Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Dynamic Network Utility Maximization with Delivery Contracts ⋆

Nikolaos Trichakis * **Argyrios Zymnis** ** **Stephen Boyd** **

* *Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02139, USA (e-mail: nitric@mit.edu)*
** *Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA (e-mail: {azymnis,boyd}@stanford.edu)*

Abstract: We consider a multi-period variation on the network utility maximization problem that includes delivery constraints. We allow the flow utilities, link capacities and routing matrices to vary over time, and we introduce the concept of delivery contracts, which couple the flow rates across time. We describe a distributed algorithm, based on dual decomposition, that solves this problem when all data is known ahead of time. We briefly describe a heuristic, based on model predictive control, for approximately solving a variation on the problem, in which the data are not known ahead of time. The formulation and algorithms are illustrated with numerical examples.

Keywords: control of networks, control under communication constraints, distributed control

1. INTRODUCTION

A network with m links supports n flows, that vary over time t, which takes (discrete) values $t = 1, \ldots, T$. At time t, each flow is associated with a fixed route in the network, *i.e.*, a subset of the network links. We describe these (possibly time-varying) routes using the *routing* or *link-route* matrix $R_t \in \mathbf{R}^{m\times n}$, $t = 1, \ldots, T$, defined as

$$(R_t)_{ij} = \begin{cases} 1 & \text{route of flow } j \text{ passes over link } i \text{ at time } t \\ 0 & \text{otherwise.} \end{cases}$$

In the most common case, the route of a flow will be a path, from a source node to a destination node. But our definition of a route as *any* subset of links is more general, and can be used to model, for example, a multicast flow (where the route links form a tree).

At time t, flow j has a nonnegative flow *rate*, which we denote f_{jt}. Each flow rate f_{jt} has a maximum permissible value given by $f_{jt}^{\max}$. We define $F \in \mathbf{R}^{n\times T}$ (with entries f_{jt}) as the *rate matrix*. We also define $F^{\max}$ (with entries $f_{jt}^{\max}$) as the *rate constraint matrix*. The tth column of F, denoted $f_t \in \mathbf{R}^n$, gives the vector of all flow rates at time t, *i.e.*, a snapshot of the flow distribution in the network at time t. Similarly, the jth row of F, which we will denote $f_j \in \mathbf{R}^T$, gives the *rate schedule* for flow j, *i.e.*, the jth flow rate for $t = 1, \ldots, T$. We distinguish between f_t (a flow snapshot) and f_j (a flow schedule) by their indices.

With the flow rate f_{jt} we associate a strictly concave, increasing, differentiable utility function U_{jt}, with $\mathbf{dom}\, U_{jt} \supseteq (0, f_{jt}^{\max}]$. The utility derived by flow rate f_{jt} is $U_{jt}(f_{jt})$; the total utility, over all flows and over time, is

$$U(F) = \sum_{j=1}^{n}\sum_{t=1}^{T} U_{jt}(f_{jt}).$$

One common choice of utility function is $U_{jt}(x) = \log x$; but we allow here the possibility that the utility functions differ for different flows, and can be time-varying as well.

The total traffic on link i at time t is the sum of the flow rates at time t, over all flows whose route includes link i. The link traffic vector, at time t, is given by $R_t f_t \in \mathbf{R}^m$. Each link in the network has a (positive) *capacity*. Let $c_t \in \mathbf{R}^m$ be the vector of the link capacities at time t. The traffic on a link cannot exceed its capacity, *i.e.*, we have

$$R_t f_t \leq c_t, \quad t = 1, \ldots, T,$$

where $\leq$ denotes componentwise inequality.

So far the problem setup is not coupled across time. The utility function U is separable across t, and the constraints for different values of t are independent (*i.e.*, involve different variables). It follows that we can maximize the utility, subject to the link capacity constraints, by solving T separate problems, once for each time $t = 1, \ldots, T$. At this point, however, we introduce some constraints that couple the flow rates at different times.

A *delivery contract* is the requirement that the total of some particular flow j, over some particular time interval $[t^{\text{init}}, t^{\text{fin}}]$, should meet or exceed some specified minimum quantity q:

$$\sum_{t=t^{\text{init}}}^{t^{\text{fin}}} f_{jt} \geq q.$$

Suppose flow j has k_j delivery contracts, with $q_j \in \mathbf{R}^{k_j}$ the vector of the associated contract quantity amounts. The contract constraints for flow j can be compactly written using the *contract indicator matrix* $C_j \in \mathbf{R}^{k_j\times T}$, defined as

⋆ This material is based on work supported by JPL award I291856, NSF award 0529426, DARPA award N66001-06-C-2021, NASA award NNX07AEIIA, and AFOSR award FA9550-06-1-0312.

$$(C_j)_{kt} = \begin{cases} 1 & k\text{th contract of flow } j \text{ is active at time step } t \\ 0 & \text{otherwise.} \end{cases}$$

(Here 'is active' means that t lies in the time interval of the contract.) We can express the delivery contract requirements for flow j as the vector inequality $C_j f_j \geq q_j$. (We have described contracts as involving a sum of flow rates over an interval in time, but everything in what follows works when contracts are any linear inequality on the flow rates across time.) The constraints that all contracts are met can be expressed as

$$C_j f_j \geq q_j, \quad j = 1, \ldots, n.$$

Now we can define the problem of network utility maximization, with delivery contracts (NUMDC). The goal is to choose the flow rates, for all time steps, in order to maximize total utility, subject to the flow rate, link capacity, and delivery contract constraints:

$$\begin{array}{lll} \text{maximize} & U(F) & \\ \text{subject to} & R_t f_t \leq c_t, & t = 1, \ldots, T \\ & C_j f_j \geq q_j, & j = 1, \ldots, n \\ & 0 \leq F \leq F^{\max}. & \end{array} \tag{1}$$

The optimization variable in this problem is rate matrix F. The problem data are the rate constraint matrix $F^{\max}$, the utility functions U_{jt}, the route matrices R_t, the link capacities c_t, the delivery contract matrices C_j, and the delivery contract quantities q_j. The NUMDC problem is a convex optimization problem, and has at most one solution, since the objective is strictly concave. It can, however, be infeasible.

The constraints on the variable matrix F have an interesting structure. The link capacity constraints impose constraints on each of the t *columns* of F, separately. The delivery contracts impose constraints on the *rows* of F, separately. With delivery contracts, the problem cannot be split into separate subproblems; the choice of all flow rates, over all times, must be coordinated.

There are a number of ways to solve problem (1), such as interior-point methods (Boyd and Vandenberghe [2004], Nocedal and Wright [1999], Wright [1997]), which are efficient, but centralized algorithms. In this paper we propose a method based on dual decomposition, which is decentralized, and so scales to very large problem sizes.

Decomposition is the standard method used to solve a large problem (or in this case its dual) by breaking it up into a set of smaller subproblems that can be solved locally. In some cases this leads to decentralized algorithms. Decomposition has a long history in optimization, going back to the Dantzig-Wolfe decomposition (Dantzig and Wolfe [1960]) and the Benders decomposition (Benders [1962]). A more recent reference on decomposition methods is (Bertsekas [1999]).

We combine the dual decomposition approach with the projected subgradient method, which is a simple algorithm to minimize a nondifferentiable convex function on a convex set. Some classic references on subgradient methods are (Shor [1985], Polyak [1987], Shor [1998]). For more recent work on subgradient methods, we refer the reader to (Nedić and Bertsekas [2001], Nedić and Ozdaglar [2007]) as well as the thesis (Nedić [2002]).

Network utility maximization (NUM), *i.e.*, the problem (1) for a single time step ($T = 1$), and with no contract constraints, has been extensively analyzed. In the seminal paper (Kelly et al. [1997]), the authors propose a dual decomposition solution to the NUM problem and interpret this as a distributed algorithm, whereby each link sets a price for flow that passes through it, and each flow adjusts its rate to locally maximize its utility. This work has led to a large body of research in decomposition methods in the context of networking problems. We refer the reader to (Low and Lapsley [1999], Chiang et al. [2007]), as well as the books (Bertsekas [1998], Srikant [2004]).

In §2, we describe a decentralized algorithm for solving the NUMDC problem based on dual decomposition, establish its convergence, and interpret the algorithm in terms of contract pricing. We give a numerical example to illustrate the method in §3. In §4 we consider the much harder problem that arises when the problem data (such as link capacities) are not known ahead of time, and describe a simple heuristic, model predictive control, that can be used to get a good, if not optimal, choice of rates even when future problem data are not fully known. We illustrate this method with the same example used to illustrate the basic NUMDC problem.

2. SOLUTION VIA DUAL DECOMPOSITION

2.1 Dual Problem

In this section we derive a dual of problem (1). Let $\lambda_t \in \mathbf{R}^m_+$ be the dual variable associated with the capacity constraints at time t, and $\mu_j \in \mathbf{R}^{k_j}_+$ the dual variable associated with the contract constraints for flow j. The partial Lagrangian (see, *e.g.*, [Boyd and Vandenberghe, 2004, Ch.5]) of problem (1) is

$$L(F, \lambda, \mu) = U(F) - \sum_{t=1}^{T} \lambda_t^T (R_t f_t - c_t) + \sum_{j=1}^{n} \mu_j^T (C_j f_j - q_j),$$

where $\lambda = (\lambda_1, \ldots, \lambda_T)$ and $\mu = (\mu_1, \ldots, \mu_n)$.

The dual function of problem (1) is

$$\begin{aligned} g(\lambda, \mu) &= \sup_{0 \leq F \leq F^{\max}} L(F, \lambda, \mu) \\ &= \sum_{t=1}^{T} \lambda_t^T c_t - \sum_{j=1}^{n} \mu_j^T q_j + \sum_{j=1}^{n} \sum_{t=1}^{T} (-U_{jt})^*(p_{jt}), \end{aligned}$$

where $p_{jt} = (R_t^T \lambda_t)_j - (C_j^T \mu_j)_t$, with $(R_t^T \lambda_t)_j$ and $(C_j^T \mu_j)_t$ denoting the jth and tth elements of vectors $R_t^T \lambda_t$ and $C_j^T \mu_j$, respectively. We define $P \in \mathbf{R}^{n \times T}$ to be the price matrix, *i.e.*, the matrix with elements p_{jt}. The function $(-U_{jt})^*$ is the conjugate of the negative utility function U_{jt} (see [Boyd and Vandenberghe, 2004§3.3]),

$$(-U_{jt})^*(y) = \sup_{0 \leq z \leq f_{jt}^{\max}} (U_{jt}(z) - yz).$$

For future reference we define

$$\begin{aligned} f_{jt}^{\star}(y) &= \operatorname*{argmax}_{0 \leq z \leq f_{jt}^{\max}} (U_{jt}(z) - yz) \\ &= \begin{cases} (U'_{jt})^{-1}(y), & (U'_{jt})^{-1}(y) \in (0, f_{jt}^{\max}] \\ f_{jt}^{\max}, & (U'_{jt})^{-1}(y) \notin (0, f_{jt}^{\max}]. \end{cases} \end{aligned}$$

(The argmax is unique, since U_{jt} is assumed to be strictly concave.) Using this definition we have that

$$(-U_{jt})^*(y) = U_{jt}(f_{jt}^\star(y)) - yf_{jt}^\star(y).$$

The functions $(-U_{jt})^*$ are convex, since by definition they are pointwise suprema of affine functions (see [Boyd and Vandenberghe, 2004, Chap. 3]). The dual function $g(\lambda, \mu)$ is also convex since it is a sum of convex functions.

The dual of problem (1) is

$$\begin{array}{ll} \text{minimize} & g(\lambda, \mu) \\ \text{subject to} & \lambda \geq 0, \quad \mu \geq 0. \end{array} \tag{2}$$

This is a convex optimization problem, with variables λ and μ. Any feasible point for this dual gives an upper bound on the optimal value of the (primal) NUMDC problem: for any $\lambda \geq 0$, $\mu \geq 0$ and any feasible F we have

$$g(\lambda, \mu) \geq U(F).$$

This implies that if the dual NUMDC problem is unbounded below, the primal NUMDC problem is infeasible. Conversely, if the dual problem is bounded below, the primal problem is feasible.

We can reconstruct $F^\star$, the optimal solution of the NUMDC problem (1) from $(\lambda^\star, \mu^\star)$, an optimal solution of the dual NUMDC problem (2) as follows:

$$f_{jt}^\star = f_{jt}^\star(p_{jt}^\star) = \operatorname*{argmax}_{0 \leq z \leq f_{jt}^{\max}} \left(U_{jt}(z) - p_{jt}^\star z \right).$$

2.2 Dual Decomposition

In this section we describe a simple distributed algorithm for solving problem (2). We start with any nonnegative $\lambda_1, \ldots, \lambda_T$, and any nonnegative $\mu_1, \ldots, \mu_n$, and repeatedly carry out the update

$$\begin{array}{lll} f_{jt} := f_{jt}^\star(p_{jt}) = \operatorname*{argmax}_{0 \leq z \leq f_{jt}^{\max}} \left(U_{jt}(z) - z p_{jt} \right), & t = 1, \ldots, T \\ & j = 1, \ldots, n \\ \lambda_t := (\lambda_t - \alpha (c_t - R_t f_t))_+, & t = 1, \ldots, T \\ \mu_j := (\mu_j - \alpha (C_j f_j - q_j))_+, & j = 1, \ldots, n, \end{array}$$

where $\alpha > 0$ is the step size, an algorithm parameter, and $(z)_+$ denotes the positive part of z, *i.e.*, $\max\{0, z\}$. The terms $c_t - R_t f_t$, $C_j f_j - q_j$ appearing in the updates are the *slacks* in the link capacity and contract constraints respectively (and can have negative terms during the algorithm execution). If we stack up these terms, we form exactly the gradient of the dual objective function.

We will later show that for α small enough, this algorithm will converge to a solution of the NUMDC problem, as long as the problem is feasible. By this we mean that

$$\begin{array}{lll} f_{jt} \to f_{jt}^\star, & j = 1, \ldots, n, & t = 1, \ldots, T \\ \lambda_t \to \lambda_t^\star, & t = 1, \ldots, T & \\ \mu_j \to \mu_j^\star, & j = 1, \ldots, n, & \end{array}$$

where $F^\star$ is the solution of the primal NUMDC problem and $(\lambda^\star, \mu^\star)$ is a solution to the dual NUMDC problem. At each algorithm iteration, we have a dual feasible point (λ, μ); but F is generally not feasible. (Indeed, if F is feasible, it must be optimal.) Thus, at each iteration we have an upper bound on the optimal value of the NUMDC (1), obtained by evaluating the dual objective function.

The algorithm above is decentralized. We can interpret λ_t as the vector of *link prices* at time t and μ_j as the vector of *contract subsidies* for flow j. All the updates are carried out based on local information. Each flow updates its rates based on information obtained from the links it passes over, and its contracts; each link price vector is updated based only on the schedules of the flows that pass over it. The contract subsidies are updated (by each flow, separately) based on the slack in the contract constraints.

The algorithm also has a natural economic interpretation. We can imagine that at each time t, flow j is *charged* a price for utilizing each of its links. The total of these prices is $(R_t^T \lambda_t)_j$; this price multiplied by the flow rate (at time t) gives a total *link usage charge*. At each time step t, the flow receives a *subsidy* for each of its contracts that is active, given by the associated value μ_{jt}. The sum of these subsidies is given by $(C_j^T \mu_j)_t$. This subsidy rate, multiplied by the flow, gives the total *contract subsidy*. The net price per unit rate, from link usage and contract subsidies, is thus given by p_{jt}. The total charge, $p_{jt} f_{jt}$, is subtracted from the utility, and the maximum net utility flow rate is chosen.

The links update their usage prices for each time t, depending on their capacity margin $c_t - R_t f_t$; if the margin is positive, the link price is decreased (but not below zero); if it is negative, which means the link capacity constraint is violated, the link price is increased. In a similar way, flow j updates its contract subsidies based on the contract delivery margin $C_j f_j - q_j$.

2.3 Convergence

In this section we establish convergence of the algorithm. A standard result is that the dual projected gradient algorithm converges for $0 < \alpha < 2/K$, where K is a Lipschitz constant for the dual objective function (see, *e.g.*, ([Polyak, 1987§7.2.1]) or ([Shor, 1985§3.4])). So in this section we derive a valid Lipschitz constant for the dual objective function.

We define a single dual variable

$$\nu = (\lambda_1, \ldots, \lambda_T, \mu_1, \ldots, \mu_n).$$

We have

$$\begin{aligned} \nabla_{\lambda_t} g(\nu) &= c_t - R_t f_t^\star(\nu), \; t = 1, \ldots, T \\ \nabla_{\mu_j} g(\nu) &= C_j f_j^\star(\nu) - q_j, \; j = 1, \ldots, n. \end{aligned}$$

We define

$$s_R = \max_t \|R_t\|, \quad s_C = \max_j \|C_j\|,$$

where $\|\cdot\|$ denotes the usual matrix norm, *i.e.*, the maximum singular value. By construction of ∇g we have

$$\|\nabla g(\nu_1) - \nabla g(\nu_2)\|_2 \leq (s_R + s_C) \|F^\star(\nu_1) - F^\star(\nu_2)\|_F, \tag{3}$$

where $\|\cdot\|_F$ denotes the matrix Frobenius norm. Let P_1 and P_2 be the price matrices corresponding to ν_1 and ν_2. Define

$$p_{jt}^{\text{crit}} = U_{jt}'(f_{jt}^{\max}), \quad V_{jt}(p) = (U_{jt}')^{-1}(p).$$

We have

$$\|F^\star(\nu_1) - F^\star(\nu_2)\|_F \leq \max_{j,t} \left| V_{jt}'(p_{jt}^{\text{crit}}) \right| \|P_1 - P_2\|_F. \tag{4}$$

Finally,

$$\|P_1 - P_2\|_F \leq 2 \max(s_R, s_C) \|\nu_1 - \nu_2\|_2. \tag{5}$$

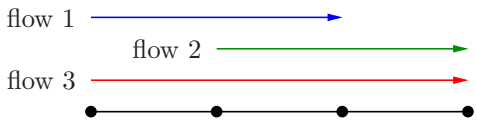

Fig. 1. Network topology.

Combining inequalities (3), (4), and (5) we get

$$\|\nabla g(\nu_1) - \nabla g(\nu_2)\|_2 \leq 2(s_R + s_C)\max(s_R, s_C)\max_{j,t}\left|V'_{jt}(p^{\text{crit}}_{jt})\right| \|\nu_1 - \nu_2\|_2. \quad (6)$$

Thus a Lipschitz constant for ∇g is

$$K = 2(s_R + s_C)\max(s_R, s_C)\max_{j,t}\left|V'_{jt}(p^{\text{crit}}_{jt})\right|. \quad (7)$$

3. NUMERICAL EXAMPLE

In this section we give a simple numerical example to illustrate the NUMDC problem and the distributed dual decomposition algorithm. Our example has $m = 3$ links and $n = 3$ flows, with time horizon $T = 10$. The routes do not vary with time and are shown in figure 1; these correspond to routing matrices

$$R_t = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad t = 1, \ldots, 10.$$

The utility functions are logarithmic: $U_{jt} = \log f_{jt}$ for all j and t. The link capacities c_{jt} are chosen randomly, from a uniform distribution on $[4, 6]$ for links 1 and 3 and $[4, 10]$ for link 2. We set $f^{\max}_{jt} = 4.5$ for all j and t.

Our example has four delivery contracts. Flow 1 must deliver an average rate of at least 4 (per time step) in the period $[1, 3]$ and an average rate of at least $10/3$ in the period $[6, 8]$. Flow 2 must deliver an average rate of at least 3 over the period $[3, 6]$. Flow 3 must deliver an average rate of 1.5 over the period $[2, 10]$. The associated contract matrices and quantities are thus

$$C_1 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}, \quad q_1 = \begin{bmatrix} 12 \\ 10 \end{bmatrix},$$
$$C_2 = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad q_2 = 12,$$
$$C_3 = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad q_3 = 12.$$

For this example we found a Lipschitz constant $K = 600$ for ∇g using (7), which implies that our proposed algorithm will converge as long as $0 < \alpha < 0.0033$. Numerical experiments suggest that the algorithm converges for $\alpha \leq 0.022$, and diverges for $\alpha \geq 0.025$. Figures 2 and 3 show the convergence of the algorithm, started with $\lambda_t = 0$ and $\mu_j = 0$, with step size $\alpha = 0.01$. Figure 2 shows the dual objective value (which is an upper bound on the optimal objective value) versus iteration, and the optimal value. Figure 3 shows the maximum link capacity and contract violations versus iteration.

The optimal flow rates are shown in figure 4. Each of the 4 delivery contract periods is depicted graphically as a shaded area. We can see that the flow rates generally increase during their contract periods, as we would expect, and are generally lower outside contract periods (to make room for other flows with contracts to meet). Figure 5 shows a set of optimal prices p_{jt}. We can see that the price generally drops when a contract is in force, due to the contract subsidy, in order to encourage increased flow.

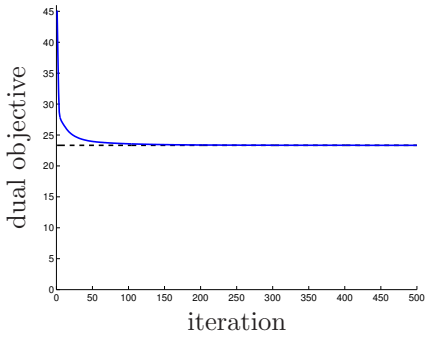

Fig. 2. Dual objective value versus iteration. The dashed line shows the optimal value.

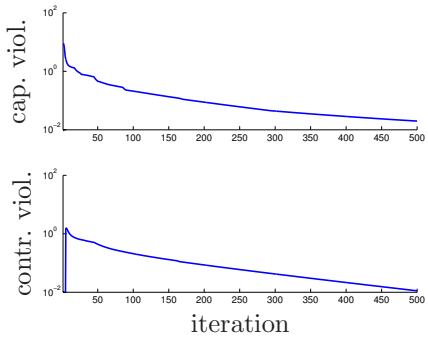

Fig. 3. Maximum link capacity violation (top) and contract violation (bottom), versus iteration.

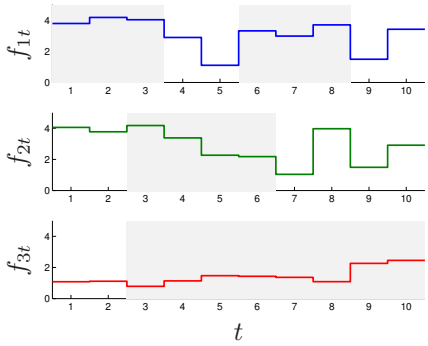

Fig. 4. Optimal flow rates. The delivery contract periods are shown as the shaded areas.

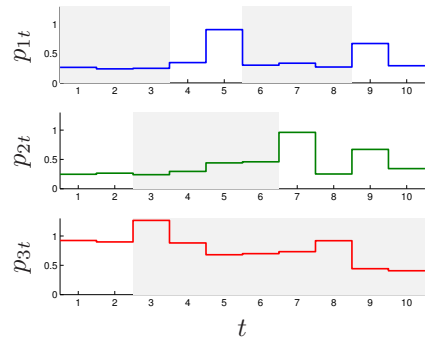

Fig. 5. Optimal prices. The delivery contract periods are shown shaded.

Figure 6 shows the total traffic and capacity for each link. Figure 7 shows a set of optimal link prices λ_{it}. These prices are zero whenever a link operates under full capacity.

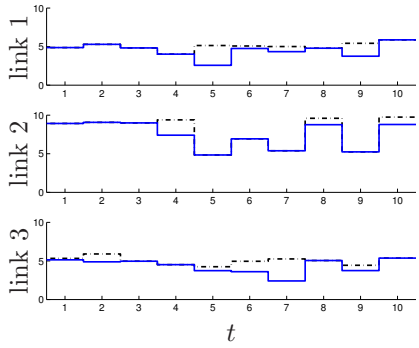

Fig. 6. Link capacity (dashdot) and total traffic (solid).

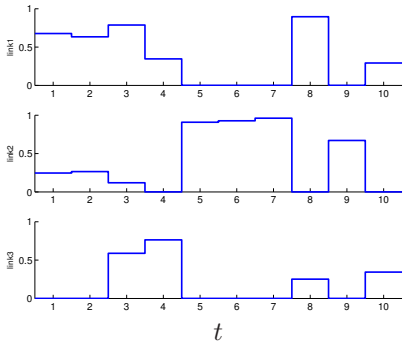

Fig. 7. Optimal link prices.

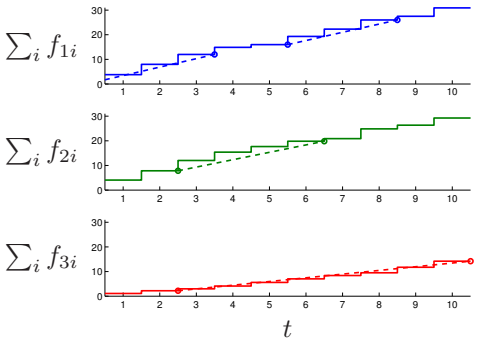

Fig. 8. Cumulative rates, with delivery contracts shown as dashed line segments.

Figure 8 shows the cumulative rate for each flow versus time, so the total rate over a contract period is given by the vertical increase in the curve over the period. The delivery contracts are shown as tilted line segments, with horizontal span showing the delivery period, and height showing required delivery quantity. The delivery contract requires that the cumulative rate lie above the righthand endpoint of the line segment. In this case all 4 delivery contracts are tight. Optimal contract subsidy prices are

$$\mu_1 = \begin{bmatrix} 0.63 \\ 0.54 \end{bmatrix}, \quad \mu_2 = 0.39, \quad \mu_3 = 0.17.$$

4. STOCHASTIC DYNAMIC NUM

In this section we describe an extension to the NUMDC problem, where the problem data is not fully known ahead of time. As above we assume that the flow utility functions and upper bounds, the routing matrices, and the contracts are known for all time steps. The link capacities, however, are random, and revealed only at each time step; future link capacities are not known. We impose a causality constraint: the flow rates at time t must be a function of the link capacities up to time t. Finding the flow rate policy that maximizes expected utility, subject to the rate, contract, and causality constraints is a convex stochastic control problem (see, *e.g.*, (Bertsekas and Shreve [1996])). It can be solved in principle, for example by solving the Bellman equation for the optimal cost-to-go, but this is practical only for simple and small problems.

We instead consider a heuristic flow policy, based on model predictive control (MPC) (Maciejowski [2002], Camacho and Bordons [2004]). To compute the flow rate at time τ we proceed as follows. Let the flow rates up to time $\tau-1$ (which have already been decided, and so are fixed) be $\bar{f}_1, \ldots, \bar{f}_{\tau-1}$. We know $c_1, \ldots, c_\tau$, but we do not know $c_{\tau+1}, \ldots, c_T$. Define

$$\hat{c}(t|\tau) = \mathbf{E}[c(t)|c(1), \ldots, c(\tau)], \quad t = \tau+1, \ldots, T.$$

The vector $\hat{c}(t|\tau)$ is the expected value of the link capacities, given the information available at time τ. We solve the following optimization problem:

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{t=\tau}^{T} \sum_j U_{jt}(f_{jt}) \\ \text{subject to} & R_\tau f_\tau \le c_\tau \\ & R_t f_t \le \hat{c}(t|\tau), \quad t = \tau+1, \ldots, T \\ & C_j f_j \ge q_j, \quad j = 1, \ldots, n \\ & 0 \le f_{jt} \le f_{jt}^{\max}, \; t = \tau, \ldots, T, \quad j = 1, \ldots, n. \end{array} \tag{8}$$

Here we use the *exact* value of the current capacity, c_τ (which is known); but for future capacities (which are unknown) we use instead the conditional mean $\hat{c}(t|\tau)$. The contract inequalities, $C_j f_j \ge q_j$, must be interpreted carefully. If a contract has expired, *i.e.*, its final time is less than τ, then it can be ignored. If a contract has not begun, *i.e.*, its initial time is greater than or equal to τ, then the contract inequality only involves future flows, and can be interpreted exactly as written. When a contract has already begun, and is still in force, *i.e.*, its start time is less than τ and its final time is at least τ, the contract inequality is interpreted as follows: the flows f_{jt} for $t < \tau$ are taken to be $\bar{f}_{jt}$, the previously chosen flow rates (which are constants). In this case $C_j f_j \ge q_j$ is essentially a contract on flow j that requires it to have, over the remaining contract period, a cumulative flow that is at least the remaining balance left on the contract.

The problem (8) is another NUMDC problem, which could be solved using the distributed dual decomposition algorithm. Let $F^\star$ be a solution of (8) Our choice of flow rates at time τ is then $\bar{f}_\tau = f_\tau^\star$.

To find the flow rates at any given time, then, we solve a NUMDC problem, that covers the remaining time up to T, and inherits any as yet unfilled contracts from the original problem. In this NUMDC problem, we substitute the expected future capacity for the actual future capacity (which we do not know).

We have described here only the simplest MPC-based heuristic. More sophisticated versions include some risk aversion or robustness in the problem solved at each step, for example by solving a stochastic programming problem, or a robust utility maximization problem. In the problem described, for example, we might replace the conditional mean $\hat{c}(t|\tau)$ with $\hat{c}(t|\tau) - \kappa\sigma(t|\tau)$, where $\kappa > 0$ is a risk

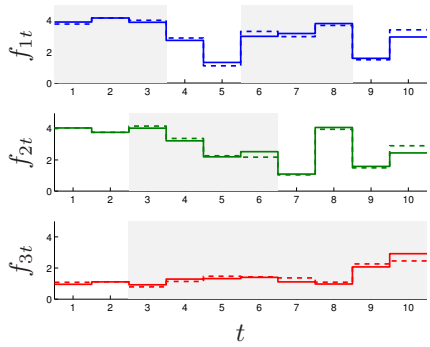

Fig. 9. Flow rates from MPC heuristic (solid), and the prescient solution (dashed). Delivery contract periods are shown as shaded areas.

aversion parameter, and $\sigma(t|\tau)$ is the conditional variance of c_t given the information available at time τ.

4.1 Example

We illustrate the MPC algorithm on the same problem instance from §3. In the MPC heuristic, we use $\hat{c}_1(t|\tau) = 5$, $\hat{c}_2(t|\tau) = 7$, and $\hat{c}_3(t|\tau) = 5$ for all τ, and $t > \tau$.

Figure 9 shows the flow rates obtained from the MPC heuristic, as well as the flows found from solving the original NUMDC problem. We can think of the solution of the original NUMDC problem as the *prescient* solution; it gives the (globally) optimal flow rates when the future capacities are fully known ahead of time. The MPC heuristic is a suboptimal, but causal, policy. In this example, the resulting flows are quite similar. The utility obtained by the MPC heuristic is 23.16; the utility obtained by the prescient solution is 23.33. The difference divided by nT gives the average utility loss per flow and time step, and is 0.06 for this example.

5. CONCLUSIONS

In this paper we presented a multi-period variation on the network utility maximization problem that includes delivery contract constraints, which couple flow rates across time. We described a distributed algorithm to solve this problem based on dual decomposition and established its convergence. We also looked at the case when some problem data is not known ahead of time and described a heuristic based on model predictive control.

There are many possible variations and extensions of these ideas and methods. We can modify the formulation in several ways. As a practical example, we can allow contract violations, imposing a penalty for contract violation. Here we subtract the total contract violation penalty charge

$$\sum_{j=1}^{q} \omega_j^T (q_j - C_j f_j)_+$$

from the over all utility $U(f)$, where $\omega_j > 0$ is the (vector of) penalty prices for contract j. (With contract penalties, the problem is always feasible.) Here the penalty is linear in the amount of contract shortfall; but any convex penalty function (*e.g.*, quadratic) can be used.

We can use more sophisticated algorithms to solve the dual problem. Our algorithm requires all rate, price, and subsidy updates to occur synchronously. If we use an incremental subgradient method (Nedić and Bertsekas [2001]) to solve the dual NUMDC problem, we would obtain an algorithm in which updates can occur asynchronously, still with guaranteed convergence.

REFERENCES

J. F. Benders. Partioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

D. Bertsekas. *Network Optimization: Continuous and Discrete Models*. Athena Scientific, 1998.

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, second edition, 1999.

D. P. Bertsekas and S. E. Shreve. *Stochastic Optimal Control: The Discrete-Time Case*. Athena Scientific, 1996.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

E. F. Camacho and C. Bordons. *Model Predictive Control*. Springer, second edition, 2004.

M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle. Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of the IEEE*, 95(1):255–312, January 2007.

G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8:101–111, 1960.

F. Kelly, A. Maulloo, and D. Tan. Rate control for communication networks: Shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1997.

S. H. Low and D. E. Lapsley. Optimization flow control I: Basic algorithms and convergence. *IEEE/ACM Transactions on Networking*, 7(6):861–874, December 1999.

J. M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, 2002.

A. Nedić. Subgradient methods for convex minimization. MIT Thesis, 2002.

A. Nedić and D. P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM J. on Optimization*, 12:109–138, 2001.

A. Nedić and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. LIDS report 2760, submitted for publication, 2007.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999.

B. Polyak. *Introduction to Optimization*. Optimization Software, Inc., 1987.

N. Z. Shor. *Minimization Methods for Non-Differentiable Functions*. Springer-Verlag, 1985.

N. Z. Shor. *Nondifferentiable Optimization and Polynomial Problems*. Kluwer Academic Publishers, 1998.

R. Srikant. *The Mathematics of Internet Congestion Control*. Birkäuser, 2004.

S. J. Wright. *Primal-Dual Interior-Point Methods*. Society for Industrial and Applied Mathematics, 1997.