



# 基于全局密度更新策略的两阶段多模态多目标进化算法

贺娟娟<sup>1,2</sup>, 杨倩<sup>1,2</sup>, 许志伟<sup>1,2</sup>, 张凯<sup>1,2</sup>, 张兴义<sup>3</sup>, 葛明峰<sup>4\*</sup>

1. 武汉科技大学计算机科学与技术学院, 武汉 430065

2. 武汉科技大学智能信息处理与实时工业系统湖北省重点实验室, 武汉 430065

3. 安徽大学计算机科学与技术学院, 合肥 230093

4. 中国地质大学(武汉)机械与电子信息学院, 武汉 430074

\* 通信作者. E-mail: gemf@cug.edu.cn

收稿日期: 2022-07-22; 修回日期: 2023-04-06; 接受日期: 2024-05-08; 网络出版日期: 2024-09-30

国家自然科学基金(批准号: 62272355, 62176191, 61702383, 62473349)和智能信息处理与实时工业系统湖北省重点实验室开放基金(批准号: ZNXX2023ZD01)资助项目

**摘要** 多模态多目标问题需要求解多个全局或局部最优帕累托解集, 找到这些最优解集具有极大的理论意义和实际经济效益. 近年来, 学者们提出许多进化算法用于解决该问题. 然而, 多数算法在进化过程中首先选择收敛性好的个体构建交配池, 然后再考虑决策空间和目标空间的多样性. 因此, 决策空间中多样性好的个体可能被目标空间收敛性好的个体所替代. 另外, 由于帕累托最优子集在决策空间中有不同的形状和位置, 因此个体很难均匀地收敛到每个帕累托子区域. 本文提出了基于全局密度更新策略的两阶段多模态多目标进化算法. 首先, 为减轻交配池和亲本与子代一对一比较的负面影响, 我们并不构建交配池, 而是提出了新的算法框架, 分两阶段寻优. 该框架利用不同阶段进化算法的特征进行子代更新, 有利于平衡种群的搜索和开发. 然后, 为了解决帕累托子集分布不均的问题, 我们设计了目标空间密度自适应策略和全局密度估计更新种群策略, 用于保持目标空间和决策空间种群多样性. 我们将提出的算法与 7 种有代表性的多模态多目标算法进行比较. 实验结果表明, 我们的算法在决策空间中能找到更多等价的解, 并且能更好地保持决策空间和目标空间多样性和收敛性的平衡, 整体性能要好于所比较算法.

**关键词** 多目标进化算法, 多模态多目标优化问题, 进化算法, 全局密度, 参考向量, 边界点聚集

## 1 引言

近年来, 多目标进化算法<sup>[1~6]</sup>(multi-objective evolutionary algorithms, 简称 MOEA) 已经被用于求解多模态多目标优化问题 (multimodal multi-objective optimization problem, 简称 MMOP). 与多目标

**引用格式:** 贺娟娟, 杨倩, 许志伟, 等. 基于全局密度更新策略的两阶段多模态多目标进化算法. 中国科学: 信息科学, 2024, doi: 10.1360/SSI-2022-0294  
He J J, Yang Q, Xu Z W, et al.. Two-stage multimodal multi-objective evolutionary algorithm based on global density updating strategy (in Chinese). Sci Sin Inform, 2024, doi: 10.1360/SSI-2022-0294

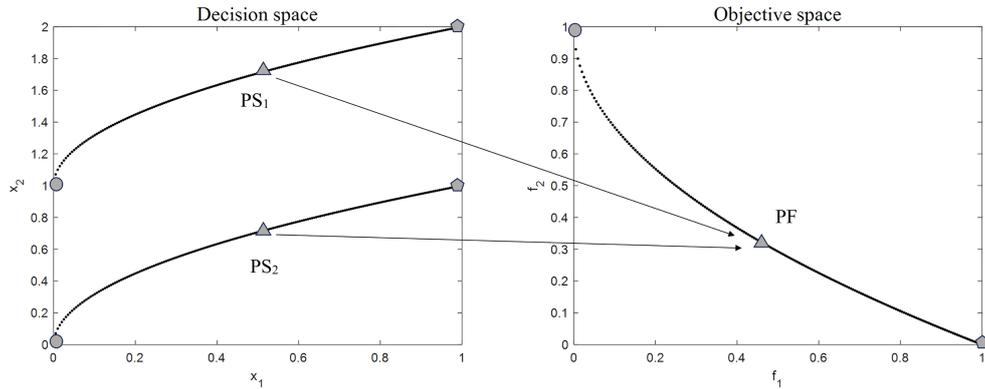


图 1 多模态多目标优化问题举例

Figure 1 An example of multimodal multi-objective optimization problem

优化问题不同的是, MMOP 需要同时寻找多组最优解<sup>[7~9]</sup>. 目前, 多模态多目标进化算法 (multimodal multi-objective evolutionary algorithms, 简称 MMOEA) 已经成功解决许多实际问题, 例如, 火箭发动机设计<sup>[10]</sup>、特征选择<sup>[11]</sup> 和路径规划问题<sup>[12]</sup> 等.

我们一般将多目标优化问题 (multi-objective optimization problem, 简称 MOP) 形式化地描述为:

$$\min f(x) = \min(f_1(x), f_2(x), \dots, f_m(x))^T, \quad (1)$$

其中,  $x = (x_1, x_2, \dots, x_n) \in \Omega$ ,  $x$  是  $n$  维决策向量,  $\Omega$  是决策空间的可行域, 映射函数  $f : \Omega \rightarrow R^m$  定义了目标函数,  $m$  是优化目标总数,  $R^m$  是目标函数空间. MOP 要求搜索一组相互非支配的帕累托最优解集 (Pareto set, 简称 PS), 对应的目标空间称为帕累托前沿 (Pareto front, 简称 PF).

MMOP 是一类特殊的 MOP, 在该问题的决策空间中存在两个或多个不同 PS 对应于完全相同的 PF. 图 1 显示了 MMOP 在决策空间中给定 2 个不同的帕累托子集 (即 PS<sub>1</sub> 和 PS<sub>2</sub>) 的一个例子. 所有的帕累托子集都对应于目标空间中的同一帕累托前沿 (即 PF). 因此, MMOP 与传统的 MOP 相比有很大的区别, 不仅需要同时满足算法的收敛性和在目标空间分布良好这两个条件, 还要满足在决策空间分布良好这个条件<sup>[13]</sup>. 这也就是说, MMOEA 目前面临着同时处理好以下三个问题的挑战: (1) 算法的收敛性, (2) 目标空间的多样性, (3) 决策空间的多样性.

MMOEA 常用的两种框架, 均取得了非常好的效果, 但仍然存在一些挑战. 目前一部分算法使用 Deb 等人在 NSGA-II<sup>[14]</sup> 提出的 N+N 框架, 如 Omni-optimizer<sup>[15]</sup>、DNEA<sup>[16]</sup>、DN-NSGA-II<sup>[17]</sup>. N+N 框架每次迭代时, 有  $N$  个父代个体, 不同算法利用不同的搜索机制生成  $N$  个子代个体, 合并父代和子代个体成为  $2N$  个个体, 不同算法利用不同选择机制从这  $2N$  个个体中挑选出  $N$  个优秀个体, 进入交配池成为下一代父本进入迭代循环. 然而, 当太多收敛性好但多样性较差的个体被选入交配池时, 决策空间多样性好的个体可能被目标空间收敛性好的个体所替代, 决策空间的多样性会渐渐恶化. 另一种常用框架是 Zhang 等人在 MOEA/D<sup>[18]</sup> 提出的 N+1 框架, 如 TriMOEATAR<sup>[19]</sup>、MOEADAD<sup>[20]</sup> 等等. 在此框架下  $N$  个父代个体每次迭代只选择一个个体进行变异 (不同算法可能有不同的变异搜索机制), 每一个新的突变候选方案只需要与相应的父代个体进行比较, 利用不同的选择机制选出优秀的个体. 在 N+1 框架中会出现 3 种情况: (1) 原始方案和新生成的候选方案都是不收敛较差的解; (2) 原始方案和新生成的候选方案其中一个是好的解; (3) 原始方案和新生成的候选方案都是较好的解. 在前两种情况中只用删除最差的解即可, 而第三种情况在同时拥有两个分布均匀且收敛的解时, 必须删

除一个,这就造成了拥有两个优秀的解,但是无法将其同时保留,反而会影响算法的寻优速度.

再者,由于 MMOP 涉及到从决策空间到目标空间的多个对一个单向映射,帕累托最优解集在决策空间中有不同的位置和形状,更新种群时一般用多样性策略,例如 Deb 等人提出的 Omni-optimizer<sup>[15]</sup> 在进化过程先考虑目标空间的多样性再考虑决策空间的多样性,决策空间的多样性没有受到重视,这可能会造成决策空间多样性恶化.因此,之后的许多方法使用特殊拥挤距离配合搜索过程中有效机制,在进化过程中同时兼顾两个空间的多样性.如 MO-Ring-PSO-SCD<sup>[21]</sup>、MO-PSO-MM<sup>[22]</sup>、MMODE<sup>[23]</sup>、SS-MOPSO<sup>[24]</sup> 等.但这些方法在计算拥挤距离时只考虑了个体与其附近个体之间的距离,忽视了解集在空间的整体分布情况,容易造成帕累托子区域分布不均匀.

针对上述问题,我们提出了基于全局密度更新策略的两阶段多模态多目标进化算法 (two-stage multimode multi-objective evolutionary algorithm based on a global density update strategy,简称 MMO-GD).本文的主要贡献如下:

(1) 设计了一种新的两阶段的 MMOEA 框架.与原有算法框架不同,该框架利用进化算法不同阶段的特点,动态调节子代生成个数,有利于平衡种群的搜索和开发,促进算法寻优,并且还能有效节约计算资源.

(2) 提出目标空间密度自适应变异策略,用参考向量将目标空间割成多个子区域.与 MOEA/D 不同,该策略找出个体数量最少的子区域当作父代进行差分进化.与传统算法相比,不仅只在更新种群阶段使用多样性策略,我们在产生新个体时引入多样性策略.能有效自适应调节目标空间的种群分布,避免子区域的丢失.

(3) 为了解决帕累托子区域分布不均匀的情况,我们提出基于全局密度估计的种群更新策略.该策略利用决策空间中所有个体的位置信息,估计个体的分布情况,保持决策空间种群多样性.

本文第 1 节为本文的引言部分.第 2 节介绍了一些相关工作和动机.第 3 节是 MMO-GD 算法的设计细节和流程.第 4 节是实验对比和结果分析部分.第 5 节是本文的结论和对未来工作展望.

## 2 相关工作和动机

MMOEAs 的设计目标与一般的 MOEAs 有很大区别,要求得到的解同时满足三个条件,即收敛性好,目标空间和决策空间分布均匀.近年来,许多 MMOEAs 提出不同策略来解决 MMOPs,大致可以分为三类,基于支配关系的方法,基于分解的方法,以及基于指标的方法.这些方法中可能存在决策空间恶化,优良解生存能力不足,决策空间和目标空间分布不均等问题.我们针对这些问题设计了一种新的两阶段的 MMOEA 框架,提出目标空间密度自适应变异策略,以及全局密度估计的种群更新策略.

### 2.1 多模态多目标优化的方法

**基于支配关系的方法.** 基于帕累托支配关系的方法,首先利用帕累托支配原则收敛解,然后增强目标空间和决策空间多样性.例如,Deb 等人<sup>[15]</sup> 提出了 Omni-optimizer,利用非支配排序的方法,同时考虑目标空间和决策空间的拥挤距离来寻找最优解集.Liu 等人<sup>[16]</sup> 提出了 DNEA,是一种双小生境进化算法,该算法采用共享函数来保持种群多样性.Liang 等人<sup>[17]</sup> 提出了 DN-NSGA-II,将 NSGA-II 和决策空间中小生境选取父代的方法相结合,针对性的改善决策空间多样性.Liu 等人<sup>[19]</sup> 提出了 TriMOEATAR,设计了一种具有两种存档和重组策略的 MMOEA,通过重新组合选出两种存档中的优秀解,构建具有良好整体性能的最终解集.Yue 等人<sup>[21]</sup> 提出 MO-Ring-PSO-SCD,将特殊拥挤距离与多目标粒子群优化算法相结合,一次搜索多个等价帕累托最优解.此外,Liu 等人<sup>[25]</sup> 提出 CPDEA,

将决策空间密度转化为选择准则, 提出了求解 MMOPs 的收敛惩罚密度法.

**基于分解的方法.** 基于分解的方法, 将给定的 MMOP 分解为一组单目标子问题, 然后整合决策空间多样性来寻找多模态最优解. 这类 MMOEAs 大多采用一组均匀分布的参考点或向量将 MMOP 分解成多个子问题, 如基于分解的多目标进化算法 (MOEA/D). Hu 等<sup>[26]</sup> 提出了一种基于分解的方法, 在决策空间中加入了多样性维护机制. 此外, Tanabe 等人<sup>[20]</sup> 提出了一种基于分解的算法, 该算法采用动态种群规模, 并加入了增减算子.

**基于指标的方法.** 基于指标的方法, 采用优化帕累托近似解集的评估指标 (例如超体积指标 Hypervolume 等) 来指导决策空间和客观空间的搜索过程. 这类 MMOEA 更倾向于通过所选指标来引导算法的进化. Ulrich 等人<sup>[27]</sup> 提出了一种基于超容量指标的 MMOEA, 该方法在决策空间和目标空间均采用 Solow-Polasky 分集技术. Ishibuchi 等人<sup>[28]</sup> 设计了双目标解决方案集来优化 hypervolume 指标和决策空间多样性. Tanabe 等人<sup>[29]</sup> 提出了 NIMMO, 是一种基于小生境指标的多模态多目标优化算法, 该算法通过生态位策略保持决策空间多样性.

## 2.2 研究动机

**决策空间中多样性遭到恶化.** 在  $N+N$  框架中, 为了获得较好的子代, 大多数 MMOEA 首先选择收敛性较好的解放入交配池, 然后考虑决策空间和目标空间多样性较好的解放入交配池. 以 Omni-optimizer 为代表, 在非支配排序选择父代的过程中, 收敛良好但多样性较差的解优于收敛较差但多样性较好的解. 在非支配排序步骤中, 优先选择收敛性强的解为父代. 然而, 一些收敛性好的解决方案可能具有较差的多样性. 当收敛性强且较拥挤的解被选中过多时, 决策空间的多样性就会恶化.

**优良解的生存能力不足.** 为了避免上述交配池的负面影响, 在  $N+1$  框架中, 每一个新的变异候选方案只需要与相应的原始方案进行比较, 将较差的方案删除, 只保留其中较好的方案. 在该方法不利于优势解的生存, 因为可能会出现这两个方案同时属于最优解集的情况, 无论删除其中的哪一个, 都是不合理的. 应该找出种群中最差的解进行删除, 这样能够提高解的生存能力, 不用造成计算资源的浪费.

**决策空间和目标空间之间的分布不均匀.** 由于帕累托最优解集在决策空间中是不同的位置和形状, 而在目标空间中的帕累托前沿是分布均匀且有规律的形状, 因此在寻找决策空间的帕累托解集往往面临很大的困难. 由于决策空间分布的复杂性, 大多数 MMOEAs 考虑两个空间的分布时, 往往优先考虑目标空间的多样性, 决策空间的多样性没有得到平等的对待. 在进化过程中, 一些目标空间分布明显较好的个体可能会取代决策空间分布稍好的个体, 因此容易造成决策空间分布不均匀.

## 3 提出的算法 MMO-GD

本文提出一种基于全局密度更新策略的两阶段多模态多目标进化算法. 对于决策空间多样性恶化和提高生存能力的问题, 我们结合  $N+N$  和  $N+1$  框架的优点设计了一种新的两阶段的 MMOEA 框架, 每一个新生成的子代与现有种群合并, 然后将种群中最差的个体删除. 在第一阶段生成大量子代, 有利于算法的全局寻优, 在第二阶段生成子代的数量恒定为 1, 有利于算法的局部探索. 此外, 我们还设计了目标空间自适应密度策略和决策空间全局密度估计更新种群来应对决策空间和目标空间之间的分布不均匀的问题.

**算法 1** 两阶段的多模态多目标进化框架**Input:**  $P$  种群,  $N$  种群大小,  $G$  最大迭代次数;**Output:** 一组最优解集;

```

1: 初始化个数为  $N$  的种群  $P = \{x_1, x_2, \dots, x_N\}$ ;
2: for  $g = 1 : G$  do
3:   for  $i = 1 : N$  do
4:      $V\_size(FES) = N - \frac{N}{n} \times \text{floor}(\frac{FES}{N})$ ; //利用公式 (4) 计算新生成子代数
5:     if  $V\_size < 1$  then
6:        $V\_size = 1$ ;
7:     end if
8:     for  $i = 1 : V\_size$  do
9:       利用算法 2 目标空间密度自适应变异策略生成新的子代个体  $Q$ ;
10:       $P = P \cup Q$ ; //子代与亲本合并
11:     end for
12:     利用非支配排序方法将  $P$  划分为不同 Rank 的非支配解集  $PF = F_1, F_2, \dots, F_{max}$ ;
13:     for  $j = 1 : V\_size$  do
14:       UpdateWithDensity(); //利用算法 3 全局密度估计更新种群
15:     end for
16:   end for
17: end for

```

**3.1** 两阶段的多模态多目标进化框架

我们提出了两阶段的多模态多目标进化框架,用于解决前期全局寻优和后期局部搜索的不足.算法 1 详细描述了该框架的基本流程.算法使用目标空间密度自适应策略,在初期生成与种群数量相同的子代个数,随着迭代次数的增加,新生成的子代数量不断减少,直到新生成的子代生成数量恒定为 1,则不再变化.算法前期个体随机分布在空间内,因此我们生成大量子代用于全局寻优,算法后期种群已大致收敛,可能个别子区域存在分布不均的情况,因此我们只生成少量的子代用于局部寻优.我们将新生成的子代和亲本合并,然后利用非支配的方法结合全局密度评估更新种群.另外,为了精确控制子代的变化,我们提出了子代生成数量  $V\_size$  的变化公式:

$$V\_size(FES) = N - \frac{N}{n} \times \text{floor}\left(\frac{FES}{N}\right), \quad (2)$$

其中,  $N$  代表的种群大小,  $FES$  代表函数评价次数,  $n$  代表子代变化系数,  $\text{floor}$  为向下取整函数.

为了更加直观的展示两阶段各自的作用,我们以  $N = 200, n = 10$  为例画出了子代种群变化趋势图.如图 2 所示,随着迭代次数(迭代次数 =  $\text{floor}(FES/N)$ )的增加,新生成的子代数量呈阶梯形下降趋势,在新生成的子代数量降为 1 之前,该算法处于全局探索阶段.在种群进化的初始阶段,为了避免陷入局部最优,算法前期更加注重探索过程.因此在进化的前期大量生成子代用于全局寻优.当新生成的子代数恒定为 1 时,由探索阶段过渡到开发阶段.此时算法已经大致收敛,有可能局部出现种群分布不均匀的情况.因此,我们在进化的后期更加注重局部开发,有利于局部寻优.同时也能避免种群较大带来的负面影响,节约计算成本.

**3.2** 目标空间密度自适应变异策略

我们的算法定义了每个子区域的邻域.与 MOEA/D 不同,本文中的参考向量仅用于划分目标空间子区域.假设种群中有  $N$  个个体,我们划分出  $N/2$  个子区域,允许每个目标子区域有重叠的解,这样就能有效保留决策空间中多个点对应于目标空间子区域中的同一个点.本文中所采用的变异方式为

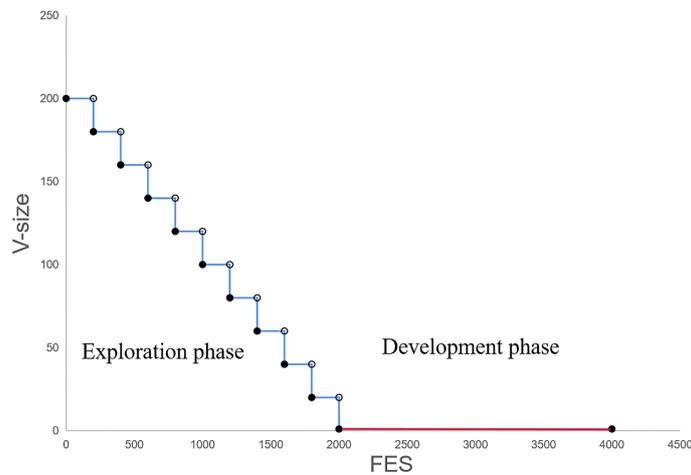


图 2 (网络版彩图) 子代种群数量随迭代次数变化图

Figure 2 (Color online) The number of offspring population changes with the number of iterations

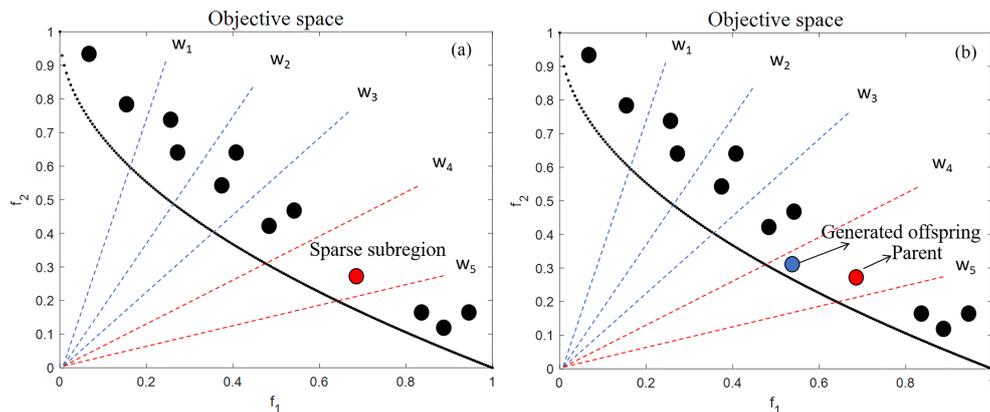


图 3 (网络版彩图) 目标空间密度自适应策略举例. (a) 目标空间变异操作前的分布情况; (b) 目标空间变异操作后的分布情况

Figure 3 (Color online) An example of the objective space density adaptive strategy. (a) Distribution before the objective space mutation operation; (b) the distribution after the objective space mutation operation

差分进化的经典变异. 在变异之前, 设计一种个体预选择机制, 将初始种群进行预处理, 首先找出解最少的子区域选择其中一个个体作为父代, 然后在种群中随机选择两个成为差分矢量, 最后使用差分进化算子生成新的子代. 这样设计的优点在于产生的子代大概率会出现在稀疏子区域, 可自适应的调节目标空间密度的目的. 为了更直观的理解, 我们画图举例说明目标空间密度自适应策略, 如图 3(a) 所示先使用参考向量将目标空间划分为多个子区域, 找到最稀疏的子区域的个体作为父代个体. 然后如图 3(b) 所示, 使用差分进化产生新的子代, 而该子代大概率会出现在父代个体附近, 因而目标空间的多样性得到了大幅度提高. 算法 2 具体描述了如何在目标子空间生成子代个体.

### 3.3 全局密度估计更新种群

本文将种群中的每个个体同等对待, 在计算聚集距离时考虑了个体在决策空间整体的分布情况, 提出了全局密度估计算子, 用于评估个体适应度, 结果越大说明决策空间分布性越好. 在这个过程中,

**算法 2** 目标空间密度自适应变异**Input:**  $P$  种群,  $N$  种群大小,  $\delta$  局部交配的概率;**Output:** 一组最优解集;

- 1: 生成一组参考向量  $W$ ;
- 2: 将  $P$  划分到各区域; //区域数等于  $N/2$
- 3: 统计各子区域个体数量;
- 4: **if**  $\text{rand} < \delta$  **then**
- 5: 从个体数最少的子区域中随机选择 1 个粒子, 在  $P$  中随机选择 2 个粒子;
- 6: **else**
- 7: 从  $P$  中随机选择 3 个粒子;
- 8: **end if**
- 9: 利用差分进化从 3 个粒子中生成一个个体  $Q$ ;

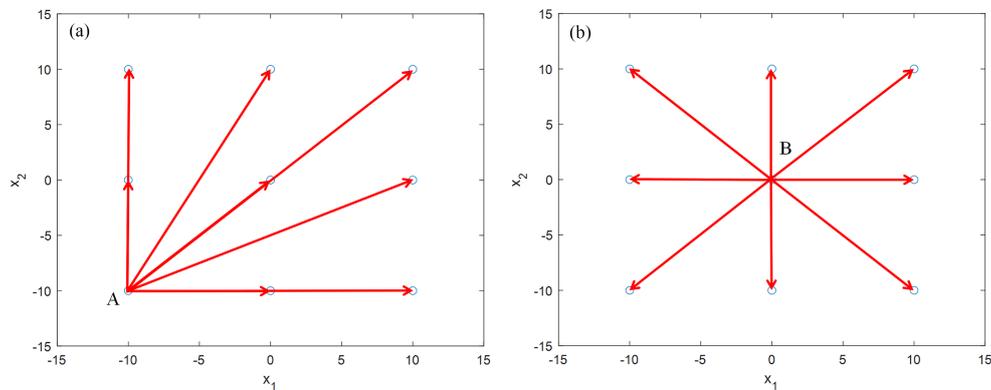


图 4 (网络版彩图) 特殊点到其他点的欧氏距离比较. (a) 中心点示例; (b) 边界点示例

**Figure 4** (Color online) Comparison of Euclidean distances between special points and other points. (a) Central point; (b) boundary point

考虑到越远离中心的点越有优势, 我们使用调和平均数代替欧氏距离. 如图 4 所示, A 点到其他点的欧氏距离之和明显大于 B 点到其他点的欧氏距离之和. 下面我们画图举例说明, 图 5(a) 为  $41 \times 41$  个点均匀分布在决策空间, 使用调和平均数计算决策空间拥挤距离时, 如图 5(b) 所示, 该方法计算出的结果使在中心点的个体平均距离小, 四周的个体平均距离大, 使得中心点的解更容易被删除, 四周的解更容易被保留会造成决策空间中间稀疏四周密集的情况. 图 5(b) 中分别标记了 C、D、E 三个特殊点, 分别为角落点、中心点和边界中点, 调和平均距离分别大约是 0.563、0.421 和 0.299, C、D 点的差值和 D、E 点的差值都在 0.7 左右.

为适应多维决策空间, 我们给每个维度上的决策向量都赋上权值, 用于调节因调和平均数所造成的误差. 首先需要判断每个个体在该维度中的位置, 我们定义了公式 (3).

$$D_n = \frac{X_n^{\max} - X_n^{\min}}{2} \times (X_n^i - X_n^{\min}), \quad (3)$$

其中,  $n$  维决策向量维度,  $X_n^{\max}$ ,  $X_n^{\min}$  分别代表最大边界和最小边界,  $X_n^i$  代表了决策空间的第  $i$  个粒子. 在该维度下, 决策边界的上界限与下界限之差的平均, 代表了中心点与下界限的距离, 再乘以决策空间中某一个体与下界限之差, 如果结果为 1 则说明该点处于中点, 结果为 0 或 2 说明该点处于边界, 这样能准确判断出该个体与中心点的距离. 然后我们设计一个权值公式 (公式 (4)), 用于调节因欧氏距离所造成的误差. 由于每个决策维度与该维度平均距离函数图像呈现出中间小两边大的凹曲线, 因

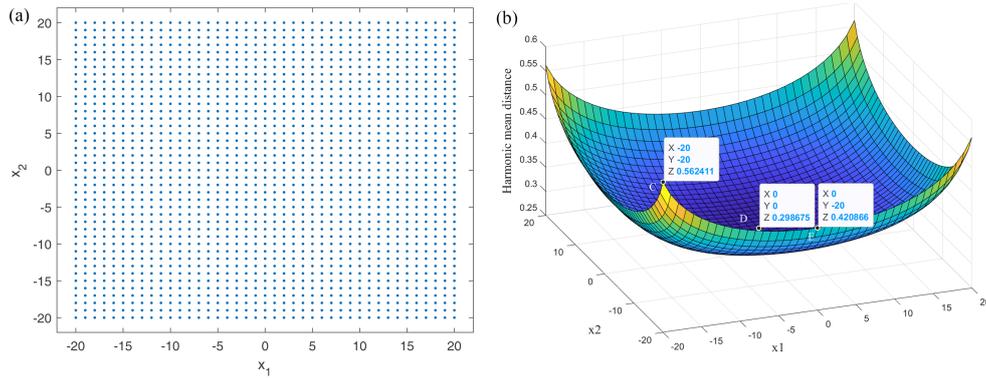


图 5 (网络版彩图) 分布均匀的决策空间中特殊点调和距离举例. (a) 均匀分布的决策空间; (b) 角落点、中心点和边界点的调和距离

Figure 5 (Color online) Examples of special points and their harmonic distances in a uniformly distributed decision space. (a) A uniformly distributed decision space; (b) Harmonic distances for corner point, center point, and boundary point

此我们引入函数图像呈凸曲线的正态分布函数用作权值函数.

$$W_n(D_n) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\mu D_n)^2}{2\sigma^2}}, \quad (4)$$

其中,  $0 \leq D_n \leq 2$  代表了个体与中心点的距离,  $\mu = 1$  代表中点所在位置,  $\sigma$  代表密度改善程度.  $\mu$  决定了其最高点的位置, 其标准差  $\sigma$  决定了分布的幅度. 由于公式 (3) 定义了当结果为 1 说明该点处于中点, 因此  $\mu = 1$ , 由于 C、D 点的差值和 D、E 点的差值都在 0.7 左右, 因此经过计算和实验得出当  $\sigma = 1.45$  时调节误差效果最好. 最后我们将权值公式和调和平均距离相结合成为公式 (5), 我们将其定义为全局密度评估算子.

$$\text{Dist}(X_i^{(t)}) = W_1 \times W_2 \times \dots \times W_n \times \frac{N-1}{\frac{1}{d_1} + \frac{1}{d_2} + \dots + \frac{1}{d_{N-1}}}. \quad (5)$$

使用全局密度估计算子能较好的解决上述中心点因为欧氏距离之和造成的误差. 如图 6, 以 MMF14 为例. 图 6(a) 为加权值调节前的全局密度, 图 6(b) 为加权调节的全局密度. 可明显看出图 6(a) 计算决策空间密度未加权值时中间点较稀疏, 四周点较密集. 图 6(b) 计算决策空间密度加权值后, 决策空间整体分别均匀. 为了更加直观地看出, 图 6(c) 和 (d) 图为俯视图. 算法 3 描述了基于全局密度估计更新种群的全部过程.

## 4 实验结果与分析

### 4.1 实验设置

#### 4.1.1 对比算法

为验证我们算法的性能, 我们选取 7 个专门为 MMOPs 设计的优秀 MMOEAs 进行比较, 包括 TriMOEATAR [19], MOEADAD [30], NIMMO [29] 和近两年提出的新方法 CPDEA [25], HREA [31], MMEA-WI [32], MMOEADC [33]. 其中 MOEADAD 是基于分解的代表, 而 NIMMO 是基于指标的 MMOEA 的代表. 其余的算法是基于非支配的 MMOEAs, 它们具有不同的决策选择策略.

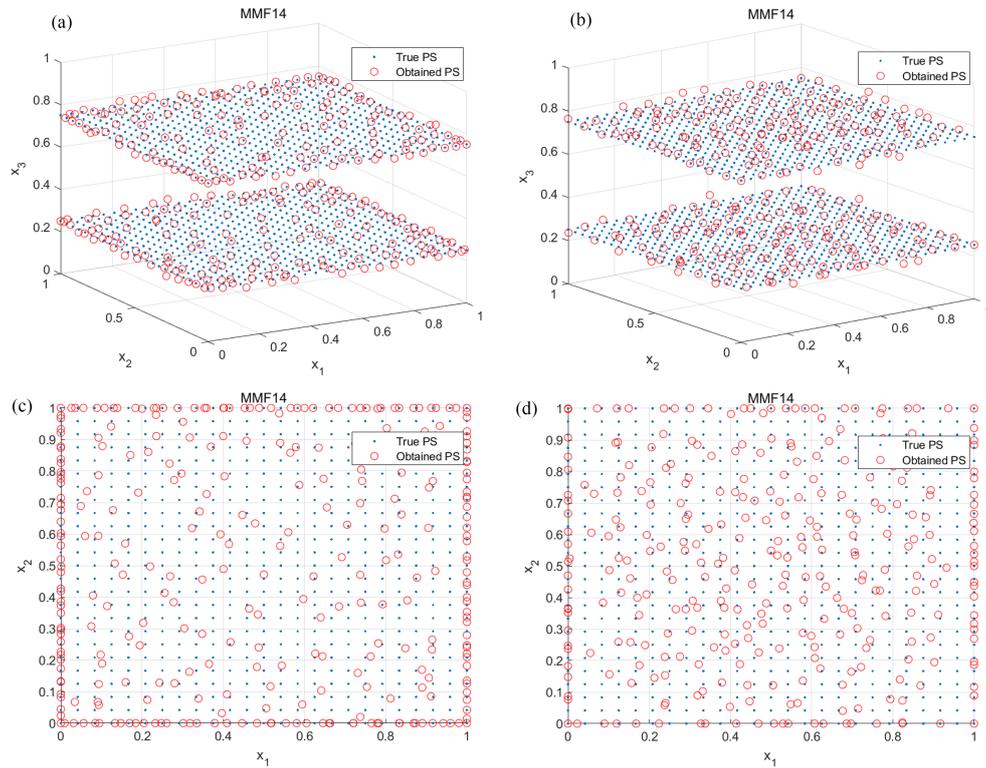


图 6 (网络版彩图) 决策空间密度对比图. (a) 未加权值的全局密度; (b) 加权调节的全局密度; (c) 未加权值的全局密度俯视图; (d) 加权值调节全局密度俯视图

Figure 6 (Color online) Comparison on the decision space density. (a) Global density before weighting adjustment; (b) global density after weighting adjustment; (c) top view of global density before weighting adjustment; (d) top view of global density after weighting adjustment

### 算法 3 全局密度估计更新种群

**Input:**  $P$ : 种群,  $N$ : 种群大小, PF: 不同 Rank 的非支配解集  $PF = F_1, F_2, \dots, F_{\max}$ ;

**Output:**  $P$ : 更新后的种群;

1:  $\text{Dist}(X_i^{(t)}) = W_1 \times W_2 \times \dots \times W_n \times \frac{N-1}{\frac{1}{d_1} + \frac{1}{d_2} + \dots + \frac{1}{d_{N-1}}}$ ; //根据公式 (7) 计算种群的全局密度;

2: **if**  $F_{\max} == 1$  **then**

3: 删除  $\text{Dist}(X_i^{(t)})$  最小的粒子;

4: **else**

5: 删除  $F_{\max}$  且  $\text{Dist}(X_i^{(t)})$  最小的粒子;

6: **end if**

#### 4.1.2 多模态多目标测试函数集

为了测试所提出算法的性能, 我们选取 22 个在 IEEE 进化计算大会 (CEC2019) 的 MMOPs 竞赛上提出的基准测试函数, 包括 MMF1~MMF15, MMF1\_e, MMF1\_z, MMF14\_a, MMF15\_a, SYM-PART simple, SYM-PART rotated 和 Omni-optimizer (详细信息见 [34]). 这些 MMOPs 的特征是凹、凸、球形的帕累托前沿, 等效的帕累托子集的数量从 2、4、9 到 27 不等. 表 1 给出了这 22 个测试函数的具体特征.

在表 1 可以看出这些测试函数特征包括 PS 数量、重叠情况、决策空间维度、目标空间维度、PS

表 1 测试函数具体特征  
Table 1 Specific characteristics of the test problems

Test problem	Number of PSs	PS overlap situation	Decision space dimension	Objective space dimension	PS geometry	PF geometry	Coexistence of global and local Pareto set
MMF1	2	No	2	2	Nonlinear	Convex	No
MMF2	2	No	2	2	Nonlinear	Convex	No
MMF3	2	Yes	2	2	Nonlinear	Convex	No
MMF4	4	No	2	2	Nonlinear	Concave	No
MMF5	4	No	2	2	Nonlinear	Convex	No
MMF6	4	Yes	2	2	Nonlinear	Convex	No
MMF7	2	No	2	2	Nonlinear	Convex	No
MMF8	4	No	2	2	Nonlinear	Concave	No
MMF9	2	No	2	2	Linear	Convex	No
MMF10	2	No	2	2	Linear	Convex	Yes
MMF11	2	No	2	2	Linear	Convex	Yes
MMF12	2	No	2	2	Linear	Convex	Yes
MMF13	2	No	3	2	Nonlinear	Convex	Yes
MMF14	2	No	3	3	Linear	Concave	No
MMF14_a	2	No	3	3	Nonlinear	Sphere	No
MMF15	2	No	3	3	Linear	Sphere	Yes
MMF15_a	2	No	3	3	Nonlinear	Sphere	Yes
MMF1_e	2	No	2	2	Nonlinear	Convex	No
MMF1_z	2	No	2	2	Nonlinear	Convex	No
Omni-test	27	Yes	3	2	Linear	Convex	No
SYM-rotated	9	Yes	2	2	Linear	Convex	No
SYM-simple	9	Yes	2	2	Linear	Convex	No

类型、PF 类型以及是否存在局部 PS. 其中 PS 的数量和重叠情况是决定测试函数求解难度的重要信息, 一般来说, PS 数量越多求解难度越高, 如果多个 PS 存在相互重叠或者相互链接, 这样会更增加求解难度. 因此, 这些测试函数涵盖多个最优子集, 线性和非线性 PS, 凸面和凹面 PF 等特征, 并具有相当难度.

### 4.1.3 性能指标

MOP 的解目标是获得一个均匀分布且尽可能逼近真实帕累托前沿的解集. 大多情况下, 通过在真实帕累托前沿上均匀抽样一定数目的解点, 然后计算这些解点与近似帕累托最优解之间的欧氏距离得到反转世代距离 (inverted generational distance, 简称 IGD) 来评估算法的收敛性和多样性. MMOP 的解目标是获得一个均匀分布且尽可能逼近真实 PS 和 PF 的解集. 在目标空间 (IGDf) 和决策空间 (IGDx) 中的反转世代距离来评价这些竞争算法的性能 [35]. IGDf 指标在公式 (6) 中定义, 它计算从这些参考真 PF 到得到的目标值的平均欧氏距离. IGDf 值越小, 说明所得到的解在目标空间中具有较好的多样性和收敛性.

$$IGDf = \frac{(\sum |PF|_{i=1}^n d_i^2)^{\frac{1}{2}}}{|PF|}, \quad (6)$$

其中,  $d_i$  表示第  $i$  个参考目标向量与得到的最近目标值之间的欧氏距离.

表 2 8 种算法在测试函数上独立运行 30 次获得的 IGD<sub>x</sub> 均值与标准差统计结果Table 2 Statistical results of IGD<sub>x</sub> mean and standard deviation obtained by running the eight algorithms independently 30 times on the test problems

Problem	TriMOEATAR	MOEADAD	NIMMO	HREA	MMEAWI	MMOEADC	CPDEA	MMO-GD
MMF1	4.2114e-2-	6.8994e-2-	5.3418e-2-	3.1261e-2 =	3.8044e-2-	4.4549e-2-	3.1820e-2-	<b>3.0889e-02</b>
MMF2	3.3990e-2-	4.1569e-2-	5.5975e-2-	4.7911e-3 =	7.3723e-3-	6.5881e-3-	5.2502e-3 =	<b>4.7596e-03</b>
MMF3	3.3609e-2-	5.4544e-2-	3.6012e-2-	4.6579e-3 =	6.8791e-3-	6.5417e-3-	5.0697e-3 =	<b>4.6385e-03</b>
MMF4	2.4452e-2-	1.7939e-2+	3.9932e-2-	<b>1.7755e-2 +</b>	2.3807e-2-	2.1526e-2-	1.9267e-2 =	1.9285e-02
MMF5	8.3147e-2-	8.1767e-2-	7.5107e-2-	5.6586e-2 =	6.7574e-2-	7.5917e-2-	5.9472e-2-	<b>5.5490e-02</b>
MMF6	7.7245e-2-	8.6145e-2-	8.8780e-2-	5.5019e-2 =	6.1624e-2-	6.8084e-2-	5.4164e-2 =	<b>5.3705e-02</b>
MMF7	<b>1.7379e-2 +</b>	2.8315e-2 =	1.7475e-2 +	1.9157e-2 +	2.2918e-2 =	2.3134e-2 =	1.8637e-2 +	2.3232e-02
MMF8	5.3796e-1-	1.4609e-1-	1.8779e-1-	<b>3.5619e-2 +</b>	4.9216e-2-	4.4133e-2-	3.7333e-2 =	3.7497e-02
MMF9	3.3173e-3 +	1.6056e-2-	<b>3.3014e-3 +</b>	9.4772e-3-	6.0339e-3 =	6.6150e-3-	6.3357e-3-	5.9881e-03
MMF10	2.0151e-1-	1.7805e-1 =	1.1758e-2 +	<b>8.5005e-3 +</b>	1.8761e-1 =	9.3842e-3 +	2.0093e-1 =	2.0097e-01
MMF11	2.5244e-1-	2.5214e-1-	<b>5.0544e-3 +</b>	1.0734e-2 +	2.4867e-1 =	7.5019e-3 +	2.4894e-1-	2.4846e-01
MMF12	2.4819e-1-	2.4598e-1-	4.5536e-3 +	<b>4.1972e-3 +</b>	2.4429e-1 =	5.0212e-3 +	2.4510e-1-	2.4407e-01
MMF13	2.6797e-1-	2.6984e-1-	1.3957e-1 +	<b>6.0381e-2 +</b>	2.5347e-1-	1.0769e-1 +	2.5430e-1-	2.4960e-01
MMF14	<b>3.7238e-2 +</b>	4.8200e-2 +	7.0938e-2-	6.2435e-2-	5.3626e-2 +	6.2306e-2-	4.6514e-2 +	5.6547e-02
MMF14_a	7.0552e-2 =	7.3822e-2 =	1.2213e-1-	<b>6.2334e-2 +</b>	7.4798e-2 =	1.0111e-1-	6.2471e-2 +	7.2355e-02
MMF15	2.7127e-1-	2.5785e-1 =	1.6813e-1 +	<b>5.3707e-2 +</b>	2.6278e-1-	6.8624e-2 +	2.2878e-1 +	2.6001e-01
MMF15_a	2.2202e-1-	2.1473e-1 =	2.1698e-1-	<b>6.8775e-2 +</b>	2.1536e-1 -	1.2761e-1 +	2.0088e-1 =	2.0699e-01
MMF1_e	1.3560e+0 =	2.1971e+0-	1.3338e+0 =	<b>1.3150e-1 +</b>	3.8510e-1 +	5.1853e-1 =	1.4056e-1 +	9.6377e-01
MMF1_z	3.3569e-2-	5.3602e-2-	3.5773e-2-	2.2777e-2 =	2.9006e-2-	2.8837e-2-	2.4595e-2-	<b>2.2445e-02</b>
Omni-test	7.2749e-1-	9.4879e-2-	3.8174e-1-	6.1168e-2-	1.6646e-1-	1.6164e-1-	7.7655e-2-	<b>5.1918e-02</b>
SYM-rotated	2.8012e+0-	<b>4.3143e-2 +</b>	6.8977e-2-	5.1917e-2 =	6.8053e-2-	5.3737e-2 =	6.7092e-2-	5.3069e-02
SYM-simple	<b>2.6203e-2 +</b>	3.5214e-2 +	4.5419e-2 +	3.9468e-2 +	5.1861e-2 =	4.2069e-2 +	3.3649e-2 +	5.0524e-02
+/-/=	4/16/2	4/13/5	8/13/1	12/3/7	2/13/7	7/12/3	6/9/7	

另一方面, IGD<sub>x</sub> 定义在公式 (7) 中, 它计算从真 PS 到得到的解的平均欧氏距离. IGD<sub>x</sub> 值越小, 说明所得到的解在决策空间中具有较好的多样性和收敛性.

$$\text{IGD}_x = \frac{(\sum |\text{PS}|_j = 1d_j^2)^{\frac{1}{2}}}{|\text{PS}|}, \quad (7)$$

其中,  $d_j$  表示第  $j$  个参考决策向量与得到的最近的决策变量之间的欧氏距离.

#### 4.1.4 实验参数

为了客观评价算法的性能, 我们将所有对比算法的种群数量、函数评价次数 (FES) 和存档大小分别设置为 200、100000、200. 对于 MMO-GD 的权值参数, 密度改善参数  $\sigma$  设为 1.5, 变化系数  $f$  设为 10. 对于所有相互竞争的算法, 模拟二进制交叉分布指数和多项式突变分布指数分别设置为 20. 我们将其他参数设置为与原始文章相同的参数. 为了进行公平的比较, 所有的比较算法都在 22 个测试问题上独立进行了 30 次.

#### 4.2 有效性实验

本文在 MMF1~MMF15、MMF1\_e、MMF1\_z、MMF14\_a、MMF15\_a、SYM-PART simple、SYM-PART rotated 和 Omni-optimizer 等测试函数上做对比实验. 这些测试函数在 IEEE 进化计算大会 (CEC2019) MMOPs 竞赛上提出, 并作为黑盒问题处理, 即求解时不使用这些问题的显式方程式.

表 3 8 种算法在测试函数上独立运行 30 次获得的 IGDF 均值与标准差统计结果

Table 3 Statistical results of IGDF mean and standard deviation obtained by running the eight algorithms independently 30 times on the test problems

Problem	TriMOEATAR	MOEADAD	NIMMO	HREA	MMEAWI	MMOEADC	CPDEA	MMO-GD
MMF1	4.0783e-3 =	6.4853e-3 -	4.8567e-3 -	3.8483e-3 =	3.4208e-3 +	3.4509e-3 +	<b>2.4357e-3 +</b>	3.8801e-03
MMF2	5.5424e-3 -	1.6636e-2 -	9.3145e-3 -	2.8186e-3 =	3.6477e-3 -	3.8219e-3 -	2.8146e-3 =	<b>2.7236e-03</b>
MMF3	4.5524e-3 -	1.6833e-2 -	8.4213e-3 -	2.7415e-3 =	3.5712e-3 -	3.9039e-3 -	2.7313e-3 =	<b>2.6726e-03</b>
MMF4	3.4823e-3 +	3.2500e-3 +	5.3828e-3 -	3.4435e-3 +	3.6461e-3 =	<b>2.2844e-3 +</b>	2.4823e-3 +	3.7702e-03
MMF5	3.2588e-3 +	5.4296e-3 -	3.7270e-3 =	4.0215e-3 =	3.5711e-3 =	3.4973e-3 =	<b>2.3920e-3 +</b>	3.7241e-03
MMF6	3.3365e-3 +	5.9095e-3 -	3.4991e-3 +	4.1245e-3 =	3.4035e-3 +	3.5608e-3 +	<b>2.3528e-3 +</b>	3.8891e-03
MMF7	3.5143e-3 +	3.4500e-3 +	2.8035e-3 +	3.8640e-3 +	3.9056e-3 +	2.7288e-3 +	<b>2.5208e-3 +</b>	4.6020e-03
MMF8	8.1355e-3 -	8.7486e-3 -	9.2897e-3 -	3.5760e-3 =	3.5575e-3 =	<b>2.2991e-3 +</b>	2.5550e-3 +	3.4382e-03
MMF9	7.4029e-2 -	1.6622e-2 =	1.1206e-2 +	3.5086e-2 -	1.6790e-2 =	1.0434e-2 +	<b>9.9312e-3 +</b>	1.5955e-02
MMF10	2.3198e-1 -	1.7684e-1 +	<b>1.7487e-1 =</b>	2.2306e-1 -	1.9040e-1 =	1.9925e-1 =	1.9434e-1 +	1.9772e-01
MMF11	1.6627e-1 -	9.7497e-2 +	9.5719e-2 =	1.2187e-1 -	9.6409e-2 +	9.7237e-2 =	<b>9.3303e-2 +</b>	9.9715e-02
MMF12	8.5191e-2 =	8.7790e-2 -	1.0760e-1 -	8.8548e-2 -	8.5434e-2 =	8.9170e-2 -	<b>8.2786e-2 +</b>	8.5438e-02
MMF13	2.5320e-1 -	1.5270e-1 =	1.4268e-1 =	2.1375e-1 -	1.4912e-1 =	1.5320e-1 =	<b>1.4779e-1 +</b>	1.5749e-01
MMF14	9.0358e-2 +	7.9443e-2 +	1.1073e-1 -	1.1179e-1 -	8.8176e-2 +	8.5691e-2 +	<b>6.9583e-2 +</b>	9.6731e-02
MMF14_a	9.9453e-2 -	8.4854e-2 +	1.2896e-1 -	9.0292e-2 =	9.6973e-2 -	9.2541e-2 =	<b>7.2974e-2 +</b>	9.3001e-02
MMF15	2.0906e-1 -	1.9408e-1 =	2.1417e-1 -	2.0503e-1 -	1.9754e-1 -	2.0991e-1 -	<b>1.8036e-1 +</b>	1.9453e-01
MMF15_a	2.0018e-1 -	1.9001e-1 =	2.2265e-1 -	2.0692e-1 -	1.9301e-1 =	2.1796e-1 -	<b>1.7750e-1 +</b>	1.8914e-01
MMF1_e	3.7949e-3 -	1.2642e-1 -	4.6628e-3 -	3.8176e-3 -	6.5950e-3 -	4.2467e-3 -	3.1487e-3 -	<b>2.8809e-03</b>
MMF1_z	3.3737e-3 =	6.5713e-3 -	4.0906e-3 -	3.2976e-3 =	3.4839e-3 =	2.7492e-3 =	<b>2.3698e-3 +</b>	3.2404e-03
Omni-test	1.9361e-2 -	1.1775e-2 =	3.4748e-2 -	1.1686e-2 =	1.1456e-2 =	3.8635e-2 -	<b>9.6931e-3 +</b>	1.1443e-02
SYM-rotated	3.9798e-2 -	1.7169e-2 =	4.4181e-2 -	1.5233e-2 =	1.6195e-2 =	<b>1.0311e-2 +</b>	1.0829e-2 +	1.5740e-02
SYM-simple	4.7427e-2 -	2.3819e-2 -	2.8987e-2 -	1.5861e-2 =	1.4504e-2 =	1.1054e-2 +	<b>9.7202e-3 +</b>	1.4124e-02
+/-/=	5/14/3	6/10/6	3/15/4	2/9/11	5/5/12	9/7/6	19/1/2	

实验结果如表 2 和表 3 所示。

我们对比每种算法的 IGDX 均值、IGDF 均值和标准差。IGDX 均值和 IGDF 均值越小, 说明算法的收敛性和多样性越好; 标准差越小, 说明算法稳定性越好。加粗表示在此测试问题上该项平均值为最优值。另外, 我们使用秩和检验在显著性水平为 0.05 下对 30 次独立运行结果进行分析。“+”表示此算法在此测试问题上与 MMO-GD 相比在该指标上表现更好,“-”表示此算法在此测试问题上与 MMO-GD 相比在该指标上表现更差,“=”表示此算法在此测试问题上与 MMO-GD 相比在该指标上的表现区别不大。每个表格的最后一行还给出了在该指标上的 +/-/= 结果的汇总。

在表 2 中, 它显示了对 22 个测试问题的 IGDX 平均值。其中 16 个为全局 MMOP, 另外 6 个为存在局部最优解的 MMOP(已用灰色底纹标记出)。从表 2 中可以看出, MMO-GD 在全局 MMOP 决策空间中取得了较好的性能, 在 8 个对比算法中, 16 个全局 MMOP 中有 7 个获得了最佳 IGDX 值(加粗字体表示)。此外, HREA、TriMOEATAR 和 NIMMO 分别在 4 个、3 个和 1 个全局 MMOP 问题上获得最好的 IGDX 值。其中 MMF10~MMF13、MMF15、MMF15\_a 这 6 个问题存在局部最优解, 而对比算法 NIMMO、HREA 和 MMOEADC 分别采用了分区域、分层次和聚类策略的 MMOEAD, 这些方法都能较好的解决存在局部最优解的 MMOP, 而其他算法在解决存在局部最优问题上还存在不足, 只能找到全局最优解或部分局部最优解。因此在存在局部最优解的 MMOP 问题上其他算法与算法 NIMMO、HREA 和 MMOEADC 的 IGDX 的结果相差很大。在 16 个全局 MMOP 测试问题上, MMO-GD 在决策空间获得 7 个最佳值, 而 NIMMO 只获得了 1 个最佳 IGDX 值, HREA 获得 4 个最

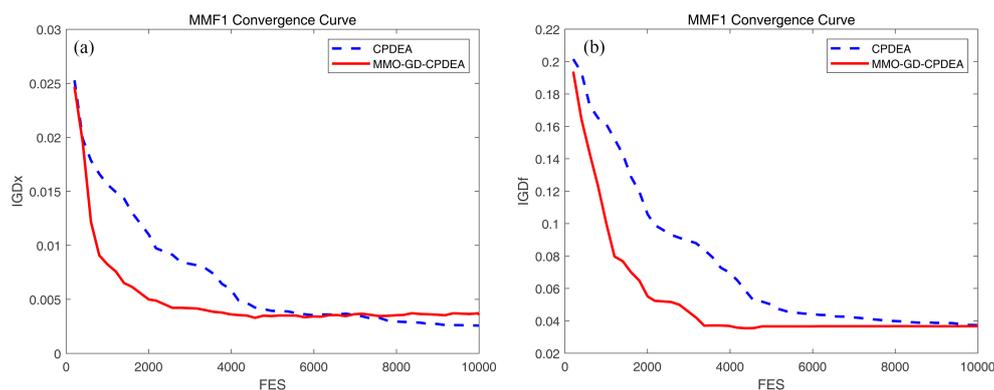


图 7 (网络版彩图) CPDEA 和 MMO-GD-CPD 在 MMF1 问题上的 IGDx (a) 与 IGDf (b) 的收敛曲线  
**Figure 7** (Color online) Convergence curves of IGDx (a) and IGDf (b) with CPDEA and MMO-GD-CPD on the MMF1 problem

佳 IGDx 值并且有 7 个 IGDx 与 MMO-GD 相差不大, MMOEADC 在全局问题上没有取得最佳 IGDx 值, 因此 NIMMO 和 MMOEADC 在解决全局问题上还存在不足. 与上述全部对比算法相比, 所提出的 MMO-GD 在大部分全局问题上能取得较好的结果. 因此说明该方法在全局 MMOP 能找到分布良好和收敛良好的解.

在表 3 中, 它显示了对 22 个测试问题的 IGDf 平均值. IGDf 指标用于评价目标空间中解集到真帕累托集的多样性和收敛性. 本文提出的 MMO-GD 虽然优于大多数对比算法, 但 CPDEA 在 IGDf 值上取得了绝对性的优势. 我们认为原因在于 CPDEA 建立了两个档案, 在第一个档案中存放多样性好的解, 另一个档案存放收敛性好的解. 为将档案一中决策空间的多样性处理得比收敛性更重要, 提出了一种收敛惩罚密度 (convergence-penalized density, 简称 CPD) 的方法, 将决策空间中的密度值和收敛程度相结合成为一个评估算子. 基于这种收敛惩罚密度的方法, 可以得到覆盖整个决策空间的解, 总体将不会过早地收敛到一些简单的等价的帕累托最优解, 能够很大程度保证解的多样性, 具有很好的实验结果.

本文提出的算法 MMO-GD 有两大优势首先提出了新的分两阶段算法框架, 该框架利用进化算法不同阶段的特征进行子代更新, 有利于平衡种群的搜索和开发. 然后, 为了解决帕累托子集分布不均的问题, 设计了目标空间密度自适应策略和全局密度估计更新种群策略, 用于保持目标空间和决策空间种群多样性. 通过实验结果说明, 我们的方法能够更好地保持决策空间和目标空间多样性和收敛性的平衡.

为了进一步说明我们方法的有效性, 我们在所提算法 MMO-GD 中加入 CPDEA 中的收敛惩罚密度方法, 将该算法命名为 MMO-GD-CPD. 加入收敛惩罚密度方法的 MMO-GD, 收敛速度得到大幅度提升. 如图 7 所示, 在函数评价次数为 4000 次时, MMO-GD-CPD 已经取得较好的收敛结果, 此时种群数量设置为 200. 由于 MMF10~MMF13、MMF15、MMF15\_a 是存在局部最优解的测试函数, MMO-GD 和 MMO-GD-CPD 这两个算法在局部最优解 MMOP 均不能取得较好效果, 因此不考虑存在局部最优解的测试函数, 而选取 MMF1~MMF9、MMF1\_e、MMF1\_z、MMF14、MMF14\_a、SYM-PART simple、SYM-PART rotated 和 Omni-optimizer 函数进行测试, 得到结果如表 4 所示.

表 4 中, 它显示了对 16 个测试问题的 IGDx 和 IGDf 平均值. 从表中可以看出, MMO-GD-CPD 在 MMOP 决策空间中取得了较好的性能, 取得最佳值用加粗字体标出, 在 16 个全局 MMOP 中获得了全部最佳 IGDx 值, 获得了 11 个最佳 IGDf 值. 在本文提出的算法中加入收敛惩罚密度方法, 能较

表 4 CPDEA 和 MMO-GD-CPD 的 IGD<sub>x</sub> 与 IGD<sub>f</sub> 均值与标准差统计结果

Table 4 Statistical results of the mean and standard deviation of IGD<sub>x</sub> and IGD<sub>f</sub> with CPDEA and MMO-GD-CPD

Problem	IGD <sub>x</sub>		IGD <sub>f</sub>	
	CPDEA	MMO-GD-CPD	CPDEA	MMO-GD-CPD
MMF1	6.3221e-2 (3.75e-3) -	<b>3.6022e-2 (1.15e-3)</b>	4.4859e-3 (2.26e-4) -	<b>3.8035e-3 (1.63e-4)</b>
MMF2	6.7604e-2 (1.56e-2) -	<b>4.7701e-3 (5.51e-3)</b>	3.3298e-2 (6.27e-3) -	<b>3.2510e-3 (1.02e-3)</b>
MMF3	7.8085e-2 (3.30e-2) -	<b>4.3176e-3 (8.71e-4)</b>	3.3518e-2 (3.38e-3) -	<b>2.7531e-3 (9.16e-4)</b>
MMF4	2.7457e-2 (2.15e-3) -	<b>1.9765e-2 (6.71e-4)</b>	<b>3.0667e-3 (1.39e-4)</b> +	3.9207e-3 (2.01e-4)
MMF5	1.0553e-1 (7.61e-3) -	<b>6.1967e-2 (1.63e-3)</b>	4.4276e-3 (1.17e-4) -	<b>3.5074e-3 (6.56e-5)</b>
MMF6	8.6727e-2 (9.99e-3) -	<b>5.4564e-2 (9.86e-4)</b>	3.8502e-3 (1.63e-4) =	<b>3.5998e-3 (2.41e-4)</b>
MMF7	2.6146e-2 (2.16e-3) -	<b>2.0937e-2 (5.99e-4)</b>	<b>3.0315e-3 (7.74e-5)</b> +	3.9045e-3 (1.60e-4)
MMF8	1.1809e-1 (8.47e-3) -	<b>4.7743e-2 (3.22e-3)</b>	5.5712e-3 (4.04e-4) -	<b>3.7492e-3 (1.42e-4)</b>
MMF9	6.5437e-3 (3.21e-4) =	<b>6.0903e-3 (8.99e-5)</b>	<b>1.0831e-2 (2.55e-4)</b> +	1.6628e-2 (1.76e-3)
MMF14	4.3154e-2 (8.66e-4) =	<b>4.2514e-2 (3.52e-4)</b>	<b>6.1732e-2 (1.17e-3)</b> +	7.1638e-2 (5.78e-4)
MMF14_a	5.8996e-2 (1.61e-3) -	<b>5.4810e-2 (4.04e-4)</b>	<b>6.8076e-2 (2.55e-3)</b> +	7.2643e-2 (1.68e-3)
MMF1_e	1.0737e+0 (4.52e-1) -	<b>3.2734e-1 (3.47e-2)</b>	1.9758e-2 (2.97e-3) -	<b>3.8744e-3 (3.92e-4)</b>
MMF1_z	4.7641e-2 (4.14e-3) -	<b>2.4965e-2 (9.01e-4)</b>	4.4058e-3 (3.49e-4) -	<b>3.2045e-3 (1.15e-4)</b>
Omni-test	5.6257e-1 (9.26e-2) -	<b>5.3266e-2 (1.35e-2)</b>	2.7490e-2 (3.89e-3) -	<b>9.0891e-3 (2.17e-4)</b>
SYM-rotated	2.5685e+0 (1.22e+0) -	<b>5.7652e-2 (5.70e-1)</b>	6.2372e-2 (2.36e-2) -	<b>2.2747e-2 (2.45e-3)</b>
SYM-simple	2.0578e+0 (1.17e+0) =	<b>5.7072e-2 (9.01e-1)</b>	3.7062e-2 (4.49e-3) -	<b>1.3401e-2 (2.00e-3)</b>
+/-/=	0/13/3		2005/10/1	

快取得收敛较好的结果, 并且最终 IGD 结果均有所提升, 这表明本文提出的 MMO-GD 算法在决策空间和目标空间的收敛性和分布性确实有效。

为了更直观展示本文提出的算法的性能, 我们绘制了 8 种算法在 4 个测试函数上得到的近似 PS 和 PF 在决策空间和目标空间图。这 4 个多模态多目标测试函数分别为 MMF1、MMF2、MMF10 和 Omni-test。

如图 8 所示, MMO-GD 在 MMF1 函数的决策空间取得了相较其他算法更均匀的分布, 而其他算法则有堆积在中间或者拐点处的情况, 说明全局密度策略能较好的解决决策空间分布不均的问题。虽然 MMO-GD 在目标空间没有取得均匀的分布, 但是在目标空间分布最好的 CPDEA 其决策空间的解存在拐点堆积的情况, 因此算法并不能完美兼顾两空间分布, 一个性能良好的 MMEOA 应该在决策空间获得更好的结果。

如图 9 所示, HREA, MMEAWI, MMOEADC, CPDEA 和 MMO-GD 在 MMF2 函数的决策空间和目标空间取得了均匀的分布, 而 TriMOEATAR, MOEADAD, NIMMO 没有均匀分布在决策空间和目标空间内, 存在 PS 子集丢失的情况。

如图 10 所示, 是存在局部最优解集的测试函数, 它存在两条真实帕累托前沿, 其中灰色“+”代表的是局部最优解集, 黑色“.”代表的是全局最优解集。MMO-GD 虽然只能找到全局最优解集, 但是 MMO-GD 全局最优解能在决策空间和目标空间均匀分布, 而 TriMOEATAR, MOEADAD, 存在全局最优解丢失的情况。NIMMO, HREA 和 MMOEADC 虽然能找到局部最优解集, 但 HREA 和 MMOEADC 的局部最优解集在决策空间没有完全收敛, NIMMO 是基于指标的 MMOEAD, 能较好找到局部最优解集, 但也存在个别局部最优解丢失的情况。

如图 11 所示, 测试函数 Omni-test 是 PS 子集较多的 MMOP, MMO-GD 能较好的找到每个子集, 并且均匀分布在其上面, 而其他算法存在 PS 子集丢失或者分布不均的情况。

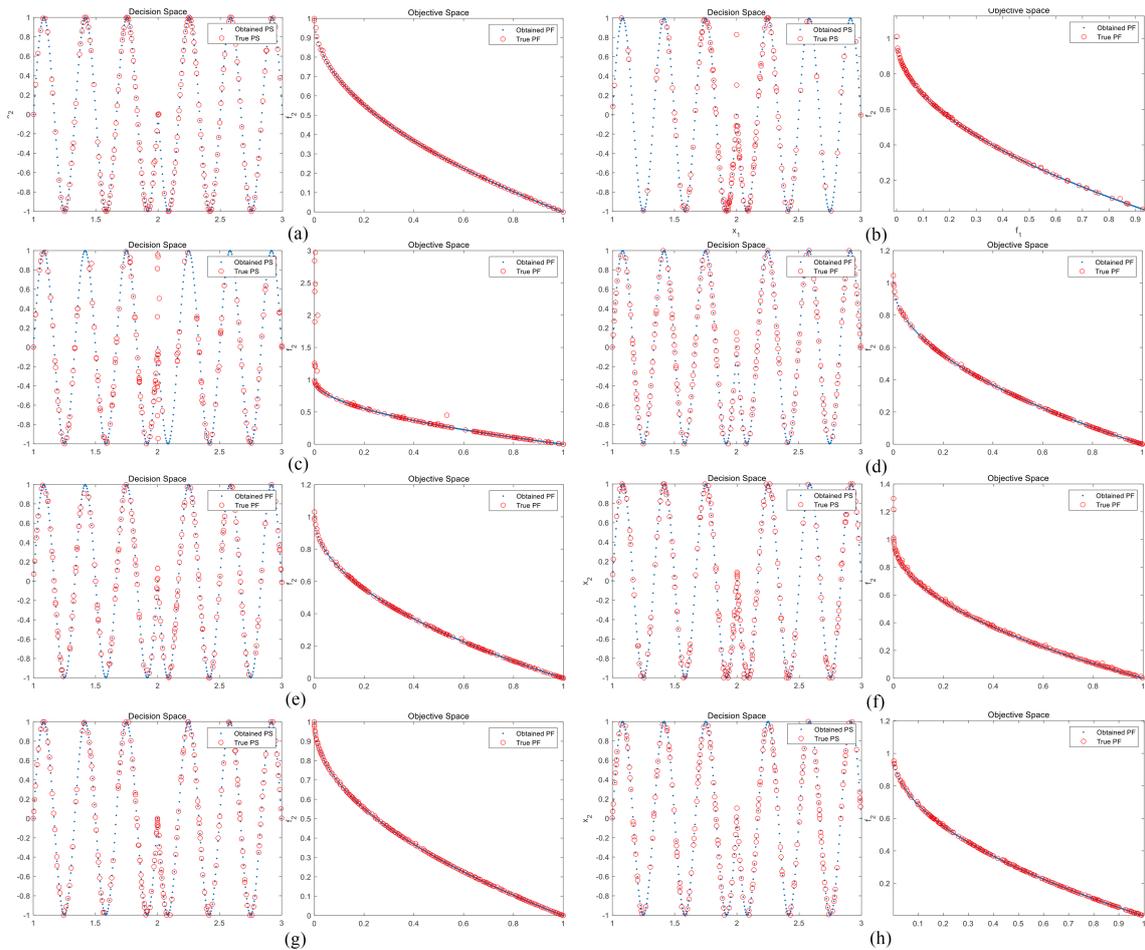


图 8 (网络版彩图) 各算法在 MMF1 的结果分布. (a) TriMOEATAR; (b) MOEADAD; (c) NIMMO; (d) HREA; (e) MMEAWI; (f) MMOEADC; (g) CPDEA; (h) MMO-GD

Figure 8 (Color online) The PSs and PFs obtained by each algorithm in MMF1. (a) TriMOEATAR; (b) MOEADAD; (c) NIMMO; (d) HREA; (e) MMEAWI; (f) MMOEADC; (g) CPDEA; (h) MMO-GD

### 4.3 收敛性试验

为了验证 MMO-GD 算法的收敛性, 本文选取 MMF1、MMF2、Omni-test 这 3 个多模态多目标测试函数, 考察 MMO-GD 算法与其他 7 种对比算法在 100000 次函数评价次数 (FES) 的过程中其 IGD<sub>x</sub> 指标值的变化情况, IGD<sub>x</sub> 为测量得到的 PS 与真正的 PS 之间的距离, 能够反映决策空间收敛性. 因此, 通过观察各算法近似解集的 IGD<sub>x</sub> 值随 FES 变化, 可以判断出算法收敛速度的快慢.

如图 12 所示, 在 MMF1 上 TriMOEATAR、CPDEA 和 MMO-GD 完全收敛需要 20000 次以上, 其他算法虽然在 10000 次内完全收敛, 但是最终结果并不能取得优势; 在 MMF2 上大多数算法在 80000 次左右才完全收敛, 虽然 MMF2 为 2 个最优子集的 MMOP, 但其中的一个最优子集容易丢失, 或者存在最优子集找不全的情况, 所以需要较多 FES 算法才能收敛; 在 Omni-test 上大多数算法在 50000 次左右才完全收敛, Omni-test 为 27 个最优子集的 MMOP, 要找到每个最优子集, 并且能均匀分布在每个最优子集上是比较困难的, 因此需要较多 FES 算法才能收敛. 从图 12 的收敛曲线可以看出, 本文提出的算法 MMO-GD 能够持续稳定的收敛并且最终获得了最优值.

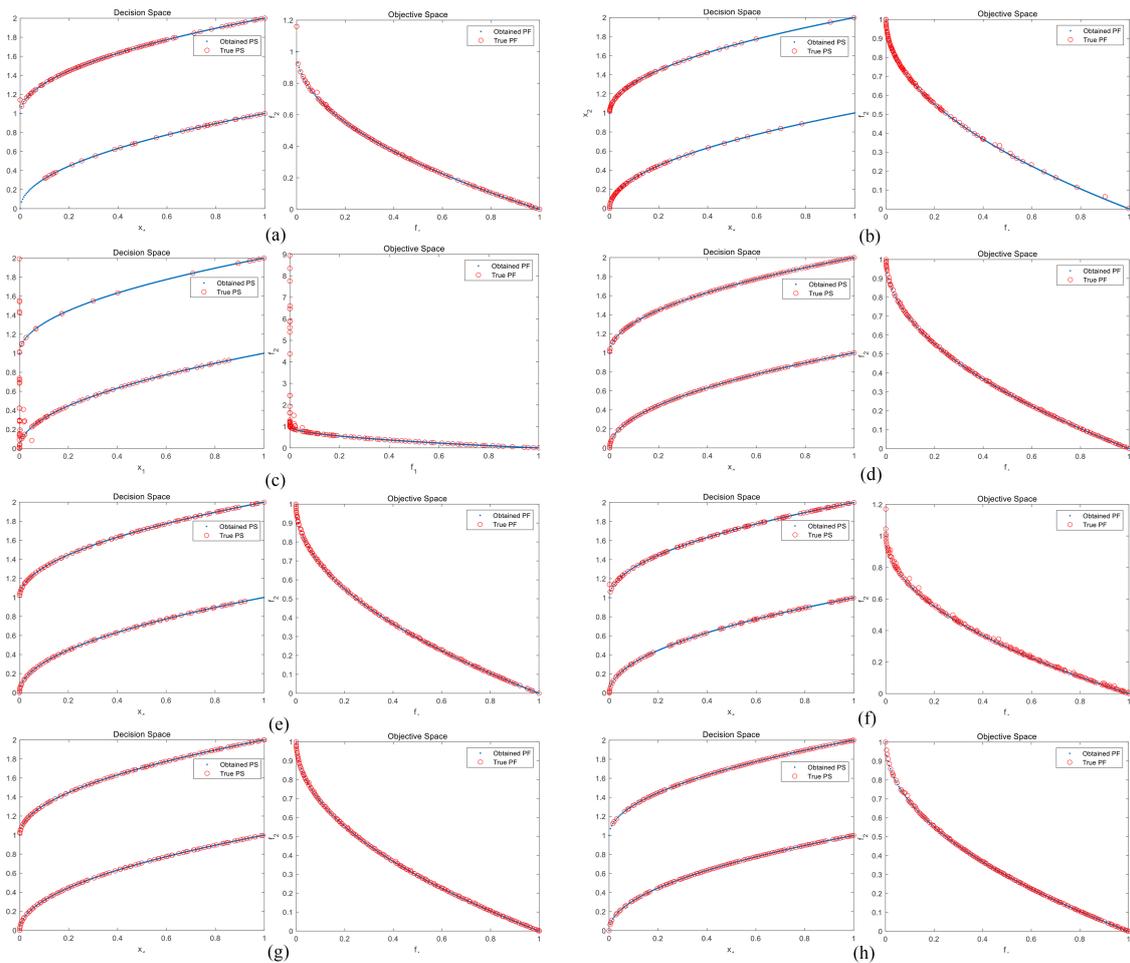


图 9 (网络版彩图) 各算法在 MMF2 的结果分布. (a) TriMOEATAR; (b) MOEADAD; (c) NIMMO; (d) HREA; (e) MMEAWI; (f) MMOEADC; (g) CPDEA; (h) MMO-GD

Figure 9 (Color online) The PSs and PFs obtained by each algorithm in MMF2. (a) TriMOEATAR; (b) MOEADAD; (c) NIMMO; (d) HREA; (e) MMEAWI; (f) MMOEADC; (g) CPDEA; (h) MMO-GD

#### 4.4 算法稳定性分析

为了说明算法的稳定性, 本节选出 MMF1~MMF4 测试函数, 画出 IGD<sub>x</sub> 与所对比算法的箱线图, 做出上下界定量分析. 如图 13 所示, 在 MMF1 上 HREA、MMOEADC、CPDEA 和 MMO-GD 的 IGD<sub>x</sub> 值都较为集中, 其中 CPDEA 和 MMO-GD 最大值和最小值相差不大且 IGD<sub>x</sub> 值优于其他算法, 这表明 CPDEA 和 MMO-GD 这两个算法在 MMF1 上具有最好的稳定性与收敛性; 在 MMF2 和 MMF3 上 HREA、MMEAWI、MMOEADC、CPDEA 和 MMO-GD 的 IGD<sub>x</sub> 值都较为集中且最大值和最小值相差不大, 并且 IGD<sub>x</sub> 值远好于 TriMOEATAR、MOEADAD 和 NIMMO, 这表明 HREA、MMEAWI、MMOEADC、CPDEA 和 MMO-GD 在 MMF2 和 MMF3 上具有最好的稳定性与收敛性, 在 MMF4 上 MOEADAD、HREA、MMEAWI、CPDEA 和 MMO-GD 的 IGD<sub>x</sub> 值都较为集中, 其中 HREA、CPDEA 和 MMO-GD 最大值和最小值相差不大, 这表明 HREA、CPDEA 和 MMO-GD 在 MMF4 上具有最好的稳定性与收敛性. 这 4 个测试函数上 MMO-GD 的箱线图分布良好, 这表明 MMO-GD 具有较好的稳定性.

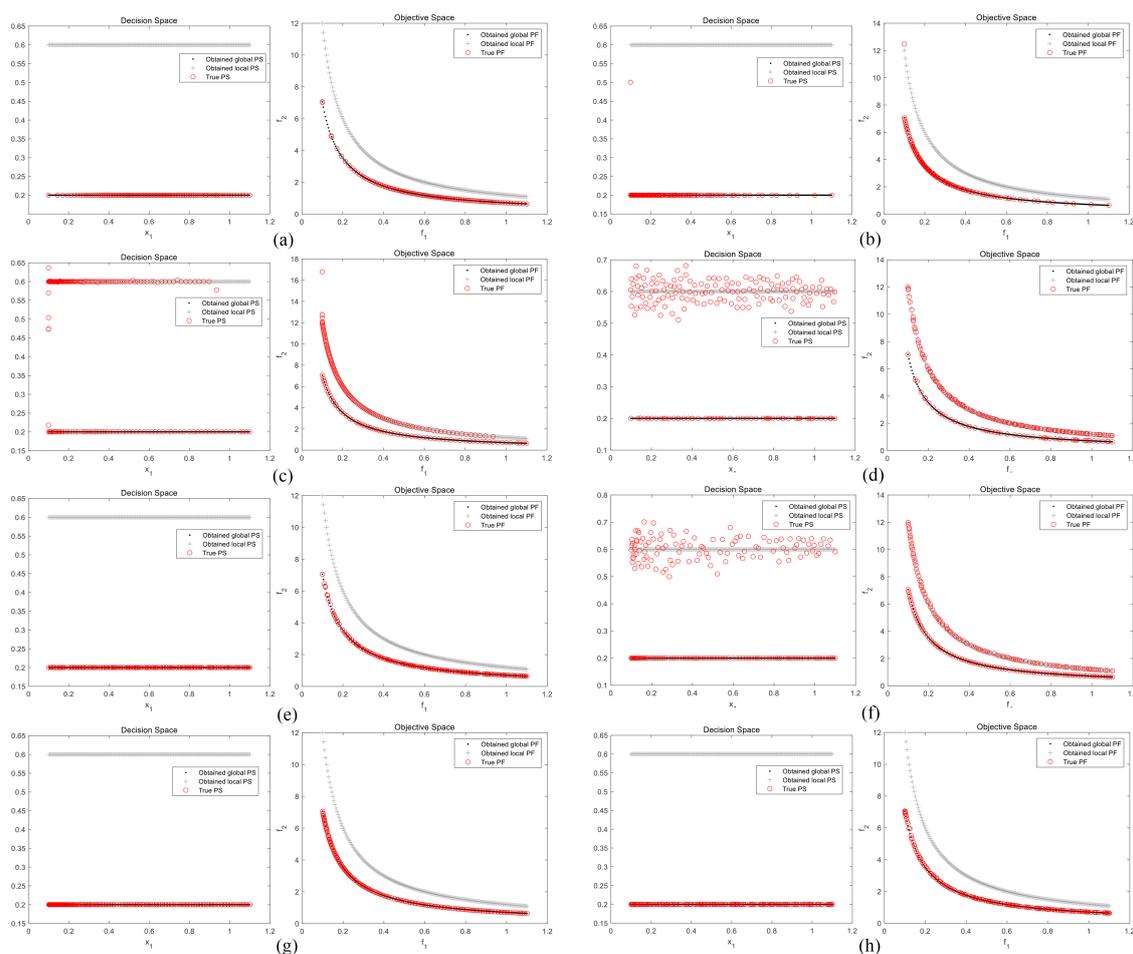


图 10 (网络版彩图) 各算法在 MMF10 的结果分布. (a) TriMOEATAR; (b) MOEADAD; (c) NIMMO; (d) HREA; (e) MMEAWI; (f) MMOEADC; (g) CPDEA; (h) MMO-GD

Figure 10 (Color online) The PSs and PFs obtained by each algorithm in MMF10. (a) TriMOEATAR; (b) MOEADAD; (c) NIMMO; (d) HREA; (e) MMEAWI; (f) MMOEADC; (g) CPDEA; (h) MMO-GD

#### 4.5 两阶段算法框架有效性分析

MMO-GD 使用两阶段进化框架平衡算法的全局搜索和局部寻优能力, 同时维护决策空间的多个最优子集. 本节主要探究两阶段搜索的方法是否对 MMOEA 产生了积极的影响. 设计两阶段算法框架有效性实验部分, 仅关注两阶段算法框架与 N+N 和 N+1 算法框架的优劣, 并不需要关注目标空间和决策空间的具体情况, 因此按顺序选择测试函数 MMF1~MMF8 进行实验. 这 3 种算法框架均采用支配的方法和全局密度计算个体间距离. 以此验证两阶段进化框架的有效性和必要性. 实验中各测试函数总评价次数为 100000 次, 这 8 种测试函数都进行 30 次独立重复实验, 利用函数评价次数作为终止条件获得的平均 IGD<sub>x</sub> 和 IGD<sub>f</sub> 指标值如表 5 和表 6 所示.

表 5 和表 6 记录了不同算法框架 IGD<sub>x</sub> 和 IGD<sub>f</sub> 均值与标准差统计结果, 实验结果显示在这 8 个测试函数中, 两阶段算法框架 IGD<sub>x</sub> 和 IGD<sub>f</sub> 大多数值优于 N+N 算法框架和 N+1 算法框架, 这表明两阶段算法框架的优越性. 虽然 N+N 算法框架和 N+1 算法框架也能取得较好的结果, 但两阶段算法框架结合这两个算法框架的优势, 在算法不同阶段针对性做了处理, 完善了对比的两个算法框架的不

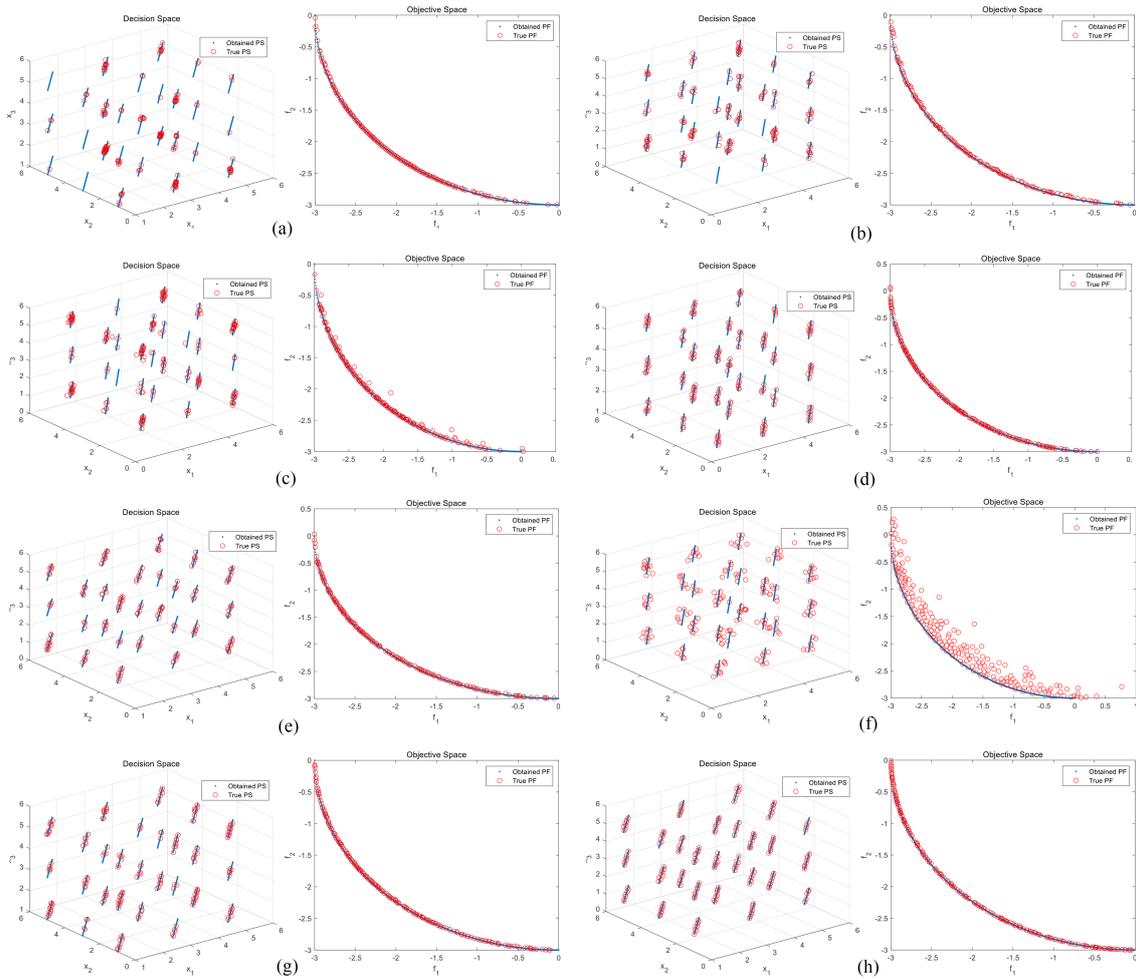


图 11 (网络版彩图) 各算法在 Omni-test 的结果分布. (a) TriMOEATAR; (b) MOEADAD; (c) NIMMO; (d) HREA; (e) MMEAWI; (f) MMOEADC; (g) CPDEA; (h) MMO-GD

Figure 11 (Color online) The PSs and PFs obtained by each algorithm in Omni-test. (a) TriMOEATAR; (b) MOEADAD; (c) NIMMO; (d) HREA; (e) MMEAWI; (f) MMOEADC; (g) CPDEA; (h) MMO-GD

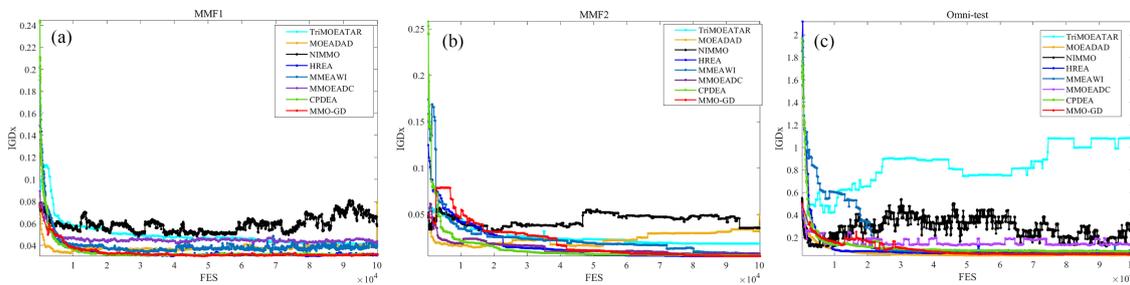


图 12 算法在不同测试函数上的 IGDx 值收敛曲线. (a) MMF1; (b) MMF2; (c) Omni-test

Figure 12 IGDx value convergence curves of the algorithm on different test functions. (a) MMF1; (b) MMF2; (c) Omni-test

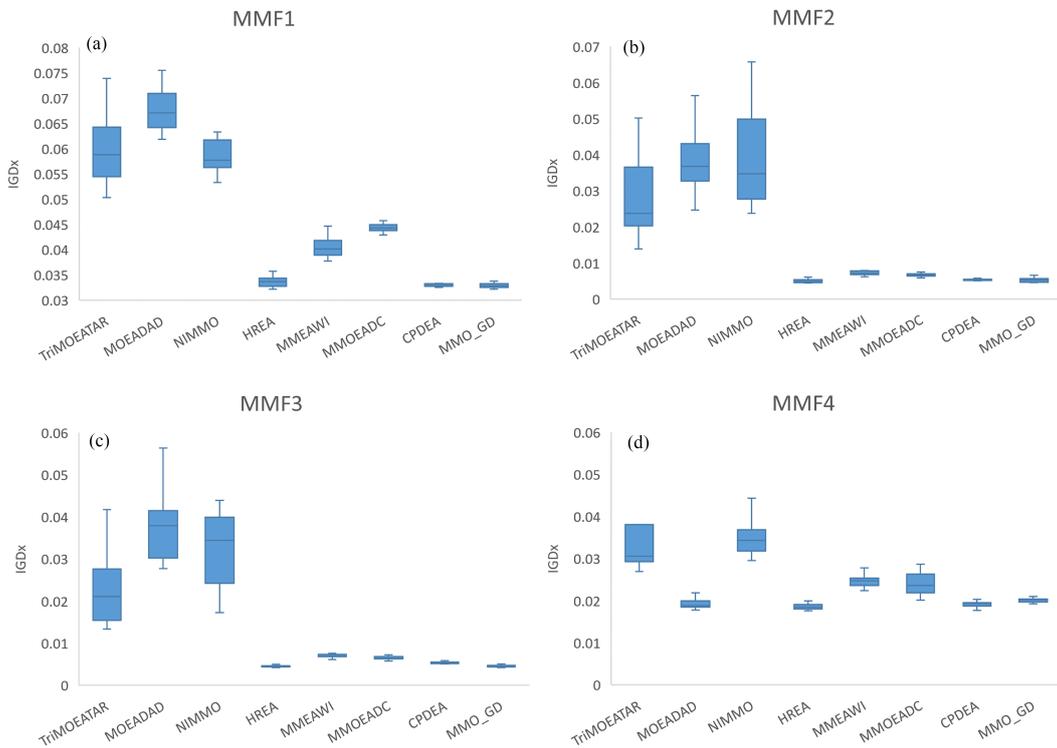


图 13 (网络版彩图) IGDx 的箱线图比较. (a) MMF1; (b) MMF2; (c) MMF3; (d) MMF4  
Figure 13 (Color online) IGDx boxplot comparison. (a) MMF1; (b) MMF2; (c) MMF3; (d) MMF4

表 5 IGDx 均值与标准差统计结果

Table 5 IGDx mean and standard deviation statistical results

Problem	MMO-GD-N+N	MMO-GD-N+1	MMO-GD
MMF1	1.1838e-1 (1.69e-2) –	3.9769e-2 (1.53e-3) =	<b>3.7878e-2 (1.51e-3)</b>
MMF2	1.6045e-2 (6.27e-2) =	7.4822e-3 (4.49e-3) –	<b>7.1371e-3 (5.35e-2)</b>
MMF3	7.7489e-3 (4.71e-2) =	5.2678e-3 (5.04e-3) =	<b>5.0001e-3 (1.33e-2)</b>
MMF4	4.9820e-2 (9.34e-3) –	2.3785e-2 (1.75e-3) –	<b>2.1145e-2 (5.36e-4)</b>
MMF5	1.8867e-1 (2.40e-2) –	7.1241e-2 (2.06e-3) –	<b>6.3766e-2 (2.60e-3)</b>
MMF6	1.4037e-1 (3.45e-2) –	5.9614e-2 (1.37e-3) –	<b>5.4618e-2 (6.00e-4)</b>
MMF7	5.0137e-2 (4.13e-3) –	2.5308e-2 (6.26e-4) =	<b>2.2438e-2 (2.01e-3)</b>
MMF8	5.0023e-1 (7.11e-2) –	5.6331e-2 (2.35e-3) =	<b>5.3978e-2 (3.85e-3)</b>
+/-/=	0/6/2	0/4/4	

足, 因此能在其基础上取得较好的结果. 本文提出的两阶段算法框架根据进化程度控制子代数量, 并没有增加时间复杂度, 与对比的两个算法框架所用时间相似.

#### 4.6 目标空间自适应变异策略有效性分析

MMO-GD 使用目标空间密度自适应策略维护目标空间多样性, 保证最优子集不被丢失. 本节主要探究目标空间密度自适应策略是否对 MMOEA 产生了积极的影响. 设计目标空间自适应变异策略消融实验时, 首先我们考虑选取不同最优子集数量的测试函数. 因此实验选取 + 测试函数 MMF1~MMF4、

表 6 IGDf 均值与标准差统计结果

Table 6 IGDf mean and standard deviation statistical results

Problem	MMO-GD-N+N	MMO-GD-N+1	MMO-GD
MMF1	8.7314e-3 (1.53e-3) –	3.8377e-3 (1.63e-4) =	<b>3.7746e-3 (2.07e-4)</b>
MMF2	3.8543e-3 (2.97e-2) =	3.4888e-3 (2.33e-3) =	<b>3.3019e-3 (6.50e-3)</b>
MMF3	2.8324e-3 (1.62e-2) =	2.8126e-3 (1.25e-3) =	<b>2.7620e-3 (5.40e-3)</b>
MMF4	4.0852e-3 (2.69e-4) –	3.9567e-3 (2.01e-4) =	<b>3.6467e-3 (2.07e-4)</b>
MMF5	7.9868e-3 (1.84e-3) –	4.1102e-3 (1.27e-4) –	<b>3.5982e-3 (1.12e-4)</b>
MMF6	4.9245e-3 (3.81e-4) –	3.7682e-3 (2.32e-4) =	<b>3.3997e-3 (2.27e-4)</b>
MMF7	4.7610e-3 (7.75e-4) =	4.1548e-3 (1.66e-4) –	<b>3.9466e-3 (1.03e-4)</b>
MMF8	4.7841e-3 (2.04e-4) =	5.8018e-3 (5.87e-4) –	<b>4.1132e-3 (4.48e-4)</b>
+/-/=	0/4/4	0/3/5	

表 7 不同变异策略 IGDx 均值与标准差统计结果

Table 7 Statistical results of IGDx mean and standard deviation with different mutation strategies

Problem	MMO-GD-random	MMO-GD
MMF1	3.8619e-2 (3.92e-4) –	<b>3.1209e-2 (1.43e-4)</b>
MMF2	1.6279e-2 (6.52e-3) –	<b>9.2961e-3 (6.61e-3)</b>
MMF3	2.2794e-2 (2.58e-2) –	<b>4.6262e-3 (3.81e-4)</b>
MMF4	1.9846e-2 (7.72e-4) =	<b>1.9695e-2 (4.44e-4)</b>
MMF1_e	2.2446e-2 (6.41e-1) =	<b>3.1440e-3 (4.05e-1)</b>
Omni-test	7.1380e-2 (5.39e-3) –	<b>4.9609e-2 (1.51e-3)</b>
SYM-rotated	5.2509e-2 (2.02e-3) =	<b>5.2099e-2 (2.47e-3)</b>
+/-/=	0/4/3	

表 8 不同变异策略 IGDf 均值与标准差统计结果

Table 8 Statistical results of IGDf mean and standard deviation with different mutation strategies

Problem	MMO-GD-random	MMO-GD
MMF1	4.1442e-3 (3.62e-4) =	<b>3.8927e-3 (3.15e-4)</b>
MMF2	2.7217e-2 (1.38e-4) –	<b>2.7878e-3 (2.10e-4)</b>
MMF3	2.6298e-2 (1.98e-4) –	<b>2.6972e-3 (7.20e-5)</b>
MMF4	3.7111e-3 (1.14e-4) =	<b>3.6989e-3 (5.78e-5)</b>
MMF1_e	5.7135e-3 (7.53e-5) –	<b>2.8984e-3 (2.78e-4)</b>
Omni-test	1.9381e-2 (6.87e-3) –	<b>1.1478e-2 (3.82e-4)</b>
SYM-rotated	1.5049e-2 (2.09e-3) =	<b>1.3879e-2 (2.53e-3)</b>
+/-/=	0/4/3	

MMF1\_e、Omni-test、SYM-rotated, 它们分别有 2, 4, 9, 27 个最优子集, 目的是证明目标空间自适应变异策略在不同数量最优子集均能有效. 对比算法在 MMO-GD 的基础上, 不使用目标空间密度自适应策略, 改为随机选择父代进行变异, 以此验证目标空间密度自适应策略有效性和必要性, 我们将对比算法命名为 MMO-GD-random. 实验中各测试函数总评价次数为 100000 次, 这 7 种测试函数都进行 30 次独立重复实验, 利用函数评价次数作为终止条件获得的平均 IGDx 和 IGDf 指标值如表 7 和表 8 所示.

表 7 和表 8 记录了不同变异策略 IGDx 和 IGDf 均值与标准差统计结果, 实验结果显示在这 7 个测试函数中, MMO-GD 使用目标空间密度自适应变异策略的 IGDx 和 IGDf 大多数值优于随机变异策

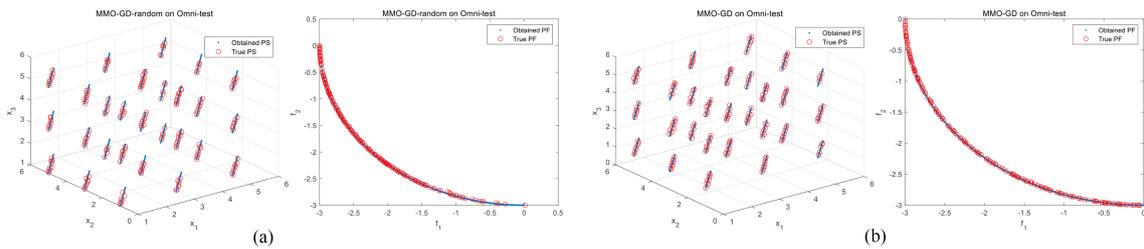


图 14 (网络版彩图) 不同变异策略在 Omni-test 结果分布图. (a) MMO-GD-random; (b) MMO-GD  
 Figure 14 (Color online) Distribution map of Omni-test results with different mutation strategies. (a) MMO-GD-random; (b) MMO-GD

表 9 真实地图问题实验结果

Table 9 Experimental results of real map problems

Problem	IGD <sub>x</sub>	IGD <sub>f</sub>
TriMOEATAR	2.3868e+0 (7.31e-1) –	2.7768e+0 (4.52e-1) –
MOEADAD	1.7913e+0 (5.92e-2) –	2.2793e+0 (9.80e-2) –
NIMMO	1.4305e+0 (5.56e-2) –	1.6292e+0 (4.19e-2) =
HREA	1.0023e+0 (1.26e-1) –	1.2929e+0 (1.63e-1) =
MMEAWI	9.2262e-1 (2.40e-2) =	1.1881e+0 (3.75e-2) =
MMOEADC	9.5801e-1 (1.49e-1) =	1.1953e+0 (1.90e-1) =
CPDEA	9.5092e-1 (8.60e-3) =	1.1991e+0 (1.39e-2) =
MMO-GD	9.5632e-1 (9.34e-2)	1.2298e+0 (1.12e-1)

略的 MMO-GD-random, 这表明目标空间自适应变异策略的优越性. 虽然随机变异也能取得较好的结果, 但目标空间自适应变异策略能更好的平衡决策空间和目标空间的多样性. 为了更直观的展示, 画出有 27 个优子集的 Omni-test 决策空间和目标空间结果分布图, 如图 14 所示, MMO-GD 在目标空间能均匀分布与收敛, 并且均匀分布到决策空间每个最优子区域, 而 MMO-GD-random 在目标空间存在分布不均匀的情况, 而且也影响了决策空间各个子最优区域的分布. 因此本文提出的目标空间密度自适应变异策略, 能有效解决目标空间分布不均的问题.

#### 4.7 实际问题

本文实际问题检验算法是由 Hiso 等人提出<sup>[36]</sup>. 该问题基于一张真实地图构建, 其中包含了 4 个类别的目标点: 六所小学、三所初中、十三家便利店以及三个火车站, 详细信息请参阅文献 [36]. 本问题旨在找出距离每种场所类型最近的距离.

表 9 显示了真实地图问题上 8 种对比算法获得的 IGD<sub>x</sub> 和 IGD<sub>f</sub> 值. 从表 9 可以看出, MMEAWI 在真实地图问题上表现最好. 所提出的 MMO-GD 的性能与 MMEAWI 类似, 并获得了比 TriMOEATAR、MOEADAD 和 NIMMO 更好的 IGD<sub>x</sub> 和 IGD<sub>f</sub>. 此外, 图 15 显示了 8 种对比算法在决策空间中的实验结果. 如图所示, MMO-GD 得到的解在 3 个帕累托子区域都有很好的收敛和分布. 因此该算法可以有效地解决基于地图的真实测试问题.

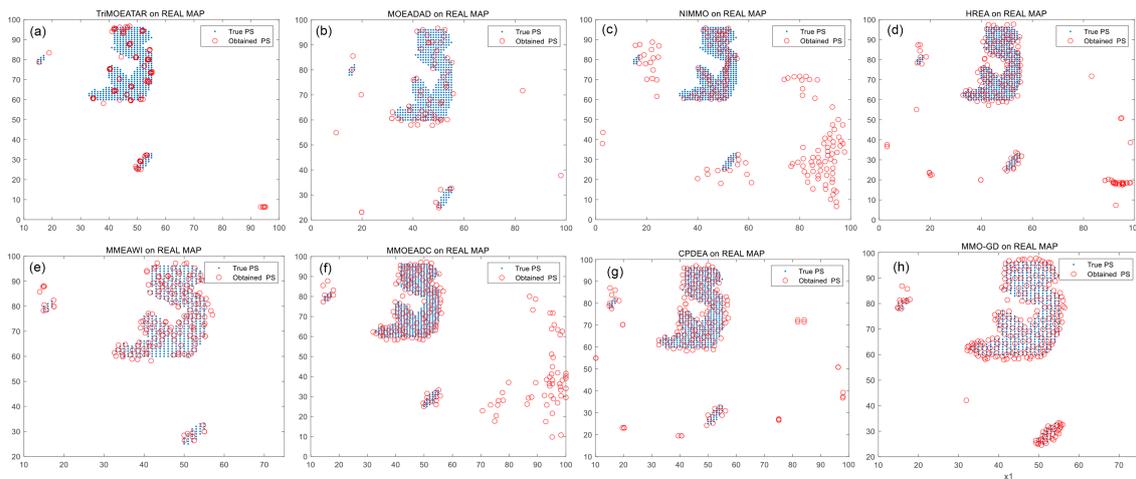


图 15 (网络版彩图) 各算法在真实地图问题的结果分布. (a) TriMOEATAR; (b) MOEADAD; (c) NIMMO; (d) HREA; (e) MMEAWI; (f) MMOEADC; (g) CPDEA; (h) MMO-GD

Figure 15 (Color online) Distribution of results of each algorithm in real-map problems. (a) TriMOEATAR; (b) MOEADAD; (c) NIMMO; (d) HREA; (e) MMEAWI; (f) MMOEADC; (g) CPDEA; (h) MMO-GD

## 5 结论

本文提出基于全局密度更新策略的两阶段多模态多目标进化算法 MMO-GD. 针对现有 MMOEA 算法框架搜索和开发子代不足的问题, 设计了一种两阶段的多模态多目标进化算法框架. 还针对目前存在的 MMOEA 寻找决策空间更多等价解的不足, 以及保持决策空间和目标空间多样性和收敛性的平衡的问题, 在目标空间中, 提出目标空间密度自适应变异策略, 用于保持目标空间的多样性; 在决策空间中, 提出全局密度估计种群更新策略, 用于寻找决策空间等价解, 保持决策空间的多样性. 为验证 MMO-GD 的性能, 我们与另外 7 种有代表性的算法进行比较. 结果表明, 本文算法取得了较好性能, 能取得较好的收敛性和多样性.

虽然所提出的 MMO-GD 可以有效地找到多组等效 PS, 但找到的局部帕累托最优解仍有改进的空间. 例如, 由于一些局部最优解可能由全局最优解主导, 因此获得局部 PS 是非常困难的. 未来可以设计基于指标的方法并加入一些策略来提高局部最优解的搜索性能.

## 参考文献

- 1 Jiang F, He J, Zeng Z, et al. Decomposition-optimization-ensemble learning approach for electricity price forecasting. *Sci Sin Inform*, 2018, 48: 1300–1315 [蒋锋, 何佳琪, 曾志刚, 等. 基于分解优化集成学习方法的电价预测. *中国科学: 信息科学*, 2018, 48: 1300–1315]
- 2 Qian C, Zhou Z H. Decomposition-based Pareto optimization for subset selection. *Sci Sin Inform*, 2016, 46: 1276–1287 [钱超, 周志华. 基于分解策略的多目标演化子集选择算法. *中国科学: 信息科学*, 2016, 46: 1276–1287]
- 3 Zhang X, Tian Y, Cheng R, et al. A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization. *IEEE Trans Evol Computat*, 2018, 22: 97–112
- 4 Zhang X, Tian Y, Jin Y. A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Computat*, 2015, 19: 761–776
- 5 Tang K, Peng F, Chen G, et al. Population-based algorithm portfolios with automated constituent algorithms selection. *Inf Sci*, 2014, 279: 94–104
- 6 Tang K, Mei Y, Yao X. Memetic algorithm with extended neighborhood search for capacitated arc routing problems.

- IEEE Trans Evol Computat, 2009, 13: 1151–1166
- 7 Zhang K, Shen C, He J, et al. Knee based multimodal multi-objective evolutionary algorithm for decision making. *Inf Sci*, 2021, 544: 39–55
  - 8 Wang Y, Li H X, Yen G G, et al. MOMMOP: multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems. *IEEE Trans Cybern*, 2015, 45: 830–843
  - 9 Zhang K, Shen C, Yen G G, et al. Two-stage double niched evolution strategy for multimodal multiobjective optimization. *IEEE Trans Evol Computat*, 2021, 25: 754–768
  - 10 Gao W, Gao Q, Sun L, et al. Design of a novel multimodal optimization algorithm and its application in logistics optimization. *Electron. Res. Arch*, 2024, 32(3): 1946–1972
  - 11 Yue C T, Liang J J, Qu B Y, et al. Multimodal multiobjective optimization in feature selection. In: *Proceedings of 2019 IEEE Congress on Evolutionary Computation*, 2019. 302–309
  - 12 Yang X, Yuan X, Dong L, et al. Manifold assistant multi-modal multi-objective differential evolution algorithm and its application in actual rolling bearing fault diagnosis. *Eng. Appl. Artif. Intell*, 2024, 133: 108040
  - 13 Tanabe R, Ishibuchi H. A review of evolutionary multimodal multiobjective optimization. *IEEE Trans Evol Computat*, 2019, 24: 193–200
  - 14 Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Computat*, 2002, 6: 182–197
  - 15 Deb K, Tiwari S. A procedure for single and multi-objective optimization. In: *carlos A. coello coello, arturo hern'andez aguirre, Eckart Zitzler, eds. Omni-Optimizer. Berlin: Springer Berlin Heidelberg, 2005. 47–61*
  - 16 Liu Y, Ishibuchi H, Nojima Y, et al. A double-niched evolutionary algorithm and its behavior on polygon-based problems. In: *Proceedings of Parallel Problem Solving from Nature-PPSN XV: 15th International Conference, 2018. 262–273*
  - 17 Liang J J, Yue C T, Qu B Y. Multimodal multi-objective optimization: a preliminary study. In: *Proceedings of 2016 IEEE Congress on Evolutionary Computation, 2016. 2454–2461*
  - 18 Zhang Q, Hui L. MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Computat*, 2008, 11: 712–731
  - 19 Liu Y, Yen G G, Gong D. A multimodal multiobjective evolutionary algorithm using two-archive and recombination strategies. *IEEE Trans Evol Computat*, 2018, 23: 660–674
  - 20 Tanabe R, Ishibuchi H. A decomposition-based evolutionary algorithm for multi-modal multi-objective optimization. In: *Proceedings of Parallel Problem Solving from Nature-PPSN XV: 15th International Conference, 2018. 249–261*
  - 21 Yue C, Qu B, Liang J. A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems. *IEEE Trans Evol Computat*, 2018, 22: 805–817
  - 22 Liang J, Guo Q, Yue C, et al. A self-organizing multi-objective particle swarm optimization algorithm for multimodal multi-objective problems. In: *Proceedings of Advances in Swarm Intelligence: 9th International Conference, 2018. 550–560*
  - 23 Liang J, Xu W, Yue C, et al. Multimodal multiobjective optimization with differential evolution. *Swarm Evolary Computation*, 2019, 44: 1028–1059
  - 24 Qu B, Li C, Liang J, et al. A self-organized speciation based multi-objective particle swarm optimizer for multimodal multi-objective problems. *Appl Soft Computing*, 2020, 86: 105886
  - 25 Liu Y, Ishibuchi H, Yen G G, et al. Handling imbalance between convergence and diversity in the decision space in evolutionary multimodal multiobjective optimization. *IEEE Trans Evol Comput*, 2020, 24: 551–565
  - 26 Hu C, Ishibuchi H. Incorporation of a decision space diversity maintenance mechanism into MOEA/D for multi-modal multi-objective optimization. In: *proceedings of the genetic and evolutionary computation conference companion, 2018. 1898–1901*
  - 27 Ulrich T, Bader J, Thiele L. Defining and optimizing indicator-based diversity measures in multiobjective search. In: *Proceedings of International Conference on Parallel Problem Solving from Nature, 2010. 707–717*
  - 28 Ishibuchi H, Yamane M, Akedo N, et al. Two-objective solution set optimization to maximize hypervolume and decision space diversity in multiobjective optimization. In: *Proceedings of The 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems, 2012. 1871–1876*

- 29 Tanabe R, Ishibuchi H. A niching indicator-based multi-modal many-objective optimizer. *Swarm Evolary Computation*, 2019, 49: 134–146
- 30 Yuan Y, Xu H, Wang B, et al. Balancing convergence and diversity in decomposition-based many-objective optimizers. *IEEE Trans Evol Computat*, 2015, 20: 180–198
- 31 Li W, Yao X, Zhang T, et al. Hierarchy ranking method for multimodal multiobjective optimization with local pareto fronts. *IEEE Trans Evol Computat*, 2023, 27: 98–110
- 32 Li W, Zhang T, Wang R, et al. Weighted indicator-based evolutionary algorithm for multimodal multiobjective optimization. *IEEE Trans Evol Computat*, 2021, 25: 1064–1078
- 33 Lin Q, Lin W, Zhu Z, et al. Multimodal multiobjective evolutionary optimization with dual clustering in decision and objective spaces. *IEEE Trans Evol Computat*, 2020, 25: 130–144
- 34 Yue C, Qu B, Yu K, et al. A novel scalable test problem suite for multimodal multiobjective optimization. *Swarm Evolary Computation*, 2019, 48: 62–71
- 35 Zhou A, Zhang Q, Jin Y. Approximating the set of pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm. *IEEE Trans Evol Computat*, 2009, 13: 1167–1189
- 36 Ishibuchi H, Akedo N, Nojima Y. A many-objective test problem for visually examining diversity maintenance behavior in a decision space. In: *proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, 2011. 649–656

## Two-stage multimodal multi-objective evolutionary algorithm based on global density updating strategy

Juanjuan HE<sup>1,2</sup>, Qian YANG<sup>1,2</sup>, Zhiwei XU<sup>1,2</sup>, Kai ZHANG<sup>1,2</sup>, Xingyi ZHANG<sup>3</sup> & Mingfeng GE<sup>4\*</sup>

1. *School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065, China;*

2. *Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, Wuhan 430065, China;*

3. *School of Computer Science and Technology, Anhui University, Hefei 230093, China;*

4. *School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan 430074, China*

\* Corresponding author. E-mail: gemf@cug.edu.cn

**Abstract** Multimodal multi-objective problems involve identifying global and local optimal Pareto solution sets. Providing decision-makers with comprehensive Pareto sets is crucial for theoretical understanding and practical application. In recent years, numerous evolutionary algorithms have been proposed to address these issues. However, most of these algorithms first select individuals based on their convergence properties to form mating pools and then consider individuals with good diversity in decision and objective spaces. This method often leads to the replacement of individuals with characteristics in the decision space by those who exhibit strong convergence in the objective area. Additionally, achieving uniform convergence of individuals to each Pareto subregion is challenging because of the varying shapes and positions of Pareto optimal subsets in the decision space. In this paper, we introduce a novel two-stage multimodal multi-objective evolutionary algorithm that employs a global density updating strategy. First, we propose an alternative two-stage optimization framework to address the limitations of traditional mating pools and one-to-one parent-progeny comparisons. The proposed framework dynamically adjusts the population size at different stages and uses distinct optimization methods, promoting a balanced exploration and exploitation of the population. Second, to resolve the uneven distribution of Pareto subsets, we designed adaptive and global density estimation strategies that update the objective space density and population. These mechanisms ensure the maintenance of diversity within the objective and decision spaces. We compared the proposed algorithm with seven representative multimodal multi-objective algorithms. The experimental results demonstrate that the proposed algorithm finds equivalent solutions in the decision space and effectively balances diversity and convergence between the decision and objective space. Overall, the proposed algorithm outperforms the compared algorithms.

**Keywords** multi-objective evolutionary algorithm, multimodal multi-objective optimization problem, evolutionary algorithm, global density, reference vector, boundary point clustering