The Multi-Path Utility Maximization Problem [*]

Xiaojun Lin and Ness B. Shroff
School of Electrical and Computer Engineering
Purdue University, West Lafayette, IN 47906
{linx,shroff}@ecn.purdue.edu

Abstract

In this paper, we study solutions to the multi-path utility maximization problem of the type in (1). Optimizations of this type arise naturally in several networking problems such as the multi-path flow control problem. When the capacity of the network is large, it can also model the optimal QoS routing problem and the optimal pricing problem. We present a distributed solution to this problem that can be implemented online. The convergence of the algorithm is established.

1 Introduction

In this paper we are concerned with the problem of the following form. Let $\vec{x}_i = [x_{ij}, j = 1, ...J(i)]$ be a vector in a convex set $C_i = \{x_{ij} \geq 0 \text{ for all } j \text{ and } \sum_{j=1}^{J(i)} x_{ij} \in D_i\}$ for some convex set $D_i \subset \mathbf{R}$. Let f_i be a concave function. Let $\vec{x} = [\vec{x}_i, i = 1, ..., I]$ and $C = \bigotimes_{i=1}^{I} C_i$. The problem is:

$$\max_{\vec{x} \in C} \quad \sum_{i=1}^{I} f_i(\sum_{j=1}^{J(i)} x_{ij}) \tag{1}$$
$$\text{subject to} \quad \sum_{i=1}^{I} \sum_{j=1}^{J(i)} E_{ij}^l x_{ij} \leq R^l \quad \text{for all } l = 1, ..., L,$$

where $R^l > 0$ and $E_{ij}^l \geq 0$. We will refer to this problem as the *multi-path utility maximization problem.* We are interested in distributed solutions to this problem that is suitable for *online* implementation.

Optimizations of this form appear in several networking problems. A few examples are given below:

Example 1: A canonical example is the *multi-path flow control problem.* The network has L links. The capacity of each link l is R^l. There are I users. Each user i has $J(i)$ alternate paths through the network. Let $H_{ij}^l = 1$ if the path j of user i uses link l, $H_{ij}^l = 0$ otherwise. Let s_{ij} be the rate with which user i sends data on path j. The total rate sent by user i is then $\sum_{j=1}^{J(i)} s_{ij}$. Let $U_i(\sum_{j=1}^{J(i)} s_{ij})$ be the utility to the user i, where U_i is

[*]This work has been partially supported by the NSF grant ANI-0099137, the Indiana 21st Century Fund 651-1285-074 and a grant through the Purdue Research Foundation.

a concave utility function. The flow control problem can be formulated as the following optimization problem [5, 14], which is essentially the same form in (1):

$$\begin{aligned} \max_{[s_{ij}]\geq 0} \quad & \sum_{i=1}^{I} U_i(\sum_{j=1}^{J(i)} s_{ij}) \\ \text{subject to} \quad & \sum_{i=1}^{I}\sum_{j=1}^{J(i)} H_{ij}^l s_{ij} \leq R^l \quad \text{for all } l = 1, ..., L. \end{aligned} \tag{2}$$

Example 2: Some seemingly very different problems can also be of the form of (1) under appropriate assumptions. We next consider the *optimal routing problem.* The network has L links. The capacity of each link l is R^l. There are I classes of users. Each class i has $J(i)$ alternate paths through the network. H_{ij}^l is defined as in Example 1. Flows of class i arrive to the network according to a Poisson process with rate λ_i. Each flow of class i, if admitted, will hold r_i amount of resource along the path it is routed to, and generate v_i amount of revenue per unit time. The service time distribution for flows of class i is *general* with mean $1/\mu_i$. The service times are i.i.d. and independent of the arrivals. The objective of the network is, by making admission and routing decisions for each incoming flow, to maximize the revenue collected from the flows that are admitted into the network.

The optimal routing problem is important for networks that attempt to support flows with rigid Quality of Service requirements. Unfortunately, the optimal routing policy is usually difficult to solve. When the service time distribution is exponential, we can formulate the problem as a Dynamic Programming problem. However, the complexity of the solution grows exponentially as the number of classes increases. When the service time distribution is general, there are few tools to solve for the optimal routing policy. Hence, most QoS routing proposals use various kinds of heuristics, such as the Widest-Shortest-Path algorithm [11], etc. However, the performance analysis of these schemes remains a challenging task. In most cases, researchers have to resort to simulation studies in order to understand the various performance tradeoff. This is especially the case when the network can only afford infrequent communication and computation [1, 11, 13].

Despite the difficulty in solving the optimal routing policy, there exists a *simple* solution to the optimal routing problem that is *asymptotically optimal when the capacity of the system is large.* It can be shown that the optimal long term average revenue that the network can ever possibly achieved is bounded by the following upper bound [6]:

$$\begin{aligned} \max_{[p_{ij}]\in\Omega} \quad & \sum_{i=1}^{I} \frac{\lambda_i}{\mu_i}\sum_{j=1}^{J(i)} p_{ij}v_i \\ \text{subject to} \quad & \sum_{i=1}^{I}\sum_{j=1}^{J(i)} \frac{\lambda_i}{\mu_i} r_i p_{ij} H_{ij}^l \leq R^l \quad \text{for all } l, \end{aligned} \tag{3}$$

where $\Omega = \bigotimes_{i=1}^{I}\Omega_i$, $\Omega_i = \{p_{ij} \geq 0, \sum_{j=1}^{J(i)} p_{ij} \leq 1, \text{ for all } j\}$. The solution to the upper bound, $\vec{p} = [p_{ij}]$, induces the following simple *proportional routing scheme.* In this scheme, each flow of class i will be admitted with probability $\sum_{j=1}^{J(i)} p_{ij}$, and once admitted, will be routed to path j with probability $p_{ij}/\sum_{j=1}^{J(i)} p_{ij}$. One can show that the above scheme asymptotically achieves the upper bound when the capacity of the network is large [6]. *This result holds even if the service time distribution is general.* Hence, if we are able to solve (3), we will obtain a simple and asymptotically optimal routing scheme. Note that (3) is a special case of (1) where the utility function is linear. It is also possible to generalize the result to the case of strictly concave utility functions [9].

Example 3: We can further extend the above model to the case when the network can use pricing to control the users' behavior. The network model is similar to that of Example 2. However, the arrival rate of flows of class i is now dependent on the price u_i that the network charged to class i, i.e., $\lambda_i = \lambda_i(u_i)$. Here u_i is the price the flow has to pay per unit time of connection. The price u_i is declared by the network at the time of the flow's arrival. Assume $\lambda_i(u_i)$ to be continuous and strictly decreasing. In a *dynamic pricing scheme*, the prices and the routing decisions could both be time-varying, depending on the current congestion level of the network. The objective of the network is, by appropriately pricing the resources, to maximize the revenue from the flows that are served. As in Example 2, this *optimal pricing problem* is difficult to solve except for a small scale system and under Markovian assumptions [8].

Let $u_i(\lambda_i)$ be the inverse function of $\lambda_i(u_i)$. $u_i(\lambda_i)$ is defined on the range of the function $\lambda_i(u_i)$ (denoted as $\mathbf{range}(i)$). Let $F_i(\lambda_i) = \lambda_i u_i(\lambda_i)$. If F_i is concave, one can show that the optimal revenue that can be achieved by a dynamic pricing scheme is bounded by the following upper bound [8]:

$$\max_{[\lambda_{ij}]\in\Omega} \quad \sum_{i=1}^{I} \frac{1}{\mu_i} F_i\Big(\sum_{j=1}^{J(i)} \lambda_{ij}\Big) \tag{4}$$

$$\text{subject to} \quad \sum_{i=1}^{I}\sum_{j=1}^{J(i)} \frac{\lambda_{ij}}{\mu_i} r_i H_{ij}^l \le R^l \quad \text{for all } l,$$

where $\Omega = \bigotimes_{i=1}^{I}\Omega_i$, $\Omega_i = \{\lambda_{ij} \ge 0 \text{ for all } j \text{ and } \sum_{j=1}^{J(i)} \lambda_{ij} \in \mathbf{range}(i)\}$. Further, *when the capacity of the network is large*, the following simple *static pricing scheme* achieves the upper bound asymptotically [8]. Let $\vec{\lambda} = [\lambda_{ij}]$ be the solution to the upper bound. In the static pricing scheme, the price charged to class i is $u_i = u_i(\sum_{j=1}^{J(i)} \lambda_{ij})$, and each flow is routed to path j with probability $\lambda_{ij}/\sum_{j=1}^{J(i)} \lambda_{ij}$. Therefore, if the capacity of the network is large, the optimal pricing problem can also be formulated as a multi-path utility maximization problem.

All of these problems can be mapped to the basic form in (1). In the next section, we will present a distributed solution to (1) that is amenable to online implementation. The convergence of the distributed algorithm will be established in Section 3.

2 The Distributed Algorithm

The difficulty in solving (1) is that the objective function is not strictly concave. The function f_i might be linear (as in Example 2). Even if f_i is strictly concave, the whole objective function is not, due to the linear relationship in $\sum_{j=1}^{J(i)} x_{ij}$. When one attempts to use a duality approach, the dual problem may not be differentiable at every point. To circumvent this difficulty, we use ideas from Proximal Optimization Algorithms [3]. The idea is to add a quadratic term to the objective function. We introduce an auxiliary variable y_{ij} for each x_{ij}. Let $\vec{y_i} = [y_{ij}, j = 1, ..., J(i)]$ and $\vec{y} = [\vec{y_1}, .., \vec{y_I}]$. The optimization becomes:

$$\begin{aligned} \max_{\vec{x}\in C, \vec{y}\in C} \quad & \sum_{i=1}^{I} f_i(\sum_{j=1}^{J(i)} x_{ij}) \quad - \sum_{i=1}^{I}\sum_{j=1}^{J(i)} \frac{c_i}{2}(x_{ij} - y_{ij})^2 \qquad (5) \\ \text{subject to} \quad & \sum_{i=1}^{I}\sum_{j=1}^{J(i)} E_{ij}^l x_{ij} \le R^l \quad \text{for all } l, \end{aligned}$$

where c_i is a positive number chosen for each i. It is easy to show that the optimal value of (5) coincides with that of (1). In fact, Let $\vec{x^*}$ denote the maximizer of (1), then $\vec{x} = \vec{x^*}, \vec{y} = \vec{x^*}$ is the maximizer of (5). Note that although a maximizer of (1) always exists, it is usually not unique since the objective function is not strictly concave.

The standard Proximal Optimization Algorithm then proceeds as follows:

Algorithm $\mathcal{P}$:

At the t-th iteration,

- P1) Fix $\vec{y} = \vec{y}(t)$ and minimize the augmented objective function with respect to $\vec{x}$. To be precise, this step solves:

$$\begin{aligned} \max_{\vec{x}\in C} \quad & \sum_{i=1}^{I} f_i(\sum_{j=1}^{J(i)} x_{ij}) \quad - \sum_{i=1}^{I}\sum_{j=1}^{J(i)} \frac{c_i}{2}(x_{ij} - y_{ij})^2 \qquad (6) \\ \text{subject to} \quad & \sum_{i=1}^{I}\sum_{j=1}^{J(i)} E_{ij}^l x_{ij} \le R^l \quad \text{for all } l. \end{aligned}$$

  Since the primal objective is now strictly concave, the maximizer exists and is unique. Let $\vec{x}(t)$ be the solution to this optimization.

- P2) Set $\vec{y}(t+1) = \vec{x}(t)$.

Step P1) can now be solved through its dual. Let $q^l, l = 1, ..., L$ be the Lagrange Multipliers for the constraints in (6). Let $\vec{q} = [q^1, ..., q^L]^T$. Define the Lagrangian as:

$$\begin{aligned} L(\vec{x}, \vec{q}, \vec{y}) &= \sum_{i=1}^{I} f_i(\sum_{j=1}^{J(i)} x_{ij}) - \sum_{l=1}^{L} q^l(\sum_{i=1}^{I}\sum_{j=1}^{J(i)} E_{ij}^l x_{ij} - R^l) - \sum_{i=1}^{I}\sum_{j=1}^{J(i)} \frac{c_i}{2}(x_{ij} - y_{ij})^2 \qquad (7) \\ &= \sum_{i=1}^{I} \left\{ f_i(\sum_{j=1}^{J(i)} x_{ij}) - \sum_{j=1}^{J(i)} x_{ij} \sum_{l=1}^{L} E_{ij}^l q^l - \sum_{j=1}^{J(i)} \frac{c_i}{2}(x_{ij} - y_{ij})^2 \right\} + \sum_{l=1}^{L} q^l R^l. \end{aligned}$$

Let $q_{ij} = \sum_{l=1}^{L} E_{ij}^l q^l$, $\vec{q_i} = [q_{ij}, j = 1, ..., J(i)]$. The objective function of the dual problem is then:

$$D(\vec{q}, \vec{y}) = \max_{\vec{x}\in C} L(\vec{x}, \vec{q}, \vec{y}) = \sum_{i=1}^{I} B_i(\vec{q_i}, \vec{y_i}) + \sum_{l=1}^{L} q^l R^l, \qquad (8)$$

where

$$B_i(\vec{q}_i, \vec{y}_i) \;=\; \max_{\vec{x}_i \in C_i} \left\{ f_i(\sum_{j=1}^{J(i)} x_{ij}) - \sum_{j=1}^{J(i)} x_{ij} q_{ij} - \sum_{j=1}^{J(i)} \frac{c_i}{2}(x_{ij} - y_{ij})^2 \right\}. \tag{9}$$

Note that in the definition of the dual objective function $D(\vec{q}, \vec{y})$ in (8), we have decomposed the original problem into I separate subproblems. Given $\vec{q}$, each subproblem B_i (9) can be solved independently. Using terminologies from the multi-path flow control problem (Example 1), if we interpret q^l as the *implicit cost* per unit bandwidth at link l, then q_{ij} is the total cost per unit bandwidth for all links in the path j of class i. Thus q_{ij}'s capture all the information we need about the paths user i traverses so that we can determine x_{ij}.

The dual problem of (6), given $\vec{y}$, is:

$$\min_{\vec{q} \geq 0} D(\vec{q}, \vec{y}).$$

Since the objective function of the primal problem (6) is strictly concave, the dual is always differentiable. The gradient of D is

$$\frac{\partial D}{\partial q^l} = R^l - \sum_{i=1}^{I} \sum_{j=1}^{J(i)} E_{ij}^l x_{ij}^0. \tag{10}$$

where x_{ij}^0's solve the local subproblems (9). The step P1) can then be solved by gradient descent iterations on the dual variable, i.e.,

$$q^l(t+1) = \left[q^l(t) - \alpha^l (R^l - \sum_{i=1}^{I} \sum_{j=1}^{J(i)} E_{ij}^l x_{ij}^0) \right]^+, \tag{11}$$

where $[.]^+$ denotes the projection to $[0, +\infty)$.

In the rest of the paper, we will consider the following class of distributed algorithms:

Algorithm $\mathcal{A}$:

- A1) Fix $\vec{y} = \vec{y}(t)$ and use gradient descent iteration (11) on the dual variable $\vec{q}$. Depending on the number of times the descent iteration is executed, we will obtain a dual variable $\vec{q}(t+1)$ that either exactly or approximately minimizes $D(\vec{q}, \vec{y}(t))$. Let K be the number of times the dual descent iteration is executed.

- A2) Let $\vec{x}(t)$ be the primal variable that maximizes the Lagrangian $L(\vec{x}, \vec{q}(t+1), \vec{y}(t))$ corresponding to the new dual variable $\vec{q}(t+1)$. Set $\vec{y}_i(t+1) = \vec{y}_i(t) + \beta_i(\vec{x}_i(t) - \vec{y}_i(t))$, where $0 < \beta_i \leq 1$ is a relaxation parameter for each i.

From now on, we will refer to (11) as the dual update, and step A2) as the primal update.

When K (the number of dual updates taken in step A1) equals to $+\infty$, (6) will be solved exactly provided that the stepsize α^l is small. Algorithm $\mathcal{A}$ and $\mathcal{P}$ are then equivalent. Using standard results in Proximal Point Algorithms [3, 12], one can show that both algorithms will converge to one solution of the original problem (1). If $1 \leq K < \infty$, at best an approximate solution to (6) is obtained at each iteration. If the accuracy of the approximate solution can be controlled appropriately (see [12]), one can still show the convergence of the algorithm. However, the number of dual updates, K, has to depend on the required accuracy and usually needs to be large.

In online implementation, however, it is impractical to carry out an algorithm in phases where each phase consists of infinite number of dual updates. It is also difficult to control the accuracy of the approximate solution to (6) in a distributed fashion. Hence, in this work we take a different approach. We allow arbitrary choice of $K \geq 1$ and we do not have any requirement on the accuracy of the approximate solution.

Define *a stationary point of the algorithm* $\mathcal{A}$ to be a primal-dual pair $(\vec{y^*}, \vec{q^*})$ such that

$$\vec{x} = \vec{y^*} \text{ maximize } L(\vec{x}, \vec{q^*}, \vec{y^*}) \ ,$$
$$\vec{q^*} \text{ is also a stationary point of (11) .}$$

By standard duality theory, any stationary point $(\vec{y^*}, \vec{q^*})$ of the algorithm $\mathcal{A}$ solves the augmented problem (5). Hence $\vec{x} = \vec{y^*}$ solves the original problem (1).

The following main result shows that K does not need to be large at all for the algorithm to converge. We will prove it in Section 3.

Proposition 1 *Fix $1 \leq K \leq \infty$. As long as the stepsize α^l is small enough, the algorithm $\mathcal{A}$ will converge to a stationary point $(\vec{y^*}, \vec{q^*})$ of the algorithm, and $\vec{x^*} = \vec{y^*}$ solves the original problem (1). The sufficient condition for convergence is:*

$$\max_l \alpha^l < \begin{cases} \frac{2}{\mathcal{S}\mathcal{L}} \min_i c_i & \text{if } K = \infty \\ \frac{1}{2\mathcal{S}\mathcal{L}} \min_i c_i & \text{if } K = 1 \\ \frac{4}{5K(K+1)\mathcal{S}\mathcal{L}} \min_i c_i & \text{if } K > 1 \end{cases} ,$$

where $\mathcal{L} = \max\{\sum_{l=1}^{L} E_{ij}^l, i = 1, ..., I, j = 1, ...J(i)\}$, and $\mathcal{S} = \max\{\sum_{i=1}^{I} \sum_{j=1}^{J(i)} E_{ij}^l, l = 1, ..., L\}$.

Proposition 1 shows that the algorithm converges even with $K = 1$. Further, the sufficient condition for convergence when $K = 1$ differs from that of $K = \infty$ only by a constant factor. Note that c_i appears on the right hand side. Hence, by making the objective function more concave, we also relax the requirement on the stepsize α^l. For $K > 1$, our result requires that the stepsize to be inversely proportional to K^2. This is probably not as tight as one could get: we conjecture that the same condition for $K = 1$ would work for any K. We leave it for our future work.

Another observation is that there is no requirement on the relaxation parameter β_i. It can thus be chosen freely.

Algorithm $\mathcal{A}$ lends naturally to distributed implementations online. For example, in the multi-path flow control problem (Example 1), each user only needs q_{ij} to determine its rate x_{ij} over all alternate paths. The core routers bear the responsibility to update the implicit costs q^l according to (11), based on the difference between the capacity R^l and the observed total traffic $\sum_{i=1}^{I} \sum_{j=1}^{J(i)} E_{ij}^l x_{ij}^0$. Hence, all computation is based on local information.

3 Proof of Proposition 1

Due to space constraints, we will sketch the main idea of the proof only. The details are in [10]. For convenience, we use vector notations wherever possible. Let E denote the matrix with L rows and $\sum_{i=1}^{I} J(i)$ columns such that the $(l, \sum_{k=1}^{i-1} J(k) + j)$ element is E_{ij}^l. Let $R = [R^1, R^2, ...R^l]^T$. Let V and B be the $\sum_{i=1}^{I} J(i) \times \sum_{i=1}^{I} J(i)$ diagonal matrices, with diagonal terms being c_i and β_i, respectively. (Each c_i or β_i is repeated $J(i)$ times.) Let A be the $L \times L$ diagonal matrix with diagonal terms being α^l.

In the sequel, it will be convenient to view the objective function in (1) as a concave function of $\vec{x}$, i.e, $\mathbf{f}(\vec{x}) = \sum_{i=1}^{I} f_i(\sum_{j=1}^{J(i)} x_{ij})$. Further, we can incorporate the constraint $\vec{x} \in C$ into the definition of the function $\mathbf{f}$ by setting $\mathbf{f}(\vec{x}) = -\infty$ if $\vec{x} \notin C$. Then the function $\mathbf{f}$ is still concave, and the problem (1) can be simply rephrased as maximizing $\mathbf{f}(\vec{x})$ subject to $E\vec{x} \leq R$. The Lagrangian (7) also becomes:

$$L(\vec{x}, \vec{q}, \vec{y}) \;=\; \mathbf{f}(\vec{x}) - \vec{x}^T E^T \vec{q} - \frac{1}{2}(\vec{x} - \vec{y})^T V (\vec{x} - \vec{y}) + \vec{q}^T R. \tag{12}$$

Using vector notations, the algorithm $\mathcal{A}$ can be rewritten as:

- A1) Let $\vec{q}(t, 0) = \vec{q}(t)$. Repeat for each $k = 0, 1, ...K - 1$:

  Let $\vec{x}(t, k) = \operatorname{argmax}_{\vec{x}} L(\vec{x}, \vec{q}(t, k), \vec{y}(t))$. Update the dual variables by

  $$\vec{q}(t, k + 1) = [\vec{q}(t, k) + A(E\vec{x}(t, k) - R)]^+. \tag{13}$$

- A2) Let $\vec{q}(t + 1) = \vec{q}(t, K)$. Let $\vec{z}(t) = \operatorname{argmax}_{\vec{x}} L(\vec{x}, \vec{q}(t + 1), \vec{y}(t))$. Update the primal variables by

  $$\vec{y}(t + 1) = \vec{y}(t) + B(\vec{z}(t) - \vec{y}(t)).$$

3.1 Preliminaries

Fix $\vec{y} = \vec{y}(t)$. Given an implicit cost vector $\vec{q}$, let $\vec{x}_0 = \operatorname{argmax}_{\vec{x}} L(\vec{x}, \vec{q}, \vec{y})$. By taking subgradients (see [12]) of the Lagrangian (12) with respect to $\vec{x}$, we can conclude that there must exist a subgradient $\nabla \mathbf{f}(\vec{x}_0)$ of $\mathbf{f}$ at $\vec{x}_0$ such that

$$\nabla \mathbf{f}(\vec{x}_0)|_{\vec{y},\vec{q}} - E^T \vec{q} - V(\vec{x}_0 - \vec{y}) = 0. \tag{14}$$

Similarly, let $(\vec{y^*}, \vec{q^*})$ denote a stationary point of algorithm $\mathcal{A}$. Then $\vec{y^*} = \operatorname{argmax}_{\vec{x}} L(\vec{x}, \vec{q^*}, \vec{y^*})$, and

$$\nabla \mathbf{f}(\vec{y^*})|_{\vec{y^*},\vec{q^*}} - E^T \vec{q^*} = 0. \tag{15}$$

Note that the $\nabla \mathbf{f}(\vec{x}_0)|_{\vec{y},\vec{q}}$ defined above depends not only on the function $\mathbf{f}$ and the vector $\vec{x}_0$, but also on $\vec{y}$ and $\vec{q}$. However, in the derivation that follows, the dependence on $\vec{y}$ and $\vec{q}$ is easy to identify. Hence, for the sake of brevity, we will drop the subscripts and write $\nabla \mathbf{f}(\vec{x}_0)$ (and $\nabla \mathbf{f}(\vec{y^*})$) when there is no ambiguity.

By the concavity of $\mathbf{f}$, we can show the following Lemma. The proof is quite technical and is available in [10].

Lemma 2 *Fix $\vec{y} = \vec{y}(t)$. Let $\vec{q}_1, \vec{q}_2$ be two implicit cost vectors, and let $\vec{x}_1$, $\vec{x}_2$ be the corresponding maximizers of the Lagrangian (12), i.e., $\vec{x}_1 = \operatorname{argmax}_{\vec{x}} L(\vec{x}, \vec{q}_1, \vec{y})$ and $\vec{x}_2 = \operatorname{argmax}_{\vec{x}} L(\vec{x}, \vec{q}_2, \vec{y})$. Then,*

1. $(\vec{x}_2 - \vec{x}_1)^T V (\vec{x}_2 - \vec{x}_1) \leq (\vec{q}_2 - \vec{q}_1)^T E V^{-1} E^T (\vec{q}_2 - \vec{q}_1)$ *, and*

2. $\left[\nabla \mathbf{f}(\vec{x}_1) - \nabla \mathbf{f}(\vec{y^*})\right]^T (\vec{x}_2 - \vec{y^*}) \leq \frac{1}{2}(\vec{q}_2 - \vec{q}_1)^T E V^{-1} E^T (\vec{q}_2 - \vec{q}_1)$.

Remark: The first part of the Lemma tells us that the mapping from $\vec{q}$ to $\vec{x}$ is continuous. In the second part, if we let $\vec{q}_2 = \vec{q}_1$, then $\vec{x}_2 = \vec{x}_1$ and we get $\left[\nabla \mathbf{f}(\vec{x}_1) - \nabla \mathbf{f}(\vec{y^*})\right]^T (\vec{x}_1 - \vec{y^*}) \leq 0$, which is simply the concavity of $\mathbf{f}$. The second part of the Lemma tells us that as long as $\vec{q}_1$ is not very different from $\vec{q}_2$, the cross-product on the left hand side in the second part of the Lemma will not be far above zero either.

3.2 The Main Proof

Now we can proceed with the main proof. We will focus on the case when $K = 1$. The proofs for the other cases are in [10]. Define the following norms:

$$||\vec{q}||_A = \vec{q}^T A^{-1} \vec{q}, \quad ||\vec{y}||_V = \vec{y}^T V \vec{y}, \quad ||\vec{y}||_{BV} = \vec{y}^T B^{-1} V \vec{y}.$$

Let $(\vec{y^*}, \vec{q^*})$ be any stationary point of algorithm $\mathcal{A}$. Our approach is to show that the norm $||\vec{q}(t) - \vec{q^*}||_A + ||\vec{y}(t) - \vec{y^*}||_{BV}$ is decreasing with respect to t. When $K = 1$, $\vec{q}(t+1) = [\vec{q}(t) + A(E\vec{x}(t) - R)]^+$. Using the property of the projection mapping, ([3, Proposition 3.2(b), p211], we have

$$\begin{aligned}
||\vec{q}(t+1) - \vec{q^*}||_A &\leq ||\vec{q}(t) - \vec{q^*}||_A - ||\vec{q}(t+1) - \vec{q}(t)||_A + 2(\vec{q}(t+1) - \vec{q^*})^T (E\vec{x}(t) - R) \\
&\leq ||\vec{q}(t) - \vec{q^*}||_A - ||\vec{q}(t+1) - \vec{q}(t)||_A + 2(\vec{q}(t+1) - \vec{q^*})^T E(\vec{x}(t) - \vec{y^*})
\end{aligned}$$

where in the last step we have used the properties of the stationary point $(\vec{y^*}, \vec{q^*})$ of the algorithm $\mathcal{A}$, i.e., $E\vec{y^*} - R \leq 0$ and $\vec{q^*}^T (E\vec{y^*} - R) = 0$. Since $y_{ij}(t+1) = (1 - \beta_i) y_{ij}(t) + \beta_i z_{ij}(t)$, we have

$$\begin{aligned}
(y_{ij}(t+1) - y^*_{ij})^2 &\leq (1 - \beta_i)(y_{ij}(t) - y^*_{ij})^2 + \beta_i (z_{ij}(t) - y^*_{ij})^2 \\
||\vec{y}(t+1) - \vec{y^*}||_{BV} - ||\vec{y}(t) - \vec{y^*}||_{BV} &\leq ||\vec{z}(t) - \vec{y^*}||_V - ||\vec{y}(t) - \vec{y^*}||_V. \qquad (16)
\end{aligned}$$

Hence,

$$\begin{aligned}
& ||\vec{q}(t+1) - \vec{q^*}||_A + ||\vec{y}(t+1) - \vec{y^*}||_{BV} - (||\vec{q}(t) - \vec{q^*}||_A + ||\vec{y}(t) - \vec{y^*}||_{BV}) \\
\leq\ & -||\vec{q}(t+1) - \vec{q}(t)||_A + 2(\vec{q}(t+1) - \vec{q^*})^T E(\vec{x}(t) - \vec{y^*}) \\
& \quad + ||\vec{z}(t) - \vec{y^*})||_V - ||\vec{y}(t) - \vec{y^*}||_V \\
\leq\ & -||\vec{q}(t+1) - \vec{q}(t)||_A \\
& \quad + \left\{ ||\vec{z}(t) - \vec{y^*})||_V - ||\vec{y}(t) - \vec{y^*}||_V - 2(\vec{z}(t) - \vec{y}(t))^T V(\vec{x}(t) - \vec{y^*}) \right\} \qquad (17) \\
& \quad + 2\left[\nabla \mathbf{f}(\vec{z}(t)) - \nabla \mathbf{f}(\vec{y^*})\right]^T (\vec{x}(t) - \vec{y^*}), \qquad (18)
\end{aligned}$$

where in the last step we have used (14) and (15). By simple algebraic manipulation, we can show that the second term (17) is equal to

$$\begin{aligned}
& ||\vec{z}(t) - \vec{y^*}||_V - ||\vec{y}(t) - \vec{y^*}||_V - 2(\vec{z}(t) - \vec{y}(t))^T V(\vec{x}(t) - \vec{y^*}) \\
=\ & ||(\vec{z}(t) - \vec{x}(t)||_V - ||\vec{y}(t) - \vec{x}(t)||_V. \qquad (19)
\end{aligned}$$

Invoking Lemma 2, part 1,

$$||\vec{z}(t) - \vec{x}(t)||_V \leq (\vec{q}(t+1) - \vec{q}(t))^T E V^{-1} E^T (\vec{q}(t+1) - \vec{q}(t)). \qquad (20)$$

For the third term (18), we can invoke Lemma 2, part 2

$$2\left[\nabla \mathbf{f}(\vec{z}(t)) - \nabla \mathbf{f}(\vec{y^*})\right]^T (\vec{x}(t) - \vec{y^*}) \leq (\vec{q}(t+1) - \vec{q}(t))^T E V^{-1} E^T (\vec{q}(t+1) - \vec{q}(t)). \quad (21)$$

Therefore, by substituting (19-21) into (17-18), we have

$$\begin{aligned} & ||\vec{q}(t+1)-\vec{q^*}||_A+||\vec{y}(t+1)-\vec{y^*}||_{BV}-(||\vec{q}(t)-\vec{q^*}||_A+||\vec{y}(t)-\vec{y^*}||_{BV}) \\ \leq \ & -(\vec{q}(t+1)-\vec{q}(t))^T(A^{-1}-2EV^{-1}E^T)(\vec{q}(t+1)-\vec{q}(t))-||\vec{y}(t)-\vec{x}(t)||_V. \end{aligned}$$

Let $C_1 = A^{-1} - 2EV^{-1}E^T$. If C_1 is positive definite, then

$$||\vec{q}(t+1)-\vec{q^*}||_A+||\vec{y}(t+1)-\vec{y^*}||_{BV} \leq (||\vec{q}(t)-\vec{q^*}||_A+||\vec{y}(t)-\vec{y^*}||_{BV}).$$

From here we can show that the sequence $\{\vec{y}(t), \vec{q}(t), t = 1, 2...\}$ must converge to a stationary point of algorithm $\mathcal{A}$. Finally, it is easy to show that a sufficient condition for C_1 to be positive definite is

$$\max_l \alpha^l < \frac{1}{2\mathcal{SL}} \min_i c_i.$$

4 Related Works and Concluding Remarks

In this paper, we present a distributed solution to the multi-path utility maximization problem and prove its convergence. Our solution uses ideas from *proximal point algorithms* [4, 12]. However, our approach is different from the way the proximal point algorithms are used in the standard results in the literature, e.g., the Method of Multipliers (MoM) and the Alternate Direction Method of Multipliers (ADMM, see [3], p244-253). The spirit of ADMM is similar to our algorithm $\mathcal{A}$ using $K = 1$, namely, the inner step A1) does not need to be solved exactly. However, the augmentation of the quadratic terms in ADMM is different from ours. Using terminologies of the multi-path flow control problem (Example 1), the objective function in MoM and ADMM is augmented by a quadratic term composed of the difference between the load and the capacity at each link, i.e., $||E\vec{x} - R||_2$. It is still possible to derive a highly distributed algorithm, except that an additional step would be required, which divides the residue capacity $(R - E\vec{x})$ evenly among all users $i = 1, ..., I$, and communicates the share to each user. This would require more communication effort among different network elements than our algorithm. There are other ways to use the ADMM algorithm, possibly involving reformulation of the original problem [7]. However, our main algorithm $\mathcal{A}$ does not appear to be one that can be derived from reformulations of ADMM. Hence a new proof for convergence is required. Further, our main Proposition 1 also covers the more general case when $K \geq 1$.

The algorithm in [14] is also similar to our algorithm $\mathcal{A}$ using $K = 1$. In [14], the authors claim that their algorithm is one of the Arrow-Hurwicz algorithms [2]. However, the convergence of the Arrow-Hurwicz algorithm was established in [2] only for the case when the objective function is strictly concave, which is not true for the problem in hand. In this paper, we present a new result that characterizes the convergence correctly.

Another approach to solve the multi-path utility maximization problem is to replace the constraints by penalty functions. Unless in the limit, the penalty function method usually does not give exact solutions to the original problem. An exception is to use non-differentiable penalty functions. The method in [5] can be viewed as of this type.

In this work we prove the convergence of our algorithm in a synchronous and deterministic setting. It is interesting to study the case when computation is asynchronous and when the quantities of interest can only be observed with noise (for instance, in Example 2 & 3, where the constraints are expressed by the average load.). For our future work, we plan to study the convergence and stability of our algorithm under these settings.

References

[1] G. Apostolopoulos, R. Guerin, S. Kamat, and S. K. Tripathi, "Quality of Service Based Routing: A Performance Perspective," in *Proceedings of ACM SIGCOMM*, (Vancouver, Canada), pp. 17–28, September 1998.

[2] K. Arrow, L. Hurwicz, and H. Uzawa, *Studies in Linear and Nonlinear Programming.* Stanford, CA: Stanford University Press, 1958.

[3] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods.* New Jersey: Prentice-Hall, 1989.

[4] J. Eckstein, *Splitting Methods for Monotone Operators With Applications to Prallel Optimization.* PhD thesis, Massachusetts Institute of Technology, Department of Civil Engineering, 1989.

[5] K. Kar, S. Sarkar, and L. Tassiulas, "Optimization Based Rate Control for Multipath Sessions," *Technical Report No. 2001-1, Institute for Systems Research, University of Maryland*, 2001.

[6] P. B. Key, "Optimal Control and Trunk Reservation in Loss Networks," *Probability in the Engineering and Informational Sciences*, vol. 4, pp. 203–242, 1990.

[7] S. A. Kontogiorgis, *Alternating Directions Methods for the Parallel Solution of Large-Scale Block-Structured Optimization Problems.* PhD thesis, University of Wisconsin-Madison, Department of Computer Science, 1994.

[8] X. Lin and N. B. Shroff, "Simplification of Network Dynamics in Large Systems," in *Tenth International Workshop on Quality of Service(IWQoS 2002)*, (Miami Beach, FL), May 2002.

[9] X. Lin and N. B. Shroff, " An Optimization Based Approach for Quality of Service Routing in High-Bandwidth Networks," *Technical Report, Purdue University, http://min.ecn.purdue.edu/∼linx/papers.html*, 2003.

[10] X. Lin and N. B. Shroff, " The Multi-Path Utility Maximization Problem," *Technical Report, Purdue University, http://min.ecn.purdue.edu/∼linx/papers.html*, 2003.

[11] Q. Ma and P. Steenkiste, "On Path Selection for Traffic with Bandwidth Guarantees," in *IEEE ICNP*, 1997.

[12] R. T. Rockafellar, "Monotone Operators and the Proximal Point Algorithm," *SIAM J. Control and Optimization*, vol. 14, pp. 877–898, August 1976.

[13] A. Shaikh, J. Rexford, and K. Shin, "Evaluating the Impact of Stale Link State on Quality-of-Service Routing," *IEEE/ACM Transactions on Networking*, vol. 9, pp. 162–176, April 2001.

[14] W. Wang, M. Palaniswami, and S. H. Low, "Optimal Flow Control and Routing in Multi-Path Networks," *Performance Evaluation*, vol. 52, pp. 119–132, April 2003.