

一种模拟羊群行为的新型群集智能算法

曲大鹏¹,许伦祥¹,鲁焱光²,原晓坤¹,黄敏³,王兴伟⁴

(1. 辽宁大学信息学院,辽宁 沈阳 110036;2. 东北大学计算机科学与工程学院,辽宁 沈阳 110169;
3. 东北大学信息科学与工程学院,辽宁 沈阳 110169;4. 东北大学软件学院,辽宁 沈阳 110169)

摘要: 针对当前多种群集智能算法往往包含过多策略而掩盖算法核心机理的现状,设计了一种模拟羊群行为的新型群集智能算法. 该算法从群集智能算法的核心出发,通过分别模拟羊群的三类行为:头羊引领、羊群互动和牧羊犬监督,设计群集智能算法中相对应的三种策略:全局探索、局部开发和跳出局部优化. 基准测试函数的实验结果表明,较之粒子群算法,该算法能获得更高质量的解,同时具有更快的收敛速度和更好的稳定性.

关键词: 群集智能; 测试函数; 全局探索; 局部开发

中图分类号: O224 **文献标识码:** A **文章编号:** 0372-2112 (2018)06-1300-06

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2018.06.004

A New Swarm Intelligence Algorithm by Simulating Sheep Behaviors

QU Da-peng¹, XU Lun-xiang¹, LU Yao-guang², YUAN Xiao-kun¹, HUANG Min³, WANG Xing-wei⁴

(1. College of Information, Liaoning University, Shenyang Liaoning 110036, China;

2. School of Computer Science and Engineering, Northeastern University, Shenyang Liaoning 110169, China;

3. College of Information Science and Engineering, Northeastern University, Shenyang Liaoning 110169, China;

4. College of Software, Northeastern University, Shenyang Liaoning 110169, China)

Abstract: The status of current various swarm intelligence algorithms contains many strategies which cover their core issues, therefore, a new swarm intelligence algorithm by simulating sheep behaviors optimization (SO) is devised. Based on the core of swarm intelligence algorithms, SO designs three kinds of strategies, namely global exploration, local exploitation and escaping from local optimization, by simulating three different kinds of corresponding sheep behaviors, namely bellwether guidance, moving to each other, and shepherd dog supervision respectively. The experimental results of benchmark test functions demonstrate that, compared with particle swarm optimization, SO gets better solution, faster convergence speed and better stability.

Key words: swarm intelligence; test function; global exploration; local exploitation

1 引言

随着人类对自然界认知的不断深入,以微粒群算法(Particle Swarm Optimization, PSO)^[1]等为代表的群集智能算法得到了越来越广泛的关注. 它们通过众多行为简单的个体之间的相互作用涌现产生整体智能行为,现已被成功应用于求解图像识别和网络设计等工程问题^[2]. 虽然模拟的生物行为不同,但它们具有某些共同特性,如进行全局探索和局部开发.

群集智能算法的理想状态应该是能够快速找到可行

解,并在陷入局部优化解时及时跳出从而继续寻找全局最优解. 但在实际应用中,现有群集智能算法往往由于个体间协同不够直接,而导致收敛较慢,如 PSO 中的惯性权重使得微粒个体继承当前速度. 而且,研究者们为了追求性能,往往加入了大量改进策略,虽然提高了性能,但也因为引入了额外参数,不仅使得算法变得复杂,而且一定程度上掩盖了群集智能算法的核心机理^[3].

因此,我们从群集智能算法核心出发,即在开始时合理进行全局探索和局部开发以快速收敛到可行解,及时判断是否陷入局部优化及执行相应跳出机制,并

收稿日期: 2017-04-26; 修回日期: 2017-08-02; 责任编辑: 郭游

基金项目: 国家自然科学基金(No. 61572123); 国家杰出青年科学基金(No. 71325002); 国家自然科学基金重点国际合作研究项目(No. 71620107003); 教育部-中国移动科研基金(No. MCM20160201); 辽宁省博士科研启动基金(No. 20170520323); 辽宁大学博士科研启动项目

且具有尽量少参数以简化算法^[4]. 通过模拟羊群 (Sheep Optimization, SO) 中三类行为: 头羊引领、羊群互动和牧羊犬监督, 分别设计相对应的三种策略: 全局探索、局部开发和跳出局部优化. 具体为: 通过头羊引领羊群实现快速全局探索, 使得羊群快速向已知最优解靠近; 通过羊群之间的互相移动来实现局部开发, 进一步加快收敛; 应用牧羊犬监督来实现跳出局部优化解.

2 相关工作

PSO 算法中的每个粒子代表优化问题的一个解, 粒子根据记忆、个体认知 (向自己历史最优解学习强化局部开发能力) 和社会认知 (向种群历史最优解学习表示粒子间的协作和信息共享) 来不断调整自己的速度和位置, 逐渐靠近最优解. 为了提高 PSO 算法的性能, 研究者们主要从参数设置和与其它算法相结合两方面进行改进, 并取得了大量的研究成果. 例如, 文献[5]提出了一种对不同粒子采用不同学习因子的方法, 即适应度值较好的粒子个体采用较大学习因子, 反之采用较小学习因子. 文献[6]将具有量子行为的 PSO 算法与单纯型算法相结合, 前者引领优化的全局域, 后者用作局部搜索以进一步调整从前者获得的解. 文献[7]将反向学习策略引入 PSO 算法, 使粒子在搜寻过程中找到当前解的反向位置, 增加寻找到全局最优解的可能性, 并且基于透镜成像原理, 引入缩放因子和搜索半径以平衡算法的全局探索和局部开发能力.

除了 PSO 算法外, 当前还有许多受其它群体现象启发的群集智能算法. 例如蚁群优化算法通过信息素进行相互协作, 形成正反馈来实现优化^[8]. 人工蜂群算法模拟蜂群的智能采蜜行为, 将搜索过程分为: 雇佣蜂阶段、观察蜂阶段和侦察蜂阶段, 前两个阶段用来寻找新的食物源, 后一个阶段用来生成新的食物源^[9]. 布谷鸟搜索算法模拟布谷鸟寄生生育雏行为, 通过 Lévy flights 随机游动和偏好随机游动来平衡局部开发和全局探索^[10].

上述这些群集智能算法都是通过对现实中的某个群体行为的模拟来实现智能优化, 正因为此, 无法与算法核心完全吻合, 即开始时合理进行全局探索和局部寻优以快速收敛到可行解, 并及时判断是否陷入局部优化及执行相应跳出机制. 同时, 算法应具有尽量少的参数以保持简洁性. 因此, 我们提出一个从群集智能算法核心出发的模拟羊群行为的新型群集智能算法, 简单高效, 容易实现.

3 羊群算法

羊群算法主要模拟羊群觅食行为. 在用来求解优化问题时, 每一只羊都是一个可以自由移动的个体, 若干只羊组成一个羊群. 在开始时, 羊群随机分布在搜索

空间, 当前适应函数值最优的羊为头羊, 引导所有羊向其靠近, 即为头羊引领阶段; 为进一步增强羊群的移动性, 每只羊还会随机选择另一只羊进行比较, 如果后者更好, 向其移动, 否则远离它, 即为羊群互动阶段; 为避免羊群陷入某个局部优化解, 特设定当本代头羊与上一代头羊的差值小于阈值时, 引入牧羊犬监督机制, 即部分羊的位置被随机重置来跳出局部优化以寻找全局最优解, 即为牧羊犬监督阶段^[11]. 下面以极小化问题为例, 分别给出三个阶段的算法描述.

3.1 头羊引领

每只羊向头羊移动的行为对应全局探索机制, 为保证搜索性能, 只有在移动后的性能变好才更新, 否则放弃本次更新.

算法 1 头羊引领算法

```

Input:  $X^{\text{old}}$ ; Output:  $X^{\text{new}}$ .
1: for each  $x_i^{\text{old}} \in X^{\text{old}}$  do
2:    $x_i^{\text{new}} = x_i^{\text{old}} + \text{rand}(0, 1) \times (x_{\text{bellwether}} - x_i^{\text{old}})$ ;
3:   if  $x_i^{\text{new}} > x_i^{\text{old}}$  then
4:      $x_i^{\text{new}} = x_i^{\text{old}}$ ;
5:   end if
6: end for
7: return  $X^{\text{new}}$ ;

```

算法 1 中的 X^{old} 和 X^{new} 分别表示执行头羊引领前与后的羊群, $x_{\text{bellwether}}$ 表示头羊, x_i^{old} 和 x_i^{new} 分别表示第 i 只羊向头羊移动前和移动后的信息, $\text{rand}(0, 1)$ 表示 $[0, 1]$ 间的一个随机数, 以随机设置本次移动过程中羊步伐. lines 3~5 表示如果本次移动羊性能没有变好, 则不更新.

3.2 羊群互动

羊群互动行为对应局部开发机制, 每只羊 x_i 会与随机选定的另外一只羊 x_j 进行互动策略, 如果 x_i 优于 x_j , 则 x_i 远离 x_j , x_j 靠近 x_i ; 否则, 执行相反操作. 同样为保证搜索性能, 在两只羊互动后, 比较执行前后的值, 如果变好则更新, 否则放弃本次更新.

算法 2 羊群互动算法

```

Input:  $X^{\text{old}}$ ; Output:  $X^{\text{new}}$ .
1: for each  $x_i^{\text{old}} \in X^{\text{old}}$  do
2:   choose a  $x_j^{\text{old}} ((x_j^{\text{old}} \in X^{\text{old}}) \cap (x_i^{\text{old}} \neq x_j^{\text{old}}))$ ;
3:   if  $x_i^{\text{old}} < x_j^{\text{old}}$  then
4:      $x_i^{\text{new}} = x_i^{\text{old}} + \text{rand}(0, 1) \times (x_i^{\text{old}} - x_j^{\text{old}})$ ;
5:      $x_j^{\text{new}} = x_j^{\text{old}} + \text{rand}(0, 1) \times (x_i^{\text{old}} - x_j^{\text{old}})$ ;
6:   else
7:      $x_i^{\text{new}} = x_i^{\text{old}} + \text{rand}(0, 1) \times (x_j^{\text{old}} - x_i^{\text{old}})$ ;
8:      $x_j^{\text{new}} = x_j^{\text{old}} + \text{rand}(0, 1) \times (x_j^{\text{old}} - x_i^{\text{old}})$ ;
9:   end if

```

```

10:   if  $x_i^{\text{new}} > x_i^{\text{old}}$  then
11:        $x_i^{\text{new}} = x_i^{\text{old}}$ ;
12:   end if
13:   if  $x_j^{\text{new}} > x_j^{\text{old}}$  then
14:        $x_j^{\text{new}} = x_j^{\text{old}}$ ;
15:   end if
16: end for
17: return  $X^{\text{new}}$ ;

```

算法 2 中的 X^{old} 和 X^{new} 分别表示执行羊群互动前与后的羊群, line 3 ~ 9 表示两羊之间性能较差的向性能较好的移动, line 10 ~ 15 表示如果移动后性能没有变好, 则不更新。

3.3 牧羊犬监督

当羊群陷入某个局部优化解, 即本代头羊与上一代头羊的差值小于一个阈值时, 牧羊犬监督机制被引入来跳出局部优化。显然, 本阶段有两个参数需要合理设置: 判断是否陷入局部优化解的头羊差阈值 ε 和被重置的羊数量 (我们将其转化为每只羊被选中重置的概率 p)。显然, 如果 ε 设置较大, 则阈值要求易被满足, 羊群易被重置, 算法跳出局部优化解的概率相应增加, 但会导致局部开发能力较弱; 反之则羊群不易被重置, 算法跳出局部优化解的概率下降, 但局部开发能力被强化。同样地, 如果 p 设置较大, 则算法跳出局部优化解的概率增加, 但局部开发能力较弱; 反之则算法跳出局部优化解的概率下降, 但局部开发能力较强。

算法 3 牧羊犬监督算法

```

Input:  $X^{\text{old}}, \varepsilon, p$ ; Output:  $X^{\text{new}}$ .
1:   if  $|x_{\text{bellwether}}^{\text{new}} - x_{\text{bellwether}}^{\text{old}}| < \varepsilon$  then
2:       for each  $x_i \in X^{\text{old}}$  do
3:           if  $x_i \neq x_{\text{bellwether}}^{\text{new}}$  then
4:               if  $\text{rand}(0, 1) < p$  then
5:                    $x_i$  is herded by shepherd;
6:               end if
7:           end if
8:       end for
9:   end if
10:  for each  $x_i \in X^{\text{old}}$  do
11:      if  $x_i$  is not herded by shepherd then
12:           $x_i^{\text{new}} = x_i^{\text{old}} + \text{rand}(0, 1) \times (x_j^{\text{new}} - x_i^{\text{old}})$ ;
13:          if  $x_i^{\text{new}} > x_i^{\text{old}}$  then
14:               $x_i^{\text{new}} = x_i^{\text{old}}$ ;
15:          end if
16:      end if
17:  end for
18:  return  $X^{\text{new}}$ ;

```

算法 3 中的 X^{old} 和 X^{new} 分别表示执行牧羊犬监督前

与后的羊群; line 1 ~ 9 表示如果头羊差小于阈值, 除领头羊外的每只羊都按重置概率被牧羊犬放牧, 即重新被初始化; line 10 ~ 17 表示对于没有被放牧即没有被重新初始化的每只羊, 随机选择一只被放牧后的羊 x_j^{new} , 并向其移动, 同样, 如果移动后效果不好, 则不更新。

3.4 羊群算法流程

通过上述对算法概念、思想和参数的讲解, 下面给出羊群算法的实现步骤:

- Step 1. 初始化算法中的相关参数, 如羊群规模、问题维度等;
- Step 2. 在搜索空间中随机初始化每只羊的位置;
- Step 3. while 算法终止条件不满足 / * 未满足误差阈值或未达到最大迭代次数 * /
 - 根据算法 1 执行头羊引领过程;
 - 根据算法 2 执行羊群互动过程;
 - 根据算法 3 执行牧羊犬监督过程;
 end while
- Step 4. 输出最优函数值及相应位置。

4 实验与结果分析

为了能够详细分析 SO 算法在求解函数优化问题时的性能, 我们从解质量、收敛速度和稳定性三个方面对 SO 算法与标准 PSO 算法进行比较与分析。实验应用了如表 1 所示的 8 个典型基准测试函数, 函数维度为 30, 种群规模大小为 40, 每个算法独立运行 50 次, 单峰函数 ($f_1 - f_3$) 和多峰函数 ($f_4 - f_8$) 的最大迭代次数分别为 10000 和 100000。

表 1 基准测试函数

函数	f_1	f_2	f_3	f_4
函数名	Sphere	Shifted Sphere	Axis Parallel Hyper-ellipsoid	Rastrigin
函数	f_5	f_6	f_7	f_8
函数名	Shifted Rastrigin	Shifted Rotated Rastrigin	Ackley	Generalize Penalized

4.1 参数分析

参数的合理设置对于群集智能算法的性能有很大影响。因此, 我们首先进行参数分析。为简化算法和提高健壮性, 可以将头羊差阈值 ε 设置为测试函数的误差阈值。从而只需分析牧羊犬监督机制中的重置概率 p 。为验证参数对算法性能的影响, 我们对不同 p 值进行了测试, 结果显示没有一个固定 p 值能使得 SO 算法在每个函数上都取得最优解, 总体来说, 对于单峰函数, p 越小, 则取得的性能越好, 因为单峰函数比较容易收敛到全局最优, 所以牧羊犬监督机制对其重要性相对较低。由于 $p = 0.2$ 时, 算法整体性能较好, 我们进一步对不同阈值 ε 进行测试, 结果显示不同 ε 对羊群算法性能影响

较小,因为虽然较大 ε 更易被满足,从而使得牧羊犬监督机制更多地被执行.但羊群算法都会进入牧羊犬监督机制,只是进入早晚有别,例如函数 f_1 在 5 个不同 ε ($10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}$) 下,分别平均在第 455、487、514、520 和 536 次迭代时满足问题的解阈值要求,执行牧羊犬监督机制的次数分别为 9723、9664、9603、9554 和 9505.显然,不同阈值 ε 的执行次数占总迭代次数的比例差别非常小.从参数分析的结果来看,本算法受参数影响较小,比较健壮.在下面的实验中,我们设定 $p=0.2, \varepsilon$ 为测试函数的误差阈值.

4.2 解质量分析

我们对算法的解质量进行比较分析,结果如表 2 所示.对于单峰函数,SO 算法的性能均明显优于标准 PSO 算法;对于多峰函数,SO 算法在函数 f_4 上取得了全局最优值,解质量远优于 PSO,在函数 f_5 和 f_8 上的解质量远优于 PSO,在函数 f_6 和 f_7 上的解质量与 PSO 相似.整体上看,SO 算法能够取得比 PSO 算法更高质量解的主要原因是其头羊引领机制能够快速引领全体向当前已知的最优解积极靠近,强化全局探索能力,羊群互动机制加快羊个体向周围的较优解靠近,并在快速收敛的同时积极判断是否陷入局部优化解和执行相应跳出机制.

表 2 SO 和 PSO 算法的解质量比较

函数	算法	均值	标准差	最优值	最差值
f_1	SO	1.56E-19	1.35E-19	1.83E-20	5.82E-19
	PSO	1.54E-18	2.99E-18	3.05E-21	1.62E-17
f_2	SO	1.50E-13	3.59E-14	1.14E-13	2.27E-13
	PSO	1.70E+02	2.87E+02	5.68E-14	1.50E+03
f_3	SO	9.38E-21	1.38E-20	5.24E-22	8.92E-20
	PSO	2.45E-20	4.14E-20	4.12E-22	1.75E-19
f_4	SO	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	PSO	3.24E+01	1.07E+01	1.69E+01	5.77E+01
f_5	SO	1.02E-13	2.57E-14	5.68E-14	1.71E-13
	PSO	1.63E+02	2.69E+01	8.16E+01	2.23E+02
f_6	SO	1.18E+02	3.09E+01	5.71E+01	1.82E+02
	PSO	5.90E+02	2.92E+02	2.33E+02	1.61E+03
f_7	SO	7.25E-15	1.00E-15	7.11E-15	1.42E-14
	PSO	7.39E-15	1.41E-15	7.11E-15	1.42E-14
f_8	SO	1.02E-29	7.53E-30	9.84E-31	2.85E-29
	PSO	2.93E-01	5.67E-01	1.57E-32	3.43E+00

4.3 收敛速度分析

我们对算法的收敛速度进行分析,具体为如果算法运行结果满足测试函数的误差阈值,则表示本次运行成功,记录其迭代次数,否则视为失败,只记录最终迭代次数.结果显示,除了带偏移旋转的多峰函数 f_6 ,SO 算法在其它 7 个函数上能够 100% 地收敛于给定误差阈值,而 PSO 算法只能在比较简单的函数 f_1, f_3 和 f_7 上达到 100% 收敛.主要原因是牧羊犬监督机制能够使得

SO 算法及时跳出局部最优解.

为更直观地展现算法在各个测试函数上的收敛情况,我们选取结果与 50 次测试平均值最为接近的一次测试,绘制算法的收敛曲线,结果如图 1 所示.可以看出,对于几乎所有函数,SO 算法都能在运行初期即快速收敛,取得相对满意的,远优于 PSO 算法的优化解.主要原因是 SO 算法的头羊引领和羊群互动机制快速引领全体向每轮迭代的最优解和较优解积极靠近,实现快速收敛.

4.4 稳定性分析

由于群集智能算法具有概率性,因此,稳定性是其一项重要需求.从图 2 可以看出,SO 算法的结果都更集中,且离散点更少,说明其具有更好的稳定性.

4.5 维度扩展性分析

测试函数的维度对于智能算法具有重要影响.表 3 和表 4 分别表示 SO 算法和 PSO 算法在 10 维和 50 维测试函数上的性能.可以看出,无论是 10 维和 50 维,对于大部分函数,SO 算法的结果都显著优于 PSO 算法.具体来说,对于单峰函数,在 10 维时,由于问题复杂度降低,SO 算法和 PSO 算法都能找到全局最优解,由于 SO 算法的头羊差阈值设定为测试函数的误差阈值,因此其求解精度低于 PSO 算法,但也已完全符合精度要求;在 50 维时,由于问题复杂化,PSO 算法在带偏移的单峰函数 f_2 上只有 4% 的概率能够满足阈值要求,而 SO 算法一直能够取得全局最优解,当然随着维度的增加,SO 算法在偏移函数如 f_3 上的性能也有所下降.

表 3 SO 和 PSO 算法的性能比较 (DIM=10)

函数	SO 算法		PSO 算法	
	均值 \pm 标准差	成功率 (%)	均值 \pm 标准差	成功率 (%)
f_1	2.87E-26 \pm 3.76E-26	100	1.06E-72 \pm 3.29E-72	100
f_2	2.16E-14 \pm 2.79E-14	100	1.14E-15 \pm 8.04E-15	100
f_3	5.30E-28 \pm 7.56E-28	100	4.37E-74 \pm 2.14E-73	100
f_4	0.00E+00 \pm 0.00E+00	100	1.13E+01 \pm 5.13E+00	0
f_5	0.00E+00 \pm 0.00E+00	100	2.69E+01 \pm 9.62E+00	0
f_6	1.31E+01 \pm 5.80E+00	0	5.48E+01 \pm 3.63E+01	0
f_7	3.13E-15 \pm 1.17E-15	100	3.55E-15 \pm 7.97E-31	100
f_8	4.71E-32 \pm 1.11E-47	100	3.11E-01 \pm 5.94E-01	66

表 4 SO 和 PSO 算法的性能比较 (DIM=50)

函数	SO 算法		PSO 算法	
	均值 \pm 标准差	成功率 (%)	均值 \pm 标准差	成功率 (%)
f_1	7.58E-17 \pm 4.00E-17	100	1.55E-14 \pm 3.50E-15	100
f_2	3.32E-13 \pm 5.92E-14	100	9.34E+02 \pm 8.10E+02	4
f_3	8.02E-18 \pm 5.41E-18	100	2.66E-10 \pm 3.32E-10	100
f_4	0.00E+00 \pm 0.00E+00	100	4.89E+01 \pm 1.37E+01	0
f_5	9.95E-01 \pm 1.10E+00	44	3.41E+02 \pm 3.41E+01	0
f_6	3.34E+02 \pm 9.90E+01	0	1.44E+03 \pm 5.87E+02	0
f_7	1.61E-14 \pm 3.15E-15	100	1.55E-14 \pm 3.50E-15	100
f_8	4.49E-27 \pm 3.04E-27	100	7.59E-02 \pm 1.14E-01	54

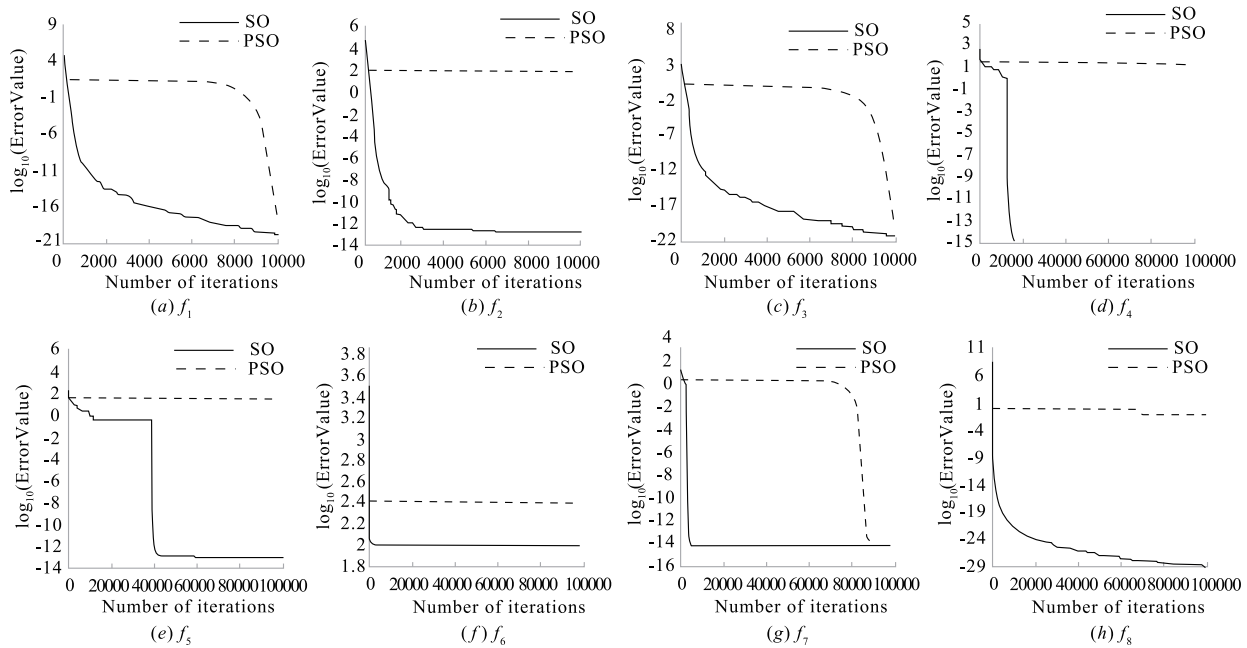


图1 SO和PSO算法的收敛曲线比较

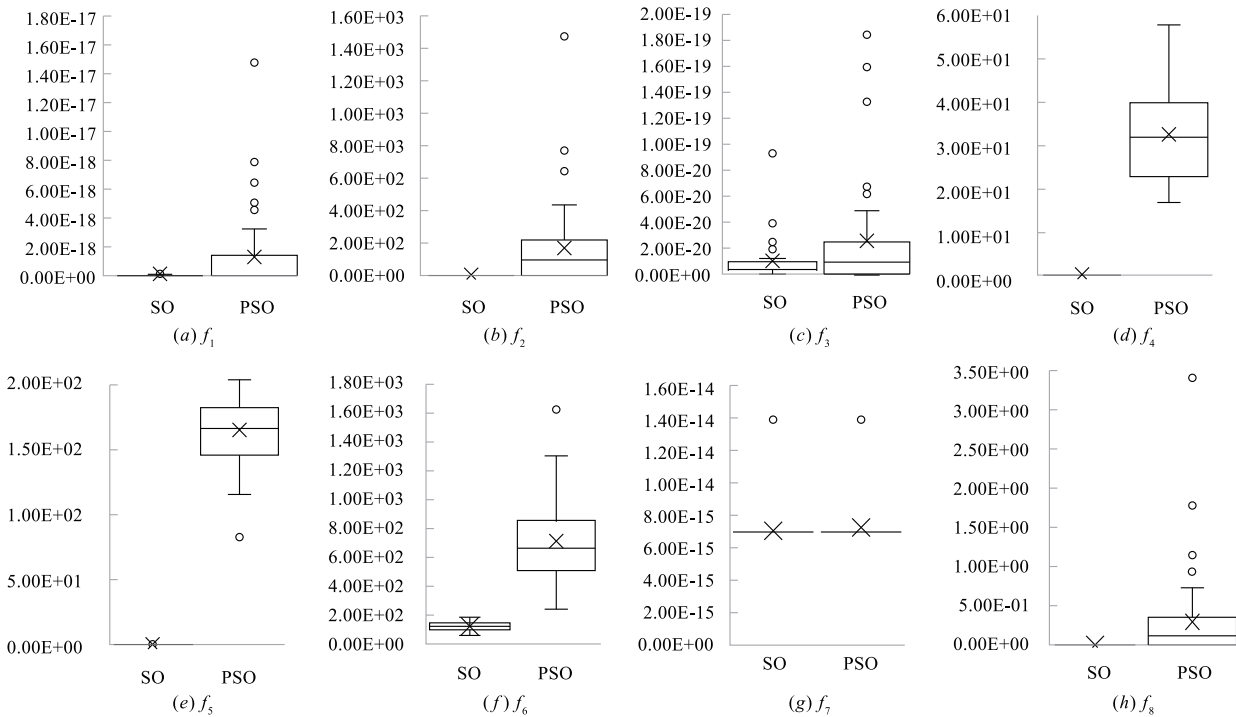


图2 SO和PSO算法的50次测试结果分布盒图比较

5 总结

本文从群集智能算法核心出发,通过模拟羊群行为,提出了一种新的群集智能算法——羊群算法.该算法通过模拟领头羊引领羊群实现快速全局探索,使得羊群快速向已知全局优化解靠近;通过羊群之间的互相移动来实现局部开发,进一步加快收敛速度;应用牧羊

犬监督机制判断是否进入局部优化并快速跳出局部优化,从而寻找全局最优解.

在未来的工作中,我们一方面将 SO 算法应用于实际工作中,通过任务分配、入侵检测和路径规划等实际工作来进一步检验算法的性能;另一方面将对算法中全局探索和局部开发的均衡,计算开销和搜索精度的均衡等问题进行分析和改进,但注意在提高性能的同

时维持其简单性.

参考文献

- [1] Zhang Y, Wang S, Ji G. A comprehensive survey on particle swarm optimization algorithm and its applications[J]. *Mathematical Problems in Engineering*, 2015, 1: 1 - 38.
- [2] Krause J, Cordeiro J, Parpinelli RS, Lopes HS. A survey of swarm algorithms applied to discrete optimization problems [J]. *Swarm Intelligence and Bio-Inspired Computation*, 2013, 4(9): 169 - 191.
- [3] Parpinelli RS, Lopes HS. New inspirations in swarm intelligence: a survey [J]. *International Journal of Bio-Inspired Computation*, 2011, 3(1): 1 - 16.
- [4] Pellegrini P, Stützle T, Birattari M. A critical analysis of parameter adaptation in ant colony optimization[J]. *Swarm Intelligence*, 2012, 6(1): 23 - 48.
- [5] Cai X, Cui Z, Zeng J, Tan Y. Particle swarm optimization with self-adjusting cognitive selection strategy[J]. *International Journal of Innovative Computing, Information and Control*, 2008, 4(4): 943 - 952.
- [6] Davoodi E, Hagh MT, Zadeh SG. A hybrid improved quantum-behaved particle swarm optimization-simplex method (IQPSOS) to solve power system load flow problems[J]. *Applied Soft Computing*, 2014, 21: 171 - 179.
- [7] 喻飞, 李元香, 魏波, 等. 透镜成像反学习策略在粒子群算法中的应用[J]. *电子学报*, 2014, 42(2): 230 - 235.
Yu F, Li Y X, Wei B, et al. The application of a novel OBL based on lens imaging principle in PSO[J]. *Acta Electronica Sinica*, 2014, 42(2): 230 - 235. (in Chinese)
- [8] Dorigo M, Blum C. Ant colony optimization theory: a survey[J]. *Theoretical Computer Science*, 2005, 344 (2-3): 243 - 278.
- [9] Karaboga D, Gorkemli B, Ozturk C, Karaboga N. A comprehensive survey: artificial bee colony (ABC) algorithm and applications [J]. *Artificial Intelligence Review*, 2014, 42(1): 21 - 57.
- [10] Yang XS, Deb S. Engineering optimisation by cuckoo search[J]. *International Journal of Mathematical Modelling and Numerical Optimisation*, 2010, 1(4): 330 - 343.
- [11] Meuret M, Provenza FD. When art and science meet: integrating knowledge of French herders with science of foraging behavior [J]. *Rangeland Ecology & Management*, 2015, 68(1): 1 - 17.

作者简介



曲大鹏 男, 博士, 1981 年生于辽宁鞍山, 现为辽宁大学信息学院副教授, 硕士生导师. 主要研究方向为未来互联网, 智能计算, 网络经济学.

E-mail: dapengqu@lnu.edu.cn



许伦祥 男, 1995 年生于湖北仙桃, 现为辽宁大学信息学院硕士生, 主要研究方向为计算机网络, 智能计算.

E-mail: xulunxiang@outlook.com



鲁杲光 男, 1995 年生于辽宁铁岭, 现为东北大学计算机科学与工程学院硕士生, 主要研究方向为计算机网络.

E-mail: yaoguang0228@outlook.com



原晓坤 男, 1995 年出生于陕西蒲城, 现为辽宁大学信息学院本科生, 主要研究方向为计算机网络, 智能计算.

E-mail: yuanxiaokun@outlook.com

黄敏 女, 博士, 1968 年生于福建长乐, 现为东北大学信息科学与工程学院教师, 博士生导师. 主要研究领域为智能算法设计与优化, 调度理论与方法.

E-mail: huangmin@mail.neu.edu.cn

王兴伟 (通信作者) 男, 博士, 1968 年生于内蒙古包头, 现为东北大学软件学院院长, 教授, 博士生导师. 主要研究领域为未来互联网, 云计算, 网络空间安全.

E-mail: wangxw@mail.neu.edu.cn