

文章编号:1671-6833(2021)01-0009-06

两阶段搜索的多模态多目标差分进化算法

汪慎文<sup>1,2</sup>, 张佳星<sup>1,2</sup>, 褚晓凯<sup>1,2</sup>, 刘 枫<sup>3</sup>, 王 晖<sup>4</sup>

(1.河北地质大学 信息工程学院,河北 石家庄 050031; 2.河北地质大学 人工智能与机器学习研究室,河北 石家庄 050031; 3.中国信息通信研究院 泰尔终端实验室,北京 100191; 4.南昌工程学院 信息工程学院,江西 南昌 330099)

**摘 要:**在多模态多目标优化问题中,Pareto 前沿的同一位置对应决策空间的多个 Pareto 最优解,而已有的多目标优化算法往往只能获得其中的一个 Pareto 最优解,因此,提出一种两阶段搜索的多模态多目标差分进化算法。该算法将优化过程分为精英搜索和分区搜索两个阶段:在精英搜索阶段通过精英变异策略生成高质量个体来保障种群的搜索精度和效率;在分区搜索阶段将决策空间分为若干子空间,利用已探测到的种群对各个子空间进行深度探索,降低问题复杂度的同时提高种群在决策空间的扩展性和均匀性。在 MMF1 等 18 个多模态多目标优化测试函数上与 NSGAI、MO\_Ring\_PSO\_SCD、DN-NSGAI、Omni-Optimizer、MMODE 5 种经典算法进行性能比较。实验结果表明,本文算法在帕累托近似性(PSP)性能指标上有 16 个测试函数优于其他 5 个对比算法。

**关键词:**多模态多目标优化;差分进化算法;两阶段搜索;精英变异;分区搜索

**中图分类号:** TP301      **文献标志码:** A      **doi:**10.13705/j.issn.1671-6833.2021.01.002

0 引言

在现实工程中,许多优化问题需要同时满足多个目标,并且这些目标往往相互冲突,这类问题称为多目标优化问题(multi-objective optimization problem,MOP)<sup>[1]</sup>。在过去的研究中,人们提出了各种多目标进化算法(multi-objective evolutionary algorithm,MOEA)<sup>[2]</sup>。这些算法通常可以很好地在目标空间搜索到完整且分布均匀的 Pareto 前沿面,为决策者提供更合适的选择。但在实际优化问题中通常会出现这样的情况:两个或多个不同的 Pareto 最优解对应同一 Pareto 前沿位置,这类问题被称为多模态多目标优化问题(multimodal multi-objective optimization problem,MMOP)<sup>[3]</sup>。虽然找到 Pareto 前沿面就可以满足优化要求,但是获得更多 PS(Pareto 最优解集)可以为决策者提供更多的备选方案<sup>[4]</sup>。现实中的出行计划就是一个很好的例子<sup>[5]</sup>,如图 1 所示。图 1 中的方案 1 和方案 4 分别代表火车和长途汽车出行,方案 2 和方案 3 同为飞机出行,但这两个方案中转站不同。根据图 1 右

侧 PF(Pareto 前沿)可以看出,方案 1 与方案 4,方案 2 与方案 3 分别具有相同的时间和成本。很明显,对于需要更短时间到达目的地的乘客,有方案 2 和方案 3 这两个选择。当某一方案的中转站受到天气等因素影响时,乘客完全可以选择备用方案;在相对成本较低的情况下,乘客可能不想换乘而选择长途汽车,或者乘客在途中有工作需要处理,在火车上工作会更方便一些。如果最终结果只能给每种乘客提供一个 PS,则不能满足不同乘客的需求。因此,研究如何同时获得更多的 PS 从而为决策者提供更多的选择具有很重要的现实意义<sup>[5]</sup>。

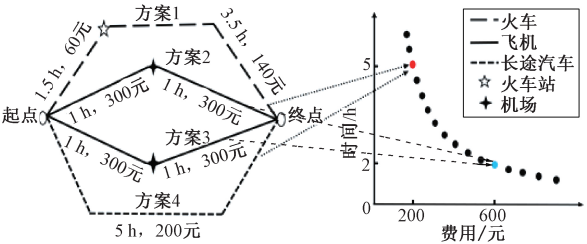


图 1 出行方案及其所需时间和费用  
Figure 1 Travel plan with needing time and cost

近年来,研究者们提出了大量多模态多目标进化算法(multimodal multi-objective evolutionary algorithms, MMEA)来求解多模态多目标优化问题。根据其算法特征,大致可分为以下几类:①基于拥挤距离和非支配排序的 MMEA 算法,这类算法引入决策空间的拥挤距离用于非支配排序,如 Omni-optimizer<sup>[6]</sup>和 DN-NSGAI<sup>[7]</sup>等;在多目标优化问题中可以有效搜索到更多的解,但在处理 PS 形状较为复杂的问题时表现并不理想。②基于小生境技术的 MMEA 算法,其核心思想是将种群个体分为多类,通过竞争选出适应度高的个体来维持种群的多样性,如 DNEA<sup>[8]</sup>、Niching-CMA<sup>[9]</sup>等。这类算法可以有效地保持种群的多样性,但由于环境选择压力较大,随着种群的进化,种群的多样性不可避免地会恶化<sup>[10]</sup>。③基于新型进化范例的 MMEA 算法,采用性能优越的 MOEA 来求解 MMOP,如 SS-MOPSO<sup>[11]</sup>、MMODE<sup>[4]</sup>等。这些算法运行效率高,易于理解,并且在提高种群多样性的同时可以更好地保护解在目标空间的分布。④基于不同策略的 MMEA 集成算法,这类算法将 MMEA 中不同策略优势互补集成到一起,进一步提高算法的性能,如 MO\_Ring\_PSO\_SCD<sup>[3]</sup>、MMOCLRPSO<sup>[12]</sup>、MMOPIO<sup>[13]</sup>等。目前,这类集成算法是研究者们关注的重点,并且在处理多模态多目标问题时取得了很好的效果。

## 1 两阶段搜索的多模态多目标差分进化算法(TS-MMODE)

针对多模态多目标优化问题,本文提出了一种两阶段搜索的多模态多目标差分进化算法(multimodal multi-objective differential evolution algorithm based on two-stage search, TS-MMODE)。将算法的进化过程分为精英搜索和分区搜索两个阶段,使用不同的变异策略求解。该算法还引入了文献[3]中基于决策空间的非支配排序策略来提高解在决策空间的多样性。

### 1.1 第1阶段——精英搜索阶段

#### 1.1.1 特殊拥挤距离

Yue 等<sup>[3]</sup>提出同时考虑决策空间和目标空间计算每个个体的特殊拥挤距离,其描述见式(1):

$$SCD_i = \begin{cases} \max(CD_{i,x}, CD_{i,f}), \\ CD_{i,x} > CD_{avg,x} \text{ 或 } CD_{i,f} > CD_{avg,y}; \\ \min(CD_{i,x}, CD_{i,f}), \text{其他。} \end{cases} \quad (1)$$

式中:  $CD_{i,x}$ 、 $CD_{i,f}$  分别表示第  $i$  个个体在决策空间和目标空间的拥挤距离;  $CD_{avg,x}$ 、 $CD_{avg,y}$  分别表示决策空间和目标空间的平均拥挤距离。

#### 1.1.2 精英变异策略

在精英搜索阶段根据非支配排序和决策空间的拥挤距离选取种群的最优的一半作为精英变异池,采用差分进化算法中的 DE/rand/2 变异策略进行变异,这样可以有效提高解的质量,加快收敛速度。差分变异策略 DE/rand/2 定义见式(2):

$$V_i(t) = x_{r1}(t) + F(x_{r2}(t) - x_{r3}(t)) + F(x_{r4}(t) - x_{r5}(t)). \quad (2)$$

式中:  $x_{r1}$ 、 $x_{r2}$ 、 $x_{r3}$ 、 $x_{r4}$ 、 $x_{r5}$  是从变异池中随机选取的互不相同的5个个体;  $i=1,2,\dots,N$ ,  $N$  为种群规模;  $V_i(t)$  表示第  $t$  代的第  $i$  个个体;  $F$  为 DE 的缩放因子,取值为  $[0,1]$ 。

#### 1.1.3 越界处理方法

经变异后的个体可能会落在决策空间以外,为此学者提出众多的修补方法,如文献[4]和文献[14]。本文提出一种新的越界处理方法,给首次越界的个体第2次变异的机会,变异方式如式(3)所示:

$$V_{i,j}(t) = x_{r1,j}(t) - F(x_{r2,j}(t) - x_{r3,j}(t)) - F(x_{r4,j}(t) - x_{r5,j}(t)). \quad (3)$$

如果第2次变异后个体仍然越界,则按照式(4)进行修补:

$$v_{i,j} = \begin{cases} \min\{U_j, 2L_j - v_{i,j}(t)\}, v_{i,j}(t) < L_j; \\ \max\{L_j, 2U_j - v_{i,j}(t)\}, v_{i,j}(t) > U_j. \end{cases} \quad (4)$$

式中:  $v_{i,j}$  表示第  $i$  个个体第  $j$  维上的值;  $U_j$ 、 $L_j$  表示决策空间的上下界。

#### 1.1.4 精英搜索阶段实现步骤

**Step 1** 精英变异池更新:通过非支配排序和决策空间拥挤距离从种群  $P$  中选取最优的一半个体放入精英变异池。

**Step 2** 变异操作:通过式(2)对精英变异池中的个体进行变异。

**Step 3** 越界操作:执行按照式(3)、(4)。

**Step 4** 杂交操作:执行二项式交叉操作形成种群  $UP$ 。

**Step 5** 选择操作:合并种群  $P$  和种群  $UP$  后,利用非支配排序和特殊拥挤距离式(1)更新种群  $P$ 。

## 1.2 第2阶段——分区搜索阶段

### 1.2.1 分区搜索方法

Fan 等<sup>[15]</sup>指出:搜索空间和目标空间共同决定了问题的复杂性。然而,在现实中很难根据目

标空间降低问题的复杂性。因此,将搜索空间划分为若干子空间是降低问题复杂度的一种很有前途的方法。

在处理多模态多目标优化问题时,分区搜索的方法操作简单,且不会因种群环境过于复杂而出现恶化<sup>[10]</sup>,可以在降低问题复杂度的同时有效地帮助算法在决策空间进行广泛搜索。本文采用的分区方法是从一个  $D(D \geq 2)$  维优化问题中随机选取  $X_a, X_b$  两个变量,并将  $X_a$  划分为  $Q$  个分段,将  $X_b$  划分为  $H$  个分段,从而把整个  $D$  维空间划分为  $Q \times H$  个子空间。以 3 维空间为例,取  $Q = 2, H = 2$  可以将决策空间划分为 4 个子空间,如图 2 所示。

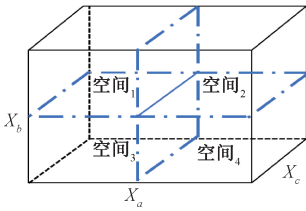


图 2 3 维 4 空间分区示意图

Figure 2 3-D and 4 spatial divisions diagram

#### 1.2.2 分区搜索阶段变异策略

在算法的分区搜索阶段,为了在已有一定分布性且适应度值较好的种群内进行更广泛的搜索,将分区后子空间的全部个体作为变异池,通过式(2)来产生下一代个体。

#### 1.2.3 分区搜索阶段实现步骤

**Step 1 分区操作:** 将种群个体放入划分好的空间  $S_k$  中,  $k = 1, 2, \dots, N_s$ 。其中  $N_s$  表示子空间个数。

**Step 2 变异操作:** 依式(2)对子空间中个体进行变异。

**Step 3 越界操作:** 执行按照式(3)、(4)。

**Step 4 杂交操作:** 执行二项式交叉操作形成种群  $UP$ 。

**Step 5 选择操作:** 利用非支配解排序和特殊拥挤距离式(1)更新种群。

### 1.3 TS-MMODE 算法实现步骤

**Step 1 初始化 TS-MMODE 参数以及种群,** 阶段值为  $ti$ , 子空间个数为  $N_s$ 。

**Step 2 对种群进行精英搜索。**

**Step 3 判断当前迭代次数是否小于阶段值  $ti$ 。** 若小于则跳转至 Step 2 继续执行;否则执行 Step 4。

**Step 4 对种群进行分区搜索。**

**Step 5 判断是否达到最大适应度评估次数,** 若没有达到则跳转至 Step 4 继续执行;否则执行 Step 6。

**Step 6 输出结果。**

## 2 实验条件

### 2.1 测试函数

本文共采用 18 个多模态多目标测试函数<sup>[3]</sup>来验证算法的性能表现,分别是 MMF1~MMF12、MMF13~MMF15 ( $n = 3$ )、Omni-test ( $n = 3$ )、SYM-PART\_simple 和 SYM-PART\_rotated。

### 2.2 评价指标

本文使用多模态多目标算法常用的评价指标:帕累托近似性 (PSP)<sup>[3]</sup>,用来评估获取的 PS 与实际 PS 之间的相似程度。

### 2.3 参数设置

在所有实验中,比较算法的参数设置均与原文献[3,4,6,7,16]参数相同,每个算法每个实验均独立运行 30 次,种群规模为 800,最大评估次数为  $1.6 \times 10^5$  次。

本文算法 TS-MMODE 涉及的参数有种群规模  $N$ 、阶段值  $ti$ 、子空间个数  $N_s$ 、缩放因子  $F$ 、交叉概率  $Cr$ 。在本次实验中取  $F = 0.5, Cr = 0.9$ 。针对阶段值  $ti$  和子空间个数  $N_s$  的设置会在第 3 节进行讨论。

## 3 实验结果及分析

### 3.1 阶段值敏感性分析

由于本文提出的算法将进化过程分为精英搜索和分区搜索两个阶段,算法何时停止精英搜索阶段进入分区搜索阶段由阶段值  $ti$  决定。表 1 中的实验取阶段值  $ti$  分别为 1、50、100 和 150,  $ti = 1$  即表示算法在一开始就进行了分区。

表 1 展示了在 18 个测试函数中,不同的阶段值  $ti$  对 TS-MMODE 算法的 PSP 的影响。从表 1 中可以看出,有 9 个函数在  $ti$  取 100 时效果最好;在 MMF5、MMF12 和 MMF14 上  $ti$  取 50 时算法表现最好;  $ti$  取 150 时算法在 MMF4、MMF7、MMF10、Omni-test ( $n = 3$ )、SYM-PART\_simple 和 SYM-PART\_rotated 上表现最为理想;而当  $ti = 1$  时,与其他 3 种取值相比,在 18 个测试函数中没有任何一个可以取得最优结果。所以此实验证明分两阶段搜索在 TS-MMODE 算法中是很有必要的。

### 3.2 子空间个数敏感性分析

表 2 展示了当阶段值  $ti = 100$  时,子空间个数  $N_s$  分别为 2、4、6、8、9 时对算法的影响。实验结果表明,在 18 个测试函数中有 9 个函数在  $N_s =$

4 时效果最好;有 6 个函数在  $N_s = 2$  时表现更为理想,且除 SYM-PART\_rotated 外的其他 5 个函数与  $N_s = 4$  时的结果相近。总体而言,对于 TS-MMODE 算法,当  $N_s = 4$  时效果是最佳的。此外,阶段值  $t_i$  和子空间个数  $N_s$  的不同取值对 PF

的影响很小,因此不再进行讨论。

### 3.3 与经典多模态多目标优化算法性能比较

如表 3 所示,为了进一步验证 TS-MMODE 算法的性能,本文引入常见的多模态多目标优化算法 MO\_Ring\_PSO\_SCD<sup>[3]</sup>、DN-NSGAII<sup>[7]</sup>、Omni-

表 1 TS-MMODE 取不同阶段值时的 PSP 值

Table 1 PSP value when TS-MMODE takes different stage values

| 测试函数             | PSP 值     |                  |                |                  |
|------------------|-----------|------------------|----------------|------------------|
|                  | $t_i = 1$ | $t_i = 50$       | $t_i = 100$    | $t_i = 150$      |
| MMF1             | 80. 39    | 96. 55           | <b>96. 97</b>  | 94. 74           |
| MMF2             | 211. 61   | 431. 81          | <b>478. 38</b> | 431. 26          |
| MMF3             | 181. 99   | 337. 95          | <b>405. 11</b> | 389. 76          |
| MMF4             | 121. 22   | 139. 00          | 140. 12        | <b>144. 62</b>   |
| MMF5             | 35. 95    | <b>43. 59</b>    | 43. 58         | 43. 04           |
| MMF6             | 40. 93    | 43. 34           | <b>43. 50</b>  | 43. 38           |
| MMF7             | 144. 29   | 147. 76          | 148. 92        | <b>150. 06</b>   |
| MMF8             | 30. 35    | 65. 47           | <b>72. 85</b>  | 72. 58           |
| MMF9             | 576. 90   | 580. 17          | <b>582. 03</b> | 577. 18          |
| MMF10            | 2 254. 30 | 2 328. 90        | 2 330. 20      | <b>2 349. 50</b> |
| MMF11            | 681. 95   | 680. 29          | <b>684. 03</b> | 681. 58          |
| MMF12            | 1 484. 20 | <b>1 508. 10</b> | 1 496. 40      | 1 505. 30        |
| MMF13            | 56. 07    | 67. 86           | <b>70. 68</b>  | 70. 27           |
| MMF14            | 32. 36    | <b>32. 40</b>    | 32. 35         | 32. 27           |
| MMF15            | 41. 60    | 41. 48           | <b>41. 71</b>  | 41. 62           |
| Omni-test        | 7. 98     | 39. 11           | 43. 38         | <b>47. 05</b>    |
| SYM-PART_rotated | 5. 54     | 2. 57            | 3. 36          | <b>6. 22</b>     |
| SYM-PART_simple  | 17. 42    | 66. 06           | 67. 30         | <b>69. 20</b>    |

表 2 TS-MMODE 子空间个数不同时的 PSP 值

Table 2 PSP value with different number of TS-MMODE subspaces

| 测试函数             | PSP 值            |                  |                |                |               |
|------------------|------------------|------------------|----------------|----------------|---------------|
|                  | $N_s = 2$        | $N_s = 4$        | $N_s = 6$      | $N_s = 8$      | $N_s = 9$     |
| MMF1             | <b>97. 07</b>    | 96. 97           | 96. 39         | 95. 61         | 95. 69        |
| MMF2             | 449. 88          | 478. 38          | 534. 67        | <b>574. 76</b> | 461. 98       |
| MMF3             | 474. 23          | 405. 11          | <b>476. 12</b> | 474. 40        | 423. 93       |
| MMF4             | <b>143. 50</b>   | 140. 12          | 117. 98        | 106. 99        | 107. 50       |
| MMF5             | 43. 09           | <b>43. 58</b>    | 42. 59         | 42. 25         | 41. 97        |
| MMF6             | 43. 36           | <b>43. 50</b>    | 42. 72         | 42. 66         | 42. 67        |
| MMF7             | 147. 19          | <b>148. 92</b>   | 147. 25        | 147. 71        | 144. 02       |
| MMF8             | 72. 77           | <b>72. 85</b>    | 67. 63         | 68. 29         | 45. 27        |
| MMF9             | 579. 14          | <b>582. 03</b>   | 568. 62        | 547. 60        | 557. 71       |
| MMF10            | 2 321. 60        | <b>2 330. 20</b> | 2 289. 30      | 2 311. 80      | 2 312. 10     |
| MMF11            | 681. 16          | <b>684. 03</b>   | 671. 83        | 666. 90        | 665. 95       |
| MMF12            | <b>1 507. 50</b> | 1 496. 40        | 1 474. 80      | 1 445. 20      | 1 475. 10     |
| MMF13            | 68. 19           | <b>70. 68</b>    | 68. 12         | 67. 28         | 69. 25        |
| MMF14            | <b>32. 37</b>    | 32. 35           | 32. 07         | 31. 90         | 31. 59        |
| MMF15            | 41. 31           | <b>41. 71</b>    | 41. 49         | 41. 07         | 40. 86        |
| Omni-test        | 46. 24           | 43. 38           | 46. 51         | 45. 61         | <b>50. 38</b> |
| SYM-PART_rotated | <b>41. 32</b>    | 3. 36            | 2. 81          | 3. 86          | 5. 32         |
| SYM-PART_simple  | <b>69. 50</b>    | 67. 30           | 58. 97         | 56. 37         | 69. 23        |

Optimizer<sup>[6]</sup>、MMODE<sup>[4]</sup> 和经典多目标算法 NS-GAII<sup>[16]</sup>,与本文算法的 *PSP* 值来进行比较。

由表 3 可以看出,本文提出的 TS-MMODE 算法在 18 个测试函数中有 16 个函数的结果优于其他 5 个算法,虽然在 MMF11 中效果不如 NSGAII,但是也高于其他 4 种多模态多目标优化算法。另外,与 MMODE 算法的比较中可以看出,在 18 个

测试函数中仅 SYM-PART\_rotated 这一函数的结果不如 MMODE。

图 3 展示了 TS-MMODE 算法在 *PSP* 值提升较大的 MMF2、MMF3、MMF7、MMF9、MMF10、MMF12 这 6 个测试函数上获得的 PF 与真实 PF。从图 3 中可以看出,TS-MMODE 算法并没有因注重决策空间分布而损坏 PF。

表 3 6 种算法在 18 个测试函数上的 *PSP* 值  
Table 3 *PSP* values of 6 algorithms on 18 test functions

| 测试函数             | <i>PSP</i> 值  |                 |           |                |              |                 |
|------------------|---------------|-----------------|-----------|----------------|--------------|-----------------|
|                  | NSGAII        | MO_Ring_PSO_SCD | DN-NSGAII | Omni-Optimizer | MMODE        | TS-MMODE        |
| MMF1             | 45.40         | 72.55           | 62.79     | 63.35          | 68.55        | <b>96.97</b>    |
| MMF2             | 147.19        | 173.56          | 90.17     | 113.00         | 287.89       | <b>478.38</b>   |
| MMF3             | 179.70        | 208.68          | 66.74     | 76.49          | 327.54       | <b>405.11</b>   |
| MMF4             | 58.02         | 122.06          | 33.75     | 34.00          | 120.37       | <b>140.12</b>   |
| MMF5             | 20.61         | 35.87           | 15.13     | 15.10          | 32.99        | <b>43.58</b>    |
| MMF6             | 18.22         | 38.84           | 16.61     | 17.24          | 24.78        | <b>43.50</b>    |
| MMF7             | 84.85         | 110.10          | 106.06    | 105.70         | 123.71       | <b>148.92</b>   |
| MMF8             | 1.58          | 54.54           | 12.74     | 12.38          | 60.07        | <b>72.85</b>    |
| MMF9             | 4.59          | 391.97          | 8.53      | 12.17          | 504.71       | <b>582.03</b>   |
| MMF10            | 1 761.50      | 306.45          | 2 025.80  | 2 009.00       | 1 635.60     | <b>2 330.20</b> |
| MMF11            | <b>761.63</b> | 533.74          | 521.74    | 547.47         | 683.89       | 684.03          |
| MMF12            | 978.54        | 670.72          | 958.23    | 1 101.96       | 1 343.30     | <b>1 496.40</b> |
| MMF13            | 36.37         | 52.67           | 23.78     | 28.35          | 49.03        | <b>70.68</b>    |
| MMF14            | 27.84         | 30.26           | 15.87     | 17.35          | 29.25        | <b>32.35</b>    |
| MMF15            | 39.51         | 36.38           | 24.86     | 26.62          | 40.10        | <b>41.71</b>    |
| Omni-test        | 7.13          | 10.22           | 1.12      | 0.87           | 43.32        | <b>43.38</b>    |
| SYM-PART_rotated | 3.29          | 25.56           | 0.88      | 0.50           | <b>59.68</b> | 3.36            |
| SYM-PART_simple  | 0.15          | 32.33           | 0.30      | 0.29           | 64.88        | <b>67.30</b>    |

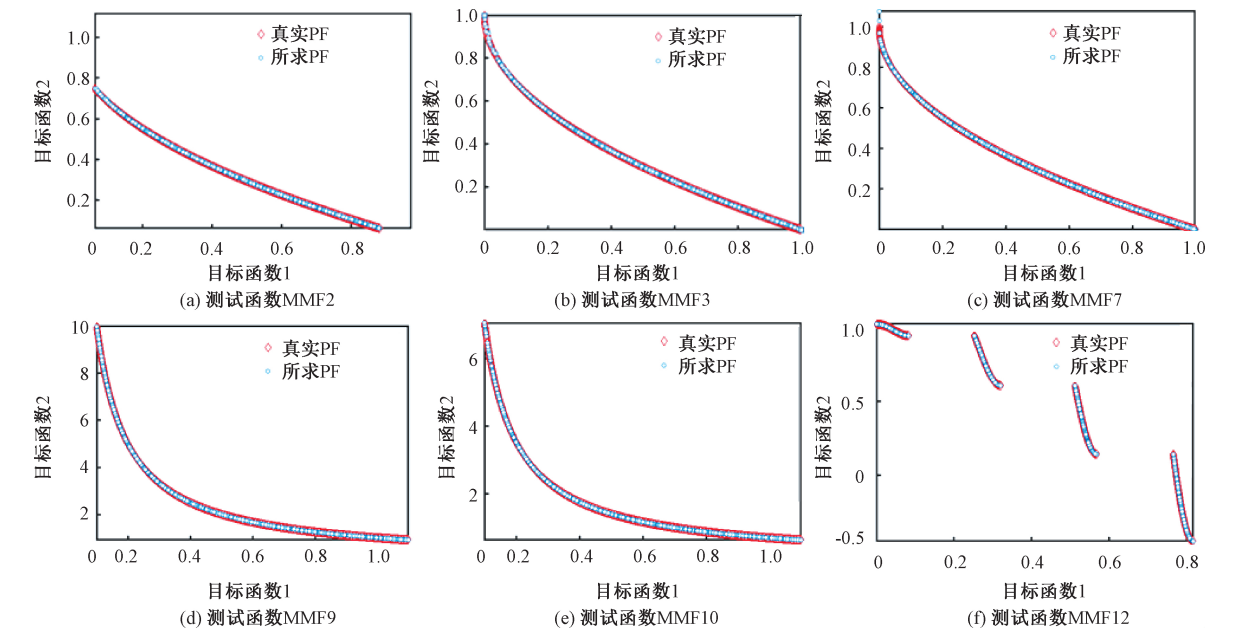


图 3 TS-MMODE 算法求得的 PF 与真实 PF  
Figure 3 Obtained PF by TS-MMODE algorithm and true PF

综上所述,本文提出的 TS-MMODE 算法在 PF 分布均匀的情况下大大提高了测试函数的 *PSP* 值,这主要是因为算法在精英搜索阶段有效地保证了解在目标空间的质量。在分区搜索阶段,对决策空间进行分区虽然很难提高 PF 的质量,但有效提高了解在决策空间的多样性和分布性,特别是对于一些 PS 几何结构相对复杂的测试函数,对决策空间进行分区可以更有效地定位解的位置并进行深度搜索。

## 4 结论

针对多模态多目标优化问题,提出了一种两阶段搜索的多模态多目标差分进化算法(TS-MMODE),算法将进化过程分解为以发散搜索为主的精英搜索阶段和以深度搜索为主的分区搜索阶段。在精英搜索阶段,通过精英变异策略增加种群多样性;在分区搜索阶段,通过对决策空间的划分实现对子空间的深度搜索。和常见的 5 种经典算法进行了比较,实验结果表明,TS-MMODE 算法在 18 个多模态多目标测试函数中有 16 个函数的 *PSP* 性能指标均优于其他 5 个算法,这表明本文提出的算法可以在维持目标空间解多样性的同时在决策空间搜索到分布更均匀的 PS。

## 参考文献:

- [1] 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] 闫李,李超,柴旭朝,等.基于多学习多目标鸽群优化的动态环境经济调度[J].郑州大学学报(工学版),2019,40(4):8-14.
- [3] YUE C T, QU B Y, LIANG J. A multi-objective particle swarm optimizer using ring topology for solving multimodal multi-objective problems[J]. IEEE transactions on evolutionary computation, 2017, 22(5): 805-817.
- [4] LIANG J, XU W W, YUE C T, et al. Multimodal multi-objective optimization with differential evolution[J]. Swarm and evolutionary computation, 2019, 44: 1028-1059.
- [5] WANG Y, YANG Z L, GOU Y J, et al. A novel multi-objective competitive swarm optimization algorithm for multi-modal multi objective problems[C]//IEEE Congress on Evolutionary Computation (CEC). New York: IEEE, 2019:271-278.
- [6] DEB K, TIWARI S. Omni-optimizer: a procedure for single and multi-objective optimization[C]// Evolutionary Multi-criterion Optimization, Third International Conference. Berlin: Springer, 2005: 47-61.
- [7] LIANG J J, YUE C T, QU B Y. Multimodal multi-objective optimization: a preliminary study[C]//IEEE Congress on Evolutionary Computation (CEC). New York: IEEE, 2016: 2454-2461.
- [8] LIU Y, ISHIBUCHI H, NOJIMA Y, et al. A double-niched evolutionary algorithm and its behavior on polygon-based problems[C]// International Conference on Parallel Problem Solving from Nature. Berlin: Springer, 2018: 262-273.
- [9] SHIR O M, PREUSS M, NAUJOKS B, et al. Enhancing decision space diversity in evolutionary multiobjective algorithms[C]// International Conference on Evolutionary Multi-Criterion Optimization. Berlin: Springer, 2009:95-109.
- [10] FAN Q Q, YAN X F. Solving multimodal multi-objective problems through zoning search[J]. IEEE transactions on systems, man, and cybernetics: systems, 2019:1-12.
- [11] QU B Y, LI Y, LIANG J, et al. A self-organized speciation based multi-objective particle swarm optimizer for multimodal multi-objective problems[J]. Applied soft computing, 2019, 86: 105886.
- [12] ZHANG W Z, LI G Q, ZHANG W W, et al. A cluster based PSO with leader updating mechanism and ring-topology for multimodal multi-objective optimization[J]. Swarm and evolutionary computation, 2019, 50: 100569.
- [13] HU Y, WANG J, LIANG J, et al. A self-organizing multimodal multi-objective pigeon-inspired optimization algorithm[J]. Science China information sciences, 2019,62(7):69-85.
- [14] 汪慎文,丁立新,张文生,等.差分进化算法研究进展[J].武汉大学学报(理学版),2014,60(4):283-292.
- [15] FAN Q Q, LI N, ZHANG Y L, et al. Zoning search using a hyper-heuristic algorithm[J]. Science China information sciences, 2019,62(9):189-191.
- [16] DEB K, AGRAWAL S, PRATAP A, et al. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II[C]// International Conference on Parallel Problem Solving from Nature. Berlin: Springer, 2000:849-858.

(下转第 110 页)

Delimitation and Evolution Pattern of Chinese Airports' Aviation  
Logistics Space Radiation Scopes

LI Yumin, LIU Yang, WANG Xianshun

(School of Management Engineering, Zhengzhou University, Zhengzhou 450001, China)

**Abstract:** It was significant to delimit reasonably the space radiation scope of airport aviation logistics and to study on the evolution pattern of airports in the development and decision-making of airport. The principal component analysis method was used to evaluate the aviation logistics comprehensive strength of 19 large airports in 2007, 2012 and 2017 in China. On the basis of this, the weighted Voronoi diagram method based on the breaking point model was adopted to divide the aviation logistics radiation scope of each airport, which revealed their evolution pattern and influencing factors of evolution in recent years. The paper drew the following conclusions: China's eastern and southern airports had strong comprehensive strength in aviation logistics, that of western and northern regions was weak. In recent years, most of the airports that their aviation logistics radiation ranges expanded, were located in the central and western regions of China. Most of the airports that their radiation ranges shrank and had no significant change were located in the southeast coast, north and northeast of China. The overall economic and export-oriented industry development of the city where the airport was located had gradually became key factors affecting the development of airport aviation logistics.

**Key words:** aviation logistics; space radiation scope; evolution pattern; principal component analysis method; weighted Voronoi diagram

(上接第 14 页)

Multimodal Multi-objective Differential Evolution Algorithm Based  
on Two-stage Search

WANG Shenwen<sup>1,2</sup>, ZHANG Jiaying<sup>1,2</sup>, CHU Xiaokai<sup>1,2</sup>, LIU Hong<sup>3</sup>, WANG Hui<sup>4</sup>

(1. School of Information Engineering, Hebei GEO University, Shijiazhuang 050031, China; 2. Laboratory of Artificial Intelligence and Machine Learning, Hebei GEO University, Shijiazhuang 050031, China; 3. Tel Terminal Laboratory, China Academy of Information and Communication, Beijing 100191, China; 4. School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, China)

**Abstract:** In multimodal multi-objective optimization problem, the same position of Pareto front often corresponded to multiple Pareto optimal solutions in decision space. However, the existing multi-objective optimization algorithms could only obtain one of the Pareto optimal solutions. Therefore, in this paper, a two-stage search multimodal multi-objective differential evolution algorithm was proposed, which divided the optimization process into two stages: elite search and partition search. In the elite search stage, elite mutation strategy was used to generate high-quality individuals to ensure the search accuracy and efficiency of the population. In the stage of partition search, the decision space was divided into several subspaces, and the detected population was used to explore each subspace in depth, so as to reduce the complexity of the problem and to improve the expansion and uniformity of the population in the decision space. The performance of the algorithm was compared with five classical algorithms NSGAI, MO\_Ring\_PSO\_SCD, DN-NSGAI, Omni-Optimizer, MMODE on 18 multimodal and multi-objective optimization test functions, such as MMF1. Experimental results showed that there were 16 test functions in the performance index of Pareto approximation (PSP) of the proposed algorithm, which were better than the other five comparison algorithms.

**Key words:** multimodal multi-objective optimization; differential evolution; two-stage search; elite variation; partition search