

动态进化多目标优化中的串式记忆方法

刘 敏^{1,2} 曾文华³ 刘玉珍^{1,2}

(湖南科技大学计算机科学与工程学院 湘潭 411201)¹

(知识处理与网络化制造湖南省普通高等学校重点实验室 湘潭 411201)²

(厦门大学软件学院 厦门 361005)³

摘 要 如何利用过去搜索到的最优解对新的环境变化做出快速响应,是动态进化多目标优化(Dynamic Evolutionary Multi-objective Optimization, DEMO)研究的一大挑战。为此提出了一种串式记忆(Bunchy Memory, BM)方法。设计了基于极小化效用函数的抽取过程,从非支配集中抽取一串记忆串,以便保持记忆的多样性;将记忆体组织成串式队列的方式,以便将过去数次环境变化下抽取的记忆串存入记忆体;提出了基于二进制锦标赛选择的检索过程以复用记忆体中过去的最优解,来快速响应新的变化。BM 方法具有良好的记忆效果,显著地提高了 DEMO 算法的收敛性和多样性。4 个标准测试问题上的实验结果表明, BM 方法比其它 3 种方法具有更好的记忆能力。相应地,集成了 BM 方法的 DEMO 算法所获得解集的收敛性与多样性也明显好于其它 3 种 DEMO 算法。

关键词 进化计算, 多目标优化, 动态环境, 记忆, Pareto 最优前沿

中图分类号 TP18 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.12.044

Bunchy Memory Method for Dynamic Evolutionary Multi-objective Optimization

LIU Min^{1,2} ZENG Wen-hua³ LIU Yu-zhen^{1,2}

(School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China)¹

(Key Laboratory of Knowledge Processing and Networked Manufacturing, College of Hunan Province, Xiangtan 411201, China)²

(Software School, Xiamen University, Xiamen 361005, China)³

Abstract One of the challenges in dynamic evolutionary multi-objective optimization (DEMO) is how to exploit past optimal solutions to help DEMO algorithm track and adapt to the changing environment quickly. To alleviate the above difficulty, this paper proposed a bunchy memory (BM) method for DEMO. In the BM method, firstly, a sampling procedure based on minimized utility function is designed to sample a bunch of memory points from the non-dominate set so as to maintain good diversity of memory. Then, the memory is organized as a bunchy queue, so that a number of bunches of memory points sampled from past environment changes can be easily stored in the memory. Next, past optimal solutions stored in the memory are reused to rapidly respond to the new change by using a retrieving procedure based on binary tournament selection. The BM method has good memory effect and improves the convergence and diversity of DEMO algorithm significantly. Experiment results on four benchmark test problems indicate that the proposed BM method has better memory performance than other three methods. Accordingly, the convergence and diversity of the DEMO algorithm, which incorporates the BM method, are also obviously better than those of the other three DEMO algorithms.

Keywords Evolutionary computation, Multi-objective optimization, Dynamic environment, Memory, Pareto optimal front

1 引言

无论是科学研究还是工程应用,许多优化问题都需要对多个目标进行同时优化,人们把这类问题称为多目标优化问题(Multi-objective Optimization Problem, MOP)。MOP 问题中多个目标经常彼此冲突,需要在各目标间进行折中,最后得出一组折中解——Pareto 最优集(Pareto Optimal Set, POS),供决策者进行决策^[1]。进化多目标优化(Evolutionary Multi-objective Optimization, EMO)主要研究如何利用进化计算求

解 MOP 问题,已成为进化计算领域的研究热点。目前 EMO 研究主要集中在静态优化领域,但是除多个目标外,许多优化问题实际上还是动态变化的。受动态环境影响,问题的目标函数、约束函数和相关参数都可能随时间不断变化。相应地,问题的 POS 也可能随时间而变化。人们将这类问题称为动态多目标优化问题(Dynamic Multi-objective Optimization Problem, DMOP)^[2]。除应具备静态 EMO 算法那样良好的收敛性和多样性外,求解 DMOP 的动态进化多目标优化(Dynamic Evolutionary Multi-objective Optimization, DEMO)算

到稿日期:2015-12-31 返修日期:2016-03-26 本文受湖南省教育厅资助科研项目(14C0437, 15C0545)资助。

刘 敏(1974—),男,博士,讲师,主要研究方向为进化计算、多目标优化, E-mail: liumin_xt@163.com; 曾文华(1964—),男,博士,教授,主要研究方向为智能计算、云计算; 刘玉珍(1979—),男,博士,讲师,主要研究方向为进化计算。

法还须利用以往的最优解对新的环境变化做出快速响应,以合理的计算开销进行快速优化搜索,以便及时跟踪不断变化的 POS,这给 EMO 研究带来了新的挑战^[3-5]。近年来,一些研究人员对 DEMO 产生了浓厚兴趣,他们分别根据进化算法、人工免疫算法、粒子群优化算法等自然计算方法设计了相应的 DEMO 算法^[6-10]。目前,DEMO 在智能机器人导航、工业设计、管理工程、控制及优化调度等领域已有了初步应用^[11-15]。因此对 DEMO 进行深入研究具有重要的科学意义和实际应用价值。

目前,预测策略与记忆方法是 DEMO 算法中应对环境变化的两种有效方法。对于具有一定可预测性的动态优化问题,预测策略一般根据已有的历史信息建立预测模型,以更好地指导种群搜索^[7,16,17]。与预测策略相比,记忆方法则更适合于应对周期性或近似周期性环境变化问题^[18]。譬如一年中的气候变化以及一天中的交通变化,此类问题在新环境下的最优解可能又重新回到以前搜索过的位置。记忆方法一般由记忆体(Memory, M)、存入(store)过程、检索(retrieve)过程组成。其中记忆体保存过去的最优解;存入过程负责将过去最优解存入记忆体;检索过程负责从记忆体中检索出最优解并将其插入到新环境下的种群中。记忆增强的进化算法可通过记忆复用以前搜索到的最优解,对新变化做出快速响应^[4]。

尽管记忆方法在动态单目标进化算法中已得到大量研究^[4,18],但是由于 DMOP 问题比动态单目标优化问题更难求解,因此至今 DEMO 的记忆方法研究成果非常少。2007 年 Deb 等人对 NSGA-II 算法^[19]进行改进,提出了动态 NSGA-II (Dynamic NSGA-II, DNSGA-II)算法^[11],DNSGA-II 中包含一种多样性引入(Diversity Introduction, DI)方法(本文将其命名为 DNSGA-II's DI,DDI),DDI 方法保留了环境变化之前种群中的部分最优解集信息,具有一定的短期记忆效果。2010 年 Wang 等人提出了一种类似于记忆的种子重新初始化策略(Re-initialization Strategy, RS)^[20],RS 包含多样性引入和高斯变异两部分,也有一定的短期记忆效果。但是 DDI 和 RS 中没有记忆体以及存入与检索过程,都不是真正的记忆方法。2010 年 Koo 等人提出了一种基于中心点-方差(Centroid-Variance, CV)的记忆方法(本文将其命名为 Centroid-Variance Memory, CVM)^[21],CVM 方法虽是一种记忆方法,但实际记忆效果不理想。

为此,本文提出了一种串式记忆(Bunchy Memory, BM)方法,旨在提高 DEMO 算法的快速响应能力和进化寻优能力。其基本思想为:1)设计了基于极小化效应函数的抽取过程,从非支配集中抽出一串记忆串,以便保持记忆的多样性;2)设计了串式队列方式的记忆体组织形式;3)提出了基于二进制锦标赛选择的快速检索过程,复用记忆体中过去的最优解集信息,快速响应新变化。实验结果表明:BM 方法具有良好的记忆能力,能显著提高 DEMO 算法的收敛性和多样性。

2 动态多目标优化问题定义与相关记忆方法

2.1 相关定义

不失一般性,一个具有 n 个决策变量, m 个目标函数的 MOP 问题可描述为^[1]:

$$\begin{cases} \min_{x \in \Omega} \mathbf{y} = \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))^T \\ \text{s. t. } g_i(\mathbf{x}) \leq 0, i=1, 2, \dots, p \\ h_j(\mathbf{x}) = 0, j=1, 2, \dots, q \end{cases} \quad (1)$$

其中, $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \Omega \subset \mathbb{R}^n$ 为 n 维决策(变量)向量, Ω 为 n 维决策空间。 $\mathbf{y} = (f_1, f_2, \dots, f_m)^T \in \Lambda \subset \mathbb{R}^m$ 为 m 维目标函数向量, Λ 为 m 维目标空间。评价函数 $\mathbf{F}(\mathbf{x}): \Omega \rightarrow \Lambda$ 定义了 m 个由决策空间到目标空间的映射。 $g_i(\mathbf{x}) \leq 0 (i=1, 2, \dots, p)$ 定义了 p 个不等式约束, $h_j(\mathbf{x}) = 0 (j=1, 2, \dots, q)$ 定义了 q 个等式约束。

作为自然的扩展,一个具有 n 个决策变量, m 个目标函数的 DMOP 问题可描述为^[2]:

$$\begin{cases} \min_{x \in \Omega} \mathbf{y} = \mathbf{F}(\mathbf{x}, t) = (f_1(\mathbf{x}, t), f_2(\mathbf{x}, t), \dots, f_m(\mathbf{x}, t))^T \\ \text{s. t. } g_i(\mathbf{x}, t) \leq 0, i=1, 2, \dots, p \\ h_j(\mathbf{x}, t) = 0, j=1, 2, \dots, q \end{cases} \quad (2)$$

其中,评价函数 $\mathbf{F}(\mathbf{x}, t)$ 、不等式约束条件函数 $g_i(\mathbf{x}, t)$ 以及等式约束条件函数 $h_j(\mathbf{x}, t)$ 都可能随时间 t 发生变化。

定义 1(Pareto 支配(dominance)关系)^[1] 向量 $\mathbf{u} = (u_1, u_2, \dots, u_m)^T$ Pareto 支配向量 $\mathbf{v} = (v_1, v_2, \dots, v_m)^T$, 记为 $\mathbf{u} < \mathbf{v}$, 当且仅当:1) $\forall k \in \{1, 2, \dots, m\}$ 满足 $u_k \leq v_k$; 2) $\exists l \in \{1, 2, \dots, m\}$ 满足 $u_l < v_l$ 。

定义 2(非支配集/档案集) 假设 P 为一个解集, $\forall \mathbf{u} \in P$, 若 $\exists \mathbf{v} \in P$, 使 $\mathbf{v} < \mathbf{u}$, 则称 \mathbf{u} 为集合 P 的非支配个体。由 P 的所有非支配个体组成的集合称为 P 的非支配集(Non-dominated Set, Nds)。某些文章也将 Nds 称为档案集(Archive, A)。

定义 3(Pareto 最优集(POS)) 对于一个给定的 MOP 问题,其 POS 定义为^[1] $\text{POS} := \{\mathbf{x} \in \Omega \mid \exists \mathbf{x}' \in \Omega, \mathbf{F}(\mathbf{x}') < \mathbf{F}(\mathbf{x})\}$ 。

相应地,DMOP 问题的 POS(t)可定义为: $\text{POS}(t) := \{\mathbf{x} \in \Omega \mid \exists \mathbf{x}' \in \Omega, \mathbf{F}(\mathbf{x}') < \mathbf{F}(\mathbf{x}, t)\}$ 。

定义 4(Pareto 最优前沿(Pareto Optimal Front, POF)) 对于一个给定的 MOP 问题,其 POF 定义为^[1]:

$$\text{POF} := \{\mathbf{y} = \mathbf{F}(\mathbf{x}) \mid \mathbf{x} \in \text{POS}\}$$

相应地,DMOP 问题的 POF(t)可定义为 $\text{POF}(t) := \{\mathbf{y} = \mathbf{F}(\mathbf{x}, t) \mid \mathbf{x} \in \text{POS}(t)\}$ 。

2.2 设计记忆方法时须考虑的 3 个要点

2001 年 Branke 就如何在动态单目标进化算法中设计记忆方法提出了以下 3 个要点^[4]: 1)应该在什么时候将种群中的哪些个体存入到记忆体中? 2)记忆体中应该保存多少个个体,哪些个体应被替换(replace)以便记忆体腾出空间来接纳其它新个体? 3)应该从记忆体中检索哪些个体并将它们重新插入到种群中?

这 3 个要点对于设计 DEMO 算法中的记忆方法来说同样重要。当新变化出现时,因为 DEMO 算法需要将 POS(t)中一些具有多样性要求的最优解存入记忆体,而非动态单目标进化算法只需存入一个或少数几个最优解,所以 DEMO 算法中记忆方法的设计难度远大于动态单目标进化算法的记忆方法。本文将在 2.3 节介绍与记忆有关的多样性引入方法;然后在 2.4 节围绕这 3 个要点对 CVM 记忆方法进行简要介绍;接下来在第 3 节的 BM 方法中对这 3 个要点进行深入阐述。

2.3 多样性引入方法

多样性引入是处理动态优化问题的一种常用方法^[3]。DNSGA-II 算法中的 DDI 就是一种处理 DMOP 问题的多样性引入方法。DDI 通过对当前种群 P_t 中的部分个体(所占种

群比例定义为 ζ) 进行随机初始化, 给 P_t 引入一定多样性, 而后对 P_t 中个体重新进行评价, 获得环境变化后的初始种群 P_t' 。因为 P_t 中保存了环境变化前搜索的最优解, 所以 DDI 潜在地记忆了上次环境变化的部分最优解信息, 即 DDI 方法具有部分短期记忆效果。总之, 采用了 DDI 的 DNSGA-II 算法的基本步骤可归纳如下:

- step1 set 迭代次数计数器 $\tau = 0$; 随机初始化种群 P_t , 并对 P_t 中的个体进行评价;
- step2 If (环境发生变化^[11])
{ 执行 DDI(P_t), 获得种群 P_t' ;}
Else set $P_t' = P_t$;
- step3 执行 NSGA-II^[19] 中的 make-new-pop(P_t'), 得到 Q_t ; set $R_t = P_t' \cup Q_t$;
- step4 对 R_t 执行 NSGA-II 中的非支配排序和截断方法, 获得种群 P_{t+1} ;
- step5 如果满足停止条件则停止; 否则, set $\tau = \tau + 1$, 并跳转至 step2。

2010 年 Wang 等人提出的 RS 策略^[20] 实际上也是一种多样性引入方法。RS 策略可表述为: 新种群 P_t' 中的一部分个体(所占种群比例定义为 pl)通过对档案集中的个体进行高斯变异产生, 另一部分个体(占种群比例为 $1 - pl$)通过随机初始化产生。该高斯变异定义为: $a = (a_1, a_2, \dots, a_n)$, $c = (c_1, c_2, \dots, c_n)$, $c_i = a_i + N(0, \sigma)$, $i = 1, 2, \dots, n$ 。其中 a 是档案集中的一个个体, c 是对 a 进行高斯变异后产生的子个体。由于 RS 在高斯变异前记忆了变化前一代档案集中的部分最优解信息, 因此 RS 也具有一定的短期记忆效果。

值得注意的是: 虽然 DDI 和 RS 都具有一定短期记忆效果, 但是它们没有设计记忆体、存入与检索过程。

2.4 基于中心点-方差的记忆方法

2010 年 Koo 等人参照 Branke 提出的 3 个要点为 DEMO 算法设计了基于中心点-方差的记忆方法(CVM)。

对于要点 1), CVM 方法的存入过程大致为: 首先计算档案集的中心点向量(C_t)和方差向量(\tilde{C}_t), 然后将 C_t 和 \tilde{C}_t 合并为一个记忆项目, 最后将该记忆项目存入记忆体。

对于要点 2), 如果记忆体存满, 则 CVM 方法将存入时间最长的记忆项目移出记忆体, 以便接纳新的记忆项目。

对于要点 3), CVM 方法的检索过程大致为: 首先备份记忆体; 其次截断备份的记忆体, 使其大小减小为预定的大小; 然后对备份记忆体中的每一个记忆项目, 根据正态分布产生新的解 x , 即令 $x = N(C_t, \tilde{C}_t)$, 并将 x 插入到档案集中。

值得注意的是: CVM 方法的截断时间复杂度为 $O(N^3)$, 具有较高的时间复杂度。此外, CVM 方法的存入过程仅保存中心点向量和方差向量而丢弃掉档案集中的最优解信息, 检索过程仅根据中心点向量和方差向量进行正态抽样。这两个过程间容易产生失真, 特别在三维目标问题上其记忆的准确性难以得到保障。

3 串式记忆方法

3.1 串式记忆方法的设计原理

本文为动态进化多目标优化提出了串式记忆(BM)方法。针对 2.2 节中 Branke 的 3 个要点, BM 方法主要由抽取、存入与检索 3 个步骤构成, 其设计原理如图 1 所示。

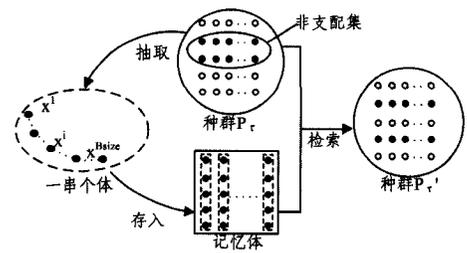


图 1 串式记忆方法的原理图

对于要点 1), 为选出要保存的个体, BM 方法中设计了一种基于极小化效用函数(minimized Utility Function, minUF)的抽取(sample)过程。当环境发生变化时, 抽取过程利用 minUF 从种群 P_t 的非支配集(Nds)中抽取一串大小为 B_{size} 的非支配个体(简称为记忆串)。下面先介绍本文提出的 minUF, 然后描述该抽取过程。

2007 年 Deb 等人采用了一种效用函数来选择动态双目标优化问题的 Pareto 最优前沿的中点^[11], 该效用函数定义如下:

$$w_1(x) = \frac{(f_1^{\max} - f_1(x)) / (f_1^{\max} - f_1^{\min})}{(f_1^{\max} - f_1(x)) / (f_1^{\max} - f_1^{\min}) + (f_2^{\max} - f_2(x)) / (f_2^{\max} - f_2^{\min})} \quad (3)$$

其中, x 是 Nds 中的非支配个体; f_k^{\max} 和 f_k^{\min} 分别是第 k 个目标上的极大极小值($k = 1, 2$); $w_1(x)$ 表示归一化后第一个目标在两个目标中的相对重要程度。

minUF 与 Deb 的方法不同之处在于: minUF 需要从 Nds 中自动抽取一串在二维 Pareto 最优前沿曲线上分布比较均匀的记忆串, 而非 Deb 方法中抽取的一个中点。为此, 在式(3)的基础上, minUF 的定义如下:

$$\min_{x \in Nds} UF_i(x, r_j) = |w_i(x) - r_j| \quad (4)$$

$$r_j = j / (B_{size} - 1), j = 0, \dots, B_{size} - 1$$

其中, r_j 是权重变量, 其值取决于变量 j 与串大小 B_{size} 。本文基于以上极小化效用函数, 通过变更权重来实现抽取过程。下面以 $i = 1$ 、目标个数 $m = 2$ 且 $B_{size} = 5$ 为例来说明该抽取过程, 相应的示意图如图 2 所示。当 $j = 0, r_j = 0$ 时, 目标函数 f_1 的权重(偏好程度)为 0, minUF 抽出个体 x^1 , 其目标空间上的映射 $y^1 = F(x^1, t)$ 为非支配前沿的一个端点。当 $j = 1, r_j = 0.25$ 时, minUF 抽出个体 x^2 , 其目标空间上的映射为 $y^2 = F(x^2, t)$; 当 $j = 2, r_j = 0.5$ 时, minUF 抽出个体 x^3 , 其目标空间上的映射 $y^3 = F(x^3, t)$ 为非支配前沿的中点; 以此类推, 最后当 $j = 4, r_j = 1$ 时, minUF 抽出个体 x^5 , 其目标空间上的映射 $y^5 = F(x^5, t)$ 为非支配前沿的另一个端点。总之, 采用 minUF 能抽出一串分布比较均匀的记忆串, 在目标空间上保持了较好的多样性。

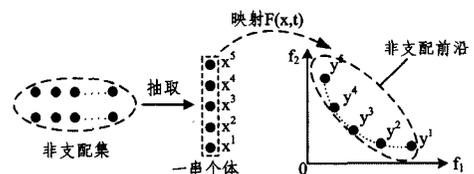


图 2 基于极小化效用函数的抽取过程示意图

对于动态双目标优化问题, 可直接利用 minUF 完成抽取过程。

对于 3 个目标的 DMOP, 因为此时 Pareto 最优前沿不再

是二维曲线,所以需要抽取过程进行相应的调整。首先,将待抽取的记忆串分成4个部分。然后参照式(4)构造3个效用函数 $UF_i(x)$ ($i=1,2,3$)。接下来通过 $\min UF$ 分别在三维曲面的每一个边界曲线上都抽取 $\text{int}(B_{size}/4)$ 个个体(int 表示取下整),从而得到前3部分待抽取个体。最后,剩余的第四部分个体从 Nds 中随机抽取。

对于3个以上目标的DMOP,可参考三维抽取过程,本文不再详述。

对于要点2),BM采用一个与种群规模大小相等的记忆体(Memory,M)。因为记忆串中各个体间彼此关联,且构成一个有机整体,所以BM方法将M设计成串式队列的形式(令记忆串为队列的基本元素)。每次存入(store)时,先将要存入的记忆串插入到队列的队尾,如果M存满,则按先进先出策略进行替换(把队首的记忆串从M中删除)。

对于要点3),鉴于需要,从M中检索(retrieve)出一些精英个体以淘汰种群 P_t 中的一些较差个体,因此合并 P_t 和M,然后通过NSGA-II中的非支配排序和截断过程^[14],选出最好的N个个体,这似乎是一个不错的检索方案。但是,该非支配排序的时间复杂度为 $O(N^2)$,具有较高的时间复杂度,因此BM方法没有采纳这种方案,而是设计了如下快速的检索过程:首先合并M的备份(原M不变)和 P_t ,得到组合种群 R_t ;然后,针对新环境根据式(2)重新计算 R_t 中个体的目标函数值和约束函数值;最后,在 R_t 上通过二进制锦标赛选择(binary tournament selection)选出N个获胜个体,得到记忆后的种群 $P_{t'}$ 。

3.2 串式记忆方法的伪代码描述

串式记忆方法的伪代码描述如下:

/* 从种群 P_t 的Nds中抽取一串记忆串存入记忆体M,然后通过检索过程得出记忆后的种群 $P_{t'}$ */

Input: 种群 P_t ; 非支配集Nds; 记忆体M

Output: 记忆后的种群 $P_{t'}$

Begin

Step 1 抽取(sample)过程:

If ($m==2$) /* m为目标个数 */

bunch= $\min UF_1(Nds)$; /* $\min UF$ 为极小化效应函数,bunch为抽取的记忆串 */

Else /* $m==3$ */

bunch= $\min UF_1(Nds) \cup \min UF_2(Nds) \cup \min UF_3(Nds) \cup \text{random}(Nds)$ /* random随机选出第4部分个体 */

Step 2 存入(store)过程:

M.push_back(bunch); /* 将bunch插入到记忆体的队尾 */

If (M存满) M.pop_front(). /* 将队首的记忆串删除 */

Step 3 检索(retrieve)过程:

$M' = \text{duplicate}(M)$; /* M' 为M的备份 */

$R_t = P_t \cup M'$;

evaluate(R_t); /* 计算目标函数值和约束函数值 */

$P_{t'} = \text{binary_tournament_selection}(R_t)$; /* 锦标赛选择 */

output $P_{t'}$.

End

3.3 串式记忆方法的时间与空间复杂度分析

设种群 P_t 和记忆体M的规模都为N,令 $B_{size} < 0.1 \times N$ 。

时间复杂度分析:每抽出一个个体的时间复杂度为 $O(N)$,且共需抽出 B_{size} 个,故抽取过程的时间复杂度为 $O(B_{size}N)$;存入过程的时间复杂度为 $O(B_{size})$;检索过程的时间复杂度为 $O(N)$,具有快速的时间复杂度。总之,BM方法

的时间复杂度为 $O(B_{size}N)$ 。因为 B_{size} 远小于N,所以BM方法也具有快速的时间复杂度。

空间复杂度分析: P_t 和M的规模同为N, R_t 空间需求为 $2N$,所以BM方法的空间复杂度为 $O(N)$ 。

4 实验

为比较DDI,RS,CVM和BM方法的记忆能力,统一采用2.3节的DNSGA-II作为基础算法。首先,将RS,CVM和BM方法集成到DNSGA-II中,即分别用这3种方法替代DNSGA-II中的DDI,从而得到DNSGA-II-RS,DNSGA-II-CVM和DNSGA-II-BM;然后用DNSGA-II,DNSGA-II-RS,DNSGA-II-CVM和DNSGA-II-BM这4种DEMO算法求解DMOP问题;最后对实验结果进行比较分析。

4.1 测试问题

采用FDA系列DMOP问题作为测试问题^[2]。2007年Deb修改了FDA2问题(修改版本记作FDA2mod)^[11]。因FDA2mod无周期性变化,本文对FDA2mod进行改进,给出一个新的FDA2(记作FDA2new)。这些测试问题的定义如表1所列。其中, τ_T 为环境变化频率,即每 τ_T 代环境发生一次变化,但 τ_T 代内DMOP问题是静态的。 n_T 为环境变化剧烈程度, n_T 值较小时,环境变化幅度较大,每个周期内不同的POF(t)个数较少; n_T 值较大时,环境变化幅度较小,每个周期内不同的POF(t)个数较多。

表1 动态多目标优化的测试问题

问题	目标函数	变量边界
DMOP1 (FDA1)	$\begin{cases} f_1(x_1) = x_1, f_2(x) = g \cdot h \\ g(x_{II}) = 1 + \sum_{x_i \in x_{II}} (x_i - G(t))^2, \\ h(f_1, g) = 1 - \sqrt{f_1/g} \\ G(t) = \sin(0.5\pi t), t = \lfloor \tau/\tau_T \rfloor / n_T \end{cases}$	$\begin{cases} x_1 = (x_1) \in [0, 1] \\ x_{II} = (x_2, \dots, \\ x_n) \in [-1, 1] \end{cases}$
DMOP2 (FDA2 new)	$\begin{cases} f_1(x_1) = x_1, f_2(x) = g \cdot h \\ g(x_{II}) = 1 + \sum_{x_i \in x_{II}} x_i^2, h(x_{III}, f_1, g) \\ = 1 - (f_1/g)^2 \\ (H(t) + \sum_{x_i \in x_{III}} (x_i - H(t)/4)^2) \\ H(t) = 2\sin(0.5\pi(t-1)), t = \lfloor \tau/\tau_T \rfloor / n_T \end{cases}$	$\begin{cases} x_1 = (x_1) \in [0, 1] \\ x_{II} = (x_2, \dots, \\ x_6) \in [-1, 1] \\ x_{III} = (x_7, \dots, \\ x_n) \in [-1, 1] \end{cases}$
DMOP3 (FDA4)	$\begin{cases} f_1(x) = (1 + g(x_{II})) \prod_{i=1}^{m-1} \cos(0.5\pi x_i) \\ f_k(x) = (1 + g(x_{II})) \\ (\prod_{i=1}^{m-k} \cos(0.5\pi x_i)) \\ \sin(0.5\pi x_{m-k+1}), k=2; m-1 \\ f_m(x) = (1 + g(x_{II})) \sin(0.5\pi x_1) \\ g(x_{II}) = \sum_{x_i \in x_{II}} (x_i - G(t))^2, G(t) \\ = \sin(0.5\pi t) , t = \lfloor \tau/\tau_T \rfloor / n_T \end{cases}$	$\begin{cases} x_{II} = (x_m, \dots, x_n) \\ x_i \in [0, 1], \\ i=1; n \end{cases}$
DMOP4 (FDA5)	$\begin{cases} f_1(x) = (1 + g(x_{II})) \prod_{i=1}^{m-1} \cos(0.5\pi y_i) \\ f_k(x) = (1 + g(x_{II})) (\prod_{i=1}^{m-k} \cos(0.5\pi y_i)) \\ \sin(0.5\pi y_{m-k+1}), k=2; m-1 \\ f_m(x) = (1 + g(x_{II})) \sin(0.5\pi y_1) \\ g(x_{II}) = G(t) + \sum_{x_i \in x_{II}} (x_i - G(t))^2, \\ G(t) = \sin(0.5\pi t) \\ y_i = x_i^{F(t)}, \text{ for } i=1, \dots, (m-1), \\ F(t) = 1 + 100\sin^4(0.5\pi t) \\ t = \lfloor \tau/\tau_T \rfloor / n_T \end{cases}$	$\begin{cases} x_{II} = (x_m, \dots, x_n) \\ x_i \in [0, 1], \\ i=1; n \end{cases}$

4.2 性能评价指标

采用逆向世代距离 $rGD(t)$ 作为评价DEMO算法性能的

指标,其定义如下^[22]:

$$rGD(t) = \frac{|\text{POF}^*(t)|}{|\text{POF}^*(t)|}, d_i = \min_{k=1}^m \sqrt{\sum_{j=1}^m (f_j^{*(i)} - f_j^k)^2} \quad (5)$$

其中, $\text{POF}^*(t)$ 为 t 时刻抽样的 Pareto 最优前沿, $Q(t)$ 为 t 时刻算法所得解集, $f_j^{*(i)}$ 为 $\text{POF}^*(t)$ 中第 i 个抽样点的第 j 个目标函数值。 $rGD(t)$ 既能评价所得解集的近似收敛程度, 又能评价解集的多样性。 $rGD(t)$ 值越小, 则表示算法获得解集的收敛性和多样性越好。

4.3 实验参数设置

采用实数制染色体编码, 模拟二进制交叉和多项式变异^[19], 参数设置如表 2 所列, 其中 n 为决策变量维数。 为研究非确定环境下各种动态变化的影响, 本文根据不同 DMOP 问题给出不同 (τ_T, n_T) 组合^[23], 对每个问题的每种组合均做了 30 次独立实验。 每次实验 4 种算法统一跟踪 100 次环境变化, 每种算法运行进化代数 = 环境变化次数 × 环境变化频率。

4.4 实验结果及其分析

表 3 是 4 种算法在不同 (τ_T, n_T) 组合下求解不同 DMOP 问题时所获得解集的 $rGD(t)$ 指标的统计结果, 该表将 DNSGA-II 缩写为 DNSGA2。

表 2 实验参数设置

参数	值
种群大小(N)	双目标:100;三目标:300
杂交概率	0.9
杂交分布指数	10
变异概率	1/n
变异分布指数	20
多样性引入比率 ^[11,20]	DDI: $\zeta=0.2$; RS: $pl=0.2$
记忆体大小 ^[21]	100
记忆串大小(B _{size})	双目标:5;三目标:12
环境变化次数	100
变化频率(τ_T)	5,10,15,20,25
变化剧烈程度(n_T)	5,10

其结果显示:

1) 在双目标 DMOP1 和 DMOP2 上, DNSGA-II-BM 获得的 $rGD(t)$ 指标具有最好的均值与方差; 在 3 个目标的 DMOP3 和 DMOP4 上, DNSGA-II-BM 获得的 $rGD(t)$ 指标具有最好的均值。 这表明对于不同问题以及不同的 (τ_T, n_T) 组合, DNSGA-II-BM 都具有最好的实验结果, 其获得解集的收敛性与多样性明显好于其它 3 种算法。

2) 对于每个测试问题, 当 (τ_T, n_T) 组合中 τ_T 值不变而 n_T 值减小时, DNSGA-II-BM 的均值变化不大, 但其它 3 种算法的均值都较大程度地变差, 这说明 DNSGA-II-BM 对不同环境变化剧烈程度都具有较好的适应能力, 其鲁棒性更好。

下面从每个测试问题的最后一组动态变化组合中抽出一组代表性实验, 画出 4 种算法不同时刻获得的解集, 对 4 种算

法的性能进行更深入的比较分析。

表 3 4 种算法获得解集的 $rGD(t)$ 指标的均值与方差

问题	τ_T, n_T	统计	DNSGA2	DNSGA2-RS	DNSGA2-CVM	DNSGA2-BM
10,10		均值	4.31E-01	7.88E-01	6.24E-01	2.81E-01
		方差	1.22E-01	3.61E-01	3.44E-01	8.40E-02
DMOP1	25,10	均值	3.67E-02	4.41E-02	3.52E-02	2.54E-02
		方差	6.37E-04	1.04E-03	5.80E-04	4.37E-04
	25,5	均值	1.65E-01	1.10E-01	1.23E-01	2.26E-02
		方差	1.82E-02	4.68E-03	1.02E-02	2.25E-03
5,10	均值	9.64E-02	6.45E-01	8.22E-02	8.17E-02	
	方差	5.32E-03	3.02E-02	5.43E-03	5.31E-03	
DMOP2	15,10	均值	1.51E-02	6.01E-01	1.54E-02	1.49E-02
		方差	4.01E-04	8.19E-02	4.01E-04	3.98E-04
	15,5	均值	3.80E-02	4.33E-01	2.94E-02	1.57E-02
		方差	8.08E-04	9.89E-02	5.77E-04	5.55E-04
20,10	均值	7.01E-02	5.76E-02	5.17E-01	4.98E-02	
	方差	6.23E-04	1.31E-04	9.73E-02	2.46E-04	
DMOP3	25,10	均值	5.57E-02	4.92E-02	5.77E-02	4.49E-02
		方差	1.23E-04	3.36E-05	1.89E-04	6.63E-05
	25,5	均值	7.61E-02	5.62E-02	1.49E-01	4.89E-02
		方差	1.87E-03	2.38E-04	2.46E-02	1.64E-03
20,10	均值	9.73E-02	8.60E-02	1.10E-01	7.60E-02	
	方差	3.16E-04	1.47E-04	9.73E-04	3.01E-04	
DMOP4	25,10	均值	8.35E-02	7.53E-02	8.65E-02	7.01E-02
		方差	9.72E-05	9.23E-05	1.55E-04	1.18E-04
	25,5	均值	1.02E-01	8.25E-02	1.69E-01	7.37E-02
		方差	8.43E-04	1.51E-04	1.86E-02	1.91E-03

4.4.1 DMOP1 问题

DMOP1 是目标数等于 2 的测试问题。 其解向量的某些分量 (x_n) 随 $G(t)$ 以正弦函数方式变化(见表 1), 故 DMOP1 的 $\text{POF}(t)$ 随时间变化。 但是, 任意时刻 t , 其 $\text{POF}(t)$ 均为 $f_2 = 1 - \sqrt{f_1}$, 不随时间变化。

4 种算法求解 DMOP1 的结果如图 3 所示。 图 3(a)~图 3(c)分别为 4 种算法在 3 个不同时刻获得的解集。 根据表 1 中 DMOP1 的正弦函数的变化特点, 这 3 个时刻所对应的环境变化幅度各不相同。 图 3(a)的时刻 $t=12.2$, 对应第 61 次环境变化(环境变化的编号为 $t \times n_T = 12.2 \times 5$)。 该次变化幅度较大, DNSGA-II-BM 获得解集能均匀地收敛到 Pareto 最优前沿 $\text{POF}(t)$, 其它 3 种算法的结果明显偏离 $\text{POF}(t)$ 。 图 3(b)的时刻 $t=13.2$, 对应第 66 次环境变化, 该次变化幅度微小, DNSGA-II-BM 获得的解集最接近 $\text{POF}(t)$, DNSGA-II-CVM 和 DNSGA-II 的结果大致相当, DNSGA-II-RS 因其较高的随机初始化比率 $(1 - pl = 0.8)$ 而不适宜微小的环境变化, 故其结果最差。 图 3(c)的时刻 $t=14.2$, 对应第 71 次环境变化, 该次变化幅度最大, DNSGA-II-BM 获得的解集依然能均匀地收敛到 $\text{POF}(t)$ 。 DNSGA-II-RS 与 DNSGA-II-CVM 的收敛性能大致相当, DNSGA-II 的结果最差。

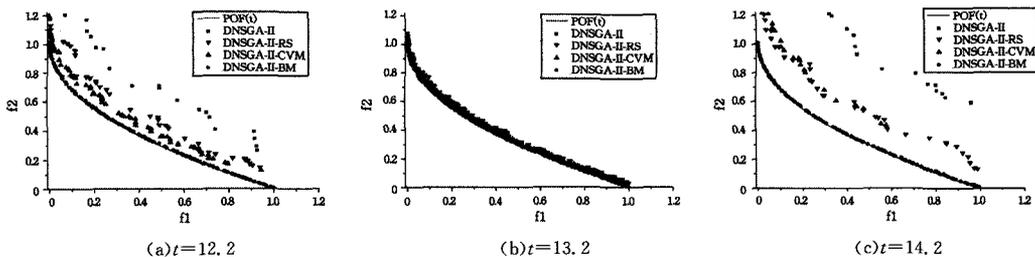


图 3 DMOP1 的实验结果

4.4.2 DMOP2 问题

DMOP2 的目标数等于 2, 其解向量的某些分量 (x_{11}) 随 $H(t)/4$ 以正弦函数方式变化 (见表 1), 故 DMOP2 的 $POS(t)$ 随时间变化。任意时刻 t , DMOP2 的 $POF(t)$ 为 $f_2 = 1 - (f_1/g)^{2^{H(t)}}$, 也随 $H(t)$ 以正弦函数方式变化。

4 种算法求解 DMOP2 的结果如图 4 所示。图 4(a)~图 4(c) 分别为 4 种算法在 3 个不同时刻获得的解集, 这 3 个时刻与图 3 的 3 个时刻相同。图 4(a) 对应的环境变化幅度较小, 此时 $POF(t)$ 是凸的, DNSGA-II-BM 获得的解集与 $POF(t)$ 最

为接近, 表明 DNSGA-II-BM 的收敛性与多样性好于其它 3 种算法。DNSGA-II-CVM 的结果与 DNSGA-II 大致相当, DNSGA-II-RS 因不适宜微小的环境变化而结果最差。图 4(b) 对应的环境变化幅度较大, 此时 $POF(t)$ 是一直线, DNSGA-II-BM 获得的解集最接近 $POF(t)$, 其它 3 种算法获得的解集明显远离 $POF(t)$ 。图 4(c) 对应的环境变化幅度最小, 此时 $POF(t)$ 是凹的, DNSGA-II-BM, DNSGA-II 和 DNSGA-II-CVM 都能较均匀地收敛到 $POF(t)$, DNSGA-II-RS 因不适宜此微小变化而结果最差。

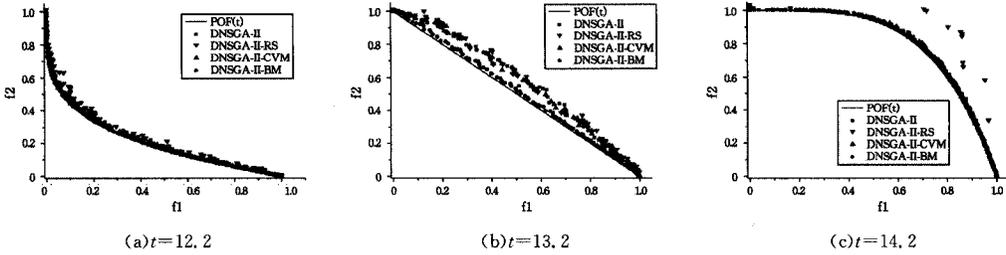


图 4 DMOP2 的实验结果

4.4.3 DMOP3 问题

DMOP3 是目标数 (m) 大于或等于 2 的测试问题 (本文实验令 $m=3$)。其解向量的某些分量 (x_{11}) 随 $G(t)$ 以正弦函数方式变化 (见表 1), 故 DMOP3 的 $POS(t)$ 随时间变化。但是, 任意时刻 t , 当 $m=3$ 时, DMOP3 的 $POF(t)$ 总是半径为 1 的 $1/8$ 球面 (在图 5 中以小黑点标识该球面), 不随时间变化。

图 5(a)~图 5(c) 分别为 4 种算法在 3 个不同时刻获得的解集。图 5(a) 的时刻 $t=10.2$, 对应第 51 次环境变化, 该次环境变化幅度较大, DNSGA-II-BM 的收敛性与多样性最好, 较好地均匀收敛到 $POF(t)$ 上。其它 3 种算法明显偏离了

$POF(t)$ 。DNSGA-II-RS 因较适宜于幅度稍大的变化而结果次之; DNSGA-II 结果第 3; DNSGA-II-CVM 因三维时记忆不准确而结果最差。图 5(b) 的时刻 $t=10.6$, 对应第 53 次环境变化, 该次变化幅度居中, DNSGA-II-BM 获得的解集仍最接近 $POF(t)$, 但此时其它 3 种算法获得的解集与 $POF(t)$ 的偏离程度有所减小。图 5(c) 的时刻 $t=11.2$, 对应第 56 次环境变化, 该次变化幅度最小, DNSGA-II-BM 的结果仍然与 $POF(t)$ 最接近, DNSGA-II 与 DNSGA-II-CVM 获得的解集也比较接近 $POF(t)$, DNSGA-II-RS 因不适宜于微小变化而结果最差。

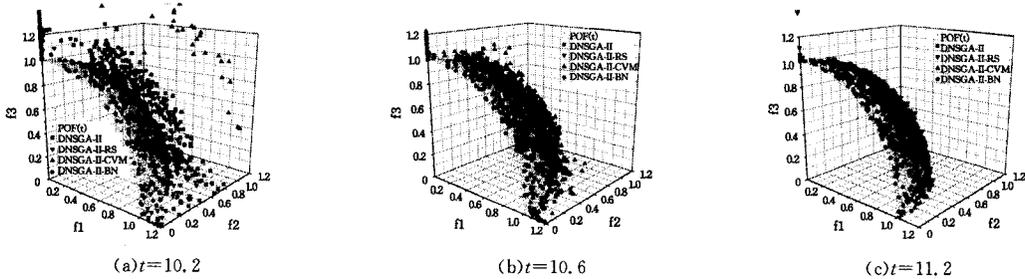


图 5 DMOP3 的实验结果

4.4.4 DMOP4 问题

DMOP4 是目标数 (m) 大于或等于 2 的测试问题 (本文令 $m=3$)。其解向量的某些分量 (x_{11}) 随 $G(t)$ 以正弦函数方式变

化 (见表 1), 故 DMOP4 的 $POS(t)$ 随时间变化。任意时刻 t , 当 $m=3$ 时, DMOP4 的 $POF(t)$ 为 $1/8$ 球面, 该球面半径大小随时刻 t 在 1 到 2 之间不断变化 (图 6 中以小黑点标识该球面)。

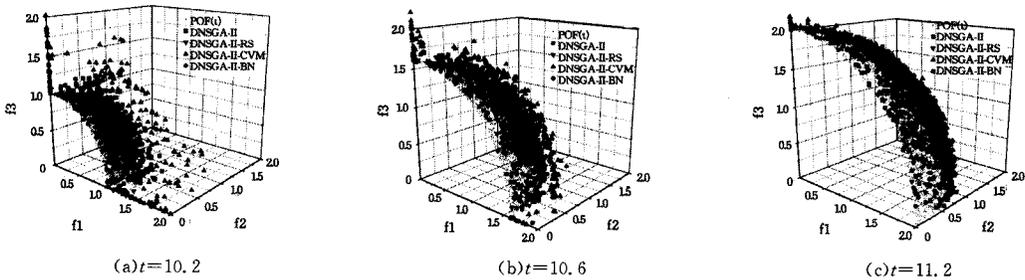


图 6 DMOP4 的实验结果

图 6(a)~图 6(c) 分别为 4 种算法在 3 个不同时刻获得的解集, 这 3 个时刻与图 5 的 3 个时刻相同。图 6(a) 对应的环境变化幅度较大, 该次环境的 $POF(t)$ 的球面半径为 1, DNSGA-II-BM 的收敛性与多样性最好, 较好地均匀收敛到

$POF(t)$; DNSGA-II-RS 比较适宜于幅度较大的变化, 其结果次之; DNSGA-II 结果第 3; DNSGA-II-CVM 因三维时的记忆不准确而结果最差。图 6(b) 的环境变化幅度居中, 该次环境的 $POF(t)$ 的球面半径为 1.59, DNSGA-II-BM 获得的解集仍

然最接近 $POF(t)$, DNSGA-II-RS 和 DNSGA-II 的结果有所改进, DNSGA-II-CVM 的结果依旧最差; 图 6(c) 的环境变化幅度最小, 此环境的 $POF(t)$ 的球面半径为 2, DNSGA-II-BM 的结果仍然与 $POF(t)$ 最接近, 其它 3 种算法获得的解集大致相当, 也比较接近 $POF(t)$ 。

通过 4.4.1 节—4.4.4 节的比较分析, 可进一步得出以下结论: DNSGA-II-BM 的记忆能力明显好于其它 3 种算法; 在不同问题以及不同环境变化下, DNSGA-II-BM 获得的解集都最接近于 Pareto 最优前沿 $POF(t)$, 且分布最为均匀, 表明 DNSGA-II-BM 在 4 种算法中具有最好的收敛性与多样性。

结束语 本文为 DEMO 提出了一种串式记忆 (BM) 方法。通过设计串式记忆体组织形式、快速的抽取与检索过程, BM 方法能有效地复用记忆体中的最优解集信息, 对新变化做出快速响应。实验结果表明, BM 方法相比其它 3 种方法 (即 DDI, RS 和 CVM) 具有更好的记忆能力。相应地, 集成了 BM 方法的 DNSGA-II-BM 算法所获得解集的收敛性与多样性也明显好于其它 3 种 DEMO 算法 (即 DNSGA-II, DNSGA-II-RS 和 DNSGA-II-CVM)。最后需要指出: DNSGA-II-BM 在三维目标问题上的记忆效果略逊于二维目标问题, 有待进一步提高, 这主要是因为基础算法 DNSGA-II 在三维目标问题上的收敛性与多样性不如二维目标问题理想。采用更好性能的 DEMO 算法作为 BM 方法的基础算法是我们今后的研究工作。

参 考 文 献

- [1] Coello Coello C A, Lamont G B, Van Veldhuizen D A. Evolutionary Algorithms for Solving Multi-Objective Problems (2nd ed) [M]. New York: Springer-Verlag, 2007
- [2] Farina M, Deb K, Amato P. Dynamic multiobjective optimization problems: Test cases, approximations, and applications [J]. IEEE Transaction on Evolutionary Computation, 2004, 8(5): 425-442
- [3] Goh C K, Tan K C. A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization [J]. IEEE Transaction on Evolutionary Computation, 2009, 13(1): 103-127
- [4] Branke J. Evolutionary Optimization in Dynamic Environments [M]. Boston: Kluwer Academic Publishers, 2002
- [5] Shang R H, Jiao L C, Gong M G, et al. An immune algorithm for dynamic multi-objective optimization [J]. Journal of Software, 2007, 18(11): 2700-2711 (in Chinese)
尚荣华, 焦李成, 公茂果, 等. 免疫克隆算法求解动态多目标优化问题 [J]. 软件学报, 2007, 18(11): 2700-2711
- [6] Liu M, Zeng W H. Memory enhanced dynamic multi-objective evolutionary algorithm based on decomposition [J]. 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 [J]. 软件学报, 2013, 24(7): 1571-1588
- [7] Muruganantham A, Tan K C, et al. Evolutionary dynamic multiobjective optimization via kalman filter prediction [J]. IEEE Transactions on Cybernetics, 2015, 5(99): 1-12
- [8] Hu C Y, Yao H, Yan X S. Multiple particle swarms coevolutionary algorithm for dynamic multi-objective optimization problems and its application [J]. Journal of Computer Research and Development, 2013, 50(6): 1313-1323 (in Chinese)
胡成玉, 姚宏, 颜雪松. 基于多粒子群协同的动态多目标优化算法及应用 [J]. 计算机研究与发展, 2013, 50(6): 1313-1323
- [9] Zhang S W, Li Z Y, Chen S M, et al. Dynamic multi-objective optimization algorithm based on ecological strategy [J]. Journal of Computer Research and Development, 2014, 51(6): 1313-1330 (in Chinese)
张世文, 李智勇, 陈少森, 等. 基于生态策略的动态多目标优化算法 [J]. 计算机研究与发展, 2014, 51(6): 1313-1330
- [10] Liu R C, Ma Y J, Zhang L, et al. Dynamic multi-objective immune optimization algorithm based on prediction strategy [J]. Chinese Journal of Computers, 2015, 38(8): 1544-1560 (in Chinese)
刘若辰, 马亚娟, 张浪, 等. 基于预测策略的动态多目标免疫优化算法 [J]. 计算机学报, 2015, 38(8): 1544-1560
- [11] Deb K, Rao U B N, Karthik S. Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling [C]//Proc of the 4th International Conference on Evolutionary Multi-criterion Optimization (EMO 2007). Berlin Heidelberg: Springer-Verlag, 2007: 803-817
- [12] Huang L, Suh I H, Abraham A. Dynamic multi-objective optimization based on membrane computing for control of time-varying unstable plants [J]. Information Sciences, 2011, 182(11): 2370-2391
- [13] Zhang Z H. Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control [J]. Applied Soft Computing, 2008, 8(2): 959-971
- [14] Nguyen S, Zhang M, et al. Automatic design of scheduling policies for dynamic multi-objective job shop scheduling via cooperative coevolution genetic programming [J]. IEEE Transactions on Evolutionary Computation, 2014, 18(2): 193-208
- [15] Ghannadpour S F, Noori S, et al. A multi-objective dynamic vehicle routing problem with fuzzy time windows: Model, solution and application [J]. Applied Soft Computing, 2014, 14(3): 504-527
- [16] Zhou A, Jin Y C, Zhang Q F. A population prediction strategy for evolutionary dynamic multiobjective optimization [J]. IEEE Transactions on Cybernetics, 2014, 44(1): 40-53
- [17] Zheng J H, Peng Z, Zhou J, et al. A prediction strategy based on guide-individual for dynamic multi-objective optimization [J]. Acta Electronica Sinica, 2015, 43(9): 1816-1825 (in Chinese)
郑金华, 彭舟, 邹娟, 等. 基于引导个体的预测策略求解动态多目标优化问题 [J]. 电子学报, 2015, 43(9): 1816-1825
- [18] Yang S X, Yao X. Population-based incremental learning with associative memory for dynamic environments [J]. IEEE Trans on Evolutionary Computation, 2008, 12(5): 542-561
- [19] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. IEEE Trans on Evolutionary Computation, 2002, 6(2): 182-197
- [20] Wang Y, Li B. Multi-strategy ensemble evolutionary algorithm for dynamic multi-objective optimization [J]. Memetic Computing, 2010, 2(1): 3-24
- [21] Koo W T, Goh C K, Tan K C. A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment [J]. Memetic Computing, 2010, 2(2): 87-110
- [22] Li X D, Branke J, Kirley M. On performance metrics and particle swarm methods for dynamic multiobjective optimization problems [C]//Proceedings of the 2007 Congress on Evolutionary Computation (CEC 2007). Singapore: IEEE Press, 2007: 576-583
- [23] Liu M, Zheng J H, et al. An adaptive diversity introduction method for dynamic evolutionary multiobjective optimization [C]//Proceedings of the 2014 Congress on Evolutionary Computation (CEC2014). Beijing: IEEE Press, 2014: 3160-3167