31卷 第8期
2014年8月

微 电 子 学 与 计 算 机
MICROELECTRONICS & COMPUTER

Vol. 31 No. 8
August 2014

混沌差分进化粒子群协同优化算法

匡芳君[1,2]，张思扬[2]，金 忠[1]，徐蔚鸿[1,3]

（1 南京理工大学 计算机科学与工程学院，江苏 南京 210094；2 湖南安全技术职业学院 电气与信息工程系，湖南 长沙 410151；3 长沙理工大学 计算机与通信工程学院，湖南 长沙 410114）

摘　要：为有效地改善差分进化粒子群算法的性能，结合反向学习策略和信息交互机制，提出了一种新的混沌差分粒子群协同优化算法. 该算法采用反向学习策略产生初始种群，使得初始个体尽可能均匀分布，然后将初始种群随机等分为双种群，对双种群分别采用改进的混沌差分进化算法和混沌粒子群优化算法进行协同寻优，并在双种群中引入信息交互学习机制，在维持种群多样性的同时加快收敛速度. 通过对四个复杂高维的标准函数寻优测试，仿真结果表明，该算法能有效避免早熟收敛，收敛速度快，寻优精度较高，具有良好的全局搜索能力，鲁棒性好.

关键词：差分进化；粒子群优化；混沌搜索；协同优化；反向学习

中图分类号：TP18　　**文献标识码**：A　　**文章编号**：1000－7180(2014)08－0029－05

Chaotic Differential Evolution Particle Swarm Cooperative Optimization Algorithm

KUANG Fang-jun[1,2], ZHANG Si-yang[2], JIN Zhong[1], XU Wei-hong[1,3]

(1 School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China; 2 Department of Electronic and Information Engineering, Hunan Vocational Institute of Safety & Technology, Changsha 410151, China; 3 College of Computer and Communications Engineering, Changsha University of Science and Technology, Changsha 410114, China)

Abstract: To improve the performance of differential evolution particle swarm optimization, a novel chaotic differential evolution particle swarm cooperative optimization algorithm is proposed, which is combined the opposition-based learning and the interactive learning strategy. In this algorithm, an initialization strategy based on the opposition-based learning is applied to diversify the initial individuals in the search space. All individuals are randomly divided into two sub-swarm, one sub-swarm searches via improved chaotic differential evolution, and the other searches via improved chaotic particle swarm optimization at the same time. The interactive learning strategy is introduced in the bi-group to maintain the population diversity and accelerate the convergence speed. Experiments on four complex benchmark functions with high dimension, simulation results further demonstrate that, the algorithm not only effectively avoids the premature convergence, but also has rapid convergence speed, high solution precision, good searching ability and robustness.

Key words: differential evolution; particle swarm optimization; chaos search; cooperative optimization; opposition-based learning

1 引言

差分进化(Differential Evolution, DE)是1995年由Storn和Price[1]首次提出，是一种基于种群差异并行搜索进化算法，原理简单，容易实现，搜索空间复杂度低，能协同搜索，鲁棒性强，但存在算法后

收稿日期：2013－09－27；**修回日期**：2013－11－13

基金项目：国家自然科学基金资助项目(61373063)；湖南省科技计划项目(2012SK4046，2013FJ4217)；湖南省教育厅资助科研项目(13C086)

期收敛速度较慢,表现不够稳定等缺陷[2],很多学者对其进行了广泛研究,如 Feoktistov 等[3]提出了一种广义的变异策略框架,便于选择变异操作类型;COELHO[4]等提出了一种混沌差分进化算法,利用混沌序列产生初始群;Lu 等[5]提出了一种自适应差分进化算法,引入了自适应交叉因子和变异因子.

粒子群优化算法(Particle Swarm Optimization,PSO)是一种随机搜索的群智能优化算法,容易实现,且收敛速度快[6].但粒子群算法虽简单,却易陷入局部最优,求解精度较差,为提高算法的性能,很多学者提出了各种改进算法,栾丽君等[7]和阳春华等[8]提出差分进化粒子群混合优化算法.

为有效改善差分进化粒子群优化算法的寻优效率和全局搜索能力,本文在文献[7-8]的基础上,将反向学习策略、混沌思想、信息交互机制引入到差分进化粒子群优化算法中,通过融合混沌差分进化算法(Chaotic Differential Evolution,CDE)和改进的混沌粒子群优化算法(Improved Chaotic Particle Swarm Optimization,ICPSO),提出了混沌差分进化粒子群协同优化算法(简称 CDE-CPSO).

2 方法描述

2.1 混沌模型

混沌现象存在于非线性动力系统中,具有初值敏感性、规律性、普适性、遍历性等优点.较典型混沌模型是 Logistic 映射[9],表达式为

$$y_{t+1}=\mu y_t(1-y_t) \tag{1}$$

式中,t 为迭代次数,$t=1,2,\cdots,T$,T 为最大迭代次数;$y_t\in(0,1)$ 是随机数,且 $y_t\notin\{0.25,0.5,0.75\}$;$\mu$ 是控制参数,一般取值为 4,此时系统处于完全混沌状态.

2.2 混沌差分进化算法

2.2.1 差分进化算法

差分进化算法(DE)是先运用当前种群个体的差分变异和交叉重组得到中间种群,然后基于贪婪选择策略对父代种群和中间种群中的个体进行选择操作得新一代种群[1-5],它与标准遗传算法一样包含选择、交叉和变异三个操作,与遗传算法的区别,DE 采用由变异到交叉,再到选择的操作顺序[10].

(1) 变异操作

变异操作是从种群中随机选取个体 $X_i(t)=x_{i1}(t),x_{i2}(t),\cdots,x_{ij}(t)$,本文采用 DE/rand/1/bin 策略[11],即根据式(2)产生变异中间个体 $V_i(t+1)$.

$$V_i(t+1)=X_{r_1}(t)+G\cdot(X_{r_2}(t)-X_{r_3}(t)) \tag{2}$$

式中,$i\in\{1,2,\cdots,M\}$ 是种群数,$j\in\{1,2,\cdots,d\}$,d 为解空间的维数,t 为当前代数,r_1,r_2,r_3 为随机正整数,且满足 $r_1\neq r_2\neq r_3\neq i$,$X_{r_1},X_{r_2},X_{r_3}$ 是从进化群体中随机选取的个体,缩放因子 $G\in[0,1]$ 用来控制差分向量 $X_{r_2}(t)-X_{r_3}(t)$ 的幅度和影响程度.

(2) 交叉操作

按式(3)进行交叉操作,利用变异个体 $V_i(t+1)$ 和目标个体 $X_i(t)$ 产生试验个体 $u_{i,j}(t+1)$.

$$u_{ij}(t+1)=\begin{cases}v_{ij}(t+1),\text{if}((\text{rand}<P_c)\text{or}(j=j_{\text{rand}}))\\ x_{ij}(t),\text{else}\end{cases} \tag{3}$$

式中,rand 是(0,1)之间均匀分布的随机数,$j\in\{1,2,\cdots,d\}$,d 为解空间的维数,$j_{\text{rand}}\in\{1,2,\cdots,d\}$ 间随机整数,P_c 是交叉因子.

(3) 选择操作

选择操作是在目标个体 $X_i(t)$ 与试验个体 $U_i(t+1)$ 进行,利用适应度函数来决定是否在下一代中使用试验个体替换目标个体,其选择原则为适应度较优的个体进入下一代,检验公式如下:

$$X_i(t+1)=\begin{cases}U_i(t+1),\text{if}(f(U_i(t+1))<f(X_i(t)))\\ X_i(t),\text{else}\end{cases} \tag{4}$$

式中,$X_i(t+1)$ 为下一代第 i 个个体,$f(x)$ 表示适应度函数.

2.2.2 混沌差分进化算法

为进一步加快差分进化算法的全局收敛速度,引入混沌随机搜索,在每代个体和当前最佳个体 X_{best} 之间的解空间内进行混沌搜索,提出新的混沌差分进化(Chaotic Differential Evolution,CDE)算法.CDE 的算法步骤如下:

步骤 1 初始化种群.包括种群规模 N_p,变异因子 G、交叉因子 P_c、种群最大迭代次数 T,第 j 维输入变量的上界 $X_{\max,j}$ 和下界 $X_{\min,j}$、混沌搜索次数 k.并按式(5)随机生成初始种群.

$$x_{ij}(0)=x_{\max j}(0)+rand(1)\cdot(x_{\max j}-x_{\min j}) \tag{5}$$

式中,$i\in\{1,2,\cdots,N_p\}$,$j\in\{1,2,\cdots,d\}$,d 为解空间的维数,$rand(1)$ 为(0,1)之间均匀分布的随机数.

步骤 2 计算初始种群的适应度值.找出适应

度值最好的个体 X_{best}，记录其适应度值 $F_{\mathrm{best}}=f(X_{\mathrm{best}})$ 及在种群的索引 $index(X_{\mathrm{best}})$.

步骤 3　变异操作. 按式(2)产生变异个体 $V_i(t+1)$. 由于缩放因子 G 的取值在很大程度上影响群体的多样性和算法的收敛速度，因此本文采取自适应缩放因子策略，即当个体适应度值趋于一致或收敛于局部最优时，缩放因子 G 适当增加，从而保持种群的多样性；当种群适应度值差异较大时，缩放因子 G 适当减小，进而提升算法的收敛速度. 自适应缩放因子的表达式定义如下：

$$G_i=\begin{cases}1,\mathrm{if}(fit_{\mathrm{worst}}=fit_{\mathrm{best}})\\ 1-\dfrac{fit_{\mathrm{worst}}-fit_i}{fit_{\mathrm{worst}}-fit_{\mathrm{best}}},\mathrm{else}\end{cases} \tag{6}$$

式中，fit_i 为第 i 个个体的适应度值，fit_{worst} 为种群中最差个体的适应度值，fit_{best} 为种群中最优个体的适应度值.

步骤 4　交叉操作. 按式(3)生成交叉试验个体 $U_i(t+1)$. 本文采用文献[5]提出的自适应交叉因子策略，将交叉因子 P_c 与最大迭代次数 T 以及当代代数 t 关联，以增强种群的多样性. 自适应交叉因子 P_c 的表达式定义如下：

$$P_c=P_d\times 2^{e^{(1-\frac{T}{t+1})}} \tag{7}$$

式中，P_d 是初始化交叉因子，取值满足[0, l]间的概率约束.

步骤 5　选择操作. 按式(4)得到下一代个体 $X_i(t+1)$.

步骤 6　比较适应度值 $f(U_i(t+1))$ 和 F_{best}. 如果 $f(U_i(t+1))<F_{\mathrm{best}}$，则 $X_{\mathrm{best}}=U_i(t+1)$，$F_{\mathrm{best}}=f(U_i(t+1))$ $index(X_{\mathrm{best}})=i$.

步骤 7　重复执行 N_p 次步骤 3 ~ 步骤 6.

步骤 8　混沌搜索. 为加快 DE 全局收敛速度，在个体 X_{best} 附近进行混沌搜索，即：

$$X_k=X_{\mathrm{best}}+\alpha y_k \tag{8}$$

式中，y_k 是式(1)的解；α 是调节因子，由式(1)可知，当初值 $y_1\in(0,1)$ 时，式(1)的解全为正解. 引入调节因子可使 X_k 向正反两个方向搜索，其取值如下：

$$\alpha=\begin{cases}1 & \mathrm{if}\quad r\geqslant 0.5\\ -1,\mathrm{else}\end{cases} \tag{9}$$

式中，r 为(0,1)中一个均匀分布的随机数.

步骤 8.1　随机产生(0,1)中一个均匀分布的随机数 y_1，且 $y_1\neq\{0.25,0.5,0.75\}$.

步骤 8.2　由式(1)、(8)、(9)计算得到 X_k.

步骤 8.3　重复执行 k 次步骤 8.2.

步骤 9　找出 X_k 中适应度最好的个体 $X_{k,\mathrm{best}}$，如果 $f(X_{k,\mathrm{best}})<F_{\mathrm{best}}$，则 $X_{\mathrm{best}}=X_{k,\mathrm{best}}$，$F_{\mathrm{best}}=f(X_{k,\mathrm{best}})$，产生随机数 $j\in\{1,2,\cdots,N_p\}$，$index(X_{\mathrm{best}})=j$，$X_j(t+1)=X_{k,\mathrm{best}}$.

步骤 10　判断算法是否达到最大迭代次数或满足求解精度要求，若满足则输出全局最优位置 $X_{k,\mathrm{best}}$ 和其适应度值 F_{best}，否则返回步骤 3.

2.3　混沌粒子群优化算法

2.3.1　粒子群优化算法

粒子群优化算法(PSO)是采用速度——位置搜索模型的一种群智能优化算法[6]，通过个体间的协作来寻找最优解，可解决一些难的优化问题. 每个粒子根据自身飞行经验和群体飞行经验来动态调整.

假设在 d 维目标解空间中，$S=\{P_1,P_2,\cdots,P_m\}$ 表示 m 个潜在解组成的种群；粒子 i 的位置记作 $P_i^d(t)(i=1,2,\cdots,m)$；粒子 i 的飞行速度记作 $V_i^d(t)(i=1,2,\cdots,S)$；个体极值 $P_{\mathrm{best}}^d(t)$ 表示粒子 i 自身找到的最好位置；全局极值 $G_{\mathrm{best}}^d(t)$ 表示整个种群目前找到的最好位置. 粒子 i 的速度和位置更新方程定义如下：

$$\begin{cases}V_i^d(t+1)=W(t)\cdot V_i^d(t)+C_1(t)\cdot R_1(t)\cdot\\ (P_{\mathrm{best}}^d(t)-P_i^d(t))+C_2(t)\cdot R_2(t)\cdot(G_{\mathrm{best}}^d(t)-P_i^d(t))\\ P_i^d(t+1)=P_i^d(t)+V_i^d(t+1)\end{cases} \tag{10}$$

式中，$V_i^d(t)$、$V_i^d(t+1)$、$P_i^d(t)$、$P_i^d(t+1)$ 分别是粒子 i 当前时刻、下一时刻的速度及所在位置；C_1 和 C_2 是学习因子，一般取值在 1.5 和 2.0 间；R_1 和 R_2 是(0,1)间的随机数；W 是惯性权重，用来平衡全局搜索和局部搜索能力.

2.3.2　改进的混沌粒子群优化算法

在 PSO 算法中，参数 C_1、C_2、R_1、R_2 和 W 都是影响 PSO 算法收敛的重要因素. 标准 PSO 收敛速度较快，但在算法后期速度缓慢，粒子群表现趋同性，易陷入局部极小，即早熟现象. 针对这些缺陷，引入混沌搜索思想，构造改进的 CPSO 算法(ICPSO)，从而有效提高算法的收敛速度，同时避免算法后期振荡.

本文提出的 ICPSO 算法基本思想主要体现在以下五个方面：

(1) 惯性因子 W 对 ICPSO 的收敛性起着非常重要的作用，为了有效平衡局部和全局搜索能力，提升算法寻优性能，加快收敛速度，将混沌引入到惯性因子 W 的优化，并映射到[$W_{\min}$, $W_{\max}$]区间中，调整公式如下：

$$W(t+1)=4.0W(t)(1-W(t)) \tag{11}$$

$$W(t)=W_{\min}+(W_{\max}-W_{\min})W(t) \tag{12}$$

式中，$[W_{\min}, W_{\max}]$ 是惯性因子的取值范围，一般取 $[0.4, 0.9]$.

(2) 为克服随机取值带来效率不高，将混沌引入到随机数 R_1 和 R_2 中，更新公式为

$$R_i(t+1) = 4.0R_i(t)(1-R_i(t)) \tag{13}$$

式中，$R_i(t) \in (0,1), i = 1,2$.

(3) 学习因子 C_1 和 C_2 代表将每个粒子拉向最优位置的随机加速项的权重，为了有效地提高算法的收敛速度，将混沌引入到学习因子 C_1 和 C_2 中，更新公式如下：

$$C_i(t+1) = 4.0C_i(t)(1-C_i(t)) \tag{14}$$

$$C_i(t) = C_{\min} + (C_{\max} - C_{\min})C_i(t) \tag{15}$$

式中，$i = 1,2$，$[C_{\min}, C_{\max}]$ 是学习因子的取值范围，一般取 $[1.5, 2.0]$.

(4) 利用混沌序列产生初始粒子的位置和速度，不仅不改变 PSO 初始化具有的随机性，而且能利用混沌序列提高种群的多样性和搜索遍历性.

(5) 将混沌引入到所有粒子群至今为止搜索到的最优位置，并将此混沌序列中最优位置作为粒子更新的位置，能很好地防止粒子位置趋同，并使惰性粒子快速跳出局部最优值.

ICPSO 算法的优化步骤如下：

步骤 1 初始化 ICPSO 算法的控制参数，粒子群体规模、最大迭代次数 T，惯性因子的范围 $[W_{\min}, W_{\max}]$，学习因子的范围 $[C_{\min}, C_{\max}]$，调用式(11) 至式(15) 产生一个 W、C_1、C_2、R_1 和 R_2 的混沌序列.

步骤 2 利用混沌序列产生初始粒子的位置和速度.

步骤 2.1 随机产生 d 维每个分量在(0,1) 之间的向量 $Z_i^d(t)$，d 为目标函数中的变量个数，代入式(1)Logistic 映射中，则可得到 N 个不同轨迹的混沌序列 $Z_i^d(t)$，作为初始群.

步骤 2.2 将 $Z_i^d(t)$ 各个分量通过 $P_i^d(t) = P_{\min}^d + (P_{\max}^d - P_{\min}^d)z_i^d(t)$ 逆映射到对应变量的取值范围内.

步骤 2.3 计算粒子适应度值，从 N 个初始种群选取性能较好的 M 个位置作为初始位置，并随机产生 M 个初始速度.

步骤 3 比较粒子的适应度值 $F(P_i^d)$ 和 $F(P_{\text{best}}^d)$，如果 $F(P_i^d) < F(P_{\text{best}}^d)$，则更新 P_{best}^d.

步骤 4 比较更新后的 $F(P_{\text{best}}^d)$ 和 $F(G_{\text{best}}^d)$，如果 $F(P_{\text{best}}^d) < F(G_{\text{best}}^d)$，则更新 G_{best}^d.

步骤 5 根据式(10) 更新粒子的速度和位置(每一代的 W、C_1、C_2、R_1 和 R_2 依次取步骤 1 中产生的混沌序列).

步骤 6 对最优位置 G_{best}^d 进行混沌优化.

步骤 6.1 将 G_{best}^d 映射到(0,1)，即 $z_i^d(t) = (G_{\text{best}}^d - P_{\min}^d)/(P_{\max}^d - P_{\min}^d)$，其中 $[P_{\min}^d, P_{\max}^d]$ 为各参数的取值范围.

步骤 6.2 将上式代入式(1)Logistic 映射中进行迭代产生混沌变量.

步骤 6.3 通过 $G_{\text{best}}^d(t) = P_{\min}^d + (P_{\max}^d - P_{\min}^d)z_i^d(t)$ 逆映射将混沌变量载波到原解空间.

步骤 6.4 计算在原解空间的混沌变量经历过的每一个可行解 G_{best}^d 的适应度值，获取性能最优的可行解 G_{best}^*.

步骤 7 利用 G_{best}^* 替代当前种群中任一粒子的位置.

步骤 8 判断算法是否达到最大迭代次数或满足求解精度要求，若满足则输出全局最优位置和其适应度值，否则返回步骤 3.

3 混沌差分进化粒子群协同优化算法

3.1 基于反向学习策略的种群初始化

为保持种群的多样性和使初始种群的个体尽可能均匀分布，本文利用基于反向学习策略产生初始种群[12]. 即首先利用混沌序列产生初始解，然后求出每个初始解所对应的反向解，最后对产生的所有解排序，将适应度值较优的解作为初始种群的解，这有助于提高解的质量和求解效率.

3.2 信息交互学习机制

为加快收敛速度，本文引入信息交互机制，将种群随机分为大小相同的子种群 D 群和 P 群，D 群用 CDE 算法进行进化，P 群用 ICPSO 算法进行进化，每隔一定进化代数就利用信息交互机制让两个子群进行交互学习，通过设置群体交互学习参数 T_G 控制群体间的交互学习，即每进化 T_G 代后，就比较 D 群和 P 群的最优个体适应度值，并用较好的那组群体的最优个体替代较差的那组群体的最差个体. 从而均衡全局搜索能力和收敛速度之间的不平衡，进而既能维持种群多样性，又能加快收敛速度，最终使算法收敛和全局最优的性能得到提升. 但是参数 T_G 的取值太小，两个子群个体会逐渐趋同，容易陷入早熟；如果取值太大，则不利两个子群的交互学习. 通过对测试函数的反复实验得出，T_G 的取值范围在 $[40, 60]$ 之间较好.

3.3 CDE-CPSO 算法描述

算法流程如下：

步骤 1　初始化 CDE-CPSO 算法的基本参数. 包括群体规模 N、最大迭代次数 T、求解精度 ε、交互学习参数 T_G 等算法参数.

步骤 2　利用反向学习策略产生初始种群的解 $X_i(i=1,2,\cdots,N)$，计算其适应度值，并将种群等分为双种群 D 群和 P 群.

步骤 3　根据 CDE 算法对 D 群中所有个体执行变异、交叉、选择等操作.

步骤 4　根据 ICPSO 算法对 P 群中所有个体进行速度、位置更新等操作.

步骤 5　找出 D 群中的最佳个体 $G_{\text{best}}^{\text{CDE}}$ 及相应适应度值 $F_{\text{best}}^{\text{CDE}}$.

步骤 6　找出 P 群中的最佳个体 $G_{\text{best}}^{\text{CPSO}}$ 及相应适应度值 $F_{\text{best}}^{\text{CPSO}}$.

步骤 7　比较 $G_{\text{best}}^{\text{CDE}}$ 和 $G_{\text{best}}^{\text{CPSO}}$ 优劣.

步骤 8　判断是否满足交互学习条件，如满足则进行交互操作，否则转到步骤 10.

步骤 9　选择较好的那组群体的最优个体替代较差的那组群体的最差个体.

步骤 10　记录迄今为止最好的解.

步骤 11　判断算法是否达到最大迭代次数或满足求解精度要求，如满足则输出最优位置和其适应度值，否则转到步骤 3.

4　仿真结果与分析

为了验证提出的 CDE-CPSO 算法寻优性能及收敛速度，选取四个 Benchmark 标准函数进行寻优测试. 四个标准函数具有不同的特点，可以充分考察算法的优化能力.

为衡量本文提出算法的优化性能，与其他搜索算法比较和增加寻优的复杂性，分别采用标准粒子群（Standard PSO，SPSO）、CPSO、DE、CDE、PSODE[12] 和混沌差分进化粒子群协同优化算法（CDE-CPSO）进行寻优测试. 实验结果表明：

（1）由表 2 和图 1～4 可以看出，所有算法对大部分函数的寻优精度都能达到，说明其都有较强的全局搜索能力，且 CDE-CPSO 比其他算法在计算精度和收敛速度方面有明显提高，说明该算法能有效地跳出局部最优，不仅提高了算法的收敛速度，而且具有较强的全局搜索能力.

（2）对于单峰函数 Sphere，从图 1 可以看出，基本上所有算法都能很快达到求解精度，但 CDE-CPSO 有更好的寻优效果. 对于非凸、病态单峰函数 Rosenbrock 而言，由于在其取值范围内走势平坦，要收敛到全局最优点的机会很少，但 CDE-CPSO 算法很快就收敛到求解精度.

（3）多峰函数 Rastrigin 和 Griewank 是复杂的非线性全局优化问题，CDE-CPSO 的求解精度和收敛速度均高于其他算法，主要是 CDE-CPSO 中的参数是动态调整，使其尽可能跳出局部最优，提高收敛精度和速度.

（4）CDE-CPSO 对有局部极小值或易陷入局部最优的非线性函数也具有较高的寻优精度和各类测试函数都具有较好的稳定性和鲁棒性，主要是因为它采用了双种群交互学习协同进化策略，即使一个种群中个体陷入局部极值，另一个种群也会继续进行寻优，确保整个群体的全局优化能力.

5　结束语

提出一种新的混沌差分进化粒子群协同优化算法（CDE-CPSO）. 通过对四个标准函数的寻优测试，仿真结果表明，该算法引入混沌序列和反向学习策略产生初始种群，和双种群协同进化，既能保持种群的多样性，又能有效避免早熟收敛和提高搜索的遍历性. 对于高维、多极值点的非线性函数，具有较好的全局搜索能力和寻优精度，且算法稳定性和鲁棒性较好. 下一步工作将考虑利用该改进算法应用到解决实际优化问题，并结合其他智能优化算法，提出性能更好的全局优化算法.

参考文献：

[1]　Storn R，Price K. Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous space[R]. Berkley：International Computer Science Institute，1995.

[2]　孟红云，张小华，刘三阳. 用于约束多目标优化问题的双群体差分进化算法[J]. 计算机学报，2008，31(2)：228-235.

[3]　Feoktistov V，Janaqi S. Generalization of the strategies in differential evolution[C]// Proc of the 18th Int Parallel and Distributed Processing Symposium. Santa Fe，2004：165-170.

[4]　Coelho L D S. Reliability redundancy optimization by means of a chaotic differential evolution approach [J]. Chaos，Solitons & Fractals，2008，37(6)：1607-1615.

[5]　Lu Y L，Zhou J Z，Qin H，et al. An adaptive chaotic differential evolution for the short-term hydrothermal generation scheduling problem[J]. Energy Conversion and Management，2010(51)：1481-1490.

（下转第 39 页）

的，不能很好的适用于任意车厢图像的车号定位，因而具有一定的局限性.

图 6 显示的是四组利用本文方法能够准确定位而文献[2-3]定位错误的图像，每组图像由 4 幅图组成，依次是：原图、车号显著图、车号定位图以及车号图像. 由图 6 可以看出，本文算法能够较好地解决反光、强光、光照不均、遮挡以及车号边缘模糊等情况下的车号定位.

4　结束语

本文提出了一种基于视觉注意力机制的列车车号定位的新方法. 该方法利用眼动仪获取真实眼动数据，提取车厢图像的显著特征，进而建立视觉注意力模型，利用该模型预测车号显著区域，并分析定位车号区域. 实验证明本文方法具有较好的鲁棒性和普适性，更能适应不同光照下各类车厢的车号定位. 然而，本文方法过程较为复杂、耗时较长.

参考文献：

[1] 耿青云，孙泳江. TMIS 中央系统整体架构概述[J]. 铁路计算机应用，2005，14(7)：39-41.

[2] 杨绍华，李建勇，王恒，等. 基于 BP 神经网络的货车车号识别方法研究[J]. 计算机应用，2007，16 (12)：4-7.

[3] 赵入宾. 铁路货车车号识别的算法研究 [D]. 河北：河北工业大学，2010.

[4] 王少杰，朱志刚，石定机，等. 货运列车车型车号自动分割和识别算法[J]. 模式识别与人工智能，1998，11 (3)：328-334.

[5] 沈兰荪，张菁，李晓光. 图像检索与压缩域处理技术的研究[M]. 北京：人民邮电出版社，2008.

[6] Zhai Y，Shah M. Visual attention detection in video sequences using spatiotemporal cues[C]// ACM Multimedia. Santa Barbara，2006：815-824.

作者简介：

葛　莉　女，(1989-)，硕士研究生. 研究方向为图像处理、视线跟踪.

(上接第 33 页)

[6] 刘道华，原思聪，兰洋，等. 混沌映射的粒子群优化方法[J]. 西安电子科技大学学报，2010，37(4)：764-768.

[7] 栾丽君，谭立静，牛奔. 一种基于粒子群优化算法和差分进化算法的新型混合全局优化算法[J]. 信息与控制，2007，36(6)：708-714.

[8] 阳春华，钱晓山，桂卫华. 一种混沌差分进化和粒子群优化混合算法[J]. 计算机应用研究，2011，28(2)：439-441.

[9] Ou C M. Design of block ciphers by simple chaotic functions [J]. IEEE Computational Intelligence Magazine，2008，3(2)：54-59.

[10] 杨妍，陈如清，俞金寿. 差分进化粒子群混合优化算法的研究与应用[J]. 计算机工程与应用，2010，46(25)：238-241.

[11] Storm R. On the usage of deferential evolution for function optimization[C]// Proc of Biennial Conference of the North American Fuzzy Information Processing Society. Piscataway：IEEE Press，1996：519-523.

[12] 吴昱，李元香，徐星. 基于群智能的新型反向混合差分进化算法[J]. 小型微型计算机系统，2009，5(4)：903-907.

作者简介：

匡芳君　女，(1976-)，博士，副教授. 研究方向为模式识别、智能信息处理技术及应用、信息安全等.

张思扬　男，(1973-)，硕士，讲师. 研究方向为人工智能、故障诊断、电子技术等.

金　忠　男，(1961-)，博士，教授，博士生导师. 研究方向为模式识别、机器学习、计算机视觉等.

徐蔚鸿　男，(1963-)，博士，教授，博士生导师. 研究方向为人工智能、模式识别、软件工程等.