

面向工业物联网的移动边缘计算任务卸载与资源分配

Research on Mobile Edge Computation Offloading Strategy for Industrial Internet of Things

蔡振启 李志军 (北方工业大学现场总线技术及自动化北京市重点实验室, 北京 100114)

摘要:针对工业物联网中终端设备计算资源和通信频谱资源难以满足时延敏感应用的业务需求,提出了一种部分计算卸载策略。首先构建物联网终端任务执行时延最小化的优化问题;然后利用自适应粒子群算法和拉格朗日乘子对终端的任务卸载比例和频谱资源分配进行求解;最后,通过仿真实验表明,该部分卸载策略能有效降低任务执行时延。

关键词:工业物联网;计算卸载;资源分配;自适应粒子群算法

Abstract:A partial computing offloading strategy is proposed in this paper to solve the problem that computing resources of terminal devices are difficult for time-sensitive applications in the industrial Internet of Things (IIoT) and that the allocation of communication spectrum resources is unreasonable. Firstly, the minimum delay model of unloading is constructed, and then the task unloading ratio and spectrum resource allocation are optimized based on the adaptive particle swarm optimization algorithm and Lagrange multiplier. Finally, the simulation results show that the partial unloading strategy can effectively reduce the task execution delay.

Keywords:Industrial Internet of Things, computation offloading, resource allocation, APSO

物联网技术是一种通过无处不在的网络将众多智能终端设备转化为网络连接设备的技术。近年来,物联网技术已经应用于许多工业场景,如智能制造和电子行业。随着无线通信网络技术和计算技术的发展,工业物联网包括越来越多的现场设备和智能终端。因此,这些设备和终端不断产生大量的工业数据,这些数据大部分被作为监测和采集并用于实时反馈控制^[1]。需要及时的传输、处理和存储。然而,如今大多物联网终端计算资源有限,无法满足对延迟敏感应用的要求。边缘计算作为云计算的补充和扩展,在工业互联网领域受到了广泛的关注。边缘服务器部署在基站的边缘,可以为物联网设备提供丰富的计算资源,又可以避免远距离网络通信中的传输延迟。为了满足延迟敏感应用的要求,计算资源有限的设备可以通过无线网络将部分计算任务转移到边缘服务器上执行。大量的物联网设备通过工业无线网络传输将数据传输到边缘计算层进行实时处理和存储,这导致了网络频谱资源和资源有限的边缘服务器的压力增加。因此,如何合理地进行终端设备和边缘服务器的协作以及频谱资源的合理分配是一个重要的问题,需要考虑物联网终端和边缘服务器上的任务可以并行执行。文献[2]针对延迟敏感应用采用物联网终端的部分或全部卸载,此外考虑到物联网终端和边缘服务器之间有限的可用带宽,部分卸载是非常实用的;文献[3]针对云计算和边缘计算的联合的卸载问题,采用 NSGA-III 方法却低估动态任务卸载,实现移动设备能耗和执行时间的联合优化;文献[4]通过优化系统的传输时间和计算卸载,使无线设备的计算率加权和最大,提出了一种基于乘法器的交替方向分解方法,以获得接近最优的性能;文献[5]提出了无线蜂窝网络中资源分配、计算卸载和内容缓存的联合优化问题,将其转化为一个凸问题,利用分布式凸算法对其进行分解和有效求解。

针对时延敏感类型终端的应用,本文通过考虑物联网终端的最大能量,上行链路的频谱资源分配,对于支持数据分区的计算任务,为充分降低系统执行时延,提出了一种新颖的部分计算卸载方法。

1 系统模型及问题描述

1.1 网络模型

如图 1 为工业物联网计算卸载的网络模型。假设该网络中

包括一个网络接入点(AC),并且在网络接入点上部署一个边缘服务器,网络中分布着 M 台无线现场设备,表示为 $\varnothing=\{1, \cdots, M\}$,假设每个现场设备都有以及计算任务且该任务支持数据分区;令终端设备 m 的计算任务为 $J_m=(l_m, C_m)$,其中, l_m 表示设备 m ($1 \leq m \leq M$) 的输入数据的大小, C_m 表示执行 l_m 大小的数据所需要的 CPU 周期数,根据文献[6]可得 $C_m=a_m l_m$, $a_m > 0$ 为常量,由终端设备计算复杂度决定, $x_m, x_m \in [0, 1]$ 代表设备 m 卸载到边缘服务器上执行的数据比例,其中, $x_m=0$ 表示任务全部在本地设备上执行, $x_m=1$ 表示任务全部卸载到边缘服务器上执行。

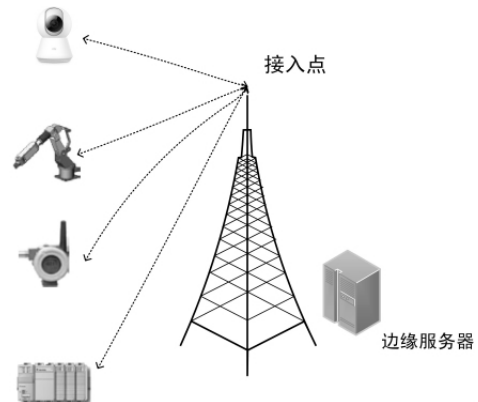


图 1 计算卸载网络模型

1.2 终端计算模型

在本文中,设备 m 的计算任务可以在全部在本地执行或者部分本地执行,当设备 m 的计算任务完全在本地执行时,这里用 f_m^l 代表设备 m 的计算能力,每秒/CPU 周期数。如果设备 m 的计算任务全部在本地执行,则执行时间和计算能耗为:

$$t_m = \frac{a_m l_m}{f_m} \quad (1)$$

当物联网设备执行计算任务 J_m ,若采用部分方式时,任务 J_m 在本地设备计算时间为:

$$t_m(x_m) = \frac{a_m l_m (1-x_m)}{f_m} \quad (2)$$

则相应的本地计算能耗为:

$$e_m(x_m) = e_m a_m l_m (1 - x_m) \quad (3)$$

1.3 通信模型

本节介绍移动边缘计算环境下无线接入方式的通信模型,考虑移动蜂窝网络下物联网终端基于 OFDMA 技术进行通信。物联网终端设备将自身的计算任务卸载到计算资源更丰富且无能耗约束的边缘服务器上,来减小任务执行时延和节省本省能耗,与此同时,任务卸载的过程将花费额外的传输时延和传输能耗。通信过程主要包括计算任务上传阶段、任务执行阶段和结果返回阶段。通常情况下,下行链路的速度远大于上行链路且下行数据比上行数据小得多,所以,本文下行链路的传输时间忽略不计。对于 OFDMA 机制来说,由于分配专一的子载波,可以忽略信道间的干扰,所以上行数据传输速率为:

$$r_m = \mu_m W \log_2 \left(1 + \frac{g_m p_m}{\sigma^2} \right) \quad (4)$$

其中, W 表示上行链路的带宽; σ^2 表示高斯白噪声; g_m 表示设备 m 和边缘服务器之间的信道增益; p_m 为终端设备 m 的传输功率且有物联网终端设备分配,并定义设备 m 最大传输功率为 p_m^{\max} , 满足:

$$0 < p_m \leq p_m^{\max} \quad (5)$$

μ_m 表示设备 m 上行链路带宽的比例, 满足:

$$\sum_{m=1}^M \mu_m = 1 \quad (6)$$

对于终端设备数, 有 $l_m x_m$ 大小的数据卸载到边缘服务器上执行, 则数据传输的时间为:

$$t_m^{\text{tr}}(x_m) = \frac{l_m x_m}{r_m} \quad (7)$$

则相应终端设备 m 的传输功耗为:

$$e_m^{\text{tr}}(x_m) = \frac{p_m l_m x_m}{r_m} \quad (8)$$

1.4 边缘计算模型

考虑实际应用中边缘服务器有电网直接供电, 因此不考虑边缘服务器的能耗问题, 令 τ_m^s 表示终端 m 卸载到边缘服务器上计算任务的执行时间为:

$$\tau_m^s = \frac{a_m l_m x_m}{f^c} \quad (9)$$

边缘服务器完成所有设备的计算任务的时延为:

$$T_e = \sum_{m=1}^M \tau_m^s \quad (10)$$

其中, f^c 为边缘服务器的计频率(周期/s), 假设在数据处理的过程中 f^c 固定不变。边缘端的任务执行的总时延为:

$$t_m^c = T_e + t_m^{\text{tr}} \quad (11)$$

1.5 问题描述

综合以上模型和时延敏感的应用, 本文提出了一种完成所有任务延迟最小的优化问题, 该问题对频谱资源分配和卸载比例进行优化。支持数据分区任务可以在物联网终端和边缘服务器并行执行, 则完成终端设备一个任务的总时延为:

$$\max \{t_m^{\text{tr}}, t_m^c\} \quad (12)$$

则物联网终端时延问题可被描述为:

$$\min_{x_m, \mu_m} \text{delay} = \frac{t_m^{\text{tr}} + t_m^c}{2} \quad \text{s.t.}$$

$$0 \leq x_m \leq 1 \quad (14)$$

$$0 \leq \mu_m \leq 1 \quad (15)$$

$$\sum_{m=1}^M \mu_m = 1 \quad (16)$$

$$0 \leq p_m \leq p_m^{\max} \quad (17)$$

$$0 \leq f_m \leq f_{m, \max} \quad (18)$$

$$0 \leq f_s \leq f_{s, \max} \quad (19)$$

$$e_m^{\text{MD}}(x_m) + e_m^{\text{tr}}(x_m) \leq L_m \quad (20)$$

其中, L_m 为最大能耗限制, 从通信模型和计算模型可以看出, 采用全部卸载、信道均匀分割等方式并不是最优的, 针对不同设备不同卸载任务量合理的分配带宽和卸载比例, 对整个系统时延优化更有利。约束(14)表示任务卸载比例在 0 到 1 之间, 约束(15)和(16)表示频谱资源分配比例小于 1, 且总频谱比例不应该大于 1。约束(17)、(18)、(19)表示终端设备的功率和频率以及边缘服务器的频率不应该大于设备最大频率; 约束(20)表示终端能耗小于最大能耗。

2 问题分解及算法描述

由于优化问题的中卸载决策和频谱资源分配变量之间的耦合, 因此若采用穷举法求解, 则求解的复杂度成指数级增长。因此为了降低求解问题的难度, 本文提出基于自适应粒子群的联合优化算法, 将原问题解耦成两个子问题进行求解, 即任务卸载子问题和频谱资源分配问题。

2.1 频谱资源分配

在给定计算任务卸载比例的前提下, 可以将任务执行时延最小化问题转化为频谱资源分配子问题。

$$P1: \min_{\mu_m} \sum_{m=1}^M \frac{l_m x_m}{\mu_m W \log_2 \left(1 + \frac{g_m p_m}{\sigma^2} \right)} \quad (21)$$

s.t. (2)(3)

问题(15)为一个关于变量 μ_m 凸函数, 可以得到问题(14)的拉格朗日函数如下:

$$L(K, U, \lambda) = \sum_{m=1}^M \frac{l_m x_m}{\mu_m W \log_2 \left(1 + \frac{g_m p_m}{\sigma^2} \right)} + u \left(\sum_{m=1}^M \mu_m - 1 \right) - \sum_{m=1}^M \lambda_m \mu_m \quad (22)$$

其中, u 和 λ_m 为拉格朗日乘子, 如果问题(15)有一个最优解 μ_m^* , 且 μ_m^* 满足 KKT 条件, 可以得到:

$$\frac{\partial L}{\partial \mu_m^*} = - \frac{l_m x_m}{\mu_m^{*2} W \log_2 \left(1 + \frac{g_m p_m}{\sigma^2} \right)} + u - \lambda_m \quad (23)$$

$$u \left(\sum_{m=1}^M \mu_m - 1 \right) = 0 \quad (24)$$

$$\lambda_m \mu_m^* = 0 \quad (25)$$

通过求解上式, 可以得到频谱资源的最优分配如下:

$$\mu_m^* = \frac{\sqrt{l_m x_m}}{\sum_{m=1}^M \sqrt{l_m x_m}} \quad (26)$$

2.2 基于自适应粒子群的联合优化算法

粒子群算法受鸟类捕食行为的启发并对这种行为进行模仿, 将优化问题的搜索空间类比为鸟群的飞行空间, 将每一只鸟抽象为一个粒子, 每个粒子可以视为 N 维搜索空间的一个搜索个体, 粒子当前位置即为对应优化问题的一个候选解, 粒子的飞行过程即为该个体的搜索过程, 粒子的分型速度可根据粒子历史最优位置和种群历史最优位置进行动态调整, 粒子仅有两个

属性:速度和位置,速度代表移动的快慢,位置代表移动的方向,每个粒子单独搜索的最优解叫做个体极值,粒子群中最优的个体极值作为当前全局最优解^[6]。不断迭代更新速度和位置,最终得到满足终止条件的最优解。

对于物联网终端的卸载决策及卸载比例和频谱资源分配的组合问题,通过对粒子群算法进行改进,求解最优值。首先,该算法在每次迭代时只需对任务卸载策略进行更新,同时以本次更新的卸载决策为基础,利用拉格朗日乘子法对通信频谱资源进行分配,从而提高了新解更新的可行性可频谱资源分配的合理性;最后,根据上述的改进措施,本节基于自适应粒子群算法的联合优化卸载策略和资源分配流程如图2。

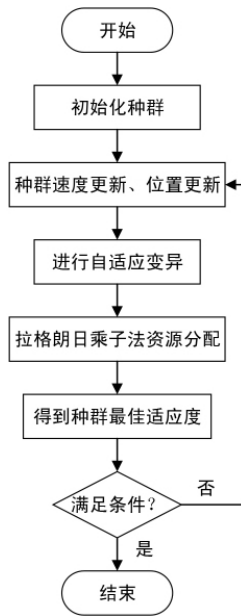


图2 粒子群优化流程图

3 仿真及结果分析

3.1 仿真参数设置

根据网络模型,本文仿真场景中设备随机分布在网络接入点附近,网络通信接入点的覆盖范围为300m。如表1所示。

表1 仿真参数意义及取值

仿真参数	参数设置
终端任务数 M_{bit}	5
终端设备常数 a_m	100
终端计算能力 f_m^c	$4 \times 10^8 \text{ cycle/sec}$
终端功率 p_m	0.8w
终端最大计算能力 f_m^{max}	$4 \times 10^8 \text{ cycle/sec}$
通信带宽 W	10MHz
高斯噪声功率 σ^2	1×10^{-9}
信道增益 g_m	$128.1 + 37.6 * \log_{10}(d_{km})$
服务器计算能力 f_m^c	$8 \times 10^8 \text{ cycle/sec}$
边缘服务器常数 a_e	100
最大能耗 L_m	1.8J

3.2 仿真结果分析

根据参数表的参数取值进行仿真时延,本文基于自适应粒子群算法和拉格朗日乘子法对任务卸载决策和频谱资源分配进行优化。首先,对本文所提算法的收敛性进行验证,如图3所示,当目标函数所有物联网终端任务执行时延方差之和随迭代次数的增大而逐渐减小,并且在迭代次数为40左右时达到稳定,算

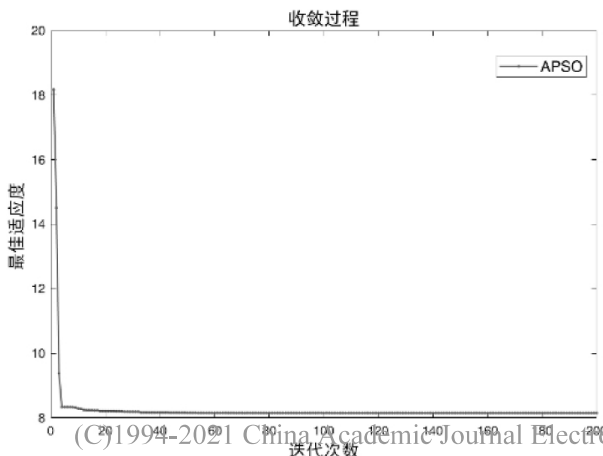


图3 收敛性过程

法收敛。

如图4所示为在设备个数不同的条件下,本文所提的自适应粒子群算法优化的部分卸载与任务全部本地执行以及任务全部边缘执行三种情况下的执行时延对比,通过对比可得物联网终端任务采用部分卸载方式能够有效降低任务执行时延;并且可以得到随着设备数量的增多,采用部分卸载方式和全部卸载方式的总任务执行时延增加,但远小于本地执行。

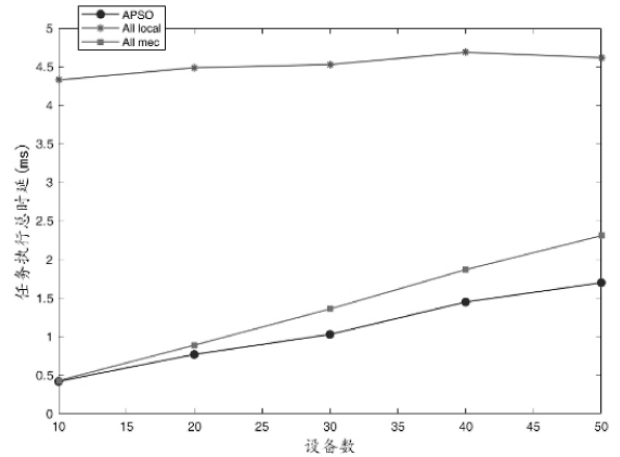


图4 系统时延与设备数

如图5所示为不同数目物联网终端数目下,总任务量、卸载任务量和本地剩余任务量。当设备数量为10时,由于边缘服务器强大的计算能力,物联网终端更倾向于将全部的任务量卸载到边缘服务器执行,随着设备数量的增多,边缘服务器处理能力下降,将逐渐增大物联网终端上任务执行的比例。

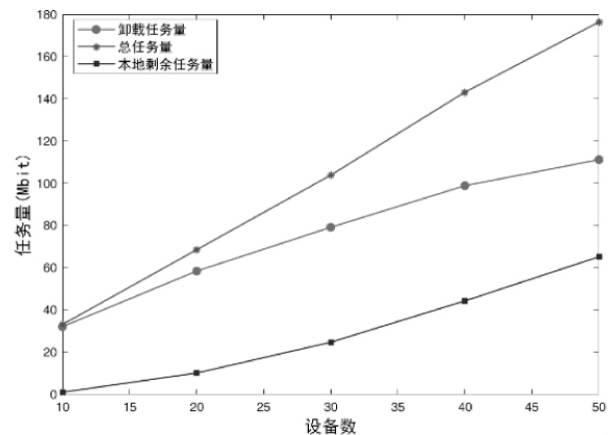


图5 任务量与设备数

4 结束语

本文针对工业物联网中物联网终端难以满足时延敏感的应用程序的要求,并且考虑在有限频谱资源的条件下,提出了一种基于自适应粒子群的任务卸载策略和频谱资源分配策略。首先,构造根据计算任务执行过程,构造计算卸载的物联网终端时延模型、通信模型以及边缘服务器时延模型;然后基于以上模型提出物联网终端任务执行时延最小化优化问题;最后,基于自适应粒子群算法和拉格朗日乘子法对任务进行卸载和资源分配。仿真结果表明本文的算法能够有效减少任务执行时延。

参考文献

- [1] B. Chen, J. Wan, L. Shu, et al. Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges [J]. IEEE Access, 2018:6505-6519

(下转第54页)

$$f_{out} = f_M / 2^{28} \times \text{FREQREG} \quad (1)$$

$$p_{out} = 2\pi / 4096 \times \text{PHASEREG} \quad (2)$$

其中 f_{out} 为输出信号频率, f_M 为时钟频率, FREQREG 为频率寄存器写入的值, p_{out} 为相移输出, PHASEREG 为相位寄存器写入的值。

本设计采用 25MHz 晶振, 当频率输出为 1MHz 时, FREQREG 的值为 0xA3D70A, 频率寄存器的低 14 位需配置为 0x107A, 高 14 位写入 0x28F。AD9833 初始化程序框图如图 3 所示。

3.2 ESP32 低功耗蓝牙模块程序设计

ESP32 的低功耗蓝牙模块设计用于接收手机终端发送来的数据, 主要包含 ESP32 低功耗蓝牙初始化及与手机终端通信流程设计。低功耗蓝牙初始化主要包括低功耗蓝牙协议栈初始化、广播参数配置、GATT sever 创建及其回调函数初始化、BLE 服务创建、Characteristic 创建及其回调函数初始化。

ESP32 的低功耗蓝牙初始化完成后, 将开始广播等待连接。连接建立后手机终端通过写特征值的方式, 将 AD9833 控制数据下发到 ESP32, 图 4 是 ESP32 与手机终端的通信流程图。

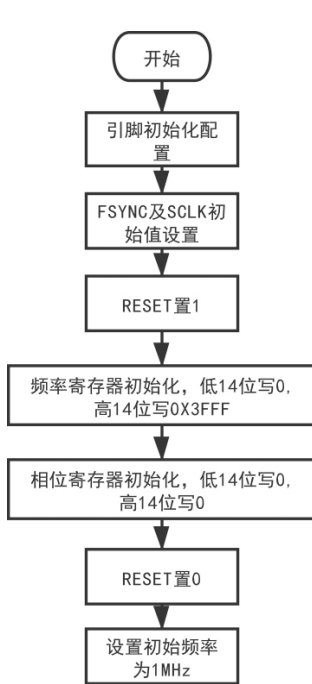


图 3 AD9833 初始化配置

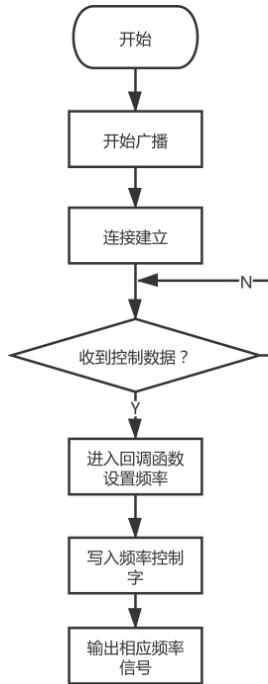


图 4 ESP32 与手机终端的通信流程

4 实验结果与分析

实验是通过 BLE 调试助手 APP 来进行控制测试。ESP32 上电后开始广播, 广播名称为“AD9833”, 使用 APP 根据广播名称进行连接, 连接建立后, 通过写特征值的方式, 在手机端写入

要设置的频率, ESP32 收到数据解析后再控制 AD9833 输出相应的频率信号。分别在手机端输入 1000、10000、100000、1000000 后, AD9833 输出的信号波形通过示波器显示如图 5 所示。

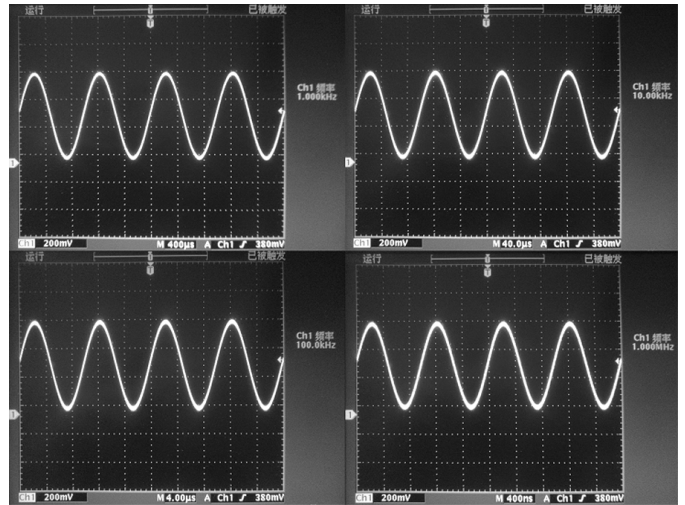


图 5 不同频率输出波形图

经实验证明, 该设计可以实现手机终端控制信号源在 0~2MHz 频带内输出光滑、稳定的正弦信号。

5 结束语

本文提出了基于低功耗蓝牙无线控制的信号发生器的设计方法, 并进行了频率控制测试。测试结果显示该系统可以实现对信号源频率输出进行稳定的远程控制, 对工业应用具有一定参考价值。

参考文献

- [1] 崔赞, 孙志毅, 谢嘉麟. 基于 STC89C52 单片机信号发生器的设计[J]. 工业控制计算机, 2019, 32(6): 127-128, 130
- [2] 张萍. 基于 DDS 的低频信号发生器的设计[J]. 自动化技术与应用, 2016, 35(6): 106-109
- [3] 王青, 杨延军, 王志军, 等. 基于 Arduino 的简易信号源实验设计[J]. 实验技术与管理, 2020, 37(8): 46-49
- [4] 张铮, 胡婷. 基于 AD9854 的多功能高精度 DDS 信号源设计[J]. 电子科技, 2017, 30(3): 182-184
- [5] 常乐. 基于 DDS 的低频信号发生器的设计研究[J]. 电子测试, 2019(24): 10-11
- [6] 张萍. 基于 DDS 数字信号发生器的设计[J]. 信息化研究, 2016, 42(5): 66-69
- [7] 石志京, 徐铁峰, 刘太君, 等. 基于 ARM、DDS 和蓝牙技术的信号发生器设计[J]. 无线电通信技术, 2014, 40(5): 86-88

[收稿日期: 2021.4.18]

(上接第 52 页)

- [2] M Salmani, T N Davidson. Energy -Optimal Multiple Access Computation Offloading: Signalling Structure and Efficient Communication Resource Allocation 2[J]. IEEE Transactions on Signal Processing, 2020, 68: 1646-1661
- [3] Y Dai, D Xu, S Maharjan, et al. Joint Computation Offloading and User Association in Multi-Task Mobile Edge Computing[J]. IEEE Transactions on Cloud Computing, 2018, 6(7): 12313-12325
- [4] X Xu, Q Liu, Y Luo, et al. A Computation Offloading Method

over Big Data for IoT-enabled Cloud-edge Computing[J]. Future Generation Computer Systems, 2019, 67(12): 522-533

- [5] S Bi, Y J Zhang. Computation Rate Maximization for Wireless Powered Mobile-Edge Computing With Binary Computation Offloading[J]. IEEE Transactions on Wireless Communications, 2018, 17(6): 4177-4190
- [6] 包子阳, 余继周, 杨杉. 智能优化算法及其 MATLAB 实例[M]. 2 版. 北京: 电子工业出版社, 2018: 112-134. <http://www.cnki.net>

[收稿日期: 2021.3.6]