

控制与决策

Control and Decision

基于双重贡献分配的多目标混合算子进化算法

耿焕同, 许可, 戴中斌, 徐小涵

引用本文:

耿焕同, 许可, 戴中斌, 徐小涵. 基于双重贡献分配的多目标混合算子进化算法[J]. *控制与决策*, 2022, 37(5): 1195–1202.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2020.1284>

您可能感兴趣的其他文章

Articles you may be interested in

基于向量角分解的高维多目标进化算法

Many-objective evolutionary algorithm based on vector angle decomposition

控制与决策. 2021, 36(3): 761–768 <https://doi.org/10.13195/j.kzyjc.2019.0925>

一种求解约束多目标问题的协作进化算法

A collaborative evolutionary algorithm for solving constrained multi-objective problems

控制与决策. 2021, 36(11): 2656–2664 <https://doi.org/10.13195/j.kzyjc.2020.0791>

基于复杂昂贵仿真的体系效能多目标优化

Complex and expensive simulation based multi-objective optimization to system-of-system effectiveness

控制与决策. 2021, 36(3): 589–598 <https://doi.org/10.13195/j.kzyjc.2019.0844>

一种自适应拟牛顿-状态转移混合智能优化算法及应用

A hybrid state transition optimization algorithm based on adaptive quasi-newton method and its application

控制与决策. 2021, 36(10): 2451–2458 <https://doi.org/10.13195/j.kzyjc.2020.0214>

基于改进多目标优化算法的分布式数据中心负载调度

Multi-objective optimization of energy and performance management in distributed data centers

控制与决策. 2021, 36(1): 159–165 <https://doi.org/10.13195/j.kzyjc.2019.0702>

基于双重贡献分配的多目标混合算子进化算法

耿焕同^{1,2†}, 许可¹, 戴中斌¹, 徐小涵¹

(1. 南京信息工程大学 计算机与软件学院, 南京 210044;

2. 江苏省气象局 南京大气科学联合研究中心, 南京 210009)

摘要: 针对多目标混合算子进化算法中各算子有效选择的自适应问题, 提出一种基于双重贡献分配的多目标混合算子进化算法(DCA-MOEA/D). 首先, 将两种现有的进化算子与两种基于方向引导的差分进化组成算子池, 每代个体以轮盘赌的方式从中选择一种进化算子产生子代; 然后, 根据子代的表现, 结合两种方法为各算子分配贡献值, 从而确定算子的选择概率; 接着, 引入外部归档集, 根据非支配关系与拥挤度策略对其进行维护; 最后, 将整个进化过程划分为5个阶段, 以达到算子选择中“探索”与“探究”之间的平衡. 以IGD与HV为性能评价指标, 通过与其他4种多目标进化算法在23个测试函数上的对比, 验证所提出算法在收敛性和分布性上的显著优势.

关键词: 多目标优化; 混合算子; 贡献分配; 算子选择

中图分类号: TP273

文献标志码: A

DOI: 10.13195/j.kzyjc.2020.1284

开放科学(资源服务)标识码(OSID):



引用格式: 耿焕同, 许可, 戴中斌, 等. 基于双重贡献分配的多目标混合算子进化算法[J]. 控制与决策, 2022, 37(5): 1195-1202.

Multi-objective evolutionary algorithm with multiple operators based on double credit assignment

GENG Huan-tong^{1,2†}, XU Ke¹, DAI Zhong-bin¹, XU Xiao-han¹

(1. School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China; 2. Nanjing Joint Center of Atmospheric Research, Meteorological Bureau of Jiangsu Province, Nanjing 210009, China.)

Abstract: In order to make the operator selection more efficient in multi-objective evolutionary algorithms (MOEAs) with multiple operators, this paper proposes a MOEA based on the decomposition with double credit assignment (DCA-MOEA/D). Firstly, the operator pool in proposed algorithm consists of two existing operators and two variants of differential evolution (DE) based on the direction-guided search strategy. Individuals use a roulette wheel-like process to pick up an operator to generate offspring at each generation. Subsequently, the credit value of each operator is determined by combining two credit assignment methods according to the performance of the offspring, and the selection probability of each operator is updated by the credits. Meanwhile, an extra archive is defined and the dominated sorting and crowded distance strategies are used to maintain it. Finally, the whole evolutionary process is divided into 5 steps to achieve the balance between exploration and exploitation in operator selection. Empirical study validates the effectiveness of the proposed algorithm through the contrast experiment with four MOEAs in terms of IGD and HV value on 23 benchmark problems.

Keywords: multi-objective optimization; multiple operators; credit assignment; operator selection

0 引言

不失一般性, 以最小化为例, 多目标优化问题(MOP)定义如下:

$$\begin{cases} \text{minimize } F(x) = (f_1(x), f_2(x), \dots, f_m(x))^T, \\ \text{subject to } x \in \Omega. \end{cases} \quad (1)$$

其中: $x = (x_1, x_2, \dots, x_n)$ 被称为 n 维决策向量; Ω 为决策空间中的可行域; R^m 被称为 m 维目标空间, $F: \Omega \rightarrow R^m$ 定义了从决策空间到目标空间的映射关系. 由于式(1)总是相互冲突的, 不存在某个点可以同时最小化所有目标函数, 只能找到一组折衷方案供决策者选择. 这种折衷最优解可以用 Pareto 最优定义. 在决策空间中, 所有 Pareto 最优解组成的集合被称作

收稿日期: 2020-09-14; 录用日期: 2021-03-16.

基金项目: 国家自然科学基金项目(51977100); 国家重点研发计划项目(2017YFC1502104).

†通讯作者. E-mail: genghuantong@163.com.

Pareto集(PS),与之对应的在目标空间中的解集被称为Pareto前沿(PF).

与传统方法相比,群智能优化算法对于解决此类问题具有明显优势,这是因为它们仅需一轮运行便能搜索到一组最优解.其中,多目标进化算法(MOEA)更是得到了学者的广泛关注和研究^[1-3].在过去的10年里,研究者们意识到使用单一的进化算子不能很好地处理不同类型的多目标优化问题^[4].自此,基于混合算子的多目标进化算法相继被提出^[5-6],这类算法的核心思想是将多种进化算子相结合以弥补各自的不足,从而提升算法的整体表现.如何通过多种算子的有机结合提高算法的优化效果及泛化性已成为该领域的研究热点.到目前为止,主流方法大致可分为以下两种:自适应进化算子选择(AOS)和多种进化算子组装(MOE).前者在每个时间点挑选出一个最合适的算子来进化种群,而后者使用多种算子同时参与到种群的每代进化中.

贡献分配是一种在混合算子进化算法中常用的技术,旨在根据各算子的表现对它们进行奖励或者惩罚.在基于分解的多目标进化算法中,通过计算个体的适应度提升值可以有效量化算子的表现,因此这类算法常以MOEA/D^[7]作为基本框架.而算子表现难以通过支配关系进行量化,因此基于支配关系的混合算子进化算法鲜有提出.但在实际问题中,并非所有类型的优化问题均适合采用MOEA/D解决,尤其是在求解复杂的优化问题时可能无法得到令人满意的结果.鉴于此,本文提出一种基于双重贡献分配的混合算子进化算法(DCA-MOEA/D).算法以MOEA/D为基本框架,同时引入外部归档集并使用基于支配关系与拥挤度的选择策略对其进行维护,一定程度上弥补了MOEA/D在解决部分优化问题时的不足;不同于传统混合算子进化算法中采用单一的贡献分配技术,由于DCA-MOEA/D拥有两个种群且分别使用不同的环境选择策略,相对应地同时采用基于个体提升率和基于个体生存率的两种贡献分配技术为各算子确定贡献值.通过与其他各类算法在标准测试集函数上的对比实验,对所提算法的有效性进行验证.

1 相关研究及存在的问题

1.1 基于混合算子的进化算法

不同进化算子具有不同的特性,因此它们擅于解决不同类型的优化问题或是适用于不同的进化阶段.将多种互补的算子混合以提高解决各类优化问题的能力成为这类算法的核心思想.通常来讲,它们可以被分为以下两类:

第1类是自适应进化算子选择(AOS),这类方法

包含两个关键技术:贡献分配以及算子选择.例如,文献[8]认为算子选择可以被看作是多臂老虎机问题(MAB),由此利用解决MAB问题的算法对算子进行自适应选择;文献[9]提出了一种基于模糊逻辑引擎的智能算子选择器,用于根据进化的不同阶段对算子进行自主选择.

第2类是多种进化算子组装(MOE),其主要特点是同时使用多种算子进化种群.例如,文献[10]提出了一种基于混合算子的免疫进化算法,它将种群划分为两个子种群并分别使用模拟二进制交叉(SBX)和差分进化(DE)算子进行进化;文献[11]在划分子种群的基础上使用一种基于Adaboost的贡献分配技术调整各子种群的大小,为算子动态地分配计算资源.

1.2 存在的问题

尽管在过去10年里已有许多学者对多目标混合算子进化算法进行了大量的研究,但其中多数采用的是基于AOS的结合方法.而针对MOE的研究起步较晚,到目前为止,国内外的相关成果仍然有限,主要困难在于难以对各算子的表现同时进行准确评估.此外,现存的混合算子进化算法简单地把几种相互孤立的进化算子组成算子池,缺乏彼此间的配合.为方便对算子的表现进行评估,多数算法选用MOEA/D作为基本框架,而只有邻域内子问题间具有较高相似度时,MOEA/D才能取得较好的求解效果.因此,这类算法仍难以摆脱求解问题类型单一的局限.

1.3 经典进化算子

进化算子作为多目标进化算法中的搜索引擎,对算法的表现起到了决定性作用.其中,差分进化(DE)^[12]与模拟二进制交叉(SBX)^[13]应用最为广泛.DE最早由Storn和Price于1997年提出,其核心思想是利用种群个体间的差异指导进化,最常使用的DE/rand/1变异策略如下:

$$v_i = x_i + F(x_{r_1} - x_{r_2}). \quad (2)$$

其中: x_i 为目标个体, x_{r_1} 和 x_{r_2} 为从种群中随机选取的两个个体.DE具有强大的全局搜索能力,这一特性使得它被快速拓展到解决多目标优化问题中^[14-15],并取得了较好的效果.

SBX是一种基于实数编码的重组算子,它使用两个父代个体对其染色体上的基因进行逐项混合产生子代.假设有两个个体 x_1 和 x_2 ,则它们的子代 c_1 和 c_2 可以用下式生成:

$$\begin{cases} c_1 = 0.5 \times [(1 + \beta) \cdot x_1 + (1 - \beta) \cdot x_2], \\ c_2 = 0.5 \times [(1 - \beta) \cdot x_1 + (1 + \beta) \cdot x_2]. \end{cases} \quad (3)$$

其中: β 由分布因子 η 动态且随机地决定,即

$$\beta = \begin{cases} (\text{rand} \times 2)^{1/(1+\eta)}, & \text{rand} \leq 0.5; \\ \left[\frac{1}{(2 - \text{rand} \times 2)} \right]^{1/(1+\eta)}, & \text{otherwise.} \end{cases} \quad (4)$$

这里 η 是一个自定义参数,取值越高代表着产生的子代越接近其父代。SBX具有较强的局部搜索能力,被广泛应用于多目标进化算法中。

1.4 算子选择策略

在基于混合算子的多目标进化算法中,通常根据各算子的表现对它们进行自适应选择。现介绍一种基于概率的算子选择策略:概率匹配(PM)。假设现有一个 $S > 1$ 的算子池 $\zeta = \{op_1, op_2, \dots, op_S\}$,各算子在 g 代的选择概率记作 $p(g) = \{p_1(g), p_2(g), \dots, p_S(g)\}$,定义经验质量 $q_s(g)$ 为

$$q_s(g+1) = q_s(g) + \alpha[r_s(g) - q_s(g)], \quad (5)$$

它反映了算子近期的表现。其中: α 为自适应率,取值范围在 $[0, 1]$ 之间, $r_s(g)$ 为算子 s 在 g 代分配到的贡献值。在PM策略中,各算子的选择概率由下式确定:

$$p_s(g+1) = \frac{p_{\min} + (1 - S \cdot p_{\min}) \cdot \frac{q_s(g+1)}{\sum_{s=1}^S q_s(g+1)}}{S}, \quad (6)$$

其中 p_{\min} 为算子被分配的最小概率。

2 基于双重贡献分配的进化算法

2.1 算法总览

本文提出一种基于双重贡献分配的多目标混合算子进化算法,算法以MOEA/D为基本框架,其主要特征有:1)定义一个外部归档集来存储当前最优解并根据支配关系与拥挤度对其进行维护;2)将两种基于方向引导技术的DE变种加入到算子池中,加强各算子间的联系;3)采用双重贡献分配技术为各算子确定贡献值;4)将整个进化过程划分为若干阶段,各阶段内概率分配互不影响。算法的整体流程如下。

step 1: 初始化种群 P ,定义外部归档集 $A = P$ 。设置各算子最小使用概率 p_{\min} 、种群规模 N 、自适应率 α 。

step 2: 如果当前处于某个进化阶段的初代,则为所有算子分配相同的经验质量和选择概率;否则使用PM方法更新经验质量并分配选择概率。

step 3: 种群中每个个体按照概率从算子池中随机选择一种算子进行进化,替换邻域内子问题关联的较差解,并计算适应度提升率。

step 4: 合并外部归档集 $A = P \cup A$,采用NSGA-II的环境选择策略从中选择 N 个个体重新组成 A 。

step 5: 计算由各算子进化出的个体的生存率,结

合个体的总适应度提升率为算子分配贡献值。

step 6: 如果满足终止条件,则输出归档集 A ;否则转至step 2。

2.2 混合进化算子

DCA-MOEA/D采用以下4种进化算子组成算子池:

1) DE/rand/1:

$$v_i = x_i + F(x_{r_1} - x_{r_2}).$$

2) SBX:

$$v_i = 0.5 \times [(1 + \beta) \cdot x_i^1 + (1 - \beta) \cdot x_i^2].$$

3) 基于邻居关系的局部探究增强型差分进化(NEEDE):

$$v_i = x_i + F \cdot (x_{rn_1} - x_{rn_1}^p) + F \cdot (x_{rn_2} - x_{rn_3}). \quad (7)$$

其中: x_{rn_1} 、 x_{rn_2} 和 x_{rn_3} 是从 x_i 邻域内随机挑选出的3个个体, $x_{rn_1}^p$ 为 x_{rn_1} 的父代。等式右边第1项为方向引导项,它通过挖掘邻域内个体的移动轨迹指引目标个体的进化;等式右边第2项为随机扰动项,旨在为进化添加一定的随机性。

4) 基于种群的全局探索增强型差分进化(PEEDE):

$$v_i = x_i + F \cdot (x_{rn_1} - x_{rn_1}^p) + F \cdot (x_{rp_1} - x_{rp_2}), \quad (8)$$

其中 x_{rp_1} 和 x_{rp_2} 是从整个种群中随机挑选出的两个个体。

NEEDE和PEEDE算子是本文基于方向引导技术^[16]提出的两种DE变种。传统的差分进化忽略了个体在进化过程中的信息,从而限制了搜索效率。在进化算法中,生存压力会迫使决策空间中的个体不断向PS移动,因此通过挖掘和分析个体的移动轨迹可以指导它们向着更有希望的方向进化。根据这一思路本文提出NEEDE。

NEEDE这种方向引导的搜索策略具有极强的局部探究能力,但是容易陷入到局部最优中,因为该策略的随机扰动取决于邻域内个体间的差异,而这种差异通常较小,尤其是在进化的中后期,这一问题将会更加严重。因此,本文提出PEEDE。

NEEDE与PEEDE的差异仅在随机扰动项中个体选择范围的不同。PEEDE由于从整个种群中随机挑选了两个个体,两者间的差异一般较大,从而扩大了该项对新产生子代的影响,因此算子具备较强的全局探索能力。

仔细观察式(7)和(8)中的方向引导项,不难发现它们的值取决于 x_{rn_1} 的产生方式,因此使用NEEDE和PEEDE产生的子代会受到其他进化算子的影响,

这样便加强了各算子间的联系. 在DCA-MOEA/D中, 种群中个体以一定概率从算子池中随机选择一种进化算子, 省去了划分和调整子种群大小带来的额外计算开销, 同时将多算子混合带来的优势发挥到极致.

2.3 算子选择概率自适应

大部分多目标混合算子进化算法会通过引入算子自适应选择机制提高算法表现, 各算子对种群进化的贡献值是算子选择的主要参考依据. 由于DCA-MOEA/D包含主种群和外部归档集两个种群, 相应地采用两种贡献分配方法对各算子的表现进行评价, 具体如下.

2.3.1 基于个体适应度提升率的贡献分配

在MOEA/D中, 算子通过子问题邻域内关联的个体产生子代, 并替换若干适应度较差的个体. 由于在进化初期时个体的适应度提升较为显著, 而在后期时趋于平稳, 采用个体适应度的提升率(FIR)衡量算法表现, 计算方法如下:

$$\text{FIR}_i = \frac{f_{i,\text{old}} - f_{i,\text{new}}}{f_{i,\text{old}}}, \quad (9)$$

其中 $f_{i,\text{old}}$ 和 $f_{i,\text{new}}$ 分别为个体 i 在被替换前和被替换后的适应度标量. 当某个新个体成功替换了其他个体时, 产生它的算子可以获得如下奖励值:

$$\eta_s = \frac{g(x^i|\lambda^i, z^*) - g(y|\lambda^i, z^*)}{g(x^i|\lambda^i, z^*)}. \quad (10)$$

其中: λ^i 为第 i 个子问题对应的权重向量, x^i 和 y 分别为被替换的个体和新个体, z^* 为理想点. 在MOEA/D中, 每次产生的子代可能会替换邻域内的多个其他个体, 假设算子在个体 i 上获得的奖励值分别为 $\eta = \{\eta_1, \eta_2, \dots, \eta_r\}$, 则算子可从该个体上获得的总奖励值为

$$\eta_i^s = \bar{\eta} = \frac{\sum_{i=1}^{|\eta|} \eta_i}{|\eta|}. \quad (11)$$

其中: $|\eta|$ 为子代可替换邻域内个体的最大数量, 如果 $|\eta| = 0$, 则 $\eta_i^s = 0$. 值得注意的是, $|\eta|$ 并不一定等于子代实际替换的个体数量, 因为在MOEA/D中通常会通过设置最大个体替换数来避免种群多样性的损失. 在一代进化完成后, 算子 s 可获得的总贡献值为

$$c_s^1 = \sum_{i=1}^{n_s} \eta_i^s. \quad (12)$$

其中: c_s^1 为算子 s 基于个体适应度提升率所分配到的贡献值, n_s 为使用算子 s 进化出的子代数量.

2.3.2 基于子代生存率的贡献分配

DCA-MOEA/D使用了一个外部归档集存储当前最优解, 本文根据各算子进化出的成功进入到外

部归档集的个体占有所有新生个体的比例(称之为生存率)来衡量算子对外部归档集所做的贡献, 即

$$c_s^2 = \frac{n_{\text{in}}}{N}. \quad (13)$$

其中: n_{in} 为算子 s 进化出的成功进入到外部归档集的个体数, N 为种群大小. 最后, 取式(12)和(13)两者中较大的值作为算子在本代的最终贡献, 即

$$c_s = \max(c_s^1, c_s^2). \quad (14)$$

接着, 采用PM策略为每个算子分配选择概率, 具体方法如式(5)、(6)所示. 在每代进化中, 个体按照概率以轮盘赌的方式随机挑选一种算子进行进化, 表现越好的算子会被越多个体选择.

2.4 划分进化阶段

若算子在进化初期表现较差, 则根据前文的贡献分配策略, 其被分配到的选择概率会愈发变小, 即使后期表现有所回升也难以调整, 但若过度关注算子在未来可能有的表现, 则无法大量采用当前最优算子, 这被称为算子选择中“探索”与“探究”之间的平衡问题. 为解决这一问题, 本文将整个进化过程均匀地划分为若干阶段并在各阶段的初代重置所有算子的选择概率. 考虑到在短时间内算子的表现通常不会发生激变, 且各算子的最小选择概率也为“探索”提供了一定保障, 因此划分的阶段数量不宜过多, 否则将会影响算子评估的连贯性. 一般地, 可将整个进化过程划分为进化前期、前中期、中期、中后期和后期5个阶段进行分析, 因此本文将划分的阶段数量设置为5.

2.5 算法时间复杂度分析

相较于传统MOEA/D, DCA-MOEA/D在维护外部归档集以及通过双重贡献分配确定各算子选择概率时有额外计算开销. 以一轮进化为例, 在这一过程中维护外部归档集的时间开销为 $O(MN^2)$. 其中: M 为目标个数, N 为种群数量. 基于个体适应度提升率的贡献分配与原MOEA/D的邻域个体替换操作同时进行, 无额外时间开销. 基于个体生存率的贡献分配及算子选择概率的计算时间复杂度均为 $O(1)$. 综上所述, 结合传统MOEA/D本身的时间开销 $O(MNT)$, DCA-MOEA/D每代循环的时间复杂度为 $O(MN^2)$.

3 实验及结果分析

本节将DCA-MOEA/D与其他具有代表性的多目标进化算法进行对比, 并对结果进行简要分析.

3.1 实验环境及测试问题

为了验证本文提出的DCA-MOEA/D的有效性, 分别将本文算法与IMMOEA^[17]、EAG-MOEA/D^[18]、

MOEA/D-FRRMAB^[8]以及MOEA/D-M2M^[19]进行对比。IMMOEA是一种采用高斯过程逆建模算法的基于支配关系的进化算法;MOEA/D-FRRMAB是基于置信概率上界(UCB)算法的一种AOS方式的混合算子进化算法;EAG-MOEA/D将基于分解与基于支配关系的两种选择策略相结合以提升算法的泛化性;MOEA/D-M2M通过分解决策空间将一个复杂的多目标优化问题转化为一系列简单的多目标优化问题进行求解。DCA-MOEA/D算法的源代码可在<https://github.com/xuke-paper/DCA-MOEA-D>查阅。

本文选取UF 1~UF 7、WFG 1~WFG 9、DTLZ 1~DTZ 7等23个经典测试函数。其中:WFG测试集设置优化目标个数 $M = 2$,决策维度 $D = 11$;DTLZ测试集设置优化目标个数 $M = 3$ 。上述测试函数涵盖了多种类型的优化问题。实验运行于PlatEMO^[20],它是一个基于Matlab开发的开源多目标进化算法测试平台。

3.2 评价指标

本文选用反世代距离(IGD)和超体积(HV)作为算法性能评价指标,两者均可同时反映算法的收敛性和分布性。

1)反世代距离(IGD),表示真实PF面上各点到算法所得PF的最短距离的平均值,计算公式如下:

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|} \quad (15)$$

其中: P^* 由沿着真实PF均匀分布的采样点构成, P 为算法找到的最终解。IGD值衡量了从 P^* 上各点到 P

的最短平均距离,值越小表明算法的表现越好。

2)超体积(HV),表示在目标空间中算法获得的解与参照点围成区域的超体积,计算公式如下:

$$HV = \text{volume}(\bigcup_{y \in S} [y_1, y_1^*] \times \dots \times [y_m, y_m^*]) \quad (16)$$

其中: S 为在目标空间中算法得到的解集; $y^* = (y_1^*, y_2^*, \dots, y_n^*)$ 为选择的参考点,它被 S 中所有解支配。HV指标越大表明算法的收敛性和分布性越好。

3.3 实验参数设置

为保证实验的公平性,所有多目标进化算法的进化算子被设置为完全相同的参数,且均采用多项式突变。具体参数设置如下:1)对于差分进化, $C_r = 1.0, F = 0.5$;2)对于模拟二进制交叉, $C_r = 1.0, \eta = 20$;3)对于多项式突变, $\eta = 20, P_m = 1/n$ 。

MOEA/D框架的通用参数设置如下: $T = 10\% \times N, n_r = 2, \delta = 0.9$ 。分解方法采用Tchebycheff聚合法,其他控制参数设置为各算法文献中的推荐值。DCA-MOEA/D中其余参数参考文献[21]设置如下:1)算子最小选择概率 $p_{\min} = 0.05$;2)经验质量更新的自适应率 $\alpha = 0.3$ 。

算法停止条件为目标函数的总评估次数。对于双目标优化问题,种群规模设置为100,评估次数为30 000;对于三目标优化问题,种群规模设置为300,评估次数为90 000。

3.4 实验结果及分析

5种算法在各测试函数上的对比结果如表1和表2所示,DCA-MOEA/D的具体求解效果如图1所示。

表1 5种算法在3类测试集上IGD的平均值和标准差

测试问题	算 法				
	DCA-MOEA/D	IMMOEA	EAG-MOEA/D	MOEA/D-FRRMAB	MOEA/D-M2M
WFG 1	1.455 0e-1 (3.37e-2)	1.147 1e+0 (6.99e-2) -	7.477 1e-1 (1.80e-1) -	1.047 3e+0 (6.29e-2) -	8.912 1e-1 (7.86e-2) -
WFG 2	1.279 0e-2 (3.82e-4)	2.085 9e-2 (1.45e-3) -	3.457 6e-2 (3.27e-2) -	2.380 6e-2 (1.30e-3) -	2.444 1e-2 (2.51e-3) -
WFG 3	1.523 6e-2 (6.59e-4)	2.294 5e-2 (8.71e-4) -	1.776 7e-1 (3.23e-2) -	1.769 8e-2 (1.63e-3) -	2.802 7e-2 (1.88e-3) -
WFG 4	1.675 2e-2 (9.25e-4)	3.421 6e-2 (2.97e-3) -	1.637 1e-2 (1.27e-3) ≈	6.004 2e-2 (9.44e-3) -	3.225 7e-2 (3.70e-3) -
WFG 5	6.774 8e-2 (1.46e-3)	7.164 2e-2 (2.38e-3) -	6.547 7e-2 (7.04e-4) +	6.904 1e-2 (1.61e-3) -	6.848 1e-2 (7.86e-4) ≈
WFG 6	6.154 8e-2 (7.56e-2)	3.075 4e-2 (3.67e-3) ≈	2.350 4e-1 (3.62e-2) -	1.794 5e-1 (8.57e-2) -	1.499 9e-1 (9.23e-2) -
WFG 7	1.655 6e-2 (9.78e-4)	2.776 2e-2 (1.30e-3) -	1.485 4e-1 (3.22e-2) -	1.508 6e-2 (5.64e-4) +	2.484 5e-2 (1.16e-3) -
WFG 8	1.050 2e-1 (4.11e-3)	1.172 0e-1 (2.27e-3) -	3.245 4e-1 (3.86e-2) -	1.039 0e-1 (4.94e-3) ≈	1.194 1e-1 (4.10e-3) -
WFG 9	4.531 7e-2 (6.21e-2)	2.998 9e-2 (3.04e-3) +	7.211 0e-2 (1.74e-2) -	1.586 2e-1 (1.01e-1) -	2.873 3e-2 (5.14e-3) +
UF 1	4.967 2e-2 (3.33e-2)	7.808 6e-2 (9.32e-3) -	1.871 1e-1 (6.06e-2) -	1.338 7e-1 (1.18e-1) -	7.655 4e-2 (1.17e-2) -
UF 2	4.020 2e-2 (2.30e-2)	4.141 4e-2 (1.24e-2) ≈	9.424 5e-2 (3.13e-2) -	4.235 4e-2 (6.33e-3) -	3.796 0e-2 (3.51e-3) ≈
UF 3	1.327 4e-1 (7.07e-2)	1.431 0e-1 (1.50e-2) -	2.207 3e-1 (4.44e-2) -	1.894 1e-1 (5.11e-2) -	1.688 8e-1 (2.76e-2) -
UF 4	4.664 1e-2 (2.72e-3)	8.719 0e-2 (3.40e-3) -	1.622 5e-1 (9.54e-3) -	8.551 9e-2 (9.94e-3) -	6.693 3e-2 (4.91e-3) -
UF 5	5.166 1e-1 (1.50e-1)	1.133 8e+0 (2.09e-1) -	6.040 5e-1 (1.41e-1) -	1.244 4e+0 (2.74e-1) -	8.572 9e-1 (1.67e-1) -
UF 6	3.149 1e-1 (1.75e-1)	3.343 3e-1 (6.16e-2) ≈	3.173 0e-1 (1.67e-1) ≈	3.651 5e-1 (9.08e-2) ≈	4.472 6e-1 (3.70e-2) -
UF 7	4.100 0e-2 (9.69e-2)	1.418 0e-1 (1.06e-1) -	2.877 6e-1 (1.45e-1) -	3.102 8e-1 (2.57e-1) -	4.790 3e-2 (9.09e-3) -
DTLZ 1	1.452 8e-2 (3.19e-4)	2.830 3e+0 (6.71e-1) -	7.432 5e-2 (7.67e-3) -	1.704 4e-2 (4.08e-3) -	1.113 1e+0 (9.45e-1) -
DTLZ 2	3.753 7e-2 (7.80e-4)	4.837 8e-2 (1.67e-3) -	1.665 9e-1 (1.49e-2) -	3.8230e-2 (2.79e-4) -	8.4694e-2 (4.25e-3) -
DTLZ 3	3.705 4e-2 (8.57e-4)	5.161 6e+1 (7.17e+0) -	2.035 9e-1 (1.35e-2) -	9.431 8e-1 (1.63e+0) -	3.487 9e+1 (1.28e+1) -
DTLZ 4	3.740 0e-2 (7.55e-4)	4.247 5e-2 (1.12e-3) -	7.725 3e-2 (1.17e-1) -	6.008 5e-2 (5.35e-2) -	4.893 5e-2 (1.47e-3) -
DTLZ 5	1.883 1e-3 (5.47e-5)	6.625 0e-3 (5.64e-4) -	4.335 6e-2 (1.21e-2) -	4.706 6e-3 (7.24e-5) -	8.478 1e-3 (7.75e-4) -
DTLZ 6	1.826 2e-3 (7.71e-5)	4.128 5e+0 (1.06e-1) -	5.322 4e-2 (1.27e-2) -	4.798 0e-3 (1.22e-5) -	7.246 4e-3 (4.48e-4) -
DTLZ 7	4.270 2e-2 (1.96e-3)	1.684 7e-1 (1.86e-2) -	1.134 0e-1 (4.52e-2) -	1.294 0e-1 (6.50e-2) -	2.606 2e-1 (4.17e-2) -

表2 5种算法在3类测试集上HV的平均值和标准差

测试问题	算 法				
	DCA-MOEA/D	IMMOEA	EAG-MOEA/D	MOEA/D-FRRMAB	MOEA/D-M2M
WFG 1	6.186 3e-1 (1.72e-2)	1.706 8e-1 (1.79e-2) -	3.335 0e-1 (6.73e-2) -	2.175 8e-1 (1.58e-2) -	2.806 6e-1 (2.65e-2) -
WFG 2	6.326 0e-1 (2.43e-4)	6.267 1e-1 (8.66e-4) -	6.253 0e-1 (1.22e-2) -	6.2765e-1 (8.07e-4) -	6.2534e-1 (2.62e-3) -
WFG 3	5.795 0e-1 (3.87e-4)	5.735 2e-1 (5.41e-4) -	5.071 0e-1 (1.35e-2) -	5.768 1e-1 (1.13e-3) -	5.712 9e-1 (1.34e-3) -
WFG 4	3.454 6e-1 (3.20e-4)	3.332 1e-1 (1.34e-3) -	3.455 1e-1 (1.04e-3) ≈	3.194 4e-1 (4.17e-3) -	3.330 7e-1 (2.38e-3) -
WFG 5	3.115 4e-1 (1.31e-3)	3.076 0e-1 (1.40e-3) -	3.125 5e-1 (4.11e-4) +	3.076 5e-1 (1.44e-3) -	3.091 6e-1 (4.59e-4) -
WFG 6	3.179 8e-1 (4.22e-2)	3.334 8e-1 (2.62e-3) ≈	2.267 9e-1 (1.93e-2) -	2.516 4e-1 (4.76e-2) -	2.650 1e-1 (5.20e-2) -
WFG 7	3.459 3e-1 (2.08e-4)	3.363 3e-1 (9.86e-4) -	2.938 9e-1 (1.13e-2) -	3.442 4e-1 (4.27e-4) -	3.393 4e-1 (6.70e-4) -
WFG 8	2.911 9e-1 (2.13e-3)	2.846 1e-1 (9.21e-4) -	1.682 2e-1 (1.69e-2) -	2.908 6e-1 (2.58e-3) ≈	2.809 5e-1 (2.10e-3) -
WFG 9	3.262 9e-1 (3.43e-2)	3.335 3e-1 (1.99e-3) +	3.105 3e-1 (4.04e-2) -	2.624 6e-1 (5.35e-2) -	3.349 0e-1 (3.42e-3) +
UF 1	6.546 5e-1 (2.57e-2)	6.050 1e-1 (1.89e-2) -	5.122 7e-1 (6.17e-2) -	5.252 1e-1 (1.25e-1) -	6.074 8e-1 (1.71e-2) -
UF 2	6.809 5e-1 (1.71e-2)	6.791 3e-1 (6.29e-3) -	6.374 5e-1 (1.63e-2) -	6.632 7e-1 (7.92e-3) -	6.7396e-1 (4.57e-3) -
UF 3	5.418 6e-1 (9.15e-2)	5.149 0e-1 (2.64e-2) -	4.425 1e-1 (4.12e-2) -	4.489 9e-1 (7.05e-2) -	5.112 9e-1 (3.30e-2) -
UF 4	3.837 4e-1 (3.00e-3)	3.241 6e-1 (4.63e-3) -	2.483 0e-1 (9.48e-3) -	3.279 3e-1 (1.32e-2) -	3.521 3e-1 (5.93e-3) -
UF 5	6.569 4e-2 (6.40e-2)	1.503 7e-3 (8.24e-3) -	4.400 4e-2 (6.13e-2) ≈	-	3.981 9e-3 (1.08e-2) -
UF 6	2.251 1e-1 (9.78e-2)	1.913 0e-1 (3.70e-2) ≈	2.702 5e-1 (7.69e-2) ≈	1.659 4e-1 (7.23e-2) -	9.363 0e-2 (3.50e-2) -
UF 7	5.372 7e-1 (6.88e-2)	4.265 1e-1 (1.01e-1) -	3.382 3e-1 (8.73e-2) -	3.177 6e-1 (1.75e-1) -	5.129 4e-1 (1.25e-2) -
DTLZ 1	8.484 4e-1 (1.55e-3)	-	7.504 5e-1 (1.56e-2) -	8.337 7e-1 (9.94e-3) -	5.091 7e-2 (1.03e-1) -
DTLZ 2	5.688 1e-1 (1.90e-3)	5.394 2e-1 (3.05e-3) -	4.519 6e-1 (1.47e-2) -	5.649 8e-1 (9.97e-4) -	4.780 9e-1 (8.53e-3) -
DTLZ 3	5.690 2e-1 (1.78e-3)	-	4.046 2e-1 (1.65e-2) -	3.051 6e-1 (2.49e-1) -	-
DTLZ 4	5.695 4e-1 (1.51e-3)	5.601 0e-1 (1.43e-3) -	5.567 3e-1 (5.64e-2) -	5.620 3e-1 (1.94e-2) -	5.473 3e-1 (2.72e-3) -
DTLZ 5	2.015 9e-1 (2.99e-5)	1.970 2e-1 (4.34e-4) -	1.676 2e-1 (7.53e-3) -	1.997 7e-1 (4.55e-5) -	1.953 6e-1 (7.41e-4) -
DTLZ 6	2.017 3e-1 (2.30e-5)	-	1.632 3e-1 (6.73e-3) -	2.001 0e-1 (5.25e-6) -	1.972 4e-1 (4.79e-4) -
DTLZ 7	2.836 6e-1 (6.87e-4)	2.232 4e-1 (7.09e-3) -	2.550 1e-1 (3.40e-3) -	2.574 5e-1 (5.05e-3) -	1.816 3e-1 (2.59e-2) -

对于各测试问题,算法独立运行30次,并分别计算IGD和HV的平均值和标准差,每项中的最优值加粗显示.若算法无法在指定评估次数内收敛,则HV指标展示为“-”.为了得到基于统计学角度的结论,

采用非参数统计Wilcoxon秩和检验将本文所提算法与其他4种算法相对比,显著水平设置为5%.表1和表2中符号“+”“-”“≈”分别表示其他算法表现显著优于、显著劣于或无差别于DCA-MOEA/D.

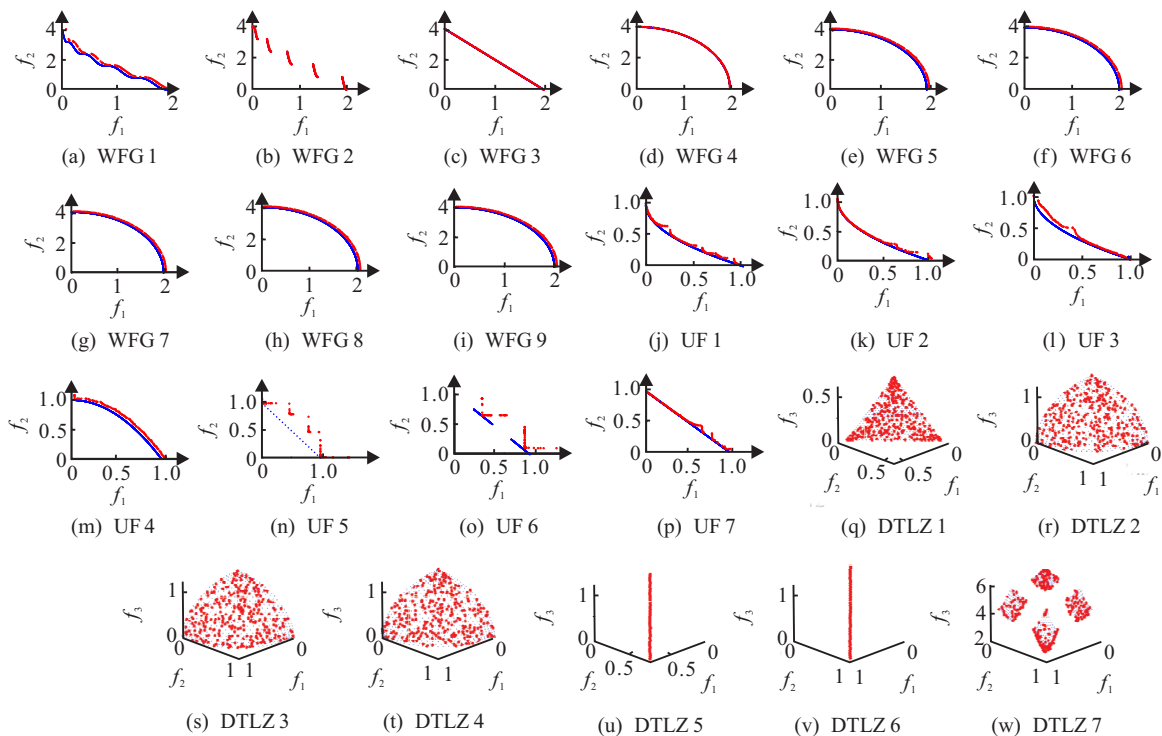


图1 DCA-MOEA/D在各测试函数上30轮运行中根据IGD指标得到的最优解

对于WFG测试集, DCA-MOEA/D在其中3个测试函数上取得最优结果, 算法在WFG上虽然无压倒性优势, 但结合图1来看, 求解效果总体上而言仍可接受. 同时, 将其与同样基于多算子混合的MOEA/D-FRRMAB作对比, 可以发现DCA-MOEA/D在WFG上的表现相对更加稳定, 无明显问题偏向, 这从侧面反映了基于MOE方式的多算子混合方式更加具有鲁棒性. 对于拥有相对复杂PS的UF测试集, DCA-MOEA/D在其中6个测试问题上的表现最优. 从结果上来看, 此类问题单独使用基于分解或基于支配关系的选择策略均无法取得令人满意的效果, 而DCA-MOEA/D通过引入基于支配关系维护的外部归档集,

将两种策略相结合, 使得算法在求解此类问题时具有明显优势. 在DTLZ测试集上, DCA-MOEA/D优势更为明显. 将DCA-MOEA/D与同样混合了两种选择策略的EAG-MOEA/D作对比, 可以发现多算子混合策略在解决三维优化问题时仍然有效. 结合两种评价指标秩和检验的结果, 从统计学的角度考虑, 可以认为本文提出的DCA-MOEA/D在求解不同类型的双目标或三目标优化问题时具有较好的泛化性.

为了直观查看DCA-MOEA/D的算子自适应选择过程, 根据各优化问题30次独立运行的结果, 对各算子在每代的平均使用次数进行了统计, 部分结果如图2所示.

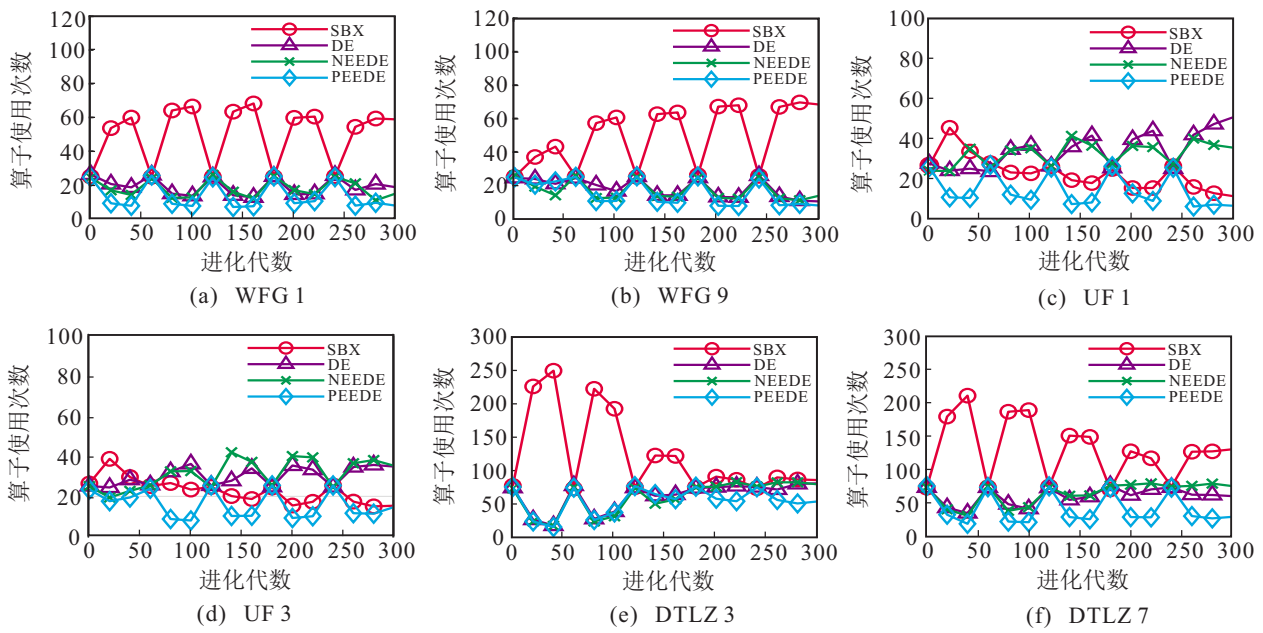


图2 各算子在每代的平均使用次数

从图2中可以清晰地看到DCA-MOEA/D在解决不同类型的优化问题时算子使用策略的差异. 从UF1、UF3和DTLZ3可以看到, 算子使用次数会随着进化阶段的改变而发生剧烈变化. 从WFG1、WFG9可以看到, 在各阶段表现最好的算子会被大量采用, 而重置操作对算子选择的连贯性产生的影响较小.

4 结论

为了混合多种进化算子来解决不同类型的多目标优化问题, 本文提出了一种基于双重贡献分配的多目标混合算子进化算法. 算法以MOEA/D为基本框架并引入一个外部归档集存储当前最优解并采用基于支配关系和拥挤度的策略对其进行维护. 结合两种贡献分配方法, 为各算子分配每代的贡献值, 以此自适应调整各算子的使用概率. 最后, 将整个进化过程划分为多个阶段来达到算子选择中“探索”与“探

究”之间的平衡. 虽然本文提出的算法对于解决双目标和三目标优化问题取得了较好的效果, 但是将其拓展到高维多目标优化问题时仍有较大的改进空间, 下一步将针对高维多目标优化问题对算法进行进一步优化.

参考文献(References)

[1] Tian Y, Zhang X Y, Cheng R, et al. Guiding evolutionary multiobjective optimization with generic front modeling[J]. IEEE Transactions on Cybernetics, 2020, 50(3): 1106-1119.
 [2] 李智翔, 贺亮, 韩杰思, 等. 一种基于偶图匹配的多目标分解进化算法[J]. 控制与决策, 2018, 33(10): 1782-1788.
 (Li Z X, He L, Han J S, et al. A bigraph matching method for decomposition multiobjective optimization[J]. Control and Decision, 2018, 33(10):

- 1782-1788.)
- [3] Panichella A. An adaptive evolutionary algorithm based on non-euclidean geometry for many-objective optimization[C]. Proceedings of the Genetic and Evolutionary Computation Conference. New York: Prague Czech Republic, 2019: 595-603.
- [4] Qin A K, Huang V L, Suganthan P N. Differential evolution algorithm with strategy adaptation for global numerical optimization[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(2): 398-417.
- [5] Kuk J, Goncalves R A, Pozo A, et al. Combining fitness landscape analysis and adaptive operator selection in multi and many-objective optimization[C]. The 8th Brazilian Conference on Intelligent Systems. Piscataway: IEEE, 2019: 503-508.
- [6] 耿焕同, 周山胜, 韩伟民, 等. 基于自适应进化策略的MOEA/D算法[J]. 计算机工程与设计, 2019, 40(4): 1106-1113.
(Geng H T, Zhou S S, Han W M, et al. MOEA/D algorithm based on adaptive evolutionary strategy[J]. Computer Engineering and Design, 2019, 40(4): 1106-1113.)
- [7] Zhang Q F, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712-731.
- [8] Li K, Fialho A, Kwong S, et al. Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition[J]. IEEE Transactions on Evolutionary Computation, 2014, 18(1): 114-130.
- [9] Santiago A, Dorronsoro B, Nebro A J, et al. A novel multi-objective evolutionary algorithm with fuzzy logic based adaptive selection of operators: FAME[J]. Information Sciences, 2019, 471(1): 233-251.
- [10] 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.
- [11] Wang C, Xu R, Qiu J F, et al. AdaBoost-inspired multi-operator ensemble strategy for multi-objective evolutionary algorithms[J]. Neurocomputing, 2020, 384: 243-255.
- [12] Storn R, Price K. Differential evolution — A simple and efficient heuristic for global optimization over continuous spaces[J]. Journal of Global Optimization, 1997, 11(4): 341-359.
- [13] Deb K, Agrawal R B. Simulated binary crossover for continuous search space[J]. Complex Systems, 1994, 9(3): 115-148.
- [14] Yan J F, Li M L, Xu Z J, et al. A simple Pareto adaptive ε -domination differential evolution algorithm for multi-objective optimization[J]. Open Automation & Control Systems Journal, 2015, 7(1): 338-345.
- [15] 李飞, 刘建昌, 石怀涛, 等. 基于分解和差分进化的多目标粒子群优化算法[J]. 控制与决策, 2017, 32(3): 403-410.
(Li F, Liu J C, Shi H T, et al. Multi-objective particle swarm optimization algorithm based on decomposition and differential evolution[J]. Control and Decision, 2017, 32(3): 403-410.)
- [16] Qiu X, Xu J X Y, Tan K C, et al. Adaptive cross-generation differential evolution operators for multiobjective optimization[J]. IEEE Transactions on Evolutionary Computation, 2016, 20(2): 232-244.
- [17] Cheng R, Jin Y C, Narukawa K, et al. A multiobjective evolutionary algorithm using gaussian process-based inverse modeling[J]. IEEE Transactions on Evolutionary Computation, 2015, 19(6): 838-856.
- [18] Li Y X, Cai X Y, Fan Z, et al. An external archive guided multiobjective evolutionary approach based on decomposition for continuous optimization[J]. IEEE Transactions on Evolutionary Computation, 2014, 1124-1130.
- [19] Liu H L, Gu F Q, Zhang Q F, et al. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems[J]. IEEE Transactions on Evolutionary Computation, 2014, 18(3): 450-455.
- [20] Tian Y, Cheng R, Zhang X Y, et al. PlatEMO: A MATLAB platform for evolutionary multi-objective optimization[J]. IEEE Computational Intelligence Magazine, 2017, 12(4): 73-87.
- [21] Venske S M, Gonçalves R A, Delgado M R. ADEMO/D: Multiobjective optimization by an adaptive differential evolution algorithm[J]. Neurocomputing, 2014, 127(15): 65-77.

作者简介

耿焕同(1973—), 男, 教授, 博士生导师, 从事计算智能与约束多目标优化、气象资料预处理与资料同化等研究, E-mail: genghuantong@163.com;

许可(1996—), 男, 硕士生, 从事多目标优化及其应用的研究, E-mail: xuke_96@qq.com;

戴中斌(1996—), 男, 硕士生, 从事多目标优化及其应用的研究, E-mail: 1209805090@qq.com;

徐小涵(1998—), 女, 硕士生, 从事多目标优化的研究, E-mail: 1064002311@qq.com.

(责任编辑: 闫妍)