

# 一类新型动态多目标鲁棒进化优化方法

陈美蓉<sup>1,2</sup> 郭一楠<sup>1</sup> 巩敦卫<sup>1</sup> 杨振<sup>1</sup>

**摘要** 传统动态多目标优化问题 (Dynamic multi-objective optimization problems, DMOPs) 的求解方法, 通常需要在新的环境下, 通过重新激发寻优过程, 获得适应环境的 Pareto 最优解. 这可能导致较高的计算代价和资源成本, 甚至无法在有限时间内执行该优化解. 由此, 提出一类寻找动态鲁棒 Pareto 最优解集的进化优化方法. 动态鲁棒 Pareto 解集是指某一时刻下的 Pareto 较优解可以以一定稳定性阈值, 逼近未来多个连续动态环境下的真实前沿, 从而直接作为这些环境下的 Pareto 解集, 以减小计算代价. 为合理度量 Pareto 解的环境适应性, 给出了时间鲁棒性和性能鲁棒性定义, 并将其转化为两类鲁棒优化模型. 引入基于分解的多目标进化优化方法和无惩罚约束处理方法, 构建了动态多目标分解鲁棒进化优化方法. 特别是基于移动平均预测模型实现了未来动态环境下适应值的多维时间序列预测. 基于提出的两类新型性能评价测度, 针对 8 个典型动态测试函数的仿真实验, 结果表明该方法得到满足决策者精度要求, 且具有较长平均生存时间的动态鲁棒 Pareto 最优解.

**关键词** 动态多目标优化, 进化算法, 鲁棒 Pareto 最优解, 鲁棒生存时间

**引用格式** 陈美蓉, 郭一楠, 巩敦卫, 杨振. 一类新型动态多目标鲁棒进化优化方法. 自动化学报, 2017, 43(11): 2014–2032

**DOI** 10.16383/j.aas.2017.c160300

## A Novel Dynamic Multi-objective Robust Evolutionary Optimization Method

CHEN Mei-Rong<sup>1,2</sup> GUO Yi-Nan<sup>1</sup> GONG Dun-Wei<sup>1</sup> YANG Zhen<sup>1</sup>

**Abstract** Traditional methods solving dynamic multi-objective optimization problems (DMOPs) often trigger the evolution process again to find the Pareto-optimal solutions as soon as new environment appears. This may lead to larger computation and resources costs, even unable to perform the optimum solution in the limited time. Therefore, a novel evolutionary optimization method is proposed looking for dynamic robust Pareto-optimal solution sets, which are the Pareto-optimal solutions for certain environment. They can approximate to the true Pareto fronts in following consecutive dynamic environments along a certain satisfaction threshold, and directly be used as Pareto solutions of these environments so as to reduce the computation cost. Two metrics including time robustness and performance robustness are presented to measure the environmental adaptability of Pareto-optimal solutions. Subsequently, they are transformed into two kinds of robust optimization models. Multi-objective evolutionary algorithm based on decomposition and penalty-parameter less constraint handling method are introduced to form the decomposition-based dynamic multi-objective robust evolutionary optimization method. Especially, a moving average prediction model is adopted to realize multi-dimensional time series prediction of these solutions. In term of eight benchmark functions and two novel metrics, the simulation results indicate that the proposed method can obtain the robust Pareto-optimal solutions meeting the need of decision makers with more average survive time.

**Key words** Dynamic multi-objective optimization, evolutionary algorithm, robust Pareto optimal solution, robust survival time

**Citation** Chen Mei-Rong, Guo Yi-Nan, Gong Dun-Wei, Yang Zhen. A novel dynamic multi-objective robust evolutionary optimization method. *Acta Automatica Sinica*, 2017, 43(11): 2014–2032

许多工程实际优化问题, 例如动态车辆路径规划、动态生产调度和最优控制器的动态设计等, 其

待优化目标的个数以及目标函数的变量维数和参数会跟随环境发生动态变化, 从而导致最优解随之发生改变. 如果待优化目标  $\geq 2$ , 且多个动态目标之间存在冲突, 则称之为动态多目标优化问题 (Dynamic multi-objective optimization problems, DMOPs). 本文主要考虑一种目标函数参数随环境动态发生改变的动态多目标优化问题. 不失一般性, 该问题描述为

$$\begin{aligned} \min \mathbf{F}(\mathbf{x}, \alpha(t)) &= \{f_1(\mathbf{x}, \alpha(t)), \dots, f_M(\mathbf{x}, \alpha(t))\} \\ \text{s.t. } \mathbf{x} &\in S \end{aligned} \quad (1)$$

其中,  $\mathbf{x}$  是决策变量,  $S \subset \mathbf{R}^N$  是决策空间;  $\mathbf{F} : (\mathbf{x},$

收稿日期 2016-03-31 录用日期 2016-11-17  
Manuscript received March 31, 2016; accepted November 17, 2016

国家重点基础研究发展计划 (973 计划) (2014CB046300), 国家自然科学基金 (61573361), 中国矿业大学创新团队 (2015QN003) 资助

Supported by National Basic Research Program of China (973 Program) (2014CB046300), National Natural Science Foundation of China (61573361), and Innovation Team of China University of Mining and Technology (2015QN003)

1. 中国矿业大学信息与电气工程学院 徐州 221116 2. 中国矿业大学数学学院 徐州 221116

1. School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116 2. School of Mathematics, China University of Mining and Technology, Xuzhou 221116

$\alpha(t) \rightarrow \mathbf{R}^M$  包含  $M$  个目标函数  $f_i(\mathbf{x}, \alpha(t))$ ,  $\mathbf{R}^M$  是目标空间.  $\alpha(t)$  为环境相关的动态参数.

通常认为, 环境变化依时间轴呈现离散特性, 即环境变化仅发生在某些非连续的时间点, 则相应的动态参数可以被离散化为  $\alpha(k)$ ,  $k = 1, 2, \dots, K$ , 并且满足  $\forall \alpha(k), \alpha(k) \neq \alpha(k+1)$ . 在任一环境中,  $\alpha(k)$  保持不变, 相应的目标函数也不发生变化. 由此, 上述动态多目标优化问题被转化为  $K$  个静态多目标优化问题, 记为  $(\min \mathbf{F}(\mathbf{x}, \alpha(1)), \dots, \min \mathbf{F}(\mathbf{x}, \alpha(K)))$ . 针对任一环境下的静态多目标优化问题, 获得的 Pareto 最优解集记为  $\text{PS}(k)$ ,  $k = 1, 2, \dots, K$ , 相应的 Pareto 前沿记为  $\text{PF}(k)$ ,  $k = 1, 2, \dots, K$ , 则动态多目标优化问题的求解目标就是要在环境发生动态改变之前, 即  $\alpha(k+1)$  出现之前, 尽快找到满足  $\min \mathbf{F}(\mathbf{x}, \alpha(k))$  的 Pareto 最优解  $\text{PS}(k)$ .

为有效解决上述动态多目标优化问题, 设计一种快速有效的优化算法, 以适应持续的环境变化成为其关键技术. 目前, 国内外研究学者已经提出了众多解决动态多目标优化问题的方法, 其解决思路可以被划分为两大类: 跟踪环境动态变化后的最优解 (Track the moving optima, TMO)<sup>[1-11]</sup> 和动态鲁棒最优解<sup>[12-16]</sup>.

TMO 方法的本质是当检测到环境发生变化时, 重新初始化种群, 并触发相应的寻优过程. 该方法的核心在于如何在优化过程中, 平衡对决策空间的探索和开发过程, 从而兼顾种群的多样性和收敛的快速性, 保证在  $\alpha(k) \rightarrow \alpha(k+1)$  的有限时间内, 找到满足  $\min \mathbf{F}(\mathbf{x}, \alpha(k))$  的 Pareto 最优解. 现有研究主要集中在环境变化的检测、历史信息的有效利用和种群多样性保持等三方面.

针对动态环境的检测问题, Farina 等<sup>[1]</sup> 将每一代种群中随机选出的 10% 个体进行重新评价. 如果重新评价前后的个体适应值不同, 则表示环境发生变化. 为有效保持种群多样性, 研究人员往往在探测到环境变化后, 将随机或变异产生的新个体, 按一定比例替代原种群, 并在新环境下评价生成的新种群.

动态多目标优化问题中, 历史信息的有效利用主要是指, 如何利用不同环境下获得的最优解  $\text{PS}(k)$  来提高新环境下的寻优效率. 一方面, 采用记忆机制来保存不同动态环境下的历史最优解<sup>[1]</sup>. 当检测到相似动态环境时, 通过历史信息的重用来提高算法性能. 记忆增强型动态多目标进化算法<sup>[2]</sup>, 则通过对不同动态环境的串式记忆方法, 利用历史最优解来加速对新环境的跟踪能力. 记忆机制能有效解决具有周期特性的动态多目标优化问题. 另一方面, 根据连续动态环境下获得的历史信息, 采用预测机制来估计未来动态环境中的较优解, 并采用该预测解集构成或部分替代新环境下初始种群. Hatzakis 等<sup>[3]</sup> 提出的前馈法预测模型中, 采用自回归预测模型估

计相邻动态环境下的 PS, 进而生成新环境下的初始种群. Zhou 等<sup>[4]</sup> 采用时间序列预测模型, 通过加入白噪声等方法来生成新环境下的初始种群. 进而, 又提出基于流形的种群预测方法<sup>[5]</sup>, 将任一环境下的 Pareto 解集, 转化为中心点和流形两部分加以描述. 通过对这两部分分别加以预测, 得到下一时刻的初始种群. 虽然这种预测方法具有较小的时间和空间复杂度, 但是仅适用于  $\text{PS}(k)$ ,  $k = 1, 2, \dots, K$  具有相似流形的动态优化问题.

多种群优化方法的引入, 为提高动态多目标优化问题的求解效率提供了一条新途径. Goh 等<sup>[6]</sup> 采用协同进化算法, 通过处于子目标空间中各个子种群之间的竞争, 以及获胜者之间的合作进化, 实现动态环境下最优 Pareto 前沿的跟踪. 基于多种群协同的动态多目标粒子群优化算法<sup>[7]</sup> 中, 通过种群之间的竞争实现对解空间的探索, 当竞争失效后, 种群之间自适应切换到协作模式, 对解空间进行搜索. 通过种群竞争和种群协作两种模式之间的合理配合, 实现动态多目标优化问题的高效求解.

可见, TMO 方法的本质在于跟踪环境的变化, 在有限时间内尽快寻找到适应于每一环境的最优解. 虽然 TMO 为动态优化问题提供了有效的解决思路, 但是仍存在以下困难. 1) 当动态优化问题的目标函数比较复杂以及目标空间或者决策空间具有较高维数时, 会导致每一环境下的最优解很难在有限时间内获得. 2) 当环境变化较快时, 若  $\text{PS}(k)$  与  $\text{PS}(k-1)$  存在显著差别, 面向某些实际工程优化问题, 采用  $\text{PS}(k)$  虽然更逼近该环境下的最优解, 但是可能导致较高的人员操作代价和资源成本, 甚至于无法在有限时间内执行该优化解. 因此, 当每次环境变换时, 都去跟踪一个新的解往往是不可行的.

基于以上考虑, Yu 等<sup>[12]</sup> 首次提出了基于时间的动态鲁棒最优解 (Robust optimization over time, ROOT) 的概念. 其核心思想在于寻找一组鲁棒解集, 每个鲁棒解可以以一定满意度适用于多个连续动态环境. Jin 等<sup>[13]</sup> 进一步给出了求解 ROOT 问题的一个通用框架, 包含基于种群的优化算法、数据库、适应度近似策略和适应度预测策略. 其中, 解的鲁棒性由其过去和将来的适应度性能来估计. Fu 等<sup>[14-15]</sup> 为寻找随时间变化的动态鲁棒优化解, 定义了生存时间和平均适应度两个可计算性能指标来描述解的鲁棒性. 以上关于求解动态鲁棒最优解的进化方法都是针对动态单目标优化问题提出的. 面向动态多目标优化问题, Chen 等<sup>[16]</sup> 提出了描述 Pareto 解环境适应性的新型鲁棒 Pareto 最优解 (Robust Pareto optima over time, RPOOT) 的定义, 给出了鲁棒 Pareto 解的生存时间和平均适应度描述.

求解 RPOOT 和已有的 ROOT 有很大的不同.

1) 由于目标个数从单一目标扩展成了多个目标, 而且多个目标之间往往存在矛盾性. 满足单一目标最优的解往往为边界点. 兼顾多个目标性能的折中解往往不能在所有目标上具有最优性能. 因此, 某一较优解的鲁棒性不能仅考虑其在连续动态环境下某一目标的适应度值, 而要度量其所有子目标的相对变化程度. 2) 求解鲁棒解的过程中往往需要预测未来连续动态环境下的适应度. ROOT 仅仅是单一目标值构成的时间序列. 由于多目标优化问题中, 彼此非占优解都会成为最优解构成 Pareto 最优解集, 所以 RPOOT 中, 多个非占优解的目标时间序列构成了一个多时间序列预测问题. 针对上述问题, 本文基于 Chen 等<sup>[16]</sup> 提出的时间鲁棒性和性能鲁棒性评价指标, 将原有目标函数转化为兼顾多个连续动态环境下解的平均适应度函数. 通过构造新型学习预测策略, 实现动态多目标鲁棒最优解集的求解. 本文所提算法适用于解决动态多目标优化问题, 特别是环境参数动态变化频繁但不剧烈, 或者相邻动态环境下解的切换代价较大的动态优化问题. 因为对于前者, 如果用传统动态多目标优化方法求解, 通常需要在每个新环境下, 通过重新激发寻优过程, 获得适应该环境的 Pareto 最优解. 这可能导致较高的计算代价; 对于后者, 切换相邻动态环境下的最优解需要付出较大的资源成本, 甚至可能无法在有限时间内执行该优化解. 本文给出的动态多目标鲁棒进化优化方法的核心思想是寻找一组鲁棒 Pareto 解集, 每个鲁棒 Pareto 解可以以一定满意度适用于多个连续动态环境, 这样可以避免频繁地求解和执行每个动态环境下的 Pareto 最优解.

本文第 1 节讨论了动态多目标优化问题的相关工作, 并且给出了动态多目标鲁棒优化问题的定义; 在第 2 节详细描述了本文给出的算法框架, 然后将所提出的算法框架应用到现有的基准函数问题中; 第 3 节的实验结果说明算法的有效性; 第 4 节给出结论和将来的工作.

## 1 动态鲁棒 Pareto 最优解的性能度量

动态多目标鲁棒优化问题区别于式 (1) 所示动态多目标优化问题, 其核心在于寻找一组鲁棒 Pareto 最优解集  $RPS = RPS(1), \dots, RPS(LS)$ ,  $LS \leq K$ , 其中, 每个鲁棒 Pareto 最优解  $RPS(i)$  适用于多个连续动态环境. 而传统动态多目标优化问题中, 评价 Pareto 解优劣仅仅依赖于其在某一动态环境下的收敛性和分布性, 未考虑该 Pareto 解在未来连续动态环境中的适用性. 在动态多目标鲁棒优化问题中, 如何评价每个 Pareto 解的环境适应性, 以及其逼近每个动态环境下真实 Pareto 前沿的收敛性至关重要. 由此, 面向式 (1) 所示动态多目标优化问题, 给出其动态多目标鲁棒 Pareto 解的时间鲁

棒性和性能鲁棒性两类新型性能指标.

### 1.1 时间鲁棒性

定义  $L_i$  为鲁棒 Pareto 最优解集 RPS 中任意鲁棒 Pareto 解  $RPS(i)$  的生存时间. 它表示在未来  $L_i$  个连续动态环境下,  $RPS(i)$  是依一定稳定性阈值  $\eta$  的满意解, 反映了  $RPS(i)$  在连续动态环境下的时间鲁棒性. 这里

$$L_i = \min_{x_i^j \in RPS(i)} \bar{l}(x_i^j) \quad (2)$$

$$\bar{l}(x_i^j) = \max \{ l(x_i^j) | \Delta(l(x_i^j)) \leq \eta, l(x_i^j) = 0, 1, \dots, K - k \} \quad (3)$$

$$\Delta(l(x_i^j)) = \frac{\| \mathbf{F}(x_i^j, \alpha(k)) - \hat{\mathbf{F}}(x_i^j, \alpha(k + l(x_i^j))) \|}{\| \mathbf{F}(x_i^j, \alpha(k)) \|} \quad (4)$$

其中,  $\Delta(l(x_i^j))$  表示鲁棒 Pareto 解  $RPS(i)$  中某个个体  $x_i^j$  在相邻动态环境中的相对适应度差值. 差值越小, 表示解的稳定性越好; 反之, 则解的稳定性越差.  $\hat{\mathbf{F}}(x_i^j, \alpha(k + l(x_i^j)))$  表示  $x_i^j$  在未来动态环境  $\alpha(k + l(x_i^j))$  中的适应度预测值.  $\eta$  为稳定性阈值, 反映了决策者对  $x_i^j$  在相邻动态环境中的相对适应度差值  $\Delta(l(x_i^j))$  的最小容忍程度. 通常, 稳定性阈值取决于决策者的偏好.

显然, 动态鲁棒 Pareto 解的时间鲁棒性取决于其在相邻动态环境中的相对适应值和稳定性阈值. 它反映了任一鲁棒 Pareto 解  $RPS(i)$  对连续动态环境的适应能力.

### 1.2 性能鲁棒性

虽然动态多目标鲁棒优化问题的寻优目的是要找到适用于多个连续动态环境的鲁棒 Pareto 解, 但是鲁棒 Pareto 解逼近其所适用的多个连续动态环境下多目标优化问题的真实 Pareto 前沿性能也是其重要的性能指标. 假设任一鲁棒 Pareto 解  $RPS(i)$  至少适用于  $T$  个连续动态环境. 记  $T$  为固定时间窗, 反映了决策者对鲁棒 Pareto 解的环境适应能力的期望. 定义鲁棒 Pareto 解  $RPS(i)$  中的某个个体  $x_i^j$  在  $T$  时间窗内的平均适应度值

$$\mathbf{F}^{\text{ave}}(x_i^j, \alpha(k)) = (f_1^{\text{ave}}(x_i^j, \alpha(k)), \dots, f_M^{\text{ave}}(x_i^j, \alpha(k))) \quad (5)$$

其中

$$f_q^{\text{ave}}(x_i^j, \alpha(k)) = \frac{1}{T} \left( f_q(x_i^j, \alpha(k)) + \right.$$

$$\sum_{l=1}^{T-1} \hat{f}_q(x_i^j, \alpha(k+l)) \quad (6)$$

鲁棒 Pareto 解的平均适应度反映了其所在时间窗内解的性能, 即性能鲁棒性. 寻找鲁棒 Pareto 最优解  $RPS(i)$  的目标就是最小化该平均适应度值, 即  $\min_{x_i^j \in RPS(i)} \mathbf{F}^{ave}(x_i^j, \alpha(k))$ .

下面举例说明鲁棒 Pareto 解的时间鲁棒性和性能鲁棒性含义. 考虑文献 [1] 给出的动态多目标测试函数 FDA1 和 FDA3, 其中, FDA1 的 Pareto 最优解  $PS(k)$  随时间  $k$  而变化, 但相应的 Pareto 前沿  $PF(k)$  不随时间变化; FDA3 的  $PS(k)$  和  $PF(k)$  都随时间  $k$  而变化. 以  $t = 0.1, 0.2, 0.3$  三个连续动态环境为例, 相应的真实 Pareto 最优解  $PS(k)$  和

Pareto 前沿  $PF(k)$  ( $k = 1, 2, 3$ ) 如图 1 和图 2 所示. 如果固定时间窗  $T = 3$ , 将  $k = 1$  时刻的  $PS(1)$  作为鲁棒 Pareto 解  $RPS(1)$ , 即  $RPS(1) = PS(1)$ , 则其在时间窗内, 两个函数的鲁棒 Pareto 解  $RPS(1)$  的平均适应度曲线如图 3 所示.

鲁棒 Pareto 解中某个个体的生存时间取决于其在相邻动态环境中的相对适应度差值, 即式 (4) 中定义的  $\Delta(l(x_i^j))$ . 本例中所取鲁棒 Pareto 解  $RPS(1)$  与相邻的  $k = 2, 3$  环境时刻的相对适应度差值记为  $\Delta(1)$  和  $\Delta(2)$ , 如图 4 和表 1 所示. 其中,  $\mathbf{F}(x_1^j, \alpha(2)), \mathbf{F}(x_1^j, \alpha(3))$  取其真实的适应度值. 设定稳定性阈值  $\eta = 0.4$ , 对于函数 FDA1, 可得鲁棒 Pareto 解  $RPS(1)$  的生存时间为  $L_1 = 2$ ; 而对于函数 FDA3, 其鲁棒 Pareto 解  $RPS(1)$  的生存时间仅

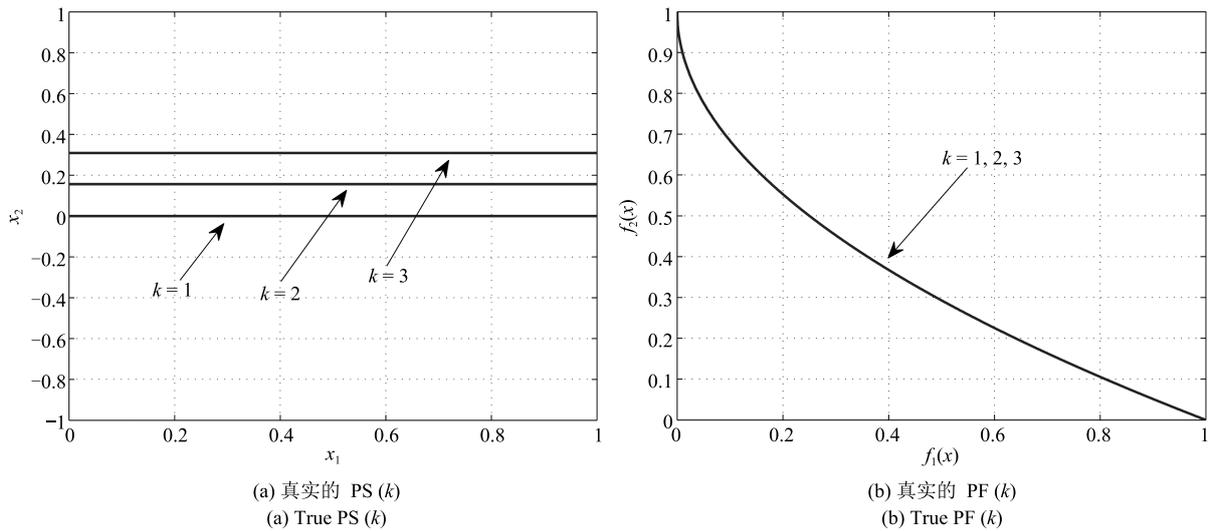


图 1 函数 FDA1 在三个时刻的真实 Pareto 解与前沿的比较  
Fig. 1 True Pareto solutions and Pareto fronts of FDA1 in three environments

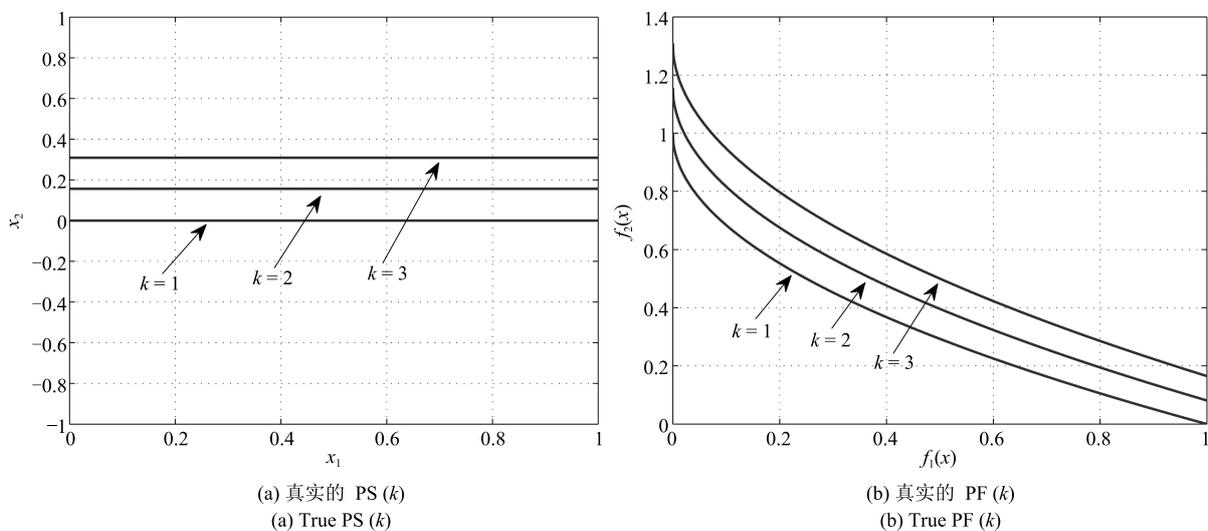


图 2 函数 FDA3 在三个时刻的真实 Pareto 解与前沿的比较  
Fig. 2 True Pareto solutions and Pareto fronts of FDA3 in three environments

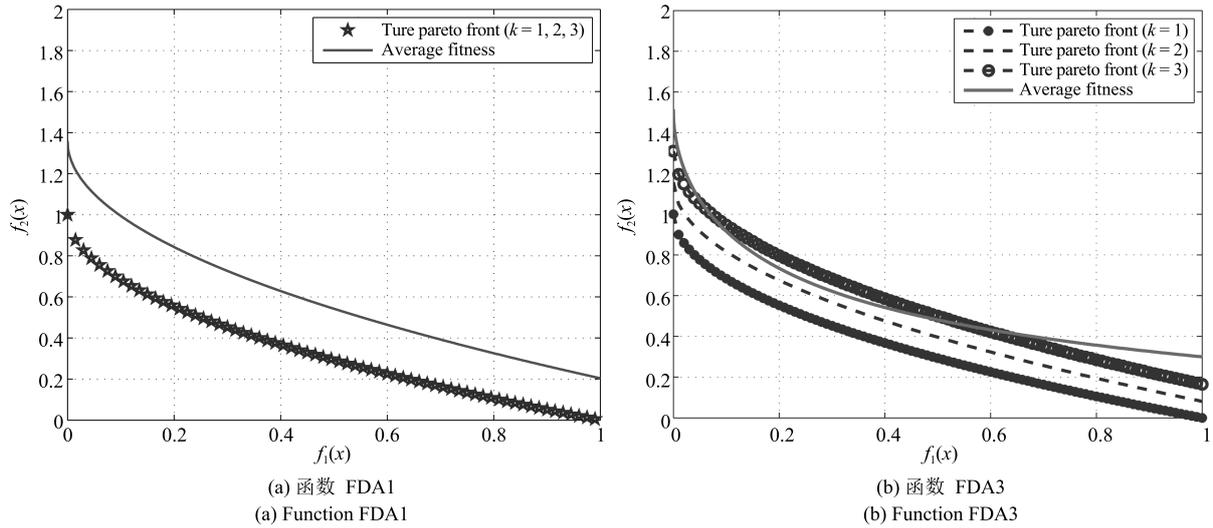


图 3 函数的平均适应度值与时刻  $k = 1, 2, 3$  的真实 Pareto 前沿的比较

Fig. 3 Average fitness and true PF( $k$ ) with  $k = 1, 2, 3$

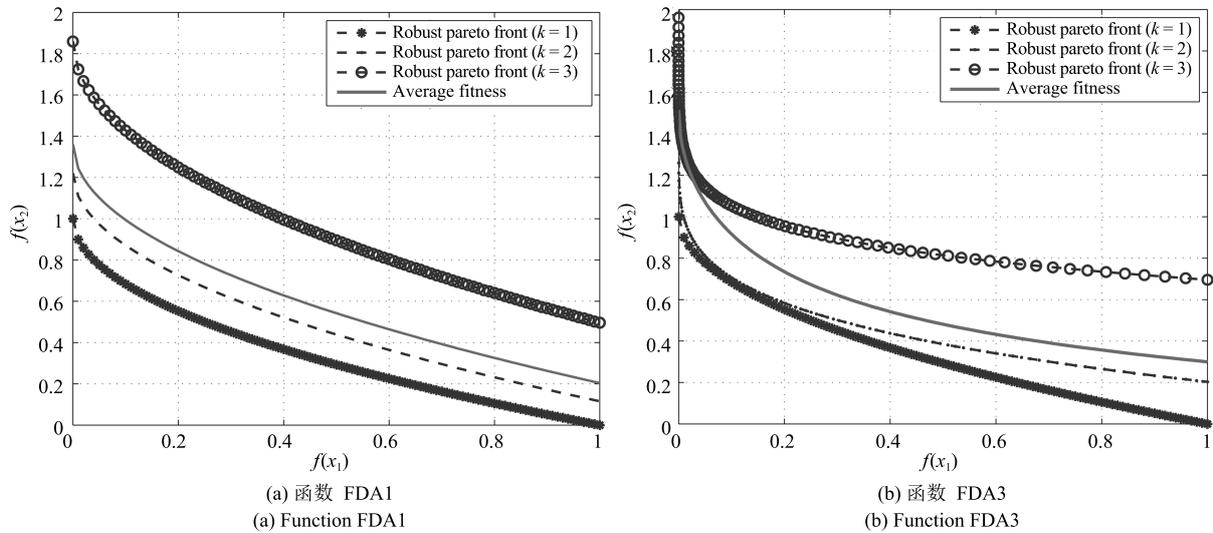


图 4 函数的平均适应度值与鲁棒解 RPS 在  $t = 1, 2, 3$  时刻的鲁棒 Pareto 前沿的比较

Fig. 4 Average fitness and robust Pareto fronts with  $k = 1, 2, 3$

表 1 RPS(1) 与相邻  $k = 2, 3$  环境下解的相对适应度差值  
Table 1 The relative fitness errors of RPS(1) and adjacent two environments

相对适应度差	$\Delta(1)$	$\Delta(2)$
FDA1	0.301	1.2177
FDA3	0.6728	1.7687

为  $L_1 = 1$ . 如果增大稳定性阈值  $\eta = 0.7$ , 则对于函数 FDA1 和 FDA3, 可得鲁棒 Pareto 解 RPS(1) 的生存时间为都为  $L_1 = 2$ . 显然, 稳定性阈值  $\eta$  体现了鲁棒 Pareto 解对相邻连续动态环境的适用程度.  $\eta$  越大, 鲁棒 Pareto 解的生存时间  $L$  越大; 反之,  $\eta$

越小, 解的生存时间  $L$  越小.

综上所述, 生存时间和平均适应度分别反映了任一鲁棒 Pareto 解在连续动态环境下的时间鲁棒性和性能鲁棒性.

## 2 动态多目标鲁棒进化优化算法

### 2.1 模型转化

在动态多目标鲁棒优化中, RPOOT 并不是寻找适用于每个动态环境的 Pareto 最优解 PS( $k$ ), 而是获得满足第 1 节定义鲁棒性的鲁棒 Pareto 最优解 RPS( $k$ ). 它不仅要近似收敛于当前动态环境下的真实 Pareto 最优解, 而且要尽可能多地逼近后续连

续动态环境下的真实 Pareto 最优解. 这就要在进化优化过程中了解 Pareto 解  $PS(k)$  的当前适应度, 及其在未来动态环境下的适应度值. 基于第 1 节给出的时间鲁棒性和性能鲁棒性定义, 构建如下两种求解动态鲁棒 Pareto 最优解的模型.

**模型 1.** 基于固定时间窗的模型转化

$$\min \mathbf{F}^{\text{ave}}(\mathbf{x}, \alpha(k)) = (f_1^{\text{ave}}(\mathbf{x}, \alpha(k)), \dots, f_M^{\text{ave}}(\mathbf{x}, \alpha(k))) \quad (7)$$

其中,  $\mathbf{F}^{\text{ave}}(\mathbf{x}, \alpha(k))$  为动态环境时刻  $k$  条件下, 解  $x$  在固定时间窗  $T$  内的平均适应度值. 该模型通过最小化固定时间窗内, 连续动态环境的平均适应度值, 来保证解  $x$  对相应连续动态环境下真实 Pareto 前沿的逼近能力.

**模型 2.** 基于约束的模型转化

$$\begin{aligned} \min \mathbf{F}^{\text{ave}}(\mathbf{x}, \alpha(k)) &= (f_1^{\text{ave}}(\mathbf{x}, \alpha(k)), \dots, f_M^{\text{ave}}(\mathbf{x}, \alpha(k))) \\ \text{s.t.} \quad &\frac{\|\mathbf{F}^{\text{ave}}(\mathbf{x}, \alpha(k)) - \hat{\mathbf{F}}(\mathbf{x}, \alpha(k+q))\|}{\|\mathbf{F}^{\text{ave}}(\mathbf{x}, \alpha(k))\|} \leq \eta \\ &q = 0, \dots, T-1 \end{aligned} \quad (8)$$

该模型将  $x$  对未来连续动态环境的适用程度转化成时间窗内的  $T$  个鲁棒约束. 其中, 解  $x$  的当前适应度函数与将来相邻的  $T-1$  个环境时刻的适应度值的标准差小于决策者的稳定性阈值  $\eta$ . 其中, 当  $q=0$  时,  $\hat{\mathbf{F}}(\mathbf{x}, \alpha(k+q))$  表示当前时刻  $k$  的真实适应度值, 即  $\hat{\mathbf{F}}(\mathbf{x}, \alpha(k+q)) = \mathbf{F}(\mathbf{x}, \alpha(k))$ .

## 2.2 基于分解的 RPOOT 进化优化算法

上述两类转化模型中, 模型 1 将动态多目标鲁棒优化问题中的两类鲁棒性度量转化为固定时间鲁棒性下的解性能鲁棒性, 即某个解在当前环境及其未来  $T-1$  个连续动态环境下的平均适应度值. 因此, 针对该类转化模型, 优化求解过程的核心是, 寻求在时间窗内同时对当前和未来动态环境具有良好收敛性的鲁棒 Pareto 最优解. 不同于传统的 DMOPs, 一方面, 优化目标是各个解在时间窗  $T$  内的平均适应值, 另一方面, 未来连续动态环境下解的适应度值是未知的, 需要采取时间序列预测方法来估计. 预测方法见第 2.3 节. 针对模型 1 的求解, 采用传统的基于分解的多目标进化算法 (Multi-objective evolutionary algorithm based on decomposition, MOEA/D). 其中, 分解方法采用基于罚值的边界交集法 (Penalty-based boundary intersection, PBI)<sup>[17]</sup>. 具体伪代码见算法 1.

不同于模型 1, 模型 2 将解的时间鲁棒性条件作为约束, 限制其平均适应度范围, 从而将原有优化问题转化为一个动态多目标约束优化问题. 针对该约

束条件, 本文基于传统 MOEA/D, 采用一种基于分解的无参数惩罚约束处理方法<sup>[18]</sup>.

### 算法 1. 基于分解的动态多目标鲁棒进化优化算法

- 1: 开始;
- 2: 初始化环境时刻  $k=1$ , 获得环境参数  $\alpha(k)$ ;
- 3: 随机初始化种群  $P$ , 估计种群中每个解  $\mathbf{x}^i$  的当前平均适应度值  $\mathbf{F}^{\text{ave}}(\mathbf{x}^i, \alpha(k))$ ,  $i=1, 2, \dots, N_{\text{pop}}$ ;
- 4: 初始化参考点  $\mathbf{z}(k) = (z_1(k), \dots, z_M(k))^T$ , 其中,  $z_j(k) = \min\{f_j^{\text{ave}}(\mathbf{x}^1, \alpha(k)), \dots, f_j^{\text{ave}}(\mathbf{x}^{N_{\text{pop}}}, \alpha(k))\}$ ;
- 5: 根据均匀分布产生权重向量  $\lambda^i = (\lambda_1^i, \dots, \lambda_M^i)$ ,  $i=1, \dots, N_{\text{pop}}$ , 利用动态 PBI 分解法将目标分解为  $N_{\text{pop}}$  个动态标量优化子问题, 其动态标量目标函数记为  $f^{\text{PBI}}(\mathbf{x}(k)|\lambda, z)$ ;
- 6: 计算任意两个权重向量的欧氏距离, 为每个权重向量选出最近的  $r$  个向量作为它的邻域, 记为  $B(\lambda^i) = (\lambda^{i1}, \dots, \lambda^{ir})$ ,  $i=1, \dots, N_{\text{pop}}$ ;
- 7: 设进化代数  $t=1$ , 总进化代数 Gen;
- 8: **while**  $t < \text{Gen}$  **do**
- 9:     **for**  $i=1$  to  $N_{\text{pop}}$  **do**
- 10:         根据差分进化和高斯变异产生子代  $P_Z(t)$ ;
- 11:         对每一个  $j=1, \dots, M$  更新参考点  $\mathbf{z}(k)$ ;
- 12:         更新邻域解;
- 13:     **end for**
- 14: **end while**
- 15: 输出  $(\mathbf{x}^1(k), \dots, \mathbf{x}^{N_{\text{pop}}}(k))$  和  $\{\mathbf{F}^{\text{ave}}(\mathbf{x}^1(k), \alpha(k)), \dots, \mathbf{F}^{\text{ave}}(\mathbf{x}^{N_{\text{pop}}}(k), \alpha(k))\}$ , 作为第  $k$  个动态环境下的鲁棒 Pareto 最优解;
- 16: 如果  $k > K$ , 转步骤 17, 否则  $k = k+1$ , 转步骤 3;
- 17: 结束.

在进化过程中, 约束处理的核心在于约束满意度的确定. 由于在式 (8) 中, 受到预测误差及环境动态变化的影响, 随着  $T$  的增加, 同时满足  $T$  个约束条件的可行解将大大减少. 为此, 将上述约束条件转化为约束满足程度, 采用转化后的弱化约束条件来对个体加以比较.

$$g_q(\mathbf{x}, \alpha(k)) = \frac{\|\mathbf{F}^{\text{ave}}(\mathbf{x}, \alpha(k)) - \hat{\mathbf{F}}(\mathbf{x}, \alpha(k+q))\|}{\|\mathbf{F}^{\text{ave}}(\mathbf{x}, \alpha(k))\|}, \quad -\eta \leq 0, \quad q = 0, \dots, T-1 \quad (9)$$

基于上述转化后的约束条件, 定义约束满意度函数为

$$V_s(\mathbf{x}, \alpha(k)) = \left\lfloor \sum_{q=0}^s \max\{g_q(\mathbf{x}, \alpha(k)), 0\} \right\rfloor, \quad s = 0, \dots, T-1$$

如果  $V_s(\mathbf{x}, \alpha(k)) = 0$ , 说明个体  $x$  是满足式 (8) 中前  $s$  个约束条件的弱可行解; 否则  $x$  不是弱可行解.

模型 2 的主体算法框架采用算法 1 所示的 MOEA/D. 不同在于: 步骤 3 的初始化和步骤

10 和更新种群操作,增加了个体的约束满意度  $V_s(\mathbf{x}, \alpha(k))$  计算,用来判断个体为时间窗内的一个弱可行域解.步骤 12 的更新邻域解操作中,采用算法 2 所示的弱约束处理方法,判断邻域解是否需要更新.

### 算法 2. 无参数惩罚约束方法

```

1: 取  $\forall \mathbf{x}^i, \mathbf{x}^j \in P(t)$ ;
2: for  $s = T - 1$  to 0 then
3:   if  $V_s(\mathbf{x}^i, \alpha(k)) = 0$ , 且  $V_s(\mathbf{x}^j, \alpha(k)) > 0$  then
4:      $\mathbf{x}^i$  替代  $\mathbf{x}^j$ , 转步骤 15;
5:   elseif  $V_s(\mathbf{x}^i, \alpha(k)) = V_s(\mathbf{x}^j, \alpha(k)) = 0$ ,
   且  $f^{\text{PBI}}(\mathbf{x}^i, k|\lambda, \mathbf{z}) < f^{\text{PBI}}(\mathbf{x}^j, k|\lambda, \mathbf{z})$  then
6:      $\mathbf{x}^i$  替代  $\mathbf{x}^j$ , 转步骤 15;
7:   elseif  $V_s(\mathbf{x}^i, \alpha(k)) > 0, V_s(\mathbf{x}^j, \alpha(k)) > 0$ ,
   且  $s > 0$  then
8:      $s = s - 1$ ;
9:   elseif  $s = 0$  then
10:    令  $f^{\text{PBI}}(\mathbf{x}, k|\lambda, \mathbf{z}) = f^{\text{PBI}}(\mathbf{x}, k|\lambda, \mathbf{z}) +$ 
     $V_{T-1}(\mathbf{x}, \alpha(k))$ ;
11:    if  $f^{\text{PBI}}(\mathbf{x}^i, k|\lambda, \mathbf{z}) < f^{\text{PBI}}(\mathbf{x}^j, k|\lambda, \mathbf{z})$  then
12:       $\mathbf{x}^i$  替代  $\mathbf{x}^j$ , 转步骤 15;
13:    end if
14:  end if
15: end for

```

### 2.3 预测机制

在动态多目标优化问题的求解中,预测机制可以有效改善算法进化效率.在传统的 TMO 求解方法中,预测的目的是在下一相邻动态环境下,能够充分利用历史和当前最优 Pareto 解和其他已知信息,来估计其动态环境参数的变化趋势,构建其新环境下的初始种群.

与 TMO 方法中的预测目的不同,在 RPOOT 中,由于要考虑鲁棒 Pareto 解的鲁棒性,确定其对未来动态环境适用情况.所以,需要使用未来动态环境下该鲁棒 Pareto 解的适应度值.对于在线的实际动态优化问题,未来动态环境下解的适应度值是未知的,需要通过预测的方法来估计.因此, RPOOT 中预测的本质是根据历史信息 and 动态环境(参数)变化趋势,估计当前解在未来连续环境下的适应度值.在模型 1 和模型 2 中,优化目标均为平均适应度函数值  $\mathbf{F}^{\text{ave}}(\mathbf{x}, \alpha(k))$ .根据式(6)可知,  $x$  在时间窗内连续动态环境下的平均适应度依赖于未来连续动态环境下  $x$  的适应度值.因此,在动态多目标鲁棒优化过程中,需要预测每代种群中所有解在多个目标上的未来适应度值,从而使 RPOOT 的预测问题转化为一个多维时间序列预测问题,其维数为  $N \times M$ .为降低计算复杂度,预测方法的设计需考虑如下两方面,1) 预测方法的空间复杂度不能太高;2) 预测方法的时间复杂度不能太大,否则会大大影响动态多目标优化算法的实时性.

本文采用移动平均(Moving average, MA)<sup>[19]</sup>

预测模型. MA 是一种简单有效的时序预测模型.首先,假设在第  $k$  个动态环境下,针对种群中的每个解  $x$  构造一个长度为  $m$  的时间序列  $(\mathbf{F}(\mathbf{x}, \alpha(k-m+1)), \dots, \mathbf{F}(\mathbf{x}, \alpha(k)))$ .其中,  $\mathbf{F}(\mathbf{x}, \alpha(i))$ ,  $k-m+1 \leq i \leq k$  是由  $f_j(\mathbf{x}, \alpha(k))$ ,  $1 \leq j \leq M$  组成的该环境下解  $x$  的适应度向量.预测的目的就是根据上述时间序列来估计未来动态环境下的  $\mathbf{F}(\mathbf{x}, \alpha(k+i))$ ,  $1 \leq i \leq T-1$ . MA 模型定义如下:

$$\hat{\mathbf{F}}(\mathbf{x}, \alpha(k+i)) = b + \varepsilon(k) \quad (10)$$

其中,  $\varepsilon(k) \sim U(0, \sigma^2)$ , 即方差为  $\sigma^2$  的高斯白噪声.  $b$  是对时间序列中相邻的前  $m$  个数据点的平均估计.

$$b = \frac{1}{m} \sum_{j=1}^m \mathbf{F}(\mathbf{x}, \alpha(k+i-j)) \quad (11)$$

方差  $\sigma^2$  通过下式估计

$$\hat{\sigma}^2 = \frac{1}{m} \sum_{j=1}^m (\mathbf{F}(\mathbf{x}, \alpha(k+i-j)) - b)^2 \quad (12)$$

可见,  $\hat{\mathbf{F}}(\mathbf{x}, \alpha(k+i))$  在未来连续动态环境下的适应度值可被预测为均值为  $b$ , 方差为  $\hat{\sigma}^2$  的值.

### 3 实验分析与比较

实验仿真的主要任务是测试本文所提出算法的有效性.首先,为有效度量和评价算法的鲁棒性、收敛性和分布性,提出了两个性能指标.其次,针对 8 个常用动态多目标优化测试函数,设计了以下 3 组实验:1) 为分析说明本文所提模型对 Pareto 解集的性能影响,通过实验对比分析了不同时间窗和不同稳定性阈值下的解性能;2) 为进一步分析本文算法对不同环境变化快慢及剧烈程度的适应能力,基于提出的两类模型,通过实验对比分析了不同动态环境频率和强度参数下鲁棒 Pareto 最优解集的性能;3) 为验证本文所提鲁棒模型转化的合理性,将基于本文模型所得动态多目标鲁棒 Pareto 解集与文献[16]所得解集进行对比分析.

#### 3.1 测试函数

本文采用 8 个常用动态多目标测试函数组,包括由 Farina 和 Deb<sup>[1]</sup> 构建的 Fun 1 ~ Fun 5 (FDA1-5), 以及由 Goh 与 Tan<sup>[6]</sup> 构建的 Fun 6 ~ Fun 8 (DMOP1-3).具体函数定义如表 2 所示.

本文核心研究上述动态多目标优化问题所获得的 Pareto 鲁棒解对连续动态环境的适应能力.因此,首先要确定其动态环境变化时刻  $k^{[1, 6]}$ .

$$k = \frac{1}{n_d} \left\lceil \frac{\tau}{\tau_d} \right\rceil \quad (13)$$

表 2 测试函数  
Table 2 Benchmark functions

函数	测试函数	变量维数	变量范围	说明
Fun 1	$f_1(X_I) = x_1, f_2 = g \times h$ $g(X_{II}) = 1 + \sum_{x_i \in X_{II}} (x_i - G(k))^2$ $h(f_1, g) = 1 - \sqrt{f_1/g}, G(k) = \sin(0.5\pi k)$	$N = 10$ $ X_I  = 1$ $ X_{II}  = 9$	$X_I = (x_1) \in [0, 1]$ $X_{II} = (x_2, \dots, x_N) \in [-1, 1]$	FDA1 in [6] PS 随环境变化 PF 不随环境变化
Fun 2	$f_1(X_I) = x_1, f_2 = g \times h$ $g(X_{II}) = 1 + \sum_{x_i \in X_{II}} x_i^2$ $h(f_1, g) = 1 - (f_1/g)^{H(k)-1}$ $H(k) = 0.75 + 0.7 \sin(0.5\pi k)$	$N = 10$ $ X_I  = 1$ $ X_{II}  = 9$	$X_1 = (x_1) \in [0, 1],$ $X_2 \in [-1, 1]$	FDA2 in [6] PS 不随环境变化 PF 随环境变化
Fun 3	$f_1(X_I) = \sum_{x_i \in X_I} x_i^{F(k)}, f_2 = g \times h$ $g(X_{II}) = 1 + G(k) + \sum_{x_i \in X_{II}} (x_i - G(k))^2$ $h(f_1, g) = 1 - \sqrt{f_1/g}, G(k) =  \sin(0.5\pi k) $ $F(k) = 10^{2 \sin(0.5\pi k)}$	$N = 10$ $ X_I  = 1$ $ X_{II}  = 9$	$X_I = (x_1) \in [0, 1]$ $X_{II} = (x_2, \dots, x_N) \in [-1, 1]$	FDA3 in [6] PS 随环境变化 PF 随环境变化
Fun 4	$f_1(x) = (1 + g) \cos(0.5\pi x_1) \cos(0.5\pi x_2)$ $f_2(x) = (1 + g) \cos(0.5\pi x_1) \sin(0.5\pi x_2)$ $f_3(x) = (1 + g) \sin(0.5\pi x_1)$ $g(x) = \sum_{i=3}^n (x_i - G(k))^2$ $G(k) =  \sin(0.5\pi k) $	$N = 12$	$x_i \in [0, 1]$ $\forall i = 1, 2, \dots, N$	FDA4 in [6] PS 随环境变化 PF 不随环境变化
Fun 5	$f_1(x) = (1 + g) \cos(0.5\pi y_1) \cos(0.5\pi y_2)$ $f_2(x) = (1 + g) \cos(0.5\pi y_1) \sin(0.5\pi y_2)$ $f_3(x) = (1 + g) \sin(0.5\pi y_1)$ $g(x) = G(k) + \sum_{i=3}^n (x_i - G(k))^2$ $y_i = x_i^{F(k)}$ $G(k) =  \sin(0.5\pi k) $ $F(k) = 1 + 100 \sin^4(0.5\pi k)$	$N = 12$	$x_i \in [0, 1]$ $\forall i = 1, 2, \dots, N$	FDA5 in [6] PS 随环境变化 PF 随环境变化
Fun 6	$f_1(x_1) = x_1, f_2 = g \times h$ $g(x_2, \dots, x_n, k) = 1 + 9 \sum_{i=2}^n x_i^2$ $h(f_1, g) = 1 - (f_1/g)^{H(k)}$ $H(k) = 0.75 \sin(0.5\pi k) + 1.25$	$N = 10$	$x_i \in [0, 1]$ $\forall i = 1, 2, \dots, N$	DMOP1 in [8] PS 不随环境变化 PF 随环境变化
Fun 7	$f_1(x_1) = x_1, f_2 = g \times h$ $g(x_2, \dots, x_n, k) = 1 + \sum_{i=2}^n (x_i - G(k))^2$ $h(f_1, g) = 1 - (f_1/g)^{H(k)}$ $H(k) = 0.75 \sin(0.5\pi k) + 1.25$	$N = 10$	$x_i \in [0, 1]$ $\forall i = 1, 2, \dots, N$	DMOP2 in [8] PS 随环境变化 PF 随环境变化
Fun 8	$f_1(x_1) = x_1, f_2 = g \times h$ $g(x_2, \dots, x_n, k) = 1 + 9 \sum_{i=2}^n (x_i - G(k))^2$ $h(f_1, g) = 1 - \sqrt{f_1/g}$ $H(t) = \sin(0.5\pi k)$	$N = 10$	$x_i \in [0, 1]$ $\forall i = 1, 2, \dots, N$	DMOP3 in [8] PS 随环境变化 PF 不随环境变化

其中,  $\tau$  是进化代数.  $\tau_d$  表示环境保持不变时 (即  $k$  不变), 所对应的进化代数规模, 用来控制环境的变化频率. 环境的变化强度用  $n_d$  来表示.

### 3.2 性能评价指标

RPOOT 本质上是在进化过程中不断寻找对未来连续动态环境具有鲁棒性的 Pareto 解集. 也就是在每个动态环境下, 搜索当前环境下的较优 Pareto 解, 而且该解能最大程度适用于未来多个动态环境. 因此, 鲁棒 Pareto 解集不仅要考虑解的鲁棒性, 而且要考虑解在其适用环境中的收敛性. 首先, 在时间尺度上定义平均生存时间评价解的鲁棒性能; 其

次, 采用平均鲁棒逆向世代距离 (Robust inverted generational distance, RIGD) 度量鲁棒解的收敛性和分布性.

#### 3.2.1 平均生存时间

动态多目标优化问题中, 鲁棒 Pareto 最优解的鲁棒性能往往体现为其可以适用的连续动态环境数目, 即 Pareto 解的生存时间. 在此基础上, 为度量所有 Pareto 解集中解的鲁棒性, 定义其平均生存时间作为 Pareto 解集的时间鲁棒性测度.

$$\bar{L} = \frac{1}{K} \sum_{k=1}^K L_k \quad (14)$$

其中,  $K$  是动态环境出现次数,  $L_k$  是鲁棒 Pareto 最优解  $RPS(k)$  的生存时间. 显然, 平均生存时间越长, 鲁棒 Pareto 解集的鲁棒性越好.

### 3.2.2 平均鲁棒逆向世代距离

鲁棒 Pareto 解集要兼顾其时间尺度和性能尺度下的鲁棒性能. 后者主要体现为鲁棒 Pareto 解集对连续动态环境下的真实 PF( $k$ ) 的逼近程度, 即其收敛性和分布性. 传统多目标优化问题中, 逆向世代距离 (Inverted generational distance, IGD) 通常用来度量 Pareto 最优解的收敛性和分布性. 因此, 将其推广到动态多目标鲁棒优化问题中, 定义新型平均鲁棒逆向世代距离来度量鲁棒 Pareto 解集的收敛性和分布性.

$$\text{RIGD}^{\text{RPOOT}} = \frac{1}{K} \sum_{k=1}^K \text{MIGD}(k) \quad (15)$$

$$\text{MIGD}(k) =$$

$$\max_{j=k, \dots, k+L_k} \frac{1}{|\text{PF}(j)|} \sum_{x^v \in \text{PF}(j)} d(x^v, \text{RPF}(j)) \quad (16)$$

$$d(x^v, \text{RPF}(j)) =$$

$$\min_{x^u \in \text{RPF}(j)} \sqrt{\sum_{j=1}^M (f_j(x^v) - f_j(x^u))^2} \quad (17)$$

其中,  $\text{PF}(j)$  和  $|\text{PF}(j)|$  分别代表第  $j$  个动态环境下的真实 Pareto 解及其规模, 本文将  $\text{PF}(j)$  分别离散化为 100 个点 (二目标) 和 300 个点 (三目标),  $\text{MIGD}(k)$  定义了第  $k$  个动态环境下鲁棒 Pareto 解  $\text{RPF}(k)$  与第  $j$  ( $j = k, k+1, \dots, k+L_k$ ) 个动态环境下真实 Pareto 解  $\text{PF}(j)$  的 IGD 的最大值.  $\text{RIGD}^{\text{RPOOT}}$  反映  $\text{RPF}(k)$  对其适用环境下的各个时刻 MIGD 测度的平均值.

### 3.3 实验参数设置

设计的全部仿真实验结果均采用 MATLAB R2011b 实现. 针对所有测试函数, 解决两种模型的 MOEA/D 算法均采用实数编码、差分进化交叉和高斯变异. 具体参数设置如下:

1) 群体规模  $N_{\text{pop}}$  与权向量的设置: 传统 MOEA/D 算法中, 权向量的数目等于种群规模, 选取为  $N_{\text{pop}} = C_{H+M-1}^{M-1}$ . 其中,  $M$  为目标函数个数,  $H$  取决于决策者. 在实验中, 目标个数分别为 2 和 3 时, 选取  $H$  为 99 和 23, 确定种群规模  $N_{\text{pop}}$  为 100 和 300. 另外, 选取权向量的邻域个数为 20.

2) 进化算子中的控制参数: 差分进化操作中, 交叉率  $\text{CR} = 1.0$ , 尺度因子  $F = 0.5$ ; 高斯变异概率  $P_m = 1/N$ ,  $N$  为决策变量的维数; PBI 分解方法中的惩罚参数  $\theta = 5$ .

3) 算法运行次数与停止准则: 针对任一测试问题, 各个算法独立运行 15 次; 最大进化代数  $\tau = 3000$ ; 动态环境变化频率  $\tau_d = 30$ ; 动态环境变化强度  $n_d = 10$ ; 动态环境变化次数  $K = 100$ .

4) 动态鲁棒优化模型中, 目标函数值的预测采用 MA 方法实现, 其构造的时间序列长度  $m = 10$ .

### 3.4 实验结果与分析

考虑到目前仅有文献 [16] 与本文所提动态多目标鲁棒优化方法相关, 因此, 仅对上述二者进行性能对比分析. 根据文献 [16] 的鲁棒 Pareto 解集定义, 采用 MOEA/D 方法求解动态多目标优化问题, 获得其在 100 个动态环境下的鲁棒 Pareto 解集, 并生成一组真正的鲁棒 Pareto 解集. 在如下实验分析中, 方法 1 代表文献 [16] 所提鲁棒优化方法; 方法 2 为本文所提鲁棒动态优化模型 1; 方法 3 为本文所给出的鲁棒动态优化模型 2.

#### 3.4.1 鲁棒 Pareto 解集对参数的敏感性分析

表 3 给出了不同稳定性阈值下, 基于两种模型的动态多目标鲁棒优化方法所获得 Pareto 解集的平均生存时间. 当时间窗  $T = 2$ , 稳定性阈值  $\eta$  分别为 0.2, 0.4 和 0.6 时, 所获得鲁棒 Pareto 解集的平均生存时间, 表明当稳定性阈值越大时, 所得 Pareto 解集的平均生存时间越大. 这是因为阈值反映决策者对平均适应度偏离程度的容忍程度.  $\eta$  越大, 决策者可以接受未来连续环境下适应值的波动程度越大, Pareto 解的时间鲁棒性越好, 但每个动态环境下的解性能变弱.

表 4 是在不同时间窗  $T = 2, 4, 6$  下, 本文所提

表 3 不同稳定性阈值下两种模型的平均生存时间比较 ( $T = 2, n_d = 10, \tau_d = 30$ )

Table 3 Average survival time of methods 2 and 3 in different stability thresholds ( $T = 2, n_d = 10, \tau_d = 30$ )

方法	参数	函数 1	函数 2	函数 3	函数 4
方法 2	$\eta = 0.2$	2.4480 ± 0.0314	7.2233 ± 0.0082	2.3433 ± 0.0342	2.6867 ± 0.0049
	$\eta = 0.4$	3.4640 ± 0.0099	21.658 ± 0.0086	3.9428 ± 0.0208	4.2027 ± 0.0096
	$\eta = 0.6$	<b>4.5720 ± 0.0077</b>	<b>49.51 ± 0</b>	<b>5.9933 ± 0.0049</b>	<b>6.1293 ± 0.0080</b>
方法 3	$\eta = 0.2$	2.5067 ± 0.0315	7.4120 ± 0.2402	2.5500 ± 0.0746	2.7800 ± 9.1E-16
	$\eta = 0.4$	3.5133 ± 0.0082	20.941 ± 0.0083	4.3147 ± 0.0280	4.3187 ± 0.0106
	$\eta = 0.6$	<b>4.5940 ± 0.0019</b>	<b>49.510 ± 0</b>	<b>6.1227 ± 0.0046</b>	<b>6.2727 ± 0.0110</b>

表 3 不同稳定性阈值下两种模型的平均生存时间比较 ( $T = 2, n_d = 10, \tau_d = 30$ ) (续)Table 3 Average survival time of methods 2 and 3 in different stability thresholds ( $T = 2, n_d = 10, \tau_d = 30$ ) (Continued)

方法	参数	函数 5	函数 6	函数 7	函数 8
方法 2	$\eta = 0.2$	1.6120 $\pm$ 0.0041	9.9680 $\pm$ 0.0041	2.2813 $\pm$ 0.0106	1.2407 $\pm$ 0.0110
	$\eta = 0.4$	2.4093 $\pm$ 0.0026	30.197 $\pm$ 0.5440	3.6000 $\pm$ 9.1E-16	1.7000 $\pm$ 7.3E-15
	$\eta = 0.6$	<b>4.3727 <math>\pm</math> 0.0059</b>	<b>49.510 <math>\pm</math> 0</b>	<b>4.8247 <math>\pm</math> 0.0210</b>	<b>2.2867 <math>\pm</math> 0.0105</b>
方法 3	$\eta = 0.2$	1.7200 $\pm$ 2.2E-16	9.9453 $\pm$ 0.0074	2.3347 $\pm$ 0.0106	1.2600 $\pm$ 2.2E-16
	$\eta = 0.4$	2.5500 $\pm$ 0.0318	30.007 $\pm$ 0.4951	3.6520 $\pm$ 0.0068	1.7800 $\pm$ 4.6E-16
	$\eta = 0.6$	<b>4.4787 <math>\pm</math> 0.0035</b>	<b>49.510 <math>\pm</math> 0</b>	<b>4.8607 <math>\pm</math> 0.0026</b>	<b>2.4033 <math>\pm</math> 0.0082</b>

表 4 不同时间窗下两种模型的平均生存时间比较 ( $\eta = 0.4, n_d = 10, \tau_d = 30$ )Table 4 Average survival time of methods 2 and 3 in different time windows ( $\eta = 0.4, n_d = 10, \tau_d = 30$ )

方法	参数	函数 1	函数 2	函数 3	函数 4
方法 2	$T = 2$	<b>3.4640 <math>\pm</math> 0.0099</b>	<b>21.658 <math>\pm</math> 0.0086</b>	3.9428 $\pm$ 0.0208	<b>4.2027 <math>\pm</math> 0.0096</b>
	$T = 4$	3.1460 $\pm$ 0.0051	21.606 $\pm$ 0.1788	3.6447 $\pm$ 0.0168	3.9440 $\pm$ 0.0203
	$T = 6$	3.1180 $\pm$ 0.0115	21.650 $\pm$ 3.6E-15	<b>4.0373 <math>\pm</math> 0.0059</b>	3.9007 $\pm$ 0.0167
方法 3	$T = 2$	<b>3.5133 <math>\pm</math> 0.0082</b>	<b>20.941 <math>\pm</math> 0.0083</b>	<b>4.3147 <math>\pm</math> 0.0280</b>	<b>4.3187 <math>\pm</math> 0.0106</b>
	$T = 4$	3.4033 $\pm$ 0.0082	20.900 $\pm$ 0	3.9533 $\pm$ 0.0082	4.1893 $\pm$ 0.0110
	$T = 6$	3.4120 $\pm$ 0.0108	20.644 $\pm$ 0.3445	4.1467 $\pm$ 0.0374	4.2313 $\pm$ 0.0125
方法	参数	函数 5	函数 6	函数 7	函数 8
方法 2	$T = 2$	2.4093 $\pm$ 0.0026	<b>30.197 <math>\pm</math> 0.5440</b>	<b>3.6000 <math>\pm</math> 9.1E-16</b>	<b>1.7000 <math>\pm</math> 7.3E-15</b>
	$T = 4$	<b>2.4500 <math>\pm</math> 0.0038</b>	29.448 $\pm$ 0.7324	3.1707 $\pm$ 0.0406	1.5220 $\pm$ 0.0056
	$T = 6$	<b>2.4500 <math>\pm</math> 0</b>	29.085 $\pm$ 0.6836	3.2027 $\pm$ 0.0448	1.5220 $\pm$ 0.0056
方法 3	$T = 2$	2.5500 $\pm$ 0.0318	<b>30.007 <math>\pm</math> 0.4951</b>	<b>3.6520 <math>\pm</math> 0.0068</b>	<b>1.7800 <math>\pm</math> 4.6E-16</b>
	$T = 4$	2.7100 $\pm$ 4.5E-16	28.068 $\pm$ 0.2995	3.3927 $\pm$ 0.0546	1.5600 $\pm$ 2.2E-16
	$T = 6$	<b>2.8813 <math>\pm</math> 0.0376</b>	28.340 $\pm$ 0.4003	3.4820 $\pm$ 0.0441	1.5940 $\pm$ 0.0356

两种模型所获得鲁棒 Pareto 最优解集的平均生存时间随着时间窗的增大, 解集的平均生存时间呈现出一定的减小趋势. 其原因在于, 随着时间窗的增大, 满足可行约束条件的 Pareto 解在不同动态环境下的目标值波动增强, 解的可适用性降低, 时间鲁棒性变差.

表 5 是在不同时间窗下, 两类算法所获得鲁棒 Pareto 解集的鲁棒逆向世代距离 RIGD 的统计结果. 该性能指标体现了鲁棒解集的收敛性和分布性. RIGD 值越小, 表明解集的收敛性越好. 数据对比可见, 当  $T$  取较小值时, 鲁棒 Pareto 解的平均鲁棒世代距离较小. 这是因为随着时间窗  $T$  的增大, 期望的时间鲁棒性更好, 但是受到预测性能影响, 鲁棒 Pareto 解集在多个连续动态环境下的适应值会随着  $T$  的增加偏离真实 PF( $k$ ), 从而导致收敛性下降.

### 3.4.2 环境变化程度对算法的影响

环境变化程度取决于动态变化频率和动态变化强度两方面. 为此, 在固定时间窗  $T = 2$  和稳定性阈值  $\eta = 0.4$  的条件下, 分别针对不同动态变化频率

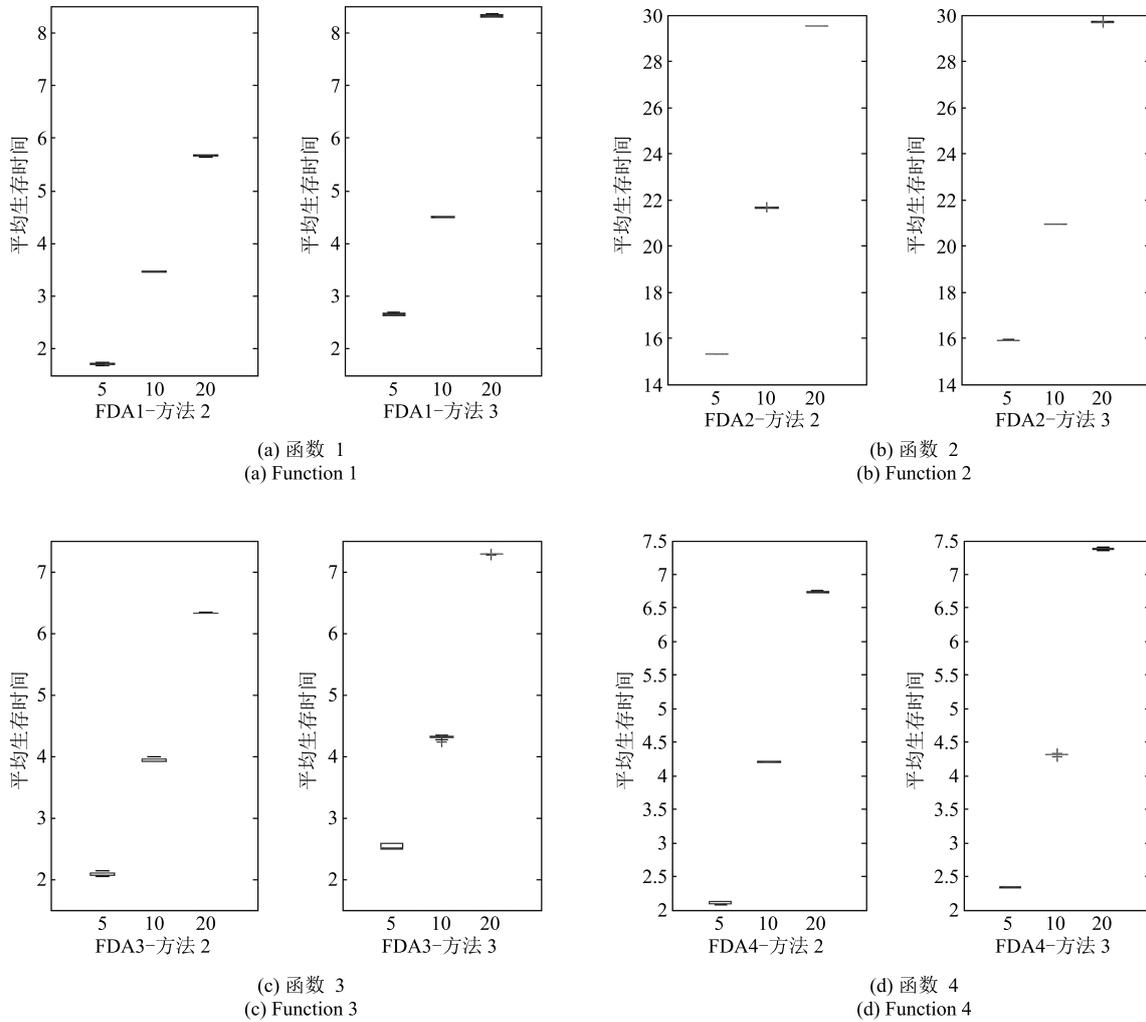
和动态变化强度设计了两类实验. 其一, 固定动态变化频率  $\tau_d = 30$ , 分别设置动态变化强度为  $n_d = 5, 10, 20$ , 对比方法 2 和方法 3 的实验结果, 如图 5 所示; 其二, 固定动态变化强度  $n_d = 10$ , 分别设置动态变化频率为  $\tau_d = 10, 20, 30$ , 对比分析不同算法的实验结果, 如图 6 所示.

1) 图 5 给出了不同动态变化强度下, 基于方法 2 和方法 3 的动态多目标鲁棒优化方法所获得鲁棒 Pareto 解集的平均生存时间黑箱图. 由图可知, 随着参数  $n_d$  增大, 获得的鲁棒 Pareto 解集平均生存时间越大, 并且方法 3 所得解集的平均生存时间略优于方法 2. 这是因为, 当参数  $n_d$  增大时, 相邻动态环境下真实 Pareto 解的动态变化程度减弱, 在一定鲁棒阈值条件下, 每个鲁棒 Pareto 解的生存时间将会增加, 从而导致鲁棒 Pareto 解集的平均生存时间越大.

2) 动态变化频率  $\tau_d$  的取值越小, 代表环境动态变化越频繁. 图 6 针对本文所提两种算法, 在不同的动态变化频率  $\tau_d = 10, 20, 30$  下, 对比 8 个测试函数的鲁棒 Pareto 解集所包含的鲁棒 Pareto 解个数.

表 5 不同时间窗下两种模型的 RIGD 测度比较 ( $\eta = 0.4, n_d = 10, \tau_d = 30$ )Table 5 RIGD of methods 2 and 3 in different time windows ( $\eta = 0.4, n_d = 10, \tau_d = 30$ )

方法	参数	函数 1	函数 2	函数 3	函数 4
方法 2	$T = 2$	$0.2267 \pm 0.0092$	<b><math>0.0816 \pm 0.0555</math></b>	<b><math>0.3815 \pm 0.0230</math></b>	$0.4009 \pm 0.0178$
	$T = 4$	<b><math>0.1944 \pm 0.0081</math></b>	$0.1364 \pm 0.0911$	$0.4142 \pm 0.0296$	$0.4009 \pm 0.0177$
	$T = 6$	$0.2270 \pm 0.0077$	$0.1552 \pm 0.1226$	$0.8548 \pm 0.1841$	<b><math>0.4008 \pm 0.0200</math></b>
方法 3	$T = 2$	$0.2276 \pm 0.0063$	<b><math>0.0849 \pm 0.0559</math></b>	<b><math>0.3842 \pm 0.0215</math></b>	$0.4520 \pm 0.0169$
	$T = 4$	<b><math>0.1976 \pm 0.0066</math></b>	$0.2238 \pm 0.1382$	$0.5184 \pm 0.1738$	<b><math>0.4057 \pm 0.0070</math></b>
	$T = 6$	$0.5411 \pm 0.1200$	$0.1432 \pm 0.0457$	$1.2358 \pm 0.0851$	$0.6615 \pm 0.0048$
方法	参数	函数 5	函数 6	函数 7	函数 8
方法 2	$T = 2$	<b><math>0.4638 \pm 0.0218</math></b>	$0.1009 \pm 0.0754$	$0.3476 \pm 0.0132$	$2.5449 \pm 0.1079$
	$T = 4$	$0.5424 \pm 0.0250$	<b><math>0.0707 \pm 0.0323</math></b>	<b><math>0.3306 \pm 0.0171</math></b>	$2.1396 \pm 0.0972$
	$T = 6$	$0.7334 \pm 0.0571$	$0.0928 \pm 0.0457$	$0.3476 \pm 0.0181$	<b><math>2.0528 \pm 0.0880</math></b>
方法 3	$T = 2$	<b><math>0.5907 \pm 0.0444</math></b>	<b><math>0.0811 \pm 0.0543</math></b>	$0.3477 \pm 0.0128$	<b><math>2.5629 \pm 0.0999</math></b>
	$T = 4$	$0.7655 \pm 0.0278$	$0.1168 \pm 0.1083$	<b><math>0.3328 \pm 0.0089</math></b>	$3.4348 \pm 0.1043$
	$T = 6$	$1.2572 \pm 0.1641$	$0.1812 \pm 0.0857$	$0.9319 \pm 0.0289$	$11.416 \pm 0.2446$



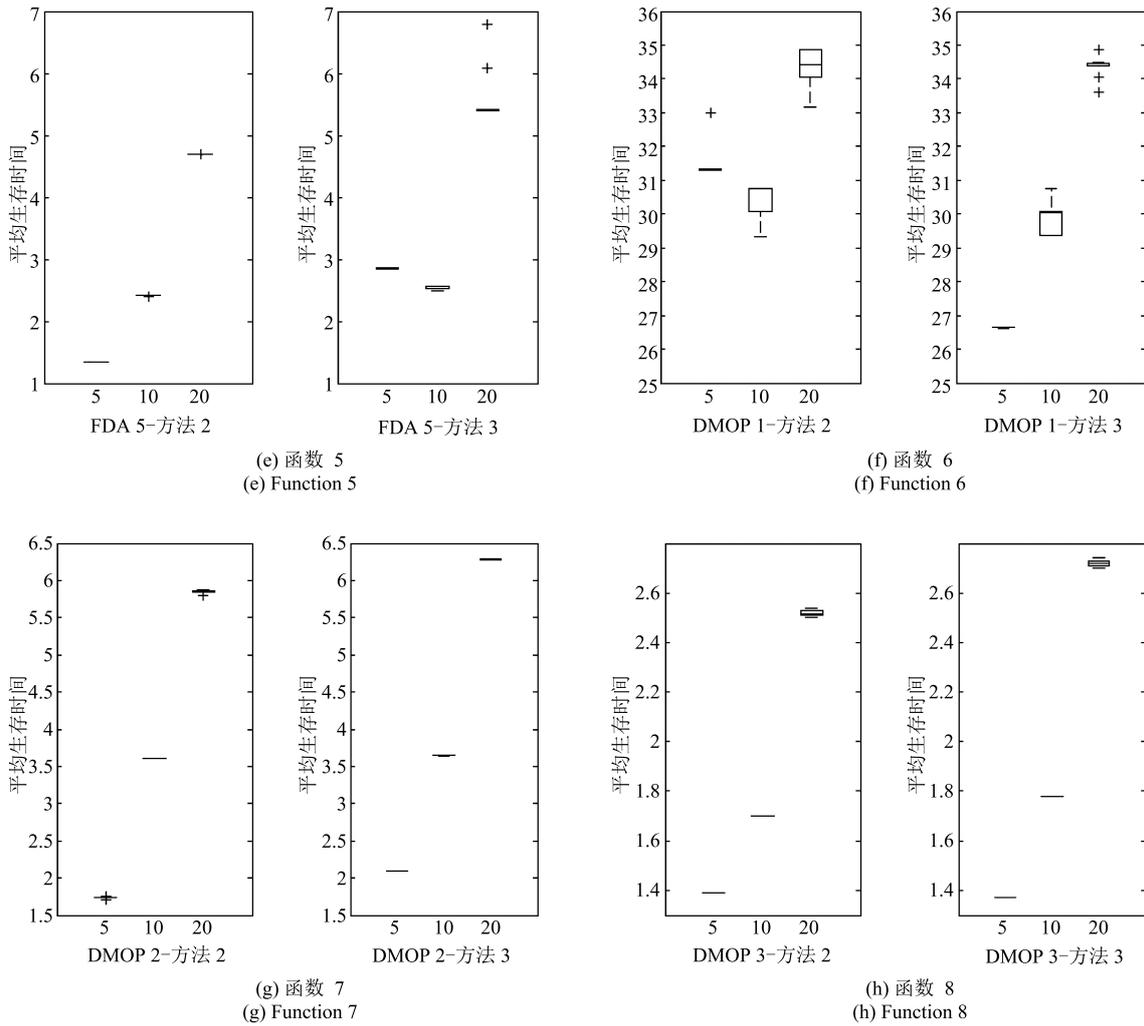


图 5 不同变化强度  $n_d = 5, 10, 20$  下方法 2 和方法 3 的平均生存时间比较 ( $\eta = 0.4, T = 2, \tau_d = 30$ )

Fig. 5 Average survival time of methods 2 and 3 in different change severity  $n_d = 5, 10, 20$  ( $\eta = 0.4, T = 2, \tau_d = 30$ )

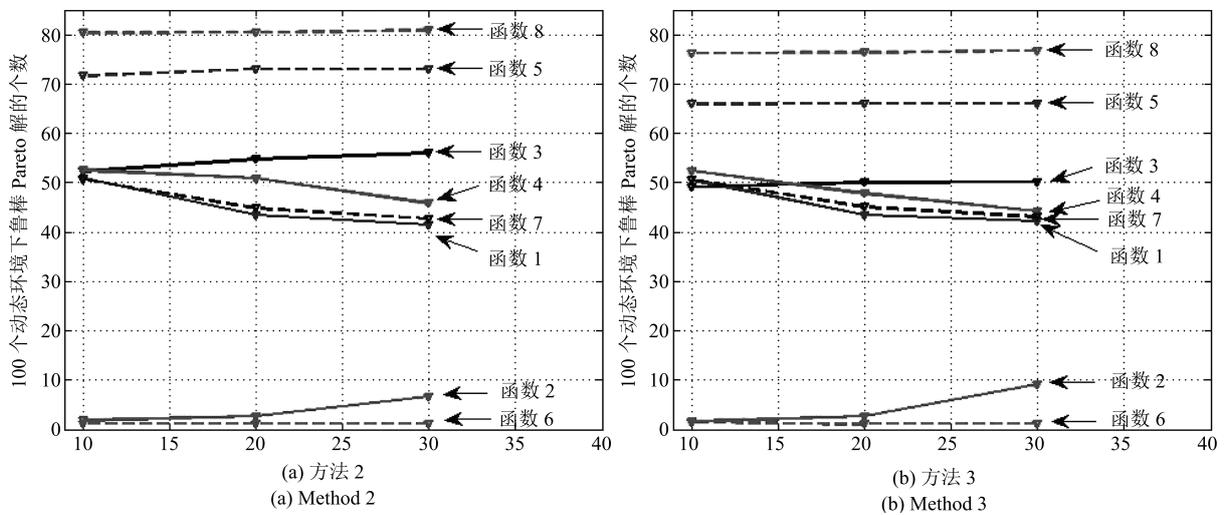


图 6 不同变化频率  $\tau_d = 10, 20, 30$  下方法 2 和方法 3 的平均生存时间比较 ( $\eta = 0.4, T = 2, n_d = 10$ )

Fig. 6 Average survival time of methods 2 and 3 in different change frequency  $\tau_d = 10, 20, 30$  ( $\eta = 0.4, T = 2, n_d = 10$ )

从图中可以看出, 动态变化频率对鲁棒 Pareto 解集规模的影响较小, 表明其解集的平均生存时间波动较小, 时间鲁棒性较稳定.

### 3.4.3 不同算法及模型下的性能比较

针对本文所提出的两种鲁棒优化模型, 考虑在时间窗  $T = 2$ , 且稳定性阈值  $\eta = 0.4$  时, 通过基于分解的鲁棒进化优化方法获得不同动态环境下的鲁棒 Pareto 解. 比较上述两种方法获得的鲁棒 Pareto 解集的平均生存时间如表 6、表 7 和图 7 所示.

1) 如表 6 和图 7 所示, 三种方法所得鲁棒 Pareto 最优解的生存时间都大于等于 1, 代表其适用于至少一个或多个动态环境. 此外, 采用本文所提方法 2 和方法 3 获得的鲁棒 Pareto 最优解集的平均生存时间明显优于方法 1 所得 Pareto 解的平均生存时间, 表明本文所提方法较前者更加合理, 获得的 Pareto 最优解集具有更强的时间鲁棒性. 对比方法 2 和方法 3 所得鲁棒 Pareto 最优解的平均生存时间, 发现方法 3 所获得解的平均生存时间略优于方法 2, 表明适应度满意度这一约束条件对时间鲁

棒性改善具有一定促进作用.

2) 图 8 给出了三种方法所获得鲁棒 Pareto 最优解集中, 平均鲁棒生存时间分别为  $L_i = 1$  和  $L_i = 2$  的解占解集规模的比例, 记为 Per 1 和 Per 2. 三种方法均选取参数  $\eta = 0.4, T = 2$ . 可见, 方法 2 和方法 3 的 Per 1 明显小于方法 1 的 Per 1, 而方法 2 和方法 3 的 Per 2 明显大于方法 1 的 Per 2. 表明本文所给的动态多目标鲁棒优化方法, 可以有效地提高动态鲁棒 Pareto 解集的时间鲁棒性能.

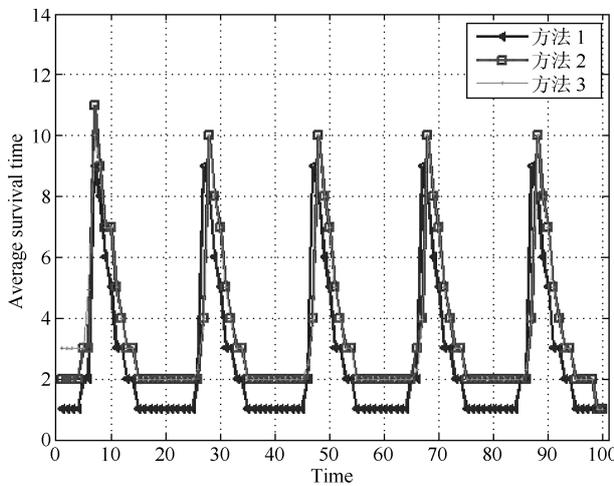
3) 表 7 统计了三种方法在阈值  $\eta = 0.4, T = 2$  下的鲁棒 Pareto 解集的鲁棒逆向世代距离. 可以看出, 本文所提两种方法的 RIGD 值大多数不如方法 1, 主要原因是方法 2 和方法 3 所得鲁棒 Pareto 解集有更好的鲁棒性, 鲁棒性和收敛性在一定程度上存在冲突.

4) 分析 8 个测试函数所得鲁棒 Pareto 最优解集的平均生存时间, 发现函数 2 和函数 6 的结果明显大于其他测试函数. 进一步分析可知, 这两个函数的  $PS(k)$  不随动态环境变化而变化, 但  $PF(k)$  却跟随动态环境变化, 并且变化不显著. 因此, 在进化过

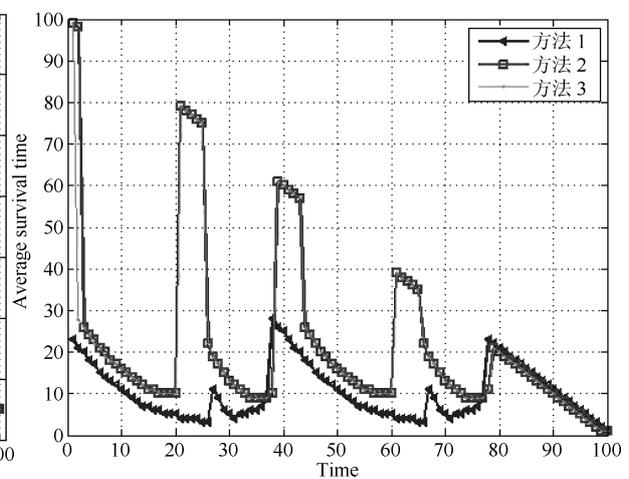
表 6 三种方法获得鲁棒 Pareto 解集的平均生存时间比较 ( $\eta = 0.4, T = 2, n_d = 10, \tau_d = 30$ )

Table 6 Average survival time of robust Pareto solutions on three methods ( $\eta = 0.4, T = 2, n_d = 10, \tau_d = 30$ )

方法	函数 1	函数 2	函数 3	函数 4
方法 1	2.5800 ± 0.0086	10.090 ± 0.1376	2.6200 ± 0.3880	2.7300 ± 0.6077
方法 2	3.4640 ± 0.0099	<b>21.658 ± 0.0086</b>	3.9428 ± 0.0208	4.2027 ± 0.0096
方法 3	<b>4.5073 ± 0.0133</b>	20.941 ± 0.0083	<b>4.3147 ± 0.0280</b>	<b>4.3187 ± 0.0106</b>
方法	函数 5	函数 6	函数 7	函数 8
方法 1	2.9900 ± 0.0560	17.940 ± 0.8734	2.2400 ± 0.3525	1.3500 ± 3.6E-15
方法 2	2.4093 ± 0.0026	<b>30.197 ± 0.5440</b>	3.6000 ± 9.1E-16	1.7000 ± 7.3E-15
方法 3	<b>2.5500 ± 0.0318</b>	30.007 ± 0.4951	<b>3.6520 ± 0.0068</b>	<b>1.7800 ± 4.6E-16</b>



(a) 函数 1  
(a) Function 1



(b) 函数 2  
(b) Function 2

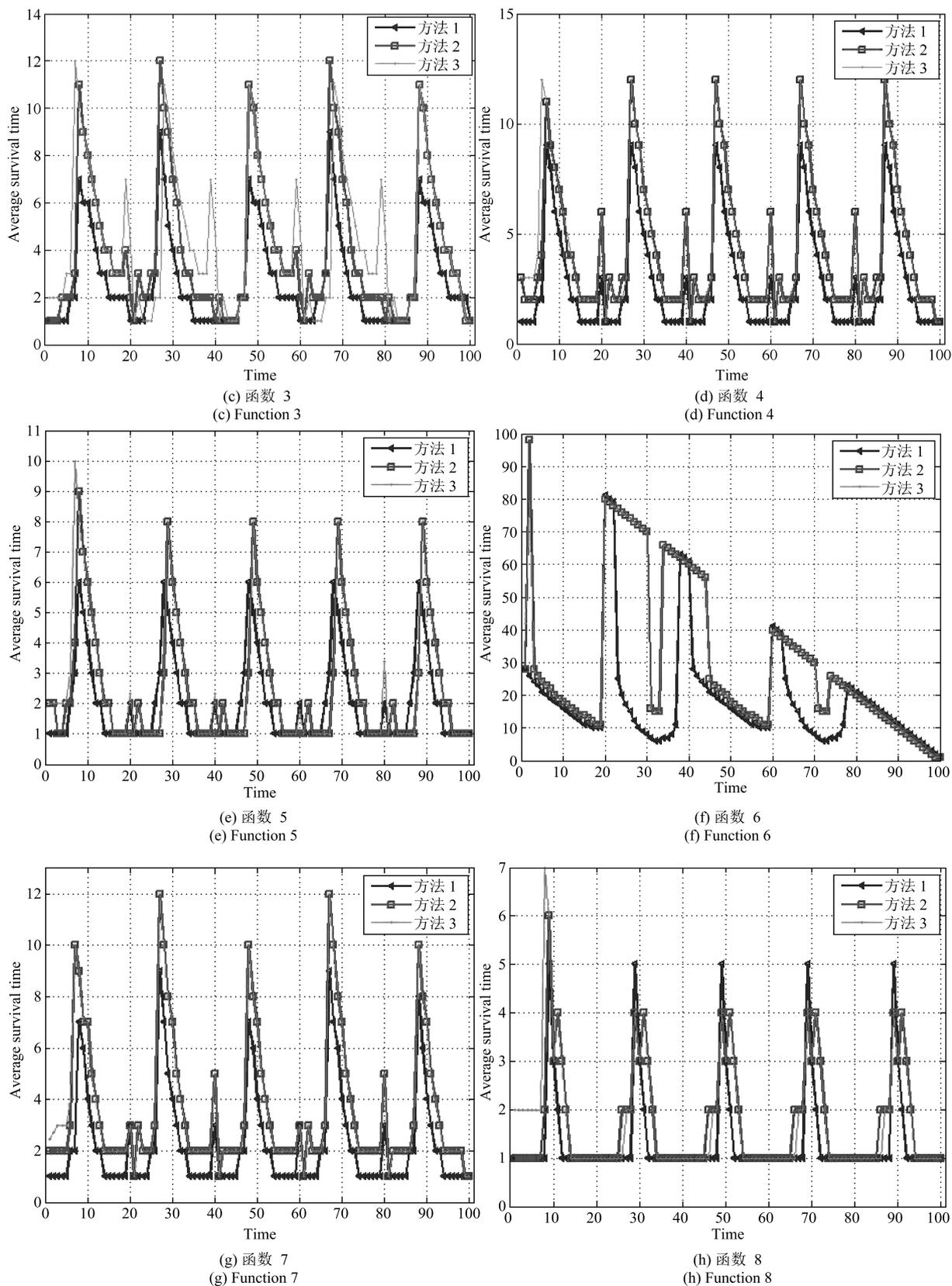


图 7 100 个动态环境下三种方法的鲁棒生存时间比较  $\eta = 0.4, T = 2, n_d = 10, \tau_d = 30$

Fig. 7 Robust survival time of three methods in 100 environments ( $\eta = 0.4, T = 2, n_d = 10, \tau_d = 30$ )

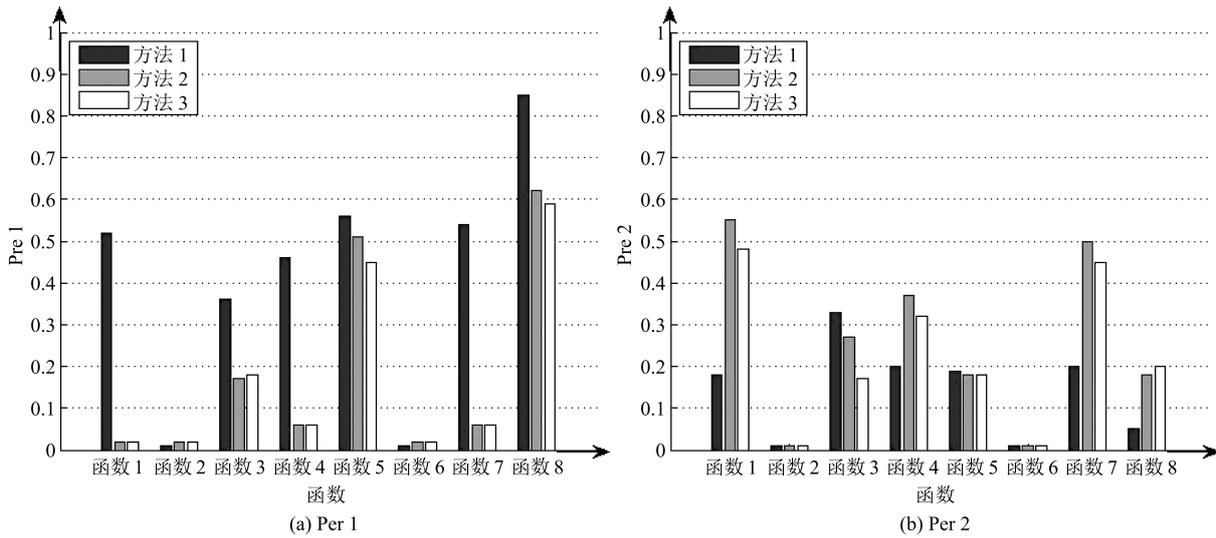


图 8 3 种方法获得鲁棒 Pareto 解集的生存时间百分比比较 ( $\eta = 0.4, T = 2, n_d = 10, \tau_d = 30$ )  
 Fig. 8 Robust survival time percentage of three methods ( $\eta = 0.4, T = 2, n_d = 10, \tau_d = 30$ )

表 7 三种方法获得鲁棒 Pareto 解集的 RIGD 测度比较 ( $\eta = 0.4, T = 2, n_d = 10, \tau_d = 30$ )  
 Table 7 RIGD of robust Pareto solutions on three methods ( $\eta = 0.4, T = 2, n_d = 10, \tau_d = 30$ )

方法	函数 1	函数 2	函数 3	函数 4
方法 1	<b>0.0095 ± 0.0021</b>	0.2438 ± 0.0316	<b>0.1544 ± 0.0579</b>	<b>0.0543 ± 0.0043</b>
方法 2	0.2267 ± 0.0092	<b>0.0816 ± 0.0555</b>	0.3815 ± 0.0230	0.4009 ± 0.0178
方法 3	0.2276 ± 0.0063	0.0849 ± 0.0559	0.3842 ± 0.0215	0.4520 ± 0.0169
方法	函数 5	函数 6	函数 7	函数 8
方法 1	<b>0.3619 ± 0.1683</b>	0.1564 ± 1.04E-7	<b>0.1335 ± 0.0743</b>	<b>0.1866 ± 1.40E-6</b>
方法 2	0.4638 ± 0.0218	0.1009 ± 0.0754	0.3476 ± 0.0132	2.5449 ± 0.1079
方法 3	0.5907 ± 0.0444	<b>0.0811 ± 0.0543</b>	0.3477 ± 0.0128	2.5629 ± 0.0999

程中很容易找到 Pareto 最优解, 且该解对未来连续动态环境的适应性较强, 可以被应用于更多动态变化环境, 导致其动态 Pareto 解集具有较好的鲁棒性.

5) 统一设定问题的动态变化参数为  $\eta = 0.4, T = 2, n_d = 10, \tau_d = 30$ , 且发生 100 次环境变化. 实验硬件为 IBM X3100M5 E3 3.10 GHz CPU, 8 GB 内存的电脑, 软件采用 MATLAB2011b. 表 8 统计了三种不同算法获得鲁棒 Pareto 解集所消耗的运行时间 (s). 比较可得, 采用本文所提两种模型, 可以在相同条件下, 以较少用时得到一组有效的鲁棒 Pareto 解集.

图 9 进一步针对  $\eta = 0.4, T = 2, n_d = 10, \tau_d = 100$  动态环境, 给出了 8 个测试函数的前 4 个鲁棒 Pareto 解与其所适用的  $L_i$  个连续动态环境下真实 Pareto 解的适应度值比较. 每个图中, 虚线为相应方法所得鲁棒 Pareto 最优前沿 RPF( $i$ ), 实线为它所适用的  $L_i$  个相邻动态环境下的真实 Pareto 前沿 PF( $i$ ). 显然, 鲁棒 Pareto 解能够以较好的收敛性逼近相应动态环境下的真实 Pareto 解.

### 4 结论

本文提出了一类寻找动态鲁棒 Pareto 最优解集的进化优化方法. 对于动态多目标优化问题, 动态鲁棒 Pareto 解集是指在某一时刻下的 Pareto 较优解可以以一定稳定性阈值逼近未来多个连续动态环境下的真实前沿, 从而直接作为这些环境下的 Pareto 解集, 以减小计算代价. 首先, 分别通过构造平均生存时间和平均适应度函数来描述解的时间鲁棒性能和性能鲁棒性. 其次, 将上述鲁棒性能评价转化为两类鲁棒优化模型, 采用基于分解的多目标进化优化方法和无惩罚约束处理方法, 构建了动态多目标分解鲁棒进化优化方法. 仿真实验结果表明, 两种模型所获得的动态鲁棒 Pareto 最优解, 在满足适应度满意阈值条件下具有更好的时间鲁棒性. 另外, 对于不同时间窗  $T$  和稳定性阈值  $\eta$  下的解性能分析表明, 较大的  $\eta$  和较小的时间窗有利于鲁棒 Pareto 解集性能. 如何针对不同动态多目标优化问题选取合理的参数是未来有待深入的问题. 此外, 对未来动态环境下的适应度预测在算法中具有显著影响, 因此预测方法的研究也是我们今后的研究工作.

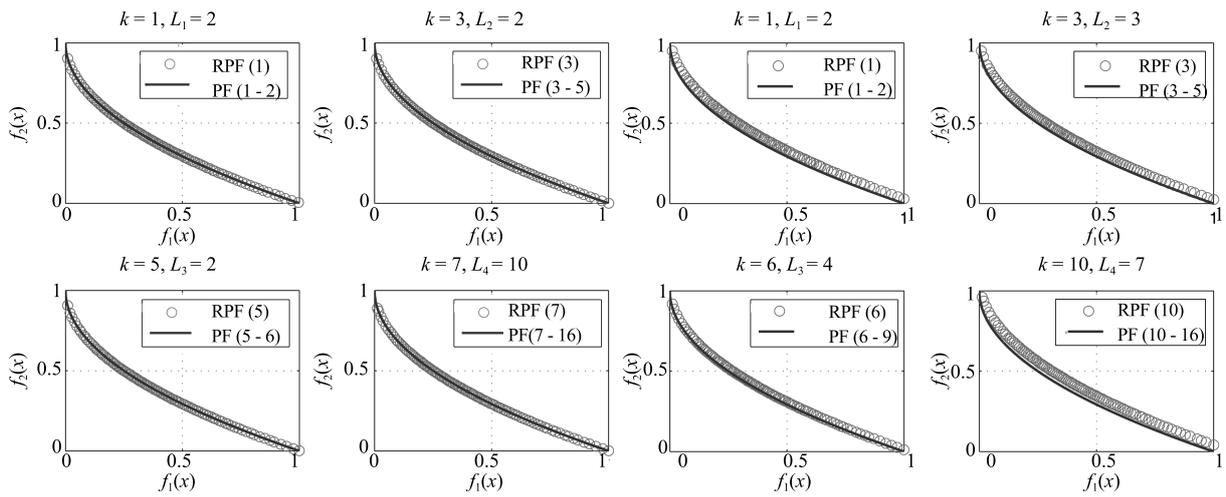
表 8 三种方法求解鲁棒 Pareto 最优解的平均运行时间 (s) 比较 ( $\eta = 0.4, T = 2, n_d = 10, \tau_d = 30$ )

Table 8 Average elapsed time of robust Pareto solutions on three methods ( $\eta = 0.4, T = 2, n_d = 10, \tau_d = 30$ )

方法	函数 1	函数 2	函数 3	函数 4
方法 1	51.3681 ± 3.4762	13.2158 ± 4.0439	70.8508 ± 1.2940	185.9928 ± 10.9027
方法 2	<b>46.0068 ± 1.4818</b>	<b>7.0358 ± 4.2498</b>	59.0582 ± 1.1731	<b>172.3075 ± 7.6934</b>
方法 3	49.3816 ± 1.8679	10.2509 ± 3.2404	<b>55.8101 ± 2.2671</b>	190.0556 ± 19.8364

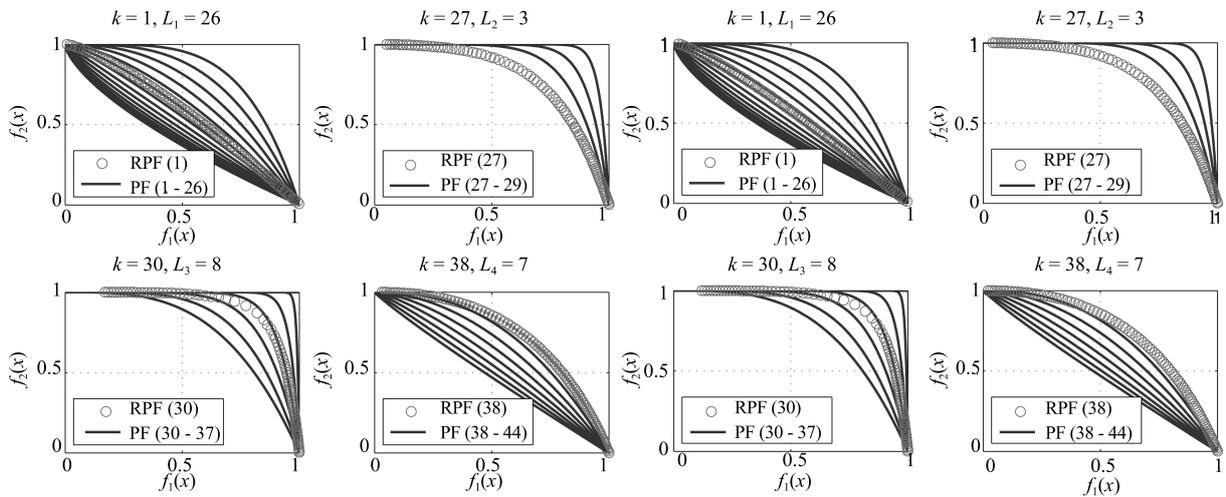
  

方法	函数 5	函数 6	函数 7	函数 8
方法 1	710.6224 ± 70.2617	1.7005 ± 0.0077	54.8422 ± 2.6081	128.4283 ± 2.3889
方法 2	<b>315.2275 ± 8.4846</b>	<b>0.8673 ± 0.0140</b>	<b>45.7206 ± 2.3813</b>	89.7115 ± 0.9206
方法 3	340.7134 ± 21.4715	1.0038 ± 0.0046	47.8325 ± 2.0709	<b>84.1739 ± 0.8742</b>



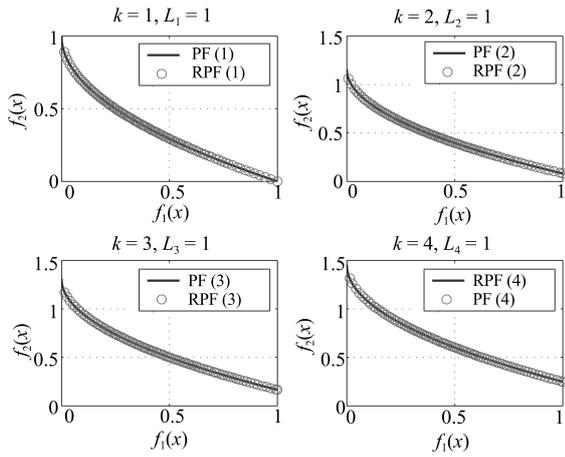
(a) 函数 1-方法 2  
(a) Function 1-method 2

(b) 函数 1-方法 3  
(b) Function 1-method 3

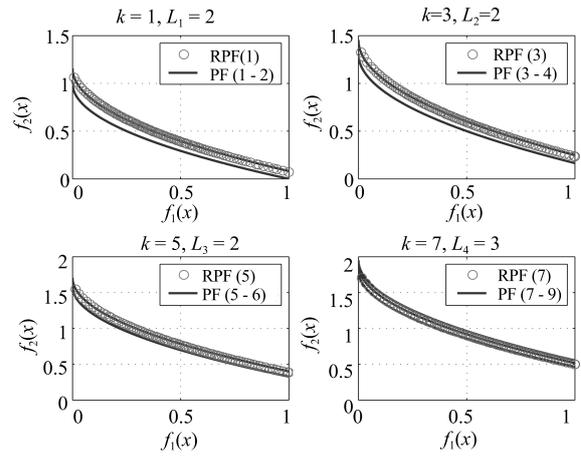


(c) 函数 2-方法 2  
(c) Function 2-method 2

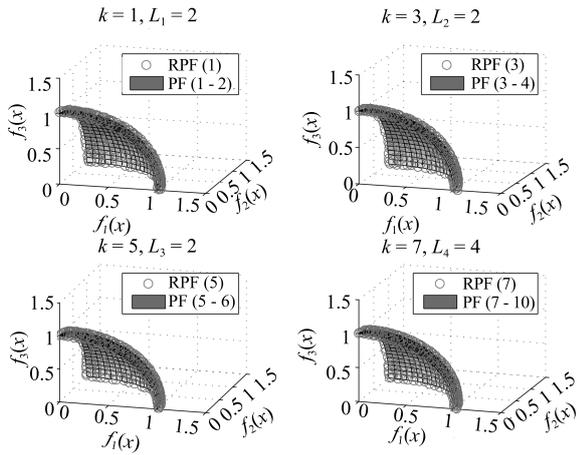
(d) 函数 2-方法 3  
(d) Function 2-method 3



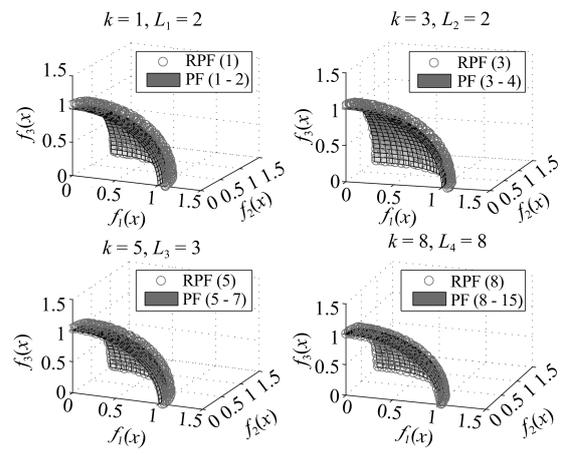
(e) 函数 3-方法 2  
(e) Function 3-Method 2



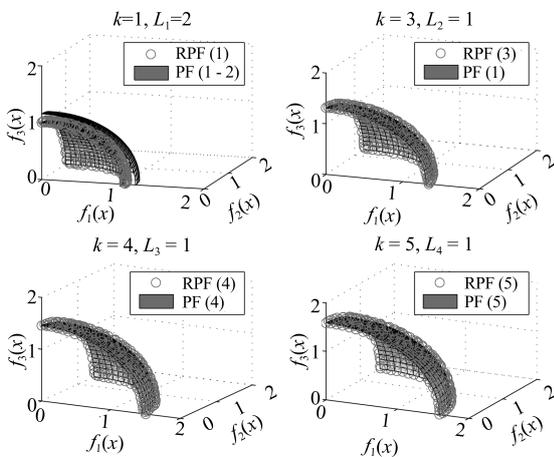
(f) 函数 3-方法 3  
(f) Function 3-method 3



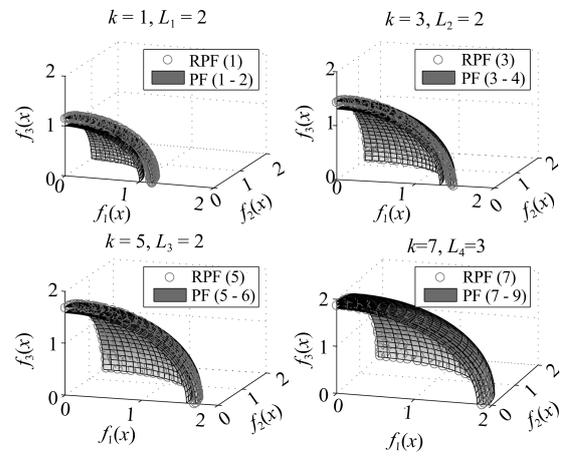
(g) 函数 4-方法 2  
(g) Function 4-method 2



(h) 函数 4-方法 3  
(h) Function 4-method 3



(i) 函数 5-方法 2  
(i) Function 5-method 2



(j) 函数 5-方法 3  
(j) Function 5-method 3

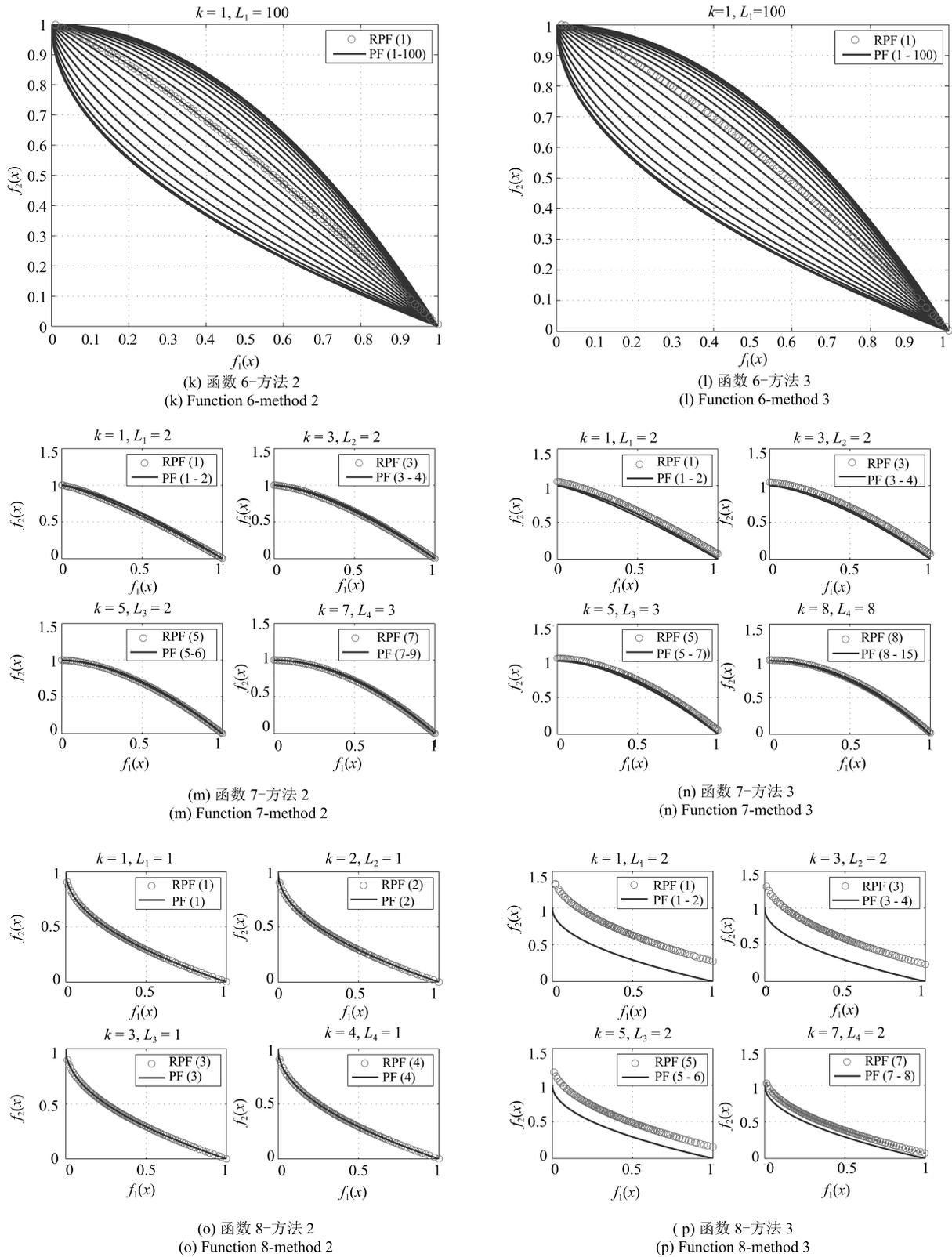


图 9 8 个测试函数的前四个鲁棒 Pareto 解与其所适应的动态环境下的真实 Pareto 解的比较图  
 $(\eta = 0.4, T = 2, n_d = 10, \tau_d = 100)$

Fig. 9 The first four robust Pareto fronts and true Pareto fronts in their adaptive environments  
 $(\eta = 0.4, T = 2, n_d = 10, \tau_d = 100)$

## References

- 1 Farina M, Deb K, Amato P. Dynamic multiobjective optimization problems: test cases, approximations, and applications. *IEEE Transactions on Evolutionary Computation*, 2004, **8**(5): 425–442
- 2 Liu Min, Zeng Wen-Hua. Memory enhanced dynamic multi-objective evolutionary algorithm based on decomposition. *Journal of Software*, 2013, **24**(7): 1571–1588  
(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)
- 3 Hatzakis I, Wallace D. Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation. New York, USA: ACM, 2006. 1201–1208
- 4 Zhou A M, Jin Y C, Zhang Q F, Sendhoff B, Tsang E. Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization. In: Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization. Berlin Heidelberg, Germany: Springer, 2007. 832–846
- 5 Zhou A M, Jin Y C, Zhang Q F. A population prediction strategy for evolutionary dynamic multiobjective optimization. *IEEE Transactions on Cybernetics*, 2014, **44**(1): 40–53
- 6 Goh C K, Tan K C. A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 2009, **13**(1): 103–127
- 7 Hu Cheng-Yu, Yao Hong, Yan Xue-Song. Multiple particle swarms coevolutionary algorithm for dynamic multi-objective optimization problems and its application. *Journal of Computer Research and Development*, 2013, **50**(6): 1313–1323  
(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. 计算机研究与发展, 2013, **50**(6): 1313–1323)
- 8 Jin Y C, Sendhoff B. Constructing dynamic optimization test problems using the multi-objective optimization concept. *Applications of Evolutionary Computing*. Berlin Heidelberg, Germany: Springer, 2004. 525–536
- 9 Nguyen T T, Yang S S, Branke J. Evolutionary dynamic optimization: a survey of the state of the art. *Swarm and Evolutionary Computation*, 2012 **6**: 1–24
- 10 Camara M, Ortega J, de Toro F. A single front genetic algorithm for parallel multi-objective optimization in dynamic environments. *Neurocomputing*, 2009, **72**(16–18): 3570–3579
- 11 Deb K, Gupta H. Introducing robustness in multi-objective optimization. *Evolutionary Computation*, 2006, **14**(4): 463–494
- 12 Yu X, Jin Y C, Tang K, Yao X. Robust optimization over time — a new perspective on dynamic optimization problems. In: Proceedings of the 2010 IEEE Congress on Evolutionary Computation. Barcelona, Spain: IEEE, 2010. 1–6
- 13 Jin Y C, Tang K, Yu X, Sendhoff B, Yao X. A framework for finding robust optimal solutions over time. *Memetic Computing*, 2013, **5**(1): 3–18
- 14 Fu H B, Sendhoff B, Tang K, Yao X. Finding robust solutions to dynamic optimization problems. *Lecture Notes in Computer Science*, 2013, **7835**: 616–625
- 15 Fu H B, Sendhoff B, Tang K, Yao X. Robust optimization over time: problem difficulties and benchmark problems. *IEEE Transactions on Evolutionary Computation*, 2015, **19**(5): 731–745
- 16 Chen M R, Guo Y N, Liu H Y. The evolutionary algorithm to find robust Pareto-optimal solutions over time. *Mathematical Problems in Engineering*, 2015, **2015**: Article ID 814210
- 17 Zhang Q F, Li H. MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 2007, **11**(6): 712–731
- 18 Jan M A, Khanum R A. A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D. *Applied Soft Computing*, 2013, **13**(1): 128–148
- 19 He Shu-Yuan. *Time Series Analysis*. Beijing: Peking University Press, 2003.  
(何书元. 应用时间序列分析. 北京: 北京大学出版社, 2003.)



陈美蓉 中国矿业大学博士研究生。2006 年获得中国矿业大学理学院硕士学位。主要研究方向为进化计算。

E-mail: cmrzl@126.com

(CHEN Mei-Rong Ph.D. candidate at China University of Mining and Technology. She received her master degree from China University of Mining and Technology in 2006. Her main research interest is evolutionary computation.)



郭一楠 中国矿业大学信息与电气工程学院教授。主要研究方向为智能优化算法与控制, 数据挖掘与知识发现。本文通信作者。E-mail: nanfly@126.com

(GUO Yi-Nan Professor at the School of Information and Electronic Engineering, China University of Mining and Technology. Her research interest covers intelligence optimization and control, data mining, and knowledge discovery. Corresponding author of this paper.)



巩敦卫 中国矿业大学信息与电气工程学院教授。1999 年在中国矿业大学获博士学位。主要研究方向为进化计算与应用。E-mail: dwgong@vip.163.com

(GONG Dun-Wei Professor at the School of Information and Electronic Engineering, China University of Mining and Technology. He received his

Ph.D. degree from China University of Mining and Technology in 1999. His research interest covers evolutionary computation and its applications.)



杨振 中国矿业大学硕士研究生。主要研究方向为多目标进化优化。

E-mail: yangzhen.cumt@foxmail.com

(YANG Zhen Master student at China University of Mining and Technology. His main research interest is multi-objective evolutionary optimization.)