

基于信息素机制的粒子群优化算法的设计与实现

吕 强¹ 刘士荣¹ 邱雪娜²

摘 要 提出了一种基于信息素机制的粒子群优化 (Particle swarm optimization based on pheromone mechanism, PSO-PM) 算法. 主要是借鉴了蚁群优化算法的信息素共享机制, 并引入到粒子群优化算法中, 设计了粒子行为的三条简单规则: 信息留存规则、信息获取和融合规则以及粒子演化规则, 从而实现了群体信息的充分分享, 相应地改善了算法的寻优能力. 采用基准函数对 PSO-PM 算法进行测试, 并与几种不同类型的改进优化算法进行对比, 数值实验结果验证了 PSO-PM 算法的有效性.

关键词 信息素机制, 粒子群优化, 蚁群优化, 演化规则, 概率分布
中图分类号 TP18

Design and Realization of Particle Swarm Optimization Based on Pheromone Mechanism

LV Qiang¹ LIU Shi-Rong¹ QIU Xue-Na²

Abstract A particle swarm optimization based on pheromone mechanism (PSO-PM) is proposed. Through introducing the idea of pheromone-shared mechanism used by ant colony optimization to the particle swarm optimization, and designing three simple behavior rules including reserving information rule, requiring and syncretizing information rule, and evolving rule, population information can be fully shared. Therefore, the algorithm's ability of searching optimum value is improved. Compared with other optimization algorithms for the benchmark functions in the experiment, the obtained results have demonstrated the effectiveness of proposed algorithm.

Key words Pheromone mechanism, particle swarm optimization (PSO), ant colony optimization, evolution rule, probability distribution

粒子群优化 (Particle swarm optimization, PSO) 算法是基于对鸟群、鱼群的模拟, 自 Kennedy 和 Eberhart^[1] 提出以来, 已经发展了 10 余年, 由于 PSO 概念和参数调整都很简单而且容易实现, 因此, 被广泛应用在模糊控制器设计^[2]、机器人路径规划^[3] 等领域中. 但粒子群优化算法也存在早熟收敛的问题, 为此, 专家和学者们在进行深入的研究和分析后, 提出了一些有价值的解决方案, 例如: 多种群方法^[4-5] 利用各个种群的独立搜索和相互通信来提高算法的寻优能力; 更新公式法^[6-7] 通过对粒子群优化算法基本更新公式的改进提高算法的寻优能力; 混合法^[8-9] 将粒子群优化算法和其他类型的优化算法相结合, 提高算法的寻优能力; 极值扰动法^[10-11] 采用群体适应度方差或停止进化的步数来

判断群体的收敛程度, 并当群体趋于收敛时, 对极值进行扰动, 从而提高算法的寻优能力.

粒子群优化算法是建立在这样一种假设基础之上的, 即“群体中信息的充分分享, 可以提供一种进化的有利条件”^[1]. 粒子群优化算法就是基于该假设而开发出来的, 由于粒子群优化算法较强的优化能力, 也在一方面验证了该假设的合理性和正确性. 在本文中, 我们站在信息交流、信息分享的角度上, 从假设基础出发, 对粒子群优化算法及其上述类型的改进算法进行分析.

从粒子群优化算法及其改进算法的位置更新公式中, 我们可以看到群体中的粒子仅仅和全局最优粒子、历史最优粒子进行交互 (有的文献也使用了分群体最优粒子等), 也就是只考虑了群体最优的信息和自己本身最优的信息, 而其他粒子所留下的信息并没有被有效利用. 换句话说, 我们在考察粒子与粒子之间的信息交流时, 发现粒子之间的信息交流仅仅是通过全局最优粒子进行的, 粒子本身的信息和粒子历史最优信息并没有进行有效交流. 尽管 Kennedy 等^[12-13] 进一步提出了邻域拓扑的思想, 粒子间信息交流的程度有所扩大, 但受限于粒子群算法的基本更新公式的框架, 算法对群体自身所具有的信息仍然没有充分挖掘, 群体中粒子之间的信

收稿日期 2008-07-15 收修改稿日期 2009-01-17
Received July 15, 2008; in revised form January 17, 2009
国家自然科学基金 (60675043), 宁波市自然科学基金 (2008A610002) 资助

Supported by National Natural Science Foundation of China (60675043) and Ningbo Natural Science Foundation (2008A610002)

1. 杭州电子科技大学自动化学院 杭州 310018 2. 宁波工程学院电信学院 宁波 315211

1. School of Automation, Hangzhou Dianzi University, Hangzhou 310018 2. School of Telecommunication, Ningbo University of Technology, Ningbo 315211
DOI: 10.3724/SP.J.1004.2009.01410

息交流程度仍然不够. 所以, 在一定的程度上, 也就限制了以粒子群优化算法的基本更新模式为基础的改进算法寻优能力的进一步提高. 如果能够使粒子位置的更新不但要考虑全局最优粒子和该粒子本身的历史最优粒子所留下的信息, 还要充分考虑其他粒子留下的信息, 那么将会有效地扩大粒子之间的信息交流程度, 提高粒子群优化算法的寻优能力, 对此蚁群优化算法给了我们重要的启发.

蚁群优化算法是由 Dorigo^[14-15] 于 20 世纪 90 年代提出的一种群体智能优化算法, 其主要思想是模拟自然界中蚁群从蚁巢到目的地之间寻找最短路径时的交互过程, 这种交互过程是通过一种称之为信息素的物质进行的, 从而使蚂蚁之间能够相互协作, 完成复杂的任务. 在此基础上, 为了提高蚁群优化算法的寻优能力, 也出现了大量的改进算法, 如 Max-Min 蚂蚁系统^[16]. 目前, 蚁群优化算法及其改进算法已经被成功地应用到了组合优化领域中, 并且在连续优化领域中也有一定的表现, 考察其优化的思想, 主要是蚂蚁通过一种信息素共享机制, 实现它们之间间接的相互通信, 信息交流充分, 从而使整个蚁群涌现出较高的智能, 可以完成较为复杂的任务.

如果能够将蚁群优化算法的信息素共享机制引入到粒子群优化算法之中, 构建一种粒子之间可以共享信息的方式, 目的是用来增加粒子之间的信息交流程度, 将会提高算法的优化能力. 也就是说, 通过借鉴蚁群优化算法信息素共享机制的思想, 在粒子群优化算法中, 为粒子之间的信息交流设计了信息交流平台, 从而使粒子之间不但实现了全局最优信息的交流, 而且也可以进行粒子本身信息和其历史最优信息的交流, 实现了群体中信息的充分享.

为了实现信息的充分享, 实现上述解决方案, 本文设计了信息素矩阵的结构和三条简单的粒子行为规则, 从而提出一种基于信息素机制的粒子群优化算法 (Particle swarm optimization based on pheromone mechanism, PSO-PM), 基准函数测试验证了 PSO-PM 算法的有效性.

1 基于信息素机制的粒子群优化算法的设计

根据引言所介绍的 PSO-PM 算法的基本思想, 我们知道, PSO-PM 算法的演化原理由三部分组成: 1) 信息留存规则; 2) 信息获取和融合规则; 3) 演化规则. 以下将分别描述三种规则的原理, 首先考察粒子状态的定义.

定义 1. 粒子的状态由两部分组成: 一部分是位置信息; 一部分是评价信息. 在时刻 t 第 i 个粒子的状态用 $I_i^t(P_i^t, F_i^t)$ 表示, 其中 P_i^t 表示在时刻 t 第 i 个粒子位置信息; F_i^t 表示在时刻 t 第 i 个粒子评价

信息.

图 1 显示的是粒子的信息留存规则. 圆圈表示粒子, 共有 8 个粒子, 每个粒子都有自己当前的状态, 分别由此时的位置信息和评价信息组成. 图 1 中显示出, 第 2 个粒子通过评价信息留存 (Reserve fitness information, RFI) 规则和位置信息留存 (Reserve position information, RPI) 规则将粒子当前状态的评价信息和位置信息分别存放到评价信息素矩阵和位置信息素矩阵.

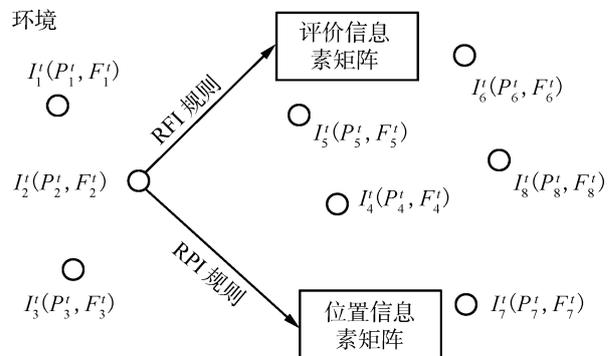


图 1 信息留存规则

Fig. 1 The rule of reserving information

图 2 显示的是粒子的信息获取和融合规则. 第 2 个粒子依据信息获取和融合 (Require and syncretize information, RSI) 规则, 从评价信息素矩阵和位置信息素矩阵获得此时群体的评价信息和位置信息, 然后产生融合信息, 并由第 2 个粒子获得. 需要说明的是, 由于评价信息和位置信息是两种不同类型的信息, 这两种类型的信息所赋予的含义是不同的, 本文所有部分, 仅仅在融合规则、信息融合中涉及到了融合的概念, 对此, 本文中“融合”所表示的含义专指这两种类型信息的组合.

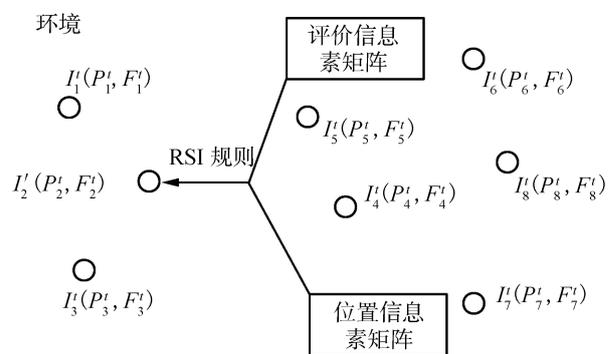


图 2 信息获取和融合规则

Fig. 2 The rule of requiring and syncretizing information

图 3 显示的是粒子演化规则. 第 2 个粒子利用所获得的融合信息, 根据粒子演化 (Particle evolution, PE) 规则, 产生新的粒子状态. 其他的粒子也

按照上述的三种规则不断地演化, 每个粒子都通过评价信息素矩阵和位置信息素矩阵间接地与其他粒子进行信息交流. 信息交流充分, 最终整个群体会涌现出较高的智能.

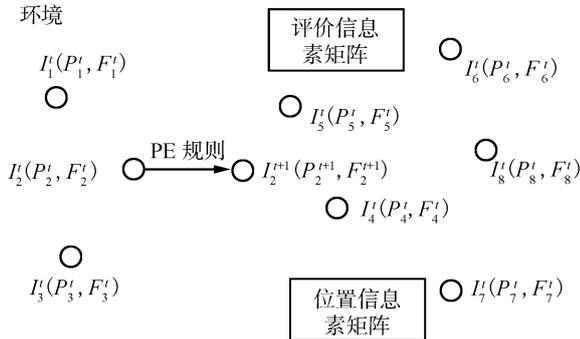


图 3 演化规则

Fig. 3 The rule of particle evolvement

2 基于信息素机制的粒子群优化算法的实现

2.1 信息素矩阵的结构

信息素矩阵共包括两部分, 一部分是评价信息素矩阵, 另一部分是位置信息素矩阵, 其构造方式如图 4 所示.

u	1	2	3	...	$d-1$	d
1	u_{11}	u_{12}	u_{13}	...	u_{1d-1}	u_{1d}
2	u_{21}	u_{22}	u_{23}	...	u_{2d-1}	u_{2d}

(a) 评价信息素矩阵 u

(a) The structure of fitness pheromone matrix u

v	1	2	3	...	$d-1$	d
1	v_{11}	v_{12}	v_{13}	...	v_{1d-1}	v_{1d}
2	v_{21}	v_{22}	v_{23}	...	v_{2d-1}	v_{2d}

(b) 位置信息素矩阵 v

(b) The structure of position pheromone matrix v

图 4 信息素矩阵的构造方式

Fig. 4 The structure of pheromone matrix

图 4(a) 表示的是评价信息素矩阵, 它存储的内容主要是有关粒子适应度的信息, 该矩阵尺寸是 $2 \times d$, 即 2 行 d 列 (d 为问题的规模, 即粒子的长度), 其中第 1 行 $u_{11}, u_{12}, \dots, u_{1d}$ 保存的是粒子和全局最优粒子之间产生的适应度信息素, 第 2 行 $u_{21}, u_{22}, \dots, u_{2d}$ 保存的是粒子和历史最优粒子之间产生的适应度信息素. 由于保存的信息素是关于适应度的信息, 即保存的是种群与环境交互后得到的评价信

息, 所以该矩阵被称为评价信息素矩阵.

图 4(b) 表示的是位置信息素矩阵, 它主要保存关于粒子位置的信息, 该矩阵的规模也是 $2 \times d$, 其中第 1 行 $v_{11}, v_{12}, \dots, v_{1d}$ 保存的是粒子和全局最优粒子之间产生的位置信息素, 第 2 行 $v_{21}, v_{22}, \dots, v_{2d}$ 保存的是关于粒子与历史最优粒子之间产生的位置信息素. 由于保存的是关于粒子在搜索空间中位置的信息, 所以该信息素矩阵被称为位置信息素矩阵.

2.2 信息留存规则

根据 PSO-PM 算法的设计思想以及信息素矩阵的结构, 评价信息留存规则 (RFI rule) 如式 (1) 和 (2) 所示.

$$\mathbf{u}_1^t = \mathbf{u}_1^t + |F(\mathbf{p}_g) - F(\mathbf{y}_j)| + m_{\text{weight}} \quad (1)$$

$$\mathbf{u}_2^t = \mathbf{u}_2^t + |F(\mathbf{p}_j) - F(\mathbf{y}_j)| + m_{\text{weight}} \quad (2)$$

其中, \mathbf{u}_1^t 表示评价信息素矩阵的第 1 行; \mathbf{u}_2^t 表示评价信息素矩阵的第 2 行; \mathbf{p}_g 是全局最优粒子; \mathbf{p}_j 是第 j 个粒子的历史最优粒子; \mathbf{y}_j 是第 j ($j = 1, 2, \dots, n$, n 是种群的规模) 个粒子; $|\cdot|$ 表示绝对值函数; $F(\cdot)$ 表示适应度函数; m_{weight} 表示加重参数; t 表示迭代次数.

位置信息留存规则 (RPI rule) 如式 (3) 和 (4) 所示.

$$\mathbf{v}_1^t = c_1 \times \mathbf{v}_1^t + c_2 \times \text{rand}(1) \times |c_1 \times \mathbf{p}_g - c_2 \times \mathbf{y}_j| \quad (3)$$

$$\mathbf{v}_2^t = c_1 \times \mathbf{v}_2^t + c_2 \times \text{rand}(1) \times |c_1 \times \mathbf{p}_j - c_2 \times \mathbf{y}_j| \quad (4)$$

其中, \mathbf{v}_1^t 表示位置信息素矩阵的第 1 行; \mathbf{v}_2^t 表示位置信息素矩阵的第 2 行; c_1, c_2 为权重参数; $\text{rand}(1)$ 表示产生满足标准正态分布的一个随机数.

2.3 信息获取和融合规则

根据 PSO-PM 算法的设计思想可知, 粒子需要将信息素矩阵中的信息提取出来, 由于信息素的类型不同, 一种是评价信息素, 另一种是位置信息素, 因此, 还需要将这两种类型的信息素融合. 所以, 信息获取和融合规则 (RSI rule) 可以根据以下不同的方法加以实现.

1) 基于正态分布的信息获取和融合规则

采用正态分布对信息素矩阵中的位置信息素和评价信息素进行融合, 规则如式 (5) 和 (6) 所示.

$$\text{temp}1_j = \text{rand}(1, d) \odot \mathbf{u}_1^t + \mathbf{v}_1^t \quad (5)$$

$$\text{temp}2_j = \text{rand}(1, d) \odot \mathbf{u}_2^t + \mathbf{v}_2^t \quad (6)$$

其中, $\text{temp}1_j$ 表示第 j 个粒子所获得的第一个融合信息, 并且, 该向量的第 i ($i = 1, 2, \dots, d$) 维

$temp1_{ji}$ 服从正态分布 $\mathcal{N}(v_{1i}^t, (1/u_{1i}^t)^2)$; \odot 表示向量之间的点除; $rand(1, d)$ 表示产生满足标准正态分布的 d 个随机数; $temp2_j$ 表示第 j 个粒子所获得的第二个融合信息, 并且, 该向量的第 i ($i = 1, 2, \dots, d$) 维 $temp2_{ji}$ 服从正态分布 $\mathcal{N}(v_{2i}^t, (1/u_{2i}^t)^2)$.

2) 基于柯西分布的信息获取和融合规则

采用柯西分布作为信息获取和融合的规则, 如式 (7) 和 (8) 所示.

$$temp1_j = v_1^t - 1 \odot (\tan(rand(1, d)\pi) \otimes u_1^t) \quad (7)$$

$$temp2_j = v_2^t - 1 \odot (\tan(rand(1, d)\pi) \otimes u_2^t) \quad (8)$$

其中, $temp1_j$ 表示第 j 个粒子所获得的第一个融合信息, 并且, 该向量的第 i ($i = 1, 2, \dots, d$) 维 $temp1_{ji}$ 服从柯西分布 $\mathcal{C}(v_{1i}^t, (1/u_{1i}^t))$; $rand(1, d)$ 表示产生满足均匀分布的 d 个随机数; \otimes 表示向量之间的点乘; $temp2_j$ 表示第 j 个粒子所获得的第二个融合信息, 并且, 该向量的第 i ($i = 1, 2, \dots, d$) 维 $temp2_{ji}$ 服从柯西分布 $\mathcal{C}(v_{2i}^t, (1/u_{2i}^t))$.

3) 基于韦伯分布的信息获取和融合规则

采用韦伯分布作为信息获取和融合的规则, 如式 (9) 和 (10) 所示.

$$temp1_j = v_1^t \otimes (-\ln(rand(1, d)))^{1 \circ u_1^t} \quad (9)$$

$$temp2_j = v_2^t \otimes (-\ln(rand(1, d)))^{1 \circ u_2^t} \quad (10)$$

其中, $temp1_j$ 表示第 j 个粒子所获得的第一个融合信息, 并且, 该向量的第 i ($i = 1, 2, \dots, d$) 维 $temp1_{ji}$ 服从韦伯分布 $\mathcal{W}(u_{1i}^t, v_{1i}^t)$; $temp2_j$ 表示第 j 个粒子所获得的第二个融合信息, 并且, 该向量的第 i ($i = 1, 2, \dots, d$) 维 $temp2_{ji}$ 服从韦伯分布 $\mathcal{W}(u_{2i}^t, v_{2i}^t)$.

2.4 演化规则

粒子在获得融合信息后, 按照粒子演化规则进行粒子位置的更新. 本文设定阈值概率 $l_{th} \in [0, 1]$, 如果满足 $rand(1) < l_{th}$, 则采用式 (11) 表示粒子的演化, 否则采用式 (12) 表示粒子的演化.

$$y_j = rand(1) \quad (11)$$

$$y_j = c_1 \times temp1_j + c_2 \times temp2_j \quad (12)$$

其中, $rand(1)$ 表示在粒子的位置空间中产生均匀分布的一个随机值.

2.5 算法流程

如上所述, PSO-PM 算法通过引入信息素共享机制, 设定了粒子简单行为规则, 实现了群体信息的充分分享, 其基本步骤如算法 1 所示.

算法 1 (PSO-PM 算法).

输入. 待优化问题;

输出. 群体的全局最优粒子和适应度;

步骤 1. 初始化:

- 1) 初始化种群数量 n 、问题规模 d 和最大迭代次数 Gen ;
- 2) 确定阈值概率 l_{th} 、加重参数 m_{weight} 、权重系数 c_1 和 c_2 ;
- 3) 在问题的解空间中初始化种群 $Popu$;
- 4) 在 $[0, 1]$ 范围内随机初始化评价信息素矩阵, 在问题的解空间中随机初始化位置信息素矩阵.

步骤 2. 如果满足停止条件, 则退出, 否则迭代执行以下步骤:

- 1) 对全体粒子进行适应度评价;
- 2) 存储群体的全局最优粒子和适应度;
- 3) 存储每个粒子的历史最优粒子和适应度;
- 4) 对每个粒子 ($j = 1, 2, \dots, n$), 循环执行以下步骤:
 - a) 执行信息留存规则;
 - b) 执行信息获取和融合规则;
 - c) 执行粒子演化规则.

3 收敛性分析

本节以采用正态分布的信息获取和融合规则的 PSO-PM 算法为例, 证明该算法的收敛性.

定义 2. 连续优化问题模型: 一个连续优化问题模型 (S, Ω, f) 包含三个方面的内容: 解空间 S 、约束集合 Ω 和目标函数 $f: S \rightarrow \mathbf{R}$.

假设 1. S 为有界闭区域, 目标函数为空间 S 上的连续函数.

PSO-PM 算法求解连续优化问题, 以极小化问题为例, 就是搜索至少一个 $\mathbf{x}^* \in S$, 满足 $f(\mathbf{x}^*) \leq f(\mathbf{x}), \forall \mathbf{x} \in S$.

由假设 1, 将解空间 S 分为两个区域^[17]:

任给 $\varepsilon > 0$, 则

$$D_0 = \{\mathbf{x} \in S \mid |f(\mathbf{x}) - f^*| < \varepsilon\}, \quad D_1 = S \setminus D_0 \quad (13)$$

其中, $f^* = \min \{f(\mathbf{x}) \mid \mathbf{x} \in S\}$.

将 PSO-PM 算法中产生的第 k 代种群 $X(k)$, 满足 $X(k) = \mathbf{p}_g \cup S_{loc} \cup S_{iter}$ (\mathbf{p}_g 为全局最优粒子, S_{loc} 为历史最优粒子群体, S_{iter} 为当前粒子群体), 根据式 (13) 划分为两种状态:

- 1) 如果种群中至少有一个个体属于 D_0 , 则状态为 S_0 ;
- 2) 如果种群中的个体均属于 D_1 , 则状态为 S_1 .

引理 1 (Borel-Cantell 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

$$P \left\{ \bigcap_{n=1}^{\infty} \bigcup_{k \geq n} A_k \right\} = 0 \quad (14)$$

若 $\sum_{k=1}^{\infty} p_k = \infty$, 且各 A_k 相互独立, 则

$$P \left\{ \bigcap_{n=1}^{\infty} \bigcup_{k \geq n} A_k \right\} = 1 \quad (15)$$

引理 2. 若记 q_{ij} ($i, j = 0, 1$) 为当 $X(k)$ 为状态 S_i 时, $X(k+1)$ 为状态 S_j 的概率, 则有^[17]:

- 1) 对于任意状态为 S_0 的点集 $X(k)$, $q_{00} = 1$;
- 2) 对于任意状态为 S_1 的点集 $X(k)$, 存在常数 $c \in (0, 1)$, 使得 $q_{11} \leq c$.

证明. PSO-PM 算法采用 \mathbf{p}_g 记录当前的最优解, 所以只要能一次找到全局最优解, 就能永远保留下来. 换言之, 如果当前群体 $X(k)$ 状态为 S_0 , 则 PSO-PM 算法就不会使该种群进化到状态为 S_1 的群体, 故 1) 显然成立.

下面证明 2) 也是成立的. 由于 $f(\mathbf{x})$ 在 S 上连续, 设 \mathbf{x}^* 为最小极值点, 则 $\exists \gamma > 0$, 使得当

$$\|\mathbf{x} - \mathbf{x}^*\| \leq \gamma \quad (16)$$

时, 有

$$|f(\mathbf{x}) - f(\mathbf{x}^*)| < \frac{\varepsilon}{2} \quad (17)$$

令

$$D' = \{\mathbf{x} \in S \mid \|\mathbf{x} - \mathbf{x}^*\| \leq \gamma\} \quad (18)$$

由式 (13) 和 (18), 可知

$$D' \subset D_0 \quad (19)$$

设 PSO-PM 算法当前种群 $X(k)$ 状态为 S_1 , 种群 $X(k+1)$ 的状态由 $\mathbf{p}_g(k+1)$ 、 $S_{loc}(k+1)$ 和 $S_{iter}(k+1)$ 的状态所决定. 而 $\mathbf{p}_g(k+1)$ 和 $S_{loc}(k+1)$ 的状态由 $S_{iter}(k+1)$ 、 $\mathbf{p}_g(k)$ 和 $S_{loc}(k)$ 决定. 由于种群 $X(k)$ 状态为 S_1 , 所以 $\mathbf{p}_g(k)$ 和 $S_{loc}(k)$ 的状态为 S_1 , 因此, $S_{iter}(k+1)$ 的状态起到了决定性的作用. 令该种群中的个体为 \mathbf{x} , 满足 $\mathbf{x} \in S_{iter}(k)$, 由于式 (11) 产生的随机解发生的概率较小, 因此, 可以合并在一起考虑. 采用式 (12) 引导后, 变为 $\mathbf{y} \in S_{iter}(k+1)$, 则

$$P(\mathbf{y} \in D') = \prod_{i=1}^d P(|y_i - x_i^*| \leq \gamma) = \prod_{i=1}^d P(x_i^* - \gamma \leq y_i \leq x_i^* + \gamma) \quad (20)$$

其中, x_i^* 、 y_i 分别表示个体 \mathbf{x}^* 和 \mathbf{y} 的第 i 个分量. 又由于根据式 (12), 有

$$y_i \sim \mathcal{N}(w_i, \sigma_i^2) \quad (21)$$

其中, $\sigma_i = \sum_{j=1}^2 (c_j/u_{ji})$, $w_i = \sum_{j=1}^2 c_j v_{ji}$. 所以

$$P(\mathbf{y} \in D') = \prod_{i=1}^d \int_{x_i^* - \gamma}^{x_i^* + \gamma} \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(x-w_i)^2}{2\sigma_i^2}} dx \quad (22)$$

根据式 (22), 有

$$0 < P(\mathbf{y} \in D') < 1 \quad (23)$$

由于 y_i 是服从正态分布的连续型随机变量, 由式 (22) 可知 $P(\mathbf{y} \in D')$ 连续, 又由于 $\mathbf{y} \in S$ 为有界闭集, 所以存在 $\mathbf{y}_0 \in S$, 满足

$$P(\mathbf{y}_0 \in D') = \min_{\mathbf{y} \in S} P(\mathbf{y} \in D') \quad (24)$$

令 q_{10} 表示当 $X(k)$ 为状态 S_1 时, $X(k+1)$ 为状态 S_0 的概率, 利用式 (19) 和 (24), 可得

$$q_{10} \geq P(\mathbf{y} \in D') \geq P(\mathbf{y}_0 \in D') \quad (25)$$

令

$$c = 1 - P(\mathbf{y}_0 \in D') \quad (26)$$

根据式 (25) 和 (26), 以及 $q_{11} + q_{10} = 1$, 可得

$$q_{11} = 1 - q_{10} \leq 1 - P(\mathbf{y}_0 \in D') = c, \quad 0 < c < 1$$

□

定理 1. 设 $\{X(m)\}$ 是由 PSO-PM 算法产生的种群序列, 其中 $\mathbf{x}^*(m) \in X(m)$ 为第 m 代种群中的最优解, 即

$$\mathbf{x}^*(m) = \arg \min_{1 \leq j \leq n} f(\mathbf{x}_j(m))$$

如果有

$$P \left\{ \lim_{m \rightarrow \infty} f(\mathbf{x}^*(m)) = f^* \right\} = 1 \quad (27)$$

成立, 即种群序列以概率 1 收敛于全局最优解, 也就是说 PSO-PM 算法收敛.

证明. 对于 $\forall \varepsilon > 0$, 记 $p_k = P\{|f(\mathbf{x}^*(k)) - f^*| \geq \varepsilon\}$, 则

$$\exists t \in \{0, 1, \dots, k\}, \text{ 若 } \mathbf{x}^*(t) \in D_0, \text{ 则 } p_k = 0;$$

$$\text{若 } \mathbf{x}^*(m) \notin D_0, m = 0, 1, \dots, k, \text{ 则 } p_k = p'_k.$$

由引理 2 可知

$$p'_k = P\{\mathbf{x}^*(m) \notin D_0, m = 0, 1, \dots, k\} = q_{11}^k \leq c^k$$

有

$$\sum_{k=1}^{\infty} p_k \leq \sum_{k=1}^{\infty} c^k = \frac{c}{1-c} < \infty$$

由引理 1 可知

$$P \left\{ \bigcap_{n=1}^{\infty} \bigcup_{k \geq n} \{|f(\mathbf{x}^*(k)) - f^*| \geq \varepsilon\} \right\} = 0$$

□

4 仿真实验

为了测试 PSO-PM 算法的优化能力, 本文主要从以下三个方面来考察该算法的性能: 1) 固定函数评价次数; 2) 给定精度; 3) 高维函数测试. 在每一个方面, 本文都选取了近年来提出的具有较强优化能力的改进优化算法作为对比算法, 用来作为评估本文提出算法性能的参照. PSO-PM 算法在所有实验中的参数选取为: 权重 $c_1 = 0.8, c_2 = 0.2$, 阈值概率 $l_{th} = 0.01$ (函数 f_3, f_{12} 时, $l_{th} = 0.1$). 另外, 根据粒子的信息获取和融合规则不同, PSO-PM 算法有三种形式: PSO-PM-N (采用高斯分布的信息获取和融合规则), PSO-PM-C (采用柯西分布的信息获

取和融合规则), PSO-PM-W (采用韦伯分布的信息获取和融合规则). 测试函数如表 1 所示.

4.1 固定函数评价次数

本节测试主要是在固定的函数评价次数下, 评测 PSO-PM-N、PSO-PM-C 和 PSO-PM-W 三个算法的优化能力, 并和 FCA^[18] 算法进行比较. 本文的三个算法的环境设置和文献 [18] 一致, 其中, 固定函数评价次数为 1×10^5 , 采用表 1 中函数 $f_1 \sim f_{13}$, 每个函数独立运行 50 次. 4 种算法的平均对比结果见表 2 所示. 可以看出, 在函数 $f_1, f_2, f_4, f_5, f_7 \sim f_9$ 以及函数 f_{13} 上, PSO-PM-N、PSO-PM-C 和 PSO-PM-W 算法均能在 50 次实验中获得函数的全

表 1 测试函数
Table 1 Test functions

f	Function	Dimension	Search space	Min/Best position
f_1	$10^6 x_1^2 + \sum_{i=2}^n x_i^2$	30	$(-100, 100)^N$	0.0/0, 0, \dots , 0
f_2	$\sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	30	$(-100, 100)^N$	0.0/0, 0, \dots , 0
f_3	$\sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	30	$(-100, 100)^N$	0.0/1, 1, \dots , 1
f_4	$\frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$(-600, 600)^N$	0.0/0, 0, \dots , 0
f_5	$\sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	30	$(-5.12, 5.12)^N$	0.0/0, 0, \dots , 0
f_6	$\sum_{i=1}^{n-1} (x_i^2 + x_{i+1}^2)^{0.25} \times [\sin(50 \times (x_i^2 + x_{i+1}^2)^{0.1}) + 1.0]$	30	$(-100, 100)^N$	0.0/0, 0, \dots , 0
f_7	$\sum_{i=1}^n x_i^2$	30	$(-100, 100)^N$	0.0/0, 0, \dots , 0
f_8	$\sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$(-10, 10)^N$	0.0/0, 0, \dots , 0
f_9	$\sum_{i=1}^n (\lfloor x_i + 0.5 \rfloor)^2$	30	$(-100, 100)^N$	0.0/0, 0, \dots , 0
f_{10}	$20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i\right) + 20 + \exp(1) \times \frac{\pi}{n} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$	30	$(-32, 32)^N$	0.0/0, 0, \dots , 0
f_{11}	$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$ $y_i = 1 + \frac{1}{4}(x_i + 1)$	30	$(-50, 50)^N$	0.0/-1, -1, \dots , -1
f_{12}	$\frac{1}{10} \{ \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30	$(-50, 50)^N$	0.0/1, 1, \dots , 1
f_{13}	$\sum_{i=1}^n x_i^2 + \left(\sum_{i=1}^n 0.5 i x_i \right)^2 + \left(\sum_{i=1}^n 0.5 i x_i \right)^4$	30	$(-5, 10)^N$	0.0/0, 0, \dots , 0

局最优值. 并且, 在函数 f_3 , f_6 和 f_{10} 上, PSO-PM-N、PSO-PM-C 和 PSO-PM-W 这三种算法也获得了较优的结果, 但在函数 f_{11} , f_{12} 上, FCA 算法获得了较优的函数平均最优值.

4.2 给定函数精度

本节测试主要是在一定的精度内, 比较算法的均值、方差和函数评价次数. 这些评价指标不但反映了算法的优化能力的强弱, 而且也反映了算法的计算代价的高低. 本节对比的算法来源于文献 [9] 提出的 PSACO 算法、文献 [5] 提出的 OEPSO 算法和文献 [19] 提出的 DPSO 算法, 根据文献 [5] 的设置方式, 本节设置最大评价次数为 3×10^5 , 选取表 1 中的函数 $f_1 \sim f_{13}$, 每个函数独立运行 50 次. 具体实验结果如表 3 和表 4 所示. 实验结果表明了本文提出的算法 PSO-PM-N、PSO-PM-C、PSO-PM-W 都可以获得较优的均值、方差以及较小的函数评价次数. 特别在函数 f_1 , f_2 , $f_4 \sim f_{10}$, f_{13} 所获得的函数评价次数非常小, 其优势相当明显. 即使在函数 f_3 , f_{11} 和 f_{12} 上, 虽然优势不如其他函数那样明显, 但所获得的函数评价次数也是令人满意的. 总的来说, PSO-PM-N、PSO-PM-C 和 PSO-PM-W 对比其他算法, 可以在耗费较小的计算代价下, 达到给定的精度, 或者说这三种算法的优化能力更加依靠群体本身的信息交流, 由于群体中的信息能够充分分享, 因此, 群体只需与环境很少的信息交流, 就可以达到给

定的精度.

4.3 高维函数测试

当函数的维数增加时, 也就增加了问题的复杂性. 为了进一步考察 PSO-PM-N、PSO-PM-C 和 PSO-PM-W 算法的高维优化能力, 当函数维数从 20 到 1000 变化时, 在给定精度的情况下, 观察这几种算法函数评价次数的变化. 实验配置和 IM-CPA^[20] 一致, 设置最大评价次数为 7×10^4 , 选取表 1 中的函数 f_3 , f_4 , f_5 和 f_{10} , 每个函数独立运行 50 次. 具体结果见表 5 和表 6 所示. 从表 5 和表 6 中可以看出, 除函数 f_3 外, 对其他测试函数来说, 维数的增加对本文提出的 3 种算法几乎没有影响. 虽然函数 f_3 显示的测试结果表明, 当问题的规模增大时, PSO-PM-N、PSO-PM-W 算法的评价次数也在增大, 但这两种算法都可以在给定的最大评价次数内达到给定的精度. 需要特别说明的是 PSO-PM-C 算法, 在测试函数 f_3 上显示的结果中, 当维数从 100 到 1000 变化时, 可以看到对 PSO-PM-C 算法的计算代价的影响并不大.

表 7 显示了上述 4 种算法在超高维的情况下, 对函数 f_4 和 f_5 的优化结果. 从中可以看出, 即使在超高维的情况下, PSO-PM-N、PSO-PM-C 和 PSO-PM-W 均都获得了较小的函数评价次数. 这也表明了尽管问题复杂, 这三种算法却可以付出较小的计算代价而达到要求的精度, 优化能力强.

表 2 PSO-PM-N、PSO-PM-C、PSO-PM-W 与 FCA 算法的性能比较

Table 2 The comparison of PSO-PM-N with PSO-PM-C, PSO-PM-W and FCA

f	f_{\min}	Mean				Std			
		PSO-PM-N	PSO-PM-C	PSO-PM-W	FCA	PSO-PM-N	PSO-PM-C	PSO-PM-W	FCA
f_1	0	0.0	0.0	0.0	/	0.0	0.0	0.0	/
f_2	0	0.0	0.0	0.0	4.53E-7	0.0	0.0	0.0	1.52E-7
f_3	0	1.96E-4	9.05E-5	1.05E-4	2.74	5.6E-4	1.42E-4	1.69E-4	2.49
f_4	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
f_5	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
f_6	0	1.58E-8	1.86E-8	1.69E-8	/	3.79E-8	3.46E-8	2.91E-8	/
f_7	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
f_8	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
f_9	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
f_{10}	0	8.88E-16	8.88E-16	8.88E-16	1.56E-7	2.02E-31	2.02E-31	2.02E-31	3.12E-7
f_{11}	0	1.13E-5	8.35E-6	4.80E-5	7.94E-11	2.26E-5	2.12E-5	1.09E-4	4.25E-12
f_{12}	0	1.72E-6	2.35E-6	2.98E-6	2.65E-14	3.07E-6	3.44E-6	3.67E-6	8.17E-14
f_{13}	0	0.0	0.0	0.0	/	0.0	0.0	0.0	/

表 3 PSO-PM-N、PSO-PM-C 与 PSACO 算法的性能比较
Table 3 The comparison of PSO-PM-N with PSO-PM-C and PSACO

f	Required accuracy	Mean			Variance			Number of function evaluations		
		PSO-PM-N	PSO-PM-C	PSACO	PSO-PM-N	PSO-PM-C	PSACO	PSO-PM-N	PSO-PM-C	PSACO
f_1	1E-10	2.81E-11	3.80E-11	9.26E-11	6.37E-22	8.51E-22	3.88E-23	150	133	75 580
f_2	1E-4	2.93E-5	3.56E-5	9.37E-5	5.59E-10	8.36E-10	2.19E-11	78	74	110 660
f_3	1E-3	2.98E-4	3.56E-4	43.977	1.06E-7	1.07E-7	4 950.2	8 682	9 879	300 000
f_4	1E-4	3.71E-5	3.02E-5	4.68E-2	8.34E-10	6.91E-10	4.10E-3	64	61	227 400
f_5	1E-4	3.66E-5	3.93E-5	1.35E-5	7.36E-10	6.74E-10	6.93E-10	54	54	123 400
f_6	1E-4	4.09E-5	4.14E-5	4.08	1.02E-9	9.07E-10	60.488	389	371	112 205
f_7	1E-4	3.61E-5	3.44E-5	9.18E-5	7.90E-10	6.08E-10	3.91E-11	57	57	39 302
f_8	1E-4	5.57E-5	6.36E-5	1.4E-5	4.85E-10	4.94E-10	4.76E-6	77	85	27 120
f_9	1E-4	0.0	0.0	0.0	0.0	0.0	0.0	30	31	13 876
f_{10}	1E-4	5.84E-5	6.31E-5	0.6255	4.95E-10	5.41E-10	1.1041	74	74	102 500
f_{11}	1E-6	4.52E-7	2.51E-7	9.21E-7	1.36E-13	7.14E-14	1.12E-14	9 071	7 651	24 859
f_{12}	1E-4	2.81E-5	3.35E-5	8.63E-5	8.05E-10	9.95E-10	3.98E-10	2 481	3 396	27 996
f_{13}	1E-8	2.70E-9	3.32E-9	1.97E-9	6.88E-18	7.34E-18	2.97E-18	125	118	33 703

表 4 PSO-PM-W 与 DPSO、OEPSO 算法的性能比较
Table 4 The comparison of PSO-PM-W with DPSO and OEPSO

f	Required accuracy	Mean			Variance			Number of function evaluations		
		PSO-PM-W	DPSO	OEPSO	PSO-PM-W	DPSO	OEPSO	PSO-PM-W	DPSO	OEPSO
f_1	1E-10	4.20E-11	9.35E-11	/	8.00E-22	1.50E-22	/	151	34 185	/
f_2	1E-4	3.57E-5	1.07E-4	9.794E-5	7.31E-10	1.23E-9	2.006E-6	77	268 220	189 607
f_3	1E-3	3.53E-4	22.65	9.723E-4	7.53E-8	1 053.3	2.219E-5	10 292	300 000	242 548
f_4	1E-4	3.56E-5	5.8E-2	1.780E-2	7.68E-10	1.6E-3	1.27E-2	60	241 130	206 236
f_5	1E-4	3.40E-5	2.98E-1	8.721E-5	9.01E-10	1.3074	2.328E-5	57	147 180	148 426
f_6	1E-4	3.46E-5	0.281	/	7.57E-10	0.2279	/	354	120 220	/
f_7	1E-4	3.38E-5	9.52E-5	9.416E-5	5.64E-10	2.68E-11	5.171E-6	58	13 464	48 207
f_8	1E-4	5.39E-5	9.61E-5	9.335E-5	4.26E-10	2.23E-11	1.435E-5	79	18 656	78 654
f_9	1E-4	0.0	0.0	0.0	0.0	0.0	0.0	30	7 974	55 033
f_{10}	1E-4	5.37E-5	9.70E-5	9.770E-5	5.33E-10	8.64E-12	1.644E-6	71	25 696	90 228
f_{11}	1E-6	3.75E-7	9.21E-7	9.619E-7	9.22E-14	1.06E-14	3.658E-8	9 740	24 597	70 576
f_{12}	1E-4	3.35E-5	8.93E-5	9.250E-5	8.68E-10	2.82E-10	1.265E-5	2 879	21 335	57 405
f_{13}	1E-8	4.12E-9	9.92E-9	/	7.78E-18	1.41E-21	/	119	269 770	/

表 5 对于函数 f_3 、 f_4 , 几种算法函数平均评价次数比较
Table 5 The comparison of numbers of function evaluations for several algorithms on functions f_3 and f_4

Dimension	$f_3 (\varepsilon = 1E-2)$				$f_4 (\varepsilon = 1E-4)$			
	PSO-PM-N	PSO-PM-C	PSO-PM-W	IMCPA	PSO-PM-N	PSO-PM-C	PSO-PM-W	IMCPA
20	2 741	1 593	3 514	/	60	60	57	2 421
100	4 676	7 569	6 773	/	61	59	60	6 713
200	7 701	8 869	5 984	/	60	63	59	8 460
400	9 758	9 861	17 074	/	58	60	57	15 365
1 000	14 925	8 612	24 543	/	64	60	61	30 906

表 6 对于函数 f_5 、 f_{10} , 几种算法函数平均评价次数比较Table 6 The comparison of numbers of function evaluations for several algorithms on functions f_5 and f_{10}

Dimension	$f_5 (\varepsilon = 1E-1)$				$f_{10} (\varepsilon = 1E-3)$			
	PSO-PM-N	PSO-PM-C	PSO-PM-W	IMCPA	PSO-PM-N	PSO-PM-C	PSO-PM-W	IMCPA
20	35	36	36	1 469	59	62	64	1 776
100	43	41	40	4 988	58	63	60	5 784
200	41	38	42	5 747	63	67	57	9 728
400	47	42	43	12 563	61	65	63	13 915
1 000	47	45	45	24 408	60	61	58	26 787

表 7 对于函数 f_4 、 f_5 , 在超高维的情况下, 几种算法函数平均评价次数比较Table 7 The comparison of numbers of function evaluations for several algorithms on functions f_4 and f_5 with higher dimensions

Dimension	$f_4 (\varepsilon = 1E-4)$				$f_5 (\varepsilon = 1E-1)$			
	PSO-PM-N	PSO-PM-C	PSO-PM-W	IMCPA	PSO-PM-N	PSO-PM-C	PSO-PM-W	IMCPA
2 000	62	60	62	43 003	50	51	49	37 879
5 000	64	60	59	125 847	48	53	52	87 245
10 000	56	61	64	147 037	52	52	52	143 700

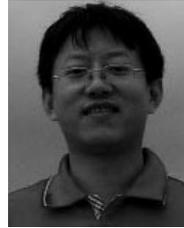
5 结论

通过在粒子群优化算法中引入信息素共享机制, 本文提出了基于信息素机制的粒子群优化算法 (PSO-PM). 该算法通过简单的三条规则: 信息留存规则、信息获取和融合规则以及粒子演化规则即可实现. 同时, 收敛性分析保证了该算法最终收敛到较优解. 并且, 仿真实验从三个方面验证了本文提出的 PSO-PM 算法具有较强的寻优能力, 即验证了信息的充分分享对于提高种群的寻优能力起到了关键性的作用.

References

- Kennedy J, Eberhart R C. Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Network. Perth, Australia: IEEE, 1995. 1942–1948
- Karakuzu C. Retraction notice to: fuzzy controller training using particle swarm optimization for nonlinear system control. *ISA Transactions*, 2008, **48**(2): 229–239
- Alici G, Jagielski R, Sekercioglu Y A, Shirinzadeh B. Prediction of geometric errors of robot manipulators with particle swarm optimisation method. *Robotics and Autonomous Systems*, 2006, **54**(12): 956–966
- Jia Dong-Li, Zhang Jia-Shu. Niche particle swarm optimization combined with chaotic mutation. *Control and Decision*, 2007, **22**(1): 117–120
(贾东立, 张家树. 基于混沌变异的小生境粒子群算法. *控制与决策*, 2007, **22**(1): 117–120)
- Cong Lin, Sha Yu-Heng, Jiao Li-Cheng. Organizational evolutionary particle swarm optimization for numerical optimization. *Pattern Recognition and Artificial Intelligence*, 2007, **20**(2): 145–153
(丛琳, 沙宇恒, 焦李成. 组织进化粒子群数值优化算法. *模式识别与人工智能*, 2007, **20**(2): 145–153)
- Veeramachaneni K, Peram T, Mohan C, Osadciw L A. Optimization using particle swarms with near neighbor interactions. In: Proceedings of Genetic and Evolutionary Computation. Chicago, USA: Springer, 2003. 110–121
- Cui Zhi-Hua, Zeng Jian-Chao. Modified particle swarm optimization based on differential model. *Journal of Computer Research and Development*, 2006, **43**(4): 646–653
(崔志华, 曾建潮. 基于微分模型的改进微粒群算法. *计算机研究与发展*, 2006, **43**(4): 646–653)
- Liu B, Wang L, Jin Y H, Tang F, Huang D X. Improved particle swarm optimization combined with chaos. *Chaos, Solitons and Fractals*, 2005, **25**(5): 1261–1271
- Shelokar P S, Siarry P, Jayaraman V K, Kulkarni B D. Particle swarm and ant colony algorithms hybridized for improved continuous optimization. *Applied Mathematics and Computation*, 2007, **188**(1): 129–142
- Hu Wang, Li Zhi-Shu. A simpler and more effective particle swarm optimization algorithm. *Journal of Software*, 2007, **18**(4): 861–868
(胡旺, 李志蜀. 一种更简化而高效的粒子群优化算法. *软件学报*, 2007, **18**(4): 861–868)
- Lv Zhen-Su, Hou Zhi-Rong. Particle swarm optimization with adaptive mutation. *Acta Electronica Sinica*, 2004, **32**(3): 416–420
(吕振肃, 侯志荣. 自适应变异的粒子群优化算法. *电子学报*, 2004, **32**(3): 416–420)
- Kennedy J, Mendes R. Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Application and Reviews*, 2006, **36**(4): 515–519
- Mendes R, Kennedy J, Neves J. The fully informed particle swarm: simpler, maybe better. *IEEE Transactions on Evolutionary Computation*, 2004, **8**(3): 204–210

- 14 Dorigo M, Maniezzo V, Colorni A. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 1996, **26**(1): 29–41
- 15 Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1997, **1**(1): 53–66
- 16 Stutzle T, Hoos H H. Max-min ant system. *Future Generation Computer Systems*, 1999, **16**(8): 889–914
- 17 Guo Chong-Hui, Tang Huan-Wen. Global convergence properties of evolution strategies. *Mathematica Numerica Sinica*, 2001, **23**(1): 105–110
(郭崇慧, 唐焕文. 演化策略的全局收敛性. 计算数学, 2001, **23**(1): 105–110)
- 18 Khilwani N, Prakash A, Shankar R, Tiwari M K. Fast clonal algorithm. *Engineering Applications of Artificial Intelligence*, 2008, **21**(1): 106–128
- 19 Xie X F, Zhang W J, Yang Z L. Dissipative particle swarm optimization. In: *Proceedings of IEEE Congress on Evolutionary Computation*. Hawaii, USA: IEEE, 2002. 1456–1461
- 20 Du Hai-Feng, Gong Mao-Guo, Jiao Li-Cheng, Liu Ruo-Chen. An immune clonal algorithm for functions with higher dimensions. *Progress in Natural Science*, 2004, **14**(8): 925–933
(杜海峰, 公茂果, 焦李成, 刘若辰. 用于高维函数优化的免疫记忆克隆规划算法. 自然科学进展, 2004, **14**(8): 925–933)



吕 强 杭州电子科技大学讲师, 博士. 主要研究方向为智能移动机器人的导航与控制. 本文通信作者.

E-mail: lvqiang@hdu.edu.cn

(**LV Qiang** Lecturer, Ph.D. at Hangzhou Dianzi University. His research interest covers navigation and control of intelligent mobile robot. Corresponding author of this paper.)



刘士荣 杭州电子科技大学教授. 主要研究方向为复杂系统建模与控制、智能机器人与智能系统.

E-mail: liushirong@hdu.edu.cn

(**LIU Shi-Rong** Professor at Hangzhou Dianzi University. His research interest covers modeling and control of complex system, intelligent robot, and intelligent system.)



邱雪娜 宁波工程学院讲师, 博士研究生. 主要研究方向为智能机器人系统.

E-mail: qiuxn26@hotmail.com

(**QIU Xue-Na** Lecturer, Ph. D. candidate at Ningbo University of Technology. Her main research interest is intelligent robot system.)