分布式进化计算综述

Wei-Neng Chen，高级会员， *IEEE*，Feng-Feng Wei、Tian-Fang Zhao、Kay Chen Tan，会士，*IEEE*，以及 Jun Zhang，会士， *IEEE*

摘要——**并行与分布式计算范式的迅速发展，为计算领域带来了巨大变革。得益于进化计算（EC）固有的并行性，在并行与分布式计算系统上实现进化计算十分自然。一方面，并行计算系统提供的计算能力能够显著提升进化计算的效率和可扩展性；另一方面，数据以分布式方式收集和处理，为进化计算带来了新的发展方向和挑战。本文旨在对分布式进化计算（DEC）进行系统综述。首先，本文从顶层设计机制到底层实现机制，提出一种新的分布式进化计算分类体系。基于该分类体系，本文从目的、算法的并行结构、实现并行模型以及实现环境等方面回顾分布式进化计算的现有研究。其次，本文阐明了分布式进化计算的两大主要目的：一是通过并行处理提升集中式优化的效率；二是让拥有部分信息的分布式个体／子种群协同进行分布式优化。再次，鉴于随着空间分布式范式的蓬勃发展，分布式进化计算的后一目的已成为进化计算中一种新兴且颇具吸引力的趋势，本文对分布式优化作出系统定义，并将其划分为维度分布式、数据分布式和目标分布式优化问题。本文给出了这些问题的形式化表述，并回顾了针对这些问题的各类分布式进化计算研究。此外，本文还探讨了相关挑战和潜在研究方向，以期启发分布式进化计算的设计并为未来发展铺平道路。**

索引词—**分布式进化计算、并行与分布式计算、分布式优化**

I. 引言

过去十年间，超级计算、云计算和边缘计算等并行与分布式计算系统迅速发展并得到广泛应用。通过汇聚海量分布式计算资源，这些计算系统为处理复杂的大规模计算应用提供了强大的计算能力。此外，在.....的助力下物联网（IoTs）技术使各类事物互联互通，因此海量数据以分布式方式收集、存储并需要处理，由此开启了大数据时代。因此，数据量不断增长以及信息处理的分布式机制，对在这些新型分布式计算系统上开发智能算法提出了迫切需求和新的挑战。

进化计算作为计算智能领域的一个活跃研究方向，在过去几十年中不断发展，并已广泛应用于解决复杂优化问题 [1]。进化计算通过模拟生物进化或社会性动物的智能行为，使解群体不断进化。它主要包括遗传算法（GA） [2]， 、遗传编程（GP）[3], 等进化算法，以及蚁群优化（ACO） [4]， 、粒子群优化（PSO）[5], 等群智能算法。进化计算具有良好的探索能力、开发能力、自学习、自组织和适应性等特点，因此能够有效解决搜索空间规则或不规则的复杂优化问题 [6], [7], [8], 。然而，迭代进化范式会降低计算效率，从而阻碍其在解决大规模或昂贵优化问题方面的应用。幸运的是，进化计算算法具有内在的并行性，这为将进化计算与新型并行与分布式计算系统相结合提供了良好契机。

随着高性能计算的快速发展，例如多处理器系统[9], 图形处理器 [10], 集群 [11], 等，大量研究工作致力于并行与分布式进化计算。它们主要旨在提高复杂集中式优化问题中进化计算的效率，在这些问题中，信息可以在一个中心进行融合[12], [13]。最广泛使用的方法是在维度级分布式并行或种群级分布式并行中实现算法 [14]。对于

稿件收到日期：XXX；修订日期：XXX；录用日期：XXX。本工作部分得到中国科学技术部国家重点研发计划项目（项目编号：2018AAA0101300）的资助；部分得到国家自然科学基金委员会项目（项目编号：61976093和61873097）的资助；部分得到韩国国家研究基金会项目（项目编号：NRF-2021H1D3A2A01082705）的资助；部分得到广东省自然科学基金研究团队项目（项目编号：2018B030312003）的资助。

Wei-Neng Chen 和 Feng-Feng Wei 任职于华南理工大学计算机科学与工程学院（中国广州 510006），同时任职于华南理工大学粤港大数据与计算智能联合创新平台（中国广州 510006）。（电子邮箱：cschenwn@scut.edu.cn，fengfeng_scut@163.com）。

Tian-Fang Zhao 任职于中国广州 510006 暨南大学新闻与传播学院（电子邮箱：tianfang09@foxmail.com）。Kay Chen Tan 任职于中国香港特别行政区香港理工大学计算学系（电子邮箱：kctan@polyu.edu.hk）。Jun Zhang 任职于韩国安山 15588 汉阳大学电气与电子工程系（电子邮箱：junzhang@ieee.org）

例如，Zhan 等。[15] 在图形处理器上实现了进化计算算法，以实现细粒度的维度级分布式并行。Tan 等。[16] 在联网计算机上设计了一种采用种群级分布式并行的协同进化算法。Jia 等。[17] 在多处理器系统上提出了一种同时采用维度级分布式并行和种群级分布式并行的双层分布式算法。它们可以在不牺牲算法性能的情况下大幅提高效率。此外，并行与分布式进化计算也已应用于工业领域，例如电磁学[18], 虚拟网络嵌入[19], 等。

随着大数据时代的到来和物联网的发展，边缘计算[20], 云计算[21], 多智能体系统（MASs）[22], 等新型分布式计算技术不断涌现，并引起了广泛的研究兴趣。因此，大量信息以分布式方式生成、收集、处理和存储，但由于计算成本高、存储空间有限以及信息隐私等原因，无法融合。因此，分布式优化问题应运而生。与集中式优化不同，用于分布式优化的进化计算中，个体无法从全局视角看待整个问题，需要相互协作。与用于集中式优化的并行与分布式进化计算相比，尽管针对分布式优化的分布式进化计算（DEC）研究较少，但如今正逐渐成为一个引人注目的研究方向。

总之，一方面，分布式计算范式的发展为计算机科学带来了突破。EC 与并行计算平台相结合，显著提升了 EC 求解大规模优化问题的效率和可扩展性。另一方面，新型分布式计算系统和多智能体系统的应用催生了许多分布式优化问题，而 DEC 已初步展现出解决这些复杂分布式优化问题的潜力。然而，目前仍缺乏对相关工作的系统综述。因此，本文对 DEC 进行了全面综述。本文的主要贡献如下。

首先，本文从两个角度回顾 DEC：a）用于集中式优化的 DEC，以及 b）用于分布式优化的 DEC。用于集中式优化的 DEC 假设算法的个体或子种群共享同一内存或数据，目标通常是利用并行与分布式计算平台提高优化效率。相反，用于分布式优化的 DEC 假设算法的个体或子种群各自拥有内存，用于保存本地数据和信息。它们无法共享内存，需要通过特定的协作策略相互配合，以实现全局优化。由于问题特征不同，这两类问题的设计也大不相同。

其次，本文提出了一种新的分类体系，用于综述分布式进化计算，其视角从顶层设计机制延伸至底层实现机制。具体而言，本文从目的与问题、算法的并行结构、实现并行模型以及实现环境等方面综述分布式进化计算算法。由于集中式优化问题和分布式优化问题分别具有集中式和分布式的特征，因此二者的目的和实现环境有所不同，而并行结构和模型则相似。并行结构是分布式进化计算算法与传统串行进化计算算法之间的核心区别，指算法中能够并行执行的模块和结构。并行模型则指算法并行实现过程中计算资源的组织与通信方式所依托的互连网络模型。

第三，本文系统地定义了分布式优化问题，并介绍了潜在的分布式进化计算的研究趋势。大多数现有的分布式进化计算算法都是针对集中式优化提出的。它们局限于具有全局目标函数和集中式决策变量的问题。然而，随着物联网的蓬勃发展，分布式优化应运而生，其中数据以分布式方式进行收集、存储和处理。对于整个问题，不存在全局视角。具体而言，我们将分布式问题分为维度分布式问题、数据分布式问题和目标分布式问题。针对每一类问题，本文都进行了详细介绍并给出了数学形式化，同时综述了分布式进化计算研究的先驱工作。

本文其余部分安排如下。第二节详细阐述本文的分类体系。第三节至第五节

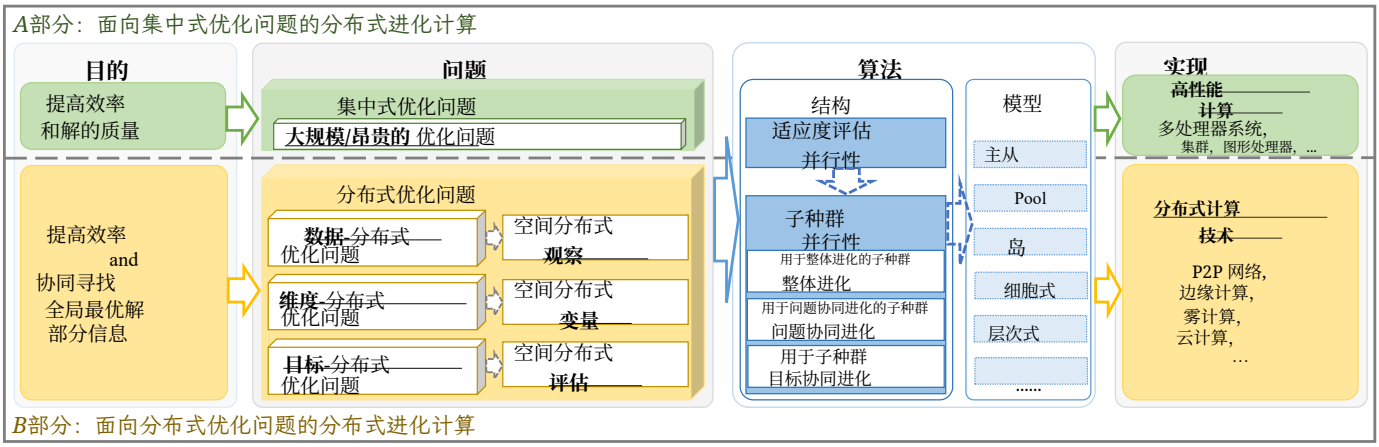


图 1。 分类体系　　本文的内容包括问题和目的、并行结构和模型，以及实现环境。

分别回顾该算法的并行结构、实现所采用的并行模型以及实现环境。第六节详细介绍新兴的分布式优化问题，并回顾针对这些问题的现有分布式进化计算。第七节讨论分布式进化计算面临的挑战和潜在研究方向。第八节总结全文。

II. 分类体系

图 1 展示了用于综述 DEC 的新分类体系。根据问题属性，我们将 DEC 求解的问题分为两类：上方绿色区域中的集中式优化问题，以及下方黄色区域中的分布式优化问题。不同类型的问题在目的、问题属性和实现环境方面各不相同。然而，DEC 算法中的并行结构类别以及用于实现的并行模型则较为相似，如图 1 中蓝色部分所示。本文根据问题和目的，从顶层算法设计到最底层实现环境进行了广泛综述，并详细介绍了用于分布式优化的 DEC。

问题与目的。一般而言，DEC 有两个不同的目的。

第一个目的是利用并行与分布式计算系统提供的计算能力，提高 EC 的计算效率。面对大规模、昂贵的优化问题时，EC 需要耗费大量时间来维持种群的迭代进化过程，因此十分耗时。在这种情况下，并行与分布式计算可以并行执行种群进化或目标函数评估，从而节省执行时间。

第二个目的是将进化计算应用于分布式系统中的分布式优化问题。传统优化问题通常是集中式的，即种群中的所有个体共享同一数据集，并专注于相同的目标，以求解全局优化问题。

然而，随着物联网和分布式计算的发展，分布式优化问题应运而生。与集中式优化问题不同，在分布式优化问题中，数据由不同的个体或智能体收集，并且由于传输效率或隐私方面的原因，数据无法汇总。因此，个体必须协作以实现全局优化。根据问题定义中哪一部分以分布式方式给出，分布式优化问题可分为三类：维度分布式优化问题、数据分布式优化问题和目标分布式优化问题。顾名思义，它们分别具有空间分布式的变量、观测和评估。

算法的并行结构。并行结构是指算法中可以同时执行的模块或结构。对于差分进化，通常可将并行结构分为两类：适应度评估并行和子种群并行。

适应度评估并行是最简单的并行方式，即并行执行每个个体的适应度评估。子种群并行是指并行执行每个子种群乃至每个个体的进化算子。根据子种群的功能，我们还可以将子种群并行进一步分为整体进化的子种群并行、子问题协同进化的子种群并行，以及目标协同进化的子种群并行。顾名思义，整体进化的子种群并行是指多个子种群并行探索整个搜索空间。子问题协同进化的子种群并行是指将原始问题分解为若干包含部分决策变量的子问题，并由多个子种群协同优化这些子问题，以找到全局最优解。目标协同进化的子种群并行专为多目标问题设计。不同子种群分别专注于

表I 分布式进化计算研究调查

目的	问题	数量 种群	并行性 结构	算法 模型	实现 环境	代表性论文
提高效率 和解的质量	集中式 优化 问题	单一	适应度 评估 并行性	岛模型	GPU	[169], 等。
		多个	子种群 用于.....的并行性 整体性 演化			[109], [129], [170], etc.
				细胞式模型	多处理器 系统	[79], [82], [83], 等。
				主从 模型	集群	[106], 等。
			子种群 并行 协作 协同进化	岛屿模型		[105], [140], 等。
协同寻找 全局最优解 部分信息	分布式 优化 问题	单一	子种群 并行 目标 协同进化	主从 模型	多智能体 系统	[153], 等。
		多种方式	子种群 并行性 整体 进化		云 计算	[131], [143], 等等。
				细胞模型	多智能体 系统	[90], [91], [92], 等等。
				岛屿模型	P2P 网络	[139], 等。

不同的目标，并且需要相互协作以找到帕累托前沿。

实现并行模型。并行模型是指关于计算资源如何组织以及如何进行通信的互连网络模型，用于并行算法实现。通常，实现并行模型可分为两类：粗粒度模型和细粒度模型。粗粒度模型在每个处理器中具有用于计算的较大子组件，例如子种群；而细粒度模型具有较小的子组件，例如个体，甚至是维度。这些处理器连接在特定的网络中。网络的特征，例如全连接或部分连接、循环或无环、有向或无向等，决定了处理器的组织方式和通信方式。因此，广泛使用的并行模型有四种类型：主从模型、岛屿模型、细胞模型和层次模型。主从模型和细胞模型是细粒度模型，而岛屿模型是细粒度模型。层次模型是上述多个模型的组合，因此可以是粗粒度模型和细粒度模型的组合形式。

实现环境。实现环境是开发分布式进化计算算法的基础。文献中，并行与分布式进化计算已在高性能计算环境（如 GPU 计算系统 [10], 多处理器系统 [9], 集群 [11], 等）以及空间分布式计算系统（如 P2P 网络 [23], 边缘计算 [20], 雾计算 [24], 云计算 [21], 等）上实现。

在高性能计算环境中，内存和数据是共享的，所有处理器都专注于同一个目标，以解决全局优化问题。使用高性能计算环境来提高效率是很常见的。在空间分布式计算环境中，数据由分布式处理器收集、存储和处理，而由于传输效率或隐私方面的原因，数据难以汇总。处理器需要协同搜索全局最优解。设计有效且高效的协同进化通信策略至关重要。

表I全面考察了分布式进化计算从目的到实现环境的各个方面，有助于理解新的分类体系并推动分布式进化计算的发展。

III. 算法的并行结构

对于进化计算，目标函数的评估以及各子种群（或个体）的进化均可并行执行。不同的并行模块会形成不同的并行结构。因此，我们将 DEC 的并行结构分为适应度评估并行和子种群并行两类，并对此进行综述。

A. 适应度评估并行

适应度评估并行是通过多个处理器评估适应度。有两种实现适应度评估并行的方法。第一种是纯适应度评估并行，其中演化在中心进行，而评估则通过多个处理器进行。

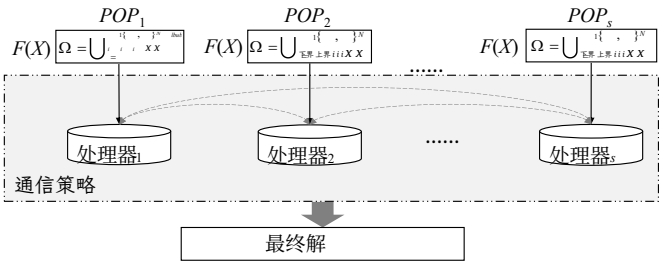


图 2。整体进化的子种群并行示意图。

另一种是适应度评估并行与子种群并行相协作。这是一种常见的并行结构，因为使用子种群进行评估更加高效。

在适应度评估并行中，一个主要问题是适应度评估预算的分配。适应度评估预算分配旨在确定每个处理器分配到多少次适应度评估。除了各处理器拥有相同适应度评估资源预算的平均分配策略外，研究者还提出了一些自适应适应度评估分配策略。Ren 等 [25] 提出了一种计算资源分配策略，根据各子种群对适应度提升的贡献，合理地为其分配适应度评估次数。Jia 等 [17] 设计了一种周期性贡献计算方法，并

根据计算出的贡献分配计算资源。Li 等人 [26] 提出了一种分布式子种群间的自适应资源分配方法。该方法旨在为表现良好的部分自适应地分配更多适应度评估。文中提供了理论分析，以展示所提方法的有效性。

B. 子种群并行

1) 整体进化的子种群并行

将原始最小化问题表示如下：

$$\text{最小值}(\ ),\ ,\ F\ X\ X\ \in\Omega$$

$$\Omega=\bigcup_{i=1}^{N}\{x_i^{lb},x_i^{ub}\} \tag{1}$$

其中 F 是待优化的目标函数。 X 是大规模搜索空间中的 D 维决策变量向量 Ω，其受上界 和下界约束

^{ub}

$_{b}\,{}_{i}x^{l}$ 的每个维度。整体进化的子种群并行是指将整个种群划分为多个子种群，并且每个子种群都在整个搜索空间中进行进化。如图 2所示，共有 s 个子种群，并且每个子种群都在整个搜 索空间 Ω 中优化问题 $F(X)$。具体而言，当每个种群包含一个个体时，这就是个体并行。整体进化的子种群并行的主要研究方向是迁移策略。

迁移策略规定何时交换和更新子种群，以及交换和更新哪些内容。由于协作发挥着重要作用，迁移策略会极大地影响性能。通常，迁移分为同步方式和异步方式。同步迁移是指等待所有子种群接收信息，而异步迁移是指信息一到达某个子种群便立即接收。

> 请将此行替换为您的稿件编号（双击此处编辑）<

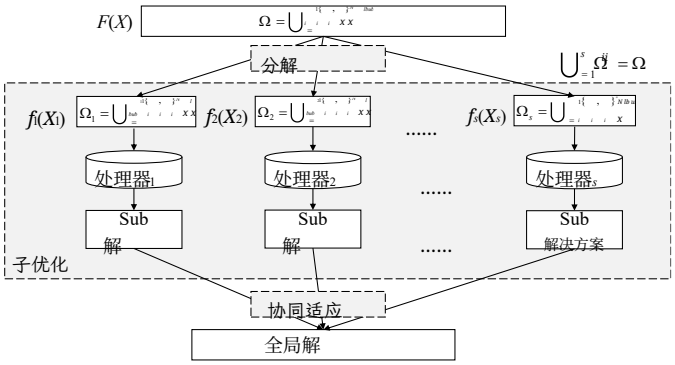


图 3。 子问题协同进化中子种群并行的示意图。

同步迁移是一种广泛使用的方式，因为它有效且易于实现。Alba 等。 [12]，[27] 以并行方式实现了进化算法，其中迁移同步进行。De Falco 等。 [28] 提出了一种基于入侵的分布式差分进化迁移模型，其中子种群以同步方式依次发送和接收个体。

然而，等待每个子种群中的信息到达是浪费时间的。因此，异步迁移应运而生，并在子种群并行性中日益普及。Apolloni *et al*. [29] 提出了一种异步迁移方法，其中待迁移的个体由选择函数随机选出。De Falco *et al*. [27] 提出了一种自适应异步迁移机制。他们采用基于概率的方法来确定是否触发迁移。一旦发生迁移，源子种群中的最佳个体就会替换目标子种群中的最差个体。此外，De Falco *et al*. [30] 利用了分布式差分进化中异步迁移的多样性。Jakobović *et al*. [31] 设计了一种异步消除式全局并行进化算法，并证明其效率高于其他并行算法。Izzo *et al*. 提出了一种异步岛模型，并在 POSIX 线程应用程序编程接口上进行了测试。Zhan *et al*. [32] 开发了一种自适应迁移策略，可以在各子种群之间提供稳健的解决方案和更好的性能。Fernández *et al*. [33] 基于分布式遗传编程分析了同步迁移和异步迁移的优缺点。他们发现，异步方式更适合多处理器系统。Alba 和 Troya [34] 研究了子种群之间的同步迁移和异步迁移。他们发现，就时间而言，异步迁移优于同步迁移。

2) 子问题协同进化的子种群并行性

子种群并行用于解决大规模问题，这类问题由于维度灾难而极难找到全局最优解。为降低优化难度，原始问题被划分为若干更易优化的子问题。分而治之是将大规模问题拆分为较小规模问题的一种常用技术 [35]。分而治之策略与进化计算算法相结合，催生了协作协同进化（CC），它是

算法 1 *DG* 的伪代码

输入：未分组变量集 $\Omega = \bigcup_{\text{下界}}^{\text{上界}}\{x_i^{1}, x_i^{2}, \ldots, x_i^{N}\}$

重复

$\Omega_s = x_{1st}$ 在 Ω, $\Omega = \Omega - x_{1st}$;

如果 Ω 不为空

for $j=2$: $|\Omega|$

如果 交互(x_{1st}, x_j) 为 **真**

$\Omega_s = \Omega_s \cup x_j$, $\Omega = \Omega - x_j$;

结束如果;

循环结束;

结束如果;

直到Ω 不为空

输出：分组变量集 $\Omega_1 = \bigcup_{i=1}^{N_1}\{x_i^{1}, x_i^{2}, \ldots, x_i^{N_1}\}$, $\Omega_2 = \bigcup_{i=1}^{N_2}\{x_i^{lb}, x_i^{ub}\}, \ldots, \Omega_s = \bigcup_{i=1}^{N_s}\{x_i^{lb}, x_i^{ub}\}$

子问题协同进化中最广泛使用的方法。如图 3 所示，有 s 个子种群，每个子种群负责在具有多个维度的子搜索空间 Ω_i 中优化一个子问题 $f_i(X_i)$。所有子搜索空间的并集覆盖整个搜索空间。子问题协同进化的子种群并行包括分解、子优化和协同适应，其中分解和协同适应是两个重要步骤。

1) 分解

分解是将大规模决策变量 X 划分为 s 个较小规模的决策变量 $X_1, X_2, \ldots, X_s$。相应地，原始搜索空间 Ω 被分解为子搜索空间 $\Omega_1, \Omega_2, \ldots, \Omega_s$。所有子搜索空间的并集 即为整个搜索空间 Ω。每个较小规模的决策变量都被视为一个子问题 $f_i(X_i)$。如果所有变量彼此独立，分解策略对优化结果的影响很小。这类问题称为 可分离问题，较易优化。如果变量之间存在某种关系，分解策略就会对优化结果产生很大影响。这类问题称为 不可分离问题，在实际应用中很常见。已有研究表明，近似最优分解能够量化子问题对全局适应度的贡献 [36]。

通常，分解策略可分为静态分解策略和动态分解策略。静态分解策略在.....之前执行。

进化过程中，分解保持不变。典型的静态分解策略包括随机分组 [37], 快速互依关系识别（FII） [38], 差分分组（DG） [39] 等。随机分组是固定组数和组大小 [37]，然后将变量随机分入各组。随机分组具有较低的时间复杂度，并且在可分优化问题中表现良好。然而，它无法很好地处理不可分优化问题。FII 将检测分为两个阶段。首先，识别可分变量和不可分变量。然后，进一步检测不可分变量之间的关系。DG 由 Omidvar 等人提出。[39] 是最广泛使用的不可分

优化问题分组方法之一。DG 的伪代码如算法 1 所示。首先，将未分组变量集合中的第一个变量视为一个新组，并将其从集合中移除。然后，检验该变量与未分组变量集合中所有其他变量之间的交互关系。如果检验结果表明存在交互，则将检测到的变量与第一个变量分为一组，并将其从剩余的未分组变量中移除。如果没有变量与第一个变量存在交互，则将第一个变量自身作为一个组。重复此过程，直到未分组变量集合为空。在 DG 提出后，研究人员开发了许多变体方法，例如 XDG [40], GDG [41], DG2 [42], RDG [43], RDG2 [44], DDG [45], MDG [46], 等。

动态分解策略可以在优化过程中动态地进行分解。动态分解策略包括频繁随机分组 [47], δ分组 [48], 动态分组（DyG）[49], 基于知识的动态变量分解（KDVD） [50], 等。Omidvar 等。 [47] 提出了一种更频繁的随机分组方法，以提高随机分组的频率，同时不增加适应度评估的预算。此外，他们考虑打破交互变量改进间隔的限制，并提出了δ分组 [48]。Zhang 等。 [49] 提出了一种动态分组方法（DyG），以自适应地将计算资源分配给精英组。Cai 等。 [50] 专注于双层多目标优化问题，并设计了一种基于知识的动态变量分解彩在线

g基于 e 的动态变量分解（KDVD）。

2) 协同适应

子问题之间的协同适应是指在适应度评估过程中共享进化信息。如何从其他子问题中选择具有代表性的信息，以形成用于评估的完整个体，至关重要。一般而言，协同适应可分为最佳协作者协同适应、最差协作者协同适应、随机协作者协同适应、精英协作者协同适应以及

基于策略的协作者协同适应。令j表示第i个子问题中的第 个体，其评估方式如下。

$$F(X)\mid X=\{\vec{x}_{1,*},\vec{x}_{2,*},...,\vec{x}_{i,j},...,\vec{x}_{s,*}\} \tag{2}$$

$\vec{x}_{o,*}$

$\vec{x}_{o,*}$

其中，（o=1、2、...、s 和$o\neq i$）是来自其他子问题的协同适应信息。最佳协作者协同适应是指采用其他子问题中的最佳个体。这是子问题并行中应用最广泛的策略之一[37], [39], [51]。具体而言，最佳协作者协同适应非常适用于可分问题，而对于不可分问题则容易导致局部最优[52]。相反，最差协作者协同适应是指采用其他子问题中的最差个体。尽管最差协作者协同适应能够提供更多备选候选项，但它并非最优策略，在协同协进化中也不常用 [53]。随机协作者协同适应是指从其他子问题中随机选取一个个体 [53]。它可以显著提升种群多样性并防止早熟收敛[16]。然而，过高的随机性可能会降低收敛速度[54]。精英协作者协同适应是指从精英池中随机选取个体，该精英池由每个子问题中最优的K个个体组成[55]。 $\vec{x}_{o,*}$

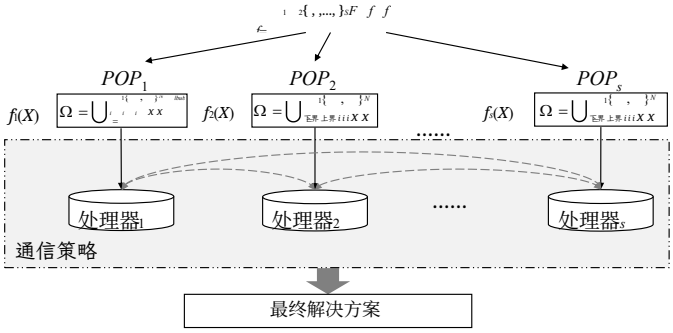


图4。目标协同进化的子种群并行示意图。

具体而言，如果K为 1，精英协作者协同适应与最佳协作者协同适应相同。由于精英池 [16]具有多样性，精英协作者协同适应可以防止早熟收敛。基于策略的协作者协同适应是指从每个子问题中通过某种特定策略进行选择，例如轮盘赌选择 [56] 或锦标赛选择 [57]。这些策略为候选个体提供了更多选择，不同策略各有优势。*3*） 目标协同进化的子种群并行

目标协同进化的子种群并行是指使用不同的子种群来进化不同的目标。如图4所示，该优化问题有s个目标，并且每个目标都由一个子种群在所有维度上进行优化$f_i(X)$。

目标协同进化的子种群并行

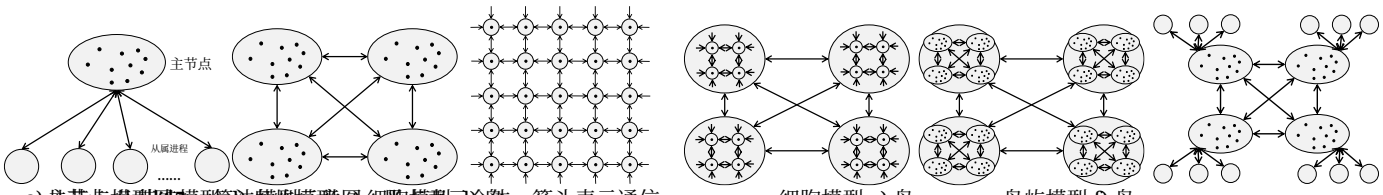
与正式的两个子种群并行相比，协同进化受到的关注较少，原因有二。首先，DEC 通常研究适应度可以同时获取的问题。换言之，所有智能体都能够获取所有目标值。其次，子种群并行要实现协同进化，必须存在多个目标，这增加了研究难度。已有一些针对异质目标评估优化的研究，其中不同目标或约束的评估在计算时间上存在差异 [58]。Allmendinger *et al*。 [59] 将种群划分为快速种群和慢速种群，以分别获取延迟较短和延迟较长的目标。Lewis *et al*。 [60] 研究了使用多目标 PSO 进行异质优化的主从模型。他们将所提出的方法扩展到了分布式环境。Scott 和 De Jong [61] 提出了一种基于主从模型的异步进化算法，以缓解评估时间偏差。Yagoubi *et al*。[62] 开发了一种具有异质评估成本的异步进化多目标算法。

IV. 实现并行模型

在文献中，差分进化常用四种并行模型，包括图5所示的主从模型、岛屿模型、细胞模型和层次模型。较大的圆形或椭圆形表示用于执行计算的分布式组件。黑点表示个体，箭头表示分布式组件之间的通信。

A. 主从模型

主从模型是一种细粒度模型。如图5 a)所示，主从模型由一个主节点



a) 从节点模型 b) 主从模型 c) 细胞模型 d) 个体，箭头表示通信。 -细胞模型 e) 岛 -岛屿模型 f) 岛

处理器和多个从节点处理器组成。主从模型最初是为适应度评估并行而提出的，在这种模式下，主节点负责进化，并将个体发送给从节点，以执行优化过程中最耗时的部分——适应度评估 [63]。主从模型中的通信仅发生在从节点处理器与主节点处理器之间。从节点处理器之间没有直接通信。*1*) *C*

通信 *S* 策略

随着主从模型变体的发展，从节点也负责部分进化任务 [64], [65]。为了提高主从模型的效率，采用了异步策略。Yang 等。 [66] 提出了一种基于请求-响应机制的异步自适应通信方法。在进化过程中，主节点处理器与从节点处理器之间的通信会被自适应地触发。Said 和 Naka mura [67] 提出了一种异步通信策略：一旦满足终止条件，从节点处理器即可开始新一轮迭代。如果从节点处理器在主节点处理器完成下一阶段所需的评估之前就已完成当前阶段，则需要根据已完成阶段的局部最优解重新初始化。

2) 改进的主从模型

此外，人们还致力于提高主从式分布式进化算法的有效性。Ismail [68] 将主从模型应用于遗传算法，并对模型进行了改进以提高效率。在改进后的模型中，从属处理器不仅负责适应度评估过程，还负责变异过程和部分交叉过程。主处理器仅负责种群的选择和分配。Yu 和 Zhang [69] 提出了一种基于主从模型的分层遗传算法。全局搜索由主处理器执行，而局部搜索则由从属处理器执行。主层中的染色体表示从属层的搜索中心，并根据从属层的搜索结果进行演化。Lorion 等。[70] 提出了一种由协调智能体和若干群体智能体组成的基于智能体的并行粒子群优化。群体智能体处于计算状态或交易状态。借助小生境方法，每个群体智能体都会出售远离中心点的粒子，并购买靠近中心点的粒子。

3) 池模型

池模型是一种相当特殊的主从模型，在该模型中，从节点而非主节点负责进化。主节点充当池，用于保存从节点的进化信息。从节点可以通过从池中读取全局信息并将局部信息存入池中，以异步方式进行进化。Roy 等. [71] 提出了一种特殊的主从模型，称为池模型，并将其应用于遗传算法。在该模型中，主处理器仅负责将个体存储在池中。从处理器从池中提取个体进行计算和进化。之后，更新后的个体会被重新写入池中。Yang 等. [66] 提出了一种基于池模型的分布式粒子群优化。主处理器维护全局最优个体，从处理器负责进化。该方法设计了一种精英引导学习策略来更新粒子。当前粒子群中的精英粒子以及不同从处理器找到的历史最优解，被用于引导各从处理器中粒子的更新。

B. 岛屿模型

岛屿模型的结构如图5 b)所示。作为一种粗粒度并行模型，岛屿模型将整个种群划分为若干子种群，并将每个子种群分配到一个计算节点。因此，岛屿模型通常用于子种群并行。

1) 通信策略

为提高算法的效率，提出了异步通信策略，以避免不必要的等待。Du 等。[72] 提出了一种用于基因表达式编程的异步分布式 EDA。在该算法中，每个处理器独立进行进化，并与其他处理器进行通信

经过若干代后，每个处理器中一定数量的较差个体会被从其他处理器接收的较优个体替换。Zhan 和 Zhang [73] 为粒子群优化提出了一种自适应异步迁移策略。群体迁移根据搜索环境和当前进化状态独立执行。

2） 改进的岛屿模型

岛屿模型最早由Tanese等。[74] 在分布式遗传算法（DGA）中引入。实验结果验证了所提出的DGA在随机Walsh多项式优化方面优于经典串行遗传算法。随后，Belding [75] 进一步验证了 DGA 在 Royal Road类问题上的有效性。之后，Romero和 Cotta [76] 将岛屿模型应用于粒子群优化（PSO），并测试了不同迁移参数的影响。迁移粒子的选择包括随机选择和选择最佳粒子。替换策略包括始终替换，以及当解更优时进行替换。岛屿拓扑包括环形拓扑和完全拓扑。Apolloni等。[29] 将单向岛屿模型应用于差分进化，其中随机选择迁移粒子，并在其更优时进行替换。Ishi mizu和Tagawa [77] 进一步探究了具有不同网络拓扑的岛屿差分进化的性能，这些网络拓扑包括环形、环面、超立方体和层次式拓扑。Zhang 等。[78] 提出了一种结合拉马克式学习和鲍德温式学习的分布式差分进化算法，并采用 Hooke–Jeeves 局部搜索策略。在该模型中，子种群通过 von Neumann 拓扑连接，这与细胞模型的拓扑结构相同。

C. 细胞模型

细胞模型是一种细粒度模型，它将种群中的每个个体分配到一个计算节点。如图5c)所示，每个个体都被视为生活在格点中的一个细胞，并且只能与其邻居进行通信。由于易于在并行硬件上实现，细胞模型很受欢迎[12], [79]。由于相邻格点存在重叠，性能良好的个体可以将其优势传播到整个种群。与粗粒度岛模型相比，当种群规模较大时，细胞模型需要更多的处理器。

1) 通信策略

在细胞进化算法的早期研究中，各个细胞彼此通信并同步更新。这种机制能够保证算法稳定性，但有时会导致效率低下。为解决这一问题，研究者提出了一种异步细胞进化计算算法，其中包含四种异步更新机制：固定方向扫描、固定随机扫描、随机新扫描和均匀选择 [80]。在固定方向扫描中，所有细胞按照预先设定且固定的顺序更新；在固定随机扫描中，所有细胞按照随机但固定的顺序更新；在随机新扫描中，所有细胞按照随机且不固定的顺序更新。在均匀选择中，每次随机选择一个细胞进行更新。Akat 和 Gazi [81] 提出了一种异步细胞粒子群优化，其中细胞可以彼此通信，并在不同时间点更新。更新粒子时会使用过时信息。每个粒子都可以直接与其邻居通信，而邻居会随时间动态变化。当邻居粒子的迭代尚未完成时，当前最优解会被用作全局最优解的候选解。

2） 改进的细胞模型

为提高进化计算算法的有效性，人们对最初的细胞模型进行了改进。Alba 和 Troya [82] 定义了一个比率，用于衡量细胞进化算法中拓扑与邻域之间的关系，并提出了一种能够自适应调整该比率的细胞进化算法。Nebro 等。[83] 提出了一种名为 MOcell 的新型细胞遗传算法，用于处理多目标优化问题。Giacobini 等。 [84] 关注了不同类型细胞进化算法中的选择强度，包括异步细胞进化算法、线性细胞进化算法 [85], 和环面细胞进化算法 [86]。此后，他们总结了有关不同类型细胞进化算法选择强度的研究，并采用基于同步和异步细胞更新策略的概率差分方程推导出选择模型[87]。

作为一种细粒度模型，细胞模型需要大量计算节点来完成种群的分布式部署。Nakashima 等。 [88] 提出了一种粗粒度细胞模型，称为组合细胞遗传算法，以解决这一问题。在组合细胞遗传算法中，若干个体被视为生活在同一格点中的细胞。只有位于格点边缘的细胞才能与邻近格点边缘的细胞通信。[89]。

3) 多智能体模型

多智能体模型是细胞模型的一种变体[90], , 其结构与细胞模型相同，即将个体分布到格点上。每个格点都被视为一个智能体并与邻居协作或竞争。与传统细胞模型不同，多智能体模型中的智能体拥有各自的行为。多智能体模型的主要行为策略包括邻域竞争或协作算子、进化算子（由具体的进化算法决定）以及自学习算子。Zhong 等. [90] 将多智能体系统与遗传算法相结合，提出了一种多智能体遗传算法（MAGA）。在 MAGA 之后，Zhao 等. [91] 提出了一种基于多智能体系统的新型粒子群优化算法（MAPSO），用于解决无功功率调度问题。随后，研究者提出了用于解决约束满足问题的 MAEA-CSPs [92], , 以及用于解决组合优化问题的 MAEA-CmOPs[93]。在通信策略方面，Laredo 等. [94] 为多智能体模型提出了一种异步通信方法，该方法能够根据网络状况自适应地设置迁移率。Ding 等. [95] 设计了一种基于多智能体系统的协同进化量子粒子群优化算法，旨在对大数据集进行属性约简时探索搜索空间和全局最优区域。Zhang 等. [96] 提出了一种用于设备布局多目标优化的进化多智能体框架。Ji 等. [97] 提出了一种多智能体进化模型，用于发现复杂网络中的社群。

D. 层次模型

层次模型是由两个或多个模型组合而成的，旨在兼顾适应度评估并行和子种群并行的优势。例如，图5 d)、e)、f) 分别展示了岛-细胞式层次模型、岛-岛层次模型和岛-主从层次模型。Folino 等人将岛屿模型与细胞模型相结合，研究了用于 P2P 计算 [98] 和模式分类 [99]的分布式 GP 算法。Herrera 等人 [100] 开发了一种采用岛-岛层次模型的分布式遗传算法。Burczynski 等人 [101], [102] 和 Lim 等人 [103] 提出将岛屿模型与主从模型相结合。他们将种群分解为若干子种群，由主节点发送给从节点。上层采用岛屿模型，以便进一步协作。

V. 实现环境

实现环境是开发分布式进化计算算法的基础。在相关文献中，并行与分布式进化计算算法既在高性能计算环境中实现，例如图形处理器（GPU）计算系统、多处理器系统、集群等；也在空间分布式计算系统中实现，例如 P2P 网络、云计算等。

A. 高性能计算

1) 多处理器系统和集群

多处理器系统是指在一个系统中包含一个以上中央处理器（中央处理器）的集合，[9] 而集群则是在两个或更多个独立系统中汇集一个以上中央处理器。它们是分布式进化计算算法提高效率的常见实现环境。Vavpotic 等人 [104] 介绍了一种多处理器方法，利用多处理器系统加快进化计算算法的执行速度。他们将种群划分为若干子种群，并让每个

子种群在一个处理器上进化。该方法已应用于 Tomita 自动机 3，且加速比与处理器数量成正比。Yang 等人 [66] 在多处理器系统上开发了分布式群优化器。在维度为 1,000 至 3,000 的优化问题中，随着处理器数量增加，该方法实现了近乎线性的加速。Zhan 等人 [15] 在中央处理器上按维度实现了基于矩阵的遗传算法和粒子群优化。对于维度为 1,000 的问题，基于矩阵的遗传算法和粒子群优化在使用 64 个核心时可实现 50 倍加速。Bazlur Rashid 等人 [105] 对 MapReduce 集群上的分布式进化计算进行了全面综述，探讨如何以更少的计算时间解决大规模优化问题。Du 等人 [106] 提出了一种基于 MapReduce 集群的并行基因表达式编程方法，用于解决符号回归问题。他们并行执行适应度评估、交叉、变异和选择等操作，在使用 64 个处理器时可实现 15 倍加速 .

2） 图形处理器

图形处理器可以并行执行大量线程。凭借强大的计算能力，图形处理器已被广泛用于处理计算负担繁重的复杂任务[10]。

Haghofer 等. [107] 在图形处理器上将进化策略实现为岛屿模型，用于显微图像中的细胞检测。实验中，优化参数设置为 200 至 5,000，最大像素数为 6,000；测试质量越高，图形处理器执行时间越长。Laguna-SaNchez 等. [108] 提出了基于英伟达图形处理器的差分进化并行实现。他们将每个个体分配到一个线程中执行，并针对 30 维问题以及不同个体数量开展实验。结果表明，当个体数量增加至 1,024、迭代次数达到 15,000 时，加速效果显著提升。Huang 等. [109] 聚焦于大规模进化多任务，并提出了一种基于 GPU 的范式，可同时优化数十万个优化请求。具体而言，在 1,000 个任务上，他们实现了最高 1,426 倍的加速。

B. 分布式计算

1) P2P 网络

P2P 网络是一种分布式基础设施，允许多个对等节点相互连接和通信 [23]。凭借较强的适应性、可靠性和性能，P2P 网络已应用于分布式优化。Wickramasinghe 等人 [110] 为 P2P 网络中的进化算法设计了一种自适应的自主选择机制。在 N 皇后问题上的实验结果表明，当 N 的最大值为 96 时，该方法具有可行性和可扩展性。Laredo 等人 [111][112] 提出了一种空间结构化进化算法，它由一群通过 P2P 协议连接的可进化智能体（EvAg）构成。他们发现，由于采用了分布式结构，EvAg 的可扩展性优于顺序遗传算法、代际遗传算法和稳态遗传算法。此外，EvAg 所需的计算资源较少，在求解含有 2 至 4 个陷阱的问题时执行效率更高。此外，Laredo 等人 [113] 研究了种群规模对 P2P 进化算法执行时间的影响。他们在 PeerSim 模拟器（一个 P2P 模拟器）中开展了实验，并发现 P2P 进化算法具有更好的可扩展性和更高的效率。他们得出了两项结论：a）种群规模相对于问题维度呈多项式阶增长，b）执行时间相对于种群规模呈分数幂增长。

2) 云计算

云计算是一种新型的分布式计算模式，计算任务在数据中心执行，并通过云 [21] 进行处理。云计算具有更高的敏捷性、更快的价值实现速度、更强的可扩展性和更高的成本效益，因此吸引了越来越多的研究关注。Malmir 等人 [114] 利用粒子群优化训练多层感知器网络，以在云计算中进行数据挖掘。他们发现，对于少量数据，并行处理并不能加快处理速度；但面对大量数据时效果良好，且能在更短时间内完成数据分类。Yu 等人 [115] 提出了一种改进的免疫进化算法，用于云计算环境下的医学数据聚类和特征提取。在 UCI 机器学习知识库中的威斯康星州乳腺癌（原始）数据集上进行的实验表明，该方法能够提取对临床诊断具有重要意义的属性，并提高诊断速度和准确率。此外，在需要聚类的样本超过 10,000,000 个时，该方法仍能达到 85% 的准确率。

C. 进化计算在计算模式方面的应用

为进化计算提供实现环境是计算模式与进化计算的结合。反过来，进化计算也用于优化计算模式中的一些问题，例如工作流调度优化、资源分配、任务卸载、能耗优化等。

对于高性能计算，Pillai 等人. [116] 使用基于遗传算法的方法优化多处理器系统的调度，以处理能耗问题。为解决处理器间的负载均衡问题，Bi 等人. [117] 提出了一种用于多处理器选择的分布式进化博弈算法。Handzel 等人. [118], [119] 使用进化计算技术优化异质多处理器系统中的任务调度与分配。Kim 等人. [120] 设计了一种用于多处理器任务分配的遗传算法。此外，Zhou 等人. [121] 对用于多处理器调度与分配问题的进化计算算法进行了理论研究。

对于空间分布式计算，Anandaraj 等。[122] 提出了一种高效且新颖的遗传算法，用于解决P2P网络中的网络编码最小化问题。Zuo 和Zhang [123] 采用进化博弈模型，提高自利路由的效率，并克服由自私路由导致的混淆状态。Dai 等。[124] 开发了一种基于进化博弈的分布式算法，以提高移动云计算的使用率。Ye 等。[125] 提出了一种改进的由膝点驱动的进化算法，用于优化云计算中的工作流调度。Zhu 等。[126] 开发了一种新颖的进化多目标优化方案，用于云计算中的工作流调度。Teijeiro 等。[127] 在Microsoft Azure 公有云平台上开发了差分进化的基于岛的并行版本。

D. 讨论

在高性能计算环境和空间分布式计算环境中实施分布式进化计算，都可以大幅提高分布式进化计算算法解决及时成本问题的效率。然而，它们之间也存在许多差异。

在高性能计算环境中，处理器共享同一全局内存，并且能够频繁相互通信以交换信息 [128]。分布式进化计算主要旨在通过海量计算资源加速并行执行。由于所有处理器共享全局信息，处理器能够更轻松地在较短时间内实现一致性和收敛。它擅长处理集中式优化问题 [14]。然而，高性能计算环境通常规模庞大且成本高昂。处理器被部署在同一区域，并通过局域网连接。开发、维护和扩展的成本都很高。求解问题所需的执行时间受计算资源规模的限制。

在高性能计算环境中设计分布式进化计算算法时，研究人员更加关注哪些模块以及如何并行执行。最常用的评估指标是加速比 R_s [13][63][129]，其计算如下：

$$R_s = \frac{T_1}{T_p} \tag{3}$$

其中，T_1 表示由 1 个处理器顺序执行的时间，T_p 表示由p个处理器并行执行的时间。它反映了随着p增加，并行执行比顺序执行快多少倍。理想情况下，当R_s=p时（其中T_1= T_p×p），这表明算法充分利用了计算资源，达到了线性加速比。在这种情况下，随着p增加，T_p 可相应缩短T_1 倍。然而，在大多数情况下，实现线性加速比并不容易。为衡量计算资源利用情况，引入效率E，其由R_s 推导得出，计算公式为E=R_s/p。算法的E较高，说明其计算资源利用率较高。显然，当算法达到线性加速比时，E=100%。随着问题规模增大、计算资源扩充，算法的可扩展性也是一项重要指标。如果计算资源和问题规模按相同比例扩展，且E没有下降，则可认为该算法在大规模问题和大规模并行计算平台上具有良好的可扩展性。

在空间分布式计算环境中，处理器只有其本地内存，并且彼此通信的成本很高 [130][131]。因此，除了通过大量计算资源提高效率之外，分布式进化计算的一个重要目的，是协作分布式处理器来搜索全局最优解。在分布式优化问题中，在空间分布式计算系统上实现分布式进化计算是合适的。在空间分布式计算环境中，处理器位于不同区域，并通过互联网连接。用户通常只需租用处理器而无需进行维护，因此费用更低且可扩展性更高。然而，由于各处理器拥有不向其他处理器公开的局部信息，因此研究在有限通信条件下所有处理器之间的一致性和收敛性非常重要 [132]。

设计空间分布式计算环境中的分布式进化计算算法时，除了提高效率外，研究人员还关注分布式通信[132][133]。由于通信成本高，研究人员需要考虑通信的内容、时机和方式。通信量和频率受到传输带宽的限制。频繁通信或大批量传输可能会降低效率；相反，个体或子种群之间过少的通信可能会严重影响全局优化中的协作。此外，还应考虑数据保护，这在如今正成为日益增长的需求 [134]。

VI. 面向分布式优化问题的分布式进化计算

上述讨论的分布式进化计算技术主要用于求解具有全局目标函数和集中式决策变量的集中式优化问题。提出这些技术的目的是通过多个处理器的协作来提高执行效率和解的质量。然而，随着互联网和大数据的蓬勃发展，大量分布式优化问题不断涌现，在这些问题中，无法从全局视角看待整个问题。分布式进化计算用于分布式优化的主要目的，是协作搜索全局最优解，这高度依赖于智能体与多个局部决策者之间的信息交换。典型应用包括云数据的分布式存储、具有局部感知能力的物联网、边缘计算等。这些新问题和应用给现有的分布式进化计算技术带来了挑战。因此，本节首先介绍分布式优化问题，然后详细阐述这类问题的定义和分类体系。重要的是，本文回顾了一些分布式进化计算技术的先导研究，为分布式优化问题的研究铺平道路。

A. 分布式优化问题

1) 引言

分布式优化问题是智能优化中新兴的一类问题。与以集中方式处理的集中式优化问题不同，分布式优化问题具有天然的分布式特征。首先，数据以分布式方式收集，将所有数据整合到融合中心的成本高昂，甚至是不可能的。其次，计算以空间分布式方式进行。分布式优化问题是指大量智能体a位于不同地点，并通过特定拓扑相互连接。它们生成或收集数据，并独立处理数据。其中很少有智能体，甚至没有智能体掌握整个问题的全局信息。每个智能体仅拥有不完整的局部信息。智能体的目的是基于自身信息和接收到的信息对问题进行优化。不同智能体的数据收集和计算彼此异步且独立。只有相互连接的智能体才能进行通信。由于通信成本高昂且需要保护隐私，通信无法频繁进行。高度的空间分布性和有限的局部信息，使模拟退火 [135], 禁忌搜索[136], 人工神经网络（ANN）[137], 等传统优化方法难以找到全局最优解。因此，分布式优化问题给现有的大多数优化方法带来了空间局部信息和异步处理方面的巨大挑战。

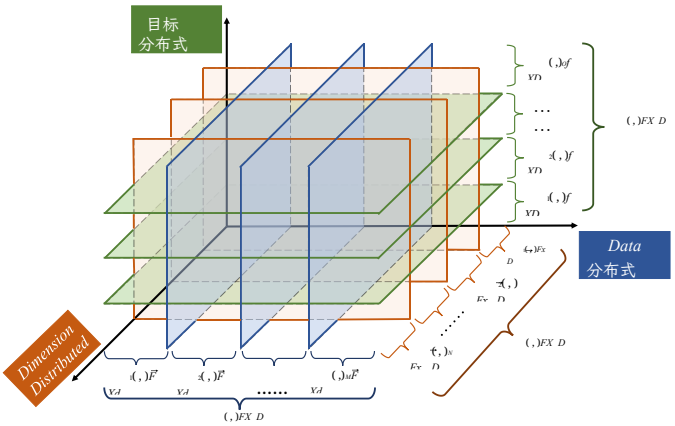


图6。分布式优化问题的分布式信息。蓝色平面将数据划分为多个分布式部分。橙色平面将维度划分为多个

分布式部分。绿色平面将目标划分为多个分布式部分。

现有的优化方法。

*2)*定义与分类体系

不失一般性，一个最小化优化问题可以表述如下：

$$\min F(X,D), X \in \Omega \tag{4}$$

其中，F 是智能体旨在通过协作最小化的目标，D 是环境数据集。X 是待优化的变量向量，Ω 是X的变量空间。在分布式优化中，空间分布信息可以是环境数据D、变量维度X和目标F。每个分布式智能体仅持有部分信息，所有智能体的信息合起来构成完整信息。

如图6所示，平面将全部信息划分为不同部分，这些部分由不同的智能体拥有。特别地，蓝色平面表示数据的划分，而维度和目标的空间是整体。这是第一种基本问题：数据分布式优化问题，其形式如下：

$$\begin{aligned} &\min F(X,\vec{d}_i), X \in \Omega, i = 1,2,...,M \\ &D = \{\vec{d}_1, \vec{d}_2, ..., \vec{d}_i, ..., \vec{d}_M\} \end{aligned} \tag{5}$$

图6中的橙色平面表示维度的划分，而数据空间和目标空间是完整的。这是第二种基本问题：维度分布式优化问题，其形式如下：

$$\begin{aligned} &\min F(\vec{x}_i, D), X \in \Omega, i = 1,2,...,N \\ &X = \{\vec{x}_1, \vec{x}_2, ..., \vec{x}_i, ..., \vec{x}_N\} \end{aligned} \tag{6}$$

图6中的绿色平面表示目标的划分，而数据空间和维度空间是整体。这是第三种基本类型的问题：目标分布式优化问题，其形式如下：

$$\begin{aligned} &\min f_i(X,D), X \in \Omega, i = 1,2,...,O \\ &F = \{f_1, f_2, ..., f_i, ..., f_O\} \end{aligned} \tag{7}$$

其中，M 是环境数据的部分数量，N 是变量维度的部分数量，O 是目标的过去部分数量。从这三个方面来看，有两项原则需要明确。

原则 1) 由不同智能体持有的分布式信息可以是同质的，也可以是异质的。

原则 2) 所有智能体拥有的分布式信息的并集覆盖整个空间。

为了直观地介绍，图7展示了分布式优化问题的类别。图7中的每个立方体代表一个智能体。长度、宽度和高度分别代表所拥有的数据、维度和目标。子图 a) 展示了数据分布式优化问题中的三个智能体 a_i、a_j、a_k 。智能体拥有相同的完整维度空间和目标空间，但数据不同。可以看出，a_i 的数据集与a_j 和a_k的都不同，而a_j 和a_k 则存在重叠区域。图 b) 所示的维度分布式优化问题和图 c) 所示的目标分布式优化问题中也存在类似情况。这体现了上述两个原则：不同智能体拥有的分布式信息可以是同质的（重叠区域）或异质的（可分区域），且所有智能体拥有的分布式信息的并集覆盖整个空间。在优化这三类基本优化问题时，设计恰当的协作技术以实现分布式和局部信息学习至关重要。

第四类分布式优化问题是三种信息的混合体。顾名思义，这类分布式优化问题可能具有两种甚至三种（a_r 在图 7 d)中）分布式信息。如子图d)所示，a_i、a_j、a_k 是两种混合信息分布式优化问题，可表述为：

$$\begin{aligned} &\min f_i(X,\vec{d}_i), X \in \Omega, i = 1,2,...,I \\ &F = \{f_1, f_2, ..., f_i, ..., f_I\} \\ &D = \{\vec{d}_1, \vec{d}_2, ..., \vec{d}_i, ..., \vec{d}_I\} \end{aligned} \tag{8}$$

$$\begin{aligned} &\min f_j(\vec{x}_j, D), X \in \Omega, j = 1,2,...,J \\ &F = \{f_1, f_2, ..., f_j, ..., f_J\} \\ &X = \{\vec{x}_1, \vec{x}_2, ..., \vec{x}_j, ..., \vec{x}_J\} \end{aligned} \tag{9}$$

$$\begin{aligned} &\min F(\vec{x}_k, \vec{d}_k), X \in \Omega, k = 1,2,...,K \\ &X = \{\vec{x}_1, \vec{x}_2, ..., \vec{x}_k, ..., \vec{x}_K\} \\ &D = \{\vec{d}_1, \vec{d}_2, ..., \vec{d}_k, ..., \vec{d}_K\} \end{aligned} \tag{10}$$

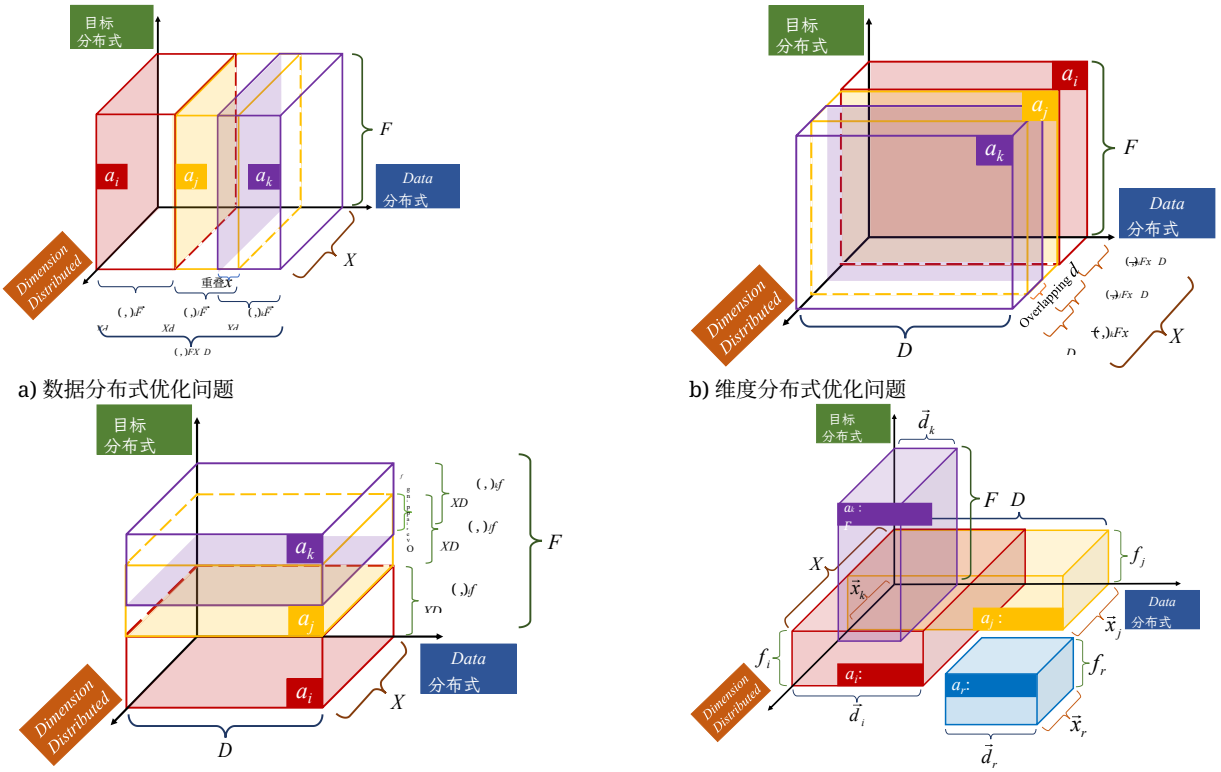
a_r 在子图 d) 中是三种混合信息的不可信优化问题，可表述为：

$$\begin{aligned} &\min f_r(\vec{x}_r, \vec{d}_r), X \in \Omega, r = 1,2,...,R \\ &F = \{f_1, f_2, ..., f_r, ..., f_R\} \\ &X = \{\vec{x}_1, \vec{x}_2, ..., \vec{x}_r, ..., \vec{x}_R\} \\ &D = \{\vec{d}_1, \vec{d}_2, ..., \vec{d}_r, ..., \vec{d}_R\} \end{aligned} \tag{11}$$

其中，i、j、k、r分别表示第 i 个、第 j个、第 k 个和第 r个智能体，而 I、J、K、R 分别表示不同情形下智能体的总数。混合信息分布式优化问题更难进行优化，因为智能体应利用更多信息开展协同学习。

B. 面向数据分布式优化问题的分布式进化计算

在数据分布式优化问题中，有两种情况受到广泛关注：异构数据



a) 数据分布式优化问题　b) 维度分布式优化问题

c) 目标分布式优化问题　d) 混合信息分布式优化问题

图7。分布式优化问题的分类体系。a) 数据分布式优化问题。b)维度分布式优化问题。c) 目标分布式优化问题。d) 混合信息分布式优化问题。每个立方体表示一个智能体所拥有的信息，其中长度、宽度和高度分别表示数据、维度和目标。

分布式优化和隐私数据分布式优化。

1)面向异构数据分布式优化的分布式进化计算

由于从多个来源收集的数据通常规模庞大、动态多变且异构，因此处理起来颇具挑战。一些挖掘任务（例如 *K*-means 聚类）因分布式数据而产生 [138], , 并推动了分布式场景下进化聚类技术的发展。Ishibuchi 等人 [139] 提出了一种分布式基于遗传学的机器学习（GBML）方法。该方法将进化计算算法用于知识提取和数据挖掘，可用于设计模糊规则分类器。模糊 GBML 能够妥善处理具有不同缺失属性的分布式异构数据。Oliveira 等人[140] 研究了可扩展数据集，并提出了两种基于 MapReduce 的进化可扩展元启发式算法，统称为 SF-EAC。第一种算法旨在通过进化技术改进*K*-means 集群，以处理分布式数据；第二种算法旨在将集群组合成集成。Hajeer 等人 [141] 开发了分布式遗传算法应用框架 DEGA-Gen，用于检测网络中的社群。他们详细说明了如何将各种网络记录为染色体。基于 DEGA-Gen，Hajeer 和 Dasgupta [142] 为遗传算法设计了一种分布式编码技术，以高效处理分布式数据。他们利用 Hadoop 上通过 Lehigh大学基准测试（LUBM）生成的分布式数据集，对所提出的方法进行了测试。结果表明，所提出的数据感知型Hadoop和进化聚类技术能够有效处理结构各异的异构数据。

2)面向隐私数据分布式优化的分布式进化计算

大规模分布式数据引发了对隐私和安全的担忧。联邦学习是一种由分布式数据驱动的机器学习范式，能够降低隐私和安全风险 [134]。它协同多个设备，基于各自的本地数据间接训练一个全局模型，而不是将所有本地数据收集到中央设备。Xu等人。[131] 利用联邦学习的优势，提出了用于隐私数据分布式优化的联邦数据驱动进化优化框架。在联邦学习的基础上，他们开发了代理辅助联邦数据驱动进化算法FDD-EA。所有本地设备训练一个径向基函数网络，并将参数发送至服务器。服务器使用排序平均法聚合全局代理模型，以辅助进化。通过这种方式，原始本地数据保存在本地设备中，从而可以防止在发送数据过程中发生数据泄露。通过在带噪声和非独立同分布（non-IID）数据的分布式环境中进行大量实验，所提出的方法与最先进的集中式数据驱动代理辅助进化算法相比，表现出了具有竞争力的性能。此外，Xu 等人。[143] 将这一思想扩展到多/多目标优化，并提出了联邦数据驱动多/多目标进化优化算法（FDD-MOEA）。这是首项将联邦学习与多目标进化算法相结合以保护数据隐私的工作。

C. 维度分布式优化问题的分布式进化计算

维度分布式优化问题分为两种情况：非重叠维度分布式优化问题和重叠维度分布式优化问题。与前者相比，后者的优化难度更大。

1)非重叠维度分布式优化的分布式进化计算

非重叠是维度分布式优化问题中一种基本且简单的情形。协作协同进化进化算法（CCEAs）是一类适用于非重叠维度分布式优化的进化计算算法。不过，CCEAs 包含分解过程，而如何将变量划分为多个互不重叠的子组件是 CCEAs 的研究热点。相较之下，维度分布式优化问题具有天然的维度子组件特性。因此，在维度分布式优化中，只需进行协同优化与协同适应。

Utkarsh 等. [144] 研究了电力配电和输电网络中配备电池储能系统（BSS）的可再生能源（RES）。他们提出了一种基于共识的维度分布式粒子群优化算法，用于优化输电系统运营商（TSO）的辅助服务，以及分布式网络运营商（DNO）和可再生能源所有者的技术与经济效益。他们以完全分布式方式实现了所提出的算法，其中每个节点代表一个智能体，且只能与其邻居通信。协同适应是指利用邻居的当前状态进行适应度评估，并以同步方式在所有节点间执行。在 30 个节点上的实验表明，维度分布式粒子群优化能够实时满足 TSO 的辅助服务请求，并为 DNO 和可再生能源所有者带来收益。此外，Utkarsh 等. [133] 聚焦于智能电网中的实时大规模优化问题。他们进一步开发了一种基于共识的维度分布式粒子群优化算法，以辅助智能分布式电网中基于可再生能源的分布式发电机（DG）和可控负荷（CL）的实时最优控制。同样，协同适应是指以同步方式利用邻居的当前状态进行适应度评估。通过在 30 节点和 119 节点配电测试系统上的仿真，结果表明所提出的方法具有更好的收敛性、适应性和最优性。此外，他们提出了一种新的梯度模块，并扩展了基于共识的维度分布式粒子群优化算法，以优化多微电网系统 [145]。

2）*DEC f*或 *Overla pping* 维度 分布式 *O* *p*优化

因式分解进化算法（FEAs）是一种新近提出的进化计算范式，它将原始问题分解为多个维度子集来优化 [146]。与协同协作进化算法类似，FEAs 通过因式分解过程将各个维度划分为多个子集。由于维度分布式优化问题天然具有维度子组件这一特征，因此求解这类问题时只需要竞争策略和共享策略。与协同协作进化算法不同，FEAs 鼓励各维度子集彼此重叠，从而允许这些子集与其他子集进行竞争或协同进化 [147]。因此，它是适用于重叠维度分布式优化的进化计算范式。

Fortier 等。[148] 提出了分布式重叠群体智能（DOSI），用于优化人工神经网络中的参数。他们为每个神经元应用一个群体。相关神经元中的群体具有需要优化的重叠参数，并且可以定期相互通信。适应度较高的群体可以将其信息分享给适应度较低的群体，同时彼此竞争。较低的计算复杂度使 DOSI 能够应用于大型神经网络。然而，分布式方式使其缺乏全局视角。基于 DOSI，Fortier 等。[147] 提出了几种多群算法，用于在贝叶斯信念网络中执行完全和部分溯因推理。每个节点都有一个群体，可以从其马尔可夫毯中学习状态分配。实验表明，DOSI 可以优于其他非群体智能方法。

D. 用于目标分布式优化问题的分布式进化计算

本质上，目标分布式优化问题是一种特殊的多/多目标优化问题，其中各目标通过不同的软件仿真、授权组织甚至物理实验进行评估。

针对目标分布式优化问题的分布式进化计算研究还不多。Oyama 等人 [149] 研究了多种汽车车型的同步结构设计，并提出了一种名为 Cheetah 的同步汽车结构设计进化算法。需要同步优化三种汽车：MAZDA CX-5（SUV 汽车）、MAZDA6 Wagon（大型汽车）和 MAZDA3 Hatchback（小型汽车）。优化目标是尽可能降低三种汽车的总重量，并尽可能增加共用厚度部件的数量。软件仿真工具 iSPAN 和 LS-DYNA 分别用于分析获得的设计方案和碰撞模式，两者相互分布。所提出的 Cheetah 算法在京计算机上通过了验证 [150], ，在由 80,000 多个节点组成的超级计算机上运行耗时超过 360 小时。

由于数百乃至数千次适应度评估耗时难以承受，人们将代理模型与进化算法结合，由此产生了代理辅助进化算法（SAEAs）[151]。为最大限度提高整个风电场的功率产出并尽可能降低单台风力涡轮机的结构载荷，Yin 等人 [152] 提出了一种数据驱动的多目标预测控制方法。鉴于风电场的结构天然具有分布式特性，且运行计算流体力学（CFD）仿真极为耗时，他们采用了若干模型来近似目标。Floris 工具用于表征气动尾流相互作用，并生成用于预测控制设计的数据。该工具由尾流衰减、尾流偏转和尾流扩张三个模型组成。风电场预测器（WFP）用于预测功率和推力载荷。它由支持向量机（SVM）构建，并与速度约束多目标粒子群优化（SMPSO）算法协同工作，以生成优化后的偏航角

设置。实验表明，所提出的方法可将风电场推力最多降低 12.96%，同时基本保持功率产出。Wei 等人. [153] 提出了一种采用按需评估的分布式、高成本进化约束优化方法。他们优化了分布式问题，其中各智能体对目标或某项约束的评估各不相同。由于每个智能体只掌握部分适应度信息，因此智能体之间通过与服务器通信进行协作。按需评估策略节省了评估预算，以供后续进化使用，从而提升了算法性能。

VII. 挑战与潜在研究方向

随着物联网和分布式计算范式的发展，以分布式方式感知和处理数据已日益普遍。进化计算固有的并行性为分布式进化计算在此类分布式计算环境中优化复杂问题提供了巨大潜力。尽管分布式进化计算的多个方面已得到研究，但它仍面临诸多挑战，也蕴含着许多机遇。这些挑战与机遇带来了许多潜在的研究方向，有望推动分布式进化计算的发展。

*1)*分布式进化计算的理论分析

收敛性分析和运行时间分析等理论分析长期以来一直是进化计算中重要且具有挑战性的问题。尽管进化计算的一些理论基础已得到研究，但在并行和分布式环境中，分布式进化计算的特点给理论分析带来了新的挑战。

i) 收敛性分析。收敛性是进化计算理论分析中的一个重要方面，已受到广泛关注 [154]，[155]。然而，分布式并行性给分布式进化计算带来了巨大挑战。首先，在子种群并行中，通信对收敛起着至关重要的作用。高频通信有助于增强全局收敛性，而低频通信则会削弱全局收敛性。研究分布式进化计算中通信与收敛之间的理论关系意义重大。此外，在协同协进化的子种群并行中，原问题被划分为若干具有不同维度集合的子问题。各子问题的优化结果会被整合为原问题的结果。因此，如何从理论上确保子问题能够收敛，是一项挑战。此外，在目标协同进化的子种群并行中，每个种群负责一个或多个目标。任何种群都无法从全局视角考察所有目标。如何从理论上保证子种群能够逼近帕累托前沿，仍有待进一步研究。

ii）效率分析。效率是进化计算中的一个广泛研究主题 [44]。由于分布式进化计算最初是为提高效率而开发的，一些研究人员对分布式进化计算中的理论效率进行了分析，尤其是适应度评估分配和通信策略 [66], [26]。执行时间的加速比是一项重要指标。一般而言，分布式进化计算的执行时间 T 由三部分组成：T_1, 进化时间、T_2, 评估时间，以及 T_3, 通信时间。然而，分布式计算技术为效率带来了新的挑战。分析。例如，不同的计算资源具有不同的执行性能，这可能导致不同的 T_1 和 T_2。此外，通信时间 T_3 也会受到传输的显著影响。因此，研究一种基于 T 的公平且稳健的效率分析指标是值得的。

iii)一致性分析。一致性是分布式优化中的一个重要研究方向。例如，维度分布式优化问题由局部目标和全局目标组成。通常，局部评估和全局评估无法实现一致性。一种可能的情况是，子问题通过局部评估逐渐趋近于更优的目标值，而所有子问题的全局评估却逐渐趋近于更差的位置。此外，不同的子问题可能具有重叠的维度。重叠的维度代表相互依赖的子问题之间的交互。由于子问题的分布，不同子问题中对重叠维度的优化可能存在冲突。因此，如何保持目标和维度的演化一致性是一个至关重要的研究问题。

*2)*新型并行与分布式计算基础设施上的分布式进化计算

随着人工智能（AI）的蓬勃发展，人工智能（AI）芯片作为研究对象受到关注，Krestinskaya 等人。 [157] 研究了利用遗传搜索实现模拟 AI 芯片设计自动化。作为 AI 的重要分支，研究 EC 和 DEC 芯片前景广阔，因为它们在学术界和工业工程领域都有广泛应用。然而，如何在芯片有限的存储空间下实现 DEC 算法是一项巨大挑战，尤其是在大规模优化方面。幸运的是，片上系统（SoC）和片上网络（NoC）为开发 EC 芯片铺平了道路。一些研究人员已着手优化 SoC 和 NoC 中的映射问题。例如，Erbas 等人。 [158] 研究了用于 SoC 应用映射问题的多目标进化算法。Ascia 等人。 [159] 研究了 NoC 上基于进化算法的知识产权模块拓扑映射。

此外，随着并行与分布式计算技术的发展，一些机器学习和进化计算平台已相继提出。例如，Ribeiro 等人 [160] 构建了开箱即用的机器学习即服务（MLaaS），提供可扩展、灵活且无阻塞的平台。Tian 等人 [161] 开发了一个用于进化多目标优化的 MATLAB 平台，Gribble 等人 [162] 则提出了一个面向基础设施服务的进化平台。受 MLaaS 和进化计算平台的启发，可以在分布式计算基础设施上开发作为服务的分布式进化计算平台，以支持大规模优化、分布式优化、移动优化等。然而，在设计分布式进化计算平台即服务时，有一些问题需要考虑。例如，平台处理能力与任务量之间的关系、平台存储空间与待处理数据量之间的关系，以及传输带宽与通信频率和通信量之间的关系等。

3)面向大数据的分布式进化计算

进化计算一直是广泛应用于数据驱动优化的方法 [151]。随着大数据时代的到来和物联网的发展，数据在不同地点生成并存储，难以集中融合。这给分布式进化计算带来了巨大挑战，即数据孤岛问题。智能体只能获取局部的部分数据，因此难以搜索全局最优解。Xu 等人 [131], [143] 聚焦这一挑战，开发了联邦式数据驱动进化计算。Guo 等人 [163] 提出了一种用于分布式数据驱动优化的边缘—云协同进化算法。他们基于本地分布式数据训练局部模型，再将其融合，聚合为用于全局优化的全局模型。此外，数据异质性也带来了很大困难。数据相互独立且同分布（i.i.d.）是理想情况。然而，在某些情况下，由于智能体行为的不确定性和设备的异质性，数据可能并非独立同分布。如何在分布式智能体之间设计通信和评估策略，是一项重大挑战。更糟的是，一些分布式数据可能由恶意攻击者生成。如何检测分布式智能体的信誉并预处理分布式数据，也值得研究。除了上述挑战之外，从海量分布式数据中挖掘并复用知识也具有潜力，由此推动了分布式进化计算在迁移优化方面的研究，以解决分布式多任务优化问题 [164]。

4)分布式进化计算的隐私与安全

随着云计算/边缘计算的发展与普及，将进化计算部署在云计算平台上，可为各种复杂问题提供稳健优化服务。在这种情况下，隐私与安全受到越来越多的关注，这也带来了分布式进化计算领域的另一项研究需求。分布式进化计算中的数据、维度或目标评估等信息分布在不同的智能体中，因此，隐私保护至关重要，各智能体也不愿向他人泄露原始信息。因此，研究分布式进化计算的隐私与安全意义重大。一些研究人员已尝试在数据分布式优化中保护数据。例如，Xu 等 [131], [143] 将联邦学习与进化计算相结合，开发了用于隐私保护的联邦数据驱动进化计算。Zhao 等 [165] 提出了一种保护隐私的多智能体粒子群优化算法。该算法能够在不泄露数据的情况下实现令人满意的全局优化。此外，Du 等 [166] 构建了一种用于社交网络隐私保护的社区结构演化博弈。对于分布式进化计算的未来发展，研究维度分布式和目标分布式优化问题中的隐私保护也值得关注。

5)分布式进化计算的应用

分布式进化计算的发展显著提升了进化计算的效率和可扩展性，使我们能够在学术界和工业界将进化计算应用于更复杂或更大规模的优化问题。例如，蛋白质折叠、天文干涉测量等科学优化问题，都是包含数千甚至数百万个变量的复杂大规模优化问题。此外，一些工业应用，如汽车模型结构优化、制造优化、交通优化等，也给进化计算带来了诸多挑战，例如评估成本高昂。值得庆幸的是，随着分布式进化计算和计算技术的发展，分布式进化计算的效率和可扩展性都得到了大幅提升。它已成为解决这类问题的一种很有前景的方法。

除了大规模和计算成本高昂的优化外，将差分进化计算应用于分布式优化问题也大有前景。例如，在无人系统和多智能体系统中，不仅需要使用进化计算算法来优化路径规划或调度问题，也迫切需要应用差分进化计算来提升系统的自主性和智能，使各实体能够通过进化操作协同优化任务 [167], [168]。然而，将差分进化计算应用于无人系统和多智能体系统时，共识理论的研究仍是一项重大且备受关注的挑战。

VIII. 结论

随着分布式计算范式的快速发展，在分布式计算平台上实现和开展进化计算日益受到关注。本文从目的、算法的并行结构、实现并行模型和实现环境等方面，为分布式进化计算提出了新的分类体系。具体而言，分布式进化计算的两大主要目的分别是：通过并行处理提高效率，以及通过具有部分信息的分布式子种群协作执行分布式优化。通过全面综述，本文表明，分布式进化计算不仅能够利用进化计算的自然并行性来提高搜索效率，还能够使具有局部变量、数据和目标的空间分布个体/子种群实现分布式协同进化，从而实现可扩展分布式优化。此外，鉴于分布式优化是分布式进化计算中一个新兴且引人关注的趋势，我们进一步对这类问题给出了系统且形式化的定义。本文综述了现有针对维度分布式、数据分布式和目标分布式优化问题的分布式进化计算研究，并讨论了这些问题对分布式进化计算提出的主要挑战。

物联网以及超级计算、云计算和边缘计算等分布式计算平台的发展，使数据感知、处理和计算无处不在。得益于进化计算固有的并行性，人们认为分布式进化计算能够成为在此类分布式计算环境中解决大规模和分布式优化问题的重要技术，但仍面临诸多挑战和有待解决的研究问题。我们讨论了分布式进化计算的一些潜在研究方向，例如分布式进化计算的理论分析、不同分布式计算基础设施上的分布式进化计算、面向大数据的分布式进化计算、分布式进化计算的隐私和安全问题，以及分布式进化计算在各种大规模计算昂贵优化问题（如模型结构优化、制造优化等）和分布式优化问题（如无人系统、多智能体系统等）中的应用。

参考文献

[1] T. Bäck、U. Hammel 和 H. P. Schwefel，“进化计算：关于其历史和现状的评论”，IEEE进化计算汇刊，第 1 卷，第 1 期，第 3–17 页，1997年，doi：10.1109/9780470544600.ch1。[2] K. S. Tang、K. F. Man、S. Kwong 和 Q. He，“遗传算法及其应用”，IEEE《信号处理杂志》，第 13， 6， 22 37, 卷，第期，1996， 10 1007 978 1 4615 5237 6_2 页码–，doi：./----.[3] W. Banzhaf、P. Nordin、R. E. Keller 和 F. D. Francone，《遗传编程：入门》，第 1 卷。1998年。[4] M. Dorigo、M. Birattari 和 T. Stutzle，“蚁群优化”，IEEE《计算智能杂志》，第 1 卷，第 4 期，第 28–39 页，2006年。[5] J. Kennedy 和 R. Eberhart，“粒子群优化”，载于 ICNN’95 国际神经网络会议，1995年，第 1942–1948 页。[6] W. N. Chen、D. Z. Tan、Q. Yang、T. Gu 和 J. Zhang，“用于控制社交网络中污染物扩散的蚁群优化”，IEEE 控制论汇刊，第 50 卷，第 9 期，第 4053–4065 页，2020年，doi：10.1109/TCYB.2019.2922266。[7] J. Tang、G. Liu 和 Q. Pan，“用于求解优化问题的代表性群体智能算法综述：应用与趋势”，IEEE/CAA《自动化学报》，第 8 卷，1643，2021年，10，1627 10 1109 2021 1004129，第期，页码–，doi：./JAS...[8] Y. H. Jia 等人，“基于集合的粒子群优化动态物流调度系统”，IEEE《系统、人类与控制论：系统》，第 48 卷，第 9 期，第 1607–1621 页，2018年，doi：10.1109/TSMC.2017.2682264。[9] H. Nikolov、T. Stefanov 和 E. Deprettere，“多处理器系统设计、编程与实现的系统化自动化方法”，IEEE《计算机辅助设计集成电路与系统汇刊》，第 27 卷，555，2008年，3，542 10 1109 2007 911337，第期，页码–，doi：./TCAD...[10] S. Ryoo 、 C. I. Rodrigues 、 S. S. Baghsorkhi 、 D. B. K. W. W. Hwu 和 S. S. Stone，“使用 CUDA 的多线程图形处理器优化原则及应用性能评估”，载于 PPoPP ’08：第 13 届 ACM SIGPLAN 并行编程原理与实践研讨会论文集，2008年，第 73–82 页。[11] K. Lee、Y. Lee、H. Choi、Y. D. Chung 和 B. Moon，“使用 MapReduce 的并行数据处理：综述”，SIGMOD Rec.，第 40 卷，第 4 期，第 11–20 页，2011年。[12] E. Alba 和 M. Tomassini，“并行性与进化算法”，IEEE进化计算汇刊，第 6 卷，第 5 期，第 443–462 页，2002年，doi：10.1109/TEVC.2002.800880。[13] M. Tomassini，“并行与分布式进化算法：综述”，《遗传算法与遗传编程》，第 1–25 页，1999年。[14] Y. J. Gong 等人，“分布式进化算法及其模型：最新进展综述”，《应用软计算期刊》，第 34 卷，第 286–300 页，2015年，doi：10.1016/j.asoc.2015.04.061。[15] Z. H. Zhan 等人，“基于矩阵的进化计算”，IEEE 计算智能新兴主题汇刊，第 1–14 页，2021年，doi：10.1109/TETCI.2020.3047410。[16] K. C. Tan、Y. J. Yang 和 C. K. Goh，“用于多目标优化的分布式协作协同进化算法”，IEEE进化计算汇刊，第 10 卷，第 5 期，第 527–549 页，2006年，doi：10.1109/TEVC.2005.860762。[17] Y. H. Jia 等人，“面向大规模优化的自适应计算资源分配分布式协同协进化”，IEEE进化计算汇刊，第 23 卷，第 2 期，第 188–202 页，2019年，doi：10.1109/TEVC.2018.2817889。[18] J. Starzyński、R. Szmurło、J. Kijanowski、B. Dawidowicz、B. Sawicki 和 S. Wincenciak，“用于电磁学优化的分布式进化算法”，IEEE《磁学汇刊》，第 42 卷，1246，2006 年，4，1243 10 1109 2006 871998，第期，页码 –，doi：/TMAG...[19] A. Song 、 W.-N. Chen 、 T. Gu 、 H. Yuan 、 S. Kwong 和 J. Zhang，“具有历史档案和基于集合的粒子群优化的分布式虚拟网络嵌入系统”，IEEE《系统、人类与控制论：系统》，第 1–16 页，2019年，doi：10.1109/tsmc.2018.2884523。[20] W. Shi、J. Cao、Q. Zhang、Y. Li 和 L. Xu，“边缘计算：愿景与挑战”，IEEE《物联网期刊》，第 3 卷，第 5 期，第 637–646 页，2016年，doi：10.1109/JIOT.2016.2579198。[21] M. N. O. Sadiku 和 S. M. Musa，“云计算：机遇与挑战”，IEEE《潜力》，第 33 卷，第 1 期，第 34–36 页，2014 年。[22] A. Dorri 、 S. S. Kanhere 和 R. Jurdak，“多智能体系统：综述”，IEEE Access，第 6 卷，第 28573–28593 页，2018年，doi：10.1109/ACCESS.2018.2831228。

[23] M. Biazzini 和 A. Montresor，“P2POEM：P2P网络中的函数优化”，《Peer-to-Peer Netw. Appl.》，第6卷，第2期，第213–232页，2013年，doi：10.1007/s12083-012-0152-8。[24] S. Yi、Z. Hao、Z. Qin 和 Q. Li，“雾计算：平台与应用”，载于《第3届Web系统与技术热点研讨会论文集》（HotWeb 2015），2016年，第73–78页，doi：10.1109/HotWeb.2015.22。[25] Z. Ren、Y. Liang、A. Zhang、Y. Yang、Z. Feng 和 L. Wang，“采用细粒度计算资源分配策略提升大规模优化中的协作协同进化”，《IEEE 控制论汇刊》，第49卷，第12期，第4180–4193页，2019年，doi：10.1109/TCYB.2018.2859635。[26] J. Li、K. Du、Z. Zhan、H. Wang 和 J. Zhang，“具有自适应资源分配的分布式差分进化”，《IEEE 控制论汇刊》，第1–14页，2022年。[27] I. De Falco、U. Scafuri、E. Tarantino 和 A. Della Cioppa，“用于分布式差分进化的异步自适应多种群模型”，载于《IEEE进化计算大会》（CEC 2016），2016年，第5010–5017页，doi：10.1109/CEC.2016.7748324。[28] I. De Falco、A. Della Cioppa、D. Maisto、U. Scafuri 和 E. Tarantino，“用于分布式差分进化的自适应入侵模型”，《信息科学》，第278卷，第653–672页，2014年，doi：10.1016/j.ins.2014.03.083。[29] J. Apolloni、G. Leguizamón、J. García-Nieto 和 E. Alba，“基于岛的分布式差分进化：混合测试平台上的实验研究”，载于《第8届混合智能系统国际会议》（HIS 2008），2008年，第696–701页，doi：10.1109/HIS.2008.87。[30] I. De Falco、A. Della Cioppa、U. Scafuri 和 E. Tarantino，“在分布式差分进化的异步迁移模型中利用多样性”，《遗传与进化计算会议伴随论文集》，第1880–1887页，2017年，doi：10.1145/3067695.3084217。[31] D. Jakobović、M. Golub 和 M. Čupić，“异步隐式并行进化计算模型”，《软计算》，第18卷，第6期，第1225–1236页，2014年，doi：10.1007/s00500-013-1140-5。[32] Z. H. Zhan 等人，“Cloudde：一种异质差分进化算法及其分布式云版本”，《IEEE 并行与分布式系统汇刊》，第28卷，第3期，第704–716页，2017年，doi：10.1109/TPDS.2016.2597826。[33] F. Fernández、G. Galeano 和 J. A. Gómez，“同步与异步并行分布式遗传编程模型的比较”，载于《欧洲遗传编程会议》，2002年，第2278卷，第326–335页，doi：10.1007/3-540-45984-7_32。[34] E. Alba 和 J. M. Troya，“同步与异步并行分布式遗传算法分析”，《未来世代计算机系统》，第17卷，第4期，第451–465页，2001年，doi：10.1016/S0167-739X(99)00129-6。[35] G. V. Neville-Neil，“分而治之”，《Commun. ACM》，第64卷，第10期，第25页，2021年，doi：10.1145/3481431。[36] M. N. Omidvar、X. Li 和 X. Yao，“基于贡献度智能利用计算资源的协同协进化算法”，载于《遗传与进化计算会议》（GECCO’11），2011年，第1115–1122页，doi：10.1145/2001576.2001727。[37] Z. Yang、K. Tang 和 X. Yao，“利用协作协同进化进行大规模进化优化”，《信息科学》，第178卷，第15期，第2985–2999页，2008年，doi：10.1016/j.ins.2008.02.017。[38] X. M. Hu、F. L. He、W. N. Chen 和 J. Zhang，“用于大规模优化的快速互依关系识别协作协同进化”，《信息科学》，第381卷，第142–160页，2017年，doi：10.1016/j.ins.2016.11.013。[39] M. N. Omidvar、Y. Ling、Y. Mei 和 X. Yao，“采用差分分组进行大规模全局优化的协同协进化”，《IEEE进化计算汇刊》，第18卷，第3期，第378–393页，2014年，doi：10.1109/FSKD.2016.7603157。[40] Y. Sun、M. Kirley 和 S. K. Halgamuge，“通过扩展差分分组，利用直接和间接变量交互进行大规模全局优化”，载于《GECCO 2015——2015年遗传与进化计算大会论文集》，2015年，第2卷，第313–320页，doi：10.1145/2739480.2754666。[41] Y. Mei、M. N. Omidvar、X. Li 和 Y. Xin，“用于无约束大规模黑盒优化的竞争式分而治之算法”，《ACM Trans. Math. Softw.》，第42卷，第2期，第13页，2016年。[42] M. N. Omidvar、M. Yang、Y. Mei、X. Li 和 X. Yao，“DG2：一种更快速、更准确的大规模黑盒优化差分分组方法”，《IEEE进化计算汇刊》，第21卷，第6期，第929–942页，2017年，doi：10.1109/TEVC.2017.2694221。[43] Y. Sun、M. Kirley 和 S. K. Halgamuge，“一种递归分解 M

ethod for Large Scale Conti$_{nuous}$ Optimization,” *IEEE T rans. Evol.*

计算，卷22，第5期，第647–661页，2018年，doi:10.1109/TEVC.2017.2778089.[44] M. Yang、A.Zhou、C. Li和X. Yao，“面向大规模连续问题的高效递归差分分组”，IEEE进化计算汇刊，卷25，第1期，第159–171页，2021年，doi: 10.1109/TEVC.2020.3009390.[45] J.-Y. Li、Z.-H.Zhan、K. C. Tan和J. Zhang，“双重差分分组：一种更通用的大规模优化分解方法”，IEEE 控制论汇刊，第1–15页，2022年，doi:10.1109/tcyb.2022.3158391. [46] X.Ma、Z. Huang、X.Li、L.Wang、Y. Qi和Z.Zhu，“面向大规模全局优化的合并差分分组”，IEEE进化计算汇刊，卷99，第1期，第1–13页，2022年，doi:10.1109/TEVC.2022.3144684.[47] M. N. Omidvar、X.Li、Z.Yang和X.Yao，“通过更频繁的随机分组实现大规模优化的协同协进化”，载于IEEE进化计算大会，CEC 2010，2010年，第1–8页，doi: 10.1109/CEC.2010.5586127.[48] M. N.Omidvar、X. Li、Y.Mei和X.Yao，“采用δ分组进行大规模非可分函数优化的协同协进化”，载于IEEE进化计算大会，CEC 2010，2010年，第1– 8页，doi:10.1109/TEVC.2013.2281543. [49] X. Y. Zhang、Y. J.Gong、Y. Lin、J. Zhang、S. Kwong和J.Zhang，“大规模优化的动态协同协进化”，IEEE进化计算汇刊，卷23，第6期，第935–948页，2019年，doi:10.1109/TEVC.2019.2895860.[50] X.Cai等，“采用基于知识的动态变量分解进行双层多目标优化的协同协进化”，IEEE进化计算汇刊，第1–1页，2022年，doi:10.1109/tevc.2022.3154057.[51] X. Li和X.Yao，“面向大规模优化的协同进化粒子群”，IEEE进化计算汇刊，卷16，第2期，第210–224页，2012年，doi:10.1109/TEVC.2011.2112662. [52] L. Panait和S. Luke，“选择信息性动作可改善协作式多智能体学习”，载于自主智能体国际会议论文集，2006年，卷2006，第760–766页，doi: 10.1145/1160633.1160767.[53] R. P. Wiegand、W.C. Liles和K.A.DeJong，“协同协进化算法中协作方法的实证分析”，载于遗传与进化计算会议论文集（GECCO-2001），2001年，第1996号，第1235–1242, [在线]。可访问：http://www.cs.bham.ac.uk/~wbl/biblio/gecco2001/d23.pdf。[54] C.-K. Au和H.-F. Leung，“协同协进化中随机移民方案协作方法的研究”，载于2009年IEEE进化计算大会，CEC 2009，2009年，第2701–2707页。 [55] E. Glorieux、B. Svensson、F. Danielsson和B.Lennartson，“面向大规模优化的改进型构造式协同协进化差分进化”，载于2015年IEEE计算智能研讨会系列论文集，SSCI 2015，2015年，第1703–1710页，doi: 10.1109/SSCI.2015.239.[56] C.Stoean，“用于分类的协同协进化中的各种协作者选择压力”，载于人工智能与数字通信国际会议论文集，2006年，第45–53页。[57] Y. S.Son和R.Baldick，“具有局部最优解的博弈中用于纳什均衡搜索的混合协同进化编程”，IEEE进化计算汇刊，卷8，第4期，第305–315页，2004年，doi:10.1109/TEVC.2004.832862.[58] R. Allmendinger和J.Knowles，“异质目标：最新进展与未来研究”，arXiv预印本arXiv:2103.15546，2021年。[59] R. Allmendinge 、 J.Handl 和 J.Knowles，“ 多目标优化：当目标表现出非均匀延迟时”，欧洲运筹学杂志，卷243，第2期，第497–513页，2015年，doi: 10.1016/j.ejor.2014.09.033. [60] A.Lewis、S. Mostaghim和I.Scriven，“不可靠分布式环境中的异步多目标优化”，载于生物启发式优化方法，卷210，2009年，第51–78页。 [61] E. O. Scott和K. A.DeJong，“异步进化算法中的评估时间偏差”，载于2015年遗传与进化计算大会，2015年，第1209–1212页，doi:10.1145/2739482.2768482. [62] M. Yagoubi、L.Thobois和M. Schoenauer，“具有异质评估成本的异步进化多目标算法”，载于2011年IEEE进化计算大会，2011年，2011年7月，第21–28页，doi:10.1109/CEC.2011.5949593.[63] J. F. Schutte、J. A. Reinbolt、B. J. Fregly、R.T.Haftka和A.D. George，“采用粒子群算法的并行全局优化”，国际期刊。《数值方法与工程》，第61卷，第13期，第2296–2315页，2004年，doi: 10.1002/nme.1149.[64] A. Mendiburu、J. A. Lozano 和 J. Miguel-Alonso，“基于概率图模型的分布估计算法并行实现”，IEEE进化计算汇刊，第9卷，第4期，第406–423页，2005年，doi: 10.1109/TEVC.2005.850299.[65] A. Ruiz-Andino、L. Araujo、F. Sáenz 和 J. Ruz，“一种用于求解有限域约束优化问题的混合进化方法”，IEEE进化计算汇刊，第4卷，第4期，第353–372页，2000年，doi: 10.1109/4235.887235.[66] Q. Yang 等人，“一种用于大规模优化的自适应通信分布式群优化器”，IEEE 控制论汇刊，第50卷，第7期，第3393–3408页，2020年，doi: 10.1109/TCYB.2019.2904543。[67] S. M. Said 和 M. Nakamura，“用于函数优化的遗传算法与分布估计算法并行混合方法的异步策略”，载于第三届国际网络与计算会议，2012年，第420–428页。[68] M. A. Ismail，“并行遗传算法（PGAs）：基于消息传递接口的主从范式方法”，载于 E-Tech，2004年，第83–87页。[69] W. Yu 和 W. Zhang，“基于主从结构遗传算法的函数优化研究”，载于国际信号处理会议论文集，ICSP，2006年，第3卷，第0–3页，doi: 10.1109/ICOSP.2006.345926。[70] Y. Lorion、T. Bogon、I. J. Timm 和 O. Drobnik，“基于智能体的并行粒子群优化——APPSO”，载于 IEEE群体智能研讨会，SIS 2009，2009年，第52–59页，doi: 10.1109/SIS.2009.4937844。[71] G. Roy、H. Lee、J. L. Welch、Y. Zhao、V. Pandey 和 D. Thurston，“一种遗传算法分布式池架构”，载于 IEEE进化计算大会，CEC 2009，2009年，第1177–1184页，doi: 10.1109/CEC.2009.4983079.[72] D. Xin、D. Lixin 和 J. Liyuan，“ 基于分布估计算法的异步分布式并行基因表达式编程 ”，载于第四届国际自然计算会议，ICNC，2008年，第1卷，第433–437页，doi: 10.1109/ICNC.2008.497。[73] Z. H. Zhan 和 J. Zhang，“采用自适应异步迁移策略的并行粒子群优化”，载于国际并行处理算法与架构会议，2009年，第5574卷（LNCS），第2009期，第490–501页，doi: 10.1007/978-3-642-03095-6_47。[74] R. Tanese、J. H. Holland 和 Q. F. Stout，《用于函数优化的分布式遗传算法》。密歇根大学，1989年。[75] T. C. Belding，“重新审视分布式遗传算法”，载于 ICGA，1995年，第1–8页。[76] J. F. Romero 和 C. Cotta，“通过岛屿结构的去中心化粒子群进行优化”，《计算智能：理论、应用与软计算进展》，第33卷，第25–33页，2006年，doi: 10.1007/3-540-31182-3_3。[77] T. Ishimizu 和 K. Tagawa，“适用于多种网络拓扑的结构化差分进化”，《国际计算与通信期刊》，第4卷，第1期，第1–8页，2010年。[78] C. Zhang、J. Chen 和 B. Xin，“通过鲍德温式学习与拉马克式学习的协同作用实现分布式模因差分进化”，《应用软计算期刊》，第13卷，第5期，第2947–2959页，2013年，doi: 10.1016/j.asoc.2012.02.028。[79] G. Folino、C. Pizzuti 和 G. Spezzano，“ 并行遗传编程的可扩展细胞式实现 ”，IEEE 进化计算汇刊，第7卷，第1期，第37–53页，2003年，doi: 10.1109/TEVC.2002.806168。[80] B. Schönfisch 和 A. De Roos，“元胞自动机中的同步与异步更新”，《生物系统》，第51卷，第3期，第123–143页，1999年，doi: 10.1016/S0303-2647(99)00025-8。[81] S. Burak Akat 和 V. Gazi，“去中心化异步粒子群优化”，载于2008年 IEEE群体智能研讨会，SIS，2008年，第1–8页，doi: 10.1109/SIS.2008.4668304。[82] E. Alba 和 J. M. Troya，“细胞进化算法：评估比例的影响”，载于国际自然启发并行问题求解会议，2000年，第1917卷，第29–38页，doi: 10.1007/3-540-45356-3 3。[83] ¯A. J. Nebro、J. J. Durillo、F. Luna、B. Dorronsoro 和 E. Alba，“MOCell：一种用于多目标优化的细胞遗传算法”，《国际智能系统期刊》，第24卷，第7期，第726–746页，2009年，doi: 10.1002/int。[84] M. Giacobini、E. Alba 和 M. Tomassini，“异步细胞进化算法中的选择强度”，载于遗传与进化计算，2003年，第2723卷，第955–966页，doi: 10.1007/3-540-45105-6 107。[85] ¯M. Giacobini、M. Tomassini 和 A. Tettamanzi，“线性细胞进化算法的选择强度建模”，载于国际

人工进化会议（人工进化），2004年，第2936卷，第345–356页，doi：10.1007/978-3-540-24621-3。[86] M. Giacobini、E. Alba、A. Tettamanzi 和 M. Tomassini，“环面细胞进化算法的选择强度建模”，载于遗传与进化计算会议，2004年，第3102卷，第1138–1149页，doi：10.1007/978-3-540-24854-5_111。[87] M. Giacobini、M. Tomassini、A. G. B. Tettamanzi 和 E. Alba，“规则晶格上的细胞进化算法选择强度”，IEEE进化计算汇刊，第9卷，第5期，第489–505页，2005年，doi：10.1109/TEVC.2005.850298。[88] T. Nakashima、T. Ariyama 和 H. Ishibuchi，“结合多个细胞遗传算法以实现高效搜索”，载于亚太模拟进化与学习会议（SEAL），2022年，第712–716页。[89] T. Nakashima、T. Ariyama、T. Yoshida 和 H. Ishibuchi，“组合细胞遗传算法在函数优化问题中的性能评估”，载于IEEE机器人与自动化计算智能国际研讨会（CIRA），2003年，第1卷，第295–299页，doi：10.1109/CIRA.2003.1222105。[90] W. Zhong、J. Liu、M. Xue 和 L. Jiao，“用于全局数值优化的多智能体遗传算法”，IEEE系统、人类与控制论汇刊B辑：控制论，第34卷，第2期，第1128–1141页，2004年，doi：10.1109/TSMCB.2003.821456。[91] B. Zhao、C. Guo 和 Y. Cao，“一种基于多智能体的粒子群优化方法，用于无功功率最优调度”，IEEE Trans. Power Syst.，第20卷，第2期，第1070–1078页，2005年。[92] J. Liu、W. Zhong 和 L. Jiao，“用于约束满足问题的多智能体进化算法”，IEEE系统、人类与控制论汇刊B辑：控制论，第36卷，第1期，第54–73页，2006年，doi：10.1109/TSMCB.2005.852980。[93] J. Liu、W. Zhong 和 L. Jiao，“用于组合优化问题的多智能体进化算法”，IEEE系统、人类与控制论汇刊B辑：控制论，第40卷，第1期，第229–240页，2010年，doi：10.1109/TSMCB.2009.2025775。[94] J. L. J. Laredo、P. A. Castillo、A. M. Mora 和 J. J. Merelo，“可进化智能体：分布式进化计算的细粒度方法——迈向点对点计算前沿”，软计算，第12卷，第12期，第1145–1156页，2008年，doi：10.1007/s00500-008-0297-9。[95] W. Ding、C. T. Lin、S. Chen、X. Zhang 和 B. Hu，“基于多智能体共识MapReduce的协同进化量子粒子群优化属性约简方法，面向大数据应用”，神经计算，第272卷，153，2018年，136 10 1016 2017 06 059 页 – doi：. /j.neucom. . . . [96] Z. Zhang、Q. Han、Y. Li、Y. Wang 和 Y. Shi，“一种用于多目标优化的进化多智能体框架”，数学问题工程，第1–18页，2020年，doi：10.1155/2020/9147649。[97] J. Ji、L. Jiao、C. Yang 和 J. Liu，“一种用于检测复杂网络中社群的多智能体进化方法”，计算智能，第32卷，第4期，第587–614页，2016年，doi：10.1111/coin.12067。[98] G. Folino 和 G. Spezzano，“P-CAGE：点对点系统中的进化计算环境”，载于欧洲遗传编程会议，2006年，第3905卷，第341–350页，doi：10.1007/11729976_31。[99] G. Folino、C. Pizzuti 和 G. Spezzano，“采用基于聚类和剪枝的选择算法训练分布式遗传编程集成，用于模式分类”，IEEE进化计算汇刊，第12卷，468，2008年，4，458 10 1109 2007 906658 第期 页 – doi：. /TEVC. . . [100] F. Herrera、M. Lozano 和 C. Moraga，“分层分布式遗传算法”，《国际智能系统期刊》，第14卷，1121，1999年11，1099 第期 页 –。[101] T. Burczynski 和 W. Kus，“使用分布式并行进化算法进行结构优化”，载于并行处理与应用数学国际会议，2004年，第572–579页。[102] T. Burczyński、W. Kuś、A. Długosz 和 P. Orantek，“使用分布式进化算法进行优化与缺陷识别”，工程应用人工智能，第17卷，第4期，第337–344页，2004年，doi：10.1016/j.engappai.2004.04.007。[103] D. Lim、Y. S. Ong、Y. Jin、B. Sendhoff 和 B. S. Lee，“采用网格计算的高效分层并行遗传算法”，未来世代计算机系统，第23卷，第4期，第658–670页，2007年，doi：10.1016/j.future.2006.10.008。[104] S. Vavpotic 和 A. Dobnikar，“利用多处理器与多计算机系统加快进化算法执行”，电气工程学报，第69卷，第3–4期，第227–33页，2002年。[105] T. Bazlur Rashid、A.N.M. Choudhury，“协同协进化与MapReduce：大规模……的综述与新见解优化”，国际信息技术项目管理杂志，第12卷，62，2021年1，29期，第–页。[106] X. Du、Y. Ni、Z. Yao、R. Xiao和D. Xie，“基于MapReduce框架的高性能并行进化算法模型”，国际计算应用技术杂志，第46卷，第3期，第290–295页，2013年，doi：10.1504/IJCAT.2013.052807。[107] A. Haghofer、S. Dorl、A. Oszwald、J. Breuss、J. Jacak和S. M. Winkler，“显微图像细胞检测图像处理的进化优化”，软计算，第24卷，第23期，第17847–17862页，2020年，doi：10.1007/s00500-020-05033-0。[108] G. A. Laguna-Sánchez、M. Olguín-Carbajal、N. Cruz-Cortés、R. Barrón-Fernández和R. Cadena Martínez，“GPU上的差分进化算法并行实现”，理论与应用信息技术杂志，第86卷，第2期，第184–195页，2016年。[109] Y. Huang、L. Feng、A. K. Qin、S. Member、M. Chen和K. C. Tan，“迈向大规模进化多任务：一种基于GPU的范式”，*IEEE*进化计算汇刊，第26卷，第3期，第585–598页，2022年。[110] W. R. M. U. K. Wickramasinghe、M. Van Steen和A. E. Eiben，“具有自适应自主选择的点对点进化算法”，*GECCO 2007*遗传与进化计算会议论文集，第1460–1467页，2007年，doi：10.1145/1276958.1277225。[111] J. L. J. Laredo、A. E. Eiben、M. Van Steen和J. J. Merelo，“可扩展的点对点进化算法：EvAg”，遗传编程与可进化机器，第11卷，第2期，第227–246页，2010年，doi：10.1007/s10710-009-9096-z。[112] J. L. J. Laredo、A. E. Eiben、M. Van Steen、P. A. Castillo、A. M. Mora和J. J. Merelo，“P2P进化算法：解决困难优化问题中大规模实例的合适方法”，载于欧洲并行处理会议，2008年，第5168卷，第622–631页，doi：10.1007/978-3-540-85451-7 66。[113] J. L. J. Laredo、P. A. Castillo、A. M. Mora、M. Merelo和C. Fernandes，“点对点进化算法对节点 churn 的弹性”，国际高性能系统架构杂志，第1卷，第4期，第260–268页，2008年，doi：10.1504/IJHPSA.2008.024210。[114] H. Malmir、F. Farokhi和R. Sabbaghi-Nadooshan，“云计算应用中采用进化算法优化数据挖掘”，载于第三届计算机与知识工程国际会议（*ICCKE 2013*）论文集，2013年，第Iccke期，第343–347页，doi：10.1109/ICCKE.2013.6682821。[115] J. Yu、H. Li和D. Liu，“云计算环境下用于医疗数据聚类和特征提取的改进免疫进化算法”，医疗工程杂志，2020年，第11卷，2020年，第1–10页，doi：10.1155/2020/1051394。[116] A. S. Pillai、K. Singh、V. Saravanan、A. Anpalagan、I. Woungang和L. Barolli，“一种基于遗传算法的方法，用于优化多处理器系统的能耗和性能”，软计算，第22卷，第10期，第3271–3285页，2018年，doi：10.1007/s00500-017-2789-y。[117] X. Bi和C. Zhang，“用于多处理器选择的分布式进化算法”，融合信息技术杂志，第8卷，第5期，第100–8页，2013年。[118] Z. Handzel、M. Gajer和D. Jamroz，“采用进化计算的异质多处理器系统中周期性任务的调度与分配。第1部分：调度算法的特性”，*Elektronika*，第56卷，第3期，第91–2页，2015年。[119] Z. Handzel、M. Gajer和D. Jamroz，“采用进化计算的异质多处理器系统中周期性任务的调度与分配。第2部分——进化计算结果”，*Elektronika*，第56卷，第4期，第60–2页，2015年。[120] Y. C. Kim和Y. S. Hong，“一种用于多处理器系统任务分配的遗传算法”，韩国信息科学学会杂志，第20卷，第1期，第43–51页，1993年。[121] Y. Zhou、J. Zhang和Y. Wang，“用于多处理器调度问题的(1+1)进化算法性能分析”，*Algorithmica*，第73卷，第1期，第21–41页，2015年，doi：10.1007/s00453-014-9898-0。[122] M. Anandaraj、K. Selvaraj、P. Ganeshkumar、K. Rajkumar和S. Sriram，“基于遗传算法的网络编码P2P网络资源最小化”，电路、系统与计算杂志，第30卷，第6期，第1–20页，2021年，doi：10.1142/S0218126621500924。[123] F. Zuo和W. Zhang，“一种基于进化博弈的机制，用于在自私节点之间路由P2P网络流量”，网络杂志，第9卷，第1期，第10–17页，2014年，doi：10.4304/jnw.9.1.10-17。

[124] M. Dai等，“移动云计算的进化研究”，《神经计算与应用》，第28卷，第9期，第2735–2744页，2017年，doi：10.1007/s00521-016-2217-8。[125] X. Ye、S. Liu、Y. Yin和Y. Jin，“基于改进膝点驱动进化算法的面向用户的多目标云工作流调度”，《知识型系统》，第135卷，124，2017年，第113–页，doi：. /j.knosys. . . . [126] Z. Zhu、G. Zhang、M. Li和X. Liu，“云环境中的进化式多目标工作流调度”，《IEEE 并行与分布式系统汇刊》，第27卷，第5期，第1344–1357页，2016年，doi：10.1109/TPDS.2015.2446459。[127] D. Teijeiro、X. C. Pardo、P. González、J. R. Banga和R. Doallo，“迈向基于云的并行元启发式算法：采用差分进化和Spark进行计算生物学案例研究”，《国际高性能计算应用杂志》，第32卷，第5期，第693–705页，2018年，doi：10.1177/1094342016679011。[128] Z. Fan、F. Qiu、A. Kaufman和S. Yoakum-Stover，“用于高性能计算的GPU集群”，载于《ACM/IEEE SC 2004会议：连接社群》，2004年，第1期，第1–12页，doi：10.1109/SC.2004.26。[129] K. Wang和Z. Shen，“用于生成日常活动计划的基于GPU的并行遗传算法”，《IEEE 智能交通系统汇刊》，第13卷，第3期，第1474–1480页，2012年，doi：10.1109/tits.2012.2205147。[130] J. Qin、W. Fu、H. Gao和W. X. Zheng，“基于多智能体一致性理论的传感器网络分布式k均值算法和模糊c均值算法”，《IEEE 控制论汇刊》，第47卷，第3期，第772–783页，2017年，doi：10.1109/TCYB.2016.2526683。[131] J. Xu、Y. Jin、W. Du和S. Gu，“联邦数据驱动进化算法”，《知识型系统》，第233卷，107532，2021年，doi：10.1016/j.knosys.2021.107532。[132] W. Ai、W. Chen和J. Xie，“一种用于网络上种群分布式优化的通用框架”，《信息科学》（纽约），第418–419卷，第16期，第136–152页，2017年，doi：10.1016/j.ins.2017.08.010。[133] K. Utkarsh、A. Trivedi、D. Srinivasan和T. Reindl，“一种基于共识的分布式计算智能技术，用于智能配电网实时最优控制”，《IEEE 计算智能新兴主题汇刊》，第1卷，第1期，第51–60页，2017年，doi：10.1109/TETCI.2016.2635130。[134] Q. Yang、Y. Liu、T. Chen和Y. Tong，“联邦机器学习：概念与应用”，《ACM 智能系统与技术汇刊》，第10卷，第2期，第1–19页，2019年，doi：10.1145/3298981。[135] D. Delahaye、S. Chaimatanan和M. Mongeau，“模拟退火：从基础到应用”，载于《运筹学与管理科学国际系列丛书》，第272卷，2019年，第1–35页。[136] R. Chelouah和P. Siarry，“应用于全局优化的禁忌搜索”，《欧洲运筹学杂志》，第123卷，第2期，第256–270页，2000年，doi：10.1016/S0377-2217(99)00255-6。[137] I. A. Basheer和M. Hajmeer，“人工神经网络：基础、计算、设计与应用”，《微生物学方法杂志》，第43卷，第1期，第3–31页，2000年，doi：10.12989/cac.2013.11.3.237。[138] C. Xia、J. Hua、W. Tong和S. Zhong，“保障局部差分隐私的分布式K均值聚类”，《计算机安全》，第90卷，第1–11页，2020年，doi：10.1016/j.cose.2019.101699。[139] H. Ishibuchi、M. Yamane和Y. Nojima，“针对具有不同缺失属性和隐私策略的多个数据集进行学习：并行分布式模糊遗传机器学习方法”，载于《IEEE国际大数据会议》，2013年，场景2，第63–70页，doi：10.1109/BigData.2013.6691735。[140] G. V. Oliveira、F. P. Coutinho、R. J. G. B. Campello和M. C. Naldi，“通过分布式可扩展元启发式算法改进k均值”，《神经计算》，第246卷，第45–57页，2017年，doi：10.1016/j.neucom.2016.07.074。[141] M. Hajeer、D. Dasgupta、A. Semenov和J. Veijalainen，“用于数据聚类和建模的分布式进化方法”，载于《IEEE计算智能与数据挖掘研讨会》，2015年，第142–148页，doi：10.1109/CIDM.2014.7008660。[142] M. Hajeer和D. Dasgupta，“利用数据感知型HDFS和进化聚类技术处理大数据”，《IEEE 大数据汇刊》，第5卷，第2期，第134–147页，2019年，doi：10.1109/tbdata.2017.2782785。[143] J. Xu、Y. Jin和W. Du，“用于昂贵的多目标/多目标优化的联邦数据驱动进化算法”，《复杂与智能系统》，第7卷，第6期，第3093–3109页，2021年，doi：10.1007/s40747-021-00506-7。[144] K. Utkarsh、D. Srinivasan、T. Reindl和A. Trivedi，“利用基于分布式计算智能的控制提供电网辅助服务

配电网络中的可再生能源与储能系统，” *IEEE*进化计算大会，第1412–1419页，2017年，doi: 10.1109/CEC.2017.7969469。[145] K. Utkarsh、D. Srinivasan、A. Trivedi、W. Zhang和T. Reindl，“智能微电网网络的分布式模型预测实时最优运行”，*IEEE Trans. Smart Grid*，第10卷，第3期，第2833–2845页，2019年，doi: 10.1109/TSG.2018.2810897。[146] S. Strasser、J. Sheppard、N. Fortier和R. Goodman，“因式分解进化算法”，*IEEE*进化计算汇刊，第21卷，第2期，第281–293页，2017年，doi: 10.1109/TEVC.2016.2601922。[147] N. Fortier、J. Sheppard和S. Strasser，“利用分布式重叠群体智能进行贝叶斯网络中的溯因推理”，软计算，第19卷，第4期，第981–1001页，2015年，doi: 10.1007/s00500-014- 1310-0。[148] N. Fortier、J. W. Sheppard和K. G. Pillai，“DOSI：利用带有局部信用分配的重叠群体智能训练人工神经网络”，载于第*6*届软计算与智能系统国际会议暨第*13*届先进智能系统国际研讨会（*SCIS/ISIS 2012*），2012年，第1420–1425页，doi: 10.1109/SCIS-ISIS.2012.6505078。[149] A. Oyama、T. Kohira、H. Kemmotsu、T. Tatsukawa和T. Watanabe，“利用京计算机对多种汽车车型进行同步结构设计优化”，*2017 IEEE*计算智能研讨会系列（*SSCI 2017*）会议录，第2018年1月卷，第1–4页，2018年，doi: 10.1109/SSCI.2017.8285350。[150] M. Yokokawa、F. Shoji、A. Uno、M. Kurokawa和T. Watanabe，“京计算机：日本下一代超级计算机开发项目”，国际低功耗电子设计研讨会会议录，第371–372页，2011年，doi: 10.1109/ISLPED.2011.5993668。[151] Y. Jin、H. Wang、T. Chugh、D. Guo和K. Miettinen，“数据驱动进化优化：综述与案例研究”，*IEEE*进化计算汇刊，第23卷，第3期，第442–458页，2019年，doi: 10.1109/TEVC.2018.2869001。[152] X. Yin、W. Zhang、Z. Jiang和L. Pan，“基于进化优化的海上风电场数据驱动多目标预测控制”，可再生能源，2020年，doi: 10.1016/j.renene.2020.05.015。[153] F.-F. Wei、W.-N. Chen、Q. Li、S.-W. Jeon和J. Zhang，“采用按需评估的分布式高成本进化约束优化”，*IEEE*进化计算汇刊，第1–1页，2022年，doi: 10.1109/tevc.2022.3177936。[154] R. Cheng和Y. Jin，“一种用于大规模优化的竞争群体优化器”，*IEEE* 控制论汇刊，第45卷，第2期，第191–204页，2015年，doi: 10.1109/TCYB.2014.2322602。[155] Q. Yang、W. N. Chen、J. Da Deng、Y. Li、T. Gu和J. Zhang，“一种用于大规模优化的基于层级学习的群体优化器”，*IEEE*进化计算汇刊，第22卷，第4期，第578–594页，2018年，doi: 10.1109/TEVC.2017.2743016。[156] B. Li、J. Gu和W. Jiang，“人工智能（AI）芯片技术综述”，载于*2019*年机器学习、大数据与商业智能国际会议，2019年，第114–117页，doi: 10.1109/MLBDBI48998.2019.00028。[157] O. Krestinskaya、K. N. Salama和A. P. James，“利用遗传搜索自动化模拟人工智能芯片设计”，先进智能系统，第2卷，第8期，第1–12页，2020年，doi: 10.1002/aisy.202000075。[158] C. Erbas、S. Cerav-Erbas和A. D. Pimentel，“面向多处理器片上系统设计中应用映射问题的多目标优化与进化算法”，*IEEE*进化计算汇刊，第10卷，第3期，第358–374页，2006年，doi: 10.1109/TEVC.2005.860766。[159] G. Ascia、V. Catania和M. Palesi，“一种用于片上网络映射问题的进化方法”，载于*2005*年*IEEE*进化计算大会，2005年，第1卷，第112–119页，doi: 10.1109/cec.2005.1554674。[160] M. Ribeiro、K. Grolinger和M. A. M. Capretz，“MLaaS：机器学习即服务”，载于*IEEE*第*14*届机器学习与应用国际会议，2016年，第896–902页，doi: 10.1109/ICMLA.2015.152。[161] Y. Tian、R. Cheng、X. Zhang和Y. Jin，“PlatEMO：用于进化多目标优化的MATLAB平台 [教育论坛],” *IEEE*《计算智能杂志》，第12卷，第4期，第73–87页，2017年，doi: 10.1109/MCI.2017.2742868。[162] S. D. Gobble、M. Welsh、E. A. Brewer和D. Culler，“MultiSpace：一种基础设施服务进化平台”，载于*1999*年*USENIX*年度技术会议，1999年，第1–29页。

[163] X. Guo、W. Chen、F. Wei、W. Mao、X. Hu和J. Zhang，“面向分布式数据驱动优化问题的边缘云协同进化算法”，*IEEE* 控制论汇刊，第1–14页，2022年。[164] K. C. Tan、L. Feng和M. Jiang，“进化迁移优化——进化计算研究的新前沿”，*IEEE*《计算智能杂志》，第16卷，第1期，第22–33页，2021年，doi: 10.1109/MCI.2020.3039066。[165] B. Zhao、X. Liu、S. Member、A. Song和W. Chen，“PRIMPSO：隐私保护多智能体粒子群优化算法”，*IEEE* 控制论汇刊，第1–14页，2022年。[166] J. Du、C. Jiang、K. C. Chen、Y. Ren和H. V. Poor，“面向社交网络隐私保护的社区结构进化博弈”，*IEEE*《信息取证与安全汇刊》，第13卷，第3期，第574–589页，2018年，doi: 10.1109/TIFS.2017.2758756。[167] E. Balestrieri、P. Daponte、L. De Vito和F. Lamonaca，“无人系统的传感器与测量：综述”，传感器，第21卷，第4期，第1–27页，2021年，doi: 10.3390/s21041518。[168] A. Byrski、R. Drezewski、L. Siwik和M. Kisiel-Dorohinicki，“进化多智能体系统”，《知识工程评论》，第30卷，第2期，第171–186页，2015年，doi: 10.1017/S0269888914000289。[169] T. Van Luong、N. Melab和E. G. Talbi，“基于GPU的进化算法岛屿模型”，载于第*12*届年度遗传与进化计算大会论文集，2010年，7月，第1089–1096页，doi: 10.1145/1830483.1830685。[170] P. Pospichal、J. Jaros和J. Schwarz，“基于CUDA架构的并行遗传算法”，载于欧洲进化计算应用会议，2010年，第6024卷，第442–451页，doi: 10.1007/bf02900894。



Wei-Neng Chen（S’07–M’12–SM’17）于2006年和2012年分别获得中国广州中山大学计算机科学学士学位和博士学位。自2016年起，他担任广州华南理工大学计算机科学与工程学院正教授。他已合著100余篇国际期刊和会议论文。

论文，其中包括发表在 IEEE Transactions系列期刊上的50多篇论文。他目前的研究兴趣包括计算智能、群体智能、网络科学及其应用。

陈博士于2016年获得IEEE计算智能学会（CIS）杰出博士论文奖，并于2016年获得国家自然科学基金优秀青年科学基金项目资助。他目前担任IEEE广州分会副主席，同时也是IEEECIS新兴主题任务组的委员。他还担任IEEE神经网络与学习系统汇刊和复杂与智能系统的副编辑。



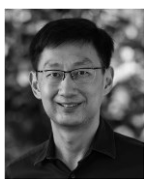
Feng-Feng Wei于2019年在中国广州的华南理工大学获得计算机科学学士学位，目前正在该校攻读博士学位。她目前的研究兴趣包括进化计算算法、多/多目标优化、约束优化，以及这些领域在昂贵优化和分布式优化中的应用。

现实世界问题。



Tian-Fang Zhao（S’17-M’21）于2021年在中国广州的华南理工大学获得计算机科学博士学位，并于2017年在中国大连的大连理工大学获得计算机科学硕士学位。

她目前是中国广州暨南大学的研究员。目前的研究兴趣包括社交媒体和数据挖掘, 复杂网络, 群体智能。



Kay Chen Tan（IEEE 会士）于1994年和1997年分别获得英国格拉斯哥大学电子与电气工程专业工学学士学位（一等荣誉）和博士学位。

他目前是香港理工大学计算学系的讲座教授，中国香港。他已发表 200 多篇同行评审论文和 6 本书，并持有一项关于表面缺陷检测的美国专利。

Tan教授现任美国IEEE计算智能学会出版事务副主席。他曾于2015–2020年担任《IEEE进化计算汇刊》主编，并于2010–2013年担任《IEEE计算智能杂志》主编，目前担任十余种期刊的编委。他现任IEEE杰出演讲者计划（DLP）讲者，并担任Springer机器学习丛书《基础、方法论与应用》的共同主编。



Jun Zhang（M’02–SM’08–F’17）于2002年获得香港城市大学电气工程博士学位。目前，他是汉阳大学电气工程系的访问教授。他的研究兴趣包括计算智能、云计算、数据挖掘和电力电子电路。他已在其研究领域发表了200余篇技术论文。张博士于2011年获得国家自然科学基金委员会颁发的国家杰出青年科学基金，并于2009年获得中国教育部颁发的自然科学研究一等奖。他现任《IEEE进化计算汇刊》和《IEEE控制论汇刊》的副编辑。