第 30 卷第 7 期
2013 年 7 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 30 No. 7
Jul. 2013

DOI: 10.7641/CTA.2013.20705

交替捕食的粒子群优化算法及其粒子轨迹收敛性分析

李军军[1,2†], 黄有方[1], 杨 斌[1]

(1. 上海海事大学 科学研究院, 上海 201306; 2. 上海海事大学 商船学院, 上海 201306)

摘要: 为避免算法陷入局部极值, 在捕食者–猎物协同进化机制基础上, 提出了一种交替捕食的粒子群优化算法(APPSO). 对该算法迭代过程进行了分析, 给出并证明了粒子运动轨迹收敛的充分条件. 为使粒子运动轨迹可靠收敛, 构建了一种参数设置方法. 通过迭代矩阵谱半径计算、SQRT序列采样, 对该算法的粒子轨迹收敛速度进行了分析. 基准测试函数仿真结果表明, 交替捕食的PSO算法具有较佳的搜索性能.

关键词: 粒子群优化; 交替; 捕食者; 猎物; 收敛性分析

中图分类号: TP301.6　　**文献标识码**: A

Alternately preying particle swarm optimization algorithm and convergence analysis of its particle trajectories

LI Jun-jun[1,2†], HUANG You-fang[1], YANG Bin[1]

(1. Scientific Research Academy, Shanghai Maritime University, Shanghai 201306, China;
2. Merchant Marine College, Shanghai Maritime University, Shanghai 201306, China)

Abstract: To avoid getting into local extremum, we put forward an alternately preying particle swarm optimization algorithm (APPSO) on the basis of predator-prey coevolution. The iteration process of APPSO is analyzed. The sufficient condition for the convergence of particle trajectories is proposed and proved. A parameter setting method is developed to make the particles motion trajectories reliably convergent. The convergence rate of motion trajectories in APPSO is analyzed based on the iteration matrix spectral radius and SQRT sequence. Simulation results of benchmark functions validate the correctness and efficiency of the proposed method.

Key words: particle swarm optimization; alternately; predator; prey; convergence analysis

1 引言(Introduction)

粒子群优化(particle swarm optimizer, PSO)算法由Kennedy和Eberhart于1995年提出[1], 是一种新的进化计算技术. PSO算法简单容易实现同时又有深刻的智能背景, 自提出以来, 在国内外得到相关领域众多学者的广泛关注和研究[2–6].

刘微、陈贺新等[7–8]借鉴协同进化理论中的生态捕食机制, 在基本PSO中引入了捕食者(predator)与猎物(prey)两种生态角色, 提出了PSOPC(particle swarm optimizer based on predator-prey coevolution)的算法. PSOPC具有较好的仿生学意义, 概念简单, 参数较少, 实现容易, 是一种较好的PSO改进方法. 但一部分粒子总在捕食, 另一部分粒子总在逃逸, 这种机制也使得算法存在易于陷入局部极值的缺点.

本文设计了一种交替捕食策略, 在算法迭代过程中, 让捕食逃逸学习因子的正负号发生变化. 即算法的子种群时而作为捕食者, 时而作为逃逸者, 这样可以加大对解空间的搜索, 避免算法陷入局部极值. 从迭代矩阵谱半径的角度, 对该算法的迭代过程进行了详细的分析, 提出并证明了粒子运动轨迹收敛的充分条件. 构建了一种保证算法粒子运动轨迹收敛的参数设置方法, 对粒子运动轨迹收敛速度进行了分析. 通过基准测试函数仿真实验, 验证了该算法的搜索性能.

2 PSOPC算法(PSOPC algorithm)

在PSOPC算法中粒子分成两类: predator粒子与prey粒子. 在算法的迭代过程中, predator粒子的状态更新方程为(1)(2), prey粒子的状态更新方程为(3)(4).

$$V_{id}^{\phi} = \omega V_{id}^{\phi} + c_1 R_1 (p_{id}^{\phi} - X_{id}^{\phi}) + c_2 R_2 (p_{g}^{\phi} - X_{id}^{\phi}) + c_3 R_3 (p_{g}^{\tau} - X_{id}^{\phi}), \quad (1)$$

$$X_{id}^{\phi} = X_{id}^{\phi} + V_{id}^{\phi}, \quad (2)$$

$$V_{id}^{\tau} = \omega V_{id}^{\tau} + c_1 R_1 (p_{id}^{\tau} - X_{id}^{\tau}) + c_2 R_2 (p_{g}^{\tau} -$$

收稿日期: 2012−06−22; 收修改稿日期: 2013−02−22.
†通信作者. E-mail: jslilj@163.com; Tel.: +86 021−38282966.
基金项目: 国家自然科学基金资助项目(51279099); 上海市科学技术委员会基金资助项目(12ZR1412500); 上海市教委科研创新基金资助重点项目(13ZZ124); 上海市教育委员会和上海市教育发展基金会“曙光计划”基金资助项目(12SG40).

$$X_{id}^{\tau}) - c_3R_3(p_g^{\phi} - X_{id}^{\tau}), \tag{3}$$

$$X_{id}^{\tau} = X_{id}^{\tau} + V_{id}^{\tau}, \tag{4}$$

其中: ϕ为predator种群标号, τ为prey种群标号, V_{id}^{ϕ}和V_{id}^{τ}为 predator 粒子、prey 粒子的速度, X_{id}^{ϕ} 和 X_{id}^{τ} 为 predator 粒子、prey 粒子的位置. ω 为惯性权重, $c_1, c_2, c_3 \geqslant 0$为学习因子, 本文将$c_3$称为捕食逃逸学习因子; R_1, R_2, R_3 为 $(0,1)$ 之间的随机数; p_{id}^{ϕ} 和 p_{id}^{τ} 分别为predator粒子、prey粒子的个体最优位置; p_g^{ϕ}和p_g^{τ}分别为predator种群、prey种群中的最好位置. 式(1)(3)等号右边的第4部分分别表示predator粒子对prey粒子的捕食搜索、prey粒子对predator粒子捕食压力的逃逸行为. predator粒子和prey粒子的种群规模分别为S^{ϕ}, S^{τ}.

3 交替捕食的PSO算法(Alternately preying particle swarm optimization algorithm)

3.1 交替捕食策略(Alternately preying strategy)

PSOPC算法中prey种群总是处于逃逸状态, predator种群总是处于追捕状态. 若prey种群陷入了局部极值, 则predator种群也容易由于一味的追捕prey种群而同样陷入局部极值.

对式(1)(3)进行对比可知, 若c_3取负, 两种群捕食、逃逸关系会发生对调. 因此, 为避免算法陷入局部极值, 本文根据p_g^{ϕ}和p_g^{τ}的优劣对c_3的正负号进行设置. 则某一种群不固定为捕食者或猎物. c_3的正负号按如下规则设置:

if (p_g^{ϕ}优于p_g^{τ}),
　if (rand $< p$), $c_3 < 0$,
　else, $c_3 > 0$,
if (p_g^{τ}优于p_g^{ϕ}),
　if (rand $< p$), $c_3 > 0$,
　else, $c_3 < 0$,

其中: rand为$(0,1)$之间的随机数, $0.5 < p < 1$. 即以较大概率取较优(全局极值较优)种群作为猎物, 另一种群作为捕食者. 这样整个种群更趋向于已知最有解, 利于加快算法搜索速度. 同时以较小概率取较差(全局极值较差)种群作为猎物, 另一种群作为捕食者. 这样也利于算法跳出局部极值.

另外, 当$p_g^{\tau} = p_g^{\phi}$ 时, 式(1)(3)等号右边的第4部分可以合并到第3部分, 算法与普通PSO算法一样, 没有捕食、逃逸行为. 相对而言, 此时算法容易陷入局部极值, 或可能已经陷入局部极值. 因此本文对算法进行如下调整:

If ($p_g^{\tau} = p_g^{\phi}$), $c_3 < 0$, $p_g^{\phi} = p_g^{\phi}[1 - A_N + 2 * A_N * \text{rand}]$,

其中A_N是扰动幅值, 一般取一个相对较小的正实数, 以使p_g^{ϕ}在其邻域内变动. 这样使得两种群恢复捕食、逃逸行为, 利于算法跳出局部极值.

3.2 粒子轨迹收敛性分析(Convergence analysis of particle trajectories)

目前, 不少文献对遗传算法、蚁群算法等智能优化算法采用Markov链、Lyapunov稳定性理论等方法分析算法的收敛性能, 但这些方法在分析PSO算法时较为复杂. 而迭代矩阵谱半径方法[3]对于PSO这类能写出迭代矩阵的算法, 可以方便地分析计算出粒子轨迹稳定收敛的参数范围; 还可以通过对谱半径的计算, 比较各种参数设置下的粒子轨迹收敛速度, 构建参数设置方法. 相对于其他方法, 该方法概念清晰, 方便使用. 因此这里采用迭代矩阵谱半径方法对交替捕食粒子群优化算法进行分析研究.

文献[3]证明了对于标准PSO, 如下推论1成立.

推论 1　标准PSO算法对任给初始微粒群收敛的一个充分条件是$\rho(A) < 1$, 即式(5)

$$\begin{cases} -1 < \omega < 1, \\ 0 < \varphi < 2 + 2\omega, \end{cases} \tag{5}$$

且当此条件满足时, $\rho(A)$越小, 收敛速度越快. 其中

$$\varphi = c_1R_1 + c_2R_2,\ A = \begin{bmatrix} \omega & -\varphi \\ \omega & 1-\varphi \end{bmatrix}.$$

文献[2]也得到了类似的结论.

令$C_1 = c_1R_1 \geqslant 0, C_2 = c_2R_2 \geqslant 0, C_3 = c_3R_3$, $\varphi = C_1 + C_2 \geqslant 0$, 整理式(1)–(4), 可得

$$V_{id}^{\phi} = \omega V_{id}^{\phi} - (\varphi + C_3)X_{id}^{\phi} + C_1p_{id}^{\phi} + C_2p_g^{\phi} + C_3p_g^{\tau}, \tag{6}$$

$$X_{id}^{\phi} = \omega V_{id}^{\phi} + (1 - \varphi - C_3)X_{id}^{\phi} + C_1p_{id}^{\phi} + C_2p_g^{\phi} + C_3p_g^{\tau}, \tag{7}$$

$$V_{id}^{\tau} = \omega V_{id}^{\tau} - (\varphi - C_3)X_{id}^{\tau} + C_1p_{id}^{\tau} + C_2p_g^{\tau} - C_3p_g^{\phi}, \tag{8}$$

$$X_{id}^{\tau} = \omega V_{id}^{\tau} + (1 - \varphi + C_3)X_{id}^{\tau} + C_1p_{id}^{\tau} + C_2p_g^{\tau} - C_3p_g^{\phi}. \tag{9}$$

式(6)–(9)可以表示为这样的矩阵形式:

$$\begin{bmatrix} V_{id}^{\phi} \\ X_{id}^{\phi} \\ V_{id}^{\tau} \\ X_{id}^{\tau} \end{bmatrix} = \begin{bmatrix} \omega & -(\varphi + C_3) & 0 & 0 \\ \omega & 1-(\varphi + C_3) & 0 & 0 \\ 0 & 0 & \omega & -(\varphi - C_3) \\ 0 & 0 & \omega & 1-(\varphi - C_3) \end{bmatrix} \begin{bmatrix} V_{id}^{\phi} \\ X_{id}^{\phi} \\ V_{id}^{\tau} \\ X_{id}^{\tau} \end{bmatrix} + \begin{bmatrix} C_1 & 0 & C_2 & C_3 \\ C_1 & 0 & C_2 & C_{31} \\ 0 & C_1 & C_2 & -C_3 \\ 0 & C_1 & C_2 & -C_3 \end{bmatrix} \begin{bmatrix} p_{id}^{\phi} \\ p_{id}^{\tau} \\ p_g^{\phi} \\ p_g^{\tau} \end{bmatrix}.$$

在APPSO算法中, 记

$$A'=\begin{bmatrix}\omega & -(\varphi+C_3) & 0 & 0\\ \omega & 1-(\varphi+C_3) & 0 & 0\\ 0 & 0 & \omega & -(\varphi-C_3)\\ 0 & 0 & \omega & 1-(\varphi-C_3)\end{bmatrix}. \tag{10}$$

推论 2 APPSO算法对任给初始微粒群收敛的一个充分条件是$\rho(A')<1$, 即

$$\begin{cases}-1<\omega<1,\\ |C_3|<\varphi<2+2\omega-|C_3|,\end{cases} \tag{11}$$

且当此条件满足时, $\rho(A')$越小, 收敛速度越快.

证 为方便描述, 令式(1)–(2)对应种群称为种群I, 式(3)–(4)对应种群称为种群II. 令

$$A_1=\begin{bmatrix}\omega & -(\varphi+C_3)\\ \omega & 1-(\varphi+C_3)\end{bmatrix},\ A_2=\begin{bmatrix}\omega & -(\varphi-C_3)\\ \omega & 1-(\varphi-C_3)\end{bmatrix},$$

则A_1和A_2分别为种群I、种群II的迭代矩阵, A_1和A_2的特征多项式分别为

$$f(\lambda_1)=\lambda^2+(\varphi+C_3-\omega-1)\lambda+\omega,$$
$$f(\lambda_2)=\lambda^2+(\varphi-C_3-\omega-1)\lambda+\omega.$$

令A_1的特征根为λ_1,λ_2, A_2的特征根为λ_3,λ_4, 有

$$\rho(A_1)=\max_{1\leqslant i\leqslant 2}|\lambda_i|=\max\{|\lambda_1|,|\lambda_2|\}, \tag{12}$$

$$\rho(A_2)=\max_{3\leqslant i\leqslant 4}|\lambda_i|=\max\{|\lambda_3|,|\lambda_4|\}, \tag{13}$$

则A的谱半径为

$$\rho(A')=\max_{1\leqslant i\leqslant 4}|\lambda_i|=\max\{|\lambda_1|,|\lambda_2|,|\lambda_3|,|\lambda_4|\}. \tag{14}$$

根据推论1, 由$\rho(A_1)<1$, $\rho(A_2)<1$可得

$$\begin{cases}-1<\omega<1,\\ 0<\varphi+C_3<2+2\omega,\\ 0<\varphi-C_3<2+2\omega.\end{cases} \tag{15}$$

由式(15)的后两式, 有

$$\begin{cases}-C_3<\varphi<2+2\omega-C_3,\\ C_3<\varphi<2+2\omega+C_3.\end{cases} \tag{16}$$

由于c_3可能取正, 也可能取负, 则C_3可能为正, 也可能为负, 则由$\rho(A')<1$, 有式(11)成立. 显然由式(11)可以推得$\rho(A')<1$, 即式(11)为APPSO法粒子轨迹收敛的充分条件. 因此推论2成立. 证毕.

3.3 粒子轨迹可靠收敛的参数设置(Parameter setting for reliably convergent trajectories of particles)

根据第3.2节, $\varphi=c_1R_1+c_2R_2\in(0,c_1+c_2)$, $|C_3|=|c_3|R_3\in(0,|c_3|)$, 则式(11)的第2个式子不能保证成立, 即原迭代公式是不能保证收敛的.

为使第2个式子能成立, 本文APPSO的参数C_1,C_2作如下调整:

$$C_1=|C_3|/2+c_1R_1,\ C_2=|C_3|/2+c_2R_2, \tag{17}$$

则有

$$\varphi=C_1+C_2=|C_3|+c_1R_1+c_2R_2. \tag{18}$$

这样式(11)中的$|C_3|<\varphi$自动成立. 由式(11)(18)有, $C_1+C_2+2|C_3|<2+2\omega$, 则

$$c_1+c_2+2|c_3|<2+2\omega. \tag{19}$$

考虑到很多PSO文献中$c_1=c_2$, 本文取

$$c_1=c_2=k|c_3|,\ k>0 \tag{20}$$

表示两个子种群相互之间的捕食、逃逸强度. 则

$$(2k+2)|c_3|<2+2\omega. \tag{21}$$

即

$$|c_3|<(1+\omega)/(k+1), \tag{22}$$

则参数按(17)–(18)设置时, APPSO粒子轨迹收敛的充分条件为

$$\begin{cases}-1<\omega<1,\\ |c_3|<(1+\omega)/(k+1).\end{cases} \tag{23}$$

具体参数设置步骤如下:

1) 在$(-1,1)$内设置ω, 在$(0,+\infty)$内设置参数k;

2) 在$(0,\dfrac{1+\omega}{k+1})$内设置参数$|c_3|$, 即可按式(20)求得$c_1,c_2$;

3) R_1,R_2,R_3确定后, 即可求得$|C_3|$, 由式(17)可以求得C_1,C_2.

即c_3的正负情况由第3.1节决定, c_3的绝对值范围由第3.3节决定. 本文将这样的算法称为交替捕食的PSO算法(alternately preying particle swarm optimization algorithm, APPSO).

3.4 APPSO粒子轨迹收敛速度分析(Convergence rate analysis of particles' trajectories of the APPSO)

3.4.1 谱半径计算(Spectral radius calculation)

在文献[3]的基础上, 可以求出APPSO的种群I粒子、种群II粒子迭代矩阵A_1和A_2谱半径: $-1<\omega\leqslant 0$时,

$$\rho(A_1)=\begin{cases}\dfrac{\alpha_1+\beta_1}{2}, & 0<\varphi+C_3<1+\omega,\\ \dfrac{-\alpha_1+\beta_1}{2}, & 1+\omega\leqslant\varphi+C_3<2+2\omega,\end{cases} \tag{24}$$

$$\rho(A_2)=\begin{cases}\dfrac{\alpha_2+\beta_2}{2}, & 0<\varphi-C_3<1+\omega,\\ \dfrac{-\alpha_2+\beta_2}{2}, & 1+\omega\leqslant\varphi-C_3<2+2\omega.\end{cases} \tag{25}$$

$0 < \omega < 1$时,

$$\rho(A_1) = \begin{cases} \dfrac{\alpha_1 + \beta_1}{2}, & 0 < \varphi + C_3 \leqslant (1 - \sqrt{\omega})^2, \\ \sqrt{\omega}, & (1-\sqrt{\omega})^2 < \varphi + C_3 < (1+\sqrt{\omega})^2, \\ \dfrac{-\alpha_1 + \beta_1}{2}, & (1 + \sqrt{\omega})^2 \leqslant \varphi + C_3 < 2 + 2\omega, \end{cases}$$

$$\rho(A_2) = \begin{cases} \dfrac{\alpha_2 + \beta_2}{2}, & 0 < \varphi - C_3 \leqslant (1 - \sqrt{\omega})^2, \\ \sqrt{\omega}, & (1-\sqrt{\omega})^2 < \varphi - C_3 < (1+\sqrt{\omega})^2, \\ \dfrac{-\alpha_2 + \beta_2}{2}, & (1 + \sqrt{\omega})^2 \leqslant \varphi - C_3 < 2 + 2\omega, \end{cases}$$

其中:

$$\alpha_1 = -(\varphi + C_3) + 1 + \omega,$$
$$\beta_1 = \sqrt{(\varphi + C_3 - 1 - \omega)^2 - 4\omega},$$
$$\alpha_2 = -\varphi + C_3 + 1 + \omega,$$
$$\beta_2 = \sqrt{(\varphi - C_3 - 1 - \omega)^2 - 4\omega}.$$

根据第3.1节, APPSO迭代矩阵的谱半径

$$\rho(A') = \max\{\rho(A_1), \rho(A_2)\}. \tag{26}$$

3.4.2　参数均未知(Parameters are all not known)

参数ω, k, c_3均未知时, 可以根据式(24)–(28), 分情况分析$\rho(A')$. 但是由于存在ω, φ, C_3这3个变量, 对$\rho(A')$的分析计算较为复杂. 本文通过SQRT序列[9]采样方法对$\rho(A')$进行分析.

由式(15)可得

$$|C_3| < 1 + \omega < 2, \tag{27}$$

$$0 < |C_3| < \varphi < 2 + 2\omega - |C_3| < 4. \tag{28}$$

即$\omega \in (-1, 1), C_3 \in (-2, 2), \varphi \in (0, 4)$. 本文在此参数范围内按SQRT序列进行采样, 采样规模为10^6. 由式(24)–(28)计算可得, 所有采样点的谱半径平均值为2.1533. 符合条件$\rho(A') < 1$的采样点有291659个, 约占30%, 这些采样点的谱半径平均值为0.7725. $\rho(A')$在$(0, 1)$区间的分布情况如图1所示.

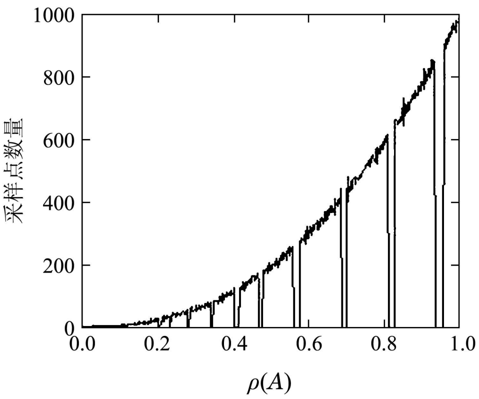

图 1　采样点数量按$\rho(A')$分布情况

Fig. 1　Distribution of the number of sampling points with $\rho(A')$

由图1可见, 随着$\rho(A')$的增加, 相应的采样点数量(近似为参数区域)大致呈现递增的趋势.

3.4.3　参数均已知(Parameters are all known)

若参数ω, k, c_3均已知, 由于R_1, R_2, R_3是$(0, 1)$范围内的随机数, $\rho(A')$与粒子轨迹收敛速度仍未知. 此时本文可以通过式(24)–(28)的分析计算来判断$\rho(A')$的大致范围, 也可以通过$\rho(A')$期望值的计算来了解粒子轨迹收敛速度.

比如$k = 2, \omega = 0.729$时, 由式(23)计算可得$|c_3| < 0.5763$, 令$|c_3| = 0.55$. 此时,

$$(1 - \sqrt{\omega})^2 = 0.0214,\ (1 + \sqrt{\omega})^2 = 3.4366,$$
$$C_3 = c_3 R_3 \in (-0.55, 0.55),$$
$$c_1 = c_2 = k|c_3| = 1.1,$$
$$\varphi = |C_3| + c_1 R_1 + c_2 R_2 \in (0, 2.75),$$
$$\varphi + C_3 = |C_3| + c_1 R_1 + c_2 R_2 + C_3 \in (0, 3.3),$$
$$\varphi - C_3 = |C_3| + c_1 R_1 + c_2 R_2 - C_3 \in (0, 3.3).$$

对照式(24)–(28), 比较$(1 - \sqrt{\omega})^2$, $(1 + \sqrt{\omega})^2$, $\varphi + C_3$, $\varphi - C_3$可知, 绝大部分情况下

$$\rho(A') = \sqrt{\omega} = 0.8538.$$

R_1, R_2, R_3在$(0, 1)$范围内按SQRT序列进行采样, 采样规模为10^6. 计算可得, 虽然有部分的采样点$\rho(A') > 0.8538$, 但这些采样点太少, $\rho(A')$期望值取4位有效数字时, 仍为0.8538.

3.4.4　部分参数已知(Some parameters are known)

若部分参数已设置, 部分参数尚未设置, 本文也可以通过式(24)–(28)的分析、$\rho(A')$期望值的计算来判断$\rho(A')$的大致范围, 了解粒子轨迹收敛速度分布, 以便完成参数设置.

比如$\omega = 0.729, k = 2$时, 由式(23)计算可得$|c_3| < 0.5763$即$-0.5763 < c_3 < 0.5763$. 类似于第3.4.3节, 计算可得

$$(1 - \sqrt{\omega})^2 = 0.0214,\ (1 + \sqrt{\omega})^2 = 3.4366,$$
$$\varphi + C_3 \in (0, 3.4578),\ \varphi - C_3 \in (0, 3.4578).$$

同样, 对照式(24)–(28), 可知大部分情况下$\rho(A') = \sqrt{\omega} = 0.8538$.

另外, 可在$(-0.5763, 0.5763)$内按步长0.005取c_3值, 同时R_1, R_2, R_3在$(0, 1)$范围内按SQRT序列进行采样, 采样规模为1000. 计算每个c_3下的$\rho(A')$期望值, 以c_3为横坐标, $\rho(A')$为纵坐标, 得到图2. 由图2可知, $\rho(A')$按直线$c_3 = 0$左右对称, 即c_3的正负对$\rho(A')$没有影响; 大部分情况下, $\rho(A')$比0.85略大点(按前面分析, 应是0.8538); 只有$c_3 = 0$附近的较小区域内, c_3向0趋近时, $\rho(A')$迅速增加. 但$c_3 = 0$附时, 经计算可得$\rho(A') = 0.8538$, 并不是趋近于1.

由图2可了解不同c_3时的粒子轨迹期望收敛速度, 便于对参数的完整设置.

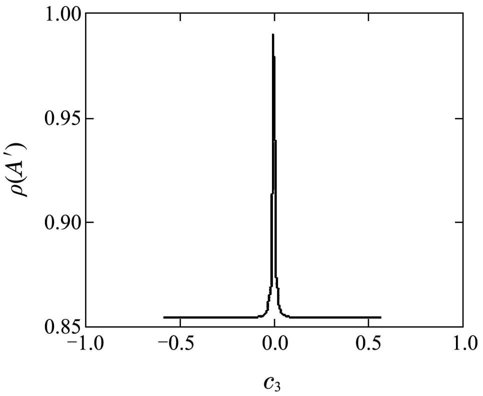

图 2 $\rho(A')$按c_3分布情况

Fig. 2 Distribution of $\rho(A')$ with c_3

4 算例(Example)

为验证APPSO的有效性, 本文对文献[10]测试函数$F1$–$F4$、文献[11]测试函数$f7$和$f8$进行仿真分析.

本文采用标准PSO(SPSO), PSOPC, APPSO进行比较. APPSO的参数按照第3.4.3节中的设置, $p = 0.85$, $A_N = 0.01$. 在SPSO中$\omega = 0.729$, $c_1 = c_2 = 1.49445$[12]. 同样, 可计算出SPSO的$\rho(A)$基本上也为$\sqrt{\omega} = 0.8538$. 可见此时SPSO, APPSO粒子运动轨迹收敛速度基本相同.

这3种方法种群规模均为30, PSOPC和APPSO中$S^{\phi} = S^{\tau} = 10$. 迭代次数均为2000. 为便于比较3种方法计算6个测试函数的收敛情况, 参考文献[10], 若求得的结果与理想最优解误差小于10^{-4}, 本文就视其为达到收敛. 算法第一次达到收敛的代数、时间分别称为收敛代数、收敛时间. 采用MATLAB 7.1编程工具, 计算机的配置为: Pentium Dual 2.00 GHZ, 2 GHz内存, Windows XP. 每个算法独立执行50次, 平均试验的结果如表1所示(APPSO的粒子运动轨迹收敛, 只是说明种群中各个粒子最终收敛到一个区域, 但不一定收敛到理想区域, 因此并不意味着计算结果应该100%收敛, 因此APPSO的收敛率虽高于其他方法, 但没有全为100%).

表 1 基准函数的优化结果比较

Table 1 Optimization results comparation of benchmark functions

算法	性能	$F1$	$F2$	$F3$	$F4$	$f7$	$f8$
SPSO	收敛率	0.4	0.38	0.2	0.38	0.44	0.3
	平均收敛代数	1230.18	853.26	1351.6	1092.24	217.96	1913.32
	平均收敛时间 / ms	47.21	31.42	159.98	85.53	24.88	131.06
PSOPC	收敛率	0.58	0.52	0.22	0.46	0.66	0.44
	平均收敛代数	908.12	703.88	1037.38	865.84	193.58	1538.26
	平均收敛时间 / ms	35.93	27.54	124.65	71.17	22.72	111.42
APPSO	收敛率	0.84	0.76	0.42	0.64	0.9	0.74
	平均收敛代数	633.04	535.02	737.08	533.04	128.68	1123.82
	平均收敛时间 / ms	27.25	22.34	98.91	45.08	16.48	84.33

由表1可以看出, PSOPC和APPSO的收敛速度、收敛率均优于SPSO; APPSO收敛速度、收敛率较PSOPC明显更优. 可见APPSO的交替捕食策略、使粒子运动轨迹可靠收敛的参数设置, 提高了算法的搜索性能.

5 小结(Conclusions)

本文在捕食者–猎物协同进化机制基础上, 在算法迭代过程中让捕食逃逸学习因子的正负号发生变化, 提出一种交替捕食的PSO算法. 通过对算法迭代过程的分析, 给出了粒子运动轨迹可靠收敛的参数取值范围, 构建了一种能保证粒子运动轨迹收敛的参数设置方法. 利用算法迭代矩阵谱半径的计算, 对APPSO算法粒子运动轨迹收敛情况进行了分析. 仿真结果显示APPSO是有效的, 具有较佳的搜索性能.

参考文献(References):

[1] EBERHART R C, KENNEDY J. Particle swarm optimization [C] //*Proceedings of IEEE Internation Conference on Neural Network*. New York: IEEE, 1995, 4: 1942 – 1948.

[2] VAN DEN BERGH F, ENGELBRECHT A P. A study of particle swarm optimization particle trajectories [J]. *Information Sciences*, 2006, 176(8): 937 – 971.

[3] 肖健梅, 李军军, 王锡淮. 梯度微粒群优化算法及其收敛性分析 [J]. 控制与决策, 2009, 24(4): 560 – 564.
(XIAO Jianmei, LI Junjun, WANG Xihuai. Convergence analysis of particle swarm optimization and its improved algorithm based on gradient [J]. *Control and Decision*, 2009, 24(4): 560 – 564.)

[4] FERNANDEZ-MARTINEZ J L. Stochastic stability analysis of the

linear continuous and discrete PSO models [J]. *IEEE Transactions on Evolutionary Computation*, 2011, 15(3): 405 – 423.

[5] 李军军, 甘世红, 许波桅. 基于伪幂函数的离散粒子群算法及其应用 [J]. 控制理论与应用, 2011, 28(6): 834 – 838.
(LI Junjun, GAN Shihong, XU Bowei. Discrete particle swarm optimization algorithm based on pseudo power function and its applications [J]. *Control Theory & Applications*, 2011, 28(6): 834 – 838.)

[6] HAO G. A new particle swarm algorithm and its globally convergent modifications [J]. *IEEE Transactions on systems*, 2011, 41(5): 1334 – 1351.

[7] LIU W, CHEN H X, CHEN H N, et al. Improved particle swarm optimizer based on predator-prey coevolution model [C] //*Proceedings of International Conference on Computational Intelligence and Industrial Application*. New York: IEEE, 2010, 8: 88 – 91.

[8] 刘微, 陈贺新, 陈瀚宁, 等. 改进的PSO算法在RFID网络调度中的应用 [J]. 吉林大学学报(信息科学版), 2011, 29(2): 121 – 127.
(LIU Wei, CHEN Hexin, CHEN Hanning, et al. Improved particle swarm optimizer for RFID network planning [J]. *Journal of Jilin University (Information Science Edition)*, 2011, 29(2): 121 – 127.)

[9] 朱云飞, 罗彪, 郑金华, 等. 基于拟蒙特卡罗方法的进化算法搜索鲁棒最优解的性能提高研究 [J]. 模式识别与人工智能, 2011, 24(2): 201 – 209.
(ZHU Yunfei, LUO Biao, ZHENG Jinhua, et al. Research on increasing the performance of evolutionary algorithm in searching robust optimal solutions based on Quasi-Monte Carlo method [J]. *Pattern Recognition and Artificial Intelligence*, 2011, 24(2): 201 – 209.)

[10] 王丽芳, 曾建潮. 基于微粒群算法与模拟退火算法的协同进化方法 [J]. 自动化学报, 2006, 32(4): 630 – 635.
(WANG Lifang, ZENG Jianchao. A cooperative evolutionary algorithm based on particle swarm optimization and simulated annealing algorithm [J]. *Acta Automatica Sinica*, 2006, 32(4): 630 – 635.)

[11] 杜海峰, 公茂果, 刘若辰, 等. 自适应混沌克隆进化规划算法 [J]. 中国科学E: 信息科学, 2005, 35(8): 817 – 829.
(DU Haifeng, GONG Maoguo, LIU Ruochen, et al. Adaptive chaos clonal evolutionary programming [J]. *Science in China, Serials E: Information Sciences*, 2005, 35(8): 817 – 829.)

[12] EBERHART R C, SHI Y. Comparing inertia weights and constriction factors in particle swarm optimization [C] //*Proceedings of the IEEE Conference on Evolutionary Computation*. New York: IEEE, 2000, 1: 84 – 88.

作者简介:

李军军　(1981–), 男, 讲师, 博士, 目前研究方向为智能控制、优化算法, E-mail: jsliljj@163.com;

黄有方　(1959–), 男, 教授, 博士生导师, 目前研究方向为物流管理与工程, E-mail: yhuang@shmtu.edu.cn;

杨　斌　(1975–), 男, 副教授, 硕士生导师, 目前研究方向为物流管理与工程, E-mail: binyang@shmtu.edu.cn.