

4-15-2024

Handling Constrained Multi-objective Optimization Problems Based on Relationship Between Pareto Fronts

Yubo Wang

School of Computer Science, China University of Geosciences, Wuhan 430074, China,
yubowang@cug.edu.cn

Chengyu Hu

School of Computer Science, China University of Geosciences, Wuhan 430074, China,
huchengyu@cug.edu.cn

Wenyin Gong

School of Computer Science, China University of Geosciences, Wuhan 430074, China

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation. For more information, please contact xtfzxb@126.com.

Handling Constrained Multi-objective Optimization Problems Based on Relationship Between Pareto Fronts

Abstract

Abstract: To address the challenges of balancing the constraint satisfaction and objective function optimization, and dealing with the complex feasible regions in constrained multi-objective optimization problems (CMOPs), a classification-based search approach is proposed based on different Pareto front relationships. A dual-population dual-phase framework is proposed in which an auxiliary population P_a and a main population P_m are evolved and the evolution process is divided into a learning phase and a search phase. During the learning phase, P_a explores unconstrained Pareto front (UPF) and P_m explores constrained Pareto front (CPF), through which the relationship between UPF and CPF is determined. After completing the learning phase, the different classified relationships guide the subsequent search strategies. In the search phase, the algorithm adaptively adjusts the search strategy of P_a to provide effective assistance for P_m according to the different classification relationships between UPF and CPF. Based on this framework, Pareto front relationships for different CMOPs are classified to achieve the more effective searching for CPF. Experimental results show that the proposed algorithm has a better performance compared with the seven state-of-the-art constrained multi-objective evolutionary algorithms (CMOEAs). Through learning and utilizing the relationship between UPF and CPF, the more appropriate search strategies can be selected to handle CMOPs with different characteristics and a more advantageous final solution set can be got.

Keywords

constrained multi-objective optimization, relationship between Pareto fronts, two-population, learning phase, search phase

Recommended Citation

Wang Yubo, Hu Chengyu, Gong Wenyin. Handling Constrained Multi-objective Optimization Problems Based on Relationship Between Pareto Fronts[J]. Journal of System Simulation, 2024, 36(4): 901-914.

基于帕累托前沿关系求解约束多目标优化问题

王昱博, 胡成玉*, 龚文引

(中国地质大学 计算机学院, 湖北 武汉 430074)

摘要: 为解决约束多目标优化问题中的平衡约束满足与目标函数优化以及可行域复杂等挑战, 提出了基于不同帕累托前沿关系的分类搜索方法。提出一种双种群双阶段框架: 进化一个辅助种群 P_a 和一个主种群 P_m , 并将进化过程分为学习阶段和搜索阶段。学习阶段, 种群 P_a 向 UPF (unconstrained Pareto front) 进行搜索, 而种群 P_m 向 CPF (constrained Pareto front) 进行搜索, 旨在探索 UPF 与 CPF 之间的关系; 完成学习后, 对不同问题的 UPF 与 CPF 关系进行分类, 以指导后续搜索策略; 在搜索阶段, 根据不同的分类关系, 调整种群 P_a 的搜索策略, 旨在使种群 P_a 为种群 P_m 提供更有效的辅助信息。基于此算法框架, 对不同类型约束多目标优化问题的帕累托前沿关系进行了分类, 实现了对 CPF 更有效的搜索。实验结果表明: 所提算法与其他7种先进的约束多目标优化算法相比具有更显著的性能优势。通过学习与利用 UPF 与 CPF 的关系, 能够选择更合适的搜索策略去应对具有不同特性的约束多目标优化问题, 以获得更具优势的最终解集。

关键词: 约束多目标优化; 帕累托前沿关系; 双种群; 学习阶段; 搜索阶段

中图分类号: TP18 文献标志码: A 文章编号: 1004-731X(2024)04-0901-14

DOI: 10.16182/j.issn1004731x.joss.22-1430

引用格式: 王昱博, 胡成玉, 龚文引. 基于帕累托前沿关系求解约束多目标优化问题[J]. 系统仿真学报, 2024, 36(4): 901-914.

Reference format: Wang Yubo, Hu Chengyu, Gong Wenyin. Handling Constrained Multi-objective Optimization Problems Based on Relationship Between Pareto Fronts[J]. Journal of System Simulation, 2024, 36(4): 901-914.

Handling Constrained Multi-objective Optimization Problems Based on Relationship Between Pareto Fronts

Wang Yubo, Hu Chengyu*, Gong Wenyin

(School of Computer Science, China University of Geosciences, Wuhan 430074, China)

Abstract: To address the challenges of balancing the constraint satisfaction and objective function optimization, and dealing with the complex feasible regions in constrained multi-objective optimization problems (CMOPs), a classification-based search approach is proposed based on different Pareto front relationships. A dual-population dual-phase framework is proposed in which an auxiliary population P_a and a main population P_m are evolved and the evolution process is divided into a learning phase and a search phase. During the learning phase, P_a explores unconstrained Pareto front (UPF) and P_m explores constrained Pareto front (CPF), through which the relationship between UPF and CPF is determined. After completing the learning phase, the different classified relationships guide the subsequent search strategies. In the search phase, the algorithm adaptively adjusts the search strategy of P_a to provide effective assistance for P_m according to the different classification relationships between UPF and CPF.

收稿日期: 2022-11-27 修回日期: 2023-01-09

基金项目: 国家自然科学基金(62073300, 62076225)

第一作者: 王昱博(1998-), 男, 博士生, 研究方向为智能计算及其应用。E-mail: yubowang@cug.edu.cn

通讯作者: 胡成玉(1978-), 男, 教授, 博导, 博士, 研究方向为智能计算及其应用。E-mail: huchengyu@cug.edu.cn

Based on this framework, Pareto front relationships for different CMOPs are classified to achieve the more effective searching for CPF. Experimental results show that the proposed algorithm has a better performance compared with the seven state-of-the-art constrained multi-objective evolutionary algorithms (CMOEA). Through learning and utilizing the relationship between UPF and CPF, the more appropriate search strategies can be selected to handle CMOPs with different characteristics and a more advantageous final solution set can be got.

Keywords: constrained multi-objective optimization; relationship between Pareto fronts; two-population; learning phase; search phase

0 引言

约束多目标优化问题在工程应用等现实生活中非常普遍且处于非常重要的地位。这些实际问题通常非常复杂、困难，是研究领域主要的热点和难点，如车辆路径^[1]、机器人夹持器优化^[2]、配水系统设计^[3]等。因此，研究者们开发了各种各样的进化算法来求解约束多目标优化问题。求解约束多目标优化问题不仅要考虑目标优化，同时要满足约束。为了更好地处理约束，进化算法通常采用了不同的约束处理方式，主要包括约束支配原则、 ϵ 约束处理方法、惩罚函数和随机排序等。

除采用不同的约束处理方式外，更多的算法框架被提出以解决约束多目标问题，如多阶段框架，多种群框架，多排序框架和混合框架。在多阶段框架中，文献[4]提出了两阶段框架。第一阶段，通过一组权重向量将约束多目标问题转化为单目标优化问题，以帮助寻找有潜力的可行区域。第二阶段，执行特定的约束多目标进化算法以获得最终的解集。文献[5]提出了推拉式搜索的两阶段框架。推阶段，不考虑任何约束，利用多目标进化算法对整个目标空间进行探索，直到搜索到UPF(unconstrained Pareto front)。拉阶段，采用改进的epsilon约束将推阶段获得的种群拉到CPF(constrained Pareto front)。在多种群框架中，文献[6]提出了双存档框架，一个种群以收敛为导向，使种群向CPF收敛，另一种群以多样性为导向，增加种群的多样性，并帮助种群跳出局部最优。在

多排序框架中，文献[7]提出了利用2种排序的适应度函数，分别使用帕累托支配关系评价目标和约束支配关系评价可行性。文献[8]提出了新的约束优化框架，具有可行性、收敛性和多样性3个指标。在混合框架中，文献[9]提出了一种混合进化模型，模型结合了进化策略和超体积牛顿法获得了可靠且快速的约束多目标算法。

一些研究人员也利用UPF信息求解约束多目标优化问题，文献[10]提出了弱协同进化框架，一个种群解决原始约束多目标优化问题，另一种群不考虑任何约束，解决派生出的辅助问题。该算法能够很好地解决UPF与CPF距离较近的约束多目标优化问题。但当UPF和CPF没有交集，或者交集特别小时，位于UPF附近的辅助种群所产生的后代几乎对解决原始约束多目标优化问题没有任何帮助^[11]。当CPF和UPF非常接近或重合时，更新辅助种群对于主种群有利。否则，应考虑另一种更有效的方法来利用辅助种群。基于此观点，文献[11]提出了一种探索UPF与CPF关系的框架，在学习阶段学习两者关系，并根据所学习的关系在下一阶段选择不同的算子，以更好地利用UPF对CPF进行搜索。由于URCMO在分类过程中使用了考虑约束的主种群信息，在具有复杂约束的约束多目标优化问题时，对UPF与CPF关系的分类可能会出现误差。

本文提出了一种新的基于帕累托前沿关系的算法(algorithm based on the relationship between Pareto fronts, RBPF)。本文所提RBPF算法进化一个辅助种群 P_a 和一个主种群 P_m ，并将整个进化过

程分为学习阶段和搜索阶段。学习阶段, P_a 忽略所有约束向 UPF 搜索, 种群 P_m 考虑所有约束向 CPF 搜索。并使用一种结合了自适应阶段转换与固定阶段转换的方法来判断学习阶段是否结束。在学习阶段结束后, 设计了一种新的仅基于种群 P_a 可行性信息对 UPF 和 CPF 分类的方法, 以减少分类误差。在搜索阶段, 根据上阶段的学习结果, 为种群 P_a 自适应调整进化策略, 旨在为种群 P_m 提供更有价值的辅助信息, 以更好地解决具有各种不同 UPF 与 CPF 关系的约束多目标优化问题。

1 约束多目标优化

以最小化问题为例, 约束多目标优化问题给出以下定义。

定义 1: 约束多目标优化问题

$$\begin{aligned} & \text{minimize } \mathbf{F}(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})\}^T, \\ & \text{s.t. } \mathbf{x} \in \mathbb{R}^D, \\ & \quad g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, p \\ & \quad h_j(\mathbf{x}) = 0, j = p + 1, p + 2, \dots, p + q \end{aligned}$$

式中: $\mathbf{F}(\mathbf{x})$ 是一个由 M 个目标函数值组成的 M 维目标向量; \mathbb{R}^D 是决策空间; $\mathbf{x} = \{x_1, x_2, \dots, x_D\}$ 是 D 维决策向量; $g_i(\mathbf{x})$ 和 $h_j(\mathbf{x})$ 分别为第 i 个不等式约束和第 $(j-p)$ 个等式约束。通常, 当目标个数 $M > 3$ 时, 称为约束超多目标优化问题, 当目标个数为 2 或者 3 时, 称为约束多目标优化问题。当解决约束多目标优化问题时, 任何一个解 \mathbf{x} 的约束违反度为

$$C(\mathbf{x}) = \sum_{i=1}^p \max(0, g_i(\mathbf{x})) + \sum_{j=p+1}^{p+q} |h_j(\mathbf{x})|$$

定义 2: Pareto 支配

对任意 2 个决策空间的解 \mathbf{x} 和 \mathbf{y} , 若满足 $f_i(\mathbf{x}) \leq f_i(\mathbf{y}), \forall i \in \{1, 2, \dots, M\}$, 且 $f_j(\mathbf{x}) < f_j(\mathbf{y}), \exists j \in \{1, 2, \dots, M\}$, 则称 \mathbf{x} 支配 \mathbf{y} , 记作 $\mathbf{x} < \mathbf{y}$ 。

定义 3: Pareto 最优解

若当前解集中不存在任何一个解支配解 \mathbf{x} , 那么称解 \mathbf{x} 为 Pareto 最优解, 也称非支配解。

定义 4: Pareto 最优解集

决策空间 \mathbb{R}^D 中所有的 Pareto 最优解组成 Pareto 最优解集。

定义 5: Pareto 前沿

Pareto 前沿是由 Pareto 最优解集中所有的 Pareto 最优解在目标空间的映射。

2 算法描述

2.1 算法框架

本文算法框架如图 1 所示。

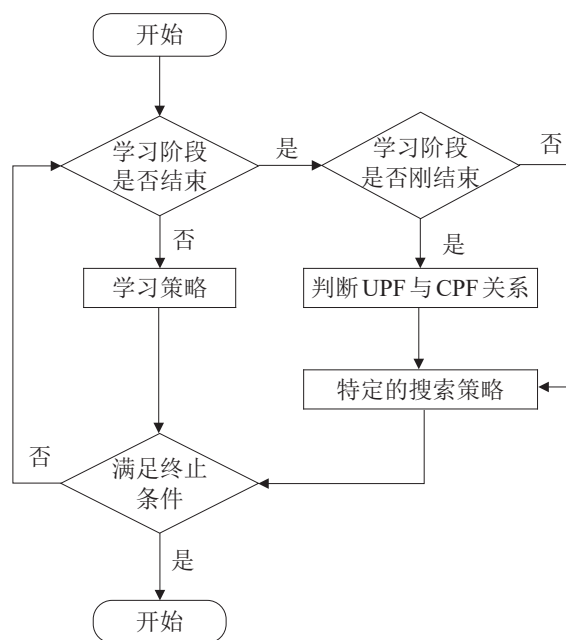


图 1 算法框架

Fig. 1 Algorithmic framework

2.2 算法设计

本文所提的 RBPF 算法框架分为 3 个模块: 学习阶段、分类方法和搜索阶段。

2.2.1 学习阶段

学习阶段的主要目的是让种群 P_a 完成对 UPF 的搜索, 以得到对 UPF 和 CPF 关系判断的基础信息。在学习阶段进化过程中, 种群 P_a 忽略所有约束, 只解决原始约束多目标优化问题所派生出的辅助问题, 旨在完成对 UPF 的搜索。

算法1 学习阶段

输入: 种群规模 N , 种群 P_a , 种群 P_m 输出: 完成学习后的种群 P_a , 种群 P_m step 1: 从种群 P_a 中通过竞标赛选择, 选择出 $N/2$ 个亲本, 记作 $Parent1$ 。step 2: 从种群 P_m 中通过竞标赛选择, 选择出 $N/2$ 个亲本, 记作 $Parent2$ 。step 3: 由亲本 $Parent1$ 产生 $N/2$ 个子代, 记作 O_a 。step 4: 由亲本 $Parent2$ 产生 $N/2$ 个子代, 记作 O_m 。step 5: 由 P_a 、 O_a 、 O_m 组成联合种群 T_a 。step 6: 由 P_m 、 O_a 、 O_m 组成联合种群 T_m 。step 7: 评估联合种群 T_a 以及 T_m 。step 8: 从联合种群 T_a 中通过环境选择策略选择出 N 个个体, 组成新的种群 P_a 。step 9: 从联合种群 T_m 中通过环境选择策略选择出 N 个个体, 组成新的种群 P_m 。

竞标赛选择和环境选择策略是基于评估解的质量来选择的。由于文献[12]所提出的截断策略以及适应度评价方法在文献[12]和文献[13]中表现出十分良好的性能, 因此, 本文同样采用了这种策略。在评估 P_a 时, 基于原始约束多目标优化问题所派生的辅助问题, 不考虑任何约束。而评估 P_m 时, 基于原始约束多目标优化问题, 考虑所有约束。在每次学习阶段开始前都要判断是否满足式(1)或者判断当前迭代次数是否已达到最大迭代次数, 若满足, 则学习阶段结束, 否则继续。

$$r_k \equiv \max \{r(z_k), r(n_k)\} \leq \gamma \quad (1)$$

式中: $r(z_k)$ 和 $r(n_k)$ 分别为10次迭代中种群 P_a 的理想点和极差点的变化率; γ 为常量, 设为0.001。若满足式(1), 则代表种群 P_a 已搜索到UPF, 并且在10次迭代中整个种群变化非常小。因此, 可以认为种群 P_a 已完成学习阶段任务, 进行阶段转换。

2.2.2 分类方法

UPF与CPF之间的关系可泛化为4种, 如图2所示。对于种类I, 由于UPF与CPF相同, 因此, 搜索UPF对搜索CPF十分有利。对于种类IV, 由于UPF远离CPF, 不包含任何CPF, 因此, 在UPF附近探索对搜索CPF是完全无用的。可以发现UPF与CPF之间的关系决定了种群 P_a 在搜索到UPF后, 是否应该继续在UPF附近搜索。因此, 对于UPF与CPF之间关系的学习和判断是十分重要的。

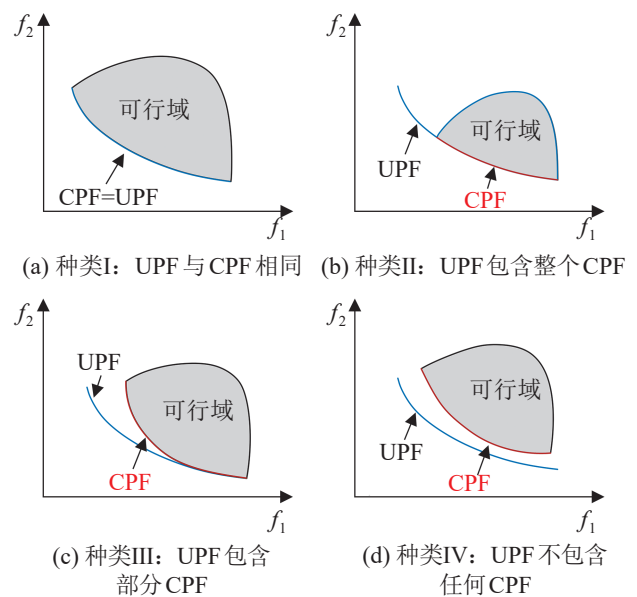


图2 UPF与CPF的关系

Fig. 2 Relationships between CPF and UPF

在学习阶段刚结束时, 通过对种群 P_a 的分布来确认UPF和CPF之间的关系。

算法2 分类方法

输入: 种群 P_a 输出: 问题类型 L step 1: 对 P_a 执行快速非支配排序, 得到非支配排序层为1的解, 记作 P_M , 数量记作 M 。step 2: 计算 P_M 中可行解的数量, 记作 M_1 。step 3: 计算 P_a 的非支配排序层为1的解的可行率 r_f , 大小为 M_1/M 。step 4: 若 $r_f < 0.3$ 则 $L=2$, 否则 $L=1$ 。

判断种群 P_a 对UPF的搜索是否能为CPF的搜索提供有利信息, 可以转为UPF包含多少CPF的信息。对于种类I和种类II类的约束多目标优化问题, UPF包含全部的CPF的信息, 对UPF的搜索对于解决原始约束多目标优化问题是十分有利的, 因此, 将种类I和种类II问题统一归为问题 L_1 。对于种类I问题, P_a 的非支配排序层为1的解的可行率 $r_f=1$ 。对于种类II问题, P_a 的非支配排序层为1的解的可行率 $r_f<1$, 且通常 >0.5 。因此, 可以通过 r_f 判断出要解决的约束多目标优化问题是否属于这两类。对于种类IV的约束多目标优化问题, UPF不包含任何CPF的信息, 在对UPF搜索完成后, 继续在UPF附近探索无法帮助解决原始约束多目标优化问题。因此, 种群 P_a 需要调整搜索策略, 同时将种类IV问题归为问题 L_2 。对于这类问题, 种群 P_a 的非支配排序层为1的解的可行率 $r_f=0$, 因此, 同样可以通过 r_f 判断要解决的约束多目标优化问题是否属于种类IV。然而, 对于种类III的约束多目标优化问题, 种群 P_a 的非支配排序层为1的解的可行率 $0<r_f<1$ 。根据问题不同, r_f 较小代表UPF包含CPF的信息很少, 可以认为与种类IV相同, 不再继续在UPF附近探索; r_f 较大代表UPF包含信息较多, 继续在UPF附近探索。因此, 若一个问题属于种类III, 可以通过种群 P_a 的非支配排序层为1的解的可行率 r_f 的大小, 进一步划分为种类III、III', 将种类III归为问题 L_1 , 将种类III'归为问题 L_2 。划分过程如下: 若 $0<r_f<0.3$, 则认为问题属于种类III', 若 $0.3\leq r_f<1$, 则认为问题属于种类III。

综上所述, 若 $0\leq r_f<0.3$, 则问题属于种类III'或IV, 既为问题 L_2 , 种群 P_a 不继续在UPF附近探索。若 $0.3\leq r_f<1$, 则问题属于种类II或III。对于种类II和III的约束多目标优化问题, 算法都将继续对UPF进行搜索, 因此, 不需要再仔细划分究竟属于哪一类问题。若 $r_f=1$, 则问题属于种类I。因此, 若 $0.3\leq r_f\leq 1$, 则为问题 L_1 , 种群 P_a 继续在UPF附近探索。整个分类过程如图3所示。

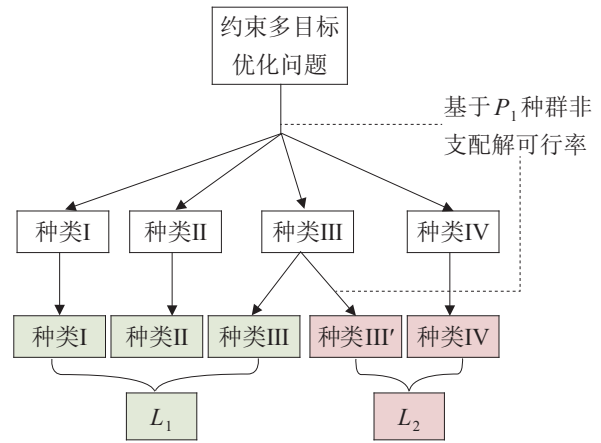


图3 分类过程

Fig. 3 Classification process

2.2.3 搜索阶段

由于种群 P_m 已经考虑了所有约束条件, 且在学习阶段与种群 P_a 相互协作已经具备较好的搜索能力, 所以, 在此阶段算法不改变种群 P_m 的搜索策略。因此, 搜索阶段根据学习任务完成后对约束多目标优化问题的分类结果, 为种群 P_a 自适应调整搜索策略。在 L_1 问题中, UPF包含了较多的CPF的信息, 因此, 在搜索阶段不调整种群 P_m 的搜索策略。在 L_2 问题中, 由于UPF包含的CPF信息较少, 因此, 在搜索阶段, 种群 P_a 将考虑所有的约束条件, 旨在使种群 P_a 对CPF进行搜索, 为种群 P_m 提供更有价值的辅助信息, 避免继续在UPF附近探索而浪费资源。

算法3 搜索阶段

输入: 种群规模 N , 种群 P_a , 种群 P_m

输出: 种群 P_a , 种群 P_m

step 1: 判断问题分类。

step 2: 从 P_a 中通过锦标赛选择, 选择出 $N/2$ 个亲本, 记作Parent1。

step 3: 从 P_m 中通过锦标赛选择, 选择出 $N/2$ 个亲本, 记作Parent2。

step 4: 由Parent1产生 $N/2$ 个子代, 记作 O_a 。

step 5: 由Parent2产生 $N/2$ 个子代, 记作 O_m 。

step 6: 由 P_a 、 O_a 、 O_m 组成联合种群 T_a 。

step 7: 由 P_m 、 O_a 、 O_m 组成联合种群 T_m 。

step 8: 评估 T_a 以及 T_m 。

step 9: 从 T_a 中通过环境选择策略选择出 N 个个体, 组成新的种群 P_a 。

step 10: 从 T_m 中通过环境选择策略选择出 N 个个体, 组成新的种群 P_m 。

算法3与算法1的框架十分相似, 不同在于算法3会对问题分类进行判断。若问题属于 L_1 , 则算法3与算法1执行过程相同。若问题属于 L_2 , 则在评估种群 P_a 的质量时, 结合了文献[12]中的适应度评价方法与约束松弛。同时, 执行竞标赛选择策略以及环境选择策略时, 都是基于上述方式所评估出的解的质量。约束松弛更新规则^[9]:

$$\varepsilon(k) = \begin{cases} (1-\tau)\varepsilon(k-1), & r_f < \alpha \\ \varepsilon(0)(1-k/T_c)^{p_c}, & r_f \geq \alpha \end{cases} \quad (2)$$

式中: k 为当前迭代次数; τ 为控制约束松弛减小速度的参数, 设为 0.05; r_f 为种群 P_a 的可行率; $\varepsilon(0)$ 为学习阶段刚结束时, 种群 P_a 总体约束违反度的最大值; T_c 为控制迭代数; p_c 为控制约束松弛减小速度的参数, 设为 2; α 控制种群对不可行区域和可行区域的搜索偏好, 设为 0.95。

2.2.4 外存档策略

在本文所提算法中, 为保证最终解集是分布良好的可行非支配解, 采用了外存档策略。在算法开始时, 外存档进行初始化, 随着每一次算法迭代, 将 2 个种群所产生的所有子代加入外存档中, 并与外存档中原所有个体进行比较, 最终选择 N 个最具优势的个体进行保留。外存档的更新策略: 将所有需要比较的个体合并成一个种群, 该种群使用文献[12]中适应度评价方法以及截断策略, 选择出 N 个最具优势的个体。

2.2.5 主框架

RBPF 算法进化了 2 个种群, 设置了 2 个阶段, 旨在学习 UPF 与 CPF 关系并用以指导后续的搜索过程。

算法4 主框架

输入: 种群规模 N , 最大迭代次数 T_{\max}

输出: 外存档 A

step 1: 算法初始化

step 1.1: 采用随机方法初始化规模为 N 的 2 个种群 P_a 、 P_m 。

step 1.2: 初始化约束松弛量 $\varepsilon=0$ 。

step 1.3: 初始化 $\mathbf{z}_{\min}=(z_1^{\min}, z_2^{\min}, \dots, z_m^{\min})$, $\mathbf{z}_{\max}=(z_1^{\max}, z_2^{\max}, \dots, z_m^{\max})$, 其中, \mathbf{z}_{\min} 和 \mathbf{z}_{\max} 的各个分量分别为 P_a 在各个目标分量上的最小值和最大值。

step 1.4: 初始化外存档 A 。

step 1.5: 初始化 $r_k=1$, $L=1$ 。

step 1.6: 初始化当前迭代次数 $T=1$ 。

step 2: 若 $T \geq 10$, 则通过式(1)计算 r_k 。

step 3: 若 $r_k \leq \gamma$ $T = 0.3 \times T_{\max}$, 则认为学习阶段结束。判断是否为学习阶段刚结束, 若刚结束, 则执行算法2更新 L 。

step 4: 若 $L=1$, 则跳转到 step 5, 否则跳转到 step 6。

step 5: 执行算法1, 跳转到 step 7。

step 6: 否则执行算法3, 通过式(2)更新 ε

step 7: 使用外存档策略更新外存档 A 。

step 8: 检查是否满足停止条件, 若是, 则算法终止并输出外存档 A , 否则 $T=T+1$ 并跳转到 step 2。

3 数值仿真实验

为了验证 RBPF 算法能够处理具有各种不同帕累托前沿关系的约束多目标优化问题, 将 RBPF 与 CMOEA-MS^[13]、BiCo^[14]、C-TAEA^[6]、PPS^[5]、ToP^[4]、TiGE-2^[8]和 URCMO^[11]在 3 个测试集上进行比较。所有实验均在 PlatEMO^[15]平台上进行。

3.1 实验设置

3.1.1 测试问题

从 DTLZ^[16]、DAS^[17]、LIR^[18]测试集中选择了 33 个具有不同特点的约束多目标优化问题。目标数 M 和决策变量数 D 设置: 对于 DTLZ 测试集, C1-DTLZ1、DC1-DTLZ1、DC2-DTLZ1 和 DC3-

DTLZ1中, $M=3$, $D=7$, 而其余问题 $D=12$; 对于DAS和LIR测试集, LIR-CMOP13、LIR-CMOP14和DAS-CMOP7、DAS-CMOP8、DAS-CMOP9中, $M=3$, $D=15$, 而其余问题 $M=2$ 。

3.1.2 遗传算子和参数设置

差分进化: 交叉概率 $R_c=1$, 比例系数 $F=0.5$; 模拟二进制交叉: 交叉概率 $p_c=0.9$, 分布指数 $\eta_c=20$; 多项式变异: 变异概率 $p_m=1/n$, 分布指数 $\eta_m=20$ 。实验种群为100, 算法独立运行次数为30。DTLZ测试集函数评价次数为60 000, DAS和LIR测试集函数评价次数为100 000, 其他参数设置和原文一致。

3.1.3 性能指标

IGD+和HV为常用的评价指标, 更小的IGD+值和更大的HV值意味着非支配解集有更好的质量。

IGD+同时兼顾了收敛和多样性的性能, 数学上可定义为

$$IGD^+(\mathbf{Z}, \mathbf{A}) = \frac{1}{|\mathbf{Z}|} \sum_{j=1}^{|\mathbf{Z}|} \min_{a_i \in \mathbf{A}} d(a_i, z_j)$$

$$d(a, z) = \sqrt{\sum_{k=1}^M (\max\{a_k - z_k, 0\})^2}$$

式中: \mathbf{Z} 为预定义参考点集合; \mathbf{A} 为解集。

HV表示约束多目标优化算法获得的非支配解集与CPF的接近性, 数学上可定义为

$$HV(\mathbf{A}) = \text{VOL}(\cup[z_1, z_1^r] \times \cdots \times [z_m, z_m^r])$$

式中: VOL为勒贝格测度; z_m^r 为预定义参考点。

3.2 实验结果

3.2.1 LIR测试集实验结果

由于LIR-CMOPs的复杂约束, LIR-CMOPs的CPF以及可行域十分多样, 意味着有各种不同的UPF与CPF关系的约束多目标优化问题。LIR-CMOP1-4的可行域十分窄小; LIR-CMOP5-6的CPF与UPF相同, 分别为凸型和凹型; LIR-CMOP7-8的UPF位于不可行区域, CPF位于约束边界上; LIR-CMOP9-10的CPF是UPF的一部分; LIR-CMOP11-12的CPF位于约束边界; LIR-CMOP13-14的CPF与UPF相同。

将LIR-CMOPs划分为

种类I: LIR-CMOP5-6, 13, 14。

种类II: LIR-CMOP9-10。

种类III: LIR-CMOP11-12。

种类IV: LIR-CMOP1-4, 7-8。

在LIR-CMOPs上IGD+和HV的统计结果分别如表1~2所示, 由结果可知, 所提RBPF算法优于其他算法。

表1 不同算法在LIR-CMOP1-14上的IGD+结果

Table 1 IGD+ results of different algorithms on LIR-CMOP1-14

问题	BiCo	C-TAEA	CMOEA-MS	PPS	ToP	TiGE-2	URCMO	RBPF
LIRCMOP5	1.230 00 (0.004 24)-	1.200 00 (0.196 00)-	0.250 00 (0.053 40)-	0.006 88 (0.001 44)+	1.200 00 (0.022 30)-	1.010 00 (0.398 00)-	0.123 00 (0.290 00)=	0.142 00 (0.297 00)
LIRCMOP6	1.350 0 (0.000 184)-	1.220 0 (0.341 000)-	0.368 0 (0.271 000)-	0.050 3 (0.245 000)+	1.300 0 (0.185 000)-	1.060 0 (0.454 000)-	0.075 9 (0.106 000)=	0.260 0 (0.450 000)
LIRCMOP13	1.320 0 (0.002 01)-	0.047 1 (0.001 36)-	0.043 9 (0.001 27)=	0.077 9 (0.004 49)-	1.320 0 (0.009 00)-	1.100 0 (0.481 00)-	0.063 4 (0.002 93)-	0.043 4 (0.001 07)
LIRCMOP14	1.270 0 (0.002 110)-	0.048 6 (0.000 738)-	0.044 2 (0.000 829)+	0.062 6 (0.002 400)-	1.290 0 (0.078 000)-	1.030 0 (0.494 000)-	0.050 9 (0.001 860)-	0.045 4 (0.001 520)
LIRCMOP9	0.909 (0.134 0)-	0.317 (0.086 5)-	0.491 (0.142 0)-	0.221 (0.084 1)-	0.362 (0.143 0)-	0.509 (0.179 0)-	0.195 (0.088 7)=	0.158 (0.028 4)
LIRCMOP10	0.933 0 (0.072 3)-	0.301 0 (0.042 2)-	0.409 0 (0.198 0)-	0.197 0 (0.097 9)-	0.332 0 (0.078 8)-	0.745 0 (0.159 0)-	0.115 0 (0.058 9)-	0.058 3 (0.081 0)

续表

问题	BiCo	C-TAEA	CMOEA-MS	PPS	ToP	TiGE-2	URCMO	RBPF
LIRCMOP11	0.677 (0.214 0)-	0.175 (0.038 7)-	0.280 (0.148 0)-	0.230 (0.144 0)-	0.413 (0.119 0)-	0.713 (0.164 0)-	0.150 (0.139 0)-	0.112 (0.018 7)
LIRCMOP12	0.556 (0.207 0)-	0.235 (0.151 0)=	0.315 (0.115 0)-	0.151 (0.071 2)=	0.223 (0.056 2)-	0.371 (0.077 2)-	0.160 (0.043 8)=	0.144 (0.029 9)
LIRCMOP1	0.188 0 (0.011 3)-	0.259 0 (0.049 5)-	0.268 0 (0.032 3)-	0.024 2 (0.033 6)+	0.266 (0.012 5)-	0.186 (0.016 0)-	0.112 0 (0.045 3)-	0.049 3 (0.030 5)
LIRCMOP2	0.110 0 (0.012 20)-	0.140 0 (0.056 00)-	0.180 0 (0.022 70)-	0.010 9 (0.007 31)+	0.185 0 (0.019 40)-	0.114 0 (0.009 56)-	0.024 1 (0.033 10)+	0.024 6 (0.012 90)
LIRCMOP3	0.187 0 (0.023 9)-	0.259 0 (0.062 7)-	0.252 0 (0.046 2)-	0.071 1 (0.061 7)=	0.282 0 (0.010 5)-	0.185 0 (0.015 2)-	0.064 7 (0.047 4)-	0.034 3 (0.021 1)
LIRCMOP4	0.130 0 (0.020 4)-	0.203 0 (0.085 5)-	0.197 0 (0.021 1)-	0.040 4 (0.037 2)=	0.202 0 (0.015 9)-	0.123 0 (0.013 8)-	0.072 9 (0.056 0)-	0.029 6 (0.021 4)
LIRCMOP7	0.479 0 (0.675 0)-	0.338 0 (0.544 0)-	0.108 0 (0.020 3)-	0.108 0 (0.031 3)-	1.310 0 (0.652 0)-	0.251 0 (0.091 2)-	0.021 7 (0.031 7)-	0.015 4 (0.023 9)
LIRCMOP8	1.240 0 (0.690 0)-	0.712 0 (0.662 0)-	0.180 0 (0.044 3)-	0.157 0 (0.051 7)-	1.540 0 (0.454 0)-	0.391 0 (0.273 0)-	0.011 6 (0.018 4)+	0.033 3 (0.072 3)
+/-/=	0/14/0	0/13/1	1/12/1	4/7/3	0/14/0	0/14/0	2/8/4	

注：“+”“-”“=”分别表示其他算法优于、劣于和等于本文算法。

表 2 不同算法在 LIR-CMOP1-14 上的 HV 结果

Table 2 HV results of different algorithms on LIR-CMOP1-14

问题	BiCo	C-TAEA	CMOEA-MS	PPS	ToP	TiGE-2	URCMO	RBPF
LIRCMOP5	0 (0)-	0.006 14 (0.033 6)-	0.145 00 (0.022 1)-	0.290 00 (0.001 3)+	0 (0)-	0.029 00 (0.054 5)-	0.252 00 (0.074 5)=	0.245 00 (0.078 9)
LIRCMOP6	0 (0)-	0.009 96 (0.026 0)-	0.092 50 (0.026 7)-	0.190 00 (0.035 9)+	0.001 90 (0.010 4)-	0.024 00 (0.037 5)-	0.169 00 (0.034 5)=	0.138 00 (0.074 3)
LIRCMOP13	0.000 115 (0.000 137)-	0.546 000 (0.001 880)-	0.556 000 (0.001 290)=	0.510 000 (0.006 180)-	0.003 310 (0.012 500)-	0.072 100 (0.138 000)-	0.535 000 (0.002 910)-	0.556 000 (0.001 050)
LIRCMOP14	0.000 37 (0.000 334)-	0.546 00 (0.000 833)-	0.556 00 (0.000 965)+	0.530 00 (0.003 720)-	0.003 68 (0.018 600)-	0.089 60 (0.153 000)-	0.549 00 (0.002 010)-	0.555 00 (0.001 240)
LIRCMOP9	0.137 (0.057 8)-	0.349 (0.056 5)-	0.279 (0.064 3)-	0.437 (0.048 3)-	0.331 (0.074 7)-	0.242 (0.083 3)-	0.442 (0.065 4)=	0.449 (0.026 3)
LIRCMOP10	0.064 7 (0.017 9)-	0.504 0 (0.034 6)-	0.413 0 (0.155 0)-	0.579 0 (0.071 9)-	0.472 0 (0.075 0)-	0.151 0 (0.048 8)-	0.639 0 (0.038 2)-	0.672 0 (0.055 7)
LIRCMOP11	0.270 (0.102 0)-	0.606 (0.250 0)-	0.490 (0.108 0)-	0.512 (0.110 0)-	0.376 (0.079 6)-	0.191 (0.072 2)-	0.577 (0.109 0)-	0.607 (0.016 8)
LIRCMOP12	0.362 (0.097 7)-	0.506 (0.054 1)=	0.440 (0.050 8)-	0.525 (0.050 2)=	0.475 (0.043 6)-	0.370 (0.052 7)-	0.521 (0.030 9)=	0.530 (0.019 3)
LIRCMOP1	0.136 (0.005 18)-	0.114 (0.021 10)-	0.108 (0.012 90)-	0.225 (0.020 50)+	0.107 (0.008 56)-	0.137 (0.008 17)-	0.176 (0.024 90)-	0.206 (0.016 50)
LIRCMOP2	0.262 (0.009 11)-	0.260 (0.030 30)-	0.221 (0.017 00)-	0.355 (0.007 64)+	0.217 (0.016 10)-	0.262 (0.008 36)-	0.338 (0.033 10)-	0.341 (0.013 20)
LIRCMOP3	0.126 0 (0.010 30)-	0.096 5 (0.027 60)-	0.102 0 (0.016 80)-	0.173 0 (0.027 20)=	0.090 5 (0.006 14)-	0.124 (0.008 65)-	0.176 (0.024 10)-	0.188 (0.008 00)

<http://www.china-simulation.com>

• 908 •

续表

问题	BiCo	C-TAEA	CMOEA-MS	PPS	ToP	TiGE-2	URCMO	RBPF
LIRCMOP4	0.224 (0.012 50)-	0.191 (0.038 20)-	0.188 (0.013 60)-	0.288 (0.276 00)=	0.185 (0.108 00)-	0.230 (0.009 14)-	0.265 (0.039 00)-	0.294 (0.016 40)
LIRCMOP7	0.187 0 (0.105 00)-	0.208 0 (0.085 70)-	0.245 0 (0.008 76)-	0.246 0 (0.013 20)-	0.052 8 (0.098 80)-	0.196 0 (0.023 00)-	0.287 0 (0.014 80)-	0.289 00 (0.011 50)
LIRCMOP8	0.066 2 (0.103 00)-	0.135 0 (0.099 10)-	0.230 0 (0.010 40)-	0.235 0 (0.016 30)-	0.021 50 (0.066 00)-	0.174 (0.039 10)-	0.291 (0.008 90)+	0.284 (0.024 00)
+/-/=	0/14/0	0/13/1	1/12/1	4/7/3	0/14/0	0/14/0	1/9/4	

注：“+”“-”“=”分别表示其他算法优于、劣于和等于本文算法。

对于种类I问题, 所有对比算法中, PPS表现最佳, 本文所提RBPF算法仅次于PPS。原因在于, 基于分解框架的PPS算法, 配备了 ϵ 方法, 能利用大量的不可行解维持多样性。因此, 表现较好。由于RBPF算法框架和LIR-CMOP5-6的复杂性, 辅助种群有时未能充分搜索UPF, 学习阶段便已结束。因此, 可能导致误判问题类型, 造成搜索能力下降。但所提RBPF算法在处理种类I的问题上, 仍表现出十分有竞争力的性能。

对于种类II问题, RBPF均获得了最佳结果。原因在于, LIR-CMOP9-10没有大规模不可行区域, 通过所用的框架和约束违反度优先方法较好地搜索了CPF。

对于种类III问题, RBPF能够很好地调整搜索阶段的策略, 使辅助种群更有效地搜索CPF, 获得了最佳结果。

对于种类IV问题, RBPF、PPS和URCMO分别取得3、2和1个最佳结果。PPS由于之前提到的策略, 能够很好地处理LIR-CMOP1-2这类可行域十分窄小的问题。URCMO思想与RBPF相似, 但其分类方法以及第2阶段使用的策略与RBPF完全不同。RBPF仅使用辅助种群的信息进行分类, 因此, 误判情况较少。并且RBPF在搜索阶段所采用的是约束处理技术的改变。相较于URCMO改变算子的策略, 在约束较为复杂的问题上, RBPF表现更好。

综上所述, 本文所提RBPF算法能够很好地利用UPF与CPF关系去解决具有不同特点的约束

多目标优化问题, 且表现出了优秀的性能优势。

3.2.2 在DAS与DTLZ测试集上的实验结果

由于大规模不可行区域的阻挡, DAS测试集的CPF也是离散的, 并且一些问题的可行域远离UPF。因此, 许多约束多目标优化算法解决DAS-CMOPs时可能会遇到一些困难: 收敛困难、可行性困难和多样性困难。表3~4分别显示了在DAS-CMOPs上的IGD+和HV统计结果, 再次证实了本文所提算法的优越性。由于RBPF算法能够根据不同问题调整搜索策略, 更有效地指导种群搜索CPF。因此, 在一定程度上克服了收敛困难、可行性困难以及多样性困难, 在DAS测试集上较其他7个约束多目标优化算法表现出了十分优秀的性能。同样地, 表5~6分别显示了在DTLZ测试集上的IGD+和HV统计结果。上述实验充分证明了RBPF算法的有效性、竞争力以及能够处理具有各种不同类型UPF与CPF关系的约束多目标优化问题的能力。

3.2.3 Friedman Test

使用KEEL软件对所有算法在所有测试集上的IGD+与HV结果进行Friedman秩和检验, 如表7所示, RBPF算法在IGD+以及HV结果上均获得了最佳排序值, 所有对比算法的 p -值均小于显著性水平0.05, 因此, 所有对比算法与RBPF相比有明显差异。证明RBPF算法比其他算法具有显著优势。

表 3 不同算法在 DAS-CMOP1-9 上的 IGD+ 结果
Table 3 IGD+ results of different algorithms on DAS-CMOP1-9

问题	BiCo	C-TAEA	CMOEA-MS	PPS	ToP	TiGE-2	URCMO	RBPF
DASCMP1	0.729 00 (0.024 800)-	0.166 00 (0.003 170)-	0.728 00 (0.053 900)-	0.059 70 (0.104 000)-	0.782 00 (0.041 500)-	0.608 00 (0.143 000)-	0.017 80 (0.085 500) =	0.002 23 (0.000 102)
DASCMP2	0.151 00 (0.007 590)-	0.056 30 (0.018 800)-	0.151 00 (0.009 150)-	0.004 03 (0.000 180)-	0.671 00 (0.231 000)-	0.130 00 (0.040 500)-	0.006 44 (0.014 300) =	0.003 86 (0.000 180)
DASCMP3	0.187 00 (0.014 100)-	0.122 00 (0.031 000)-	0.197 00 (0.015 300)-	0.129 00 (0.082 0000)-	0.685 00 (0.161 000)-	0.190 00 (0.024 000)-	0.139 00 (0.073 500)-	0.005 54 (0.000 142)
DASCMP4	0.063 20 (0.113 00)-	0.007 59 (0.001 50)-	0.044 10 (0.086 40) =	0.243 00 (0.145 00)-	NaN (NaN)	0.018 10 (0.007 27)-	0.112 00 (0.143 00)-	0.007 27 (0.035 80)
DASCMP5	0.075 80 (0.155 000)-	0.005 72 (0.000 833)-	0.071 00 (0.160 000) =	0.167 00 (0.253 000)-	NaN (NaN)	0.018 70 (0.004 000)-	0.037 60 (0.099 900)-	0.001 83 (0.000 081)
DASCMP6	0.145 0 (0.179 00)-	0.015 4 (0.008 06) =	0.216 0 (0.221 00)-	0.247 0 (0.303 00)-	NaN (NaN)	0.075 2 (0.139 00)-	0.361 0 (0.175 00)-	0.039 7 (0.092 20)
DASCMP7	0.025 8 (0.001 710)-	0.027 2 (0.001 010)-	0.023 0 (0.000 633) =	0.187 0 (0.164 000)-	NaN (NaN)	0.099 2 (0.017 6000)-	0.030 8 (0.011 000)-	0.023 4 (0.001 010)
DASCMP8	0.019 4 (0.002 570) =	0.022 4 (0.002 570)-	0.018 4 (0.000 588)+	0.165 0 (0.208 000)-	NaN (NaN)	0.059 0 (0.015 800)-	0.022 9 (0.004 160)-	0.018 8 (0.000 753)
DASCMP9	0.251 0 (0.027 10)-	0.186 0 (0.031 90)-	0.232 0 (0.003 00)-	0.166 0 (0.128 00)-	0.458 0 (0.094 30)-	0.211 0 (0.026 10)-	0.023 6 (0.001 02)-	0.021 2 (0.001 12)
+/-/=	0/8/1	0/8/1	1/5/3	0/9/0	0/4/0	0/9/0	0/7/2	

注：“+”“-”“=” 分别表示其他算法优于、劣于和等于本文算法。

表 4 不同算法在 DAS-CMOP1-9 上的 HV 结果
Table 4 HV results of different algorithms on DAS-CMOP1-9

问题	BiCo	C-TAEA	CMOEA-MS	PPS	ToP	TiGE-2	URCMO	RBPF
DASCMP1	0.007 04 (0.004 950)-	0.167 00 (0.002 790)-	0.007 89 (0.012 100)-	0.195 00 (0.028 900)-	0.002 44 (0.004 550)-	0.03 25 (0.040 000)-	0.207 00 (0.025 100) =	0.212 00 (0.000 588)
DASCMP2	0.249 0 (0.004 700)-	0.306 0 (0.008 150)-	0.254 0 (0.003 580)-	0.355 0 (0.000 113) =	0.061 7 (0.079 200)-	0.266 0 (0.014 400)-	0.353 0 (0.012 000) =	0.355 0 (0.000 145)
DASCMP3	0.212 0 (0.003 860)-	0.243 0 (0.015 200)-	0.209 0 (0.000 217)-	0.243 0 (0.046 100)-	0.036 7 (0.052 600)-	0.212 0 (0.011 900)-	0.238 0 (0.041 600)-	0.312 0 (0.000 298)
DASCMP4	0.177 (0.044 00)-	0.195 (0.005 15)-	0.190 (0.024 10)-	0.126 (0.057 50)-	NaN (NaN)	0.180 (0.008 06)-	0.157 (0.056 10)-	0.203 (0.007 35)
DASCMP5	0.306 (0.092 700)-	0.348 (0.000 706)-	0.311 (0.092 800) =	0.263 (0.129 000)-	NaN (NaN)	0.338 (0.002 920)-	0.329 (0.058 000)-	0.352 (0.000 100)
DASCMP6	0.236 (0.094 20)-	0.307 (0.002 21) =	0.200 (0.115 00)-	0.202 (0.124 00)-	NaN (NaN)	0.275 (0.064 80)-	0.128 (0.086 70)-	0.292 (0.051 60)
DASCMP7	0.287 (0.000 187)-	0.287 (0.000 765)-	0.289 (0.000 198)+	0.212 (0.073 700)-	NaN (NaN)	0.252 (0.008 890)-	0.284 (0.005 740)-	0.289 (0.000 417)
DASCMP8	0.206 (0.001 900)-	0.203 (0.001 620)-	0.208 (0.000 261)+	0.142 (0.073 000)-	NaN (NaN)	0.182 (0.007 310)-	0.204 (0.002 570)-	0.207 (0.000 332)
DASCMP9	0.125 0 (0.007 510)-	0.143 0 (0.010 200)-	0.131 0 (0.009 750)-	0.152 0 (0.039 100)-	0.063 9 (0.024 200)-	0.125 0 (0.008 000)-	0.203 0 (0.000 482)-	0.204 (0.000 526)
+/-/=	0/9/0	0/8/1	2/6/1	0/8/1	0/4/0	0/9/0	0/7/2	

注：“+”“-”“=” 分别表示其他算法优于、劣于和等于本文算法。

表5 不同算法在DTLZ测试集上的IGD+结果
Table 5 IGD+ results of different algorithms on DTLZ benchmark suit

问题	BiCo	C-TAEA	CMOEA-MS	PPS	ToP	TiGE-2	URCMO	RBPF
C1_DTLZ1	0.014 6 (0.000 221)-	0.016 3 (0.000 128)-	0.014 7 (0.000 275)-	0.019 0 (0.000 838)-	NaN (NaN)	0.249 0 (0.088 100)-	0.018 0 (0.001 410)-	0.014 2 (0.000 172)
C1_DTLZ3	1.120 0 (2.750 0)-	0.128 0 (0.461 0)-	0.230 0 (1.460 0)=	2.970 0 (3.910 0)-	0.492 0 (1.470 0)-	5.210 0 (3.460 0)-	2.900 0 (3.730 0)-	0.024 2 (0.001 4)
C2_DTLZ2	0.019 2 (0.000 610)=	0.023 5 (0.000 438)-	0.019 5 (0.000 907)-	0.024 8 (0.000 716)-	0.036 4 (0.007 100)-	0.037 4 (0.002 860)-	0.022 2 (0.002 090)-	0.018 5 (0.000 654)
C3_DTLZ4	0.059 3 (0.002 16)-	0.061 0 (0.002 38)-	0.321 0 (0.041 70)-	0.097 2 (0.034 60)-	0.106 0 (0.007 05)-	0.086 7 (0.003 83)-	0.069 4 (0.003 33)-	0.056 1 (0.002 29)
DC1_DTLZ1	0.008 67 (0.000 599)=	0.010 60 (0.000 126)-	0.010 60 (0.003 260)-	0.029 50 (0.045 600)-	0.018 90 (0.004 020)-	0.455 00 (0.306 000)-	0.009 86 (0.000 668)-	0.008 44 (0.000 417)
DC1_DTLZ3	0.014 4 (0.001 160)=	0.016 1 (0.000 708)-	0.076 1 (0.089 700)=	0.469 0 (0.726 000)-	1.600 0 (2.300 000)-	1.820 0 (0.668 000)-	0.027 9 (0.028 900)-	0.014 4 (0.001 210)
DC2_DTLZ1	0.131 0 (0.075 900)-	0.016 4 (0.000 175)+	0.015 1 (0.000 520)+	0.031 5 (0.038 200)-	NaN (NaN)	0.255 0 (0.069 600)-	0.028 9 (0.028 400)-	0.024 5 (0.037 900)
DC2_DTLZ3	0.565 (0.003 07)-	0.374 (0.258 00)=	0.455 (0.218 00)-	0.375 (0.264 00)-	NaN (NaN)	0.989 (0)=	0.579 (0.016 10)-	0.152 (0.206 00)
DC3_DTLZ1	0.083 90 (0.082 600)-	0.006 51 (0.000 165)+	0.025 80 (0.006 820)-	0.561 00 (0.956 000)-	2.070 00 (2.960 000)-	1.450 00 (0.777 000)-	0.018 00 (0.067 200)-	0.010 60 (0.030 000)
DC3_DTLZ3	1.200 00 (0.491 000)-	0.009 81 (0.000 503)+	0.299 00 (0.424 000)+	1.940 00 (1.770 000)-	7.880 00 (3.980 000)-	3.580 00 (0.724 000)-	0.722 00 (0.496 000)-	0.573 00 (0.320 000)
+/-/=	0/7/3	3/6/1	2/6/2	0/10/0	0/7/0	0/9/1	0/10/0	

注：“+”“-”“=”分别表示其他算法优于、劣于和等于本文算法。

3.3 模块作用验证

将RBPF与其5个变体在LIR、DAS和DTLZ测试集上进行比较。RBPF1在完成学习阶段后不考虑学习结果，在搜索阶段 P_a 始终不考虑任何约束；RBPF2在完成学习阶段后不考虑学习结果，在搜索阶段 P_a 采用约束松弛；RBPF3不进行学习阶段的学习，种群初始化后种群 P_a 便采用约束松弛；RBPF4分组判断条件 r_f 设为0.2；RBPF5分组判断条件 r_f 设为0.4。

表8为所提RBPF算法与其变体在所有测试问题上的对比结果。

由表8可知：①RBPF显著优于RBPF1与RBPF2。原因在于这2个变体没有根据学习结果选择更合适搜索策略以应对具有不同特点的问题。

由此可证明，RBPF能够很好地按照算法设想，处理各种类型约束多目标优化问题，同时也证明了本文所提分类方法的准确性以及搜索阶段根据问题分类调整搜索策略的有效性。②RBPF显著优于RBPF3，证明了算法在进化开始，对种群 P_a 在学习阶段忽略所有约束，使其对UPF与CPF关系学习的有效性。③RBPF在3个测试集上的统计结果与RBPF4和RBPF5十分相似。 r_f 的大小影响的仅是对种类III问题的进一步分类。实验中可以发现，属于种类III问题的 r_f 值基本上是大于0.4的，而属于种类III'问题的 r_f 值基本上是小于0.2的。因此， r_f 在0.2~0.4中取值的敏感性并不强，这一点在实验数据上也得到了印证。综上所述，最终将分组判断条件 r_f 设为中间值0.3。至此，算法各个模块的有效性得到了充分证明。

表 6 不同算法在 DTLZ 测试集上的 HV 结果
Table 6 HV results of different algorithms on DTLZ benchmark suit

问题	BiCo	C-TAEA	CMOEA-MS	PPS	ToP	TiGE-2	URCMO	RBPF
C1_DTLZ1	0.836 (0.004 63)-	0.837 (0.001 19)-	0.833 (0.005 24)-	0.818 (0.003 52)-	NaN (NaN)	0.245 (0.193 00)-	0.828 (0.005 99)-	0.839 (0.003 22)
C1_DTLZ3	0.437 00 (0.212 00)-	0.503 00 (0.123 00)-	0.528 00 (0.111 00)=	0.325 00 (0.252 00)-	0.346 00 (0.212 00)-	0.001 16 (0.006 38)-	0.251 00 (0.255 00)-	0.558 00 (0.002 43)
C2_DTLZ2	0.516 (0.001 46)=	0.507 (0.001 78)-	0.516 (0.001 30)-	0.499 (0.002 80)-	0.468 (0.012 50)-	0.461 (0.008 77)-	0.509 (0.003 83)-	0.517 (0.001 60)
C3_DTLZ4	0.788 (0.001 27)-	0.785 (0.001 56)-	0.546 (0.075 80)-	0.756 (0.029 30)-	0.755 (0.004 96)-	0.766 (0.002 89)-	0.781 (0.001 96)-	0.780 (0.001 37)
DC1_DTLZ1	0.631 0 (0.002 02)-	0.627 0 (0.001 11)-	0.619 0 (0.019 90)-	0.562 0 (0.073 70)-	0.577 0 (0.025 10)-	0.090 7 (0.078 20)-	0.628 0 (0.002 20)-	0.632 0 (0.001 44)
DC1_DTLZ3	0.471 (0.002 10)=	0.463 (0.002 25)-	0.433 (0.056 80)=	0.264 (0.156 00)-	0.159 (0.183 00)-	0 (0)-	0.446 (0.051 30)-	0.472 (0.002 21)
DC2_DTLZ1	0.549 (0.200 000)-	0.838 (0.000 355)+	0.839 (0.001 120)+	0.784 (0.093 800)-	NaN (NaN)	0.239 (0.149 000)-	0.807 (0.070 3000)-	0.818 (0.092 000)
DC2_DTLZ3	0.013 30 (0.000 733)=	0.201 00 (0.259 000)=	0.122 00 (0.221 000)-	0.201 00 (0.251 000)-	NaN (NaN)	0 (0)=	0.009 18 (0.001 740)-	0.406 00 (0.209 000)
DC3_DTLZ1	0.323 00 (0.215 00)-	0.523 00 (0.003 15)+	0.409 00 (0.035 00)-	0.123 00 (0.169 00)-	0.027 50 (0.084 40)-	0.006 03 (0.020 20)-	0.515 00 (0.097 30)-	0.521 00 (0.079 10)
DC3_DTLZ3	0 (0)-	0.361 0 (0.001 67)+	0.226 0 (0.167 00)+	0.020 8 (0.065 10)=	0 (0)-	0 (0)-	0.048 5 (0.114 00)=	0.048 4 (0.126 00)
+/-/=	0/7/3	3/6/1	2/6/2	0/9/1	0/7/0	0/9/1	0/9/1	

注：“+”“-”“=” 分别表示其他算法优于、劣于和等于本文算法。

表 7 不同算法的 Friedman 秩和检验结果
Table 7 Friedman test results of different algorithms

算法	IGD+ 排序值	p-值	HV 排序值	p-值
BiCo	5.333 0	0	5.560 6	0
C-TAEA	3.909 1	0.000 072	3.697 0	0.000 036
CMOEA-MS	4.484 8	0.000 001	4.424 2	0.000 002
PPS	4.242 4	0.000 006	4.121 2	0.000 019
ToP	7.212 1	0	7.242 4	0
TiGE-2	5.818 2	0	5.924 2	0
URCMO	3.484 8	0.001 089	3.484 8	0.001 299
RBPF	1.515 2		1.545 5	

表 8 RBPF 和其变体在所有测试集上的对比结果
Table 8 Results of RBPF and its variants on all suites

变体	IGD+(+/-/=)	HV(+/-/=)
RBPF1	5/17/11	4/19/10
RBPF2	2/17/14	3/17/13
RBPF3	2/22/9	1/23/9
RBPF4	0/1/32	0/1/32
RBPF5	0/2/31	0/1/32

注：“+”“-”“=” 分别表示其他算法优于、劣于和等于本文算法。

表 9 3 个实际约束多目标优化问题的参数
Table 9 Parameters in three real-world CMOPs

问题	目标数量	决策变量 数量	约束数量
Bulk Carrier Design ^[19]	3	6	9
Process Flow Sheeting ^[20]	2	3	3
Process Synthesis ^[21]	2	2	2

3.4 实际约束多目标优化问题实验结果

为进一步证明所提 RBPF 算法的性能，在 3 个实际约束多目标优化问题上进行了实验，如表 9~10 所示，进一步证明了所提算法的有效性。

表 10 不同算法在 3 个实际约束多目标优化问题上的 HV 结果
Table 10 HV results of different algorithms on three real-world CMOPs

问题	BiCo	C-TAEA	CMOEA-MS	PPS	ToP	TiGE-2	URCMO	RBPF
Bulk Carrier Design	2.5879e-1 (1.91e-2) -	2.5172e-1 (1.53e-2) -	2.4285e-1 (6.15e-2) -	2.1847e-1 (5.19e-2) -	2.4058e-1 (4.68e-2) -	2.3525e-1 (5.27e-2) -	2.9038e-1 (3.15e-2) =	3.0239e-1 (4.37e-2)
Process Flow Sheeting	1.5709e+11 (6.74e+11)+	2.8933e+3 (1.07e+4) -	3.4969e+12 (1.91e+13) =	1.0634e+7 (2.18e+7) -	1.9473e+9 (7.15e+9) -	4.1662e+4 (2.04e+5) -	8.3815e+9 (2.61e+10) -	4.2693e+10 (1.70e+11)
Process Synthesis	2.4152e-1 (1.07e-5) -	2.4114e-1 (1.81e-4) -	2.3160e-1 (1.18e-5) -	2.4121e-1 (5.36e-5) -	2.4120e-1 (7.65e-5) -	1.8858e-1 (1.87e-2) -	2.4151e-1 (1.38e-5) -	2.4154e-1 (1.13e-5)
+/-/=	1/2/0	0/3/0	0/2/1	0/3/0	0/3/0	0/3/0	0/2/1	

注：“+”“-”“=” 分别表示其他算法优于、劣于和等于本文算法。

4 结论

本文针对约束多目标优化问题, 从探索和利用 UPF 与 CPF 之间的关系出发, 提出了一种基于帕累托前沿关系的约束多目标优化算法。该优化算法, 在学习阶段确定给定问题的 UPF 与 CPF 之间的关系。学习阶段结束后, 通过本文设计的分类方法, 对两者关系进一步分类。在搜索阶段, 根据分类结果, 为种群配备特定的搜索策略, 旨在为搜索 CPF 提供更有价值的信息。通过在 33 个具有不同特点的基准函数和 3 个实际约束多目标优化问题上的实验, 验证了本文所提算法能够很好地利用帕累托前沿关系, 解决具有不同特点的约束多目标优化问题, 并且相较其他算法具有显著的性能优势。此外, 通过消融实验进一步验证了本文各个模块的有效性和必要性。但所提算法仍有一些不足之处。例如, 当处理 LIR-CMOPs 中的一些特别复杂的约束多目标优化问题时, 由于辅助种群未能充分搜索 UPF, 所提的分类方法可能误判给定问题的类型, 继而采用不合适的搜索策略造成算法的搜索能力下降。未来还需要在以下方面进行改进: 设计基于关系分类的预选策略以节省评价次数。加强算法搜索能力, 从而实现更为精准的分类。从决策空间和目标空间提取更多特征, 为不同类型问题制定更加合适的搜索策略。未来还需尝试将本文所提算法用于更多仿真实验中以解决实际约束多目标优化问题所带来的难点^[22-23]。

参考文献:

- [1] Nicolas Jozefowicz, Frédéric Semet, El-Ghazali Talbi. Multi-objective Vehicle Routing Problems[J]. *European Journal of Operational Research*, 2008, 189(2): 293-309.
- [2] Saravanan R, Ramabalan S, N Godwin Raja Ebenezer, et al. Evolutionary Multi Criteria Design Optimization of Robot Grippers[J]. *Applied Soft Computing*, 2009, 9(1): 159-172.
- [3] Mala-Jetmarova H, Nargiz Sultanova, Savic D. Lost in Optimisation of Water Distribution Systems? A Literature Review of System Operation[J]. *Environmental Modelling & Software*, 2017, 93: 209-254.
- [4] Liu Zhizhong, Wang Yong. Handling Constrained Multiobjective Optimization Problems with Constraints in Both the Decision and Objective Spaces[J]. *IEEE Transactions on Evolutionary Computation*, 2019, 23(5): 870-884.
- [5] Fan Zhun, Li Wenji, Cai Xinye, et al. Push and Pull Search for Solving Constrained Multi-objective Optimization Problems[J]. *Swarm and Evolutionary Computation*(S2210-6502), 2019, 44:665-679.
- [6] Li Ke, Chen Renzhi, Fu Guangtao, et al. Two-archive Evolutionary Algorithm for Constrained Multiobjective Optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2019, 23(2): 303-315.
- [7] Ma Zhongwei, Wang Yong, Song Wu. A New Fitness Function with Two Rankings for Evolutionary Constrained Multiobjective Optimization[J]. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021, 51(8): 5005-5016.
- [8] Zhou Yalan, Zhu Min, Wang Jiahai, et al. Tri-goal Evolution Framework for Constrained Many-objective Optimization[J]. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020, 50(8): 3086-3099.
- [9] Sosa Hernández, Víctor Adrián, Schütze Oliver, et al. The Set-based Hypervolume Newton Method for Bi-

- objective Optimization[J]. IEEE Transactions on Cybernetics, 2020, 50(5): 2186-2196.
- [10] Tian Ye, Zhang Tao, Xiao Jianhua, et al. A Coevolutionary Framework for Constrained Multiobjective Optimization Problems[J]. IEEE Transactions on Evolutionary Computation, 2021, 25(1): 102-116.
- [11] Liang Jing, Qiao Kangjia, Yu Kunjie, et al. Utilizing the Relationship Between Unconstrained and Constrained Pareto Fronts for Constrained Multiobjective Optimization[J]. IEEE Transactions on Cybernetics, 2023, 53(6): 3873-3886.
- [12] Eckart Zitzler, Marco Laumanns, Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm: TIK-report 103[R]. Zurich: ETH Zurich, 2001,103.
- [13] Tian Ye, Zhang Yajie, Su Yansen, et al. Balancing Objective Optimization and Constraint Satisfaction in Constrained Evolutionary Multiobjective Optimization[J]. IEEE Transactions on Cybernetics, 2022, 52(9): 9559-9572.
- [14] Liu Zhizhong, Wang Bingchuan, Tang Ke. Handling Constrained Multiobjective Optimization Problems via Bidirectional Coevolution[J]. IEEE Transactions on Cybernetics, 2022, 52(10): 10163-10176.
- [15] Tian Ye, Cheng Ran, Zhang Xingyi, et al. PlatEMO: A MATLAB Platform for Evolutionary Multi-objective Optimization[J]. IEEE Computational Intelligence Magazine, 2017, 12(4): 73-87.
- [16] Deb Kalyanmoy, Jain Himanshu. An Evolutionary Many-objective Optimization Algorithm Using Reference-point-based Nondominated Sorting Approach, Part I: Solving Problems with Box Constraints[J]. IEEE Transactions on Evolutionary Computation, 2014, 18(4):577-601.
- [17] Fan Zhun, Li Wenji, Cai Xinye, et al. Difficulty Adjustable and Scalable Constrained Multiobjective Test Problem Toolkit[J]. Evolutionary Computation, 2020, 28(3): 339-378.
- [18] Fan Zhun, Li Wenji, Cai Xinye, et al. An Improved Epsilon Constraint-handling Method in MOEA/D for CMOPs with Large Infeasible Regions[J]. Soft Computing, 2019, 23(23): 12491-12510.
- [19] Parsons M G, Scott R L. Formulation of Multicriterion Design Optimization Problems for Solution with Scalar Numerical Optimization Methods[J]. Journal of Ship Research, 2004, 48(1): 61-76.
- [20] Floudas C A. Nonlinear and Mixed-integer Optimization: Fundamentals and Applications[M]. Oxford: Oxford University Press, 1995.
- [21] Kocis G R, Grossmann I E. A Modelling and Decomposition Strategy for the Minlp Optimization of Process Flowsheets[J]. Computers & Chemical Engineering, 1989, 13(7): 797-819.
- [22] 张进峰, 杨涛宁, 马伟皓. 基于多目标粒子群算法的船舶航速优化[J]. 系统仿真学报, 2019, 31(4): 787-794.
Zhang Jinfeng, Yang Taoning, Ma Weihao. Ship Speed Optimization Based on Multi-objective Particle Swarm Algorithm[J]. Journal of System Simulation, 2019, 31(4): 787-794.
- [23] 闫秀英, 党苗苗. 基于改进多目标粒子群算法的家庭用电时段优化[J]. 系统仿真学报, 2022, 34(1): 70-78.
Yan Xiuying, Dang Miaomiao. Optimization of Household Electricity Consumption Period Based on Improved Multi-objective Particle Swarm Optimization[J]. Journal of System Simulation, 2022, 34(1): 70-78.