

doi:10.16055/j.issn.1672-058X.2022.0001.009

基于双决策和快速分层的多目标粒子群算法

王世华¹, 李娜娜², 刘衍民³

(1. 贵州大学 数学与统计学院, 贵阳 550025; 2. 贵州民族大学 数据科学与信息工程学院, 贵阳 550025;

3. 遵义师范学院 数学学院, 贵州 遵义 563006)

摘要:针对粒子群算法在求解多目标优化问题时存在的收敛性不足和多样性缺失等问题,提出一种基于双决策和快速分层的新型多目标粒子群算法(DDFSMOPSO);在该算法中,采用外部存档对迭代产生的非劣解进行存储,并利用拥挤距离和绝对距离相结合的双决策策略对外部存档规模进行维护,使得优秀粒子在随后的进化过程中易于保留和发展;同时,采用快速分层策略从外部存档中选取全局学习样本,用于领导种群中粒子的进化,促使种群中的粒子向真实的 Pareto 前沿移动;将 DDFSOPSO 算法和 3 种经典的多目标粒子群算法在 ZDT 和 DTLZ 系列的部分测试函数上进行仿真实验;实验结果表明:相比其他几种经典算法,DDFSOPSO 算法表现出较好的收敛性和多样性,因此,DDFSOPSO 算法可以作为求解多目标优化问题的有效算法。

关键词:多目标优化;粒子群算法;双决策;快速分层

中图分类号:TP18

文献标志码:A

文章编号:1672-058X(2022)01-0062-09

0 引言

自然科学和社会科学等多个领域中的优化问题几乎都是多目标优化问题(MOPs)^[1]。由于多目标优化问题中任何两个目标或者多个目标之间是相互制约的,所以它的求解方法与单目标优化问题的求解策略不同。通常是利用传统的加权、混合等方法将多目标优化问题进行转化,然后采用单目标优化问题中比较成熟的求解方法对转化后的多目标优化问题进行求解。但是,随着多目标优化问题中待优化目标的不断增多,传统的优化方法不再适用于解决这类复杂的优化问题,因此,学者们提出启发式算

法来解决一些比较复杂的多目标优化问题。粒子群算法(PSO)^[2]是一种经典的基于种群的启发式算法,因其具有收敛速度快和易于实现等特点,在单目标优化问题中得到了广泛应用。一些研究表明,改进的 PSO 适合求解 MOPs,如 Coello 等^[3]提出的多目标粒子群算法(MOPSO),该算法采用外部存档对迭代产生的非劣解进行存储,并选择这些非劣解来引导其他粒子飞行。MOPSO 为 PSO 求解多目标优化问题提供了思路和方法,但由于 MOPSO 在求解一些含有多个目标的优化问题时容易陷入局部最优并且多样性差,许多学者对其进行了改进^[4-7]。Li 等^[4]将粒子的排序优势和切比雪夫距离集成到 MOPSO 中,增加最优粒子的选择限制,促使粒子向

收稿日期:2020-12-08;修回日期:2021-02-05.

基金项目:国家自然科学基金(71461027);贵州省科技创新人才团队(黔科合平台人才[2016]5619).

作者简介:王世华(1997—),男,贵州毕节人,硕士研究生,从事优化理论及智能算法研究.

通讯作者:刘衍民(1978—),男,山东临沂人,教授,博士生导师,从事优化理论、智能算法领域的研究. Email:yanmin7813

@163.com.

真实的 Pareto 前沿移动,增强了算法的收敛性,但该方法在解决多峰问题时并没有表现出较大优势;Mostaghim 等^[5]采用 sigma 法来选取全局最优粒子,该方法的使用虽然提升了种群的多样性,但是选出的粒子通常分布在非劣解比较集中的区域,在求解一些多目标优化问题时容易陷入局部最优;Li 等^[6]利用目标函数加权法自适应地选取全局最优粒子,并采用变异来防止算法陷入局部最优,有效地提升了算法的收敛性和多样性,但在解决高维目标问题上还有一定的不足;Leng 等^[7]利用网格距离法对外部存档的维护策略和全局最优粒子的选取方式进行改进,使得算法的收敛性和多样性有了一定程度的提升,但增加了计算的复杂度。

上述各种策略从种群的多样性和非劣解的收敛性等方面对 MOPSO 进行改进和优化,虽然取得了较好的研究成果,但是这些策略只对 MOPSO 的某一性能进行了改进,在其他性能指标上并没有表现出较大的优势。例如文献[5]中,多样性方面得到了很大程度的提升,但是其收敛性方面还存在很大的不足。因此,本文提出一种基于双决策和快速分层的多目标粒子群算法(DDFSMOPSO)来进一步提升种群的多样性和非劣解的收敛性。DDFSMOPSO 算法加入了拥挤距离和绝对距离,对 MOPSO 的外部存档规模的维护策略进行了改进,有效地提升了种群的多样性;与此同时,对全局学习样本的选取方式进行了优化,提出了快速分层策略,增强了非劣解的收敛性。

1 相关概念

1.1 多目标优化问题

有 n 个决策变量, m 个目标的多目标优化问题表示如下:

$$\begin{aligned} \min y = F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \\ \text{s. t. } \begin{cases} g_i(x) \geq 0, i=1, 2, \dots, p \\ h_j(x) = 0, j=1, 2, \dots, q \end{cases} \end{aligned}$$

其中, $x = (x_1, x_2, \dots, x_n) \in X \subset R^n$ 为 n 维决策变量; $y = (y_1, y_2, \dots, y_m) \in Y \subset R^m$ 为 m 维目标变量; $g_i(x) \geq 0 (i=1, 2, \dots, p)$ 和 $h_j(x) = 0 (j=1, 2, \dots, q)$ 为约束条件。

1.2 粒子群算法

粒子群算法(PSO)是通过选择适当的粒子来引导种群对优化空间进行搜索,从而获得最优解的一种启发式优化算法,因其概念简单、易于实现等特点,在诸多领域得到了广泛应用。由于 PSO 在求解各领域的单目标优化问题时取得了良好的效果,基于此,学者们将 PSO 扩展到多目标优化领域,提出了基于外部存档思想和 Pareto 占优原理的多目标粒子群算法(MOPSO)。在每次迭代过程中,不论是单目标粒子群算法(PSO)还是多目标粒子群算法(MOPSO),粒子 i 都遵循式(1)和式(2)来更新其速度和位置。

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1(p_i(t) - x_i(t)) + c_2 \cdot r_2(g(t) - x_i(t)) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

其中, c_1, c_2 为学习因子; r_1, r_2 为 $[0, 1]$ 区间上的随机数; w 为惯性权重; $p_i(t)$ 为粒子的个体学习样本,它表示在迭代的 t 时刻,粒子 i 所经历过的最优位置; $g(t)$ 为粒子的全局学习样本,它表示在迭代的 t 时刻种群中最优粒子的位置。

2 基于双决策和快速分层的改进算法(DDFSMOPSO)介绍

2.1 外部存档的维护

多目标粒子群算法一般采用外部存档对每一次迭代产生的非劣解进行存储,随着迭代的进行,外部存档中的非劣解逐渐增多,当外部存档规模超出预设的最大规模时,将超出规模的那一部分粒子进行有效删除,不但可以提高算法的搜索能力,而且还能维护种群的多样性。因此,这里采用拥挤距离(Crowding Distance, CD)和绝对距离(Absolute Distance, AD)相结合的双决策(Double Decision, DD)策略来维护外部存档规模。本文将粒子与其相邻粒子欧式距离之和作为其拥挤距离,将粒子与其相邻粒子的欧氏距离之差的绝对值作为其绝对距离,式(3)和式(4)给出了拥挤距离和绝对距离的计算公式。

$$D_c(X_i) = \begin{cases} d(X_i, X_{i-1}) + d(X_i, X_{i+1}), 1 < i < n \\ \infty, i=1 \\ \infty, i=n \end{cases} \quad (3)$$

$$D_A(X_i) = |d(X_i, X_{i-1}) - d(X_i, X_{i+1})| \quad (4)$$

其中: n 为外部存档中粒子的数量; $d(X_i, X_{i-1})$ 为粒子 X_i 到粒子 X_{i-1} 的欧式距离; $d(X_i, X_{i+1})$ 为粒子 X_i 到粒子 X_{i+1} 的欧式距离; $D_C(X_i)$ 为粒子 X_i 的拥挤距离; $D_A(X_i)$ 为粒子 X_i 的绝对距离。

随着迭代的进行, 当外部存档规模超出预设的最大规模时, 双决策策略被启用。其具体流程如下: 首先根据式(3)计算外部存档中所有粒子的拥挤距离, 然后根据拥挤距离的值从小到大删除超出存档规模的那部分粒子, 若存在 $b(b > 1)$ 个粒子的拥挤距离相等, 则运用式(4)计算这些粒子的绝对距离, 并将绝对距离最大的粒子进行删除。相比于文献[3]用到的随机法、文献[8]用到的拥挤距离法, 双决策策略的优点是可以使外部存档中粒子的分布更均匀。下面通过一个实例来进行说明, 如图 1 所示, 以外部存档中非劣解分布最密集的一块区域为例, 图 1(a) 中一共有 7 个粒子, 假如要删除其中的一个, 图 1(b) 是采用拥挤距离法删除过程示意图(删除粒子 B); 图 1(c) 是采用随机法删除过程示意图(删除粒子 C); 图 1(d) 是采用拥挤距离和绝对距离相结合的双决策策略删除过程示意图(删除粒子 A)。通过图 1(b), 1(c) 和 1(d) 的对比, 可以直观地看出: 采用拥挤距离和绝对距离相结合的双决策策略对外部存档规模进行维护, 使得存档中粒子的分布更均匀。

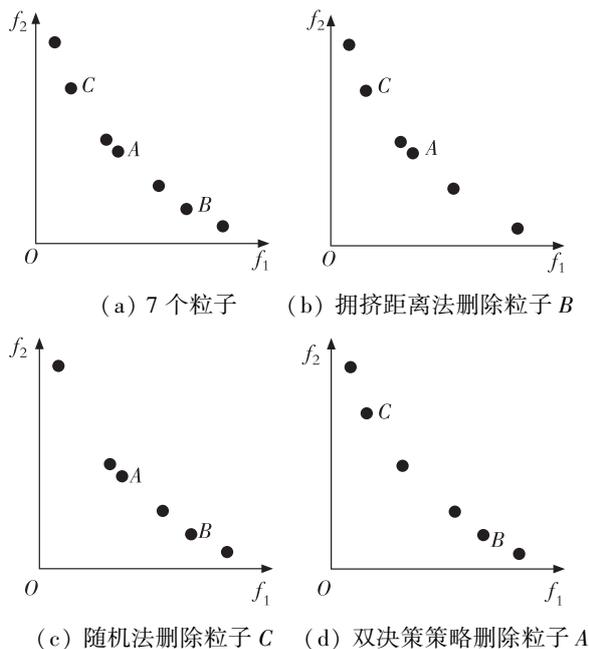


图 1 外部存档的维护示意图

Fig. 1 Maintenance diagram of external archive

2.2 全局学习样本的选取

在多目标粒子群算法中, 全局学习样本的优劣是决定算法是否有效的关键因素, 因此, 选取一个较优的全局学习样本至关重要。通常采用文献[3]中的轮盘赌策略选取全局学习样本, 该策略操作简单, 易实现, 但不利于保持算法的收敛性。鉴于此, 这里采用快速分层(Fast Stratification, FS)策略对全局学习样本进行选取。本文将粒子的标准化目标值作为其快速分层值, 于是粒子 X_i 的快速分层值为

$$S_F(X_i) = \sum_{j=1}^M \left(\frac{f_j(X_i) - f_j^{\min}}{f_j^{\max} - f_j^{\min}} \right) \quad (5)$$

其中, M 为目标个数; $f_j(X_i)$ 为粒子 X_i 的第 j 个目标函数值; f_j^{\min} 和 f_j^{\max} 分别为第 j 个目标函数在目标空间中的最小值和最大值; $S_F(X_i)$ 表示粒子 X_i 的快速分层值。快速分层策略的具体流程如下: 首先运用式(5)计算外部存档中每个粒子的快速分层值, 并根据分层值的大小将粒子进行升序分层, 第一层为第一等级, 第二层为第二等级, 依次给出所有粒子的等级, 然后从第一等级的粒子中随机选择一个作为全局学习样本。如图 2 所示, 以二维目标空间中的 3 个粒子为例, 图 2 中给出了将粒子进行快速分层后的示意图, 从图 2 中可以看出, 第一等级的粒子距离真实的 Pareto 前沿(PF) 较近。选择第一等级的粒子作为全局学习样本来引导飞行, 可以提升粒子向真实的 Pareto 前沿飞行的概率, 进而提升算法的收敛性。

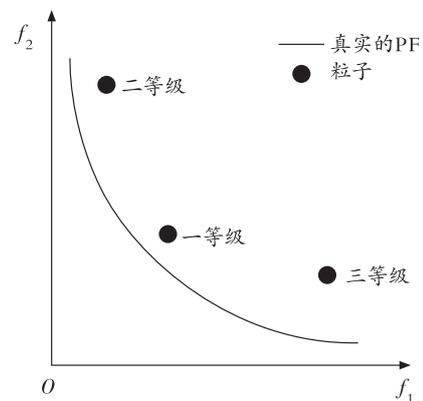


图 2 全局学习样本的选取示意图

Fig. 2 Schematic diagram of global learning sample selection

2.3 算法流程

图 3 为 DDFSMOPSO 算法的流程图。

具体步骤如下:

- ① 设置 DDFS MOPSO 算法的相关参数,初始化种群;
- ② 对粒子位置进行变异操作,并计算其适应度值;
- ③ 对于每个粒子,将当前迭代的适应度值与该粒子的历史最优适应度值进行比较,优者即被选为个体学习样本(pbest),若相等,则随机选择一个;
- ④ 建立外部存档,然后利用拥挤距离和绝对距离相结合的双决策策略对其规模进行控制,并将存档中的粒子予以更新;
- ⑤ 利用快速分层策略在外部存档中选择一个粒子作为全局学习样本(gbest)引导飞行;
- ⑥ 运用式(1)和式(2)更新每个粒子的位置和速度;
- ⑦ 判断是否满足终止条件,若满足则迭代结束,否则返回第②步。

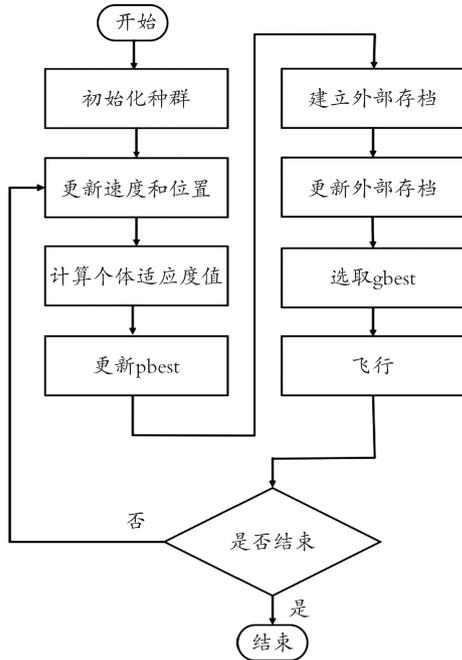


图 3 DDFS MOPSO 算法流程图

Fig. 3 Flow chart of DDFS MOPSO algorithm

3 实验结果与分析

3.1 评价指标

本文采用收敛性指标(GD)^[9]度量算法所获得的非劣解与真实的 Pareto 前沿的逼近程度,GD 值越

小意味着非劣解越接近真实的 Pareto 前沿,即算法的收敛性越好;采用分布性指标(Δ)^[10]度量算法所获得的非劣解的分布情况, Δ 值越小意味着非劣解的分布越均匀,多样性也就越好;采用标准化空间指标(I_H)^[11]度量算法所获非劣解覆盖真实的 Pareto 前沿与参考点 RP 所包围空间的比率, I_H 值越大意味着算法的综合性能越好。由于篇幅所限,3 种评价指标的表达式请参阅文献[9-11]。

3.2 实验条件及参数设置

为了更好地验证 DDFS MOPSO 算法的性能,选取 4 个二维测试函数(ZDT1, ZDT2, ZDT3, ZDT4)^[12]和两个三维测试函数(DTLZ1, DTLZ2)^[13]对 4 种算法(DDFS MOPSO, NSPSO^[14], MOPSO^[3], SMOPSO^[5])进行测试。DDFS MOPSO 算法的参数设置如下:种群规模为 200,外部存档规模为 200,迭代次数为 2 000,其他算法的参数与各种算法提出时保持一致。

3.3 算法的仿真结果分析

图 4—图 7 分别展示了 DDFS MOPSO, MOPSO, NSPSO 和 SMOPSO 4 种算法在二维测试函数 ZDT1, ZDT2, ZDT3, ZDT4 上的仿真结果图。图 8—图 9 分别展示了 DDFS MOPSO, MOPSO, NSPSO 和 SMOPSO 4 种算法在三维测试函数 DTLZ1, DTLZ2 上的仿真结果图。

从图 4—图 7 可以得出:在测试函数 ZDT1 和 ZDT2 上,DDFS MOPSO 算法所获得的非劣解的收敛性明显好于算法 NSPSO 和 SMOPSO;在测试函数 ZDT3 上,DDFS MOPSO, MOPSO 和 SMOPSO 3 种算法所获得的非劣解都能收敛到真实的 Pareto 前沿,并且 DDFS MOPSO 算法所获得的非劣解的分布性明显优于 MOPSO 算法,算法 NSPSO 所获得的非劣解与真实的 Pareto 前沿还有一定的距离;值得注意的是,在测试函数 ZDT4 上,DDFS MOPSO 算法所获得的非劣解几乎能覆盖整个 Pareto 前沿,无论是解的分布性还是收敛性,DDFS MOPSO 算法都要好于算法 MOPSO, NSPSO, SMOPSO。从图 8—图 9 可以得出:在测试函数 DTLZ1 上,DDFS MOPSO 算法所获得的非劣解的分布性明显优于算法 MOPSO, NSPSO 和 SMOPSO;在测试函数 DTLZ2 上,DDFS MOPSO 算法所获得的非劣解的分布性优于 MOPSO 算法,略差

于算法 NSPSO 和 SMOPSO。综上所述,相比于算法 MOPSO, NSPSO 和 SMOPSO, DDFSOPSO 算法在解的分布性和收敛性上都表现出较大的优势。

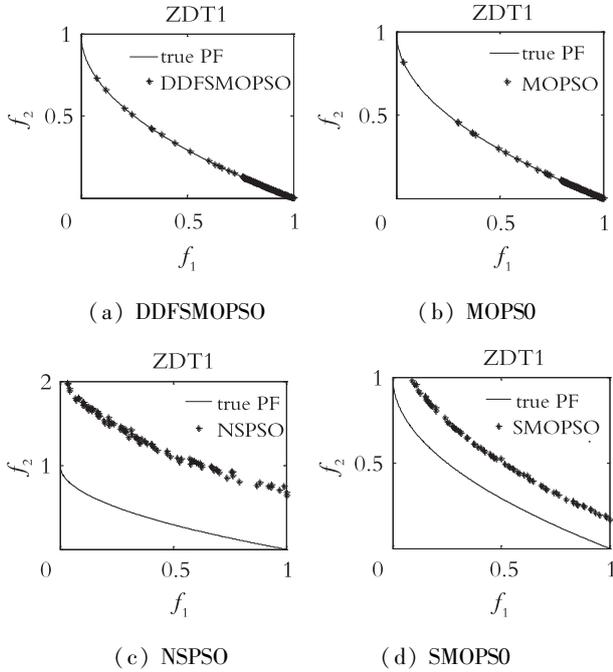


图 4 算法在测试函数 ZDT1 上的仿真结果
Fig. 4 Simulation results of algorithm on test function ZDT1

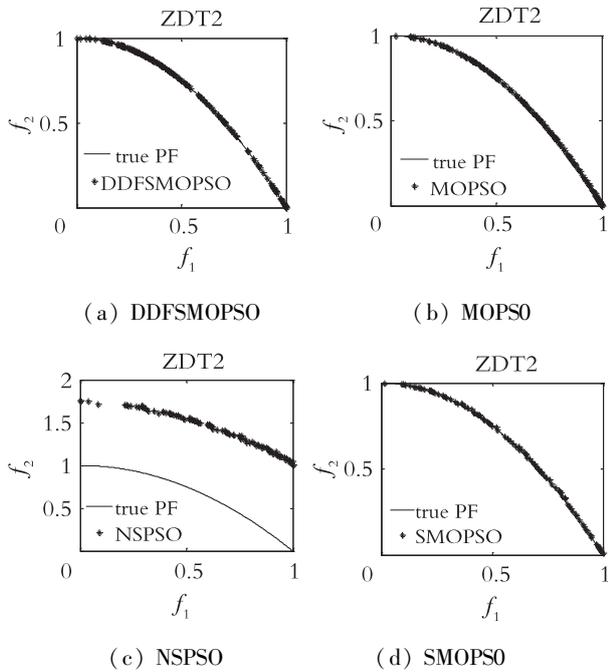


图 5 算法在测试函数 ZDT2 上的仿真结果
Fig. 5 Simulation results of algorithm on test function ZDT2

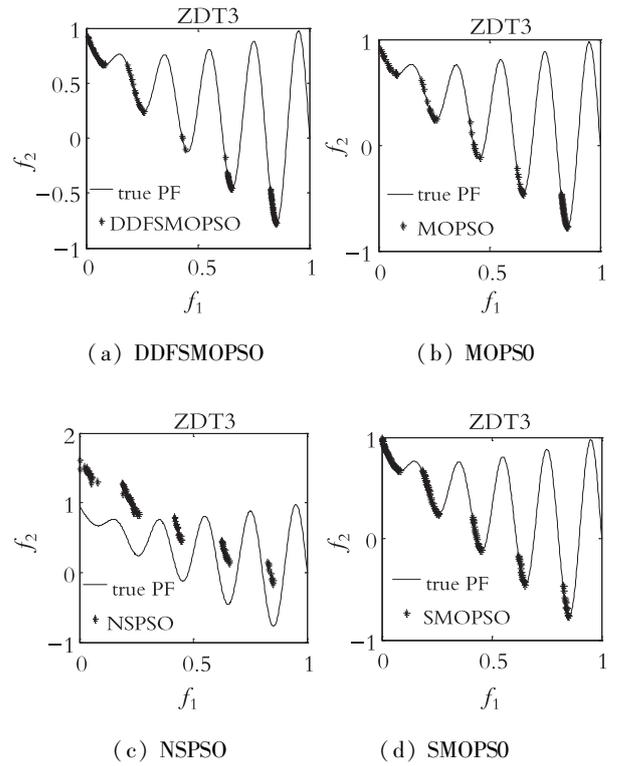


图 6 算法在测试函数 ZDT3 上的仿真结果
Fig. 6 Simulation results of algorithm on test function ZDT3

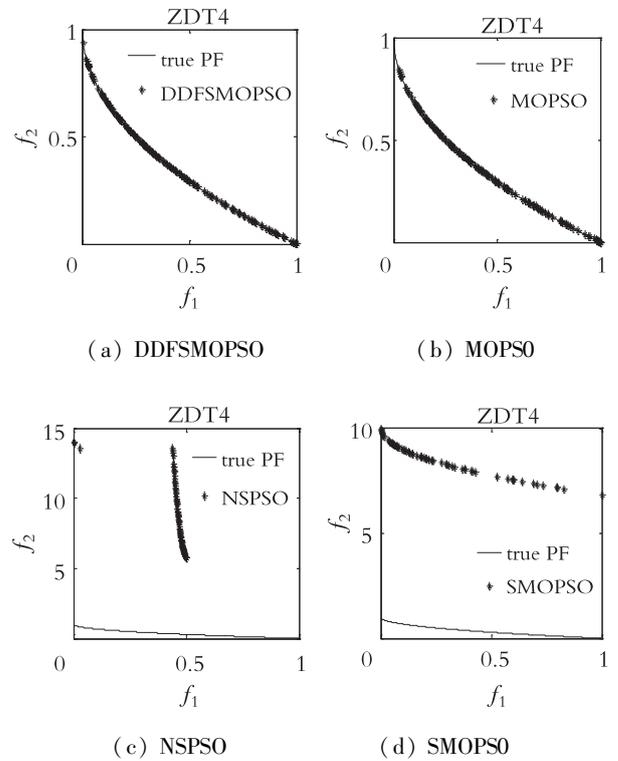


图 7 算法在测试函数 ZDT4 上的仿真结果
Fig. 7 Simulation results of algorithm on test function ZDT4

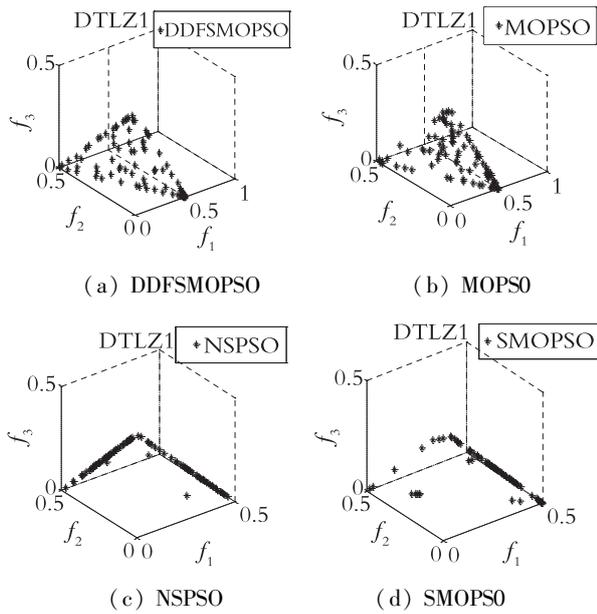


图 8 算法在测试函数 DTLZ1 上的仿真结果

Fig. 8 Simulation results of algorithm on test function DTLZ1

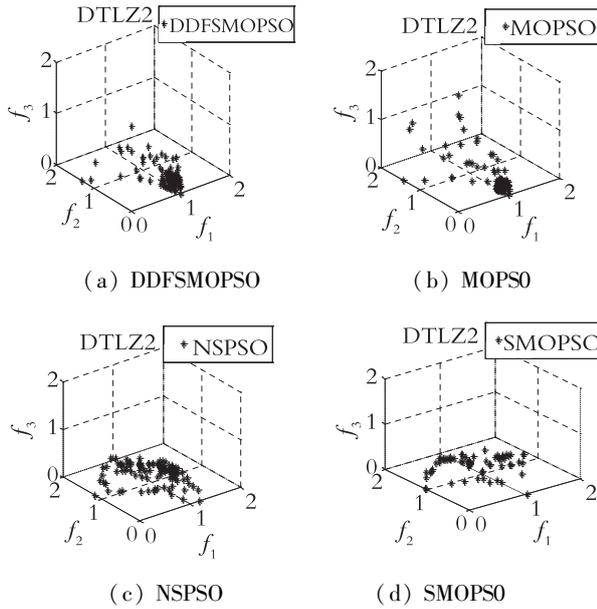


图 9 算法在测试函数 DTLZ2 上的仿真结果

Fig. 9 Simulation results of algorithm on test function DTLZ2

3.4 算法的收敛性和多样性分析

表 1 和表 2 分别展示了 DDFS MOPSO, MOPSO, NSPSO 和 SMOPSO 4 种算法在测试函数 ZDT1, ZDT2, ZDT3 和 ZDT4 上运行 30 次所获得的收敛性指标(GD)和分布性指标(Δ)的最大值、最小值、平均值和标准差,图 10 和图 11 分别展示了各种算法运行 30 次所获得的 GD 值和 Δ 值的盒状统计图(横坐标,D 表示 DDFS MOPSO 算法;M 表示 MOPSO 算

法;N 表示 NSPSO 算法;S 表示 SMOPSO 算法)。下面将根据各种算法所获得的 GD 指标和 Δ 指标,对算法的收敛性和多样性进行分析。

表 1 各种算法在测试函数 ZDT1-ZDT4 上的 GD 指标
Table 1 GD index of various algorithms on test functions ZDT1-ZDT4

函数	GD	DDFSMOPSO	MOPSO	NSPSO	SMOPSO
ZDT1	Min	5.86E-04	6.39E-04	2.85E-01	3.55E-03
	Max	1.36E-03	1.13E-03	1.22E+00	5.33E-01
	Mean	7.41E-04	8.23E-04	6.59E-01	6.66E-02
	Std	1.58E-04	1.46E-04	1.98E-01	1.26E-01
ZDT2	Min	7.53E-04	1.29E-03	5.06E-01	2.11E-03
	Max	1.13E-03	1.72E-03	2.16E+00	5.83E-01
	Mean	8.84E-04	1.53E-03	1.05E+00	9.42E-02
	Std	8.77E-05	9.56E-05	4.16E-01	1.49E-01
ZDT3	Min	3.83E-03	4.47E-03	1.11E-01	4.96E-03
	Max	5.20E-03	5.74E-03	8.60E-01	2.68E-01
	Mean	4.43E-03	5.17E-03	3.76E-01	3.00E-02
	Std	3.54E-04	2.51E-04	1.99E-01	5.55E-02
ZDT4	Min	6.18E-04	7.86E-04	5.34E+00	5.55E+00
	Max	9.84E-04	1.05E-03	2.24E+01	1.47E+01
	Mean	7.11E-04	9.16E-04	9.90E+00	9.73E+00
	Std	7.46E-05	5.03E-05	3.43E+00	1.94E+00

表 2 各种算法在测试函数 ZDT1-ZDT4 上的 Δ 指标
Table 2 Δ index of various algorithms on test functions ZDT1-ZDT4

函数	Δ	DDFSMOPSO	MOPSO	NSPSO	SMOPSO
ZDT1	Min	1.46E+00	1.41E+00	5.67E-01	7.14E-01
	Max	1.79E+00	1.72E+00	7.01E-01	1.14E+00
	Mean	1.61E+00	1.61E+00	6.21E-01	8.49E-01
	Std	7.35E-02	7.32E-02	3.72E-02	1.18E-01
ZDT2	Min	6.73E-01	1.01E+00	4.08E-01	7.38E-01
	Max	1.13E+00	1.18E+00	8.87E-01	1.13E+00
	Mean	9.18E-01	1.07E+00	6.73E-01	8.87E-01
	Std	1.07E-01	4.71E-02	8.81E-02	1.07E-01
ZDT3	Min	1.33E+00	1.46E+00	5.56E-01	9.16E-01
	Max	1.54E+00	1.57E+00	7.64E-01	1.27E+00
	Mean	1.42E+00	1.52E+00	6.53E-01	1.09E+00
	Std	4.66E-02	3.11E-02	5.18E-02	7.75E-02
ZDT4	Min	7.19E-01	9.40E-01	6.06E-01	9.63E-01
	Max	1.25E+00	1.16E+00	9.88E-01	1.07E+00
	Mean	8.69E-01	1.02E+00	6.80E-01	9.98E-01
	Std	9.84E-02	5.34E-02	7.05E-02	2.91E-02

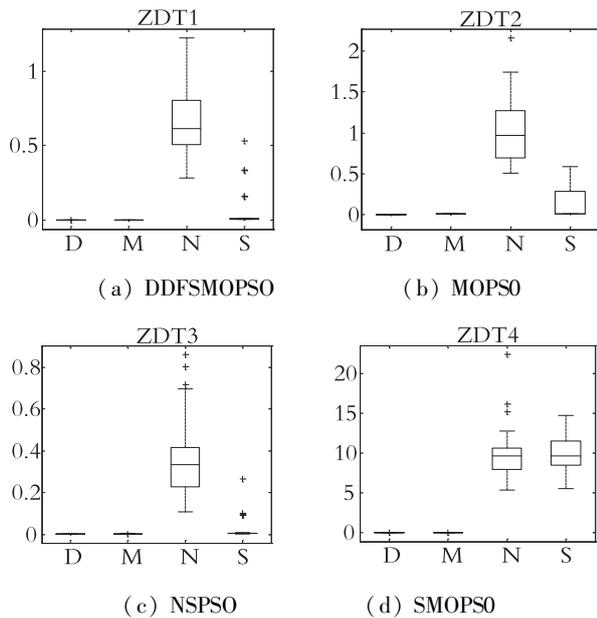


图 10 GD 值的盒状统计图

Fig. 10 Box chart of GD values

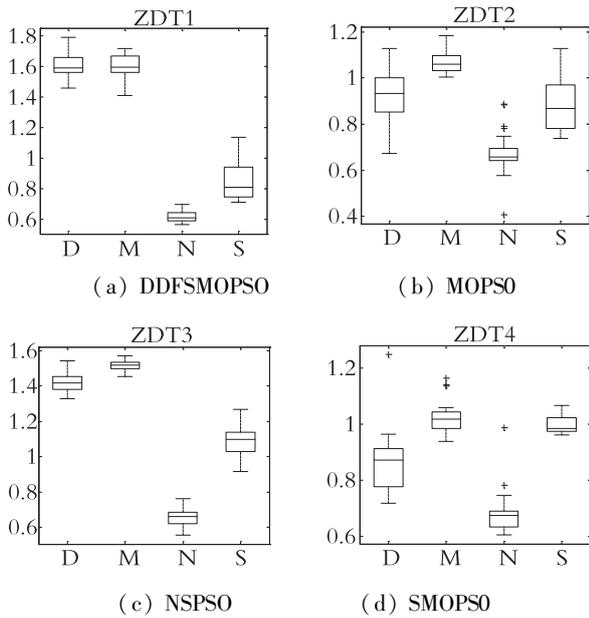


图 11 Δ 值的盒状统计图

Fig. 11 Box chart of Δ values

从表 1 可以看出:在测试函数 ZDT1—ZDT4 上,DDFSMOPSO 算法获得了比其他 3 种算法更优的 GD 值,说明 DDFSOPSO 算法的收敛性要优于算法 MOPSO,NSPSO 和 SMOPSO。从图 10 可以看出,在测试函数 ZDT1, ZDT2, ZDT3 和 ZDT4 上,DDFSMOPSO 算法所获得的 GD 值的平均值比算法 MOPSO,NSPSO 和 SMOPSO 更接近于零,这与表 1 的分析结果一致,进一步说明 DDFSOPSO 算法具

有较强的收敛性。

从表 2 可以看出:在大多数测试函数上,DDFSMOPSO 算法所获得的 Δ 值明显好于 MOPSO 算法,略差于 NSPSO 和 SMOPSO 算法,这意味着 DDFSOPSO 算法的多样性好于 MOPSO 算法,相比于算法 NSPSO 和 SMOPSO,DDFSMOPSO 算法的多样性还有提升的空间。从图 11 可以看出,在测试函数 ZDT2,ZDT3 和 ZDT4 上,DDFSMOPSO 算法所获得的 Δ 值的平均值比 MOPSO 算法更接近于零,这与表 2 的分析结果一致。

3.5 算法的综合性分析

表 3 展示了 DDFSOPSO, MOPSO, NSPSO 和 SMOPSO 4 种算法在三维测试函数 DTLZ1 和 DTLZ2 上运行 30 次所获得的 I_H 指标的最大值、最小值、平均值和标准差,图 12 展示了各种算法运行 30 次所获得的 I_H 值的盒状统计图(横坐标 D 表示 DDFSOPSO 算法;M 表示 MOPSO 算法;N 表示 NSPSO 算法;S 表示 SMOPSO 算法)。

表 3 各种算法在测试函数 DTLZ1—DTLZ2 上的 I_H 指标
Table 3 I_H index of various algorithms on test functions DTLZ1—DTLZ2

函数	I_H	DDFSMOPSO	MOPSO	NSPSO	SMOPSO
DTLZ1	Min	5.18E-01	5.22E-01	4.99E-01	4.68E-01
	Max	6.58E-01	7.35E-01	6.98E-01	7.18E-01
	Mean	6.02E-01	5.85E-01	5.98E-01	5.75E-01
	Std	3.16E-02	4.38E-02	4.33E-02	5.22E-02
DTLZ2	Min	5.05E-01	4.97E-01	5.33E-01	4.95E-01
	Max	7.26E-01	7.28E-01	6.81E-01	6.80E-01
	Mean	6.09E-01	5.90E-01	5.96E-01	5.81E-01
	Std	5.31E-02	4.92E-02	3.86E-02	4.59E-02

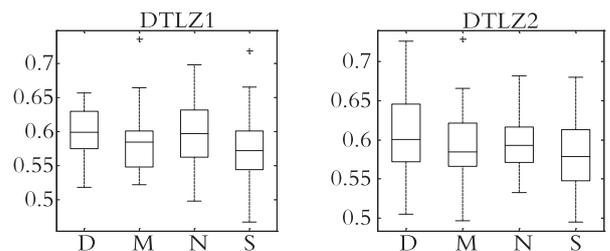


图 12 I_H 值的盒状统计图

Fig. 12 Box chart of I_H values

从表 3 可以看出,在测试函数 DTLZ1 和 DTLZ2

上,DDFSMOPSO 算法所获得的 I_H 值明显优于算法 MOPSO, NSPSO 和 SMOPSO,说明 DDFSOPSO 算法有较好的综合性能。从图 12 可以看出,在测试函数 DTLZ1 和 DTLZ2 上,DDFSMOPSO 算法所获得的 I_H 值的平均值都大于算法 MOPSO, NSPSO 和 SMOPSO,说明 DDFSOPSO 算法的综合性能好于与之相比的 3 种算法。

4 结束语

为了提升种群的多样性和收敛性,本文提出了基于双决策和快速分层的多目标粒子群算法(DDFSMOPSO)。该算法采用拥挤距离和绝对距离相结合的双决策策略对外部存档规模进行维护,使得优秀粒子在随后的进化过程中易于保留和发展,有效地提升了种群的多样性。同时,为提升粒子收敛到真实 Pareto 前沿的概率,采用快速分层策略对全局学习样本进行选取。将 DDFSOPSO 算法与 MOPSO, NSPSO 和 SMOPSO 3 种算法在同一环境下进行仿真实验,实验结果表明:与其他 3 种算法相比,DDFSOPSO 算法在多样性、收敛性以及综合性上均有明显改善,这意味着 DDFSOPSO 可以作为求解多目标优化问题的有效算法。

参考文献(References):

- [1] WANG L, SINGH C. Environmental economic power dispatch using a fuzzified multi-objective particle swarm optimization algorithm [J]. *Electric Power Systems Research*, 2007, 77(12):1654—1664.
- [2] KENNEDY J, EBERHART R C. Particle swarm optimization [C]//*IEEE International Conference on Neural Networks*. Piscataway, 1995:1942—1948.
- [3] COELLO C A C, PULIDO G T, LECHUGA M S. Handling multiple objectives with particle swarm optimization [J]. *IEEE Trans on Evolutionary Computation*, 2004, 8(3):256—279.
- [4] LI L, WANG W, LI W, et al. A novel ranking-based optimal guides selection strategy in MOPSO[J]. *Procedia Computer Science*, 2016, 91(1):1001—1010.
- [5] MOSTAGHIM S, TEICH J. Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO) [C]// *Proceedings of IEEE Swarm Intelligence Symposium*. Piscataway, 2003: 26—33.
- [6] LI X, LIA X L, WANG K, et al. A multi-objective particle swarm optimization algorithm based on enhanced selection[J]. *IEEE ACCESS*, 2019, 7(1):168091—168103.
- [7] LENG R, OUYANG A, LIU Y, et al. A multi-objective particle swarm optimization based on grid distance [J]. *International Journal of Pattern Recognition and Artificial Intelligence*, 2019, 34(3): 1—23.
- [8] DEB K, PRATAP A. A fast and elitist multi-objective genetic algorithm: NSGA-II [J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2):182—197.
- [9] SCHUTZE O, ESQUIVE X, LARA A, et al. Using the averaged Hausdorff distance as a performance measure in evolutionary multi-objective optimization [J]. *IEEE Transactions on Evolutionary Computation*, 2012, 16(4): 504—522.
- [10] SCOTT J R. Fault tolerant design using single and multi-criteria genetic algorithm optimization [J]. *Cellular Immunology*, 1995, 37(1):1—13.
- [11] STORN R, PRICE K. Differential evolution: a simple and efficient Heuristic for global optimization over continuous spaces [J]. *Journal of Global Optimization*, 1997, 11(4): 341—359.
- [12] ZITZLER E, DEB K, THIELE L. Comparison of multi-objective evolutionary algorithms: empirical results [J]. *Evolutionary Computation*, 2000, 8(2): 173—195.
- [13] DEB K, THIELE L, LAUMANN M, et al. Scalable multi-objective optimization test problems [J]. *Evolutionary Computation*, 2002, 1(1): 825—830.
- [14] LI X D. A non-dominated sorting for particle swarm optimizer for multi-objective optimization [C]// *Proceedings of the International Conference on Genetic and Evolutionary Computation*. Atlanta, 2003: 37—48.

Multi-objective Particle Swarm Optimization Based on Double Decision and Fast Stratification

WANG Shi-hua¹, LI Na-na², LIU Yan-min³

(1. School of Mathematics and Statistics, Guizhou University, Guiyang 550025, China; 2. School of Data Science and Information Engineering, Guizhou Minzu University, Guiyang 550025, China; 3. School of Mathematics, Zunyi Normal College, Guizhou Zunyi 563006, China)

Abstract: In order to solve the problems of insufficient convergence and diversity in multi-objective particle swarm optimization, a new multi-objective particle swarm optimization based on double decision and fast stratification(DDFSMOPSO) is proposed. In this algorithm, the external archive is used to store the non inferior solutions generated by iteration, and the double decision strategy combining crowding distance and absolute distance is used to maintain the external archive scale, which makes the excellent particles easy retained and developed in the subsequent evolution process. At the same time, the fast stratification strategy is used to select the global learning samples from the external archiving, which can be used to lead the evolution of particles in the population and promote the particles in the population to move to the real Pareto front. DDFSOPSO algorithm and three classical multi-objective particle swarm optimization are compared in simulation experiment on some test functions of ZDT and DTLZ series. Experimental results show that compared with other classical algorithms, DDFSOPSO algorithm has better convergence and diversity. Therefore, DDFSOPSO can be used as an effective algorithm for solving multi-objective optimization problems.

Key words: multi-objective optimization; particle swarm optimization; double decision; fast stratification

责任编辑:李翠薇

引用本文/Cite this paper:

王世华,李娜娜,刘衍民. 基于双决策和快速分层的多目标粒子群算法[J]. 重庆工商大学学报(自然科学版),2022,39(1): 62—70.

WANG Shi-hua, LI Na-na, LIU Yan-min. Multi-objective Particle Swarm Optimization Based on Double Decision and Fast Stratification[J]. Journal of Chongqing Technology and Business University (Natural Science Edition),2022,39(1):62—70.