引入改进鸽群搜索算子的粒子群优化算法

马 龙[1] 卢才武[1] 顾清华[1] 阮顺领[1]

摘 要 随着迭代计算过程的推进,标准粒子群算法后期容易出现收敛速度较慢、精度较低、早熟及开发探索能力较弱等问题.针对上述不足,文中提出引入改进鸽群搜索算子的粒子群优化算法,采用 Beta 反向学习策略进行种群的初始化,实现种群粒子分布的多样性.首先采用线性和非线性变异策略改进鸽群算法中的地图罗盘算子,提高鸽群算法的开发和探索能力.然后采用改进后的组合优化鸽群算子更新粒子群中粒子位置和速度,提高收敛速度和计算精度,避免算法陷入局部最优.实验表明,文中算法提高收敛计算速度,精度达到函数设定的理想值.

关键词 改进鸽群搜索算子,粒子群优化,Beta 分布函数,反向学习策略

引用格式 马 龙,卢才武,顾清华,阮顺领.引入改进鸽群搜索算子的粒子群优化算法.模式识别与人工智能,2018,31(10):909-920.

DOI 10.16451/j.cnki.issn1003-6059.201810005 **中图法分类号** TP 18

Particle Swarm Optimization with Search Operator of Improved Pigeon-Inspired Algorithm

MA Long[1], LU Caiwu[1], GU Qinghua[1], RUAN Shunling[1]

ABSTRACT The standard particle swarm optimization is easy to have problems of low convergence speed and precision, prematurity and poor exploring ability during later period. Aiming at these problems, an optimized particle swarm optimization based on improved pigeon search operator is proposed. Population initialization is determined by Beta opposition-based learning strategy, and the diversity of population distribution is realized. The map compass operator is improved by the linear and nonlinear mutation strategy to improve the development and the exploration ability of the pigeon-inspired algorithm. Then, the location and the speed are updated by the improved combination optimization operator to speed up the convergence, enhance the precision and avoid falling into local optimal solution in the particle swarm optimization. Simulation experimental results show that the convergence speed is improved by IPSO, and the accuracy reaches the ideal value set by the functions.

Key Words Improved Pigeon Swarm Search Operator, Particle Swarm Optimization, Beta Distribution Function, Opposition-Based Learning Strategy

Citation MA L, LU C W, GU Q H, RUAN S L. Particle Swarm Optimization with Search Operator of Improved Pigeon-Inspired Algorithm. Pattern Recognition and Artificial Intelligence, 2018, 31(10): 909-920.

收稿日期: 2018-05-22; **录用日期:** 2018-09-28

Manuscript received May 22, 2018;
accepted September 28, 2018

国家自然科学基金面上项目(No. 51774228)、**国家安全生产重特大事故防治关键技术科技项目**(No. 0020-2018AQ)、**陕西省社会科学基金项目**(No. Z20180285)、**陕西省教育厅专项计划项目**(No. 17JK0425) **资助**

Supported by National Natural Science Foundation of China(No. 51774228), National Safety Production Technology Foundation of Major Accidents Prevention Key Technology of China(No. 0020-2018AQ), Social Science Foundation of Shannxi Province(No. Z20180285), Foundation of Shannxi Educational Committee(No. 17JK0425)

本文责任编委 付俊

Recommended by Associate Editor FU Jun

1. **西安建筑科技大学 管理学院 西安** 710055

1. School of Management, Xi´an University of Architecture and Technology, Xi´an 710055

受自然界鸟群捕食行为的启发,Kennedy 等[1]提出粒子群算法(Particle Swarm Optimization,PSO).PSO 具有原理简单和容易实现等特点[2],自提出后众多学者从算法改进和工程应用方面展开深入研究. Nichabadi 等[3]在研究惯性权重策略的基础上,采用群体成功率作为反馈参数,确定粒子的搜索状态,但该种群后期的聚集度较低,全局搜索能力较弱. Marinakis 等[4]采用变邻域搜索策略优化粒子的搜索位置,解决约束最短路径问题,但搜索精度较低,计算时间较长. 于颖等[5]采用惩罚因子的确定策略,对波纹管工程实例中的非线性约束离散变量进行优化设计,波纹管的单位重量比在用产品提高79%左右,但计算精度较差. 万春秋等[6]采用动态评价方法改进粒子群算法中惩罚因子选取困难问题,解决风电场微观选址优化问题,但算法呈现出随机误差较大的问题. 张国富等[7]采用动态区域搜索方法和自适应扰动策略分别提高算法的全局收敛性能和增强局部最优的能力,由此解决多层非线性规划问题,但计算速度较慢. Du 等[8]使用异构信息策略,解决算法在后期的收敛速度较慢的问题,但在滤波器设计上求解精度较低. Sakthivel 等[9]使用混沌序列对种群进行初始化,解决算法后期收敛速度较慢和早熟的问题,但在三相异步电机参数优化上的精度较差. Chen 等[10]使用质数和指数关系权重策略,解决算法容易陷入局部最优和精确性较差的问题,但改进后的算法无法达到均衡的开发和探索能力. Zavala 等[11]使用修正的环邻域扰动算子,保证种群搜索的多样性和算法的探索能力,但单向环邻域结构降低算法的收敛速度,在优化线性和非线性问题时,算法耗时过长.

上述理论研究和工程应用虽然可以充分体现PSO的优越性,但在解决实际的复杂函数问题时,PSO既无法同时避免后期的收敛速度较慢、容易早熟收敛、计算精度较低等问题,又无法均衡算法的开发和探索能力.

受鸽子归巢行为启发,Duan 等[12]提出自主归巢行为的优化算法——鸽群算法(Pigeon-Inspired Optimization,PIO),算法具有明确的搜索方向和快速的收敛速度、计算精度,理论和应用研究的成果较多. Bolaji 等[13]提出二进制编码的鸽群算法,解决多维背包求解问题. Li 等[14]提出 Bloch 球面坐标编码的量子鸽群算法,快速解决连续函数优化问题. Zhang 等[15]提出捕食逃逸策略的鸽群算法,解决无人机三维路径优化问题. Zhang 等[16]提出集中检测方案的改进鸽群算法,拟提高全球卫星集中检测精度.这些理论和应用成果均从不同角度展现鸽群算法的求解速度、计算精度等优势,并对鸽群算法进行大量改进工作.

综上可知,虽然现有的 PSO 在改进和应用方面进行大量工作,但能使算法同时满足收敛速度、计算精度、早熟收敛及均衡开发探索能力的研究还不够完善.因此,本文提出引入改进鸽群搜索算子的粒子群优化算法(Particle Swarm Optimization with Search Operator of Improved Pigeon-Inspired Algorithm,SOIPPSO).基于文献[17]和文献[18]利用 Beta 分布和反向学习融合策略以改进初始种群的思想,提出粒子种群初始化策略.以文献[12]为基础,首先使用线性变异策略和非线性变异策略对地图罗盘因子进行改进,然后对两个独立运行阶段的鸽群算子进行组合,最后使用改进后的鸽群组合搜索新算子对粒子的位置和速度进行更新.算法充分利用标准鸽群算法的快速收敛速度和计算精度,改善粒子群算法存在的不足.同时,算法采用线性变异和非线性变异策略实现粒子群算法求解复杂函数问题的能力,平衡算法的开发和探索能力.

1 基本算法

1.1 标准粒子群算法原理

PSO 作为成熟的仿生进化算法,在迭代计算初期,算法的种群采用随机方式进行初始化,导致每个粒子在 D 维空间呈现随机分布状态,完成粒子的速度和位置的调整方式更多的是以历史经验为主.设种群中第 i 个粒子的位置为

$$\boldsymbol{x}_i^k = [x_{i1}^k, x_{i2}^k, \cdots, x_{iD}^k]^{\mathrm{T}},$$

经过 k 次迭代计算后,个体粒子的最佳位置为

$$\boldsymbol{p}_i^k = (p_{i1}^k, p_{i2}^k, \cdots, p_{iD}^k);$$

群体粒子的最佳位置为

$$\boldsymbol{p}_{gbest}^k = (p_{gbest1}^k, p_{gbest2}^k, \cdots, p_{gbestD}^k),$$

第 i 个粒子的速度表示为

$$\boldsymbol{v}_i^k = [v_{i1}^k, v_{i2}^k, \cdots, v_{iD}^k]^{\mathrm{T}},$$

经过每代的迭代计算,在第 k 时刻,D 维空间上第 i 个粒子的位置和速度计算表达如下[19]:

$$v_{i,d}^{k+1} = v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (p_{gbest,d}^k - x_{id}^k), \quad (1)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^k, \quad (2)$$

$$i = 1, 2, \cdots, S, d = 1, 2, \cdots, D,$$

其中,$r_1 \in [0,1]$,$r_2 \in [0,1]$,c_1、c_2 分别表示粒子的加速度参数,可使种群内单个粒子不断向 $\boldsymbol{p}_i^k$ 及 $\boldsymbol{p}_{gbest}^k$ 的位置加速移动.

在 PSO 中,若在第 d 维空间中粒子的移动速度超过设定的最大迭代速度 $v_{\max,d}^k$ 时,粒子飞出最优解范围,收敛能力变弱. 另外,算法的迭代速度与解的精度相关,$v_{\max,d}^k$ 的值越大,算法结果的精度越低,$v_{\max,d}^k$ 的值越小,算法结果的精度越高,则算法的全局搜索能力也会变弱. 基于上述不足,Shi 等[2] 在粒子的速度中引入惯性权重 ω,对算法进行改进:

$$v_{i,d}^{k+1} = \omega v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (p_{gbest,d}^k - x_{id}^k),$$

其中: ω 为惯性权重; 经过 k 次迭代计算后,惯性权重 ω 可表示为

$$\omega_k = (\omega_{\text{start}} - \omega_{\text{end}}) \left(\frac{v_{\max}^k - k}{v_{\max}^k} \right) + \omega_{\text{end}},$$

其中: ω_{start}、ω_{end} 分别表示惯性权重的最初值、最终值,$v_{\max}^k$ 表示算法的最大迭代次数.

1.2　改进的鸽群进化算法

鸽群算法主要由地图罗盘算子和地标算子组成,这两个算子处于独立运行阶段,即先在地图罗盘算子阶段对鸽子的位置和速度进行初始化,并迭代更新,然后在地标算子阶段,每次迭代都会使鸽子数量减少一半,舍弃远离目标的鸽子,对于剩余中心位置的鸽子作为飞行参考方向. 该算法经过两个独立阶段的迭代计算后,具有全局搜索能力较强、求解计算速度较快和避免早熟收敛等特点. 为了便于描述本文算法的特点,对基本的鸽群进化算法参考文献[12],下面只介绍鸽群算法中改进的地图罗盘算子.

1.2.1　改进的地图罗盘算子

鸽群算法中的地图罗盘因子 R 为均衡算法搜索速度和开发能力的关键因子. 文献[20]的研究结果表明 R 越小,搜索能力越强,开发潜力越大. 为了解决两者的均衡发展问题,本文采用线性变异策略[20]与非线性变异策略[21]分别对地图罗盘因子 R 进行动态改进:

$$R = \left(R_{\min} + R_{\max} \frac{n}{t_{\max}} \right) (1 + p_r (rand - 1)),$$

其中,$R_{\max}$ 表示最大的地图罗盘因子值,$R_{\min}$ 表示最小的地图罗盘因子值,p_r 表示变异概率. 本文还是使用非线性变异策略对 R 进行动态改进:

$$R = \left(R_{\min} + R_{\max} \frac{4n^{\frac{3}{2}}}{3t_{\max}\sqrt{\pi}} \right) (1 + p_r (rand - 1)),$$

其中,$R_{\min}$、$R_{\max}$、p_r 的含义与上式相同,$rand()$ 表示在 0 ~ 1 之间取值的随机数,分数部分表达探索适合地图罗盘因子 R 非线性变量的方法.

1.2.2　鸽群算法的组合搜索新算子

在 PSO 中,p_{id}^k 表示第 i 个粒子历史最优的第 d 维位置值. 经过 k 次迭代运算后,不同粒子的当前最优位置值会逐渐形成历史最优位置值,这些历史最优位置值关联后形成一条局部最优位置值的轨迹链. 当局部搜索区域聚集的粒子数较多时,容易出现早熟现象,即此时算法的搜索能力减弱. 因此,如何同时提高算法的开发和探索能力,及算法快速逃逸局部最优的能力,这是优化算法性能的关键. 针对上述问题,本文组合改进的地图罗盘算子与地标算子,然后对 PSO 的位置和速度进行更新计算. 另外,文献[22]和文献[23]也说明该算法对复杂多峰函数和在多目标优化问题方面具有较好的性能. 这主要取决 Qiu 等[23] 受 PSO 的启发,提出 PIO 组合搜索算子:

$$\begin{aligned} z_{id}^t = {} & v_{id}^{t-1} e^{-R \cdot t} - \\ & \delta \left[(rand_2 - rand_1) \lg_{t_{\max}}^t + rand_1 \right] x_{id}^{t-1} + \\ & rand_1 \delta (1 - \lg_{t_{\max}}^t) x_{gbestd} + rand_2 \delta \lg_{t_{\max}}^t x_{centerd}^{t-1}, \end{aligned} \tag{3}$$

$$\begin{aligned} z_{id}^t = {} & v_{id}^{t-1} e^{-R \cdot t} + \\ & \{1 - \delta \left[(rand_2 - rand_1) \lg_{t_{\max}}^t + rand_1 \right] \} x_{id}^{t-1} + \\ & rand_1 \delta (1 - \lg_{t_{\max}}^t) x_{gbestd} + rand_2 \delta \lg_{t_{\max}}^t x_{centerd}^{t-1}. \end{aligned} \tag{4}$$

其中: δ 表示过渡因子,在 0 ~ 1 之间设定取值; $rand_1()$、$rand_2()$ 分别为 0 ~ 1 之间的随机数; v_{id}^{t-1} 表示在时刻 $t-1$ 第 i 只鸽子的速度; x_{id}^{t-1} 表示在时刻 $t-1$ 第 i 只鸽子的位置; $x_{centerd}^{t-1}$ 表示在时刻 $t-1$ 折半保留下的鸽子的中心坐标位置; R 表示地图罗盘因子.

为了使改进后的粒子群算法更好地解决复杂的多维单峰、多峰函数及线性和非线性函数问题,本文将改进后的地图罗盘因子 R 与 PIO 组合搜索算子进行有效组合,提出改进鸽群组合搜索算子:

$$\begin{aligned} z_{id}^t = {} & v_{id}^{t-1} \exp \left(- \left(R_{\min} + R_{\max} \frac{n}{t_{\max}} \right) (1 + p_r (rand - 1)) t \right) - \\ & \delta \left[(rand_2 - rand_1) \lg_{t_{\max}}^t + rand_1 \right] x_{id}^{t-1} + rand_1 \delta (1 - \lg_{t_{\max}}^t) x_{gbestd} + rand_2 \delta \lg_{t_{\max}}^t x_{centerd}^{t-1}, \end{aligned} \tag{5}$$

$$z_{id}^{t}=v_{id}^{t-1}\exp\left(-\left(R_{\min}+R_{\max}\frac{4n^{\frac{3}{2}}}{3t_{\max}\sqrt{\pi}}\right)(1+p_r(rand-1))t\right)+$$
$$\{1-\delta[(rand_2-rand_1)\lg_{t_{\max}}^{t}+rand_1]\}x_{id}^{t-1}+rand_1\delta(1-\lg_{t_{\max}}^{t})x_{gbestd}+rand_2\delta\lg_{t_{\max}}^{t}x_{centerd}^{t-1}, \tag{6}$$

其中参数与变量取值与式(3)和式(4)一致.

2 改进鸽群搜索算子的粒子群优化算法

2.1 算法思想

传统粒子群算法主要采用随机方法产生种群的初始解,这是因为缺乏先验信息的指引而导致种群中初始粒子处于随机均匀状态,这种状态不利于种群边界上粒子向最优解靠拢,也不利于粒子向最优解快速形成合围态势. 然而,李建平等[17]提出Beta分布初始化策略,肖文显等[18]提出基于反向学习的初始化策略,均获得较好的效果. 本文充分利用这两种初始化种群策略的优势,提出Beta反向学习融合初始化种群策略,虽然综合文献[17]和文献[18]种群初始化策略的优势,但是主要区别在于:1)该策略对粒子群初始种群边界同时进行动态选取,避免单个策略选取导致个别优秀粒子的逃逸;2)利用反向学习策略拉近个体粒子向最优解靠近,再利用Beta分布加固最优解的包围状态. 初始化方法的具体步骤如下.

算法1 粒子群初始化方法

设置种群规模为S,优化空间为D维,候选解为$\boldsymbol{x}_i$;

Beta分布阶段

for $i=1$ to S do
  for $j=1$ to D do
    随机产生$\beta(a,b)\in(0,1)$
    $x_{ij}=x_{\min,j}+(x_{\max,j}-x_{\min,j})\beta rand(a,b,1,1)$
  end for
end for

反向学习阶段

for $i=1$ to S
  for $j=1$ to D do
    $OX_{ij}=x_{\min,j}+x_{\max,j}-x_{i,j}$
  end for
end for

将分布在$\{X(S)\cup OX(S)\}$中的多个粒子根据最佳适应度进行选择,并将适应度最好的S个粒子作为算法的初始种群.

2.2 算法步骤

在标准PSO中,种群的随机初始化使算法在后期寻优过程中收敛速度变慢,容易出现早熟等问题,这些问题严重影响算法的搜索计算性能. 本文算法的基本思路是:首先采用线性变异策略和非线性变异策略对鸽群算法中的地图罗盘因子R进行修正,然后将鸽群算法中分段执行的改进地图罗盘算子与地标算子进行优化组合,最后在标准PSO的迭代计算中,采用改进鸽群组合搜索的算子对粒子群算法中的速度和位置进行更新,使新的搜索算子快速搜索到整个粒子种群的历史最优位置$\boldsymbol{p}_g^k(k=1,2,\cdots,S)$,避免早熟问题. 同时,根据本文提出的种群初始化方法,综合考虑粒子群算法的收敛计算速度和全局收敛计算能力,有效提高粒子群算法的开发和探索能力. 算法的具体步骤如下.

算法2 改进鸽群搜索算子的粒子群优化算法

初始化. 设定认知参数c_1和社会参数c_2,
  惯性权重ω,最大函数评价次数

在初始化解空间中,采用Beta反向学习的初始化方法形成S个粒子构成初始种群$\boldsymbol{x}_i^k$和初始速度$\boldsymbol{v}_i^k$,$i=1,2,\cdots,S$

计算S个粒子的最佳适应度函数值,并更新各粒子的局部最优值$\boldsymbol{p}_i^k$和全局最优值$\boldsymbol{p}_g^k$

while 算法的终止条件不满足 do
  for $i=1$ to S do
    for $j=1$ to D do
      根据式(1)更新粒子的速度
      根据式(2)更新粒子的位置
    end for
    if $f(\boldsymbol{v}_i^k)<f(\boldsymbol{p}_i^k)$ do
      $\boldsymbol{p}_i^k=\boldsymbol{v}_i^k$
    end if
    for $i=1$ to S do
      for $d=1$ to D do
        采用式(5)或式(6)在p_{id}^k周围搜索候选解z_{gd}^k
        if $f(z_{id}^t)<f(p_{id}^k)$ do
          $p_{id}^k=z_{id}^t$; $x_{id}^k=z_{id}^t$
        end if
      end for

end for
if $f(\boldsymbol{p}_i^k) < f(\boldsymbol{p}_g^k)$ do
$\boldsymbol{p}_g^k = \boldsymbol{p}_i^k$
end if
end for
end while
输出最优值及最优解

注　本文将引入改进鸽群搜索算子的粒子群算法分别记为: SOIPPSO-1 表示采用搜索算子(5), SOIPPSO-2 表示采用搜索算子(6).

3　仿真实验及结果分析

3.1　测试函数

1) Sphere's 函数:

$$f_1 = \sum_{i=1}^{D} x_i^2,\ x_i \in (-5.12, 5.12).$$

Sphere 函数的类型为多维单峰函数,理想的收敛情况是在点 $\boldsymbol{x} = (0,0,\cdots,0)$ 处取得极小值 0.

2) Rosenbrock's 函数:

$$f_2 = \sum_{i=1}^{D} \left(100(x_i^2 - x_{i+1})^2 + (x_1 - 1)^2\right),$$

$$x_i \in (-2.048, 2.048).$$

Rosenbrock 函数的类型为多维病态二次函数,理想的收敛情况是在点 $\boldsymbol{x} = (1,1,\cdots,1)$ 处取得极小值 0.

3) Griewank's 函数:

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

$$x_i \in (-600, 600).$$

Griewank 函数作为多维多峰函数,理想的收敛情况是在点 $\boldsymbol{x} = (0,0,\cdots,0)$ 处取得极小值 0.

4) Ackley's 函数:

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

$$x_i \in (-32.7, 32.7).$$

Ackley 函数作为多维多峰函数,理想的收敛情况是在点 $\boldsymbol{x} = (0,0,\cdots,0)$ 处取得极小值 0.

5) Penalized1 函数:

$$f_5 = \frac{\pi}{D}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_D - 1)^2\right\} + \sum_{i=1}^{D}\mu(x_i, 10, 100, 4),$$

$$x_i \in (-50, 50),$$

$$\mu(x_i, \alpha, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leqslant x_i \leqslant a \\ -k(-x_i - a)^m, & x_i < a \end{cases}$$

$$y_i = 1 + \frac{1}{4}(x_i + 1).$$

Penalized1 函数的类型为多维多峰分段函数,理想的收敛情况是在点 $\boldsymbol{x} = (1,1,\cdots,1)$ 处取得极小值 0.

6) Penalized2 函数:

$$f_6 = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{D-1}(x_i - 1)^2\left[1 + \sin^2(3\pi x_{i+1})\right] + (x_D - 1)^2\left[1 + \sin^2(2\pi x_D)\right]\right\} + \sum_{i=1}^{D}\mu(x_i, 5, 100, 4),$$

$$x_i \in (-50, 50),$$

$$\mu(x_i, \alpha, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leqslant x_i \leqslant a \\ -k(-x_i - a)^m, & x_i < a \end{cases}$$

$$y_i = 1 + \frac{1}{4}(x_i + 1).$$

Penalized2 函数的类型为多维多峰分段函数,理想的收敛情况是在点 $\boldsymbol{x} = (1,1,\cdots,1)$ 处取得极小值 0.

7) Shifted Rastrigin 函数:

$$f_7 = \sum_{i=1}^{D}(y_i^2 - 10\cos(2\pi y_i) + 10),\ \boldsymbol{y} = \boldsymbol{M}\boldsymbol{x},$$

$$x_i \in (-5.12, 5.12).$$

Shifted Rastrigin 函数作为多维多峰的分段函数,理想的收敛情况是在点 $\boldsymbol{x} = (0,0,\cdots,0)$ 处取得极小值 0.

8) Shifted Griewank 函数:

$$f_2 = \sum_{i=1}^{D} \frac{y_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{y_i}{\sqrt{i}}\right) + 1,\ \boldsymbol{y} = \boldsymbol{M}\boldsymbol{x},$$

$$x_i \in (-600, 600).$$

Shifted Griewank 函数的类型为多维多峰分段

函数,理想的收敛情况是在点 $\boldsymbol{x}=(0,0,\cdots,0)$ 处取得极小值0.

3.2 参数设置和算法对比

实验环境: Inter Core(TM) i5 - 2450MCPU@3.2 GHz,**内存为** 4 GB,Window7 **操作系统**,Matlab R2015a **版本**.

参数设置:种群规模 $S=300$,$K=3$,**维度** $D=100$,**最大函数评价次数取为** $Max\ FEs=5D$,**最大迭代计算次数** $t_{\max}=1\ 000$,$t=100$,**过渡因子** $\delta=0.9$.

对比算法如下:全局版本的 PSO(A Global Version of PSO,GPSO),**局部版本的** PSO(A Local-Version of PSO,LPSO),**冯诺依曼拓扑结构的** PSO(Von Neumann Topological Structure of PSO,VPSO),**完全性** PSO(Fully Informed PSO,FIPS),**线性时变加速系数的** PSO(PSO with Linearly Time-Varying Acceleration Coefficients,HPSO-TVAC),**动态多群体的** PSO(Dynamic Multi-swarm PSO,DMS-PSO),**综合学习的** PSO(Comprehensive-Learning PSO,CLPSO),**自适应** PSO(Adaptive PSO,APSO).

各算法参数设置如表1所示.

表1 算法参数设置

Table 1 Algorithm parameter setting

算法	参数设置	文献
GPSO	$\omega:0.9\rightarrow0.4$,$c_1=c_2=2.0$	[2]
LPSO	$\omega:0.9\rightarrow0.4$,$c_1=c_2=2.0$	[24]
VPSO	$\omega:0.9\rightarrow0.4$,$c_1=c_2=2.0$	[24]
FIPS	$\chi=0.729$,$\sum_i c_i=4.1$	[25]
HPSO-TVAC	$\omega:0.9\rightarrow0.4$,$c_1:2.5\rightarrow0.5$,$c_2:0.5\rightarrow2.5$	[26]
DMS-PSO	$\omega:0.9\rightarrow0.2$,$c_1=c_2=2.0$,$m=3$,$R=5$	[27]
CLPSO	$\omega:0.9\rightarrow0.4$,$c=1.494$,$m=7$	[28]
APSO	$\omega:0.9\rightarrow0.4$,$c_1=c_2=2.0$	[29]
SOIPPSO	$\omega:0.9\rightarrow0.4$,$c_1=c_2=2.0$,$R=0.9$,$p_r=0.2$	–

3.3 参数控制策略

根据式(5)和式(6)中的参数及粒子群算法的基本参数可知,SOIPPSO **性能的优劣性主要受惯性权重** ω、**学习因子** c_1 **和** c_2、**地图罗盘因子** R **及变异概率** p_r **这4个主要参数的影响.算法处于不同状态下的参数组合选取策略如表2所示.**

策略1 在探索状态下,增加 c_1、p_r,减少 c_2、R、$t_{\max}$,可使算法处于最佳的探索状态,并有助于个体粒子达到历史最优位置,群体粒子也不会向当前局部最优粒子靠拢.另外,罗盘因子 R 的减少会使收敛速度下降,但可通过减少迭代次数以弥补下降的速度.

策略2 在开发状态下,明显增加 c_1,明显减少 c_2,减少 R、p_r,增加 $t_{\max}$,算法充分利用局部最优信息向历史最优位置靠拢,避免算法陷入局部最优状态.

策略3 在收敛状态下,明显增加 c_1、c_2,减少 p_r、R、$t_{\max}$,算法搜索到全局最优区域,另外,减少 c_1,加快算法的收敛速度.

策略4 在逃逸状态下,减少 c_1、R,增加 c_2、p_r、$t_{\max}$,算法跳出局部最优,一旦某个粒子发现新的搜索方向后,其它粒子以尽可能快的速度跟随.

表2 算法参数选取策略

Table 2 Selection strategy of algorithm parameter

状态	策略	c_1	c_2	R	$t_{\max}$	p_r
探索	1	增加	减少	减少	减少	增加
开发	2	明显增加	明显减少	减少	增加	减少
收敛	3	明显增加	明显增加	减少	减少	减少
逃逸	4	减少	增加	减少	增加	增加

3.4 时间复杂度分析

下面将本文提出的改进粒子群算法(SOIPPSO**)与传统的** PSO **的时间复杂度进行理论分析和对比.因为** PSO **在迭代计算过程中的粒子数量不变,**PSO **优化所需的计算时间为粒子数量、最大迭代次数和迭代一次的运行时间三者的乘积.根据本文算法参数的设置可知,**PSO **优化计算的时间为** $St_{\max}t$. **而对于** SOIPPSO,**粒子数量会受改进鸽群算法中地标算子优化的影响,逐步减少后期搜索能力弱的粒子,假设** SOIPPSO **迭代一次的计算时间为** T_D,SOIPPSO **优化所需的总计算时间为** $\frac{T_D}{2}\sum_{i=1}^{D}S_i$. **由上述分析可知,**PSO **与** SOIPPSO **的复杂度的差异主要取决于每次迭代中粒子数量和每次迭代时间.**

为了验证改进粒子群算法的性能,以文献[3]中8个较典型的测试函数为例,通过使用 PSO、PIO **与** SOIPPSO-1、SOIPPSO-2 **分别对8个测试函数进行仿真测试,并选取各自算法独立运行1 000次后的最优值、中间值、最差值、平均值、标准差和收敛计算时间.4种算法的计算结果如表3所示.**

从表3结果可知,在函数的理想收敛情况下,4种算法分别在100维下,经过1 000次的仿真测试,发现除 Rosebrock、Penalized1 **和** Penalized2 **函数外,其余5种函数经过100次的实验测试后几乎均可收**

敛到理想值.

针对 Sphere、Griewank、Ackley和 Shifted Griewank 函数,PSO、SOIPPSO-1 和 SOIPPSO-2 均可收敛到理想最优值,而 PIO 的寻优能力较弱. 这主要是因为基本鸽群算法在后期搜索过程中减少一半鸽子数目,从而降低粒子群算法的搜索能力. 研究 Shifted Rastrigin 函数的计算结果发现, 虽然采用 SOIPPSO-1 和 SOIPPSO-2 收敛计算的结果几乎接近函数的理想值,但其计算结果还不如 PSO,这主要是因为该函数是分段函数,算法在收敛计算时参数设置的差异造成. 另外,这些多维单峰和多峰函数经过 PSO 的收敛计算耗时较长,而采用 PIO 计算后耗时明显减少,但所有函数均未能收敛到全局理想最优值,因此,本文利用 PIO 后期运行过程中的鸽群数量减半特征,解决 PSO 后期收敛速度较慢、容易出现早熟的问题.

表 3　4 种算法的计算结果

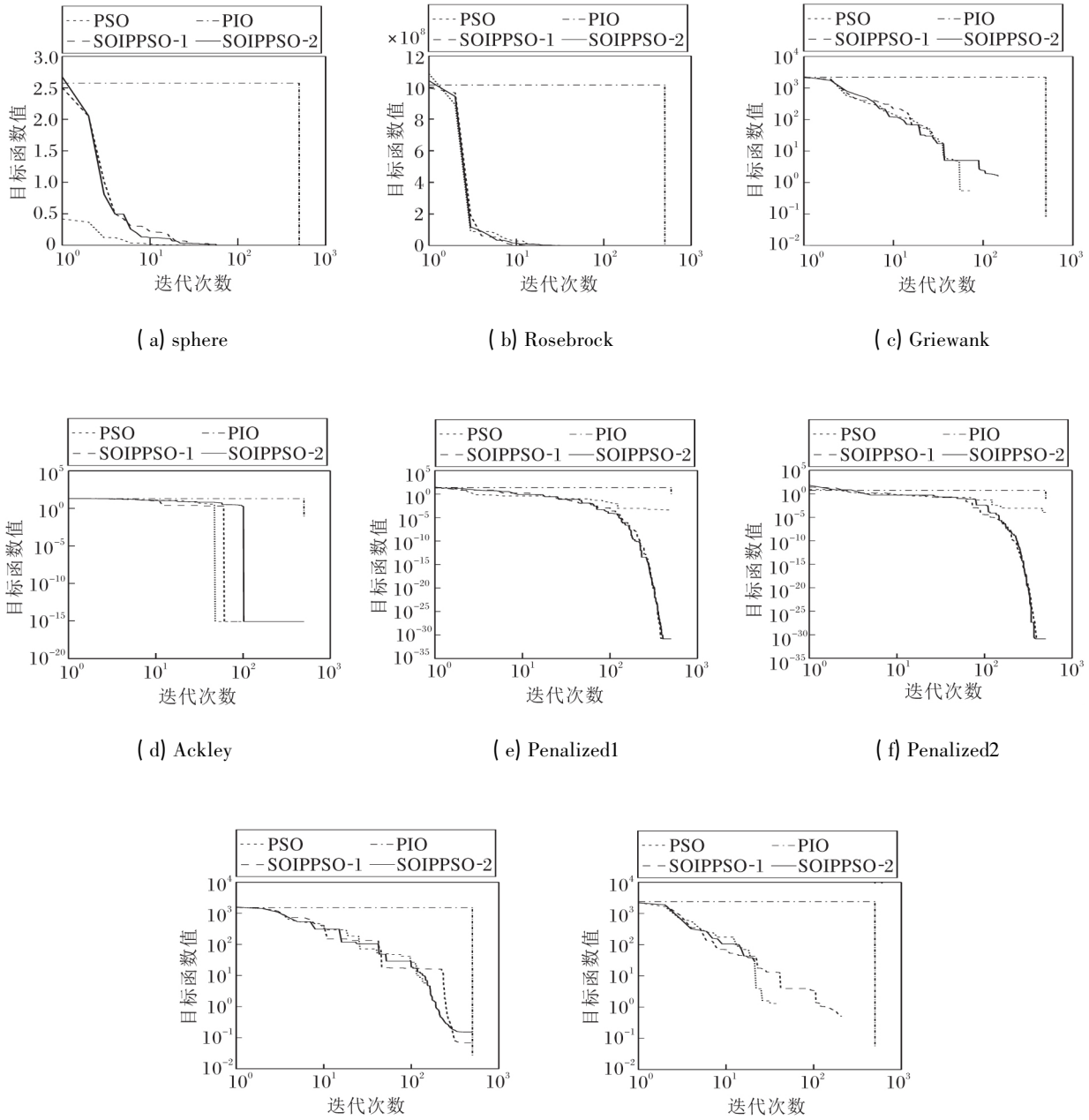
Table 3　Computational results of 4 algorithms

函数	算法	最优值	中间值	最差值	平均值	标准差	计算收敛时间 /s
Sphere	PSO	0	0	0	0	0	30.533440
	PIO	0.3999	0.00115	-0.0089	0.002085526	0.063235	2.355140
	SOIPPSO-1	0	0	0	0	0	2.339770
	SOIPPSO-2	0	0	0	0	0	2.356503
Rosebrock	PSO	98.9492	0.0006	0.0068	0.00072	0.0030308	34.226726
	PIO	1.0649e+03	-0.2824	-0.0027	-0.011547	0.2939231	3.276093
	SOIPPSO-1	9.0319e+04	0.05585	-29.4585	-0.250688	2.9946087	3.458104
	SOIPPSO-2	3.5656e+03	0.08135	24.5681	0.327201	3.044318	3.438308
Griewank	PSO	0	0	0	0	0	35.329833
	PIO	0.0827	0.008	-0.0133	-0.011388	0.166331	4.625459
	SOIPPSO-1	0	0	0	0	0	3.917831
	SOIPPSO-2	0	0	0	0	0	4.004048
Ackley	PSO	8.8818e-16	0	0	0	0	32.251217
	PIO	0.0868	0.0102	0.0106	0.004045	0.0171355	3.825825
	SOIPPSO-1	8.8818e-16	0	0	0	0	3.621461
	SOIPPSO-2	8.8818e-16	0	0	0	0	3.659961
Penalized1	PSO	1.0749e-04	12.1342	0	7.353278	36.775331	52.116227
	PIO	0.6561	-0.89745	-0.0003	-0.056101	0.9214338	22.366660
	SOIPPSO-1	1.5705e-31	-3.6307	0	-5.205078	33.61834	23.943064
	SOIPPSO-2	1.5705e-31	2.7278	0	4.710413	27.13389	23.570529
Penalized2	PSO	1.1292e-04	-0.7714	0	-0.107915	39.907506	71.569027
	PIO	0.0949	-0.67225	0.0073	-0.027318	0.691442	42.476757
	SOIPPSO-1	1.3498e-31	-1	0	-1.39692	33.59139	43.238373
	SOIPPSO-2	1.3498e-31	0.0699	0	-0.621116	36.18578	43.663245
Shifted Rastrigin	PSO	0	0	0	0	0	33.606242
	PIO	0.0270	-0.0001	0.0003	0.000027	0.0011706	3.404639
	SOIPPSO-1	0.0678	0.0003	-0.0004	0.000354	0.001808	3.273656
	SOIPPSO-2	0.1513	0.00025	-0.0019	0.000281	0.002742	3.271231
Shifted Griewank	PSO	0	0	0	0	0	30.958354
	PIO	0.0510	0.0057	0.0666	-0.007049	0.145669	4.606753
	SOIPPSO-1	0	0	0	0	0	4.080107
	SOIPPSO-2	0	0	0	0	0	3.938443

图 1 为 4 种算法的收敛曲线. 可清晰看出,提出的改进粒子群算法由于引入改进鸽群组合搜索算子、Beta 反向学习初始化策略,因此在计算复杂多峰函数时可快速跳出局部最优解范围,并在最短时间内向全局最优的方向收敛.

算法在处理单峰函数时收敛计算时间较少. 然而,虽然 PSO 和 PIO 在计算多峰函数时可以逃出局部最优,但是收敛计算速度明显慢于SOIPPSO-1、SOIPPSO-2.

从图 1、表 3 中的计算结果可知,SOIPPSO-1 与 SOIPPSO-2 在寻优计算能力上优于 PSO 和 PIO,但从部分函数的测试结果也可看出,改进后的算法在某些函数测试上也不能完全表现出最佳性能,这主要是因为算法初始参数选取、各自算法本身固有的弊端总会导致部分寻优效果不佳,而 SOIPPSO-1、SOIPPSO-2 从总体性能上优于 PSO 和 PIO.



(a) sphere (b) Rosebrock (c) Griewank

(d) Ackley (e) Penalized1 (f) Penalized2

(g) Shifted Rastrigin (h) Shifted Griewank

图1 4种算法在8个函数上的收敛曲线对比

Fig. 1 Comparison of convergence curves of 4 algorithms on 8 functions

为了进一步测试SOIPPSO的优越性,本文将SOIPPSO-1、SOIPPSO-2与GPSO[2]、APSO[3]、LPSO[23]、VPSO[23]、FIPS[25]、HPSO-TVAC[26]、DMS-PSO[27]、CLPSO[27]的优化结果进行对比分析,具体均值和方差如表4所示. 其它算法的计算结果均取自文献[5].

从表4结果可看出,除Rosebrock函数外,SOIPPSO-1和SOIPPSO-2的精度均有明显改进,个别函数的优化计算结果与理想值之间的误差很小,而其它函数均与理想值一致.

表 4　9 种算法在 8 个函数上的计算结果

Table 4　Computational results of 9 algorithms on 8 functions

函数	指标	GPSO	LPSO	VPSO	FIPS	HPSO-TVAC	DMS-PSO	CLPSO	APSO	SOIPPSO-1	SOIPPSO-2
f_1	均值	5.11e-038	4.77e-29	1.98e-53	3.21e-030	3.38e-041	3.85e-054	1.89e-019	1.45e-150	0	0
	方差	1.91e-037	1.13e-28	7.08e-53	3.60e-030	8.50e-041	1.75e-053	1.49e-019	5.73e-150	0	0
f_2	均值	28.1	21.8627	37.6469	22.5387	13	32.3	11	2.84	9.0319e+04	3.5656e+03
	方差	24.6	11.1593	24.9378	0.3102	16.5	24.1	14.5	3.27	2.9946087	3.04432
f_3	均值	2.37e-2	1.10e-2	1.31e-002	9.04e-004	1.07e-002	1.13e-002	6.45e-013	1.67e-002	0	0
	方差	2.57e-2	1.60e-2	1.35e-002	2.78e-003	1.41e-002	1.73e-002	2.07e-012	2.41e-002	0	0
f_4	均值	1.15e-14	1.85e-14	1.4e-14	7.69e-15	2.06e-10	8.52e-15	2.01e-12	1.11e-14	8.8818e-16	8.8818e-16
	方差	2.27e-15	4.80e-15	3.48e-15	9.33e-16	9.45e-10	1.79e-15	9.22e-13	3.55e-15	0	0
f_5	均值	1.04e-2	2.18e-30	3.46e-003	1.22e-031	7.07e-030	2.05e-032	1.59e-021	3.76e-031	1.5705e-31	1.5705e-31
	方差	3.16e-2	5.14e-30	1.89e-002	4.85e-032	4.05e-030	8.12e-033	1.93e-021	1.2e-030	0	0
f_6	均值	2.51e-2	2.03e-30	6.29e-002	1.32e-30	6.9e-023	2.61e-029	1.01e-021	5.15e-031	1.3498e-31	1.3498e-31
	方差	5.84e-2	2.89e-30	8.68e-002	7.86e-30	6.89e-023	6.60e-029	2.07e-021	1.43e-031	0	0
f_7	均值	30.7	34.9	34.09	29.98	2.39	28.1	2.57e-011	5.8e-015	0	0
	方差	8.68	7.25	8.07	10.92	3.71	6.42	6.64e-011	1.01e-014	0	0
f_8	均值	2.39e-2	1.13e-031	1.38e-003	9.14e-031	1.18e-030	1.13e-032	6.45e-031	1.67e-021	0	0
	方差	2.58e-2	1.62e-031	1.39e-002	2.81e-031	1.43e-030	1.73e-033	2.07e-031	2.41e-021	0	0

3.5　稳定性分析

为了验证 SOIPPSO 在不同参数组合选取策略下的稳定性,本文在 8 种不同算法的参数组合选取策略下,对 GPSO、VPSO、LPSO、FIPS、HPSO-TVAC、DMS-PSO、CLPSO、APSO、SOIPPSO 进行实验测试,获得 9 种算法的平均评价次数、计算时间和成功率,如表 5 所示. 由表 5 可知,除 Griewank 函数外,SOIPPSO 在其它函数上的成功率均能达到 100%. 从平均可靠性上看,SOIPPSO 的可靠性最高,为 97.58%,GPSO 性能优于 LPSO,FIPS 性能优于 GPSO、LPSO、VPSO、HPSO-TVAC、DMS-PSO. 对于多峰函数 Ackley,FIPS 可获得更高的精度,而 CLPSO 和 DMS-PSO 对于多峰函数 Griewank、Penalized1、Penalized2、Shifted Griewank 性能最好. 但是,SOIPPSO 对单峰和多峰函数的优化性能都表现较好,这是因为采用改进的鸽群组合搜索算子后,改进粒子群算法提高收敛速度,避免陷入局部最优,计算线性函数和非线性函数的性能更好.

表 5　9 种算法在 8 个函数上的收敛速度与可靠性对比

Table 5　Comparison of convergence speed and reliability of 9 algorithms on 8 functions

函数	指标	GPSO	LPSO	VPSO	FIPS	HPSO-TVAC	DMS-PSO	CLPSO	APSO	SOIPPSO
f_1	平均评价次数	105695	118197	112408	32561	30011	91496	72081	7074	7031
	计算时间 /s	0.96	1.12	1.04	0.36	0.29	0.85	0.48	0.11	0.09
	成功率 /%	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
f_2	平均评价次数	101579	102259	103643	13301	33689	87518	74815	5334	5309
	计算时间 /s	0.99	1.05	1.05	0.16	0.35	0.86	0.55	0.09	0.05
	成功率 /%	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
f_3	平均评价次数	111733	125777	117946	42604	34154	97213	81422	7568	7392
	计算时间 /s	1.46	1.86	1.68	0.72	0.48	1.29	0.95	0.16	0.11
	成功率 /%	40.0	60.0	46.7	100.00	56.7	56.7	100.00	66.7	75.3
f_4	平均评价次数	110844	125543	118926	38356	52516	100000	76646	40736	38527
	计算时间 /s	1.40	1.81	1.68	0.62	0.70	1.27	0.79	0.93	0.85
	成功率 /%	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
f_5	平均评价次数	99541	107452	102779	19404	44491	95830	59160	21538	19461
	计算时间 /s	2.17	2.57	2.38	0.50	0.98	2.10	1.38	0.68	0.41
	成功率 /%	90.00	100.00	96.70	100.00	100.00	100.00	100.00	100.00	100.00
f_6	平均评价次数	97658	113912	106841	19308	45376	96117	58263	20134	18268
	计算时间 /s	2.09	2.40	2.57	0.56	0.86	2.03	1.27	0.51	0.39
	成功率 /%	85.00	100.00	95.8	100.00	100.00	100.00	100.00	100.00	100.00
f_7	平均评价次数	94379	99074	98742	87760	7829	127423	53416	3531	3328
	计算时间 /s	1.24	1.38	1.31	1.34	0.10	1.67	0.95	0.08	0.05
	成功率 /%	96.70	96.70	100.00	93.3	100.00	100.00	100.00	100.00	100.00
f_8	平均评价次数	101753	135867	114996	43194	36184	98243	81516	7571	7462
	计算时间 /s	1.52	1.46	1.65	0.71	0.41	1.39	1.13	0.18	0.14
	成功率 /%	60.0	65.0	47.5	100.00	52.7	57.7	100.00	67.0	100.00
平均可靠性 /%		85.34	81.68	83.71	91.75	86.01	84.92	90.58	96.45	97.58

为了进一步验证本文算法的可信度,采用文献[30]和文献[31]中自由度为29,置信度$\alpha=0.05$的双侧t检验,如表6所示.

表6中的最底部给出SOIPPSO相对其它算法的性能,表现最差的是值为0的函数,总体竞争优势上SOIPPSO比GPSO总体表现更优.另外,通过对比各算法的t_value和p_value的值发现,SOIPPSO更佳.虽然性能在部分函数上表现较弱,但总体上SOIPPSO比其它算法性能更优.

为了验证本文提出的初始化方法的效果,对SOIPPSO-1、SOIPPSO-2在不同初始化策略下进行实验仿真,结果如表7所示.本次实验取20次独立计算后的误差限[3]所需的评价次数.

从表7的初始化效果可看出,使用本文提出的Beta反向学习初始化种群策略对7种函数进行初始化,使用提出的改进算法对函数优化求解达到理想效果,在相同的误差限条件下,采用本文的初始化方法对函数的评价次数少于采用随机初始化方法.另外,从表7的初始化效果可进一步说明SOIPPSO-2的性能明显优于SOIPPSO-1,这是因为采用非线性变异策略后,算法在求解计算复杂高维多峰函数时的效率更高,也能体现出算法的综合优越性.

表6 SOIPPSO与其它算法的t检验($\alpha=0.05$)

Table 6 Comparison of SOIPPSO and other algorithms on t-tests($\alpha=0.05$)

函数	秩和检验	GPSO	LPSO	VPSO	FIPS	HPSO-TVAC	DMS-PSO	CLPSO	APSO
f_1	t-value	1.52851	2.31098*	1.4671	4.88501*	2.17917*	1.20579	6.93676*	2.13071*
	p-value	0.13182	0.02441	0.14775	0.00001	0.03339	0.23279	0.0000	0.01362
f_2	t-value	5.57538*	8.96094*	7.58036*	32.85535*	3.29589*	6.62263*	3.00317*	3.68591
	p-value	0.000000	0.00000	0.000000	0.00000	0.00168	0.00000	0.00394	0.00000
f_3	t-value	1.08486	−1.07192	−0.70658	−3.56237*	−1.23569	−0.6606	−3.79146*	−0.06233
	p-value	0.28247	0.28820	0.48265	0.00074	0.22155	0.51148	0.00036	0.23358
f_4	t-value	0.46159	6.73442*	3.78212*	−5.12379*	1.19682	−3.58847*	11.89982*	0.298165
	p-value	0.64610	0.000000	0.00037	0.00000	0.23624	0.00068	0.00000	0.000000
f_5	t-value	1.79505	1.87465	1.00000	−1.15597	8.68004*	−1.61897	4.51261*	1.65229
	p-value	0.07786	0.06588	0.32146	0.25243	0.00000	0.11088	0.00003	0.059774
f_6	t-value	1.89565	1.82425	1.43688	−1.16562	8.57794*	−1.68334	4.53882*	1.80267
	p-value	0.08085	0.05766	0.33546	0.24556	0.00000	0.13422	0.00006	0.025613
f_7	t-value	27.42668*	35.28576*	25.33892*	15.03442*	3.52542*	24.02031*	2.12311	26.33876*
	p-value	0.000000	0.000000	0.000000	0.000000	0.000083	0.000000	0.03802	0.000000
f_8	t-value	1.08976	−1.05389	−0.80563	−3.56554*	−1.25946	−0.6803	−3.73086*	−1.03891
	p-value	0.23472	0.24357	0.48565	0.00024	0.23264	0.54335	0.00029	0.23659
1(好)		7	9	8	7	8	8	8	9
0(相同)		5	3	3	1	3	2	3	3
−1(差)		0	0	0	3	0	2	1	2
总体竞争优势		7	9	8	4	8	6	7	7

表7 初始化方法的影响效果

Table 7 Effect of initial method

函数	误差限	初始化策略	SOIPPSO-1	SOIPPSO-2
f_1	0.01	随机方法	85	83
		本文方法	80	79
f_2	100	随机方法	103	99
		本文方法	98	97
f_3	50	随机方法	86	82
		本文方法	69	62
f_4	50	随机方法	84	77
		本文方法	52	49
f_7	0.01	随机方法	110	98
		本文方法	38	31
f_8	0.01	随机方法	69	63
		本文方法	48	42
f_{10}	100	随机方法	98	92
		本文方法	92	91

由上述实验结果可看出,SOIPPSO使用Beta反向学习初始化方法和改进鸽群组合搜索算子的方式提高算法的搜索计算效率,有效克服PSO的后期收敛计算速度较慢、求解结果精度较低、易于陷入局部最优等问题.另外,根据文献[32]的定理,任何算法不是万能的,不可能适用所有的问题.对于SOIPPSO表现出的优良特性,给出如下理论分析.

1)从种群初始化的方法上看,本文采用Beta反向学习策略,随机初始化粒子种群位置.这种方法的优点是采用Beta分布函数和反向学习策略达到初始种群优化的目的,确保初始种群的粒子分布在最优解周围而逼近最优解.而采用随机均匀分布的其它算法,对于最优解无法形成包围态势,这对于逼近最优解的优势差于SOIPPSO.

2) 在算法搜索过程中,算法的开发和探索能力、计算精度和速度的平衡发展是制约其性能的瓶颈. 从 SOIPPSO 的种群位置更新上看,本文采用改进的鸽群组合搜索算子,该算子中的地标算子迭代一次后,鸽群数量减半,较快地提高算法的搜索速度和目标位置的精度,进一步提高优化性能.

3) 从粒子位置和速度更新方式上看,在标准 PSO 的基础上,受鸽群进化算法的启示,本文增加基于线性变异策略和非线性变异策略,修正地图罗盘因子,从而对鸽群算法中两个独立运行的算子进行有效组合,用于更新所有粒子位置和速度的算子. 在算法全局搜索初期,有利于减少搜索能力最弱的粒子数量. 随着迭代次数的逐渐减小,算子作用逐渐减弱,有利于粒子向最优解方向聚集,进一步保障算法的开发和探索能力.

4　结 束 语

本文结合粒子群算法和鸽群算法的原理,将鸽群算法思想引入粒子数算法,提出引入改进鸽群搜索算子的粒子群优化算法. 采用 Beta 反向学习策略,初始化粒子种群的初始分布状态,较好地保持种群的多样性,并利用线性变异策略与非线性变异策略对鸽群算法中的地图罗盘算子进行修正,提高标准粒子群算法求解复杂多峰函数的能力. 另外,将鸽群算法中两个独立运行的算子进行优化组合,进而对粒子群算法中位置和速度进行更新,有效保障粒子群算法的开发和探索能力. 实验结果表明,SOIPPSO 在全局收敛率、收敛速度和精度方面都有明显提高,较好地平衡全局探索能力和局部开发能力,稳定性较好. 通过参数选取策略和种群初始化策略的实验可以发现,算法进化的优劣状态受种群初始化、算法参数的选取策略的影响较大,依然存在很大的提高空间,今后将深入研究复杂高维问题应用改进的粒子群优化及参数控制.

参 考 文 献

[1] KENNEDY J, EBERHART R C. Particle Swarms Optimization // Proc of the IEEE International Conference on Neural Networks. Washington, USA: IEEE, 1995: 1942-1948.

[2] SHI Y H, EBERHART R C. A Modified Particle Swarm Optimizer // Proc of the IEEE International Conference on Evolutionary Computation Proceedings. Washington, USA, IEEE, 1998: 69-73.

[3] NICKABADI A, EBADZADEH M M, SAFABAKHSH R. A Novel Particle Swarm Optimization Algorithm with Adaptive Inertia Weight. Applied Soft Computing, 2011, 11(4): 3658-3670.

[4] MARINAKIS Y, MIGDALAS A, SIFALERAS A. A Hybrid Particle Swarm Optimization-Variable Neighborhood Search Algorithm for Constrained Shortest Path Problems. European Journal of Operational Research, 2017, 261(3): 819-834.

[5] 于 颖,李永生,於孝春. 粒子群算法在工程优化设计中的应用. 机械工程学报, 2008, 44(12): 226-231.
(YU Y, LI Y S, YU X C. Application of Particle Swarm Optimization in the Engineering Optimization Design. Chinese Journal of Mechanical Engineering, 2008, 44(12): 226-231.)

[6] 万春秋,王 峻,杨 耕,等. 动态评价粒子群优化及风电场微观选址. 控制理论与应用, 2011, 28(4): 449-456.
(WAN C Q, WANG J, YANG G, *et al*. Dynamic Evaluation Based Particle Swarm Optimization and Wind Farm Micrositing. Control Theory and Applications, 2011, 28(4): 449-456.)

[7] 张国富,蒋建国,齐美彬,等. 基于粒子群算法求解多层非线性规划问题. 模式识别与人工智能, 2007, 20(6): 745-750.
(ZHANG G F, JIANG J G, QI M B, *et al*. Solutions of Nonlinear Multilevel Programming Based on Particle Swarm Optimization. Pattern Recognition and Artificial Intelligence, 2007, 20(6): 745-750.)

[8] DU W B, YING W, YAN G, *et al*. Heterogeneous Strategy Particle Swarm Optimization. IEEE Transactions on Circuits and Systems II (Express Briefs), 2017, 64(4): 467-471.

[9] SAKTHIVEL V P, BHUVANESWARI R, SUBRAMANIAN S. An Improved Particle Swarm Optimization for Induction Motor Parameter Determination. International Journal of Computer Applications, 2011, 1(2): 62-67.

[10] CHEN S W, XU Z M, TANG Y, *et al*. An Improved Particle Swarm Optimization Algorithm Based on Centroid and Exponential Inertia Weight. Mathematical Problems in Engineering, 2014. DOI: 10.1155/2014/976486.

[11] ZAVALA A E M, AGUIRRE A H, DIHARCE E R V, *et al*. Constrained Optimization with an Improved Particle Swarm Optimization Algorithm. International Journal of Intelligent Computing and Cybernetics, 2008, 1(3): 425-453.

[12] DUAN H B, QIAO P X. Pigeon-Inspired Optimization: A New Swarm Intelligence Optimizer for Air Robot Path Planning. International Journal of Intelligent Computing and Cybernetics, 2014, 7(1): 24-37.

[13] BOLAJI A L, BABATUNDE B S, SHOLA P B. Adaptation of Binary Pigeon-Inspired Algorithm for Solving Multidimensional Knapsack Problem // PANT M, RAY K, SHARMA T, *et al*., eds. Soft Computing: Theories and Applications. Berlin, Germany: Springer, 2017, 538: 743-751.

[14] LI H H, DUAN H B. Bloch Quantum-Behaved Pigeon-Inspired Optimization for Continuous Optimization Problems // Proc of the IEEE Chinese Guidance, Navigation and Control Conference. Washington, USA: IEEE, 2014: 2634-2638.

[15] ZHANG B, DUAN H B. Predator-Prey Pigeon-Inspired Optimization for UAV Three-Dimensional Path Planning // Proc of the 5th International Conference on Swarm Intelligence. Berlin, Germany: Springer, 2014: 96-105.

[16] ZHANG T J, DUAN H B. A Modified Consensus Algorithm for Multi-UAV Formations Based on Pigeon-Inspired Optimization with a Slow Diving Strategy. CAAI Transactions on Intelligent System, 2017, 12(4): 570-581.

[17] 李建平,宫耀华,卢爱平,等. 改进的粒子群算法及在数值函数优化中应用. 重庆大学学报, 2017, 40(5): 95-103.
(LI J P, GONG Y H, LU A P, *et al*. Application of Improved Particle Swarm Optimization to Numerical Function Optimization. Journal of Chongqing University, 2017, 40(5): 95-103.)

[18] 肖文显,刘 震. 一种融合反向学习和量子优化的粒子群算法. 微电子学与计算机, 2013, 30(6): 126-130.
(XIAO W X, LIU Z. Particle Swarm Optimization Based on Opposition-Based Learning and Quantum Optimization. Microelectronics and Computer, 2013, 30(6): 126-130.)

[19] 张聚伟,王 宇,杨 挺. 基于模糊粒子群算法的有向传感器网络路径覆盖策略. 模式识别与人工智能, 2017, 30(2): 183-192.
(ZHANG J W, WANG Y, YANG T. Path Coverage Scheme Based on Fuzzy Particle Swarm Optimization Algorithm for Directional Sensor Networks. Pattern Recognition and Artificial Intelligence, 2017, 30(2): 183-192.)

[20] ZHOU H G, ZHENG C W, HU X H, *et al*. Path Planner for Unmanned Aerial Vehicles Based on Modified PSO Algorithm // Proc of the IEEE International Conference on Information and Automation. Washington, USA: IEEE, 2008: 541-544.

[21] HAO R, LUO D L, DUAN H B. Multiple UAVs Mission Assignment Based on Modified Pigeon-Inspired Optimization Algorithm // Proc of the IEEE Chinese Guidance, Navigation and Control Conference. Washington, USA: IEEE, 2014: 2692-2697.

[22] 段海滨,邱华鑫,范彦铭. 基于捕食逃逸鸽群优化的无人机紧密编队协同控制. 中国科学(技术科学), 2015, 45(6): 559-572.
(DUAN H B, QIU H X, FAN Y M. Unmanned Aerial Vehicle Close Formation Cooperative Control Based on Predatory Escaping Pigeon-Inspired Optimization. Chinese Science(Technological Sciences), 2015, 45(6): 559-572.)

[23] QIU H X, DUAN H B. Multi-objective Pigeon-Inspired Optimization for Brushless Direct Current Motor Parameter Design. Science China(Technological Sciences), 2015, 58(1): 1915-1923.

[24] KENNEDY J, MENDES R. Population Structure and Particle Swarm Performance // Proc of the Congress on Evolutionary Computation. Washington, USA: IEEE, 2002: 1671-1676.

[25] MENDES R, KENNEDY J, NEVES J. The Fully Informed Particle Swarm: Simpler, Maybe Better. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 204-210.

[26] RATNAWEERA A, HALGAMUGE S K, WATSON H C. Self-organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 240-255.

[27] ZHAO S Z, SUGANTHAN P N, PAN Q K, *et al*. Dynamic Multi-swarm Particle Swarm Optimizer with Harmony Search. Expert Systems with Applications, 2011, 38(4): 3735-3742.

[28] LIANG J J, QIN A K, SUGANTHAN P N, *et al*. Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281-295.

[29] CIUPRINA G, IOAN D, MUNTEANU I. Use of Intelligent-Particle Swarm Optimization in Electromagnetics. IEEE Transactions on Magnetics, 2002, 38(2): 1037-1040.

[30] YAO X, LIU Y, LIN G M. Evolutionary Programming Made Faster. IEEE Transactions on Evolutionary Computation, 1999, 3(2): 82-102.

[31] BOX G E P, HUNTER J S, HUNTER W G. Statistics for Experiment: Design, Innovation, and Discovery. Technometrics. 2nd Edition. New York, USA: Wiley, 2005.

[32] WOLPERT D H, MACREADY W G. No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 67-82.

作者简介

马 龙,博士研究生,讲师,主要研究方向为智能计算、系统建模与优化. E-mail: malong1982@126.com.
(**MA Long**, Ph. D. candidate, lecturer. His research interests include intelligent computing, system modeling and optimization.)

卢才武,博士,教授,主要研究方向为系统优化理论、信息技术、信息管理. E-mail: lucaiwu@126.com.
(**LU Caiwu**, Ph. D., professor. His research interests include system optimization theory, information technology and information management.)

顾清华(通讯作者),博士,副教授,主要研究方向为系统优化理论、智能计算. E-mail: qinghuagu@126.com.
(**GU Qinghua** (Corresponding author), Ph. D., associate professor. His research interests include system optimization theory and intelligent computing.)

阮顺领,博士,讲师,主要研究方向为智能计算、系统建模与优化. E-mail: ruanshunling@163.com.
(**RUAN Shunling**, Ph. D., lecturer. His research interests include intelligent computing, system modeling and optimization.)