13 Faris H, Aljarah I, Al-Betar MA, *et al.* Grey wolf optimizer: A review of recent variants and applications. *Neural Computing and Applications*, 2018, 30(2): 413–435. [doi: [10.1007/s00521-017-3272-5](https://doi.org/10.1007/s00521-017-3272-5)]
- 14 Kohli M, Arora S. Chaotic grey wolf optimization algorithm for constrained optimization problems. *Journal of Computational Design and Engineering*, 2017. [doi: [10.1016/j.jcde.2017.02.005](https://doi.org/10.1016/j.jcde.2017.02.005)]
- 15 Natesan G, Chokkalingam A. Opposition learning-based grey wolf optimizer algorithm for parallel machine scheduling in cloud environment. *International Journal of Intelligent Engineering and Systems*, 2017, 10(1): 186–195. [doi: [10.22266/ijies2017.0228.20](https://doi.org/10.22266/ijies2017.0228.20)]
- 16 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. *控制与决策*, 2016, 31(11): 1991–1997.
- 17 Mittal N, Singh U, Sohi BS. Modified grey wolf optimizer for global engineering optimization. *Applied Computational Intelligence and Soft Computing*, 2016, 2016(8). [doi: [10.1155/2016/7950348](https://doi.org/10.1155/2016/7950348)]
- 18 Singh N, Singh SB. Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance. *Journal of Applied Mathematics*, 2017, 2017: 2030489. [doi: [10.1155/2017/2030489](https://doi.org/10.1155/2017/2030489)]
- 19 Tawhid MA, Ali AF. A hybrid grey wolf optimizer and genetic algorithm for minimizing potential energy function. *Memetic Computing*, 2017, 9(4): 347–359. [doi: [10.1007/s12293-017-0234-5](https://doi.org/10.1007/s12293-017-0234-5)]
- 20 龙文, 伍铁斌. 协调探索和开发能力的改进灰狼优化算法. *控制与决策*, 2017, 32(10): 1749–1757.
- 21 Muro C, Escobedo R, Spector L, *et al.* Wolf-pack (*Canis lupus*) hunting strategies emerge from simple rules in computational simulations. *Behavioural Processes*, 2011, 88(3): 192–197. [doi: [10.1016/j.beproc.2011.09.006](https://doi.org/10.1016/j.beproc.2011.09.006)]
- 22 石绍应, 王小谟, 曹晨, 等. 规则数确定的自适应模糊分类器. *西安电子科技大学学报(自然科学版)*, 2017, 44(2): 81–87.