CEC：基于众包的分布式优化进化计算

Feng-Feng Wei、Wei-Neng Chen，高级会员，*IEEE*，Xiao-Qi Guo、Bowen Zhao，会员，*IEEE*，Sang-Woon Jeon，会员，*IEEE*，以及 Jun Zhang，会士，*IEEE*

摘要——众包是一种新兴的计算范式，它利用众包人群的智慧有效解决复杂问题。除了收集和处理数据，众包人群也有很大的需求来执行优化。受此启发，本文旨在将众包引入进化计算（进化计算），提出一种用于分布式优化的基于众包的进化计算（CEC）范式。进化计算有助于解决众包优化任务，而众包反过来可以打破进化计算在大规模分布式优化中的空间限制。因此，本文首先介绍基于众包的分布式优化范式。随后，详细阐述 CEC。CEC 基于服务器和一组工作者执行优化，其中服务器将大型任务派发给工作者。工作者通过进化计算优化器搜索有前景的解，并与相连邻居协作。为消除工作者行为和设备异质性带来的不确定性，服务器采用竞争排名和不确定性检测策略来指导工作者协作。为展示 CEC 的良好性能，本文实现了一种基于众包的群体优化器作为示例，并进行了大量实验。在基准函数和一个分布式聚类优化问题上的比较结果展现了 CEC 的潜在应用价值。

索引 术语—众包、分布式优化、进化计算、粒子群优化

u

1 引言

C众包是一种随着社交媒体和互联网发展而兴起的技术，旨在从众包人群中获取工作成果、信息或意见[1]。具体而言，如图 1 所示，众包的一般范式包括请求者、众包平台和众包人群。众包通常分为四个步骤。1）请求者向众包平台发布任务。2）众包平台将任务外包给众包人群，即工作者。3）每位工作者执行任务，并通过互联网或社交媒体向众包平台反馈信息。4）众包平台处理收集到的信息，并将结果发送给请求者。众包利用众包人群的集体智慧有效解决复杂问题，能够获得多样性较高的解决方案，并在较短时间内完成大型任务。过去几年中，众包发展出了许多变体，例如空间众包[2], [3], [4], 团队众包[5]，并成功应用于许多信息收集任务[6], [7], [8], ，成为商业领域[9] 和学术界[10]都广泛采用的一种技术。自其出现以来。

然而，传统的空间型或通用型众包主要用于跨空间、大规模的信息感知。随着物联网的发展 [11] 以及数据的爆发式增长 [12], ，边缘智能 [13] 和群体计算 [14]作为颠覆性技术脱颖而出。除了收集信息，让众包人群处理数据并做出决策也极具吸引力且十分必要。例如，在电力系统中，请求者会在众包平台上发布一项任务，要求优化分布式能源资源的协调。众包平台将该任务外包给众包人群，由他们根据收集到的电力生产和消耗信息，优化能源资源的协调 [15]。在联邦学习中，请求者会发布一项优化全局模型的任务。众包人群从平台接收任务后，需要收集各自空间分散的数据，并利用这些数据训练本地模型。平台通过整合本地模型来优化全局模型 [16]。在无线传感器网络中，分布式聚类优化是一项常见需求，其目标是根据收集到的所有数据，获得全局优化的聚类结果。接到任务后，众包人群中的每位工作者不仅需要感知本地数据，还需要结合一些全局信息来优化本地聚类 [17]。鉴于跨空间和大规模感知的特点，我们将这种模式称为基于众包的分布式优化。顾名思义，在基于众包的分布式优化中，众包人群不仅需要收集信息，还需要利用邻居的信息，甚至

• *Feng-Feng Wei*、*Wei-Neng Chen* 和 *Xiao-Qi Guo* 任职于中国广州 *510006* 华南理工大学计算机科学与工程学院、亚热带建筑科学国家重点实验室、华南理工大学粤港大数据与计算智能联合创新平台，以及中国广州 *510006* 琶洲实验室。电子邮箱：*fengfeng_scut@163.com*、*cwn-raul634@aliyun.com*、*kallyqi@outlook.com*。

• *Bowen Zhao* 就职于中国广州*510006*的广州理工学院西安电子科技大学。电子邮箱：*bwinzhao@gmail.com*。

• *Sang-Woon Jeon* 和 *Jun Zhang* 就职于韩国安山*15588*的汉阳大学电子与通信工程系。电子邮箱：*sangwoonjeon@hanyang.ac.kr*、*jun-zhang@ieee.org*。

xxxx-xxxx/0x/ $xx.00 © 200x IEEE 由 IEEE 计算机学会出版

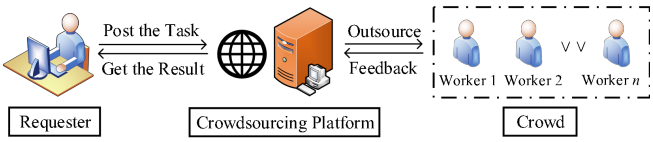


图 1. 众包示意图。

平台或请求者的信息来开展优化。随着移动群智感知与计算的兴起 [14], , 工业界和学术界的更多应用本质上都属于基于众包的分布式优化。

进化计算（EC），包括进化算法（EAs） [18], [19] [20], 和 [21], 群体智能（SI），是复杂优化领域蓬勃发展的研究方向。一方面，进化计算算法是一类无导数优化方法，在处理非凸、大规模、黑箱、昂贵等特征的复杂优化问题时具有强大的能力和鲁棒性[22], [23], [24]。另一方面，进化计算具有内在的并行性，因此能够借助众包人群的力量，在分布式优化方面展现出巨大潜力。一些研究人员已经探索将进化计算用于分布式优化。例如，Xu 等人 [25], [26] 提出了一种联邦数据驱动进化计算算法，利用由分布式数据训练的分布式局部模型来训练全局模型。Liu 等人 [27] 设计了一种分布式代理辅助进化计算算法，用于高维特征选择优化。Wei 等人 [28] 基于分布式数据和评估，开发了一种采用按需评估的分布式进化计算算法，用于具有众多约束的昂贵优化问题。Guo 等人 [29] 构建了一种边缘云进化计算框架来处理分布式数据。尽管已有许多工作致力于将进化计算用于分布式优化，但仍存在一些局限性。

首先，进化计算在优化大规模问题时会受到维数灾难的困扰。维数灾难是进化计算中一个众所周知的挑战：在求解大规模问题时，进化计算的效率和优化性能会下降 [30]。尽管人们已通过多处理器或超级计算技术做出诸多努力来缓解这一问题 [31], , 但在将进化计算应用于超大规模优化时，这仍是一个亟待解决的主要问题。

第二个局限是，大多数现有的用于分布式优化的进化计算在数据有限时无法取得良好表现。在文献中，大多数关于进化计算的研究都聚焦于集中式优化，在这种场景下，全局信息通常由所有处理器共享 [31]。但在分布式优化中，信息由本地工作者持有，他们需要通过特定的协作机制相互通信。尽管少数研究人员尝试将进化计算应用于数据分布式优化问题 [25], [26], [28], [29], , 但他们仅考虑了工作者数量较少的问题。如何在数据有限且工作者数量众多的情况下，使大规模问题取得有效性能，仍有待研究。

第三个局限是，若没有准确的适应度评估，进化计算就难以进化。进化计算基于适者生存原则进行迭代进化，其中适应度评估是明确已知的 [18], [19], 。然 [20], [21] 而，在分布式优化中，设备异质性和工作者行为可能会带来不确定性，并降低数据质量 [32]。我们将这类不确定性称为环境感知不确定性。环境感知不确定性会对适应度评估产生重大影响，可能误导进化方向并降低算法性能。在分布式环境中设计进化计算时，值得研究如何减轻环境感知不确定性带来的负面影响。

幸运的是，将众包范式引入进化计算，有望克服上述局限。进化计算与众包有相似之处：二者都利用大量个体协作完成全局任务。不同的是，进化计算旨在进行优化，而众包主要用于收集和处理数据。将众包与进化计算结合具有一些优势。首先，将优化任务派发给众包人群，以继承众包范式，有助于进化计算高效、有效地解决问题。众包人群中的许多工作者协作处理原本的大型任务，这有助于缓解维数灾难。其次，众包人群针对全局关注区域获取并处理的数据，可以为进化计算提供数据支持。在将进化计算应用于大规模、复杂的分布式优化时，这有助于克服数据缺失的局限。第三，庞大的众包人群能够通过不确定性检测，减轻环境不确定性的影响。众包中的人群通常规模庞大。如果存在对环境不确定性具有较强感知能力的工作者，就可以检测并将其从进化群体中移除，而不影响整个进化过程。因此，数据和评估会更加准确，整体算法性能也能得到提升。基于以上考虑，我们打算将进化计算与众包结合，用于基于众包的分布式优化。

因此，本文引入了基于众包的分布式优化，并针对该问题提出了基于众包的进化计算（CEC）。本文的贡献如下。

首先，介绍基于众包的分布式优化范式。与众包类似，其中有三个角色：请求者、云服务器和工作者。请求者将优化任务发布到云服务器。云服务器将其派发给一组工作者。每个工作者代表一个个体，而所有工作者共同组成一个完整的进化种群。不同工作者的评估具有不同程度的环境感知不确定性。工作者可以与相连邻居和云服务器进行通信。

其次，针对上述分布式优化，提出了基于众包的进化计算（CEC）。它继承了众包范式，并发挥了进化计算的优势。工作者通过进化计算优化器搜索有希望的解，并与相连邻居进行协作。此外，他们还可以与云服务器通信，发送局部信息并获取全局信息，以进行进化。总而言之，该问题通过云服务器的引导和工作者之间的协作得到优化。

第三，提出了一种不确定性检测策略来

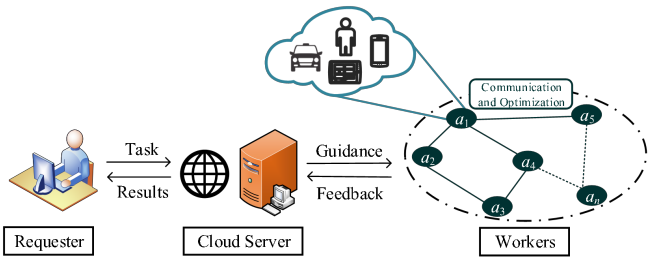


图 2. 基于众包的分布式优化示意图，其中 a_i 表示i^{th} 工作者代理。

检测不确定性。具体而言，互联的工作者可以交换带有环境感知不确定性的候选解适应度值。他们将收到的适应度值与自身的适应度值进行比较，并将比较结果发送到云服务器。云服务器通过竞争排名策略对所有工作者进行排名。它根据排名变化检测不确定性较大的工作者。此外，排名结果还用于指导工作者的进化。

例如，基于众包的群体优化器 CLLSO 已被实现，用以展示 CEC 的应用前景。在 CLLSO 中，采用的进化计算算法是层级学习群体优化器（LLSO），它擅长处理复杂的大规模优化问题 [33]。在基准函数和分布式聚类优化问题上开展的大量实验表明，CEC 在基于众包的分布式优化方面表现良好。

本文其余部分安排如下。第 2 节介绍基于众包的分布式优化范式。第 3 节详细阐述所提出的 CEC。第 4 节在基准测试套件和分布式聚类优化问题上开展大量实验，以研究 CEC 的性能及其应用前景。第 5 节总结全文。

2 基于众包的分布式优化优化

基于众包的分布式优化源于众包中的优化需求，其中的优化涉及大量空间分布数据，需要结合众包人群进行优化。因此，众包人群不仅需要收集信息，还需要通过多个循环进行优化和决策，以寻找最优解。在不失一般性的前提下，最小化优化问题的数学公式表述如下：

$$\min F = f(X,\Omega) \tag{1}$$

其中，F 是待优化的目标函数，f 是纯目标函数评估，X 是决策变量向量，Ω 表示环境数据集。

顾名思义，基于众包的分布式优化继承了众包范式。具体而言，基于众包的分布式优化范式如图 2 所示，包括请求者 R、云服务器 S（在众包中称为众包平台）以及一组工作者 W（也称为众包人群）。一些

表格 1 符号及模型与过程之间的对应关系

符号说明	描述	职责
R	请求者	1) 帖子 P
S	云服务器	1) 将P 外包给W 2) W的引导搜索 3) 从 G 到 R的反馈
W	工作节点集合	1) 进化 W 以进行搜索 2）与 S 通信
A	工作智能体	/
E	边矩阵	
P	task	
X	决策变量	
Ω	环境数据	
D	搜索域	
F	目标函数	
$f(X_{best})$	全局最优适应度	
X_{best}	全局最优变量	
M	比较结果矩阵	

广泛使用的符号说明，以及过程与数学公式表述之间的映射关系如表 1 所示。下面将对该范式进行详细描述。

2.1 请求者

请求者 R 是任务发布者。在基于众包的优化开始时，R 会向云服务器发布一个优化任务，记为 $P = \langle X, D, F\rangle$，由云服务器完成，其中

- $X = \{x_1, x_2, \dots, x_m\}$ 是决策变量向量，m 是维数。
- $D = \{D_1, D_2, \dots, D_m\} = \{[lb_1, ub_1], \dots, [lb_m, ub_m]\}$ 是每个维度定义域的集合。lb ub $i =_i$ 和 $_i$ $(,,\dots,)$ 分别表示第 $_{ith}$ 维的下界和上界。
- $F = f(X, \Omega)$ 是待优化的目标函数，它可能有数学公式表述，也可能没有。它可能具有复杂特性，例如非凸、大规模、受约束、多目标、多模态、计算代价高等。

请求者的需求是要求云服务器反馈已找到的最佳目标函数值 $f(X_{best})$ 及相应的决策变量向量 X_{best}。

2.2 云服务器

云服务器 S 是连接请求者与工作者的机构。它主要有两项功能。

1) 任务外包。云服务器 S 从请求者处收到 P 后，会将其外包给工作者，因为完成该任务需要大量空间分布式信息，或者该任务过于庞大、复杂，云服务器无法完成。

2) 搜索指导。作为协调工作者的机构，S 具有更高的安全性，并受到所有工作者的信任。它参与所有工作者的优化过程。工作者可以将其优化信息发送给 S。收到工作者的信息后，S 会对其进行处理，并将可用的全局信息反馈给工作者，以进一步指导其搜索方向。

优化结束时，云服务器处理所有优化后的数据和反馈 $f(X_{best})$，并将其 X_{best} 给 R。

2.3 工作者

工作者 $W = \langle A, E \rangle$ 是主要的任务执行者，并具有通信拓扑，其中

- $A = a_1, a_2, \ldots, a_n$是一组工作智能体，$n$ 是工作者总数。每位工作者对应一个个体。
- $E = \{e_{ij}\}$是众包网络的边矩阵，其中$e_{ij} = 1$ 表示工作者a_i与工作者 a_j 相连，而$e_{ij} = 0$ 表示工作者a_i与工作者a_j 不是邻居。

工作者可以是车辆、使用手机的人或笔记本电脑，这些设备的计算能力较弱且存储空间有限。因此，在整个搜索空间中，每位工作者对应一个个体。除数据收集和环境感知外，他们还具备少量的评估和优化能力。由于工作者行为的不确定性以及设备的异质性，工作者具有不同程度的环境感知不确定性。对i^{th} 工作者的每次评估都会受到不确定性r_i的影响，其中$F(X_i) = f(X_i) + r_i$。工作者从云服务器接收任务后，主要通过通信和优化这两种操作协同对其进行优化。

1) 通信。由于局部信息有限，工作者需要与其相连邻居和云服务器通信，以交换和更新信息。需要注意的是，工作者对原始数据很敏感。与相连邻居通信时，他们只发送其环境感知不确定性适应度值 $F(X_i)$。与 S 通信时，他们只发送与邻居的比较结果。对于每个工作者，其决策变量向量对其他工作者保密，除非更高的安全要求其发送。

y 机构，例如 S。

2) 优化

为完成优化任务，工作者需要执行特定的优化方法，这些方法取决于 P 的特征。在优化过程中，工作者与邻居进行通信以交换信息。除了局部信息外，工作者还与 S 通信，以获取部分全局信息，从而引导搜索方向。

具体而言，工作者具有移动性，因此其连接的邻居会随时间随机变化。

2.4 特征

基于众包的分布式优化与通用分布式优化有很大不同。以（1）为例，对于通用分布式优化，Ω 已知，只需基于Ω 进行优化，以求解 $\min f$ 。评估过程是明确给定的，大多数关于通用分布式优化的研究主要关注优化机制。此类问题不涉及数据收集，通常也不考虑环境感知不确定性、时变拓扑和数据敏感性等特征。

不同的是，在基于众包的分布式优化中，Ω 是未知的。它需要同时开展众包感知以收集数据，并使种群进化以进行进化。在基于众包的分布式优化中，每个工作者的评估取决于数据和环境可能受到环境感知型不确定性、不断变化的拓扑和所收集数据的影响。因此，在设计基于众包的分布式优化算法时，应认真考虑环境感知型不确定性、弱中心化、随时间变化的拓扑和数据敏感性等特征。

根据上述描述，基于众包的分布式优化具有以下特征。

i) **不确定性适应度**。每位工作者的评估都具有环境感知不确定性。这是众包感知所带来的一种影响。由于基于众包的分布式优化同时进行数据感知和优化，因此设计算法时应考虑这一点。

ii) **弱中心化**。工作者可以与云服务器通信，发送部分局部信息并接收部分全局信息。云服务器具有较高的可信度，但无法从全局视角获取所有局部信息。因此，基于众包的分布式优化是弱中心化的。

iii) **时变拓扑**。工作者在随机移-动，通信区域内的邻居也在随机变化。因此，连通边也会随机时变。

iv)**数据敏感性**。工作者重视数据保护。除非云服务器要求他们为进化共享，否则他们自己的决策变量向量会被保密。

3 CEC

3.1 CEC 框架

CEC 的框架如图 3 所示，伪代码见算法1。CEC 包含两种角色：云服务器 S W P 和 S W W 工作智能体 。收到 后， 将其派发给 。随后， 便在搜索空间中进行初始化。它们与各自相连邻居交流不确定性适应度值。之后，不确定性适应度的比较结果会发送给云服务器，由云服务器对比较结果进行竞争排序。接下来，S 进行不确定性检测，以找出不可靠且不确定性水平较高的个体。如果尚未满足终止条件，S 便利用排名和检测结果来引导进化并更新W。否则，将排名最高的候选解提交给R 作为优化后的解。详细过程介绍如下。

步骤 1) 任务调度。S 将优化任务 P （目标函数为 $f(X)$、搜索空间为 $D = \{D_1, D_2, \ldots, D_m\} = \{[lb_1, ub_1], \ldots, [lb_m, ub_m]\}$）派发给 n 个工作智能体。

步骤 2) 初始化。每个工作者代理在 D 中随机生成一个个体 X_i 。为模拟环境感知的不确定性，每个个体在适应度评估 $f(X_i)$ 上附加一个区间均匀随机数 $r_i = U(b_i, e_i)$。i 是 i^{th} 个体的索引。U 表示在区间内生成一个均匀分布值，其中包含 the

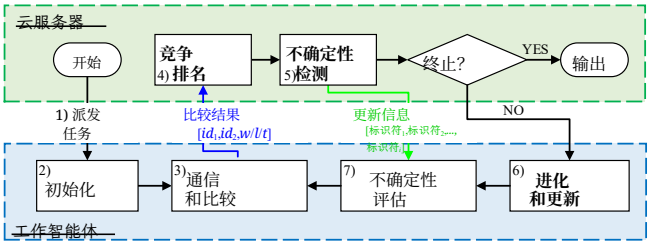


图 3. CEC 的框架。

开始于b_i并结束于e_i。然后，所有智能体通过环境感知适应度评估$F(X_i) = f(X_i) + r_i$来评估其个体。拓扑是随机初始化的。（第 1 行至第 4 行）

步骤 3) 通信。工作智能体不仅需要与邻居通信，还需要与 S 通信。首先，智能体可以与其相连邻居通信。由于数据敏感，除非S提出请求，否则智能体不愿透露其决策变量向量。每个智能体a_i 将不确定性适应度$F(X_i)$发送给每个 $j \in N(i)$，其中$N(i)$是a_i的邻居集合。其次，智能体可以通过发送比较结果元组 $[id_1, id_2, w/l/t]$ 与S 通信。具体而言， id_i 和id_2 分别是两个相连智能体的索引。w、 l、 t 分别表示智能体id_1 的不确定性适应度优于、劣于或等于智能体id_2。例如，对于最优值大于 0 的最小化优化问题，两个相连智能体 a_i、 a_j、 $[i, j, w]$ 表示$F(X_i) < FQX_j$R， $[i, j, l]$ 表示 $F(X_i) > FQX_j$R， $[i, j, t]$ 表示 $F(X_i) = FQX_j$R。（第 6 至 12 行）

第 4 步).竞争排序。接收到比较结果后， S 可以形成比较矩阵 $\mathbf{M} = \begin{matrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{matrix}$Z。对于每个个体 a_i $p_{ij} =$ NAN i NAN 、

$p_{ii} = NAN$和 ，如果 $j \notin N()$。这意味着智能体不会与自身或未连接的智能体进行比较。因此，当拓扑稀疏时，**M** 可能也是稀疏的。基于**M**，S 进行竞争排序，并获得一个个体索引的排名序列$\boldsymbol{Seq}$，其中排名越高越好。竞争排序的细节将在下一部分中详细阐述。（第 13 行）

第 5步) 不确定性检测。基于排名序列，S 进行不确定性检测，以检测出具有高水平不确定性的智能体。这些智能体将从工作众包人群中移除，以节省适应度评估的预算。不确定性检测的详细内容将在第3.3 节中阐述。如果达到终止条件，即适应度评估的预算耗尽， S 将在不确定性检测后评估排名最高的候选者。需要注意的是，S 没有环境感知不确定性，因此能够获得任务的真实适应度。否则，算法将转到第6 步，引导智能体进行进一步进化。（第 14 行至第 18 行）

步骤 6) 通信与进化。完成不确定性检测后，S 对所有智能体的可靠性有了总体认知。它可以请求适应度较高的智能体将其决策变量发送给适应度较低的邻居。适应度较低的邻居可以通过特定的进化策略进行自我更新。（第 19 至 20 行）

算法1 *CEC*伪代码**输入**：优化任务 P ，目标函数为 $f(X)$，搜索空间为 D /* 初始化 */ 1. 在每个工作者 2 $F\ X_i\ = f\ X_i\ + r_i$ 3 上随机初始化一个个体 $X_i \in D$ 。评估 ( ) ( )。随机初始化拓扑 4 $g = 1$ $fes = 0$ 。 5 进化世代，已耗尽的评估次数。**当**适应度评估尚未耗尽**执行** /* 通信 */ 6. **对于**每个工作者 i7 j i 。**如果**是的邻居。将 (X_i) 发送给 j9 $F\ X_i$ $F(X_j)$ 。比较 ( ) 和 。将比较结果 $[id_1, id_2, w\ l\ t]$ S / / 发送给。 11 **结束如果**12. **结束对于**/* 竞争排序 */ 13. 根据公式 (1)-(5) 对 S 进行竞争排序 /* 不确定性检测 */ 14. **If** $mod(g, u) = 0$ 15 $rank_i^j = 1$ $rank_i^j = 4$ $j = g - u + 1\ g$ 。**如果**或 16 i 对所有：。检测到不可靠，并将其从众包人群中移除 17. **结束如果** 18. **结束如果**/* 通信与进化 */ 19. 由 S 向工作者发送排名信息 20. 工作者通过特定的进化算子，根据排名更高的邻居更新自身 /* 不确定性评估 */ 21 $F\ X_i\ = f\ X_i\ + r_i$ 。评估 ( ) ( )。更新和 22 fes $g = g + 1$ 。随机改变拓扑 24. **结束当 输出**: 23 优化后的解 X

步骤 7) 不确定性评估。完成进化和更新后，从适应度较高的邻居处学习的智能体会进行不确定性评估，随后算法继续执行步骤 3)。同时，已耗尽的评估次数和进化代数也会更新。由于众包人群具有移动性，拓扑是动态的。（第 21 至 23 行）

3.2 竞争排序

竞争排序仅依据 **M** 来区分智能体。实际上，成对邻居比较结果与体育比赛或游戏中的竞争情形极为相似。在体育比赛或游戏中，不可能所有选手都有机会相互较量，排名通常通过累积得分计算得出 [34]。受此启发，我们基于 **M** 提出了竞争排序。为便于标记，我们使用 w、 l、 t 分别表示胜、负和平局。具体而言，在 **M** 中，w、 l、 t 分别标记为 1 、 0 、 0.5。竞争排序的流程如下。

1）初始化模糊矩阵 $\mathbf{M}_f$、累计胜场次数矩阵 $\mathbf{M}_w$ 和累计负场次数矩阵 $\mathbf{M}_l$ 。具体而言，模糊矩阵初始化为 $\mathbf{M}_f$ =U 0 5 ⋮ 0 5 … ⋱ … 0 5 ⋮ 0 5 。 . . . . Z $n \times n$，这表示在比较之前没有结果信息，所有智能体都被视为相同。 $\mathbf{M}_w$ 和 $\mathbf{M}_l$ 初始化为 U 0 ⋮ 0 … ⋱ … 0 ⋮ 0 Z $n \times n$ ，这是因为在比较之前，没有

既没有获胜也没有失败的累计时间。

2) 根据$\mathbf{M}$更新$\mathbf{M}_w$和$\mathbf{M}_l$。更新规则如下。

$$\begin{aligned}&\mathbf{M}_w(i,j)=\mathbf{M}_w(i,j)+1, \quad \text{if } \mathbf{M}(i,j)=1\\&\mathbf{M}_l(i,j)=\mathbf{M}_l(i,j)+1, \quad \text{if } \mathbf{M}(i,j)=0\\&\left.\begin{aligned}\mathbf{M}_w(i,j)=\mathbf{M}_w(i,j)+0.5\\\mathbf{M}_l(i,j)=\mathbf{M}_l(i,j)+0.5\end{aligned}\right\},\ \text{if } \mathbf{M}(i,j)=0.5\end{aligned} \tag{2}$$

对于每个i，$j\in[1,2,\ldots,n]$，$\mathbf{M}(i,j)=1$表示$F(X_i)<FQX_j\mathrm{R}$，$\mathbf{M}(i,j)=0$表示$F(X_i)>FQX_j\mathrm{R}$，$\mathbf{M}(i,j)=0.5$表示$F(X_i)=FQX_j\mathrm{R}$，而$\mathbf{M}(i,j)=NAN$表示$i=j$或$j\notin N(i)$。由(2)可得$\mathbf{M}_w+\mathbf{M}_l=\mathbf{M}_t$，其中$\mathbf{M}_t$中的每个元素都是比较总次数。特别地，$\mathbf{M}_t$是一个对称矩阵，其中$\mathbf{M}_t(i,j)=\mathbf{M}_t(j,i)$。

3) 根据$\mathbf{M}_w$和$\mathbf{M}_l$更新$\mathbf{M}_f$。此过程采用[34]中提出的余弦相似度度量，该方法已被证明能够使更新后的模糊矩阵具有完美的一致性。更新规则如下：

$$\mathbf{M}_f(i,j)=\begin{cases}\frac{\mathbf{M}_w(i,j)}{\mathbf{M}_t(i,j)}-\lambda\times 2^{-\mathbf{M}_t(i,j)/\max\mathbf{M}_t}, & \text{if } \mathbf{M}_w(i,j)>\frac{\mathbf{M}_t(i,j)}{2}\\0.5, & \text{if } \mathbf{M}_w(i,j)=\frac{\mathbf{M}_t(i,j)}{2}\\\frac{\mathbf{M}_w(i,j)}{\mathbf{M}_t(i,j)}+\lambda\times 2^{-\mathbf{M}_t(i,j)/\max\mathbf{M}_t}, & \text{if } \mathbf{M}_w(i,j)<\frac{\mathbf{M}_t(i,j)}{2}\end{cases} \tag{3}$$

其中，$\max\mathbf{M}_t$是$\mathbf{M}_t$中的最大数。模糊值$\mathbf{M}_f(i,j)$由获胜次数除以总次数确定。这意味着，使用修正后的胜率比较两个智能体，以避免胜率相同但总次数不同的智能体具有相同模糊值的情况。因此，使用$\lambda\times 2^{-\mathbf{M}_t(i,j)/\max\mathbf{M}_t}$。

然后，$\mathbf{M}_f$按如下方式进行转换。

$$\mathbf{M}_f(i,j)=\frac{\mathbf{M}_f(i,j)}{1-\mathbf{M}_f(i,j)} \tag{4}$$

要使用余弦相似度度量，$\mathbf{M}_f$需转换如下。

$$\mathbf{M}_f(i,j)=\frac{\mathbf{M}_f(i,j)}{\sqrt{\mathbf{M}_f(*,j)}} \tag{5}$$

其中$\mathbf{M}_f(*,j)=\sum_{i=1}^{n}\mathbf{M}_f(i,j)$，为各列之和。优先级向量$\boldsymbol{PRI}$按如下方式计算。

$$\boldsymbol{PRI}=\frac{\mathbf{M}_f(i,*)}{\mathbf{M}_f(*,*)} \tag{6}$$

其中，$\mathbf{M}_f(i,*)=\sum_{j=1}^{n}\mathbf{M}_f(i,j)$，且$\mathbf{M}_f(*,*)=\sum_{i=1}^{n}\mathbf{M}_f(i,*)$。$\boldsymbol{PRI}$的值越大，工作者的排名越高。

这是由$\mathbf{M}$与$\boldsymbol{PRI}$之间的关系在$\mathbf{M}$时完全一致所导致的。对于任意两个向量$\langle\vec{u},\vec{v}\rangle$，余弦相似度度量计算为$cos\langle\vec{u},\vec{v}\rangle$ = ;;;·v;·?|;;;|??|v;·|?.已证明，当满足条件$\mathbf{M}_f(i,j)=\boldsymbol{PRI}(i)\boldsymbol{PRI}(j)$ [35]时，$\mathbf{M}$完全一致。因此，如果$\mathbf{M}$完全一致，列向量$\mathbf{M}_f(*,j)$与$\boldsymbol{PRI}$之间的~~余弦相似~~度度量等于1。

为便于理解，我们以一个包含五名工作者的全连接拓扑为简单示例。下面

suppose $\mathbf{M}=\begin{pmatrix}NAN&0&0&0&1\\1&NAN&1&1&1\\1&0&NAN&0&1\\1&0&1&NAN&1\\0&0&0&0&NAN\end{pmatrix}$. By calculating $\mathbf{M}_w,\mathbf{M}_l,\mathbf{M}_t$ by (2), $\mathbf{M}_w=\begin{pmatrix}0&0&0&0&1\\1&0&1&1&1\\1&0&0&0&1\\1&0&1&0&1\\0&0&0&0&0\end{pmatrix}$, $\mathbf{M}_l=\begin{pmatrix}0&1&1&1&0\\0&0&0&0&0\\0&1&0&1&0\\0&1&0&0&0\\1&1&1&1&0\end{pmatrix}$, $\mathbf{M}_t=\begin{pmatrix}0&1&1&1&1\\1&0&1&1&1\\1&1&0&1&1\\1&1&1&0&1\\1&1&1&1&0\end{pmatrix}$. Originally, $\mathbf{M}_f=\begin{pmatrix}0.5&0.5&0.5&0.5&0.5\\0.5&0.5&0.5&0.5&0.5\\0.5&0.5&0.5&0.5&0.5\\0.5&0.5&0.5&0.5&0.5\\0.5&0.5&0.5&0.5&0.5\end{pmatrix}$. After calculation of (3), $\mathbf{M}_f=\begin{pmatrix}0.5&0.005&0.005&0.005&0.995\\0.995&0.5&0.995&0.995&0.995\\0.995&0.005&0.5&0.005&0.995\\0.995&0.005&0.995&0.5&0.995\\0.005&0.005&0.005&0.005&0.5\end{pmatrix}$. By calculating (4) and (5), $\mathbf{M}_f=\begin{pmatrix}0.0029&0.005&0&0&0.5\\0.5773&0.9999&0.7071&1&0.5\\0.5773&0.005&0.0036&0&0.5\\0.5773&0.005&0.7071&0.005&0.5\\0&0.005&0&0&0.0025\end{pmatrix}$. Finally, the priority vector $\boldsymbol{PRI}$ calculated by (6) is $(0.0707,0.5270,0.1512,0.2499,0.0011)^{\mathrm{T}}$. Therefore, the final ranking result in terms of sorted index is [4,1,3,2,5]. It indicates the sorted index from the best to the worst after ranking, which means the 2nd agent is best and the 5th agent is worst.

3.3 不确定性检测

不确定性检测是一种识别不可靠智能体的策略。不可靠智能体可能是由智能体行为的不确定性和设备异质性造成的。更糟糕的情况是，众包人群中存在恶意攻击者。因此，不确定性检测对于识别不可靠智能体并节省适应度评估以用于后续进化至关重要。

不确定性检测基于优先级向量$\boldsymbol{PRI}$。该过程按照算法1第14至18行执行，如图4所示。首先，根据$\boldsymbol{PRI}$将所有工作智能体按优先级从高到低重新排序。然后将它们分为四个层级$L1$、$L2$、$L3$、$L4$，每个层级平均包含$\lfloor n/4\rfloor$个智能体，其中$\lfloor n/4\rfloor$是总共n个智能体除以4个层级后的向下取整函数。因此，$L1$包含优先级最高的$\lfloor n/4\rfloor$个智能体，$L2$包含优先级第二高的$\lfloor n/4\rfloor$个智能体，$L3$包含优先级第三高的$\lfloor n/4\rfloor$个智能体，而$L4$包含优先级最低的$n-\lfloor n/4\rfloor\times 3$个智能体。智能体在不同世代被分到不同层级是正常情况。这是因为拓扑在每一代都会变化，因此智能体的进化过程不可控。相反，如果某个智能体在多个世代中经常被分到$L1$或$L4$，则它可能是不可靠的智能体。

上述现象是合理的。不确定性

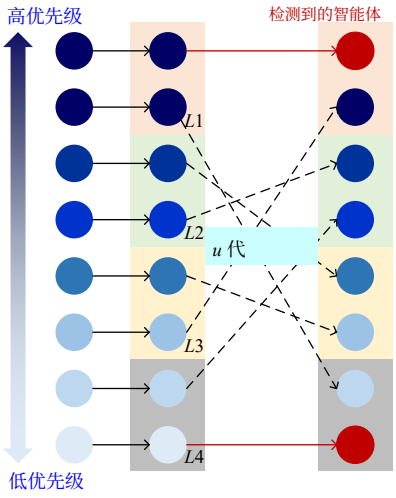


图4. 不确定性检测示意图，其中L表示分类等级，u 表示检测到的代数。

第 i 个个体 $F(X_i)$ 的适应度受纯适应度 $f(X_i)$ 和不确定性 $r_i = U(b_i, e_i)$ 的影响。尽管个体只知道 $F(X_i)$，但 $f(X_i)$ 与 r_i 之间的关系对其有很大影响。以最小化问题为例。如果 $|r_i| \ll f(X_i)$，这意味着 r_i 的绝对值远小于 $f(X_i)$，那么 $F(X_i)$ 主要受 $f(X_i)$ 影响。在这种情况下，更新 $f(X_i)$ 可能会导致不同的排序结果，如图4中的蓝色圆圈所示。如果 $|r_i| \gg f(X_i)$，这意味着 r_i 的绝对值远大于 $f(X_i)$，那么 $F(X_i)$ 主要受 r_i 影响。更新 $f(X_i)$ 对 $F(X_i)$ 的影响很小。r_i 有两种情况。当 r_i 是一个远大于 $f(X_i)$ 的正数时，该个体远差于其他个体，并且经常被归类为排名中的末位，如图4中的底部红色圆圈所示。当 r_i 是一个远小于 $f(X_i)$ 的负数时，该个体远优于其他个体，并且经常被归类为排名中的首位，如图4中的顶部红色圆圈所示。

因此，在 u 代中频繁被归为第一类或最后一类的智能体会被判定为不可靠个体。它们将被排除在众包人群之外，不再优化该任务。此处，检测代数 u 是一个重要参数，经实验研究后将其设为 100。

4 Experiment Investigation of CEC

本节首先介绍测试套件。其次，给出实验设置，其中包括 CEC、CLLSO 的示例以及参数设置。然后，我们进行参数研究和策略研究。接下来，通过对 CLLSO 与其他传统的集中式大规模优化进化计算算法开展实验，展示 CEC 的性能。最后，通过比较 CLLSO 与其他分布式优化方法在分布式聚类优化方面的表现，展示 CEC 的应用前景。

4.1 测试套件

为展示 CEC 的性能，我们在两个应用最广泛的大规模全局优化测试套件 CEC2010 [36] 和 CEC2013 [37]上进行了实验。

算法2 *LLSO* 伪代码

输入：整个种群 P
1. 根据适应度将种群 P 划分为 L 个层级
2. 将第一层级中的个体保留到下一代
3. **对于** 每个个体，适用于从 2nd 到 4th 级别
4. 随机选择两个不同的较高层级索引 $k1, k2$
5. **如果**它是 2nd 级别
6. $k1 = k2 = 1$
7. **结束 If**
8. 随机选择一个个体用于 $k1$、 $k2$、 $x_{k1,r1}, x_{k2,r2}$
9. 使用更新规则生成子代
10. 检查并修改决策边界的边界
11. **结束循环**

输出：更新后的种群

详细介绍和实现代码可见其提出论文。具体而言，CEC2010 包含 20 个具有不同属性的测试问题，而 CEC2013 包含 15 个属性更复杂的测试问题。CEC2010 和 CEC2013 中问题的维数均为 1,000，这带来了维数灾难的挑战。

4.2 实验设置

4.2.1 CLLSO

为通过大量实验说明 CEC 的良好性能，本部分以 CLLSO 为例进行介绍。CLLSO 使用 LLSO 作为进化优化器。LLSO 旨在增强进化计算在大规模优化中的开发和探索能力[33]。CLLSO 的流程与图 3 相同，其中 **进化与更新** 流程采用 LLSO 进化过程。LLSO 的伪代码见算法2，具体流程如下。

在每个世代中，群体 P 首先根据适应度分为四个层级。为保持开发能力，第一个层级中的粒子直接进入下一世代。为提高探索能力，其他层级中的粒子可以向更高层级中的粒子学习。具体而言，首先随机选择两个不同的更高层级 $k1$、 $k2$ 。如果当前粒子 x_i 位于2nd 层级，则 $k1 = k2 = 1$。然后从每个层级中随机选择一个粒子，形成两个学习粒子 $x_{k1,\ r1}$、 $x_{k2,\ r2}$。x_i 基于以下更新规则进行更新。

$$v_i^d = r_1 \times v_i^d + r_2 \times (x_{k1,r1}^d - x_i^d) + \varphi \times r_3 \times (x_{k2,r2}^d - x_i^d) \quad (7)$$

$$x_i^d = x_i^d + v_i^d \quad (8)$$

其中，v_i^d 表示第 i个粒子在第d维中的速度。r_1、 r_2、 r_3 是[0,1]之间的三个随机小数。φ 是控制参数，在[33]中设为4。如果更新后的粒子在任意维度上超出边界，则将其重新初始化至搜索域内。

由于在大规模优化中表现出色，LLSO已在昂贵优化、[38], 多任务优化[39], 和故障诊断[40] 方面取得了进展。因此，本文采用它作为进化计算优化器，以实现一个CEC示例。

4.2.2 参数设置

有三个参数需要明确，即 FES、 NP 和 u。适应度评估的最大次数为 FES

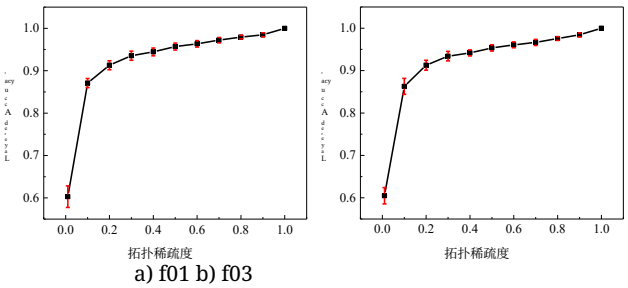
种群规模（也称为群体规模）NP 根据 [33] 中的建议进行设置。具体而言，$FES = 1000 \times D$，其中 D 为维数。对于维数为 1,000 的问题，$NP = 500$ 。新参数 u 即检测代数，通过实验研究设置为 100。

为了模拟环境感知不确定性，我们还给出了不确定性水平的设置。为区分不同的不确定性水平，使用指数形式 2^{ul_i} 设置第 i 个个体的边界值 bv_i，其中 ul_i 为不确定性水平。在不确定性检测实验中，我们讨论 1）正不确定性，其中 $r_i = U(0, bv_i)$；以及 2）负不确定性，其中 $r_i = U$ 0 (bv_i,)。在其他实验中，我们采用正不确定性 $r_i = U(0, bv_i)$。

此外，实验在配备 Intel® CoreTM i5-9400 CPU @2.90GHz 处理器和 16.0 GB 内存的机器上进行。所有结果均取 25 次独立运行的平均值，以确保比较公平。

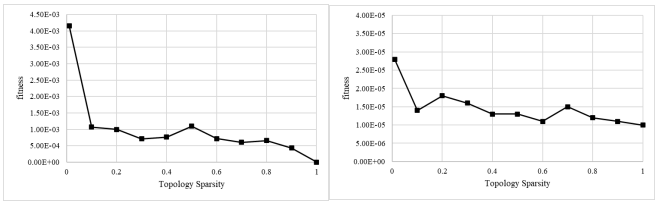
4.3 竞争排序的研究

本节研究竞争排序的影响。排序结果完全取决于比较矩阵，其中拓扑稀疏度是一个重要因素。根据排序结果，将智能体划分为四个层级，以执行基于层级的学习进化算子。因此，分层准确率对进化至关重要。我们在实验中将智能体的拓扑稀疏度设为 0 4、0 01、0 1、0 2、0 3、0 5、0 6、0 7、0 8、0 9、1 0.以研究从稀疏到稠密情况下的分层准确率，这意味着每个智能体的邻居数分别等于 0 01 × NP 。，0 1 × NP 0 2 × NP 0 3 × NP 0 4 × NP 0 5 × NP 0 6 × NP .，.，.，.，.，.，.，0 7 × NP 8 × NP 9 × NP 0 × NP 0 01 ×.， NP 0.，0.，1.。特别是，. 表示更稀疏的拓扑，而 1 0 × NP. 表示全连接拓扑。尽管每一代的拓扑都会随时间变化，但最大邻居数是固定的。为便于说明，我们以 CEC2010 中的 f01 和 f03 为例，展示不同拓扑稀疏度下的分层准确率，见图5；优化结果见图6。黑点表示进化过程中种群的平均分层准确率，红线表示标准差。



a) f01 b) f03

图5. 不同拓扑稀疏度水平下的分层准确率，其中 a) f01，b) f03。黑点表示进化过程中种群的平均分层准确率，红线表示标准差。



a) f01 b) f03

图6. 不同拓扑稀疏度水平下的优化适应度，其中 a) f01，b) f03。

由结果可以得出，更稀疏的拓扑会导致较低的分层准确率，而更稠密的拓扑会带来较高的分层准确率。具体而言，全连接拓扑达到100%的平均分层准确率和 0的标准差，说明了竞争排序策略的有效性。相应地，随着拓扑变得更加稠密，优化性能也会变得更好。这一现象很容易理解，因为稀疏拓扑所反映的信息少于稠密拓扑。此外，当拓扑稀疏度大于 0.1 时，分层准确率高于90%，且优化适应度与全连接拓扑相比保持稳定。因此，我们的方法对拓扑稀疏度的变化具有鲁棒性。云端的竞争排序能够与基于层级的学习算子良好协作，从而引导工作智能体的进化。为了模拟稀疏拓扑，我们将拓扑稀疏度设置为 0.1，作为后续实验的示例。

4.4 不确定性检测研究

本节讨论两种情况下的不确定性检测策略：1）存在 $r_i =$ 的正不确定性 $U(0, bv_i)$ 和 2）负不确定性，其中 $r_i =$ U 0 10 (bv_i,)。不失一般性，我们将不确定性绝对值大于 1 的个体比例设为 %。其余 90% 个体的不确定性绝对值小于 1。$|bv_i|$ 的最大值设为 $2^{30} = 1$，原因有二。首先，bv_i 的最大值小于随机初始化个体的适应度值 $f(X_i)$。在算法开始时，$r_i \ll f(X_i)$ 和不确定性适应度 $F(X_i) = f(X_i) + r_i$ 主要由 $f(X_i)$ 主导。其次，bv_i 的最大值大于全局最优适应度值。当算法进化到有希望的区域时，$r_i \gg f(X_i)$ 和 $F(X_i)$ 主要由 r_i 主导。因此，在算法进化过程中，r_i 与 $f(X_i)$ 之间的关系会从 $r_i \ll f(X_i)$ 变为 $r_i \gg f(X_i)$。这有助于在这两种情况下测试和观察算法性能。具体设置如下：

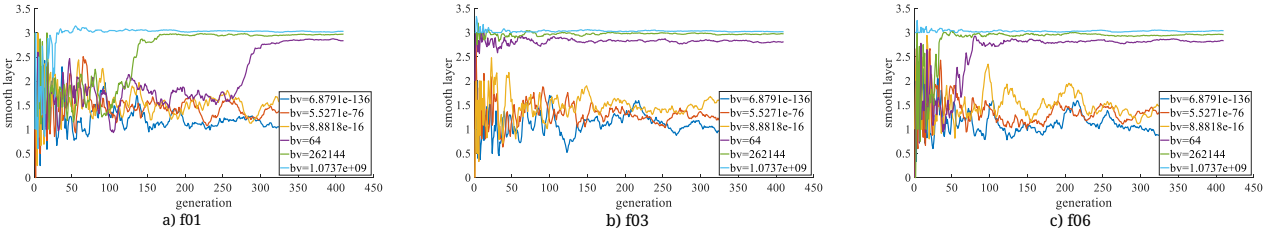
$$|bv_i| = \begin{cases} 2^{-(NP\times 90\%-i)}, & i = 1,\ 2,\ \ldots\ldots,\ NP \times 90\%\ 30 \\ 2^{(i-NP\times 90\%)\times \frac{\times}{NP10\%}}, & i = NP \times 90\% + 1,\ \ldots\ldots,\ NP \end{cases}$$

对于时变拓扑，每个个体的最大邻居数设为 $NP \times 0.01$。

4.4.1 不确定性检测的分层分析

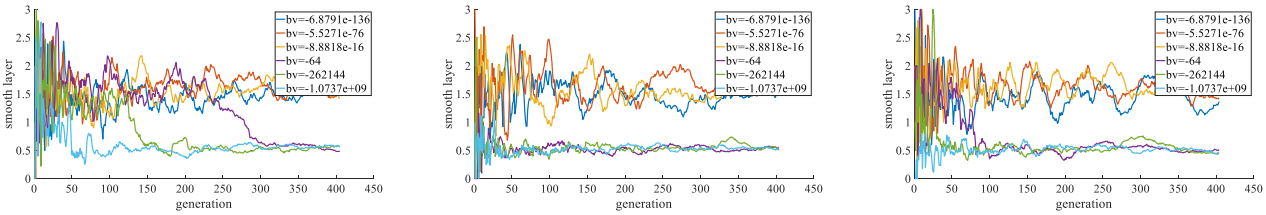
本节分析整个进化过程中具有不同不确定性水平的智能体的分层结果。我们选取不确定性水平为 $|bv_i| = 6\ 8791e -$ 的六 136 5 5271e − 76 8 8818e − 16 64 262144 1 0737e + 个智能体进行说明。，.，.，.，.，. 09。对于正不确定性，$bv_i = |bv_i|$ 和 $r_i = U(0, bv_i)$。在这种情况下，进化过程中的分层结果如图 7 所示。对于负不确定性，$bv_i = -|bv_i|$ 和 $r_i = U(bv_i,$ 0)。在这种情况下，进化过程中的分层结果如图 8 所示。我们对连续 10 代取平均值，以计算平滑的层。受篇幅限制，图 7 和图 8 仅展示了 f01、f03、f06 的结果。

从图中可以看出，不确定性水平较高的智能体 $|bv_i|$ 更容易且更早地被



a) f01 b) f03 c) f06

图 7. 具有正不确定性的 6 个智能体的分层结果，其中 a) f01、b) f03、c) f06。bv 是不确定性水平的边界值。



a) f01 b) f03 c) f06

图 8. 6 个 ag 的分层结果　　具有负不确定性的 ents，其中 a) f01，b) f03，c) f06。bv 是不确定性水平的边界值。　　.

检测到。这是因为较大的不确定性值对排序结果的影响更大。$F(X_i) = f(X_i) + r_i$ 受到$f(X_i)$和r_i的影响。如果$|r_i|$ 远大于$f(X_i)$，$F(X_i)$主要受r_i的影响。在进化开始时，$f(X_i)$较大，只有极高水平的$|r_i|$ 才能支配$F(X_i)$。因此，只有具有较大不确定性的智能体（例如$|bv| = 1.0737E+09$）会被检测到。随着种群进化，适应度逐渐提高，具有相对较小不确定性的智能体（例如$|bv| = 262144$ 或64）也会被检测到。因此，CEC进行不确定性检测是有效的，并且经过充分进化后，可以检测到更小的不确定性。

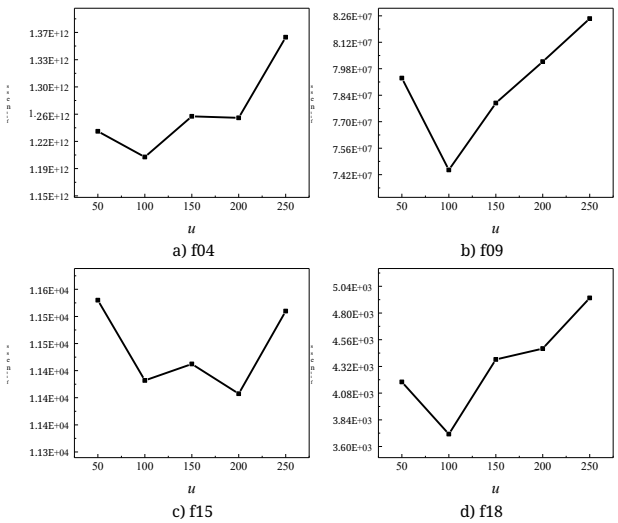
4.4.2 检测代数的考察

本节考察不同检测代数

u　　$u = 50$ 100 150 的影响。200　　250　　04，09，15

我们设置了、和进行实验。f、f和f18的结果如图9所示。

从图中可以发现，$u = 100$ 是一个适当的设置。在不失一般性的情况下，当u 小于100时，随着u的增加，算法性能会变得更好。在少数几代内维持在顶部或底部水平，可能是由较优个体的偶然因素造成的，而不是不可靠的 不确定性



a) f04 b) f09 c) f15 d) f18

图 9. 不同检测代数 u 下的性能比较，其中 a) f04，b) f09，c) f15，d) f18。

个体。丢弃这些有前景的个体不利于种群进化。在一种极端情况下，当每一代都检测并丢弃　　$u =$ 1　　个

有前景的个体时，会对进化过程产生很大影响。当 u 大于 100 时，随着 u 的增加，算法性能会变差。检测到的世代数越多，不可靠个体存在的时间就越长。它们可能会误导种群进化，并耗费更多适应度评估。当 u 等于最大进化世代数时，则不会进行不确定性检测，种群会完全受到具有较大不确定性的个体影响。因此，根据实验结果，我们在算法中设置了 $u = 100$ 。

4.4.3 不确定性检测的影响分析

为考察不确定性检测的影响，本节通过实验比较采用不确定性检测的 CLLSO（CLLSO¯wUD）与未采用不确定性检测的 CLLSO（CLLSO_woUD）所找到的最优解。结果记录于表2。

可以得出结论：在20个问题中，有16个问题上 CLLSO¯wUD 的表现优于 CLLSO¯woUD。基于两个原因，进行不确定性检测是必要的。首先，不确定性较大的个体会对种群进化产生重大影响。它们可能会将种群引向没有希望的方向，导致无法找到近似最优解。其次，不确定性较大的个体会在每一代消耗适应度评估。在适应度评估预算有限的情况下，与其将评估浪费在不确定性较大的个体上，不如让可靠个体进化。因此，不确定性检测有助于检测不确定性较大的个体，并推动种群进化。

4.5 与全局优化算法的比较

基于众包的分布式优化具有不确定性适应度、弱中心化、时变拓扑和数据敏感性等特点，带来了巨大的求解挑战。为展示 CEC 的全局优化能力，本部分通过实验将 CLLSO 与其他传统集中式进化计算算法进行比较。具体而言，我们选取了三种适用于大规模优化的粒子群优化器变体：社会

表 2 结果 比较 CLLSO wUD 与 CLLSOwoUD_ _

概率-lems	CLLSO woUD_		CLLSO wUD_	
	mean	std	mean	std
f01	1.0200E-04	1.00E-05	0.0000E+00	0.00E+00
f02	1.1241E+04	1.06E+02	1.1216E+04	1.07E+02
f03	2.0000E-05	1.00E-06	0.0000E+00	0.00E+00
f04	1.5189E+12	2.90E+11	1.2034E+12	2.02E+11
f05	3.3006E+08	1.18E+07	3.2079E+08	1.07E+07
f06	3.9900E-04	3.00E-05	0.0000E+00	0.00E+00
f07	2.4702E+05	1.07E+05		3.40E+02
f08	3.8162E+07	1.59E+05	3.5462E+07	2.75E+05
f09	8.1738E+07	3.90E+06		7.08E+06
f10	1.1343E+04	9.67E+01	1.1343E+04	9.37E+01
f11	5.5500E-04	4.90E-05	0.0000E+00	0.00E+00
f12	1.6861E+06	9.26E+04	9.8556E+04	1.03E+04
f13	8.5754E+02	3.38E+02	9.0334E+02	2.87E+02
f14	3.4154E+08	2.01E+07	2.7799E+08	1.86E+07
f15	1.1388E+04	9.35E+01	1.1401E+04	7.26E+01
f16	4.1606E-02	2.05E-01	2.5000E-05	7.00E-06
f17	3.7611E+06	1.27E+05	1.8145E+06	2.76E+05
f18	3.4836E+03	2.12E+03	3.7119E+03	4.86E+03
f19	1.0563E+07	5.35E+05	9.6938E+06	6.83E+05
f20	1.0482E+03	5.11E+01	1.0574E+03	8.26E+01

学习粒子群优化器（SL-PSO）[41], 竞争群体优化器（CSO） [42], 动态基于等级学习的群体优化器（DLLSO）[33]，以及一种适用于大规模优化的协同进化优化器——结合协同进化与差分分组的差分进化算法（DECC-DG）[43]。为确保比较公平，所有参数均按照各算法原始论文中的建议进行设置。适应度评估的最大次数设为 $1000 \times D$。值得强调的是，所有参与比较的算法都是集中式优化算法，不受环境感知不确定性、数据保护或连接拓扑等因素的限制。换言之，它们是在传统全局优化问题上进行测试，而非分布式优化问题，因为在如此多的限制条件下，实现这些进化原理颇具挑战。CEC2010 和 CEC2013 中维度为 1,000 的问题结果分别见表 3 和表 4。

除了均值和标准差外，Wilcoxon 秩和检验还对 25 次独立运行进行统计显著性检验。最后一行汇总了显著性水平为 0.05 时的检验结果，其中 #(+/–/≈) 表示与 CLLSO 相比，被比较算法显著更优、显著更差或差异不显著的问题数量。根据这些表格可以得出以下结论。

与 CLLSO 相比，DECC-DG 在 20 个 CEC2010 问题中的 15 个问题和15 个 CEC2013问题中的 12 个问题上表现显著更差。DECC-DG是一种协同协进化算法，并使用差分分组结果进行进化。其性能高度依赖于决策变量的分组。它无法像其他集中式全局优化算法（如 SL-PSO、CSO、DLLSO）那样快速获取全局信息。因此，它在大多数问题上无法找到令人满意的结果。与 CLLSO 相比，SL-PSO 在 20 个 CEC2010 问题中的 15 个问题和15个 CEC2013 问题中的 12 个问题上表现显著更差。SL-PSO 是一种广泛使用的PSO，因为它加入了社会认知部分，帮助群体搜索有潜力的区域并摆脱局部最优。然而，对于大规模优化而言，它缺乏局部搜索策略。因此，SL-PSO无法在大规模优化问题上取得良好表现。与 CLLSO 相比，CSO 在 20 个 CEC2010 问题中的 16 个问题和15个 CEC2013 问题中的 12 个问题上表现显著更差。CSO专为大规模优化而设计。由于采用竞争学习策略，它可以显著提高种群多样性，从而摆脱局部最优。然而，由于没有局部最优粒子或全局最优粒子的引导。因此，其收敛速度可能较慢。与 CLLSO 相比，DLLSO 在 20 个 CEC2010 问题中的 17 个问题和 15 个 CEC2013 问题中的 12 个问题上表现显著更差。DLLSO 的灵感来源于教育学中的基于层级的学习。它将群体划分为若干层，并促使低层级粒子向高层级粒子学习。尽管其探索和开发能力得到了提升，但也可能存在收敛速度较慢的问题。

尽管 CEC 是为存在诸多限制的分布式优化而设计的，但它仍具备令人满意的全局优化能力。其找到的最佳结果优于或可与传统进化计算算法在集中式

表 3 R结果 C对比 CLLSO 与 F四种 O其他 A算法在 CEC2010

	DECC-DG			SL-PSO			CSO			DLLSO			CLLSO	
	mean	std	p	mean	std	p	mean	std	p	mean	std	p	mean	std
f01	6.02E+05	6.9E+05	9.7E-11	3.62E+4	5.3E+4	9.7E-11	3.62E+2	4.3E+1	9.7E-11	2.23E-1	8.9E-2	9.7E-11	0.00E+0	0.0E+0
f02	4.41E+03	1.4E+02	1.4E-09	9.25E+3	3.9E+2	1.4E-09	9.38E+3	1.2E+2	1.4E-09	8.40E+3	2.8E+2	1.4E-09	1.12E+4	1.1E+2
f03	1.66E+01	3.9E-01	9.7E-11	2.11E+1	5.5E-2	9.7E-11	4.53E-2	1.6E-2	9.7E-11	3.38E-4	7.0E-5	9.7E-11	0.00E+0	0.00+0
f04	3.24E+13	8.9E+12	1.4E-09	2.34E+6	1.1E+6	1.4E-09	6.1E+12	6.E+11	1.4E-09	2.6E+12	5.E+11	1.5E-08	1.2E+12	2.E+11
f05	2.08E+08	2.8E+07	1.4E-09	4.95E+6	2.7E+6	1.4E-09	4.15E+6	1.4E+6	1.4E-09	2.94E+8	1.1E+7	1.1E-07	3.21E+8	1.1E+7
f06	1.66E+01	3.8E-01	1.2E-09	2.11E+7	3.1E+4	1.2E-09	4.21E+6	8.0E+6	1.2E-09	4.37E-2	4.7E-2	1.2E-09	0.00E+0	0.0E+0
f07	4.15E+07	1.4E+07	1.4E-09	7.75E+7	2.6E+7	1.4E-09	2.93E+6	6.9E+5	1.4E-09	1.24E+5	3.1E+4	1.4E-09	3.33E+2	3.4E+2
f08	5.89E+07	3.0E+07	1.4E-09	2.52E+8	2.8E+8	1.4E-09	1.02E+8	1.0E+8	1.4E-09	4.62E+7	1.2E+7	1.4E-09	3.55E+7	2.8E+5
f09	3.47E+08	2.5E+07	1.4E-09	6.65E+6	9.5E+6	1.4E-09	2.93E+8	2.3E+7	1.4E-09	1.58E+8	1.5E+7	1.4E-09	7.45E+7	7.1E+6
f10	7.34E+03	1.3E+02	1.4E-09	1.26E+4	4.3E+2	5.9E-09	9.95E+3	7.6E+1	1.4E-09	1.01E+4	8.6E+1	1.4E-09	1.13E+4	9.4E+1
f11	1.21E+01	7.1E-01	1.3E-09	2.37E+2	7.7E-2	1.3E-09	5.01E+1	2.4E+1	1.3E-09	6.26E-1	1.8E+0	1.3E-09	0.00E+0	0.0E+0
f12	1.18E+05	7.3E+03	1.4E-01	1.56E+6	1.1E+5	1.4E-09	1.62E+6	9.6E+4	1.4E-09	5.34E+5	6.6E+4	1.4E-09	9.86E+4	1.0E+4
f13	1.98E+09	5.4E+08	1.4E-09	1.36E+4	9.8E+3	1.4E-09	3.34E+3	3.5E+3	2.2E-06	1.55E+3	5.7E+2	5.6E-06	9.03E+2	2.9E+2
f14	1.27E+09	6.5E+07	1.4E-09	2.94E+8	3.3E+8	4.3E-06	1.27E+9	6.5E+7	1.4E-09	5.39E+8	4.3E+7	1.4E-09	2.78E+8	1.9E07
f15	7.22E+03	8.9E+01	1.4E-09	1.12E+4	1.2E+2	4.6E-09	1.02E+4	5.2E+1	1.4E-09	1.04E+4	5.7E+1	1.4E-09	1.14E+4	7.3E+1
f16	1.83E-02	1.3E-03	3.7E-07	4.32E+2	1.1E-1	1.4E-09	6.89E+0	7.6E+0	1.4E-09	2.93E+0	2.6E+0	3.2E-09	2.50E-5	7.0E-6
f17	4.27E+06	1.8E+04	1.4E-09	3.47E+6	1.5E+5	1.4E-09	3.45E+6	1.7E+5	1.4E-09	1.99E+6	9.2E+4	3.5E-06	1.81E+6	2.8E+5
f18	9.69E+10	9.3E+09	1.4E-09	7.17E+4	1.8E+4	1.4E-09	3.97E+4	1.1E+4	1.4E-09	3.75E+4	1.0E+4	1.4E-09	3.71E+3	4.9E+3
f19	9.86E+06	1.7E+05	1.4E-09	1.27E+7	9.2E+5	2.0E-09	1.12E+7	5.6E+5	9.3E-09	1.15E+7	6.1E+5	4.1E-09	9.69E+6	6.8E+5
f20	8.22E+09	1.6E+09	1.4E-09	1.10E+4	1.6E+3	1.4E-09	7.32E+3	6.2E+2	1.4E-09	2.69E+3	3.6E+2	1.4E-09	1.06E+3	8.3E+1
	#(+/–/≈)=4/15/1			#(+/–/≈)=5/15/0			#(+/–/≈)=4/16/0			#(+/–/≈)=3/17/0				

表 4 结果 比较 CLLSO 与 另外四种 其他 算法在 CEC2013

	DECC-DG			SL-PSO			CSO			DLLSO			CLLSO	
	mean	std	p	mean	std	p	mean	std	p	mean	std	p	mean	std
f01	1.64E+6	3.0E+6	2.5E-10	4.81E+4	7.8E+4	2.5E-10	4.02E+2	6.8E+1	2.5E-10	4.99E-1	2.5E-1	2.5E-10	2.00E-6	0.0E+0
f02	1.26E+4	5.2E+2	1.4E-09	1.02E+4	4.0E+2	6.5E-02	9.82E+3	1.1E+2	1.8E-07	8.23E+3	4.3E+2	1.4E-09	1.04E+4	3.9E+2
f03	1.69E+1	3.1E-1	9.7E-11	2.10E+1	8.5E-2	9.7E-11	5.33E+0	5.1E+0	9.7E-11	3.49E-4	5.9E-5	9.7E-11	0.00E+0	0.0E+0
f04	2.5E+11	1.E+11	1.4E-09	3.7E+10	7.0E+9	1.4E-09	2.6E+10	4.4E+9	1.4E-09	2.6E+10	6.7E+9	1.4E-09	1.1E+10	1.7E+9
f05	6.97E+6	4.0E+5	5.2E-09	6.19E+6	1.8E+6	1.3E-04	8.98E+5	1.6E+5	1.4E-09	8.03E+5	1.3E+6	1.4E-09	7.93E+6	2.0E+5
f06	1.86E+4	3.1E+4	1.4E-09	1.06E+6	1.7E+3	1.4E-09	1.67E+1	4.9E+0	1.4E-09	4.86E+0	3.3E+0	1.4E-09	6.21E-3	2.0E-2
f07	1.31E+9	4.0E+8	1.4E-09	2.53E+9	8.6E+8	1.4E-09	4.05E+8	2.3E+8	1.4E-09	3.67E+7	1.3E+7	5.1E-06	1.74E+7	1.6E+7
f08	9.3E+15	5.E+15	1.4E-09	7.6E+15	1.E+15	1.4E-09	3.5E+15	5.E+14	1.4E-09	4.2E+14	9.E+13	1.4E-09	1.5E+14	3.E+13
f09	5.97E+8	3.5E+7	5.5E-02	5.95E+8	2.0E+8	9.8E-01	7.95E+7	1.8E+7	1.4E-09	5.59E+7	2.5E+7	1.4E-09	5.78E+8	3.3E+7
f10	2.37E+1	1.4E+1	1.4E-09	9.00E+7	1.9E+7	1.4E-09	8.46E+2	3.8E+2	1.4E-09	1.60E+2	3.8E+1	1.4E-09	6.69E-3	3.4E-3
f11	3.1E+11	1.E+11	1.4E-09	3.4E+11	6.E+10	1.4E-09	7.9E+10	4.E+10	1.4E-09	5.66E+9	5.2E+9	2.3E-09	1.10E+9	3.9E+8
f12	6.4E+11	4.E+10	1.4E-09	1.28E+4	2.9E+3	1.4E-09	7.65E+3	7.5E+2	1.4E-09	3.21E+3	1.3E+3	1.4E-09	1.16E+3	1.3E+2
f13	5.3E+10	8.9E+9	1.4E-09	3.3E+10	7.3E+9	1.4E-09	1.9E+10	3.8E+9	1.4E-09	3.25E+9	9.9E+8	2.4E-06	1.63E+9	8.8E+8
f14	9.9E+10	3.E+10	1.4E-09	5.4E+11	1.E+11	1.4E-09	3.2E+10	1.E+10	1.4E-09	8.41E+9	3.2E+9	1.4E-09	1.53E+8	7.6E+7
f15	2.78E+7	1.6E+7	1.9E-07	6.43E+7	1.2E+7	2.4E-01	9.51E+7	8.1E+6	5.2E-09	1.25E+8	1.6E+7	1.4E-09	6.66E+7	7.1E+6
	#(+/−/≈)=2/12/1			#(+/−/≈)=0/12/3			#(+/−/≈)=3/12/0			#(+/−/≈)=3/12/0				

大规模优化中的表现相媲美。竞争排名策略可以与弱中心化和基于等级的学习进化有效结合，从而克服数据敏感性和拓扑稀疏问题。不确定性检测策略可以检测等级不确定性较大的智能体，以减轻环境感知不确定性的影响。因此，CEC 仍具有令人满意的全局优化性能。

为研究 CEC 在分布式优化中的收敛性，以及传统进化计算算法在集中式大规模优化中的收敛性，我们在图10中以f01、f03、f11、f13为例。由于数量级过大，我们取基于e 的收敛适应度值的对数值（ln ($fitness$)）。可以发现，CLLSO 具有更快的收敛速度，并且找到的最佳解更优。首先，尽管 CEC中的智能体具有环境感知不确定性，但通过不确定性检测策略可以检测到这些不确定性，从而减轻其负面影响。同时，检测到的不可靠智能体会被移出众包人群，不再消耗评估次数。其次，通过竞争排序与基于水平的学习进化策略相结合，智能体能够在保持多样性的同时较好地搜索全局最优解。此外，表3、表4和图10中记录的结果也表明，CEC 确实具有良好的鲁棒性。尽管每个工作者的不确定性水平是固定的，但真实的不确定性值是

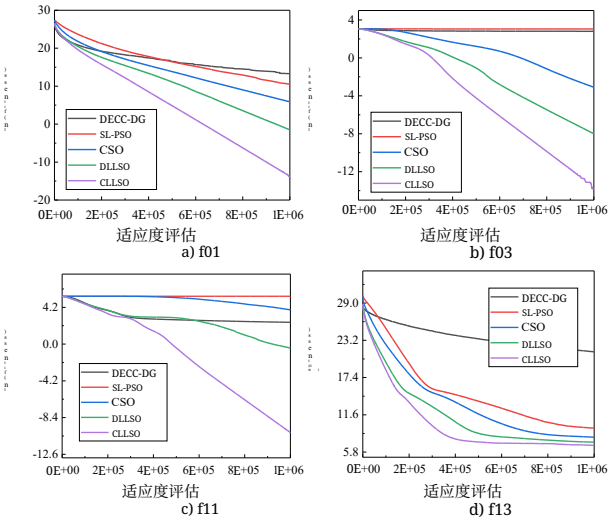


图10. 收敛速度比较，其中 a) f01，b) f03，c) f11，e) f13。

每次评估的不确定性水平均会有所变化。然而，CEC 不仅具有更优的适应度值，标准差也更小。这意味着该算法对不确定性的变化具有鲁棒性。因此，CEC 具有令人满意的全局优化能力和鲁棒性。

4.6 分布式聚类优化比较

在分布式传感器收集数据后，对无线传感器网络（WSNs）中的优化需求日益增加。因此，出现了许多亟待解决的新问题，而分布式聚类优化就是其中一个代表性问题。因此，我们以分布式聚类优化为实际示例，说明 CEC的应用。CEC 在场景分割、监测应用等方面具有广泛的应用[44]。

不失一般性，我们采用了来自 UCL 机器学习资料库（https://ar- chive.ics.uci.edu/ml/index.php）的 UrbanGB 数据集。该数据集包含英国各城市地区发生的 360,177 起道路事故的坐标，包括经度 [-5.55599、1. 75834] 和纬度 [50.0797、57.6956], 。这些坐标总共对应 469 个带标签的聚类中心。尽管数据集中已知道路事故的坐标，但我们可以将其视为由位于英国不同城市地区的不同传感工作者收集，并由一台云服务器协调所有工作者。在这种情况下，UrbanGB 中的聚类优化可视为一个基于众包的分布式优化问题，并用于测试 CEC 的性能。

由于计算负担较重，我们使用前10,000个坐标和出现频率最高的100个聚类中心作为测试样例。这些数据如图11所示，其中黑色圆圈表示坐标，红色星形表示聚类中心。

具体而言，决策变量向量是聚类中心的坐标。考虑到计算负担，我们还将待优化的聚类中心数量设为 100。目标是最小化簇内平方和（WCSS），其计算方法如下。

$$\min \sum_{j=1}^{k} \sum_{D_i \in j} \| D_i - C_j \|^2 \qquad (9)$$

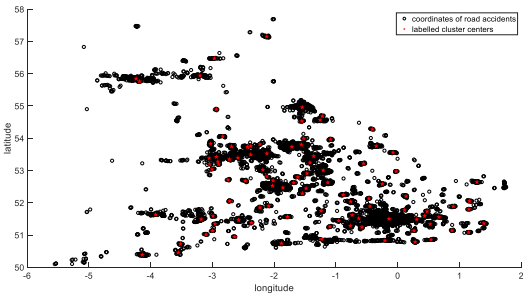


图 11. 10,000 个坐标及 100 个出现频率最高的标注聚类中心示意图。黑色圆圈表示道路事故的坐标，红色星形表示标注的聚类中心。

表 5 结果 对比：CLLSO 与 两种 分布式聚类 优化 算法

	DKM	MNID	CLLSO
均值	1033.154	1277.831	977.2407
std	26.32203	157.8187	14.39224
p	2.64E-09	9.73E-11	—

其中，k 表示簇的数量，在实验中设为 100。这意味着优化后的维数为 200，最大适应度评估次数为 200,000。C_j 表示第 j 个簇的坐标。D_i 是待聚类的坐标数据，$D_i \in j$ 表示 D_i 与中心 j的距离最近。出现频率最高的 100 个聚类中心的 WCSS 为 695.7181。

在分布式聚类优化中，有多种方法可以模拟环境感知不确定性。例如，每个工作者收集的数据各不相同、每个工作者的数据存在部分缺失、每个工作者的数据存在部分冗余等。在我们的实验中，我们采用替换方法：第 i 个个体有 i 个不确定点，并随机替换 i 个原始数据。为了展示 CEC 的应用前景，我们选取两种分布式聚类优化算法进行比较，分别是分布式 k-means 算法（DKM）[44] 和基于最小归一化信息距离的算法（MNID）[17]。我们按照其原始论文中的设置实现了这两种算法。比较结果记录在表 5 中。可以发现，CLLSO 的 WCSS 值更低。此外，在显著性水平为 0.05 时进行的 Wilcoxon 秩和检验表明，CLLSO 的表现显著优于其他参与比较的算法。因此，CEC 在工业界和学术界都具有广阔的应用前景。

5 结论

本文关注一种由众包和大数据时代推动的新兴分布式优化，即基于众包的分布式优化。针对其特点，本文将众包范式与进化计算相结合，提出了基于众包的进化计算（CEC）。通过竞争排序与不确定性检测的协同作用，CEC 能够有效缓解环境感知不确定性、中心化程度较弱、拓扑随时间变化以及数据敏感性等挑战。例如，本文实现了 CLLSO，并在基准函数和分布式聚类优化上开展了广泛实验。结果表明，CEC 在基于众包的分布式优化中具有令人满意的性能和广阔的应用前景。

对于未来的潜在研究，值得探索基于众包的高代价优化。大数据时代爆炸式增长的数据大幅增加了适应度评估的计算负担。如何在如此海量的数据中提高效率，对分布式优化具有重要意义。

致谢

本工作部分得到了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；部分得到了国家自然科学基金委员会（项目编号：61976093和61873097）的资助；部分得到了韩国国家研究基金会（项目编号：NRF-2021H1D3A2A01082705）的资助；部分得到了广东—深圳联合重点项目（项目编号：2022B1515120076）的资助；并且部分得到了广东省自然科学基金研究团队项目（项目编号：2018B030312003）的资助。(通讯作者：*Wei-Neng Chen*)

参考文献

[1] D. C. Brabham，“众包作为问题解决模型：简介与案例”，*Converg. Int. Res. into New Media Technol.*，第14卷，第1期，第75–90页，2008年，doi：10.1177/1354856507084420。[2] L. Wang 等人，“三维空间众包中的任务调度：社会福利视角”，*IEEE Trans. Mob. Comput.*，第14卷，第8期，第1–14页，2015年，doi：10.1109/tmc.2022.3175305。[3] S. S. Bhatti、J. Fan、K. Wang、X. Gao、F. Wu和G. Chen，“空间众包中有界任务分配问题的近似算法”，*IEEE Trans. Mob. Comput.*，第20卷，第8期，第2536–2549页，2021年，doi：10.1109/TMC.2020.2984380。[4] Y. Xu、M. Xiao、J. Wu、S. Zhang和G. Gao，“具有未知社会感知特征工作者的空间众包激励机制：三阶段Stackelberg博弈方法”，*IEEE Trans. Mob. Comput.*，第1–14页，2022年，doi：10.1109/TMC.2022.3157687。[5] W. Wang 等人，“战略性社会团队众包：为社交网络中的众包组建诚实工作者团队”，*IEEE Trans. Mob. Comput.*，第18卷，第6期，第1419–1432页，2019年，doi：10.1109/TMC.2018.2860978。[6] J. Tan、H. Wu、K.-H. Chow和S.-H. G. Chan，“用于联合射频与地磁指纹识别的隐式多模态众包”，*IEEE Trans. Mob. Comput.*，第1–14页，2021年，doi：10.1109/tmc.2021.3088268。[7] B. Zhao、X. Liu、W. N. Chen、W. Liang、X. Zhang和R. H. Deng，“PRICE：面向隐私与可靠性的实时众包感知激励系统”，*IEEE Internet Things J.*，第8卷，第24期，第17584–17595页，2021年，doi：10.1109/JIOT.2021.3081596。[8] B. Zhao、X. Liu、W. N. Chen和R. Deng，“CrowdFL：通过联邦学习实现隐私保护的移动群智感知系统”，*IEEE Trans. Mob. Comput.*，第1–13页，2022年，doi：10.1109/TMC.2022.3157603。[9] T. Kohler和M. Nickel，“持久的众包商业模式”，*J. Bus. Strategy*，第38卷，第2期，第25–32页，2017年1月，doi：10.1108/JBS-10-2016-0120。[10] G. D. Saxton、O. Oh和R. Kishore，“众包规则：模型、问题与控制系统”，*Inf. Syst. Manag.*，第30卷，第1期，第2–20页，2013年，doi：10.1080/10580530.2013.739883。[11] L. Da Xu、W. He和S. Li，“工业物联网：综述”，*IEEE Trans. Ind. Informatics*，第10卷，2243，2014年，4，2233 10 1109 2014 2300753 no. pp. – doi: . /TII. . . [12] C. Roy、S. Swarup Rautaray和M. Pandey，“大数据优化技术：综述”，*Int. J. Inf. Eng. Electron. Bus.*，第10卷，第4期，第41–48页，2018年，doi：10.5815/ijieeb.2018.04.06。[13] S. Deng、H. Zhao、W. Fang、J. Yin、S. Dustdar和A. Y. Zomaya，

边缘智能：边缘计算与人工智能的融合”，IEEE Internet Things J.，第7卷，第8期，第7457–7469页，2020年，doi：10.1109/JIOT.2020.2984887。[14] B. Guo等人，“移动群智感知与计算：新兴人力驱动感知范式综述”，ACM Comput. Surv.，第48卷，第1期，第7页，2015年。[15] T. Yang等人，“分布式优化综述”，Annu. Rev. Control，第47卷，第278–305页，2019年，doi：10.1016/j.arcontrol.2019.05.006。[16] Q. Yang、Y. Liu、T. Chen和Y. Tong，“联邦机器学习：概念与应用”，ACM Trans. Intell. Syst. Technol.，第10卷，第2期，第1–19页，2019年，doi：10.1145/3298981。[17] J. Qin、Y. Zhu和W. Fu，“基于归一化信息度量的传感器网络分布式聚类算法”，IEEE Trans. Signal Process.，第68卷，第3266–3279页，2020年，doi：10.1109/TSP.2020.2995506。[18] B. Lorenzo和S. Glisic，“使用遗传算法的多跳蜂窝网络最优路由与流量调度”，IEEE Trans. Mob. Comput.，第12卷，第11期，第2274–2288页，2013年，doi：10.1109/TMC.2012.204。[19] D. Whitley，“进化算法概述：实践问题与常见陷阱”，Inf. Softw. Technol.，第43卷，第14期，第817–831页，2001年，doi：10.1016/S0950-5849(01)00188-4。[20] H. Yang、F. Ye和B. Sikdar，“一种基于群体智能的移动汇聚节点网络数据采集协议”，IEEE Trans. Mob. Comput.，第7卷，第8期，第931–945页，2008年，doi：10.1109/TMC.2007.70783。[21] A. Chakraborty和A. K. Kar，“群体智能：算法综述”，载于Modeling and Optimization in Science and Technologies，第10卷，2017年，第475–494页。[22] Y. Jin、H. Wang、T. Chugh、D. Guo和K. Miettinen，“数据驱动的进化优化：概述与案例研究”，IEEE Trans. Evol. Comput.，第23卷，458，2019年，3，442 10 1109 2018 2869001 no. pp. – doi: . /TEVC. . . [23] B. Xue、M. Zhang、W. N. Browne和X. Yao，“进化计算特征选择方法综述”，IEEE Trans. Evol. Comput.，第20卷，第4期，第606–626页，2016年，doi：10.1109/TEVC.2015.2504420。[24] 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 Comput. Intell. Mag.，第16卷，第1期，第22–33页，2021年，doi：10.1109/MCI.2020.3039066。[25] J. Xu、Y. Jin、W. Du和S. Gu，“联邦数据驱动进化算法”，Knowledge-Based Syst.，第233卷，文章107532，2021年，doi：10.1016/j.knosys.2021.107532。[26] J. Xu、Y. Jin和W. Du，“用于昂贵多目标/多目标优化的联邦数据驱动进化算法”，Complex Intell. Syst.，第7卷，第6期，第3093–3109页，2021年，doi：10.1007/s40747-021-00506-7。[27] S. Liu、H. Wang、W. Peng和W. Yao，“一种采用并行随机分组的代理辅助进化特征选择算法，用于高维分类”，IEEE Trans. Evol. Comput.，第1–15页，2022年，doi：10.1109/TEVC.2022.3149601。[28] F.-F. Wei、W.-N. Chen、Q. Li、S.-W. Jeon和J. Zhang，“采用按需评估的分布式昂贵进化约束优化”，IEEE Trans. Evol. Comput.，第1–1页，2022年，doi：10.1109/tevc.2022.3177936。[29] 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 Trans. Cybern.，第1–14页，2022年。[30] A. Song、W. Chen、Y. Gong、X. Luo和J. Zhang，“一种用于大规模虚拟网络嵌入的分治进化算法”，IEEE Trans. Evol. Comput.，第24卷，第3期，第566–580页，2020年，doi：10.1109/TEVC.2019.2941824。[31] Y. J. Gong等人，“分布式进化算法及其模型：最新研究综述”，Appl. Soft Comput. J.，第34卷，第286–300页，2015年，doi：10.1016/j.asoc.2015.04.061。[32] B. Zhao、S. Tang、X. Liu、X. Zhang和W. N. Chen，“IronM：面向移动群智感知的异构数据隐私保护可靠性估计”，IEEE Internet Things J.，第7卷，第6期，第5159–5170页，2020年，doi：10.1109/JIOT.2020.2975546。[33] Q. Yang、W. N. Chen、J. Da Deng、Y. Li、T. Gu和J. Zhang，“一种面向大规模优化的基于等级学习的群体优化器”，IEEE Trans. Evol. Comput.，第22卷，第4期，第578–594页，2018年，doi：10.1109/TEVC.2017.2743016。[34] X. Chao、G. Kou、T. Li和Y. Peng，“姬克对战AlphaGo：一种利用决策方法对不完备信息大规模数据进行排名的方法”，Eur. J. Oper. Res.，第265卷，第1期，第239–247页，2018年，doi：https://doi.org/10.1016/j.ejor.2017.07.030. [35] Y. M. Wang、Z. P. Fan 和 Z. Hua，《一种从乘法型和模糊偏好关系中获取优先级向量的卡方方法》，《欧洲运筹学杂志》，第 182 卷，第 1 期，第 356–366 页，2007 年，doi：10.1016/j.ejor.2006.07.020. [36] T. Ke、L. Xiaodong、S. P. N.、Y. Zhenyu 和 W. Thomas，《CEC’2010 大规模全局优化特别专题与竞争的基准函数》，2010 年。[在线]。网址：http://goanna.cs.rmit.edu.au/~xiaodong/cec13-lsgo/competition/cec2013-lsgo-benchmark-tech-report.pdf。[37] L. Xiaodong、T. Ke、S. P. N.、Y. Zhenyu 和 K. Qin，《CEC’2013 大规模全局优化特别专题与竞争的基准函数》，2013 年。 [在线]。网址：http://goanna.cs.rmit.edu.au/~xiaodong/cec13-lsgo/competition/cec2013-lsgo-benchmark-tech-report.pdf。[38] F.-F. Wei 等，《一种用于高代价优化的分类器辅助基于等级学习的群体优化器》，*IEEE* 进化计算汇刊，第 25 卷，第 2 期，第 219–233 页，2021 年，doi：10.1109/tevc.2020.3017865。[39] Z. Tang、M. Gong、Y. Xie、H. Li 和 A. K. Qin，《采用动态邻居和基于等级的任务间学习的多任务粒子群优化》，*IEEE* 新兴计算智能主题汇刊，第 6 卷，第 2 期，第 300–314 页，2022 年，doi：10.1109/TETCI.2021.3051970。[40] J. Guo、X. Li、Z. Lao、Y. Luo、J. Wu 和 S. Zhang，《采用基于等级学习的群体优化器的极限学习机对工业机器人减速器进行故障诊断》，先进机械工程，第 13 卷，第 5 期，第 1–10 页，2021 年，doi：10.1177/16878140211019540。[41] R. Cheng 和 Y. Jin，《一种可扩展优化的社会学习粒子群优化算法》，信息科学（纽约），第 291 卷，C 期，第 43–60 页，2015 年，doi：10.1016/j.ins.2014.08.039。[42] R. Cheng 和 Y. Jin，《一种面向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。[43] M. N. Omidvar、Y. Ling、Y. Mei 和 X. Yao，《采用差分分组的大规模全局优化协同协进化》，*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。[44] J. Qin、W. Fu、H. Gao 和 W. X. Zheng，《基于多智能体共识理论的传感器网络分布式 k 均值算法与模糊 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。



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，她是华南理工大学计算机科学与工程学院及中国广州琶洲实验室的博士生。她目前的研究兴趣是包括群体智能、进化计算及其在现实问题中昂贵且分布式优化中的应用。



Wei-Neng Chen（IEEE 高级会员）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教授，目前还担任广州琶洲实验室教授，中国。他合著了100多篇国际期刊和会议论文，其中50多篇发表于IEEE 汇刊。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。Chen 博士于2016年获得 IEEE 计算智能学会（CIS）杰出博士论文奖，并于同年获得国家优秀青年科学基金资助。他现任 IEEE 广州分会副主席，也是 IEEE CIS 新兴主题工作组委员。他担任 IEEE TRANSACTIONSON NEURAL NETWORKS AND LEARNING SYSTEMS和复杂与智能系统的副编辑。



Xiao-Qi Guo于2018年获得中国广州华南理工大学计算机科学与技术学士学位，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及其在分布式网络中的应用。



Bowen Zhao（IEEE会员）于2020年获得中国华南理工大学网络空间安全博士学位。现任广州理工学院、西安电子科技大学广州校区副教授。他目前的研究兴趣包括隐私保护计算与学习，以及隐私保护移动群智感知。



Sang-Woon Jeon（IEEE会员）于2003年和2006年分别获得韩国首尔延世大学电气工程学士学位和硕士学位，并于2011年获得韩国大田韩国科学技术院（KAIST）电气工程博士学位。他现任军事信息系副教授。

工程学（本科院校）和电子与通信工程系（研究生院），韩国安山汉阳大学，自2017年起。2011年至2013年，他在瑞士洛桑的洛桑联邦理工学院计算机与通信科学学院担任博士后研究员。

2013年至2017年，他担任韩国安东国立大学信息与通信工程系助理教授。他的研究兴趣包括网络信息论、无线通信、传感器网络，以及这些领域在物联网和大数据中的应用。Jeon博士于2017年获得海东青年学者奖，该奖由海东基金会赞助，并由韩国通信与信息科学学会（KICS）颁发；他还获得了2016年KICS期刊最佳论文奖、2015年IEEE国际通信会议最佳论文奖、2012年KAIST电气工程学部最佳学位论文奖、2010年KICS夏季会议最佳论文奖，以及2009年三星Humantech论文奖铜奖。



Jun Zhang（IEEE会士）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学部访问教授。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200余篇技术论文。Zhang博士

他于2011年获得国家自然科学基金委员会颁发的国家自然科学基金优秀青年科学家基金，并于2009年获得中国教育部颁发的自然科学研究一等奖。他目前担任 IEEE 《进化计算 汇刊》以及 IEEE《控制论 汇刊》的副编辑。