

## 工业物联网中大规模受损边缘计算网络修复机制

田辉<sup>1</sup>, 伍浩<sup>1</sup>, 田洋<sup>1</sup>, 任建阳<sup>1</sup>, 崔亚娟<sup>1</sup>, 艾文宝<sup>2</sup>, 袁健华<sup>2</sup>

(1. 北京邮电大学网络与交换技术国家重点实验室, 北京 100876; 2. 北京邮电大学理学院, 北京 100876)

**摘 要:** 针对工业物联网中边缘计算网络与其余子网的相互依赖特性所导致的网络大规模级联故障问题, 考虑到网络修复初期的资源有限性, 提出了一种联合考量计算需求与修复开销的网络修复机制。考虑到受损网络结构(拓扑关系与链路容量)和动态特征(边缘计算节点计算需求), 基于节点计算量守恒定理构建了链路修复策略集与网络计算迁移的联合分析框架。基于 Benders 分解算法, 将原 NP-hard 问题转化为相互依赖的主问题与子问题, 通过割平面的不断逼近, 实现对原问题最优解在多项式时间内的高效探索。结合局部分支法, 进一步保障 Benders 分解算法的上界在迭代过程中的非增特性, 加快算法收敛速度。仿真结果表明, 所提算法的系统总开销性能优于传统基于拓扑结构的修复算法, 并且可以在多场景下保持其性能优势。

**关键词:** 工业物联网; 边缘计算; 网络修复; Benders 分解算法; 局部分支法

**中图分类号:** TN92

**文献标识码:** A

**DOI:** 10.11959/j.issn.1000-436x.2021100

## Recovery mechanism of large-scale damaged edge computing network in industrial Internet of things

TIAN Hui<sup>1</sup>, WU Hao<sup>1</sup>, TIAN Yang<sup>1</sup>, REN Jianyang<sup>1</sup>, CUI Yajuan<sup>1</sup>, AI Wenbao<sup>2</sup>, YUAN Jianhua<sup>2</sup>

1. State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

2. School of Sciences, Beijing University of Posts and Telecommunications, Beijing 100876, China

**Abstract:** Given the limited resources at early stages for recovery, a failure recovery mechanism of the edge computing network considering both computational demands and repair costs was proposed, which intends to tackle the problem of the high probability of large-scale cascading failure caused by the interdependence between the edge computing network and other subnetworks in industrial Internet of things (IIoT). Considering the network structure (topology and link capacity) and network dynamics (computational demands), a joint link recovery selection and computation migration optimization problem was formulated under the conservation of node computing requirements. By leveraging the Benders decomposition algorithm, the NP-hard problem was transformed into a main problem and a sub-problem, which were interdependent and could be solved in polynomial time through the approximation of cutting planes. A local branching method was further introduced to guarantee the non-increasing nature of the Benders upper bound, thus accelerating the convergence of Benders decomposition. Simulation results demonstrate that the proposed algorithm outperforms the conventional topology-based recovery algorithm in system utility, and can perform well in multiple scenarios.

**Keywords:** industrial Internet of things, edge computing, network recovery, Benders decomposition algorithm, local branching

收稿日期: 2021-01-11; 修回日期: 2021-04-08

基金项目: 国家自然科学基金资助项目 (No.62071068); 北京邮电大学优秀博士生创新基金资助项目 (No.CX2019108)

**Foundation Items:** The National Natural Science Foundation of China (No.62071068), The BUPT Excellent Ph.D. Students Foundation (No.CX2019108)

## 1 引言

工业物联网 (IIoT, industrial Internet of things) 被视为全球工业体系智能化变革的重要推手。依靠数以亿计无缝部署的感知器、采集器与控制器, 工业物联网可以实现对生产制造过程全周期的模拟、预测与控制<sup>[1]</sup>。作为工业物联网的“大脑”, 边缘计算网络为无线采集设备提供了更充足的计算处理能力, 有效降低了处理、传输时延, 为数字孪生 (DT, digital twin)<sup>[2]</sup>、虚拟现实 (VR, virtual reality) 等企业高层应用打下了坚实的基础<sup>[3]</sup>。同时, 边缘计算节点间有线的链路连接方式, 也使节点间的计算任务迁移更加顺畅, 有效缓解了工业物联网计算需求空时波动所造成的计算资源空时分配不均衡问题。

边缘计算网络的正常、稳定运行是工业物联网高效运转的关键, 一旦“大脑”受损, 工业物联网系统将失去对“四肢”(例如供应链监控、数据可视化分析)的有效控制, 带来难以计数的经济损失, 甚至威胁生命。然而, 边缘计算网络的稳定性需求面临内外两方面的挑战。一方面, 边缘计算网络与工业物联网中的电网、控制网等多种子网相互耦合, 形成了极易受损的互依赖网络<sup>[4]</sup>。其他子网的任意微小波动, 都有可能传导到边缘计算网络, 造成大规模的系统级联故障。另一方面, 自然灾害与人为攻击等不可预测事件也随时考验着边缘计算网络的稳健性<sup>[5]</sup>。为应对上述挑战, 一方面可以加强边缘计算网络的可靠性, 使其能够自适应地应对各种网络波动, 防止网络出现故障; 另一方面, 也是更重要的, 需要探究网络受损后的快速修复机制, 使网络性能尽快恢复到接近受损前的水平。

尽管修复机制的设计对网络可持续稳定运行至关重要, 当前尚无特定针对工业物联网中边缘计算网络场景的研究文献。鉴于网络拓扑、网络动态等特征的相似性, 部分现存网络修复策略仍可以为工业物联网中边缘计算网络修复机制设计提供一定的借鉴。现存网络修复机制的研究主要着重于局部网络受损后的快速修复。文献[6]将网络单一节点或链路受损问题构建为一个整数线性规划问题, 提出了一种数据迁移感知的修复模型, 实现了服务中断率与修复成本的有效均衡。进一步地, 文献[7-8]考虑到多节点失效场景网络连通性受损问题, 提出可以通过设备到设备 (D2D, device to device) 方式

利用用户作为失连边缘节点间的中转节点<sup>[7]</sup>或者利用可移动接入节点实现网络节点间数据的处处可达<sup>[8]</sup>, 保障受损网络的连通性。与上述研究不同, 文献[9]考虑到网络受攻击情况下的持续性受损状态, 将网络动态修复问题转化为微分博弈论问题, 并通过纳什均衡必要条件与竞争策略集增强了网络修复能力。然而, 上述针对局部网络修复的研究, 往往忽略了全局网络节点、链路间的动态特征 (例如流迁移) 和实际场景限制 (例如链路空间布局无法改动), 难以有效扩展至工业物联网中更可能发生的网络大规模受损场景。

网络大规模受损后, 初期可利用的修复资源 (例如修理人员数量、可替换设备数量) 往往是有限的。如何有效平衡修复初期受限的系统修复资源与急迫的系统性能恢复需求是当前工业物联网中亟待解决的问题。当前研究主要集中于对网络拓扑结构的分析。文献[10]认为, 大度节点, 即节点度大的节点, 在网络连通性方面有着更重要的作用, 需要优先修复。类似地, 文献[11]考虑链路受损情况下, 应优先修复介数中心性 (BC, betweenness centrality) 大的链路。文献[12]通过实际网络数据分析发现, 网络中的弱连接节点, 即自身的度不高但连接着数个大度节点的节点, 在网络连通性上发挥着最关键的作用, 其修复优先级应比大度节点更高。然而, 上述基于网络连通性不断壮大最大连通子图的方案均是静态网络架构分析, 忽略了网络的动态性特征。真实环境下的网络大规模修复往往是先形成相互独立的子图, 然后将多子图连接形成最大连通图<sup>[13]</sup>。因此, 工程分析中, 需要融入更多的网络设备细节与实际传输动态分析<sup>[14]</sup>。这类问题也被称为网络设计问题 (NDP, network design problem)。由于网络动态性特征的加入, 网络设计问题的复杂性极高 (至少是 NP 完全问题), 传统动态规划算法将引发“维度灾难”<sup>[15]</sup>。为解决上述问题, 文献[16]提出了一种更易求解的启发式算法, 然而, 其缺乏网络宏观分析, 算法性能方差大, 无法提供可靠的性能保障。类似地, 模拟退火算法<sup>[17]</sup>与遗传算法<sup>[18]</sup>均面临一般启发式算法的相同困境, 且极易陷入局部最优解。尽管通过爬山算法<sup>[19]</sup>或梯度下降算法<sup>[20]</sup>可以方便地跳出局部最优点并极大降低问题计算复杂度, 但解的最优性仍无法得到保障。

针对当前研究所面临的现实困境, 本文提出了一种针对工业物联网环境下受损边缘计算网络的

修复机制。区别于现有文献，本文基于工业物联网子网间的互依赖特性所造成的网络脆弱性，深入挖掘了大规模受损边缘计算网络的结构特性（拓扑关系、链路容量）与动态特性（节点计算需求），联合考虑了链路优先修复集合决策与网络计算迁移问题，以期实现网络修复初期计算需求与修复开销的高效均衡。本文主要贡献总结如下。

1) 提出了一种边缘计算网络大规模受损情况下的网络修复机制。结合边缘计算网络结构与动态特征，提供了一种网络修复初期优先修复集合决策与资源调度分析框架。

2) 基于 Benders 分解算法，将复杂的混合整数问题分解为更易求解的主问题与子问题两部分，通过各自解集的相互迭代，逐渐逼近系统最优解。针对多变量组的子问题，通过添加虚拟源节点与目的节点，将问题转化为网络最大流问题进行求解。

3) 设计了一种基于局部分支的 Benders 分解加速算法。采用基于汉明距离的信赖域，收缩可行域的搜索范围，加速算法的收敛速度。

仿真结果表明，本文所提算法具有较好的收敛性能与较低的系统总开销。相对现有随机修复、最大连通图修复和介数中心性排序修复算法等基于拓扑结构的修复算法，本文所提算法在多场景下均具有较优表现和具有很好的可扩展性与适应性。

## 2 系统模型

针对工业物联网中边缘计算网络，考虑一个含有  $N$  个节点的边缘计算网络，用集合  $\forall i \in \mathcal{N} = \{1, 2, \dots, N\}$  表示。边缘节点之间均通过有线链路连接，链路集合用  $E$  表示，网络中总链路数量为  $|E|$ 。由于有线链路通常是可靠的，且相对边缘节点的计算任务只需要少量的信道编码复杂性，2 个边缘节点间的链路可以认为是无差错的<sup>[21]</sup>。系统参数如表 1 所示。值得注意的是，实际工业物联网中边缘计算网络往往是由有线链路和无线链路组成的混合链路传输网络。由于无线链路几乎不存在受损情况，且其修复成本相对于有线链路可以忽略不计，本文只考虑所有传输链路均为有线链路的情况。尽管如此，如果不考虑短期无线信道的深衰弱，静态场景下的无线链路（不考虑修复过程中的频谱重分配）可视为容量不变的不可受损链路，混合链路传输网络可以等效为纯有线链路网络进行考虑，本文所提算法仍将适用。

表 1 系统参数

参数	含义
$N$	边缘计算网络节点数量
$E$	边缘计算网络所有链路集合
$E^0$	边缘计算网络所有受损链路集合
$E^1$	边缘计算网络所有正常链路集合
$q$	受损链路占所有链路的百分比
$c_{ij}$	链路 $ij$ 的修复成本
$c_i$	节点 $i$ 丢弃每单位数据的代价
$p_i$	节点 $i$ 的本地计算数据计算量
$\bar{p}_i$	节点 $i$ 的最大计算能力
$r_i$	节点 $i$ 的本地数据计算量需求
$d_i$	节点 $i$ 丢弃的计算数据量
$f_{ij}$	节点 $i$ 到节点 $j$ 的计算迁移量
$\bar{f}_{ij}$	链路 $ij$ 的链路容量
$e_{ij}$	是否修复链路 $ij$ 的决策变量
$t$	Benders 分解的迭代次数
$\theta^t$	子问题在第 $t$ 次迭代中的目标函数的最优值
$\Delta^t$	第 $t$ 次迭代时的左枝问题阈值

受损边缘计算网络拓扑关系如图 1 所示。在实际网络中，由于边缘节点常拥有复杂的自我保护机制（例如过热保护），通常不易发生损坏，因此本文不考虑节点损坏的情况，重点关注链路受损问题。为模拟因自然灾害等因素而导致大面积网络故障的状态，假设初始时刻网络中有  $q|E|$  条有线链路遭到破坏，其中  $q$  表示受损链路占所有链路的百分比，集合  $E^0$  表示受损链路集合， $E^1 = E \setminus E^0$  表示仍可以正常工作的链路集合。链路的受损将导致边缘节点间平衡的计算迁移流被打破，甚至形成计算孤岛（无法进行计算迁移，如图 1 中节点 1 所示），从而造成计算需求与计算能力不匹配，影响网络计算性能。受限于网络修复初期的有限修复资源（例如修理人员数量、可替换设备库存），同时修复所有的受损链路不切实际。为使网络状态尽快得到恢复，可以根据网络计算需求分布，优先修复部分受损链路，用集合  $E^t (E^t \subseteq E^0)$  表示。对于任意链路  $ij \in E^0$ ，由于受损程度不同，对其进行维护、检修、替换的成本  $c_{ij}$  也各不相同。

考虑边缘节点  $i \in \mathcal{N}$  的本地数据计算需求为  $r_i$ ，本地实际计算量为  $p_i$ 。需要注意的是， $r_i$  为网络修复初期阶段边缘节点  $i$  的总计算到达量。对于给定场景， $r_i$  是一个不随时间变化的固定值，可以

根据历史计算需求进行较准确的估算。对于任意边缘节点  $i$ ,  $p_i$  满足

$$0 \leq p_i \leq \bar{p}_i, \forall i \in \mathcal{N} \quad (1)$$

其中,  $\bar{p}_i$  为节点  $i$  的最大计算能力。

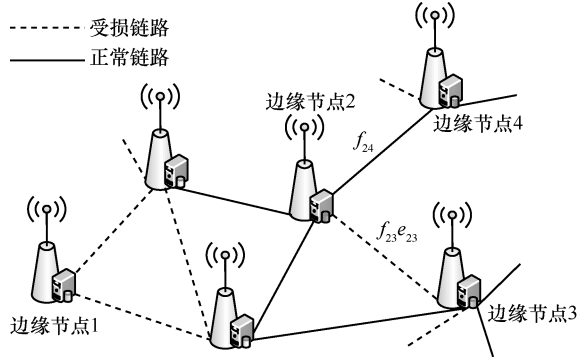


图 1 受损边缘计算网络拓扑关系

令节点  $i$  与节点  $j \in \mathcal{N} \setminus \{i\}$  之间的迁移计算量为  $f_{ij}$ 。若  $f_{ij} > 0$ , 数据计算任务由节点  $i$  迁移至节点  $j$ ; 若  $f_{ij} < 0$ , 数据计算任务由节点  $j$  迁移至节点  $i$ 。受有线链路容量  $\bar{f}_{ij}$  限制, 链路  $ij$  实际计算迁移量满足

$$|f_{ij}| \leq \bar{f}_{ij}, \forall ij \in E \quad (2)$$

进一步地, 对于受损链路, 其实际计算迁移量不仅取决于有线链路容量, 也受到链路修复决策的影响, 即

$$|f_{ij}| \leq \bar{f}_{ij} e_{ij}, \forall ij \in E^0 \quad (3)$$

其中, 有

$$e_{ij} \in \{0, 1\}, \forall ij \in E^0 \quad (4)$$

其中,  $e_{ij} = 1$  表示优先修复链路  $ij$ , 即  $ij \in E^r$ ; 当  $e_{ij} = 0$  表示链路  $ij$  未被修复, 其实际计算迁移量  $f_{ij} = 0$ 。

此外, 对于单一链路  $ij$ , 由迁移流对称性可知

$$f_{ij} = -f_{ji}, \forall ij \in E \quad (5)$$

当集合  $E^r$  中链路得到修复时, 根据节点的计算量守恒定律可知

$$p_i + \sum_{ij \in E^0} f_{ij} + \sum_{ij \in E^r} f_{ij} + d_i = r_i, \quad i \in \mathcal{N} \quad (6)$$

其中,  $r_i$  表示节点  $i$  的本地数据计算量需求;  $d_i$  表示因计算能力有限导致数据积压, 边缘节点  $i$  不得不舍弃的计算任务量, 有

$$d_i \geq 0, \forall i \in \mathcal{N} \quad (7)$$

为了修复尽可能多的链路, 可以减少数据的丢弃量, 提升系统性能, 但会造成较大的系统修复开销。修复链路过少, 系统修复开销固然变小, 但可能造成数据因堆积在本地无法处理而被丢弃, 损害网络性能。为进行量化分析, 本文中网络性能通过系统数据丢弃总代价来进行衡量, 数据丢弃量越大, 数据丢弃总代价越高, 网络性能越差。因此, 为均衡网络修复初期的修复开销与网络性能 (数据丢弃总代价), 可以构建如下的系统总开销最小化问题

$$\begin{aligned} P: \min_{e, d, f, p} \phi &= \sum_{ij \in E^0} c_{ij} e_{ij} + \sum_{i \in \mathcal{N}} c_i d_i \\ \text{s.t.} & \text{式(1)~式(7)} \end{aligned}$$

其中,  $c_{ij}$  表示修复链路  $ij \in E^0$  所需要的修复成本,  $c_i$  表示节点  $i \in \mathcal{N}$  丢弃每单位数据的代价, 链路修复决策向量  $e = (e_{ij}, \forall ij \in E^0)$ , 数据丢弃决策向量  $d = (d_i, \forall i \in \mathcal{N})$ , 流分配向量  $f = (f_{ij}, \forall ij \in E)$ , 本地实际计算量向量  $p = (p_i, \forall i \in \mathcal{N})$ 。问题 P 中, 系统总开销包含两部分内容: 系统修复开销与数据丢弃总代价。如前文所述, 系统修复开销  $\sum_{ij \in E^0} c_{ij} e_{ij}$  与数据丢弃总代价  $\sum_{i \in \mathcal{N}} c_i d_i$  是一对相互矛盾的量, 一个

值的增加将导致另一个值的减少, 最小化它们的和可以有效均衡两者影响。当链路修复成本  $c_{ij}$  较大时, 表明计算数据重要性相对不高, 系统倾向于暂时修复较少的受损链路, 并丢弃无法处理的数据; 而当每单位数据丢弃代价  $c_i$  较大时, 系统倾向于修复更多的受损链路以减少计算数据的丢弃。

问题 P 是一个混合整数问题, 是 NP-hard 问题, 且其数据规模较大, 目前还没有已知的多项式时间算法能够解决上述问题。当前求解上述问题的方法可以分为三大类: 启发式算法<sup>[22]</sup>、近似算法<sup>[23]</sup>与精确算法<sup>[24]</sup>。启发式算法速度快、应用简单, 但缺乏严格的理论证明, 结果常常严重偏离最优解。近似算法, 例如松弛变量法, 其求解规模有限, 松弛问题的解也不能准确描述原问题最优解。与上述 2 种方法不同, 精确算法, 例如割平面算法, 通过割平面的迭代更新, 可以很好地探索问题最优解, 在混合整数问题求解过程中被广泛采用。

### 3 算法设计

Benders 分解算法<sup>[25]</sup>是一种经典的割平面算

法，被广泛用于处理现实中的混合整数规划问题（例如机车调度、航空路线规划）。该算法既不会像分支定界法那样迭代次数随着运算变量的增多而显著增加，也不会像动态规划那样产生维度灾难，同时不会出现启发式、模拟退火等算法的巨大方差<sup>[26]</sup>。本节将基于 Benders 分解算法，给出问题 P 的高效解决方案。

### 3.1 子问题描述与转换

在 Benders 分解算法中，原问题可以被分解为主问题与子问题两部分，主问题中的优化变量称为复杂变量。当复杂变量固定后，原问题中剩下的优化问题（即子问题）变得相对容易求解。对于问题 P，若给定第  $t$  次迭代过程中的复杂变量  $e_{ij}^t$  的值，则子问题可以表示为

$$S: \min_{d, f, p} \theta = \sum_{i \in \mathcal{N}} c_i d_i$$

s.t. 式(1)~式(2), 式(5)~式(7)

$$|f_{ij}| \leq \bar{f}_{ij} e_{ij}^t, \forall ij \in E^0 \quad (8)$$

由于原问题 P 中的 0-1 变量  $e_{ij}$  被分解到主问题中，上述子问题中优化变量均为连续变量，该问题可以等效为一个最小费用流问题。本文考虑工业物联网中典型的环境监控场景，边缘网络中各边缘节点负责处理无线传感器采集的环境数据。在环境监控场景下，各边缘节点处的计算任务有着相同的计算优先级<sup>[3, 15]</sup>，即节点丢弃每单位数据的代价相同 ( $c_i = c_j, \forall i, j \in \mathcal{N}$ )。从而问题可进一步转化为网络最大流问题<sup>[27]</sup>。图 2 通过一个含有 4 个节点的边缘网络对上述等效关系进行了说明。

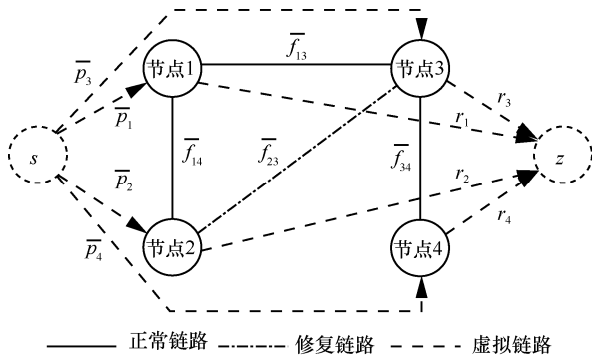


图 2 子问题等效最大流问题示例

在图 2 中，节点 2 与节点 3 之间的连接为迭代过程中所确定的需要修复的受损链路，即  $\{ij, ij \in E^r, E^r \subseteq E^0\}$ 。源节点  $s$  与目的节点  $z$  为新增

的虚拟节点，源节点  $s$  与 4 个边缘节点的虚拟链路容量为 4 个边缘节点的最大计算能力，4 个边缘节点与目的节点  $z$  的虚拟链路表示各边缘节点的本地数据计算量需求。图 2 中各边上的数值表示该链路的链路容量，从而最大化目的节点  $z$  的总到达流等价于最小化总数据的丢弃量。上述转化后的最大流问题，可以通过已有算法，如 Ford-Fulkerson 算法<sup>[28]</sup>，进行高效的多变量问题求解。

### 3.2 割平面生成与主问题构建

在 Benders 分解算法中，子问题的解将代入主问题用于生成主问题中的线性约束，即 Benders 割平面。由于本文考虑了完善资源<sup>[29]</sup>，即对于任意可行的主问题解，总有可行的子问题解，不需要再生成可行割，只需要构建最优割即可。为形成 Benders 最优割平面，需要利用对偶问题的互补松弛性原理<sup>[30]</sup>，根据上述子问题的解，提取限制条件式(3)所对应的对偶变量  $\mu_{ij}^t, ij \in E^0$ ，即可修复链路  $ij$  的边缘系统增量。

**定理 1** 在第  $t$  次迭代中的 Benders 最优割平面可以表示为

$$\eta \geq \theta^t + \sum_{ij \in E^0} \mu_{ij}^t \bar{f}_{ij} (e_{ij} - e_{ij}^t) \quad (9)$$

其中， $\eta$  为子问题目标函数的最优解的上界， $\theta^t$  为子问题在第  $t$  次迭代中所求得的目标函数的最优值，即第  $t$  次迭代中的最小数据丢弃总代价。

**证明** 见附录 1。

将上述 Benders 最优割平面代入主问题求解的限制条件中，可得主问题为

$$M: \min_{e, \eta} \phi = \sum_{ij \in E^0} c_{ij} e_{ij} + \eta$$

s.t. 式(4), 式(9)

其中， $\phi$  表示原问题 P 的最优值下界，这是因为主问题只考虑了部分约束情况，是原问题的松弛问题。

与子问题确定边缘网络的数据迁移、计算与丢弃不同，主问题负责确定受损链路集合  $E^0$  中需要被优先修复的链路集合  $E^r$ 。在主、子问题的循环迭代过程中，主问题的求解都是在给定数据迁移、计算与丢弃量的情况下进行的。主问题求解所得的  $E^r$  反过来被用于子问题的进一步求解，从而逐步逼近系统的最优数据迁移、计算、丢弃与链路修复策略。

### 3.3 基于 Benders 分解理论的迭代路径修复与计算迁移算法

将上述主问题初始化后代入子问题，并开始循环迭代寻找最优解。若在迭代过程中主问题的决策变量不能满足所有约束条件，则终止算法，原问题无解；否则持续迭代过程直至找到网络最优配置。上述过程的具体步骤如算法 1 所示。

**算法 1** 基于 Benders 分解理论的迭代路径修复与计算迁移算法

定义  $t=1$ ，给定网络参数  $\{c_{ij}, \bar{f}_{ij} | \forall ij \in E\}$ ， $\{r_i, \bar{p}_i, c_i | \forall i \in \mathcal{N}\}$ 。初始化算法上界  $\bar{\phi} = +\infty$  与算法下界  $\underline{\phi} = -\infty$ ，迭代精度参数  $\varepsilon$

1) 循环

2) 求解主问题 M，如果主问题无解，终止算法，原问题无解；否则得到  $e' = (e'_{ij}, \forall ij \in E^0)$  和算法下界  $\underline{\phi}$

3) 求解子问题 S，得到  $d' = (d'_i, \forall i \in \mathcal{N})$ ， $p' = (p'_i, \forall i \in \mathcal{N})$ ， $f' = (f'_{ij}, \forall ij \in E)$ ，对偶变量  $\mu' = (\mu'_{ij}, \forall ij \in E^0)$  以及最小目标函数值  $\theta'$

4) 更新算法上界  $\bar{\phi} = \min\{\bar{\phi}, \sum_{ij \in E^0} c_{ij} e'_{ij} + \theta'\}$

5)  $t = t + 1$

6) 直到算法收敛，收敛条件为  $(\bar{\phi} - \underline{\phi}) / \bar{\phi} < \varepsilon$

当算法收敛后，系统将根据当前迭代计算所得的  $e'$ ，修复其中值为 1 的链路。之后，利用子问题的解  $f'$  确定节点间的数据流迁移量，依据  $p'$  调整各节点的实际计算能力，并丢弃超出计算能力的的数据  $d'$ 。需要注意的是，主问题中  $\eta$  为一个连续变量， $e$  为离散变量，该问题仍是一个混合整数问题。尽管可以采用诸如遗传蚁群算法在内的算法进行求解，但由于其搜索域空间较大，算法复杂度依旧很高。此外，在每次迭代过程中，主问题中新的割平面的引入使算法下界  $\underline{\phi}$  保持了非减特性，但并没有类似机制保障算法上界  $\bar{\phi}$  的单调特性。这种非单调约束界特性将进一步加剧上述算法的计算时间开销。

## 4 Benders 分解加速算法设计

为解决算法 1 提出的基于 Benders 分解理论的迭代路径修复与计算迁移算法所存在的问题，本文在迭代求解过程中进一步引入了局部分支技术<sup>[31]</sup>。其主要目的在于在每次迭代过程中找到更好的问

题上界，以期实现上下界的内向夹逼，减少主问题的计算复杂性。

### 4.1 信赖域与汉明距离

如前文所述，基于割平面的 Benders 分解算法并不是一种稳定的算法，在迭代的早期，问题的解会在不同的可行域内大范围波动，从而导致较慢的收敛速度。在本文所考虑的工业物联网中的边缘计算场景下，大规模网络受损所引入的大量优化变量，特别是初始搜索空间为  $2^{E^0}$  的修复链路决策向量  $e$  的引入，将使该收敛速度问题进一步恶化。信赖域是解决上述大范围波动特性问题的一种极佳策略<sup>[32]</sup>。考虑到主问题中的  $e$  是一组 0-1 变量，可以采用汉明距离来限制两次迭代的距离。

假设  $(e', d', p', f')$  为原问题第  $t$  次迭代所得到的可行解，此时所有 0-1 优化变量中值为 1 的集合可以表示为  $E^t = \{e'_{ij} | e'_{ij} = 1, e'_{ij} \in E^0\}$ ，则第  $t+1$  次迭代与第  $t$  次迭代的汉明距离为

$$D(e^{t+1}, e^t) = \sum_{ij \in E^t} (1 - e^{t+1}_{ij}) + \sum_{ij \in E^0 \setminus E^t} e^{t+1}_{ij} \quad (10)$$

即第  $t+1$  次迭代中  $e$  相对第  $t$  次迭代中变化的二进制变量的数目。

为加快收敛速度，可以将解空间分解为 2 个相互独立的信赖域

$$D(e^{t+1}, e^t) \leq \Delta^{t+1} \quad (11)$$

$$D(e^{t+1}, e^t) \geq \Delta^{t+1} + 1 \quad (12)$$

其中， $\Delta^{t+1}$  表示在第  $t+1$  次迭代中信赖域的规模大小，其值的选取取决于主问题的复杂性及搜索范围的波动性需求。通过上述方式，原问题自然地分为 2 个子解空间，局部分支法便是基于此产生。在局部分支法中，式(11)和式(12)分别被称为左枝和右枝。

### 4.2 局部分支法

基于汉明距离，原问题 P 的解空间可根据 Benders 分解第  $t$  次迭代所得的可行解  $(e', d', p', f')$ ，分割为 2 个紧密相连的邻域空间。设  $e^k, k \in K^t$  为 Benders 分解第  $t$  次迭代过程中局部分支法迭代所计算的所有  $e$  的解（包括可行解与非可行解），其中可行解的集合表示为  $L^t$  ( $L^t \subseteq K^t$ )。根据汉明距离，原问题 P 可以分支为 2 个独立的左右枝问题。其中左枝问题为

$$P_k : \min_{e,d,f,p} \phi = \sum_{ij \in E^0} c_{ij} e_{ij} + \sum_{i \in \mathcal{N}} c_i d_i$$

s.t. 式(1)~式(7)

$$D(e_{ij}, e_{ij}^k) \geq 1, \forall ij \in E^0, k \in K^t \quad (13)$$

$$D(e_{ij}, e_{ij}^l) \geq \Delta^t + 1, \forall ij \in E^0, l \in L \quad (14)$$

$$D(e_{ij}, e_{ij}^m) \leq \Delta^t, \forall ij \in E^0 \quad (15)$$

其中，式(13)表示不重复比较之前已比较过的  $e$  值，式(14)表示当前左枝应该是上一级分支中的右分枝的分枝，式(15)表示左枝限制条件， $e_{ij}^m$  为当前获取的最优解。相应地可以得到右枝问题为

$$\bar{P}_k : \min_{e,d,f,p} \phi = \sum_{ij \in E^0} c_{ij} e_{ij} + \sum_{i \in \mathcal{N}} c_i d_i$$

s.t. 式(1)~式(7), 式(13)~式(14)

$$D(e_{ij}, e_{ij}^m) \geq \Delta^t + 1, \forall ij \in E^0 \quad (16)$$

其中，式(16)表示右枝限制。

令  $(e^{k+1}, d^{k+1}, p^{k+1}, f^{k+1})$  为左枝问题  $P_k$  的最优解，对应的目标函数值为  $\phi_{k+1}$ ，则有局部分支算法流程如算法2所示。

#### 算法2 局部分支算法

定义  $k=1$ ，令局部分支迭代的初始可行解  $(e^k, d^k, p^k, f^k) = (e^t, d^t, p^t, f^t)$ ，对应目标函数值为  $\phi_k$ 。给定  $\Delta^t$ ，最大允许计算时间  $T_{\max}$  以及迭代精度参数  $\bar{\varepsilon}$

- 1) 循环
- 2) 将当前分支分为左枝  $P_k$  与右枝  $\bar{P}_k$ ，计算  $\phi_{k+1}$  以及  $(e^{k+1}, d^{k+1}, p^{k+1}, f^{k+1})$
- 3) 如果  $\phi_{k+1} < \phi_k$  成立
- 4) 则更新  $L = L \cup \{k\}$ ， $K^t = K^t \cup \{k\}$ ，并跳转到右枝
- 5) 否则， $\phi_{k+1} \geq \phi_k$  或者  $P_k$  无解
- 6) 令  $\Delta^t = \Delta^t + 1$ ，向限制条件式(13)中添加  $D(e_{ij}, e_{ij}^k) \geq 1, \forall ij \in E^0$ ，并更新  $K^t = K^t \cup \{k\}$ 。重新计算左枝问题  $P_k$  的  $(e^{k+1}, d^{k+1}, p^{k+1}, f^{k+1})$  以及  $\phi_{k+1}$ ，返回步骤3)
- 7)  $k = k + 1$
- 8) 直到算法计算时间达到  $T_{\max}$  或满足精度要求  $(\phi_k - \phi_{k+1}) / \phi_k < \bar{\varepsilon}$

算法2循环条件中的  $T_{\max}$  是为了避免在计算过程中，由于汉明距离设置过大导致分枝问题无法解出的情况。在大规模受损网络下，修复链路决策  $e$  的

解空间规模十分庞大，选取较小的信赖域规模  $\Delta^t$  将更有利于提升左枝问题的计算速度。

在迭代过程中，可以获得原问题的严格上界  $\bar{\phi}_k = \min_{k \in K^t} \{\phi_k\}$ 。同时由于原问题  $P$  的主要难度在于下界的获取，通过每次迭代过程中所产生的一系列最优割平面，也可以加快搜索速度，从而使 Benders 分解在迭代过程中同时增强问题最优解的搜索上下界。

#### 4.3 基于局部分支法的 Benders 分解加速算法

将局部分支算法融入 Benders 分解得到的基于局部分支法加速的 Benders 分解算法如算法3所示。

**算法3** 基于局部分支法的 Benders 分解加速理论的迭代路径修复与计算迁移算法

定义  $t=1$ ，给定网络参数  $\{c_{ij}, \bar{f}_{ij} \mid \forall ij \in E\}$ ， $\{r_i, \bar{p}_i, c_i \mid \forall i \in \mathcal{N}\}$ 。初始化算法上界  $\bar{\phi} = +\infty$  与算法下界  $\underline{\phi} = -\infty$ ，迭代精度参数  $\varepsilon$

- 1) 循环
- 2) 求解主问题  $M$ ，如果问题无解，终止算法，原问题无解；否则得到  $e^t = (e_{ij}^t, \forall ij \in E^0)$  和算法下界  $\underline{\phi}$
- 3) 求解子问题  $S$ ，得到  $d^t = (d_i^t, \forall i \in \mathcal{N})$ ， $p^t = (p_i^t, \forall i \in \mathcal{N})$ ， $f^t = (f_{ij}^t, \forall ij \in E)$ ，对偶变量  $\mu^t = (\mu_{ij}^t, \forall ij \in E^0)$  以及最小目标函数值  $\theta^t$
- 4) 更新算法上界  $\bar{\phi} = \min\{\bar{\phi}, \sum_{ij \in E^0} c_{ij} e_{ij}^t + \theta^t\}$
- 5) 如果  $(\bar{\phi} - \underline{\phi}) / \bar{\phi} < \varepsilon$
- 6) 则最优解已找到，终止算法
- 7) 否则，生成新的 Benders 最优割平面
- 8) 运行算法2，获取一个增强上界  $\bar{\phi}_k$ ，以及不同可行解产生的一系列 Benders 最优割平面
- 9)  $t = t + 1$
- 10) 直到算法收敛

与算法1类似，当算法3收敛后，系统将根据当前迭代计算所得的  $e^t$  确定修复链路集合。然后，根据计算所得  $f^t$ 、 $p^t$  及  $d^t$  的值，确定网络中数据的迁移、计算与丢弃量。不同于算法1，算法3通过局部分支技术，保证了原问题  $P$  的上界在迭代过程中的严格递减。对于大规模受损网络，这意味着每次迭代过程中，主问题  $M$  所需优化的链路修复决策向量  $e^t$  的搜索空间逐步减少，解的搜索速度随着迭代次数的增加而逐渐加快。

上述算法每次迭代获得的新割平面中，并不是所有的割平面都需要加入主问题求解过程，可以只添加最深的割平面（即使可行域最大程度减少的割平面）。考虑到算法的收敛性要求与 Benders 分解初始阶段的糟糕表现，可以仅在初始迭代阶段添加信赖域限制。当迭代趋于稳定后，解除该限制条件。

值得注意的是，在每次分支过程中，分支问题  $P_1, P_2, \dots$  都拥有与原问题  $P$  相同的结构。因此针对每个分支问题，可以单独使用算法 1 进行求解。之前的分支问题求得的割平面也可以继续用于后续分支问题的求解。

### 5 仿真分析

本节对所提基于 Benders 分解理论的迭代路径修复与计算迁移算法（下文简称为 Benders 分解算法）以及基于局部分支法的 Benders 分解加速理论的迭代路径修复与计算迁移算法（下文简称为 Benders 分解加速算法）性能进行了仿真测试，并验证分析了所提算法相较于其他基准算法的性能优势。

#### 5.1 仿真参数与对比算法

本文考虑一个包含  $N = 50$  个节点的边缘计算网络，网络修复初期各节点的最大计算能力  $\bar{p}_i$  相互独立且均匀分布于  $[10, 25]$  Gbit。考虑到计算能力与计算需求的匹配性，网络修复初期节点的计算需求  $r_i$  同样服从  $[10, 25]$  Gbit 上的均匀分布。不失一般性地，假设边缘网络原始拓扑中有线链路数量  $|E| = 2N$ ，网络拓扑同随机网络<sup>[4]</sup>，且节点间数据传输处处可达。考虑到计算节点的处理能力波动范围，链路容量  $\bar{f}_{ij}$  服从  $[5, 15]$  Gbit 上的均匀分布。为模拟网络大规模受损情况，在下文中，如无特殊说明，链路受损比例  $q = 75\%$ 。假设修复每条受损链路的成本  $c_{ij}$  均匀分布于  $[1 \times 10^4, 2 \times 10^4]$  元，因计算能力不足而丢弃数据的代价  $c_i$  为  $10^4$  元/Gbit。

为了验证本文所提受损网络修复机制的效果，将本文算法与以下基准算法进行比较。

1) 随机修复算法。随机选择网络中受损的链路进行修复，不考虑网络的具体拓扑结构与网络计算流迁移的动态特征。

2) 最大连通图修复算法<sup>[16]</sup>。修复网络的最大连通图中的所有受损链路，确保网络中最大连通图

中的所有节点能达到强连通状态。

3) 介数中心性排序修复算法<sup>[11]</sup>。分析边缘网络受损前的拓扑结构，对各条有线链路的介数中心性进行排序，优先修复介数中心性大的受损链路，修复链路数量与所提算法相同。

#### 5.2 算法收敛性

图 3 为所提 Benders 分解算法与 Benders 分解加速算法的收敛性对比。从图 3 可以看出，2 种算法可以在有限次数内实现快速迭代，具有较好的收敛性。需要注意的是，图 3 中迭代次数表征的是算法 1 与算法 3 中主问题与子问题的总循环迭代次数。在每次迭代过程中，子问题的计算复杂度为  $O(|N||E|^2)$ ，主问题计算复杂度为  $O(n|E|^3)$ ，其中  $n$  为主问题求解的内循环迭代次数。尽管 Benders 分解加速算法相对 Benders 分解算法，总迭代次数只减少一次，但其实际计算时间复杂度降低  $O(|N||E|^2 + n|E|^3)$ 。在大规模受损网络环境下，该计算时间复杂度的改善仍十分可观。从图 3 中相同迭代次数情况下算法上下界的对比可以看出，相对于 Benders 分解算法，局部分支法辅助的 Benders 分解加速算法由于有更深割平面的引入，收敛速度得到显著提升，能够更好地适应大规模网络的应用需求。此外，Benders 分解算法与 Benders 分解加速算法在收敛后系统总开销趋于一致。因此，在下文中，为便于与基准算法进行对比，Benders 分解算法和 Benders 分解加速算法统称为“本文算法”。

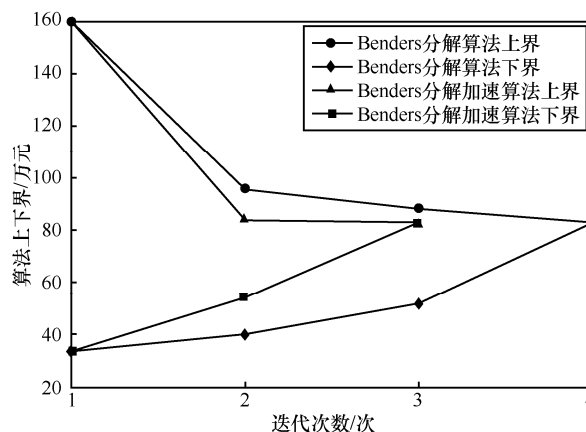


图 3 所提 Benders 分解算法与 Benders 分解加速算法收敛性对比

#### 5.3 算法性能

图 4 和图 5 分别为不同网络受损程度  $q$  以及不同网络规模  $N$  情况下，本文算法、随机修复算法、最大连通图修复算法、介数中心性排序修复算法的

系统总开销曲线。从图 4 和图 5 可以看出，在不同网络受损程度以及网络规模情况下，由于考虑了网络中节点的计算需求与实际计算能力、链路容量限制等网络动态特征，本文算法性能均优于基准算法。值得注意的是，最大连通图修复算法的系统总开销性能是所有算法中最差的，甚至弱于随机修复算法，这与现实网络中网络修复呈现的状态十分吻合。在现实网络中，修复过程总是先使网络中形成大大小小各自独立的簇，在修复的最后才将各个簇连接起来形成最大连通图<sup>[13]</sup>。

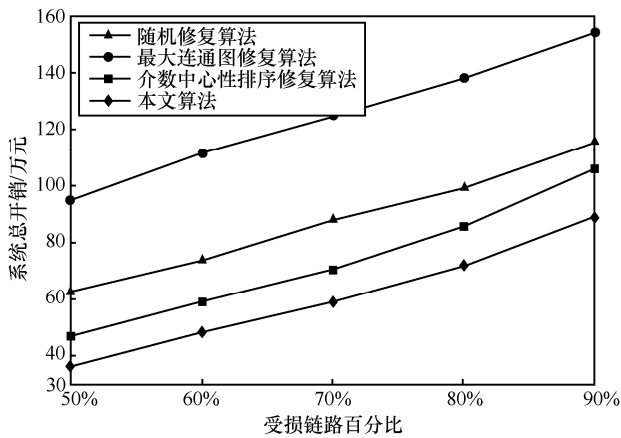


图 4 不同网络受损程度下算法性能对比

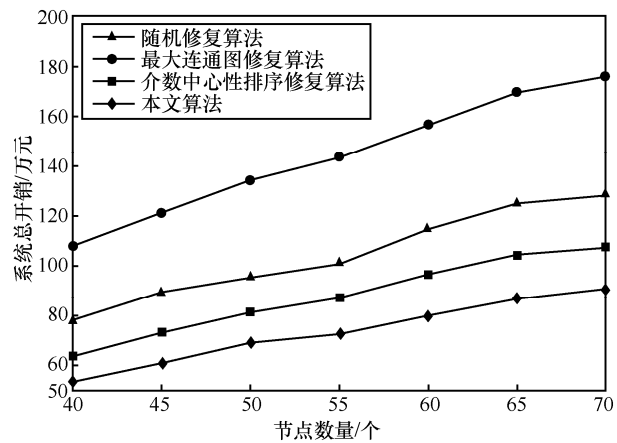


图 5 不同网络规模下算法性能对比

### 5.4 算法多场景适用性

图 6 和图 7 分别表示不同单位数据丢弃代价场景下本文算法修复的受损链路数目与计算数据丢弃量。考虑到本文算法系统总开销对于链路修复成本和计算数据丢弃的双重考量，随着单位数据丢弃代价  $c_i$  的增大，修复链路数目与数据丢弃量呈现出相反的增减关系。与经验相符，随着  $c_i$  增大，即数据丢弃成本增大，网络倾向于修复更多的链路使计

算需求得以满足。当  $c_i$  取值在  $[1, 1.5]$  万元/Gbit 时，图 7 中数据丢弃量开始出现显著下降，这是因为在该范围内，系统刚好处于敏感临界点附近。在该值附近，系统均衡修复成本与丢弃数据开销的能力最强。任意微小的  $c_i$  值变化都可能带来均衡决策的较大幅度改变。在  $c_i$  超过 2 万元/Gbit 后，继续修复受损链路带来的边际效益降低，网络修复链路数目与计算数据丢弃量保持稳定。可以看出，本文算法能够很好地适用于各种不同数据丢弃代价场景，具有较好的可扩展性与适应性。

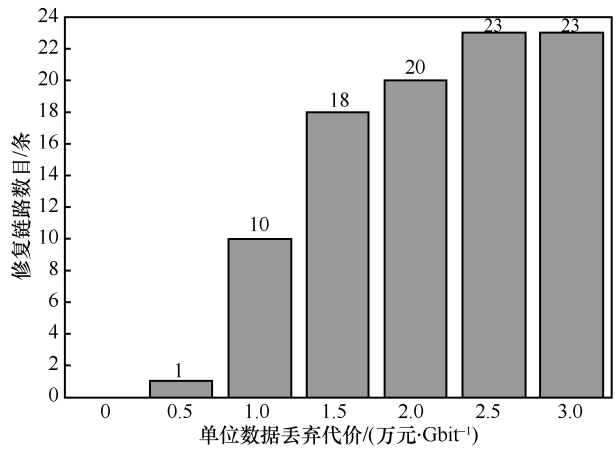


图 6 不同单位数据丢弃代价下本文算法受损链路修复数目

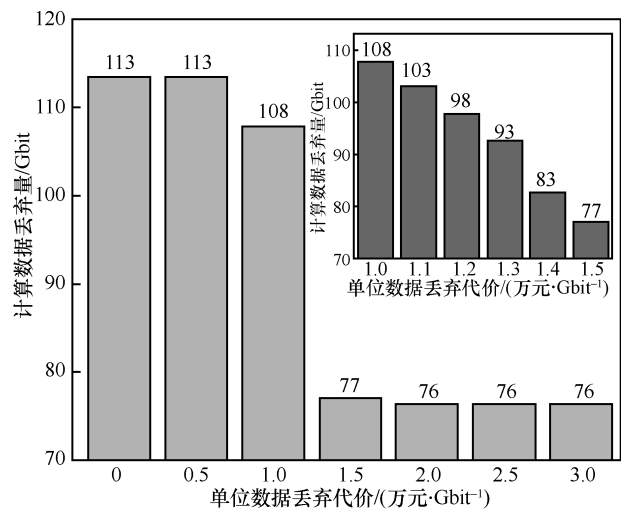


图 7 不同单位数据丢弃代价下本文算法计算数据丢弃量

为进一步分析本文算法在不同网络拓扑场景下的性能，本文将网络形式由单一的随机网络场景扩展至栅格网络与无标度网络<sup>[33]</sup>。与随机网络不同，栅格网络各节点的度完全相同，无标度网络节点的度服从幂律分布。从图 8 可以看出，在 3 类网络下，各算法性能均会产生不同程度的波动，但本

文算法的系统总开销性能均优于基准算法。得益于对于网络拓扑与动态特征的联合分析，本文算法具有很好的多场景适应性。

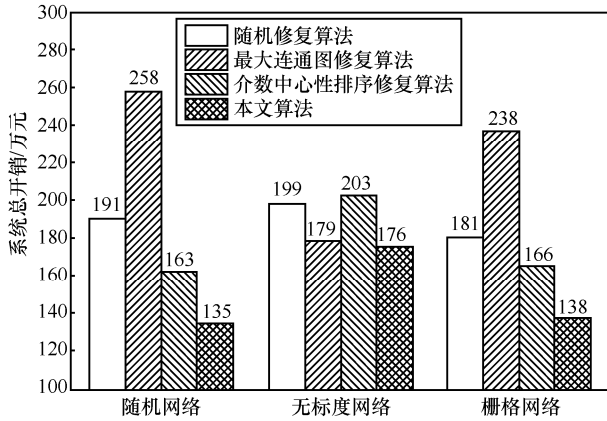


图 8 不同网络拓扑下算法对比情况 (N=100)

## 6 结束语

本文针对工业物联网中边缘计算网络的易受损特征，提出了一种网络大规模受损后的修复机制，给出了优先修复链路集合决策与计算迁移配置的联合优化方法，有效缓解了网络修复初期有限修复资源与大量数据计算需求的矛盾。考虑到原始问题的求解困难，利用 Benders 分解算法将其转化为更易求解的主问题与子问题，通过割平面的不断更新，逐步逼近问题最优解。进一步地，结合局部分支法，设计了一种 Benders 分解加速算法，有效提升了算法收敛速度。仿真结果表明，相对现有基于拓扑结构的修复算法，本文算法具有较好的系统修复性能。

尽管出于数据公平性考量，本文考虑计算数据具有相同处理优先级，即不同节点丢弃每单位数据的代价相同，本文思想仍可以十分便捷地扩展至每单位数据丢弃代价不同的场景，只需将子问题解法替换为现有最小费用流算法（如 Dinic 算法）即可。

本文算法主要针对节点间为有线链路连接的情况。对于存在动态无线链路的场景，例如无人机充当移动边缘节点的场景，信道容量随无人机位置改变而改变，需要进一步联合考量无人机运行轨迹、无线频谱分配等问题。这将使本已复杂的网络修复问题变得更加难以求解，已超出本文的考虑范围，留待后续研究。

## 附录 1 Benders 最优割平面证明

当给定  $e'_{ij}, ij \in E^0$  时，由原问题 P 可得

$$\theta(e'_{ij}) = \min_{d, f, p} \{ \sum_{i \in \mathcal{N}} c_i d_i \mid (1)-(3), (5)-(7) \} \quad (17)$$

上述问题的等效对偶问题的目标函数可以写为

$$\theta(e'_{ij}) = \max_{\alpha, \beta, \lambda, \mu} \{ \sum_{i \in \mathcal{N}} (\alpha_i r_i + \beta_i \bar{p}_i) + \sum_{ij \in E^1} \lambda_{ij} \bar{f}_{ij} + \sum_{ij \in E^0} \mu_{ij} \bar{f}_{ij} e'_{ij} \} \quad (18)$$

其中， $\alpha = (\alpha_i, \forall i \in \mathcal{N})$ ， $\beta = (\beta_i, \forall i \in \mathcal{N})$ ， $\lambda = (\lambda_{ij}, \forall ij \in E^1)$ ， $\mu = (\mu_{ij}, \forall ij \in E^0)$  为四组对偶问题优化变量（对偶变量）。设第  $t$  次迭代过程中，假设上述对偶变量组解为  $\alpha^t = (\alpha_i^t, \forall i \in \mathcal{N})$ ， $\beta^t = (\beta_i^t, \forall i \in \mathcal{N})$ ， $\lambda^t = (\lambda_{ij}^t, \forall ij \in E^1)$  以及  $\mu^t = (\mu_{ij}^t, \forall ij \in E^0)$ ，问题最优解  $\theta^t = \theta(e'_{ij})$ ，则有

$$\theta(e_{ij}) \geq \sum_{i \in \mathcal{N}} (\alpha_i^t r_i + \beta_i^t \bar{p}_i) + \sum_{ij \in E^1} \lambda_{ij}^t \bar{f}_{ij} + \sum_{ij \in E^0} \mu_{ij}^t \bar{f}_{ij} e_{ij} \quad (19)$$

在  $e'_{ij}, ij \in E^0$  附近进行线性化操作可得

$$\begin{aligned} \eta \geq \theta(e_{ij}) &\geq \sum_{i \in \mathcal{N}} (\alpha_i^t r_i + \beta_i^t \bar{p}_i) + \sum_{ij \in E^1} \lambda_{ij}^t \bar{f}_{ij} + \\ &\sum_{ij \in E^0} \mu_{ij}^t \bar{f}_{ij} e'_{ij} + \sum_{ij \in E^0} \mu_{ij}^t \bar{f}_{ij} (e_{ij} - e'_{ij}) = \\ &\theta^t + \sum_{ij \in E^0} \mu_{ij}^t \bar{f}_{ij} (e_{ij} - e'_{ij}) \end{aligned} \quad (20)$$

证毕。

## 参考文献:

- [1] WU H, TIAN H, NIE G F, et al. Wireless powered mobile edge computing for industrial Internet of Things systems[J]. IEEE Access, 2020, 8: 101539-101549.
- [2] XIE G Q, YANG K H, XU C, et al. Digital twinning based adaptive development environment for automotive cyber-physical systems[J]. IEEE Transactions on Industrial Informatics, 2021, PP(99): 1.
- [3] WU H, LYU X C, TIAN H. Online optimization of wireless powered mobile-edge computing for heterogeneous industrial Internet of things[J]. IEEE Internet of Things Journal, 2019, 6(6): 9880-9892.
- [4] XING L D. Cascading failures in Internet of things: review and perspectives on reliability and resilience[J]. IEEE Internet of Things Journal, 2021, 8(1): 44-64.
- [5] PROKHORENKO V, ALI BABAR M. Architectural resilience in cloud, fog and edge systems: a survey[J]. IEEE Access, 2020, 8: 28078-28095.
- [6] AYOUBI S, ASSI C, CHEN Y H, et al. Restoration methods for cloud multicast virtual networks[J]. Journal of Network and Computer Applications, 2017, 78: 180-190.
- [7] SATRIA D, PARK D, JO M. Recovery for overloaded mobile edge computing[J]. Future Generation Computer Systems, 2017, 70: 138-147.
- [8] TENG R, LI H B, MIURA R. Dynamic recovery of wireless multi-hop infrastructure with the autonomous mobile base station[J]. IEEE Access, 2016, 4: 627-638.
- [9] LI P D, YANG X F. On dynamic recovery of cloud storage system under advanced persistent threats[J]. IEEE Access, 2019, 7: 103556-103569.
- [10] BAXTER G J, TIMÁR G, MENDES J F F. Targeted damage to inter-

- dependent networks[J]. Physical Review E, 2018, 98(3): 032307.
- [11] BRUMMITT C D, D'SOUZA R M, LEICHT E A. Suppressing cascades of load in interdependent networks[J]. Proceedings of the National Academy of Sciences, 2012, 109(12): 680-689.
- [12] MORONE F, MAKSE H A. Influence maximization in complex networks through optimal percolation[J]. Nature, 2015, 524(7563): 65-68.
- [13] RUDNICK H, MOCARQUER S, ANDRADE E, et al. Disaster management[J]. IEEE Power and Energy Magazine, 2011, 9(2): 37-45.
- [14] PUNZO G, TEWARI A, BUTANS E, et al. Engineering resilient complex systems: the necessary shift toward complexity science[J]. IEEE Systems Journal, 2020, 14(3): 3865-3874.
- [15] WU H, TIAN H, FAN S S, et al. Data age aware scheduling for wireless powered mobile-edge computing in industrial Internet of things[J]. IEEE Transactions on Industrial Informatics, 2021, 17(1): 398-408.
- [16] SMITH A M, PÓSFAI M, ROHDEN M, et al. Competitive percolation strategies for network recovery[J]. Scientific Reports, 2019, 9: 11843.
- [17] QIN J, MIAO L X. Combined simulated annealing algorithm for logistics network design problem[C]//2009 International Workshop on Intelligent Systems and Applications. Piscataway: IEEE Press, 2009: 1-4.
- [18] KHELIFI M, SAIDI M Y, BOUDJIT S. Genetic algorithm based model for capacitated network design problem[C]//2016 24th International Conference on Software, Telecommunications and Computer Networks. Piscataway: IEEE Press, 2016: 1-6.
- [19] LIAN H B. Network design problems, formulations and solutions[D]. Richardson: The University of Texas at Dallas, 2012.
- [20] LI D Q, ZHANG Q, ZIO E, et al. Network reliability analysis based on percolation theory[J]. Reliability Engineering & System Safety, 2015, 142: 556-562.
- [21] LYU X C, REN C S, NI W, et al. Distributed optimization of collaborative regions in large-scale inhomogeneous fog computing[J]. IEEE Journal on Selected Areas in Communications, 2018, 36(3): 574-586.
- [22] ZHAO P T, TIAN H, QIN C, et al. Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing[J]. IEEE Access, 2017, 5: 11255-11268.
- [23] ZHAO P T, TIAN H, CHEN K C, et al. Context-aware TDD configuration and resource allocation for mobile edge computing[J]. IEEE Transactions on Communications, 2020, 68(2): 1118-1131.
- [24] CODATO G, FISCHETTI M. Combinatorial Benders' cuts for mixed-integer linear programming[J]. Operations Research, 2006, 54(4): 756-766.
- [25] RAHMANIANI R, CRAINIC T G, GENDREAU M, et al. The Benders decomposition algorithm: a literature review[J]. European Journal of Operational Research, 2017, 259(3): 801-817.
- [26] WU H, TIAN H, NIE G F. Energy-efficient inter-frequency small cell discovery in dense urban environments[J]. IEEE Wireless Communications Letters, 2019, 8(1): 41-44.
- [27] F PEREIRA M V, V G PINTO L M, F CUNHA S H, et al. A decomposition approach to automated generation/transmission expansion planning[J]. IEEE Transactions on Power Apparatus and Systems, 1985, 104(11): 3074-3083.
- [28] ERICKSON J. Algorithms[M]. Illinois: Independently Published, 2019.
- [29] OLIVEIRA F, GROSSMANN I E, HAMACHER S. Accelerating Benders stochastic decomposition for the optimization under uncertainty of the petroleum product supply chain[J]. Computers & Operations Research, 2014, 49: 47-58.
- [30] 运筹学教材编写组. 运筹学: 本科版[M]. 第4版. 北京: 清华大学出版社, 2013.
- Textbook Compilation Team. Operations research[M]. Beijing: Tsinghua University Press, 2013.
- [31] FISCHETTI M, LODI A. Local branching[J]. Mathematical Programming, 2003, 98(1/2/3): 23-47.
- [32] SANTOSO T, AHMED S, GOETSCHALCKX M, et al. A stochastic programming approach for supply chain network design under uncertainty[J]. European Journal of Operational Research, 2005, 167(1): 96-115.
- [33] YU A Q, WANG N, WU N. Scale-free networks: characteristics of the time-variant robustness and vulnerability[J]. IEEE Systems Journal, 2020, PP(99): 1-11.

## [作者简介]



田辉(1963-),女,河南郑州人,博士,北京邮电大学教授、博士生导师,主要研究方向为无线资源管理、智能边缘计算、移动社交网络。



伍浩(1994-),男,江西新余人,北京邮电大学博士生,主要研究方向为工业物联网、无线资源管理、网络弹性、移动边缘计算、复杂网络。

田洋(1991-),男,河南郑州人,北京邮电大学博士生,主要研究方向为大数据环境下的链路预测、复杂网络上的传播动力学等。

任建阳(1997-),男,北京人,北京邮电大学硕士生,主要研究方向为边缘计算、无线资源管理等。

崔亚娟(1991-),女,河南平顶山人,北京邮电大学博士生,主要研究方向为无人机网络、车联网、移动自组织网络等。

艾文宝(1962-),男,江西高安人,博士,北京邮电大学教授、博士生导师,主要研究方向为最优化理论和算法及其应用。

袁健华(1979-),女,湖南郴州人,博士,北京邮电大学教授、博士生导师,主要研究方向为最优化理论和算法、有限元方法及其工程应用。