

一种基于动态规划的最小化最大加权响应时间的中心控制节点选举算法

万柏麟, 杨奇, 闫中江, 杨懋, 李波

(西北工业大学 电子信息学院, 陕西 西安 710072)

摘要:为了最小化网络中任意节点到达中心控制节点的最大加权响应时间,提出了一种基于动态规划的中心控制节点选举算法。无线网络中的节点和链路的响应时间被建模为网络拓扑图中的节点权值和边权值,进而最小化网络中任意节点到达中心控制节点的最大加权响应时间的中心控制节点选举问题被建模为 K -中心问题,其中 K 表示中心控制节点的个数。采用基于动态规划的插点法可求出任意2个点之间的最小加权响应时间,所建模的 K -中心问题被转化为若干个 R -控制集问题。将若干个 R -控制集问题转化为若干个0-1整数规划问题,采用分支定界的方法逐个求解每个整数规划问题。给出了 $K=1$ 时上述算法的简化实现方法,证明了所提算法的最优性并分析了算法的复杂度。仿真结果表明,所提算法选举的中心控制算法可最小化网络最大加权响应时间。

关键词:无线网络;中心节点选举;动态规划

中图分类号: TN92

文献标志码: A

文章编号: 1000-2758(2023)01-0073-08

在无线网络协议架构中,基于传统的开放互联系统模型^[1](open system interconnection, OSI)的分层标准结构,人们提出了基于平面的标准结构下的软件定义网络(software defined network, SDN)结构模型。该结构模型将开放的互联系统分为控制面和业务面,提供了在基于OSI模型结构下传统通信中插入调度系统的可能。在调度系统中存在统一的中心控制节点集是关键,中心控制节点集负责整体网络调度以及协调工作。例如,在无线网络中进行频点优化、路由优化,在复杂的物联网环境中负责分配带宽资源等,而在弹性通信网络中,需要快速地在弹性组织网络中选举一个中心控制节点集。因此,在自组织网络下,中心控制节点集的选举就十分重要。

国内外对中心控制集的选举问题做了大量的研究,文献[2]将参照能量和点对点的传输时间作为评估标准进行中心控制节点自选举;文献[3]在网络拓扑中的任一节点获取自身的加权平均值进行自

选举,该加权平均值用来描述所述网络中节点作为中心节点的适合程度;文献[4]是一种分布式中心控制节点选举方法,将最高节点度数(即节点的一跳邻居节点数)作为标准;文献[5]使用节点的剩余能量,集群范围内活动的邻居节点的数量以及节点与基站之间的距离来计算成为簇头的几率;文献[6]通过动态规划下计算树状网络的控制集完成中心控制节点选举;文献[7]在多重竞争下根据能量、邻节点的个数以及与已选簇头的距离来决定簇头的位置;文献[8]将节点度和介数中心性抽象成领导因素,再由领导因素通过算法找到簇头节点;文献[9]基于混沌粒子群算法下,对低功耗自适应集簇型分层协议(low energy adaptive clustering hierarchy, LEACH)进行改进;文献[10]利用遗传算法在LEACH上改进。

现有研究尚存在下述不足。首先,现有工作未对最小化最大响应时间进行研究,而在现有的工业

收稿日期:2022-05-19

基金项目:国家自然科学基金(61771392,61871322,61771390)与航空科学基金(201955053002,201855533035)资助

作者简介:万柏麟(1998—),西北工业大学硕士研究生,主要从事无线自组织网络协议研究。

通信作者:闫中江(1983—),西北工业大学副教授,主要从事无线网络组网协议设计研究。e-mail:zhijyan@nwpu.edu.cn

物联网、家庭物联网等无线接入网络中,节点移动性不强,无法保障时延性能,因此,考虑最小化响应时间具有重要研究意义;其次,部分算法基于 LEACH 协议,该算法具有随机性,即使是改进后也有可能选举到非最优点;最后,部分可选举最优节点的算法只适用于树状拓扑,且只适用于点权值为正数下的情况。

本文针对网络中最大加权响应时间的最小化问题,提出了适合于无线网络下的中心控制节点自选举方案。首先,将最小化最大响应时间的中心控制节点选举问题建模为 K -中心问题,其中 K 表示中心控制节点的个数。然后,基于动态规划的插点法,求出任意 2 个点之间的距离,从而将所建模的 K -中心问题转化为若干个 R -控制集问题。接着,将若干个 R -控制集问题转化为若干个 0-1 整数规划问题。最后,对每个 0-1 规划问题采用分支定界法进行求解,并设计了一种适用于 $K = 1$ 时的简化算法,当 $K > 1$ 时时间复杂度为 $O(n^{K+2}\log_2 n)$,而当 $K = 1$ 时时间复杂度可降为 $O(n^3)$ 。

1 系统建模

1.1 无线网络描述

无线网络可建模为无向加权网络图 $T = (V, E; f, w)$,其中 E 是网络拓扑中各个节点之间的链路的集合, V 是网络图 T 中各个节点的集合。映射 $w: V \rightarrow \{x \in \mathbf{R} | x \geq 0\}$ 表示点到权值的映射,该权值称为点权值,用以描述节点处理数据包所需的时间,该权值越大表示该节点的业务负载越大,从而导致单个数据包传输时延越大。映射 $f: E \rightarrow \mathbf{R}^+$ 表示链路到权值的映射,该权值称为边权值,用以表示一条链路传输包所需的时间,该权值越大同一个大小的数据包所消耗的传输时间越多。 e_i 与 v_i 分别代表集合 E 与集合 V 中的元素,分别代表了网络图中的链路和节点。

节点 v_i 到 v_j 加权响应时间,指的是从节点 v_i 发出请求数据包到节点 v_j 收到该请求数据包的时间,其中包括链路传输的时间和节点中继转发的时间。

1.2 目标函数建立和问题建模

在无线网络中,根据节点是否具有数据包中继转发功能,所有节点可被分为桥节点和叶子节点两类,其中桥节点具有 AP 或 Master 网卡而叶子节点不具有 AP 或 Master 网卡。不同类型的节点可通过

安装不同类型的网卡实现,表 1 给出了 Wi-Fi 网络和蓝牙网络中的网卡类型与节点类型的对照关系。

表 1 Wi-Fi 和蓝牙网络中桥与叶子节点的网卡示例

网卡类型	桥节点	叶子节点
AP 或 Master 网卡	✓	×
STA 或 Slave 网卡	✓	✓

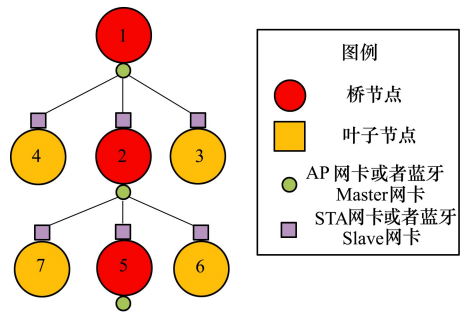


图 1 网络连接示意图

图 1 给出了网络连接示意图,其中节点 1,2,5 为桥节点,分别装配有 Wi-Fi 中的 AP (access point) 网卡或蓝牙 Master 网卡,而节点 3,4,6,7 为叶子节点,仅装配有 Wi-Fi 中的 STA (station) 网卡或蓝牙 Slave 网卡。节点 v_i 到 v_j 的加权响应时间包括两部分,链路传输所消耗的时间和节点中继转发的时间,因此在网络中节点 v_i 到 v_j 的加权响应时间 t_{Rspij} 可写成

$$t_{Rspij} = \sum_{v_k \in w, v_k \neq v_j} w(v_k) + \sum_{e_k \in w} f(e_k) \quad (1)$$

公式(1)中, w 表示路径空间,即表示节点 v_i 到 v_j 的路由路径上经过节点 v_k 以及对应的链路 e_k ,用符号表达为

$$w = \{v_i \xrightarrow{e_a} v_a \cdots \xrightarrow{e_k} v_j\} \quad (2)$$

根据上述分析,最小化最大加权响应时间的目标优化模型为

$$\min_{S \subseteq V} \max_{v_j \in V} \min_{v_i \in V} t_{Rspij} \quad (3)$$

其物理含义为,从节点集合 V 中选举 K 个节点为中心控制节点集 S ,使得最大的加权响应时间最小。该问题在数学中被称为 K -中心问题^[11]。

2 算法设计

K -中心问题是一个非确定多项式时间问题 (non-deterministic polynomial, NP) 问题^[12]。 K -中心

问题是指在无向赋权网络中,从备选节点中选择 K 个中心节点,使其到非中心节点的最大距离最小。

2.1 任意 K 下基于动态规划的算法设计

2.1.1 基于动态规划的距离 d 的计算

该算法用于计算全局网络图拓扑的任意节点间的最小加权响应时间,使用动态规划思路进行求解。

算法核心思想是将所有节点依次插入任意两节点之间,以求得任意两点间的最小加权响应时间。当所有节点插入完成后,形成的加权响应时间矩阵为任意节点间最小加权响应时间矩阵,该矩阵记作矩阵 M 。根据文献[13]可知,算法时间复杂度为 $O(n^3)$ 。

2.1.2 K -中心问题的计算方法

根据文献[6]相关内容,最小化最大加权响应时间问题类似于一个 K -中心问题,该问题是寻找满足要求的中心控制节点集 D ,且 $|D| \leq K$,使得全拓扑中节点到达 D 中的最大加权响应时间 r 最小。

对于 K -中心问题中的最优解 r_0 ,当使用加权响应时间 $r < r_0$ 进行覆盖时,必然有满足要求的中心控制集合 $|D| > K$,反之,对于覆盖的加权响应时间 $r \geq r_0$,必存在满足要求的中心控制集合 $|D| \leq K$ 。假设行向量 $D = \{d^{(1)}, d^{(2)}, \dots, d^{(n^2)}\}$, D 中元素为 2.1 节中 M 矩阵的元素从小到大排列,且 $r_0 \in D$ 。因此,利用上述最优解的特点,并采用二分法迭代求得 r_0 的大小,其算法流程图如下:

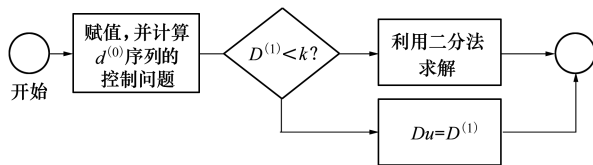


图 2 算法流程图

将根据覆盖范围 r ,寻找最少中心控制节点个数的问题称为 R -控制集问题^[6],最后该 K -中心问题就会转换为若干个 R -控制集问题。上述算法伪代码如算法 1 所示。

第一步是计算距离矩阵 M ,并将其中元素排列成从小到大的集合 D ,第二步是设置初始值,并计算集合 D 起始元素是否符合约束条件,若符合则直接输出,第三步是采用二分法,利用覆盖问题的相关性质,使 d^u 逼近最优解 r_0 ,第四步输出结果。

算法 1 K -中心算法:

step1 通过动态规划中的算法求出任何两点

间的加权响应时间,并将 M 中元素从小到大排列成行向量 $D = \{d^{(1)}, d^{(2)}, \dots, d^{(n^2)}\}$;

step2 令 $U = n^2, L = 1, r = d^{(1)}$,执行 R -控制集算法,输出控制集 D_1 ,若 $|D_1| \neq 0$,则直接跳到 step 4 输出结果;

step3 while ($U > L + 1$) 令 $r = d^{(\lceil (U+L)/2 \rceil)}$,利用 R -控制集算法计算出输出集合 $D^{(\lceil (U+L)/2 \rceil)}$ 若 $|D^{(\lceil (U+L)/2 \rceil)}| \neq 0, U = \lceil \frac{U+L}{2} \rceil$,否则, $L = \lceil \frac{U+L}{2} \rceil$

END

step4 若 $|D^U| = k$,则直接输出,否则,选举任意剩余节点补充至 D 。

R -控制集问题中通过 2.1.1 节动态规划计算任意两点间的加权响应时间,并得出任意两点间加权响应时间矩阵 M ,对于 M 中的任意元素,有约束的加权响应时间 r ,使得约束矩阵 G ,对于 $\forall v_i, v_j \in E$,若 $M(v_i, v_j) \leq r$,则 $G(v_i, v_j) = 1$,否则, $G(v_i, v_j) = 0$,1 表示 v_i, v_j 可以互相覆盖,0 表示节点 v_i, v_j 之间无法互相覆盖,根据相关 R -控制集问题的表述^[6],可将问题转化为以下整数规划。设列向量 $X^T = [x_1, x_2, \dots, x_n]$,当 $x_n = 1$ 时,对应 v_n 为中心控制节点,当 $x_n = 0$ 时,则不是中心控制节点,对于 X 有以下整数规划

$$\min \sum x_i \tag{4}$$

$$\text{s.t.} \begin{cases} G \cdot X \geq \begin{bmatrix} 1 \\ 1 \\ \vdots \end{bmatrix} \\ x_i = 1 \text{ 或者 } 0 \end{cases} \tag{5}$$

算式(4)表示最优化目标,即启用尽可能少的中心控制节点,(5)式表示其约束条件,即表示选择的中心控制节点集必须至少覆盖一次 V 中的节点。

根据相关文献[14],对于该整数规划问题,可以采用分支定界的思路进行求解,目标函数的系数全为 1 情况下,可以使用隐枚举的方法进行求解^[15],但是该方法仅针对整数规划进行讨论,没有结合 K -中心问题的一些特性,借鉴隐枚举中的思路,增设条件,若目前枚举的输出集大于要求的 K 则停止当前分支的枚举,算法伪代码如算法 2 所示。

算法 2 限定控制集合数量的 R -控制集算法:

step1 map \leftarrow 动态规划计算的加权响应时间矩阵, $k \leftarrow K$

step 2 for($i < |E|$) // 变换为约束矩阵

for($j < |E|$)

if($\text{map}(i, j) \leq r$) $G(i, j) = 1$

else $G(i,j) = 0$

end

end

step3 // 形成空集根层节点, before = 0, behind = 1, record, $i = 1$, mapTree[0] 清空用以表示控制集

while ($i \leq |E|$)

// 插入第 i 号节点, record = behind

while (before < record)

if (mapTree[before] 的个数 < k) // 一个插入 v_i 节点, 另一个不插入 v_i 节点, 分别填入 mapTree[behind + 1] 和 mapTree[behind + 2] 中, 推进 behind, 并记录前一个的结果 z

// 检查前一个的结果

if (z 的结果符合约束条件) // 结束并输出结果 z

end

end

before ++

end

// 终止条件检查

// 若 record = behind, 则返回空集并终止

End

// 返回空集

第一步是变量赋值, 第二步是将距离矩阵 M 变为约束矩阵 G , 第三步采用分支定界的思路对整数规划问题进行讨论, 若没有符合约束条件的解, 则输出空集, 否则输出可行解。

该算法的核心是当枚举控制集中的个数等于 K 时, 若其不符合约束条件, 则其分支不再插入新的节点, 当一个结果符合约束条件并小于要求的 K 时, 则认为找到符合要求的解并直接输出。

相较于文献[13]中的算法, 算法 1 延续了其二分法查找的思想, 算法 2 主要是改变了对 R -控制集问题的求解方法, 文献[13]中使用动态规划将树状网络分层, 再将分层的树状网络逐级求解, 计算出中心节点。算法 2 使用动态规划方法得出任意节点间的最小加权响应时间, 然后通过整数规划的思路解决问题。相较于之前只适用于求解树状网络拓扑的方案, 本文方案能够求出非树状网络图的 K -中心问题, 有更好的适应范围。

2.2 $K=1$ 下基于动态规划的算法设计

若所求控制集 $K=1$, 可以直接通过矩阵 M 直接

得出结果。

算法 3 特殊求解算法

step1 按照动态规划得出矩阵 M

step2 $Out \leftarrow \infty$

step3 // 找出横排中最大的元素

// 找出这些最大中最小的, 记为 M_i

$Out \leftarrow M_i$

该算法的核心思想是先找寻所有节点的最大加权响应时间, 然后将所有节点的最小的加权响应时间进行比对, 找出其中最小的, 选中的节点就是中心控制节点。

算法 3 仅使用了简单的动态规划求出任意两节点间的加权响应时间矩阵 M , 并通过查询直接找出每个节点的最大加权响应时间, 从而获得最小的最大加权响应时间, 相较于文献[13]中的算法, 本算法拥有较低的复杂度, 且适用于非树状网络, 但只适用于 $K=1$ 时的情况。

2.3 算法性能分析

定理 1 在算法 2 中, 若输出为 \emptyset , 则不存在一个满足条件的控制集 D , 使得 $|D| \leq K$ 。

其证明如下(反证法):

若 $\exists D$ 满足最优性, 且 $|D| \leq K$, 且 $D = \{x_1, x_2, \dots, x_{n-1}, x_n\}$, 则存在子集 $D_1 = \{x_1, x_2, \dots, x_{n-1}\}$, 且 $|D_1| < K$, 以此类推, 对于 $D_{n-1} = \{x_1\}$, 且 $|D_{n-1}| < K$, 因此必然能存在推导路径 $\emptyset \rightarrow D_{n-1} \rightarrow \dots \rightarrow D_1 \rightarrow D$, 与输出集为 \emptyset 的结论不符合, 矛盾, 因此结论成立。

定理 2 算法 1 求得结果为最优解。

设该问题的最优值为 x_0 , 而算法 1 输出值为 x_1 , 在算法 3 中, 若直接在 step2 中输出集合 $|D_1| \neq 0$, 则该结果没有比起始元素更小, 必然为最优解, 则必然有 $x_0 = x_1$, 若 $|D_1| = 0$, 由于 x_0 时最优的, 必然有 $x_0 \leq x_1$, 而根据定理 1 中的相关内容, 且迭代后有 $U = L + 1$, $|D_L| = 0$, 因此仅当 $x_0 \geq x_1$ 时才存在 $|D_u| \leq K$, 综上所述, $x_0 = x_1$ 。

定理 3 对于 $K = k$, 算法 1 的时间复杂度为 $O(n^{k+2} \log_2 n)$ 。

首先, 对于 $K = k$ 的 K -问题, 算法主要步骤集中在 step3 的双循环下的演算步骤中, 在最差情况下, 输出组合为一空集时, 终端排列组合的情况有 C_n^k 种解的存在, 各自解解还有其对应的子集, 即一共需要验证 kC_n^k 个结果, 检查方案为将序列的每个位置与矩阵 G 中的各个行向量对应相乘, 并判断是否均大于

1,因此,该部分对应的时间复杂度为 $O(n^2)$,又因为 $kC_n^k < kn^k$, 因此该段代码的时间复杂度为 $O(n^{k+2})$ 。

接着,对于 $K=k$ 的算法 1 问题,step1 对于 $K=k$, 下动态规划的距离计算方法时间复杂度为 $O(n^3)$, 再将 $O(n^2)$ 个数据排列,则该阶段的时间复杂度 $O(n^2 \log_2 n)$,之后在 step2 中循环次数为 $O(\log_2 n)$, 根据定理 2 总共的时间复杂度为 $O(n^{k+2} \log_2 n)$, 因此,综合的时间复杂度为 $O(n^{k+2} \log_2 n)$ 。

定理 4 算法 3 的时间复杂度为 $O(n^3)$ 。

对于算法 3,其动态规划的时间复杂度为 $O(n^3)$,step2 中需要对 $O(n^2)$ 进行比较,因此这一步的时间复杂度为 $O(n^2)$,因此整体的时间复杂度为 $O(n^3)$ 。

3 结果仿真

3.1 仿真系统搭建

首先是比较各种中心控制节点在 $K=1$ 的等权情况下的仿真效果,仿真系统中有 $n+1$ 个节点,其中包含 1 个仅有 1 个网卡的桥节点、 $0.25n$ 个多网卡的桥节点的设备、 $0.75n$ 个叶子节点, $n=4,8,12,16,20$ 。比较在这些情况下的动态规划方案、特殊求解方案、随机中心控制节点方案,测度决定方案的最大加权响应时间的最小值。

其次是比较 $K=1$ 情况下,非等边权情况下测试各个方案的性能,引入部分蓝牙节点改变边权值,根据带宽计算,Wi-Fi 系统的标准带宽为 20 MHz, $0.75n$ 个终端中, $0.25n$ 个是 2 MHz 带宽的低功耗蓝牙 (bluetooth low energy, BLE), $0.25n$ 个 1 MHz 带宽的蓝牙增强速率 (enhanced data rate, EDR), $0.25n$ 个 Wi-Fi,假设 Wi-Fi 传输的权值为 1,则根据蓝牙的带宽与 Wi-Fi 的带宽比值,蓝牙应当约为 10 和 20,节点个数配置与上一个测试情况一致。

最后在相同连通图的情况下,测试 K 的变化对提出算法效果的影响,即在节点个数为 $n=4,8,12,16,20$ 时,测试 $K=1,2,3,4$ 对仿真结果的影响。

3.2 仿真结果

仿真系统下,节点之间的连接方式如图 1 中所示, $K=1$ 的等权情况下的仿真结果如图 3 所示。

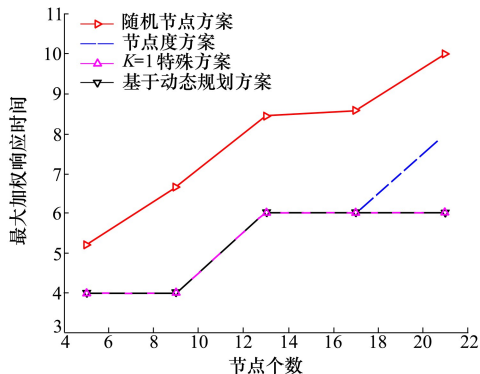


图 3 $K=1$ 时等权情况最大加权响应时间随节点数量变化曲线

图 3 中横坐标代表仿真场景中节点的数量,纵坐标代表了该方案下产生的中心控制节点的最小化最大响应时间,数值越大,网络拓扑中实际最大的加权响应时间越大。该图仿真的是在选择一个中心控制节点并在网络图节点权值与边权值均为 1 时,各个方案的性能指标。从图中可知,随机节点方案其最大相应的均值在同节点数情况下一定大于其他方案,其他 3 种方案在节点数小于等于 17 的结果均一致,节点数大于 17 之后,节点度方案高于特殊求解方案和动态规划方案,当节点数从 5 到 9 时,由于节点是均匀对称地插入,最大响应时延没有改变,如图 4 所示,假设节点权值和边权值均为 1,节点右上角为该节点对应的最大加权响应时间,左侧最优的节点为 1 或者 2,以该节点为中心控制节点最大加权响应时间为 4,而右侧最优为 2,同理最大响应时间为 4。

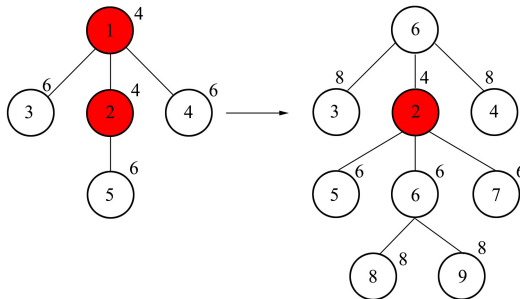


图 4 插入节点变化图

$K=1$ 的非等边权情况下的仿真结果如图 5 所示。

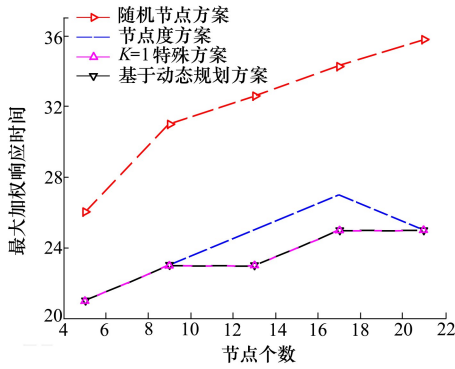


图 5 $K=1$ 时非等权时最大加权响应时间随节点数变化曲线

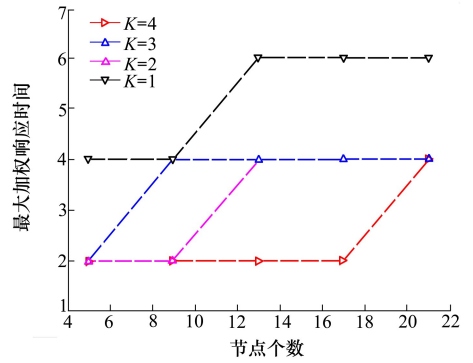


图 6 $K \neq 1$ 时最大加权响应时间随节点数变化曲线

图 5 中横坐标代表仿真场景中节点的数量,纵坐标代表了该方案下产生的中心控制节点最小化最大响应时间。该图仿真的是选择一个中心控制节点时,节点权值为 1,但边权值不一致时各个方案性能的差异,从图 5 中可以看出随机节点选举的方案总体大于其他 2 种方案,整体随着节点增加而增大,而节点度方案在 $n=9$ 时与基于动态规划方案偏离整体大于等于动态规划方案,并在 $n=19$ 处出现拐点。

图 6 中横坐标代表仿真场景中节点的数量,纵坐标代表了不同 K 值下产生的中心控制节点最小化最大响应时间。该图仿真选择的是一个中心控制节点时,节点权值为 1,但边权值不一致时不同中心控制节点数量的性能差异。根据图 6 所示, $K=1$ 的曲线总大于等于其他曲线 $n=5$ 时, $K=2$ 偏离其他曲线,总体大于等于 $K=3$ 以及 $K=4$, $n=9$ 时, $K=3$ 分离,总体大于等于 $K=4$ 。

4 结 论

针对最小化网络中任意节点到达中心控制节点的最大加权响应时间的问题,本文提出了一种基于动态规划的中心控制节点选举算法。首先,将无线网络中的节点和链路的响应时间建模为无向加权网络图中的节点权值和链路权值,并将加权响应时间表示为路径上节点权值与链路权值之和。然后,采用基于动态规划的插点法可求出任意 2 个点之间的距离,将问题转化为多个整数规划问题,并采用分支定界的方法逐个求解每个整数规划问题。最后,当 $K=1$ 时提出了上述算法的简化实现方法,并证明了所提算法的相关性能。

仿真结果表明,动态规划与特殊处理法效果优于测度优先以及随机节点选举法。前两者最大加权响应时间,优于节点度方案与随机节点方案的最大加权响应时间,具有更好的延时特性。而相较于文献[13]中的算法,本文中设计的 2 种算法能够适用于非树状网络拓扑中。

参考文献:

[1] 林华, 薛静宜. 软件定义网络(SDN)/网络功能虚拟化(NFV)技术研究及部署[J]. 广播电视网络, 2022, 29(1): 106-108
 LIN Hua, XUE Jingyi. Research and deployment of software definition network(SDN)/network function virtualization(NFV) technology[J]. Radio and Television Network, 2022, 29(1): 106-108 (in Chinese)

[2] 杨坤, 吴昊, 刘松洁, 等. 网络中心节点的确定方法、装置及设备节点: 中国, CN108075912A[P]. 2018-05-25

[3] 孙耀, 高孟杰, 秦爽, 等. 一种基于无线自组织网络的分布式簇头选举方法: 中国, CN110943920B[P]. 2020-10-27

[4] ABIDI W, EZZEDINE T. Fuzzy cluster head election algorithm based on LEACH protocol for wireless sensor networks[C]//2017 13th International Wireless Communications and Mobile Computing Conference, 2017: 993-997

[5] DJOUAMA A, ABDENNEBI M. Clusterhead selection algorithm based on energy and mobility in wireless mobile sensor networks [C]//2019 IEEE Symposium on Computers and Communications, 2019: 982-986

- [6] 李建平, 刘旭, 朱娟萍. 赋权树状网络中 R -控制集问题和 K -中心问题[J]. 运筹学学报, 2009, 13(2): 111-118
LI Jianping, LIU Xu, ZHU Juanping. R -control set problem and K -center problem in weighted tree network[J]. Journal of Operations Research, 2009, 13(2): 111-118 (in Chinese)
- [7] 贾银亮, 张驰宇, 梁康武. 基于多重竞争的无线传感器网络簇头选择算法[J]. 传感器与微系统, 2018, 37(2): 140-142
JIA Yinliang, ZHANG Chiyu, LIANG Kangwu. Cluster head selection algorithm for wireless sensor networks based on multiple competition[J]. Sensors and Microsystems, 2018, 37(2): 140-142 (in Chinese)
- [8] 魏长宝, 陈中良, 姚汝贤. 车联网中基于模糊逻辑的簇头选择算法[J]. 测控技术, 2016, 35(9): 83-86
WEI Changbao, CHEN Zhongliang, YAO Ruxian. Cluster head selection algorithm based on fuzzy logic in vehicle network[J]. Measurement and Control Technology, 2016, 35(9): 83-86 (in Chinese)
- [9] 刘晓理, 王出航, 王雪. 基于混沌粒子群的改进 LEACH 算法[J]. 长春工业大学学报, 2021, 42(5): 451-454
LIU Xiaoli, WANG Chuhang, WANG Xue. Improved LEACH algorithm based on chaotic particle swarm optimization[J]. Journal of Changchun University of Technology, 2021, 42(5): 451-454 (in Chinese)
- [10] 刘涛, 庞博. 基于遗传算法的异构无线传感器网络分簇算法[J]. 现代电子技术, 2022, 45(5): 25-30
LIU Tao, PANG Bo. Clustering algorithm for heterogeneous wireless sensor networks based on genetic algorithm[J]. Modern Electronic Technology, 2022, 45(5): 25-30 (in Chinese)
- [11] HAKIMI S. Optimal locations of switching centers and the absolute centers and medians of a graph[J]. Operations Research, 1964, 12(3): 450-459
- [12] 黄书强, 江秀美, 范人胜. 一种加权距离连续 K -中心选址问题求解方法[J]. 小型微型计算机系统, 2020, 41(2): 310-315
HUANG Shuqiang, JIANG Xiumei, FAN Rensheng. A weighted distance continuous K -center location problem solving method [J]. Minicomputer System, 2020, 41(2): 310-315 (in Chinese)
- [13] 刘旭. 赋权树状网络中 R -控制集问题和 K -中心问题[D]. 昆明: 云南大学, 2007
LIU Xu. R -control set problem and K -center problem in weighted tree network[D]. Kunming: Yunnan University, 2007 (in Chinese)
- [14] 郝自军, 何尚录. 最短路问题的 Floyd 算法的若干讨论[J]. 重庆工学院学报, 2008, 138(5): 156-159
HAO Zijun, HE Shanglu. Some discussion on Floyd algorithm for the shortest problem[J]. Journal of Chongqing of Technology, 2008, 138(5): 156-159 (in Chinese)
- [15] 姚恩瑜, 何勇, 陈仕平. 数学规划与组合优化[M]. 杭州: 浙江大学出版社, 2001
YAO Enyu, HE Yong, CHEN Shiping. Mathematical planning and combination optimization [M]. Hangzhou: Zhejiang University Press, 2001 (in Chinese)

Central node selection algorithm of minimizing maximum weighted response time based on dynamic programming

WAN Bailin, YANG Qi, YAN Zhongjiang, YANG Mao, LI Bo

(School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: In order to minimize the maximum weighted response time for any node in the network to reach the central control node, a central control node election algorithm based on dynamic programming is proposed. Firstly, the response times of nodes and links in the wireless network are modeled as the node weights and edge weights in the network topology, and then the central control node election problem that minimizes the maximum weighted response time for any node in the network to reach the central control node is modeled as the central problem, where represents the number of central control nodes. Then, by using the interpolation method based on dynamic programming, the weighted response time between the two points can be obtained, and the K -central problem modeled is transformed into several R -control set problems. Then, several control R -set problems are transformed into several 0-1 integer programming problems, and each integer programming problem can be solved one by one by using the branch and bound method. Finally, a simplified implementation method based on the above algorithm is given when $K=1$, the optimality of the proposed algorithm is proved and the complexity of the algorithm is analyzed. Simulation results show that the proposed central control algorithm can minimize the maximum weighted response time of the network.

Keywords: wireless network; central node selection; dynamic programming

引用格式: 万柏麟, 杨奇, 闫中江, 等. 一种基于动态规划的最小化最大加权响应时间的中心控制节点选举算法[J]. 西北工业大学学报, 2023, 41(1): 73-80

WAN Bailin, YANG Qi, YAN Zhongjiang, et al. Central node selection algorithm of minimizing maximum weighted response time based on dynamic programming[J]. Journal of Northwestern Polytechnical University, 2023, 41(1): 73-80 (in Chinese)