

基于预测策略的动态多目标免疫优化算法

刘若辰 马亚娟 张 浪 尚荣华

(西安电子科技大学智能感知与图像理解教育部重点实验室 西安 710071)

摘 要 为了有效解决动态多目标优化问题,文中提出了一种新的基于预测策略的动态多目标免疫优化算法.该算法首先采用相似性检测算子较好地检测到环境的变化,同时利用前几个时刻的最优非支配抗体解集建立新的预测模型来预测产生新时刻的初始抗体种群,进一步提高了算法对环境变化的反应能力.此外,通过引入基于两种不同的父代个体选择策略而改进的差分交叉算子来加快算法的收敛速度.文中采用几个典型的标准测试问题验证算法的有效性,实验结果表明,提出的相似性检测算子的预测模型可以提高算法的跟踪能力,而改进的差分交叉算子能够提高算法的收敛性能.

关键词 预测模型;差分进化;动态多目标优化;免疫优化算法

中图法分类号 TP301

DOI号 10.11897/SP.J.1016.2015.01544

Dynamic Multi-Objective Immune Optimization Algorithm Based on Prediction Strategy

LIU Ruo-Chen MA Ya-Juan ZHANG Lang SHANG Rong-Hua

(Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China,
Xidian University, Xi'an 710071)

Abstract In this paper, a new dynamic multi-objective immune optimization algorithm based on prediction strategy is proposed for solving dynamic multi-objective optimization problems effectively. Firstly a similarity detection operation is used to detect the environment change. Then, a new forecasting model, which is established by the non-dominated antibodies in previous optimum locations, is adopted to generate the initial antibody population in order to improve the ability of responding to the environment change. Moreover, an improved differential evolution crossover operator based on two different selection strategies is introduced to speed the convergence of algorithm. The proposed algorithm is validated on several benchmark testing problems, the experimental result shows that the forecasting model based on the similarity detection operation can improve the tracking ability and the improved differential crossover operation can enhance the convergence.

Keywords forecasting model; differential evolution; dynamic multi-objective optimization; Immune optimization algorithm

1 引 言

现实中的许多功能系统都是随时间或环境的变

化而变化的,在不同时刻或环境下,系统会表现不同的特性.近年来,越来越多的学者开始关注动态单目标优化问题^[1-3].文献[4]采用了多目标优化的方法来处理动态环境中单目标优化问题.虽然也有学者

收稿日期:2013-02-18;最终修改稿收到日期:2014-10-27.本课题得到国家自然科学基金(61373111,61003199,61371201)、中央高校基本科研基金(K50511020014,JB150227)和陕西省自然科学基金(2014JM8321)资助.刘若辰,女,1974年生,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为免疫计算、多目标优化以及数据挖掘.E-mail: ruochenliu@xidian.edu.cn.马亚娟,女,1981年生,硕士,主要研究方向为多目标优化和数据挖掘.张浪,男,1983年生,硕士,主要研究方向为多目标优化.尚荣华,女,1979年,博士,副教授,目前主要从事智能信息处理、模式识别和数据发掘等方面的研究.

开始关注动态多目标优化(Dynamic Multi-objective Optimization, DMO)问题^[5-7], 但是到目前为止, 相较于动态单目标优化, 关于解决 DMO 问题的有效的研究成果并不多。

动态正交多目标进化的算法(DOMOEA)^[8]采用一种正交设计方法, 来提高在两个连续的环境变化中, 进化种群对变化的适应能力. Goh 和 Tan^[9]提出了一种新的协同进化的模式, 即竞争与合作机制相结合, 来处理静态多目标优化问题, 同时该方法被扩展到处理动态多目标优化问题. 该算法通过一种竞争与合作的相互迭代过程实现在特别时刻的优化需求下不同的子种群能够优化各个子目标, 进而能有效处理静态和动态多目标优化问题. Wang 等人^[10]提出了一种新的动态多目标算法, 在多目标问题的静态优化过程中, 首先, 将静态多目标优化问题转变成一个两目标优化问题, 然后, 设计了一种新的适用于环境变化的交叉和变异算子来提高算法优化多目标问题的能力. Koo 等人^[11]提出了一种应用于快速环境变化的动态多目标梯度算法(dMO-EGS), 在该算法中, 作者提出一种新的预测策略, 这种预测策略不仅实现起来比较简单, 而且能够使得产生的新时刻的解分布在 Pareto 最优解附近, 进而改善了种群的收敛性。

在动态多目标优化问题中, 目标函数、约束条件函数和相关参数都可能会随着时间的变化而变化, 真实 Pareto 前沿(Pareto Front, PF)和真实 Pareto 解集(Pareto Set, PS)的位置也会随着时间的变化而移动. 因此动态多目标优化的目标是, 在给定的计算代价内, 能有效地追踪 PF 和 PS 的位置移动. 这种动态特性要求, 在下次环境变化之前, 算法要快速找到当前时刻的最优解集. 如果将静态多目标优化算法直接用来解决动态多目标优化问题会有很多的局限性, 例如, 当环境变化时, 静态多目标优化算法不能及时地对环境变化做出反应, 不能适应新的环境, 因而不能很好地追踪 PF 和 PS 的位置变化. 动态多目标优化中有两个不同于静态多目标优化的关键问题, 即如何检测环境变化以及如何应对环境变化. 对于如何检测环境变化, 目前常用的方法有两种, 一种方法是直接利用上一时刻得到的 Pareto 最优解集来计算新时刻的目标函数值, 如果目标函数值发生变化, 则认为环境发生了改变; 另一种做法类似文献[12]中的方法, 从上一时刻的最优 Pareto 解集中选出一部分个体, 计算它们在新时刻的目标函

数值, 如果目标函数值发生变化, 则认为环境发生了改变. 对于如何应对环境变化, 主要在 3 个方面来应对: (1) 在种群的历史信息保存方面; (2) 在参数的自适应方面; (3) 在种群的重新初始化方面. 目前, 第 3 个方面的研究, 即在检测到环境变化时如何重新初始化种群, 是研究的重点方向. 当未检测到环境变化时, 一般会直接利用上一时刻的 Pareto 最优解集作为新时刻的初始种群或者选择一部分 Pareto 最优解集进行高斯变异来作为新时刻的初始种群. 而当检测到环境变化时, 最简单最传统的方法是随机产生新时刻的种群, 并重新开始静态多目标优化过程. Deb 等人^[13]对 NSGA2 进行了改进并将其应用于在线优化动态水热电力调度问题, 该文中使用了两种初始化方法, 一种是将前一时刻得到的最优解集的一部分重新随机初始化, 另一种是对前一时刻 Pareto 最优解集的一部分进行高斯变异. 这种基于历史 Pareto 最优解集来产生新时刻初始种群的方法, 也称为基于记忆的方法. Wang 等人^[5]提出了几种基于记忆的动态多目标优化算法, 实验结果表明基于记忆的策略对动态多目标优化问题是有效的. 然而这种方法没有考虑环境变化的特点, 如时间序列中一些特殊点或者是环境变化历史产生的 PS 位置之间的相互关系, 因而算法对环境变化的反应能力是有限的. Hatzakis 等人^[14]提出了基于历史信息的前向预测方法, 这种方法只用来预测新时刻的两个边界值点. 而 Zhou 等人^[12]用一种线性的预测模型来预测新时刻的全部初始解. 本文中, 我们利用历史最优 Pareto 解集, 使用预测模型^[12]来指导下一时刻种群的搜索. 我们的方法和文献[12]的不同之处在于算法中用来检测环境是否发生变化的相似性检测(Similarity Detection)算子. 文献[12]是通过重新计算上一时刻的一些最优解在新时刻的函数值, 来判断环境是否发生了改变, 但是这种方法要求环境中没有噪声干扰, 而本文使用这些函数值的统计平均信息来检测环境变化, 能降低噪声的干扰. 此外, 当检测到环境变化时, 在环境变化的初期, 由于历史信息太少不能形成有效的预测模型, 我们通过从前一时刻的 Pareto 最优解集进行高斯变异来得到新时刻的初始解. 在环境变化的后期, 由于历史信息丰富, 可以根据历史最优 PS 解集的位置来有效地预测新时刻的最优 PS 位置, 从而产生新时刻初始解的位置. 实验结果表明, 预测机制可以有效地提高算法对环境变化的反应能力以及收敛速度。

近年来,将免疫系统应用到动态优化问题上的方法逐渐引起了人们的关注,这是因为免疫系统有着对新威胁快速反应的自然特性以及免疫记忆在遇到抗原时有快速识别的能力. Shang 等人^[15]提出了一种基于非一致性变异算子的克隆选择算法(ICADMO)来处理 DMO 问题. 非一致性变异算子将变异算子与进化代数相联系,在进化的早期,种群在一个较大范围内搜索以避免陷入局部最优,在进化的后期,进行局部搜索来帮助种群快速收敛. Zhang 等人^[16-17]提出了基于免疫系统的优化算法来解决决策空间或者目标空间的维数可能随时间改变的 DMO 问题,而且算法已经分别成功地应用于温室控制和信号仿真中. Zhang 等人^[18]提出了一种基于人工免疫系统的算法,来处理决策变量维数随时间变化的非线性约束多目标问题. 在这个算法中, T 模块利用历史信息来产生新环境的初始种群, B 模块用来搜索一个给定的环境下的 Pareto 最优解, M 模块用来临时存储非支配解来帮助产生下一个环境到来后的初始候选解. 基于非支配邻域选择的静态多目标免疫优化算法(NNIA)^[19], 经过系统的比较和分析,发现对于静态多目标优化有很好的效果, 本文将其改进,并扩展用于动态多目标优化问题. 此外,我们加入了一个改进的差分交叉算子来提高算法的性能. 不同于经典的差分进化算法,为了提高种群的多样性和收敛速度,避免算法陷入局部最优,在改进的差分交叉算子中,我们提出了两种选择父代抗体的策略来产生新的个体. 实验结果表明,改进的差分交叉算子进一步提高了算法的收敛性能.

2 背景知识

2.1 动态多目标优化的定义

本文中,我们处理如下的动态多目标优化问题^[20]:

$$\begin{cases} \min F(x, t) = (f_1(x, t), f_2(x, t), \dots, f_m(x, t))^T \\ \text{s. t. } x \in X \end{cases} \quad (1)$$

其中, $t=0, 1, 2, \dots$ 表示时间, $x \in R^D$ 表示决策空间, $x=(x_1, \dots, x_t) \in R^D$ 表示决策变量. $F: (X, t) \rightarrow R^m$ 组成了 m 个随着时间变化的实值目标函数 $f_i(x, t)$, $i=1, 2, \dots, m$, R^m 代表目标空间.

Farina 等人^[20]根据环境变化对最优 PS 位置和最优 PF 位置的影响,把 DMO 问题分为 4 种类型:

第 1 类. 随着环境变化,最优 PS 位置发生改变,最优 PF 位置不改变.

第 2 类. 随着环境变化,最优 PS 位置和最优 PF 位置都发生改变.

第 3 类. 随着环境变化,最优 PS 位置不改变,而最优 PF 位置发生改变.

第 4 类. 随着环境变化,最优 PS 位置和最优 PF 位置都不改变,但是优化的问题却在改变.

虽然第 4 类情况在特定的情况下也会发生,但它通常被认为是平稳的或者静态的多目标优化问题,用一般的静态多目标优化算法就可以解决. 在本文中,我们着重研究前 3 类常见问题.

2.2 预测模型

当相似性检测算子检测到环境变化时,怎样利用种群的历史信息来预测新环境的最优 PS 的位置是所有预测模型的研究重心.

本文中使用的预测模型^[12]工作原理如下:

假定当环境发生变化时,历史时刻记录的抗体 $Q_1, \dots, Q_t$ 可以为预测 $(t+1)$ 时刻的最优 PS 提供信息,则在某种意义上, $(t+1)$ 时刻的最优 PS 与历史时刻的抗体位置 $Q_1, \dots, Q_t$ 存在函数关系:

$$Q_{t+1} = F(Q_1, \dots, Q_t, t) \quad (2)$$

其中 Q_{t+1} 表示 $(t+1)$ 时刻的新的 PS 的位置.

实际上,很多时间预测方法都可以用来估计函数 $F(\cdot)$, 假定 $x_1, \dots, x_t, x_i \in Q_i, i=1, \dots, t$ 是一组可以表示最优 PS 移动动向的抗体,那么用来预测 $(t+1)$ 时刻抗体初始位置的通用预测模型可以表示为

$$x_{t+1} = F(x_t, x_{t-1}, \dots, x_{t-K+1}, t) \quad (3)$$

式(3)表明,预测 $(t+1)$ 时刻的一个新解 x_{t+1} 时需要用到 $(t+1)$ 时刻之前 K 个时刻的最优解 $(x_t, x_{t-1}, \dots, x_{t-K+1})$. 其中当 $K=3$ 时,预测模型的工作机制如图 1 所示.

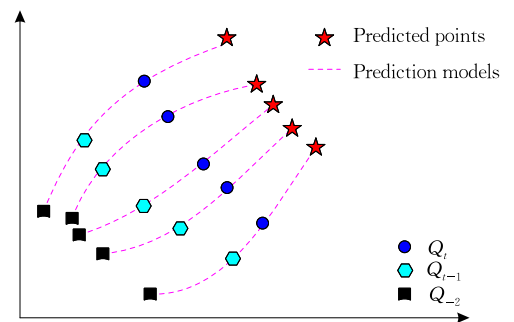


图 1 预测模型的示意

一个启发式算法被用来识别储存抗体 $Q_1, \dots, Q_t$ 之间的关系. 对于 t 时刻的存储抗体 Q_t 中的某个抗体 x_t , 它在上个时刻, 即 $(t-1)$ 时刻的父代抗体 x_{t-1} 可以被定义为存储抗体 Q_{t-1} 中与其最相似的抗体,

$$x_{t-1} = \arg \min_{y \in Q_{t-1}} \|y - x_t\|_2 \quad (4)$$

如果在种群中对于每一个抗体 x_t 来说, 其与上个时刻的最相似抗体 x_{t-1} 构成时间序列后, 可以采用一个简单的线性模型来预测下个时刻的抗体:

$$x_{t+1} = F(x_t, x_{t-1}) = x_t + (x_t - x_{t-1}) \quad (5)$$

然后对预测位置添加一个高斯噪声 $\epsilon \sim N(0, \mathbf{I}\delta)$, 其中 $\mathbf{I}$ 是单位矩阵, δ 是标准差, 那么通过之前构成的时间序列的均方误差就可以得到

$$\delta^2 = \frac{1}{4n} \|x_t - x_{t-1}\|_2^2 \quad (6)$$

其中 n 表示决策向量的维数.

2.3 差分进化

在连续区间的优化领域中, 差分进化(Differential Evolution, DE)算法^[21]是一种简单有效的进化算法.

DE 的主要过程是: 首先在决策空间随机初始化种群 $X_0 = (X_{1,0}, X_{2,0}, \dots, X_{N,0})$, 其中 N 为种群规模, 然后利用种群间的个体差异对另外的个体进行扰动实现变异操作, 接着进行交叉操作产生中间种群, 当中间种群的个体适应度优于其对应的父代个体时, 中间个体就被保留下来进入种群, 否则, 父代个体进入种群. 设 $X_{r_1,t}, X_{r_2,t}, X_{r_3,t}$ 是从 t 代中随机选取的互不相同的 3 个个体, 经差分变异后得变异个体 $V_{i,t+1}$, 即

$$V_{i,t+1} = X_{r_1,t} + F \times (X_{r_2,t} - X_{r_3,t}) \quad (7)$$

其中, $F \in [0, 2]$ 是缩放因子.

将变异个体的变量与种群中另外的父代个体变量按一定规则混合来产生新个体, 该操作即为 DE 算法的交叉操作, 设 D 是实变量的维数, 则 DE 交叉操作可以表示为

$$U_{i,t+1} = (U_{1,t+1}, U_{2,t+1}, \dots, U_{N,t+1}),$$

$$U_{ij,t+1} = \begin{cases} V_{ij,t+1}, & (\text{rand}(j) \leq CR) \text{ or } j = j_{\text{rand}} \\ X_{ij,t}, & (\text{rand}(j) > CR) \text{ or } j \neq j_{\text{rand}} \end{cases},$$

$$i = 1, 2, \dots, N, j = 1, 2, \dots, D \quad (8)$$

其中 $\text{rand}(j)$ 是 $[0, 1]$ 之间的一个均匀分布随机数, j_{rand} 是随机产生一个 $[1, D]$ 之间的自然数, $CR \in [0, 1]$ 是变异概率. 本文在提出的算法中引入改进的差分交叉算子来改进算法的性能.

3 基于预测策略的动态多目标免疫优化算法

本文提出的算法的主要思想是: 在新的时间环境刚到来之后, 和处理动态多目标问题的一般策略不同, 相似性检测被用来检测环境是否发生变化, 然后利用提出的预测机制来产生新时刻的初始抗体, 最后基于改进的差分交叉算子的 NNIA 算法来指导种群进化. 下面分别详细介绍算法的各个步骤、具体流程以及改进的算子.

3.1 相似性检测和预测机制

在动态优化过程中, 种群需要及时知道环境是否发生了变化, 以此来实施应对策略. 相似性检测(Similarity Detection)是用来判断环境是否发生改变的策略. 环境检测通常有两种方法, 其中一种方法是文献[12-13]中的方法: 从父代种群中随机选择一部分解, 并且对其进行评价, 如果任何一个目标函数或者约束条件发生变化, 就认为问题发生了变化. 虽然这种方法简单、易于操作, 但是它在函数评价时对环境中存在的噪声敏感, 容易造成误判. 因此, 本文中, 我们使用另一种更稳定的方法: 基于种群统计信息的相似性检测算子^[20], 其表达式如下:

$$\epsilon(t) = \frac{\sum_{j=1}^{N_\delta} \left\| \frac{(f_j(X, t) - f_j(X, t-1))}{\mathbf{R}(t) - \mathbf{U}(t)} \right\|}{N_\delta} \quad (9)$$

其中, $\mathbf{R}(t) = (r_1, r_2, \dots, r_m)^T$ 是 t 时刻种群在各个目标函数取到的最大值向量, m 为目标函数的个数, r_i 对应着在 t 时刻第 i 个目标函数在种群中取到的最大值; 同样地, $\mathbf{U}(t) = (u_1, u_2, \dots, u_m)^T$ 是 t 时刻种群在各个目标函数取到的最小值向量, u_i 对应着在 t 时刻第 i 个目标函数在种群中取到的最小值. N_δ 是检测的目标种群的规模, 当得到的相似值 $\epsilon(t)$ 大于给定的阈值 δ , 即 $\epsilon(t) > \delta$ 时, 我们认为系统中发生了显著的变化, 利用时间序列中的历史时刻中最优 PS 的位置来预测新时刻最优 PS 的位置, 然后在预测的新位置的周围生成新时刻的初始解.

当检测到环境变化时, 基于预测机制的重新初始化策略可以描述为如下.

算法 1. 基于预测机制的重新初始化策略.

输入: 前两个时刻得到的非支配种群 $Q_{t-1}(G_{\max_t})$,

$Q_{t-2}(G_{\max_t})$

输出: t 时刻的初始种群 $Q_t(0)$

1. 从 $Q_{-1}(G_{\max_t})$ 中随机选择 5 个个体作为检测种群, 计算出相似性检测算子 $\varepsilon(t)$ 的值;

2. 当 $\varepsilon(t)$ 的值大于阈值, 即断定环境的变化是显著的时, 如果 $t < 3$, 转步 3, 否则, 转步 4;

3. 对非支配种群 $Q_{-1}(G_{\max_t})$ 的部分个体进行高斯变异, 把变异后的 $Q_{-1}(G_{\max_t})$ 种群作为新时刻的初始解. 为了和文献[5]中的算法对比, 本文中设置变异个体比例为 20%;

4. 用预测模型来预测新时刻最优 PS 的位置, 然后在预测位置的周围产生新时刻的初始解.

其中, $G_{\max_t}$ 表示每个时刻允许进化的最大代数. 在刚开始环境变化时, 由于储存的历史信息很少, 不能有效地预测新时刻的最优 PS 位置, 因此我们对前一时刻得到的最优 PS 进行高斯变异, 作为新时刻的初始解. 随着越来越多次的环境改变, 历史信息存储越来越丰富, 可以有效地预测新时刻的最优 PS 位置, 则可利用预测模型来产生新时刻的初始种群.

3.2 基于预测模型的动态多目标免疫优化算法

一般来说, 任何的静态多目标优化算法加上动态环境处理技术都可以用来解决动态多目标优化问题. 本文中, 我们使用基于非支配邻域选择的静态多目标免疫优化算法 (NNIA) 作为算法的框架. 改进的动态多目标优化算法 (PSDMIO) 的流程如下.

算法 2. PSDMIO 算法流程.

输入: 非支配抗体群的最大规模 n_D , 活性抗体群的规模 n_A , 克隆种群的规模 n_c , t 时刻的最大迭代代数 $G_{\max_t}$, 最大的时间步数 $T_{\max}$

输出: 每个时刻的 Pareto 最优解集

1. 随机初始化得到规模为 n_D 的抗体种群 $P(0)$, 评价 $P(0)$, 得到非支配抗体种群 $B(0)$, 按照拥挤距离排序, 从中选择 n_A 个构成活性抗体种群 $A(0)$, 令 $t=0$;

2. 若满足 $t = T_{\max}$, 输出每个环境状态下的最优 PS: $Q_1, \dots, Q_{T_{\max}}$, 结束; 否则, 转步 3;

3. 若 $t > 0$, 基于预测机制得到初始种群 $P_t(0)$, 进而得到非支配抗体种群 $B_t(0)$ 和活性抗体种群 $A_t(0)$, 令 t 时刻的进化代数 $g=0$, 转步 4; 否则, 令 t 时刻的进化代数 $g=0$, 转步 4;

4. 若 $g > G_{\max_t}$, 转步 5; 否则, 输出 t 时刻的非支配抗体群 $B_t(g+1)$ 和活性抗体群 $A_t(g+1)$, 转步 8;

5. 对活性抗体群 $A_t(g)$ 进行比例克隆得到克隆种群 $C_t(g)$;

6. 对克隆种群 $C_t(g)$ 进行改进的差分交叉, 然后进行多项式变异操作得到种群 $C'_t(g)$;

7. 将 $C'_t(g)$ 和非支配抗体群 $B_t(g)$ 合并, 并从 $C'_t(g) \cup B_t(g)$ 中找出非支配抗体群 $B_t(g+1)$ 和活性抗体群 $A_t(g+1)$, 令 $g=g+1$, 转步 4;

8. 令 $t=t+1$, 转步 2.

在算法 2 中, 我们可以认为在每一个环境状态下, 即在每一个时刻 t 内, 环境没有发生变化, 此时的动态多目标优化问题可以看成是静态多目标优化问题.

在免疫学中, 克隆是通过无性繁殖来完成的, 可以表述如下:

$$d_i = \left\lceil n_c \times \frac{r_i}{\sum_{i=1}^{n_A} r_i} \right\rceil \quad (10)$$

其中, $d_i (i=1, 2, \dots, n_A)$ 是第 i 个活性抗体被克隆的个数, $d_i=1$ 表示抗体 a_i 没有进行克隆增殖, r_i 表示活性抗体 a_i 的拥挤距离归一化值. 因式 (10) 采用的是 ceil 函数, 克隆种群的规模有时会大于预先设定的值 n_c . 然而, 基于拥挤度的非支配抗体的删减策略可以保证非支配抗体种群规模不会大于 n_D , 活性抗体种群的规模不会大于 n_A . 作用在活性抗体种群的克隆操作算子 C 可以描述如下:

$$C(a_1, a_2, \dots, a_{n_A}) = \{C(a_1), C(a_2), \dots, C(a_{n_A})\} = \left\{ \underbrace{\{a_1, a_1, \dots, a_1\}}_{d_1}, \underbrace{\{a_2, a_2, \dots, a_2\}}_{d_2}, \dots, \underbrace{\{a_{n_A}, a_{n_A}, \dots, a_{n_A}\}}_{d_{n_A}} \right\} \quad (11)$$

以下介绍本文提出的算法中与 NNIA 不同的几个算子.

3.3 基于拥挤距离的非支配抗体删减机制

在每个时刻的初始, 相似性检测算子决定是否采用基于预测机制的重新初始化策略. 在克隆算子之后, 进行差分交叉算子和多项式变异操作, 然后从合并种群 $C'_t(g) \cup B_t(g)$ 中找出非支配抗体. 当选出的非支配抗体的数目大于非支配种群规模的上限 n_D 或者非支配抗体种群的规模 n_D 大于活性抗体种群规模的最大值 n_A 时, 都是使用拥挤距离^[22]来删减非支配抗体种群或者选择活性抗体的. 在删减的过程中, 我们采用动态删除的策略, 即删除拥挤距离最小的前 k 个个体, 如果非支配抗体的规模仍然大于标准的规模, 重新计算抗体的拥挤距离, 重复以上删除方法, 直到非支配抗体的数目达到标准为止.

3.4 改进的差分交叉算子

改进的差分交叉算子中, 当选择父代抗体产生新抗体时, 采用了一个混合选择机制, 这个机制分为策略 1 和策略 2. 在每个新环境状态的进化初期, 使用策略 1, 当进化代数达到预先设定值时, 开始启用

策略 2. 两个选择策略中父代个体是从非支配种群和克隆种群中选出的, 其中, 非支配种群是由每代得到的非支配抗体组成的, 然后对非支配种群进行删减, 得到一定比例的、拥挤度相对较小的个体, 即活性抗体, 对活性抗体进行复制, 得到克隆种群. $X_{r_1,t}$ 是基本父代抗体, 在两种选择策略中, 都是随机从克隆种群中选择得到的, $X_{r_2,t}, X_{r_3,t}$ 是用来产生差分向量的父代抗体. 在选择策略 1 中, 从非支配种群中随机选择 $X_{r_2,t}$ 和 $X_{r_3,t}$, 在选择策略 2 中, 从克隆种群中随机选择 $X_{r_2,t}$ 和 $X_{r_3,t}$. 这两种不同的选择方法如图 2 所示. 选择过程表示如下:

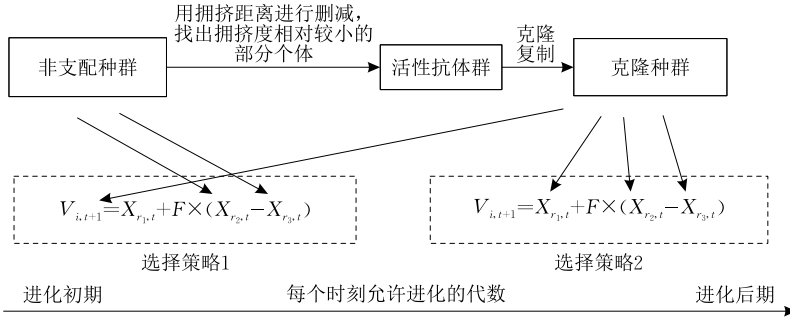


图 2 改进的差分交叉算子中用来产生新个体的父代个体的两种选择策略

3.5 算法复杂度分析

算法复杂度是用来衡量算法实用性的重要标准之一. 算法中决定了复杂度大小的参数有: 最大环境改变次数, 即允许的最大时间步数 $T_{\max}$, t 时刻当前的进化代数 g 和允许的最大迭代代数 $G_{\max}$, 非支配抗体群的最大规模 n_D , 活性抗体群的规模 n_A , 克隆种群的规模 n_c , 每次动态删除的抗体数目 k . 由文献[22]中的分析可知, 除去优化目标个数的影响, 对于规模为 n 的种群, 找出非支配抗体的计算复杂度为 $O(n^2)$, 对非支配抗体计算拥挤距离的 PSDMIO 算法一次迭代所需的最大复杂度如下:

克隆算子的计算复杂度为 $O(n_c)$, 从 $C'_t(g) \cup B_t(g)$ 找出支配抗体群 $B_t(g+1)$ 的计算复杂度为 $O((n_D + n_c)^2)$, 抗体拥挤距离亲和度的计算复杂度为 $O(n_D \log(n_D))$, 更新非支配抗体群的最小的计算复杂度为 $O((n_c/k)(n_D + n_c) \log(n_D + n_c))$.

同时, 从 $t=0$ 时刻的初始化种群中找出非支配种群 $B(0)$ 的计算复杂度为 $O(n_D^2)$, 预测机制的最大计算复杂度为 $O(n_D^2)$, 则算法总的计算复杂度如下:

$$\begin{aligned} & O(n_D^2) + T_{\max} \times O(n_D^2) + T_{\max} \times G_{\max} \times \\ & (O(n_c) + O(n_D \log(n_D)) + O((n_D + n_c)^2) + \\ & O((n_c/k)(n_D + n_c) \log(n_D + n_c))) \end{aligned} \quad (13)$$

$$Selection = \begin{cases} Selection1, & gen < \lambda \times g_{\max} \\ Selection2, & \text{其他} \end{cases} \quad (12)$$

其中, gen 表示当前进化代数, $g_{\max}$ 是 t 时刻允许的最大迭代代数. λ 是控制选择压力因子, 本文令 $\lambda = 0.4$. 改进的差分算子有两个优势: (1) 由于基本父代抗体 $X_{r_1,t}$ 是在克隆种群中随机选择的, 而克隆种群中的抗体都是拥挤度较小的非支配抗体, 这就保证了最优解的分布性质量和算法的收敛速度; (2) 因为在进化的初期, 我们没有用克隆种群产生差分向量, 这样可以保证种群的多样性, 使算法避免陷入局部最优.

在本文中, 令 k 等于 n_c , 根据算子 O 的计算规则, 上式简化结果为

$$O((n_D + n_c)^2) \quad (14)$$

上式表明, 算法的计算复杂度主要集中在计算非支配抗体上. 在后面的实验中, 我们给出了算法具体的运行时间.

4 实验与结果

4.1 测试函数

本文中, 我们测试了 4 个不同的问题, DMOP1^[20] 属于第 1 类问题 (真实 PS 位置随时间变化, 真实 PF 位置不随时间变化), DMOP2^[9] 属于第 3 类问题 (真实 PS 位置不随时间变化, 真实 PF 位置随时间变化), DMOP3^[9] 属于第 2 类问题 (真实 PS 位置和 PF 位置都随时间变化), DMOP4^[20] 属于第 1 类问题 (真实 PS 位置随时间变化, 真实 PF 位置不随时间变化). 其中, 前三个问题是两目标优化问题, 最后一个问题是三目标优化问题. 这 4 个测试函数的数学表达式如表 1 所示. 4 个问题随时间变化的真实 PS 位置和真实 PF 位置如图 3 所示.

表 1 DMO 问题

测试问题	目标函数	决策变量区间
DMOP1	$\begin{cases} f_1(x,t)=x_1, f_2(x,t)=g \cdot (1-\sqrt{f_1/g}) \\ g=1+\sum_{j=2}^n (x_j-G(t))^2, G=\sin(\pi t/2n_T) \end{cases}$	$[0,1] \times [-1,1]^{n-1}$ $n=20$
DMOP2	$\begin{cases} f_1(x,t)=x_1, f_2(x,t)=g \cdot (1-(f_1/g)^H) \\ g=1+9\sum_{j=2}^n x_j^2, H=1.25+0.75 \cdot \sin(\pi t/2n_T) \end{cases}$	$[0,1] \times [-1,1]^{n-1}$ $n=20$
DMOP3	$\begin{cases} f_1(x,t)=x_1, f_2(x,t)=g \cdot (1-(f_1/g)^H) \\ g=1+\sum_{j=2}^n (x_j-G(t))^2 \\ G=\sin(\pi t/2n_T), H=1.25+0.75 \cdot \sin(\pi t/2n_T) \end{cases}$	$[0,1] \times [-1,1]^{n-1}$ $n=20$
DMOP4	$\begin{cases} f_1(x,t)=(1+g) \cdot \cos(0.5\pi x_2) \cdot \cos(0.5\pi x_1) \\ f_2(x,t)=(1+g) \cdot \cos(0.5\pi x_2) \cdot \sin(0.5\pi x_1) \\ f_3(x,t)=(1+g) \cdot \sin(0.5\pi x_2) \\ g(x,t)=\left \sum_{j=3}^n (x_j-G(t))^2\right , G=\sin(\pi t/2n_T) \end{cases}$	$[0,1]^n$ $n=12$

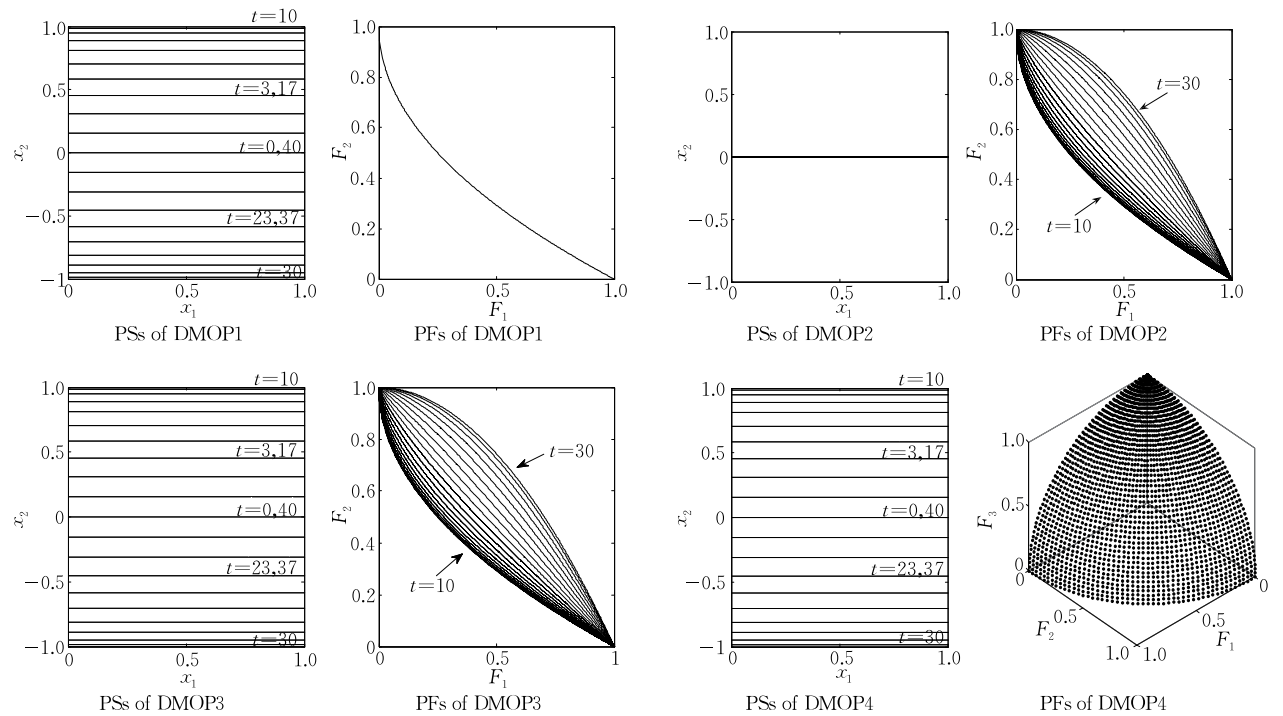


图 3 4 个动态多目标优化问题的真实 PSs 位置和 PFs 位置

4.2 评价指标

在本文的实验研究中,我们用了反代距离 IGD (Inverted Generation Distance)^[23]、两个解集的收敛比较指标 CTS (Coverage of Two Sets)^[6]和间距 S (Spacing)^[24] 3 个评价指标来反映算法的性能。

反代距离. 反代距离指征解集的收敛性。一般而言,代距离越小,表明该算法所得到的解集收敛性越好,越接近于真实的 PF。

令 P^* 表示真实的 PF 上均匀分布的点的集合, A 是算法得到的近似的 PF 点的集合,则 P^* 到 A 的反代距离 IGD 可以表示为

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

(15)

其中 $d(A,v)$ 是指 P^* 中某个个体到 A 中所有个体的欧式距离中的最小取值。若 IGD_t 表示第 t 时刻的反代距离值,则所有时刻的平均反代距离可以表示成如下的形式:

$$\overline{IGD}=\frac{\sum_{t=1}^{T_{\max}}IGD_t}{T_{\max}}$$

(16)

其中, $T_{\max}$ 是最大的时间步数。

收敛比较指标. 收敛比较指标代表一个算法与另一个算法的性能比较情况. 令 A 和 B 分别表示两种算法得到的近似 Pareto 最优解集, 函数 $CTS(A, B)$ 将 (A, B) 之间的收敛关系映射到 $[0, 1]$:

$$CTS(A, B) \triangleq \frac{|\{b \in B; \exists a \in A: a \geq b\}|}{|B|} \quad (17)$$

其中, $\geq$ 表示支配或者相等(也叫弱支配), $CTS(A, B) = 1$ 表示 B 的所有决策向量都被 A 弱支配, 即 A 解集优于 B 解集; $CTS(A, B) = 0$ 表示 B 中没有决策向量被 A 弱支配, 即 B 解集优于 A 解集. 特别要注意收敛比较指标的比较方向, 因为 $CTS(A, B)$ 的值不一定等于 $1 - CTS(B, A)$.

若 CTS_t 表示第 t 时刻算法的 CTS 指标值, 则所有时刻的平均 CTS 指标可以表示成如下的形式:

$$\overline{CTS} = \frac{\sum_{t=1}^{T_{\max}} CTS_t}{T_{\max}} \quad (18)$$

间距. 间距指征解集的均匀性. 一般而言, 间距越小, 表明该算法所得到的解集在目标空间越接近等间距分布.

令 A 代表算法得到的近似的 PF 点, 间距指标 S 定义为

$$S \triangleq \sqrt{\frac{1}{|A| - 1} \sum_{i=1}^{|A|} (\bar{d} - d_i)^2} \quad (19)$$

其中, $d_i = \min \left\{ \sum_{m=1}^k |f_m(a_i) - f_m(a_j)| \right\} (a_i, a_j \in A; i, j = 1, 2, \dots, |A|)$, $\bar{d}$ 是所有 d_i 的平均值, k 表示目标函数的个数.

若 S_t 表示第 t 时刻算法的间距指标, 则所有时刻的平均的间距指标可以表示成如下的形式:

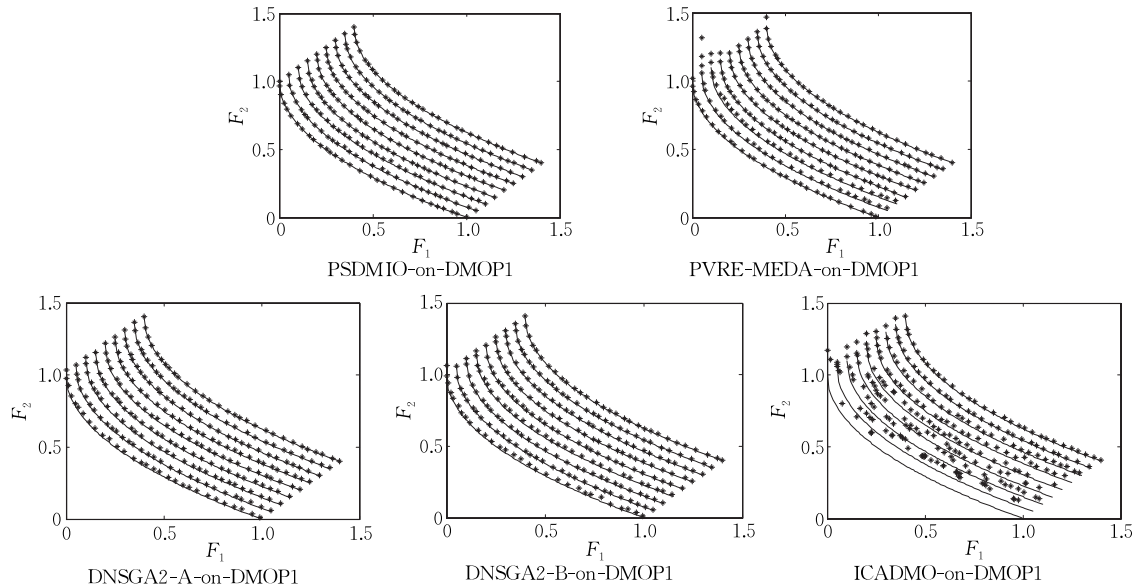


图 4 5 个算法对于 DMOP1 在 $t=0, 1, 2, \dots, 9$ 时得到的非支配解集

$$\bar{S} = \frac{\sum_{t=1}^{T_{\max}} S_t}{T_{\max}} \quad (20)$$

4.3 PSDMIO 与其他 4 个动态多目标算法的实验比较

为了验证提出的 PSDMIO 算法的性能, 我们将它与其他 4 个动态多目标优化算法进行比较. 第 1 个对比算法是由文献[12]中提出, 文中把静态多目标优化算法 RE-MEDA^[12] 和 3 种动态环境预测机制结合起来处理动态多目标优化问题, 并且通过实验证明了, 混合重新初始化方法的性能最优. 这种混合重新初始化方法为, 当检测出环境变化后, 新时刻的初始种群的 50% 个体是通过对上一时刻得到的 Pareto 最优解进行高斯变异得到的, 剩余 50% 个体是通过预测模型得到的. 混合重新初始化策略与 RE-MEDA 算法相结合的算法, 称为 PVRE-MEDA, 即本节中的对比算法之一, 其余 3 个对比算法分别为 DNSGA2-A^[13], DNSGA2-B^[13] 和 ICADMO^[15]. 对于这 5 个算法, 为了保证它们在第 0 时刻的收敛性, 我们设置第 0 个时刻的进化代数 $G_{\max_0} = 100$, 其余时刻 $G_{\max_t} = 50$, 最大时间步数 $T_{\max} = 30$, PSDMIO, DNSGA2-A, DNSGA2-B 和 ICADMO 的种群规模 N 都是 100. RVRE-MEDA 中, LPCA 训练的最大步数 $T = 50$, 聚类类别数 $K_c = 5$, 交叉概率 $p_c = 1$, 变异概率 $p_m = 1/n$, 其中, n 表示决策空间的维数; DNSGA2-A 和 DNSGA2-B 中的交叉概率 p_c 和变异概率 p_m 与 RVRE-MEDA 设置相同; ICADMO 中的克隆比例是 3; PSDMIO 的参数: $n_D = n_c = 100$, $n_A = 20$, $n_T = 10$, 差分算子中的参数设置为 $F = 0.5$, $CR = 0.1$.

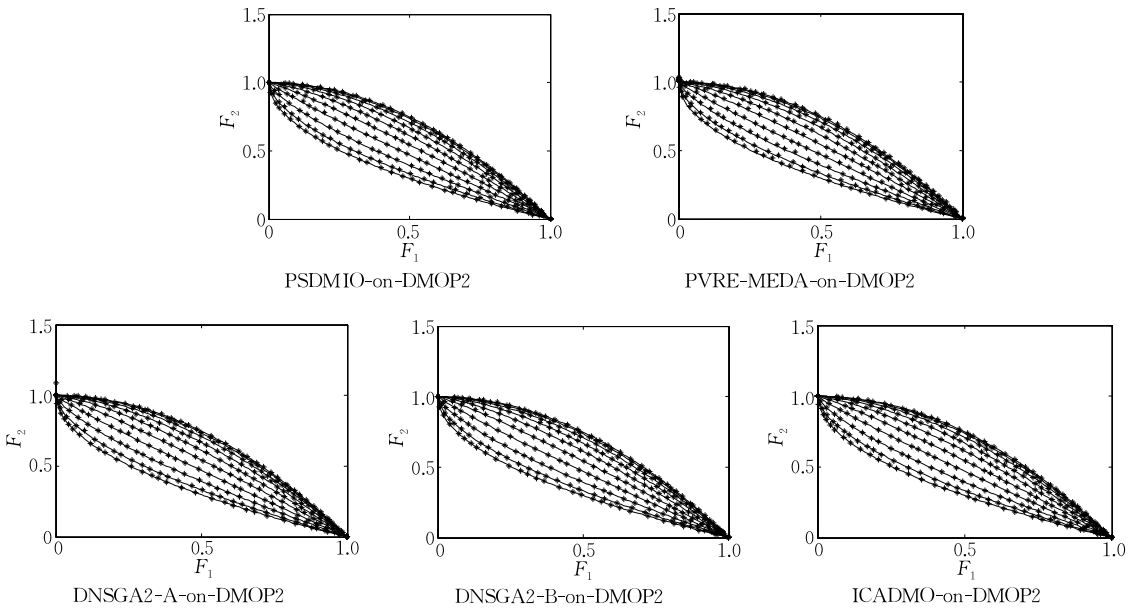


图 5 5 个算法对于 DMOP2 在 $t=11,13,15,\cdots,29$ 时得到的非支配解集

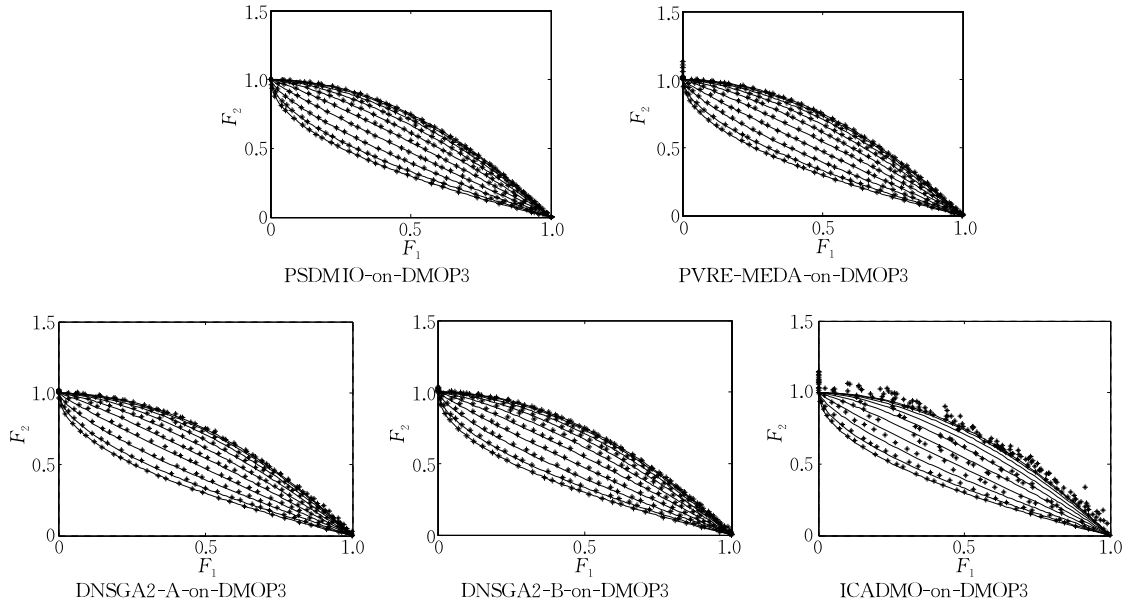


图 6 5 个算法对于 DMOP3 在 $t=11,13,15,\cdots,29$ 时得到的非支配解集

4.3.1 5 个算法得到的非支配解集的分布情况

图 4~图 7 为 5 个算法在 4 个问题上得到的非支配解集. 在图 4~图 6 中, 黑线代表真实的 Pareto 面, 图 7 中, 灰色的面代表真实的 Pareto 面. 黑色的点表示算法得到的近似 Pareto 最优解. 为了可视化, 在图 4~图 7 中, 我们从算法得到的 Pareto 最优解中选择拥挤度最小的 30 个解来画图. DMOP1 的真实 Pareto 面不随时间变化, 在图 3 中, 我们对每个时刻的目标函数值 f_1 和 f_2 进行了平移, 即 $(f_1 + t/20, f_2 + t/20)$, t 表示第 t 时刻.

对于 DMOP1 问题, 从图 4 可以看出, PSDMIO 得到的非支配解集最接近真实的 Pareto 面, 而且解的分布相对比较均匀. PVRE-MEDA 在 $t=2,3,4$ 时

收敛性不好, 之后的收敛性有所改善. DNSGA2-A 和 DNSGA2-B 解的逼近程度相似, 略次于 PSDMIO. ICADMO 得到的非支配解的分布效果最差, 在前几个时刻, 甚至找不到真实的 Pareto 面. 对于 DMOP2 问题, 此问题的真实 PS 是不随时间变化的, 从图 5 可以看出, 5 个算法得到的非支配解与真实 Pareto 前沿的逼近程度基本差不多, 但是 PVRE-MEDA 和 DNSGA2-A 在点 $(0,1)$ 附近收敛的不好.

对于 DMOP3 问题, 从图 6 中可以看出, PSDMIO 得到的非支配解的逼近程度最好, 均匀性相对较好. PVRE-MEDA 中, 点 $(0,1)$ 处收敛不好. DNSGA2-A 和 DNSGA2-B 的性能略次于 PSDMIO, ICADMO 的性能最差, 在大多数时刻不能及时的跟踪到 Pareto

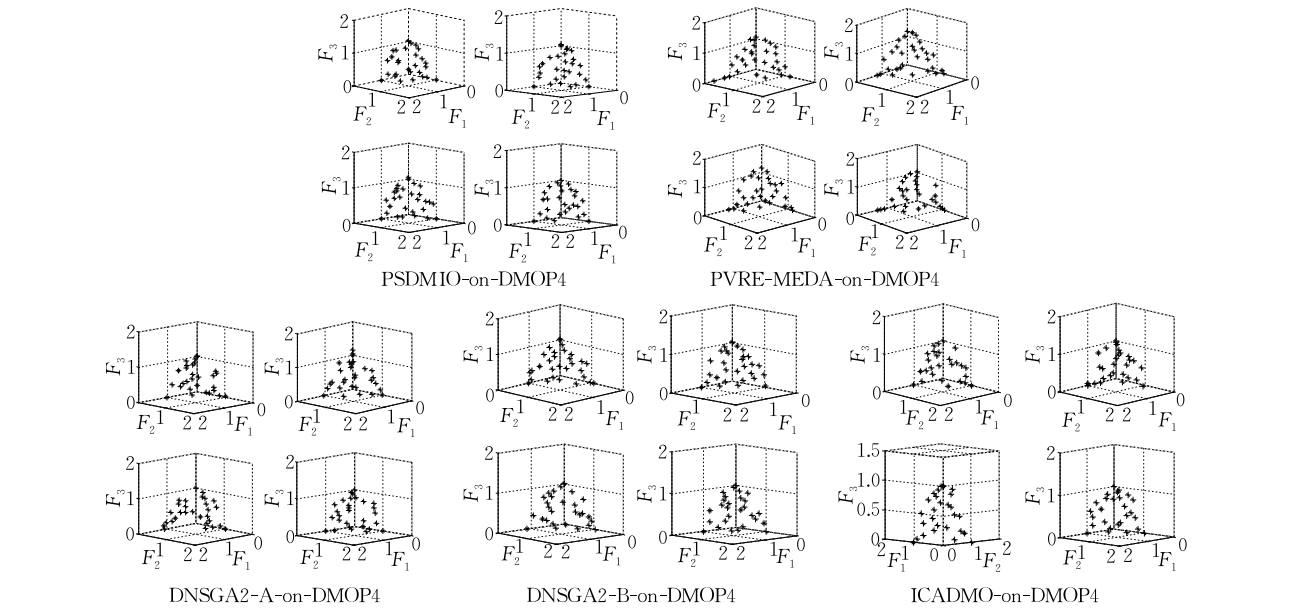


图 7 5 个算法对于 DMOP4 在 $t=14,19,24,29$ 时得到的非支配解集

前沿的移动。

对于 DMOP4 问题,从图 7 我们可以看出,PSDMIO 和 ICADMO 得到的非支配解的逼近程度差不多,比其他 3 个算法相对较好,PSDMIO 解的均匀性相对较好. DNSGA2-A 和 DNSGA2-B 得到的解的逼近性和均匀性都相当. ICADMO 在此问题中收敛性表现最差,许多点偏离真实 Pareto 面较远。

综合上面对 4 个问题所得到的非支配解集结果的分析,可以清楚地看出,PSDMIO 在 4 个问题上的解的质量是最好的,相比其他 4 个算法,解的均匀性也较好. PVRE-MEDA 在 DMOP1 的前几个时刻收敛不好,对于 DMOP2 和 DMOP3 问题在点(0,1)处的收敛性不好,在 DMOP4 上的收敛性能最差. DNSGA2-A 和 DNSGA2-B 的性能很接近,在 DMOP1 和 DMOP3 上解的质量优于 PVRE-MEDA 和 ICADMO,在 DMOP2 上 DNSGA2-B 和 ICADMO 接近,比 DNSGA2-A 收敛性更好. ICADMO 除了在 DMOP2 上优于 PVRE-MEDA 和 DNSGA2-A,在 DMOP4 问题上优于 PVRE-MEDA、DNSGA2-A 和 DNSGA2-B 之外,在另外两个问题上的性能很差,所以该算法十分不稳定。

4.3.2 5 个算法的 IGD 跟踪情况和性能盒图^[25]

本小节中,为了将实验的图表示得更为清晰,我们对 5 个算法进行了编号,如表 2 所示。

表 2 对算法进行编号	
编号	算法
1	PSDMIO
2	PVRE-MEDA
3	DNSGA2-A
4	DNSGA2-B
5	ICADMO

图 8 是 5 个算法在 4 个问题上的 IGD 跟踪曲线图。

从图 8 可以看出,对于 DMOP1 问题,PSDMIO 的 IGD 跟踪性能最好,PVRE-MEDA 的跟踪性能仅次于 PSDMIO,DNSGA2-A 和 DNSGA2-B 的跟踪性能很相近且差于 PVRE-MEDA,ICADMO 的跟踪性能最差;在 $0 < t < 3$ 时,进化初期历史信息不足,不能形成预测模型,但 PSDMIO 的跟踪性能还是最好的,随着进化的进行,历史信息逐渐积累,可以形成预测模型,PSDMIO 的 IGD 跟踪优势变得越来越明显。

对于 DMOP2 问题,在 $0 < t < 7$ 时,PSDMIO 的 IGD 跟踪曲线波动有点大,说明算法对环境变化敏感,虽然在进化的初期,其 IGD 值大于 DNSGA2-A 和 DNSGA2-B 的 IGD 值,但是 PSDMIO 有很快的收敛速度,在给定的迭代代数内,最终得到的 IGD 值依然是最好的;在 $t > 7$ 时,PSDMIO 的性能最好. DNSGA2-A 和 DNSGA2-B 的总体跟踪性能仅次于 PSDMIO,PVRE-MEDA 的 IGD 跟踪曲线成一个平稳的波动状态. ICADMO 在进化初期表现不好,后期的 IGD 跟踪性能有所改善。

对于 DMOP3 问题,所有算法的表现和在 DMOP1 问题上的表现相似,但是在绝大多数时刻,PSDMIO 的跟踪性能都是最好的. ICADMO 直到最后几个时刻才优于 PVRE-MEDA,而且在前面时刻 IGD 指标值和其他算法差距很大。

对于 DMOP4 问题,在整个进化过程中,PSDMIO 的 IGD 跟踪性能一直是最好的,ICADMO 的性能

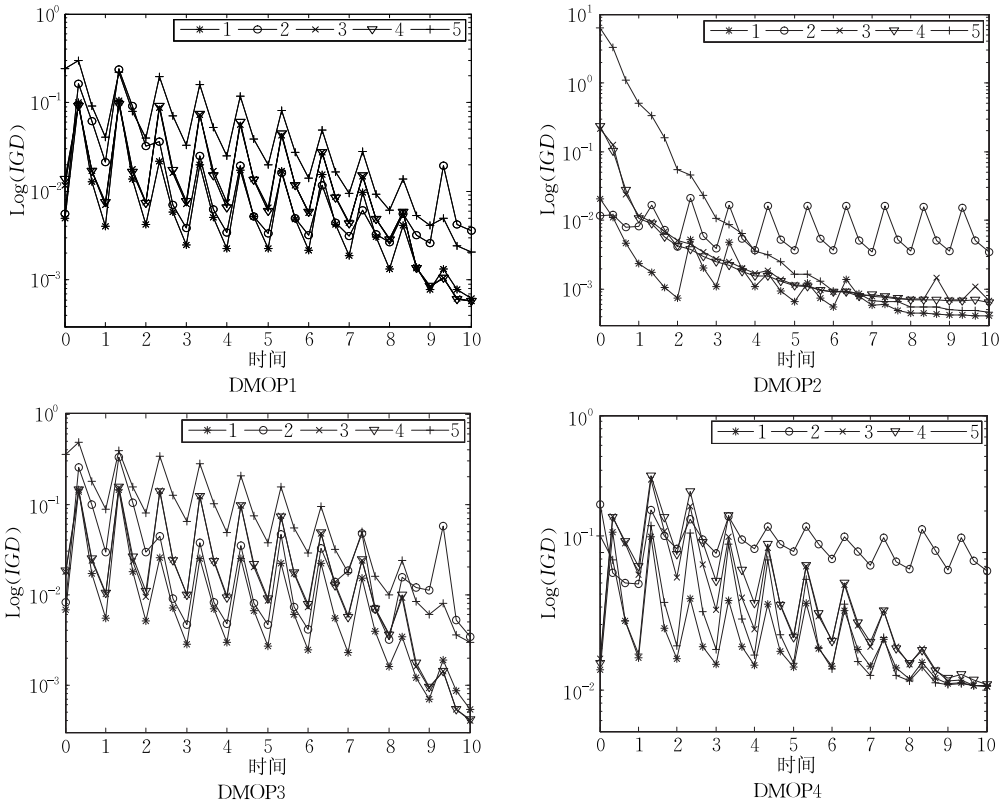


图 8 5 个算法的 IGD 跟踪曲线

次之,PVRE-MEDA 跟踪性能最差.

从 IGD 指标跟踪图的分析中,我们可以看到 IGD 值和前面的非支配解集的分布图基本是一致的,由此我们也可以看出 PSDMIO 的性能是最好的,也是稳定的.

图 9 是 5 个算法的 20 次独立运行的 $\overline{IGD}$ 、 $\overline{CTS}$ 和 $\overline{S}$ 性能盒图比较.

就 $\overline{IGD}$ 指标来说,PSDMIO 在 4 个测试函数上的表现都是最好的,而且有明显的性能优势. PVRE-MEDA 的性能在 DMOP1、DMOP2 和 DMOP3 上仅次于 PSDMIO,在 DMOP3 上的表现最差;DNSGA2-A 和 DNSGA2-B 这 2 个算法性能相当,且居于第三;ICADMO 在 DMOP1、DMOP2 和 DMOP3 上的性能都是最差的,在 DMOP3 上的性能仅次于 PSDMIO.

就 $\overline{CTS}$ 指标来说,从图 9 可以看出,PSDMIO 得到的非支配解集支配在所有问题上都支配其他的算法得到的非支配解集,也就是说,PSDMIO 算法得到的 Pareto 最优解最逼近于真实的 Pareto 前沿.

就 $\overline{S}$ 指标来说,PSDMIO 算法对于 4 个测试函数得到的解集的均匀性是最好的. PVRE-MEDA 对于 DMOP1、DMOP2 和 DMOP3 得到的解集的均匀性略次于 PSDMIO,而对于 DMOP4 的解集的均匀性最差;DNSGA2-A 和 DNSGA2-B 这 2 个算法的

解集的均匀性位居第三;ICADMO 对于前 3 个测试函数的解集的均匀性都是最差的,而对于 DMOP4 的解集的均匀性仅次于 PSDMIO.

总之,PSDMIO 算法在 IGD 的跟踪上,除了 DMOP2 问题,在其他测试问题上,算法的性能都是最好的,对于 DMOP2,算法在前几个时刻对环境的变化比较敏感,说明 PSDMIO 在处理真实 PS 不随时间变化而 PF 随时间变化的多目标问题时还需要改进,但进化一定代数之后解集的收敛性依然最好;此外,PSDMIO 算法在 $\overline{IGD}$ 、 $\overline{CTS}$ 和 $\overline{S}$ 这 3 个指标上都展现了最好的性能.

4.4 PSDMIO 算法里算子和参数的研究

本节中,我们将讨论预测机制、改进的差分交叉算子两个机制对算法性能的影响. 另外,也将讨论两个主要参数对 PSDMIO 算法性能的影响:相似性检测中,用于检测环境是否变化的 $\epsilon(t)$ 的阈值 δ ,以及每个时刻允许进化的最大代数 G_{max} .

4.4.1 预测机制和改进的差分交叉算子对算法性能的影响

本小节的实验参数设置和第 4.3 节的设置一致. 就预测机制来说,从图 10 和图 11 可以清楚地看到,对于 DMOP1、DMOP3 和 DMOP4,预测机制使算法在动态环境中得到的解集对 IGD 跟踪能力

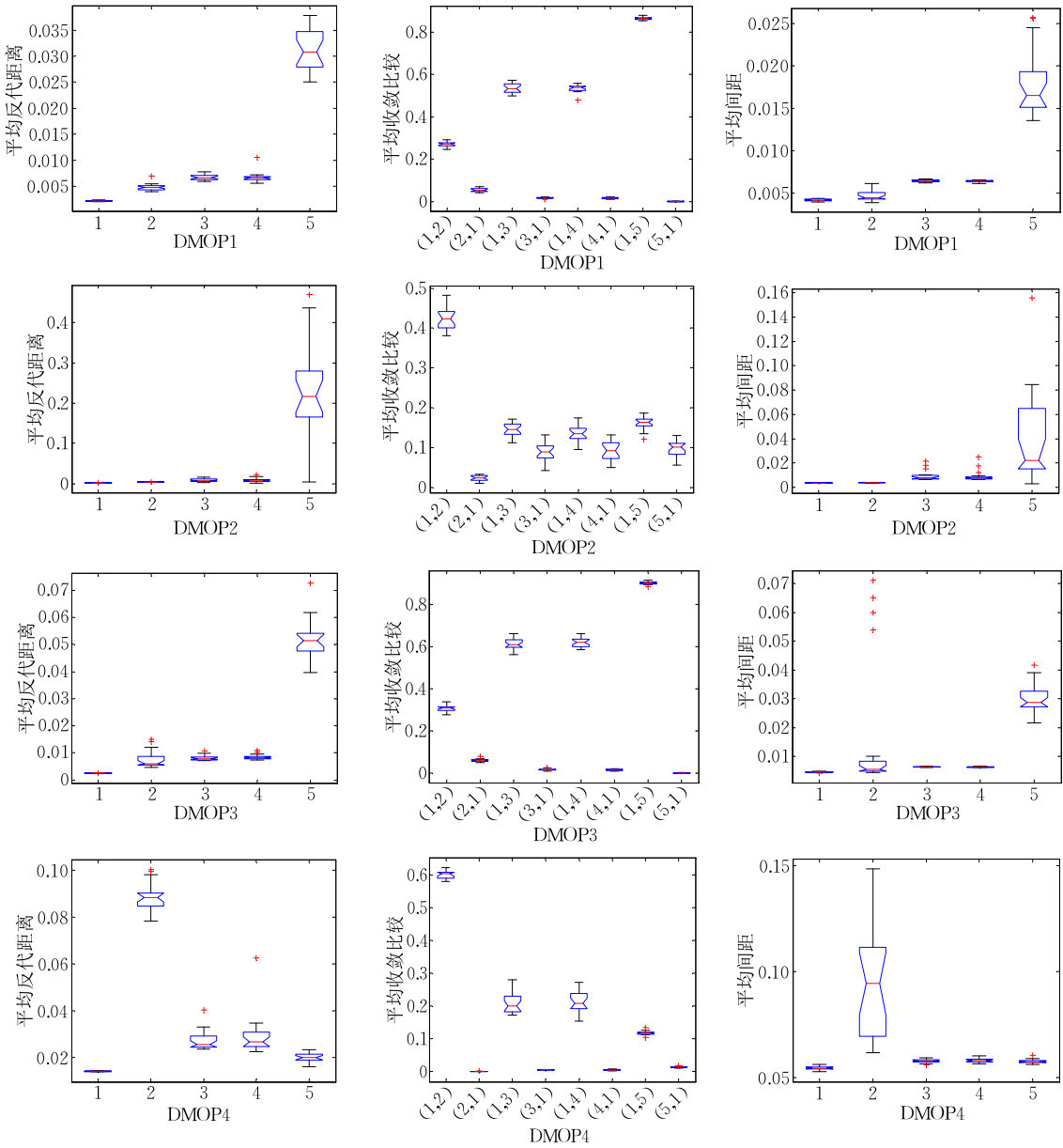


图 9 5 个算法的性能盒图,第 1 列为 $\overline{IGD}$ 盒图;第 2 列为 $\overline{CTS}$ 盒图;第 3 列为 $\overline{S}$ 盒图

大大增强. 而对于 DMOP2, 预测机制基本无效, 这也许是因为 DMOP2 的真实 PS 位置是不随时间变化的, 新环境来到后, 继续在上一时刻的最优 PS 位置上进行搜索会加速新时刻下算法的收敛速度, 而通过预测机制产生的新时刻的初始解分布在上一时刻最优 PS 位置的周围, 不能在上个时刻进化的基础上继续进化, 要耽误一些迭代次数, 因而对 DMOP2 这一类测试函数, 预测机制会使算法的跟踪性能和收敛速度有一定程度的减弱.

就改进的差分交叉算子来说, 从图 10 可以看到, 在每个时刻中的最后一代, 基于改进的差分交叉算子的算法得到的解集的 IGD 指标最小, 算法的收敛性更好, 从而可以证明, 差分交叉算子加快了算法

的收敛速度. 我们从图 11 可以看出, 在 4 个测试问题上, DE 交叉算子的加入, 均能使算法的覆盖率更好, 算法得到的近似 PS 更逼近于真实值, 进一步证明了改进的差分交叉算子可以提高算法的收敛速度.

4. 4. 2 检测环境变化的阈值 δ

本实验我们令环境检测的阈值 δ 的范围为 $[0.01, 0.1]$, 观察算法 20 次独立运行的 $\overline{IGD}$ 指标值和环境检测阈值 δ 之间的关系. 从 4. 3 节的实验, 我们已经知道, PSDMIO 对于 DMOP1、DMOP2 和 DMOP3 这 3 个测试函数的性能相似, 因此本文只给出了 DMOP1 和 DMOP2 的 $\overline{IGD}$ 指标和环境检测阈值 δ 的关系曲线图, 如图 12 所示.

从图 12 我们可以看到, 对于 DMOP1 问题, 除

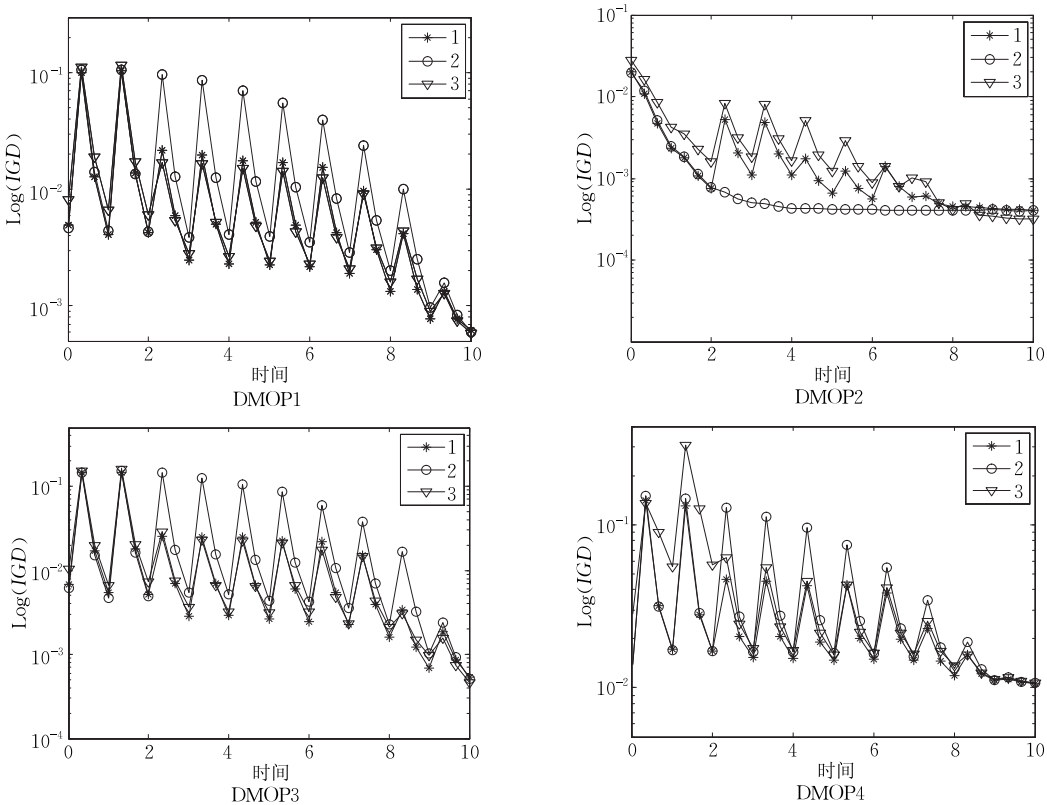


图 10 PSDMIO 与在其框架下不使用预测策略或者改进的差分交叉算子的算法得到的 IGD 跟踪的比较 (图中 1 表示 PSDMIO,2 表示不加预测策略的算法,3 表示使用 SBX 而不使用 DE 的算法)

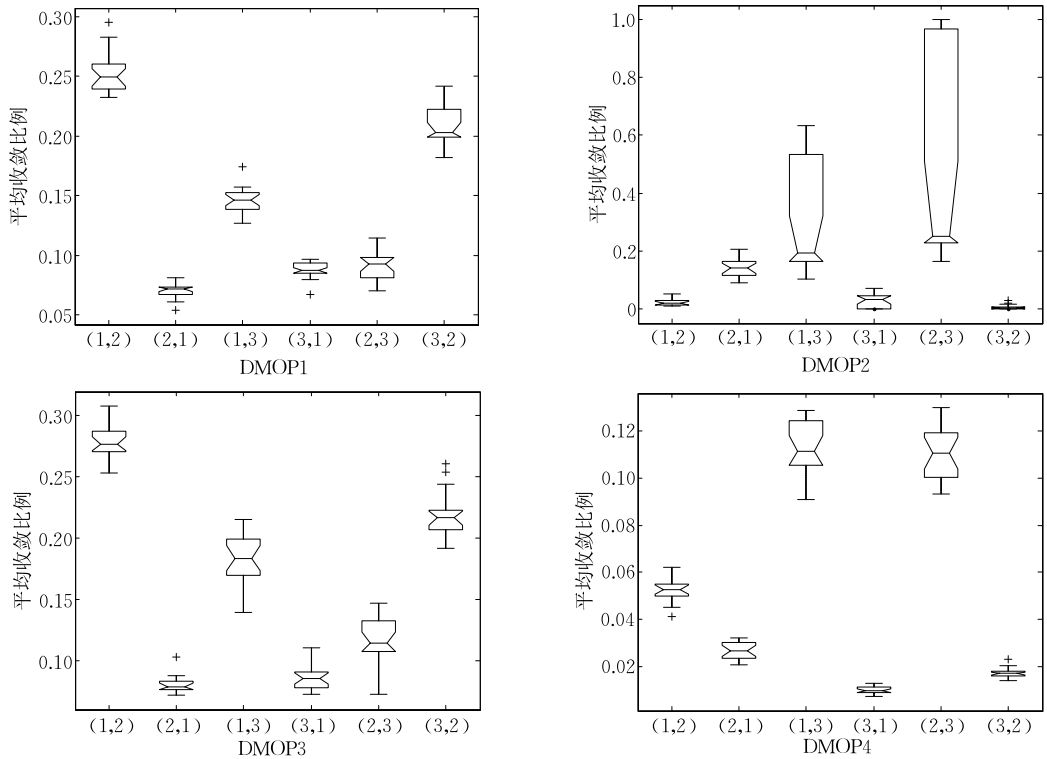


图 11 PSDMIO 与在其框架下不使用预测机制或者改进的差分交叉算子的算法 $\overline{CTS}$ 比较 (图中 1 表示 PSDMIO,2 表示不加预测机制的算法,3 表示使用 SBX 而不使用 DE 的算法)

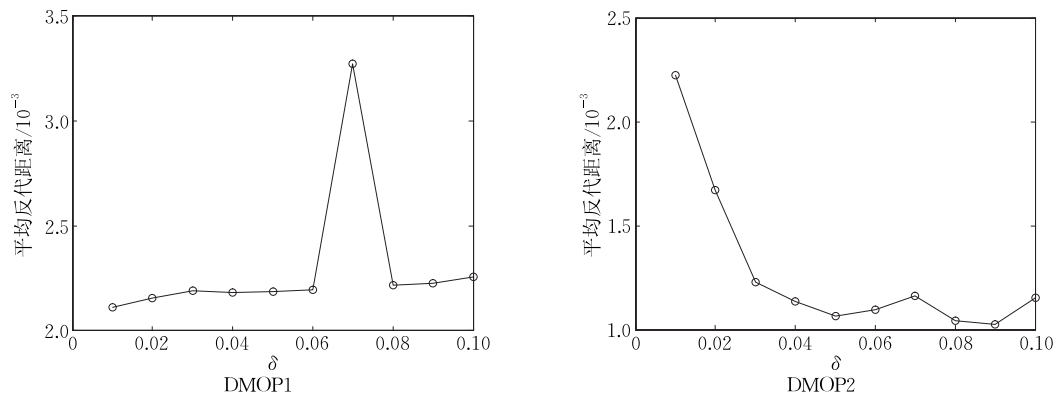


图 12 $\overline{IGD}$ 随检测环境阈值 δ 的变化曲线

了当 δ 取值为 0.07 时, $\overline{IGD}$ 的值相对较大外, 算法性能相对较弱, 而 δ 的取值从 0.01 到 0.1 变化时, $\overline{IGD}$ 总体上有增大的趋势, 但是变化比较缓慢. 因此, 对于 DMOP1 这类测试函数, 我们建议 δ 的取值范围为 $[0.01, 0.05]$. 对于 DMOP2 函数, 随着 δ 的增大, $\overline{IGD}$ 有变小的趋势, 从图 12 中可以看出, δ 从 0.05 之后, $\overline{IGD}$ 的值波动不是很大, 因此对于 DMOP2 这类测试函数, 我们建议 δ 的范围为 $[0.05, 0.10]$.

4.4.3 每个时刻允许进化的最大代数 $G_{\max_i}$ 的研究

本实验中, 和 4.3.2 节中一样, 1 表示 PSDMIO, 2 表示 PVRE-MEDA, 3 表示 DNSGA2-A, 4 表示 DNSGA2-B, 5 表示 ICADMO. 只有 $G_{\max_i}$ 有变化, 其

余参数设置与 4.3 节一样. 图 13 是算法独立运行 20 次得到的 $\overline{IGD}$ 指标值与 $G_{\max_i}$ 的关系曲线图, 表 3 是不同 $G_{\max_i}$ 取值下的算法独立运行 20 次平均运行的时间. 本文运行的平台是 Matlab7.0.1, 电脑配置: Intel(R) Core(TM) 2 CPU (2.33 GHz, 2 GB, RAM).

从图 13 可以看出, 对于 4 个测试问题, 在每个时刻下, 允许迭代的最大代数 $G_{\max_i}$ 取值变化时, PSDMIO 的性能始终都是最好的, 而且随着 $G_{\max_i}$ 取值的增大, PSDMIO 得到的 $\overline{IGD}$ 指标曲线平稳下降, 即使 $G_{\max_i}$ 很小, PSDMIO 也能跟踪到算法的变化. PVRE-MEDA 在 DMOP1 和 DMOP3 函数上的

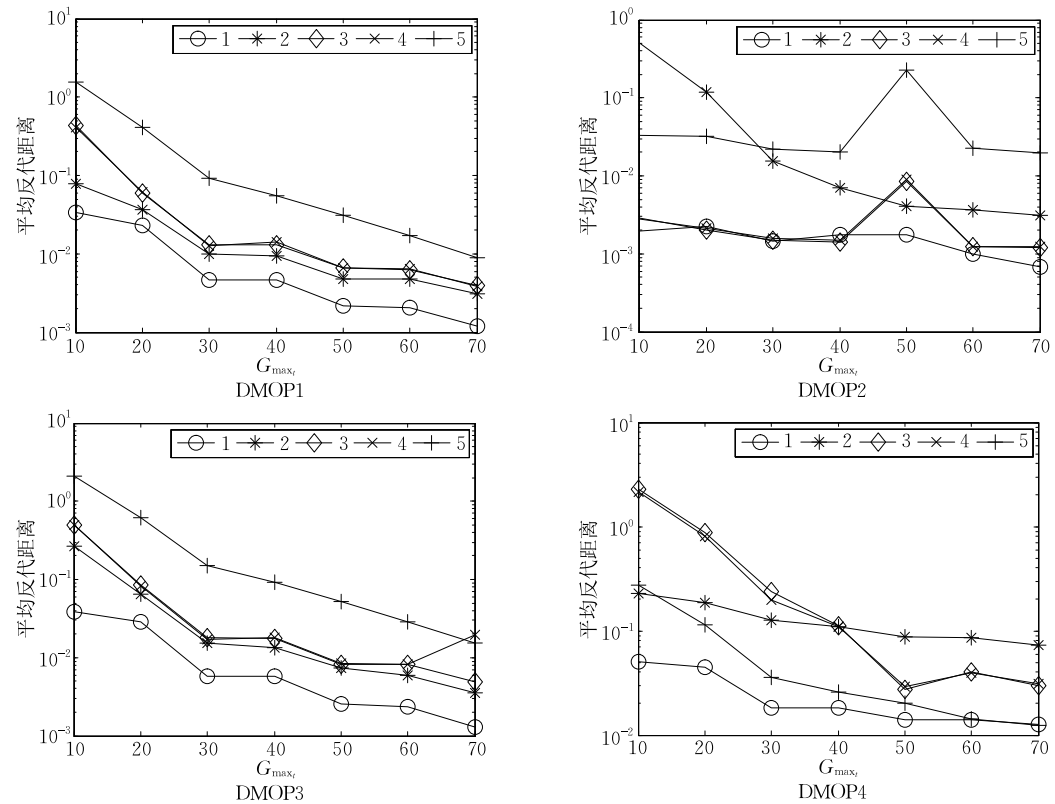


图 13 $\overline{IGD}$ 随 $G_{\max_i}$ 的变化曲线

性能仅次于 PSDMIO,但对于 DMOP2 和 DMOP4 函数,算法在 $G_{\max_i} > 40$ 时表现最差. DNSGA2-A 和 DNSGA2-B 这 2 个算法在 DMOP1 和 DMOP3 上性能都居于第三,对于 DMOP2 函数,除 $G_{\max_i} = 50$ 外,

其余性能都是仅次于 PSDMIO,对于 DMOP4 问题,当 $G_{\max_i} < 40$ 时,算法的跟踪性能最差,之后的性能有一点改善. ICADMO 在 DMOP1 和 DMOP3 问题上表现最差,在 DMOP3 问题上性能仅次于 PSDMIO.

表 3 5 个算法在不同 $G_{\max_i}$ 下的算法运行时间 (单位:s)

$G_{\max_i}$	PSDMIO	PVRE-MEDA	DNSGA2-A	DNSGA2-B	ICADMO
10	0.6410 ± 0.001	2.4357 ± 0.032	182.997 ± 8.225	182.6779 ± 5.490	7.3915 ± 1.334
20	1.2779 ± 0.024	3.9286 ± 0.135	315.906 ± 9.086	315.7608 ± 7.991	11.612 ± 3.221
30	29.1711 ± 1.258	29.443 ± 1.987	449.655 ± 11.002	448.6881 ± 14.007	25.108 ± 1.987
40	39.6210 ± 2.002	40.229 ± 3.457	582.118 ± 10.252	584.9025 ± 13.005	35.513 ± 1.998
50	59.524 ± 2.987	62.954 ± 3.012	849.213 ± 21.340	847.3321 ± 19.237	57.602 ± 2.045
60	60.929 ± 3.554	63.450 ± 2.730	1272.700 ± 36.005	1151.257 ± 22.05	58.166 ± 2.872
70	68.423 ± 4.112	75.002 ± 3.334	1843.566 ± 28.007	1905.009 ± 34.998	65.122 ± 5.009

从表 3 可以看出,DNSGA2-A 和 DNSGA2-B 耗时最长,PSDMIO 和 PVRE-MEDA, ICADMO 在 $G_{\max_i}$ 取值变化的情况下基本耗时相同,而且远小于 DNSGA2-A 和 DNSGA2-B. 动态多目标优化问题的算法运行时间也是衡量算法性能的重要标准之一,PSDMIO 算法能够在保证算法性能的前提下,在运行时间上也达到较好的水平.

4.5 PSDMIO 的预测机制和其他预测机制的比较

在这一部分中,我们对 PSDMIO 的预测机制进行了修改,换成了文献[26]的预测机制,相应的算法起名为 PSDMIO-SBM,用以说明预测机制的特点,以及算法的可以改进之处. 在 PSDMIO-SBM 中,除

了预测机制不同以外,其他算子都和 PSDMIO 的算子一样,然后将 PSDMIO-SBM 得到的结果和文中所提出的算法得到的结果进行了比较. 从原理上看,SBM 需要记录多个时刻的 Pareto 最优解,消耗了存储空间,但是选择的是具有代表性的子问题,均匀性得到了很好的保持. 两个算法的 IGD 跟踪性能的比较如图 14 所示.

从图 14 可以看出,两种预测机制各有优势和劣势,对于问题 2 和问题 4,SBM 的预测机制取得的结果更好一些,对于问题 1 和问题 3,两种预测机制在不同的时刻各有优势,说明本文的算法在预测机制上是很有效的,在以后的工作中,可以进一步改进.

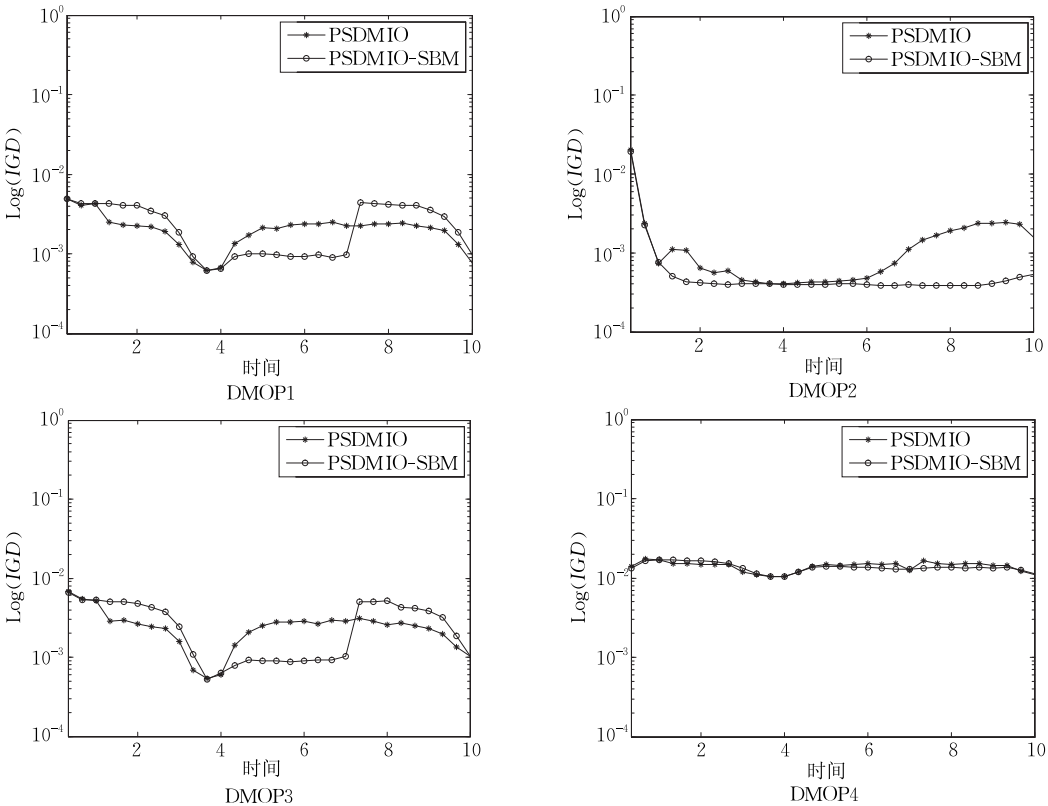


图 14 PSDMIO 算法和 PSDMIO-SBM 算法的IGD 跟踪比较

5 结 论

本文中,我们提出了一种新的动态多目标免疫优化算法(PSDMIO). 算法中引入了两种算子,包括预测机制和改进的差分交叉算子. 实验结果证明,预测机制可以提高算法对环境改变的适应能力和收敛速度,而改进的差分交叉算子又能进一步提高算法的收敛速度. 从实验结果分析中可以看出,PSDMIO 算法在处理传统的动态多目标优化问题上是有优势的. 然而,我们也知道动态多目标优化问题的类型是多种多样的,场景变化也是十分复杂的,所以 PSDMIO 算法是依赖于目标问题的种类和属性的. 当 PSDMIO 应用于前两种类型的动态多目标优化问题时,即真实 PS 随时间变化而 PF 不随时间变化的动态多目标优化问题,或者真实 PS 和 PF 都随时间变化的动态多目标优化问题时,PSDMIO 算法的性能是很好的,尤其当动态多目标优化问题的真实 PS 随时间呈现线性变化趋势的时候,PSDMIO 的优势明显,但是当处理第三种类型的动态多目标优化问题,即真实 PS 不随时间变化而 PF 随时间变化的动态多目标优化问题时,由于预测模型是失效的,PSDMIO 算法的性能不是很理想,还有待改进. 如何改进算法,将算法应用于更多的场景,是我们下一步研究的首要任务. 此外,将提出的算法应用到其他的 DMO 问题以及实际问题中也是我们下一步工作的内容.

参 考 文 献

- [1] Yang S X, Yao X. Population-based incremental learning with associative memory for dynamic environments. *IEEE Transactions on Evolutionary Computation*, 2008, 12(5): 542-561
- [2] Yu X, Jin Y C, Tang K. Robust optimization over time—A new perspective on dynamic optimization problems//*Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2010)*. Barcelona, Spain, 2010: 1-6
- [3] Jin Y C, Branke J. Evolutionary optimization in uncertain environments—A survey. *IEEE Transactions on Evolutionary Computation*, 2005, 9(3): 303-317
- [4] Bui L T, Abbass H A, Branke J. Multi-objective optimization for dynamic environments//*Proceedings of the IEEE Congress on Evolutionary Computation (CEC'2005)*. Edinburgh, Scotland, 2005: 2349-2356
- [5] Wang Y, Li B. Investigation of memory-based multi-objective optimization evolutionary algorithm in dynamic environment//*Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2009)*. Trondheim, Norway, 2009: 630-637
- [6] Zitzler E, Thiele L. Multiobjective optimization using evolutionary algorithms—A comparative case study//*Proceedings of the 5th International Conference on Computer Science*. Amsterdam, Netherlands, 1998, 1498: 292-301
- [7] Liu R C, Zhang W, Jiao L C, et al. A sphere-dominance based preference immune-inspired algorithm for dynamic multi-objective optimization//*Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2010)*. Portland, USA, 2010: 423-430
- [8] Zeng S Y, Chen G, Zheng L, et al. A dynamic multi-objective evolutionary algorithm based on an orthogonal design//*Proceedings of the IEEE Congress on Evolutionary Computation (CEC'2006)*. Vancouver, Canada, 2006: 573-580
- [9] Goh C K, Tan K C. A competitive-cooperative coevolutionary paradigm for dynamic multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2009, 13(1): 103-127
- [10] Wang Y P, Dang C Y. An evolutionary algorithm for dynamic multi-objective optimization. *Applied Mathematics and Computation*, 2008, 205(1): 6-18
- [11] Koo W T, Goh C K, Tan K C. A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment. *Memetic Computing*, 2010, 2(2): 87-110
- [12] Zhou A M, Jin Y C, Zhang Q F, et al. Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization//*Proceedings of the 4th International Conference on Computer Science (EMO 2007)*. Matsushima, Japan, 2007, 4403: 832-846
- [13] Deb K, Bhaskara U N, Karthik S. Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling//*Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*. Matsushima, Japan, 2007, 4403: 803-807
- [14] Hatzakis I, Wallace D. Dynamic multi-objective optimization with evolutionary algorithms: A forward-looking approach//*Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2006)*. Seattle, USA, 2006: 1201-1208
- [15] Shang Rong-Hua, Jiao Li-Cheng, Gong Mao-Guo, Ma Wen-Ping. An immune clonal algorithm for dynamic multi-objective optimization. *Journal of Software*, 2007, 18(11): 2700-2711 (in Chinese)
- [16] Zhang Z H, Qian S Q. Multi-objective optimization immune algorithm in dynamic environments and its application to greenhouse control. *Applied Soft Computing*, 2008, 8(2): 959-971
- [17] Zhang Z H, Qian S Q. Multi-objective immune optimization in dynamic environments and its application to signal simulation

- //Proceedings of the 2009 International Conference on Measuring Technology and Macaronis Automation (ICMTMA). Hunan, China, 2009, 3: 246-250
- [18] Zhang Z H, Qian S Q. Artificial immune system in dynamic environments solving time-varying non-linear constrained multi-objective problems. *Soft Computing*, 2011, 15(7): 1333-1349
- [19] Gong M G, Jiao C, Du H F, Bo L F. Multi-objective immune algorithm with non-dominated neighbor-based selection. *Evolutionary Computation*, 2008, 16(2): 225-255
- [20] Farina M, Amato P, Deb K. Dynamic multi-objective optimization problems: Test cases, approximations and applications. *IEEE Transactions on Evolutionary Computation*, 2004, 8(5): 425-442
- [21] Price K V, Storn R M, Lampinen J A. *Differential Evolution: A Practical Approach to Global Optimization*. Berlin, Germany: Springer, 2005
- [22] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-197
- [23] Van Veldhuizen D A. *Multi-Objective Evolutionary Algorithms: Classification, Analysis, and New Innovations* [Ph.D. dissertation]. Air Force Institute of Technology, Wright-Patterson AFB, 1999
- [24] Schott J R. *Fault Tolerant Design Using Single and Multi-Criteria Genetic Algorithm Optimization* [M. S. dissertation]. Massachusetts Institute of Technology, Massachusetts, 1995
- [25] McGill R, Tukey J W, Larsen W A. Variations of boxplots. *The American Statistician*, 1978, 32(1): 12-16
- [26] Liu Min, Zeng Wen-Hua. Memory enhanced dynamic multi-objective evolutionary algorithm based on decomposition. *Journal of Software*, 2013, 24(7): 1571-1588(in Chinese) (刘敏, 曾文华. 记忆增强的动态多目标分解进化算法. *软件学报*, 2013, 24(7): 1571-1588)



LIU Ruo-Chen, born in 1974, Ph.D., associate professor. Her research interests include immune computation, multi-objective optimization and data mining.

MA Ya-Juan, born in 1981, M. S. Her research interests include multi-objective optimization and data mining.

ZHANG Lang, born in 1984, M. S. His research interest focus on multi-objective optimization.

SHANG Rong-Hua, born in 1979, Ph. D., associate professor. Her research interests include intelligent information processing, pattern recognition, and data mining.

Background

This work is supported by the National Natural Science Foundation of China under Grant Nos. 61373111, 61003199, and 61371201, the Provincial Natural Science Foundation of Shaanxi of China (No. 2010JM8030), and the Fundamental Research Funds for the Central Universities (Nos. K50511020014, K5051302084). Dynamic multi-objective optimization is a challenging task because of either the objective function or constraint or the associated problem parameters variation with time, at the same time, the positions of the true Pareto Front (PF) and Pareto Set (PS) are also changed. Given the computation expense, the dynamic multi-objective optimization aims to trace changing Pareto optimal front and PS. The changing environment requires that algorithms can converge to the optimal Pareto set as soon as possible. Those existed static multi-objective optimizations fail to track the variation of environment timely and can not trace the changing Pareto optimal front and PS well if they are applied to the dynamic multi-objective optimization problem directly.

In this paper, a new dynamic multi-objective immune

optimization algorithm based on prediction strategy is proposed. In the proposed algorithm, firstly, when a change in the objective space is detected, a modified forecasting model based on the non-dominated antibodies in previous optimum locations is used to construct a forecasting model and generate the initial antibody population so as to improve the ability of response to changing environment. A similarity detection operator based on the population statistics information is used to detect the variation of environment. Secondly, in the early generation, it is difficult to form the forecasting model due to little history information, and the initial solutions at generation t can be generated by performing a Gaussian disturbance on the Pareto optimal set of preceding generation; in the late generation, the abundance history information can be used to construct the forecasting model. Finally, in order to avoid speed the convergence, an improved differential evolution crossover with two selection strategies is proposed. Experimental results show that the proposed algorithm has a good ability to track dynamic changes.