



研究与开发

面向多元可控负荷调控的云边协同负荷资源分配策略

李思维^{1,2}, 靳莉², 于龙², 杜立石², 岳靓², 张喜润²

(1. 天津大学智能电网教育部重点实验室, 天津 300072;

2. 北京中电飞华通信有限公司, 北京 100071)

摘要: 针对多元可控负荷资源进行可控负荷管理时需要占用大量计算资源, 且无法实现自动功率精准控制的问题, 提出了一种面向多元可控负荷调控的云边协同负荷资源分配策略。首先, 设计了云边协同调控架构, 整合处理各种多元可控负荷资源数据; 其次, 考虑不同边缘节点计算任务的相似性, 以所有计算任务的时间开销最小为优化目标, 给出云端计算资源分配策略, 合理分配计算资源; 最后, 通过基于自适应交叉-变异概率的遗传算法进行计算资源分配的求解。实验结果表明, 所提算法在任务完成时间和执行成本上具有较为明显的优势, 并且任务数量越多, 计算资源越小时优势越明显, 可以显著提升计算效率, 降低计算耗时。

关键词: 负荷管理; 多元可控负荷; 资源分配; 云边协同

中图分类号: TM734

文献标志码: A

doi: 10.11959/j.issn.1000-0801.2024192

Adaptive distributed cloud edge collaborative load control strategy for load management

LI Siwei^{1,2}, JIN Li², YU Long², DU Lishi², YUE Liang², ZHANG Xirun²

1. Key Laboratory of Smart Grid of Ministry of Education, Tianjin University, Tianjin 300072, China

2. Beijing FibrLINK Communications Co., Ltd., Beijing 100071, China

Abstract: To solve the problem that the controllable load management of multiple controllable load resources requires a lot of computing resources and can not achieve accurate automatic power control, a cloud-edge cooperative load resource allocation strategy for multiple controllable load regulation was proposed. Firstly, the collaborative control framework of cloud edge was designed to integrate and process the data of various controllable load resources. Secondly, considering the similarity of computing tasks of different edge nodes, the optimization goal was to minimize the time cost of all computing tasks, and the cloud computing resource allocation strategy was given to allocate computing resources reasonably. Finally, the computational resource allocation was solved by genetic algorithm based on adaptive cross-mutation probability. Finally, the calculation of resource allocation was solved using a genetic algorithm based on adaptive crossover mutation probability. The experimental results show that the algorithm proposed

收稿日期: 2024-02-28; 修回日期: 2024-07-19

通信作者: 张喜润, xirunzhang@163.com

基金项目: 国家电网公司总部科技项目 (No.5400-202320223A-1-1-ZN)

Foundation Item: Science and Technology Project of State Grid Corporation of China (No.5400-202320223A-1-1-ZN)

has significant advantages in task completion time and execution cost, and these advantages become more pronounced as the number of tasks increases and computing resources decrease. It can significantly improve computing efficiency and reduce computing time.

Key words: load management, multiple controllable load, resource allocation, cloud edge collaboration

0 引言

随着传统电力系统向新型电力系统转型升级,新能源汽车、新型储能、虚拟电厂等新型主体不断涌现,电源结构、负荷特性、平衡模式、市场环境等各方面都将迎来深刻变化,电力供应保障面临极大挑战。电力负荷管理成为新型电力系统管理的重要组成部分,对保障电力系统的安全稳定运行、提高能源利用效率以及推动可持续发展具有重要意义^[1-3]。电力系统中的各种可调节、可控制的负荷资源,不仅包括工业负荷、商业负荷、居民负荷等传统负荷,还包括电动汽车、电采暖设备、空调热泵等新型可控负荷,这些资源具有不同的特性,如响应时间、调节范围、调节精度等,可以为电力系统的负荷管理提供有力支持^[4-5]。对电动汽车、电采暖设备、空调热泵等新型可控负荷实施灵活的管理控制,能够有效提高电力系统的运行韧性与经济性。然而,有别于传统储能电站、工厂、商超等大容量可控负荷,新型负荷的容量普遍较小、分布分散、数量庞大,对如此海量的负荷进行采集、分析、建模与优化并实施管理无疑存在数据体量庞大、计算任务繁重、资源分配不均衡等问题^[6-7]。具体而言,在对多元可控负荷资源进行可控负荷管理时需要占用大量计算资源,传统的计算资源分配方法忽略了可控负荷边缘节点数据特性,从而导致相似类型的可控负荷边缘节点需要反复分配计算资源,使运行效率低下,资源利用不均衡,无法在有限的计算资源下完成边缘节点数据分析任务^[8];其次,现有方法在对多元可控负荷资源实施负荷管理时缺乏针对性,无法实现自动

功率精准控制^[9]。

利用云计算、边缘计算等先进信息技术,可实现电力系统中的负荷资源在云平台上的协同调控^[10-11]。利用云计算强大的数据处理能力和资源调度能力,可以对电力系统中的企业、行业用电数据、储能数据、分布式光伏数据等海量数据进行分析 and 处理,给出电力负荷精细化预测、源荷供需互动等业务的数据支撑,为电力负荷管理提供决策依据。利用边缘计算设备可实现负荷资源的精细化管理及多元化互动服务。因此,针对多元可控负荷资源进行可控负荷管理时需要占用大量计算资源,且无法实现自动功率精准控制的问题,提出一种面向多元可控负荷调控的云边协同负荷资源分配策略。

1 云边协同调控架构

为了实现对分散多元可控负荷资源的优化管理与高效利用,结合云计算与边缘计算等先进的信息技术,设计了面向多元可控负荷调控的云边协同调控系统,其框架如图1所示。系统的核心理念是通过智能化的监测和控制手段,整合处理各种多元可控负荷资源数据(如电动汽车、空调、分散式电采暖设备、热泵负荷等),使这些可控负荷资源能够以最优方式分配供给^[12],实现对多元可控负荷资源的实时监测、合理调度和控制。

云边协同调控系统架构包含了“端、边、云”3层:终端层通过各类型的传感器、计量表计等实现多元负荷运行状态的可靠感知,是云端控制中心的主要数据来源;边缘层负责对多元负荷进行直接管理,一方面从终端层汇集多元负荷

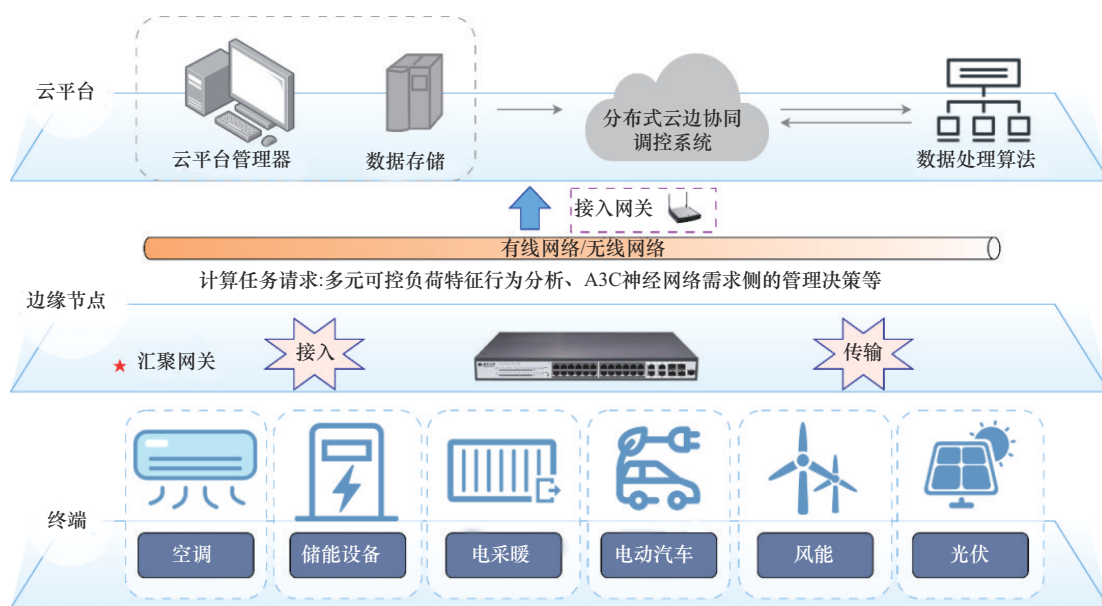


图1 面向多元可控负荷调控的云边协同调控系统框架

的运行状态并上传至云端控制中心，另一方面接收来自云端控制中心的调度指令对多元负荷实施管理控制；云端控制中心则根据收到的多元负荷运行状态数据进行面向多元用户行为特征的云平台分布式精准调控，并向边缘层的任务需求分配计算资源以满足云边协同最优负荷管理调度^[13]。

云端控制中心依托云平台管理器（cloud manager, CM），负责整个多元可控负荷资源调控系统的监控、管理和协调^[14-15]，主要功能包括计算资源管理与分配、边缘节点间协同优化运行、云边协同负荷调控系统监控和优化等功能。CM通过上述功能，实现对整个云边协同负荷调控系统的全面管控，使系统能够高效运行、计算资源得以合理分配利用、可控负荷边缘节点的任务请求能够顺利执行，同时也提供了良好的用户体验和安全保障。

边缘设备在对电动汽车、空调、分散式电采暖设备、热泵负荷等可控负荷设备进行本地化调控时，利用其具备的数据分析处理能力，承担数据收集、数据预处理、实时负荷控制和安全保障等重要任务。由于边缘设备更加靠近负荷侧，因此与传统的集中式负荷控制方案相比具有更低的

数据交互量、更低的控制维度和更强的可扩展性能，从而有效提升电力系统的运行韧性与灵活性。

2 云平台计算资源优化分配策略

云边协同负荷调控系统中，多元可控负荷特征行为分析、可调功率计算都将产生计算需求。在云边协同的多元可控负荷调控体系下，CM需要精准、高效地管理和调度计算资源，快速响应边缘节点所发起的差异化计算请求。然而，当不同的边缘节点在同一时间并发地提起计算资源请求时，现有的云端计算资源调度方法大多为每个边端单独分配计算资源，忽略了不同边缘节点计算任务间的相似性，导致CM在分配计算资源时需要频繁地、重复地分配相近的计算资源，因而分配时效性较低。事实上，不同边缘节点下所管辖的部分可控负荷类型可能是一致的，不同边缘节点下对于同一类可控负荷的特征分析可以采用同一个模型。CM可以仅使用一个边缘节点上传的该类可控负荷状态数据进行特征分析模型的训练，并将该模型用于其他边缘节点下同一类型可控负荷的特征分析，可有效节约宝贵的云端计算

资源并提升计算时效性。

2.1 云端计算资源优化分配目标

CM在响应来自边缘节点发起的计算任务请求时,存在三大特征,即边缘计算需求的多样性、计算资源分配的灵活性及效率、可扩展性。当用户行为特征发生变化或新增不同类型的可控负荷用户时,云边协同负荷调控系统能够灵活适应并调整计算资源分配策略。

CM在收集汇总各边缘节点在同一时刻并发请求的计算任务后,在考虑边缘节点最大任务数目约束和任务最大计算资源分配约束的基础上,根据计算任务相似度将计算任务分为若干类;其次,针对重新分组后的计算任务的资源优化分配目标,采用遗传算法进行求解,以所有计算任务的时间开销最小为优化目标合理分配计算资源。其中,考虑传统遗传算法的种群更新策略随机性较强,容易陷入局部最优,且收敛速度较慢,本文给出了基于改进遗传算法的优化算法。具体如下。

设 $E=\{e_k|1\leq k\leq N_E, N_E\in\mathbf{Z}^+\}$ 表示各多元可控负荷边缘节点编号,其中, e_k 为第 k 个边缘节点, N_E 为边缘节点总数。各边缘节点向CM发起计算任务,从而设 $W=\{w_{i,e_k}|1\leq i\leq N_{W,e_k}, N_{W,e_k}\in\mathbf{Z}^+, e_k\in E\}$ 表示参考各边缘节点计算任务编号,其中 w_{i,e_k} 为 e_k 节点的第 i 个计算任务编号, N_{W,e_k} 为 e_k 节点的计算任务总数。令CM为每个计算任务所提供的计算资源为 $S_{w_{i,e_k}}$ 。鉴于计算任务的时效性要求,本文设定云端资源优化分配目标为最小化所有计算任务的时间开销,即:

$$\min T_{\Sigma} = \sum_{k=1}^{N_E} \sum_{i=1}^{N_{W,e_k}} \left(\frac{F_{w_{i,e_k}}}{S_{w_{i,e_k}}} + \frac{F_{w_{i,e_k}}}{V_{e_k}^U} + \frac{R_{w_{i,e_k}}}{V_{e_k}^D} \right) \quad (1)$$

其中, $F_{w_{i,e_k}}$ 为每个计算任务的数据规模, $R_{w_{i,e_k}}$ 为计算任务处理完成后返回的数据量, $V_{e_k}^U$ 、 $V_{e_k}^D$ 分别为边缘节点 e_k 与云管理器间的上、下行信道速率, T_{Σ} 为所有计算任务的时间开销。

2.2 云端计算资源优化约束条件

(1) 计算任务时效性约束

为了保持云边协同负荷调控系统的负载均衡性,单个边缘节点过多的任务请求可能导致云服务器无法及时响应边缘节点,从而降低云服务器的响应速度,甚至造成整个云边协同负荷调控系统瘫痪。所以,为了保证系统的安全性、实时性和响应性,计算任务需要在规定的时间内完成。计算时间主要包括数据传输时间和计算任务处理时间,即:

$$\frac{F_{w_{i,e_k}}}{S_{w_{i,e_k}}} + \frac{F_{w_{i,e_k}}}{V_{e_k}^U} + \frac{R_{w_{i,e_k}}}{V_{e_k}^D} \leq T_{w_{i,e_k}}^{\max}, \quad (2)$$

$$k=1,2,\dots,N_E, i=1,2,\dots,N_{W,e_k}$$

其中, $T_{w_{i,e_k}}^{\max}$ 为第 k 个边缘节点的第 i 个计算任务所允许的最大计算时间。

(2) 任务最大计算资源分配约束

为了保证云边协同负荷调控系统的稳定性和安全性,设定最大计算资源分配限制可以防止某个边缘节点滥用系统资源。为每个边缘节点设定最大计算资源分配限制,可以避免激烈的资源竞争,系统可以更好地控制任务的执行时间和性能,可以确保每个任务都能够在合理的时间内完成,提高服务质量,故任务最大计算资源分配约束如下。

$$S_{w_{i,e_k}} \leq S_{\max} \quad (3)$$

其中, $S_{\max}$ 为单个任务所消耗的计算资源上限。

(3) CM承载力约束

CM的计算资源是有限的,因此,对于所有并发到达的计算任务而言,其资源分配总额满足上限约束:

$$\sum_{k=1}^{N_E} \sum_{i=1}^{N_{W,e_k}} S_{w_{i,e_k}} \leq S_{\max} \quad (4)$$

2.3 云端计算资源分配求解算法

根据优化目标,计算资源优化调度问题为典型的放置问题^[16]。当计算任务并发数量较大时,直接使用搜索树^[17]或状态空间方法^[18]进行求解存在计



算耗时过长的问题,不利于边端请求计算资源的快速分配与相应计算任务的高效求解。而现有启发式求解(如遗传算法^[19]、粒子群算法^[20]等)则存在易陷入局部最优、收敛速度过慢的问题。因此,本部分给出一种基于自适应交叉—变异概率的遗传算法进行计算资源分配的求解算法。不失一般性,为了简化问题求解难度,首先作如下合理假设。

- 为了简化问题、降低复杂性并使系统设计更加可控,本文假定云服务器总计算资源 S 为定值,使云边协同负荷调控系统可以更容易地对资源进行静态分配和规划,无须频繁地调整资源配置,提高系统的可维护性和可管理性。
- 为了简化系统设计、保证系统的稳定性以及方便资源管理,忽略节点故障、网络波动或移动边缘设备的参与等干扰因素,假设边缘节点固定不变。
- 在执行计算资源分配前,CM采用轮询或边缘节点主动上报的方式获取当前的可控负荷类型、计算任务类型、计算任务规模。相较于多元可控负荷状态数据的上传、计算任务处理和计算结果返回过程中产生的时延,轮询和上报产生的时延忽略不计。

2.3.1 计算任务重分组

CM根据可控负荷类型和计算任务类型,将计算任务 $W=\{w_i|1\leq i\leq N_{w,e_k}, N_{w,e_k}\in\mathbf{Z}^+\}$ 重新分为若干组别:

$$\{W_1, W_2, \dots, W_h\} \subseteq W \quad (5)$$

其中, $W_h = \{w_{j,h}|1\leq j\leq N'_{w,e_k}, h=1,2,\dots,H\}$, 其中 $w_{j,h}$ 为第 h 组的第 j 个计算任务, N'_{w,e_k} 为重新分组后 e_k 节点的计算任务总数, H 为组别数。显然,重组后每个计算任务子集合内的任务数量满

$$N'_{w,e_k} \leq \sum_{k=1}^{N_E} N_{w,e_k}.$$

由于分组后的每个子集合内任务具有相似性,因此本文采用数据规模最大的边缘节点的数据上传至CM进行任务计算,计算后的结果反馈至其他具有相同负荷类型和相同计算任务的边缘节点,从而有效降低数据交互过程开展的通信开销和计算开销。因此,目标函数式(1)可以进一步地转化为:

$$\min T_{\Sigma} = \sum_{h=1}^H \left(\frac{\max\{F_{w_{j,h}}\}}{S_{W_h}} + \frac{\max\{F_{w_{j,h}}\}}{V_{\max\{F_{w_{j,h}}\}}^U} \right) + \sum_{k=1}^{N_E} \sum_{i=1}^{N_{w,e_k}} \left(\frac{R_{W_h}}{V_{e_k}^D} \right) \quad (6)$$

其中, S_{W_h} 为CM第 h 组计算任务分配的计算资源, $V_{\max\{F_{w_{j,h}}\}}^U$ 为分组后计算任务子集合 W_h 中具有最大数据规模的边缘计算节点与CM间的上行信道速率, R_{W_h} 为第 h 组计算任务处理完成后返回的数据量。

2.3.2 染色体编码与初始种群生成

在准备好基于任务相似度的重新分组任务后,利用改进的遗传算法对目标函数进行求解。本步骤中,初始解以服从 $(0, S_{\max})$ 上均匀分布的方式创建初始种群,即初始解集合为:

$$\{S_{W_h}^{(m)}(1), S_{W_h}^{(m)}(1), \dots, S_{W_h}^{(m)}\}, m=1,2,\dots,M \quad (7)$$

其中, M 为种群数量。

遗传算法中,初始解使用染色体编码的方式进行等效表示,编码方式采用整数编码,编码过程是将问题的解转换成染色体的过程,可根据问题的特性进行定制,具体编码流程为将问题的解表示为整数数组,其中每个整数对应于解的一个部分或特征,每个整数可以表示一个任务的起始时间,详见文献[21]。每条染色体表示计算资源的可能分配量。

2.3.3 种群适应度评估

针对种群中的个体都需要计算其适应度这一问题,本文采用目标函数(式(1))的倒数作

为适应度函数, 即:

$$g^{(m)}(k) = \frac{1}{T_{\Sigma}^{(m)}(k)} \quad (8)$$

其中, $k \in \mathbb{Z}^+$ 为迭代次数。显然, 较优秀的个体具有更大的适应度值, 按照降序对适应度函数值进行排序, 从而当前迭代过程中整体种群的适应度值为 $J(k) = \max\{g^{(1)}(k), g^{(2)}(k), \dots, g^{(m)}(k)\}$ 。

2.3.4 种群生成与精英选择

遗传算法模拟生物进化“适者生存”的过程。由于适应度更高代表该计算资源分配方案能够更快地响应边缘节点的计算任务请求, 因此, 该个体所蕴含的基因应当有更高的概率被保留下来。本文采用轮盘赌方案生成新种群, 采用该个体适应度函数在整个种群中的累计概率决定其在下一代种群中出现的概率。累计概率计算如下。

$$P_m = \frac{\sum_{h=1}^m g^{(h)}(k)}{\sum_{m=1}^M g^{(m)}(k)} \quad (9)$$

此外, 为避免轮盘赌生成新种群时适应度最高的精英种群被遗失而影响收敛速度, 在每一代

的选择过程中将该最优个体保留至新种群中。从而在每一代生成的新种群为 $\{S_{W_h}^{(m)}(k), S_{W_h}^{(m)}(k), \dots, S_{W_h}^{(m)}(k)\}$ 。

2.3.5 种群中染色体自适应交叉与变异

染色体交叉是通过将两个或多个父代个体的信息融合生成新的子代个体。交叉的目的是保留父代个体的优良特征, 同时引入新的变化, 以希望在搜索空间中找到更优解; 染色体变异是通过引入新的基因将染色体上的编码信息进行低概率的修改, 从而跳出局部最优陷阱, 进一步提升遗传算法的全局寻优能力。

传统的染色体交叉与变异方案采用固定概率实现。在算法寻优前期, 为了实现算法的快速收敛性, 交叉与变异概率应适当偏大, 以增加基因的多样性从而快速定位可行解空间; 而在算法收敛后期, 交叉与变异概率应适当偏小以避免跳出最优解, 即找不到所有计算任务的时间开销最小的最优计算资源分配方案。因此, 本文提出一种基于 $\tanh$ 函数的交叉和变异概率自适应调整机制, 如下。

$$C^{(m)}(k) = \begin{cases} (C_{\max} - C_{\min}) \times \tanh\left[\beta \left(\frac{g^{(m)}(k) - g_{\text{avg}}(k)}{g_{\max}(k) - g_{\text{avg}}(k)}\right)\right] + C_{\min}, & g^{(m)}(k) \geq g_{\text{avg}}(k) \\ C_{\max}, & g^{(m)}(k) < g_{\text{avg}}(k) \end{cases} \quad (10)$$

$$M^{(m)}(k) = \begin{cases} (M_{\max} - M_{\min}) \times \tanh\left[\beta \left(\frac{g^{(m)}(k) - g_{\text{avg}}(k)}{g_{\max}(k) - g_{\text{avg}}(k)}\right)\right] + M_{\min}, & g^{(m)}(k) \geq g_{\text{avg}}(k) \\ M_{\max}, & g^{(m)}(k) < g_{\text{avg}}(k) \end{cases} \quad (11)$$

$$\tanh\left[\beta \left(\frac{g^{(m)}(k) - g_{\text{avg}}(k)}{g_{\max}(k) - g_{\text{avg}}(k)}\right)\right] = \frac{\exp\left[\beta \left(\frac{g^{(m)}(k) - g_{\text{avg}}(k)}{g_{\max}(k) - g_{\text{avg}}(k)}\right)\right] - \exp\left[-\beta \left(\frac{g^{(m)}(k) - g_{\text{avg}}(k)}{g_{\max}(k) - g_{\text{avg}}(k)}\right)\right]}{\exp\left[\beta \left(\frac{g^{(m)}(k) - g_{\text{avg}}(k)}{g_{\max}(k) - g_{\text{avg}}(k)}\right)\right] + \exp\left[-\beta \left(\frac{g^{(m)}(k) - g_{\text{avg}}(k)}{g_{\max}(k) - g_{\text{avg}}(k)}\right)\right]} \quad (12)$$

其中, $\tanh(\cdot)$ 如式 (12) 所示, $g^{(m)}(k)$ 为个体在交叉和变异过程中的适应度函数, $g_{\max}(k)$ 为最大适应度值, $g_{\text{avg}}(k)$ 为平均适应度值, $M_{\max}$ 和 $M_{\min}$ 分别为交叉函数的上、下界, $C_{\max}$ 和 $C_{\min}$ 分别为

变异函数的上、下界, β 为自适应参数。

$\tanh$ 函数曲线如图 2 所示, 可见 $\tanh$ 曲线的左侧和右侧趋于 1, 由此可以得到所提算法相较于传统固定交叉与变异概率方案的 3 个优势, 具



体如下。

(1) 根据式 (10) 和式 (11)，当个体的适应度超过整个种群的平均适应度时，交叉概率与变异概率趋于 $C_{\min}$ 和 $M_{\min}$ ，即优秀个体的基因有更大的概率被保留，从而收敛速度加快；而个体的适应度小于整个种群的平均适应度时，交叉概率与变异为 $C_{\max}$ 和 $M_{\max}$ ，从而增加基因多样性以提高全局寻优能力。

(2) 随着算法的迭代，种群的整体适应度函数将越来越均衡。根据 $\tanh$ 函数的形式，算法前期的交叉与变异概率将大于算法后期这二者的值，从而实现了算法前期多样性诉求和算法后期精确定位最优解的目的。

(3) 根据图2， β 决定了交叉与变异概率的变化速率，从而可以有效提升算法设计的灵活性，在算法迭代过程中可以进一步选择不同的 β 值以改善算法的全局寻优能力与收敛速度。

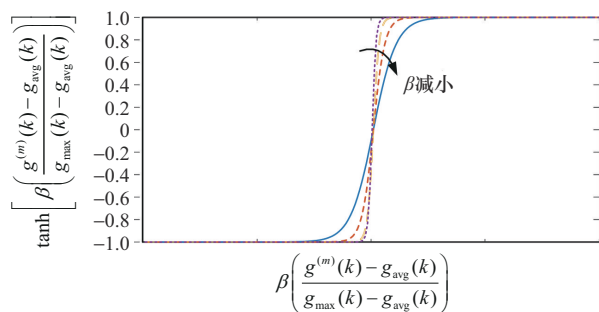


图2 $\tanh$ 函数曲线

2.3.6 终止、生成新个体与结果输出

通过选择、交叉和变异操作，生成一个新的种群。这个新种群包含从父代继承的优秀特征，以及新的变化和多样性。在执行一定数量的迭代或达到特定的终止条件之后，检查算法是否满足终止的条件。本文中设定的迭代终止条件为：

$$|J_{k+1} - J_k| \leq 1\% \text{ 或 } k \geq I_{\max} \quad (13)$$

其中， J_k 为第 k 次迭代生成种群的适应度值， $I_{\max}$ 为最大迭代次数。

3 算例仿真验证与分析

为了验证本文提出方法的有效性，将本文提出的面向云边协同多元负荷调控的云平台计算资源分配策略和对比方法应用于典型的分布式云边协同负荷管理调控系统，具体仿真包括算法仿真参数设置、实验分析与算法比较。

3.1 参数设置

本实验过程采用 *fastone* 仿真云平台，在该仿真云平台中将用户群规模设定为 2 000，每个用户边缘节点可向云平台发送计算任务请求，每个用户边缘节点可发送最大任务数量为 200 个，任务类型包括基于长短期记忆 (long short term memory, LSTM) 网络的多元用户行为模拟任务和基于异步优势执行者—评论家 (asynchronous advantage actor-critic, A3C) 算法的需求侧管理决策任务，云平台仿真参数设置见表 1。

表 1 云平台仿真参数设置

云边协同调控系统	参数	数值
计算任务	最大任务数据规模 F	10^{10}
	最大总任务数量/个	1 000
	任务类型/个	2
资源节点	最大资源节点数/个	100
	最大下载速率/(Mbit·s ⁻¹)	100
	最大上传速率/(Mbit·s ⁻¹)	1 000
云服务器	$S_{\max}$ /个	10
	$T_{w_i, e_k}^{\max}/s$	900

3.2 算法收敛速度与寻优性能验证

实验设置了 20 个资源节点和 100 个计算任务。为了验证本文提出算法在收敛速度与寻优性能的有效性，将本文提出算法与以下 3 种算法对比：文献[22]所提轮询算法按照顺序逐个为每个任务分配一定的计算资源，然后按照顺序循环执行，直到完成所有任务计算资源分配；文献[23]所提粒子群算法采用启发式搜索迭代策略进行计算资源分配；文献[24]所提传统遗传算法采用固

定交叉与变异概率进行计算资源分配。

由于文献[22]所提轮询算法为无迭代的进化算法,故不参与收敛性对比。图3给出了在相同计算任务和可分配计算资源下,本文所提算法和对比算法(传统遗传算法和粒子群算法)的收敛速度对比。从图3可以看出,随着迭代次数的增加,粒子群算法在第89次迭代后达到稳定,稳定在15 s;遗传算法在第95次迭代后达到稳定,稳定在14 s,可见由于传统的遗传算法具有交叉和变异策略,虽然收敛速度慢,但是最终最优解更优;而本文所提方法则仅在61次迭代后就达到稳定,稳定在10 s。可见,本文所提方法的收敛速度较粒子群算法和遗传算法分别提升了76.4%和77.9%,任务完成时间分别缩短了33.3%和28.5%。这是因为本文提出的算法加入了染色体自适应交叉变异策略,可以在前期快速定位可行解空间,后期精准定位最优解,因此,本文提出的算法在收敛速度和全局寻优方面具有更好的性能。

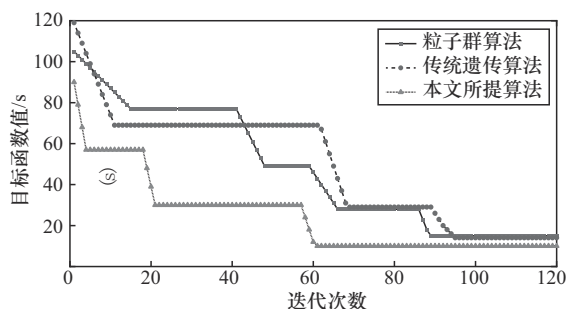


图3 算法的收敛速度对比

3.3 相同计算任务、不同资源节点数下计算任务完成时间对比

为了探究在不同资源节点数的情况下,处理相同任务时本文提出算法和对比方法的寻优能力差异性,本文固定任务数为500个,而资源节点数从20个逐渐增加到100个,将本文提出的方法和轮询算法、粒子群算法及遗传算法的处理任务完成时间进行对比,具体结果如图4所示。

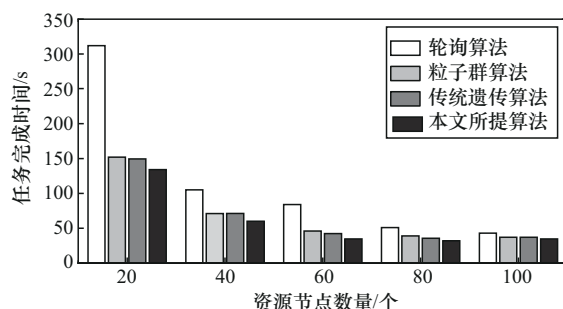


图4 相同计算任务、不同资源节点数下各算法的任务完成时间对比

从图4中可以看出,轮询算法在资源节点数为20时,任务完成时间极长,达到312 s,而其他3种启发式算法任务完成时间则短很多,粒子群算法、遗传算法和本文提出算法分别为152 s、149 s和134 s,可见当资源节点数较少时,本文所提算法任务完成时间最短,分别比轮询算法、粒子群算法和遗传算法降低57.1%、11.8%和10.1%。这是由于在计算资源有限的情况下,完成任务计算较为困难,很难寻找到最优解完成资源分配,而本文提出的算法加入了自适应交叉变异函数,可以更快速并且精准地定位最优计算资源分配方案;此外,还可以看出当资源节点数逐渐增加时,本文提出的算法与其他3种对比方法的完成任务时间逐渐接近,当资源节点数增加至100时,轮询算法、粒子群算法、遗传算法与本文所提算法的任务完成时间分别为43 s、37 s、36 s和34 s,本文所提算法仅比轮询算法、粒子群算法和遗传算法降低20.9%、8.1%和5.6%,这是由于当资源节点数量足够多时,不再需要过于精准的优化算法也可以在较短的时间内完成计算任务资源分配,所以当资源节点数量逐渐增加时,本文提出的算法与其他3种对比算法的完成任务时间逐渐接近。

3.4 相同资源节点数、不同计算任务下计算任务完成时间对比

为了进一步探究在不同计算任务数情况下,使用相同资源节点数时本文所提算法和对比方法



的寻优能力差异性,本文固定资源节点数为20个,而任务数从20个逐渐增加到100个,使用本文所提算法和轮询算法、粒子群算法及遗传算法的处理任务完成时间对比如图5所示。

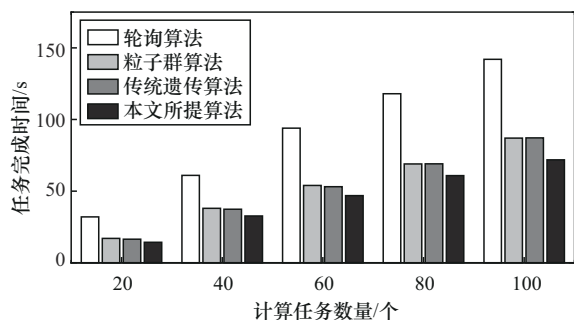


图5 相同资源节点数、不同任务数下各算法的任务完成时间对比

从图5可以看出,本文所提算法处理所有对比任务数的任务完成时间最短。当任务数为20个时,本文所提算法与其他3种方法的任务完成时间较为接近,轮询算法、粒子群算法、遗传算法与本文所提算法的任务完成时间分别为32 s、17 s、16 s和14 s,本文所提算法仅比轮询算法、粒子群算法和遗传算法的任务完成时间低56.5%、17.6%和12.5%,这是由于当任务数较少时,计算资源较充沛,不需要过于精准的优化算法也可以较好地完成任务资源分配,但是轮询算法受任务排序和虚拟资源节点的影响,算法的任务处理时间会相对较长,而粒子群算法和遗传算法搜索能力较强,所以任务完成时间与本文所提算法相差不大。但是,随着计算任务数量逐渐增加,本文所提算法与其他算法的完成时间差距逐渐增大,当任务数为100个时,轮询算法任务完成时间达到142 s,而其他3种启发式算法任务完成时间降低很多,粒子群算法、遗传算法和本文所提算法分别为87 s、88 s和71 s,可见本文所提算法在任务数较多时,任务完成时间是最短的,分别比轮询算法、粒子群算法和遗传算法任务低50.0%、18.3%和19.3%。这是因为在计算资

源有限的情况下,完成任务计算较为困难,很难寻找到最优解完成资源分配,而本文提出的算法加入了自适应交叉变异函数,可以更快速并且精准地定位最优计算资源分配方案。综上所述,本文所提算法在实现云平台计算资源分配过程中可以充分利用有限的云计算资源,减少任务完成时间。

4 结束语

本文针对电动汽车、电采暖设备、空调热泵等多元可控负荷,设计了一种面向多元可控负荷调控的云边协同负荷资源分配策略,保障云平台对边缘计算任务请求的快速响应,避免了传统针对每个计算任务平均分配计算资源导致的高时延和网络拥塞问题。算例结果表明,本文所提算法在实现云平台计算资源分配过程中可以充分利用有限的云计算资源,更精准、高效地管理和调度计算资源,减少了同时将大量负荷数据传输到云端进行处理的巨大开销,避免了云端控制中心在处理高并发计算任务时存在的高时延和网络拥塞问题,提升了云平台对计算任务的响应质量。

参考文献:

- [1] 周颖,龚桃荣,陈宋宋,等.面向新型电力负荷管理的分层分区动态调控架构展望[J].电力信息与通信技术,2023,21(4): 51-58.
ZHOU Y, GONG T R, CHEN S S, et al. Prospect of hierarchical and partitioned dynamic regulation architecture for new power load management[J]. Electric Power Information and Communication Technology, 2023, 21(4): 51-58.
- [2] 国家发展改革委,国家能源局.电力负荷管理办法(2023年版)[J].电力需求侧管理,2023,25(6):122-125.
National Development and Reform Commission, National Energy Administration. Measures for power load management (2023 edition)[J]. Power Demand Side Management, 2023, 25(6): 122-125.
- [3] WANG W, YU T, HUANG Y P, et al. The situation and suggestions of the new energy power system under the background of

- carbon reduction in China[J]. *Energy Reports*, 2021(7): 1477-1484.
- [4] 程林, 万宇翔, 张放, 等. 基于负荷聚合商业业务的空调服务运作模式及控制策略[J]. *电力系统自动化*, 2018, 42(18): 8-16.
CHENG L, WAN Y X, ZHANG F, et al. Operation mode and control strategy for air-conditioning service based on business of load aggregator[J]. *Automation of Electric Power Systems*, 2018, 42(18): 8-16.
- [5] 侯慧, 徐焄, 肖振锋, 等. 计及可调控负荷的发电一体化综合优化调度[J]. *电网技术*, 2020, 44(11): 4294-4301.
HOU H, XU T, XIAO Z F, et al. Generation and load integrated optimal scheduling considering adjustable load[J]. *Power System Technology*, 2020, 44(11): 4294-4301.
- [6] 张春芳. 面向智能制造的边缘计算网络资源分配方法[D]. 大连: 大连海事大学, 2022.
ZHANG C F. Resource allocation method of edge computing networks for intelligent manufacturing[D]. Dalian: Dalian Maritime University, 2022.
- [7] 刘建明, 林弘宇, 陶雄强. 物联网与能源互联网[M]. 北京: 电子工业出版社, 2021.
LIU J M, LIN H Y, TAO X Q. Internet of things and energy Internet[M]. Beijing: Publishing House of Electronics Industry, 2021.
- [8] CHEN L J, ZHU X, CAI J L, et al. Multi-time scale coordinated optimal dispatch of microgrid cluster based on MAS[J]. *Electric Power Systems Research*, 2019, 177: 105976.
- [9] CHANG H H, CHEN H, ZHANG J Z, et al. Decentralized deep reinforcement learning meets mobility load balancing[J]. *IEEE/ACM Transactions on Networking*, 2023, 31(2): 473-484.
- [10] 程钊, 陈羽, 孙伶雁. 考虑服务配置的细粒度电力任务云边协同优化调度策略[J]. *电力系统保护与控制*, 2023, 51(7): 53-62.
CHENG Q, CHEN Y, SUN L Y. Cloud-edge collaborative optimization scheduling strategy for fine-grained power tasks considering service configuration[J]. *Power System Protection and Control*, 2023, 51(7): 53-62.
- [11] 黄杰, 肖志清, 毛冬. 面向电力物联网的云边数据协同方法[J]. *电力信息与通信技术*, 2022, 20(1): 35-42.
HUANG J, XIAO Z Q, MAO D. Cloud-edge data collaboration method for power IoTs[J]. *Electric Power Information and Communication Technology*, 2022, 20(1): 35-42.
- [12] 孔祥玉, 马玉莹, 艾芊, 等. 新型电力系统多元用户的用电特征建模与用电负荷预测综述[J]. *电力系统自动化*, 2023, 47(13): 2-17.
KONG X Y, MA Y Y, AI Q, et al. Review on electricity consumption characteristic modeling and load forecasting for diverse users in new power system[J]. *Automation of Electric Power Systems*, 2023, 47(13): 2-17.
- [13] 陈兆雁, 柴宜, 平常, 等. 基于5G边缘计算的可调节负荷互动技术研究[J]. *华电技术*, 2021, 43(3): 9-13.
CHEN Z Y, CHAI Y, PING C, et al. Research on adjustable load interaction technology based on 5G edge computing[J]. *Huadian Technology*, 2021, 43(3): 9-13.
- [14] 郭芬. 面向虚拟机的云平台资源部署与调度研究[D]. 广州: 华南理工大学, 2015.
GUO F. Research on deploying and scheduling for cloud resources based on virtual machines[D]. Guangzhou: South China University of Technology, 2015.
- [15] HEIMOVSKI G B, TURCHETTI R C, WICKBOLDT J A, et al. FT-Aurora: a highly available IaaS cloud manager based on replication[J]. *Computer Networks*, 2020(168): 107041.
- [16] KANAMARU S, KAWAMURA K, TANAKA S, et al. Solving constrained slot placement problems using an Ising machine and its evaluations[J]. *IEICE Transactions on Information and Systems*, 2021, E104.D(2): 226-236.
- [17] 樊玉琦, 王铮, 杨洋, 等. 快速构建BIT*搜索树的机器人路径规划[J]. *网络安全技术与应用*, 2023(10): 38-41.
FAN Y Q, WANG Z, YANG Y, et al. Robot path planning for rapidly constructing BIT* search tree[J]. *Network Security Technology & Application*, 2023(10): 38-41.
- [18] 张晓笛, 王金昌, 杨仲轩, 等. 基于状态空间法的阶梯型变截面水平受荷桩分析方法[J]. *岩土工程学报*, 2023, 45(9): 1944-1952.
ZHANG X D, WANG J C, YANG Z X, et al. Analytical solutions for laterally loaded step-tapered piles by state space method[J]. *Chinese Journal of Geotechnical Engineering*, 2023, 45(9): 1944-1952.
- [19] KONG W B, CHEN J Z, SONG Y B, et al. Sobel edge detection algorithm with adaptive threshold based on improved ge-



netic algorithm for image processing[J]. International Journal of Advanced Computer Science and Applications, 2023, 14(2): 557-562.

- [20] NA R S, LI Q, WU L J. Particle swarm optimization with negative gradient[J]. Advanced Materials Research, 2012, 468-471: 2550-2553.

- [21] 过羽丰. 基于改进遗传算法的配电网络规划研究[D]. 成都: 西华大学, 2012.

GUO Y F. Distribution network planning based on improved genetic algorithm[D]. Chengdu: Xihua University, 2012.

- [22] 季航旭, 姜苏, 赵宇海, 等. 分布式环境中的多作业执行调度策略与优化[J]. 计算机工程与科学, 2021, 43(6): 951-961.

JI H X, JIANG S, ZHAO Y H, et al. Scheduling and optimization of multi-job execution in distributed environment[J]. Computer Engineering & Science, 2021, 43(6): 951-961.

- [23] PAN J H, WANG H, YANG X G. A random particle swarm optimization algorithm with application[J]. Advanced Materials Research, 2013, 634-638: 3940-3944.

- [24] 岑柏霖. 改进GA-BP算法在电力系统负荷预测模型中的应用研究[J]. 自动化与仪器仪表, 2023(11): 53-57.

CEN B L. Application of improved GA-BP algorithm in power system load forecasting model[J]. Automation & Instrumentation, 2023(11): 53-57.

[作者简介]



李思维 (1988-), 男, 天津大学智能电网教育部重点实验室博士生, 北京中电飞华通信有限公司高级工程师, 主要从事电力系统需求响应和优化运行与控制工作。



靳莉 (1981-), 女, 北京中电飞华通信有限公司工程师, 主要从事电力系统需求响应和优化运行与控制工作。



于龙 (1989-), 男, 北京中电飞华通信有限公司工程师, 主要从事电力系统需求响应和优化运行与控制工作。



杜立石 (1985-), 男, 北京中电飞华通信有限公司工程师, 主要从事电力系统负荷管理和电力系统优化运行与控制工作。



岳靓 (1995-), 女, 北京中电飞华通信有限公司工程师, 主要从事电力系统需求响应和优化运行与控制工作。



张喜润 (1981-), 男, 北京中电飞华通信有限公司工程师, 主要从事电力信息通信和电力负荷管理及优化调控工作。