

无线传感网中节点能量和链路容量约束的二阶分布式流控制方法

穆元彬, 史清江, 徐伟强, 余晴里, 汪亚明

(浙江理工大学信息学院, 浙江杭州 310018)

摘要: 分布式流控制设计是确保无线传感网络性能的关键问题. 本文针对无线传感网络中的分布式流控制设计问题, 建立了一种节点能量和链路容量共同约束的流控制优化模型. 由于对偶次梯度分解算法存在收敛速度慢、步长选择敏感等缺点, 本文设计了基于矩阵分裂的分布式牛顿算法来实现网络效用最大化. 该算法具有与对偶次梯度分解算法几乎相同的通信复杂度, 但能够达到快速的二次收敛速度. 仿真结果表明, 该算法的收敛性能明显优于对偶次梯度算法.

关键词: 无线传感网络; 矩阵分裂; 分布式牛顿算法; 网络效用

中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112 (2015)05-1000-07

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2015.05.1025

Node Energy and Link Capacity Constrained Distributed Flow Control in Wireless Sensor Networks: A Second-Order Method

MU Yuan-bin, SHI Qing-jiang, XU Wei-qiang, YU Qing-li, WANG Ya-ming

(College of Information, Zhejiang Sci-Tech University, Hangzhou, Zhejiang 310018, China)

Abstract: Distributed flow control is a key issue in wireless sensor network design to achieve good network performance. This paper presents a node energy and link capacity constrained flow control optimization model for distributed flow control in wireless sensor networks. Since subgradient-based dual-decomposition method suffers from slow convergence rate and sensitivity in step-size choosing, this paper adopts matrix-splitting based distributed Newton algorithm to maximize the network utility. The proposed algorithm can achieve quadratic convergence rate while requiring almost the same communication overhead as the subgradient-based dual-decomposition algorithm. Moreover, it is shown by simulations that the proposed algorithm has a much better convergence performance than the subgradient-based dual-decomposition algorithm.

Key words: wireless sensor networks; matrix splitting; distributed Newton method; network utility

1 引言

无线传感网络(WSN)是一种集数据采集、信号处理和数据传输功能于一体的智能化信息网络系统. 这种网络由大量资源有限、价格低廉的无线节点通过自组网方式组成, 具有低功耗、低速率、高灵活和快速组网的特点. WSN在军事、工业、医疗、运输等领域具有非常广泛的应用前景^[1,2]. 随着网络数据通信业务的不断增加, 集中式处理整个网络的数据已经相当困难, 因此, 如何通过局部信息交互来分布式解决网络实际问题, 便成为网络设计中的一个关键问题.

影响网络性能的因素众多, 在这众多因素中, 一个

最重要的性能指标就是数据传输速率. 为此, 越来越多的研究人员致力于资源分配^[3,4]、拥塞控制^[5-7]、数据传输^[8]和最大化寿命路由^[9]的研究, 运用网络效用最大化方法来实现流速率的有效分配. 由于无线传感网主要通过能量有限的电池进行供电, 近年来许多研究者开始致力于无线传感网络中流速率控制问题的研究. He SB等人^[10]首次提出了无线传感网络中联合物理层、MAC层和链路层协同优化的网络寿命最大化问题, 并设计了高效的启发式算法, 提高了网络的整体性能. Abdulla A等人^[11]设计了一种混合多跳路由算法, 解决了无线传感网络中寿命最大和资源有限约束下的流控制问题. Chen JM等人^[12]研究了无线传感网络中网络寿命和信道干

扰联合约束下的异步流控制算法. 以上这些研究都采用了传统的对偶分解算法和次梯度算法. 虽然通过局部信息交互, 这些算法能够实现分布式求解, 然而在网络执行时存在收敛速度慢、步长选择敏感等缺陷.

20 世纪 80 年代 Bertsekas 和 Klinecicz 等人开始用牛顿算法解决网络优化问题^[13,14], 然而其均没有考虑如何分布式实施, 并且这些研究都使用投影梯度来寻找可行的迭代方向, 而 Wei E 等人^[15]的研究表明内点法效率更高. 文献[16]中作者基于高斯置信算法研究了网络效用最大化问题的分布式求解算法, 虽然实际操作中算法效果明显, 却不能保证完全收敛. 文献[17]中作者针对网络流控制问题模型设计了一种分布式牛顿算法, 但其没有考虑到信道容量不能取负值, 并且该方法在实际操作时收敛速度比较慢. 以上所有这些研究都没有考虑节点能量限制, 而无线传感网主要通过能量有限的电池进行供电, 因此这些研究都不适用于无线传感网络.

针对上述不足, 本文设计了具有二阶收敛性能的牛顿算法对该优化问题进行分布式求解, 以实现网络效用最大化. 通过文献调研发现, 目前国际上关于分布式牛顿算法方面的研究仍处于初级阶段, 国内关于这方面的研究还非常稀少. 本文将引入矩阵分裂技术, 通过矩阵分裂^[18], 实现原始变量和对偶变量的分布式计算.

2 问题描述

2.1 网络模型

图 1 为一个简单的无线传感网络拓扑图. 文章用 $\mathcal{G} = \{\mathcal{N}, \mathcal{L}\}$ 表示路由确定的无线传感网络, 其中集合 $\mathcal{N} = \{1, \dots, N\}$ 表示维度为 N 的无线传感节点集, 集合 $\mathcal{L} = \{1, \dots, L\}$ 表示维度为 L 的网络链路集, 定义列向量 $\mathbf{c} = [c_l]_{l \in \mathcal{L}}$ 为所有链路的最大固定容量. 为区别网络中的普通节点, 令 $\mathcal{S} = \{1, \dots, S\}$, $s \in \mathcal{S}$ 表示源节点集.

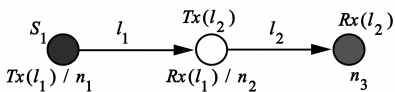


图1 简单无线传感网络拓扑图

如图 1 所示, 对于网络中的某条链路 l , 本文用符号 $Tx(l)$ 表示链路 l 的流入节点, $Rx(l)$ 表示链路 l 的流出节点, $\mathcal{S}(l)$ 表示传输信息流时所有经过链路 l 的源节点集, 类似的, 对于网络中的源节点 i , 文章用 $\mathcal{L}(i)$ 表示源节点 i 传输路由上的链路集, $\mathcal{N}(i)$ 表示源节点 i 传输路由上的传感节点集, 列向量 $\mathbf{s} = [s_i]_{i \in \mathcal{S}}$ 表示取值非负的流速率. 为方便描述, 引入矩阵 $\mathbf{R} \in \mathbb{R}^{L \times S}$ 表示网络拓扑的路由矩阵, 其中 \mathbf{R} 中第 l 行 j 列元素为,

$$R_{lj} = \begin{cases} 1, & \text{链路 } l \text{ 在源节点 } j \text{ 路由上} \\ 0, & \text{其他} \end{cases} \quad (1)$$

2.2 能量模型

假设网络是个连通的流守恒系统. 对于网络中某个传感节点 n , 令 e_n 表示该传感节点所具有的初始能量, e^s 表示其空闲状态时单位时间内消耗的能量, 同时令 e^l 和 e^r 分别表示该传感节点单位时间内发送和接收单位信息流所消耗的能量, T_n^{goal} 表示传感节点 n 的预定寿命. 则对于节点 n , 根据假设, 其消耗的能量主要有三部分组成^[12]: (1) 传输数据流消耗的能量; (2) 接收数据流消耗的能量; (3) 空闲状态时消耗的能量. 因此, 节点 n 的能量限制条件可以表示如下,

$$e^l \sum_{l \in \mathcal{O}(n)} \sum_{i \in \mathcal{S}(l)} s_i + e^r \sum_{l \in \mathcal{I}(n)} \sum_{i \in \mathcal{S}(l)} s_i + e^s \leq e_n^{goal}, \quad \forall n = 1, \dots, N \quad (2)$$

其中, $e_n^{goal} = e_n / T_n^{goal}$, $\mathcal{O}(n)$ 表示节点 n 的所有流出链路集, $\mathcal{I}(n)$ 表示节点 n 的所有流入链路集.

定义列向量 $\mathbf{e} = [e_1^{goal} - e^s, \dots, e_N^{goal} - e^s]^T$ 和矩阵 $\tilde{\mathbf{R}} \in \mathbb{R}^{N \times S}$, 其中 $\tilde{\mathbf{R}}$ 中第 n 行第 s 列的元素 \tilde{R}_{ns} 为,

$$\tilde{R}_{ns} = \begin{cases} e^l + e^r, & \text{节点 } n \text{ 为会话 } s \text{ 的中继节点} \\ e^l, & \text{节点 } n \text{ 为会话 } s \text{ 的源节点} \\ e^r, & \text{节点 } n \text{ 为会话 } s \text{ 的目的节点} \\ 0, & \text{其他} \end{cases} \quad (3)$$

根据上面定义, 不等式约束条件(2)可以用矩阵形式表示如下,

$$\tilde{\mathbf{R}} \cdot \mathbf{s} \leq \mathbf{e} \quad (4)$$

2.3 容量模型

假设无线传感网络链路的容量有限, 那么无线传感网络传输信息流时, 流经某条链路 l 的所有源数据流之和不应该超过该条链路的最大容量 c_l . 因此, 链路容量约束条件表示如下,

$$\sum_{j=1}^S R_{lj} s_j \leq c_l, \quad \forall l = 1, \dots, L \quad (5)$$

其中, S 表示网络中源节点的总个数, 即总会话数.

根据公式(1)中路由矩阵 \mathbf{R} 的定义, (5)式可以表示成如下矩阵形式,

$$\mathbf{R} \cdot \mathbf{s} \leq \mathbf{c} \quad (6)$$

2.4 效用最大化问题模型

定义函数 $U_i(s_i): \mathbb{R}_+ \rightarrow \mathbb{R}$ 表示源节点 i 流速率的效用函数. 假设整个网络的效用为各源节点流速率效用之和. 那么, 最终的网络效用最大化问题如下所示,

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^S U_i(s_i) \\ & \text{subject to} && \sum_{j \in \mathcal{S}(l)} s_j \leq c_l, \quad \forall l = 1, \dots, L \\ & && e^l \sum_{l \in \mathcal{O}(n)} \sum_{i \in \mathcal{S}(l)} s_i + e^r \sum_{l \in \mathcal{I}(n)} \sum_{i \in \mathcal{S}(l)} s_i + e^s \leq e_n^{goal}, \end{aligned}$$

$$\forall n = 1, \dots, N$$

$$s_i \geq 0, \quad \forall i = 1, \dots, S \quad (7)$$

令矩阵 $\mathbf{M} = [\mathbf{R}^T, \tilde{\mathbf{R}}^T]^T$, $\mathbf{M} \in \mathbb{R}^{(L+N) \times S}$, 向量 $\mathbf{d} = [\mathbf{c}^T, \mathbf{e}^T]^T$, $\mathbf{d} \in \mathbb{R}^{(L+N) \times 1}$, 则上面优化问题可用矩阵形式表示如下,

$$\begin{aligned} & \text{maximize} \quad \sum_{i=1}^S U_i(s_i) \\ & \text{subject to} \quad \mathbf{M} \cdot \mathbf{s} \leq \mathbf{d} \\ & \quad \quad \quad s_i \geq 0, \quad \forall i = 1, \dots, S \end{aligned} \quad (8)$$

与传统模型相比,考虑电池能量资源利用率是本模型的优势所在.需要指出,用来解决这一问题的常规算法是基于次梯度的原始-对偶分解算法,虽然通过对偶变量的局部信息交互,该算法可以在网络内分布式执行,然而在实际操作中,算法收敛速度较慢,并且对步长选择非常敏感.本文则通过快速分布式牛顿算法解决这一优化问题.特别需要指出的是,与文献[15]不同,本文由于能量约束的引入使得分布式牛顿算法的设计及其在网络内的执行完全不同.

为方便使用分布式牛顿算法,文章对优化问题目标函数做出如下标准假设^[19]:目标函数为严格凹、单调递增、二次连续可微,并且 $-U_i(s_i): \mathbb{R}_+ \rightarrow \mathbb{R}$ 为自治函数.由于问题(8)中所有限制条件均为线性约束条件,所以该问题是个凸优化问题.

3 问题转化及中心牛顿算法

3.1 问题转化

为了用牛顿算法求解原始问题(8),文章首先通过引入松弛变量和障碍函数,把该问题转化成等式约束条件下的最小化问题.

令 $\mathbf{x} = [\mathbf{s}^T, \mathbf{y}^T]^T$, $\mathbf{x} \in \mathbb{R}^{(L+N+S) \times 1}$, 其中 $\mathbf{y} = [y_1, \dots, y_{L_n}]^T \geq 0$ 是针对约束 $\mathbf{M}\mathbf{s} \leq \mathbf{d}$ 引入的松弛变量, L_n 表示松弛变量 \mathbf{y} 中元素的总个数.同时定义矩阵 $\mathbf{F} = [\mathbf{M}\mathbf{I}(L_n)]$, 其中 $\mathbf{F} \in \mathbb{R}^{(L+N) \times (L+N+S)}$, $\mathbf{I}(n)$ 表示 $n \times n$ 维的单位矩阵,则不等式限制条件 $\mathbf{M}\mathbf{s} \leq \mathbf{d}$ 可转化为等式限制条件 $\mathbf{F}\mathbf{x} = \mathbf{d}$.

再引入障碍函数,原始问题可以等价地转化成如下等式约束最小化问题,

$$\begin{aligned} & \min f(\mathbf{x}) \\ & \text{s.t.} \quad \mathbf{F}\mathbf{x} = \mathbf{d} \end{aligned} \quad (9)$$

其中, $f(\mathbf{x}) = -\mu \sum_{i=1}^S U_i(x_i) - \sum_{i=1}^{S+L_n} \log(x_i)$, $\mu > 0$ 表示惩罚因子.

参数 μ 决定问题(9)的求解精度,随着 μ 的不断增大, $f(\mathbf{x})$ 中第一部分对障碍函数起决定作用,当 $\mu \rightarrow \infty$ 时,问题(9)的求解结果近似等于问题(8)的求解结果.

3.2 集中式牛顿算法

给定严格满足限制条件的初始值 \mathbf{x}_0 , 问题(9)中自变量的最优解可通过下式迭代得到,

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \lambda^k \Delta \mathbf{x}^k \quad (10)$$

式(10)中 $\lambda^k \geq 0$ 表示牛顿迭代步长, $\Delta \mathbf{x}^k$ 表示牛顿迭代方向,根据 $f(\mathbf{x})$ 的泰勒二次展开式和问题(9)的 KKT 方程^[19], $\Delta \mathbf{x}^k$ 可以由线性方程组(11)求解得到,

$$\begin{pmatrix} \mathbf{H}_k & \mathbf{F}^T \\ \mathbf{F} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta \mathbf{x}^k \\ \boldsymbol{\omega}^k \end{pmatrix} = - \begin{pmatrix} \nabla f(\mathbf{x}^k) \\ \mathbf{F}\mathbf{x}^k - \mathbf{d} \end{pmatrix} \quad (11)$$

上式中, $\boldsymbol{\omega}^k = [\boldsymbol{\omega}_m^k]_{m \in \mathcal{L} \cup \mathcal{N}}$, $\boldsymbol{\omega}^k \in \mathbb{R}^{(L+N) \times 1}$ 表示与等式约束 $\mathbf{F}\mathbf{x} = \mathbf{d}$ 有关的对偶变量,物理上可认为网络中源节点在传输信息流时使用链路容量和消耗节点能量的价格,符号 \mathbf{H}_k 表示目标函数 $f(\mathbf{x})$ 对向量 \mathbf{x} 的二阶导数.由于目标函数 $f(\mathbf{x})$ 对各个源节点是相互独立的,所以 \mathbf{H}_k 是一个对角矩阵,而其逆矩阵 \mathbf{H}_k^{-1} 的对角元素为:

$$[\mathbf{H}_k^{-1}]_{ii} = \left(\frac{\partial^2 f}{(\partial x_i^k)^2} \right)^{-1}.$$

求解线性方程组(11)可得,

$$\Delta \mathbf{x}^k = -\mathbf{H}_k^{-1}(\nabla f(\mathbf{x}^k) + \mathbf{F}^T \boldsymbol{\omega}^k) \quad (12)$$

$$\mathbf{F}\mathbf{H}_k^{-1}\mathbf{F}^T \boldsymbol{\omega}^k = -\mathbf{F}\mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k) \quad (13)$$

由于海森矩阵 \mathbf{H}_k 为对角矩阵,根据网络拓扑结构和系数矩阵 $\mathbf{F} = [\mathbf{M}\mathbf{I}(L_n)]$ 可知,公式(12)很容易通过局部信息求解得到,具体的分布式计算过程将会在 4.3.1 节给出具体讨论.然而根据(13)式求解对偶变量 $\boldsymbol{\omega}^k$ 时,由于 $\mathbf{F}\mathbf{H}_k^{-1}\mathbf{F}^T$ 求逆需要用到全局信息,所以不能直接进行分布式执行,下文将进一步讨论如何分布式求解公式(13).

3.3 分布式牛顿算法

由 $f(\mathbf{x})$ 和 \mathbf{x} 的结构可得,

$$\begin{aligned} \nabla f(\mathbf{x}) &= \begin{cases} -\mu U'_i(x_i) - 1/x_i, & \text{当 } 1 \leq i \leq S \text{ 时} \\ -1/x_i, & \text{当 } S < i \leq S + L_n \text{ 时} \end{cases}; \\ \nabla^2 f(\mathbf{x}) &= \begin{cases} -\mu U''_i(x_i) + 1/x_i^2, & \text{当 } 1 \leq i \leq S \text{ 时} \\ 1/x_i^2, & \text{当 } S < i \leq S + L_n \text{ 时} \end{cases} \end{aligned}$$

其中 $\nabla f(\mathbf{x})$ 和 $\nabla^2 f(\mathbf{x})$ 分别表示函数 $f(\mathbf{x})$ 对向量 \mathbf{x} 中第 i 个元素 x_i 求一阶导和二阶导.

文章将使用公式迭代计算牛顿方向和对偶变量.下面具体讨论牛顿方向和对偶变量的分布式计算过程.

3.3.1 牛顿方向的分布式计算

根据式(12)可得,给定对偶变量结果后,牛顿方向可通过下面式子迭代计算得到,

$$\Delta x_i^k = -(\mathbf{H}_k^{-1})_{ii}(\nabla f(\mathbf{x}^k) + (\mathbf{F}^T \boldsymbol{\omega}^k)_i) \quad (14)$$

其中, $(\mathbf{F}^T \boldsymbol{\omega}^k)_i = [\mathbf{F}^T]^{[i]} \cdot \boldsymbol{\omega}^k = \sum_{m \in \mathcal{L}(i) \cup \mathcal{N}(i)} F_{mi} \cdot \boldsymbol{\omega}_m^k$

上式中 F_{mi} 表示源节点 i 路由上链路 m (或节点 m) 的价格加权系数, 因此 $(\mathbf{F}^T \omega^k)_i$ 可以由源节点 i 的目的节点收集节点 i 路由上的链路容量使用价格和节点能量使用价格计算得到, 又因源节点 i 已知 $(\mathbf{H}_k^{-1})_{ii}$ 和 $\nabla f(\mathbf{x}^k)$, 因此根据式(14), 牛顿迭代方向可以通过局部信息交互计算得到。

这样, 在对偶变量已知的前提下, 分布式牛顿算法可以简述如下:

给定严格可行的初始变量 \mathbf{x}_0 , 最优解可以通过式(15)迭代计算得到,

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \lambda^k \cdot \Delta \tilde{\mathbf{x}}^k \quad (15)$$

迭代步长 λ^k 可选为^[15,20],

$$\lambda^k = \begin{cases} \frac{5}{6(\theta^k + 1)}, & \text{如果 } \theta^k \geq \frac{1}{4} \\ 1, & \text{其他} \end{cases} \quad (16)$$

其中, 式(16)中, $\theta^k = \sqrt{(\Delta \tilde{\mathbf{x}}^k)^T \mathbf{H}_k \Delta \tilde{\mathbf{x}}^k}$ 表示 Newton Decrement^[19].

3.3.2 对偶变量的分布式计算

为了分布式求解公式(13), 首先简单介绍下矩阵分裂定理^[18], 这一定理最初用来求解线性等式系统. 考虑线性等式系统 $\mathbf{G}\mathbf{y} = \mathbf{a}$, 其中 \mathbf{G} 为 $n \times n$ 维矩阵, \mathbf{a} 为 n 维列向量, 假设矩阵 \mathbf{G} 可以表示为某个可逆矩阵 \mathbf{M} 和任一矩阵 \mathbf{N} 的和, 如 $\mathbf{G} = \mathbf{M} + \mathbf{N}$, 那么对于任一给定的初始 n 维列向量 \mathbf{y}_0 , 当 \mathbf{M} 和 \mathbf{N} 满足 $\rho(\mathbf{M}^{-1}\mathbf{N}) < 1$ 时, 迭代公式 $\mathbf{y}^{k+1} = -\mathbf{M}^{-1}\mathbf{N}\mathbf{y}^k + \mathbf{M}^{-1}\mathbf{a}$ 随着 k 的增加趋于收敛, 并且当 $k \rightarrow \infty$ 时, \mathbf{y}^k 的极限值即为 $\mathbf{G}\mathbf{y} = \mathbf{a}$ 的最优解。

本文使用矩阵分裂定理来求解对偶变量 ω . 定义矩阵 Λ_k 、 Ω_k 和 $\tilde{\Omega}_k$, 其中 Λ_k 和 $\tilde{\Omega}_k$ 为对角矩阵, 矩阵元素分别为 $(\Lambda_k)_{ii} = (\mathbf{F} \cdot \mathbf{H}_k^{-1} \cdot \mathbf{F}^T)_{ii}$, $(\tilde{\Omega}_k)_{ii} = \sum_j (\Omega_k)_{ij}$, $\Omega_k = \mathbf{F} \cdot \mathbf{H}_k^{-1} \cdot \mathbf{F}^T - \Lambda_k$.

定理 1 假设 $k > 0$, 矩阵 Λ_k 、 Ω_k 、 $\tilde{\Omega}_k$ 定义如上, 那么对于任意给定的初始对偶变量 $\omega(0)$, 迭代公式

$$\omega(t+1) = (\Lambda_k + \tilde{\Omega}_k)^{-1} (\Omega_k - \tilde{\Omega}_k) \omega(t) - (\Lambda_k + \tilde{\Omega}_k)^{-1} \mathbf{F} \mathbf{H}_k^{-1} \nabla f(\mathbf{x}^k) \quad (17)$$

随着 t 的增加趋于收敛, 并且当 $t \rightarrow \infty$ 时, 它的极限值即为式(13)的最优值。

其中, 定理中的 $\omega(t)$ 表示第 t 次迭代求得的对偶变量。

为说明上述定理成立, 首先介绍两个引理^[21].

引理 1 令矩阵 \mathbf{G} 为实对称矩阵, 如果 $\mathbf{G} = \mathbf{M} + \mathbf{N}$ 中矩阵 \mathbf{M} 可逆, 并且 $\mathbf{M} + \mathbf{N}$ 和 $\mathbf{M} - \mathbf{N}$ 均为正定矩阵, 那么 $\mathbf{M}^{-1}\mathbf{N}$ 的谱半径满足 $\rho(\mathbf{M}^{-1}\mathbf{N}) < 1$.

引理 2 如果一个对称矩阵 \mathbf{Q} 是严格对角优势矩

阵, 即满足 $|Q_{ii}| > \sum_{i \neq j} |Q_{ij}|$, 且对所有的 i 都有 $Q_{ii} > 0$, 则 \mathbf{Q} 是正定矩阵。

由于矩阵 $\mathbf{F} \mathbf{H}_k^{-1} \mathbf{F}^T$ 可表示成 $(\Lambda_k + \tilde{\Omega}_k) + (\Omega_k - \tilde{\Omega}_k)$, 根据 Λ_k 和 $\tilde{\Omega}_k$ 的定义不难发现 $\Lambda_k + 2\tilde{\Omega}_k - \Omega_k$ 为严格对角优势矩阵, 所以 $(\Lambda_k + \tilde{\Omega}_k) - (\Omega_k - \tilde{\Omega}_k)$ 为正定矩阵. 又因目标函数是严格凸函数, 且 \mathbf{H}_k 为对角矩阵, $\mathbf{F} \geq 0$ 满足行满秩, 所以 $\mathbf{F} \mathbf{H}_k^{-1} \mathbf{F}^T$ 为正定实对称矩阵. $(\Lambda_k + \tilde{\Omega}_k)$ 可逆, 故根据矩阵分裂定理和上面两个引理, 可知定理 1 成立。

定义 $\mathcal{M}(i)$ 表示源节点 i 传递信息时所经过的链路和节点集, $\phi_i(t) = [\mathbf{F}^T][i]\omega(t)$ 表示第 t 次迭代时源节点 i 的路由价格总和, 这里的价格包括路由上所有链路容量使用价格和所有节点能量使用价格总和, $\Phi_i(t) = (\mathbf{H}_k^{-1})_{ii} \sum_{m \in \mathcal{M}(i)} F_{mi} \omega_m(t)$ 表示第 t 次迭代时源节点 i 的加权价格总和。

按照上述定义, 公式(17)等价于式(18),

$$\omega_m(t+1) = \frac{1}{\sum_{i \in \mathcal{S}(m)} F_{mi} \cdot \Phi_i(0) + (\mathbf{H}_k^{-1})_{(S+m)(S+m)}} \left(\begin{aligned} & \left(\sum_{i \in \mathcal{S}(m)} F_{mi} \cdot \Phi_i(0) \right) \omega_m(t) - \sum_{i \in \mathcal{S}(m)} F_{mi} \cdot (\mathbf{H}_k^{-1})_{ii} \nabla f(\mathbf{x}^k) \\ & - \sum_{i \in \mathcal{S}(m)} F_{mi} \cdot \Phi_i(t) - (\mathbf{H}_k^{-1})_{(S+m)(S+m)} \nabla_{S+m} f(\mathbf{x}^k) \end{aligned} \right) \quad (18)$$

其中, $\mathcal{S}(m)$ 表示所有经过 $m \in \mathcal{L} \cup \mathcal{N}$ 的源节点集合。

简单介绍定理 1 中公式(17)如何分布式计算得到式(18). 只需依次给出 $(\Lambda_k + \tilde{\Omega}_k)_{mm}$ 、 $(\tilde{\Omega}_k)_{mm}$ 、 $(\Omega_k \omega(t))_m$ 和 $(\mathbf{F} \mathbf{H}_k^{-1} \nabla f(\mathbf{x}^k))_m$ 的分布式计算方法即可。

路由加权价格 $\Phi_i(t)$ 可进一步表示为 $\Phi_i(t) = (\mathbf{H}_k^{-1})_{ii} \phi_i(t) = [\mathbf{H}_k^{-1} \mathbf{F}^T][i]\omega(t)$, 把矩阵 \mathbf{H}_k^{-1} 、 \mathbf{F} 和 $\omega(t)$ 的结构代入此式并整理得到 $(\mathbf{F} \mathbf{H}_k^{-1} \mathbf{F}^T \omega(t))_m = \sum_{i \in \mathcal{S}(m)} F_{mi} \Phi_i(t) + (\mathbf{H}_k^{-1})_{(S+m)(S+m)} \omega_m(t)$. 令 $\omega(0) = [1, \dots, 1]^T$ 并代入 $(\mathbf{F} \mathbf{H}_k^{-1} \mathbf{F}^T \omega(0))_m$ 得 $(\mathbf{F} \mathbf{H}_k^{-1} \mathbf{F}^T \omega(0))_m = \sum_{i=1}^S F_{mi} \Phi_i(0) + (\mathbf{H}_k^{-1})_{(S+m)(S+m)}$. 把 $\mathbf{F} \mathbf{H}_k^{-1} \mathbf{F}^T = \Lambda_k + \Omega_k$ 代入上式又得 $(\mathbf{F} \mathbf{H}_k^{-1} \mathbf{F}^T \omega(0))_m = ((\Lambda_k + \Omega_k) \omega(0))_m = (\Lambda_k)_{mm} + \sum_j (\Omega_k)_{nj}$, 所以 $(\Lambda_k + \tilde{\Omega}_k)_{mm} = \sum_{i \in \mathcal{S}(m)} F_{mi} \Phi_i(0) + (\mathbf{H}_k^{-1})_{(S+m)(S+m)}$. 又因 $(\Lambda_k)_{mm}$ 结果已知, 故 $(\tilde{\Omega}_k)_{mm}$ 已知. 同理, 可得 $(\Omega_k \omega(t))_m$ 和 $(\mathbf{F} \mathbf{H}_k^{-1} \nabla f(\mathbf{x}^k))_m$ 结构分别为

$$\begin{aligned} (\Omega_k \omega(t))_m &= ((\mathbf{F} \mathbf{H}_k^{-1} \mathbf{F}^T - \Lambda_k) \omega(t))_m = \sum_{i \in \mathcal{S}(m)} F_{mi} \cdot \Phi_i(t) \\ & - \sum_{i \in \mathcal{S}(m)} F_{mi}^2 \cdot (\mathbf{H}_k^{-1})_{ii} \omega_m(t) \end{aligned} \quad (19)$$

$$\begin{aligned} (\mathbf{F} \mathbf{H}_k^{-1} \nabla f(\mathbf{x}^k))_m &= \sum_{i \in \mathcal{S}(m)} F_{mi} \cdot (\mathbf{H}_k^{-1})_{ii} \\ & \nabla_{mf} f(\mathbf{x}^k) + \nabla_{S+m}^2 f(\mathbf{x}^k) \nabla_{S+m} f(\mathbf{x}^k) \end{aligned} \quad (20)$$

根据(18)式,下面讨论对偶变量的分布式计算过程:

分布式计算过程

网络初始化:

(1)各源节点 i 计算 $(H_k)_{ii}$ 和 $\nabla f(\mathbf{x}^k)$, 并发送给所在路上的链路和节点 $m \in \mathcal{M}(i)$, 每个链路或节点 m 分别计算 $(H_k^{-1})_{(s+m)(s+m)}$, $(H_k^{-1})_{(s+m)(s+m)} \nabla_{s+mf}(\mathbf{x}^k)$, $\sum_{i \in \mathcal{S}(m)} F_{mi}^2 \cdot (H_k^{-1})_{ii}$ 和 $\sum_{i \in \mathcal{S}(m)} F_{mi} \cdot (H_k^{-1})_{ii} \nabla f(\mathbf{x}^k)$;

(2)各 m 初始化自己的容量或能量使用价格 $\omega_m(0) = 1$, 源节点 i 的目的节点收集 $\mathcal{M}(i)$ 上的路由价格得到 $\psi_i(0)$, 并将其发送给源节点 i , 源节点 i 计算 $\Phi_i(0)$, 然后再发送给其路上的各个链路和节点 m ;

(3)各链路或节点 $m \in \mathcal{M}(i)$ 根据(18)式求解 $\omega_m(1)$.

更新对偶变量:

(4)各链路或节点 $m \in \mathcal{M}(i)$ 根据公式(18)迭代更新 $\omega_m(t)$, 源 i 的目的节点收集 $\mathcal{M}(i)$ 上的路由价格得到 $\psi_i(t)$, 并将其广播至源节点;

(5)源节点 i 计算 $\Phi_i(t)$, 并发送给各链路或节点 $m \in \mathcal{M}(i)$;

(6)重复步骤(4), 至满足终止条件.

综上所述,对偶变量的计算也可通过局部信息交互迭代计算得到,因此通过上面对偶变量的计算,可以近似得到(13)式的最优解.

4 仿真验证

本节用 Matlab 仿真平台仿真验证分布式牛顿算法与对偶次梯度算法的性能. 仿真参数设置如下: 分布式牛顿算法中对偶变量设置为 $[1, \dots, 1]$, 对偶变量迭代误差小于 10^{-6} 时终止迭代, 原始变量迭代终止误差设置为 10^{-3} , 迭代步长按照公式(16)进行初始化. 对偶次梯度算法选择固定步长, 本文仿真时步长选择分别为 $st = 0.01$, $st = 0.02$, $st = 0.001$, $st = 0.002$, $st = 0.0001$, $st = 0.0002$. 为了公平起见, 对偶次梯度算法原始变量迭代终止误差同样设置为 10^{-3} . 文章仿真时优化问题效用目标函数选为对数函数 $\log(\mathbf{x}_i)$.

4.1 网络拓扑结构

如图2所示,七个无线传感节点随机分布在 $800\text{m} \times 800\text{m}$ 的方形区域内, 传感节点的最大初始能量设置为 10 焦耳, 能耗参数设置为 $e^l = e^r = 2.5$, 链路容量在 $(0, 10)$ 范围内随机取值, 网络拓扑中总共有 6 条会话 7 条链路, 源节点 $n1 \sim n6$ 以一定的流速率向目的结点 $n7$ 传输信息流, 它们各自的路由如图2中虚线箭头所示.

4.2 算法特性

本文首先以图2所示的网络拓扑结构为例仿真验证分布式牛顿算法的收敛特性. 收敛特性图如图3所示, 图像中每一个阶梯变化表示一次原始变量迭代, 而

阶梯跨越的长度表示矩阵分裂时对偶变量的迭代次数, 从图中可以看出, 分布式牛顿算法大约经过 780 步迭代后收敛到最优解, 其中图中的最优解用 CVX 工具箱求解^[22]. 注意, 与文献[20]不同, 此处的迭代次数不仅包括原始变量的迭代次数, 也包括对偶变量的迭代次数.

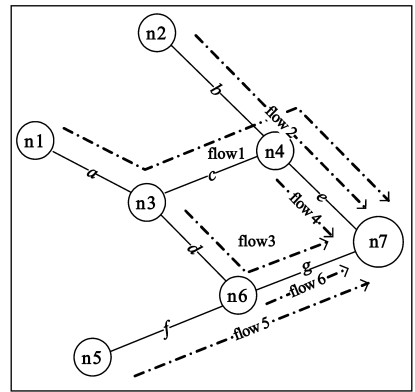


图2 网络拓扑结构

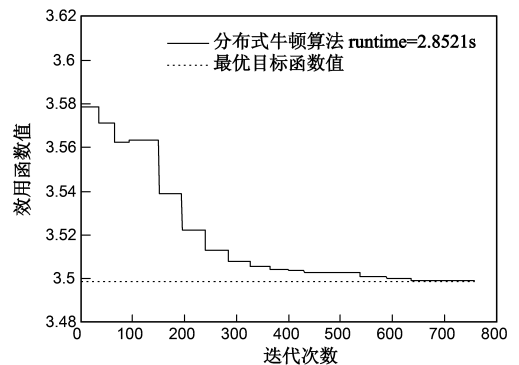


图3 分布式牛顿算法的收敛特性

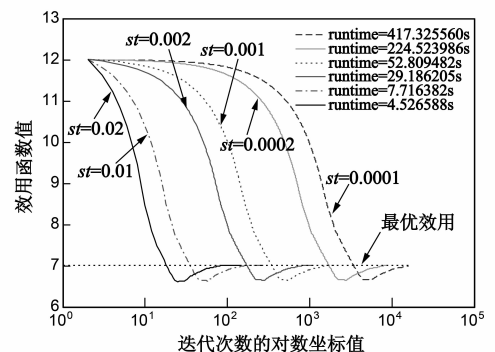


图4 对偶分解算法的收敛特性

图4给出了图2拓扑对应的对偶分解算法收敛图, 图中 st 表示对偶分解时选择的固定步长, 横坐标表示原始变量迭代总次数, 由图可知, 当固定步长变化范围较小时, 对偶次梯度算法的总运行时间和迭代次数变化非常大, 如当 st 取 0.0001 时, 算法总迭代次数为

15826 次,运行时间约为 417s,而步长选择为 0.0002 时,算法总迭代次数为 8531 次,运行时间约为 225s,结果表明对偶次梯度算法对步长选择非常敏感。

为比较分布式牛顿算法与对偶分解算法的收敛特性,图 5 分别给出了拓扑 1 对应的分布式牛顿算法与对偶次梯度算法收敛特性比较图,由图可知,分布式牛顿算法运行时间仅需要 3.6624s,而对偶次梯度算法则需要运行 229.4911s,此外,对偶次梯度算法需要迭代 8400 次才能达到牛顿算法 750 次迭代时的求解精度.这表明分布式牛顿算法收敛速度要明显快于对偶次梯度算法,实际上,不计对偶变量迭代次数,牛顿算法原始变量量只需迭代 56 次即可收敛到最优解,算法收敛速度比对偶算法提高两个数量级。

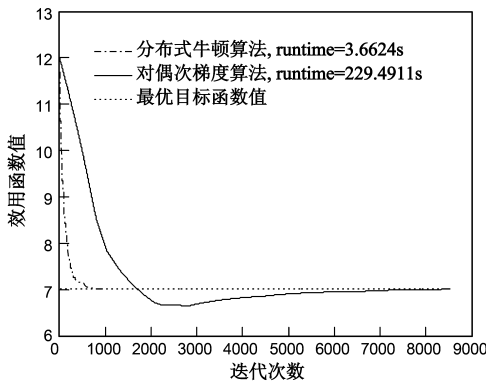


图5 算法收敛比较图

最后,为了充分比较算法的收敛性能,文章还随机产生了 30 个无线传感网络拓扑图,这些网络的平均节点数和会话数分别为 15 和 8,所有的节点均随机分布在 $800\text{m} \times 800\text{m}$ 的方形区域内,两个节点之间的最大通信距离为 300m,其中网络中的节点能量和链路容量初始值及算法变量初始值设置方法同图 2 拓扑中的设置方法.在这 30 个网络拓扑上运行两种算法后,两种算法的收敛速度比较图如图 6 所示。

图 6 中横坐标表示无线传感网络拓扑图的标号,纵

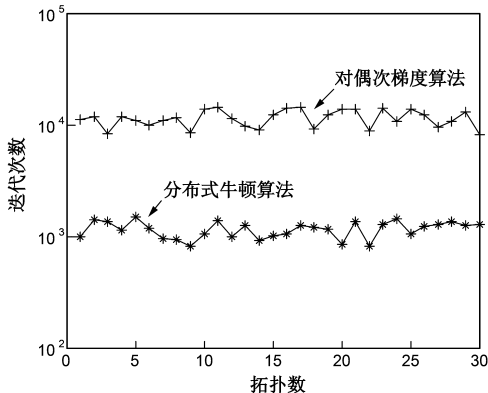


图6 各个拓扑网络中算法迭代次数比较图

坐标表示在这个拓扑上运行两种算法时的迭代次数.针对这 30 个随机产生的网络运行两种算法后,分布式牛顿算法的平均迭代次数为 1156.7 次,对偶次梯度算法的平均迭代次数为 11467.1 次,通过比较进一步说明,分布式牛顿算法收敛速度要明显优于传统一阶收敛速度。

5 总结

文章建立了一种更加符合实际场景的流控制优化模型.为了求得该问题的最优解,本文设计了具有二次收敛性能的分布式牛顿算法.研究表明通过问题等价转化和矩阵分裂,牛顿迭代方向和对偶变量都可以只需要通过有限的局部信息交互迭代计算得到.仿真结果进一步表明,分布式牛顿算法不仅时耗较短,而且收敛速度也远快于传统的分布式求解方法。

参考文献

- [1] Lyengar SS, Brooks RR. Distributed Sensor Networks: Sensor Networking and Applications[M]. Boca Raton, Florida: Chemical Rubber Company Press, 2012.
- [2] Dargie W, Christian P. Fundamentals of Wireless Sensor Networks: Theory and Practice[M]. John Wiley & Sons Press, 2010.
- [3] Li HQ, Guo Q, Tang T. Distributed resource allocation for cognitive radio network with imperfect spectrum sensing[A]. Vehicular Technology Conference, 2013 IEEE 78th[C]. Las Vegas: IEEE, 2013. 1-6.
- [4] 李静叶, 罗涛, 乐光新. 联合网络容量与认知用户满意度的资源分配优化方案[J]. 电子学报, 2012, 40(7): 1315-1322.
Li Jingye, Luo Tao, Le Guangxin. A resource allocation scheme jointly considering network capacity and user satisfaction in cognitive radio networks[J]. Acta Electronica Sinica, 2012, 40(7): 1315-1322. (in Chinese)
- [5] 罗成, 谢维信. 传感器网络拥塞避免与控制的模糊 AQM 算法[J]. 电子学报, 2014, 42(4): 679-684.
Luo Cheng, Xie Weixin. Fuzzy AQM for congestion avoidance and control in sensor networks[J]. Acta Electronica Sinica, 2014, 42(4): 679-684. (in Chinese)
- [6] 张玉鹏, 刘凯, 王广学. 基于无线传感器网络的跨层拥塞控制协议[J]. 电子学报, 2011, 39(10): 2258-2262.
Zhang Yupeng, Liu Kai, Wang Guangxue. Cross-layer congestion control for wireless sensor networks[J]. Acta Electronica Sinica, 2011, 39(10): 2258-2262. (in Chinese)
- [7] 徐伟强, 吴铁军, 等. 强动态 Ad Hoc 网的拥塞控制: 价格协作和滚动优化[J]. 软件学报, 2008, 19(9): 2389-2402.
Xu Weiqiang, Wu Tiejun, et al. Congestion control for high dynamic ad hoc networks: price cooperation and receding opti-

- mization[J]. Journal of Software, 2008, 19(9): 2389 – 2402. (in Chinese)
- [8] Yuen K, B Liang, Li B C. A distributed framework for correlated data gathering in sensor networks[J]. IEEE Transactions on Vehicular Technology, 2008, 57(1): 578 – 593.
- [9] Madan R, Lall S. Distributed algorithms for maximum lifetime routing in wireless sensor networks[J]. IEEE Transaction on Wireless Communications, 2006, 5(8): 2185 – 2193.
- [10] He S B, Chen J M, Yau D K, Sun Y X. Cross-layer optimization of correlated data gathering in wireless sensor networks [J]. IEEE Transactions on Mobile Computing, 2012, 11(11): 1678 – 1691.
- [11] Abdulla A, Nishiyama H, Yang J. Hymn: A novel hybrid multi-hop routing algorithm to improve the longevity of WSNs[J]. IEEE Transactions on Wireless Communications, 2012, 11(7): 2531 – 2541.
- [12] Chen J M, Xu W Q, He S B. Utility-based asynchronous flow control algorithm for wireless sensor networks[J]. IEEE Journal Selected Areas on Communications, 2010, 28(7): 1116 – 1126.
- [13] Bertsekas D P, Gafni E M. Projected newton methods and optimization of multi-commodity flows[J]. IEEE Transactions on Automatic Control, 1983, 28(12): 1090 – 1096.
- [14] Klinecicz J G. A newton method for convex separable network flow problems[J]. Networks, 1983, 13(3): 427 – 442.
- [15] Wei E, Ozdaglar A, Jadbabaie A. A distributed newton method for network utility maximization [A]. IEEE Conference on Decision and Control[C]. Atlanta: IEEE, 2010. 15 – 17.
- [16] Bickson D, Tock Y, Zymnis A. Distributed large scale network utility maximization [A]. IEEE International Symposium on Information Theory [C]. Seoul: IEEE, 2009. 829 – 833.
- [17] Jadbabaie A, Ozdaglar A, Zargham M. A distributed Newton method for network optimization [A]. IEEE Conference on Decision and Control[C]. Shang hai: IEEE, 2009. 16 – 18.
- [18] Cottle R, Pang J, Stone R. The Linear Complementarity Problem[M]. Waltham: Academic Press, 1992.
- [19] Boyd S, Vandenberg L. Convex Optimization[M]. New York: Cambridge University Press, 2004.
- [20] Liu J, Sherali H D. A distributed newton method for joint multi-hop routing and flow control: theory and algorithm [A]. IEEE INFOCOM[C]. Orlando: IEEE, 2012. 2489 – 2497.
- [21] Horn R A, Johnson C R. Matrix Analysis [M]. New York: Cambridge University Press, 1990.
- [22] Grant M, Boyd S, Ye Y. CVX: Matlab software for disciplined convex programming, version 2.0 [EB/OL]. <http://cvxr.com/cvx>, 2013-4-15.

作者简介



穆元彬 男, 1986 年生于山东泰安. 硕士研究生. 研究方向为 无线传感网络、网络分布式优化.

E-mail: muyuanbin@126.com



史清江 (通信作者) 男, 1980 年生于浙江上虞. 副教授. 研究方向为 无线通信系统、网络优化算法设计与分析.

E-mail: shiqj@zstu.edu.cn



徐伟强 男, 1975 年生于浙江杭州. 教授. 研究方向为 通信网络的分布式优化、无线网络资源分配.

E-mail: wqxu@zstu.edu.cn

余晴里 女, 硕士研究生. 研究方向为无线传感器网络.

汪亚明 男, 教授. 研究方向为计算机图像、计算机网络.