GWO 与 ABC 的混合优化算法及其聚类优化

张新明[1,2],王 霞[1],康 强[1],程金凤[1]
(1. 河南师范大学计算机与信息工程学院,河南新乡 453007;2. 河南省高校计算智能与数据挖掘工程技术研究中心,河南新乡 453007)

摘 要: 灰狼优化算法(Grey Wolf Optimizer,GWO)和人工蜂群算法(Artificial Bee Colony,ABC)是两种流行且高效的群智能优化算法. GWO 具有局部搜索能力强等优势,但存在全局搜索能力弱等缺陷;而 ABC 具有全局搜索能力强等优点,但存在收敛速度慢等不足. 为实现二者优势互补,提出了一种 GWO 与 ABC 的混合算法(Hybrid GWO with ABC,HGWOA). 首先,使用静态贪心算法替代 ABC 雇佣蜂阶段中的动态贪心算法来强化探索能力,同时为弥补其收敛速度降低的不足,提出一种新型的搜索蜜源方式;然后,去掉影响收敛速度的侦查蜂阶段,在雇佣蜂阶段再添加反向学习策略,以避免搜索陷入局部最优;最后,为了平衡以上雇佣蜂阶段的探索能力,在观察蜂阶段,自适应融合 GWO,以便增强开采能力和提高优化效率. 大量的函数优化和聚类优化的实验结果表明,与 state-of-the-art 方法相比,HGWOA 具有更好的优化性能及更强的普适性,且能更好地解决聚类优化问题.

关键词: 智能优化算法; 灰狼优化算法; 人工蜂群算法; 混合优化算法; 聚类优化
中图分类号: TP181 **文献标识码:** A **文章编号:** 0372-2112 (2018)10-2430-13
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2018.10.017

Hybrid Grey Wolf Optimizer with Artificial Bee Colony and Its Application to Clustering Optimization

ZHANG Xin-ming[1,2], WANG Xia[1], KANG Qiang[1], CHENG Jin-feng[1]
(1. *College of Computer and Information Engineering, Henan Normal University, Xinxiang, Henan* 453007, *China*;
2. *Engineering Technology Research Center for Computing Intelligence & Data Mining of Henan Province, Xinxiang, Henan* 453007, *China*)

Abstract: Grey Wolf Optimizer (GWO) and Artificial Bee Colony (ABC) are two popular and efficient intelligent optimization algorithms. GWO has some features such as strong exploitation but weak exploration. ABC has other ones such as strong global search ability but slow convergence. In order to realize their complementary advantages, a hybrid GWO with ABC (HGWOA) was proposed. Firstly, a static greedy algorithm was used to replace the dynamic greedy algorithm in the employed bee phase to enhance the exploration ability, and a new search method was created to make up for the lost convergence quality. Secondly, the scout bee phase which affects the convergence speed was removed, and an opposition learning strategy was embedded into the employed bee phase to keep the algorithm from falling into the local optima. Finally, in order to balance the exploration ability of the employed bee phase, GWO was added to the onlooker bee phase to strengthen the exploitation and improve the optimization efficiency. Experimental results on many function and clustering optimization problems show that compared with state-of-the-art methods, HGWOA has better optimization performance and stronger universality and it can solve clustering optimization problems more efficiently.

Key words: intelligent optimization algorithm; grey wolf optimizer; artificial bee colony algorithm; hybrid optimization algorithm; clustering optimization

1 引言

随着社会的发展,亟待优化的问题越来越多且越来越复杂,传统的优化方法远远不能满足现代社会的需求,由此群智能优化算法应运而生[1]. 这是一类基于种群的启发式随机优化算法,受启发于自然界生物群体的觅食行为等. 典型的群智能优化算法主要包括粒子群优化(Particle Swarm Optimization,PSO)算法[2]、灰

收稿日期:2018-01-10;修回日期:2018-06-27;责任编辑:孙瑶
基金项目:河南省重点科技攻关项目(No. 132102110209);河南省高等学校重点科研项目(No. 19A520026)

狼优化算法(Grey Wolf Optimizer,GWO)[3]和人工蜂群算法(Artificial Bee Colony,ABC)[4]等等. 因其结构简单、易于实现、调节参数少等优点,群智能优化算法已被广泛应用于解决函数优化、聚类优化等问题[3~7]. 但依据无免费午餐定理[8],没有一种群智能优化算法能够独立解决所有的优化问题,每种算法都有自己的优势和劣势,因此许多学者提出了混合的智能优化算法,通过算法间的混合,达到优势互补,获得较好的普适性. 本文重点研究 GWO 和 ABC 的混合,由于 GWO 是最近提出的群智能优化算法,有许多地方值得研究和完善,并且国内外目前还没有 GWO 与 ABC 混合算法的研究,故研究它们的混合是有意义的.

GWO 是 Mirjalili 等人提出的新型群智能优化算法[3]. GWO 具有收敛速度快,局部搜索能力强等优势,但存在全局搜索能力不足等问题[6]. 因此,一些 GWO 的改进算法被提出,其中包含混合算法. 如 Vikram 等人[9]提出了一种 GWO 和 PSO 的混合算法,用于解决单个区域组合问题. 张新明等人[5]提出了 GWO 和差分进化的混合算法,该算法采用 GWO 提高局部搜索能力,采用自适应调节参数的差分进化算法提高全局搜索能力. Jayabarathi 等人[10]提出了一种混合的 GWO,它融合了进化算法中的变异和交叉算子,以此提高了算法的优化性能,并用于解决经济调度问题.

ABC 是 Karaboga 等人提出的群智能优化算法[4]. ABC 具有全局搜索能力强等优势,但存在收敛速度慢等不足[11]. 为解决其不足,一些学者提出了 ABC 的混合算法. 例如,Liang 等人[11]提出了一种融合自适应差分算子的改进 ABC,其中利用自适应差分算子提高了算法的探索能力,同时在观察蜂阶段提出一种新的选择机制来提高算法的开采能力. Shidrokh 等人[12]提出了一种基于 ABC 和 PSO 的混合算法,其中,通过 PSO 来加强 ABC 的三个阶段:初始化阶段、雇佣蜂阶段以及观察蜂阶段,提高了 ABC 的收敛速度. Kefayat 等人[13]提出了一种蚁群算法(Ant Colony Optimization,ACO)与 ABC 的混合算法,该算法发挥了 ABC 的全局搜索能力和 ACO 的局部搜索能力的优势.

本文针对 GWO 和 ABC 存在的优势和不足,提出了一种 GWO 与 ABC 的混合优化算法(Hybrid GWO with ABC,HGWOA). 首先,将 GWO 融入到改进的 ABC 中,构建 HGWOA,以便两种算法优势互补;最后,将 HGWOA 用于不同类型的函数优化以及聚类优化上,以获得较好的优化性能.

2　灰狼优化算法与人工蜂群算法

2.1　灰狼优化算法

GWO 受启发于自然界中灰狼种群的社会等级制度和狩猎行为[3]. 在灰狼社会等级制度中,存在着四个社会等级,从高到低依次为 α、β、δ 和 ω 狼. 狩猎过程主要分为三个阶段:跟踪、靠近猎物阶段;追捕、包围和骚扰猎物阶段;攻击猎物阶段[3,5].

包围行为的数学模型为

$$\boldsymbol{D} = |\boldsymbol{C} \cdot \boldsymbol{X}_p(t) - \boldsymbol{X}(t)| \tag{1}$$

$$\boldsymbol{X}(t+1) = \boldsymbol{X}_p(t) - \boldsymbol{A} \cdot \boldsymbol{D} \tag{2}$$

其中,t 表示当前迭代次数,$\boldsymbol{A} = 2\boldsymbol{a}\boldsymbol{r}_1 - \boldsymbol{a}$,$\boldsymbol{C} = 2\boldsymbol{r}_2$,$\boldsymbol{X}_p$ 表示猎物的位置向量,$\boldsymbol{X}$ 表示灰狼的位置向量,$\boldsymbol{a}$ 是从 2 到 0 线性递减的参数,$\boldsymbol{r}_1$ 和 $\boldsymbol{r}_2$ 表示在[0,1]中均匀分布的随机向量.

跟踪猎物的数学模型为

$$\boldsymbol{D}_\alpha = |\boldsymbol{C}_1 \cdot \boldsymbol{X}_\alpha(t) - \boldsymbol{X}(t)| \tag{3}$$

$$\boldsymbol{D}_\beta = |\boldsymbol{C}_2 \cdot \boldsymbol{X}_\beta(t) - \boldsymbol{X}(t)| \tag{4}$$

$$\boldsymbol{D}_\delta = |\boldsymbol{C}_3 \cdot \boldsymbol{X}_\delta(t) - \boldsymbol{X}(t)| \tag{5}$$

$$\boldsymbol{X}_1 = \boldsymbol{X}_\alpha(t) - \boldsymbol{A}_1 \cdot \boldsymbol{D}_\alpha \tag{6}$$

$$\boldsymbol{X}_2 = \boldsymbol{X}_\beta(t) - \boldsymbol{A}_2 \cdot \boldsymbol{D}_\beta \tag{7}$$

$$\boldsymbol{X}_3 = \boldsymbol{X}_\delta(t) - \boldsymbol{A}_3 \cdot \boldsymbol{D}_\delta \tag{8}$$

$$\boldsymbol{X}(t+1) = (\boldsymbol{X}_1 + \boldsymbol{X}_2 + \boldsymbol{X}_3)/3 \tag{9}$$

在狩猎过程中,由 α、β 和 δ 狼定位猎物的位置并形成一个包围圈,ω 狼在 α、β 和 δ 狼的引导下,围捕猎物. GWO 的流程见图 1.

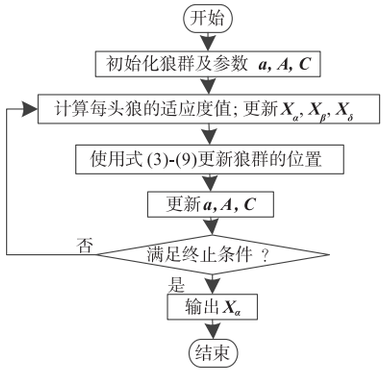

图 1　GWO的流程图

GWO 首先在搜索空间中随机初始化狼群的位置,然后在每次迭代中,依据式(3)~(9)更新其位置,随后计算其适应度值并依据其适应度值更新当前最优的三个解(静态更新 α、β 和 δ 狼位置)见图 1. GWO 的主要优势:①由于所有灰狼均依据 α、β 和 δ 狼的位置进行更新,故狼群会迅速趋向当前最优位置,能较快收敛到最优解,体现出很好的局部搜索能力;②静态更新有助于稳定性;③每代仅保存三个最优解,故占存储空间少.

2.2　人工蜂群算法

ABC 源自于蜜蜂的觅食行为[4]. 在 ABC 中,蜂群被划分为雇佣蜂、观察蜂和侦查蜂,搜索经过对应的三个阶段. 雇佣蜂和观察蜂的数量各占蜜源数量的一半. 一

个蜜源的位置代表优化问题的一个候选解,蜜源中的花蜜量代表相应候选解的适应度值.

2.2.1 雇佣蜂阶段

雇佣蜂负责寻找蜜源并保存蜜源信息. 它根据式(10)在蜜源附近搜索新的蜜源

$$v_{id} = x_{id} + \varphi_{id}(x_{id} - x_{kd}) \tag{10}$$

其中,x_{id}表示第 i 个蜜源第 d 维的值,v_i表示新蜜源的位置,k 表示在 N 个蜜源中随机选择的一个蜜源,且 $k \neq i$; d 表示随机选取的某一维,φ_{id}是[-1,1]之间均匀分布的随机数.

蜜源的花蜜量(适应度值)采用式(11)计算(假如解决的优化问题是最小值问题).

$$fit_i = \begin{cases} 1/(1+f_i), & f_i > 0 \\ 1 + |f_i|, & \text{otherwise} \end{cases} \tag{11}$$

其中,fit_i表示第 i 个蜜源的适应度值,f_i表示第 i 个解的目标函数值.

采用贪心算法更新蜜源如式(12)所示

$$x_i = \begin{cases} v_i, & fit(x_i) < fit(v_i) \\ x_i, & \text{otherwise} \end{cases} \tag{12}$$

式(12)表示如果新蜜源 v_i的适应度值优于原蜜源 x_i的适应度值,则用 v_i替换 x_i,否则保留 x_i.

2.2.2 观察蜂阶段

观察蜂等待雇佣蜂完成蜜源的搜索后,根据雇佣蜂提供的蜜源信息,采用轮赌选择法以概率 P_i选择蜜源,概率 P_i的计算如式(13)所示

$$P_i = fit_i / \sum_{i=1}^{N} fit_i \tag{13}$$

从式(13)可以看出,适应度值越大,P_i越大,故在轮赌选择法选择蜜源时,更优的蜜源以较大概率被选中,在被选择的蜜源附近采用式(10)进一步搜索.

2.2.3 侦查蜂阶段

在 *limit*(阈值)次开采之后,某一蜜源的花蜜量仍然没有改变,说明此时算法陷入局部最优,则该蜜源被放弃,对应的雇佣蜂转化为侦查蜂,侦查蜂的作用是防止算法陷入局部最优,用式(14)随机产生新的蜜源位置.

$$x_{id} = a_d + rand * (b_d - a_d) \tag{14}$$

其中,*rand* 是区间[0,1]之间的均匀分布随机数,a_d和 b_d分别表示第 d 维搜索空间的下边界和上边界. ABC 的伪代码见算法 1.

算法 1 人工蜂群算法(ABC)

```
Step1 初始化蜜源数 N、最大迭代次数 MaxDT 和阈值 limit
Step2 随机初始化蜜源的位置,置当前迭代次数 t = 0
Step3 计算初始蜜源的目标函数值 f_i 和适应度值 fit_i
Step4 While t < MaxDT do
  Step4.1 雇佣蜂阶段
    For i = 1 to N/2 do
      根据式(10)在蜜源附近产生新的蜜源
      计算目标函数值,根据式(11)计算新蜜源的适应度值
      根据式(12)更新蜜源,并更新 trial(i)
    End For
  Step4.2 观察蜂阶段
    根据式(13)计算概率 P_j
    While j < N/2 do
      依据 P_j 采用轮赌选择法选择一个蜜源
      根据式(10)在此蜜源附近产生新的蜜源
      计算目标函数值,并依式(11)计算蜜源的适应度值
      根据式(12)更新蜜源,并更新 trial(j)
      j = j + 1
    End While
  Step4.3 侦查蜂阶段
    If trial(i) < limit
      根据式(14)随机产生新的蜜源位置
    End If
  Step4.4 t = t + 1
End While
```

算法 1 中,*trial*(i)表示第 i 个蜜源保存的试验失败次数,*limit* 为试验失败次数阈值. 从算法 1 可知:①式(10)中的随机选择一维和随机选择一个蜜源的方式和一个随机因子构成的单维搜索产生的新蜜源随机分布在整个搜索空间,具有很强的探索能力;②观察蜂阶段是在雇佣蜂阶段的基础上对较优的蜜源进行进一步的搜索,产生更优的蜜源,体现了一定的开采能力;③侦查蜂阶段采用式(14)产生新的蜜源位置,使算法能够跳出局部最优;④由于采用贪心算法每次获得的更好蜜源又被放到当前蜜源群组中,在雇佣蜂和观察蜂阶段可能再一次被选择来产生新蜜源,进一步获得更好的蜜源,如此构成动态贪心算法,故其能够加快算法的收敛速度;⑤在雇佣蜂和观察蜂阶段采用(随机)单维的搜索方式更加强化全局搜索能力[14].

3 HGWOA 算法

3.1 GWO 与 ABC 的优势与不足

根据第 2 节内容可知,GWO 具有开采能力强、收敛速度快和占用存储空间少等优点,但灰狼群根据式(3)~(9)进行位置更新,α、β 和 δ 狼的位置确定之后,所有的狼群都向最优解逼近,多样性逐渐降低,致使全局搜索能力趋弱等. 在 ABC 中,雇佣蜂和观察蜂阶段都采用式(10)单维操作获得新蜜源,具有较强的全局搜索能力,但搜索效率低下. 虽然采用动态贪心算法和在观察蜂阶段采用轮赌选择法能够强化开采能力,但这种开采能力与较强的探索能力未能达到平衡,从而导致整体优化性能受到了限制,且动态贪心算法有时容易使算法陷入局部最优和不稳定;轮赌选择法虽然保证适

应度值高的蜜源容易被选中,但仍然存在许多不足[15].另外,侦查蜂阶段可以降低算法陷入局部最优的概率,但在搜索后期会降低收敛速度.

针对 GWO 和 ABC 存在的问题,并依据探索与开采必须平衡的现代智能优化思想,提出了 HGWOA. 其主要工作及特色是:

(1)为了提高算法的稳定性和探索能力,摒弃雇佣蜂阶段的动态贪心算法,改用静态贪心算法;为弥补静态贪心算法收敛速度慢的不足,采用局部搜索能力较强的新型搜索蜜源方式,在雇佣蜂阶段和观察蜂阶段分别采用不同的新型搜索方式,在强调全局搜索能力的同时,也注重收敛性能.

(2)去掉影响收敛速度的侦查蜂阶段,在雇佣蜂阶段添加反向学习策略,弥补去掉侦查蜂阶段的不足,避免搜索陷入局部最优.

(3)将局部搜索能力强的 GWO 融入到观察蜂阶段,更加提升局部搜索能力和解的精度;另外也发挥 ABC 和 GWO 二者优势,增强优化的普适性.

(4)为了提高优化效率,在观察蜂阶段,丢弃效率低下的轮赌选择法,改用新型的榜样学习选择方法,这种方法不仅计算复杂度低,而且能确保选择的解一定不比当前解差,使得搜索向着优质解趋近,以提高开采能力.

3.2　新型搜索方式:两种搜索方式融合

3.2.1　雇佣蜂阶段搜索方式

由第 2.2 节可知,在雇佣蜂阶段采用式(10)的搜索方式,体现了较强的探索能力. 为了平衡这种探索能力,ABC 采用动态贪心算法来加快收敛速度,但这种动态贪心算法会降低算法的稳定性和易造成算法早熟. 因此本文摒弃原 ABC 的动态贪心算法,采用静态贪心算法. 为弥补收敛速度降低的不足,在雇佣蜂阶段提出新型的搜索蜜源方式,在强调探索能力的同时,也注重收敛质量. 新的搜索蜜源方式如式(15).

$$v_{id}=x_{id}+\varphi_{id}(x_{id}-x_{kd})+rand(x_{gd}-x_{id}) \tag{15}$$

其中,x_{gd}表示全局最优蜜源第 d 维的值. 式(15)在式(10)的基础上添加了一项用于强化局部搜索,以提高算法的开采能力.

雇佣蜂以等概率的方式选择式(10)和式(15)两种搜索方式,保证探索,也强化开采.

3.2.2　观察蜂阶段搜索方式

由第 2.2 节可知,观察蜂阶段利用轮赌选择法不断选择优质蜜源,采用式(10)在蜜源附近进一步产生新的蜜源,体现了一定的局部搜索能力. 但由于采用单维操作,局部搜索能力受限,为提高局部搜索能力,提出了一种新的搜索方式,如式(16)所示.

$$v_{id}=x_{gd}+\varphi_{id}(x_{gd}-x_{id}) \tag{16}$$

式(16)使观察蜂在当前维的全局最优蜜源附近产生新的蜜源,观察蜂趋近于最优蜜源,加快了收敛速度,强化了局部搜索能力.

观察蜂以等概率的方式选择式(10)和式(16)两种搜索方式,同样保证探索,也强化开采.

3.3　反向学习雇佣蜂阶段

为弥补去掉侦查蜂阶段的不足,在雇佣蜂阶段添加反向学习策略. 反向学习(Opposition Learning)是由 Tizhoosh 等人[16]在 2005 年提出的,其目的在于提高算法的全局搜索能力,避免陷入局部最优,这种策略也被成功用于多种算法中[16,17].

利用反向学习产生新的蜜源,如式(17).

$$x_{id}=a_d+(b_d-x_{id}) \tag{17}$$

反向学习雇佣蜂阶段的伪代码见算法 2.

算法 2　反向学习雇佣蜂阶段

```
在 1 到 N/2 中随机选取一个蜜源 num
for i = 1 to N/2 do
  if i 不等于 num
    3.2.1 节的搜索方式产生新的蜜源
  else
    执行反向学习策略产生新的蜜源
  end if
end for
计算蜜源的适应度值
用式(12)的贪心算法更新蜜源,并更新 trial(i)
```

从算法 2 看出:①在雇佣蜂阶段,反向学习策略与 3.2.1 节的新型搜索方式有机融合,避免算法陷于局部最优;②反向学习策略在每次迭代仅用于随机选择的一个蜜源上,既可保证种群多样性,又不用设置用于反向学习的蜜源数,提高了算法可操作性;③式(12)贪心算法用在 for 循环之外,即每个蜜源的更新不会影响当前种群产生的新蜜源,这种贪心算法为静态贪心算法,可以提升算法的稳定性,但可能会降低收敛速度.

3.4　GWO 观察蜂阶段

为了进一步提高算法的收敛速度、优化效率和求解精度,将局部搜索能力强的 GWO 搜索方式融入到观察蜂阶段. 具体操作过程为:在采蜜过程中,若蜜源经过 *limit* 次迭代其适应度值一直不变,则采用 GWO 搜索方式,以便提高整个算法的开采能力. GWO 观察蜂阶段的伪代码如算法 3.

算法 3　GWO 观察蜂阶段

```
for i = 1 to N/2 do
    榜样学习选择一个蜜源 i
  if trial(i) < limit
```

```
    按照3.2.2节所述产生新的蜜源
  else
    按照式(3)~(9)产生新蜜源
  end if
  计算蜜源的适应度值
  根据式(12)的贪心算法更新蜜源,并更新 trial(i)
end for
```

若试验失败次数 $trial(i)$ 达到了所设置的阈值 $limit$ 时,则采用 GWO 的搜索产生新的蜜源,即执行式(3)~(9). 在观察蜂阶段,还采用榜样学习选择蜜源的方式,这种方式和 GWO 都有利于提高算法的开采能力.

3.5 榜样学习策略选择蜜源

在观察蜂阶段,采用榜样学习策略替代轮赌选择法. 其原因如下:①在观察蜂阶段,使用轮赌选择法选择待更新蜜源,根据随机概率选择蜜源虽然能够保证以较大的概率使优秀的蜜源更容易被选中,但选择失败经常发生. 实验证明选择失败的次数约为观察蜂个数的 10 倍,这无疑会耗费计算机资源[15],故轮赌选择法的效率较低;②榜样学习选择方法不仅计算复杂度低,而且能确保选择的蜜源一定不比当前蜜源差,提高了局部搜索能力. 榜样学习策略的详细介绍见文献[18]. 榜样学习策略选择蜜源的过程是:首先按照蜜源的花蜜量由大到小排序,排在前面的蜜源是排在后面蜜源的榜样,然后为蜜源 i 随机获取一个榜样,如式(18)所示

$$num = ceil(i * rand) \tag{18}$$

其中 $ceil(\,)$ 为取整函数. 榜样学习选择蜜源的方式只对蜜源的花蜜量进行排序,不对整个蜜源进行排序,且不需要计算式(11)(12),相比于轮赌选择法,不仅降低了计算复杂度,而且榜样学习策略选择蜜源的方式使得新蜜源趋近于更优蜜源,加快了算法的收敛速度,提高了局部搜索能力,同时提高了搜索精度.

综上所述,采用单维搜索、式(10)搜索方式、反向学习策略以及静态贪心算法都是为了提高探索能力;而为了平衡这种探索能力,主要采用开采能力强的 GWO 搜索方法、动态贪心算法和榜样学习策略以及添加式(15)(16)的搜索方式,使得探索与开采二者达到很好的平衡,以便提高整体的优化性能. 另外,将 ABC 单维操作与 GWO 全维操作、它们的两类搜索机制、GWO 开采与 ABC 探索、静态贪心法与动态贪心法等互补,以便获得较强的普适性等.

3.6 HGWOA 算法的基本流程

第 3.2 至 3.5 节的改进融合,构成了本文提出的 HGWOA,其具体实现步骤如下:

步骤 1 参数初始化,随机初始化蜜源的位置并计算适应度值.

步骤 2 反向学习雇佣蜂阶段见算法 2.

步骤 3 GWO 观察蜂阶段见算法 3.

步骤 4 未满足终止条件时,重复迭代步骤 2~3,否则迭代停止,输出最优解.

HGWOA 的流程图如图 2 所示.

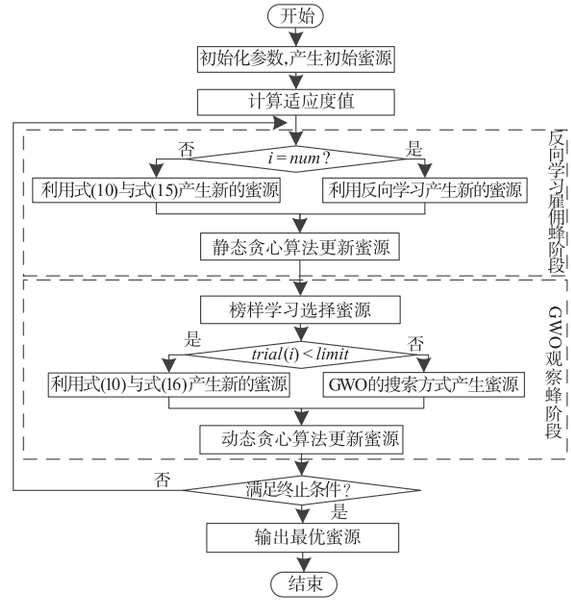

图 2 HGWOA的流程图

4 函数优化实验结果与分析

为验证 HGWOA 的有效性,采用 25 个不同类型的 Benchmark 函数在不同维度上进行优化测试,并将 HGWOA 与 7 个对比算法进行性能比较. 在 25 个函数中,前 23 个函数是许多有关优化算法文献普遍采用的基准函数,详细情况见文献[19]. f_1 ~ f_7 是高维单峰函数. f_8 ~ f_{13} 是高维多峰函数,其中局部最小点的数量随着维数呈指数增长;f_{14} ~ f_{25} 是低维多峰函数. 所有这些函数用于全面地测试 HGWOA 的优化性能. 另外还增加了两个 2 维函数 f_{24} 和 f_{25},其细节如下

$$f_{24}(x) = -21.5 - x_1\sin(4\pi x_1) - x_2\sin(20\pi x_2) \tag{19}$$

其中,$-3 \leqslant x_1 \leqslant 12.1$,$4.1 \leqslant x_2 \leqslant 5.8$,全局最优值为 -38.8503.

$$f_{25}(x) = 100\sin(5x_1) - 300\sin(2x_2) + 4.5x_1 - 4x_2 + x_1^2 + 2x_2^2 - 2x_1x_2 + 4x_1^4 - 2x_1^2x_2 \tag{20}$$

其中,$-5 \leqslant x_i \leqslant 5$,全局最优值为 -398.6720.

4.1 对比算法及参数设置

本文选取 7 个具有代表性的对比算法与 HGWOA 进行对比实验,它们分别是 ABC[4]、GWO[3]、MEABC[20]、GWOepd[21]、EPSDE[22]、CCS[23] 和 HCLPSO[24]. MEABC(Multi-strategy Ensemble ABC)是一种多策略集

成的 ABC,其中,一组不同的搜索策略在整个搜索过程中共存并竞争产生后代,更好地平衡了 ABC 的探索与开采能力. GWOepd(Evolutionary Population Dynamics GWO)是 Saremi 等人提出的一种融合 EPD 的 GWO,它借助动态进化种群来提高 GWO 的探索能力. EPSDE(Ensemble of Parameters and mutation Strategies in DE)是 Mallipeddi 等人提出的一种融合控制参数和突变策略的差分进化算法. CCS(Chaotic Cuckoo Search)是 Wang 等人提出的一种混沌布谷鸟搜索算法,它将混沌搜索和精英策略融入 CS 中,全面提高了 CS 的搜索能力. HCLPSO(Heterogeneous Comprehensive Learning PSO)是 Nandar 等人提出的一种异构综合学习策略的 PSO 算法,其中,综合学习策略用于生成两个子群的样本,整体上提高了算法的探索与开采能力. ABC 和 GWO 属于经典算法,MEABC 和 GWOepd 属于 ABC 和 GWO 的改进算法.

为了公平起见,8 个算法在独立运行次数和低维函数上的参数设置采用了目前大多数相关文献的标准设置[19],即独立运行次数 $Num = 30$;对于低维函数,$N = 100$,MaxDT = 100(函数 f_{15} 和 f_{20} 的 MaxDT = 200). 但对于高维函数($f_1 \sim f_{13}$),采用了更严苛的最大函数评价次数(MaxFEs),以便更凸显 HGWOA 更强的搜索能力,即种群大小 $N = 40$,MaxDT = 2000,其 MaxFEs 小于文献[19]中推荐的最小 MaxFEs,即测试条件更加严苛. 另外,对于 HGWOA,仅有一个 *limit* 参数需要调整,依据多次试验结果,*limit* 设为 18 得到最好的优化性能. 对于其他算法的参数设置见相应的文献. 为更好地分析算法的性能,采用 Wilcoxon 符号秩和检验[25],它是一种统计性检验方法,可以检验出两个算法之间存在的显著性差异,本文的显著性水平设置为 0.05. 实验环境采用 Windows 7 操作系统,3.10GHz CPU 和 4GB 内存的 PC,编程语言为 MATLAB R2014a.

4.2　优化性能对比

表 1 给出了 8 个算法在 25 个 Benchmark 函数上的实验结果,其中包含均值(Mean)、方差(Std)和符号(Sign),Mean 表示算法获得的全局最优值与目标函数的理论最优值之间的偏差(误差)均值. 其中最优者用粗体表示,"+/-/="分别表示 HGWOA 优于、劣于、等于对比算法的个数.

与 ABC 和 GWO 比较. 由表 1 可知,在 7 个单峰函数上,HGWOA 获得 6 次最好值,其中在函数 f_1、f_2、f_4 和 f_6 上 HGWOA 取得了理论最优值. 在函数 f_3 上 HGWOA 虽未获得最好值,但其仅次于 GWO. 整体上来说,在获得最好值的单峰函数个数上,HGWOA 要多于 ABC 和 GWO,说明 HGWOA 的优化性能要优于 ABC 和 GWO. 从均值来看,GWO 优于 ABC 的个数为 5,说明 ABC 在单峰函数上的搜索精度不及 GWO,即在局部搜索能力上,GWO 强于 ABC. 在 18 个多峰函数上,HGWOA 在 3 个算法中获得了 17 次最好值,其中在函数 f_9、f_{11}、f_{16} 和 f_{19} 上,HGWOA 取得了理论最优值,体现了 HGWOA 较强的全局搜索能力;而在函数 f_{10} 上 HGWOA 虽未获得最好值,其优化性能仅次于 GWO. 整体上来说,HGWOA 在多峰函数上的优化性能优于 ABC 和 GWO,表明 HGWOA 具有较强的全局搜索能力,说明反向学习和静态贪心算法等策略的使用是有效的. 从均值来看,ABC 优于 GWO 的个数为 15,说明 ABC 在多峰函数上,尤其在低维多峰函数上,大幅度优于 GWO,这说明在全局搜索能力上,ABC 强于 GWO. 综上所述,GWO 具有较强的局部搜索能力,ABC 具有较强的全局搜索能力,而 HGWOA 的优化性能大幅度优于 GWO 和 ABC,这是由于 HGWOA 是 GWO 和 ABC 的融合改进结果,实现了二者优势互补. 多数情况下 HGWOA 是三个算法中方差最小的,说明其稳定性较好.

表 1　8 个对比算法在 25 个 benchmark 函数上的优化结果

Fun	Opt	HGWOA	ABC	GWO	MEABC	GWOepd	EPSDE	CCS	HCLPSO
f_1	Mean	0	1.6903e-28	5.3890e-134	5.1118e-25	8.0892e-106	5.3373e-52	6.8758e-10	4.0911e-15
	Std	0	4.0079e-28	2.2223e-133	3.3859e-25	1.5326e-105	2.0325e-51	6.8967e-10	6.7365e-15
	Sign		+	+	+	+	+	+	+
f_2	Mean	0	4.2187e-15	1.0736e-77	1.0503e-13	5.6467e-61	2.6796e-26	4.6105e-05	5.7301e-08
	Std	0	3.5368e-15	2.3552e-77	3.7561e-14	1.2610e-60	3.8330e-26	2.1853e-05	2.0024e-07
	Sign		+	+	+	+	+	+	+
f_3	Mean	3.6891e-06	1.1002e+04	6.6871e-40	1.1634e+04	5.0722e-24	1.0347e-03	8.2942e+00	1.5977e+01
	Std	1.9988e-05	1.6518e+03	2.2751e-39	1.5872e+03	2.2300e-23	3.1653e-03	4.6186e+00	7.3411e+00
	Sign		+	-	+	-	+	+	+

续表

Fun	Opt	HGWOA	ABC	GWO	MEABC	GWOepd	EPSDE	CCS	HCLPSO
f_4	Mean	0	1.3363e+01	2.7789e-33	1.0564e+01	1.3935e-24	4.3397e+00	2.5818e-01	4.6999e-01
	Std	0	1.4242e+00	4.0705e-33	1.4032e+00	6.9647e-24	2.1522e+00	1.5050e-01	2.2101e-01
	Sign		+	+	+	+	+	+	+
f_5	Mean	1.1041e+00	1.7827e+00	2.6599e+01	9.7730e-01	2.6874e+01	3.5830e+00	2.7369e+01	4.8992e+01
	Std	4.5089e+00	1.5229e+00	7.3617e-01	1.0460e+00	9.6165e-01	2.4379e+00	1.5357e+01	3.0056e+01
	Sign		+	+	-	+	+	+	+
f_6	Mean	0	0	0	0	0	1.0000e-01	1.0000e-01	0
	Std	0	0	0	0	0	3.0513e-01	3.0513e-01	0
	Sign		=	=	=	=	+	+	=
f_7	Mean	7.9374e-04	8.2230e-02	8.9025e-04	5.0939e-02	1.2421e-03	3.4035e-03	2.6544e-02	8.8507e-03
	Std	1.0498e-03	1.5155e-02	5.3694e-04	9.0092e-03	5.8723e-04	1.2182e-03	9.1599e-03	3.3937e-03
	Sign		+	+	+	+	+	+	+
f_8	Mean	8.7918e-12	1.8311e-11	6.2417e+03	9.0343e-12	7.0634e+03	7.2760e-12	2.9444e+03	1.4701e+03
	Std	8.3880e-13	2.4876e-11	7.0746e+02	3.3210e-13	1.3336e+03	0	4.1165e+02	2.1354e+02
	Sign		+	+	+	+	-	+	+
f_9	Mean	0	3.1959e-07	2.2600e-01	0	0	5.0330e-15	1.0221e+02	1.0097e+01
	Std	0	6.9428e-07	1.2379e+00	0	0	1.3988e-14	2.1302e+01	3.4189e+00
	Sign		+	+	=	=	+	+	+
f_{10}	Mean	1.9125e-14	3.9713e-13	7.0462e-15	1.1999e-12	8.1120e-15	5.3883e-15	1.3796e-01	2.2098e-08
	Std	1.5230e-14	1.5759e-13	2.2242e-15	5.2941e-13	2.9108e-15	1.5283e-15	3.6331e-01	1.6404e-08
	Sign		+	-	+	-	-	+	+
f_{11}	Mean	0	3.2888e-04	0	2.0724e-16	2.0268e-03	4.9307e-04	3.4889e-04	8.2812e-03
	Std	0	1.8014e-03	0	7.9726e-16	5.4934e-03	1.8764e-03	1.8180e-03	1.3312e-02
	Sign		+	=	+	+	+	+	+
f_{12}	Mean	1.5705e-32	2.0622e-28	2.5769e-02	5.4001e-27	1.6913e-02	1.5705e-32	7.2672e-01	9.3739e-17
	Std	5.5674e-48	8.7683e-28	1.2795e-02	4.2493e-27	1.0994e-02	5.5674e-48	8.0338e-01	1.0891e-16
	Sign		+	+	+	+	=	+	+
f_{13}	Mean	1.3498e-32	1.6764e-28	4.3893e-01	1.5255e-25	4.3372e-01	1.3621e-32	4.7971e-09	3.1657e-15
	Std	5.5674e-48	3.7249e-28	1.1553e-01	1.4792e-25	2.7637e-01	3.7610e-34	6.3113e-09	4.6217e-15
	Sign		+	+	+	+	+	+	+
f_{14}	Mean	4.4409e-17	8.7415e-14	2.0598e+00	0	5.2074e-10	4.4409e-16	7.6352e-10	1.0233e-12
	Std	9.0336e-17	3.6153e-13	3.4203e+00	0	3.3391e-10	1.8440e-16	3.1033e-09	1.1304e-12
	Sign		+	+	-	+	+	+	+
f_{15}	Mean	2.8644e-04	6.0248e-04	1.4441e-03	3.6955e-04	1.9856e-04	1.8774e-07	3.4995e-04	2.1159e-04
	Std	1.7496e-04	2.2776e-04	5.0654e-03	9.9627e-05	1.5677e-04	4.3034e-07	9.6807e-05	1.1593e-04
	Sign		+	+	+	-	-	+	-
f_{16}	Mean	0	1.1450e-14	4.7964e-08	2.8503e-14	4.3614e-07	9.9270e-10	9.3365e-08	6.6758e-09
	Std	0	1.6155e-14	4.6321e-08	1.5612e-13	1.0248e-06	5.4314e-09	1.1536e-07	2.6721e-08
	Sign		+	+	+	+	+	+	+

续表

Fun	Opt	HGWOA	ABC	GWO	MEABC	GWOepd	EPSDE	CCS	HCLPSO
f_{17}	Mean	1.1842e-16	1.7637e-07	4.0686e-03	2.9563e-07	4.0912e-04	6.4555e-08	4.4997e-07	4.2973e-07
	Std	4.5068e-16	3.8914e-07	1.8926e-02	1.0167e-06	1.5571e-03	2.1783e-07	6.6898e-07	2.2893e-06
	Sign		+	+	+	+	+	+	+
f_{18}	Mean	1.6283e-16	8.5785e-04	1.0345e-04	2.5812e-11	3.2759e-05	1.2760e-14	1.2443e-06	5.7773e-10
	Std	4.1205e-16	2.3812e-03	1.2682e-04	7.3988e-11	3.8895e-05	1.1195e-14	2.5095e-06	6.4825e-10
	Sign		+	+	+	+	+	+	+
f_{19}	Mean	0	1.1450e-10	1.1640e-03	0	5.1138e-05	5.7139e-15	8.6544e-07	7.7012e-10
	Std	0	4.2295e-10	2.2354e-03	0	8.1821e-05	5.3846e-15	9.3630e-07	6.8597e-10
	Sign		+	+	=	+	+	+	+
f_{20}	Mean	3.7007e-16	1.0713e-10	7.8561e-02	1.2782e-09	2.0494e-02	1.0857e-05	8.8310e-04	3.1873e-04
	Std	1.6833e-16	3.0008e-10	8.0580e-02	4.1600e-09	4.6556e-02	5.4803e-05	6.0679e-04	1.5364e-03
	Sign		+	+	+	+	+	+	+
f_{21}	Mean	2.3685e-16	2.7818e-04	1.4759e+00	7.4459e-02	9.3674e-01	2.0891e-02	5.0613e-02	1.6868e-01
	Std	6.1417e-16	7.4149e-04	2.5616e+00	2.0566e-01	2.1568e+00	6.3180e-02	5.2521e-02	9.2241e-01
	Sign		+	+	+	+	+	+	+
f_{22}	Mean	8.2897e-16	4.2968e-04	3.4876e-03	1.4938e-03	1.8352e-01	2.2033e-04	6.3729e-02	4.7711e-04
	Std	1.0149e-15	8.6897e-04	1.8502e-03	5.8161e-03	9.6264e-01	8.4565e-04	7.7935e-02	2.4490e-03
	Sign		+	+	+	+	+	+	+
f_{23}	Mean	9.4739e-16	2.7928e-04	3.5061e-03	5.1800e-03	7.9236e-03	3.6388e-04	1.0188e-01	1.7875e-01
	Std	1.0149e-15	5.4629e-04	2.2032e-03	1.1911e-02	4.2938e-03	1.9057e-03	7.6405e-02	9.7888e-01
	Sign		+	+	+	+	+	+	+
f_{24}	Mean	4.7370e-16	1.8123e-04	1.0062e-01	0	1.1150e-01	2.2533e-08	7.7512e-02	7.2776e-02
	Std	1.8027e-15	8.9361e-04	3.3366e-02	0	3.6809e-02	1.1481e-07	4.7495e-02	1.0259e-01
	Sign		+	+	-	+	+	+	+
f_{25}	Mean	3.0316e-14	2.8726e-07	5.3747e-01	1.5015e-02	3.1579e-01	1.7844e-09	2.0096e-02	2.7859e-02
	Std	3.2477e-14	1.1810e-06	6.7670e-01	3.0544e-02	5.3159e-01	4.4481e-09	2.5758e-02	3.6608e-02
	Sign		+	+	+	+	+	+	+
+/-/=			24/0/1	21/2/2	19/3/3	20/3/2	21/3/1	25/0/0	23/1/1

与 ABC 和 GWO 的改进算法比较. HGWOA 优于、劣于和等于 MEABC 的个数分别是 19、3 和 3,说明 HGWOA 在 19 个函数上明显优于 MEABC. HGWOA 优于、劣于和等于 GWOepd 的个数分别是 20、3 和 2,说明 HGWOA 在 20 个函数上明显优于 GWOepd.

与目前最优秀的其它优化算法比较. HGWOA 优于、劣于和等于 EPSDE 的个数分别是 21、3 和 1,说明 HGWOA 在 21 个函数上明显优于 EPSDE. HGWOA 优于 CCS 的个数是 25,说明 HGWOA 在 25 个函数上都明显优于 CCS. HGWOA 优于、劣于和等于 HCLPSO 的个数分别是 23、1 和 1,说明 HGWOA 在 23 个函数上明显优于 HCLPSO.

为直观地查看 HGWOA 与 7 个对比算法的优化性能,本文对 8 个算法勾画出性能简图如图 3 所示. 图中 τ 值采用的是以 2 为底的对数. 性能简图的详细介绍参见文献[26].

从图 3 可知,HGWOA 的曲线一直在 7 个对比算法的上方,说明 HGWOA 的优化效果最好,与 7 个对比算法相比,获得了更多的优势. $\tau = 0$ 时,HGWOA 在约 95% 的函数上表现最好;τ 值约为 1.5 至 5 时,HGWOA 可以解决所有的函数优化问题;τ 值约为 5 至 8 时,不仅 HGWOA,而且 EPSDE 和 MEABC 都可以解决所有的函数优化问题. GWO 在最开始时优于 CCS,可以解决大约 40% 的问题,但 τ 值约为 2.5 至 5.5 时 GWO 与 CCS

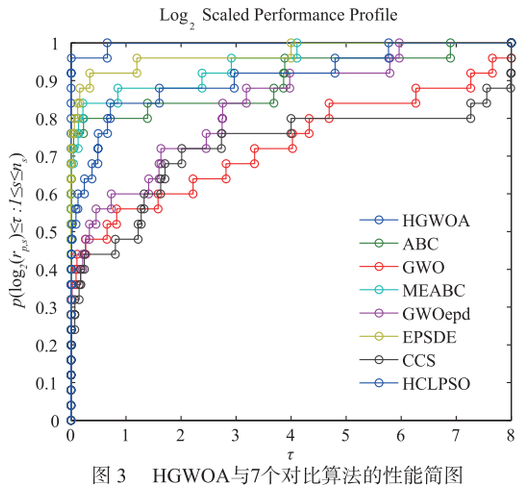

图 3 HGWOA与7个对比算法的性能简图

交叉缠绕,说明此时两个算法之间的优化效果较为相近,但随着 τ 值的不断增大,GWO 的曲线最终稍优于 CCS,说明其优化效果整体上优于 CCS. 其他算法的优化性能可做同样分析. 图 3 显示 8 个算法的整体排名顺序为 HGWOA、EPSDE、MEABC、ABC、HCLPSO、GWOepd、GWO 和 CCS.

综上所述,与 7 个对比算法相比,在 25 个不同维、不同复杂性和不同类别等的函数优化问题上,HGWOA 表现出杰出的优化性能和普适性,从整体上大幅度优于当前最优秀的智能优化算法. 从 Std 上可以看出,HGWOA 在大多数函数上获得的 Std 值是 8 个算法中最小的,说明具有极强的稳定性.

4.3 稳定性分析

对 HGWOA 的稳定性分析,主要包括一阶和二阶稳定性分析[27],以证明搜索最终收敛到一个固定点,即最优蜜源位置,且其方差收敛到零. 一阶稳定性用于保证均值的收敛性,二阶稳定性用于保证方差的收敛性[27]. 本节将 x_{gd} 设为不变条件,即这个值在一次迭代中不进行更新. 从第 3 节中可知,在 HGWOA 中采用了新型搜索方式和 GWO 搜索方式. 因篇幅所限,仅对观察蜂阶段的新型搜索方式(式(16))的稳定性进行分析.

式(16)重写为

$$v_{id}=(1+\varphi_{id})x_{gd}-\varphi_{id}x_{id} \tag{21}$$

可简化为

$$y_{t+1}=(1+\varphi)A-\varphi y_t \tag{22}$$

其中,$y_{t+1}=v_{id}$,$\varphi=\varphi_{id}$,$y_t=x_{id}$,$A=x_{gd}$.

定理 1 y_t的期望序列收敛,且收敛到 A.

证明 对其等式两边求期望可得

$$E(y_{t+1})=(1+\mu_\varphi)A-\mu_\varphi E(y_t) \tag{23}$$

其中,μ_φ 表示 φ 的期望值,φ 是[-1,1]之间均匀分布的随机数,因此 $\mu_\varphi=0$,得到式(24)

$$E(y_{t+1})=A \tag{24}$$

式(24)表明该搜索方式的期望值为常数,说明其总是收敛到 A. 证毕.

$$\begin{aligned}y_{t+1}^2&=[(1+\varphi)A-\varphi y_t]^2\\&=(1+\varphi^2+2\varphi)A^2+\varphi^2y_t^2-2A(\varphi+\varphi^2)y_t\end{aligned} \tag{25}$$

式(25)的期望值是

$$\begin{aligned}E(y_{t+1}^2)&=E[(1+\varphi^2+2\varphi)A^2+\varphi^2y_t^2\\&\quad-2A(\varphi+\varphi^2)y_t]\\&=[1+E(\varphi^2)+2\mu_\varphi]A^2+E(\varphi^2)E(y_t^2)\\&\quad-2A[\mu_\varphi+E(\varphi^2)]E(y_t)\end{aligned} \tag{26}$$

由于 φ 是[-1,1]之间均匀分布的随机数,因此

$$E(\varphi^2)=1/3 \tag{27}$$

$$E(y_{t+1}^2)=\frac{1}{3}E(y_t^2)-\frac{2}{3}AE(y_t)+\frac{4}{3}A^2 \tag{28}$$

根据式(23),可以得到

$$E^2(y_{t+1})=A^2 \tag{29}$$

定理 2 方差 $D(y_t)$序列收敛,且收敛到 0.

证明 根据式(28)和式(29)得到

$$\begin{aligned}D(y_{t+1})&=E(y_{t+1}^2)-E^2(y_{t+1})\\&=\frac{1}{3}E(y_t^2)-\frac{2}{3}AE(y_t)+\frac{4}{3}A^2-A^2\\&=\frac{1}{3}E[(y_t-A)^2]\end{aligned} \tag{30}$$

因为

$$y_{t+1}-A=\varphi A-\varphi y_t \tag{31}$$

可得

$$E((y_t-A)^2)=\frac{1}{3}E((y_{t-1}-A)) \tag{32}$$

递推关系可表示为

$$D(y_t)=\frac{1}{3^t}E((y_0-A)^2) \tag{33}$$

证明 $D(y_t)$是收敛的且收敛到 0,意味着当 $t\to\infty$ 时,$D(y_0)=0$.

$$\begin{aligned}\lim_{t\to\infty}D(y_t)&=\lim_{t\to\infty}\left\{\frac{1}{3^t}E[(y_0-A)^2]\right\}\\&=E[(y_0-A)^2]\lim_{t\to\infty}\frac{1}{3^t}=0\end{aligned} \tag{34}$$

证毕.

综上所述,观察蜂阶段的新型搜索方式均满足一阶稳定性和二阶稳定性,说明其均值和方差均是收敛的,该算法的稳定性较好.

4.4 运行时间对比

图 4 给出了 8 个算法独立运行 30 次的平均耗时图. 篇幅所限,在此仅展示 30 维单峰和多峰函数的平均耗时,时间单位为秒(s).

从图 4 可知,HGWOA 的平均耗时为 1.0188s,是 8 个算法中的最小值,分别大约是 ABC(1.4292s)和 GWO

(1.0840s)平均耗时的 71% 和 94%,约为 MEABC(1.3113s)、GWOepd (1.4164s)、EPSDE (2.4282s)、CCS(1.1156s)和 HCLPSO(1.8661s)平均耗时的 78%、72%、42%、91%和55%.故 HGWOA 的运行速度比7个对比算法的运行速度快.主要原因(与 ABC 和 GWO 相比较)如下.

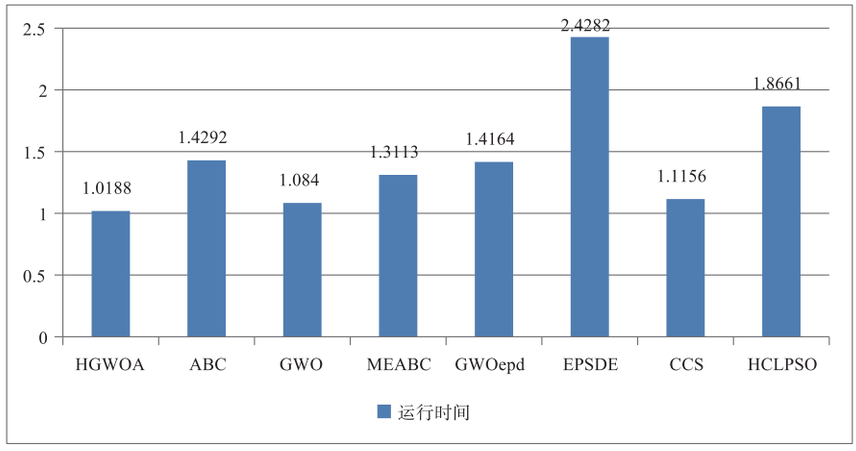

图 4　平均运行时间对比图

与 ABC 相比.在雇佣蜂阶段:①HGWOA 采用静态贪心算法,以并行计算方式计算目标函数,提高了运行速度,从而减少了运行时间;②由 3.3 节可知,HGWOA 按照等概率或者随机选择一个蜜源,采用式(10)或者式(15)或者式(17)三种方式生成新蜜源,三种方式比较,式(15)的计算复杂度比式(10)稍高.但与并行计算节省的时间相比,由式(15)计算增加的时间少.在观察蜂阶段,依据试验失败次数与设定的阈值进行比较,采用等概率计算式(10)或者式(16)或者按照 GWO 产生新解,三者相比,GWO 产生新解方式的计算复杂度大于式(10)计算复杂度.但在此阶段,将计算复杂度高的轮赌选择方法用计算复杂度低的榜样学习选择方案替代,这种降低程度大于前者的增加程度,故在观察蜂阶段,HGWOA 的计算复杂度要比 ABC 的计算复杂度低.综合以上两个阶段的讨论(以及 ABC 还有侦查蜂阶段的计算),故与 ABC 相比,HGWOA 的运行时间少.

与 GWO 相比.不管是雇佣蜂阶段还是观察蜂阶段,HGWOA 在多数情况下采用单维操作,且仅在观察蜂阶段部分嵌入 GWO,故在计算复杂度上 HGWOA 低于 GWO.

综上所述,HGWOA 有较低的计算复杂度,较快的运行速度.结合 4.2 节讨论的优化性能,HGWOA 在最少的时间内,完成的优化问题最好,说明了其具有最佳的优化效率.

4.5　收敛性分析

为检验 HGWOA 的收敛性能,对其进行收敛性分析.主要从以下两个方面比较:在高维函数上比较目标函数评价次数和在低维函数上对比收敛图.

目标函数评价次数对比.本文选取最新提出的 MTABC[28]与 HGWOA 进行对比.在高维函数上,HGWOA 的 MaxFEs = 8e+04.表 2 列出了两个算法的结果,其中 MTABC 的数据直接取自文献[28],其 MaxFEs = 1e+05,并取两个算法在相同函数、相同维度(D = 30)和相同独立运行次数上的数据对比.

表 2　HGWOA 与 MTABC 在 30 维上的 Mean 对比结果

函数	HGWOA	MTABC
	MaxFEs = 8e+04	MaxFEs = 1e+05
f_1	0	1.11e-39
f_2	0	5.90e-21
f_4	0	1.93e+00
f_5	1.10e+00	7.05e+00
f_6	0	0
f_7	7.94e-04	6.34e-03
f_9	0	0
f_{10}	1.91e-14	1.69e-14
f_{11}	0	1.05e-13
f_{12}	1.57e-32	1.57e-32
f_{13}	1.35e-32	1.35e-32

从表 2 可知,在 30 维的 MaxFEs 上,HGWOA 均小于 MTABC,说明 HGWOA 的运行条件更加苛刻.在 11 个函数上,其中在 6 个函数的均值上 HGWOA 均小于 MTABC,在 4 个函数上均值相等.整体上来说,HGWOA 在 MaxFEs 更少的前提下,仍然具有更快的收敛速度和

更优搜索精度.

收敛图对比. 限于篇幅,本文仅在低维多峰函数(f_{14} ~ f_{25})中选取 3 个函数(函数 f_{17}、f_{18} 和 f_{20})作为代表,给出其 8 个算法的收敛图如图 5.

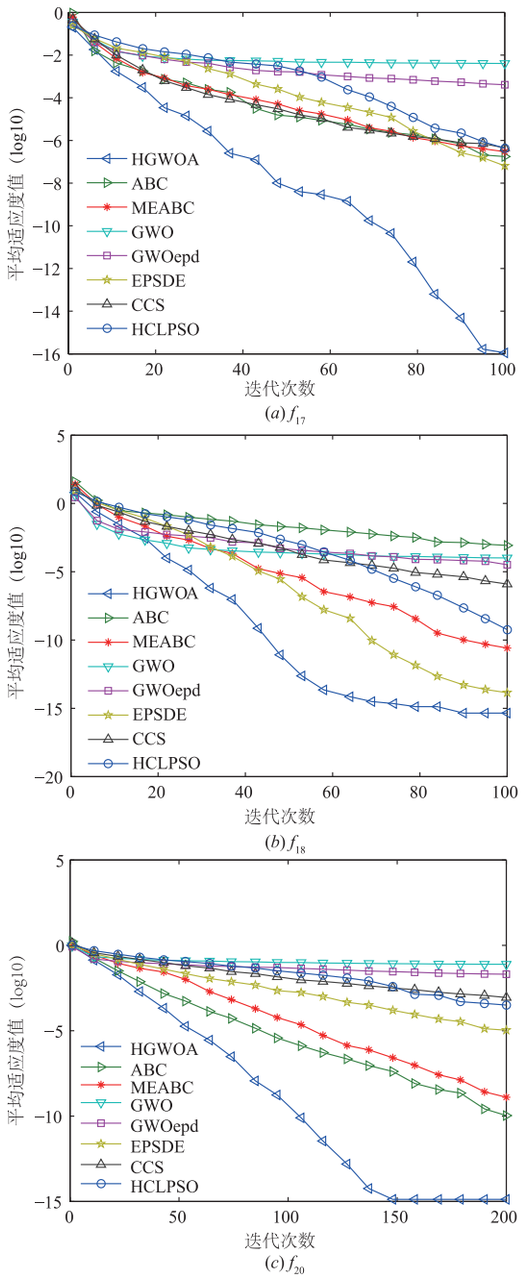

图 5 算法在低维多峰函数上的收敛图

由图 5 可知,与 7 个对比算法相比,HGWOA 具有明显的优势,具有更快的收敛速度,在收敛性能上,HGWOA 远优于 7 个对比算法.

以上两方面对比证明:改进的搜索方式和嵌入局部搜索能力强的 GWO 等使 HGWOA 具有很好的收敛性能.

5 HGWOA 在聚类优化上的应用

聚类[29]是一种无监督的机器学习任务,可应用于模式识别、图像处理等诸多领域. 它是指将数据划分成有意义或有用的组(簇),其最终目的是使得组(或群组)内的对象的相似性最大化,两个不同组中的对象的相似性最小化. 组内的相似性越大,组间的差别越大,聚类效果越好.

聚类问题可定义为求解目标函数的最小化问题,因此可以使用优化算法来解决. 采用 HGWOA 解决聚类优化问题,把每一个蜜源当作聚类优化问题的一个候选解,获得目标函数最小值的解是最优解,聚类目标函数表达式见式(35)

$$f = \sum_{j=1}^{K} \sum_{X_i \in C_i} d(X_i, Z_j) \tag{35}$$

其中,$d(X_i, Z_j)$ 表示样本点 X_i 与所属类中心 Z_j 之间的欧式距离,且 $d(X_i, Z_j) = \| X_i - Z_j \|$,$n_j$ 表示第 j 个组(簇)中的样本点个数,K 表示聚类中心个数,X_i 表示第 i 个样本点.

本文采用 6 组标准数据集验证 HGWOA 的有效性,同时选取 4 个对比算法进行对比. 6 组数据集来自于 UCI 机器学习数据库标准的数据分类集. 4 个对比算法包括 ABC 和 GWO 的改进算法,即 NSABC[14] 和 GWOepd[21],擅长于聚类优化的改进 PSO 算法(IPSO)和改进差分进化算法(IDE). 其中 IPSO 和 IDE 来源于"http://yarpiz.com/64/ypml101-evolutionary-clustering". 所有算法设置相同的参数,$N = 50$,MaxDT = 200,$Num = 30$.

表 3 给出了 5 个对比算法聚类优化的结果. Mean 表示独立运行 30 次获得的最小距离的平均值,Std 表示方差,Time 表示运行时间,单位为秒(s). 数据集名称后的括号中表示的是样本数、属性数和类数.

从表 3 可知,HGWOA 在 6 个数据集上均得到最优的 Mean 和 Std,且 HGWOA 的运行时间最少,这说明 HGWOA 的聚类优化效果大幅度优于 4 个对比算法,说明了 HGWOA 在聚类优化上的有效性.

总结 HGWOA 在函数优化和聚类优化上的表现,与对比算法相比,HGWOA 具有如下的主要优势:①优化性能好;②运行速度快;③优化效率高;④收敛质量佳;⑤稳定性强;⑥普适性好.

6 结论

鉴于 GWO 和 ABC 存在的优势和不足,本文提出了一种 GWO 与 ABC 的混合优化算法(HGWOA). 首先,对 ABC 进行改进,在雇佣蜂阶段摒弃原 ABC 的动态贪心算法,改用静态贪心算法,同时提出新型的搜索蜜源

表 3　聚类优化的结果对比

数据集		HGWOA	NSABC	GWOepd	IPSO	IDE
Wine (178,13,3)	Mean	88.7552	99.6761	90.5469	89.0906	103.6378
	Std	0.1096	1.988	4.57	2.6559	3.4068
	Time	1.1126	1.1668	1.1847	1.9129	1.7244
Heart (270,13,2)	Mean	283.7743	287.3292	283.8399	285.0073	284.9708
	Std	1.0651e-02	1.2027	2.9441e-02	5.2426	6.7765e-01
	Time	1.1279	1.1798	1.1799	1.9207	1.7576
Iris (150,4,3)	Mean	29.1487	29.5556	29.1533	29.357	30.1836
	Std	1.9456e-04	2.4784e-01	2.3717e-03	1.0048	5.2885e-01
	Time	1.0464	1.0961	1.0625	1.8194	1.6631
Glass (214,9,6)	Mean	56.1733	72.3941	62.2627	58.1619	70.4601
	Std	1.8382	2.7449	3.724	3.6794	1.5653
	Time	1.2031	1.2534	1.3052	2.0162	1.8164
Blance (625,4,3)	Mean	356.0201	356.7464	356.0487	356.0516	357.5575
	Std	1.3104e-01	1.5815e-01	1.6318e-01	1.8643e-01	2.6512e-01
	Time	1.2569	1.3073	1.2802	2.0283	1.8664
Baloon (20,4,2)	Mean	16.9458	16.9461	17.0063	17.0501	16.9541
	Std	4.7805e-05	6.9452e-04	6.7885e-02	1.1663e-01	1.7712e-02
	Time	0.97783	1.0206	0.99303	1.7509	1.5711

方式;去掉影响收敛速度的侦查蜂阶段,在雇佣蜂阶段添加反向学习策略,弥补去掉侦查蜂阶段的不足,进一步提高了探索能力. 然后,在观察蜂阶段,自适应地将局部搜索能力强的 GWO 算法融入到观察蜂阶段,提高了局部搜索能力、求解精度和优化效率. 由此发挥了二者优势,获得较强的普适性. 25 个 Benchmark 函数的优化结果表明,与 7 个典型的算法相比,HGWOA 的优化效果更好,运行速度更快,普适性更强. HGWOA 用于聚类优化的结果表明,与 4 个对比算法相比,HGWOA 拥有绝对的优势. 故 HGWOA 是有效的,可应用于其它实际的优化问题上.

参考文献

[1] 王东风,孟丽,赵文杰. 基于自适应搜索中心的骨干粒子群算法[J]. 计算机学报,2016,39(12):2652-2667.
WANG Dong-feng, MENG Li, ZHAO Wen-jie. Improved bare bones particle swarm optimization with adaptive search center[J]. Chinese Journal of Computers, 2016, 39(12):2652-2667. (in Chinese)

[2] TAO Y, LIU F, CHEN B. New particle swarm optimization algorithm with H′enon chaotic map structure[J]. Chinese Journal of Electronics, 2017, 26(4):747-753.

[3] MIRJALILI S, MIRJALILI SM, LEWIS A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69(3):46-61.

[4] DANG C T, WU Z, WANG Z, et al. A novel hybrid data clustering algorithm based on artificial bee colony algorithm and K-means[J]. Chinese Journal of Electronics, 2015, 24(4):694-701.

[5] 张新明,涂强,康强,等. 灰狼优化与差分进化的混合算法及函数优化[J]. 计算机科学,2017,44(9):93-124.
ZHANG Xin-ming, TU Qiang, KANG Qiang, et al. Hybrid optimization algorithm based on grey wolf optimization and differential evolution for function optimization[J]. Computer Science, 2017, 44(9):93-124. (in Chinese)

[6] HEIDARI A A, PAHLAVANI P. An efficient modified grey wolf optimizer with Lévy flight for optimizationtasks[J]. Applied Soft Computing, 2017, 60:115-134.

[7] 杜振鑫,刘广钟,韩德志,等. 基于全局无偏搜索策略的精英人工蜂群算法[J]. 电子学报,2018,46(2):308-314.
DU Zhen-xin, LIU Guang-zhong, HAN De-zhi, et al. Artificial bee colony algorithm with global and unbiased search strategy[J]. Acta Electronica Sinica, 2018, 46(2):308-314. (in Chinese)

[8] WOLPERT D H, MACREADY W G. No freelunch theorems for optimization[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1):67-82.

[9] KAMBOJ V K. A novel hybrid PSO-GWO approach for u-

nit commitment problem[J]. Neural Computing & Applications,2016,27(6):1643 – 1655.
[10] JAYABARATHI T, RAGHUNATHAN T, ADARSH B R,et al. Economic dispatch using hybrid grey wolf optimizer[J]. Energy,2016,111:630 – 641.
[11] LIANG Z, HU K, ZHU Q, et al. An enhanced artificial bee colony algorithm with adaptive differential operators [J]. Applied Soft Computing,2017,58:480 – 494.
[12] GOUDARZI S,WAN H H,ANISI M H,et al. ABC-PSO for vertical handover in heterogeneous wireless networks [J]. Neurocomputing,2017,256(SI):63 – 81.
[13] KEFAYAT M,ARA A L,NIAKI S A N. A hybrid of ant colony optimization and artificial bee colony algorithm for probabilistic optimal placement and sizing of distributed energy resources[J]. Energy Conversion & Management, 2015,92(3):149 – 161.
[14] SHI Y,PUN C M,HU H,et al. An improved artificial bee colony and its application[J]. Knowledge-basedSystems, 2016,107:14 – 31.
[15] 贺桂娇,周树亮,冯冬青. 求解高维复杂优化问题的改进人工蜂群算法[J]. 计算机工程与应用,2018,54(12):126 – 132.
HE Gui-jiao, ZHOU Shu-liang, FENG Dong-qing. Improved artificial bee algorithm for high dimensional complex optimization problems [J]. Computer Engineering and Applications,2018,54(12):126 – 132. (in Chinese)
[16] LIU H,XU G,DING G,et al. Integrating opposition-based learning into the evolution equation of bare-bones particle swarm optimization[J]. Soft Computing,2015,10 (10): 539 – 546.
[17] SHAHZAD F,MASOOD S,KHAN N K. Probabilistic opposition-based particle swarm optimization with velocity clamping[J]. Knowledge and Information Systems,2014, 39(3):703 – 737.
[18] 张新明,涂强,尹欣欣. 混合迁移的高效 BBO 算法及其在图像分割中的应用[J]. 计算机科学与探索,2016,10(10):1459 – 1468.
ZHANG Xin-ming, TU Qiang, YIN Xin-xin. Efficient BBO algorithm based on hybrid migration and its application to image segmentation [J]. Journal of Frontiers of Computer Science & Technology,2016,10(10):1459 – 1468. (in Chinese)
[19] YAO X,LIU Y,LIN G. Evolutionary programming made faster[J]. IEEE Transactions on Evolutionary Computation,1999,3(2):82 – 102.
[20] WANG H,WU Z,RAHNAMAYAN S,et al. Multi-strategy ensembleartificial bee colony algorithm[J]. Information Sciences,2014,279:587 – 603.
[21] SAREMI S, MIRJALILI S Z, MIRJALILI S M. Evolutionary population dynamics and grey wolf optimizer[J]. Neural Computing & Applications, 2015, 26 (5): 1257 – 1263.
[22] MALLIPEDDI R, SUGANTHAN P N, PAN Q K, et al. Differential evolution algorithm with ensemble of parameters and mutation strategies[J]. Applied Soft Computing, 2011,11(2):1679 – 1696.
[23] WANG G G, DEB S, GANDOMI AH, et al. Chaotic cuckoo search[J]. Soft Computing, 2016, 20(9): 3349 – 3362.
[24] LYNN N,SUGANTHAN P N. Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation[J]. Swarm & Evolutionary Computation,2015,24:11 – 24.
[25] DERRAC J, GARCIA 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 & Evolutionary Computation,2011,1(1):3 – 18.
[26] DOLAN E D, MORE J J. Benchmarking optimization software with performance profiles [J]. Mathematical Programming,2001,91(2):201 – 213.
[27] LIU Z G,JI X H,LIU Y X. Hybrid non-parametric particle swarm optimization and its stability analysis[J]. Expert Systems with Applications,2018,92:256 – 275.
[28] SONG X Y, YAN Q F, ZHAO M. An adaptive artificial bee colony algorithm based on objective function value information [J]. Applied Soft Computing, 2017, 55: 384 – 401.
[29] KARABOGA D, OZTURK C. A novel clustering approach:artificial bee colony (ABC) algorithm[J]. Applied Soft Computing,2011,11(1):652 – 657.

作者简介

张新明 男. 1963 年出生,湖北孝感人. 教授、硕士生导师、CCF 会员. 主要研究方向是智能优化算法、数字图像处理和模式识别等.
E-mail:xinmingzhang@126.com

王 霞 女. 1993 年出生,河南新乡人. 硕士研究生. 主要研究方向是智能优化算法和数字图像分割.
E-mail:wangxia0801@qq.com