

扩展的微粒群算法

莫思敏^{1,2}, 曾建潮¹, 谢丽萍¹

(1. 太原科技大学 复杂系统与计算智能实验室, 山西 太原 030024; 2. 兰州理工大学 电气工程与信息工程学院, 甘肃 兰州 730050)

摘要: 微粒间的作用方式是影响微粒群算法的关键因素. 为克服微粒群算法的早熟问题, 提出一种扩展的微粒群算法(EP SO). 基于拟态物理学中的引斥力思想, 重新构建微粒间的作用方式. 通过微粒间适应值的比较定义微粒间作用的引斥力规则, 使微粒在所有微粒对其产生的引斥力的合力方向上随机地移动寻找最优解. 扩展的微粒群算法与相关算法进行比较, 仿真结果表明: 它能够有效提高微粒群算法的全局优化性能.

关键词: 微粒群算法; 拟态物理学; 引斥力规则

中图分类号: TP301.6 **文献标识码:** A

Extended particle-swarm optimization algorithm

MO Si-min^{1,2}, ZENG Jian-chao¹, XIE Li-ping¹

(1. Complex System and Computational Intelligence Laboratory, Taiyuan University of Science and Technology, Taiyuan Shanxi 030024, China;

2. College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou Gansu 730050, China)

Abstract: The interaction among particles is a key factor affecting the performance of particle swarm optimization (PSO) algorithm. To overcome the premature convergence, an extended particle swarm optimization (EP SO) algorithm is proposed, in which the interaction mechanism among particles is redefined based on the idea of attraction and repulsion forces in Artificial Physics. Furthermore, the rule of attraction and repulsion among particles is defined by comparing particle fitness values. To look for the global optimum, each particle randomly moves along the direction of the resultant force produced by all particles. Simulation results show that EP SO algorithm effectively improves the global performances of other related algorithms.

Key words: particle-swarm optimization algorithm; artificial physics; attraction and repulsion force rule

1 引言(Introduction)

微粒群算法(PSO)^[1]是由美国社会心理学家 Kennedy和电子工程师Eberhart于1995年共同提出的一种基于种群寻优的随机搜索算法. 算法受鸟群和鱼群等群体觅食行为的启发, 通过群体间微粒的协作来实现对问题最优解的搜索.

为了提高微粒群算法的全局收敛性, 一方面, 研究者通过改变微粒间的作用方式来增强种群多样性^[2-5], 避免算法过早陷入局部极小. 文献[2]改变微粒间的作用方式, 提出保证种群多样性的微粒群算法(ARPSO). 根据种群多样性控制微粒飞向或远离群体历史最优位置, 动态地调整算法“勘探”与“开发”比例, 以提高算法性能. 文献[3]则受生物中捕食猎物协同进化的启发, 将微粒分成两类: Predator与Prey, Predator粒子在搜索过程中迫使陷入局部最优点的粒子逃离, 而Prey粒子则受Predator粒子的排斥作用逐步靠近全局最优解. 以上改进方法中, 微粒

受到一些微粒的排斥作用, 避免了算法的早熟收敛, 但是由于微粒仅受个别微粒的影响, 而其余微粒的有价值信息未被运用到微粒的搜索过程中, 因此, 影响了算法的性能. 另一方面, 通过拓展微粒间信息交互方式^[6-7]来增强算法的全局搜索性能, 也是一种十分有效的避免算法早熟收敛的方法. 文献[7]设计微粒的搜索过程受其邻域所有微粒的影响. 通过拓展微粒间信息交互方式, 提出了完全信息的微粒群算法(FIPS), 在一定程度上改善了PSO算法的性能. 但是由于微粒仅受邻域微粒的吸引作用, 因此, 微粒仍然易过早聚集, 而导致算法停止进化.

本文引用拟态物理学中的引斥力思想, 提出扩展的微粒群算法(EP SO), 并证明它能以概率1收敛到全局最优解. EP SO算法首先拓展微粒间信息交互方式, 使每个微粒在搜索过程中受所有微粒历史最好的影响; 其次, 从拟态物理学角度重新构建微粒之间的作用方式——吸引和排斥, 以此来平衡算法的开采

与开发,进而提高算法的全局性能.

2 拟态物理学(Artificial physics)

由美国怀俄明州立大学Spear W等人提出的拟态物理学^[8]是一种模拟物体间存在虚拟力作用以及物体运动遵循牛顿力学定律的方法.该方法指具有速度和位置的个体,在个体之间虚拟力(引力和斥力)的作用下产生速度,进而改变个体位置.通过模拟个体之间的虚拟力控制机器人完成各种任务,如群机器人编队等任务.在文献[9]的群机器人编队中,利用牛顿万有引力定律定义机器人之间虚拟作用力的大小为 $F = Gm_i m_j / r^p$,其中: G 为万有引力常量, m 为机器人的质量, r 为机器人之间的距离, p 是用户定义的一个权重, $p \in [-5, 5]$.而定义机器人 i 和 j 之间所受力的方向,即引斥力规则为:当 $r < R$ 时,虚拟力表现为斥力;当 $r > R$ 时,虚拟力表现为引力;当 $r = R$ 时,微粒间斥力和引力达到平衡, R 为引斥力平衡距离.

3 扩展的微粒群算法EPSO(Extended particle swarm optimization algorithm)

3.1 EPSO算法(EPSO algorithm)

考虑如下全局优化问题:

$$\min \sigma = f(x), \text{ s.t. } x \in \Omega \subset \mathbb{R}^n. \quad (1)$$

本文对微粒群算法进行扩展,将微粒仅受自身历史最好和种群历史最好的影响,扩展为微粒受种群中所有微粒历史最好的影响.同时,从拟态物理学引斥力角度构造了微粒间的作用方式,将PSO和FIPS中微粒仅受其他微粒吸引的作用方式,扩展为微粒受比自身适应值优的微粒的吸引,同时受比自身适应值劣的微粒的排斥,在这些引力和斥力所产生的合力的作用下进行速度和位置的调整.

微粒间引斥力规则定义为:

微粒 j 吸引微粒 i : $p_{jk}(t) - x_{ik}(t)$, 如果 $j \in B(i)$. $B(i) = \{j | f(P_j) \leq f(X_i), \forall j \in S\}$, 即 $B(i)$ 集合存放历史最优适应值比微粒 i 适应值优的微粒.

微粒 j 排斥微粒 i : $-(p_{jk}(t) - x_{ik}(t))$, 如果 $j \in W(i)$. $W(i) = \{j | f(P_j) > f(X_i), \forall j \in S\}$, 即 $W(i)$ 集合存放历史最优适应值比微粒 i 适应值劣的微粒.

其中: S 表示种群,集合 $B(i)$ 和 $W(i)$ 的元素个数分别被表示为 $|B(i)|$ 和 $|W(i)|$.

EPSO的速度和位置更新方程为

$$v_{ik}(t+1) = \omega v_{ik}(t) + \sum_{j \in B(i)} c_j r_{jk} (p_{jk}(t) - x_{ik}(t)) - \sum_{j \in W(i)} c_j r_{jk} (p_{jk}(t) - x_{ik}(t)), \quad (2)$$

$$x_{ik}(t+1) = x_{ik}(t) + v_{ik}(t+1), \quad (3)$$

其中: $P_i = \{p_{i1}, \dots, p_{in}\}$ 代表微粒 i 所经历过的最好位置, $X_i(t) = \{x_{i1}(t), x_{i2}(t), \dots, x_{in}(t)\}$ 和 $V_i(t) = \{v_{i1}(t), v_{i2}(t), \dots, v_{in}(t)\}$ 分别是第 t 代微粒 i 的位置和速度矢量, ω 为惯性权重,用来平衡算法的全局搜索和局部搜索,在区间 $[0, 1]$ 上取值, $c_j (j = 1, \dots, N)$ 是加速常数, r_j 是在 $[0, 1]$ 均匀分布的相互独立的随机数.

3.2 EPSO算法步骤(Steps of EPSO)

Step 1 初始化种群. 种群 S 中每个微粒每一维的速度和位置分别在 $[-v_{\max}, v_{\max}]$ 和 $[-x_{\max}, x_{\max}]$ 内随机的产生; 计算个体适应值; 微粒历史最优位置 $P_i(0)$ 等于各微粒的初始位置, 进化代数 $t = 0$;

Step 2 根据公式(2)计算微粒的下一代速度;

Step 3 根据公式(3)计算微粒的下一代位置;

Step 4 计算微粒适应值并更新每一微粒的历史最优位置 P_i ;

Step 5 判断是否满足结束条件, 若满足, 则停止计算, 并输出最优结果; 若不满足, 进化代数 $t = t + 1$, 返回步骤Step 2.

4 EPSO算法的收敛分析(Convergence analysis of EPSO)

4.1 收敛条件(The convergent condition)

本节基于线性系统稳定性理论对EPSO的收敛性进行分析. 在方程(2)–(3)中, 由于每一维变量的更新独立于其他维变量, 因此, 以一维来讨论EPSO算法的收敛性, 并将方程(2)和方程(3)合并为

$$x_i(t+1) + \omega x_i(t-1) - (1 + \omega - (\sum_{j \in B(i)} c_j r_j - \sum_{j \in W(i)} c_j r_j)) x_i(t) = \sum_{j \in B(i)} c_j r_j p_j(t) - \sum_{j \in W(i)} c_j r_j p_j(t). \quad (4)$$

将方程(4)看作是以 $\sum_{j \in B(i)} p_j(t) - \sum_{j \in W(i)} p_j(t)$ 为输入的离散时间二阶系统.

为了分析 $\{E x_i(t)\}$ (随机变量 $x_i(t)$ 的期望)序列的收敛条件, 将方程(4)表达为

$$E x_i(t+1) - (1 + \omega - \frac{1}{2} (\sum_{j \in B(i)} c_j - \sum_{j \in W(i)} c_j)) E x_i(t) + \omega E x_i(t-1) = \frac{1}{2} \sum_{j \in B(i)} c_j p_j(t) - \frac{1}{2} \sum_{j \in W(i)} c_j p_j(t). \quad (5)$$

则方程(5)的特征方程为

$$\lambda^2 - (1 + \omega - \frac{1}{2} (\sum_{j \in B(i)} c_j - \sum_{j \in W(i)} c_j)) \lambda + \omega = 0. \quad (6)$$

由线性系统稳定性理论可知, $\{E x_i(t)\}$ 的收敛条件为两个特征根 λ_1 和 λ_2 的幅值均小于1, 求解得

$\{E x_i(t)\}$ 的收敛条件为

$$0 \leq \omega < 1 \text{ 且 } 0 < \sum_{j \in B(i)} c_j - \sum_{j \in W(i)} c_j < 4(1 + \omega).$$

设 $\lim_{t \rightarrow +\infty} E X_i(t) = X$, 则将 X 代入方程(4)得

$$c_j(X - p_g) + \sum_{\substack{j \in B(i) \\ j \neq g}} c_j(X - E p_j(t)) - \sum_{j \in W(i)} c_j(X - E p_j(t)) = 0. \quad (7)$$

当且仅当 $E x_i(t) = X = E p_j(t) = p_g$ 时, 方程(7)成立, 即当所有微粒满足收敛条件时, 它们将收敛于 p_g 位置.

群体中微粒的加速系数取值决定了微粒是做收敛运动还是发散运动, 因此, 加速系数对EPSO算法具有重要影响. 若仅考虑最简单的情况 $c_1 = \dots = c_n = c$, 则 $\{E x_i(t)\}$ 收敛条件可变为: $0 \leq \omega < 1$ 且 $0 < c(|B(i)| - |W(i)|) < 4(1 + \omega)$.

1) 对于满足 $|B(i)| - |W(i)| \leq 0$ 的微粒 i 而言, 此时不论 c 取何值, $\{E x_i(t)\}$ 是发散的.

2) 对于满足 $|B(i)| - |W(i)| > 0$ 的微粒 i 而言, a) 因为 $|B(\text{worst})| = \max(|B(i)|) = N$ 且 $|W(\text{worst})| = \min(|W(i)|) = 0$, 所以 $|B(\text{worst})| - |W(\text{worst})| = \max(|B(i)| - |W(i)|) = N$. 这样当 $c \in (0, \frac{4(1 + \omega)}{N})$ 时, $\{E x_i(t)\}$ 是收敛的; b) $\min(|B(i)| - |W(i)|) = 1$, 这样当 $c \in (4(1 + \omega), \infty)$ 时, $\{E x_i(t)\}$ 是发散的; c) 当 $c \in (\frac{4(1 + \omega)}{N}, 4(1 + \omega))$ 时, $\{E x_i(t)\}$ 可能是发散的也可能是收敛的.

由于 r_j 是服从一定分布的随机变量, 因此对于 $c \in (\frac{4(1 + \omega)}{N}, 4(1 + \omega))$ 和 $c \in (4(1 + \omega), \infty)$, c 越小种群中做收敛运动的个体越多, 做发散运动的个体越少; 而 c 越大种群中做收敛运动的个体越少, 做发散运动的个体越多.

4.2 全局收敛性分析(Analysis of global convergence)

假设 1 1) 问题(1)的可行域 Ω 是 \mathbb{R}^n 中的有界闭区域; 2) 目标函数 $f(x)$ 在区域 Ω 上连续.

定义 1 若 $P\{\lim_{n \rightarrow \infty} \xi_n = \xi\} = 1$, 或者对 $\forall \varepsilon > 0$, 有 $P\{\bigcap_{n=1}^{\infty} \bigcup_{k \geq n} A_k\} = 0$, 则称随机序列 $\{\xi_n\}$ 以概率 1 收敛于随机变量 ξ .

引理 1 波雷尔-坎特里(Borel-Cantelli)引理

设 A_1, A_2, \dots 是概率空间上的一事件序列, 令 $p_k = P\{A_k\}$. 若 $\sum_{k=1}^{\infty} p_k < \infty$, 则 $P\{\bigcap_{n=1}^{\infty} \bigcup_{k \geq n} A_k\} = 0$. 若 $\sum_{k=1}^{\infty} p_k = \infty$, 且各 A_k 相互独立, 则 $P\{\bigcap_{n=1}^{\infty} \bigcup_{k \geq n} A_k\} = 1$.

引理 2 设 $\Delta x_{i,k} = x_{i,k}(t + 1) - x_{i,k}(t)$, $r_j \sim N(0, 1)$, $j = 1, \dots, N$, 那么 $\Delta x_{i,k} \sim N(\mu_i, \sigma_i^2)$.

证 设 $\Delta x = \omega v_{ik} + \sum_{j \in B(i)} c_j r_{jk}(p_{jk} - x_{ik}) - \sum_{j \in W(i)} c_j r_{jk}(p_{jk} - x_{ik})$, 设 $\mu_i = \omega v_{ik}$, $\phi_j = c_j(p_{jk} - x_{ik})$, $j \in 1, \dots, N$, 则 $\Delta x_{i,k} = \mu_i + \sum_{j \in B(i)} r_j \phi_j - \sum_{j \in W(i)} r_j \phi_j$. 因为 $r_j \sim N(0, 1)$, 因此, 显然有 $\Delta x_{i,k} \sim N(\mu_i, \sigma_i^2)$. 证毕.

由假设 1 可知, 若任给 $\forall \varepsilon > 0$, 则记 $D_0 = \{x \in \Omega \mid |f(x) - f^*| < \varepsilon\}$; $D_1 = \Omega \setminus D_0$, 其中 $f^* = \min\{f(x) : x \in \Omega\}$. 因此, 在规模为 N 的EPSO算法中, 各微粒第 k 代的历史最优位置 $P(k)$ 被分成两种状态: 1) 至少有一个点的历史最优位置属于 D_0 , 记为状态 I_0 ; 2) N 个点的历史最优位置均属于 D_1 , 记为状态 I_1 .

引理 3 设问题(1)满足假设 1, 若当 $P(k)$ 为状态 I_i 时, $P(k + 1)$ 为状态 I_j 的概率为 $q_{ij}(i, j = 0, 1)$, 则有

- 1) 对于任意状态为 I_0 的点集 $P(k)$, $q_{00} = 1$.
- 2) 对于任意状态为 I_1 的点集 $P(k)$, 存在常数 $b \in (0, 1)$, 使得 $q_{11} \leq b$.

证 根据EPSO算法步骤可知, $q_{00} = 1$ 显然成立.

$f(x)$ 是区域 Ω 上的连续函数, 若 x_0 为 $f(x)$ 上的一个最小值点, 则存在 $r > 0$, 使得当 $\|x - x_0\| \leq r$ 时, 有 $|f(x) - f(x_0)| < \varepsilon/2$. 若记 $Q_{x_0,r} = \{x \in \Omega \mid \|x - x_0\|_{\infty} \leq r\}$, 则有

$$Q_{x_0,r} \subset D_0. \quad (8)$$

由引理 2 知 $\Delta x_{i,k} \sim N(\mu_i, \sigma_i^2)$, 则

$$P(x_i) = P\{(x_i + \Delta x) \in Q_{x_0,r}\} = \prod_{k=1}^n \int_{x_{0,k} - x_{i,k} - r}^{x_{0,k} - x_{i,k} + r} \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(y - \mu_i)^2}{2\sigma_i^2}} y, i \in S, \quad (9)$$

其中 $x_{0,k}, x_{i,k}$ 分别表示 x_0 和 x_i 的第 k 个分量.

则由式(9)可知 $0 < P(x_i) < 1$. 由于EPSO算法中微粒 i 的历史最好位置

$$P_i = \begin{cases} P_i, & f(x_i + \Delta x) > f(P_i), \\ x_i + \Delta x, & f(x_i + \Delta x) \leq f(P_i). \end{cases}$$

所以, $0 < P(P_i) < 1$, 而在有限集合 S 中, 存在 $l \in S$, 使得

$$P(P_l) = \min_{i \in S} (P(P_i)), 0 < P(P_i) < 1. \quad (10)$$

由式(8)(10)得 $q_{10} \geq P(P_i) \geq P(P_l)$, 由于 $q_{11} + q_{10} = 1$, 则 $q_{11} = 1 - q_{10} \leq 1 - P(P_l) = b, 0 < b < 1$. 证毕.

定理 1 设 $\{P_g\}$ 是EPSO算法产生的解序列, 其中 $P_g(k) = \arg \min_{1 \leq i \leq N} f(P_i(k))$ 为第 k 代时微粒群的最优位置, 如果问题(1)满足假设 1, 则有

$$P\{\lim_{k \rightarrow \infty} f(P_g(k)) = f^*\} = 1$$

成立, 即解序列 $\{P_g\}$ 以概率1收敛于问题(1)的全局最优解.

证 对于 $\forall \varepsilon > 0$, 令 $P_k = P\{|f(P_g(k)) - f^*| \geq \varepsilon\}$, 其中 f^* 为全局最优解, 则

$$P_k = \begin{cases} 0, & \exists T \in (0, \dots, k), \text{使得} |f(P_g(T)) - f^*| < \varepsilon; \\ \bar{P}_k, & \forall T \in (0, \dots, k), \text{使得} |f(P_g(T)) - f^*| \geq \varepsilon. \end{cases}$$

由引理3可知 $\bar{P}_k = q_{11}^k \leq b^k$, 所以 $\sum_{k=1}^{\infty} \bar{P}_k \leq \sum_{k=1}^{\infty} b^k = \frac{b}{1-b} < \infty$, 又由引理1可知,

$$P\{\bigcap_{n=1}^{\infty} \bigcup_{k \geq n} |f(P_g(k)) - f^*| \geq \varepsilon\} = 0,$$

因此, 由定义1可得, $f(P_g(k))$ 以概率1收敛于 f^* .

证毕.

4.3 全局收敛性的物理学解释(Physical interpretation of global convergence)

全局收敛性证明表明微粒以概率1遍历搜索空间中的每一点. 加速系数 c 和微粒的适应值共同影响微粒所受合力的大小及其运动方向. 由收敛条件 $0 \leq \omega < 1$ 且 $0 < c(|B(i)| - |W(i)|) < 4(1 + \omega)$ 可知, 种群中存在 $|B(i)| - |W(i)| > 0$ 的微粒, 这些微粒在不同的加速系数 c 的取值范围内做收敛或发散运动, 对于做收敛运动的微粒而言, 它们从不同方向收敛于 P_g , 并在其附近进行精细的局部搜索. 而种群中同时存在 $|B(i)| - |W(i)| \leq 0$ 的微粒, 它们背离 P_g 所在的位置, 从不同的方向以一定概率发散地探索未搜索区域. 微粒在遍历空间中的任意一点时, 根据相应的适应值和加速系数 c , 以相应的步长做收敛于或发散于群体最优位置的运动. 因此, 微粒可以根据遍历位置的不同, 在空间中的任意方向上, 以任意的合力大小运动到空间中的任意位置.

微粒在引斥力的合力方向上随机移动的搜寻范围是Benchmark优化函数所定义的上下界范围, 而其搜索的宽度为360度.

5 仿真测试(Simulation tests)

为了评价EPSO算法的性能, 本文选择5个经典的优化函数来进行测试, 函数表达式如下:

1) Sphere函数

$$f_1(x) = \sum_{i=1}^n x_i^2,$$

其中 $-100 \leq x_i \leq 100$, 在 $x_i = 0$ 时达到极小值0.

2) Rosenbrock函数

$$f_2(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2),$$

其中 $-50 \leq x_i \leq 50$, 在 $x_i = 1$ 时达到极小值0.

3) Griewank函数

$$f_3(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$$

其中 $-600 \leq x_i \leq 600$, 在 $x_i = 0$ 时达到极小值0.

4) Ackley函数

$$f_4(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e,$$

其中 $-32 \leq x_i \leq 32$, 在 $x_i = 0$ 时达到极小值0.

5) Rastrigin函数

$$f_5(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10),$$

其中 $-5.12 \leq x_i \leq 5.12$, 在 $x_i = 0$ 时达到极小值0.

5.1 加速系数对EPSO算法性能的影响(Impact of accelerating coefficients on EPSO)

由于加速系数 c 不仅影响微粒的收敛运动, 同时还决定了微粒移动的步长, 因此, 本实验研究加速系数对EPSO算法性能的影响. 根据收敛条件取 $c = 1$, $c = 4$, $c = 10$ 和 $c = 20$ 进行试验. 其中, $c = 1$ 和 $c = 4$ 属于 $(\frac{4(1+\omega)}{N}, 4(1+\omega))$, 即种群中同时存在做收敛或发散运动的微粒, 且 c 值越小做收敛运动的个体越多. 而 $c = 10$ 和 $c = 20$ 属于 $(4(1+\omega), \infty)$ 即种群中大部分微粒做发散运动, 且 c 值越大做发散运动的个体越多. 实验数据均源于算法独立运行30次的平均抽样. 函数的维数 $n = 50$ 维, 群体规模 $N = n$. 若算法在连续500代内所求函数最优解未发生变化时, 则认为该算法已收敛, 算法停止运行. $V_{\max} = X_{\max}$. 惯性权重 ω 采取从0.9到0.4线性递减取值. 实验结果如表1所示.

从表1可知, EPSO算法的性能随着 c 值的增加而变优. $c = 1$ 时, 算法具有最差的性能. 而算法在 $c = 10$ 和 $c = 20$ 下性能相同且对于大部分函数而言, EPSO算法都能获得相应函数的全局最优解. 原因可能是: $c = 1$ 和4较之 $c = 10$ 和20种群中做收敛运动的个体多, 而做发散运动的个体少, 因此, 群体多样性差于后者. 同时在 $c = 1$ 时, 由于微粒受到较小的力, 导致微粒长时间在各自的很小领域内进行搜索, 大大减弱了算法的全局搜索能力. 当 $c = 10$ 和 $c = 20$ 时, 由于种群中做发散运动的个体增多, 种群多样性较好, 且微粒受到了相对较大的力, 以较大的速度进行全局搜索从而带来了最佳的算法性能. 从表1中当 $c = 10$ 和 $c = 20$ 时的平均值、方差和CPU时间数据看出, 算法具有非常稳定的性能及较短的运行时间.

表 1 不同加速系数下EPSO算法的性能

Table 1 Performances of EPSO under different acceleration coefficients

函数	$c = 1$		$c = 4$		$c = 10(c = 20)$		
	平均值	方差	平均值	方差	平均值	方差	CPU时间/s
f_1	8.2E+2	0	0	0	0	0	17.17(12.50)
f_2	3.3E+5	4.9E+1	4.9E+1	0	0	0	24.4(22.9)
f_3	8.3	0	0	0	0	0	13.53(27.06)
f_4	6.3	5.9E-16	5.9E-16	0	0	0	42.70(35.93)
f_5	1.5E+1	2.5E-1	0	0	0	0	14.03(15.10)

注 1 表1中 $c = 10$ 和 20 下, 算法的平均值和方差相同而CPU运行时间略有不同, 括号中的数据表示 $c = 20$ 下的运行时间. 由于在 $c = 10$ 和 $c = 20$ 下算法性能优于 $c = 1$ 和 $c = 4$ 下, 因此, 实验仅给出 $c = 10$ 和 20 下的CPU时间. 一次实验的CPU时间计算方式是算法首次达到对应函数平均值的时间. 若在一次实验中, 算法未能达到对应函数的平均值, 则CPU时间为本次实验算法运行结束的时间. 表中1中的CPU时间数据是30次实验的平均值.

5.2 EPSO在高维函数上的性能(Performances of EPSO in high-dimension functions)

通过5.1实验数据及分析, 本实验选择加速系数 $c = 10$, 来测试EPSO算法在高维函数上的性能. 算法的种群规模 $N = n$. 惯性权重 ω 的取值和算法停止条件与5.1实验相同.

从表2可以看出, 对于大部分函数而言, EPSO算法都能找到其全局最优解. 这表明EPSO算法比较适合求解高维函数.

表 2 EPSO在高维函数上的性能

Table 2 Performances of EPSO in high-dimension functions

函数	100		200		300	
	平均值	方差	平均值	方差	平均值	方差
f_1	0	0	0	0	0	0
f_2	9.9E+1	0	1.9E+2	0	2.9E+2	0
f_3	0	0	0	0	0	0
f_4	5.9E-16	0	5.9E-16	0	5.9E-16	0
f_5	0	0	0	0	0	0

5.3 EPSO与PSO, ARPSO和拟态物理优化算法(APO)^[10]的性能比较(Comparisons of Performance of EPSO with PSO, ARPSO and APO)

通过5.1实验数据及分析, 本实验选择加速系数 $c = 10$, 进行各算法的性能比较. 同时为了尽可能的在相同条件下与各算法进行比较, EPSO算法的

参数——种群规模、函数评价次数、惯性权重的取值和函数范围, 随比较算法的不同而各异. 其中PSO和ARPSO算法的参数及实验结果取自文献[2], 而APO算法的参数及实验结果取自文献[10]. 且选取这些文献中相同的测试函数及函数维数 $n = 50$ 进行比较. 比较结果如表3所示.

表 3 EPSO与PSO, ARPSO和APO算法性能比较

Table 3 Comparisons of performance of EPSO with PSO, ARPSO and APO

算法	Griewank	Ackley	Rastrigin
PSO	1.3E-2	6.7E-1	4.7E+1
ARPSO	3.0E-12	2.7E-2	2.0E-2
EPSO	0	5.9E-16	0
APO	0	2.4E-1	0
EPSO	0	5.9E-16	0

从表3可知, 对于多峰函数Griewank, Ackley和Rastrigin而言, EPSO在指定时间内所得解的质量明显优于PSO和ARPSO. 在相同环境下, EPSO算法性能胜过APO算法, 而对于Griewank和Rastrigin函数而言, 两者表现都最佳. 图1-3从最优解的角度描述了EPSO和APO算法的平均动态性能. 从图上可知, 对于Rastrigin, Griewank和Ackely函数而言, EPSO算法大概在5代左右就可找到表3中相应函数的解, 而APO算法却需要分别在140代、180代和200代以后找到上表相应的解, 这充分说明EPSO算法的搜索速度明显快于APO算法.

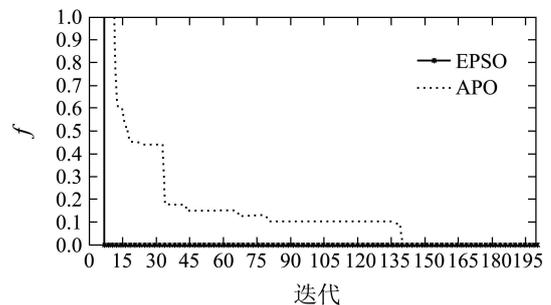


图 1 Rastrigin函数的适应值演化过程

Fig. 1 Fitness evolution of Rastrigin function

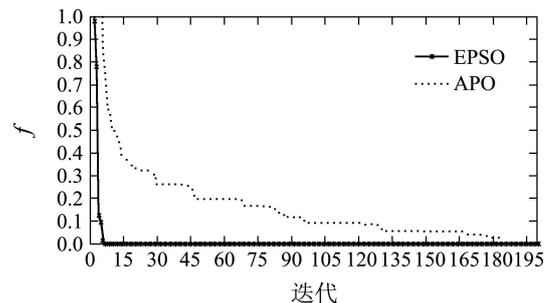


图 2 Griewank函数的适应值演化过程

Fig. 2 Fitness evolution of Griewank function

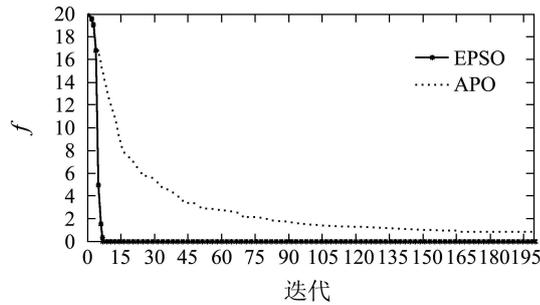


图3 Ackley函数的适应值演化过程

Fig. 3 Fitness evolution of Ackley function

由于ARPSO算法中存在扩散运动,因此,群体多样性较好.但是群体中价值微粒的信息未被运用到每个微粒的搜索过程中,导致ARPSO的性能劣于EPSO.而EPSO与APO算法的作用力规则相同,都是通过微粒(个体)之间的适应值比较控制微粒的受力方向.由于微粒间斥力作用的存在以及算法搜索过程中参考了所有微粒的信息,因此,EPSO和APO表现更佳.EPSO性能优于APO的原因可能是:1)EPSO中微粒具有记忆性,而APO中个体不具有记忆性.2)在EPSO算法中,微粒受力的大小是在 $[0, \text{加速常数}]$ 区间上随机选择,引斥力的大小与微粒的适应值无关,微粒在引斥力所产生的合力方向上以随机步长向适应值优的解靠近,保持了群体的多样性,而APO中个体的质量是个体适应值的函数,适应值较优的个体具有较大的引力吸引其他个体运动到其邻域进行搜索,这样势必会加快其他个体向其靠近的速度,从而降低了群体多样性.

6 结论(Conclusion)

本文从拟态物理学角度扩展了PSO算法,利用拟态物理学中的引斥力思想构造了微粒间相互作用的引斥力规则,建立了一种扩展的微粒群算法(EPSO).该算法首先将标准PSO中微粒仅受种群历史最好微粒影响扩展为微粒受种群所有微粒的历史最好影响,这使微粒能够获得更为全面的搜索信息;其次,通过微粒间适应值的比较构建了微粒间的引斥力规则,使微粒在所有微粒的引斥力的合力方向上随机移动寻找全局最优解.仿真结果表明EPSO具有快的

收敛速度和好的群体多样性.

参考文献(References):

- [1] KENNEDY J, EBERHART R C. Particle swarm optimization [C] // *Proceedings of IEEE International Conference on Neural Networks*. Piscataway, NJ: IEEE Service Center, 1995, 4: 1942 – 1948.
- [2] RIGET J, VESTTERSTROM J S. A diversity-guided particle swarm optimizer-the ARPSO [R]. *EVALife: Department of Computer Science, University of Aarhus*. Denmark, 2002.
- [3] SILVA A, NEVES A, COSTA E. An empirical comparison of particle swarm and predator prey optimization [C] // *The 13th Irish International Conference Artificial Intelligence and Cognitive Science 2002*. Limerick, Ireland: Springer, 2002: 103 – 110.
- [4] NIU Ben, ZHU Yunlong, HE Xiaoxian, et al. An improved particle swarm optimization based on bacterial chemotaxis [C] // *Proceedings of the 6th World Congress on Intelligent Control and Automation*. Dalian, China: IEEE, 2006: 3193 – 3197.
- [5] 陈保娣, 曾建潮. 改进的吸引扩散微粒群算法 [J]. *控制理论与应用*, 2010, 27(4): 451 – 456.
(CHEN Baodi, ZENG Jianchao. Modified attractive and repulsive particle swarm optimization [J]. *Control Theory & Applications*, 2010, 27(4): 451 – 456.)
- [6] HE S, WU Q H, WEN J Y, et al. A particle swarm optimizer with passive congregation [J]. *Biosystems*, 2004, 78(1/3): 135 – 147.
- [7] 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.
- [8] SPEARS W, SPEARS D, HEIL R, et al. An overview of physicomimetics [C] // *Proceedings of Swarm Robotics*. Berlin: Springer, 2004: 84 – 97.
- [9] SPEARS W, HEIL R, SPEARS D, et al. Physicomimetics for mobile robot formations [C] // *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi Agent Systems (AA-MAS'04)*. Washington, USA: IEEE Computer Society, 2004, 3: 1528 – 1529.
- [10] XIE Liping, ZENG Jianchao. The performance analysis of artificial physics optimization algorithm driven by different virtual forces [J]. *ICIC Express Letters*, 2010, 1(4): 239 – 244.

作者简介:

莫思敏 (1977-), 女, 副教授, 博士研究生, 目前研究方向为智能计算, E-mail: minsim@tom.com;

曾建潮 (1963-), 男, 教授, 博士生导师, 目前研究方向为智能计算, E-mail: zengjianchao@263.net;

谢丽萍 (1978-), 女, 副教授, 博士, 目前研究方向为智能计算, E-mail: Xieliping1978@gmail.com.