

一种改进的微粒群算法^{*}

王霄鹏, 胡劲松

(华南理工大学 计算机科学与工程学院, 广州 510006)

摘要: 通过在微粒群算法中引入排雷策略的思想, 对微粒群优化算法进行改进, 使微粒群算法能摆脱局部极值点的束缚; 另外通过在算法的迭代过程中加入旋转方向法, 加快算法的收敛速度, 从而形成一种新的改进粒子群算法。通过对三个典型函数进行优化计算, 并与其他文献的改进微粒群算法的优化结果进行比较, 表明基于排雷策略的改进算法很好地解决了粒子群优化算法早收敛、难以跳出局部极值点和收敛较慢的问题。

关键词: 微粒群优化算法; 排雷策略; 旋转方向法; 收敛

中图分类号: TP301.6 **文献标志码:** A **文章编号:** 1001-3695(2009)10-3642-03

doi 10.3969/j.issn.1001-3695.2009.10.011

Modified particle swarm optimization algorithm

WANG Xu-peng HU Jin-song

(School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, China)

Abstract To deal with the problem of premature convergence and slow search speed, this paper proposed a new particle swarm optimization (PSO). The new method was based on clearing of mines, which was guaranteed to converge to the global optimization solution with probability one. In addition, combined the new method with rotating direction method, which was beneficial for the convergence speed. Through the calculation of three typical function and made comparison with other improvement particle swarm optimization algorithm, they show that the new method which solves the problem of premature convergence and slow search speed.

Keywords particle swarm optimization (PSO); strategy of clearing the mine; rotating direction method; convergence

0 引言

目前, 通过模拟生物群体行为解决计算问题已经成为新的研究热点, 并且形成了以群体智能为核心的理论体系。通过对生物群体的观察和研究发现, 生物群体内个体间的合作与竞争等复杂性行为产生的群体智能往往能对某些特定的问题提供高效的解决方法。Kennedy 等人^[1,2]于 1995 年提出了一种基于种群搜索的自适应进化计算算法——微粒群优化 (PSO) 算法。该算法最初受到飞鸟和鱼类集群活动的规律性启发, 用组织社会行为代替了进化算法的自然选择极值, 通过种群间个体协作来实现对问题的最优解的搜索。PSO 算法随机产生一个初始种群, 并赋予每个微粒一个随机速度, 通过自身以及同伴的飞行经验来进行动态调整, 整个群体有飞向更好搜索区域的能力。PSO 算法概念简单, 容易实现, 参数较少, 能有效解决复杂优化任务, 所以在过去几年获得了飞速发展, 并在图像处理、模式识别、多目标优化和游戏设计等多领域得到广泛应用。

目前, PSO 算法作为一种崭新的随机搜索算法, 仍然存在早熟收敛和收敛较慢这两个难题, 并且具有种群多样性随代数增加下降过快、有可能不收敛到全局最优解等缺点。为了避免早熟收敛, 一些研究者提出了通过控制种群多样性来提高算法性能的方法。一方面, 通过解决微粒间的冲突和聚集^[3]、分析微粒和种群最优位置的距离^[4]、增加环境检测和响应、种群随

机多代初始化^[5]等思想, 给出了不同策略来增强种群多样性, 使算法不至于过早陷入局部极小; 另一方面, 通过引入遗传算法的变异操作来增强全局搜索性能也是一种十分有效的方式。文献[6]通过对微粒速度或位置引入一个小概率随机变异操作来增强种群多样性 (dissipative PSO, DPSO), 使算法能够有效地进行全局搜索, 但是过大的变异率在增加种群多样性的同时也将导致群体发生混乱, 使种群不能进行精确的局部搜索, 延缓了算法的收敛速度。Zhang 等人^[7]将差分进化算子变异引入 PSO 中, 并采用 PSO 算法和变异操作交替迭代的方式。郑小霞等人^[8]提出基于差分进化算子变异的改进粒子群算法等。这些改进虽然对 PSO 算法性能有一定的改善, 增强了群体的多样性, 使算法的全局优化性能得到了一定的提高, 但这些改进要么无法从根本上解决早熟收敛问题, 要么很难在提高搜索速度与保持种群多样性之间达到平衡。

为了使 PSO 更容易逃脱局部极值, 提高搜索效率和扩大应用范围, 本文根据排雷策略的思想^[9-11]提出一种替换策略, 用于替换种群中的适应值最差的微粒, 从而避免 PSO 过早收敛的问题。同时, 为了加快收敛, 在算法的迭代过程中又结合局部搜索算法——旋转方向法^[12], 形成了一种新的粒子群优化算法。为了深入对比分析改进后的 PSO 算法的效果, 本文引入了三个著名的测试函数 (Schaffer Griewank Rastrigin) 进行数字实验, 并与基本 PSO、随机微粒群 (SPSO) 算法^[13]、MPSO

收稿日期: 2009-01-21; 修回日期: 2009-03-19 基金项目: 国家自然科学基金资助项目 (60574078); 广东省自然科学基金资助项目 (31454); 广州市科技计划应用基础研究项目 (2006J1-C0321)

作者简介: 王霄鹏 (1984-), 男, 硕士研究生, 主要研究方向为智能算法优化 (nudtwxp@yeah.net); 胡劲松 (1973-), 男, 副教授, 主要研究方向为智能优化算法、神经网络。

©1994-2012 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

算法^[14]、基于差分进化算子变异的粒子群算法^[11]、DPSO^[6]和由 Shi 等人^[15]的 PSO 改进方法进行了对比分析。结果显示, 改进后的粒子群算法与其他算法相比在全局寻优能力上有所提高。

1 改进的 PSO 算法

1.1 粒子群优化算法的基本原理

在 PSO 中, 每个优化问题的潜在解都是搜索空间中的一只鸟, 称之为“粒子”。所有的粒子都是一个由被优化的函数决定的适应值 (fitness value), 每个粒子还有一个速度决定其飞翔的方向和距离。在每一次迭代中, 粒子通过跟踪两个极值来更新自己。第一个就是粒子自身所找到的最优解, 这个解称为个体极值 p_{id} ; 另一个是整个种群目前找到的最优解, 这个极值是全局极值 p_{gd} 。假设在 D 维空间, 有 m 个微粒组成一微粒群, 其中第 i 个微粒的空间位置为 $x_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{iD})$, $i = 1, 2, \dots, m$, 第 i 个微粒所经历的最优位置为其个体历史最优位置 P_i , 微粒飞行速度为 V_i 。所有微粒经历过的最优位置为全局历史最优位置 P_g , 相应的适应值为全局历史最优适应值 F_g 。对每一代微粒, 根据如下方程迭代:

$$V_{id} = [\omega \times V_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id})] \quad (1)$$

$$x_{id} = x_{id} + V_{id} \quad (2)$$

其中: ω 为惯性权值, 当 $\omega \neq 0$ 时, 微粒具有扩展搜索空间的趋势, 即具有一定的全局搜索能力; c_1 和 c_2 为加速系数; r_1 和 r_2 是两个在 $[0, 1]$ 内变化的随机数。

式 (1) 所描述的速度进化方程中, 第一部分为微粒先前的速度; 第二部分为认知部分, 因为它仅仅考虑了微粒自身的经验, 表示微粒本身的思考。如果基本微粒群算法的速度进化方程仅包含认知部分, 则其性能变差。其主要原因是不同的微粒间缺乏信息交流, 即没有社会信息共享, 微粒间没有交互, 因而得到最优解的概率非常小。式 (1) 的第三部分为社会部分, 表示微粒间的社会信息共享。若速度进化方程中仅包含社会部分, 则微粒没有认识能力, 也就是只有社会的模型。这样, 微粒在相互作用下, 有能力到达新的搜索空间, 虽然其收敛速度比基本微粒群算法更快, 但对于复杂问题, 则容易陷入局部最优解。为了增强全局搜索能力, 一些学者提出了随机粒子群算法, 它是在标准 PSO 的基础上作了一些改进: 在每一次迭代过程中, 随机产生新的粒子代替种群中全局最优粒子。

由式 (1) 可以看出, 粒子在追逐最优粒子的过程中, 越接近最优粒子, 其速度越小, 因此粒子群呈现强烈的趋同性, 表现在优化性能就是基本 PSO 算法易陷入局部极值点。随机 PSO 算法通过随机产生新的粒子, 在一定程度上减少了陷入局部极值点的可能性, 但由于算法具有随机性, 无法从根本上跳出局部极值点。

1.2 排雷策略

排雷策略巧妙地利用已找到的极值点为边界, 形成新的起点, 直接从局部极小值点跳出来, 而不像模拟退火、Tabu 算法是逐渐摆脱, 因此排雷法克服了局部极小这一难题, 并且算法不因克服局部极小而增加很多额外的计算量, 效率高。

为了让 PSO 算法更加容易逃脱局部极值点, 笔者根据排雷策略的思想, 在随机粒子群算法的基础上, 通过在每次迭代后选择合适的微粒来替代种群中适应值较差的粒子, 形成了一种新的改进粒子群算法。因为新的算法并没有改变随机粒子

群算法的结构, 与随机粒子群算法一样能保证收敛^[13]。

具体替换策略: 通过排雷策略在每次迭代后计算每个粒子到种群中其他所有粒子的距离以及每个粒子到边界的距离, 并把两个距离相加 (按照式 (3) 计算距离和), 找出这个与最大的粒子代替当前种群中适应值最差的粒子, 从而解决在寻优过程中的早熟现象。

$$\text{distance}[i] = \sum_{j=1}^N (x_i - x_j)^2 + (x_i - \text{XM N})^2 + (x_i - \text{XM AX})^2 \quad (3)$$

其中: $\text{distance}[i]$ 表示种群中第 i 个粒子到种群中其他所有粒子的距离以及到边界的距离和。式 (3) 中的第一部分表示种群中第 i 个粒子到种群中其他所有粒子的距离和, 第二、三部分分别表示种群中第 i 个粒子到两个边界的距离。

1.3 旋转方向法

旋转方向法作为一种经典的优化方法, 是 Rosenbrock 等人在 1960 年提出的, 它通过沿搜索方向取离散步来实现^[5]。旋转方向法 (Rosenbrock 方法) 是一种直接搜索^[7]的优化算法, 它由两个动作组成^[8]: 沿一组正交方向的探测移动和旋转正交方向组。这个方法第 K 次迭代可描述如下: 设当前迭代点为 X^k , 首先沿当前正交方向组作探测移动 (初始点 X_0 的正交方向组可取为坐标轴方向), 得到 X^{k+1} 。若 $\|X_{k+1} - X_k\| < \varepsilon$, 迭代停止; 否则, 按某种规则产生新的正交方向组 (相当于把原方向组旋转一个角度), 进入下一次迭代。

为了进一步提高 PSO 算法跳出局部极值点的概率, 加快算法的收敛速度, 算法还作了下面的改进: 如果全局最优值在连续若干次 (本文程序中设置为 20 次) 迭代过程中没有改变, 此时找出种群中当前适应值最优的微粒, 通过式 (4) 计算生成新的位置, 用新的位置替换当前种群中适应值第二差的粒子的位置。为了加速算法的收敛速度, 对新生成的粒子使用旋转方向法, 从而加速产生新的局部极值 (因为全局极值点也是一个局部极值点)。

1) 当 $|X_i - \text{XM N}| \leq |X_i - \text{XM AX}|$ 时,

$$X_{\text{badi}} = X_i + (\text{XM AX} - X_i) / 2 \quad (4)$$

2) 当 $|X_i - \text{XM N}| > |X_i - \text{XM AX}|$ 时,

$$X_{\text{badi}} = X_i + (\text{XM N} - X_i) / 2 \quad (5)$$

其中: XM N 表示粒子的位置坐标下限; XM AX 表示粒子的位置坐标上限; X_i 表示最优粒子的第 i 维在空间中的位置; X_{badi} 表示适应值第二差粒子的第 i 维在空间中的位置。

1.4 改进的 PSO 算法流程

a) 初始化设置改进粒子群算法的各参数。

b) 按优化函数评价各微粒的初始适应值。

c) 如果当前种群中有粒子的适应值正好与到目前为止找到的全局最优值相同以及该粒子的历史最好适应相等时, 则随机产生新的粒子代替该粒子。

d) 根据式 (2) (5) 计算各微粒新的位置和微粒新的速度, 并对微粒位置和位置分别进行必要的处理。

e) 更新并储存各微粒的个体历史最优位置和个体历史最优适应值; 更新并储存微粒群的全局历史最优位置和全局历史最优适应值。

f) 如果全局历史最优适应值已经连续若干次没有得到改进, 按照式 (4) 产生新的粒子位置替换当前种群中适应值第二差的粒子位置, 并对其使用旋转方向法, 然后转到 g)。

g) 根据排雷策略的思想, 根据式 (3) 计算的结果, 选择式 (3) 计算结果最大的那个粒子替换当前种群中适应值最差的

粒子的位置。

h)若满足停止条件(适应值误差小于设定的适应值误差限或迭代次数超过最大允许迭代次数),搜索停止,输出全局历史最优位置和全局历史最优适应值为所求结果;否则,返回c)继续搜索。

2 测试函数的优化与结果讨论

为了分析改进后的 PSO 算法的收敛速度、全局寻优能力,本文进行了两组实验。实验 1 选择 PSO、随机粒子群算法、MPSO 算法、基于差分进化算子变异的粒子群算法^[8]与改进后的粒子群算法进行对比实验,引入 Schaffer Griewank Rastrigin 三个函数优化问题来进行分析。实验 2 选择 DPSO 和文献[15]中的 PSO 改进方法与本文的方法进行比较,引入 Rastrigin Griewank 两个函数优化问题来进行分析。

1) Schaffer函数

$$f(x_1, x_2) = 0.5 + [(\sin \sqrt{x_1^2 + x_2^2})^2 - 0.5] / [1 + 0.001 \times (x_1^2 + x_2^2)]^2 \\ - 10 \leq x_1, x_2 \leq 10$$

函数在(0,0)处取得全局极小值 0,在距离全局极小值点大约 3.14 的范围内存在无限多的局部极小值点,函数强烈振荡的形态使其难以实现全局最优化。

2) Griewank 函数

$$f(x_1, x_2, \dots, x_n) = \frac{1}{4000} \sum_{i=1}^n (x_i^2 / 4000) - \prod_{i=1}^n \cos(x_i \sqrt{i}) + 1 \\ - 300 \leq x_i \leq 300$$

函数在 $x_i = 0 (i = 1, 2, \dots, n)$ 处取得全局极小值 0,在 $x_i = \pm k\pi\sqrt{i} (i = 1, 2, \dots, n; k = 0, 1, 2, \dots, n)$ 处取局部极小值。

3) Rastrigin 函数

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^n (x_i^2 - A \cos(2\pi x_i) + A) \\ - 5.12 \leq x_i \leq 5.12, A = 10$$

该函数在 $x_i = 0 (i = 1, 2, \dots, n)$ 处取得全局极小值 0,在定义域内大约有 $10n$ 个局部极小值点。

实验 1 参数设置: PSO、SPSO、MPSO、带差分进化变异的 PSO 和本文改进的 PSO 五种算法的种群规模大小都取 30,惯性权重从 1 线性减小到 0.5, $c_1 = c_2 = 2$, 函数 f_1 的维数为 2, 函数 f_2 的维数取 10, 函数 f_3 的维数取 10, 五种算法求解各运行 50 次,其他参数和运行结果见表 1~3, 表 1 中的 0 表示计算结果小于 $1e-20$, 收敛率为 $1e-04$ 。

表 1 Schaffer 函数的运行结果

算法种类	Schaffer 函数				
	迭代次数	平均值	最优值	收敛率/%	评估函数调用次数
PSO	3 000	7.163e-05	5.128e-16	100	90 000
SPSO	3 000	7.359e-05	2.409e-18	100	90 000
MPSO	3 000	2.512e-07	3.426e-12	100	90 000
基于差分进化算子变异的 PSO	3 000	0	0	100	90 000
本文的改进 PSO	500	0	0	100	15 196

表 2 Griewank 函数的运行结果

算法种类	Griewank 函数				
	迭代次数	平均值	最优值	收敛率/%	评估函数调用次数
PSO	50 000	0.088 1	1.983e-03	0	1 500 000
SPSO	50 000	0.079 1	2.420e-03	0	1 500 000
MPSO	50 000	0.016 2	2.451e-06	25	1 500 000
基于差分进化算子变异的 PSO	3 000	0.025 37	8.253e-08	40	90 000
本文的改进 PSO	1 000	8.388e-12	0	100	37 950

表 3 Rastrigin 函数的运行结果

算法种类	Rastrigin 函数				
	迭代次数	平均值	最优值	收敛率/%	评估函数调用次数
PSO	20 000	1.492 6	5.430e-06	25	600 000
SPSO	20 000	1.387 5	1.972e-09	30	600 000
MPSO	20 000	0.050 2	5.389e-12	75	600 000
基于差分进化算子变异的 PSO	20 000	0.006 64	4.198e-13	81.5	600 000
本文的改进 PSO	20 000	1.589e-08	3.901e-12	100	603 480

从表 1、2 可看出:对于给定的 Schaffer 和 Griewank 函数,改进的 PSO 算法的迭代次数都远小于其他四种 PSO 算法,并且在所讨论的精度范围 ($1e-04$) 内收敛都要高于其他四种 PSO 算法,在找到全局最优点的过程中,评估函数的调用次数也小于其他四种算法。

从表 3 可以看出,对于 Rastrigin 函数,改进的 PSO 算法在迭代次数与其他四种 PSO 算法相同的情况下,全局寻优能力更强,但由于算法中加入了旋转方向法,导致评估函数调用次数略多于其他四种 PSO 算法。

对于多维复杂函数,改进的 PSO 算法跳出局部极值点的概率大大提高,收敛速度也明显加快。由于在寻优过程中加入了旋转方向法,一次迭代的时间开销会有所增加。实验 2 参数设置如表 4 所示。

表 4 实验 2 参数设置表

类型	SF ₀	SF ₁	DF ₂	DF ₃
PSO 版本	SPSO	SPSO	DPSO	DPSO
初始化	不对称	对称	对称	对称
惯性权值	0.9~0.4	0.9~0.4	0.9~0.4	0.4

另外, $c_1 = c_2 = 2$, 本文改进的 PSO 算法惯性权值也是从 0.9~0.4 线性减少。所有方法对于函数运行 50 次取平均值,实验结果见表 5 和 6。

表 5 Rastrigin 函数平均值

算法种类	粒子数	维数	迭代次数	平均值
SF ₀	20	10	1 000	5.557 2
	40			3.562 3
SF ₁	20	10	1 000	5.206 20
	40			3.569 74
DF ₂	20	10	1 000	3.081 28
	40			1.629 99
DF ₃	20	10	1 000	0.470 68
	40			0.076 19
本文的改进 PSO	20	10	1 000	0.198 99
	40			0.052 46

表 6 Griewank 函数平均值

算法种类	粒子数	维数	迭代次数	平均值
SF ₀	20	10	1 000	0.091 9
	40			0.086 2
SF ₁	20	10	1 000	0.096 09
	40			0.086 22
DF ₂	20	10	1 000	0.089 37
	40			0.081 70
DF ₃	20	10	1 000	0.065 06
	40			0.056 73
本文的改进 PSO	20	10	1 000	7.0499e-15
	40			1.1146e-15

从表 5、6 可以看出:在相同的迭代次数下,本文改进的 PSO 算法在寻优能力方面,在相同迭代次数和相同参数的情况下,强于文献[6,15]中改进的 PSO 方法。其中 SF₀、SF₁、DF₂、DF₃ 四种方法的计算结果数据是由文献[6]给出的。

3 结束语

本文在排雷思想的启发下,在随机粒子群算法的基础上提出了选择替换策略。在每次迭代后,利用排雷思想对种群进行精心选择,用精心选择后的粒子取代当前种群中适应度较差的那些粒子,在不破坏 PSO 算法结构简单的基础上,在很大程度上加大了算法逃脱局部极值点的概率,特别是针对复杂的多峰函数,与本文中提到的其他 PSO 改进算法相比,更加容易跳出局部极值点,避免早收敛。另外还在迭代过程中适当地加入了旋转方向法,提高了收敛速度(迭代次数少于本文中的其他算法)。但是改进后的算法在每次迭代后要

(下转第 3648 页)

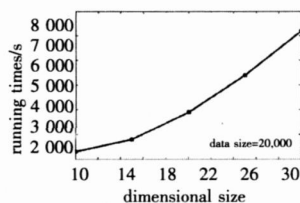


图 3 算法 CSL 对数据集维数的可扩展性

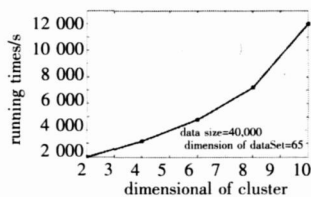


图 4 算法 CSL 对子空间维数的可扩展性

4 结束语

高维数据的稀疏性、噪声数据多等特点使得经典聚类算法无法发现其中有意义的聚类结果。子空间聚类是解决高维聚类的重要手段。事实上哪些属性子集包含有意义的聚类结果是未知的,因此本文将子空间聚类问题转换为由未知所引起的不确定性推理问题。在 D-S 证据理论的基础上,给出了可信子空间的定义及采用贪心策略的启发式搜索算法 CSL。实验结果表明,CSL 具有发现高维数据集的真实子空间的能力,为经典聚类算法处理高维数据的聚类问题提供了新的途径。

参考文献:

- [1] PAVEL B. Survey of clustering data mining techniques[R]. San Jose, California: Acme Software Inc, 2002
- [2] RUI X, WUNSCH D. Survey of clustering algorithms[J]. *IEEE Transactions on Neural Networks*, 2005, 16(3): 645-678
- [3] HAN Jia-wei, KAMBER M. 数据挖掘: 技术与概念[M]. 范明, 孟小峰, 译. 2版. 北京: 机械工业出版社, 2007
- [4] 孙吉贵, 刘杰, 赵连宇. 聚类算法研究[J]. *软件学报*, 2008, 19(1): 48-61

(上接第 3644 页)通过排雷策略,精心选择合适的粒子用于取代种群中适应值较差的粒子,而且又加入了旋转方向法,随着问题维数的增加,一次迭代的时间开销也随之增加。进一步的工作主要是在不影响寻优能力的前提下,尽可能地减少算法一次迭代的时间开销。

参考文献:

- [1] KENNEDY J, EBERHART R C. Particle swarm optimization[C]//Proc of IEEE International Conference on Neural Networks, Perth, WA: [s.n.], 1995: 1942-1948
- [2] EBERHART R C, KENNEDY J A new optimizer using particle swarm theory[C]//Proc of the 6th International Symposium on Micro Machine and Human Science, Nagoya [s.n.], 1995: 39-43
- [3] KRINK T, VESTERSTROM J S, RIGET J Particle swarm optimization with spatial particle extension[C]//Proc of IEEE International Conference on Evolutionary Computation, Honolulu: IEEE Inc, 2002: 1474-1497
- [4] KAZEM IB A I, MOHAN C K. Multi-phase generalization of the particle swarm optimization algorithm[C]//Proc of IEEE International Conference on Evolutionary Computation, Honolulu: IEEE Inc, 2002: 489-494
- [5] HU Xiao-hui, EBERHART R C. Adaptive particle swarm optimization: detection and response to dynamic system[C]//Proc of IEEE International Conference on Evolutionary Computation, Honolulu: IEEE Inc, 2002: 1666-1670
- [6] XIE Xiao-feng, ZHANG Wen-jun, YANG Zhi-lian. A dissipative particle

- [5] PARSONS L, HAQUE E, LIU Huan. Subspace clustering for high dimensional data: a review[J]. *ACM SIGKDD Explorations Newsletter*, 2004, 6(1): 90-105.
- [6] AGRAWAL R, GEHRKE J, GUNOPULOS D, et al. Automatic subspace clustering of high dimensional data for data mining applications[C]//Proc of ACM-SIGMOD International Conference on Management of Data, Seattle, WA: [s.n.], 1998: 94-105.
- [7] CHENG C H, FU A W, ZHANG Y i. Entropy-based subspace clustering for mining numerical data[C]//Proc of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining [S.l.]: ACM Press, 1999: 84-93
- [8] GOIL S, NAGESH H, CHOUDHARY A. MAFLA: efficient and scalable subspace clustering for very large data sets. Technical Report CP-DC-TR-9906-010[R]. [S.l.]: Northwestern University, 1999
- [9] KALING K, KRUEGER H, KROGER P. Density-connected subspace clustering for high dimensional data[C]//Proc of the 4th SIAM International Conference on Data Mining, Lake Buena Vista, FL: [s.n.], 2004: 246-257
- [10] DEMPSTER A, UPPER L. Probabilities induced by multi-valued mapping[J]. *Annals of Mathematical Statistics*, 1967, 38(2): 325-339
- [11] ORPONEN P. Dempster's rule of combination is P-complete[J]. *Artificial Intelligence*, 1990, 44(1-2): 245-253
- [12] 诸葛建伟, 王大为, 陈显, 等. 基于 D-S 证据理论的网络异常检测方法[J]. *软件学报*, 2006, 17(3): 463-471
- [13] 周晓云, 孙志辉, 张柏礼, 等. 高维数据流子空间聚类发现及维护算法[J]. *计算机研究与发展*, 2006, 43(5): 834-840

cle swarm optimization[C]//Proc of IEEE International Conference on Evolutionary Computation, Honolulu: IEEE Inc, 2002: 1456-1461.

- [7] ZHANG Wen-jun, XIE Xiao-feng. DEPSO: hybrid particle swarm with differential evolution operator[C]//Proc of IEEE International Conference on Systems Man and Cybernetics, 2003: 3816-3821
- [8] 郑小霞, 钱锋. 一种改进的微粒群优化算法[J]. *计算机工程*, 2006, 32(15): 25-27
- [9] HU Jing-song, HU Gui-wu, WANG Jia-bing. FCMAC based on mine-sweeping strategy[C]//Proc of International Conference on Machine Learning and Cybernetics, 2005: 784-787.
- [10] HU Jing-song, WANG Jia-bing, YANG Chuang-xin. Self-optimum fuzzy controller based on mine-sweeping strategy[C]//Proc of International Conference on Computational Intelligence and Security Workshops, Washington DC: IEEE Computer Society, 2007: 19-22
- [11] HU Jing-song, WANG Jia-bing, HU Gui-wu. Time-varying modifying factor fuzzy controller based on the mine-sweeping strategy[C]//Proc of the 5th International Conference on Fuzzy Systems and Knowledge Discovery, 2008: 484-488
- [12] 粟塔山, 彭维杰, 周作益, 等. 最优化计算原理与算法程序设计[M]. 长沙: 国防科学技术大学出版社, 2001: 87-91.
- [13] 曾建潮, 崔志华. 一种保证全局收敛的 PSO 算法[J]. *计算机研究与发展*, 2004, 41(8): 1333-1338
- [14] 袁代林, 程世娟, 陈虬. 一种新形式的微粒群算法[J]. *计算机工程与应用*, 2008, 44(33): 57-59.
- [15] SHI Y u-hui, EBERHART R C. Empirical study of particle swarm optimization[C]//Proc of Congress on Evolutionary Computation, 1999: 1945-1950.