



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

廖作文¹, 龚文引^{1*}, 王凌^{2*}

1. 中国地质大学(武汉) 计算机学院, 武汉 430074

2. 清华大学自动化系, 北京 100084

* 通信作者. E-mail: wygong@cug.edu.cn, wangling@tsinghua.edu.cn

收稿日期: 2019-07-20; 修回日期: 2019-08-11; 接受日期: 2019-08-23; 网络出版日期: 2020-02-26

国家自然科学基金(批准号: 61573324, 61873328) 和国家杰出青年基金(批准号: 61525304) 资助项目

摘要 求解非线性方程组要求在一次运行中同时求解(联解)其多个根,在数值计算中这是一项重要但困难的工作.为了实现非线性方程组多根求解,本文提出了一种改进的环拓扑混合群体智能算法,其主要特点是:(i)设计一种改进的环形拓扑结构,以弥补基于下标的相邻个体在搜索空间上不相邻的缺点,进而能更有效利用邻域信息;(ii)采用混合的群体智能方法以提升算法的搜索能力;(iii)引入个体重新初始化机制,以增强群体多样性.为了验证算法的性能,选择8个含有多个根的非线性方程组作为测试集.实验结果表明,所提出的方法不仅能在一次运行中找到多个根,而且与代表性算法对比,在找根率和成功率上有着显著优势.

关键词 非线性方程组, 群体智能, 多根联解, 人工蜂群算法, 环拓扑

1 引言

许多实际应用的问题,如物理^[1]、工程^[2]、经济学^[3]等,能够转化成非线性方程组^[4].因此研究非线性方程组的求解具有重要的意义.非线性方程组通常存在多个根,每个根具有同等重要性,因为决策者可以根据它们做出最合适的决定.近年来,通过一次运行求解非线性方程组多个根已受到越来越多的关注.然而,这在数值计算中是一项很困难的工作.

在数值计算中存在许多经典的方法,如牛顿法^[5]、迭代递归法^[6]被用于非线性方程组的求解.然而,这些方法存在着一些不足,比如过度依赖初始值、容易陷入局部最优,以及需要导数信息等.更重要的是,这些方法主要是用于求一个根,很难在一次运行中求解非线性方程组的多个根.

除了传统数值方法,进化算法也被用于求非线性方程组多个根.使用进化算法求解非线性方程组之前,首先要将方程组转化为优化问题,然后再用进化算法求解该优化问题.常用的转换方法有两种:

引用格式: 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, doi: 10.1360/SSI-2019-0154
Liao Z W, Gong W Y, Wang L. A hybrid swarm intelligence with improved ring topology for nonlinear equations (in Chinese). *Sci Sin Inform*, 2020, 50: 396-407, doi: 10.1360/SSI-2019-0154

一种是将方程组转化为单目标优化问题, 另一种则是将其转化为多目标优化问题. 对于单目标优化问题, 非线性方程组与多模态问题^[7]的求解目标相似, 因为两者都是需要找出多个最优解. Gong 等^[8]将自适应参数设置、多样性保持机制以及排斥技术结合起来求非线性方程组的多根问题. Liao 等^[9]提出了一种动态排斥技术和进化算法相结合的方法. He 等^[10]提出了基于方向的模糊邻域差分进化算法求解非线性方程组的多个根, 并获得了较好的结果. 采用多目标进化算法求非线性方程组多根的难点是如何将其转化成多目标优化问题. Grosan 和 Abraham^[11]提出了一种 m 个目标的多目标转换技术. Song 等^[12]提出了一种双目标转换技术, 并提出 MONES 算法来求解非线性方程组. 然而, 由于双目标优化问题构造方法的缺陷, MONES 可能会遗漏一些根. 为了解决这个问题, Gong 等^[13]提出了一个权重双目标转换方法 (A-WeB), 实验结果表明所提出的方法比 MONES 获得更好效果.

近年来, 人们从群体生活的生物社会行为上得到了启发, 提出了许多群体智能优化算法. 每个智能体 (agent) 都是具有经验和智慧的, 智能体之间通过相互作用进而形成强大的群体智慧来解决复杂问题. 常见的群体智能算法有: 人工蜂群算法^[14]、蚁群算法^[15]、鸽群优化算法^[16]、粒子群算法^[17]等. 目前, 群体智能算法已经被广泛用于各个领域, 包括无人机集群协同决策^[18]、数值优化问题^[19]、在线参数辨识^[20]等. 群体智能算法也被应用于非线性方程组问题的求解^[21,22]. 但文献 [21] 只关注求解非线性方程组的一个根, 而文献 [22] 中所求解根的精度较低.

采用群体智能方法对非线性方程组的多根联解需要解决两个问题: 一是增强群体的多样性; 二是加快算法的收敛速度. 群体的多样性是为了能在一次运行中获得多个不同的根, 而加快收敛速度则是在有限的计算资源的前提下, 找到符合精度的根. 为了使算法一次运行实现非线性方程组联解 (即同时求解), 本文提出一种改进的环拓扑混合群体智能算法 (IHABC). 首先设计了一种改进的环拓扑结构, 使算法能有效利用邻域信息; 其次, 采用了一种基于人工蜂群算法和差分进化算子的混合群体智能方法, 保证群体多样性的同时也提高了算法的搜索效率; 最后, 提出了个体初始化机制以增强群体的多样性. 为了验证所提出算法的性能, 选择了 8 个含有多个根的非线性方程组作为测试集. 实验结果表明, 所提出的方法能够在一次运行中找到多个根; 与代表性算法对比, 所提出的算法在找根率和成功率上有显著优势.

2 背景知识

本节首先介绍非线性方程组问题, 然后对经典人工蜂群算法进行简要介绍.

2.1 非线性方程组

一般来说, 一个非线性方程组表示如下:

$$\mathbf{e}(\mathbf{x}) = \begin{cases} e_1(\mathbf{x}) = 0, \\ \vdots \\ e_m(\mathbf{x}) = 0, \end{cases} \quad (1)$$

其中, m 为方程的个数, $\mathbf{x} = (x_1, x_2, \dots, x_D)^T$ 为一个 D 维的决策向量; $\mathbf{x} \in \mathcal{S}$, \mathcal{S} 为搜索空间的可行区域, 一般地, $\mathcal{S} = [\underline{x}_i, \bar{x}_i]^D$, 其中 $i = 1, \dots, D$, $\underline{x}_i, \bar{x}_i$ 分别为 x_i 的下限和上限. 值得说明的是, 在方程组 (1) 中至少包含一个非线性方程. 用优化算法求解非线性方程组之前, 一般需要将它转化成为最小

优化问题:

$$\text{minimize } f(\mathbf{x}) = \sum_{i=1}^m e_i^2(\mathbf{x}), \quad (2)$$

随后, 求解非线性方程组等同于求上述转化问题 (2) 的全局最小值. 当 $f(\mathbf{x}^*) < \epsilon$ 时, 则 \mathbf{x}^* 视为非线性方程组的一个根¹⁾.

2.2 人工蜂群算法

受蜜蜂觅食群体行为的启发, 人工蜂群算法 (ABC) [14] 是近年提出的一种群体智能优化算法, 其包含有 3 类蜜蜂: 雇佣蜂、观察蜂和侦查蜂. 每个蜜源代表一个可行解, 蜜源的质量对应可行解的适应值. 人工蜂群算法的主要步骤如下:

(1) 初始化参数. 人工蜂群算法的参数有: 蜜源的个数 (NP)、控制次数 (limit), 假设蜜源的维度为 D , 蜜源的产生如下所示:

$$x_{ij} = \underline{x}_{ij} + (\bar{x}_{ij} - \underline{x}_{ij}) \times r_1, \quad j \in \{1, \dots, D\}, i \in \{1, \dots, \text{NP}\}, \quad (3)$$

其中的 x_{ij} 对应的是第 i 个蜜源、第 j 维的变量, r_1 是范围为 $[0, 1]$ 的随机数, \underline{x}_{ij} 和 \bar{x}_{ij} 分别是第 i 个蜜源第 j 维变量的上界和下界.

(2) 雇佣蜂阶段. 每个雇佣蜂会在与之对应的蜜源附近产生一个新的蜜源, 如下所示:

$$x'_{ij} = x_{ij} + (x_{ij} - x_{kj}) \times r_2, \quad j \in \{1, \dots, D\}, k \in \{1, \dots, \text{NP}\}, \quad (4)$$

其中 $k \neq i$, r_2 是 $[-1, 1]$ 中均匀分布的实数, 如果新蜜源的适应值优于或等于原蜜源的适应值, 则新蜜源取代原蜜源, 否则保留原蜜源.

(3) 观察蜂阶段. 观察蜂通过概率 p_i 选择一个蜜源 \mathbf{x}_i 进行操作, p_i 的计算公式如下所示:

$$p_i = \frac{f_i}{\sum_{i=1}^{\text{NP}} f_i}, \quad (5)$$

其中 f_i 表示蜜源 \mathbf{x}_i 的适应值, 蜜源的适应值越大, 被选择的概率就越大. 蜜源被选择后, 通过式 (4) 得到一个新蜜源, 如果新蜜源的适应度值优于 \mathbf{x}_i 的适应度值, 则新蜜源替代 \mathbf{x}_i .

(4) 如果蜜源 \mathbf{x}_i 经过 limit 次迭代操作后适应值都没有改进, 则舍弃这个蜜源并启用侦查蜂. 侦查蜂通过式 (3) 产生一个新蜜源.

3 基于改进环形拓扑的混合群体智能算法

3.1 改进的环形拓扑结构

文献 [23] 提出了环形拓扑的粒子群算法来定位多模态问题的多个全局和局部最优解, 它是基于下标 (index-based) 的排列结构. 然而基于邻近的下标排列的个体, 名义上虽然是相邻的, 但是在搜索空间上可能两个个体离的相对较远. 图 1 的左子图为一个基于下标的环拓扑结构. 从图中可以看出点 6 和 4 是点 5 的左右邻域个体, 虽然在下标上是相邻的, 但实际在搜索空间上 (按欧氏距离计算), 点 1 和 3 离点 5 更近. 针对上述的缺点, 本文提出了一种改进的环拓扑结构, 具体步骤如算法 1 所示.

1) 本文中, 如果 $D \leq 5$, $\epsilon = 10^{-6}$; 其他情况 $\epsilon = 10^{-4}$.

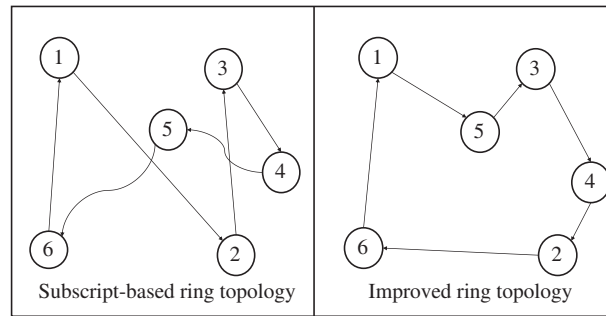


图 1 基于下标的和改进的环拓扑比较

Figure 1 Comparison between subscript-based ring topology and improved ring topology

Algorithm 1 The improved ring topology**Input:** Initial population \mathcal{P} .

- 1: Select the best nectar source, set it as the first node of the ring topology, and remove it from \mathcal{P} ;
- 2: **for** $i = 2$ to NP **do**
- 3: Find the nectar source in \mathcal{P} nearest to node x_{i-1} in the topological structure of the ring topology, set it as x_i , and remove it from \mathcal{P} ;
- 4: **end for**

Output: the improved ring topology.

图 1 的右子图为得到的一个改进的环拓扑结构, 该拓扑结构的个体的左右邻居在搜索空间也距离较近. 如, 点 5 的左右邻居分别为点 1 和 3, 这两个点在搜索空间上离点 5 较近. 因此点 5 能够利用这两个邻居的信息产生新的个体, 提高搜索的效率. 而图 1 的左子图, 点 5 的左右邻居为点 6 和 4, 这两个点在距离上离点 5 较远. 如果采用点 6 和 4 的信息来指导点 5 的进化, 可能会导致在搜索非线性方程组多个根时出现振荡的问题.

3.2 混合群体智能操作

由于人工蜂群算法简单、高效和容易实现等特点, 本文以人工蜂群算法作为基本优化算法, 同时为了一次运行能求解非线性方程组的多根, 设计一种混合的方法来改进雇佣蜂和观察蜂的操作.

在雇佣蜂阶段, 原始的雇佣蜂操作产生新的蜜源用于找到全局最优解. 然而, 该操作存在群体多样性丢失的缺点, 不利于非线性方程组的多根联解. 受粒子群优化算法的启发采用如下方式来产生新的蜜源:

$$\mathbf{x}'_i = \mathbf{x}_i + (\text{localBest}(i) - \mathbf{x}_i) \times c1 \times \text{rand} + (\text{nBest}(i) - \mathbf{x}_i) \times c2 \times \text{rand}, \quad (6)$$

其中 $c1, c2$ 为给定的固定值; rand 为在 $(0, 1)$ 之间的随机数; $\text{localBest}(i)$ 在当前蜜源 i 所经历过的最好位置; 而 $\text{nBest}(i)$ 则表示在当前蜜源 i 的左右两个邻居 (从改进的环拓扑结构中选取) 中所经历的最好位置.

改进的雇佣蜂操作虽然能够保证群体的多样性, 但是也减缓了算法在不同根附近的收敛速度. 为了在有限的计算资源中获得满足精度的多个根, 在观察蜂阶段, 引入了邻域拥挤的差分进化算子^[7]机制来替换原有观察蜂操作. 采用邻域拥挤技术的主要目的在邻域的范围进行变异交叉操作, 提高算法搜索多个蜜源位置的速度. 本文中, 个体的邻域大小为 5.

3.3 个体初始化机制

随着迭代的进行, 观察蜂将找到最佳的蜜源位置 (非线性方程组的根), 如果不对观察蜂处理, 持续地搜索将会导致在下次循环时定位到同一个根. 因此, 一旦观察蜂找到最佳蜜源, 则把这个根存储到外部存档. 此外, 为了增强群体多样性, 将该观察蜂转换为侦查蜂, 在搜索空间中随机初始化.

3.4 算法流程

算法 2 是所提出方法的伪代码. FEs 为适应值评价次数, FES_{\max} 为最大适应值评价次数, $\text{fit}(\cdot)$ 是个体适应值, $\text{neighborhoodBest}(\cdot)$ 代表第 i 个个体邻居最好的位置. 第 4 行是采用改进方法建立基于欧氏距离的环拓扑结构; 第 5~17 行是雇佣蜂操作, 该操作用于保存群体多样性, 其中, 第 5~10 行更新 localBest, 第 11~13 行用于更新 nBest, 第 14~17 行雇佣蜂使用 localBest 和 nBest 的信息来更新蜜源位置; 第 18~26 行是观察蜂操作, 通过采用邻域变异交叉操作, 提高了算法的收敛速度; 第 27 行是侦查蜂操作. 算法运行结束将返回所有已找到的根.

Algorithm 2 A hybrid swarm intelligence algorithm with improved ring topology

```

1: Initialize population  $\mathcal{P}$  and evaluate the fitness;
2: localBest = nBest =  $\mathcal{P}$ , Iter = 1;
3: while FEs <  $FES_{\max}$  do
4:   Create a ring topology  $\alpha$  via Algorithm 1;
5:   for  $i = 1$  to NP do
6:     Find the nearest nectar source localBest(index) to  $\alpha_i$  in localBest;
7:     if  $\text{fit}(\alpha_i) < \text{fit}(\text{localBest}(\text{index}))$  then
8:       localBest(index) =  $\alpha_i$ ;
9:     end if
10:  end for
11:  for  $i = 1$  to NP do
12:    nBest( $i$ )  $\leftarrow$  neighborhoodBest(localBest( $i - 1$ ), localBest( $i$ ), localBest( $i + 1$ ));
13:  end for
14:  for  $i = 1$  to NP do
15:    Update nectar source location via Eq. (6);
16:    FEs = FEs + NP;
17:  end for
18:  for  $j = 1$  to NP do
19:    Generate a new position via Eq. (6), and calculate its fitness;
20:    FEs = FEs + 1;
21:    if the root is found then
22:      Store the root to an archive;
23:      Generate a new nectar source via Eq. (3) and calculate the fitness;
24:      FEs = FEs + 1;
25:    end if
26:  end for
27:  If nectar source has not been updated for more than limit, generate a new nectar source via Eq. (3);
28:  FEs = FEs + 1;
29:  Iter = Iter + 1;
30: end while
31: Return all the found roots.
    
```

3.5 算法复杂度分析

在所提出的方法中, 算法复杂度的来源主要包含 4 个部分: 构造环形拓扑结构, 雇佣蜂产生新蜜源, 观察蜂在蜜源附近采蜜, 侦查蜂操作. 主要的复杂度如下:

表 1 不同方法的参数设置

Table 1 Parameter settings of different algorithms

Algorithm	Parameter settings
IHABC	NP = 100, $c_1 = c_2 = 2.05$, $F = 0.5$, CR = 0.9, limit = 50
CADE ^[24]	NP = 100, $F = 0.5$, CR = 0.9, $T = 10$
MONES ^[12]	NP = 100, $H_m = NP$
A-WeB ^[13]	NP = 100, $H_m = NP$
RADE ^[8]	NP = 100, $H_m = 200$
DREA ^[9]	NP = 10, $u_{CR} = 0.5$, $u_F = 0.5$, $c = 0.1$
MODFA ^[22]	NP = 100, $\alpha = 0.23$, $\beta_0 = 1$, $\delta = 0.98$, $\gamma = 1$
FONDE ^[10]	NP = 100, $F = 0.5$, CR = 0.9, $m = 11$

(1) 在构造环形拓扑结构时, 复杂度为 $\mathcal{O}(\text{Iter} \times \text{NP} \times \log(\text{NP}))$, 其中, Iter 为迭代次数, NP 为群体个数;

(2) 雇佣蜂产生新蜜源, 复杂度为 $\mathcal{O}(\text{Iter} \times \text{NP})$;

(3) 观察蜂采蜜, 复杂度为 $\mathcal{O}(\text{Iter} \times \text{NP})$;

(4) 侦查蜂操作, 复杂度为 $\mathcal{O}(\text{Iter} \times \log(\text{NP}))$.

因此, 所提出算法总复杂度为 $\mathcal{O}(\text{Iter} \times \text{NP} \times \log(\text{NP}))$.

4 实验结果与分析

为了验证所提出算法的性能, 本文的测试集采用文献 [24] 中包含有根较多的 8 个方程组. 虽然 F08 只包含 2 个根, 但是不易计算; 此外, 其他的非线性方程组包含的根都大于 7, 能较好地评价算法的性能. 其余的详细信息可参考文献 [24]. 同时, 为了评价算法的性能, 与文献 [8,9,24] 类似, 采用找根率 (root ratio, RR) 和成功率 (success rate, SR) 作为评价指标.

4.1 IHABC 与其他算法对比

本文中 IHABC 主要和以下几种方法进行比较: CADE^[24], MONES^[12], A-WeB^[13], RADE^[8], DREA^[9], MODFA^[22] 和 FONDE^[10]. 上述几种方法的基本参数设置如表 1 所示. 为了公平比较, 所有的算法都独立运行 30 次. 所有算法都在 Matlab 2013b 中执行, 所有的对比实验在 Windows 10 64 位操作系统下使用 Intel Core i7-6700 处理器 @3.40 GHz, 8 GB RAM 的台式 PC 上执行的.

表 2 和 3 分别表示各算法在求 8 个测试函数时所获得的平均找根率 (RR) 和平均成功率 (SR). 从表中可以看出 IHABC 获得了最高的平均找根率 (0.97) 和最高的平均成功率 (0.86). 此外, FONDE, CADE, RADE, DREA, A-WeB, MODFA 和 MONES 分别排名第 2~8 位. 同时, 从表中可以看出, IHABC 在 F03~F07 获得了最优的 RR 和 SR 值. 因此, 所提出的 IHABC 算法比其余 7 种算法在求解非线性方程组多根的性能上更具优势.

图 2 显示了不同算法在求解 F04 和 F07 获得的找根率 RR. 由于 MONES 和 MODFA 获得的结果较差, 为了便于观察, 这两种方法的曲线没有在图中显示出来. 从图中可以看出, 相比较于其他算法, IHABC 开始时 RR 较低. 如图 2(a) 所示, IHABC 在初始阶段比 A-WeB, RADE, DREA 所得到的找根率低. 同样地, 在图 2(b) 中, IHABC 在开始时的找根率比 RADE, DREA 低. 这是由于算法为了保

表 2 不同算法的找根率 (RR) 对比

Table 2 Comparison of different algorithms with respect to root ratio (RR)

Problem	IHABC	CADE	MONES	A-WeB	RADE	DREA	MODFA	FONDE
F01	0.97	0.93	0.59	1.00	0.90	0.72	0.90	0.96
F02	0.99	1.00	1.00	0.94	0.99	1.00	0.59	0.99
F03	1.00	1.00	0.77	0.83	0.99	1.00	0.95	1.00
F04	0.89	0.70	0.43	0.88	0.63	0.84	0.82	0.86
F05	1.00	1.00	0.19	0.97	0.98	0.77	0.86	1.00
F06	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00
F07	0.98	0.93	0.14	0.15	0.56	0.87	0.00	0.91
F08	0.98	0.95	0.31	0.23	0.83	1.00	0.00	1.00
Average	0.97	0.94	0.55	0.75	0.86	0.90	0.64	0.96

表 3 不同算法的成功率 (SR) 对比

Table 3 Comparison of different algorithms with respect to success rate (SR)

Problem	IHABC	CADE	MONES	A-WeB	RADE	DREA	MODFA	FONDE
F01	0.73	0.46	0.00	1.00	0.31	0.00	0.06	0.52
F02	0.96	1.00	1.00	0.60	0.93	1.00	0.00	0.98
F03	1.00	1.00	0.00	0.12	0.98	1.00	0.80	1.00
F04	0.43	0.06	0.00	0.28	0.00	0.20	0.20	0.28
F05	1.00	1.00	0.00	0.76	0.89	0.00	0.40	1.00
F06	1.00	1.00	1.00	1.00	0.94	1.00	1.00	1.00
F07	0.80	0.40	0.00	0.00	0.00	0.00	0.00	0.28
F08	0.96	0.90	0.07	0.02	0.67	1.00	0.00	1.00
Average	0.86	0.72	0.25	0.47	0.59	0.53	0.30	0.75

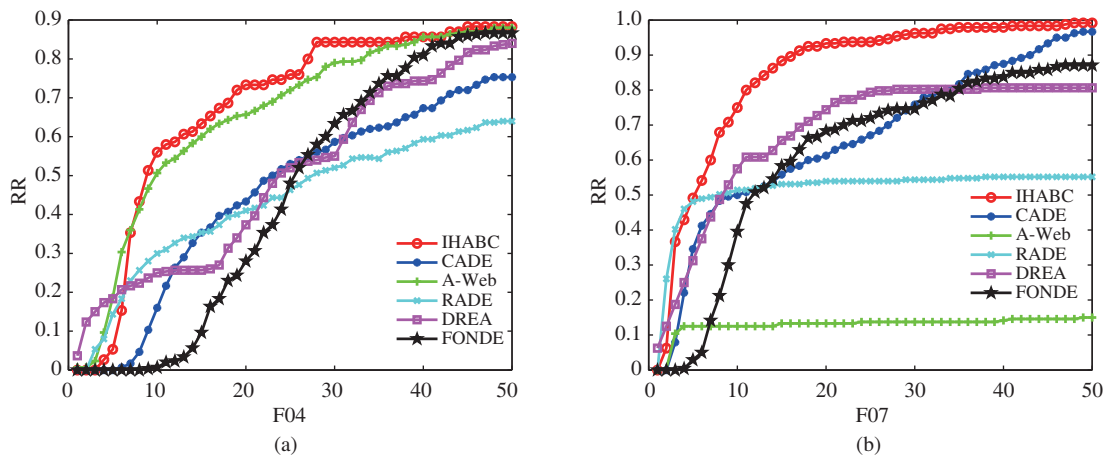


图 2 (网络版彩图) 不同算法的 RR 收敛曲线对比

Figure 2 (Color online) Comparison of RR convergence curves of different algorithms. (a) F04; (b) F07.

表 4 IHABC 不同组成部分的找根率 (RR) 对比
 Table 4 Comparison of different components in IHABC with respect to root ratio (RR)

Problem	IHABC-1	IHABC-2	IHABC-3	IHABC-4	IHABC-5	IHABC-6	IHABC-7	IHABC
F01	0.65	0.77	0.06	0.83	0.98	0.87	0.68	0.97
F02	0.57	0.81	0.00	0.97	0.98	0.99	0.80	0.99
F03	0.60	0.84	0.00	0.95	1.00	1.00	0.93	1.00
F04	0.69	0.64	0.18	0.63	0.71	0.60	0.51	0.88
F05	0.67	0.97	0.16	0.98	1.00	1.00	1.00	1.00
F06	0.69	0.72	0.10	0.73	1.00	1.00	0.99	1.00
F07	0.13	0.14	0.05	0.45	0.56	0.83	0.56	0.98
F08	0.91	0.90	0.00	0.93	0.78	0.98	0.58	0.98
Average	0.61	0.72	0.07	0.81	0.87	0.91	0.76	0.97

持多样性, 牺牲了一定的收敛速度. 但随着进化代数不断增加, 在群体多样性保持的条件下, IHABC 在改进的观察蜂算子作用下, 较快地收敛到不同的根, 所以 IHABC 获得的 RR 明显优于其他算法.

4.2 IHABC 中不同改进的影响

在第 3 节中指出, 所提出的 IHABC 算法用来求解非线性方程组的多个根. 它主要包含 3 个部分, 改进的环形拓扑结构及雇佣蜂算子, 基于邻域拥挤的差分进化算法替换观察蜂操作, 个体初始化机制. 本小节主要研究 IHABC 中不同部分的组成在求解非线性方程组时的影响. 以下算法采用如表 1 中 IHABC 相同的参数设置.

- (1) IHABC-1, 采用改进的环形拓扑结构及雇佣蜂算子, 其他 ABC 算法操作不变.
- (2) IHABC-2, 采用邻域拥挤的差分进化算法替换观察蜂操作, 其他 ABC 算法操作不变.
- (3) IHABC-3, 采用个体初始化机制操作求方程组的多根, 其他 ABC 算法操作不变.
- (4) IHABC-4, 采用改进的环形拓扑结构及雇佣蜂算子, 邻域拥挤的差分进化算法.
- (5) IHABC-5, 采用改进的环形拓扑结构及雇佣蜂算子, 个体初始化机制两种操作.
- (6) IHABC-6, 采用邻域拥挤的差分进化算法和个体初始化机制两种操作.
- (7) IHABC-7, 采用基于下标的环拓扑结构替换改进的环拓扑结构, 其他 IHABC 算法操作不变.
- (8) IHABC, 本文提出的方法.

表 4 和 5 展示 IHABC 不同组成部分在求解非线性方程组多根问题时详细的找根率 (RR) 和成功率 (SR). 接下来我们将对这些结果进行分析.

- IHABC-3 在原始 ABC 的基础上加入了个体初始化机制来求方程组的解. 虽然在表 4 中可以看出能找到多个根, 但是效果较差. 主要的原因有: (1) 原始算法可能搜索性能有限; (2) 原始 ABC 主要用于求单解的, 虽然加入个体初始化机制, 但对性能提升较少. 此外, 从表 4 中可以看出, 该方法在 F02, F03, F08 中一个根都找不到, 这说明了原有的 ABC 算法的搜索效率需要提高.

- IHABC-1 和 IHABC-2 比 IHABC-3 获得了更好的结果. 这反映了改进的环形拓扑结构及雇佣蜂算子和邻域拥挤的差分进化算法在一定程度上能提高非线性方程组找根的效果. 但是从表 4 和 5 中看来, 找根率和成功率仍然较低.

- IHABC-4, IHABC-5 和 IHABC-6 分别是将本文提出的 3 种改进方法进行两两结合来求解非线性方程组. 从表 4 和 5 中可以看出, 两两结合获得的结果比单一用一种改进方法的性能要提升很多.

表 5 IHABC 不同组成部分的成功率 (SR) 对比

Table 5 Comparison of different components in IHABC with respect to success rate (SR)

Problem	IHABC-1	IHABC-2	IHABC-3	IHABC-4	IHABC-5	IHABC-6	IHABC-7	IHABC
F01	0.00	0.00	0.00	0.00	0.76	0.10	0.00	0.73
F02	0.03	0.06	0.00	0.76	0.90	0.96	0.06	0.96
F03	0.00	0.20	0.00	0.70	1.00	1.00	0.56	1.00
F04	0.06	0.06	0.00	0.03	0.06	0.00	0.00	0.43
F05	0.00	0.73	0.00	0.90	1.00	1.00	1.00	1.00
F06	0.00	0.00	0.00	0.00	1.00	1.00	0.96	1.00
F07	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.80
F08	0.86	0.80	0.00	0.86	0.57	0.96	0.16	0.96
Average	0.12	0.23	0.00	0.40	0.66	0.63	0.34	0.86

表 6 不同邻域大小对 IHABC 的影响

Table 6 Influence of different neighborhood sizes on IHABC

Problem	RR					SR				
	$n = 5$	$n = 10$	$n = 15$	$n = 20$	$n = 25$	$n = 5$	$n = 10$	$n = 15$	$n = 20$	$n = 25$
F01	0.97	0.99	0.98	0.97	0.94	0.73	0.87	0.80	0.67	0.40
F02	0.99	1.00	1.00	1.00	1.00	0.96	1.00	1.00	1.00	1.00
F03	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F04	0.88	0.91	0.70	0.76	0.81	0.43	0.60	0.07	0.20	0.33
F05	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	0.93
F06	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F07	0.98	0.96	0.86	0.75	0.69	0.80	0.47	0.00	0.00	0.00
F08	0.98	0.90	0.93	0.90	0.87	0.96	0.80	0.87	0.80	0.73
Average	0.97	0.96	0.94	0.92	0.91	0.86	0.84	0.72	0.71	0.68

• IHABC 是本文提出的算法, 从表 4 和 5 中可以看出, 它获得了最好的找根率和成功率. 这与 3 个部分的共同协作是密不可分. 改进的环形拓扑结构及雇佣蜂算子用于保持种群的多样性; 邻域拥挤的差分进化算法用于提高寻找不同蜜源的效率; 而个体初始化机制则用来进一步提高种群多样性. 因此, IHABC 能获得最好的结果.

• 值得关注的是关于 IHABC 与 IHABC-7 的比较. IHABC-7 用基于下标的环拓扑结构替换改进的环拓扑结构, 其他 IHABC 算法操作不变. 从表 4 和 5 中可以看出, IHABC 得到的结果明显优于 IHABC-7. 这说明了基于下标的环拓扑结构虽然能够在一定程度上保证种群的多样性, 但是由于相邻下标的个体在搜索区域 (按欧氏距离计算) 不一定是相邻的, 因此降低了算法的搜索效率. 而改进的环拓扑结构能较好地弥补这个缺陷, 提升了 IHABC 的性能.

4.3 IHABC 中不同邻域大小的影响

在 3.2 小节中, 观察蜂阶段引入了邻域大小的参数. 在之前的实验中, $n = 5$, n 表示选择邻域个体的大小. 本小节取不同的邻域值 $n = \{10, 15, 20, 25\}$, 以研究该参数对所提出算法性能的影响.

详细的找根率 (RR) 和成功率 (SR) 结果在表 6 中显示, 从表中可以看出, 邻域大小 $n = 5$ 时获得

的效果最好,随着 n 的增大, IHABC 的性能逐渐下降. 原因如下: n 较小时,选择的个体较为相似,因此能达到快速收敛的作用;当 n 较大时,选择个体进行变异时,可能会导致差分向量过大,从而搜索到其他的区域,影响了算法的收敛性. 因此, $n \in [5, 10]$ 是一个较为合理的邻域大小.

5 结论

为了实现非线性方程组多根同时求解,本文提出了一种基于改进环形拓扑结构的混合人工蜂群算法. 改进的环拓扑结构能够弥补基于下标的环拓扑结构中相邻个体可能在搜索区间上不相邻的问题. 为了更好地利用环拓扑结构中邻域的信息,设计了改进的雇佣蜂算子. 此外,利用基于邻域拥挤的差分进化算法替换原有的观察蜂操作,提高了算法的搜索效率. 最后,为了节约计算资源和保持种群多样性,引入了个体初始化机制. 为了验证所提出方法 IHABC 的有效性,选择了 8 个非线性方程组作为测试集. 同时,选择 7 种代表性方法进行对比,实验结果显示 IHABC 获得了最好的结果.

参考文献

- 1 Kastner M. Phase transitions and configuration space topology. *Rev Mod Phys*, 2008, 80: 167–187
- 2 Guo D S, Nie Z Y, Yan L C. The application of noise-tolerant ZD design formula to robots' kinematic control via time-varying nonlinear equations solving. *IEEE Trans Syst Man Cybern Syst*, 2018, 48: 2188–2197
- 3 Facchinei F, Kanzow C. Generalized Nash equilibrium problems. *Ann Oper Res*, 2010, 175: 177–211
- 4 Mehta D, Grosan C. A collection of challenging optimization problems in science, engineering and economics. In: *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, 2015. 2697–2704
- 5 Darvishi M T, Barati A. A third-order Newton-type method to solve systems of nonlinear equations. *Appl Math Comput*, 2007, 187: 630–635
- 6 Knoll D A, Keyes D E. Jacobian-free Newton-Krylov methods: a survey of approaches and applications. *J Comput Phys*, 2004, 193: 357–397
- 7 Qu B Y, Suganthan P N, Liang J J. Differential evolution with neighborhood mutation for multimodal optimization. *IEEE Trans Evol Comput*, 2012, 16: 601–614
- 8 Gong W Y, Wang Y, Cai Z H, et al. Finding multiple roots of nonlinear equation systems via a repulsion-based adaptive differential evolution. *IEEE Trans Syst Man Cybern Syst*, 2018. doi: 10.1109/TSMC.2018.2828018
- 9 Liao Z W, Gong W Y, Yan X S, et al. Solving nonlinear equations system with dynamic repulsion-based evolutionary algorithms. *IEEE Trans Syst Man Cybern Syst*, 2018. doi: 10.1109/TSMC.2018.2852798
- 10 He W, Gong W Y, Wang L, et al. Fuzzy neighborhood-based differential evolution with orientation for nonlinear equation systems. *Knowl-Based Syst*, 2019, 182: 104796
- 11 Grosan C, Abraham A. A new approach for solving nonlinear equations systems. *IEEE Trans Syst Man Cybern A*, 2008, 38: 698–714
- 12 Song W, Wang Y, Li H X, et al. Locating multiple optimal solutions of nonlinear equation systems based on multiobjective optimization. *IEEE Trans Evol Comput*, 2015, 19: 414–431
- 13 Gong W Y, Wang Y, Cai Z H, et al. A weighted biobjective transformation technique for locating multiple optimal solutions of nonlinear equation systems. *IEEE Trans Evol Comput*, 2017, 21: 697–713
- 14 Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J Glob Optim*, 2007, 39: 459–471
- 15 Zhou F, Zhang X F, Zhang G B, et al. The effects of parameters settings of ant colony algorithm on the performance of ultrasonic echo estimation. *Sci Sin Inform*, 2013, 43: 243–253 [周方, 张小凤, 张光斌, 等. 蚁群算法中参数设置对超声回波估计性能的影响. *中国科学: 信息科学*, 2013, 43: 243–253]
- 16 Duan H B, Ye F. Progresses in pigeon-inspired optimization algorithms. *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]
- 17 Kennedy J. Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*, Piscataway, 1995. 1942–1948

- 18 Duan H B, Zhang D F, Fan Y M, et al. From wolf pack intelligence to UAV swarm cooperative decision-making. *Sci Sin Inform*, 2019, 49: 112–118 [段海滨, 张岱峰, 范彦铭, 等. 从狼群智能到无人机集群协同决策. *中国科学: 信息科学*, 2019, 49: 112–118]
- 19 Zhu G P, Kwong S. Gbest-guided artificial bee colony algorithm for numerical function optimization. *Appl Math Comput*, 2010, 217: 3166–3173
- 20 Li S T, Duan H B. Artificial bee colony approach to online parameters identification for hypersonic vehicle. *Sci Sin Inform*, 2012, 42: 1350–1363 [李霜天, 段海滨. 基于人工蜂群优化的高超声速飞行器在线参数辨识. *中国科学: 信息科学*, 2012, 42: 1350–1363]
- 21 Zhang X M, Wan Q, Fan Y H. Applying modified cuckoo search algorithm for solving systems of nonlinear equations. *Neural Comput Appl*, 2019, 31: 553–576
- 22 Ariyaratne M K A, Fernando T G I, Weerakoon S. Solving systems of nonlinear equations using a modified firefly algorithm (MODFA). *Swarm Evolary Comput*, 2019, 48: 72–92
- 23 Li X D. Niching without niching parameters: particle swarm optimization using a ring topology. *IEEE Trans Evol Comput*, 2010, 14: 150–169
- 24 Wnag K, Gong W Y. Solving nonlinear equations system with an improved differential evolution. *Control Decis*, 2019. doi: 10.13195/j.kzyjc.2018.1739 [王开, 龚文引. 求解非线性方程组系统的改进差分进化算法. *控制与决策*, 2019. doi: 10.13195/j.kzyjc.2018.1739]

A hybrid swarm intelligence with improved ring topology for non-linear equations

Zuowen LIAO¹, Wenyin GONG^{1*} & Ling WANG^{2*}

1. *School of Computer Science, China University of Geosciences, Wuhan 430074, China;*

2. *Department of Automation, Tsinghua University, Beijing 100084, China*

* Corresponding author. E-mail: wygong@cug.edu.cn, wangling@tsinghua.edu.cn

Abstract Solving nonlinear equations (NEs) is one of the most important yet challenging tasks in numerical computation, especially for the simultaneous location of multiple roots in a single run. In this paper, a hybrid swarm intelligence approach with improved ring topology is proposed to tackle this problem. It has three main features: (i) the improved ring topology is developed to effectively use the neighborhood knowledge; (ii) the hybrid swarm intelligence enhances the search efficiency; and (iii) an individual reinitialization mechanism is proposed to enrich the population diversity. The performance of this approach is tested on eight NE problems with multiple roots, experimentally confirming that it can simultaneously locate multiple roots in a single run. In addition, it can provide better results than conventional methods in terms of both root and success rates.

Keywords nonlinear equations, swarm intelligence, simultaneously multiple roots locating, artificial bee colony, ring topology



Zuowen LIAO received the B.S. and M.S. degrees in computer science from Wuhan Institute of Chemical Technology, Wuhan, in 2006 and 2010, respectively. He is currently pursuing the Ph.D. degree in computer science at China University of Geosciences, Wuhan. His current research interests include evolutionary computation, memetic computation, and computational intelligence.



Wenyin GONG received the B.E., M.E., and Ph.D. degrees in computer science from China University of Geosciences, Wuhan, in 2004, 2007, and 2010, respectively. He is currently a professor at the School of Computer Science, China University of Geosciences, Wuhan. His research interests include evolutionary algorithms, evolutionary optimization, and their applications.



Ling WANG received the B.S. degree in automation and the Ph.D. degree in control theory and control engineering from Tsinghua University, Beijing, China, in 1995 and 1999, respectively. Since 1999, he has worked at the Department of Automation, Tsinghua University, where he became a full professor in 2008. His current research interests include intelligent optimization and production scheduling.