

文章编号: 1005-0523(2024)01-0070-08



UDN中面向安全卸载的多目标联合优化算法研究

周天清, 王博博

(华东交通大学信息工程学院, 江西 南昌 330013)

摘要:【目的】随着各类移动应用与服务的迅猛增加, 移动终端电池容量同能量消耗之间的矛盾日益突出。此外, 超密集网络(UDN)中小基站的超密集部署使得网络干扰变得更为复杂, 且位于网络边缘的服务器易遭恶意攻击。【方法】针对多任务UDN, 通过联合优化用户设备(UE)关联、密码服务分派、UE功率控制、UE与小基站的计算资源分配以最小化标准总能耗与标准总安全成本之权和。首先, 为UDN构建MEC和本地计算的系统模型; 然后, 构建上述权和最小化问题并为其设计了进一步改进的分层自适应搜索(FIHAS)算法。【结果】仿真中, FIHAS能够获得较其他算法更低的权和, 且在降低总成本方面更具优势。【结论】总体上, FIHAS能够获得较其他算法更优的系统性能。

关键词: 超密集网络; 安全卸载; 遗传算法; 粒子群算法

中图分类号: TN929.5; U495

文献标志码: A

本文引用格式: 周天清, 王博博. UDN中面向安全卸载的多目标联合优化算法研究[J]. 华东交通大学学报, 2024, 41(1): 70-77.

Research on Multi-Objective Joint Optimization Algorithm for Secure Offloading in UDN

Zhou Tianqing, Wang Bobo

(School of Information Engineering, East China Jiaotong University, Nanchang 330013, China)

Abstract: 【Objective】With the rapid growth of various mobile applications and services, the contradiction between the battery capacity and the energy consumption at mobile terminals is becoming increasingly prominent. In addition, the ultra-dense deployment of small base stations (SBSs) in ultra-dense network (UDN) makes network interference more complicated, and servers deployed at the edge of the network are also vulnerable to malicious attacks. 【Method】By jointly optimizing the user equipment (UE) association, cryptographic service assignment, UE power control, and computational resource allocation of UEs and SBSs, the sum of weighted standardized total energy consumption and standardized total security cost is minimized for the multi-task UDN. Specifically, mobile edge computing (MEC) and local computing models are first built for multi-task UDN. Then, a further improved hierarchical adaptive search (FIHAS) algorithm is designed for a problem with minimizing the sum of weighted normalized total energy consumption and normalized total safety cost. 【Result】In the simulation, FIHAS can obtain lower weighted sum than other algorithms and has an advantage in reducing total cost. 【Conclusion】In general, FIHAS may achieve the better system performance than other algorithms.

Key words: ultra-dense networks; secure offloading; genetic algorithm; particle swarm optimization

收稿日期: 2023-04-29

基金项目: 国家自然科学基金项目(62261020, 62062034); 江西省自然科学基金项目(20232ACB212005, 20224BAB202001); 江西省研究生创新专项资金项目(YC2022-s549)

Citation format: ZHOU T Q, WANG B B. Research on multi-objective joint optimization algorithm for secure offloading in UDN[J]. Journal of East China Jiaotong University, 2024, 41(1): 70-77.

【研究意义】在移动边缘计算(mobile edge computing, MEC)辅助的超密集网络(ultra-dense networks, UDN)中,如何在有限通信资源限制下实现低能耗卸载及安全卸载是亟待解决的问题^[1]。

【研究进展】近年来,为降低通信能耗,绿色的计算卸载策略得到了广泛且深入的研究。Guo等^[2]联合优化UE(user equipment, UE)功率、计算卸载决策和资源分配以最小化超密集小小区网络中UE能耗,并提出了基于遗传算法(genetic algorithm, GA)和粒子群优化(particle swarm optimization, PSO)算法的分层搜索算法。Guo等^[3]针对超密集网络(ultra-dense networks, UDN),在每一类计算任务的处理时间约束下,提出了一种多任务启发式的贪婪卸载算法以最小化总能耗。Lu等^[4]针对UDN,在计算资源、信道资源和电池容量限制下,联合优化任务卸载、信道选择和资源调度以最小化系统成本,并开发了基于内点法和GA的问题求解算法。Zhou等^[5]针对超密集多任务物联网,在比例计算资源分配和UE时延约束下,联合优化计算卸载、设备关联和资源分配以最小化网络系统能耗,并提出了基于自适应保护多样性的遗传算法(adaptive diversity-guided genetic algorithm, ADGGA)和自适应粒子群优化(adaptive particle swarm optimization, APSO)算法的改进的分层自适应搜索算法(improved hierarchical adaptive search, IHAS)。

不难发现,上述工作仅涉及能耗优化,实现了绿色通信的目标,但忽视了因卸载而带来的安全隐患。为确保安全卸载,目前一些研究已开始考虑在计算卸载方案中引入安全保护措施。Zahed等^[6]联合优化任务卸载、安全服务分配和缓存以最小化物联网系统的总成本,其中成本为量化后的能耗和安全漏洞成本的组合。Elgandy等^[7]提出了一种多用户多任务计算卸载模型,来保护传输的数据免受网络攻击。

【创新特色】虽然上述卸载方案引入了安全保护措施,但鲜有研究关注多任务UDN。不同于现有研究,本文针对多任务UDN,在时延、功率与计算资源等约束条件下,联合优化UE关联、密码服务分

派、UE功率控制、UE与小基站(small base station, SBS)计算资源分配以最小化加权的标准化总能耗与标准化总安全成本之和。

【关键问题】鉴于所构建的优化问题具备非线性和混合整数的形式,采用传统凸优化手段很难获取其最优解。为处理该问题,本文设计了进一步改进的分层自适应搜索(further improved hierarchical adaptive search, FIHAS)算法。该算法首先利用改进的ADGGA进行粗粒度搜索,再利用APSO算法进行细粒度搜索。在改进的ADGGA中,自适应交叉和变异概率的权重采用了动态权重而非文献[5]的静态权重。

1 系统模型

1.1 网络模型

本文关注由 K 个多任务UE, 1个宏基站(macro base station, MBS)与 N 个SBS组成的控制面(C-Plane)与UE面(U-Plane)分离的UDN。在该网络中,每个SBS装备了一个有限计算容量的MEC服务器, MBS负责网络资源管理与控制, SBS负责数据通信。换言之, UE仅能将计算任务卸载至SBS执行。此外, UDN中SBS数大于等于UE数; UE索引集为 $\mathcal{K}=\{1, 2, \dots, K\}$; SBS基站索引集为 $\mathcal{N}=\{1, 2, \dots, N\}$; 每个UE的任务索引集为 $\mathcal{I}=\{1, 2, \dots, I\}$; 任意SBS n 的带宽为 W_n , 它被均等地分配给其所服务的UE任务。

1.2 通信模型

在等带宽分配原则下, UE k 的任务 i 上传至SBS n 的上行数据速率为

$$r_{n,k,i} = \frac{W_n}{\sum_{m \in \mathcal{K}} \sum_{j \in \mathcal{I}} x_{n,m,j}} \log_2 \left(1 + \frac{p_k g_{n,k}}{\sigma^2} \right) \quad (1)$$

式中: $x_{n,k,i}$ 表示UE k 的任务 i 是否卸载至SBS n ; $x_{n,k,i}=1$ 表示UE k 的任务 i 卸载至SBS n ; $x_{n,k,i}=0$ 表示UE k 的任务 i 不卸载至SBS n ; $\sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} x_{n,k,i}$ 为关联SBS n 的总任务数, 任意UE的任意任务至多只能选择一个SBS; p_k 为UE k 的发射功率; σ^2 为噪声功率; $g_{n,k}$ 为UE k 与SBS n 间的信道增益。

1.3 计算模型

对于任意UE, 其计算任务 i 由5元组表示, 即

$\{d_i, \omega_i, \tau_i^{\max}, \rho_i, \zeta_i\}$ 。其中, d_i 为任务 i 的数据大小, ω_i 为完成任务 i 所需 CPU 周期数(cycles), τ_i^{\max} 为任务 i 的最大执行时延, ρ_i 为任务 i 的预期安全级别, ζ_i 为任务 i 的安全风险系数。

1) 本地计算。当 UE^k 的任务 i 由自身完成时, 其本地执行时间(时延)为

$$\tau_{k,i}^{\text{loc}} = \omega_i / f_{k,i}^{\text{loc}} \left(1 - \sum_{n \in \mathcal{N}} x_{n,k,i}\right) \quad (2)$$

相应地, 其能耗为

$$\varepsilon_{k,i}^{\text{loc}} = \alpha \omega_i (f_{k,i}^{\text{loc}})^2 \left(1 - \sum_{n \in \mathcal{N}} x_{n,k,i}\right) \quad (3)$$

式中: α 为 UE 的 CPU 能量系数; $f_{k,i}^{\text{loc}}$ 为 UE^k 分配给任务 i 的计算资源量。

2) 边缘计算。当 UE^k 的任务 i 卸载至某 SBS 时, 其上行传输时间为 $\tau_{k,i}^{\text{up}} = \sum_{n \in \mathcal{N}} x_{n,k,i} d_i / r_{n,k,i}$ 。相应地, 其上行传输能耗为

$$\varepsilon_{k,i}^{\text{up}} = p_k \tau_{k,i}^{\text{up}} = \sum_{n \in \mathcal{N}} x_{n,k,i} p_k d_i / r_{n,k,i} \quad (4)$$

此外, UE^k 的任务 i 在 SBS 上的计算时间为

$$\tau_{k,i}^{\text{mccc}} = \sum_{n \in \mathcal{N}} x_{n,k,i} \omega_i / f_{n,k,i}^{\text{mccc}} \quad (5)$$

相应地, 其能耗为 $\varepsilon_{k,i}^{\text{mccc}} = \beta \sum_{n \in \mathcal{N}} \omega_i x_{n,k,i} (f_{n,k,i}^{\text{mccc}})^2$ 。

式中: $f_{n,k,i}^{\text{mccc}}$ 为 SBSⁿ 分配给 UE^k 的任务 i 的计算资源量; β 为边缘服务器的 CPU 能量系数。

1.4 安全模型

假设 L 个密码算法的索引集 $\mathcal{L} = \{1, 2, \dots, L\}$ 及它们安全级别的集合 $\mathcal{V} = \{v_1, v_2, \dots, v_L\}$ 。其中, $v_l = l$ 表示密码算法 l 的安全级别^[4]。对于密码算法 l , 其加密、解密单位比特数据所需计算资源量分别为 δ_l (CPU cycles/bit) 和 $\bar{\delta}_l$ (CPU cycles/bit), 且加密、解密单位比特数据的能耗为 $\hat{\delta}_l$ (10^{-7} J/bit)^[8]。

当任务 i 使用密码算法 l 保护任务时, 其失败概率为 $\bar{P}_{i,l} = 1 - e^{-\zeta(\rho_i - v_l)}$ [9]。

UE^k 的任务 i 在本地加密的时延为

$$\tau_{k,i}^{\text{loc}} = \sum_{n \in \mathcal{N}} x_{n,k,i} \sum_{l \in \mathcal{L}} y_{i,l} \delta_l d_i / f_{k,i}^{\text{loc}} \quad (6)$$

其加密能耗为 $\varepsilon_{k,i}^{\text{loc}} = \sum_{n \in \mathcal{N}} x_{n,k,i} \sum_{l \in \mathcal{L}} y_{i,l} \hat{\delta}_l d_i$, 此外 UE^k 的任务 i 在 SBS 的解密时延为

$$\tau_{k,i}^{\text{mccc}} = \sum_{n \in \mathcal{N}} x_{n,k,i} \sum_{l \in \mathcal{L}} y_{i,l} \bar{\delta}_l d_i / f_{n,k,i}^{\text{mccc}} \quad (7)$$

其解密能耗为 $\varepsilon_{k,i}^{\text{mccc}} = \sum_{n \in \mathcal{N}} x_{n,k,i} \sum_{l \in \mathcal{L}} y_{i,l} \hat{\delta}_l d_i$ 式中: $y_{i,l}$ 指示任务 i 是否选择密码算法 l ; $y_{i,l} = 1$ 表示任务 i 选择密码算法 l ; $y_{i,l} = 0$ 表示任务 i 不选择密码算法 l 。

于是, 所有 UE 所有任务的安全成本, 即 UE 总安全成本为

$$\psi = \sum_{n \in \mathcal{N}} \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} x_{n,k,i} \sum_{l \in \mathcal{L}} \lambda_l y_{i,l} \bar{P}_{i,l} \quad (8)$$

式中: λ_l 为任务 i 因安全保护失败而产生费用。

2 问题建模

UE^k 的任务 i 的处理时延(时间)为

$$\tau_{k,i} = \tau_{k,i}^{\text{loc}} + \tau_{k,i}^{\text{up}} + \tau_{k,i}^{\text{mccc}} + \tau_{k,i}^{\text{loc}} + \tau_{k,i}^{\text{mccc}} \quad (9)$$

此外, UE 总能耗为

$$E^{\text{UE}} = \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} (\varepsilon_{k,i}^{\text{loc}} + \varepsilon_{k,i}^{\text{up}} + \varepsilon_{k,i}^{\text{mccc}}) \quad (10)$$

UE 总能耗上界为

$$E^{\text{UE-max}} = \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} \max \left\{ \alpha \omega_i (f_k^{\text{loc-max}})^2, p_k^{\text{max}} d_i / \bar{r}_{n,k,i} + d_i \bar{\delta}^{\text{max}} \right\} \quad (11)$$

式中:

$$n_{k,i} = \arg \max_{n \in \mathcal{N}} \left\{ p_k^{\text{max}} d_i / \bar{r}_{n,k,i} \right\} \quad (12)$$

$$\bar{r}_{n,k} = W_n K^{-1} \log_2 \left(1 + p_k^{\text{max}} g_{n,k} / \sigma^2 \right) \quad (13)$$

在任务时延、UE 功率、UE 计算资源与 SBS 计算资源的约束下, 本文联合优化任务关联矩阵 $\mathbf{X} = \{x_{n,k,i}, \forall n \in \mathcal{N}, \forall k \in \mathcal{K}, \forall i \in \mathcal{I}\}$, 安全密码算法选择矩阵 $\mathbf{Y} = \{y_{i,l}, \forall i \in \mathcal{I}, \forall l \in \mathcal{L}\}$, UE 计算资源的分配矩阵 $\mathbf{F}^{\text{loc}} = \{f_{k,i}^{\text{loc}}, \forall k \in \mathcal{K}, \forall i \in \mathcal{I}\}$, UE 发射功率集 $\mathbf{p} = \{p_k, \forall k \in \mathcal{K}\}$ 与 SBS 计算资源的分配矩阵 $\mathbf{F}^{\text{mccc}} = \{f_{n,k,i}^{\text{mccc}}, \forall n \in \mathcal{N}, \forall k \in \mathcal{K}, \forall i \in \mathcal{I}\}$ 以最小化建加权的标准化总能耗与标准化总安全成本之和。具体问题如下

$$\begin{aligned} \min_{\mathbf{X}, \mathbf{Y}, \mathbf{F}^{\text{loc}}, \mathbf{F}^{\text{mccc}}, \mathbf{p}} \quad & \zeta E^{\text{UE}} / E^{\text{UE-max}} + (1 - \zeta) \psi / K \sum_{i \in \mathcal{I}} \lambda_i \\ \text{s.t.} \quad & C_1: \tau_{k,i} \leq \tau_i^{\text{max}}, \forall k \in \mathcal{K}, \forall i \in \mathcal{I}, \\ & C_2: \sum_{n \in \mathcal{N}} x_{n,k,i} \leq 1, \forall k \in \mathcal{K}, \forall i \in \mathcal{I}, \\ & C_3: \sum_{l \in \mathcal{L}} y_{i,l} = 1, \forall i \in \mathcal{I}, \\ & C_4: \theta \leq p_k \leq p_k^{\text{max}}, \forall k \in \mathcal{K}, \\ & C_5: \sum_{i \in \mathcal{I}} f_{k,i}^{\text{loc}} \left(1 - \sum_{n \in \mathcal{N}} x_{n,k,i}\right) \leq f_k^{\text{loc-max}}, \\ & C_6: \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} f_{n,k,i}^{\text{mccc}} x_{n,k,i} \leq f_n^{\text{mccc-max}}, \\ & C_7: y_{i,l} \in \{0, 1\}, \forall i \in \mathcal{I}, \forall l \in \mathcal{L}, \\ & C_8: x_{n,k,i} \in \{0, 1\}, \forall n \in \mathcal{N}, \forall k \in \mathcal{K}, \forall i \in \mathcal{I}, \end{aligned} \quad (14)$$

式中: ζ 为用于调整标准能耗与标准安全成本的权重; C_1 表示 UE^k 的任务 i 的处理时延不能超过该任务的最后期限; C_2 表示 UE^k 的任务 i 至多关联一个 SBS; C_3 表示任务 i 只能选择一个密码算法; C_4 表示 UE^k 的发射功率不能低于 θ 且不能高于其最大发射功率 p_k^{max} , θ 取足够小的常量值以避免“0/0”的现象; C_5 表示 UE^k 分配给自身所有任务的计算资源不

能超过其最大计算资源量 $f_k^{\text{loc-max}}$; C_6 表示 SBS n 分配给所关联 UE 任务的计算资源不能超过它的最大计算资源量 $f_n^{\text{mec-max}}$ 。

3 算法设计

容易发现,问题(14)为混合整数、非线性优化问题,且其优化变量耦合。为处理该类复杂问题,本文结合 GA 与 PSO 算法设计该问题的求解算法 FIHAS,它为 IHAS 的改进版。在 FIHAS 中,改进后的 ADGGA 用于粗粒度搜索,APSO 算法用于细粒度搜索。不同于 IHAS 中 ADGGA 为自适应交叉和变异概率设置静态权重,本文设计的 FIHAS 为其设置动态权重。

3.1 改进的 ADGGA

改进的 ADGGA 具体步骤如下。

1) 染色体。首先,定义种群,即 Z 个个体的集合 \mathcal{Z} ,然后将 X 编码成染色体 $\bar{A}=\{\bar{a}_{z,u}, \forall z \in \mathcal{Z}, \forall u \in \bar{\mathcal{K}}\}$,将 Y 编码成染色体 $\bar{B}=\{\bar{b}_{z,u}, \forall z \in \mathcal{Z}, \forall u \in \mathcal{I}\}$,将 F^{loc} 编码成染色体 $\bar{C}=\{\bar{c}_{z,u}, \forall z \in \mathcal{Z}, \forall u \in \bar{\mathcal{K}}\}$,将 F^{mec} 编码成染色体 $\bar{D}=\{\bar{d}_{z,u}, \forall z \in \mathcal{Z}, \forall u \in \bar{\mathcal{K}}\}$,将 p 编码成染色体 $\bar{Q}=\{\bar{q}_{z,u}, \forall z \in \mathcal{Z}, \forall u \in \mathcal{K}\}$,其中 $\bar{\mathcal{K}}=\{1, 2, \dots, KI\}$ 为由所有 UE 的所有任务构成的虚拟 UE 的索引集, $\bar{a}_{z,u}$, $\bar{b}_{z,u}$, $\bar{c}_{z,u}$, $\bar{d}_{z,u}$ 与 $\bar{q}_{z,u}$ 的取值分别表示个体 z 中 UE u 所关联 SBS 的索引号, UE u 所选择密码算法的索引号, UE u 在本地的计算资源分配量, UE u 在 SBS 的计算资源分配量及 UE u 的发射功率。

2) 初始化种群。对于任意个体 z 中的任意 UE u ,按如下规则初始化: $\bar{a}_{z,u}^0 = \text{randi}(\mathcal{N} \cup \{0\})$, $\bar{b}_{z,u}^0 = \text{randi}(\mathcal{L})$, $\bar{c}_{z,u}^0 = \text{rand}(f_k^{\text{loc-max}})$, $\bar{a}_{z,u}^0 \neq 0$ 时 $\bar{d}_{z,u}^0 = \text{rand}(f_n^{\text{mec-max}})$, $\bar{a}_{z,u}^0 = 0$ 时 $\bar{d}_{z,u}^0 = \theta$, $\bar{q}_{z,u}^0 = \text{rand}(p_u^{\text{max}})$, 其中 $\bar{a}_{z,u}^0$, $\bar{b}_{z,u}^0$, $\bar{c}_{z,u}^0$, $\bar{d}_{z,u}^0$ 与 $\bar{q}_{z,u}^0$ 分别表示 $\bar{a}_{z,u}$, $\bar{b}_{z,u}$, $\bar{c}_{z,u}$, $\bar{d}_{z,u}$ 与 $\bar{q}_{z,u}$ 的初始值。 $[k, i] = \text{ind 2sub}(KI, u)$ 可用于寻找虚拟 UE u 所对应的真实 UE 的索引号 k 与任务索引号 i , $\text{ind 2sub}()$ 把 $K \times I$ 数组或者矩阵的线性索引转化为相应的下标, $\text{randi}(\mathcal{S})$ 从任意集合 \mathcal{S} 中随机输出一个元素, $\text{rand}(v)$ 从区间 $(0, v)$ 随机输出一个数。

3) 适应度函数。在 GA 中,适应度函数主要用于评估个体的适应度。不难发现,式(14)的约束 C_1 、 C_5 和 C_6 为混合整数且耦合的形式,在遗传操作和 GA 初始化中难以满足。鉴于此,将此类约束作为惩罚项引入适应度函数。因此,任意个体 z 的适应

度函数可定义为

$$\begin{aligned} \mathcal{G}_z &= G(\bar{A}_z, \bar{B}_z, \bar{C}_z, \bar{D}_z, \bar{Q}_z) \\ &= \zeta E/E^{\text{max}} + (1 - \zeta)\psi / K \sum_{i \in \mathcal{I}} \lambda_i \\ &\quad - \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} \eta_{k,i} \max(\tau_{k,i} - \tau_i^{\text{max}}, 0) \\ &\quad - \sum_{n \in \mathcal{N}} \bar{\eta}_n \max(\sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} f_{n,k,i}^{\text{mec}} x_{n,k,i} - f_n^{\text{mec-max}}, 0) \\ &\quad - \sum_{k \in \mathcal{K}} \hat{\eta}_k \max(\sum_{i \in \mathcal{I}} f_{k,i}^{\text{loc}} (1 - \sum_{n \in \mathcal{N}} x_{n,k,i}) - f_k^{\text{loc-max}}, 0) \end{aligned} \quad (15)$$

式中: \mathcal{G}_z 为个体 z 的适应度函数值; $\eta_{k,i}$ 为 UE k 的任务 i 的时延约束的惩罚因子; $\bar{\eta}_n$ 为 SBS n 的计算资源约束的惩罚因子; $\hat{\eta}_k$ 为 UE k 的计算资源约束的惩罚因子。

4) 选择。本文采用锦标赛方法选择个体形成新的种群。为进一步提升性能,历史最优个体总是保存在种群中。

5) 自适应交叉。对于任意相邻个体 z 和 $\hat{z} = z + 1$, 随机选择位置进行交叉操作,从交叉位置开始交换相应染色体片段产生两个新个体。其中,任意相邻个体 z 和 $\hat{z} = z + 1$ 交叉概率 $\hat{P}_{z,\hat{z}}$ 为

$$\hat{P}_{z,\hat{z}} = \begin{cases} (1 - \omega)(\bar{\mathcal{G}}_{z,\hat{z}} - \mathcal{G}^{\text{min}}) / (\mathcal{G}^{\text{ave}} - \mathcal{G}^{\text{min}}), & \bar{\mathcal{G}}_{z,\hat{z}} < \mathcal{G}^{\text{ave}} \\ b_1, & \bar{\mathcal{G}}_{z,\hat{z}} \geq \mathcal{G}^{\text{ave}} \end{cases} \quad (16)$$

其中, ω 为自适应权重且为

$$\omega = \omega^{\text{max}} - t_1(\omega^{\text{max}} - \omega^{\text{min}}) / T_1 \quad (17)$$

式中: $0 \leq \omega^{\text{min}} < \omega^{\text{max}} \leq 1$ 为常数; $\bar{\mathcal{G}}_{z,\hat{z}}$ 为个体 z 和 $\hat{z} = z + 1$ 中适应度较小个体的适应度值; \mathcal{G}^{min} 和 \mathcal{G}^{ave} 分别为种群的最小适应度值和平均适应度值; b_1 是取值于区间 $(0, 1)$ 的常量。

6) 自适应变异。对于任意个体 z , 其基因以以下概率进行变异操作。

$$\check{P}_z = \begin{cases} (1 - \omega)(\mathcal{G}^{\text{max}} - \mathcal{G}_z) / (\mathcal{G}^{\text{max}} - \mathcal{G}^{\text{ave}}), & \mathcal{G}_z < \mathcal{G}^{\text{ave}} \\ b_2, & \mathcal{G}_z \geq \mathcal{G}^{\text{ave}} \end{cases} \quad (18)$$

式中: \mathcal{G}^{max} 为种群的最大适应度值; b_2 是取值区间 $(0, 1)$ 的常量;

在变异概率 \check{P}_z 下,任意个体 z 的基因按如下规则执行变异操作为

$$\bar{a}_{z,u} = \begin{cases} \text{round}(\kappa_1(N+1) + (1 - \kappa_1)\bar{a}_{z,u}), & \kappa_2 > 0.5, \\ \text{round}((1 - \kappa_1)\bar{a}_{z,u}), & \kappa_2 \leq 0.5, \end{cases} \quad (19)$$

$$\bar{b}_{z,u} = \begin{cases} \text{round}(\kappa_1 L + (1 - \kappa_1)\bar{b}_{z,u}), & \kappa_2 > 0.5, \\ \text{round}(\kappa_1 + (1 - \kappa_1)\bar{b}_{z,u}), & \kappa_2 \leq 0.5, \end{cases} \quad (20)$$

$$\bar{c}_{z,u} = \begin{cases} \kappa_1 f_k^{\text{loc-max}} + (1 - \kappa_1)\bar{c}_{z,u}, & \kappa_2 > 0.5, \\ (1 - \kappa_1)\bar{c}_{z,u}, & \kappa_2 \leq 0.5, \end{cases} \quad (21)$$

$$\bar{d}_{z,u} = \begin{cases} \kappa_1 f_{\bar{a}_{z,u}}^{\text{mvec}} + (1 - \kappa_1) \bar{d}_{z,u}, & \bar{a}_{z,u} \neq 0, \kappa_2 > 0.5, \\ (1 - \kappa_1) \bar{d}_{z,u}, & \bar{a}_{z,u} \neq 0 \text{ 且 } \kappa_2 \leq 0.5, \\ \theta, & \bar{a}_{z,u} = 0, \end{cases} \quad (22)$$

$$\bar{q}_{z,u} = \begin{cases} \kappa_1 p_u^{\text{max}} + (1 - \kappa_1) \bar{q}_{z,u}, & \kappa_2 > 0.5, \\ (1 - \kappa_1) \bar{q}_{z,u}, & \kappa_2 \leq 0.5, \end{cases} \quad (23)$$

式中: $\text{round}(v)$ 对任意数 v 向下取整; κ_1 和 κ_2 为服从 0~1 均匀分布的随机数; 为保证种群的多样性, 从而避免 GA 早熟收敛, 在自适应变异和交叉操作之前引入一个多样性变异。对于 n 维数值问题, 多样性测度 M 定义为

$$\begin{aligned} M = 0.2 & \left((ZY_1)^{-1} \sum_{z \in Z} \sqrt{\sum_{u \in \bar{\mathcal{K}}} (\bar{a}_{z,u} - \bar{a}_u^{\text{ave}})^2} \right. \\ & + (ZY_2)^{-1} \sum_{z \in Z} \sqrt{\sum_{u \in \mathcal{D}} (\bar{b}_{z,u} - \bar{b}_u^{\text{ave}})^2} \\ & + (ZY_3)^{-1} \sum_{z \in Z} \sqrt{\sum_{u \in \bar{\mathcal{K}}} (\bar{c}_{z,u} - \bar{c}_u^{\text{ave}})^2} \\ & + (ZY_4)^{-1} \sum_{z \in Z} \sqrt{\sum_{u \in \bar{\mathcal{K}}} (\bar{d}_{z,u} - \bar{d}_u^{\text{ave}})^2} \\ & \left. + (ZY_5)^{-1} \sum_{z \in Z} \sqrt{\sum_{u \in \mathcal{K}} (\bar{q}_{z,u} - \bar{q}_u^{\text{ave}})^2} \right) \end{aligned} \quad (24)$$

式中: Y_1 和 Y_2 、 Y_3 、 Y_4 、 Y_5 分别为 \bar{A} 、 \bar{B} 、 \bar{C} 、 \bar{D} 与 \bar{Q} 的可行域对角线的长度; \bar{a}_u^{ave} 、 \bar{b}_u^{ave} 、 \bar{c}_u^{ave} 、 \bar{d}_u^{ave} 与 \bar{q}_u^{ave} 分别为 UEⁿ 在种群中所关联 SBS 的索引号的均值, UEⁿ 在种群中所选择密码算法的索引号的均值, UEⁿ 在种群中本地计算资源分配量的均值, UEⁿ 在种群中 SBS 的计算资源分配量的均值与 UEⁿ 在种群中发射功率的均值。

然后, 定义种群多样性引导的变异概率 \tilde{P} 为

$$\begin{cases} \tilde{P} = \hbar_1, M < m_1; \tilde{P} = \hbar_2, m_1 \leq M < m_2; \\ \tilde{P} = \hbar_3, \text{ 其它情况,} \end{cases} \quad (25)$$

式中: \hbar_1 、 \hbar_2 与 \hbar_3 是预先设置的概率; m_1 与 m_2 是阈值常量。

在变异概率 \tilde{P} 下, 任意个体 z 的基因按上述变异规则执行变异操作。

3.2 自适应 PSO

在利用 ADGGA 获得问题(14)的粗粒度解后, 本文再次利用 APSO 进行细粒度搜索以进一步优化问题细粒度解, 其具体步骤如下。

1) 初始化粒子位置与速度。首先, 对于任意粒子(个体) z , 假定它由 5 个子粒子组成, 并以 ADGGA 的输出分别初始化位置, 即 $\tilde{A}_z = \bar{A}_z$, $\tilde{B}_z = \bar{B}_z$, $\tilde{C}_z = \bar{C}_z$, $\tilde{D}_z = \bar{D}_z$ 和 $\tilde{Q}_z = \bar{Q}_z$, 其中 $\tilde{A}_z = \{\tilde{a}_{z,k}, \forall k \in \bar{\mathcal{K}}\}$, $\tilde{B}_z = \{\tilde{b}_{z,k}, \forall k \in \mathcal{D}\}$, $\tilde{C}_z = \{\tilde{c}_{z,k}, \forall k \in \bar{\mathcal{K}}\}$, $\tilde{D}_z = \{\tilde{d}_{z,k}, \forall k \in \bar{\mathcal{K}}\}$, $\tilde{Q}_z = \{\tilde{q}_{z,k}, \forall k \in \mathcal{K}\}$; 然后对于任意粒子 z 的 5 个子粒

子, 分别以 (0,1) 区间的随机数初始化其速度 \tilde{A}_z , \tilde{B}_z , \tilde{C}_z , \tilde{D}_z 和 \tilde{Q}_z , 其中 $\tilde{A}_z = \{\tilde{a}_{z,k}, \forall k \in \bar{\mathcal{K}}\}$, $\tilde{B}_z = \{\tilde{b}_{z,k}, \forall k \in \mathcal{D}\}$, $\tilde{C}_z = \{\tilde{c}_{z,k}, \forall k \in \bar{\mathcal{K}}\}$, $\tilde{D}_z = \{\tilde{d}_{z,k}, \forall k \in \bar{\mathcal{K}}\}$, $\tilde{Q}_z = \{\tilde{q}_{z,k}, \forall k \in \mathcal{K}\}$ 。

2) 初始化粒子的历史最优位置。对于任意粒子 z 的 5 个子粒子, 以 ADGGA 的输出分别初始化其历史最优位置, 即 $\tilde{A}_z = \bar{A}_z$, $\tilde{B}_z = \bar{B}_z$, $\tilde{C}_z = \bar{C}_z$, $\tilde{D}_z = \bar{D}_z$ 和 $\tilde{Q}_z = \bar{Q}_z$, 其中, $\tilde{A}_z = \{\tilde{a}_{z,k}, \forall k \in \bar{\mathcal{K}}\}$, $\tilde{B}_z = \{\tilde{b}_{z,k}, \forall k \in \mathcal{D}\}$, $\tilde{C}_z = \{\tilde{c}_{z,k}, \forall k \in \bar{\mathcal{K}}\}$, $\tilde{D}_z = \{\tilde{d}_{z,k}, \forall k \in \bar{\mathcal{K}}\}$ 与 $\tilde{Q}_z = \{\tilde{q}_{z,k}, \forall k \in \mathcal{K}\}$ 。

3) 初始化全局最优粒子的位置。以 ADGGA 的输出分别初始化全局最优粒子的 5 个子粒子的位置, 即 $\hat{A} = \bar{A}_z$ 、 $\hat{B} = \bar{B}_z$ 、 $\hat{C} = \bar{C}_z$ 、 $\hat{D} = \bar{D}_z$ 和 $\hat{Q} = \bar{Q}_z$, 其中, \bar{z} 为所有个体中历史最优位置中最优位置所对应的个体, 即为全局最优粒子, $\hat{A} = \{\hat{a}_{z,k}, \forall k \in \bar{\mathcal{K}}\}$, $\hat{B} = \{\hat{b}_{z,k}, \forall k \in \mathcal{D}\}$, $\hat{C} = \{\hat{c}_{z,k}, \forall k \in \bar{\mathcal{K}}\}$, $\hat{D} = \{\hat{d}_{z,k}, \forall k \in \bar{\mathcal{K}}\}$ 与 $\hat{Q} = \{\hat{q}_{z,k}, \forall k \in \mathcal{K}\}$ 。

4) 更新任意普通粒子 $z \in Z \setminus \{\bar{z}\}$ 的速度, 即分别更新其子粒子的速度 \tilde{A}_z 、 \tilde{B}_z 、 \tilde{C}_z 、 \tilde{D}_z 和 \tilde{Q}_z 为

$$\tilde{A}_z^{t+1} = \mu_z^t \tilde{A}_z^t + e_1 \gamma_z (\tilde{A}_z^t - \tilde{A}_z^{t-1}) + e_2 \bar{\gamma}_z (\hat{A}^t - \tilde{A}_z^t) \quad (26)$$

$$\tilde{B}_z^{t+1} = \mu_z^t \tilde{B}_z^t + e_1 \gamma_z (\tilde{B}_z^t - \tilde{B}_z^{t-1}) + e_2 \bar{\gamma}_z (\hat{B}^t - \tilde{B}_z^t) \quad (27)$$

$$\tilde{C}_z^{t+1} = \mu_z^t \tilde{C}_z^t + e_1 \gamma_z (\tilde{C}_z^t - \tilde{C}_z^{t-1}) + e_2 \bar{\gamma}_z (\hat{C}^t - \tilde{C}_z^t) \quad (28)$$

$$\tilde{D}_z^{t+1} = \mu_z^t \tilde{D}_z^t + e_1 \gamma_z (\tilde{D}_z^t - \tilde{D}_z^{t-1}) + e_2 \bar{\gamma}_z (\hat{D}^t - \tilde{D}_z^t) \quad (29)$$

$$\tilde{Q}_z^{t+1} = \mu_z^t \tilde{Q}_z^t + e_1 \gamma_z (\tilde{Q}_z^t - \tilde{Q}_z^{t-1}) + e_2 \bar{\gamma}_z (\hat{Q}^t - \tilde{Q}_z^t) \quad (30)$$

式中: e_1 与 e_2 为常量; γ_z 与 $\bar{\gamma}_z$ 是粒子 z 取值于 [0,1] 区间的随机数。

在式(26)~式(30)中, 任意普通粒子 $z \in Z \setminus \{\bar{z}\}$ 的惯性权重 μ_z^t 为

$$\mu_z^{t+1} = \mu^{\text{max}} - t_2 (\mu^{\text{max}} - \mu^{\text{min}}) / T_2 \quad (31)$$

式中: μ^{max} 和 μ^{min} 分别为最小和最大惯性权重。

更新粒子的速度后, 任意普通粒子 $z \in Z \setminus \{\bar{z}\}$ 的位置可以更新为

$$\begin{cases} \tilde{A}_z^{t+1} = \text{round}(\tilde{A}_z^t + \tilde{A}_z^{t+1}), \tilde{B}_z^{t+1} = \text{round}(\tilde{B}_z^t + \tilde{B}_z^{t+1}) \\ \tilde{C}_z^{t+1} = \tilde{C}_z^t + \tilde{C}_z^{t+1}, \tilde{D}_z^{t+1} = \tilde{D}_z^t + \tilde{D}_z^{t+1}, \tilde{Q}_z^{t+1} = \tilde{Q}_z^t + \tilde{Q}_z^{t+1} \end{cases} \quad (32)$$

那么, 全局最优粒子 \bar{z} 的速度可以更新为

$$\tilde{A}_z^{t+1} = -\tilde{A}_z^t + \tilde{A}_z^{t+1} + e_3 \tilde{A}_z^t + v^t (1 - 2v_z^t) \quad (33)$$

$$\tilde{B}_z^{t+1} = -\tilde{B}_z^t + \tilde{B}_z^{t+1} + e_3 \tilde{B}_z^t + v^t (1 - 2v_z^t) \quad (34)$$

$$\tilde{C}_z^{t+1} = -\tilde{C}_z^t + \tilde{C}_z^{t+1} + e_3 \tilde{C}_z^t + v^t (1 - 2v_z^t) \quad (35)$$

$$\vec{D}_z^{t_2+1} = -\vec{D}_z^{t_2} + \vec{D}^{t_2} + e_3 \vec{D}_z^{t_2} + v^{t_2}(1 - 2v_z^{t_2}) \quad (36)$$

$$\vec{Q}_z^{t_2+1} = -\vec{Q}_z^{t_2} + \vec{Q}^{t_2} + e_3 \vec{Q}_z^{t_2} + v^{t_2}(1 - 2\hat{v}_z^{t_2}) \quad (37)$$

然后,更新全局最优粒子 \vec{z} 的位置

$$\vec{A}_z^{t_2+1} = \text{round}(\vec{A}^{t_2} + e_3 \vec{A}_z^{t_2} + v^{t_2}(1 - 2v_z^{t_2})) \quad (38)$$

$$\vec{B}_z^{t_2+1} = \text{round}(\vec{B}^{t_2} + e_3 \vec{B}_z^{t_2} + v^{t_2}(1 - 2\hat{v}_z^{t_2})) \quad (39)$$

$$\vec{C}_z^{t_2+1} = \vec{C}^{t_2} + e_3 \vec{C}_z^{t_2} + v^{t_2}(1 - 2v_z^{t_2}) \quad (40)$$

$$\vec{D}_z^{t_2+1} = \vec{D}^{t_2} + e_3 \vec{D}_z^{t_2} + v^{t_2}(1 - 2v_z^{t_2}) \quad (41)$$

$$\vec{Q}_z^{t_2+1} = \vec{Q}^{t_2} + e_3 \vec{Q}_z^{t_2} + v^{t_2}(1 - 2\hat{v}_z^{t_2}) \quad (42)$$

式中: e_3 为常量; $v_z = \{v_{z,k}, \forall k \in \mathcal{K}\}$, $\hat{v}_z = \{\hat{v}_{z,k}, \forall k \in \mathcal{T}\}$, $\hat{v}_z = \{\hat{v}_{z,k}, \forall k \in \mathcal{K}\}$ 的元素来自于 $[0, 1]$ 区间的随机数。

在式(33)~式(42)中, v^{t_2} 更新为

$$\begin{cases} v^{t_2+1} = 2v^{t_2}, \text{fi} > e_4; v^{t_2+1} = 0.5v^{t_2}, \text{fl} > e_5; \\ v^{t_2+1} = v^{t_2}, \text{其它情况} \end{cases} \quad (43)$$

式中: fi 为连续成功的次数; fl 为连续失败的次数; e_4 和 e_5 为阈值参数。

综上所述,FIHAS算法具体步骤如下。

步骤1 设置ADGGA的最大迭代次数 T_1 ,其迭代指示 t_1 为1。初始化种群中的 Z 个个体,根据式(15)计算所有个体的适应度值,并找到当前最优个体,以当前最优个体更新历史最优个体。

步骤2 以锦标赛算法挑选新种群,若历史最优个体未被选入新种群,则用它取代新种群中最差个体。

步骤3 所有个体根据概率式(25)执行多样性增强变异,并对优化变量进行边界溢出检测和处理。

步骤4 根据式(15)计算所有个体的适应度值。

步骤5 任意两个相邻个体以概率式(16)执行交叉操作,并对优化变量进行边界溢出检测和处理。

步骤6 所有个体根据式(19)~式(23)以概率式(18)执行变异操作,并对优化变量进行边界溢出检测和处理。

步骤7 根据式(15)计算所有个体的适应度值,并更新当代最优个体和历史最优个体。

步骤8 $t_1 = t_1 + 1$;若 $t_1 < T_1$,则回到步骤2,否则输出当代种群并终止循环。

步骤9 设置最大迭代次数 T_2 且使 $t_2 = 1$ 与 $v^{t_2} = 1$;以步骤8的输出初始化任意粒子的位置和速度、粒子的历史最优位置、全局最优粒子的位置。

步骤10 根据式(31)更新惯性权重。

步骤11 根据式(26)~式(30)、式(32)分别更新普通粒子的速度和位置,并对优化变量进行边界溢出检测和处理。

步骤12 根据式(15)计算所有粒子的适应度值;对于任意粒子,若它的当代适应度值高于自身的历史最高值,则更新自身的历史最优粒子,然后在所有历史最优粒子中找到全局最优粒子。

步骤13 根据式(33)~式(37)、式(38)~式(42)更新全局最优粒子的速度和位置,并对优化变量进行边界溢出检测和处理。

步骤14 根据式(43)更新缩放因子 v^{t_2} 。

步骤15 $t_2 = t_2 + 1$;若 $t_2 < T_2$,回到步骤10,否则输出。

4 仿真及对比分析

4.1 参数设置

超密集网络中,SBS与UE被随机分布于半径500 m的宏蜂窝(宏基站覆盖区域)内,考虑1个宏基站,25个SBS,每个UE有10个任务,种群规模为32, $\theta = 10^{-20}$, $\hat{h}_1 = 0.6$, $\hat{h}_2 = 0.03$, $\hat{h}_3 = 10^{-5}$, $\omega^{\max} = 0.9$, $\omega^{\min} = 0.4$, $\mu^{\max} = 0.9$, $L = 6$, $\mu^{\min} = 0.4$, $e_1 = 2$, $e_2 = 2$, $e_3 = 2$, $e_4 = 15$, $b_1 = 0.8$, $b_2 = 0.3$, $\sigma^2 = 10^{-11} \text{ mW}$, $d_i = 200 \sim 500 \text{ KB}$, $\alpha = 10^{-24}$, $W_n = 20 \text{ MHz}$, $f_{k,i}^{\text{ioc}} = 0.5 \sim 2 \text{ GHz}$, $f_n^{\text{mmec}} = 2.5 \text{ GHz}$, $\beta = 10^{-26}$, $\tau_i^{\max} = 0.1 \sim 0.5 \text{ ms}$ 。

在仿真中,最强卸载(SO)是指在UE发射功率采用最大功率的情况下,将计算任务全部卸载至信号强度最大的SBS执行,并通过选择最低安全成本的密码算法来实现任务卸载的算法。本地计算(LC)是指UE自行完成所有计算任务,LC算法和SO算法根据任务计算需求量占总需求量的比例来分配计算资源。改进的分层自适应搜索(IHAS)是现有结合GA和PSO的卸载算法;FIHAS是研究通过改进IHAS而提出的算法,它们主要区别在于: IHAS中AGADGM的自适应变异和交叉概率采用静态权重,FIHAS中ADGGA的自适应变异和交叉概率采用动态(自适应)权重。选择SO算法是为了反映任务全部卸载情况下的性能,选择LC算法则是可以反映出任务未卸载时的性能,而FIHAS是在IHAS上改进的,因此也选择IHAS作为对比算法。

4.2 结果分析

图1(a)为揭示网络UE数对总成本影响的示意

图,其中总成本是指所有UE任务的安全成本之和。由于网络UE数量的增加导致SBS可利用带宽下降,UE发射能量的增加迫使UE最大总能量的增加。虽然网络UE数量的增加也可能导致UE最大成本的增加,但这个增加量不如最大总能耗的增加量大。网络UE数量的增加使目标函数更加关注总成本的优化。因此,随着网络UE数量的增加,总成本可能会下降。图1(a)显示了SO始终选择安全成本最低的密码算法,从而实现比IHAS和FIHAS更低的总成本。由于LC中的UE任务不涉及加密,因此它不需要支付任何费用,所以图1(a)未对其进行描述。正如图1(a)所示,本文联合优化能耗与安全成本,在使用自适应权重的情况下,FIHAS的搜索空间比IHAS更大,因此FIHAS找到了比IHAS更优的目标,前者实现了更低的总成本。

图1(b)为网络UE数对网络总能耗影响的示意图,其中网络总能耗是指UE端与SBS能耗的总和。根据图1(b)显示的结果,当网络UE数增加时,由于计算任务的增多,所有算法也产生了更高的总能耗。由于LC算法,它在本地执行任务,因此不需

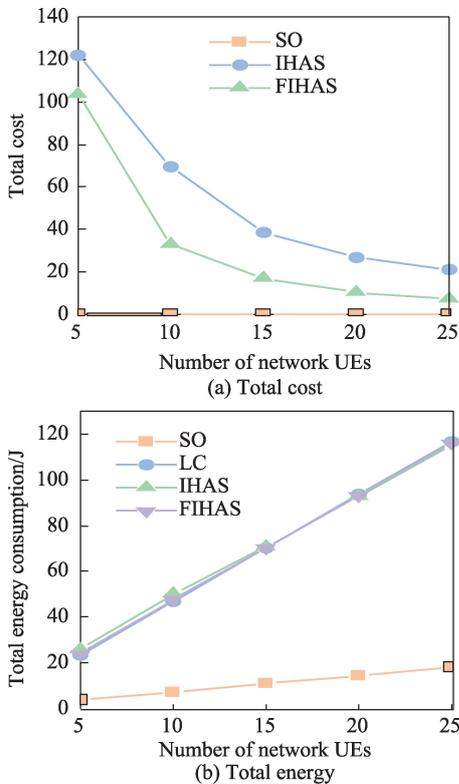


图1 网络UE数对总成本和总能耗的影响

Fig. 1 The impact of number of network UEs on total cost and total energy consumption

要为任务加密开销能量。然而,FIHAS和IHAS算法需要对卸载任务进行加密,从而产生了额外的加密能耗,同时加密过后的任务会在SBS上进行解密,这使它们总能耗略高于LC算法。虽然SO算法也需对任务进行加密,但由于它将所有UE任务卸载至信号强度最大的SBS上执行,极大降低了发射能耗,因此可能达到最低总能耗。

这说明为了实现更低的总成本,FIHAS和IHAS在能耗和安全成本之间折中,使负载相对均衡,资源得到更加有效的利用。值得注意的是,在优化问题目标函数中,能耗与安全成本是联合优化的。与IHAS中交叉与变异概率公式的静态权重不同,FIHAS利用了自适应权重,更充分地搜索问题可行解的空间。因此,后者达到更低的能耗。

图2(a)为揭示网络UE数对支持率影响的示意图,其中支持率是指满足时延约束条件的任务数占总任务数的比例。随着网络UE数量的增加,任务数量也会随之增加。这导致需要更多的带宽才能完成任务。但SBS可提供的带宽是有限的,因此随着UE数量的增加,SBS可供利用的带宽就越来越少。因此,支持分布式计算的SO、FIHAS和IHAS在网络UE数量增加时的支持率会下降,如图2(a)所示。与此不同的是,本地计算(LC)不受SBS可用带宽的影响,因此其支持率不随网络UE数的变化而变化。由于SO存在本地加密和上行发射的时延等问题,而LC不存在这些问题,因此LC的支持率比SO高。通过设置合理的惩罚因子,可以迫使更多的任务满足时延限制,从而提高FIHAS和IHAS的支持率,甚至可能达到比LC更高的支持率。图2(a)还显示,相对于IHAS,FIHAS具有更多的本地执行任务,这意味着FIHAS中边缘执行的支持率高于本地执行的支持率,从而使得FIHAS的支持率略低于IHAS的支持率。

图2(b)为揭示网络UE数影响目标函数的示意图。图1(a)已揭示随着网络UE数的增加,目标函数更加侧重于总成本的优化。这可能导致目标函数反而下降。因此,如图2(b)所示,在自适应权重下,FIHAS能够更充分地搜索问题可行解的空间,达到了比IHAS更低的目标函数值。

5 结论

针对超密集网络,设计了一种UE关联、密码服

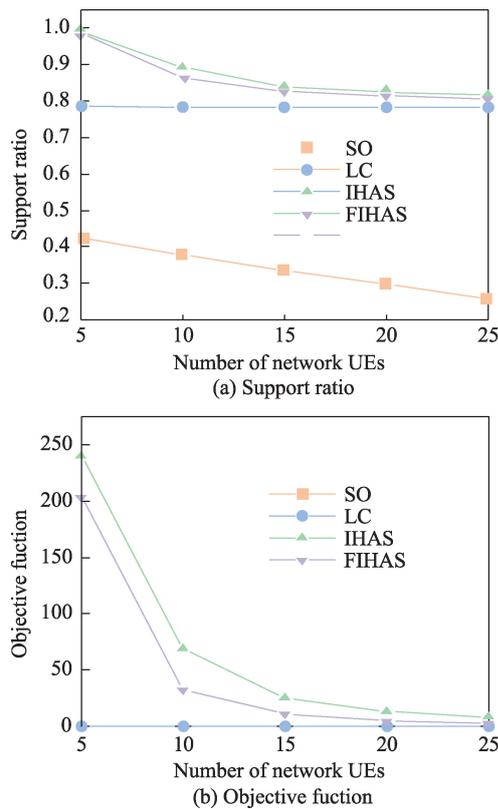


图2 网络UE数对支持率和目标函数的影响

Fig. 2 The impact of number of network UEs on support ratio and objective function

务分派、UE功率控制、UE与SBS计算资源分配联合优化策略,并且进一步改进了IHAS算法,提出了FIHAS算法。通过仿真分析,得出以下结论。

1) 该联合优化策略能够较好地最小化标准总能耗与标准总安全成本之加权和。

2) 总体上,与其他卸载算法相比,FIHAS在降低总成本方面更具优势。

3) 未来工作可以考虑网络运营成本的优化和子信道分配等。

参考文献:

- [1] HUANG B, LI Z, TANG P, et al. Security modeling and efficient computation offloading for service workflow in mobile edge computing[J]. *Future Generation Computer Systems*, 2019, 75: 775-774.
- [2] GUO F, ZHANG H, JI H, et al. An efficient computation

- offloading management scheme in the densely deployed small cell networks with mobile edge computing[J]. *IEEE/ACM Transactions on Networking*, 2018, 26(6): 2651-2664.
- [3] GUO H, LIU J, ZHANG J, et al. Computation offloading for multi-access mobile edge computing in ultra-dense networks[J]. *IEEE Communications Magazine*, 2018, 56(8): 14-99.
- [4] LU Y, CHEN X, ZHANG Y, et al. Cost-efficient resources scheduling for mobile edge computing in ultra-dense networks[J]. *IEEE Transactions on Network and Service Management*, 2022, 19(3): 3163-3173.
- [5] ZHOU T, YUE Y, QIN D, et al. Joint device association, resource allocation and computation offloading in ultra-dense multidevice and multitask IoT networks[J]. *IEEE Internet of Things Journal*, 2022, 9(19): 18695-18709.
- [6] ZAHED M I A, AHMAD I, HABIBI D, et al. Green and secure computation offloading for cache-enabled IoT networks[J]. *IEEE Access*, 2020, 8: 63840-63855.
- [7] ELGENDY I A, ZHANG W Z, ZENG Y et al. Efficient and secure multi-user multi-task computation offloading for mobile-edge computing in mobile IoT networks[J]. *IEEE Transactions on Network and Service Management*, 2020, 17(4): 2410-2422.
- [8] ZHOU T, QIN D, NIE X, et al. Energy-efficient computation offloading and resource management in ultradense heterogeneous networks[J]. *IEEE Transactions on Vehicular Technology*, 2021, 70(12): 13101-13114.
- [9] JIANG W, JIANG K, ZHANG X, et al. Energy optimization of security-critical real-time applications with guaranteed security protection[J]. *Journal of Systems Architecture*, 2015, 77(61): 282-292.



通信作者:周天清(1983—),男,副教授,博士,硕士生导师,研究方向为超密集组网、移动边缘计算与智能算法等。E-mail:zhoutian930@163.com。

(责任编辑:吴海燕)