

基于路径计算方法的 WSN 时延优化研究

朱鹏¹, 任继军¹, 任智源²

(1.西安邮电大学 通信与信息工程学院, 陕西 西安 710121;
2.西安电子科技大学 综合业务网理论及关键技术国家重点实验室, 陕西 西安 710071)

摘 要:随着物联网(IoT)行业的快速发展,无线传感器网络(WSN)融合云计算技术面临着任务处理时延高、传感器节点能量有限的挑战。因此,提出了一种基于云雾网络架构的路径计算方法,利用雾计算层的网络边缘设备计算资源,将 WSN 监测任务合理地部署到指定边缘设备上完成处理,以减少能耗制约下的任务处理时延。为了将任务有效地分配到雾计算层,采用了一种任务映射规则,将有向无环图表示的监测任务映射到无向图表示的雾计算层网络;结合时延和能耗约束建立了一个关于寻求最优映射关系的二值优化问题;采用模拟退火-离散二值粒子群优化(SA-BPSO)算法实现了对该优化问题的求解。仿真结果显示,在数据量为 10 Mb 时,该方法的时延性能相比较 WSN 融合云计算技术提高了约 40%。

关 键 词:无线传感器网络;路径计算;云雾网络架构;模拟退火-离散二值粒子群优化

中图分类号:TN915 **文献标志码:**A **文章编号:**1000-2758(2022)06-1394-10

无线传感器网络(wireless sensor network, WSN)对整个物联网(internet of things, IoT)行业的发展起着非常重要的作用,是实现物联网应用的关键技术,其应用领域十分广泛,包括智慧城市、预警灾害、精细农业以及军事应用^[1]。一般而言,在这些应用中 WSN 的主要作用是通过大量传感器节点捕获和发送关于监测区域内感知对象的信息,并使用多跳、自组织无线移动通信网络将这一信息传送给用户,以使用户能够及时了解感知对象的变化。

目前,WSN 融合云计算技术因其强大的任务处理能力和存储能力,已在各领域得到充分应用。比如:由 WSN 组成的智能交通系统,通过中心云处理实时大数据,实现车流量的管控;由 WSN 组成的环境监测系统,通过大量传感器收集环境数据,经大数据处理和分析后,达到污染治理和预防自然灾害的目的。然而,这项技术要求将 WSN 收集的数据上传到云中心进行处理,远程数据传输往往会产生较高的传输时延,同时部署在环境复杂的 WSN 存在着能量供应问题,因此无法用于监测具有低时延和低能

耗要求的任务^[2]。

为了解决云计算模式存在的问题,国内外许多研究人员在实现数据的低时延和低功耗传输方面开展了诸多研究^[3]。在时延优化的研究中,学者们提出了一种路径式协同计算方法,利用分布在网络边缘的设备集群进行多点协同计算,使数据在传输的过程中得到处理。Shukla 等^[4]为了改变通信网络路径计算速率缓慢的状况,使用有向无环图表示计算任务,同时提出一个函数流模型和线性程序公式,以此确定最大化计算速率,但未将计算节点的处理能力考虑进去。在能耗优化的研究中,牛祺君等^[5]针对 WSN 能耗受限的问题,采用了按照 WSN 节点剩余能量的簇首轮询机制,提出了一种依靠蜂群算法进行分簇的层次路由选择算法,解决了节点能耗过快的问题,延长了 WSN 寿命。虽然该能耗优化方法可以在一定程度上降低 WSN 的能耗,但是没有考虑到节点在空闲时的能耗。

针对上述存在的问题,本文研究了一种云雾网络架构,并基于此架构设计了一种路径计算方法。

收稿日期:2022-03-25

基金项目:陕西省重点研发计划(2021GY-100)资助

作者简介:朱鹏(1999—),西安邮电大学硕士研究生,主要从事边缘计算研究。

通信作者:任继军(1980—),西安邮电大学高级工程师、硕士生导师,主要从事软件无线电研究。e-mail:renjijun@xupt.edu.cn。

首先,该方法使用有向无环图(DAG)表示 WSN 的监测任务,即中心云平台利用 WSN 上传的实时数据进行大数据分析和处理的任务,同时利用无向图(UG)表示雾计算层的雾设备集群;其次,指定 DAG 至 UG 的任务映射规则,将监测任务分配给边缘网络设备,并依靠边缘网络设备的高算力实现协同任务处理;最后,为了求解 DAG 至 UG 的最优映射关系,建立了一个关于时延与能耗的二值优化问题,通过模拟退火-离散二值粒子群优化(SA-BPSO)算法得到问题的最优值^[6]。仿真结果表明,该路径计算方法可以在能耗约束下实现降低任务处理时延的目的,能够完成低时延和低能耗要求的监测任务。

1 云雾网络架构

为了发挥边缘网络设备的任务处理能力,实现路径计算,本节研究了一种云雾网络架构,该架构从上至下有 3 层,分别为云计算层、雾计算层和感知层^[7],如图 1 所示。

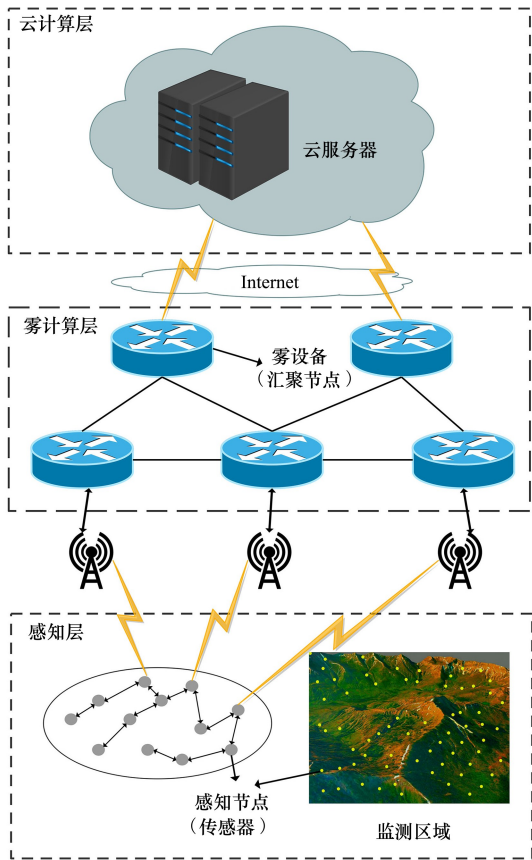


图 1 云雾网络架构

在该网络架构中,感知层主要由传感器节点和无线链路连接的智能终端组成,通常用于数据采集和传输。雾计算层与感知层相连,由计算和存储能力较弱的边缘网络设备(汇聚节点)组成,汇聚节点之间通过无线链路连接。当云计算中心下发的监测任务到达数据传输路径中的汇聚节点后,监测数据会被卸载到该节点上实现任务的协同计算,使数据在传输过程中完成处理,并将数据的处理结果发送给管理用户。云计算层与雾计算层相连,由存储和处理能力十分优越的云服务集群组成,主要负责管理和监测 WSN,同时需要按照汇聚节点的任务处理能力、链路状况等信息制定监测任务的调度规则,并将该规则发放到具体的汇聚节点。

2 任务映射规则

为了实现将 WSN 的监测任务从云中心迁移到汇聚节点,首先需要将具体的监测任务划分成多个子任务。由于子任务之间存在数据依赖关系和处理顺序的优先级,本文使用 DAG 表示它们,其中图的有向边集可以表示子任务之间的这种依赖关系,图中的箭头指向能够表示子任务之间的优先权约束,这也意味着某个任务在其前面任务处理完成之前不能开始执行。同时,图中的无环代表任务流程的方向,即总任务的执行需要通过若干个子任务逐步执行,最后汇总到任务终点,不能出现折返式处理子任务的情况。为了将 DAG 表示的监测任务映射到 UG 表示的雾网络,本节介绍了具体的映射方法。

2.1 DAG 至 UG 的映射规则

本文通过有向无环图 $D = (\Omega, \Gamma)$ 表示监测任务模型,并使用 $\Omega = \{n_1, n_2, \dots, n_s, n_{s+1}, n_{s+2}, \dots, n_{t-1}, n_t \mid s \geq 1, t \geq s + 1\}$ 代表所有的监测子任务,其中 $n_1, n_2, \dots, n_s$ 表示监测任务起点, $n_{s+1}, n_{s+2}, \dots, n_{t-1}$ 表示中间监测任务节点, n_t 表示监测任务终点;定义 Γ 为图 D 的有向边集合, $\Psi_+(n_i) = \{n_j \mid (n_j, n_i) \in \Gamma\}$ 为子任务节点 n_i 的所有前向任务节点。

此外,本文通过无向图 $U = (V, S)$ 展现雾网络的拓扑关系,其中 $V = \{v_1, v_2, \dots, v_s, v_{s+1}, v_{s+2}, \dots, v_{m-1}, v_m \mid s \geq 1, m \geq s + 1\}$ 为图 U 中的所有子节点,即雾网络中的汇聚节点, $v_1, v_2, \dots, v_s$ 为任务起始节点, $v_{s+1}, v_{s+2}, \dots, v_{m-1}$ 为任务中转节点, v_m 为任务目标节点; S 为图 U 中的所有边。

如果图 U 中的每条边 S 都可实现节点到节点的

信息传输与交换,并且任何两节点之间最少有一条链路能够保持数据传输,则本文使用 $t_{v_i v_j}$ 表示 v_i 和 v_j 两节点间传送单位信息的最少时间;节点 v_i 到节点 v_j 的最短路径用 $P_{v_i v_j}$ 表示,定义 $\mathbf{P} = \{P_{v_i v_j} \mid v_i, v_j \in \mathbf{V}\}$ 为所有 $P_{v_i v_j}$ 的集合;定义 $\mathbf{N}_{v_i v_j} = \{N_{v_i v_j}^1, N_{v_i v_j}^2, \dots, N_{v_i v_j}^n\}$ 为两节点传输数据期间所经过的数据中转节点集合。以上参数可通过 Dijkstra 算法并结合图 $\mathbf{U}$ 中节点之间的连接关系和网络边权重 $R_{v_i v_j}$ 求出。

图 $\mathbf{D}$ 至图 $\mathbf{U}$ 的映射规则主要包含两部分:①将图 $\mathbf{D}$ 中的子任务节点映射至图 $\mathbf{U}$ 中的汇聚节点;②将有向边映射为图 $\mathbf{U}$ 中节点间的最短通路 $P_{v_i v_j}$ 。结合文献[8],具体映射规则如下:

1) 子任务映射规则

图 $\mathbf{D}$ 中的所有监测子任务 Ω 映射至图 $\mathbf{U}$ 中的汇聚节点 $\mathbf{V}$ 的规则为 γ , 如(1)式所示:

$$\gamma(n_i) = \begin{cases} v_i, & \forall i \in \{1, 2, \dots, s\} \\ v_m, & i = t \\ v_j, & \forall v_j \in \mathbf{V} \setminus \{v_1, v_2, \dots, v_s, v_m\} \end{cases} \quad (1)$$

γ 将 Ω 中的任务起点 $n_1, n_2, \dots, n_s$ 映射到 $\mathbf{V}$ 中的任务起始节点 $v_1, v_2, \dots, v_s$; 将中间子任务节点 $n_{s+1}, n_{s+2}, \dots, n_{t-1}$ 随机映射到任务中转节点 $v_{s+1}, v_{s+2}, \dots, v_{m-1}$; 将任务终点 n_t 映射到任务目标节点 v_m 。

同时,在子任务节点通过 γ 映射至汇聚节点时存在着多对多的映射关系,本文使用关系矩阵 $\mathbf{X}$ 来表示这种映射关系,矩阵 $\mathbf{X}$ 如(2)式所示

$$\mathbf{X} = \begin{pmatrix} x_{n_1 v_1} & \cdots & x_{n_1 v_m} \\ \vdots & & \vdots \\ x_{n_t v_1} & \cdots & x_{n_t v_m} \end{pmatrix} \quad (2)$$

$$x_{n_p v_q} \in \{0, 1\}, \forall n_p \in \Omega, \forall v_q \in \mathbf{V} \quad (3)$$

在(2)式中, t 和 m 分别表示监测子任务和汇聚节点的个数;(3)式中的 $x_{n_p v_q}$ 表示某个监测子任务到某个汇聚节点的映射情况,当 $x_{n_p v_q} = 1$ 时,说明子任务节点 n_p 可以映射至汇聚节点 v_q ; 当 $x_{n_p v_q} = 0$ 时,说明子任务节点 n_p 无法映射至汇聚节点 v_q 。

2) 有向边映射规则

图 $\mathbf{D}$ 中的集合 $\mathbf{F}$ 映射至图 $\mathbf{U}$ 中的集合 $\mathbf{P}$ 的规则为 $\mathbf{T}$, 如(4)式所示

$$\mathbf{T}(n_i, n_j) = \{P_{\gamma(n_i)\gamma(n_j)} \mid \gamma(n_i), \gamma(n_j) \in \mathbf{V}\} \quad (4)$$

$\mathbf{T}$ 将集合 $\mathbf{F}$ 中的有向边 (n_i, n_j) 映射到图 $\mathbf{U}$ 中的节点 $\gamma(n_i)$ 至节点 $\gamma(n_j)$ 的最短路径 $P_{\gamma(n_i)\gamma(n_j)}$ 。

2.2 映射规则的改进模型

对于相同的图 $\mathbf{D}$ 和图 $\mathbf{U}$, 如果遵循上述映射规则,将会有多个映射关系,从而对应于多种任务计算路径和任务处理时延。为了筛选出最小任务处理时延所对应的映射关系,从时延与能耗角度对 2.1 节的映射规则做出改进,同时建立一个关于任务处理时延的最优化问题模型。

在某次任务的映射关系中,子任务节点 n_i 的处理总时延等于进行到此任务节点的累积时延与完成此任务所用的计算时延之和^[9], 计算公式为

$$T_{\text{total}}(n_i) = T_{\text{accu}}(n_i) + T_{\text{calc}}(n_i) \quad (5)$$

因为要考虑任务节点与汇聚节点的映射关系,所以需要对公式(5)进行修改,其中 $T_{\text{accu}}(n_i)$ 与 $T_{\text{calc}}(n_i)$ 的具体表达式分别为:

$$T_{\text{accu}}(n_i) = \max_{n_j \in \Psi^+(n_i)} [T_{\text{total}}(n_j) + \sum_{p=1}^m \sum_{q=1}^m t_{v_p v_q} D_{n_j p_i} x_{n_j v_q} x_{n_i v_p}] \quad (6)$$

$$T_{\text{calc}}(n_i) = \sum_{p=1}^m \frac{D_{n_i} \alpha}{p_{v_p}} x_{n_i v_p} \quad (7)$$

在(6)式中, $D_{n_j p_i}$ 表示子任务节点 n_j 与 n_i 之间需要传递的数据量,根据子任务映射规则可知,它还表示图 $\mathbf{U}$ 中汇聚节点 $\gamma(n_j)$ 与 $\gamma(n_i)$ 之间需要传递的数据量; $x_{n_j v_q}$ 和 $x_{n_i v_p}$ 分别表示子任务节点 n_j 到汇聚节点 v_q 的映射关系和子任务节点 n_i 到汇聚节点 v_p 的映射关系。由于某一个具体的子任务节点与汇聚节点之间的映射关系是一对多,对于每个子任务节点的累积时延计算都需要遍历它们与所有的汇聚节点之间的映射关系并进行求和。同时,因为子任务节点可能存在多个前向任务节点,所以对应的会有多个累积时延,本文选择其中的最大值作为该子任务节点的累积时延。

在(7)式中, D_{n_i} 代表在此次映射中子任务节点 n_i 的计算数据量,通过子任务映射规则可知,它也代表图 $\mathbf{U}$ 中汇聚节点 $\gamma(n_i)$ 所需计算的数据量; α 为当前任务的难度系数; p_{v_p} 为图 $\mathbf{U}$ 中汇聚节点 v_p 的任务处理能力系数。考虑到单一子任务节点与汇聚节点的映射关系,在计算子任务的计算时延时需要遍历该节点与其他所有汇聚节点之间的映射关系并进行求和。

通过以上分析可知,模型 $\mathbf{D}$ 的任务总时延等于节点 n_i 的处理总时延,即

$$T(\mathbf{D}) = T_{\text{total}}(n_i) \quad (8)$$

同时,由于在处理总时延公式中变化的只有任务节点与汇聚节点的映射关系,因此图 D 的任务总时延可以表示为关于 X 的函数,如(9)式所示

$$T(D) = F(X) \quad (9)$$

尽管从时延优化方面可以进一步求解出最优映射关系,得到最小任务处理时延,可是考虑到 WSN 通常部署在野外,面临着能耗受限的问题,在求解最优映射关系时就必须将能耗约束考虑进去。此外,本文的工作在于如何将监测任务映射至雾网络中的汇聚节点和优化映射规则,以此改善 WSN 融合云计算的时延性能。因此,接下来在时延优化的基础上,

主要从雾网络的总能耗要求出发进一步优化映射规则。WSN 执行监测任务时,汇聚节点可能处于工作或者待机状态。因此,在活动 and 空闲时所产生的能耗是图 U 中节点 v_i 的主要能耗来源^[10],即

$$E_{v_i}^{\text{total}} = E_{v_i}^{\text{active}} + E_{v_i}^{\text{idle}} \quad (10)$$

通常雾网络中节点 $\gamma(n_i)$ 的活动能耗 $E_{\gamma(n_i)}^{\text{active}}$ 包括计算和传输两大部分,其中传输能耗分为发送和接收数据能耗^[11];空闲能耗 $E_{\gamma(n_i)}^{\text{idle}}$ 是节点在待机模式下能量消耗,它与空闲时的功率和处于空闲的时间段有关。两者的具体表达式如(11)~(12)式所示:

$$E_{\gamma(n_i)}^{\text{active}} = E_{\gamma(n_i)}^{\text{calc}} + E_{\gamma(n_i)}^{\text{tx}} + E_{\gamma(n_i)}^{\text{rx}} = kp_{\gamma(n_i)}^{\sigma} T_{\gamma(n_i)}^{\text{calc}} + (E_{\gamma(n_i)}^{\text{elec}} + E_{\gamma(n_i)}^{\text{amp}}) + D_{n_i n_{i+1}} E_{\gamma(n_i)}^{\text{elec}} =$$

$$\begin{cases} \sum_{p=1}^m kp_{v_p}^{\sigma} \frac{D_{n_i} \alpha}{p_{v_p}} x_{n_i v_p} + \sum_{p=1}^m \sum_{q=1}^m x_{n_i v_q} x_{n_{i+1} v_p} D_{n_i n_{i+1}} (E_{\text{elec}} + \varepsilon_f R_{v_p v_q}^2 N_{v_p v_q}^1), & i = 1 \\ \sum_{p=1}^m kp_{v_p}^{\sigma} \frac{D_{n_i} \alpha}{p_{v_p}} x_{n_i v_p} + \sum_{p=1}^m \sum_{q=1}^m x_{n_i v_q} x_{n_{i+1} v_p} D_{n_i n_{i+1}} (E_{\text{elec}} + \varepsilon_f R_{v_p v_q}^2 N_{v_p v_q}^1) + \sum_{p=1}^m \sum_{r=1}^m x_{n_{i-1} v_p} x_{n_i v_r} D_{n_{i-1} n_i} E_{\text{elec}}, & i = 2, 3, \dots, t-1 \\ \sum_{p=1}^m kp_{v_p}^{\sigma} \frac{D_{n_i} \alpha}{p_{v_p}} x_{n_i v_p} + \sum_{p=1}^m \sum_{r=1}^m x_{n_{i-1} v_p} x_{n_i v_r} D_{n_{i-1} n_i} E_{\text{elec}}, & i = t \end{cases} \quad (11)$$

$$E_{\gamma(n_i)}^{\text{idle}} = P_{\gamma(n_i)}^{\text{idle}} T_{\gamma(n_i)}^{\text{idle}} = P_{\gamma(n_i)}^{\text{idle}} (T_{\text{total}}(n_i) - T_{\gamma(n_i)}^{\text{calc}} - T_{\gamma(n_i)}^{\text{tx}} - T_{\gamma(n_i)}^{\text{rx}}) =$$

$$\begin{cases} \sum_{p=1}^m P_{v_p}^{\text{idle}} x_{n_i v_p} \left(T_{\text{total}}(n_i) - \sum_{p=1}^m \frac{D_{n_i} \alpha}{p_{v_p}} x_{n_i v_p} - \sum_{p=1}^m \sum_{q=1}^m x_{n_i v_q} x_{n_{i+1} v_p} \frac{D_{n_i n_{i+1}} (E_{\text{elec}} + \varepsilon_f R_{v_p v_q}^2 N_{v_p v_q}^1)}{P_{\text{tx}}} \right), & i = 1 \\ \sum_{p=1}^m P_{v_p}^{\text{idle}} x_{n_i v_p} \left(T_{\text{total}}(n_i) - \sum_{p=1}^m \frac{D_{n_i} \alpha}{p_{v_p}} x_{n_i v_p} - \sum_{p=1}^m \sum_{q=1}^m x_{n_i v_q} x_{n_{i+1} v_p} \frac{D_{n_i n_{i+1}} (E_{\text{elec}} + \varepsilon_f R_{v_p v_q}^2 N_{v_p v_q}^1)}{P_{\text{tx}}} - \sum_{p=1}^m \sum_{r=1}^m x_{n_{i-1} v_p} x_{n_i v_r} \frac{D_{n_{i-1} n_i} E_{\text{elec}}}{P_{\text{rx}}} \right), & i = 2, 3, \dots, t-1 \\ \sum_{p=1}^m P_{v_p}^{\text{idle}} x_{n_i v_p} \left(T_{\text{total}}(n_i) - \sum_{p=1}^m \frac{D_{n_i} \alpha}{p_{v_p}} x_{n_i v_p} - \sum_{p=1}^m \sum_{r=1}^m x_{n_{i-1} v_p} x_{n_i v_r} \frac{D_{n_{i-1} n_i} E_{\text{elec}}}{P_{\text{rx}}} \right), & i = t \end{cases} \quad (12)$$

(11)式中的 $E_{\gamma(n_i)}^{\text{calc}}$ 为节点计算数据所需要的能耗; $E_{\gamma(n_i)}^{\text{tx}}$ 为节点发送数据所需要的能耗,等于传输信号电路的能耗 $E_{\gamma(n_i)}^{\text{elec}}$ 和放大信号电路的能耗 $E_{\gamma(n_i)}^{\text{amp}}$ 之和; $E_{\gamma(n_i)}^{\text{rx}}$ 为节点接收数据所需要的能耗,只与传输信号电路的能耗有关; E_{elec} 为节点在接收或发送单位数据量时所需要的能耗; ε_f 为节点的功率放大能量系数。(12)式中的 $P_{\gamma(n_i)}^{\text{idle}}$ 是节点处于空闲时的功率,通常是一个恒定的值; $T_{\gamma(n_i)}^{\text{calc}}$, $T_{\gamma(n_i)}^{\text{tx}}$, $T_{\gamma(n_i)}^{\text{rx}}$ 分别表示节点计算数据、发送数据和接收数据所需要的

时间段。结合文献[12], P_{tx} , P_{rx} 分别代表节点的发送数据和接收数据功率,通常是固定的常数; k, σ 为正实数分别取 10^{-27} 和 3; R 代表了图 U 的网络边权重,用来表示节点间的距离。

结合上述分析,整个雾网络的总能耗为

$$E_{\text{fog}}^{\text{total}} = \sum_{i=1}^t E_{\gamma(n_i)}^{\text{total}} = \sum_{i=1}^t E_{\gamma(n_i)}^{\text{active}} + \sum_{i=1}^t E_{\gamma(n_i)}^{\text{idle}} \quad (13)$$

假设在某次监测任务中雾网络的最大能量为 E_{max} , 通过时延优化模型得出的最优映射关系所对应的能耗必须满足整个雾网络的能耗约束,即小于

或等于 $E_{\max}$ 。

根据上文可知,为了找到这种既符合能耗要求,又满足任务最小总时延的最优映射关系,需要建立一个优化问题模型。同时,由于图 **D** 至图 **U** 的映射关系结果只存在是与否 2 种可能,相当于该时延优化问题的变量只取 0 或 1,所以该问题是一个二值优化问题,因此建立 (14) 式的二值优化问题模型。

$$\begin{aligned} & \mathbf{X} = \operatorname{argmin}(F(\mathbf{X})) \\ & \begin{cases} x_{n_p v_q} \in \{0, 1\}, \forall n_p \in \Omega, \forall v_q \in V \\ P_{\gamma(n_p) \gamma(n_q)} \in P, \gamma(n_p), \gamma(n_q) \in V \\ x_{n_p v_q} = 1, p = t, q = m \\ x_{n_p v_q} = 1, \exists p, q \in [1, s] \\ \text{s.t. } x_{n_p v_q} = 1, \exists p \in [s + 1, t - 1], \\ \qquad \qquad \qquad \exists q \in [s + 1, m - 1] \\ x_{n_p v_q} = 0, p \in [1, s], \forall q \in [s + 1, m - 1] \\ x_{n_p v_q} = 0, v_q = v_m, \forall p \in [s + 1, t - 1] \\ E_{\text{fog}}^{\text{total}} \leq E_{\max} \end{cases} \end{aligned} \tag{14}$$

3 基于 SA 的 BPSO 算法

为求解该二值优化问题模型,本文使用了一种基于 BPSO 的改进算法 (SA-BPSO)。因为 BPSO 算法在迭代后期容易收敛于局部最优,同时会出现停滞状况,从而导致误差结果较大。相比 BPSO 算法,SA-BPSO 提高了算法的收敛和全局搜寻能力,减少了发生陷入局部最优的概率,能够更加快速和准确地求解数学模型 (14) 的最优映射关系。

3.1 模拟退火算法

模拟退火 (SA) 算法在迭代期间有一定的概率接收到不理想的解,因此可以避开局部最优。它的思想受固体退火过程的启发,利用控制温度参数 T 、降温速率 R 和终止条件温度 E 等参数来控制算法的流程,在参数 T 逐渐衰减的过程中利用 Metropolis 接受准则持续更新可行解,其中接受新可行解的概率如 (15) 式所示。该算法只要参数 T 的取值充分大,且 T 的衰减速度十分慢,就可以收敛到优化问题的全局最优可行解。SA 算法流程图如图 2 所示。

$$\min \left\{ 1, \exp \left(- \frac{f(x_2) - f(x_1)}{T_k} \right) \right\} > \operatorname{rand}[0, 1] \tag{15}$$

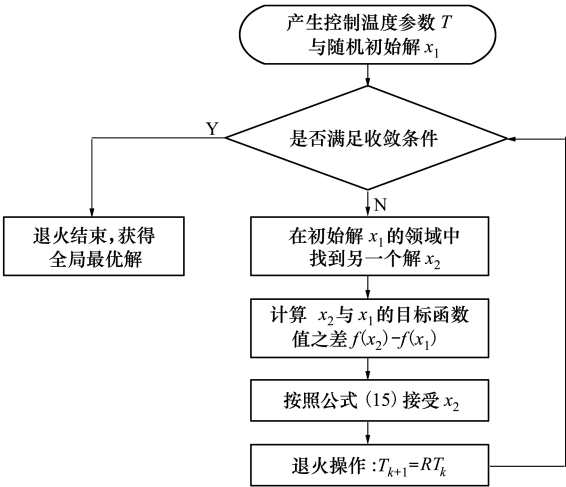


图 2 SA 算法流程图

3.2 SA-BPSO 算法

在使用 SA-BPSO 算法求解数学模型 (14) 时,假设粒子群的总体个数为 Q ; 算法的当前迭代次数为 n , 其范围为 $n \in \{1, 2, \dots, P_{\max}\}$, $P_{\max}$ 为迭代的总数, 则第 i 个粒子的位置矩阵和速度矩阵分别表示如下:

$$\mathbf{X}^n(i) = \begin{pmatrix} x_{n_1 v_1}^n(i) & \cdots & x_{n_1 v_m}^n(i) \\ \vdots & & \vdots \\ x_{n_t v_1}^n(i) & \cdots & x_{n_t v_m}^n(i) \end{pmatrix} \tag{16}$$

$$\mathbf{V}^n(i) = \begin{pmatrix} v_{n_1 v_1}^n(i) & \cdots & v_{n_1 v_m}^n(i) \\ \vdots & & \vdots \\ v_{n_t v_1}^n(i) & \cdots & v_{n_t v_m}^n(i) \end{pmatrix} \tag{17}$$

在算法的第 n 次迭代过程中,第 i 个粒子的速度更新公式为

$$\begin{aligned} \mathbf{V}^{n+1}(i) = & w \mathbf{V}^n(i) + c_1 \gamma_1 (\mathbf{p}_{\text{best}}^n(i) - \mathbf{X}^n(i)) + \\ & c_2 \gamma_2 (\mathbf{p}_{\text{best}}^n(i) - \mathbf{X}^n(i)) \end{aligned} \tag{18}$$

式中, $\mathbf{X}^n(i)$ 与 $\mathbf{V}^n(i)$ 分别为粒子的当前位置和速度; c_1 与 c_2 都是加速常数; γ_1 与 γ_2 都是位于 $[0, 1]$ 区间的随机数; w 为惯性权重; $\mathbf{p}_{\text{best}}^n(i)$ 为粒子的个体最优位置。

特别地, (18) 式中的 $\mathbf{p}_{\text{best}}^n(i)$ 替代了原始速度更新公式里的全局最优位置 $\mathbf{g}_{\text{best}}^n(i)$, 它是在算法迭代的过程中按一定概率从粒子众多的 $\mathbf{p}_{\text{best}}^n(i)$ 选出的劣值。当算法在迭代后期陷入局部最优时, 如果此时粒子群的全局最优位置刚好更新为某个劣值, 这样所有粒子都会向着劣值移动, 从而使粒子群逃离该局部最优。同时为了接收性能更加优良的劣

值,利用轮盘赌思想将其被选中的概率(突跳概率)增大。概率具体的计算公式和第 i 个粒子的位置更新公式分别如(19)~(21)式所示:

$$P_{\text{jump}} = \frac{e^{-\frac{(f(p_{\text{best}}^n(i)) - f(g_{\text{best}}^n(i)))}{t}}}{\sum_{i=1}^Q e^{-\frac{(f(p_{\text{best}}^n(i)) - f(g_{\text{best}}^n(i)))}{t}}} \quad (19)$$

$$s(V^{(n+1)}(i)) = \frac{1}{1 + \exp(-V^{n+1}(i))} \quad (20)$$

$$X^{n+1}(i) = \begin{cases} 1 & \text{if rand} \leq s(V^{n+1}(i)) \\ 0 & \text{otherwise} \end{cases} \quad (21)$$

适用于该算法的自适应函数如(22)式所示

$$f(X) = F(X) = T(D) \quad (22)$$

综上所述,SA-BPSO 算法求解该二值优化问题

模型的总体流程图如图 3 所示。通过图 3 可知,在使用 SA-BPSO 求解(14)式的优化问题模型中,关于算法复杂度的分析主要包含三部分。第一个是使用 Dijkstra 算法寻求图 U 中两节点间传送单位数据量的最短时间和最短路径,该算法复杂度主要体现在顶点的个数,即图 U 中节点数量,同时本文要求各个节点之间的最短距离,所以复杂度是 $O(n^3)$ 。第二个是针对任务总时延和总能耗的计算,该过程的算法复杂度主要体现在遍历任务节点到所有雾节点的映射情况,所以复杂度为 $O(n^2)$ 。第三个是针对粒子群寻找最优映射关系的过程,其中算法复杂度主要由粒子群数目和迭代次数决定,也就是 $O(N_{\max}Q)$ 。

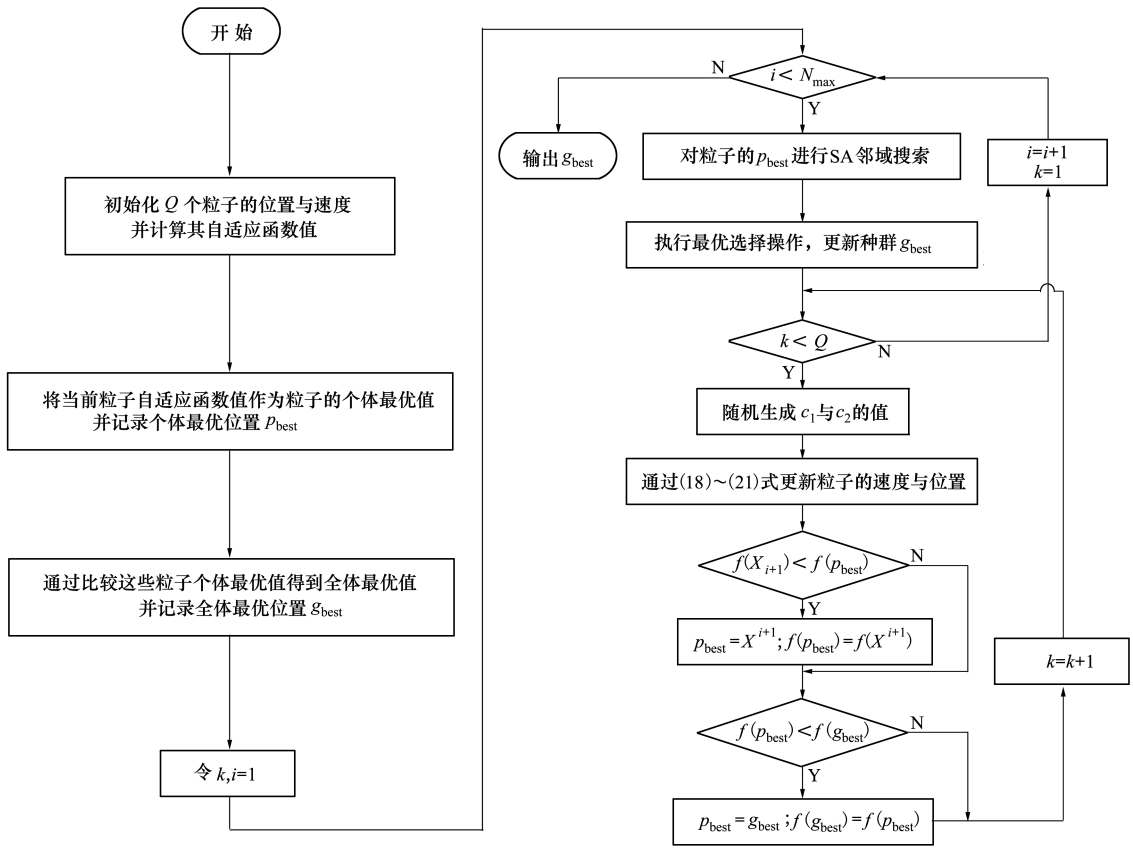


图 3 SA-BPSO 算法总体流程图

4 仿真实验与结果分析

仿真参数设置如表 1 所示, $e_{\max}$ 为单个汇聚节点所能携带的最大能量,本次仿真共设有 18 个汇聚节点,所以这些汇聚节点具有的最大能量范围为

[10,40]J; f_c 为云服务器的任务处理能力; p_{fog} 为汇聚节点的任务处理能力; R_{fog} 为雾网络的链路数据传输速率; α 为子任务的平均难度系数^[13-15]。SA-BPSO 算法的参数为:粒子群的总体数量 $Q = 40$,算法的总迭代次数 $N_{\max} = 100$,加速常数 $c_1 = c_2 = 1$,惯性权重 $w = 1.5$ 。仿真采用的 DAG 与 UG 模型分别

如图 4~5 所示。

表 1 参数设置表

名称	取值	名称	取值
$R_{\text{fog}}/(\text{Mb} \cdot \text{s}^{-1})$	[20,50]	e_{max}/J	[0.56,2.22]
$p_{\text{fog}}/\text{GHz}$	[1,5]	f_c/GHz	20
P^{idle}/W	0.15	α	1 900/8
P_{tx}/W	0.72	$\varepsilon_f/(\text{pJ} \cdot \text{bit}^{-1} \cdot \text{m}^{-2})$	10
P_{rx}/W	0.58	$E_{\text{elec}}/(\text{nJ} \cdot \text{bit}^{-1})$	50

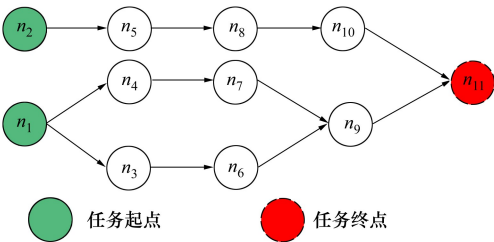


图 4 任务模型图

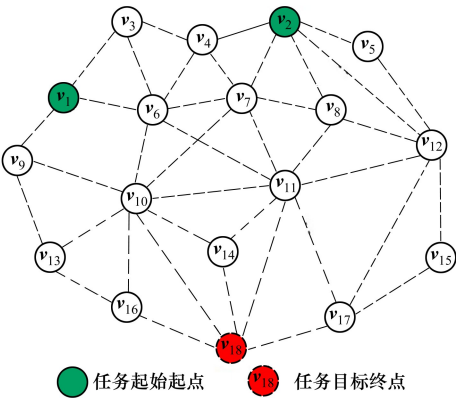


图 5 网络拓扑结构图

4.1 路径计算与 WSN 融合云计算的任务处理时延对比

为验证本文提到的路径计算方法优于传统的云计算任务处理方式,在平均任务难度系数 α 和 E_{max} 相同的情况下进行仿真,其中 E_{max} 设置为 40 J,仿真结果如图 6 所示。由图 6 可知在数据量为 1 Mb 时,路径计算的总时延为 0.771 s,而云计算的总时延 (0.895 s) 稍大于路径计算。在数据量从 1 Mb 增加至 10 Mb 的过程中,路径计算的计算时延平均增速大于传输时延,分别是 0.451 s/Mb 和 0.173 s/Mb;云计算的传输时延平均增速较大,为 0.8 s/Mb,而计算时延平均增速仅为 0.095 s/Mb。在数据量为

10 Mb 时,路径计算的任务处理总时延比云计算减少了 3.547 s,性能提升约 40%。综上可知,路径计算的时延性能优于云计算。

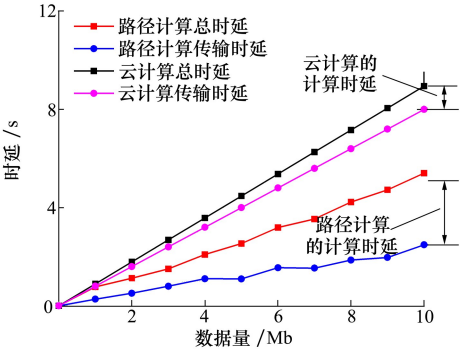


图 6 路径计算与云计算的时延性能对比

4.2 任务处理时延与任务难度系数的变化关系

在数据量和 E_{max} 分别取 10 Mb 和 40 J 的情况下,仿真结果如图 7 所示。

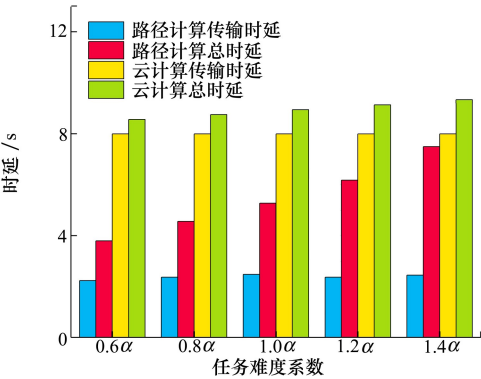


图 7 任务难度系数对任务处理时延的影响

从图 7 得知,在任务难度系数从 0.6α 提高到 1.4α 的过程中,路径计算的传输时延与计算时延分别增加了 0.184,3.486 s;云计算的传输时延未发生改变,计算时延只增加了 0.76 s。路径计算与云计算的传输时延几乎无变化,两者的计算时延虽然都呈上升趋势,但是路径计算的增幅远大于云计算。主要原因是任务难度系数只会影响计算时延,不会对传输时延造成影响,同时随着任务难度系数逐渐提高,云计算会体现出其任务处理能力优势,而汇聚节点的任务处理能力有限,从而导致两者的计算时延存在较大差距。

4.3 总能耗与数据量和任务难度系数的关系

为说明在不同数据量和任务难度系数下雾网络的总能耗特性,在时延优化的基础上进行仿真,其仿

真结果如图 8 所示。

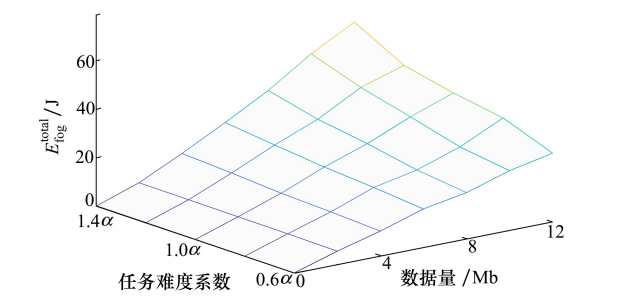


图 8 数据量和任务难度系数对总能耗的影响

仿真结果显示当任务难度系数固定时,随着数据量增加,雾网络的总能耗呈逐渐上升趋势,同时随着任务难度系数增大,相应的总能耗上升幅度也在增加。具体来讲,当数据量由 2 Mb 增加至 12 Mb,任务难度系数分别取 0.6α 、 1.4α 时,总能耗的增幅分别为 22.8 和 46.7 J。因为随着数据量与任务难度系数的增加,雾节点处理任务所需的时延将升高,导致能耗变大。当任务难度系数逐渐增大时,较小数据量的总能耗增长趋平缓,但随着数据量的增加,增幅变得越来越大。比如:在任务难度系数从 0.6α 增加至 1.4α 时,数据量取 2 Mb 与 12 Mb 的能耗分别增加了 13% 和 105%。原因是在数据量较小时,任务难度系数的改变对任务处理时延的影响不大,因此能耗变化程度很小;但随着数据量的增加,任务难度系数的影响会越发明显,所以能耗急剧增长。

4.4 多种映射算法的时延性能对比

本小节从 $E_{\max}$ 角度出发,比较了 SA-BPSO 算法同贪婪负载均衡算法 (Greedy-LB)、加权轮转算法 (WRR) 和随机动态算法 (Pick-KX) 的任务计算时延,仿真结果如图 9 所示。同时将 SA-BPSO 算法与 BPSO 算法从收敛速度方面进行了对比,仿真结果如图 10 所示。

图 9 在数据量与任务难度系数分别取 10 Mb 和 α 时,随着 $E_{\max}$ 的增加,4 种算法对应的计算时延都逼近为一个常数,同时在这个过程中 SA-BSPO 算法的任务计算时延一直优于其他 3 种算法。在 $E_{\max}$ 取 40 J 时,SA-BSPO、Greedy-LB、WRR 和 Pick-KX 的计算时延分别为 2.919、6.032、6.647 和 7.701 s,SA-BSPO 算法相比其他 3 种算法分别降低了 51.60%、56.08% 和 62.09%。

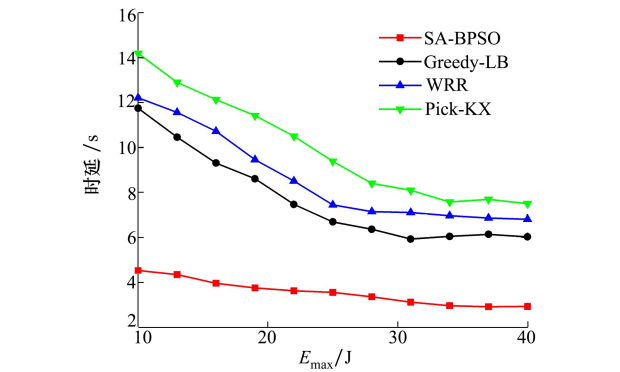


图 9 不同雾网络最大能耗对四种算法计算时延的影响

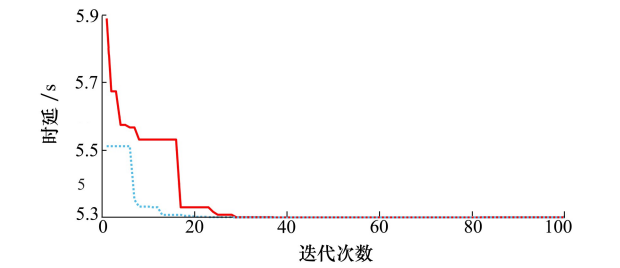


图 10 BPSO 算法与 SA-BPSO 算法收敛速度对比

图 10 在 $E_{\max}$ 取 40 J、数据量取 10 Mb 和任务难度系数为 α 的情况下,2 种算法随着迭代次数增加,都呈现先下降后趋于 5.3 s 的趋势,但可以看出 SA-BPSO 收敛所用的迭代次数明显少于 BPSO。在迭代次数为 100 时,SA-BPSO 与 BPSO 算法的路径计算总时延分别为 5.317 3、5.331 8 s。因此,SA-BPSO 算法相比于 BPSO 算法具有良好的收敛性和鲁棒性,求得的解更准确。因为 SA-BPSO 算法随着代次数增加,接收劣值的概率逐步降低,从而改善了 BPSO 算法收敛性较差的问题;同时 SA-BPSO 算法相比 BPSO 算法增强了全局搜索能力,使算法的迭代速度与收敛的精确度得到提高。

5 结 论

本文针对 WSN 融合云计算技术时效性不好的问题,提出了基于云雾网络架构的路径计算方法,并研究了基于 SA-BPSO 算法的映射规则,同时将 WSN 的能耗问题纳入研究范围。仿真结果表明,该方法有效地解决了云计算模式中任务处理时延比较高的问题,同时对比其他映射算法,本文的映射算法降低了任务计算时延,实现了对监测任务的最优映射。虽然本次研究验证了该方法的有效性,但是依

然存在着许多未考虑的地方,比如:任务映射规则仅研究了单任务场景和约束条件只考虑了雾网络的总能耗要求。因此,在以后的研究中应该从单个汇聚

节点的能耗要求和剩余能量、多任务场景或 WSN 链路质量等方面出发继续深化研究内容,进一步在能耗约束下减少任务处理时延,延长 WSN 寿命。

参考文献:

- [1] 冉卓衡. 基于 NB-IoT 网络的实时环境监测系统在众包模式下的应用[J]. 通信电源技术, 2020, 37(12): 88-90
RAN Zhuoheng. Application of NB-IoT network based real-time environment monitoring system in crowdsourcing mode[J]. Telecom Power Technology, 2020, 37(12): 88-90 (in Chinese)
- [2] 孙浩然. 大规模农田无线传感器网络高能效数据汇集方法[D]. 长春: 吉林农业大学, 2020: 8-10
SUN Haoran. Energy-efficient data collection method for large-scale farmland wireless sensor network[D]. Changchun: Jilin Agricultural University, 2020: 8-10 (in Chinese)
- [3] 马璐, 刘铭, 李超, 等. 面向 6G 边缘网络的云边协同计算任务调度算法[J]. 北京邮电大学学报, 2020, 43(6): 66-73
MA Lu, LIU Ming, LI Chao, et al. A cloud-edge collaborative computing task scheduling algorithm for 6G edge networks[J]. Journal of Beijing University of Posts and Telecommunications, 2020, 43(6): 66-73 (in Chinese)
- [4] SHUKLA S, VYAVAHARE P, KURI J, et al. Fast arbitrary function computation over a wireless network: a linear programming [C]//IEEE Wireless Communications and Networking Conference. Piscataway, NJ, USA, 2015
- [5] 牛祺君, 张永辉. 基于蜂群算法的无线传感器网络层次路由优化[J]. 计算机仿真, 2018, 35(12): 229-232
NIU Qijun, ZHANG Yonghui. Optimization of WSN hierarchical routing based on artificial bee colony algorithm[J]. Computer Simulation, 2018, 35(12): 229-232 (in Chinese)
- [6] 汪小威, 林宁, 胡玉平. 移动边缘计算中利用 BPSO 的任务卸载策略[J]. 计算机工程与设计, 2021, 42(12): 3333-3341
WANG Xiaowei, LIN Ning, HU Yuping. Task offloading strategy using BPSO in mobile edge computing[J]. Computer Engineering and Design, 2021, 42(12): 3333-3341 (in Chinese)
- [7] 马步云, 马新策, 黄松, 等. WSN 低功耗低时延路径式协同计算方法[J]. 无线电通信技术, 2021, 47(2): 168-177
MA Buyun, MA Xince, HUANG Song, et al. Low-power low-latency path-based collaborative computing scheme for WSN[J]. Radio Communications Technology, 2021, 47(2): 168-177 (in Chinese)
- [8] VYAVAHARE P, SHETTY A. On optimal embeddings for distributed computation of arbitrary functions[C]//10th International Conference on Signal Processing and Communications, Piscataway, NJ, USA, 2014
- [9] HOU X, REN Z, CHENG W, et al. Fog based computation offloading for swarm of drones[C]//2019 IEEE International Conference on Communications, Piscataway, NJ, USA, 2019
- [10] 张玉健. 面向能耗优化的异构计算系统任务调度研究[D]. 南京: 东南大学, 2018: 60-62
ZHANG Yujian. Energy-efficient task scheduling on heterogeneous computing systems[D]. Nanjing: Southeast University, 2018: 60-62 (in Chinese)
- [11] 方丁. 基于低能耗的无线传感器网络栅栏覆盖方法研究[D]. 杭州: 浙江工业大学, 2020: 10-11
FANG Ding. Research on barrier covering method of wireless sensor network based on low energy consumption[D]. Hangzhou: Zhejiang University of Technology, 2020: 10-11 (in Chinese)
- [12] HOU X, REN Z, WANG J, et al. Distributed fog computing for latency and reliability guaranteed swarm of drones[J]. IEEE Access, 2020, 8: 7117-7130
- [13] LI X, ZHU L, CHU X, et al. Edge computing-enabled wireless sensor networks for multiple data collection tasks in smart agriculture[J]. Journal of Sensors, 2020, 9: 1-9
- [14] XU S, WANG X, HUANG M. Capacity analysis method for MLSN based on improved DGA[C]//2015 11th International Conference on Natural Computation, Piscataway, 2015
- [15] CAO S Z, WEI J Y, HAN H, et al. Space edge cloud enabling network slicing for 5G satellite network[C]//2019 15th International Wireless Communications & Mobile Computing Conference, Piscataway, 2019

WSN latency optimization based on path calculation method

ZHU Peng¹, REN Jijun¹, REN Zhiyuan²

(1.School of Communications and Information Engineering,
Xi'an University of Posts and Telecommunications, Xi'an 710121, China;
2.State Key Laboratory of Integrated Services Networks,Xidian University, Xi'an 710071, China)

Abstract: With the rapid development of the Internet of Things (IoT) industry, wireless sensor network (WSN) fusion cloud computing technology is encountering the challenges of high task processing latency and limited sensor node energy. Therefore, a path calculation method based on cloud computing network architecture is proposed. WSN monitoring tasks are deployed to specific edge devices reasonably by using the computing resources of network edge devices in the fog computing layer to reduce the task processing latency under the constraints of energy consumption. In order to efficiently assign tasks to the fog computing layer, a task mapping rule is used to map the monitoring tasks represented by the directed acyclic graph to the fog computing layer network represented by the acyclic graph. At the same time, a binary optimization problem for finding the optimal mapping relationship is established with time latency and energy constraints. Finally, the simulated annealing-discrete binary particle swarm optimization (SA-BPSO) algorithm is used to solve the optimization problem. The simulation results show that the latency performance under this method is about 40% higher than that of WSN fusion cloud computing technology when the data volume is 10 Mb.

Keywords: wireless sensor network; path calculation; cloud and fog network architecture; SA-BPSO

引用格式:朱鹏,任继军,任智源. 基于路径计算方法的 WSN 时延优化研究[J]. 西北工业大学学报,2022,40(6): 1394-1403
ZHU Peng, REN Jijun, REN Zhiyuan. WSN latency optimization based on path calculation method[J]. Journal of Northwestern Polytechnical University, 2022, 40(6): 1394-1403 (in Chinese)