基于多元信息引导的人工蜂群算法

周新宇[1],刘 颖[1,2],吴艳林[1],郭京蕾[3]
(1. 江西师范大学计算机信息工程学院,江西南昌 330022;2. 长沙理工大学计算机与通信工程学院,湖南长沙 410114;
3. 华中师范大学计算机学院,湖北武汉 430079)

摘 要: 利用优秀个体增强解搜索方程的开采能力是改进人工蜂群算法的一种主流思路. 然而,现有相关工作往往仅以适应度信息作为评价个体的唯一标准,易导致算法出现早熟收敛等问题. 本文提出一种多元信息引导的人工蜂群算法,分别设计了基于适应度、位置以及相似度信息的3种解搜索方程,并在雇佣蜂阶段和观察蜂阶段采用了不同的使用方式. 同时,为保存侦察蜂阶段的搜索经验,采用一种微调后的邻域搜索机制用于处理被放弃蜜源. 在CEC2013测试集和一个实际优化问题上进行了大量实验验证,与6种衍生算法和5种知名的相关改进人工蜂群算法进行了对比,结果表明本文算法性能竞争优势明显,在结果精度和收敛速度上均有更好表现.

关键词: 人工蜂群算法;优秀个体;多元信息;解搜索方程;邻域搜索

基金项目: 国家自然科学基金项目(No.61966019);江西省自然科学基金项目(No.20192BAB207030);中央高校基本科研业务费资助项目(No.CCNU20TS026)

中图分类号: TP18 **文献标识码:** A **文章编号:** 0372-2112(2024)04-1349-15

电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20220146

Artificial Bee Colony Algorithm Based on Multiple Information Guidance

ZHOU Xin-yu[1], LIU Ying[1,2], WU Yan-lin[1], GUO Jing-lei[3]
(1. *School of Computer and Information Engineering, Jiangxi Normal University, Nanchang, Jiangxi* 330022, *China*;
2. *School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, Hunan* 410114, *China*;
3. *School of Computer Science, Central China Normal University, Wuhan, Hubei* 430079, *China*)

Abstract: As one of the main ideas to improve the artificial bee colony (ABC) algorithm, the superior individuals are used to enhance the exploitative capability of the solution search equation. However, in the related works, the fitness information is often considered as the sole criterion for evaluating the individuals, which may easily cause some problems, e. g., the premature convergence. In this work, an improved ABC variant is proposed based on multiple information guidance, called ABC-MIG. In ABC-MIG, three different solution search equations are designed by using the fitness, position, and similarity information, respectively, and these new solution search equations are used in different ways for the employed bee phase and onlooker bee phase. Meanwhile, to save the search experience for the scout bee phase, a modified neighborhood search strategy is used to handle the abandoned food sources. To verify the effectiveness of ABC-MIG, extensive experiments are carried out on the CEC2013 test suite and one real-world optimization problem, and six derivative algorithms and five well-known improved ABC variants are included in the performance comparison. The results confirm that ABC-MIG has very competitive performance, in terms of the result accuracy and convergence speed.

Key words: artificial bee colony algorithm; superior individuals; multiple information; solution search equation; neighborhood search

Foundation Item(s): National Natural Science Foundation of China (No.61966019); Natural Science Foundation of Jiangxi Province (No.20192BAB207030); Fundamental Research Funds for the Central Universities (No.CCNU20TS026)

收稿日期:2022-01-30;修回日期:2022-09-19;责任编辑:王天慧

1 引言

群智能优化算法是一类高效的全局优化方法[1],与一些经典的最优化方法相比,其对问题的数学性质几乎没有要求,例如:不要求问题有连续可导等性质.随着对群智能优化算法研究的不断深入,涌现出了很多不同类型的算法,它们通过模拟自然界中社会性生物群体的相互协作和信息交互行为来实现寻优[2],例如:蚁群算法模拟了蚂蚁的觅食行为、粒子群优化算法模拟了鸟群的飞翔行为.

在这些算法中,人工蜂群算法(Artificial Bee Colony algorithm, ABC)是较为新颖的一种[3],它模拟了蜂群的智能采蜜行为,将搜索优化问题的全局最优解类比于蜂群寻找花蜜量最大的蜜源.已有研究工作指出[4~6],与其他代表性的群智能优化算法相比,ABC在很多经典的测试函数上表现出了更好或相似性能.因其结构简单、控制参数少以及性能优良等特点,ABC在很多实际优化问题中也得到了应用,例如:特征选择问题[7]、情感分类问题[8].

虽然ABC有较好性能,但求解一些复杂优化问题时还存在收敛速度慢、求解精度低的不足.主要原因在于其解搜索方程有较强随机性,导致算法勘探能力强,而开采能力弱.为此,研究人员提出了很多不同的解决方法,其中一种主流思路是利用种群的最优个体或精英个体来改进解搜索方程[4, 5, 9].例如,Zhu等人[4]提出了一种基于种群最优个体引导的ABC(Gbest-guided ABC, GABC),在解搜索方程中引入了最优个体这一项,期望挖掘最优个体的有益信息增强开采能力;Kong等人[5]提出了精英组引导的ABC(ABC based on Elite group guidance and Combined breadth-depth search strategy, ECABC),把最好的一些个体视为精英个体,再以精英组的中心点作为解搜索方程的搜索起点,以发挥精英个体的引导作用来提高开采能力.对比经典ABC,这些相关工作能在一定程度上更好地平衡勘探和开采能力,从而增强ABC求解复杂优化问题的性能.

然而,需要指出的是,上述思路还存在一定不足.从个体信息的角度来看,最优个体或精英个体是针对适应度而言,若仅以适应度信息来引导算法搜索,则可能会导致算法变得过于贪婪,引起早熟等问题,增加了算法陷入局部最优的风险.事实上,个体自身所包含的信息并不局限于适应度信息,还包括可表征个体间离散程度的位置信息,以及个体间相似程度的相似度信息等.因此,如何有效地利用好这些多元信息,提高算法搜索过程中信息引导的多样性,是设计高效ABC的关键.为此,本文提出了一种多元信息引导的ABC,记作ABC-MIG(ABC based on Multiple Information Guidance),其主要特点如下:

(1)不同于现有ABC仅以适应度信息来引导算法搜索,ABC-MIG同时采用了适应度、位置以及相似度信息进行协同引导,构建了对应于3种信息的3种解搜索方程来生成后代个体,避免过度依赖于适应度信息而导致算法早熟.

(2)为合理使用构建的3种解搜索方程,在ABC-MIG的雇佣蜂阶段和观察蜂阶段分别采用了随机选择和贪婪选择的使用方式,以更好地平衡算法的勘探和开采能力.

(3)为保存好侦察蜂阶段的搜索经验,ABC-MIG采用了一种微调后的邻域搜索机制来代替随机初始化方法,以进一步提高侦察蜂阶段搜索的有效性.

为验证算法性能,在CEC2013测试集上开展实验,与6种衍生算法和5种知名的相关改进ABC进行对比.结果表明ABC-MIG性能有很强竞争力,特别是在复杂的多峰函数类型上有更好性能.此外,ABC-MIG还被应用于求解一个实际优化问题,即扩频雷达的多相位编码设计问题,在该问题上ABC-MIG的性能同样非常优秀.

2 相关工作

2.1 经典ABC

ABC以随机初始化种群来开启搜索过程.设第i个蜜源为$X_i=\left(x_{i,1},x_{i,2},\cdots,x_{i,D}\right)$,$i\in\{1,2,\ \cdots,\mathrm{SN}\}$,SN为群体规模,$D$表示优化问题维度,则$X_i$可由下式生成:

$$x_{i,j}=x_j^{\min}+\mathrm{rand}(0,1)\cdot\left(x_j^{\max}-x_j^{\min}\right) \tag{1}$$

其中,$x_{i,j}\in[x_j^{\min},x_j^{\max}]$,$x_j^{\min}$和$x_j^{\max}$分别代表优化问题的第$j$维边界,$\mathrm{rand}(0,1)$为$[0,1]$范围内的均匀随机数.在随机初始化种群之后,ABC的搜索过程可分为雇佣蜂阶段、观察蜂阶段以及侦察蜂阶段,这3个阶段会迭代运行,直至算法停机.

(1)雇佣蜂阶段

雇佣蜂会在整个搜索空间内勘探新蜜源,通过下式所示的解搜索方程更新蜜源位置:

$$v_{i,j}=x_{i,j}+\varnothing_{i,j}\cdot\left(x_{i,j}-x_{k,j}\right) \tag{2}$$

其中,V_i为新蜜源(即子代个体),X_i为原蜜源(即父代个体),$\varnothing_{i,j}\in[-1,1]$的均匀随机数,$X_k$是随机选出的蜜源,且$X_k\neq X_i$.注意,$j$为任意选中的一个维度,$X_i$和$V_i$仅在该维度上不同.若$V_i$的花蜜量要多于$X_i$,则$V_i$将取代$X_i$进入下一次迭代;否则,$X_i$保持不变.

(2)观察蜂阶段

观察蜂会按蜜源的花蜜量选择优秀蜜源进一步开采,以式(2)所示的解搜索方程来搜索新蜜源.蜜源的花蜜量即为个体的适应度值,按下式计算:

$$\mathrm{fit}_i=\begin{cases}\dfrac{1}{1+f(X_i)}, & f(X_i)\geqslant 0\\ 1+\mathrm{abs}(f(X_i)), & f(X_i)<0\end{cases} \tag{3}$$

其中,fit_i为X_i的适应度值,$f(\cdot)$为目标函数值,abs(·)为取绝对值. 蜜源的花蜜量越多,其被观察蜂选中的概率越高,按下式计算:

$$p_i=\frac{\mathrm{fit}_i}{\sum_{j=1}^{\mathrm{SN}}\mathrm{fit}_i} \tag{4}$$

其中,p_i为X_i的选择概率. 在得到所有蜜源的选择概率后,观察蜂会以轮盘赌机制进行选择.

(3)侦察蜂阶段

该阶段作用是防止种群陷入搜索停滞,避免开采殆尽蜜源占用过多计算资源. 在雇佣蜂阶段和观察蜂阶段,先为每个蜜源设置一个计数器来记录蜜源是否成功更新. 若成功更新,计数器重置为0;若未成功,则计数器加1. 当计数器值超过阈值limit,则认为该蜜源已开采殆尽,与之相关的雇佣蜂会转变为侦察蜂,再通过式(1)重新生成一个新蜜源.

2.2　相关改进ABC

近年来,很多改进ABC相继被提出,大致可分为以下三类:

(1)如何改进解搜索方程

经典ABC的解搜索方程中,因随机个体的影响,使得搜索方向有很强的随机性,导致算法存在勘探能力强,但开采弱的不足. 为此,很多相关工作提出利用优秀个体来改进解搜索方程,增强开采能力. Zhu等人[4]提出了一种基于种群最优个体引导的ABC(Gbest-guided ABC,GABC),在解搜索方程中增加了种群最优个体Gbest项,利用Gbest的有益信息来调整搜索方向. 还有一些其他工作利用精英个体,比如:Kong等人[5]提出了精英组引导的ABC(ABC based on Elite group guidance and Combined breadth-depth search strategy,ECABC),以精英组的中心点作为解搜索方程的搜索起点,这可在一定程度上避免早熟问题.

(2)如何结合多种解搜索方程

不同解搜索方程的搜索能力一般不同,因此如何结合多种解搜索方程是近年来的研究热点. Wang等人[10]提出了一种多策略集成的ABC(Multi-strategy Ensemble ABC,MEABC),选择3种不同的解搜索方程用于构建策略候选池,设计了随机选择的方式从候选池中选用解搜索方程. Kiran等人[11]提出了一种可变搜索策略的ABC(ABC with Variable Search Strategy,ABCVSS),选用5种不同的解搜索方程来构建策略候选池,并设计了一种自适应策略选择机制,为每种策略设置一个计数器来记录其成功更新的蜜源数量,以此计算每种策略被选用的概率.

(3)如何与其他搜索技术相结合

还有一部分工作集中在如何与其他搜索技术相结合,以实现不同搜索技术的优势互补. 基于邻域搜索技术,Zhou等人[12]提出了一种结合全局邻域搜索的改进ABC(Modified ABC with Neighborhood Search,MABC-NS),其主要特点是在Gbest的邻域范围内进行细粒度搜索,以增强算法的局部开采能力. 为避免Gbest或精英个体可能导致的早熟问题,Gao等人[6]提出了结合方向学习策略和精英学习策略的ABC(ABC with direction Learning and elite Learning,LLABC),方向学习策略用于调整搜索方向,辅助精英学习策略.

3　本文算法

3.1　主要动机

为提高ABC性能,很多改进算法的思路都集中在如何利用优秀个体来增强开采能力,试图在勘探和开采之间取得更好平衡. 例如,这方面的开创性工作GABC[4]在解搜索方程中利用了最优个体. 当然,还有一些利用精英个体的工作,例如:ECABC[5],其改进的解搜索方程如下:

$$v_{i,j}=\mathrm{xc}_j+\varnothing_{i,j}\cdot\left(\mathrm{Gbest}_j-x_{k,j}\right) \tag{5}$$

其中,$\mathrm{XC}=\left(\mathrm{xc}_1,\mathrm{xc}_2,\cdots,\mathrm{xc}_D\right)$是种群中最好的$T$个精英个体构成的精英组中心,$T=\mathrm{ceil}(p\cdot\mathrm{SN})$,参数$p$控制了精英个体的比例,取值为0.1,即种群中前10%个体被视作为精英个体. 因此,精英组中心XC可由下式计算得到:

$$\mathrm{XC}=\frac{1}{T}\sum_{i=1}^{T}\mathrm{XE}_i \tag{6}$$

其中,XE_i为第i个精英个体.

虽然这些利用优秀个体的相关工作有效改进了ABC性能,但需指出的是,优秀个体的选择方式在很大程度上决定了算法性能. 在这些相关工作中,它们定义的"优秀个体"是从适应度的角度来评价,即把适应度信息作为判断个体是否优秀的唯一标准. 然而,这种评价方式过于局限和单一,易使算法变得过于贪婪,出现早熟等问题,这也促使进一步思考是否还能从其他角度评价个体. 沿着这一思路,发现对于个体而言,除可直观反映个体是否优秀的适应度信息,个体在搜索空间中还有自身的位置信息,以及可评估个体间相似程度的相似度信息. 这3类信息(适应度信息、位置信息、相似度信息)的特点各不相同,如果用作评价个体的标准,则有一定的互补性. 因此,本文提出基于这3类信息来评价个体,分别选择出对应的3类"优秀个体"用于改进解搜索方程,避免仅用单一的适应度信息,这在一

定程度上能增加算法搜索过程中的多样性，并且继续保持“优秀个体”的引导作用.

3.2 多元信息引导机制

多元信息引导机制旨在同时利用适应度信息、位置信息以及相似度信息的协同引导作用. 具体而言，分别以这3类信息作为评价标准来选择“优秀个体”，再分别设计3种对应的改进解搜索方程来发挥“优秀个体”的引导作用.

(1)基于适应度信息的解搜索方程

适应度信息是评价个体优劣最直观的标准，也是目前最常用的标准. 为此，本文继续用适应度信息来选择优秀个体，并在解搜索方程中利用这些优秀个体. 需说明的是，本文主要动机在于如何利用多元信息来引导算法搜索，而非设计新的解搜索方程，所以直接采用ECABC的改进解搜索方程作为适应度信息的利用方式[5]：

$$v_{i,j}=\mathrm{xc}_j^f+\varnothing_{i,j}\cdot\left(\mathrm{Gbest}_j-x_{k,j}\right) \tag{7}$$

其中，$\mathrm{XC}^f=\left(\mathrm{xc}_1^f,\mathrm{xc}_2^f,\cdots,\mathrm{xc}_D^f\right)$是以适应度信息作为评价标准选出的优秀个体构成的中心. 该中心的计算方式如下：

$$\mathrm{XC}^f=\frac{1}{T}\sum_{i=1}^{T}X_i^f \tag{8}$$

其中，X_i^f是第i个优秀个体，T是优秀个体的数量. 与ECABC相同，本文中优秀个体的数量设置为种群规模的10%，即$T=\lceil 0.1\cdot \mathrm{SN}\rceil$.

(2)基于位置信息的解搜索方程

位置信息是个体在搜索空间中的具体表现，能反映出个体间的远近关系和聚集程度. 但与适应度信息不同，位置信息无法直接用于评价个体的优劣程度. 若从问题的适应度地形来看，种群中的不同个体一般分布在具有不同特征的搜索区域，而位置相近的一些个体，其所在的搜索区域会有较大可能性具备相同或类似的特征. 通常，对于适应度较好的个体而言，其所处搜索区域很可能包含了全局最优解，值得算法进一步搜索. 因此，与该个体位置相近的其他个体所在的搜索区域同样也值得搜索. 按这一思路，可把种群中的最优个体作为参照个体，若种群中的其他个体离参照个体越近，则认为该个体的位置信息越优秀. 类似于式(7)，基于位置信息的解搜索方程如下所示：

$$v_{i,j}=\mathrm{xc}_j^p+\varnothing_{i,j}\cdot\left(\mathrm{Gbest}_j-x_{k,j}\right) \tag{9}$$

其中，$\mathrm{XC}^p=\left(\mathrm{xc}_1^p,\mathrm{xc}_2^p,\cdots,\mathrm{xc}_D^p\right)$是以位置信息作为评价标准而选出的优秀个体构成的中心. 该中心的计算方式如下：

$$\mathrm{XC}^p=\frac{1}{T}\sum_{i=1}^{T}X_i^p \tag{10}$$

其中，X_i^p是第i个优秀个体，T是优秀个体的数量，其值大小依旧设置为10%种群规模. 对于个体间距离的度量，采用最常用的欧式距离. 下式给出了如何计算优秀个体X_i^p与参照个体之间的欧式距离：

$$d=\sqrt{\sum_{j=1}^{D}\left(x_{i,j}^p-\mathrm{Gbest}_j\right)^2} \tag{11}$$

(3)基于相似度信息的解搜索方程

相似度信息表征了个体间的相似程度，包括个体在结构上和搜索方向上的相似度. 与位置信息类似，相似度信息也无法直接用于评价个体优劣，仅能通过与参照个体的比较来进行间接评价. 鉴于适应度较好的个体具备较优的个体结构和搜索方向，这值得种群中其他个体进行学习，从而有较大可能性促进算法朝着全局最优解的方向进行搜索. 为此，把最优个体作为参照个体，通过计算种群中的其他个体与参照个体的相似度来进行个体评价. 若某一个体与参照个体的相似度越高，则认为该个体的相似度信息越优秀. 同样地，类似于式(7)，基于相似度信息的解搜索方程如下：

$$v_{i,j}=\mathrm{xc}_j^s+\varnothing_{i,j}\cdot\left(\mathrm{Gbest}_j-x_{k,j}\right) \tag{12}$$

其中，$\mathrm{XC}^s=\left(\mathrm{xc}_1^s,\mathrm{xc}_2^s,\cdots,\mathrm{xc}_D^s\right)$是以相似度信息作为评价标准而选出的优秀个体构成的中心. 该中心的计算方式如下：

$$\mathrm{XC}^s=\frac{1}{T}\sum_{i=1}^{T}X_i^s \tag{13}$$

其中，X_i^s是第i个优秀个体，T是优秀个体的数量，仍然设置为种群规模的10%. 采用余弦相似度来度量个体相似度，即计算两个个体所构成夹角的余弦值$s\in[-1,1]$. 若某一个体对应的s值越趋近于1，则表明该个体越优秀. 下式给出了优秀个体X_i^s与参照个体间余弦相似度的计算方式：

$$s=\frac{\sum_{j=1}^{D}x_{i,j}^s\cdot\mathrm{Gbest}_j}{\sqrt{\sum_{j=1}^{D}(x_{i,j}^s)^2}\sqrt{\sum_{j=1}^{D}\mathrm{Gbest}_j^2}} \tag{14}$$

为合理使用这3种解搜索方程，本文结合ABC内在机制的特点，为雇佣蜂阶段和观察蜂阶段分别设计了不同的使用方式，目标是平衡好算法的勘探和开采能力，具体如下：

对于雇佣蜂阶段而言，其主要职责是在整个搜索空间中勘探新蜜源，应具备较好的全局搜索能力. 为此，为该阶段设计了一种随机选择的方式来使用3种解搜索方程，即为每只雇佣蜂随机选用一种解搜索

方程来生成新蜜源. 该方式可避免雇佣蜂对某一信息的过度依赖,均衡好3种信息的引导作用,从而保持雇佣蜂阶段应有的全局搜索能力. 这一过程可形式化如下:

$$V_i=\begin{cases}\mathrm{XC}^f+\varnothing_i\cdot\left(\mathrm{Gbest}-X_k\right)\\ \mathrm{XC}^p+\varnothing_i\cdot\left(\mathrm{Gbest}-X_k\right)\\ \mathrm{XC}^s+\varnothing_i\cdot\left(\mathrm{Gbest}-X_k\right)\end{cases}\tag{15}$$

在观察蜂阶段,观察蜂主要负责在较好蜜源附近进一步开采,应有较好的局部搜索能力. 因此,为观察蜂阶段设计了一种贪婪选择的方式,即把雇佣蜂阶段中表现最好的解搜索方程保留至观察蜂阶段继续使用. 为量化不同解搜索方程的优化效果,采用适应度改进量这一指标,如下所示:

$$\Delta_i=\begin{cases}\mathrm{fit}\left(X_i\right)-\mathrm{fit}\left(V_i\right), & \mathrm{fit}\left(V_i\right)<\mathrm{fit}\left(X_i\right)\\ 0, & \mathrm{fit}\left(V_i\right)\geqslant\mathrm{fit}\left(X_i\right)\end{cases}\tag{16}$$

其中,Δ_i表示雇佣蜂阶段中第i个蜜源所对应的适应度改进量. 通过统计雇佣蜂阶段中所有蜜源的适应度改进量,可分别计算出3种解搜索方程的具体优化效果,从而为观察蜂阶段选出表现最好的解搜索方程. 此外,我们考虑了2种极端情况:(1)当所有蜜源的适应度改进量都为0时,即3种解搜索方程均未产生优化效果,则采用经典的解搜索方程,以期通过其较强的勘探能力引导算法跳出可能陷入的局部最优;(2)当3种解搜索方程均有优化效果,且表现相同时,则为观察蜂随机选用一种解搜索方程.

3.3　改进的侦察蜂阶段

在经典ABC的侦察蜂阶段,对于被放弃蜜源,会采用随机初始化的方式生成一个新蜜源. 这一机制可防止被放弃蜜源诱导整个种群陷入搜索停滞. 然而,随机初始化方法可能会引发一项副作用,破坏当前的搜索经验. 具体而言,被放弃蜜源通常有较好质量,在某种程度上来说,它可能是离全局最优解较近的某一局部最优解,值得算法在搜索过程中继续利用. 因此,为提高侦察蜂阶段的有效性,一些相关改进方法相继被提出. 例如,Peng等人[13]采用邻域搜索机制在被放弃蜜源的邻域范围内进行细粒度搜索,以提高找到全局最优解的可能性,实验结果表明该方法有很好的效果.

受这些相关工作的启发,为避免随机初始化方法可能带来的副作用,直接在Peng等人采用的邻域搜索机制的基础上做了一些微调,把多元信息引导机制中构建的优秀个体中心进行了融合,主要目的是在邻域搜索过程中进一步发挥多元信息的引导作用. 下式给出了Peng等人采用的邻域搜索机制的具体表达式[13]:

$$\mathrm{TX}_i=r_1\cdot X_i+r_2\cdot\mathrm{Gbest}+r_3\cdot\left(X_j-X_k\right)\tag{17}$$

其中,TX_i是通过邻域搜索机制生成的新蜜源,X_i是被放弃蜜源. 系数r_1、r_2和r_3是[0,1]区间内的随机数,且满足条件:$r_1+r_2+r_3=1$. X_j和X_k是随机选出的两个互不相同的蜜源,且两者与X_i也不同. 把X_j替换为优秀个体中心$\mathrm{XC}\in\left\{\mathrm{XC}^f,\ \mathrm{XC}^p,\ \mathrm{XC}^s\right\}$,该中心的具体取值由观察蜂阶段所采用的解搜索方程决定. 例如,观察蜂阶段选用了基于适应度信息的解搜索方程,则$\mathrm{XC}=\mathrm{XC}^f$. 经微调后的邻域搜索机制的具体表达式如下:

$$\mathrm{TX}_i=r_1\cdot X_i+r_2\cdot\mathrm{Gbest}+r_3\cdot\left(\mathrm{XC}-X_k\right)\tag{18}$$

若观察蜂阶段选用了经典的解搜索方程,则随机选择一个优秀个体中心作为XC.

3.4　算法框架

与经典ABC相比,ABC-MIG有两点改进:(1)采用3种信息引导的解搜索方程,在雇佣蜂阶段和观察蜂阶段分别采用不同方式使用;(2)为提高侦察蜂阶段的有效性,采用多元信息引导的邻域搜索机制来保存搜索经验. 算法1给出了ABC-MIG的伪代码描述,SN为蜜源数量,FEs是已消耗的适应度函数的评估次数,MaxFEs是预设的适应度函数的最大评估次数,也是算法停机条件.

4　实验验证

为验证算法有效性,在广泛使用的CEC2013测试集上开展实验. 该测试集有28个函数[14],其中F01~F05为单峰函数、F06~F20为多峰函数、F21~F28为组合函数. 测试维度包括$D=30$和50,对应MaxFEs=10 000·D. ABC-MIG参数设置为:SN=60、limit=200. 为降低随机误差的影响,实验中所有算法均在每个测试函数上独立运行30次,以平均结果作为算法的最终结果. 此外,为确保实验结果对比具有统计意义,采用了两种非参数检验方法[15]:Wilcoxon秩和检验以及Friedman检验,其显著性水平均设为0.05. 两种检验的目的不同,前者用于函数层面,检验ABC-MIG与对比算法是否存在显著差异,用符号"+"、"="、"–"分别表示对比算法性能要优于、相当于、差于ABC-MIG;后者用于算法层面,以平均排名的方式给出算法的整体性能,排名值越小说明算法性能越好.

4.1　limit参数值分析

参数limit控制了侦察蜂阶段的执行频率,对算法性能有重要影响. 本节对该参数的取值情况进行分析,目的在于:(1)分析其对算法性能有何影响,(2)应取何值以确保算法有较好性能. 通常,不同的改进ABC的limit取值一般不同,但主要包括四种:50[13]、100[16]、

算法 1 ABC-MIG

输入:参数 SN、limit、MaxFEs

输出:全局最优个体 Gbest

1. 按式(1)生成初始种群,令 FEs=SN, $trial_i$=0.
2. while (FEs≤MaxFEs) do
3. for i=1 to SN do //雇佣蜂阶段
4. 按式(15)为 X_i 选择解搜索方程生成 V_i.
5. 若 $\mathrm{fit}(V_i) < \mathrm{fit}(X_i)$,令 $X_i=V_i$, $trial_i$=0;
6. 否则,令 $trial_i$++.
7. FEs++.
8. end for
9. for i=1 to SN do //观察蜂阶段
10. 按式(4)计算 X_i 被选中概率 p_i.
11. if rand(0, 1)<p_i do
12. 按贪婪方式为 X_i 选择解搜索方程生成 V_i.
13. 若 $\mathrm{fit}(V_i) < \mathrm{fit}(X_i)$,令 $X_i=V_i$, $trial_i$=0;
14. 否则,令 $trial_i$++.
15. FEs++.
16. end if
17. end for
18. for i=1 to SN do //侦察蜂阶段
19. if $trial_i$>limit do
20. 按式(18)为 X_i 生成 TX_i.
21. 令 $trial_i$=0, FEs++.
22. end if
23. end for
24. end while

200[6]以及 SN·D[5, 11]. 当 limit 取值较大时,侦察蜂阶段的执行频率较低,表明被放弃蜜源有较大可能会诱导整个种群陷入搜索停滞,从而降低算法搜索过程中的多样性. 相反地,limit 取值较小,则侦察蜂阶段会以较高的频率执行,虽然有利于避免种群出现搜索停滞的问题,但会破坏已获得的搜索经验,降低算法搜索的有效性.

下面分别对上述 4 种典型的 limit 取值进行实验,测试维度为 D=30. 表 1 给出了实验结果,最好结果以粗体突显. 可看出,在 5 个单峰函数上,50 和 SN·D 都未取得较好结果,而 100 和 200 能在其中 3 个函数上获得最好结果. 这说明对于单峰函数,limit 取值不宜过大,也不应该过小,适中的取值能得到最好性能. 在 15 个多峰函数上,200 是最佳取值,能在 11 个函数上得到最好结果. 此外,SN·D 在多峰函数上也有较好结果,说明较大 limit 能在多峰函数上取得更好结果. 原因是多峰函数的适应度地形比单峰函数崎岖,如果侦察蜂阶段能以较低频率执行,可保障算法在一些陡峭的区域内搜索得更充分,也更有利于发挥优秀个体中心的引导作用. 对于 8 个组合函数,100 和 200 的结果接近. 然而,组合函数的求解难度比单峰和多峰函数要高,这种情况下 limit 取值也应适中. 不难得出,limit=200 能确保 ABC-MIG 的总体性能最优,从表 1 最后一行的 Friedman 检验结果也可看出其排名第一.

表 1 ABC-MIG 采用不同 limit 取值的对比结果

函数	50	100	200	SN·D
F01	2.27×10^{-13}	**0.00**	**0.00**	**0.00**
F02	6.28×10^{6}	6.08×10^{6}	**6.03×10^{6}**	6.77×10^{6}
F03	5.76×10^{7}	**3.50×10^{7}**	4.16×10^{7}	4.54×10^{7}
F04	3.88×10^{4}	**3.82×10^{4}**	3.83×10^{4}	3.91×10^{4}
F05	8.38×10^{-13}	2.08×10^{-13}	**1.10×10^{-13}**	1.14×10^{-13}
F06	2.90×10	2.70×10	**2.46×10**	2.54×10
F07	6.31×10	5.32×10	5.29×10	**5.24×10**
F08	2.10×10	2.10×10	**2.09×10**	**2.09×10**
F09	2.56×10	**2.52**×10	2.54×10	2.53×10
F10	9.78×10^{-1}	1.11	**8.08×10^{-1}**	8.23×10^{-1}
F11	5.50×10^{-14}	**0.00**	**0.00**	**0.00**
F12	1.38×10^{2}	1.16×10^{2}	**8.64×10**	9.76×10
F13	2.00×10^{2}	1.66×10^{2}	**1.48×10^{2}**	1.48×10^{2}
F14	1.01×10	1.03×10	**2.80**	4.19
F15	**2.93×10^{3}**	2.97×10^{3}	3.17×10^{3}	3.31×10^{3}
F16	**9.76×10^{-1}**	1.02	9.98×10^{-1}	1.09
F17	3.06×10	3.05×10	**3.04×10**	**3.04×10**
F18	**3.00×10**	**3.00×10**	**3.00×10**	**3.00×10**
F19	6.31×10^{-1}	4.39×10^{-1}	**3.56×10^{-1}**	4.42×10^{-1}
F20	1.03×10	1.02×10	**9.91**	9.98
F21	3.27×10^{2}	**3.00×10^{2}**	3.20×10^{2}	3.03×10^{2}
F22	1.11×10^{2}	1.15×10^{2}	**1.04×10^{2}**	1.10×10^{2}
F23	4.42×10^{3}	4.30×10^{3}	4.24×10^{3}	**4.09×10^{3}**
F24	2.36×10^{2}	2.33×10^{2}	**2.32×10^{2}**	2.34×10^{2}
F25	**2.58×10^{2}**	2.71×10^{2}	2.80×10^{2}	2.78×10^{2}
F26	**2.00×10^{2}**	**2.00×10^{2}**	**2.00×10^{2}**	**2.00×10^{2}**
F27	8.16×10^{2}	7.05×10^{2}	**5.24×10^{2}**	6.02×10^{2}
F28	2.93×10^{2}	**2.80×10^{2}**	2.93×10^{2}	2.93×10^{2}
Fried.	3.34	2.48	**1.80**	2.38

4.2 算法有效性分析

为验证算法的有效性,本节对 ABC-MIG 进行消融实验,分别验证多元信息引导机制和邻域搜索机制. 为此,设计了 4 种对比算法:

ABC-1:基于适应度信息引导的 ABC;

ABC-2:基于位置信息引导的 ABC;

ABC-3:基于相似度信息引导的 ABC;

ABC-4:基于邻域搜索机制的 ABC.

前 3 种对比算法仅采用单一信息引导,即只保留一种解搜索方程,其他部分与 ABC-MIG 相同. ABC-4 保留多元信息引导机制,但在侦察蜂阶段采用了 Peng 等人[13]提出的邻域搜索机制. 测试维度为 D=30,表 2 给

出了 ABC-MIG 与这 4 种对比算法的实验结果,最好结果同样以粗体进行突显.

首先分析多元信息引导机制的有效性. 从表 2 可看出,前 3 种对比算法在 5 个单峰函数上的性能各有千秋,这说明不同的信息引导方式各有优点,并非某一信息能取得压倒性优势. 通过综合 3 种信息引导,ABC-MIG 在单峰函数上的结果要好于 3 种对比算法,这验证了多元信息引导要明显优于单一信息引导. 在 15 个多峰函数上,ABC-1 仅在 2 个函数上优于 ABC-MIG,而 ABC-2 和 ABC-3 在所有多峰函数上都未能更优. 并且,ABC-MIG 在 8 个多峰函数上取得了最好结果,这说明多元信息引导的方式在求解难度更大的多峰函数上同样优于单一信息引导. 在 8 个组合函数上的对比情况类于多峰函数,ABC-1 和 ABC-2 仅在 F22 优于 ABC-MIG,而 ABC-3 的结果均要差于或相当于 ABC-MIG. 表 2 最后两行的两种检验结果也验证了多元信息引导机制的有效性.

其次分析微调后的邻域搜索机制的有效性. 从表 2 可看出,ABC-4 仅在 F04 上的结果要优于 ABC-MIG,可能原因是 F04 的全局最优解位于一块非常狭长的区域上,对算法的搜索方向非常敏感,有利于勘探能力强的原机制取得更好结果. 相比之下,ABC-MIG 在 4 个函数上更优. 综合来看,微调后的邻域搜索机制有一定改进效果,且未降低原机制的有效性. 表 2 最后一行的 Friedman 检验结果也表明 ABC-MIG 的排名要好于 ABC-4.

表 2　算法有效性验证的实验结果

函数	ABC-1	ABC-2	ABC-3	ABC-4	ABC-MIG
F01	1.06×10^{-13}	2.12×10^{-13}	2.27×10^{-13}	**0.00**	**0.00**
F02	9.67×10^{6}	6.42×10^{6}	7.46×10^{6}	6.26×10^{6}	**6.03×10^{6}**
F03	1.93×10^{8}	6.00×10^{7}	7.11×10^{7}	6.57×10^{7}	**4.16×10^{7}**
F04	8.30×10^{4}	4.41×10^{4}	4.17×10^{4}	**3.41×10^{4}**	3.83×10^{4}
F05	1.78×10^{-13}	4.55×10^{-13}	6.33×10^{-13}	**1.10×10^{-13}**	**1.10×10^{-13}**
F06	2.46×10	2.30×10	2.83×10	**2.06×10**	2.46×10
F07	9.20×10	5.90×10	5.80×10	**5.17×10**	5.29×10
F08	**2.09×10**	**2.09×10**	**2.09×10**	2.10×10	**2.09×10**
F09	2.92×10	**2.40×10**	2.50×10	2.64×10	2.54×10
F10	**2.53×10^{-1}**	8.44×10^{-1}	9.10×10^{-1}	9.05×10^{-1}	8.08×10^{-1}
F11	1.89×10^{-15}	5.31×10^{-14}	5.50×10^{-14}	**0.00**	**0.00**
F12	1.14×10^{2}	1.32×10^{2}	1.12×10^{2}	8.98×10	**8.64×10**
F13	1.64×10^{2}	1.94×10^{2}	1.61×10^{2}	**1.39×10^{2}**	1.48×10^{2}
F14	**1.50**	3.28	4.27	5.20	2.80
F15	3.69×10^{3}	3.48×10^{3}	3.58×10^{3}	3.38×10^{3}	**3.17×10^{3}**
F16	1.36	1.28	1.25	1.11	**9.98×10^{-1}**
F17	3.05×10	3.05×10	3.05×10	3.05×10	**3.04×10**
F18	3.00×10	3.00×10	3.00×10	3.00×10	**3.00×10**
F19	4.74×10^{-1}	5.00×10^{-1}	5.05×10^{-1}	5.14×10^{-1}	**3.56×10^{-1}**
F20	1.16×10	1.04×10	1.03×10	**9.84**	9.91
F21	**2.99×10^{2}**	3.07×10^{2}	3.17×10^{2}	3.12×10^{2}	3.20×10^{2}
F22	**8.21×10**	9.21×10	9.69×10	1.23×10^{2}	1.04×10^{2}
F23	4.62×10^{3}	4.63×10^{3}	4.38×10^{3}	4.29×10^{3}	**4.24×10^{3}**
F24	2.76×10^{2}	2.38×10^{2}	2.37×10^{2}	**2.31×10^{2}**	2.32×10^{2}
F25	3.05×10^{2}	2.77×10^{2}	2.92×10^{2}	**2.63×10^{2}**	2.80×10^{2}
F26	**2.00×10^{2}**	**2.00×10^{2}**	**2.00×10^{2}**	**2.00×10^{2}**	**2.00×10^{2}**
F27	5.84×10^{2}	6.46×10^{2}	**4.93×10^{2}**	5.88×10^{2}	5.24×10^{2}
F28	3.00×10^{2}	3.36×10^{2}	3.43×10^{2}	3.00×10^{2}	**2.93×10^{2}**
+/=/−	3/7/18	1/12/15	0/13/15	1/23/4	—
Friedman	3.50	3.36	3.57	2.55	2.02

4.3　随机选择对比自适应选择

在多元信息引导机制中,雇佣蜂阶段采用了随机选择的方式来使用 3 种解搜索方程,这促使进一步思考,是否还存在其他更加高效的使用方式? 比如:自适

应选择方式. 目前,已有不少自适应ABC相继被提出[11, 17],主要思路是以算法的历史经验来选择策略,通常包括以下3个步骤:(1)采用多个解搜索方程来构建策略候选池;(2)在算法初始化阶段,赋予每种策略均等的选用概率;(3)在后续迭代过程中,每隔若干次迭代作为一个周期,比如:每隔20代,在这一周期内统计每种策略的成功率,以此作为下一周期内该策略的选用概率.

本节对随机选择和自适应选择进行对比. 在ABC-MIG基础上,设计了两种对比算法:(1)ABC-5:基于蜜源个数的自适应ABC;(2)ABC-6:基于适应度改进量的自适应ABC. 这两种算法均采用自适应选择的方式,以每隔20代作为一个周期. 不同的是,ABC-5统计每种策略在每个周期内成功更新的蜜源个数来计算策略的选用概率,而ABC-6是统计成功更新的蜜源的适应度改进量来计算概率. 除选择方式不同外,这两种算法的其他部分与ABC-MIG保持相同. 此外,把经典ABC作为参照基准也加入本节的对比实验.

表3给出了实验结果,测试维度为$D=30$. 从结果精度来看,与经典ABC相比,ABC-5、ABC-6以及ABC-MIG都有更好性能,这说明无论采用自适应选择还是随机选择,应用多元信息引导机制的ABC都要优于经典ABC. 在单峰函数上,从Wilcoxon秩和检验结果可看出,ABC-MIG与两种对比算法的性能基本上相当. 事实上,单峰函数的特点是搜索空间中仅存在一个全局最优解,适应度地形比较平滑,理论上采用自适应选择的方式会占有一定优势,但实际结果表明随机选择的方式同样非常具有竞争力,可能原因是多元信息引导机制中3种不同的信息引导方式能够优势互补,从而进一步提高了随机选择方式的有效性. 在15个多峰函数中,ABC-MIG在12个上取得了最优结果,性能要明显优于ABC-5和ABC-6. 这种结果是符合预期的,因为多峰函数的适应度地形中往往遍布局部极值,算法搜索的难度要远高于单峰函数,这种情况下自适应选择方式采用的历史经验反而有很大概率会出现误导,而随机选择方式有利于保持算法搜索过程中的多样性,使得随机选择要好于自适应选择. 在组合函数上,对比情况类似于多峰函数,ABC-5和ABC-6都未能在任一函数上优于ABC-MIG. 从表3最后两行的检验结果可看出,随机选择要优于自适应选择,并且算法结构也更简洁和易于实现.

4.4 与相关改进ABC算法对比

为进一步验证ABC-MIG性能,选用了以下5种相关的改进ABC进行对比:

GABC[4]:最优个体引导的ABC;

ECABC[5]:精英组引导的ABC;

ABCVSS[11]:可变搜索策略的ABC;

LLABC[6]:基于方向学习和精英学习双策略的ABC;

NABC[13]:基于最优邻居引导和邻域搜索机制的ABC.

选取上述5种算法的原因包括两方面:(1)相关性. ABC-MIG的解搜索方程涉及了优秀个体,因此对比算法的解搜索方程也应包含优秀个体(最优个体或精英个体). GABC是利用最优个体的开创性工作,非常有代表性;ECABC采用精英组的概念,本文基于适应度信息的解搜索方程也来源于该算法. ABCVSS是一种多策略ABC,采用了5种不同的解搜索方程进行自适应选用,ABC-MIG采用了3种解搜索方程,在某种程度上也可看作是多策略ABC,因此选取了ABCVSS进行对比. NABC采用邻域搜索技术,ABC-MIG对其进行了微调,因此有必要进行对比. LLABC采用方向学习机制用于辅助精英学习策略,该机制对于引导算法搜索很有帮助,ABC-MIG的多元信息引导机制也是如何引导算法搜索,因此与LLABC对比有一定的针对性. (2)全面性. 本文相关工作把现有ABC划分为3类,5种对比算法可涵盖这3类工作:GABC和ECABC属于第1类、ABCVSS为第2类、LLABC和NABC是第3类. 因此,与这5种算法对比还兼顾了全面性. 为确保对比公平,对比算法的参数设置与原文献保持相同,但对于种群规模,所有算法均设置为$SN=60$,测试维度为$D=30$和50.

表4给出了$D=30$的对比结果,最好结果以粗体突显. 在5个单峰函数上,ABC-MIG要优于对比算法,特别是在F01和F03上的结果精度更高. 对15个多峰函数,ABC-MIG在8个上要优于ECABC和NABC,在9个上优于ABCVSS和LLABC,在10个上优于GABC;而对比算法中表现最好的ABCVSS仅在3个上优于ABC-MIG. 当然,ABC-MIG在F14和F17上的结果不理想,分别差于4个和3个对比算法,原因是这两个函数虽为多峰类型,但其适应度地形是一种“漏斗型”的单峰函数,采用适应度信息引导的算法会有一定优势,比如GABC要优于ABC-MIG. 在8个组合函数上,ABC-MIG在4个上取得了最优结果,其性能与ABCVSS和LLABC类似. 对ABCVSS而言,它在4个上优于ABC-MIG,但也在4个上差于ABC-MIG. ABCVSS的优良性能来源于采用了5种不同的策略且能自适应选择使用,这有利于提高搜索的灵活性. LLABC在3个上优于ABC-MIG,但在4个上更差. LLABC的不错性能是因其方向学习机制可使算法在搜索过程中动态调整方向,有利于提高搜索的有效性. 图1中给出了上述6种算法在4个代表性函数上的收敛曲线图,包括1个单峰函数、2个多峰函数、以及1个组合函数,可看出ABC-MIG的总体收敛速度

表3　随机选择对比自适应选择的实验结果

函数	ABC	ABC-5	ABC-6	ABC-MIG
F01	5.15×10^{-13}	**0.00**	**0.00**	**0.00**
F02	8.01×10^{6}	5.59×10^{6}	**5.42×10^{6}**	6.03×10^{6}
F03	3.88×10^{8}	**3.19×10^{7}**	4.83×10^{7}	4.16×10^{7}
F04	7.08×10^{4}	**3.82×10^{4}**	3.96×10^{4}	3.83×10^{4}
F05	7.54×10^{-13}	2.24×10^{-13}	2.16×10^{-13}	**1.10×10^{-13}**
F06	**1.31×10**	2.23×10	2.22×10	2.46×10
F07	1.01×10^{2}	5.50×10	5.69×10	**5.29×10**
F08	**2.09×10**	2.10×10	2.10×10	**2.09×10**
F09	2.90×10	**2.44×10**	2.55×10	2.54×10
F10	1.71	1.17	1.13	**8.08×10^{-1}**
F11	8.91×10^{-14}	**0.00**	**0.00**	**0.00**
F12	2.45×10^{2}	1.13×10^{2}	1.11×10^{2}	**8.64×10**
F13	3.00×10^{2}	1.72×10^{2}	1.66×10^{2}	**1.48×10^{2}**
F14	1.95	7.88	6.48	**2.80**
F15	3.56×10^{3}	**3.01×10^{3}**	3.07×10^{3}	3.17×10^{3}
F16	1.39	1.06	1.04	**9.98×10^{-1}**
F17	3.06×10	3.05×10	3.05×10	**3.04×10**
F18	**3.00×10**	**3.00×10**	**3.00×10**	**3.00×10**
F19	4.18×10^{-1}	4.95×10^{-1}	5.08×10^{-1}	**3.56×10^{-1}**
F20	1.18×10	1.02×10	1.01×10	**9.91**
F21	**1.77×10^{2}**	3.23×10^{2}	3.10×10^{2}	3.2×10^{2}
F22	**2.95×10**	1.18×10^{2}	8.83×10	1.04×10^{2}
F23	4.67×10^{3}	4.28×10^{3}	4.30×10^{3}	**4.24×10^{3}**
F24	2.85×10^{2}	2.33×10^{2}	**2.32×10^{2}**	**2.32×10^{2}**
F25	3.12×10^{2}	**2.71×10^{2}**	2.73×10^{2}	2.80E+02
F26	2.01×10^{2}	**2.00×10^{2}**	**2.00×10^{2}**	**2.00×10^{2}**
F27	**4.00×10^{2}**	6.20×10^{2}	7.10×10^{2}	5.24×10^{2}
F28	**2.21×10^{2}**	3.00×10^{2}	2.93×10^{2}	2.93×10^{2}
+/=/−	3/5/20	0/19/9	0/19/9	—
Fried.	3.14	2.55	2.41	1.89

较快.

表5给出了$D=50$的对比结果,可看出总体对比情况类似于$D=30$,ABC-MIG在大部分函数上要优于对比算法. 具体而言,ABC-MIG在5个单峰函数上仍然取得了最好结果. 在15个多峰函数上的对比情况要好于$D=30$,$D=30$时8个函数上能取得最好结果,但$D=50$时却有12个,且在F17上的结果并未差于对比算法,这说明虽然函数的求解难度随维度增加而加大,但ABC-MIG却能保持较好的鲁棒性. 在组合函数上的对比情况也有改善,ABC-MIG仍可在其中的4个上取得最优,但ABCVSS却从$D=30$的4个更优减少到了$D=50$的3个,LLABC也从3个减少到了1个. 综合来看,ABC-MIG在$D=50$的结果非常有竞争力. 从表4和表5的最后一行Friedman检验结果来看,ABC-MIG均排在第一.

进一步对ABC-MIG的运行时间进行分析,记录其在CEC2013测试集上的实际CPU运行时间(单位:s),与上述5种改进ABC进行对比. 为确保对比公平,所有算法均在每个函数上独立运行30次,以平均CPU运行时间作为最终结果. 算法运行平台的配置信息如下:

CPU:Inter(R)Core i7-9750H

内存:16 GB

操作系统:Microsoft Windows 10 Professional

编程语言:Java

表6给出了算法的CPU运行时间情况,倒数第二行是算法在28个函数上的总体平均时间,最后一行为总体平均时间的排名情况. 可看出,总体运行时间最短的是ECABC,而ABC-MIG的时间最长. 该情况的主要原因是,多元信息引导机制涉及了位置信息和相似度信息,基于这两种信息构建优秀个体中心需计算个体间

表4 与相关改进ABC的对比结果($D=30$)

函数	GABC	ECABC	ABCVSS	LLABC	NABC	ABC-MIG
F01	3.33×10^{-13}	1.06×10^{-13}	2.35×10^{-13}	5.15×10^{-13}	4.62×10^{-13}	**0.00**
F02	9.41×10^{6}	9.67×10^{6}	1.04×10^{7}	8.07×10^{6}	7.99×10^{6}	**6.03×10^{6}**
F03	2.47×10^{8}	1.93×10^{8}	3.85×10^{8}	2.82×10^{8}	1.00×10^{8}	**4.16×10^{7}**
F04	6.40×10^{4}	8.30×10^{4}	8.50×10^{4}	7.14×10^{4}	4.62×10^{4}	**3.83×10^{4}**
F05	5.31×10^{-13}	1.78×10^{-13}	4.21×10^{-13}	6.59×10^{-13}	7.16×10^{-13}	**1.10×10^{-13}**
F06	1.54×10	2.46×10	1.65×10	**1.46×10**	1.98×10	2.46×10
F07	8.41×10	9.20×10	9.91×10	1.04×10^{2}	6.32×10	**5.29×10**
F08	**2.09×10**	**2.09×10**	**2.09×10**	**2.09×10**	2.10×10	**2.09×10**
F09	2.82×10	2.92×10	2.96×10	2.96×10	2.75×10	**2.54×10**
F10	1.53	**2.53×10^{-1}**	2.51	1.42	1.01	8.08×10^{-1}
F11	6.06×10^{-14}	1.89×10^{-15}	5.68×10^{-14}	5.87×10^{-14}	9.66×10^{-14}	**0.00**
F12	1.36×10^{2}	1.14×10^{2}	1.51×10^{2}	2.38×10^{2}	1.52×10^{2}	**8.64×10**
F13	2.03×10^{2}	1.64×10^{2}	2.26×10^{2}	2.96×10^{2}	1.87×10^{2}	**1.48×10^{2}**
F14	3.69×10^{-1}	1.50	**2.08×10^{-2}**	2.10	1.08	2.80
F15	3.95×10^{3}	3.69×10^{3}	3.79×10^{3}	3.64×10^{3}	**3.09×10^{3}**	3.17×10^{3}
F16	1.72	1.36	1.26	1.37	1.04	**9.98×10^{-1}**
F17	**2.96×10**	3.05×10	2.98×10	3.01×10	3.05×10	3.04×10
F18	3.00×10	3.00×10	2.95×10	**2.94×10**	3.00×10	3.00×10
F19	5.12×10^{-1}	4.74×10^{-1}	3.61×10^{-1}	**3.39×10^{-1}**	4.14×10^{-1}	3.56×10^{-1}
F20	1.16×10	1.16×10	1.18×10	1.19×10	1.02×10	**9.91**
F21	2.07×10^{2}	2.99×10^{2}	2.04×10^{2}	**1.75×10^{2}**	3.17×10^{2}	3.20×10^{2}
F22	9.15×10	8.21×10	**1.52×10**	5.92×10	1.08×10^{2}	1.04×10^{2}
F23	4.92×10^{3}	4.62×10^{3}	4.74×10^{3}	4.72×10^{3}	4.82×10^{3}	**4.24×10^{3}**
F24	2.77×10^{2}	2.76×10^{2}	2.83×10^{2}	2.82×10^{2}	2.38×10^{2}	**2.32×10^{2}**
F25	3.02×10^{2}	3.05×10^{2}	3.02×10^{2}	3.08×10^{2}	2.81×10^{2}	**2.80×10^{2}**
F26	2.01×10^{2}	**2.00×10^{2}**	2.01×10^{2}	2.01×10^{2}	**2.00×10^{2}**	**2.00×10^{2}**
F27	4.22×10^{2}	5.84×10^{2}	4.64×10^{2}	**4.00×10^{2}**	5.92×10^{2}	5.24×10^{2}
F28	2.62×10^{2}	3.00×10^{2}	2.46×10^{2}	**2.01×10^{2}**	3.00×10^{2}	2.93×10^{2}
+/=/−	4/5/19	3/7/18	7/3/18	4/6/18	2/10/16	—
Friedman	3.80	3.59	3.79	3.80	3.70	2.32

的欧氏距离和余弦相似度，导致ABC-MIG的运行时间更长. 然而，需要指出的是，这种额外计算开销会随着问题求解难度的增加而相对减少，因为欧氏距离和余弦相似度的计算开销对于不同难度的问题基本相当，使得算法运行时间主要由问题的求解时间来决定，例如：在F01上，ABC-MIG的平均时间是ECABC的3.83倍，但在F09上减少至了1.07倍，在F26上更是降至了1.05倍. 因此，可以得出的是，随着问题求解难度的增加，ABC-MIG运行时间的影响会大幅降低. 结合ABC-MIG在收敛精度上的明显优势，其总体性能仍是相当有竞争力.

4.5 在实际优化问题上的应用

为进一步验证ABC-MIG的性能，将其用于求解一个实际优化问题：扩频雷达的多相位编码设计问题[18]. 该问题是指，在设计采用脉冲压缩的雷达系统时，如何选定恰当的波形，使得自相关函数$\phi(x)$的最大样本模块能够实现最小化，即最小化如下目标函数$f(x)$：

$$\text{Global min}\, f(x)=\max\left\{\phi_1(x),\cdots,\phi_{2m}(x)\right\} \tag{19}$$

其中，$x\in X, X\in\left\{\left(x_1,\cdots,x_D\right)\in R^D \mid 0\leqslant x_j\leqslant 2\pi\right\}$，且$j=1,\cdots,D$和$m=2D-1$，代表了对称的相位差. 自相关函数$\phi(x)$的表达式如下：

$$\phi_{2p-1}(x)=\sum_{j=p}^{D}\cos\left(\sum_{k=|2p-j-1|}^{j}x_k\right) \tag{20}$$

$$\phi_{2q}(x)=0.5+\sum_{j=q+1}^{D}\cos\left(\sum_{k=|2q-j|+1}^{j}x_k\right) \tag{21}$$

$$\phi_{m+i}(x)=-\phi_i(x) \tag{22}$$

其中，$p=1,\cdots,D$；$q=1,\cdots,D-1$；$i=1,\cdots,m$. 该问题是连续变量区间内的最小-最大(min-max)全局优化问

表5　与相关改进ABC的对比结果($D=50$)

函数	GABC	ECABC	ABCVSS	LLABC	NABC	ABC-MIG
F01	7.96×10^{-13}	2.43×10^{-13}	6.21×10^{-13}	1.08×10^{-12}	1.09×10^{-12}	**1.74×10^{-13}**
F02	1.39×10^{7}	1.58×10^{7}	1.85×10^{7}	1.25×10^{7}	8.05×10^{6}	**7.39×10^{6}**
F03	1.37×10^{9}	1.51×10^{9}	1.84×10^{9}	1.46×10^{9}	2.41×10^{8}	**1.52×10^{8}**
F04	1.28×10^{5}	1.50×10^{5}	1.57×10^{5}	1.37×10^{5}	6.92×10^{4}	**6.33×10^{4}**
F05	1.14×10^{-12}	3.60×10^{-13}	9.28×10^{-13}	1.32×10^{-12}	1.63×10^{-12}	**2.92×10^{-13}**
F06	4.30×10	4.61×10	4.32×10	**4.26×10**	4.57×10	4.51×10
F07	1.27×10^{2}	1.26×10^{2}	1.40×10^{2}	1.47×10^{2}	8.58×10	**8.20×10**
F08	**2.11×10**	**2.11×10**	**2.11×10**	**2.11×10**	**2.11×10**	**2.11×10**
F09	5.68×10	5.79×10	5.89×10	5.84×10	5.44×10	**5.23×10**
F10	2.03	**7.08×10^{-1}**	3.16	1.67	1.64	1.43
F11	1.74×10^{-13}	5.50×10^{-14}	1.46×10^{-13}	1.33×10^{-13}	2.86×10^{-13}	**2.65×10^{-14}**
F12	4.44×10^{2}	3.49×10^{2}	4.39×10^{2}	6.19×10^{2}	3.07×10^{2}	**2.56×10^{2}**
F13	5.37×10^{2}	4.64×10^{2}	5.87×10^{2}	7.05×10^{2}	4.37×10^{2}	**3.77×10^{2}**
F14	3.19	2.48	**3.17×10^{-2}**	5.89	4.45	8.88
F15	8.27×10^{3}	7.75×10^{3}	7.68×10^{3}	7.81×10^{3}	7.21×10^{3}	**7.11×10^{3}**
F16	2.38	1.66	1.61	1.83	1.41	**1.35**
F17	**5.08×10**	**5.08×10**	**5.08×10**	5.09×10	5.09×10	**5.08×10**
F18	5.02×10	5.02×10	5.02×10	**4.94×10**	5.02×10	5.02×10
F19	1.16	7.99×10^{-1}	7.61×10^{-1}	8.75×10^{-1}	9.86×10^{-1}	**7.08×10^{-1}**
F20	2.08×10	2.07×10	2.11×10	2.10×10	1.86×10	**1.84×10**
F21	2.79×10^{2}	3.06×10^{2}	**2.37×10^{2}**	2.42×10^{2}	3.00×10^{2}	3.20×10^{2}
F22	3.98×10	3.84×10	**9.47**	2.33×10	1.28×10	6.10×10
F23	9.81×10^{3}	9.65×10^{3}	9.91×10^{3}	9.65×10^{3}	1.02×10^{4}	**9.10×10^{3}**
F24	3.59×10^{2}	3.58×10^{2}	3.59×10^{2}	3.64×10^{2}	3.10×10^{2}	**3.07×10^{2}**
F25	**4.05×10^{2}**	4.14×10^{2}	4.10×10^{2}	4.17×10^{2}	4.16×10^{2}	4.08×10^{2}
F26	2.02×10^{2}	2.02×10^{2}	2.02×10^{2}	2.02×10^{2}	**2.01×10^{2}**	**2.01×10^{2}**
F27	9.47×10^{2}	1.54×10^{3}	1.26×10^{3}	**7.09×10^{2}**	1.51×10^{3}	1.33×10^{3}
F28	**4.00×10^{2}**	**4.00×10^{2}**	**4.00×10^{2}**	**4.00×10^{2}**	**4.00×10^{2}**	**4.00×10^{2}**
+/=/−	4/5/19	2/7/19	5/6/17	4/4/20	2/12/14	—
Friedman	3.88	3.55	3.84	4.09	3.46	2.18

题,有非线性非凸特点,包含大量的局部极值,求解难度非常大,已被证明是NP难问题[18].

实验中,选择该问题的两个最难实例进行求解,即:D=19和D=20. 同时,4.4节中对比的5种相关改进ABC也用于求解该问题. 关于这6种ABC的参数设置与前面实验保持相同,但目标函数的最大评估次数为300 000. 每个算法独立运行30次,以均值作为最终结果,表7给出了对比结果. 可看出,ABC-MIG在两个实例上均取得了最优结果,结果精度较GABC、ECABC、ABCVSS有较大提升,相对于LLABC和NABC有提升. 在该问题上的应用也进一步验证了ABC-MIG性能是有竞争力的.

4.6　进一步讨论

本节对ABC-MIG做进一步剖析,就影响算法性能的重要步骤和主要因素进行定量分析和讨论,包括:(1)算法重要步骤的定量分析和讨论. 从ABC-MIG结构来看,雇佣蜂阶段和观察蜂阶段是算法的两个重要步骤,分别采用了随机选择和贪婪选择来实现多元信息引导机制,因此有必要厘清这两个阶段对算法性能的贡献;(2)算法主要因素的定量分析和讨论. 多元信息引导机制是影响ABC-MIG性能的主要因素,同时采用了适应度、位置、以及相似度信息,这3种信息的协同合作是该机制能取得良好效果的关键,那么有必要分析这3种信息的引导作用是否存在差异. 为此,就这两个方面开展了相应实验,具体如下:

(1)算法重要步骤的定量分析和讨论

为量化雇佣蜂阶段和观察蜂阶段对算法性能的贡献,实验中分别统计这两阶段对蜜源的成功更新次数,再对两者的次数做归一化处理,折算为百分比形

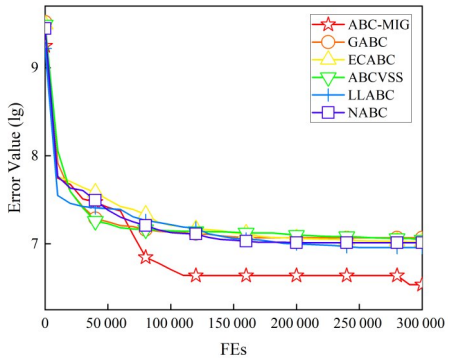
表6 与相关改进ABC的CPU运行时间对比结果 时间:s

函数	GABC	ECABC	ABCVSS	LLABC	NABC	ABC-MIG
F01	0.81	**0.59**	0.86	0.83	0.90	2.26
F02	**0.77**	0.87	**0.77**	**0.77**	0.81	2.66
F03	1.17	**1.06**	1.18	1.20	1.23	2.96
F04	0.52	**0.48**	0.51	0.54	0.55	2.24
F05	0.35	**0.34**	0.36	0.38	0.39	1.93
F06	0.47	**0.44**	0.46	0.48	0.49	2.23
F07	1.94	**1.84**	1.93	1.97	2.05	3.69
F08	1.57	**1.50**	1.58	1.61	1.67	3.32
F09	25.31	**24.92**	25.42	25.79	25.57	26.78
F10	0.87	**0.83**	0.87	0.88	0.89	2.66
F11	0.77	**0.72**	0.77	0.78	0.79	2.36
F12	1.55	**1.49**	1.59	1.61	1.57	3.29
F13	1.69	**1.58**	1.72	1.76	1.71	3.38
F14	0.91	0.96	**0.85**	1.00	1.01	2.83
F15	1.29	**1.24**	1.39	1.31	1.25	3.03
F16	7.66	**7.58**	8.20	7.98	7.66	9.29
F17	0.58	**0.54**	0.60	0.62	0.59	2.26
F18	1.22	**1.14**	1.26	1.27	1.23	2.90
F19	0.58	**0.56**	0.62	0.61	0.60	2.40
F20	1.19	**1.16**	1.21	1.28	1.19	2.89
F21	3.48	**3.27**	3.53	3.80	3.50	5.01
F22	3.25	**3.16**	3.36	3.89	3.25	4.96
F23	4.08	4.06	4.49	4.64	**4.05**	5.77
F24	**29.21**	29.45	30.44	31.20	29.22	30.69
F25	28.83	**28.74**	30.48	31.11	29.20	30.59
F26	32.21	**32.04**	33.14	33.25	32.95	33.83
F27	31.96	**31.66**	32.44	32.51	32.13	33.19
F28	7.97	**7.70**	8.00	8.13	7.99	9.23
平均时间	6.86	**6.78**	7.07	7.18	6.94	8.52
排名	2	**1**	4	5	3	6

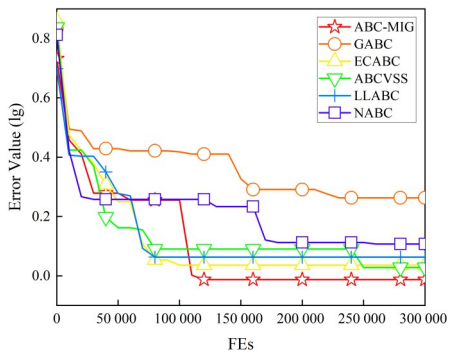
式,以此来区分两阶段对算法性能的贡献差异. 注意,因ABC-MIG是迭代式算法,因此应统计两阶段在算法整个迭代过程中的成功更新的总次数. 还需说明的是,实验中并未采用统计蜜源成功更新量的方式,原因是雇佣蜂阶段在观察蜂阶段之前,这使得雇佣蜂阶段的更新量要明显多于观察蜂阶段,进而导致两者的对比不公平. 为避免随机因素的影响,ABC-MIG独立运行30次,取两阶段成功更新总次数的平均值作为最终结果. 图2以雷达图的形式给出了两阶段对算法性能贡献的对比情况,蓝色为雇佣蜂阶段的贡献率,红色为观察蜂阶段的贡献率,贡献率越高所对应的覆盖半径越大. 可看出,在大部分函数上雇佣蜂阶段的贡献率要高于观察蜂阶段,说明雇佣蜂阶段在算法的整个迭代过程中的成功更新总次数更多,可能原因有两点:(1)雇佣蜂阶段以随机选择来使用3种信息引导,比观察蜂阶段以贪婪选择使用单一信息引导更易成功更新食物源;(2)雇佣蜂阶段要先于观察蜂阶段执行,虽然统计的成功更新次数相比成功更新量更加合理,但在一定程度上还会存在雇佣蜂阶段要“自然占优”的情况,使得其贡献率更高. 可得出结论,雇佣蜂阶段和观察蜂阶段对算法性能的贡献各有千秋,两者的结合能优势互补,发挥出多元信息引导机制的作用.

(2)算法主要因素的定量分析和讨论

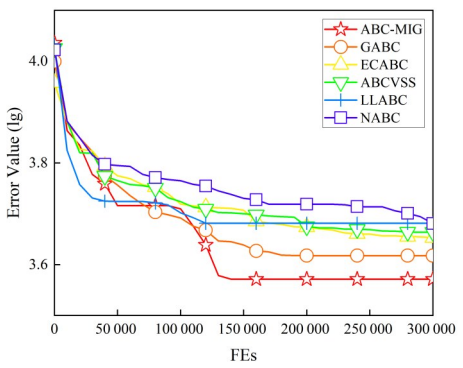
为量化适应度、位置以及相似度信息在多元信息引导机制中的作用,实验中统计3种信息在雇佣蜂阶段成功引导蜜源更新的改进量,并对改进量做归一化处理,折算成百分比形式进行对比. 类似于对算法重要步骤的定量分析,实验中同样统计3种信息在算法整个迭代过程中的成功更新总量. 值得注意的是,



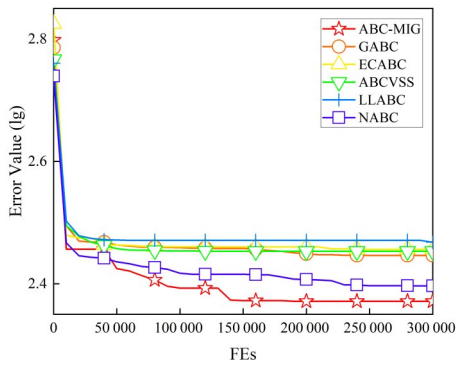
(*a*) F02



(*b*) F16



(*c*) F20



(*d*) F24

图 1　与相关改进 ABC 的收敛曲线对比情况

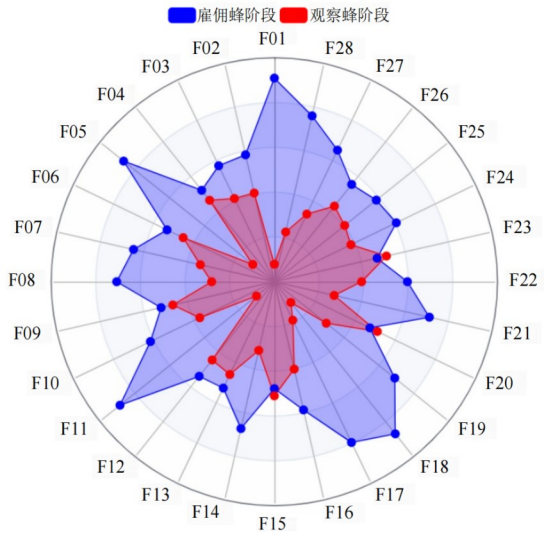


图 2　雇佣蜂阶段与观察蜂阶段的贡献对比情况

表 7　求解实际优化问题的实验结果

算法	*D*=19	*D*=20
GABC	1.01E+00	1.00E+00
ECABC	1.08E+00	1.05E+00
ABCVSS	1.02E+00	1.01E+00
LLABC	9.45E-01	9.49E-01
NABC	9.23E-01	9.73E-01
ABC-MIG	**8.93E-01**	**9.05E-01**

此处仅关注雇佣蜂阶段的情况,因为该阶段对 3 种信息引导是以随机方式进行选择,即 3 种解搜索方程被选用的概率均等;同时,此处未统计更新次数,原因是统计操作限定在雇佣蜂阶段,更新量相比更新次数更加合理. 类似地,ABC-MIG 独立运行 30 次,取成功更新总量的平均值作为最终结果,图 3 给出了 3 种信息引导作用的百分比情况. 可看出,在多数函数上 3 种信息的百分比情况基本相当,说明 3 种信息的引导作用较为均衡,不会出现某种信息引导发挥主要作用的情况. 然而,在个别函数上也还存在百分比失衡的情况,比如:F03、F07、F28. 在 F03 和 F07 上,适应度信息的引导作用最强;但在 F28 上,位置信息的引导作用最强. 这种情况说明,某些函数对不同的信息引导有较强偏好,比如:对偏好适应度信息引导的 F03 和 F07 而言,NABC 的结果也较为优秀,原因是 NABC 采用了最佳邻居作为解搜索方程的起始项,能表现出较强开采能力;同样地,对偏好位置信息引导的 F28 而言,LLABC 有较好表现,因其采用了方向学习机制,能及时调整搜索方向. 总体来说,3 种信息在多元信息引导机制中能发挥出较为均衡的作用,相比于单一信息更有优势.

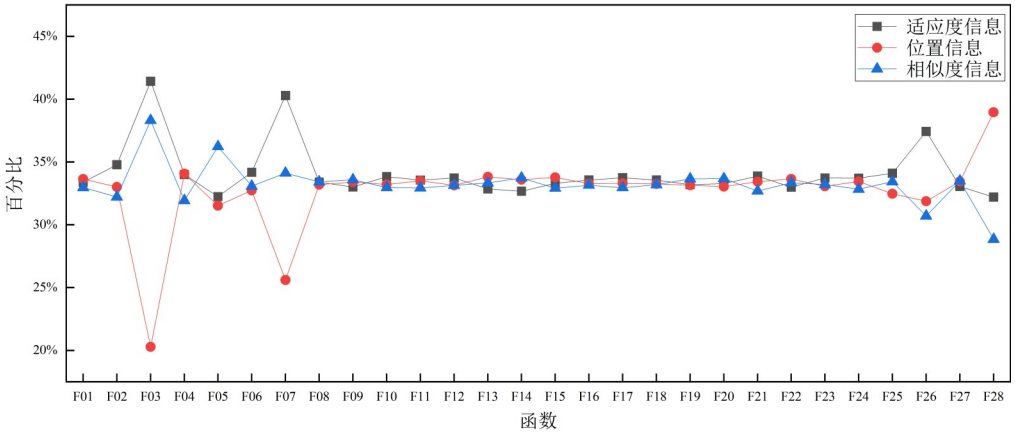


图3 多元信息引导机制中三种信息引导作用的对比情况

5 结论

经典ABC存在勘探能力强,而开采能力弱的不足,现有的很多相关改进ABC通常会在解搜索方程中利用优秀个体来引导算法搜索,从而增强开采能力. 然而,这种方式易导致算法的搜索能力过于贪婪,引起早熟收敛等问题. 为此,本文提出了一种多元信息引导的ABC,主要贡献包括两点:(1)设计了多元信息引导机制,同时采用适应度、位置以及相似度信息进行协同引导,并构建了3种对应的解搜索方程,在雇佣蜂阶段和观察蜂阶段分别采用随机选择和贪婪选择的方式来使用这3种解搜索方程;(2)在侦察蜂阶段采用了一种微调的邻域搜索机制用于处理被放弃蜜源,避免随机初始化方法容易破坏搜索经验的问题.

在CEC2013测试集和一个实际优化问题上进行了大量的实验验证,可得到如下6点结论:(1)对算法涉及的limit参数取值进行了实验分析,得出limit取值不宜过大或过小,建议设置limit=200;(2)对多元信息引导机制和微调后的邻域搜索机制进行了消融实验,验证了这两种机制的有效性;(3)对多元信息引导机制中的随机选择方式进行了专门分析,与两种不同的自适应选择方式做了实验对比,结果表明随机选择方式具有更好效果;(4)与5种知名的相关改进ABC进行了对比,不论在CEC2013测试集,还是实际优化问题上,本文算法性能都更具竞争力;(5)因位置信息和相似度信息分别涉及欧式距离和相似度的计算,这会增加本文算法的运行时间;(6)多元信息引导机制中的3种信息在大部分测试函数上能发挥出较为均衡的引导作用. 在未来,如何进一步提高多元信息引导机制中不同解搜索方程的使用方式是值得继续研究的一项工作,比如:可尝试结合问题适应度地形的不同特征来进行选择. 同时,还可尝试把本文算法用于求解更多的实际优化问题,比如:光伏电池的参数辨识问题.

参考文献

[1] 陈健瑞, 王景璟, 侯向往, 等. 挺进深蓝: 从单体仿生到群体智能[J]. 电子学报, 2021, 49(12): 2458-2467.
CHEN J R, WANG J J, HOU X W, et al. Advance into ocean: From Bionic Monomer to swarm intelligence[J]. Acta Electronica Sinica, 2021, 49(12): 2458-2467. (in Chinese)

[2] ZHAN Z, SHI L, TAN K C, et al. A survey on evolutionary computation for complex continuous optimization[J]. Artificial Intelligence Review, 2022, 55(1): 59-110.

[3] ZHOU X Y, WU Y L , ZHONG M S, et al. Artificial bee colony algorithm based on multiple neighborhood topologies[J]. Applied Soft Computing, 2021, 111: 107697.

[4] ZHU G P, KWONG S. Gbest-guided artificial bee colony algorithm for numerical function optimization[J]. Applied Mathematics and Computation, 2010, 217(7): 3166-3173.

[5] KONG D, CHANG T, DAI W, et al. An improved artificial bee colony algorithm based on elite group guidance and combined breadth-depth search strategy[J]. Information Sciences, 2018, 442: 54-71.

[6] GAO W F, SHENG H L, WANG J, et al. Artificial bee colony algorithm based on novel mechanism for fuzzy portfolio selection[J]. IEEE Transactions on Fuzzy Systems, 2019, 27(5): 966-978.

[7] WANG X, ZHANG Y, SUN X, et al. Multi-objective feature selection based on artificial bee colony: An acceleration approach with variable sample size[J]. Applied Soft Computing, 2020, 88: 106041.

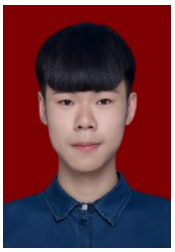
[8] ZHANG M, PALADE V, WANG Y, et al. Attention-based word embeddings using artificial bee colony algorithm for aspect-level sentiment classification[J]. Information Sciences, 2021, 545: 713-738.

[9] ZHOU X Y, LU J X, HUANG J H, et al. Enhancing artifi-

cial bee colony algorithm with multi-elite guidance[J]. Information Sciences, 2021, 543: 242-258.
[10] WANG H, WU Z J, RAHNAMAYAN S, et al. Multi-strategy ensemble artificial bee colony algorithm[J]. Information Sciences, 2014, 279: 587-603.
[11] KIRAN M S, HAKLI H, GUNDUZ M, et al. Artificial bee colony algorithm with variable search strategy for continuous optimization[J]. Information Sciences, 2015, 300: 140-157.
[12] ZHOU X Y, WANG H, WANG M W, et al. Enhancing the modified artificial bee colony algorithm with neighborhood search[J]. Soft Computing, 2017, 21(10): 2733-2743.
[13] PENG H, DENG C S, WU Z J, et al. Best neighbor-guided artificial bee colony algorithm for continuous optimization problems[J]. Soft Computing, 2019, 23(18): 8723-8740.
[14] LIANG J J, QU B Y, SUGANTHAN P N, et al. Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization[R]. Zhengzhou: Zhengzhou University, 2013.
[15] DERRAC J, GARCÍA S, MOLINA D, et al. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms[J]. Swarm and Evolutionary Computation, 2011, 1(1): 3-18.
[16] WANG H, WANG W J, ZHOU X Y, et al. Artificial bee colony algorithm based on knowledge fusion[J]. Complex & Intelligent Systems, 2021, 7(3): 1139-1152.
[17] XUE Y, JIANG J M, ZHAO B P, et al. A self-adaptive artificial bee colony algorithm based on global best for global optimization[J]. Soft Computing, 2018, 22(9): 2935-2952.
[18] MLADENOVIĆ N, PETROVIĆ J, KOVAČEVIĆ-VUJČIĆ V, et al. Solving spread spectrum radar polyphase code design problem by tabu search and variable neighborhood search[J]. European Journal of Operational Research, 2003, 151(2): 389-399.

作者简介

周新宇　男,1987年1月出生于江西省修水县. 现为江西师范大学计算机信息工程学院副教授、博士生导师. 主要研究方向为智能优化算法及应用,在国内外发表学术论文40余篇.
E-mail: xyzhou@jxnu.edu.cn



刘　颖　男,1999年7月出生于江西省吉安市. 2022年6月于江西师范大学获计算机科学与技术专业学士学位,现为长沙理工大学硕士研究生. 主要研究方向为智能优化算法和数据中心网络.
E-mail: yingliu@stu.csust.edu.cn

吴艳林　男,1994年12月出生于安徽省亳州市. 2022年6月于江西师范大学获软件工程专业硕士学位,现为广东工业大学博士研究生. 主要研究方向为智能优化算法、组合优化.
E-mail: ylwucomm@foxmail.com

郭京蕾　女,1977年1月出生于湖北省武汉市. 现为华中师范大学计算机学院副教授、硕士生导师. 主要研究方向为智能优化算法及应用.
E-mail: guojinglei@mail.ccnu.edu.cn