

基于改进型进化机制的萤火虫优化算法

符强 童楠 钟才明 赵一鸣
(宁波大学科学技术学院 宁波 315212)

摘要 分析了萤火虫算法的进化计算机制,并利用实例对萤火虫算法中容易发生进化过早停滞的原因进行了研究。提出了一种基于新型进化计算模式的改进型萤火虫优化算法,该算法在进化初期利用种群最优萤火虫激发群中其他个体的寻优能力,在萤火虫相互之间建构了有效的信息交互网络后,各萤火虫将借助各自视觉范围内的更优近邻个体完成后期搜索和进化,当种群陷入局部最优区域时,利用高斯变异改善萤火虫个体的多样性。利用标准测试函数进行了实验分析,结果表明,改进后的萤火虫算法能有效改善过早进化停滞问题。

关键词 萤火虫算法,群智能,进化机制,高斯变异

中图分类号 TP183 **文献标识码** A

Firefly Algorithm Based on Improved Evolutionism

FU Qiang TONG Nan ZHONG Cai-ming ZHAO Yi-ming
(College of Science and Technology, Ningbo University, Ningbo 315212, China)

Abstract Analyzing the evolutionary computation mechanism of the Firefly algorithm, a new evolutionary computation model for Firefly algorithm was proposed to solve the evolutionary premature stagnation problem. At the beginning, the fireflies achieve evolution by following the best firefly in global area, and when the mutual system is established among the fireflies for exchanging the information, each firefly is attracted by the brighter glow of other neighboring fireflies. When the population is in local optimization area, Gaussian mutation is used to improve firefly's diversity. The experiment results of 5 classic benchmark functions indicate the feasibility and validity of the improved Firefly algorithm.

Keywords Firefly algorithm (FA), Swarm intelligence, Evolutionism, Gaussian mutation

1 引言

借鉴萤火虫利用发光来传递信息的生物特性, Yang^[1]提出了一种新型的群智能算法——萤火虫算法 (Firefly Algorithm)。萤火虫算法本质上是一个随机优化算法,因结构简单,需要调节的参数较少,同时具有较好的寻优搜索能力而引起国内外学者的关注,并广泛应用于优化问题^[2,3]、软件测试^[4]、价格预测^[5]、网络可靠性分析^[6]、自动控制^[7]、图像压缩^[8]、聚类^[9]、组合问题^[10,11]等多个领域。目前国内针对萤火虫算法的性能优化的研究还处于初期阶段,其中 Gandomi^[12]提出了结合混沌的优化方法,其能更为有效地提高萤火虫算法的全局优化能力,取得了一定成果。

萤火虫算法具有良好的寻优能力,但是在求解高维复杂优化问题,或者求解区间过大时,容易产生萤火虫个体进化过早停滞的现象。本文通过分析萤火虫算法的进化计算机制,指出了该算法易于出现过早收敛现象的原因,同时提出了一种新的进化计算模式,其能够在保留算法原有进化优势的同时,有效改善和平衡萤火虫算法的全局与局部搜索性能。文

中还利用典型优化函数进行了性能测试。

2 常规萤火虫算法 (FA)

2.1 算法描述

在常规萤火虫算法中,多个萤火虫随机散布于整个搜索空间内,所有萤火虫都具有一个由被优化问题决定的适应度值,对应于该萤火虫的荧光亮度。然后每个萤火虫个体都在其视觉范围内追随荧光亮度更强的萤火虫飞行,经过多次群体运动后,所有个体将聚集在荧光亮度最强的萤火虫附近,完成最终寻优。在此寻优过程中,荧光亮度和吸引度是实现萤火虫群体进化的两个关键要素。

萤火虫的荧光亮度由其个体所在位置决定,位置越好则荧光亮度越强,进而对其领域内的其他萤火虫的吸引度越强。

假设萤火虫种群中的萤火虫总数为 m , 搜索空间为 N 维, 其中第 i 只萤火虫在 N 维空间的位置可表示为 $x_i = (x_{i1}, x_{i2}, \dots, x_{iN})$ 。

定义 1 第 i 只萤火虫的最大荧光亮度

$$I_{0i} = f(x_i) \tag{1}$$

到稿日期:2013-03-23 返修日期:2013-06-17 本文受浙江省教育厅科研项目(Y201326872),浙江省自然科学基金项目(Y1090851),宁波大学科研基金项目(XYL12009),十二五浙江省重点学科建设项目(计算机应用技术)资助。

符强(1975—),男,硕士,讲师,主要研究方向为智能控制与算法优化、人工智能技术, E-mail: fuqiang@nbu.edu.cn;童楠(1981—),女,硕士,讲师,主要研究方向为智能控制与算法优化、数据挖掘技术;钟才明(1969—),男,博士,副教授,主要研究方向为模式识别、数据挖掘;赵一鸣(1958—),男,教授,主要研究方向为图像识别、智能虚拟技术。

式中, $f(x_i)$ 为第 i 只萤火虫所在位置对应的目标适应度值。

定义 2 第 i 只萤火虫的相对荧光亮度

$$I_i = I_{0i} * e^{-\gamma r_{ij}} \quad (2)$$

式中, γ 为光强吸收因子, 表示萤火虫的荧光亮度受传播介质的影响而逐渐变化的特性。取值与求解域的区间有关, 一般设为常数。

$r_{ij} = \|x_i - x_j\|$, 表示第 i 只萤火虫与第 j 只萤火虫之间的空间距离。

定义 3 第 i 只萤火虫的吸引度

$$\beta_i = \beta_{0i} * e^{-r_{ij}^2} \quad (3)$$

式中, β_{0i} 为最大吸引度因子, 表示最大荧光亮度处位置的吸引度大小。式(3)描述了萤火虫发出的荧光对其他个体的吸引随着距离增大和传播介质的吸收而逐渐减小的特性。

则寻优过程中第 i 只萤火虫的位置进化可由式(4)表示:

$$x_i(t+1) = x_i(t) + \beta * (x_j(t) - x_i(t)) + \alpha * (rand - 1/2) \quad (4)$$

式中, α 为步长因子, 取值为 $[0, 1]$ 内的常数; $rand$ 为 $[0, 1]$ 内服从均匀分布的随机因子; 扰动项 $\alpha(rand - 1/2)$ 能有效扩大萤火虫的搜索范围, 防止萤火虫过早陷入局部最优。群体中的第 i 只萤火虫将在其视觉范围内不断搜索荧光亮度 I 优于自身的萤火虫, 当发现第 j 只萤火虫满足条件时, 则受其吸引, 根据式(4)实现自身的位置优化。若萤火虫的荧光亮度都一样, 则萤火虫各自随机移动。萤火虫群体在多次进行位置进化后, 最终所有个体都将聚集在荧光亮度最强的萤火虫附近, 完成寻优。

2.2 优化机制分析

萤火虫算法的基本进化机制在于, 每个萤火虫个体都在其视觉范围内追随荧光亮度更强的萤火虫飞行, 经过多次群体运动后, 所有个体将聚集在荧光亮度最强的萤火虫附近, 完成最终寻优。

然而从式(4)中可以看到, 能对萤火虫个体起到牵引作用的更优萤火虫不但应当具有更高的荧光亮度, 同时还必须具备一个非常重要的特征, 即与被牵引萤火虫的距离不能过大, 否则会因为荧光强度衰减而失去对其他萤火虫个体的影响力。

下面以四峰函数(如式(5)所示)为例来分析萤火虫算法的寻优过程。

$$f(x) = e^{-(x-4)^2 - (y-4)^2} + e^{-(x+4)^2 - (y-4)^2} + 2e^{-x^2 - (y+4)^2} + 2e^{-x^2 - y^2} \quad (5)$$

如图 1 所示, 四峰函数为二维多峰值函数, 4 个极大峰值均位于 $[-5, 5]$ 区间, 其中理论最大值为 2。

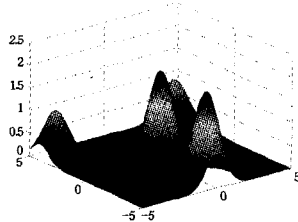


图 1 四峰函数

我们利用萤火虫算法对四峰函数组织了二组寻优实验, 如图 2 所示。其中, 第一组实验选取的求解区间为 $[-5, 5]$, 第二组实验选取的求解区间为 $[-10, 10]$ 。两组实验的其他

条件均相同。

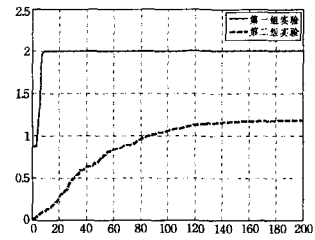


图 2 第一组实验与第二组实验对比

可以看出, 当在 $[-5, 5]$ 区间内寻优时, 萤火虫算法能快速有效地捕捉到全局最优值; 而将求解区间放大到 $[-10, 10]$ 区间时, 萤火虫算法进化速度明显放慢, 并且最终发生早熟收敛现象。我们利用四峰函数的 contour 图对两组实验条件下的萤火虫分布变化做了进一步分析, 结果如图 3、图 4 所示。

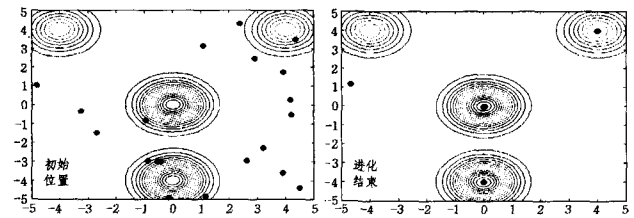


图 3 第一组实验 $[-5, 5]$ 区间中萤火虫分布变化图

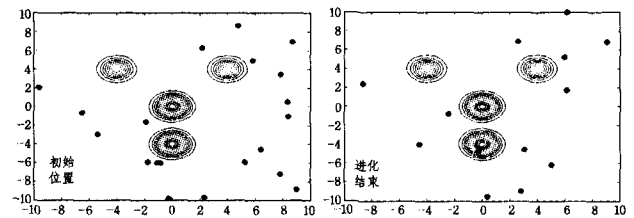


图 4 第二组实验 $[-10, 10]$ 区间中萤火虫分布变化图

由图 3、图 4 可以发现: 在 $[-5, 5]$ 区间内寻优时, 由于萤火虫个体相互之间距离较近, 因此萤火虫个体间的信息交流较充分, 所有个体在相互学习中迅速完成进化过程, 找到全局最优目标; 而在 $[-10, 10]$ 区间内寻优时, 只有相互邻近的部分萤火虫能够进行信息传递和进化操作, 而距离较远的萤火虫难于获取其他个体的信息, 只能在初始位置波动, 在进化初期就已经失去了进化能力, 成为“失效”个体。

另外由式(3)、式(4)也可以看出, 吸引度 β 随距离 r_{ij} 的增大而快速衰减, 因此种群中的萤火虫个体在分布疏散的情况下, 较难找到能够吸引自己的萤火虫, 而只能在原来的位置随机振荡。

可见, 只有当萤火虫个体之间具有足够密切的消息传递能力的前提下, 萤火虫算法才能快速有效地获取全局寻优目标。而当萤火虫的个数不够多, 但需分布区域较大时, 萤火虫个体之间就难于组织充分畅通的信息交流, 从而易陷入进化停滞状态。

3 基于新型进化机制的萤火虫优化算法(IEFA)

实现群体智能的个体应当在所处环境中体现出自主性、反应性、学习性和自适应性等智能特性, 并在简单个体之间构建行之有效的合作进化关系以实现寻优要求。萤火虫算法只能在相互邻近的萤火虫个体间组织有效的信息交互, 从而容易过早出现进化停滞现象。在现有进化计算模式下, 如要完

成全局寻优,则必须配置大量的萤火虫才能满足要求,大大增加了算法的计算时间。

从式(4)所示的萤火虫个体进化计算模式中可以看出,萤火虫个体的进化过程主要依靠邻近具有更高荧光强度的萤火虫提供的牵引作用,吸引力 β 与萤火虫个体间的距离平方呈指数衰减关系,这能帮助萤火虫种群在充分进行个体间消息交流时迅速准确地执行进化任务。从拓扑结构来看,萤火虫算法利用局部模型建立了自身的寻优模式。全局模型利用整个群体的最优值为群中所有个体指引进化方向,往往容易陷入局部最优;而在局部模型中,群中萤火虫跟随拓扑邻近个体当中的局部更优值来寻找优化目标。与全局模型中只具有单一吸引子的模式不同,局部模型中提供了多个吸引子,在一定程度上提高了算法的鲁棒性,更利于满足全局寻优的要求。但当搜索区域较大,且萤火虫个体间的距离较远时,萤火虫之间的信息交流就容易受阻,导致进化过慢,甚至停止。

为此我们对萤火虫算法的进化机制进行了调整,提出了一种既能保持原有算法中的进化优势,又能适当提高个体间交互沟通行为的新颖萤火虫优化算法。具体改进措施有以下几点。

3.1 新型进化计算模式

为提高种群中萤火虫个体的活动能力,加强它们之间的信息往来,我们增加了种群最优值对群中个体提供的牵引作用。新型进化计算模式如式(6)所示:

$$x_i(t+1) = x_i(t) + \omega * rand * (x_{best}(t) - x_i(t)) + \beta * (x_j(t) - x_i(t)) + \alpha * (rand - 1/2) \quad (6)$$

$\omega * rand * (x_{best}(t) - x_i(t))$ 体现了当前种群最优个体对其他萤火虫个体的影响度,其中, $rand$ 为(0,1)区间的随机量, ω 为吸引权重。

与式(4)相比,式(6)中增加了种群最优萤火虫对群中所有其他个体产生的吸引力。这样,在进化初期萤火虫分布过于稀疏时,每个萤火虫依然能接收到有效的社会信息,有效避免了因与邻近萤火虫距离过远而成为“失效萤火虫”的可能。同时,式(6)提出的改进型进化计算模式在原有吸引子的基础上,添加了新的全局吸引子,促进了萤火虫种群进化过程的多样性,在寻优初期能高效搜索较大的求解区间,较快地定位最优解的大致位置。

为了平衡全局和局部搜索能力的双向要求,我们利用 $rand$ 来进一步增加萤火虫个体进化的多样性,同时设置了随迭代次数增加而线性减小的吸引权重 ω ;在进化初期(算法运行初期)设置较大的 ω ,有利于更好地完成全局寻优的任务;而随着迭代的不断推进,逐步减小 ω 的取值,以达到在算法运行后期能快速收敛的目的。另外,逐步减小 ω 的取值,还能有效平衡种群最优值对其他个体产生的影响,既能在开始进化时调动所有萤火虫进入寻优状态,同时又能避免因种群最佳萤火虫对其他个体吸引过大而造成的早熟收敛现象。

3.2 高斯变异操作

若萤火虫种群在连续5次迭代中都没有发生进化,则判断其已经陷入局部最优区域。为帮助萤火虫种群跳出停滞状态,恢复进化能力,本文采用高斯变异因子对萤火虫种群进行扰动。高斯分布是一种工程中常用的重要概率分布,其概率

密度函数如式(7)所示:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, -\infty < x < \infty \quad (7)$$

式中, σ 为高斯分布的方差, μ 为期望。

将所有萤火虫按照适应度大小排序,利用最优萤火虫将排名最后的萤火虫群体进行状态替换更新,同时对更新萤火虫群体的状态进行高斯变异处理,如式(8)所示。

$$x_i = x_i + x_i * N(0, 1) \quad (8)$$

式中, $N(0, 1)$ 为服从期望为0、方差为1的高斯分布的随机向量。

3.3 步长收缩与域约束

为平衡进化前期的全局搜索及后期的快速收敛的双向要求,我们对步长因子 α 也进行了逐代递减的处理。

$$\alpha = \alpha * \Delta\alpha \quad (9)$$

式中, $\Delta\alpha$ 为步长衰减系数,在(0.95, 1)内取值。

同时,为保证萤火虫种群中的个体始终在指定搜索空间内进行有效搜索,本算法中对所有萤火虫个体进行了域约束处理。

设 x_{min} 为搜索空间下限, x_{max} 为搜索空间上限。则按下式对萤火虫个体实行域约束:

$$x_i = \begin{cases} x_{min}, & x_i < x_{min} \\ x_{max}, & x_i > x_{max} \end{cases} \quad (10)$$

这样就能有效保证种群中的所有个体有效组织正常搜索,当个体脱离指定范围时,能有效将个体拉回到指定空间,完成正常的全局寻优任务。

3.4 IEFA 算法流程实现

步骤1 初始化参数。设置萤火虫数目 m ,光强吸收系数 γ ,最大吸引力因子 β ,步长因子 α 等,设定收敛条件,并在求解空间内随机生成各萤火虫的初始位置 $x_i (i=1, 2, \dots, m)$ 。

步骤2 计算各萤火虫的适应度值 $f(x_i)$ 作为各自最大荧光强度 I_{0i} ,并将种群中的最佳适应度值记为 f_{best} ,同时记录最佳个体的位置。

步骤3 由式(2)计算萤火虫个体的相对荧光亮度 I_i ,比较邻域内萤火虫的荧光亮度 I 的大小,确定各萤火虫的位置进化方向。

步骤4 由式(3)计算萤火虫个体的吸引力 β ,并根据式(6)更新萤火虫的空间位置。

步骤5 根据式(7)收缩步长,并对所有萤火虫个体进行域约束操作。

步骤6 重新计算各萤火虫的适应度值,若获取种群更优适应度值,则更新 f_{best} ;若发现连续5次迭代过程中 f_{best} 都没有更新,则利用群中最优萤火虫状态替换更新种群中最差萤火虫的状态,得到过渡种群,并按照式(8)对过渡种群进行高斯变异操作。

步骤7 判断萤火虫群体是否满足收敛条件,若是,则转步骤8;否则转步骤2。

步骤8 结束迭代,输出结果。

4 实验仿真及分析

本文采用了5个标准测试函数来分析与验证IEFA算法

的收敛速度及寻优能力:

a) FourPeaks 函数

$$f_1(x, y) = -e^{-(x-4)^2 - (y-4)^2} - e^{-(x+4)^2 - (y-4)^2} - 2e^{-x^2 - (y+4)^2} - 2e^{-x^2 - y^2} \quad (11)$$

(为统一化分析, 将原 FourPeaks 函数取相反数得式 (11))

b) Shaffer's f6 函数

$$f_2(x) = -0.5 + \frac{\sin \sqrt{x^2 + y^2} - 0.5}{(1.0 + 0.001(x^2 + y^2))^2} \quad (12)$$

c) Rosenbrock 函数

$$f_3(x) = \sum_{i=1}^N [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \quad (13)$$

d) Ackley 函数

$$f_4(x, y) = 20 + e - 20e^{-0.2\sqrt{\frac{x^2 + y^2}{2}}} - e^{\frac{\cos 2\pi x}{2} + \cos 2\pi y} \quad (14)$$

e) Griewank 函数

$$f_5(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (15)$$

各测试函数的具体参数设置如表 1 所列。

表 1 测试函数具体参数表

函数	维数(N)	搜索区间	目标值(最小值)
f_1	2	$(-50, 50, -50, 50)^N$	-2
f_2	2	$(-100, 100, -100, 100)^N$	-1
f_3	30	$(-50, 50)^N$	0
f_4	30	$(-100, 100)^N$	0
f_5	30	$(-100, 100)^N$	0

在 IEMA 算法与 FA 算法的对比测试实验中, 萤火虫种群数目均取为 100, 迭代次数为 200。算法的参数取值为: 最大吸引度因子 $\beta_0 = 1$, 步长因子 $\alpha = 0.3$, 步长衰减系数 $\Delta\alpha = 0.97$, 光强吸收因子 $\gamma = 1$ 。利用 IEMA 算法与 FA 算法对每个标准函数各独立测试 20 次。表 2 所列为各组实验的最优适应度统计结果。

表 2 IEFA 与 FA 的实验结果对比

函数	算法	最差值	最优值	平均值	方差
f_1	IEFA	-2	-2	-2	0
	FA	-1.0932e-16	-0.5889	-0.0209	0.0673
f_2	IEFA	-0.9902	-1	-0.9966	0.0048
	FA	-0.5661	-0.9555	-0.7190	0.1253
f_3	IEFA	28.9000	26.7033	28.6271	0.1630
	FA	1.4029e9	1.1029e9	1.3031e9	1.5218e8
f_4	IEFA	7.6719e-15	0	6.3167e-16	9.7267e-16
	FA	21.4723	21.2477	21.3459	0.0647
f_5	IEFA	1.0998e-16	0	5.4989e-18	2.4592e-17
	FA	19.0617	15.0966	17.2020	1.2009

从表 2 中可以看出, 由于二维函数 f_1, f_2 的搜索区间设置过大, FA 算法中的各萤火虫相互距离过远, 已难于在种群中建立有效的信息传递系统, 因此很快就出现进化停滞现象; 同样, 在求解高维函数 f_3, f_4, f_5 时, FA 算法也不能有效组织萤火虫个体协同寻优。IEFA 算法则对各测试函数均表现出良好的寻优性能: f_1, f_2, f_4, f_5 测试结果已达到预期要求; f_3 的测试结果虽与目标值还有差距, 但与目前其他改进型群智能算法的寻优结果相比, 依然具有一定优势。方差结果进一步说明了 IEFA 算法具有很好的适应性和鲁棒性。

为了更加直观全面地认识 IEFA 算法的有效性, 我们

对 IEFA 算法与 FA 算法的最优适应度值的进化过程进行了比较, 结果如图 5—图 9 所示。

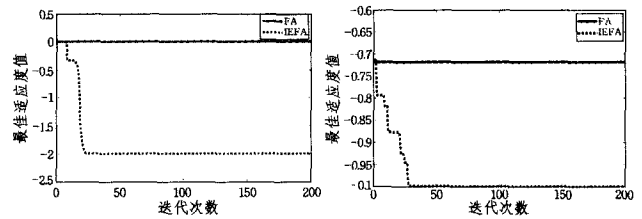


图 5 Fourpeaks 函数进化适应度曲线图

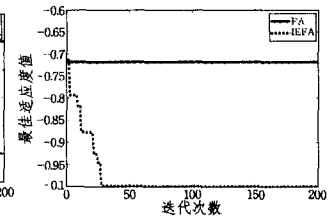


图 6 Shaffer's f6 函数进化适应度曲线图

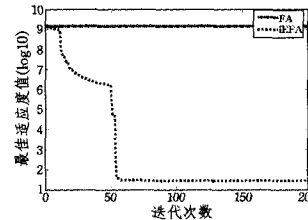


图 7 Rosenbrock 函数进化适应度曲线图(对数坐标)

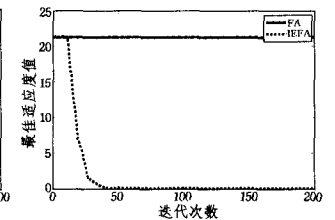


图 8 Ackley 函数进化适应度曲线图

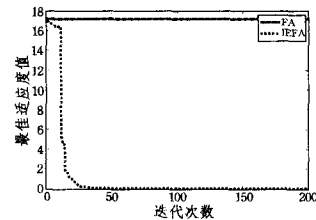


图 9 Griewank 函数进化适应度曲线图

从图 5—图 9 所示的各函数最优适应度进化曲线可以看出, IEFA 算法在进化早期利用种群最优萤火虫牵引所有萤火虫个体进行全局寻优, 有效解决了 FA 算法由于各萤火虫距离过远而过早停滞进化的问题; IEFA 算法同时还吸收了 FA 算法的原有特点, 并能利用高斯变异跳出局部最优区域, 在一定程度上平衡了全局寻优与局部寻优的综合要求, 整体表现出较快的收敛速度及较高的收敛精度。

结束语 萤火虫算法结构简单, 调节参数少, 同时具有良好的寻优能力, 但在求解高维优化问题或者求解区域过大时容易出现进化过早停滞的现象。本文对萤火虫算法的进化机制进行了研究, 提出综合考虑种群最佳萤火虫对其他个体的牵引作用能有效提高算法的效率, 同时利用高斯变异可帮助萤火虫群体克服早熟收敛问题。对改进型优化机制下的萤火虫算法 IEFA 进行了实验验证, 结果表明 IEFA 算法改善了进化过早停滞的现象, 同时能够有效平衡局部搜索及全局搜索的全面要求, 具有较好的适应性和鲁棒性。今后开展的相关工作主要侧重于 IEFA 算法的理论分析, 以及 IEFA 算法中的结构、参数设置对寻优效果的影响研究。

参考文献

[1] Yang Xin-she. Nature-inspired metaheuristic algorithms [M]. Luniver Press, 2008: 83-96

(下转第 248 页)

$$\Delta_2^*(e_7) = A_2 \wedge \Delta^*(e_5) = c_2$$

$$\Delta_3^*(e_7) = A_3 \wedge \Delta^*(e_2) \wedge \Delta^*(e_5) = c_3 \wedge (c_3 \wedge (c_1 \vee c_2)) = c_3 \wedge (c_1 \vee c_2)$$

$$\overline{\Delta^*(e_7)} = \Delta_1^*(e_7) \vee \Delta_2^*(e_7) \vee \Delta_3^*(e_7) = (c_1 \wedge c_3) \vee c_2 \vee (c_3 \wedge (c_1 \vee c_2)) = (c_1 \wedge c_3) \vee c_2$$

故基于对象 e_7 的近似约简有 $\{c_1, c_3\}, \{c_2\}$ 。

(8) 基于对象 e_8 的相对约简有 $A_1 = \{c_1\}, A_2 = \{c_2\}, A_3 = \{c_3\}$ 。由 $[e_8]_{A_1} = \{e_4, e_5, e_6, e_7, e_8\}, [e_8]_{A_2} = \{e_1, e_2, e_3, e_4, e_6, e_8\}, [e_8]_{A_3} = \{e_3, e_6, e_8\}$ 知

$$\Delta_1^*(e_8) = A_1 \wedge \Delta^*(e_4) \wedge \Delta^*(e_5) \wedge \Delta^*(e_6) \wedge \Delta^*(e_7) = c_1 \wedge c_3$$

$$\Delta_2^*(e_8) = A_2 \wedge \Delta^*(e_1) \wedge \Delta^*(e_2) \wedge \Delta^*(e_3) \wedge \Delta^*(e_4) \wedge \Delta^*(e_5) = c_2 \wedge c_3 \wedge (c_3 \wedge (c_1 \vee c_2)) \wedge (c_1 \wedge c_3) \wedge c_3 = c_1 \wedge c_2 \wedge c_3$$

$$\Delta_3^*(e_8) = A_3 \wedge \Delta^*(e_3) \wedge \Delta^*(e_6) = c_3 \wedge (c_1 \wedge c_3) = c_1 \wedge c_3$$

$$\overline{\Delta^*(e_8)} = \Delta_1^*(e_8) \vee \Delta_2^*(e_8) \vee \Delta_3^*(e_8) = (c_1 \wedge c_3) \vee (c_1 \wedge c_2 \wedge c_3) \vee (c_1 \wedge c_3) = c_1 \wedge c_3$$

故基于对象 e_8 的近似约简有 $\{c_1, c_3\}$ 。

由例 1 的计算结果容易看出, 基于对象的近似约简结果不仅简洁, 而且可以获得多种约简方式, 打破了整体约简的局限性。同时, 关于对象的近似约简具有很大的应用价值。以例 1 中的决策表 DT 为例, 如果采取整体约简, 则医生在诊断与下药时对 DT 中的每一个病例都要考虑头痛和体温两个病症。但如果采用对象约简便可以针对不同病人快速准确地诊断与下药, 避免不必要的病症对医生的工作产生干扰, 同时也可以减少对药品的浪费。

结束语 本文首先对粗糙集的基本理论知识进行介绍, 针对不协调目标信息系统中基于对象的近似约简问题展开研究; 给出近似约简的等价定义和判定定理, 通过区分矩阵与区分函数给出近似约简的计算方法; 并通过实例分析整体约简

与基于对象的近似约简各自的优势及适用的场景。本文提出的基于区分矩阵与区分函数的算法简单易理解, 且能够找出任一对象的所有近似约简, 但该方法时间复杂度为 $O(|A|^2|U|^2)$, 其中 $|A|$ 是条件属性个数, $|U|$ 是记录个数^[10], 故用于海量数据约简时效率偏低。因此我们下一步将对此方法进行改进, 以扩大其使用范围。

参考文献

- [1] Pawlak Z. Rough sets [J]. International Journal of Computer and Information Science, 1982, 11: 341-356
- [2] Pawlak Z. Rough Sets; Theoretical Aspects of Reasoning about Data[M]. Boston: Kluwer Academic Publishers, 1991
- [3] Zhang X H, Zhou B, Li P. A general frame for intuitionistic fuzzy rough sets[J]. Information Sciences, 2012, 216: 34-49
- [4] 张文修, 梁怡, 吴伟志. 信息系统与知识发现[M]. 北京: 科学出版社, 2003
- [5] 杜卫锋, 秦克云. 决策表正域约简区分函数条件的改进[J]. 计算机工程与应用, 2006, 42(20): 16-18
- [6] Kryszkiewicz M. Comparative study of alternative types of knowledge reduction in inconsistent systems[J]. International journal of intelligent systems, 2001, 16: 105-120
- [7] 张文修, 吴伟志, 梁吉业, 等. 粗糙集理论与方法[M]. 北京: 科学出版社, 2001
- [8] Skowron A, Rauszer C. The discernibility matrices and functions in information systems. Intelligent Decision Support; Handbook of Applications and Advances to Rough Sets Theory[M]. Dordrecht: Kluwer Academic Publishers, 1992: 331-362
- [9] Yao Y Y, Zhao Y. Discernibility matrix simplification for constructing attribute reducts[J]. Information Sciences, 2009, 179(7): 867-882
- [10] 蒋云良, 杨章显, 刘勇. 不协调信息系统快速属性分布约简方法[J]. 自动化学报, 2012, 38(3): 382-388
- [7] dos Santos Coelho L, Mariani V C. Firefly algorithm approach based on chaotic Tinkerbell map applied to multivariable PID controller tuning[J]. Computers & Mathematics with Applications, 2012, 64(8): 2371-2382
- [8] Horng M-H. Vector quantization using the firefly algorithm for image compression[J]. Expert Systems with Applications, 2012, 39(1): 1078-1091
- [9] Senthilnath J, Omkar S N, Mani V. Clustering using firefly algorithm; Performance study [J]. Swarm and Evolutionary Computation, 2011, 1(3): 164-171
- [10] Yang Xin-she, Deb S. Eagle strategy using lévy walk and firefly algorithms for stochastic optimization[J]. Studies in Computational Intelligence, 2010, 28(4): 101-111
- [11] 刘长平, 叶春明. 一种新颖的仿生群智能优化算法: 萤火虫算法[J]. 计算机应用研究, 2011, 28(9): 3295-3297
- [12] Gandomi A H, Yang X-S, Talatahari S, et al. Firefly algorithm with chaos[J]. Communications in Nonlinear Science and Numerical Simulation, 2013, 18(1): 89-98

(上接第 231 页)

- [2] Gandomi A H, Yang Xin-she, Alavi A H. Mixed variable structural optimization using Firefly Algorithm[J]. Computers & Structures, 2011, 89(23): 2325-2336
- [3] Sayadi M K, Hafezalkotob A, Naini S G J. Firefly-inspired algorithm for discrete optimization problems; An application to manufacturing cell formation[J]. Journal of Manufacturing Systems, 2013, 32(1): 78-84
- [4] Srivatsava P R, Mallikarjun B, Yang Xin-she. Optimal test sequence generation using firefly algorithm Original Research Article[J]. Swarm and Evolutionary Computation, 2013, 8(1): 44-53
- [5] Kazem A, Sharifi E, Hussain F K, et al. Support vector regression with chaos-based firefly algorithm for stock market price forecasting[J]. Applied Soft Computing, 2013, 13(2): 947-958
- [6] Chandrasekaran K, Simon S P. Network and reliability constrained unit commitment problem using binary real coded firefly algorithm[J]. International Journal of Electrical Power & Energy Systems, 2012, 43(1): 921-932