调研各类演化算法并进行归纳和分析

**杨祎勃**1 **李麒飞**2 **陈鹏如**3 **杨一轩**4 **张阳**5 **钟洋**6

1 南方科技大学 广东省 深圳市 518055

2 南方科技大学 广东省 深圳市 518055

3 南方科技大学 广东省 深圳市 518055

4 南方科技大学 广东省 深圳市 518055

5 南方科技大学 广东省 深圳市 518055

6 南方科技大学 广东省 深圳市 518055

**摘 要：演化算法（Evolutionary Algorithms，EAs）以其模拟自然进化机制的全局优化特性在复杂系统研究中备受瞩目。本文对各类演化算法进行深入调研，以提供研究人员全面系统的理论框架和应用参考。在演化算法的基本理论方面，文中深度解析了其核心原理，包括自然选择、遗传、变异等生物进化机制。这些机制使演化算法能够高效应对传统算法难以解决的非线性、多峰、高维问题，为优化复杂系统提供新的思路和方法。文中还追溯了演化算法的发展历程，从最初的仿生启发到如今的跨学科研究领域。通过深入分析不同演化算法的演进轨迹，揭示了其在算法性能和应用范围上的优劣之处，突显了其作为实际问题解决工具的重要性。最后，文中系统总结了演化算法在工程优化、人工智能、生态模型等多领域的广泛应用。通过案例研究和实际应用，突显了演化算法在解决复杂问题和优化系统中的卓越效果，为研究人员提供了更全面的视角，助力更好理解演化算法的实际应用场景和潜在贡献。综合研究各类演化算法，本文旨在推动演化算法在复杂系统优化中更广泛的应用，为相关研究领域提供有力支持。**

**关键词：** 演化算法；全局优化；生物进化机制；复杂系统研究；发展历程；广泛应用场景。

**中图法分类号**  TP183

**Survey, Summarization, and Analysis of Various Evolutionary Algorithms**

Yang Yibo1 ，Li Qifei2 ，Chen Pengru2 ，Yang Yixuan4 ，Zhang Yang5 and Zhong Yang6

1 School of SUSTech，Shenzhen 518055， China

2 School of SUSTech，Shenzhen 518055， China

3 School of SUSTech，Shenzhen 518055， China

4 School of SUSTech，Shenzhen 518055， China

5 School of SUSTech，Shenzhen 518055， China

6 School of SUSTech，Shenzhen 518055， China

**概要：**本文综述了演化算法（Evolutionary Algorithms, EAs）的基本理论、发展历程及广泛应用场景。演化算法，源于20世纪中叶的生物进化原理，已成为跨学科研究的重要领域。文章介绍了遗传算法（GA）、进化策略（ES）、遗传编程（GP）、差分进化（DE）和粒子群优化（PSO）等主要演化算法，并分析了它们在优化问题、机器学习、艺术创作等领域的应用。文章还探讨了不同算法的性能对比、知名开源演化算法工具包的分析，以及未来演化算法的发展趋势，强调了其在理论和实践上的重要性。

**关键词：**演化计算、优化算法、各类优化算法的应用、UFLP问题、工程优化、人工智能、算法工具包

引言  
在复杂系统的探索与研究中，演化算法（Evolutionary Algorithms，EAs）作为一种高效的全局优化策略，引起了科研人员的广泛兴趣。这类算法以生物进化机制为蓝本，通过模拟自然选择、遗传、变异等生物进化过程，解决了许多传统算法难以克服的非线性、多峰、高维问题。随着计算能力的提升和理论研究的深入，演化算法已经从简单的仿生启发发展成为一门跨学科的研究领域，其应用范围覆盖了工程优化、人工智能、生态模型等众多领域。本文将综述演化算法的基本理论、发展历程及其广泛的应用场景，旨在为研究人员提供一个全面、系统的理论框架和应用参考。

演化算法的历史和背景  
演化算法的历史始于20世纪中叶，当时科学家们受到达尔文的自然选择和孟德尔的遗传学说的启发，开始尝试将生物进化原理应用于算法设计之中。1975年，美国密歇根大学的约翰·霍兰德教授提出了系统化的遗传算法，并在其后的几十年中不断完善。几乎同时，德国科学家英戈·雷申贝格发展了进化策略，这两种算法成为了演化算法研究的两大支柱。进入20世纪80年代和90年代，随着计算机技术的飞速发展，演化算法迎来了发展的黄金时期。遗传编程作为一种新型的演化算法形式被提出，其将算法的演化延伸到了程序的自动生成。同时，差分进化和粒子群优化等算法也相继问世，进一步拓宽了演化算法的研究与应用领域。

演化算法的分类和原理  
演化算法是一大类算法的总称，它们可以根据个体表示、种群结构、适应度函数和变异算子的不同而被分为不同的类别。主要包括：

遗传算法（Genetic Algorithms, GA）：这是最早期和最基本的演化算法之一。它使用一个编码的字符串（通常是二进制编码）来表示问题的解，通过选择、交叉和变异等操作模拟自然进化过程。

进化策略（Evolution Strategies, ES）：进化策略主要用于实数优化问题，其通过对解向量的参数进行变异和重组，来优化目标函数。

遗传编程（Genetic Programming, GP）：遗传编程扩展了遗传算法的概念，不仅仅是寻找一个静态的解，而是演化出能够执行特定任务的计算机程序

差分进化（Differential Evolution, DE）：差分进化通过考虑种群中个体间的差异来指导搜索方向，特别适用于多峰和高维的连续空间优化问题。

粒子群优化（Particle Swarm Optimization, PSO）：粒子群优化算法模拟鸟群狩猎行为，每个粒子代表潜在的解，通过个体与群体间的信息共享来指导搜索过程。演化算法的基本原理围绕着“适者生存”的理念。算法开始时随机生成一个解的种群，然后通过适应度函数评价每个个体的质量。根据个体的适应度进行选择，适应度高的个体有更大的机会被选中参与下一代的繁殖。通过交叉和变异操作产生新个体，这个过程模拟了生物的遗传和变异。通过这样的迭代过程，种群中的个体将逐渐向优解聚集。演化算法的效能受到多种因素的影响，包括编码方式、适应度函数的设计、选择压力、交叉和变异算子的设置等。在实际应用中，这些参数的选择和调整需要根据具体问题的性质来决定，这也是演化算法研究中的重要课题之一。随着研究的深入，演化算法已经发展出许多变种和改进方法，使其在解决现实世界问题中的适应性和鲁棒性不断增强。演化算法最基本的模型为模拟达尔文进化论的遗传算法（Genetic Algorithm，简称GA），最早由J.Holland教授于1975年提出。遗传算法中种群每个个体都是解空间上的一个可行解，通过模拟生物的进化过程，进行遗传、变异、交叉、复制从而在解空间内搜索最优解。基本过程及原理为：

初始化候选解（种群初始化）：编码，解的问题表示成遗传空间的染色体或者个体，通过数字编码表示。初始种群随机生成。

通过适应度函数计算候选解的适应度（个体评价）：依据求解问题设计适应度函数，对种群每个个体计算适应度。适应度函数的复杂度直接影响算法的性能。

依据适应度决定解的去与留（选择）：依据计算出的适应度的大小评估解，并选择剔除或保留解。

对保留的解进行操作生成新的候选解（遗传变异）：交叉手段将两个父体部分结构进行重组，变异手段直接将个体改变，两种方式都是为了生成新解。基于这个基本模型，将个体表达为解向量，便诞生差分进化算法（Differential Evolution, DE）；将个体表达为粒子群，便诞生粒子群优化（Particle Swarm Optimization, PSO）；将问题分解，分组优化，便诞生合作协同进化算法（Cooperative Co-Evolution Algorithm，CCEAs）；将概率模型用于新解生成，候选解反作用于概率模型的优化，便诞生分布估计算法（Estimation of Distribution Algorithm，EDA）；将该算法运用于计算机程序的生成，便诞生遗传编程（Genetic Programming, GP）。

演化算法的应用案例分析  
演化算法在各类优化问题、机器学习、艺术创作等领域都能发挥重要作用[1]。

优化问题  
演化算法在解决复杂的优化问题中发挥着重要作用。例如：

结构工程优化  
在结构工程领域，遗传算法被用于优化建筑设计，以提高材料的利用效率和减少成本。例如，遗传算法可以用于桥梁设计，通过调整桥梁的形状和材料分配，以达到最佳的承重和稳定性[2]。

运输调度优化  
啥的进化策略被应用于优化公共交通系统的调度。通过模拟不同的调度方案，可以找到减少延误和提高效率的最佳方案[3]。

机器学习  
在机器学习领域，演化算法用于优化算法模型和参数，提高学习效率和准确性[4]。

神经网络结构优化  
遗传编程被用于自动设计和优化神经网络结构。例如，通过遗传编程演化出的深度学习模型在图像识别任务中表现出优越的性能。

特征选择  
遗传算法可用于特征选择，以提高模型的预测能力。通过选择最相关的特征子集，可以在不牺牲准确性的情况下减少模型的复杂度。

艺术创作  
演化算法在艺术创作中也有独特的应用，它们被用于生成创新的艺术作品。

音乐创作  
遗传算法被用于生成音乐旋律和和声。例如，算法可以模拟不同的音乐风格和结构，创作出独特的音乐作品[5]。

视觉艺术  
在视觉艺术领域，进化策略被用于生成复杂的图形设计。艺术家可以通过设置不同的参数和规则，使用算法创作出独一无二的视觉作品[6][7]。

各类演化算法的性能比较和分析  
在深入探索演化算法时，理解不同算法的性能特点至关重要。本节将对主要演化算法——遗传算法（GA）、进化策略（ES）、遗传编程（GP）、差分进化（DE）和粒子群优化（PSO）——进行详细的比较和分析。

遗传算法（GA）  
适用范围：GA适用于广泛的优化问题，特别是离散问题，如排列优化和调度问题。性能特点：GA在全局搜索能力方面表现出色，但在局部搜索和收敛速度方面可能不如其他算法高效。优点：简单、易于理解和实现，适应性强。缺点：容易陷入局部最优，参数调整较为复杂。

进化策略（ES）  
适用范围：ES特别适用于连续参数的优化问题。性能特点：在处理高维连续空间问题时，ES显示出良好的收敛性和稳定性。优点：参数自适应调整，对初始种群不敏感。缺点：在多峰问题上可能表现不佳，计算复杂度相对较高。

遗传编程（GP）  
适用范围：GP适合解决需要程序设计或自动生成代码的问题。性能特点：在自动生成复杂结构（如程序或电路）方面非效。优点：能够发现新的、未预见的解决方案设计。缺点：解空间巨大，计算成本高，难以调整和控制。

差分进化（DE）  
适用范围：DE非常适合连续空间的优化问题，尤其是多峰问题。性能特点：在保持多样性的同时保证了良好的收敛速度，特别适用于全局优化。优点：算法简单，容易实现，参数较少。缺点：在某些类型的问题上可能过于激进，导致早熟收敛。

粒子群优化（PSO）  
适用范围：PSO适用于连续优化问题，特别是高维搜索空间的问题。性能特点：在搜索空间的探索和利用方面取得了良好平衡，收敛速度快。优点：实现简单，调整参数少，易于并行化。缺点：可能陷入局部最优，对于多峰问题的适应性较差。  
不同类型的演化算法各有优势和局限性。在选择合适的算法时，需要考虑问题的特性、算法的复杂度、求解的精度和效率等多个因素。总的来说，没有绝对完美的算法，最佳选择往往依赖于具体问题和应用场景。随着演化算法理论和技术的不断进步，未来有望出现更加强大和适应性更强的算法。

5.1 UFLP问题（无容量仓库选址问题）下不同优化算法的对比测试  
无容量设施选址问题（Uncapacitated Facility Location Problem，UFLP）是运筹学和组合优化中的一个经典问题。在这个问题中，我们要决定在哪些地点开设设施，并确定如何将客户分配到这些设施上，以最小化设施的开设成本和客户服务成本的总和[9]。UFLP的特点是设施的容量没有限制，即每个设施可以服务任意数量的客户而不会增加成本。这个问题在许多实际应用中都有体现，如仓库位置的选择、零售店的分布规划等。UFLP问题通常可以用数学规划方法来表达和求解。它是一个NP难问题，意味着目前没有已知的多项式时间算法能够解决所有的UFLP实例。因此，研究者常常采用启发式或近似算法来求解实际问题，如遗传算法、模拟退火、禁忌搜索等。  
我们选取了几个代表性的算法（如图1，2，3，4所示），包括粒子群优化（DPSO）、自适应差分进化（SabDE）和进化游戏理论优化算法（EGTOA）。在cap71、capa、MQ1和MR1四个不同规模与特点的样例中，通过迭代次数与目标值的关系，我们观察了算法性能的动态变化[11][12][13]。从cap71实例中可以看出，EGTOA在早期迭代中快速下降，显示出强大的全局搜索能力，但随着迭代的深入，SabDE展现出更稳定的收敛性，最终取得了更低的目标值。在MQ1实例中，DPSO在初始阶段的表现优于其他算法，然而EGTOA和SabDE最终达到了更优的解，说明它们在后期的局部搜索中更为有效。在更大规模的capa和MR1实例中，算法之间的性能差异更为显著。特别是在MR1实例中，EGTOA显著超越了其他算法，这可能归因于其在处理高维搜索空间中的有效性。此外，我们也观察到在所有测试实例中，启用了拉格朗日松弛（Lagrangian Relaxation, LAV）技术的算法变种通常比原始算法表现出更快的收敛速度和更好的最终性能，这表明LAV技术在UFLP问题中应用是有效的。

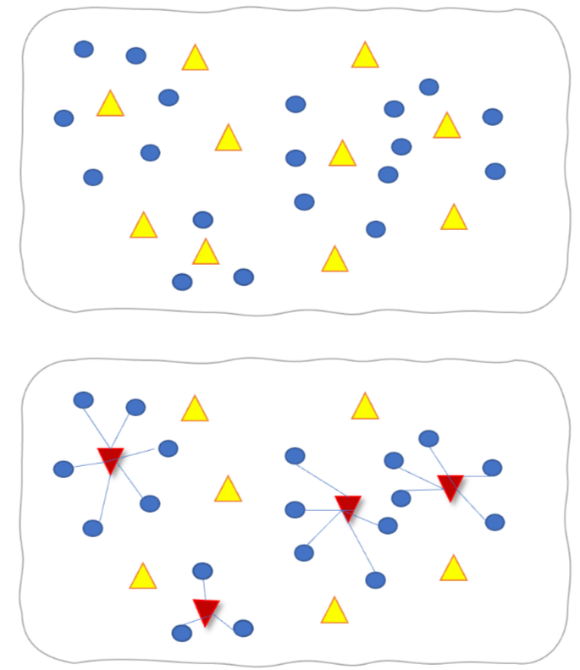


图1

Fig.1

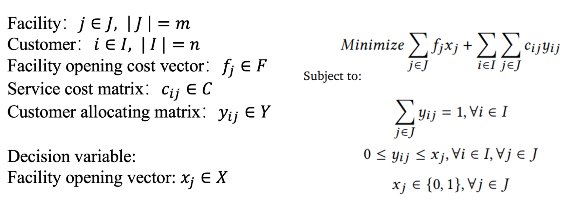


图2

Fig.2

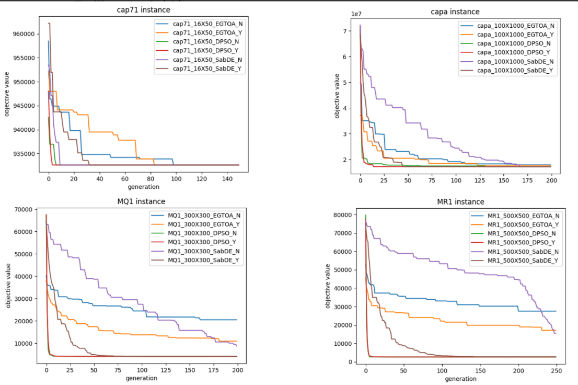


图3

Fig.3

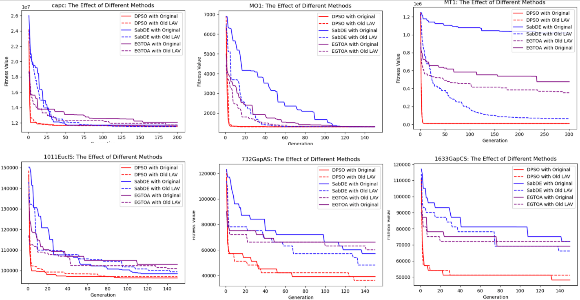


图4

Fig.4

5.2 知名开源演化算法工具包的对比测试  
在不同的具体问题上，不同的演化算法实现的效果也可能不同，我们调研了cmaes，pycma，DEAP，PySwarms，nevergrad，EvoJAX等知名演化算法工具包，以及我们自己实现的NCS算法（负相关搜索）[8]。我们的研究流程遵循了以下几个步骤：1. 开源优化算法库概览：调研过的主要开源库名单以及每个库的主要特点。2. 标准接口设计：常见的标准接口设计模式。3. 知名算法包特点分析：每个算法工具包的特色、优点与不足对比各个库的接口设计方式；如何定义问题；如何配置算法；如何执行优化；如何分析结果；算法评测标准；各个库采用的评测标准或方法以及对比不同库之间的评测差异和特点。4. 综合比较：接口设计比较：各个库的接口设计方式；易用性比较：哪些库更容易调用、文档更详细合理；通用性：哪些库能处理的问题类型更多；可拓展性比较：哪些库更便于更改参数、更改数据范围以及稳定性比较：哪些库能更稳定的找出最优解。  
探讨演化算法的多样性和应用范围时，我们不仅关注算法的效率和准确性，还深入分析了它们在不同场景下的适用性。特别是在解决连续和离散优化问题的能力方面，演化算法展现出各自独特的优势和局限性。我们的研究涵盖了一系列知名的算法实现，包括CMA-ES、PyCMA、DEAP、PySwarms、Nevergrad和EvoJAX，以及我们自主实现的NCS-C算法（负相关搜索）[8]，结果如图5，6所示。 cmaes和pycma是对同一个具体演化算法实现出了不同的风格，一个更为精简易用，一个有着更强的通用性。针对CMA-ES和PyCMA，我们发现虽然它们基于同一演化算法原理，但展现出不同的实现风格，其中一个注重易用性，另一个具有更广泛的应用范围。PySwarms作为粒子群算法的实现，它的接口设计展示了算法的灵活性，特别是在处理连续和离散优化问题方面。DEAP、Nevergrad和EvoJAX则代表了更为全面的优化框架，适用于广泛的实际问题。pyswarms是对粒子群算法的实现，提供了适应度函数接口和搜索策略接口。cmaes、pycma、pyswarms属于单一算法工具包。DEAP是专为进化算法设计的模块化框架，nevergrad则为一个全面的优化库，EvoJAX则代表硬件加速的神经进化，他们三者都属于算法计算框架。由于CMA-ES的设计目标是连续优化，它的核心机制——协方差矩阵的适应，并不支持在离散空间上调整“步长”和“方向”，所以cmaes和pycma这两个工具包来说，它们只适合处理连续优化问题。- PSO算法本身是为连续空间设计的，其中每个“粒子”代表一个潜在的解，粒子的位置和速度都是连续值，但pyswarms提供了一个二进制粒子群优化(Binary PSO)的实现，专门用于离散优化问题。在二进制PSO中， 粒子的位置表示为 0 或 1 的二进制值，而不是连续值，这使得它可以应用于特征选择、组合优化等问题。 因此，pyswarms是一个灵活的工具，既善于处理连续优化问题，也可以解决离散优化问题。DEAP、nevergrad和EvoJAX作为通用性强大的算法框架工具包，他们都支持采用不同的优化算法解决实际问题，因此只要选择合适的算法，连续优化问题和离散优化问题都能使用它们来解决。此外，我们自己实现了一种新型的演化算法NCS-C[8]，并对其性能进行了着重测试，优化CEC2005的基准测试函数及其对应超参数设置如图7所示：

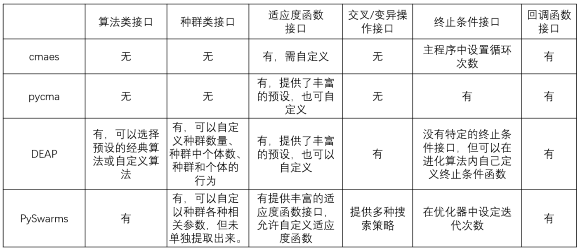


图5

Fig.5



图6

Fig.6

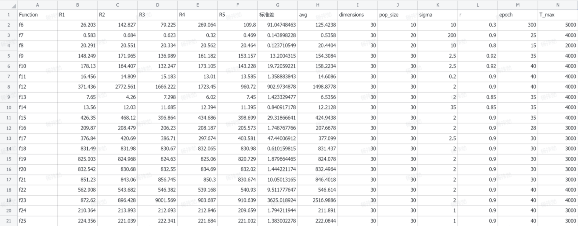


图7

Fig.7

几种知名演化算法的benchmark基准测试  
我们使用CEC2005的标准测试函数对不同算法进行测试[9]，包括遗传算法（GA）、粒子群优化器（PSO）[14]、进化策略（ES）[19]、差分进化（DE）[15]和散布搜索（SS）[16]。为了比较，我们选择了每种算法的最新版本，如GL-25 [17]、CLPSO [18]、CMA-ES [23]、SaDE [20]等。为了确保比较的公平性，所有比较算法的候选解都表示为实值参数。模拟退火（SA）和禁忌搜索（TS）都实现了相同的高斯变异算子，其中σi初始化为决策变量范围的十分之一。演化算法的参数r和epoch分别设为0.99和10。使用确定性冷却时间表来控制SA的温度。具体来说，温度初始化为1，然后每100次迭代以0.85的因子减少。在中使用了TS，最近生成的5个解决方案保留在禁忌列表中。根据其标准程序，SS的ReferenceSet大小设置为10。为了比较，使用了20个多模态连续问题的基准测试集（编号为F6-F25），这是CEC2005实数优化竞赛的一部分。每个问题的维度设置为30。每个比较算法在生成了300,000个解决方案后终止，迄今为止获得的最佳解决方案是最终解决方案。最终解决方案的质量以评估函数误差来衡量，即获得的解决方案的目标函数值与问题的最优解的目标函数值之间的差异（这些基准问题的最优解是已知的）。实验得出了不同主流优化算法在不同优化问题的效果情况分析，基本符合图8中的情况。

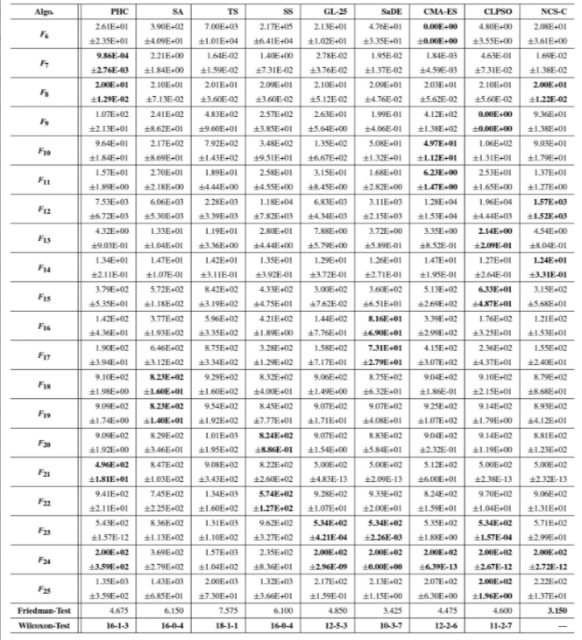


图8

Fig.8

未来展望和趋势  
演化算法（EA）作为一种高效的全局优化方法，在过去几十年中取得了显著的发展。但随着计算科学、人工智能和复杂系统理论的不断进步，未来的演化算法研究和应用将面临新的挑战和机遇。以下是对未来演化算法发展的几个主要预测和趋势：

多学科融合与创新  
随着生物学、物理学、社会学等多学科理论的深入，演化算法的研究将更加注重跨学科的融合和创新。例如，利用量子计算理论来提升演化算法的搜索能力和效率。

算法复合与集成  
预计将有更多的研究致力于将不同类型的演化算法或与其他优化方法（如机器学习、深度学习等）进行复合和集成，以充分利用各种算法的优势，提高解决复杂问题的能力。

自适应与智能化  
未来的演化算法将更加智能化和自适应，能够根据问题的特性和求解过程中的反馈信息自动调整算法参数，提升算法的适应性和鲁棒性。

大数据与云计算的应用  
在大数据和云计算的支持下，演化算法将能够处理更加庞大和复杂的数据集，尤其是在数据挖掘和机器学习方面的应用将会大大增加。

实时性与动态优化  
随着物联网和智能制造的发展，实时性和动态优化问题将成为未来演化算法研究的重点。这要求算法能够在动态变化的环境中快速有效地找到最优解。

理论与实践的深入融合  
未来的演化算法研究不仅要深入探索理论基础，更要注重与实际问题的结合，提高算法的实用性和市场应用价值。

可解释性与透明度  
在某些关键领域（如医疗、金融等），算法的可解释性和透明度日益受到重视。未来的研究将致力于提高演化算法的解释能力，使其决策过程更加透明和可信。

**结束语**

在对演化算法（Evolutionary Algorithms, EAs）的研究中，我们深入探讨了这些算法的基本理论、发展历程、分类和原理，并广泛应用于各种场景。演化算法的起源可追溯到20世纪中叶，它们模拟自然选择和遗传学说，已成为跨学科研究的重要领域。我们识别了主要的演化算法类型，包括遗传算法（GA）、进化策略（ES）、遗传编程（GP）、差分进化（DE）和粒子群优化（PSO），并分析了它们的特点和适用范围。

通过比较这些算法的性能，我们发现在选择合适的算法时，需要考虑问题的具体特性和算法的适用性。以不定容量设施选址问题（UFLP）为例，我们展示了不同算法在解决实际问题中的效果。此外，我们还调研和比较了多个知名的开源演化算法工具包，如CMA-ES、PyCMA、DEAP、PySwarms、Nevergrad、EvoJAX以及自研的NCS-C算法。

我们的研究预测了演化算法的未来发展趋势，包括多学科融合、算法复合与集成、自适应和智能化、在大数据和云计算中的应用、实时性和动态优化问题的解决，以及理论与实践的深入融合。我们还注意到，演化算法的可解释性和透明度将成为未来研究的重要方向。综上所述，演化算法作为一种全局优化策略，不仅在理论上有重要价值，而且在实际应用中展现出巨大潜力。

**参考文献**

[1] Katoch, S., Chauhan, S. S., & Kumar, V. (2020). A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications. <https://doi.org/10.1007/s11042-020-10139-6>

[2] 唐文艳, 顾元宪. (2002). 遗传算法在结构优化中的研究进展. 力学进展, 32(1): 26-40. doi: 10.6052/1000-0992-2002-1-J2000-084

[3] FENG Shuai, LIU Xiaoming. (2021). A survey of research on demand responsive transit and its route optimization

[4] ZENG Shui-Ling, SONG Luan-Jiao, XU Wei-Hong. (2005). Neural Network Structure Optimization Based on Genetic Algorithm. journal6, 26(3): 118-120

[5] 刘艳梅．遗传算法在旋律创作中的应用研究[D]．河南师范大学,2014

[6] L. Jiang and W. Zhou, "Application of Interactive Genetic Algorithm in Fractal Art Pattern Design System," 2023 International Conference on Network, Multimedia and Information Technology (NMITCON), Bengaluru, India, 2023, pp. 1-5, doi: 10.1109/NMITCON58196.2023.10276254.

[7] D. Sun, "Interactive Genetic Algorithm in Fractal Art Teaching and Pattern Design," 2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballar, India, 2023, pp. 1-6, doi: 10.1109/ICDCECE57866.2023.10150452.

[8] K. Tang, P. Yang and X. Yao, "Negatively Correlated Search," in IEEE Journal on Selected Areas in Communications, vol. 34, no. 3, pp. 542-550, March 2016, doi: 10.1109/JSAC.2016.2525458.

[9] P. N. Suganthan et al., “Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization,” Nanyang Technological University (NTU), Singapore, Tech. Rep., May 30, 2005.

[10] Fazhan Zhang, Yichao He, Haibin Ouyang, Wenben Li, A fast and efficient discrete evolutionary algorithm for the uncapacitated facility location problem, Expert Systems with Applications, Volume 213, Part B, 2023, 118978, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2022.118978. (https://www.sciencedirect.com/science/article/pii/S0957417422019960)

[11] Y. Xu, Q. Wang and J. Hu, "An Improved Discrete Particle Swarm Optimization Based on Cooperative Swarms," 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Sydney, NSW, Australia, 2008, pp. 79-82, doi: 10.1109/WIIAT.2008.103.

[12] V. L. Huang, A. K. Qin, P. N. Suganthan and M. F. Tasgetiren, "Multi-objective optimization based on self-adaptive differential evolution algorithm," 2007 IEEE Congress on Evolutionary Computation, Singapore, 2007, pp. 3601-3608, doi: 10.1109/CEC.2007.4424939.

[13] A fast and efficient discrete evolutionary algorithm for the uncapacitated facility location problem, March 2023, [Expert Systems with Applications](https://www.researchgate.net/journal/Expert-Systems-with-Applications-0957-4174?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19) 213(6):118978, DOI:[10.1016/j.eswa.2022.118978](http://dx.doi.org/10.1016/j.eswa.2022.118978)

[14] J. Kennedy, “Particle swarm optimization,” in Encyclopedia of Machine Learning. New York, NY, USA: Springer, 2010, pp. 760–766.

[15] R. Storn and K. Price, “Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces,” J. Global Optim., vol. 11, no. 4, pp. 341–359, 1997.

[16] R. Martí, M. Laguna, and F. Glover, “Principles of scatter search,” Eur. J. Oper. Res., vol. 169, no. 2, pp. 359–372, 2006.

[17] C. García-Martínez, M. Lozano, F. Herrera, D. Molina, and A. M. Sánchez, “Global and local real-coded genetic algorithms based on parent-centric crossover operators,” Eur. J. Oper. Res., vol. 185, no. 3, pp. 1088–1113, 2008.

[18] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, “Comprehensive learning particle swarm optimizer for global optimization of multimodal functions,” IEEE Trans. Evol. Comput., vol. 10, no. 3, pp. 281–295, Jun. 2006.

[19] N. Hansen and A. Ostermeier, “Completely derandomized self-adaptation in evolution strategies,” Evol. Comput., vol. 9, no. 2, pp. 159–195, 2001.

[20] A. K. Qin, V. L. Huang, and P. N. Suganthan, “Differential evolution algorithm with strategy adaptation for global numerical optimization,” IEEE Trans. Evol. Comput., vol. 13, no. 2, pp. 398–417, Apr. 2009.

[21] A. Corana, M. Marchesi, C. Martini, and S. Ridella, “Minimizing multimodal functions of continuous variables with the ‘simulated annealing’ algorithm Corrigenda for this article is available here,” ACM Trans. Math. Softw., vol. 13, no. 3, pp. 262–280, 1987.

[22] R. Chelouah and P. Siarry, “Tabu search applied to global optimization,” Eur. J. Oper. Res., vol. 123, no. 2, pp. 256–270, 2000.

[23] H.-G. Beyer and H.-P. Schwefel, “Evolution strategies—A comprehensive introduction,” Nat. Comput., vol. 1, no. 1, pp. 3–52, 2002.