◎数据库、信号与信息处理◎

一种改进的基于粒子群优化的SVM训练算法

童 燕,李 映,白本督,张艳宁
TONG Yan,LI Ying,BAI Ben-du,ZHANG Yan-ning

西北工业大学 计算机学院,西安 710072
Department of Computer Science and Engineering,Northwest Polytechnical University,Xi'an 710072,China
E-mail:toyeah2004@126.com

TONG Yan,LI Ying,BAI Ben-du,et al.Improved particle swarm optimization for SVM training.Computer Engineering and Applications,2008,44(20):138-141.

Abstract: Since training a SVM requires solving a constrained quadratic programming problem which becomes difficult for very large datasets,an improved particle swarm optimization algorithm is proposed as an alternative to current numeric SVM training methods.In the improved algorithm,the particles studies not only from itself and the best one but also from the mean value of some other particles.In addition,adaptive mutation is introduced to reduce the rate of premature convergence.The experimental results show that the improved algorithm is feasible and effective for SVM training.

Key words: Support Vector Machine(SVM);particle swarm optimization algorithm;adaptive mutation

摘 要: 支持向量机的训练需要求解一个带约束的二次规划问题,但在数据规模很大的情况下,经典的训练算法将会变得非常困难。提出了一种改进的基于粒子群的优化算法,用于替代支持向量机中现有的训练算法。在改进后的粒子群优化算法中,粒子不仅向自身最优和全局最优学习,还以一定的概率向其他部分粒子的均值学习。同时,还引进了自适应变异算子,以降低未成熟收敛的概率。实验表明,提出的改进训练算法相对改进前的算法在性能上有显著提高。

关键词: 支持向量机;粒子群优化算法;自适应变异
DOI: 10.3778/j.issn.1002-8331.2008.20.042 **文章编号:** 1002-8331(2008)20-0138-04 **文献标识码:** A **中图分类号:** TP18

1 引言

支持矢量机(Support Vector Machine,SVM)[1,2]是Vapnik等人于20世纪90年代中期提出的一类新型机器学习方法,其理论基础是统计学习理论。与基于经验风险最小化原理的传统的统计学习方法不同,SVM基于的是结构风险最小化原理。SVM不仅结构简单,而且各种技术性能尤其是推广能力比神经网络等方法有明显提高。

标准的SVM的实现涉及求解线性约束的二次规划问题,该问题可以收敛到全局最优解。但当训练数据很多时——这是很多实际问题所碰到的情况,二次规划问题的求解受到存储器容量的限制,而且分类速度也得不到保证,从而限制了SVM在很多问题上的应用。常用的做法是将问题分解为若干个子问题,然后再对这些子问题进行逐一优化。这样做的缺陷是:结果可能只是一个次优解,并且分解的时候可能需要多次分解才能将问题的规模转化为能解决的程度。

粒子群优化算法[3,4](Particle Swarm Optimization,PSO)是由Kennedy和Eberhart提出的一种集群优化算法,它采用的是随机性的多点搜索,能够在高维空间中快速找到问题的最优解。利用PSO来训练SVM是在最近才提出来的,可以说是一种新颖的尝试。Paquet等人首先是提出了一种LPSO算法,用于求解带线性等式约束的优化问题[5],然后将这种算法用于SVM的训练[6]。但是仅仅利用基本的LPSO算法得到的训练结果并不理想,特别是在未成熟收敛问题上存在着困难。

本文提出了一种改进的LPSO算法,即ILPSO(Improved Linear Particle Swarm Optimization)。在这种改进后的算法中,不仅提出了一种全新的粒子学习方式,还加入了自适应变异算子,以增强全局搜索能力和跳出局部最优解。在Iris数据子集、双螺旋数据集及MNIST数据集上的分类实验证明了改进算法的良好性能。

2 支持向量机

不失一般性,以两分类问题为例,x_i设为训练样本,y_i是输

基金项目: 国家自然科学基金(the National Natural Science Foundation of China under Grant No.60472072);陕西省自然科学基础研究计划(the Natural Science Foundation of Shaanxi Province of China No.2006F05);航空科学基金(the Aeronautical Science Foundation No.05153076)。

作者简介: 童燕(1981-),女,硕士,主要研究领域为图像处理,模式识别,人工智能;李映(1969-),女,博士,副教授,主要研究领域为模式识别和信号处理,计算智能与混合系统;白本督(1972-),男,博士,主要研究领域为神经网络,小波分析及优化理论;张艳宁(1967-),女,博士,教授,博导,主要研究领域为信号与信息处理,图像处理,模式识别与计算机视觉。

收稿日期: 2007-09-28 **修回日期:** 2007-12-11

入样本 x_i 的类标记,即:

$$\{(x_1,y_1),\cdots,(x_n,y_n)\}\subset R^N\times\{+1\} \tag{1}$$

SVM 算法的出发点是利用核技巧在高维空间里进行有效的计算,找出支持向量及其系数构造最优分类面。而此最优分类面的构造问题实质上是在约束条件下求解一个二次优化问题,以得到一个最优的决策函数:

$$f(x)=\mathrm{sgn}(\sum_{i=1}^{n}\alpha_i y_i K(x_i,x)+b) \tag{2}$$

其中,K(·,·)为核函数,n 为训练样本的数目,b 是一个阈值,系数 α_i 则是二次优化问题

$$\begin{aligned}&\mathrm{Max}(Q(\alpha))=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_i,x_j)\\&\text{s.t.}\sum_{i=1}^{n}\alpha_i y_i=0(i=1,2,\cdots,n)\\&\text{s.t.}0\leqslant\alpha_i\leqslant C(i=1,2,\cdots,n)\end{aligned} \tag{3}$$

的最优解。C 为惩罚因子,它控制的是训练错误率与模型复杂度间的折衷。容易证明,该优化问题的解中只有一部分(通常是很少的部分)α_i 不为零。这些不为零的 α_i 对应的样本就是支持向量,只有支持向量影响最终的划分结果。

3　用于求解线性等式约束优化问题的 LPSO 算法

3.1　基本 PSO 算法

基本粒子群优化是基于群体智能理论的随机优化算法,其智能性就体现在整个群体之间的合作上[6]。由于 PSO 的内在并行性本质和它对于目标优化函数的独立性特点,因此用于求解高维空间中的非线性优化问题是非常适用的。

基本 PSO 算法可描述如下:问题的解对应于搜索空间中一只鸟的位置,称这些鸟为"粒子"。每个粒子都有自己的位置和速度,分别记为 $P_i=(P_{i1},P_{i2},\cdots,P_{in})$ 和 $V_i=(v_{i1},v_{i2},\cdots,v_{in})$,$i=1,2,\cdots,m$,$m$ 为粒子的个数,n 为粒子的维数。每个粒子还有一个由被优化函数决定的适应值。粒子根据如下的式(4)和式(5)来更新自己的速度和位置:

$$v_{i,j}^{(t+1)}=w^t*v_{i,j}^{(t)}+c_1r_{1j}^{(t)}(pbest_{i,j}^{(t)}-p_{i,j}^{(t)})+c_1r_{2j}^{(t)}(pg_j^{(t)}-p_{i,j}^{(t)}) \tag{4}$$

$$p_{i,j}^{(t+1)}=p_{i,j}^{(t)}+v_{i,j}^{(t+1)} \tag{5}$$

其中 t 是迭代次数,为惯性权重,c_1、c_2 是学习因子,通常取 $c_1=c_2=2.05$;r_{1j} 和 r_{2j} 为[0,1]之间均匀分布的随机数,体现了算法的随机性。$pbest_i=(pbest_{i1},pbest_{i2},\cdots,pbest_{in})$ 为粒子 i 搜索到的最佳位置,$pg=(pg_1,pg_2,\cdots,pg_n)$ 为整个粒子群到目前为止搜索到的最佳位置。另外,已经有人提出了一种带压缩因子[7]的 PSO 算法,使用下述公式(6)替代公式(4):

$$v_{i,j}^{(t+1)}=K[v_{i,j}^{(t)}+c_1r_{1,j}^{(t)}(pbest_{i,j}^{(t)}-p_{i,j}^{(t)})+c_2r_{2j}(pg_i^{(t)}-p_{i,j}^{(t)})] \tag{6}$$

上式中 K 称为压缩因子。Eberhart R C 等人已经证实带压缩因子的 PSO 算法对于限定粒子每一维上的速度起着很重要的作用,因此在本文中采用的是有压缩因子的速度更新公式。

3.2　用于求解线性等式约束优化问题的 LPSO 算法

由于基本的 PSO 只能用于求解无约束的优化问题,文献[5]提出了一种带线性等式约束的 LPSO 算法,将基本 PSO 中的速度更新公式变为:

$$v_i^{(t+1)}=w^t*v_i^{(t)}+c_1r_1^{(t)}(pbest_i^{(t)}-p_i^{(t)})+c_1r_2^{(t)}(pg^{(t)}-p_i^{(t)}) \tag{7}$$

与基本 PSO 算法的不同之处在于:速度更新公式中对于每个分量的更新都采用相同的随机数 r_1 和 r_2;粒子的速度初始化为 0;粒子的位置在初始化时也必须满足给定的线性等式约束条件 $AP=b$,$A\in R^{l\times n}$,$b\in R^l$。

LPSO 算法的基本步骤可以描述为:

(1)初始化:初始化起始搜索点的速度 V_i 及位置 P_i。速度初始化为 0,位置初始化时必须满足上述线性等式条件。

(2)评价:计算每个粒子的适应值,比较个体极值和当前适应值,将更优的一个设置为新的 $pbest_i$。全局极值则是个体极值中最好的。

(3)更新:用公式(7)和(5)对每一个粒子的速度和位置进行更新。

(4)检查:如果当前的迭代次数达到了预先设定的最大次数(或达到最小错误要求),则停止迭代,输出最优解,否则转到(2)。

4　ILPSO 算法及其在 SVM 中的应用

4.1　ILPSO 算法

PSO 算法在搜索的初期收敛速度很快,但在后期却易于陷入局部最小,这是 PSO 算法的主要缺点。存在此缺点的原因在于:算法在运行过程中,如果某粒子搜索到了一个当前最优位置,其他粒子便会迅速向其靠拢。如果该最优位置恰好是局部最优点时,整个粒子群就会陷入一种停滞不前的状态,即出现所谓的早熟收敛现象。此时粒子群中的粒子都会出现"聚焦"现象,要么所有粒子都聚集在某一特定位置,要么聚集在某几个特定位置[8]。因此,本文对 LPSO 算法提出以下几点改进以缓解未成熟收敛问题:

(1)提出了一种新的粒子学习方式:在迭代的前期,粒子以较大的概率向自身最优和其他部分粒子的均值学习;而在迭代的后期,为了趋于收敛,粒子则以较大的概率向自身最优和全局最优学习。

(2)引进自适应变异策略。即先计算粒子群的群体适应度[9]:

$$\sigma^2=\frac{1}{n}\sum_{i=1}^{n}\left(\frac{f_i-f_{avg}}{f}\right)^2 \tag{8}$$

其中 f_i 为第 i 个粒子的适应度,f_{avg} 为粒子群目前的平均适应度,f 为归一化定标因子,用于限制 σ^2 的大小,取值为:$f=\max\{\max\{|f_i-f_{avg}|\},1\}$。

然后计算变异概率 p_m[9]:

$$p_m=\begin{cases}k,\sigma^2<\sigma_d^2\\0,\text{其它}\end{cases} \tag{9}$$

其中 k 是取值在[0.1,0.3]之间的常数;σ_d^2 是一个阈值,应根据不同的实际问题来取值。

如果变异概率达到足够大,其中一部分粒子就会进行重新初始化来达到变异的目的。

4.2　ILPSO 在训练 SVM 中的应用

SVM 的数学本质即是要求解支持向量系数,故将粒子编码为 $P=(\alpha_1,\alpha_2,\cdots,\alpha_n)$,适应度函数为

$$f(P)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_i,x_j) \tag{10}$$

ILPSO 算法用于训练 SVM 的基本流程如下:

(1)初始化粒子群中的所有粒子。根据 LPSO 算法的要求,粒子的速度均初始化为0。为了满足粒子在搜索初期的多样性

需求,对粒子在进行初始化时尽量使其均匀分布,即在[0,C]之间随机产生粒子的位置,然后再对粒子的位置进行微调,使其满足线性约束条件 $\sum_{i=1}^{n}\alpha_i y_i=0$。置当前迭代次数 $t=0$。

(2)对每个粒子执行下述操作:

①计算每个粒子的适应度值,并且更新 $pbest_i$;

②产生随机数 r,并计算 $R_t=t/max\_iterations$,如果 $r>R_t$,则采用公式

$$v_i^{(t+1)}=K[v_i^{(t)}+c_1r_1^{(t)}(pbest_i^{(t)}-p_i^{(t)})+c_2r_2^{(t)}(pavg_i^{(t)}-p_i^{(t)})] \quad (11)$$

更新粒子的速度,否则采用下述公式(12)进行更新:

$$v_i^{(t+1)}=K^*[v_i^{(t)}+c_1r_1^{(t)}(pbest_i^{(t)}-p_i^{(t)})+c_2r_2^{(t)}(pg^{(t)}-p_i^{(t)})] \quad (12)$$

公式(11)中 $pbest_i$ 为随机产生的 1/5 粒子的位置平均值。对于任何超出速度范围的粒子 $v_{ij}<velocity\_min$ 或 $v_{ij}>velocity\_max$ 进行惩罚,使其重新落入[$velocity\_min$,$velocity\_max$];

③用公式(5)更新粒子的位置矢量:

对于任何超出位置范围的粒子 $p_{ij}<0$ 或 $p_{ij}>C$ 进行惩罚,使其重新落入[0,C],并且像初始化时一样,对整个群体的位置矢量进行微调,使其再次满足线性约束条件;

④计算相应的 pg 值。

(3)分别根据公式(8)和(9)计算群体适应度方差 σ^2 和自适应变异概率 p_m。

(4)产生随机数 $r\in[0,1]$,如果 $r<p_m$,随机选取 1/5 的粒子(不包括最优粒子)进行变异——位置重新初始化。否则,转向步骤(5)。

(5)置当前迭代次数 $t=t+1$,如果 $t>max\_iterations$,则 pg 记录的即为 SVM 的训练结果,否则返回步骤(2)重复上述过程。

5 实验结果与分析

这里给出了三组实验结果。第一组实验是在 Iris 数据子集(Iris 数据集的一部分)上进行的,第二组是对典型的双螺旋问题进行的实验,第三组则给出手写体字符数据集的实验结果。

实验中 SVM 的核函数均采用高斯核函数:

$$K(X)=\exp(-\frac{|X-X_i|}{2\sigma^2}) \quad (13)$$

实验中,粒子的个数为 50;其他参数设置为:加速系数 $c_1=c_2=2.05$,压缩因子 $K=0.73$,变异概率常数 $k=0.3$,粒子群体适应度阈值 $\sigma_d^2=0.2$,粒子最小速度 $velocity\_min=-4$,粒子最大速度 $velocity\_max=4$。

5.1 Iris 数据集的分类实验

此数据集合是从 Iris 数据集中选取 Iris-setosa 和 Iris-versicolor 两类样本得到的,包括 40 个训练样本和 60 个测试样本。分别利用 ST_SVM(标准 SVM)、LPSO_SVM 和 ILPSO_SVM 三种方法得到的分类结果如图 1 所示,实验统计数据见表 1。在 ST_SVM 算法中 $C=100$,$\sigma=0.5$;在 LPSO_SVM 和 ILPSO_SVM 算法中 $C=100$,$\sigma=0.5$。不难看出,利用所提出的算法得到的分类线跟最佳分类线非常接近,而支持向量的数目则大大降低了。

表 1 不同分类方法得到的统计数据(1)

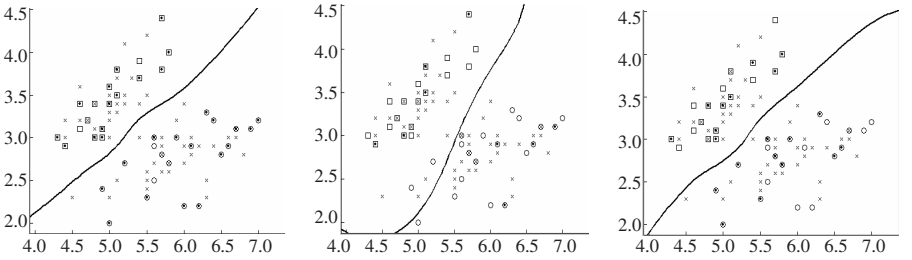
方法	支持向量个数/个	支持向量比率/%	识别个数/个	识别率/%
ST_SVM	31	77.50	59	98.33
LPSO_SVM	9	22.50	57	95.00
ILPSO_SVM	21	52.50	59	98.33

5.2 双螺旋数据集的分类实验

此数据集是一种典型的非线性可分数据集合。本组数据中训练样本为 316 个,测试数据为 156 个。分别利用 ST_SVM、LPSO_SVM 和 ILPSO_SVM 三种方法得到的分类结果如图 2 所示,实验统计数据见表 2。在 ST_SVM 算法中 $C=100$,$\sigma=0.9$;在 LPSO_SVM 和 ILPSO_SVM 算法中 $C=2.5$,$\sigma=0.9$。很容易发现,

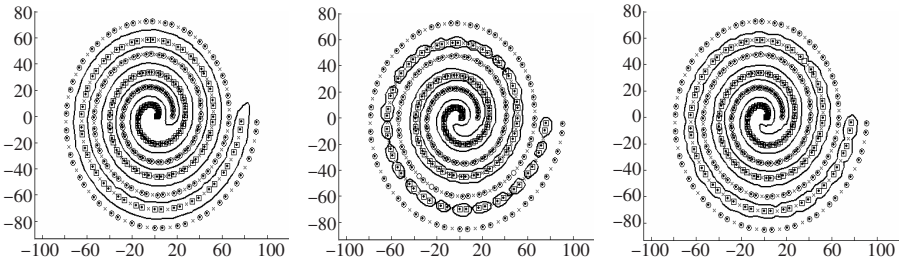
表 2 不同分类方法得到的统计数据(2)

方法	支持向量个数/个	支持向量比率/%	识别个数/个	识别率/%
ST_SVM	316	100.00	156	100.00
LPSO_SVM	310	98.10	144	92.31
ILPSO_SVM	309	97.78	156	100.00



(a)ST_SVM 训练结果 (b)LPSO_SVM 训练结果 (c)ILPSO_SVM 结果

图 1 不同分类方法得到的训练结果(1)



(a)ST_SVM 训练结果 (b)LPSO_SVM 训练结果 (c)ILPSO_SVM 结果

图 2 不同分类方法得到的训练结果(2)

利用提出的算法获得的识别率和最佳解得到的识别率是一样的,但是支持向量个数却更少。

5.3 MNIST 数据集的分类实验

MNIST 是 AT&T 贝尔实验室的手写数字识别评测数据库。样本来自 250 个人，共 60 000 个训练样本和 10 000 个测试样本,每个样本是一个 28×28 大小的灰度图像,构成一个 784 维的向量。从训练样本中选取 10 000 个数据用于训练,并将数字“5”作为一类,其余数字作为另一类构成两类问题。实验统计数据见表 3。在 ST_SVM 算法中 C=100,σ=0.2;在 LPSO_SVM 和 ILPSO_SVM 算法中 C=2.5,σ=0.2。可以发现,对于这样的大数据集，经典的 ST_SVM 解法不再可行——发生内存溢出错误,而利用 ILPSO_SVM 算法则能正常求解，同时也获得了较高的识别率。

表 3　不同分类方法得到的统计数据(3)

方法	支持向量个数/个	支持向量比率/%	识别个数/个	识别率/%
ST_SVM	–	–	–	–
LPSO_SVM	990 2	99.02	911 3	91.13
ILPSO_SVM	995 1	99.51	962 5	96.25

上述实验表明:本文提出的 ILPSO_SVM 算法用于 SVM 训练过程之中时,克服了 SVM 本身存在的缺陷,能在取得次优解的基础上获得相当甚至更高的识别率,并在一定程度上减少了支持向量的数目。相对于 LPSO_SVM 算法来说,在算法性能方面有很大提高。本文提出的算法并没有从速度上进行优化处理,所以在运行时间方面大大超过了 SVM 训练算法本身,这也是今后研究工作的一部分。

6　结束语

本文针对 SVM 训练问题,提出了一种改进的 LPSO 算法,对原始的 LPSO 算法的速度更新方式进行了修改,并加入了自适应变异算子，在一定程度上缓解了未成熟收敛问题。在 Iris 数据集、双螺旋数据集及 MNIST 手写数字数据集上的实验结果表明,本文提出的改进训练算法相对改进前的算法在性能上有显著的提高。

参考文献:

[1] Vapnik V N.The nature of statistical learning theory[M].New York:Springer-Verlag,1995.

[2] Cortes C,Vapnik V N.Supporter vector networks[J].Machine Learning,1995,20(3):273-297.

[3] Kennedy J,Eberhart R C.Particle swarm optimization[C]//Proceedings of IEEE International Conference on Neural Networks,1995,4:1942-1948.

[4] Shi Y,Eberhart R C.A modified particle swarms optimizer[C]//Proceedings of IEEE Conference on Evolutionary Computation,Anchorage,Alaska,May 4-9,1998:69-73.

[5] Paquet U,Engelbrecht A P.A new particle swarm optimizer for linearly constrained optimization[C]//Proceedings of IEEE Conference on Evolutionary Computation,2003,1:227-233.

[6] Paquet U,Engelbrecht A P.Training support vector machines with particle swarms[C]//Proceedings of International Joint Conference on Neural Networks,2003,2:1593-1598.

[7] Eberhart R C,Shi Y.Comparing inertia weights and constriction factors in particle swarm optimization[C]//Proceedings of IEEE Conference on Evolutionary Computation,2000,1:84-88.

[8] Pasupuleti S,Battiti R.The Gregarious Particle Swarm Optimizer (G-PSO)[C]//Proceedings of GECCO'06,2006,1:67-74.

[9] 吕振肃,侯志荣.自适应变异的粒子群优化算法[J].电子学报,2004,32(3):416-420.

(上接 22 页)

于最小生成树思想的求解 RSMT 的近似算法计算所得结果仅比最小生成树改进 7%~10%,因此说明本算法对于 RSMT 具有良好的求解效果。此外,从图 4(a)~图 4(d)可以看出,随着循环次数的增加,平均 Steiner 比基本呈现递减的趋势(对于图(a),由于数据的特殊也可能存在最小树长变化不大的现象)。而循环次数的增加则意味着计算时间的延长,因此应根据需求的不同来确定时间与效益的舍与取。此外,由于算法具有一定的随机性，在不同循环次数下进行单次运算或运算次数较少时,可能会有循环次数较少情况下所得计算结果反而优于循环次数较多时所求得的解的情况出现。

5　结束语

本文给出了 RSMT 问题的元胞蚂蚁算法求解过程。由于算法利用元胞自动机的原理,采用元胞演化规则进行 s 点的选取与定位,可使蚂蚁逐步集中于某些区域,加快优化过程。从算法的寻优机理不难看出，其核心过程能利用搜索空间局部特性和邻域的扩大,防止搜索陷入局部最优,从而具有更强的全局优化能力。

参考文献:

[1] Gilbert E N,Pollak H O.Steiner minimal trees[J].SIAM J Appl Math,1968,16(1):1-29.

[2] Fößmeier U,Kaufmann M.On exact solutions for the rectilinear Steiner tree problem Part I:theoretical results [J].Algorithmica,2000,26(1):68-99.

[3] Garey M R,Johnson D S.The rectilinear Steiner tree problem is NP-complete[J].SIAM Journal on Applied Mathematics,1977,32(4):826-834.

[4] Zachariasen M,Rohe A.Rectilinear group Steiner trees and applications in VLSI design[J].Mathematical Programming,2003,94(2/3):407-433.

[5] Hanan M.On Steiner's problem with rectilinear distance[J].SIAM Journal on Applied Mathematics,1966,14(2):255-265.

[6] Snyder T L.On the exact location of Steiner points in general dimension[J].SIAM J Comput,1992,21(2):163-180.

[7] Hwang F K.On Steiner minimal trees with rectilinear distance[J].SIAM Journal on Applied Mathematics,1976,30(1):104-114.

[8] Von Neumann J,Burks A W.Theory of self-reproducing automata[M].Illinois,USA:University of Illinois,1966.

[9] Wolfram S.Theory and application of cellular automata[M].Singapore:The World Scientific Publishing Co Ltd,1986.

[10] 朱刚,马良.TSP 的元胞蚂蚁算法求解[J].计算机工程与应用,2007,43(10):79-80.

[11] 朱刚,马良.函数优化的元胞蚂蚁算法[J].系统工程学报,2007,22(3):305-308.