

一种基于强化学习的自适应多邻域人工蜂群算法

周新宇¹⁾ 尹子悦¹⁾ 高卫峰²⁾ 谭贵森¹⁾ 易玉根³⁾

¹⁾(江西师范大学 计算机信息工程学院 南昌 330022)

²⁾(西安电子科技大学 数学与统计学院 西安 710126)

³⁾(江西师范大学 软件学院 南昌 330022)

摘要 邻域拓扑是提高人工蜂群算法性能的一种有效手段. 然而, 现有相关工作主要是在种群层次上实现了单一邻域拓扑, 这种方式忽略了不同类型的邻域拓扑能优势互补, 使得算法性能还有一定局限性. 为此, 本文结合强化学习, 提出在个体层次上实现多邻域拓扑. 将种群中的个体视作智能体, 设计了基于邻域拓扑的状态和动作, 选用 4 种不同特征的邻域拓扑用于构建邻域候选池, 之后采用 Q-learning 方法根据个体的奖励情况为其自适应选择不同的邻域拓扑. 该方式相对于现有的单一邻域拓扑, 更能充分发挥不同邻域信息对算法搜索的引导作用. 在 CEC2013 和 CEC2017 两套测试集以及两个实际优化问题上进行了大量实验, 与 4 种邻域相关 ABC 和 4 种知名改进 ABC 进行了性能对比, 结果表明该算法的收敛精度和速度均有更好表现, 可有效增强邻域人工蜂群算法的性能.

关键词 群智能; 人工蜂群; 个体; 强化学习; 邻域拓扑

中图法分类号 TP311 **DOI 号** 10.11897/SP.J.1016.2024.01521

Adaptive Multi-Neighborhood Artificial Bee Colony Algorithm Based on Reinforcement Learning

ZHOU Xin-Yu¹⁾ YIN Zi-Yue¹⁾ GAO Wei-Feng²⁾ TAN Gui-Sen¹⁾ YI Yu-Gen³⁾

¹⁾(School of Computer and Information Engineering, Jiangxi Normal University, Nanchang 330022)

²⁾(Department of Applied Mathematics, Xidian University, Xi'an 710126)

³⁾(School of Software, Jiangxi Normal University, Nanchang 330022)

Abstract Neighborhood topology is an effective way to enhance the performance of artificial bee colony (ABC) algorithm. However, for the existing related works, they mainly focus on a single neighborhood topology at the population level. This way overlooks the complementary advantages of different neighborhood topologies, and limits the algorithm performance to some extent. To remedy this issue, in this work, we propose the multiple neighborhood topologies mechanism at the individual level by combining the reinforcement learning. Specifically, the individuals in the population as considered as agents, and then the state and action of the agent are designed based on the neighborhood topology. After that, four different types of neighborhood topologies are used to construct the neighborhood candidate pool, and then the Q-learning method is employed to adaptively select different neighborhood topologies for the individuals according to their rewards. Compared with the existing methods of a single

收稿日期: 2023-08-02; 在线发布日期: 2024-04-18. 本课题得到国家自然科学基金(62366022, 61966019, 62276202, 62062040)、江西省自然科学基金杰出青年基金项目(20212ACB212003)、江西省主要学科学术和技术带头人培养计划(20212BCJ23017)、江西省自然科学基金项目(20232BAB202048)资助. 周新宇(通信作者), 博士, 副教授, 中国计算机学会(CCF)会员, 主要研究领域为智能计算. E-mail: xyzhou@jxnu.edu.cn. 尹子悦, 本科生, 主要研究领域为智能优化算法. 高卫峰, 博士, 教授, 国家级青年人才, 中国计算机学会(CCF)会员, 主要研究领域为智能计算. 谭贵森, 硕士研究生, 主要研究领域为智能优化算法. 易玉根, 博士, 副教授, 主要研究领域为机器学习.

neighborhood topology, our approach can make full use of different neighborhood information in guiding the algorithm search. Extensive experiments are conducted on the CEC2013 and CEC2017 test suites, as well as two real-world optimization problems. Comparative results against four neighborhood-based ABC variants and other four well-known ABC variants demonstrate that our approach shows better performance in terms of both convergence accuracy and speed, and it can significantly improve the performance of neighborhood-based ABC.

Keywords swarm intelligence; artificial bee colony; individual; reinforcement learning; neighborhood topology

1 引 言

群智能优化算法是近年来较为流行的一种全局优化技术^[1],它通过模拟自然界中社会性生物群体的群集行为来实现寻优,例如:粒子群优化算法模拟了鸟群的飞翔行为^[2]、蚁群算法模拟了蚂蚁的觅食行为^[3]。作为一种代表性的群智能优化算法,人工蜂群算法(Artificial Bee Colony, ABC)模拟了蜂群的智能采蜜行为^[4],是 Karaboga 等人于 2005 年首次提出^[5],它根据不同种类蜜蜂的分工协作来实现整个蜂群采蜜量的最大化,即找到待优化问题的最优解。因算法结构简单、易实现,且无需过多的参数调节,ABC 自提出以来便备受关注,已成功应用于解决多种实际优化问题,比如:无线传感器网络的节点布局问题^[6]、机器人路径规划问题^[7]以及半导体最终测试调度问题^[8]等。

虽然 ABC 在解决许多优化问题时已显示出较好的有效性和高效性,但在面对一些复杂的优化问题时,例如:多峰优化问题,ABC 会出现收敛速度慢、求解精度低等不足。已有一些相关研究工作指出^[9],导致这种不足的主要原因是 ABC 的解搜索方程有较强的勘探能力,但开采能力却较弱,使得算法的勘探和开采能力不平衡。为解决这些不足,很多不同类型的改进 ABC 相继被提出,其中的一种主流改进思路是通过利用最优个体或精英个体来增强算法的开采能力^[9-11],相应的一些实验结果也验证了这一思路的可行性。然而,过度使用最优个体或精英个体易使得 ABC 较为贪婪,从而引发算法的早熟问题。

为避免上述问题,最近研究人员开始从邻域拓扑的角度来改进 ABC^[12-14],为平衡算法的开采和勘探能力提供了新的思路。这些相关工作的主要贡献在于利用了邻域信息来指导算法搜索,可避免直接

使用最优个体或精英个体,从而在提高算法开采能力的同时还能保持较好的种群多样性。例如,Wang 等人^[13]提出了一种基于环形邻域拓扑的改进 ABC (NSABC),为种群构建了基于个体索引关系的环形邻域拓扑,再选择邻域内的最优个体来指导搜索;不同于 NSABC 只采用单一邻域拓扑的方式,Zhou 等人^[14]提出了一种基于多邻域拓扑的改进 ABC (ABC-ANT),选择了三种不同类型的邻域拓扑来构建候选池,再采用适应度-距离相关性技术 (Fitness Distance Correlation, FDC)来自适应选择最为恰当的邻域拓扑。毋庸置疑,这些邻域相关的改进 ABC 有效提高了 ABC 性能。然而,它们还存在一定的局限性,主要表现在:(1)通常仅使用单一邻域拓扑,未充分发挥出不同邻域拓扑的作用;(2)即使有少量相关工作开始关注多邻域拓扑,但其仅从种群或子种群的层次来使用邻域,这种粗粒度的方式灵活性不够。

为此,为进一步提高邻域拓扑在 ABC 中的作用,本文从个体层次提出了一种基于强化学习的自适应多邻域 ABC,记作 RLABC。在该算法中,我们首先选择了四种不同类型的邻域拓扑用于构建邻域候选池,在算法的进化初期为每个个体随机选择一种邻域拓扑,再根据不同邻域拓扑在算法进化过程中的表现,采用强化学习技术来更新每种邻域拓扑的奖励情况,使得算法能自适应地为每个个体选择最恰当的邻域拓扑。本文主要贡献可概括如下:

(1)不同于现有的邻域 ABC,RLABC 从个体层次来实现多邻域拓扑,这种细粒度的邻域使用方式更能充分发挥出邻域信息的引导作用,可有效增强邻域拓扑使用的灵活性。

(2)为自适应选择邻域拓扑,采用了强化学习技术中的 Q-learning 方法,把每个个体视作一个智能体,定义了基于邻域拓扑的状态和动作,再根据不同邻域拓扑在算法进化过程中的表现来更新 Q 表,可

更加精准地选择邻域。

(3)在 CEC2013 和 CEC2017 测试集以及两个实际优化问题上进行了大量实验,与 4 种邻域相关的改进 ABC 和 4 种知名的其他改进 ABC 进行了对比,结果表明 RLABC 具有更好性能。

需说明的是,在本文的外审返修阶段,我们被提醒 QLPSO^[15]同样采用了强化学习在个体层次上进行邻域自适应.然而,本文算法与 QLPSO 的主要区别是:(1)从邻域的角度来看,QLPSO 是在单一邻域拓扑的基础上进行邻域规模的自适应,比如:针对环形邻域设计了 2 个、4 个、8 个等不同规模的邻域大小,但本文算法的重点是多种邻域拓扑的自适应,为不同个体自适应选择不同类型的邻域拓扑;(2)从强化学习的角度来看,QLPSO 在算法的每次迭代中均使用了强化学习,但本文算法通过阈值来控制强化学习的使用频率,这有助于提供更加稳定的邻域信息.综合来看,在群智能优化算法中,基于强化学习的自适应多邻域的相关工作依旧非常少,因此本文工作有较好的创新性。

本文结构安排如下:第 2 节介绍一些代表性的相关工作,第 3 节详细介绍本文提出的 RLABC,并给出了算法的伪代码描述,第 4 节进行相关实验验证和分析,最后一节对本文工作进行总结和展望。

2 相关工作

2.1 经典 ABC

经典 ABC 模拟了蜂群的智能采蜜行为,根据蜜蜂在采蜜过程中的分工协作情况,ABC 包含了三类蜜蜂:雇佣蜂、观察蜂以及侦察蜂.这三种蜜蜂各司其职^[16]:(1)雇佣蜂负责勘探和记录蜜源,并将蜜源的位置和花蜜量等信息分享给观察蜂;(2)观察蜂根据收到的蜜源信息,会进一步选择较优的蜜源进行开采;(3)当某一蜜源连续多次未能提供更多花蜜时,该蜜源会被认定为开采殆尽,对应的雇佣蜂将转变为侦察蜂重新寻找新蜜源.需要注意的是,蜜源对应于优化问题的候选解,花蜜量代表候选解的质量,即适应度值.蜜源与雇佣蜂一一对应,且数量与雇佣蜂或观察蜂的数量相同。

与其他群智能优化算法类似,ABC 以一个初始种群开始寻优.假设种群有 SN 个蜜源, $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$ 表示第 i 个蜜源, $i \in \{1, 2, \dots, SN\}$, D 为优化问题的维度. X_i 可由下式完成初始化:

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

其中, $x_{i,j} \in [x_j^{min}, x_j^{max}]$, x_j^{max} 和 x_j^{min} 分别代表优化问题第 j 维的上下界, $rand(0,1)$ 是 $[0,1]$ 区间内均匀分布的随机数.在种群初始化之后,ABC 的三个搜索阶段开始循环迭代:雇佣蜂阶段、观察蜂阶段以及侦察蜂阶段,下面简介这三个阶段。

(1) 雇佣蜂阶段

在该阶段,雇佣蜂负责在整个搜索空间内搜索新蜜源,下式给出了搜索新蜜源所用的解搜索方程:

$$v_{i,j} = x_{i,j} + \phi_{i,j} \cdot (x_{i,j} - x_{k,j}) \quad (2)$$

其中, $V_i = (v_{i,1}, v_{i,2}, \dots, v_{i,D})$ 为新蜜源, X_i 为原蜜源; $\phi_{i,j}$ 是 $[-1,1]$ 范围内服从均匀分布的随机数; $X_k = \{x_{k,1}, x_{k,2}, \dots, x_{k,D}\}$ 是从种群中随机选出的蜜源,需满足 $X_k \neq X_i$; $j \in \{1, 2, \dots, D\}$ 是任意选中的一个维度,表明 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 会取代 X_i ,否则 X_i 保持不变.需指出的是,上述的解搜索方程是 ABC 生成新蜜源的主要方式。

(2) 观察蜂阶段

当所有雇佣蜂完成搜索后,观察蜂会采用轮盘赌机制选择优秀的蜜源做进一步开采,并用式(2)所示的解搜索方程产生新蜜源.蜜源的花蜜量(即适应度值)可用下式计算:

$$fit(X_i) = \begin{cases} \frac{1}{1 + f(X_i)}, & f(X_i) \geq 0 \\ 1 + abs(f(X_i)), & f(X_i) < 0 \end{cases} \quad (3)$$

其中, $fit(\cdot)$ 为适应度函数, $f(\cdot)$ 为目标函数, $abs(\cdot)$ 表示取绝对值.与雇佣蜂阶段不同,观察蜂阶段的目的是在优秀的蜜源周边进行精细搜索.因此,观察蜂的搜索行为与雇佣蜂不同.在一定程度上,观察蜂阶段起到了局部搜索的作用,而雇佣蜂阶段则对应全局搜索.下式给出了每个蜜源被观察蜂选中的概率:

$$p_i = \frac{fit(X_i)}{\sum_{j=1}^{SN} fit(X_j)} \quad (4)$$

其中, p_i 为 X_i 被选中的概率.不难看出,选择概率与蜜源的适应度值成正比,因此适应度值越好的蜜源,被选中的可能性越高,甚至可能会被多次选中。

(3) 侦察蜂阶段

不同于上述两个阶段,侦察蜂阶段目的并非是搜索新蜜源,而是防止某些蜜源占用过多计算资源而导致整个种群陷入搜索停滞.为此,每个蜜源都有一个相应的计数器 $trial$ 来记录其连续未被更新的次数.若蜜源在雇佣蜂阶段或观察蜂阶段有成功更新,则计数器重置为 0;否则,计数器加 1.若某一蜜

源的计数器值超过了预设的阈值 $limit$, 则该蜜源会被认定为开采殆尽, 进而被称作“被放弃的蜜源”, 对应的雇佣蜂将转变为侦察蜂用式(1)重新初始化该蜜源.

2.2 相关工作

近年来, 为提高 ABC 性能, 许多改进 ABC 相继被提出. 根据改进思路上的差异, 这些相关工作可大致分为以下三类, 下面简介其中的一些代表性工作.

(1) 如何改进解搜索方程

在经典 ABC 中, 解搜索方程因受随机个体 X_k 的影响, 导致其搜索方向和步长有很强的随机性, 使得 ABC 面临勘探能力过强, 而开采能力又不足的问题. 为此, 许多改进 ABC 引入最优个体或精英个体至解搜索方程中, 以增强 ABC 的开采能力, 并加快收敛速度. 例如, Zhu 和 Kwong^[11] 提出了一种基于种群最优个体引导的 ABC(GABC), 把最优个体作为解搜索方程中的一个新增项, 以利用最优个体所包含的有益信息指导算法朝更优的搜索方向前进. Aslan 等人^[17] 提出了一种改进的 $iqABC(iqABC)$, 它利用最优个体来指导搜索. 与 GABC 不同的是, $iqABC$ 通过一个计数器来控制最优个体的使用频率, 以避免算法陷入局部最优. 不同于 $iqABC$ 利用最优个体的方式, Zhou 等人^[18] 设计了一种高斯精简的解搜索方程来改进 ABC(GBABC), 利用高斯分布在当前个体和最优个体构成的搜索空间内抽样产生新个体. 类似地, Zhang 等人^[19] 提出了基于高斯分布和元胞邻域拓扑的改进 ABC(CGABC), 同样采用了一种基于高斯分布的解搜索方程, 并结合了局部吸引子信息, 以进一步提高算法的开采能力. 最近, Zhou 等人^[9] 提出了一种多精英引导的 ABC(MGABC), 利用多个精英个体来构建精英组, 并把该精英组引入到解搜索方程中以增强 ABC 的开采能力, 这种方式相比于直接利用最优个体可在一定程度上避免算法早熟.

(2) 如何利用邻域拓扑

上述改进 ABC 通过利用最优个体或精英个体有效提高了 ABC 的开采能力, 但应指出的是最优个体或精英个体一旦被过度利用, 容易导致算法的搜索行为过于贪婪, 从而引起早熟问题. 为此, 如何利用邻域拓扑来改进 ABC 是近年来的研究热点. 例如, Wang 等人^[13] 提出了一种基于环形邻域拓扑的改进 ABC(NSABC), 它根据个体间的索引关系为种群构建了环形邻域拓扑, 并选择邻域中的最优个体作为解搜索方程的搜索起点, 实验结果表明

NSABC 有非常好的性能. 不同于 NSABC 采用固定邻域半径的方式, Xiao 等人^[20] 提出了一种可动态调节邻域半径的改进 ABC(ABCNG), 它通过一个简单的规则来逐步地动态调整环形邻域拓扑的半径. 与 NSABC 和 ABCNG 仅采用单一邻域拓扑不同, Zhou 等人^[21] 设计了同时采用三种不同邻域拓扑的 ABC(ABC-MNT), 先把种群划分为三个子种群, 再为每个子种群选用一种邻域拓扑, 使得算法能综合不同邻域拓扑的优势.

(3) 如何与其他搜索技术相结合

除上述两类工作, 还有一些工作主要关注如何将 ABC 与其他搜索技术或进化算法相结合, 以实现不同搜索技术的优势互补. Kang 等人^[22] 在 ABC 中引入了 Rosenbrock 方法(RABC), 该方法是一种无导数的局部搜索技术, 可在一定程度上更好地平衡 ABC 的勘探和开采能力. Jadon 等人^[23] 将 ABC 与差分进化算法(DE)相结合(HABCDE), 把 DE 的变异算子和交叉算子融入到 ABC 的观察蜂阶段, 以利用 DE 开采能力强的特点来提升 ABC 的开采能力. Gao 等人^[24] 提出了一种基于信息学习的改进 ABC(ILABC), 它通过多种群技术的方式来提高 ABC 性能, 该多种群技术划分种群的方式类似于聚类算法, 可使不同的子种群专注于不同的子区域, 以提高找到全局最优解的可能性.

3 本文算法

3.1 主要动机

近年来, 如何利用邻域拓扑来改进 ABC 是相关工作中的研究热点. 相比于直接利用最优个体或精英个体的改进 ABC, 这些邻域相关的改进 ABC 在利用优秀个体的方式上有更好的灵活性, 因为邻域拓扑刻画了个体间的组织方式, 能控制信息散播的快慢程度. 然而, 需指出的是, 现有的邻域相关的改进 ABC 往往仅采用单一的邻域拓扑, 比如: 在前文中提到的 NSABC 和 ABCNG^[13,20], 它们均采用了环形邻域拓扑, 即将种群中所有个体按索引关系组织成环形结构. 虽然, 单一的邻域拓扑在算法结构上更加简洁, 但同时也忽视了不同邻域拓扑的优点, 使得算法性能还有一定局限性. 当前, 有个别算法开始从多邻域拓扑的角度来改进 ABC, 比如: 在前文中提到的 ABC-ANT 和 ABC-MNT^[14,21], 这两种算法都采用了三种不同类型的邻域拓扑, 它们比单一邻域拓扑的 ABC 有更好性能.

实际上,设计多邻域拓扑的 ABC 涉及两个关键点:(1)采用哪些邻域拓扑?(2)如何应用邻域拓扑?针对第一个关键点,通常宜选用可优势互补的邻域拓扑,比如:随机邻域拓扑的信息散播速度非常快,有很强的开采能力,而环形邻域拓扑的信息散播速度相对更慢,有较好的勘探能力,这两种类型的邻域拓扑就能优势互补,适合选用.针对第二个关键点,从 ABC 属于群智能优化算法的角度来看,可从种群或个体的层次来应用邻域拓扑.在第一个关键点上,现有的两种多邻域拓扑 ABC,即 ABC-ANT 和 ABC-MNT,都采用了可优势互补的三种邻域拓扑;但在第二个关键点上这两者采用了不同方式,ABC-MNT 是多子种群共存的方式,ABC-ANT 是种群自适应的方式.严格来说,这两种方式都是从种群层次来应用邻域拓扑,要求种群中所有个体或部分个体同时应用同一种邻域拓扑,然而,相比种群层次的粗粒度方式,个体层次可使种群中的不同个体能同时应用不同的邻域拓扑,有更好的灵活性.为此,本文在个体层次上设计多邻域拓扑的 ABC,并用强化学习的方式来确定选用何种邻域拓扑.

3.2 强化学习

强化学习是一种代表性的机器学习方法^[25,26],通过智能体与环境间的交互来自主学习最优的行为策略,目标是最大化智能体在环境中获得的总奖励,即长期收益.在强化学习中,智能体根据当前状态来选择动作并执行,而环境则根据智能体的动作给予即时奖励,再更新下一次状态.在智能体需选择下一次动作时,智能体会根据奖励和状态的变化来调整自身行为策略,以期在未来做出更优的选择.图 1 给出了强化学习的基本框架.

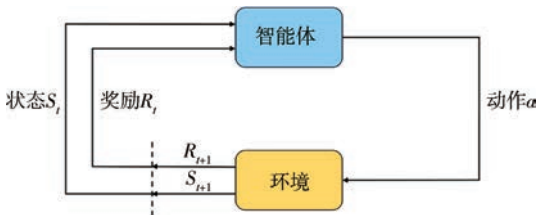


图 1 强化学习的基本框架

强化学习包含的算法有多种,本文采用其中最经典的 Q-learning 方法^[27].Q-learning 是一种基于价值函数 Q 的强化学习算法,通过 Q 表来记录每个“状态-动作”组合的价值,即预期的累积奖励^[28].假设智能体的可能状态有 p 种,能采取的动作有 q 种,可构建如表 1 中所示的 Q 表:

表 1 Q-learning 方法中的 Q 表示例

	动作			
	a_1	a_2	...	a_q
s_1	$Q(s_1, a_1)$	$Q(s_1, a_2)$...	$Q(s_1, a_q)$
状态 s_2	$Q(s_2, a_1)$	$Q(s_2, a_2)$...	$Q(s_2, a_q)$
\vdots	\vdots	\vdots	\vdots	\vdots
s_p	$Q(s_p, a_1)$	$Q(s_p, a_2)$...	$Q(s_p, a_q)$

其中元素 $Q(s_i, a_j)$ 表示智能体在状态 s_i 时选择动作 a_j 所获得的预期累积奖励.初始时, Q 表的元素通常设置为 0 或随机值.在智能体与环境交互后,智能体在每个时间步都会按当前环境状态查询 Q 表,选择有最高 Q 值的动作为下一步决策.随着与环境不断交互, Q 表会持续更新,因此智能体也能逐渐学习到最优的行为策略,并在未来的决策中做出更优选择.

Q-learning 方法的核心思想是^[29],在学习过程中通过计算每次环境状态下的期望奖励,来选择下一次环境状态应该采取的动作,使得期望奖励最大化.具体而言,首先算法根据当前状态 s_t 和动作 a_t 的价值函数 $Q(s_t, a_t)$,计算出在当前状态下采取的动作所能获得的期望奖励 r ;之后,算法根据期望奖励和目标奖励之间的差值来更新当前状态的 $Q'(s_t, a_t)$.具体更新公式如下式所示:

$$Q'(s_t, a_t) = (1 - \alpha) \cdot Q(s_t, a_t) + \alpha \cdot [r + \gamma \cdot \max_a Q(s_{t+1}, a)] \quad (5)$$

其中, $\alpha \in [0, 1]$ 表示学习率, $\gamma \in [0, 1]$ 为折扣因子, $Q(s_{t+1}, a)$ 是在下一个状态 s_{t+1} 下采取动作 a 的 Q 值.学习率 α 是控制算法更新 Q 值的速度和幅度的参数,表示新 Q 值对原 Q 值的影响程度^[30]. α 越小,算法越倾向于利用现有信息;而 α 越大,对原 Q 值的依赖就越小,算法越倾向于当前探索的信息.折扣因子 γ 控制了算法对未来奖励的重视程度,它表示未来奖励对当前 Q 值的贡献比例^[31]. γ 越大,算法越倾向于追求长期收益;而 γ 越小,意味着算法越倾向于追求即时奖励.在本文算法中,我们采用了常规的 α 和 γ 值,分别设置为 0.75 和 0.2.

在基本 Q-learning 方法中, $Q(s_{t+1}, a)$ 的选取方式采用了贪心策略(Greedy policy),即总是会选择使得下一个状态 s_{t+1} 的 Q 值最大的动作 a .这种方式使算法尽可能地选择最优动作,从而最大化期望回报;然而,也会存在一个明显的缺点,容易导致算法陷入局部最优.为此,我们采用了 ϵ -贪心策略(ϵ -greedy policy)^[29],会在一定概率下随机选择一个动作,而非总是选择最大 Q 值的动作,有助于增

加算法的多样性。 ϵ -贪心策略可形式化如下:

$$Q(s_t, a_t) = \begin{cases} \max_a Q(s_t, a), & \text{if } rand < \epsilon, \\ Q(s_t, a_{rand}), & \text{otherwise.} \end{cases} \quad (6)$$

其中, $rand$ 为 $[0, 1]$ 范围内的随机数, a_{rand} 是从动作集合 $\{a_1, a_2, \dots, a_q\}$ 中以概率 ϵ 随机选择的一个动作. 概率 ϵ 设为常规值 0.85.

3.3 自适应多邻域拓朴

如 3.1 节“动机”部分所述, 设计多邻域拓朴的 ABC 涉及了两个关键点, 它们对算法性能有着重要影响. 因此, 在 RLABC 的设计中, 我们充分考虑了这两点, 并针对性地做了相应设计. 对于第一个关键点, 采用了 4 种不同类型的邻域拓朴来构建候选池, 分别是: 小世界邻域拓朴、随机邻域拓朴、环形邻域拓朴以及元胞邻域拓朴, 这 4 种邻域拓朴各有特点, 能够实现优势互补. 图 2 给出了这 4 种邻域拓朴的示意图, 其中黄色圆点代表当前个体 X_i , 深蓝色圆点是 X_i 的邻居, 浅蓝色圆点为种群中的非邻居个体, 箭头代表个体间存在关联, 能够进行信息传递. 下面简介这 4 种邻域拓朴的特点:

(1) 小世界邻域拓朴. 该邻域拓朴同时结合了规则连接和随机连接两种方式, 有聚集性高、鲁棒性强等特点. 邻域中所有个体按个体间的索引关系组成环形结构, 但与环形邻域拓朴不同的是, 个体 X_i 的邻居不一定是其索引两侧的个体, 而是以一定概率

β 来决定. 本文中, β 值采用经典设置 $\beta = 0.05$, 邻域规模设置为 $0.5 \cdot SN$, 其中 SN 表示种群大小. 结合本文设置的邻域规模, 小世界邻域拓朴能表现出较快的信息传播速度, 有很强的开采能力.

(2) 随机邻域拓朴. 该邻域拓朴的特点是邻域内的个体间都相互存在关联, 且任意两个个体间的连接没有偏好, 连接概率相同. 邻域规模设置为 $0.3 \cdot SN$, 因此该邻域拓朴同样有较快的信息传播速度, 也能表现出较强的开采能力.

(3) 环形邻域拓朴. 在该邻域中, 个体按照索引关系, 顺序地构成一个环形结构. 然而, 与小世界邻域拓朴不同的是, 环形邻域中个体 X_i 的邻居只包括索引顺序上相邻的个体, 其邻居的选择范围与邻域半径密切相关. 因此, 与小世界邻域拓朴相比, 环形邻域拓朴的邻居范围在灵活性上更弱, 信息散播速度也相对更慢. 本文中, 邻域半径设为经典值 $0.1 \cdot SN$, 即个体 X_i 有 $0.2 \cdot SN$ 个邻居, 能表现出较为均衡的勘探和开采能力.

(4) 元胞邻域拓朴. 该邻域拓朴可理解为个体均分布在一个球面上, 但为方便可视化, 一般描述为平面的矩形结构, 邻域中个体间的邻居关系同样由个体的索引关系来决定. 邻域规模设置为 4, 即个体 X_i 的邻居为其上下左右四个方向上对应的索引相邻的个体. 与上述 3 种邻域拓朴相比, 元胞邻域拓朴的信息传播速度最慢, 有较强的勘探能力.

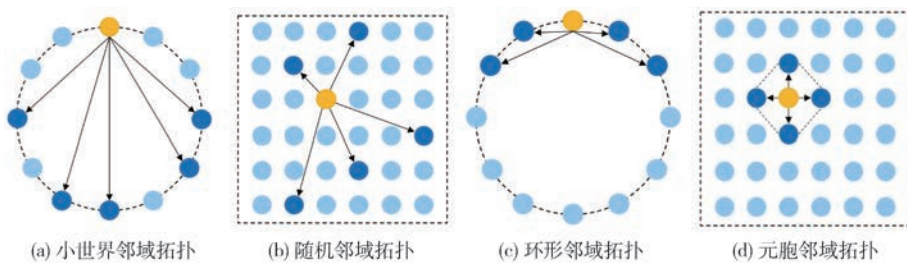


图 2 四种邻域拓朴示意图

不难看出, 上述 4 种邻域拓朴的特征各异, 搜索能力也各有千秋, 有助于在个体层次上实现高效的多邻域拓朴 ABC.

针对第二个关键点, 本文在个体层次上应用多邻域拓朴, 并采用强化学习中的 Q-learning 方法为每个个体自适应地更新邻域拓朴. 为此, 我们把个体视作智能体, 而搜索空间视作环境; 每个个体对应一张 Q 表, 用于记录在不同状态下执行不同动作的价值. 同时, 将邻域候选池中的 4 种邻域拓朴作为个体的可能状态, 把切换邻域定义为个体所能采取的动作.

当个体需要采取动作时, 即切换邻域拓朴, RLABC 会根据个体适应度的变化情况给予相应的奖励, 定义的奖励函数如下:

$$r = \begin{cases} 1, & \text{fit}(X_i^{t+1}) > \text{fit}(X_i^t) \\ 0, & \text{fit}(X_i^{t+1}) \leq \text{fit}(X_i^t) \end{cases} \quad (7)$$

其中, $fit(\cdot)$ 是适应度函数, $fit(X_i^t)$ 表示个体 X_i 在 t 时刻的适应度, $fit(X_i^{t+1})$ 为 $t+1$ 时刻的适应度. 从该式可看出, 若选用的邻域拓朴能促进个体更新, 那么给予奖励值 1; 反之, 若对个体更新不起作用, 那么不会进行奖励, 即奖励值为 0. 在个体获得

奖励后,再对个体的 Q 表进行更新.

在图 3 中给出了 Q 表更新的一个示例.假定个体 X_i 在 t 时刻处于状态 s_4 ,且会选择最优动作 a_1 ,即:个体 X_i 已应用了元胞邻域拓扑,且将要切换为小世界邻域拓扑,对应的 $Q(s_4, a_1) = 0.2677$.当进入 $t+1$ 时刻,可按式(5)所示的 Q 值更新公式来更新 X_i 对应的 Q 表,此时因个体切换为小世界邻域拓扑,对应的状态也更新为 s_1 ,并在该状态 s_1 下采用 ϵ -贪心策略选择了最优动作 s_3 .需说明的是,个体在切换邻域拓扑时不能保持现有邻域拓扑不变,需

	a_1	a_2	a_3	a_4		a_1	a_2	a_3	a_4	
s_1	0	0.4734	1.1588	0.2874	$t = t+1$	s_1	0	0.4734	1.1588	0.2874
s_2	0.2374	0	1.0405	0.2424		s_2	0.2374	0	1.0405	0.2424
s_3	0.2374	0.3049	0	0.2623		s_3	0.2374	0.3049	0	0.2623
s_4	0.2677	0.2212	0.0849	0		s_4	0.2407	0.2212	0.0849	0

图 3 Q 表更新示例

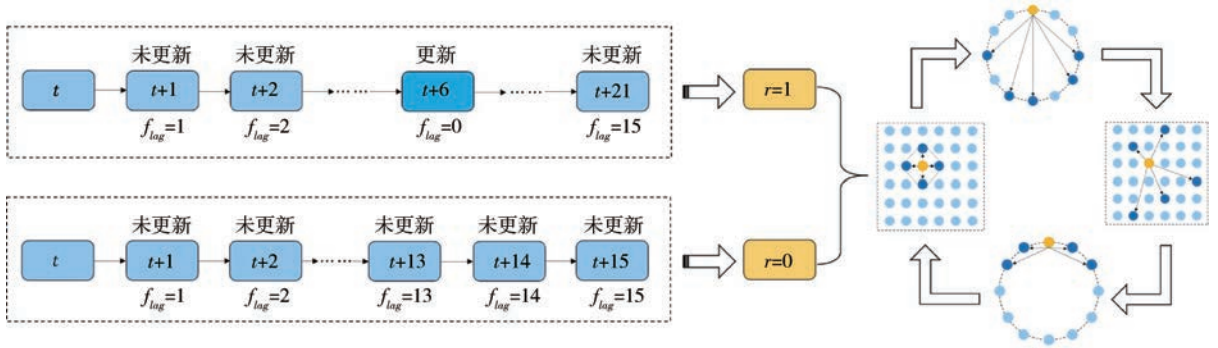


图 4 基于阈值的邻域拓扑切换过程

在经典的 Q-learning 方法中,算法每次迭代都需更新 Q 表并重新选择状态,这会使得智能体频繁地切换邻域拓扑.为此,我们定义了一个阈值 L 用于控制邻域拓扑的切换频率,主要原因是群智能优化算法本身有一定随机性,如果被选用的邻域拓扑仅保留使用一代,那么这种方式不见得会准确,应该给予一定的使用周期,以方便判断被选用的邻域拓扑是否真的有用.相应地,为每个个体设置了一个计数器 $flag$,用于判断是否达到了阈值 L .当个体未更新时 $flag$ 计数加 1,而当个体有更新时 $flag$ 重置为 0;若个体的 $flag$ 达到阈值 L ,说明个体当前选用的邻域拓扑不再适合,此时个体应切换邻域拓扑,即智能体应采取新动作.本文中,阈值 L 设置为 15,相关的参数敏感性分析在实验部分 4.2 节中给出了详细说明.

需注意的,若个体的适应度值在其关联的 $flag$ 达到阈值前存在过更新,说明被选用的邻域拓

切换至候选池中的其他任一邻域拓扑,这可避免同一邻域拓扑浪费计算资源,因此 Q 表的主对角线元素均为 0.

需进一步说明的是,从强化学习的角度来看,为智能体选用上述 4 种邻域拓扑的主要原因是,这些邻域拓扑的特征各异,例如:小世界邻域拓扑有较强的开采能力,而元胞邻域拓扑则有较强的勘探能力,因此当智能体采取动作,即切换不同的邻域拓扑时,不会存在冗余的动作,且能较好地涵盖智能体的可能动作.

扑曾起过作用,则按式(7)给予奖励值 1;反之,若未存在过更新,则说明被选用的邻域拓扑在 L 次迭代内一直未起过作用,那么按式(7)给予奖励值 0,即本质上不给予奖励.为方便理解该过程,图 4 给出基于阈值的邻域拓扑切换过程,可看出当个体的未更新次数达到阈值,即个体关联的计数器 $flag$ 等于 L ,那么个体需切换邻域拓扑.

为进一步提高 RLABC 性能,我们不再使用式(2)所示的原始解搜索方程,而采用了 Gao 等人^[32]提出的 ABC/best/1 策略,该策略相比于原始解搜索方程有更强的开采能力.需指出的是,本文的重点在于如何设计个体层次上的多邻域拓扑,而非设计新的解搜索方程,因此采用了现有的改进解搜索方程. ABC/best/1 策略如下:

$$v_{i,j} = x_{best,j} + \phi_{i,j} \cdot (x_{r_1,j} - x_{r_2,j}) \quad (8)$$

其中, X_{best} 是种群最优个体, X_{r_1} 和 X_{r_2} 是从种群中随机选取的两个个体,这些个体互不相同, $\phi_{i,j} \in [-1, 1]$,

1]. 考虑到 RLABC 中邻域拓扑的特点, 我们对 ABC/best/1 进行了微调, 将 X_{best} 替换为邻域内的最优个体 X_{nbest} , 微调后的解搜索方程如下:

$$v_{i,j} = x_{nbest,j} + \phi_{i,j} \cdot (x_{r1,j} - x_{r2,j}) \quad (9)$$

在观察蜂阶段, 考虑到观察蜂和雇佣蜂的区别, 我们进一步将式(9)所示的解搜索方程做了微调, 把随机个体 X_{r1} 也替换为邻域内的最优个体 X_{nbest} , 调整后的解搜索方程如下:

$$v_{i,j} = x_{nbest,j} + \phi_{i,j} \cdot (x_{nbest,j} - x_{r,j}) \quad (10)$$

因此, 观察蜂阶段的解搜索方程与雇佣蜂阶段略有不同, 能更好地满足观察蜂阶段对更强开采能力的需求。

在经典 ABC 的观察蜂阶段, 蜜源的适应度决定了其被观察蜂选中的概率, 适应度越好, 被选中的概率越大。然而, 我们的前期研究工作^[33]表明, 这种基于适应度来计算选择概率的方式容易受到某些极端食物源的干扰, 导致最终的选择结果出现很大偏差。为此, 在 RLABC 中, 我们采用基于适应度排序的方式来计算选择概率, 不再直接采用适应度本身。在该方式中, 将蜜源按适应度从好至差进行一次排序, 以确定每个食物源的排序值 $Sort_i$, 再按下式算出选择概率:

$$p_i = \frac{1/Sort_i}{\sum_{i=1}^{SN} (1/Sort_i)} \quad (11)$$

该方式能确保食物源的选择概率仅受排序值影响, 而不再受具体的适应度影响。

3.4 改进的侦察蜂阶段

在经典 ABC 的侦察蜂阶段, 如果一个食物源连续 $limit$ 次迭代后仍无法得到改善, 即食物源关联的计数器 $trial_i$ 达到了阈值 $limit$, 那么该食物源会被认定为开采殆尽, 将按公式(1)所示的重新初始化方法来产生新的食物源。该方法简单有效, 但易丢失已获得的搜索经验。实际上, 被放弃的食物源一般有较好质量, 当优化问题是多峰类型时, 被放弃的食物源甚至可能是某一局部最优解, 能够为后续的搜索过程提供一些有价值的信息, 在一定程度上可视为搜索经验。因此, 为保留这种搜索经验, 我们在 RLABC 中引入了双精英搜索策略来处理被放弃的食物源。该策略是由 Zhou 等人^[14]在 ABC-ANT 中提出的, 具体如下式所示:

$$v_{i,j} = \begin{cases} a \cdot x_{i,j} + b \cdot x_{e1,j} + c \cdot x_{e2,j} & \text{if case} \\ x_{i,j} & \text{otherwise} \end{cases} \quad (12)$$

其中, $case$ 代表条件: $rand(0,1) \leq CR$ or $j = j_{rand}$ 。 X_i 是被放弃的食物源, X_{e1} 和 X_{e2} 是从精英组中随机抽取的两个精英个体, 该精英组由种群中适应度最好的前 10% 个体构成。参数 CR 决定了 X_i 有多少决策变量能够传递给 V_i , 这也是搜索经验保存的一种形式。在 RLABC 中, CR 取值与 ABC-ANT 保持相同, 即 $CR=0.5$, 使得生成的新食物源既不会过于偏向 X_i , 也不会过于随机, 能在保留搜索经验和避免种群陷入局部最优之间达到较好平衡。符号 $j_{rand} \in \{1, 2, \dots, D\}$ 代表一个随机选定的维度, 以确保 V_i 至少在一个维度上与 X_i 不同。通过引入双精英搜索策略能进一步提高 RLABC 性能。关于这一策略对 RLABC 性能的影响, 我们在实验部分 4.3 节中给出了相关讨论。

3.5 算法流程图和伪代码

与经典 ABC 相比, 本文提出的 RLABC 主要包含两点改进之处: (1) 基于强化学习的自适应多邻域拓扑; (2) 利用双精英搜索策略改进侦察蜂阶段。为更清楚地描述 RLABC, 图 5 给出了算法的流程图, 可看出算法的整体框架是较为简洁的。同时, 我们在算法 1 中给出了 RLABC 的伪代码描述, 其中符号 FES 为已消耗的适应度函数评估次数, $MaxFES$ 为预设的最大适应度函数评估次数, 也是算法的终止条件。

算法 1. RLABC 算法伪代码。

输入: SN (食物源数量)、 $limit$ (侦察蜂阶段的控制参数)、 L (自适应多邻域拓扑的阈值)、 CR (双精英搜索策略的控制参数)、 $MaxFES$ (最大适应度函数评估次数)

输出: $gbest$ (种群最优个体)

1. 初始化种群和 Q 表, 为个体随机选择状态和动作;
2. 计算个体适应度, 令 $FES=SN, trial_i=0, flag_i=0$;
3. WHILE $FES \leq MaxFES$ DO
4. /* * 选择状态和动作, 并更新 Q 表 * */
5. FOR $i=1$ TO SN DO
6. IF $flag_i \geq L$ THEN
7. $state_{i+1} = action_i$;
8. 选择动作 $action_{i+1}$, 计算奖励 r , 更新 Q 表;
9. END IF
10. END FOR
11. /* * 雇佣蜂阶段 * */
12. FOR $i=1$ TO SN DO
13. 按式(9)为个体 X_i 生成新食物源 V_i ;
14. IF $fit(V_i) \geq fit(X_i)$ THEN
15. 令 $X_i = V_i, trial_i = 0, flag_i = 0, FES++$;
16. ELSE
17. 令 $trial_i++, flag_i++, FES++$;


```

18.   END IF
19. END FOR
20. /* * 观察蜂阶段 * */
21. 按式(11)计算每个个体的选择概率
22. FOR  $i=1$  TO SN DO
23.   通过轮盘赌方法选择个体  $X_j$ , 生成新食物源  $V_j$ ;
24.   IF  $fit(V_j) \geq fit(X_j)$  THEN
25.     令  $X_j = V_j, trial_j = 0, flag_j = 0, FE_s++$ ;
26.   ELSE
27.     令  $trial_j++$ ,  $flag_j++$ ,  $FE_s++$ ;
28.   END IF
29. END FOR
30. /* * 侦察蜂阶段 * */
31. FOR  $i=1$  TO SN DO
32.   IF  $trial_i \geq limit$  THEN
33.     按式(12)生成新食物源  $V_i$ ;
34.     令  $X_i = V_i, trial_i = 0, flag_i = 0, FE_s++$ ;
35.   END IF
36. END FOR
37. END WHILE

```

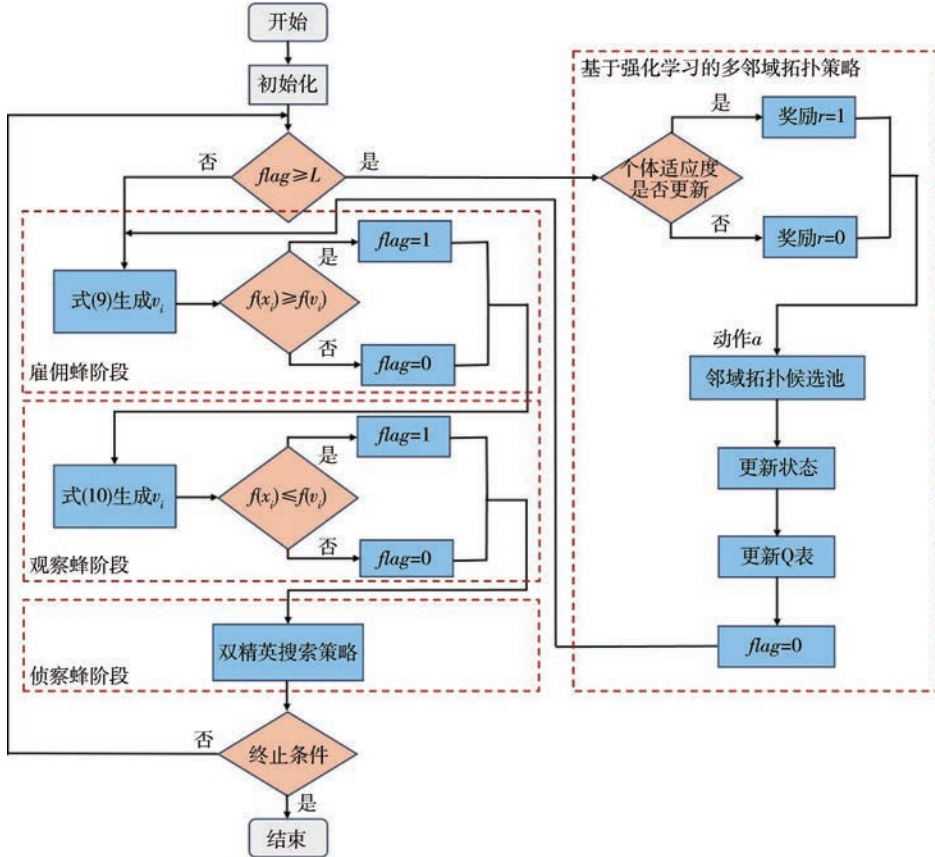


图5 算法流程图

4 实验验证

为验证 RLABC 性能,我们在 CEC2013^[34] 和 CEC2017^[35] 两套测试集以及两个实际优化问题上进行了大量实验,主要包括以下 5 个部分:

- (1) 参数敏感性分析;
- (2) 策略有效性验证;
- (3) 与 4 种邻域相关 ABC 对比;
- (4) 与 4 种其他知名改进 ABC 对比;
- (5) 在实际优化问题上对比。

需说明的是,针对“参数敏感性分析”和“策略有

效性验证”两部分,仅在 CEC2013 测试集上进行相关实验. 主要原因是 CEC2013 测试集包含了 20 个单独的问题,而 CEC2017 只有 9 个. 对单独的问题而言,其结构特征比混合问题和组合问题更清晰,更适合用于验证算法的策略有效性和设置算法的参数值。

4.1 测试函数和参数设置

CEC2013 测试集有 28 个测试问题^[34], 包括了 3 种类型的问题: 单峰问题 (F01-F05)、多峰问题 (F06-F20) 以及组合问题 (F21-F28); CEC2017 测试集有 29 个测试问题^[35], 分为 4 种类型: 单峰问题 (F01-F02)、多峰问题 (F03-F09)、混合问题 (F10-F19) 以及组合问题 (F20-F29)。这两套测试集的所

有测试问题都有相同的定义域: $[-100, 100]$, 具体的问题定义可参考文献[34-35]. 实验中, 算法的停机条件均为 $MaxFEs = 10000 \cdot D$. RLABC 的参数设置为: 种群规模 $SN = 60$, 控制参数 $limit = 100$. 为消除随机因素对结果的影响, 所有算法在每个测试问题上独立运行 30 次, 以平均值作为算法的最终结果.

此外, 为确保实验结果对比有统计意义, 我们采用了两种非参数检验方法: Wilcoxon 秩和检验和 Friedman 检验, 显著性水平均设置为 $\alpha = 0.05$. 这两种统计检验方法有不同的使用目的, Wilcoxon 秩和检验用于评估 RLABC 与对比算法在单个测试问题上的性能是否存在差异, 并用符号“+”、“=”、“-”表示 RLABC 性能要好于、类似于、差于对比算法; 而 Friedman 检验用于评估算法在测试集上的整体性能, 通过算法的平均排名情况来体现, 排名值越小说明整体性能越好.

4.2 参数敏感性分析

4.2.1 阈值 L 的敏感性分析

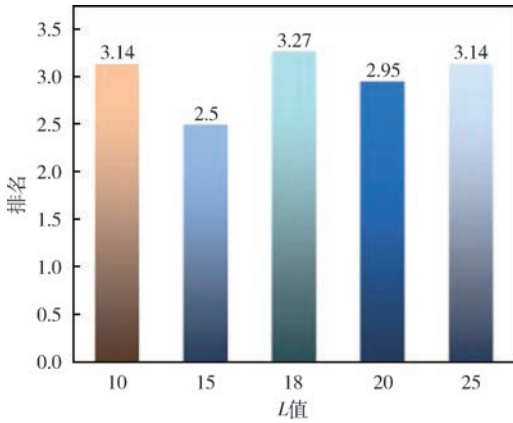
在 RLABC 中, 阈值 L 控制了邻域拓扑的切换

频率, 对算法性能有重要影响. 因此, 有必要对 L 进行敏感性分析, 以确定最优取值. 一般而言, 如果 L 过小, 那么 Q 表更新会很频繁, 使得个体不断地切换邻域拓扑, 导致无法确定现有邻域拓扑是否适应于个体的当前状态; 反之, 若 L 过大, Q 表会缓慢更新, 削弱了多邻域拓扑对 RLABC 的改进效果. 为此, 我们在 CEC2013 测试集上对 5 种代表性的 L 取值进行测试, 分别是: $L = 10, 15, 18, 20$ 和 25 , 测试函数的维度 $D = 30$.

表 2 给出了不同 L 取值的实验结果, 其中最好结果用粗体凸显, 表格的最后一行给出了每种取值能获得的最好结果个数. 从表中可看出, $L = 15$ 是最佳取值, 能在 14 个函数上取得最好结果; 其次为 $L = 20$, 在 11 个函数上取得了最好结果; 而 $L = 10, 18, 25$ 的表现则明显差于前两种取值, 分别只在 8 个、6 个和 8 个函数上取得最好结果. 图 6 给出了这 5 种取值通过 Friedman 检验得到的排名情况, 也可看出 $L = 15$ 的平均排名最好. 因此, 本文选择 $L = 15$ 作为阈值.

表 2 RLABC 采用不同 L 取值的实验结果

函数	$L=10$	$L=15$	$L=18$	$L=20$	$L=25$
F01	0.00E+00±0.00E+00	0.00E+00±0.00E+00	0.00E+00±0.00E+00	7.58E-15±4.08E-14	0.00E+00±0.00E+00
F02	4.16E+06±1.11E+06	3.77E+06±1.27E+06	3.94E+06±1.16E+06	3.94E+06±1.08E+06	3.47E+06±1.24E+06
F03	1.60E+07±1.06E+07	1.43E+07±9.61E+06	1.57E+07±9.94E+06	1.57E+07±9.80E+06	1.68E+07±1.28E+07
F04	2.58E+04±4.36E+03	2.41E+04±3.24E+03	2.52E+04±3.23E+03	2.60E+04±4.62E+03	2.51E+04±3.90E+03
F05	1.10E-13±2.04E-14	1.10E-13±2.04E-14	1.14E-13±0.00E+00	1.10E-13±2.04E-14	1.14E-13±0.00E+00
F06	1.65E+01±2.66E+00	1.61E+01±3.45E+00	1.63E+01±4.32E-01	1.70E+01±2.67E+00	1.70E+01±3.86E+00
F07	4.64E+01±7.77E+00	4.50E+01±7.70E+00	4.65E+01±6.15E+00	4.29E+01±7.68E+00	4.40E+01±7.73E+00
F08	2.10E+01±3.42E-02	2.09E+01±5.54E-02	2.10E+01±3.84E-02	2.09E+01±4.99E-02	2.10E+01±5.41E-02
F09	2.53E+01±2.52E+00	2.46E+01±2.95E+00	2.57E+01±2.12E+00	2.56E+01±2.34E+00	2.55E+01±2.56E+00
F10	7.38E-01±2.72E-01	6.75E-01±2.85E-01	7.08E-01±2.70E-01	7.70E-01±3.07E-01	6.25E-01±2.61E-01
F11	0.00E+00±0.00E+00	0.00E+00±0.00E+00	3.32E-02±1.79E-01	0.00E+00±0.00E+00	3.79E-15±1.42E-14
F12	3.75E+01±6.87E+00	3.95E+01±8.18E+00	3.98E+01±6.73E+00	3.98E+01±9.22E+00	3.93E+01±6.36E+00
F13	6.55E+01±1.23E+01	6.52E+01±1.88E+01	6.56E+01±1.34E+01	6.59E+01±1.45E+01	6.46E+01±1.34E+01
F14	2.42E+00±2.59E+00	3.15E+00±2.95E+00	3.16E+00±2.98E+00	2.14E+00±1.98E+00	2.47E+00±2.69E+00
F15	3.04E+03±3.58E+02	3.00E+03±3.16E+02	3.10E+03±2.49E+02	3.02E+03±2.26E+02	3.12E+03±2.47E+02
F16	1.09E+00±1.82E-01	1.07E+00±1.64E-01	1.09E+00±1.72E-01	1.10E+00±1.38E-01	1.11E+00±1.53E-01
F17	3.05E+01±4.22E-02	3.05E+01±5.54E-02	3.05E+01±4.19E-02	3.05E+01±4.04E-02	3.05E+01±3.42E-02
F18	7.04E+01±7.24E+00	6.92E+01±7.16E+00	6.95E+01±6.36E+00	6.94E+01±7.11E+00	6.91E+01±6.60E+00
F19	1.87E-01±1.55E-01	1.59E-01±1.37E-01	2.09E-01±2.13E-01	2.05E-01±1.62E-01	2.50E-01±2.29E-01
F20	1.16E+01±2.00E+00	1.22E+01±2.24E+00	1.21E+01±2.15E+00	1.17E+01±2.03E+00	1.18E+01±2.06E+00
F21	2.87E+02±7.58E+01	2.76E+02±6.40E+01	2.91E+02±7.36E+01	2.58E+02±6.02E+01	3.02E+02±8.24E+01
F22	1.60E+02±4.43E+01	1.49E+02±4.27E+01	1.52E+02±4.49E+01	1.47E+02±4.49E+01	1.47E+02±4.39E+01
F23	3.34E+03±4.75E+02	3.46E+03±3.70E+02	3.45E+03±3.78E+02	3.71E+03±4.12E+02	3.58E+03±4.70E+02
F24	2.29E+02±6.39E+00	2.27E+02±7.01E+00	2.27E+02±5.21E+00	2.27E+02±4.74E+00	2.28E+02±7.15E+00
F25	2.83E+02±7.91E+00	2.82E+02±6.55E+00	2.79E+02±6.65E+00	2.82E+02±8.67E+00	2.83E+02±7.48E+00
F26	2.00E+02±3.91E-02	2.00E+02±0.00E+00	2.00E+02±4.01E-02	2.00E+02±3.83E-02	2.00E+02±5.20E-02
F27	4.76E+02±1.07E+02	4.62E+02±1.09E+02	4.29E+02±6.16E+01	4.87E+02±1.37E+02	4.32E+02±8.21E+01
F28	3.00E+02±1.28E-13	2.95E+02±2.73E+01	3.00E+02±1.49E-13	2.89E+02±4.29E+01	2.93E+02±3.59E+01
个数	8	14	6	11	8

图6 阈值 L 不同取值的平均排名情况

4.2.2 种群规模 SN 的敏感性分析

在本文提出的多邻域拓扑策略中,是在个体层次上采用强化学习来选择不同的邻域拓扑,因此种群中个体数量对算法性能同样有重要影响.为此,在

本节中对种群规模 SN 进行敏感性分析,讨论不同种群规模对算法性能的影响,并确定最佳取值.实验中,我们在 CEC2013 测试集上讨论分析 4 种不同的取值,分别是: $SN=30、60、100、200$,测试函数维度 $D=30$.

在表 3 中给出了上述 4 种不同取值的实验结果,表的最后一行汇总了每种取值获得的最好结果个数.从对比结果可看出,较小的取值能使 RLABC 有更好性能, SN 为 30 和 60 时可在 13 个函数上取得最好结果;相反, SN 为 100 和 200 时仅分别在 5 个和 8 个函数上取得最好结果.产生这种情况的原因主要有两方面:(1)本文算法的停机条件是适应度函数的评估次数,较大的种群规模会导致算法迭代次数减少,使得算法的收敛精度降低,该原因同样适用于其他类型 ABC,从其他类型 ABC 的种群规模很少有大于 100 的情况也可佐证该原因;(2)本文算

表 3 RLABC 采用不同 SN 取值的实验结果

函数	$SN=30$	$SN=60$	$SN=100$	$SN=200$
F01	1.06E-13±1.13E-13	0.00E+00±0.00E+00	0.00E+00±0.00E+00	0.00E+00±0.00E+00
F02	2.36E+06±8.39E+05	3.77E+06±1.27E+06	5.34E+06±1.59E+06	7.53E+06±1.43E+06
F03	4.21E+06±4.50E+06	1.43E+07±9.61E+06	2.68E+07±1.50E+07	6.24E+07±4.20E+07
F04	2.76E+04±6.47E+03	2.41E+04±3.24E+03	2.43E+04±3.53E+03	2.77E+04±3.22E+03
F05	1.67E-13±5.67E-14	1.10E-13±2.04E-14	8.72E-14±4.81E-14	1.14E-13±0.00E+00
F06	1.62E+01±2.77E+00	1.61E+01±3.45E+00	1.71E+01±2.28E+00	1.80E+01±2.80E+00
F07	5.14E+01±1.10E+01	4.50E+01±7.70E+00	4.97E+01±5.30E+00	5.98E+01±4.81E+00
F08	2.09E+01±5.51E-02	2.09E+01±5.54E-02	2.10E+01±4.68E-02	2.09E+01±4.32E-02
F09	2.01E+01±3.83E+00	2.46E+01±2.95E+00	2.72E+01±2.12E+00	2.69E+01±1.21E+00
F10	9.81E-02±4.88E-02	6.75E-01±2.85E-01	1.20E+00±1.54E-01	1.36E+00±1.59E-01
F11	3.32E-01±5.35E-01	0.00E+00±0.00E+00	0.00E+00±0.00E+00	0.00E+00±0.00E+00
F12	3.81E+01±8.08E+00	3.95E+01±8.18E+00	4.37E+01±8.79E+00	5.34E+01±1.13E+01
F13	6.13E+01±1.31E+01	6.52E+01±1.88E+01	6.98E+01±1.55E+01	8.95E+01±1.28E+01
F14	4.36E+01±5.47E+01	3.15E+00±2.95E+00	8.22E-01±1.17E+00	2.21E-01±3.29E-01
F15	2.82E+03±3.60E+02	3.00E+03±3.16E+02	3.19E+03±3.07E+02	3.50E+03±3.37E+02
F16	7.69E-01±1.27E-01	1.07E+00±1.64E-01	1.33E+00±1.68E-01	1.47E+00±2.14E-01
F17	3.07E+01±2.47E-01	3.05E+01±5.54E-02	3.05E+01±1.52E-02	3.05E+01±1.88E-03
F18	6.25E+01±6.89E+00	6.92E+01±7.16E+00	8.45E+01±6.67E+00	9.73E+01±1.11E+01
F19	1.15E-01±1.36E-01	1.59E-01±1.37E-01	2.59E-01±1.89E-01	3.74E-01±1.70E-01
F20	1.24E+01±2.06E+00	1.22E+01±2.24E+00	1.23E+01±2.12E+00	1.23E+01±1.56E+00
F21	2.84E+02±6.94E+01	2.76E+02±6.40E+01	3.02E+02±6.67E+01	3.06E+02±8.60E+01
F22	1.74E+02±7.55E+01	1.49E+02±4.27E+01	1.50E+02±4.49E+01	1.55E+02±4.95E+01
F23	3.27E+03±3.99E+02	3.46E+03±3.70E+02	3.77E+03±4.03E+02	3.86E+03±4.38E+02
F24	2.29E+02±7.14E+00	2.27E+02±7.01E+00	2.33E+02±6.16E+00	2.45E+02±5.54E+00
F25	2.84E+02±6.61E+00	2.82E+02±6.55E+00	2.89E+02±3.22E+00	2.91E+02±4.44E+00
F26	2.00E+02±2.97E-02	2.00E+02±0.00E+00	2.00E+02±4.69E-02	2.00E+02±7.12E-02
F27	5.25E+02±1.26E+02	4.62E+02±1.09E+02	4.48E+02±1.07E+02	4.01E+02±7.73E-01
F28	2.93E+02±3.59E+01	2.95E+02±2.73E+01	3.00E+02±1.74E-13	2.88E+02±4.45E+01
个数	13	13	5	8

法采用的解搜索方程涉及到了邻域内的最优个体,如果种群规模过小,意味着邻域规模也同样小,使得邻域内的最优个体在一定程度上与普通个体的差距不明显,无法为算法搜索提供更加有价值的邻域信息.综合来看,种群规模应适中,不宜过大,也不能过小,所以本文对种群规模的设置为 $SN=60$.同时,从 SN 为 30 和 60 的结果可看出,在两者均未取得最优结果的 4 个函数上, $SN=60$ 可在其中的 3 个函数上要优于 $SN=30$,这也可说明 $SN=60$ 更加合适.在图 7 中给出了这 4 种取值通过 Friedman 检验得到的排名情况,可看出 $SN=60$ 排在第一位.

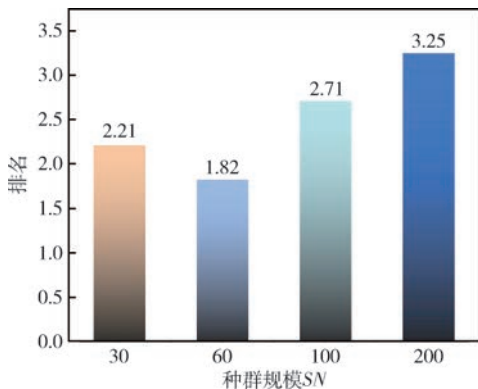


图 7 种群规模 SN 不同取值的平均排名情况

4.3 策略有效性验证

4.3.1 多邻域拓扑策略验证

为验证本文提出的多邻域拓扑策略的有效性,我们设计了两组对比实验:(1)与单邻域拓扑对比,检验多邻域拓扑的有效性;(2)与其他类型的多邻域拓扑对比,检验强化学习方式的有效性.下面介绍这

两组实验.

在第 1 组实验中,以 RLABC 涉及的 4 种邻域拓扑设计了以下对比算法:

- (1) RLABC-1:基于小世界邻域拓扑的 RLABC
- (2) RLABC-2:基于随机邻域拓扑的 RLABC
- (3) RLABC-3:基于环形邻域拓扑的 RLABC
- (4) RLABC-4:基于元胞邻域拓扑的 RLABC

上述 4 种对比算法均采用了单邻域拓扑,即只保留 RLABC 中的一种邻域拓扑,同时删除了基于强化学习的自适应多邻域拓扑部分,算法的其余部分与 RLABC 完全相同,因此可验证多邻域拓扑策略的有效性.在 CEC2013 测试集上进行实验,测试函数维度为 $D=30$.表 4 给出了 RLABC 与上述 4 种对比算法的实验结果,最好结果用粗体凸显.

从表 4 的 Wilcoxon 秩和检验结果可看出,RLABC 在所有算法中表现最好,说明多邻域拓扑策略是有效的.具体而言,与 RLABC-1 和 RLABC-2 这两种有较强开采能力的单邻域拓扑算法相比,RLABC 分别在 9 个和 8 个测试函数上取得了最好结果,仅在 F10 上略差于 RLABC-1.与 RLABC-3 和 RLABC-4 这两种有较强勘探能力的单邻域拓扑算法相比,RLABC 仍有较大优势,分别在 7 个和 9 个测试函数上取得了最好结果,而 RLABC-3 和 RLABC-4 仅在 F14 和 F17 上优于 RLABC.总之,与这 4 种单邻域拓扑的对比算法相比,实验结果表明了 RLABC 的多邻域拓扑策略比单邻域拓扑更加有效.图 8 给出了通过 Friedman 检验得到的 5 种算法的平均排名情况,可看出 RLABC 排名最好,其次为 RLABC-3.

表 4 与单邻域拓扑对比的实验结果

函数	RLABC-1	RLABC-2	RLABC-3	RLABC-4	RLABC
F01	7.58E-15±3.48E-14+	3.12E-14±4.22E-14+	0.00E+00±0.00E+00=	0.00E+00±0.00E+00=	0.00E+00±0.00E+00
F02	3.84E+06±2.39E+06=	4.10E+06±1.12E+06+	4.35E+06±3.24E+06+	4.75E+06±2.78E+06+	3.77E+06±1.27E+06
F03	1.53E+07±1.01E+07=	1.45E+07±1.02E+07=	2.31E+07±1.17E+07+	2.76E+07±1.03E+07+	1.43E+07±9.61E+06
F04	2.52E+04±3.08E+03=	2.51E+04±4.20E+03=	2.63E+04±5.22E+03=	3.11E+04±4.91E+03=	2.41E+04±3.24E+03
F05	1.14E-13±0.00E+00=	1.10E-13±2.04E-14=	1.14E-13±0.00E+00=	1.14E-13±0.00E+00=	1.10E-13±2.04E-14
F06	1.65E+01±3.21E+00=	1.78E+01±4.54E+00=	1.68E+01±3.22E+00=	1.63E+01±4.44E+00=	1.61E+01±3.45E+00
F07	3.95E+01±5.47E+00=	4.36E+01±5.95E+00=	4.42E+01±7.37E+00=	5.21E+01±7.75E+00+	4.50E+01±7.70E+00
F08	2.09E+01±4.87E-02=	2.09E+01±5.74E-02=	2.09E+01±5.28E-02=	2.10E+01±4.47E-02=	2.09E+01±5.54E-02
F09	2.65E+01±3.34E+00+	2.48E+01±2.93E+00=	2.67E+01±2.27E+00+	2.69E+01±2.02E+00+	2.46E+01±2.95E+00
F10	5.21E-01±2.85E-01-	5.96E-01±2.17E-01=	7.25E-01±2.63E-01+	8.27E-01±2.15E-01+	6.75E-01±2.85E-01
F11	3.79E-15±1.42E-14+	6.63E-02±2.48E-01+	0.00E+00±0.00E+00=	0.00E+00±0.00E+00=	0.00E+00±0.00E+00
F12	3.64E+01±7.03E+00=	3.45E+01±6.71E+00=	3.97E+01±9.62E+00=	4.40E+01±1.03E+01=	3.95E+01±8.18E+00
F13	6.00E+01±1.52E+01=	5.92E+01±1.31E+01=	6.06E+01±1.09E+01=	7.62E+01±1.27E+01+	6.52E+01±1.88E+01
F14	9.71E+00±2.24E+01+	6.40E+00±2.48E+00+	2.12E+00±2.56E+00+	1.34E+00±1.31E+00-	3.15E+00±2.95E+00
F15	2.90E+03±3.50E+02=	2.97E+03±3.07E+02=	2.97E+03±2.66E+02=	3.04E+03±3.20E+02=	3.00E+03±3.16E+02
F16	1.19E+00±1.25E-01+	1.19E+00±1.56E-01+	1.11E+00±1.72E-01=	1.09E+00±1.76E-01=	1.07E+00±1.64E-01

(续表)

函数	RLABC-1	RLABC-2	RLABC-3	RLABC-4	RLABC
F17	3.05E+01±6.94E-02=	3.05E+01±4.17E-02=	3.04E+01±1.09E-02-	3.04E+01±1.73E-03-	3.05E+01±5.54E-02
F18	7.12E+01±6.30E+00=	7.03E+01±7.17E+00=	7.05E+01±5.62E+00=	7.75E+01±8.13E+00+	6.92E+01±7.16E+00
F19	2.43E-01±1.67E-01+	1.71E-01±1.23E-01=	1.84E-01±1.85E-01=	1.63E-01±1.05E-01=	1.59E-01±1.37E-01
F20	1.15E+01±2.23E+00=	1.22E+01±2.28E+00=	1.16E+01±1.84E+00=	1.25E+01±2.19E+00=	1.22E+01±2.24E+00
F21	2.78E+02±7.23E+01=	2.99E+02±8.30E+01=	2.76E+02±8.06E+01=	2.89E+02±6.33E+01=	2.76E+02±6.40E+01
F22	1.65E+02±4.53E+01+	1.73E+02±4.57E+01+	1.64E+02±4.15E+01+	1.57E+02±4.45E+01=	1.49E+02±4.27E+01
F23	3.41E+03±4.10E+02=	3.26E+03±3.57E+02=	3.35E+03±4.02E+02=	3.89E+03±4.18E+02+	3.46E+03±3.70E+02
F24	2.28E+02±6.81E+00=	2.28E+02±7.51E+00=	2.28E+02±5.08E+00=	2.31E+02±5.63E+00=	2.27E+02±7.01E+00
F25	2.96E+02±7.60E+00+	2.98E+02±7.77E+00+	2.82E+02±8.05E+00=	2.85E+02±4.90E+00=	2.82E+02±6.55E+00
F26	2.00E+02±3.85E-02+	2.00E+02±4.38E-02+	2.00E+02±4.48E-02+	2.00E+02±4.43E-02+	2.00E+02±0.00E+00
F27	4.78E+02±1.11E+02=	4.76E+02±1.00E+02=	4.80E+02±9.44E+01=	4.84E+02±1.27E+02=	4.62E+02±1.09E+02
F28	3.00E+02±1.14E-13=	3.00E+02±1.28E-13=	2.93E+02±3.59E+01=	2.96E+02±2.03E+01=	2.95E+02±2.73E+01
+ / = / -	9/18/1	8/20/0	7/20/1	9/17/2	--

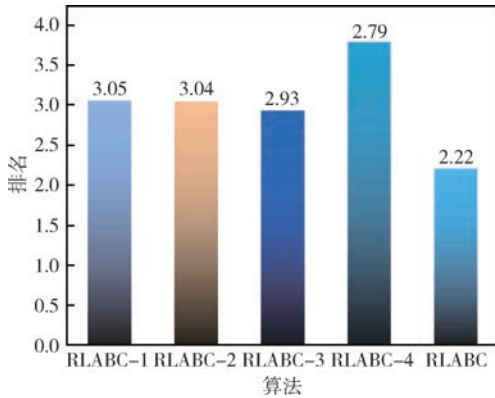


图 8 多邻域拓扑对比单邻域拓扑的平均排名情况

在第 2 组实验中,结合本文提出的多邻域拓扑策略的特点,设计了以下 3 种类型的多邻域拓扑对比算法:

- (1) RLABC-5:基于随机选择的多邻域 RLABC
- (2) RLABC-6:基于历史经验的自适应多邻域 RLABC
- (3) RLABC-7:采用 3 种邻域拓扑的 RLABC

在 RLABC-5 中,不再采用强化学习的方式,而是以随机方式为个体从候选池选择一个邻域拓扑.类似地,在 RLABC-6 中,构造了一种基于历史经验的方式来自适应选择多邻域,该方式与经典的 SaDE 算法^[36]相同,主要步骤是:(1)算法初始时,以随机方式为个体选用邻域拓扑;(2)设定阈值 L 作为算法的学习周期,记录每种邻域拓扑在一个周期内的成功率,即生成的后代个体能进入下一代的成功比率;(3)在下一个周期内,用该成功率作为邻域拓扑的选择概率,以此实现自适应选择.不同于 RLABC-5 和 RLABC-6,RLABC-7 继续保留了强化学习的方式,但只采用 3 种邻域拓扑:随机邻域、

环形邻域以及元胞邻域,这 3 种邻域拓扑也在前文提到的 ABC-ANT^[14]中采用了,因此可通过 RLABC-7 验证增加的第 4 种邻域拓扑,即小世界邻域,对算法性能的提高是否有贡献.

测试函数的维度 $D=30$,表 5 给出了 RLABC 与上述 3 种对比算法在 CEC2013 测试集上的实验结果,最好结果用粗体凸显.可看出 RLABC 在 17 个函数上获得了最好结果,是所有对比算法中情况最好的,这表明本文提出的基于强化学习的多邻域拓扑策略比其他类型的多邻域拓扑策略更好.从 Wilcoxon 秩和检验结果可看出,与 RLABC-5 和 RLABC-6 相比,RLABC 分别在 11 个和 8 个函数上显著更优;更重要的是,RLABC 未在任一函数上要差于 RLABC-5,且仅在 1 个函数上略差于 RLABC-6,这有效验证了强化学习方式的有效性.与 RLABC-7 相比,RLABC 同样未在任一函数上更差,且在 3 个函数上更优,这表明增加的第 4 种邻域拓扑对算法性能的提高是有贡献的.图 9 给出了通过 Friedman 检验得到的 4 种算法的平均排名,可看出 RLABC 排名最好,其次是 RLABC-6、RLABC-5 和 RLABC-7.

此外,我们在图 10 中给出了 RLABC 在 3 个代表性函数(单峰函数 F04、多峰函数 F08 以及组合函数 F23)上的各邻域拓扑使用频率.从图中可看出,在处理单峰函数时,开采能力较强的小世界邻域拓扑和随机邻域拓扑的使用频率更高;而对于多峰问题和混合问题,勘探能力较强的环形邻域拓扑和元胞邻域拓扑的使用频率则更高.此外,无论在何种测试函数上,均未出现某一邻域拓扑的使用频率过低的情况,这也进一步验证了基于强化学习的自适应多邻域拓扑策略的有效性.

表 5 与其他多邻域拓扑对比的实验结果

函数	RLABC-5	RLABC-6	RLABC-7	RLABC
F01	0.00E+00±0.00E+00=	0.00E+00±0.00E+00=	1.52E-14±5.67E-14=	0.00E+00±0.00E+00
F02	4.40E+06±7.49E+05+	5.56E+06±1.44E+06+	4.34E+06±1.21E+06+	3.77E+06±1.27E+06
F03	2.37E+07±1.64E+07+	3.51E+07±1.97E+07+	1.63E+07±1.32E+07=	1.43E+07±9.61E+06
F04	2.41E+04±4.68E+03=	1.51E+04±4.19E+03-	2.48E+04±5.33E+03=	2.41E+04±3.24E+03
F05	1.14E-13±0.00E+00=	1.02E-13±3.41E-14=	1.10E-13±2.04E-14=	1.10E-13±2.04E-14
F06	1.63E+01±2.59E+00=	1.63E+01±1.58E+00=	1.57E+01±2.34E+00=	1.61E+01±3.45E+00
F07	4.54E+01±6.11E+00=	4.64E+01±7.42E+00=	4.67E+01±6.31E+00=	4.50E+01±7.70E+00
F08	2.09E+01±5.31E-02=	2.09E+01±5.24E-02=	2.09E+01±5.68E-02=	2.09E+01±5.54E-02
F09	2.62E+01±3.25E+00+	2.47E+01±2.90E+00=	2.64E+01±1.54E+00+	2.46E+01±2.95E+00
F10	8.44E-01±2.51E-01+	1.42E+00±1.82E-01+	7.81E-01±2.85E-01+	6.75E-01±2.85E-01
F11	0.00E+00±0.00E+00=	0.00E+00±0.00E+00=	3.32E-02±1.79E-01=	0.00E+00±0.00E+00
F12	3.87E+01±9.23E+00=	4.05E+01±7.34E+00=	3.98E+01±6.04E+00=	3.95E+01±8.18E+00
F13	7.35E+01±1.33E+01+	6.74E+01±1.51E+01=	6.64E+01±1.47E+01=	6.52E+01±1.88E+01
F14	3.01E+00±2.50E+00=	1.99E+00±2.47E+00=	2.29E+00±2.10E+00=	3.15E+00±2.95E+00
F15	3.20E+03±4.01E+02+	3.20E+03±3.02E+02+	3.07E+03±3.64E+02=	3.00E+03±3.16E+02
F16	1.16E+00±1.38E-01+	1.08E+00±2.16E-01=	1.10E+00±1.90E-01=	1.07E+00±1.64E-01
F17	3.05E+01±2.76E-02=	3.05E+01±5.13E-02=	3.04E+01±1.01E-02=	3.05E+01±5.54E-02
F18	6.74E+01±6.65E+00=	7.53E+01±7.14E+00+	7.04E+01±6.74E+00=	6.92E+01±7.16E+00
F19	1.65E-01±1.08E-01=	1.21E-01±8.95E-02=	1.53E-01±1.05E-01=	1.59E-01±1.37E-01
F20	1.38E+01±2.32E+00+	1.31E+01±2.18E+00=	1.26E+01±2.28E+00=	1.22E+01±2.24E+00
F21	2.85E+02±8.44E+01=	2.71E+02±7.45E+01=	2.68E+02±5.86E+01=	2.76E+02±6.40E+01
F22	1.69E+02±4.13E+01+	1.76E+02±4.45E+01+	1.66E+02±4.54E+01=	1.49E+02±4.27E+01
F23	3.69E+03±4.82E+02+	3.68E+03±3.35E+02+	3.49E+03±3.42E+02=	3.46E+03±3.70E+02
F24	2.28E+02±6.04E+00=	2.28E+02±5.83E+00=	2.26E+02±3.96E+00=	2.27E+02±7.01E+00
F25	2.83E+02±5.04E+00=	2.80E+02±7.49E+00=	2.82E+02±6.08E+00=	2.82E+02±6.55E+00
F26	2.00E+02±3.78E-02+	2.00E+02±5.44E-02+	2.00E+02±3.38E-02=	2.00E+02±0.00E+00
F27	4.28E+02±8.51E+01=	4.57E+02±1.07E+02=	4.67E+02±1.24E+02=	4.62E+02±1.09E+02
F28	3.00E+02±1.14E-13=	3.00E+02±1.46E-13=	3.00E+02±1.14E-13=	2.95E+02±2.73E+01
+ / = / -	11/17/0	8/19/1	3/25/0	--

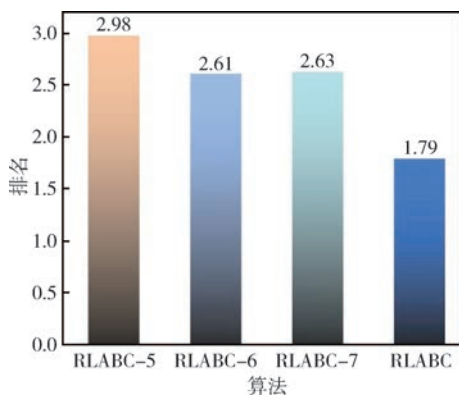


图 9 多邻域拓扑策略验证中算法的平均排名情况

4.3.2 双精英搜索策略验证

在 RLABC 中,引入了双精英搜索策略用于改进侦察蜂阶段.为验证该策略对 RLABC 性能的影响,我们设计了以下 2 种对比算法:

(1) RLABC-8:采用原始侦察蜂阶段的 RLABC

(2) RLABC-9:采用双精英搜索策略的 ABC

对比算法 RLABC-8 是将 RLABC 的双精英搜索策略替换为原始的侦察蜂阶段,用于验证双精英搜索策略对 RLABC 性能的影响. RLABC-9 是将经典 ABC 与双精英搜索策略进行结合,用于验证双精英搜索策略的有效性.同时,为方便对比,把经典 ABC 作为比较基准也加入算法对比.表 5 给出了所涉及的 4 种算法在 CEC2013 测试集上的实验结果,函数维度为 $D=30$.

从表 6 的实验结果可看出,与 RLABC-8 相比,RLABC 在 18 个函数上的结果要显著更优,仅在 4 个函数上略差,这说明双精英搜索策略比原始侦察蜂阶段的随机初始化方式更有利于 RLABC,能大幅提高其性能.同时,与 RLABC-9 的对比也能表明双精英搜索策略是有效的.事实上,与这两者的对比

还能得到一个重要的结论,即结合多邻域拓扑策略和双精英搜索策略能进一步提高 RLABC 性能,这两种策略的结合不会削弱对方,而是会产生积极的

影响,共同提高算法性能.图 11 给出实验中涉及的 4 种算法的排名情况,从中可进一步验证上述的实验分析.

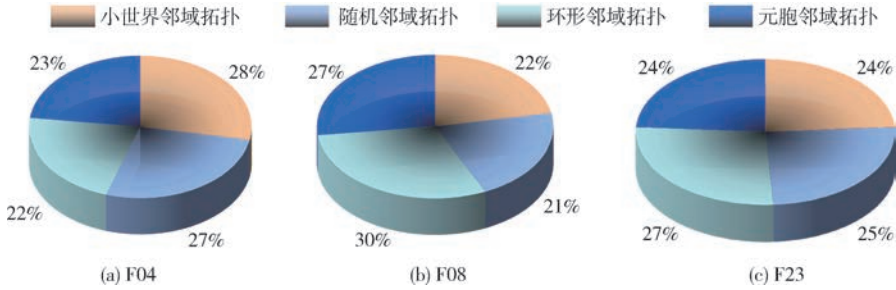


图 10 RLABC 在 3 个代表函数上的各邻域拓扑的使用频率

表 6 双精英搜索策略验证的实验结果

函数	ABC	RLABC-8	RLABC-9	RLABC
F01	6.37E-13±1.23E-13+	2.27E-13±0.00E+00+	0.00E+00±0.00E+00=	0.00E+00±0.00E+00
F02	1.42E+07±3.52E+06+	1.08E+07±2.45E+06+	6.87E+06±2.50E+06+	3.77E+06±1.27E+06
F03	8.64E+08±4.81E+08+	2.28E+08±1.22E+08+	6.30E+07±5.13E+07+	1.43E+07±9.61E+06
F04	7.08E+04±7.51E+03+	6.15E+04±7.31E+03+	2.01E+04±4.21E+03-	2.41E+04±3.24E+03
F05	1.04E-12±1.66E-13+	2.50E-13±4.55E-14+	1.14E-13±0.00E+00=	1.10E-13±2.04E-14
F06	1.50E+01±3.86E+00=	1.73E+01±4.45E+00=	1.49E+01±4.08E+00=	1.61E+01±3.45E+00
F07	1.17E+02±1.52E+01+	9.06E+01±9.56E+00+	7.81E+01±1.18E+01+	4.50E+01±7.70E+00
F08	2.10E+01±4.87E-02=	2.09E+01±5.03E-02=	2.09E+01±5.33E-02=	2.09E+01±5.54E-02
F09	2.96E+01±1.55E+00+	2.86E+01±1.28E+00+	2.86E+01±1.42E+00+	2.46E+01±2.95E+00
F10	2.75E+00±5.94E-01+	1.65E+00±2.24E-01+	1.83E+00±3.78E-01+	6.75E-01±2.85E-01
F11	1.88E-13±6.75E-14+	4.93E-14±1.93E-14+	0.00E+00±0.00E+00=	0.00E+00±0.00E+00
F12	2.55E+02±2.88E+01+	9.13E+01±1.74E+01+	9.88E+01±1.26E+01+	3.95E+01±8.18E+00
F13	3.04E+02±2.97E+01+	1.51E+02±1.98E+01+	1.49E+02±2.45E+01+	6.52E+01±1.88E+01
F14	5.11E+00±2.39E+00+	7.60E-02±4.04E-02-	2.93E+00±1.52E+00=	3.15E+00±2.95E+00
F15	4.00E+03±2.96E+02+	3.89E+03±2.96E+02+	3.65E+03±3.25E+02+	3.00E+03±3.16E+02
F16	1.62E+00±2.03E-01+	1.73E+00±1.83E-01+	1.14E+00±1.73E-01=	1.07E+00±1.64E-01
F17	3.06E+01±4.96E-02+	3.04E+01±8.56E-03-	3.06E+01±5.62E-02+	3.05E+01±5.54E-02
F18	3.38E+02±2.38E+01+	1.75E+02±1.56E+01+	1.23E+02±1.17E+01+	6.92E+01±7.16E+00
F19	9.28E-01±2.75E-01+	8.33E-02±9.05E-02-	2.58E-01±7.03E-02+	1.59E-01±1.37E-01
F20	1.45E+01±2.58E-01+	1.42E+01±3.87E-01=	1.29E+01±1.52E+00+	1.22E+01±2.24E+00
F21	2.03E+02±1.85E+01-	2.84E+02±6.94E+01=	1.92E+02±1.71E+01-	2.76E+02±6.40E+01
F22	1.31E+02±1.56E+01=	1.55E+02±4.60E+01=	1.24E+02±4.88E+00=	1.49E+02±4.27E+01
F23	4.78E+03±2.72E+02+	4.30E+03±4.19E+02+	4.29E+03±5.05E+02+	3.46E+03±3.70E+02
F24	2.89E+02±4.65E+00+	2.78E+02±3.98E+00+	2.54E+02±7.17E+00+	2.27E+02±7.01E+00
F25	3.06E+02±4.82E+00+	2.95E+02±3.77E+00+	2.94E+02±4.17E+00+	2.82E+02±6.55E+00
F26	2.01E+02±1.78E-01+	2.01E+02±1.68E-01+	2.00E+02±1.19E-01+	2.00E+02±0.00E+00
F27	4.00E+02±2.52E-01-	4.00E+02±4.83E-04-	4.54E+02±1.58E+02-	4.62E+02±1.09E+02
F28	2.30E+02±6.64E+01-	2.93E+02±3.59E+01=	2.72E+02±4.42E+01-	2.95E+02±2.73E+01
+/-/=	22/3/3	18/6/4	16/8/4	--

4.4 与 4 种邻域相关 ABC 对比

本小节中,我们将 RLABC 与 4 种邻域相关 ABC 进行对比,这 4 种对比算法如下:

(1) NSABC:基于新的邻域选择方法的 ABC^[13]

(2) ABCNG:基于自适应邻域搜索和高斯扰动的 ABC^[20]

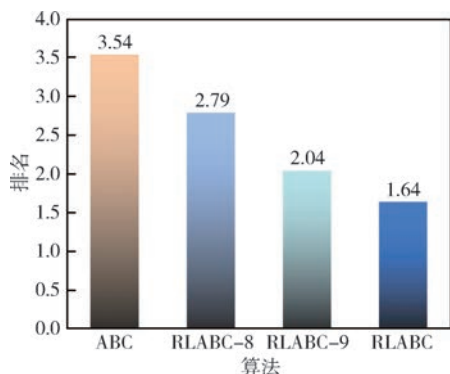


图 11 双精英搜索策略验证中算法的平均排名情况

(3) ABC-MNT: 多邻域拓扑的 ABC^[21]

(4) ABC-ANT: 自适应多邻域拓扑的 ABC^[14]

上述 4 种对比算法都引入了邻域拓扑来指导算法搜索。前 2 种采用了单邻域拓扑,而后 2 种则采用了多邻域拓扑。NSABC 的主要亮点是设计了一种基于环形邻域拓扑的新型搜索策略,以代替观察蜂阶段中的轮盘赌选择机制。类似于 NSABC,ABCNG 同样采用了环形邻域拓扑,但它的邻域半径是动态变化的。不同于单邻域拓扑的 ABC,ABC-MNT 首次在 ABC 中引入了多邻域拓扑,设计了一种多种群技术来同时使用 3 种邻域拓扑。不同于 ABC-MNT,ABC-ANT 采用了一种适应度地形分析在种群层次上应用多邻域拓扑。关于这 4 种对比算法的更多介绍,可参考本文第 2 节的“相关工作”部分。

为确保实验中对比的公平性,我们对算法所涉及的公共参数采用了相同设置,比如:种群大小 $SN=60$ 、算法终止条件 $MaxFEs=10000 \cdot D$,其它的特定参数则按算法原文献的推荐值进行设定。实验分别在 CEC2013 和 CEC2017 两套测试集上进行,测试函数维度为 $D=30$ 。下面按测试集来分别介绍算法的对比情况。

4.4.1 在 CEC2013 测试集上对比

表 7 给出了 5 种算法在 CEC2013 测试集上 $D=30$ 的实验结果,其中最好结果用粗体凸显。从表中最后一行的 Wilcoxon 秩和检验结果可看出,RLABC 在大部分测试函数上的结果要显著优于 4 种对比算法。具体分析如下:

(1) 在单峰问题上(F01-F05),RLABC 在所有问题上的结果均要显著优于 ABCNG。相比于 NSABC 和 ABC-MNT,RLABC 仅在 1 个问题上要差,但在其他的 4 个问题能表现出更好或相当的性能。对于 ABC-ANT,它在 F02、F03 和 F04 这 3 个测试函数上与 RLABC 性能相当,但在 F01 和

F05 上却要差于 RLABC。值得注意的是,F01 和 F05 属于可分离问题,这类问题可拆分为若干个仅涉及单个变量的问题,且不同问题之间无变量关联,这表明 RLABC 不仅能很好地处理不可分离问题,对可分离问题同样也有较好效果。

(2) 在多峰问题上(F06-F20),RLABC 能在 8 个函数上取得最好结果,整体性能远超过对比算法。具体来说,NSABC 和 ABCNG 在多峰问题上的表现显著差于 RLABC,它们仅在 F14 和 F17 这 2 个函数上稍占优势。ABC-MNT 和 ABC-ANT 这 2 种采用多邻域拓扑的算法也展现出了一定的竞争力,分别在 4 个和 3 个函数上取得了最好结果,但总体性能仍不及 RLABC。一般来说,多峰问题的复杂度要高于单峰问题,因为多峰问题存在大量的局部最优解。然而,RLABC 能在个体层次上自适应调整邻域拓扑,可有效缓解算法陷入局部最优解的问题,使其在多峰问题上表现更为出色。

(3) 在组合问题上(F21-F28),从 Wilcoxon 秩和检验的结果同样可看出,RLABC 的整体表现与在单峰问题上类似,其性能要显著地优于 NSABC、ABCNG、ABC-MNT。具体而言,RLABC 在 4 个函数上要优于 NSABC 和 ABC-MNT,在 6 个函数上要优于 ABCNG;这 3 个对比算法均只在 1 个函数上优于 RLABC。值得注意的是,ABC-ANT 在组合问题上取得了不错性能,与 RLABC 难分伯仲,能在 1 个函数上更优,也仅在 2 个函数上更差。不难看出,ABC-ANT 和 RLABC 都是自适应多邻域 ABC,因此在难度最大的组合问题上都表现不俗性能,但 RLABC 在个体层次上的自适应还是略优于种群层次上自适应的 ABC-ANT。考虑到前两类问题上的整体性能,RLABC 的优势更加明显。

通过表 7 的实验结果和上述结果分析,我们可注意到对比算法 ABC-ANT 的性能非常有竞争力。对 ABC-ANT 而言,其同样采用了自适应多邻域拓扑的思路,但与本文算法 RLABC 不同的是,它是在种群层次上实现多邻域拓扑,并且以 FDC 技术来实现自适应选择。除此之外,ABC-ANT 采用的是 3 种邻域拓扑,而 RLABC 则是 4 种。那么,非常自然地想到,若 RLABC 同样采用 3 种邻域拓扑,其性能是否依然要优于 ABC-ANT。为此,我们对第 4.3.1 节中设计的 RLABC-7 与 ABC-ANT 的结果进行了 Wilcoxon 秩和检验。需注意的是,RLABC-7 是在 RLABC 的基础上删除了小世界邻域拓扑,其他算法部分保持不变,使得 RLABC-7 只保留与 ABC-

ANT 相同的 3 种邻域拓扑, 这两者的 Wilcoxon 秩和检验结果是 RLABC-7 在 14 个函数上要显著优于 ABC-ANT, 而仅在 4 个函数上要更差. 这表明即

使 RLABC 只保留 3 种邻域拓扑, 其性能依然要好于 ABC-ANT, 也进一步说明了 RLABC 采用的强化学习对算法性能有重要贡献.

表 7 与邻域相关 ABC 在 CEC2013 测试集上的实验结果

函数	NSABC	ABCNG	ABC-MNT	ABC-ANT	RLABC
F01	4.55E-14±9.10E-14+	4.02E-13±9.64E-14+	2.27E-13±1.01E-28+	3.79E-14±8.47E-14+	0.00E+00±0.00E+00
F02	7.93E+06±2.80E+06+	1.61E+07±4.09E+06+	6.15E+06±2.46E+06+	3.81E+06±9.29E+05=	3.77E+06±1.27E+06
F03	9.84E+07±1.12E+08+	1.54E+09±7.81E+08+	3.66E+07±3.66E+07+	1.12E+07±1.21E+07=	1.43E+07±9.61E+06
F04	3.82E+04±9.42E+03+	8.15E+04±1.46E+04+	1.66E+04±3.67E+03-	2.19E+04±5.19E+03=	2.41E+04±3.24E+03
F05	1.06E-13±2.84E-14-	4.74E-13±7.22E-14+	1.74E-13±5.67E-14=	1.17E-13±2.04E-14+	1.10E-13±2.04E-14
F06	2.07E+01±6.37E+00=	2.28E+01±4.95E+00+	1.54E+01±4.03E+00-	1.64E+01±3.67E+00=	1.61E+01±3.45E+00
F07	6.35E+01±1.21E+01+	1.11E+02±1.50E+01+	4.64E+01±1.32E+01=	5.19E+01±7.79E+00+	4.50E+01±7.70E+00
F08	2.10E+01±4.59E-02=	2.10E+01±6.11E-02=	2.10E+01±4.60E-02=	2.09E+01±5.09E-02=	2.09E+01±5.54E-02
F09	2.87E+01±1.05E+00+	3.00E+01±1.45E+00+	2.21E+01±4.09E+00-	2.66E+01±2.52E+00+	2.46E+01±2.95E+00
F10	1.47E+00±2.61E-01+	4.67E+00±1.54E+00+	2.33E-01±9.31E-02-	2.87E-01±1.31E-01-	6.75E-01±2.85E-01
F11	3.32E-02±1.79E-01+	1.12E-13±1.03E-14+	5.87E-14±1.02E-14+	3.79E-15±1.42E-14=	0.00E+00±0.00E+00
F12	6.90E+01±1.41E+01+	1.99E+02±3.01E+01+	7.71E+01±1.44E+01+	4.13E+01±9.74E+00=	3.95E+01±8.18E+00
F13	1.04E+02±2.13E+01+	2.32E+02±2.71E+01+	1.33E+02±1.93E+01+	7.44E+01±1.51E+01+	6.52E+01±1.88E+01
F14	1.08E+00±7.41E-01-	3.02E-01±2.66E-01-	6.03E-01±1.05E+00-	3.18E+00±2.67E+00=	3.15E+00±2.95E+00
F15	4.13E+03±5.77E+02+	4.36E+03±2.22E+02+	4.05E+03±4.02E+02+	3.19E+03±3.72E+02+	3.00E+03±3.16E+02
F16	2.08E+00±2.99E-01+	1.46E+00±2.23E-01+	1.35E+00±2.06E-01+	1.03E+00±1.37E-01=	1.07E+00±1.64E-01
F17	3.05E+01±3.96E-02+	3.04E+01±2.49E-02-	3.04E+01±1.80E-03=	3.05E+01±4.86E-02+	3.05E+01±5.54E-02
F18	1.23E+02±1.12E+01+	2.40E+02±2.57E+01+	1.39E+02±1.08E+01+	7.61E+01±7.64E+00+	6.92E+01±7.16E+00
F19	2.89E-01±2.10E-01+	5.91E-01±1.88E-01+	9.41E-01±1.80E-01+	1.80E-01±1.57E-01=	1.59E-01±1.37E-01
F20	1.18E+01±2.63E-01=	1.19E+01±2.13E-01=	1.16E+01±3.71E-01=	9.36E+00±4.27E-01-	1.22E+01±2.24E+00
F21	3.47E+02±6.18E+01+	2.90E+02±3.30E+01+	2.97E+02±1.83E+01+	2.43E+02±6.38E+01=	2.76E+02±6.40E+01
F22	1.01E+02±4.51E+01=	7.38E+01±3.73E+01-	9.32E+01±3.75E+01-	9.86E+01±4.09E+01-	1.49E+02±4.27E+01
F23	4.65E+03±6.31E+02+	5.06E+03±5.00E+02+	4.83E+03±5.64E+02+	3.81E+03±4.66E+02+	3.46E+03±3.70E+02
F24	2.78E+02±4.92E+00+	2.81E+02±6.46E+00+	2.40E+02±1.19E+01+	2.29E+02±5.87E+00=	2.27E+02±7.01E+00
F25	2.78E+02±3.25E+00-	3.02E+02±7.01E+00+	2.81E+02±9.75E+00=	2.80E+02±9.17E+00=	2.82E+02±6.55E+00
F26	2.00E+02±2.09E-01+	2.01E+02±3.00E-01+	2.00E+02±6.40E-02+	2.00E+02±2.49E-02+	2.00E+02±0.00E+00
F27	6.34E+02±3.10E+02=	4.63E+02±1.85E+02=	6.64E+02±2.49E+02=	4.48E+02±9.70E+01=	4.62E+02±1.09E+02
F28	3.00E+02±0.00E+00=	3.03E+02±4.68E+00+	3.00E+02±0.00E+00=	2.90E+02±3.92E+01=	2.95E+02±2.73E+01
+/-/=	19/6/3	22/3/3	14/8/6	10/15/3	--

为方便查看上述 5 种算法的整体对比情况, 图 12 给出了算法的平均排名结果, 可看出 RLABC 获得了最好的排名, 其次是 ABC-ANT. 综上所述, RLABC 在个体层次上实现的自适应多邻域拓扑, 比单一邻域拓扑或其他形式的多邻域拓扑更好.

4.4.2 在 CEC2017 测试集上对比

为进一步评估 RLABC 性能, 在本小节中, 我们将其与邻域相关的 ABC 在 CEC2017 测试集上进行对比. 相比于 CEC2013 测试集, CEC2017 测试集难度更大, 它涵盖了 4 种不同类型的测试问题. 本次实验中的参数设置与 CEC2013 的实验相同, 测试函数维度同样保持为 $D=30$.

表 8 给出了 5 种算法的实验结果, 其中最好结果用粗体凸显. 不难看出, RLABC 依旧是 5 种算法

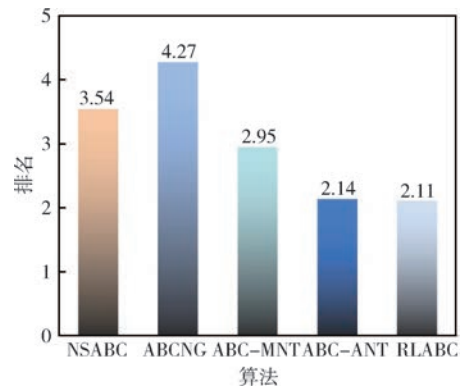


图 12 与邻域相关 ABC 在 CEC2013 测试集上的平均排名

中整体性能最好的. 具体而言, 在单峰问题和多峰问题上, RLABC 在至少 6 个函数上取得了最好结果, 超过 NSABC、ABCNG 和 ABC-MNT 这 3 种算法.

而对于 ABC-ANT,它只在 F01 和 F05 上略优于 RLABC;但在混合问题和组合问题上,RLABC 表现出了更加明显的优异性能.从表中最后一行的 Wilcoxon 秩和检验结果来看,RLABC 分别在 10 个、14 个和 10 个函数上好于前 3 种算法,而前 3 种算法仅在 3 个函数上超过了 RLABC;另外,RLABC

在大多数函数上要优于或相当于对比算法中表现最好的 ABC-ANT,仅在 4 个函数上略差于 ABC-ANT.由此可见,RLABC 在 CEC2017 测试集上同样有非常明显的性能优势.图 13 给出了通过 Friedman 检验得到的算法平均排名情况,可看出 RLABC 和 ABC-ANT 的排名相当.

表 8 与邻域相关 ABC 在 CEC2017 测试集上的实验结果

函数	NSABC	ABCNG	ABC-MNT	ABC-ANT	RLABC
F01	3.06E+02±5.06E+02-	3.95E+02±2.69E+02-	3.68E+02±3.72E+02-	1.54E+03±3.27E+03-	3.00E+03±4.33E+03
F02	9.60E+04±1.96E+04+	1.06E+05±1.94E+04+	2.63E+03±1.20E+03+	2.21E+04±5.90E+03+	1.60E+04±3.58E+03
F03	1.14E+01±1.85E+01=	5.17E+01±2.89E+01+	4.59E+01±3.03E+01+	2.56E+01±3.29E+01=	1.87E+01±3.06E+01
F04	2.54E+01±3.62E+00+	4.79E+01±8.44E+00+	5.14E+01±7.83E+00+	1.59E+01±2.67E+00+	1.47E+01±4.56E+00
F05	7.20E-14±9.51E-14-	3.60E-13±1.58E-13-	1.67E-13±5.67E-14-	4.17E-14±5.48E-14-	9.20E-01±5.21E-01
F06	5.45E+01±2.25E+00+	7.94E+01±7.44E+00+	8.84E+01±6.07E+00+	4.63E+01±2.74E+00=	4.57E+01±3.18E+00
F07	2.62E+01±4.52E+00+	5.37E+01±6.99E+00+	5.87E+01±7.75E+00+	1.65E+01±2.68E+00=	1.60E+01±2.57E+00
F08	1.10E+00±1.28E+00+	2.72E+02±1.20E+02+	6.63E-01±6.57E-01+	3.06E-02±8.89E-02=	3.50E-02±6.38E-02
F09	1.95E+03±1.93E+02+	2.24E+03±2.20E+02+	2.24E+03±2.81E+02+	1.61E+03±2.48E+02+	1.22E+03±2.43E+02
F10	9.27E+01±3.48E+01+	8.42E+02±5.31E+02+	4.57E+01±2.80E+01+	3.32E+01±1.34E+01=	3.16E+01±1.32E+01
F11	3.82E+05±1.51E+05+	6.54E+05±3.63E+05+	2.68E+05±2.37E+05=	2.28E+05±1.66E+05=	2.26E+05±1.41E+05
F12	1.44E+04±1.29E+04=	7.46E+02±5.75E+02-	1.06E+04±1.04E+04=	9.65E+03±9.46E+03=	1.05E+04±1.16E+04
F13	4.79E+04±3.07E+04+	1.19E+05±8.28E+04+	1.29E+04±1.51E+04-	3.28E+04±3.49E+04+	2.22E+04±2.99E+04
F14	5.78E+03±6.37E+03=	1.15E+02±6.37E+01+	5.02E+03±5.49E+03=	4.29E+03±5.13E+03=	8.79E+03±1.04E+04
F15	4.57E+02±1.17E+02+	6.65E+02±1.59E+02+	5.43E+02±1.35E+02+	4.00E+02±1.26E+02+	3.27E+02±1.18E+02
F16	8.05E+01±3.71E+01+	1.53E+02±7.16E+01+	1.06E+02±5.23E+01+	3.17E+01±1.59E+01+	2.42E+01±1.52E+01
F17	2.07E+05±9.12E+04+	2.55E+05±1.27E+05+	9.06E+04±5.37E+04=	6.74E+04±2.67E+04=	8.20E+04±4.59E+04
F18	6.80E+03±6.45E+03=	7.10E+01±6.14E+01-	9.41E+03±8.30E+03=	6.45E+03±6.47E+03=	7.07E+03±7.58E+03
F19	1.03E+02±5.68E+01+	2.09E+02±7.58E+01+	1.51E+02±7.20E+01+	9.81E+01±5.55E+01+	5.70E+01±4.99E+01
F20	2.27E+02±1.69E+01+	2.41E+02±3.62E+01+	2.45E+02±3.88E+01+	2.14E+02±2.21E+01=	2.13E+02±2.34E+01
F21	1.00E+02±0.00E+00=	1.14E+02±7.03E+00+	1.00E+02±0.00E+00=	1.00E+02±0.00E+00=	1.00E+02±0.00E+00
F22	3.77E+02±6.53E+00+	3.99E+02±1.12E+01+	4.05E+02±8.00E+00+	3.64E+02±3.06E+01=	3.63E+02±4.67E+00
F23	4.79E+02±9.32E+01+	4.26E+02±1.38E+02-	4.93E+02±1.17E+01+	4.54E+02±7.16E+00=	4.53E+02±7.86E+00
F24	3.83E+02±3.02E+00-	3.86E+02±1.55E+00=	3.87E+02±1.14E+00=	3.86E+02±1.57E+00+	3.85E+02±1.53E+00
F25	8.01E+02±5.20E+02=	2.86E+02±4.21E+01-	1.31E+03±6.53E+02+	6.97E+02±4.59E+02=	9.79E+02±3.68E+02
F26	4.69E+02±1.11E+01-	5.15E+02±4.82E+00+	5.09E+02±6.46E+00+	5.03E+02±3.97E+00=	5.02E+02±4.81E+00
F27	3.15E+02±2.85E+01-	4.08E+02±3.31E+00+	3.45E+02±4.56E+01-	3.61E+02±4.26E+01-	3.93E+02±1.85E+01
F28	4.24E+02±5.71E+01-	5.49E+02±5.80E+01+	5.47E+02±5.23E+01+	4.59E+02±4.84E+01=	4.55E+02±3.23E+01
F29	3.55E+03±2.05E+03-	7.78E+03±2.90E+03-	6.45E+03±2.90E+03-	5.55E+03±2.34E+03-	1.08E+04±4.18E+03
+/-/=	16/6/7	21/1/7	17/7/5	8/17/4	--

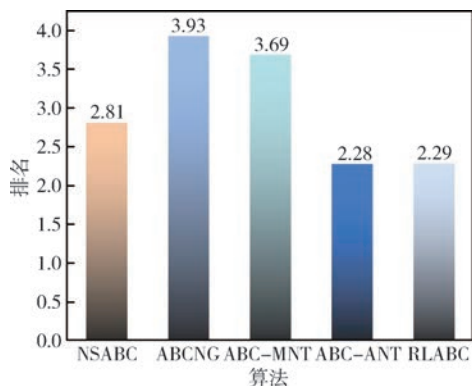


图 13 与邻域相关 ABC 在 CEC2017 测试集上的平均排名

4.5 与 4 种其他知名改进 ABC 对比

4.5.1 在两套测试集上对比

为更好地评估 RLABC 性能,我们将其与 4 种其他知名改进 ABC 进行了对比,具体如下:

- (1) iqABC:改进的 qABC^[17]
- (2) CGABC:基于高斯分布的元胞 ABC^[19]
- (3) MGABC:基于多精英引导的 ABC^[9]
- (4) ILABC:基于信息学习策略的 ABC^[24]

前 3 种对比算法的共性之处是均采用了种群中的最优个体或精英个体来改进搜索方程,以期提

高 ABC 的开采能力. iqABC 改进了经典的 qABC, 引入了种群的最优个体, 设计了专用控制参数来调整最优个体的使用频率. CGABC 采用了一种基于高斯分布的解搜索方程, 并引入局部吸引子来指导搜索. MGABC 在解搜索方程中引入了由多个精英个体组成的精英组. 第 4 个对比算法 ILABC 则是如何与其他搜索技术相结合的代表性工作, 采用了多种群技术, 按聚类划分的方式将种群划分为若干个子种群.

为确保实验对比的公平性, 与本文的前面实验类似, 本次实验中所有算法的公共参数均采用相同值. 但需专门指出的是, 因 ILABC 采用了多种群机制, 因此 ILABC 的种群大小采用原文献的方式进行设置, 而其余 4 种算法的种群大小均为 $SN=60$. 对其他特定参数, 实验中采用了算法原文献中的推荐值. 本次实验同样在 CEC2013 和 CEC2017 两套测试集上进行, 测试的问题维度为 $D=30$. 下面简介实验对比结果.

针对 CEC2013 测试集, 表 9 给出了 5 种算法的

实验结果. 从表中最后一行的 Wilcoxon 秩和检验结果可看出, RLABC 在大部分函数上的性能依然显著优于 4 种对比算法. 具体来说, 在单峰问题上 (F01-F05), RLABC 要好于前 3 种算法 ILABC、CGABC 和 iqABC, 它能在 4 个函数上取得最优结果, 而 MGABC 与 RLABC 在这些函数上表现相当. 在多峰问题上 (F06-F20), RLABC 有更加明显的优势, 对比前 3 种算法, 它在至少 8 个函数上取得了最好结果, 而且在 11 个函数上的性能要显著优于 MGABC, 这表明 RLABC 能更有效地解决多峰问题. 在组合问题上 (F21-F28), RLABC 在 5 个函数上取得了最好结果, 仅在 1 个函数上略差于 4 种对比算法. 综上, 对于这三类问题, RLABC 都表现出了优异性能.

针对 CEC2017 测试集, 表 10 给出了实验结果. 从该表中可看出, 在单峰问题和多峰问题上 (F01-F09), RLABC 在 5 个函数上取得了最好结果, 而 4 种对比算法都仅在 2 个函数上的性能优于 RLABC. 对于混合问题和组合问题 (F10-F29), RLABC 能在

表 9 与其他知名改进 ABC 在 CEC2013 测试集上的实验结果

函数	iqABC	CGABC	MGABC	ILABC	RLABC
F01	4.32E-13±6.84E-14+	0.00E+00±0.00E+00=	7.57E-14±1.07E-13+	2.19E-13±4.07E-14+	0.00E+00±0.00E+00
F02	8.19E+06±2.61E+06+	1.58E+07±5.21E+06+	1.01E+06±4.60E+05-	1.14E+07±3.79E+06+	3.77E+06±1.27E+06
F03	3.09E+08±3.27E+08+	4.05E+08±3.22E+08+	1.41E+08±1.51E+08+	2.01E+08±1.69E+08+	1.43E+07±9.61E+06
F04	8.36E+04±1.23E+04+	8.84E+04±1.64E+04+	1.96E+04±2.32E+03-	7.31E+04±8.58E+03+	2.41E+04±3.24E+03
F05	6.33E-13±8.15E-14+	1.14E-13±5.05E-29+	1.10E-13±2.05E-14=	2.65E-13±6.12E-14+	1.10E-13±2.04E-14
F06	1.27E+01±4.35E+00-	2.38E+01±1.21E+00+	2.03E+01±2.15E+01+	1.84E+01±3.61E+00+	1.61E+01±3.45E+00
F07	1.07E+02±1.51E+01+	1.17E+02±1.49E+01+	4.81E+01±1.89E+01=	9.55E+01±1.18E+01+	4.50E+01±7.70E+00
F08	2.09E+01±6.18E-02=	2.11E+01±5.99E-02+	2.09E+01±5.63E-02=	2.09E+01±5.47E-02=	2.09E+01±5.54E-02
F09	3.07E+01±1.57E+00+	3.22E+01±2.50E+00+	1.96E+01±3.86E+00-	2.89E+01±1.62E+00+	2.46E+01±2.95E+00
F10	1.57E+00±3.10E-01+	1.35E+00±2.29E-01+	2.03E-01±9.22E-02-	2.16E+00±5.59E-01+	6.75E-01±2.85E-01
F11	6.25E-14±1.72E-14+	0.00E+00±0.00E+00=	1.14E-14±2.27E-14+	4.17E-14±2.51E-14+	0.00E+00±0.00E+00
F12	2.71E+02±3.54E+01+	8.69E+01±1.99E+01+	1.22E+02±2.62E+01+	1.16E+02±1.46E+01+	3.95E+01±8.18E+00
F13	3.21E+02±3.61E+01+	1.50E+02±2.64E+01+	1.56E+02±2.33E+01+	1.68E+02±3.17E+01+	6.52E+01±1.88E+01
F14	2.36E-01±3.04E-01-	4.17E-01±6.23E-01-	2.26E-01±2.72E-01=	2.13E-01±1.94E-01-	3.15E+00±2.95E+00
F15	3.56E+03±4.18E+02+	5.22E+03±8.63E+02+	3.74E+03±6.44E+02+	4.37E+03±4.08E+02+	3.00E+03±3.16E+02
F16	9.69E-01±1.40E-01-	2.90E+00±6.58E-01+	2.26E+00±2.63E-01+	1.85E+00±2.42E-01+	1.07E+00±1.64E-01
F17	3.06E+01±6.70E-02+	3.04E+01±6.63E-03=	3.04E+01±1.80E-02-	3.04E+01±7.11E-15-	3.05E+01±5.54E-02
F18	2.99E+02±3.70E+01+	1.21E+02±1.63E+01+	1.21E+02±3.17E+01+	1.94E+02±1.88E+01+	6.92E+01±7.16E+00
F19	4.12E-01±1.06E-01+	1.11E-01±7.87E-02=	1.41E+00±5.45E-01+	2.75E-01±1.06E-01+	1.59E-01±1.37E-01
F20	1.19E+01±4.00E-01=	1.25E+01±4.33E-01=	9.95E+00±6.84E-01-	1.19E+01±3.39E-01=	1.22E+01±2.24E+00
F21	1.65E+02±3.62E+01-	3.07E+02±2.49E+01+	3.20E+02±4.00E+01+	2.14E+02±3.37E+01-	2.76E+02±6.40E+01
F22	3.01E+01±1.15E+01-	1.03E+02±2.98E+01-	1.15E+02±3.39E+01-	1.80E+01±1.80E+01-	1.49E+02±4.27E+01
F23	4.81E+03±5.19E+02+	5.85E+03±8.83E+02+	3.96E+03±5.91E+02+	5.13E+03±4.05E+02+	3.46E+03±3.70E+02
F24	2.89E+02±6.90E+00+	2.85E+02±5.54E+00+	2.29E+02±1.18E+01=	2.73E+02±7.51E+00+	2.27E+02±7.01E+00
F25	3.14E+02±6.55E+00+	2.83E+02±5.75E+00=	2.73E+02±9.51E+00=	3.01E+02±5.56E+00+	2.82E+02±6.55E+00
F26	2.01E+02±4.00E-01+	2.01E+02±3.34E-01+	2.00E+02±0.00E+00=	2.01E+02±1.80E-01+	2.00E+02±0.00E+00
F27	5.80E+02±3.14E+02+	4.13E+02±5.05E+01-	6.92E+02±1.03E+02+	4.28E+02±1.30E+02-	4.62E+02±1.09E+02
F28	2.37E+02±9.26E+01=	3.02E+02±8.39E+00+	3.58E+02±2.76E+02=	2.99E+02±5.21E+00+	2.95E+02±2.73E+01
+/-/-	19/3/6	18/6/4	13/8/7	21/2/5	--

表 10 与其他知名改进 ABC 在 CEC2017 测试集上的实验结果

函数	iqABC	CGABC	MGABC	ILABC	RLABC	
F01	9.25E+01±1.02E+02-	7.54E+02±8.56E+02-	1.76E+03±1.49E+03=	1.03E+02±1.70E+02-	3.00E+03±4.33E+03	
F02	1.07E+05±1.54E+04+	1.06E+05±2.26E+04+	5.79E+02±5.56E+02-	1.05E+05±1.38E+04+	1.60E+04±3.58E+03	
F03	2.45E+01±2.65E+01+	2.26E+01±9.52E+00+	1.53E+01±2.54E+01=	3.79E+01±2.37E+01+	1.87E+01±3.06E+01	
F04	7.49E+01±9.48E+00+	3.37E+01±8.53E+00+	4.62E+01±1.19E+01+	3.96E+01±6.72E+00+	1.47E+01±4.56E+00	
F05	6.45E-08±3.92E-08-	0.00E+00±0.00E+00-	1.00E-12±3.37E-12-	0.00E+00±0.00E+00-	9.20E-01±5.21E-01	
F06	9.90E+01±9.92E+00+	5.04E+01±5.45E+00+	7.10E+01±1.32E+01+	6.56E+01±5.61E+00+	4.57E+01±3.18E+00	
F07	8.29E+01±1.02E+01+	3.34E+01±8.96E+00+	4.56E+01±8.46E+00+	4.16E+01±6.03E+00+	1.60E+01±2.57E+00	
F08	5.91E+02±3.16E+02+	2.16E+00±1.99E+00+	2.44E-01±3.00E-01+	2.19E+01±9.31E+00+	3.50E-02±6.38E-02	
F09	2.18E+03±1.87E+02+	2.55E+03±6.62E+02+	2.17E+03±4.18E+02+	2.05E+03±3.03E+02+	1.22E+03±2.43E+02	
F10	3.35E+02±2.79E+02+	2.04E+02±1.58E+02+	4.09E+01±3.43E+01=	4.28E+02±3.18E+02+	3.16E+01±1.32E+01	
F11	3.28E+05±1.99E+05+	7.20E+05±4.44E+05+	6.07E+04±4.07E+04-	4.26E+05±2.92E+05+	2.26E+05±1.41E+05	
F12	4.08E+03±4.16E+03-	1.14E+04±1.12E+04=	1.34E+04±1.11E+04=	9.19E+03±7.56E+03=	1.05E+04±1.16E+04	
F13	5.18E+04±3.87E+04+	2.51E+05±2.34E+05+	9.74E+03±1.17E+04-	1.12E+05±6.08E+04+	2.22E+04±2.99E+04	
F14	6.85E+02±7.73E+02-	3.60E+03±4.26E+03-	3.72E+03±4.37E+03-	4.10E+03±3.62E+03=	8.79E+03±1.04E+04	
F15	6.00E+02±1.24E+02+	6.26E+02±1.83E+02+	4.80E+02±2.09E+02+	5.64E+02±1.18E+02+	3.27E+02±1.18E+02	
F16	1.61E+02±6.74E+01+	1.94E+02±1.36E+02+	1.11E+02±9.38E+01+	1.06E+02±4.38E+01+	2.42E+01±1.52E+01	
F17	1.21E+05±4.54E+04+	5.88E+05±6.74E+05+	1.21E+05±6.56E+04+	2.05E+05±9.33E+04+	8.20E+04±4.59E+04	
F18	7.72E+02±8.12E+02-	4.79E+03±5.81E+03=	4.87E+03±4.98E+03=	8.40E+03±5.93E+03=	7.07E+03±7.58E+03	
F19	2.26E+02±8.63E+01+	2.42E+02±9.94E+01+	1.19E+02±5.31E+01+	1.77E+02±5.96E+01+	5.70E+01±4.99E+01	
F20	2.47E+02±7.04E+01+	2.39E+02±1.04E+01+	2.46E+02±9.66E+00+	2.20E+02±4.67E+01+	2.13E+02±2.34E+01	
F21	2.72E+02±6.37E+02+	1.00E+02±3.50E-01+	1.00E+02±0.00E+00=	1.00E+02±5.39E-01=	1.00E+02±0.00E+00	
F22	3.99E+02±4.37E+01+	3.81E+02±1.05E+01+	3.99E+02±1.36E+01+	3.77E+02±4.52E+01+	3.63E+02±4.67E+00	
F23	4.61E+02±1.93E+02+	5.38E+02±3.15E+01+	4.85E+02±1.94E+01+	4.48E+02±1.11E+02-	4.53E+02±7.86E+00	
F24	3.84E+02±8.84E-01+	3.78E+02±1.05E+00+	3.91E+02±1.09E+01+	3.86E+02±1.18E+00=	3.85E+02±1.53E+00	
F25	2.42E+02±8.16E+01-	8.58E+02±3.42E+02-	1.40E+03±5.65E+02+	3.41E+02±3.10E+02+	9.79E+02±3.68E+02	
F26	5.21E+02±7.10E+00+	5.00E+02±0.00E+00-	5.18E+02±6.62E+00+	5.13E+02±3.97E+00+	5.02E+02±4.81E+00	
F27	4.01E+02±2.56E+00+	4.93E+02±5.52E+00+	3.11E+02±3.16E+01-	4.07E+02±3.72E+00+	3.93E+02±1.85E+01	
F28	6.24E+02±6.46E+01+	3.97E+02±7.40E+01-	4.98E+02±8.63E+01+	5.24E+02±4.01E+01+	4.55E+02±3.23E+01	
F29	5.57E+03±1.60E+03-	1.05E+03±6.88E+02-	4.67E+03±1.99E+03-	5.66E+03±1.29E+03-	1.08E+04±4.18E+03	
	+/-/-	22/0/7	20/2/7	16/6/7	20/5/4	--

13 个函数上优于 ILABC 和 CGABC, 在 15 个函数上优于 iqABC, 在 11 个函数上优于 MGABC; 而对比算法中表现最好的 MGABC 仅在 5 个函数上优于 RLABC.

为更直观地对比上述 5 种算法在两套测试集上的整体性能, 图 14 给出了算法的平均排名情况. 从该图中可看出, RLABC 在两套测试集上均能取得排名第一的好成绩, MGABC 紧随其后, ILABC 和 CGABC 难分伯仲, 而 iqABC 则在最后.

4.5.2 多维度测试

从上一小节的实验结果可看出, RLABC 在测试函数维度为 $D=30$ 时能取得非常有竞争力的性能. 为继续评估 RLABC 性能, 在本小节中将检验算法在两种不同维度下的性能情况, 分别是 $D=50$ 和 100, 实验在 CEC2017 测试集上进行, 算法的停机条件与 4.1 节中的设置保持相同.

对于 $D=50$, 表 11 给出了这 5 种算法的实验结

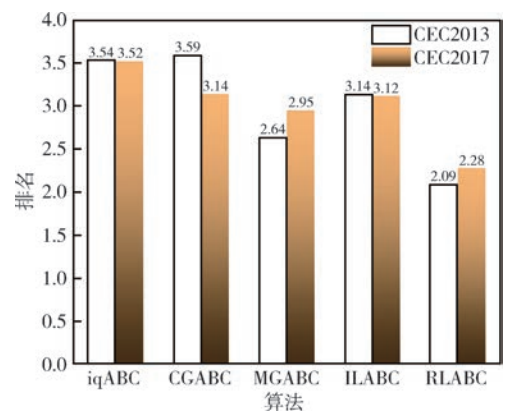


图 14 与其他知名改进 ABC 在两套测试集上的平均排名

果情况. 与 $D=30$ 相比, 此维度下的问题更加复杂, 但 RLABC 在大多数函数上仍然取得了最好结果, 且性能并没有出现下降. 具体来说, 在与 ILABC 的对比中, RLABC 在 23 个函数上优于 ILABC, 比 $D=30$ 时多了 3 个函数. 在与 CGABC 和 iqABC 的

对比中,RLABC 分别在 17 个和 21 个函数上优于它们,虽然比 $D=30$ 时少了 3 个和 1 个函数,但它们在好于 RLABC 的函数个数上分别少了 2 个和 3 个,这表明 RLABC 并没有因为维度增加而失去对这两个算法的优势. 在与 MGABC 的对比中,RLABC 在 17 个函数上优于 MGABC,比 $D=30$ 多了 1 个函数. 综合来看,RLABC 在 $D=50$ 时依然有非常好的性能.

然而,值得注意的是,在 CEC2017 测试集的混合问题(F10-F19)上,按 Wilcoxon 秩和检验结果来看,MGABC 在 4 个函数优于 RLABC,而在 3 个函

数上弱于 RLABC,因此 MGABC 在混合问题上的性能要略好于 RLABC. 主要原因是这些混合问题所混合的函数类型有所区别,例如:F10 混合了 3 种函数(Zakharov 函数、Rosenbrock 函数、Rastrigin 函数),其中前 2 种是单峰类型函数,混合比例占到 60%,对于 MGABC 这样开采能力特别强的算法更有利. 类似地,对于 F11 也是如此,因为 F11 同样包含了 60%的单峰函数. 然而,RLABC 搜索能力更加均衡,对勘探和开采平衡得更好,因此算法的整体性能也更优,特别是在求解难度更大的组合问题(F20-F29)上更是如此.

表 11 多维度测试中 $D=50$ 的实验结果

函数	iqABC	CGABC	MGABC	ILABC	RLABC
F01	1.53E+03±1.78E+03=	1.97E+03±2.75E+03=	2.22E+03±3.51E+03-	5.67E+02±6.55E+02-	4.39E+03±5.52E+03
F02	2.32E+05±2.55E+04+	2.52E+05±3.50E+04+	1.05E+05±7.07E+03+	2.04E+05±2.22E+04+	5.09E+04±8.03E+03
F03	3.70E+01±1.32E+01=	3.21E+01±6.91E+00=	5.90E+01±4.41E+01=	5.01E+01±1.74E+01+	3.59E+01±2.28E+01
F04	1.80E+02±1.54E+01+	8.50E+01±1.80E+01+	1.36E+02±2.37E+01+	9.31E+01±1.11E+01+	3.32E+01±4.31E+00
F05	8.96E-08±4.82E-08-	0.00E+00±0.00E+00-	3.69E-07±6.26E-07-	0.00E+00±0.00E+00-	1.37E+00±6.45E-01
F06	2.14E+02±1.67E+01+	1.03E+02±1.04E+01+	1.74E+02±2.66E+01+	1.35E+02±9.70E+00+	8.20E+01±4.60E+00
F07	1.78E+02±1.85E+01+	8.58E+01±1.64E+01+	1.35E+02±2.66E+01+	9.69E+01±1.09E+01+	3.57E+01±5.70E+00
F08	3.03E+03±9.02E+02+	2.25E+01±1.36E+01+	3.85E+00±4.83E+00+	2.05E+02±9.24E+01+	5.07E-01±6.04E-01
F09	3.88E+03±3.74E+02+	5.28E+03±8.29E+02+	4.10E+03±5.28E+02+	4.00E+03±4.42E+02+	2.81E+03±2.79E+02
F10	1.03E+03±6.23E+02+	2.01E+03±1.04E+03+	8.80E+01±2.06E+01-	1.98E+03±1.43E+03+	2.16E+02±1.43E+02
F11	1.90E+06±9.10E+05+	4.54E+06±2.32E+06+	6.95E+05±2.82E+05-	2.48E+06±9.81E+05+	1.36E+06±5.18E+05
F12	2.62E+03±2.83E+03=	3.14E+03±3.48E+03=	2.59E+03±2.99E+03=	3.45E+03±3.25E+03=	5.07E+03±6.52E+03
F13	6.30E+05±4.02E+05+	2.63E+06±1.97E+06+	4.21E+04±2.44E+04-	5.96E+05±3.27E+05+	1.38E+05±1.28E+05
F14	7.00E+03±4.37E+03=	1.02E+04±1.10E+04=	8.60E+03±5.34E+03=	4.99E+03±2.95E+03=	6.78E+03±6.48E+03
F15	1.20E+03±1.69E+02+	1.07E+03±2.61E+02+	1.03E+03±3.57E+02+	1.14E+03±2.11E+02+	7.73E+02±2.16E+02
F16	8.60E+02±1.55E+02+	1.08E+03±2.78E+02+	8.45E+02±2.19E+02+	8.93E+02±1.44E+02+	4.62E+02±1.26E+02
F17	6.32E+05±2.97E+05+	3.73E+06±2.50E+06+	2.01E+05±8.37E+04-	1.03E+06±5.41E+05+	3.69E+05±1.53E+05
F18	5.96E+03±3.91E+03-	1.46E+04±5.19E+03=	1.47E+04±8.04E+03=	1.48E+04±4.89E+03=	1.50E+04±1.12E+04
F19	6.71E+02±1.42E+02+	9.38E+02±3.37E+02+	5.08E+02±1.87E+02+	6.20E+02±1.51E+02+	2.38E+02±1.36E+02
F20	3.84E+02±4.59E+01+	2.93E+02±1.63E+01+	3.15E+02±2.11E+01+	3.03E+02±9.08E+00+	2.36E+02±8.05E+00
F21	4.41E+03±1.22E+03+	3.62E+03±2.98E+03=	2.73E+03±2.52E+03=	3.67E+03±1.96E+03+	3.21E+03±1.07E+03
F22	6.30E+02±6.20E+01+	4.69E+02±1.59E+01=	5.52E+02±2.79E+01+	5.45E+02±1.44E+01+	4.68E+02±1.20E+01
F23	9.57E+02±5.73E+01+	8.13E+02±5.97E+01+	6.58E+02±3.06E+01+	7.52E+02±3.00E+01+	5.98E+02±1.31E+01
F24	5.13E+02±1.99E+01+	4.31E+02±3.40E-02-	5.69E+02±3.03E+01+	5.26E+02±1.39E+01+	4.88E+02±2.18E+01
F25	1.30E+03±1.49E+03-	1.23E+03±1.67E+02-	2.56E+03±1.01E+03+	2.15E+03±4.68E+02+	1.50E+03±2.26E+02
F26	6.91E+02±4.07E+01+	5.00E+02±0.00E+00-	6.59E+02±5.03E+01+	6.51E+02±2.07E+01+	5.67E+02±1.31E+01
F27	4.88E+02±1.28E+01+	4.98E+02±6.39E+00+	4.98E+02±2.86E+01+	5.04E+02±1.34E+01+	4.62E+02±4.95E+00
F28	1.08E+03±1.30E+02+	6.90E+02±1.55E+02+	7.84E+02±2.33E+02+	8.22E+02±1.07E+02+	4.87E+02±1.04E+02
F29	7.15E+05±4.61E+04-	2.60E+03±1.66E+03-	9.15E+05±1.82E+05=	8.08E+05±6.83E+04-	8.94E+05±1.03E+05
+/=-/	21/4/4	17/7/5	17/6/6	23/3/3	--

对于 $D=100$,表 12 给出了实验结果. 从中可看出,5 种算法的总体对比情况要类似于 $D=50$,但与 CGABC 的对比情况要更好. 在 $D=50$ 时,RLABC 分别在 17 个和 5 个函数上要优于和差于 CGABC; 但 $D=100$ 时,RLABC 能取得更好结果的函数个数

增长至了 22 个,而更差结果的函数个数减少到了 3 个. 因此,从这两种维度上的对比情况来看,RLABC 的性能有较好的鲁棒性. 在图 15 中给出这两种维度上的算法平均排名情况,可看出 RLABC 均排名第一,且 $D=100$ 时的情况要略好于 $D=50$.

表 12 多维度测试中 $D=100$ 的实验结果

函数	iqABC	CGABC	MGABC	ILABC	RLABC
F01	2.81E+03±2.54E+03-	4.20E+03±4.24E+03-	3.94E+03±4.16E+03-	1.25E+03±1.67E+03-	1.07E+04±1.05E+04
F02	5.48E+05±4.01E+04+	6.50E+05±5.01E+04+	2.74E+05±7.80E+03+	5.51E+05±2.79E+04+	1.67E+05±2.12E+04
F03	2.02E+02±1.94E+01=	1.98E+02±2.30E+01=	1.69E+02±4.71E+01-	2.14E+02±2.25E+01+	1.95E+02±1.54E+01
F04	5.76E+02±2.83E+01+	3.30E+02±5.33E+01+	4.74E+02±6.45E+01+	3.41E+02±2.58E+01+	1.02E+02±1.02E+01
F05	1.21E-07±5.83E-08-	6.95E-10±3.20E-09-	1.22E-02±3.23E-02-	0.00E+00±0.00E+00-	5.89E+00±1.05E+00
F06	6.84E+02±3.66E+01+	2.94E+02±3.13E+01+	6.49E+02±8.01E+01+	3.81E+02±3.50E+01+	1.89E+02±1.17E+01
F07	6.03E+02±4.40E+01+	3.26E+02±6.24E+01+	4.48E+02±6.59E+01+	3.44E+02±2.60E+01+	1.01E+02±1.06E+01
F08	2.30E+04±3.00E+03+	1.09E+04±4.36E+03+	1.65E+03±1.33E+03+	7.35E+03±1.75E+03+	4.50E+01±3.55E+01
F09	1.07E+04±4.74E+02+	1.34E+04±1.93E+03+	1.13E+04±1.14E+03+	1.11E+04±6.07E+02+	8.97E+03±5.04E+02
F10	5.88E+04±1.30E+04+	5.32E+04±9.66E+03+	3.04E+02±6.07E+01-	6.38E+04±1.24E+04+	1.37E+04±3.59E+03
F11	1.13E+07±3.37E+06+	2.27E+07±8.61E+06+	9.44E+05±2.76E+05-	1.72E+07±5.33E+06+	5.51E+06±2.09E+06
F12	2.54E+03±1.18E+03=	2.47E+03±1.89E+03=	4.94E+03±3.76E+03=	3.14E+03±1.97E+03=	5.32E+03±5.78E+03
F13	2.94E+06±1.29E+06+	1.44E+07±7.96E+06+	1.74E+05±5.62E+04-	6.13E+06±1.97E+06+	1.06E+06±3.60E+05
F14	8.23E+02±3.18E+02-	8.46E+02±6.39E+02-	2.01E+03±2.56E+03=	1.82E+03±1.58E+03=	3.35E+03±3.94E+03
F15	3.20E+03±3.05E+02+	3.10E+03±3.92E+02+	3.42E+03±5.34E+02+	3.19E+03±2.72E+02+	2.51E+03±3.32E+02
F16	2.47E+03±2.65E+02+	3.26E+03±6.55E+02+	2.76E+03±5.01E+02+	2.66E+03±3.28E+02+	2.28E+03±3.50E+02
F17	2.43E+06±7.73E+05+	1.15E+07±5.94E+06+	7.01E+05±2.91E+05-	5.21E+06±1.07E+06+	8.82E+05±3.34E+05
F18	7.36E+02±5.12E+02=	1.52E+03±1.48E+03=	1.42E+03±2.02E+03=	3.20E+03±1.96E+03=	3.65E+03±4.18E+03
F19	2.35E+03±2.24E+02+	3.39E+03±6.74E+02+	2.16E+03±3.70E+02+	2.48E+03±3.24E+02+	1.93E+03±3.80E+02
F20	7.92E+02±4.00E+01+	5.35E+02±5.21E+01+	5.68E+02±5.39E+01+	5.76E+02±2.88E+01+	3.33E+02±1.18E+01
F21	1.20E+04±6.72E+02+	1.48E+04±2.09E+03+	1.25E+04±1.10E+03+	1.21E+04±2.29E+03+	9.37E+03±5.17E+02
F22	8.50E+02±2.45E+01+	6.70E+02±2.54E+01+	7.63E+02±2.67E+01+	7.18E+02±2.10E+01+	6.03E+02±1.35E+01
F23	1.44E+03±3.54E+01+	1.12E+03±3.89E+01+	1.30E+03±6.66E+01+	1.20E+03±3.30E+01+	1.01E+03±1.39E+01
F24	7.29E+02±2.94E+01+	7.27E+02±4.53E+01+	7.95E+02±6.16E+01+	7.90E+02±3.23E+01+	6.58E+02±4.99E+01
F25	8.34E+03±3.17E+03+	5.66E+03±1.02E+03+	9.42E+03±3.74E+03+	6.65E+03±2.48E+02+	4.38E+03±1.64E+02
F26	8.04E+02±2.86E+01+	7.27E+02±3.69E+01+	8.46E+02±6.34E+01+	7.50E+02±2.16E+01+	6.40E+02±1.54E+01
F27	5.93E+02±1.16E+01+	5.83E+02±1.87E+01+	5.60E+02±3.54E+01=	6.34E+02±2.11E+01+	5.51E+02±1.07E+01
F28	3.89E+03±2.70E+02+	3.68E+03±3.95E+02+	2.96E+03±4.65E+02+	3.47E+03±2.29E+02+	2.50E+03±5.41E+02
F29	9.53E+03±3.58E+03-	1.22E+04±2.77E+03=	6.42E+03±3.18E+03-	1.10E+04±2.93E+03-	1.31E+04±3.67E+03
	+/-/-	22/3/4	22/4/3	17/4/8	23/3/3

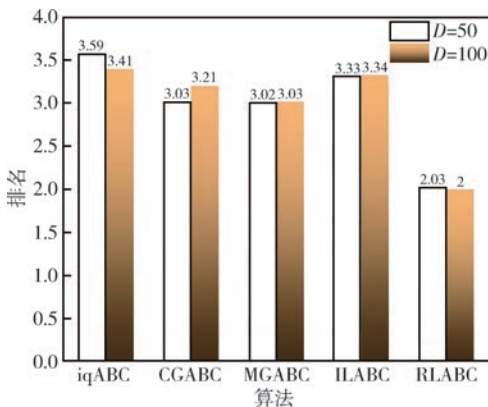


图 15 多维度测试中不同算法的平均排名

4.6 在实际优化问题上对比

为进一步验证 RLABC 性能,我们将其用于 2 个实际优化问题:(1)调频声波的参数估计问题(Parameter Estimation for Frequency-modulated Sound Waves, PEFSW)^[18];(2)压力容器设计问题

(Pressure Vessel Design, PVD)^[37]. 实验中,我们将 4.4 节和 4.5 节中涉及的 8 种对比算法与 RLABC 进行性能比较,算法的参数设置与前面实验保持相同. 下面分别介绍在这两个优化问题上的相关实验.

(1)优化 PEFSW 问题

PEFSW 问题常见于音频系统或雷达系统等领域的信号调制过程,目的是从一组给定的声波信号中估计出信号的幅度、频率、相位和调制指数等重要参数,这些参数对于信号的分析 and 处理有重要意义. 求解该问题的关键是找到一个最优的向量 $X = \{a_1, \omega_1, a_2, \omega_2, a_3, \omega_3\}$,使得由式(13)生成的估计声波与式(14)给定的目标声波尽可能相似,即估计误差最小^[18].

$$y(t) = a_1 \sin(\omega_1 t \theta + a_2 \sin(\omega_2 t \theta + a_3 \sin(\omega_3 t \theta))) \quad (13)$$

$$y_0(t) = 1.0 \sin(0.5 t \theta + 1.5 \sin(4.8 t \theta + 2.0 \sin(4.9 t \theta))) \quad (14)$$

其中 $\theta=2\pi/100, X_i \in [-6, 4, 6, 35]$. 因此, PEFSW 问题的优化目标可表达为最小化估计声波与目标声波之间的平方误差之和, 如下式所示:

$$f(X) = \sum_{t=0}^{100} (y(t) - y_0(t))^2 \quad (15)$$

该问题是一个高度非线性、存在大量局部最优解的多峰问题, 最优值为 $f(X)=0$.

实验中, 算法的终止条件设置为 $MaxFEs = 200,000$; 每种算法独立优化该问题 30 次, 记录 30 次运行中的最小值、最大值、均值以及标准差. 表 13 给出了所涉及的 9 种算法结果, 其中最好结果用粗体凸显. 从表中可看出, RLABC 是所有算法中表现最好的, 且精度较其他算法有较大提高. 这个实验表明, RLABC 在 PEFSW 问题上的性能要优于其他 8 种对比算法.

表 13 优化 PEFSW 问题的实验结果

算法	均值	最小值	最大值	标准差
NSABC	6.38E+00	4.02E-01	1.27E+01	3.90E+00
ABCNG	1.13E+01	4.55E-02	1.59E+01	3.59E+00
ABC-MNT	4.12E+00	9.09E-29	1.33E+01	4.99E+00
ABC-ANT	4.49E+00	3.53E-10	1.27E+01	5.29E+00
iqABC	9.84E+00	5.11E-01	1.55E+01	5.07E+00
CGABC	1.29E+01	4.79E-01	2.28E+01	6.71E+00
MGABC	7.58E+00	0.00E+00	1.60E+01	5.68E+00
ILABC	1.18E+01	3.98E+00	1.83E+01	3.52E+00
RLABC	2.63E+00	0.00E+00	1.56E+01	5.34E+00

(2) 优化 PVD 问题

在工业环境中, 压力容器应有较高的安全系数, 其设计起到了至关重要的作用. 通常, 对于圆柱形的压力容器(如图 16 所示), 其设计过程涉及了 4 个变量: 桶厚 x_1 、头厚 x_2 、桶内径 x_3 和桶长 x_4 . 该问题的优化目标是, 如何在给定的约束条件下对 4 个变量进行取值, 使得总体制造成本最低. 下式给出了具体的优化目标函数^[37]:

$$\min f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (16)$$

服从如下约束:

$$\begin{cases} g_1(x) = -x_1 + 0.0193x_3 \leq 0 \\ g_2(x) = -x_3 + 0.00954x_3 \leq 0 \\ g_3(x) = -\pi x_3^2x_4 - \frac{3}{4}\pi x_3^3 + 1296000 \leq 0 \\ g_4(x) = x_4 - 240 \leq 0 \end{cases} \quad (17)$$

其中, $0.1 \leq x_i \leq 100, i=1, 2; 10 \leq x_i \leq 200, i=3, 4$.

因该问题是一个带约束的优化问题, 本文算法

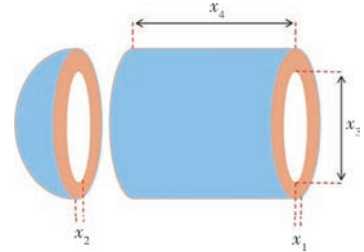


图 16 PVD 问题示意图

和对比算法无法直接用于求解该问题, 因此我们采用了可行性规则技术来处理约束. 对于任意给定的 2 个解 X_a 和 X_b 进行对比, 可行性规则技术主要包括 3 条对比准则: (1) 若 X_a 和 X_b 均为可行解, 则目标函数值更优的解胜出; (2) 若 X_a 和 X_b 均为不可行解, 则约束违反程度小的解胜出; (3) 若一个为可行解, 而另一个是不可行解, 则可行解胜出.

实验中, 我们对本文算法和对比算法均采用了可行性规则技术. 算法的终止条件设置为 $MaxFEs = 10000 \cdot D$; 同样地, 每种算法独立优化该问题 30 次, 记录 30 次运行中的最小值、最大值、均值以及标准差. 在表 14 中, 我们给出了实验中涉及的 9 种算法的实验结果, 可看出本文算法同样取得了最好结果. 值得注意的是, 不同于 PEFSW 问题, iqABC 在该问题上的结果同样也不错.

表 14 优化 PVD 问题的实验结果

算法	均值	最小值	最大值	标准差
NSABC	3.36E+03	2.66E+03	3.39E+03	9.46E+01
ABCNG	3.31E+03	9.21E+02	3.54E+03	4.45E+02
ABC-MNT	3.37E+03	3.37E+03	3.41E+03	7.29E+00
ABC-ANT	2.81E+03	1.63E+03	3.31E+03	3.81E+02
iqABC	2.44E+03	6.76E+02	3.37E+03	8.45E+02
CGABC	3.38E+03	3.37E+03	3.40E+03	8.50E+00
MGABC	8.61E+04	1.64E+04	2.04E+05	4.37E+04
ILABC	3.38E+03	3.37E+03	3.48E+03	2.15E+01
RLABC	2.12E+03	6.31E+02	2.97E+03	3.17E+02

5 结 论

如何结合邻域拓扑来提高 ABC 性能是近年来群智能优化算法领域的研究热点. 不同于现有的相关工作, 本文提出的自适应多邻域 ABC, 简称 RLABC, 主要有两点创新之处: (1) 在个体层次上实现了多邻域拓扑, 这种细粒度的邻域使用方式能充分发挥出不同邻域拓扑的优势, 进一步增强邻域信息对算法搜索过程的引导作用; (2) 采用了强化学习

中的 Q-learning 方法来自适应选择邻域拓扑, 能根据个体与环境的交互情况来实现高效的自适应选择。

在 CEC2013 和 CEC2017 两套测试集以及两个实际优化问题上进行大量实验验证, 可得到如下 5 点结论: (1) 邻域拓扑的切换频率宜适中, 不应过于频繁或太少; (2) 对于邻域拓扑的选择, 强化学习要好于随机方式和历史经验的方式; (3) 在不同类型的测试函数上, 不会出现某一邻域拓扑的使用频率过低, 强化学习能合理选择对应的邻域拓扑; (4) 在侦察蜂阶段, 引入的双精英搜索策略比随机初始化的方式更有利于提高 RLABC 性能; (5) 与 4 种邻域相关 ABC 和 4 种其他知名改进 ABC 的对比表明 RLABC 性能有很强的竞争力。然而, 需指出的是, RLABC 在结合强化学习进行多邻域自适应选择时, 忽略了智能体之间的相互关联, 这在一定程度上会影响智能体决策的准确性。在未来, 我们将尝试结合一些新颖的分布式强化学习模型来解决该问题。

致 谢 感谢武汉大学吴志健教授对本文提出的中肯意见, 使得本文的质量得以提升! 感谢南昌工程学院王晖教授在算法对比实验部分提供的帮助! 感谢江西师范大学倪文龙教授在论文返修阶段中对强化学习部分的讨论!

参 考 文 献

- [1] Wang Feng, Zhang Heng, Han Meng-Chen, et al. Co-evolution based mixed-variable multi-objective particle swarm optimization for UAV cooperative multi-task allocation problem. *Chinese Journal of Computers*, 2021, 44(10): 1967-1983 (in Chinese)
(王峰, 张衡, 韩孟臣等. 基于协同进化的混合变量多目标粒子群优化算法求解无人机协同多任务分配问题. *计算机学报*, 2021, 44(10): 1967-1983)
- [2] Tao Xin-Min, Guo Wen-Jie, Li Xiang-Ke, et al. Density peak based multi subpopulation particle swarm optimization with dimensionally reset strategy. *Journal of Software*, 2023, 34(4): 1850-1869 (in Chinese)
(陶新民, 郭文杰, 李向可等. 基于密度峰值的依维度重置多种群粒子群算法. *软件学报*, 2023, 34(4): 1850-1869)
- [3] Liu J, Anavatti S, Garratt M, et al. Multi-operator continuous ant colony optimisation for real world problems. *Swarm and Evolutionary Computation*, 2022, 69: 100984
- [4] Wang C, Shang P, Shen P. An improved artificial bee colony algorithm based on Bayesian estimation. *Complex & Intelligent Systems*, 2022, 8(6): 4971-4991
- [5] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization; Artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 2007, 39(3): 459-471
- [6] Zhu S, Pun C M, Zhu H, et al. An artificial bee colony algorithm with a balance strategy for wireless sensor network. *Applied Soft Computing*, 2023, 136: 110083
- [7] Cui Y, Hu W, Rahmani A. A reinforcement learning based artificial bee colony algorithm with application in robot path planning. *Expert Systems with Applications*, 2022, 203: 117389
- [8] Lü Yang, Qian Bin, Hu Rong, et al. Enhanced artificial bee colony algorithm to solve semiconductor final test scheduling problem. *Acta Electronica Sinica*, 2021, 49(9): 1708-1715 (in Chinese)
(吕阳, 钱斌, 胡蓉等. 增强人工蜂群算法求解半导体最终测试调度问题. *电子学报*, 2021, 49(9): 1708-1715)
- [9] Zhou X, Lu J, Huang J, et al. Enhancing artificial bee colony algorithm with multi-elite guidance. *Information Sciences*, 2021, 543: 242-258
- [10] 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. *Information Sciences*, 2018, 442: 54-71
- [11] Zhu G, Kwong S. Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied Mathematics & Computation*, 2010, 217(7): 3166-3173
- [12] Peng H, Deng C, Wu Z. Best neighbor-guided artificial bee colony algorithm for continuous optimization problems. *Soft Computing*, 2019, 23(18): 8723-8740
- [13] Wang H, Wang W, Xiao S, et al. Improving artificial bee colony algorithm using a new neighborhood selection mechanism. *Information Sciences*, 2020, 527: 227-240
- [14] Zhou X, Wu Y, Zhong M, et al. Artificial bee colony algorithm based on adaptive neighborhood topologies. *Information Sciences*, 2022, 610: 1078-1101
- [15] Xu Y, Pi D. A reinforcement learning-based communication topology in particle swarm optimization. *Neural Computing & Applications*. 2020, 32(14): 10007-10032
- [16] Karaboga D, Gorkemli B, Ozturk C, et al. A comprehensive survey; Artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review*, 2014, 42: 21-57
- [17] Aslan S, Badem H, Karaboga D. Improved quick artificial bee colony (iqABC) algorithm for global optimization. *Soft Computing*, 2019, 23(24): 13161-13182
- [18] Zhou X, Wu Z, Wang H, et al. Gaussian bare-bones artificial bee colony algorithm. *Soft Computing*, 2016, 20(3): 907-924
- [19] Zhang M, Tian N, Palade V, et al. Cellular artificial bee colony algorithm with Gaussian distribution. *Information Sciences*, 2018, 462: 374-401
- [20] Xiao S, Wang H, Wang W, et al. Artificial bee colony algo-

- rithm based on adaptive neighborhood search and Gaussian perturbation. *Applied Soft Computing*, 2021, 100: 106955
- [21] Zhou X, Wu Y, Zhong M, et al. Artificial bee colony algorithm based on multiple neighborhood topologies. *Applied Soft Computing*, 2021, 111: 107697
- [22] Kang F, Li J, Ma Z. Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions. *Information Sciences*, 2011, 181(16): 3508-3531
- [23] Jadon S S, Tiwari R, Sharma H, et al. Hybrid artificial bee colony algorithm with differential evolution. *Applied Soft Computing*, 2017, 58: 11-24
- [24] Gao W, Huang L, Liu S, et al. Artificial bee colony algorithm based on information learning. *IEEE Transactions on Cybernetics*, 2015, 45(12): 2827-2839
- [25] Carleo G, Cirac I, Cranmer K, et al. Machine learning and the physical sciences. *Reviews of Modern Physics*, 2019, 91(4): 045002
- [26] Botvinick M, Ritter S, Wang J X, et al. Reinforcement learning, fast and slow. *Trends in Cognitive Sciences*, 2019, 23(5): 408-422
- [27] Rakshit P, Konar A, Bhowmik P, et al. Realization of an adaptive memetic algorithm using differential evolution and Q-learning: A case study in multirobot path planning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2013, 43(4): 814-831
- [28] Hu Z, Gong W, Li S. Reinforcement learning-based differential evolution for parameters extraction of photovoltaic models. *Energy Reports*, 2021, 7: 916-928
- [29] Chen R, Yang B, Li S, et al. A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem. *Computers & Industrial Engineering*, 2020, 149: 106778
- [30] Samma H, Lim C P, Saleh J M. A new reinforcement learning-based memetic particle swarm optimizer. *Applied Soft Computing*, 2016, 43: 276-297
- [31] Nussenbaum K, Hartley C. A Reinforcement learning across development: What insights can we draw from a decade of research? *Developmental Cognitive Neuroscience*, 2019, 40: 100733
- [32] Gao W, Liu S, Huang L. A global best artificial bee colony algorithm for global optimization. *Journal of Computational and Applied Mathematics*, 2012, 236(11): 2741-2753
- [33] Zhou X, Wang H, Wang M, et al. Selection mechanism in artificial bee colony algorithm: A comparative study on numerical benchmark problems//*Proceedings of the 24th International Conference on Neural Information Processing (ICONIP 2017)*. Guangzhou, China, 2017: 61-69
- [34] 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. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report: 201212.34*, 2013
- [35] Awad N H, Ali M Z, Suganthan P N, et al. Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization. *School of EEE, Nanyang Technological University, Singapore and School of Computer Information Systems, Jordan University of Science and Technology, Jordan and School of Electrical Engineering, Zhengzhou University, Zhengzhou, China, Technical Report*, 2017
- [36] Qin A K, Huang V L, Suganthan P N. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 2008, 13(2): 398-417
- [37] Yang X, Li H, Huang Y. An adaptive dynamic multi-swarm particle swarm optimization with stagnation detection and spatial exclusion for solving continuous optimization problems. *Engineering Applications of Artificial Intelligence*, 2023, 123: 106215.



ZHOU Xin-Yu, Ph. D., associate professor. His main research interests include intelligent computing and machine learning.

YIN Zi-Yue, undergraduate. His main research interest is swarm intelligent optimization algorithms.

GAO Wei-Feng, Ph. D., professor, national young talent. His main research interests include intelligent computing and machine learning.

TAN Gui-Shen, master. His main research interest is swarm intelligent optimization algorithms.

YI Yu-Gen, Ph. D., associate professor. His main research interests include pattern recognition and deep learning.

Background

In recent years, global optimization problems have been becoming increasingly difficult to solve due to the rapid development of application scenarios, thus most of them show some unfavorable properties, such as nondifferentiable, highly nonlinear, and multimodal. For

these kinds of problems, the traditional mathematical optimization methods face great challenges, especially for the gradient-based methods. In this scenario, the swarm intelligent optimization (SIO) algorithm is a good alternative, which almost has no requirement on the problem

properties. As a representative paradigm of the SIO algorithm, the artificial bee colony (ABC) algorithm has attracted much attention for its simple structure yet good performance. Therefore, how to improve the ABC algorithm or use it to solve real-world problems is a hot research topic in the community of SIO.

As an emerging idea of improving the ABC algorithm, the neighborhood-based ABC variants have been shown promising performance, which aim to utilize the neighborhood information for guiding the search. However, for the existing related works, only a single neighborhood topology is used, resulting in limited algorithm performance. In this work, we attempt to use multiple neighborhood topologies for the neighborhood-based ABC variant, so that the complementary advantages of different neighborhood topologies can be fully taken. To reasonably assign the most suitable neighborhood topology for different individuals, the

reinforcement learning is used by considering the individuals as agents, thus the neighborhood topology with the highest reward can be chosen. To the best of our knowledge, our work may be the first attempt to use multiple neighborhood topologies at the individual level in the community of ABC. Extensive experimental results confirm that our work can effectively enhance the performance of neighborhood-based ABC variant.

This work is supported by the National Natural Science Foundation of China under Grant Nos. 62366022, 61966019, 62276202, and 62062040, the Outstanding Youth Project of Jiangxi Natural Science Foundation under Grant No. 20212ACB212003, the Jiangxi Province Key Subject Academic and Technical Leader Funding Project under Grant No. 20212BCJ23017, and the Jiangxi Provincial Natural Science Foundation under Grant No. 20232BAB202048.