计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.201100205

基于移动边缘计算的区块链计算资源分配和收益分享研究

徐 旭[1] 钱丽萍[1] 吴 远[2]
1 浙江工业大学信息工程学院 杭州 310023
2 澳门大学智慧城市物联网国家重点实验室 澳门 氹仔 999078
(xxu_zjut@163.com)

摘 要 针对移动终端设备本地计算资源有限的现状,提出了一种结合移动边缘计算机制的区块链系统。通过综合考虑系统中移动终端设备和边缘服务器的计算资源分配,以及移动终端设备的收益分配,提出了一个联合优化问题来最大化移动终端设备和边缘服务器的系统效用。为了快速求解该联合优化问题,设计了一种基于循环块坐标下降思想的多层分解算法。首先给定收益分享变量的值,通过对相应的子问题进行求解,得到移动终端设备以及边缘服务器的计算资源分配结果。然后把得到的结果作为固定的值继续求解移动终端设备的收益分享问题。最后,交替优化两部分变量直到算法收敛。仿真结果显示,所提算法能快速得到联合优化问题的最优解并有效提升区块链系统的系统效用。

关键词:区块链;工作量证明;移动边缘计算;收益分享;计算资源分配
中图法分类号 TP391

Computation Resource Allocation and Revenue Sharing Based on Mobile Edge Computing for Blockchain

XU Xu[1],QIAN Li-ping[1] and WU Yuan[2]
1 College of Information Engineering,Zhejiang University of Technology,Hangzhou 310023,China
2 State Key Laboratory of Internet of Things for Smart City,University of Macau,Ilha da Taipa,Macau 999078,China

Abstract This paper proposes a mobile edge computing (MEC) assisted blockchain system in which mobile terminals (MT) do not have enough local computation resources to solve the proof of work (PoW) puzzle. By combining the computation resource allocation of MTs and edge server (ES) with the revenue sharing of MTs,a joint optimization problem is formulated to maximize the system-wide utility of all MTs and the ES. To solve the optimization problem efficiently,a multi-layer decomposition algorithm based on cyclic block coordinate descent (CBCD) is proposed. First,given the revenue sharing variables in advance,the corresponding sub-problem can be solved to obtain the computation resource allocation results of both MTs and ES. Then,with the obtained computation resource allocation,the revenue sharing variables of MTs are optimized. Finally,this paper optimizes the two sub-problems alternately until the algorithm reaches convergence. The numerical results show that the proposed algorithm can obtain the optimal solution of the joint optimization problem effectively and improve the system-wide utility.

Keywords Blockchain,Proof of work,Mobile edge computing,Revenue sharing,Computation resource allocation

1 引言

近年来,区块链技术的迅猛发展引起了学术界和工业界的广泛关注。区块链是一种新兴的分布式账本系统,其内部每个节点都存储着相同的信息记录,通过去中心化、共识机制等方式保障数据的存储安全[1-3]。因为区块链具有开放性、透明性、信息可溯源、匿名性等优势,所以目前已有很多研究将物联网与区块链系统相结合,来解决数据的安全存储以及传输问题。文献[4]将区块链技术应用在当前的车联网环境中,并设计了一个消息认证方案,该方案能较好地解决安全以及隐私问题。文献[5]基于区块链技术实现了物联网系统中的数据共享以及分布式存储。文献[6]利用区块链以及密码学技术设计了一种安全可靠的物联网存储系统。文献[7]结合区块链以及智能合约技术实现了车辆边缘网络中数据的安全共享。文献[8]提出了一种用户身份认证方案,利用区块链技术有效保障了网络通信以及信息安全。

在大部分与物联网相结合的区块链系统中,移动终端设备(Mobile Terminal,MT)需要作为节点执行工作量证明(Proof of Work,PoW)任务来达成共识,以保证系统内部所有节点存储的数据一致。但是移动终端设备的本地计算资源十分有限,难以提供足够的计算资源来执行工作量证明任务[9]。得益于移动边缘计算(Mobile Edge Computing,MEC)技术的

到稿日期:2020-11-30 返修日期:2021-03-13 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金(62072490)
This work was supported by the National Natural Science Foundation of China(62072490).
通信作者:钱丽萍(lpqian@zjut.edu.cn)

普及[10-11]，移动终端设备可以向边缘服务器（Edge Server，ES）发起调度计算资源的请求，额外获取边缘服务器上的计算资源。这一过程能有效增强移动终端设备的计算能力。

因此，针对在区块链系统中利用移动边缘计算机制来提高移动终端设备的计算能力，学者们进行了大量的研究。文献[12]研究了在区块链赋能的边缘计算系统中资源合理调度的问题。文献[13]介绍了移动区块链的边缘计算概念和计算资源管理方法。文献[14]提出了一种基于拍卖的优化方法来最大化移动区块链的系统收益。文献[15]提出了一种基于联合博弈的区块链网络计算资源分配方案。文献[16]针对数据处理任务和挖掘任务，在区块链系统中提出了一种多跳协同计算卸载算法。文献[17]基于区块链去中心化的特点提出了一种边缘计算资源分配方式。文献[18]研究了移动区块链中边缘服务器的计算资源分配以及系统效用最大化问题。

但是上述研究均未考虑区块链系统中移动终端设备本地计算资源的边际成本，以及在移动终端设备和边缘服务器共同受益于边缘计算的过程中，移动终端设备如何合理分配自身的计算资源执行工作量证明任务。为此，基于收益分享机制[19-20]，本文提出了一个关于计算资源分配以及收益分享的联合优化问题。在该问题中，每个移动终端设备根据其本地计算资源的边际成本，合理分配本地计算资源以及边缘服务器的计算资源来完成工作量证明任务。在任务完成后，每个移动终端设备分享部分收益至边缘服务器作为对边缘服务器的补偿。为了快速求解该联合优化问题，本文提出了一种基于循环块坐标下降（Cyclic Block Coordinate Descent，CBCD）思想[21]的多层分解算法。最后通过大量的数值结果来验证本文提出算法的有效性和准确性。与 LINGO 软件[22]求解的结果相比，本文提出的算法能快速解决联合优化问题。同时，与其他计算资源分配机制相比，本文算法能有效提升系统效用。

2　系统框架与模型建立

2.1　系统框架

本文考虑在区块链系统中存在着一组移动终端设备（用集合 $\mathcal{I}=\{1,2,3,\cdots,I\}$ 表示）。每个移动终端设备竞争地执行工作量证明任务以获得相应的收益。为了提高自身的计算能力，移动终端设备从边缘服务器获取额外的计算资源，并分享其获得的部分收益给边缘服务器作为补偿。具体来说，对于每一个移动终端设备 i，用 $\boldsymbol{\gamma}_i$ 来表示移动终端设备 i 的收益分享变量。每个移动终端设备保留 $\boldsymbol{\gamma}_i$ 部分的收益，并将 $(1-\boldsymbol{\gamma}_i)$ 部分的收益分享给边缘服务器。图 1 给出了本文构建的系统模型。

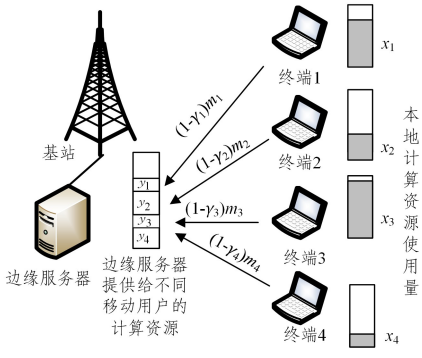

图 1　系统模型图

Fig. 1　System model

2.2　问题建模

在本文构建的系统模型中，每个移动终端设备都可以通过获取边缘服务器的计算资源来提高其计算能力。因此每个移动终端设备的计算资源总量由两部分组成。一部分是移动终端设备用于执行工作量证明任务的本地计算资源 $\boldsymbol{x}_i$。与文献[14-18]中本地计算资源固定为常量不同，本文考虑每个移动用户会根据本地计算资源的边际成本合理分配其用于执行工作量证明任务的本地计算资源。另一部分是移动终端设备从边缘服务器获取的计算资源 $\boldsymbol{y}_i$。$\boldsymbol{\alpha}_i$ 表示每个移动终端设备当前计算资源占整个系统计算资源的比重：

$$\boldsymbol{\alpha}_i=\frac{\boldsymbol{x}_i+\boldsymbol{y}_i}{\sum\limits_{i\in\mathcal{I}}\boldsymbol{x}_i+\sum\limits_{i\in\mathcal{I}}\boldsymbol{y}_i} \tag{1}$$

根据文献[10,14-15]，在执行工作量证明任务的过程中，每个移动终端设备需要先计算出工作量证明问题的结果（本文用服从泊松分布的参数 θ 表示单位时间内该事件发生的平均次数），之后移动终端设备需要将该结果通过广播传递给其他移动终端设备并达成共识。但是，由于网络延迟等的存在，首个求解出工作量证明问题的移动终端设备有一定的概率未能完成整个共识过程并成功将新生成的块添加到区块链中。这个概率用 $p_{\text{orphan}}(t_i)$ 来表示：

$$p_{\text{orphan}}(t_i)=1-\mathrm{e}^{-\theta\tau(t_i)} \tag{2}$$

其中，参数 t_i 表示移动终端设备 i 生成的块的大小。根据文献[14-15]，$\tau(t_i)$ 表示生成的块在区块链网络中传播的时间，$\tau(t_i)=z\times t_i$，z 是一个给定的延迟因子。

结合式(2)以及文献[10,14-15]，每个设备成功完成工作量证明任务并获得相应收益的概率可以表示为：

$$P_i(\boldsymbol{\alpha}_i,t_i)=\boldsymbol{\alpha}_i(1-p_{\text{orphan}}(t_i)) \tag{3}$$

根据以上定义，每个移动终端设备在执行工作量证明任务的过程中得到的收益可表示为：

$$\boldsymbol{m}_i=(R+rt_i)P_i(\boldsymbol{\alpha}_i,t_i)=(R+rt_i)\frac{\boldsymbol{x}_i+\boldsymbol{y}_i}{\sum\limits_{i\in\mathcal{I}}\boldsymbol{x}_i+\sum\limits_{i\in\mathcal{I}}\boldsymbol{y}_i}\mathrm{e}^{-\theta zt_i} \tag{4}$$

其中，参数 R 为区块链系统收益的固定项，r 表示给定的可变收益因子。

每个移动终端设备获得收益之后，会分享 $(1-\boldsymbol{\gamma}_i)$ 部分的收益给边缘服务器。根据式(4)，边缘服务器的最终收益可表示为：

$$\boldsymbol{m}_{\text{ES}}=\sum_{i\in\mathcal{I}}(1-\gamma_i)\boldsymbol{m}_i-C(\sum_{i\in\mathcal{I}}\boldsymbol{y}_i) \tag{5}$$

其中，函数 $C(\sum\limits_{i\in\mathcal{I}}\boldsymbol{y}_i)$ 是边缘服务器提供计算资源 $\sum\limits_{i\in\mathcal{I}}\boldsymbol{y}_i$ 给相应的移动终端设备而产生的成本，其表达式为 $C(\sum\limits_{i\in\mathcal{I}}\boldsymbol{y}_i)=p\sum\limits_{i\in\mathcal{I}}\boldsymbol{y}_i$，参数 p 表示边缘服务器上计算资源的边际成本。

基于上述建模，本文提出了一个联合优化问题来最大化所有移动终端设备和边缘服务器的系统效用：

$$\text{P1}:\max\sum_{i\in\mathcal{I}}\ln(\boldsymbol{\gamma}_i\boldsymbol{m}_i-\pi_i\boldsymbol{x}_i)+\ln(\boldsymbol{m}_{\text{ES}})$$

$$\text{s.t. }(4),(5)$$

$$0\leqslant\boldsymbol{x}_i\leqslant C_i^{\max},\ \forall\, i\in\mathcal{I} \tag{6}$$

$$0\leqslant\boldsymbol{y}_i,\ \forall\, i\in\mathcal{I} \tag{7}$$

$$0\leqslant\sum_{i\in\mathcal{I}}\boldsymbol{y}_i\leqslant C^{\text{tot}} \tag{8}$$

$$0\leqslant\boldsymbol{\gamma}_i\leqslant 1,\ \forall\, i\in\mathcal{I} \tag{9}$$

变量：$\{\boldsymbol{x}_i\}_{i\in\mathcal{I}},\{\boldsymbol{y}_i\}_{i\in\mathcal{I}},\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$

基于比例公平性的网络效益优化一直是网络优化中颇受

关注的研究方向[23-24]，考虑到移动终端设备和边缘服务器都希望从计算资源分配协作的过程中获益，本文在问题 P1 的目标函数中采用基于比例公平性的效用函数[23-24]来建模移动终端设备和边缘服务器各自的效用，从而确保移动终端设备和边缘服务器能从计算资源分配协助中共同获益。其中，π_i 表示移动终端设备使用本地计算资源的边际成本；约束条件(6)表示每个移动终端设备本地计算资源的上限 $C_i^{\max}$；约束条件(8)表示所有移动终端设备从边缘服务器得到的计算资源不能超过边缘服务器的计算资源上限 C^{tot}。虽然问题 P1 形式简单，但其目标函数难以直接求解。本文利用了问题 P1 的隐藏凸性(当部分变量是给定的)进行多层分解并求解，具体细节将在下一节中说明。

3 问题求解及优化算法设计

在本节中，问题 P1 通过给定部分变量的方式被解耦为两个子问题。但是，在给定收益分享变量后，得到的子问题仍是一个难以直接求解的问题。通过继续给定部分变量的方式，将该子问题再解耦为两个内部问题，并提出相应的算法对内部问题进行求解。在该子问题内部达到收敛后，执行外部迭代算法。整个算法持续迭代直到问题 P1 收敛。具体的算法分解形式如图 2 所示。

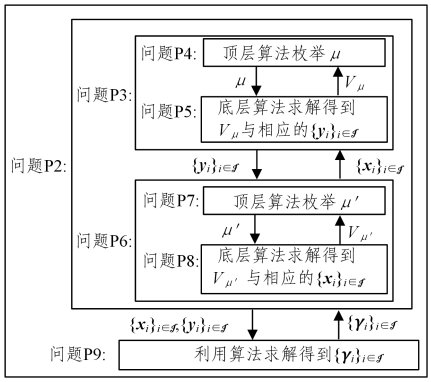

图 2 算法结构图

Fig. 2 Structure diagram of the proposed algorithm

3.1 给定收益分享变量 $\{\gamma_i\}_{i\in\mathcal{I}}$ 时的子问题 P2

首先，当给定收益分享变量 $\{\gamma_i\}_{i\in\mathcal{I}}$ 时，原问题 P1 可以转化为以下关于计算资源优化的问题：

$$\text{P2}:\max \sum_{i\in\mathcal{I}}\ln(\gamma_i m_i-\pi_i x_i)+\ln\Big(\sum_{i\in\mathcal{I}}(1-\gamma_i)m_i-C\Big(\sum_{i\in\mathcal{I}}y_i\Big)\Big)$$

s. t. (4), (5), (6), (7), (8)

变量：$\{x_i\}_{i\in\mathcal{I}}$，$\{y_i\}_{i\in\mathcal{I}}$

在给定收益分享变量 $\{\gamma_i\}_{i\in\mathcal{I}}$ 得到问题 P2 后，其依旧是一个难以直接求解的问题。对此，继续固定其中部分变量，将问题解耦为两个内部问题分别进行求解。以下是对问题 P2 分解的具体步骤。

3.1.1 给定变量 $\{x_i\}_{i\in\mathcal{I}}$ 时的内部问题 P3

当给定移动终端设备本地计算资源变量 $\{x_i\}_{i\in\mathcal{I}}$ 时，问题 P2 可以转化为：

$$\text{P3}:\max \sum_{i\in\mathcal{I}}\ln(\gamma_i m_i-\pi_i x_i)+\ln\Big(\sum_{i\in\mathcal{I}}(1-\gamma_i)m_i-C\Big(\sum_{i\in\mathcal{I}}y_i\Big)\Big)$$

s. t. (4), (5), (7), (8)

变量：$\{y_i\}_{i\in\mathcal{I}}$

为了求解上述问题，引入辅助变量 $\boldsymbol{\mu}=\sum_{i\in\mathcal{I}}\boldsymbol{y}_i$，将其垂直分解为两个子问题。

当给定 $\boldsymbol{\mu}$ 时，问题 P3 对应的底层问题为：

$$\text{P4}:\boldsymbol{V}_{\boldsymbol{\mu}}=\max\sum_{i\in\mathcal{I}}\ln(\boldsymbol{\gamma}_i\boldsymbol{A}_i(\boldsymbol{x}_i+\boldsymbol{y}_i)-\pi_i\boldsymbol{x}_i)+\ln\Big(\sum_{i\in\mathcal{I}}(1-\boldsymbol{\gamma}_i)\boldsymbol{A}_i(\boldsymbol{x}_i+\boldsymbol{y}_i)-C(\boldsymbol{\mu})\Big)$$

$$\text{s. t.}\sum_{i\in\mathcal{I}}\boldsymbol{y}_i=\boldsymbol{\mu} \tag{10}$$

变量：$\{\boldsymbol{y}_i\geqslant 0\}_{i\in\mathcal{I}}$

为了便于表达，在问题 P4 中引入参数 $\boldsymbol{A}_i$：

$$\boldsymbol{A}_i=(R+rt_i)\frac{\mathrm{e}^{-\theta rt_i}}{\sum_{i\in\mathcal{I}}\boldsymbol{x}_i+\boldsymbol{\mu}}$$

在问题 P4 中，$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$，$\{\boldsymbol{x}_i\}_{i\in\mathcal{I}}$ 都是给定的，同时当给定 $\boldsymbol{\mu}$ 时，对应每个移动终端设备，参数 $\boldsymbol{A}_i$ 是一个固定的量。因此，在给定 $\boldsymbol{\mu}$ 的条件下，问题 P4 是严格凸优化的[25]。

在求解完底层问题 P4 并且得到每个给定的 $\boldsymbol{\mu}$ 对应的 $\boldsymbol{V}_{\boldsymbol{\mu}}$ 之后，继续寻找最大的 $\boldsymbol{V}_{\boldsymbol{\mu}}$ 对应的最优的 $\boldsymbol{\mu}$。因此问题 P3 对应的顶层问题为：

$$\text{P5}:\max \boldsymbol{V}_{\boldsymbol{\mu}}$$

变量：$0\leqslant\boldsymbol{\mu}\leqslant C^{\text{tot}}$

然而，解决顶层问题 P5 的难点在于不能表达出 $\boldsymbol{V}_{\boldsymbol{\mu}}$ 的解析式。通过分析可以得到，$\boldsymbol{\mu}$ 的可行域是一个固定的间隔：$[0,C^{\text{tot}}]$。因此，可以利用步长足够小的线性搜索来求解顶层问题 P5。

3.1.2 求解问题 P3 得到 $\{\boldsymbol{y}_i\}_{i\in\mathcal{I}}$

本节首先对问题 P3 的底层问题 P4 进行求解。因为约束条件(10)在问题 P4 的最优处是严格约束的，可以通过在问题 P4 中引入松弛变量 $\boldsymbol{\lambda}$ 的方式进行求解。相应的拉格朗日函数为：

$$L(\{\boldsymbol{y}_i\}_{i\in\mathcal{I}},\boldsymbol{\lambda})=\sum_{i\in\mathcal{I}}\ln(\boldsymbol{\gamma}_i\boldsymbol{A}_i(\boldsymbol{x}_i+\boldsymbol{y}_i)-\pi_i\boldsymbol{x}_i)+\ln\Big(\sum_{i\in\mathcal{I}}(1-\boldsymbol{\gamma}_i)\boldsymbol{A}_i(\boldsymbol{x}_i+\boldsymbol{y}_i)-C(\boldsymbol{\mu})\Big)+\boldsymbol{\lambda}\Big(\boldsymbol{\mu}-\sum_{i\in\mathcal{I}}\boldsymbol{y}_i\Big) \tag{11}$$

根据式(11)，通过对 $\boldsymbol{y}_i$ 求偏导可得：

$$\frac{\partial L}{\partial\boldsymbol{y}_i}=\frac{\boldsymbol{\gamma}_i\boldsymbol{A}_i}{\boldsymbol{\gamma}_i\boldsymbol{A}_i(\boldsymbol{x}_i+\boldsymbol{y}_i)-\pi_i\boldsymbol{x}_i}+\frac{(1-\boldsymbol{\gamma}_i)\boldsymbol{A}_i}{\sum_{i\in\mathcal{I}}(1-\boldsymbol{\gamma}_i)\boldsymbol{A}_i(\boldsymbol{x}_i+\boldsymbol{y}_i)-C(\boldsymbol{\mu})}-\boldsymbol{\lambda}=0 \tag{12}$$

在式(12)中引入辅助变量 $\boldsymbol{Q}$：

$$\boldsymbol{Q}=\sum_{i\in\mathcal{I}}(1-\boldsymbol{\gamma}_i)\boldsymbol{A}_i(\boldsymbol{x}_i+\boldsymbol{y}_i)-C(\boldsymbol{\mu}) \tag{13}$$

根据式(13)可以得到式(12)化简后的结果为：

$$\boldsymbol{x}_i+\boldsymbol{y}_i=\frac{1}{\boldsymbol{\lambda}-\dfrac{(1-\boldsymbol{\gamma}_i)\boldsymbol{A}_i}{\boldsymbol{Q}}}+\frac{\pi_i\boldsymbol{x}_i}{\boldsymbol{\gamma}_i\boldsymbol{A}_i} \tag{14}$$

将式(14)代入式(13)，变形得到：

$$\boldsymbol{Q}=\sum_{i\in\mathcal{I}}(1-\boldsymbol{\gamma}_i)\boldsymbol{A}_i\left(\frac{1}{\boldsymbol{\lambda}-\dfrac{(1-\boldsymbol{\gamma}_i)\boldsymbol{A}_i}{\boldsymbol{Q}}}+\frac{\pi_i\boldsymbol{x}_i}{\boldsymbol{\gamma}_i\boldsymbol{A}_i}\right)-C(\boldsymbol{\mu})$$

在化简得到关于 $\boldsymbol{Q}$ 的表达式之后，可以发现，在给定 $\boldsymbol{\lambda}$ 的情况下，$\boldsymbol{Q}$ 值是唯一确定的($\boldsymbol{Q}>0$)。通过这一结论可以对式(11)进行求解。在式(12)中，结合问题 P4 的约束条件 $\{\boldsymbol{y}_i\geqslant 0\}_{i\in\mathcal{I}}$，枚举 $\mathcal{I}$ 所有可能的子集，用 ψ 表示每次枚举的集合：$\psi=\{i\in\mathcal{I}\mid\boldsymbol{y}_i>0\}$。对于 ψ 内的 $\boldsymbol{y}_i$，根据式(14)，可以转换得到 $\boldsymbol{y}_i$ 相应的表达式；对于不在集合 ψ 内的 $\boldsymbol{y}_i$，直接设置其等于 0。

$$\boldsymbol{y}_i=\begin{cases}\dfrac{1}{\boldsymbol{\lambda}-\dfrac{(1-\boldsymbol{\gamma}_i)\boldsymbol{A}_i}{\boldsymbol{Q}}}+\dfrac{\pi_i\boldsymbol{x}_i}{\boldsymbol{\gamma}_i\boldsymbol{A}_i}-\boldsymbol{x}_i, & i\in\psi\\ 0, & i\in\mathcal{I}\backslash\psi\end{cases}\tag{15}$$

根据上述推导，本文提出了一种双层对分搜索算法（见算法1）来求解问题P4。双层对分搜索算法分为两部分执行，内部对分搜索确定 $\boldsymbol{Q}$ 的值，同时利用得到的 $\boldsymbol{Q}$ 进行外部对分搜索以确定 $\boldsymbol{\lambda}$ 以及对应的 $\{\boldsymbol{y}_i\}_{i\in\mathcal{I}}$。具体步骤如算法1所示。

算法 1

输入：$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$，$\{\mathbf{x}_i\}_{i\in\mathcal{I}}$，$\boldsymbol{\mu}$

输出：$\{\mathbf{y}_i\}_{i\in\mathcal{I}}$

1. 初始化参数：设置 $\boldsymbol{\lambda}$ 对分上限 $\bar{\boldsymbol{\lambda}}$ 为一个非常大的数，对分下限 $\underline{\boldsymbol{\lambda}}$ 为0，对分搜索误差 ζ 为非常小的数。
2. While $|\bar{\boldsymbol{\lambda}}-\underline{\boldsymbol{\lambda}}|>\zeta$ do
3. 更新 $\boldsymbol{\lambda}^{\mathrm{cur}}=\dfrac{\bar{\boldsymbol{\lambda}}+\underline{\boldsymbol{\lambda}}}{2}$。
4. 更新 $\mathbf{Q}$ 对分上限 $\bar{\mathbf{Q}}$ 为一个非常大的数，对分下限 $\underline{\mathbf{Q}}$ 为0，搜索误差 ε 为非常小的数。
5. While $|\bar{\mathbf{Q}}-\underline{\mathbf{Q}}|>\varepsilon$ do
6. 更新 $\mathbf{Q}^{\mathrm{cur}}=\dfrac{\bar{\mathbf{Q}}+\underline{\mathbf{Q}}}{2}$。设置

$$\mathbf{Q}^{*}=\sum_{i\in\mathcal{I}}(1-\boldsymbol{\gamma}_i)\mathbf{A}_i\left(\frac{1}{\boldsymbol{\lambda}-\dfrac{(1-\boldsymbol{\gamma}_i)\mathbf{A}_i}{\mathbf{Q}}}+\frac{\pi_i\mathbf{x}_i}{\boldsymbol{\gamma}_i\mathbf{A}_i}\right)-\mathrm{C}(\boldsymbol{\mu})$$

7. If $\mathbf{Q}^{*}<\mathbf{Q}^{\mathrm{cur}}$ then
8. 更新搜索上限 $\bar{\mathbf{Q}}=\mathbf{Q}^{\mathrm{cur}}$，执行步骤6。
9. Else if $\mathbf{Q}^{*}>\mathbf{Q}^{\mathrm{cur}}$ then
10. 更新搜索下限 $\underline{\mathbf{Q}}=\mathbf{Q}^{\mathrm{cur}}$，执行步骤6
11. Else
12. 跳出内部循环，执行步骤15。
13. End if
14. End while
15. 对集合 $\mathcal{I}$ 所有可能的子集进行枚举，根据式(15)得到各组子集对应的 $\{\mathbf{y}_i\}_{i\in\mathcal{I}}$。
16. 选择满足条件 $\{\mathbf{y}_i\geqslant 0\}_{i\in\mathcal{I}}$ 的各组 $\{\mathbf{y}_i\}_{i\in\mathcal{I}}$ 代入 $\sum_{i\in\mathcal{I}}\ln(\boldsymbol{\gamma}_i\mathbf{A}_i(\mathbf{x}_i+\mathbf{y}_i)-\pi_i\mathbf{x}_i)+\ln(\sum_{i\in\mathcal{I}}(1-\boldsymbol{\gamma}_i)\mathbf{A}_i(\mathbf{x}_i+\mathbf{y}_i)-\mathrm{C}(\boldsymbol{\mu}))$ 中得到相应的目标函数值。
17. 比较步骤16中所有得到的目标函数值，令 $\{\mathbf{y}_i^{*}\}_{i\in\mathcal{I}}$ 为其中最大函数值对应的 $\{\mathbf{y}_i\}_{i\in\mathcal{I}}$（如果所有枚举的子集合所对应的 $\{\mathbf{y}_i\}_{i\in\mathcal{I}}$ 均不满足约束条件，更新 $\mathbf{y}_i^{*}=0$，$i\in\mathcal{I}$）。
18. 在得到 $\{\mathbf{y}_i^{*}\}_{i\in\mathcal{I}}$ 之后，计算相应的 $\sum_{i\in\mathcal{I}}\mathbf{y}_i^{*}$。
19. If $\sum_{i\in\mathcal{I}}\mathbf{y}_i^{*}>\boldsymbol{\mu}$ then
20. 更新对分搜索下限 $\underline{\boldsymbol{\lambda}}=\boldsymbol{\lambda}^{\mathrm{cur}}$，执行步骤3。
21. Else if $\sum_{i\in\mathcal{I}}\mathbf{y}_i^{*}<\boldsymbol{\mu}$ then
22. 新对分搜索上限 $\bar{\boldsymbol{\lambda}}=\boldsymbol{\lambda}^{\mathrm{cur}}$，执行步骤3。
23. Else
24. 跳出外部循环，得到最优的 $\{\mathbf{y}_i\}_{i\in\mathcal{I}}$。
25. End if
26. End while

在底层问题P4求解完成后，对 $\boldsymbol{\mu}\in[0,C^{\mathrm{tot}}]$ 执行线性搜索继续求解顶层问题P5。同时，为了提高线性搜索的效率，得到问题P4可行的必要条件：

$$\boldsymbol{\mu}\geqslant\frac{\max\{C(\boldsymbol{\mu})-\sum_{i\in\mathcal{I}}(1-\boldsymbol{\gamma}_i)\boldsymbol{A}_i\boldsymbol{x}_i,0\}}{\max\limits_{i\in\mathcal{I}}\{(1-\boldsymbol{\gamma}_i)\boldsymbol{A}_i\}}\tag{16}$$

根据式(16)，对于每一个满足条件的 $\boldsymbol{\mu}$，采用算法1来求解对应的底层问题P4，并获取相应的 $\boldsymbol{V}_{\boldsymbol{\mu}}$。最后，根据所有 $\boldsymbol{\mu}$ 对应的 $\boldsymbol{V}_{\boldsymbol{\mu}}$ 得到最优的 $\{\boldsymbol{y}_i\}_{i\in\mathcal{I}}$。具体算法流程如算法2所示。

算法 2

输入：$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$，$\{\mathbf{x}_i\}_{i\in\mathcal{I}}$

输出：$\{\mathbf{y}_i\}_{i\in\mathcal{I}}$

1. 初始化：设置 Δ 为一个非常小的步长，并设置 $\boldsymbol{\mu}=\Delta$。
2. For $\boldsymbol{\mu}=\Delta:\Delta:C^{\mathrm{tot}}$ do
3. If $\boldsymbol{\mu}\geqslant\dfrac{\max\{C(\boldsymbol{\mu})-\sum_{i\in\mathcal{I}}(1-\boldsymbol{\gamma}_i)\mathbf{A}_i\mathbf{x}_i,0\}}{\max\limits_{i\in\mathcal{I}}\{(1-\boldsymbol{\gamma}_i)\mathbf{A}_i\}}$ then
4. 利用算法1求解得到 $\boldsymbol{\mu}$ 对应的 $\mathbf{V}_{\boldsymbol{\mu}}$ 以及对应的 $\{\mathbf{y}_i\}_{i\in\mathcal{I}}$。
5. End if
6. End for
7. 比较所有 $\boldsymbol{\mu}$ 中最大的 $\mathbf{V}_{\boldsymbol{\mu}}$ 所对应的 $\{\mathbf{y}_i\}_{i\in\mathcal{I}}$ 并输出。

3.1.3 给定变量 $\{\boldsymbol{y}_i\}_{i\in\mathcal{I}}$ 时的内部问题P6

当给定变量 $\{\boldsymbol{y}_i\}_{i\in\mathcal{I}}$ 时，问题P2可以转换成以下内部问题：

$$\mathrm{P6}:\max\sum_{i\in\mathcal{I}}\ln(\boldsymbol{\gamma}_i\boldsymbol{m}_i-\pi_i\boldsymbol{x}_i)+\ln(\sum_{i\in\mathcal{I}}(1-\boldsymbol{\gamma}_i)\boldsymbol{m}_i-C(\sum_{i\in\mathcal{I}}\boldsymbol{y}_i))$$

s.t. (4)，(5)

$$0\leqslant\boldsymbol{x}_i\leqslant C_i^{\max}$$

变量：$\{\boldsymbol{x}_i\}_{i\in\mathcal{I}}$

同样，问题P6也是一个非凸优化问题。通过引入辅助变量 $\boldsymbol{\mu}'$ 将其垂直分解为两个子问题。

当给定 $\boldsymbol{\mu}'$ 时，底层问题的表达形式为：

$$\mathrm{P7}:\boldsymbol{V}_{\boldsymbol{\mu}'}=\max\sum_{i\in\mathcal{I}}\ln(\boldsymbol{\gamma}_i\boldsymbol{B}_i(\boldsymbol{x}_i+\boldsymbol{y}_i)-\pi_i\boldsymbol{x}_i)+\ln(\sum_{i\in\mathcal{I}}(1-\boldsymbol{\gamma}_i)\boldsymbol{B}_i(\boldsymbol{x}_i+\boldsymbol{y}_i)-C(\sum_{i\in\mathcal{I}}\boldsymbol{y}_i))$$

s.t. $\sum_{i\in\mathcal{I}}\boldsymbol{x}_i=\boldsymbol{\mu}'$

变量：$\{\boldsymbol{x}_i\}_{i\in\mathcal{I}}$

其中，参数 $\boldsymbol{B}_i=(R+rt_i)\dfrac{\mathrm{e}^{-\theta zt_i}}{\boldsymbol{\mu}'+\sum_{i\in\mathcal{I}}\boldsymbol{y}_i}$。

问题P6对应的顶层问题的表示形式为：

$$\mathrm{P8}:\max\boldsymbol{V}_{\boldsymbol{\mu}'}$$

变量：$0\leqslant\boldsymbol{\mu}'\leqslant\sum_{i\in\mathcal{I}}C_i^{\max}$

3.1.4 求解问题P6得到 $\{\boldsymbol{x}_i\}_{i\in\mathcal{I}}$

本节首先对问题P6的底层问题P7进行求解。因为问题P7是凸优化问题[25]，引入松弛变量 $\boldsymbol{\beta}$，得到相应的拉格朗日函数为：

$$L(\{\boldsymbol{x}_i\}_{i\in\mathcal{I}},\boldsymbol{\beta})=\sum_{i\in\mathcal{I}}\ln(\boldsymbol{\gamma}_i\boldsymbol{B}_i(\boldsymbol{x}_i+\boldsymbol{y}_i)-\pi_i\boldsymbol{x}_i)+\ln(\sum_{i\in\mathcal{I}}(1-\boldsymbol{\gamma}_i)\boldsymbol{B}_i(\boldsymbol{x}_i+\boldsymbol{y}_i)-C(\sum_{i\in\mathcal{I}}\boldsymbol{y}_i))+\boldsymbol{\beta}(\boldsymbol{\mu}'-\sum_{i\in\mathcal{I}}\boldsymbol{x}_i)\tag{17}$$

在式(17)中对 $\boldsymbol{x}_i$ 求偏导可得：

$$\frac{\partial L}{\partial\boldsymbol{x}_i}=\frac{\boldsymbol{\gamma}_i\boldsymbol{B}_i-\pi_i}{\boldsymbol{\gamma}_i\boldsymbol{B}_i(\boldsymbol{x}_i+\boldsymbol{y}_i)-\pi_i\boldsymbol{x}_i}+\frac{(1-\boldsymbol{\gamma}_i)\boldsymbol{B}_i}{\sum_{i\in\mathcal{I}}(1-\boldsymbol{\gamma}_i)\boldsymbol{B}_i(\boldsymbol{x}_i+\boldsymbol{y}_i)-C(\sum_{i\in\mathcal{I}}\boldsymbol{y}_i)}-\boldsymbol{\beta}=0\tag{18}$$

在式(18)中引入辅助变量 $\boldsymbol{W}$：

$$\boldsymbol{W}=\sum_{i\in\mathcal{I}}(1-\boldsymbol{\gamma}_i)\boldsymbol{B}_i(\boldsymbol{x}_i+\boldsymbol{y}_i)-C(\sum_{i\in\mathcal{I}}\boldsymbol{y}_i)\tag{19}$$

将式(19)代入式(18)，可以得到化简后的表达式如下：

$$\boldsymbol{x}_i=\frac{1}{\boldsymbol{\beta}-\frac{(1-\boldsymbol{\gamma}_i)\boldsymbol{B}_i}{\boldsymbol{W}}}-\frac{\boldsymbol{\gamma}_i\boldsymbol{B}_i\boldsymbol{y}_i}{\boldsymbol{\gamma}_i\boldsymbol{B}_i-\pi_i} \tag{20}$$

根据式(19)以及式(20),可以化简得到:

$$\boldsymbol{W}=\sum_{i\in\mathcal{I}}(1-\boldsymbol{\gamma}_i)\boldsymbol{B}_i\left(\frac{1}{\boldsymbol{\beta}-\frac{(1-\boldsymbol{\gamma}_i)\boldsymbol{B}_i}{\boldsymbol{W}}}-\frac{\pi_i\boldsymbol{y}_i}{\boldsymbol{\gamma}_i\boldsymbol{B}_i-\pi_i}+\boldsymbol{y}_i\right)-C\left(\sum_{i\in\mathcal{I}}\boldsymbol{y}_i\right)$$

结合问题 P7 的约束条件,枚举 $\mathcal{I}$ 所有可能的子集,用集合 Φ 表示每次枚举的集合。对于集合 Φ 内的 $\boldsymbol{x}_i$,根据式(20)转换得到 $\boldsymbol{x}_i$ 相应的表达式;对于不在集合 Φ 内的 $\boldsymbol{x}_i$,设置其等于 0。

$$\boldsymbol{x}_i=\begin{cases}\dfrac{1}{\boldsymbol{\beta}-\frac{(1-\boldsymbol{\gamma}_i)\boldsymbol{B}_i}{\boldsymbol{W}}}-\dfrac{\boldsymbol{\gamma}_i\boldsymbol{B}_i\boldsymbol{y}_i}{\boldsymbol{\gamma}_i\boldsymbol{B}_i-\pi_i}, & i\in\Phi\\ 0, & i\in\mathcal{I}\backslash\Phi\end{cases} \tag{21}$$

根据上述部分的推导,利用算法 3 来求解问题 P6。内部对分搜索确定 $\boldsymbol{W}$ 的值,同时利用得到的 $\boldsymbol{W}$ 进行外部对分搜索以确定 $\boldsymbol{\beta}$ 以及对应的最优 $\{\boldsymbol{x}_i\}_{i\in\mathcal{I}}$。具体步骤如算法 3 所示。

算法 3

输入:$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$,$\{\boldsymbol{y}_i\}_{i\in\mathcal{I}}$,$\boldsymbol{\mu}'$

输出:$\{\mathbf{x}_i\}_{i\in\mathcal{I}}$

1. 初始化参数:设置 $\boldsymbol{\beta}$ 对分上限 $\bar{\boldsymbol{\beta}}$ 为一个非常大的数,对分下限 $\underline{\boldsymbol{\beta}}$ 为 0,搜索误差 ζ 为非常小的数。
2. While $|\bar{\boldsymbol{\beta}}-\underline{\boldsymbol{\beta}}|>\zeta$ do
3. 更新 $\boldsymbol{\beta}^{\mathrm{cur}}=\frac{\bar{\boldsymbol{\beta}}+\underline{\boldsymbol{\beta}}}{2}$。
4. 更新 $\mathbf{W}$ 对分上限 $\overline{\mathbf{W}}$ 为一个非常大的数,对分下限 $\underline{\mathbf{W}}$ 为 0,搜索误差 ε 为非常小的数。
5. While $|\overline{\mathbf{W}}-\underline{\mathbf{W}}|>\varepsilon$ do
6. 更新 $\mathbf{W}^{\mathrm{cur}}=\frac{\overline{\mathbf{W}}+\underline{\mathbf{W}}}{2}$。设置

$$\mathbf{W}^*=\sum_{i\in\mathcal{I}}(1-\boldsymbol{\gamma}_i)\mathbf{B}_i\left(\frac{1}{\boldsymbol{\beta}-\frac{(1-\boldsymbol{\gamma}_i)\mathbf{B}_i}{\mathbf{W}}}-\frac{\pi_i\mathbf{y}_i}{\boldsymbol{\gamma}_i\mathbf{B}_i-\pi_i}+\mathbf{y}_i\right)-C\left(\sum_{i\in\mathcal{I}}\mathbf{y}_i\right)$$

7. If $\mathbf{W}^*<\mathbf{W}^{\mathrm{cur}}$ then
8. 更新搜索上限 $\overline{\mathbf{W}}=\mathbf{W}^{\mathrm{cur}}$,执行步骤 6。
9. Else if $\mathbf{W}^*>\mathbf{W}^{\mathrm{cur}}$ then
10. 更新搜索下限 $\underline{\mathbf{W}}=\mathbf{W}^{\mathrm{cur}}$,执行步骤 6。
11. Else
12. 跳出内部循环,执行步骤 15。
13. End if
14. End while
15. 对集合 $\mathcal{I}$ 所有可能的子集进行枚举,根据式(21)得到各组子集对应的 $\{\mathbf{x}_i\}_{i\in\mathcal{I}}$。
16. 选择满足条件 $\{0\leqslant\mathbf{x}_i\leqslant C_i^{\max}\}_{i\in\mathcal{I}}$ 的各组 $\{\mathbf{x}_i\}_{i\in\mathcal{I}}$ 代入 $\sum_{i\in\mathcal{I}}\ln(\boldsymbol{\gamma}_i\mathbf{B}_i(\mathbf{x}_i+\mathbf{y}_i)-\pi_i\mathbf{x}_i)+\ln(\sum_{i\in\mathcal{I}}(1-\boldsymbol{\gamma}_i)\mathbf{B}_i(\mathbf{x}_i+\mathbf{y}_i)-C(\sum_{i\in\mathcal{I}}\mathbf{y}_i))$ 得到相应的目标函数值。
17. 比较步骤 16 中所有得到的目标函数值,令 $\{\mathbf{x}_i^*\}_{i\in\mathcal{I}}$ 为其中最大函数值对应的 $\{\mathbf{x}_i\}_{i\in\mathcal{I}}$(如果所有枚举的子集合所对应的 $\{\mathbf{x}_i\}_{i\in\mathcal{I}}$ 均不满足约束条件,更新 $\{\mathbf{x}_i^*\}_{i\in\mathcal{I}}$ 为 0)。
18. 在得到 $\{\mathbf{x}_i^*\}_{i\in\mathcal{I}}$ 后,计算相应的 $\sum_{i\in\mathcal{I}}\mathbf{x}_i^*$。
19. If $\sum_{i\in\mathcal{I}}\mathbf{x}_i^*>\boldsymbol{\mu}'$ then
20. 更新对分搜索下限 $\underline{\boldsymbol{\beta}}=\boldsymbol{\beta}^{\mathrm{cur}}$,执行步骤 3。
21. Else if $\sum_{i\in\mathcal{I}}\mathbf{x}_i^*<\boldsymbol{\mu}'$ then
22. 更新对分搜索上限 $\bar{\boldsymbol{\beta}}=\boldsymbol{\beta}^{\mathrm{cur}}$,执行步骤 3。
23. Else
24. 跳出外部循环,得到最优的 $\{\mathbf{x}_i\}_{i\in\mathcal{I}}$。
25. End if
26. End while

对于问题 P6 的顶层问题 P8,通过对 $\boldsymbol{\mu}'$ 执行线性搜索进行求解。对于每一个满足枚举条件的 $\boldsymbol{\mu}'\in[0,\sum_{i\in\mathcal{I}}C_i^{\max}]$,采用算法 4 来求解底层问题 P6,并获取相应的计算结果。最后,根据所有 $\boldsymbol{\mu}'$ 的计算结果得到相应的最优 $\{\boldsymbol{x}_i\}_{i\in\mathcal{I}}$。

算法 4

输入:$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$,$\{\boldsymbol{y}_i\}_{i\in\mathcal{I}}$

输出:$\{\mathbf{x}_i\}_{i\in\mathcal{I}}$

1. 初始化:设置 Δ 为一个非常小的步长,并设置 $\boldsymbol{\mu}'=\Delta$。
2. For $\boldsymbol{\mu}'=\Delta:\Delta:\sum_{i\in\mathcal{I}}C_i^{\max}$ do
3. If $\boldsymbol{\mu}'\geqslant\frac{\max\{C(\sum_{i\in\mathcal{I}}\mathbf{y}_i)-\sum_{i\in\mathcal{I}}(1-\boldsymbol{\gamma}_i)\mathbf{B}_i\mathbf{y}_i,0\}}{\max_{i\in\mathcal{I}}\{(1-\boldsymbol{\gamma}_i)\mathbf{B}_i\}}$ then
4. 利用算法 3 得到 $\boldsymbol{\mu}'$ 对应的 $\mathbf{V}_{\boldsymbol{\mu}'}$ 以及相对的 $\{\mathbf{x}_i\}_{i\in\mathcal{I}}$。
5. End if
6. End for
7. 比较得到所有 $\boldsymbol{\mu}'$ 中最大的 $\mathbf{V}_{\boldsymbol{\mu}'}$ 所对应的 $\{\mathbf{x}_i\}_{i\in\mathcal{I}}$ 并输出。

3.1.5 求解问题 P2 得到 $\{\boldsymbol{x}_i\}_{i\in\mathcal{I}}$ 和 $\{\boldsymbol{y}_i\}_{i\in\mathcal{I}}$

利用基于循环块坐标下降的算法 5 来求解问题 P2。在该算法中,将 $\{\boldsymbol{x}_i\}_{i\in\mathcal{I}}$ 与 $\{\boldsymbol{y}_i\}_{i\in\mathcal{I}}$ 作为两部分的变量进行迭代,直到 $\{\boldsymbol{x}_i\}_{i\in\mathcal{I}}$ 不再发生变化。具体步骤如算法 5 所示。

算法 5

输入:$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$

输出:$\{\mathbf{x}_i\}_{i\in\mathcal{I}}$,$\{\mathbf{y}_i\}_{i\in\mathcal{I}}$

1. 初始化:随机给定每个移动终端设备的本地计算资源 $\{\mathbf{x}_i\}_{i\in\mathcal{I}}$:$\{\mathbf{x}_i^{\mathrm{ini}}\}_{i\in\mathcal{I}}\in[0,C_i^{\max}]$。设置 $\{\mathbf{x}_i^{\mathrm{pre}}\}_{i\in\mathcal{I}}=\emptyset$,$\{\mathbf{x}_i^{\mathrm{cur}}\}_{i\in\mathcal{I}}=\emptyset$。
2. 更新 $\{\mathbf{x}_i^{\mathrm{pre}}\}_{i\in\mathcal{I}}=\{\mathbf{x}_i^{\mathrm{ini}}\}_{i\in\mathcal{I}}$。
3. While $\{\mathbf{x}_i^{\mathrm{cur}}\}_{i\in\mathcal{I}}\neq\{\mathbf{x}_i^{\mathrm{pre}}\}_{i\in\mathcal{I}}$ do
4. 根据给定的 $\{\mathbf{x}_i^{\mathrm{pre}}\}_{i\in\mathcal{I}}$,利用 3.1.2 节中提出的算法 2 得到最优的 $\{\mathbf{y}_i\}_{i\in\mathcal{I}}$。
5. 根据步骤 4 中得到的 $\{\mathbf{y}_i\}_{i\in\mathcal{I}}$,利用 3.1.4 节提出的算法 4 得到当前最优的 $\{\mathbf{x}_i\}_{i\in\mathcal{I}}$,用 $\{\mathbf{x}_i^{\mathrm{cur}}\}_{i\in\mathcal{I}}$ 表示。
6. If $\{\mathbf{x}_i^{\mathrm{cur}}\}_{i\in\mathcal{I}}=\{\mathbf{x}_i^{\mathrm{pre}}\}_{i\in\mathcal{I}}$ then
7. 跳出循环,执行步骤 12。
8. Else
9. 更新 $\{\mathbf{x}_i^{\mathrm{pre}}\}_{i\in\mathcal{I}}=\{\mathbf{x}_i^{\mathrm{cur}}\}_{i\in\mathcal{I}}$,回到步骤 4 继续迭代过程。
10. End if
11. End while
12. 输出问题 P2 对应的最优 $\{\mathbf{x}_i\}_{i\in\mathcal{I}}$,$\{\mathbf{y}_i\}_{i\in\mathcal{I}}$。

3.2 给定资源分配变量 $\{\boldsymbol{x}_i\}_{i\in\mathcal{I}}$ 和 $\{\boldsymbol{y}_i\}_{i\in\mathcal{I}}$ 时的子问题 P9

在给定计算资源分配变量 $\{\boldsymbol{x}_i\}_{i\in\mathcal{I}}$,$\{\boldsymbol{y}_i\}_{i\in\mathcal{I}}$ 的情况下,问题 P1 可以转化为以下关于移动终端设备收益分享的优化问题:

$$\text{P9}:\max\sum_{i\in\mathcal{I}}\ln(\boldsymbol{\gamma}_i\boldsymbol{E}_i-\boldsymbol{F}_i)+\ln(\sum_{i\in\mathcal{I}}(1-\boldsymbol{\gamma}_i)\boldsymbol{E}_i-\boldsymbol{G})$$

$$\text{变量}:0\leqslant\boldsymbol{\gamma}_i\leqslant1,\forall\, i\in\mathcal{I}$$

为了便于表达，在问题 P9 中引入的参数表示如下：

$$\boldsymbol{E}_i=(R+rt_i)\frac{(\boldsymbol{x}_i+\boldsymbol{y}_i)}{\sum\limits_{i\in\mathcal{I}}\boldsymbol{x}_i+\sum\limits_{i\in\mathcal{I}}\boldsymbol{y}_i}\mathrm{e}^{-\theta zt_i},\ \forall\, i\in\mathcal{I} \tag{22}$$

$$\boldsymbol{F}_i=\pi_i\boldsymbol{x}_i,\ \forall\, i\in\mathcal{I} \tag{23}$$

$$\boldsymbol{G}=C(\sum_{i\in\mathcal{I}}\boldsymbol{y}_i) \tag{24}$$

在给定$\{\boldsymbol{x}_i\}_{i\in\mathcal{I}}$，$\{\boldsymbol{y}_i\}_{i\in\mathcal{I}}$的情况下，$\boldsymbol{E}_i$，$\boldsymbol{F}_i$ 和 $\boldsymbol{G}$ 都是常数，因此问题 P9 是关于$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$的严格凸优化问题[25]。

根据问题 P9 的凸性，可以利用 KKT（Karush-Kuhn-Tucker）条件推导出其最优解。根据 KKT 定则，当问题 P9 取到最优值时，下列式子恒成立（集合 Ω 表示 $\mathcal{I}$ 的子集，$\Omega=\{i\in\mathcal{I}\mid 0<\boldsymbol{\gamma}_i<1\}$）：

$$\boldsymbol{\gamma}_i\boldsymbol{E}_i=\sum_{i\in\Omega}\boldsymbol{E}_i-\sum_{i\in\Omega}\boldsymbol{E}_i\boldsymbol{\gamma}_i-\boldsymbol{G}+\boldsymbol{F}_i,\ \forall\, i\in\Omega \tag{25}$$

根据式(25)，可以得到关于 $\boldsymbol{\gamma}_i$ 的表达式，对于每个不在 Ω 内的移动终端设备 i，将直接给定其对应的收益分享函数 $\boldsymbol{\gamma}_i=1$。

$$\boldsymbol{\gamma}_i=\begin{cases}\dfrac{\sum\limits_{i\in\Omega}\boldsymbol{E}_i-\boldsymbol{G}+\boldsymbol{F}_i}{\boldsymbol{E}_i\cdot(|\Omega|+1)}, & i\in\Omega\\ 1, & i\in\mathcal{I}\backslash\Omega\end{cases} \tag{26}$$

在式(26)的基础上，本节提出了算法 6 来求解问题 P9。具体来说，对 $\mathcal{I}$ 的所有可能的子集进行枚举，并根据式(26)设置对应的$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$。对于上述所有满足约束条件$\{0\leqslant\boldsymbol{\gamma}_i\leqslant 1\}_{i\in\mathcal{I}}$的各组$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$，将其代入问题 P9 中的目标函数并得到相应的值。最后找到最大的目标函数值对应的$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$。具体步骤如算法 6 所示。

算法 6

输入：$\{\mathbf{x}_i\}_{i\in\mathcal{I}}$，$\{\mathbf{y}_i\}_{i\in\mathcal{I}}$

输出：$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$

1. 初始化：根据式(22)－式(24)得到$\{\mathbf{E}_i\}_{i\in\mathcal{I}}$，$\{\mathbf{F}_i\}_{i\in\mathcal{I}}$以及 $\mathbf{G}$。
2. While $\mathcal{I}$的子集仍未枚举完 do
3. 　更新 Ω 为一个仍未枚举的 $\mathcal{I}$的子集。
4. 　根据式(26)设置 $\boldsymbol{\gamma}_i$ 的值。
5. 　If $0\leqslant\boldsymbol{\gamma}_i\leqslant 1,\forall\, i\in\mathcal{I}$ then
6. 　　将$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$代入$\sum\limits_{i\in\mathcal{I}}\ln(\boldsymbol{\gamma}_i\mathbf{E}_i-\mathbf{F}_i)+\ln(\sum\limits_{i\in\mathcal{I}}(1-\boldsymbol{\gamma}_i)\mathbf{E}_i-\mathbf{G})$得到相应的目标函数值。
7. 　Else
8. 　　执行步骤 3，枚举下一集合。
9. 　End if
10. End while
11. 比较所有符合情况的子集对应的函数值，并输出其中最大值所对应的$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$。

3.3　求解原始问题 P1

结合 3.1 节与 3.2 节提出的关于计算资源分配和收益分享的优化算法，利用基于循环块坐标下降思想的多层分解算法（算法 7）来解决原始问题 P1，将$\{\boldsymbol{x}_i\}_{i\in\mathcal{I}}$，$\{\boldsymbol{y}_i\}_{i\in\mathcal{I}}$看成一个部分，与$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$作为两部分的变量进行迭代，具体步骤如下。

首先，随机给定$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$的值。当给定$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$时，根据 3.1 节中提出的算法 5 更新得到$\{\boldsymbol{x}_i\}_{i\in\mathcal{I}}$，$\{\boldsymbol{y}_i\}_{i\in\mathcal{I}}$。

在得到了相应$\{\boldsymbol{x}_i\}_{i\in\mathcal{I}}$，$\{\boldsymbol{y}_i\}_{i\in\mathcal{I}}$的基础上，使用 3.2 节提出的算法 6 得到$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$。然后，将此次得到的$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$与本轮迭代中当前设置的$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$进行对比，如果对应的 $\boldsymbol{\gamma}_i$ 存在不相等的部分，则继续执行迭代。上述过程一直迭代到$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$不再发生变化为止。整个多层分解算法的流程如图 3 所示，具体步骤如算法 7 所示。

算法 7

输入：$\{\mathrm{C}_i^{\max}\}_{i\in\mathrm{I}}$，$\{\pi_i\}_{i\in\mathrm{I}}$，$\{\mathrm{t}_i\}\in_{i\in\mathrm{I}}$

输出：$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$，$\{\mathbf{x}_i\}_{i\in\mathcal{I}}$，$\{\mathbf{y}_i\}_{i\in\mathcal{I}}$

1. 初始化：在 0－1 范围内随机给定每个移动终端设备的收益分享变量 γ_i：$\{\boldsymbol{\gamma}_i^{\mathrm{ini}}\}_{i\in\mathcal{I}}\in[0,1]$。设置$\{\boldsymbol{\gamma}_i^{\mathrm{pre}}\}_{i\in\mathcal{I}}=\varnothing$，$\{\boldsymbol{\gamma}_i^{\mathrm{cur}}\}_{i\in\mathcal{I}}=\varnothing$。
2. 更新$\{\boldsymbol{\gamma}_i^{\mathrm{pre}}\}_{i\in\mathcal{I}}=\{\boldsymbol{\gamma}_i^{\mathrm{ini}}\}_{i\in\mathcal{I}}$。
3. While $\{\boldsymbol{\gamma}_i^{\mathrm{cur}}\}_{i\in\mathcal{I}}\neq\{\boldsymbol{\gamma}_i^{\mathrm{pre}}\}_{i\in\mathcal{I}}$ do
4. 　给定$\{\boldsymbol{\gamma}_i^{\mathrm{pre}}\}_{i\in\mathcal{I}}$时，根据 3.1 节提出算法 5 得到当前的$\{\mathbf{x}_i\}_{i\in\mathcal{I}}$，$\{\mathbf{y}_i\}_{i\in\mathcal{I}}$。
5. 　根据步骤 4 中得到的$\{\mathbf{x}_i\}_{i\in\mathcal{I}}$，$\{\mathbf{y}_i\}_{i\in\mathcal{I}}$，利用 3.2 节提出的算法 6 得到当前的$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$，用$\{\boldsymbol{\gamma}_i^{\mathrm{cur}}\}_{i\in\mathcal{I}}$表示。
6. 　If $\{\boldsymbol{\gamma}_i^{\mathrm{cur}}\}_{i\in\mathcal{I}}=\{\boldsymbol{\gamma}_i^{\mathrm{pre}}\}_{i\in\mathcal{I}}$ then
7. 　　跳出循环，执行步骤 12。
8. 　Else
9. 　　更新$\{\boldsymbol{\gamma}_i^{\mathrm{pre}}\}_{i\in\mathcal{I}}=\{\boldsymbol{\gamma}_i^{\mathrm{cur}}\}_{i\in\mathcal{I}}$，执行步骤 4。
10. 　End if
11. End while
12. 输出问题 P1 的对应最优$\{\boldsymbol{\gamma}_i\}_{i\in\mathcal{I}}$，$\{\mathbf{x}_i\}_{i\in\mathcal{I}}$，$\{\mathbf{y}_i\}_{i\in\mathcal{I}}$。

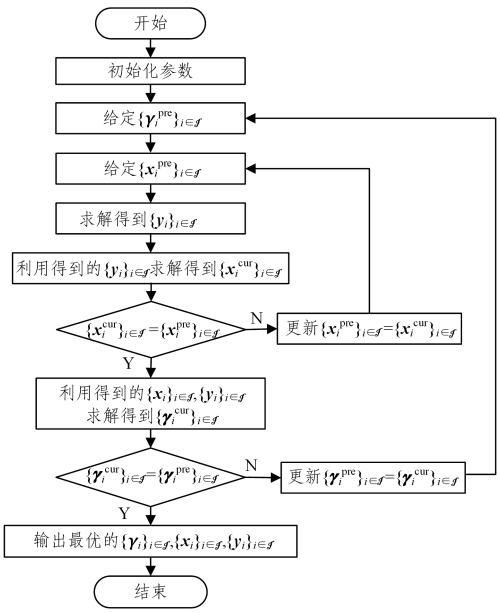

图 3　算法流程图

Fig. 3　Flow chart of proposed algorithm

4　算法仿真与结果分析

4.1　仿真环境与参数

为了验证本文算法的有效性以及性能，本节采用 MATLAB 进行仿真实验。根据文献[14,18]，设置仿真参数如下。

（1）区块链收益回报的固定项 $R=7000$ \$，给定的可变收益参数 $r=20$ \$/Mbit，$\theta=\dfrac{1}{600\,\mathrm{sec}}$，$z=5\times10^{-3}$。

（2）每个移动终端设备本地计算资源上限 $C_i^{\max}$ 均匀分布于[0,3]GHash/sec，本地计算资源的边际成本 π_i 均匀分布于[10,20] \$/GHash，生成块的大小 t_i 均匀分布于[0,1]Mbits。边缘服务器计算资源的边际成本 $p=40$ \$/GHash，边缘服务器计算资源上限 $C^{\mathrm{tot}}=30$ GHash/sec。

4.2　仿真结果与分析

4.2.1　算法的有效性以及性能优势

本节首先对提出算法的有效性进行分析。图 4、图 5 和

表 1 是在包含 3 个移动终端设备的区块链场景下得到的实验结果。

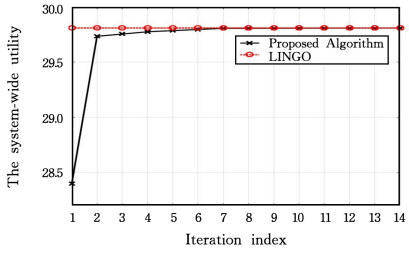

图 4 系统效用的迭代过程

Fig. 4 Convergence of system-wide utility

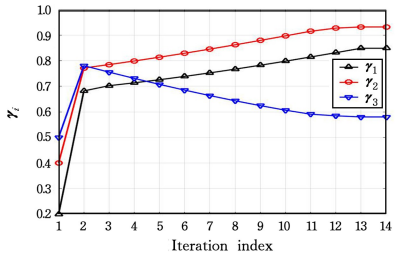

图 5 收益分享变量$\{\gamma_i\}_{i\in\mathcal{I}}$的迭代过程

Fig. 5 Convergence of all MTs' revenue-sharing variable $\{\gamma_i\}_{i\in\mathcal{I}}$

表 1 计算资源优化结果

Table 1 Optimization results of MTs' computation resources

	MT 1	MT 2	MT 3
x_i	0.19	0.10	0.49
y_i	0.76	0.72	0.32

图 4 和图 5 给出了本文提所算法的收敛过程。同时，将迭代结果与 LINGO 软件求解问题 P1 获得的全局最优解进行比较(LINGO 软件采用分支定界算法来解决优化问题，并渐近地逼近全局最优解，然而使用 LINGO 的缺点是运算时间非常长)。从图 4 和图 5 中可以看出，本文提出的算法可以快速收敛到最优解。表 1 显示了当前场景下，移动终端设备的计算资源优化结果。可以看到，当移动终端设备的本地计算资源不足时，会从边缘服务器获取更多额外的计算资源。

为了进一步体现本文所提算法的优越性，图 6 给出了在不同移动终端设备数量下，本文算法求解与直接用 LINGO 求解的系统效用，图 7 对比了两者的运行时间。从图 6 中可以看到，随着不同场景下系统中移动终端设备数量的增多，系统效用也在相应增长。同时，图 6 和图 7 的结果表明，本文提出的算法可以得到与 LINGO 几乎相同的计算结果，但是平均运行时间节约了约 82.99%。

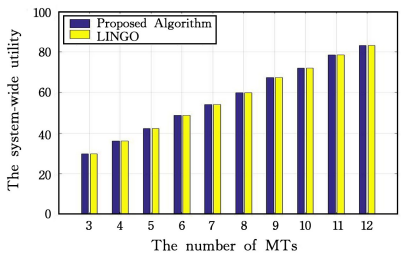

图 6 本文算法与 LINGO 求解的结果比较

Fig. 6 Results comparison between the proposed algorithm and LINGO

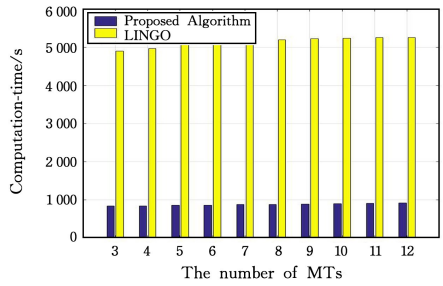

图 7 本文算法与 LINGO 求解的运行时间比较

Fig. 7 Computation-time comparison between the proposed algorithm and LINGO

根据图 6 和图 7 中不同数量下终端设备的参数设置，图 8 在上述场景中将$\{\gamma_i\}_{i\in\mathcal{I}}$设置为常量，并从 0.2 变化到 0.8，然后分别求解得到对应的系统效用。同时将本文提出的算法求解得到的系统效用作为对比。可以看到，本文提出的算法得到的系统效用远远优于给定$\{\gamma_i\}_{i\in\mathcal{I}}$下的得到系统效用。

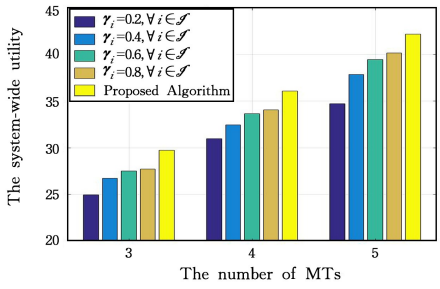

图 8 不同$\{\gamma_i\}_{i\in\mathcal{I}}$下的系统效用

Fig. 8 System-wide utility with different given $\{\gamma_i\}_{i\in\mathcal{I}}$

最后，为了体现本文算法的性能优势，选取了图 8 中 5 个移动终端设备的场景，分别设置 R 从 7000 变换到 10000，图 9 给出了两种计算资源分配机制下的系统效用：1)$\boldsymbol{x}_i=0$，$\forall i\in\mathcal{I}$。移动终端设备不使用本地计算资源，全部使用边缘服务器计算资源来执行工作量证明任务。2)$\boldsymbol{x}_i=C_i^{\max}$，$\forall i\in\mathcal{I}$。移动设备在获取边缘服务器计算资源的同时将本地计算资源全部用于执行任务。同时，将本文提出的算法得到的系统效用作为对比。从图 9 中可以看出，本文提出的算法通过优化移动终端设备以及边缘服务器的计算资源分配，可以有效提高系统效用。

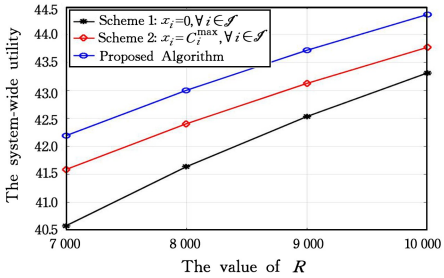

图 9 不同计算资源分配机制下的系统效用

Fig. 9 System-wide utility with different computation resource allocation scheme

4.2.2 场景动态仿真

本节模拟了移动终端设备动态加入区块链系统的场景：

初始时刻系统内存在 3 个移动终端设备，随后其他移动终端设备逐渐加入。移动终端设备的参数给定之后，不再发生变化。图 10 和图 11 给出了不同时刻下的系统效用和相应的区块链系统内用于执行工作量证明任务的计算资源的变化趋势。同时，为了展示不同区块链系统的收益系数对区块链系统的影响，设置了 $R=7000$ 以及 $R=10000$ 两个场景。

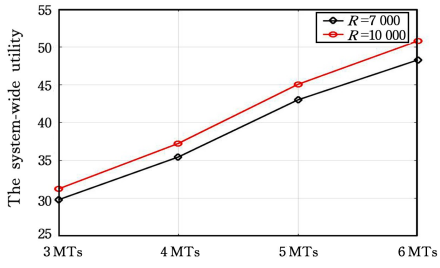

图 10　移动终端设备动态加入时系统效用的变化

Fig. 10　Change of system-wide utility when MTs add dynamically

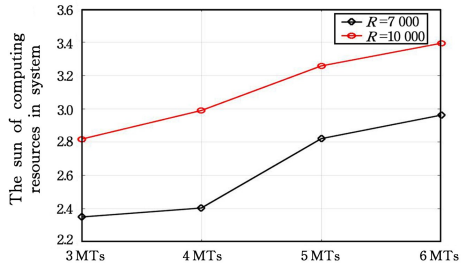

图 11　移动终端设备动态加入时系统计算资源的变化

Fig. 11　Change of computation resources in system when MTs add dynamically

通过图 10 和图 11 可以看到，当区块链系统中移动终端设备逐渐加入时，系统总效用相应增长，并且整个系统内用于执行工作量证明任务的计算资源量也相应增加。通过对区块链奖励系数的改变，可以发现当奖励系数升高时，会激励移动终端设备使用更多的计算资源来执行工作量证明任务。

结束语　本文研究了移动区块链系统中的计算资源分配以及收益分享问题。针对移动终端设备从边缘服务器获取计算资源并合理分配自身计算资源来执行工作量证明任务的过程，提出了一个最大化所有移动终端设备和边缘服务器的系统效用的联合优化模型。虽然联合优化问题难以直接求解，但本文利用其隐含的凸性，提出了有效的算法对其进行多层分解。实验结果验证了本文所提算法的有效性和性能优势。未来的工作将进一步引入多个边缘服务器，考虑移动终端设备可以自由加入不同的区块链系统执行任务的场景。

参 考 文 献

[1] ZHU X D, ZHANG Y Y, YAO R K, et al. Research on Government Information Opening and Sharing Model and Application Based on Blockchain[J]. Journal of Chongqing Technology and Business University(Natural Science Edition), 2020, 37(5): 122-128.

[2] XU C J, LI X F. Data Privacy Protection Method of Block Chain Transanction[J]. Computer Science, 2020, 47(3): 281-286.

[3] XIE J, YU F R, HUANG T, et al. A Survey on the Scalability of Blockchain Systems[J]. IEEE Network, 2019, 33(5): 166-173.

[4] XIONG L, Li F G, LIU Z C. Conditional Privacy-preserving Authentication Scheme Based on Blockchain for Vehicular Ad Hoc Networks[J]. Computer Science, 2020, 47(11): 55-59.

[5] JIANG Y N, GE X H, YANG Y, et al. 6G oriented blockchain based Internet of things data sharing and storage mechanism [J]. Journal on Communications, 2020(10): 48-55.

[6] RUINIAN L, TIANYI S, BO M, et al. Blockchain For Large-Scale Internet of Things Data Storage and Protection[J]. IEEE Transactions on Services Computing, 2018, 2(5): 762-771.

[7] KANG J, YU R, HUANG X, et al. Blockchain for Secure and Efficient Data Sharing in Vehicular Edge Computing and Networks[J]. IEEE Internet of Things Journal, 2019, 6(3): 4660-4670.

[8] HU Z P, DING W P, GAO Z, et al. Multi-stage Cascade Wireless Security Authentication Scheme Based on Blockchain Technology[J]. Computer Science, 2019, 46(12): 180-185.

[9] WU J G, LIU T L, LI J Y, et al. Research Progress on Blockchain Technology in Mobile Edge Computing[J]. Computer Engineering, 2020, 46(8): 1-13.

[10] ZEHUI X, YANG Z, DUSIT N, et al. When Mobile Blockchain Meets Edge Computing[J]. IEEE Communications Magazine, 2017, 56(8): 33-39.

[11] LIANG H J, HAN J T. Research on Decentralized Transaction Consensus Mechanism of Cloud Computing Resources Based on Block Chain[J]. Computer Science, 2019, 46(S2): 548-552.

[12] ZHANG P, LI S L, LIU Y M, et al. Resource management in blockchain-enabled heterogeneous edge computing system[J]. Journal on Communications, 2020(10): 1-14.

[13] YANG R, YU F R, SI P, et al. Integrated Blockchain and Edge Computing Systems: A Survey, Some Research Issues and Challenges[J]. IEEE Communications Surveys & Tutorials, 2019, PP(2): 1.

[14] XIONG Z, FENG S, WANG W, et al. Cloud/fog computing resource management and pricing for blockchain networks[J]. IEEE Internet of Things Journal, 2019, 6(3): 4585-4600.

[15] ZHAO N, WU H, CHEN Y. Coalition Game-Based Computation Resource Allocation for Wireless Blockchain Networks[J]. IEEE Internet of Things Journal, 2019, PP(99): 1.

[16] CHEN W, ZHANG Z, HONG Z, et al. Cooperative and Distributed Computation Offloading for Blockchain-Empowered Industrial Internet of Things[J]. IEEE Internet of Things Journal, 2019, 6(5): 8433-8446.

[17] DAI J J, SHEN S B. A Method of Network Edge Resourcc Allocation Based on Blockchain[J]. Computer Engineering, 2020, 46(8): 35-42.

[18] WU Y, XU X, QIAN L, et al. Revenue-Sharing based Computation-Resource Allocation for Mobile Blockchain[C]// IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2020: 56-61.

[19] WU Y, KIM H, HANDE P H, et al. Revenue sharing among ISPs in two-sided markets[C]// 2011 Proceedings IEEE INFOCOM. Shanghai, 2011: 596-600.

[20] CAO Z, ZHANG H, LIU B, et al. A Game-theoretic Framework for Revenue Sharing in Edge-Cloud Computing System[C]// 2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC). IEEE, 2019: 1-8.

[21] DIAKONIKOLAS J. Block Coordinate Descent and Exact Minimization[EB/OL]. http://people. csail. mit. edu/joanne/WOLA18-slides/Diakonikolas-Jelena.

[22] SCHRAGE L. Optimization Modeling With LINGO[M]. Lindo Systems, Inc., Chicago, 2006.

[23] LIN X, SHROFF N B, SRIKANT R. A tutorial on cross-layer optimization in wireless networks[J]. IEEE Journal on Selected areas in Communications, 2006, 24(8): 1452-1463.

[24] LOW S H, LAPSLEY D E. Optimization flow control. I. Basic algorithm and convergence[J]. IEEE/ACM Transactions on networking, 1999, 7(6): 861-874.

[25] BOYD S, VANDENBERGHE L. Convex Optimization[J]. IEEE Transactions on Automatic Control, 2006, 51(11): 1859.

XU Xu, born in 1996, postgraduate. His main research interests include blockchain and mobile edge computing.

QIAN Li-ping, born in 1981, Ph.D, professor, Ph.D supervisor, is a member of China Computer Federation. Her main research interests include wireless communication and networking, IoT and vehicle network.