

## ◎理论与研发◎

## 动态邻居维度学习的多目标粒子群算法

肖闪丽, 王宇嘉, 聂善坤

XIAO Shanli, WANG Yujia, NIE Shankun

上海工程技术大学 电子电气工程学院, 上海 201620

School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China

XIAO Shanli, WANG Yujia, NIE Shankun. Multi-objective particle swarm optimization based on dynamic neighborhood for dimensional learning. *Computer Engineering and Applications*, 2017, 53(20):31-37.

**Abstract:** Focus on the poor behavior of the diversity for multi-objective particle swarm optimization and the selection pressure of population increasing with the variable dimension, a Multi-Objective Particle Swarm Optimization based on Dynamic Neighborhood of Dimensional Learning (DNDL-MOPSO) is proposed. Firstly, an optimum dimensional individual is established. Then based on the individual and social knowledge, the proposed algorithm improves the formula of the velocity updating and uses a strategy that each dimensional learning object is not fixed. Finally, the random guide learning strategy is used to alleviate the selection pressure. The experimental results indicate that the new algorithm can improve the global convergence and increase the diversity of population. It is effective to solve the benchmark multimodal optimization problems.

**Key words:** Particle Swarm Optimization(PSO); multi-objective optimization; dynamic neighbor; optimum dimensional individual; random guide learning

**摘 要:**针对多目标粒子群算法多样性较差,种群选择压力随着变量维度增加的问题,提出了基于动态邻居维度学习的多目标粒子群算法(DNDL-MOPSO)。该算法首先构建最优维度个体,然后在“个体认知”和“社会认知”的基础上,对粒子速度更新公式进行改进,采用每一维上学习对象不固定的交流方式,最后利用随机向导学习策略,增加种群多样性。实验结果表明该方法能够提高算法的全局收敛性,增加种群的多样性,缓解选择压力,有效解决多峰多目标优化问题。

**关键词:**粒子群算法;多目标优化;动态邻居;最优维度粒子;随机向导学习

**文献标志码:**A **中图分类号:**TP202 **doi:**10.3778/j.issn.1002-8331.1606-0137

## 1 引言

粒子群优化算法<sup>[1]</sup>(Particle Swarm Optimization, PSO)的思想起源于对鸟群捕食行为的研究,是一种基于群体智能的优化方法。PSO算法的特点是收敛速度快,易于实现,计算效率高<sup>[2]</sup>。由于PSO算法在单目标优化问题上的出色表现<sup>[3]</sup>,大量学者将其扩展用于解决多目标优化<sup>[4-6]</sup>,但是随着优化问题复杂程度的增加,种

群的选择压力增大,多样性降低,导致算法陷入局部最优。

文献[7]提出了一种混合PSO算法,采用了基于模拟退火加权的方式进行内部搜索,防止过早收敛,但该方法以降低收敛性为代价,难以搜索到整个平衡面。文献[8]给出了拟小世界动态拓扑邻域结构来平衡粒子的全局搜索能力和局部搜索能力,这种方法的局限性在于

**基金项目:**国家自然科学基金(No.61403249);上海工程技术大学研究生科研创新项目(No.E309031601178)。

**作者简介:**肖闪丽(1990—),女,硕士研究生,研究领域为粒子群优化算法,E-mail:shanli0681@163.com;王宇嘉(1979—),通讯作者,女,博士,副教授,研究领域为多目标优化、群体智能算法;聂善坤(1991—),男,硕士研究生,研究领域为多目标粒子群优化、神经网络。

**收稿日期:**2016-06-12 **修回日期:**2016-08-01 **文章编号:**1002-8331(2017)20-0031-07

**CNKI网络优先出版:**2016-12-07, <http://www.cnki.net/kcms/detail/11.2127.TP.20161219.1613.002.html>

只对区域里的粒子搜索能力强,对于不在粒子包围的邻域范围作用很小。Alvarez-Benitez等<sup>[9]</sup>提出了三种全局最优位置的选取方法,通过赋予粒子优先权增加其被选为全局最优位置的概率,这种方法存在陷入局部最优的风险性。文献[10]提出了一种新的基于矢量演化的多目标粒子群算法(VE-MOPSO),该方法引进子种群的思想,通过子群间的相互合作,搜索Pareto最优解,但由于粒子数目增加,加大了计算量。文献[11]提出一种自适应网络和拥挤距离的多目标粒子群算法,通过密度信息和拥挤距离来确定学习样本,但该算法对含有多个局部最优解的函数易出现早熟收敛现象。Tripathi等<sup>[12]</sup>提出了一种惯性权重和加速度系数随迭代次数变化的多目标粒子群优化方法,算法在多样性上有所改进,但在收敛速度和精度方面相比于 $\epsilon$ DMOPSO算法较差<sup>[13]</sup>。

目前,对社会学习样本的选择大多是基于邻域中运行最好的一个个体,没有充分利用相邻粒子的信息<sup>[14]</sup>,而个体间学习过程的交流方式将直接影响粒子获取全局信息的能力。为了充分利用邻居信息,提高种群的多样性,缓解种群的选择压力,本文提出了动态邻居维度学习的多目标粒子群算法(Multi-Objective Particle Swarm Optimization based on Dynamic Neighborhood for Dimensional Learning, DNDL-MOPSO),该方法采用动态拓扑结构,在每个粒子“个体认知”和“社会认知”基础上,引入动态邻居维度学习,通过每一维学习对象不固定的交流方式增加种群的多样性,提高算法的全局收敛性。实验结果表明,DNDL-MOPSO算法在保证收敛的前提下,能有效提高种群的多样性,取得令人满意的效果。

## 2 粒子群算法

PSO算法将个体看作 $D$ 维搜索空间中一个理想情形的粒子。假设群体规模为 $N$ ,第 $i(i=1, 2, \dots, D)$ 个粒子在搜索空间中的飞行速度可以表示为 $v_i(v_{i1}, v_{i2}, \dots, v_{iD})$ ,它的位置可以表示为 $x_i(x_{i1}, x_{i2}, \dots, x_{iD})$ 。当粒子迭代到第 $t+1$ 代时,全局版本PSO算法第 $i$ 个粒子按式(1)更新自身的速度与位置,局部版本PSO算法按式(2)进行更新<sup>[4]</sup>:

$$\begin{cases} v_{ij}(t+1) = \omega v_{ij}(t) + c_1 r_1 (pbest_{ij}(t) - x_{ij}(t)) + \\ c_2 r_2 (gbest_j(t) - x_{ij}(t)) \\ x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \end{cases} \quad (1)$$

$$\begin{cases} v_{ij}(t+1) = \omega v_{ij}(t) + c_1 r_1 (pbest_{ij}(t) - x_{ij}(t)) + \\ c_2 r_2 (pneighbor_{ij}(t) - x_{ij}(t)) \\ x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \end{cases} \quad (2)$$

其中, $j=1, 2, \dots, D$ 表示粒子在目标空间上的维度; $pbest_{ij}$ 为粒子 $i$ 的第 $j$ 维历史最优位置; $gbest_j$ 为第 $j$ 维的全局最优位置; $pneighbor_{ij}$ 为粒子 $i$ 的第 $j$ 维邻居

最优位置; $\omega$ 为粒子速度的更新权重; $c_1$ 和 $c_2$ 为学习因子; $r_1$ 和 $r_2$ 为随机数。从式(1)和(2)中可以看出,粒子的速度更新可以看成三部分:一是粒子先前的速度;二是粒子的“个体认知”,表示粒子对个体自身认知活动的认识,即粒子对其自身认知活动信息的再次加工过程;三是“社会认知”,表示粒子间的信息共享机制及相互间的协作,其作用增强了群体的全局搜索能力,使收敛性能得到大幅提升。

## 3 动态邻居维度学习的多目标粒子群算法(DNDL-MOPSO)

PSO已经在多目标优化问题上广泛应用,但是随着变量维度的增加,种群的选择压力增大,降低了种群的多样性。为了提高种群的多样性,DNDL-MOPSO引入动态邻居最优维度学习,通过随机向导学习策略缓解选择压力,在粒子更新过程中,采取每一维学习对象不固定策略,增加种群多样性,具体介绍如下。

### 3.1 动态邻居拓扑结构

不同的拓扑结构决定了粒子学习样本的不同。静态拓扑结构会导致种群陷入局部最优的概率增加,因此Suganthan等提出了将环形结构与星形结构结合起来的动态邻居构建策略<sup>[15]</sup>。在粒子邻居数增长模型中,当任意两个粒子满足式(3)时,认定上述两粒子为邻居关系:

$$\frac{\|x_i - x_{bin}\|}{d_{\max}} < \epsilon, \epsilon = \frac{3t + 0.6t_{\max}}{t_{\max}} \quad (3)$$

其中, $x_i$ 和 $x_{bin}$ 为种群中任意两粒子; $t$ 为算法当前迭代次数, $t_{\max}$ 为算法最大迭代次数; $d_{\max}$ 是任意两个粒子间的最大距离。这样使算法在初始迭代时有较好的探索性能,而在迭代后期有较好的开发性能。

### 3.2 构建最优维度粒子

传统的PSO算法中,粒子 $i$ 的每一维都是向同一个粒子相应维度学习,其学习样本选择如图1所示。

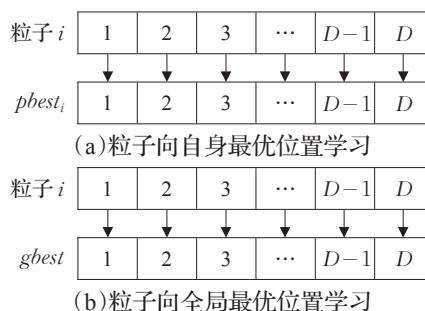
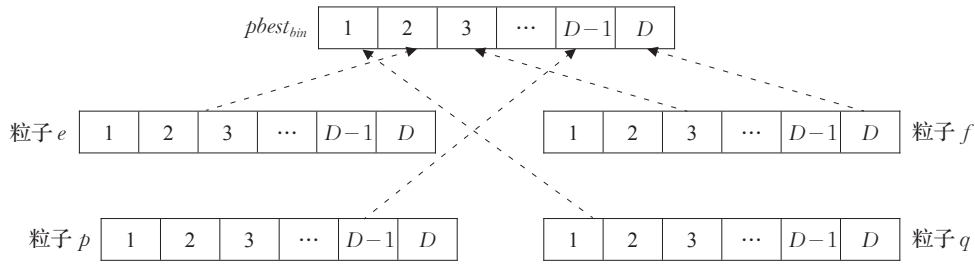


图1 传统PSO学习样本选取方法

这种学习策略会使粒子的一部分维数向最优位置逼近,而另一部分维数可能远离最优位置,从而导致粒子远离全局最优解,陷入局部最优<sup>[16]</sup>。为平衡学习过程中的探索性能和开发性能<sup>[17]</sup>,在DNDL-MOPSO算法中

图2 选取粒子*i*邻居中最优维度组成  $pbest_{bin}$  的方法

选取每个粒子邻居中最优维度构成新的学习个体  $pbest_{bin}$ ,如图2所示。

在图2中,粒子  $e$ 、 $f$ 、 $p$  和  $q$  分别代表粒子  $i$  邻居的位置。粒子  $i$  的每一维学习对象根据式(4)选取。

$$p_{bin(i)}^d = \arg \left\{ \frac{\sum(\text{fitness}(\text{bin}_j) - \text{fitness}(pbest_i))}{|bin_{j,d} - x_{i,d}|} \right\} \quad (4)$$

其中,  $p_{bin(i)}^d$  表示粒子  $i$  的第  $d$  维学习样本,  $neighbor_i$  表示粒子  $i$  的所有邻居;  $bin_j$  代表粒子  $i$  的第  $j$  个邻居,  $pbest_i$  表示粒子  $i$  对应的历史最优位置;  $\arg\{\}$  表示找出对应粒子的位置;  $N$  代表种群的个数,  $D$  为粒子的维度。公式(4)说明粒子与它所有邻居的相应维度比较距离,距离最小的将会作为粒子相应维度的学习样本。

### 3.3 学习样本的选取和随机向导学习策略

#### 3.3.1 全局最优位置的选取

在DNDL-MOPSO处理多目标优化问题时,使用外部档案文件存储非劣解,档案文件的维护和更新参考文献[18]。粒子的全局最优解根据密集距离<sup>[19-20]</sup>进行选择,密集距离计算公式如式(5)所示,选取较大的值作为全局最优粒子。

$$distance_i = distance_i + \frac{\text{fitness}_{i+1,m} - \text{fitness}_{i-1,m}}{\text{fitness}_m^{\max} - \text{fitness}_m^{\min}} \quad (5)$$

其中,  $distance_i$  为粒子  $i$  的密集距离,  $m$  代表第  $m$  个目标函数,  $\text{fitness}_{i+1,m}$  和  $\text{fitness}_{i-1,m}$  分别代表第  $i+1$  和第  $i-1$  个粒子在第  $m$  个目标函数上的值,  $\text{fitness}_m^{\max}$ 、 $\text{fitness}_m^{\min}$  为第  $m$  个目标函数的最大值和最小值。

#### 3.3.2 个体最优位置的更新

为充分利用每个粒子的  $pbest$  信息,提出一种随机向导学习策略更新  $pbest$ ,即当一个粒子连续  $T$  代个体最优位置没有得到提升时,随机产生若干个粒子,与  $pbest$  进行支配性比较,在支配的粒子中随机选择一个粒子替代  $pbest$ ;当条件不满足时,按支配性对  $pbest$  进行更新。  $pbest$  的更新过程伪代码如下所示。

1. while the termination criterion ( $count \leq T$ ) is not satisfied
2. do compare  $x_i$  and  $pbest_i$  dominate
3. if  $pbest_i$  dominates  $x_i$
4.  $count \leftarrow count + 1$

5. else if  $x_i$  dominates  $pbest_i$
6.  $pbest_i = x_i$
7. else if  $pbest_i$  and  $x_i$  is equal
8. if  $\text{rand}() < 0.5$
9.  $pbest_i = x_i$
10. else if  $count \leftarrow count + 1$
11. end if
12. end while
13. if the termination criterion ( $count > T$ ) is satisfied
14. for  $k \leftarrow 1$  to  $num$
15. update particle is in terms of (1)
16. compare  $x_i$  and  $newx(k,:)$  dominate
17. if  $newx(k,:)$  dominates  $x_i$
18.  $renewx \leftarrow newx(k,:)$
19. end for
20. end if
21.  $mx \leftarrow \text{length}(renewx)$
22.  $randx \leftarrow$  randomly select an number between 1 to  $mx$
23.  $pbest_i = renewx(randx,:)$

### 3.4 基于动态邻居维度学习的粒子速度更新

从公式(1)、(2)可以看出,PSO算法的更新主要依靠  $pbest$ 、 $gbest$  或  $pneighbor$  的位置,而粒子的每一维只从固定的学习样本进行学习,这样的更新策略存在一定的局限性,导致种群陷入局部最优。为提高种群的多样性,按如下方法对粒子进行更新:

步骤1 随机选出粒子  $i$  的  $m(j \in m)$  个维数,根据式(6a)使粒子  $i$  向  $gbest$  学习:

$$v_{i,j} = \omega v_{i,j} + \text{rand}() (gbest_j - x_{i,j}) \quad (6a)$$

步骤2 再随机选出粒子  $i$  的  $n(j \in n)$  个维数,根据式(6b)使粒子  $i$  向邻居中最优维度组成的新个体  $pbest_{bin}$  学习:

$$v_{i,j} = \omega v_{i,j} + \text{rand}() (pbest_{bin[j]} - x_{i,j}) \quad (6b)$$

步骤3 剩余维数  $D - m - n(j \in D - m - n)$  根据式(6c)使粒子  $i$  向自身的  $pbest$  学习:

$$v_{i,j} = \omega v_{i,j} + \text{rand}() (pbest_{i,j} - x_{i,j}) \quad (6c)$$

其中,  $n$  和  $m$  分别由学习概率  $p_c$  和精英概率  $p_m$  决定,  $p_c$  决定了粒子的某一维向  $pbest$  或  $pbest_{bin}$  的学习能力,  $p_m$  决定了粒子的某一维向  $gbest$  的学习能力。粒子速度更新过程如图3所示,其伪代码如下所示。



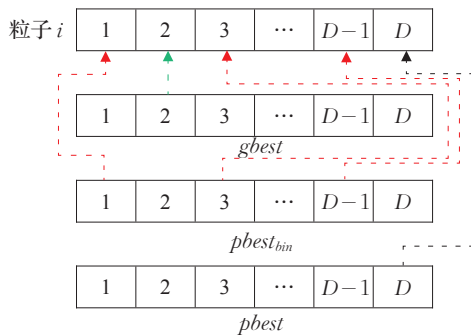


图3 每一维学习对象不固定交流的更新过程

```

1. Initialize particles' position, velocity,  $pbest$ ,  $gbest$ ,  $p_m$ ,
 $p_c$ , Number of particles  $\leftarrow N$ 
2. while the termination criterion is not satisfied do
3. Every ten generations to update  $a_i$ ,  $b_i$  and  $bin_i$ 
4.  $rc \leftarrow$  generate a random number 1 to  $D$ 
5. Calculate  $a_i(rc(1:m)) \leftarrow 1$ ,  $m \leftarrow (N \times p_m)$ ,
 $b_i = \lfloor rand(1, D) - 1 + p_c \rfloor$ ,  $bin_i = \lfloor rand(1, D) \times N \rfloor \leftarrow 1$ 
6. for  $i \leftarrow 1$  to  $N$  do
7.   for  $j \leftarrow 1$  to  $D$  do
8.     if  $a_i[d] \leftarrow 1$ 
9.        $v_{i,j} \leftarrow \omega v_{i,j} + rand() \cdot (gbest_j - x_{i,j})$ 
10.    else if  $b_i[d] \leftarrow 1$ 
11.       $v_{i,j} \leftarrow \omega v_{i,j} + rand() \cdot (pbest_{bin_i[j]} - x_{i,j})$ 
12.    else
13.       $v_{i,j} \leftarrow \omega v_{i,j} + rand() \cdot (pbest_{i,j} - x_{i,j})$ 
14.    end if
15.     $x_{id} \leftarrow x_{id} + v_{id}$ 
16.  end for
17. end for
18. end while

```

### 3.5 DNDL-MOPSO 算法流程

**步骤1** 算法初始化, 包括: 种群规模  $N$ , 外部档案规模  $NP$ , 粒子位置  $x$  和速度  $v$ , 粒子维数  $D$ , 惯性权重  $\omega$ , 学习因子  $c_1, c_2$ , 粒子全局最优位置  $gbest$ , 个体最优历史位置  $pbest$ , 粒子最优维度个体  $pbest_{bin}$ , 学习概率  $p_c$ , 精英概率  $p_m$ , 随机向导学习持续代数阈值  $T$ , 最大迭代次数  $t_{max}$ , 目标向量  $f_m$ 。

**步骤2** 初始化外部档案  $NP$  中的粒子。

**步骤3** 从外部档案中随机选取两个粒子, 根据式(5)计算密集距离确定粒子的全局最优位置  $gbest$ 。

**步骤4** 根据公式(3)确定粒子的邻居个数, 利用公式(4)确定粒子每一维学习样本。

**步骤5** 采用随机向导学习策略更新粒子个体最优历史位置  $pbest$ 。

**步骤6** 根据公式(6a)、(6b)、(6c)更新粒子的速度  $v$ 。

**步骤7** 对外部档案文件  $NP$  进行更新和维护。

**步骤8** 如果终止条件成立, 则停止搜索, 输出非劣解和非劣解所对应的目标值, 否则转到步骤3。

## 4 实验结果与分析

### 4.1 性能指标与测试函数

为了比较不同算法的性能, 采用如下三种指标评价算法性能, 即: 代距 ( $GD$ )、间距 ( $SP$ ) 和最大覆盖度 ( $MS$ )<sup>[21]</sup>, 性能指标的定义分别如式(7)、(8)、(9)所示, 其中  $GD$  用来描述算法所获得的非劣解与真实的 Pareto 前沿的距离,  $SP$  描述解向量的分布,  $MS$  描述非劣解在 Pareto 前沿上的覆盖度。

$$GD = \frac{\sqrt{\sum_{i=1}^n dist_i^2}}{n} \quad (7)$$

$$SP = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\overline{dist} - dist_i)^2}, \quad \overline{dist} = \frac{1}{n} \sum_{i=1}^n dist_i \quad (8)$$

$$MS = \sqrt{\frac{1}{M} \sum_{k=1}^m \left\{ \frac{\min(f_k^{\max}, F_k^{\max}) - \max(f_k^{\max}, F_k^{\max})}{F_k^{\max} - f_k^{\max}} \right\}} \quad (9)$$

本文选取 ZDT 系列测试函数和 DTLZ 系列测试函数中的 DTLZ1、DTLZ2 和 DTLZ7 验证算法性能, 其参数及特性如表1所示。

### 4.2 参数设置

在 DNDL-MOPSO 算法中, 设置种群规模和外部档案规模均为 100, 最大迭代次数为 200, 随机向导学习持续代数阈值  $T$  为 7,  $p_c$  为 0.1,  $p_m$  为 0.4。

测试函数参数设置如下: ZDT1-ZDT3 的决策变量取 30 维, ZDT4 和 ZDT6 取 10 维, 所有的 ZDT 测试函数包含两个目标函数; DTLZ1、DTLZ2 和 DTLZ7 包含三个目标函数, 决策变量取 12 维。为了验证 DNDL-MOPSO

表1 测试函数特性

测试函数	变量范围	特性
ZDT1	$D=30, 0 \leq x_i \leq 1$	Pareto 曲面连续, 凸
ZDT2	$D=30, 0 \leq x_i \leq 1$	Pareto 曲面连续, 非凸
ZDT3	$D=30, 0 \leq x_i \leq 1$	Pareto 曲面由 5 段不连续凸曲面组成
ZDT4	$D=10, 0 \leq x_i \leq 1$	多峰函数, 包含 $21^9$ 个局部极值
ZDT6	$D=10, 0 \leq x_i \leq 1$	Pareto 最优解沿着 Pareto 前沿非均匀分布
DTLZ1	$D=12, 0 \leq x_i \leq 1$	具有线性 Pareto 最优边界的 $M$ 个目标测试问题, 包含多个局部极值
DTLZ2	$D=12, 0 \leq x_i \leq 1$	Pareto 前沿面是第一象限内的单位球面
DTLZ7	$D=12, 0 \leq x_i \leq 1$	具有多个不连续的 Pareto 前沿面

算法的有效性,分别与NSGA2<sup>[22]</sup>、ACG-MOPSO<sup>[11]</sup>和MOCLPSO<sup>[17]</sup>算法进行比较,四种算法的相关参数如表2所示。

表2 算法相关参数设置

主要参数	$c_1, c_2$	$w$	种群规模	存档规模
NSGA2	—	—	100	100
ACG-MOPSO	$c_1 = c_2 = 1.491$	0.729	100	100
MOCLPSO	$c_1 = c_2 = 2$	随着迭代次数从0.9线性递减到0.4	100	100
DNDL-MOPSO	$c_1 = c_2 = 2$	随着迭代次数从0.9线性递减到0.4	100	100

4.3 实验结果与分析

考虑随机初始化的影响,对于每一个测试函数,所有算法全部运行30次。DNDL-MOPSO和其他三种算法得到的测试函数Pareto前沿如图4~图7所示。从图

中可以看出,DNDL-MOPSO算法在所有测试函数上均收敛到了真实的Pareto前沿,说明本文所提出的算法具有较好的收敛性和多样性。

对于测试函数ZDT,通过图4(a)~(e)可以看出NSGA2算法在分布性和收敛性方面均达不到满意的效果;ACG-MOPSO是一种基于自适应网络和拥挤距离的多目标粒子群算法,从图4(a)~(c)中可以看出,ACG-MOPSO均收敛到了真实的Pareto前沿,而在图4(d)~(e)中,ACG-MOPSO算法的收敛性劣于其他三种算法,由此得出该算法在解决最优前沿是凸、非凸和不连续优化问题上具有良好的收敛性能和多样性,而在处理拥有多个局部极值的ZDT4和Pareto前沿非均匀分布的ZDT6测试函数上,相比于其他三种算法是较差的,其原因是ACG-MOPSO算法不能够描述个体间的相邻位置关系,导致多样性下降,无法跳出局部最优,从而出现早熟收敛现象。从图4(a)~(e)中可以看出,MOCLPSO均

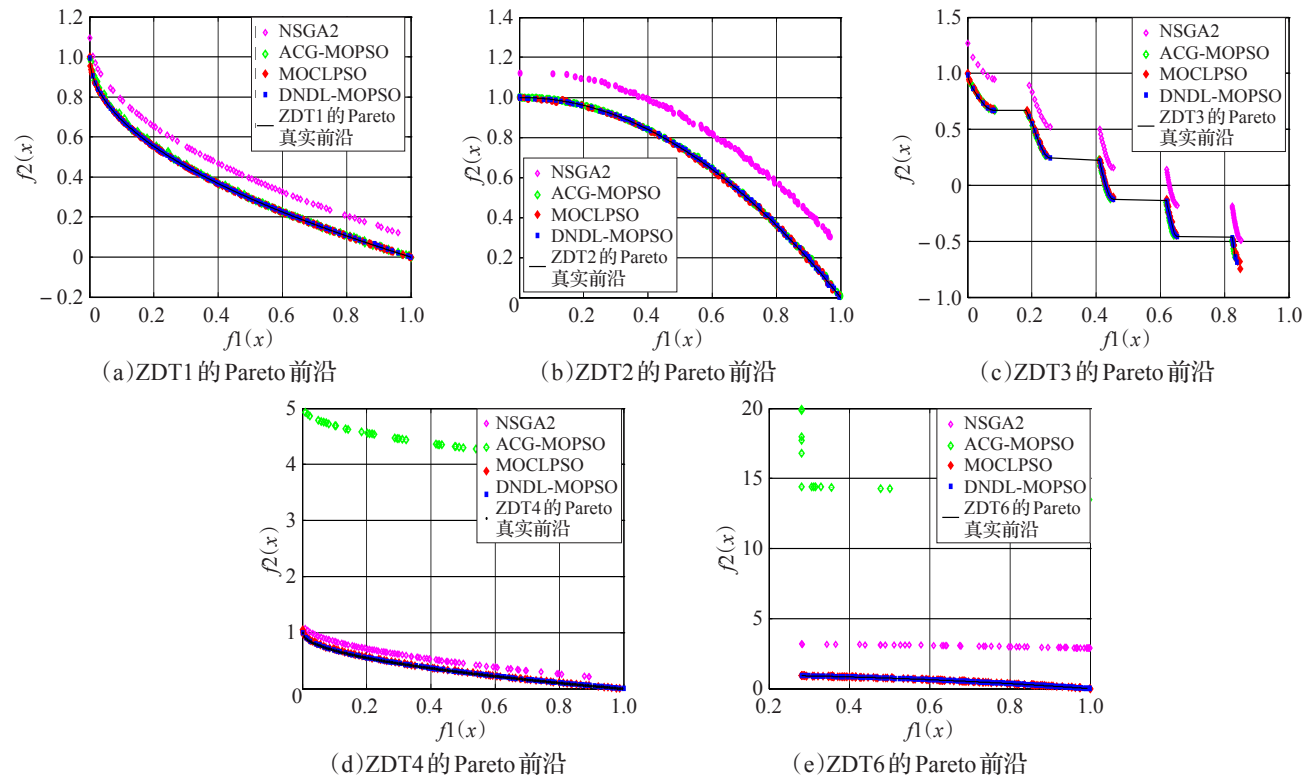


图4 ZDT1、ZDT2、ZDT3、ZDT4和ZDT6在几种算法上的Pareto前沿

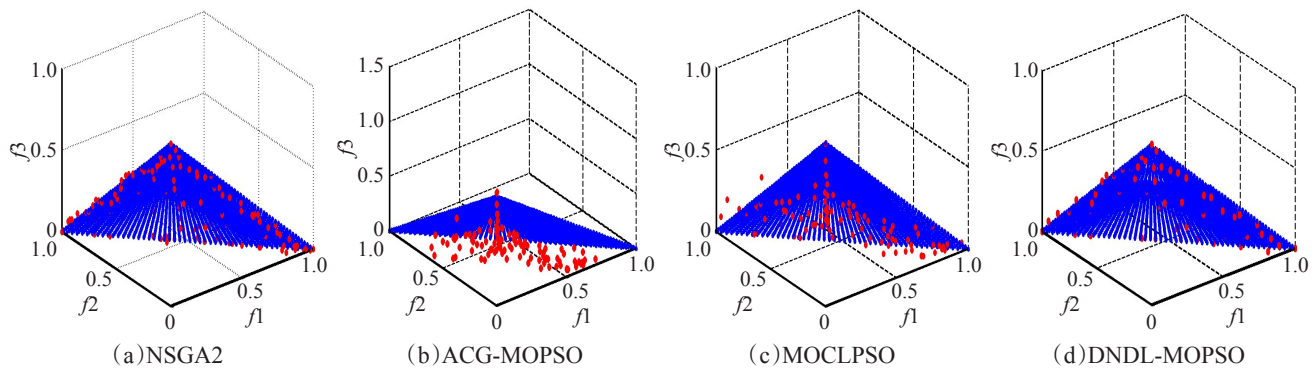


图5 四种算法得到DTLZ1的Pareto前沿

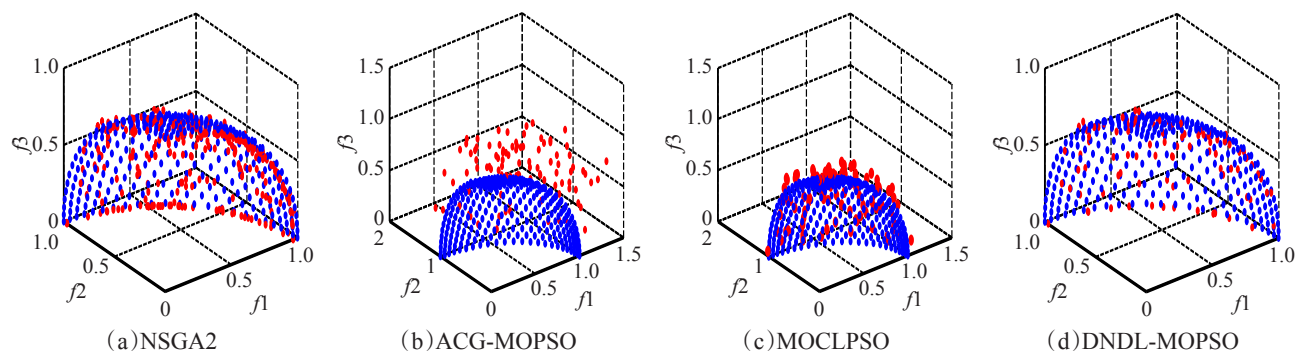


图6 四种算法得到DTLZ2的Pareto前沿

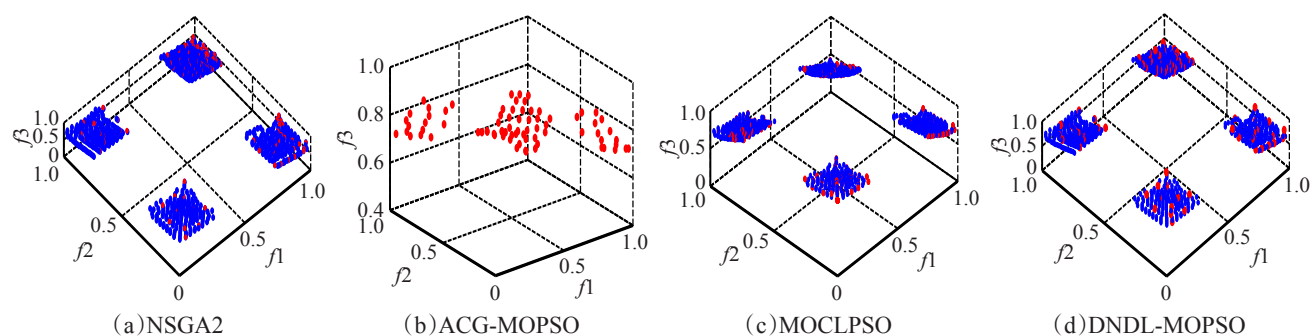


图7 四种算法得到DTLZ7的Pareto前沿

能收敛到真实的 Pareto 前沿,说明了该算法具有跳出局部最优的能力。

对于测试函数 DTLZ,从图 5(a)(d)和图 6(a)(d)中可以发现,NSGA2 和 DNDL-MOPSO 具有很好的收敛性和多样性,但是 DNDL-MOPSO 产生的非劣解分布比较均匀;从图 5(c)(d)~图 7(c)(d)中可以发现,相比于 DNDL-MOPSO,MOCLPSO 的收敛性较差,无法收敛到真实的 Pareto 前沿,而 ACG-MOPSO 只有很少的一部分收敛到真实的 Pareto 前沿,尤其是对于测试函数 DTLZ7 来说,ACG-MOPSO 算法无法得到 Pareto 前沿,如图 7(b)所示;通过图 7(a)(d)可以发现, DNDL-MOPSO 在 DTLZ7 上收敛性和多样性方面均优于 NSGA2,并且具有较好的分布均匀性。

所有算法的性能指标  $GD$ 、 $SP$  和  $MS$  的平均值如表 3 所示。从表 3 中可以看出, DNDL-MOPSO 算法的  $GD$ 、 $SP$  和  $MS$  指标值均优于其他三种算法,说明 DNDL-MOPSO 在总体上具有较好的收敛性和多样性。

从图 4(a)~(e)中可以看出,MOCLPSO 均能收敛到 Pareto 前沿,但是从表 3 中的  $GD$  数值上发现,MOCLPSO 的值均大于 ACG-MOPSO 的值;对于  $SP$  的值,MOCLPSO 均比 ACG-MOPSO 的值小,说明 MOCLPSO 相对于 ACG-MOPSO 在多样性方面占优,而在收敛性方面没有优势。相比于以上两种算法, DNDL-MOPSO 的收敛性指标  $GD$  优于 ACG-MOPSO,  $SP$  指标优于 MOCLPSO。

对于含有多个局部极值的 ZDT4 和 DTLZ1 这两个测试函数来说, DNDL-MOPSO 算法的  $SP$  指标值最小,说明 DNDL-MOPSO 有效地提高了种群的多样性,使算

法跳出局部最优。DNDL-MOPSO 和其他三个对比算法在测试函数 ZDT4 和 DTLZ1 上的收敛性随迭代次数变化曲线图如图 8 所示,从图中可以看出,对于测试函数 ZDT4 和 DTLZ1 来说, DNDL-MOPSO 的收敛性最好,MOCLPSO 在 DTLZ1 上的收敛性比 NSGA2 较差,再一次说明了 MOCLPSO 算法在收敛性上不占优势,而 DNDL-MOPSO 在增加种群多样性的同时,也保证了算法的收敛性。

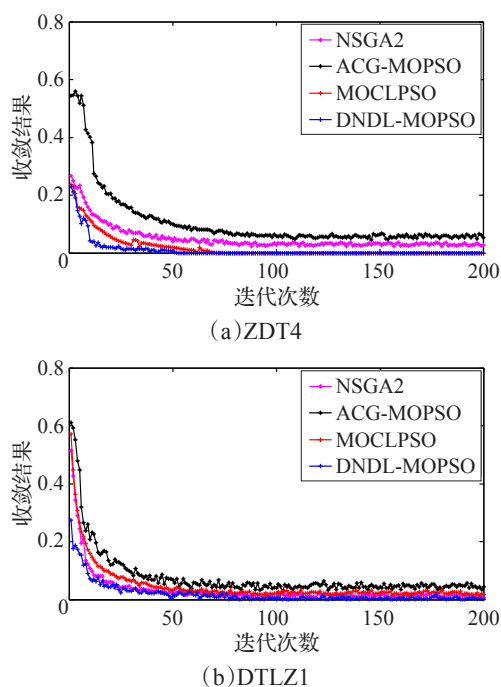


图8 测试函数ZDT4和DTLZ1收敛性随迭代次数变化曲线

表3 四种算法在30次测试中关于性能指标GD,SP,MS的平均值

测试函数	指标	NSGA2	ACG-MOPSO	MOCLPSO	DNDL-MOPSO
ZDT1	GD	0.010 1	4.142 3E-04	5.174 5E-04	1.354 2E-04
	SP	0.034 0	0.009 1	0.005 7	0.004 3
	MS	0.959 1	0.998 1	0.997 8	1.000 0
ZDT2	GD	0.024 0	3.480 8E-04	3.801 0E-04	2.413 0E-04
	SP	0.031 1	0.018 1	0.006 5	0.004 7
	MS	0.911 9	0.997 9	0.994 4	0.998 8
ZDT3	GD	0.035 0	0.005 7	0.006 1	0.004 6
	SP	0.037 2	0.024 7	0.008 1	0.007 1
	MS	0.977 2	0.985 1	0.984 2	1.000 0
ZDT4	GD	0.023 0	0.563 5	0.002 7	2.600 3E-04
	SP	0.083 4	0.099 1	0.008 7	0.003 9
	MS	0.918 1	—	0.997 8	1.000 0
ZDT6	GD	0.414 6	5.135 6	7.019 4E-04	1.309 0E-04
	SP	0.012 4	0.399 1	0.009 1	0.004 6
	MS	—	—	1.000 0	1.000 0
DTLZ1	GD	0.054 2	0.295 2	0.073 2	0.051 7
	SP	0.541 9	0.901 2	0.513 5	0.451 5
	MS	0.903 1	0.324 1	0.890 1	0.928 6
DTLZ2	GD	0.068 1	0.345 2	0.289 7	0.052 0
	SP	0.534 8	0.898 2	0.720 2	0.412 6
	MS	0.925 1	0.358 6	0.646 2	0.936 7
DTLZ7	GD	0.092 2	0.516 2	0.089 8	0.041 3
	SP	0.627 6	—	0.593 1	0.384 1
	MS	0.727 7	—	0.502 4	0.992 5

基于上述两点,表明DNDL-MOPSO在增加种群多样性后并没有以降低收敛性为代价。

5 结束语

本文主要研究了动态邻居维度学习的多目标粒子群算法,采用动态邻居拓扑结构,构建最优维度个体。通过深化维度学习,增加种群的多样性,缓解选择压力。实验结果表明DNDL-MOPSO能够较好地提高种群的多样性,同时保证算法的收敛性,有效解决多峰多目标优化问题。

参考文献:

[1] Kennedy J, Eberhart R. Particle swarm optimization[C]// IEEE International Conference on Neural Network, 1995: 1942-1948.

[2] Kerdphol T, Fuji K, Mitani Y, et al. Optimization of a battery energy storage system using particle swarm optimization for stand-alone microgrids[J]. International Journal of Electrical Power & Energy Systems, 2016, 81: 32-39.

[3] Baumgartner U, Magele C, Renhart W. Pareto optimality and particle swarm optimization[J]. IEEE Transactions on Magnetism, 2004, 40(2): 1172-1175.

[4] Clerc M, Kennedy J. The particle swarm explosion stability and convergence in a multidimensional complex space[J].

IEEE Transactions on Evolutionary Computation, 2002, 2(6): 58-72.

[5] Wang Yong, Cai Zixing. Combining multi-objective optimization with differential evolution to solve constrained optimization problems[J]. IEEE Transactions on Evolutionary Computation, 2012, 16(1): 117-134.

[6] Zhang Guoli, Zuo Hua. Solution analysis of multi-objective programming problem[C]// 2013 International Conference on Machine Learning and Cybernetics, Tianjin, 2013, 3: 1039-1044.

[7] Wei Jingxuan, Wang Yuping. A new model based hybrid particle swarm algorithm for multi-objective optimization[C]// Third International Conference on Natural Computation (ICNC 2007), Haikou, 2007, 3: 497-501.

[8] 任子晖, 王坚. 动态拓扑结构的多目标粒子群优化算法[J]. 同济大学学报: 自然科学版, 2011, 39(8): 1222-1226.

[9] Alvarez-Bentiez J E, Everson R M, Fieldsend J E. A MOPSO algorithm based exclusively on pareto dominance concept[J]. Evolutionary Multi-Criterion Optimization, 2005, 3410: 459-475.

[10] Zhang Yong, Gong Dunwei, Qi Chengliang. Vector evolved multiobjective particle swarm optimization algorithm[C]// 2011 International Conference in Electrics, Communication and Automatic Control Proceedings, 2012: 295-301.

(下转60页)



## 参考文献:

- [1] Golan J S. Semirings and their applications[M]. London: Kluwer Academic Publishers, 1999.
- [2] Chowdhury K R, Sultana A, Mitra N K, et al. On matrices over semirings[J]. Annals of Pure and Applied Mathematics, 2014, 6(1): 1-10.
- [3] Reutenauer C, Straubing H. Inversion of matrices over a commutative semiring[J]. Journal of Algebra, 1984, 88(2): 350-360.
- [4] Tan Yijia. On invertible matrices over commutative semirings[J]. Linear and Multilinear Algebra, 2013, 61(6): 710-724.
- [5] Sirasuntorn N, Sombatoriboon S, Udomsub N. Inversion of matrices over Boolean semirings[J]. Thai Journal of Mathematics, 2009, 7(1): 105-113.
- [6] Liao Yalin, Wang Xueping. Note on invertible matrices over commutative semirings[J]. Linear and Multilinear Algebra, 2016, 64(3): 477-483.
- [7] Pshenitsyna O A. Mappings that preserve the invertibility of matrices over semirings[J]. Translation in Russian Math Surveys, 2009, 64: 162-164.
- [8] 杨阳, 任苗苗, 邵勇. 保持 ai-半环上矩阵的 Moore-Penrose 逆的线性算子[J]. 计算机工程与应用, 2015, 51(8): 37-41.
- [9] 陈艳平, 谭宜家. 关于半环上矩阵的广义逆[J]. 福州大学学报, 2007, 35(6): 797-802.
- [10] 赵萃魁. 分配格中矩阵的可逆条件[J]. 内蒙古大学学报: 自然科学版, 1991(4): 477-480.
- [11] Mora W, Wasanawichit A, Kemprasit Y. Invertible matrices over idempotent semirings[J]. Chamchuri Journal of Mathematics, 2009, 1(2): 55-61.
- [12] Sombatoriboon S, Mora W, Kemprasit Y. Some result concerning invertible matrices over semirings[J]. Science Asia, 2011, 37(2): 130-135.
- [13] Dolzan D, Oblak P. Invertible and nilpotent matrices over antirings[J]. Linear Algebra and Its Applications, 2009, 430(1): 271-278.
- [14] Howie J M. Fundamentals of semigroup theory[M]. Oxford: Oxford Science Publication, 1995.
- [15] Poplin P L, Hartwig R E. Determinantal identities over commutative semirings[J]. Linear Algebra Appl, 2004, 387: 99-132.
- [16] Petrich M, Reilly N R. Completely regular semigroups[M]. New York: Wiley-Interscience Publication, 1999.
- (上接37页)
- [11] 丁晓霖, 侍洪波. 基于自适应网络与动态拥挤距离的多目标粒子群算法及应用[J]. 华东理工大学学报, 2015(2): 173-184.
- [12] Tripathi P K, Bandyopadhyay S, Pal S K. Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients[J]. Information Sciences, 2007, 177(22): 5033-5049.
- [13] Margarita R S, Carlos A, Coello C. Improving PSO-based multi-objective optimization using crowding, mutation and  $\epsilon$ -dominance[J]. Evolutionary Multi-Criterion Optimization, 2005, 3410: 505-519.
- [14] Fernandes C M, Laredo J L J, Merelo J J, et al. Particle swarm optimization with dynamic topology and conservation of evaluations[J]. Computational Intelligence, 2015, 620: 97-111.
- [15] Suganthan P N. Particle swarm optimizer with neighborhood operator[C]//Proceedings of IEEE Congress on Evolutionary Computation (CEC), Piscataway, NJ, USA, 1999, 3: 1958-1962.
- [16] Van D B F, Engelbrecht A P. A cooperative approach to particle swarm optimization[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 225-238.
- [17] Lynn N, Suganthan P N. Comprehensive learning particle swarm optimizer with guidance vector selection[C]//2013 IEEE Symposium on Swarm Intelligence (SIS), Singapore, 2013: 80-84.
- [18] Turkey A M, Abdullah S. A multi-population harmony search algorithm with external archive for dynamic optimization problems[J]. Information Sciences, 2014, 272: 84-95.
- [19] Raquel C R, Naval P C. An effective use of crowding distance in multiobjective particle swarm optimization[C]//Genetic and Evolutionary Computation Conference, Washington DC, 2005: 257-264.
- [20] Zhang Jing, Li Huanqin. A global-crowding-distance based multi-objective particle swarm optimization algorithm[C]//2014 Tenth International Conference on Computational Intelligence and Security (CIS), Kunming, 2014: 1-6.
- [21] 雷德明, 严新平. 多目标智能进化算法及其应用[M]. 北京: 科学出版社, 2009: 38-41.
- [22] Deb K, Pratap A. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 256-279.