entropy

MDPI

Article

Gaussian Belief Propagation for Solving Network Utility Maximization with Delivery Contracts

Shengbin Liao [1,2,*] and Jianyong Sun [3]

[1] National Engineering Center for E-Learning, Huazhong Normal University, Wuhan 430079, China
[2] The National Engineering Laboratory for Educational Big Data Technology, Huazhong Normal University, Wuhan 430079, China
[3] The National Engineering Laboratory for Big Data Analytics and The School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China
* Correspondence: liaoshengbin@mail.ccnu.edu.cn; Tel.: +86-139-7156-8196

Received: 13 June 2019; Accepted: 17 July 2019; Published: 19 July 2019

Abstract: Classical network utility maximization (NUM) models fail to capture network dynamics, which are of increasing importance for modeling network behaviors. In this paper, we consider the NUM with delivery contracts, which are constraints to the classical model to describe network dynamics. This paper investigates a method to distributively solve the given problem. We first transform the problem into an equivalent model of linear equations by dual decomposition theory, and then use Gaussian belief propagation algorithm to solve the equivalent issue distributively. The proposed algorithm has faster convergence speed than the existing first-order methods and distributed Newton method. Experimental results have demonstrated the effectiveness of our proposed approach.

Keywords: network utility maximization; delivery contracts; Gaussian belief propagation; distributed algorithms

1. Introduction

Since the publication of the seminal paper [1] by Kelly et al., the framework of network utility maximization (NUM) has received a great deal of interest in the past two decades, which has been developed into a mathematical theory of network architectures [2]. Many important network design and resource allocation problems can be formulated as a NUM model. The utility concept, originally proposed in economics, is used to measure the satisfaction degree of a consumer for a good or service. In the basic NUM model, the utility of a network user is defined as a function of its data rate. The goal of network system is designed to maximize the overall utility of all the users in the network.

Consider a network with L links and R users, where each link l has a capacity of c_l bps. Let $\mathcal{L}$ be the set of all links and a route r is a non-empty subset of $\mathcal{L}$, let $\mathcal{R}$ be the set of possible routes, and associate a route r with a user r (or a data source r), i.e., $\mathcal{L} = \{1, 2, \ldots, L\}$ and $\mathcal{R} = \{1, 2, \ldots, R\}$. Set $A_{lr} = 1$ if $l \in r$, so that the link l lies on route r, and set $A_{lr} = 0$ otherwise. This defines a 0-1 routing matrix $A = (A_{lr}, l \in \mathcal{L}, r \in \mathcal{R})$.

Suppose that if a rate x_r is allocated to user r then this has utility $U_r(x_r)$ to the user. Assume that the utility $U_r(x_r)$ is increasing, strictly concave, and continuously differentiable over the range $x_r \geq 0$. Let $U = (U_r(x_r), r \in \mathcal{R})$ and $C = (c_l, l \in \mathcal{L})$. Under this model, the network seeks a rate allocation $x = (x_r, r \in \mathcal{R})$ which solves the following optimization problem [3].

$$\begin{array}{ll} \max & \sum\limits_{r\in\mathcal{R}} U_r(x_r) \\ \text{subject to} & Ax \le C \\ & x \ge 0 \end{array} \tag{1}$$

However, the basic NUM model (1) does not consider the network dynamics such as time-varying link capacities and user demands for quality of service (QoS). In this paper, we investigate a dynamic NUM model with QoS constraints, i.e., the model is time-varying over time t, which takes values in the set of time indices $\mathcal{T} = \{=1,2,\ldots,T\}$. The model was first introduced in [4] as following

$$\begin{array}{ll} \max & \sum_{t=1}^{T}\sum_{r=1}^{R} U_r^t(x_r^t) \\ \text{subject to} & A^t x^t \le C^t \ \forall t \in \mathcal{T} \\ & B_r x_r \ge q_r \ \forall r \in \mathcal{R} \\ & x^t \ge 0 \ \forall t \in \mathcal{T} \end{array} \tag{2}$$

where x_r^t denotes the source rate for user r at time step t, $x^t = (x_r^t, r \in \mathcal{R})$ is the rate vector of all users at time step t and $x_r = (x_r^t, t \in \mathcal{T})$ is the source rate allocation for user r. The second constraint in model (2) is the QoS constraints or delivery contracts. For each user, a delivery contract is the required minimal flow to be delivered over some particular time interval. Assume user r has k_r delivery contracts and $q_r \in R^{k_r}$ is the associated contract quantity amounts. A contract is active at time index t if t is in the time interval of the contract. We can define a 0-1 matrix $B_r \in R^{k_r \times T}$ to represent the delivery contract indicator matrix by setting the $(B_r)_{kt} = 1$ if the kth contract of user r is active at time t, and setting $(B_r)_{kt} = 0$ otherwise. Thus, the QoS constraints that all delivery contracts are met can be given by $B_r x_r \ge q_r \ \forall r \in \mathcal{R}$, i.e., the second constraint in model (2).

In the dynamic NUM model (2), utility function U_r^t, route matrix A^t and link capacity C^t are all dependent on the time index t, this means they are all possibly time-varying. There are many different ways to solve the problem (2), such as interior-point methods [5] and primary-dual algorithms [6]. The interior-point methods are efficient for solving the problem (2); however, they are centralized algorithms. The primary-dual algorithms are decentralized, but they suffer from slow convergence speed. In this paper, we concentrate on the issue of designing a distributed algorithm with fast convergence speed for solving the problem (2).

Only a few studies so far have investigated the problem (2) [4,7]. The goal of these works are similar to us. In [4], the authors presented a distributed primary-dual algorithm for solving the problem (2) based on dual decomposition and first-order methods. However, their work suffers from slow convergence speed. Of particular relevance to our work is [7], where a distributed Newton-type algorithm has been developed for solving the dynamic NUM model (2). Unlike these works, we use Gaussian belief propagation (GaBP) algorithm [8] to compute the Newton step and obtain an efficient distributed algorithm for solving the problem (2).

Our proposed solution is a *three-step method* for addressing the given issue. The first step is to obtain the optimality conditions for solving the problem (2) by introducing slack variables. The first step is similar to that approach used in the primary-dual algorithms for solving the problem (2). However, we do not adopt the primary-dual algorithms to solve the problem, which suffer from slow convergence speed.

The second step is to transform the obtained optimality conditions into an inference problem in a probabilistic graph model describing a certain Gaussian distribution with unknown parameters, which equal to optimal solutions for the problem (2). The method used in the second step transfers an algebraic problem into a probabilistic inference problem, which was first raised in [9].

The third step is to use the GaBP algorithm to evaluate distributively the parameter values of the Gaussian distribution. Essentially the GaBP algorithm is used to compute the Newton step in a primary-dual interior-point method [10]. This is similar to the work in [11]. However, this work does not consider the delivery contracts and is the special case of our work.

The outline of this paper is as follows. We first discuss the background and related work in Section 2, then present our method in Section 3. Section 4 provides experimental results and a discussion. Finally, Section 5 concludes this paper.

2. Background and Related Work

Before we present our idea, we first introduce a basic distributed optimization algorithm [3] which solves the model (1). Our work belongs to extensions of their works to dynamic model.

2.1. Basic Primary-Dual Algorithm

Low et al. present in [3] the following basic distributed optimization algorithm for solving the model (1).

The Lagrangian dual function for problem (1) is

$$\begin{aligned} D(\mu) &= \max_{x_r \geq 0} \Big\{ \sum_{r \in \mathcal{R}} U_r(x_r) - \mu^{Tr}(Ax - C) \Big\} \\ &= \max_{x_r \geq 0} \Big\{ \sum_{r \in \mathcal{R}} U_r(x_r) - x_r \sum_{l \in \mathcal{L}} A_{lr}\mu_l \Big\} + \mu^{Tr} C \end{aligned} \tag{3}$$

where $\mu = (\mu_l, l \in \mathcal{L})$ is a vector of Lagrange multipliers and μ^{Tr} denotes the transpose of the vector μ. Here, the second equality follows due to the definition of the matrix A. Thus, the dual problem for primary problem (1) is

$$\min_{\mu \geq 0} \quad D(\mu) \tag{4}$$

In the dual formulation, Lagrange multiplier μ_l can be interpreted as congestion price on link l. A key observation from Equation (3) is that sources can compute their optimal rate individually, based on the total congestion price $\sum_{l \in \mathcal{L}} A_{lr}\mu_l$, using the following source rate algorithm

$$x_r = \arg \max_{x_r \geq 0} \Big\{ \sum_{r \in \mathcal{R}} U_r(x_r) - x_r \sum_{l \in \mathcal{L}} A_{lr}\mu_l \Big\} \tag{5}$$

To solve the dual problem (4), one can use the following projected gradient method

$$\mu_l(t+1) = \Big[\mu_l(t) - \alpha(t)(c_l - \sum_{l \in \mathcal{L}} A_{lr} x_r) \Big]^+ \tag{6}$$

where $\alpha(t)$ is a positive scalar stepsize, and $[a]^+$ denotes the projection of a onto the set R^+ of non-negative real numbers.

According to the duality theory [12] and the assumption that the utility $U_r(x_r)$ is increasing, strictly concave and continuously differentiable over the range $x_r \geq 0$, the optimal solutions to both primary problem (1) and dual problem (4) can be found simultaneously by solving iteratively in Equations (5) and (6), respectively. This suggests treating the network links and the sources as processors in a distributed computation system to solve the primary problem (1) and the dual problem (4). The algorithm (5) and (6) is often referred to as the *basic primary-dual algorithm*. A large number of studies based on NUM framework belong to extensions of the basic primary-dual algorithm, the interested readers along this line please refer to [13].

2.2. Related Work

In the basic NUM model (1), the utility $U_r(x_r)$ of a network user r is defined as function of its data rate x_r, this means that all utility functions are separable. Due to the characteristic of separability of utility functions, a basic distributed NUM algorithm is derived to maximize aggregate user utility by the dual decomposition theory [12]. Along this way, so many extended NUM models and resultant distributed algorithms have been proposed for network architectural design, cross-layer optimization

and resource allocation in wireless as well as wireline networks [14–19]. There are some works which studied the extended NUM models with the non-strictly concave or non-concave utility functions such as in [20–23]. When the utility functions are not strictly concave, the subgradient method is usually used to solve the dual problems instead of the gradient method in the *basic primary-dual algorithm*. If the utility functions are not concave, the extented duality method [24] can be used to construct distributed algorithms.

The design of the utility functions depends on applications of NUM problem. NUM-based approaches have been explored in different applications. Based on NUM, Liu et al. [25] presented a distributed and adaptive solution that jointly computes the data collection rates for each node and finds the schedule transmissions for rechargeable sensor networks. The concept of Water flow Driven Sensor Networks was introduced for leakage and contamination monitoring based on NUM [26]. Sadagopan et al. [27] use NUM approach constructing an energy balance tree in sensor networks, where each sensor node's utility depends on the selection of its parent node. There exist some challenges to define utility functions based on performance metrics of different applications. The relationship between performance metrics and utility functions please refer to [28].

All the above works considered the static NUM models. Dynamic NUM models also belong to the extension of the basic NUM model. In [29], the authors presented a dynamic NUM model in adversarial environments. Their work focuses on the tradeoff between total queue length and utility regret. However, in this paper, we concentrate on the issue of devising a distributed algorithm to solve a dynamic NUM. In [30], the authors proposed a dynamic NUM model with time-varying fading channels. However, the utility functions and route matrix in their work are fixed. Moreover, their work focuses on the convergence behavior and tracking errors of the iterative primary-dual scaled gradient algorithm. Parametric network utility maximization model was presented in [31]. If the parameters are regarded as time steps, their model is equivalent to ours. However, their work concentrates on the tracking of algorithm trajectory by using a pathfollowing method on the parametric optimization problem [32].

The works in [4,7] are particular relevance to this work. The dynamic NUM with delivery contracts was first proposed in [4], and the authors presented a distributed primary-dual algorithm to solve the problem. The distributed primary-dual algorithm provided in [4] is based on dual decomposition theory, which is similar to the *basic primary-dual algorithm*. These primary-dual algorithms usually suffer from the slow rate of convergence [33,34].

The work in [7] also investigates the same model with ours in this paper. They proposed a distributed Newton method for solving the given problem. Their distributed algorithm obtained fast convergence speed compared with the distributed primary-dual algorithms. the method in [7] approximates the Newton direction at each iteration by using the matrix splitting technique. However, our method in this paper uses GaBP algorithm to evaluate the Newton direction.

Our proposed algorithm is a kind of primary-dual interior-point method. The primary-dual interior-point method was used for solving the NUM problem in [35]; however, the proposed algorithm is not decentralized. The work in [11] provided a distributed algorithm for solving a NUM by using the GaBP algorithm to compute the Newton direction. This is similar to our work. However, their model is static, and our model includes delivery contracts. Their work can be regarded as a special case of our work.

3. Our Method

In this section, we develop a *three-step method* to solve distributively the dynamic NUM problem (2). Before we present our idea, we first introduce some notations to simplify the model (2). let $x = (x_1^{Tr}, x_2^{Tr}, \ldots, x_R^{Tr})^{Tr}$ and $C = (C_1^{Tr}, C_2^{Tr}, \ldots, C_T^{Tr})^{Tr}$ be the rate vector of users and the capacity vector of links respectively, where a^{Tr} is the transpose of the vector a. let matrix A denote the corresponding routing matrix for all time steps given as

$$A = \begin{pmatrix} A^1 & 0 & \dots & 0 \\ 0 & A^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & A^T \end{pmatrix}$$

Then we can write down the first constraint of the dynamic NUM problem (2) as $Ax \leq C$.

Similarly, let $q = (q_1^{Tr}, q_2^{Tr}, \dots, q_R^{Tr})^{Tr}$ be the contract quantity vector of users, and the matrix B denote the delivery contract matrix for all users given as

$$B = \begin{pmatrix} B_1 & 0 & \dots & 0 \\ 0 & B_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & B_R \end{pmatrix}$$

Thus, the second constraint of the dynamic NUM problem (2) can be written as $Bx \geq q$. Let X be the rate matrix with entries $(x_r^t, r \in \mathcal{R}, t \in \mathcal{T})$, define $U(X) = \sum_{t=1}^{T} \sum_{r=1}^{R} U_r^t(x_r^t)$

Equivalently, we can transform the dynamic NUM problem (2) as following,

$$\begin{array}{ll} \max & U(X) \\ \text{subject to} & Ax \leq C \\ & Bx \geq q \\ & x \geq 0 \end{array} \tag{7}$$

Next, we will present our method to solve the problem (7).

3.1. Optimality Conditions

The Lagrangian associated with the NUM problem (7) is

$$L(x_r^t; \lambda, \mu, \alpha) = U(X) - \lambda^{Tr}(Ax - C) + \mu^{Tr}(Bx - q) + \alpha^{Tr}x \tag{8}$$

where λ, μ and α are Lagrange multiplier vectors which are associated the inequality constraints in the NUM problem (7). Therefore, the dual function is given by

$$D(\lambda, \mu, \alpha) = \max_{x_r \geq 0} L(x_r^t; \lambda, \mu, \alpha)$$

Thus, the dual problem for model (7) is given by

$$\min_{\lambda \geq 0, \mu \geq 0, \alpha \geq 0} D(\lambda, \mu, \alpha) \tag{9}$$

Assume $\hat{x}_r^t$, $(\hat{\lambda}, \hat{\mu}, \hat{\alpha})$ are the optimal solutions of the primary problem (7) and dual problem (9), according to the Karush-Kuhn-Tucker (KKT) conditions [12], we can obtain the optimality conditions as following,

$$\begin{array}{l} -\nabla U(\hat{X}) + A^{Tr}\hat{\lambda} - B^{Tr}\hat{\mu} - \hat{\alpha} = 0 \\ \mathbf{diag}(\hat{\lambda})(C - A\hat{x}) = 0 \\ \mathbf{diag}(\hat{\mu})(B\hat{x} - q)) = 0 \\ \mathbf{diag}(\hat{\alpha})\hat{x} = 0 \end{array} \tag{10}$$

where $\mathbf{diag}(\cdot)$ denotes a diagonal matrix formed from its vector argument.

3.2. Inference Problem

We can modify the optimality conditions (10) and apply the primary-dual interior-point method on the modified optimality conditions for solving the primary problem (7) and dual problem (9)

in an iterative manner with the given error of the duality gap [12]. The modification is parametrized by a parameter k as [35],

$$\begin{array}{l} -\nabla U(X) + A^{Tr}\lambda - B^{Tr}\mu - \alpha = 0 \\ \mathbf{diag}(\lambda)(C - Ax) = (\frac{1}{k})\mathbf{1} \\ \mathbf{diag}(\mu)(Bx - q)) = (\frac{1}{k})\mathbf{1} \\ \mathbf{diag}(\alpha)x = (\frac{1}{k})\mathbf{1} \end{array} \tag{11}$$

where $k \geq 0$ is a parameter. We know from (11) that the modified optimality conditions approximate the optimality conditions as $k \to \infty$ and different values of k set the different accuracies of the approximation. We can compactly write the modified optimality conditions as following,

$$r_t(x, \lambda, \mu, \alpha) = \begin{bmatrix} -\nabla U(X) + A^{Tr}\lambda - B^{Tr}\mu - \alpha \\ \mathbf{diag}(\lambda)(C - Ax) - (\frac{1}{k})\mathbf{1} \\ \mathbf{diag}(\mu)(Bx - q)) - (\frac{1}{k})\mathbf{1} \\ \mathbf{diag}(\alpha)x - (\frac{1}{k})\mathbf{1} \end{bmatrix} = 0$$

The search direction of the primary-dual interior-point method is the Newton step for solving the modified optimality conditions $r_t(x, \lambda, \mu, \alpha) = 0$. If $y = (x, \lambda, \mu, \alpha)^{Tr}$ is the current point, the Newton step $\Delta y = (\Delta x, \Delta\lambda, \Delta\mu, \Delta\alpha)^{Tr}$, then we have,

$$r(y + \Delta y) \approx r_t(y) + r_t^{'}(y)\Delta y = 0,$$

where $r_t^{'}(y)$ denotes the derivative of $r_t(y)$. The above equation means

$$-r_t(x, \lambda, \mu, \alpha) = \begin{bmatrix} -\nabla^2 U(X) & A^{Tr} & -B^{Tr} & -\mathbf{I} \\ -\mathbf{diag}(\lambda)A & \mathbf{diag}(C - Ax) & 0 & 0 \\ \mathbf{diag}(\mu)B & 0 & \mathbf{diag}(Bx - q)) & 0 \\ \mathbf{diag}(\alpha) & 0 & 0 & \mathbf{diag}(x) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta\mu \\ \Delta\alpha \end{bmatrix} \tag{12}$$

Searching the Newton step by Equation (12) is the main computational bottleneck in the primary-dual interior-point method. However, in this paper, we do not directly calculate Newton's direction by Equation (12). We transform the problem solving the liner Equation (12) into a probabilistic inference which can be computed based on GaBP. We first transfer the matrix in the right side of Equation (12) into a symmetric matrix by multiplying $r_t(x, \lambda, \mu, \alpha)$ a factor $(1, -1/\lambda, -1/\mu, -1/\alpha)$ as following,

$$\begin{aligned} -\hat{r}_t(x, \lambda, \mu, \alpha) &= \begin{bmatrix} -\nabla^2 U(X) & A^{Tr} & -B^{Tr} & -\mathbf{I} \\ A & -\mathbf{diag}(C - Ax)/\lambda & 0 & 0 \\ -B & 0 & -\mathbf{diag}(Bx - q)/\mu & 0 \\ -\mathbf{I} & 0 & 0 & -\mathbf{diag}(x)/\alpha \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta\mu \\ \Delta\alpha \end{bmatrix} \\ &= \mathcal{A} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta\mu \\ \Delta\alpha \end{bmatrix} \end{aligned} \tag{13}$$

where $\hat{r}_t(x, \lambda, \mu, \alpha)$ and $\mathcal{A}$ are defined as,

$$\hat{r}_t(x, \lambda, \mu, \alpha) = r_t(x, \lambda, \mu, \alpha) \cdot (1, -1/\lambda, -1/\mu, -1/\alpha)^{Tr} = \begin{bmatrix} -\nabla U(X) + A^{Tr}\lambda - B^{Tr}\mu - \alpha \\ -(C - Ax) + (\frac{\lambda}{k})\mathbf{1} \\ -(Bx - q)) + (\frac{\mu}{k})\mathbf{1} \\ -x + (\frac{\alpha}{k})\mathbf{1} \end{bmatrix}$$

and

$$\mathcal{A} = \begin{bmatrix} -\nabla^2 U(X) & A^{Tr} & -B^{Tr} & -\mathbf{I} \\ A & -\mathbf{diag}(C - Ax)/\lambda & 0 & 0 \\ -B & 0 & -\mathbf{diag}(Bx - q)/\mu & 0 \\ -\mathbf{I} & 0 & 0 & -\mathbf{diag}(x)/\alpha \end{bmatrix}$$

For notational simplicity, let $b = -\hat{r}_t(x, \lambda, \mu, \alpha)$, $w = (\Delta x, \Delta\lambda, \Delta\mu, \Delta\alpha)^{Tr}$, we can write the Equation (13) as,

$$\mathcal{A}w = b \tag{14}$$

Therefore, the Equation (14) for computing the Newton step is a system of linear equations with a symmetric coefficient matrix, which can be efficiently solved by using the GaBP algorithm [9]. We can define an undirected probabilistic graphical model $\mathcal{G} = (\mathcal{W}, \mathcal{E})$, where $\mathcal{W}$ is a set of nodes which consist of the variables of linear Equation (14), and $\mathcal{E}$ is a set of edges which are determined by the non-zero entries of the coefficient matrix $\mathcal{A}$. Given the matrix $\mathcal{A}$ and vector b, we can build up a Gaussian density function $p(w) \sim \exp(-\frac{1}{2}w^{Tr}\mathcal{A}w + b^{Tr}w)$, which corresponds to the probabilistic graph $\mathcal{G}$. Let $M = TR + TL + \sum_{r=1}^{R} Tk_r + TR$ be the dimension of the vector b (or vector w) (refer to the original model (2), we know that the number of the objective functions is TR, the capacity constraints are TL, the constraints of the delivery contracts are $\sum_{r=1}^{R} Tk_r$, and the non-negativity constraints are TR. Therefore, the dimension of the vector b is $TR + TL + \sum_{r=1}^{R} Tk_r + TR$.). The probabilistic graph $\mathcal{G}$ has edge potentials (or compatibility functions) ψ and self-potentials (or evidence) ϕ. These graph potentials are determined by the following pairwise factorization of Gaussian distribution,

$$p(w) \propto \prod_{i=1}^{M} \phi_i(w_i) \prod_{\{i,j\}} \psi_{ij}(w_i, w_j), \tag{15}$$

resulting in $\phi_i(w_i) \doteq \exp(-\frac{1}{2}\mathcal{A}_{ii}w_i^2 + b_i w_i)$ and $\psi_{ij}(w_i, w_j) \doteq \exp(-\frac{1}{2}w_i\mathcal{A}_{ij}w_j)$. Using this probabilistic graph, we can transform the problem of solving the linear Equation (14) from the algebraic domain to a parameter estimation problem in the domain of probabilistic inference, as stated in the following theorem [9].

Theorem 1. *The computation of the solution vector $w^* = \mathcal{A}^{-1}b$ is identical to the inference of the vector marginal means $\theta \doteq \{\theta_1, \ldots, \theta_R\}$ over the graph $\mathcal{G}$ with the associated joint Gaussian probability density function $p(w) \sim \mathcal{N}(\theta, \mathcal{A}^{-1})$.*

Proof. See Appendix A. □

The above theory shows that if we can distributively evaluate the mean of the Gaussian distribution $p(w)$, then we can use the primary-dual interior-point method to distributively solve the primary problem (7) and dual problem (9). The next section will present the method for solving the mean of the inference problem (15) based on GaBP algorithm.

3.3. Parameter Evaluation Based on GaBP

Belief propagation is a kind of local message-passing algorithm and has been found to have excellent performance in many applications [36]. GaBP is a special case of the belief propagation algorithm, in which the underlying distributions are Gaussian. According to the statements in above section, in order to solve the linear equation problem (14) we need to infer the marginal densities $p(w_i)$, which must also be Gaussian, i.e.,

$$p(w_i) \sim \mathcal{N}(\theta_i = \{\mathcal{A}^{-1}b\}_i,\ P_i^{-1} = \{\mathcal{A}^{-1}\}_{ii})$$

where θ_i and P_i are the marginal mean and inverse variance (also known as the precision), respectively. Let $N(i)$ be the set of all the nodes neighboring the node i (excluding node i). The set $N(i) \setminus j$ includes all the nodes in the set of $N(i)$ except node j. The following Algorithm 1 provides the GaBP algorithm update rules for inferring the mean θ_i.

Algorithm 1: GaBP Algorithm

- Step 0 Initialization:
  Set a convergence threshold ϵ, $P_{ki} = 0$ and $\theta_{ki} = 0, \forall k \in N(i)$. Compute $P_{ii} = \mathcal{A}_{ii}$ and $\theta_{ii} = b_i/\mathcal{A}_{ii}$.
- Step 1 Iteration:
  Propagate the messages P_{ki} and $\theta_{ki}, \forall k \in N(i)$. Compute $P_{ij} = -\mathcal{A}_{ij}^2/(P_{ii} + \sum_{k \in N(i)\setminus j} P_{ki})$, $\theta_{ij} = (P_{ii}\theta_{ii} + \sum_{k \in N(i)\setminus j} P_{ki}\theta_{ki})/\mathcal{A}_{ij}$.
- Step 2 Convergence check:
  If the message P_{ij} and θ_{ij} do not converge, return to Step 1, else, go to Step 3.
- Step 3 Inference:
  Compute the marginal means $\theta_i = (P_{ii}\theta_{ii} + \sum_{k \in N(i)} P_{ki}\theta_{ki})/((P_{ii} + \sum_{k \in N(i)} P_{ki}))$.
- Step 4 Output:
  Output the solution $w_i = \theta_i$.

4. Experiments and Analysis

4.1. Experimental Settings

In this section, we justify empirically the effectiveness of the proposed algorithm and compare the performance with the classical primary-dual algorithm and the primary-dual interior-point method. The classical primary-dual algorithm is based on dual decomposition. The primary-dual interior-point method used here is an iterative method for solving approximately a Newton system, which is usually called a truncated Newton primary-dual interior-point method [37].

We consider a network which has 100 flows and 200 links, and all of the utility functions are set to logarithmic functions, i.e., $U_r^t(x_r^t) = log(x_r^t)$, which are the most widely used in NUM problems [2]. The network was randomly generated and similar to that used in [34,35], this means that we need generate the link capacities and the routing matrix. The link capacities are chosen independently from a uniform distribution on [0.1, 1] and all of the required minimal flows to be delivered over different time intervals are set to 0.1, and the elements of the routing matrix A are generated randomly and independently, so that the average route length is 6 links. The time index T and all stepsizes are set to 10 and 0.001, respectively. After the network was generated, our proposed algorithm, primary-dual algorithm, and truncated Newton algorithm are performed once on it, respectively. The experimental results and comparisons are provided in the next section.

4.2. Experimental Results

We first evaluate the convergence of the proposed algorithm and compare with the classical primary-dual algorithm, these two algorithms are all distributed. Figure 1 shows the convergence curves of total utilities and Figure 2 provides the duality gap or corresponding residual values between the primary function and dual function.

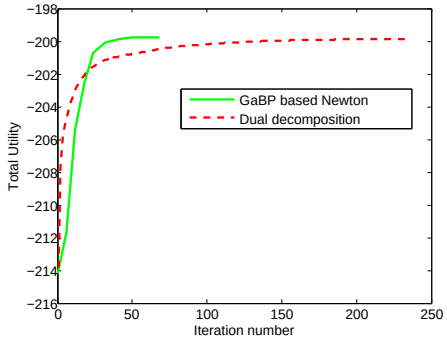

Figure 1. The convergence comparison of the proposed method and the dual decomposition algorithm. The green curve denotes the convergence speed of total utility for our proposed algorithm which is a distributed Newton method based on GaBP, and the red curve denotes the convergence speed of total utility for the dual decomposition algorithm which is a distributed primary-dual algorithm.

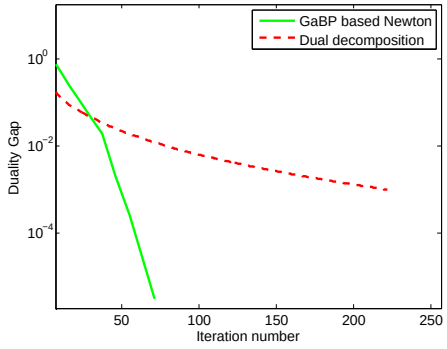

Figure 2. The duality gap comparison of the proposed method and the dual decomposition algorithm. The green curve denotes the estimation errors between the primary function and dual function versus iteration number for our proposed method, and the red curve denotes the estimation errors between the primary function and dual function versus iteration number for the dual decomposition algorithm.

We also compare the performance of our proposed method with the truncated Newton method which has achieved a very fast convergence speed and very good accuracy for solving nonlinear equation system. However, the truncated Newton method is centralized. Figures 3 and 4 show the convergence curves of total utilities and the duality gap curves versus iteration number for our proposed method and the truncated Newton method, respectively.

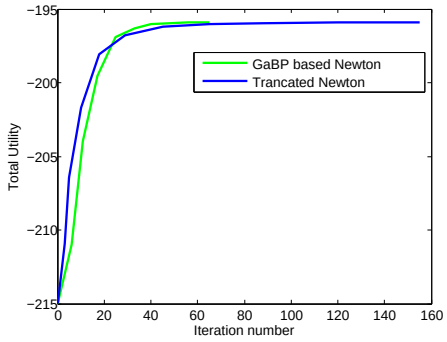

Figure 3. The convergence comparison of the proposed method and the truncated Newton method. The green curve denotes the convergence speed of total utility for our proposed algorithm which is a distributed Newton method based on GaBP, and the blue curve denotes the convergence speed of total utility for the truncated Newton method which is a centralized algorithm.

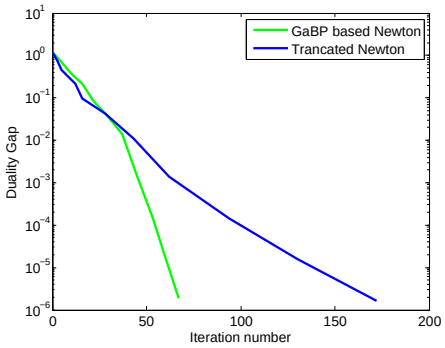

Figure 4. The duality gap comparison of the proposed method and the truncated Newton method. The green curve denotes the estimation errors between the primary function and dual function versus iteration number for our proposed method, and the blue curve denotes the estimation errors between the primary function and dual function versus iteration number for the truncated Newton algorithm.

Our proposed method uses GaBP algorithm to compute the Newton and the truncated Newton method adopts the preconditioned conjugate gradient (PCG) algorithm [38] for computing the Newton step. While the above Figures 3 and 4 provide the performance comparison in term of the Newton steps, we also give the comparison of the iteration count in each Newton step for these two algorithms in Tables 1 and 2. Tables 1 and 2 are the experimental results for two networks which have 100 flows and 200 links, and 500 flows and 1000 links, respectively.

Table 1. Experimental results of the iteration count for each Newton step in a small network.

Newton Step Number	GaBP	PCG
1	6	3
2	6	2
3	6	2
4	7	5
5	9	9
6	10	12
7	12	13
8	15	22
9	14	29
10	13	34
11		43
total	98	174

Table 2. Experimental results of the iteration count for each Newton step in a bigger network.

Newton Step Number	GaBP	PCG
1	6	2
2	5	4
3	5	8
4	6	3
5	7	16
6	7	20
7	7	36
8	7	64
9		101
total	50	253

4.3. Experiment Analysis

4.3.1. Analysis and Comparison with a Distributed Method

We will analyze the experimental results in terms of convergence speed and solution accuracy. From Figure 1, we can see that our proposed algorithm and the dual decomposition algorithm can converge to the optimal value Within a certain range of errors. However, the convergence speed of our method is much faster than the dual decomposition algorithm.

The duality gap between the primary function and the dual function depicts the accuracy of the obtained solution. For a convex optimization problem, the primary variables and dual variables will eventually approach the optimal solution as the duality gap tends to zero. Naturally, we expect that our proposed algorithm will obtain a smaller duality gap. From Figure 2, we can see that the duality gap achieved by our proposed algorithm is smaller than that obtained by the dual decomposition method.

4.3.2. Analysis and Comparison with a Centralized Approach

The truncated Newton method is a centralized approach, which is an efficient primary-dual interior-point method and achieves good performance in many optimization problems [39]. We gave the performance comparison for our proposed method and the truncated Newton method in this section.

From Figure 3, we can see that the convergence speed of our proposed method is very fast, which is slightly faster than the truncated Newton method. This means that both methods had comparable convergence speed. However, as specified before, our proposed approach is distributed, while the truncated Newton method is centralized.

Figure 4 provides the accuracy comparison of the solutions obtained by both methods. From Figure 4, we can see that our proposed method has a smaller duality gap than that achieved by the truncated Newton method. This means that although both methods had comparable convergence rate, the accuracy of the solution obtained by our method is better than that achieved by the truncated Newton method.

As these two methods computed the Newton steps based on GaBP and PCG algorithms respectively, we also compared the iteration count for each Newton step. From Tables 1 and 2, we can see that the iteration count of GaBP algorithm is smaller than that required by the PCG algorithm except for the first few Newton steps. Moreover, the total iteration count of GaBP algorithm is also smaller than that achieved by the PCG algorithm.

Another advantage for GaBP algorithm is that as the Newton step number increases, the iteration count for each Newton step tends to a stable value. However, the iteration count required by PCG algorithm always increases; moreover, the increased magnitudes grows bigger as the scale of the network grows.

5. Conclusions

We propose a three-step method for distributively solving network utility maximization with delivery contracts (or dynamic NUM). This paper first obtained the optimality conditions for solving the dynamic NUM problem by dual decomposition theory. Then we transform the problem for searching the Newton step in solving the optimality conditions into a probabilistic inference. Finally, GaBP algorithm was used to compute the probabilistic inference.

NUM problems are usually solved by means of the classical primary-dual algorithm, which is used as the benchmark algorithm for testing the effectiveness of our proposed method. By comparing and analyzing the experimental results, we can reach a conclusion that the proposed method is effective in convergence speed and solution accuracy compared with the classical primary-dual algorithm.

Our proposed method belongs to distributed primary-dual interior-point methods. Therefore, we also compared the performance of our proposed method with the primary-dual interior-point method based on PCG, which had achieved a very fast convergence speed and very good accuracy

for solving nonlinear equation problems. The experimental results also validated the effectiveness of our proposed method.

Author Contributions: S.L. conceived and designed the experiments; J.S. performed the experiments; S.L. and J.S. wrote the paper.

Funding: This research was funded by the Fundamental Research Funds for the Central Universities(grant number: CCNU19TS021), and the National Key Technology Research Program of the Ministry of Science and Technology of China (grant number: 2015BAK33B02), and National Natural Science Foundation of China (grant number: 61671483, 61072051).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

NUM	Network utility maximization
GaBP	Gaussian belief propagation
QoS	Quality of service
KKT	Karush-Kuhn-Tucker
PCG	preconditioned conjugate gradient
$(\cdot)^{Tr}$	The transpose of a matrix or a vector

Appendix A

Proof of Theorem 1. Solving the linear equation system $\mathcal{A}w = b$ is equivalent to maximizing the quadratic form $-\frac{1}{2}w^{Tr}\mathcal{A}w + b^{Tr}w$, which is further equivalent to finding the maximal value of the exponential function $\exp(-\frac{1}{2}w^{Tr}\mathcal{A}w + b^{Tr}w)$.

Next, we can define the joint Gaussian probability density function as follows:

$$p(w) \doteq \mathcal{Z}^{-1}\exp(-\frac{1}{2}w^{Tr}\mathcal{A}w + b^{Tr}w) \tag{A1}$$

where $\mathcal{Z}$ is a normalization factor to make $p(w)$ to be a probability distribution. Let $\theta \doteq \mathcal{A}^{-1}b$, we can rewrite the joint Gaussian probability density function as following,

$$\begin{aligned} p(w) = \mathcal{Z}^{-1}\exp(\frac{1}{2}\theta^{Tr}\mathcal{A}\theta)\exp(-\frac{1}{2}w^{Tr}\mathcal{A}w + \theta^{Tr}\mathcal{A}w - \frac{1}{2}\theta^{Tr}\mathcal{A}\theta) \\ = \xi\exp(-\frac{1}{2}(w-\theta)^{Tr}\mathcal{A}(w-\theta)) \\ = \mathcal{N}(\theta, \mathcal{A}^{-1}) \end{aligned} \tag{A2}$$

where $\xi = \mathcal{Z}^{-1}\exp(\frac{1}{2}\theta^{Tr}\mathcal{A}\theta)$ is the new normalization factor after the distribution $p(w)$ has been rewritten.

From the above Equation (A2), we see that the mean vector of the Gaussian distribution $p(w)$ defined by Equation (A1) is θ, which is equal to the our target solution $w^* = \mathcal{A}^{-1}b$. This proves Theorem 1. □

References

1. Kelly, F.P.; Maulloo, A.; Tan, D. Rate control for communication networks: Shadow prices, proportional fairness and stability. *J. Oper. Res. Soc. Am.* **1998**, *49*, 237–252. [CrossRef]
2. Chiang, M.; Low, S.H.; Calderbank, A.R.; Doyle, J.C. Layering as optimization decomposition: A mathematical theory of network architectures. *Proc. IEEE* **2007**, *95*, 255–312. [CrossRef]
3. Low, S.H.; Lapsley, D.E. Optimal flow control, I: Basic algorithm and convergence. *IEEE/ACM Trans. Netw.* **1999**, *7*, 861–874. [CrossRef]

4. Trichakis, N.; Zymnis, A.; Boyd, S. Dynamic Network Utility Maximization with delivery contracts. In Proceedings of the IFAC World Congress, Seoul, Korea, 6–11 July 2008; pp. 2907–2912.
5. Nocedal, J.; Wright, S.J. *Numerical Optimization*; Springer: New York, NY, USA, 1999.
6. Bertsekas, D.P. *Nonlinear Programming*, 2nd ed.; Athena Scientific: Nashua, NH, USA, 1999.
7. Weiy, E.; Ozdaglary, A.; Eryilmazz, A.; Jadbabaie, A. A Distributed Newton Method for Dynamic Network Utility Maximization with Delivery Contracts. In Proceedings of the 46th Annual Conference on Information Sciences and Systems (CISS 2012), Princeton, NJ, USA, 21–23 March 2012; pp. 1–6.
8. Weiss, Y.; Freeman, W.T. Correctness of Belief Propagation in Gaussian Graphical Models of Arbitrary Topology. *Neural Comput.* **2001**, *13* , 2173–2200. [CrossRef] [PubMed]
9. Danny, B. Gaussian Belief Propagation: Theory and Application. Ph.D. Thesis, The Hebrew University of Jerusalem, Jerusalem, Israel, 2009.
10. Wright, S.J. *Primal-Dual Interior-Point Methods*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 1997.
11. Bickson, D.; Tock, Y.; Zymnis, A.; Boyd, S.P.; Dolev, D. Distributed large scale network utility maximization. In Proceedings of the IEEE International Symposium on Information Theory (ISIT 2009), Seoul, Korea, 28 June–3 July 2009; pp. 829–833.
12. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.
13. Pham, Q.V.; Hwang, W.J. Network utility maximization-based congestion control over Wireless Networks: A survey and potential directives. *IEEE Commun. Surv. Tutor.* **2017**, *9* , 1173–1200. [CrossRef]
14. Shengbin, L.; Jianhua, H. Design and analysis of distributed utility maximization algorithm for multihop wireless network with inaccurate feedback. *Int. J. Commun. Syst.* **2014**, *27*, 4280–4299.
15. Jan, V. Dynamic Scoring: Probabilistic Model Selection Based on Utility Maximization. *Entropy* **2019**, *21*, 36. [CrossRef]
16. Im, Y.; Joe-Wong, C.; Ha, S.; Sen, S.; Kwon, T.; Chiang M. AMUSE: Empowering users for cost-aware offloading with throughput-delay tradeoff. *IEEE Trans. Mob. Comput.* **2016**, *15*, 1062–1076. [CrossRef]
17. Merayo, N.; Pavon-Marino, P.; Aguado, J.C.; Duran, R.J.; Burrull, F.; Bueno-Delgaado, V. Fair bandwidth allocation algorithm for PONS based on network utility maximization. *J. Opt. Commun. Netw.* **2017**, *9*, 75–86. [CrossRef]
18. Abhishek, S.; Eytan, M. Network utility maximization with heterogeneous traffic flows. In Proceedings of the 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt 2018), Shanghai, China, 7–11 May 2018; pp. 1–10.
19. Shengbin, L.; Qingfu, Z. A multiutility framework with application for studying tradeoff between utility and lifetime in wireless sensor networks. *IEEE Trans. Veh. Technol.* **2011**, *64*, 4701–4711.
20. Wang, W.; Palaniswami, M.; Low S.H. Optimal flow control and routing in multi-path net-works. *Perform. Eval.* **2003**, *52*, 119–132. [CrossRef]
21. Lin, X.; Shroff, N.B. Utility maximization for communication networks withmultipath routing. *IEEE Trans. Autom. Contr.* **2006**, *51*, 766–781. [CrossRef]
22. Pal, A.; Kant, K. On the Feasibility of Distributed Sampling Rate Adaptation in Heterogeneous and Collaborative Wireless Sensor Networks. In Proceedings of the 25th International Conference on Computer Communication and Networks (ICCCN 2016), Waikoloa, HI, USA, 1–4 August 2016; pp. 1–9.
23. Liao, S.; Sun, J.; Chen, Y.; Wang, Y.; Zhang, P. Distributed power control for wireless networks via the alternating direction method of multipliers. *J. Netw. Comput. Appl.* **2015**, *55*, 81–88. [CrossRef]
24. Chen, Y.; Chen, M. Extended duality for nonlinear programming. *Comput. Optim. Appl.* **2010**, *47*, 33–59. [CrossRef]
25. Liu, R.; Sinha, P.; Koksal, C.E. Joint energy management and resource allocation inrechargeable sensor networks. In Proceedings of the 29th Conference on Computer Communications (INFOCOM 2010), San Diego, CA, USA, 15–19 March 2010; pp. 902–910.
26. Pal, A.; Kant, K. Water flow driven sensor networks for leakage and contamination monitoring. In Proceedings of the IEEE 16th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM 2015), Boston, MA, USA, 14–17 June 2015; pp. 1–9.
27. Sadagopan, N.; Singh, M.; Krishnamachari, B. Decentralized utility-based sensor network design. *Mob. Netw. Appl.* **2006**, *11*, 341–350. [CrossRef]

28. Zhao, Y.; Mao, S.; Neel, J.; Reed, J. Performance evaluation of cognitive radios: Metrics, utility functions, and methodology. *Proc. IEEE* **2009**, *97*, 642–659. [CrossRef]
29. Qingkai, L.; Eytan, M. Network Utility Maximization in Adversarial Environments. In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM 2018), Honolulu, HI, USA, 15–19 April 2018; pp. 1–10.
30. Junting, C.; Vincent, K.N.L.; Yong, C. Distributive network utility maximization over time-varying fading channels. *IEEE Trans. Signal Process.* **2011**, *59*, 2395–2404.
31. Lutbat, Y.; Enkhbat, R.; Suk-Hwan, L.; Won-Joo, H. Parametric network utility maximization problem. *Optim. Lett.* **2014**, *8*, 889–901.
32. Guddat, J.; Guerra, V.F. *Parametric Optimization: Singularities, Pathfollowing and Jumps*; Wiley: New York, NY, USA, 1990.
33. Ehsan, N.; Tansu, A.; Girish, N.N.; Robin, J.E. Convergence analysis of quantized primal-dual algorithms in network utility maximization problems. *IEEE Trans. Control Netw. Syst.* **2018**, *5*, 284–297.
34. Michael, Z.; Alejandro, R.; Asuman, O.; Ali, J. Accelerated dual descent for network flow optimization. *IEEE Trans. Autom. Control* **2014**, *59*, 905–920.
35. Zymnis, A.; Trichakis, N.; Boyd, S.; O'Neill, D. An interior-point method for large scale network utility maximization. In Proceedings of the Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, 26–28 September 2007; pp. 877–882.
36. Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*; Morgan Kaufmann: San Francisco, CA, USA, 1988.
37. Kelley, C.T. *Iterative Methods for Linear and Nonlinear Equations*; Society for Industrial and Applied Mathematics (SIAM): Philadelphia, PA, USA, 1995; ISBN 9780898713527.
38. Avriel, M. *Nonlinear Programming: Analysis and Methods*; Dover Publishing: Mineola, NY, USA, 2003; ISBN 0-486-43227-0.
39. Bonnans, J.F.; Gilbert, J.C.; Lemaréchal, C.; Sagastizábal, C.A. *Numerical Optimization: Theoretical and Practical Aspects*; Springer: Berlin, Germany, 2006; ISBN 3-540-35445-X.

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).