

# 控制与决策

Control and Decision

## 动态多目标优化:测试函数和算法比较

武燕, 石露露, 周艳

引用本文:

武燕, 石露露, 周艳. 动态多目标优化:测试函数和算法比较[J]. *控制与决策*, 2020, 35(10): 2372–2380.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2019.0236>

---

## 您可能感兴趣的其他文章

Articles you may be interested in

### 一种基于参考点约束支配的NSGA-III算法

A reference point constrained dominance-based NSGA-III algorithm

*控制与决策*. 2019, 34(2): 369–376 <https://doi.org/10.13195/j.kzyjc.2017.1067>

### 基于参考点的高维多目标粒子群算法

Reference-point-based particle swarm optimization algorithm for many-objective optimization

*控制与决策*. 2017, 32(4): 607–612 <https://doi.org/10.13195/j.kzyjc.2016.0387>

### 一种基于进化知识融合的多目标人工蜂群算法

A multi-objective artificial bee colony based on evolutionary knowledge integrated

*控制与决策*. 2017, 32(12): 2176–2182 <https://doi.org/10.13195/j.kzyjc.2016.1322>

### 基于环境Pareto支配选择策略的有约束多目标差分进化算法

Differential evolution algorithm based on environment Pareto dominated selection strategy in constrained multi-objective optimization problem

*控制与决策*. 2016, 31(1): 45–51 <https://doi.org/10.13195/j.kzyjc.2014.1885>

### 基于环境经济调度问题的空间自适应划分多目标优化

Environmental economic power dispatch based on multi-objective evolution algorithm with adaptive space partition

*控制与决策*. 2015, 30(11): 1974–1980 <https://doi.org/10.13195/j.kzyjc.2014.1037>

# 动态多目标优化：测试函数和算法比较

武燕<sup>1†</sup>, 石露露<sup>1</sup>, 周艳<sup>2</sup>

(1. 西安电子科技大学 数学与统计学院, 西安 710126; 2. 陆军边海防学院 基础部, 西安 710108)

**摘要:** 生活中存在大量的动态多目标优化问题, 应用进化算法求解动态多目标优化问题受到越来越多的关注, 而动态多目标测试函数对算法的评估起着重要的作用. 在已有动态多目标测试函数的基础上, 设计一组新的动态多目标测试函数. Pareto 最优解集和 Pareto 前沿面的不同变化形式影响着动态多目标测试函数的难易程度, 通过引入 Pareto 最优解集形状的变化, 结合已有的 Pareto 最优解集移动模式, 设计一组测试函数集. 基于提出的测试函数集, 对 3 个算法进行测试, 仿真实验结果表明, 所设计的函数给 3 个算法带来了挑战, 并展现出算法的优劣.

**关键词:** 动态多目标优化; 进化算法; 测试函数; 性能比较

中图分类号: TP18

文献标志码: A

DOI: 10.13195/j.kzyjc.2019.0236

开放科学(资源服务)标识码(OSID):



引用格式: 武燕, 石露露, 周艳. 动态多目标优化: 测试函数和算法比较 [J]. 控制与决策, 2020, 35 (10): 2372-2380.

## Dynamic multi-objective optimization: Test function and algorithm comparisons

WU Yan<sup>1†</sup>, SHI Lu-lu<sup>1</sup>, ZHOU Yan<sup>2</sup>

(1. School of Mathematics and Statistics, Xidian University, Xi'an 710126, China; 2. Department of the Foundation, Army Academy of Border and Coastal Defence, Xi'an 710108, China)

**Abstract:** In real world, there are many dynamic multi-objective optimization problems. In the field of evolutionary algorithms, there is a rapid increase in research work dealing with dynamic multi-objective optimization problems. However, dynamic multi-objective test functions play an important role in measuring algorithms performance. This paper proposes a set of new dynamic multi-objective test functions to evaluate the performance of algorithms based on the existing dynamic multi-objective test functions. The difficulty of dynamic multi-objective test functions is affected by shapes of Pareto-optimal solutions and Pareto-optimal front. According to this idea, a set of test functions is designed by introducing the change shape of Pareto-optimal solutions and combining with the existing shift mode of Pareto-optimal solutions. Three algorithms are tested on the proposed test functions. The simulation results show that the designed functions bring challenges to the algorithms and reveal the advantages and disadvantages of the algorithms.

**Keywords:** dynamic multi-objective optimization; evolutionary algorithm; test function; performance comparison

## 0 引言

现实生活中的许多优化问题都是动态多目标优化问题 (dynamic multi-objective optimization problems, DMOPs), 这类问题广泛存在于工业应用和科学研究领域, 如证券投资优化、控制器优化、医院资源管理以及机器学习. 因此, 解决动态多目标优化问题具有重要的实际意义. 但目前的算法, 还不能直接应用于实际问题, 动态多目标测试函数的设计就显得尤为重要. 本文设计一组新的测试函数集, 帮助研究者分析算法性能的优缺点. 这有助于研究者进一步对算法作出改进, 进而提升算法性能, 推进算法在现实生活

的应用.

动态多目标问题的模型<sup>[1]</sup>如下:

$$\begin{aligned} \min F(x, t) &= (f_1(x, t), \dots, f_M(x, t))^T. \\ \text{s.t. } h_i(x, t) &= 0, \quad i = 1, 2, \dots, n_h; \\ g_i(x, t) &\geq 0, \quad i = 1, 2, \dots, n_g; \\ x &\in \Omega_x, \quad t \in \Omega_t. \end{aligned} \quad (1)$$

其中:  $M$  是目标个数,  $F(x, t) : \Omega_x \times \Omega_t \rightarrow R^M$  是时间  $t$  下  $x$  的目标函数值,  $n_h$  和  $n_g$  分别是等式和不等式约束的个数,  $x$  是决策变量,  $\Omega_x$  是决策空间,  $t$  是时间变量,  $\Omega_t$  是时间空间.

收稿日期: 2019-03-02; 修回日期: 2019-05-09.

基金项目: 国家自然科学基金项目 (61772391).

责任编委: 陈家伟.

<sup>†</sup>通讯作者. E-mail: yanerch@163.com.

对于某一确定的环境  $t$ , 如果不存在解  $x' \in \Omega_x$ , 使得  $f_j(x') \leq f_j(x) (j = 1, 2, \dots, M)$  且  $F(x) \neq F(x')$ , 则称  $x \in \Omega_x$  为 Pareto 最优解集, 记作 PS. 称 Pareto 最优解集对应的目标函数值为 Pareto 前沿面, 记作 PF.

由模型 (1) 可以看出, 其决策变量、约束函数、目标函数均可能随时间发生变化, 导致 Pareto 最优解集和 Pareto 前沿面随时间发生变化. 这要求算法能够在环境变化后快速跟踪到新的 PF 和 PS. 文献 [1] 依据 PS 与 PF 是否随时间变化, 将 DMOPs 分成了以下 4 种类型.

类型 1: PS 随时间变化, 而 PF 不随时间变化;

类型 2: PS 和 PF 都随时间变化;

类型 3: PS 不随时间变化, 而 PF 随时间变化;

类型 4: 目标函数随时间变化, 但 PS 和 PF 都不随时间变化.

对于 DMOPs, 测试函数的难易程度不一, PS 和 PF 的不同变化形式是影响测试函数难易程度的重要因素, PS 变化越复杂测试函数就越难求解. 通过引入不同的 PF 变化和 PS 变化的方式, 研究者们已经设计出了一些动态多目标优化测试函数.

基于静态多目标问题 DTLZ<sup>[2]</sup> 和 ZDT<sup>[3]</sup>, 文献 [1] 设计了一套 FDA 动态多目标测试函数集, FDAs 中决策变量之间无相关性, PS 和 PF 的变化都较为简单. FDAs 测试函数的 PS 形状无变化, PF 的形状有轻微变化, 且 PS 和 PF 的移动模式为垂直移动.

在文献 [4] 中, 作者引入随机性的动态变化构造了一组测试函数 dMOPs, 使得所设计的测试函数更接近现实问题. 在 2013 年, 文献 [5] 引入决策变量的相关性使得动态问题的求解更加复杂, 同时作者引入新的 PS 的移动模式来丰富测试函数的动态变化形式, 依据 FDA 问题设计了一组新的动态多目标函数 ZJZs. 其中 PS 的移动模式有“花瓣”型、“水平 8 字”型和“水平心”型. 文献 [6] 在测试函数中引入了新的动态变化方式, 使得 PF 的移动模式增多, 如垂直移动、水平移动、对角移动和旋转, 并基于静态测试函数 UFs<sup>[7]</sup> 的正弦曲线和多项式两种 PS 类型, 设计出了一些新的测试函数 UDFs. 在文献 [8] 中, 为了使测试问题更具复杂性和挑战性, 作者从增加 PF 形状变化的角度出发, 提出了一套新的测试函数集 JYs, 使得 PF 的形状变化变得更加复杂. 除了提到的这些测试函数集, 还有一些其他的动态多目标优化测试函数<sup>[9-15]</sup>.

在所提出的测试函数中, FDAs 存在简单的 PF 和 PS 移动模式, dMOPs 通过引入随机变量使 PS 的改变具有随机性, ZJZs 在 PS 的移动模式上作出了改进,

UDFs 则侧重于 PF 移动模式的改进, 而 JYs 更侧重于 PF 形状变化的改进. 研究发现, PS 的形状变化几乎没有被引入到测试函数的设计中, 而 PS 形状的变化同样影响着测试函数的难度, 因此将 PS 的形状变化加入动态多目标测试函数的设计中非常有意义.

考虑到 PS 形状变化的重要性, 在 PS 移动的基础上, 结合 PS 形状的变化, 本文设计一组新的测试函数集. 本文利用新设计的动态多目标测试函数集, 结合两个性能指标, 通过实验来测评 3 个动态多目标优化算法的性能.

### 1 测试函数

现有的动态多目标测试函数通过引入不同的动态变化方式, 如 ZJZs 增加 PS 的移动模式, UDFs 丰富了 PF 的移动模式, JYs 引入了 PF 形状的不同变化形式, 使测试函数更加接近现实问题. 然而, 研究者对 PS 形状变化的研究并不多, 而 PS 形状的变化同样影响着动态多目标优化测试函数的难易程度. 因此, 考虑到 PS 形状变化对测试函数的重要性, 基于 PS 的不同形状变化和移动模式, 本文将提出一组 PS 移动与 PS 形状变化相结合的测试函数集.

$$SW : \begin{cases} \min F(x, t) = (f_1(x, t), \dots, f_M(x, t))^T; \\ f_1(x, t) = |x_1 - a|^H (1 + g_I(x, t)), \\ f_2(x, t) = |x_1 - a - b|^H (1 + g_{II}(x, t)). \end{cases} \quad (2)$$

基于式 (2) 所示的框架, 本文设计一些新的测试函数, 其中  $g_I(x, t)$  和  $g_{II}(x, t)$  均是和决策变量  $x$  和时间  $t$  相关的非负函数, 且  $g_I(x, t)$  和  $g_{II}(x, t)$  的最小值都为 0. 变量  $a$ 、 $b$  和  $H$  是与时间  $t$  相关的变量, 且  $t = (1/n_t) \lfloor (\tau/\tau_t) \rfloor$ . 其中:  $n_t$  为环境变化的强度,  $n_t$  越大意味着环境变化的强度越小;  $\tau$  为进化代数,  $\tau_t$  为环境变化的频率,  $\tau_t$  越大环境变化越慢,  $\tau_t$  越小则环境变化越快. 测试函数框架 SW 的 PF 随时间变化, 且 PF 为  $f_1^{1/H} + f_2^{1/H} = b, 0 \leq f_1^{1/H} \leq b$ . 本文设计的测试函数中的参数  $b$  和  $H$  是固定的, 故所有测试函数的 PFs 是同一个. 本文所设计的测试函数的 PFs 如图 1 所示.

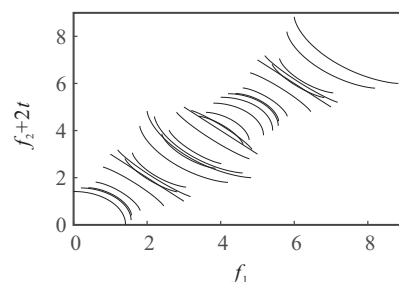


图 1 测试函数的 PFs

本文的测试函数主要是通过修改函数

$$g_I(x, t) = \sum_{k=1}^{\lfloor 2/n \rfloor} (x_{2k+1} - |A \sin(w\pi x_1)|^\alpha - \lambda_t x_1)^2, \quad (3)$$

$$g_{II}(x, t) = \sum_{k=1}^{\lfloor 2/n \rfloor} (x_{2k} - |A \sin(w\pi x_1)|^\alpha - \lambda_t x_1)^2 \quad (4)$$

来引入不同的PS形状变化和PS移动模式,且最小值都为0,故PS为 $a \leq x_1 \leq a+b, x_i = |A \sin(w\pi x_1)|^\alpha + \lambda_t x_1, i = 2, 3, \dots, n$ ,其中 $a$ 和 $b$ 是与时间 $t$ 相关的变量.

下面介绍3种PS形状变化形式和一种PS移动形式.首先固定 $A = 0.5, \lambda_t = 0, \alpha = 1$ ,取 $w = \lfloor 6 \sin(0.5\pi t) \rfloor$ ,图2(a)给出了 $x_2$ 和 $x_1$ 的PS形状波形数量的变化图,为了更清楚地观察曲线变化,对 $x_2$ 加

5t.由图2(a)可以观察到,PS形状的波形数量变化随时间发生变化,当 $w$ 增大时,PS的波形数量增多,反之, $w$ 越小,PS的波形数量跟着减少.固定 $w = 2, \alpha = 1$ 和 $\lambda_t = 0, A$ 为时间相关变量,不同的 $A$ 会影响PS曲线的振幅值,从而会有不同的PS形状.图2(b)为 $x_2$ 和 $x_1$ 的PS振幅的变化图,当 $A$ 增大时,振幅也增大,反之, $A$ 变小时,振幅也跟着减小.固定 $w = 1.5, A = 1$ 和 $\lambda_t = 0, \alpha$ 随着时间的变化而变化,图2(c)为 $\alpha$ 不同时的 $x_2$ 和 $x_1$ 的PS形状变化图.为了更清楚地观察曲线变化,对 $x_2$ 加2t,由图2(c)可以看出,随着 $\alpha$ 的改变,PS形状也发生着变化.固定 $w$ 和 $A$ 均为0,且 $\alpha = 1, \lambda_t$ 为时间相关变量, $\lambda_t$ 变化时, $x_2$ 和 $x_1$ 的PS随时间旋转,如图2(d)所示.

在PS的不同移动方式的基础上,结合PS不同的形状变化形式,介绍本文提出的9个动态多目标优化测试函数,表1展示了它们具有的特性.下面对这些测试函数进行具体描述.

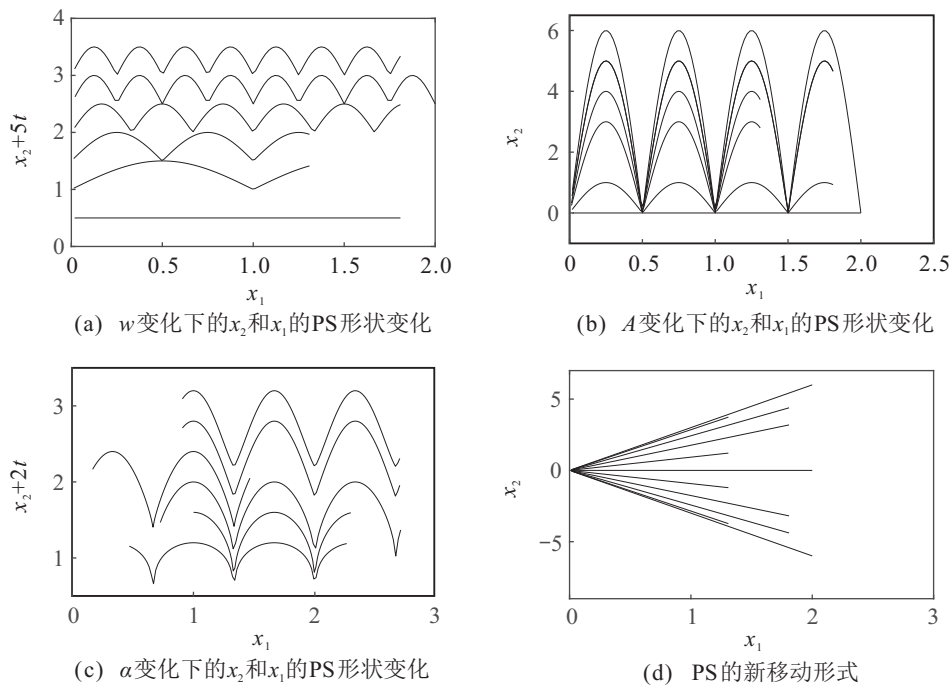


图2 PS的新变化方式

表1 测试函数SW1~SW9的特性

测试函数	变量相关性	PS移动变化	PS形状变化	随机性变化
SW1	无	无	无	无
SW2	无	垂直移动	无	无
SW3	有	无	波形数量变化	无
SW4	有	垂直移动	振幅变化	无
SW5	有	“水平8”型	波形数量变化	无
SW6	有	“花瓣”型	幂值变化	无
SW7	有	旋转	振幅变化	无
SW8	有	垂直移动	波形数量变化和幂值变化	有
SW9	有	旋转	无	有

$$\text{SW1: } \begin{cases} \min F(x, t) = (f_1(x, t), \dots, f_M(x, t))^T. \\ f_1(x, t) = |x_1 - a|^H(1 + g_I(x, t)); \\ f_2(x, t) = |x_1 - a - b|^H(1 + g_{II}(x, t)); \\ a = |\sin(1.5\pi t)|, b = 1 + |\cos(2\pi t)|; \\ H = 0.5 + \sin^2(0.5\pi t), y_i = x_i; \\ g_I(x, t) = \sum_{k=1}^{\lfloor 2/n \rfloor} y_{2k+1}^2, g_{II}(x, t) = \sum_{k=1}^{\lfloor 2/n \rfloor} y_{2k}^2; \\ x_1 \in [0, 3], x_i \in [-1, 1], i = 2, 3, \dots, n. \end{cases} \quad (5)$$

如式(5)所示, SW 是一个类型 2 的问题, 即 PS 和 PF 随时间变化, 其中 PS 为  $a \leq x_1 \leq a + b, x_i = 0, i = 2, 3, \dots, n$ . SW1 的  $x_2$  与  $x_1$  的 PS 如图 3(a) 所示, 为了便于观察, 对  $x_2$  加  $2t$  图示, 从图中可以发现, 不同时间变量  $t$  下的 PS 的长度不同.

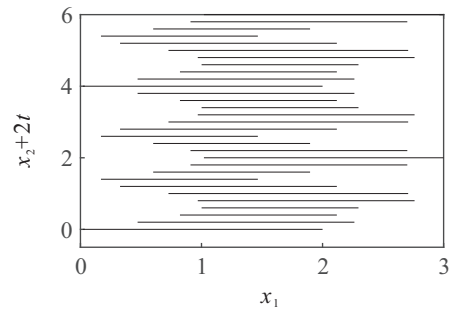
$$\text{SW2: } \begin{cases} \min F(x, t) = (f_1(x, t), \dots, f_M(x, t))^T. \\ f_1(x, t) = |x_1 - a|^H(1 + g_I(x, t)); \\ f_2(x, t) = |x_1 - a - b|^H(1 + g_{II}(x, t)); \\ a = |\sin(1.5\pi t)|, b = 1 + |\cos(2\pi t)|; \\ H = 0.5 + \sin^2(0.5\pi t); \\ y_i = x_i - G(t), G(t) = \sin(0.5\pi t); \\ g_I(x, t) = \sum_{k=1}^{\lfloor 2/n \rfloor} y_{2k+1}^2, g_{II}(x, t) = \sum_{k=1}^{\lfloor 2/n \rfloor} y_{2k}^2; \\ x_1 \in [0, 3], x_i \in [-1, 1], i = 2, 3, \dots, n. \end{cases} \quad (6)$$

如式(6)所示, 测试函数 SW2 的 PS 是  $a \leq x_1 \leq a + b, x_i = G(t) = \sin(0.5\pi t), i = 2, 3, \dots, n$ . 决策变量  $x_2$  与  $x_1$  的 PS 变化如图 3(b) 所示, 随着时间变量  $t$  值的改变, 图中 PS 在做有规律的垂直移动, 这是因为  $G(t)$  函数具有周期性.

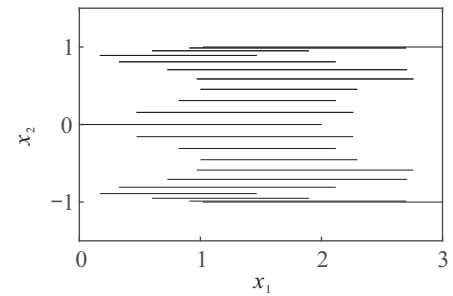
$$\text{SW3: } \begin{cases} \min F(x, t) = (f_1(x, t), \dots, f_M(x, t))^T. \\ f_1(x, t) = |x_1 - a|^H(1 + g_I(x, t)); \\ f_2(x, t) = |x_1 - a - b|^H(1 + g_{II}(x, t)); \\ a = |\sin(1.5\pi t)|, b = 1 + |\cos(2\pi t)|; \\ H = 0.5 + \sin^2(0.5\pi t); \\ y_i = x_i - |\sin(w\pi x_1)|, w = [6 \sin(0.5\pi t)]; \\ g_I(x, t) = \sum_{k=1}^{\lfloor 2/n \rfloor} y_{2k+1}^2, g_{II}(x, t) = \sum_{k=1}^{\lfloor 2/n \rfloor} y_{2k}^2; \\ x_1 \in [0, 3], x_i \in [-1, 1], i = 2, 3, \dots, n. \end{cases} \quad (7)$$

如式(7)所示的测试函数 SW3 增加了变量之

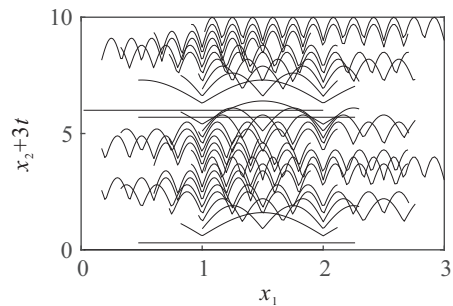
间的相关性,  $x_i (i = 2, 3, \dots, n)$  的取值受  $x_1$  取值的影响. SW3 的 PS 为  $a \leq x_1 \leq a + b, x_i = |\sin(w\pi x_1)|, i = 2, 3, \dots, n$ . 图 3(c) 所示为 SW3 的  $x_2$  与  $x_1$  的 PS 变化图, 为了更好地观察 SW3 的 PS 变化, 对  $x_2$  加  $3t$  图示. 由图 3(c) 观察得,  $x_2$  与  $x_1$  的 PS 曲线有波形数量的变化, 因为  $w$  是一个时间相关变量, 不同的  $w$  影响了  $x_2$  与  $x_1$  的 PS 形状,  $w$  值增大 PS 曲线波形数量增多,  $w$  值减小 PS 曲线波形数量减少. PS 的波形数量变化使得问题的解的多样性也在发生变化, 因此, 算法不仅要克服变量相关性的影响, 还要在动态环境下维持解的多样性.



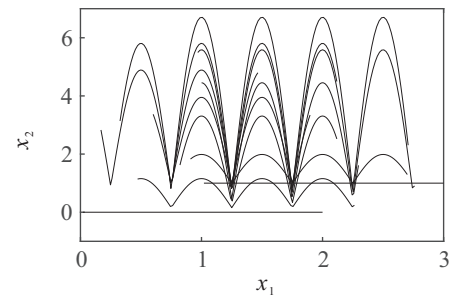
(a) SW1 在  $t=0, 0.1, \dots, 3$  时  $x_2$  和  $x_1$  的 PS 图



(b) SW2 在  $t=0, 0.1, \dots, 3$  时  $x_2$  和  $x_1$  的 PS 图



(c) SW3 在  $t=0, 0.1, \dots, 3$  时  $x_2$  和  $x_1$  的 PS 图



(d) SW4 在  $t=0, 0.1, \dots, 1$  时  $x_2$  和  $x_1$  的 PS 图

图 3 SW1 ~ SW4 的  $x_2$  和  $x_1$  的 PS 图



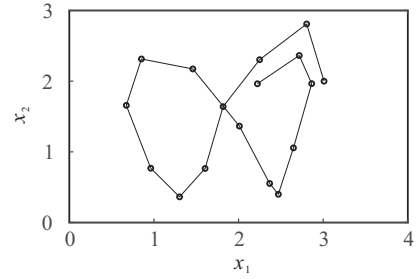
$$\text{SW4: } \left\{ \begin{array}{l} \min F(x, t) = (f_1(x, t), \dots, f_M(x, t))^T. \\ f_1(x, t) = |x_1 - a|^H(1 + g_I(x, t)); \\ f_2(x, t) = |x_1 - a - b|^H(1 + g_{II}(x, t)); \\ a = |\sin(1.5\pi t)|, b = 1 + |\cos(2\pi t)|; \\ H = 0.5 + \sin^2(0.5\pi t); \\ y_i = x_i - |A \cos(2\pi x_1)| - G(t); \\ A = \lfloor 6 \sin(\pi t) \rfloor, G(t) = \sin(0.5\pi t); \\ g_I(x, t) = \sum_{k=1}^{\lfloor 2/n \rfloor} y_{2k+1}^2, g_{II}(x, t) = \sum_{k=1}^{\lfloor 2/n \rfloor} y_{2k}^2; \\ x_1 \in [0, 3], x_i \in [-1, 2], i = 2, 3, \dots, n. \end{array} \right. \quad (8)$$

测试函数SW4如式(8)所示,其PS为 $a \leq x_1 \leq a + b, x_i = |A \cos(2\pi x_1)| + G(t), i = 2, 3, \dots, n$ . PS不仅在做垂直移动,且其波形的振幅也发生变化,这两种PS变化的结合丰富了测试函数的PS变化形式. SW4在 $t = 0, 0.1, \dots, 1$ 上的 $x_2$ 与 $x_1$ 的PS变化如图3(d)所示. 变量A是个时间相关变量,且决定着PS曲线的振幅,A越大振幅越大,A越小振幅越小,从图3(d)中可以清晰地看到PS形状的振幅变化. PS形状的振幅变化与PS垂直移动的结合使得算法解决SW4测试函数变得更有难度.

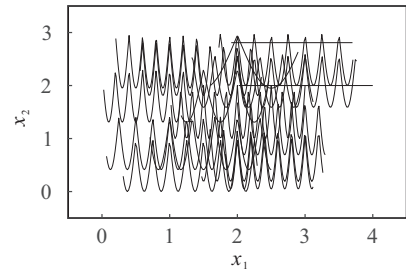
$$\text{SW5: } \left\{ \begin{array}{l} \min F(x, t) = (f_1(x, t), \dots, f_M(x, t))^T. \\ f_1(x, t) = |x_1 - a|^H(1 + g_I(x, t)); \\ f_2(x, t) = |x_1 - a - b|^H(1 + g_{II}(x, t)); \\ a = 1 + \cos(1.5\pi t), b = 1 + |\cos(2\pi t)|; \\ H = 0.5 + \sin^2(0.5\pi t); \\ y_i = x_i + |\sin(w\pi x_1)| - G(t); \\ w = \lfloor 6 \sin(0.5\pi t) \rfloor, G(t) = \sin(3\pi t) + 2; \\ g_I(x, t) = \sum_{k=1}^{\lfloor 2/n \rfloor} y_{2k+1}^2, g_{II}(x, t) = \sum_{k=1}^{\lfloor 2/n \rfloor} y_{2k}^2; \\ x_1 \in [0, 4], x_i \in [0, 3], i = 2, 3, \dots, n. \end{array} \right. \quad (9)$$

式(9)所示的是测试函数SW5,其PS为 $a \leq x_1 \leq a + b, x_i = G(t) - |\sin(w\pi x_1)|, i = 2, 3, \dots, n$ . 为了增加测试函数的难度,该测试问题将PS形状变化和PS的“水平8”移动方式结合在一起. 其中w值决定了PS曲线的波形数量,w值越大PS曲线波形数量越多. 如图4(a)和图4(b)是SW5在 $t = 0, 0.1, \dots, 1.6$ 时PS均值点和 $x_2$ 与 $x_1$ 的PS图. 在PS形状变化基础上增加PS移动方式的测试函数,无疑更考验算法快速应对复杂环境变化的能力和动态环境下算法的收敛

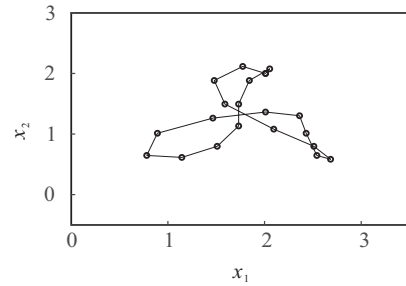
能力.



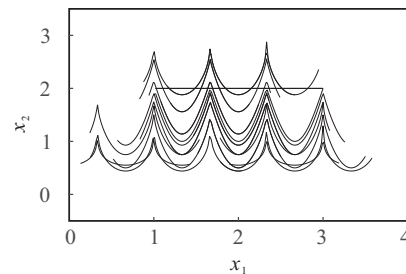
(a) SW5在 $t=0, 0.1, \dots, 1.6$ 时的PS均值点图



(b) SW5在 $t=0, 0.1, \dots, 1.6$ 的PS图



(c) SW6在 $t=0, 0.1, \dots, 2$ 时的PS均值点图



(d) SW6在 $t=0, 0.1, \dots, 2$ 时 $x_2$ 和 $x_1$ 的PS图

图4 SW5, SW6的 $x_2$ 和 $x_1$ 的PS图

$$\text{SW6: } \left\{ \begin{array}{l} \min F(x, t) = (f_1(x, t), \dots, f_M(x, t))^T. \\ f_1(x, t) = |x_1 - a|^H(1 + g_I(x, t)); \\ f_2(x, t) = |x_1 - a - b|^H(1 + g_{II}(x, t)); \\ a = 1 + \cos(1.5\pi t) \sin(0.5\pi t); \\ b = 1 + |\cos(2\pi t)|, H = 0.5 + \sin^2(0.5\pi t); \\ y_i = x_i + |\cos(1.5\pi x_1)|^\alpha - G(t); \\ \alpha = |\sin(1.5\pi t)|; \\ G(t) = \cos(1.5\pi t) \cos(0.5\pi t) + 2; \\ g_I(x, t) = \sum_{k=1}^{\lfloor 2/n \rfloor} y_{2k+1}^2, g_{II}(x, t) = \sum_{k=1}^{\lfloor 2/n \rfloor} y_{2k}^2; \\ x_1 \in [0, 4], x_i \in [0, 3], i = 2, 3, \dots, n. \end{array} \right. \quad (10)$$

如式(10)所示,测试函数SW6的PS为  $a \leq x_1 \leq a+b, x_i = G(t) - |\cos(1.5\pi x_1)|^\alpha, i = 2, 3, \dots, n$ . SW6结合了PS的“花瓣式”移动和PS形状变化,随时间变化的 $\alpha$ 影响着PS形状. 图4(c)和图4(d)是SW6在  $t = 0, 0.1, \dots, 2$ 时PS均值点图和  $x_2$ 与 $x_1$ 的PS图,由图4(c)和图4(d)可知PS随时间发生着变化.

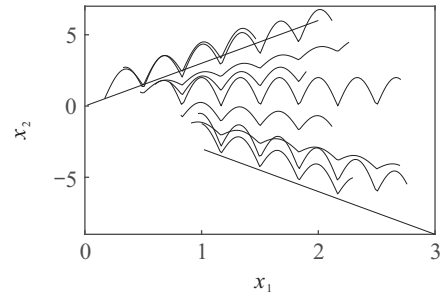


图5 SW7在  $t = 0, 0.1, \dots, 1$ 时  $x_2$ 和 $x_1$ 的PS图

$$\text{SW7: } \begin{cases} \min F(x, t) = (f_1(x, t), \dots, f_M(x, t))^T. \\ f_1(x, t) = |x_1 - a|^H(1 + g_I(x, t)); \\ f_2(x, t) = |x_1 - a - b|^H(1 + g_{II}(x, t)); \\ a = |\sin(1.5\pi t)|, b = 1 + |\cos(2\pi t)|; \\ H = 0.5 + \sin^2(0.5\pi t); \\ y_i = x_i - \lambda_t x_1 - |A \cos(3\pi x_1)|; \\ A = \sin(\pi t), \lambda_t = |2 \cos(1.5\pi t)|; \\ g_I(x, t) = \sum_{k=1}^{\lfloor 2/n \rfloor} y_{2k+1}^2, g_{II}(x, t) = \sum_{k=1}^{\lfloor 2/n \rfloor} y_{2k}^2; \\ x_1 \in [0, 3], x_i \in [0, 5], i = 2, 3, \dots, n. \end{cases} \quad (11)$$

测试函数SW7如式(11)所示,PS为  $a \leq x_1 \leq a + b, x_i = \lambda_t x_1 + |A \cos(3\pi x_1)|, i = 2, 3, \dots, n$ ,其中  $\lambda_t \geq 0$ 是一个时间相关变量. 它影响着  $x_2$ 与 $x_1$ 的PS曲线的斜率,  $\lambda_t$ 越大曲线越陡,  $\lambda_t$ 越小曲线越平缓,因此产生了PS移动模式之一旋转. 图5所示的是SW7的  $x_2$ 与 $x_1$ 的PS图,由图5可知,SW7结合了PS形状的振幅变化和PS的旋转,两者的结合无疑增加了测试问题的难度.

如式(12)所示,SW8中引入了随机整数  $R_1 \in (0, 10]$ 和  $R_2 \in [0, 30]$ ,使得  $x_2$ 与 $x_1$ 的PS曲线的周期变化和幂变化具有随机性. 当环境发生一次变化,随机数  $R_1$ 和  $R_2$ 便改变一次,它们会影响PS的形状,这个随机机制使得算法处理这个问题变得更加困难. 当  $R_2 = 0$ 时,变量之间保持独立性,此时,PS为  $x_i = 2 + G(t), i = 2, 3, \dots, n$ ;当  $R_1 \neq 0$ 且  $R_2 \neq 0$ 时,变量之间有相关性,此时的PS变为  $x_i = |\sin(w\pi x_1)|^\alpha + 1 + G(t), i = 2, 3, \dots, n$ . 由于变化的随机性,本文无法画出SW8的PS图形.

前面提到的测试函数SW3~SW8的PS曲线类型是正弦曲线类型,下面提出的测试函数的PS类型是多项式类型.

$$\text{SW8: } \begin{cases} \min F(x, t) = (f_1(x, t), \dots, f_M(x, t))^T. \\ f_1(x, t) = |x_1 - a|^H(1 + g_I(x, t)); \\ f_2(x, t) = |x_1 - a - b|^H(1 + g_{II}(x, t)); \\ a = |\sin(1.5\pi t)|, b = 1 + |\cos(2\pi t)|; \\ H = 0.5 + \sin^2(0.5\pi t); \\ y_i = x_i - |\sin(w\pi x_1)|^\alpha - 1 - G(t); \\ w = \frac{R_1}{2}, \alpha = \frac{R_2}{10}, G(t) = \sin(0.5\pi t); \\ g_I(x, t) = \sum_{k=1}^{\lfloor 2/n \rfloor} y_{2k+1}^2, g_{II}(x, t) = \sum_{k=1}^{\lfloor 2/n \rfloor} y_{2k}^2; \\ x_1 \in [0, 3], x_i \in [0, 3], i = 2, 3, \dots, n. \end{cases} \quad (12)$$

$$\text{SW9: } \begin{cases} \min F(x, t) = (f_1(x, t), \dots, f_M(x, t))^T. \\ f_1(x, t) = |x_1 - a|^H(1 + g_I(x, t)); \\ f_2(x, t) = |x_1 - a - b|^H(1 + g_{II}(x, t)); \\ a = |\sin(0.5\pi t)|, b = 1 + |\cos(2\pi t)|; \\ H = 0.5 + \sin^2(0.5\pi t); \\ y_i = x_i - 3 - G(t) + \lambda_t x_1^\alpha; \\ \lambda_t = \frac{R_1}{10}, \alpha = \frac{R_2}{10}, G(t) = \sin(0.5\pi t); \\ g_I(x, t) = \sum_{k=1}^{\lfloor 2/n \rfloor} y_{2k+1}^2, g_{II}(x, t) = \sum_{k=1}^{\lfloor 2/n \rfloor} y_{2k}^2; \\ x_1 \in [0, 3], x_i \in [-1, 4], i = 2, 3, \dots, n. \end{cases} \quad (13)$$

SW9如式(13)所示,结合了PS的垂直移动及旋转和PS形状的变化,变量  $\lambda_t$ 决定了  $x_2$ 与 $x_1$ 的PS曲线的倾斜角度,  $\alpha$ 则决定着PS形状的弯曲程度. SW9中引入了两个随机整数  $R_1 \in (0, 10]$ 和  $R_2 \in (0, 10]$ ,当环境变化一次,随机数  $R_1$ 和  $R_2$ 也会重新选取随机数. 当  $R_1 = 0$ 时,变量之间保持独立性,此时的PS为  $x_i = 3 + G(t), i = 2, 3, \dots, n$ ;当  $R_1 \neq 0$ 时,变量之间

存在相关性,此时的PS变为 $x_i = 3 + G(t) - \lambda_t x_{i1}^\alpha, i = 2, 3, \dots, n$ .

## 2 比较算法和性能指标

### 2.1 比较算法

本文使用新提出的测试函数来测评3个动态多目标优化进化算法的性能. 这3个算法分别是PPS<sup>[5]</sup>、SGEA<sup>[16]</sup>和DSS<sup>[17]</sup>算法.

PPS算法是文献[5]提出的一种基于预测的方法. 首先,PS的信息被分为两部分:群体中心和流形. 基于群体中心和流形的历史信息,PPS利用一个单变量自回归模型来预测下一代群体的中心和流形. 当环境发生变化时,根据预测的群体中心和流形,PPS会产生新环境下的初始群体.

当检测到环境变化时,SGEA算法利用新旧环境信息,预测下一群体的移动方向. 环境变化后,SGEA利用预测方向和最远距离选择方法重新初始化群体. 与传统的遗传算法一样,SGEA利用环境选择来保护下一代的好解.

DSS算法利用旧信息预测下一个群体的移动方向,当检测到环境变化后,DSS通过预测方向和局部搜索重新初始化群体. 另外,DSS中引入了一个类似于梯度的加速策略,提高了算法的收敛性.

### 2.2 性能指标

好的比较实验能够帮助研究者更好地比较算法的性能和测评测试函数的难度. 本实验中选用了两个多目标优化中的性能度量IGD<sup>[18]</sup>和HVD<sup>[19]</sup>的改进指标 $\overline{IGD}$ <sup>[5]</sup>和 $\overline{HVD}$ ,用来度量动态多目标优化进化算法的性能并评价所提出测试函数的难度,这两个性能度量能够同时度量算法的收敛性和分布性.

#### 2.2.1 改进指标 $\overline{IGD}$

设PF\*是真实的Pareto前沿面的一组均匀分布的点,PF是获得的Pareto前沿面的一组近似解. IGD的计算方法如下:

$$IGD = \frac{1}{n_{PF^*}} \sum_{i=1}^{n_{PF^*}} d_i. \quad (14)$$

其中: $n_{PF^*} = |PF^*|$ , $d_i$ 是PF\*的第*i*个成员与PF的欧氏距离. IGD测量算法所得的解的收敛性和分布性,IGD值越小越好. 一个好的IGD值说明一个算法同时在分布性和收敛性两方面表现好.

为了度量动态多目标优化算法的性能,文献[5]将IGD度量改进为

$$\overline{IGD} = \frac{1}{T_n} \sum_{T=1}^{T_n} IGD(T). \quad (15)$$

其中:IGD(*T*)是在第*T*个环境下的IGD值, $T_n$ 是环境总数.

#### 2.2.2 改进指标 $\overline{HVD}$

HVD测量的是算法得到的Pareto前沿面和真实Pareto前沿面的HV的差,其中HV表示的是由解集中的个体与参考点在目标空间中所围成的超立方体的体积. HVD值越小越好,越小说明得到的Pareto前沿面越接近真实Pareto前沿面. HVD的计算方法如下:

$$HVD = HV(PF) - HV(PF^*). \quad (16)$$

其中:HV(PF)是算法求得的Pareto前沿面的HV值,HV(PF\*)是真实Pareto前沿面的HV值,HVD的参考点为 $(z_1 + 0.5, z_2 + 0.5, \dots, z_M + 0.5)$ ,其中 $z_j$ 是真实Pareto前沿面PF\*的第*j*个目标的最大值, $M$ 是目标函数个数.

同样,HVD的度量改进为

$$\overline{HVD} = \frac{1}{T_n} \sum_{T=1}^{T_n} HVD(T). \quad (17)$$

其中HVD(*T*)是在第*T*个环境下的HVD值.

## 3 仿真实验研究

### 3.1 参数设置

本文实验中所使用的动态多目标优化算法的参数与文献[5,16-17]的参数一致. 下面将一些统一的参数作一说明:实验中,测试函数的决策变量个数*n*均为11. 3个算法的种群规模*N*均取100. 实验的终止条件是进化代数 $\tau$ 达到 $50 + 3n_t\tau_t$ ,实验中统一取环境变化频率 $\tau_t = 10$ ,环境变化强度 $n_t = 10$ . 每个算法对每个提出的测试函数都进行20次独立实验.

### 3.2 实验结果分析

基于新提出的测试函数,表2和表3分别记录了算法PPS、SGEA和DSS在两种性能度量 $\overline{IGD}$ 和 $\overline{HVD}$ 上的实验结果,表中加粗的值是这3个算法在该测试函数上表现最优的度量值.

表2是3个算法在仿真实验中得到的 $\overline{IGD}$ 的均值和方差, $\overline{IGD}$ 的均值反映了一个算法所得解的多样性和收敛性. 通过观察表中的实验结果可知,除了在测试函数SW1上的SGEA结果略优于DSS外,在一些更复杂的问题上,DSS的 $\overline{IGD}$ 值总是最好的,而表现最差的是PPS. 这说明DSS算法的收敛性和多样性在这3个算法中是最好的. 由表2还可知,3个算法在这些具有不同难度的问题上的表现基本具有一致性:测试函数难度增加,3个算法的结果都会变差.



表2 PPS、SGEA和DSS算法运行20次后的 $\overline{IGD}$ 度量的均值和方差

测试函数	PPS	SGEA	DSS
SW1	0.033 4(0.008 4)	<b>0.012 4(0.001 2)</b>	0.018 8(0.001)
SW2	0.113 1(0.059)	0.021 2(0.000 8)	<b>0.019 2(0.003 1)</b>
SW3	0.180 8(0.025 8)	0.122 4(0.020 2)	<b>0.097 1(0.029 8)</b>
SW4	1.794 7(0.185 7)	0.199 9(0.026 9)	<b>0.146 (0.045 1)</b>
SW5	0.477 7(0.05)	0.210 2(0.028 9)	<b>0.042 9(0.174 2)</b>
SW6	0.324 7(0.031 7)	0.133 6(0.010 9)	<b>0.055 4(0.006 3)</b>
SW7	0.793 6(0.053)	0.629 2(0.055 3)	<b>0.045 5(0.021 2)</b>
SW8	0.283(0.039 7)	0.154 3(0.043 7)	<b>0.040 3(0.014)</b>
SW9	0.400 2(0.061 9)	0.183 5(0.060 9)	<b>0.061 4(0.013 6)</b>

表3 PPS、SGEA和DSS算法运行20次后的 $\overline{HVD}$ 度量的均值和方差

测试函数	PPS	SGEA	DSS
SW1	0.118 4(0.030 3)	<b>0.059 5(0.006 6)</b>	0.067 7(0.003 6)
SW2	0.439 3(0.277)	0.069 9(0.003 9)	<b>0.060 3(0.008 2)</b>
SW3	0.741 3(0.121 4)	<b>0.565(0.077 3)</b>	0.597 5(0.163 3)
SW4	19.286 8(3.427 2)	2.366 5(0.700 5)	<b>1.081 6 (0.694)</b>
SW5	2.495 2(0.400 7)	0.953 8(0.158 7)	<b>0.204 4(0.036 7)</b>
SW6	1.647 5(0.235 2)	0.742 4(0.059 1)	<b>0.215 6(0.053 3)</b>
SW7	7.959 2(1.388 2)	3.573 5(1.498 8)	<b>0.171 9(0.075 6)</b>
SW8	1.455 5(0.286 7)	0.748 1(0.163 3)	<b>0.138 9(0.058 9)</b>
SW9	2.038 2(0.57)	0.641 4(0.245 6)	<b>0.244 7(0.089 2)</b>

测试函数增加PS形状变化、PS移动模式变化和随机性,会使函数的求解难度增加,也会给算法带来挑战。对于SW1和SW2,由于SW2增加了PS垂直移动,SW2的求解显然比SW1更难。由表2中的数据可知,3个算法求解SW2问题的结果显然差于SW1的结果。SW3在SW1的基础上增加了PS波形数量的形状变化,增加PS形状变化会提升函数的难度,表中实验结果显示,3个算法在SW3上的表现均差于SW1。同样地,SW4在SW2的基础上增加了PS的振幅形状变化,实验结果也验证了SW4的求解难度大于SW2。相比SW3和SW5增加了“水平8”的PS移动变化,3个算法在SW5上的实验结果都劣于在SW3上的实验结果。SW4~SW7测试函数均结合了PS的形状变化和PS移动,由表2中的实验数据可知,相比只增加PS移动或PS形状变化,PS形状变化和PS移动的结合

使问题变得更有难度。SW8和SW9中均引入了随机性,SW8中引入的是形状随机,相比SW3的实验结果可得,测试函数中引入形状随机比引入有规律的形状变化对算法而言更具有挑战性。SW9中增加了移动随机,相比SW2在表2中的数据可知,对算法而言,测试函数中引入随机移动比引入固定的移动模式更具有挑战性。

相比SW1和SW2,测试函数SW3~SW9均存在变量相关性的特性。观察表2可知,PPS、SGEA和DSS在SW3~SW9上的实验结果均比在SW1、SW2上的结果差。这说明,变量相关性对算法也是一个挑战。

表3是PPS、SGEA和DSS这3个算法在实验中获得 $\overline{HVD}$ 的均值和方差。3个算法在 $\overline{HVD}$ 上的表现和在 $\overline{IGD}$ 上的表现基本一致。与此同时,测试函数难度的增加,如PS形状发生变化,使得算法的 $\overline{HVD}$ 值变差,这也与 $\overline{IGD}$ 值保持一致。由表3可知,DSS在测试函数SW2和SW4~SW9上的表现优于PPS和SGEA。但在测试函数SW1和SW3上,算法SGEA的表现要比DSS和PPS好。由表3中的数据可以看出,虽然SGEA在SW1和SW3上的结果比DSS好,但DSS的结果与SGEA的结果很接近。

## 4 结论

动态多目标优化测试函数的设计是动态多目标优化研究的一个重要方向。本文从PS形状变化的角度,结合已有的PS移动模式,设计了一组新的测试函数集。本文利用所提出的测试函数集对3个具有代表性的算法进行了测试,仿真实验结果与已有文章的结论一致,这说明所设计的测试函数能够有效地对算法进行评估。实验结果同样说明,引入PS形状的复杂变化,使动态多目标优化测试函数的求解变得更加困难,且这些测试函数给算法带来了挑战。在未来的研究工作中,动态多目标优化测试函数的设计不仅要考虑动态问题的复杂性,还要考虑动态问题是否更接近现实。

## 参考文献(References)

- [1] Farina M, Deb K, Amato P. Dynamic multiobjective optimization problems: Test cases, approximations, and applications[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(5): 425-442.
- [2] Deb K, Thiele L, Laumanns M, et al. Scalable test problems for evolutionary multiobjective optimization[C]. Evolutionary Multiobjective Optimization. London: Springer, 2005: 105-145.

- [3] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results[J]. *Evolutionary Computation*, 2000, 8(2): 173-195.
- [4] 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.
- [5] Zhou A M, Jin Y C, Zhang Q F. A population prediction strategy for evolutionary dynamic multiobjective optimization[J]. *IEEE Transactions on Cybernetics*, 2013, 44(1): 40-53.
- [6] Biswas S, Das S, Suganthan P N, et al. Evolutionary multiobjective optimization in dynamic environments: A set of novel benchmark functions[C]. *Proceedings of IEEE Congress on Evolutionary Computation*. Beijing: IEEE, 2014: 3192-3199.
- [7] Zhang Q F, Zhou A M, Zhao S Z, et al. Multiobjective optimization test instances for the cec 2009 special session and competition[R]. University of Essex: The School of Computer Science and Electronic Engineering, 2008: 1-30.
- [8] Jiang S Y, Yang S X. Evolutionary dynamic multiobjective optimization benchmarks and algorithm comparisons[J]. *IEEE Transactions on Cybernetics*, 2017, 47(1): 198-211.
- [9] Jin Y C, Sendhoff B. Constructing dynamic optimization test problems using the multi-objective optimization concept[C]. *Proceedings of the Evoworkshops 2004*. Berlin: Springer, 2004: 525-536.
- [10] 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.
- [11] 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, 181(11): 2370-2391.
- [12] Helbig M, Engelbrecht A P. Benchmarks for dynamic multi-objective optimisation[C]. *IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments*. Singapore: IEEE, 2013: 84-91.
- [13] Jiang S Y, Yang S X. A benchmark generator for dynamic multi-objective optimization problems[C]. *The 14th UK Workshop on Computational Intelligence*. Bradford: IEEE, 2014: 1-8.
- [14] Gee S B, Tan K C, Abbass H A. A benchmark test suite for dynamic evolutionary multiobjective optimization[J]. *IEEE Transactions on Cybernetics*, 2017, 47(2): 461-472.
- [15] Zille H, Kottenhahn A, Mostaghim S. Dynamic distance minimization problems for dynamic multi-objective optimization[C]. *Proceedings of IEEE Congress on Evolutionary Computation*. San Sebastian: IEEE, 2017: 952-959.
- [16] Jiang S Y, Yang S X. A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2017, 21(1): 65-82.
- [17] Wu Y, Jin Y C, Liu X X. A directed search strategy for evolutionary dynamic multiobjective optimization[J]. *Soft Computing*, 2015, 19(11): 3221-3235.
- [18] Yuan Y, Xu H, Wang B, et al. Balancing convergence and diversity in decomposition-based many-objective Optimizers[J]. *IEEE Transactions on Evolutionary Computation*, 2016, 20(2): 180-198.
- [19] Zhou A M, Jin Y C, Zhang Q F, et al. Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization[C]. *International Conference on Evolutionary Multi-Criterion Optimization*. Berlin: Springer, 2007: 832-846.

### 作者简介

武燕(1976—),女,副教授,从事组合优化、动态优化等研究, E-mail: yanerch@163.com;

石露露(1994—),女,硕士生,从事动态优化、多目标优化的研究, E-mail: shi12lu@163.com;

周艳(1973—),女,副教授,从事应用数学等研究, E-mail: zylxb2000@163.com.

(责任编辑:孙艺红)