

Centralized Network Utility Maximization over Aggregate Flows

Riten Gupta

UtopiaCompression Corporation

11150 West Olympic Blvd. #820

Los Angeles, CA 90064

Email: riten@utopiacompression.com

Lieven Vandenberghe

Electrical Engineering Dept.

University of California, Los Angeles

Los Angeles, CA 90095

Email: vandenbe@ee.ucla.edu

Mario Gerla

Computer Science Dept.

University of California, Los Angeles

Los Angeles, CA 90095

Email: gerla@cs.ucla.edu

Abstract—We study a network utility maximization (NUM) decomposition in which the set of flow rates is grouped by source-destination pairs. We develop theorems for both single-path and multipath cases, which relate an arbitrary NUM problem involving all flow rates to a simpler problem involving only the aggregate rates for each source-destination pair. The optimal aggregate flows are then apportioned among the constituent flows of each pair. This apportionment is simple for the case of α -fair utility functions. We also show how the decomposition can be implemented with the alternating direction method of multipliers (ADMM) algorithm.

I. INTRODUCTION

The last two decades have seen a great deal of research in network utility maximization (NUM) [1] [2] [3], which has cast light on traditional networking protocols [4] and has facilitated the design of promising future protocols [5] as well. Most NUM researchers have focused on developing distributed solutions to various utility maximization problems. These distributed solutions, which follow nicely from dual decompositions [2], are ideal for internets, in which cooperation among flow sources cannot be assumed, and minimal communication between links and nodes is desired. In recent years there has been growing interest in the software defined networking (SDN) paradigm, in which data and control planes are separated [6] [7]. In this framework, certain network functions such as flow control, congestion control, and throughput optimization may be assigned to a central controller. Central control is feasible for closed networks, such as in data centers [8] or communication satellite networks [9].

In some networks with central control, the number of flows K may be much larger than the number of source-destination pairs N . For example, the Iridium satellite network employs 66 satellites and facilitates tens of thousands of flows [10]. A similar phenomenon may occur in small data centers. In this paper, we study a primal decomposition in which the set of flows is grouped into flow classes, each corresponding to a source-destination pair. Many congestion control algorithms inherently group flows by source-destination pair [11] and several related primal decompositions have been studied, for example in [2]. However, because the source-destination decomposition is only applicable to centralized control, it has received little attention. Given the recent popularity of SDN, however, the decomposition may prove to be beneficial. To

this end, we develop a comprehensive theory of the source-destination decomposition in this paper. (We also discuss briefly in Section V a potential benefit of this decomposition in a network with “semi-distributed” control.) We derive theorems that decompose a NUM problem with K variables into one with only N variables, followed by an allocation problem which apportion the aggregate rate for each class among the class’s constituent flows. In some cases, this apportionment is simple. In other cases, the alternating direction method of multipliers (ADMM) algorithm can exploit the decomposition numerically.

The remainder of the paper is organized as follows. In Section II we present the aggregate flow decomposition and the main results relating the original NUM problem to the simpler aggregate flow problem. This analysis is extended to the multipath case in Section III. In Section IV we discuss numerical algorithms, which exploit the aggregate flow decomposition. Finally, in Section V, we conclude the paper.

II. OPTIMIZATION OVER AGGREGATE FLOWS

Consider a communication network with M nodes and L links (edges). Let N be the number of source-destination pairs in use among all flows. Then $N \leq M(M-1)$. Number the source-destination pairs $1, \dots, N$ and call the set of flows in pair i the i th flow class. Let K_i be the number of flows in class i . Then the total number of flows is $K = \sum_{i=1}^N K_i$. Let $u_{i,k}$ be the rate of the k th flow in class i . Finally, define the $L \times N$ binary routing matrix \mathbf{R} as

$$R_{l,i} = \begin{cases} 1, & \text{traffic of class } i \text{ passes through link } l \\ 0, & \text{otherwise.} \end{cases}$$

Note that all flows within a class follow the same path. Consider the following utility maximization problem.

$$\begin{aligned} & \underset{\{u_{i,k}\}}{\text{maximize}} && \sum_{i=1}^N \sum_{k=1}^{K_i} f_{i,k}(u_{i,k}) \\ & \text{subject to} && \sum_{i=1}^N \sum_{k=1}^{K_i} R_{l,i} u_{i,k} \leq c_l, \quad l = 1, \dots, L \end{aligned} \quad (1)$$

where $f_{i,k}$ is a utility function for the k th flow in class i , c_l is the capacity of link l , and the constraints imply that no link is overloaded. Next, let $x_i = \sum_{k=1}^{K_i} u_{i,k}$ be the aggregate rate of

class i and consider the aggregate flow utility maximization problem

$$\begin{aligned} & \underset{\{x_i\}}{\text{maximize}} && \sum_{i=1}^N f_i(x_i) \\ & \text{subject to} && \sum_{i=1}^N R_{l,i} x_i \leq c_l, \quad l = 1, \dots, L \end{aligned} \quad (2)$$

where f_i is an aggregate utility function for class i . The domains of the utility functions $f_{i,k}$ and aggregate utility functions f_i are not stated here, but are usually subsets of \mathbb{R}_+ as negative flow rates are not allowed.

A. Decomposition by Supremal Convolutions

Definition 1: Let the functions f_1 and f_2 be concave and proper on \mathbb{R}^n . The supremal convolution of f_1 and f_2 is

$$(f_1 \diamond f_2)(x) = \sup_{\{(x_1, x_2): x_1 + x_2 = x\}} f_1(x_1) + f_2(x_2).$$

The supremal convolution of the concave functions f_1 and f_2 is simply the negative infimal convolution of the convex functions $-f_1$ and $-f_2$. By [12, Theorem 5.4], $f_1 \diamond f_2$ is concave. Observe that problem (2) is equivalent to problem (1) when each f_i is the K_i -fold supremal convolution

$$f_i(x_i) = \sup_{\substack{\{u_{i,1}, \dots, u_{i,K_i}\}: \\ \sum_k u_{i,k} = x_i}} \sum_k f_{i,k}(u_{i,k}) = (f_{i,1} \diamond \dots \diamond f_{i,K_i})(x_i)$$

and for each flow class i , the optimal subflow rates solve

$$\underset{\{u_{i,k}\}}{\text{maximize}} \sum_{k=1}^{K_i} f_{i,k}(u_{i,k}), \quad \text{subject to} \sum_{k=1}^{K_i} u_{i,k} = x_i^*$$

where x_i^* is the solution of problem (2) (provided it exists) with f_i defined as above. Thus, when the utilities are concave and proper, problem (1) can be decomposed into an aggregate optimization and N optimizations over the subflows as long as the supremal convolutions can be calculated. This decomposition lends itself to parallel implementations, as the N subproblems are independent.

Let $f^*(y) = \inf_x (xy - f(x))$ denote the concave Fenchel conjugate. From [12, Theorem 16.4], the conjugate supremal convolution is $(f_{i,1} \diamond \dots \diamond f_{i,K_i})^* = \sum_k f_{i,k}^*$. Thus the concave closure of the supremal convolution is $(\sum_k f_{i,k}^*)^*$. By [12, Corollary 20.1.1], if the $f_{i,k}$'s are closed and $\cap_k \text{relint}(\text{dom} f_{i,k}^*) \neq \emptyset$, then $f_{i,1} \diamond \dots \diamond f_{i,K_i}$ is closed, so

$$f_i = f_{i,1} \diamond \dots \diamond f_{i,K_i} = \left(\sum_k f_{i,k}^* \right)^*. \quad (3)$$

B. Decomposition with Functions of Legendre Type

Definition 2: A pair (f, \mathcal{D}) is of Legendre type if \mathcal{D} is a nonempty open convex set, f is a strictly concave differentiable function on \mathcal{D} and $\lim_{n \rightarrow \infty} \|\nabla f(x_n)\| = +\infty$ for any sequence $\{x_n\}$ in \mathcal{D} converging to a boundary point of \mathcal{D} .

Although the Legendre type property applies to a pair (f, \mathcal{D}) , we will refer to a function f as being of Legendre type when $(f, \text{int}(\text{dom} f))$ is of Legendre type. Note that when

$\text{dom} f = \mathbb{R}_{++}$, the last condition in Definition 2 is equivalent to $\lim_{x \downarrow 0} f'(x) = +\infty$.

The (concave) Legendre conjugate (or Legendre transform) [12] [13] of a pair (f, \mathcal{D}) , where $\mathcal{D} \subset \mathbb{R}$ is open and f is differentiable on \mathcal{D} , is the pair (g, \mathcal{E}) where $g(y) = yf'^{-1}(y) - f(f'^{-1}(y))$ and \mathcal{E} is the image of \mathcal{D} under f' .

From [12, Theorem 26.5], if $f_{i,k}$ is closed, $\mathcal{D} = \text{int}(\text{dom} f_{i,k})$, and $\mathcal{D}^* = \text{int}(\text{dom} f_{i,k}^*)$, then $(f_{i,k}, \mathcal{D})$ is of Legendre type if and only if $(f_{i,k}^*, \mathcal{D}^*)$ is of Legendre type. When these pairs are of Legendre type, $(f_{i,k}^*, \mathcal{D}^*)$ is the Legendre conjugate of $(f_{i,k}, \mathcal{D})$, which is the Legendre conjugate of $(f_{i,k}^*, \mathcal{D}^*)$, so conjugation is involutory: $(f_{i,k}^{**}, \mathcal{D}^{**}) = (f_{i,k}, \mathcal{D})$, and

$$(f_{i,k}^*)' = f_{i,k}'^{-1}. \quad (4)$$

Note that if the $f_{i,k}$'s are closed and Legendre type, with domain \mathcal{D} and $\cap_k \text{relint}(\text{dom} f_{i,k}^*) \neq \emptyset$, then $\sum_k f_{i,k}^*$ is Legendre type (and therefore differentiable), and the supremal convolution $(\sum_k f_{i,k}^*)^*$ is closed and Legendre type.

Let the $f_{i,k}$'s have domain $\mathcal{D} = \mathbb{R}_{++}$. Since there are no equality constraints in problems (1) and (2) and the inequality constraints are all affine, Slater's condition guarantees strong duality for each problem as long as a feasible point exists in the relative interior of the problem domain [13, Sec. 5.2.3], which is \mathbb{R}_{++}^K for problem (1) and \mathbb{R}_{++}^N for (2). Clearly, setting all optimization variables to a small $\epsilon > 0$ yields such a point, so strong duality holds for both problems. With Legendre type functions, problem (1) is strictly concave with convex feasible region and has a unique solution. Thus there is a unique primal-dual optimal pair satisfying the Karush-Kuhn-Tucker (KKT) conditions for problem (1) with Legendre-type utility functions.

Theorem 1: Let the functions $f_{i,k}$ be closed, concave, and Legendre type with domain \mathbb{R}_{++} and $\cap_k \text{relint}(\text{dom} f_{i,k}^*) \neq \emptyset$. For each i, k , let $g_{i,k} = f_{i,k}^*$, $g_i = \sum_k g_{i,k}$, and $f_i = g_i^*$. Let $\{x_i^*\}$ be a primal solution to problem (2) with this definition of $\{f_i\}$. Then (1) has unique primal solution

$$u_{i,k}^* = g_{i,k}'(f_i'(x_i^*)), \quad \forall i, k \quad (5)$$

and the corresponding dual solutions of (1) and (2) are equal. Note that $f_i = f_{i,1} \diamond \dots \diamond f_{i,K_i}$.

Proof. First note that f_i and $g_{i,k}$ are Legendre type and therefore differentiable. Let $h_{i,k} = g_{i,k}'$ for each i, k . From (4) we have $h_{i,k} = f_{i,k}'^{-1}$. The Lagrangian for problem (1) is

$$\mathcal{L}_1(\mathbf{u}, \boldsymbol{\rho}) = \sum_i \sum_k f_{i,k}(u_{i,k}) - \sum_{l=1}^L \rho_l \left(\sum_i \sum_k R_{l,i} u_{i,k} - c_l \right).$$

The KKT sufficient conditions for optimality of (1) are

$$\sum_i R_{l,i} \sum_k u_{i,k} \leq c_l, \quad \forall l \quad (6)$$

$$\boldsymbol{\rho} \geq \mathbf{0} \quad (7)$$

$$\rho_l \left(\sum_i R_{l,i} \sum_k u_{i,k} - c_l \right) = 0, \quad \forall l \quad (8)$$

$$u_{i,k} = h_{i,k}(\boldsymbol{\rho}^T \mathbf{r}_i), \quad \forall i, k \quad (9)$$

where \mathbf{r}_i is the i th column of \mathbf{R} . Condition (9) is equivalent to $\partial \mathcal{L}_1 / \partial u_{i,k} = 0$. Now, let

$$h_i = \sum_k h_{i,k} = \sum_k g'_{i,k} = g'_i$$

for each i and consider problem (2) with $f_i = g'_i$. Since f_i is Legendre type, it is strictly concave and thus $\{x_i^*\}$ is the unique solution to problem (2). Since, in addition, f_i is closed, we can use (4) to get $h_i = f_i'^{-1}$. The Lagrangian is

$$\mathcal{L}_2(\mathbf{x}, \boldsymbol{\lambda}) = \sum_i f_i(x_i) - \sum_l \lambda_l \left(\sum_i R_{l,i} x_i - c_l \right).$$

The KKT conditions for problem (2) are thus

$$\begin{aligned} \sum_i R_{l,i} x_i &\leq c_l, \quad \forall l \\ \boldsymbol{\lambda} &\geq \mathbf{0} \\ \lambda_l \left(\sum_i R_{l,i} x_i - c_l \right) &= 0, \quad \forall l \\ x_i &= h_i(\boldsymbol{\lambda}^T \mathbf{r}_i), \quad \forall i. \end{aligned} \quad (10)$$

Next, let $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ be the primal-dual solution for problem (2) and set $\boldsymbol{\rho} = \boldsymbol{\lambda}^*$. Then condition (7) is immediately satisfied. Next let

$$u_{i,k} = h_{i,k}(\boldsymbol{\lambda}^{*T} \mathbf{r}_i) = h_{i,k}(f'_i(x_i^*)), \quad \forall i, k.$$

Then condition (9) is satisfied and using $h_i = f_i'^{-1} = \sum_k h_{i,k}$, and (10), we have

$$\sum_k u_{i,k} = \sum_k h_{i,k}(\boldsymbol{\lambda}^{*T} \mathbf{r}_i) = h_i(\boldsymbol{\lambda}^{*T} \mathbf{r}_i) = x_i^*, \quad \forall i$$

which ensures $\sum_k u_{i,k}^* = x_i^*$, and therefore conditions (6) and (8) are satisfied. Finally, since the image of $h_{i,k}$ is \mathbb{R}_{++} , $u_{i,k} > 0$ for each i, k . \square

C. Examples

Here we apply the aggregate flow decomposition to some example utility maximization problems. Theorem 1 can be applied to utility functions belonging to the family of α -fair functions [14], while the decomposition using supremal convolutions must be used for more general problems.

1) *Weighted Logarithm Utilities*: Let $f_{i,k}(u_{i,k}) = w_{i,k} \log u_{i,k}$ with $w_{i,k} \geq 0$, so that the overall utility is a sum of weighted logarithms of individual flows. These functions belong to the class of α -fair utilities with $\alpha = 0$ and are appealing as they yield proportionally fair rate allocations [14]. They are also clearly Legendre type so Theorem 1 can be used. The Legendre conjugates can be calculated using (4). We have $f'_{i,k}(u) = w_{i,k}/u$, $g'_{i,k}(v) = w_{i,k}/v$, and $g'_i(v) = \sum_k w_{i,k}/v$. Next $f'_i(x) = g_i'^{-1}(x) = \sum_k w_{i,k}/x$ and $f_i(x) = \sum_k w_{i,k} \log x$. Finally, using (5)

$$u_{i,k}^* = \frac{w_{i,k}}{\sum_{k'} w_{i,k'}} x_i^*. \quad (11)$$

Note that all of the above functions have domain \mathbb{R}_{++} . From (11), the optimized aggregate flows should be apportioned to the subflows in proportion to their weights. Note that the utility

function of problem (2) is also a sum of weighted logarithms, where the i th weight is the sum weight of the i th class.

The weighted logarithm case can also be proven using proportional fairness [15]. Let $\{x_i^*\}$ be the solution to problem (2) with $f_i(x_i) = w_i \log x_i$ and let the subflow rates be

$$u_{i,k} = \frac{w_{i,k}}{w_i} x_i^* \quad (12)$$

where $\{w_{i,k}\}$ are any non-negative weights such that $\sum_k w_{i,k} = w_i$. From [15], the unique solution to (2) $\{x_i^*\}$ is such that the rates per unit charge are proportionally fair. That is, if $\{\hat{x}_i\}$ is any other set of rates then

$$\sum_i w_i \frac{\hat{x}_i - x_i^*}{x_i^*} \leq 0. \quad (13)$$

Now let $\{\hat{u}_{i,k}\}$ be any set of subflow rates not equal to those found by (12) and let $\hat{x}_i = \sum_k \hat{u}_{i,k}$. From (12) we have $w_i/x_i^* = w_{i,k}/u_{i,k}$ for all i, k . From (13) we have

$$0 \geq \sum_i \frac{w_i}{x_i^*} \sum_k (\hat{u}_{i,k} - u_{i,k}) = \sum_i \sum_k w_{i,k} \frac{\hat{u}_{i,k} - u_{i,k}}{u_{i,k}}$$

So the proportionally allocated solution is such that the rates per unit charge are proportionally fair. Thus it is the unique solution to (1) with $f_{i,k}(u_{i,k}) = w_{i,k} \log u_{i,k}$.

2) *Weighted Power Utilities (Negative Exponent)*: As another example, let $f_{i,k}(u_{i,k}) = -w_{i,k} u_{i,k}^{-a}$ with $a \geq 1$. These functions are also part of the α -fair family. When $a = 1$, the allocation satisfies minimum potential delay fairness and as $a \rightarrow +\infty$, the allocation is max-min fair [14]. The utilities are also of Legendre type and we can use Theorem 1. We have $f'_{i,k}(u) = a w_{i,k} u^{-(a+1)}$ and $g'_{i,k}(v) = (a w_{i,k}/v)^{\frac{1}{a+1}}$. Next we have $g'_i(y) = \sum_k g'_{i,k}(y) = f_i'^{-1}(y)$. Thus $f'_i(x) = (a/x^{a+1})(\sum_k w_{i,k}^{\frac{1}{a+1}})^{a+1}$ and the optimum subflow rates are

$$u_{i,k}^* = g'_{i,k}(f'_i(x_i^*)) = \frac{w_{i,k}^{\frac{1}{a+1}}}{\sum_{k'} w_{i,k'}^{\frac{1}{a+1}}} x_i^*.$$

The utility functions for problem (2) are $f_i(x_i) = -x_i^{-a} (\sum_k w_{i,k}^{\frac{1}{a+1}})^{a+1}$. Again, all of the above functions have domain \mathbb{R}_{++} .

3) *Quadratic Utilities*: Quadratic functions are not of Legendre type and are not necessarily increasing on \mathbb{R}_+ , rendering them unsuitable for use as utility functions. However, the aggregate flow decomposition can be useful when implementing a gradient projection algorithm. In a gradient projection algorithm, steepest ascent iterations are followed by projections onto the feasible set [16]. Such a projection is a quadratic program (QP) that can be simplified by decomposing with supremal convolutions.

Let $\{z_{i,k}\}$ be the set of variables obtained after an iteration of steepest ascent for problem (1). This set must be projected onto the routing polytope $\{\{u_{i,k}\} : \sum_i \sum_k R_{l,i} u_{i,k} \leq c_l, l = 1, \dots, L\}$. The projection QP is problem (1) with quadratic utility $f_{i,k}(u_{i,k}) = -\frac{1}{2}(u_{i,k} - z_{i,k})^2$, and domain $\{u_{i,k} \geq 0\}$. (In this section all functions are equal to $-\infty$ outside

their domains). Note that $f_{i,k}$ is not Legendre type. However $f_{i,k}$ is closed, concave, and proper on \mathbb{R} and has conjugate $f_{i,k}^*(y) = -\frac{1}{2}y^2 + z_{i,k}y$, with domain $\{y \leq z_{i,k}\}$. Thus any point less than $z_{i,\min} = \min_k z_{i,k}$ lies in $\text{relint}(\text{dom} f_{i,k}^*)$ for all (i,k) , and therefore (3) can be used to find f_i . The conjugate aggregate utility is $f_i^*(y) = \sum_k f_{i,k}^*(y) = -K_i y^2/2 + \bar{z}_i y$, with domain $\{y \leq z_{i,\min}\}$, where $\bar{z}_i = \sum_k z_{i,k}$. The aggregate function is obtained by conjugating f_i^* , which yields $f_i(x_i) = -\frac{1}{2K_i}(x_i - \bar{z}_i)^2$, with domain $\{x_i \geq \bar{z}_i - K_i z_{i,\min}\}$. Finally, for each class i , the subflows minimize $\sum_k \frac{1}{2}(u_{i,k} - z_{i,k})^2$ subject to $\sum_k u_{i,k} = x_i^*$ and $u_{i,k} \geq 0$ for each k , where x_i^* is the solution to the aggregate problem. (Thus $x_i^* \geq \bar{z}_i - K_i z_{i,\min}$.) The subflow problem is strictly convex and has unique solution $u_{i,k}^* = z_{i,k} + \frac{1}{K_i}(x_i^* - \bar{z}_i)$.

4) *Piecewise Linear Utilities*: Piecewise linear functions are important as they are often used as approximations of functions that are difficult to work with analytically or are incompletely known [17]. In this case Theorem 1 is not applicable but supremal convolutions can be calculated using (3). Let $f_{i,k}$ be concave and piecewise linear with non-negative breakpoints $0 = c_1 < c_2 < \dots < c_B$ and corresponding non-negative slopes $m_1 > m_2 > \dots > m_B = 0$, and let $f_{i,k}(c_1) = f_{i,k}(0) = 0$ and $f_{i,k}(x) = -\infty$ for $x < 0$. (The number of breakpoints B need not be the same for all utilities.) Then $f_{i,k}$ is closed and from [17, Sec. 8F], the conjugate of $f_{i,k}$ is also concave and piecewise linear with breakpoints $0 = m_B < m_{B-1} < \dots < m_1$ and corresponding slopes $c_B > c_{B-1} > \dots > c_1$, and $f_{i,k}^*(m_1) = 0$. That is, the breakpoints of $f_{i,k}^*$ are the slopes of $f_{i,k}$ and the slopes of $f_{i,k}^*$ are the breakpoints of $f_{i,k}$. Finally, $\text{dom} f_{i,k}^* = \mathbb{R}_+$ and thus (3) can be used.

The aggregate utility function f_i can be found with the following prescription: For each $f_{i,k}$, find $f_{i,k}^*$ by exchanging breakpoints and slopes, as described above. Sum these conjugates to find the conjugate of the aggregate utility $f_i^* = \sum_k f_{i,k}^*$. Thus f_i^* is piecewise linear and concave as well. Finally, exchange slopes and breakpoints of f_i^* to arrive at f_i .

Therefore, the piecewise-linear problem, which is a linear program (LP) in $K = \sum_i K_i$ variables, can be decomposed into one LP in N variables, followed by N parallel sub-LP's, the i th sub-LP having K_i variables.

III. EXTENSION TO MULTIPATH CASE

Now suppose that for each flow class i , traffic may be split into subflows and routed over multiple paths. (Here, a subflow refers to that portion of a flow routed over a certain path, as opposed to a constituent flow of a flow class). Let J be the number of paths and assume J is the same for all flow classes. For each class i , define the $L \times J$ routing matrix \mathbf{S}_i as

$$[\mathbf{S}_i]_{l,j} = \begin{cases} 1, & \text{traffic on the } j\text{th path of class } i \\ & \text{passes through link } l \\ 0, & \text{otherwise.} \end{cases}$$

and let the overall $L \times NJ$ routing matrix be $\mathbf{R} = [\mathbf{S}_1, \dots, \mathbf{S}_N]$. Finally, let $u_{i,j,k}$ be the rate on the j th path of

flow k of class i . The multipath utility maximization problem is

$$\begin{aligned} & \text{maximize}_{\{u_{i,j,k}\}} && \sum_{i=1}^N \sum_{k=1}^{K_i} f_{i,k} \left(\sum_{j=1}^J u_{i,j,k} \right) \\ & \text{subject to} && \sum_{i=1}^N \sum_{j=1}^J \sum_{k=1}^{K_i} [\mathbf{S}_i]_{l,j} u_{i,j,k} \leq c_l, \quad \forall l \\ & && u_{i,j,k} \geq 0, \quad \forall i, j, k. \end{aligned} \quad (14)$$

Letting $x_{i,j} = \sum_k u_{i,j,k}$ be the aggregate rate on path j of class i , the aggregate flow problem in the multipath case is

$$\begin{aligned} & \text{maximize}_{\{x_{i,j}\}} && \sum_{i=1}^N f_i \left(\sum_{j=1}^J x_{i,j} \right) \\ & \text{subject to} && \sum_{i=1}^N \sum_{j=1}^J [\mathbf{S}_i]_{l,j} x_{i,j} \leq c_l, \quad \forall l \\ & && x_{i,j} \geq 0, \quad \forall i, j. \end{aligned} \quad (15)$$

For these problems, explicit constraints for non-negativity of the throughputs are added because, for example $\sum_j u_{i,j,k}$ can be non-negative even with some negative subflows. Unlike the single-path case, neither problem is strictly convex.

A. Multipath Supremal Convolution Decomposition

Let the $f_{i,k}$'s be concave and proper on \mathbb{R} with $\text{dom} f_{i,k} \subset \mathbb{R}_+$. For each i, k pair, define

$$\phi_{i,k}(\mathbf{u}_{i,k}) = \begin{cases} f_{i,k}(\mathbf{1}_J^T \mathbf{u}_{i,k}), & \mathbf{u}_{i,k} \in \mathbb{R}_+^J \\ -\infty, & \text{otherwise} \end{cases}$$

where $\mathbf{u}_{i,k} = [u_{i,1,k}, \dots, u_{i,J,k}]^T$. Then $\phi_{i,k}$ is concave and proper on \mathbb{R}^J (but not strictly concave, even if $f_{i,k}$ is) and problem (14) is equivalent to

$$\text{maximize}_{\{\mathbf{u}_{i,k}\}} \sum_{i=1}^N \sum_{k=1}^{K_i} \phi_{i,k}(\mathbf{u}_{i,k}) \quad (16)$$

subject to the link load constraints of (14). Define the aggregate flow problem by

$$\text{maximize}_{\{\mathbf{x}_i\}} \sum_{i=1}^N \phi_i(\mathbf{x}_i) \quad (17)$$

with the link load constraints of (15). Here $\phi_i(\mathbf{x}_i)$ is a function from \mathbb{R}^J to \mathbb{R} and $\mathbf{x}_i = [x_{i,1}, \dots, x_{i,J}]^T$. Similar to the argument in Section II-A, problem (17) is equivalent to problem (16) with concave aggregate functions $\phi_i = \phi_{i,1} \diamond \dots \diamond \phi_{i,K_i}$ if for each flow class i , the optimal subflow rates solve the problem

$$\text{maximize}_{\{\mathbf{u}_{i,k}\}} \sum_{k=1}^{K_i} \phi_{i,k}(\mathbf{u}_{i,k}), \quad \text{subject to } \sum_{k=1}^{K_i} \mathbf{u}_{i,k} = \mathbf{x}_i^*$$

where \mathbf{x}_i^* is the solution of problem (17) (provided it exists) with ϕ_i defined as above. If the $f_{i,k}$'s are closed and $\cap_k \text{relint}(\text{dom} f_{i,k}^*) \neq \emptyset$, then it can be shown that the same is true of the $\phi_{i,k}$'s and the aggregate functions can be found using $\phi_i = (\sum_k \phi_{i,k}^*)^*$ where $\phi^*(\mathbf{y}) = \inf_{\mathbf{u}} (\mathbf{u}^T \mathbf{y} - \phi(\mathbf{u}))$.

B. Multipath Legendre-Type Case

As in the single-path case, when $\text{dom} f_{i,k} = \mathbb{R}_{++}$, strong duality of (14) and (15) follows from Slater's condition. However, in the multipath case, neither problem is strictly convex and uniqueness of the solutions cannot be guaranteed.

Theorem 2: Let $f_{i,k}$, $g_{i,k}$, f_i , and g_i satisfy the conditions of Theorem 1 for all i, k . Let $\{x_{i,j}^*\}$ be a primal solution to (15) with this definition of $\{f_i\}$. Then a solution of the following constrained system of linear equations

$$\begin{aligned} \sum_k u_{i,j,k} &= x_{i,j}^*, \quad \forall i, j \\ \sum_j u_{i,j,k} &= g'_{i,k}(f'_i(\bar{x}_i^*)), \quad \forall i, k \\ u_{i,j,k} &\geq 0, \quad \forall i, j, k \end{aligned} \quad (18)$$

(where $\bar{x}_i^* = \sum_j x_{i,j}^*$), is a primal solution to (14). Furthermore, if λ^* is the dual solution to (15) corresponding to the link load constraints and $\mu_{i,j}^*$ is the dual solution to (15) corresponding to the non-negativity constraint of $x_{i,j}$, then the dual solution to (14) corresponding to the link load constraints is $\rho^* = \lambda^*$ and the dual solution to (14) corresponding to the non-negativity constraint of $u_{i,j,k}$ is $\sigma_{i,j,k}^* = \mu_{i,j}^*$, for each k .

Proof. Let $h_{i,k} = g'_{i,k} = f_{i,k}'^{-1}$ for each i, k and let $\bar{u}_{i,k} = \sum_j u_{i,j,k}$. The Lagrangian for problem (14) is

$$\begin{aligned} \mathcal{L}_1(\mathbf{u}, \boldsymbol{\rho}, \boldsymbol{\sigma}) &= \sum_{i,k} f_{i,k}(\bar{u}_{i,k}) - \sum_{l=1}^L \rho_l \left(\sum_{i,j,k} [\mathbf{S}_i]_{l,j} u_{i,j,k} - c_l \right) \\ &\quad + \sum_{i,j,k} \sigma_{i,j,k} u_{i,j,k}. \end{aligned}$$

Setting the derivative with respect to $u_{i,j,k}$ to zero gives

$$f'_{i,k}(\bar{u}_{i,k}) = [\mathbf{S}_i^T \boldsymbol{\rho}]_j - \sigma_{i,j,k}, \quad \forall i, j, k. \quad (19)$$

Thus, the following seven relations constitute the KKT conditions for problem (14):

$$\sum_{i,j,k} [\mathbf{S}_i]_{l,j} u_{i,j,k} \leq c_l, \quad \forall l \quad (20)$$

$$u_{i,j,k} \geq 0, \quad \forall i, j, k \quad (21)$$

$$\boldsymbol{\rho} \geq \mathbf{0} \quad (22)$$

$$\boldsymbol{\sigma} \geq \mathbf{0} \quad (23)$$

$$\rho_l \left(\sum_{i,j,k} [\mathbf{S}_i]_{l,j} u_{i,j,k} - c_l \right) = 0, \quad \forall l \quad (24)$$

$$\sigma_{i,j,k} u_{i,j,k} = 0, \quad \forall i, j, k \quad (25)$$

$$h_{i,k}([\mathbf{S}_i^T \boldsymbol{\rho}]_j - \sigma_{i,j,k}) = \bar{u}_{i,k}, \quad \forall i, j, k. \quad (26)$$

Next, turning to problem (15) with $f_i = g_i^*$, let $h_i = \sum_k h_{i,k} = \sum_k g'_{i,k} = g'_i$ for each i and let $\bar{x}_i = \sum_j x_{i,j}$. The Lagrangian is

$$\begin{aligned} \mathcal{L}_2(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) &= \sum_i f_i(\bar{x}_i) - \sum_{l=1}^L \lambda_l \left(\sum_{i,j} [\mathbf{S}_i]_{l,j} x_{i,j} - c_l \right) \\ &\quad + \sum_{i,j} \mu_{i,j} x_{i,j}. \end{aligned}$$

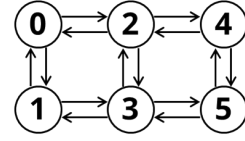


Fig. 1. Small example graph

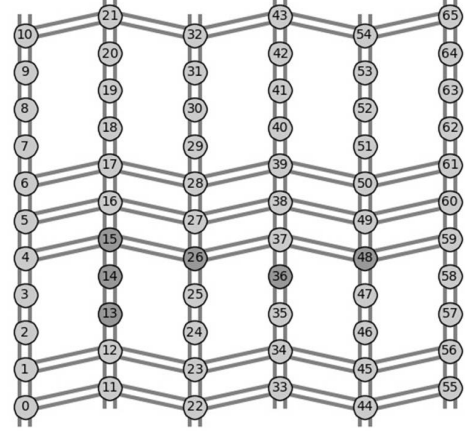


Fig. 2. Large example graph

The KKT conditions for problem (15) are

$$\sum_{i,j} [\mathbf{S}_i]_{l,j} x_{i,j} \leq c_l, \quad \forall l$$

$$x_{i,j} \geq 0, \quad \forall i, j$$

$$\boldsymbol{\lambda} \geq \mathbf{0}$$

$$\boldsymbol{\mu} \geq \mathbf{0}$$

$$\lambda_l \left(\sum_{i,j} [\mathbf{S}_i]_{l,j} x_{i,j} - c_l \right) = 0, \quad \forall l$$

$$\mu_{i,j} x_{i,j} = 0, \quad \forall i, j$$

$$\bar{x}_i = h_i([\mathbf{S}_i^T \boldsymbol{\lambda}]_j - \mu_{i,j}), \quad \forall i, j$$

Now set $\boldsymbol{\rho} = \boldsymbol{\lambda}^*$ and $\sigma_{i,j,k} = \mu_{i,j}^*$ for all i, j, k and let $\{u_{i,j,k}^*\}$ be a solution to (18). Then it can be seen that all KKT conditions (20)–(26) are satisfied, and furthermore, using (19)

$$\sum_{j,k} u_{i,j,k}^* = \sum_k h_{i,k}(f'_i(\bar{x}_i^*)) = h_i(f'_i(\bar{x}_i^*)) = \bar{x}_i^*$$

which ensures $\sum_{j,k} u_{i,j,k}^* = \sum_j x_{i,j}^*$. \square

Note that when $J = 1$, the second equation of problem (18) reduces to (5) which guarantees that the first equation and the non-negativity condition hold.

The subflow allocation problem given by (18) can be decomposed into N parallel problems (one for each class i). Let $\mathbf{u}_{i,k} = [u_{i,1,k}, \dots, u_{i,J,k}]^T$, $\mathbf{u}_i = [\mathbf{u}_{i,1}^T, \dots, \mathbf{u}_{i,K_i}^T]^T$, $\mathbf{x}_i = [x_{i,1}, \dots, x_{i,J}]^T$, and define the matrices $\mathbf{A}_i = \mathbf{1}_{K_i}^T \otimes \mathbf{I}_J$ and $\mathbf{B}_i = \mathbf{I}_{K_i} \otimes \mathbf{1}_J^T$. Then the i th optimal subflow vector \mathbf{u}_i^* solves

$$\begin{bmatrix} \mathbf{A}_i \\ \mathbf{B}_i \end{bmatrix} \mathbf{u}_i = \begin{bmatrix} \mathbf{x}_i^* \\ \mathbf{g}_i \end{bmatrix} \quad (27)$$

$$\mathbf{u}_i \geq \mathbf{0}$$

TABLE I
COMPARISON OF ADMM, CP, AND GRAD. PROJ. FOR SMALL GRAPH EXAMPLE

N	f^*	ADMM			f^*	Gradient Projection			f^*	Chambolle-Pock		
		l_{\max}	n_{iter}	t (sec)		l_{\max}	n_{iter}	t (sec)		l_{\max}	n_{iter}	t (sec)
10	-92.084	10.000	194	0.0193	-92.085	10.000	431	1.1607	-92.084	10.000	74	0.0030
15	-136.800	10.000	207	0.0259	-136.809	10.000	500	1.2534	-136.800	10.000	112	0.0050
20	-182.002	10.000	304	0.0446	-182.002	10.000	590	1.6405	-182.002	10.000	258	0.0130
25	-243.806	10.000	296	0.0508	-243.806	10.000	1288	3.4499	-243.806	10.000	225	0.0130
30	-289.040	10.000	296	0.0574	-289.040	10.000	2029	5.9341	-289.040	10.000	256	0.0170

where $\mathbf{g}_i = [g'_{i,1}(f'_i(\bar{x}_i^*)), \dots, g'_{i,K_i}(f'_i(\bar{x}_i^*))]^T$.

Note that the only component of (27) that depends on the utility functions is \mathbf{g}_i . As an example, for the case of weighted logarithm utilities (see Section II-C1) with $f_{i,k}(u) = w_{i,k} \log u$, we have $\mathbf{g}_i = (\bar{x}_i^*/\bar{w}_i)\mathbf{w}_i$, where $\mathbf{w}_i = [w_{i,1}, \dots, w_{i,K_i}]^T$, and $\bar{w}_i = \sum_k w_{i,k}$.

IV. AGGREGATE DECOMPOSITION WITH ADMM

Here we show that the alternating direction method of multipliers (ADMM) algorithm [18] can inherently decompose problem (1) into an optimization over aggregate flows and N parallel optimizations over the constituent flows. Assume that the utility functions $f_{i,k}$ have domain \mathbb{R}_+ .

A. ADMM Algorithm

To apply ADMM to problem (1) we recast it as

$$\begin{aligned} & \underset{\{u_{i,k} \in \mathbb{R}_+, x_i \in \mathbb{R}, y_l \in \mathbb{R}\}}{\text{minimize}} && \sum_i \sum_k -f_{i,k}(u_{i,k}) + h(\mathbf{y}) \\ & \text{subject to} && \sum_k u_{i,k} = x_i, \quad i = 1, \dots, N \\ & && \mathbf{y} = \mathbf{R}\mathbf{x} \end{aligned} \quad (28)$$

where $\mathbf{x} = [x_1, \dots, x_N]^T$ and $\mathbf{y} = [y_1, \dots, y_L]^T$. The function h indicates that the links are not overloaded. That is $h(\mathbf{y}) = 0$ if $\mathbf{y} \leq \mathbf{c}$, and $+\infty$ otherwise, with $\mathbf{c} = [c_1, \dots, c_L]^T$. The augmented Lagrangian for problem (28) is

$$\begin{aligned} \mathcal{L}_r(\mathbf{u}, \mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}, \boldsymbol{\rho}) &= \sum_i \sum_k -f_{i,k}(u_{i,k}) + h(\mathbf{y}) + \boldsymbol{\lambda}^T(\mathbf{s} - \mathbf{x}) \\ &+ \boldsymbol{\rho}^T(\mathbf{y} - \mathbf{R}\mathbf{x}) + \frac{r}{2}(\|\mathbf{x} - \mathbf{s}\|^2 + \|\mathbf{R}\mathbf{x} - \mathbf{y}\|^2) \end{aligned}$$

where $\mathbf{s} = [\sum_k u_{1,k}, \dots, \sum_k u_{N,k}]^T$. Here $\boldsymbol{\lambda} \in \mathbb{R}^N$ and $\boldsymbol{\rho} \in \mathbb{R}^L$ are the dual variables and r is the penalty parameter. The ADMM method involves repeated minimizations of \mathcal{L}_r over (\mathbf{u}, \mathbf{y}) , and then \mathbf{x} . The minimizer with respect to \mathbf{x} is $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ where $\mathbf{A} = \mathbf{I} + \mathbf{R}^T\mathbf{R}$ and $\mathbf{b} = \mathbf{s} + \mathbf{R}^T\mathbf{y} + (\boldsymbol{\lambda} + \mathbf{R}^T\boldsymbol{\rho})/r$. Minimization of \mathcal{L}_r with respect to \mathbf{y} is decoupled from that of \mathbf{u} and is achieved by a simple projection of $\mathbf{R}\mathbf{x} - \boldsymbol{\rho}/r$ onto the box $\{\mathbf{y} : \mathbf{y} \leq \mathbf{c}\}$. Finally, minimization with respect to \mathbf{u} involves N parallel minimizations of the form

$$\begin{aligned} & \underset{\{u_{i,k} \in \mathbb{R}_+\}}{\text{minimize}} && \sum_k -f_{i,k}(u_{i,k}) + \lambda_i(\sum_k u_{i,k} - x_i) + \\ & && \frac{r}{2}(\sum_k u_{i,k} - x_i)^2. \end{aligned} \quad (29)$$

B. Numerical Examples

Here we apply the ADMM algorithm with the aggregate flow decomposition to a few example cases and compare performance against a gradient projection algorithm and the primal-dual algorithm of Chambolle and Pock [19]. We examine two sample graphs. The first, which we call the *small graph*, shown in Figure 1, has $M = 6$ nodes and $L = 14$ links. The maximum number of source-destination pairs is $N_{\max} = 30$. The second example graph, the *large graph* is shown in Figure 2. This represents the topology of the Iridium low earth orbit satellite constellation [10], and includes $M = 66$ satellites (nodes) and $L = 192$ links (the actual topology changes as satellites enter and exit polar regions). The dark nodes in Figure 2 represent satellites linked to ground stations which connect flows to terrestrial networks. Thus, we assume all flows either originate or terminate at one of these nodes. The resulting maximum number of source-destination pairs is $N_{\max} = 750$. For both example graphs, we set all link capacities to 10 units. Thus $\mathbf{c} = 10 \cdot \mathbf{1}_L$. In all examples, the number of flows in any flow class (source-destination pair) is uniformly distributed between 10 and 20. The total number of flows is thus $15N$ on average. Finally, the path (route) for each source-destination pair is found using Dijkstra's algorithm.

1) *ADMM with Weighted Logarithm Utilities:* We let the utility function for the k th flow of class i be $f_{i,k}(u_{i,k}) = w_{i,k} \log u_{i,k}$ with weights $w_{i,k}$ chosen uniformly from $(0, 1)$ and solve the optimization problem with ADMM. The minimizer of the augmented Lagrangian with respect to the individual flows $\{u_{i,k}\}$ is found by solving (29) for each i which gives

$$u_{i,k}^* = \frac{2w_{i,k}}{\psi_i + \sqrt{\psi_i^2 + 4r\bar{w}_i}} > 0,$$

where $\psi_i = \lambda_i - rx_i$ and $\bar{w}_i = \sum_k w_{i,k}$. The ADMM iteration is then given by

$$\begin{aligned} \psi_i^{(n+1)} &= \lambda_i^{(n)} - rx_i^{(n)} \\ \mathbf{u}_i^{(n+1)} &= 2\mathbf{w}_i[\psi_i^{(n+1)} + ((\psi_i^{(n+1)})^2 + 4r\bar{w}_i)^{1/2}]^{-1} \\ \mathbf{y}^{(n+1)} &= [\mathbf{R}\mathbf{x}^{(n)} - \boldsymbol{\rho}^{(n)}/r]^+ \\ \mathbf{x}^{(n+1)} &= \mathbf{A}^{-1}(\mathbf{s}^{(n+1)} + \boldsymbol{\lambda}^{(n)}/r + \mathbf{R}^T(\mathbf{y}^{(n+1)} + \boldsymbol{\rho}^{(n)}/r)) \\ \boldsymbol{\lambda}^{(n+1)} &= \boldsymbol{\lambda}^{(n)} + r(\mathbf{s}^{(n+1)} - \mathbf{x}^{(n+1)}) \\ \boldsymbol{\rho}^{(n+1)} &= \boldsymbol{\rho}^{(n)} + r(\mathbf{y}^{(n+1)} - \mathbf{R}\mathbf{x}^{(n+1)}) \end{aligned}$$

where $[\cdot]^+$ represents projection onto the box $\{\mathbf{y} : \mathbf{y} \leq \mathbf{c}\}$, and $\mathbf{s}^{(n+1)} = [\sum_k u_{1,k}^{(n+1)}, \dots, \sum_k u_{N,k}^{(n+1)}]^T$.

TABLE II
PARAMETERS FOR SMALL GRAPH EXAMPLE

N	ADMM		Grad Proj α	Chambolle-Pock		
	r	pct		σ	τ	θ
10	20	10^{-4}	1.07×10^{-2}	1.0	0.020	1.0
15	20	10^{-4}	1.07×10^{-2}	1.0	0.015	1.0
20	20	10^{-4}	1.31×10^{-2}	1.0	0.015	1.0
25	20	10^{-4}	6.17×10^{-3}	1.0	0.015	1.0
30	20	10^{-4}	6.19×10^{-3}	1.0	0.013	1.0

2) *Gradient Projection Optimizer*: We compare the ADMM algorithm with a simple gradient projection optimizer. The gradient projection optimizer utilizes Theorem 1 with aggregate utilities $f_i(x_i) = \bar{w}_i \log x_i$ and optimal subflow rates $u_{i,k}^* = w_{i,k} x_i^* / \bar{w}_i$ (see Section II-C1). The update rule for the aggregate problem is

$$\begin{aligned} \nabla f(\mathbf{x}^{(n)})_i &= \bar{w}_i / x_i^{(n)} \\ \mathbf{x}^{(n+1)} &= P_{\mathbf{R}}(\mathbf{x}^{(n)} + \alpha \nabla f(\mathbf{x}^{(n)})) \end{aligned}$$

where $\alpha > 0$ is a step size and $P_{\mathbf{R}}$ is the function which projects onto the routing polytope $\{\mathbf{x} : \mathbf{R}\mathbf{x} \leq \mathbf{c}, \mathbf{x} \geq \mathbf{0}\}$. In all examples that follow, $P_{\mathbf{R}}$, which solves a QP, is implemented using the CVXOPT QP solver [20].

3) *Chambolle-Pock Optimizer*: Problem (1) can be solved with the Chambolle-Pock (CP) algorithm by writing it as

$$\underset{\{\mathbf{u} \in \mathbb{R}_{++}^K\}}{\text{minimize}} \quad - \sum_i \sum_k f_{i,k}(u_{i,k}) + g(\mathbf{Q}\mathbf{u})$$

where \mathbf{u} is the concatenation of the N subflow rate vectors $\{\mathbf{u}_i\}$ and g is the indicator function of the box $\{\mathbf{y} : \mathbf{y} \leq \mathbf{c}\}$. The $L \times K$ matrix \mathbf{Q} is defined by

$$\mathbf{Q} = [\underbrace{\mathbf{r}_1, \dots, \mathbf{r}_1}_{K_1}, \dots, \underbrace{\mathbf{r}_N, \dots, \mathbf{r}_N}_{K_N}]$$

The algorithm requires evaluation of the proximal operators [12] of σg^* and τf where $f(\mathbf{u}) = -\sum_{i,k} f_{i,k}(u_{i,k})$, g^* is the convex conjugate of g , and σ and τ are positive constants. Using Moreau's theorem [12, Theorem 31.5] we get $\text{prox}_{\sigma g^*}(\mathbf{z}) = \mathbf{z} - \sigma[\mathbf{z}/\sigma]^+$ (again $[\cdot]^+$ signifies projection onto $\{\mathbf{y} : \mathbf{y} \leq \mathbf{c}\}$). The proximal operator of τf is

$$\text{prox}_{\tau f}(\mathbf{z})_{i,k} = \frac{z_{i,k} + \sqrt{z_{i,k}^2 + 4\tau w_{i,k}}}{2}.$$

The algorithm consists of the following iteration

$$\begin{aligned} \mathbf{y}^{(n+1)} &= \text{prox}_{\sigma g^*}(\mathbf{y}^{(n)} + \sigma \mathbf{Q}\mathbf{v}^{(n)}) \\ \mathbf{u}^{(n+1)} &= \text{prox}_{\tau f}(\mathbf{u}^{(n)} - \tau \mathbf{Q}^T \mathbf{y}^{(n+1)}) \\ \mathbf{v}^{(n+1)} &= \mathbf{u}^{(n+1)} + \theta(\mathbf{u}^{(n+1)} - \mathbf{u}^{(n)}) \end{aligned}$$

where $\theta \in [0, 1]$.

4) *Algorithm Comparison with Small Graph*: Each iteration of ADMM contains three sparse (0-1 matrix)-vector multiplies with \mathbf{R} and \mathbf{R}^T and one $N \times N$ set of linear equations with the same coefficient matrix \mathbf{A} . The CP iterations contain two multiplications with the sparse 0-1 matrices \mathbf{Q} and \mathbf{Q}^T .

Finally, each iteration of the gradient projection algorithm solves a QP (with sparse constraint matrix $\mathbf{G} = [\mathbf{R}^T, -\mathbf{I}_N]^T$). Thus, the gradient algorithm has the highest per-iteration cost, followed by ADMM and CP.

The algorithms' performances are summarized in Table I for various numbers of source-destination pairs N . For each algorithm, the converged objective value f^* is shown, along with the maximum link load l_{\max} , the number of iterations n_{iter} , and the optimization time t . In Table II the algorithm parameters are listed, including the ADMM penalty parameter r , the gradient projection step-size α , and the percent threshold (pct). This value is used as the stopping criterion for ADMM (i.e., when the augmented Lagrangian changes by less than pct percent, stop). Also shown are the three CP parameters σ , τ , and θ . The gradient projection step-sizes and CP parameters are individually tuned for fastest convergence. The optimization times are averaged over 10 runs (with identical random number generator seeds). All simulations were performed using Python/Numpy, and the projection step in the gradient projection algorithm uses the CVXOPT QP solver (which in turn uses the CHOLMOD sparse Cholesky solver). The optimization time of ADMM and CP is plotted versus N in Figure 3. From Table I, the number of CP iterations required for this example is consistently less than the number of ADMM iterations. As it has a lower per-iteration cost, the convergence time of CP is lower. The gradient projection algorithm has the highest per-iteration cost as well as the largest number of iterations, and thus converges slowest. Note that, without the aggregate flow decomposition (Theorem 1), the gradient projection optimizer would be far slower.

5) *Algorithm Comparison with Large Graph*: Next, we repeat the experiment using the large graph. Table III shows the results along with the algorithm parameters. The gradient projection algorithm has been omitted as its convergence times are far greater than ADMM and CP. The optimization times for the ADMM and CP algorithms are plotted in Figure 4. In this example, as N increases, the number of ADMM iterations grows slower than the number of CP iterations. Thus, although the CP per-iteration cost is lower, the larger number of iterations for large N renders CP slower than ADMM.

V. CONCLUSION

We have shown that for many types of utilities, the solution to a K -flow NUM problem can be found by solving a simpler N -variable problem. This principle holds for both single-path and multipath NUM problems. The results of this paper have applicability for software-defined networks in which a controller must solve the global NUM problem. These results can also be beneficial for networks consisting of several hub-spoke clusters. For example, with N clusters and K_i sources in the i th cluster, the problem (2) can be substituted for problem (1). This simpler problem could then be solved in a distributed manner by the hub nodes, which would in turn allocate subflow rates to the spoke nodes.

TABLE III
COMPARISON OF ADMM AND CP FOR LARGE GRAPH EXAMPLE

N	ADMM						Chambolle-Pock						
	r	pct	f^*	l_{\max}	n_{iter}	t (sec)	σ	τ	θ	f^*	l_{\max}	n_{iter}	t (sec)
50	40	10^{-4}	-1326.781	10.000	100	0.0358	10.0	3.0×10^{-4}	1.0	-1326.780	10.000	70	0.0120
75	40	10^{-4}	-2002.522	10.001	162	0.0777	10.0	2.0×10^{-4}	1.0	-2002.522	10.000	314	0.0760
100	40	10^{-4}	-2589.978	10.000	179	0.1093	10.0	5.0×10^{-4}	0.1	-2589.978	10.000	745	0.2280
125	40	10^{-4}	-3333.174	10.007	208	0.1529	10.0	4.9×10^{-4}	0.1	-3333.174	10.000	843	0.3119

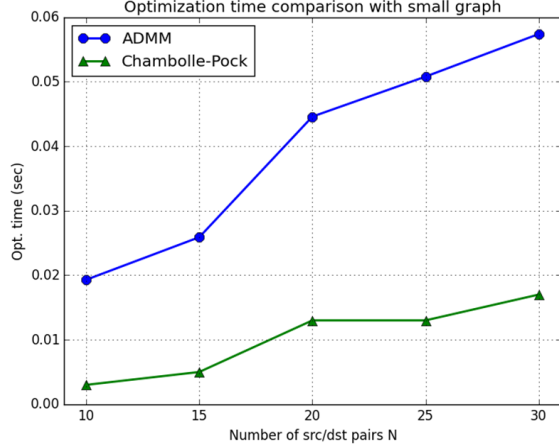


Fig. 3. Optimization time comparison with small graph example

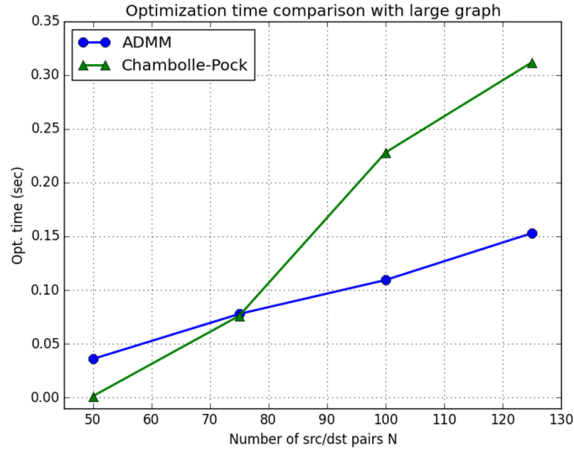


Fig. 4. Optimization time comparison with large graph example

VI. ACKNOWLEDGEMENTS

The first author wishes to thank Dr. Joseph Yadegar from UtopiaCompression Corporation and Dr. You Lu of Google for insightful conversations regarding this work. This research was partly supported by the United States Air Force under contract number FA9453-14-C-0060. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the United States Air Force.

REFERENCES

- [1] F. P. Kelly, A. K. Maulloo, and D. K. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, pp. 237–252, 1998.
- [2] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, Aug. 2006.
- [3] M. Chiang, S. H. Low, R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition," *Proceedings of IEEE*, 2006.
- [4] S. H. Low, "A duality model of TCP and queue management algorithms," *Networking, IEEE/ACM Transactions on*, vol. 11, no. 4, pp. 525–536, 2003.
- [5] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, "FAST TCP: motivation, architecture, algorithms, performance," *IEEE/ACM Transactions on Networking (ToN)*, vol. 14, no. 6, pp. 1246–1259, 2006.
- [6] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.
- [7] N. McKeown, "Software-defined networking," *INFOCOM keynote talk*, vol. 17, no. 2, pp. 30–32, 2009.
- [8] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*. ACM, 2010, pp. 267–280.
- [9] A. Donner, M. Berioli, and M. Werner, "MPLS-based satellite constellation networks," *Selected Areas in Communications, IEEE Journal on*, vol. 22, no. 3, pp. 438–448, 2004.
- [10] S. R. Pratt, R. A. Raines, C. Fossa, and M. A. Temple, "An operational and performance overview of the IRIDIUM low earth orbit satellite system," *Communications Surveys & Tutorials, IEEE*, vol. 2, no. 2, pp. 2–10, 1999.
- [11] D. Bertsekas and R. Gallager, *Data Networks*. Upper Saddle River, NJ: Prentice Hall, 1992.
- [12] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1970.
- [13] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [14] R. Srikant and L. Ying, *Communication Networks: An Optimization, Control, and Stochastic Networks Perspective*. Cambridge University Press, 2013.
- [15] F. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, no. 1, pp. 33–37, 1997.
- [16] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.
- [17] R. T. Rockafellar, *Network Flows and Monotropic Optimization*. Wiley-Interscience, 1984.
- [18] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [19] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.
- [20] M. Andersen, J. Dahl, and L. Vandenberghe, *CVXOPT: A Python Package for Convex Optimization*, www.cvxopt.org, 2015.