变步长自适应的改进人工鱼群算法

朱旭辉 倪志伟 程美英
(合肥工业大学管理学院 合肥 230009)
(合肥工业大学过程优化与智能决策教育部重点实验室 合肥 230009)

摘 要 针对人工鱼群算法在函数优化中存在陷入局部最优、后期收敛速度慢及结果精度不高等问题,通过改进鱼群算法中觅食行为及自适应调整人工鱼步长,提出了一种变步长自适应的改进人工鱼群算法。证明了该算法的全局收敛性,从而增加了其理论基础。最后,10个标准函数测试结果表明,改进后的人工鱼群算法在跳出局部最优、收敛速度、精度和稳定性方面都优于原鱼群算法和萤火虫算法,在结果精度和稳定性方面优于文献[9,23,24]的方法。

关键词 人工鱼群算法,变步长,自适应步长,全局收敛,函数优化
中图法分类号 TP18 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.2.044

Self-adaptive Improved Artificial Fish Swarm Algorithm with Changing Step

ZHU Xu-hui NI Zhi-wei CHENG Mei-ying
(School of Management, Hefei University of Technology, Hefei 230009, China)
(Key Laboratory of Process Optimization and Intelligent Decision-making, Ministry of Education, Hefei 230009, China)

Abstract The artificial fish swarm algorithm in function optimization problems has some defectives, such as falling into local optimum value, converging slowly in the later period and acquiring solutions inaccurately. In order to overcome these shortcomings, a new self-adaptive artificial fish swarm algorithm with changing step was proposed by improving foraging behavior and adjusting self-adaptive step of artificial fish swarm algorithm. In addition, the paper strengthened the theoretical basis of the algorithm by proving the global convergence. Finally, the experimental results of 10 typical functions show that the proposed algorithm is superior to the original artificial fish swarm algorithm and artificial glowworm swarm optimization algorithm in overcoming the local optimum, convergence efficiency, computational precision and stability. Furthermore, the method is superior to the paper [23], [24] and [9] in computational precision and stability.

Keywords Artificial fish swarm algorithm, Changing step, Self-adaptive step, Global convergence, Function optimization

1 引言

各个领域中许多问题都需要建立模型来解决,最终都要归结为函数优化问题,因此函数优化问题研究越来越重要,其特点是:多变量、多极值、震荡性强、非线性。人们从仿生学机理中获得启发,提出了许多群体智能优化算法,但每一种群体智能算法都有其缺陷。例如蚁群算法前期搜索盲目性较大,计算资源消耗大;遗传算法编码实现过程复杂;粒子群算法易陷入局部最优等。

2001年李晓磊[1]通过模仿鱼类觅食行为将动物自治体概念引入到群体优化方法中,提出了人工鱼群算法(Artificial Fish Swarm Algorithm)思想;2002年李晓磊[2]进一步对人工鱼群算法进行完善及改进,该算法的特点是:初始值要求不高;鲁棒性强;收敛速度快;全局搜索能力强。人工鱼群算法设计参数较少,实现较简单且效率高,该算法的思想已应用到计算机领域[3-5]。该算法还有以下缺点:前期收敛速度快,但后期收敛速度慢;寻优结果精度不高。国内外学者针对以上问题展开研究:肖键梅等[6]为避免算法的退化,引入适应值变化率和变化方差作为衡量参数变化的标准;黄华娟等[7]通过改变步长和拥挤度因子提高算法效率;Si He等[8]将模糊聚类思想引入算法中以提高其效率;张大斌等[9]基于差分策略改进人工鱼群算法提高其搜索效率;张梅凤等[10]引入变异算子和小生境思想改进人工鱼群算法;王培崇等[11]将小生境排挤机制引入人工鱼群算法并证明其收敛性;李咏梅等[12]将人工鱼群算法思想应用到萤火虫算法以提高算法效率;刘薇等[13]将K-means和人工鱼群算法混合应用于聚类问题;段其昌等[14]将粒子群算法与人工鱼群算法相结合,以提高收敛速度;陶杨等[15]将人工鱼之间的平均距离作为确定步长因素并引入遗传变异算子,提出基于群体行为的自适应变异算子算法;刘彦君等[16]建立人工鱼视野范围与步长的线性关系且通

到稿日期:2014-03-24 返修日期:2014-07-29 本文受国家自然科学基金(71271071),国家"863"云制造主题项目(2011AA040501),青年科学基金项目(71301041)资助。

朱旭辉(1991—),男,博士生,主要研究方向为进化计算等,E-mail:943177204@qq.com;倪志伟(1963—),男,教授,博士生导师,主要研究方向为人工智能、机器学习、云计算等;程美英(1983—),女,博士生,主要研究方向为智能计算等。

过不同方式确定视野范围，提出自适应视野和步长的改进人工鱼群算法；欧阳喆等[17]引入荧光因子调整步长的自适应机制应用到萤火虫算法中；张仲海等[18]将变步长随机共振原理引入到粒子群算法中以提高算法效率；马宪民等[19]提出自适应视野改进人工鱼群算法并将其应用于最短路问题。

上述学者的研究在一定程度上提高了人工鱼群算法的优化性能，但后期收敛速度慢、结果精度低等问题仍未得到很好的解决。因此，本文在此基础上进一步改进算法，进行改进提出变步长自适应的改进人工鱼群算法，主要是通过对鱼群算法中觅食行为进行改进，加大其跳出局部最优的可能，从而达到全局最优；及改进算法搜索过程中的步长，根据寻优过程中的具体情况确定合适的步长，显著提高了算法后期的收敛速度及精度。论文重要创新点之一是证明了改进人工鱼群算法的全局收敛性。最后通过仿真实验演示函数优化的过程以及比较计算结果，表明变步长自适应的改进人工鱼群算法具有跳出局部最优能力强、全局收敛速度快和计算精度较高等特点。

2 人工鱼群算法简介

人工鱼群算法的基本思想是在一片水域中，营养物质最丰富的地方往往就是鱼的数目最多的地方，根据这一特点模仿鱼群觅食的行为，进而实现全局最优。

假设有 N 条人工鱼，其中一条人工鱼状态 $X_i=(x_1,x_2,\cdots,x_n)$，其中 $x_i(i=1,2,\cdots,n)$ 为寻优变量，适应度函数 $Y=f(X)$（食物浓度），人工鱼之间的距离 $d_{ij}=\|X_i-X_j\|$，人工鱼感知范围为 $Visual$，人工鱼移动步长为 $Step$，人工鱼拥挤度因子为 δ，尝试次数为 try_number，(0,1)随机数用 $Rand()$ 表示。

(1)觅食行为

在解决极大值问题时，设当前人工鱼的状态为 X_i，其适应度函数为 Y_i，随机选择其感知范围内一个状态 X_j，其适应度函数为 Y_j，若 $Y_i<Y_j$，则向 X_j 移动一步；反之，若随机一定次数后仍不满足前进条件，则随机移动一步，觅食行可以表示为：

$$X_{i/next}=\begin{cases}X_i+Rand*Step*\dfrac{X_j-X_i}{\|X_j-X_i\|}, & Y_i<Y_j\\ X_i+Rand*Step, & \text{反之}\end{cases}$$

(2)聚群行为

设当前人工鱼的状态为 X_i，其适应度函数为 Y_i，搜索该人工鱼范围内人工鱼数目 n_f 及中心位置 X_c。若 $Y_c/n_f>\delta Y_i$，表明中心位置食物较多且不太拥挤，则向中心位置移动一步；反之，立即执行觅食行为。聚群行为可以表示为：

$$X_{i/next}=\begin{cases}X_i+Rand*Step*\dfrac{X_c-X_i}{\|X_c-X_i\|}, & Y_c/n_f>\delta Y_i\\ prey(X_i), & \text{反之}\end{cases}$$

(3)追尾行为

当前人工鱼搜索其视野范围内食物浓度最大的人工鱼 $X_{\max}$，若 $Y_{\max}/n_f>\delta Y$，则向该鱼移动一步；反之，则立即执行觅食行为。追尾行为可以表示为：

$$X_{i/next}=\begin{cases}X_i+Rand*Step*\dfrac{X_{\max}-X_i}{\|X_{\max}-X_i\|}, & Y_{\max}/n_f>\delta Y_i\\ prey(X_i), & \text{反之}\end{cases}$$

(4)公告板

将状态最优的人工鱼记录下来，以保证最终寻优的结果输出。

3 变步长自适应的改进人工鱼群算法

假设常系数为 C，当前迭代次数为 l，最大迭代次数为 L，概率因子为 $\alpha(0\leqslant\alpha\leqslant1)$。

3.1 鱼群算法中觅食行为的改进

人工鱼群算法具有较强的全局收敛性，在全局收敛过程中跳出局部最优的能力是由算法中觅食行为保证的。觅食行为中存在随机移动性能使算法跳出局部最优达到全局最优，随机移动的盲目性，降低了其收敛速度以及达到全局最优的可能性。

因此，本文对觅食行为进行改进（以解决极大值问题为例）。

原觅食行为：设当前人工鱼状态为 X_i，其适应度函数为 Y_i，在其感知范围内随机选择某一状态 X_j，其适应度函数为 Y_j，令 $\Delta Y=Y_j-Y_i$，若 $\Delta Y>0$，则接受新解；反之，重新寻找状态 X_j，尝试 try_number 次后仍不满足 $\Delta Y>0$，则随机移动一步。

改进觅食行为：设当前人工鱼状态 X_i，其适应度函数为 Y_i，在其感知范围内随机选择某一状态 X_j，其适应度函数为 Y_j，令 $\Delta Y=Y_j-Y_i$，若 $\Delta Y>0$，则接受新解；若 $\Delta Y\leqslant0\&\&M*k*\exp\Delta Y>\alpha$，则亦接受新解；否则，重新寻找状态 X_j，尝试 try_number 次仍不满足上述条件，则随机移动一步。

原觅食行为与改进觅食行为如图 1 所示，通过比较可以看出，改进后的觅食行为不仅可以接受优解，而且还能以一定概率接受次优解，以防止其陷入局部最优，从而使其从局部最优中跳出，获取全局最优解。

原觅食行为

```
试探次数 k=1;
while(k<trynumber)
    ΔY=Yj-Yi;
    if(ΔY>0) then 接受新解,跳出循环
    esle k=k+1;
  end
if (k>trynumber) then 随机移动一步
```

改进觅食行为

```
试探次数 k=1;
迭代次数 l;
while(k<trynumber)
    ΔY=Yj-Yi;
    if(ΔY>0) then 接受新解,跳出循环
    esle if (exp(C*l*ΔY)>α)
    then 亦接受新解,跳出循环;
    k=k+1;
    end
if (k>trynumber) then 随机移动一步
```

图 1 原觅食行为与改进觅食行为比较

3.2 鱼群算法中采用变自适应步长技术

人工鱼群算法是一种较好的全局搜索算法，但算法后期

人工鱼个体徘徊在峰值附近，甚至会出现振荡现象，这是导致收敛速度慢、结果精度不高的主要原因。多次仿真实验分析表明，这种现象主要是人工鱼移动步长选取不够恰当造成的。由于在人工鱼的觅食过程中，每一条人工鱼采用固定步长进行移动，若固定步长较大，则可能导致后期人工鱼跳离最优解，出现收敛速度慢、精度低问题；若固定步长较小，结果精度较高，但收敛速度太慢，易陷入局部极值而难以跳出。因此，算法搜索过程中需要选取合适的移动步长以保证其收敛速度、结果精度以及防止其陷入局部最优无法跳出。

通过对人工鱼群算法存在的问题进行分析，本文提出了变步长自适应人工鱼群算法，即在同一次迭代中，对离更优值距离较远的个体使用较大的移动步长，加快其收敛速度；相反，对离更优值距离较近的个体使用较小的移动步长，避免产生振荡现象，提高寻优结果精度。这种自适应变步长选择方式，既考虑到算法收敛速度，又提高了求解精度。

设当前迭代次数 $l<L$ 时，当前人工鱼状态为 X_i，其适应度函数为 Y_i，在其感知范围内某一状态 X_j，其适应度函数为 Y_j，若 $Y_j>Y_i$，则选择其移动步长 $Step=Rand*\|X_j-X_i\|$。该自适应变步长既可以加快收敛速度，又可以提高结果精度，加入 $Rand()$ 函数是为防止其收敛过快而陷入局部最优。在聚群追尾后的觅食行为中，若寻找不到更优解，为了增大随机性使其跳出局部最优，继而找到全局最优解，仍然采用原人工鱼群算法中较为通用的移动步长 $Step=0.3$。

3.3 改进人工鱼群算法实现步骤

Step1 初始化：初始化 N 条人工鱼，感知范围 $Visual$，移动步长 $Step$，拥挤度因子 δ，尝试次数 try_number，当前迭代次数 $l=0$，最大迭代次数 L，概率因子 $\alpha(0<\alpha<1)$，常系数 C。

Step2 公告板初始化：计算出 N 条人工鱼的适应度函数，得到最优值，赋值给公告板。

Step3 执行聚群行为：若中心位置 X_c 优于当前人工鱼位置 X_i 且不太拥挤，则令 $Step=Rand*\|X_c-X_i\|$，向中心位置移动一步；否则，转 Step5。

Step4 执行追尾行为：若当前人工鱼 X_i 感知范围内最优鱼 X_{max} 优于当前人工鱼且不太拥挤，则令 $Step=Rand*\|X_{max}-X_i\|$，向最优鱼移动一步；否则，转 Step5。

Step5 执行觅食行为：选择当前人工鱼 X_i 感知范围内某一随机状态 X_j，若 $\Delta Y=Y_j-Y_i>0$ 或者 $\exp(C*l*\Delta Y)>\alpha$，则令 $Step=Rand*\|X_j-X_i\|$，向该人工鱼移动一步；否则，尝试 try_number 次后结果仍未变，则更新公告板。

Step6 若 $l<L$，则令 $l=l+1$，转 Step3；若 $l\geqslant L$，则转 Step7。

Step7 输出公告板信息，即最优值。

4 变步长自适应的改进人工鱼群算法性能分析

4.1 变步长自适应的改进人工鱼群算法收敛性分析

为了增加其理论基础，通过建立吸收态 Markov 过程模型证明其全局收敛性，人工鱼群算法中个体鱼通过聚群、追尾和觅食过程不断更新人工鱼状态，最终达到最优状态。

定义 1 设 $\{X(t)\}_{t=0}^{\infty}$ 对任意 n 个不同的 $t_1,t_2,\cdots,t_n\in T$ 且 $t_1<t_2<\cdots<t_{n-1}$，有

$$P(X(t_n)\leqslant X_n\mid X(t_{n-1})=X_{n-1},\cdots,X(t_1)=X_1)$$
$$=P(X(t_n)\leqslant X_n\mid X(t_{n-1})=X_{n-1})$$

则称 $\{X(t)\}_{t=0}^{\infty}$ 为 Markov 过程。

定义 2 任意给定某一区域 Markov 过程 $\{X(t)\}_{t=0}^{\infty}$ 和该区域的最优状态空间 $Y^*\subset Y$，若满足 $P(X(t+1)\notin Y^*\mid X(t)\in Y^*)=0$，则称 $\{X(t)\}_{t=0}^{\infty}$ 为一个吸收态 Markov 过程。

引理 1 设鱼群算法对应的随机过程为 $\{X(t)\}_{t=0}^{\infty}$，则 $\{X(t)\}_{t=0}^{\infty}$ 具有 Markov 性，同时 $\{X(t)\}_{t=0}^{\infty}$ 也是一个吸收态 Markov 过程。

证明：由鱼群算法寻优过程可知，$\{X(t)\}_{t=0}^{\infty}$ 为离散时间随机过程，因为人工鱼状态 $X(t)$ 只由 $X(t-1)$ 决定，$X(0)$ 在初始化时可以随机选取，所以 $P(X(t)\mid X(t-1),X(t-2),\cdots,X(0))=P(X(t)\mid X(t-1))$。即 $\{X(t)\}_{t=0}^{\infty}$ 具有 Markov 性（设 $X(t)\in Y$ 为离散状态空间，即 $\{X(t)\}_{t=0}^{\infty}$ 为 Markov 链）。

由鱼群算法聚群、追尾和觅食行为知，当 $X(t)\in Y^*$ 为最优解空间时，$X(t+1)\in Y^*$，因此 $P(X(t+1)\notin Y^*\mid X(t)\in Y^*)=0$，即 $\{X(t)\}_{t=0}^{\infty}$ 是一个吸收态 Markov 过程。证毕。

定义 3 给定某一区域的吸收态 Markov 过程 $\{X(t)\}_{t=0}^{\infty}$（$\forall X(t)\in Y$）和最优状态空间 $Y^*\subset Y$，记 $\delta(t)=p(X(t)\in Z^*)$ 表示 t 时刻在某一区域达到最优状态的概率，若 $\lim\limits_{t\to\infty}\delta(t)=1$，则称 $\{X(t)\}_{t=0}^{\infty}$ 收敛。

引理 2 给定变步长自适应的改进人工鱼群算法在某一区域对应一个吸收态的 Markov 过程 $\{X(t)\}_{t=0}^{\infty}$（$\forall X(t)\in Y$）和最优状态空间 $Y^*\subset Y$，若 $P(X(t)\in Y^*\mid X(t-1)\notin Y^*)\geqslant d(t-1)\geqslant 0$ 且 $\lim\limits_{t\to\infty}\prod\limits_{i=0}^{t}[1-d(i)]=0$，则 $\lim\limits_{t\to\infty}\delta(t)=1$，其中 $\delta(t)=P(X(t)\in Y^*)$。

证明：根据全概率公式，

$$\delta(t)=[1-\delta(t-1)]P(X(t)\in Y^*\mid X(t-1)\notin Y^*)+\delta(t-1)P(X(t)\in Y^*\mid X(t-1)\in Y^*)(\forall t=1,2,\cdots)$$

由引理 1 知，$\{X(t)\}_{t=0}^{\infty}$ 是一个吸收态 Markov 过程，则 $P(X(t)\in Z^*\mid X(t-1)\in Z^*)=1$，即有

$$\begin{aligned}\delta(t)&=[1-\delta(t-1)]P\{X(t)\in Y^*\mid X(t-1)\notin Y^*\}+\delta(t-1)\\&\Rightarrow 1-\delta(t)=[1-P(X(t)\in Y^*\mid X(t-1)\notin Y^*)]*[1-\delta(t-1)]\\&\Rightarrow 1-\delta(t)\leqslant[1-d(t-1)]\cdot[1-\delta(t-1)]\\&=[1-\delta(0)]\prod_{i=0}^{t-1}[1-d(i)]\\&\Rightarrow\lim_{t\to\infty}\delta(t)\geqslant 1-[1-\delta(0)]\lim_{t\to\infty}\prod_{i=0}^{t-1}[1-d(i)]\\&\geqslant 1-[1-\delta(0)]\lim_{t\to\infty}\prod_{i=0}^{t}[1-d(i)]\\&\Rightarrow\lim_{t\to\infty}\delta(t)\geqslant 1\end{aligned}$$

因为概率 $\delta(t)\leqslant 1$，所以 $\lim\limits_{t\to\infty}\delta(t)=1$。证毕。

定理 变步长自适应的改进人工鱼群算法以概率 1 收敛到全局最优解。

证明：假设问题的解空间为 Ω，不妨设其有 m 个局部最优区域，分别为 $\Omega_1,\Omega_2,\cdots,\Omega_m$。

鱼群算法寻优过程中，必有人工鱼搜索到某一个局部最优值区域，设人工鱼 X_i 进入局部最优值区域 Ω_1，由引理 2

知，该人工鱼收敛到区域 Ω_1 最优值 Y_1。由于人工鱼群算法中进行聚群与追尾操作时，目标位置不能太拥挤(由拥挤度因子控制)，否则，人工鱼会向其他区域移动，即每个局部最优区域人工鱼数量不能超过一定数目，因此当局部最优区域 Ω_1 内人工鱼数目达到一定数量后，其他人工鱼会搜索剩下的局部最优区域，不妨设为 Ω_2，同时也会收敛到局部最优区域 Ω_2 最优值 Y_2。同理，人工鱼会遍历剩下 $m-2$ 个局部最优区域，其局部最优解分别为 $Y_3, Y_4, \cdots, Y_m$。

在改进的人工鱼群中，人工鱼会以一定概率接受次优解，这样可以保证人工鱼以一定的概率跳出局部最优区域，加快鱼群算法搜索到全局最优解。

鱼群算法中有公告板记录每次迭代的最优值，当达到某一条件结束搜索时，输出公告板记录的最优值，即全局最优值 $Y_{最优}=\max\{Y_1, Y_2, \cdots, Y_m\}$。

综上所述：变步长自适应的改进人工鱼群算法以概率 1 收敛到全局最优解。

4.2 改进人工鱼群算法的时间复杂度分析

各种群体算法在实际应用中通常采用时间复杂度估算算法执行效率的高低。在算法的设计和分析中，仍然采用实用性的复杂度概念，把求解时间的关键操作(如加、减、乘、除、比较等运算)定义为基本操作，一般把算法执行基本操作的次数视为算法的时间复杂度[20]。

鱼群算法在寻优过程中假设其种群规模为 N，根据改进人工鱼群算法实现步骤分析算法的时间复杂度。

Step1 初始化 N 条人工鱼需要 N 次，时间复杂度为 $O(N)$，初始化其他参数需要常数次。

Step2 初始化公告板，需要赋值 1 次，比较 $N-1$ 次，时间复杂度为 $O(N)$。

Step3 聚群行为计算拥挤度因子需要 N 次，判断 1 次，移动 1 次，N 条人工鱼需要聚群 N 次，故时间复杂度为 $O(N^2+2*N)$。

Step4 追尾行为寻找最优状态需要 N 次，计算拥挤度因子需要 N 次，判断 1 次，移动 1 次，N 条人工鱼需要追尾 N 次，故时间复杂度为 $O(2*N^2+2*N)$。

Step5 觅食行为需要最多 try_number 次，最少 1 次，N 条人工鱼需要觅食 N 次，故时间复杂度最多为 $O(N*try\_number)$。

Step6，Step7，时间复杂度为 $O(1)$。

综上所述：改进的人工鱼群算法经过 l 次迭代后，整个算法的时间复杂度为：

$$O(l*(3*N^2+N*try\_number+6*N))$$

5 仿真实验分析

5.1 仿真实验环境

仿真实验环境：本文所涉及的代码均采用 Matlab R2012a 软件编写，编译运行的 PC 机参数为 32 位 Windows 7 操作系统、Intel(R) Core(TM)2 E7500 2.93GHz CPU、2.00GB 内存。

5.2 测试函数集

为了测试变步长自适应的改进人工鱼群算法的有效性，本文选取了 8 个经典的测试函数(f_3 和 f_7 函数相同，但定义域范围不同)，f_1 选自文献[2]，f_2-f_4 选自文献[23]，f_5-f_8 选自文献[24]，如图 2 所示。

$f_1(x)=\prod_{i=1}^{2}\sin(x_i)/x_i, x_i\in[-10,10]$;

$\max f_1=1$

$f_2(x)=\sum_{i=1}^{2}x_i^2, x_i\in[-5,5]$;

$\min f_2=0$

$f_3(x)=4x_1^2-2.1x_1^4+\frac{x_1^6}{3}+x_1x_2-4x_2^2+4x_2^4, x_i\in[-5,5]$ $(i=1,2)$;

$\min f_3=-1.03162845348988$

$f_4(x)=(x_1^2+x_2^2)^{\frac{1}{4}}[\sin(50(x_1^2+x_2^2)^{0.1})+1.0], x_i\in[-10,10]$ $(i=1,2)$;

$\min f_4=0$

$f_5(x)=x_1^2+x_2^2+25(\sin^2x_1+\sin^2x_2), x_i\in[-2\pi,2\pi](i=1,2)$;

$\min f_5=0$

$f_6(x)=100(x_1^2-x_2)^2+(1-x_1)^2, x_i\in[-2.048,2.048]$ $(i=1,2)$;

$\min f_6=0$

$f_7(x)=(4-2.1x_1^2+\frac{x_1^4}{3})x_1^2+x_1x_2+(-4+4x_2^2)x_2^2, x_1\in[-3,3], x_2\in[-2,2]$;

$\min f_7=-1.0316$

$f_8(x)=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi})\cos x_1+10$, $x_1\in[-5,10], x_2\in[0,15]$;

$\min f_8=0.398$

$f_9(x)=-\sum_{i=1}^{2}x_i\sin(\sqrt{|x_i|}), x_1, x_2\in[-500,500]$(Generalize-Schwefel(n=2)函数)

$\min f_9=-837.96$

$f_{10}(x)=\{\sum_{i=1}^{5}i[(i+1)x_1+i]\}\{\sum_{i=1}^{5}i[(i+1)x_2+i]\}, x_1x_2\in[-10,10]$(SH-Shubbert(n=2)函数)

$\min f_{10}=-186.7309$

图 2 函数测试集

5.3 结果分析

为了验证本文算法与其它文献算法之间比较的可靠性，实验结果均为本文算法独立运行 20 次所得平均值。

(1)采用测试函数 f_1 将本文算法与原人工鱼群算法[2]、萤火虫算法[21]比较。

设置参数与原人工鱼群算法相同，种群规模 $N=50$，人工鱼感知范围 $Visual=2.5$，尝试次数 $try\_number=50$，常系数 $C=200$，忽略拥挤因子 δ(局部极值不是很严重的函数优化问题，可以忽略掉拥挤因子 δ)，概率因子 $\alpha=0.9$，萤火虫算法参数设置如文献[14](种群规模 $N=50$)，最大迭代次数 $L=20$，50，100，200，300，测试函数 f_1 为非线性函数，在全局最优值的周围存在许多局部极最优值，寻常算法极易陷入局部最优，出现振荡现象，适用于验证算法的性能。

由表 1 和图 3 可以看出，本文算法收敛速度及精度明显优于原鱼群算法及萤火虫算法，尤其本文算法在 10 次迭代后收敛的精度就已经比较高了，萤火虫算法初始较快，但寻优收敛速度及精度明显不如鱼群算法。

由表 2 可知，本文算法在 20，50，100，200，300 次的迭代中，x_1，x_2 取值精度及函数值的精度均显著优于原鱼群算法和萤火虫算法，尤其本文算法在寻优 50 次的结果精度已经高

于原鱼群算法和萤火虫算法寻优 300 次的结果，表明本文算法在收敛速度和精度方面有了显著提高。同时可以看出，萤火虫算法在寻优过程中略逊于原鱼群算法。

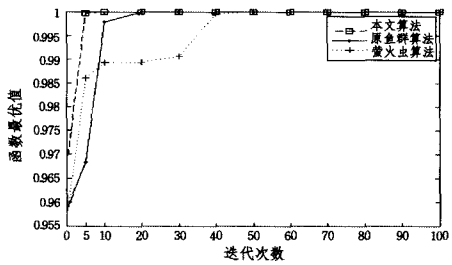


图 3　算法收敛速度对比

表 1　本文算法、原鱼群算法、萤火虫算法收敛速度对比

迭代次数	本文算法	原鱼群算法	萤火虫算法
5	0.999735304493828	0.968432881570534	0.986044430428463
10	0.999998979981686	0.997792934633674	0.989230045268146
20	0.999999992061611	0.999953689426745	0.989268080564116
30	0.999999999995475	0.999899487480609	0.990591373374653
40	0.999999999999985	0.999999950048759	0.999553219744618
50	0.999999999999945	0.999993776121705	0.999998329515829
60	0.999999999999931	0.999999981969207	0.999997747621443
70	0.999999999999702	0.999999970638393	0.999983792681562
80	0.999999999999999	0.999999830921065	0.999994362129887
90	0.999999999999982	0.999999936134158	0.999999872354650
100	1.000000000000000	0.999999758792546	0.999999804972407

表 2　本文算法、原鱼群算法、萤火虫算法性能对比

算法	迭代次数	x_1 取值	x_2 取值	函数最优解
萤火虫算法		−0.141103264232798 1.0e−00	0.216043736811509 1.0e−00	0.988949659879086
原鱼群算法	20	0.67207160497879 1.0e−01	−0.24888151205153 1.0e−01	0.999144213806106
本文算法		0.003050053377020 1.0e−03	0.307317956618782 1.0e−03	0.999999984257729
萤火虫算法		0.1557953172178 1.0e−02	0.4673435363146 1.0e−02	0.999995955302800
原鱼群算法	50	−0.1514343734196 1.0e−02	−0.2914786325828 1.0e−02	0.999998201798475
本文算法		0.054586123932757 1.0e−05	0.166631714378227 1.0e−05	0.999999999999488
萤火虫算法		0.0329613019945 1.0e−02	0.2244373128369 1.0e−02	0.999999142357647
原鱼群算法	100	−0.486654927248361 1.0e−03	0.513427349291676 1.0e−03	0.999999916593226
本文算法		0.205319117653655 1.0e−07	−0.161288516988170 1.0e−07	1.000000000000000
萤火虫算法		0.1601496470543 1.0e−02	0.1316552108483 1.0e−02	0.999999283650137
原鱼群算法	200	−0.116310732311997 1.0e−03	−0.116310732311997 1.0e−03	0.999999994233091
本文算法		−0.170094505303123 1.0e−07	0.107743932689767 1.0e−07	1.000000000000000
萤火虫算法		0.031069301484802 1.0e−03	−0.534587146167033 1.0e−03	0.999999952208548
原鱼群算法	300	−0.914533657868921 1.0e−04	0.745694489536902 1.0e−04	0.999999997679280
本文算法		0.377444430208452 1.0e−08	0.004539805276508 1.0e−08	1.000000000000000

图 4－图 9 分别为本文算法、原鱼群算法和萤火虫算法在迭代 50、100 次后的人工鱼或萤火虫的分布情况。图 4 和图 5 表明，本文算法在 50 和 100 次迭代后 50 条的人工鱼均已聚集在全局最优值点附近，跳出了局部最优，且 100 次迭代后比 50 次迭代更加靠近全局最优值点（图 4 坐标精确到 10^{-6}，图 5 坐标精确到 10^{-8}）。图 6 和图 7 表明原鱼群算法中人工鱼分布在数个局部最优值点附近，100 次迭代后相对 50 次局部最优值点有所减少，但不易跳出局部最优。图 8 和图 9 表明萤火虫在 100 次迭代后比 50 次有所聚集，但萤火虫仍然闪乱分布在解空间中。图 4 和图 6、图 8，图 5 和图 7、图 9 之间的比较表明，本文算法比原鱼群算法和萤火虫算法能更加快速跳出局部最优达到全局最优，收敛速度和精度（由坐标精确度知）均明显优于原鱼群算法和萤火虫算法。

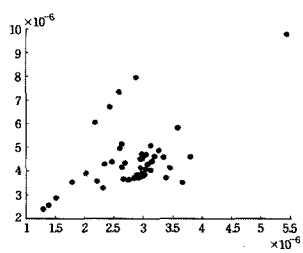


图 4　本文算法 50 次迭代后

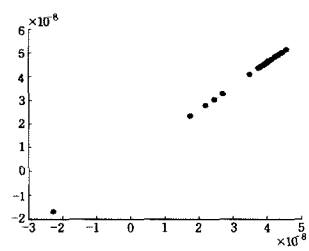


图 5　本文算法 100 次迭代后

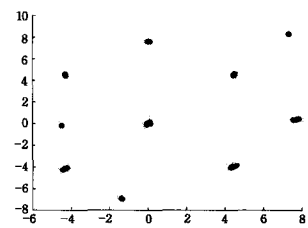


图 6　原鱼群算法 50 次迭代后

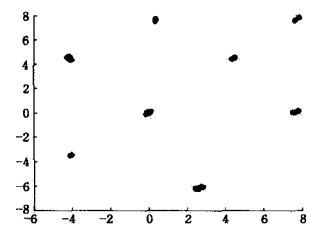


图 7　原鱼群算法 100 次迭代后

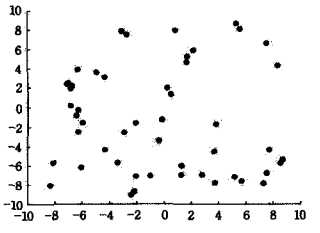


图 8 萤火虫算法 50 次迭代后

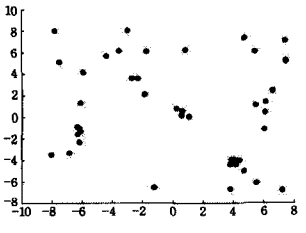


图 9 萤火虫算法 100 次迭代后

(2)采用 f_2,f_3,f_4 将本文算法与文献[23]比较。

重新设置参数,与文献[23]相同,种群规模 $N=40$,人工鱼感知范围 $Visual=3.5$,尝试次数 $try\_number=10$,常系数 $C=200$,拥挤因子 $\delta=1$,概率因子 $\alpha=1$,最大迭代次数 $L=1000$。函数 f_2,f_3,f_4 为文献[23]中典型的测试函数,适用于测试算法的性能。由表 3 可看出,对于测试函数 f_2,其在(0,0)取得最小值 0,通过本文算法寻优所得到的 x_1,x_2 值及最小值明显优于文献[22]和文献[23]所得结果;对于测试函数 f_3,min $f_3=-1.03162845348988$,本文算法所得结果比文献[22]和文献[23]更加接近函数 f_3 的最小值;对于测试函数 f_4,min $f_4=0$,本文算法所得结果可以直接优化到 0,明显优于文献[22]和文献[23]所优化的结果。因此,通过表 3 表明,本文算法在结果精度方面优于文献[23]。

表 3 本文算法与文献[23]性能对比

函数	算法	x_1 取值	x_2 取值	函数最优解
f_2	文献[22]	—	—	4.1429719958297 1.0e−08
	文献[23]	0.056908746651 1.0e−05	−0.304614922991 1.0e−05	9.6028856754391 1.0e−12
	本文算法	0.215741467807036 1.0e−07	−0.024164429363780 1.0e−07	4.712830057801123 1.0e−16
f_3	文献[22]	—	—	−1.03162845340261
	文献[23]	0.08977028425282	−0.71264366491184	−1.03162843301603
	本文算法	−0.089841976960426	0.712656446800480	−1.031628453489855
f_4	文献[22]	—	—	3.14955995514226 1.0e−15
	文献[23]	0.79702554479856	−0.36375290147197	5.195880464128456 1.0e−16
	本文算法	−0.382957335543697	0.214841113357172	0

(3)采用 f_5,f_6,f_7,f_8 将本文算法与文献[24]比较。

再次设置参数,与文献[24]相同,种群规模 $N=20$,人工鱼感知范围 $Visual=2.5$,尝试次数 $try\_number=50$,常系数 $C=200$,概率因子 $\alpha=1$,最大迭代次数 $L=200$,函数 f_5,f_6,f_7,f_8 为文献[24]中典型的测试函数,均为求最小值问题。

从表 4 可看出,对于测试函数 f_5,f_6,f_7,f_8,通过 20 次独立重复实验,本文算法对测试函数的最大值、最小值和平均值明显优于萤火虫算法[21],尽管测试函数的最小值 f_7,f_8 最小值比文献[24]差了一点,但总体的平均值及稳定性均优于文献[24]。总体来说,本文算法略优于文献[24]。

表 4 本文算法与文献[24]的性能对比

函数	算法	最大值	最小值	平均值
f_5	萤火虫算法	1.837903576227591 1.0e+01	5.010932209468918 1.0e−05	6.374158785505032 1.0e+00
	文献[24]	9.488197339106492 1.0e+00	3.973504078093357 1.0e−15	1.897639467821368 1.0e+00
	本文算法	9.488197339106259 1.0e+00	2.278948208429388 1.0e−20	0.948832943837911 1.0e+00
f_6	萤火虫算法	1.367237036655822 1.0e+01	3.069990664029084 1.0e−06	6.935818211719148 1.0e−01
	文献[24]	1.564780291305616 1.0e−09	2.528773728093723 1.0e−15	9.012636291214016 1.0e−11
	本文算法	3.081211650603261 1.0e−12	4.347013263280433 1.0e−18	8.333341312833874 1.0e−13
f_7	萤火虫算法	−2.154627967292517 1.0e−01	−1.031628206306398 1.0e+00	−9.908010040830296 1.0e−01
	文献[24]	−1.031628453487101 1.0e+00	−1.031628453489877 1.0e+00	−1.031628453489614 1.0e+00
	本文算法	−1.031628453489364 1.0e+00	−1.031628453489876 1.0e+00	−1.0316284539740 1.0e+00
f_8	萤火虫算法	4.761320209970957 1.0e+00	3.978874846424088 1.0e−01	1.181430751192128 1.0e+00
	文献[24]	3.978873577297399 1.0e−01	3.978873577297382 1.0e−01	3.978873577297389 1.0e−01
	本文算法	3.978873577297440 1.0e−01	3.978873577297380 1.0e−01	3.978873577297405 1.0e−01

(4)采用 f_9,f_{10} 将本文算法与文献[9]比较。

设置参数与文献[9]相同,种群规模 $N=50$,人工鱼感知范围 $Visual=50$,尝试次数 $try\_number=5$,常系数 $C=200$,概率因子 $\alpha=0.98$,最大迭代次数 $L=100$。为了测试本文算

法的有效性，将其与基本鱼群算法（AFSA）、全局鱼群算法（GAFSA）、基本粒子群算法（PSO）和文献[9]进行比较。函数 f_9，f_{10} 为文献[9]中典型的测试函数，均为求最小值问题。

从表 5 可看出，对于测试函数 f_9，f_{10}，通过 20 次独立重复实验，本文算法对测试函数的最优值、最差值、平均值和标准差明显优于 PSO 算法、AFSA 算法、GAFSA 算法，测试函数的最优值、最差值、平均值略优于文献[9]，且标准差也优于文献[9]，表明了本文算法在结果精度、稳定性方面优于文献[9]。

表 5　本文算法与文献[9]的性能对比

函数	算法	最优值	最差值	平均值	标准差
f_9	PSO 算法	−837.9657745448676	−719.5274387308298	−814.2781075619998	47.375333965735159
	AFSA 算法	−837.9656470364499	−620.8245612288606	−815.0078936353132	57.938651700294685
	GAFSA 算法	−837.9657705127778	−719.5263099819072	−832.0416423731224	25.812788731274459
	文献[9]	−837.9657727602282	−837.9651112498286	−837.9656812340666	1.665067860347421 1.0e−4
	本文算法	−837.9657745448676	−837.9654116187867	−837.9657077452153	1.192286769018275 1.0e−4
f_{10}	PSO 算法	−186.7309088310240	−186.4309088310239	−186.7009588310239	0.078057975249172
	AFSA 算法	−186.7296511547145	−167.0416947405776	−185.0028558070632	4.345896495868516
	GAFSA 算法	−186.7307859325576	−185.2559782688145	−186.5655756907131	0.380469483027250
	文献[9]	−186.7309062347991	−186.7241916812254	−186.7288496510064	−0.001983979381105
	本文算法	−186.7309088310240	−186.7302308575590	−186.7308162947341	2.061151529741477 1.0e−4

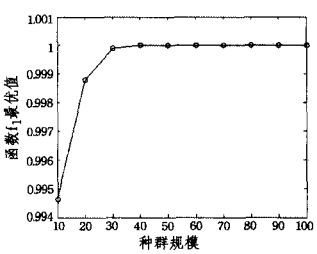
5.4　参数分析

由于篇幅限制，本节的参数分析均以函数 f_1 为例，最大迭代次数为 20。

本文对参数视野范围 *Visual* 与最大迭代次数 L 进行分析时建议，视野范围 $Visual=2.5$ 时聚类能力最强，效果最好，具体分析见文献[25]；最大迭代次数 $L=100$ 时，函数 f_1 最优值即可达到良好效果，见本文 5.3 节(1)。

对于拥挤度因子 δ，δ 越大，人工鱼跳出局部最优能力越强，但收敛的速度会减慢。由于人工鱼在逼近局部最优值时，会因避免过分拥挤而随机游动，因此不能精确逼近局部最优值点，影响算法的收敛速度及精确度，故对于某些局部极值不是很严重的具体问题，可以忽略拥挤因素，这样不仅简化了算法，也加快了算法的收敛速度且提高了结果的精度；对于局部最优较严重的问题，可以设置拥挤度因子，建议设置为 $\delta=0.618$（见文献[2]）。

由图 10 可知，算法性能与种群规模成正相关，当种群规模到达 40 时，此次测试性能达到平稳，性能并没有明显提高，因此不宜一味增大种群规模，本测试种群规模设置为 40～50 为宜，若搜索区域较大时可增大种群规模。本文取种群规模 $N=50$。



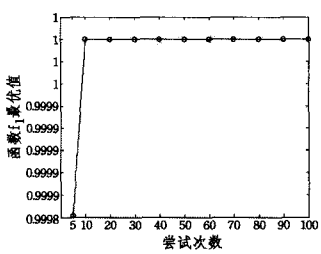
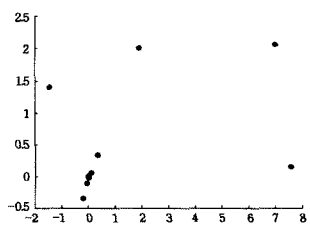


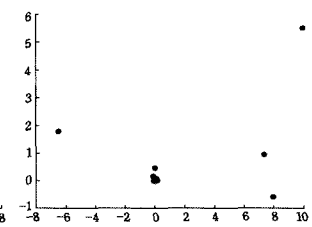
图 10　种群规模对算法性能影响　图 11　尝试次数对算法性能影响

由图 11 可知，随着尝试次数的增加，算法性能有所提高，但到达一定次数之后，并不能大幅提高算法性能，其几乎趋于稳定。若尝试次数太多，则会影响算法运行时间空间资源；若尝试次数太少，则会影响算法搜索效率。因此，尝试次数选择要适宜，本文建议 $try\_number=50$。

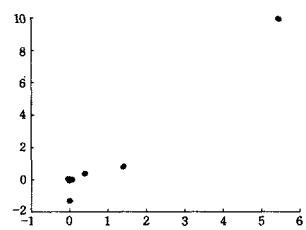
由图 12 可知，随着概率因子的增大，人工鱼的分布就越集中，因为概率因子越小，人工鱼跳出局部最优的能力就越强，分布就越散乱，影响算法收敛速度、精度；概率因子越大，人工鱼跳出局部最优的能力就越弱，分布就越集中，易陷入局部最优。通过图 12(a)－(e)的比较可得，概率因子 $\alpha=0.9$ 较为合适。



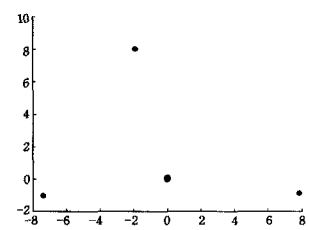
(a) 概率因子 $\alpha=0.8$ 时的种群分布



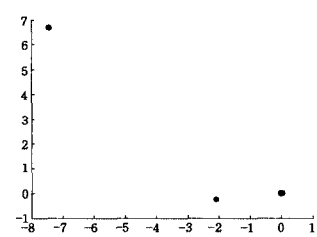
(b) 概率因子 $\alpha=0.85$ 时的种群分布



(c) 概率因子 $\alpha=0.9$ 时的种群分布



(d) 概率因子 $\alpha=0.95$ 时的种群分布



(e) 概率因子 $\alpha=0.99$ 时的种群分布

图 12

结束语　本文针对人工鱼群算法目前存在的问题及不足，提出了变步长自适应的改进人工鱼群算法，主要通过改进其觅食行为和搜索过程中的移动步长来改进人工鱼群算法。证明了其全局收敛性，增加了算法的理论基础，并分析其时间复杂度。最后通过仿真实验表明，本文算法在跳出局部最优的能力、全局收敛速度、优化精度和稳定性等方面的性能得到了显著的提高。

（下转第 246 页）

Science and Engineering,2012,9(4):701-709

[5] Huang Zhou, Zhang Jiang. Generating SAT Instances from First-Order Formulas[J]. Journal of Software,2005,16(3):327-335

[6] CHEN Yuan,SHI Zhong-Zhi. The Comparison and Analysis of Solving Approachs to Constrain Satisfaction Problems[J]. Computer Science,1998,25(1):8-12

[7] Wang Teng-fei,Xu Xhou-bo,Gu Tian-long. Symbolic ADD Algorithms for Arc Consistency and Application in Constraint Satisfaction Problem Solving[J]. Computer Science,2013,40(12):243-247

[8] Li Hong-bo, Li Zhan-shan, Wang Tao. Improving Coarse-Grained Arc Consistency Algorithms in Solving Constraint Satisfaction Problems[J]. Journal of Software,2012,23(7):1816-1823

[9] Zhu Hai-bin. Role Mechanisms in Collaborative Systems [J]. International Journal of Production Research, 2006, 44(1): 181-193

[10] Zhu Hai-bin,Zhou Meng-chu. Efficient Role Transfer Based on Kuhn-Munkres Algorithm[J]. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 2012, 42(2):491-496

[11] Zhu Hai-bin, Zhou Meng-chu, Alknis R. Group Role Assignment via a Kuhn-Munkres Algorithm-based Solution[J]. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans,2012,42(3):739-750

[12] Huang Da-feng. The construction and equivalence analysis of the solutions to the N-Queens Problem[D]. Guangzhou: Graduate School of Sun Yat-Sen University,2008

[13] Wang Zi-wen, Li Zhan-shan, Ai Yang, et al. Algorithm for solving constraint satisfaction problems based on dynamic value ordering heuristic[J]. Computer Integrated Manufacturing Systems,2011,17(4):832-837

[14] Abramson B, Yung M. Construction Through Decomposition: A Divide-and-Conquer Algorithm for the N-Queens Problem[C]// Proceedings of the 1986 Fall Joint Computer Conference. 1986: 620-628

[15] Sun Ji-gui, Zhu Xing-jun, Zhang Yong-gang, et al. An Approach of Solving Constraint Satisfaction Problem Based on Preprocessing[J]. Chinese Journal of Computers,2008,31(6):919-926

[16] Zhu Hai-bin. Fundamental Issues in the Design of a Role Engine [C]// Proceedings of the 6th International Symposium on Collaborative Technologies and Systems. Irvine, CA, USA, 2008: 399-407

(上接第 216 页)

参 考 文 献

[1] 李晓磊,钱积新. 人工鱼群算法:自下而上的寻优模式[C]// 过程系统工程会论文集. 2001:76-82

[2] 李晓磊,邵之江,钱积新. 一种基于动物自治体的寻优模式:鱼群算法[J]. 系统工程理论与实践,2002,22(11):32-38

[3] 王兴伟,秦培玉,黄敏. 基于人工鱼群的 ABC 支持型 QoS 单播路由机制[J]. 计算机学报,2010,33(4):718-725

[4] 孙伟,朱正礼,郑磊. 基于人工鱼群和微粒群混合算法的 WSN 节点部署策略[J]. 计算机科学,2012,39(11):83-85

[5] 刘艳林,马苗,刘艳丽,等. 基于改进人工鱼群算法的含噪图像分割方法[J]. 计算机工程与应用,2013,49(20):157-160

[6] Xiao J M, Zheng X M, Wang X H, et al. A Mdified Artificial Fish-Swam Algorithm[C]// Proceedings of the 6th World Congress on Intelligent Control and Automation. Dalian,2006:3456-3460

[7] 黄华娟,周永权. 改进型人工鱼群算法及复杂函数全局优化方法[J]. 广西师范大学学报:自然科学版,2008,26(1):194-197

[8] Si He, Nabil Belacel, Habib Hamam, et al. Fuzzy Clustering with Improved Artificial Fish Swarm Algorithm[C]// International Joint Conference on Computational Sciences and Optimization 2009. Hainan,2009(2):317-321

[9] 张大斌,杨添柔,温梅,等. 基于差分进化的鱼群算法及其函数优化应用[J]. 计算机工程,2013,39(5):18-27

[10] 张凤梅,邵波. 多峰函数优化的生境人工鱼群算法[J]. 控制理论与应用,2008,25(4):773-776

[11] 王培崇,雷凤君,钱旭. 改进人工鱼群算法及其收敛性分析[J]. 科学技术与工程,2013,13(3):616-620

[12] 李咏梅,周永权,姚祥光. 基于追尾行为的改进型人工萤火虫群算法[J]. 计算机科学,2011,38(3):248-251

[13] 刘薇,刘柏嵩,王洋洋. 基于改进鱼群和 K-means 的混合聚类算法[J]. 计算机工程与应用,2013,49(22):119-122

[14] 段其昌,唐若笠,徐宏英,等. 粒子群优化鱼群算法仿真分析[J]. 控制与决策,2013,28(9):1436-1440

[15] 陶杨,韩维,张磊. 基于群体行为的自适应变异算子鱼群算法[J]. 中国电子科学研究院学报,2013,8(5):491-495

[16] 刘彦君,江铭炎. 自适应视野和步长的改进人工鱼群算法[J]. 计算机工程与应用,2009,45(25):35-37,47

[17] 欧阳喆,周永权. 自适应步长萤火虫优化算法[J]. 计算机应用,2011,31(7):1804-1807

[18] Zhang Zhong-hai. Research of self-adaptive step-changed stochastic resonance using particle swarm optimization [J]. Journal of vibration and shock,2013,32(19):125-130

[19] 马宪民,刘妮. 自适应视野的人工鱼群算法求解最短路径问题[J]. 通讯学报,2014,35(1):1-6

[20] Manber U. Introduction to Algorithms: A Creative Approach [M]. Milano, Italy: Addison-Wesley, 1989

[21] Krishnanand K N, Ghose D. Glowworm swarm optimisation: a new method for Optimising motilmodal functions[J]. International Journal of Computational Intelligence Studies, 2009, 1(1):93-119

[22] Qu L, He D. Novel Artificial Fish-school Algorithm Based on Chaos Search [J]. Computer Engineering and Applications, 2010,46(22):40-42

[23] Jiang Jing-qing, Bo Yu-ling, Song Chu-yi, et al. Hybrid Algorithm Based on Particle Swarm Optimization and Artificial Fish Swarm Algorithm [J]. Lecture Notes in Computer Science, 2012,7367:607-614

[24] 张军丽,周永权. 一种用 Powell 方法局部优化的人工萤火虫算法[J]. 模式识别与人工智能,2011,24(5):680-684

[25] 吴月萍,杜奕. 改进的人工鱼群算法的参数分析[J]. 计算机工程与应用,2012,48(13):48-52