

基于协同进化的约束多目标优化算法

张祥飞, 鲁宇明*, 张平生

(南昌航空大学 航空制造工程学院, 南昌 330063)

(* 通信作者电子邮箱 luyuming69@163.com)

摘要:针对约束多目标优化算法存在难以有效地兼顾收敛性和多样性的问题,提出一种基于协同进化的约束多目标优化算法。第一阶段,通过基于稳态演化的可行解搜索方式得到一个具有一定数量可行解的种群;第二阶段,将这个种群拆分为两个子种群,并通过双子种群协同进化的方式实现对收敛性和多样性的兼顾;最后采用标准约束多目标优化问题 CF1~CF7、DOC1~DOC7 和实际工程问题进行仿真实验,以测试所提算法的求解性能。实验结果表明,与基于约束支配准则的非支配排序遗传算法(NSGA-II-CDP)、两阶段算法(ToP)、推拉搜索算法(PPS)和约束多目标优化的双存档进化算法(C-TAEA)相比,所提算法在反向世代距离(IGD)和超体积(HV)两个指标上均取得了良好的结果,说明所提算法可以有效地兼顾收敛性和多样性。

关键词:约束多目标优化问题;双种群;协同进化;差分进化;Pareto 前沿

中图分类号:TP301.6 **文献标志码:**A

Constrained multi-objective optimization algorithm based on coevolution

ZHANG Xiangfei, LU Yuming*, ZHANG Pingsheng

(College of Aeronautical Manufacturing Engineering, Nanchang Hangkong University, Nanchang Jiangxi 330063, China)

Abstract: In view of the problem that it is difficult for constrained multi-objective optimization algorithms to effectively balance convergence and diversity, a new constrained multi-objective optimization algorithm based on coevolution was proposed. Firstly, a population with certain number of feasible solutions was obtained by using the feasible solution search method based on steady-state evolution. Then, this population was divided into two sub-populations and both convergence and diversity were achieved by coevolution of the two sub-populations. Finally, standard constrained multi-objective optimization problems CF1~CF7, DOC1~DOC7 and the practical engineering problems were used for simulation experiments to test the solution performance of the proposed algorithm. Experimental results show that compared with Nondominated Sorting Genetic Algorithm II based on Constrained Dominance Principle (NSGA-II-CDP), Two-Phase algorithm (ToP), Push and Pull Search algorithm (PPS) and Two-Archive Evolutionary Algorithm for Constrained multiobjective optimization (C-TAEA), the proposed algorithm achieves good results in both Inverted Generational Distance (IGD) and HyperVolume (HV), indicating that the proposed algorithm can effectively balance convergence and diversity.

Key words: Constrained Multiobjective Optimization Problem (CMOP); double populations; coevolution; differential evolution; Pareto frontiers

0 引言

在实际工程应用中,很多优化问题在优化多个目标的同时还需要处理各种类型的约束,这类问题被称为约束多目标优化问题(Constrained Multi-objective Optimization Problem, CMOP)。早在十几年前,研究人员针对 CMOP 求解已经提出了以无约束的多目标优化方法加约束处理技术的求解方法^[1-3]。随着科技的发展,实际工程应用中的要求也随之提高,使用以往的方法已经难以满足实际使用需求,因此各种新型约束多目标优化算法被陆续提出。文献[4]中提出了一种多阶段的优化方法,在不同的阶段采取不同的优化方法;文献[5]中提出了一种推拉搜索框架,将求解过程分成推拉两个搜索阶段,采用多阶段的算法一般具有较快的收敛速度。

由于协同进化算法在求解大规模、高维度和动态优化问题时取得了良好的效果^[6-8],近些年部分研究人员开始运用协同进化的思想求解 CMOP,并取得了良好的效果。文献[9]中设立了面向收敛的存档和面向多样性的存档两个种群,面向收敛的存档同时优化约束和目标,面向多样性的存档仅优化目标,两个种群在交配选择和环境选择相互合作;文献[10]中将 CMOP 分解成约束单目标优化问题,对每个约束单目标优化问题设置独立的种群进行演化,同时设置了一个用于收集其他子种群中非劣解的子种群,基于协同进化的算法往往可以获取良好的结果。

以上两种类型约束多目标优化算法虽然具有各自的优势,但也存在一定的不足。例如文献[4-5]中提出的算法虽然

收稿日期:2020-09-02;**修回日期:**2021-01-06;**录用日期:**2021-01-25。 **基金项目:**国家自然科学基金面上项目(61866025);江西省教育厅科技项目(GJJ170572);南昌航空大学研究生创新专项基金资助项目(YC2019013)。

作者简介:张祥飞(1995—),男,江西南昌人,硕士研究生,主要研究方向:约束优化、进化算法; 鲁宇明(1969—),女,江西南昌人,教授,博士,主要研究方向:优化理论、进化算法; 张平生(1980—),男,江西南昌人,讲师,博士,主要研究方向:增材制造。

具有良好的收敛速度,但高度依赖决定搜索阶段切换的参数大小选取;文献[9]中算法在高维CMOP上的平衡收敛性和多样性具有良好表现,但是在低维优化问题上表现一般。文献[11]中从MOEA/D内在机制及约束处理技术的适应性上展开研究,提出了一种权重向量和个体间的重新匹配策略实现对多样性和收敛性的兼顾。本文借鉴文献[11]的研究思路,从多阶段和协同进化两种优化方式的互补性展开研究,提出基于协同进化的约束多目标优化算法,通过融合多阶段和协同进化这两种优化方式,具有良好的收敛性能,同时能够较好地兼顾多样性和收敛性。

1 相关研究

1.1 CMOP及其相关定义

不失一般性,一个CMOP表达如式(1)所示:

$$\begin{aligned} \min \quad & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x}))^T \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 0; i = 1, 2, \dots, l \\ & h_j(\mathbf{x}) = 0; j = 1, 2, \dots, q \\ & \mathbf{x} = (x_1, x_2, \dots, x_D)^T \in \Omega \end{aligned} \quad (1)$$

式中: $\mathbf{x}$ 是一个 D 维的决策变量; Ω 表示决策空间; $\mathbf{F}(\mathbf{x})$ 由 n 个实值目标函数组成的目标空间; l 和 q 分别表示不等式和等式约束的数量。

通常情况下,等式约束都会转化成不等式约束的形式,如式(2)所示:

$$|h_j(\mathbf{x})| - \delta \leq 0; j = 1, 2, \dots, q \quad (2)$$

其中: δ 为等式约束的容忍参数,一般设定为0.0001。

决策变量 $\mathbf{x}$ 的约束违反程度如式(3)所示:

$$G(\mathbf{x}) = \sum_{i=1}^l \max(0, g_i(\mathbf{x})) + \sum_{j=1}^q \max(0, |h_j(\mathbf{x})| - \delta) \quad (3)$$

定义1 可行解。决策变量 $\mathbf{x}$ 当且仅当 $G(\mathbf{x}) = 0$ 时,决策变量 $\mathbf{x}$ 为可行解,否则决策变量 $\mathbf{x}$ 为不可行解。

定义2 Pareto支配。决策变量 $\mathbf{x}_u$ Pareto支配决策变量 $\mathbf{x}_v$,记为 $\mathbf{x}_u < \mathbf{x}_v$,当且仅当:

- 1) $\forall i \in \{1, 2, \dots, m\}$ 满足 $f_i(\mathbf{x}_u) \leq f_i(\mathbf{x}_v)$;
- 2) $\exists j \in \{1, 2, \dots, m\}$ 满足 $f_j(\mathbf{x}_u) < f_j(\mathbf{x}_v)$ 。

定义3 Pareto最优解。决策变量 $\mathbf{x}_u$ 作为Pareto最优解,当且仅当决策变量 $\mathbf{x}_u$ 在整个目标集中为非支配解。

定义4 理想点(ideal point)。理想点 $\mathbf{z}^*$ 可表示为 $\mathbf{z}^* = [z_1^*, z_2^*, \dots, z_n^*]^T$,其中, $z_i^* = \min f_i(\mathbf{x}) (i = 1, 2, \dots, n)$,决策变量 $\mathbf{x}$ 为可行解。

定义5 极值点(nadir point)。极值点 $\mathbf{z}^{\text{nad}}$ 可表示为 $\mathbf{z}^{\text{nad}} = [z_1^{\text{nad}}, z_2^{\text{nad}}, \dots, z_n^{\text{nad}}]^T$,其中, $z_i^{\text{nad}} = \max f_i(\mathbf{x}) (i = 1, 2, \dots, n)$,决策变量 $\mathbf{x}$ 为Pareto最优解。

1.2 约束处理技术

近二十几年来,针对单目标优化问题已经有很多约束处理技术被提出,主要有可行性准则、罚函数法、 ε 约束处理法和多目标优化法等^[12]。但这些约束处理技术一般不能直接运用于求解CMOP,虽然它们通过拓展或修改可以运用于求解CMOP,但这类方法具有很大的局限性,在处理低可行域、强约束和不连续帕累托前沿等类型问题时难以取得良好的效果。

在求解CMOP上,主要有多阶段和协同进化两种优化方

式。多阶段优化方式一般是将优化过程划分为多个阶段,不同的阶段采用不同的约束处理技术或优化不同的对象,如文献[4-5]中通过多阶段的形式实现了多种约束处理技术的优势互补。协同进化是指多个对象通过一定的机制和策略开展协同搜索的进化技术,比如多种群、多算法、多操作和多策略集成式进化^[13],这类方法大多采取多种群或多种约束处理技术集成式进化。文献[14-15]中通过多种群的形式集成了多种约束处理技术,不同的种群采用不同的约束处理技术,然后通过种群之间的各种交互方式实现信息共享,有效地发挥了多种约束处理技术优势。

2 基于协同进化的约束多目标优化算法

考虑从可行解的搜索阶段和协同进化阶段入手设计本算法。

1)第一阶段,创建多个子种群的稳态演化方法,保证在演化初始阶段可以获取一批高质量可行解,同时保持种群的多样性;

2)第二阶段,基于可行性准则和指定的子种群规模,将种群中的个体分成两个子种群,并对两个子种群分别采用网格约束分解和加权转化单目标的方法引导两个子种群的进化,分别负责算法的多样性和收敛性。

综上所述,给出本文算法的主要步骤:

步骤1 初始化规模为 $N_1 + N_2$ 的种群 P 。

步骤2 若种群 P 中可行解数量小于 N_1 ,执行第一阶段演化方法;否则,在种群 P 中选择 N_1 个可行解为子种群 P_1 ,子种群 $P_2 = P - P_1$,执行第二阶段演化方法,合并两个子种群。

步骤3 若满足算法停止条件,则输出种群中的可行解;否则,转步骤2。

2.1 稳态演化方法

由于约束数量、形式及数学特性等方面的影响,一个约束优化问题的可行域可能会非常小或离散,这往往会导致求解该问题的算法收敛效果不佳,甚至无法收敛的情况发生。对于一个约束多目标优化问题来讲,在优化多个目标的同时需要处理约束问题,因此在进行多目标优化之前,为配合协同进化,我们需要获取一个拥有一定数量可行解且具有良好多样性的种群。因此,本文提出了一种稳态的演化方法。差分进化^[16]因其简单和有效的搜索能力而受到广泛使用,所以稳态演化选用差分进化作为该方法的搜索引擎。差分进化主要由变异、交叉和选择组成,其中一般常使用的变异算子如式(4)~(6)所示:

DE/rand/1:

$$\mathbf{v}_i = \mathbf{x}_{r_1} + \text{rand} \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \quad (4)$$

DE/current-to-best/1:

$$\mathbf{v}_i = \mathbf{x}_i + F \cdot (\mathbf{x}_{\text{best}} - \mathbf{x}_i) + F \cdot (\mathbf{x}_{r_1} - \mathbf{x}_{r_2}) \quad (5)$$

DE/current-to-rand/1:

$$\mathbf{v}_i = \mathbf{x}_i + \text{rand} \cdot (\mathbf{x}_{r_1} - \mathbf{x}_i) + F \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \quad (6)$$

其中: $\mathbf{x}_{r_1}$ 、 $\mathbf{x}_{r_2}$ 和 $\mathbf{x}_{r_3}$ 是三个相互不同的个体; $\mathbf{x}_{\text{best}}$ 是种群中最好的个体; F 是变异因子。

交叉算子主要在目标向量 $\mathbf{x}_i$ 和变异向量 $\mathbf{v}_i$ 当中生成实验向量 $\mathbf{u}_i = (u_{i,1}, u_{i,2}, \dots, u_{i,D})$,二项式交叉算子如式(7)所示:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{rand}_j < CR \text{ 或 } j = j_{\text{rand}} \\ x_{i,j}, & \text{其他} \end{cases} \quad (7)$$

其中: rand_j 是在 $[0, 1]$ 区间的随机数; j_{rand} 是 $\{1, 2, \dots, D\}$ 中随机的一个整数; CR 是交叉因子。

图 1 展示了稳态演化方法的框架: 首先将种群随机地划分为 m 个子种群, 然后以式 (6)、(7) 进行变异和交叉操作, 得出相应子种群的后代, 在每个子种群之间使用约束支配准则 (Constrained Dominance Principle, CDP)^[1] 更新筛选父本与子代, 得到一个新的子种群, 最后这些新的子种群集成新的种群 P 。该方法的目的在于保持种群多样性的同时寻找到高质量的可行解。

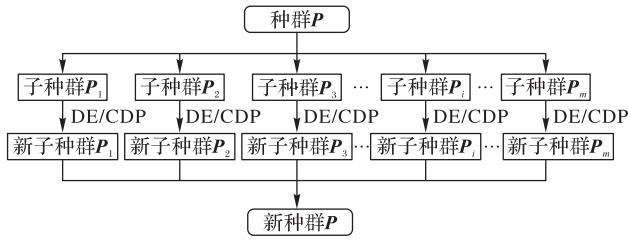


图 1 稳态演化结构

Fig. 1 Steady-state evolution structure

2.2 协同进化策略

在协同进化部分, 考虑到 CMOP 的 Pareto 前沿可能具有各种形状, 例如凸或凹型、连续或不连续等, 为了更好地维持种群的多样性, 进而获取完整的 Pareto 前沿, 本文采用一种基于网格的约束分解方法维持种群的多样性。相较于当前常用的权值求和、切比雪夫和基于惩罚的边界交叉 (Penalty-based Boundary Intersection, PBI) 分解的方法, 该方法利用网格系统维持种群的多样性, 在 Pareto 前沿为凹型时更具优势, 有助于提升算法的普适性。基于网格的约束分解和加权的方法被分别运用于优化两个子种群 P_1 和 P_2 , 以达到同时兼顾多样性和收敛性的目的, 并结合一定的交互机制实现两个子种群之间的信息共享, 进而实现两个子种群之间的协同进化。

2.2.1 基于网格的约束分解

基于网格约束分解的方法由文献[17]针对多目标优化问题 (Multi-objective Optimization Problem, MOP) 提出, 并结合基于分解的排序和字典排序选择进入下一代的解, 该方法能够较好维持进化过程中种群的多样性。因此, 本文引用文献[17]的演化方法运用于子种群 P_1 的演化。其中, 基于网格的约束分解方法具体介绍如下:

1) 网格系统的设置。

首先, 各个目标根据理想点和极值点之间区间划分为 K 个子区间, 则每个子区间的间隔宽度可以定义如式 (8) 所示:

$$d_j = (z_j^{\text{nad}} - z_j^* + 2 \times \sigma) / K \quad (8)$$

对于可行解 x 的第 j 个目标所在的子区间计算如式 (9) 所示:

$$g_j(x) = \lceil (f_j(x) - z_j^* + \sigma) / d_j \rceil \quad (9)$$

其中: z_j^{nad} 和 z_j^* 分别表示极值点和理想点; K 表示子区间的数量; σ 是一项确保 $g_j(x)$ 既不小于 0 也不大于 K 的极小正值参数; $\lceil \cdot \rceil$ 表示向上取整函数。

2) 约束分解的定义。

约束分解的定义以网格系统为基础, 其中第 l 个目标上的第 k 个子问题定义如式 (9) 所示:

$$\min f_l(x) \quad (10)$$

$$\text{s.t. } g_j(x) = k_j$$

$$j = 1, 2, \dots, n, j \neq l$$

$$k_j \in \{1, 2, \dots, K\}$$

$$x \in U$$

其中: K 是一项决定子区间 (网格) 数量的参数; U 表示决策空间中的可行域。

通过将每个目标均匀分成 K 个区间, 把多目标优化问题分解成 $n \times K^{n-1}$ 个子问题。其中, 第 l 个目标的第 k 个子问题所包含的解集定义如式 (11) 所示:

$$S_l(k) = \{x | g_1(x) = k_1, \dots, g_{l-1}(x) = k_{l-1}, \\ g_{l+1}(x) = k_{l+1}, \dots, g_n(x) = k_n\}, \quad (11)$$

$$\text{s.t. } l \in \{1, 2, \dots, n\}, k \in \{1, 2, \dots, K\}^{n-1}$$

3) 基于排序的选择方法。

基于排序的选择方法主要由基于分解的排序和字典排序组成, 主要用于选择进入下一代演化的个体。以上部分介绍了约束分解的子问题, 每个解集 $S_l(k)$ 都包含一个解集, 基于分解的排序主要对于每个解集 $S_l(k)$ 按式 (11) 进行排序, 因此每个解都具有 n 个排序值, 然后通过字典排序选择前 N_1 个解作为下一代的父本种群。如图 2 和表 1 所示, 该问题为网格参数 $K = 4$ 时的双目标优化问题, 经过基于分解的排序后, 每个解的排序值如表 1 第二列所示 (括号前数字为排序值), 第三列前 7 个解为字典排序后选择的解 (标注 $\checkmark$)。

表 1 $K = 4$ 时基于分解的排序和字典排序的过程

Tab. 1 Procedures of decomposition-based ranking and lexicographic sorting when $K = 4$

$R(x)$	$R'(x)$	字典排序
1. (1, 2)	1. (1, 2)	1. (1, 2) $\checkmark$
2. (2, 1)	2. (1, 2)	2. (1, 2) $\checkmark$
3. (1, 3)	3. (1, 3)	8. (1, 2) $\checkmark$
4. (2, 2)	4. (2, 2)	3. (1, 3) $\checkmark$
5. (3, 1)	5. (1, 3)	5. (1, 3) $\checkmark$
6. (4, 5)	6. (4, 5)	12. (1, 3) $\checkmark$
7. (1, 4)	7. (1, 4)	14. (1, 3) $\checkmark$
8. (2, 1)	8. (1, 2)	7. (1, 4) $\times$
9. (3, 3)	9. (3, 3)	4. (2, 2) $\times$
10. (4, 2)	10. (2, 4)	13. (2, 2) $\times$
11. (5, 4)	11. (4, 5)	10. (2, 4) $\times$
12. (1, 3)	12. (1, 3)	9. (3, 3) $\times$
13. (2, 2)	13. (2, 2)	6. (4, 5) $\times$
14. (3, 1)	14. (1, 3)	11. (4, 5) $\times$

2.2.2 加权转化方法

求解 CMOP 最终目的是为了获取 Pareto 前沿, 虽然通过加权的方式将 CMOP 转化成约束单目标优化问题仅仅只能获取该问题 Pareto 前沿中的某一个解, 但该方式可以显著地提高算法的收敛性能。设置合适的权值既可以保持算法具有良好的收敛性又能尽可能地搜索到 Pareto 前沿的中部位置。

在权值的设置上, 文献[4]中提出了一种特殊的加权方式将 CMOP 转化为约束单目标优化问题, 该方法显著地提升了算法的收敛能力。在子种群 P_2 的演化过程中, 为提升算法整体收敛能力, 同样也采用类似的加权方法将 CMOP 转化成约束单目标优化问题来处理, 并采用式 (5) 的变异方式和基于可行性准则的约束处理技术对子种群 P_2 进行演化。其中, 加权方法如式 (12) 所示:

$$\begin{aligned} \min \quad & f(\mathbf{x}) = \sum_{i=1}^n \frac{f_i(\mathbf{x})}{n} \\ \text{s. t.} \quad & g_j(\mathbf{x}) \leq 0; j = 1, 2, \dots, l \\ & h_j(\mathbf{x}) = 0; j = l + 1, \dots, q \\ & \mathbf{x} = (x_1, x_2, \dots, x_D)^T \in \Omega \end{aligned} \quad (12)$$

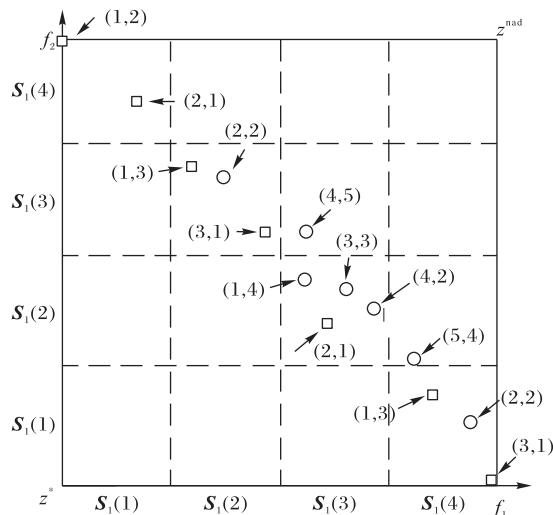


图2 双目标优化问题在网格系统中所有解的排序

Fig. 2 Ranking of all solutions of bi-objective optimization problem in grid system

2.2.3 交互机制

在协同进化过程中,通过种群间的有效交互方式实现信息共享,可以显著地改善算法的求解效果。本文基于可行性准则及设定的子种群规模将种群 P 划分为两个子种群 P_1 和 P_2 ,并分别采用基于网格的约束分解和加权的方式进行演化。在2.2.1节中,我们采用文献[17]的演化方式对子种群 P_1 进行演化,虽然该方法可以较好地维持种群的多样性,但在约束优化问题中,这种基于网格的演化方法难以跳出不可行域,进而会导致算法无法收敛。为解决该问题,在子种群 P_2 的变异方式中,随机地选择子种群 P_1 中构成理想点的个体作为 $\mathbf{x}_{\text{best}}$,引导子种群 P_2 向理想点的位置演化,演化过程中子种群 P_2 所获取的高质量个体通过种群重组的方式进入子种群 P_1 ,进而引导子种群 P_1 的进化,并跳出不可行域。

3 实验与分析

为验证本文算法的性能,选择变量维度在6到16之间且具有凹和凸型Pareto前沿的约束多目标优化问题CF^[18]和DOC^[4]测试问题集进行测试及分析。选择了基于约束支配准则的非支配排序遗传算法(Nondominated Sorting Genetic Algorithms II based on Constrained Dominance Principle, NSGA-II-CDP)^[1]和近几年新提出的约束多目标优化算法:两阶段算法(Two-Phase algorithm, ToP^[4])、推拉搜索算法(Push and Pull Search algorithm, PPS^[5])和约束多目标优化的双存档进化算法(Two-Archive Evolutionary Algorithm for Constrained Multiobjective Optimization, C-TAEA)^[9]作为对比。

3.1 评价指标

本实验中选取反向世代距离(Inverted Generational Distance, IGD)和超体积指标(Hypervolume, HV)作为评价指标。IGD指标衡量的是真实Pareto前沿中的点到所求得Pareto前沿近似解集之间最小距离的平均值;HV指标是指被

非支配解集覆盖的目标空间区域大小。设 P^* 是真实Pareto前沿上均匀采样的解集, P^R 是算法求得的Pareto前沿近似解集,其定义分别如式(13)、(14)所示:

$$IGD(P^R, P^*) = \frac{1}{|P^*|} \sum_{\mathbf{x} \in P^*} \text{dist}(\mathbf{x}, P^R) \quad (13)$$

$$HV(S) = \text{vol} \left(\bigcup_{\mathbf{x} \in P^R} [f_1(\mathbf{x}), r_1^*] \times [f_2(\mathbf{x}), r_2^*] \times \dots \times [f_m(\mathbf{x}), r_m^*] \right) \quad (14)$$

其中: $\text{dist}(\mathbf{x}, P)$ 表示个体 $\mathbf{x} \in P^*$ 到 P^R 上离其最近个体之间的欧氏距离; $|P^*|$ 是集合 P^* 中的基数; $\text{vol}(\cdot)$ 表示勒贝格测度; $r^* = [r_1^*, r_2^*, \dots, r_m^*]$ 表示分布在目标空间的参考点,参考点为Pareto前沿极值点的1.1倍。IGD的值越小或HV值越大,表示该算法具有越好的收敛性和多样性,也说明 P^R 更加近似Pareto前沿。

3.2 实验参数设置

设置种群大小 $N = 450$,根据文献[17]中的建议设置子种群 $N_1 = 300$,网格划分参数 $K = 140$,邻居间隔设置为5; $N_2 = 150$,稳态演化的子种群划分数量 $m = 30$;所有算法的终止条件为函数评估次数达到300 000;对于所有测试问题集算法独立运行51次,最后结果取平均值。

3.3 实验结果及分析

实验比较了本文算法与NSGA-II-CDP、ToP、PPS和C-TAEA在测试集CF和DOC的IGD和HV指标的平均值和均方差,结果如表2和表3所示,其中粗体IGD和HV指标结果为五种算法中最优结果,NaN表示未寻找到可行解,括号内为均方差结果。在测试集中,测试集CF1~CF3的Pareto前沿是这些问题在无约束情况下Pareto前沿的一部分;测试集CF4~CF7的主要难点在于搜索前沿,因为很多Pareto前沿的点分布在约束边界位置。相对于前者,测试集CF4~CF7对算法是否能够有效兼顾多样性和收敛性能提出了更高的要求。测试集DOC同时拥有目标与决策变量之间约束,更符合实际应用需求。

与NSGA-II-CDP和C-TAEA算法相比,对于测试集CF和DOC,本文算法在IGD和HV指标上均能获得较好的结果。

与PPS算法相比,对于CF测试集中CF3、CF4和CF7以及DOC测试集,本文算法在IGD指标和HV指标上获得较好的结果,而PPS算法在测试问题CF1、CF2和CF5上取得较好结果。

与ToP算法相比,对于测试集中CF2、CF4和CF7以及DOC测试集,本文算法在IGD指标和HV指标上获得较好结果,对于测试问题CF3,两种算法在HV指标上获得了相似的结果,而ToP算法在测试问题CF1和CF5取得较好结果。

在收敛性和多样性平衡方面,协同进化过程中,单目标演化起到引导种群进化的作用,多目标演化起到维持种群多样性的作用,图3分别描绘了本文算法在DOC1问题上早中晚三个时期的两个子种群的目标空间中分布情况。从图中可以看出,在协同进化初期阶段,第一阶段的稳态演化在搜索到一定的可行解后,种群中的可行解在目标空间分布相对均匀,保持了良好的多样性;在协同进化中期阶段,子种群 P_2 引导子种群 P_1 演化,跳出不可行域,同时子种群 P_1 基于网格约束分解的演化方法主要通过参考点的方式维持种群的多样性,避免

收敛到局部的 Pareto 前沿;在协同进化后期阶段,子种群 P_1 引导种群演化过程结束,子种群 P_1 中个体逐渐聚集于 Pareto 前沿的一个端点,子种群 P_2 采用的多样性管理措施发挥作用。基于邻居关系的演化方法和网格多样性管理方法在维持种群多样性方面具有良好的优势,因此,采用单目标和多目标相结合的协同进化方式在兼顾收敛性和多样性方面具有良好的收敛效果。

综上所述,本文算法在测试集上取得的良好结果,得益于多阶段和协同进化两种优化方式的良好结合,在第一阶段的

稳态演化以多个子种群分别演化的形式保持种群的多样性,并搜索高质量的可行解;第二阶段的协同进化阶段以双种群协同演化的方式兼顾收敛性和多样性。其中,两个子种群分别以基于网格约束分解的方式和加权的方式实现算法的多样性和收敛性,两个子种群通过交互机制实现各自所分配任务的协同。因此,本文算法在求解具有低可行域、强约束和不连续 Pareto 前沿的测试集 DOC 中表现出了良好的优势,在求解更注重搜索能力的测试集 CF 时也具有良好的效果。

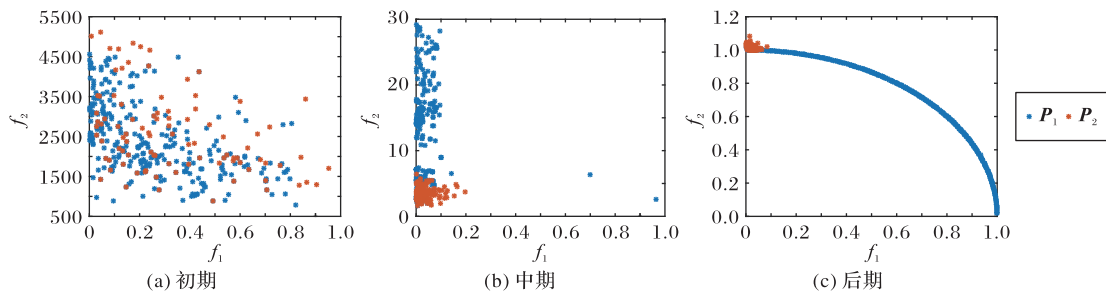


图3 DOC1的协同进化过程

Fig. 3 Coevolutionary process of DOC1

表2 IGD的均值与均方差对比

Tab. 2 Comparison of mean and mean squared deviation for IGD

实例	NSGA-II-CDP	PPS	ToP	C-TAEA	本文算法
CF1	9.134 1e-3 (1.51e-3)	9.986 9e-4 (3.61e-4)	6.248 1e-4 (3.09e-4)	3.694 3e-2 (4.82e-3)	5.458 7e-3 (4.95e-4)
CF2	2.521 8e-2 (1.63e-2)	3.695 4e-3 (7.25e-4)	2.701 9e-2 (1.28e-1)	2.258 0e-2 (5.52e-3)	1.863 9e-2 (5.34e-3)
CF3	3.109 7e-1 (1.94e-1)	2.384 5e-1 (1.37e-1)	2.021 1e-1 (7.37e-2)	2.560 9e-1 (8.79e-2)	1.643 6e-1 (3.61e-2)
CF4	1.030 9e-1 (3.96e-2)	3.784 9e-2 (3.49e-2)	4.976 7e-2 (9.45e-2)	8.731 1e-2 (3.28e-2)	5.864 5e-3 (1.75e-3)
CF5	2.732 6e-1 (9.62e-2)	2.100 5e-1 (1.09e-1)	2.503 8e-1 (9.12e-2)	3.568 4e-1 (1.45e-1)	2.495 2e-1 (1.29e-1)
CF6	8.521 8e-2 (3.67e-2)	5.343 1e-2 (1.88e-2)	5.184 5e-2 (3.84e-2)	1.011 8e-1 (4.12e-2)	5.327 5e-2 (2.39e-2)
CF7	3.349 7e-1 (2.28e-1)	2.286 3e-1 (8.51e-2)	2.468 4e-1 (1.78e-1)	2.876 4e-1 (1.52e-1)	8.795 6e-2 (4.87e-2)
DOC1	6.318 2e+1 (7.01e+1)	5.189 7e-2 (2.80e-2)	5.980 6e-3 (2.65e-4)	2.717 7e+2 (0.00e+0)	1.851 2e-3 (1.21e-04)
DOC2	NaN (NaN)	4.874 2e-2 (3.87e-2)	3.510 2e-1 (1.57e-1)	NaN (NaN)	6.534 9e-3 (4.68e-03)
DOC3	5.830 9e+2 (2.63e+2)	4.873 9e-1 (3.73e-1)	8.729 0e+1 (1.51e+2)	NaN (NaN)	4.267 4e-3 (1.79e-03)
DOC4	6.684 1e-1 (3.84e-1)	2.519 7e-1 (3.94e-2)	5.469 8e-2 (4.69e-2)	4.605 2e+2 (0.00e+0)	2.358 2e-2 (5.94e-03)
DOC5	NaN (NaN)	7.187 9e+1 (6.42e+1)	9.323 7e+0 (2.84e+1)	NaN (NaN)	2.367 3e-2 (7.86e-03)
DOC6	2.064 3e+0 (2.26e+0)	4.617 5e-1 (9.87e-2)	2.576 1e+0 (8.36e-1)	1.881 3e+1 (0.00e+0)	2.103 4e-3 (2.08e-04)
DOC7	6.361 2e+0 (2.55e+0)	3.519 4e-1 (1.90e-1)	3.805 6e-1 (1.27e-1)	NaN (NaN)	1.859 4e-3 (1.47e-04)

表3 HV的均值与均方差对比

Tab. 3 Comparison of mean and mean squared deviation for HV

实例	NSGA-II-CDP	PPS	ToP	C-TAEA	本文算法
CF1	5.547 2e-1 (1.93e-3)	5.640 3e-1 (4.67e-4)	5.660 0e-1 (3.93e-4)	5.198 7e-1 (4.87e-3)	5.586 5e-1 (5.61e-04)
CF2	6.410 2e-1 (1.86e-2)	6.789 1e-1 (1.31e-3)	6.675 9e-1 (3.84e-2)	6.523 8e-1 (1.36e-2)	6.757 1e-1 (3.56e-03)
CF3	1.860 8e-1 (5.72e-2)	1.952 6e-1 (5.94e-2)	2.023 9e-1 (3.91e-2)	1.986 4e-1 (4.69e-2)	1.999 6e-1 (2.35e-02)
CF4	4.205 6e-1 (3.58e-2)	4.947 1e-1 (3.79e-2)	4.843 8e-1 (7.64e-2)	4.220 6e-1 (3.97e-2)	5.301 9e-1 (1.61e-02)
CF5	2.928 7e-1 (6.680e-2)	3.531 2e-1 (7.48e-2)	2.950 9e-1 (5.99e-2)	2.570 5e-1 (7.98e-2)	3.458 1e-1 (7.37e-02)
CF6	6.423 6e-1 (1.77e-2)	6.530 9e-1 (1.76e-2)	6.624 0e-1 (2.75e-2)	6.439 8e-1 (2.08e-2)	6.940 6e-1 (1.84e-02)
CF7	4.458 2e-1 (1.98e-1)	4.568 0e-1 (6.43e-2)	4.235 3e-1 (1.28e-1)	4.438 9e-1 (8.01e-2)	5.891 2e-1 (5.85e-02)
DOC1	0.000 0e+0 (0.00e+0)	2.949 2e-1 (2.15e-2)	3.452 9e-1 (6.23e-4)	0.000 0e+0 (0.00e+0)	3.527 6e-1 (1.79e-3)
DOC2	NaN (NaN)	5.450 1e-1 (5.72e-2)	3.365 8e-1 (1.84e-1)	NaN (NaN)	6.241 6e-1 (6.93e-3)
DOC3	0.000 0e+0 (0.00e+0)	1.141 6e-1 (9.87e-2)	0.000 0e+0 (0.00e+0)	NaN (NaN)	3.256 2e-1 (4.64e-3)
DOC4	9.963 3e-2 (1.41e-1)	2.866 9e-1 (3.44e-2)	5.006 2e-1 (5.82e-2)	0.000 0e+0 (0.00e+0)	5.385 1e-1 (4.92e-3)
DOC5	NaN (NaN)	1.530 4e-1 (1.95e-1)	2.810 0e-1 (1.98e-1)	NaN (NaN)	4.846 2e-1 (5.79e-3)
DOC6	3.984 5e-2 (9.73e-2)	2.408 6e-1 (4.66e-2)	0.000 0e+0 (0.00e+0)	0.000 0e+0 (0.00e+0)	5.168 6e-1 (5.93e-3)
DOC7	0.000 0e+0 (0.00e+0)	2.178 3e-1 (9.95e-2)	1.479 0e-1 (1.42e-1)	NaN (NaN)	5.185 2e-1 (7.26e-3)

3.4 参数敏感性分析

3.4.1 子种群数量 m

为研究稳态演化中设定的子种群数量对算法性能的影响,分别在测试问题 CF1、CF3、DOC1、DOC4 和 DOC6 上测试了当 m 分别为 10、15、25、30、45 时的求解效果,以算法独立运行 51 次结果的平均 IGD 值作为评价标准,函数评价次数为 300 000。测试结果如表 4 所示,可以看出,本文算法在稳态演化阶段中采用具有相对较低敏感性的子种群数量。

表 4 m 取不同值时的 IGD 均值对比

Tab. 4 Comparison of mean values of IGD with different m

实例	m				
	10	15	25	30	45
CF1	0.008 9	0.006 8	0.005 9	0.005 5	0.005 3
CF3	0.308 6	0.274 1	0.164 6	0.164 4	0.164 9
DOC1	0.002 9	0.002 0	0.001 9	0.001 9	0.001 9
DOC4	0.034 9	0.030 8	0.023 5	0.023 6	0.029 1
DOC6	0.003 1	0.002 7	0.002 3	0.002 1	0.002 2

3.4.2 子种群规模 N_2

为研究协同进化中子种群数量对算法性能的影响,本文分别在测试问题 CF1、CF3、DOC1、DOC4 和 DOC6 上测试了四种子种群规模 (N_1, N_2) 组合形式别为 $(300, 50)$ 、 $(300, 100)$ 、 $(300, 150)$ 、 $(300, 200)$ 时求解效果。由于本文采用基于网格约束分解的方法,所以按文献[17]中的建议设定 $N_1 = 300$ 。以算法独立运行 51 次结果的平均 IGD 值作为评价标准,函数评价次数为 300 000。测试结果如表 5 所示,可以看出,本文算法在稳态演化阶段中采用具有相对较低敏感性的子种群数量。

表 5 N_2 取不同值时的 IGD 均值对比

Tab. 5 Comparison of mean values of IGD with different N_2

实例	(300, 50)	(300, 100)	(300, 150)	(300, 200)
CF1	0.005 8	0.005 6	0.005 5	0.005 6
CF3	0.153 9	0.158 2	0.164 4	0.165 8
DOC1	0.002 0	0.001 9	0.001 9	0.001 9
DOC4	0.024 9	0.024 0	0.023 6	0.023 6
DOC6	0.002 3	0.002 1	0.002 1	0.002 2

4 实例验证

采用盘式制动设计问题^[19]进一步检验本文算法的有效性。该问题具有两个目标,分别是制动器的重量和停止时间;包含四个变量,其中 x_1 和 x_2 分别是制动器内外圈直径, x_3 表示啮合力, x_4 表示摩擦面的数量;该问题的约束条件分别是半径之间的最小距离、制动的最大长度、压力、温度和扭矩限制,具体优化问题及约束如式 (15) 所示,其中: $x_1 \in [55, 80]$, $x_2 \in [75, 110]$, $x_3 \in [1\ 000, 3\ 000]$, $x_4 \in [2, 20]$ 。

$$\min \begin{cases} f_1(\mathbf{x}) = 4.9 \times 10^{-5} (x_2^2 - x_1^2)(x_4 - 1) \\ f_2(\mathbf{x}) = \frac{9.82 \times 10^6 (x_2^2 - x_1^2)}{x_3 x_4 (x_2^3 - x_1^3)} \end{cases} \quad (15)$$

$$\text{s. t. } g_1(\mathbf{x}) = (x_2 - x_1) - 20 \geq 0$$

$$g_2(\mathbf{x}) = 30 - 2.5(x_4 + 1) \geq 0$$

$$g_3(\mathbf{x}) = 0.4 - \frac{x_3}{3.14(x_2^2 - x_1^2)} \geq 0$$

$$g_4(\mathbf{x}) = 1 - \frac{2.22 \times 10^{-3} x_3 (x_2^3 - x_1^3)}{(x_2^2 - x_1^2)^2} \geq 0$$

$$g_5(\mathbf{x}) = \frac{2.66 \times 10^{-2} x_3 x_4 (x_2^3 - x_1^3)}{(x_2^2 - x_1^2)} - 900 \geq 0$$

各算法在该问题上独立运行 51 次,函数评价次数设定为 300 000,比较求解结果 HV 值的最优值(best)、均值(mean)与标准差(std),参考点按照本文算法获取的最大极值点的 1.1 倍设定。

求解结果如表 6 所示,可以看出,本文算法较对比算法具有一定的优势。

表 6 五种算法在盘式制动设计问题上的 HV 指标比较

Tab. 6 Comparison of five algorithms on disk brake design problem in terms of HV indicator

算法	best	mean	std
NSGA-II-CDP	0.770 2	0.764 8	0.005 6
PPS	0.772 1	0.765 1	0.005 6
ToP	0.774 5	0.764 9	0.005 6
C-TAEA	0.771 5	0.764 2	0.005 6
本文算法	0.778 6	0.765 4	0.005 6

5 结语

本文提出了一个基于协同进化的约束多目标优化算法,通过网格约束分解和加权的方式实现协同进化,能有效地兼顾进化过程中算法的收敛性和多样性,并结合稳态演化方法以保证该算法前期具有良好搜索性能,最后通过测试集和实例验证了算法的有效性。算法的不足之处在于采用基于网格约束分解的方法导致算法种群规模较大,相同评价次数的情况下,本文算法迭代次数偏低,影响求解质量,因此,多样性管理措施将是未来主要研究方向。

参考文献 (References)

- [1] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.
- [2] WOLDESENBET Y G, YEN G G, TESSEMA B G. Constraint handling in multiobjective evolutionary optimization [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(3): 514-525.
- [3] JAN M A, KHANUM R A. A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D [J]. Applied Soft Computing, 2013, 13(1): 128-148.
- [4] LIU 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.
- [5] FAN Z, LI W, CAI X, et al. Push and pull search for solving constrained multi-objective optimization problems [J]. Swarm and Evolutionary Computation, 2019, 44: 665-679.
- [6] YANG Z, TANG K, YAO X. Large scale evolutionary optimization using cooperative coevolution [J]. Information Sciences, 2008, 178(15): 2985-2999.
- [7] WANG H, JIAO L, YAO X. Two_Arch2: an improved two-archive algorithm for many objective optimization [J]. IEEE Transactions on Evolutionary Computation, 2015, 19(4): 524-541.
- [8] GOH C K, TAN K C. A competitive cooperative coevolutionary

- paradigm for dynamic multiobjective optimization [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(1):103-127.
- [9] LI K, CHEN R, FU G, et al. Two-archive evolutionary algorithm for constrained multiobjective optimization [J]. IEEE Transactions on Evolutionary Computation, 2019, 23(2):303-315.
- [10] WANG J, LIANG G, ZHANG J. Cooperative differential evolution framework for constrained multiobjective optimization [J]. IEEE Transactions on Cybernetics, 2019, 49(6):2060-2072.
- [11] 张磊, 毕晓君, 王艳娇. 基于重新匹配策略的 ε 约束多目标分解优化算法[J]. 电子学报, 2018, 46(5):1032-1040. (ZHANG L, BI X J, WANG Y J. The ε constrained multi-objective decomposition optimization algorithm based on re-matching strategy [J]. Acta Electronica Sinica, 2018, 46(5):1032-1040.)
- [12] 李智勇, 黄滔, 陈少森, 等. 约束优化进化算法综述[J]. 软件学报, 2017, 28(6):1529-1546. (LI Z Y, HUANG T, CHEN S M, et al. Overview of constrained optimization evolutionary algorithms [J]. Journal of Software, 2017, 28(6):1529-1546.)
- [13] 王凌, 沈婧楠, 王圣尧, 等. 协同进化算法研究进展[J]. 控制与决策, 2015, 30(2):193-202. (WANG L, SHEN J N, WANG S Y, et al. Advances in co-evolutionary algorithms [J]. Control and Decision, 2015, 30(2):193-202.)
- [14] QU B Y, SUGANTHAN P N. Constrained multi-objective optimization algorithm with an ensemble of constraint handling methods[J]. Engineering Optimization, 2011, 43(4):403-416.
- [15] MALLIPEDDI R, SUGANTHAN P N. Ensemble of constraint handling techniques [J]. IEEE Transactions on Evolutionary Computation, 2010, 14(4):561-579.
- [16] STORN R, PRICE K. Differential evolution - simple and efficient heuristic for global optimization over continuous spaces [J]. Journal of Global Optimization, 1997, 11(4):341-359.
- [17] CAI X, MEI Z, FAN Z, et al. A constrained decomposition approach with grids for evolutionary multiobjective optimization [J]. IEEE Transactions on Evolutionary Computation, 2018, 22(4):564-577.
- [18] ZHANG Q, ZHOU A, ZHAO S, et al. Multiobjective optimization test instances for the CEC 2009 special session and competition: CES-487[R]. Colchester: University of Essex, 2009.
- [19] RAY T, LIEW K M. A swarm metaphor for multiobjective design optimization [J]. Engineering Optimization, 2002, 34(2):141-153.

This work is partially supported by Surface Program of the National Natural Science Foundation of China (61866025), the Jiangxi Education Department Science and Technology Project (GJJ170572), the Postgraduate Innovation Special Project of Nanchang Hangkong University (YC2019013).

ZHANG Xiangfei, born in 1995, M. S. candidate. His research interests include constrained optimization, evolutionary algorithm.

LU Yuming, born in 1969, Ph. D., professor. Her research interests include optimization theory, evolutionary algorithm.

ZHANG Pingsheng, born in 1980, Ph. D., lecturer. His research interests include additive manufacturing.