

文章编号: 1001-0920(2011)12-1840-06

## 模拟生物理想自由分布模型的粒子群算法

秦全德, 李荣钧

(华南理工大学工商管理学院, 广州 510640)

**摘要:** 在分析生物觅食行为中资源斑块选择理想自由分布模型的基础上, 提出一种新型的粒子群算法——理想自由分布粒子群优化算法(IFDPSO). 该算法将所有粒子中3个不重叠的个体最优位置的适应度视为资源斑块的食物质量, 根据理想自由分布模型随机分配相应数量的粒子到各资源斑块中. 为保证群体的多样性, 各资源斑块的群体最优位置保持随迭代次数增加而线性递减的距离. 在间隔一定的迭代次数后, 将各资源斑块的粒子重新组合. 标准测试函数的仿真结果表明了IFDPSO算法的有效性.

**关键词:** 粒子群算法; 理想自由分布模型; 多峰函数

**中图分类号:** TP18

**文献标识码:** A

## Particle swarm optimization algorithm mimicking biological ideal free distribution model

QIN Quan-de, LI Rong-jun

(School of Business Administration, South China University of Technology, Guangzhou 510640, China. Correspondent: QIN Quan-de, E-mail: qinquande@gmail.com)

**Abstract:** A novel particle swarm optimization(PSO) algorithm, ideal free distribution(IFD) PSO, is proposed based on the analysis of IFD model, in which, three non-overlapping personal best positions of the particles are selected, and their fitness values are regarded as food quality of resource patch. Particles are randomly assigned to each resource patch according to ideal free distribution model. Particles in each sub-population search the optima independently in accordance with standard PSO algorithm. In order to guarantee the diversity of the whole population, the best position of each sub-population is set to keep a distance, which linearly decreases with iterations. After a certain number of iterations, all sub-population particles are regrouped. The experimental results of benchmark functions show the effectiveness of IFDPSO algorithm.

**Key words:** particle swarm optimization algorithm; ideal free distribution model; multimodal functions

### 1 引言

粒子群优化(PSO)算法是在模拟鸟群觅食行为的规律上而提出的一种基于种群搜索的全局优化算法<sup>[1]</sup>. 由于该算法概念简单、需要调整的参数少、收敛速度快, 且易于计算机编程实现<sup>[2]</sup>, 目前已将其成功应用于函数优化、神经网络训练、模糊系统控制和金融优化等众多领域<sup>[3-5]</sup>. 但是, PSO算法同其他的随机搜索方法类似, 在优化复杂多维函数时, 易陷入局部最优, 出现早熟收敛现象<sup>[6]</sup>. 针对这一问题, 较多的研究者给出了许多改进方法, 可以简单归纳为以下几类: 1) 算法的参数调节. Shi等人<sup>[3]</sup>对PSO算法的速度项引入了惯性权重, 并提出在迭代过程中动态调整惯性权重以平衡算法的收敛速度和全局寻优; Clerc等

人<sup>[7]</sup>给出了带压缩因子的粒子群算法(CPSO); 张顶学等人<sup>[8]</sup>根据群体中平均粒子相似程度的概念, 提出了动态改变惯性权重的自适应粒子群算法. 2) 群体拓扑结构的改进. Suganthan<sup>[9]</sup>根据粒子之间的距离, 通过领域算子动态调整领域结构; 倪庆剑等人<sup>[10]</sup>提出了动态可变多簇拓扑结构的概率粒子群算法. 3) 融入其他算法的算子. Angeline<sup>[11]</sup>将进化计算中的选择操作算子嵌入到粒子群算法中; Zhang等人<sup>[12]</sup>提出了融合差分进化算子的粒子群算法(DEPSO). 4) 新的学习策略. Liang等人<sup>[13]</sup>提出了综合学习粒子群算法(CLPSO), 每个粒子的速度更新基于所有其他粒子的历史个体最优位置. 5) 融入其他生物行为机制. He等人<sup>[14]</sup>根据动物群中的被动聚众现象提出了一种新型粒子群优

收稿日期: 2010-07-30; 修回日期: 2010-11-04.

基金项目: 国家自然科学基金项目(71071057); 高等院校博士学科点专项基金项目(20060561002).

作者简介: 秦全德(1979—), 男, 博士生, 从事智能计算的研究; 李荣钧(1946—), 男, 教授, 博士生导师, 从事智能计算、系统优化等研究.

化算法 (PSOPC). PSOPC 算法的速度更新不仅考虑了个体的经验和邻居中最好个体的经验, 同时还考虑了群体中其他同伴的经验.

PSO 算法是源于对鸟群群集行为的模拟, 群体之间按照一种合作的方式觅食. 因此, 期望将自然界中的一些生物行为机制融入到 PSO 算法中, 从而改进算法的性能. 一些学者对此开展了相关研究<sup>[14-15]</sup>, 但这方面的研究还不够深入. 初始的粒子群算法中, 每个粒子根据自身的个体最优位置和整个种群迄今找到的群体最优位置调整飞行方向和速度. 如果将群体最优位置视为食物资源最丰富的斑块, 则该种觅食方式显然不符合“资源斑块(食物源)选择”模式和最优觅食理论的“边际值原理 (MVT)”<sup>[16]</sup>. 在生物的觅食策略中, 资源斑块的选择至关重要, 觅食行为发生前觅食者必须首先决定觅食的地点. 理想自由分布 (IFD) 模型描述了生物群体在觅食行为中, 生物个体数量在不同资源斑块中的分布格局<sup>[17]</sup>.

本文在分析生物觅食行为中资源斑块选择 IFD 模型的基础上, 提出了更符合现实的新型粒子群算法——IFDPSO. IFDPSO 算法在搜索过程中, 能够较好地保持群体的多样性, 增强全局寻优能力. 标准测试函数的实验结果表明了 IFDPSO 的有效性, 更能凸显多峰函数的优化性.

## 2 理想自由分布模型

在生态学中, IFD 模型是描述觅食者在资源斑块中的分配方式. 根据资源斑块中食物的丰富程度, 觅食者按照一定的比例在各个资源斑块中聚集. “理想”是指所有的觅食者能够对资源斑块中食物的质量进行合理评价, “自由”表示觅食者能够无阻碍地在各个资源斑块移动. IFD 模型假设资源斑块中每个觅食者的觅食策略遵循最优觅食原理, 且觅食者密度的增加将会使觅食者的收益按比例减少<sup>[17]</sup>. 随着进入同一资源斑块内觅食者数量的增多, 竞争强度的增加和收益的下降将会促使部分觅食者迁移到其他的资源斑块. 最终, 在每一个资源斑块里觅食者的数量会形成一种均衡, 即每个觅食者在任何一个资源斑块都只能获得同样的收益. 因此, 在食物质量丰富的资源斑块里分布的觅食者数量多; 反之, 觅食者数量较少.

为了使 IFD 模型更具有生态学意义, Sutherland 将觅食者处理食物的时间和觅食者个体间的相互干扰因素纳入到 IFD 模型中<sup>[18]</sup>. 他指出在资源斑块上分布的觅食者的比例  $p_i$ , 与该资源斑块上食物资源的比例  $h_i$  相关, 其关系式为

$$p_i = \alpha(h_i^{\frac{1}{m}}). \quad (1)$$

其中:  $m$  为觅食者间的干扰系数;  $\alpha$  为一个用来修正  $p_i$  并使之不超过 1 的常数, 本文取值为 1. 干扰系数  $m$

对  $p_i$  和  $h_i$  之间关系的影响如图 1 所示. 当  $m = 1$  时, 表示无任何干扰, 觅食者比例与食物比例呈线性关系, 即每个资源斑块中觅食者比例与食物的比例基本相同. 在大多数情况下,  $0 < m < 1$ , 觅食者间存在一定的干扰, 高质量资源斑块里会吸引相对更多的觅食者, 从而形成正密度相关. 当  $m > 1$  时, 食物质量较低的资源斑块会吸引相对较多的觅食者.

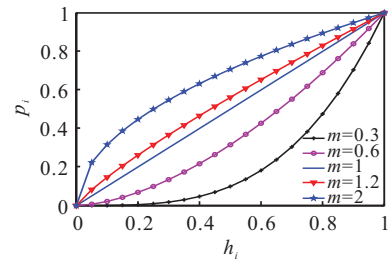


图 1 干扰系数  $m$  对  $h_i$  和  $p_i$  之间关系的影响

## 3 IFDPSO 算法

### 3.1 粒子群算法基本框架

PSO 算法的基本思想是将每个粒子视为问题的一个可行解, 多个粒子共存合作寻优, 粒子的好坏由一个事先设定的适应度函数确定. 粒子  $i$  在  $t$  次迭代时的状态属性由两个向量描述: 位置向量  $x_i^t = [x_{i1}^t, x_{i2}^t, \dots, x_{id}^t]$  和速度向量  $v_i^t = [v_{i1}^t, v_{i2}^t, \dots, v_{id}^t]$ . 其中:  $x_{id}^t \in (L_d, U_d)$ ,  $L_d, U_d$  分别表示搜索空间的下限和上限,  $d$  表示搜索空间的维数;  $v_{id}^t \in [V_{\min}, V_{\max}]$ ,  $V_{\min}, V_{\max}$  分别代表粒子飞行的最小和最大速度. 在算法的迭代过程中, 每一个粒子的速度和位置将按照下式更新:

$$v_{id}^{t+1} = \omega v_{id}^t + c_1 \text{rand}_1(\text{pbest}_{id}^t - x_{id}^t) + c_2 \text{rand}_2(\text{gbest}_d^t - x_{id}^t), \quad (2)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}. \quad (3)$$

其中:  $\text{pbest}_i^t$  表示  $t$  次迭代时迭代粒子  $i$  的最好个体位置, 记为  $\text{pbest}_i^t = [\text{pbest}_{i1}^t, \text{pbest}_{i2}^t, \dots, \text{pbest}_{id}^t]$ ;  $\text{gbest}^t$  是指到  $t$  次迭代为止群体的最佳位置,  $\text{gbest}^t = [\text{gbest}_1^t, \text{gbest}_2^t, \dots, \text{gbest}_d^t]$ ;  $\text{rand}_1, \text{rand}_2$  为均匀分布在  $[0,1]$  区间的随机数;  $c_1, c_2$  为正的加速系数;  $\omega$  称为惯性权重. 一般设定  $\omega$  按下式线性递减<sup>[19]</sup>:

$$\omega = (\omega_{\text{start}} - \omega_{\text{end}}) \times \left( \frac{\text{iter}_{\text{max}} - \text{iter}}{\text{iter}_{\text{max}}} \right) + \omega_{\text{end}}, \quad (4)$$

其中  $\text{iter}_{\text{max}}$  和  $\text{iter}$  分别表示最大迭代次数和当前的迭代次数. Shi 等人经过多组实验, 建议采用  $\omega_{\text{start}} = 0.9$ ,  $\omega_{\text{end}} = 0.4$ . 本文将这种惯性权重递减的 PSO 算法称为标准粒子群算法 (SPSO).

### 3.2 IFDPSO 基本思想

在 IFDPSO 算法中, 首先按照标准粒子群算法迭代  $\eta$  次, 此时由于惯性权重较大, 群体的粒子进行全局探索; 此后, 对群体中粒子的个体最优位置按照适

应度大小进行排序. 本文设定粒子的适应度越大越好, 因此在对函数最小化的优化中取函数值的倒数作为适应度值. 在 IFDPSO 算法中, 将粒子的 pbest 个体最优位置的适应度视为资源斑块的食物质量, 显然, 资源斑块的数量将会直接影响 IFDPSO 的性能. 如果资源斑块的数量过多, 分布到每个资源斑块的粒子数便会较少, 从而不利于粒子之间的协作, 影响算法的收敛速度. 反之, 如果资源斑块的数量过小, 算法的多样性得不到保证, 则容易陷入局部最优. 综合考虑计算时间和寻优搜索的需要, 本文设定资源斑块的数量为 3 个. 在现实的生物觅食行为中, 资源斑块常常相隔一定的距离. 在 IFDPSO 算法中资源斑块之间的距离较大时, 粒子分布的空间比较分散, 这有利于全局探索; 反之, 可以提高局部搜索能力. 本文设定资源斑块之间的距离随着迭代次数的增加而线性递减, 其中各资源斑块采用如下的方法确定.

Step 1: 为每个粒子的 pbest 设定半径为  $r$  的一个区域,  $r$  的大小随着迭代次数的增加从搜索范围的 5% 线性递减到 0.

Step 2: 将所有粒子 pbest 的适应度进行排序.

Step 3: 首先确定适应度最佳的 pbest 作为资源斑块 1, 定义为基本 gbest, 如图 2 中 A1.

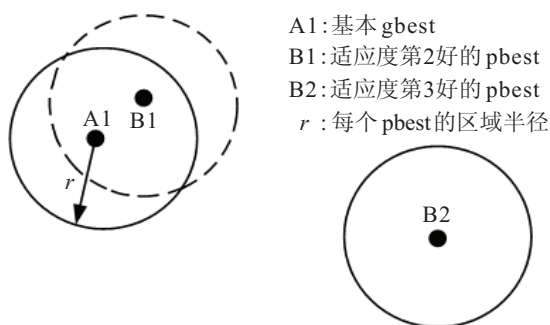


图 2 资源斑块的确定

Step 4: 选择适应度第 2 好的 pbest 暂时作为资源斑块 2, 如果其与基本 gbest 的区域不重叠, 则将其作为资源斑块 2; 反之, 将其舍弃, 选择适应度第 3 好的 pbest 与基本 gbest 的区域进行比较. 重复以上过程, 确定资源斑块 2. 如图 2 中适应度第 2 好的 B1 位于基本 gbest 的区域内, 适应度第 3 好的 B2 将作为资源斑块 2.

Step 5: 依照 Step 4, 确定资源斑块 3, 其不能与资源斑块 1 和资源斑块 2 的区域重叠.

在确定 3 个资源斑块的基础上, 根据各个资源斑块的食物(适应度)占 3 个斑块总食物的比例  $h_i (i = 1, 2, 3)$ , 按式(1)计算得到分配到各个资源斑块粒子数量的比例  $p_i (i = 1, 2, 3)$ , 利用归一化的方法得到  $p_i^m (i = 1, 2, 3)$ , 其中  $p_i^m = p_i / \sum_{i=1}^3 p_i (i = 1, 2, 3)$ .

为了使每个资源斑块分配的粒子为整数, 对资源斑块 1 和资源斑块 2 分配的粒子数量分别取  $\lfloor p_1^m n \rfloor$  和  $\lfloor p_2^m n \rfloor$ , 其余的粒子都分配到资源斑块 3, 其中  $n$  表示群体规模, 函数  $\lfloor x \rfloor$  表示不大于  $x$  的整数. 然后, 随机选择相应数量的粒子分配到各资源斑块. 式(1)中的干扰系数  $m$  随迭代次数的增加从 1.2 线性递减到 0.8, 这是为了在搜索前期各个资源斑块的粒子数量相差不要太大, 从而有利于全局搜索; 而在后期让食物质量好的资源斑块汇聚更多的粒子进行局部搜索, 以便找到更好的解.

分配到各个资源斑块的粒子按照标准粒子群算法进行寻优搜索. 为保证粒子群体的多样性, 避免各资源斑块粒子的 gbest 重叠, 采用了与确定资源斑块类似的方式更新各个资源斑块的 gbest. 每个资源斑块的 gbest 同样设定一个半径为  $r$  的区域, 不同的是对各个资源斑块粒子群体 pbest 的适应度进行单独排序. 当一个资源斑块的 gbest 位于已经更新的资源斑块的区域内, 则将该资源斑块内的第 2 好的 pbest 作为暂时的 gbest, 再与已经确定的 gbest 区域进行比较判断, 直至更新好所有资源斑块粒子的 gbest. 间隔  $k$  次迭代后, 各资源斑块的粒子重新组合, 再次按照食物质量确定新的资源斑块并随机分配粒子进行搜索. 在算法搜索过程中, 如果连续 30 次迭代 3 个资源斑块中最佳的 gbest 没有发生变化, 则在最佳 gbest 所在的资源斑块中随机选择一个粒子在任一维上的速度按下式发生变异:

$$v_{id}^t = \begin{cases} 0.5 \times V_{\max} \times \text{rand}_3, & \text{rand}_4 \leq 0.5; \\ -0.5 \times V_{\max} \times \text{rand}_3, & \text{反之.} \end{cases} \quad (5)$$

其中:  $V_{\max}$  表示粒子飞行的最大允许速度,  $\text{rand}_3$  和  $\text{rand}_4$  表示均匀分布在  $[0, 1]$  之间的随机数. 发生变异时, 从统计意义上而言, 最佳 gbest 所在的资源斑块中每一个粒子和其每一维速度是以同样的概率被选择发生变异. 因为仅仅只有一个粒子中的一维速度发生了变异, 所以几乎不会破坏群体的结构.

### 3.3 IFDPSO 算法步骤

IFDPSO 的算法步骤可描述为:

Step 1: 初始化粒子群体的位置和速度, 设定相应的参数.

Step 2: 按照标准的粒子群算法运行  $\eta$  次.

Step 3: 确定 3 个资源斑块, 按照理想自由分布模型确定分配到各资源斑块的粒子数量, 并随机选择相应数量的粒子到各资源斑块.

Step 4: 3 个资源斑块的粒子分别按照标准粒子群算法寻优搜索, 更新后每个资源斑块粒子群体的 gbest 区域不能重叠.

Step 5: 每个资源斑块的粒子单独运行  $k$  次后重新组合.

Step 6: 判断是否发生变异, 如果满足条件, 则按式 (5) 进行变异操作; 否则, 转到 Step 7.

Step 7: 判断程序终止条件是否满足, 若满足则算法终止, 输出优化解; 否则, 转到 Step 3.

## 4 实验分析

### 4.1 测试函数

本文选取进化优化中 5 个常用的测试函数进行分析, 详细描述见表 1. 其中  $f_1$  和  $f_2$  是单峰函数,  $f_2 \sim f_5$  属于多峰函数. 在表 1 中给出了每个函数的表达式, 以及搜索和初始化范围.

表 1 测试函数及其搜索和初始化范围

函数名称	函数数学表达式	搜索范围	初始化范围
Schwefel's Problem 1.2	$f_1(x) = \sum_{i=1}^D \left( \sum_{j=1}^i x_j \right)^2$	$(-100, 100)^D$	$(50, 100)^D$
Rosenbrock	$f_2(x) = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i)^2 + (x_i - 1)^2)$	$(-30, 30)^D$	$(10, 30)^D$
Ackley	$f_3(x) = -20 \exp \left( -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left( \frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e$	$(-32, 32)^D$	$(10, 20)^D$
Rastrigin	$f_4(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$(-10, 10)^D$	$(2.56, 5.12)^D$
Griewank	$f_5(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \left( \frac{x_i - i}{\sqrt{i}} \right) + 1$	$(-600, 600)^D$	$(300, 600)^D$

### 4.2 实验环境设置

本文采用非对称的方法初始化, 即初始化范围的  $[X_{\min}, X_{\max}]$  取值为搜索范围的一部分. 在所有比较的算法中速度大小限制为  $V_{\max} = U_d, V_{\min} = L_d$ . PSOPC 使用的参数依据文献 [14],  $\omega_{\text{start}} = 0.9, \omega_{\text{end}} = 0.7, c_1 = c_2 = 0.5, c_3$  从 0.4 线性递增到 0.6. 在 IFDPSO 中, 经过实验仿真,  $\eta$  设定为  $(0.1 \times \text{iter}_{\max})$ . IFDPSO 同 SPSO 设置相同的  $c_1, c_2$  和  $\omega, c_1 = c_2 = 2, \omega$  以式 (4) 进行调整. CPSO 的参数设置参考文献 [7],  $c_1 = c_2 = 2.05, \lambda = 0.729$ . 测试函数的维数都设置为 30,  $D = 30$ . 进行比较的每种算法粒子的数量都设置为 80, 实际情况经常需要对变量的取值范围进行限制. 本文采用文献 [20] 的方法对搜索边界进行处理, 粒子  $i$  在更新位置时, 如果  $X_{id}^t > U_d$  或者  $X_{id}^t < L_d$ , 则  $V_{id}^t = 0, X_{id}^t = \min(\max(X_{id}^t, L_d), U_d)$ .

为了测试 IFDPSO 的性能对参数  $k$  值是否灵敏, 本文采用不同的  $k$  值对  $f_2$  和  $f_4$  的 30 维函数进行仿真实验. 实验设置的最大迭代次数设为 3 000 次, 每个实验独立运行 30 次得到的结果如表 2 所示.

从表 2 可以看出, 对于单峰函数而言, 当  $k = 200$

表 2 不同  $k$  值下  $f_2$  和  $f_4$  的测试结果

函数名称	不同 $k$ 值下求解结果的均值和标准差				
	20	50	100	150	200
$f_2$	41.5434	30.5819	21.3073	19.3390	12.5565
	(13.9023)	(11.2013)	(9.0437)	(7.4691)	(6.2013)
$f_4$	7.2280	14.0017	19.4072	21.9346	24.9524
	(2.7296)	(7.4789)	(6.8121)	(4.5284)	(5.5645)

时算法的性能较好; 当  $k = 20$  时多峰函数优化的性能较好. 在本文的仿真实验中, 优化单峰函数和多峰函数的  $k$  值分别设置为 200 和 20.

### 4.3 实验结果与分析

将比较的 SPSO, CPSO, PSOPC 和 IFDPSO 四种粒子群算法对 5 个标准测试函数分别独立运行 30 次, 最大迭代次数为 6 000 次, 得到的实验结果整理如表 3 所示. 在表 3 中列出了各算法对每个测试函数优化的平均值和标准差. 比较的 4 种算法中, 实验结果的最好值加粗表示. 图 3~图 5 描述了每种算法对  $f_1, f_2$  和  $f_4$  优化的过程中平均最优值(取 10 为底的对数)的变化曲线. 从表 3 和图 3~图 5 中可以看出, 与其他算法相比, IFDPSO 表现出了较好的综合性能, 对多峰函数的优化性能更加凸显, 不容易陷入局部最优. 在对单峰函数的优化过程中, IFDPSO 的性能明显优于 SPSO 和 PSOPC. 在对  $f_1$  的优化中, CPSO 的性能优于 IFDPSO.  $f_2$  是一个经典复杂优化问题, 其全局最优点位于一个平滑、狭长的抛物线山谷内, 算法很难辨别搜索方向, 找到全局最优点的机会很小. IFDPSO 对  $f_2$  的优化性能好于 CPSO. 从对多峰函数  $f_3, f_4$  和  $f_5$  的优化结果可以看出, IFDPSO 体现了很好的优化效果, 特别是对于具有大量局部最优点的复杂多峰函数  $f_4$ , PSOPB 不容易早熟, 有良好的持续优化能力. IFDPSO 在对  $f_2, f_3, f_4$  和  $f_5$  的优化中, 寻优的结果相比其他 3 种算法的标准差少, 表明 IFDPSO 具有较好的鲁棒性.

在 IFDPSO 中, 所有的粒子不是单一地向个体最



表3 各种算法的测试性能比较结果

函数名称	函数维数	最大迭代次数	指标	比较的 PSO 算法			
				SPSO	CPSO	PSOPC	IFDPSO
$f_1$	30	6000	平均值	0.5987	1.0991e-17	0.6683	0.0019
			标准差	0.4457	1.8367e-17	0.5335	0.0012
$f_2$	30	6000	平均值	30.7252	6.1971	23.5664	4.3933
			标准差	26.1358	3.8022	4.4037	2.2515
$f_3$	30	6000	平均值	6.2172e-015	0.4765	3.1086e-014	2.3242e-017
			标准差	5.4153e-015	0.7744	4.8643e-014	5.3572e-016
$f_4$	30	6000	平均值	19.8333	43.6796	27.6621	3.5984
			标准差	5.3515	11.4073	6.7261	2.3849
$f_5$	30	6000	平均值	0.0150	0.0118	0.0098	0.0049
			标准差	0.0221	0.0107	0.0071	0.0058

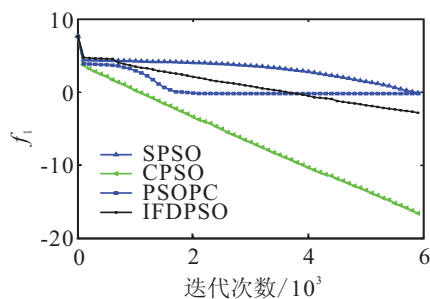
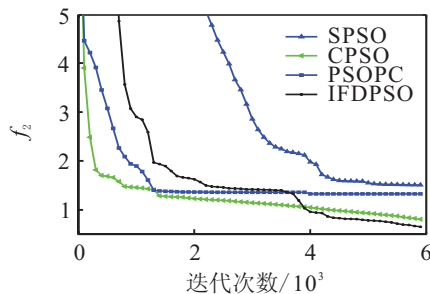
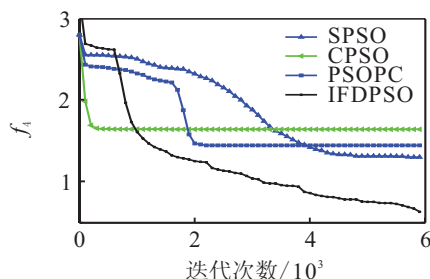
优位置和群体的最优位置所决定的方向飞行,而且各个资源斑块的粒子在间隔一定的迭代次数后重组再次随机分配到新确定的资源斑块中.这种机制保证了整个粒子群体的多样性,使其不容易陷入局部最优,特别适合于多峰函数的优化.根据“没有免费午餐定理”,在保证群体多样性的同时也导致了收敛速度不够快.从图3~图5可以看出,IFDPSO的收敛速度虽然比SPSO快,但比CPSO和PSOPC慢.

## 5 结论

将大自然的其他生物机制嵌入到PSO算法中是提高其性能的一条可行途径.本文提出了一种模拟生物觅食行为中资源斑块选择理想自由分布模型的新型粒子群算法——IFDPSO.在IFDPSO中将个体最优位置的适应度视为资源斑块的食物质量,在确定了3个不重叠的资源斑块后,按IFD模型随机分配数量的粒子到各个资源斑块.各个资源斑块的粒子依据标准粒子群算法搜索,间隔一定的迭代次数后各个斑块的粒子重组.实验分析表明,单峰函数间隔200代重组性能较佳;反之,对于多峰函数而言,间隔20代体现了良好的优化效果.对标准测试函数的实验表明,IFDPSO具有良好的综合优化性能和全局搜索能力.在IFDPSO中,参数的大小对算法性能的影响较大,在未来的研究中期望建立一个自适应调整值的机制. IFD模型是以最优化理论为基础,与自然界的真实情况有出入,许多新的理论是在IFD模型的基础上得以发展<sup>[21]</sup>.如何嵌入更符合现实的生物学机制到PSO算法中,以及应用IFDPSO到实际经济管理和工程优化中是未来研究的重点.

## 参考文献(References)

- [1] Eberchart R C, Kennedy J. Particle swarm optimization[C]. IEEE Int Conf on Neural Networks. Perth, 1995: 1942-1948.
- [2] 谢晓锋, 张文俊, 杨之廉. 微粒群算法综述[J]. 控制与决策, 2003, 18(2): 129-133.  
(Xie X F, Zhang W J, Yang Z L. Overview of particle swarm optimization[J]. Control and Decision, 2003, 18(2): 129-133.)
- [3] Shi Y, Eberchart R C. A modified particle swarm optimizer[C]. IEEE Congress on Evolutionary Computation. Anchorage, 1998: 69-73.
- [4] Mendes R, Cortez P, Rocha M, et al. Particle swarms for feed forward neural network training[C]. Int Joint Conf on Neural Networks. Seattle, 2002: 1895-1899.

图3 各种算法  $f_1$  的平均最优值变化图4 各种算法  $f_2$  的平均最优值变化图5 各种算法  $f_4$  的平均最优值变化

- [5] Cura T. Particle swarm optimization approach to portfolio optimization[J]. *Nonlinear Analysis: Real World Applications*, 2009, 10(4): 2396-2406.
- [6] Angeline P J. Evolutionary optimization versus particle swarm optimization and philosophy and performance difference[C]. *Proc of 7th Annual Conf on Evolutionary Programming*. San Diego, 1998: 601-610.
- [7] Clerc M, Kennedy J. The particle swarm: Explosion, stability, and convergence in multidimensional complex space[J]. *IEEE Trans on Evolutionary Computation*, 2002, 6(1): 58-73.
- [8] 张顶学, 关治洪, 刘新芝. 一种动态改变惯性权重的自适应粒子群算法[J]. *控制与决策*, 2008, 23(11): 1253-1257. (Zhang D X, Guan Z H, Liu X Z. Adaptive particle swarm optimization algorithm with dynamically changing inertia weight[J]. *Control and Decision*, 2008, 23(11): 1253-1257.)
- [9] Suganthan P N. Particle swarm optimizer with neighborhood operator[C]. *Proc of the IEEE Congress of Evolutionary Computation*. Washington DC, 1999: 1958-1961.
- [10] 倪庆剑, 张志政, 王蓁蓁, 等. 一种基于可变多簇结构的动态概率粒子群优化算法[J]. *软件学报*, 2009, 20(2): 339-349. (Ni Q J, Zhang Z Z, Wang Z Z, et al. Dynamic probabilistic particle swarm optimization based on varying multi-cluster structure[J]. *J of Software*, 2009, 20(2): 339-349.)
- [11] Angeline P J. Using selection to improve particle swarm optimization[C]. *Proc of IEEE World Congress on Computational Intelligence*. Anchorage, 1998: 84-89.
- [12] Zhang W J, Xie X F. DEPSO: Hybrid particle swarm with differential evolution operator[C]. *Proc of IEEE Int Conf on System, Man and Cybernetics*. Washington DC, 2003: 3816-3821.
- [13] Liang J J, Qin K, Suganthan P N. Comprehensive learning particle swarm optimization for global optimization of multimodal functions[J]. *IEEE Trans on Evolutionary Computation*, 2006, 6(3): 281-295.
- [14] He S, Wu Q H, Wen J Y, et al. A particle swarm optimizer with passive congregation[J]. *BioSystems*, 2004, 78: 135-147.
- [15] Silva, Neves A, Costa E. An empirical comparison of particle swarm and predator prey optimisation[M]. *Lecture Notes in Artificial Intelligence*. Berlin: Springer, 2002: 103-110.
- [16] Charnov E L. Optimal foraging: The marginal value theorem[J]. *Theoretical Population Biology*, 1976, 9(2): 129-136.
- [17] Fretwell D S, Lucas H L. On territorial behavior and other factors influencing habitat distribution in birds[J]. *Acta Biotheoretica*, 1970, 19(1): 16-32.
- [18] Sutherland W J, Parker G A. The relationship between continuous input and interference models of ideal free distributions with unequal competitors[J]. *Animal Behaviour*, 1992, 44(2): 345-355.
- [19] Shi Y, Eberhart R C. Parameter selection in particle swarm optimization[C]. *Lecture Notes in Computer Science*. Berlin: Springer, 1998: 591-600.
- [20] Carlisle A, Dozier G. An off-the-shelf PSO[C]. *Proc of the Workshop on Particle Swarm Optimization*. Indianapolis, 2001: 1-6.
- [21] Wajnberg E, Bernhard P, Hamelin F, et al. Optimal patch time allocation for time-limited foragers[J]. *Behavioral Ecology and Sociobiology*, 2006, 60(1): 1-10.

~~~~~

(上接第1839页)

- [6] Li H, Jiang T, Zhang K. Efficient and robust feature extraction by maximum margin criterion[J]. *IEEE Trans Neural Network*, 2006, 17(1): 157-165.
- [7] Li zhi-feng, Liu Wei, Lin Du-hua, et al. Noparametric discriminant analysis for face recognition[J]. *IEEE Trans on Pattern Analysis on Machine Intelligence*, 2009, 31(4): 755-761.
- [8] Liu Jun, Chen Song-can, Tan Xiao-yang, et al. Comments on "efficient and robust feature extraction by maximum margin criterion"[J]. *IEEE Trans Neural Network*, 2007, 18(6): 1862-1864.
- [9] Saito N, Beylkin G. Multiresolution representation using the auto-correlation functions of compactly supported wavelets[J]. *IEEE Trans on Signal Processing*, 1993, 41(12): 3584-3590.
- [10] 冉启文, 谭立英. 小波分析与分数傅里叶变换及应用[M]. 北京: 国防工业出版社, 2003: 6-42. (Ran Q W, Tan L Y. *Wavelet analysis and fractional fourier transform and applications*[M]. Beijing: National Defence Industry Press, 2003: 6-42.)