

种群动力学优化算法

黄光球 李涛 陆秋琴

(西安建筑科技大学管理学院 西安 710055)

摘要 为了快速求解大规模复杂优化问题,基于种群动力学理论构造出了可全局收敛的种群动力学优化算法。在该算法中,每个种群对应着优化问题的一个试探解,种群的一个特征对应于试探解的一个变量;采用正交拉丁方原理构造出了种群初始值确定方法,以达到对搜索空间的均衡分散性和整齐可比性覆盖;将任意两种群间的竞争、互利、捕食-被食、融合、突变和选择等行为用于构造种群的进化策略,以使种群的适应度指数要么保持原状不变,要么向好的方向转移,从而确保整个算法的全局收敛性;在种群演变过程中,种群从一种状态转移到另一种状态,实现了种群对优化问题全局最优解的搜索。应用可归约随机矩阵的稳定性条件证明了本算法具有全局收敛性。测试结果表明本算法是高效的。

关键词 进化计算,函数优化,生物地理学优化算法,种群动力学

中图分类号 TP18 **文献标识码** A

Population Dynamics-based Optimization

HUANG Guang-qiu LI Tao LU Qiu-qin

(School of Management, Xi'an University of Architecture & Technology, Xi'an 710055, China)

Abstract To solve large-scale optimization problems(OP), a population dynamics-based optimization algorithm with global convergence was constructed based on the population dynamics theory. In the algorithm, each population is just an alternative solution of OP, and a feature attribute of a population corresponds to a variable of an alternative solution. The principle of orthogonal Latin squares is used to produce initial values of populations so as to cover search space with balance dispersion and neat comparability. The competition, mutual-benefit, predator-prey, mergence, mutation and selection behavior between any two populations are used to construct evolution policies of populations so as to ensure population suitability index(PSI) of each population to keep either to stay unchanged or to transfer toward better states, therefore the global convergence is ensured. During evolution process of populations, each population's transferring from one state to another realizes the search for the global optimum solution. The stability condition of a reducible stochastic matrix was applied to prove the global convergence of the algorithm. The case study shows that the algorithm is efficient.

Keywords Evolutionary computation, Function optimization, Biogeography-based optimization, Population dynamics

1 引言

种群动力学模型是描述种群与环境、种群与种群之间相互作用的动力学关系的数学模型^[1],可用于描述、预测以至调节和控制物种的发展过程与发展趋势。常用的种群动力学模型包括 Malthus 人口模型^[2]、Logistic 模型^[1,2]和 Lotka-Volterra 模型^[3]。近年来,种群动力学模型得到广泛的研究和应用,先后得到了 Logistic 模型和 Lotka-Volterra 模型的滞后效应模型^[4]、功能性反应作用模型^[5,6]、反应扩散模型^[7]及年龄结构种群模型^[8]等,这些修正模型考虑到了种群的增长率和环境容纳量的时间效应^[9]、非线性繁殖特性^[1]、进化过程的时滞效应^[10]和随机干扰等因素^[1,2]。

2008 年, Dan Simon 用生物地理学的方法提出了生物地理学优化算法(Biogeography-Based Optimization, BBO)^[11]。该算法通过种群在栖息地间的迁移实现了对优化问题最优解的搜索。BBO 算法解决优化问题主要依赖以下 2 方面^[12]: (1) 栖息地的特征向量 SIV 对应优化问题的试探解; 栖息地的适宜度指数(HSI)对应于优化问题的目标函数值,好的试探解具有较高 HSI 值; (2) 栖息地的迁入和迁出机制对应优化算法中的信息交互机制,高 HSI 的试探解以一定的迁出率进行相应操作,将信息共享给低 HSI 试探解;低 HSI 试探解通过高 HSI 的试探解接受许多新的特征,这些额外的新特征可以提高低 HSI 试探解的质量。若栖息地的较高 HSI 使得该栖息地种群数量增多,则调低迁入率、调高迁出率。

到稿日期:2013-01-27 返修日期:2013-04-05 本文受陕西省科学技术研究发展计划项目(2013K11-17),陕西省教育厅科技计划项目(12JK0789),陕西重点学科建设专项资金资助项目(E08001),陕西房地产技术经济及管理研究(E08005)资助。

黄光球(1964—),男,博士,教授,主要研究方向为智能计算、计算机模拟, E-mail: huangnan93@sohu.com; 李涛(1988—),男,硕士生,主要研究方向为智能计算; 陆秋琴(1966—),女,博士,教授,主要研究方向为先进计算、计算机模拟。

本文提出的种群动力学优化算法(Population Dynamics-based Optimization,简称 PDO算法)采用与 BBO 算法完全不同的设计思路,即: BBO 算法采用的是生物地理学理论,而 PDO 算法采用的是种群动力学理论,这两种理论存在很大差别。PDO 算法假定一生态系统中存在多种不同种类的生物种群,种群间进行自由竞争、互利、捕食-被食以及融合等活动,通过优胜劣汰,强壮者获得进化,虚弱者被淘汰。而 BBO 算法是通过种群的迁入和迁出实现进化。

测试结果表明, PDO 算法具有很强的搜索能力,且具有全局收敛性,可以用于求解大规模优化问题。

2 种群动力学优化算法构造方法

2.1 一般优化问题定义

考虑一般优化问题:

$$\begin{aligned} \min f(\mathbf{X}) \\ \text{s. t. } \begin{cases} g_i(\mathbf{X}) \geq 0, i=1, 2, \dots, I \\ h_i(\mathbf{X}) = 0, i=1, 2, \dots, E \\ \mathbf{X} \in S \subset R^n, \mathbf{X} \geq 0 \end{cases} \end{aligned} \quad (1)$$

式中, R^n 是 n 维欧氏空间; $\mathbf{X}=(x_1, x_2, \dots, x_n)$ 是一个 n 维非负实数决策向量; S 为搜索空间, 又称解空间; $f(\mathbf{X})$ 为目标函数; $g_i(\mathbf{X}) \geq 0$ 为第 i 个约束条件, $i=1, 2, \dots, I$, I 为不等式约束条件个数; $h_i(\mathbf{X}) = 0$ 为第 i 个等式约束条件, $i=1, 2, \dots, E$, E 为等式约束条件个数。目标函数 $f(\mathbf{X})$ 和约束条件 $g_i(\mathbf{X})$ 、 $h_i(\mathbf{X})$ 不需要特殊的限制条件, 传统的基于函数连续性和可导性的数学优化方法^[1]无法解决该问题。

为了使本文提出的算法适用于各种优化问题, 将优化问题(1)的目标函数改写成式(2):

$$F(\mathbf{X}) = \begin{cases} \forall i \in \{1, 2, \dots, I\}, g_i(\mathbf{X}) \geq 0; \\ f(\mathbf{X}), \quad \forall j \in \{1, 2, \dots, E\}, h_j(\mathbf{X}) = 0; \\ \mathbf{X} \in S \subset R^n, \mathbf{X} \geq 0 \\ F_{\max}, \quad \text{其它} \end{cases} \quad (2)$$

式中, F_{\max} 为非常大的实数, 用于对不满足约束条件的试探解进行惩罚。

2.2 Lotka-Volterra 种群动力学模型

生物群落是在特定时间聚集在一定地域或生境中所有生物种群的集合。设群落由 N 个种群组成, $N \geq 2$ 。种群间相互作用关系可分为两大类: a) 无相互作用关系: 包括互不影响的中性作用, 一方受害、另一方不受影响的偏害作用以及一方受益、另一方不受影响的偏利作用; b) 相互作用关系: 包括竞争(资源利用性竞争和相互干扰性竞争)、互利(专性互利共生和兼性互利共生)和捕食作用(捕食-被食)。PDO 算法中的算子可利用上述各类相互作用关系进行构建。可描述各种种群间相互作用的种群动力学模型为 Lotka-Volterra 模型, N 个种群的 Lotka-Volterra 模型如下:

$$\frac{dx_i}{dt} = x_i(r_i - \sum_{j=1}^N a_{ij}x_j), i=1, 2, \dots, N \quad (3)$$

式中, x_i 是种群 i 的大小, $x_i \geq 0$; $r_i > 0$ 称为种群 i 的内禀增长率; a_{ij} ($i \neq j$) 的正负符号和绝对值表示种群 j 对种群 i 的影响性质和强度; a_{ii} 表示种群 i 的种内调节参数, $a_{ii} < 0$ 表示种内竞争或种内密度制约^[1]。下面我们重点研究 2 个种群间的 Lotka-Volterra 模型, 由式(3)可得:

$$\begin{cases} \frac{dx_1}{dt} = x_1(r_1 + a_{11}x_1 + a_{12}x_2) \\ \frac{dx_2}{dt} = x_2(r_2 + a_{21}x_1 + a_{22}x_2) \end{cases} \quad (4)$$

式中, 各符号及其含义说明如下:

(1) $r_1 < 0$ 且 $r_2 < 0$, 表示互利共生两个种群如果分离则不能单独存活的死亡率。

(2) $r_1 > 0$ 且 $r_2 > 0$, 表示种群 1 和 2 的自然增长率, 种群 1 和种群 2 可以单独存活。

(3) $r_1 > 0$ 且 $r_2 < 0$, 表示种群 1 可以离开种群 2 而单独存活, 但种群 2 不能离开种群 1 而单独存活, 这是一种捕食-被食关系。

(4) 当 $a_{12} < 0$ 且 $a_{21} > 0$ 时, 表示式(4)是捕食-被食关系系统。

(5) 当 $a_{12} < 0$ 且 $a_{21} < 0$ 时, 表示式(4)是竞争关系系统。

(6) 当 $a_{12} > 0$ 且 $a_{21} > 0$ 时, 表示式(4)是互利关系系统。

(7) 当 $a_{12} = 0$ 且 $a_{21} > 0$ 或 $a_{12} > 0$ 且 $a_{21} = 0$ 时, 表示式(4)是偏利关系系统。

(8) 当 $a_{12} = 0$ 且 $a_{21} < 0$ 或 $a_{12} < 0$ 且 $a_{21} = 0$ 时, 表示式(4)是偏害关系系统。

(9) 当 $a_{12} = 0$ 且 $a_{21} = 0$ 时, 表示式(4)是中性关系系统。

(10) 当 $a_{11} < 0$ 且 $a_{22} < 0$ 时, 表示式(4)是种内密度制约的。

2.3 优化问题试探解与种群动力学参数的关系

生态系统中一个种群由 n 个特征来表示, 一个种群对应于一个优化问题的试探解, N 个种群所对应的优化问题的试探解解集为 $S = \{X_1, X_2, \dots, X_N\}$, $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$, $i=1, 2, \dots, N$; 种群 i 中的一个特征 j 对应于优化问题试探解 X_i 中的一个变量 x_{ij} ; 种群的适应度指数 PSI(Population Suitability Index, PSI)对应于优化问题的目标函数值。好的试探解对应具有较高 PSI 指数的种群, 差的试探解对应具有较低 PSI 指数的种群, 即对于优化问题(1), 种群 i 中的适应度指数即 PSI 指数为:

$$PSI(X_i) = F_{\max} - F(X_i) \quad (5)$$

2.4 种群动力学进化算子

PDO 算法利用种群动力学模型来构造进化算子实现种群间信息交换, 进而实现对优化问题解空间的搜索。PDO 算法的进化算法如下所述。

(1) 竞争算子。种群动力学竞争算子描述的是任意 2 个种群间的竞争行为。依据式(4), 可以构造出种群 i 与种群 j 间的竞争算子, 即对于 $i, j \in \{1, 2, \dots, N\}$, $i \neq j$, $s \in \{1, 2, \dots, n\}$, 有

$$\begin{cases} v_{is}^t = x_{is}^{t-1}(1 + r_1 - a_{11}x_{is}^{t-1} - a_{12}x_{js}^{t-1}) \\ v_{js}^t = x_{js}^{t-1}(1 - r_2 - a_{21}x_{is}^{t-1} - a_{22}x_{js}^{t-1}) \end{cases} \quad (6)$$

式中, v_{is}^t 和 v_{js}^t 为种群 i 与种群 j 的特征 s 在时期 t 时的状态值, $\mathbf{V}_i^t = (v_{i1}^t, v_{i2}^t, \dots, v_{in}^t)$, $\mathbf{V}_j^t = (v_{j1}^t, v_{j2}^t, \dots, v_{jn}^t)$; x_{is}^{t-1} 和 x_{js}^{t-1} 为种群 i 与种群 j 的特征 s 在时期 $t-1$ 时状态值, $\mathbf{X}_i^{t-1} = (x_{i1}^{t-1}, x_{i2}^{t-1}, \dots, x_{in}^{t-1})$, $\mathbf{X}_j^{t-1} = (x_{j1}^{t-1}, x_{j2}^{t-1}, \dots, x_{jn}^{t-1})$; 参与竞争的种群 $i = \text{Rand}(1, N)$, $j = \text{Rand}(1, N)$, $i \neq j$; $\text{Rand}(a, b)$ 表示在 $[a, b]$ 区间产生一个均匀分布随机数; 因将正负号直接放入到表达式中, 所以有 $r_1 > 0$, $a_{11} > 0$, $a_{12} > 0$, $r_2 > 0$, $a_{21} > 0$, $a_{22} > 0$, 下同。

(2) 互利算子。种群动力学互利算子描述的是任意 2 个种群间的互利行为,依据式(4),可以构造出种群 i 与种群 j 间的互利算子,即对于 $i, j \in \{1, 2, \dots, N\}, i \neq j, s \in \{1, 2, \dots, n\}$, 有

$$\begin{cases} v_i^s = x_i^{s-1}(1+r_1-a_{11}x_i^{s-1}+a_{12}x_j^{s-1}) \\ v_j^s = x_j^{s-1}(1-r_2+a_{21}x_i^{s-1}-a_{22}x_j^{s-1}) \end{cases} \quad (7)$$

式中,参与互利的种群 $i = \text{Rand}(1, N), j = \text{Rand}(1, N), i \neq j$ 。

(3) 捕食-被食算子。种群动力学捕食-被食算子描述的是任意 2 个种群间的捕食-被食行为,依据式(4),可以构造出种群 i 与种群 j 间的捕食-被食算子,即对于 $i, j \in \{1, 2, \dots, N\}, i \neq j, s \in \{1, 2, \dots, n\}$, 有

$$\begin{cases} v_i^s = x_i^{s-1}(1+r_1-a_{11}x_i^{s-1}-a_{12}x_j^{s-1}) \\ v_j^s = x_j^{s-1}(1-r_2+a_{21}x_i^{s-1}-a_{22}x_j^{s-1}) \end{cases} \quad (8)$$

式中,参与捕食-被食的种群 $i = \text{Rand}(1, N), j = \text{Rand}(1, N), i \neq j$ 。

(4) 融合算子。融合算子描述的是当前种群 i 的某些特性与其它若干个种群的某些特性融合在一起(而余下的特性却保留下来),产生新一代种群,即对于 $i \in \{1, 2, \dots, N\}, s \in \{1, 2, \dots, n\}$, 有

$$v_i^s = \begin{cases} \alpha x_i^{s-1} + \sum_{k=1}^B \beta_k x_{u_k}^{s-1}, & \text{若 } \text{Rand}(0, 1) \leq \mu \\ x_i^{s-1}, & \text{否则} \end{cases} \quad (9)$$

式中, $u_k = \text{Rand}(1, N); \forall k, w \in \{1, 2, \dots, B\}, u_k \neq u_w \neq i; B$ 为与种群 i 发生融合的其他种群数,称为融合种群数, $B \geq 1, \alpha, \beta_k, \mu$ 称为融合选择率,且 $0 < \alpha, \beta_k < 1, 0 < \mu < 1$; 计算时取 $\alpha = \text{Rand}(0, 1), \beta_k = \text{Rand}(0, 1)$ 。

(5) 突变算子。一个 PSI 较低的种群若发生突变,则很有可能突变成更好的种群。对于 $i \in \{N/2, N/2+1, \dots, N\}; s \in \{1, 2, \dots, n\}$, 种群突变算子定义如下:

$$v_i^s = \begin{cases} \sum_{k=1}^{m_I} \gamma_k x_{i_{2(k-1)}^s}^{t-1} - \sum_{k=1}^{m_E} \delta_k x_{i_{2k+1}^s}^{t-1}, & \text{若 } \text{Rand}(0, 1) \leq MR_0 \\ x_i^{s-1}, & \text{否则} \end{cases} \quad (10)$$

式中, MR_0 为突变概率,且 $0 \leq MR_0 \leq 1; \forall k, w \in \{1, 2, \dots, N\}, i_k \neq i_w \neq i; \delta_k, \gamma_k$ 为突变常数, $0 < \delta_k, \gamma_k < 1$, 计算时取 $\delta_k = \text{Rand}(0, 1), \gamma_k = \text{Rand}(0, 1); m_I$ 和 m_E 为参与当前种群突变的其它种群数, M_I 为融入当前种群的其它种群数, M_E 为被挤出的种群数, $m_I \geq 2, m_E \geq 1$ 。

(6) 选择算子。PDO 算法通过竞争算子、互利算子、捕食-被食算子、融合算子和突变算子产生新一代种群之后,采用一对一选择算子将新一代种群与相应的父代种群进行比较,较优者保存到下一代种群中。对于最小化优化问题,其选择算子可以描述为:

$$X_i^t = \begin{cases} V_i^t, & \text{若 } \text{PSI}(V_i^t) > \text{PSI}(X_i^{t-1}) \\ X_i^{t-1}, & \text{其它} \end{cases}, i = 1, 2, \dots, N \quad (11)$$

一旦新的种群形成之后, PDO 算法继续通过竞争、互利、捕食-被食、融合、突变和选择等算子对群落不断进行演化,直到找到最优解。

2.5 高维空间初始点的选择方法

众所周知,对于低维优化问题,进化算法一般都具有很好

的求解性能。但对于高维优化问题,进化算法的求解性能则急剧下降。一个很重要的原因是在高维空间,单位体积内初始解的含量极低,当然很难做到均匀分布。例如,在边长为 10 的 n 维空间立方体中(即 $0 \leq x_i \leq 10, i = 1, 2, \dots, n$),若 100 个初始解分布于其中,则单位体积内的初始解的个数为 $10^{-(n-2)}$ 。在 2 维空间内,单位体积内的初始解的个数为 1 个,而在 1000 维空间内,单位体积内的初始解的个数为 10^{-998} 个,两者相差 10^{1000} 倍。若在 1000 维空间要做到单位体积内含 1 个初始解(即保持与 2 维空间具有相同的搜索效力),则需要 10^{1000} 个初始解。显然,这在工程应用中是不可能实现的。如何让很少的初始解尽可能地代表高维空间的分布特性呢? 正交表^[13]就是一个很好的解决方案。

正交表是遵循概率统计规律,依照一定原则(如正交拉丁方)生成的。通常等水平正交表写成 $L_n(t^n)$, 其中 n 表示正交表的行数或初始解的个数; t 表示正交表同一列中出现的因素的水平数(位级数); n 表示正交表的列数(即优化问题解空间的维数)。正交表之所以可用较少的试验次数来获得到较优的覆盖度,关键在于它的均衡分散性和整齐可比性。正是这种均衡性带来“冒尖”(即优良的组合明显化),这一点在非参数统计中有以下定理 1 保证。

定理 1^[13] 设随机变量 X 遵守连续概率分布 $P(X)$, X_1, X_2, \dots, X_N 为总体 X 的一个容量为 N 的简单样本。把 N 次随机变数 X_1, X_2, \dots, X_N 按照观测值从小到大排列起来, 记作 $X_1^* \leq X_2^* \leq \dots \leq X_N^*$; 便有: (1) $E\{P(X_i^*)\} = \frac{1}{N+1}, i = 1, 2, \dots, N$; (2) 特别有 $E\{P(X_N^*)\} = \frac{N}{N+1}$ 。

命题(2)的含义是,简单样本中最好的结果优于总体占比例 $\frac{N}{N+1}$ 的结果,而劣于总体中占比例 $\frac{1}{N+1}$ 的结果。例如,设优化问题的维数为 500,若在搜索空间内每维离散成 100 个点,则理论上需要初始解 100^{500} 个。若按正交表 $L_{100}(100^{500})$ 安排选择初始解,则仅需 100 个初始解,这 100 个初始解中的最优解优于总体离散点 100^{500} 中占比例 $\frac{100}{100+1} = 99.01\%$ 的点。

假设每个变量的搜索区间为 $[l_i, u_i], i = 1, 2, \dots, n$, 则利用正交拉丁方生成方法产生 N 个初始解的正交表 $L_N(N^n)$ 的构造算法如算法 1 所述。

算法 1 产生 N 个初始解的正交表 $L_N(N^n)$ 的构造算法

Step 1 计算每个变量的离散点 y_{ij} :

$$y_{ij} = l_i + (j-1)(u_i - l_i)/(N-1), i = 1, 2, \dots, n; j = 1, 2, \dots, N$$

Step 2 根据正交拉丁方的生成方法计算初始解:

$$x_{ij} = y_{jk}, i = 1, 2, \dots, N, j = 1, 2, \dots, n$$

式中, $k = (i+j-1) \bmod N$; 若 $k=0$, 则 $k=N$ 。

算法 1 所确定的 N 个初始解 $X_i = (x_{i1}, x_{i2}, \dots, x_{in}) (i = 1, 2, \dots, N)$ 具有很好的均衡分散性和整齐可比性。

2.6 PDO 算法设计

算法 2 PDO 算法

(1) 初始化: a) 令 $t=0$; b) 按表 2 初始化本算法涉及到的所有参数; c) 按算法 1 确定 N 个初始解; d) 对 N 个初始解的从优到劣进行排序,使 PSI 值较高的初始解排在前面。

(2) 执行下列操作:

FOR $t=1$ TO $G//G$ 为演化最大代数

$EI_i = X_i^{t-1}, i=1, 2, \dots, KP//$ 将前 KP 个最佳试探解作为精英保存下来;

FOR $i=1$ TO N

随机选择种群 $j = \text{Rand}(1, N)$, 确保 $i \neq j$;

FOR $s=1$ TO n

IF $p_1 \leq p_A$ THEN// p_A 为当前种群 i 与种群 j 发生竞争行为的概率上限, p_1 为该竞争行为出现的实际概率, $p_1 = \text{Rand}(0, 1)$

按式(6)执行竞争算子, 得到 v_{is}^i 和 v_{js}^j ;

$$\text{计算 } v_{is}^i = \frac{1}{2}(v_{is}^i + v_{js}^j);$$

ELSEIF $p_A < p_2 \leq p_B$ THEN // p_B 为当前种群 i 与种群 j 发生互利行为的概率上限, p_2 为该互利行为出现的实际概率, $p_2 = \text{Rand}(0, 1)$

按式(7)执行互利算子, 得到 v_{is}^i 和 v_{js}^j ;

$$\text{计算 } v_{is}^i = \frac{1}{2}(v_{is}^i + v_{js}^j);$$

ELSEIF $p_B < p_3 \leq p_C$ THEN // p_C 为当前种群 i 与种群 j 发生捕食-被食行为的概率上限, p_3 为该捕食-被食行为出现的实际概率, $p_3 = \text{Rand}(0, 1)$

按式(8)执行互利算子, 得到 v_{is}^i 和 v_{js}^j ;

$$\text{计算 } v_{is}^i = \frac{1}{2}(v_{is}^i + v_{js}^j);$$

ELSEIF $p_B < p_3 \leq p_D$ THEN // p_D 为当前种群 i 与其它种群发生融合行为的概率上限, p_4 为该种群融合行为出现的实际概率

按式(9)执行融合算子, 得到 v_{is}^i ;

ELSE // 执行静止行为, p_E 为当前种群 i 的静止行为的概率上限, $p_E = 1$; p_5 为该种群静止行为出现的实际概率, $p_5 = 1 - (p_1 + p_2 + p_3 + p_4)$

$v_{is}^i = x_{is}^{t-1}$; // 静止行为指的是种群 i 某些特征在演化过程中不发生变化

END IF

END FOR

END FOR

重新计算 N 个新试探解 V_i^t 的 PSI 指数, 对 N 个新试探解的从优到劣进行排序;

按式(10)对较差的后 $N/2$ 个试探解进行突变操作, 得到 $V_i^t, i=1, 2, \dots, N$; // 执行突变操作

按式(11)对所有新获得的 N 个试探解 V_i^t 和原试探解 X_i^{t-1} 进行选择操作, 得到下一代试探解 X_i^t , 其中 $i=1, 2, \dots, N$; // 执行选择操作
根据 N 个新试探解的 PSI 指数, 对 N 个新试探解的从优到劣进行排序。

IF 新得到的全局最优解与最近一次已保存的当前全局最优解之间的误差满足最低要求 ϵ THEN 转(3)

将倒数 KP 个最差的解用保存下来的精英进行替换, 即 $X_{N-k+1}^t = EI_k^{t-1}, k=1, 2, \dots, KP$ 。

END FOR

(3) 结束。

2.7 算法的时间复杂度

PDO 算法(算法 2)的时间复杂度计算过程如表 1 所列, 其时间复杂度与演化次数 G 、种群规模 N 、变量个数 n 、搜索算子的时间复杂度以及其他辅助操作相关。

表 1 PDO 算法时间复杂度计算表

操作	时间复杂度	最多循环次数
初始化	$O(n+4nN+N(N+1))$	1
迭代准备	$O(nKP)$	G
竞争算子	$O(2n)$	GN
互利算子	$O(2n)$	GN
捕食-被食算子	$O(2n)$	GN
融合算子	$O(8nB)$	GN
目标函数计算	$O(n^2)$	GN
结果排序	$O(4n+(N+1))$	GN
突变算子	$O(3m_i+4nm_E)$	GN
选择算子	$O(3n)$	GN
结果排序	$O(4n+(N+1))$	GN
精英替换	$O(nKP)$	G
结果输出	$O(n)$	1

3 PDO 算法的全局收敛性证明

由 PDO 算法(算法 2)知, 种群 S 是一个离散的空间, 但每个种群 $X_i^t (i=1, 2, \dots, N)$ 是在连续的实数空间取值。种群总个数为 N , 每个种群即为优化问题(1)的一个候选解, 其目标函数值为 $F(X_i^t)$ (按式(2)计算), 则所有种群的状态所形成的集合为:

$$F = \{F(X_i^t) | X_i^t \in S\}$$

进一步令

$$F = \{F_1, F_2, \dots, F_N\}, F_1 \leq F_2 \leq \dots \leq F_N$$

$\forall X \in S$ 有 $F_1 \leq F(X) \leq F_N$, 将 S 划分为非空子集为:

$$X_i^s = \{X | X \in S \text{ 且 } F(X) = F_i\}, i=1, 2, \dots, N$$

$$\sum_{i=1}^N |X_i^s| = N; \forall i \in \{1, 2, \dots, N\}, X_i^s \neq \emptyset$$

$$\forall i \neq j, X_i^s \cap X_j^s = \emptyset, \bigcup_{i=1}^N X_i^s = S$$

令 $X^{i,j} (i=1, 2, \dots, N; j=1, 2, \dots, |X_i^s|)$ 表示 X_i^s 中第 j 个状态。种群在进化过程中, 从一个状态转移到另外的状态可表示为 $X^{i,j} \rightarrow X^{k,l}$, 则从 $X^{i,j}$ 到 $X^{k,l}$ 的转移概率为 $p_{ij,kl}$, 从 $X^{i,j}$ 到 X_i^s 中任一状态的转移概率为 $p_{ij,k}$, 从 X_i^s 中任一状态到 X_i^s 中任一状态的转移概率为 $p_{i,k}$, 则有:

$$p_{ij,k} = \sum_{l=1}^{|X_i^s|} p_{ij,kl}, \sum_{k=1}^N p_{ij,k} = 1, p_{i,k} \geq p_{ij,k}$$

$$p_{i,k} \geq p_{ij,k} \rightarrow \sum_{k=1}^N p_{i,k} \geq \sum_{k=1}^N p_{ij,k} = 1$$

而 $0 \leq \sum_{k=1}^N p_{i,k} \leq 1$, 故有

$$\sum_{k=1}^N p_{i,k} = 1$$

引理 1 在 PDO 算法(算法 2)中, $\forall X^{i,j} \in X_i^s, i=1, 2, \dots, N; j=1, 2, \dots, |X_i^s|$ 满足:

$$\forall k > i, p_{i,k} = 0 \quad (12)$$

$$\exists k < i, p_{i,k} > 0 \quad (13)$$

(1)式(12)的证明。设 $X^{i,j}$ 为第 t 次演化后的种群的状态, 记为 Y^t , 设 Y^t 中状态最好的种群为 $gBest^t = X^*$, 即有 $F(gBest^t) = F_i$ 。通过 PDO 算法在每次演化过程中对当前种群最好状态的更新可知:

$$F(X^{t+1}) \leq F(X^t) \Rightarrow \forall k > i, p_{ij,kl} = 0 \Rightarrow \forall k > i, p_{ij,k} = \sum_{l=1}^{|X_i^s|} p_{ij,kl}$$

$$p_{ij,kl} = 0 \Rightarrow \forall k > i, p_{i,k} = 0$$

(2)式(13)的证明。在PDO化算法中,设 $gBest^{t+1}$ 为 Y^{t+1} 中最优种群的状态,此时种群随机选择活动行为,有竞争行为、互利行为、捕食-被食行为、融合行为和静止行为5种情况。

情形1 当前种群 i 选择竞争行为。若种群 i 的状态可以转移到更优的状态上,则其将以概率 $P_1 = p_1 p_M p_S + p_1 (1 - p_M) p_S = p_1 p_S$ 更新到新状态,显然 $P_1 > 0$,命题得证。其中, p_M 为突变算子执行成功的概率(即被突变了); p_S 为选择算子执行成功的概率(即被选中了)。

情形2 当前种群 i 选择互利行为。若种群 i 的状态可以转移到更优的状态上,则其将以概率 $P_2 = p_2 p_M p_S + p_2 (1 - p_M) p_S = p_2 p_S$ 更新到新状态,显然 $P_2 > 0$,命题得证。

情形3 当前种群 i 选择捕食-被食行为。若种群 i 的状态可以转移到更优的状态上,则其将以概率 $P_3 = p_3 p_M p_S + p_3 (1 - p_M) p_S = p_3 p_S$ 更新到新状态,显然 $P_3 > 0$,命题得证。

情形4 当前种群 i 选择融合行为。若种群 i 的状态可以转移到更优的状态上,则其将以概率 $P_4 = p_4 p_M p_S + p_4 (1 - p_M) p_S = p_4 p_S$ 更新到新状态,显然 $P_4 > 0$,命题得证。

情形5 当前种群 i 选择静止行为。若种群 i 的状态可以转移到更优的状态上,则其将以概率 $P_5 = p_5 p_M p_S + p_5 (1 - p_M) p_S = p_5 p_S$ 更新到新状态,显然 $P_5 > 0$,命题得证。

在任意一个时间段内,任一种群 i 必选上述5种行为之一,即 $p_1 + p_2 + p_3 + p_4 + p_5 = 1$ 。综合上述各种情况,可得总概率为:

$$P_d = P_1 + P_2 + P_3 + P_4 = (p_1 + p_2 + p_3 + p_4) p_S = p_S$$

因 $0 \leq p_S \leq 1$,易知 $P_d \geq 0$;如果 $P_d > 0$,命题得证;如果 $P_d = 0$,说明 $p_S = 0$,即 $F(X_i^t) > F(X_i^{t-1})$ 出现的概率永为0,表明种群演化已经到达全局最优解状态 $gBest^{t+1} = X^*$,而不是局部最优解。

综合上述情况可得 $\exists k < i, p_{i,k} > 0$,证毕。

定理2^[14] 设 P' 是一 n 阶可归约随机矩阵,也就是通过相同的行变换和列变换后可以得到 $P' = \begin{bmatrix} C & \cdots & 0 \\ R & \cdots & T \end{bmatrix}$,其中 C 是 m 阶本原随机矩阵并且 $R \neq 0, T \neq 0$,则有:

$$P'^{\infty} = \lim_{k \rightarrow \infty} P'^k = \lim_{k \rightarrow \infty} \begin{bmatrix} C^k & \cdots & 0 \\ \sum_{i=1}^{k-1} T^i R C^{k-i} & \cdots & T^k \end{bmatrix} = \begin{bmatrix} C^{\infty} & \cdots & 0 \\ R^{\infty} & \cdots & 0 \end{bmatrix}$$

上述的矩阵是一个稳定的随机矩阵且 $P'^{\infty} = 1' P'^{\infty}$, $P'^{\infty} = P'^0 P'^{\infty}$ 唯一确定并且与初始分布无关。 P'^{∞} 满足如下条件:

$$P'^{\infty} = [p'_{ij}]_{n \times n}, \begin{cases} p'_{ij} > 0, 1 \leq i \leq n, 1 \leq j \leq m \\ p'_{ij} = 0, 1 \leq i \leq n, m < j \leq n \end{cases}$$

定理2的证明过程非常复杂,具体证明过程可参见文献[14]。

定理3 PDO算法(算法2)具有全局收敛性。

证明:对于每个 $X_i^s, i = 1, 2, \dots, N$,可将其看为是有限

Markov链上的一个状态,根据引理1中式(12)的结论可得,该Markov链的转移矩阵为:

$$P' = \begin{bmatrix} p_{1,1} & 0 & \cdots & 0 \\ p_{2,1} & p_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ p_{N,1} & p_{N,2} & \cdots & p_{N,N} \end{bmatrix} = \begin{bmatrix} C & 0 \\ R & T \end{bmatrix}$$

根据引理1中式(13)的结论得:

$$p_{2,1} > 0, R = (p_{2,1}, p_{3,1}, p_{4,1}, \dots, p_{N,1})^T$$

$$T = \begin{bmatrix} p_{2,2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ p_{N,2} & \cdots & p_{N,N} \end{bmatrix} \neq 0, C = (p_{1,1}) = (1) \neq 0$$

由以上可知,转移矩阵 P' 是 N 阶可归约随机矩阵,满足定理2的条件,所以下式成立:

$$P'^{\infty} = \lim_{k \rightarrow \infty} \begin{bmatrix} C^k & \cdots & 0 \\ \sum_{i=1}^{k-1} T^i R C^{k-i} & \cdots & T^k \end{bmatrix} = \begin{bmatrix} C^{\infty} & \cdots & 0 \\ R^{\infty} & \cdots & 0 \end{bmatrix}$$

因 $C^{\infty} = C = (1), T^{\infty} = 0$,故必有 $R^{\infty} = (1, 1, \dots, 1)^T$,这是因为Markov转移矩阵 P' 中每行的概率之和为1。因此有 P'^{∞}

$$= \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}, \text{且为稳定的随机矩阵。}$$

上式表明,当 $k \rightarrow \infty$ 时,概率 $p_{i,1} = 1, i = 1, 2, \dots, N$,也即无论初始状态如何,最后都能以概率1收敛到全局最优状态1上。于是得:

$$\lim_{t \rightarrow \infty} p\{F(X_i^t) \rightarrow F(X^*)\} = 1, i = 1, 2, \dots, N$$

因此,PDO化算法具有全局收敛性,证毕。

定理3表明:

(1)若随机搜索的每个状态都能以概率 $p > 0$ 转移到一个比当前状态更优的状态上,则该随机搜索收敛,即搜索几乎以概率1找到全局或局部最优解。概率 p 越大,收敛越快。

(2)随机搜索的某个状态若无法以概率 $p > 0$ 转移到一个比当前状态更优的状态上,则宁愿保持当前状态不变也不要转移到比当前状态更差的状态上。若该情况经常出现,则收敛过程很慢。

(3)若随机搜索的状态以同样的概率转移到一个比当前状态更好、保持不变或更差的状态上(即下一个状态是好是坏或保持不变随机出现),则该随机搜索发散,即该随机搜索找到全局或局部最优解的概率几乎为0。文献[15]用很复杂的理论证明了粒子群算法不收敛,就是这个原因。

顺便指出,当随机搜索陷入局部最优解状态时,很多随机搜索算法采用所谓的逃逸方法试图跳出局部最优解状态。然而,若跳出后的状态比局部最优解状态差,这种情况几乎以很高的概率出现,则该随机搜索很难收敛。因此,逃逸策略的设计应高度谨慎。

定理3并没有告诉我们如何提高随机搜索中每个状态向更好的状态转移的概率,该问题需要结合具体优化问题的特性来解决。对于优化问题(1),若当前搜索位置位于空间点

X_0 , 文献[16]介绍的非线性优化方法告诉我们:最快的搜索方向是目标函数在 X_0 点的负梯度方向,非线性优化方法以最快的方式实现步步高,快速达到最优点(但该最优点可能是局部最优点、稳定点或全局最优点),这与定理3的结论不谋而合;随机搜索虽然无法找到最快搜索方法,但只要做到步步高(至少应是步步高不差),依然能收敛到全局最优点。

4 例子研究与对比分析

算法2所用的测试例子选自文献[17]提供的 Ackley 测试函数和至今未发现其理论最优解的 BUMP 函数。

(1) n 维 Ackley 函数。该函数在搜索空间中存在大量局部极小值点,当 $x_i=0 (i=1, 2, \dots, n)$ 时达到全局最小值 0。当 $n=2$ 时,该函数的图形如图 1 所示,对于高维 Ackley 函数,其局部极小值的分布情况更是无法想象。

$$\min f_1(\mathbf{X}) = 20 + e - 20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right), -600 \leq x_i \leq 600, i=1, 2, \dots, n$$

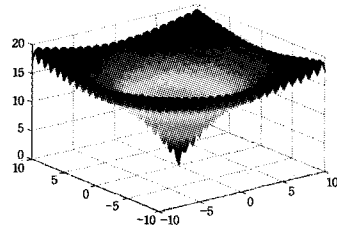


图1 当 $n=2$ 时 Ackley 函数的三维图形

为了让 PDO 算法能对上述优化问题求解,令 $x_i = y_i - 600$, 则

$$\min f_1(\mathbf{Y}) = 20 + e - 20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - 600)^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi(y_i - 600))\right), 0 \leq y_i \leq 1200, i=1, 2, \dots, n$$

式中, $\mathbf{Y} = (y_1, y_2, \dots, y_n)$ 。

(2) BUMP 函数,全局理论最优值至今未知,随着维数的增加,该函数极难优化,而且约束条件 $\prod_{i=1}^n x_i \geq 0.75$ 中的 $\prod_{i=1}^n x_i$ 容易溢出。当 $n=2$ 时,该函数的图形如图 2 所示。

$$\min f_2(\mathbf{X}) = -\left|\sum_{i=1}^n \cos^4(x_i) - 2 \prod_{i=1}^n \cos^2(x_i)\right| / \sqrt{\sum_{i=1}^n x_i^2}$$

s. t. $\prod_{i=1}^n x_i \geq 0.75, \sum_{i=1}^n x_i \leq 7.5n,$
 $0 < x_i \leq 10, i=1, 2, \dots, n$

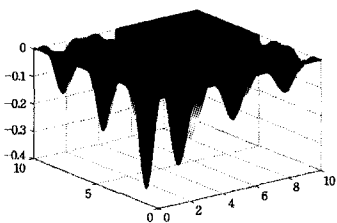


图2 当 $n=2$ 时 BUMP 函数的三维图形

优化问题 $f_1(\mathbf{X})$ 和 $f_2(\mathbf{X})$ 求解过程中的取值如表 2 所

列。计算时,表 2 中各参数取值方法及其依据如下所述。

表 2 求解过程各参数取值

参数名	取值
演化次数 G	$G=20000$
变量数 n	对于优化问题 $f_1(\mathbf{X})$: $n=100, 200, 400, 600,$ 800, 1000, 1200; 对于优化问题 $f_2(\mathbf{X})$: $n=20, 30, 50, 100, 1000$
种群数 N	$N=500$
融入和挤出种群数 m_I, m_E	$m_I=2, m_E=1$
融合种群数 B	$B=3$
竞争行为出现概率上限 p_A	$p_A=1/5$
互利行为出现概率上限 p_B	$p_B=2/5$
捕食-被食行为出现概率上限 p_C	$p_C=3/5$
融合行为出现概率上限 p_D	$p_D=4/5$
静止行为出现概率上限 p_E	$p_E=1$
突变概率 MR_0	$MR_0=0.3$
融合选择率 μ	$\mu=0.3$
精英保存个数 KP	$KP=2$
Lotka-Volterra 模型参数 r_i, a_{ij}	$r_1=r_2=0.1, a_{12}=a_{21}=0.3,$ $a_{11}=a_{22}=0.5$

(1) 尽管 N 取较大值可扩大搜索空间,但由表 1 知,算法总时间复杂度与 N 成正比,因此, N 不能取得太大,该参数的取值无需太高的精确性,只需要依据具体的优化问题和计算机的速度而定,种群数 $N=500$ 即合适。

(2) Lotka-Volterra 模型参数 r_i, a_{ij} , 其取值依据可参见文献[18]。

(3) 对于当前某个种群来说,其融入的种群数 m_I 必须大于被挤出的种群数 m_E , 否则该种群的规模会越来越小,直至消亡;但由表 1 知, m_I 和 m_E 太大会影响计算速度,因此 m_I 取最小值 2, m_E 取最小值 1 即可。

(4) 为确保种群多样性,融合种群数 B 不能太小,但由表 1 知, B 太大会影响计算速度,因此 $B=3$ 即可。

(5) 突变概率 MR_0 , 融合选择率 μ , 精英保存个数 KP , 其取值依据可参见文献[11]。

(6) 对于竞争行为、互利行为、捕食-被食行为、融合行为和静止行为的出现概率上限 p_A, p_B, p_C, p_D 和 p_E 按各行为出现机会均要求等设置,即竞争行为出现概率区间为 $0 \sim 1/5$, 互利行为出现概率区间为 $1/5 \sim 2/5$, 捕食-被食行为出现概率区间为 $2/5 \sim 3/5$, 融合行为出现概率区间为 $3/5 \sim 4/5$, 静止行为出现概率区间为 $4/5 \sim 1$ 。

(7) 迭代误差 $\epsilon = 1 \times 10^{-8}$, 其取值依据是我们希望获得的最优解的精度高低。最大迭代代数 $G=20000$, 其取值依据是防止迭代过程不满足收敛条件 ϵ 时出现无限迭代。

必须指出,上述参数的取值无需太高的精确性,取值不合适仅影响收敛速度,对全局最优解的精度没有影响。

优化问题 $f_1(\mathbf{Y})$ 求解结果及其与 BBO 算法[11]的对比结果如表 3 所列。

表 3 PDO 算法与 BBO 算法的测试效果对比

维数(n)	PDO 算法		BBO 算法	
	目标函数值	耗时(h)	目标函数值	耗时(h)
100	1.56823923082819E-15	0.32	0.5176	0.47
200	3.00741092006394E-15	0.41	1.2518	0.52
400	5.58467931024977E-15	0.53	6.1287	0.84
600	7.44089209850063E-15	0.59	22.8912	1.13
800	6.12768572343908E-14	0.64	34.1629	3.87
1000	7.43449147817038E-14	0.74	56.0127	4.26
1200	1.08420217248553E-13	0.86	78.7812	6.48

对于 BUMP 问题 $f_2(X)$, 运用本文求解算法进行求解, 计算结果如表 4 所列。

表 4 PDO 算法与 BBO 算法的 BUMP 函数测试效果对比

维数 n	目标函数值(PDO 算法)	目标函数值(BBO 算法)	耗时(h)
20	-0.803605881101687	-0.4513385612369127	2.3
30	-0.821789549623916	-0.4781292382376343	2.7
50	-0.832132927828923	-0.5191276301252371	3.4
100	-0.844349528432361	-0.5512423812523893	4.6
1000	-0.843128203982247	-0.6572382712651393	7.3

从表 3 可以看出, 对于普通函数优化问题, 随着维数的增加, PDO 算法要越来越优于 BBO 算法, 而且随着维数的增加, 这种优势越来越明显; 从表 4 可以看出, 对于极难优化的 BUMP 问题, PDO 算法获得最优解的精度要明显高于 BBO 算法。

结束语 本文运用种群动力学理论构造出了可求解大规模优化问题的优化算法, 算法中运用种群动力学基本模型提出了竞争、互利、捕食-被食、融合、突变和选择等算子, 运用正交拉丁方生成方法提出了试探解的初始化方法, 确保了初始试探解具有很好的均衡分散性和整齐可比性。这些方法的运用大幅提升了 PDO 算法的性能, 该 PDO 算法具有全局收敛性, 且适用于求解高维优化问题。今后需要进一步研究的方向是:

- (1) 如何将两个种群间的竞争、互利、捕食-被食行为扩展为多个种群;
- (2) 如何深入分析 PDO 算法的动态行为;
- (3) 如何将更多、更反映种群进化实际情况的种群动力学理论用于改善或构造出新的种群动力学优化算法。

参 考 文 献

[1] 王顺庆, 王万雄, 徐海根. 生态学稳定性理论与方法[M]. 北京: 科学出版社, 2004
 [2] 陆征一, 周义仓. 数学生物学进展[M]. 北京: 科学出版社, 2006

(上接第 264 页)

参 考 文 献

[1] Pawalk Z. Rough Sets [J]. International Journal of Computer and Information Science, 1982, 11(5): 341-356
 [2] Jensen R, Shen Q. Fuzzy-rough sets assisted attribute selection [J]. IEEE Transactions on Fuzzy Systems, 2007, 15(1): 73-89
 [3] Skowron A, Swiniarski R, Synak P. Approximation Spaces and Information Granulation[C]// Proc of the International Conference on Rough Sets and Current Trends in Computing. Uppsala, Sweden, 2004: 116-126
 [4] Guan J W, Bell D A, Guan Z. Matrix Computation for Information System[J]. Information Sciences, 2001, 131(1-4): 129-156
 [5] 杨勇. 粗糙集的矩阵定义[J]. 计算机工程与应用, 2007, 43(14): 1-2

[3] 李长生. 一类 n 维 Lotka-Volterra 系统稳定平衡点的判别[J]. 潍坊学院学报, 2003, 3(6): 1-2
 [4] 程惠东, 孟新柱, 王芳. 一类时滞非自治 Lotka-Volterra 扩散生态系统的全球吸引性[J]. 中山大学学报: 自然科学版, 2008, 47(2): 19-22
 [5] 周桦, 刘佳. 带扩散的具有 Holling-IV 类功能性反应的捕食模型的性质[J]. 南京工业大学学报, 2007, 29(5): 66-69
 [6] 吴兴杰. 具有 Holling IV 功能反应的脉冲食物链模型的动力学性质[J]. 桂林电子科技大学学报, 2008, 28(03): 248-253
 [7] 郑宝剑, 林怡平. 一类具有时滞 Holling-IV 型捕食-食饵系统的 Hopf 分支[J]. 云南民族大学学报: 自然科学版, 2007, 16(01): 18-21
 [8] 赵延忠. 具有反应扩散的 Leslie 两种群生物竞争模型的稳定性[J]. 青海大学学报, 2007, 25(5): 62-65
 [9] 庞国萍, 陈兰荪. 具有脉冲效应和 Holling IV 功能性反应的捕食者食饵系统分析[J]. 南京师大学报: 自然科学版, 2007, 30(2): 1-5
 [10] 田灿荣. 一类带时滞竞争模型的周期解[J]. 生物数学学报, 2007, 22(3): 431-440
 [11] Simon D. Biogeography-based Optimization[J]. IEEE Transactions; Evolutionary Computation, 2008, 12(6): 702-713
 [12] 王存睿, 王楠楠, 段晓东, 等. 生物地理学优化算法综述[J]. 计算机科学, 2010, 37(7): 34-38
 [13] 中国现场统计研究会三次设计组. 正交法和三次设计[M]. 北京: 科学出版社, 1987
 [14] Iisufescu M. Finite Markov Processes and Their Applications [M]. Wiley; Chichester, 1980
 [15] 任子晖, 王坚, 高岳林. 马尔科夫链的粒子群优化算法全局收敛性分析[J]. 控制理论与应用, 2011, 28(4): 462-466
 [16] 王宜举, 修乃华. 非线性优化理论[M]. 北京: 科学出版社, 2012
 [17] 崔志华, 曾建潮. 微粒群优化算法[M]. 北京: 科学出版社, 2011
 [18] 陈兰荪, 孟新柱, 焦建军. 生物动力学[M]. 北京: 科学出版社, 2009

[6] Liu Gui-long. The Axiomatization of Rough Set Upper Approximation Operation[J]. Fundamenta Informaticae, 2006, 69(3): 331-342
 [7] 王磊, 等. 基于矩阵的粗糙集上下近似的计算方法[J]. 模式识别与人工智能, 2011, 24(6): 757-762
 [8] 刘清. 粗糙集及 Rough 推理[M]. 北京: 科学出版社, 2001: 60-70
 [9] 陈昊, 等. 变精度粗糙集的属性核和最小属性约简算法[J]. 计算机学报, 2012, 35(5): 1012-1016
 [10] Cheng Y. The incremental method for fast computing the rough fuzzy approximations[J]. Data & Knowledge Engineering, 2011, 70: 84-100
 [11] Li T R, Ruan D, Gerret W, et al. A rough set based characteristic relation approach for dynamic attribute generalization in data mining[J]. Knowledge-based Systems, 2007, 20(5): 485-494