

# 基于投票机制和动态分配价值点的集成超目标优化算法

刘新平<sup>1</sup>, 孙越泓<sup>1,2</sup>, 刘佛祥<sup>2,3</sup>

(1. 南京师范大学数学科学学院, 江苏 南京 210023)

(2. 大规模复杂系统数值模拟教育部重点实验室, 江苏 南京 210023)

(3. 南昌大学信息工程学院, 江西 南昌 330029)

**[摘要]** 在超目标优化问题中, 目标之间的冲突会导致没有一个解可以同时优化所有目标, 求解时存在大量非支配解。选择合适的解排序算法评估解的质量, 对算法性能起着关键作用。而不同的解排序算法, 在处理不同的超目标问题时有着各自的优劣。因此, 本文提出一个基于投票机制和动态分配价值点的集成框架 (ensemble many-objective evolutionary algorithm based on voting and dynamic value point, VDVP-EMEA), 将不同解排序算法聚合在一起协同工作。首先, 根据每种解排序算法的有效投票率, 动态分配每个专家拥有的价值点, 有效投票越多的解排序算法, 相应赋予更多的价值点, 反之则对价值点进行惩罚。然后使用末位淘汰制, 废弃能力最差的专家的投票。其次, 在环境选择过程中, 使用精英选择策略, 通过投票结果和价值点来定义个体适应度, 适应度越大的个体越优先被选择。最后, 为了测试 VDVP-EMEA 算法的性能, 进行大量试验, 将 VDVP-EMEA 与 4 种常用的单一解排序算法 NSGA-III, SPEA2, BiGE, GrEA 和一种先进的集成算法 VMEF 进行了比较。实验结果表明, VDVP-EMEA 的收敛性和多样性明显优于这些算法。

**[关键词]** 集成框架, 投票机制, 超目标优化问题, 解排序算法

**[中图分类号]** TP18 **[文献标志码]** A **[文章编号]** 1001-4616(2024)04-0059-09

## An Ensemble Many-objective Optimization Algorithm Based on Voting and Dynamic Value Point

Liu Xinping<sup>1</sup>, Sun Yuehong<sup>1,2</sup>, Liu Foxiang<sup>2,3</sup>

(1. School of Mathematical Sciences, Nanjing Normal University, Nanjing 210023, China)

(2. Ministry of Education Key Laboratory of NSLSCS, Nanjing 210023, China)

(3. School of Information Engineering, Nanchang University, Nanchang 330029, China)

**Abstract:** In many-objective optimization problems, the conflicts among the objectives lead to the situation where solutions cannot optimize all objectives simultaneously, and a large number of non-dominated solutions exist in the process. An appropriate solution-sorting method plays a crucial role in evaluating the quality of solutions and the performance of the algorithm. Generally, different solution-sorting methods have their own pros and cons when handling different many-objective problems. Therefore, an ensemble many-objective evolutionary algorithm based on voting and dynamic value point (VDVP-EMEA) is proposed, which can integrate different solution-sorting methods and cooperate together. First, the value points of each expert are dynamically allocated by the voting success rate of each solution-sorting method, and the solution-sorting method with more voting success rate will be correspondingly assigned more value points. Otherwise, the value points will be punished. The last elimination system is used to cancel the votes of the most inefficient experts. The elite selection strategy is used to define fitness of individuals by voting results and value points, and the individuals with greater fitness value are preferentially selected in the process of environment selection. Finally, a large number of experiments are conducted to test the performance of VDVP-EMEA, and VDVP-EMEA is compared with five advanced many-objective evolutionary algorithms NSGA-III, SPEA2, BiGE, GrEA, and VMEF. Experimental results indicate that the overall performance of VDVP-EMEA is distinctly better than these algorithms.

**Key words:** ensemble framework, voting mechanism, many-objective optimization problems, solution-sorting algorithms

收稿日期: 2024-06-25.

基金项目: 大规模复杂系统数值模拟教育部重点实验室开放课题基金资助项目 (202409)、国家自然科学基金项目 (12471290)、江西省自然科学基金项目 (20224BAB212003).

通讯作者: 孙越泓, 博士, 副教授, 研究方向: 智能优化及图像处理. E-mail: 05234@njjnu.edu.cn

## 1 相关介绍

大多数现实世界的优化问题需要同时优化多个目标,这些目标互相之间通常是冲突的.例如在经济、军事、管理和工程设计等领域,衡量一个方案的好坏往往需要用多个目标来比较.这样针对多个目标的优化问题,称为多目标优化问题(multi-objective optimization problems, MOPs).若一个多目标优化问题有超过 3 个以上的目标,则称为超目标优化问题(many-objective optimization problems, MaOPs).首先,给出多目标优化问题的一般描述.

设有  $m$  个目标且可以相互冲突,同时最小化  $m$  个目标可表示为

$$\begin{cases} \min \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})) \\ \text{s.t. } \mathbf{x} \in X, \mathbf{f} \in Y, \end{cases} \quad (1)$$

式中,  $X \in \mathbb{R}^n$  是决策空间,  $\mathbf{x} = (x_1, x_2, \dots, x_n) \in X$  是决策向量,  $Y \in \mathbb{R}^m$  是目标空间,  $\mathbf{f} \in Y$  是由  $m$  个目标函数  $f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})$  将  $\mathbf{x}$  从  $X$  映射到  $Y$  得到的目标向量.

由于目标之间存在冲突,一个目标的改进可能导致其他目标的退化,因此不存在一个解可以同时优化所有目标,但存在一组在不同目标之间进行权衡的最优解集.

在超目标优化中,一个解  $\mathbf{x}_1$  支配另一个解  $\mathbf{x}_2$  当且仅当:

$$\begin{cases} \forall i \in 1, 2, \dots, m, f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2) \\ \exists j \in 1, 2, \dots, m, f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2) \end{cases}, \quad (2)$$

若一个解  $\mathbf{x}^*$  不能被其他可行解支配,则称解  $\mathbf{x}^*$  为 Pareto 最优解.所有的 Pareto 最优解  $\mathbf{x}^*$  的并集称为 Pareto 最优解集(Pareto optimal set, PS).所有  $\mathbf{f}(\mathbf{x}^*)$  的并集称为 Pareto 最优前沿(Pareto front, PF).大多数连续的 MOPs 的 PS 是由无限个 Pareto 最优解组成,因此实际只获取一个具有代表性的 PS 的子集.注意,多目标优化算法获得的最终非支配解集不一定满足 Pareto 最优.

在求解超目标优化问题时,存在大量的非支配解,环境选择的本质是用解排序算法将非支配解中的优势个体赋予更高的排名,以便于保存到下一代.每个解排序算法在求解不同超目标优化问题时有不同的优势,因此,本文考虑集成不同的解排序算法,以减轻环境选择的压力.

近年来,集成策略被广泛研究用于求解多目标优化问题.首先,集成不同进化算子或集成同种算子的不同变体以提高算法的寻优能力. Lin 等<sup>[1]</sup>集成了二项式交叉和差分进化两种交叉算子,充分利用两种不同搜索策略的互补特性;文献[2]集成了两种差分进化算子的变体和二项式交叉算子,组合成重组算子来增强搜索能力;Wu 等<sup>[3]</sup>集成了带有 3 种突变策略的差分进化算子进行搜索;Gong 等<sup>[4]</sup>集成了两种带有自适应策略选择的差分进化算子进行搜索.其次,将集成思想与其他策略相结合. Wang 等<sup>[5]</sup>基于多种群策略,集成不同进化算子,并结合资源分配方法处理多目标优化问题. Pal 等<sup>[6]</sup>将集成思想与问题转换策略结合起来,把多目标问题转换成多个单目标问题,集成多个单目标进化算法来求解多目标优化问题.最后,一些学者在其他方面对集成策略进行研究. Mallipeddi 等<sup>[7]</sup>集成了 4 种约束处理技术,处理有约束的多目标优化问题. Zhao 等<sup>[8]</sup>在基于分解的多目标进化算法中集成了多种尺寸的邻域.因此,如何在超目标优化问题的求解中有效利用集成技术,值得我们进一步研究.

Qiu 等<sup>[9]</sup>集成不同解排序算法来处理超目标优化问题.受文献[9]的启发,本文提出了一个基于投票机制和动态分配价值点的集成框架(ensemble many-objective evolutionary algorithm based on voting and dynamic value point, VDVP-EMEA),实现多种解排序算法的协同工作,用于求解各种超目标优化问题.与文献[9]直接统计专家票数不同,本文引入价值点概念,用于计算个体适应度,目的是细化不同专家的决策能力,并引入末位淘汰制.本文的贡献如下:

- (1)提出一种新的集成框架,便于不同的解排序算法协同作用,产生更优秀的个体.
- (2)将投票机制与价值点概念结合,重新定义个体的适应度.
- (3)设计一种动态分配价值点的方法.

本文组织架构如下:第二节介绍了 VDVP-EMEA 算法及其具体实现,第三节给出了 VDVP-EMEA 与其他算法对比的实验结果,第四节总结了全文,并提出了未来的研究方向.

## 2 VDVP-EMEA 算法

在我们提出的新集成框架下,每一种解排序算法被视为一位专家. 本文将每一位专家的投票权,进行离散化,称之为价值点. 通过统计票数和价值点,计算所有个体的适应度,适应度值越大,表明个体越有优势. 对于具有相同适应度值的个体,优先保留聚集密度较为稀疏的个体. 每隔一定代数,根据专家的决策效果,重新评估专家的决策能力,从而自适应地动态调整专家的价值点. 同时实行末位淘汰制,即在下次迭代中,废弃决策能力最差专家的投票.

### 2.1 投票机制

设种群大小为  $N$ , 初始化产生种群  $P$ , 通过遗传算法产生子代  $C$ , 将父代种群和子代种群合并成联合种群  $Q$ . 假设解排序算法集合为  $SSM$ , 将决策能力最差的专家索引记为  $Poor$ , 初始时  $Poor = 0$ , 专家对所有个体的投票结果记为矩阵  $Votes$ , 定义方式如下:

定义 1(投票结果矩阵): 第  $i$  个专家对第  $j$  个个体的投票结果记为  $Votes(i, j)$ ,

$$Votes(i, j) = \begin{cases} 1, & \text{若第 } j \text{ 个体被选中} \\ 0, & \text{否则} \end{cases} \quad (3)$$

式中, 若第  $j$  个体被选中, 则对应的投票结果标记为 1, 反之标记为 0. 每个专家共有  $N$  张选票.

#### 算法 1: 投票机制

```

输入: 联合种群  $Q$ ; 解排序算法集合  $SSM$ ; 最差的专家的索引  $Poor$ ;
输出: 投票结果矩阵  $Votes$ ;
1: for  $i = 1 : |SSM|$  do
2: 使用第  $i$  个解排序算法对联合种群  $Q$  进行环境选择;
3: 将被选中的个体标记为 1, 否则标记为 0;
4: 用公式 (3) 把联合种群  $Q$  中所有个体对应的标记存储到  $Votes(i, :)$  中;
5: end for
6: if  $Poor > 0$  do
7:  $Votes(Poor, :) = 0$ ;
8: end if
9: result  $Votes$ 

```

算法 1 展示了投票机制. 第 7 行将最差的专家的投票结果赋为 0, 即无效化其投票. 这样做可以使整体框架偏向更优秀的解排序算法, 加速寻优速度.

### 2.2 精英选择策略

由投票结果矩阵和专家的价值点, 给出个体适应度的计算公式.

定义 2(适应度): 第  $j$  个体的适应度值, 是专家投票结果的加权求和:

$$Fitness(j) = \sum_{i=1}^{|SSM|} ValuePoint(i) \cdot Votes(i, j), \quad (4)$$

式中,  $Fitness(j)$  表示第  $j$  个个体的适应度;  $ValuePoint(i)$  表示第  $i$  个专家的价值点, 即权重, 初始状态下每位专家的价值点  $ValuePoint$  均为 1;  $Votes(i, j)$  表示第  $i$  个专家对第  $j$  个个体的投票结果.

计算联合种群  $Q$  中所有个体的适应度值, 然后将适应度值降序排序, 采用精英选择策略, 筛选前  $N$  个个体到下一代种群中.

### 2.3 动态分配价值点策略

价值点意味着专家的投票权, 价值点越大, 说明专家的决策能力越强. 衡量专家决策能力的标准是使用解排序算法选择出的优秀个体最终进入下一次迭代的个体数目. 若某种解排序算法选择的个体被保留到下一代, 则认为该专家的投票是有效的. 因此, 定义专家的有效投票率如下:

定义 3(有效投票率): 第  $i$  个解排序算法在第  $G$  代的有效率为:

$$ERate_i^G = \frac{EVotes_i^G}{\sum_{i=1}^{|SSM|} EVotes_i^G} \quad (5)$$

式中,  $EVotes_i^G$  为第  $i$  个解排序算法在第  $G$  代的有效投票的个数, 即第  $i$  个专家在第  $G$  代的投票中成功保留优秀个体的数目。

#### 算法 2: 动态分配价值点策略

输入: 下一代种群  $P$ ; 投票结果矩阵  $Votes$ ; 解排序算法集合  $SSM$ ; 当前代数  $G$ ; 间隔代数  $\Delta T$ ;  
输出: 价值点  $ValuePoint$ ;  
1: 计算所有解排序算法在第  $G$  代的有效投票个数  $EVotes_i^G$ ;  
2: 用公式(5)计算所有解排序算法在第  $G$  代的有效率  $ERate_i^G$ ;  
3: if  $\text{mod}(G, \Delta T) = 0$  do  
4: 用公式(6)计算所有解排序算法每  $\Delta T$  代的累计有效投票率  $CERate_i$ ;  
5: 用公式(7)记录累计有效投票率最小的专家索引  $Poor$ ;  
6: 用公式(8)更新所有解排序算法对应的价值点  $ValuePoint(i)$ ;  
7: end if  
8: result  $ValuePoint$

为保持迭代过程中环境选择的稳定性, 无需每一代都评估专家的决策能力。因此, 每隔一定代数  $\Delta T$ , 计算专家的累计有效投票率。

**定义 5**(累计有效投票率): 对第  $i$  个解排序算法每隔  $\Delta T$  代的有效投票率求和, 记为其累计有效投票率:

$$CERate_i = \sum_{g=G-\Delta T+1}^G ERate_i^g. \quad (6)$$

比较所有专家的累计有效投票率, 将决策能力最差的专家的索引记为  $Poor$ :

$$Poor = \arg \min_{i \in I} CERate_i, \quad (7)$$

式中:  $I = \{1, 2, \dots, |SSM|\}$ 。

每隔  $\Delta T$  代, 根据累计有效投票率, 重新评估专家的决策能力, 其价值点更新形式如下:

**定义 6**(价值点更新公式): 若  $CERate_i$  表示第  $i$  个解排序算法在每  $\Delta T$  代的累计有效投票率,  $|SSM|$  表示解排序算法的数目, 则第  $i$  个解排序算法的价值点  $ValuePoint(i)$  更新公式为:

动态分配价值点策略详见算法 2。算法 2 第 6 行根据累计有效投票率, 动态分配价值点。需要注意所有专家对应的价值点总和保持不变, 即  $\sum_{i=1}^{|SSM|} ValuePoint(i) = |SSM|$ 。

#### 2.4 VDVP-EMEA 总体框架

本文提出的 VDVP-EMEA 的总体框架详见算法 3。通过专家投票机制将多种解排序算法协同作用(第 5 行); 根据投票结果和价值点, 定义新的适应度值(第 6 行), 使用精英选择策略产生新的种群(第 7 行); 计算专家累计有效投票率, 根据其决策能力按比例分配解排序算法的价值点(第 8 行), 进而动态调整专家的决策权。

#### 算法 3: VDVP-EMEA 总体框架

输入: 种群大小  $N$ ; 最大评估次数  $MaxFE$ ; 解排序算法集合  $SSM$ ;  
输出: 最优种群  $P$ ;  
1: 初始化种群  $P$ ; 评估次数  $FEs = 0$ ; 迭代次数  $G = 0$ ;  $ValuePoint = 1 \times |ssm|$ ;  $Poor = 0$ ;  $Votes = \Phi$ ;  
2: while  $FEs \leq MaxFE$   
3: 种群  $P$  通过进化算法产生子代  $C$ ;  
4: 生成联合种群  $Q = P \cup C$ ;  
5: 用投票机制产生所有专家的投票矩阵  $Votes$ (算法 1), 更新  $FEs$ ;  
6: 用公式(4)计算  $Q$  中个体的适应度值, 并按降序排序;  
7: 根据精英选择策略, 筛选出前  $N$  个个体到下一代种群  $P$  中;  
8: 使用动态分配价值点策略, 更新解排序算法的价值点(算法 2);  
9:  $G = G + 1$ ;  
10: end while  
11: return  $P$

### 3 性能评估

在本节中,将本文算法与一种集成算法 VMEF<sup>[9]</sup>以及 NSGA-III<sup>[10]</sup>、SPEA2<sup>[11]</sup>、GrEA<sup>[12]</sup>、BiGE<sup>[13]</sup>这4种单一的解排序算法进行对比实验,考察本算法的多样性和收敛性. NSGA-III 和 SPEA2 是基于 Pareto 支配的超目标进化算法, BiGE 将多目标优化问题转换成双目标优化问题. GrEA 通过使用3种基于网格的指标,保持了收敛性和多样性之间的良好平衡. 由于上述4种解排序算法性质不同,和 VMEF 算法一样,本文也选择将这4种算法一同纳入到新算法框架中. 本文算法和所有比较算法都在 MATLAB2020b 下实现,使用的 CPU 是 Intel(R) Core(TM) i5-6500 CPU@ 3.20 GHz 3.19 GHz, RAM 8.00 GB,并嵌入到进化多目标优化平台 PlatEMO<sup>[14]</sup>中,遵循 PlatEMO 中所有比较算法的默认设置.

#### 3.1 实验设置

(1) 基准问题:选取 WFG(walking fish group)测试函数套件中的 WFG1-WFG9<sup>[15]</sup>这9个基准测试问题,分别设置5、8、10、15和20个目标. 这些基准测试问题具有欺骗、偏移和多模特性,涵盖不规则的 PF 轮廓、复杂的 PS 和多模态等多种情况.

(2) 种群大小:根据测试问题的目标数来设置种群大小. 对 M 个目标的优化问题,使用边界交叉(normal-boundary intersection, NBI)<sup>[16]</sup>,采用两层向量生成策略. 对于5、8、10、15和20个目标的问题,种群大小分别设置为212、156、275、136和230.

(3) 运行次数和停止准则:每个算法在基准测试实例上均独立运行30次,以达到函数最大评估次数作为算法终止条件. 因为 WFG3 存在许多局部最优解,求解难度较大, WFG3 对应的最大函数评估次数设置为150 000. 其余测试函数最大函数评估次数设置为100 000.

(4) 评价指标:反向世代距离(inverted generational distance, IGD)指标<sup>[17]</sup>是评估多目标进化算法获得 Pareto 解质量的重要指标. 计算 IGD 需要使用构成真实 PF 的参考点,计算模型如下:

$$IGD(P^*, P) = \frac{\sum_{p \in P^*} D_{\min}(P^*, P)}{|P^*|}, \quad (9)$$

式中,  $P$  表示待评价种群,  $P^*$  表示参考点集合,  $D_{\min}(P^*, P)$  表示从点  $P^*$  到  $P$  中所有点的最小欧几里德距离. 获得 IGD 值较小的算法意味着种群具有更好的多样性和收敛性. 参考点数目设置为10 000个.

#### 3.2 性能表现

对于本文的45个基准测试实例,表1为6种算法独立运行30次的平均 IGD 值,括号中为 IGD 值的方差,采取5%显著性水平下的 Friedman 秩和检验,表1中最后一行符号-、+和 $\approx$ 分别表示另外5种算法对应的 IGD 值比本文算法差、好和相似. 对于45个测试实例,本文算法在26个测试实例上优于其他5种算法, VMEF、SPEA2、以及 GrEA 分别在5、4、和10个测试实例上得到最好结果, NSGA-III 和 BiGE 在所有测试实例上均未达到最好效果. 这表明本文算法明显比其他5种算法性能更优.

对于可分的单峰问题 WFG1, 本文算法在5、8、15和20目标上性能明显优于其他算法,且在10目标上与最好算法 VMEF 相似.

WFG2 具有断开的 PF, 虽然本文算法不是最优算法,但在8、10、15和20目标上仍优于 GrEA,在8、10和15目标上优于 NSGA-III.

WFG3 是不可分单峰且退化问题,决策空间中存在大量局部最优解,尽管求解难度较大,本文算法仍在5种目标上均展现出最优效果.

WFG4 是不可分的多峰问题, WFG5 是一个欺诈性的测试问题. 本文算法在 WFG4 和 WFG5 的5、15和20目标上优于其他算法,在8和10目标上性能近似于最好算法 GrEA.

WFG6 和 WFG8 是不可分的单峰问题,本文算法在这个问题上的5个测试实例中均优于或者近似优于其他算法.

WFG7 为可分单峰问题,本文算法在5和20目标上性能表现最好,在15目标上性能与最好的 GrEA 相似,在8和10目标上表现超过 VMEF 和 NSGA-III.

WFG9 为不可分多峰问题,并且是多模态的,更具有欺骗性. 本文算法在5、8和15目标上优于其他算

法,在 10 和 20 目标上表现次优.

为便于观察 VMEF、NSGA-III、SPEA2、BiGE 和 GrEA 和本文算法的性能差距,分别绘制 45 个测试实例的对数 IGD 值对比图,见图 1-图 5. 图 1 中,蓝色代表本文算法,红色代表 VMEF. 直观发现,本文算法胜出的测试实例数目比 VMEF 更多. 具体地,本文算法在 WFG2、10 目标的 WFG1、10 目标的 WFG4 和 20 目标的 WFG9 以外的 37 个测试实例上,对数 IGD 值均比集成算法 VMEF 更小. 综合观察图 1-图 5,随目标维数增大,本文算法与其他算法相比,更擅长处理超目标优化问题.

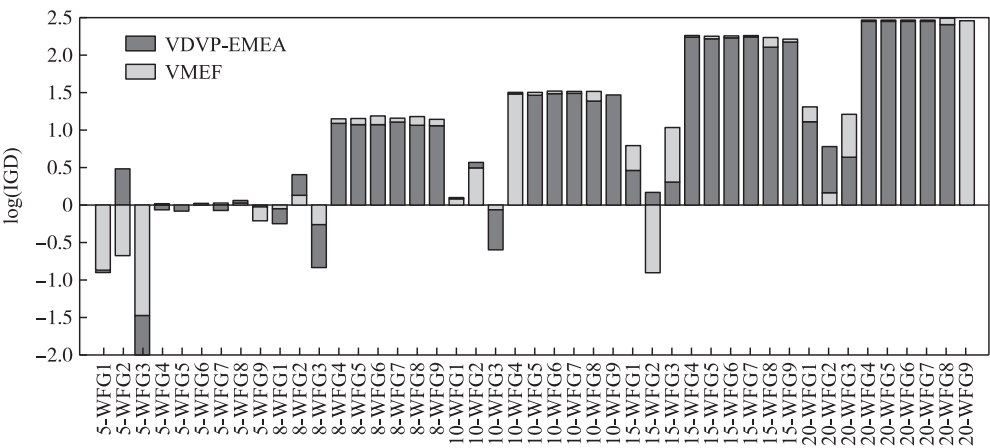


图 1 VDVP-EMEA 与 VMEF 对比图  
Fig. 1 The comparison between VDVP-EMEA and VMEF

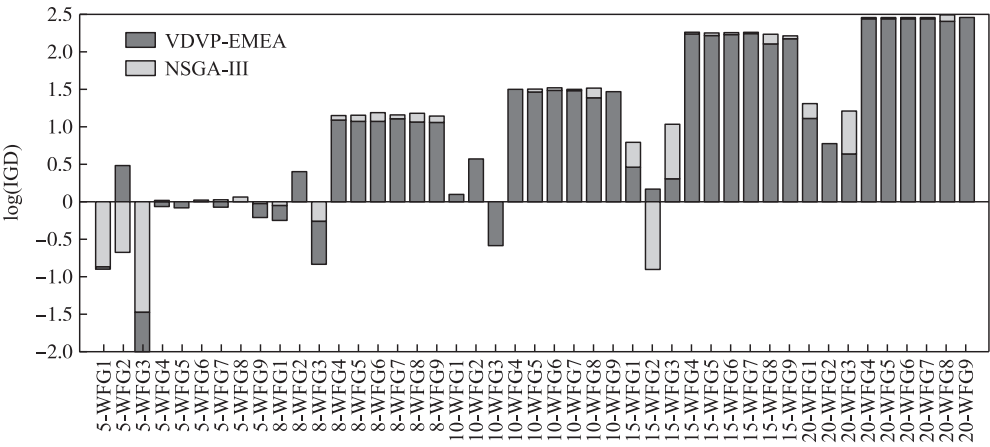


图 2 VDVP-EMEA 与 NSGA-III 对比图  
Fig. 2 The comparison between VDVP-EMEA and NSGA-III

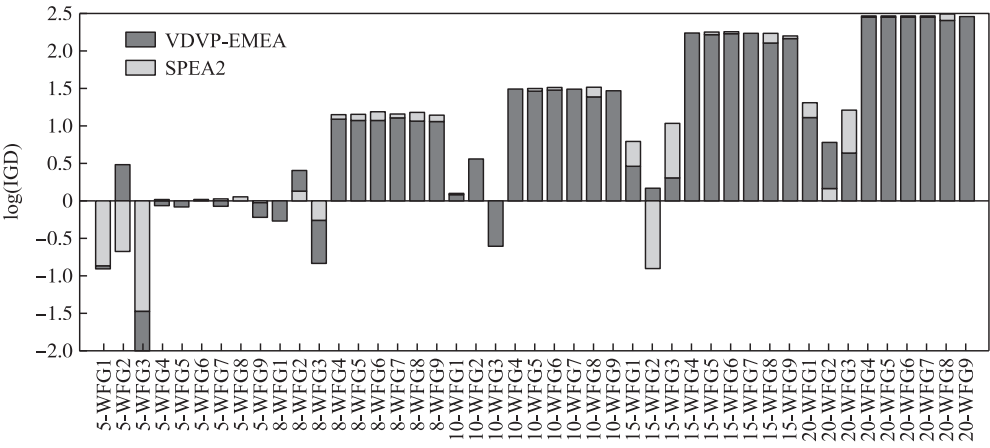


图 3 VDVP-EMEA 与 SPEA2 对比图  
Fig. 3 The comparison of between VDVP-EMEA and SPEA2

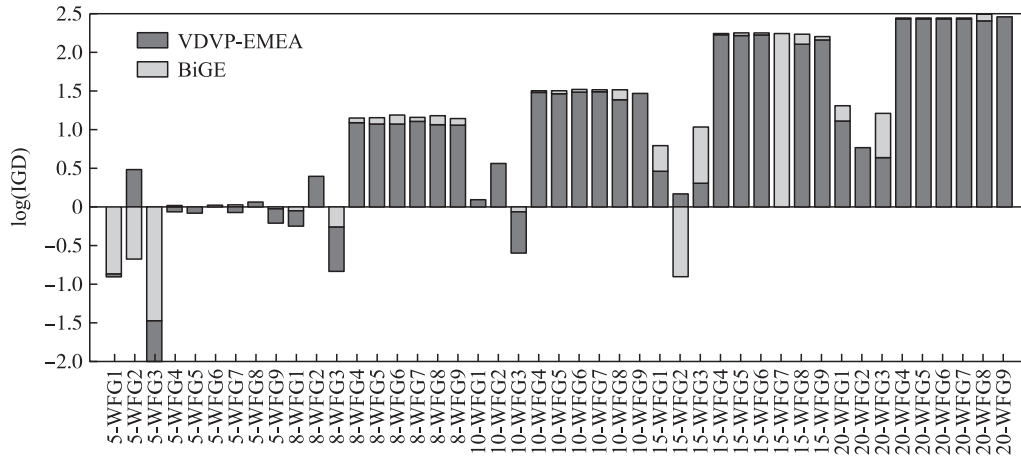


图4 VDVP-EMEA 与 BiGE 对比图

Fig. 4 The comparison of between VDVP-EMEA and BiGE

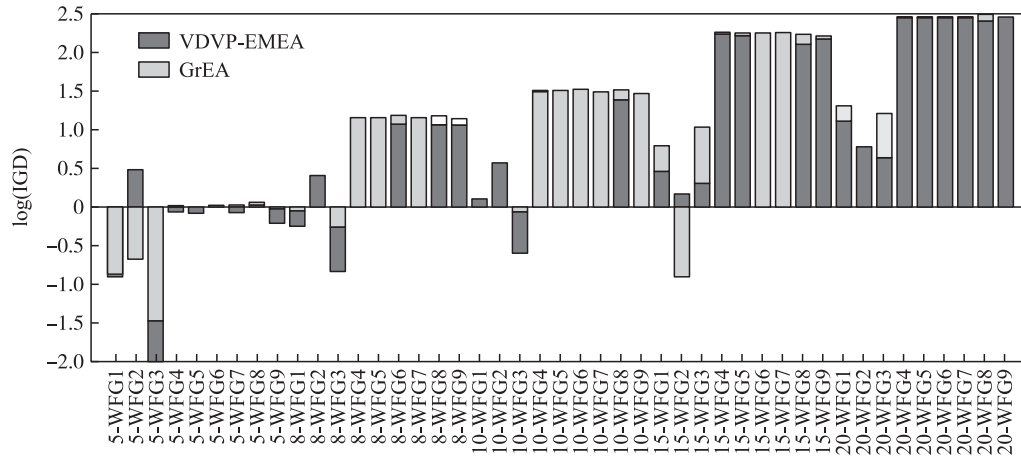


图5 VDVP-EMEA 与 GrEA 对比图

Fig. 5 The comparison of between VDVP-EMEA and GrEA

表1 六种算法在 WFG1-WFG9 上的 IGD 均值和标准差

Table 1 The means and standard deviations of IGD for the six algorithms on WFG1-WFG9

问题	目标 数目	超目标优化算法					
		VDVP-EMEA	VMEF	NSGA-III	SPEA2	BiGE	GrEA
WFG1	5	<b>4.09E-01 (1.45E-02)</b>	4.20E-01 (1.77E-02)	5.46E-01 (5.11E-02)	4.77E-01 (3.41E-02)	5.18E-01 (2.20E-02)	5.50E-01 (4.40E-02)
	8	<b>7.78E-01 (3.62E-02)</b>	9.42E-01 (4.32E-02)	9.68E-01 (8.90E-02)	1.37E+00 (1.29E-01)	1.28E+00 (1.73E-01)	8.58E-01 (1.50E-01)
	10	1.10E+00 (8.53E-02)	<b>1.09E+00 (4.80E-02)</b>	1.41E+00 (1.00E-01)	1.70E+00 (1.43E-01)	1.62E+00 (2.02E-01)	1.45E+00 (2.13E-01)
	15	<b>1.58E+00 (5.71E-02)</b>	2.21E+00 (2.12E-01)	1.90E+00 (2.96E-01)	2.55E+00 (1.14E-01)	2.28E+00 (2.49E-01)	2.93E+00 (4.63E-01)
	20	<b>3.04E+00 (8.30E-02)</b>	3.70E+00 (1.79E-01)	4.66E+00 (5.86E-01)	4.73E+00 (2.79E-01)	4.31E+00 (3.96E-01)	5.87E+00 (5.57E-01)
WFG2	5	1.63E+00 (6.17E-01)	<b>5.05E-01 (6.54E-02)</b>	6.98E-01 (3.00E-02)	7.13E-01 (2.02E-02)	6.35E-01 (8.88E-02)	9.78E-01 (1.51E-01)
	8	1.50E+00 (1.26E+00)	<b>1.13E+00 (1.46E-01)</b>	2.16E+00 (1.65E+00)	1.47E+00 (2.39E-01)	1.88E+00 (2.16E-01)	2.58E+00 (5.70E-01)
	10	1.77E+00 (5.06E-01)	<b>1.65E+00 (3.44E-01)</b>	4.26E+00 (2.15E+00)	2.26E+00 (5.31E-01)	2.18E+00 (3.45E-01)	2.81E+00 (2.18E-01)
	15	1.18E+00 (2.47E+00)	4.07E-01 (9.50E-01)	8.84E+00 (3.78E+00)	<b>2.80E-01 (4.29E-01)</b>	1.07E+01 (8.19E-01)	6.28E+00 (9.61E-01)
	20	2.19E+00 (3.52E+00)	1.19E+00 (1.31E+00)	1.86E+01 (2.04E+00)	<b>6.20E-01 (5.29E-01)</b>	1.30E+01 (1.24E+00)	6.93E+00 (1.39E+00)
WFG3	5	<b>1.36E-01 (4.96E-02)</b>	2.30E-01 (2.47E-02)	4.43E-01 (4.74E-02)	4.48E-01 (3.93E-02)	2.22E-01 (5.38E-02)	3.50E-01 (3.70E-02)
	8	<b>4.39E-01 (6.83E-02)</b>	7.64E-01 (9.87E-02)	1.72E+00 (2.63E-01)	1.69E+00 (1.69E+00)	5.81E-01 (9.62E-02)	8.35E-01 (1.45E-01)
	10	<b>5.50E-01 (9.11E-02)</b>	9.33E-01 (1.50E-01)	1.93E+00 (2.40E-01)	1.86E+00 (1.24E-01)	5.82E-01 (1.45E-01)	1.49E+00 (1.80E-01)
	15	<b>1.36E+00 (2.46E-01)</b>	2.78E+00 (3.70E-01)	3.95E+00 (1.85E+00)	4.24E+00 (1.46E-01)	1.64E+00 (3.45E-01)	2.93E+00 (4.17E-01)
	20	<b>1.90E+00 (3.82E-01)</b>	3.32E+00 (4.11E-01)	6.90E+00 (2.32E+00)	5.73E+00 (1.37E-01)	1.92E+00 (3.92E-01)	5.80E+00 (5.63E-01)
WFG4	5	<b>9.34E-01 (2.66E-02)</b>	1.01E+00 (8.85E-03)	9.66E-01 (1.10E-03)	9.67E-01 (6.20E-04)	1.11E+00 (2.07E-02)	9.65E-01 (7.55E-03)
	8	<b>2.96E+00 (3.81E-03)</b>	3.15E+00 (4.01E-02)	2.99E+00 (1.14E-01)	2.96E+00 (4.12E-03)	3.35E+00 (5.66E-02)	2.89E+00 (1.78E-02)
	10	4.46E+00 (1.14E-01)	4.42E+00 (3.50E-02)	4.53E+00 (1.57E-02)	4.55E+00 (1.04E-02)	4.49E+00 (7.67E-02)	<b>4.13E+00 (4.72E-02)</b>
	15	<b>9.31E+00 (1.16E-01)</b>	9.45E+00 (9.08E-02)	9.38E+00 (7.48E-02)	9.38E+00 (2.16E-02)	9.43E+00 (2.23E-01)	9.82E+00 (4.78E-01)
	20	<b>1.15E+01 (8.89E-02)</b>	1.16E+01 (9.14E-02)	1.27E+01 (7.91E-01)	1.15E+01 (9.77E-03)	1.18E+01 (2.67E-01)	1.30E+01 (4.38E-01)

续表 1 Table 1 continued

问题	目标 数目	超目标优化算法					
		VDVP-EMEA	VMEF	NSGA-III	SPEA2	BiGE	GrEA
WFG5	5	<b>9.19E-01(6.71E-02)</b>	1.00E+00(1.03E-02)	9.59E-01(7.00E-04)	9.58E-01(4.99E-04)	1.15E+00(2.09E-02)	9.61E-01(7.49E-03)
	8	2.89E+00(3.87E-02)	3.16E+00(3.37E-02)	2.94E+00(3.38E-03)	2.94E+00(2.08E-03)	3.43E+00(6.36E-02)	<b>2.89E+00(1.58E-02)</b>
	10	4.30E+00(5.11E-02)	4.45E+00(3.56E-02)	4.51E+00(1.07E-02)	4.53E+00(7.27E-03)	4.61E+00(6.12E-02)	<b>4.08E+00(3.88E-02)</b>
	15	<b>9.03E+00(4.39E-02)</b>	9.47E+00(8.45E-02)	9.28E+00(1.48E-02)	9.28E+00(1.21E-02)	9.60E+00(2.31E-01)	1.00E+01(2.93E-01)
	20	<b>1.14E+01(2.23E-01)</b>	1.16E+01(7.39E-02)	1.18E+01(1.04E+00)	1.15E+01(6.41E-03)	1.19E+01(1.30E-01)	1.25E+01(2.75E-01)
WFG6	5	1.00E+00(9.15E-02)	1.02E+00(9.40E-03)	9.65E-01(2.44E-03)	<b>9.62E-01(2.17E-03)</b>	1.10E+00(2.31E-02)	9.73E-01(7.72E-03)
	8	<b>2.92E+00(6.17E-02)</b>	3.24E+00(3.71E-02)	2.96E+00(5.74E-03)	2.97E+00(9.02E-03)	3.30E+00(5.45E-02)	2.93E+00(2.16E-02)
	10	4.38E+00(1.41E-01)	4.51E+00(3.86E-02)	4.57E+00(1.30E-02)	4.58E+00(1.21E-02)	4.44E+00(5.82E-02)	<b>4.06E+00(2.89E-02)</b>
	15	9.23E+00(1.43E-01)	9.44E+00(5.66E-02)	9.53E+00(5.32E-01)	9.36E+00(4.88E-02)	9.48E+00(4.44E-01)	<b>9.00E+00(4.18E-01)</b>
	20	<b>1.14E+01(1.53E-01)</b>	1.17E+01(4.34E-02)	1.38E+01(9.48E-01)	1.16E+01(2.24E-02)	1.20E+01(3.86E-01)	1.16E+01(5.11E-01)
WFG7	5	<b>9.34E-01(2.75E-02)</b>	1.02E+00(9.57E-03)	9.67E-01(6.84E-04)	9.68E-01(7.46E-04)	1.12E+00(2.41E-02)	9.74E-01(7.38E-03)
	8	2.97E+00(5.21E-03)	3.17E+00(4.46E-02)	2.97E+00(7.06E-03)	2.98E+00(6.64E-03)	3.33E+00(5.26E-02)	<b>2.91E+00(1.86E-02)</b>
	10	4.43E+00(1.30E-01)	4.46E+00(4.49E-02)	4.61E+00(1.99E-01)	4.55E+00(3.82E-02)	4.59E+00(7.17E-02)	<b>4.11E+00(2.89E-02)</b>
	15	9.27E+00(1.63E-01)	9.43E+00(9.32E-02)	9.34E+00(1.34E-01)	9.40E+00(3.01E-02)	9.22E+00(1.38E-01)	<b>9.01E+00(3.73E-01)</b>
	20	<b>1.15E+01(8.48E-02)</b>	1.17E+01(9.58E-02)	1.48E+01(8.42E-01)	1.16E+01(3.03E-02)	1.16E+01(1.80E-01)	1.20E+01(2.54E-01)
WFG8	5	1.03E+00(1.30E-01)	1.05E+00(7.12E-03)	9.95E-01(9.35E-03)	<b>9.86E-01(1.65E-03)</b>	1.14E+00(2.27E-02)	1.01E+00(9.03E-03)
	8	<b>2.87E+00(1.78E-02)</b>	3.26E+00(2.32E-02)	3.27E+00(1.90E-01)	3.08E+00(2.56E-02)	3.33E+00(3.73E-02)	3.05E+00(5.31E-02)
	10	<b>4.02E+00(1.17E-01)</b>	4.51E+00(3.69E-02)	4.57E+00(2.65E-01)	4.66E+00(2.73E-02)	4.41E+00(7.13E-02)	5.12E+00(2.81E-01)
	15	<b>8.13E+00(2.99E-01)</b>	9.26E+00(9.97E-02)	9.07E+00(2.97E-01)	9.37E+00(2.91E-02)	1.01E+01(2.65E-01)	1.05E+01(9.99E-02)
	20	<b>1.10E+01(4.88E-01)</b>	1.20E+01(3.65E-01)	1.36E+01(8.08E-01)	1.17E+01(8.18E-02)	1.36E+01(1.88E-01)	1.50E+01(1.13E-01)
WFG9	5	<b>8.15E-01(8.03E-03)</b>	9.78E-01(1.05E-02)	9.35E-01(3.66E-03)	9.43E-01(2.63E-03)	1.15E+00(3.08E-02)	9.34E-01(5.05E-03)
	8	<b>2.86E+00(3.69E-02)</b>	3.11E+00(2.88E-02)	2.94E+00(2.09E-02)	2.94E+00(7.69E-03)	3.43E+00(6.48E-02)	2.91E+00(1.37E-02)
	10	4.31E+00(3.24E-02)	4.33E+00(3.62E-02)	4.37E+00(6.25E-02)	4.51E+00(1.73E-02)	4.60E+00(6.38E-02)	<b>4.14E+00(3.56E-02)</b>
	15	<b>8.64E+00(8.64E+00)</b>	9.00E+00(8.98E-02)	8.72E+00(9.36E-02)	9.15E+00(2.60E-02)	9.49E+00(2.22E-01)	9.40E+00(3.10E-01)
	20	1.17E+01(1.93E-01)	<b>1.16E+01(1.07E-01)</b>	1.35E+01(5.06E-01)	1.19E+01(7.16E-02)	1.23E+01(3.81E-01)	1.23E+01(2.31E-01)
‘-’/‘+’/‘≈’			34/5/6	35/1/9	37/3/5	35/2/8	29/9/7

4 结论

本文研究在超目标进化中,将多种解排序算法集成到一个框架中,并将进化算法的环境选择看做一个投票过程,利用专家的投票结果和其价值点计算个体的适应度值,采用精英策略选择下一代种群. 通过计算所有解排序算法的累计有效投票率动态调整其价值点,价值点越高说明对应的解排序算法越适合解决该超目标优化问题. 为了验证所提出的 VDVP-EMEA 算法的性能,在 45 个测试问题上,将本文算法与 VMEF、NSGA-III、SPEA2、BiGE、GrEA 5 种算法进行比较,数值实验表明本算法比其他五种算法性能更好,能够更有效地引导种群靠近真实的 PF.

由于 VDVP-EMEA 的性能在很大程度上受解排序算法的组合方式的影响,因此如何选择集成的解排序算法,并将其应用于解决现实世界中的超目标优化问题,是我们进一步的研究方向.

[ 参考文献 ]

[ 1 ] LIN Q Z, CHEN J Y, ZHAN Z H, et al. A hybrid evolutionary immune algorithm for multiobjective optimization problems[ J ]. IEEE transactions on evolutionary computation, 2016, 20( 5 ): 711-729.

[ 2 ] QI Y T, HOU Z T, YIN M L, et al. An immune multiobjective optimization algorithm with differential evolution inspired recombination[ J ]. Applied soft computing, 2015, 29: 395-410.

[ 3 ] WU G H, MALLIPEDDI R, SUGANTHAN P N, et al. Differential evolution with multi-population based ensemble of mutation strategies[ J ]. Information sciences, 2016, 329: 329-345.

[ 4 ] GONG W Y, FIALHO Á, CAI Z H, et al. Adaptive strategy selection in differential evolution for numerical optimization: An empirical study[ J ]. Information sciences, 2011, 181( 24 ): 5364-5386.

[ 5 ] WANG W J, YANG S Q, LIN Q Z, et al. An effective ensemble framework for multiobjective optimization[ J ]. IEEE transactions on evolutionary computation, 2019, 23( 4 ): 645-659.

- [ 6 ] PAL M,BANDYOPADHYAY S. ESOEA;Ensemble of single objective evolutionary algorithms for many-objective optimization[J]. Swarm and evolutionary computation,2019,50:100511.
- [ 7 ] MALLIPEDDI R,SUGANTHAN P N. Ensemble of constraint handling techniques[J]. IEEE transactions on evolutionary computation,2010,14(4):561-579.
- [ 8 ] ZHAO S Z,SUGANTHAN P N,ZHANG Q. Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes[J]. IEEE transactions on evolutionary computation,2012,16(3):442-446.
- [ 9 ] QIU W B,ZHU J H,WU G H,et al. Ensemble many-objective optimization algorithm based on voting mechanism[J]. IEEE transactions on systems,2012,52(3):1716-1730.
- [ 10 ] DEB K,JAIN H. An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach,Part I:Solving problems with box constraints[J]. IEEE transactions on evolutionary computation,2014,18(4):577-601.
- [ 11 ] ZITZLER E, LAUMANN S M, THIELE L. SPEA2: improving the strength pareto evolutionary algorithm for multiobjective optimization[C]//Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems. Proceedings of the EUROGEN'2001. Athens. Greece,2001.
- [ 12 ] YANG S X,LI M Q,LIU X H,et al. A grid-based evolutionary algorithm for many-objective optimization[J]. IEEE transactions on evolutionary computation,2013,17(5):721-736.
- [ 13 ] LI M Q,YANG S X,LIU X H. Bi-goal evolution for many-objective optimization problems[J]. Artificial intelligence,2015,228:45-65.
- [ 14 ] YE T,CHENG R,ZHANG X Y,et al. PlatEMO:A MATLAB platform for evolutionary multi-objective optimization[ Educational Forum][J]. IEEE computational intelligence magazine,2017,12(4):73-87.
- [ 15 ] HUBAND S,HINGSTON P,BARONE L,et al. A review of multiobjective test problems and a scalable test problem toolkit[J]. IEEE transactions on evolutionary computation,2006,10(5):477-506.
- [ 16 ] DAS J,DENNIS J E. Normal-boundary intersection:A new method for generating the Pareto surface in nonlinear multicriteria optimization problems[J]. SIAM journal on optimization,1998,8(3):631-657.
- [ 17 ] CZYZZAK P,JASZKIEWICZ A. Pareto simulated annealing: a metaheuristic technique for multi-objective combinatorial optimization[J]. Journal of multi-criteria decision analysis,1998,7(1):34-47.

[ 责任编辑:陆炳新 ]