中国科学: 技术科学 2023 年 第 53 卷 第 2 期: 268 ~ 279
SCIENTIA SINICA Technologica techcn.scichina.com
《中国科学》杂志社 SCIENCE CHINA PRESS
论 文 群体智能激发汇聚及应用专辑

平衡探索与利用的广义鸽群优化算法

程适[1], 张明明[1], 史玉回[2*], 路辉[3], 雷秀娟[1], 王锐[4]

1. 陕西师范大学计算机科学学院, 西安 710119;
2. 广东省类脑智能计算重点实验室(南方科技大学), 深圳 518055;
3. 北京航空航天大学电子信息工程学院, 北京 100191;
4. 国防科技大学系统工程学院, 长沙 410073
*E-mail: shiyh@sustech.edu.cn

收稿日期: 2021-08-14; 接受日期: 2021-11-04; 网络版发表日期: 2022-08-03
国家自然科学基金项目(批准号: 61806119)、广东省重点实验室项目(编号: 2020B121201001)、国家自然科学优秀青年基金项目(批准号: 62122093)、季华实验室项目(编号: X210101UZ210)、中央高校基本科研业务费专项资金项目(编号: GK202003078)和陕西师范大学研究生创新团队项目课题(编号: TD2020014Z)资助

摘要 为了平衡鸽群优化算法的探索与利用能力, 本文提出了一种广义鸽群优化算法. 传统的鸽群优化算法包含两种优化算子, 分别为地图与指南针算子和地标算子. 这两种算子依次执行, 在一次算法运行中, 仅执行一轮迭代. 在广义鸽群优化算法中, 将算法搜索分为多个阶段, 每个阶段分别执行两种算子. 在算法的一次运行中, 两种算子执行多轮. 地图与指南针算子侧重于算法的探索能力, 而地标算子侧重于算法的利用能力. 改进算法仅改变了两种算子的执行顺序, 无需增加额外的函数值计算. 此外, 广义鸽群优化算法扩展了解集合结构和算子参数设置, 这对于提高算法的搜索质量大有裨益. 在11个单目标测试函数和8个多模态优化测试函数上进行仿真对比试验, 结果表明广义鸽群优化算法提高了鸽群优化算法的搜索效率, 改进了算法的搜索结果.

关键词 群体智能, 鸽群优化算法, 探索与利用, 多模态优化

1 引言

群体智能(swarm intelligence)的核心思想是, 若干个简单个体构成一个群体, 个体间通过合作、竞争、认知与学习等机制表现出高级和复杂的功能, 在缺少局部信息和模型的情况下, 仍能够完成对复杂问题的建模、学习和优化求解. 广义的群体智能可以分为四个方面: 基于群体智能现象的仿真与建模、群体优化算法、群体学习算法、群体智能的应用. 狭义的群体智能指群体智能优化算法, 是一类利用群体智能来高效求解优化问题的方法. 这类算法求解优化问题的模式为随机初始化求解变量, 计算各个解对应的目标函数值, 利用多个解之间的竞争和协作, 不断迭代更新解集合, 直到算法找到符合需求的解或者达到最大迭代次数. 算法具有不依赖于梯度信息, 对待求解问题无连续、可导等要求的特点, 使得该类算法既适应连续型数值优化问题, 也适应离散型组合优化问题. 同时, 群体智能优化算法潜在的并行性和分布式特点使

引用格式: 程适, 张明明, 史玉回, 等. 平衡探索与利用的广义鸽群优化算法. 中国科学: 技术科学, 2023, 53: 268–279
Cheng S, Zhang M M, Shi Y H, et al. Generalized pigeon-inspired optimization algorithm for balancing exploration and exploitation (in Chinese). Sci Sin Tech, 2023, 53: 268–279, doi: 10.1360/SST-2021-0371

其在处理大数据时具备显著优势. 因此, 群体智能优化算法受到各个领域越来越多的关注, 成为一个热门的重要研究方向.

鸽群优化(pigeon-inspired optimization, PIO)算法是一种典型的群体智能优化算法, 算法设计基于鸽群独特的导航能力[1,2]. 目前, 群体智能优化算法在大多数低维单目标优化问题上取得了较好的优化结果, 然而对于多模态优化问题, 尚未获得良好的优化结果. 多模态优化的目标是在算法的一次搜索过程中, 尽可能寻找到多个极值解, 并将这些极值解保持到搜索结束. 多模态优化问题的难点在于如何平衡算法的求解精度和满足要求解的个数之间的矛盾. 一味满足精度的要求, 算法易陷入单个解中, 解的个数不足; 而过于强调解的个数, 解集合的精益求精能力不足, 优化解难以达到满意的精度.

相同算法在不同的优化问题上求解性能差异明显, 如何利用搜索中获得的信息, 将问题的结构知识与优化算法的搜索状态相结合, 是提高优化算法性能的重要途径. 在群体智能优化算法中, 可以根据求解信息, 调整算法的探索(exploration)能力与利用(exploitation)能力, 从而提高算法的搜索性能. 使用过多的算法搜索信息, 即过于侧重算法的利用能力, 算法容易陷入局部最优解中; 而相应地, 使用过少的算法搜索信息, 即过于侧重算法的探索能力, 算法的搜索速度较慢, 搜索效率低下. Ishii等人[3]研究了强化学习中探索与利用的参数控制方法; Alba和Dorronsoro[4]研究了细胞遗传算法搜索模型中的探索与利用比率; Lin和Gen[5]提出了一种自动调节策略来平衡演化算法的探索与利用能力; Črepinšek与Hussain等人[6,7]分别总结了演化算法和基于群体的元启发式算法中的探索与利用能力; Li和Tan[8]总结了烟花算法中探索能力与利用能力的平衡策略.

本文提出了一种平衡探索能力和利用能力的广义鸽群优化(generalized pigeon-inspired optimization, GPIO)算法, 并将算法应用于多模态优化问题的求解. 本文组织结构如下: 第2部分介绍了本文的预备知识, 包括群体智能、鸽群优化算法和多模态优化的基础知识; 第3部分提出了一种平衡探索能力和利用能力的改进鸽群优化算法, 并改进了解集合结构和参数设置; 第4部分给出了改进鸽群优化算法在求解单目标优化问题和多模态优化问题上的实验结果; 第5部分是本文结论和未来的研究方向.

2　背景知识

本节对群体智能、鸽群优化算法和多模态优化问题进行简要介绍.

2.1　群体智能

广义群体智能指基于多个简单个体组成的群体, 利用群体间的协作、竞争、认知与学习模式, 完成对复杂问题的学习和优化求解. 群体智能包括四个方面的内容: 基于群体智能现象的仿真和建模(swarm simulation)[9]、基于群体的智能优化算法(swarm optimization)、基于群体的智能学习算法(swarm learning)[10]和基于群体智能的应用, 如群体机器人[11]、基于群体智能的协同通讯模型和协同感知技术[12].

群体优化算法是群体智能中的热门研究领域, 以粒子群优化算法和蚁群优化算法为起始, 基于物理学现象、生物学现象和人类群体行为的大量新算法被提出, 其中代表性算法包括烟花算法(fireworks algorithm)[8]、头脑风暴优化算法(brain storm optimization)[13,14]和鸽群优化算法[2,15]. 群体智能方法获得了广泛的研究和应用, 如基于群体智能的协同通讯模型和协同感知技术[12]、群体机器人的参数控制模型优化[16]、多机器人室内环境地图构建[17]等.

2.2　鸽群优化算法

受自然界中鸽群归巢行为的启发, 鸽群优化算法模仿了鸽群在不同阶段使用不同导航工具的行为. 算法1给出了鸽群优化算法的基本流程. 图1是鸽群优化

算法1　鸽群优化算法基本流程

输入: N_c=1, 随机生成N_p个初始个体(初始解), 并分别计算所有个体的函数值;

while未达到预先设定的最大迭代次数do
　if 迭代次$N_c \leqslant N_{c1}^{\max}$ **then**
　　利用地图与指南针算子优化解集合, 更新X_g, N_c递增;
　end if
　if迭代次数$N_{c1}^{\max} < N_c \leqslant N_{c2}^{\max}$**then**
　　利用地标算子优化解集合, 更新X_g, N_c递增;
　end if
end while

输出: 算法找到的函数$f()$全局最优解: X_g.

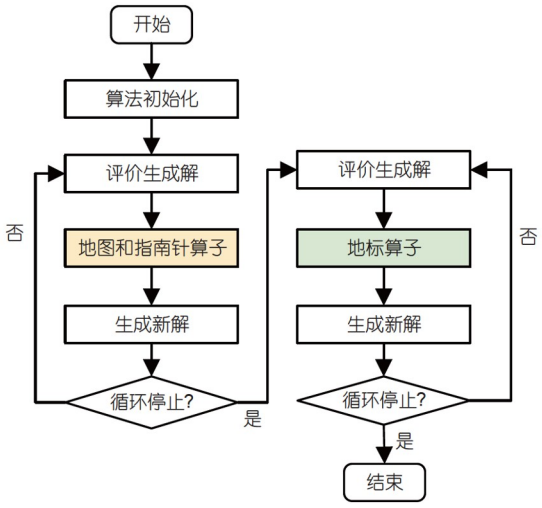

图 1 (网络版彩图)鸽群优化算法的基本流程图

Figure 1 (Color online) Flowchart of the basic pigeon-inspired algorithm.

算法的基本流程图. 在初始算法中, 算法的运行过程分为两个部分, 使用了两种算子, 分别为地图与指南针算子(map and compass operator)和地标算子(landmark operator). 两种算子依次执行, 其中地图与指南针算子先执行$N_{c1}^{\max}$次, 接着地标算子执行$N_{c2}^{\max}-N_{c1}^{\max}$次. 群体智能优化算法需要考虑搜索过程中探索能力和利用能力的平衡, 而两种算子对于这两种能力的侧重不同.

地图与指南针算子模仿了鸽群的远距离搜索方式[1,18], 第i个解的移动步长V_i的计算方式为

$$V_i^{N_c}=V_i^{N_c-1}\cdot \mathrm{e}^{-R\times N_c}+rand\cdot(X_{\mathrm{g}}-X_i^{N_c-1}), \tag{1}$$

其中, $R\in(0,1)$是地图与指南针因子; N_c是当前迭代次数; $rand$是(0,1)间的随机数; $V_i^{N_c}$表示在第N_c次迭代时第i个解的移动步长; X_g是当前解集合中的全局最优解, 即$f(X_g)$是当前的最佳函数优化值.

新解位置X_i的计算方式为

$$X_i^{N_c}=X_i^{N_c-1}+V_i^{N_c}, \tag{2}$$

其中, $X_i^{N_c}$表示在第N_c次迭代时第i个解的位置, 即决策变量的取值.

地标算子模拟了鸽群的近距离搜索方式[1,18]. 在迭代过程中, 选取部分较好的解, 将这些解的中心$X_c^{N_c-1}$作为地标, 所有解朝着地标方向进行搜索. 地标算子计算方式为式(3)~(5):

$$X_c^{N_c-1}=\frac{\sum_{i=1}^{N_p^{N_c-1}}X_i^{N_c-1}\cdot F\left(X_i^{N_c-1}\right)}{N_p^{N_c-1}\sum_{i=1}^{N_p^{N_c-1}}F\left(X_i^{N_c-1}\right)}, \tag{3}$$

$$N_p^{N_c}=\frac{N_p^{N_c-1}}{2}, \tag{4}$$

$$X_i^{N_c}=X_i^{N_c-1}+rand\cdot(X_c^{N_c-1}-X_i^{N_c-1}), \tag{5}$$

其中, $F(X_i^{N_c-1})$是一个评估函数值, 用来衡量在第N_c-1次迭代时, 当前解集合中解X_i产生的函数值$f(X_i^{N_c-1})$的优劣程度. 优化解的函数值$f(X_i)$越好, $F(X_i)$值越大, 即$F(X_i)$与最大化问题的函数值成正比, 与最小化问题的函数值成反比. $N_p^{N_c}$为第N_c次迭代时的种群规模.

在算法设计上, 鸽群优化算法发展出不同的变体. 例如, Cheng等人[19]提出了一种具有不同拓扑结构的改进鸽群优化算法, 并将环形(ring)结构应用于地图与指南针算子; Yang等人[20]提出了一种基于二进制和实数混合编码的改进鸽群优化算法; Xiang等人[21]提出了一种带有禁忌列表的综合学习鸽群优化方法; Duan等人提出了Predator-Prey鸽群优化算法[22]和变异多目标鸽群优化算法[23]等改进方法.

鸽群优化算法应用于多种问题, 在求解新型优化问题方面, 算法应用于求解超多目标优化问题[24]和多目标多模态优化问题[25]等. 在实际工程问题中, 改进算法应用于检测蛋白质复合物问题[26]; 基于捕食逃逸的改进鸽群优化算法应用于无人机紧密编队协同控制问题[2]; 基于莱维飞行的鸽群优化算法应用于仿雁群无人机编队控制器设计问题[27]; 动态离散鸽群优化算法应用于求解无人机群体的协同搜索攻击任务规划问题[28]; 基于参数知识鸽群算法应用于求解离散车间能效优化问题[29]等.

2.3 多模态优化

传统的单目标优化的求解目标是找到优化问题的全局最优解, 最优解的个数往往只有一个. 而多模态优化(multimodal optimization)的目标是在算法的一次搜索过程中, 尽可能寻找到多个极值解(包括全局最优和局部最优), 并将这些极值解保持到搜索结束. 问题优化解的个数不确定性、解分布的不确定性等特征, 决定了多模态问题的复杂性.

多模态优化已经应用于多种科学和工程问题中,

如求解非线性方程组问题就是一个典型应用，需要一次运行得到待求解方程组的多个根. 改进环拓扑混合群体智能算法是求解非线性方程组的一种有效方法[30].

对于单目标优化问题，解的评价仅需考虑解的精度. 在多模态优化中，解集合的评价需考虑两个方面的因素，解的精度和符合精度的解的多样性. 常用的评价指标为多次运行后符合要求的不同极值解的个数(number of optima found, NPF)和找到的极值解占所有极值解的比率(optima ratio, PR). 极值解比率计算方式如下:

$$\mathrm{PR}=\frac{\sum_{\mathrm{run}=1}^{\mathrm{NR}}\mathrm{NPF}_i}{\mathrm{NKP}\times\mathrm{NR}}=\frac{\mathrm{NPF}}{\mathrm{NKP}\times\mathrm{NR}},\tag{6}$$

其中，NPF_i指第i次运行中，所有找到的极值解的数目，NR为算法运行的次数(number of runs)，NKP为测试函数已知的极值解数目(number of known optima).

3　改进鸽群优化算法

经典的群体智能优化算法以模型驱动的方法为主, 待求解问题被建模为“黑盒”优化问题, 算法不需要待求解问题的结构信息，仅使用解的函数优化值作为求解的引导信息. 算法的探索能力为算法对未知区域的搜索能力，体现了搜索的广度，而利用能力指算法不断搜索可能存在较好解的区域，体现了搜索的精度. 如何在搜索的过程中, 不断平衡这两种能力, 是在算法设计时首先需要考虑的问题. 本节提出了一种平衡探索能力与利用能力的广义鸽群优化算法.

3.1　广义鸽群优化算法

在图1鸽群优化算法的基本流程中, 鸽群优化算法依次执行了两种算子. 在算法的一次运行中, 两种算子各执行一轮迭代. 在算法的运行前期, 采用了地图与指南针算子, 目的是找到优化解存在的大概区域; 而在算法的运行后期, 采用了地标算子, 目的是对于搜索得到的解集合进行“精益求精”.

图2是广义鸽群优化算法的流程图，算法2为广义鸽群优化算法的基本流程. 改进算法的搜索分为多个阶段，每个阶段分别执行两种优化算子. 在算法的一次运行中, 两种算子进行多轮迭代. 其中, 地图与指南针算子侧重于算法的探索能力，而地标算子侧重于算法的利用能力. 每个阶段分别执行两种算子. 改进算法通过算子的不断交替，调整算法探索和利用能力, 在算法的一次运行中, 两种算子执行n轮(n>1). 改进算法仅改变了两种算子的执行顺序，无需增加额外的函数值计算.

算法2　广义鸽群优化算法基本流程

输入: N_c=1, 随机生成N_p个初始个体(初始解), 并分别计算所有个体的函数值, 算子执行n轮, 初始化n=1;

while未达到预先设定的最大迭代次数**do**
　if迭代次数$(n-1)\times N_{c2}<N_c\leqslant N_{c1}+(n-1)\times N_{c2}$**then**
　　利用地图与指南针算子优化解集合, 更新X_n,$N_c=N_c+1$;
　end if
　if迭代次数$\mathrm{N}_{c1}+(n-1)\times N_{c2}<N_c\leqslant n\times N_{c2}$**then**
　　利用地标算子优化解集合, 更新X_n, $N_c=N_c+1$;
　end if
　if一组算子运行结束**then**
　　进入下一轮迭代, $n=n+1$;
　end if
end while

输出: 算法找到的函数$f()$符合要求的优化解集合.

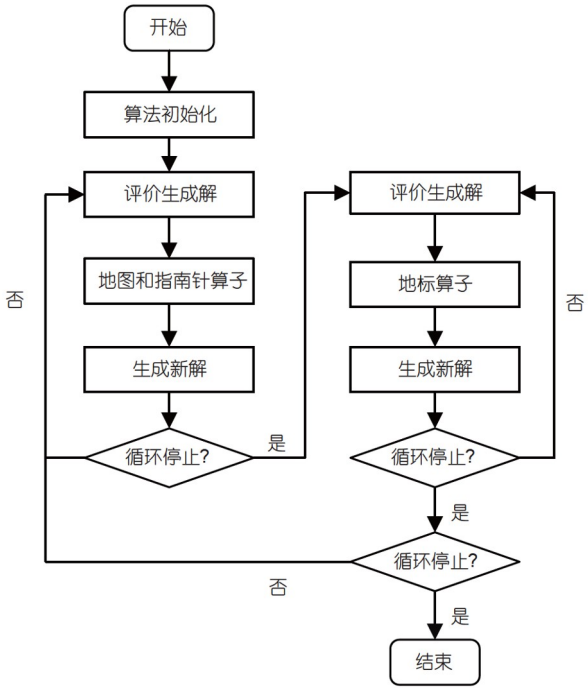

图 2　广义鸽群优化算法的流程图

Figure 2　Flowchart of generalized pigeon-inspired algorithm.

3.2　改进地图与指南针算子

鸽群优化算法包含两种算子模型，在优化中仅需

设置算法的参数，在一次求解中，两种算子依次执行各一轮. 在地图与指南针算子中，如式(1)所示，算法使用了当前解集合的最优解X_g信息；在地标算子中，如式(3)所示，算法使用了解集合的加权中心位置X_c信息.

在初始地图与指南针算子中，X_g为全局最优解信息，而多次使用全局最优解信息，算法容易陷入局部最优. 在改进地图与指南针算子中，为解集合添加新的结构，将全局最优解信息X_g更改为局部最优解信息X_n，X_n为解的邻域最优解，可以减缓算法的收敛速度，保持解集合的多样性[19].

3.3 改进地标算子

在初始地标算子中，式(4)为种群规模的缩减方式. 图3给出了三种设置下的缩减速率示例. 由图可知，在初始的鸽群优化算法中，k设置为2，当算法种群规模N_p=1024时，地标算子进行10代后，种群规模缩减为1. 而当k=1.5时，在10次迭代后，算法种群由58缩减为1 (1.5^{10}=57.665)；当k = 1.25时，在10次迭代后，算法种群由10缩减为1(1.25^{10} = 9.313). 在优化算法求解实际问题中，种群规模N_p常小于1000，k值过快的缩减为1，则$X_c^{N_c-1}$退化为当前全局最优解$X_g^{N_c-1}$，算法将快速收敛于$X_g^{N_c-1}$附近，解集合中所有解聚集于一点，算法易陷入局部最优，从而降低搜索效率.

为了增加解集合的多样性，降低解集合的收敛速度，式(4)扩展为式(7):

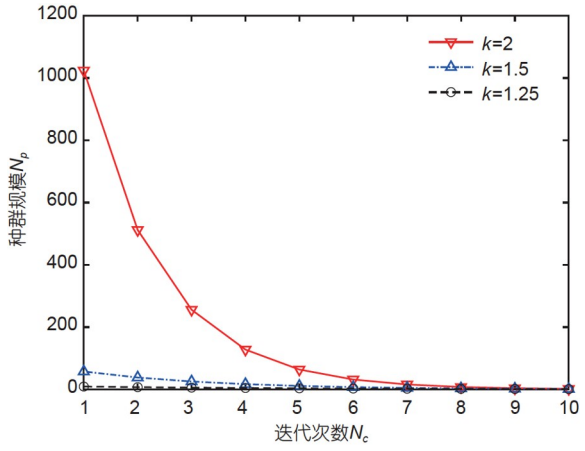

图 3 (网络版彩图)地标算子中种群规模N_p的缩减速率示例
Figure 3 (Color online) Decreasing ratio of N_p in landmark operator.

$$N_p^{N_c} = \frac{N_p^{N_c-1}}{k}, \tag{7}$$

其中，k为缩减因子. 缩减因子k的设置决定了地标算子中解集合的缩减速率. 采用较小的k值(如k = 1.25)，或将迭代后的种群规模N_p维持在固定数值，可以避免种群规模过快缩减. 在具有简化(simplified)地标算子的鸽群优化算法中，种群规模N_p固定为初始种群规模的一半.

4 对比实验与结果分析

为了检验提出的广义鸽群优化算法的搜索效率，实验采用了11个单目标优化测试函数和8个多模态测试函数，所选取的函数与文献[19]中一致.

4.1 算法参数设置

在实验中，我们采用了6种算法，分别为3种鸽群优化算法(PIO，PIOr，PIOrs)和3种广义鸽群优化算法(GPIO，GPIOr，GPIOrs). 符号‘G’指广义鸽群优化算法，两种算子在一次运行中执行n轮；符号‘r’指地图与指南针算子中，解集合采用环形结构；符号‘s’指简化地标算子，在地标算子中，种群规模固定为原有种群的一半. 例如，‘GPIOrs’指广义鸽群优化算法，采用了环形解结构和简化地标算子.

在实验中，算法均采用了相同的迭代次数和共同参数设置，地图与指南针因子R = 0.2，初始种群规模N_p = 100，改进地标算子中，种群规模缩减因子k = 1.25. 所有优化问题均执行50次.

4.2 单目标优化问题

实验采用了11个单目标最小化测试函数，如表1所示，其中f_1~f_5为5个单峰问题，f_6~f_{11}为6个多峰问题. 测试函数均为50维优化问题，所有算法迭代1000次. 鸽群优化算法中迭代次数$N_{c1}^{\max}$ = 900，$N_{c2}^{\max}$ = 1000；广义鸽群算法中，N_{c1} = 180，N_{c2} = 200，n = 5.

表2给出了6种算法在单峰问题的求解结果，表3给出了6种算法在多峰问题的求解结果. 表格中，结果加粗部分为实验中6种算法的最优结果. 实验结果显示，在50次测试中，GPIO算法找到的最优解和平均解，都优于对应的PIO算法，这表明GPIO算法明显改善了PIO

表 1　单目标优化测试函数

Table 1　Eleven benchmark functions for single objective optimization

Function	Function name	Dimension n	Search space	$f_{\min}$
f_1	Sphere	50	$[-100,100]^n$	−450.0
f_2	Schwefel's P2.22	50	$[-10,10]^n$	−330.0
f_3	Schwefel's P1.2	50	$[-100,100]^n$	−450.0
f_4	Step	50	$[-100,100]^n$	330.0
f_5	Quadric noise	50	$[-1.28,1.28]^n$	−450.0
f_6	Rosenbrock	50	$[-10,10]^n$	−330.0
f_7	Rastrigin	50	$[-5.12,5.12]^n$	120.0
f_8	Noncontinuous Rastrigin	50	$[-5.12,5.12]^n$	330.0
f_9	Ackley	50	$[-32,32]^n$	−330.0
f_{10}	Griewank	50	$[-600,600]^n$	−450.0
f_{11}	Generalized Penalized	50	$[-50,50]^n$	180.0

表 2　单峰单目标测试函数优化结果

Table 2　Result comparisons on five unimodal single objective optimization problems

Algo.	Best	Mean	std. dev.	Best	Mean	std. dev.	Best	Mean	std. dev.
		f_1			f_2			f_3	
PIO	−450	2553.6026	900.736	−330	−330	4.88E−14	2452.4958	19129.578	7906.689
PIOr	−446.6151	9541.1378	2081.0016	−329.532	−302.384	4.89391	4257.812	29019.223	7776.601
PIOrs	−450	**−450**	0	−330	**−330**	0	3129.324	28953.727	8762.280
GPIO	−450	−449.9999	9.27E−14	−330	−330	3.21E−14	−450	**8787.922**	4572.892
GPIOr	−450	−450	5.42E−13	−330	−330	1.39E−14	724.6366	12988.222	3220.161
GPIOrs	−450	**−450**	0	−330	**−330**	0	1595.000	11884.687	3130.817
		f_4			f_5				
PIO	330	3305.58	919.878	−449.9999	−449.9999	0.00010			
PIOr	2053	22006.72	4148.972	−449.9999	−449.7388	0.10033			
PIOrs	330	**330**	0	−449.9999	−449.9999	9.75E−05			
GPIO	330	330.1	0.3	−449.9999	−449.9999	1.09E−05			
GPIOr	330	330.26	0.6575	−449.9999	**−449.9999**	1.27E−05			
GIPOrs	330	**330**	0	−449.9999	−449.9999	1.35E−05			

算法的搜索效率. 同时, GPIO算法在50次测试时, 解的标准差显著小于PIO算法, 表明了GPIO算法的搜索稳定性.

图4展示了算法求解时, 算法误差逐步减少过程, 其中, 横轴表示算法共执行1000次迭代; 纵轴表示算法的求解误差, 即所找到的最优解的函数值$f(X^*)$与$f_{\min}$之差. 由图可知, GPIO的搜索效率明显优于PIO算法, 在较少的迭代次数中, 求解误差下降的速度更快.

在图4中, GPIO的求解误差曲线在大部分函数上呈现阶梯状, 即每隔200代, 求解误差呈现出明显下降趋势. 这是由于实验中GPIO算法分为5个阶段, 每个阶段地图与指南针算子执行180次, 地标算子执行20次. 图4清晰地展示了两种算子的不同搜索能力, 说明算法侧重于探索和利用能力的交替.

4.3　多模态优化问题

表4列出了实验所采用的多模态最大化测试函数[18,31]. 所有算法迭代500次, 求解精度$\epsilon=10^{-3}$. 鸽群优

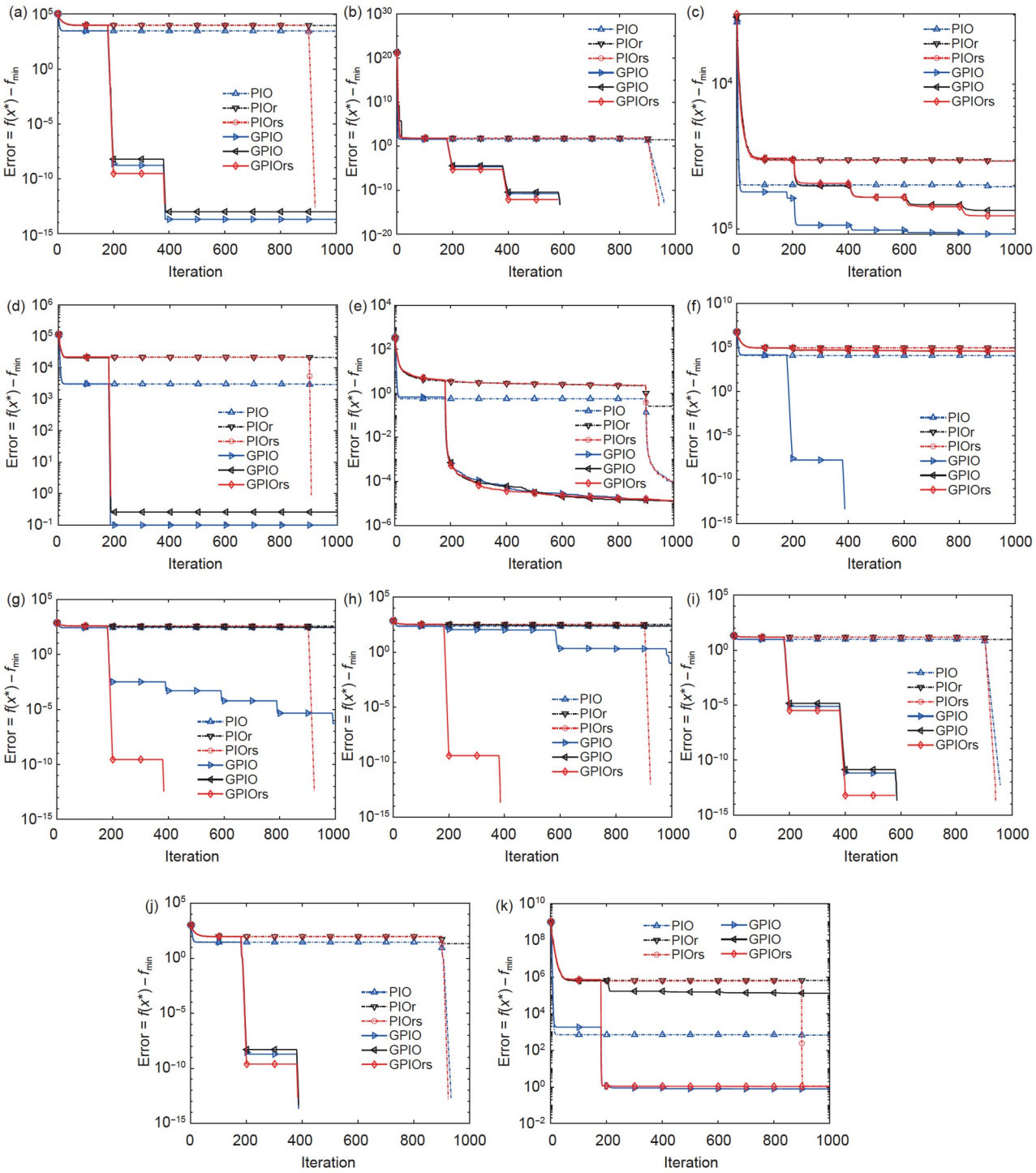

图 4 算法求解时, 算法误差逐步减小过程. (a) Sphere f_1 函数; (b) Schwefel's P2.22 f_2函数; (c) Schwefel's P1.2 f_3函数; (d) Step f_4 函数; (e) Quadric noise f_5函数; (f) Rosenbrock f_6函数; (g) Rastrigin f_7函数; (h) 非连续Rastrigin f_8函数; (i) Ackley f_9函数; (j) Griewank f_{10}函数; (k) Generalized penalized f_{11}函数.

Figure 4 Error results of PIO variants in solving eleven single objective optimization problems. (a) Sphere f_1; (b) Schwefel's P2.22 f_2; (c) Schwefel's P1.2 f_3; (d) Step f_4; (e) Quadric noise f_5; (f) Rosenbrock f_6; (g) Rastrigin f_7; (h) Noncontinuous Rastrigin f_8; (i) Ackley f_9; (j) Griewank f_{10}; (k) Generalized penalized f_{11}.

表 3　多峰单目标测试函数优化结果

Table 3　Result comparisons on six multimodal single objective optimization problems

Algo.	Best	Best	std. dev.	Best	Mean	std. dev.	Best	Mean	std. dev.
		f_6			f_7			f_8	
PIO	−330	11370.900	6787.271	120.0029	390.7320	46.9230	330.1820	555.3179	50.8865
PIOr	4531.2676	90826.058	28165.454	227.2797	502.3923	45.2793	418.9130	666.2776	42.9942
PIOrs	−330	85447.988	34071.193	120	**120**	0	330	**330**	0
GPIO	−330	**−330**	3.11E-14	120.0000	120.0000	4.06E-07	330.0000	330.1076	0.55530
GPIOr	−330	37637.890	15072.542	120.0004	417.1970	63.7461	331.7031	579.9657	41.0225
GPIOrs	2873.698	38306.134	13855.809	120	**120**	0	330	**330**	0
		f_9			f_{10}			f_{11}	
PIO	−330	−330	3.59E−14	−450	−450	6.17E−14	181.0571	847.8306	1595.752
PIOr	−329.7144	−320.7820	1.3832	−449.1828	−428.3960	5.4950	182.6996	643507.58	598373.08
PIOrs	−330	**−330**	0	−450	**−450**	0	180.8938	181.0990	0.08449
GPIO	−330	−330	4.01E−14	−450	−450	5.33E−14	180.3791	**180.7795**	0.19001
GPIOr	−330	−330	4.01E−14	−450	−450	5.02E−14	180.4114	129872.910	123638.67
GPIOrs	−330	**−330**	0	−450	**−450**	0	180.8485	181.0556	0.08037

表 4　多模态优化测试函数

Table 4　Benchmark functions of multimodal optimization

Function	Function name	Optima (global/local)	Niche radius r	Maximum	Number of known optima (NKP)
f_1 (1D)	Five-uneven-peak trap	2/3	0.01	200.0	2
f_2 (1D)	Equal maxima	5/0	0.01	1.0	5
f_3 (1D)	Uneven decreasing maxima	1/4	0.01	1.0	1
f_4 (2D)	Himmelblau	4/0	0.01	200.0	4
f_5 (2D)	Six-hump camel back	2/4	0.5	4.126513	2
f_6 (2D)	Shubert	$D\cdot 3^D$/many	0.5	186.73090	18
f_6 (3D)			0.5	2709.09350	81
f_7 (2D)	Vincent	6^D/0	0.2	1.0	36
f_7 (3D)			0.2	1.0	216
f_8 (2D)	Modified Rastrigin - All global optima	$\prod_{i=1}^{D} k_i$/0	0.01	−2.0	12
f_8 (8D)			0.01	−8.0	12

化算法中迭代次数$N_{c1}^{\max}=450$, $N_{c1}^{\max}=500$; 广义鸽群算法中, $N_{c1}=90$, $N_{c2}=100$, $n=5$.

表5给出了算法在8种多模态最大化测试函数上的极值解比率, 其中NKP为测试函数已知的极值解的数目, NR=50为算法运行次数; NPF为算法在NR次运行后, 找到的极值解的总数, PR为找到的极值解与所有极值解的比率. 对于同一测试问题, 算法得到的越大, PR越大, 表明算法求解多模态问题的性能越好. 表6展示了算法的求解精度结果. 在表中, 结果加粗部分为实验中6种算法的最优结果. 表6为表5结果的补充, 目的是展示当前最好解(一个解)与最优解的距离, 并不是比较不同算法搜索单个全局最优值的能力. 相比单目

表 5 多模态最大化函数的优化解比率

Table 5 The results of the optima ratio (PR) on eight maximum problems (ϵ=10^{-3})

Function	NKP×NR	PIO		PIOr		PIOrs		GPIO		GPIOr		GPIOrs	
		NPF	PR	NPF	PR	NPF	PR	NPF	PR	NPF	PR	NPF	PR
f_1 (1D)	100	73	0.73	99	0.99	**100**	1.0	66	0.66	**100**	1.0	**100**	1.0
f_2 (1D)	250	54	0.216	248	0.992	**249**	0.996	53	0.212	244	0.976	245	0.98
f_3 (1D)	50	**50**	1.0	**50**	1.0	**50**	1.0	**50**	1.0	**50**	1.0	**50**	1.0
f_4 (2D)	200	50	0.25	55	0.275	58	0.29	50	0.25	105	0.525	**122**	0.61
f_5 (2D)	100	50	0.5	83	0.83	83	0.83	50	0.5	94	0.94	**95**	0.95
f_6 (2D)	900	**100**	0.1111	10	0.0111	6	0.0067	**100**	0.1111	46	0.0511	59	0.0656
f_6 (3D)	4050	76	0.0188	0	0	0	0	**80**	0.0198	0	0	0	0
f_7 (2D)	1800	50	0.0278	273	0.1517	269	0.1494	50	0.0278	**382**	0.2122	377	0.2094
f_7 (3D)	10800	49	0.0045	47	0.0044	52	0.0048	50	0.0046	208	0.0193	**226**	0.0209
f_8 (2D)	600	50	0.0833	119	0.1983	101	0.1683	50	0.0833	**253**	0.4217	252	0.42
f_8 (8D)	600	0	0	0	0	0	0	**2**	0.0033	0	0	0	0

表 6 多模态最大化测试函数的优化精度

Table 6 Search accuracy comparisons on eight multimodal optimization problems

Algo.	Best	Best	std. dev.	Best	Mean	std. dev.	Best	Mean	std. dev.
	f_1 (1D)			f_2 (1D)			f_3 (1D)		
PIO	200	**199.1075**	3.3121	1	**1**	2.82E−16	0.99999	0.99999	2.76E−06
PIOr	200	197.7171	6.0551	1	0.99999	3.70E−12	0.99999	0.99999	1.87E−06
PIOrs	200	198.5327	5.2644	1	0.99999	7.47E−12	0.99999	0.99999	3.17E−07
GPIO	200	197.9158	5.1987	1	**1**	2.10E−16	0.99999	0.99999	3.98E−05
GPIOr	200	198.7412	3.7440	1	0.99999	2.94E−15	0.99999	0.99999	1.74E−10
GPIOrs	200	198.2589	6.3267	1	0.99999	9.59E−14	0.99999	**0.99999**	4.94E−12
	f_4 (2D)			f_5 (2D)			f_6 (2D)		
PIO	200	**199.9999**	5.55E−12	4.1265	**4.12651**	2.30E−15	186.73090	186.73090	2.57E−07
PIOr	199.9999	199.9989	0.00165	4.1265	4.12638	0.00029	186.73088	186.40628	1.08541
PIOrs	199.9999	199.9984	0.00310	4.1265	4.12635	0.00046	186.73081	186.59201	0.23135
GPIO	200	199.9999	1.24E-10	4.1265	**4.12651**	2.09E−15	186.73090	**186.73090**	1.12E−08
GPIOr	199.9999	199.9999	0.00016	4.1265	4.12649	9.85E−05	186.73090	186.71634	0.02847
GPIOrs	199.9999	199.9999	0.00020	4.1265	4.12649	9.00E−05	186.73090	186.72486	0.01077
	f_6 (3D)			f_7 (2D)			f_7 (3D)		
PIO	2709.0935	2657.5709	179.599	1	0.99999	4.45E−05	1	0.99996	0.00017
PIOr	2708.1524	2450.2618	275.754	0.99999	0.99998	2.68E−05	0.99998	0.99892	0.00123
PIOrs	2706.9301	2491.9856	248.4029	0.99999	0.99998	6.88E−05	0.99999	0.99865	0.00169
GPIO	2709.0935	**2683.5454**	133.2934	1	**0.99999**	3.52E−08	1	**0.99999**	3.80E−06
GPIOr	2709.0618	2667.2709	57.7885	0.99999	0.99999	6.11E−07	0.99999	0.99999	4.00E−06
GPIOrs	2709.0232	2653.5038	85.0034	0.99999	0.99999	1.86E−07	0.99999	0.99999	1.52E−06
	f_8 (2D)			f_8 (8D)					
PIO	−2	−2.00000	2.43E−06	−8.00369	−8.30233	0.29963			
PIOr	−2.0000	−2.00025	0.00040	−8.88167	−10.34639	0.83598			
PIOrs	−2.0000	−2.00025	0.00043	−8.46401	−10.41010	0.99101			
GPIO	−2	**−2**	6.72E−15	−8.00012	**−8.19927**	0.35208			
GPIOr	−2.0000	−2.00000	2.61E−05	−8.08811	−8.77948	0.34530			
GPIOrs	−2.0000	−2.00000	1.48E−05	−8.06431	−8.79702	0.54160			

标测试函数, 多模态测试函数的维度较小, 大部分算法在多次运行中均可以找到较好的一个解.

表5结果显示, 在大部分多模态函数上(除函数f_6外), 对于PIO算法和GPIO算法, 具有环形结构的算法均优于对应的具有全局结构的算法, 说明具有环形结构的算法能够找到更多的极值解, 具有更好的多样性. 然而, 表6结果显示, GPIO在大部分函数上, 获得了最好的搜索精度, 即搜索结果平均最优值最大.

这表明算法得到的解集合多样性与算法的求解精度之间的矛盾, GPIO与PIO算法在函数f_6上优化效果优于其他算法, 也是由于其他算法在此问题上利用能力不足, 导致结果精度不够. 因此, 对于不同的多模态问题, 协调算法多样性与搜索精度, 即平衡探索与利用的关系, 也是算法设计需要考虑的重要问题.

5 结论

本文研究了一种平衡探索能力和利用能力的广义鸽群优化算法, 改进了算法中解结构和算法参数的设置, 并将改进算法应用于求解单目标优化问题和多模态优化问题. 实验结果表明, 改进算法提高了鸽群优化算法的求解效率, 改进了算法的求解精度.

当前的广义鸽群优化算法中, 将算法分为多个阶段; 对于不同算子, 每个阶段使用了相同的迭代次数. 算法对于问题和求解过程中的解集合的信息利用较少. 因此, 未来研究工作将主要在: (1) 研究具有自适应机制的广义鸽群优化算法, 针对不同求解阶段, 自适应选择求解算子; (2) 研究具有学习能力的广义鸽群优化算法, 提高算法的求解能力, 并将学习算子扩展到不同类型的群体智能优化算法中.

参考文献

1 Duan H, Qiao P. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning. Int J Intell Comput Cybern, 2014, 7: 24–37

2 Duan H B, Qiu H X, Fan Y M. Unmanned aerial vehicle close formation cooperative control based on predatory escaping pigeon-inspired optimization (in Chinese). Sci Sin Tech, 2015, 45: 559–572 [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: 559–572]

3 Ishii S, Yoshida W, Yoshimoto J. Control of exploitation-exploration meta-parameter in reinforcement learning. Neural Networ, 2002, 15: 665–687

4 Alba E, Dorronsoro B. The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. IEEE Trans Evol Computat, 2005, 9: 126–142

5 Lin L, Gen M. Auto-tuning strategy for evolutionary algorithms: Balancing between exploration and exploitation. Soft Comput, 2009, 13: 157–168

6 Črepinšek M, Liu S H, Mernik M. Exploration and exploitation in evolutionary algorithms. ACM Comput Surv, 2013, 45: 1–33

7 Hussain K, Salleh M N M, Cheng S, et al. On the exploration and exploitation in popular swarm-based metaheuristic algorithms. Neural Comput Appl, 2019, 31: 7665–7683

8 Li J, Tan Y. A comprehensive review of the fireworks algorithm. ACM Comput Surv, 2020, 52: 1–28

9 Witkowski O, Ikegami T. How to make swarms open-ended? Evolving collective intelligence through a constricted exploration of adjacent possibles. Artif Life, 2019, 25: 178–197

10 Warnat-Herresthal S, Schultze H, Shastry K L, et al. Swarm learning for decentralized and confidential clinical machine learning. Nature, 2021, 594: 265–270

11 Dorigo M, Theraulaz G, Trianni V. Swarm robotics: Past, present, and future. Proc IEEE, 2021, 109: 1152–1165

12 Sun J, Wang J, Chen J, et al. Cooperative communication based on swarm intelligence: Vision, model, and key technology (in Chinese). Sci Sin Inf, 2020, 50: 307–317 [孙佳琛, 王金龙, 陈瑾, 等. 群体智能协同通信: 愿景、模型和关键技术. 中国科学: 信息科学, 2020, 50: 307–317]

13 Cheng S, Qin Q, Chen J, et al. Brain storm optimization algorithm: A review. Artif Intell Rev, 2016, 46: 445–458

14 Ma L, Cheng S, Shi Y. Enhancing learning efficiency of brain storm optimization via orthogonal learning design. IEEE Trans Syst Man Cybern Syst, 2021, 51: 6723–6742

15 Duan H, Qiu H. Advancements in pigeon-inspired optimization and its variants. Sci China Inf Sci, 2019, 62: 70201

16 Ma L, Wang Y, He Y M, et al. A multi-objective optimization method for intelligent swarm robotic control model with changeable parameters (in Chinese). Sci Sin Tech, 2020, 50: 526–537 [王原, 马力, 王凌, 等. 智能机器人可变参数群体控制模型的多目标优化方法. 中国科学: 技术科学, 2020, 50: 526–537]
17 Lu H, Yang S, Zhao M, et al. Multi-robot indoor environment map building based on multi-stage optimization method. Complex Syst Model Simul, 2021, 1: 145–161
18 Duan H, Ye F. Progresses in pigeon-inspired optimization algorithms (in Chinese). J Beijing Univ Technol, 2017, 43: 1–7 [段海滨, 叶飞. 鸽群优化算法研究进展. 北京工业大学学报, 2017, 43: 1–7]
19 Cheng S, Lei X, Lu H, et al. Generalized pigeon-inspired optimization algorithms. Sci China Inf Sci, 2019, 62: 70211
20 Yang Z, Liu K, Fan J, et al. A novel binary/real-valued pigeon-inspired optimization for economic/environment unit commitment with renewables and plug-in vehicles. Sci China Inf Sci, 2019, 62: 70213
21 Xiang S, Xing L, Wang L, et al. Comprehensive learning pigeon-inspired optimization with Tabu list. Sci China Inf Sci, 2019, 62: 70208
22 Duan H, Huo M, Yang Z, et al. Predator-prey pigeon-inspired optimization for UAV ALS longitudinal parameters tuning. IEEE Trans Aerosp Electron Syst, 2019, 55: 2347–2358
23 Duan H, Huo M, Shi Y. Limit-cycle-based mutant multiobjective pigeon-inspired optimization. IEEE Trans Evol Computat, 2020, 24: 948–959
24 Cui Z, Zhang J, Wang Y, et al. A pigeon-inspired optimization algorithm for many-objective optimization problems. Sci China Inf Sci, 2019, 62: 70212
25 Hu Y, Wang J, Liang J, et al. A self-organizing multimodal multi-objective pigeon-inspired optimization algorithm. Sci China Inf Sci, 2019, 62: 70206
26 Lei X, Ding Y, Wu F X. Detecting protein complexes from DPINs by density based clustering with Pigeon-Inspired Optimization Algorithm. Sci China Inf Sci, 2016, 59: 070103
27 Yang Z Y, Duan H B, Fan Y M. Unmanned aerial vehicle formation controller design via the behavior mechanism in wild geese based on Levy flight pigeon-inspired optimization (in Chinese). Sci Sin Tech, 2018, 48: 161–169 [杨之元, 段海滨, 范彦铭. 基于莱维飞行鸽群优化的仿雁群无人机编队控制器设计. 中国科学: 技术科学, 2018, 48: 161–169]
28 Duan H, Zhao J, Deng Y, et al. Dynamic discrete pigeon-inspired optimization for multi-UAV cooperative search-attack mission planning. IEEE Trans Aerosp Electron Syst, 2021, 57: 706–720
29 Shan X, Wang Y, Ji Z C. Energy efficiency optimization for discrete workshop based on parametric knowledge pigeon swarm algorithm (in Chinese). J Syst Simul, 2017, 29: 2140–2148 [单鑫, 王艳, 纪志成. 基于参数知识鸽群算法的离散车间能效优化. 系统仿真学报, 2017, 29: 2140–2148]
30 Liao Z, Gong W, Wang L. A hybrid swarm intelligence with improved ring topology for nonlinear equations (in Chinese). Sci Sin Inf, 2020, 50: 396–407 [廖作文, 龚文引, 王凌. 基于改进环拓扑混合群体智能算法的非线性方程组多根联解. 中国科学: 信息科学, 2020, 50: 396–407]
31 Li X, Engelbrecht A, Epitropakis M G. Benchmark functions for CEC'2013 special session and competition on niching methods for multimodal function optimization. Report. Melbourne: Evolutionary Computation and Machine Learning Group, RMIT University, 2013

Generalized pigeon-inspired optimization algorithm for balancing exploration and exploitation

CHENG Shi[1], ZHANG MingMing[1], SHI YuHui[2], LU Hui[3], LEI XiuJuan[1] & WANG Rui[4]

[1] *School of Computer Science, Shaanxi Normal University, Xi'an 710119, China;*
[2] *Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation, Southern University of Science and Technology, Shenzhen 518055, China;*
[3] *School of Electronics and Information Engineering, Beihang University, Beijing 100191, China;*
[4] *College of Systems Engineering, National University of Defense Technology, Changsha 410073, China*

A generalized pigeon-inspired optimization (GPIO) algorithm for balancing the exploration and exploitation abilities is proposed herein. The traditional pigeon-inspired optimization algorithm includes two operators, namely the map and compass operator and the landmark operator. These two operators are implemented only for one round at a single run. In the GPIO algorithm, the search process is divided into multiple stages, and two operators are implemented in each stage. These two operators are implemented for multiple rounds at one single run. The map and compass operator focuses on the exploration ability, while the landmark operator focuses on the exploitation ability. The GPIO algorithm changes the execution order of the two operators without additional objective function evaluation. Moreover, the structure of the solutions and the parameter settings are extended in the GPIO algorithm, which is beneficial to search quality improvement. The simulation results show that the GPIO algorithm improves the search efficiency and the search results of the algorithm.

swarm intelligence, pigeon-inspired algorithm, exploration and exploitation, multimodal optimization

doi: 10.1360/SST-2021-0371