

模糊自适应排序变异多目标差分进化算法

董明刚^{1,2} 刘 宝¹ 敬 超^{1,2}

(桂林理工大学信息科学与工程学院 广西 桂林 541004)¹

(广西嵌入式技术与智能系统重点实验室 广西 桂林 541004)²

摘 要 为提高多目标差分进化算法在求解问题时的收敛性和多样性,提出了一种模糊自适应排序变异多目标差分进化算法。首先,采用模糊系统自适应调节排序变异参数,均衡了算法的局部搜索能力和全局探索能力,在加快算法收敛速度的同时,减小了陷入局部最优的可能性;其次,采用均匀种群初始化方法,在算法开始阶段获得了一个分布均匀的初始种群,提高了算法的稳定性和多样性;最后,增加一个临时的种群以存储被丢弃的个体,用于每一代优化后的最终选择,提高了种群进化过程中的多样性。采用 7 个标准测试函数和 3 个具有偏好特征的测试函数进行仿真实验,并将所提算法与其他 4 种多目标进化算法进行对比。实验结果表明,所提算法在收敛性和多样性方面整体上优于其他几种对比算法,可以有效地逼近真实 Pareto 前沿。同时,实验也验证了所提算法中模糊自适应排序变异策略的有效性。

关键词 多目标优化问题,差分进化,排序变异,模糊系统,自适应策略

中图分类号 TP183 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.07.034

Multi-objective Differential Evolution Algorithm with Fuzzy Adaptive Ranking-based Mutation

DONG Ming-gang^{1,2} LIU Bao¹ JING Chao^{1,2}

(College of Information Science and Engineering, Guilin University of Technology, Guilin, Guangxi 541004, China)¹

(Guangxi Key Laboratory of Embedded Technology and Intelligent System, Guilin, Guangxi 541004, China)²

Abstract In order to improve the convergence and diversity of the multi-objective differential evolution algorithm in solving multi-objective optimization problems, this paper proposed a multi-objective differential evolution algorithm with fuzzy adaptive ranking-based mutation. Firstly, the global exploration and the local exploitation are balanced by using the fuzzy system which adaptively adjust the parameters of ranking-based mutation, so the convergence rate of the algorithm is accelerated and the possibility of the algorithm falling into a local optimum is reduced. Secondly, for the sake of improving the stability and diversity of the algorithm, an initial population with good diversity is obtained through the uniform population initialization method at the beginning of the algorithm. Finally, the discarded individuals is stored by adding them to a temporary population for the final selection in the end of each iteration, therefore, the population diversity during the evolution process is improved. Simulation experiments were conducted on the seven standard test functions and three test functions with bias features. The experimental results show that compared with other four algorithms, the proposed algorithm has better convergence and diversity, and it can effectively approach to the real Pareto frontier. The effectiveness of the fuzzy adaptive ranking-based mutation strategy in the proposed algorithm is also verified by experimental comparison method.

Keywords Multi-objective optimization problem, Differential evolution, Ranking-based mutation, Fuzzy system, Adaptive strategy

1 引言

多目标优化问题广泛存在于工程实践和科学研究中,与

单目标优化问题相比,多目标优化问题拥有两个或者更多的目标函数需要同时优化,并且这些目标函数之间相互联系,彼此冲突,很难找到一个解能在所有优化目标函数上同时达到

最优值,因此多目标优化问题的主要求解目标是找出一个在所有目标函数上折衷解的集合,即 Pareto 最优解集^[1]。多目标差分进化算法作为一种新的优化算法,采用实数编码,主要通过变异操作进行全局寻优搜索,具有较强的寻优性能。随着多目标差分进化算法在电力系统调度优化^[2]、带时间窗的车辆路径问题^[3]和无人机路径规划^[4]等多目标优化问题领域的广泛应用,多目标差分进化算法的研究具有越来越重要的理论意义和实际意义。

近年来,为提高多目标差分进化算法的寻优性能,Zhang 等在基于分解的多目标进化算法(Multi-objective Evolutionary Algorithm based on Decomposition, MOEA/D)^[5]的基础上,融入差分进化方法,提出了一种基于分解的多目标差分进化算法(Multi-Objective Evolutionary Algorithm based on Decomposition with Differential Evolution, MOEA/D-DE)^[6],用以解决多目标优化中具有复杂 Pareto 集合的问题。针对差分进化算法求解多目标优化问题时容易陷入局部最优的问题,宋通等^[7]通过对相反进化方向产生的两个子代个体进行评价,并采用多种群机制,降低了算法陷入局部最优的风险。谢承旺等提出了一种带差分局部搜索的改进型 NSGA2 算法^[8],利用差分进化中变异算子的定向引导作用,与 NSGA2 算法相结合,有效地改善了算法在解群分布上的均匀性和广度性。Jariyatantiwait 等提出了一种基于模糊系统反馈策略的多目标差分进化算法^[9],在不同优化阶段动态平衡算法的局部搜索能力和全局探索能力,从而在加快算法收敛性的同时,保持种群的多样性。Wang 等提出了一种自适应多种群多目标差分进化算法^[10],将数据挖掘的方法引入到多目标差分进化算法中,引导种群的优化,并采用一种均匀种群初始化方法,有效地提高了初始种群的多样性。Ali 等^[11]提出了最优基向量变异策略和基于双种群的选择方法,加快了多目标差分进化算法的收敛速度。Chen 等将文献^[12]中的排序变异思想引入到多目标优化算法中,提出了一种基于排序变异的多目标差分进化算法(Multi-Objective Differential Evolution with Ranking-based Mutation Operator, MODE-RMO)^[13]。在进行变异操作时,对种群中的所有个体依据其适应度值分别赋予不同的选中概率,使得较优个体获得更多的机会参与变异操作,较为有效地提高了算法的寻优性能。

总体而言,尽管近几年来多目标差分进化算法在提高算法寻优性能方面取得了重要进展,但是仍存在收敛性偏慢和多样性欠佳等不足,影响了寻优性能。本文在 MODE-RMO 的基础上,通过采用文献^[9]中的模糊系统反馈策略自适应调节排序变异参数,在加快算法收敛速度的同时,有效地平衡了算法的局部搜索能力和全局探索能力。并采用文献^[10]中的均匀种群初始化方法和文献^[11]中的基于双种群的选择方法,提出了一种模糊自适应排序变异多目标差分进化算法(Multi-Objective Differential Evolution algorithm with Fuzzy Adaptive Ranking-based Mutation, MODE-FARM)。将 MODE-FARM 与其他 4 种算法在 10 个多目标测试函数上进行实验对比,实验结果表明,MODE-FARM 在收敛性和多样

性方面整体上优于其他对比算法,从而验证了本文所提算法在求解多目标优化问题时具有更好的寻优性能。

2 多目标优化问题

多目标优化问题可用公式描述为^[1]:

$$\begin{cases} \min \mathbf{Y} = F(\mathbf{X}) = (f_1(\mathbf{X}), f_2(\mathbf{X}), \dots, f_M(\mathbf{X}))^T \\ \text{s. t. } g_i(\mathbf{X}) \leq 0, i=1, 2, \dots, q \\ h_j(\mathbf{X}) = 0, j=1, 2, \dots, p \end{cases} \quad (1)$$

其中, \mathbf{Y} 为优化目标向量,包含了 M 个子目标函数。 $\mathbf{X} = (x_1, \dots, x_n) \in \Omega \subset \mathbb{R}^n$ 为决策变量, Ω 为决策空间。所有满足约束条件 $g_i(\mathbf{X}) \leq 0$ 和 $h_j(\mathbf{X}) = 0$ 的 \mathbf{X} 构成可行解集合。以下是多目标优化问题中的几个重要定义。

定义 1(Pareto 占优^[1]) 假设 \mathbf{X}_A 和 \mathbf{X}_B 是上述多目标优化问题的可行解,则称与 \mathbf{X}_B 相比, \mathbf{X}_A 是 Pareto 占优的(也称 \mathbf{X}_A 支配 \mathbf{X}_B , 记作 $\mathbf{X}_A > \mathbf{X}_B$), 当且仅当:

$$\begin{aligned} \forall i=1, 2, \dots, M, f_i(\mathbf{X}_A) &\leq f_i(\mathbf{X}_B) \wedge \\ \exists j=1, 2, \dots, M, f_j(\mathbf{X}_A) &< f_j(\mathbf{X}_B) \end{aligned} \quad (2)$$

定义 2(Pareto 最优解^[1]) 可行域内的一个解 \mathbf{X}^* 被称为 Pareto 最优解(或非支配解), 当且仅当:

$$\rightarrow \exists \mathbf{X} \in \Omega: \mathbf{X} > \mathbf{X}^* \quad (3)$$

定义 3(Pareto 最优解集^[1]) 所有 Pareto 最优解的集合被称为 Pareto 最优解集, 即:

$$\mathbf{P}^* = \{ \rightarrow \exists \mathbf{X} \in \Omega: \mathbf{X} > \mathbf{X}^* \} \quad (4)$$

定义 4(Pareto 前沿^[1]) Pareto 最优解集 \mathbf{P}^* 中的所有 Pareto 最优解对应的目标向量组成的曲线或曲面被称为 Pareto 最优前沿 PF^* 。

$$PF^* = \{ F(\mathbf{X}^*) = (f_1(\mathbf{X}^*), f_2(\mathbf{X}^*), \dots, f_M(\mathbf{X}^*))^T \mid \mathbf{X}^* \in \mathbf{P}^* \} \quad (5)$$

3 多目标差分进化算法和 MODE-RMO^[13]

3.1 多目标差分进化算法

传统的多目标差分进化算法^[14]的主要思想可分为如下 5 步:

步骤 1 构建一个规模为 NP 的初始种群,并计算种群中个体的适应度值。针对种群中的每一个体 \mathbf{X}_i , 依次执行步骤 2—步骤 4。

步骤 2 依据变异式(6),对个体 \mathbf{X}_i 执行变异操作。

$$\mathbf{V}_i = \mathbf{X}_{r_1} + F \cdot (\mathbf{X}_{r_2} - \mathbf{X}_{r_3}) \quad (6)$$

其中, F 为规模因子,一般在 $(0, 1)$ 内取值。 \mathbf{X}_{r_1} , \mathbf{X}_{r_2} 和 \mathbf{X}_{r_3} 为种群中随机选取的不同于 \mathbf{X}_i 的个体。

步骤 3 依据式(7)将 $\mathbf{X}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,n})$ 与 $\mathbf{V}_i = (v_{i,1}, v_{i,2}, \dots, v_{i,n})$ 进行交叉操作,得到实验向量 $\mathbf{U}_i = (u_{i,1}, u_{i,2}, \dots, u_{i,n})$ 。其中 CR 为交叉概率,在 $(0, 1)$ 内取值, $rand_j \in (0, 1)$, sn 为 $\{1, 2, \dots, n\}$ 中随机选取的一个数。

$$u_{ij} = \begin{cases} v_{ij}, & \text{if } rand_j \leq CR \text{ or } j = sn \\ x_{ij}, & \text{otherwise} \end{cases}, j=1, 2, \dots, n \quad (7)$$

步骤 4 通过比较 \mathbf{X}_i 和 \mathbf{U}_i 之间的支配关系,选取进入

当前种群的个体。若 U_i 支配 X_i , 则用 U_i 替代 X_i ; 若 X_i 支配 U_i , 则直接丢弃 U_i ; 若 X_i 和 U_i 互不支配, 则将 U_i 加入到当前种群中。

步骤 5 当种群中的所有个体完成变异、交叉和选择后, 当前种群规模增加到了 NP 和 $2NP$ 之间。此时需进行裁剪操作, 即依据 NSGA-II (Nondominated Sorting Genetic Algorithm II)^[15] 中的非支配排序算法和拥挤距离计算公式对种群中的个体进行排序后, 选取前 NP 个较好的个体进入下一代的优化, 直到满足终止条件为止。

3.2 MODE-RMO^[13]

MODE-RMO 与传统的多目标差分进化算法相比, 唯一不同的是将 3.1 节中步骤 2 的变异操作改用排序变异操作, 排序变异公式形式和传统的变异式(6)的形式相同, 但是 X_{r1} 和 X_{r2} 的选择依据不一样, 其具体实现方法如下。

首先, 依据 NSGA-II 中的非支配排序算法和拥挤距离计算公式将所有个体进行排序。再按照式(8)依次赋予所有个体相应的排序值 R_i 。

$$R_i = NP - i, i = 1, 2, \dots, NP \quad (8)$$

其次, 根据式(9)依次计算出种群中所有个体的选择概率, 适应度值较好的个体获得较大的选择概率。

$$P_i = \frac{R_i}{NP}, i = 1, 2, \dots, NP \quad (9)$$

最后, 从种群中依次选择出 X_{r1} , X_{r2} 和 X_{r3} 。对于 X_{r1} , 通过将当前种群中随机选取的不同于 X_i 的个体作为 X_{r1} , 并将其概率值 P_{r1} 与 $(0, 1)$ 范围内的一个随机数进行比较, 若 P_{r1} 不小于随机数值, 则选择成功, 否则重新进行选择。对于 X_{r2} , 其选取方法与 X_{r1} 相同, 且保证 X_{r1} 和 X_{r2} 互不相同。对于 X_{r3} , 只需从当前种群中随机选取一个不同于 X_{r1} 和 X_{r2} 的个体。此时便完成了 X_{r1} , X_{r2} 和 X_{r3} 的选择, 依据式(6)完成对当前种群中个体 X_i 的变异操作。

4 模糊自适应排序变异多目标差分进化算法

4.1 模糊自适应排序变异策略

MODE-RMO 在变异过程中采用排序操作, 以较大的概率选取适应度值较好的个体参与变异, 加快了算法的收敛速度。但是, 由于适应度值较差的个体参与变异的概率较低, 在一定程度上影响着整个种群在进化过程中多样性的保持。

文献[9]提出了一种基于模糊系统反馈策略的多目标差分进化算法, 采用模糊变异式(10)进行变异操作。

$$V_i = \gamma X_{best} + (1 - \gamma) * X_i + F * (X_{r1} - X_{r2}) \quad (10)$$

其中, X_{best} 是当前种群中适应度值最好的个体, X_{r1} 和 X_{r2} 是互不相同且不同于 X_i 的个体。在每一代优化完成后, 计算当代种群中非支配解集的性能度量指标与前一代的相对变化值, 并将其作为模糊系统的输入。通过模糊系统求出下一代的平衡参数 γ 和规模因子 F , 在不同优化阶段动态平衡算法的局部搜索能力和全局探索能力, 从而在加快算法收敛性的同时, 保持种群的多样性。

本文在排序变异操作的基础上, 通过融入文献[9]中的模糊系统反馈方法, 提出了一种模糊自适应排序变异策略。该策略的模糊排序变异公式如式(11)所示:

$$V_i = \gamma X_{r1} + (1 - \gamma) * (X_{r2} + F * (X_{r3} - X_{r4})) \quad (11)$$

其中, X_{r1} 与 3.2 节的排序变异公式中 X_{r1} 的选取方法相同, 即以较大的概率选取当前种群适应度值较好的个体。 X_{r2} , X_{r3} 和 X_{r4} 为当前种群中随机选出的不等于 X_i 的个体, 且它们互不相同。 γ 和 F 的取值方法与文献[9]相同, 即通过将当前种群中非支配解集的性能度量指标与前一代的相对变化值输入模糊系统中计算得出, 并将其应用到下一代的优化。

模糊排序变异式(11)与 3.2 节中描述的排序变异公式相比, 增加了平衡参数 γ , 并且赋予了规模因子 F 一个可以自适应调整的动态值, 每一代优化后通过模糊系统求出下一代的 γ 和 F , 从而在保证排序变异公式收敛性的同时, 提高了种群的多样性。

模糊排序变异式(11)与模糊变异式(10)相比, 将 X_{best} 替换为 X_{r1} , 使得每一个体变异中的基向量可以从当前种群中较好的个体中随机选择, 有利于种群多样性的保持。此外, 由于当前种群包括了未进行变异操作的个体和已经完成择优选择的个体, 因此用当前种群中的随机个体 X_{r2} 替换 X_i , 使得已完成进化后较优的部分个体可以参与未进化个体的变异操作, 有助于算法收敛速度的提高。

为了更清楚地理解模糊自适应排序变异策略在算法的整个运行过程中变异参数的调整变化, 随机选取算法 MODE-FARM 在测试函数 ZDT1^[15] 上的一次运行结果, 绘制算法在整个运行过程中的平衡参数 γ 和规模因子 F 的变化曲线, 如图 1 和图 2 所示。从图中可以看出, 平衡参数 γ 在算法的整个运行过程中逐渐减小, 在 80 代左右趋于稳定; 规模因子 F 在算法的整个运行过程中逐渐增加, 当增加到最大值 1 后保持不变。算法前期收敛阶段, 平衡参数 γ 较大而规模因子 F 较小, 充分利用当前种群中的适应度值较好的个体, 使得整个种群朝着最优区域快速收敛。而当种群中大多数个体都收敛到 Pareto 前沿上或其附近时, 平衡参数 γ 较小而规模因子 F 较大, 此时以当前种群中每一个体作为变异基向量, 从而使得种群中所有个体在 Pareto 前沿上均匀分布。

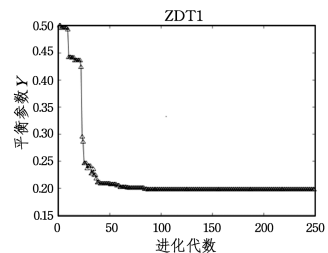


图 1 ZDT1 上平衡参数 γ 的变化曲线

Fig. 1 Variation curve of balance parameter γ on ZDT1

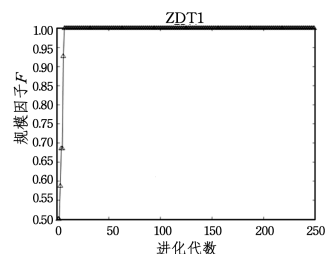


图 2 ZDT1 上规模因子 F 的变化曲线

Fig. 2 Variation curve of scaling factor F on ZDT1

模糊自适应排序变异策略将模糊系统反馈策略融入排序变异操作,在每一代优化完成后,通过模糊系统求出参数 γ 和 F 的值并应用到下一代优化,在不同的优化阶段动态调算法的局部搜索能力和全局探索能力,使得算法在增加收敛速度的同时,保证了种群的多样性,从而提高了算法的寻优性能。

4.2 均匀种群初始化方法

在所有的进化算法中,第一步都是初始化一个规模为 NP 的种群。传统的种群初始化方法是在求解问题可行域内随机产生。为了减少随机性的影响,使得算法每次运行都能获得一个多样性相对较优的初始种群,本文采用文献[10]中介绍的均匀种群初始化方法。

该方法的主要思想是将求解问题每一维度上的取值范围再细划分为 L 个相等的子区域,对于种群的每一个个体,采用轮盘赌算法选择一个子区域,在所选子区域内随机产生,使得种群中的个体能够均匀地分布在所有的子区域内,从而使得整个初始种群具有更好的多样性。具体流程如图 3 所示。

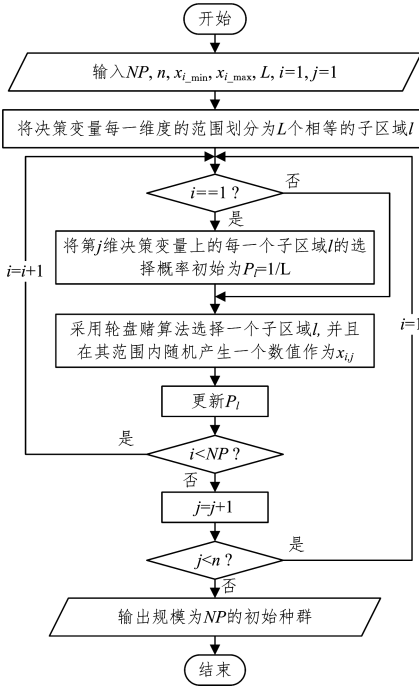


图 3 均匀种群初始化方法

Fig. 3 Uniform population initialization method

图 3 中, NP 表示种群规模, n 为求解问题的维度。 $x_{i,\min}$ 和 $x_{i,\max}$ 为求解问题在第 i 维上取值范围的最小值和最大值。 $\mathbf{X}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,n})$ 表示种群中的第 i 个个体, P_l 为子区域 l 的选择概率,其更新方法如式(12)所示:

$$P_l = \begin{cases} P_l - 1/NP, & \text{if subrange } l \text{ was selected} \\ P_l + 1/(NP * (L-1)), & \text{else} \end{cases} \quad (12)$$

4.3 基于双种群的选择方法

MODE-RMO 在选择操作时,通过对比 \mathbf{U}_i 和 \mathbf{X}_i 之间的支配关系,将适应度值较好的个体加入当前种群参与其他个体的进化操作,加快了算法的收敛速度。但是,较差个体的丢弃会造成种群多样性的降低,因此本文采用文献[11]中所提

出的基于双种群的选择方法。

该方法在种群的每一代进化过程中,同时设置当前执行进化操作的种群 $\mathbf{Pop1}$ 和一个辅助种群 $\mathbf{Pop2}$ 。若 \mathbf{U}_i 支配 \mathbf{X}_i , 则用 \mathbf{U}_i 替代 \mathbf{X}_i 存入 $\mathbf{Pop1}$, 同时将 \mathbf{X}_i 存入 $\mathbf{Pop2}$; 否则将 \mathbf{U}_i 存入 $\mathbf{Pop2}$ 。当所有个体完成进化后,再将 $\mathbf{Pop1}$ 和 $\mathbf{Pop2}$ 相结合,通过非支配关系和拥挤距离的比较,裁剪出前 NP 个较好的个体作为 $\mathbf{Pop1}$ 进入下一代进化操作。具体的执行过程如 4.4 节中的图 4 所示。

基于双种群的选择方法将 \mathbf{U}_i 和 \mathbf{X}_i 中较好的个体存入当前种群 $\mathbf{Pop1}$, 使得未参与变异的个体可以从包含已完成进化个体的 $\mathbf{Pop1}$ 中选择个体作为自身变异操作中的参与向量,加快了算法的收敛速度。通过增加种群 $\mathbf{Pop2}$ 来存储 \mathbf{U}_i 和 \mathbf{X}_i 中较差的个体,在每一代进化结束后进行裁剪操作,有助于种群多样性的保持。

4.4 模糊自适应排序变异多目标差分进化算法

本文所提出的 MODE-FARM 与 MODE-RMO 相比,采用了模糊自适应排序变异策略、均匀种群初始化以及基于双种群的选择方法。其实现步骤如下:

步骤 1 输入优化问题和相关参数,如种群规模 NP , 交叉概率 CR , 规模因子 F 在第一代的初始值以及最大值 F_{\max} 和最小值 F_{\min} , 平衡系数 γ 在第一代的初始值以及最大值 γ_{\max} 和最小值 γ_{\min} , 均匀种群初始化过程中的区域划分个数 L , 以及算法的终止条件。

步骤 2 采用均匀种群初始化方法,得到一个规模为 NP 的当前种群 $\mathbf{Pop1}$, 同时构建一个同样规模大小的空种群 $\mathbf{Pop2}$ 。计算当前种群 $\mathbf{Pop1}$ 中所有个体的适应度值。

步骤 3 从当前种群 $\mathbf{Pop1}$ 的第 i ($i=1$) 个个体开始,依次对种群中的所有个体 \mathbf{X}_i 进行如下操作:首先采用模糊自适应排序变异策略,依据式(11)得到变异向量 \mathbf{V}_i ; 然后依据式(7)将 \mathbf{V}_i 和 \mathbf{X}_i 进行交叉,得到实验向量 \mathbf{U}_i ; 再对 \mathbf{U}_i 进行越界处理,若不超过边界值,则继续,否则赋值 \mathbf{U}_i 为距离其最近的边界值;最后依据 4.3 节基于双种群的选择方法对 \mathbf{U}_i 和 \mathbf{X}_i 进行选择,并更新种群 $\mathbf{Pop1}$ 和 $\mathbf{Pop2}$ 。

步骤 4 此时完成了种群中所有个体的一次迭代,将 $\mathbf{Pop1}$ 和 $\mathbf{Pop2}$ 合并为一个规模为 $2NP$ 的种群并赋值给 $\mathbf{Pop1}$ 。依据自适应排序变异策略,通过将合并后种群中非支配解集的性能度量指标与前一代的相对变化值输入模糊系统,来求得下一代种群进化的规模因子 F 和平衡参数 γ 。

步骤 5 采用非支配排序算法和拥挤距离计算公式对合并后的 $\mathbf{Pop1}$ 进行裁剪操作,取前 NP 个较好的个体作为下一代优化的当前种群 $\mathbf{Pop1}$ 。

步骤 6 终止条件判断。若不满足终止条件,则返回步骤 3, 进入下一代优化; 否则, 输出 $\mathbf{Pop1}$ 作为最终所求问题的近似 Pareto 最优解集并结束程序运行。

MODE-FARM 的算法流程如图 4 所示。

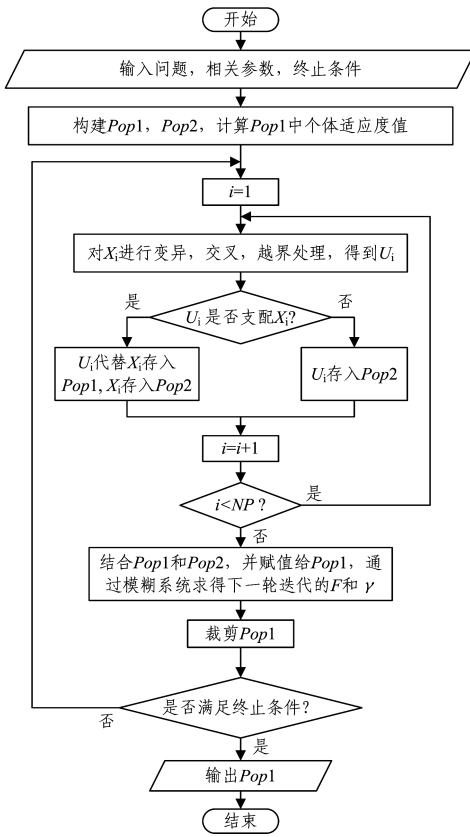


图4 MODE-FARM流程图

Fig. 4 Flowchart of MODE-FARM

5 算法复杂度分析

相比 MODE-RMO, MODE-FARM 只是在其基础上增加了少量的操作, 因此 MODE-FARM 仍保持了多目标差分进化算法的框架特征。由于采用了模糊自适应排序变异策略, MODE-FARM 在算法第一代优化前的准备阶段, 需要对种群 $Pop1$ 中的个体进行一次排序操作。但是从第二代开始直到算法结束, 由于在每一代算法末尾的裁剪部分已经对当前种群 $Pop1$ 进行了排序, 因此在下一代的变异操作中不需要再次进行排序操作。对于模糊系统中的计算操作、均匀种群初始化方法以及基于双种群的选择方法, 也只是增加了一些简单的计算量, 因此 MODE-FARM 和 MODE-RMO 的时间复杂度相同, 为 $O(T_{max} \cdot M \cdot NP^2)$ 。其中, T_{max} 是最大进化代数, M 是目标函数的个数, NP 是种群规模。

6 实验分析

为验证本文算法 MODE-FARM 在多目标优化问题上的求解性能, 选择 10 个多目标测试函数进行实验。其中, 5 个 ZDT 函数^[15] (ZDT1-ZDT4, ZDT6) 具有 2 个目标函数, 2 个 DTLZ 函数^[16] (DTLZ6-DTLZ7) 具有 3 个目标函数, 它们为标准多目标优化测试函数。此外, 本文还选取了 3 个具有偏好特征的 BT 函数^[17] (BT1, BT3, BT5), 属于优化难度较大的测试函数。本文实验环境为硬件配置为 Pentium CPU2.60 GHz 和 4 GB 内存的个人计算机, 采用 MATLAB2016a 编写

程序。

6.1 对比算法和参数设置

本文选取 MODE-RMO 和多目标进化算法开源平台 PlatEMO(A MATLAB Platform for Evolutionary Multi-Objective Optimization)^[18] 上的 MOEA/D, MOEA/D-DE, IM-MOEA(Inverse Modeling Multi-objective Evolutionary Algorithm) 作为实验对比算法。

为了实验对比的公平性, 所有算法的初始种群规模 NP 均设置为 100。采用最大函数评估次数作为算法的终止条件, ZDT 系列函数的最大函数评估次数设置为 25000, DTLZ6 和 DTLZ7 的最大函数评估次数分别设置为 50000 和 20000。BT 系列函数的最大函数评估次数设置为 200000。针对本文所提出的 MODE-FARM, F 和 γ 在第一代的初始值均设置为 0.5, $F_{max}=1, F_{min}=0.4, \gamma_{max}=1, \gamma_{min}=0, CR=0.3, L=5$ 。针对其他对比算法, 其参数值均依据原文献和平台系统默认值进行设置。

6.2 算法性能的度量指标

本文选取世代距离^[19] (Generational Distance, GD)、间隔指标^[20] (Spacing, SP), 以及反向世代距离^[19] (Inverted Generational Distance, IGD) 作为性能度量指标来评价解集的收敛性和多样性, 3 种指标均是数值越小表示解集的质量越好。

GD 用于评价算法的收敛性, 其定义如式 (13) 所示, P 是算法求得非支配解集的目标函数值, P^* 是真实 Pareto 前沿上的一组采样点, $dis(X, Z)$ 代表 X 和 Z 之间的欧几里得距离, $|P|$ 表示 P 的个数。

$$GD(P, P^*) = \frac{\sqrt{\sum_{Z \in P^*} \min_{X \in P} dis(X, Z)^2}}{|P|} \quad (13)$$

SP 反映解集的分布情况, 在收敛性相同的情况下, 分布均匀的解集具有更好的多样性, 其定义如式 (14) 所示:

$$SP = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad (14)$$

其中, $d_i = \min_j (\sum_{k=1}^M |f_k(x) - f_k(x)|)$, $i, j = 1, 2, \dots, n$, \bar{d} 是 d_i 的平均值, n 是所求解的个数。

IGD 能同时反映算法的收敛性和多样性, 其定义如式 (15) 所示, IGD 通过计算求得 P^* 中的每个点到 P 中离它最近点距离的平均值。

$$IGD(P, P^*) = \frac{\sum_{X \in P^*} \min_{Z \in P} dis(X, Z)}{|P^*|} \quad (15)$$

6.3 实验结果与分析

本文实验中的所有算法在每个测试函数上均独立运行 30 次, 统计平均值 (Mean) 和标准差 (Standard Deviation, SD)。表 1—表 3 分别为 GD, SP 和 IGD 的统计结果, 并将其中的最优结果值用加粗字体表示, 以便观察。此外, 本文采用 Wilcoxon 秩和检验, 从统计学角度比较了算法之间差异的显著性, 显著性水平为 0.05, 表中“+”“−”和“ \approx ”分别表示所提算法 MODE-FARM 明显优于、明显劣于和近似于其他对比算法。

从表 1 中可以看出,在反映算法收敛性的 GD 指标方面,与 MODE-RMO 和 IM-MOEA 相比,MODE-FARM 在 10 个函数上均明显较优;与 MOEA/D 相比,MODE-FARM 在 8 个函数上明显较优,在 2 个函数上明显较劣;与 MOEA/D-DE 相比,MODE-FARM 在 9 个函数上明显较优,在 1 个函数上明显较劣。

MODE-FARM 在函数 ZDT1,ZDT2,ZDT3,ZDT6,BT1,BT3 和 BT5 测试函数上的 GD 值均小于其他对比算法,说明 MODE-FARM 在这些问题上的收敛性好于其他对比算法。MOEA/D 和 MODE/D-DE 分别在 ZDT4,DTLZ7 和 DTLZ6 上获得了相对较小值,说明这两种算法在相应的

问题上具有更好的收敛性。

在表 2 中,相对于其他算法,MODE-FARM 在大多数函数上明显较优,并且 MODE-FARM 在 ZDT1~ZDT4,DTLZ6,DTLZ7 和 BT5 上取得了最优 SP 值,说明其在这些函数上所求得解集具有更好的分布性。

在表 3 中,与 MODE-RMO 和 MOEA/D 相比,MODE-FARM 在 10 个函数上均明显较优;与 MOEA/D-DE 和 IM-MOEA 相比,MODE-FARM 在 9 个函数上明显较优,在 1 个函数上明显较劣。除了 ZDT4 和 ZDT6 外,MODE-FARM 在剩余 8 个测试函数上的 IGD 值都达到了最优,说明其在这些函数上具有更好的收敛性和多样性。

表 1 MODE-FARM 和其他 4 种算法在 GD 上的实验结果

Table 1 Results of MODE-FARM and other four algorithms on GD

Problems	MODE-FARM	MOED-RMO	MOEA/D	MOEA/D-DE	IM-MOEA
ZDT1	2.39×10^{-4} (3.64×10^{-5})	3.85×10^{-3} (3.68×10^{-4}) +	5.53×10^{-4} (1.64×10^{-4}) +	3.80×10^{-3} (2.51×10^{-3}) +	1.55×10^{-1} (1.35×10^{-1}) +
ZDT2	9.43×10^{-5} (5.26×10^{-6})	7.07×10^{-3} (8.85×10^{-4}) +	8.40×10^{-4} (4.92×10^{-4}) +	2.75×10^{-3} (1.41×10^{-3}) +	2.24×10^{-1} (1.83×10^{-1}) +
ZDT3	1.58×10^{-4} (1.26×10^{-5})	4.97×10^{-3} (6.74×10^{-4}) +	2.47×10^{-3} (3.04×10^{-3}) +	7.59×10^{-3} (6.51×10^{-3}) +	1.30×10^{-1} (8.23×10^{-2}) +
ZDT4	6.12×10^{-3} (1.28×10^{-2})	2.87×10^{-1} (3.57×10^{-1}) +	3.89×10^{-3} (3.07×10^{-3}) -	1.06×10^{-1} (1.32×10^{-1}) +	4.87×10^{-2} (7.64×10^{-2}) +
ZDT6	7.07×10^{-5} (4.61×10^{-6})	2.85×10^{-2} (1.24×10^{-2}) +	1.18×10^{-3} (4.71×10^{-4}) +	1.23×10^{-2} (2.18×10^{-2}) +	7.63×10^{-1} (1.03×10^{-1}) +
DTLZ6	9.63×10^{-5} (6.19×10^{-6})	4.39×10^{-1} (3.38×10^{-2}) +	1.55×10^{-3} (4.02×10^{-3}) +	8.33×10^{-5} (2.01×10^{-6}) -	6.27×10^{-1} (1.67×10^{-2}) +
DTLZ7	5.57×10^{-3} (4.13×10^{-4})	2.18×10^{-2} (5.21×10^{-3}) +	4.83×10^{-3} (1.01×10^{-3}) -	6.35×10^{-2} (5.90×10^{-2}) +	6.88×10^{-2} (4.14×10^{-2}) +
BT1	2.61×10^{-3} (3.84×10^{-3})	7.45×10^{-2} (1.96×10^{-2}) +	1.14×10^{-2} (6.26×10^{-3}) +	1.13×10^{-1} (2.53×10^{-2}) +	3.86×10^{-1} (1.45×10^{-2}) +
BT3	1.01×10^{-3} (3.39×10^{-4})	2.01×10^{-2} (9.58×10^{-3}) +	2.47×10^{-3} (9.93×10^{-4}) +	5.11×10^{-2} (1.27×10^{-2}) +	3.21×10^{-1} (1.15×10^{-2}) +
BT5	1.71×10^{-3} (2.19×10^{-3})	6.17×10^{-2} (1.86×10^{-2}) +	2.96×10^{-2} (1.43×10^{-2}) +	1.11×10^{-1} (3.11×10^{-2}) +	4.09×10^{-1} (2.19×10^{-2}) +
+/≈/-		10/0/0	8/0/2	9/0/1	10/0/0

表 2 MODE-FARM 和其他 4 种算法在 SP 上的实验结果

Table 2 Results of MODE-FARM and other four algorithms on SP

Problems	MODE-FARM	MOED-RMO	MOEA/D	MOEA/D-DE	IM-MOEA
ZDT1	6.33×10^{-3} (6.06×10^{-4})	7.16×10^{-3} (2.62×10^{-3}) ≈	9.34×10^{-3} (4.95×10^{-3}) +	1.17×10^{-2} (1.75×10^{-3}) +	5.62×10^{-1} (6.69×10^{-1}) +
ZDT2	6.55×10^{-3} (4.14×10^{-4})	9.32×10^{-3} (1.75×10^{-3}) +	1.18×10^{-2} (1.82×10^{-2}) ≈	9.42×10^{-3} (4.28×10^{-3}) ≈	5.97×10^{-1} (6.08×10^{-1}) +
ZDT3	6.95×10^{-3} (7.86×10^{-4})	1.13×10^{-2} (1.75×10^{-3}) +	2.18×10^{-2} (4.74×10^{-3}) +	2.66×10^{-2} (6.37×10^{-3}) +	4.68×10^{-1} (5.41×10^{-1}) +
ZDT4	6.71×10^{-3} (1.35×10^{-3})	6.46×10^{-1} (9.41×10^{-1}) +	1.43×10^{-2} (8.15×10^{-3}) +	2.01×10^{-1} (6.36×10^{-1}) +	4.00×10^{-1} (6.13×10^{-1}) +
ZDT6	8.56×10^{-3} (8.17×10^{-4})	4.97×10^{-2} (1.19×10^{-1}) +	4.40×10^{-3} (1.03×10^{-3}) -	1.24×10^{-1} (2.17×10^{-1}) ≈	1.32×10^{-1} (6.54×10^{-2}) +
DTLZ6	1.21×10^{-2} (1.36×10^{-3})	2.85×10^{-1} (3.24×10^{-2}) +	8.52×10^{-2} (1.72×10^{-2}) +	5.84×10^{-2} (1.21×10^{-2}) +	4.35×10^{-1} (3.93×10^{-2}) +
DTLZ7	6.94×10^{-2} (9.27×10^{-3})	8.48×10^{-2} (2.21×10^{-2}) ≈	1.72×10^{-1} (5.11×10^{-2}) +	6.51×10^{-1} (4.92×10^{-1}) +	3.53×10^{-1} (1.10×10^{-1}) +
BT1	7.52×10^{-3} (2.55×10^{-3})	6.92×10^{-3} (6.42×10^{-4}) ≈	1.16×10^{-2} (1.15×10^{-2}) +	1.02×10^{-1} (4.95×10^{-2}) +	5.94×10^{-2} (3.30×10^{-2}) +
BT3	6.32×10^{-3} (2.46×10^{-3})	2.51×10^{-3} (5.52×10^{-4}) -	6.00×10^{-3} (4.94×10^{-3}) +	3.16×10^{-1} (6.32×10^{-2}) +	5.09×10^{-2} (1.46×10^{-2}) +
BT5	5.35×10^{-3} (9.88×10^{-4})	5.61×10^{-3} (5.07×10^{-4}) +	2.47×10^{-2} (1.32×10^{-2}) +	1.31×10^{-1} (7.23×10^{-2}) +	5.75×10^{-2} (2.22×10^{-2}) +
+/≈/-		6/3/1	8/1/1	8/2/0	10/0/0

表 3 MODE-FARM 和其他 4 种算法在 IGD 上的实验结果

Table 3 Results of MODE-FARM and other four algorithms on IGD

Problems	MODE-FARM	MOED-RMO	MOEA/D	MOEA/D-DE	IM-MOEA
ZDT1	4.36×10^{-3} (1.29×10^{-4})	3.67×10^{-2} (2.76×10^{-3}) +	1.84×10^{-2} (1.56×10^{-2}) +	3.66×10^{-2} (2.25×10^{-2}) +	1.78×10^{-1} (1.12×10^{-2}) +
ZDT2	4.50×10^{-3} (1.51×10^{-4})	6.71×10^{-2} (7.07×10^{-3}) +	7.35×10^{-2} (9.29×10^{-2}) +	3.33×10^{-2} (4.93×10^{-2}) +	2.89×10^{-1} (1.91×10^{-2}) +
ZDT3	4.93×10^{-3} (1.49×10^{-4})	5.09×10^{-2} (5.11×10^{-3}) +	3.13×10^{-2} (1.59×10^{-2}) +	7.65×10^{-2} (4.40×10^{-2}) +	1.69×10^{-1} (9.13×10^{-3}) +
ZDT4	6.08×10^{-2} (1.22×10^{-1})	2.10×10^{-1} (2.36×10^{-1}) +	3.90×10^{-2} (2.77×10^{-2}) +	4.38×10^{-1} (2.32×10^{-1}) +	6.23×10^{-3} (2.32×10^{-4}) -
ZDT6	5.41×10^{-3} (1.88×10^{-4})	2.28×10^{-1} (1.93×10^{-2}) +	9.89×10^{-3} (3.05×10^{-3}) +	3.11×10^{-3} (2.28×10^{-5}) -	2.19 (1.41×10^{-1}) +
DTLZ6	6.79×10^{-3} (3.63×10^{-4})	3.93 (3.00×10^{-1}) +	3.38×10^{-2} (5.95×10^{-6}) +	1.45×10^{-2} (2.70×10^{-5}) +	4.46 (1.71×10^{-1}) +
DTLZ7	7.42×10^{-2} (3.72×10^{-3})	1.76×10^{-1} (1.29×10^{-2}) +	2.38×10^{-1} (2.30×10^{-1}) +	2.74×10^{-1} (1.33×10^{-1}) +	3.56×10^{-1} (4.45×10^{-2}) +
BT1	2.29×10^{-2} (3.52×10^{-2})	3.39×10^{-2} (8.99×10^{-3}) +	9.92×10^{-2} (5.78×10^{-2}) +	8.55×10^{-1} (1.74×10^{-1}) +	2.78 (1.05×10^{-1}) +
BT3	1.33×10^{-2} (4.74×10^{-3})	2.06×10^{-2} (3.17×10^{-3}) +	2.67×10^{-2} (1.01×10^{-2}) +	1.26×10^{-1} (5.02×10^{-2}) +	2.20 (1.17×10^{-1}) +
BT5	1.48×10^{-2} (2.05×10^{-2})	2.80×10^{-2} (8.27×10^{-3}) +	2.60×10^{-1} (1.10×10^{-1}) +	7.16×10^{-1} (1.73×10^{-1}) +	2.79 (1.08×10^{-1}) +
	+ / ≈ / -	10/0/0	10/0/0	9/0/1	9/0/1

总体来讲,对于本文所给出的 10 个多目标测试函数, MODE-FARM 在收敛性和多样性方面都好于其他对比算法,这说明本文所提算法 MODE-FARM 中的一系列策略可以有效地提高 MODE-RMO 的寻优性能。

为了更加直观地了解 MODE-FARM 的性能,本文绘制出算法运行一次获得的非支配解集。图 5—图 14 为本文所有算法在 ZDT,DTLZ 和 BT 系列测试函数上某次运行的 Pareto 前沿分布图(其中 True PF 是相应优化问题的真实 Pareto 前沿)。

在图 5—图 14 中,整体上来看,相对于其他算法,MODE-FARM 的 Pareto 前沿能更好地逼近真实的 Pareto 前沿。

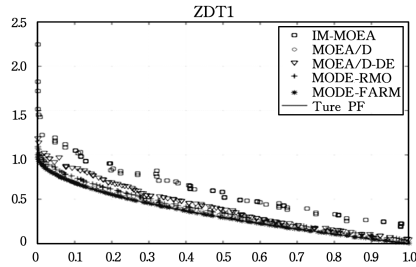


图 5 ZDT1 上非支配解集的 Pareto 前沿

Fig. 5 Pareto front of non-dominated solutions on ZDT1

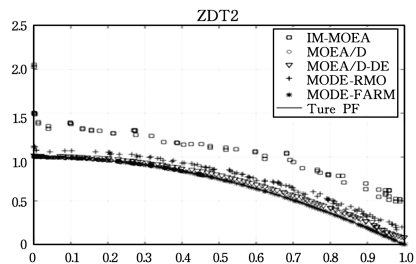


图 6 ZDT2 上非支配解集的 Pareto 前沿

Fig. 6 Pareto front of non-dominated solutions on ZDT2

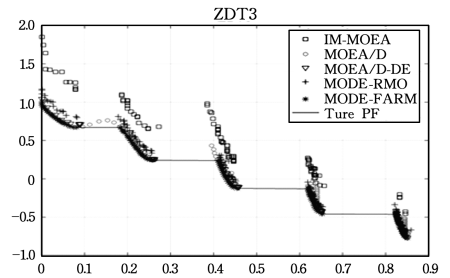


图 7 ZDT3 上非支配解集的 Pareto 前沿

Fig. 7 Pareto front of non-dominated solutions on ZDT3

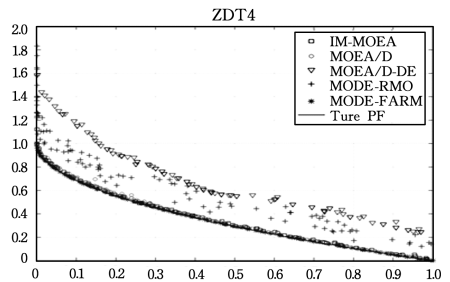


图 8 ZDT4 上非支配解集的 Pareto 前沿

Fig. 8 Pareto front of non-dominated solutions on ZDT4

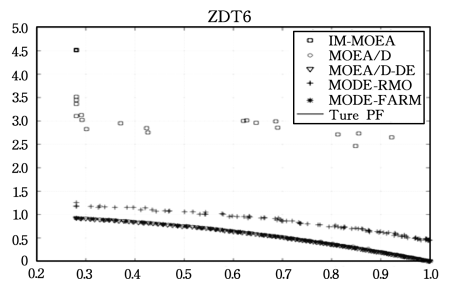


图 9 ZDT6 上非支配解集的 Pareto 前沿

Fig. 9 Pareto front of non-dominated solutions on ZDT6

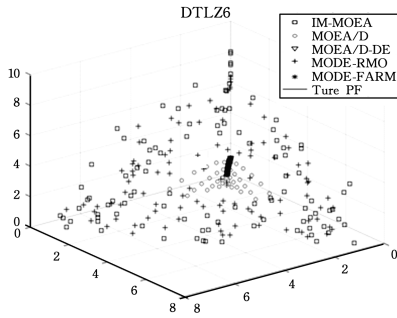


图 10 DTLZ6 上非支配解集的 Pareto 前沿

Fig. 10 Pareto front of non-dominated solutions on DTLZ6

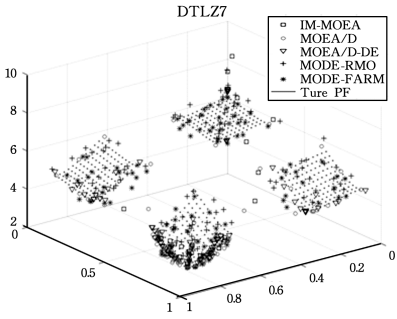


图 11 DTLZ7 上非支配解集的 Pareto 前沿

Fig. 11 Pareto front of non-dominated solutions on DTLZ7

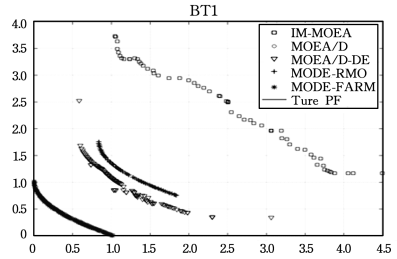


图 12 BT1 上非支配解集的 Pareto 前沿

Fig. 12 Pareto front of non-dominated solutions on BT1

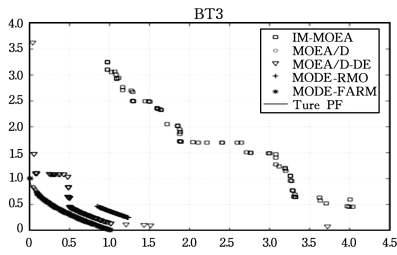


图 13 BT3 上非支配解集的 Pareto 前沿

Fig. 13 Pareto front of non-dominated solutions on BT3

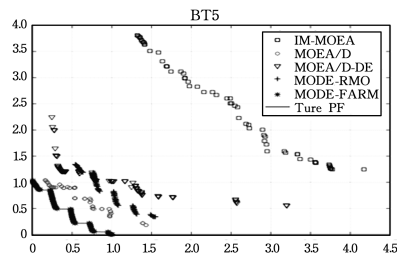


图 14 BT5 上非支配解集的 Pareto 前沿

Fig. 14 Pareto front of non-dominated solutions on BT5

6.4 模糊自适应排序变异策略验证

为了进一步验证模糊自适应排序变异策略在提高算法寻优性能方面的有效性,本文将所提出的算法 MODE-FARM 中的模糊自适应排序变异策略替换为传统的变异策略,保留均匀种群初始化和基于双种群的选择方法,记为 MODE-FARM1。通过将 MODE-FARM 和 MODE-FARM1 在 ZDT, DTLZ 和 BT 系列测试函数上进行 30 次的独立实验,得到表 4—表 6 的实验结果。

表 4 MODE-FARM 和 MODE-FARM1 在 GD 上的实验结果

Table 4 Experimental results of MODE-FARM and MODE-FARM1 on GD

Problems	MODE-FARM		MODE-FARM1	
	Mean	SD	Mean	SD
ZDT1	2.39×10^{-4}	3.64×10^{-5}	2.85×10^{-3}	4.14×10^{-4}
ZDT2	9.43×10^{-5}	5.26×10^{-6}	3.60×10^{-3}	3.22×10^{-4}
ZDT3	1.58×10^{-4}	1.26×10^{-5}	3.03×10^{-3}	9.23×10^{-4}
ZDT4	6.12×10^{-3}	1.28×10^{-2}	3.54×10^{-3}	5.56×10^{-3}
ZDT6	7.07×10^{-5}	4.61×10^{-6}	1.09×10^{-2}	8.68×10^{-4}
DTLZ6	9.63×10^{-5}	6.19×10^{-6}	4.22×10^{-1}	2.92×10^{-2}
DTLZ7	5.57×10^{-3}	4.13×10^{-4}	2.13×10^{-2}	9.54×10^{-3}
BT1	2.61×10^{-3}	3.84×10^{-3}	8.43×10^{-2}	1.36×10^{-2}
BT3	1.01×10^{-3}	3.39×10^{-4}	3.39×10^{-2}	9.08×10^{-3}
BT5	1.71×10^{-3}	2.19×10^{-3}	7.54×10^{-2}	1.42×10^{-2}

表 5 MODE-FARM 和 MODE-FARM1 在 SP 上的实验结果

Table 5 Experimental results of MODE-FARM and MODE-FARM1 on SP

Problems	MODE-FARM		MODE-FARM1	
	Mean	SD	Mean	SD
ZDT1	6.33×10^{-3}	6.06×10^{-4}	7.54×10^{-3}	4.08×10^{-3}
ZDT2	6.55×10^{-3}	4.14×10^{-4}	7.88×10^{-3}	2.92×10^{-3}
ZDT3	6.95×10^{-3}	7.86×10^{-4}	1.01×10^{-2}	4.84×10^{-3}
ZDT4	6.71×10^{-3}	1.35×10^{-3}	6.51×10^{-3}	6.49×10^{-4}
ZDT6	8.56×10^{-3}	8.17×10^{-4}	6.65×10^{-3}	6.38×10^{-4}
DTLZ6	1.21×10^{-2}	1.36×10^{-3}	2.78×10^{-1}	3.46×10^{-2}
DTLZ7	6.94×10^{-2}	9.27×10^{-3}	8.13×10^{-2}	2.19×10^{-2}
BT1	7.52×10^{-3}	2.55×10^{-3}	9.36×10^{-3}	9.58×10^{-4}
BT3	6.32×10^{-3}	2.46×10^{-3}	4.67×10^{-3}	3.86×10^{-3}
BT5	5.35×10^{-3}	9.88×10^{-4}	6.86×10^{-3}	6.33×10^{-4}

表 6 MODE-FARM 和 MODE-FARM1 在 IGD 上的实验结果

Table 6 Experimental results of MODE-FARM and MODE-FARM1 on IGD

Problems	MODE-FARM		MODE-FARM1	
	Mean	SD	Mean	SD
ZDT1	4.36×10^{-3}	1.29×10^{-4}	2.69×10^{-2}	2.24×10^{-3}
ZDT2	4.50×10^{-3}	1.51×10^{-4}	3.58×10^{-2}	2.53×10^{-3}
ZDT3	4.93×10^{-3}	1.49×10^{-4}	3.43×10^{-2}	2.64×10^{-3}
ZDT4	6.08×10^{-2}	1.22×10^{-1}	3.67×10^{-2}	5.44×10^{-2}
ZDT6	5.41×10^{-3}	1.88×10^{-4}	1.08×10^{-1}	8.87×10^{-3}
DTLZ6	6.79×10^{-3}	3.63×10^{-4}	3.78	2.57×10^{-1}
DTLZ7	7.42×10^{-2}	3.72×10^{-3}	1.72×10^{-1}	1.37×10^{-2}
BT1	2.29×10^{-2}	3.52×10^{-2}	8.66×10^{-1}	1.42×10^{-1}
BT3	1.33×10^{-2}	4.74×10^{-3}	5.11×10^{-1}	9.94×10^{-2}
BT5	1.48×10^{-2}	2.05×10^{-2}	7.71×10^{-1}	1.41×10^{-1}

对于表 4 中 GD 指标的实验结果,相比 MODE-FARM1, MODE-FARM 除了在 ZDT4 上的均值和标准差略差以外,在其他函数上的均值和标准差均明显好于 MODE-FARM1。在表 5 中,MODE-FARM 在大多测试函数上 SP 指标的均值和标准差要好于 MODE-FARM1。在表 6 中的 IGD 指标上,除

ZDT4 以外,MODE-FARM 在剩余其他所有测试函数上均明显好于 MODE-FARM1。因此,整体来说,MODE-FARM 在 10 个测试函数上的收敛性和多样性要优于 MODE-FARM1,从而验证了本文所提出的模糊自适应排序变异策略的有效性。

结束语 本文在 MODE-RMO 的基础上,提出了一种模糊自适应排序变异多目标差分进化算法(MODE-FARM)。通过将模糊系统反馈策略融入排序变异操作,提出一种新的模糊自适应排序变异策略,利用模糊系统自适应调节排序变异操作中的参数值,有效地提高了算法的收敛性和多样性。引入均匀种群初始化和基于双种群的选择方法,进一步提高了算法的寻优性能。实验结果表明,MODE-FARM 在收敛性和多样性方面整体上优于其他对比算法。并且通过进一步的实验验证了模糊自适应排序变异策略对算法性能改进的有效性。由于本文只是针对无约束条件的多目标测试函数进行实验,对于带有约束条件多目标优化问题的研究,以及算法在工程优化中的应用,是今后工作的主要内容。

致谢 衷心感谢安徽大学 BIMK 团队为本研究提供的帮助!

参 考 文 献

- [1] GONG M G, JIAO L C, YANG D D, et al. Research on evolutionary multi-objective optimization algorithm [J]. *Journal of Software*, 2009, 20(2): 271-289. (in Chinese)
公茂果, 焦李成, 杨咚咚, 等. 进化多目标优化算法研究[J]. *软件学报*, 2009, 20(2): 271-289.
- [2] SUN C F, ZHOU H Y, ZHANG Y H. Dynamic environment economic dispatch based on differential evolution algorithm[J]. *Computer Science*, 2012, 39(11): 208-211. (in Chinese)
孙成富, 周海岩, 张亚红. 基于差分进化算法的动态环境经济电力系统调度优化[J]. *计算机科学*, 2012, 39(11): 208-211.
- [3] SONG X Y, ZHU J Y, SUN H L. Hybrid differential evolution algorithm for vehicle routing problem with time windows[J]. *Computer Science*, 2014, 41(12): 220-225. (in Chinese)
宋晓宇, 朱加园, 孙焕良. 一种求解带时间窗车辆路径问题的混合差分进化算法[J]. *计算机科学*, 2014, 41(12): 220-225.
- [4] SUN Z, WU J, YANG J, et al. Path planning for GEO-UAV bi-static SAR using constrained adaptive multiobjective differential evolution [J]. *IEEE Transactions on Geoscience & Remote Sensing*, 2016, 54(11): 6444-6457.
- [5] ZHANG Q F, LI H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition [J]. *IEEE Transactions on Evolutionary Computation*, 2007, 11(6): 712-731.
- [6] LI H, ZHANG Q F. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II [J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(2): 284-302.
- [7] SONG T, ZHUANG Y. A kind of multi-objective optimization algorithm based on differential evolution with multi-population mechanism [J]. *Computer Science*, 2012, 39(8): 205-209. (in Chinese)
宋通, 庄毅. 基于多种群差分进化的多目标优化算法[J]. *计算机科学*, 2012, 39(8): 205-209.
- [8] XIE C W, LI K, LIAO G Y. Improved NSGA2 algorithm with differential evolution local search [J]. *Computer Science*, 2013, 40(10): 235-238. (in Chinese)
谢承旺, 李凯, 廖国勇. 一种带差分局部搜索的改进型 NSGA2 算法[J]. *计算机科学*, 2013, 40(10): 235-238.
- [9] JARIYATANTIWAIT C, YEN G G. Fuzzy multiobjective differential evolution using performance metrics feedback [C] // *IEEE Congress on Evolutionary Computation*. Beijing, China: IEEE Press, 2014: 1959-1966.
- [10] WANG X, TANG L. An adaptive multi-population differential evolution algorithm for continuous multi-objective optimization [J]. *Information Sciences*, 2016, 348(2): 124-141.
- [11] ALI M, SIARRY P, PANT M. An efficient differential evolution based algorithm for solving multi-objective optimization problems [J]. *European Journal of Operational Research*, 2011, 217(2): 404-416.
- [12] GONG W Y, CAI Z H. Differential evolution with ranking-based mutation operators [J]. *IEEE Transactions on Cybernetics*, 2013, 43(6): 2066-2081.
- [13] CHEN X, DU W L, QIAN F. Multi-objective differential evolution with ranking-based mutation operator and its application in chemical process optimization [J]. *Chemometrics & Intelligent Laboratory Systems*, 2014, 136(16): 85-96.
- [14] ROBITC T, FILIPIC B. DEMO: differential evolution for multiobjective optimization [C] // *International Conference on Evolutionary Multi-Criterion Optimization*. Berlin Heidelberg: Springer-Verlag Press, 2005: 520-533.
- [15] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-197.
- [16] TIAN Y, CHENG R, ZHANG X Y, et al. An indicator based multiobjective evolutionary algorithm with reference point adaptation for better versatility [J]. *IEEE Transactions on Evolutionary Computation*, 2018, 4(22): 609-622.
- [17] LI H, ZHANG Q F, DENG J. Biased multiobjective optimization and decomposition algorithm [J]. *IEEE Transactions on Cybernetics*, 2016, 47(1): 52-66.
- [18] TIAN Y, CHENG R, ZHANG X Y, et al. PlatEMO: a MATLAB platform for evolutionary multi-objective optimization [Educational Forum [J]. *IEEE Computational Intelligence Magazine*, 2017, 12(4): 73-87.
- [19] TIAN Y, ZHANG X Y, CHENG R, et al. A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric [C] // *Proceedings of the 2016 IEEE Congress on Evolutionary Computation*. Vancouver, BC, Canada: IEEE Press, 2016: 5222-5229.
- [20] COELLO C A C, PULIDO G T, LECHUGA M S. Handling multiple objectives with particle swarm optimization [J]. *IEEE Transactions on Evolutionary Computation*, 2004, 8(3): 256-279.