计算机工程与应用
Computer Engineering and Applications
ISSN 1002-8331,CN 11-2127/TP

《计算机工程与应用》网络首发论文

题目：　优化复杂函数的粒子群-鸽群混合优化算法
作者：　顾清华，孟倩倩
网络首发日期：　2018-12-21
引用格式：　顾清华，孟倩倩．优化复杂函数的粒子群-鸽群混合优化算法[J/OL]．计算机工程与应用. http://kns.cnki.net/kcms/detail/11.2127.tp.20181218.1035.010.html

网络首发：在编辑部工作流程中，稿件从录用到出版要经历录用定稿、排版定稿、整期汇编定稿等阶段。录用定稿指内容已经确定，且通过同行评议、主编终审同意刊用的稿件。排版定稿指录用定稿按照期刊特定版式（包括网络呈现版式）排版后的稿件，可暂不确定出版年、卷、期和页码。整期汇编定稿指出版年、卷、期、页码均已确定的印刷或数字出版的整期汇编稿件。录用定稿网络首发稿件内容必须符合《出版管理条例》和《期刊出版管理规定》的有关规定；学术研究成果具有创新性、科学性和先进性，符合编辑部对刊文的录用要求，不存在学术不端行为及其他侵权行为；稿件内容应基本符合国家有关书刊编辑、出版的技术标准，正确使用和统一规范语言文字、符号、数字、外文字母、法定计量单位及地图标注等。为确保录用定稿网络首发的严肃性，录用定稿一经发布，不得修改论文题目、作者、机构名称和学术内容，只可基于编辑规范进行少量文字的修改。

出版确认：纸质期刊编辑部通过与《中国学术期刊（光盘版）》电子杂志社有限公司签约，在《中国学术期刊（网络版）》出版传播平台上创办与纸质期刊内容一致的网络版，以单篇或整期出版形式，在印刷出版之前刊发论文的录用定稿、排版定稿、整期汇编定稿。因为《中国学术期刊（网络版）》是国家新闻出版广电总局批准的网络连续型出版物（ISSN 2096-4188，CN 11-6037/Z），所以签约期刊的网络版上网络首发论文视为正式出版。

网络首发时间：2018-12-21 09:01:14
网络首发地址：http://kns.cnki.net/kcms/detail/11.2127.tp.20181218.1035.010.html

优化复杂函数的粒子群-鸽群混合优化算法

顾清华，孟倩倩

GU Qinghua, MENG Qianqian

西安建筑科技大学 管理学院，西安 710055

School of Management, Xi'an University of Architecture and Technology, Xi'an 710055, China

GU Qinghua, MENG Qianqian. The Hybrid Particle Swarm Optimization and Pigeon-Inspired Optimization Algorithm for Solving Complex Functions. Computer Engineering and Applications

Abstract: In order to solve the complicated function optimization problems, a two-stage hybrid optimization algorithm was proposed.The basic particle swarm optimization and pigeon-inspired optimization algorithm were improved.The inertia factor and jump operator were used to enhance the searching ability of particle swarm optimization, and the interference operator was used to increase the population diversity of the pigeon-inspired optimization.The improved algorithm was combined to form the two-stage hybrid optimization algorithm. Meanwhile, a diversity function was defined to detect the population diversity in real-time monitoring to ensure the diversity of the population.From the simulation results,it was shown that the algorithm is suitable for solving complex function optimization problems, and has good convergence speed and convergence accuracy.

Keywords: complex function optimization; particle swarm optimization; pigeon-inspired optimization; two-stage hybrid optimization algorithm

摘　要：针对复杂函数优化问题，提出一种两阶段混合优化算法。首先，对基本粒子群和鸽群算法进行改进，引入惯性因子和跳跃算子增强了粒子群算法的搜索能力，提出干扰算子增加了鸽群算法的种群多样性。然后，将改进后的两种算法相结合形成两阶段混合优化算法，同时定义了一种多样性函数对种群多样性进行实时监测以保证种群的多样性。最后，采用两组经典测试函数，对算法性能进行测试。结果表明，算法适用于求解复杂函数优化问题且具有较好的收敛速度和收敛精度。

关键词：复杂函数优化；粒子群算法；鸽群算法；两阶段混合算法

文献标志码：A　　中图分类号：TP301.6　　doi：10.3778/j.issn.1002-8331.1808-0151

基金项目：国家自然科学基金资助项目（No.51774228）；陕西省自然科学基金项目（No.2017JM5043）。

作者简介：顾清华(1981－),男,博士,副教授,主要研究方向为系统优化、系统建模及仿真研究等；孟倩倩（1991-），女，硕士研究生，主要研究方向：系统建模与仿真研究，Email：1056659206@qq.com。

1 引言

实际生产中，大量的生产过程优化问题，建立数学模型后，都可以看作是复杂函数的优化问题，这类问题一般存在非线性、不连续、多个极小值、不可微、平坦区等特点。采用传统的非线性优化方法，如梯度下降法等，求解此类问题存在一定的局限性，一方面，当问题的目标函数和约束条件是非线性不连续不可微，甚至不存在数学表达式，但又存在大量局部区域连续可微的凸区域时，此类方法无法解决，另一方面，此类方法对于初始值的选择依赖很大，如果初始值选择的不好，极易陷入局部极值，但是目前仍没有完善的理论方法指导其初始值的选择。近年来，研究者提出采用群智能优化算法求解优化问题，群智能算法具有原理简单、鲁棒性强且对于目标函数和约束条件没有特殊要求等特点，对于求解上述复杂函数优化问题具有重要意义。

粒子群算法（PSO）[1]是群智能优化算法的一种，因其具有原理简单、初始参数较少、容易实现、前期收敛速度快等优点[2]。但是 PSO 算法中粒子主要依靠个体经验和种群经验来调节其飞行方向，后期种群逐渐呈现同一性，在求解复杂函数优化问题时会出现后期收敛速度缓慢，且易陷入局部最优等问题[3]。大多学者对 PSO 算法进行改进研究，并将改进后的算法常被用来解决工程中的优化问题。例如，文献[4]改进粒子群算法中包含重组、子群规模调整和探测三个模块，采用重组、种群规模调整以及探测三个模块对算法进行改进，有效的提高了算法的探测和开采能力。文献[5]引入混沌搜索和自适应惯性权重策略提高算法的全局搜索能力，并采用 SQP 策略提高局部搜索能力，并将改进后的算法用于结构优化问题。文献[6]将改进粒子群算法与 NEH 算法将结合，用于解决车间调度问题；文献[7]针对粒子群算法易陷入局部最优的缺点对其进行改进研究，并将其用于函数优化问题；文献[8]对粒子群算法最优值的选取，惯性权值和学习因子的更新方式等进行了改进，并将其应用于 PID 参数优化。

鸽群算法（PIO）是 2014 年由 Duan 等人提出的，是一种基于地图、指南针算子和地标算子的新型仿生群智能算法[9]。该算法具有收敛速度快的优点，但是与大多数算法一样当问题变得复杂时极易陷入局部最优[10-11]。研究者针对该算法的缺点进行改进并将其成功应用于工程优化问题。例如，文献[12]针对轨道航天器位置重定向问题，提出了高斯分布鸽群算法，用高斯分布代替均匀分布，增加算法的全局搜索能力。文献[13]针对无人机的目标检测问题，提出采用概率来区分地图指南针算子和地标算子，同时采用高斯扰动和模拟退火策略，提高了算法的局部搜索和全局搜索能力。文献[14]通过引入导航工具过渡因子与捕食逃逸机制对鸽群算法进行改进，提高了算法局部搜索能力。文献[15]在标准鸽群算法的基础上引入罚函数的思想，改进后的算法对于求解复杂带约束优化的问题更加高效。文献[16]提出一种引入威胁启发机制的鸽群算法，利用已知的威胁信息作为启发信息指导鸽子的飞行。文献[17]引入多 Agent 系统机制和雁群机制对鸽群算法进行改进，提高算法的收敛速度和全局搜索能力。文献[18]采用鸽群算法优化控制参数，并引入概率因子对算法进行改进，使粒子以一定的概率选择不同的更新公式。文献[19]采用鸽群算法对包含等式和不等式约束的滑翔轨迹生成问题进行优化。

综合考虑，复杂函数优化问题的特点和 PSO 算法及 PIO 算法各自的优缺点，本文提出一种基于改进粒子群与鸽群的两阶段混合优化算法（PSO-PIO）。针对 PSO 算法易陷入局部最优的缺点，在基本粒子群算法中引入动态惯性因子，既保证了算法的全局搜索能力又满足算法局部搜索的需要，同时提出跳跃算子对 PSO 算法进行改进，使粒子以自适应的跳跃概率，跳出当前位置，增强了粒子跳出局部极值的能力。针对 PIO 算法易陷入局部最优的缺点引入干扰算子对种群中每个粒子增加干扰项，从而增加种群的多样性。在两种算法改进的基础上，按照利用优点规避缺点的原则将两种算法相结合，并采用多样性函数对算法进行实施监控，以防陷入局部最优，该混合算法能够快速有效的寻到复杂函数的最优解或较优解。

2 粒子群和鸽群算法

2.1 带跳跃算子的粒子群算法

基本 PSO 算法的设计如下：假设在 D 维搜索空间中，由 m 个粒子组成一个种群。种群在 D 维空间中搜索寻找最优解，其中每个粒子都有自己的位置和速度。在 t 时刻，第 i(0<i≤m)个粒子的位置和速度为：$X^t_i=(x^t_{1i},x^t_{2i},...,x^t_{Di})$，$V^t_i=(v^t_{1i},v^t_{2i},...,v^t_{Di})$，该粒子在 t+1 时刻，根据下列公式（1）和（2）更新自身的位置和速度，公式（1）和（2）为：

$$v_{id}^{t+1}=\omega\cdot v_{id}^{t}+c_1\cdot r_1\cdot(pbest_{id}^{t}-x_{id}^{t})+c_2\cdot r_2\cdot(gbest_{d}^{t}-x_{id}^{t}) \quad (1)$$

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

$$1\le i\le m,\ \ 1\le d\le D$$

其中：ω 为动态惯性因子[20]，ω 较大适用于较大范围的搜索，较小适用于小范围的搜索，迭代初期种群适合在较大范围内搜索，避免陷入局部最优，迭代后期种群整体趋向最优解的方向，此时适合在小范围内精确搜索，有利于提高收敛速度，所以本文 ω 随着迭代过程线性递减，变化公式为：$\omega=(\omega_{max}-\omega_{min})\times(T_{max}-T)/T_{max}+\omega_{min}$；$c_1$、$c_2$ 为学习因子；r_1、r_2 为[0,1]之间的随机数；pbest 为个体极值，gbest 为全局极值。

PSO 算法中种群的所有粒子都依靠个体极值（*pbest*）和全局极值(*gbest*)改变自己的飞行方向，然而 *pbest* 和 *gbest* 是相互联系的，这是典型的星型结构，中心节点为 *gbest*，所有的个体无条件的跟随 *gbest*。迭代初期，种群的搜索范围较广，随着迭代的进行，粒子不断向最优 *gbest* 飞行，种群多样性下降。迭代后期，种群将跟随 *gbest* 在较小区域内搜索，如果这个区域是局部极值所在区域时，粒子将很难跳出这个局部极值。因此本文提出一种自适应跳跃算子，通过比较 *pbest* 与 *gbest* 的相似程度，赋予粒子不同的跳跃概率。在 *t* 时刻，第 *i* 个粒子跳出当前位置的概率公式（3）和跳跃公式（4）为：

$$p_m=\exp(f(gbest^t)-f(pbest_i^t)) \quad (3)$$

$$x_i^t=x_i^t+rand\cdot(ub-lb) \quad (4)$$

其中：*D* 为搜索空间的维数；*rand* 为[0,1]之间的随机数；*ub* 和 *lb* 为问题的上下限。从公式中可以看出当粒子的 *pbest* 与种群 *gbest* 越接近时，粒子产生跳跃的概率较大，也就是说可以使停止更新的粒子，以较大的概率跳出极值点，对于距离种群 *gbest* 较远的粒子，跳跃算子对其的作用较小。

2.2 带干扰算子的鸽群算法

PIO 算法通过地图、指南针和地标两个不同的算子分两个阶段进行寻优。算法的设计如下：地图、指南针算子中，假设在 *D* 维搜索空间中，由 *m* 个粒子组成一个种群，每

个粒子都有自身的速度和位置，每次迭代，粒子都通过最优信息更新自身的速度和位置。在 t 时刻，第 i（$0<i\le m$）个粒子的位置和速度为：$X^t_i=(x^t_{1i},x^t_{2i},...,x^t_{Di})$，$V^t_i=(v^t_{1i},v^t_{2i},...,v^t_{Di})$，在 $t+1$ 时刻，该粒子根据公式（5）和（6）更新自己的速度和位置，更新公式为：

$$v_{id}^{t+1}=v_{id}^{t}\cdot e^{R\cdot(t+1)}+rand\cdot(gbest-x_{id}^{t}) \quad (5)$$

$$x_{id}^{t+1}=x_{id}^{t}+v_{id}^{t} \qquad 0<i\le m,\quad 0<d\le D \quad (6)$$

其中：R 为地图和指南针因数，在 0~1 之间取值；t 为当前迭代次数；$rand$ 为[0,1]之间的随机数；$gbest$ 为全局最优解。当算法达到一定的迭代次数，地图、指南针算子的运算结束，进入地图算子。

在地标算子中，每次迭代种群的数量都会减半，如公式（7）所示。那些不熟悉地标的鸽子，不具有识别路径的能力，将会被舍弃。每次迭代，粒子依靠种群的中心位置更新自身的位置，种群中心位置和各粒子位置根据公式（8）和（9）进行更新，公式为：

$$m^{t+1}=\frac{m^t}{2} \quad (7)$$

$$x_{center}^{t}=\frac{\sum_{i=1}^{m^t}x_i^t\cdot fitness(x_i^t)}{m^t\sum_{i=1}^{m^t}fitness(x_i^t)} \quad (8)$$

$$x_i^{t+1}=x_i^t+rand\cdot(x_{center}^t-x_i^t) \quad (9)$$

其中：x^t_{center} 为种群的中心位置，作为地标指导种群的飞行方向；m 为种群大小；$fitness(x)$为适应度函数，在函数优化中，由目标函数构成。

PIO 算法应用到实际问题中，在有限的迭代次数下，易陷入局部最优，尤其在求解复杂函数优化中。本文为了提高算法的全局搜索能力，在 PIO 算法的地标算子中增加一种干扰算子。干扰公式为：

$$pert(t)=0.1\cdot rand\cdot(1-\frac{t}{t\max}) \quad (10)$$

$$x_i^t=x_i^t+pert(t)\cdot(ub-lb)\cdot(r_1-r_2) \quad (11)$$

其中：$rand$、r_1 和 r_2 为[0,1]上的随机数。$pert(t)$ 为扰动系数，随着迭代次数的增加，干扰系数逐渐递减，干扰算子对粒子的干扰程度也逐渐减弱。如图 1 所示，干扰系数随着迭代次数的变化过程。该干扰算子前期对粒子的位置更新产生较大的扰动，增加种群的多样性，从而有效的增加种群的全局搜索能力。后期对粒子的干扰逐渐减小最终为零，避免影响算法后期的局部搜索以及收敛性。

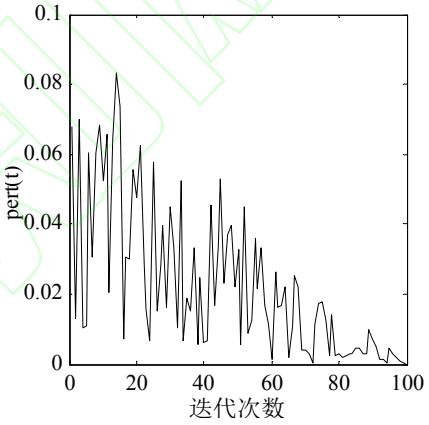


图 1　干扰系数随着迭代次数的变化过程

3　粒子群-鸽群混合优化算法设计

在算法设计之前，定义了一种多样性函数。一种衡量种群多样性的方法，对种群的多样性进行实时检测。

定义 1：种群粒子与最优粒子的欧氏距离：

$$l_i^t=\sqrt{\sum_{d=1}^{D}\left(x_i^t-gbest\right)^2} \quad (12)$$

其中：D 为搜索空间的维数；i 为第 i 个粒子；t 为当前迭代次数。

定义 2：种群粒子与最优粒子欧氏距离的平均值：

$$l_{ave}^{t}=\frac{1}{m}\cdot\sum_{i=1}^{m}l_{i}^{t} \tag{13}$$

其中：m 为种群的大小。

定义 3：种群粒子与最优粒子欧氏距离的方差：

$$\sigma^{t}=\frac{1}{m}\cdot\sum_{i=1}^{m}\left(l_{i}^{t}-l_{ave}^{t}\right)^{2} \tag{14}$$

定义 4：种群粒子与最优粒子欧氏距离方差的最大值：

$$\sigma_{\max}=\max_{j\in\{1,2,\cdots,t\}}\{\sigma_{j}\} \tag{15}$$

其中：j 为第 j 次迭代；σ_j 为第 j 代种群的方差。

定义 5：多样性函数为：

$$div(t)=\sigma^{t}\Big/\sigma_{\max} \tag{16}$$

当种群粒子与最优粒子欧氏距离较大时，粒子间差异性较大，种群多样性函数值较大，种群的多样性也较大，相反则多样性较小，种群多样性函数值被设定在[0,1]之间。

PSO-PIO 算法分两阶段进行寻优，第一阶段利用改进后的 PSO 进行寻优，第二阶段利用改进后 PIO 算法的地标算子进行进一步搜索。PSO-PIO 算法详细设计过程如下：第一阶段，种群按照改进后的 PSO 算法进行位置和速度的更新，不断向目的地靠近，同时采用多样性函数对种群的多样性进行实时监控，当种群多样下降到一定阈值时，算法的第一阶段终止，同时进入第二阶段的优化，此时，种群将目的地锁定在较小范围内；第二阶段，此时种群距离目的地较近，利用鸽子对周围熟悉地标的识别，采用基于地标算子的改进 PIO 算法在已经锁定的较小范围内展开全局搜索，由于搜索范围较小，种群的收敛速度加快，当满足终止条件时，终止算法。该混合算法，首先利用改进后 PSO 算法前期收敛速度快且具有较强的全局和局部搜索能力的特点，快速锁定最优解所在区域，缩小种群的搜索范围，同时对种群的多样性进行监控，防止陷入局部极值，当多样性下降到一定程度时，利用 PIO 算法的地标算子在已锁定范围内进行搜索，使种群能够快速有效的找到最优解。

PSO-PIO 算法的具体实施流程为：

Step 1 第一阶段改进 PSO 算法：初始化相关参数：种群大小 m；空间维数 D；惯性权值 ω_{min}、ω_{max}；学习因子 c_1、c_2；初始种群的选取。

Step 2 通过适应度函数，计算各粒子的适应度值，找出个体极值 *pbest* 和全局极值 *gbest*；

Step 3 按照 PSO 的更新公式，更新各粒子的位置和速度，比较粒子 *pbest* 与 *gbest* 的相似程度，通过公式计算粒子的跳跃概率 p_m，随机生成 $p_o\in[0,1]$，若 $p_m>p_o$，则粒子按照跳跃公式，跳出当前位置；否则留在当前位置；计算找出新一代的 *pbest* 和 *gbest*；

Step 4 采用多样性函数对种群的多样性进行评估，判断多样性值 $div(t)$是否小于设定的多样性阈值 div_{low}(适合取 0.65~0.5)，若小于则终止 PSO 算法，进入第二阶段，转步骤 5，否则转步骤 3。

Step 5 第二阶段改进 PIO 算法的地标算子：通过中心位置公式，计算出鸽群中心位置 x_{center}。

Step 6 按照 PIO 地标算子的更新公式，更新鸽群的中心位置 x_{center} 和各粒子的位置，增加扰动因子，重新计算鸽群粒子的位置；

Step 7 判断是否满足终止条件，若满足则终止算法，输出结果，否则转步骤 6。

4 实验仿真与分析

4.1 阈值的选取测试

为了使 PSO-PIO 算法两阶段的转换更加科学合理，本文以 Rosenbrock 测试函数为例，通过设定不同的阈值，测试阈值的设定对算法性能的影响。测试函数的维数为 30，种群规模为 100，最大迭代次数为 500。图2给出了不同阈值条件下函数最优值随迭代次数的变化曲线。

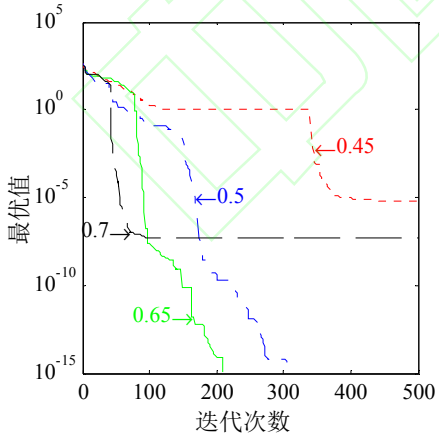


图 2 不同的阈值 Rosenbrock 函数进化曲线

图 2 中可以看出，当阈值设定为 0.5 和 0.65 时，线段经过两个阶段的快速下降，算法最终快速寻得最优值 0；当阈值设定为 0.7 时，线段中间有一大段最优值停止更新，当进入第二阶段时最优值又开始更新，但是最终仍未找到函数最优值；当阈值设定为 0.45 时，线段开始有短暂的快速下降，进入第二阶段，最优值更新一段时间后停止，最终未找到最优值；综上可知：阈值设定在 0.65~0.5 之间较为合适，此时 PSO 和 PIO 两种算法的转换合理，PSO-PIO 算法的性能良好；当阈值设定大于 0.65 时，算法第一阶段迭代次数过少，导致 PSO 算法没能起到缩小搜索范围的作用；当阈值设定小于 0.5 时，算法第一阶段迭代次数过多，后期算法停止收敛陷入局部最优。

4.2 测试函数

为了验证本文提出的 PSO-PIO 算法处理复杂函数优化问题的性能，引入了 7 个常用的经典标准测试函数[21]进行分析。7 个测试函数可以分为两组：第一组：f_1 和 f_2 是单峰函数，可以检测算法的收敛速度，f_2 是一个经典的复杂优化问题，在取值区间内走势平坦，函数只提供少量的寻优信息，所以极难找到函数的最优解，是测试算法收敛速度的极佳函数；第二组：f_3～f_7 是复杂的非线性多峰函数，具有大量的局部极值点，能够有效检验算法的全局搜索能力以及跳出局部极值的能力。表 1 中列出了函数名称、表达式、x*表示函数理论上的最小值点、$f(x^*)$ 表示函数理论上的最小值、取值范围以及实验时的维度。

算法的测试环境为：Matlab 8.3.0.532 (R2014a)为编程语言；在 Intel(R)Celeron(R) CPU 1007@ 1.50GHZ，内存 4GB 的电脑上运行程序。

表 1　7 个标准测试函数

函数名称	表达式	x^*	$f(x^*)$	取值范围	维度
Sphere	$f_1=\sum_{i=1}^{D}x_i^2$	[0,0,...,0]	0	[-100,100]	30
Rosenbrock	$f_2=\sum_{i=1}^{D}[100\cdot(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	[1,1,...,1]	0	[-30,30]	30
Rastrigin	$f_3=\sum_{i=1}^{D}[x_i^2-10\cos(2\pi x_i)+10]$	[0,0,...,0]	0	[-5.12,5.12]	30
Griewangk	$f_4=\frac{1}{4000}\sum_{i=1}^{D}(x_i^2)-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})+1$	[0,0,...,0]	0	[-600,600]	30
Ackley	$f_5=-20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2})-\exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i))+20+e$	[0,0,...,0]	0	[-30,30]	30
Schaffers	$f_6=0.5+\frac{\sin^2\sqrt{x_1^2+x_2^2}-0.5}{[1+0.001(x_1^2+x_2^2)]^2}$	[0,0]	0	[-100,100]	2
Schwefel	$f_7=418.9829\times D-\sum_{i=1}^{D}x_i\cdot\sin(\lvert x_i\rvert^{\frac{1}{2}})$	[420.96,...,420.96]	0	[-500,500]	30

4.3　对比算法与参数设置

为了证明本文 PSO 和 PIO 算法改进的必要性。本节将在基本 PSO 和基本 PIO 算法的基础上逐步增加改进策略，HPSO 是在 PSO 构架中增加跳跃算子，IPIO 是在 PIO 的构架中增加干扰算子，PSO-PIO 是将改进后的两种算法相融合的混合算法，将几种算法在相同测试环境下对相同函数的同一维数进行寻优测试。

同时为了证明 PSO-PIO 算法的性能，将该算法与一种新的 Tent 混沌人工蜂群与粒子群的混合算法(HTCAP)[22]进行对比实验。

表 2 中列出了各算法的参数设置，对比算法除种群大小其他参数与原文献相同。

表 2　算法的参数设置

算法	参数
PSO	种群大小 m：100；　惯性权值 ω：0.9；学习因子 c_1、c_2：2
PIO	种群大小 m：100；地图和指南针因数 R：0.3； 地图指南针算子最大迭代次数 tmax1：300
HPSO	种群大小 m：100；惯性因子：ω_{max}=0.9，ω_{min}=0.4；学习因子 c_1、c_2：2
IPIO	种群大小 m：100；地图和指南针因数 R：0.3； 地图指南针算子最大迭代次数 tmax1：300
HTCAP	种群大小 m：100；　粒子群：50；　引领蜂蜜：25；　跟随蜂：25
PSO-PIO	种群大小 m：100；惯性因子：ω_{max}=0.9，ω_{min}=0.4； 学习因子 c_1、c_2：2；　多样性阈值：div_{low}=0.6

4.4　实验结果及分析

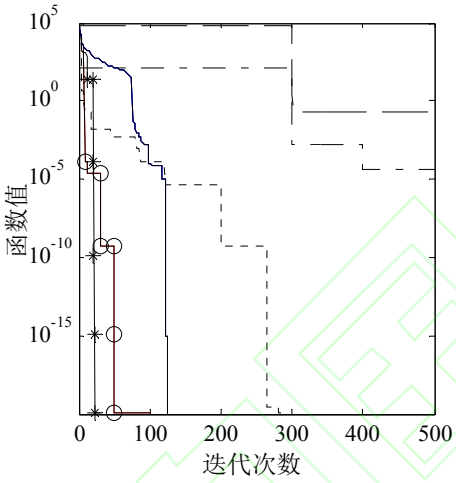
通过对每个测试函数独立运行 30 次所得优化结果的最好值、最差值、平均值、标准差来评价算法的性能。PSO、PIO、EPSO、改进 PIO、HTCAP 和本文 PSO-PIO 算法性能对比结果如表 3 所示，其中每个函数的最好优化结果加粗表示。

表 3　算法收敛精度性能对比

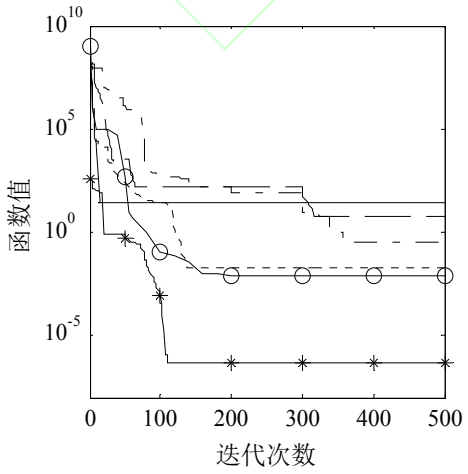
函数	算法	最好值	最差值	平均值	标准差
f_1	PSO	**0**	**0**	**0**	**0**
	PIO	1.26e-1	8.85	4.01	2.40e-1
	HPSO	**0**	**0**	**0**	**0**
	IPIO	2.13e-6	5.15e-2	8.70e-3	1.58e-2
	HTCAP	**0**	**0**	**0**	**0**
	PSO-PIO	**0**	**0**	**0**	**0**
f_2	PSO	3.84	1.42e+2	2.87e+1	9.26
	PIO	5.27	4.57e+2	5.25e+1	2.81e+2
	HPSO	4.54e-1	1.01	7.14e-1	2.14e-1
	IPIO	5.24e-1	1.67	9.28e-1	6.91e-1
	HTCAP	1.88e-3	8.91e-3	6.47e-3	8.45e-2
	PSO-PIO	**2.92e-11**	**6.22e-2**	**3.03e-3**	**9.31e-2**
f_3	PSO	1.44e+2	2.23e+2	1.93e+2	3.86e+2

	PIO	1.30e-2	6.55e-2	3.08e-2	1.15e-2
	HPSO	0	1.23e-16	5.29e-17	3.73e-17
	IPIO	2.08e-5	1.93e-4	8.37e-5	8.53e-4
	HTCAP	**0**	**0**	**0**	**0**
	PSO-PIO	**0**	**0**	**0**	**0**
f_4	PSO	3.64e-1	9.95e-1	7.82e-1	2.51e-1
	PIO	2.00e-3	3.25e-1	1.64e-1	5.36e-2
	HPSO	1.04e-5	2.63e-3	7.94e-4	1.09e-4
	IPIO	2.14e-3	6.07e-3	3.16e-3	2.72e-4
	HTCAP	0	1.109e-16	9.250e-17	3.85e-17
	PSO-PIO	**0**	**0**	**0**	**0**
f_5	PSO	2.82e-2	5.86	2.60	8.92e-1
	PIO	2.51e-2	6.59e-2	3.70e-2	2.19e-2
	HPSO	2.41e-4	4.62e-3	1.00e-4	8.99e-3
	IPIO	4.95e-4	5.87e-3	1.94e-3	5.28e-3
	HTCAP	5.99e-10	2.51e-9	9.13e-10	1.26e-9
	PSO-PIO	**2.08e-16**	**8.79e-16**	**6.88e-16**	**6.65e-16**
f_6	PSO	**0**	**0**	**0**	**0**
	PIO	6.83e-7	1.93e-4	1.97e-5	3.68e-5
	HPSO	**0**	**0**	**0**	**0**
	IPIO	**0**	**0**	**0**	**0**
	HTCAP	**0**	**0**	**0**	**0**
	PSO-PIO	**0**	**0**	**0**	**0**
f_7	PSO	3.82e-2	4.20e+1	1.83e-1	3.54e-1
	PIO	6.30e-2	1.85	8.06e-1	1.83e-1
	HPSO	8.19e-4	4.25e-1	5.61e-2	4.28e-1
	IPIO	1.49e-2	5.77e-2	3.57e-2	2.73e-2
	HTCAP	6.31e-3	7.541e-2	1.58e-2	4.19e-2
	PSO-PIO	**3.82e-4**	**3.82e-4**	**3.82e-4**	**6.94e-16**

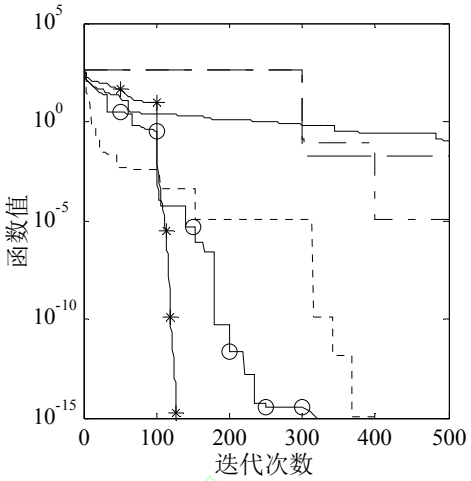
从表 3 中可以看出：对于单峰函数，其中函数 f_1 较简单，因此除了 PIO 算法和改进 PIO，其他算法均能寻到最优解，性能相同；另外由于 f_2 属于非凸、病态单峰函数，因此全局搜索能力更重要，PSO-PIO 算法寻到比其它算法优的结果，但是标准差较大，寻优结果稳定性不够，其全局搜索能力还有待提高。对于多峰函数，PSO-PIO 均能寻到最优解或者精度较高的解，且标准差较小，特别是函数 f_5 和 f_7 明显优于其他算法。由对比结果可知，PSO-PIO 算法在收敛精度上有明显提高，由此说明该算法针对复杂函数优化问题的有效性和适用性。



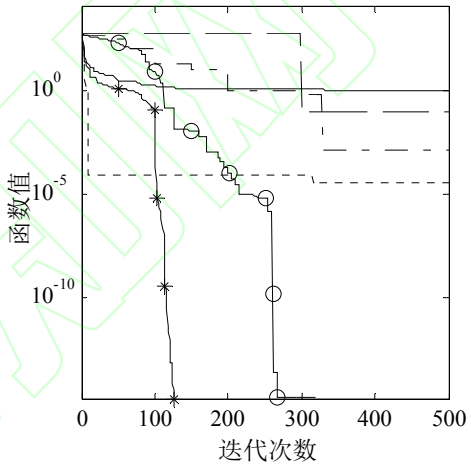
f_1 函数



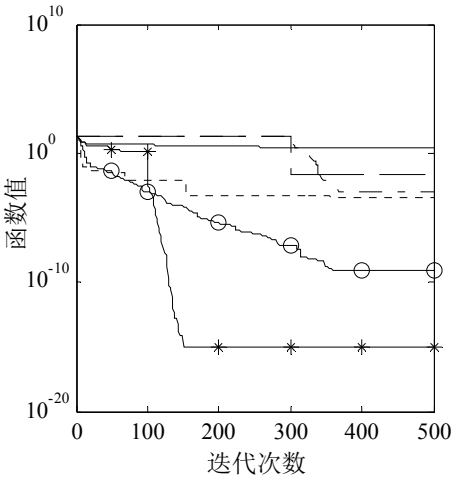
f_2 函数



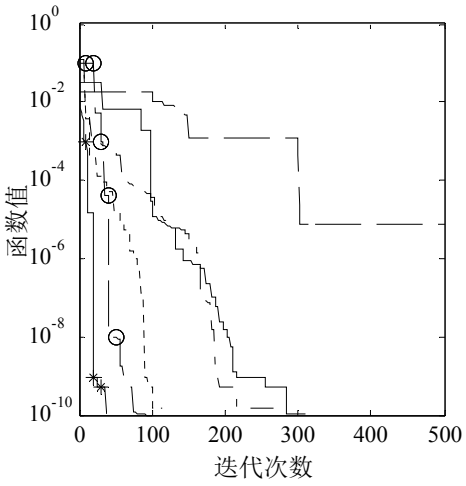
f_3 函数



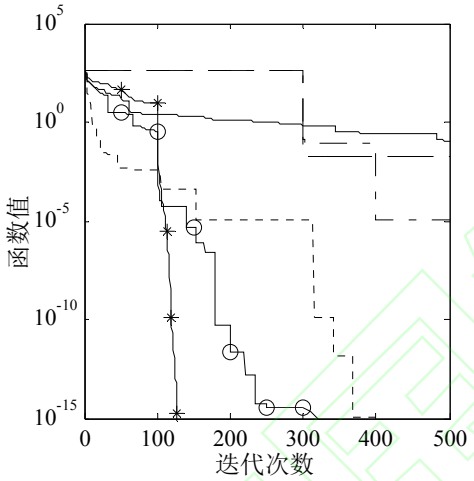
f_4 函数



f_5 函数



f_6 函数



f_7 函数

图 3　f_1～f_7 函数进化曲线

PSO　　PIO

HPSO　　IPIO

HTCAP　　PSO-PIO

图 3 显示了 7 个测试函数随着迭代次数种群最优解的变化情况，横坐标是迭代次数，纵坐标是最优解的对数。从上图中可以直观的看出，PSO 和 HPSO 算法的对比，当 PSO 已经陷入局部最优停止收敛时，HPSO 继续搜索寻找更优的解，由此可以说明跳跃算子跳出局部极值的有效性。同时，由图可以看出 IPIO 也能在 PIO 停止收敛时继续搜索，由此可以说明干扰算子能够有效的增加种群的多样性，使种群有机会寻找到更优的解。将 PSO-PIO 与 HPSO、IPIO 算法上述收敛图形进行对比可以看出，PSO-PIO 不仅能够得到比其他算法更优的解，且收敛速度更快，由此可以说明将 HPSO、IPIO 两种算法结合的必要性。从图 3 中将 PSO-PIO 与 HTCAP 算法进行相比，可以明显看出 PSO-PIO 能够更加快速收敛到更优的解。综上所述，本文提出的 PSO-PIO 算法有效的提高了复杂函数优化问题求解的精度以及收敛速度。

图中 PSO-PIO 算法的曲线总是在出现一段平缓后，在某一迭代次数时，出现较大的下降趋势（其中函数 f_2、f_3、f_4、f_5、f_7 较明显），此时算法进入第二阶段进行寻优。由此可以说明 PSO-PIO 当第一阶段算法陷入局部极值时，通过多样性函数的有效监控，进入第二阶段从而跳出局部极值，此时已经将最优解锁定在较小范围内，算法能够快速找到最优解。

5 结论

（1）针对复杂函数优化问题的特点以及 PSO 和 PIO 算法各自的优缺点，提出了一种基于改进粒子群和鸽群的混合优化算法，分别采用改进 PSO 算法和改进 PIO 算法进行寻优，既保留了两种算法的优点又规避了缺点，使算法能够快速有效的寻到复杂函数的最优解或较优解。

（2）PSO-PIO 混合优化算法对 7 个经典标准测试函数进行优化，并与其他算法从收敛速度和收敛精度两个评价指标进行了对比，结果表明本文算法在收敛速度、收敛精度及稳定方面都有明显的优势，但是对于信息较少且走势平坦的一类单峰复杂函数稳定性还有待提高，这将是下一步的研究方向。

参考文献：

[1] Eberhart R, Kennedy J. New optimizer using particle swarm theory [C]// Proc of the 6th International Symposium on Micro Machine and Human Science. Piscataway, NJ, USA: IEEE Press, 1995: 38-43.

[2] 胥小波, 郑康锋,李丹,等. 新的混沌粒子群优化算法[J]. 通信学报,2012, 33(01):24-30+37.

[3] Jia D L,Zheng G X, Qu B Y, et al. A hybrid particle swarm optimization algorithm for high-dimensional problems[J]. Computers and Industrial Engineering (S0360-8352), 2011, 61(4): 1117-1122.

[4] 曾辉, 王倩, 夏学文,等. 基于自适应多种群的粒子群优化算法[J]. 计算机工程与应用, 2018, 54(10): 59-65.

[5] 郑庆新，顾晓辉，张洪铭. 基于 SQP 和自适应搜索的混沌粒子群算法[J]. 计算机工程与应用，2018, 54(13): 131-136.

[6] 张其亮, 陈永生. 基于混合粒子群-NEH 算法求解无等待柔性流水车间调度问题[J]. 系统工程理实践, 2014, 34(03):802-809.

[7] 李荣雨,周志勇.成长性的粒子群算法及其在函数优化中的应用[J].信息与控制,2017,46(02):224-230.

[8] 夏立荣, 李润学,刘启玉,等. 基于动态层次分析的自适应多目标粒子群优化算法及其应用[J]. 控制与决策, 2015, 30(2): 215-221.

[9] Duan H B，Qiao P X. Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning[J]. International Journal of Intelligent Computing and Cybernetics,2014,7(1): 24-37.

[10] Duan H B，Qiu H X，Fan Y M. Optimal combat coordination control for unmanned aerial vehicle using predator-escaped pigeon[J]. Chinese Science TechnologicalSciences,2015,45(6):559-572.

[11] Zhang X，Duan H B，Yang C. Pigeon-inspired optimization approach to multiple UAVs formation reconfiguration controller design [C] // Guidance，Navigation and Control Conference. Yantai: IEEE,2014:2707-2712.

[12] Zhang S,Duan H.Gaussian pigeon-inspired optimization approach to orbital spacecraft formation reconfiguration [J]. Chinese Journal of Aeronautics, 2015,28(01):200–205.

[13] Li C, Duan H. Target detection approach for UAVs via improved Pigeon-inspired Optimization and Edge Potential Function[J]. Aerospace Science & Technology, 2014, 39:352-360.

[14] 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 [J]. 中国科学:技术科学,2015, 45(6):559-572.

[15] 张亚平,孙佩华,李昱辉, 等. 基于改进鸽群算法的高超声速飞行器轨迹优

化[J].飞行力学,2017, 35(04):60-64.

[16] 蒋飘蓬,周凯,朱乾坤,等. 采用威胁启发鸽群优化的武装直升机航路规划[J]. 电光与控制, 2017, 24(07):57-61.

[17] 周凯,姜文志,陈邓安,等. 基于改进鸽群优化的直升机协同目标分配[J]. 火力与指挥控制, 2017, 42 (7): 94-98 +104.

[18] Deng Y, Duan H. Control parameter design for automatic carrier landing system via pigeon-inspired optimization[J]. Nonlinear Dynamics, 2016, 85(1):1-10.

[19] Zhao J, Zhou R. Pigeon-inspired optimization applied to constrained gliding trajectories[J]. Nonlinear Dynamics, 2015, 82(4):1-15.

[20] Eberhart R C, Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization[C]. Proc of the 2000 Congressional Evolutionary Computation. Piscataway: IEEE, 2000: 84-88.

[21] 钱锋.粒子群算法及其工业应用[M]. 北京: 科学出版社,2013.101-104.

[22] 匡芳君,金忠,徐蔚鸿,等. Tent 混沌人工蜂群与粒子群混合算法[J].控制与决策,2015,30(05):839-847.