

doi:10.16055/j.issn.1672-058X.2022.0002.003

# 基于自适应网格混合机制的多目标粒子群算法

邹康格<sup>1</sup>, 刘衍民<sup>2</sup>

(1. 贵州大学 数学与统计学院, 贵阳 550025; 2. 遵义师范学院 数学学院, 贵州 遵义 563006)

**摘要:**当粒子群算法(PSO)解决多目标优化问题时,由于 PSO 有较快的收敛效果,使得种群在寻优过程中多样性不足,易使算法早熟收敛。为有效设计多目标粒子群算法,提出基于自适应网格混合机制的多目标粒子群算法(ammMOPSO)。该算法采用自适应网格和混合机制的一种双重维护策略,以保证外部存档中的非劣解分布均匀,避免种群快速退化,影响粒子开发能力;利用混合机制中的加权策略在外部存档的非劣解中确定全局最优样本,增加了种群的多样性,提升粒子飞向真实 Pareto 前沿的概率;同时,为防止算法停滞,陷入局部最优的问题,还引入一个变异操作对粒子的位置进行动态变异,增强了粒子的探索能力。仿真实验结果表明:所提算法与其他 3 个国际经典的多目标粒子群算法相比,具有较优的收敛性和多样性,且有较好的空间化效果。

**关键词:**多目标优化;粒子群算法;自适应网格;混合机制;变异操作

**中图分类号:**TP18

**文献标志码:**A

**文章编号:**1672-058X(2022)02-0014-10

## 0 引言

当今多数科学和工程问题通常有多个目标,决策者需要同时使多个目标在给定的范围内尽可能得到最佳优化,即多目标优化问题。在多目标优化问题中,由于各目标之间存在冲突,多目标优化问题不存在唯一的最优解,因此分析一个多目标优化问题的目标是找到一组非劣解而不是单个解,非劣解也称为帕累托解,且一组非劣解构成了客观空间中的帕累托前沿,因此解决多目标优化问题的核心目标就是使这些非劣解尽可能接近真实的帕累托前沿且

保持分布均匀。粒子群算法(PSO)由于收敛速度快,易于实现,在许多单目标优化问题中得到广泛应用,并取得了良好的效果。当 PSO 运用于多目标优化问题时,即多目标粒子群算法(MOPSO),在解决多目标问题中由于 MOPSO 算法收敛速度快,容易导致种群陷入局部最优解,使得算法早熟收敛。对于上述问题,有很多学者提出了改良的多目标粒子群算法:在文献[1]中,提出带有随机迁移的多目标粒子群算法,该算法设置一个年龄阈值,当一个粒子没有改良它的个体位置时,会增加它的年龄,并且当粒子年龄超过这个年龄阈值时,粒子将被重新初始化,以此提高种群多样性,但分解替换占优关系,在

收稿日期:2020-12-15;修回日期:2021-05-22.

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:邹康格(1996—),男,贵州岑巩人,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,博士研究生导师,从事优化理论智能算法研究. Email: yanmin7813@

解决一些复杂的多目标优化问题时,表现不佳。在文献[2]中,提出一种改进的时变参数跟随蜂搜索的多目标粒子群算法,应用个体最优解的权重随着迭代的进行而逐步减小,该方法消除了相对于全局最优解质量较差的影响,提高了算法的收敛能力,但在解决多局部前沿的多目标优化问题时存在困难,且非劣解多样性较差。在文献[3]中,提出一种基于健康度的多目标粒子群算法,该算法定义了健康度的概念,当粒子越来越接近最优解时,该粒子健康度上升,反之下降,使得粒子跳出局部最优,虽然该算法能保证种群的多样性和在 Pareto 前沿上分布均匀,但算法仍存在收敛性不足的问题。在文献[4]中,提出一种动态多个子群的多目标粒子群算法,该算法通过动态群策略自适应地调整搜索过程中粒子群的数量,该算法根据需要分配适当数量的蜂群,以支持蜂群之间的收敛性和多样性准则,但种群数量的增长速度不够高,缺乏良好的策略来加强种群之间和种群内部的交流。

上述算法的有效性在一定程度得到了提升,但是,当面对复杂的多个目标且多个变量的优化问题,算法极易出现早熟收敛。为了解决这个问题,进一步提升 MOPSO 算法的性能,本文提出基于自适应网格混合机制的多目标粒子群算法(ammMOPSO)。对 MOPSO 主要改进包括:为保障种群的收敛性和多样性,在使用网格技术基础上使用“加权和函数”策略去挑选全局最优样本;在外部存档中使用“加权和函数”和相邻距离的混合机制删除外部存档中的粒子,以提高非劣解的质量;为了保证粒子有跳出局部最优的能力,使用一种变异操作策略对粒子位置进行变异。仿真实验结果表明,所提出的改进算法是对多目标粒子群算法的有效改进。

## 1 多目标优化问题与粒子群算法

### 1.1 多目标优化问题

一般多目标优化问题的定义描述如下:

$$\min F(x) = (f_1(x), f_2(x), \dots, f_k(x))$$

$$\text{s. t. } g_i(x) \leq 0, i = 1, 2, \dots, m$$

$$h_j(x) = 0, j = 1, 2, \dots, p$$

其中,  $x = (x_1, x_2, \dots, x_n)^T$  是  $n$  维决策向量,  $f_i: R^n \rightarrow R, i = 1, 2, \dots, k$  是目标函数,  $g_i, h_j: R^n \rightarrow R, i = 1, 2, \dots, m; j = 1, 2, \dots, p$  分别是不等式约束和等式约束。

### 1.2 基本粒子群算法

PSO 通过在规定范围内随机初始化一个均匀分布的粒子,群体中的全部粒子都具备影象功效,每一个粒子都有自己的位置和速度,且由全局最优粒子( $g_{\text{best}}$ )和个体最优粒子( $p_{\text{best}}$ )驱动粒子寻找最优解。在单目标粒子群算法中,由于只有一个目标函数,只需比较其适应度值就可以唯一确定  $g_{\text{best}}$  和  $p_{\text{best}}$ 。在多目标粒子群算法中,关于选择  $p_{\text{best}}$ ,如果粒子的每个目标都好则该粒子更优,如果不能严格对比谁更优则随机选择其中一个作为个体最优;关于选择  $g_{\text{best}}$ ,有很多的非劣解都可以作为全局最优个体,然而在 PSO 中,每个粒子只能选择一个作为全局最优个体,因此保存这些非劣解和挑选出较好的一个非劣解是粒子群算法处理多目标优化问题的核心。粒子的速度和位置更新方程如下:

$$\begin{aligned} v(t+1) &= \omega v(t) + c_1 r_1 (p(t) - x(t)) + \\ &\quad c_2 r_2 (g(t) - x(t)) \\ x(t+1) &= x(t) + v(t+1) \end{aligned} \quad (1)$$

其中,  $\omega$  是惯性权重,它的大小控制着搜寻本领的大小;  $c_1, c_2$  分别是个体经验系数、社会经验系数;  $r_1, r_2$  是在范围  $[0, 1]$  之间的随机数;  $p$  和  $g$  分别是个体最优解、全局最优解。

## 2 改进算法

全局最优样本的选取、外部存档规模的控制和粒子的变异操作是粒子群算法处理多目标优化问题的主要改进部分。本文在全局最优样本选取部分,提出自适应网格和“加权和函数”策略,并结合“加权和函数”和相邻距离的外部存档维护,提出一种变异操作对粒子的位置进行变异,进一步改善多目标粒子群算法的性能。对于自适应网格的方法,请参考文献[5]。

### 2.1 基于加权策略的全局最优样本选取

在多目标粒子群算法的每一次迭代中都会产生很多的非劣解,由于它们互不占优,很难选择出最优

的粒子,所以需要考虑保障非劣解解集的多样性。为了保障非劣解解集的多样性,本文使用“加权和函数” $F(x)$ 和轮盘赌策略对同一个网格中的粒子选出全局最优样本。

首先粒子的所有目标值使用一个标准化的过程,标准化过程可以通过将每个目标函数除以其最大值来实现;由于搜索空间已知,可以很容易地计算出每个目标函数的最大值;其次该粒子的所有目标函数标准化后,得到一个“加权和函数” $F(x)$ 是该粒子的所有目标函数标准化之和。

粒子的每个目标函数权重为粒子的每个目标值分别除以在所有粒子中所对应目标函数的最大值。其定义如下:

$$w_m = \frac{f_m(x)}{F_m}$$

其中, $f_m(x)$ 为粒子 $x$ 第 $m$ 个目标函数值, $F_m$ 为所有粒子在第 $m$ 个目标函数值中的最大值。

粒子的标准化过程为粒子的每个目标函数权重乘以其对应的目标函数值。

粒子的每个目标函数权重乘以其对应的目标函数值之和作为该粒子的“加权和函数”值,其定义如下:

$$F(x) = \sum_{m=1}^M w_m f_m(x) \quad (2)$$

其中, $f_m(x)$ 为粒子 $x$ 第 $m$ 个目标函数值, $w_m$ 为粒子 $x$ 第 $m$ 个目标函数权重, $M$ 是目标函数个数。

为了说明在同一网格的非劣解中“加权和函数”值越小,则该非劣解粒子处于网格中所有粒子的中心位置(中心粒子)或者越接近网格中央;假设在同一网格中有5个非劣解,它们的函数值坐标分别为: $P_1(1,5)$ 、 $P_2(2,4)$ 、 $P_3(3,3)$ 、 $P_4(4,2)$ 、 $P_5(5,1)$ ,根据式(2)得到这5个非劣解的“加权和函数”值分别是5.24、4.36、4.52,可以得出 $P_3(3,3)$ 是最小的,又由 $P_3(3,3)$ 是中心粒子且越有可能接近网格中央的非劣解,所以在同一网格的非劣解中“加权和函数”值越小,则该非劣解是中心粒子或越可能接近网格中央的粒子。

如图1所示,其中实心圆点和空心圆点分别代表全局最优样本和非劣解,在同一网格中选取“加权和函数”值最小的粒子(网格中心粒子)作为全局

最优,若有相同的最小值,则随机选择其中一个,使得本文提出的算法在搜寻过程中引导粒子分别向距离相对较远的非劣解飞行,以增强种群的多样性。

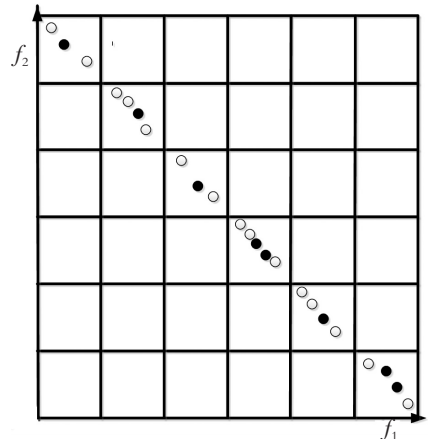


图1 全局最优样本的选取

Fig. 1 Selection of global optimal sample

## 2.2 基于混合机制的外部存档维护

在多目标粒子群算法的每一次迭代中都将非劣解储存在外部存档中,随着算法运行,非劣解的数量也随着增加,当外部存档达到饱和时,则需要对外部存档进行维护。在很多的非劣解粒子群算法中都使用自适应网格策略对外部存档中的非劣解进行删选,其具体方式是找到粒子数最多的网格,随机删除网格中的粒子或者删除网格中拥挤度最大的粒子,这些策略都有可能非劣解分布不均匀。因此,本文提出“加权和函数”和相邻距离的混合策略,确保外部存档中的非劣解保持良好的分布,从而为全局最优解的选择提供较好的候选解。

定义粒子 $x_i$ 的相邻距离为粒子 $x_{i+1}$ 与粒子 $x_{i-1}$ 之间的欧式距离。在处于同一网格的粒子中,计算某粒子的相邻距离等于邻居两个粒子的欧氏距离大小,粒子的相邻距离越小,则该粒子的密度越大。其相邻距离定义如下:

$$\begin{cases} d(i) = \sum_{m=1}^M \sqrt{(f_m(x_{i+1}) - f_m(x_{i-1})))^2} \\ d(1) = \infty \\ d(n) = \infty \end{cases} \quad (3)$$

在图2中,图2(a)代表外部存档中的非劣解分布情况,图2(b)、图2(c)和图2(d)分别代表随机删除、拥挤度删除和“加权和函数”与相邻距离的混合删除;在图2(b)中从非劣解数量最多的网

格中随机删除 40% 的粒子数,这种随机性可能导致网格中的非劣解有些分散,有些拥挤;在图 2(c) 中从非劣解数量最多的网格中根据拥挤程度大小删除 40% 拥挤程度最大的粒子数,虽然保障了密度最大的网格中的非劣解分布均匀,但是有可能与其他相邻的网格中的边界非劣解比较拥挤;在图 2(d) 中从非劣解数量最多的网格中首先根据式(3)删除 20% 相邻距离最小的粒子数,其次由于在同一网格中非劣解的“加权和函数”值越大,则越可能接近网格的边界,所以再从非劣解数量最多的网格中根据式(2)删除 20% “加权和函数”值最大的粒子数,这种策略既能保持密度最大的网格中的非劣解分布均匀,也能与其他相邻网格中边界粒子保持距离。比较图 2(b)、图 2(c) 和图 2(d) 可知,本文提出混合机制的外部存档维护,更好地改善了非劣解分布情况。

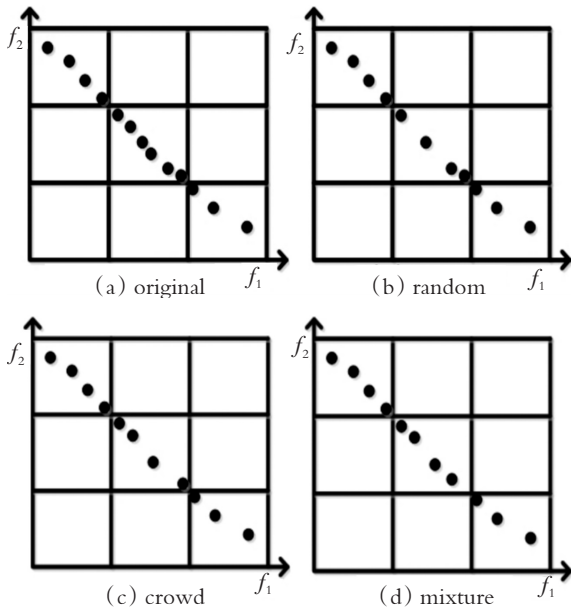


图 2 外部存档的维护示意图

Fig. 2 A maintenance schematic diagram of the external archive

### 2.3 变异操作

MOPSO 算法是一种群体智能算法,在迭代早期搜索能力较强,可以不断搜索新的领域,但 MOPSO 算法具有快速收敛的效果,在迭代后期该算法极有可能在局部最优解的周围精细搜索,使得该算法早熟收敛。因此,为了使算法在迭代早期保持良好的种群多样性,在迭代后期提升种群多样性,使用变异

操作对位置递增程度的变异。粒子的位置变异程度随着迭代次数的增加线性递增,在迭代初期,种群的多样性良好,使粒子的变异程度较小,从而种群的多样性保持稳定,在迭代后期,种群的多样性不足,使粒子的变异程度较大,以增强种群的多样性。在 ammmMOPSO 算法中,粒子速度的更新公式为式(1),粒子位置的更新公式如下:

$$x(t+1) = \frac{e^{-\frac{n}{N}}}{1+e^{-\frac{n}{N}}} x(t) + v(t+1) \quad (4)$$

其中, $N$  为最大迭代次数, $n$  为当前迭代次数, $e$  为自然对数。

### 2.4 算法的主要流程

算法的主要流程如下:

**Step 1** 初始化。设置种群数目、最大的外部存档数目、粒子的维数、网格数目和最大迭代次数;随机初始化位置和速度;对每一个粒子位置初始值作为该粒子的个体最优值;创建外部存档并设为空集。

**Step 2** 计算粒子的适应度。计算每个粒子的适应度,把非劣解储存到外部存档中。

**Step 3** 建立自适应网格。对目标空间创建网格,分别用式(2)和式(3)计算每一个网格中的非劣解。

**Step 4** 个体最优样本和全局最优样本的选取。个体最优样本是当前位置与个体历史最优位置之间的优者,若互不占优,则随机选择一个,全局最优样本是利用轮盘赌方法对每一个网格的非劣解中选择“加权和函数”值最小的非劣解。

**Step 5** 更新速度和位置。对每个粒子的速度和位置利用飞行式(1)和式(4)更新。

**Step 6** 更新外部存档。若外部存档没有到达饱和状态,则继续引入非劣解;若外部存档达到饱和状态,找到网格粒子数最多的网格,首先根据式(3)删除 20% 相邻距离最小的粒子数,再根据式(2)删除 20% “加权和函数”值最大的粒子数,再引入非劣解进入外部存档,使得外部存档得到更新。

**Step 7** 终止条件。若当前迭代次数小于最大迭代次数,则返回 Step2,输出最优解集。

ammmMOPSO 算法流程如图 3 所示:

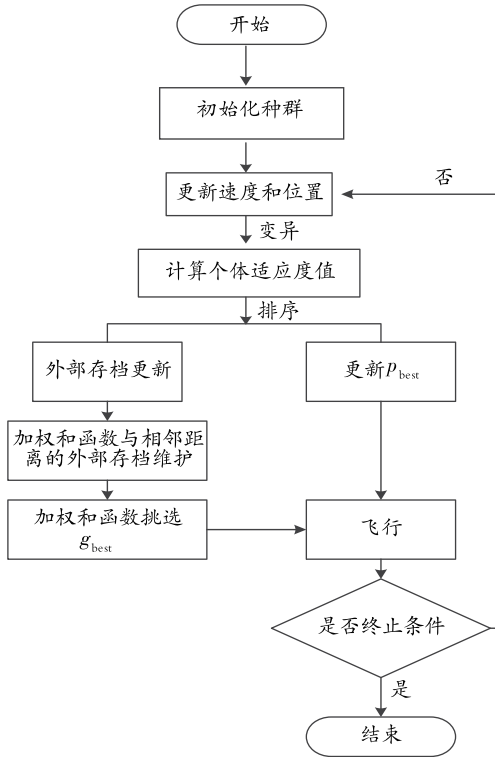


图 3 ammmMOPSO 算法的流程图

Fig. 3 ammmMOPSO algorithm flowchart

### 3 实验仿真分析

#### 3.1 性能评价指标

为了评估各算法性能。本文采用收敛性指标 ( $G_D$ )、分布性指标 ( $\Delta$ )、标准化空间指标 ( $I_H$ ) 分别对算法进行评价。 $G_D$ <sup>[8]</sup> 是权衡算法所获得的前沿与真实 Pareto 前沿之间距离大小,  $G_D$  越小表示算法的

收敛性越好。 $\Delta$ <sup>[9]</sup> 衡量算法所获得非劣解的分布情况,  $\Delta$  越小表明非劣解越有均匀的分布。 $I_H$ <sup>[10]</sup> 衡量算法在空间中非劣解分布性情况,  $I_H$  越大表明空间中非劣解分布性越好。

#### 3.2 参数选择

ammmMOPSO 算法的参数设置: 种群数量设置为 200, 最大迭代次数为 2 000 次, 外部存档规模大小设置为 200, 网格数为 50,  $c_1 = c_2$ ,  $\omega = 0.4$ 。其他 3 种比较的多目标粒子群算法的参数设置和原文献保持一致。

#### 3.3 实验结果及数据分析

为了验证 ammmMOPSO 算法的有效性, ammmMOPSO 算法与国际经典的 3 种多目标粒子群算法在双目标问题 (ZDT1-ZDT4) 和三目标问题 (DTLZ2、DTLZ4 和 DTLZ5) 上分别进行实验仿真, 对比算法包括: MOPSO<sup>[5]</sup>、SMOPSO<sup>[6]</sup> 和 EMMOPSO<sup>[7]</sup> 算法。各算法都是在同一测试函数上独立运行 30 次, 并且均迭代 2 000 次所得出的实验结果。所有实验运行的环境均是在 Intel (R) Core (TM) i7-7820 HQ CPU @ 2.90 GHZ、Windows10 操作系统和 MATLAB 2013a。

在 4 个双目标测试函数上, 表 1 和表 2 分别显示 ammmMOPSO 算法与国际经典的 3 种多目标粒子群算法的  $G_D$  指标和  $\Delta$  指标的平均值和标准差, 在 3 个三目标测试函数上, 表 3 显示 ammmMOPSO 算法与国际经典的 3 种多目标粒子群算法的  $I_H$  指标的平均值和标准差, 其中加粗的数据表示各项指标的最好结果。

表 1 不同算法在 ZDT 系列测试函数上的  $G_D$  值

Table 1  $G_D$  values of different algorithms on ZDT series test functions

function	$G_D$	ammmMOPSO	MOPSO	SMOPSO	EM-MOPSO
ZDT1	Mean	<b>6.36E-04</b>	8.11E-04	6.72E-01	7.98E-02
	Std	<b>5.64E-05</b>	1.05E-04	1.74E-01	9.70E-02
ZDT2	Mean	<b>8.29E-04</b>	1.46E-03	1.22E-01	1.16E-03
	Std	4.53E-04	<b>1.21E-04</b>	1.56E-01	2.22E-04
ZDT3	Mean	4.85E-03	5.02E-03	3.73E-02	<b>3.58E-03</b>
	Std	3.29E-04	<b>2.88E-04</b>	4.15E-02	3.40E-04
ZDT4	Mean	<b>7.36E-04</b>	9.37E-04	9.66E+00	4.25E+01
	Std	6.95E-05	<b>6.85E-05</b>	2.46E+00	8.82E+00

表 2 不同算法在 ZDT 系列测试函数上的  $\Delta$  值Table 2  $\Delta$  values of different algorithms on ZDT series test functions

function	$\Delta$	ammmMOPSO	MOPSO	SMOPSO	EM-MOPSO
ZDT1	Mean	1.62E+00	1.02E+00	8.08E-01	<b>7.14E-01</b>
	Std	<b>4.99E-02</b>	7.00E-02	2.35E-01	5.19E-02
ZDT2	Mean	9.57E-01	1.07E+00	8.28E-01	<b>8.24E-01</b>
	Std	<b>3.20E-02</b>	5.20E-02	1.76E-01	3.39E-02
ZDT3	Mean	1.49E+00	1.51E+00	1.08E+00	<b>8.26E-01</b>
	Std	3.47E-02	<b>2.32E-02</b>	9.55E-02	8.47E-02
ZDT4	Mean	9.53E-01	1.03E+00	1.00E+00	<b>8.69E-01</b>
	Std	4.09E-02	<b>3.88E-02</b>	3.99E-02	7.08E-02

表 3 不同算法在 DTLZ 系列测试函数上的  $I_H$  值Table 3  $I_H$  values of different algorithms on DTLZ series test functions

function	$I_H$	ammmMOPSO	MOPSO	SMOPSO	EM-MOPSO
DTLZ2	Mean	<b>5.97E-01</b>	5.93E-01	5.93E-01	5.81E-01
	Std	3.55E-02	<b>3.25E-02</b>	5.21E-02	4.13E-02
DTLZ4	Mean	<b>5.94E-01</b>	5.78E-01	5.76E-01	5.89E-01
	Std	<b>4.38E-02</b>	4.67E-02	5.27E-02	4.66E-02
DTLZ5	Mean	<b>5.95E-01</b>	5.89E-01	5.93E-01	5.92E-01
	Std	4.23E-02	3.92E-02	4.75E-02	<b>3.66E-02</b>

从表 1 可知,在 4 个双目标测试函数上 ammmMOPSO 算法获得 3 个最优的 GD 平均值和 1 个最优的标准差。即使在 ZDT<sub>3</sub> 测试函数上 ammmMOPSO 算法的 GD 指标略低于 EMMOPSO 算法,但是比 MOPSO 算法和 SMOPSO 算法有着明显的收敛性优势;综上可知,ammmMOPSO 算法有着良好的收敛性。正是 ammmMOPSO 算法在全局样本的选取部分使用“加权和函数”去筛选同一网格的非劣解,以及对粒子的位置进行变异操作,使得该算法避开局部最优解,从而获得更好的最优解。

从表 2 可知,在 4 个双目标测试函数上 ammmMOPSO 算法获得 2 个最优的标准差和 2 个次优的标准差,说明 ammmMOPSO 算法所获得的  $\Delta$  数据比较稳定。在测试函数 ZDT<sub>2</sub> - ZDT<sub>4</sub> 上 ammmMOPSO 算法的  $\Delta$  平均值都要低于 MOPSO 算法;在测试函数 ZDT<sub>4</sub> 上 ammmMOPSO 算法的  $\Delta$  平均值同时低于 MOPSO 算法和 SMOPSO 算法;综上可知,ammmMOPSO 算法有较好的分布性。正是 ammmMOPSO 算法在外部存档中使用“加权和函数”与相邻距离的混合策略,使得 ammmMOPSO 算

法能够保留较好的非劣解并且删掉较差的非劣解,从而保障非劣解分布均匀。

从表 3 可知,在 3 个三目标测试函数上 ammmMOPSO 算法获得了 3 个最优的  $I_H$  平均值和 1 个最优的标准差。由此看出,比其他 3 种多目标粒子群算法表现更加优越。

为了直观地观察各算法所得的非劣解解集的收敛性和多样性,在图 4-图 7 中分别给出了在双目标测试函数 ZDT1-ZDT4 上求得的非劣解集与真实的帕累托前沿对比的图片。仿真实验结果表明,在 ZDT1 函数上,SMOPSO 算法的收敛性略显不足,ammmMOPSO 算法、MOPSO 算法和 EMMOPSO 算法有较好的收敛性;在 ZDT2 函数上,SMOPSO 算法表现出收敛性和多样性严重不足,ammmMOPSO 算法、MOPSO 算法和 EMMOPSO 算法都有较好的收敛性和多样性,其 ammmMOPSO 算法表现效果最佳;在 ZDT3 函数上,4 种算法都有较好的收敛性和多样性,其中 ammmMOPSO 算法表现最佳;在 ZDT4 函数上,SMOPSO 算法和 EMMOPSO 算法的多样性和收敛性都表现出严重不足,ammmMOPSO 算法和

MOPSO 算法的多样性和收敛性表现良好;其中 ammmMOPSO 算法表现最佳;综上可知, ammmMOPSO 算法在双目标测试函数 ZDT1~ZDT4 上更加接近真实的帕累托前沿且有着良好的分布。从图 8—图 10 可以看出, ammmMOPSO 算法比其他多目标粒子群算法具有更好的非劣解分布。

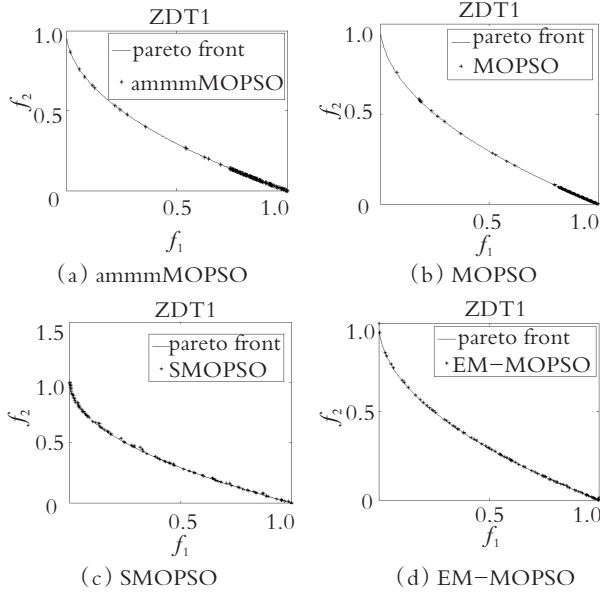


图 4 对于 ZDT1 函数 4 种不同算法的真实 Pareto 前沿效果图

Fig. 4 Real Pareto frontier effect of four different algorithms for ZDT1 functions

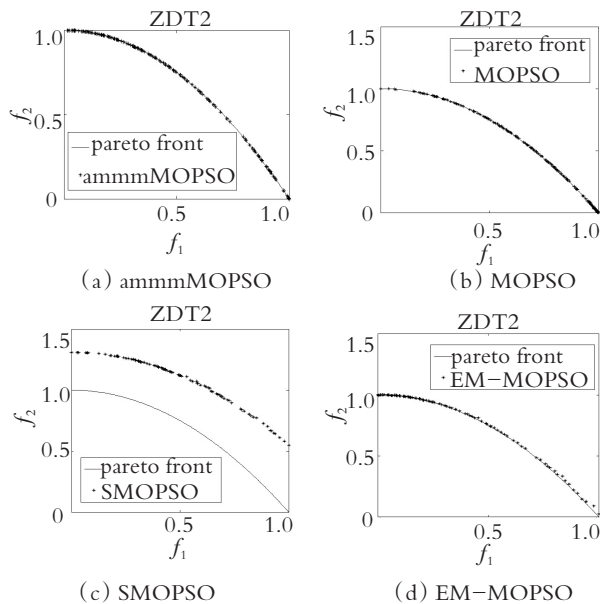


图 5 对于 ZDT2 函数 4 种不同算法的真实 Pareto 前沿效果图

Fig. 5 Real Pareto frontier effect of four different algorithms for ZDT2 functions

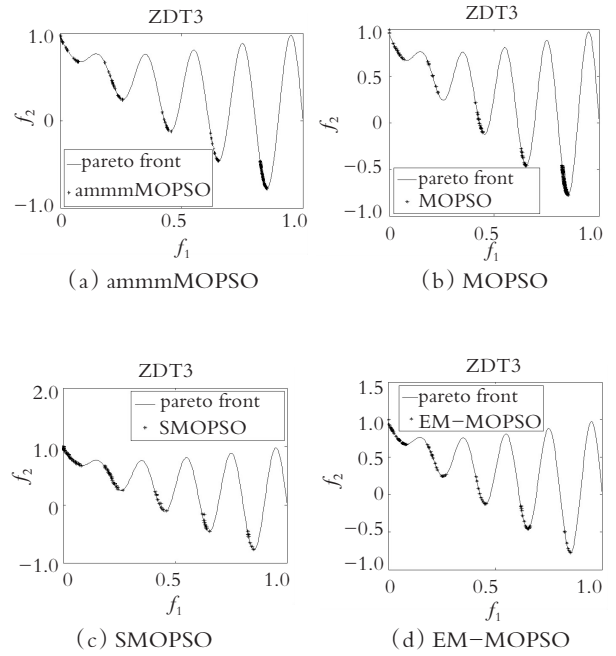


图 6 对于 ZDT3 函数 4 种不同算法的真实 Pareto 前沿效果图

Fig. 6 Real Pareto frontier effect of four different algorithms for ZDT3 functions

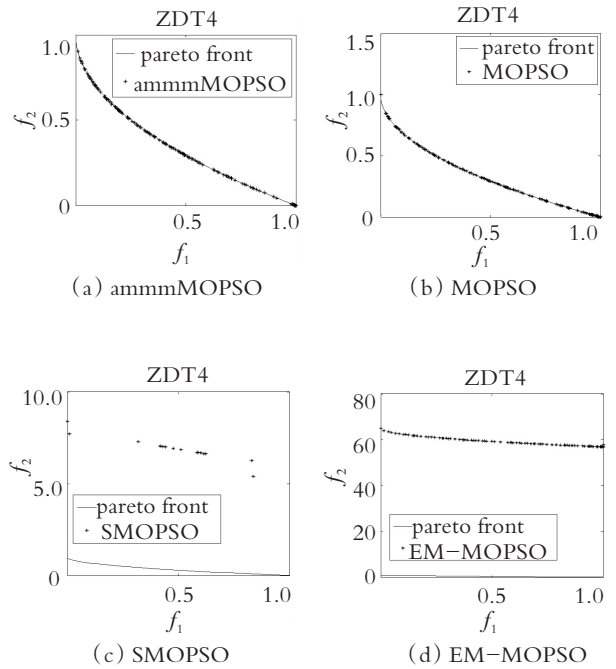


图 7 对于 ZDT4 函数 4 种不同算法的真实 Pareto 前沿效果图

Fig. 7 Real Pareto frontier effect of four different algorithms for ZDT4 functions

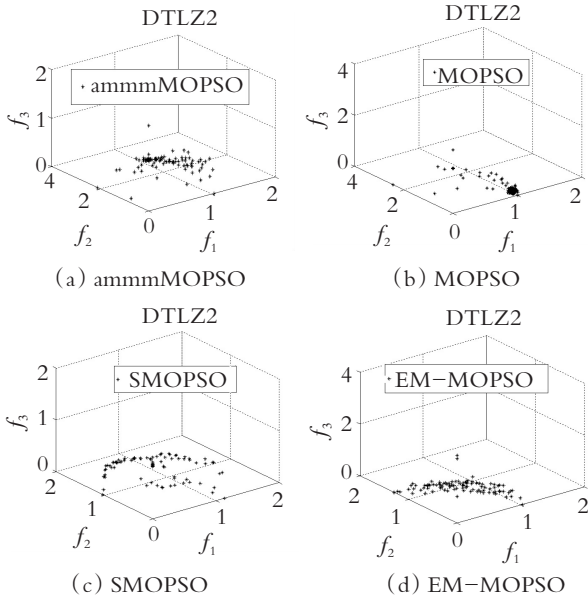


图 8 对于 DTLZ2 函数 4 种不同算法的 Pareto 前沿效果图

Fig. 8 Real Pareto frontier effect of four different algorithms for DTLZ2 function

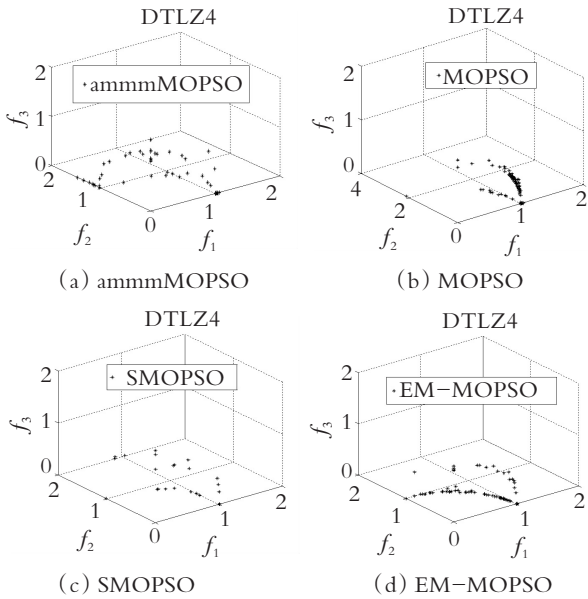


图 9 对于 DTLZ4 函数 4 种不同算法的 Pareto 前沿效果图

Fig. 9 Real Pareto frontier effect of four different algorithms for DTLZ4 functions

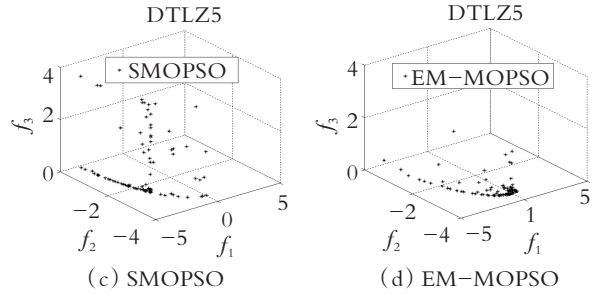
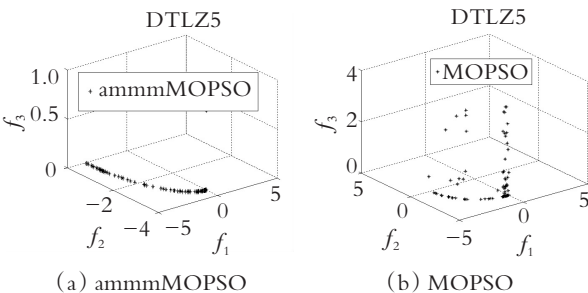


图 10 对于 DTLZ5 函数 4 种不同算法的 Pareto 前沿效果图

Fig. 10 Pareto frontier effect of four different algorithms for DTLZ5 functions

为了更直观地观察各种算法在双目标测试函数 ZDT1~ZDT4 和三目标测试函数 DTLZ2、DTLZ4 和 DTLZ5 上的实验数据分布,图 11~图 12 分别给出了 ammmMOPSO 算法和其他 3 种多目标粒子群算法在双目标测试函数 ZDT1~ZDT4 上的 GD 和  $\Delta$  指标的箱线图,图 13 给出了 ammmMOPSO 算法和其他 3 种多目标粒子群算法在三目标测试函数 DTLZ2、DTLZ4 和 DTLZ5 上  $I_H$  指标的箱线图,其中箱型图的横坐标 1~4 分别代表 ammmMOPSO 算法、MOPSO 算法、SMOPSO 算法和 EMMOPSO 算法。由图 11 可以看出,在求解这 4 种两目标测试函数时 ammmMOPSO 算法都比其他 3 种多目标粒子群算法有明显的优势,所获得的数据有良好的稳定性;由图 12 可以看出, ammmMOPSO 算法  $\Delta$  的波动范围都要优于其他 3 种多目标优化算法,说明 ammmMOPSO 算法所获得的数据更加稳定;由图 13 可知, ammmMOPSO 算法的  $I_H$  中值和波动范围都优于其他 3 种多目标算法;综上所述可知, ammmMOPSO 算法具有较好的综合性能。

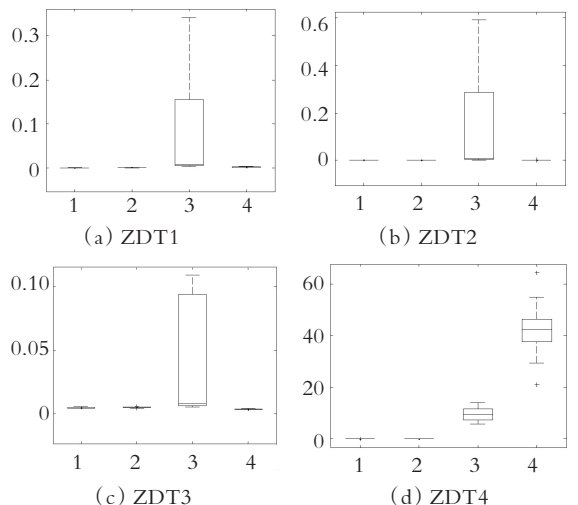
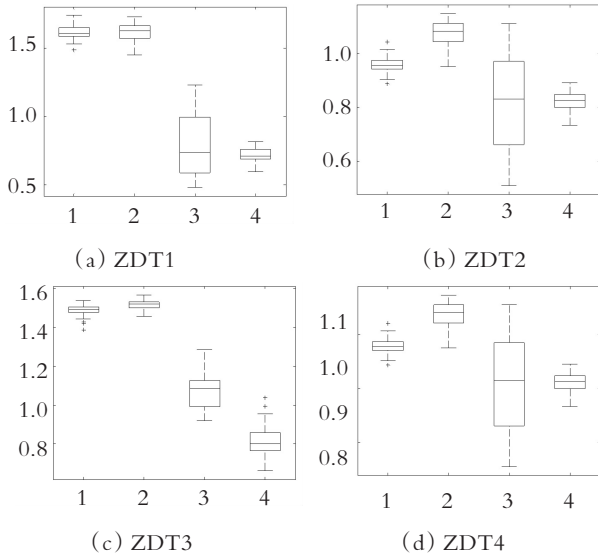
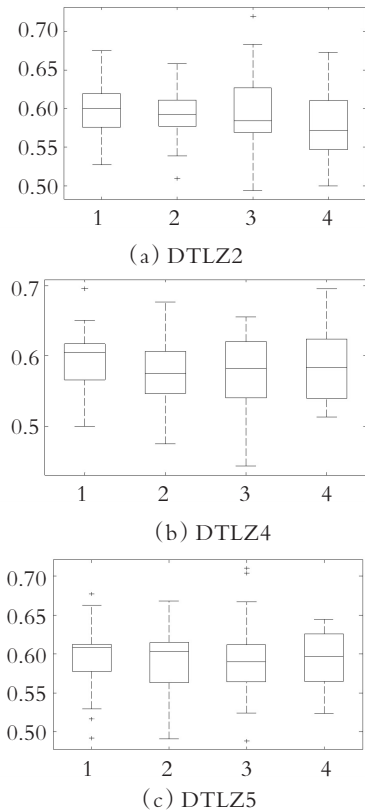


图 11 GD 值的盒状统计图

Fig. 11 Box-shaped statistical graph of GD value



图 12  $\Delta$  值的盒状统计图Fig. 12 Box-shaped statistical graph of  $\Delta$  value图 13  $I_H$  值的盒状统计图Fig. 13 Box-shaped statistical graph of  $I_H$  value

## 4 结束语

本文提出基于自适应网格混合机制的多目标粒子群算法, ammmMOPSO 算法对外部存档规模的维护和全局最优样本的选取提出新的策略。在全局最优样本的选取中本文使用加权和函数策略对外部存

档中的非劣解进行选取, 提升了 MOPSO 算法的分布性和收敛性; 对于外部存档的维护, 本文提出相邻距离和“加权和函数”的混合策略, 对密度最大的网格进行有针对性地删除 40% 的粒子数, 为选取全局最优解提供了较好的粒子质量; 为了防止后期算法停滞, 无法逃脱局部最优解, 提出了位置的变异操作, 以增强粒子的开发能力; 为证实 ammmMOPSO 算法的有效性, ammmMOPSO 算法、MOPSO 算法、SMOPSO 算法以及 EMMOPSO 算法采用双目标测试函数 ZDT1~ZDT4 和三目标测试函数 DTLZ2、DTLZ4 和 DTLZ5 进行仿真实验, 实验仿真结果表明, ammmMOPSO 算法具备较好的收敛性和分布性, 且具有良好的空间化效果。

## 参考文献 (References):

- [1] NAL A N, GULGUN K. Multi-objective particle swarm optimization with random immigrants [J]. *Complex & Intelligent Systems*, 2020, 6(3): 635—650.
- [2] CHAO Z, QING L, PENG C, et al. An improved multi-objective particle swarm optimization and its application in raw ore dispatching [J]. *Advances in Mechanical Engineering*, 2018, 10(2): 1—7.
- [3] 胡玉蝶, 张涛, 黄瑾, 等. 一种基于健康度的多目标粒子群算法 [J]. *长江大学学报(自然科学版)*, 2019, 16(10): 79—84.
- [4] HU Yu-die, ZHANG Tao, HUANG Jin, et al. A multi-objective particle swarm optimization algorithm based on health [J]. *Journal of Changjiang University (Natural Science Edition)*, 2019, 16(10): 79—84.
- [5] YEN G G, LEONG W F. Dynamic multiple swarms in multi-objective particle swarm optimization [J]. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, 2009, 39(4): 890—911.
- [6] 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.
- [6] LI X D. A non-dominated sorting for particle swarm optimizer for multi-objective optimization [C]// *In Proceedings of the International Conference on Genetic and Evolutionary Computation*, Atlanta, 2003, 37—48.

- [7] JANGA R M, NAGESH K D. An efficient multi-objective optimization algorithm based on swarm intelligence for engineering design[J]. *Engineering Optimization*, 2007, 39(1):49—68.
- [8] SCHUTZE O, ESQUIVEL X, LARA A, et al. Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2012, 16(4): 504—522.
- [9] SCOTT J R. Fault tolerant design using single and multicriteria genetic algorithm optimization [J]. *Cellular Immunology*, 1995, 37(1): 1—13.
- [10] 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.

## Multi-objective Particle Swarm Optimization Based on Adaptive Mesh Mixing Mechanism

ZOU Kang-ge, LIU Yan-min

(1. School of Mathematics and Statistics, Guizhou University, Guiyang 550025, China;

2. School of Mathematics, Zunyi Normal College, Guizhou Zunyi 563006, China)

**Abstract:** When particle swarm optimization (PSO) is used to solve multi-objective optimization problems, the PSO has a fast convergence effect, which makes the diversity of population in the optimization process insufficient and makes the algorithm converge early. In order to effectively design multi-objective particle swarm optimization algorithm, a multi-objective particle swarm optimization (MOPSO) algorithm based on adaptive mesh mixing mechanism is proposed. The algorithm adopts a dual maintenance strategy of adaptive grid and mixing mechanism to ensure the uniform distribution of non-inferior solutions in the external archive, and avoid rapid population degradation and affecting the development ability of particles. The weighted strategy in the mixing mechanism is used to determine the global optimal sample in the external non-inferior solution, which increases the diversity of population and improves the probability of particles flying to the real Pareto frontier. At the same time, in order to prevent the algorithm from stagnating and falling into the local optimal problem, a mutation operation is introduced to dynamically change the position of particles, which enhances the exploration ability of particles. The simulation results show that the proposed algorithm has better convergence and diversity and better spatial effect than the other three classical multi-objective particle swarm optimization algorithms.

**Key words:** multi-objective optimization; particle swarm optimization; adaptive mesh; mixed mechanism; mutation operation

责任编辑:罗珊珊

引用本文/Cite this paper:

邹康格,刘衍民. 基于自适应网格混合机制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(2): 14—23.  
ZOU Kang-ge, LIU Yan-min. Multi-objective particle swarm optimization based on adaptive mesh mixing mechanism[J]. *Journal of Chongqing Technology and Business University (Natural Science Edition)*, 2022, 39(2): 14—23.