

DOI: 10.12158/j.2096-3203.2024.05.010

# 考虑资源弹性配置的配电网保护控制终端协同任务分配方法

刘媛媛<sup>1</sup>, 陈元桦<sup>2</sup>, 蔡泽祥<sup>1</sup>, 刘文泽<sup>1</sup>, 屈径<sup>1</sup>, 胡凯强<sup>1</sup>

(1. 华南理工大学电力学院, 广东 广州 510641;

2. 南方电网电力科技股份有限公司, 广东 广州 510630)

**摘要:**随着新型电力系统的发展,利用智能终端处理愈发复杂的配电网保护控制任务时,对资源供给与需求的平衡要求越来越高。因此,文中提出一种考虑资源弹性配置的配电网保护控制终端(protect and control intelligent terminal, PCIT)协同任务优化分配方法。首先,阐述多终端协同的技术架构,并建立基于容器的PCIT的弹性资源模型、任务处理模型。其次,提出双层模型用于优化保护控制任务在终端间的协同分配、资源的弹性调度,并利用隐枚举法对该模型进行求解,从而充分发挥任务处理时资源的灵活性,提升任务处理性能。最后,算例验证了文中所提方法的可行性与先进性,各智能终端计算资源的占用率降低约28.85%,任务平均处理延时减少约4.12%。

**关键词:**资源弹性配置;保护控制终端(PCIT);容器;协同优化;双层模型;隐枚举法

中图分类号:TM73

文献标志码:A

文章编号:2096-3203(2024)05-0100-12

## 0 引言

随着新型电力系统的发展,分布式能源、储能、可控负荷等广泛接入配电网,使系统运行、设备管理、数据处理等均发生较大改变<sup>[1-3]</sup>。现有的垂直式主子站架构难以满足日渐增长的多元化配电业务处理需求,采用多功能一体化的保护控制终端(protect and control intelligent terminal, PCIT)协同处理配电网日益复杂的数据处理任务是必然趋势<sup>[4-6]</sup>。因管控区域内的设备类型、数量等差异,PCIT的资源利用程度不同,进而导致任务处理性能不同。因此,优化任务在各PCIT间的分配,对提高资源利用率,任务处理性能有重要意义<sup>[7-8]</sup>。

基于PCIT的配电网分布式决策体系须考虑不同配电任务的处理特性与PCIT的资源利用情况。一方面,配电业务的处理呈现空间与时间的分布特征,空间特征指任务的处理数据来自本地PCIT管控的局部区域,或来自多个PCIT管控的跨区域,前者仅由本地PCIT即可完成处理,而后者需要多台PCIT协同处理;时间特征指任务的处理须遵循一定的任务耦合关系,该关系包括任务处理先后、处理结果的调用等。另一方面,PCIT承担本地任务及与其他PCIT协同任务的分配,需要根据不同任务的分配特征进行终端资源的分配,PCIT资源利用情况影响其所能承担的任务类型与数量。并且,环境天气、拓扑结构等变化会导致任务的计算负荷波动,

须根据不同负荷差异合理化配置容器资源,从而实现资源的最优利用。

针对多智能体协同与任务调配,文献[9]提出一种多用户系统协同的任务动态分配方法,实现了云端资源辅助下的多边缘系统间的优化分配,但未考虑任务间的关联关系。文献[10-11]提出考虑边缘计算终端部署的任务协同处理方法,为任务的分配提供思路,但未考虑任务负荷变化时对资源过度占用的问题,且缺少对资源弹性配置的研究。文献[12]提出一种针对随机计算负荷的任务分配与计算资源配置方法,实现了能耗开销最小化,但未考虑多元化任务处理延时需求。针对资源弹性配置,文献[13]提出基于微服务建模的终端计算资源配置方法,利用延时容忍度、资源灵敏度等实现容器资源的弹性伸缩配置,但缺少分析动态场景下负荷变化对资源配置的影响。文献[14]提出基于深度学习的动态优化算法,解决了动态流量场景的资源分配问题,但该算法对于短时间尺度优化问题优势较小。文献[15]提出一种面向数据流的集群资源调度框架,通过资源的细粒度分配,既能有效缩减任务处理时间,又能提升图形处理器(graphics processing unit, GPU)利用率,为容器弹性资源配置与终端集群协作提供思路。

因此,文中提出一种考虑资源弹性配置的配电网PCIT协同任务优化分配方法。首先,阐述了配电网保护控制技术的架构体系,包括PCIT间的协同架构以及基于容器的PCIT架构。其次,建立了配电网PCIT任务处理模型,包括任务关系模型与任务协同处理模型。再次,提出了PCIT协同的优

收稿日期:2024-02-18;修回日期:2024-04-22

基金项目:广东省重点领域研发计划资助项目(2019B1111-09002)

化分配双层模型,上层模型求解业务在各台 PCIT 间的优化分配,下层模型优化 PCIT 中各个任务的弹性资源配置。最后,采用隐枚举法对模型进行求解,利用算例验证了文中所提方法的可行性与先进性,为配电网保护控制任务高效处理与资源优化利用奠定基础。

## 1 配电网保护控制技术架构

### 1.1 PCIT 间的协同架构

围绕实际电网一次系统构建配电网保护控制架构,根据环网点母线分布、装置设备接入情况等实现配电网的 PCIT 管控区域划分<sup>[16-17]</sup>。图 1 为典型配电网保护控制架构,分为一次系统、PCIT 分布式决策系统,其中,k1—k5、k 为连接不同设备元件的开关。

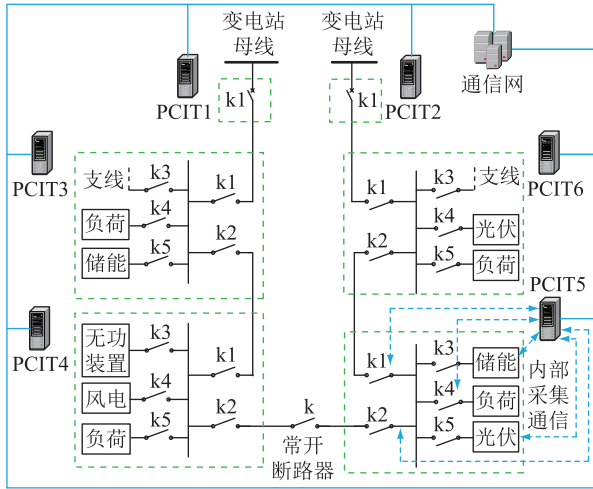


图 1 配电网保护控制架构及 PCIT 管控区域

Fig.1 Protection and control architecture of distribution network and PCIT control area

在一次系统中,不同负荷、储能装置、光伏等分布式电源根据其自身特点与分布情况接入配电网<sup>[18]</sup>。文中将接入同一环网点母线的支线划分为一个单元区域,由一台 PCIT 管控,如图 1 中绿色虚线框所示。在实际系统中,单元区域可为配电房、开闭所、光伏并网点等。单元区域中 PCIT 的监测管控对象包括:母线电压、连接在环网点母线的所有开关状态和支路电流、接入该环网点的所有可以接收控制指令的光伏装置、储能装置以及可调控负荷的开关等,文中将该类设备称为末端设备 (terminal device, TD)。

在 PCIT 分布式决策系统中,数据交互方式包括:(1) 单元区域内的 TD 与 PCIT 间的数据交互,通信双方属于不同节点层级;(2) 不同 PCIT 间的数据交互,通信双方属于同一节点层级。PCIT 通过内

部采集通信获取单元区域内的电压、电流、功率、开关状态、设备状态等信息,如图 1 中带箭头的蓝色虚线所示<sup>[19]</sup>;PCIT 通过通信网络实现与其他区域的 PCIT 间的数据交互,如图 1 中蓝色实线所示。在不同的事件触发或任务请求下,PCIT 综合所获单元区域或其他 PCIT 协同的信息进行数据分析计算,并做出相应的决策指令<sup>[20]</sup>。

### 1.2 基于容器的 PCIT 架构

容器技术是 PCIT 处理本地任务以及进行协同控制的关键,也是 PCIT 中用于处理任务的最小资源空间<sup>[21]</sup>,既可实现不同任务处理过程的隔离,又能满足 PCIT 处理多元化任务时资源的共享。在 PCIT 处理任务的过程中,不同类型的任务被放置于相应的容器中,当任务被调度执行时,该容器获取相应的资源进行处理。PCIT 为不同类型的电力任务创建专属的容器,但任务的计算负荷常因系统运行方式变化、通信噪声干扰等因素而发生动态变化,若因计算负荷变化而频繁创建容器,也将产生不可忽视的系统延时和能耗。此外,通过资源的可定义技术为容器灵活分配资源<sup>[13]</sup>,使容器的资源空间具备一定的伸缩弹性,文中将容器资源弹性伸缩范围定义为容器的弹性因子。在 PCIT 中,操作系统通过调节容器弹性因子,使其可调用容器中的空闲资源或扩展容器的资源容量空间。弹性因子为容器的资源调节参数,其数值越大,容器受操作系统调控范围越大。一方面,当任务计算负荷较小时,容器的计算资源将被其他容器占用,此刻,处理延时增大;另一方面,当计算负荷增长较大时,又可从其他容器调用空闲的资源,使任务处理延时不超过允许范围。因此,文中提出利用容器资源弹性配置的方式为任务提供资源,容器资源弹性供给模型  $S_j$  如式(1)所示。

$$\begin{cases} S_j = [s_{j,\text{cpu}}, s_{j,\text{ram}}, s_{j,\text{bw}}, \lambda_j] \\ S_{j,0} = [s_{j,\text{cpu},0}, s_{j,\text{ram},0}, s_{j,\text{bw},0}] \quad \lambda_j = 0 \end{cases} \quad (1)$$

$$\begin{cases} (1 - \lambda_j)s_{j,\text{cpu},0} \leq s_{j,\text{cpu}} \leq (1 + \lambda_j)s_{j,\text{cpu},0} \\ (1 - \lambda_j)s_{j,\text{ram},0} \leq s_{j,\text{ram}} \leq (1 + \lambda_j)s_{j,\text{ram},0} \\ (1 - \lambda_j)s_{j,\text{bw},0} \leq s_{j,\text{bw}} \leq (1 + \lambda_j)s_{j,\text{bw},0} \end{cases} \quad (2)$$

式中: $s_{j,\text{cpu}}$ 、 $s_{j,\text{ram}}$ 、 $s_{j,\text{bw}}$  分别为容器  $j$  的计算、内存、带宽资源的供给量; $\lambda_j$  为容器  $j$  的分配资源的弹性因子,且  $0 < \lambda_j < 1$ ;  $S_{j,0}$  为  $\lambda_j = 0$  时容器  $j$  的资源供给量,此时容器  $j$  的计算、内存、带宽资源为固定容量大小,分别为  $s_{j,\text{cpu},0}$ 、 $s_{j,\text{ram},0}$ 、 $s_{j,\text{bw},0}$ 。故由式(1)可知,PCIT 可调度容器  $j$  的资源容量区间  $S_j$  须满足式(2)所示的约束条件。

## 2 配电网 PCIT 任务处理模型

### 2.1 任务特征模型

(1) 单终端任务(single-terminal task, STT)与跨终端任务(cross-terminal task, CTT)。根据处理数据来源的不同范围将任务分为 STT 与 CTT<sup>[10]</sup>。一般而言,STT 具有计算负荷较小、延时容忍度较小、任务周期较短等特征,是指由单元区域中 TD 请求的处理任务,且各 TD 的 STT 请求由本地 PCIT 处理。CTT 具有计算负荷与延时容忍度较大、任务周期较长等特征,是指其任务处理需要获取多个单元区域内 TD 的采集数据,在完成数据获取后,可由本地 PCIT 处理,也可通过 PCIT 间交互共同协作完成。

(2) 任务计算负荷。终端处理任务与任务的计算负荷密切相关。文中通过建立任务的特征模型表征不同类型的任务,通常为计算资源与延时容忍度<sup>[13]</sup>。其中,计算资源主要与计算负荷有关,计算负荷的大小与任务类型、处理数据量大小有关,如式(3)所示。而式(4)阐述了任务  $a$  的数据量为  $u_a$  时的计算负荷大小,任务  $a$  的计算负荷表征为单位数据的计算负荷与数据量的乘积。

$$\theta_a = [p_a \quad w_a(t) \quad T_a] \quad a = 1, 2, \dots, m \quad (3)$$

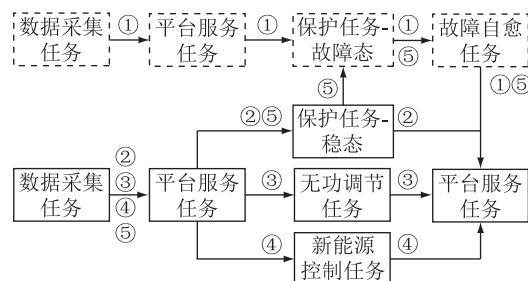
$$w_a(t) = q_a u_a(t) \quad (4)$$

$$\mathbf{W} = [w_1(t) \quad w_2(t) \quad \dots \quad w_m(t)] \quad (5)$$

式中: $\theta_a$  为任务的特征模型; $p_a$  为任务类型; $w_a(t)$  为计算负荷的大小; $T_a$  为延时约束; $m$  为总任务数; $q_a$  为处理  $p_a$  类任务时每单位数据的计算负荷; $u_a(t)$  为任务随时间  $t$  变化的数据量; $\mathbf{W}$  为所有任务的计算负荷矩阵。

### 2.2 任务耦合关系

为实现电力系统的安全稳定运行,常须调用不同应用功能对数据进行相应的分析计算。不同应用功能并非简单的串行、并行处理关系,而是遵循一定的逻辑耦合关系。逻辑耦合关系与任务类型相关,文中考虑的配电网保护控制场景的各任务耦合关系如图 2 所示。其中,STT 主要为了实现单元区域内的保护功能,由 PCIT 执行相应的应用功能,包括数据采集、终端平台服务、保护功能;各 STT 间呈串行的处理关系。而 CTT 主要为了实现对系统的最优控制,与网络拓扑、光伏与储能的分布特点、故障发生位置等因素相关,根据 PCIT 管控设备类型与特点;CTT 主要包括拓扑分析、资源响应能力分析、无功调节、主动源荷控制、故障自愈;CTT 间呈串行、并行、串并行处理等逻辑耦合关系。PCIT 承担的各任务内容如表 1 所示。



①—时序逻辑链1; ②—时序逻辑链2; ③—时序逻辑链3; ④—时序逻辑链4; ⑤—时序逻辑链5。

图 2 PCIT 任务耦合关系

Fig.2 The coupling relationship of PCIT tasks

表 1 PCIT 承担的任务类型及相应的功能内容

Table 1 Types of tasks undertaken by PCIT and the contents of the respective functions

任务编号	任务名称	功能内容
STT <sub>1</sub>	数据采集	同步数据、预处理数据、插值数据等
STT <sub>2</sub>	平台服务	三遥、配置管理、通信、录波、事件记录、定值服务等
STT <sub>3</sub>	保护	(故障态)过流保护、低压保护、零序电流/电压保护、加速保护、失灵保护、故障定位等; (稳态)报文解析、计算幅值、检测保护装置状态、备份数据、故障预测/智能告警等
CTT <sub>1</sub>	拓扑分析	接线方式分析、运行方式分析、拓扑变换分析等
CTT <sub>2</sub>	资源响应能力分析	光伏供电能力分析、柔性负荷容量及响应速度分析、储能充放状态及容量分析等
CTT <sub>3</sub>	无功调节	电压计算、无功计算、确定补偿位置、计算补偿容量、电容器组控制、开环矢量控制、调压变压器控制等
CTT <sub>4</sub>	主动源荷控制	跟踪计划出力、跟踪功率、削峰填谷、确定调节目标、平滑输出功率、控制新能源逆变器、备用电源自动投入、合环转供电、自动需求响应、精准负荷控制等
CTT <sub>5</sub>	故障自愈	识别相邻开关过流、开关试合闸、识别相邻开关电流方向、定位故障区域等

在图 2 所示的任务耦合关系中,串行关系表现为任务的依次、有序处理,并行关系则表现为任务同时开始,且构成的整体处理完成的标志是最后一个任务处理完成。进一步地,文中根据任务的耦合关系建立数据传输模型  $y_{av}$ ,如式(6)所示。

$$y_{av} = \begin{cases} 1 & \text{任务 } a \text{ 的处理结果传输给任务 } v \\ 0 & \text{其他情况} \end{cases} \quad (6)$$

对不同的  $a, v$ , 变量  $y_{av}$  的取值也将不同,所有取值共同组成矩阵  $\mathbf{Y}=(y_{av})_{m \times m}$ ,以表明各个任务间的数据流向。

### 2.3 PCIT 任务协同处理过程

PCIT 既面向单元区域内多种设备的任务处理,又参与其他 PCIT 的协同过程,因此 PCIT 常面

临多任务的处理。而各任务对 PCIT 的资源需求表现出一定的竞争性,为提高 PCIT 处理多任务的效率,须合理分配 PCIT 中各容器的资源。图 3 为 PCIT 并行任务处理结构示意图。各单元区域内设备或用户向本地 PCIT 请求各类型任务处理,PCIT 决策任务留在本地处理或转移至其他 PCIT 中处理,并生成待处理任务队列与待调度任务队列。进一步地,PCIT 根据业务请求信息创建初始容量的各类型容器,并将待处理队列中的各个任务映射至适当的容器中,且随任务计算负荷变化,根据弹性因子调节容器资源容量。而其他 PCIT 将处理结果返回给本地 PCIT,并将控制指令下发给 TD 的执行元件。

### 3 PCIT 协同的任务优化分配双层模型

利用多台 PCIT 协同处理配电网保护控制任务,为了优化信息流分布、提高资源利用效率、加速数据分析计算与任务处理,建立 PCIT 协同的任务优化分配双层模型。其中,上层模型求解任务在各台 PCIT 间的优化分配,下层模型则优化 PCIT 中各个任务分得的资源数量。

#### 3.1 上层模型

##### 3.1.1 上层模型目标函数

利用多台 PCIT 协同处理多元化的配电网保护控制任务,该过程中占用系统的计算资源与通信资源,产生计算延时与通信延时。因此,为实现资源的合理利用与任务的高效处理,建立上层模型,其目标函数为:

$$\min C(t) = \sum_{e=1}^K (C_{1,e}(t) + C_{2,e}(t)) \quad (7)$$

式中: $C(t)$ 为系统任务处理过程中随时间  $t$  变化的总延时; $e$  为 PCIT 的编号; $C_{1,e}(t)$ 、 $C_{2,e}(t)$  分别为 PCIT <sub>$e$</sub>  处理任务时产生的计算延时和通信延时; $K$  为系统 PCIT 的数量。

##### 3.1.2 通信延时

在 PCIT 协同处理任务的过程中,数据交互存在于 TD 与 PCIT 间、各 PCIT 间,采用的通信方式包括无线和有线 2 种。其中,无线方式包含无线局域网(wireless fidelity, Wi-Fi)、远距无线(long range radio, LoRa)通信、4G/5G 等,有线方式有低压载波通信、光纤等。根据 TD 类型及其所处位置的不同,采用的通信方式不同,如负荷开关采用低压载波通信,充电桩采用 LoRa 通信等。通信关系利用 0-1 变量  $x_{ie}$  与  $x_{ee'}$  进行表征,分别如式(8)、式(9)所示。

$$x_{ie} = \begin{cases} 1 & \text{TD}_i \text{ 的 STT 由 PCIT}_e \text{ 进行处理} \\ 0 & \text{其他情况} \end{cases} \quad (8)$$

$$x_{ee'} = \begin{cases} 1 & \text{PCIT}_{e'} \text{ 向 PCIT}_e \text{ 传输数据} \\ 0 & \text{其他情况} \end{cases} \quad (9)$$

式中: $i$  为 TD 编号; $e'$  为区别于  $e$  的 PCIT 的编号。

在图 1 所示的 PCIT 协同架构中,PCIT <sub>$e$</sub>  处理不同任务时产生的数据传输延时如式(10)所示,其由 PCIT 与 TD、PCIT 与 PCIT 间的传输延时两部分组成。通信延时取决于传输内容与传输速率,由香农第二定理可知,数据传输速率与信道容量大小有关,且传输速率与通信延时呈反比,故文中利用式(11)、式(12)对通信延时进行刻画。式(10)所利

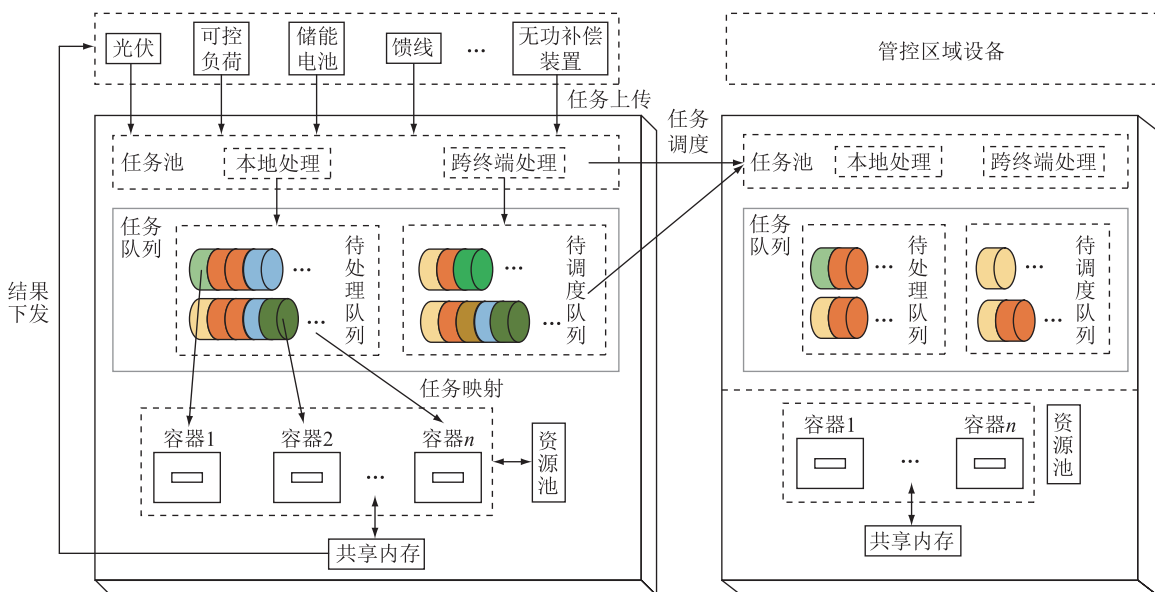


图 3 PCIT 并行任务处理结构示意图

Fig.3 Schematic diagram of PCIT parallel task processing structure

用的  $d_{ie}(t)$  与  $d_{ee'}(t)$  分别由式(11)与式(12)计算得到。而式(13)为利用香农第二定理计算所得的信息传输速率,其考虑了不同设备间通信带宽大小、受噪声影响情况,有效模拟了不同场景下设备的通信情况。

$$C_{2,e}(t) = \sum_{i=1}^I d_{ie}(t) + \sum_{e'=1}^K d_{ee'}(t) \quad (10)$$

$$d_{ie}(t) = \begin{cases} 0 & \text{PCIT}_e \text{ 部署于 TD}_i; i = 1, 2, \dots, I \\ \frac{u_{ie}(t)}{R_{ie}(t)} x_{ie} & \text{其他情况} \end{cases} \quad (11)$$

$$d_{ee'}(t) = \frac{u_{ee'}(t)}{R_{ee'}(t)} x_{ee'} \quad (12)$$

$$\begin{cases} R_{ie}(t) = B_{ie} \log_2(1 + P_{ie}(t)) \\ R_{ee'}(t) = B_{ee'} \log_2(1 + P_{ee'}(t)) \end{cases} \quad (13)$$

式中: $d_{ie}(t)$ 为  $\text{TD}_i$ 与  $\text{PCIT}_e$ 间传输数据量为  $u_{ie}(t)$ 时产生的传输延时; $d_{ee'}(t)$ 为  $\text{PCIT}_e$ 与  $\text{PCIT}_{e'}$ 间传输数据量为  $u_{ee'}(t)$ 时产生的传输延时; $I$ 为  $\text{TD}$ 的数量; $R_{ie}(t)$ 、 $R_{ee'}(t)$ 分别为  $\text{TD}_i$ 与  $\text{PCIT}_e$ 、 $\text{PCIT}_e$ 与  $\text{PCIT}_{e'}$ 间的信息传输速率; $B_{ie}$ 、 $B_{ee'}$ 为带宽; $P_{ie}(t)$ 、 $P_{ee'}(t)$ 分别为  $\text{TD}_i$ 与  $\text{PCIT}_e$ 、 $\text{PCIT}_e$ 与  $\text{PCIT}_{e'}$ 间的信噪比。

### 3.2 下层模型

#### 3.2.1 下层模型目标函数

PCIT 为数据处理的主要设备,其承担的任务类型与数量会直接影响任务的处理性能优劣以及并行任务间的资源分配容量。因此,文中建立下层模型以优化 PCIT 的资源利用,其目标函数如式(14)所示。

$$\min C_1(t) = \sum_{e=1}^K C_{1,e}(t) \quad (14)$$

式中: $C_1(t)$ 为总的计算延时。

#### 3.2.2 计算延时

在 PCIT 中,不同任务被放置于不同处理容器中,并根据任务属性为每个容器分配一定数量的计算、通信资源,以期在满足 STT 的延时约束条件下为 CTT 提供更多的剩余资源。在实际应用中,受任务数量、任务类型等因素的影响,PCIT<sub>e</sub>处理多任务时产生的计算延时与任务本身的计算负荷、托管该任务的容器计算资源容量大小相关,表征为:

$$C_{1,e}(t) = \sum_{a=1}^m d_{ae}(t) \propto x_{ae} \quad (15)$$

$$d_{ae}(t) = \frac{w_a(t)}{S_{j,\text{cpu}}} x_{ae} \quad (16)$$

$$a = 1, 2, \dots, m; e = 1, 2, \dots, K \quad (16)$$

$$x_{ae} = \begin{cases} 1 & \text{任务 } a \text{ 分配至 PCIT}_e \text{ 上处理} \\ 0 & \text{其他情况} \end{cases} \quad (17)$$

式中: $d_{ae}(t)$ 为  $\text{PCIT}_e$ 处理任务  $a$  所需的时间; $x_{ae}$ 为任务  $a$  与  $\text{PCIT}_e$ 间对应关系的 0-1 变量。式(15)中单个任务处理所需的计算延时由式(16)进行说明,式(16)中的变量由式(1)一式(4)、式(17)分别进行阐述。

#### 3.2.3 资源与延时约束

除上述模型外,PCIT 处理多任务时还须满足一定的资源与延时约束条件。任意一台 PCIT 处理任务  $a$  时应满足的计算资源约束条件如式(18)所示,其表示 PCIT 中创建的任意容器的计算资源容量小于 PCIT 的固有资源容量。保护控制任务的处理须满足一定的延时容忍度要求,任务的处理延时主要分为数据上行与指令下行的传输延时、PCIT 处理数据时产生的计算延时两部分,如式(19)所示。

$$\begin{cases} 0 < s_{j,\text{cpu}} \leq F_e \\ 0 < \sum_{j=1}^n \sum_{a=1}^m s_{j,\text{cpu}} x_{ae} \leq F_e \end{cases} \quad (18)$$

$$\begin{cases} 2d_{ie}(t) + d_{ae}(t)x_{ie}x_{ae} \leq T_a & a = 1, 2, \dots, m_1 \\ 2d_{ee'}(t) + d_{ae}(t)x_{ee'} \leq T_a & a = 1, 2, \dots, m_2 \end{cases} \quad (19)$$

式中: $F_e$ 为  $\text{PCIT}_e$ 的计算资源总量; $m_1$ 、 $m_2$ 分别为 STT、CTT 的任务编号的最大值,且  $m_1 + m_2 = m$ 。

### 3.3 双层模型决策变量及耦合关系分析

#### 3.3.1 决策变量分析

由式(10)、式(15)及相关的推导可知,目标函数  $\min C(t)$  等价于与变量  $x_{ae}$  及  $x_{ee'}$  有关的多项式函数  $g(x_{ae}, x_{ee'})$ ,如式(20)所示。

$$\begin{aligned} \min C(t) &= \sum_{e=1}^K (C_{1,e}(t) + C_{2,e}(t)) = \\ & \sum_{e=1}^K \left( \sum_{a=1}^m d_{ae}(t) + \sum_{i=1}^I d_{ie}(t) + \sum_{e'=1}^K d_{ee'}(t) \right) = \\ & g(x_{ae}, x_{ee'}) \end{aligned} \quad (20)$$

由式(20)可知,式(7)所示的目标函数决策变量为  $x_{ae}$  与  $x_{ee'}$ 。当任务的耦合关系已知时,PCIT 间的交互关系  $x_{ee'}$  取决于各 CTT 在 PCIT 间的分配关系,即  $x_{ee'}$  为  $x_{ae}$  的隐函数。在图 2 所示的耦合关系中,该隐函数如式(22)所示,其进一步简化了目标函数的决策变量个数。

$$\begin{cases} \eta_{uv} = \sum_{a=1}^m x_{ae} |_{e=u} \cdot y_{av} & \forall u, v \\ U = (\eta_{uv}) \end{cases} \quad (21)$$

$$x_{ee'} |_{e=u, e'=v} = \sum_{v=1}^K \eta_{uv} |_{u=e} \cdot x_{ae} |_{a=v} \quad (22)$$

式中: $U$ 为  $x_{ee'}$ 与  $x_{ae}$ 间转换的系数矩阵; $\eta_{uv}$ 为系数

矩阵  $U$  中的元素。进一步地,因矩阵  $U$  的存在,可构建如式(22)所示的  $x_{ee'}$  与  $x_{ae}$  间的转换关系。

通过上述转换,可将多项式函数  $g(x_{ae}, x_{ee'})$  等价于  $h(x_{ae})$ 。所以求解上述双层模型的目标函数  $\min C(t)$  等价于求解变量  $x_{ae}$  的全部取值。

### 3.3.2 双层模型间耦合关系分析

上层模型根据当前 PCIT 的资源利用情况,以最小化各任务处理延时为目标,完成任务在 PCIT 间的优化分配;下层模型根据 PCIT 承担的任务类型与数量,结合任务处理需求,以 PCIT 内任务处理延时最小化为目标,优化终端资源在任务间的分配,并设置适合的弹性因子。上、下层模型间的耦合关系如图 4 所示。

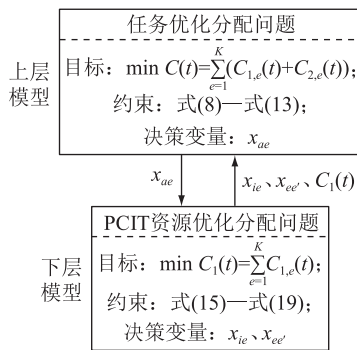


图 4 上、下层模型间的耦合关系

Fig.4 Coupling relationship between upper and lower models

任务优化分配的目的在于根据各 PCIT 处理多个任务时的资源利用情况,同时考虑任务耦合关系,优化各 CTT 在 PCIT 间的分配。PCIT 管控区域内请求的任务类型及计算负荷的差异使得 PCIT 的资源利用情况不同,进一步导致每台 PCIT 对 CTT 的资源供给程度不同。因此,在下层模型中,根据已知的单元区域内 STT 的资源请求情况,优化终端的资源利用,并将相应的处理结果返回至上层模型。而上层模型根据各台 PCIT 的资源利用情况以及可用的资源容量,将 CTT 分配至合适的 PCIT 中处理。此时,PCIT 承担的处理任务增加,进一步根据所承担的任务类型与数量优化 PCIT 内的资源利用情况,并将结果返回上层模型,从而迭代以达到最优的任务处理效果。

## 4 模型求解

### 4.1 双层模型求解适用方法分析

双层模型的求解可分为如下 3 个过程:一是优化决策每台 PCIT 承担的 CTT;二是根据分配的任务,优化 PCIT 中的资源利用结构;三是根据各 PCIT

任务处理情况调整 CTT 的分配,以得到最优的任务处理性能。

CTT 的分配在 PCIT 的协同处理过程中属于 0-1 规划问题,目前解决这类问题的常用方法有枚举法、变换法、隐枚举法以及利用求解器 CPLEX 求解等<sup>[23-25]</sup>。然而,每种方法各有优劣,且并非适用于求解各类问题。例如,枚举法求解时具有迭代次数多、计算时间长、计算量大等缺点;变换法通常适用于特定的 0-1 规划问题,而文中数学模型的限制使其无法构造出变换法需要的效率矩阵;利用求解器 CPLEX 求解的前提为构建的模型、约束条件等均呈线性,因此须将非线性的部分转化为线性再进行求解,增加了求解难度。故而文中采用带边界条件的隐枚举法求解所提双层模型。

PCIT 内的资源分配主要是利用操作系统的资源管控能力,实现为每个任务创建所需的资源空间。目前常用的算法是首次适应算法(first-fit algorithm, FFA),其优先利用低址的资源空间,保留高址的资源空闲区,有利于为后续任务提供所要求的较大容量的容器。

### 4.2 基于隐枚举法的双层模型求解方法

采用带边界条件的隐枚举法求解的步骤可简述为:(1) 分别确定上、下层模型的可行解范围,两者迭代进一步确定双层模型的可行解;(2) 根据给定初始条件分析模型计算结果,并设置求解边界;(3) 根据分支选取新的可行解并计算结果,若该结果小于边界值,则将其作为新的边界;(4) 更新边界,并缩小各分支的可行解范围;(5) 所有分支内的可行解是否都经过判断或计算,若是,则边界为最优解,若否,则重复步骤(2)~步骤(5)。

利用隐枚举法求解上层模型,主要实现任务在 PCIT 间的分配以及 PCIT 内的资源配置优化,求解过程中须明确求解分支与求解边界。在求解分支的选取中,文中将上层模型中每一 CTT 的  $K$  种分配方案作为分支的  $K$  条支路,  $m_2$  条首尾相连的支路构成上层模型的一个可行解。此外,在下层模型中根据任务负荷差异设置弹性因子,并将计算得到的延时结果作为求解边界。基于隐枚举法的双层模型求解过程如图 5 所示。在获取管控区域内的各 TD 信息及数据处理需求、任务类型及时序逻辑关系后,将获得的数据输入至双层模型中,双层模型根据获得的信息给定初始的分配方案并设置边界值。上、下层模型分别根据式(7)~式(13)、式(14)~式(17)进行分配方案的延时计算,比较计算值与边界值的大小关系,更新边界值并进一步明确可行解

的范围,以重新进行迭代求解。

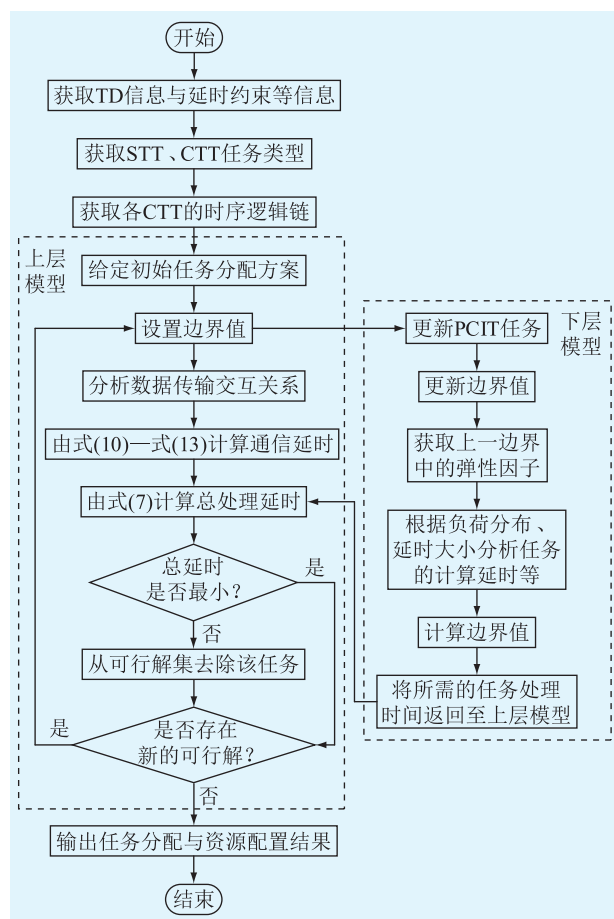


图5 基于隐枚举法的双层模型求解过程

Fig.5 The process of solving the two-layer model based on the implicit enumeration method

## 5 算例分析

### 5.1 参数设置

文中以图1所示的某地区典型配电系统为例,在软件CloudSim及MATLAB中进行仿真分析,其中CloudSim提供了容器、主机等模块,有利于搭建PCIT资源模型及进行相关研究,而MATLAB在编程及图像生成方面的适用性广泛,为任务的协同分配研究提供工具<sup>[26-27]</sup>。结合图1所示的配电系统结构、参数<sup>[28-29]</sup>以及当前关于配电网保护控制任务计算负荷的相关研究<sup>[9-11,13,27]</sup>,文中设置保护控制任务的相关参数如表2所示,其中,CTT<sub>1</sub>与CTT<sub>2</sub>的计算负荷变化情况如图6所示,PCIT参数及创建的多种初始容器参数如表3所示<sup>[27,30]</sup>,MI(million instructions)为百万指令数。

### 5.2 容器资源容量对任务处理延时的影响分析

为探究容器的不同资源容量对同一任务的处理延时的影响,文中设置4种容量的容器,且使容量

表2 配电网保护控制任务的相关参数

Table 2 Relevant parameters for distribution network protection and control tasks

任务编号	数据量/ KB	单位计算负荷/ (10 <sup>-5</sup> MI·B <sup>-1</sup> )	延时 约束/s
STT <sub>1</sub>	100~200	60~100	≤0.10
STT <sub>2</sub>	150~250	140~200	≤0.15
STT <sub>3</sub>	200~250	160~280	≤0.20
CTT <sub>1</sub>	350~450	3 000~5 000	≤1.50
CTT <sub>2</sub>	200~300	2 200~4 000	≤1.50
CTT <sub>3</sub>	350~450	2 600~4 400	≤1.50
CTT <sub>4</sub>	250~350	2 000~3 500	≤1.50
CTT <sub>5</sub>	250~350	2 600~4 600	≤1.50

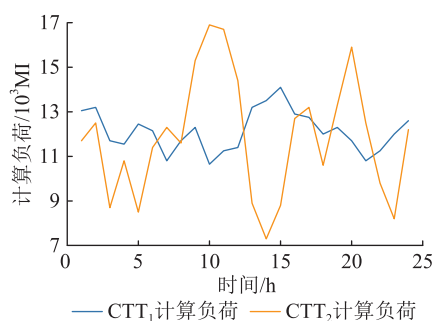


图6 CTT<sub>1</sub>与CTT<sub>2</sub>的计算负荷变化情况

Fig.6 Variation of computational loads for CTT<sub>1</sub> and CTT<sub>2</sub>

表3 配电网PCIT的相关参数

Table 3 Relevant parameters for distribution network PCIT

实体设备	F <sub>c</sub> / (MI·s <sup>-1</sup> )	内存/ GB	带宽/ (Mb·s <sup>-1</sup> )	功率/W	核心数
PCIT	37 274	128	100	93~135	8
	4 658	16	12.5	12~17	1
容器	9 316	32	25	24~34	1
	18 636	64	50	47~68	1

2、容量3、容量4分别为容量1的2倍、3倍、4倍。当利用4种容量的容器托管同一任务时,得到如图7所示的处理延时变化曲线。图7(a)为任务计算负荷在较大区间内变化时的处理延时变化情况,且其中虚线矩形框放大部分如图7(b)所示。

由图7(b)可知,任务的处理延时不随计算负荷的增大而线性增大,其实际由若干线段组成。这是因为仿真过程中考虑了容器的创建延时,且该延时不随容器资源容量与任务计算负荷等变化而变化。因此在图7(b)中,当计算负荷处于较小值时,总处理延时较小,容器的创建延时占总延时的比例较大,从而影响了处理延时随计算负荷增长的线性化。而当计算负荷在较大区间内变化时,计算延时

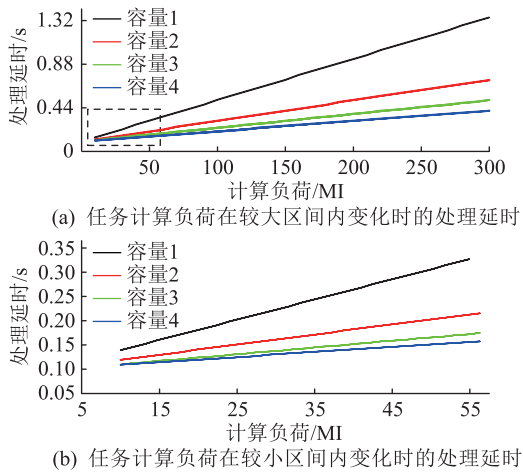


图7 容器资源容量对任务处理延时的影响情况

Fig.7 The effect of container resource capacity on task processing delay

远大于容器的创建延时,即容器的创建延时对处理延时的影响较小,因此在随计算负荷变化时,处理延时呈现出图7(a)所示的线性化的变化情况。

通过数据拟合可以确定任务计算负荷在较小或较大范围内波动时,处理延时的变化情况是一致的。比较各拟合直线的斜率,可以发现容器资源容量以一定倍数增长时,任务的处理速率不以该倍数提高,如容量2、容量3、容量4的处理速率为容量1的1.876倍、2.597倍与3.289倍。而且可进一步确定,随着容器资源容量的增加,资源的效用逐渐降低,如容量2、容量3、容量4的资源效用较容量1分别下降6.6%、15.5%与21.6%。所以随着容器资源容量的提升,任务的总处理延时降低,但容器的创建延时在总延时的占比提高,资源的效用下降。另一方面,若容器已创建,则增大容器资源容量对于降低任务处理延时具有明显效果。为保证系统的安全可靠运行,常按任务所需的最大资源需求进行配置,即按最大方式配置,导致资源总是冗余配置。而系统任务的计算负荷波动进一步加剧了资源利用的不平衡程度。因此,文中通过设置弹性因子灵活配置容器资源以适应任务的计算负荷波动。

### 5.3 弹性因子对任务处理性能的影响分析

为研究资源的弹性配置对任务处理性能的影响,文中首先研究弹性因子的大小对PCIT创建容器数量的影响,如图8所示。容器是PCIT托管不同任务的载体,其数量直接决定了PCIT所能同时处理任务的个数,弹性因子越大,容器资源容量区间越大,但也导致了PCIT中容器数量出现较大波动。图8中,若PCIT分配的资源总是不满足任务处理需求,则将弹性配置用于扩展容器的资源容量,从而造成PCIT中运行的容器数量减少,文中称此时的

配置为最小弹性配置方式,即在该方式下PCIT仅能增加容器的资源容量,使得资源的弹性较小。另一方面,若任务计算负荷减少,则在保证任务处理需求的前提下,PCIT既可选择减少对各容器的资源配置以形成多余的空闲资源区间,从而处理更多并行任务;又可选择增大部分容器的资源容量,从而加速该部分任务的处理,文中称此时的资源配置方式为最大弹性配置方式。

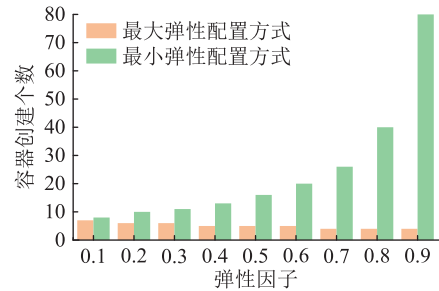


图8 不同弹性因子下PCIT中创建的容器数量

Fig.8 Number of containers created in PCIT with different elasticity factors

其次,文中研究不同弹性因子对PCIT处理多任务的延时的影响,如图9所示。PCIT中处理并行任务所需的时间取决于延时最大的任务,若不同并行任务的计算负荷存在波动,则资源的分配不均将进一步增加并行任务的处理延时,而通过设置合理的弹性因子可有效应对计算负荷波动。图9(a)~(f)展示了弹性因子分别为0.1、0.2、0.3、0.4、0.5和0.6时PCIT处理任务 $CTT_1$ 、 $CTT_2$ 的处理延时变化情况。由图9可知,随着弹性因子的增大,处理延时的不平衡程度由大变小再变大。增加容器的资源容量可有效降低多任务的处理延时,但在最小弹性配置方式下设置较大弹性因子可能导致其他任务的处理延时过大;而设置的弹性因子较小时,又无法有效填补不同负荷的峰谷差,仍将出现一者的处理延时较大的情况。所以,应根据不同任务的计算负荷差异选择适当的弹性因子大小。

### 5.4 任务分配与资源利用分析

PCIT既承担本地STT的处理,又与其他PCIT协同处理各CTT,且任务处理性能的优劣与PCIT承担的任务数量、分配给每一任务的资源容量大小相关,进一步地,与CTT在PCIT间的分配有关。所有可能的任务分配方案共有 $K^{m_2}$ 种,可知随着任务数量的增多,任务分配结果也将剧增。而在下层模型中,若弹性因子的调节步长为0.1,则PCIT<sub>e</sub>将产生

$\left( \sum_{a=1}^{m_2} x_{ae} + m_1 \right)^{30}$ 种资源分配方案。故文中提出利



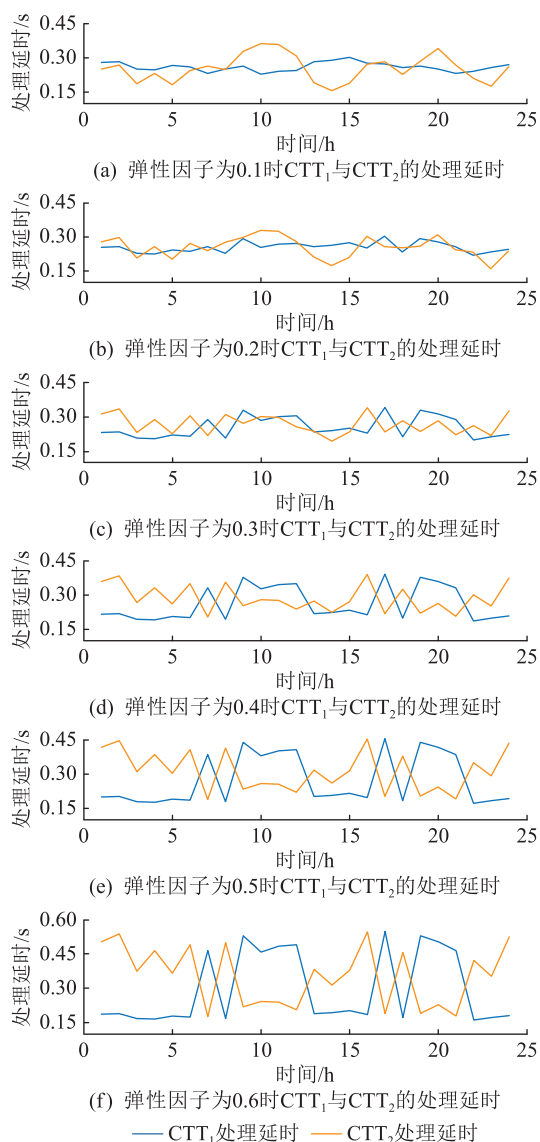


图9 不同弹性因子下处理 CTT<sub>1</sub>与 CTT<sub>2</sub>产生的处理延时

Fig.9 Processing delay curves generated by processing CTT<sub>1</sub> and CTT<sub>2</sub> under different elasticity factors

用隐枚举法求解 PCIT 间任务与资源的优化分配方案,求解方法见 4.2 节。

任务部署于同一 PCIT 有利于减少对通信资源的占用,降低通信延时,但也将导致每一任务在单位时间内占用的计算资源减少,计算延时增加,因此比较两者的相对变化值有利于求解得到双层模型的最优解。图 10 为不同 CTT 分配方案下的处理延时变化曲线,横坐标每一刻度表示一种任务分配方案,方案的编排规则为按各 PCIT、任务编号从小到大依次组合,具体如下:(1) 改变 CTT<sub>5</sub>的分配位置,同时其他任务保持不变,即可得到 6 种分配方案;(2) 改变 CTT<sub>4</sub>的分配位置,每一种 CTT<sub>4</sub>的部署可对应 6 种 CTT<sub>5</sub>的分配结果,即改变 2 个任务时可获得 36 种分配方案;(3) 以此类推,依次改变 CTT

的分配位置,最终可获得 6<sup>5</sup>种分配方案,在获得各种分配方案的延时处理结果之后,按延时从小到大进行排序,得到图 10 所示结果。

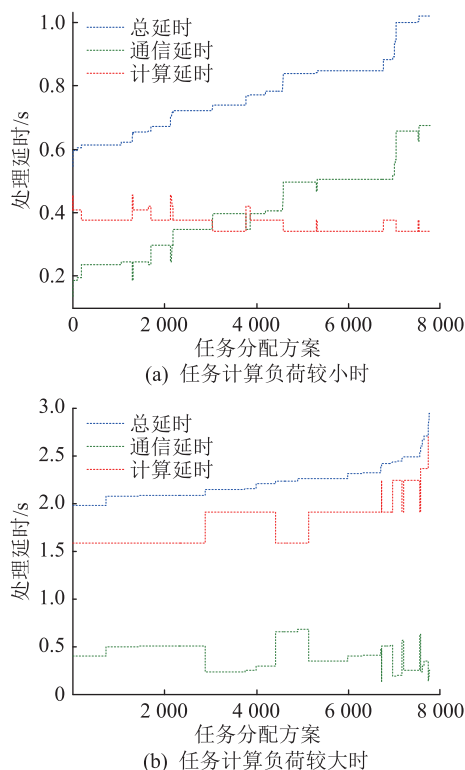


图 10 不同 CTT 分配方案下的处理延时变化曲线

Fig.10 Processing delay variation curves under different CTT allocation schemes

表 4 展示了总处理延时最小的任务分配与任务处理延时结果。

表 4 基于隐枚举法的任务处理情况

Table 4 Task processing based on the implicit enumeration method

任务编号	部署位置	计算延时/s	通信延时/s	平均总延时/s	平均等待延时/s
STT <sub>1</sub>		0.013~0.043	0.020~0.040	0.057 1	0.031 3
STT <sub>2</sub>		0.045~0.107		0.073 0	0.057 1
STT <sub>3</sub>		0.069~0.150		0.106 3	0.130 1
CTT <sub>1</sub>	PCIT <sub>1</sub>	0.771~1.275	0.055~0.065	1.062 5	0.241 0
CTT <sub>2</sub>	PCIT <sub>1</sub>	0.789~1.263	0.040~0.050	1.043 5	0.241 0
CTT <sub>3</sub>	PCIT <sub>3</sub>	0.659~0.943	0.055~0.065	0.855 8	1.302 5
CTT <sub>4</sub>	PCIT <sub>3</sub>	0.751~0.898	0.045~0.055	0.867 3	1.302 5
CTT <sub>5</sub>	PCIT <sub>1</sub>	0.365~0.588	0.045~0.055	0.495 3	1.302 5

在求解双层模型的 0-1 规划问题时,文中采用基于隐枚举法的求解方法,其相较于单层模型而言,求解结果一致但所需的求解时间明显降低,耗时降低约 45.01%~51.44%,而计算模型所得的理论延时与实际 PCIT 测量的延时的误差范围约为 0.11%~0.98%。此外,文中比较了任务的待处理数

据量增大时,通信延时与计算延时的增长情况,如图 10 所示。在图 10(a)中,总延时与通信延时的变化趋势具有一定的相似性;而当使待处理数据量增大时,不同分配方案下的延时变化情况呈现一定的差异性,如图 10(b)所示,此时总延时变化趋势与通信延时不再呈现一定的关联性。因此,不同分配方案下处理任务的计算延时、通信延时变化情况不同,一般体现为部署于同一台 PCIT 时计算延时增大而通信延时减小,部署于不同 PCIT 时计算延时减小而通信延时增大。当任务的数据量较小时,不同分配方案中计算延时的差异较小,导致通信延时在总延时中的比重增加;而当数据量增大时,不同方案中计算延时的差异愈发明显,即计算延时对总延时的影响程度增大,计算延时所增加的比重约为 19.24%。

进一步地,将文献[11,31]中所采用的任务处理方式下 PCIT 的资源利用与任务处理情况与文中对比,结果如表 5 所示,并利用中央处理器(central processing unit,CPU)表示计算资源。在不考虑弹性容器资源下,为保证任务满足所需的延时容忍度,须提高托管任务的容器资源容量。而选用资源容量较大的容器将导致过多占用 PCIT 的资源,并不利于计算负荷变化较大的场景,所以相较文献[11],文中在任务处理时的平均资源占用率降低约 4.22%~74.54%,任务的等待延时降低约 20.16%,处理延时提高约 2.48%。

表 5 不同任务处理方式下资源利用与任务处理对比

Table 5 Comparison of resource utilization and task processing under different task processing methods

任务处理方式	平均资源占用率/%			平均等待 延时/s	平均处理 延时/s	
	CPU	内存	带宽			
PCIT 协同 处理	弹性容器 资源	46.94	2.08	33.38	1.762	2.932
	固定容器 资源	49.01	8.17	33.38	2.207	2.861
本地 PCIT 处理	弹性容器 资源	59.00	12.50	23.86	1.238	3.058
	固定容器 资源	65.97	65.97	23.86	1.981	2.866

相较于文献[31]的本地化处理、不考虑 PCIT 协同,文中所提方法在 PCIT 的平均资源占用率有明显降低,如减少占用计算资源约 28.85%、内存资源约 96.85%,且平均处理延时最多可减少 4.12%。任务的本地化处理虽减少了不同区域间的数据传输,降低了对带宽资源的占用,但增加了本地 PCIT 计算与内存资源的占用,而且部分任务在各台 PCIT

被重复计算,增加了任务的处理延时。因此,相较于本地处理方式,文中所提方法可以减少配电网约 20%的 PCIT 资源的投入,即在相同资源配置下可承受 20%的任务计算负荷增长。

## 6 结论

随着配电网接入设备、用户类型与数量剧增,愈发复杂的多元化任务给 PCIT 的资源利用带来新的挑战,因此文中提出了一种考虑资源弹性配置的配电网 PCIT 协同任务优化分配方法,阐述了 PCIT 协同架构体系,并提出了任务分配与容器资源弹性配置的双层优化模型,得到如下结论:

(1) 配电网保护控制任务模型是刻画 PCIT 处理不同任务时所需的资源容量的有效手段,任务的计算负荷、耦合关系从侧面反映了 PCIT 资源利用随时间的变化趋势。

(2) PCIT 协同的任务处理方式提高了使用区域内资源的灵活性,通过将任务分配至相邻的 PCIT,既减少了任务的处理延时,又优化了信息流的分布。文中所提方法为实际工程应用中所属同一区域的 PCIT 资源的协同优化提供了重要参考。

(3) 通过容器技术手段实现了对 PCIT 资源的弹性配置,在避免资源的冗余配置上具有显著效果,验证了容器化的 PCIT 在新型电力系统发展建设过程中的有效性与可行性。

### 参考文献:

- [1] 秦强,刘文泽,谭炜豪,等. 面向软件定义网络的配电网边缘计算终端优化部署方法[J]. 电力建设,2023,44(1):82-90.
- [2] 刘向军,周凯,孙毅,等. 面向新型电力系统的边缘缓存与计算资源迁移策略[J]. 电力系统保护与控制,2023,51(24):134-142.
- [3] LIU Xiangjun,ZHOU Kai,SUN Yi,et al. Edge cache and computing resource migration strategy for new power systems[J]. Power System Protection and Control,2023,51(24):134-142.
- [4] LIMA P M,CARVALHO L K,MOREIRA M V. Ensuring confidentiality of cyber-physical systems using event-based cryptography[J]. Information Sciences,2023,621:119-135.
- [5] TAO J S,UMAIR M,ALI M,et al. The impact of Internet of Things supported by emerging 5G in power systems:a review[J]. CSEE Journal of Power and Energy Systems,2020,6(2):344-352.
- [6] 苏盛,汪干,刘亮,等. 电力物联网终端安全防护研究综述[J]. 高电压技术,2022,48(2):513-525.
- [7] SU Sheng,WANG Gan,LIU Liang,et al. Review on security of

- power Internet of Things terminals[J]. High Voltage Engineering, 2022, 48(2): 513-525.
- [6] 刘媛媛, 蔡泽祥, 陈元樑, 等. 基于容器的配电网保护控制终端业务调度模型与策略分析[J]. 电力建设, 2023, 44(10): 95-106.
- LIU Yuanyuan, CAI Zexiang, CHEN Yuanju, et al. Business scheduling model and strategy analysis of distribution network protection control terminal based on container[J]. Electric Power Construction, 2023, 44(10): 95-106.
- [7] 司徒友, 苏俊妮, 张鑫, 等. 基于 K8s 的配电边缘计算终端资源动态配置方法[J]. 供用电, 2023, 40(8): 49-57.
- SITU You, SU Junni, ZHANG Xin, et al. Resource dynamic allocation method in distribution edge computing terminal based on K8s[J]. Distribution & Utilization, 2023, 40(8): 49-57.
- [8] DA SILVA T B, DOS SANTOS CHAIB R P, ARISMAR C S, et al. Toward future Internet of Things experimentation and evaluation[J]. IEEE Internet of Things Journal, 2022, 9(11): 8469-8484.
- [9] SUN Y Y, CAI Z X, GUO C S, et al. Collaborative dynamic task allocation with demand response in cloud-assisted multi-edge system for smart grids[J]. IEEE Internet of Things Journal, 2022, 9(4): 3112-3124.
- [10] 陈元樑, 蔡泽祥, 岑伯维, 等. 考虑任务耦合的配电网边缘计算终端部署与任务分配方法[J]. 电力系统自动化, 2022, 46(22): 32-40.
- CHEN Yuanju, CAI Zexiang, CEN BOWEI, et al. Deployment and task allocation method of edge computing terminals in distribution network considering task coupling[J]. Automation of Electric Power Systems, 2022, 46(22): 32-40.
- [11] 陈元樑, 蔡泽祥, 孙宇嫣, 等. 电力物联网边缘计算终端部署与业务分配优化方法[J]. 南方电网技术, 2022, 16(3): 100-107.
- CHEN Yuanju, CAI Zexiang, SUN YUYAN, et al. An optimization method of edge computing terminal deployment and service allocation in power Internet of Things[J]. Southern Power System Technology, 2022, 16(3): 100-107.
- [12] ZHANG W Y, ZHANG Z J, ZEADALLY S, et al. Energy-efficient workload allocation and computation resource configuration in distributed cloud/edge computing systems with stochastic workloads[J]. IEEE Journal on Selected Areas in Communications, 2020, 38(6): 1118-1132.
- [13] 岑伯维, 蔡泽祥, 武志刚, 等. 电力物联网边缘计算终端的微服务建模与计算资源配置方法[J]. 电力系统自动化, 2022, 46(5): 78-91.
- CEN BOWEI, CAI ZEXIANG, WU ZHIGANG, et al. Microservice modeling and computing resource configuration method for edge computing terminal in electric Internet of Things[J]. Automation of Electric Power Systems, 2022, 46(5): 78-91.
- [14] 周大成, 陈鸿昶, 何威振, 等. 基于深度强化学习的微服务多维动态防御策略研究[J]. 通信学报, 2023, 44(4): 50-63.
- ZHOU Dacheng, CHEN Hongchang, HE Weizhen, et al. Research on multidimensional dynamic defense strategy for microservice based on deep reinforcement learning[J]. Journal on Communications, 2023, 44(4): 50-63.
- [15] 汤小春, 赵全, 符莹, 等. 面向 Dataflow 的异构集群混合式资源调度框架研究[J]. 软件学报, 2022, 33(12): 4704-4726.
- TANG Xiaochun, ZHAO Quan, FU Ying, et al. Research of hybrid resource scheduling framework of heterogeneous clusters for Dataflow[J]. Journal of Software, 2022, 33(12): 4704-4726.
- [16] IMPRAM S, VARBAK NESE S, ORAL B. Challenges of renewable energy penetration on power system flexibility: a survey[J]. Energy Strategy Reviews, 2020, 31: 100539.
- [17] MANSOURI S A, AHMARINEJAD A, NEMATBAKHS E, et al. A sustainable framework for multi-microgrids energy management in automated distribution network by considering smart homes and high penetration of renewable energy resources[J]. Energy, 2022, 245: 123228.
- [18] KIM H, CHOI H, KANG H, et al. A systematic review of the smart energy conservation system; from smart homes to sustainable smart cities[J]. Renewable and Sustainable Energy Reviews, 2021, 140: 110755.
- [19] NING Z L, ZHANG K Y, WANG X J, et al. Intelligent edge computing in Internet of vehicles: a joint computation offloading and caching solution[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(4): 2212-2225.
- [20] DONG L, HE W J, YAO H P. Task offloading and resource allocation for tasks with varied requirements in mobile edge computing networks[J]. Electronics, 2023, 12(2): 366.
- [21] 吴逸文, 张洋, 王涛, 等. 从 Docker 容器看容器技术的发展: 一种系统文献综述的视角[J]. 软件学报, 2023, 34(12): 5527-5551.
- WU Yiwen, ZHANG Yang, WANG Tao, et al. Development exploration of container technology through Docker containers: a systematic literature review perspective[J]. Journal of Software, 2023, 34(12): 5527-5551.
- [22] 林志超, 钟红梅, 韦园清, 等. 基于容器的配网保护微服务建模与计算资源调度方法[J]. 电力科学与技术学报, 2022, 37(5): 174-180, 197.
- LIN Zhichao, ZHONG Hongmei, WEI Yuanqing, et al. Container-based microservice modeling and computing resource scheduling method for distribution network protection[J]. Journal of Electric Power Science and Technology, 2022, 37(5): 174-180, 197.
- [23] 李竹根, 刘波峰, 张晓飞, 等. 基于改进灰狼算法与隐枚举法的配电网分区故障定位研究[J]. 电测与仪表, 2024, 61(8): 157-165.
- LI Zhugen, LIU Bofeng, ZHANG Xiaofei, et al. Research on sub-region fault location of distribution network based on improved grey wolf optimizer and implicit enumeration method[J]. Electrical Measurement & Instrumentation, 2024, 61(8): 157-165.
- [24] 龚定飞. 基于匈牙利法的船舶人员疏散规划模型研究与软件开发[D]. 武汉: 华中科技大学, 2022.

- GONG Dingfei. Research on ship passenger evacuation planning model and software development based on Hungarian algorithm[D]. Wuhan:Huazhong University of Science and Technology,2022.
- [25] 霍佳丽,叶辛頔,何卫斌,等. 考虑广义负荷的配电网弹性电力电量平衡策略研究[J]. 浙江电力,2022,41(1):19-25. HUO Jiali, YE Xindi, HE Weibin, et al. Study of an elastic electric power and energy balance strategy for distribution networks considering generalized loads [J]. Zhejiang Electric Power,2022,41(1):19-25.
- [26] 曹望璋,李彬,祁兵,等. 一种考虑业务可靠性的需求响应边缘云部署方法[J]. 中国电机工程学报,2021,41(3):846-857. CAO Wangzhang, LI Bin, QI Bing, et al. A deployment method of demand response edge cloud considering services reliability [J]. Proceedings of the CSEE,2021,41(3):846-857.
- [27] 郑竣杰,岑伯维,蔡泽祥,等. 基于 EdgeCloudSim 的配电网边缘计算终端建模与性能分析[J]. 电力系统自动化,2022,46(23):51-58. ZHENG Junjie, CEN BOWEI, CAI ZEXIANG, et al. EdgeCloudSim-based modeling and performance analysis of edge computing terminals in distribution network[J]. Automation of Electric Power Systems,2022,46(23):51-58.
- [28] 陈沛东,曹华珍,何璇,等. 中压配电网近邻交互式分布式拓扑辨识算法[J]. 电力工程技术,2023,42(2):139-146. CHEN Peidong, CAO Huazhen, HE Xuan, et al. Distributed topology identification algorithm of medium-voltage distribution network based on neighboring interaction[J]. Electric Power Engineering Technology,2023,42(2):139-146.
- [29] 高崇,陈沛东,曹华珍,等. 中压配电网分布式智能后备保护方案[J]. 南方电网技术,2023,17(8):77-84. GAO Chong, CHEN Peidong, CAO Huazhen, et al. Distributed intelligent backup protection scheme of medium voltage distribution network[J]. Southern Power System Technology,2023,17(8):77-84.
- [30] 蒋博天. 基于节省存储和快速构图的 MLCS 分支定界法[D]. 西安:西安电子科技大学,2022. JIANG Botian. Two MLCS branch-and-bound algorithms based on memory saving or fast construction of graph[D]. Xi'an:Xi-dian University,2022.
- [31] 康逸群,蔡泽祥,曾兴,等. 考虑卸载成本的配电网能量管理业务云边协同调度方法[J]. 南方电网技术,2021,15(9):61-68. KANG Yiqun, CAI Zexiang, ZENG Xing, et al. An offloading cost based cloud-edge collaborative scheduling method for power distribution network energy management applications [J]. Southern Power System Technology,2021,15(9):61-68.

#### 作者简介:



刘媛媛

刘媛媛(1999),女,硕士,研究方向为配电网保护与控制、电力物联网、边缘计算(E-mail:lly990904@163.com);

陈元桦(1997),男,硕士,从事电力物联网、云计算、边缘计算工作;

蔡泽祥(1960),男,博士,教授,研究方向为保护与控制、电力物联网、云计算、边缘计算。

## Collaborative task allocation method for protection and control intelligent terminal in distribution networks considering elastic allocation of resources

LIU Yuanyuan<sup>1</sup>, CHEN Yuanju<sup>2</sup>, CAI Zexiang<sup>1</sup>, LIU WENZE<sup>1</sup>, QU Jing<sup>1</sup>, HU Kaiqiang<sup>1</sup>

(1. School of Electric Power Engineering, South China University of Technology, Guangzhou 510641, China;

2. China Southern Power Grid Power Technology Co., Ltd., Guangzhou 510630, China)

**Abstract:** With the development of new power systems, the protection and control tasks of distribution networks have become increasingly complex. When intelligent terminals are employed to handle these tasks, the requirements for balancing resource supply and demand have become more stringent. Therefore, a collaborative task allocation method for protection and control intelligent terminals (PCIT) in distribution networks considering elastic allocation of resources is proposed in this paper. Firstly, the technical architecture of multi-terminal collaboration is elaborated, and the container-based elastic resource model and task processing model within the terminal are established. Secondly, a two-layer model is proposed to optimize the cooperative assignment of protection control tasks among terminals and the elastic scheduling of resources. The implicit enumeration method is used to solve the model, so as to give full play to the flexibility of resources and improve the performance of task processing. Finally, the arithmetic example verifies the feasibility and advancement of the proposed method. The occupancy rate of computing resources of each intelligent terminal is reduced by about 28.85%, and the average processing delay of tasks is reduced by about 4.12%.

**Keywords:** resource elastic allocation; protection and control intelligent terminal (PCIT); container; collaborative optimization; double-layer model; implicit enumeration method

(编辑 陆海霞)