

## 一种基于数据场的多目标引力搜索算法

张爱竹<sup>1</sup>, 孙根云<sup>1†</sup>, 王振杰<sup>1</sup>, 麻德明<sup>2</sup>

(1. 中国石油大学(华东)地球科学与技术学院, 山东 青岛 266580; 2. 国家海洋局第一海洋研究所, 山东 青岛 266061)

**摘要:** 提出一种基于数据场的多目标引力搜索算法(DFMOGSA). 该算法利用外部档案存储非支配解, 同时将外部档案视为目标空间的数据场, 通过计算非支配解的势能来判断每个解的密度; 密度最低的解被选为第1类引导粒子, 直接吸引粒子向低密度区域收敛, 提高解分布的均匀性; 另外, 为了确保算法收敛在种群内, 选择较优粒子作为第2类引导粒子, 通过引力引导粒子搜索. 对比实验结果表明了DFMOGSA算法的有效性和优越性.

**关键词:** 数据场; 引力搜索算法; 多目标优化; 密度; 引导粒子

**中图分类号:** TP301

**文献标志码:** A

## A multi-objective gravitational search algorithm based on data field

ZHANG Ai-zhu<sup>1</sup>, SUN Gen-yun<sup>1†</sup>, WANG Zhen-jie<sup>1</sup>, MA De-ming<sup>2</sup>

(1. School of Geosciences, China University of Petroleum(East China), Qingdao 266580, China; 2. The First Institute of Oceanography, State Oceanic Administration, Qingdao 266061, China)

**Abstract:** A multi-objective gravitational search algorithm(GSA) based on data field(DFMOGSA) is proposed, in which the external archive is applied to store the obtained non-dominated solutions, and is mapped into a data field in the objective space. Accordingly, each non-dominated solution is assigned a density value based on its potential energy. The solution with the smallest density value is chosen as the first kind of guide-particle. This guide-particle directly leads population particles convergence towards the low density region, and thus improves the solution distribution. Moreover, several superior population particles are selected as the second kind of guide-particles. These guide-particles direct each of the population particles fully explore the feasible search space through their resultant gravitational force, which ensures the convergence performance. Simulation results on benchmark test problems show the effectiveness and superiority of the DFMOGSA for multi-objective optimization problems(MOPs).

**Keywords:** data field; gravitational search algorithm; multi-objective optimization; density; guide-particle

### 0 引言

多目标优化问题(MOPs)广泛存在于科学研究和实际工程应用中<sup>[1]</sup>. 与单目标优化致力于寻找一个全局最优解不同, MOPs需要同时满足几个彼此冲突的目标, 因此多目标优化得到的解是一组互不支配解(非支配解), 即Pareto最优解集<sup>[2]</sup>. 元启发式搜索算法具备群体搜索性能, 单次优化运行能够同时获取多个非支配解, 已被成功应用于多目标优化领域<sup>[3-7]</sup>. 其中比较经典的算法包括基于粒子群算法提出的MOPSO<sup>[4]</sup>和基于遗传算法提出的NSGA-II<sup>[5]</sup>、SPEA2<sup>[6]</sup>等多目标优化算法.

引力搜索算法(GSA)<sup>[8]</sup>是在引力定律和牛顿第二定律基础上提出的一种新型的元启发式搜索算法. 该算法中, 每个粒子代表待优化问题的一个解, 粒子之间通过万有引力相互吸引实现位置移动, 从而在搜索空间寻找最优解. GSA具有结构简单、易于实现、参数少、全局优化能力强等特点. 现有的研究表明, GSA的寻优精度和收敛速度都要优于粒子群算法和遗传算法等元启发式搜索算法, 在许多单目标优化问题中得到了成功应用<sup>[9-11]</sup>. 因而, Hassanzadeh等和Nobahari等先后尝试将GSA算法应用到MOPs中, 提出了MOGSA<sup>[12]</sup>、NSGSA<sup>[13]</sup>算法. 但是, 目前应

收稿日期: 2015-10-11; 修回日期: 2015-12-22.

基金项目: 国家自然科学基金项目(41471353); 中央高校基本科研业务费专项资金项目(14CX02039A, 15CX06001A); 海洋公益性行业科研专项项目(201405028).

作者简介: 张爱竹(1988—), 女, 博士生, 从事智能算法、图像数据处理的研究; 孙根云(1979—), 男, 副教授, 博士生导师, 从事智能算法、图像数据处理等研究.

†通信作者. E-mail: genyunsun@163.com

用 GSA 求解 MOPs 的研究还停留在模仿经典多目标优化算法的阶段, 尤其是引导粒子的选择方式还不够成熟. 例如, MOGSA 算法模仿小生境方法为外部档案中的非支配解分配适应度, 然后选择适应度较好的非支配解作为引导粒子; NSGSA 算法中引导粒子的选择主要依赖于 NSGA-II 的非支配排序.

由于多目标优化算法的性能强烈依赖引导粒子的质量(性能), 如何选择收敛性与分布性较好的引导粒子是提高多目标优化算法性能的一个关键问题. 与 GSA 算法类似, 数据场方法也是一种基于物理场论提出的数据挖掘方法<sup>[14]</sup>. 常用的物理场有核辐射衍生数据场、引力衍生数据场和热衍生数据场等<sup>[14]</sup>. 数据场能够表征数据点的密度, 从而反映数据的空间聚类特性, 因而被广泛应用于图像分割与数据聚类领域<sup>[15]</sup>. 受此启发, 本文将数据场理论引入多目标优化, 协同非支配排序方法选择分布性、多样性以及收敛性较好的引导粒子, 提出一种基于数据场的多目标引力搜索算法(DFMOGSA).

在 DFMOGSA 算法中, 提出两种引导粒子的选择方法及对应的学习机制. 第 1 种方法是基于数据场理论, 从外部档案中选取势能最小的非支配解作为第 1 类引导粒子, 该类粒子通过差分方法引导种群粒子; 第 2 种方法是依据非支配关系从种群内部选择支配等级较高的一系列粒子作为第 2 类引导粒子, 该类粒子通过万有引力引导种群粒子. 这两类引导粒子的协同作用能够保证 DFMOGSA 算法的收敛性与所获得的非支配解分布的均匀性. 与其他 5 种算法的对比结果表明了 DFMOGSA 算法求解多目标优化问题的有效性和优越性.

## 1 背景知识

### 1.1 引力搜索算法(GSA)

牛顿万有引力定律表明, 宇宙空间中任意两个物体之间通过万有引力相互吸引, 引力的大小与它们各自的质量(Mass<sub>1</sub> 和 Mass<sub>2</sub>)成正比, 与它们之间距离的平方成反比, 即

$$F = G \frac{\text{Mass}_1 \text{Mass}_2}{R_{12}^2}. \quad (1)$$

其中:  $G$  为万有引力常数,  $R_{12}$  为两物体之间距离. 根据牛顿第二定律, 在万有引力的作用下, 质量小的物体会向质量较大的物体运动. 假设上述粒子 Mass<sub>1</sub> < Mass<sub>2</sub>, 则 Mass<sub>2</sub> 运动的加速度为  $a_2 = F/\text{Mass}_2 = G(\text{Mass}_1/R_{12}^2)$ .

受此启发, GSA 假定在一个大小为  $N$  的种群中, 每个粒子  $i$  是  $D$  维搜索空间的一个物体, 其位置可以表示为  $x_i = [x_i^1, x_i^2, \dots, x_i^D]$ , 对应的速度为  $v_i = [v_i^1, v_i^2, \dots, v_i^D]$ . 每个粒子的适应度值(fit <sub>$i$</sub> )被归一化

之后作为粒子的质量, 即

$$\text{Mass}_i(t) = \frac{nm_i(t)}{\sum_{j=1}^N nm_j(t)}. \quad (2)$$

其中  $nm$  表示归一化后的适应度值, 即

$$nm_i(t) = \frac{\text{fit}_i(t) - \text{worst}(t)}{\text{best}(t) - \text{worst}(t)}.$$

对于最小化问题, best 和 worst 分别为适应度的最小和最大值.

粒子受种群内部其他粒子施加的万有引力吸引, 并遵循牛顿第二定律向着质量较大的粒子运动, 最终完成收敛. 粒子的速度和位置更新方式为

$$v_i^d(t+1) = \text{rand}v_i^d(t) + a_i^d(t), \quad (3)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1). \quad (4)$$

### 1.2 数据场

数据场是一种与物理学中各种能量场(比如电场、磁场、重力场等)类似的数域空间, 其本身具有一些与物理场类似的特性<sup>[14]</sup>. 以引力衍生数据场为例, 场内各个数据都独立地向外辐射并接受其他数据的辐射能量, 类似于宇宙空间的物体对其他物体施加引力并同时被其他物体吸引. 因而, 数域空间中任意一个数据点都受其他数据点辐射能量的影响, 其所受数据辐射能量之和, 称为该点处数据场的势能<sup>[14]</sup>. 势能的大小能够描述数据的分布状态, 引力衍生数据场中数据点  $x_i$  的势能计算公式为

$$\psi(x_i) = \sum_{j=1, j \neq i}^N \frac{\rho_j \sigma}{R_{ij}^2}. \quad (5)$$

其中:  $\rho_j$  为粒子  $j$  的属性值, 在图像中可以设为灰度值、灰度频率或常数等, 在 GSA 算法中可以表示为粒子质量;  $\sigma$  为影响因子, 类似于万有引力算法中的万有引力常数  $G$ . 图 1 为一组离散的二维数据点(离散点●)及其对应的三维势能图.

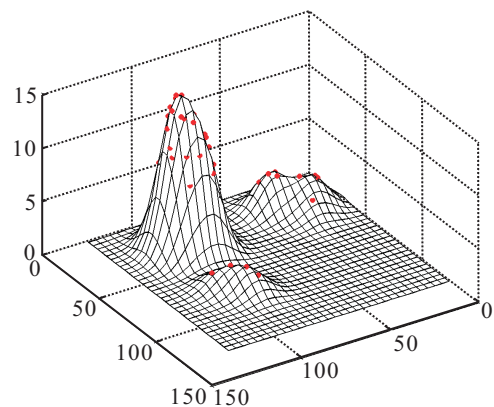


图 1 势能图像

如图 1 所示, 数据场的势能可以描述数据的聚集特性和密度, 势能越高的数据点, 所处位置的数据密度越高. 因此可以利用这一特性描述外部档案中非支

配解的拥挤程度,用来指导引导粒子的选择.

### 1.3 多目标优化问题

多目标优化问题通常被描述为

$$\begin{aligned} \min f(x) &= [f_i(x), i = 1, 2, \dots, M]; \\ \text{s.t. } g_j(x) &\leq 0, j = 1, 2, \dots, J. \end{aligned} \quad (6)$$

其中:  $f(x)$  为待求解的多目标优化问题(这里假设为最小化问题),  $x$  为  $D$  维搜索空间中  $N$  个备选解的集合(即  $x = [x_1, \dots, x_i, \dots, x_N]$  且  $x_i = [x_i^1, x_i^2, \dots, x_i^D]$ ),  $M$  为目标数,  $f_i(x)$  为第  $i$  个目标的适应度值,  $g_j(x)$  为  $j$  个约束条件,  $J$  为总的约束条件数.

不同于单目标优化,多目标优化需要同时对多个目标进行优化,而这些目标之间又互相冲突.所以,对于多目标优化问题,不存在唯一的全局最优解.算法寻优过程中,按照粒子之间的支配关系可以同时得到多个非支配解,每个非支配解通常称为 Pareto 最优解,所有 Pareto 最优解构成的解集即为 Pareto 最优解集.

## 2 DFMOGSA

由于引导粒子的质量(性能)直接影响多目标优化算法的性能,本文将数据场引入多目标引力搜索算法中,提出一种基于数据场的多目标引力搜索算法(DFMOGSA).该算法中,为了选择分布性、多样性、收敛性较好的引导粒子,分别基于数据场和非支配排序方法选择两类引导粒子.第1类引导粒子在外部档案中依据数据场原理进行选择,并利用差分方法对种群粒子进行直接的引导;第2类引导粒子在种群中基于非支配关系选择,并通过万有引力对种群粒子进行引导.通过两类引导粒子的共同作用,使得算法能够较全面、迅速地完成了对可行区域的搜索,获取高质量的非支配解集.

### 2.1 基于数据场的引导粒子选择

与一些经典的多目标优化算法相同,本文首先将非支配解存储于一个外部档案 Rep 中;然后将外部档案映射到数据场空间,通过计算数据的势能选择第1类引导粒子.本文选择的数据场是引力衍生场,具体步骤如下.

首先统计当前  $t$  时刻外部档案中非支配解的个数  $N_{\text{rep}}(t)$ ;然后根据引力衍生场的场强函数(5)计算各个非支配解的势能,即

$$\psi(\text{rep}_i(t)) = \sum_{j=1, j \neq i}^{N_{\text{rep}}(t)} \frac{\rho_j \sigma}{R_{ij}^2}. \quad (7)$$

其中:  $R_{ij}(t)$  为  $t$  时刻非支配解  $\text{rep}_i$  与  $\text{rep}_j$  在目标空间的欧氏距离,  $\rho_j$  为第  $j$  个非支配解的质量.如前所述,在单目标 GSA 算法中,质量通常用适应度值来表示,但是在多目标算法中,由于所有的非劣解都互不

支配,基于适应度值不能比较粒子的优劣,无法直接通过适应度值计算非支配解的质量.但是,数据场理论主要描述的是数据分布的密集程度,因而在目标空间将质量设定为1,并不影响数据分布特性的计算.因此,本文将  $\rho_j$  设为常数1,以简化运算.

得到各非支配解的势能之后,根据势能的大小对其排序,选择势能最小的粒子作为第1类引导粒子,命名为 GF.如果同时存在多个势能最小的粒子,则从中随机选择一个粒子作为 GF.  $t$  时刻 GF 对种群粒子  $i$  的引导方式为

$$\text{GF}v_i(t+1) = \text{GF}(t) - x_i(t). \quad (8)$$

其中:  $x_i$  为粒子  $i$  在搜索空间的位置,  $\text{GF}v_i$  为粒子  $i$  被引导粒子 GF( $t$ ) 吸引产生的速度.

根据数据场理论,GF( $t$ ) 为位于外部档案中数据最稀疏区域的非支配解,其对种群粒子的吸引可以促使粒子向非支配解分布稀疏的区域收敛,从而使解的分布更加均匀.

### 2.2 基于非支配排序的引导粒子选择

单纯受 GF 的吸引,会导致算法收敛过快,种群多样性降低,从而出现早熟收敛.为此,在 DFMOGSA 算法中,从种群内部选择第2类引导粒子来增强引导粒子的多样性.具体过程如下.

首先,对种群内的粒子进行非支配排序,处于同一支配层的粒子被赋予相同序号(层序),该层序设定为粒子的适应度值.假设种群内所有粒子分为  $L$  层,第1层的粒子适应度值均为  $\text{fit} = 1$ ,第  $L$  层的粒子适应度值均为  $\text{fit} = L$ .根据式(2)计算各粒子的质量 Mass,显然,层序越小的粒子,质量越大,能够更好地引导种群粒子.

然后,选择排序在前  $[K_{\text{best}} \times L]$  层( $\lceil \cdot \rceil$  表示向上取整)的  $N_{\text{GS}}$  个非支配解作为第2类引导粒子 GS.这类引导粒子通过万有引力对种群内所有粒子进行引导.具体而言,GS 对种群中粒子  $i$  的引导方式为

$$\text{GS}v_i(t+1) = \sum_{j=1, \text{GS}_j(t) \neq x_i(t)}^{N_{\text{GS}}} G \frac{\text{Mass}_{\text{GS}_j}(t)}{R_{ij}(t)} (\text{GS}_j(t) - x_i(t)), \quad (9)$$

其中万有引力常数  $G$  是一个随时间逐渐变小的实数.本文算法的设置与 GSA 算法中保持一致,即  $G = G_0 e^{-\beta/T_{\text{max}}}$ ,其中  $G$  为初始引力常数,  $\beta$  为缩减系数,  $T_{\text{max}}$  为最大迭代次数.

为了使算法在收敛前期进行广泛的搜索,后期实现快速的收敛,  $K_{\text{best}}$  的大小随着迭代过程线性减小,即  $K_{\text{best}} = 1 - t/T_{\text{max}}$ .

DFMOGSA 算法中,每个种群粒子在这两类引导粒子的共同作用下运动,粒子  $x_i$  的速度和位置更新公

式分别为

$$v_i(t+1) = r \times v_i(t) + GFv_i(t+1) + GSv_i(t+1), \quad (10)$$

$$x_i(t+1) = x_i(t) + v_i(t+1), \quad (11)$$

其中  $r$  是  $[0,1]$  之间的随机数。

### 2.3 算法流程

DFMOGSA 算法的详细流程如下:

Step 1: 设定目标个数  $M$ , 种群大小  $N$ , 搜索空间维度  $D$ , 空间上下限范围  $ub$ 、 $lb$ , 外部档案最大值  $M_{Rep}$ , 最大适应度计算次数 FEs.

Step 2: 初始化每个粒子的位置  $x_i$  和速度  $v_i$ .

Step 3: 计算每个粒子在各个目标上的适应度值, 根据支配关系更新外部档案 Rep.

Step 4: 判断 Rep 内非支配解的数目是否超出容量限制  $M_{Rep}$ , 如果超出容量限制, 则进行 Step 5, 否则进行 Step 6.

Step 5: 根据 2.1 节的方法, 计算外部档案中每个粒子的势能, 逐个删除势能最大的非支配解, 直至非支配解数目为  $M_{Rep}$ .

Step 6: 根据 2.1 节的方法, 计算外部档案中每个粒子的势能, 从中选择第 1 类引导粒子 GF.

Step 7: 根据 2.2 节的方法, 从种群粒子中选择第 2 类引导粒子 GS.

Step 8: 根据式 (10) 和 (11), 更新每个种群粒子的速度和位置.

Step 9: 判断是否达到终止条件 FEs. 若不满足, 则循环操作 Step 3 ~ Step 9; 若满足, 则终止循环, 输出 Rep 中存储的非支配解.

## 3 实验结果与分析

为了定性和定量地分析 DFMOGSA 算法求解 MOPs 的有效性, 充分测试算法的性能, 本文选择 6 个具有不同特点的标准测试函数 (无约束函数 ZDT1<sup>[5]</sup>、ZDT3<sup>[5]</sup>、ZDT4<sup>[5]</sup>、DTLZ2<sup>[16]</sup> 和有约束函数 BNH<sup>[17]</sup>、OSY<sup>[17]</sup>) 进行优化. 各个函数表达式如表 1 所示. 比较算法为 3 种经典的多目标优化算法: SPEA2、MOPSO 和 NSGA-II, 以及两种基于 GSA 算法的多目标优化算法: MOGSA 与 NSGSA. 本文处理约束优化问题时采用 MOPSO 算法中的处理方法, 即: 当解之间两两比较更新外部档案时, 先检查它们是否违反约束条件. 如果两个解都不违反任何约束条件, 则处于支配地位的解取胜; 如果其中只有一个解违反约束条件, 则另外一个解取胜; 如果两个解都违反了约束条件, 则违反约束条件少的解取胜 (违反条件数目相等时, 随机选一个作为胜出解).

表 1 测试函数

函数	搜索空间	目标函数
ZDT1	$D = 30$ $x \in [0, 1]^{30}$	$f_1(x) = x_1$
		$f_2(x) = h(x) \left[ 1 - \sqrt{\frac{x_1}{h(x)}} \right]$
		$h(x) = 1 + 9 \left( \sum_{i=2}^D x_i \right) / (D - 1)$
ZDT3	$D = 30$ $x \in [0, 1]^{30}$	$f_1(x) = x_1$
		$f_2(x) = h(x) \left[ 1 - \sqrt{\frac{x_1}{h(x)}} - \frac{x_1}{h(x)} \sin(10\pi x_1) \right]$
		$h(x) = 1 + 9 \left( \sum_{i=2}^D x_i \right) / (D - 1)$
ZDT4	$D = 10$ $x_1 \in [0, 1]$ $x_i \in [-5, 5]$ $i = 2, 3, \dots, 10$	$f_1(x) = x_1$
		$f_2(x) = h(x) \left[ 1 - \sqrt{\frac{x_1}{h(x)}} \right]$
		$h(x) = 1 + 10(D - 1) + \sum_{i=2}^D [x_i^2 - 10 \cos(4\pi x_i)]$
DTLZ2	$D = 12$ $x \in [0, 1]^{12}$	$f_1(x) = (1 + h(x)) \cos\left(\frac{x_1\pi}{2}\right) \cos\left(\frac{x_2\pi}{2}\right)$
		$f_2(x) = (1 + h(x)) \cos\left(\frac{x_1\pi}{2}\right) \sin\left(\frac{x_2\pi}{2}\right)$
		$f_3(x) = (1 + h(x)) \sin\left(\frac{x_1\pi}{2}\right)$
BNH	$D = 2$ $x_1 \in [0, 5]$ $x_2 \in [0, 3]$	$f_1(x) = 4x_1^2 + 4x_2^2$
		$f_2(x) = (x_1 - 5)^2 + (x_2 - 5)^2$
		$g_1(x) = (x_1 - 5)^2 + x_2^2 \leq 25$ $g_2(x) = (x_1 - 8)^2 + (x_2 + 3)^2 \geq 7.7$
OSY	$D = 6$ $x_1 \in [0, 10]$ $x_2 \in [0, 10]$ $x_3 \in [1, 5]$ $x_4 \in [0, 6]$ $x_5 \in [1, 5]$ $x_6 \in [0, 10]$	$f_1(x) = -[25(x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 1)^2]$
		$f_2(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2$
		$g_1(x) = x_1 + x_2 - 2 \geq 0$
		$g_2(x) = 6 - x_1 - x_2 \geq 0$
		$g_3(x) = 2 - x_2 + x_1 \geq 0$
		$g_4(x) = 2 - x_1 + 3x_2 \geq 0$ $g_5(x) = 4 - (x_3 - 3)^2 - x_6 \geq 0$ $g_6(x) = (x_5 - 3)^2 + x_6 - 4 \geq 0$

### 3.1 参数设置

为了公平地比较不同算法的性能, 设置所有算法对所有函数进行优化的终止条件为最大适应度计算次数, 即 FEs = 250 000, 种群大小为  $N = 100$ . 外部档案大小为  $M_{Rep} = 100$ , 对比算法的其他参数都按照原文献经过测试后的参数进行设定. DMMOGSA 中设置  $G = 100$ ,  $\beta = 20$ ,  $\sigma = 10^{-6}$ . 为了避免偶然性, 每个算法对每个测试函数分别进行 30 次独立实验, 并计算它们的统计值.

### 3.2 测试函数

表 1 详细列出了各测试函数可行的搜索范围、空间维度以及目标函数的表达式.

表 1 所示的测试函数中, ZDT 系列的 3 个函数为双目标函数, 其中 ZDT1 的 Pareto 前端为连续凸型,

ZDT3 的 Pareto 前端为非连续凸型, ZDT4 有多个不同的局部凸型 Pareto 前端; DTLZ2 为 3 个目标的函数且 Pareto 前端为多峰凹型; BNH 为带 2 个约束条件的函数, Pareto 前端为连续凸型; OSY 为带 6 个约束条件的函数, Pareto 前端为非凸型。

### 3.3 性能评价指标

现有研究中, 一般通过计算所获得解逼近真实解的程度与解集分布性来评价多目标优化算法的性能。具体而言, 常用当代距离 (GD) 和分布性 (SP) 两个指标来衡量算法的收敛性和分布性, 本文也采用这两个性能指标对算法性能进行比较。GD 和 SP 定义如下:

$$GD = \frac{1}{n} \sqrt{\sum_{i=1}^n \text{dist}_i^2}, \quad (12)$$

$$SP = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\overline{\text{dist}} - \text{dist}_i)^2}. \quad (13)$$

其中:  $n$  为通过算法得到的非支配的个数,  $\text{dist}_i$  为算法得到的第  $i$  个解到真实 Pareto 前端的最小距离,  $\overline{\text{dist}}$  为  $\text{dist}_i$  的平均值。GD 越小, 说明通过算法得到的非支配解集越逼近真实解集, 收敛性越好; SP 越小, 说明算法得到的非支配解集分布的均匀性越好。

### 3.4 实验结果评价

为了定性地比较各算法的效果, 图 2~图 7 给出了不同算法得到的 GD 中值对应的 Pareto 前端的结果。其中浅色的点是真实 Pareto 前端, 深色的点是各算法产生的 Pareto 前端。表 2 与表 3 分别为 6 种算法关于 6 个测试函数的 GD 与 SP 值。

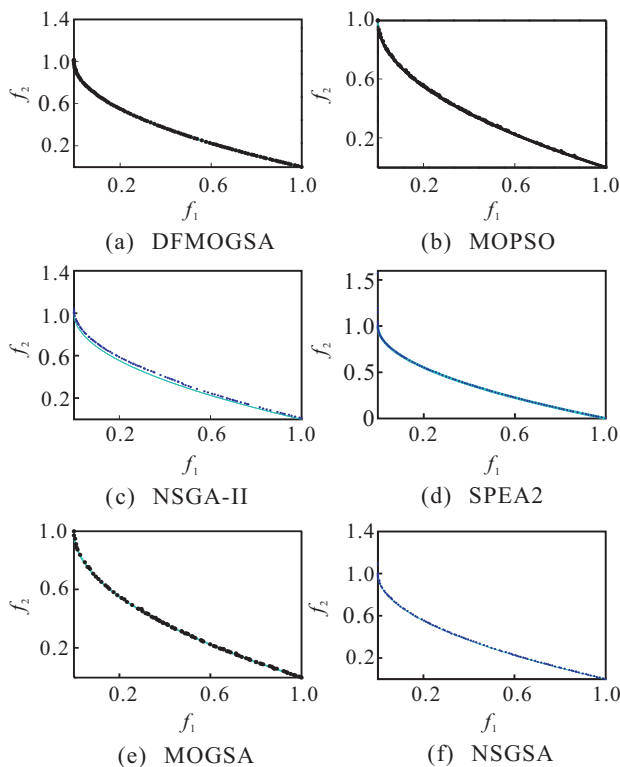


图 2 6 种算法对于 ZDT1 获得的 Pareto 前端

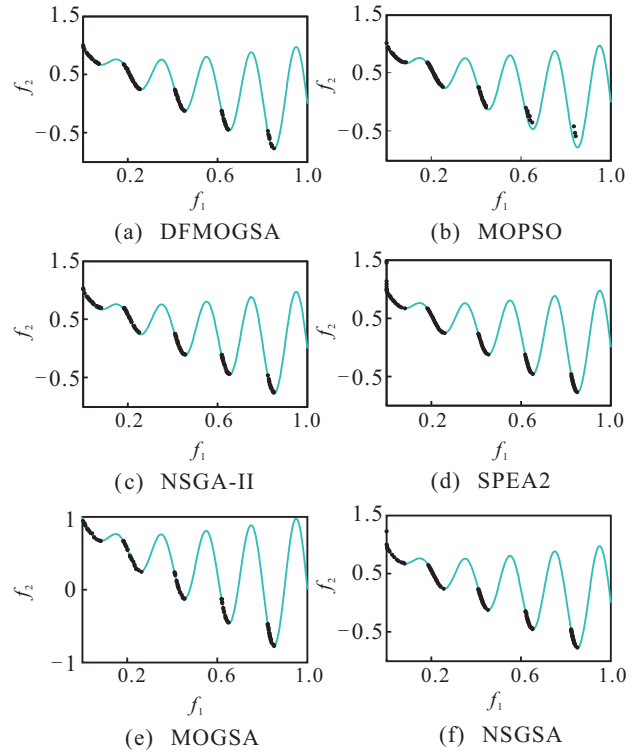


图 3 6 种算法对于 ZDT3 获得的 Pareto 前端

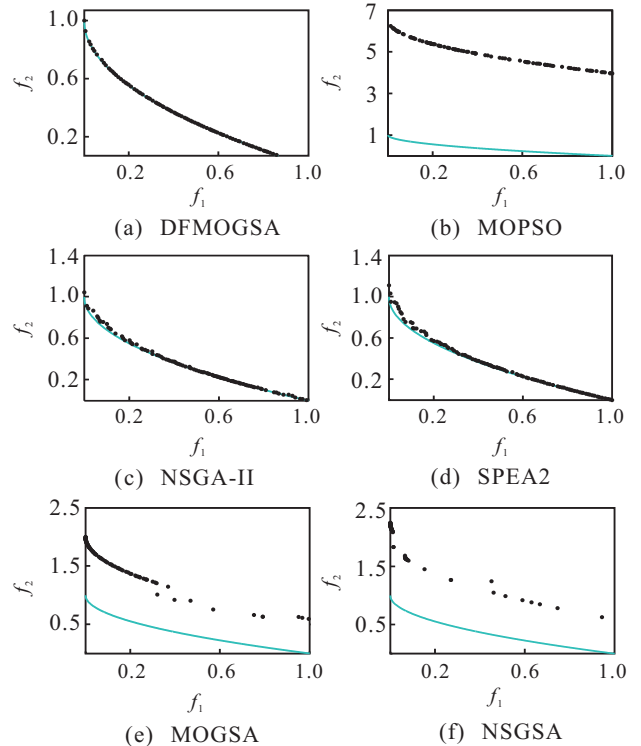


图 4 6 种算法对于 ZDT4 获得的 Pareto 前端

由表 2 和表 3 可以看出, 对于 ZDT1 函数, 虽然 MOGSA 算法得到的 GD 指标的均值最小, 获得了收敛效果最好的解, 但是 DFMOGSA 算法得到的 SP 指标的均值明显小于其他对比算法, 即该算法的解具有最优的分布性; 并且 DFMOGSA 算法解的收敛性也要优于 MOPSO、NSGA-II、SPEA2 和 NSGSA 算法; 同时, DFMOGSA 算法得到的非支配解具有最好的分布性。图 2 中的收敛图也表明相比于其他 5 种算法,

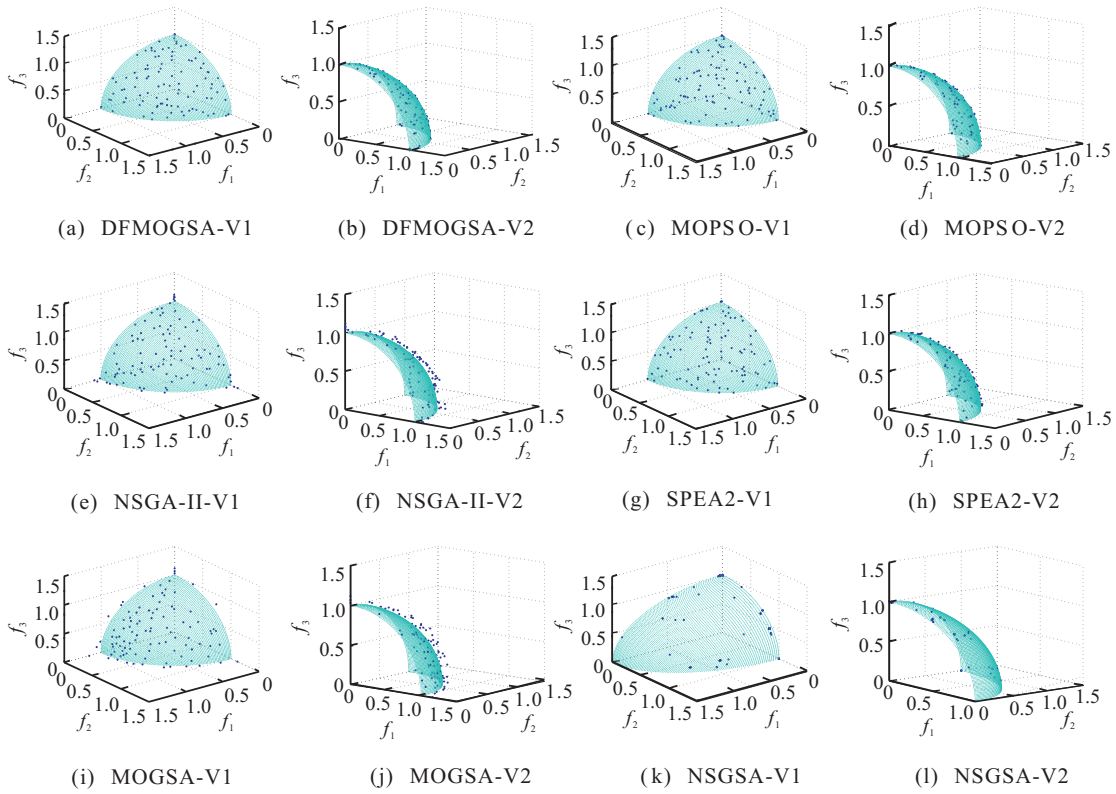


图 5 6种算法对于DTLZ2获得的 Pareto 前端

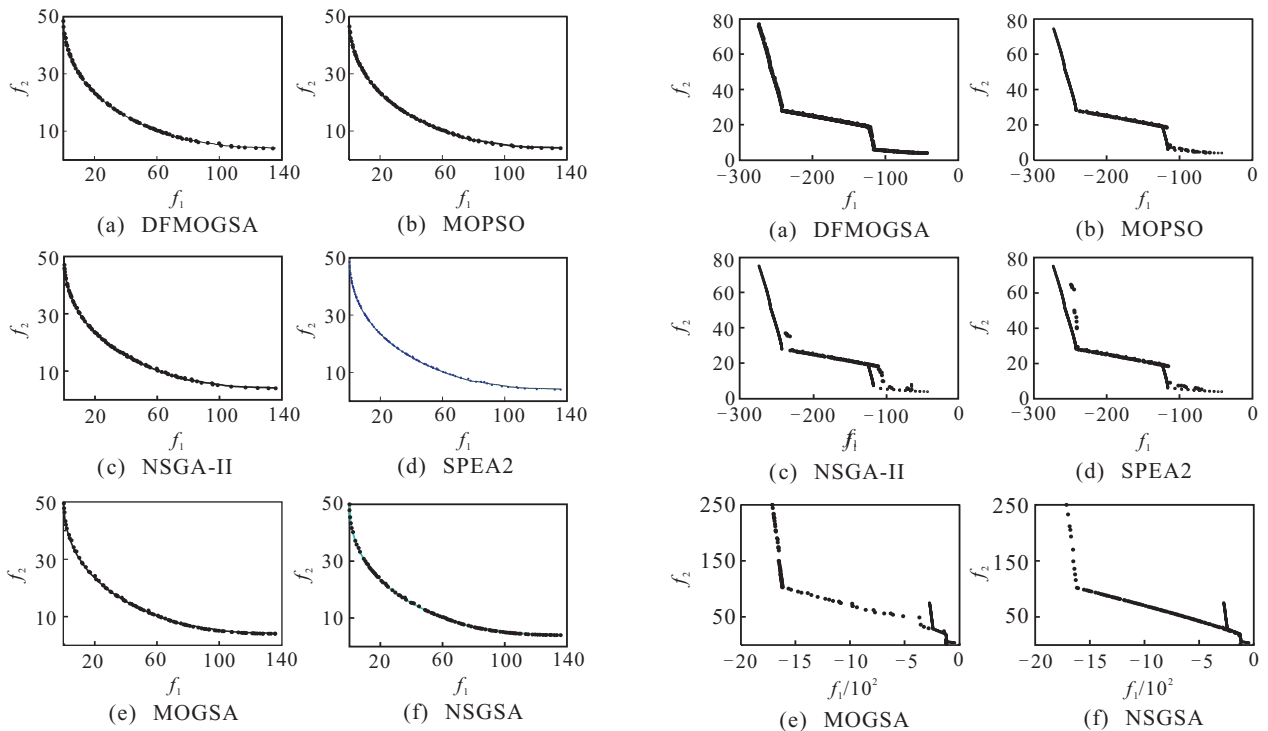


图 6 6种算法对于BNH获得的 Pareto 前端

图 7 6种算法对于OSY获得的 Pareto 前端

表 2 GD 的均值与方差

函数	DFMOGSA		NSGA-II		SPEA2		MOPSO		MOGSA		NSGSA	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
ZDT1	0.002 51	0.001 04	0.019 08	0.000 83	0.015 83	0.007 17	0.002 70	0.001 50	<b>0.000 30</b>	0.000 51	0.009 23	0.001 11
ZDT3	<b>0.000 38</b>	0.000 10	0.000 67	0.000 02	0.001 07	0.000 04	0.010 55	0.005 17	0.000 42	0.000 25	0.002 23	0.000 80
ZDT4	<b>0.000 14</b>	0.000 27	2.012 24	0.966 54	0.006 22	0.999 35	0.011 33	0.987 34	0.072 47	0.013 38	0.105 55	0.036 29
DTLZ2	<b>0.000 17</b>	0.000 13	0.210 19	0.013 05	0.018 64	0.043 37	0.007 95	0.006 27	0.316 00	0.003 74	0.000 69	0.000 12
BNH	<b>0.006 75</b>	0.003 97	0.013 20	0.007 93	0.024 13	0.062 41	0.021 37	0.003 45	0.035 50	0.006 83	0.033 55	0.008 55
OSY	<b>0.044 52</b>	0.060 22	0.655 77	0.802 24	0.689 93	0.214 34	0.908 87	0.650 01	116.381 71	0.023 31	79.418 78	0.002 60

表 3 SP 的均值与方差

函数	DFMOGSA		NSGA-II		SPEA2		MOPSO		MOGSA		NSGSA	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
ZDT1	<b>0.000 94</b>	0.000 51	0.004 10	0.000 61	0.003 14	0.006 17	0.009 37	0.000 43	0.002 23	0.003 82	0.008 61	0.002 07
ZDT3	<b>0.002 99</b>	0.003 95	0.006 56	0.003 17	0.006 02	0.002 07	0.019 04	0.009 43	0.002 01	0.003 57	0.022 34	0.008 77
ZDT4	<b>0.002 42</b>	0.007 32	0.029 34	0.006 04	0.064 75	0.017 57	0.031 09	0.006 17	0.222 60	0.016 61	0.268 87	0.033 62
DTLZ2	<b>0.001 69</b>	0.004 50	0.008 58	0.020 06	0.007 84	0.013 35	0.002 19	0.003 50	0.026 03	0.007 72	0.023 04	0.005 58
BNH	<b>0.002 04</b>	0.001 93	0.053 46	0.063 88	0.012 19	0.034 12	0.081 61	0.012 43	0.026 03	0.006 08	0.068 87	0.007 18
OSY	<b>0.007 14</b>	0.005 52	0.350 14	0.195 56	0.363 12	0.248 70	0.624 71	0.265 76	502.608 16	0.004 98	458.109 10	0.054 64

DFMOGSA 较好地兼顾了收敛性与分布性。

对于 ZDT3 和 ZDT4 函数, DFMOGSA 算法得到的 GD 与 SP 指标的均值最小, 即同时获得了最好的收敛性和分布性. 表 2 和表 3 的数据表明, DFMOGSA 算法是唯一一个能够较理想地解决 ZDT4 函数优化问题的算法. 从收敛图分析, 对于 ZDT3 函数, 如图 3 所示, 虽然不同算法的收敛性与分布性差别不大, 但是整体而言 DFMOGSA 表现最好, MOPSO 在  $f_2$  末端收敛性较差, SPEA2 和 NSGSA 在  $f_1$  末端存在收敛较差的点, 而 NSGA-II 和 MOGSA 的分布性要差于 DFMOGSA. 对于 ZDT4 函数, 如图 4 所示, NSGA-II 与 SPEA2 的收敛效果类似, 都差于 DFMOGSA, 而 MOPSO、MOGSA、NSGSA 都陷入局部 Pareto 解, 效果明显较差.

表 2 和表 3 的数据还表明, 对于三目标的 DTLZ2 函数, DFMOGSA 同样获得了最小的 GD 与 SP 指标的均值, 即表现出最好的收敛性和分布性. 图 5 为两个角度 ( $V_1$  与  $V_2$ ) 的 Pareto 前端分布图, 该图也证实了这一观点, 尤其是 NSGSA 算法, 虽然其获得的解距离真实 Pareto 解集的距离很近, 收敛性较好, 但是该算法获得的解聚集到几个区域, 分布性较差.

对于约束优化问题 BNH 函数, 由于搜索空间维度较低, 函数较为简单, 各种算法得到解的差异性不是非常明显. 如图 6 所示, 不同算法得到的 Pareto 前端差异性较小. 但表 2 和表 3 的数据展现了 DFMOGSA 的优越性. 而对于约束条件较多的 OSY 函数, 无论是定性还是定量的角度, DFMOGSA 算法都表现出了直观、明确的优越性.

此外, 为检验 DFMOGSA 算法与其他 5 种算法相比是否具有显著性的优势, 本文应用秩和检验法<sup>[18]</sup>对 30 次独立运行获得的 GD 与 SP 数据进行非参数检验. 本文设定置信度  $\alpha = 0.05$ . 在原假设下, DFMOGSA 与其他算法所得 GD 与 SP 的秩和检验结果如表 4 和表 5 所示, 如果对应数值小于 0.05, 则表示 DFMOGSA 具有显著优越性. 表 4 和表 5 的算法一栏中的 1 表示 DFMOGSA, 2 表示 NSGA-II, 3 表示 SPEA2, 4 表示 MOPSO, 5 表示 MOGSA, 6 表示 NSGSA.

由表 4 可看出, 除了对应 ZDT1 函数的一个数据

表 4 GD 的秩和检验 ( $\alpha = 0.05$ )

算法	ZDT1	ZDT3	ZDT4	DTLZ2	BNH	OSY
(1,2)	4.13e-3	3.14e-4	1.70e-11	1.70e-11	2.92e-8	2.92e-8
(1,3)	4.13e-3	4.13e-3	5.54e-7	2.92e-8	2.92e-8	2.92e-8
(1,4)	1.25e-2	4.13e-3	5.54e-7	5.54e-7	1.70e-11	2.33e-11
(1,5)	<b>1.04e-1</b>	1.25e-2	3.46e-8	3.46e-8	2.33e-11	3.80e-13
(1,6)	3.14e-4	1.25e-2	2.92e-8	1.01e-2	1.70e-11	3.80e-13

表 5 SP 的秩和检验 ( $\alpha = 0.05$ )

算法	ZDT1	ZDT3	ZDT4	DTLZ2	BNH	OSY
(1,2)	4.13e-3	5.54e-7	2.92e-8	3.46e-8	5.54e-7	1.70e-11
(1,3)	4.13e-3	5.54e-7	1.64e-5	3.46e-8	5.54e-7	1.70e-11
(1,4)	3.56e-8	1.64e-5	1.64e-5	4.55e-7	3.56e-8	3.56e-8
(1,5)	3.14e-4	1.25e-2	1.64e-5	3.14e-4	3.14e-4	3.80e-13
(1,6)	2.92e-8	2.92e-8	3.46e-8	5.54e-7	1.70e-11	3.80e-13

(1.04e-1) 大于 0.05, 其他所有数值均小于 0.05, 所以对于 ZDT1 函数之外所有的测试函数, DFMOGSA 算法的收敛性能都有显著的优越性. 虽然 MOGSA 算法在 ZDT1 函数上收敛性更优, 但表 3 的数据证实其获得解的分布均匀性不够. 对于 ZDT1 函数, 虽然从表 2 的数据可以发现, DFMOGSA 与 MOPSO 得到的 GD 即收敛性差异不大, 但从统计角度分析, DFMOGSA 在收敛性方面具有显著的优越性. 同样, 表 5 中所有数值均小于 0.05, 表明 DFMOGSA 在 SP 上具有明显的优越性, 即该算法能够获得在分布性方面明显优于其他 5 种对比算法的非支配解, 这与表 3 所示的结果具有一致性.

## 4 结 论

本文将数据场理论引入多目标优化领域, 提出一种基于数据场的多目标引力搜索算法 (DFMOGSA). 由于多目标优化算法的性能强烈地依赖引导粒子的质量 (性能), 本文引入数据场理论与非支配排序的方法, 选择两类引导粒子. 第 1 类引导粒子在外部档案中选取, 将外部档案视为数据场, 通过计算非支配解的势能, 判断每个解的所处位置密度, 密度最低的粒子被选为第 1 类引导粒子. 该类直接吸引粒子向低密度区域收敛, 提高解的分布性. 第 2 类引导粒子根据粒子的支配关系从种群中选取, 通过引力引导粒子搜索, 确保算法收敛性. 对比测试结果表明, DFMOGSA 算法优化各测试函数得到的 Pareto 解集, 不论在收敛性还是均匀分布性上, 均具有一定的优越性. 目前算

法处理约束问题的方法比较简单,后续工作可以借鉴单目标优化中处理约束问题的方法,提出解决多目标约束优化问题的新方法,并将其应用于工程设计中,以实现其社会价值。

### 参考文献(References)

- [1] 徐志丹,莫宏伟.多目标扰动生物地理学优化算法[J].控制与决策,2014,29(2):231-235.  
(Xu Z D, Mo H W. Multi-objective disturbance biogeography-based optimization algorithm[J]. Control and Decision, 2014, 29(2): 231-235.)
- [2] Coello C A C, Lechuga M S. MOPSO: A proposal for multiple objective particle swarm optimization[C]. Proc of the 2002 Congress on Evolutionary Computation. Honolulu: IEEE, 2002, 2: 1051-1056.
- [3] 王艳,曾建潮.一种基于拟态物理学优化的多目标优化算法[J].控制与决策,2010,25(7):1040-1044.  
(Wang Y, Zeng J C. Multi-objective Optimization Algorithm based on Artificial Physics Optimization[J]. Control and Decision, 2010, 25(7): 1040-1044.)
- [4] Coello C A C, Pulido G T, Lechuga M S. Handling multiple objectives with particle swarm optimization[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 256-279.
- [5] 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.
- [6] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength pareto evolutionary algorithm[C]. Evolutionary Methods for Design, Optimization and Control with Application to Industrial Problems(EUROGEN 2001). Athens: International Center for Numerical Methods in Engineering(CIMNE), 2002: 95-100.
- [7] Fonseca C M, Fleming P J. Genetic algorithms for multiobjective optimization: Formulation discussion and generalization[C]. Proc of the 5th Int Conf on Genetic Algorithm. San Matco: Morgan Kaufman Publishers, 1993, 93: 416-423.
- [8] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: A gravitational search algorithm[J]. Information Sciences 2009, 179 (13): 2232-2248.
- [9] Liu Y, MA L. Gravitational search algorithm for location problem[J]. Computer Engineering and Applications, 2012, 48(27): 42-44.
- [10] Rashedi E, Nezamabadi-Pour H, Saryazdi S. Filter modeling using gravitational search algorithm[J]. Engineering Applications of Artificial Intelligence, 2011, 24(1): 117-122.
- [11] Jiang S, Ji Z, Shen Y. A Novel hybrid particle swarm optimization and gravitational search algorithm for solving economic emission load dispatch problems with various practical constraints[J]. Electrical Power and Energy Systems, 2014, 55: 628-644.
- [12] Hassanzadeh H R, Rouhani M. A multi-objective gravitational search algorithm[C]. 2010 2nd Int Conf on Computational Intelligence, Communication Systems and Networks(CICSyN). Liverpool: IEEE, 2010: 7-12.
- [13] Nobahari H, Nikusokhan M, Siarry P. A multi-objective gravitational search algorithm based on non-dominated sorting[J]. Int J of Swarm Intelligence Research(IJSIR), 2012, 3(3): 32-49.
- [14] 李德仁,王树良,李德毅.空间数据挖掘理论与应用[M].北京:科学出版社,2006:364-385.  
(Li D R, Wang S L, Li D Y. Spatial data mining theories and application[M]. Beijing: Science Press, 2006: 364-385.)
- [15] 王海军,邓羽,王丽,等.基于数据场C均值聚类方法研究[J].武汉大学学报:信息科学版,2009,34(5):626-629.  
(Wang H J, Deng Y, Wang L, et al. A C-means algorithm based on data field[J]. Geomatics and Information Science of Wuhan University, 2009, 34(5): 626-629.)
- [16] Jiao L C, Wang H, Shang R H, et al. A co-evolutionary multi-objective optimization algorithm based on direction vectors[J]. Information Sciences, 2013, 228: 90-112.
- [17] Kaveh A, Laknejadi K. A novel hybrid charge system search and particle swarm optimization method for multi-objective optimization[J]. Expert Systems with Applications, 2011, 38(12): 15475-15488.
- [18] Garc'ia S, Molina D Lozano M., et al. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC2005 special session on real parameter optimization[J]. J of Heuristics, 2009, 15(6): 617-644.

(责任编辑:孙艺红)