

控制与决策

Control and Decision

一种求解约束多目标问题的协作进化算法

王蕊, 顾清华

引用本文:

王蕊, 顾清华. 一种求解约束多目标问题的协作进化算法[J]. 控制与决策, 2021, 36(11): 2656–2664.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2020.0791>

您可能感兴趣的其他文章

Articles you may be interested in

基于R2指标和目标空间分解的高维多目标粒子群优化算法

R2 indicator and objective space partition based many-objective particle swarm optimizer

控制与决策. 2021, 36(9): 2085–2094 <https://doi.org/10.13195/j.kzyjc.2020.0113>

基于正态云模型的状态转移算法求解多目标柔性作业车间调度问题

State transition algorithm based on normal cloud model for solving multi-objective flexible job shop scheduling problem

控制与决策. 2021, 36(5): 1181–1190 <https://doi.org/10.13195/j.kzyjc.2019.1233>

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>

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>

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>

一种求解约束多目标问题的协作进化算法

王蕊¹, 顾清华^{1,2†}

(1. 西安建筑科技大学 管理学院, 西安 710055; 2. 西安建筑科技大学 资源工程学院, 西安 710055)

摘要: 针对约束多目标进化算法求解约束多目标问题时难以平衡收敛性、多样性和可行性的问题, 提出一种协作进化算法 (ConMOEA). 将自适应形状估计进化算法 (AGE-MOEA) 和非支配排序遗传算法 (NSGA-II) 优势融合, 采用 Deb 约束支配原则非支配排序组合种群实现个体优选, 在临界层中根据最大拥挤距离或生存值选择所需个体, 最终形成新种群, 实现种群快速接近 Pareto 前沿并具有良好分布性. 为验证所提出算法的性能, 对近期提出的一组 DOC 基准函数进行仿真计算, 采用反世代距离 (IGD) 和超体积 (HV) 两个通用评价指标, 与 NSGA-II-CDP、C-TAEA、PPS、ToP、A-NSGA-III、AGE-MOEA 约束多目标算法进行比较分析, 实验结果证明 ConMOEA 具有更优的收敛性和多样性.

关键词: 约束多目标优化; Deb 约束支配; AGE-MOEA; NSGA-II; 收敛性; 多样性

中图分类号: TP273 文献标志码: A

DOI: 10.13195/j.kzyjc.2020.0791

开放科学(资源服务)标识码(OSID):



引用格式: 王蕊, 顾清华. 一种求解约束多目标问题的协作进化算法[J]. 控制与决策, 2021, 36(11): 2656-2664.

A collaborative evolutionary algorithm for solving constrained multi-objective problems

WANG Rui¹, GU Qing-hua^{1,2†}

(1. School of Management, Xi'an University of Architecture and Technology, Xi'an 710055, China; 2. School of Resources Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China)

Abstract: The balance of convergence, diversity and feasibility is a difficulty for the constrained multi-objective evolutionary algorithms. Thus, a collaborative constrained multi-objective algorithm (ConMOEA) is proposed, which integrates the advantages of the adaptive geometry estimation based MOEA (AGE-MOEA) and the non-dominated sorting genetic algorithm (NSGA II). Firstly, the Deb constraint dominance is applied to sort the combined population. Then the individuals in critical layer are selected according to the maximum crowding distance or individual survival score. Finally, a new population is formed that can fast approach the Pareto front and has good distribution. The effectiveness of the proposed algorithm is validated by comparing with NSGA-II-CDP, C-TAEA, PPS, ToP, A-NSGA-III, AGE-MOEA on the DOC test suit. And the performance of algorithms is evaluated by the inverted generational distance (IGD) and hypervolume (HV). The simulation results show that the ConMOEA has better convergence and diversity.

Keywords: constrained multi-objective optimization; Deb constraint dominance; AGE-MOEA; NSGA-II; convergence; diversity

0 引言

多目标优化近年来赢得广泛关注, 在多目标优化问题求解中, 相比无约束算法, 约束多目标进化算法 (constrained multi-objective evolutionary algorithms, CMOEAs) 相关研究较少. 实际优化问题通常具有各种约束, 在科学与工程领域, 大多数问题可数学表达为约束多目标优化问题, 诸如网络服务器位置分配^[1], 临近空间通信部署优化^[2], 复杂物流车辆

调度^[3]等, 因而对 CMOEAs 的研究具有重要理论和实践意义.

CMOEAs 是公认的求解约束多目标优化问题的有效方法, 约束处理技术和多目标进化算法是其关键组成部分. 对于约束处理技术主要分为 4 种:

1) 基于排序的约束处理技术, 如经典的 Deb 约束支配^[4], 简单且不需要额外参数; epsilon 约束处理技术^[5]需要合理控制保证约束松弛的参数; 模糊排序方

收稿日期: 2020-06-16; 修回日期: 2020-09-21.

基金项目: 国家自然科学基金项目 (51774228, 51864046); 陕西省自然科学基金杰青项目 (2020JC-44); 中国博士后科学基金项目 (2019M662505).

责任编辑: 巩敦卫.

†通讯作者. E-mail: qinghuagu@126.com.

法^[6]通过概率参数确定比较目标或者约束。

2) 基于罚函数的约束处理技术, 罚函数法^[7]构造简单但惩罚系数难以确定。为解决此问题, 文献[8]通过自适应的适应度值公式衡量不可行解, 从而进行惩罚; 文献[9]提出利用遗传算法的自适应罚函数法, 引进具有低目标值和约束违反度的解参与进化过程; 文献[10]将自适应惩罚函数作为个体的搜索偏好进化种群。基于罚函数的算法因参数选取对解在可行域内的搜索具有很大影响, 从而难以保证解的质量。

3) 将约束转换为额外目标^[11], 如不可行驱动进化算法 (IDEA)^[12], 但是目标维数变高增加了求解难度, 这种算法旨在维持约束和目标的平衡, 但难以保证解的可行性。

4) 其他约束处理技术, 如多存档集技术^[13], 近来文献[14]提出了两个存档集的 C-TAEA 算法, 该算法通常从两个存档集中分别选取父代个体来产生子代, 因而产生的多数子代位于两个存档集之间, 导致子代不具备较好的可行性和收敛性, 使得算法无法找到一组位于可行域内且均匀收敛的解, 并且算法收敛速度较慢。

当前的 CMOEAs 根据多目标进化算法可分为基于支配和基于分解两类。基于支配的 CMOEAs 诸如 NSGA-II-CDP^[4]、A-NSGA-III^[15]、C-VaEA^[16] 等算法, 其中 NSGA-II-CDP^[4] 作为基于支配的 CMOEAs 最典型代表, 结合 Deb 约束支配将 NSGA-II 算法拓展到解决约束多目标优化问题。但这些算法使用 Deb 约束支配强调可行性, 当具有复杂约束时, 获得的解难以收敛和分布在多处可行区域上。基于分解的 CMOEAs 因基于分解的多目标进化算法容易直接嵌套约束处理技术而被用于处理约束多目标优化问题。文献[17]是添加了一种约束技术的基于分解 MOEA/D 算法的拓展版, 其约束技术中的约束阈值通过约束类型、可行空间大小和搜索输出自适应调整, 从而增加了种群的选择压力。代表性算法还有 MOEA/D-ASR^[18], 使用自适应模糊排序机制改善算法的搜索能力, 但是其基于约束违反度的排序函数过于强调可行性, 容易忽略收敛性和多样性; CMOEAs/D-DE-CDP^[19] 需要依靠具有少量可行解的初始种群保证最终的求解质量; C-MOEAs/DD^[20] 通过保留孤立区域的解使算法跳出局部可行域, 但在产生新解过程中未充分利用分布良好的不可行解, 因而不利于种群收敛性能。总之, 过分强调可行性则可能导致解偏离性能良好的搜索区域, 过分强调解的质量则可能导致解偏离可行区域。综上分析, 在求解约束多目标问题时, 以上算法面临着难以有效平衡收敛性、多样性和可行性的问题。

为提高约束算法收敛性、多样性和可行性的平衡能力, 本文组合了在多目标优化中具有高效收敛性的 NSGA-II^[4] 算法和能够快速估计 Pareto 前沿形状的 AGE-MOEA^[21] 算法, 同时结合经典 Deb 约束支配提出新的约束算法——协作进化算法 (ConMOEA), 用于求解约束多目标问题。

1 研究基础

1.1 一般约束多目标问题数学模型

$$\begin{aligned} \min F(x) &= (f_1(x), \dots, f_M(x))^T. \quad (1) \\ \text{s.t. } g_j(x) &\geq a_j, \quad j = 1, 2, \dots, q; \\ h_j(x) &= b_j, \quad j = q + 1, \dots, l; \\ x &\in \Omega; \\ x &= (x_1, \dots, x_n)^T. \end{aligned}$$

其中: M 为目标数目, Ω 为决策空间, x 为一个候选解, n 为决策变量数目, $g_j(x)$ 为第 j 个不等式约束, $h_j(x)$ 为第 j 个等式约束。

1.2 Deb 约束支配

Deb 约束支配原则^[4]: 存在解 x, y , 解 x 约束支配解 y 满足下列条件: 1) 解 x, y 均为不可行解, $CV(x) < CV(y)$; 2) 解 x 为可行解, 解 y 为不可行解; 3) 解 x, y 均为可行解, $x \prec y$ 。

解 x 在第 j 个约束的约束违反度^[14] 为

$$c_j(x) = \begin{cases} \left\langle \frac{g_j(x)}{a_j} - 1 \right\rangle, & j = 1, 2, \dots, q; \\ \left\langle \left\| \frac{h_j(x)}{b_j} - 1 \right\| - \epsilon \right\rangle, & j = q + 1, \dots, l. \end{cases} \quad (2)$$

其中: ϵ 是一个非常小的容忍度 (如 $\epsilon = 10^{-6}$), 当 $\alpha \geq 0$ 时, $\langle \alpha \rangle$ 返回 0, 否则返回 $-\alpha$ 。解 x 的整体约束违反度为

$$CV(x) = \sum_{j=1}^l c_j(x). \quad (3)$$

当 x 为可行解时, $CV(x) = 0$; 否则 x 为不可行解。

1.3 差分进化算子

差分进化计算公式^[22] 为

$$y = \begin{cases} x_{r1} + F(x_{r2} - x_{r3}), & p < CR; \\ x_{r1}, & \text{otherwise.} \end{cases} \quad (4)$$

其中: F 和 CR 为两个控制参数; 子代 $y = (y_1, \dots, y_D)$, D 为决策变量数目; x_{r1}, x_{r2}, x_{r3} 为 3 个不同父代个体; p 为 $[0, 1]$ 的随机数。

1.4 AGE-MOEA 算法基本思想

AGE-MOEA 算法原理如下: 首先, 随机产生 N 个个体作为初始种群 P_t ; 其次, 根据个体所在的非支配层和生存分数选取父代进入匹配池, 通过交叉变异

产生 N 个子代构成种群 Q_t ; 然后, 合并初始种群 P_t 和子代种群 Q_t 产生合并种群 R_t ; 最后, 从 R_t 中选出 N 个较优个体存储到种群 P_{t+1} 中. N 个较优个体的选取: 根据目标值非支配排序 R_t , 将 R_t 中的个体划分为不同层级 (F_1, F_2, \dots, F_l) , 当 $\{F_1 \cup F_2 \cup \dots \cup F_{l-1}\}$ 的规模小于 N 且 $\{F_1 \cup F_2 \cup \dots \cup F_l\}$ 的规模大于 N 时, 保留 $\{F_1 \cup F_2 \cup \dots \cup F_{l-1}\}$ 中的所有个体, 存储到下一代种群 P_{t+1} ; 评估不同层级中解的收敛性和多样性, 形成每个个体的生存分数, 选取 F_l 层级中生存分数最大的 $N - |P_{t+1}|$ 个个体存储于下一代种群 P_{t+1} .

2 ConMOEA 算法

本文提出的 ConMOEA 算法融合 NSGA-II 及 AGE-MOEA 算法的优势和必要性: 由于约束条件造成的可行域不连通, Pareto 前沿变得复杂, 增加了求解难度. AGE-MOEA 的融合使提出的 ConMOEA 算法能够估计不同 Pareto 前沿的形状, 对种群进化具有指导作用, 使解尽可能分布在整个 Pareto 前沿上. 在保证良好求解结果的前提下, 提升求解速度也是学者致力于约束算法研究的方向, 当前的很多约束算法并不高效, 融合 NSGA-II 算法能够继承其求解速度快和收敛好的优势.

2.1 ConMOEA 算法流程

step 1: 初始种群 P_t , 种群大小为 N , 迭代次数 $t = 0$, 最大迭代次数 \max_t , 随机产生 N 个个体形成初始种群 P_t .

step 2: 通过差分进化算子产生子代 Q_t , 合并种群 $R_t = P_t \cup Q_t$.

step 3: 根据 Deb 约束支配原则对 R_t 进行非支配排序, 形成 F_1, \dots, F_l 层.

step 4: 1) 当前代数 $t < \frac{1}{3} \max_t$, 执行下列操作:

① 计算个体拥挤度距离;

② 根据非支配排序结果将 F_1, \dots, F_{l-1} 按顺序存储, 临界层 (F_l 层) 中选择具有较大拥挤度距离的个体, 直到选出 N 个最优个体, 构成新种群 P_{t+1} , $t = t + 1$.

2) 当前代数 $t \geq \frac{1}{3} \max_t$, 执行下列操作:

① 根据归一化后的目标值计算每层个体生存值;

② 根据非支配排序结果将 F_1, \dots, F_{l-1} 按顺序存储, 临界层 (F_l 层) 中选择具有较大生存值的个体, 直到选出 N 个最优个体, 构成新种群 P_{t+1} , $t = t + 1$.

step 5: 当迭代次数 t 达到最大迭代次数 \max_t 时

结束运算, 否则返回 step 2.

在 ConMOEA 算法中, 终止代数的 1/3 被作为种群进化中使用两个不同策略的分割点, 设计动机如下: 整个进化过程分为进化前期、中期、后期 3 段, 每一个进化段长度为 1/3 总迭代数. 由于采用 Deb 约束支配准则强调解的可行性, 容易忽略收敛性和多样性, 本研究仅在进化前期采用 NSGA-II 算法, 保证快速收敛. 进化的中后期采用 AGE-MOEA 算法, 使用生存值算子在临界层选解利用了个体的收敛性和多样性信息, 有助于探索未开发区域, 使算法尽可能获得整个 Pareto 前沿.

2.2 计算拥挤距离

拥挤距离计算公式^[4]为

$$P[i]_{\text{distance}} = \sum_{k=1}^m \frac{P[i+1]f_k - P[i-1]f_k}{f_k^{\max} - f_k^{\min}}. \quad (5)$$

其中: $P[i]_{\text{distance}}$ 为个体 i 的拥挤距离, $P[i+1]f_k$ 为个体 $i+1$ 第 k 维目标值, $P[i-1]f_k$ 为个体 $i-1$ 第 k 维目标值, m 为目标维数, f_k^{\max} 和 f_k^{\min} 分别为该目标的最大值和最小值.

2.3 归一化目标值

目标值归一化分首层和其他层个体归一化两部分, 本文采用与 NSGA-III^[23] 算法相同的归一化方式. 不同于 AGE-MOEA 算法, 在归一化首层个体时, ConMOEA 算法使用的极值点 Z^{\max} 是在当代的整个种群中寻找而不是首层非支配层, 理想点 Z^{\min} 是迄今为止找到的最小目标值对应的点, 其他层个体归一化使用的参数超平面截距 a_i 为首层个体归一化时计算出的值. 当超平面无法构建或出现错误时, 超平面截距 a_i 为种群中最大目标值与理想点目标值之差.

2.4 计算每层个体生存值

生存值的计算分为首层非支配层个体和其他非支配层个体生存值计算两部分. 在计算生存值时, L_p 范数中 p 是一个很重要的自适应计算参数, 有

$$p = \frac{\log(M)}{\log(M) - \log\left(\sum_{i=1}^M C_i\right)}. \quad (6)$$

其中: M 为目标数目; C 为首层非支配层的中心点, 有

$$C = \arg \min \text{dist}^\perp(f(S)^T, \beta), \quad \forall S \in F_1, \quad (7)$$

指的是 F_1 层中与向量 $\beta = (1)_{1 \times M}$ 具有最小垂直距离的点, 距离的计算公式为

$$\text{dist} = \left\| f(S)^T - \frac{f(S)^T \cdot \beta^T}{\|\beta\|} * \beta \right\|, \quad (8)$$

$f(S) = (f^1(S), \dots, f^M(S))^T$ 表示个体 S 的目标值, “ \cdot ” 表示一般矩阵乘积, “ $*$ ” 表示各元素乘积.

个体生存值计算步骤如下:

step 1: 找到 F_1 层中的极值点并存储到 E 中, 将 F_1 层中的其余解存储到 Q 中, 极值点生存值 $\text{survivalscore} = \infty$;

step 2: 计算解 $S \in Q$ 的近似性, $\text{prox}(S) = \|f(S)\|_p$;

step 3: 计算解 $S_1, S_2 \in Q$ 之间的距离, ($S_1 \neq S_2$), $\text{distance}[S_1, S_2] = \|f(S_1) - f(S_2)\|_p$;

step 4: 计算解 $S \in Q$ 的多样性, $\text{div}(S) = \min \text{distance}[S, T], T = 2, \dots, \min \text{distance}[S, T]$ 表示其他解与解 S 具有的最小距离和次小距离之和;

step 5: 计算解 $S \in Q$ 的生存值 $\text{survivalscore} =$

$\text{div}(S)/\text{prox}(S)$;

step 6: 计算其余非支配层中个体 $s \in F_2, \dots, F_l$ 的生存值 $\text{survivalscore} = 1/\|f(s) - Z^{\min}\|_p$.

3 实验仿真与分析

3.1 测试函数与评价指标

本实验研究了 ConMOEA 在 DOC 系列测试问题上的性能. 根据文献 [24] 在表 1 中描述了测试函数的信息, DOC 测试问题是近来提出的同时具有决策约束和目标约束的测试函数, 更加接近于模拟真实应用. 为了比较不同算法性能, 选择反世代距离 (IGD)^[25] 和超体积 (HV)^[26] 两个广泛使用的评价指标, 其中 IGD 指标值越小表示结果越好, 而 HV 指标值越大表示结果越好.

表 1 DOC 测试函数信息

测试问题	目标数目	决策变量数目	目标约束数目	决策约束数目	不等式约束数目	等式约束数目	Pareto 前沿特征	可行区域特征
DOC-1	2	6	1	6	7	0	凹连续	非线性
DOC-2	2	16	2	5	7	0	凸不连续	非常小非线性
DOC-3	2	10	4	6	6	4	凹不连续多模态	非常小非线性多模态
DOC-4	2	8	2	4	6	0	线性不连续	非线性小
DOC-5	2	8	3	6	4	5	不连续多模态	非常小非线性
DOC-6	2	11	2	8	10	0	混合多模态	非常小非线性
DOC-7	2	11	3	3	3	3	混合多模态	非常小多模态
DOC-8	3	10	1	6	7	0	线性不连续	非常小非线性
DOC-9	3	11	1	13	14	0	退化多模态	非常小非线性多模态

3.2 对比算法

本文算法基于约束 NSGA-II 算法进行了两部分创新: 一是在匹配池机制中引入了差分进化算子产生新解; 二是进化中后期在临界层中根据个体生存值选解. 为验证各策略的有效性, 将约束 NSGA-II 算法 (NSGA-II-CDP)、加入差分进化算子的约束 NSGA-II 算法 (NSGA-II-CDPDE) 与 ConMOEA 算法进行对比研究.

为进一步验证所提出 ConMOEA 算法的性能, 选取 C-TAEA^[14]、PPS^[27]、ToP^[24]、A-NSGA-III^[15]、AGE-MOEA^[21] 6 个著名的约束多目标优化算法进行对比实验研究. 其中 AGE-MOEA 是求解无约束多目标问题的算法, 由于本文算法融合了该算法, 将引入 Deb 约束支配的 AGE-MOEA 作为对比算法之一.

3.3 实验配置与参数设置

实验环境为 Inter Core (TM) i7-8550UCPU, 内存 16 GB, Window 10 操作系统, Matlab R2019b 版本. 实验操作平台为通用的基于 Matlab 的多目标优化工具 PlatEMO^[28], 版本 2.5.

各算法种群大小和迭代次数设置: 参照文献 [24], DOC-1 ~ DOC-7 测试函数的种群大小 N 设置为

100, 评估次数 200 000; DOC-8、DOC-9 测试函数的种群大小 N 设置为 300, 评估次数 400 000.

PPS^[27]: 从邻域选择个体的概率 $\text{delta} = 0.9$; 每个子代所替换解的最大数目 $n_r = 2$; 代数控制参数 $T_c = 800$; 控制可行域和不可行域搜索参数 $\alpha = 0.95$; 在第 k 代可行解占比 $\text{rf}_k < \alpha$ 时, 减少约束松弛的速度控制参数 $\tau = 0.1$, 否则减少约束松弛的速度控制参数 $\text{cp} = 2$; 最后 l 代的 $l = 20$; 用户定义的搜索行为转换阈值 $\epsilon = 10^{-3}$.

ToP^[24]: 差分进化算子中的 $F = 1$, $\text{CR} = 0.5$, 划分搜索阶段的条件参数 $\delta = 0.2$.

NSGA-II-CDPDE、ConMOEA: 差分进化算子中的 $F = 1$, $\text{CR} = 0.5$, 变异概率 $\text{Pro}_M = 1$, 变异分布数 $\text{dis}_M = 20$.

A-NSGA-III^[15]、AGE-MOEA^[21]、NSGA-II-CDP^[4]: 遗传操作中的交叉概率 $p_c = 1$; 变异概率 $p_m = 1/n$, n 为决策变量维数; 交叉分布指标 $\eta_c = 20$; 变异分布指标 $\eta_m = 20$.

3.4 实验结果与分析

在实验结果部分, 每个算法的指标结果为独立运行 30 次得到的平均值和标准差, 加粗项为同一

测试函数中获得的最优值. 在结果比较中, 采用了 Wilcoxon 秩和检验^[29] 比较算法性能差异性, 置信度 95%, 其中“+”“-”“=”分别表示显著优、显著劣、无差异于 ConMOEA 算法.

3.4.1 算法策略有效性验证

表2和表3分别展示了 NSGA-II-CDP、NSGA-II-CDPDE、ConMOEA 算法在测试问题上的 IGD、

HV 指标值. 观察表2和表3, NSGA-II-CDP 和 NSGA-II-CDPDE 算法的指标结果显示了差分进化算子使原始算法性能显著提升. ConMOEA 算法和 NSGA-II-CDPDE 算法的比较结果彰显了生存值选解策略具有一定的优越性.

图1展示了 ConMOEA、NSGA-II-CDP、NSGA-II-CDPDE 算法分别在 DOC-2、DOC-5 测试问题上

表2 算法在 DOC 系列测试函数上 IGD 的平均值和标准差

测试函数	NSGA-II-CDP		NSGA-II-CDPDE		ConMOEA	
	平均值	标准差	平均值	标准差	平均值	标准差
DOC-1	2.689 7e+0-	2.51e+0	6.050 7e-3-	4.05e-4	5.656 4e-3	4.28e-4
DOC-2	NaN	NaN	5.029 5e-3=	2.55e-3	4.662 5e-3	1.92e-3
DOC-3	6.549 5e+2-	2.20e+2	4.445 6e+2=	4.24e+2	4.857 8e+2	4.45e+2
DOC-4	6.714 8e-1-	5.32e-1	1.694 7e-2=	2.79e-3	1.791 0e-2	3.23e-3
DOC-5	NaN	NaN	3.016 6e+1=	5.72e+1	5.406 4e-2	8.05e-2
DOC-6	1.909 3e+0-	1.92e+0	3.348 9e-3-	2.00e-4	2.945 6e-3	1.41e-4
DOC-7	6.345 8e+0-	2.54e+0	2.952 5e-3-	1.43e-4	2.582 9e-3	1.83e-4
DOC-8	5.785 4e+1-	5.65e+1	6.347 7e-2-	3.74e-3	3.895 2e-2	1.10e-3
DOC-9	1.131 7e-1=	8.00e-2	8.685 6e-2=	8.22e-3	8.554 0e-2	1.22e-2

表3 算法在 DOC 系列测试函数上 HV 的平均值和标准差

测试函数	NSGA-II-CDP		NSGA-II-CDPDE		ConMOEA	
	平均值	标准差	平均值	标准差	平均值	标准差
DOC-1	3.603 8e-2-	9.66e-2	3.444 0e-1-	5.21e-4	3.452 7e-1	6.01e-4
DOC-2	NaN	NaN	6.199 4e-1=	3.32e-3	6.203 6e-1	2.72e-3
DOC-3	0.000 0e+0-	0.00e+0	7.589 7e-2=	1.40e-1	7.634 9e-2	1.41e-1
DOC-4	1.285 4e-1-	1.29e-1	5.437 6e-1=	3.31e-3	5.427 9e-1	3.71e-3
DOC-5	NaN	NaN	3.672 4e-1=	2.12e-1	4.714 6e-1	4.32e-2
DOC-6	5.036 3e-2-	1.09e-1	5.393 3e-1-	5.34e-3	5.429 2e-1	4.71e-3
DOC-7	0.000 0e+0-	0.00e+0	5.509 6e-1=	6.11e-3	5.476 9e-1	8.47e-3
DOC-8	0.000 0e+0-	0.00e+0	7.970 8e-1-	4.08e-3	8.273 5e-1	1.06e-3
DOC-9	0.000 0e+0	0.00e+0	NaN	NaN	NaN	NaN

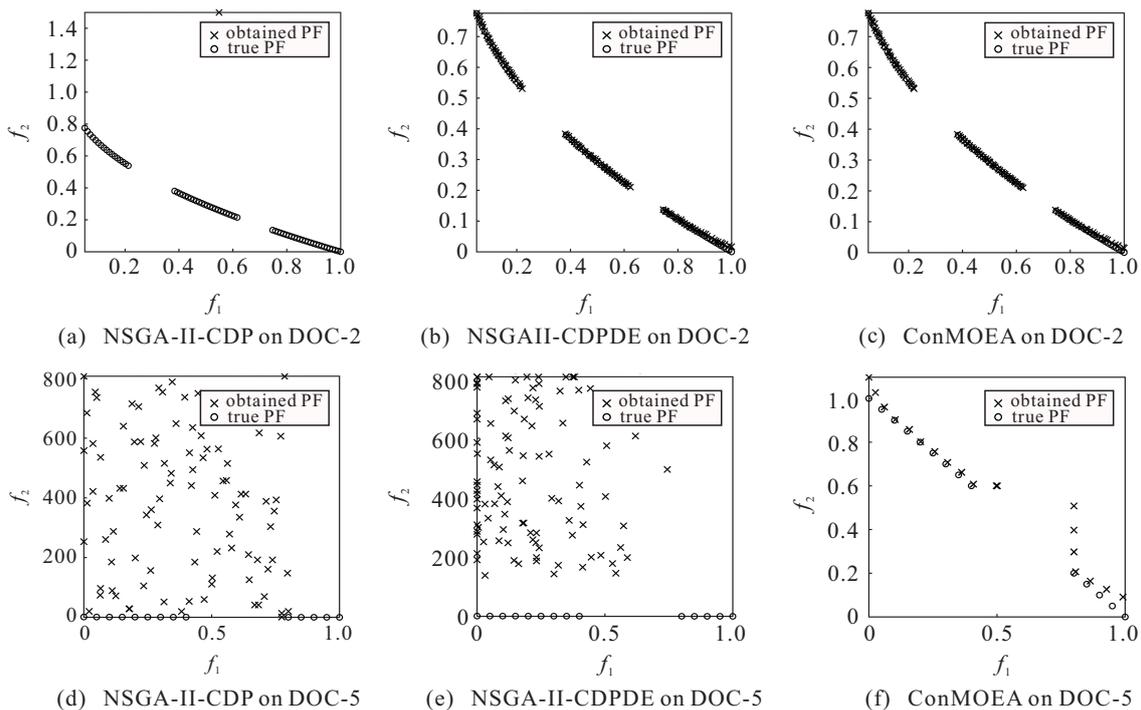


图1 3个算法在 DOC-2、DOC-5 测试问题的结果显示

的求解结果. 可以看出: NSGA-II-CDP 在 DOC-2 问题上表现较差, 仅获得了一个解, 并且距离真实前沿面较远. NSGA-II-CDPDE 和 ConMOEA 算法均能找到 DOC-2 问题的近似前沿, 虽然两个算法获得的解分布情况非常相近, 但指标值显示 ConMOEA 优于 NSGA-II-CDPDE. 对于 DOC-5 测试问题, NSGA-II-CDP 和 NSGA-II-CDPDE 的求解结果离散地分布在目标空间中, 而 ConMOEA 算法获得了近似 Pareto 前沿.

3.4.2 本文算法性能验证

表 4 给出了 C-TAEA、ToP、A-NSGA-III、AGE-MOEA、PPS 和 ConMOEA 算法的 IGD 指标平均值和标准差. 分析表 4 可得: ConMOEA 算法在 DOC-1、DOC-2、DOC-4~DOC-8 所求结果显著优于其他算法. DOC-2 测试问题的 Pareto 前沿凸且不连续, 并且可行区域非线性, 这对约束多目标算法是一个挑

战, ConMOEA 算法获得了较小的 IGD 值. DOC-5 测试问题的真实 Pareto 前沿具有不连续和多模态特征, 可行区域小且非线性, ConMOEA 比 ToP、PPS 得到了更好的结果. DOC-3 和 DOC-9 测试问题的可行区域具有非常小、非线性、多模态的共性, 而 ConMOEA 算法均未获得最好的 IGD 指标值, 由此可得 ConMOEA 在这类问题上算法性能不优.

表 5 给出了各算法的 HV 指标平均值和标准差, 从整体来看, 多数算法获得的指标值为 0.000 0e+0, 说明算法无法收敛到 Pareto 前沿上, ConMOEA 算法在 DOC-1~DOC-8 问题上均表现出优越性. 在 DOC-1 测试函数上, ToP 算法仅次于 ConMOEA. 综合分析 HV 值和 IGD 值得出, ConMOEA 能够在大部分 DOC 问题上获得较好收敛性和多样性的解.

表 6 展示了 6 个算法在 DOC 系列测试函数上的运行时间, 易观察到 ToP 算法在 DOC-1 测试问题上的

表 4 6 个算法在 DOC 系列测试函数上 IGD 的平均值和标准差

测试函数	C-TAEA		ToP		A-NSGA-III		AGE-MOEA		PPS		ConMOEA	
	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
DOC-1	4.587 9e+2-	2.05e+2	6.002 2e-3=	3.61e-4	2.782 8e+0-	2.37e+0	3.456 4e+0-	2.38e+0	6.942 8e-2-	5.05e-2	5.823 2e-3	3.31e-4
DOC-2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5.275 8e-3	2.72e-3
DOC-3	NaN	NaN	2.023 6e+2=	2.01e+2	6.268 4e+2-	2.19e+2	7.016 6e+2-	1.10e+2	1.992 8e+2=	1.83e+2	4.163 1e+2	4.41e+2
DOC-4	3.429 3e+2-	6.20e+2	1.127 3e-1-	7.40e-2	6.974 5e-1-	6.87e-1	8.2669e-1-	6.56e-1	3.0265e-1-	8.55e-2	1.821 5e-2	3.69e-3
DOC-5	NaN	NaN	9.6732e+1-	2.02e+2	NaN	NaN	NaN	NaN	7.114 5e+1-	1.55e+2	1.960 9e+1	4.78e+1
DOC-6	2.837 6e+1-	1.88e+1	4.389 8e+0-	1.45e+0	1.859 2e+0-	2.94e+0	1.353 8e+0-	1.30e+0	4.942 4e-1-	1.17e-1	3.002 1e-3	1.39e-4
DOC-7	NaN	NaN	8.933 5e-1-	4.81e-1	6.419 9e+0-	1.69e+0	5.515 9e+0-	2.07e+0	9.174 0e-1-	3.34e-1	2.515 9e-3	1.18e-4
DOC-8	4.515 8e+2-	1.42e+2	5.104 5e+1-	2.17e+1	3.829 4e+1-	3.09e+1	5.243 8e+1-	4.24e+1	6.473 0e+1-	1.56e+1	3.908 1e-2	1.15e-3
DOC-9	9.480 8e-1-	2.33e-1	1.987 7e-1-	3.29e-2	2.070 5e-2+	5.00e-2	9.531 5e-2=	1.04e-1	2.299 5e-1-	1.24e-2	8.536 6e-2	1.36e-2

表 5 6 个算法在 DOC 系列测试函数上 HV 的平均值和标准差

测试函数	C-TAEA		ToP		A-NSGA-III		AGE-MOEA		PPS		ConMOEA	
	平均值	标准差	平均值	标准差								
DOC-1	0.000 0e+0-	0.00e+0	3.443 9e-1-	4.29e-4	2.940 2e-2-	6.63e-2	3.713 2e-3-	1.49e-2	2.875 2e-1-	2.91e-2	3.453 4e-1	4.37e-4
DOC-2	NaN	NaN	6.195 6e-1	3.61e-3								
DOC-3	NaN	NaN	0.000 0e+0-	0.00e+0	0.000 0e+0-	0.00e+0	0.000 0e+0-	0.00e+0	7.425 7e-3-	2.55e-2	1.056 2e-1	1.52e-1
DOC-4	0.000 0e+0-	0.00e+0	4.315 9e-1-	8.47e-2	1.577 9e-1-	1.72e-1	1.282 5e-1-	1.70e-1	2.449 4e-1-	7.28e-2	5.423 3e-1	4.26e-3
DOC-5	NaN	NaN	1.4066e-1-	1.52e-1	NaN	NaN	NaN	NaN	2.289 2e-1-	1.64e-1	4.101 4e-1	1.79e-1
DOC-6	0.000 0e+0-	0.00e+0	0.000 0e+0-	0.00e+0	7.403 8e-2-	1.29e-1	3.945 7e-2-	9.60e-2	1.807 4e-1-	6.02e-2	5.422 6e-1	5.32e-3
DOC-7	NaN	NaN	2.742 4e-2-	5.10e-2	0.000 0e+0-	0.00e+0	0.000 0e+0-	0.00e+0	1.758 0e-2-	8.59e-2	5.490 5e-1	7.52e-3
DOC-8	0.000 0e+0-	0.00e+0	0.000 0e+0-	0.00e+0	0.000 0e+0-	0.00e+0	2.005 3e-5-	1.10e-4	0.000 0e+0-	0.00e+0	8.272 5e-1	1.35e-3
DOC-9	0.000 0e+0	0.00e+0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

表 6 6 个算法在 DOC 系列测试函数上运行时间的平均值和标准差

测试函数	C-TAEA		ToP		A-NSGA-III		AGE-MOEA		PPS		ConMOEA	
	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
DOC-1	2.666 4e+2-	2.11e+1	8.178 4e+0+	1.36e+0	1.492 7e+1-	1.68e+0	2.728 9e+1-	1.63e+0	6.767 5e+1-	5.79e+0	1.243 2e+1	9.29e-1
DOC-2	3.828 2e+2-	7.65e+1	1.214 9e+2-	1.83e+1	2.137 2e+1-	2.36e+0	2.7252e+1-	3.14e+0	9.8794e+1-	9.60e+0	1.592 5e+1	1.25e+0
DOC-3	2.818 6e+2-	3.34e+1	5.435 9e+1-	5.45e+0	1.698 5e+1+	1.62e+0	2.844 4e+1-	1.51e+0	6.852 9e+1-	5.09e+0	1.978 2e+1	5.28e+0
DOC-4	2.571 7e+2-	1.93e+1	3.275 6e+1-	1.84e+1	1.654 1e+1-	1.73e+0	2.133 1e+1-	2.06e+0	6.896 1e+1-	8.41e+0	1.342 4e+1	1.45e+0
DOC-5	2.656 9e+2-	2.04e+1	4.487 5e+1-	3.57e+0	1.591 6e+1-	1.23e+0	2.650 9e+1-	2.00e+0	6.429 0e+1-	4.71e+0	1.401 9e+1	5.43e+0
DOC-6	4.159 5e+2-	2.21e+2	8.753 0e+1-	4.32e+0	2.480 0e+1-	5.20e+0	3.607 9e+1-	9.16e+0	6.672 7e+1-	4.96e+0	1.330 1e+1	1.07e+0
DOC-7	4.684 1e+2-	3.92e+1	5.307 5e+1-	3.89e+0	1.539 3e+1-	1.24e+0	2.703 7e+1-	1.99e+0	7.192 1e+1-	5.49e+0	1.426 0e+1	1.20e+0
DOC-8	1.348 4e+3-	1.09e+2	1.118 2e+2-	8.29e+0	3.757 1e+1+	3.42e+0	1.8494e+2-	1.68e+1	1.7771e+2-	1.39e+1	6.975 1e+1	4.98e+0
DOC-9	1.223 7e+3-	1.12e+2	1.105 7e+2-	7.12e+0	4.377 4e+1+	3.02e+0	1.149 4e+2-	1.43e+1	1.812 1e+2-	1.52e+1	8.014 1e+1	6.48e+0

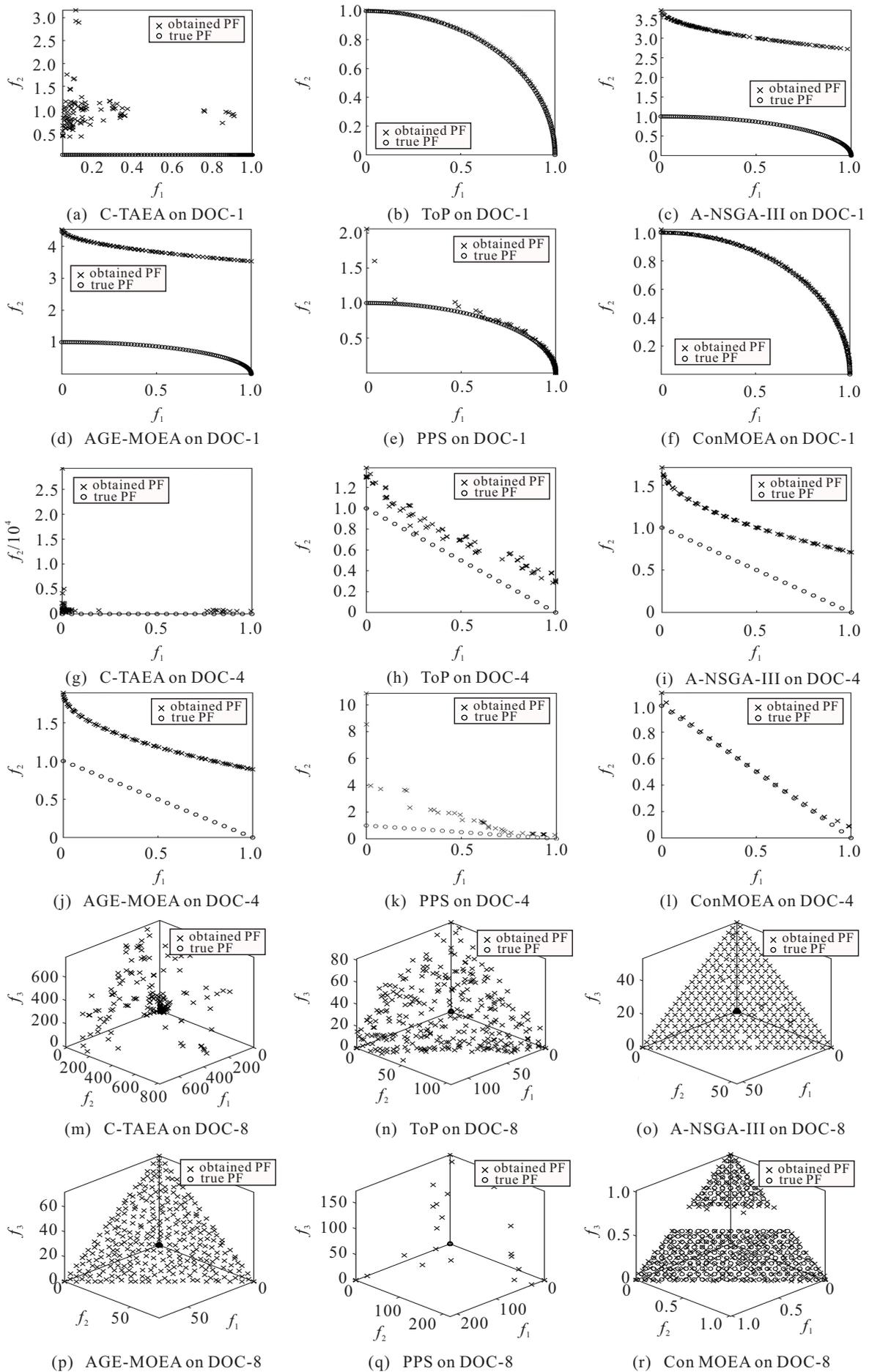


图2 6个算法在DOC-1、DOC-4、DOC-8测试问题结果显示

时间耗费最少,A-NSGA-III算法在DOC-3、DOC-8、DOC-9测试问题上耗时最短.而ConMOEA算法在DOC-2、DOC-4~DOC-7问题上运行时间最短.在DOC-8和DOC-9测试问题上,ConMOEA算法比A-NSGA-III算法耗时略长,比其他算法短.总之,ConMOEA算法在大部分测试问题上运行时间较短,且获得的解不论是分布性还是收敛性都较好.

观察表4~表6中各个算法与ConMOEA的检验结果,比较IGD指标,A-NSGA-III算法在1个测试问题上优于ConMOEA算法,在6个测试问题上劣于ConMOEA算法. ConMOEA分别在5、6、6、7个测试问题上优于C-TAEA、ToP、AGE-MOEA、PPS.对于HV指标,ConMOEA算法均优于其他对比算法.对比运行时间,A-NSGA-III算法在3个测试问题上运行时间比ConMOEA算法短,ToP在1个测试问题上胜过ConMOEA算法,在其他问题上均比ConMOEA算法运行时间长,而剩余对比算法的运行时间没有明显优势.

图2展示了各算法在DOC-1、DOC-4、DOC-8测试问题上独立运行30次的平均结果,在DOC-1问题上,C-TAEA算法完全没有接近Pareto前沿,从 f_2 的范围可以看出,算法在 f_2 上结果特别差. A-NSGA-III和AGE-MOEA算法形成了一组远离真实前沿的凸近似Pareto前沿,但真实Pareto前沿为凹状且 f_1 和 f_2 的值都在0~1范围内,而A-NSGA-III在 f_2 上最大值达到3.5,AGE-MOEA在 f_2 上最大值超过4.仅ToP和ConMOEA算法获得了较优的近似Pareto前沿,但是两端的解稍微稀疏一些,PPS算法获得部分近似前沿.对于DOC-4测试函数,C-TAEA算法的多样性较差,只有个别解分布范围很广,获得的大多数解比较聚集,主要集中在两小部分真实前沿附近. PPS算法获得了小部分接近真实Pareto前沿的有效解,而ConMOEA算法几乎获得整个近似真实Pareto前沿. A-NSGA-III、AGE-MOEA算法求解结果远离真实Pareto前沿且不符合真实前沿的几何形状. ToP算法求解结果存在一些支配解,可能需要进化更多代或一个有效的去除重复解策略来解决这一问题. DOC-8是一个具有3目标的线性不连续前沿测试问题,ConMOEA算法获得的解接近真实Pareto前沿且分布较均匀,而其他5个算法均未获得可行解.

4 结论

本文考虑NSGA-II和AGE-MOEA算法所分别具有的高效收敛性和快速估计Pareto前沿形状的优势,组合Deb约束支配提出了一种协作进化算法.种群进

化前期以NSGA-II算法为搜索引擎寻找最优解,实现快速收敛;中后期采用AGE-MOEA算法,以前期保存的较优可行解为基础作为下一代优选解,能够增加算法运行效率,调整获得的Pareto前沿使其更加符合真实Pareto前沿形状.综合分析协作进化算法与其他6个约束多目标算法在DOC系列基准测试函数上的结果比较,显示协作进化算法的求解结果更加收敛和多样.本文所提出的ConMOEA算法融合了NSGA-II算法,所以NSGA-II算法在求解高维多目标问题时出现性能退化问题也是本研究不足之处,下一步研究将算法应用于求解实际问题以及通过提出有效策略,维持高维多目标约束问题求解结果的收敛性、多样性和可行性平衡.

参考文献(References)

- [1] Tan B X, Ma H, Mei Y. Evolutionary multi-objective optimization for web service location allocation problem[J]. IEEE Transactions on Services Computing, 2018, 14(2): 458-471.
- [2] Gong M G, Wang Z, Zhu Z X. A similarity-based multiobjective evolutionary algorithm for deployment optimization of near space communication system[J]. IEEE Transactions on Evolutionary Computation, 2017, 21(6): 878-897.
- [3] Mendes J B, D'Angelo M F, Maia N A. A hybrid multiobjective evolutionary algorithm for truck dispatching in open-pit-mining[J]. IEEE Transactions on Evolutionary Computation, 2016, 14(3): 1329-1334.
- [4] Deb K, Pratap A, Agarwal S. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.
- [5] Takahama T, Sakai S. Soft computing as transdisciplinary science and technology[M]. Berlin: Springer, 2005: 1019-1029.
- [6] Runarsson T P, Yao X. Stochastic ranking for constrained evolutionary optimization[J]. IEEE Transactions on Evolutionary Computation, 2000, 4(3): 284-294.
- [7] Gong W Y, Cai Z H, L D W. Adaptive ranking mutation operator based differential evolution for constrained optimization[J]. IEEE Transactions on Cybernetics, 2014, 45(4): 716-727.
- [8] Farmani R, Wright J A. Self-adaptive fitness formulation for constrained optimization[J]. IEEE Transactions on Evolutionary Computation, 2003, 7(5): 445-455.
- [9] Tessema B, Yen G G. An adaptive penalty formulation for constrained evolutionary optimization[J]. IEEE

- Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2009, 39(3): 565-578.
- [10] 甘敏, 彭辉, 王勇. 多目标优化与自适应惩罚的混合约束优化进化算法[J]. 控制与决策, 2010, 25(3): 378-382.
(Gan M, Peng H, Wang Y. Multiobjective optimization and adaptive penalty function based constrained optimization evolutionary algorithm[J]. Control and Decision, 2010, 25(3): 378-382.)
- [11] Wang Y, Cai Z X, Guo G Q. Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B, 2007, 37(3): 560-575.
- [12] Ray T, Singh H K, Isaacs A. Constraint-handling in evolutionary optimization[M]. Springer, 2009: 145-165.
- [13] 孟红云, 张小华, 刘三阳. 用于约束多目标优化问题的双群体差分进化算法[J]. 计算机学报, 2008, 31(2): 228-235.
(Meng H Y, Zhang X H, Liu S Y. A differential evolution based on double populations for constrained multi-objective optimization problem[J]. Chinese Journal of Computers, 2008, 31(2): 228-235.)
- [14] Li K, Chen R Z, Fu G T. Two-archive evolutionary algorithm for constrained multiobjective optimization[J]. IEEE Transactions on Evolutionary Computation, 2018, 23(2): 303-315.
- [15] Jain H, Deb K. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach[J]. IEEE Transactions on Evolutionary Computation, 2013, 18(4): 602-622.
- [16] Xiang Y, Peng J, Zhou Y R. An angle based constrained many-objective evolutionary algorithm[J]. Applied Intelligence, 2017, 47(3): 705-720.
- [17] Asafuddoula M, Ray T, Sarker R. An adaptive constraint handling approach embedded MOEA/D[C]. 2012 IEEE Congress on Evolutionary Computation. Brisbane: IEEE, 2012: 1-8.
- [18] Ying W Q, He W P, Huang Y X. An adaptive stochastic ranking mechanism in MOEA/D for constrained multi-objective optimization[C]. 2016 International Conference on Information System and Artificial Intelligence (ISAI). DOI: 10.1109/ISAI.2016.0115.
- [19] Jan M A, K R A. A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D[J]. Applied Intelligence, 2013, 13(1): 128-148.
- [20] Li K, Deb K, Zhang Q F. An evolutionary many-objective optimization algorithm based on dominance and decomposition[J]. IEEE Transactions on Evolutionary Computation, 2014, 19(5): 694-716.
- [21] 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: ACM, 2019: 595-603.
- [22] Li H, Zhang Q F. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2008, 13(2): 284-302.
- [23] Deb K, Jain H. 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, 2013, 18(4): 577-601.
- [24] Liu Z Z, Wang Y. 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.
- [25] Coello C A C, Cortés N C. Solving multiobjective optimization problems using an artificial immune system[J]. Genetic Programming and Evolvable Machines, 2005, 6(2): 163-190.
- [26] Van V, David A. Multiobjective evolutionary algorithms: classifications, analyses, and new innovations[R]. Air Force Inst of Tech Wright-Pattersonafb on School of Engineering, 1999.
- [27] Fan Z, Li W J, Cai X Y. Push and pull search for solving constrained multi-objective optimization problems[J]. Swarm and Evolutionary Computation, 2019, 44: 665-679.
- [28] Tian Y, Cheng R, Zhang X Y. PlatEMO: A Matlab platform for evolutionary multi-objective optimization[J]. IEEE Computational Intelligence Magazine, 2017, 12(4): 73-87.
- [29] Derrac J, García S, Molina D. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms[J]. Swarm and Evolutionary Computation, 2011, 1(1): 3-18.

作者简介

王蕊(1996—), 女, 硕士生, 从事多目标进化算法的研究, E-mail: wangruijdyd@163.com;

顾清华(1981—), 男, 教授, 博士, 从事多目标优化、系统工程理论与应用等研究, E-mail: qinghuagu@126.com.

(责任编辑: 孙艺红)