

# 知识引导的多目标多智能体进化算法

吴亚丽<sup>†</sup>, 薛芬

(西安理工大学 自动化与信息工程学院, 陕西 西安 710048)

**摘要:** 将智能体模型与知识模型相结合, 提出一种知识引导的多目标多智能体进化算法. 算法定义了智能体的不同邻域环境, 并通过邻域之间的竞争、正交交叉、知识学习等操作实现种群的演化过程. 算法采用一种新颖的方法求非劣解集, 并使用循环拥挤排序法对外部归档集进行维护. 通过对多个测试函数的仿真结果表明, 知识的引入不仅增加了种群多样性, 而且提高了算法的收敛性.

**关键词:** 多智能体; 邻域; 多目标优化; 知识引导

**中图分类号:** TP18      **文献标识码:** A

## Knowledge-guided multi-objective multi-agent evolutionary algorithm

WU Ya-li<sup>†</sup>, XUE Fen

(Automation and Information Engineering School, Xi'an University of Technology, Xi'an Shaanxi 710048, China)

**Abstract:** Combining the agent model and the knowledge model, we propose a knowledge-guided multi-objective multi-agent evolutionary algorithm. Different kinds of neighbor environments of the agent are defined firstly. And then the population evolution process is realized through three operators named competition, orthogonal crossover and knowledge learning in the proposed algorithm. A novel method for finding the non-dominated sets is developed and the circular crowded sorting method is adopted to maintain the external archive-set in the proposed algorithm. Simulation results of several benchmark functions show that the knowledge-guided multi-objective multi-agent evolutionary algorithm not only increases the diversity of the population but also improves the convergence of the algorithm.

**Key words:** multi-agent; neighbor; multi-objective optimization; knowledge-guide

### 1 引言(Introduction)

实际的工程问题具有多种表现形式且一般含有多个目标, 这些目标之间往往是相互制衡的, 只能在各个目标之间进行协调处理, 使所有目标都尽可能地达到最优, 这类问题被称为多目标优化问题(multi-objective optimization problem, MOP). 一般地, 具有 $n$ 个决策变量、 $m$ 个目标变量的多目标优化问题可表述为

$$\begin{aligned} \min F(\mathbf{x}) &= (\mathbf{f}_1(\mathbf{x}), \mathbf{f}_2(\mathbf{x}), \dots, \mathbf{f}_m(\mathbf{x}))^T, \\ \text{s.t.} \quad &\begin{cases} g_i(\mathbf{x}) \leq 0, & i = 1, 2, \dots, p, \\ h_j(\mathbf{x}) = 0, & j = p + 1, \dots, n, \end{cases} \end{aligned} \quad (1)$$

其中:  $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) \in X \subset \mathbb{R}$  表示决策空间  $X$  中的一个  $n$  维向量;  $\mathbf{f} = (\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_m) \in F \subset \mathbb{R}$  表示目标空间  $F$  中的一个  $m$  维向量;

$$g_i(\mathbf{x}) \leq 0 (i = 1, 2, \dots, p)$$

是  $p$  个不等式约束;

$$h_j(\mathbf{x}) = 0 (j = p + 1, \dots, n)$$

是  $n - p$  个等式约束. 对于MOP的几个重要定义见文献[1].

国内外学者对MOP进行了大量研究, 以Deb等<sup>[2]</sup>提出的NSGA-II算法为代表, 该算法通过进行非劣排序和拥挤距离计算得到新种群; Coello Coello等<sup>[1]</sup>提出了多目标粒子群算法, 该算法不需要根据Pareto支配进行适应度赋值, 有较好的全局优化性能; Weiyi Qian等<sup>[3]</sup>提出了多目标自适应差分进化算法, 将一个新的选择算子和自适应参数应用于全局优化中; Daneshyari M等<sup>[4]</sup>将文化框架与粒子群算法结合, 提出一种基于文化的多目标粒子群优化算法; 林浒等<sup>[5]</sup>引入适应度共享策略, 提出面向多目标优化的适应度共享免疫克隆算法; 武星等<sup>[6]</sup>提出一种精英导向型多目标遗传算法, 利用无损有限精度法和归一增量距离保持种群多样性.

多智能体技术已广泛应用于计算机学科各个领域. 由于多智能体系统的分布式并行结构避免了算法进入早熟, 而智能体本身的邻域结构则提高了种群

的局部搜索能力,维持了种群的多样性.钟伟才等<sup>[7]</sup>提出一种多智能体遗传算法,用于解决函数优化问题;曾孝平等<sup>[8]</sup>引入链式结构的智能体进化算法,减少了次优个体取得“顶端优势”导致过早收敛的发生;吴亚丽等<sup>[9]</sup>将链式多种群多智能体进化算法用于求解高维优化问题;潘晓英等<sup>[10]</sup>将智能体技术与关系网模型相结合,提出基于智能体的多目标社会进化算法;韩璞等<sup>[11]</sup>提出多智能体量子多目标进化算法,将量子编码与多智能体进化算法相结合,用于求解环境经济负荷分配(economic/emission load dispatch, EELD)问题.

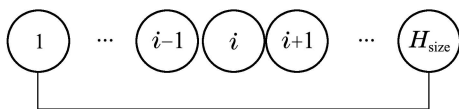
本文将智能体模型与知识模型相结合,提出一种知识引导的多目标多智能体进化算法(the knowledge-guided multi-objective multi-agent evolutionary algorithm, KMMEA).算法采用一种新颖的方法求非解集,使用劣循环拥挤排序法对外部归档集进行维护,智能体邻域之间通过竞争、正交交叉、知识学习等策略完成演化过程.最后,通过对测试函数的仿真验证KMMEA算法的性能优劣.

## 2 智能体邻域及知识学习(Agent neighbor and knowledge learning)

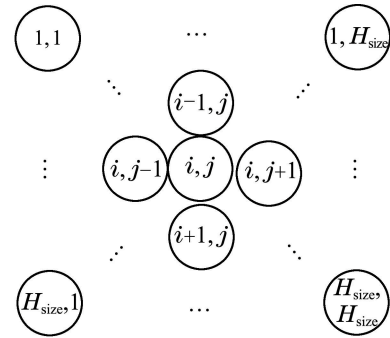
### 2.1 智能体描述(Description of agent)

智能体(agent)是一个物理的或抽象的实体,它能够作用于自身和环境,并对周围的环境做出反应.智能体所能感知的有限环境称之为邻域,每个智能体都具有局部感知能力,可以与其邻域内的智能体相互作用,而该邻域内的智能体又可以和它自身邻域内的智能体相互作用,这样就可以将信息扩散到整个系统中,进而通过智能体与环境以及智能体之间的相互作用达到优化全局的目的.由此可见,智能体邻域的选择对智能体系统的演化起重要的作用.

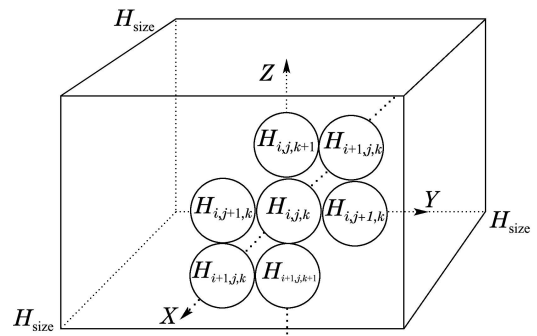
本文首先借鉴维数的思维方式,给出了较为通用的智能体邻域结构,即:一维的线性邻域、二维的平面邻域、三维的空间邻域结构.在一维线性空间中,智能体存在于大小为 $1 \times H_{\text{size}}$ 的链式环境中,每个智能体可以与其相邻的左右两个智能体相互作用,如图1(a)所示;在二维平面空间中,智能体存在于大小 $H_{\text{size}} \times H_{\text{size}}$ 的网格环境中,每个智能体可以与其相邻的上、下、左、右4个智能体相互作用,如图1(b)所示;在三维立体空间中,智能体存在于大小为 $H_{\text{size}} \times H_{\text{size}} \times H_{\text{size}}$ 的立体环境中,每个智能体可以与其相邻的上、下、左、右、前、后6个智能体相互作用,如图1(c)所示;图中的圆圈表示智能体,圆圈中的数字表示智能体在该环境中的位置.



(a) 线性空间邻域



(b) 平面空间邻域



(c) 立体空间邻域

图1 各空间中智能体邻域结构

Fig. 1 The agent neighbor structure of each space

以三维智能体邻域为例,假设 $H_{i,j,k}$ 是智能体立体环境内位置为 $(i, j, k)$  ( $i, j, k = 1, 2, \dots, H_{\text{size}}$ )的一个智能体,那么三维智能体邻域环境可定义为

$$\text{local}(H_{i,j,k}) = \{H_{i_1,j_1,k_1}, H_{i_2,j_2,k_2}, H_{i_3,j_3,k_3}, H_{i_4,j_4,k_4}, H_{i_5,j_5,k_5}, H_{i_6,j_6,k_6}\},$$

其中:

$$\begin{cases} i_1 = \begin{cases} i-1, & i \neq 1, \\ H_{\text{size}}, & i = 1, \end{cases} & j_1 = j, k_1 = k, \\ i_2 = i, & j_2 = \begin{cases} j-1, & j \neq 1, \\ H_{\text{size}}, & j = 1, \end{cases} & k_2 = k, \\ i_3 = i, & j_3 = j, & k_3 = \begin{cases} k-1, & k \neq 1, \\ H_{\text{size}}, & k = 1, \end{cases} \\ i_4 = \begin{cases} i+1, & i \neq H_{\text{size}}, \\ 1, & i = H_{\text{size}}, \end{cases} & j_4 = j, k_4 = k, \\ i_5 = i, & j_5 = \begin{cases} j+1, & j \neq H_{\text{size}}, \\ 1, & j = H_{\text{size}}, \end{cases} & k_5 = k, \\ i_6 = i, & j_6 = j, & k_6 = \begin{cases} k+1, & k \neq H_{\text{size}}, \\ 1, & k = H_{\text{size}}. \end{cases} \end{cases} \quad (2)$$

### 2.2 知识引导的算法(Knowledge-guided algorithm)

知识能够促进群体以一定的速度进化并更快地适应环境,个体知识的积累及种群内部知识的共享交流

均能为群体的进化提供有效的信息. 对智能优化方法的引导, 一方面通过传统人工智能的手段来实现, 另一方面通过特定的知识模型来实现<sup>[12]</sup>. 将智能优化模型和知识模型相结合的方法对优化算法进行引导, 智能优化模型对优化问题的可行空间进行搜索, 知识模型则从前期优化过程中挖掘有用的知识, 采用知识来指导智能优化模型的后续优化过程<sup>[13]</sup>.

多智能体进化算法是把进化算法中的个体看作一个智能体, 该个体就具备了智能体的局部感知、竞争协作及自学习的能力, 可以利用自身的知识进行更新, 提高对环境的适应性, 同时作为其他智能体的邻域, 智能体之间也存在着知识的交互学习, 两者共同影响智能体的进化, 使智能体向更优的方向发展.

### 3 KMMEA 算法实现 (Implementation of KMMEA)

#### 3.1 KMMEA 的算法步骤 (Steps of KMMEA)

**步骤 1** 初始化智能体规模为  $H_{size}$ , 群体规模为 Popsiz, 外部归档集大小为 PS, 最大的迭代次数为 Gen, 令  $iter = 1$ ;

**步骤 2** 初始化种群, 将产生的初始种群分配到每个智能体环境中, 计算每个个体的目标函数值;

**步骤 3** 对环境中的每个智能体由邻域择优算法选择出邻域内 Pareto 最优个体  $W_{best1}$ , 并保存到非劣解集 POS1 中;

**步骤 4** 对智能体进行邻域竞争操作;

**步骤 5** 对智能体进行正交交叉操作;

**步骤 6** 对智能体进行知识学习操作;

**步骤 7** 更新每个智能体的邻域内 Pareto 最优个体  $W_{best2}$ , 并保存到非劣解集 POS2 中;

**步骤 8** 由非劣解集构造法选出 POS1 和 POS2 中的非支配解, 放入外部归档集 NDS 中, 若 NDS 中非支配解数大于 PS, 用循环拥挤排序法进行删减操作;

**步骤 9** 判断是否达到最大迭代次数, 若是, 结束, 输出 Pareto 最优解集; 否则令  $iter = iter + 1$ , 则转步骤 3.

下面对上述步骤中提到的关键操作进行详细介绍.

#### 3.2 智能体邻域择优 (Agent neighbor selection)

邻域择优比较的具体过程为: 每个智能体将与其邻域环境内的所有智能体比较它们的支配关系, 寻找邻域最优解, 若该智能体支配了邻域内的其他智能体或与其他智能体无法比较, 则该智能体为邻域内的最优智能体; 若该智能体被其邻域内的其他智能体所支配, 那么该智能体被取代. 通过引入邻域择优算子可以找到智能体邻域内的最优智能体, 再通过各邻域内最优智能体的比较找到整个种群的最优智能体, 避免了智能体与种群中所有智能体的比较, 节省了算法的运行时间, 提高了算法的速度.

#### 3.3 构造非劣解集 (Non-dominated set construction)

多目标优化问题, 通常采用 NSGA-II<sup>[2]</sup> 中的非支配排序法构造非劣解集, 但该方法容易导致互不支配个体的遗漏. 因此, 本文采用以下方法求非劣解集<sup>[14]</sup>:

**步骤 1** 将所有个体序号初始化为 1;

**步骤 2** 对任意两个个体  $a, b$ , 如果  $a \prec b$ , 那么  $b$  的序号  $rank(b) = rank(b) + 1$ ; 如果  $b \prec a$ , 那么  $a$  的序号  $rank(a) = rank(a) + 1$ ; 如果两者互不支配, 那么个体  $a, b$  的序号不变;

**步骤 3** 将序号为 1 的个体放入非劣解集中, 形成该代的非劣解集.

#### 3.4 智能体进化策略 (Agent evolutionary strategy)

A) 邻域竞争.

假设智能体  $M_{i,j,k} = (m_1, m_2, \dots, m_n)$  为智能体环境中位置为  $(i, j, k)$  的智能体  $H_{i,j,k} = (l_1, l_2, \dots, l_n)$  邻域内 Pareto 最优序列, 其中  $n$  为变量维数. 若  $H_{i,j,k}$  满足  $H_{i,j,k} \prec M_{i,j,k}$ , 则继续留在智能体环境中, 否则采用以下方法产生新智能体:

**步骤 1** 初始化占据概率  $P_0$  和新智能体 NewAgent =  $(e_1, e_2, \dots, e_n)$ , 令  $k = 0$ ;

**步骤 2** 产生随机数  $U(0, 1)$ , 如果  $U(0, 1) < P_0$ , 则转步骤 3, 否则转步骤 4;

**步骤 3** 令

$$e_k = \begin{cases} X_{lk}, & m_k + U(-1, 1) \times (m_k - l_k) < X_{lk}, \\ X_{uk}, & m_k + U(-1, 1) \times (m_k - l_k) > X_{uk}, \\ m_k + U(-1, 1) \times (m_k - l_k), & \text{其他,} \end{cases}$$

其中:  $k = 1, 2, \dots, n$ ;  $X_{lk}$  为第  $k$  维变量的下限;  $X_{uk}$  为第  $k$  维变量的上限;  $U(-1, 1)$  表示产生  $(-1, 1)$  区间内的一个随机数;

**步骤 4** 令  $e_k = e_k^1 + e_k^2$ , 其中:  $e_k^1 = p_{match} \times m_k$ ,  $e_k^2 = (1 - p_{match}) \times m_k$ ,  $p_{match} = \text{rand}/2$ ,  $p_{match}$  为拆分概率;

**步骤 5** 令  $k = k + 1$ , 如果  $k < n$ , 则转步骤 2, 否则转步骤 6;

**步骤 6** 更新智能体环境, 输出邻域内最优个体.

B) 正交交叉.

假设正交交叉<sup>[15]</sup>作用在  $M_{i,j,k} = (m_1, m_2, \dots, m_n)$  和  $H_{i,j,k} = (l_1, l_2, \dots, l_n)$  上,  $H_{i,j,k}$  和  $M_{i,j,k}$  确定的搜索范围为  $[X_1, X_u]$ , 其中:

$$X_1 = (\min(m_1, h_1), \min(m_2, h_2), \dots, \min(m_n, h_n)),$$

$$X_u = (\max(m_1, h_1), \max(m_2, h_2), \dots, \max(m_n, h_n)).$$

第  $r$  维定义域被量化为  $\sigma_{r,1}, \sigma_{r,2}, \dots, \sigma_{r,Q}$ , 其中:

$$\sigma_{r,j} = \begin{cases} \min(m_r, h_r), & j = 1, \\ \min(m_r, h_r) + (j-1) \cdot \frac{m_r - h_r}{Q-1}, & 2 \leq j \leq Q, \\ \max(m_r, h_r), & j = Q, \end{cases}$$

式中 $Q$ 为常数。

具体操作步骤如下:

① 随机生成 $L-1$ 个整数 $k_1, k_2, \dots, k_{L-1}, 1 < k_1 < k_2 < \dots < k_{L-1} < n, n$ 为维数;

② 对任意智能体 $a = (a_1, \dots, a_n)$ 产生 $L$ 个因素 $l_1 = (x_1, \dots, x_{k_1}), l_2 = (x_{k_1+1}, \dots, x_{k_2}), \dots, l_L = (x_{k_{L-1}+1}, \dots, x_n)$ , 第 $i$ 个因素 $l_i$ 所具有的 $Q$ 个级别为

$$\begin{cases} l_i(1) = (\sigma_{k_{i-1}+1,1}, \sigma_{k_{i-1}+2,1}, \dots, \sigma_{k_i,1}), \\ l_i(2) = (\sigma_{k_{i-1}+1,2}, \sigma_{k_{i-1}+2,2}, \dots, \sigma_{k_i,2}), \\ \vdots \\ l_i(Q) = (\sigma_{k_{i-1}+1,Q}, \sigma_{k_{i-1}+2,Q}, \dots, \sigma_{k_i,Q}); \end{cases}$$

③ 用正交矩阵 $H_w(Q^L) = [r_{i,j,k}]_{w \times L}$ 生成 $w$ 个智能体:

$$\begin{cases} (l_1(r_{1,1}), l_2(r_{1,2}), \dots, l_L(r_{1,L})), \\ (l_1(r_{2,1}), l_2(r_{2,2}), \dots, l_L(r_{2,L})), \\ \vdots \\ (l_1(r_{w,1}), l_2(r_{w,2}), \dots, l_L(r_{w,L})); \end{cases}$$

④ 更新智能体环境, 输出邻域内最优个体。

C) 知识学习。

智能体本身知识的学习可以提高算法局部优化的能力, 而智能体与其他智能体间的相互学习可以加速算法的收敛性, 提高算法全局优化的能力。产生随机数 $U(0, 1)$ , 若 $U(0, 1) < P_m$  ( $P_m$ 为变异概率), 采用智能体间知识学习, 否则采用智能体自身知识学习。

**步骤 1** 初始化 $P_m$ 和新智能体 $a'$ , 令 $k = 1$ ;

**步骤 2** 若 $U(0, 1) < P_m$ , 转步骤 3, 采用智能体间知识学习, 否则转步骤 4, 采用智能体自身知识学习;

**步骤 3** 任选 2 个智能体 $h_p, h_q (1 \leq p \leq \text{Popsizel}, 1 \leq q \leq \text{Popsizel})$ , 并任意选择一维 $u$ , 采用公式

$$H'(k, u) = H(k, u) + \text{rand} \cdot (H(p, u) - H(q, u))$$

对 $L(k, u)$ 进行指导性变异操作, 其中 $H'(k, u)$ 表示变异后第 $k$ 个智能体的第 $u$ 维;

**步骤 4** 对第 $k$ 个智能体的任意一维 $v$ , 采用

$$H'(k, v) = X_{\min} + \text{rand} \cdot (X_{\max} - X_{\min})$$

对 $H(k, v)$ 进行学习;

**步骤 5** 若变量的某一维超出其范围, 则将边界值赋给该维变量, 令 $k = k + 1$ ;

**步骤 6** 若 $k < \text{Popsizel}$ , 则转步骤 2, 否则转步骤 7;

**步骤 7** 更新智能体环境, 输出邻域内最优个体。

### 3.5 外部归档集维护策略(Maintenance policy of external archive)

本文采用循环拥挤排序法<sup>[16]</sup>对外部归档集进行改善, 具体操作为:

**步骤 1** 初始化每个个体的拥挤距离,  $P[i]_{\text{distance}} = 0$ ;

**步骤 2** 对每个目标函数值进行排序, 设 $f_1$ 和 $f_2$ 为两个子目标, 根据公式 $P[i]_{\text{distance}} = (P[i+1].f_1 - P[i-1].f_1) + (P[i+1].f_2 - P[i-1].f_2)$ 计算每个个体的拥挤距离;

**步骤 3** 为了保证边界点每次迭代过程中都能进入下一代, 给它们赋予最大值, 删除外部归档集中拥挤距离最小的个体;

**步骤 4** 重新对每个目标函数值进行排序, 根据步骤 2 中的公式更新每个个体的拥挤距离;

**步骤 5** 判断外部归档集大小是否溢出, 若是, 结束, 否则转步骤 2。

## 4 实验设计与结果分析(Experiment designs and results)

### 4.1 测试函数与评价指标(Test functions and evaluation indexes)

为了验证算法的性能, 本文采用文献[2]中的测试函数SCH, FON, ZDT1, ZDT2, ZDT3, ZDT4, ZDT6等对算法进行仿真测试。采用收敛性指标( $\gamma$ )和多样性指标( $\Delta$ )对算法得到的Pareto前沿的性能进行评价和分析。

收敛性指标公式为

$$\gamma = \frac{1}{|N|} \sum_{y \in \mathcal{T}} \min\{|y - \bar{y}|, \bar{y} \in \bar{\mathcal{T}}\}, \quad (3)$$

其中:  $\mathcal{T}$ 为算法得到的非劣解集,  $N$ 为非劣解个数,  $\bar{\mathcal{T}}$ 为真实的非劣解集。收敛性指标越小, 说明所得的非劣解集越接近真实的Pareto前沿。

多样性指标公式为

$$\Delta = \frac{d_f + d_1 + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_1 + (N-1)\bar{d}}, \quad (4)$$

其中:  $d_i$ 为连续两个非劣解间的欧式距离,  $\bar{d}$ 为所有 $d_i$ 的平均值,  $d_f$ 和 $d_1$ 分别为算法获得的边界解与相应极端解间的距离。保持Pareto解集的多样性能使其具有较好的均匀性, 多样性指标越小, 说明所得的非劣解集在Pareto前端越均匀。

### 4.2 参数设置及仿真结果分析(Parameter setting and simulation results)

1) 不同维数邻域结构对算法性能的影响。

假设一维空间中的算法为KMMEA-I, 二维空间中

的算法为KMMEA-II, 三维空间中的算法为KMMEA-III. KMMEA-I中 $H_{size}$ 取100, 即种群规模 $Popsiz = H_{size} = 100$ , 迭代次数为250次, KMMEA-II中 $H_{size}$ 取10, 即种群规模 $Popsiz = H_{size} \times H_{size} = 100$ , 迭代次数为250次, KMMEA-III中 $H_{size}$ 取5, 即种群规模 $Popsiz = H_{size} \times H_{size} \times H_{size} = 125$ , 迭代次数为

200次, 3种邻域结构中总的函数评价次数均为25000次, 外部归档集的规模均为100, 占据概率 $P_0 = 0.85$ , 变异概率 $P_m = 0.3$ , 交叉操作中 $Q = 3$ .

表1统计了这3种不同维数邻域结构执行30次的收敛度( $\gamma$ )和多样性( $\Delta$ )的均值、方差以及平均运行时间. 其中:  $M$ 为均值, VAR为方差.

表1 不同邻域环境中KMMEA算法性能评价指标

Table 1 Evaluation index of KMMEA in different neighbor environments

函数	算法	收敛性指标 $\gamma$		多样性指标 $\Delta$		时间/s
		$M$	VAR	$M$	VAR	
FON	I	9.1807e-4	3.2744e-9	0.0161	5.7266e-5	2.70561
	II	9.2842e-4	3.5102e-9	0.0171	3.8359e-5	2.83334
	III	9.5007e-4	3.9594e-9	0.0183	4.8254e-5	3.16879
SCH	I	0.0032	2.8936e-8	0.0222	8.6342e-5	2.12460
	II	0.0032	3.8985e-8	0.0234	8.4426e-5	2.17220
	III	0.0032	4.0808e-8	0.0258	1.5525e-5	2.23973
ZDT1	I	0.0011	5.4467e-8	0.0110	2.5633e-5	7.27542
	II	0.0011	1.3278e-8	0.0124	3.9017e-5	7.56367
	III	0.0011	1.8614e-8	0.0136	4.6674e-5	7.91430
ZDT2	I	8.0068e-4	3.8394e-9	0.0138	4.2092e-5	7.45913
	II	7.9842e-4	2.2997e-9	0.0142	5.9630e-5	8.06291
	III	8.0131e-4	2.7845e-9	0.0166	4.3022e-5	8.51436
ZDT3	I	0.0012	7.0098e-9	0.0111	3.8981e-5	7.24397
	II	0.0012	8.0206e-9	0.0117	4.0816e-5	7.81969
	III	0.0012	5.7122e-9	0.0122	3.2206e-5	8.35193
ZDT4	I	9.8694e-4	1.8862e-8	0.0117	2.6964e-5	7.08159
	II	9.9311e-4	2.0582e-8	0.0129	3.6105e-5	7.57184
	III	0.0010	1.4093e-8	0.0139	4.2144e-5	7.94070
ZDT6	I	0.0884	0.0097	0.0749	0.0405	7.67501
	II	0.0046	4.0029e-8	0.0186	8.1906e-5	8.39433
	III	0.0040	3.4527e-8	0.0190	3.7766e-5	9.06130

从表1中可以看出, 在相同的参数设置下, 对于FON, SCH, ZDT1, ZDT2, ZDT3和ZDT4, 由于算法具有相同的知识学习策略, KMMEA-I, KMMEA-II与KMMEA-III具有相近的收敛性, 智能体间的相互知识学习加速了算法的收敛性; 而KMMEA-I比KMMEA-II和KMMEA-III的多样性相对较好, 这是由于每个智能体在其邻域内进行择优操作中, 而不是在整个种群中进行, 从而保证了群体的多样性; 由于ZDT6是非均匀分布的函数, KMMEA-I得到的收敛性和多样性指标均不如KMMEA-II与KMMEA-III; 从算法运行时间看, 由于KMMEA-I与邻域内的2个智能体相互作用, KMMEA-II与邻域内的4个智能体相互作用, KMMEA-III与邻域内的6个智能体相互作用. 因此, 运行时间从优到劣依次为KMMEA-I, KMMEA-II和KMMEA-III.

为了更直观地表明算法的性能, 图2给出具有不

同维数邻域的KMMEA对各测试函数的仿真结果图. 图2中, 实线表示优化问题理想的Pareto最优前沿, 实点表示算法所得到的Pareto前沿. 由图2可知, 对于FON, SCH, ZDT1, ZDT2, ZDT3和ZDT4, 3种维数的邻域结构算法均收敛到Pareto前沿上, 特别是对难于优化的ZDT4问题, 算法得到的结果不仅收敛性非常好, 而且在目标空间中均匀分布; 对于ZDT6问题, KMMEA-II与KMMEA-III均收敛到Pareto最优前沿上, 而KMMEA-I得到的收敛性则不如这两者好.

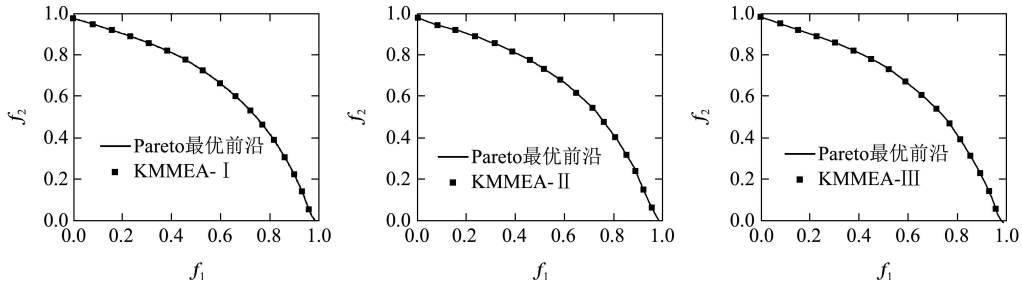
2) 本文算法与文献算法比较.

由于KMMEA-II的综合性能较优, 本文以KMMEA-II为例, 将其与文献中的NSGA-II<sup>[2]</sup>、MOPSO<sup>[1]</sup>、ADEA<sup>[3]</sup>、RM-MEDA<sup>[17]</sup>的性能进行比较分析. 表2-3分别统计了各算法的收敛度指标( $\gamma$ )和多样性指标( $\Delta$ )的均值及方差, 其中:  $M$ 为均

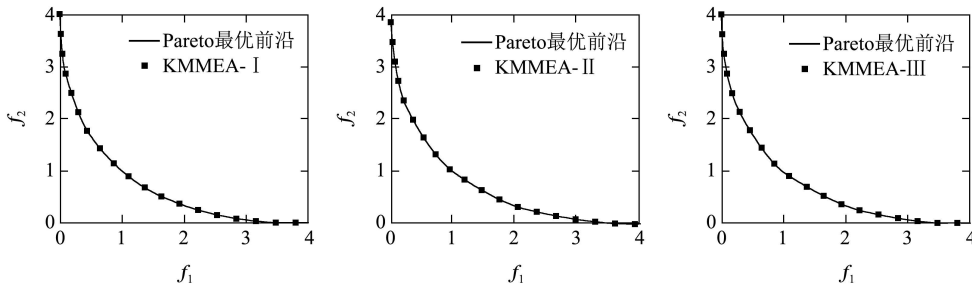
值, VAR为方差, —表示文献中无对应项统计结果.

从算法的收敛性来看, 对于SCH, FON, ZDT1, ZDT2, ZDT3, ZDT4, KMMEA得到的收敛性指标的均值和方差均优于其他算法, 而对于ZDT6, 虽然KMMEA的收敛性均值不如ADEA, 但却优于其他算法, KMMEA得到的Pareto最优解集更接近函数

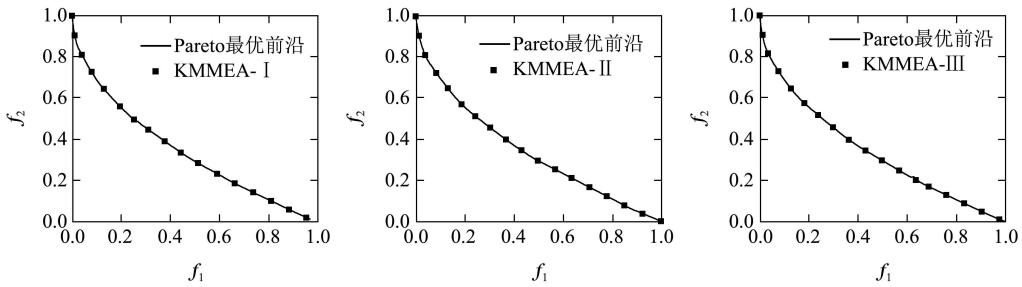
真实的Pareto前沿, 有较好的收敛性. 从解的多样性来看, KMMEA对所有函数的多样性指标均优于其他4种算法, 这是由于每个智能体在其邻域内进行Pareto择优操作, 并不是在整个种群中进行的, 从而使得种群具有较好的多样性, KMMEA得到的Pareto最优前沿具有较好的均匀性.



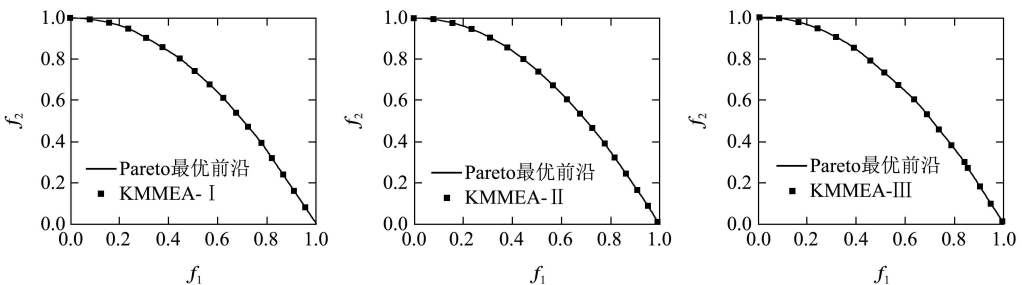
(a) FON



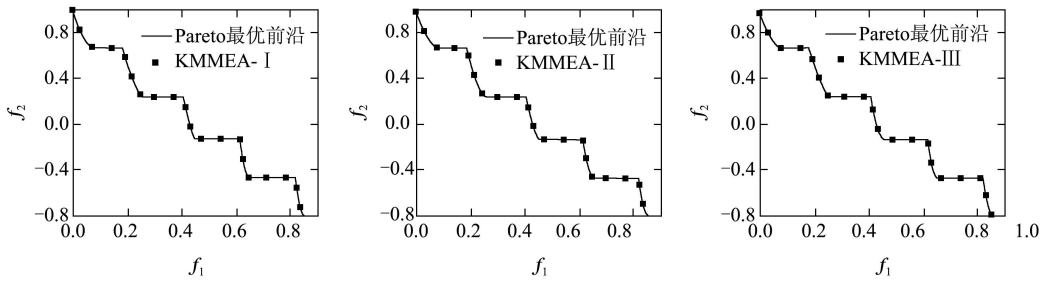
(b) SCH



(c) ZDT1



(d) ZDT2



(e) ZDT3

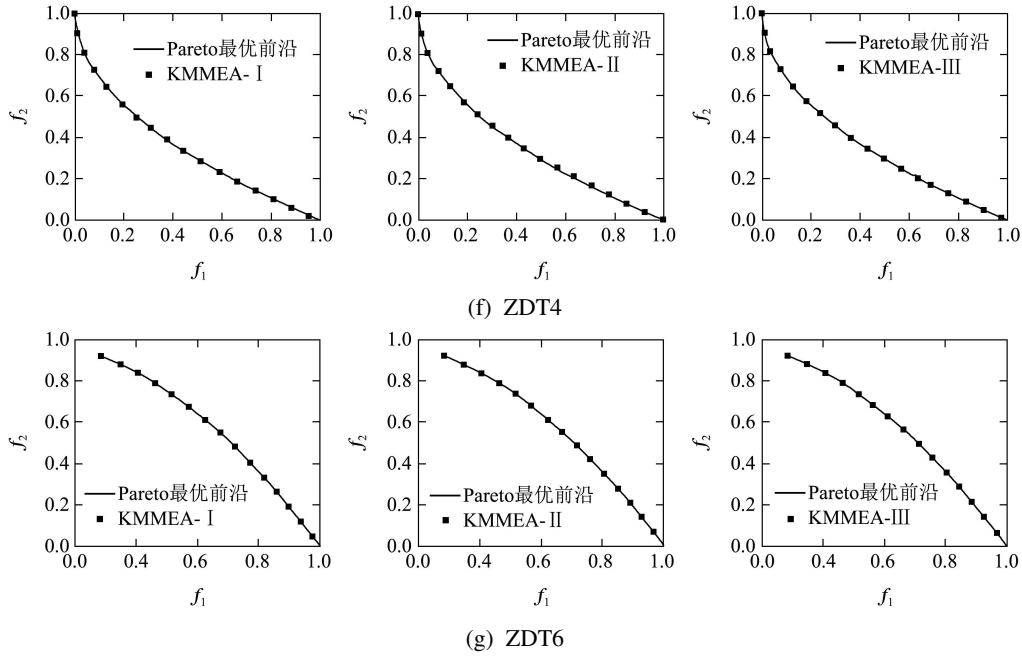


图 2 不同邻域结构的KMMEA对测试函数仿真结果

Fig. 2 Simulation results of the test functions with different neighbor structure of KMMEA

表 2 收敛性 $\gamma$ 的均值和方差

Table 2 Mean and variance of the convergence metric  $\gamma$

算法	指标	SCH	FON	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
KMMEA	$M$	0.0032	9.2842e-4	0.0011	7.9842e-4	0.0012	9.9311e-4	0.0046
	VAR	3.8985e-8	3.5102e-9	1.3278e-8	2.2997e-9	8.0206e-9	2.0582e-8	4.0029e-8
ADEA	$M$	—	—	0.002741	0.002203	0.002741	0.100100	0.000624
	VAR	—	—	0.000385	0.000297	0.000120	0.446200	0.000060
RM-MEDA	$M$	—	—	0.02701	0.03456	0.05592	55.09667	0.94110
	VAR	—	—	0.00011	0.00021	0.00043	5.52171	0.03743
NSGA-II	$M$	0.003391	0.001931	0.033482	0.072391	0.114500	0.513053	0.296564
	VAR	0	0	0.004750	0.031689	0.007940	0.118460	0.013135
MOPSO	$M$	0.01148	—	0.00133	0.00089	0.00418	7.37429	—
	VAR	0	—	0	0	0	5.48286	—

表 3 多样性 $\Delta$ 的均值和方差

Table 3 Mean and variance of the convergence metric  $\Delta$

算法	指标	SCH	FON	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
KMMEA	$M$	0.0234	0.0183	0.0124	0.0142	0.0117	0.0129	0.0186
	VAR	8.4426e-5	4.8254e-5	3.9017e-5	5.9630e-5	4.0816e-5	3.6105e-5	8.1906e-5
ADEA	$M$	—	—	0.382890	0.345780	0.525770	0.436300	0.361100
	VAR	—	—	0.001435	0.003900	0.043030	0.110000	0.036100
RM-MEDA	$M$	—	—	0.23787	0.28013	0.69307	0.76690	0.79547
	VAR	—	—	0.00110	0.00386	0.00648	0.00362	0.00180
NSGA-II	$M$	0.477899	0.378065	0.390307	0.430776	0.738540	0.702612	0.668025
	VAR	0.003471	0.000639	0.001876	0.004721	0.019706	0.064648	0.009923
MOPSO	$M$	0.76097	—	0.68132	0.63922	0.83195	0.96194	—
	VAR	0.01643	—	0.01335	0.00114	0.00892	0.00114	—

## 5 结论(Conclusions)

本文将人工智能领域中的多智能体模型与知识模型相结合,提出了一种知识引导的多智能体多目标进化算法.算法采用循环拥挤排序法控制外部归档集大小,获得在目标空间上均匀分布的非劣解,提高了Pareto前沿的均匀性.智能体通过邻域择优操作寻找个体Pareto最优解,避免与整个种群空间的比较,节省了时间.各智能体通过邻域竞争,正交叉以及知识学习等操作指导算法向Pareto最优前沿逼近.通过对一系列测试函数的仿真,并对智能体的不同邻域结构对算法性能的影响和文献算法进行比较研究,结果表明本文算法在求解多目标优化问题时,在保证收敛性的同时又具有较好的多样性和均匀性.

## 参考文献(References):

- [1] COELLO C C A, PULIDO G T, LECHUGA M S. Handling multiple objectives with particle swarm optimization [J]. *IEEE Transaction on Evolutionary Computation*, 2004, 8(3): 256 – 279.
- [2] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182 – 197.
- [3] QIAN W Y, LI A J. Adaptive differential evolution algorithm for multiobjective optimization problems [J]. *Applied Mathematics and Computation*, 2008, 201(1/2): 431 – 440.
- [4] DANESHYARI M, YEN G G. Cultural-based multiobjective particle swarm optimization [J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2011, 41(2): 553 – 567.
- [5] 林浒, 彭勇. 面向多目标优化的适应度共享免疫克隆算法 [J]. *控制理论与应用*, 2011, 28(2): 206 – 214.  
(LIN Hu, PENG Yong. Immune clonal algorithm with fitness sharing for multi-objective optimization [J]. *Control Theory & Applications*, 2011, 28(2): 206 – 214.)
- [6] 武星, 楼佩煌, 唐敦兵. 基于精英进化导向的多目标PID参数优化 [J]. *控制理论与应用*, 2010, 27(9): 1235 – 1239.  
(WU Xing, LOU Peihuang, TANG Dunbing. Multi-objective optimization for PID parameter based on elitist-evolution guidance [J]. *Control Theory & Applications*, 2010, 27(9): 1235 – 1239.)
- [7] LIU J, ZHONG W C, JIAO L C. A multiagent evolutionary algorithm for constraint satisfaction problems [J]. *IEEE Transactions on System, Man, and Cybernetics, Part B: Cybernetics*, 2006, 36(1): 1128 – 1141.
- [8] ZENG X P, LI Y M, JIAN Q. A dynamic chain-like agent genetic algorithm for global numerical optimization and feature selection [J]. *Neurocomputing*, 2009, 72(4/5/6): 1214 – 1228.
- [9] 吴亚丽, 靳笑一, 刘格. 链式多种群多智能体进化算法 [J]. *控制理论与应用*, 2013, 30(1): 37 – 53.  
(WU Yali, JIN Xiaoyi, LIU Ge. Chainlike multi-population multi-agent evolutionary algorithm [J]. *Control Theory & Applications*, 2013, 30(1): 37 – 53.)
- [10] 潘晓英, 刘芳, 焦李成. 基于智能体的多目标社会进化算法 [J]. *软件学报*, 2009, 20(7): 1703 – 1713.  
(PAN Xiaoying, LIU Fang, JIAO Licheng. Multiobjective social evolutionary algorithm based on multi-agent [J]. *Journal of Software*, 2009, 20(7): 1703 – 1713.)
- [11] 韩璞, 刘立衡, 王东风. 多智能体量子多目标进化算法及其在EELD问题中的应用 [J]. *系统仿真学报*, 2010, 22(4): 872 – 876.  
(HAN Pu, LIU Liheng, WANG Dongfeng. Multi-agent Quantum multi-objective evolutionary algorithm and its application to EELD problem [J]. *Journal of System Simulation*, 2010, 22(4): 872 – 876.)
- [12] 邢立宁, 陈英武. 基于知识的智能优化引导方法研究进展 [J]. *自动化学报*, 2011, 37(11): 1285 – 1289.  
(XING Lining, CHEN Yingwu. Research progress on intelligent optimization guidance approaches using knowledge [J]. *Acta Automatica Sinica*, 2011, 37(11): 1285 – 1289.)
- [13] 邢立宁, 陈英武. 知识型智能优化方法研究 [M]. 长沙: 国防科学技术大学出版社, 2010.  
(Xing Lining, Chen Yingwu. *Research on the Knowledge-based Intelligent Approaches* [M]. Changsha: National University of Defense Technology Press, 2010.)
- [14] 吴亚丽, 徐丽青. 一种基于粒子群算法的改进多目标文化算法 [J]. *控制与决策*, 2012, 27(8): 1127 – 1132.  
(WU Yali, XU Liqing. An improved multi-objective cultural algorithm based on particle swarm optimization [J]. *Control and Decision*, 2012, 27(8): 1127 – 1132.)
- [15] LEUNG Y W, WANG Y P. An orthogonal genetic algorithm with quantization for global numerical optimization [J]. *IEEE Transactions on Evolutionary Computation*, 2001, 5(1): 41 – 53.
- [16] 罗辞勇, 陈民铀, 张聪誉. 采用循环拥挤排序策略的改进NSGA-II算法 [J]. *控制与决策*, 2010, 25(2): 227 – 231.  
(LUO Ciyong, CHEN Minyou, ZHANG Congyu. Improved NSGA-II algorithm with circular crowded sorting [J]. *Control and Decision*, 2010, 25(2): 227 – 231.)
- [17] ZHANG Q F, ZHOU A, JIN Y. RM-MEDA: a regularity model-based estimation of distribution algorithm [J]. *IEEE Transactions on Evolutionary Computation*, 2008, 12(1): 41 – 63.

## 作者简介:

**吴亚丽** (1975–), 女, 副教授, 主要研究方向为智能优化算法理论及应用研究、复杂系统建模与优化, E-mail: yliwu@xaut.edu.cn;

**薛芬** (1990–), 女, 硕士研究生, 主要研究方向为大系统理论与方法, E-mail: xuefenfen6@163.com.