

# 基于多点非均匀变异的多目标极值优化算法研究

陈小波<sup>1</sup>, 叶铁丰<sup>1</sup>, 郑明<sup>1</sup>, 潘锡杰<sup>1</sup>, 吴烈<sup>2,3</sup>

(1. 国网浙江瑞安供电公司, 浙江瑞安 325000;

2. 浙江九宏电力工程有限公司, 浙江苍南 325802;

3. 温州大学电气数字化设计技术国家地方联合工程实验室, 浙江温州 325035)

**摘要:** 多目标进化算法因其在解决含有多个矛盾目标函数的多目标优化问题中的强大处理能力, 正受到越来越多的关注与研究; 极值优化作为一种新型的进化算法, 已在各种离散优化、连续优化测试函数以及工程优化问题中得到了较为成功的应用, 但有关多目标 EO 算法的研究却十分有限; 将采用 Pareto 优化的基本原理引入到极值优化算法中, 提出一种求解连续多目标优化问题的基于多点非均匀变异的多目标极值优化算法; 通过对 6 个国际公认连续多目标优化测试函数的仿真实验结果表明: 文章提出算法相比 NSGA-II、PAES、SPEA 和 SPEA2 等经典多目标优化算法在收敛性和分布性方面均具有优势。

**关键词:** 多目标优化; 进化算法; 极值优化; 多点非均匀变异

## Research on Multi-objective Extremal Optimization Algorithm with Multi-non-uniform Mutation

Cheng Xiaobo<sup>1</sup>, Ye Tiefeng<sup>1</sup>, Zheng Ming<sup>1</sup>, Pan Xijie<sup>1</sup>, Wu Lie<sup>2,3</sup>

(1. State Grid Zhejiang Ruian Electric Power Supply Co., Ltd., Ruian 325000, China;

2. Zhejiang Jiuhong Electric Engineering Co., Ltd., Cangnan 325802, China;

3. National-Local Joint Engineering Laboratory of Digitalize Electrical Design Technology, Wenzhou 325000, China)

**Abstract:** Multi-objective evolutionary algorithms have attracted more and more attentions because their powerful solving ability of dealing with several conflict objective functions for a multi-objective optimization problem. As a novel evolutionary algorithm, extremal optimization has been applied successfully to a variety of discrete, continuous optimization test functions and engineering optimization problems, but there is limited research works concerning multi-objective extremal optimization. This paper introduces the basic ideas of Pareto optimization to extremal optimization, and proposes a multi-objective extremal optimization algorithm with multi-non-uniform mutation for solving the continuous multi-objective optimization problems. The simulation experiments on six well-known continuous multi-objective optimization test functions have demonstrated that the proposed algorithm in this paper is superior to other traditional multi-objective optimization algorithms such as NSGA-II, PAES, SPEA and SPEA2 in terms of convergence and diversity.

**Keywords:** multi-objective optimization; evolutionary algorithms; extremal optimization; multi-non-uniform mutation

## 0 引言

大量的实际工程优化问题都可以描述为典型的计及多个性能指标且满足多种约束条件的多目标优化问题<sup>[1]</sup>。传统的多目标优化方法往往将多个目标函数进行加权求和, 整合成单目标优化问题, 而这显然是违背多目标优化的基本思想。多目标进化算法因其在解决多个矛盾目标函数优

化问题时的强大处理能力, 正受到越来越多的研究与关注<sup>[2-4]</sup>。目前, 多目标优化算法存在的主要问题包括: 如何促使算法加快或更好地逼近问题的真实的 Pareto 解集 (True Pareto), 这就是所谓的收敛性问题; 另一个问题就是如何最获得更加均匀分布的最终 Pareto 解集, 这就是所谓的分布性问题<sup>[1,5]</sup>。以上两个问题可以说是多目标优化算法的核心问题所在。

极值动力学优化算法 (Extremal optimization, EO) 作为一种新颖的优化算法<sup>[6-7]</sup>, 最早由 Boettcher 和 Percus 于 1999 年提出, 起源于统计物理领域的自组织临界 (Self-organized criticality, SOC) 理论和 Bark-Sneppen 生物演化模型 (BS 模型)<sup>[8]</sup>。相比于遗传算法 (GA) 和粒子群算法

收稿日期:2017-12-04; 修回日期:2018-01-16。

基金项目:国家自然科学基金(51207112);浙江省自然科学基金(LY16F030011);国网浙江省电力公司群众性创新项目(5211W617000M)。

作者简介:陈小波(1980-),男,浙江瑞安人,工程师,主要从事配网工程管理、配网运维检修管理工作方向的研究。

(PSO) 等传统优化算法而言, EO 算法具有远离平衡态的动力学特征, 从操作上看就是 EO 算法总是选择当前的最差组员和与其有关系的组员来进行变异。所以, 算法的实质操作只有变异操作而不存在交叉等操作。EO 算法自提出起来, 在离散优化、连续优化以及各种工程优化问题中得到了较为成功的应用<sup>[9-16]</sup>, 但有关多目标 EO 算法的研究却十分有限<sup>[17-21]</sup>。

本文将提出一种能够解决连续多目标优化问题的基于多点非均匀变异 (Multi - non - uniform mutation, MNUM) 的多目标极值优化算法, 该算法以下简称为 MOEO-MNUM。该算法的基本思想是将采用 Pareto 优化的基本原理引入到极值优化算法中, 即在对当前个体逐个组员变异后所产生的个体中, 利用 Pareto 优化的概念, 选取其中一个非支配来无条件取代当前个体并进行外部存档, 经逐代优化后获得最终的 Pareto 解集供决策者使用。通过对典型连续多目标优化测试函数的仿真测试实验从而验证本文提出算法相比其它经典多目标优化算法的有效性。

### 1 多目标优化问题

以目标函数均为最小化问题为例, 连续多目标优化问题定义如下<sup>[1]</sup>:

$$\begin{aligned} \min\{ & f_1(x), f_2(x), \dots, f_M(x)\}, x = (x_1, x_2, \dots, x_N) \\ \text{s. t.} & \quad g_j(x) \geq 0, j = 1, 2, \dots, J; \\ & h_k(x) = 0, k = 1, 2, \dots, K; \\ & x_i^L \leq x_i \leq x_i^U \end{aligned} \tag{1}$$

其中:  $N$  表示决策变量的个数,  $M$  表示目标函数的个数,  $J$  为不等式约束的个数,  $K$  为等式约束的个数,  $x_i^L$  为第  $i$  个决策变量的下限,  $x_i^U$  为第  $i$  个决策变量上限。

定义 1 (Pareto 支配, Pareto Dominance) 决策向量  $x_u \in X$ : 若 Pareto 支配决策向量  $x_v \in X$ , 记为  $x_u < x_v$ , 当且仅当:

- 1)  $\forall i \in \{1, \dots, m\}$  满足  $f_i(x_u) \leq f_i(x_v)$ ;
- 2)  $\exists j \in \{1, \dots, m\}$  满足  $f_j(x_u) < f_j(x_v)$ ;

此时, 也称决策向量  $x_v$  Pareto 支配于决策向量  $x_u$ 。若决策向量  $x_u$  与决策向量  $x_v$  不存在 Pareto 支配关系, 则称它们非支配 (Non-dominated)。

定义 2 (Pareto 最优解, Pareto Optimality) 决策向量  $x_u \in X$  称为  $X$  上的 Pareto 最优解, 当且仅当  $\exists x_v \in X$  使得  $x_v < x_u$ 。

定义 3 (Pareto 最优解集, Pareto Optimal Set, POS) 对于给定的多目标优化问题  $f(x)$ , Pareto 最优解集 ( $\rho^*$ ) 定义为:  $\rho^* = \{x_u \in X \mid \exists x_v \in X, x_v < x_u\}$ 。Pareto 最优解集中的个体也称为非支配个体。

定义 4 (Pareto 前沿, Pareto Front, PF) 对于给定的多目标优化问题  $f(x)$  和 Pareto 最优解集 ( $\rho^*$ ), Pareto 前沿 ( $\rho f^*$ ) 定义为:  $\rho f^* = \{f(x_u) \mid x_u \in \rho^*\}$ 。显然

算法得到的最优 Pareto 解集是逼近 Pareto 前沿。

### 2 MOEO-MNUM 算法设计

MOEO-MNUM 算法的主要部分包括: 随机产生单个个体, 利用 MNUM 变异来更新后代, 基于非支配排序的 Pareto 适应度评价准则和一个外部存档更新机制。我们首先给出 MOEO-MNUM 的主要算法流程, 其后在给出流程中使用的细节模块。

#### 2.1 主要算法流程

输入: 一个连续多目标优化测试问题, MOEO-MNUM 算法的可调整参数: 最大迭代次数  $I_{max}$ 、外部存档最大个数  $A_{max}$  和 MNUM 变异的参数  $b$ 。

输出: MOEO-PLM 算法优化后的最终 Pareto 解集。

步骤 1: 随机产生一个个体  $S_c$ , 并令外部存档  $A$  为空。

步骤 2: 通过当前个体  $S_c$  逐个组员进行 MNUM 变异, 并保持其它组员不变, 产生  $N$  个子代个体  $\{S_i, i = 1, 2, \dots, N\}$ , 其中  $N$  为变量个数即组员个数。MNUM 变异具体实现如下方程所示:

$$S_i(x_i) = \begin{cases} S_c(x_i) + (x_i^U - S_c(x_i)) \cdot A(t), & \text{if } r < 0.5 \\ S_c(x_i) + (S_c(x_i) - x_i^L) \cdot A(t), & \text{if } r \geq 0.5 \\ S_c(x_i), & \text{otherwise} \end{cases} \tag{2}$$

$$A(t) = \left[ r_1 \left( 1 - \frac{t}{I_{max}} \right) \right]^b \tag{3}$$

其中:  $r$  和  $r_1$  表示  $0 \sim 1$  范围内的随机数,  $t$  表示算法运行的当前迭代次数。

步骤 3: 对这  $N$  个子代个体  $\{S_i, i = 1, 2, \dots, N\}$  使用基于非支配排序的 Pareto 适应度评价准则进行排序。

步骤 4: 如果只存在一个非支配个体, 令该个体为  $S_{nd}$ ; 如果存在多个则随机选择一个个体作为  $S_{nd}$ 。

步骤 5: 启用基于拥挤度距离的外部存档更新机制 Update\_Achieve ( $S_{nd}$ , Achieve), 用  $S_{nd}$  来更新外部存档。

步骤 6: 将当前个体  $S_c$  无条件替代为  $S_{nd}$ 。

步骤 7: 不断重复步骤 2~7, 直至满足停止条件或达到最大迭代次数。

步骤 8: 将外部存档作为到目前为止最优化的 Pareto 解集输出。

#### 2.2 基于非支配排序的 Pareto 适应度评价准则

MOEO-MNUM 利用 Pareto 支配的概念来定义变异后个体的适应度值。具体适应度赋值准则为: 个体适应度值 = 解集中支配该个体的其他个体的个数。显然, 适应度值为 0 的个体为解集中的非支配个体, 适应度值越小该个体越优。

以两目标优化问题为例, 目标函数分别为  $f_1$  和  $f_2$ , 假设当前个体为  $S = (x_1, x_2, x_3, x_4, x_5)$ , 通过当前个体  $S$  逐个组员变异并保持其它组员不变的基于 MNUM 变异方式, 产生 5 个子代个体, 分别为  $S_A = (x_{m1}, x_2, x_3, x_4,$

$x_5$ ),  $S_B = (x_1, x_{m2}, x_3, x_4, x_5)$ ,  $S_C = (x_1, x_2, x_{m3}, x_4, x_5)$ ,  $S_D = (x_1, x_2, x_3, x_{m4}, x_5)$  和  $S_E = (x_1, x_2, x_3, x_4, x_{m5})$  且如图 1 所示。由 Pareto 适应度评价准则知, 以上 5 个解的 Pareto 适应度值分别为  $f(S_A) = 0$ ,  $f(S_B) = f(S_C) = f(S_D) = 1$ ,  $f(S_E) = 4$ 。

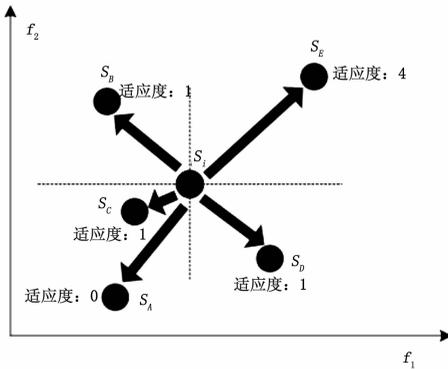


图 1 Pareto 适应度评价准则示意图

### 2.3 外部存档更新机制 Update\_Achieve

MOEO-MNUM 为了不丢失历代寻优过程中的优秀非支配解, 引入外部存档更新机制来保存这些优秀个体。MOEO-MNUM 的外部存档的保存的最大个数是一定的, 有用户自行设置。因此, 当外部存档达到最大个数时, 新的个体若要进入外部存档就需要进行甄别选择, 达到真

正存优的目的。

假设新产生的个体为  $S_{nd}$ , 则 MOEO-MNUM 外部存档更新机制如下<sup>[17]</sup>:

- 1) 如果外部存档中至少有一个体能够支配个体  $S_{nd}$ , 则个体  $S_{nd}$  不能加入外部存档。
- 2) 如果个体  $S_{nd}$  能够支配外部存档中的某些个体, 则将这些个体移除, 并将个体  $S_{nd}$  加入外部存档。
- 3) 如果外部存档中的个体与个体  $S_{nd}$  互不支配时, 若外部存档个数未达到最大个数  $A_{max}$ , 则将个体  $S_{nd}$  加入外部存档; 若外部存档个数达到最大个数  $A_{max}$ , 则若个体  $S_{nd}$  位于外部存档最拥挤的地方 (由下面的拥挤距离准则判断得出), 则不加入外部存档; 否则个体  $S$  将替代掉位于外部存档最拥挤的地方的个体而加入外部存档。

MOEO-MNUM 利用由 Deb 等人<sup>[2]</sup>在 NSGA-II 中提出的拥挤距离准则来选择出外部存档中处于最拥挤的地方的个体。我们首先将新产生的个体  $S_{nd}$  连同外部存档中的所有个体联合在一起计算每个个体的拥挤距离。需要注意的是为保护每个目标的极端值, 将所在个体的拥挤距离赋值为无穷大。

### 3 实验测试与分析

采用国际上通用的 6 个经典多目标标准测试函数<sup>[2]</sup>来验证本文所设计的 MOEO-MNUM 算法的性能。表 1 列出

表 1 典型的连续多目标优化测试问题

测试问题	变量维数 $N$	变量界限	目标函数	Pareto 特点
SCH	1	$[-10^3, 10^3]$	$f_1(x) = x^2$ $f_2(x) = (x - 2)^2$	凸
ZDT1	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{x_1/g(x)}]$ $g(x) = 1 + 9(\sum_{i=2}^N x_i)/(n - 1)$	凸
ZDT2	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - (x_1/g(x))^2]$ $g(x) = 1 + 9(\sum_{i=2}^N x_i)/(n - 1)$	非凸
ZDT3	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{x_1/g(x)} - \frac{x_1}{g(x)}\sin(10\pi x_1)]$ $g(x) = 1 + 9(\sum_{i=2}^N x_i)/(n - 1)$	凸, 不连续
ZDT4	10	$x_1 \in [0, 1]$ $x_i \in [0, 1]$ $i = 2, \dots, N$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{x_1/g(x)}]$ $g(x) = 1 + 10(n - 1) + \sum_{i=2}^N [x_i - 10\cos(4\pi x_i)]$	非凸
ZDT6	10	$[0, 1]$	$f_1(x) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ $g(x) = 1 + 9[(\sum_{i=2}^N x_i)/(n - 1)]^{0.25}$	非凸, 非均匀分布

了本文所用的测试函数 SCH、ZDT1、ZDT2、ZDT3、ZDT4 和 ZDT6，包括测试函数的表达式、变量维数、变量界限、Pareto 前沿特点。

利用文献 [2] 中的两个性能指标来衡量不同算法的性能。第一个指标是收敛性指标  $\gamma$ ，衡量的是算法优化后得到的 Pareto 解集与真实 Pareto 解集的逼近程度。第二个指标是分布性指标  $\Delta$ ，衡量的是算法优化后得到的 Pareto 解集的分布均匀性。计算公式如下：

$$\Delta = \frac{d_f + d_i + \sum_{i=1}^{N-1} |d_i - d_m|}{d_f + d_i + (N - 1)d_m} \quad (4)$$

式中， $d_f$  和  $d_i$  是算法优化后的 Pareto 前沿的边界点和真实 Pareto 前沿的边界点的欧氏距离，而  $d_i$  是算法优化后的 Pareto 前沿中两个连续点之间的欧氏距离， $d_m$  是所有  $d_i$  ( $1, 2, \dots, N - 1$ ) 的平均值。显然一个越好的算法它的收敛性指标越小，分布性指标越小。

MOEO-MNUM 算法的参数设置为  $I_{\max} = 6\ 000$ ， $A_{\max}$

$= 100$ ， $b = 2$ 。所有实验基于 MATLAB2012b 软件进行，运行环境为 3.10 GHz，i5-2400 的 2 GB RAM 的 PC 机，每个测试函数独立运行 30 次。

表 2 列出了 MOEO-MNUM 与 NSGA-II<sup>[2]</sup>、PAES<sup>[22]</sup>、SPEA<sup>[23]</sup> 和 SPEA2<sup>[24]</sup> 等经典多目标进化算法的性能比较数据，包括收敛性指标  $\gamma$  的平均值和方差，分布性指标  $\Delta$  的平均值和方差及其两个指标的排名。为了便于读者对比分析，表格中采用粗体对获得的最好性能指标进行了标记。

从表 2 可以看出，在收敛性方面，尽管 MOEO-MNUM 在求解 SCH 问题时稍逊于 PAES，在其它 5 各测试问题的性能排名第 1，但综合来看 MOEO-MNUM 在收敛性上面还是令人满意的；在分布性方面，MOEO-MNUM 占据很大优势，在各个测试问题上都排名第 1，体现出 MOEO-MNUM 的独特性能即具有优秀的分布性能。

更进一步地，MOEO-MNUM 算法求解 6 个典型测试问题获得的最优 Pareto 前沿如图 2 所示。不难看出，针对

表 2 MOEO-MNUM 与其它经典优化算法的性能比较

测试问题	优化算法	收敛性指标 $\gamma$			分布性指标 $\Delta$		
		$\gamma$ -平均值	$\gamma$ -方差	排名	$\Delta$ -平均值	$\Delta$ -方差	排名
SCH	MOEO-MNUM	0.003312	1.12E-07	2	<b>0.162634</b>	5.97E-05	<b>1</b>
	NSGA-II <sup>[2]</sup>	0.003391	0	3	0.477899	0.003471	2
	PAES <sup>[22]</sup>	<b>0.001313</b>	0.000003	1	1.063288	0.002868	4
	SPEA <sup>[23]</sup>	0.003403	0	4	1.021110	0.004372	3
	SPEA2 <sup>[24]</sup>	/	/	/	/	/	/
ZDT1	MOEO-MNUM	<b>0.001356</b>	<b>1.73E-07</b>	<b>1</b>	<b>0.264432</b>	<b>0.007756</b>	<b>1</b>
	NSGA-II <sup>[2]</sup>	0.033482	4.75E-03	4	0.390307	0.001876	2
	PAES <sup>[22]</sup>	0.082085	8.68E-03	5	1.229794	0.004839	5
	SPEA <sup>[23]</sup>	0.001799	1.00E-06	3	0.784525	0.004440	4
	SPEA2 <sup>[24]</sup>	0.001448	3.17E-04	2	0.472254	0.097072	3
ZDT2	MOEO-MNUM	<b>0.000726</b>	<b>1.12E-08</b>	<b>1</b>	<b>0.190023</b>	<b>0.000537</b>	<b>1</b>
	NSGA-II <sup>[2]</sup>	0.072391	0.031689	4	0.430776	0.004721	2
	PAES <sup>[22]</sup>	0.126276	0.036877	8	1.165942	0.007682	5
	SPEA <sup>[23]</sup>	0.001339	0	3	0.755148	0.004521	4
	SPEA2 <sup>[24]</sup>	0.000743	8.33E-05	2	0.473808	0.0939	3
ZDT3	MOEO-MNUM	0.003567	2.52E-07	1	<b>0.448567</b>	<b>0.002767</b>	<b>1</b>
	NSGA-II <sup>[2]</sup>	0.114500	0.007940	5	0.738540	0.019706	4
	PAES <sup>[22]</sup>	0.023872	0.000010	3	0.789920	0.001653	5
	SPEA <sup>[23]</sup>	0.047517	0.000047	4	0.672938	0.003587	3
	SPEA2 <sup>[24]</sup>	0.003716	0.000586	2	0.606826	0.191406	2
ZDT4	MOEO-MNUM	<b>0.020345</b>	<b>0.000902</b>	<b>1</b>	<b>0.538378</b>	<b>0.045975</b>	<b>1</b>
	NSGA-II <sup>[2]</sup>	0.513053	0.118460	3	0.702612	0.064648	2
	PAES <sup>[22]</sup>	0.854816	0.527238	4	0.870458	0.101399	7
	SPEA <sup>[23]</sup>	7.340299	6.572516	5	0.798463	0.014616	4
	SPEA2 <sup>[24]</sup>	0.028492	0.047482	2	0.705629	0.266162	3
ZDT6	MOEO-MNUM	<b>0.000603</b>	<b>5.43E-10</b>	<b>1</b>	<b>0.180326</b>	<b>0.004123</b>	<b>1</b>
	NSGA-II <sup>[2]</sup>	0.296564	0.013135	5	0.668025	0.009923	2
	PAES <sup>[22]</sup>	0.085469	0.006664	3	1.153052	0.003916	5
	SPEA <sup>[23]</sup>	0.221138	0.000449	4	0.849389	0.002713	4
	SPEA2 <sup>[24]</sup>	0.011643	0.002397	2	0.670549	0.077009	3

每个测试问题, MOEO-MNUM 获得的最优 Pareto 前沿都与真实 Pareto 前沿基本重合。换句话说, MOEO-MNUM 算法具有求解多目标优化问题最优 Pareto 前沿的潜力。

#### 4 结语

本文提出了一种高效解决连续多目标优化问题的基于多点非均匀变异的多目标极值优化算法, 简称为 MOEO-MNUM。通过对国际上通用的 6 个多目标标准测试函数

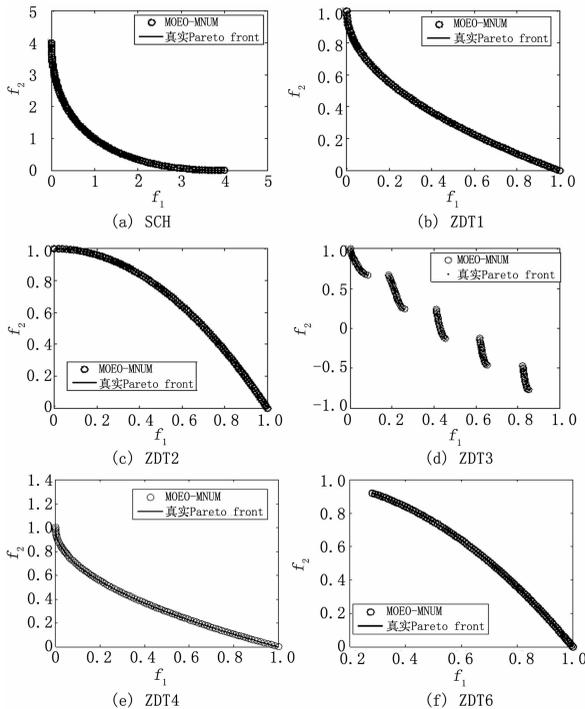


图 2 MOEO-MNUM 求解 6 个典型测试问题获得的最优 Pareto 前沿

(包括 SCH、ZDT1、ZDT2、ZDT3、ZDT4 和 ZDT6) 的仿真实验, 从而验证了本文所设计的 MOEO-MNUM 算法相比与 NSGA-II<sup>[2]</sup>、PAES<sup>[22]</sup>、SPEA<sup>[23]</sup> 和 SPEA2<sup>[24]</sup> 等经典多目标进化算法在多样性和分布性方面都具有明显的优势。另外, 本文也对比了 MOEO-MNUM 求解 6 个典型测试问题获得的最优 Pareto 前沿与真实 Pareto 前沿, 更进一步地验证了 MOEO-MNUM 算法具有求解多目标优化问题最优 Pareto 前沿的潜力。

本文提出的多目标极值优化算法进一步研究方向包括: 1) 通过设计更有效的变异操作因子或引入更高效的外部存档更新机制对算法进行改进; 2) 将本文提出算法推广应用到复杂控制系统优化设计、智能电网多目标规划和能量管理等实际工程优化问题中。

#### 参考文献:

[1] Coello C A C, Lamont G, Van Veldhuizen D. Evolutionary algorithms for solving multiobjective problems, 2nd ed [M]. New York: Springer, 2007.

[2] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2002, 6 (2): 182-197.

[3] Mukhopadhyay A, et al. A survey of multiobjective evolutionary algorithms for data mining: Part I [J]. Evolutionary Computation, 2014, 18 (1): 4-19.

[4] Tusar T, Filipic B. Visualization of Pareto front approximations in evolutionary multiobjective optimization: a critical review and the prosection method [J]. Evolutionary Computation, 2015, 19 (2): 225-245.

[5] Li H, Deb K. Challenges for evolutionary multiobjective optimization algorithms in solving variable-length problems [A]. Evolutionary Computation. IEEE [C]. 2017: 2217-2224.

[6] Boettcher S, Percus A G. Nature's way of optimizing [J]. Artificial Intelligence, 1999, 119 (1): 275-286.

[7] Boettcher S, Percus A G. Optimization with extremal dynamics. [J]. Physical Review Letters, 2001, 86 (23): 5211-5214.

[8] Bak P, Sneppen K. Punctuated equilibrium and criticality in a simple model of evolution [J]. Physical Review Letters, 1993, 71 (24): 4083.

[9] Duch J, Arenas A. Community detection in complex networks using extremal optimization [J]. Physical Review E, 2005, 72 (2): 027104.

[10] 曾国强, 吕勇哉, 毛维杰, 等. 极值优化的理论、算法及其应用 [J]. 控制工程, 2012, 19 (6): 1221-1228.

[11] Guo P, Cheng W, Wang Y. A modified generalized extremal optimization algorithm for the quay crane scheduling problem with interference constraints [J]. Engineering Optimization, 2014, 46 (10): 1411-1429.

[12] Luo J, Chen M R. Multi-phase modified shuffled frog leaping algorithm with extremal optimization for the MDVRP and the MDVRPTW [J]. Computers & Industrial Engineering, 2014, 72: 84-97.

[13] De Falco I, Laskowski E, Olejnik R, et al. Extremal Optimization applied to load balancing in execution of distributed programs [J]. Applied Soft Computing, 2015, 30: 501-513.

[14] Li L M, et al. A novel real-coded population-based extremal optimization algorithm with polynomial mutation: A non-parametric statistical study on continuous optimization problems [J]. Neurocomputing, 2016, 174: 577-587.

[15] Lu Y Z, Chen Y W, Chen M R, et al. Extremal optimization: fundamentals, algorithms, and applications [M]. CRC Press & Chemical Industry Press: City, State abbreviation if USA and Canada, Country, 2016.

[16] De Falco I, Laskowski E, Olejnik R, et al. Parallel extremal optimization in processor load balancing for distributed applications [J]. Applied Soft Computing, 2016, 46: 187-203.

[17] 陈混融. 基于极值动力学的优化方法及其应用研究 [D]. 上海: 上海交通大学, 2008.

[18] Chen M R, Lu Y Z. A novel elitist multiobjective optimization algorithm: Multiobjective extremal optimization [J]. European Journal of Operational Research, 2008, 188 (3): 637-651.