基于二进制交叉和变异的粒子群算法及应用

刘衍民[1,2] 牛 奔[3] 赵庆祯[2]
(遵义师范学院数学系 遵义 563002)[1] (山东师范大学管理与经济学院 济南 250014)[2]
(深圳大学管理学院 深圳 518060)[3]

摘 要 粒子群算法在求解多峰问题时极易陷入局部最优解,提出了基于模拟二进制交叉和多项式变异的粒子群算法(SPDPSO)。在该算法中,为了更好地利用每个粒子的历史信息,引入了外部存档存储每个粒子的最优位置(*pbest*);同时,对外部存档中的 *pbest* 进行二进制交叉,而对新产生的全局最优粒子进行多项式变异。基准函数的测试结果显示,SPDPSO 算法在求解多峰问题上有一定的优势。在实际应用中,以 TSP 为研究对象,结果显示 SPDPSO 算法获得了比其它算法更好的解。

关键词 粒子群算法,模拟二进制交叉,多项式变异
中图法分类号 TP301 **文献标识码** A

Particle Swarm Optimizer with Simulated Binary Crossover and Polynomial Mutation and its Application

LIU Yan-min[1,2] NIU Ben[3] ZHAO Qing-zhen[2]
(Department of Math, Zunyi Normal College, Zunyi 563002, China)[1]
(School of Management and Economics, Shandong Normal University, Jinan 250014, China)[2]
(College of Management, Shenzhen University, Shenzhen 518060, China)[3]

Abstract PSO may easily get trapped in a local optimum, when it comes to solving multimodal problems. In view of the default, we presented a variant of particle swarm optimizer(PSO) with simulated binary crossover and polynomial mutation(SPDPSO for short). In SPDPSO, additionally, the external archive was introduced to store the personal best performing particle(*pbest*), and simulated binary crossover and polynomial mutation were used to produce new particles. In benchmark function, the results demonstrate good performance of the SPDPSO algorithm in solving complex multimodal problems compared with the other algorithms. In practical application, the experimental results show that the SPDPSO algorithm can achieve better solutions that other PSOs.

Keywords Particle swarm optimizer, Simulated binary crossover, Polynomial mutation

1 引言

最优化问题是进化计算领域的一个研究热点。随着现实世界的优化问题变得越来越复杂,对优化问题的研究也提出了更高的要求。粒子群算法(Particle swarm optimization-PSO)[1]是一种相对新的优化技术,它概念简单,控制参数少,寻优结果与初值无关,具有一定的并行性等特点,自提出以来便受到了学术界的广泛关注。其中一种较为实用的 PSO 变型是将惯性重量 ω 引入原始算法中,通过惯性权重 ω 来权衡粒子的全局和局部搜索能力[2]。Peram 等[3]提出了适应距离比例(Fitness-Distance-Ratio)的带有近邻合作 PSO 算法(FDR-PSO),其通过调整适应距离比例来提升粒子的学习能力。Mendes 和 Kennedy[4]提出了一种完全信息粒子群,在该算法中,每个粒子的所有邻居都参与位置更新,以更好地利用种群中每个粒子的信息。文献[5]提出了一种共协作粒子群最优法(CPSO-H)。该算法用一维(1-D)群来独立地搜索每一维变量,这些搜索结果按照事先规定的原则结合在一起,以提升算法的运行效率。上述这些改进算法在求解大多数优化问题时表现十分出色,但是,它在求解复杂的多峰问题时极易陷入局部最优解。为了克服这个缺点,本文在文献[6]提出的算法的基础上,提出一种基于模拟二进制交叉(Simulated Binary Crossover, SBC)和多项式变异(Polynomial Mutation Operator, PMO)的改进粒子群算法。

2 基于模拟二进制交叉和多项式变异的 PSO 算法

2.1 模拟二进制交叉和多项式变异

2.1.1 多项式变异

一个种群停止进化时(即当种群中所有粒子的速度几乎等于零时),将导致整个种群陷入局部最优解。在文献[7]中,作者提出在这种状态下,通过变异来产生新的全局最优粒子(*gbest*)使得当前种群跳出局部最优状态,通过实验分析,验证了这是一种有效的方法。然而值得注意的是变异产生的新

到稿日期:2010-06-19 返修日期:2010-09-27 本文受国家 863 项目(2008AA04A105)和贵州教育厅社科项目(0705204)资助。
刘衍民(1978—),男,博士,讲师,主要研究方向为运筹学理论、进化计算,E-mail:yanmin7813@163.com;牛 奔(1980—),男,博士,硕士生导师,主要研究方向为进化计算;赵庆祯(1943—),男,博士,教授,博士生导师,主要研究方向为运筹学理论、进化计算、物流工程等。

粒子或许远离全局最优解，因此，在 SPDPSO 算法中，不同于通常的变异策略(一个粒子以一定的概率进行变异)，我们提出的策略如下：当粒子的 *gbest* 连续 n 代没有得到提升时，对新产生的 *gbest*(*Ngbest*)进行多项式变异。在这里，经过试验，n 的取值为 4。多项式变异因子[8]的表达式如下：

$$x_i(t+1)=x_i(t)+(x_i^U(t)-x_i^L(t))\cdot\delta_i \tag{1}$$

式中，$x_i(t)$表示在迭代时刻 t 当前粒子 i 的位置。$x_i^U(t)$和 $x_i^L(t)$分别表示 $x_i(t)$的上界和下界。δ_i是粒子 i 的扰动项，它的生成见式(2)。

$$\begin{aligned}&\delta_k=(2r_k)^{\frac{1}{\eta_m+1}}-1,\text{if } r_k<0.5\\&\delta_k=1-[2\cdot(1-r_k)]^{\frac{1}{\eta_m+1}},\text{if } r_k\geqslant 0.5\end{aligned} \tag{2}$$

式中，r_k表示(0,1)之间均匀分布的随机数。η_m 为变异分布指数，它等于人口规模。

2.1.2 模拟二进制交叉

为了更好地利用 *pbest* 的每一维信息，避免“Two steps forward, one step back”现象[5]，本文提出一个大胆的设想，即在每一次迭代中，采用外部存档存储产生的 *pbest*，并且依据 SBC 准则对外部存档中的 *pbest* 应用模拟二进制交叉[11]产生新的粒子。新产生的粒子融合了每次迭代产生的 *pbest* 的各维信息，增强了粒子间各维信息之间的交流，有效地提升了粒子跳出局部最优解的能力。文献[9]也已经证实 SBC 在帮助种群跳出局部最优解、避免早熟收敛方面起到了积极的作用。模拟二进制交叉准则表达式如下：

$$\begin{aligned}x_{1,k}&=\frac{1}{2}[(1-\beta_k)\cdot x_{r,1,k}+(1+\beta_k)x_{r,2,k}]\\x_{2,k}&=\frac{1}{2}[(1+\beta_k)\cdot x_{r,1,k}+(1-\beta_k)x_{r,2,k}]\end{aligned} \tag{3}$$

式中，$x_{r,i,k}$是随机选择的粒子。$x_{i,k}(i=1,2)$表示在 SBC 后新产生的粒子的第 k 维。$\beta_k(\geqslant 0)$ 通过等式(4)产生。

$$\begin{aligned}\beta(u)&=(2u)^{\frac{1}{\eta_c+1}}\\\beta(u)&=[2(1-u)]^{\frac{-1}{\eta_c+1}}\end{aligned} \tag{4}$$

式中，u 是(0,1)之间的随机数，η_c 定义新产生的粒子的分布指数，在 SPDPSO 算法中，它等于粒子规模。

2.1.3 模拟二进制交叉和多项式变异的作用分析

为了检测模拟二进制交叉和多项式变异的作用，5 个 30 维变量的 Benchmark 函数(Sphere, Rosenbrock, Ackley, Griewanks 和 Rastrigin)用来检测如下 4 种 PSO 算法：具有动态邻居的粒子群算法(DPSO)；具有动态邻居和 PMO 的粒子群算法(PDPSO)；具有动态邻居和 SBC 的粒子群算法(SDPSO)；具有动态邻居、SBC 和 PMO 的粒子群算法(SPDPSO)。这里，每个检测函数独立运行 30 次，最大的函数评价次数(Fitness evaluations, FEs)设置为 3×10^4。由于不同检测函数最优值的量纲不同，因此对所有的适应函数值用式(5)进行标准化处理，以消除量纲的影响；同时将标准化的数据按照 Mendes 的方法[4]进行合并。

$$X=\frac{x_{ij}-\mu_{ij}}{\sigma_{ij}} \tag{5}$$

式中，X 表示标准化数值；x_{ij} 表示第 j 个算法在第 i 个检测函数上的仿真值；μ_{ij} 和 σ_{ij} 分别表示 x_{ij} 的均值和标准差。

由于所有的检测函数都是最小化问题，因此标准化数值越小，意味着运行效果越好。正如图 1 所示，SPDPSO 算法有最好的结果，PDPSO 和 SDPSO 算法几乎有相同的结果，DPSO 算法运行效果最差。因此，可以推断出模拟二进制交叉和多项式变异能够提升算法的运行效率。

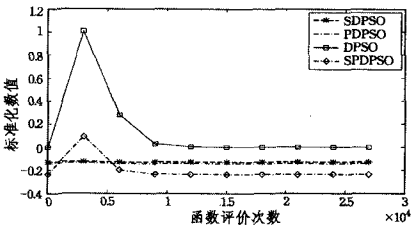


图 1 不同算法的标准化数值

2.2 基于模拟二进制交叉和多项式变异的 PSO 算法

为了提升粒子跳出局部最优解的能力，采用文献[6]提出的学习样本选择策略，即一个粒子的学习样本不是它邻居中表现最好的一个粒子，而是该粒子的所有邻居(包括它自己)。这样，每个粒子的速度更新过程按照式(6)、式(7)来进行。

$$\begin{aligned}\overline{v_i}(t)=&w\cdot\overline{v_i}(t-1)+\varphi_1\cdot r_1(\overline{p_g}-\overline{x_i}(t-1))+\\&\varphi_2\cdot r_2(\overline{p_{bin_{(i)}}}-\overline{x_i}(t-1))\end{aligned} \tag{6}$$

$$\overline{x_i}(t)=\overline{x_i}(t-1)+\overline{v_i}(t)$$

$$p^d_{bin_{(i)}}=\arg\{\max[\frac{Fitness_k(p_j)-Fitness_k(p_i)}{|p_{jd}-x_{id}|}]\} \tag{7}$$

$$d\in(1,2,\cdots,n)\quad i=1,2,\cdots,ps,j\in neighbor_i,k=1,\cdots,m$$

式中，$\overline{p_g}$表示群中表现最好的粒子，$\overline{p_{bin_{(i)}}}$表示粒子 i 每一维学习对象，$neighbor_i$ 表示粒子 i 的邻居构成的集合，$p^d_{bin_{(i)}}$ 表示$\overline{p_{bin_{(i)}}}$的第 d 维。ps 表示种群规模，$\overline{v_i}(t)=(v_i^1,v_i^2,\cdots,v_i^n)$表示第 i 个粒子在迭代时刻 t 的速度，$\overline{x_i}(t)=(v_i^1,v_i^2,\cdots,v_i^n)$表示第 i 个粒子在迭代时刻 t 的位置，$\overline{p_g}=(p_g^1,p_g^2,\cdots,p_g^n)$表示在迭代时刻 t 整个种群所发现的最优运行的粒子，φ_1 和 φ_2 表示加速常数，r_1 和 r_2 表示在(0,1)之间的随机数，$\overline{p_j}=(p_j^1,p_j^2,\cdots,p_j^n)j\in neighbor_i$表示粒子 i 的邻居中，任何一个粒子的 *pbest*，$Fitness(p)$表示向量 p 的函数值。这里，$w=0.729$，$\varphi_1=\varphi_2=1.4945$。这种学习策略使整个群体拥有更广阔的搜索空间，我们将其称为广泛学习策略。算法 1 给出了 SPDPSO 算法的伪代码。

算法 1

```
Initialize positions and associated velocities of all particles
While(fitcount < Max_FES) &&(k<iteration)
    If Mstaynum= 4
        Mstaynum= 0
        Ngbest begins the polynomial mutation
    End
    For each particle(i=1:ps)
        Select an exemplar from external archive
        Updating velocity and position, pbest and Mstaynum
    End for
    For i=1:N/2
        Randomly choose 2 pbest in the external archive
        Simulated binary crossover
    End for
End While
```

3 仿真实验及其分析

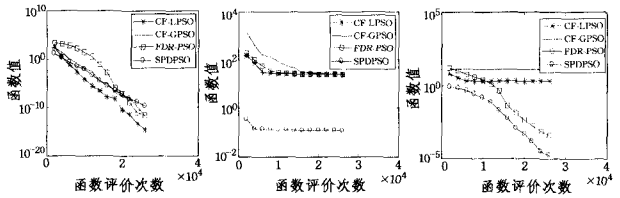
3.1 Benchmark 函数和相比较的算法

为了测试提出的算法，我们选择 2 个单峰和 5 个多峰 Benchmark 函数。它们的表达式见文献[3,4]。将 SPDPSO 算法与 3 种有代表性的 PSO 算法进行比较，它们分别是：CF-LPSO[2]，CF-GPSO[2]，FDR-PSO[3]。为了使不同的算法具有可比性，实验的相关设置如下：所有 PSO 算法的种群规模为

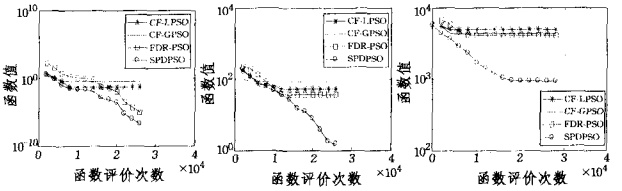
30 个粒子，每个检测函数独立运行 30 次，每次运行 3×10^4 次函数评价。

3.2 实验结果及分析

表 1 给出了检测函数在 3×10^4 次函数评价后的均值和 95%的置信区间，最好的运行结果用黑体字标出。表 2 给出了各种算法的运行时间，计算方法为：在型号为 ThinkPad-SL400 电脑上应用 Matlab7. 0. 1. 24704(R14)Service Pack 1 软件中的(tic,toc)命令。图 2 给出了 PSO 算法收敛特征曲线图。



(a)Sphere 函数　(b)Rosenbrock 函数　(c)Ackley 函数



(d)Griewanks 函数　(e)Rastrigin 函数　(f)schewfel 函数

图 2　非旋转检测函数收敛特征图

从图 2 可以看出，对于所有的非旋转的 Benchmark 函数，除了 Sphere 和 Schewfel 函数外，SPDPSO 算法明显优于其它 3 种算法，尤其对 Rosenbrock 函数，SPDPSO 算法表现出快速的收敛特性，这是由于其采用了 SBC 和 PMO 操作，使得每个粒子都能够更好地进行局部搜索，提升了粒子跳出局部最优解的能力。

表 1　检测函数在 3×10^4 次函数评价后的均值和置信区间

算法	Sphere(f_1)	Rosenbrock(f_2)	Ackley(f_3)	Griewanks(f_4)	Rastrigin(f_5)	Schewfel(f_6)
CF-LPSO	4. 5502e−017± 1. 3201e−017	2. 4218e+001± 1. 1236e+001	1. 8978e+000± 1. 1374e+000	4. 6610e−002± 3. 7631e−002	5. 1738e+001± 2. 5426e+001	4. 9945e+003± 2. 5123e+003
CF-GPSO	4. 7546e−014± 2. 3421e−014	2. 4078e+001±1 1. 7456e+001	1. 3769e+001± 2. 8168e+001	3. 2886e−001± 1. 4561e−001	8. 2581e+001± 1. 6346e+001	4. 3474e+003± 1. 9653e+003
FDR-PSO	9. 0614e−017± 2. 195e−017	2. 3227e+001± 1. 0211e+001	1. 3796e−004± 2. 1511e−004	1. 3762e−006± 2. 6731e−006	3. 5819e+001± 2. 5643e+001	4. 1454e+003± 2. 5184e+003
SPDPSO	1. 5392e−011± 1. 0836e−011	9. 8654e−001± 4. 6832e−001	3. 5621e−006± 1. 9653e−006	4. 6139e−008± 2. 6183e−008	5. 4832e−001± 2. 3345e−001	9. 3891e+002± 4. 6631e+002

表 2　达到停止标准时各种算法运行时间(单位：秒)

算法	f_1	f_2	f_3	f_4	f_5	f_6
CF-LPSO	7	12	19	29	50	36
CF-GPSO	8	10	18	27	46	37
FDR-PSO	9	9. 8	20	35	45	35
SPDPSO	8. 1	10	20	30	51	49

从表 1 也可以看出 SPDPSO 算法在函数 f_2，f_3，f_4，f_5 上都取得了最好的运行结果。另外，SPDPSO 算法融合了动态邻居拓扑结构、二进制交叉和变异操作，从而极大地提升了算法跳出局部最优解的能力。因此，探讨这些策略是否增加了算法的计算复杂性极其重要，这里我们利用 Matlab 软件中的(tic,toc)命令来度量达到停止标准时，每种算法的所需时间。停止标准为：每种算法运行 3×10^4 次函数评价。从表 2 可以明显地看出 SPDPSO 算法在所有的 Benchmark 函数上的运行时间与其它算法在同一数量级，引入的策略并没有增加算法的计算复杂度。

4　SPDPSO 算法在求解 TSP 问题中的应用

实验选用了 TSPLIB 中几个具有代表性的实例，所有算法均用 Matlab 软件编程实现。对于 PSO 算法求解 TSP 问题的方法，我们采用文献[10]提出的策略。所有算法的种群规模为 30 个粒子，每个实例独立运行 10 次，每次运行 6×10^3 次函数评价。为了比较各种算法的运行效率，采用通用的偏差计算式(8)作为解的好坏的评价标准。表 3 给出了各种 PSO 算法在 TSP 实例上的搜索性能，这里 n 表示城市数量，T(单位：秒)表示运行时间。

表 3　PSO 算法在 TSP 实例上的搜索性能比较

实例	n	S_0	CF-LPSO		CF-GPSO		FDR-PSO		SPDPSO	
			σ%	T	σ%	T	σ%	T	σ%	T
Att532	532	27686	0. 3925	33	0. 4764	40	0. 0256	73	0. 0011	54
Pcb1173	1173	56892	0. 9753	147	1. 4361	159	0. 1458	182	0. 0034	171
Vm1748	1748	336556	1. 2937	391	1. 7842	432	0. 2748	607	0. 0067	568
Pr2392	2392	378032	1. 8642	564	2. 1479	595	0. 3284	1137	0. 0246	998
Rl5934	5934	556045	2. 9467	835	3. 9964	987	0. 4023	1643	0. 0863	1452
	平均值		1. 4945	394	1. 9682	443	0. 2354	728	0. 0244	649

$$\sigma\%=(\sum_{i=1}^{N}(S_i-S_0))/(N\times S_0)\times100\% \qquad (8)$$

式中，N 表示独立实验的次数。S_0 表示已知最优解路径长度。S_i 为每次实验得到的最优解路径长度。

从表 3 中，根据偏差(σ%)可以看出，SPDPSO 算法求得 TSP 问题解的质量要优于其它 5 种 PSO 算法。在运行时间上 SPDPSO 要优于 FDR-PSO，但是差别不大，几乎都在同一数量级上。它要比 CF-LPSO 和 CF-GPSO 算法耗费更多的时间，但这种时间代价换来了解的质量提高。

结束语　本文提出了求解多峰问题的改进粒子群算法(SPDPSO)，它通过引入外部存档来存储在每一代产生的个体最优位置(*personal best*，*pbest*)，并且对外部存档中的 *pbest* 进行二进制交叉，以增强粒子间信息的交流；最后对新产生的全局最优位置(*global best*，*gbest*)进行多项式变异操作，来产生新的全局最优粒子。从基准函数仿真结果和实际应用来看，这些策略提升了 PSO 算法求解多峰问题的能力。本文对粒子算法的改进是一种尝试，仍有进一步研究的空间，如交叉

和变异对种群多样性影响的内在机制如何等，通过这些内在机制的研究，找出提升种群多样的策略，都是下一步需要进行研究的工作。

参 考 文 献

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

[2] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space [J]. IEEE Transactions on Evolutionary Computation, 2002, 2(6): 58-73

[3] Peram T, Veeramachaneni K, Mohan C K. Fitness-distance-ratio based particle swarm optimization[C]//Proceedings of the IEEE Swarm Intelligence Symposium. Indianapolis, Indiana, 2003: 174-181

[4] Mendes R, Kennedy J, Neves J. The fully informed particle swarm: Simpler, maybe better [J]. IEEE Transactions on Evolutionary Computation, 2004, 7(8): 204-210

[5] van den Bergh F, Engelbrecht A P. A cooperative approach to particle swarm optimization [J]. IEEE Transactions on Evolutionary Computation, 2004, 6(8): 225-239

[6] 刘衍民，赵庆祯. 一种基于动态邻居和变异因子的粒子群算法[J]. 控制与决策，2010，25(7)：968-974

[7] Stacey A, Jancic M, Grundy I. Particle swarm optimization with mutation[C]// Proceeding of IEEE Congress on Evolutionary Computation. Canberra, Australia, 2003: 1425-1430

[8] Deb K, Goyal M. A combined genetic adaptive search(GeneAS) for engineering design [J]. Computer Science and Informatics, 1996, 26(4): 30-45

[9] Martinez S Z, Coello C. Hybridizing an evolutionary algorithm with mathematical programming techniques for multi-objective optimization [C] // Proceedings of Genetic and Evolutionary Computation Conference. New York, USA, 2008: 769-770

[10] Shi X H, Liang Y C, Lee H P, et al. Particle swarm optimization-based algorithms for TSP and generalized TSP [J]. Information Processing Letters, 2007, 103(5): 169-176

[11] Deb K, Agrawal R B. Simulated binary crossover for continuous search space [J]. Complex Systems, 1995, 9(3): 115-148

(上接第 226 页)

$$\text{错误率}=\frac{N_e}{N}\times 100\% \tag{17}$$

$$\text{拒识率}=\frac{N_r}{N}\times 100\% \tag{18}$$

$$\text{可靠性}=\frac{N-N_e-N_r}{N-N_r}\times 100\% \tag{19}$$

式中，N_e 为错误识别的样本数，N_r 为拒识的样本数，N 为样本总数。从表 2 数据可以看出本文提出的基于 AP 的仿生模式识别方法对训练集进行筛选时，使用了原训练集中部分代表样本作为样本空间的特征点集，并以此构建特征子空间。识别过程依据子空间的相对划分完成，获得了较为理想的识别结果，识别结果明显优于其它分类器，证明本文方法能去除离群样本干扰，构建有效的代表样本集，同时能够保持字符模式的稳定性。

表 2 CENPARMI 库上的比较识别结果

样本数	分类器	错误率(%)	拒识率(%)	可靠性(%)
6000	K nearest neighbors	2.71	7.28	97.29
	Virtual SVM[14]	1.28	0	98.72
	Local Learning Framework[15]	1.86	0	98.14
	AP+BPR	0.86	1.45	99.14

结束语 本文提出了一种基于 AP 的仿生模式识别方法。从理论上讲，识别算法在训练集规模越大时越好，但同时所需的训练时间也越多，限制了其在计算、存储资源有限的特定系统中的应用。基于空间覆盖的识别方法，在小样本识别方面具有传统识别方法不可替代的优越性，而 AP 方法在样本筛选方面取得了较好的结果，类条件后验概率估计也在一定程度上提高了识别率。但在样本空间的构建和降低时间复杂度方面还需作进一步的研究。

参 考 文 献

[1] 王守觉. 仿生模式识别(拓扑模式识别)——一种模式识别新模型的理论与应用[J]. 电子学报，2002，30(10)：1417-1420

[2] 王守觉，徐健，王宪保，等. 基于仿生模式识别的多镜头人脸身份确认系统研究[J]. 电子学报，2003，31(1)：1-5

[3] 王守觉，曲延锋，李卫军，等. 基于仿生模式识别与传统模式识别的人脸识别效果比较研究[J]. 电子学报，2004，33(7)：1057-1061

[4] 高庆吉，王晓华，赵为平. 对粘连和缺损数字串分割的研究[J]. 模式识别与人工智能，2000，13(1)：99-102

[5] Teow L N, Loe K F. Robust vision-based features and classification schemes for off-line handwritten digit recognition[J]. Pattern Recognition, 2002, 35(3): 2355-2364

[6] 荆晓远，杨静宇. 基于相关性和有效互补性分析的多分类器组合方法[J]. 自动化学报，2000，26(6)：741-747

[7] Liu C L, Nakashima K, Sako H, et al. Handwritten digit recognition: benchmarking of state-of-the-art techniques [J]. Pattern Recognition, 2003, 36: 2271-2285

[8] Frey B J, Dueck D. Clustering by passing messages between data points[J]. Science, 2007, 315(5814): 972-976

[9] Smith S J, Bourgoin M O, Sims K, et al. Handwritten character classification using nearest neighbor in large databases[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994, 16(9): 915-919

[10] 娄震，金忠，杨静宇. 基于类条件置信变换的后验概率估计方法[J]. 计算机学报，2005，28(1)：18-24

[11] Kittle J, Hatef M, Duin R P W. On combining classifiers[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20(3): 226-239

[12] 王正群，叶晖，孙兴华，等. 基于模糊方向特征的手写体汉字识别[J]. 模式识别与人工智能，2001，14(3)：317-320

[13] Cheng L L, Nakashima K, Sako H, et al. Handwritten digit recognition: investigation of normalization and feature extraction techniques[J]. Pattern Recognition, 2004, 37(2): 265-279

[14] Scholkoph B, Burges C, Vapnik V. Incorporating invariances in support vector learning machines[C]// The International Conference on Artificial Neural Networks. Springer Lecture Notes in Computer Science, 1996, 1112: 47-52

[15] Dong J X, Krzyzak A, Suen C Y. Local learning framework for handwritten character recognition[J]. Engineering Applications of Artificial Intelligence, 2002, 15(2): 151-159

[16] Wierer J, Boston N. A handwritten digit recognition algorithm using two-dimensional Hidden Markov Models for feature extraction[C]// The 32nd IEEE International Conference on Acoustics, Speech and Signal Processing. Honolulu, Hawaii, USA, 2007: 505-508