

## 从动态适值空间补偿信息: 一种抗病态合作协同进化算法

彭星光, 张 帅, 刘 坤

(西北工业大学 航海学院, 西安 710072)

**摘 要:** 针对合作协同进化算法(CCEA)动态适值空间的特点, 研究信息补偿方法以消除由问题分解所导致的病态现象, 并提出基于动态多种群进化策略的抗病态CCEA. 每个协进化种群可动态分离出多个变化的子种群, 利用它们同时获取多个全局或局部最优解作为交互信息, 以实现信息补偿. 针对引发病态行为的标准测试函数, 与3种典型CCEA进行比较分析, 实验结果表明所提出算法能有效克服病态现象, 具有良好的全局优化能力.

**关键词:** 合作协同进化算法; 动态多种群策略; 信息补偿

**中图分类号:** TP273

**文献标志码:** A

## Compensate information from dynamic landscapes: An anti-pathology cooperative coevolutionary algorithm

PENG Xing-guang, ZHANG Shuai, LIU Kun

(School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China.

Correspondent: PENG Xing-guang, E-mail: pxg@nwpu.edu.cn)

**Abstract:** In order to counteract the pathologies of cooperative coevolutionary algorithms(CCEAs) caused by information loss when dealing with problem decomposition, the information compensation strategy is investigated with respect to the dynamic nature of the landscapes of the CCEAs. A dynamic multi-population evaluation based anti-pathology CCEA is proposed. In the algorithm, several dynamic child populations can be split off from a coevolutionary population and search global or local optimum which are used as the interacting information to compensate information. Two pathology-causing benchmark problems are used to test and compare the proposed algorithm to three classical CCEAs. Experimental results show that the proposed algorithm effectively counteract the relative overgeneralization pathology and significantly improve the rate of global-optimization convergence.

**Keywords:** cooperative coevolutionary algorithm; dynamic multi-population strategy; information compensation

### 0 引 言

近40年来,生物学上兴起了另一个重要理论——协同进化论,认为某些物种的进化与另一些物种的进化相互关联、相互受益,不同物种不同个体之间既有相互受益又有相互制约.受此协同进化现象的启发,竞争型协同进化算法和合作型协同进化算法相继被提出.前者模拟自然界中的捕食与被捕食关系,后者则模拟自然中的共生现象.

合作协同进化算法(CCEA)的个体适应度评价方式与传统进化算法截然不同,子种群中的个体不再对所有决策变量进行编码,而是仅对其中的部分

变量进行编码.在进行适应度评价时,需要利用其他种群所交互的个体信息构成一个完整的决策变量编码,方可进行适应度评价.因此,CCEA独特的种群协作进化机制具有一个显著的特点——“分而治之”,可将复杂问题转化为多个相对简单的少变量优化问题.CCEA已被成功地应用于神经网络优化<sup>[1]</sup>、产品设计<sup>[2]</sup>、协调与调度<sup>[3]</sup>,以及无人机协同航迹规划<sup>[4]</sup>等领域.

作为进化计算领域的新兴研究方向,国内外对CCEA的研究主要是从理论和实验的角度来研究如何确保算法全局最优性.文献[5]指出,可认为协进化

收稿日期: 2013-11-13; 修回日期: 2014-03-03.

基金项目: 国家自然科学基金项目(61105068, 61473233); 西北工业大学基础研究基金项目(JCY20130110); 陕西省自然科学基金基础研究计划项目(2014JQ8330); 中央高校基本科研业务费项目(3102014ZD0042).

作者简介: 彭星光(1981—),男,副教授,博士,从事进化计算、水下航行器等研究; 张帅(1991—),男,硕士生,从事进化计算的研究.

算法中进行适应度评价的,实际上并不是个体本身,而是种群间的交互内容.因此,大多数研究工作主要是在传统CCEA的基础上设计额外的信息交互和使用机制,以克服相对过分概括等传统CCEA固有的病态现象.

文献[6]提出了基于存档的CCEA,通过保留在协进化过程中与其他个体合作较好的个体,为个体的正确评价提供更多信息,从而有效提高了算法在搜索空间中存在无限Nash均衡情况下,找到全局最优解的概率.此外,文献[7]还通过将每个子种群的个体连成一个环,在一定程度上保存了进化信息,并通过偏好参数影响个体适应度的评价.

文献[8]提出了一种博弈记忆策略,通过将Nash均衡进行记忆,有效提高了算法的性能;然而,由于该算法假设记忆空间是无限的,无法用于工程实践.此外,文献[9]提出了Pareto协进化概念,其思想是用多目标优化中Pareto解的思想对个体进行评价,而不是求平均值.

国内学者同样对协进化算法展开了卓有成效的研究,在CCEA方面取得了一系列研究成果.文献[10]在种群间引入局部交互,并通过机器人推箱实验验证了算法的有效性;文献[11]提出了组织协同进化算法求解数值优化问题;文献[12]对CCEA进行了深入研究,并探讨了服装设计和机器人路径规划等领域的应用问题.

实际上,造成CCEA出现病态现象的根本原因是问题分解时所造成的进化信息丢失,导致个体适应度无法被正确地评价.这种病态现象是随着CCEA“分而治之”的进化机制与生俱来的,与传统进化算法的局部最小问题存在着根本的不同.此外,由于CCEA中个体适应度的评价依赖于其他协进化种群中个体的状态,而且每个协进化种群都不断演变,因此,协进化种群的进化过程可视作在动态适值空间进行<sup>[5]</sup>.

为此,本文基于动态多种群策略,在动态适值空间中同时获取多个全局或局部最优解作为交互信息,对协进化信息进行补偿,从而提高CCEA的抗病态和全局优化能力.

## 1 基于动态多种群策略的CCEA

动态适值空间进化算法大致可分为<sup>[13]</sup>: 1) 重启、重初始化法; 2) 记忆法; 3) 自适应变异法; 4) 修正选择法; (5) 基于历史信息的预测法; 6) 多种群法.

上述6类方法中,方法1)作为一种朴素方法,具有很好的通用性,但它过于简单,动态优化性能有

限,常被用作对比算法来验证其他算法的性能.方法2)~方法5)都具有一定的针对性,即这些方法所适用的问题具有一定的特殊性.比如:记忆法,只有在适值空间的变化具有循环性时才能有效工作;预测法,则要求最优点在动态适值空间中的变化是可预测的.

由于协进化过程的复杂性,对于每个协进化种群适值空间的变化情况很难进行分析和预测,因此方法2)~方法5)中对适值空间具有特殊要求的方法很难保证有效工作.相比而言,多种群方法利用多个种群,对适值空间任意时刻出现的当前局部或全局最优解进行处理,而这些最优解中很有可能包含真正的全局最优解或其近似解.因此,从这个角度看,多种群法所发现的信息是能够代表当前协进化种群状态的“重要”信息.如果将这些信息作为交互内容发送给其他协进化种群,则既能帮助其他协进化种群更准确地进行适应度评估,又能有的放矢,有选择性地确定交互内容,这样可避免大量的随机交互内容.

### 1.1 动态多种群策略

自组织侦测法(SOS)<sup>[14]</sup>是一种动态适值空间下的多种群进化算法,它类似于聚类的种群动态生成方法,为利用进化算法求解无先验知识的动态优化问题提供了有效途径和思路.本节将对SOS算法进行优化和调整,使其能够引入到本文算法的框架下,以验证基于动态多种群策略CCEA的有效性.通过对SOS算法的研究,发现其参数多、过于复杂.因此,在与本文框架结合时,进行了适当调整和简化.

1) 简化了子种群的分离判断过程.原SOS算法检测子种群分离的条件是:在同时满足种群收敛半径条件(收敛半径、种群大小、相对最优个体的适应度)的多个簇中,选择密度最大的作为分离子种群,而本文仅在主种群严格收敛至一定条件时才对其进行分离.实际上,原SOS方法中,其他种群簇在经过进一步进化之后,最终还是收敛至一个当前最优的高密度种群中,而这正是本文算法所要检测的分离子种群.这样设计避免了对主种群反复地分簇计算,从而降低了算法的复杂度.

2) 略去了原算法中子种群规模的动态调整.原算法通过额外引入两个算法参数,使子种群规模与其最优解适应度成线性关系逐渐变小.实际上,子种群经过移动和收敛半径的收缩,其外围的一些个体也会脱离种群,从而使子种群规模逐渐减小.鉴于此,本文不再专门对子种群规模进行调整.

### 1.1.1 子种群的动态生成

本文基于种群分离<sup>[14-15]</sup>的思想, 根据协进化种群在搜索过程中的分布情况, 动态生成多个子种群, 达到同时对适值空间中多个局部最优点的监控。

对于一个协进化种群(总个体数  $n_{\text{total}}$ ), 它包含一个主种群  $P_0$  和若干子种群  $P_i, i = 1, 2, \dots, N_{\text{co}}$ . 当适值空间内一个全局或局部的最优解被发现(即主种群按照一定标准收敛到一个较小适值区域)后, 一个子种群从主种群中分离, 对该解进行“监测”. 子种群的分离必须同时满足以下条件:

1) 主种群  $P_0$  的收敛半径  $r_0$  (最优个体距离其他个体的最大哈明距离) 小于子种群的最大半径  $r_{\text{max}}$ ;

2) 新生成的子种群  $P_i (i = 1, 2, \dots, N_{\text{co}})$  的个体数量为  $n_i = n_{\text{init}}$ , 其中  $n_{\text{init}}$  为子种群的初始个体数量, 并且子种群在后续算法过程中, 始终满足种群规模的约束, 即  $n_{s \text{ min}} \leq n_i \leq n_{s \text{ max}}$ , 其中  $n_{s \text{ min}}$  和  $n_{s \text{ max}}$  分别为子种群的最小、最大种群数;

3) 主种群在子种群分离后, 其个体数量

$$n_0 = n_{\text{total}} - \sum_i^{N_{\text{co}}} n_i \leq n_{0 \text{ min}},$$

其中  $n_{0 \text{ min}}$  为主种群的最小种群规模。

子种群分离后, 主种群将重新初始化, 随机散布在剩余的适值空间, 继续对新的最优解进行搜索. 这里, 主种群和子种群都采用常规的遗传算法进行搜索。

### 1.1.2 子种群搜索空间的收缩

子种群  $P_i$  的搜索半径  $r_i$  将随着其生成之后的进化代数  $g_i$ , 按照收缩因子  $f_r$  逐渐减小, 即

$$r_i(t+1) = r_{\text{min}} + (r_i(t) - r_{\text{min}})^{-g_i/f_r}. \quad (1)$$

$r_i$  的初始值为子种群的最大搜索半径  $r_{\text{max}}$ , 并且在搜索过程中始终不小于  $r_{\text{min}}$ , 因此有  $r_{\text{min}} \leq r_i \leq r_{\text{max}}$ .

在子种群的搜索范围即种群个体的分布范围收缩后, 子种群中的某些个体会位于该范围之外. 此时, 将这些个体移至“回收站”, 另行处理. 同时, 在子种群进行交叉、变异等遗传操作时, 要保证后代个体仍处于搜索范围之内. 若后代处于范围之外, 则重新执行遗传操作, 直到满足要求。

### 1.1.3 子种群的合并

由于子总群的中心由其内部最优个体表示, 在搜索过程中, 子种群的搜索中心  $c_i (i = 1, 2, \dots, N_{\text{co}})$  会随着最优解的变化而进行调整, 导致子种群搜索范围的移动. 当两个子种群的搜索范围接近时, 其搜索空间可以发生重叠; 但是, 当一个子种群的中心落入另一个子种群的搜索范围时, 两个子种群将合并。

对于两个准备合并的子种群  $P_1$  和  $P_2$ , 适应度较小的一方将被另一方“吞并”. 具体而言, 假设  $P_1$  最优个体的适应度优于  $P_2$ , 以二维适值空间为例, 新的搜索半径可计算如下:

$$r_1 = \min(\sqrt{r_1^2 + r_2^2}, r_{\text{max}}). \quad (2)$$

在搜索空间合并之后,  $P_2$  中的所有个体将被转移, 其中距  $P_1$  搜索中心距离小于  $r_1$  的个体将被整合入  $P_1$ , 并满足子种群规模  $[P_{s \text{ min}}, P_{s \text{ max}}]$  和半径  $[r_{\text{min}}, r_{\text{max}}]$  的约束, 同时将合并过程中不满足要求和约束的个体移至“回收站”。

### 1.1.4 “回收站”的处理

对于子种群而言, 在搜索范围的移动、收缩、合并等过程中, 一些个体会位于搜索空间之外, 那么这些个体将被删除至“回收站”. 需要注意的是, 当子种群的规模达到最小值  $P_{s \text{ min}}$  时, 落入搜索空间外的个体将直接重新初始化, 并使其位于搜索空间内。

对于主种群  $P_0$  而言, 当其个体落入到子种群搜索空间中时, 这些个体也将被移至“回收站”中统一处理. 因此, “回收站”是个体的临时存放处, 在每代进化的最后都需要对“回收站”中的个体进行处理和清空. 具体过程如下: 将“回收站”中的个体用随机产生的个体替代, 并插入主种群中, 以补充主种群对新的最优解的持续搜索能力. 需要注意的是, 随机产生的新个体不允许位于已有子种群的搜索空间内, 若新个体位于某一子种群搜索范围内, 则重新生成。

综上所述, 本文的动态多种群策略算法的具体步骤如下。

**Step 1:** 分离条件检测. 若  $r_0 < r_{\text{max}}$  且

$$n_0 = n_{\text{total}} - \sum_i^{N_{\text{co}}} n_i \leq n_{0 \text{ min}},$$

则转 **Step 2**, 否则转 **Step 3**.

**Step 2:** 将  $P_0(t)$  中适应度最高的  $p_{\text{init}}$  个体分离出来, 形成新的子种群,  $N_{\text{co}} := N_{\text{co}} + 1$ , 并重新初始化  $P_0(t)$ , 保证  $P_0(t)$  的个体不在  $P_i(t) (i = 1, 2, \dots, N_{\text{co}})$  的搜索范围内。

**Step 3:** 对  $P_0(t)$  和  $P_i(t)$  进行交叉、变异等遗传操作,  $t := t + 1$ .

**Step 4:** 更新  $c_i(t)$ .

**Step 5:** 更新  $r_0(t)$  和  $r_i(t)$ . 其中  $r_0(t)$  为主种群的搜索中心  $c_0(t)$  距最远个体的哈明距离,  $r_i(t+1) = r_{\text{min}} + (r_i(t) - r_{\text{min}})^{-g_i/f_r}$ .

**Step 6:** 对所有子种群  $P_i(t)$  进行合并条件检测. 若当某子种群的中心落入另一个子种群的搜索范围

时,发生合并,则执行 Step 7,否则直接转 Step 8.

Step 7: 对检测出需合并的子种群进行合并.

Step 8: 释放“回收站”内的临时个体,将其重新随机生成,重新加入主种群.

Step 9: 输出  $c_i(t)$ .

## 1.2 基于动态多种群策略的 CCEA 实现

下面将详细描述基于动态多种群策略的 CCEA (mCCEA) 实现方法. 将动态多种群策略引入 CCEA, 不仅涉及到协进化种群搜索算法的调整, 还影响到种群间的交互机制以及个体评价过程.

对于每个协进化种群  $\text{coevo } P_j = P_0 \cup P_{ij}$ ,  $i = 1, 2, \dots, N_{\text{co}}$ ,  $j = 1, 2, \dots, N$ ,  $N$  为优化问题的维度, 最大值优化问题的 mCCEA 的具体步骤如下.

Step 1: 对协进化种群  $\text{coevo } P_j$  ( $j = 1, 2, \dots, N$ ) 进行随机初始化, 交互间隔计数  $n_c := 0$ , 子种群数  $N_{\text{co}}(j) := 0$ ,  $t := 0$ .

Step 2: 如果接收到其他协进化种群  $\text{coevo } P_k$  ( $j \in N, j \neq k$ ) 所发送的交互  $R_k = \{c_{ik}(t)\}$ ,  $i = 1, 2, \dots, N_{\text{co}}(k)$ , 则转 Step 3, 否则转 Step 4, 其中  $c_{ik}$  为  $\text{coevo } P_k$  的子种群中心个体.

Step 3: 对于协进化种群  $\text{coevo } P_j$  中的每个个体, 分别与  $R_k$  进行组合, 构成优化问题的完整解并进行适应度评价, 将其中的最大值作为其适应度.

Step 4: 利用动态多种群策略进化一代, 得到  $c_{ij}(t)$ ,  $i = 1, 2, \dots, N_{\text{co}}$ .

Step 5:  $n_c := n_c + 1$ ; 若  $n_c = N_\delta$ ,  $N_\delta$  为设定的交互发送间隔, 则  $n_c := 0$ , 转 Step 6, 否则转 Step 8.

Step 6: 若  $\text{coevo } P_j$  的子种群数  $N_{\text{co}} = 0$ , 则交互

$$S_j := \text{BestIndividuals}(\text{coevo } P_j(t));$$

否则,  $S_j := \{c_{ij}(t)\}$ ,  $i = 1, 2, \dots, N_{\text{co}}$ ,  $j = 1, 2, \dots, N$ .

Step 7: 将交互  $R$  发送给其他协进化种群  $\text{coevo } P_k$ .

Step 8: 若  $t = t_{\text{max}}$  (最大进化代数), 则结束并输出协进化种群  $\text{coevo } P_j$  的最优解

$$\text{BI}_j := \text{BestIndividual}(\text{coevo } P_j(t_{\text{max}})),$$

整个问题的最优解为  $\{\text{BI}_j\}$ ,  $j = 1, 2, \dots, N$ , 适应度为  $F(\{\text{BI}_j\})$ , 否则转 Step 2.

由上述算法步骤可以看出, mCCEA 相对于传统 CCEA 的主要区别在于:

1) 用动态多种群策略代替传统进化算法, 作为每个协进化种群的搜索算法;

2) 协进化种群间的交互内容不再像传统

CCEA 那样, 仅从适应度的角度考虑, 而是从适值空间分布特性的角度, 将当前动态适值空间中全局或局部最优个体信息作为协进化种群间的交互信息. 因此, 协进化种群之间能够彼此相互了解对方当前的全局或局部最优解. 这样的交互内容在相同的通信量下, 能够传递更多的协进化信息, 有利于准确评价当前个体的适应度. 从信息效用的角度看, 这实际上进行了信息补偿.

## 1.3 算法复杂度分析

基于动态多种群策略的 CCEA 复杂度分析如下:

1) 初始化种群, 并构建协进化种群的时间复杂度为  $O\left(\sum_{i=1}^N n_i + n_o\right)$ ;

2) 在存在交互  $R_k$  的情况下, 对协进化种群进行适应度评估, 时间复杂度为  $O\left(2\left(\sum_{i=1}^N n_i + n_o\right)\right)$ ;

3) 种群进行条件分离的时间复杂度为  $O(3n_o)$ ;

4) 以交叉概率  $P_c$ 、变异概率  $P_m$  对种群  $P_0$  和  $P_j$  中的个体进行交叉和变异, 产生下一代种群的时间复杂度为

$$O\left((P_c + P_m)\left(\sum_{i=1}^N n_i + n_o\right)\right);$$

5) 更新  $c_i(t)$ , 并计算种群个体的哈密距离, 其时间复杂度为  $O\left(N + \sum_{i=1}^N n_i + n_o\right)$ ;

6) 子种群合并的时间复杂度为  $O\left(2\sum_{i=1}^N n_i\right)$ ;

7) 回收站处理的时间复杂度为  $O(n_o)$ ;

8) 在交互隔代处理下, 将对动态多种群的构建产生  $1/N_\delta$  的时间平均.

由于  $\sum_{i=1}^N n_i + n_o = n_{\text{total}}$ , 可得算法在最差情况下的时间复杂度如下:

$$\begin{aligned} & O\left(n_{\text{total}} + 2n_{\text{total}} + (P_c + P_m)\left(\sum_{i=1}^N n_i + n_o\right) + \right. \\ & \left. \frac{1}{N_\delta}\left(3n_o + N + n_{\text{total}} + 2\sum_{i=1}^N n_i + n_o\right)\right) = \\ & O\left(3n_{\text{total}} + \frac{1}{N_\delta}(3n_{\text{total}} + 2n_o + N)\right). \end{aligned} \quad (3)$$

## 2 实验及分析

### 2.1 用作对比的 CCEA

为了与 mCCEA 进行对比实验, 这里使用了 3 个具有代表性的 CCEA 作为对比算法.

1) 传统 CCEA<sup>[7]</sup> (tCCEA).

tCCEA 作为传统 CCEA 的代表, 每次交互时, 一

个协种群向其他协种群发送  $K$  个个体. 在这  $K$  个个体中, 有一个是当前种群中的最优解, 剩余  $K - 1$  个个体随机从当前种群中选出.

### 2) 诱导 CCEA<sup>[6]</sup> (bCCEA).

bCCEA 在 tCCEA 的基础上, 通过对最佳协作解的启发式估计来修正个体适应度的评估, 以达到对 CCEA 的诱导. 交互内容为当前子种群中适应度最高的两个解.

### 3) 完全交互 CCEA<sup>[18]</sup> (cCCEA).

cCCEA 实际上是 tCCEA 的一种极端情况. 当 tCCEA 中的参数  $K$  为协进化种群的个体数量时, tCCEA 成为 cCCEA, 即 cCCEA 中的交互内容为整个协进化种群.

## 2.2 针对病态行为的测试函数

本文主要针对 CCEA 最重要的病态现象——“相对过分概括”, 选用极易引发传统 CCEA “相对过分概括”病态现象的最大双二次方程 (MTQ) 函数. 该函数已经被国内外学者广泛接受<sup>[6-7, 16]</sup>, 作为检验 CCEA 抗病态能力的标准测试问题.

考察如下 MTQ 问题:

$$\text{MTQ}(x, y) \leftarrow \max \begin{cases} H_1 \times \left(1 - \frac{16 \times (x - X_1)^2}{S_1} - \frac{16 \times (y - Y_1)^2}{S_1}\right), \\ H_2 \times \left(1 - \frac{16 \times (x - X_2)^2}{S_2} - \frac{16 \times (y - Y_2)^2}{S_2}\right). \end{cases} \quad (4)$$

其中:  $x$  和  $y$  为在  $[0, 1]$  区间的决策变量;  $H_1$  和  $H_2$  影响两个峰的高度;  $S_1$  和  $S_2$  影响两个峰所在适值空间覆盖的区域, 值越大峰所覆盖的区域越大;  $X_1, Y_1, X_2$  和  $Y_2$  影响两个峰在适值空间中的位置, 进而影响两个峰的相关性. 由  $\{X_2, Y_2, H_2, S_2\}$  所决定的 2 号峰为全局最高峰, 由  $\{X_1, Y_1, H_1, S_1\}$  所决定的 1 号峰为干扰, 挑战 CCEA 的全局最优搜索能力. 在下面的实验组, 本文使用文献 [18] 中的两组设置, 得到以下两个 MTQ 问题.

1) MTQ 50:  $H_1 = 50, H_2 = 150, S_1 = 16/10, S_2 = 1/32, X_1 = 3/4, Y_1 = 3/4, X_2 = 1/4, Y_2 = 1/4$ , 最优解为  $\text{MTQ}(1/4, 1/4) = 150$ .

2) MTQ 125:  $H_1 = 125, H_2 = 150, S_1 = 16/10, S_2 = 1/32, X_1 = 3/4, Y_1 = 3/4, X_2 = 1/4, Y_2 = 1/4$ , 最优解为  $\text{MTQ}(1/4, 1/4) = 150$ .

对于 CCEA 而言, “相对过分概括”现象是指: 种群更加倾向于收敛至具有较大覆盖区域的峰, 尽管这个峰有可能是次优的. 因此, 当干扰峰 (1 号峰) 的覆盖

区域较大时, 便容易引起 CCEA 的“相对过分概括”现象, 而且两个峰的高度越接近, 给算法带来的迷惑性越强. 由此可见, MTQ 125 相对于 MTQ 50 的问题难度更大.

## 2.3 实验参数设置

在以下各组实验中, 各算法的协进化种群规模  $n_{\text{total}}$  均为 50; 选择方式为竞争选择, 竞争数为 2; 变异操作均采用随机选择性变异, 变异概率为 0.1; 交叉操作为模拟二进制交叉, 其参数基因交换概率和多项式阶数分别为 0.5 和 5, 交叉操作概率为 0.8; 交互间隔  $N_\delta = 10$ , 每次运行进化代数  $t_{\text{max}}$  为 1000, 实验次数为 50 次. tCCEA 的交互个体数  $K = 4$ , mCCEA 的具体参数设置如表 1 所示.

表 1 mCCEA 的仿真参数设置

参数名称	参数值
主种群最小规模 $n_{0 \min}$	10
子种群最大规模 $n_{s \max}$	20
子种群最小规模 $n_{s \min}$	4
子种群初始规模 $n_{\text{init}}$	10
子种群最大搜索半径 $r_{\max}$	0.2
子种群最小搜索半径 $r_{\min}$	0.1
子种群搜索半径收缩因子 $f_r$	2.0

## 2.4 实验结果及分析

### 1) 全局最优收敛率比较.

表 2 给出了在利用 mCCEA, tCCEA, bCCEA 和 cCCEA 求解 MTQ 50 和 MTQ 125 问题时, 全局最优收敛率的比较结果 (最终解和真正全局最优解的欧式距离小于 0.1 即认为收敛).

表 2 全局收敛率对比结果 %

问题	mCCEA	tCCEA	bCCEA	cCCEA
MTQ 50	98	8	74	68
MTQ 125	98	0	44	44

由表 2 可以看出: tCCEA 在处理 MTQ 问题时表现出了严重的相对过分概括病态行为; bCCEA 得益于启发式适应度估计, 相对 tCCEA 的收敛率有了大幅改善, 但由于其估计方法仍无法实现对最佳合作解的准确估计, 导致收敛率仍不理想; 同样, 尽管 cCCEA 做到了穷尽式交互, 但由于其所采用的传统进化算法无法同时搜索动态适值空间中的多个“可疑”区域, 造成了交互个体仍集中在某一高适值区域, 交互信息无法全面描述适值空间的准确情况, 仍然容易导致相对过分概括现象的发生.

相对上述 3 种算法, 本文的 mCCEA 由于采用了多种群进化搜索方法, 能够同时允许多个子种群的存在, 而这些子种群所“关注”的正是适值空间中的局部

或全局最优解,因此协进化种群间的交互信息能够更准确地反映各自进化的情况,达到了信息补偿的效果,从而有效抑制了相对过分概括现象的发生。

## 2) 动态性能比较.

下面将分别从目标空间和决策空间对各算法的动态性能进行比较和分析。

图1和图2给出了目标空间中,4种算法经过50次实验后的平均动态性能曲线.图2(a)中,bCCEA的动态性能曲线显示不完整,完整曲线如图2(b)所示.由图1和图2可以看出:tCCEA和cCCEA很快便收敛于局部最优解,发生相对过分概括现象,其适应度曲线保持平直,算法的搜索基本停滞;bCCEA由于其启发估计方法的诱导作用,在无序波动(见图3和图4)很长一段时间后才能估计出较准确的最佳合作解,诱导算法向正确的方向发展.相比之下,本文mCCEA的进化曲线在经过短暂调整后,便朝着正确的方向进化,对相对过分概括现象有很好的抑制作用。

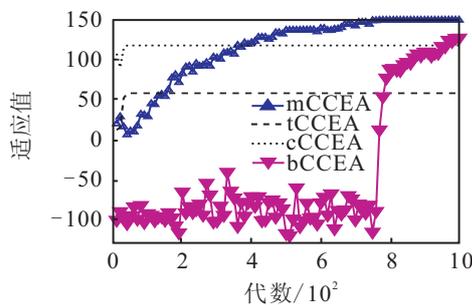
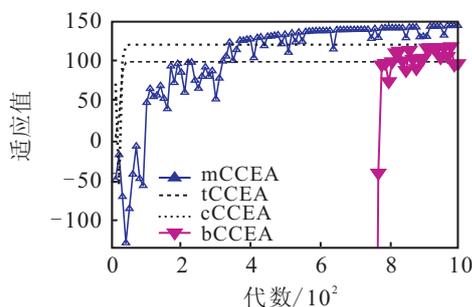
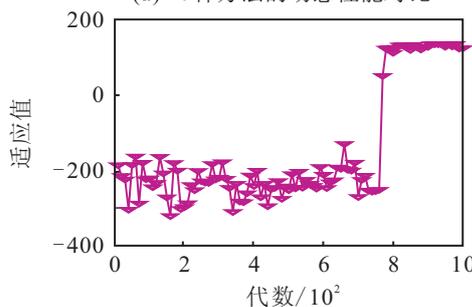


图1 针对MTQ 50问题的目标空间动态性能对比



(a) 4种方法的动态性能对比



(b) bCCEA的动态性能

图2 针对MTQ 125问题的目标空间动态性能对比

图3和图4给出了决策空间中,4种算法经过50次实验后的平均动态性能曲线。

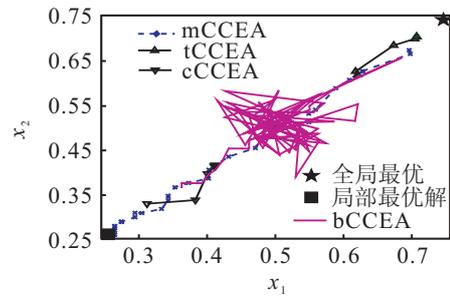


图3 针对MTQ 50问题的决策变量空间动态性能对比

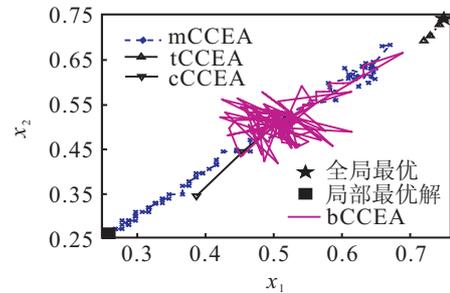


图4 针对MTQ 125问题的决策变量空间动态性能对比

由图3和图4可以看出:除了tCCEA朝着局部最优解方向搜索外,其他算法均朝着全局最优方向进行,说明tCCEA受相对过分概括病态行为的影响很明显,导致算法失败;cCCEA之所以能够朝着正确方向进行,要归功于巨大的交互信息量,然而它最终却无法到达全局最优点,这说明通过增加交互个体数量的方式,对信息补偿的效果是十分有限的,不足以确保算法收敛到全局最优;本文的mCCEA由于信息补偿措施有效,能够给算法提供更有效的协进化信息,使协进化种群正确地进行适应度评估,最终能够到达最优点,进一步验证了mCCEA的有效性和正确性。

## 3 结论

本文从信息丢失是引发CCEA病态现象的本质出发,针对CCEA协进化适值空间为动态变化的特点,将动态多种群策略与CCEA相结合,提出了基于动态多种群策略的CCEA,即mCCEA.该算法在动态适值空间搜索更能表征协进化种群状态的局部或全局最优解,将其作为交互信息,可起到信息补偿的作用,这样可提高算法抵抗如“相对过分概括”等病态现象的能力.在实验部分,针对MTQ 125和MTQ 50标准测试函数进行测试,并与tCCEA, bCCEA和cCCEA三种典型算法进行对比分析.实验结果表明,mCCEA能有效抑制相对过分概括现象的发生,大大提高了算法的全局最优收敛能力.此外,从目标空间和决策空间的动态性能来看,mCCEA相对其他3种算法能更好地保持正确的进化方向,最终获得全局最优解。

在未来工作中,将对围绕交互间隔 $N_\delta$ 和动态多种群策略的相关参数开展研究,深入分析参数对算法

的影响, 进一步提高算法性能. 此外, 将本文算法思想推广至其他动态多种群进化策略, 如分簇聚类法等<sup>[17-18]</sup>, 提出通用的动态多种群信息补偿框架, 这也是未来工作的重点.

### 参考文献(References)

- [1] Hoverstad B A, Moe H A, Shi M. Entropy and mutual information can improve fitness evaluation in coevolution of neural networks[C]. Proc of the 2009 IEEE Congress on Evolutionary Computation. Trondheim: IEEE, 2009: 3199-3206.
- [2] 陈羽, 滕弘飞. 变粒度协同进化设计算法及其在卫星舱布局设计中应用[J]. 大连理工大学学报, 2010, 6(50): 931-936.  
(Chen Y, Teng H F. Coevolutionary algorithm with coarse-to-fine grain strategy and its application to layout design of satellite module[J]. J of Dalian University of Technology, 2010, 6(50): 931-936.)
- [3] 王冲, 景宇, 李军, 等. 协同进化方法求解多中心卫星任务规划问题[J]. 航空学报, 2010, 31(9): 1832-1840.  
(Wang C, Jing Y, Li J, et al. Solving multi-center satellite mission scheduling problems by coevolutionary method[J]. Acta Aeronautica et Astronautica Sinica, 2010, 31(9): 1832-1840.)
- [4] Berger J, Happe J, Gagne C, et al. Co-evolutionary information gathering for a cooperative unmanned aerial vehicle team[C]. The 12th Int Conf on Information Fusion. Seattle, WA: IEEE, 2009: 347-354.
- [5] Wiegand R P. An analysis of cooperative coevolutionary algorithms[D]. Fairfax: Department of Computer Science, George Mason University, 2003.
- [6] Panait L, Luke S, Harrison J F. Archive-based cooperative coevolutionary algorithms[C]. Proc of the 8th Annual Conf on Genetic and Evolutionary Computation. Seattle: ACM, 2006: 345-352.
- [7] Panait L, Luke S, Wiegand R P. Biasing coevolutionary search for optimal multiagent behaviors[J]. IEEE Trans on Evolutionary Computation, 2006, 10(6): 629-645.
- [8] Ficici S G, Pollack J B. A game-theoretic memory mechanism for coevolution[C]. Genetic and Evolutionary Computation. Chicago: Springer Berlin Heidelberg, 2003: 286-297.
- [9] Ficici S G, Pollack J B. A game-theoretic approach to the simple coevolutionary algorithm[C]. Parallel Problem Solving from Nature PPSN VI. Paris: Springer Berlin Heidelberg, 2000: 467-476.
- [10] 罗杰, 段建民, 陈建新. 一种引入局部交互的群体协作为协同进化机制[J]. 机器人, 2007, 29(4): 313-319.  
(Luo J, Duan J M, Chen J X. A mechanism of cooperative coevolution with local interaction for collective cooperation behaviors[J]. Robot, 2007, 29(4): 313-319.)
- [11] 刘静. 协同进化算法及其应用研究[D]. 西安: 西安电子科技大学工程学院, 2004.  
(Liu J. Coevolutionary algorithms and their applications[D]. Xi'an: School of Electronic Engineering, XiDian University, 2004.)
- [12] 巩敦卫, 孙晓燕. 协同进化遗传算法理论及应用[M]. 北京: 科学出版社, 2009: 102-140.  
(Gong D W, Sun X Y. Coevolutionary genetic algorithms and their applications[M]. Beijing: Science Press, 2009: 102-140.)
- [13] Peng X, Gao X, Yang S. Environment identification-based memory scheme for estimation of distribution algorithms in dynamic environments[J]. Soft Computing, 2011, 15(2): 311-326.
- [14] Branke J. Evolutionary optimization in dynamic environments[M]. Norwell: Kluwer Academic Publishers, 2001: 53-98.
- [15] Tsutsui S, Fujimoto Y, Ghosh A. Forking genetic algorithms: GAs with search space division schemes[J]. Evolutionary Computation, 1997, 5(1): 61-80.
- [16] Bucci A, Pollack J B. On identifying global optima in cooperative coevolution[C]. Proc of the 2005 Conf on Genetic and Evolutionary Computation. Singapore: ACM, 2005: 539-544.
- [17] Blackwell T, Branke J. Multiswarms, exclusion, and anti-convergence in dynamic environments[J]. IEEE Trans on Evolutionary Computation, 2006, 10(4): 459-472.
- [18] Li C, Yang S. A general framework of multipopulation methods with clustering in undetectable dynamic environments[J]. IEEE Trans on Evolutionary Computation, 2012, 16(4): 556-577.

(责任编辑: 滕蓉)