面向工业场景数据安全的优化卸载方法

王飚, 王妲, 柯吉, 马雨庆, 张懿璞, 王长青, 李爱军

引用本文

王飚, 王妲, 柯吉, 马雨庆, 张懿璞, 王长青, 李爱军面向工业场景数据安全的优化卸载方法[J]. 计算机科学, 2023, 50(8): 286-293.

WANG Biao, WANG Da, KE Ji, MA Yuqing, ZHANG Yipu, WANG Changqing, LI Aijun. Study on Optimized Offloading for Data Security in Industrial Scene [J]. Computer Science, 2023, 50(8): 286-293.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230100082

面向工业场景数据安全的优化卸载方法

王 飚[1] 王 妲[2] 柯 吉[1] 马雨庆[2] 张懿璞[1] 王长青[3] 李爱军[3]
1 长安大学能源与电气工程学院 西安 710061
2 长安大学电子与控制工程学院 西安 710061
3 西北工业大学自动化学院 西安 710072
(wangbiao@chd.edu.cn)

摘 要 针对工业场景数据传输过程中存在的安全卸载问题,文中首次将安全策略作为决策变量融入优化问题,应用计算卸载原理以及差分进化算法,提出了一种数据安全卸载算法。首先针对工业现场设备的本地计算、本地边缘计算、跨车间边缘计算和云计算4种计算模式以及数据安全进行数学建模,将多级安全策略、任务卸载和资源分配相融合,构建了数据安全卸载模型。综合考虑时延和安全风险概率的影响,设计最大化设备满意度的目标函数,形成了安全优化卸载方案。针对该优化问题,提出了一种基于改进的差分进化策略的数据安全卸载算法,在满足最优解的同时,在满足时延和安全风险的要求下实现系统的设备满意度最大化。相比GASORA算法、GSOJRA算法和DEDSTO-NS算法,所提算法不仅使现场设备满足了时延和风险概率的要求,并在保障数据安全性的同时,将设备满意度提高了35%。仿真结果证实了所提方法的有效性,且有一定的现实应用价值。

关键词:安全策略;数据安全卸载;差分进化;安全风险概率;设备满意度
中图法分类号 TP393

Study on Optimized Offloading for Data Security in Industrial Scene

WANG Biao[1],WANG Da[2],KE Ji[1],MA Yuqing[2],ZHANG Yipu[1],WANG Changqing[3] and LI Aijun[3]
1 School of Energy and Electrical Engineering,Chang'an University,Xi'an,710061,China
2 School of Electronics and Control Engineering,Chang'an University,Xi'an,710061,China
3 School of Automation,Northwestern Polytechnical University,Xi'an,710072,China

Abstract The problem of security offloading in data transmission in industrial scenarios has gained wide attention. This paper is the first to integrate security policy as a decision variable into the optimization problem. It applies computational offloading principles and differential evolutionary algorithms,and proposes a data security offloading algorithm. Firstly,mathematical modeling conducted for four computing modes of industrial field devices:local computing,local edge computing,cross-plant edge computing in this paper,and cloud computing,as well as data security,and a data security offloading model is constructed by integrating multi-level security policies,task offloading,and resource allocation. Then,the security-optimized offloading scheme is formed by designing the objective function of maximizing device satisfaction by considering the effects of time delay and security risk probability. Finally,for this optimization problem,a data security offloading algorithm based on an improved differential evolution strategy is proposed to maximize the device satisfaction of the system while satisfying the optimal solution with the latency and security risk requirements. Compared with the GASORA,GSOJRA and DEDSTO-NS algorithms,the proposed algorithm enables the field devices to satisfy the delay and risk probability requirements. Furthermore,it improves the device satisfaction by 35% while guaranteeing data security. Simulation results confirm the effectiveness of the proposed method and have some realistic application value.

Keywords Security strategy,Data security offloads,Differential evolution,Security risk probability,Equipment satisfaction

到稿日期:2023-01-16 返修日期:2023-04-23
基金项目:陕西省自然科学基础研究计划重点项目(2019JZL-06);陕西省2023年重点研发计划(2023-YBSF-285)
This work was supported by the Key Projects of Natural Science Basic Research Program of Shaanxi Province(2019JZL-06) and 2023 Key Research and Development Program of Shaanxi Province,China(2023-YBSF-285).
通信作者:柯吉(keji@chd.edu.cn)

1　引言

随着工业互联网的发展以及生产效率的不断提高，生产过程中所产生的数据量也在不断增长，大多数设备都需要更多的计算和通信资源，然而资源和能量有限，难以满足工业的实时性要求。边缘计算因具有快速性、便捷性和分布式等特点，在数据卸载及安全领域发挥着重要的作用。由于边缘计算需要面对数量庞大的接入设备和更为复杂的异构网络，且边缘节点资源比较有限[1]，因此边缘节点也面临从物理、协议、隐私和数据角度受到的恶意攻击。基于上述描述，保证数据安全对工业系统设备是至关重要的。因此，对生产制造、煤炭采掘等工业场景下数据安全卸载的研究具有重要的理论和实际意义。数据安全、身份认证、隐私保护和访问控制这 4 个部分构成了云边协同模式下数据安全与隐私保护体系[2]。目前大多数研究的重点在数据安全、身份认证与隐私保护这 3 个领域[3]，根据这 3 个领域经常使用的技术，将考虑安全与隐私保护的计算卸载工作划分为：基于加密算法的计算卸载、基于身份认证的计算卸载和基于差分隐私的计算卸载。

针对加密算法的计算卸载，Elgendy 等[4]提出了一种带有数据安全的多用户资源分配和计算卸载模型，该模型在整个系统的能量和时间性能方面具有一定的有效性。Elgendy 等[5]还提出了一种高级加密的方法以满足数据安全的要求，最终证明了该算法具有一定的适应性和扩展性，可以节省更多的卸载开销。针对身份认证的计算卸载，Song 等[6]提出了一种基于车辆移动性的任务卸载算法。该方案可以在保证边缘计算系统的安全性能的同时大大减少系统能耗。针对差分隐私的计算卸载，Jiang 等[7]提出了一种高效双拍卖模型，该拍卖模型在保证较少的计算开销的情况下确保了用户隐私。Yao 等[8]通过在增广拉格朗日函数中添加时变高斯噪声的方式来提供差分隐私保护，在一定程度上减少了数据传输过程中可能发生的隐私泄露。

随着工业物联网的发展，工业物联网系统所要考虑的重要问题之一是设备数据安全问题。Zhou 等[9]提出了一种拥有较高通信效率的带有隐私保护特性的多维度安全查询方案，但该方案的计算及通信开销较大，通信效率较低。Ren 等[10]提出了基于多级身份验证和轻量级加密的电力物联网数据安全方案，采用加密算法对敏感数据进行处理，但是加密数据共享的方式不够灵活。He 等[11]提出了一种物联网环境下云数据存储安全及隐私保护策略，但未考虑用户间资源分配的公平性问题。

综上，以往针对工业场景的实际应用，或仅考虑到安全保护方法存在的缺陷，或仅考虑安全卸载问题，而同时综合考虑任务卸载和安全保护的应用较少。本文提出了一种具有安全保护服务的云边端协同系统框架，构建安全卸载计算模式；引入安全策略、任务卸载以及资源分配这 3 种决策变量，考虑将计算时间、数据的安全性以及边缘/云服务器的计算资源作为约束，构建一个最大化设备满意度优化问题，并使用融合了两种变异策略的差分进化算法对该问题进行求解。

2　系统建模及其问题形成

2.1　系统结构分析

在现场设备将计算任务上传至边缘计算层或云计算层的过程中，计算任务很容易遭到恶意攻击而导致重要生产数据泄露，因此在云边协同系统架构基础上引入了安全保护服务，提出了一种具有安全保护服务的云边端协同系统框架，如图 1 所示。

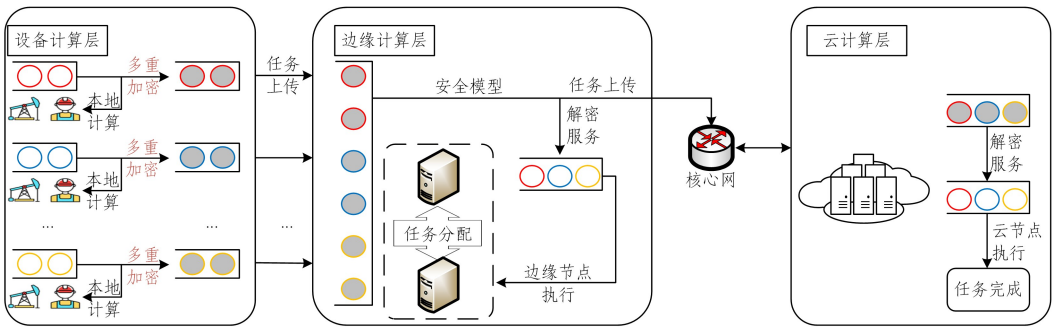

图 1　带有多级安全策略的云边端协同系统框架

Fig. 1　Cloud-side collaboration system framework with multi-level security policies

该架构具备设备计算层、边缘计算层和云计算层这 3 种不同类型的计算资源，计算资源类型的多样性为生产制造提供了十分充足的算力储备，可以有效地减少现场设备的时延和能耗。同时该全新的系统架构考虑了数据安全问题对整个任务卸载过程可能造成的影响，为将要发送至边缘服务器或云服务器的计算任务提供了安全保护服务，避免了计算任务遭受恶意网络攻击。

安全保护服务的工作原理即某个现场设备生成的计算任务将要发送至边缘服务器或云服务器时，先使用加密算法对计算任务进行加密，之后的操作有两种情况：如果发送的目标对象是边缘服务器，其就会对加密后的数据解密并进行计算，计算完成之后将计算结果重新回传给现场设备；如果发送的目标对象是云服务器，那么它的作用就是中继节点，它会将来自现场设备的数据不做任何处理即转发到云服务器上，云服务器会进行数据解密和运算，再将计算结果回传给现场设备。

2.2　系统建模

每个现场设备有 4 种计算模式可供选择，即本地计算模式、本地边缘计算模式、跨车间边缘计算模式和云计算模式，分别用符号 I_i^L，I_i^{ML}，$I_{i,h}^{MS}$ 和 I_i^C 表示。

当现场设备 i 选择将计算任务上传至边缘服务器或者

云服务器时，由于计算任务有数据安全方面的要求，因此需要对待传送的数据提供加密服务，然后才能通过网络传输加密后的数据。一般来说，现场设备偏向于信任那些有良好声誉的边缘服务器或云服务器。现场设备 i 提交的计算任务对边缘服务器的安全要求 Sd_{edge} 和云服务器的安全要求 Sd_{cloud} 会随当前所处的网络环境发生变化，安全保护服务也要据此提供不同级别的安全策略，因此定义一个多级安全策略集合 $K=\{p_1,p_2,\cdots,p_k\}$，$K\in\{1,2,\cdots,k\}$，其中每个安全策略 p_K 代表了相应的安全级别 K 和一种加密算法，与每种加密算法相对应的参量有安全水平 Sl_K、加密服务计算量 α_K（单位为 CPU 周期数/bit）、每位数据的能量消耗 γ_K（单位为 mJ/bit）以及解密服务计算量 β_K（单位为 CPU 周期/bit）。为建立一个有效的安全模型，定义一个二进制变量 $S_i\in\{0,1\}$ 来表示边缘服务器或云服务器与安全策略之间的关系：$S_i=1$ 表示现场设备 i 对计算任务使用安全策略 p_K 为其提供安全服务；$S_i=0$ 表示现场设备 i 不对计算任务采取安全策略。安全水平 Sl_K 是根据算法执行时间来衡量的，将其标准化定义为[0，1]，并认为最慢的算法安全水平最高。本文所有公式的相关符号说明如表 1 所列。

表 1 符号说明

Table 1 Symbol definition

符号	描述
f_i^L	现场设备 i 的计算能力
c_i	任务的计算量（单位为 CPU 周期数/MB）
d_i	任务的数据量大小
κ	有效电容系数
g_θ	路径损失常数
θ	路径损失指数
u_θ	参考距离
u_i	设备与本地服务器的物理距离
p_i	设备 i 上行发射功率
W	系统带宽
σ	背景噪声功率
f	本地边缘服务器 λ_i 分配给设备 i 的计算能力
$R_{\lambda_i,h}$	本地边缘服务器 λ_i 与跨车间边缘服务器 h 之间的传输速率，其为常值
$f_{i,h}^M$	跨车间边缘服务器 h 分配给设备 i 的计算能力
R_{upcloud}	到云服务器的传输速率
f_i^C	云服务器分配给设备 i 的计算能力
t_{dl}	计算任务完成计算的最大时间限制
pr	计算任务需要满足的最大安全风险概率
DoS_i	现场设备满意度
$\mathscr{P}$	对计算任务使用的安全策略集合
$Pr_i(Sl_K)$	计算任务的风险概率
ξ	计算任务的完成时间超过了期限时间时的惩罚因子
F	变异因子，也可以被称为缩放因子
X_{best}	适应度值最优的个体
T	当前迭代次数
μ	自适应调节因子
n_{rand}	取值范围为[1，D]的随机整数

2.2.1 数据安全模型

智能制造过程中所产生的生产数据是极为重要的知识产权，它可能会被黑客恶意更改和窃取，从而对生产制造造成无法估计的损失。假设网络攻击的时机满足泊松分布，然后将安全策略 p_K 下计算任务的风险概率建模为指数分布，其表达式如式(1)所示[12]：

$$Pr_i(Sl_K)=1-\exp(-\pi\max\{Sd-Sl_K,0\}) \tag{1}$$

为了保障现场设备 i 上传的计算任务的安全性，整体的安全要求应该满足条件，如式(2)和式(3)所示：

$$Sd_i=\max\{Sd_i^{\text{edge}},Sd_i^{\text{cloud}}\} \tag{2}$$

$$Sd=\max\{Sd_i\} \tag{3}$$

因此，安全策略 p_K 所能提供的安全水平 Sl_K 只能大于或等于整体的安全要求 Sd，如式(4)所示：

$$Sl_K=\text{argmin}(Sl_K-Sd),Sl_K-Sd\geqslant 0 \tag{4}$$

当 $Pr_i(Sl_K)=0$ 时，代表采取安全策略 p_K 为计算任务提供的安全水平 Sl_K 能够有效地保护现场设备的数据安全。

2.2.2 本地计算模型

当现场设备选择本地计算模式时，现场设备 i 不需要对计算任务采取任何安全策略，因此延时和能耗只与设备本地 CPU 的性能有关。本地计算模式下任务的时间消耗如式(5)所示：

$$D_i^L=\frac{c_id_i}{f_i^L} \tag{5}$$

本地计算模式下任务的能量消耗可以通过式(6)得到。

$$E_i^L=\kappa(f_i^L)^2c_id_i \tag{6}$$

2.2.3 本地边缘计算模型

当现场设备选择本地边缘计算模式时，则在现场设备 i 将计算任务卸载至本地边缘服务器之前，应当执行安全服务以保证数据的安全性。因此，计算任务将经历 4 个步骤，即数据加密服务、数据传输、数据解密服务和执行任务，上述过程如图 2 所示。现场设备 i 的主要职责是为生成的计算任务添加加密服务，然后将计算任务传输到本地边缘服务器上；本地边缘服务器的主要职责是对计算任务进行解密服务和执行任务这两项。

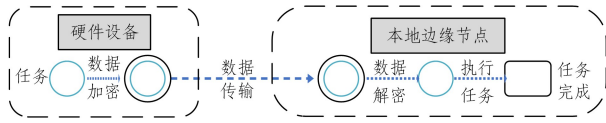

图 2 本地边缘计算模式下的任务卸载过程

Fig. 2 Task offloading process in local edge computing mode

当数据加密完成后，现场设备需要将加密后的数据传输给本地边缘服务器。对于大多数加密算法来说，其加密前后的数据量相差不大。因此计算任务加密后的数据量仍然可以用 d_i 表示。此时现场设备 i 到本地边缘服务器的传输速率为[13]：

$$R_{i,\lambda_i}=W\log_2\left(1+\frac{p_ig_0(u_0/u_i)^\theta}{\sigma^2}\right) \tag{7}$$

设备 i 与本地边缘服务器之间的传输时间为：

$$D_{i,\lambda_i}^{\text{LtoML}}=\frac{d_i}{R_{i,\lambda_i}} \tag{8}$$

本地边缘服务器执行任务的计算时间为：

$$D_{i,l_i}^{\text{MLexe}}=\frac{c_id_i}{f_{i,\lambda_i}^M} \tag{9}$$

考虑了传输时间和计算时间之后，还需要讨论计算任务在加密与解密操作上的时间消耗，因此本地边缘计算模式下设备 i 生成的计算任务产生的时延为：

$$D_i^{\mathrm{ML}}=S_i\cdot\frac{\alpha_l d_i}{f_i^{\mathrm{L}}}+\frac{d_i}{R_{i,\lambda_i}}+S_i\cdot\frac{\beta_l d_i}{f_{i,\lambda_i}^{\mathrm{M}}}+\frac{c_l d_i}{f_{i,\lambda_i}^{\mathrm{M}}} \tag{10}$$

能量消耗只考虑设备端，本地边缘计算模式下任务产生的能耗不仅包含了设备 i 到本地边缘服务器之间的传输能耗，同时还包含本地设备为计算任务添加加密服务所产生的能量消耗，因此该计算模式下任务产生的总能耗为：

$$E_i^{\mathrm{ML}}=S_i\gamma_l d_i+p_i\frac{d_i}{R_{i,\lambda_i}} \tag{11}$$

2.2.4　跨车间边缘计算模型

跨车间边缘计算模式下的计算任务卸载过程如图 3 所示，现场设备 i 会对计算任务增添加密服务并发送到中继边缘节点上，中继边缘节点只是作为中间站起到中转的作用，不会对计算任务本身进行任何不必要的操作，最终计算任务会被传送到跨车间边缘节点上，跨车间边缘节点会将计算任务进行解密操作并执行该任务。综上所述，任务完成所需的时间不仅由设备与本地边缘服务器之间的传输时间、跨车间服务器之间的传输时间和跨车间边缘服务器执行任务的计算时间这 3 部分组成，还包含现场设备 i 进行加密操作所消耗的时间和跨车间边缘服务器进行解密操作所消耗的时间。

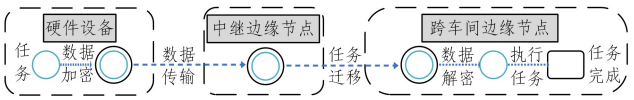

图 3　跨车间边缘计算模式下的任务卸载过程

Fig. 3　Task offloading process in cross-shop edge computing mode

本地边缘服务器与跨车间边缘服务器之间的传输时间为：

$$D_{i,h}^{\mathrm{MLtoMS}}=\frac{d_i}{R_{\lambda_i,h}} \tag{12}$$

跨车间边缘服务器执行任务的计算时间为：

$$D_{i,h}^{\mathrm{MSexe}}=\frac{c_i d_i}{f_{i,h}^{\mathrm{M}}} \tag{13}$$

除了对传输时间和计算时间进行讨论，还需要加上计算任务在加密与解密操作上的时间消耗，则跨车间边缘计算模式下设备 i 生成的计算任务卸载到跨车间边缘服务器 h 上进行计算的总时延为：

$$D_{i,h}^{\mathrm{MS}}=S_i\frac{\alpha_l d_i}{f_i^{\mathrm{L}}}+\frac{d_i}{R_{i,\lambda_i}}+\frac{d_i}{R_{\lambda_i,j}}+S_i\frac{\beta_l d_i}{f_{i,h}^{\mathrm{M}}}+\frac{c_l d_i}{f_{i,h}^{\mathrm{M}}} \tag{14}$$

跨车间边缘计算模式下任务所产生的能耗与本地边缘计算模式类似，包含了设备到本地边缘服务器之间的传输能耗以及为计算任务添加加密服务所产生的能量消耗，因此跨车间边缘计算模式下任务的总能耗为：

$$E_{i,\lambda_i}^{\mathrm{MS}}=S_i\gamma_l d_i+p_i\frac{d_i}{R_{i,\lambda_i}} \tag{15}$$

2.2.5　云计算模型

云计算模式下的任务卸载过程如图 4 所示，现场设备 i 在本地对计算任务进行数据加密，然后通过无线网络将加密后的计算任务传送到本地边缘服务器，本地边缘服务器作为中继边缘节点把计算任务转发到云服务器上，云服务器接收到计算任务之后首先进行解密操作，然后执行计算任务并把计算结果回传给现场设备 i，至此任务完成。

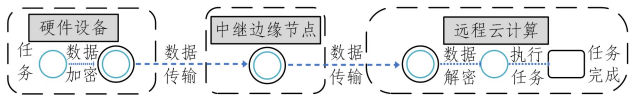

图 4　云计算模式下的任务卸载过程

Fig. 4　Task offloading process in cloud computing mode

综上所述，云计算模式下计算任务完成所需的时延不仅包括传输时间和计算时间，还包括加密和解密所消耗的时间，下面分别对云计算模式下的时延和能耗进行阐述。

由于所有车间的边缘服务器与云服务器之间的传输速率是一致的，因此任务由边缘服务器上传到云服务器的传输时延为：

$$D_i^{\mathrm{MtoC}}=\frac{d_i}{R_{\mathrm{upcloud}}} \tag{16}$$

任务由云服务器提供的计算资源进行运算，产生的运算时延为：

$$D_i^{\mathrm{Cexe}}=\frac{c_i d_i}{f_i^{\mathrm{C}}} \tag{17}$$

综合完传输时延和运算时延之后，再考虑加入对计算任务进行加密和解密操作所产生的时间消耗，最终云计算模式下现场设备 i 的总时延为：

$$D_i^{\mathrm{C}}=S_i\frac{\alpha_l d_i}{f_i^{\mathrm{L}}}+\frac{d_i}{R_{i,\lambda_i}}+\frac{d_i}{R_{\mathrm{upcloud}}}+S_i\frac{\beta_l d_i}{f_i^{\mathrm{C}}}+\frac{c_l d_i}{f_i^{\mathrm{C}}} \tag{18}$$

云计算模式下任务所产生的能耗与边缘计算模式类似，同样包含了设备到本地边缘服务器之间的传输能耗以及针对计算任务的加密解密操作的能量消耗，可表述为：

$$E_i^{\mathrm{C}}=S_i\gamma_l d_i+p_i\frac{d_i}{R_{i,\lambda_i}} \tag{19}$$

基于对以上 4 种计算模式和数据安全模型的阐述，可以得出现场设备 i 生成的计算任务的总时延如式(20)所示：

$$D_i=I_i^{\mathrm{L}}D_i^{\mathrm{L}}+I_i^{\mathrm{ML}}D_i^{\mathrm{ML}}+\sum_{h\in\mathcal{M}_i}I_{i,h}^{\mathrm{MS}}D_{i,h}^{\mathrm{MS}}+I_i^{\mathrm{C}}D_i^{\mathrm{C}} \tag{20}$$

计算任务所产生的总能耗表达式如式(21)所示：

$$E_i=I_i^{\mathrm{L}}E_i^{\mathrm{L}}+I_i^{\mathrm{ML}}E_i^{\mathrm{ML}}+\sum_{h\in\mathcal{M}_i}I_{i,h}^{\mathrm{MS}}E_{i,h}^{\mathrm{MS}}+I_i^{\mathrm{C}}E_i^{\mathrm{C}} \tag{21}$$

2.3　优化卸载问题形成

智能生产过程中，由于工业应用程序越来越复杂且对时间延迟愈发敏感，要求现场设备在一定的时间内完成任务。同时为满足智能工厂对于数据安全性的硬性要求，需要通过为计算任务提供安全保护服务使其风险概率 $Pr(Sl_K)$ 为 0。定义一个二进制变量 DoS(Degree of Satisfaction)为单个现场设备的满意度，DOS 定义为：

$$DoS_i=\begin{cases}1, & D_i<t_{dl}\ \text{and}\ Pr_i(Sl_K)<pr\\ 0, & \text{其他}\end{cases} \tag{22}$$

当计算任务在规定的时间限制内完成任务且其风险概率 $Pr(Sl_K)$ 为 0 时，现场设备 i 的满意度 $DoS_i=1$，否则满意度 $DoS_i=0$。

目标函数是最大化所有现场设备的满意度。该优化问题的决策变量分别为任务卸载决策 $\mathcal{A}=\{I_i^{\mathrm{L}},I_i^{\mathrm{ML}},I_{i,h}^{\mathrm{MS}},I_i^{\mathrm{C}}\}$、计算资源分配 $\mathcal{F}=\{f_{i,j}^{\mathrm{M}},f_i^{\mathrm{C}}\}$ 和安全策略 $\mathcal{P}=\{p_1,p_2,\cdots,p_k\}$，这 3 种决策变量决定了问题的目标函数值，形成的优化问题如式(23)所示：

$$\begin{aligned}
&\max_{\mathscr{A},\mathscr{F},\mathscr{P}} \frac{1}{N}\sum_{i\in N} DoS_i \\
&\text{s.t. C1}: I_i^{\mathrm{L}}, I_i^{\mathrm{ML}}, I_{i,h}^{\mathrm{MS}}, I_i^{\mathrm{C}} \in \{0,1\}, \forall\, i\in\mathscr{N}, h\in\mathscr{M}_i \\
&\quad \text{C2}: I_i^{\mathrm{L}} + I_i^{\mathrm{ML}} + \sum_{h\in\mathscr{M}_i} I_{i,h}^{\mathrm{MS}} + I_i^{\mathrm{C}} = 1, \forall\, i\in\mathscr{N} \\
&\quad \text{C3}: \sum_{i\in\mathscr{N}} f_{i,j}^{\mathrm{M}} \leqslant f_j^{\mathrm{M_{max}}}, \forall\, j\in\mathscr{M} \\
&\quad \text{C4}: \sum_{i\in\mathscr{N}} f_i^{\mathrm{C}} \leqslant f^{\mathrm{C_{max}}} \\
&\quad \text{C5}: f_{i,j}^{\mathrm{M}}, f_i^{\mathrm{C}} > 0, \forall\, i\in\mathscr{N}, j\in\mathscr{M} \\
&\quad \text{C6}: D_i < t_{dl} \\
&\quad \text{C7}: Pr_i(Sl_K) = 0
\end{aligned} \tag{23}$$

其中，约束条件 C1 表示计算模式选择为二进制变量；约束条件 C2 表示计算任务是原子化的且不能被分割，只能在一个位置进行计算；约束条件 C3 和 C4 表示边缘服务器和云服务器的计算资源并不是没有限制的；约束条件 C5 表示边缘服务器和云服务器分配给现场设备 i 的计算资源不能小于 0；约束条件 C6 表示计算任务必须在一定时间限制下完成；约束条件 C7 表示计算任务的数据安全性要求。

3 基于改进的差分进化策略的数据安全卸载算法

针对安全优化卸载问题，本文提出了基于改进的差分进化策略的数据安全卸载算法。下面分别从染色体编码和适应度函数、变异操作、交叉操作及选择操作 4 个方面详细阐述改进的差分进化算法处理考虑数据安全的任务卸载与计算资源分配问题的设计思路。

3.1 染色体编码和适应度函数

差分进化(Differential Evolution，DE)算法一开始初始化一组初始种群，种群包含了若干个体，每个个体都有其独一无二的基因。假设种群随机产生了 M 个维度为 D 的个体向量，定义一个变量 X_m 来表示种群中第 m 个个体基因，所以种群矩阵变量为：

$$\boldsymbol{POP}=[X_1\ X_2\cdots X_m\cdots X_M] \tag{24}$$

种群矩阵中的每个个体基因向量都可以被看作是问题的一个待定解，与其他进化算法一样，每个个体向量通常不能够直接参与到运算中，需要对其进行编码操作。与遗传算法采用的二进制编码方式不同，DE 算法为了方便求解往往选择使用实数编码方式。相较于二进制编码方式，实数编码就方便了很多，它直接将决策变量的真实值作为每个个体基因向量的值，省去了解码的麻烦，提高了数值的精度并简化了操作难度，从而降低了算法的复杂度，使算法可以更快地达到收敛状态。优化问题旨在最大化所有现场设备的满意度，该问题的决策变量包括卸载决策变量 $\mathscr{A}$、安全策略 $\mathscr{P}$ 和计算资源分配方案 $\mathscr{F}$，将这 3 个决策变量融合到一起，每个个体基因表示为：

$$X_m=[\mathscr{A}\ \mathscr{P}\ \mathscr{F}] \tag{25}$$

其中，每个个体中卸载决策变量 $\mathscr{A}$ 和安全策略 $\mathscr{P}$ 这两种决策变量是整数变量，但是计算资源分配方案 $\mathscr{F}$ 却是实数变量，DE 算法擅长求解连续变量的优化问题，无法直接求解这种整数变量与实数变量混合的优化问题。

为了让 DE 算法可以求解混合整数优化问题，考虑加入映射环节。该环节的思路如下：计算资源分配变量 $\mathscr{F}$ 的取值范围是所有决策变量中最大的，因此以该变量作为基准，卸载决策变量 $\mathscr{A}$ 的取值范围为[0，1]，安全策略变量 $\mathscr{P}$ 的取值范围为[1，k]。对卸载决策变量 $\mathscr{A}$ 进行二次编码，将计算资源分配变量 $\mathscr{F}$ 的取值范围划为二等份，然后取值一一对应子区间上的一个整数。依据同样的原理对安全策略变量 $\mathscr{P}$ 进行二次编码，将计算资源分配变量 $\mathscr{F}$ 的取值范围划为 k 等份，之后取值分别对应子区间上的一个整数。计算资源分配变量 $\mathscr{F}$ 的编码方式不变，直接使用实数作为个体变量。通过以上方式，采用实数编码方式对个体基因变量进行编码，在计算适应度值时实数变量可以映射为相对应的整数变量，从而实现了求解混合整数优化问题的目的。

适应度函数值会根据决策变量的变化而发生变化，因此我们将其作为评估个体基因好坏的标准，同时为了平衡约束条件的影响加入了惩罚项，则适应度函数的表达式为：

$$Fitness=\frac{1}{N}\sum_{i\in\mathscr{N}} DoS_i+\xi\sum_{i\in\mathscr{N}}\max(0,t_{dl}-D_i) \tag{26}$$

3.2 变异操作

DE 算法的变异操作运用了差分方式，算法名称中的“差分”也由此而来。变异策略有很多种方法可供选择，常见的方法主要有两种，分别是 DE/rand/1/bin 和 DE/best/1/bin。

当选择 DE/rand/1/bin 时，个体的变异策略表达式如式(27)所示：

$$V_m(T+1)=X_{r3}(T)+F(X_{r1}(T)-X_{r2}(T)) \tag{27}$$

当选择 DE/best/1/bin 时，个体的变异策略表达式如式(28)所示：

$$V_m(T+1)=X_{\mathrm{best}}(T)+F(X_{r1}(T)-X_{r2}(T)) \tag{28}$$

分析以上两种方法的优缺点，方法 DE/rand/1/bin 由于采取了随机挑选的策略来选择待变异的个体，因此可以使算法具有很强的全局寻优能力，同时具有优良的种群多样性，但是缺点也很明显，即会导致算法收敛速度变慢；方法 DE/best/1/bin 挑选待变异个体时只选择适应度值最优的个体，所以算法可以很快找到局部最优解，大大提高了算法效率，其缺点就是容易陷入局部最优的情况。为了发挥两种方案优点，同时尽量避免其缺点，采用自适应交替变异算子将两种方法进行融合。在算法迭代前期，采用 DE/rand/1/bin 方法扩大算法的搜索范围以避免陷入局部最优，到了算法迭代后期，采用 DE/best/1/bin 方法提高算法的收敛速度。融合两种变异策略后的全新变异算子如式(29)所示：

$$V_m(T+1)=\begin{cases} X_{r3}(T)+F\cdot(X_{r1}(T)-X_{r2}(T)), & \mu<rand \\ X_{\mathrm{best}}(T)+F\cdot(X_{r1}(T)-X_{r2}(T)), & \text{其他} \end{cases} \tag{29}$$

可以根据迭代次数选择变异策略，具体表达式为：

$$\mu=2-\mathrm{e}^{T/G_{\max}\cdot\lg 2} \tag{30}$$

3.3 交叉操作

DE 算法中交叉操作是在变异操作之后进行的，其目的是提高种群多样性，扩大搜索范围。定义两个变量 $X_{mn}(T)$ 和 $V_{mn}(T+1)$ 分别代表当前种群个体 $X_m(T)$ 和变异个体 $V_m(T+1)$ 的第 n 维元素，交叉操作的作用是在交叉概率 CR 的判定下，种群个体 $X_m(T)$ 所有维度的元素与变异个体 $V_m(T+1)$ 所有维度上的元素随机进行重新组合，从而生成新

的候选个体 $U_m(T+1)$。同时为了保证种群中个体基因是处在不断进化过程中，新生成的候选个体如式(31)所示：

$$U_{mn}(T+1)=\begin{cases}V_{mn}(T+1), & rand<CR \text{ or } n=n_{\text{rand}}\\ X_{mn}(T), & \text{其他}\end{cases} \tag{31}$$

3.4　选择操作

DE 算法的选择操作本质上是用贪婪算法去选择适应度值最优的个体加入到新的种群中，表达式如式(32)所示：

$$X_m(T+1)=\begin{cases}U_m(T+1),\\ \quad Fitness(U_m(T+1))>Fitness(X_m(T))\\ X_m(T), \quad \text{其他}\end{cases} \tag{32}$$

综上所述，DEDSTO 算法的迭代步骤如算法 1 所示。

算法 1　DEDSTO 算法

输入：种群个体数目 M，个体维度 D，最大迭代次数 G_{max}，变异因子 F，交叉概率 CR

输出：最优的任务卸载决策 $\mathscr{A}^*$、安全策略 $\mathscr{P}^*$ 和计算资源分配策略 $\mathscr{F}^*$，全局最优满意度

1. 随机生成一个具有 M 个个体的种群，形成初始种群；
2. 根据适应度函数公式计算出种群中所有个人的适应度函数值，记下具有最优适应度函数值的个体；
3. for T=1:G_{max} do
4.　对某些个体进行变异操作，得到一个临时种群；
5.　对临时种群中所有个体进行交叉操作，得到一个候选种群；
6.　判断候选解种群中所有个体的优劣，选择优秀的个体作为新一代目标种群；
7.　计算新一代目标种群的适应度函数值，并从中选出最优个体；
8.　T=T+1
9.　end for

4　仿真验证及结果分析

4.1　实验设置

本文仿真实验场景设计为 10 个车间，每个车间都配备有一台本地边缘服务器和若干个现场设备，为了方便比较系统性能，每个车间的现场设备数目相同，系统中总共有 N 个用户(N 的总取值为 10 的倍数)。为了整合所有车间的生产数据，平台仿真实验中还添加了一个计算资源有限的云服务器。仿真实验采用参考文献[14]的数据集，具体系统参数情况如下：计算任务数据大小为 400 MB，现场设备计算能力设定为1 GHz，边缘服务器计算能力设定为 15 GHz，云服务器最大计算能力设定为 50 GHz，从边缘到云的传输速率设定为 2 MB/s。针对工业数据面临的数据泄露问题，设计了一个多级安全策略集，策略集主要包含 6 种策略，详细信息如表 2 所列。本文采用文献[15]使用的安全策略，并将安全策略的能量 α_K 消耗的单位设置为 mJ/bit。每种安全策略包含加密算法和解密算法，所以使用变量表示加密算法的计算工作负载(单位为 CPU 周期数/bit)，以此来区分加密算法和解密算法。根据文献[15]中报告的观察结果来定义 α_K 的大小，结果表明加密算法的执行持续时间随着安全级别的提高而增加。解密计算工作负载 β_K(单位为 CPU 周期数/bit)的数值确定依赖于文献[15-17]中报告的解密持续时间和加密之间的关系。文献[18]中的 IDEA 加密算法，其解密和加密的持续时间几乎相同。因此对于采用 IDEA 加密算法的安全策略 p_4，我们设置了解密计算工作负载 $\beta_4=300$，它与加密计算工作负载 α_4 的值相同。其他安全策略的解密计算工作负载也以类似的方式设置。

表 2　安全策略详细信息

Table 2　Security policy details

安全策略	安全水平	加密算法	α_l	γ_l	β_l
p_1	0.38	RC4	100	2.5296	90
p_2	0.57	RC5	200	5.0425	280
p_3	0.62	BLOWFISH	250	6.8370	350
p_4	0.84	IDEA	300	7.8528	300
p_5	0.91	SKIPJACK	350	8.7073	400
p_6	1.00	3DES	1050	26.3643	1700

为验证 DEDSTO 算法的有效性，将其与以下 3 种算法进行对比。

(1)基于遗传算法的安全卸载与资源分配(Security Offloading and Resource Allocation Based On Genetic Algorithm，GASORA)策略：依据文献[19]使用一种结合基于知识的交叉算子的免疫遗传算法。该算法扩展了可行解的范围，并快速生成全局最优解。

(2)基于贪婪算法的安全卸载与资源分配(Greedy Security Offloading and Joint Resource Allocation，GSOJRA)策略：依据文献[20]使用一种基于贪婪算法的卸载决策算法。该算法考虑了相关的能量参数和各种动态参数。

(3)无安全服务的 DEDSTO 算法(DEDSTO-NS)：将其作为对照算法来说明安全服务对系统的影响。

4.2　实验结果分析

4.2.1　现场设备数目变化对系统性能指标的影响

图 5 给出了智能制造场景下现场设备数目的变化与系统总能耗之间的关系。根据图 5 可知，随着现场设备数目的增加，系统总能量消耗也增加。DEDSTO，GASORA 和 GSOJRA 这 3 种算法都具备安全服务，本质是比较差分进化算法、遗传算法和贪婪算法这 3 种算法的性能，其中本文提出的 DEDSTO 算法与其他两种算法相比能量消耗最小，原因在于本文算法加入了映射环节，使改进后算法可以更好地处理混合整数非线性问题。图 5 中还有一种与以上 3 种算法截然不同的算法——DEDSTO-NS 算法，它没有考虑加入安全服务的情况，因为添加安全服务会产生一定的能量损耗，因此 DEDSTO-NS 算法产生的系统总能耗相比具有安全服务的 DEDSTO 算法有所减少。

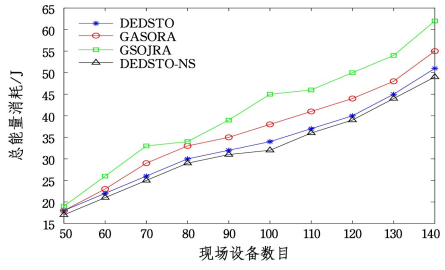

图 5　现场设备数目与系统总能耗的关系

Fig. 5　Relationship between the number of field devices and total system energy consumption

图 6 给出了参与智能制造的现场设备数目的变化与设备

综合满意度变化的关系。从图中可以看出，DEDSTO-NS 算法的满意度不会随着设备数目的增加而提升，基本稳定在 45%左右，原因是该算法没有应用安全服务，从而导致计算任务几乎都留在设备的本地 CPU 进行处理。由于没有对其提供安全服务，生产数据很容易被对手窃取或者篡改，现场设备的综合满意度会受到严重的不良影响。

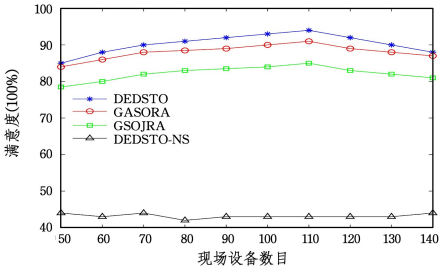

图 6 现场设备数目对设备满意度的影响

Fig. 6 Impact of the number of field devices on system satisfaction

具备安全模型的 3 种算法有一些相同点：当数量较小时，设备满意度会随着现场设备数目的增多而提高；然而当数量达到了某个阈值时，设备满意度反而会随着现场设备数目的增加而降低。这是因为当许多设备竞争无线通信资源和计算资源来卸载其任务时，发送到边缘服务器或者云服务器的开销会增大，导致所有设备的满意度降低。并且 3 种算法中，本文提出的 DEDSTO 算法一直保持最高的综合满意度，证明了该算法的有效性，且具备一定的优势。

4.2.2 云服务器最大计算能力变化对设备满意度的影响

图 7 给出了不同的云服务器最大计算能力对设备综合满意度的影响。DEDSTO-NS 算法的设备综合满意度没有受到云服务器最大计算能力的影响，其余 3 种算法中本文提出的 DEDSTO 算法的设备综合满意度始终保持最高。3 种算法的设备综合满意度会随着云服务器最大计算能力的增大而提高，因为云计算服务器最大计算能力的提高会导致现场设备的卸载操作获益更多，从而使时延降低，计算任务的安全性提高。

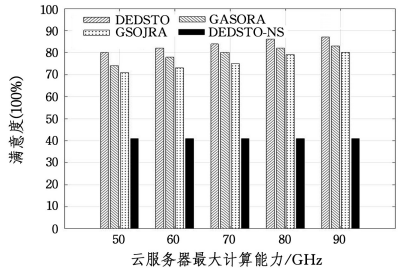

图 7 不同的云服务器最大计算能力对系统性能的影响

Fig. 7 Impact of different maximum computing power of cloud servers on system performance

4.2.3 不同安全要求对设备满意度的影响

图 8 给出了所有算法的安全要求 *Sd* 与满意度之间的关系。设备满意度是由时延要求和安全风险概率来决定的，当安全要求小于 0.5 时，除 DEDSTO-NS 外，其他算法的满意度曲线呈下降趋势。这是因为此时时间开销会随着安全要求的提高而增大，导致部分设备无法满足时延要求。当安全要求超过 0.5 之后，安全时间开销逐步增大，安全风险概率逐渐降低，由于两者之间存在相互制约的关系，因此设备满意度趋近于一个稳定值。DEDSTO-NS 算法的满意度曲线波动不大且几乎保持在 43%左右，这表明 DEDSTO-NS 算法与安全要求 *Sd* 无关。

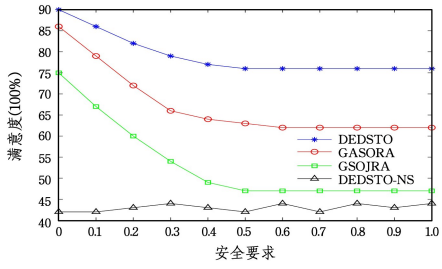

图 8 不同安全要求下设备综合满意度

Fig. 8 Comprehensive equipment satisfaction under different security requirements

图 9 给出了安全要求 *Sd* 与风险概率之间的关系，即对于确定的安全策略，越高的安全要求 *Sd* 会带来更高的安全风险概率。因此对于具有较高安全要求 *Sd* 的系统应使用更高等级的安全服务，以此来保护计算任务的数据安全。

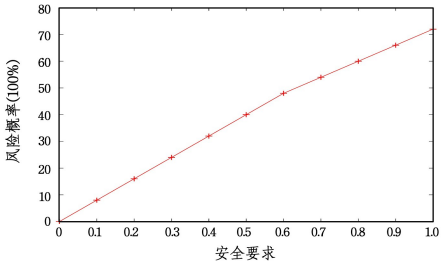

图 9 安全要求与风险概率的关系

Fig. 9 Relationship between security requirements and risk probability

4.2.4 有无安全服务的系统性能对比

假设实验场景有 100 个现场设备、10 个边缘服务器和 1 个云计算中心，各方面硬件条件都保持一致，只有一个区别，即是否应用安全服务。经过仿真统计分析得到有无安全服务的系统性能对比，如图 10 所示。

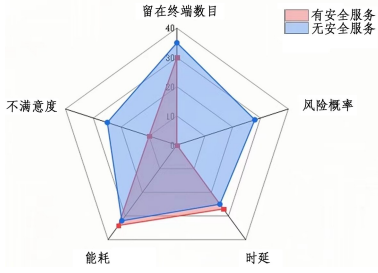

图 10 有无安全服务的系统性能对比图

Fig. 10 Comparison of system performance with and without security services

由于安全服务会产生额外的时延和能量消耗，因此具备安全服务的情境下产生的时延和能耗会比无安全服务的情况多，但是多出的时延和能耗十分有限，不影响智能制造车间的正常工作。应用安全服务的系统模型具有更高的设备综合满意度，现场设备产生的计算任务会倾向于卸载到边缘服务器和云服务器上，并且计算任务在卸载过程中所面临的风险

概率几乎降到了零。虽然添加安全服务的系统在时延和能耗方面表现稍差，但其在保障数据安全性方面表现突出，可以有效地防止因工业制造数据丢失而带来的巨大损失，因此在智能制造系统架构中添加安全服务非常必要。

结束语　针对云边协同架构所面临的数据安全挑战，本文引入了多级安全策略作为工业应用程序的保护层，保护数据在卸载过程中免受网络攻击，提出了一种基于改进差分进化策略的数据安全卸载算法，然后对差分进化算法的初始化步骤和变异操作做出一定的改进并应用于该优化问题的求解。仿真结果表明，本文提出的算法相较于其他 3 种基准算法可以让更多的现场设备满足响应时延限制和风险概率要求，使系统所有现场设备满意度平均提高了 35%，相比无安全服务的策略，本文提出的 DEDSTO 算法可以有效地保障数据安全性。本文的创新点在于在原有任务卸载的基础上添加了安全策略，并且创造性地将安全策略作为决策变量之一融入优化问题中。在实际场景中，计算任务的执行时间和传输速率可能会因为网络环境的不确定性和突如其来的延迟问题而实时发生变化，未来将随机优化和优化算法相结合，综合考虑系统不确定性，以此来应对时变的调度问题。

参 考 文 献

[1] TALEB T, SAMDANIS K, MADA B, et al. On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration[J]. IEEE Communications Surveys and Tutorials, 2017, 19(3): 1657-1681.

[2] YIN Y, CAO Z, XU Y, et al. QoS prediction for service recommendation with features learning in mobile edge computing environment[J]. IEEE Transactions on Cognitive Communications and Networking, 2020, 6(4): 1136-1145.

[3] FANG W, DING S, LI Y, et al. OKRA: Optimal task and resource allocation for energy minimization in mobile edge computing systems[J]. Wireless Networks, 2019, 25(5): 2851-2867.

[4] ELGENDY I A, ZHANG W, TIAN Y C, et al. Resource allocation and computation offloading with data security for mobile edge computing [J]. Future Generation Computer Systems, 2019, 100: 531-541.

[5] ELGENDY I A, MUTHANNA A, HAMMOUDEH M, et al. Advanced deep learning for resource allocation and security aware data offloading in industrial mobile edge computing[J]. Big Data, 2021, 9(4): 265-278.

[6] SONG Y B, JIN X Y, YAN F, et al. Secure and energy-efficient offloading strategies for mobile edge computing in vehicular networking[J]. Journal of Tsinghua University (Natural Science Edition), 2021, 61(11): 1246-1253.

[7] JIANG X, SUN Y, LIU B, et al. Combinatorial double auction for resource allocation with differential privacy in edge computing[J]. Computer Communications, 2022, 185: 13-22.

[8] YAO Y, WANG Z, ZHOU P. Privacy-preserving and energy efficient task offloading for collaborative mobile computing in IoT: an ADMM approach[J]. Computers and Security, 2020, 96: 101886.

[9] ZHOU Y S, TAN C, TANG F. A multidimensional security query scheme for fog-enhanced industrial Internet of Things[J]. Journal of Communication, 2020, 41(8): 175-186.

[10] REN T Y, WANG X H, GUO G X, et al. Design of data security system for power IoT based on multi-level authentication and lightweight encryption[J]. Journal of Nanjing University of Posts and Telecommunications (Natural Science Edition), 2020, 40(6): 12-19.

[11] HE X L, REN Z Y, SHI C H, et al. Cloud-fog network for medical big data and its distributed computing scheme[J]. Journal of Xi'an Jiaotong University, 2016, 50(10): 71-77.

[12] LI Z, CHANG V, HU H, et al. Profit maximization for security-aware task offloading in edge-cloud environment[J]. Journal of Parallel and Distributed Computing, 2021, 157: 43-55.

[13] ZHANG Y, LIU Y, ZHOU J, et al. Slow-movement particle swarm optimization algorithms for scheduling security-critical tasks in resource-limited mobile edge computing[J]. Future Generation Computer Systems, 2020, 112: 148-161.

[14] REN J, HE Y, YU G, et al. Joint communication and computation resource allocation for cloud-edge collaborative System [C]// 2019 IEEE Wireless Communications and Networking Conference. 2019: 1-6.

[15] JIANG W, ZHANG X, MA Y. Energy aware real-time scheduling policy with guaranteed security protection[C]// 19th Asia and South Pacific Design Automation Conference. 2014: 317-322.

[16] SINGHAL N, RAINA J. Comparative analysis of AES and RC4 algorithms for better utilization[J]. International Journal of Computer Trends and Technology, 2011, 2(6): 177-181.

[17] TRAD A, BAHATTAB A A, OTHMAN S B. Performance trade-offs of encryption algorithms for wireless sensor networks [C]// 2014 World Congress on Computer Applications and Information Systems (WCCAIS). IEEE, 2014: 1-6.

[18] ALAM M, KHAN M. Performance and efficiency analysis of different block cipher algorithms of symmetric key cryptography [J]. International Journal of Advanced Research in Computer Science and Software Engineering, 2013, 3(10): 713-720.

[19] ZHU A, WEN Y. Computing offloading strategy using improved genetic algorithm in mobile edge computing system[J]. Journal of Grid Computing, 2021, 19(3): 38.

[20] SARVABHATLA M, KONDA S, VORUGUNTI C S, et al. A network aware energy efficient offloading algorithm for mobile cloud computing over 5G network[C]// 2017 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM). IEEE, 2017: 69-74.

WANG Biao, born in 1969, Ph.D professor. His main research interests include analysis and optimization of complex networks and multi-agent control.

KE Ji, born in 1982, Ph.D, lecturer. His main research interests include analysis and control of complex networks, edge computing, hybrid feedback control and energy management.

（责任编辑：何杨）