

融合自适应差分进化机制的多目标灰狼优化算法

赵云涛 谌竟成 李维刚

(武汉科技大学冶金自动化与检测技术教育部工程研究中心 武汉 430081)

摘要 针对灰狼算法易于陷入局部最优问题,提出了一种融合自适应差分进化机制的多目标灰狼优化算法。首先,将外部种群 Archive 按目标函数值的距离进行分组以避免存储相似个体。其次,设置头狼选择机制,在外部种群中选择头狼。最后,在更新过程中引入差分进化,择优选择下一代灰狼,同时差分进化参数可根据候选解加权目标函数值动态地自适应调整,平衡算法的局部开发与全局探测性能。基于 8 个多目标测试函数的验证结果表明,提出的多目标灰狼优化算法的收敛性与分布性优于其他 3 种算法。

关键词 多目标优化,灰狼算法,参数自适应,差分进化

中图分类号 TP301.6 **文献标识码** A

Multi-objective Grey Wolf Optimization Hybrid Adaptive Differential Evolution Mechanism

ZHAO Yun-tao CHEN Jing-cheng LI Wei-gang

(Engineering Research Center for Metallurgical Automation and Detecting Technology of Ministry of Education,
Wuhan University of Science and Technology, Wuhan 430081, China)

Abstract Due to the grey wolf algorithm is easy to fall into local optimum, a multi-objective grey wolf optimization based on adaptive differential evolution mechanism was proposed. Firstly, the external archive is grouped according to the distance of the objective function value to avoid storing similar individuals. Secondly, the selection mechanism of the head wolf is adopted. Finally, differential evolution is introduced into the updating process to select the next generation of grey wolves. At the same time, the parameters of differential evolution are adaptively adjusted according to the objective value of candidate solutions, to balance the local exploitation and the global exploration performance. The experimental results show that the proposed multi-objective grey wolf optimization has better convergence and distribution than the other three algorithms.

Keywords Multi-objective optimization, Grey wolf algorithm, Parametric adaptation, Differential evolution

1 引言

生产生活中存在许多优化问题,相比单目标优化,因要满足多个目标要求,多目标问题求解往往比单目标困难^[1]。通常将多目标问题分解,通过加权系数将多目标转化为单目标问题,但在工程实际中加权系数往往很难确定。因此,近年来,许多学者尝试将单目标优化问题上表现优异的进化算法用于求解多目标优化^[2],如多目标差分算法^[3-4]、多目标遗传算法^[5-6]、多目标粒子群算法^[7]等。

灰狼算法^[8]是 2015 年提出的一种新的智能优化算法,由模拟灰狼群体捕食行为而来,主要包括“跟踪”“包围”“追捕”“攻击”4 个步骤^[9]。该算法因其收敛速度快、性能好,在众多领域得到了研究与应用^[10-13]。基于灰狼算法优秀的搜索性能, Mirjalili 等提出了多目标灰狼算法(MOGWO)^[14], MOGWO 基于网格机制和选择引导更新外部种群 Archive,但其也存在陷入局部最优的缺点。

为了解决上述问题,本文将外部种群 Archive 按目标函数值的距离进行分组,避免存储相似个体。在种群更新过程中引入差分进化机制,让个体狼具有多个可选位置并择优选择。同时,个体狼在更新位置时,根据可选解加权目标函数值

动态地自适应调整差分进化参数,平衡算法的局部开发与全局探测性能。最终得到融合自适应差分进化机制的多目标灰狼优化算法(DE-MOGWO)。通过 8 个多目标测试函数仿真测试的结果验证了 DE-MOGWO 算法有效性。

2 基本概念

2.1 多目标优化

假设 n 维解向量 $x = (x_1, x_2, \dots, x_i, \dots, x_n)$, $i = 1, \dots, n$, m 个目标函数 $f = (f_1(x), f_2(x), \dots, f_j(x), \dots, f_m(x))$, $j = 1, \dots, m$ 。本文给出如下定义。

定义 1 假设解向量 A 和 B 满足 $f_j(A) < f_j(B)$ 时,称为解 A 支配解 B , 记为 $A > B$ 。

定义 2 解 x^* 被称为最优解, 仅当满足 $x^* > x$, ($x^* \neq x$)。

定义 3 所有最优解对应的目标函数值组成的曲面称为 Pareto 前沿面, 记为: PF^* 。

2.2 灰狼优化算法

灰狼优化算法模仿自然界具有严格等级制度的狼群的捕食过程, 由头狼领导, 其他个体狼不断地向头狼靠近^[8]。

在灰狼优化算法中, 当前最优的 3 个解记为 α 狼、 β 狼及

δ 狼,其他个体通过 3 只头狼的引导来围捕猎物,围捕过程如下:

$$D_{\alpha,\beta,\delta} = |C \cdot X_p(t) - X_j(t)|, p = \alpha, \beta, \delta \quad (1)$$

$$X_j(t+1) = ((X_\alpha(t) - A \cdot D_\alpha) + (X_\beta(t) - A \cdot D_\beta) + (X_\delta(t) - A \cdot D_\delta)) * (1/3) \quad (2)$$

$$C = 2r_1 \quad (3)$$

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

$$a = 2 - 2\left(\frac{t}{Max_iter}\right) \quad (5)$$

其中, t 为当前的迭代次数, $X(t)$ 表示个体狼当前的位置,通过式(1)计算出个体狼与头狼的距离,通过式(2)围捕猎物,其中 r_1 和 r_2 为 $[0,1]$ 内的随机数, Max_iter 为最大迭代次数, a 值随着迭代次数的增加线性减少, A 和 C 为摆动因子,由上式可知算法迭代前期因具有较大 A 值,所以进行全局性探测,后期 A 值减小完成局部精细开发。

3 多目标灰狼算法

为了将灰狼算法应用于多目标优化问题,在灰狼算法中引入外部种群 Archive,用于存储非支配最优解。采用领导者选择策略,从外部种群 Archive 选择捕食过程中的领导者 α 狼、 β 狼及 δ 狼。

3.1 外部种群 Archive 机制

算法每次迭代会产生新的个体,当个体要加入外部种群 Archive 时,将这些个体逐一与 Archive 中的个体进行比较,将会出现 3 种情况:

- 1) 如果新个体被 Archive 中的所有个体支配,则该个体不加入 Archive;
- 2) 如果新个体支配 Archive 中的一个或多个个体,则新个体加入 Archive,同时将被其支配的个体从 Archive 中删除;
- 3) 如果新个体与 Archive 中的任一个体互不支配,则将个体加入 Archive。

为了避免存储太多相似个体,Archive 会将所有个体按目标函数值的距离进行分组,分组策略如下:

- 1) 计算所有个体的目标函数即 $(f_1(x), f_2(x), \dots, f_j(x), \dots, f_m(x))$, 分别找出其中每一个目标函数 $f_j(x)$ 的极值,记作 $f_{j\max}$ 和 $f_{j\min}$;
- 2) 对于每一个目标函数 $f_j(x)$, 以 $f_{j\max}$ 和 $f_{j\min}$ 为上下界,平均划分出 Q 个区间并分别标注为 1 到 Q ;
- 3) 对于每一个个体,将其第 j 个目标函数值 $f_j(x)$ 对应的区间数记作 $q_j, q = 1, \dots, Q$, 则 $(q_1, q_2, \dots, q_j, \dots, q_m)$ 就是该个体所在的组。上述分组策略中的 Q 由实际分组需求给出。

在迭代过程中,若 Archive 种群中的空间已满,算法在最拥挤组中随机删除某些个体,新的个体被插入到不拥挤组中,提高接近最优前沿的多样性。

3.2 头狼选择机制

在 Archive 中存放所有的非支配最优解,采用轮盘赌的方式从 Archive 中选取头狼(包括 α 狼、 β 狼及 δ 狼)。为了提高算法的探索能力,每一个个体被选择的概率与其所在组的个体数成反比,如式(6)所示:

$$P_j = \left(\frac{1}{N_j}\right)^c \quad (6)$$

其中, C 为大于 1 的常数,根据实际需求设置; N_j 为该个体所

在组的个体总数。

3.3 融合自适应差分进化机制

差分进化算法由遗传算法改进而来^[15-16],将差分进化算子融入灰狼算法。个体向头狼更新时,保留原个体狼向头狼更新位置时每只头狼各占 1/3 的策略,利用头狼之间的个体差异性,在空间进行跳跃搜索,增加种群的多样性。因 α, β 及 δ 狼为从 Archive 中选择出的不同狼,没有差劣之分。每只头狼通过差分策略可更新得到 3 个可供个体狼选择的位置,加上原来各 1/3 权重产生的解,头狼便可提供 4 个候选解。具体如下:

$$X_{j1}(t+1) = u \cdot (X_\alpha(t) - A \cdot D_\alpha) + k \cdot ((X_\beta(t) - A \cdot D_\beta) - (X_\delta(t) - A \cdot D_\delta)) \quad (7)$$

$$X_{j2}(t+1) = u \cdot (X_\beta(t) - A \cdot D_\beta) + k \cdot ((X_\alpha(t) - A \cdot D_\alpha) - (X_\delta(t) - A \cdot D_\delta)) \quad (8)$$

$$X_{j3}(t+1) = u \cdot (X_\delta(t) - A \cdot D_\delta) + k \cdot ((X_\alpha(t) - A \cdot D_\alpha) - (X_\beta(t) - A \cdot D_\beta)) \quad (9)$$

$$X_{j4}(t+1) = ((X_\delta(t) - A \cdot D_\delta) + (X_\alpha(t) - A \cdot D_\alpha) + (X_\beta(t) - A \cdot D_\beta)) * (1/3) \quad (10)$$

其中, u, k 为差分参数。

保留头狼的特异性,当该头狼附近的解更好时,个体狼最大限度地该头狼靠近寻优,而不是毫无方向地处于三只狼之间的位置,丢失优解;同时,引入另外两只头狼的差异性作为扰动量,防止个体狼在移动时过于向当前头狼靠近,使得群狼聚集,丧失多样性。

差分参数 u 反映不同头狼的支配性,保留每只头狼的最大支配特性,取值在 1 附近左右; k 为差分扰动乘积因子,取值范围为 $[0,1]$,一般小于参数 u ;为了平衡全局性能与局部探索能力,根据个体狼信息动态调整差分参数 k ,调整公式如下:

$$k = k_0 + (q_1 - q_2) \cdot \frac{S_{j1} - \frac{S_{j2} + S_{j3}}{2}}{S_{j2} + S_{j3} - S_{j4}} \quad (11)$$

其中, S_{jm} 为当前个体产生的新位置目标函数值加权和,经排序后有 $S_{j1} < S_{j2} < S_{j3} < S_{j4}$; 参数 k_0 为 k 的基值,参数 q_1, q_2 为 k 基值影响因子。

式(11)动态自动调整差分进化的参数 k 值,若当前产生的 4 个个体差异较小时, k 值越大,下次增大差分扰动量,加强全局性探寻更多的解;反之,如果差异较大时,则减小差分扰动量,进行局部精密开发。

3.4 贪心机制

通过式(7)一式(10),个体狼共产生了 4 个候选解,采用非支配解与目标函数加权求和的方式择优选取,保证种群的优秀率;另引入权值求和,让原算法可以适应不同情况下的多目标函数,增加算法的灵活性。具体如下:

1) 求出个体狼 4 个解中的非支配解。

2) 若有多个非支配解,对多目标函数值加权求和进行排序:

$$S_{jg} = \sum_{j=0}^m \omega_j \cdot f_j(x_{jg}), g = (1, 2, 3, 4) \quad (12)$$

其中, ω_j 为加权参数, $\sum_{j=0}^m \omega_j = 1$ 。

3) 选取 S 值最小的解 x_{jg} 作为当前个体狼最终的更新位置。

3.5 改进后的算法流程

引入自适应差分进化机制的多目标灰狼算法的流程如下:

Step1 初始化狼群,设置外部 Archive,初始化分组参数 Q 、轮盘赌参数 C 、最大迭代次数 Max_iter 、差分参数等。

Step2 计算狼群非支配解,更新外部 Archive。

Step3 按照式(5)与式(11)求取当代初始化参数 k 值与 a 值,按照轮盘赌的方式从 Archive 选取 3 只个体狼。

Step4 依据式(1)–式(5)、式(7)–式(10),得到当前个体狼的 4 个候选解。

Step5 求取候选解中的非支配解,并按照加权和贪心机制,依据式(12)进行排序,择优选取每只狼的更新位置。

Step6 判断本代狼群位置是否更新完毕,如果未更新完毕,则返回至 Step4;否则执行 Step7。

Step7 获取更新完毕后的狼群非支配解,按照 3.1 节中的规则,添加这些非支配解至 Archive 和删除 Archive 部分解。

Step8 判断是否达到最大迭代次数,若是则输出 Archive 解,否则返回至 Step3。

4 实验仿真分析

实验中采用 MatlabR2014a 仿真,运行环境为 Inter(R) Core(TM)i5-4210 处理器,运行内存为 4GB。将多目标灰狼算法(MOGWO)^[14]、多目标粒子群算法(MOPSO)^[17]、多目标差分算法(MODE)^[18]与融合自适应差分机制的多目标灰狼算法(DE-MOGWO)进行比较。DE-MOGWO 的时间复杂度为 $O(mn^2)$,其中 n 为灰狼个数, m 为目标数,其复杂性与其他 3 种算法相当。设置 Max_iter 分别为 15 代、50 代、100 代,搜索算子均为 100,外部种群规模为 100。DE-MOGWO 算法中取 $u=0.85, k_0=0.08, q_1=0.1, q_2=0.9$ 。

4.1 测试函数

表 1 列出了 8 个 CEC2009 竞赛的多目标测试函数, $\epsilon > 0$ 。这些测试函数均被认为是具有挑战性的测试问题,提供了不同的 Pareto 最优前沿。

表 1 多目标测试函数

函数名	函数公式
UF1	$f_1 = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} [x_j - \sin(6\pi x_1 + \frac{j\pi}{n})]^2, f_2 = 1 - \sqrt{x_1} + \frac{2}{ J_2 } \sum_{j \in J_2} [x_j - \sin(6\pi x_1 + \frac{j\pi}{n})]^2$ $J_1 = \{j 2 \leq j \leq n\} \& j \text{ is odd}, J_2 = \{j 2 \leq j \leq n\} \& j \text{ is even}$
UF2	$f_1 = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} y_j^2, f_2 = 1 - \sqrt{x_1} + \frac{2}{ J_2 } \sum_{j \in J_2} y_j^2$ $y_j = \begin{cases} x_j - [0.3x_1^2 \cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1] \cos(6\pi x_1 + \frac{j\pi}{n}), & \text{if } j \in J_1 \\ x_j - [0.3x_1^2 \cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1] \cos(6\pi x_1 + \frac{j\pi}{n}), & \text{if } j \in J_2 \end{cases}$ $J_1 = \{j 2 \leq j \leq n\} \& j \text{ is odd}, J_2 = \{j 2 \leq j \leq n\} \& j \text{ is even}$
UF3	$f_1 = x_1 + \frac{2}{ J_1 } (4 \sum_{j \in J_1} y_j^2 - 2 \prod_{j \in J_1} \cos(\frac{20y_j\pi}{\sqrt{j}}) + 2), f_2 = 1 - \sqrt{x_1} + \frac{2}{ J_2 } (4 \sum_{j \in J_2} y_j^2 - 2 \prod_{j \in J_2} \cos(\frac{20y_j\pi}{\sqrt{j}}) + 2)$ $y_j = x_j - x_1^{0.5(1+\frac{3(j-2)}{n-2})}, j = 2, 3, \dots, n$ $J_1 = \{j 2 \leq j \leq n\} \& j \text{ is odd}, J_2 = \{j 2 \leq j \leq n\} \& j \text{ is even}$
UF4	$f_1 = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} h(y_j), f_2 = 1 - x_1^2 + \frac{2}{ J_2 } \sum_{j \in J_2} h(y_j)$ $y_j = x_j - \sin(6\pi x_1 + \frac{j\pi}{n}), j = 2, 3, \dots, n, h(t) = \frac{ t }{1 + e^{2 t }}$ $J_1 = \{j 2 \leq j \leq n\} \& j \text{ is odd}, J_2 = \{j 2 \leq j \leq n\} \& j \text{ is even}$
UF5	$f_1 = x_1 + (\frac{1}{2N} + \epsilon) \sin(2N\pi x_1) + \frac{2}{ J_1 } \sum_{j \in J_1} h(y_j), f_2 = 1 - x_1 + (\frac{1}{2N} + \epsilon) \sin(2N\pi x_1) + \frac{2}{ J_2 } \sum_{j \in J_2} h(y_j)$ $y_j = x_j - \sin(6\pi x_1 + \frac{j\pi}{n}), j = 2, 3, \dots, n, h(t) = 2t^2 - \cos(4\pi t) + 1$ $J_1 = \{j 2 \leq j \leq n\} \& j \text{ is odd}, J_2 = \{j 2 \leq j \leq n\} \& j \text{ is even}$
UF6	$f_1 = x_1 + \max\{0, 2(\frac{1}{2N} + \epsilon) \sin(2N\pi x_1)\} + \frac{2}{ J_1 } (4 \sum_{j \in J_1} y_j^2 - 2 \prod_{j \in J_1} \cos(\frac{20y_j\pi}{\sqrt{j}}) + 2)$ $f_2 = 1 - x_1 + \max\{0, 2(\frac{1}{2N} + \epsilon) \sin(2N\pi x_1)\} + \frac{2}{ J_2 } (4 \sum_{j \in J_2} y_j^2 - 2 \prod_{j \in J_2} \cos(\frac{20y_j\pi}{\sqrt{j}}) + 2)$ $y_j = x_j - \sin(6\pi x_1 + \frac{j\pi}{n}), j = 2, 3, \dots, n$ $J_1 = \{j 2 \leq j \leq n\} \& j \text{ is odd}, J_2 = \{j 2 \leq j \leq n\} \& j \text{ is even}$
UF7	$f_1 = \sqrt[5]{x_1} + \frac{2}{ J_1 } \sum_{j \in J_1} y_j^2, f_1 = 1 - \sqrt[5]{x_1} + \frac{2}{ J_2 } \sum_{j \in J_2} y_j^2$ $y_j = x_j - \sin(6\pi x_1 + \frac{j\pi}{n}), j = 2, 3, \dots, n$ $J_1 = \{j 2 \leq j \leq n\} \& j \text{ is odd}, J_2 = \{j 2 \leq j \leq n\} \& j \text{ is even}$
UF8	$f_1 = 0.5[\max\{0, (1+\epsilon)(1-4(2x_1-1)^2)\} + 2x_1]x_2 + \frac{2}{ J_1 } \sum_{j \in J_1} (x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $f_2 = 0.5[\max\{0, (1+\epsilon)(1-4(2x_1-1)^2)\} + 2x_1]x_2 + \frac{2}{ J_2 } \sum_{j \in J_2} (x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $f_3 = 1 - x_2 + \frac{2}{ J_3 } \sum_{j \in J_3} (x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $J_1 = \{j 3 \leq j \leq n\} \& j-1 \text{ is a multiplication of } 3, J_2 = \{j 3 \leq j \leq n\} \& j-2 \text{ is a multiplication of } 3,$ $J_3 = \{j 3 \leq j \leq n\} \& j \text{ is a multiplication of } 3$

4.2 性能指标

为了更加直观地比较算法在改进前后的性能差异,选取反向世代距离(IGD)^[17]作为算法的性能指标。该指标的计算公式如下:

$$IGD = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|} \quad (13)$$

其中, P 为算法所求得的 Pareto 边界的近似解集, P^* 是多目标问题真实 Pareto 边界上的个体集合, $|P^*|$ 表示 P^* 的个体数目, d 表示 P 中的个体 v 与 P^* 的最小欧几里德距离。

IGD 用于评价算法的综合性能,可以较为全面地反映算法的收敛性和分布性。IGD 越小,算法的计算结果与真实的 Pareto 边界越接近,算法的收敛性与分布性就越好。

4.3 实验结果

DE-MOGWO 算法、MOGWO 算法、MOPSO 算法、MODE 算法对所有的测试函数迭代运行 100 次并进行 30 次独立运算,取 IGD 均值和方差作为评价算法的性能指标,结果如表 2 所列,表中的较小值被加粗显示。为了更加清晰地显示 4 种算法的结果,采用箱线图(见图 1、图 2)展示,其中水平线表示中位数,上下两条线分别表示四分位数。图 1 和

图 2 中, x 轴上的 1-4 分别对应 DE-MOGWO 算法、MOGWO 算法、MOPSO 算法、MODE 算法。

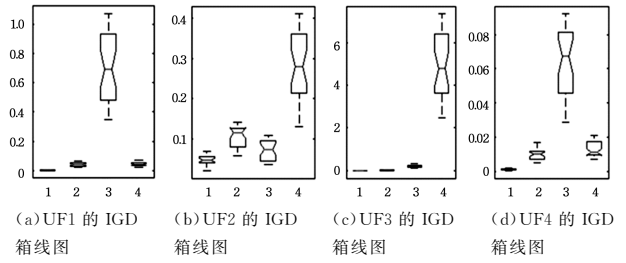


图 1 UF1-UF4 IGD 箱线图

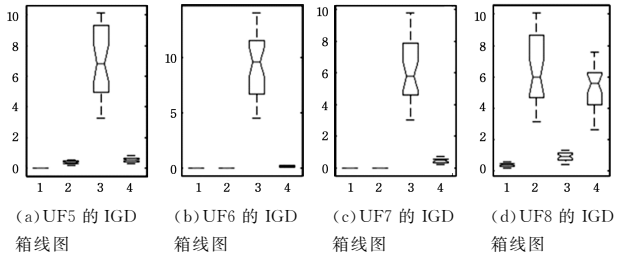


图 2 UF5-UF8 IGD 箱线图

表 2 IGD 测试结果

MOP	均值				方差			
	DE-MOGWO	MOGWO	MOPSO	MODE	DE-MOGWO	MOGWO	MOPSO	MODE
UF1	2.33×10^{-3}	3.62×10^{-2}	6.54×10	4.34×10^{-2}	5.47×10^{-2}	3.35×10	4.66	3.63
UF2	4.10×10^{-2}	8.57×10^{-2}	6.89×10^{-2}	2.45×10	3.13×10^{-3}	3.50×10^{-2}	2.21×10^{-2}	9.21×10^{-2}
UF3	3.02×10^{-3}	8.92×10^{-3}	2.04×10	4.35	4.43×10^{-4}	1.07×10^{-3}	6.17×10^{-2}	9.34×10
UF4	1.08×10^{-3}	9.91×10^{-3}	5.49×10^{-2}	1.27×10^{-2}	9.53×10^{-4}	2.32×10^{-3}	3.39×10^{-3}	1.21×10^{-3}
UF5	9.09×10^{-3}	3.21×10	6.04	4.83×10	8.91×10^{-2}	1.22×10^{-2}	5.46×10	2.01×10^{-2}
UF6	1.24×10^{-3}	9.80×10^{-3}	8.31	1.32×10	6.82×10^{-4}	3.44×10^{-3}	4.33×10	2.11×10^{-4}
UF7	5.01×10^{-3}	4.32×10^{-3}	5.75	4.32×10	5.31×10^{-2}	3.57×10^{-2}	8.24×10	9.51
UF8	3.41×10	6.03	7.75×10	4.52	6.51×10^{-2}	4.03×10^{-2}	9.68×10^{-2}	3.89×10

箱线图反映了 IGD 统计数据的分散情况,由图 1 和图 2 的箱线图可知,DE-MOGWO 中位数较小,数据分布集中解的质量较其他算法更优。IGD 均值反映了算法的收敛性和分布性,方差代表算法的鲁棒性。由表 2 可知,改进后的 DE-MOGWO,UF1-UF8 的 IGD 均值比其他 4 种算法都要小,说明算法收敛性和分布性较其他 3 种算法有更强的竞争力;对于方差结果,除 UF5、UF7,DE-MOGWO 方差均小于其他 3 种算法,算法有较强的稳定性。

图 3—图 10 中,蓝色为 DE-MOGWO 所寻的最优非支配解,红色为 MOGWO 所寻最优非支配解,黑色为 MOPSO 所寻最优非支配解,绿色为 MODE 所寻最优非支配解。对于多目标函数 UF1,UF5,UF6,UF7,由图 3、图 7—图 9 可知,MOPSO 无法找到最优的 Pareto 前沿面,相比较而言,MOGWO 解的质量得到提高,但是由图 9 可知,MOGWO 表现不稳定,陷入局部最优,而 DE-MOGWO 所找到的非支配解更加接近于真实的 Pareto 前沿面,且未出现不稳定现象。

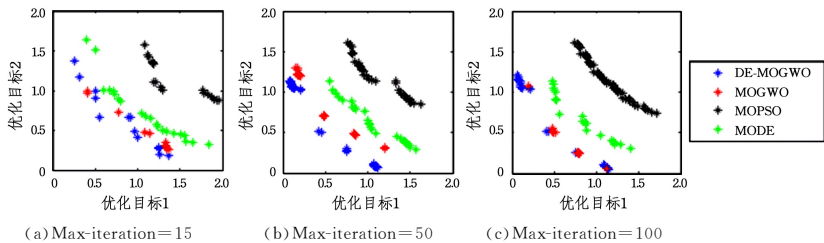


图 3 UF1 的 Pareto 前沿解集(电子版为彩色)

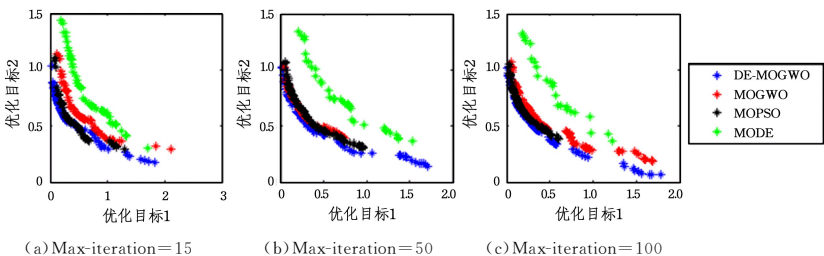


图 4 UF2 的 Pareto 前沿解集(电子版为彩色)

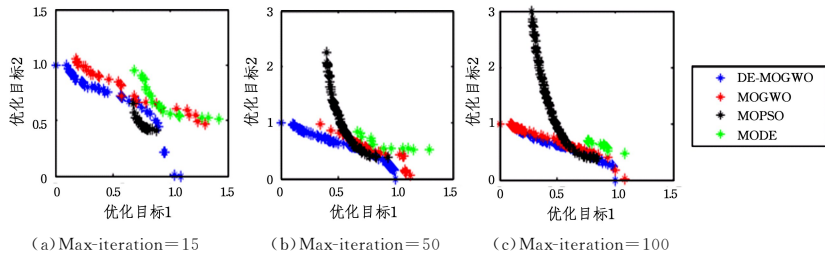


图 5 UF3 的 Pareto 前沿解集(电子版为彩色)

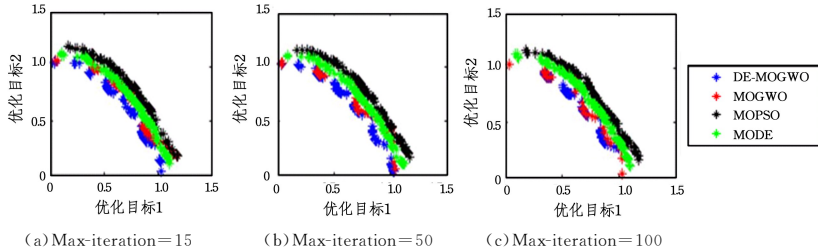


图 6 UF4 的 Pareto 前沿解集(电子版为彩色)

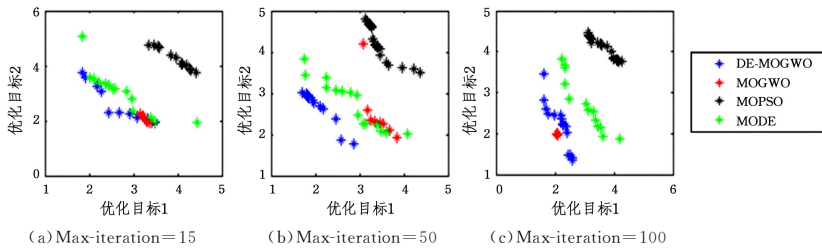


图 7 UF5 的 Pareto 前沿解集(电子版为彩色)

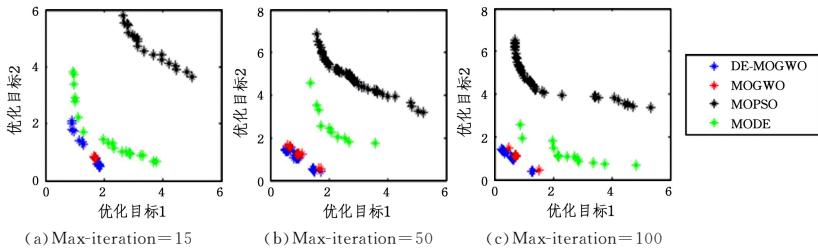


图 8 UF6 的 Pareto 前沿解集(电子版为彩色)

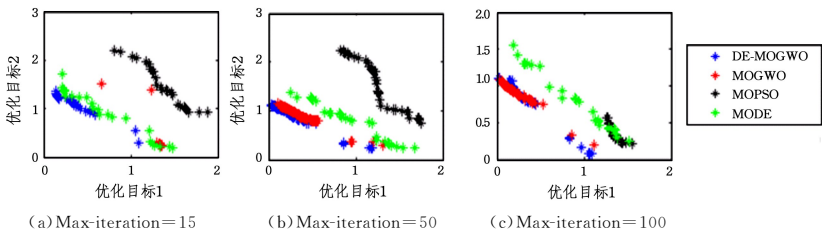


图 9 UF7 的 Pareto 前沿解集(电子版为彩色)

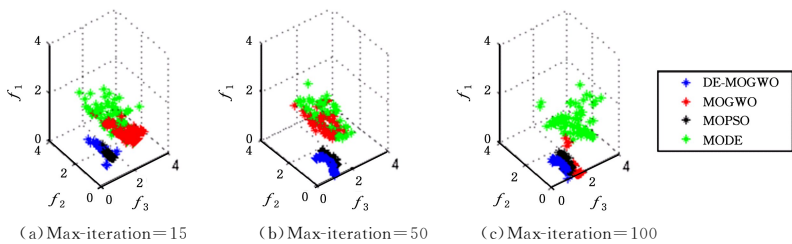


图 10 UF8 的 Pareto 前沿解集(电子版为彩色)

对于多目标函数 UF2,UF3,UF4,由图 4—图 6 可知,虽然 DE-MOGWO 与 MOGWO 算法有部分重叠,但是 DE-MOGWO 得到的前沿面分布较广泛,多样性好,接近于真实的 Pareto 前沿面;由图 4 可知,MODE 对此类多目标函数易

陷入局部最优,不宜于实际应用。对于 3 目标函数 UF8,由图 10 可知,DE-MOGWO 算法相对于 MOPSO, MOGWO, MODE,非支配解的分布也比较广泛,能形成真实的前沿面。

此外,经过多次独立测试,在相同的迭代次数下,DE-

MOGWO与MOPSO和MOGWO所用时间相差无几,但是由图3—图10可知,在相同迭代次数下,DE-MOGWO算法能够探索到更优的非支配解,且解的多样性也较MOPSO,MOGWO,MODE更好。

结束语 基于灰狼算法提出了一种新的融合自适应差分进化机制的多目标灰狼算法。在种群更新过程中,引入外部种群与头狼选择机制,结合差分进化,增加个体狼在更新位置时的多样性。通过非支配解与权值分配机制择优选取,同时自适应调整个体狼的差分参数,均衡种群的全局性能与局部性能。与MOPSO,MOGWO,MODE的对比测试表明,融合自适应差分进化的多目标灰狼算法的性能强于其他3种算法。

DE-MOGWO通过差分进化会产生多余的非支配解,这些解往往含有优秀信息,在算法执行过程中被丢弃较为可惜。后续的研究中,可选用不同的策略,利用多个非支配解获取更优信息。

参考文献

- [1] 王亚辉,吴金妹,贾晨辉.基于动态种群多策略差分进化模型的多目标进化算法[J].电子学报,2016,44(6):1472-1474.
- [2] 周欢,孟利民,王丽萍,等.动态邻域的分解多目标进化算法[J].小型微型计算机系统,2017(9):2039-2044.
- [3] YE H T, LUO F, XU Y G. Differential evolution for solving multi-objective optimization problems: a survey of the state-of-the-art [J]. Control Theory & Applications, 2013, 30(7): 922-927.
- [4] 刘宝,董明刚,敬超.改进的排序变异多目标差分进化算法[J].计算机应用,2018,38(8):2157-2163.
- [5] 王福才,周鲁平.混合精英策略的元胞多目标遗传算法及其应用

[J].电子学报,2015,38(7):1398-1405.

- [6] 王国豪,李庆华,刘安丰.多目标最优化工作流调度进化遗传算法[J].计算机科学,2018,45(5):38-44.
- [7] 章恩泽,陈庆伟.改进的r支配高维多目标粒子群优化算法[J].控制理论与应用,2015,32(5):623-629.
- [8] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer [J]. Advances in Engineering Software, 2014, 69(3): 46-61.
- [9] 陈闯, CHELLAI R, 刑尹, 等.采用动态权重和概率扰动策略改进的灰狼优化算法[J].计算机应用,2017,37(10):2854-2860.
- [10] 张悦,孙惠香,魏政军,等.具有自适应调整策略的混沌灰狼优化算法[J].计算机科学,2017,44(11A):120-123.
- [11] 白建川,夏克文,牛文佳,等.新型灰狼算法的粗糙集属性约简及应用[J].计算机工程与应用,2017,53(24):182-186.
- [12] 徐辰华,李成县,喻昕,等.基于Cat混沌与高斯变异的改进灰狼优化算法[J].计算机工程与应用,2017,53(4):1-9.
- [13] 胡小平,曹敬.改进灰狼优化算法在WSN节点部署中的应用[J].传感技术学报,2018,31(5):101-106.
- [14] MIRJALILI, SAREMI. Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization [J]. Expert Systems with Applications, 2016, 47: 106-119.
- [15] 李龙涛,翁晴晴.基于反向学习的自适应差分进化算法[J].计算机应用,2018,38(2):399-404.
- [16] 赵志伟,杨景明,呼子宇,等.基于角度邻域的多目标差分进化算法[J].控制理论与应用,2017,34(1):22-29.
- [17] 屈敏,高岳林,江巧永.基于Pareto邻域交叉算子的多目标粒子群优化算法[J].计算机应用,2011,31(7):1789-1792.
- [18] 侯莹,韩红桂,乔俊飞.基于参数动态调整的多目标差分进化算法[J].控制与决策,2017,32(11):1986-1990.

(上接第61页)

验结果表明,其真实度远高于文献[1,3]算法,由此可见,本文算法具有高效率和高收敛性,在航迹拟合上远优于其他算法。

参考文献

- [1] 黄卫芳,王伟,刘鸿飞.雷达飞行航迹拟合分析系统初步研究与实现[J].空中交通管理,2008(5):10-12.
- [2] 汤继强.某机场终端区进近程序优化设计研究[D].成都:电子科技大学,2011.
- [3] 连德忠,吴文城,游德有,等.三次样条插值的新算法[J].龙岩学院学报,2017,35(5):4-7.
- [4] 邢丽.一维插值算法在实际问题中的应用和比较[J].上海第二工业大学学报,2013,30(4):311-314.
- [5] 邓兴升,汤仲安.移动格林基函数样条二维插值算法研究[J].大地测量与地球动力学,2011,31(6):69-72.
- [6] 王娜,王健,曹智明,等.加权平均预测的一阶多智能体一致性的采样控制[J].空军工程大学学报(自然科学版),2017,18(1):105-110.
- [7] ZIMA K. Fuzzy case based reasoning in sports facilities unit cost estimating[C]//International Conference of Numerical Analysis & Applied Mathematics. 2016.
- [8] 刘喜春,王超,王文广,等.基于多Agent强化学习的战时备件供应保障动态协调机制[J].空军工程大学学报(自然科学版),2009(3):59-63.
- [9] GANJEHKAVIRI A, MOHD JAAFAR M N, HOSSEINI S E, et al. Genetic algorithm for optimization of energy systems: Solution uniqueness, accuracy, Pareto convergence and dimension

reduction[J]. Energy, 2017, 119: 167-177.

- [10] 孙京浩,杨嘉雄,王硕,等.基于Actor-Critic和神经网络的闭环脑机接口控制器设计[J/OL]. https://doi.org/10.13195/j.kzyjc.2017.0791.
- [11] 翟建伟.基于深度Q网络算法与模型的研究[D].苏州:苏州大学,2017.
- [12] 周海平.陆态网络基准网数据处理策略分析[J].地理空间信息,2018,16(9):120-122,12.
- [13] SHI Y B, SI L, FENG G B, et al. Numerical and experimental study on liquid crystal optical phased array beam steering combined with stochastic parallel gradient descent algorithm[J]. Optik- International Journal for Light and Electron Optics, 2016, 127(3):1450-1454.
- [14] MOKHTARI A, RIBEIRO A. Regularized stochastic BFGS algorithm[P]. 2013.
- [15] 郝志峰,黄浩,蔡瑞初,等.基于多特征融合与双向RNN的细粒度意见分析[J].计算机工程,2018,44(7):199-204,211.
- [16] KUMAR R, BASKAR S. B-spline quasi-interpolation based numerical methods for some Sobolev type equations[J]. Journal of Computational and Applied Mathematics, 2016, 292(C): 41-66.
- [17] 张振兴,杨任农,张彬超,等.空战飞行对敌目标逼近航迹预测仿真[J].空军工程大学学报(自然科学版),2018,19(2):33-37.
- [18] 刘进忙,冯有前,张晓刚.基于最小二乘法Lagrange插值基函数的拟合推广[J].空军工程大学学报(自然科学版),2002,(4):84-87.
- [19] 王莉莉,彭勃.航迹点特征的时间窗分割算法的航迹聚类[J].空军工程大学学报(自然科学版),2018,19(3):19-23.