A Tutorial on Decomposition Methods for Network Utility Maximization

Daniel P. Palomar, *Member, IEEE*, and Mung Chiang, *Member, IEEE*

Tutorial Paper

***Abstract*—A systematic understanding of the decomposability structures in network utility maximization is key to both resource allocation and functionality allocation. It helps us obtain the most appropriate distributed algorithm for a given network resource allocation problem, and quantifies the comparison across architectural alternatives of modularized network design. Decomposition theory naturally provides the mathematical language to build an analytic foundation for the design of modularized and distributed control of networks.**

In this tutorial paper, we first review the basics of convexity, Lagrange duality, distributed subgradient method, Jacobi and Gauss–Seidel iterations, and implication of different time scales of variable updates. Then, we introduce primal, dual, indirect, partial, and hierarchical decompositions, focusing on network utility maximization problem formulations and the meanings of primal and dual decompositions in terms of network architectures. Finally, we present recent examples on: systematic search for alternative decompositions; decoupling techniques for coupled objective functions; and decoupling techniques for coupled constraint sets that are not readily decomposable.

***Index Terms*—Congestion control, cross-layer design, decomposition, distributed algorithm, network architecture, network control by pricing, network utility maximization, optimization, power control, resource allocation.**

I. Introduction

MANY NETWORK resource allocation problems can be formulated as a constrained maximization of some utility function. There are at least three levels of understanding as to what it means to "efficiently solve" a network utility maximization problem. The first is on theoretical properties such as global optimality and duality gap. It is well known that for a convex optimization (minimizing a convex function over a convex constraint set), a local optimum is also a global optimum and the duality gap is zero under mild conditions. The second is on computational properties. There are provably polynomial-time and practically fast and scalable (but centralized) algorithms, such as interior-point methods, to solve convex optimization. The third is on decomposability structures, which may lead to distributed (and often iterative) algorithms that converge to the global optimum. Distributed solutions are particularly attractive in large-scale networks where a centralized solution is infeasible, nonscalable, too costly, or too fragile. It is the third level that we concern ourselves with in this tutorial paper.

The importance of "decomposability" to distributed solutions is similar to that of "convexity" to efficient computation of global optimum.[1] Similar to transformations that may turn an apparently nonconvex optimization into a convex one, there are alternative problem representations that may reveal hidden decomposability structures, even though representing the problem in a different way does not change the optimal solution. For a given problem representation, there are often many choices of distributed algorithms, each with possibly different characteristics of the following attributes: rate and robustness of convergence, tradeoff between local computation and global communication, and quantity and symmetry of message passing. Which alternative is the best depends on the specifics of the application.

A systematic understanding of the decomposability structures in network utility maximization is key to both *resource allocation* and *functionality allocation*. It obviously helps us obtain the most appropriate distributed algorithm for a given network resource allocation problem, ranging from distributed routing and scheduling to power control and congestion control. Perhaps even more importantly, it quantifies the comparison across architectural alternatives of modularized network design. A paramount issue in the design of network architecture is *where* to place functionalities and how to *connect* them, an issue that is often more critical than the detailed design of *how* to carry out a certain functionality. Decomposition theory naturally provides the mathematical language to build an analytic foundation for the design of *modularized* and *distributed* control of networks.

In particular, the framework of network utility maximization (NUM) has recently been substantially extended from an analytic tool of reverse-engineering transmission control protocol (TCP) congestion control to a mathematical theory of layered network architectures. This framework of "Layering as Optimization Decomposition" (surveyed in [1], see also discussions in [2] and another tutorial in this special issue [3]) rigorously integrates the various protocol layers into a single coherent theory,

Manuscript received April 20, 2006; revised May 8, 2006. This work was supported in part by the National Science Foundation under Grant CCF-0448012, Grant CNS-0417607, Grant CNS-0427677, and Grant CNS-0519880.

D. P. Palomar is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: danielp@princeton.edu).

M. Chiang is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA. He is also affiliated with Applied and Computational Mathematics, Princeton University, Princeton, NJ 08544 USA (e-mail: chiangm@princeton.edu).

Digital Object Identifier 10.1109/JSAC.2006.879350

[1] However, unlike the notion of convexity, the notion of decomposability does not have a precise definition. It is often quantified by the least amount of global communications needed among decomposed modules to solve a reference problem.

by regarding them as carrying out an asynchronous distributed computation over the network to implicitly solve a global NUM problem. Different layers iterate on different subsets of the decision variables at different time scales using local information to achieve individual optimality. These local algorithms collectively achieve a global objective. This approach provides a unifying view and holistic methodology to study performance and architectural issues in protocol layering.

Most of the papers in the vast, recent literature on NUM use a standard dual-based distributed algorithm. While the basic reverse engineering results for TCP shows that the current protocol is doing a dual-based distributed algorithm, it is also known that dual decomposition has major drawbacks when the application is inelastic and utility functions are nonconcave, which leads to divergence of congestion control. While most of the recent publications on "Layering As Optimization Decomposition" use congestion price as the "layering price," it is also known that congestion price can be a poor coordination across layers such as TCP and Internet protocol (IP). Contrary to the apparent impression that a simple dual decomposition is the only possibility, there are in fact many alternatives to solve a given network utility problem in different but all distributed manners. Each different decomposition of the mathematical problem formulation represents a new possibility of network architecture. But to develop such a theory, alternative decompositions must be fully explored to understand architectural possibilities, both "vertically" across functional modules (i.e., the layers), and "horizontally" across disparate network elements.

There is a large body of general results on the mathematics of distributed computation, some of which are summarized in standard textbooks such as [4]–[7]. In this tutorial paper, we will first in Section II review the basics of convexity, Lagrange duality, distributed subgradient method, Jacobi and Gauss–Seidel iterations, and implication of different time scales of variable updates. Then, we will introduce the basic techniques of primal, dual, indirect, partial, and hierarchical decompositions in Section III, focusing on NUM problem formulations and the associated engineering implications. In Section IV, we will present recent examples on: 1) systematic search for alternative decompositions; 2) decoupling techniques for coupled objective functions; and 3) decoupling techniques for coupled constraint sets that are not readily decomposable.

II. Basic Concepts

A. Convex Optimization and Lagrange Duality

Convex optimization is a well-developed area in both the theoretical and practical aspects, especially during the last two decades when a number of fundamental and practical results have been obtained [8], [9]. Consider the following generic optimization problem:

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & f_0(\mathbf{x}) \\ \text{subject to} & f_i(\mathbf{x}) \le 0 \quad 1 \le i \le m, \\ & h_i(\mathbf{x}) = 0 \quad 1 \le i \le p \end{array} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the optimization variable, f_0 is the *cost* or *objective function*, $f_1, \cdots, f_m$ are the m inequality constraint functions, and $h_1, \cdots, h_p$ are the p equality constraint functions.

If the objective and inequality constraint functions are convex[2] and the equality constraint functions are linear (or, more generally, affine), the problem is then a *convex optimization problem* (or *convex program*). A point $\mathbf{x}$ in the domain of the problem (set of points for which the objective and all constraint functions are defined) is *feasible* if it satisfies all the constraints $f_i(\mathbf{x}) \le 0$ and $h_i(\mathbf{x}) = 0$. The problem (1) is said to be feasible if there exists at least one feasible point and *infeasible* otherwise. The *optimal value* (minimal value) is denoted by $f^\star$ and is achieved at an optimal solution $\mathbf{x}^\star$, i.e., $f^\star = f_0(\mathbf{x}^\star)$.

Convexity is often viewed as the "watershed" between easy and hard optimization problems. This is in part because a local optimum of convex optimization is also globally optimal, duality gap is zero under certain constraint qualifications, the Karush–Kuhn–Tucker (KKT) conditions are both necessary and sufficient for primal-dual optimality, and there are centralized algorithms that are provably polynomial-time and very fast and scalable in practice. For details, the reader is referred to books [8], [9] and another tutorial in this special issue [10].

In particular, convex optimization has highly useful Lagrange duality properties, which also lead to decomposability structures. Lagrange duality theory links the original minimization problem (1), termed primal problem, with a dual maximization problem, which sometimes readily presents decomposition possibilities. The basic idea in Lagrange duality is to relax the original problem (1) by transferring the constraints to the objective in the form of a weighted sum. The *Lagrangian* of (1) is defined as

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^{p} \nu_i h_i(\mathbf{x}) \tag{2}$$

where λ_i and ν_i are the *Lagrange multipliers* or *prices* associated with the ith inequality constraint $f_i(\mathbf{x}) \le 0$ and with the ith equality constraint $h_i(\mathbf{x}) = 0$, respectively.

The optimization variable $\mathbf{x}$ is called the *primal variable* and the Lagrange multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$ are also termed the *dual variables*. Similarly, the original objective function $f_0(\mathbf{x})$ is referred to as the *primal objective*, whereas the *dual objective* $g(\boldsymbol{\lambda}, \boldsymbol{\nu})$ is defined as the minimum value of the Lagrangian over $\mathbf{x}$

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) \tag{3}$$

which is concave even if the original problem is not convex because it is the pointwise infimum of a family of affine functions of $(\boldsymbol{\lambda}, \boldsymbol{\nu})$. Note that the infimum in (3) is with respect all $\mathbf{x}$ (not necessarily feasible points). The dual variables $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ are *dual feasible* if $\boldsymbol{\lambda} \ge \mathbf{0}$.

It turns out that the primal and dual objectives satisfy $f_0(\mathbf{x}) \ge g(\boldsymbol{\lambda}, \boldsymbol{\nu})$ for any feasible $\mathbf{x}$ and $(\boldsymbol{\lambda}, \boldsymbol{\nu})$. The dual function can then

[2] A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if, for all $\mathbf{x}, \mathbf{y} \in \operatorname{dom} f$ and $\theta \in [0, 1]$, $\theta\mathbf{x} + (1-\theta)\mathbf{y} \in \operatorname{dom} f$ (i.e., the domain is a convex set) and $f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \le \theta f(\mathbf{x}) + (1-\theta) f(\mathbf{y})$.

be maximized to obtain a lower bound on the optimal value $f^\star$ of the original problem (1):

$$\begin{array}{ll}\underset{\boldsymbol{\lambda},\boldsymbol{\nu}}{\text{maximize}} & g(\boldsymbol{\lambda},\boldsymbol{\nu}) \\ \text{subject to} & \boldsymbol{\lambda} \geq \mathbf{0}\end{array} \tag{4}$$

which is always a convex optimization problem even if the original problem is not convex.

The difference between the optimal primal objective $f^\star$ and the optimal dual objective $g^\star$ is called the *duality gap*, which is always nonnegative $f^\star - g^\star \geq 0$ (weak duality). A central result in convex analysis [8], [9] is that when the problem is convex, under some mild technical conditions (called constraint qualifications[3]), the duality gap reduces to zero at the optimal (i.e., strong duality holds). Hence, the primal problem (1) can be equivalently solved by solving the dual problem (4).

B. Gradient and Subgradient Algorithms

After performing a decomposition, the objective function of the resulting master problem may or may not be differentiable. For differentiable/nondifferentiable functions a gradient/subgradient method is very convenient because of its simplicity, little requirements of memory usage, and amenability for parallel implementation [7], [8], [11].

Consider the following general concave maximization over a convex set:

$$\begin{array}{ll}\underset{\mathbf{x}}{\text{maximize}} & f_0(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in \mathcal{X}.\end{array} \tag{5}$$

Both the gradient and subgradient projection methods generate a sequence of feasible points $\{\mathbf{x}(t)\}$ as

$$\mathbf{x}(t+1) = [\mathbf{x}(t) + \alpha(t)\mathbf{s}(t)]_{\mathcal{X}} \tag{6}$$

where $\mathbf{s}(t)$ is a gradient of f_0 evaluated at the point $\mathbf{x}(t)$ if f_0 is differentiable and a subgradient otherwise, $[\cdot]_{\mathcal{X}}$ denotes the projection onto the feasible set $\mathcal{X}$, and $\alpha(t)$ is a positive step-size. It is interesting to point out that each iteration of the subgradient method may not improve the objective value as happens with a gradient method. What makes the subgradient method converge is that, for sufficiently small step-size, the distance of the current solution $\mathbf{x}(t)$ to the optimal solution $\mathbf{x}^\star$ decreases.

There are many results on convergence of the gradient/subgradient method with different choices of step-sizes [8], [11], [12]. For example, for a diminishing step-size rule $\alpha(t) = (1+m)/(t+m)$, where m is a fixed nonnegative number, the algorithm is guaranteed to converge to the optimal value (assuming bounded gradients/subgradients) [12]. For a constant step-size $\alpha(t) = \alpha$, more convenient for distributed algorithms, the gradient algorithm converges to the optimal value provided that the step-size is sufficiently small (assuming that the gradient is Lipschitz) [8], whereas for the subgradient algorithm the best value converges to within some range of the optimal value (assuming bounded subgradients) [12].

C. Order of Updates: Gauss–Seidel and Jacobi Algorithms

Consider the following general minimization problem:

$$\begin{array}{ll}\underset{\mathbf{x}}{\text{minimize}} & f(\mathbf{x}_1, \cdots, \mathbf{x}_n) \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i \quad \forall i\end{array} \tag{7}$$

where $\mathbf{x} = [\mathbf{x}_1^T, \cdots, \mathbf{x}_n^T]^T$ and the feasible set is the Cartesian product of closed convex sets $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$.

The nonlinear Gauss–Seidel algorithm [4, Sec. 3.3.5] (also termed block-coordinate descent algorithm [8]) consists of iteratively optimizing in a circular fashion with respect to one set of variables while keeping the rest fixed. Formally, it is defined as

$$\mathbf{x}_i^{(t+1)} = \arg\min_{\mathbf{x}_i \in \mathcal{X}_i} f\left(\mathbf{x}_1^{(t+1)}, \cdots, \mathbf{x}_{i-1}^{(t+1)}, \mathbf{x}_i, \mathbf{x}_{i+1}^{(t)}, \cdots, \mathbf{x}_n^{(t)}\right) \tag{8}$$

where t is the index for a global iteration.

The nonlinear Jacobi algorithm [4, Sec. 3.3.5] consists of iteratively optimizing in a parallel fashion with respect to one set of variables while keeping the rest fixed. Formally, it is defined as

$$\mathbf{x}_i^{(t+1)} = \arg\min_{\mathbf{x}_i \in \mathcal{X}_i} f\left(\mathbf{x}_1^{(t)}, \cdots, \mathbf{x}_{i-1}^{(t)}, \mathbf{x}_i, \mathbf{x}_{i+1}^{(t)}, \cdots, \mathbf{x}_n^{(t)}\right). \tag{9}$$

If the function f is continuously differentiable and convex on the set $\mathcal{X}$, and each of the minimizations with respect to each single variable $\mathbf{x}_i$ is uniquely attained, then every limit point of the sequence $\{\mathbf{x}^{(t)}\}$ generated by the nonlinear Gauss–Seidel algorithm in (8) minimizes f over $\mathcal{X}$ [4, Ch. 3, Prop. 3.9], [8, Prop. 2.7.1]. Observe that only if the original problem (7) has a unique solution can we guarantee that the sequence $\{\mathbf{x}^{(t)}\}$ will have a unique limit point and, hence, that it converges.

Suppose that the function f is continuously differentiable and the mapping defined by $T(\mathbf{x}) = \mathbf{x} - \gamma \nabla f(\mathbf{x})$ is a contraction for some positive scalar γ with respect to the block-maximum norm $\|\mathbf{x}\| = \max_i \|\mathbf{x}_i\|_2 / w_i$, where the w_i's are positive scalars. Then, there exists a unique solution $\mathbf{x}^\star$ to problem (7) and the sequence $\{\mathbf{x}^{(t)}\}$ generated by the nonlinear Gauss–Seidel algorithm in (8) and by the Jacobi algorithm in (9) converges to $\mathbf{x}^\star$ geometrically [4, Ch. 3, Prop. 3.10]. For conditions for the contraction assumption to be satisfied, see [4, Ch. 3, Prop. 1.10].

D. Timescale of Updates

The update of the different variables in an optimization problem can be done in many different ways. To start with, the variables can be optimized either in one-shot (with full convergence) or iteratively [as with the gradient/subgradient algorithm in (6)]. Also, optimization can be done according to different update schedules; in particular, the two most common ones are the sequential optimization in (8) and the parallel optimization in (9), i.e., Gauss–Seidel and Jacobi algorithms. Even the frequency of updates of different subsets of variables can be different. The combination of all these possibilities in the updates of the different variables of an optimization problem leads to a variety of different algorithms that may operate on a

[3] One simple version of the constraint qualifications is Slater's condition, which is satisfied when the problem is strictly feasible (i.e., when there exists $\mathbf{x}$ such that $f_i(\mathbf{x}) < 0$ for $1 \leq i \leq m$ and $h_i(\mathbf{x}) = 0$ for $1 \leq i \leq p$) [8], [9].

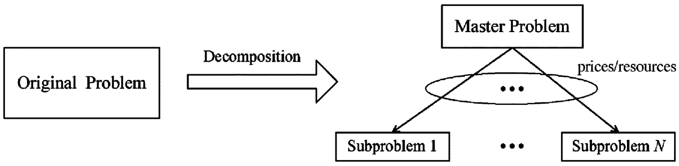

Fig. 1. Decomposition of a problem into several subproblems controlled by a master problem through prices (dual decomposition) or direct resource allocation (primal decomposition).

single or several time scales, clearly with different implications for implementation.

E. Implicit/Explicit Message Passing and Converge Properties

Different distributed algorithms can be obtained based on a combination of different decompositions (as discussed in the next section) with different algorithms (e.g., gradient/subgradient, Gauss–Seidel, and Jacobi). Each of them has different characteristics in terms of amount of message passing and convergence properties.

Whenever a primal or dual decomposition is applied, the original problem is essentially decomposed into a master problem and subproblems with communication between the two levels. This communication is in the form of message passing which introduces some overhead in the network. In some cases, however, this message passing can be implicit in the system, e.g., delay and packet error probability, as these quantities have physical meanings and can be implicitly measured without the need of explicit signaling.

The algorithms presented in this paper have a provable convergence to the global optimum of the original problem which will be assumed convex. The speed of convergence is more difficult to quantify and will depend on many factors such as the number of levels in the decomposition (similarly, number of time scales), the amount of signaling, and the particular combination of numerical algorithms employed.

III. Decomposition Theory

The basic idea of decomposition is to decompose the original large problem into distributively solvable *subproblems* which are then coordinated by a high-level *master problem* by means of some kind of signaling (see Fig. 1) [4], [7], [8]. Most of the existing decomposition techniques can be classified into *primal decomposition* and *dual decomposition* methods. The former is based on decomposing the original primal problem, whereas the latter is based on decomposing the Lagrangian dual problem [8], [11].

Primal decomposition methods correspond to a *direct resource allocation* since the master problem allocates the existing resources by directly giving each subproblem the amount of resources that it can use. Dual decomposition methods correspond to a *resource allocation via pricing* since the master problem sets the price for the resources to each subproblem, which has to decide the amount of resources to be used depending on the price.

A. Dual Decomposition

A dual decomposition is appropriate when the problem has a coupling constraint such that, when relaxed, the optimization problem decouples into several subproblems. Consider, for example, the following problem:

$$\begin{array}{ll} \underset{\{\mathbf{x}_i\}}{\text{maximize}} & \sum_i f_i(\mathbf{x}_i) \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i \quad \forall i \\ & \sum_i \mathbf{h}_i(\mathbf{x}_i) \le \mathbf{c}. \end{array} \tag{10}$$

Clearly, if the constraint $\sum_i \mathbf{h}_i(\mathbf{x}_i) \le \mathbf{c}$ were absent, then the problem would decouple. Therefore, it makes sense to form the Lagrangian by relaxing the coupling constraint in (10) as

$$\begin{array}{ll} \underset{\{\mathbf{x}_i\}}{\text{maximize}} & \sum_i f_i(\mathbf{x}_i) - \boldsymbol{\lambda}^T \left(\sum_i \mathbf{h}_i(\mathbf{x}_i) - \mathbf{c} \right) \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i \quad \forall i \end{array} \tag{11}$$

such that the optimization separates into two levels of optimization. At the lower level, we have the subproblems (i.e., the Lagrangians), one for each i, in which (11) decouples

$$\begin{array}{ll} \underset{\mathbf{x}_i}{\text{maximize}} & f_i(\mathbf{x}_i) - \boldsymbol{\lambda}^T \mathbf{h}_i(\mathbf{x}_i) \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i. \end{array} \tag{12}$$

At the higher level, we have the master dual problem in charge of updating the dual variable $\boldsymbol{\lambda}$ by solving the dual problem:

$$\begin{array}{ll} \underset{\boldsymbol{\lambda}}{\text{minimize}} & g(\boldsymbol{\lambda}) = \sum_i g_i(\boldsymbol{\lambda}) + \boldsymbol{\lambda}^T \mathbf{c} \\ \text{subject to} & \boldsymbol{\lambda} \ge \mathbf{0} \end{array} \tag{13}$$

where $g_i(\boldsymbol{\lambda})$ is the dual function obtained as the maximum value of the Lagrangian solved in (12) for a given $\boldsymbol{\lambda}$. This approach is in fact solving the dual problem instead of the original primal one. Hence, it will only give appropriate results if strong duality holds (e.g., when the original problem is convex and there exist strictly feasible solutions).

If the dual function $g(\boldsymbol{\lambda})$ is differentiable, then the master dual problem in (13) can be solved with a gradient method. In general, however, it may not be differentiable, and the subgradient method becomes again a convenient approach which only requires the knowledge a subgradient for each $g_i(\boldsymbol{\lambda})$ given by [8, Sec. 6.1]

$$\mathbf{s}_i(\boldsymbol{\lambda}) = -\mathbf{h}_i\left(\mathbf{x}_i^\star(\boldsymbol{\lambda})\right) \tag{14}$$

where $\mathbf{x}_i^\star(\boldsymbol{\lambda})$ is the optimal solution of problem (12) for a given $\boldsymbol{\lambda}$. The global subgradient is then $\mathbf{s}(\boldsymbol{\lambda}) = \sum_i \mathbf{s}_i(\mathbf{y}) + \mathbf{c} = \mathbf{c} - \sum_i \mathbf{h}_i(\mathbf{x}_i^\star(\boldsymbol{\lambda})))$. The subproblems in (12) can be locally and independently solved with knowledge of $\boldsymbol{\lambda}$.

B. Dual Decomposition Application

We illustrate the one-level, full dual decomposition by applying the standard method (see, e.g., [13] and [14]) to the basic NUM problem in [19] for distributed end-to-end rate allocation.

Consider a communication network with L links, each with a fixed capacity of c_l, and S sources or nodes, each transmitting at a source rate of x_s. Each source s emits one flow, using a fixed set of links $L(s)$ in its path, and has a utility function $U_s(x_s)$. NUM is the problem of maximizing the total utility $\sum_s U_s(x_s)$, over the source rates $\mathbf{x}$, subject to linear flow constraints $\sum_{s:l \in L(s)} x_s \leq c_l$ for all links l

$$\begin{array}{ll} \underset{\mathbf{x} \geq \mathbf{0}}{\text{maximize}} & \sum_s U_s(x_s) \\ \text{subject to} & \sum_{s:l \in L(s)} x_s \leq c_l \quad \forall l \end{array} \tag{15}$$

where the utilities U_s are twice-differentiable, increasing, and strictly concave functions.

One of the standard distributed algorithms to solve (15) is based on a dual decomposition. We first form the Lagrangian of (15)

$$\begin{aligned} L(\mathbf{x}, \boldsymbol{\lambda}) &= \sum_s U_s(x_s) + \sum_l \lambda_l \left(c_l - \sum_{s:l \in L(s)} x_s \right) \\ &= \sum_s L_s(x_s, \lambda^s) + \sum_l c_l \lambda_l \end{aligned} \tag{16}$$

where $\lambda_l \geq 0$ is the Lagrange multiplier (link price) associated with the linear flow constraint on link l, $\lambda^s = \sum_{l \in L(s)} \lambda_l$ is the aggregate path congestion price of those links used by source s, and $L_s(x_s, \lambda^s) = U_s(x_s) - \lambda^s x_s$ is the sth Lagrangian to be maximized by the sth source.

The dual decomposition results in each source s solving, for the given λ^s

$$x_s^\star(\lambda^s) = \arg \max_{x_s \geq 0} \left[U_s(x_s) - \lambda^s x_s \right] \quad \forall s \tag{17}$$

which is unique due to the strict concavity of U_s.

The master dual problem is

$$\begin{array}{ll} \underset{\boldsymbol{\lambda}}{\text{minimize}} & g(\boldsymbol{\lambda}) = \sum_s g_s(\boldsymbol{\lambda}) + \boldsymbol{\lambda}^T \mathbf{c} \\ \text{subject to} & \boldsymbol{\lambda} \geq \mathbf{0} \end{array} \tag{18}$$

where $g_s(\boldsymbol{\lambda}) = L_s(x_s^\star(\lambda^s), \lambda^s)$. Since the solution in (17) is unique, it follows that the dual function $g(\boldsymbol{\lambda})$ is differentiable and the following gradient method can be used:

$$\lambda_l(t+1) = \left[\lambda_l(t) - \alpha \left(c_l - \sum_{s:l \in L(s)} x_s^\star\left(\lambda^s(t)\right) \right) \right]^+ \quad \forall l \tag{19}$$

where t is the iteration index, $\alpha > 0$ is a sufficiently small positive step-size, and $[\cdot]^+$ denotes the projection onto the nonnegative orthant.

The dual variable $\boldsymbol{\lambda}(t)$ will converge to the dual optimal $\boldsymbol{\lambda}^\star$ as $t \to \infty$ and, since the duality gap for (15) is zero and the solution to (17) is unique, the primal variable $\mathbf{x}^\star(\boldsymbol{\lambda}(t))$ will also converge to the primal optimal variable $\mathbf{x}^\star$.

Summarizing, we have the following algorithm.

Standard Dual Algorithm to solve the basic NUM (15):

- Parameters: each source needs its utility U_s and each link its capacity c_l.
- Initialization: set $t = 0$ and $\lambda_l(0)$ equal to some nonnegative value for all l.

1) Each source locally solves its problem by computing (17) and then broadcasts the solution $x_s^\star(\lambda^s(t))$.
2) Each link updates its prices with the gradient iterate (19) and broadcasts the new price $\lambda_l(t+1)$.
3) Set $t \leftarrow t+1$ and go to step 1 (until satisfying termination criterion).

Note that there is no need for explicit message passing since λ^s can be measured by each source s as queuing delay and $\sum_{s:l \in L(s)} x_s$ can be measured by each link l as the total traffic load.

C. Primal Decomposition

A primal decomposition is appropriate when the problem has a coupling variable such that, when fixed to some value, the rest of the optimization problem decouples into several subproblems. Consider, for example, the following problem:

$$\begin{array}{lll} \underset{\mathbf{y}, \{\mathbf{x}_i\}}{\text{maximize}} & \sum_i f_i(\mathbf{x}_i) & \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i & \forall i \\ & \mathbf{A}_i \mathbf{x}_i \leq \mathbf{y} & \\ & \mathbf{y} \in \mathcal{Y}. & \end{array} \tag{20}$$

Clearly, if variable $\mathbf{y}$ were fixed, then the problem would decouple. Therefore, it makes sense to separate the optimization in (20) into two levels of optimization. At the lower level, we have the subproblems, one for each i, in which (20) decouples when $\mathbf{y}$ is fixed

$$\begin{array}{ll} \underset{\mathbf{x}_i}{\text{maximize}} & f_i(\mathbf{x}_i) \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i \\ & \mathbf{A}_i \mathbf{x}_i \leq \mathbf{y}. \end{array} \tag{21}$$

At the higher level, we have the master problem in charge of updating the coupling variable $\mathbf{y}$ by solving

$$\begin{array}{ll} \underset{\mathbf{y}}{\text{maximize}} & \sum_i f_i^\star(\mathbf{y}) \\ \text{subject to} & \mathbf{y} \in \mathcal{Y} \end{array} \tag{22}$$

where $f_i^\star(\mathbf{y})$ is the optimal objective value of problem (21) for a given $\mathbf{y}$. If the original problem (20) is a convex optimization problem, then the subproblems as well as the master problem are all convex programs.

If the function $\sum_i f_i^\star(\mathbf{y})$ is differentiable, then the master problem (22) can be solved with a gradient method. In general, however, the objective function $\sum_i f_i^\star(\mathbf{y})$ may be nondifferentiable, and the subgradient method is a convenient approach which only requires the knowledge a subgradient (see Section II-B for details on subgradients) for each $f_i^\star(\mathbf{y})$ as given by [7, Ch. 9], [8, Sec. 6.4.2]

$$\mathbf{s}_i(\mathbf{y}) = \boldsymbol{\lambda}_i^\star(\mathbf{y}) \tag{23}$$

where $\boldsymbol{\lambda}_i^\star(\mathbf{y})$ is the optimal Lagrange multiplier corresponding to the constraint $\mathbf{A}_i\mathbf{x}_i \le \mathbf{y}$ in problem (21). The global subgradient is then $\mathbf{s}(\mathbf{y}) = \sum_i \mathbf{s}_i(\mathbf{y}) = \sum_i \boldsymbol{\lambda}_i^\star(\mathbf{y})$. The subproblems in (21) can be locally and independently solved with the knowledge of $\mathbf{y}$.

D. Primal Decomposition Application

Primal decomposition is naturally applicable to resource sharing scenarios where "virtualization" or "slicing" of the resources are carried out by dividing the total resource to multiple parts, one for each of the entities (e.g., source-destination pairs sharing link capacities, or experiment running on a shared communication infrastructure) competing for the resource. As will be shown at the end of the next subsection, the points where the resources are divided can be represented by auxiliary variables in a primal decomposition. If these variables are fixed, we would have a static slicing of the resources, which can be suboptimal. If these variables are optimized by a master problem and used to coordinate the allocation of resources to the subproblems, we would have an optimal dynamic slicing. In summary, a canonical engineering example for dual decomposition is end-to-end pricing feedback (pricing coordinated control), whereas that for primal decomposition is dynamic slicing (direct resource allocation).

Before finishing with this introduction of the basic techniques of primal and dual decompositions, we briefly mention two related topics. First, note that "primal decomposition" in this paper is not the same concept as "primal penalty function approach" or the "primal-driven network control" in, e.g., [19]. Second, there is also a celebrated decomposition in [19] of the SYSTEM problem (i.e., the basic NUM problem) into one NETWORK problem and local USER problems, which can be obtained, for example, in this way (more details in [19] and [20]): first turn the SYSTEM problem into an unconstrained optimization with one penalty function term for each of the original constraints (e.g., linear capacity constraints), then write down the gradient update of the new objective function in the unconstrained problem, and finally identify the weights w_s in the weighted log maximization in the NETWORK problem as $x_s U_s'(x_s)$ and identify the price λ_s as the sum of the derivatives of the penalty functions along the path used by source s. Then, the decomposition in problem domain (from SYSTEM problem into NETWORK problem plus USER problems), and the associated distributed algorithm, can be recovered.

E. Indirect Decomposition

As we have seen, problems with coupling constraints are naturally suited for a dual decomposition, whereas problems with coupling variables are convenient for a primal decomposition. However, this is not a strict rule as often the problem can be reformulated, and then more effective primal and dual decompositions can be indirectly applied. The introduction of *auxiliary variables* is the key element that provides much flexibility in terms of choosing a primal or a dual decomposition. For example, in [21] and Section IV-B, an indirect dual decomposition is applied to a problem with coupling among the utilities based on the use of auxiliary variables.

The basic techniques of indirect decomposition are illustrated as follows. Problem (20) contains the coupling variable $\mathbf{y}$ and was decoupled in (21) and (22) via a primal decomposition approach. However, it can also be solved with an indirect dual decomposition by first introducing the additional auxiliary variables $\{\mathbf{y}_i\}$

$$\begin{array}{lll} \underset{\mathbf{y},\{\mathbf{y}_i\},\{\mathbf{x}_i\}}{\text{maximize}} & \sum_i f_i(\mathbf{x}_i) & \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i & \forall i \\ & \mathbf{A}_i\mathbf{x}_i \le \mathbf{y}_i & \\ & \mathbf{y}_i = \mathbf{y} & \\ & \mathbf{y} \in \mathcal{Y} & \end{array} \tag{24}$$

and then relaxing the coupling constraints $\mathbf{y}_i = \mathbf{y}$ via a dual decomposition.

Consider now problem (10) which contains the coupling constraint $\sum_i \mathbf{h}_i(\mathbf{x}_i) \le \mathbf{c}$ and was decoupled in (12) and (13) via a dual decomposition. By introducing again additional auxiliary variables $\{\mathbf{y}_i\}$ the problem becomes

$$\begin{array}{lll} \underset{\{\mathbf{y}_i\},\{\mathbf{x}_i\}}{\text{maximize}} & \sum_i f_i(\mathbf{x}_i) & \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i & \forall i \\ & \mathbf{h}_i(\mathbf{x}_i) \le \mathbf{y}_i & \\ & \sum_i \mathbf{y}_i \le \mathbf{c} & \end{array} \tag{25}$$

and the coupling variable $\mathbf{y} = [\mathbf{y}_1^T, \cdots, \mathbf{y}_N^T]^T$ can be dealt with using a primal decomposition.

Another example is the following problem:

$$\begin{array}{ll} \underset{\mathbf{x},\mathbf{y}}{\text{maximize}} & \mathbf{c}^T\mathbf{x} + \tilde{\mathbf{c}}^T\mathbf{y} \\ \text{subject to} & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \tilde{\mathbf{A}}\mathbf{y} \le \tilde{\mathbf{b}} \\ & \mathbf{F}\mathbf{x} + \tilde{\mathbf{F}}\mathbf{y} \le \mathbf{h}. \end{array} \tag{26}$$

We can take a dual decomposition approach by relaxing the coupling constraint $\mathbf{F}\mathbf{x} + \tilde{\mathbf{F}}\mathbf{y} \le \mathbf{h}$. However, another alternative is to transform the coupling constraint into a coupling variable. This is easily acomplished by introducing the auxiliary variable $\mathbf{z}$ and rewriting the coupling constraint as

$$\mathbf{F}\mathbf{x} \le \mathbf{z} \tag{27}$$

$$\tilde{\mathbf{F}}\mathbf{y} \le \mathbf{h} - \mathbf{z}. \tag{28}$$

At this point, we can use a primal decomposition to deal with the coupling variable $\mathbf{z}$.

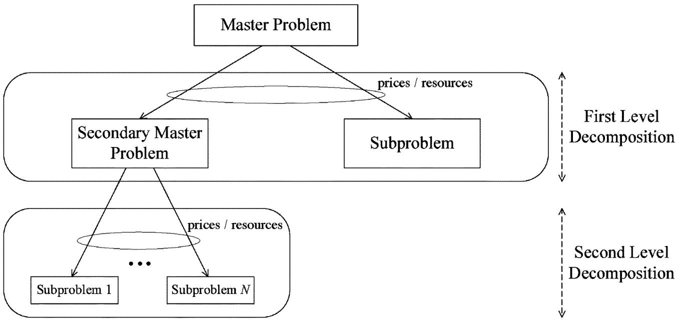

Fig. 2. Example of a hierarchical primal/dual decomposition with two levels.

F. Hierarchical Decomposition

An important factor that leads to alternatives of distributed architectures is the application of primal/dual decompositions recursively: the basic decompositions are repeatedly applied to the problem to obtain smaller and smaller subproblems, as illustrated in Fig. 2.

For example, consider the following problem which includes both a coupling variable and a coupling constraint:

$$\begin{array}{lll} \underset{\mathbf{y},\{\mathbf{x}_i\}}{\text{maximize}} & \sum_i f_i(\mathbf{x}_i, \mathbf{y}) & \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i & \forall i \\ & \sum_i \mathbf{h}_i(\mathbf{x}_i) \leq \mathbf{c} & \\ & \mathbf{A}_i \mathbf{x}_i \leq \mathbf{y} & \\ & \mathbf{y} \in \mathcal{Y}. & \end{array} \tag{29}$$

One way to decouple this problem is by first taking a primal decomposition with respect to the coupling variable $\mathbf{y}$, and then a dual decomposition with respect to the coupling constraint $\sum_i \mathbf{h}_i(\mathbf{x}_i) \leq \mathbf{c}$. This would produce a two-level optimization decomposition: a master primal problem, a secondary master dual problem, and the subproblems. Observe that an alternative approach would be to first take a dual decomposition and then a primal one. The order of decompositions is important as this determines the relationship between the different time scales of the resulting algorithm and the different modules of the network control protocol.

Another example that shows flexibility in terms of different decompositions is the following general problem with two sets of constraints:

$$\begin{array}{lll} \underset{\mathbf{x}}{\text{maximize}} & f_0(\mathbf{x}) & \\ \text{subject to} & f_i(\mathbf{x}) \leq 0 & \forall i \\ & g_i(\mathbf{x}) \leq 0. & \end{array} \tag{30}$$

One way to deal with this problem is via the dual problem with a full relaxation of both sets of constraints to obtain the dual function $g(\boldsymbol{\lambda}, \boldsymbol{\mu})$. At this point, instead of minimizing g directly with respect to $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, it can be minimized over only one set of Lagrange multipliers first, and then over the remaining one: $\min_{\boldsymbol{\lambda}} \min_{\boldsymbol{\mu}} g(\boldsymbol{\lambda}, \boldsymbol{\mu})$. This approach corresponds to first applying a full dual decomposition, and then a primal one on the dual problem.

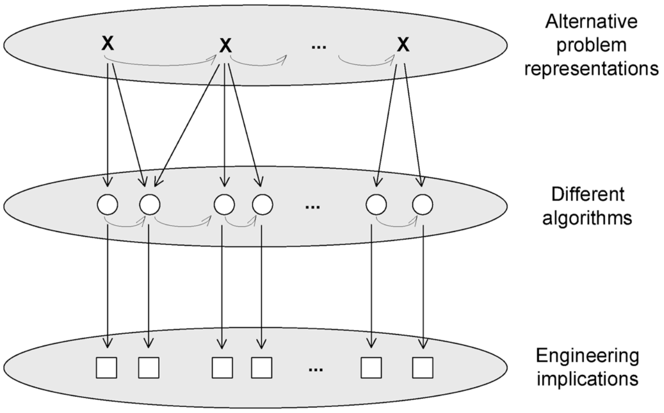

Fig. 3. Alternative problem representations may lead to different decomposability structures, which in turn may lead to a choice of distributed algorithms, each with different engineering implications to network architectures.

Alternatively, problem (30) can be approached via the dual but with a partial relaxation of only one set of constraint, say $f_i(\mathbf{x}) \leq 0, \forall i$, obtaining the dual function $g(\boldsymbol{\lambda})$ to be minimized by the master problem. Observe now that in order to compute $g(\boldsymbol{\lambda})$ for a given $\boldsymbol{\lambda}$, the partial Lagrangian has to be maximized subject to the remaining constraints $g_i(\mathbf{x}) \leq 0 \ \forall i$, for which yet another relaxation can be used. This approach corresponds to first applying a partial dual decomposition and then, for the subproblem, another dual decomposition.

When there are more than one level of decomposition, and all levels conduct some type of iterative algorithms, such as the subgradient method, convergence and stability are guaranteed if the lower level master problem is solved on a faster time scale than the higher level master problem, so that at each iteration of a master problem all the problems at a lower level have already converged. If the updates of the different subproblems operate on similar time scales, convergence of the overall system can still be guaranteed under certain technical conditions [4], [22].

IV. Variations and Extensions

A. Systematic Search for Alternative Decomposition [23]

Section III provides us with the fundamental building blocks to obtain a variety of different algorithms for the same network problem. As illustrated in Fig. 3, each distributed algorithm may have different characteristics among the following attributes: speed and robustness of convergence, amount and symmetry of message passing, amount and symmetry of local computation, implications to engineering implementation, etc. Sometimes, even an alternative representation of the same problem may lead to new decomposability structures, and thus new alternatives in distributed algorithms. Even though problem transformations do not alter the optimal solutions, they may provide new algorithms to attain the optimal solutions.

Examples of alternative "vertical decompositions," which further lead to different layered protocol stacks, can be found, for example, in [15] where a new scheduling layer is introduced, in [16] where the routing functionality is absorbed into congestion control and scheduling, and in [17] and [18] where different time scales of joint congestion and contention control provide implementation choices.

Examples of alternative "horizontal decompositions" can be found in [23]. One of the examples, on quality-of-service (QoS) rate allocation, is briefly summarized below to illustrate the basic idea of alternative decompositions. Sometimes a rate allocation mechanism needs to differentiate users in different QoS classes. For example, the total link capacity received by each QoS class must lie within a range prescribed in the service level agreement. Such constraints introduce new coupling to the basic NUM problem and lead to alternative decomposition possibilities. We will see in this section two different distributed algorithms to solve this type of QoS rate allocation problem, both with a differential pricing interpretation to the new set of Lagrange multiplier introduced.

Consider now the basic NUM but with different classes of users that will be treated differently. We constrain the rates to be within a range for each class. To simplify the exposition we consider only two classes of users. Denoting by $y_l^{(1)}$ and $y_l^{(2)}$ the aggregate rates of classes 1 and 2, respectively, along the lth link, the problem formulation is

$$\begin{array}{lll} \underset{\mathbf{x},\mathbf{y}^{(1)},\mathbf{y}^{(2)}\geq \mathbf{0}}{\text{maximize}} & \sum_s U_s(x_s) & \\ \text{subject to} & \sum_{s\in S_i: l\in L(s)} x_s \leq y_l^{(i)} & \forall l, i = 1,2 \\ & \mathbf{y}^{(1)} + \mathbf{y}^{(2)} \leq \mathbf{c} & \\ & \mathbf{c}_{\min}^{(i)} \leq \mathbf{y}^{(i)} \leq \mathbf{c}_{\max}^{(i)}. & \end{array} \tag{31}$$

Observe that in the absence of the constraints $\mathbf{c}_{\min}^{(i)} \leq \mathbf{y}^{(i)} \leq \mathbf{c}_{\max}^{(i)}$, problem (31) becomes the basic NUM in (15). Also note that, without loss of generality, the equality flow constraints can be rewritten as inequality flow constraints. We will consider two decompositions: a primal decomposition with respect to the aggregate rate of each class, and a dual decomposition with respect to the total aggregate rate constraints from both classes.

1) Primal-Dual Decomposition: Consider first a primal decomposition of (31) by fixing the aggregate rates $\mathbf{y}^{(1)}$ and $\mathbf{y}^{(2)}$. Problem (31) becomes two independent subproblems, for $i = 1$, 2, identical to the basic NUM in (15)

$$\begin{array}{lll} \underset{\mathbf{x}\geq \mathbf{0}}{\text{maximize}} & \sum_{s\in S_i} U_s(x_s) & \\ \text{subject to} & \sum_{s\in S_i: l\in L(s)} x_s \leq y_l^{(i)} & \forall l \end{array} \tag{32}$$

where the fixed aggregate rates $y_l^{(i)}$ play the role of the fixed link capacities in the basic NUM of (15). These two independent basic NUMs can be solved as explained in Section III-B.

The master primal problem is

$$\begin{array}{lll} \underset{\mathbf{y}^{(1)},\mathbf{y}^{(2)}\geq \mathbf{0}}{\text{maximize}} & U_1^\star\left(\mathbf{y}^{(1)}\right) + U_2^\star\left(\mathbf{y}^{(2)}\right) & \\ \text{subject to} & \mathbf{y}^{(1)} + \mathbf{y}^{(2)} \leq \mathbf{c} & \\ & \mathbf{c}_{\min}^{(i)} \leq \mathbf{y}^{(i)} \leq \mathbf{c}_{\max}^{(i)} & i = 1,2 \end{array} \tag{33}$$

where $U_i^\star\left(\mathbf{y}^{(i)}\right)$ is the optimal objective value of (32) for a given $\mathbf{y}^{(i)}$, with a subgradient given by the Lagrange multiplier $\boldsymbol{\lambda}^{(i)}$ associated to the constraints $\sum_{s\in S_i: l\in L(s)} x_s \leq y_l^{(i)}$ in (32). Observe that $\boldsymbol{\lambda}^{(i)}$ is the set of differential prices for the QoS class i. The master primal problem (33) can now be solved with a subgradient method by updating the aggregate rates as

$$\begin{bmatrix} \mathbf{y}^{(1)}(t+1) \\ \mathbf{y}^{(2)}(t+1) \end{bmatrix} = \left[\begin{bmatrix} \mathbf{y}^{(1)}(t) \\ \mathbf{y}^{(2)}(t) \end{bmatrix} + \alpha \begin{bmatrix} \boldsymbol{\lambda}^{\star(1)}\left(\mathbf{y}^{(1)}(t)\right) \\ \boldsymbol{\lambda}^{\star(2)}\left(\mathbf{y}^{(2)}(t)\right) \end{bmatrix} \right]_{\mathcal{Y}} \tag{34}$$

where $[\cdot]_{\mathcal{Y}}$ denotes the projection onto the feasible convex set $\mathcal{Y} \triangleq \{(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}) : \mathbf{y}^{(1)} + \mathbf{y}^{(2)} \leq \mathbf{c}, \mathbf{c}_{\min}^{(i)} \leq \mathbf{y}^{(i)} \leq \mathbf{c}_{\max}^{(i)} \text{ for } i = 1,2\}$. Nicely enough, this feasible set enjoys the property of naturally decomposing into a Cartesian product for each of the links: $\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_L$. Therefore, this subgradient update can be performed independently by each link simply with the knowledge of its corresponding Lagrange multipliers $\lambda_l^{(1)}$ and $\lambda_l^{(2)}$, which in turn are also updated independently by each link as in Section III-B.

Summarizing, we have the following algorithm.

Primal-Dual Algorithm to solve QoS rate allocation (31):

- Parameters: each source needs its utility U_s and each link its capacity c_l and rate limits for each class $c_{\min}^{(i)}$ and $c_{\max}^{(i)}$.
- Initialization: set $t = 0$ and $\lambda_l(0)$ equal to some nonnegative value for all l.

1) Solve basic NUMs in (32) with the canonical dual algorithm → this implies an iterative algorithm.
2) Each link updates its aggregate rate with the subgradient iterate (34) and broadcasts the new rates $\mathbf{y}^{(1)}(t+1)$ and $\mathbf{y}^{(2)}(t+1)$.
3) Set $t \leftarrow t+1$ and go to step 1 (until satisfying termination criterion).

Observe that there are two levels of decompositions (steps 1 and 2): on the highest level there is a master primal problem, on a second level there is a secondary master dual problem, and on the lowest level the subproblems. There is no explicit signaling required.

2) Partial Dual Decomposition: Consider now a dual decomposition of (31) by relaxing the flow constraints $\sum_{s\in S_i: l\in L(s)} x_s \leq y_l^{(i)}$. This problem decomposes into one maximization for each source, as (17) in the basic NUM, plus the following additional maximization to update the aggregate rates:

$$\begin{array}{lll} \underset{\mathbf{y}^{(1)},\mathbf{y}^{(2)}\geq \mathbf{0}}{\text{maximize}} & \boldsymbol{\lambda}^{(1)T}\mathbf{y}^{(1)} + \boldsymbol{\lambda}^{(2)T}\mathbf{y}^{(2)} & \\ \text{subject to} & \mathbf{y}^{(1)} + \mathbf{y}^{(2)} \leq \mathbf{c} & \\ & \mathbf{c}_{\min}^{(i)} \leq \mathbf{y}^{(i)} \leq \mathbf{c}_{\max}^{(i)} & i = 1,2 \end{array} \tag{35}$$

which can be solved independently by each link with knowledge of its corresponding Lagrange multipliers $\lambda_l^{(1)}$ and $\lambda_l^{(2)}$, which in turn are also updated independently by each link.

The master dual problem corresponding to this dual decomposition is updated with the following subgradient method (similarly to (19)):

$$\lambda_l^{(i)}(t+1)=\left[\lambda_l^{(i)}(t)-\alpha\left(y_l^{(i)}(t)-\sum_{s\in S_i:l\in L(s)} x_s^\star\left(\lambda^{(i)s}(t)\right)\right)\right]^+ \quad \forall l, i=1,2. \tag{36}$$

Summarizing, we have the following algorithm.

Partial-Dual Algorithm to solve QoS rate allocation (31):

- Parameters: each source needs its utility U_s and each link its capacity c_l and rate limits for each class $c_{\min}^{(i)}$ and $c_{\max}^{(i)}$.
- Initialization: set $t=0$ and $\lambda_l(0)$ equal to some nonnegative value for all l.

1) Each source locally solves its problem by computing (17) and broadcasts the solution $x_s^\star\left(\lambda^{(i)s}(t)\right)$.
2) Each link obtains the per-class capacities $\mathbf{y}^{(i)}$ by solving (35), updates its price for each class $\lambda_l^{(i)}$ with the subgradient iterate (36), and broadcasts the new price $\lambda_l^{(i)}(t+1)$.
3) Set $t \leftarrow t+1$ and go to step 1 (until satisfying termination criterion).

Observe that this approach contains only one level of decomposition and no explicit signaling is required.

3) Summary: In the primal-dual decomposition approach, each link updates the aggregate rates on a slower time scale and the prices on a faster time scale, whereas in the partial dual decomposition approach each link updates the prices on a slower time scale and the aggregate rates on a faster time scale (actually in one shot); therefore, the speed of convergence of the partial dual approach is likely to be faster in general. In both cases, the users are privy of the existence of classes and only the links have to take this into account by having one price for each class. In other words, this is a way to give each class of users a different price than the one based on the standard dual-based algorithm so that they can be further controlled. In the extreme case of one user per class, the difference between these two alternatives is similar to that between rate-based congestion control, such as XCP and RCP, and pricing-based congestion control, such as Reno and Vegas.

B. Decoupling of Coupled Objectives [21]

The majority of the utility problem formulations considered in the literature concern uncoupled utilities where the local variables corresponding to one node do not directly disturb the utilities of the other nodes. Systems with competition or cooperation, however, do not satisfy this assumption and the utilities are indeed coupled (see Fig. 4). An example of cooperation model can be found in networks where nodes form clusters and the utility obtained by each node depends on the rate allocated to others within the same cluster. An example of competition model is wireless power control and digital subscriber line (DSL) spectrum management of copper wires in a cable binder, where the utilities are functions of the signal-to-interference ratios (SIRs) that are dependent on the transmit powers of other users.

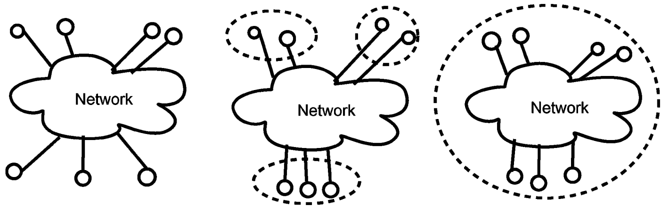

Fig. 4. Three network illustrations in terms of coupling. (a) All uncoupled utilities. (b) Partially coupled utilities within clusters. (c) Fully coupled utilities. (The dotted lines indicate coupling and where the consistency prices are exchanged.)

Consider the following NUM problem where each utility not only depends on the local variable but also on variables of other nodes:

$$\begin{array}{ll}\underset{\{\mathbf{x}_k\}}{\text{maximize}} & \sum_{k=1}^K U_k\left(\mathbf{x}_k, \{\mathbf{x}_l\}_{l\in\mathcal{L}(k)}\right)\\ \text{subject to} & \mathbf{x}_k \in \mathcal{X}_k \quad \forall k\\ & \sum_{k=1}^K \mathbf{g}_k(\mathbf{x}_k) \le \mathbf{c}\end{array} \tag{37}$$

where the (strictly concave) utilities U_k depend on a local vector variable $\mathbf{x}_k$ and on variables of other utilities $\mathbf{x}_l$ for $l \in \mathcal{L}(k)$ (i.e., coupled utilities), $\mathcal{L}(k)$ is the set of nodes coupled with the kth utility, the sets $\mathcal{X}_k$ are arbitrary convex sets representing local constraints, and the coupling constraining function $\sum_k \mathbf{g}_k(\mathbf{x}_k)$ is not necessarily linear, but still convex. Note that this model has two types of coupling: coupling constraints and coupled utilities.

The key idea to address coupled utilities is to introduce auxiliary variables and additional equality constraints, thus transferring the coupling in the objective function to coupling in the constraints, which can then be decoupled by dual decomposition and solved by introducing additional *consistency pricing*. It is reasonable to assume that if two nodes have their individual utilities dependent on each other's local variables, then there must be some communication channels in which they can locally exchange pricing messages. It turns out that the global link congestion price update of the standard dual algorithm is not affected by the local *consistency price* updates, which can be conducted via these local communication channels among the nodes.

We first introduce in problem (37) auxiliary variables $\mathbf{x}_{kl}$ for the coupled arguments in the utility functions and additional equality constraints to enforce consistency

$$\begin{array}{ll}\underset{\{\mathbf{x}_k\},\{\mathbf{x}_{kl}\}}{\text{maximize}} & \sum_{k} U_k\left(\mathbf{x}_k, \{\mathbf{x}_{kl}\}_{l\in\mathcal{L}(k)}\right)\\ \text{subject to} & \mathbf{x}_k \in \mathcal{X}_k \quad \forall k\\ & \sum_{k} \mathbf{g}_k(\mathbf{x}_k) \le \mathbf{c}\\ & \mathbf{x}_{kl} = \mathbf{x}_l \quad \forall k, l \in \mathcal{L}(k)\end{array} \tag{38}$$

where $\mathbf{x}_k$ and $\mathbf{x}_{kl}$ are local variables at the kth node. Next, to obtain a distributed algorithm, we take a dual decomposition approach

$$\begin{aligned} \underset{\{\mathbf{x}_k\},\{\mathbf{x}_{kl}\}}{\text{maximize}} \quad & \sum_k U_k\big(\mathbf{x}_k, \{\mathbf{x}_{kl}\}_{l\in\mathcal{L}(k)}\big) \\ & + \boldsymbol{\lambda}^T\Big(\mathbf{c} - \sum_k \mathbf{g}_k(\mathbf{x}_k)\Big) + \sum_{k,l\in\mathcal{L}(k)} \boldsymbol{\gamma}_{kl}^T(\mathbf{x}_l - \mathbf{x}_{kl}) \\ \text{subject to} \quad & \mathbf{x}_k \in \mathcal{X}_k \quad \forall k \\ & \mathbf{x}_{kl} \in \mathcal{X}_l \quad \forall k, l \in \mathcal{L}(k) \end{aligned} \tag{39}$$

where $\boldsymbol{\lambda}$ are the *link prices* and the $\boldsymbol{\gamma}_{kl}$'s are the *consistency prices*. By exploiting the decomposability structure of the Lagrangian, it separates into many subproblems where maximization is done using local variables (the kth subproblem uses *only* variables with the first subscript index k)

$$\begin{aligned} \underset{\mathbf{x}_k,\{\mathbf{x}_{kl}\}}{\text{maximize}} \quad & U_k\left(\mathbf{x}_k, \{\mathbf{x}_{kl}\}_{l\in\mathcal{L}(k)}\right) - \boldsymbol{\lambda}^T \mathbf{g}_k(\mathbf{x}_k) \\ & + \left(\sum_{l:k\in\mathcal{L}(l)} \boldsymbol{\gamma}_{lk}\right)^T \mathbf{x}_k - \sum_{l\in\mathcal{L}(k)} \boldsymbol{\gamma}_{kl}^T \mathbf{x}_{kl} \\ \text{subject to} \quad & \mathbf{x}_k \in \mathcal{X}_k \\ & \mathbf{x}_{kl} \in \mathcal{X}_l \quad \forall l \in \mathcal{L}(k) \end{aligned} \tag{40}$$

where $\{\mathbf{x}_{kl}\}_{l\in\mathcal{L}(k)}$ are auxiliary local variables for the kth node.

The optimal value of (39) for a given set of $\boldsymbol{\gamma}_{kl}$'s and $\boldsymbol{\lambda}$ defines the dual function $g(\{\boldsymbol{\gamma}_{kl}\}, \boldsymbol{\lambda})$. The dual problem is thus given by

$$\underset{\boldsymbol{\lambda},\{\boldsymbol{\gamma}_{kl}\}}{\text{minimize}} \quad g\left(\{\boldsymbol{\gamma}_{kl}\}, \boldsymbol{\lambda}\right) \quad \text{subject to} \quad \boldsymbol{\lambda} \geq \mathbf{0} \tag{41}$$

which can be solved with the following updates:

$$\boldsymbol{\lambda}(t+1) = \left[\boldsymbol{\lambda}(t) - \alpha\left(\mathbf{c} - \sum_k \mathbf{g}_k(\mathbf{x}_k)\right)\right]^+ \tag{42}$$

$$\boldsymbol{\gamma}_{kl}(t+1) = \boldsymbol{\gamma}_{kl}(t) - \alpha\left(\mathbf{x}_l(t) - \mathbf{x}_{kl}(t)\right), l \in L(k). \tag{43}$$

It is worthwhile noting that (41) is equivalent to

$$\underset{\boldsymbol{\lambda}}{\text{minimize}} \left(\underset{\{\boldsymbol{\gamma}_{kl}\}}{\text{minimize}} \quad g\left(\{\boldsymbol{\gamma}_{kl}\}, \boldsymbol{\lambda}\right)\right) \quad \text{subject to} \quad \boldsymbol{\lambda} \geq \mathbf{0}. \tag{44}$$

Solving the dual function [either (41) or (44)] is equivalent to solving the original problem.

Summarizing, we have the following two algorithms based on problems (41) and (44), respectively.

Full Dual Algorithm to solve coupled-objective NUM (37):

- Parameters: each node needs its utility U_k and constraint function $\mathbf{g}_k$, and each link its capacity c_l.
- Initialization: set $t = 0$, $\boldsymbol{\lambda}(0)$ equal to some nonnegative value, and the set $\{\boldsymbol{\gamma}_{kl}(0)\}$ equal to some value.

1) Each source locally solves its problem (40) and broadcasts the solution $\mathbf{x}_k^\star$ (not the auxiliary variables $\{\mathbf{x}_{kl}^\star\}$).
2) Price updating:
   i) Each link updates the link prices with the iterate in (42) and broadcasts the new price $\lambda_l(t+1)$.
   ii) Each node k updates the consistency prices with the iterate in (43), then broadcast the new prices $\boldsymbol{\gamma}_{kl}(t+1)$, for all l, to the coupled nodes within the cluster.
3) Set $t \leftarrow t+1$ and go to step 1 (until satisfying termination criterion).

Dual-Primal Algorithm to solve coupled-objective NUM (37):

- Parameters: each node needs its utility U_k and constraint function $\mathbf{g}_k$, and each link its capacity c_l.
- Initialization: set $t = 0$, $\boldsymbol{\lambda}(0)$ equal to some nonnegative value, and the set $\{\boldsymbol{\gamma}_{kl}(0)\}$ equal to some value.

1) Each source locally solves its problem (40) and broadcasts the solution $\mathbf{x}_k^\star$ (not the auxiliary variables $\{\mathbf{x}_{kl}^\star\}$).
2) Fast price updating: each node k updates the consistency prices with the iterate in (43), then broadcasts the new prices $\boldsymbol{\gamma}_{kl}(t+1)$, for all l, to the coupled nodes within the cluster.
3) Set $t \leftarrow t+1$ and go to step 1 (until satisfying termination criterion).
4) Slow price updating: each link updates the link prices with the iterate in (42) and broadcasts the new price $\lambda_l(t+1)$.
5) Go to step 1 (until satisfying termination criterion).

In the full-dual algorithm, the link prices and the consistency prices are simultaneously updated to solve (41) using a subgradient algorithm. In the dual-primal algorithm, however, the inner minimization of (44) is fully performed (by repeatedly updating the set of $\boldsymbol{\gamma}_{kl}$'s) for each update of $\boldsymbol{\lambda}$. This latter approach implies two time scales: a fast time scale in which each cluster updates the corresponding consistency prices and a slow time scale in which the network updates the link prices. In comparison, the former approach has just one time scale. The alternative of two time scales has an interest from a practical perspective since consistency prices can often be exchanged very quickly over local communication channels only by nodes that are coupled together.

Coupling Through Interference: Of particular interest is the case where the coupling between utilities is through an interference term that contains an additive combination of functions of the coupling variables

$$U_k\left(\mathbf{x}_k, \{\mathbf{x}_l\}_{l\in\mathcal{L}(k)}\right) = U_k(\mathbf{x}_k, \mathbf{i}_k) \tag{45}$$

where $\mathbf{i}_k = \sum_{l\in\mathcal{L}(k)} \mathbf{h}_{kl}(\mathbf{x}_l)$ and the $\mathbf{h}_{kl}$'s are convex functions. Note that, by definition of *interference*, each utility $U_k(\mathbf{x}_k, \mathbf{i}_k)$ is decreasing in $\mathbf{i}_k$. The interference term has a physical implication in practice where network nodes such as DSL modems already have the capability to locally measure the total interference from other competing network nodes.

The problem with auxiliary variables $\mathbf{i}_k$ for the coupled variables is

$$\begin{aligned} \underset{\{\mathbf{x}_k\},\{\mathbf{i}_k\}}{\text{maximize}} \quad & \sum_k U_k(\mathbf{x}_k, \mathbf{i}_k) \\ \text{subject to} \quad & \mathbf{x}_k \in \mathcal{X}_k \quad \forall k \\ & \sum_k \mathbf{g}_k(\mathbf{x}_k) \leq \mathbf{c} \\ & \mathbf{i}_k \geq \sum_{l \in \mathcal{L}(k)} \mathbf{h}_{kl}(\mathbf{x}_l) \quad \forall k \end{aligned} \tag{46}$$

where the interference inequality constraint is satisfied with equality at an optimal point since utilities are decreasing in the interference term. Note that if the functions $\mathbf{h}_{kl}$ are linear, then the interference inequality constraints can be substituted by equality constraints. The only modification to our earlier algorithm is that the update of the consistency prices in (43) is replaced by

$$\boldsymbol{\gamma}_k(t+1) = \left[\boldsymbol{\gamma}_k(t) - \alpha \left(\mathbf{i}_k - \sum_{l \in \mathcal{L}(k)} \mathbf{h}_{kl}(\mathbf{x}_l) \right) \right]^+ \tag{47}$$

which can be done at the kth node with knowledge of the local variables and of the linear combination of the coupling variables from other nodes.

By leveraging the structure of the interference term, only one consistency price is needed for each interference term (which may contain many coupled variables), substantially reducing the amount of signaling. Indeed, message passing overhead, measured by the number of consistency prices to update in (43) and (47), is of the order $O(N^2)$ and $O(N)$, respectively, where N is the number of nodes in a cluster.

C. *Decoupling for Coupled Constraint Sets [29]*

When the coupled constraint sets do not seem to be readily primal or dual decomposable, reparametrization may reveal hidden decomposability structures. In this subsection, we briefly outline a recent example on an important problem in wireless networks.

In cellular wireless networks, uplink power control is an important mechanism to control interference. In the early to mid 1990s, distributed power control for fixed SIR assignment was extensively studied (e.g., [24]), from the classical received-power equalization in code-division multiple-access (CDMA) systems to the Foschini–Miljanic power control [25] and its many extensions. Then, motivated by CDMA-based third-generation (3G) wireless data networks, researchers studied joint optimization of SIR assignment (or, in general, QoS assignment) and power control, where optimality is with respect to a total utility function of the SIRs over a feasible SIR region. This utility maximization over the complicated, coupled constraint set of feasible QoS assignments is illustrated in Fig. 5. Distributed algorithms were proposed to attain Nash equilibrium (e.g., [26]), which can be socially suboptimal. Optimal algorithms were proposed [27], [28] but required centralized computation. A major bottleneck is that the solution reduces to computing Perron–Frobenius eigenvectors of a certain matrix defined by the channel gains, an operation that is not readily distributed.

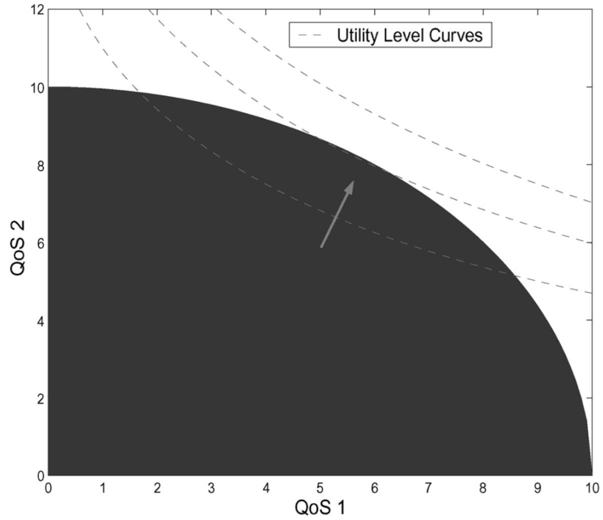

Fig. 5. A fundamental problem of joint QoS and power control in cellular uplinks: maximizing concave utilities, whose level curves are shown, over a convex set of feasible QoS region. Here, QoS metrics are functions of SIR.

This open problem of distributed, optimal SIR assignment and power control has recently been solved in [29], where the key insight is to reparameterize the coupling constraint set through left, rather than right, eigenvectors, which reveals a hidden decomposability structure and leads to a dual decomposition. The basic problem setup and algorithm is outlined in the rest of this subsection.

Consider uplink transmissions in a multicellular wireless data network where each uplink is indexed by i and maintains a strictly concave utility function U_i of SIR. Each mobile station i is served by a base station σ_i and each base station serves a set S_k of mobile stations. Let h_{ij} denote the path loss (channel gain) from the transmitter of logical link j to the receiver of logical link i, and $\mathbf{G}$ be a matrix where the diagonal entries are 0 and the (i,j) off-diagonal entry is h_{ij}/h_{ii}. Let the transmit power of link i be p_i and the receiver noise be η_i. The SIR on link i is defined as $\gamma_i = p_i G_{ii}/(\sum_{j \neq i} p_j G_{ij} + \eta_i)$. Let $\mathbf{D}(\boldsymbol{\gamma})$ be a diagonal matrix with entry $D_{ii} = \gamma_i$. It turns out that the problem of maximizing network utility over the feasible SIR region can be stated as the following convex optimization:

$$\begin{aligned} \underset{\{p_i,\gamma_i\}}{\text{maximize}} \quad & \sum_i U_i(\gamma_i) \\ \text{subject to} \quad & \rho\left(\mathbf{D}(\boldsymbol{\gamma})\mathbf{G}\right) \leq 1 \end{aligned} \tag{48}$$

where ρ denotes the Perron–Frobenius eigenvalue (the largest modulus eigenvalue) of the positive matrix $\mathbf{D}(\boldsymbol{\gamma})\mathbf{G}$. The constraint set can be verified to be convex but is clearly coupled in a way that is not readily primal or dual decomposable.

Fortunately, we can "change coordinates" through a reparametrization of the feasible SIR region by the left eigenvectors $(\mathbf{s}, \mathbf{r})$ of $\mathbf{D}(\boldsymbol{\gamma})\mathbf{G}$ and $\mathbf{G}\mathbf{D}(\boldsymbol{\gamma})$ (which have interpretations of load and spillage vectors), rather than the right eigenvectors of $\mathbf{D}(\boldsymbol{\gamma})\mathbf{G}$ and $\mathbf{G}\mathbf{D}(\boldsymbol{\gamma})$ (which have interpretations of power and interference vectors). This new description of the coupled constraint set leads to the following dual-based distributed algorithm that only requires limited message passing between each base station and the mobile stations it serves.

Dual Algorithm to solve joint SIR and power control (48):

- Parameters: step-size $\delta > 0$ and utility functions $\{U_i(\gamma_i)\}$.
- Initialization: arbitrary $\mathbf{s}[0] \succ 0$.

1) BS k broadcasts the BS-load factor $\ell_k[t] = \sum_{i \in S_k} s_i[t]$.
2) Compute the spillage-factor $r_i[t]$ according to $r_i = \alpha \sum_{j \neq i, j \in S_{\sigma_i}} s_j + \sum_{k \neq \sigma_i} h_{ki} \ell_k$.
3) Assign SIR values $\gamma_i[t] = \rho s_i[t]/r_i[t]$.
4) Measure the resulting interference $q_i[t]$.
5) Update the load factor $s_i[t]$ in the ascent direction given by: $\Delta s_i = (U_i'(\gamma_i)\gamma_i/q_i) - s_i$
$$s_i[t+1] = s_i[t] + \delta \Delta s_i[t].$$
6) Set $t \leftarrow t+1$ and go to step 1 (until satisfying termination criterion).

Proof of convergence of the above algorithm to global optimum (under certain curvature and fairness conditions on the utility functions) and extensions to related problem formulations can be found in [29].

V. Conclusion

The importance of "decomposability" to distributed solutions is similar to that of "convexity" to efficient computation of global optimum. Recognizing and utilizing decomposition methods help network designers to systematically compare modularized architectures and to distributively optimize resource allocation. As surveyed in this tutorial paper, one can employ different combinations of the basic "building blocks" of primal and dual decompositions, search among the alternatives of distributed algorithms, and try to reveal hidden decomposability structures through different representations of the same optimization problem. Some of these efforts still depend on trial-and-error rather than a systematic methodology, and the analysis of some important attributes such as rate of convergence still lack a fully developed theoretical foundation. Towards a systematic comparison of alternative decompositions, a key challenge is to study the metrics for such comparison that are not easily quantified or do not have obvious ordering relationships.

Acknowledgment

The authors would like to thank all the collaborators on the subject of decomposition methods and distributed algorithms for network utility maximization, especially, coauthors M. Bresler, R. Calderbank, J. Doyle, P. Hande, J. He, S. Low, S. Rangan, J. Rexford, and C. W. Tan, whose joint work with us form parts of the materials in Section IV. They are also grateful for discussions on this subject with many colleagues, especially recent ones with F. P. Kelly, X. Lin, S. Shakkottai, N. B. Shroff, R. Srikant, and J. Zhang.

References

[1] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," *Proc. IEEE*, Dec. 2006, to be published.

[2] M. Chiang, "Balancing transport and physical layers in wireless multihop networks: Jointly optimal congestion control and power control," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 104–116, Jan. 2005.

[3] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE J. Sel. Area Commun.*, vol. 24, no. 8, pp. 1452–1463, Aug. 2006.

[4] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[5] J. L. R. Ford and D. R. Fulkerson, *Flow in Networks*. Princeton, NJ: Princeton Univ. Press, 1962.

[6] D. P. Bertsekas and R. G. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[7] L. S. Lasdon, *Optimization Theory for Large Systems*. New York: Macmillian, 1970.

[8] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 1999.

[9] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[10] Z.-Q. Luo and W. Yu, "An introduction to convex optimization for communications and signal processing," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1426–1438, Aug. 2006.

[11] N. Z. Shor, *Minimization Methods for Non-Differentiable Functions*. Berlin, Germany: Springer-Verlag, 1985.

[12] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex Analysis and Optimization*. Belmont, MA: Athena Scientific, 2003.

[13] R. Srikant, *The Mathematics of Internet Congestion Control*. Cambridge, MA: Birkhauser, 2004.

[14] S. H. Low, "A duality model of TCP and queue management algorithms," *IEEE/ACM Trans. Netw.*, vol. 11, no. 4, pp. 525–536, Aug. 2003.

[15] J. W. Lee, M. Chiang, and R. A. Calderbank, "Price-based distributed algorithm for optimal rate-reliability tradeoff in network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 5, pp. 962–976, May 2006.

[16] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 2006.

[17] X. Wang and K. Kar, "Cross-layer rate control for end-to-end proportional fairness in wireless networks with random access," in *Proc. ACM Mobihoc*, 2005, pp. 157–168.

[18] J. W. Lee, M. Chiang, and R. A. Calderbank, "Jointly optimal congestion and contention control in wireless ad hoc networks," *IEEE Commun. Lett.*, vol. 10, no. 3, pp. 216–218, Mar. 2006.

[19] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, no. 3, pp. 237–252, Mar. 1998.

[20] M. Bresler, "Distributed approaches to maximizing network utilities," B.A. thesis, Princeton Univ., Princeton, NJ, May 2006.

[21] C. W. Tan, D. P. Palomar, and M. Chiang, "Distributed optimization of coupled systems with applications to network utility maximization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Toulouse, France, May 14–19, 2006, pp. 981–984.

[22] R. T. Rockafellar, "Saddle-points and convex analysis," in *Differential Games and Related Topics*, H. W. Kuhn and G. P. Szego, Eds. Amsterdam, The Netherlands: North-Holland, 1971.

[23] D. P. Palomar and M. Chiang, "Alternative decompositions for distributed maximization of network utility: Framework and applications," in *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 23–29, 2006.

[24] R. D. Yates, "A framework for uplink power control in cellular radio systems," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 7, pp. 1341–1347, Sep. 1995.

[25] G. J. Foschini and Z. Miljanic, "A simple distributed autonomous power control algorithm and its convergence," *IEEE Trans. Veh. Technol.*, vol. VT-42, no. 4, pp. 641–646, Apr. 1993.

[26] C. Saraydar, N. Mandayam, and D. Goodman, "Pricing and power control in a multicell wireless data network," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 10, pp. 1883–1892, Oct. 2001.

[27] M. Chiang, "Balancing supply and demand of bandwidth in cellular networks: A utility maximization over powers and rates," in *Proc. IEEE INFOCOM*, Hong Kong, China, Mar. 2004.

[28] D. O'Neill, D. Julian, and S. Boyd, "Adaptive management of network resources," in *Proc. IEEE Veh. Technol. Conf.*, Oct. 2003, pp. 1929–1933.

[29] P. Hande, S. Rangan, and M. Chiang, "Distributed uplink power control for optimal SIR assignment in cellular data networks," in *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 2006.

Daniel P. Palomar (S'99–M'03) received the electrical engineering and Ph.D. degrees from the Technical University of Catalonia (UPC), Barcelona, Spain, in 1998 and 2003, respectively.

Since March 2006, he has been a Research Associate at Princeton University, Princeton, NJ. During 1998, he was with the Department of Electronic Engineering, King's College London (KCL), London, U.K. From January 1999 to December 2003, he was a Research Assistant with the Department of Signal Theory and Communications, UPC. From April to November 2001, he held a visiting research appointment at the Department of Electrical Engineering, Stanford University, Stanford, CA. From January to December 2002, he was a Visiting Researcher at the Telecommunications Technological Center of Catalonia (CTTC), Barcelona. From August to November 2003, he was a Guest Researcher at the Department of Signals, Sensors, and Systems, Royal Institute of Technology (KTH), Stockholm, Sweden. From November 2003 to February 2004, he was a Visiting Researcher at the INFOCOM Department, University of Rome "La Sapienza," Rome, Italy. From March 2004 to February 2006, he was a Fulbright Research Fellow at Princeton University. In 2005, he held a position as a Visiting Professor at UPC.

Dr. Palomar received the 2004 Young Author Best Paper Award by the IEEE Signal Processing Society, the 2002/2003 Best Ph.D. Prize within the area of Information Technologies and Communications by the Technical University of Catalonia (UPC), the 2002/2003 Rosina Ribalta first prize for the Best Doctoral Thesis within the areas of Information Technologies and Communications by the Epson Foundation; and the 2004 prize for the best Doctoral Thesis in Advanced Mobile Communications by the Vodafone Foundation and COIT. He has also been awarded with a Fulbright Research Fellowship.

Mung Chiang (S'00–M'03) received the B.S. (Hon.) degree in electrical engineering and mathematics, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1999, 2000, and 2003, respectively.

He is an Assistant Professor of Electrical Engineering, and an affiliated faculty of Applied and Computational Mathematics at Princeton University, Princeton, NJ. He conducts research in the areas of optimization of communication systems, analytic foundations of network architectures, algorithms in broadband access, and information theory.

Prof. Chiang has been awarded a Hertz Foundation Fellow and received the Stanford University School of Engineering Terman Award, the SBC Communications New Technology Introduction Contribution Award, the National Science Foundation CAREER Award, and the Princeton University Howard B. Wentz Junior Faculty Award. He is the Lead Guest Editor of the IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATIONS (Special Issue on Nonlinear Optimization of Communication Systems), a Guest Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY and IEEE/ACM TRANSACTIONS ON NETWORKING (Joint Special Issue on Networking and Information Theory), and the Program Co-Chair of the 38th Conference on Information Sciences and Systems.