Chapter 5
Nonconvex Optimization for Communication Networks

Mung Chiang

Summary. Nonlinear convex optimization has provided both an insightful modeling language and a powerful solution tool to the analysis and design of communication systems over the last decade. A main challenge today is on nonconvex problems in these applications. This chapter presents an overview on some of the important nonconvex optimization problems in communication networks. Four typical applications are covered: Internet congestion control through nonconcave network utility maximization, wireless network power control through geometric and sigmoidal programming, DSL spectrum management through distributed nonconvex optimization, and Internet intradomain routing through nonconvex, nonsmooth optimization. A variety of nonconvex optimization techniques are showcased: sum-of-squares programming through successive SDP relaxation, signomial programming through successive GP relaxation, leveraging specific structures in these engineering problems for efficient and distributed heuristics, and changing the underlying protocol to enable a different problem formulation in the first place. Collectively, they illustrate three alternatives of tackling nonconvex optimization for communication networks: going "through" nonconvexity, "around" nonconvexity, and "above" nonconvexity.

Key words: Digital subscriber line, duality, geometric programming, Internet, network utility maximization, nonconvex optimization, power control, routing, semidefinite programming, sum of squares, TCP/IP, wireless network

Mung Chiang
Electrical Engineering Department, Princeton University, Princeton, NJ 08544, U.S.A.
e-mail: chiangm@princeton.edu

D.Y. Gao, H.D. Sherali, (eds.), *Advances in Applied Mathematics and Global Optimization*
Advances in Mechanics and Mathematics 17, DOI 10.1007/978-0-387-75714-8_5,

5.1 Introduction

There have been two major "waves" in the history of optimization theory and its applications: the first started with linear programming (LP) and the simplex method in the late 1940s, and the second with convex optimization and the interior point method in the late 1980s. Each has been followed by a transforming period of "appreciation-application cycle": as more people appreciate the use of LP/convex optimization, more look for their formulations in various applications; then more work on its theory, efficient algorithms, and software; the more powerful the tools become, and in turn more people appreciate its usage. Communication systems benefit significantly from both waves; the vast array of many success stories includes multicommodity flow solutions (e.g., Bellman–Ford algorithm) from LP, and network utility maximization and robust transceiver design from convex optimization.

Much of the current research is about the potential of the third wave, on nonconvex optimization. If one word is used to differentiate between easy and hard problems, convexity is probably the "watershed." But if a longer description length is allowed, useful conclusions can be drawn even for nonconvex optimization. Indeed, convexity is a very disturbing watershed, because it is not a topological invariant under change of variable (e.g., see geometric programming) or higher-dimension embedding (e.g., see sum of squares method). A variety of approaches has been proposed to tackle nonconvex optimization problems: from successive convex approximation to dualization, from nonlinear transformation to turn an apparently nonconvex problem into a convex problem to characterization of attraction regions and systematically jumping out of a local optimum, and from leveraging the specific structures of the problems (e.g., difference of convex functions, concave minimization, low rank nonconvexity) to developing more efficient branch-and-bound procedures.

Researchers in communications and networking have been examining nonconvex optimization using domain-specific structures in important problems in the areas of wireless networking, Internet engineering, and communication theory. Perhaps four typical topics best illustrate the variety of challenging issues arising from nonconvex optimization in communication systems:

- Nonconvex objective to be minimized. An example is congestion control for inelastic application traffic, where a nonconcave utility function needs to be maximized.
- Nonconvex constraint set. An example is power control in the low SIR regime.
- Integer constraints. Two important examples are single path routing and multiuser detection.
- Constraint sets that are convex but require an exponential number of inequalities to explicitly describe. An example is optimal scheduling in multihop wireless networks under certain interference models. The problem of wireless scheduling will not be discussed in this chapter. Interested readers can refer to [73] for a unifying framework of the problem.

This chapter overviews the latest results in recent publications about the first two topics, with a particular focus on showing the connections between the engineering intuitions about important problems in communication networks and the state-of-the-art algorithms in nonconvex optimization theory. Most of the results surveyed here were obtained in 2005–2006, and the problems driven by fundamental issues in the Internet, wireless, and broadband access networks. As this chapter illustrates, even after much progress made in recent years, there are still many challenging mysteries to be resolved on these important nonconvex optimization problems.

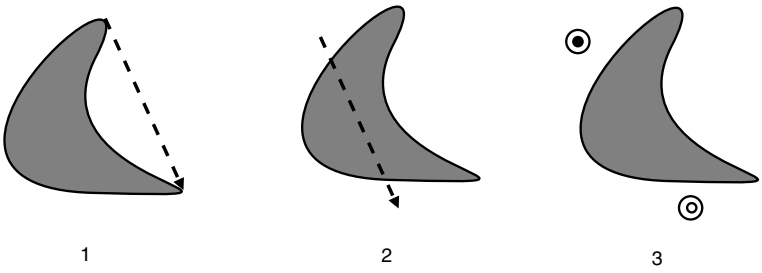

Fig. 5.1 Three major types of approaches when tackling nonconvex optimization problems in communication networks: Go (1) through, (2) around, or (3) above nonconvexity.

It is interesting to point out that, as illustrated in Figure 5.1, there are at least three very different approaches to tackle the difficult issue of nonconvexity.

- *Go "through" nonconvexity.* In this approach, we try to solve the difficult nonconvex problem; for example, we may use successive convex relaxations (e.g., sum-of-squares, signomial programming), utilize special structures in the problem (e.g., difference of convex functions, generalized quasiconcavity), or leverage smarter branch and bound methods.
- *Go "around" nonconvexity.* In this approach, we try to avoid solving the convex problem; for example, we may discover a change of variables that turns the seemingly nonconvex problem into a convex one, determine conditions under which the problem is convex or the KKT point is unique, or make approximations to make the problem convex.
- *Go "above" nonconvexity.* In this approach, we try to reformulate the nonconvex problem in the first place to make it more "solvable" or "approximately solvable." We observe that optimization problem formulations are induced by some underlying assumptions on what the network architectures and protocols should look like. By changing these assumptions, a different, much easier-to-solve or easier-to-approximate formulations may result. We refer to this approach as *design for optimizability*, which is concerned with redrawing architectures to make the resulting optimization

problem easier to solve. This approach of changing a hard problem into an easier one is in contrast to *optimization*, which tries to solve a given, possibly difficult, problem.

The four topics chosen in this chapter span a range of application contexts and tasks in communication networks. The sources of difficulty in these nonconvex optimization problems are summarized in Table 5.1, together with the key ideas in solving them and the type of approaches used. For more details beyond this brief overview chapter, please refer to the related publications [29, 19, 14, 35, 7, 70, 71] by the author and coworkers and the references therein.

Table 5.1 Summary of four nonconvex optimization problems in this chapter

Section	*Application*	*Task*	*Difficulty*	*Solution*	*Approach*
5.2	Internet	Congestion control	Nonconcave U	Sum of squares	"Through"
5.3	Wireless	Power control	Posynomial ratio	Geometric program	"Around"
5.4	DSL	Spectrum management	Posynomial ratio	Problem structure	"Around"
5.5	Internet	Routing	Nonconvex constraint	Approximation	"Above"

5.2 Internet Congestion Control

5.2.1 Introduction

Basic Network Utility Maximization

Since the publication of the seminal paper [37] by Kelly, Maulloo, and Tan in 1998, the framework of network utility maximization (NUM) has found many applications in network rate allocation algorithms and Internet congestion control protocols (e.g., surveyed in [45, 60]). It has also led to a systematic understanding of the entire network protocol stack in the unifying framework of "layering as optimization decomposition" (e.g., surveyed in [13, 49, 44]). By allowing nonlinear concave utility objective functions, NUM substantially expands the scope of the classical LP-based network flow problems.

Consider a communication network with L links, each with a fixed capacity of c_l bps, and S sources (i.e., end-users), each transmitting at a source rate of x_s bps. Each source s emits one flow, using a fixed set $L(s)$ of links in its path, and has a utility function $U_s(x_s)$. Each link l is shared by a set $S(l)$ of

sources. Network utility maximization, in its basic version, is the following problem of maximizing the total utility of the network $\sum_s U_s(x_s)$, 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:

$$\begin{aligned} &\text{maximize } \textstyle\sum_s U_s(x_s) \\ &\text{subject to } \textstyle\sum_{s\in S(l)} x_s \leq c_l, \ \ \forall l, \\ &\qquad\qquad \mathbf{x} \succeq 0, \end{aligned} \tag{5.1}$$

where the variables are $\mathbf{x} \in \mathbf{R}^S$.

There are many nice properties of the basic NUM model due to several simplifying assumptions of the utility functions and flow constraints, which provide the mathematical tractability of problem (5.1) but also limit its applicability. In particular, the utility functions $\{U_s\}$ are often assumed to be increasing and strictly concave functions.

Assuming that $U_s(x_s)$ becomes concave for large enough x_s is reasonable, because the law of diminishing marginal utility eventually will be effective. However, U_s may not be concave throughout its domain. In his seminal paper in 1995, Shenker [57] differentiated inelastic network traffic from elastic traffic. Utility functions for elastic traffic were modeled as strictly concave functions. Although inelastic flows with nonconcave utility functions represent important applications in practice, they have received little attention and rate allocation among them has only a limited mathematical foundation. There have been three recent publications [41, 29, 19] (see also earlier work in [69, 42, 43] related to the approach in [41]) on this topic.

In this section, we investigate the extension of the basic NUM to maximization of nonconcave utilities, as in the approach of [19]. We provide a centralized algorithm for offline analysis and establishment of a performance benchmark for nonconcave utility maximization when the utility function is a polynomial or signomial. Based on the semialgebraic approach to polynomial optimization, we employ convex sum-of-squares (SOS) relaxations solved by a sequence of semidefinite programs (SDP), to obtain increasingly tighter upper bounds on total achievable utility for polynomial utilities. Surprisingly, in all our experiments, a very low-order and often a minimal-order relaxation yields not just a bound on attainable network utility, but the globally maximized network utility. When the bound is exact, which can be proved using a sufficient test, we can also recover a globally optimal rate allocation.

Canonical Distributed Algorithm

A reason that the assumption of a utility function's concavity is upheld in many papers on NUM is that it leads to three highly desirable mathematical properties of the basic NUM:

- It is a convex optimization problem, therefore the global minimum can be computed (at least in centralized algorithms) in worst-case polynomial-time complexity [4].
- Strong duality holds for (5.1) and its Lagrange dual problem. A zero duality gap enables a dual approach to solve (5.1).
- Minimization of a separable objective function over linear constraints can be conducted by distributed algorithms based on the dual approach.

Indeed, the basic NUM (5.1) is such a "nice" optimization problem that its theoretical and computational properties have been well studied since the 1960s in the field of monotropic programming (e.g., as summarized in [54]). For network rate allocation problems, a dual-decomposition-based distributed algorithm has been widely studied (e.g., in [37, 45]), and is summarized below.

Zero duality gap for (5.1) states that solving the Lagrange dual problem is equivalent to solving the primal problem (5.1). The Lagrange dual problem is readily derived. We first form the Lagrangian of (5.1):

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \sum_s U_s(x_s) + \sum_l \lambda_l \left(c_l - \sum_{s \in S(l)} x_s \right),$$

where $\lambda_l \geq 0$ is the Lagrange multiplier (can be interpreted as the link congestion price) associated with the linear flow constraint on link l. Additivity of total utility and linearity of flow constraints lead to a Lagrangian dual decomposition into individual source terms:

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

where $\lambda^s = \sum_{l \in L(s)} \lambda_l$. For each source s, $L_s(x_s, \lambda^s) = U_s(x_s) - \lambda^s x_s$ only depends on local x_s and the link prices λ_l on those links used by source s.

The Lagrange dual function $g(\boldsymbol{\lambda})$ is defined as the maximized $L(\mathbf{x}, \boldsymbol{\lambda})$ over $\mathbf{x}$. This "net utility" maximization obviously can be conducted distributively by each source, as long as the aggregate link price $\lambda^s = \sum_{l \in L(s)} \lambda_l$ is available to source s, where source s maximizes a strictly concave function $L_s(x_s, \lambda^s)$ over x_s for a given λ^s:

$$x_s^*(\lambda^s) = \operatorname{argmax} \left[U_s(x_s) - \lambda^s x_s \right], \quad \forall s. \tag{5.2}$$

The Lagrange dual problem is

$$\begin{aligned} &\text{minimize} \quad g(\boldsymbol{\lambda}) = L(\mathbf{x}^*(\boldsymbol{\lambda}), \boldsymbol{\lambda}) \\ &\text{subject to } \boldsymbol{\lambda} \succeq 0, \end{aligned} \tag{5.3}$$

where the optimization variable is $\boldsymbol{\lambda}$. Any algorithms that find a pair of primal–dual variables $(\mathbf{x}, \boldsymbol{\lambda})$ that satisfy the KKT optimality condition would solve (5.1) and its dual problem (5.3). One possibility is a distributed, iterative subgradient method, which updates the dual variables $\boldsymbol{\lambda}$ to solve the dual problem (5.3):

$$\lambda_l(t+1) = \left[\lambda_l(t) - \alpha(t)\left(c_l - \sum_{s\in S(l)} x_s(\lambda^s(t))\right)\right]^+, \quad \forall l, \tag{5.4}$$

where t is the iteration number and $\alpha(t) > 0$ are step sizes. Certain choices of step sizes, such as $\alpha(t) = \alpha_0/t,\ \alpha_0 > 0$, guarantee that the sequence of dual variables $\boldsymbol{\lambda}(t)$ will converge to the dual optimal $\boldsymbol{\lambda}^*$ as $t \to \infty$. The primal variable $\mathbf{x}(\boldsymbol{\lambda}(t))$ will also converge to the primal optimal variable $\mathbf{x}^*$. For a primal problem that is a convex optimization, the convergence is towards the global optimum.

The sequence of the pair of algorithmic steps (5.2, 5.4) forms what we refer to as the *canonical distributed algorithm*, which solves the network utility optimization problem (5.1) and the dual (5.3) and computes the optimal rates $\mathbf{x}^*$ and link prices $\boldsymbol{\lambda}^*$.

Nonconcave Network Utility Maximization

It is known that for many multimedia applications, user satisfaction may assume a nonconcave shape as a function of the allocated rate. For example, the utility for voice applications is better described by a sigmoidal function: with a convex part at low rate and a concave part at high rate, and a single inflexion point x^0 (with $U_s''(x^0) = 0$) separating the two parts. Furthermore, in some other models of utility functions, the concavity assumption on U_s is also related to the elasticity assumption on rate demands by users. When demands for x_s are not perfectly elastic, $U_s(x_s)$ may not be concave.

Suppose we remove the critical assumption that $\{U_s\}$ are concave functions, and allow them to be any nonlinear functions. The resulting NUM becomes nonconvex optimization and significantly harder to be analyzed and solved, even by centralized computational methods. In particular, a local optimum may not be a global optimum and the duality gap can be strictly positive. The standard distributive algorithms that solve the dual problem may produce infeasible or suboptimal rate allocation.

There have been several recent publications on distributed algorithms for nonconcave utility maximization. In [41], a "self-regulation" heuristic is proposed to avoid the resulting oscillation in rate allocation and is shown to converge to an optimal rate allocation asymptotically when the proportion of nonconcave utility sources vanishes. In [29], a set of sufficient conditions and necessary conditions is presented under which the canonical distributed algo-

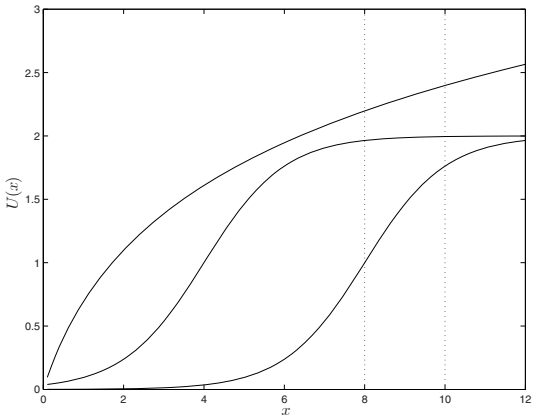

Fig. 5.2 Some examples of utility functions $U_s(x_s)$: it can be concave or sigmoidal as shown in the graph, or any general nonconcave function. If the bottleneck link capacity used by the source is small enough, that is, if the dotted vertical line is pushed to the left, a sigmoidal utility function effectively becomes a convex utility function.

rithm still converges to the globally optimal solution. However, these conditions may not hold in many cases. These two approaches illustrate the choice between admission control and capacity planning to deal with nonconvexity (see also the discussion in [36]). But neither approach provides a theoretically polynomial-time and practically efficient algorithm (distributed or centralized) for nonconcave utility maximization.

In [19], using a family of convex semidefinite programming (SDP) relaxations based on the sum-of-squares (SOS) relaxation and the positivstellensatz theorem in real algebraic geometry, we apply a centralized computational method to bound the total network utility in polynomial time. A surprising result is that for all the examples we have tried, wherever we could verify the result, the tightest possible bound (i.e., the globally optimal solution) of NUM with nonconcave utilities is computed with a very low-order relaxation. This efficient numerical method for offline analysis also provides the benchmark for distributed heuristics.

These three different approaches: proposing distributed but suboptimal heuristics (for sigmoidal utilities) in [41], determining optimality conditions for the canonical distributed algorithm to converge globally (for all nonlinear utilities) in [29], and proposing an efficient but centralized method to compute the global optimum (for a wide class of utilities that can be transformed into polynomial utilities) in [19] (and this section), are complementary in the study of distributed rate allocation by nonconcave NUM.

5.2.2 Global Maximization of Nonconcave Network Utility

Sum-of-Squares Method

We would like to bound the maximum network utility by γ in polynomial time and search for a tight bound. Had there been no link capacity constraints, maximizing a polynomial is already an NP-hard problem, but can be relaxed into an SDP [58]. This is because testing if the following bounding inequality holds $\gamma \geq p(\mathbf{x})$, where $p(\mathbf{x})$ is a polynomial of degree d in n variables, is equivalent to testing the positivity of $\gamma - p(\mathbf{x})$, which can be relaxed into testing if $\gamma - p(\mathbf{x})$ can be written as a sum of squares (SOS): $p(\mathbf{x}) = \sum_{i=1}^{r} q_i(\mathbf{x})^2$ for some polynomials q_i, where the degree of q_i is less than or equal to $d/2$. This is referred to as the SOS relaxation. If a polynomial can be written as a sum of squares, it must be nonnegative, but not vice versa. Conditions under which this relaxation is tight have been studied since Hilbert. Determining if a sum of squares decomposition exists can be formulated as an SDP feasibility problem, thus polynomial-time solvable.

Constrained nonconcave NUM can be relaxed by a generalization of the Lagrange duality theory, which involves nonlinear combinations of the constraints instead of linear combinations in the standard duality theory. The key result is the positivstellensatz, due to Stengle [62], in real algebraic geometry, which states that for a system of polynomial inequalities, either there exists a solution in $\mathbf{R}^n$ or there exists a polynomial which is a certificate that no solution exists. This infeasibility certificate has recently been shown to be also computable by an SDP of sufficient size [51, 50], a process that is referred to as the sum-of-squares method and automated by the software SOSTOOLS [52] initiated by Parrilo in 2000. For a complete theory and many applications of SOS methods, see [51] and references therein.

Furthermore, the bound γ itself can become an optimization variable in the SDP and can be directly minimized. A nested family of SDP relaxations, each indexed by the degree of the certificate polynomial, is guaranteed to produce the exact global maximum. Of course, given the problem is NP-hard, it is not surprising that the worst-case degree of certificate (thus the number of SDP relaxations needed) is exponential in the number of variables. What is interesting is the observation that in applying SOSTOOLS to nonconcave utility maximization, a very low-order, often the minimum-order relaxation already produces the globally optimal solution.

Application of SOS Method to Nonconcave NUM

Using sum-of-squares and the positivstellensatz, we set up the following problem whose objective value converges to the optimal value of problem (5.1),

where $\{U_i\}$ are now general polynomials, as the degree of the polynomials involved is increased.

$$\begin{array}{l}\text{minimize} \quad \gamma \\ \text{subject to} \\ \gamma - \sum_s U_s(x_s) - \sum_l \lambda_l(\mathbf{x})(c_l - \sum_{s\in S(l)} x_s) \\ - \sum_{j,k} \lambda_{jk}(\mathbf{x})(c_j - \sum_{s\in S(j)} x_s)(c_k - \sum_{s\in S(k)} x_s) - \\ \ldots - \lambda_{12\ldots n}(\mathbf{x})(c_1 - \sum_{s\in S(1)} x_s)\ldots(c_n - \sum_{s\in S(n)} x_s) \\ \text{is SOS,} \\ \lambda_l(\mathbf{x}), \lambda_{jk}(\mathbf{x}), \ldots, \lambda_{12\ldots n}(\mathbf{x}) \text{ are SOS.}\end{array} \tag{5.5}$$

The optimization variables are γ and all of the coefficients in polynomials $\lambda_l(\mathbf{x})$, $\lambda_{jk}(\mathbf{x})$, ..., $\lambda_{12\ldots n}(\mathbf{x})$. Note that $\mathbf{x}$ is not an optimization variable; the constraints hold for all $\mathbf{x}$, therefore imposing constraints on the coefficients. This formulation uses Schmüdgen's representation of positive polynomials over compact sets [56].

Let D be the degree of the expression in the first constraint in (5.5). We refer to problem (5.5) as the SOS relaxation of order D for the constrained NUM. For a fixed D, the problem can be solved via SDP. As D is increased, the expression includes more terms, the corresponding SDP becomes larger, and the relaxation gives tighter bounds. An important property of this nested family of relaxations is guaranteed convergence of the bound to the global maximum.

Regarding the choice of degree D for each level of relaxation, clearly a polynomial of odd degree cannot be SOS, so we need to consider only the cases where the expression has even degree. Therefore, the degree of the first nontrivial relaxation is the largest even number greater than or equal to degree $\sum_s U_s(x_s)$, and the degree is increased by 2 for the next level.

A key question now becomes: how do we find out, after solving an SOS relaxation, if the bound happens to be exact? Fortunately, there is a sufficient test that can reveal this, using the properties of the SDP and its dual solution. In [31, 39], a parallel set of relaxations, equivalent to the SOS ones, is developed in the dual framework. The dual of checking the nonnegativity of a polynomial over a semialgebraic set turns out to be finding a sequence of moments that represent a probability measure with support in that set. To be a valid set of moments, the sequence should form a positive semidefinite moment matrix. Then, each level of relaxation fixes the size of this matrix (i.e., considers moments up a certain order) and therefore solves an SDP. This is equivalent to fixing the order of the polynomials appearing in SOS relaxations. The sufficient rank test checks a rank condition on this moment matrix and recovers (one or several) optimal $\mathbf{x}^*$, as discussed in [31].

In summary, we have the following algorithm for centralized computation of a globally optimal rate allocation to nonconcave utility maximization, where the utility functions can be written as or converted into polynomials.

Algorithm 1. Sum-of-squares for nonconcave utility maximization.

1. Formulate the relaxed problem (5.5) for a given degree D.
2. Use SDP to solve the Dth order relaxation, which can be conducted using SOSTOOLS [52].
3. If the resulting dual SDP solution satisfies the sufficient rank condition, the Dth-order optimizer $\gamma^*(D)$ is the globally optimal network utility, and a corresponding $\mathbf{x}^*$ can be obtained.[1]
4. Increase D to $D+2$, that is, the next higher-order relaxation, and repeat.

In the following section, we give examples of the application of SOS relaxation to the nonconcave NUM. We also apply the above sufficient test to check if the bound is exact, and if so, we recover the optimum rate allocation $\mathbf{x}^*$ that achieve this tightest bound.

5.2.3 Numerical Examples and Sigmoidal Utilities

Polynomial Utility Examples

First, consider quadratic utilities (i.e., $U_s(x_s) = x_s^2$) as a simple case to start with (this can be useful, for example, when the bottleneck link capacity limits sources to their convex region of a sigmoidal utility). We present examples that are typical, in our experience, of the performance of the relaxations.

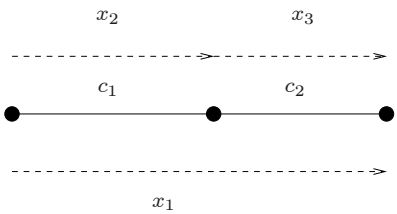

Fig. 5.3 Network topology for Example 5.1.

Example 5.1. *A small illustrative example.* Consider the simple 2-link, 3-user network shown in Figure 5.3, with $\mathbf{c} = [1,\ 2]$. The optimization problem is

$$\begin{aligned} &\text{maximize } \textstyle\sum_s x_s^2 \\ &\text{subject to } x_1 + x_2 \le 1 \\ &\qquad\qquad x_1 + x_3 \le 2 \\ &\qquad\qquad x_1, x_2, x_3 \ge 0. \end{aligned} \tag{5.6}$$

[1] Otherwise, $\gamma^*(D)$ may still be the globally optimal network utility but is only provably an upper bound.

The first level relaxation with $D = 2$ is

$$\begin{array}{l}\text{minimize } \gamma \\ \text{subject to} \\ \gamma - (x_1^2 + x_2^2 + x_3^2) - \lambda_1(-x_1 - x_2 + 1) - \lambda_2(-x_1 \\ -x_3 + 2) - \lambda_3 x_1 - \lambda_4 x_2 - \lambda_5 x_3 - \lambda_6(-x_1 - x_2 + 1) \\ (-x_1 - x_3 + 2) - \lambda_7 x_1(-x_1 - x_2 + 1) - \lambda_8 x_2(-x_1 \\ -x_2 + 1) - \lambda_9 x_3(-x_1 - x_2 + 1) - \lambda_{10} x_1(-x_1 - x_3 + 2) \\ -\lambda_{11} x_2(-x_1 - x_3 + 2) - \lambda_{12} x_3(-x_1 - x_3 + 2) - \\ \lambda_{13} x_1 x_2 - \lambda_{14} x_1 x_3 - \lambda_{15} x_2 x_3 \text{ is SOS}, \\ \lambda_i \geq 0, \ i = 1, \ldots, 15.\end{array} \tag{5.7}$$

The first constraint above can be written as $x^T Q x$ for $x = [1,\ x_1,\ x_2,\ x_3]^T$ and an appropriate Q. For example, the (1,1) entry which is the constant term reads $\gamma - \lambda_1 - 2\lambda_2 - 2\lambda_6$, the (2,1) entry, coefficient of x_1, reads $\lambda_1 + \lambda_2 - \lambda_3 + 3\lambda_6 - \lambda_7 - 2\lambda_{10}$, and so on. The expression is SOS if and only if $Q \geq 0$. The optimal γ is 5, which is achieved by, for example, $\lambda_1 = 1$, $\lambda_2 = 2$, $\lambda_3 = 1$, $\lambda_8 = 1$, $\lambda_{10} = 1$, $\lambda_{12} = 1$, $\lambda_{13} = 1$, $\lambda_{14} = 2$ and the rest of the λ_i equal to zero. Using the sufficient test (or, in this example, by inspection) we find the optimal rates $\mathbf{x}_0 = [0,\ 1,\ 2]$.

In this example, many of the λ_i could be chosen to be zero. This means not all product terms appearing in (5.7) are needed in constructing the SOS polynomial. Such information is valuable from the decentralization point of view, and can help determine to what extent our bound can be calculated in a distributed manner. This is a challenging topic for future work.

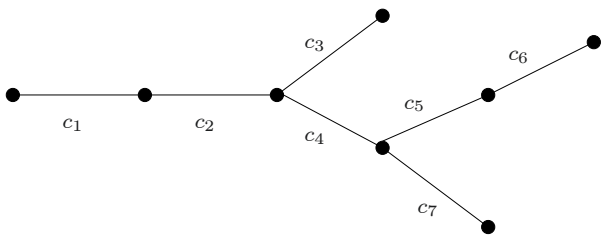

Fig. 5.4 Network topology for Example 5.2.

Example 5.2. *Larger tree topology.* As a larger example, consider the network shown in Figure 5.4 with seven links. There are nine users, with the following routing table that lists the links on each user's path.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
1,2	1,2,4	2,3	4,5	2,4	6,5,7	5,6	7	5

For $\mathbf{c} = [5,\ 10,\ 4,\ 3,\ 7,\ 3,\ 5]$, we obtain the bound $\gamma = 116$ with $D = 2$, which turns out to be globally optimal, and the globally optimal rate vector can be recovered: $\mathbf{x}_0 = [5,\ 0,\ 4,\ 0,\ 1,\ 0,\ 0,\ 5,\ 7]$. In this example, exhaustive search is too computationally intensive, and the sufficient condition test plays an important role in proving the bound is exact and in recovering $\mathbf{x}_0$.

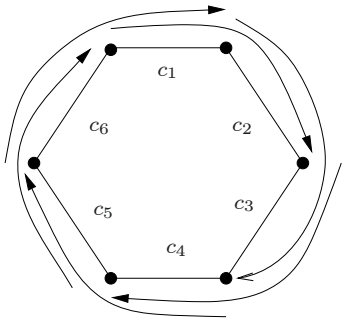

Fig. 5.5 Network topology for Example 5.3.

Example 5.3. *Large m-hop ring topology.* Consider a ring network with n nodes, n users, and n links where each user's flow starts from a node and goes clockwise through the next m links, as shown in Figure 5.5 for $n = 6$, $m = 2$. As a large example, with $n = 25$, $m = 2$, and capacities chosen randomly for a uniform distribution on $[0,\ 10]$, using relaxation of order $D = 2$ we obtain the exact bound $\gamma = 321.11$ and recover an optimal rate allocation. For $n = 30$, $m = 2$, and capacities randomly chosen from $[0,\ 15]$, it turns out that $D = 2$ relaxation yields the exact bound 816.95 and a globally optimal rate allocation.

Sigmoidal Utility Examples

Now consider sigmoidal utilities in a standard form:

$$U_s(x_s) = \frac{1}{1 + e^{-(a_s x_s + b_s)}},$$

where $\{a_s, b_s\}$ are constant integers. Even though these sigmoidal functions are not polynomials, we show the problem can be cast as one with polynomial cost and constraints, with a change of variables.

Example 5.4. *Sigmoidal utility.* Consider the simple 2-link, 3-user example shown in Figure 5.3 for $a_s = 1$ and $b_s = -5$.

The NUM problem is to

$$\begin{array}{rl} \text{maximize} & \sum_s \frac{1}{1+e^{-(x_s-5)}} \\ \text{subject to} & x_1 + x_2 \leq c_1 \\ & x_1 + x_3 \leq c_2 \\ & \mathbf{x} \geq 0. \end{array} \tag{5.8}$$

Let $y_s = 1/\left(1 + e^{-(x_s-5)}\right)$, then $x_s = -\log((1/y_s) - 1) + 5$. Substituting for $x_1,\ x_2$ in the first constraint, arranging terms and taking exponentials, then multiplying the sides by $y_1 y_2$ (note that $y_1, y_2 > 0$), we get

$$(1 - y_1)(1 - y_2) \geq e^{(10-c_1)} y_1 y_2,$$

which is polynomial in the new variables $\mathbf{y}$. This applies to all capacity constraints, and the nonnegativity constraints for x_s translate to $y_s \geq 1/\left(1 + e^5\right)$. Therefore the whole problem can be written in polynomial form, and SOS methods apply. This transformation renders the problem polynomial for general sigmoidal utility functions, with any a_s and b_s.

We present some numerical results, using a small illustrative example. Here SOS relaxations of order 4 ($D = 4$) were used. For $c_1 = 4$, $c_2 = 8$, we find $\gamma = 1.228$, which turns out to be a global optimum, with $\mathbf{x}_0 = [0,\ 4,\ 8]$ as the optimal rate vector. For $c_1 = 9$, $c_2 = 10$, we find $\gamma = 1.982$ and $\mathbf{x}_0 = [0,\ 9,\ 10]$. Now place a weight of 2 on y_1, and the other y_s have weight one; we obtain $\gamma = 1.982$ and $\mathbf{x}_0 = [9,\ 0,\ 1]$.

In general, if $a_s \neq 1$ for some s, however, the degree of the polynomials in the transformed problem may be very high. If we write the general problem as

$$\begin{array}{rl} \text{maximize} & \sum_s \frac{1}{1+e^{-(a_s x_s + b_s)}} \\ \text{subject to} & \sum_{s \in S(l)} x_s \leq c_l, \quad \forall l, \\ & \mathbf{x} \geq 0, \end{array} \tag{5.9}$$

each capacity constraint after transformation will be

$$\begin{array}{c} \prod_s (1 - y_s)^{r_{ls} \Pi_{k \neq s} a_k} \geq \\ \exp(-\prod_s a_s (c_l + \sum_s r_{ls}/a_s b_s)) \prod_s y_s^{r_{ls} \prod_{k \neq s} a_k}, \end{array}$$

where $r_{ls} = 1$ if $l \in L(s)$ and equals 0 otherwise. Because the product of the a_s appears in the exponents, $a_s > 1$ significantly increases the degree of the polynomials appearing in the problem and hence the dimension of the SDP in the SOS method.

It is therefore also useful to consider alternative representations of sigmoidal functions such as the following rational function:

$$U_s(x_s) = \frac{x_s^n}{a + x_s^n},$$

where the inflection point is $x^0 = \left((a(n-1))\,/\,(n+1)\right)^{1/n}$ and the slope at the inflection point is $U_s(x^0) = \left((n-1)\,/4n\right)\left((n+1)\,/\,(a(n-1))\right)^{1/n}$. Let

$y_s = U_s(x_s)$; the NUM problem in this case is equivalent to

$$\begin{array}{ll} \text{maximize} & \sum_s y_s \\ \text{subject to} & x_s^n - y_s x_s^n - a y_s = 0 \\ & \sum_{s \in S(l)} x_s \le c_l, \ \forall l \\ & \mathbf{x} \ge 0 \end{array} \tag{5.10}$$

which again can be accommodated in the SOS method and be solved by Algorithm 1.

The benefit of this choice of utility function is that the largest degree of the polynomials in the problem is $n + 1$, therefore growing linearly with n. The disadvantage compared to the exponential form for sigmoidal functions is that the location of the inflection point and the slope at that point cannot be set independently.

5.2.4 Alternative Representations for Convex Relaxations to Nonconcave NUM

The SOS relaxation we used in the last two sections is based on Schmüdgen's representation for positive polynomials over compact sets described by other polynomials. We now briefly discuss two other representations of relevance to the NUM, that are interesting from both theoretical (e.g., interpretation) and computational points of view.

LP Relaxation

Exploiting linearity of the constraints in NUM and with the additional assumption of nonempty interior for the feasible set (which holds for NUM), we can use Handelman's representation [30] and refine the positivstellensatz condition to obtain the following convex relaxation of nonconcave NUM problem.

$$\begin{array}{l} \text{maximize } \gamma \\ \text{subject to} \\ \gamma - \sum_s U_s(x_s) = \sum_{\alpha \in N^L} \lambda_\alpha \prod_{l=1}^{L} (c_l - \sum_{s \in S(l)} x_s)^{\alpha_l}, \ \forall \mathbf{x} \\ \lambda_\alpha \ge 0, \ \forall \alpha, \end{array} \tag{5.11}$$

where the optimization variables are γ and λ_α, and α denotes an ordered set of integers $\{\alpha_l\}$.

Fixing D where $\sum_l \alpha_l \le D$, and equating the coefficients on the two sides of the equality in (5.11), yields a linear program (LP). There are no

SOS terms, therefore no semidefiniteness conditions. As before, increasing the degree D gives higher-order relaxations and a tighter bound.

We provide a pricing interpretation for problem (5.11). First, normalize each capacity constraint as $1 - u_l(x) \geq 0$, where $u_l(x) = \sum_{s \in S(l)} x_s / c_l$. We can interpret $u_l(x)$ as *link usage*, or the probability that link l is used at any given point in time. Then, in (5.11), we have terms linear in u such as $\lambda_l(1 - u_l(x))$, in which λ_l has a similar interpretation as in concave NUM, as the price of using link l. We also have product terms such as $\lambda_{jk}(1-u_j(x))(1-u_k(x))$, where $\lambda_{jk} u_j(x) u_k(x)$ indicates the probability of simultaneous usage of links j and k, for links whose usage probabilities are independent (e.g., they do not share any flows). Products of more terms can be interpreted similarly.

Although the above price interpretation is not complete and does not justify all the terms appearing in (5.11) (e.g., powers of the constraints, product terms for links with shared flows), it does provide some useful intuition: this relaxation results in a pricing scheme that provides better incentives for the users to observe the constraints, by giving an additional reward (because the corresponding term adds positively to the utility) for simultaneously keeping two links free. Such incentive helps tighten the upper bound and eventually achieve a feasible (and optimal) allocation.

This relaxation is computationally attractive because we need to solve an LPs instead of the previous SDPs at each level. However, significantly more levels may be required [40].

Relaxation with No Product Terms

Putinar [53] showed that a polynomial positive over a compact set[2] can be represented as an SOS-combination of the constraints. This yields the following convex relaxation for nonconcave NUM problem.

$$\begin{array}{l} \text{maximize} \quad \gamma \\ \text{subject to} \\ \gamma - \sum_s U_s(x_s) = \sum_{l=1}^{L} \lambda_l(\mathbf{x})(c_l - \sum_{s \in S(l)} x_s), \ \forall \mathbf{x} \\ \lambda(\mathbf{x}) \text{ is SOS}, \end{array} \tag{5.12}$$

where the optimization variables are the coefficients in $\lambda_l(\mathbf{x})$. Similar to the SOS relaxation (5.5), fixing the order D of the expression in (5.12) results in an SDP. This relaxation has the nice property that no product terms appear: the relaxation becomes exact with a high enough D without the need of product terms. However, this degree might be much higher than what the previous SOS method requires.

[2] With an extra assumption that always holds for linear constraints as in NUM problems.

5.2.5 Concluding Remarks and Future Directions

We consider the NUM problem in the presence of inelastic flows, that is, flows with nonconcave utilities. Despite its practical importance, this problem has not been studied widely, mainly due to the fact it is a nonconvex problem. There has been no effective mechanism, centralized or distributed, to compute the globally optimal rate allocation for nonconcave utility maximization problems in networks. This limitation has made performance assessment and design of networks that include inelastic flows very difficult.

In one of the recent works on this topic [19], we employed convex SOS relaxations, solved by a sequence of SDPs, to obtain high-quality, increasingly tighter upper bounds on total achievable utility. In practice, the performance of our SOSTOOLS-based algorithm was surprisingly good, and bounds obtained using a polynomial-time (and indeed a low-order and often minimal-order) relaxation were found to be exact, achieving the global optimum of nonconcave NUM problems. Furthermore, a dual-based sufficient test, if successful, detects the exactness of the bound, in which case the optimal rate allocation can also be recovered. This performance of the proposed algorithm brings up a fundamental question on whether there is any particular property or structure in nonconcave NUM that makes it especially suitable for SOS relaxations.

We further examined the use of two more specialized polynomial representations, one that uses products of constraints with constant multipliers, resulting in LP relaxations; and at the other end of spectrum, one that uses a linear combination of constraints with SOS multipliers. We expect these relaxations to give higher-order certificates, thus their potential computational benefits need to be examined further. We also show they admit economics interpretations (e.g., prices, incentives) that provide some insight on how the SOS relaxations work in the framework of link congestion pricing for the simultaneous usage of multiple links.

An important research issue to be further investigated is decentralization methods for rate allocation among sources with nonconcave utilities. The proposed algorithm here is not easy to decentralize, given the products of the constraints or polynomial multipliers that destroy the separable structure of the problem. However, when relaxations become exact, the sparsity pattern of the coefficients can provide information about partially decentralized computation of optimal rates. For example, if after solving the NUM offline, we obtain an exact bound, then if the coefficient of the cross-term $x_i x_j$ turns out to be zero, it means users i and j do not need to communicate to each other to find their optimal rates. An interesting next step in this area of research is to investigate a distributed version of the proposed algorithm through limited message passing among clusters of network nodes and links.

5.3 Wireless Network Power Control

5.3.1 Introduction

Due to the broadcast nature of radio transmission, data rates and other quality of service (QoS) issues in a wireless network are affected by interference. This is particularly important in CDMA systems where users transmit at the same time over the same frequency bands and their spreading codes are not perfectly orthogonal. Transmit power control is often used to tackle this problem of signal interference [12]. We study how to optimize over the transmit powers to create the optimal set of signal-to-interference ratios (SIR) on wireless links. Optimality here can be with respect to a variety of objectives, such as maximizing a systemwide efficiency metric (e.g., the total system throughput), or maximizing a QoS metric for a user in the highest QoS class, or maximizing a QoS metric for the user with the minimum QoS metric value (i.e., a maxmin optimization).

The objective represents a systemwide goal to be optimized; however, individual users' QoS requirements also need to be satisfied. Any power allocation must therefore be constrained by a feasible set formed by these minimum requirements from the users. Such a constrained optimization captures the tradeoff between user-centric constraints and some network-centric objective. Because a higher power level from one transmitter increases the interference levels at other receivers, there may not be any feasible power allocation to satisfy the requirements from all the users. Sometimes an existing set of requirements can be satisfied, but when a new user is admitted into the system, there exist no more feasible power control solutions, or the maximized objective is reduced due to the tightening of the constraint set, leading to the need for admission control and admission pricing, respectively.

Because many QoS metrics are nonlinear functions of SIR, which is in turn a nonlinear (and neither convex nor concave) function of transmit powers, in general, power control optimization or feasibility problems are difficult nonlinear optimization problems that may appear to be NP-hard problems. Following [14, 35], this section shows that, when SIR is much larger than 0 dB, a class of nonlinear optimization called geometric programming (GP) can be used to efficiently compute the globally optimal power control in many of these problems, and efficiently determine the feasibility of user requirements by returning either a feasible (and indeed optimal) set of powers or a certificate of infeasibility. This also leads to an effective admission control and admission pricing method.

The key observation is that despite the apparent nonconvexity, through log change of variable the GP technique turns these constrained optimizations of power control into convex optimization, which is intrinsically tractable despite its nonlinearity in objective and constraints. However, when SIR is comparable to or below 0 dB, the power control problems are truly nonconvex

with no efficient and global solution methods. In this case, we present a heuristic that is provably convergent and empirically almost always computes the globally optimal power allocation by solving a sequence of GPs through the approach of successive convex approximations.

The GP approach reveals the hidden convexity structure, which implies efficient solution methods and the global optimality of any local optimum in power control problems with nonlinear objective functions. It clearly differentiates the tractable formulations in a high-SIR regime from the intractable ones in a low-SIR regime. Power control by GP is applicable to formulations in both cellular networks with single-hop transmission between mobile users and base stations, and ad hoc networks with multihop transmission among the nodes, as illustrated through several numerical examples in this section. Traditionally, GP is solved by centralized computation through the highly efficient interior point methods. In this section we present a new result on how GP can be solved distributively with message passing, which has independent value to general maximization of coupled objective, and applies it to power control problems with a further reduction of message-passing overhead by leveraging the specific structures of power control problems.

More generally, the technique of nonlinear change of variables, including the log change of variables, to reveal "hidden" convexity in optimization formulations has recently become quite popular in the communication network research community.

5.3.2 Geometric Programming

GP is a class of nonlinear, nonconvex optimization problems with many useful theoretical and computational properties. It was invented in 1967 by Duffin, Peterson, and Zener [17], and much of the development by the early 1980s was summarized in [1]. Because a GP can be turned into a convex optimization problem, a local optimum is also a global optimum, the Lagrange duality gap is zero under mild conditions, and a global optimum can be computed very efficiently. Numerical efficiency holds both in theory and in practice: interior point methods applied to GP have provably polynomial-time complexity [48], and are very fast in practice with high-quality software downloadable from the Internet (e.g., the MOSEK package). Convexity and duality properties of GP are well understood, and large-scale, robust numerical solvers for GP are available. Furthermore, special structures in GP and its Lagrange dual problem lead to distributed algorithms, physical interpretations, and computational acceleration beyond the generic results for convex optimization. A detailed tutorial of GP and comprehensive survey of its recent applications to communication systems and to circuit design can be found in [11] and [3], respectively. This section contains a brief introduction of GP terminology.

There are two equivalent forms of GP: standard form and convex form. The first is a constrained optimization of a type of function called posynomial, and the second form is obtained from the first through a logarithmic change of variable.

We first define a monomial as a function $f : \mathbf{R}^n_{++} \rightarrow \mathbf{R}$:

$$f(\mathbf{x}) = d x_1^{a^{(1)}} x_2^{a^{(2)}} \ldots x_n^{a^{(n)}},$$

where the multiplicative constant $d \geq 0$ and the exponential constants $a^{(j)} \in \mathbf{R}$, $j = 1, 2, \ldots, n$. A sum of monomials, indexed by k below, is called a posynomial:

$$f(\mathbf{x}) = \sum_{k=1}^{K} d_k x_1^{a_k^{(1)}} x_2^{a_k^{(2)}} \ldots x_n^{a_k^{(n)}},$$

where $d_k \geq 0$, $k = 1, 2, \ldots, K$, and $a_k^{(j)} \in \mathbf{R}$, $j = 1, 2, \ldots, n$, $k = 1, 2, \ldots, K$. For example, $2x_1^{-\pi} x_2^{0.5} + 3x_1 x_3^{100}$ is a posynomial in $\mathbf{x}$, $x_1 - x_2$ is not a posynomial, and x_1/x_2 is a monomial, thus also a posynomial.

Minimizing a posynomial subject to posynomial upper bound inequality constraints and monomial equality constraints is called GP in *standard form*:

$$\begin{array}{ll} \text{minimize} & f_0(\mathbf{x}) \\ \text{subject to} & f_i(\mathbf{x}) \leq 1, \quad i = 1, 2, \ldots, m, \\ & h_l(\mathbf{x}) = 1, \quad l = 1, 2, \ldots, M, \end{array} \tag{5.13}$$

where f_i, $i = 0, 1, \ldots, m$, are posynomials: $f_i(\mathbf{x}) = \sum_{k=1}^{K_i} d_{ik} x_1^{a_{ik}^{(1)}} x_2^{a_{ik}^{(2)}} \ldots x_n^{a_{ik}^{(n)}}$, and h_l, $l = 1, 2, \ldots, M$, are monomials: $h_l(\mathbf{x}) = d_l x_1^{a_l^{(1)}} x_2^{a_l^{(2)}} \ldots x_n^{a_l^{(n)}}$.

GP in standard form is not a convex optimization problem, because posynomials are not convex functions. However, with a logarithmic change of the variables and multiplicative constants: $y_i = \log x_i$, $b_{ik} = \log d_{ik}$, $b_l = \log d_l$, and a logarithmic change of the functions' values, we can turn it into the following equivalent problem in $\mathbf{y}$.

$$\begin{array}{ll} \text{minimize} & p_0(\mathbf{y}) = \log \sum_{k=1}^{K_0} \exp(\mathbf{a}_{0k}^T \mathbf{y} + b_{0k}) \\ \text{subject to} & p_i(\mathbf{y}) = \log \sum_{k=1}^{K_i} \exp(\mathbf{a}_{ik}^T \mathbf{y} + b_{ik}) \leq 0, \quad i = 1, 2, \ldots, m, \\ & q_l(\mathbf{y}) = \mathbf{a}_l^T \mathbf{y} + b_l = 0, \quad l = 1, 2, \ldots, M. \end{array} \tag{5.14}$$

This is referred to as GP in *convex form*, which is a convex optimization problem because it can be verified that the log-sum-exp function is convex [4].

In summary, GP is a nonlinear, nonconvex optimization problem that can be transformed into a nonlinear convex problem. GP in standard form can be used to formulate network resource allocation problems with nonlinear objectives under nonlinear QoS constraints. The basic idea is that resources are often allocated proportional to some parameters, and when resource allo-

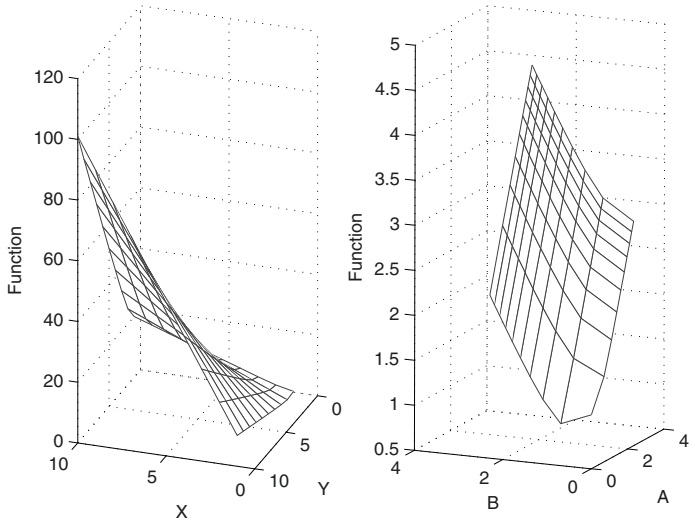

Fig. 5.6 A bivariate posynomial before (left graph) and after (right graph) the log transformation. A nonconvex function is turned into a convex one.

cations are optimized over these parameters, we are maximizing an inverted posynomial subject to lower bounds on other inverted posynomials, which are equivalent to GP in standard form.

SP/GP, SOS/SDP

Note that, although the posynomial seems to be a nonconvex function, it becomes a convex function after the log transformation, as shown in an example in Figure 5.6. Compared to the (constrained or unconstrained) minimization of a polynomial, the minimization of a posynomial in GP relaxes the integer constraint on the exponential constants but imposes a positivity constraint on the multiplicative constants and variables. There is a sharp contrast between these two problems: polynomial minimization is NP-hard, but GP can be turned into convex optimization with provably polynomial-time algorithms for a global optimum.

In an extension of GP called signomial programming discussed later in this section, the restriction of nonnegative multiplicative constants is removed. This results in a general class of nonlinear and truly nonconvex problems that is simultaneously a generalization of GP and polynomial minimization over the positive quadrant, as summarized in the comparison Table 5.2.

Table 5.2 Comparison of GP, constrained polynomial minimization over the positive quadrant (PMoP), and signomial programming (SP). All three types of problems minimize a sum of monomials subject to upper bound inequality constraints on sums of monomials, but have different definitions of monomial: $c\prod_j x_j^{a^{(j)}}$, as shown in the table. GP is known to be polynomial-time solvable, but PMoP and SP are not.

	GP	*PMoP*	*SP*
c	$\mathbf{R}_+$	$\mathbf{R}$	$\mathbf{R}$
$a^{(j)}$	$\mathbf{R}$	$\mathcal{Z}_+$	$\mathbf{R}$
x_j	$\mathbf{R}_{++}$	$\mathbf{R}_{++}$	$\mathbf{R}_{++}$

The objective function of signomial programming can be formulated as minimizing a ratio between two posynomials, which is not a posynomial (because posynomials are closed under positive multiplication and addition but not division). As shown in Figure 5.7, a ratio between two posynomials is a nonconvex function both before and after the log transformation. Although it does not seem likely that signomial programming can be turned into a convex optimization problem, there are heuristics to solve it through a sequence of GP relaxations. However, due to the absence of algebraic structures found in polynomials, such methods for signomial programming currently lack a theoretical foundation of convergence to global optimality. This is in contrast to the sum-of-squares method [51], which uses a nested family of SDP relax-

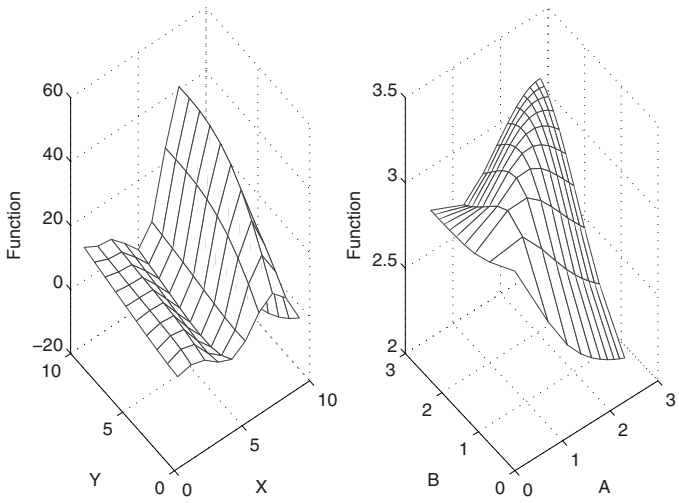

Fig. 5.7 Ratio between two bivariate posynomials before (left graph) and after (right graph) the log transformation. It is a nonconvex function in both cases.

ations to solve constrained polynomial minimization problems as explained in the last section.

5.3.3 Power Control by Geometric Programming: Convex Case

Various schemes for power control, centralized or distributed, have been extensively studied since the 1990s based on different transmission models and application needs (e.g., in [2, 26, 47, 55, 63, 72]). This section summarizes the new approach of formulating power control problems through GP. The key advantage is that globally optimal power allocations can be efficiently computed for a variety of nonlinear systemwide objectives and user QoS constraints, even when these nonlinear problems appear to be nonconvex optimization.

Basic Model

Consider a wireless (cellular or multihop) network with n logical transmitter/receiver pairs. Transmit powers are denoted as $P_1, \ldots, P_n$. In the cellular uplink case, all logical receivers may reside in the same physical receiver, that is, the base station. In the multihop case, because the transmission environment can be different on the links comprising an end-to-end path, power control schemes must consider each link along a flow's path.

Under Rayleigh fading, the power received from transmitter j at receiver i is given by $G_{ij}F_{ij}P_j$ where $G_{ij} \geq 0$ represents the path gain (it may also encompass antenna gain and coding gain) that is often modeled as proportional to $d_{ij}^{-\gamma}$, where d_{ij} denotes distance, γ is the power fall-off factor, and F_{ij} model Rayleigh fading and are independent and exponentially distributed with unit mean. The distribution of the received power from transmitter j at receiver i is then exponential with mean value $\mathbf{E}[G_{ij}F_{ij}P_j] = G_{ij}P_j$. The SIR for the receiver on logical link i is:

$$\mathsf{SIR}_i = \frac{P_i G_{ii} F_{ii}}{\sum_{j \neq i}^{N} P_j G_{ij} F_{ij} + n_i} \tag{5.15}$$

where n_i is the noise power for receiver i.

The constellation size M used by a link can be closely approximated for MQAM modulations as follows. $M = 1 + (-\phi_1 / (\ln(\phi_2 \mathsf{BER}))) \mathsf{SIR}$, where BER is the bit error rate and ϕ_1, ϕ_2 are constants that depend on the modulation type. Defining $K = -\phi_1 / (\ln(\phi_2 \mathsf{BER}))$ leads to an expression of the data rate R_i on the ith link as a function of the SIR: $R_i = (1/T)\log_2(1 + K\mathsf{SIR}_i)$, which can be approximated as

$$R_i = \frac{1}{T}\log_2(K\mathsf{SIR}_i) \tag{5.16}$$

when $K\mathsf{SIR}$ is much larger than 1. This approximation is reasonable either when the signal level is much higher than the interference level or, in CDMA systems, when the spreading gain is large. For notational simplicity in the rest of this section, we redefine G_{ii} as K times the original G_{ii}, thus absorbing constant K into the definition of SIR.

The aggregate data rate for the system can then be written as

$$R_{\text{system}} = \sum_i R_i = \frac{1}{T}\log_2\left[\prod_i \mathsf{SIR}_i\right].$$

So in the high SIR regime, aggregate data rate maximization is equivalent to maximizing a product of SIR. The system throughput is the aggregate data rate supportable by the system given a set of users with specified QoS requirements.

Outage probability is another important QoS parameter for reliable communication in wireless networks. A channel outage is declared and packets lost when the received SIR falls below a given threshold SIR_{th}, often computed from the BER requirement. Most systems are interference-dominated and the thermal noise is relatively small, thus the ith link outage probability is

$$\begin{aligned} P_{\text{o},i} &= \mathbf{Prob}\{\mathsf{SIR}_i \le \mathsf{SIR}_{\text{th}}\} \\ &= \mathbf{Prob}\{G_{ii}F_{ii}P_i \le \mathsf{SIR}_{\text{th}}\sum_{j\neq i} G_{ij}F_{ij}P_j\}. \end{aligned}$$

The outage probability can be expressed as [38]

$$P_{\text{o},i} = 1 - \prod_{j\neq i}\frac{1}{1+\frac{\mathsf{SIR}_{\text{th}}G_{ij}P_j}{G_{ii}P_i}},$$

which means that the upper bound $P_{\text{o},i} \le P_{\text{o},i,\max}$ can be written as an upper bound on a posynomial in $\mathbf{P}$:

$$\prod_{j\neq i}\left(1+\frac{\mathsf{SIR}_{\text{th}}G_{ij}P_j}{G_{ii}P_i}\right) \le \frac{1}{1-P_{\text{o},i,\max}}. \tag{5.17}$$

Cellular Wireless Networks

We first present how GP-based power control applies to cellular wireless networks with one-hop transmission from N users to a base station. These results extend the scope of power control by the classical solution in CDMA

systems that equalizes SIRs, and those by the iterative algorithms (e.g., in [2, 26, 47]) that minimize total power (a linear objective function) subject to SIR constraints.

We start the discussion on the suite of power control problem formulations with a simple objective function and basic constraints. The following constrained problem of maximizing the SIR of a particular user i^* is a GP.

$$\begin{array}{ll} \text{maximize} & R_{i^*}(\mathbf{P}) \\ \text{subject to} & R_i(\mathbf{P}) \geq R_{i,\min}, \quad \forall i, \\ & P_{i1}G_{i1} = P_{i2}G_{i2}, \\ & 0 \leq P_i \leq P_{i,\max}, \quad \forall i. \end{array}$$

The first constraint, equivalent to $\mathsf{SIR}_i \geq \mathsf{SIR}_{i,\min}$, sets a floor on the SIR of other users and protects these users from user i^* increasing her transmit power excessively. The second constraint reflects the classical power control criterion in solving the near–far problem in CDMA systems: the expected received power from one transmitter $i1$ must equal that from another $i2$. The third constraint is regulatory or system limitations on transmit powers. All constraints can be verified to be inequality upper bounds on posynomials in transmit power vector $\mathbf{P}$.

Alternatively, we can use GP to maximize the minimum rate among all users. The maxmin fairness objective

$$\text{maximize}_{\mathbf{P}} \quad \min_i \{R_i\}$$

can be accommodated in GP-based power control because it can be turned into equivalently maximizing an auxiliary variable t such that $\mathsf{SIR}_i(\mathbf{P}) \geq \exp(t)$, $\forall i$, which has a posynomial objective and constraints in $(\mathbf{P}, t)$.

Example 5.5. *A small illustrative example.* A simple system comprised of five users is used for a numerical example. The five users are spaced at distances d of $1, 5, 10, 15$, and 20 units from the base station. The power fall-off factor $\gamma = 4$. Each user has a maximum power constraint of $P_{\max} = 0.5$ mW. The noise power is 0.5 μW for all users. The SIR of all users, other than the user we are optimizing for, must be greater than a common threshold SIR level β. In different experiments, β is varied to observe the effect on the optimized user's SIR. This is done independently for the near user at $d = 1$, a medium distance user at $d = 15$, and the far user at $d = 20$. The results are plotted in Figure 5.8.

Several interesting effects are illustrated. First, when the required threshold SIR in the constraints is sufficiently high, there is no feasible power control solution. At moderate threshold SIR, as β is decreased, the optimized SIR initially increases rapidly. This is because it is allowed to increase its own power by the sum of the power reductions in the four other users, and the noise is relatively insignificant. At low threshold SIR, the noise becomes more significant and the power tradeoff from the other users less significant,

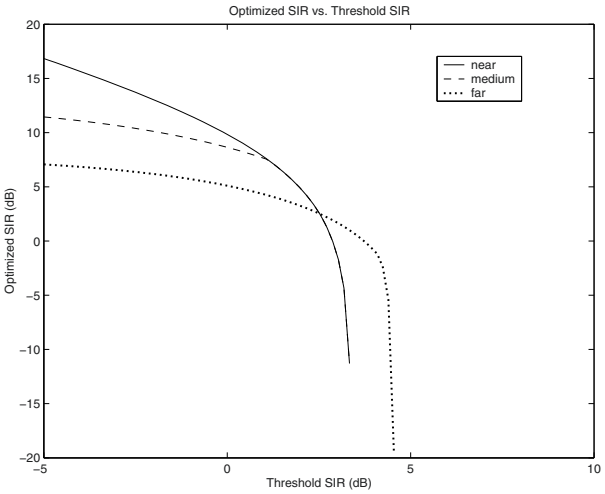

Fig. 5.8 Constrained optimization of power control in a cellular network (Example 5.5).

so the curve starts to bend over. Eventually, the optimized user reaches its upper bound on power and cannot utilize the excess power allowed by the lower threshold SIR for other users. This is exhibited by the transition from a sharp bend in the curve to a much shallower sloped curve.

We now proceed to show that GP can also be applied to the problem formulations with an overall system objective of total system throughput, under both user data rate constraints and outage probability constraints.

The following constrained problem of maximizing system throughput is a GP.

$$\begin{array}{ll} \text{maximize} & R_{\text{system}}(\mathbf{P}) \\ \text{subject to} & R_i(\mathbf{P}) \geq R_{i,\min}, \quad \forall i, \\ & P_{\text{o},i}(\mathbf{P}) \leq P_{\text{o},i,\max}, \quad \forall i, \\ & 0 \leq P_i \leq P_{i,\max}, \quad \forall i \end{array} \tag{5.18}$$

where the optimization variables are the transmit powers $\mathbf{P}$. The objective is equivalent to minimizing the posynomial $\prod_i \text{ISR}_i$, where ISR is $1/\mathsf{SIR}$. Each ISR is a posynomial in $\mathbf{P}$ and the product of posynomials is again a posynomial. The first constraint is from the data rate demand $R_{i,\min}$ by each user. The second constraint represents the outage probability upper bounds $P_{\text{o},i,\max}$. These inequality constraints put upper bounds on posynomials of $\mathbf{P}$, as can be readily verified through (5.16) and (5.17). Thus (5.18) is indeed a GP, and efficiently solvable for global optimality.

There are several obvious variations of problem (5.18) that can be solved by GP; for example, we can lower bound R_{system} as a constraint and maximize R_{i^*} for a particular user i^*, or have a total power $\sum_i P_i$ constraint or objective function.

Table 5.3 Suite of power control optimization solvable by GP

Objective Function	*Constraints*
(A) Max R_{i*} (specific user)	(a) $R_i \geq R_{i,\min}$ (rate constraint)
(B) Max $\min_i R_i$ (worst-case user)	(b) $P_{i1}G_{i1} = P_{i2}G_{i2}$ (near–far constraint)
(C) Max $\sum_i R_i$ (total throughput)	(c) $\sum_i R_i \geq R_{\text{system,min}}$ (sum rate constraint)
(D) Max $\sum_i w_i R_i$ (weighted rate sum)	(d) $P_{\text{o},i} \leq P_{\text{o},i,\max}$ (outage probability constraint)
(E) Min $\sum_i P_i$ (total power)	(e) $0 \leq P_i \leq P_{i,\max}$ (power constraint)

The objective function to be maximized can also be generalized to a weighted sum of data rates, $\sum_i w_i R_i$, where $\mathbf{w} \succeq 0$ is a given weight vector. This is still a GP because maximizing $\sum_i w_i \log \mathsf{SIR}_i$ is equivalent to maximizing $\log \prod_i \mathsf{SIR}_i^{w_i}$, which is in turn equivalent to minimizing $\prod_i \text{ISR}_i^{w_i}$. Now use auxiliary variables $\{t_i\}$, and minimize $\prod_i t_i^{w_i}$ over the original constraints in (5.18) plus the additional constraints $\text{ISR}_i \leq t_i$ for all i. This is readily verified to be a GP in $(\mathbf{x}, \mathbf{t})$, and is equivalent to the original problem.

Generalizing the above discussions and observing that high-SIR assumption is needed for GP formulation only when there are sums of $\log(1 + \mathsf{SIR})$ in the optimization problem, we have the following summary.

Proposition 5.1. *In the high-SIR regime, any combination of objectives* (A)–(E) *and constraints* (a)–(e) *in Table* 5.3 (*pick any one of the objectives and any subset of the constraints*) *is a power control optimization problem that can be solved by GP, that is, can be transformed into a convex optimization with efficient algorithms to compute the globally optimal power vector. When objectives* (C)–(D) *or constraints* (c)–(d) *do not appear, the power control optimization problem can be solved by GP in any SIR regime.*

In addition to efficient computation of the globally optimal power allocation with nonlinear objectives and constraints, GP can also be used for admission control based on feasibility study described in [11], and for determining which QoS constraint is a performance bottleneck, that is, met tightly at the optimal power allocation.[3]

[3] This is because most GP solution algorithms solve both the primal GP and its Lagrange dual problem, and by the complementary slackness condition, a resource constraint is tight at optimal power allocation when the corresponding optimal dual variable is nonzero.

Extensions

In wireless multihop networks, system throughput may be measured either by end-to-end transport layer utilities or by link layer aggregate throughput. GP application to the first approach has appeared in [10], and those to the second approach in [11]. Furthermore, delay and buffer overflow properties can also be accommodated in the constraints or objective function of GP-based power control.

5.3.4 Power Control by Geometric Programming: Nonconvex Case

If we maximize the total throughput R_{system} in the medium to low SIR case (i.e., when SIR is not much larger than 0 dB), the approximation of $\log(1+\mathsf{SIR})$ as $\log \mathsf{SIR}$ does not hold. Unlike SIR, which is an inverted posynomial, $1+\mathsf{SIR}$ is not an inverted posynomial. Instead, $1/(1+\mathsf{SIR})$ is a ratio between two posynomials:

$$\frac{f(\mathbf{P})}{g(\mathbf{P})} = \frac{\sum_{j\neq i} G_{ij}P_j + n_i}{\sum_j G_{ij}P_j + n_i}. \tag{5.19}$$

Minimizing, or upper bounding, a ratio between two posynomials belongs to a truly nonconvex class of problems known as complementary GP [1, 11] that is in general an NP-hard problem. An equivalent generalization of GP is signomial programming [1, 11]: minimizing a signomial subject to upper bound inequality constraints on signomials, where a signomial $s(\mathbf{x})$ is a sum of monomials, possibly with negative multiplicative coefficients: $s(\mathbf{x}) = \sum_{i=1}^{N} c_i g_i(\mathbf{x})$ where $\mathbf{c} \in \mathbf{R}^N$ and $g_i(\mathbf{x})$ are monomials.[4]

Successive Convex Approximation Method

Consider the following nonconvex problem,

$$\begin{array}{l} \text{minimize} \quad f_0(\mathbf{x}) \\ \text{subject to } f_i(\mathbf{x}) \leq 1, \quad i = 1, 2, \ldots, m, \end{array} \tag{5.20}$$

where f_0 is convex without loss of generality,[5] but the $f_i(\mathbf{x})$s, $\forall i$ are nonconvex. Because directly solving this problem is NP-hard, we want to solve it by

[4] An SP can always be converted into a complementary GP, because an inequality in SP, which can be written as $f_{i1}(\mathbf{x}) - f_{i2}(\mathbf{x}) \leq 1$, where f_{i1}, f_{i2} are posynomials, is equivalent to an inequality $f_{i1}(\mathbf{x})/(1+f_{i2}(\mathbf{x})) \leq 1$ in complementary GP.

[5] If f_0 is nonconvex, we can move the objective function to the constraint by introducing auxiliary scalar variable t and writing minimize t subject to the additional constraint $f_0(\mathbf{x}) - t \leq 0$.

a series of approximations $\tilde{f}_i(\mathbf{x}) \approx f_i(\mathbf{x}), \forall \mathbf{x}$, each of which can be optimally solved in an easy way. It is known [46] that if the approximations satisfy the following three properties, then the solutions of this series of approximations converge to a point satisfying the necessary optimality Karush–Kuhn–Tucker (KKT) conditions of the original problem.

(1) $f_i(\mathbf{x}) \leq \tilde{f}_i(\mathbf{x})$ for all $\mathbf{x}$.
(2) $f_i(\mathbf{x}_0) = \tilde{f}_i(\mathbf{x}_0)$ where x_0 is the optimal solution of the approximated problem in the previous iteration.
(3) $\nabla f_i(\mathbf{x}_0) = \nabla \tilde{f}_i(\mathbf{x}_0)$.

The following algorithm describes the generic successive approximation approach. Given a method to approximate $f_i(\mathbf{x})$ with $\tilde{f}_i(\mathbf{x})$, $\forall i$, around some point of interest $\mathbf{x}_0$, the following algorithm provides the output of a vector that satisfies the KKT conditions of the original problem.

Algorithm 2. Successive approximation to a nonconvex problem.

1. Choose an initial feasible point $\mathbf{x}^{(0)}$ and set $k = 1$.
2. Form an approximated problem of (5.20) based on the previous point $\mathbf{x}^{(k-1)}$.
3. Solve the kth approximated problem to obtain $\mathbf{x}^{(k)}$.
4. Increment k and go to step 2 until convergence to a stationary point.

Single condensation method. Complementary GPs involve upper bounds on the ratio of posynomials as in (5.19); they can be turned into GPs by approximating the denominator of the ratio of posynomials, $g(\mathbf{x})$, with a monomial $\tilde{g}(\mathbf{x})$, but leaving the numerator $f(\mathbf{x})$ as a posynomial.

The following basic result can be readily proved using the arithmetic-mean–geometric-mean inequality.

Lemma 5.1. *Let* $g(\mathbf{x}) = \sum_i u_i(\mathbf{x})$ *be a posynomial. Then*

$$g(\mathbf{x}) \geq \tilde{g}(\mathbf{x}) = \prod_i \left(\frac{u_i(\mathbf{x})}{\alpha_i} \right)^{\alpha_i}. \tag{5.21}$$

If, in addition, $\alpha_i = u_i(\mathbf{x}_0)/g(\mathbf{x}_0), \ \forall i$, *for any fixed positive* $\mathbf{x}_0$, *then* $\tilde{g}(\mathbf{x}_0) = g(\mathbf{x}_0)$, *and* $\tilde{g}(\mathbf{x})$ *is the best local monomial approximation to* $g(\mathbf{x})$ *near* $\mathbf{x}_0$ *in the sense of first-order Taylor approximation.*

The above lemma easily leads to the following

Proposition 5.2. *The approximation of a ratio of posynomials* $f(\mathbf{x})/g(\mathbf{x})$ *with* $f(\mathbf{x})/\tilde{g}(\mathbf{x})$ *where* $\tilde{g}(\mathbf{x})$ *is the monomial approximation of* $g(\mathbf{x})$ *using the arithmetic-geometric mean approximation of Lemma* 5.1 *satisfies the three conditions for the convergence of the successive approximation method.*

Double condensation method. Another choice of approximation is to make a double monomial approximation for both the denominator and numerator in (5.19). However, in order to satisfy the three conditions for the convergence

of the successive approximation method, a monomial approximation for the numerator $f(\mathbf{x})$ should satisfy $f(\mathbf{x}) \leq \tilde{f}(\mathbf{x})$.

Applications to Power Control

Figure 5.9 shows a block diagram of the approach of GP-based power control for a general SIR regime [64]. In the high SIR regime, we need to solve only one GP. In the medium to low SIR regimes, we solve truly nonconvex power control problems that cannot be turned into convex formulation through a series of GPs.

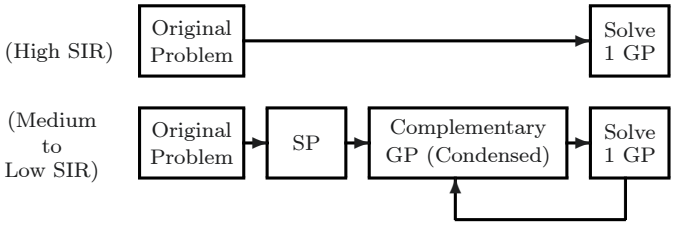

Fig. 5.9 GP-based power control in different SIR regimes.

GP-based power control problems in the medium to low SIR regimes become SP (or, equivalently, complementary GP), which can be solved by the single or double condensation method. We focus on the single condensation method here. Consider a representative problem formulation of maximizing total system throughput in a cellular wireless network subject to user rate and outage probability constraints in problem (5.18), which can be explicitly written out as

$$\begin{array}{ll} \text{minimize} & \prod_{i=1}^{N} \frac{1}{1+\mathsf{SIR}_i} \\ \text{subject to} & (2^{TR_{i,\min}}-1)\frac{1}{\mathsf{SIR}_i} \leq 1, \quad i=1,\ldots,N, \\ & (\mathsf{SIR}_{\text{th}})^{N-1}(1-P_{\text{o},i,\max})\prod_{j\neq i}^{N}\frac{G_{ij}P_j}{G_{ii}P_i} \leq 1, \quad i=1,\ldots,N, \\ & P_i(P_{i,\max})^{-1} \leq 1, \quad i=1,\ldots,N. \end{array} \tag{5.22}$$

All the constraints are posynomials. However, the objective is not a posynomial, but a ratio between two posynomials as in (5.19). This power control problem can be solved by the condensation method by solving a series of GPs. Specifically, we have the following single-condensation algorithm.

Algorithm 3. Single condensation GP power control.

1. Evaluate the denominator posynomial of the objective function in (5.22) with the given $\mathbf{P}$.

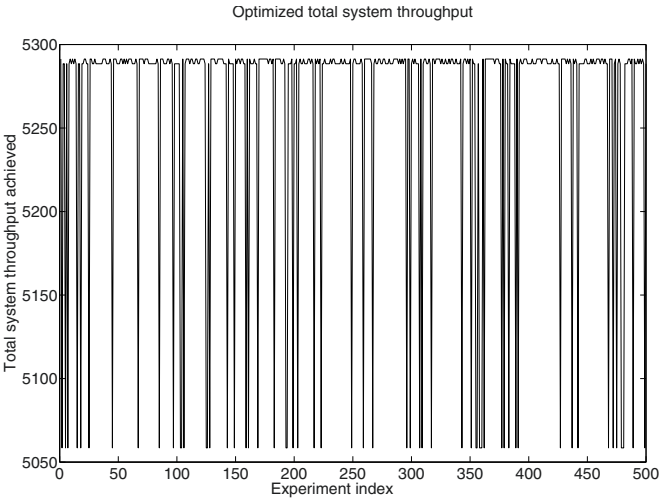

Fig. 5.10 Maximized total system throughput achieved by the (single) condensation method for 500 different initial feasible vectors (Example 5.6). Each point represents a different experiment with a different initial power vector.

2. Compute for each term i in this posynomial,

$$\alpha_i = \frac{\text{value of } i\text{th term in posynomial}}{\text{value of posynomial}}.$$

3. Condense the denominator posynomial of the (5.22) objective function into a monomial using (5.21) with weights α_i.

4. Solve the resulting GP using an interior point method.

5. Go to step 1 using $\mathbf{P}$ of step 4.

6. Terminate the kth loop if $\left\|\mathbf{P}^{(k)} - \mathbf{P}^{(k-1)}\right\| \leq \epsilon$ where ϵ is the error tolerance for exit condition.

As condensing the objective in the above problem gives us an underestimate of the objective value, each GP in the condensation iteration loop tries to improve the accuracy of the approximation to a particular minimum in the original feasible region. All three conditions for convergence are satisfied, and the algorithm is convergent. Empirically through extensive numerical experiments, we observe that it almost always computes the globally optimal power allocation.

Example 5.6. *Single condensation example.* We consider a wireless cellular network with three users. Let $T = 10^{-6}$ s, $G_{ii} = 1.5$, and generate G_{ij}, $i \neq j$, as independent random variables uniformly distributed between 0 and 0.3. Threshold SIR is $\mathsf{SIR}_{\text{th}} = -10$ dB, and minimal data rate requirements are 100 kbps, 600 kbps, and 1000 kbps for logical links 1, 2, and 3, respectively.

Maximal outage probabilities are 0.01 for all links, and maximal transmit powers are 3 mW, 4 mW, and 5 mW for links 1, 2, and 3, respectively. For each instance of problem (5.22), we pick a random initial feasible power vector $\mathbf{P}$ uniformly between 0 and $\mathbf{P}_{\max}$. Figure 5.10 compares the maximized total network throughput achieved over 500 sets of experiments with different initial vectors. With the (single) condensation method, SP converges to different optima over the entire set of experiments, achieving (or coming very close to) the global optimum at 5290 bps 96% of the time and a local optimum at 5060 bps 4% of the time. The average number of GP iterations required by the condensation method over the same set of experiments is 15 if an extremely tight exit condition is picked for SP condensation iteration: $\epsilon = 1 \times 10^{-10}$. This average can be substantially reduced by using a larger ϵ; for example, increasing ϵ to 1×10^{-2} requires on average only 4 GPs.

We have thus far discussed a power control problem (5.22) where the objective function needs to be condensed. The method is also applicable if some constraint functions are signomials and need to be condensed [14, 35].

5.3.5 Distributed Algorithm

A limitation for GP-based power control in ad hoc networks (without base stations) is the need for centralized computation (e.g., by interior point methods). The GP formulations of power control problems can also be solved by a new method of distributed algorithm for GP. The basic idea is that each user solves its own local optimization problem and the coupling among users is taken care of by message passing among the users. Interestingly, the special structure of coupling for the problem at hand (all coupling among the logical links can be lumped together using interference terms) allows one to further reduce the amount of message passing among the users. Specifically, we use a dual decomposition method to decompose a GP into smaller subproblems whose solutions are jointly and iteratively coordinated by the use of dual variables. The key step is to introduce auxiliary variables and to add extra equality constraints, thus transferring the coupling in the objective to coupling in the constraints, which can be solved by introducing "consistency pricing" (in contrast to "congestion pricing"). We illustrate this idea through an unconstrained GP followed by an application of the technique to power control.

Distributed Algorithm for GP

Suppose we have the following unconstrained standard form GP in $\mathbf{x} \succ 0$,

$$\text{minimize } \textstyle\sum_i f_i(x_i, \{x_j\}_{j \in I(i)}), \tag{5.23}$$

where x_i denotes the local variable of the ith user, $\{x_j\}_{j\in I(i)}$ denotes the coupled variables from other users, and f_i is either a monomial or posynomial. Making a change of variable $y_i = \log x_i, \forall i$, in the original problem, we obtain

$$\text{minimize } \textstyle\sum_i f_i(e^{y_i}, \{e^{y_j}\}_{j\in I(i)}).$$

We now rewrite the problem by introducing auxiliary variables y_{ij} for the coupled arguments and additional equality constraints to enforce consistency:

$$\begin{array}{l} \text{minimize } \ \textstyle\sum_i f_i(e^{y_i}, \{e^{y_{ij}}\}_{j\in I(i)}) \\ \text{subject to } y_{ij} = y_j, \forall j \in I(i), \forall i. \end{array} \tag{5.24}$$

Each ith user controls the local variables $(y_i, \{y_{ij}\}_{j\in I(i)})$. Next, the Lagrangian of (5.24) is formed as

$$\begin{aligned} L(\{y_i\}, \{y_{ij}\}; \{\gamma_{ij}\}) &= \sum_i f_i(e^{y_i}, \{e^{y_{ij}}\}_{j\in I(i)}) + \sum_i \sum_{j\in I(i)} \gamma_{ij}(y_j - y_{ij}) \\ &= \sum_i L_i(y_i, \{y_{ij}\}; \{\gamma_{ij}\}), \end{aligned}$$

where

$$L_i(y_i, \{y_{ij}\}; \{\gamma_{ij}\}) = f_i(e^{y_i}, \{e^{y_{ij}}\}_{j\in I(i)}) + \Bigg(\sum_{j:i\in I(j)} \gamma_{ji}\Bigg) y_i - \sum_{j\in I(i)} \gamma_{ij} y_{ij}. \tag{5.25}$$

The minimization of the Lagrangian with respect to the primal variables $(\{y_i\}, \{y_{ij}\})$ can be done simultaneously and distributively by each user in parallel. In the more general case where the original problem (5.23) is constrained, the additional constraints can be included in the minimization at each L_i.

In addition, the following master Lagrange dual problem has to be solved to obtain the optimal dual variables or consistency prices $\{\gamma_{ij}\}$,

$$\max_{\{\gamma_{ij}\}} g(\{\gamma_{ij}\}), \tag{5.26}$$

where

$$g(\{\gamma_{ij}\}) = \sum_i \min_{y_i, \{y_{ij}\}} L_i(y_i, \{y_{ij}\}; \{\gamma_{ij}\}).$$

Note that the transformed primal problem (5.24) is convex with zero duality gap; hence the Lagrange dual problem indeed solves the original standard GP problem. A simple way to solve the maximization in (5.26) is with the following subgradient update for the consistency prices,

$$\gamma_{ij}(t+1) = \gamma_{ij}(t) + \delta(t)(y_j(t) - y_{ij}(t)). \tag{5.27}$$

Appropriate choice of the stepsize $\delta(t) > 0$, for example, $\delta(t) = \delta_0/t$ for some constant $\delta_0 > 0$, leads to convergence of the dual algorithm.

Summarizing, the ith user has to: (i) minimize the function L_i in (5.25) involving only local variables, upon receiving the updated dual variables $\{\gamma_{ji}, j : i \in I(j)\}$, and (ii) update the local consistency prices $\{\gamma_{ij}, j \in I(i)\}$ with (5.27), and broadcast the updated prices to the coupled users.

Applications to Power Control

As an illustrative example, we maximize the total system throughput in the high SIR regime with constraints local to each user. If we directly applied the distributed approach described in the last section, the resulting algorithm would require knowledge by each user of the interfering channels and interfering transmit powers, which would translate into a large amount of message passing. To obtain a practical distributed solution, we can leverage the structures of power control problems at hand, and instead keep a local copy of each of the effective received powers $P_{ij}^R = G_{ij}P_j$. Again using problem (5.18) as an example formulation and assuming high SIR, we can write the problem as (after the log change of variable)

$$\begin{array}{l} \text{minimize} \quad \sum_i \log\left(G_{ii}^{-1}\exp(-\tilde{P}_i)\left(\sum_{j\neq i}\exp(\tilde{P}_{ij}^R)+\sigma^2\right)\right) \\ \text{subject to } \tilde{P}_{ij}^R = \tilde{G}_{ij} + \tilde{P}_j, \end{array} \tag{5.28}$$

Constraints are local to each user, for example, (a), (d), and (e) in Table 5.3. The partial Lagrangian is

$$\begin{aligned} L = &\sum_i \log\left(G_{ii}^{-1}\exp(-\tilde{P}_i)\left(\sum_{j\neq i}\exp(\tilde{P}_{ij}^R)+\sigma^2\right)\right) \\ &+ \sum_i \sum_{j\neq i} \gamma_{ij}\left(\tilde{P}_{ij}^R - \left(\tilde{G}_{ij}+\tilde{P}_j\right)\right), \end{aligned} \tag{5.29}$$

and the local ith Lagrangian function in (5.29) is distributed to the ith user, from which the dual decomposition method can be used to determine the optimal power allocation $\mathbf{P}^*$. The distributed power control algorithm is summarized as follows.

Algorithm 4. Distributed power allocation update to maximize R_{system}.

At each iteration t:

1. The ith user receives the term $\left(\sum_{j\neq i}\gamma_{ji}(t)\right)$ involving the dual variables from the interfering users by message-passing and minimizes the following local Lagrangian with respect to $\tilde{P}_i(t), \left\{\tilde{P}_{ij}^R(t)\right\}_j$ subject to the local constraints.

$$
\begin{aligned}
&L_i\left(\tilde{P}_i(t), \left\{\tilde{P}_{ij}^R(t)\right\}_j; \{\gamma_{ij}(t)\}_j\right) \\
&= \log\left(G_{ii}^{-1}\exp(-\tilde{P}_i(t))\left(\sum_{j\neq i}\exp(\tilde{P}_{ij}^R(t)) + \sigma^2\right)\right) \\
&\quad + \sum_{j\neq i}\gamma_{ij}\tilde{P}_{ij}^R(t) - \left(\sum_{j\neq i}\gamma_{ji}(t)\right)\tilde{P}_i(t).
\end{aligned}
$$

2. The ith user estimates the effective received power from each of the interfering users $P_{ij}^R(t) = G_{ij}P_j(t)$ for $j \neq i$, updates the dual variable by

$$
\gamma_{ij}(t+1) = \gamma_{ij}(t) + (\delta_0/t)\left(\tilde{P}_{ij}^R(t) - \log G_{ij}P_j(t)\right), \tag{5.30}
$$

and then broadcasts them by message passing to all interfering users in the system.

Example 5.7. *Distributed GP power control.* We apply the distributed algorithm to solve the above power control problem for three logical links with $G_{ij} = 0.2$, $i \neq j$, $G_{ii} = 1$, $\forall i$, maximal transmit powers of 6 mW, 7 mW, and 7 mW for links 1, 2, and 3 respectively. Figure 5.11 shows the convergence of the dual objective function towards the globally optimal total throughput of the network. Figure 5.12 shows the convergence of the two auxiliary variables in links 1 and 3 towards the optimal solutions.

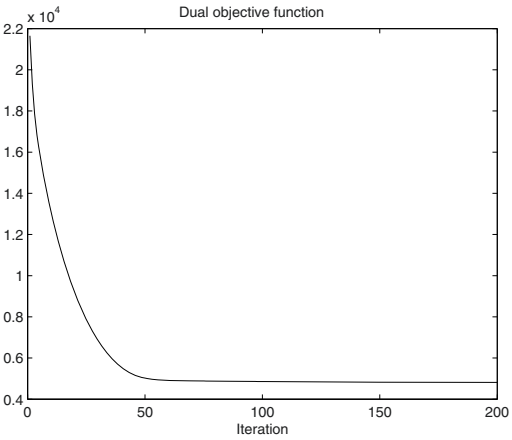

Fig. 5.11 Convergence of the dual objective function through distributed algorithm (Example 5.7).

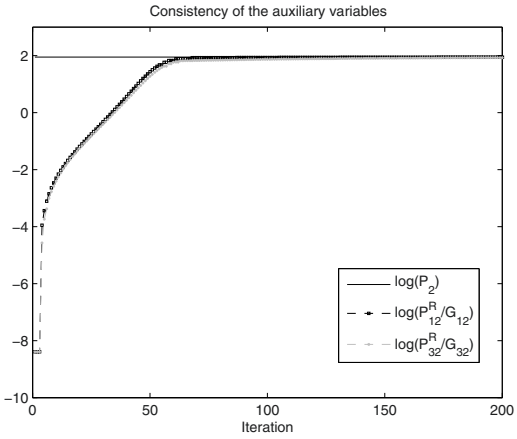

Fig. 5.12 Convergence of the consistency constraints through distributed algorithm (Example 5.7).

5.3.6 Concluding Remarks and Future Directions

Power control problems with nonlinear objective and constraints may seem to be difficult, NP-hard problems to solve for global optimality. However, when SIR is much larger than 0 dB, GP can be used to turn these problems into intrinsically tractable convex formulations, accommodating a variety of possible combinations of objective and constraint functions involving data rate, delay, and outage probability. Then interior point algorithms can efficiently compute the globally optimal power allocation even for a large network. Feasibility analysis of GP naturally leads to admission control and pricing schemes. When the high SIR approximation cannot be made, these power control problems become SP and may be solved by the heuristic of the condensation method through a series of GPs. Distributed optimal algorithms for GP-based power control in multihop networks can also be carried out through message passing.

Several challenging research issues in the low-SIR regime remain to be further explored. These include, for example, the reduction of SP solution complexity (e.g., by using high-SIR approximation to obtain the initial power vector and by solving the series of GPs only approximately except the last GP), and the combination of SP solution and distributed algorithm for distributed power control in low SIR regime. We also note that other approaches to tackle nonconvex power control problems have been studied, for example, the use of a particular utility function of rate to turn the problem into a convex one [28].

5.4 DSL Spectrum Management

5.4.1 Introduction

Digital subscriber line (DSL) technologies transform traditional voice-band copper channels into high-bandwidth data pipes, which are currently capable of delivering data rates up to several Mbps per twisted-pair over a distance of about 10 kft. The major obstacle for performance improvement in today's DSL systems (e.g., ADSL and VDSL) is *crosstalk*, which is the interference generated between different lines in the same binder. The crosstalk is typically 10–20 dB larger than the background noise, and direct crosstalk cancellation (e.g., [6, 27]) may not be feasible in many cases due to complexity issues or as a result of unbundling. To mitigate the detriments caused by crosstalk, *static spectrum management* which mandates spectrum mask or flat power backoff across all frequencies (i.e., tones) has been implemented in the current system.

Dynamic spectrum management (DSM) techniques, on the other hand, can significantly improve data rates over the current practice of static spectrum management. Within the current capability of the DSL modems, each modem has the capability to shape its own power spectrum density (PSD) across different tones, but can only treat crosstalk as background noise (i.e., no signal level coordination, such as vector transmission or iterative decoding, is allowed), and each modem is inherently a single-input–single-output communication system. The objective would be to optimize the PSD of all users on all tones (i.e., continuous power loading or discrete bit loading), such that they are "compatible" with each other and the system performance (e.g., weighted rate sum as discussed below) is maximized.

Compared to power control in wireless networks treated in the last section, the channel gains are not time-varying in DSL systems, but the problem dimension increases tremendously because there are many "tones" (or frequency carriers) over which transmission takes place. Nonconvexity still remains a major technical challenge, and high SIR approximation in general cannot be made. However, utilizing the specific structures of the problem (e.g., the interference channel gain values), an efficient and distributed heuristic is shown to perform close to the optimum in many realistic DSL network scenarios.

Following [7], this section presents a new algorithm for spectrum management in frequency selective interference channels for DSL, called autonomous spectrum balancing (ASB). It is the first DSL spectrum management algorithm that satisfies all of the following requirements for performance and complexity. It is autonomous (distributed algorithm across the users without explicit information exchange) with linear-complexity, while provably convergent, and comes close to the globally optimal rate region in practice. ASB

overcomes the bottlenecks in the state-of-the-art algorithms in DSM, including IW, OSB, and ISB summarized below.

Let K be the number of tones and N the number of users (lines). The *iterative waterfilling* (IW) algorithm [74] is among one of the first DSM algorithms proposed. In IW, each user views any crosstalk experienced as additive Gaussian noise, and seeks to maximize its data rate by "waterfilling" over the aggregated noise plus interference. No information exchange is needed among users, and all the actions are completely autonomous. IW leads to a great performance increase over the static approach, and enjoys a low complexity that is linear in N. However, the greedy nature of IW leads to a performance far from optimal in the near–far scenarios such as mixed CO/RT deployment and upstream VDSL.

To address this, an *optimal spectrum balancing* (OSB) algorithm [9] has been proposed, which finds the best possible spectrum management solution under the current capabilities of the DSL modems. OSB avoids the selfish behaviors of individual users by aiming at the maximization of a total weighted sum of user rates, which corresponds to a boundary point of the achievable rate region. On the other hand, OSB has a high computational complexity that is exponential in N, which quickly leads to intractability when N is larger than 6. Moreover, it is a completely centralized algorithm where a spectrum management center at the central office needs to know the *global information* (i.e., all the noise PSDs and crosstalk channel gains in the same binder) to perform the algorithm.

As an improvement to the OSB algorithm, an *iterative spectrum balancing* (ISB) algorithm [8] has been proposed, which is based on a weighted sum rate maximization similar to OSB. Different from OSB, ISB performs the optimization iteratively through users, which leads to a quadratic complexity in N. Close to optimal performance can be achieved by the ISB algorithm in most cases. However, each user still needs to know the global information as in OSB, thus ISB is still a centralized algorithm and is considered to be impractical in many cases.

This section presents the ASB algorithm [7], which attains near-optimal performance in an implementable way. The basic idea is to use the concept of a reference line to mimic a "typical" victim line in the current binder. By setting the power spectrum level to protect the reference line, a good balance between selfish and global maximizations can be achieved. The ASB algorithm enjoys a linear complexity in N and K, and can be implemented in a completely autonomous way. We prove the convergence of ASB under both sequential and parallel updates.

Table 5.4 compares various aspects of different DSM algorithms. Utilizing the structures of the DSL problem, in particular, the lack of channel variation and user mobility, is the key to provide a linear complexity, distributed, convergent, and almost optimal solution to this coupled, nonconvex optimization problem.

Table 5.4 Comparison of different DSM algorithms

Algorithm	Operation	Complexity	Performance	Reference
IW	Autonomous	$O(KN)$	Suboptimal	[74]
OSB	Centralized	$O(Ke^N)$	Optimal	[9]
ISB	Centralized	$O(KN^2)$	Near optimal	[8]
ASB	Autonomous	$O(KN)$	Near optimal	[7]

5.4.2 System Model

Using the notation as in [9, 8], we consider a DSL bundle with $\mathcal{N} = \{1, \ldots, N\}$ modems (i.e., lines, users) and $\mathcal{K} = \{1, \ldots, K\}$ tones. Assume discrete multitone (DMT) modulation is employed by all modems; transmission can be modeled independently on each tone as

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{z}_k.$$

The vector $\mathbf{x}_k = \{x_k^n, n \in \mathcal{N}\}$ contains transmitted signals on tone k, where x_k^n is the signal transmitted onto line n at tone k. $\mathbf{y}_k$ and $\mathbf{z}_k$ have similar structures. $\mathbf{y}_k$ is the vector of received signals on tone k. $\mathbf{z}_k$ is the vector of additive noise on tone k and contains thermal noise, alien crosstalk, single-carrier modems, radio frequency interference, and so on. $\mathbf{H}_k = [h_k^{n,m}]_{n,m \in \mathcal{N}}$ is the $N \times N$ channel transfer matrix on tone k, where $h_k^{n,m}$ is the channel from TX m to RX n on tone k. The diagonal elements of $\mathbf{H}_k$ contain the direct-channels whereas the off-diagonal elements contain the crosstalk channels. We denote the transmit power spectrum density (PSD) $s_k^n = \mathcal{E}\left\{|x_k^n|^2\right\}$. In the last section's notation for single-carrier systems, we would have $s_k^n = P_n$, $\forall k$. For convenience we denote the vector containing the PSD of user n on all tones as $\mathbf{s}^n = \{s_k^n, k \in \mathcal{K}\}$. We denote the DMT symbol rate as f_s.

Assume that each modem treats interference from other modems as noise. When the number of interfering modems is large, the interference can be well approximated by a Gaussian distribution. Under this assumption the achievable bit loading of user n on tone k is

$$b_k^n = \log\left(1 + \frac{1}{\Gamma} \frac{s_k^n}{\sum_{m \neq n} \alpha_k^{n,m} s_k^m + \sigma_k^n}\right), \tag{5.31}$$

where $\alpha_k^{n,m} = |h_k^{n,m}|^2 / |h_k^{n,n}|^2$ is the normalized crosstalk channel gain, and σ_k^n is the noise power density normalized by the direct channel gain $|h_k^{n,n}|^2$. Here Γ denotes the SINR-gap to capacity, which is a function of the desired BER, coding gain, and noise margin [61]. Without loss of generality, we assume $\Gamma = 1$. The data rate on line n is thus

$$R^n = f_s \sum_{k \in \mathcal{K}} b_k^n. \tag{5.32}$$

Each modem n is typically subject to a total power constraint P^n, due to the limitations on each modem's analogue frontend.

$$\sum_{k \in \mathcal{K}} s_k^n \leq P^n. \tag{5.33}$$

5.4.3 Spectrum Management Problem Formulation

One way to define the spectrum management problem is start with the following optimization problem.

$$\begin{array}{l} \text{maximize } R^1 \\ \text{subject to } R^n \geq R^{n,\text{target}}, \ \ \forall n > 1 \\ \qquad\qquad \sum_{k \in \mathcal{K}} s_k^n \leq P^n, \forall n. \end{array} \tag{5.34}$$

Here $R^{n,\text{target}}$ is the target rate constraint of user n. In other words, we try to maximize the achievable rate of user 1, under the condition that all other users achieve their target rates $R^{n,\text{target}}$. The mutual interference in (5.31) causes Problem (5.34) to be coupled across users on each tone, and the individual total power constraint causes Problem (5.34) to be coupled across tones as well.

Moreover, the objective function in Problem (5.34) is nonconvex due to the coupling of interference, and the convexity of the rate region cannot be guaranteed in general.

However, it has been shown in [75] that the duality gap between the dual gap of Problem (5.34) goes to zero when the number of tones K gets large (e.g., for VDSL), thus Problem (5.34) can be solved by the dual decomposition method, which brings the complexity as a function of K down to linear. Moreover, a frequency-sharing property ensures the rate region is convex with large enough K, and each boundary point of the boundary point of the rate region can be achieved by a weighted rate maximization as (following [9]),

$$\begin{array}{l} \text{maximize } R^1 + \sum_{n>1} w^n R^n \\ \text{subject to } \sum_{k \in \mathcal{K}} s_k^n \leq P^n, \forall n \in \mathcal{N}, \end{array} \tag{5.35}$$

such that the nonnegative weight coefficient w^n is adjusted to ensure that the target rate constraint of user n is met. Without loss of generality, here we define $w^1 = 1$. By changing the rate constraints $R^{n,\text{target}}$ for users $n > 1$ (or equivalently, changing the weight coefficients, w^n for $n > 1$), every boundary point of the convex rate region can be traced.

We observe that at the optimal solutions of (5.34), each user chooses a PSD level that leads to a good balance of maximization of its own rate and minimization of the damages it causes to the other users. To accurately calculate the latter, the user needs to know the global information of the noise PSDs and crosstalk channel gains. However, if we aim at a less aggressive objective and only require each user give enough protection to the other users in the binder while maximizing her own rate, then global information may not be needed. Indeed, we can introduce the concept of a "reference line", a virtual line that represents a "typical" victim in the current binder. Then instead of solving (5.34), each user tries to maximize the achievable data rate on the reference line, subject to its own data rate and total power constraint. Define the rate of the reference line to user n as

$$R^{n,\mathrm{ref}} = \sum_{k\in\mathcal{K}} \tilde{b}_k^n = \sum_{k\in\mathcal{K}} \log\left(1 + \frac{\tilde{s}_k}{\tilde{\alpha}_k^n s_k^n + \tilde{\sigma}_k}\right).$$

The coefficients $\{\tilde{s}_k, \tilde{\sigma}_k, \tilde{\alpha}_k^n, \forall k, n\}$ are parameters of the reference line and can be obtained from field measurement. They represent the conditions of a "typical" victim user in an interference channel (here a binder of DSL lines), and are known to the users a priori. They can be further updated on a much slower time scale through channel measurement data. User n then wants to solve the following problem local to itself,

$$\begin{array}{ll} \text{maximize} & R^{n,\mathrm{ref}} \\ \text{subject to} & R^n \geq R^{n,\mathrm{target}}, \\ & \sum_{k\in\mathcal{K}} s_k^n \leq P^n. \end{array} \tag{5.36}$$

By using Lagrangian relaxation on the rate target constraint in Problem (5.36) with a weight coefficient (dual variable) w^n, the relaxed version of (5.36) is

$$\begin{array}{ll} \text{maximize} & w^n R^n + R^{n,\mathrm{ref}} \\ \text{subject to} & \sum_{k\in\mathcal{K}} s_k^n \leq P^n. \end{array} \tag{5.37}$$

The weight coefficient w^n needs to be adjusted to enforce the rate constraint.

5.4.4 ASB Algorithms

We first introduce the basic version of the ASB algorithm (ASB-I), where each user n chooses the PSD $\mathbf{s}^n$ to solve (5.36) , and updates the weight coefficient w^n to enforce the target rate constraint. Then we introduce a variation of the ASB algorithm (ASB-II) that enjoys even lower computational complexity and provable convergence.

ASB-I

For each user n, replacing the original optimization (5.36) with the Lagrange dual problem

$$\max_{\lambda^n \geq 0,\, \sum_{k\in\mathcal{K}} s_k^n \leq P^n} \sum_{k\in\mathcal{K}} \max_{s_k^n} J_k^n \left(w^n, \lambda^n, s_k^n, s_k^{-n}\right), \tag{5.38}$$

where

$$J_k^n \left(w^n, \lambda^n, s_k^n, s_k^{-n}\right) = w^n b_k^n + \tilde{b}_k^n - \lambda^n s_k^n. \tag{5.39}$$

By introducing the dual variable λ^n, we decouple (5.36) into several smaller subproblems, one for each tone. And define J_k^n as user n's objective function on tone k. The optimal PSD that maximizes J_k^n for given w^n and λ^n is

$$s_k^{n,I} \left(w^n, \lambda^n, s_k^{-n}\right) = \arg \max_{s_k^n \in [0, P^n]} J_k^n \left(w^n, \lambda^n, s_k^n, s_k^{-n}\right), \tag{5.40}$$

which can be found by solving the first-order condition,

$$\partial J_k^n \left(w^n, \lambda^n, s_k^n, s_k^{-n}\right) / \partial s_k^n = 0,$$

which leads to

$$\frac{w^n}{s_k^{n,I} + \sum_{m\neq n} \alpha_k^{n,m} s_k^m + \sigma_k^n} - \frac{\tilde{\alpha}_k^n \tilde{s}_k}{\left(\tilde{s}_k + \tilde{\alpha}_k^n s_k^{n,I} + \tilde{\sigma}_k\right)\left(\tilde{\alpha}_k^n s_k^{n,I} + \tilde{\sigma}_k\right)} - \lambda^n = 0. \tag{5.41}$$

Note that (5.41) can be simplified into a cubic equation that has three solutions. The optimal PSD can be found by substituting these three solutions back to the objective function $J_k^n \left(w^n, \lambda^n, s_k^n, s_k^{-n}\right)$, as well as checking the boundary solutions $s_k^n = 0$ and $s_k^n = P^n$, and picking the one that yields the largest value of J_k^n.

The user then updates λ^n to enforce the power constraint, and updates w^n to enforce the target rate constraint. The complete algorithm is given as follows, where ε_λ and ε_w are small stepsizes for updating λ^n and w^n.

Algorithm 5. Autonomous Spectrum Balancing.

repeat
  for each user $n = 1, \ldots, N$
    repeat
      for each tone $k = 1, \ldots, K$, find
        $s_k^{n,I} = \arg\max_{s_k^n \geq 0} J_k^n$
      $\lambda^n = \left[\lambda^n + \varepsilon_\lambda \left(\sum_k s_k^{n,I} - P^n\right)\right]^+$;
      $w^n = \left[w^n + \varepsilon_w \left(R^{n,\text{target}} - \sum_k b_k^n\right)\right]^+$;
    until convergence
  end
until convergence

ASB-II with Frequency-Selective Waterfilling

To obtain the optimal PSD in ASB-I (for fixed λ^n and w^n), we have to solve the roots of a cubic equation. To reduce the computational complexity and gain more insights of the solution structure, we assume that the reference line operates in the high SIR regime whenever it is active: If $\tilde{s}_k > 0$, then $\tilde{s}_k \gg \tilde{\sigma}_k \gg \alpha_k^{n,m} s_k^n$ for any feasible s_k^n, $n \in \mathcal{N}$, and $k \in \mathcal{K}$. This assumption is motivated by our observations on optimal solutions in the DSL type of interference channels. It means that the reference PSD is much larger than the reference noise, which is in turn much larger than the interference from user n. Then on any tone $k \in \bar{\mathcal{K}} = \{k \mid \tilde{s}_k > 0, k \in \mathcal{K}\}$, the reference line's achievable rate is

$$\log\left(1 + \frac{\tilde{s}_k}{\tilde{\alpha}_k^n s_k^n + \tilde{\sigma}_k}\right) \approx \log\left(\frac{\tilde{s}_k}{\tilde{\sigma}_k}\right) - \frac{\tilde{\alpha}_k^n s_k^n}{\tilde{\sigma}_k},$$

and user n's objective function on tone k can be approximated by

$$J_k^{n,II,1}\left(w^n, \lambda^n, s_k^n, s_k^{-n}\right) = w^n b_k^n - \frac{\tilde{\alpha}_k^n s_k^n}{\tilde{\sigma}_k} - \lambda^n s_k^n + \log\left(\frac{\tilde{s}_k}{\tilde{\sigma}_k}\right).$$

The corresponding optimal PSD is

$$s_k^{n,II,1}\left(w^n, \lambda^n, s_k^{-n}\right) = \left(\frac{w_n}{\lambda^n + \tilde{\alpha}_k^n/\tilde{\sigma}_k} - \sum_{m \neq n} \alpha_k^{n,m} s_k^m - \sigma_k^n\right)^+. \tag{5.42}$$

This is a waterfilling type of solution and is intuitively satisfying: the PSD should be smaller when the power constraint is tighter (i.e., λ_n is larger), or the interference coefficient to the reference line $\tilde{\alpha}_k^n$ is higher, or the noise level on the reference line $\tilde{\sigma}_k$ is smaller, or there is more interference plus noise $\sum_{m \neq n} \alpha_k^{n,m} s_k^m + \sigma_k^n$ on the current tone. It is different from the conventional waterfilling in that the water level in each tone is not only determined by the dual variables w^n and λ^n, but also by the parameters of the reference line, $\tilde{\alpha}_k^n/\tilde{\sigma}_k$.

On the other hand, on any tone where the reference line is inactive, that is, $k \in \bar{\mathcal{K}}^C = \{k \mid \tilde{s}_k = 0, k \in \mathcal{K}\}$, the objective function is

$$J_k^{n,II,2}\left(w^n, \lambda^n, s_k^n, s_k^{-n}\right) = w^n b_k^n - \lambda^n s_k^n,$$

and the corresponding optimal PSD is

$$s_k^{n,II,2}\left(w^n, \lambda^n, s_k^{-n}\right) = \left(\frac{w^n}{\lambda^n} - \sum_{m \neq n} \alpha_k^{n,m} s_k^m - \sigma_k^n\right)^+. \tag{5.43}$$

This is the same solution as the iterative waterfilling.

The choice of optimal PSD in ASB-II can be summarized as the following.

$$s_k^{n,II}\left(w^n,\lambda^n,s_k^{-n}\right)=\begin{cases}\left(\frac{w^n}{\lambda^n+\tilde{\alpha}_k^n/\tilde{\sigma}_k}-\sum_{m\neq n}\alpha_k^{n,m}s_k^m-\sigma_k^n\right)^+, & k\in\bar{\mathcal{K}},\\ \left(\frac{w^n}{\lambda^n}-\sum_{m\neq n}\alpha_k^{n,m}s_k^m-\sigma_k^n\right)^+, & k\in\bar{\mathcal{K}}^C.\end{cases} \tag{5.44}$$

This is essentially a waterfilling type of solution, with different water levels for different tones (frequencies). We call it *frequency selective waterfilling*.

5.4.5 Convergence Analysis

In this section, we show the convergence for both ASB-I and ASB-II, for the case where users fix their weight coefficients w^n, which is also called *rate adaptive* (RA) spectrum balancing [61] that aims at maximizing users' rates subject to power constraint.

Convergence in the Two-User Case

The first result is on the convergence of ASB-I algorithm, with fixed $\mathbf{w}=\left(w^1,w^2\right)$ and $\boldsymbol{\lambda}=\left(\lambda^1,\lambda^2\right)$.

Proposition 5.3. *The ASB-I algorithm converges in a two-user case under fixed* $\mathbf{w}$ *and* $\boldsymbol{\lambda}$*, if users start from initial PSD values* $\left(s_k^1,s_k^2\right)=\left(0,P^2\right)$ *or* $\left(s_k^1,s_k^2\right)=\left(P^1,0\right)$ *on all tones.*

The proof of Proposition 5.3 uses supermodular game theory [65] and strategy transformation similar to [32].

Now consider the ASB-II algorithm where two users sequentially optimize their PSD levels under fixed values of $\mathbf{w}$, but adjust $\boldsymbol{\lambda}$ to enforce the power constraint. Denote $s_k^{n,t}$ as the PSD of user n in tone k after iteration t, where $\sum_k s_k^{n,t}=P^n$ is satisfied for any n and t. One iteration is defined as one round of updates of all users. We can show that

Proposition 5.4. *The ASB-II algorithm globally converges to the unique fixed point in a two-user system under fixed* $\mathbf{w}$*, if* $\max_k\alpha_k^{2,1}\max_k\alpha_k^{1,2}<1$.

The convergence result of iterative waterfilling in the two-user case [74] is a special case of Proposition 5.4 by setting $\tilde{s}_k=0,\forall k$.

We further extend the convergence results to a system with an arbitrary $N>2$ of users. We consider both sequential and parallel PSD updates of the users. In the more realistic but harder-to-analyze parallel updates, time is divided into slots, and each user n updates the PSD simultaneously in each time slot according to (5.44) based on the PSDs in the previous slot, where the λ^n is adjusted such that the power constraint is satisfied.

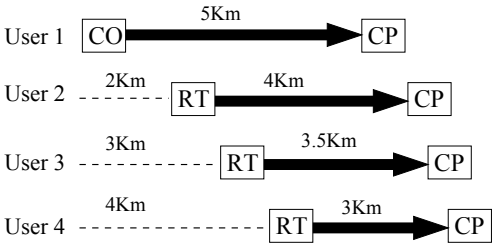

Fig. 5.13 An example of mixed CO/RT deployment topology (Example 5.8).

Proposition 5.5. *Assume* $\max_{n,m,k} \alpha_k^{n,m} < 1/(N-1)$*; then the ASB-II algorithm globally converges (to the unique fixed point) in an N-user system under fixed* $\mathbf{w}$*, with either sequential or parallel updates.*

Proposition 5.5 contains the convergence of iterative waterfilling in an N-user case with sequential updates (proved in [15]) as a special case of ASB convergence with sequential or parallel updates. Moreover, the convergence proof for the parallel updates turns out to be simpler than the one for sequential updates. The proof extends that of Proposition 5.4, and can be found in [7].

5.4.6 Simulation Results

Example 5.8. *Mixed CO-RT DSL.* Here we summarize a typical numerical example comparing the performances of ASB algorithms with IW, OSB, and ISB. We consider a standard mixed central office (CO) and remote terminal (RT) deployment. A four-user scenario has been selected to make a comparison with the highly complex OSB algorithm possible. As depicted in Figure 5.13 the scenario consists of one CO distributed line, and three RT distributed lines. The target rates on RT1 and RT2 have both been set to 2 Mbps. For a variety of different target rates on RT3, the CO line attempts to maximize its own data rate either by transmitting at full power in IW, or by setting its corresponding weight w_{co} to unity in OSB, ISB, and ASB.

This produces the rate regions shown in Figure 5.14, which shows that ASB achieves near optimal performance similar to OSB and ISB, and significant gain over IW even though both ASB and IW are autonomous. For example, with a target rate of 1 Mbps on CO, the rate on RT3 reaches 7.3 Mbps under the ASB algorithm, which is a 121% increase compared with the 3.3 Mbps achieved by IW. We have also performed extensive simulations (more than $10,000$ scenarios) with different CO and RT positions, line lengths, and reference line parameters. We found that the performance of ASB is

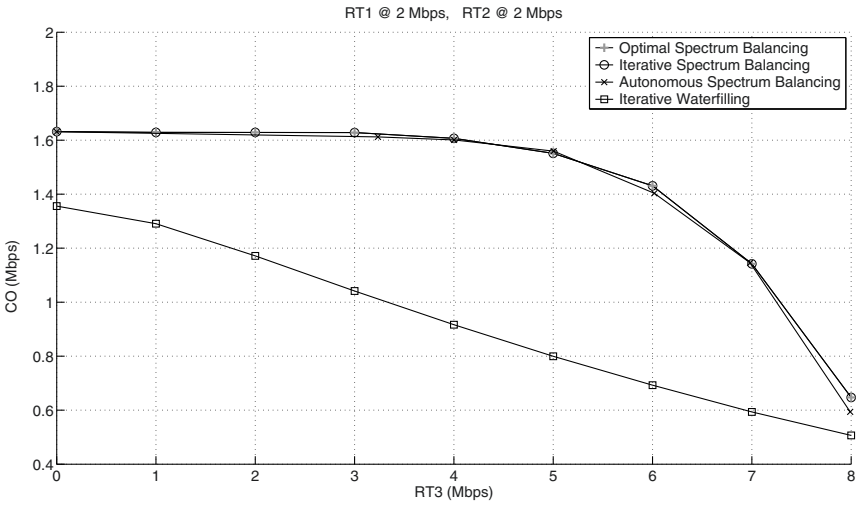

Fig. 5.14 Rate regions obtained by ASB, IW, OSB, and ISB (Example 5.8).

very insensitive to definition of the reference line: with a single choice of the reference line we observe good performance in a broad range of scenarios.

5.4.7 Concluding Remarks and Future Directions

Dynamic spectrum management techniques can greatly improve the performance of DSL lines by inducing cooperation among interfering users in the same binder. For example, the iterative waterfiling algorithm is a completely autonomous DSM algorithm with linear complexity in the number of users and number of tones, but the performance could be far from optimal in the mixed CO/RT deployment scenario. The optimal spectrum balancing and iterative spectrum balancing algorithms achieve optimal and close to optimal performances, respectively, but have high complexities in terms of the number of users and are completely centralized.

This section surveys an autonomous dynamic spectrum management algorithm called autonomous spectrum balancing. ASB utilizes the concept of "reference line", which mimics a typical victim line in the binder. By setting the power spectrum level to protect the reference line, a good balance between selfish and global maximizations can be achieved. Compared with IW, OSB, and ISB, the ASB algorithm enjoys completely autonomous operations, low (linear) complexity in both the number of users and number of tones. Simulation shows that the ASB algorithm achieves close to optimal performance and is robust to the choice of reference line parameters.

We conclude this section by highlighting the key ideas behind ASB. The reference line represents the statistical average of all victims within a typical network, and can be thought as a "static pricing". This differentiates the ASB algorithm with power control algorithms in the wireless setting, where pricing mechanisms have to be adaptive to the change of channel fading states and network topology, or Internet congestion control, where time-varying congestion pricing signals are used to align selfish interests for social welfare maximization. By using static pricing, no explicit message passing among the users is needed and the algorithm becomes autonomous across the users. This is possible because of the static nature of the channel gains in DSL networks.

Mathematically, the surprisingly good rate region results by ASB means that the specific engineering problem structures in this nonconvex and coupled optimization problem can be leveraged to provide a very effective approximation solution algorithm. Furthermore, robustness of the attained rate region with respect to perturbation of the reference line parameters has been verified to be very strong. This means that the dependence of the values of the local maxima of this nonconvex optimization problem on crosstalk channel coefficients is sufficiently insensitive for the observed robustness to hold.

There are several exciting further directions to pursue with ASB, for example, convergence conditions for ASB-I, extensions to intercarrier-interference cases, and bounds on optimality gap that are empirically verified to be very small. Interactions of ASB with link layer scheduling have resulted in further improvement of throughput in DSL networks [33, 67].

5.5 Internet Routing

5.5.1 Introduction

Most large IP (Internet protocol) networks run interior gateway protocols (IGPs) such as OSPF (open shortest path first) or IS-IS (intermediate system–intermediate system) that select paths based on link weights. Routers use these protocols to exchange link weights and construct a complete view of the topology inside the autonomous system (AS). Then, each router computes shortest paths (where the length of a path is the sum of the weights on the links) and creates a table that controls the forwarding of each IP packet to the next hop in its route. To handle the presence of multiple shortest paths, in practice, a router typically splits traffic roughly evenly over each of the outgoing links along a shortest path to the destination. The link weights are typically configured by the network operators or automated management systems, through centralized computation, to satisfy traffic-engineering goals, such as minimizing the maximum link utilization or the sum of link cost [24]. Following common practice, we use the the sum of some increasing and convex

link cost functions as the primary comparison metric and the optimization objective in this section.

Setting link weights under OSPF and IS-IS can be categorized as *link-weight-based* traffic engineering, where a set of link weights can uniquely and distributively determine the flow of traffic within the network for any given traffic matrix. The traffic matrix can be computed based on traffic measurements (e.g., [20]) or may represent explicit subscriptions or reservations from users. Link-weight-based traffic engineering has two key components: a *centralized* approach for setting the routing parameters (i.e., link weights) and a *distributed* way of using these link weights to decide the routes to forward packets. Setting the routing parameters based on a networkwide view of the topology and traffic, rather than the local views at each router, can achieve better performance [22].

Evaluation of various traffic engineering schemes, in terms of total link cost minimization, can be made against the performance benchmark of optimal routing (OPT), which can direct traffic along any paths in any proportion. The formulation can be found, for example, in [70]. OPT models an idealized routing scheme that can establish one or more explicit paths between every pair of nodes, and distribute an arbitrary amount of traffic on each of the paths.

It is easy to construct examples where OSPF, one of the most prevalent IP routing protocols today, with the best link weighting performs substantially (5000 times) worse than OPT in terms of minimizing sum of link cost. In addition, finding the best link weights under OSPF is NP-hard [24]. Although the best OSPF link weights can be found by solving an integer linear program (ILP) formulation, such an approach is impractical even for a midsize network. Many heuristics, including local search [24] and simulated annealing [5, 18] have been proposed to search for the best link weights under OSPF. Among them, local-search technique is the most attractive method in finding a good setting of the link weights for large-scale networks. Even though OSPF with a good setting of the weights performs within a few percent of OPT for some practical scenarios [24, 18, 5], there are still many realistic situations where the performance gap between OSPF and OPT could be significant even at low utilization.

There are two main reasons for the difficulty in tuning OSPF for good performance. First, the routing mechanism restricts the traffic to be routed only on shortest paths (and evenly split across shortest paths, an issue that has been addressed in [59]). Second, link weights and the traffic matrix are not integrated into the optimization formulation.

Both bottlenecks are overcome in the distributed exponentially weighted flow splitting (DEFT) protocol developed in [70]:

1. Traffic is allowed to be routed on nonshortest paths, with exponential penalty on path lengths.

2. An innovative optimization formulation is proposed, where both link weights and flows are variables. It leads to an effective two-stage iterative method.

As a result, DEFT, discussed in this section, has the following desirable properties.

- It determines a unique flow of traffic for a given link weight setting in polynomial time.
- It is provably always better than OSPF in terms of minimizing the maximum link utilization or the sum of link cost.
- It is readily implemented as an extension to the existing IGP (e.g., OSPF).
- The traffic engineering under DEFT with the two-stage iterative method realizes near-optimal flow of traffic even for large-scale network topologies.
- The optimizing procedure for DEFT converges much faster than that for OSPF.

In summary, DEFT provides a new way to compute link weights for OSPF that exceeds the current benchmark based on local search methods while reducing computational complexity at the same time. Furthermore, the performance turns out to be very close to the much more complicated and difficult to implement family of MPLS-type routing protocols, which allows arbitrary flow splitting.

More recently in [71], we have proved that a variation of DEFT, called PEFT, can provably achieve the optimal traffic engineering as a link-state routing protocol with hop-by-hop forwarding, with the optimal link weights computed in polynomial time and much faster than local search methods for link weight computation for OSPF. This answers the question on optimal traffic engineering by link state routing conclusively and positively.

5.5.2 DEFT: Framework and Properties

Given a directed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ with capacity $c_{u,v}$ for each link (u, v), let $D(s, t)$ denote the traffic demand originated from node s and destined for node t. $\Phi(f_{u,v}, c_{u,v})$ is a strictly increasing convex function of flow $f_{u,v}$ on link (u, v), typically a piecewise linear cost [24, 59] as shown in equation (5.45). The networkwide objective is to minimize $\sum_{(u,v)\in\mathbf{E}} \Phi(f_{u,v}, c_{u,v})$.

$$\Phi(f_{u,v}, c_{u,v}) = \begin{cases} f_{u,v} & f_{u,v}/c_{u,v} \le 1/3 \\ 3f_{u,v} - 2/3\, c_{u,v} & 1/3 \le f_{u,v}/c_{u,v} \le 2/3 \\ 10f_{u,v} - 16/3\, c_{u,v} & 2/3 \le f_{u,v}/c_{u,v} \le 9/10 \\ 70f_{u,v} - 178/3\, c_{u,v} & 9/10 \le f_{u,v}/c_{u,v} \le 1 \\ 500f_{u,v} - 1468/3\, c_{u,v} & 1 \le f_{u,v}/c_{u,v} \le 11/10 \\ 5000f_{u,v} - 16318/3\, c_{u,v} & 11/10 \le f_{u,v}/c_{u,v}. \end{cases} \tag{5.45}$$

In link-weight-based traffic engineering, each router u needs to make an independent decision on how to split the traffic destined for node t among its outgoing links only using link weights. Therefore, it calls for a function ($\Gamma(\cdot) \geq 0$) to represent the traffic allocation.

Shortest path routing (e.g., OSPF) evenly splits flow across all the outgoing links as long as they are on the shortest paths. First of all, we need a variable to indicate whether link (u, v) is on the shortest path to t. Denote $w_{u,v}$ as the weight for link (u, v), and d_u^t as the shortest distance from node u to node t; then $d_v^t + w_{u,v}$ is the distance from u to t when routed through v. The gap of the two above distances, $h_{u,v}^t = d_v^t + w_{u,v} - d_u^t$ is always larger than or equal to 0. Then (u, v) is on the shortest path to t if and only if $h_{u,v}^t = 0$. Accordingly, we can use a unit step function of $h_{u,v}^t$ to represent the traffic allocation for OSPF as follows.

$$\Gamma(h_{u,v}^t) = \begin{cases} 1, & \text{if } h_{u,v}^t = 0 \\ 0, & \text{if } h_{u,v}^t > 0. \end{cases} \tag{5.46}$$

The flow proportion on the outgoing link (u, v) destined for t at u is

$$\Gamma(h_{u,v}^t) / \sum_{(u,j)\in \mathbf{E}} \Gamma(h_{u,j}^t).$$

Denote $f_{u,v}^t$ as the flow on link (u, v) destined for node t and f_u^t as the flow sent along the shortest path of node u destined for t; then

$$f_{u,v}^t = f_u^t \, \Gamma(h_{u,v}^t). \tag{5.47}$$

The $\Gamma(h_{u,v}^t)$ function (5.46) (i.e., evenly splitting) results in intractability in searching for the best link weights under OSPF. In part inspired by Fong et al.'s work in [21], we can define a new $\Gamma(h_{u,v}^t)$ function to allow for flow on nonshortest paths. Intuitively, we may want to send more traffic on the shortest path than on a nonshortest path. Moreover, the traffic on a nonshortest path should be 0 if the distance gap between the nonshortest path and the shortest path is infinitely large. Based on the above intuition, $\Gamma(h_{u,v}^t)$ should be a strictly decreasing continuous function of $h_{u,v}^t$ bounded within $[0, 1]$. The exponential function is one of the natural choices, and the performance of using such function turns out to be excellent.

In [70], we propose an IGP with distributed exponentially weighted flow splitting:

$$\Gamma(h_{u,v}^t) = \begin{cases} e^{-h_{u,v}^t}, & \text{if } d_u^t > d_v^t \\ 0, & \text{otherwise;} \end{cases} \tag{5.48}$$

that is, the routers can direct traffic on nonshortest paths, with an exponential penalty on longer paths.

The following properties of DEFT can be proved [70].

Proposition 5.6. *DEFT can realize any acyclic flow for a single-destination demand within polynomial time. It can also achieve optimal routing with a single destination within polynomial time. For any traffic matrix, it can determine a unique flow for a given link weighting within polynomial time.*

Proposition 5.7. *DEFT is always better than OSPF in terms of minimizing total link cost or the maximum link utilization.*

5.5.3 DEFT: Optimization Formulation and Solutions

Note that it is still difficult to directly integrate the exponentially weighted flow splitting of DEFT into an optimization formulation because of its discrete feature; that is, the traffic destined for node t can be sent through link (u, v) if and only if $d_u^t > d_v^t$. Instead of introducing some binary variables, we relax (5.48) into (5.49) first, and then, by properly setting the lower bound of all link weights, a constant parameter $w_{\min}$, make such relaxation as tight as we want:

$$\Gamma(h_{u,v}^t) = e^{-h_{u,v}^t}. \tag{5.49}$$

Indeed, consider a flow solution satisfying (5.49); there is a link (u, v) where $d_v^t \geq d_u^t$ and $f_{u,v}^t > 0$, then $f_{u,v}^t \leq f_u^t\, e^{-h_{u,v}^t} = f_u^t\, e^{-(d_v^t + w_{u,v} - d_u^t)} \leq f_u^t\, e^{-w_{\min}}$. If $w_{\min}$ is large enough, this flow portion, which is infeasible to DEFT on link (u, v), could be neglected.

Therefore, we present the following optimization problem, called ORIG, using the relaxed rule of flow splitting as the approximation for the traffic engineering under DEFT.

$$\begin{aligned}
\text{minimize} \quad & \sum_{(u,v)\in\mathbf{E}} \Phi(f_{u,v}, c_{u,v}) && (5.50)\\
\text{subject to} \quad & \sum_{z:(y,z)\in\mathbf{E}} f_{y,z}^t - \sum_{x:(x,y)\in\mathbf{E}} f_{x,y}^t = D(y,t), \forall y \neq t && (5.51)\\
& f_{u,v} = \textstyle\sum_{t\in\mathbf{V}} f_{u,v}^t, && (5.52)\\
& h_{u,v}^t = d_v^t + w_{u,v} - d_u^t, && (5.53)\\
& f_{u,v}^t = f_u^t\, e^{-h_{u,v}^t}, && (5.54)\\
& f_u^t = \max_{(u,v)\in\mathbf{E}} f_{u,v}^t, && (5.55)\\
\text{variables} \quad & w_{u,v} \geq w_{\min}, f_u^t, d_u^t, h_{u,v}^t, f_{u,v}^t, f_{u,v} \geq 0. && (5.56)
\end{aligned}$$

Note that both the flow splittings and the link weights are incorporated as optimization variables in one problem, with further constraints relating them. Constraint (5.51) is to ensure flow conservation at an intermediate node y. Constraint (5.52) is for flow aggregation on each link. Constraint (5.53) is from the definition of gap of shortest distance. Constraints (5.54) and (5.55) come from (5.47) and (5.49). In addition, (5.54) and (5.55) also imply that

$f^t_{u,v} \le f^t_u$, and that $h^t_{u,v}$ of at least one of an outgoing links (u, v) of node u destined for node t should be 0; that is, the link (u, v) is on the shortest path from node u to node t.

Problem ORIG is nonsmooth and nonconvex due to nonsmooth constraint (5.55) and nonlinear equality (5.54). In [70], we propose a two-stage iterative relaxation to solve problem ORIG.

First, we relax constraint (5.55) into (5.57) below:

$$f^t_u \le \sum\nolimits_{(u,v)\in\mathbf{E}} f^t_{u,v}, \quad \forall t \in \mathbf{V}, \quad \forall u \in \mathbf{V}. \tag{5.57}$$

Equations (5.50)–(5.54), (5.56), and (5.57) constitute problem APPROX.

We only need to obtain a "reasonably" accurate solution (link weighting $\mathbf{W}$) to problem APPROX because the inaccuracy caused by the relaxation (5.57) will be compensated by a successive refinery process later. From the $\mathbf{W}$, we can derive the shortest path tree $\mathbf{T}(\mathbf{W}, t)$[6] for each destination t, and all other dependent variables $(d^t_u, h^t_{u,v}, f^t_u, f^t_{u,v}, f_{u,v})$ within DEFT.

We then use these values as the initial point (which is also strictly feasible) for a new problem REFINE, which consists of equations (5.50)–(5.54), (5.56), and (5.58) below:

$$f^t_u = f^t_{u,v}, \quad \forall t \in \mathbf{V} \cap \forall u \in \mathbf{V} \cap (u, v) \in \mathbf{T}(\mathbf{W}, t). \tag{5.58}$$

With the two-stage iterative method, we are left with two optimization problems, APPROX and REFINE, both of which have convex objective functions and twice continuously differentiable constraints. To solve the large-scale nonlinear problems APPROX and REFINE (with $O(|V||E|)$ variables and constraints), we extend the primal–dual interior point filter line search algorithm, IPOPT [68], by solving a set of barrier problems for a decreasing sequences of barrier parameters μ converging to 0.

In summary, in solving problem APPROX, we mainly want to determine the shortest path tree for each destination (i.e., deciding which outgoing link should be chosen on the shortest path). Then in solving problem REFINE, we can tune the link weights (and the corresponding flow) with the same shortest path trees as in APPROX.

The pseudocode of the proposed two-stage iterative method for DEFT is shown in Algorithms 6A and 6B. Most instructions are self-explanatory. Function DEFT_FLOW($\mathbf{W}$) is used to derive a flow from a set of link weights $\mathbf{W}$. Given the initial and ending values for barrier parameter μ, maximum iteration number, with/without initial link weighting/flow, function DEFT_IPOPT() returns a new set of link weights as well as a new flow. Note that, as shown in Algorithm 6B, when DEFT_IPOPT() is used for problem APPROX, it returns with the last iteration rather than the iteration with the best $\mathbf{Flow}_i$ in terms of the objective value as in problem REFINE. This

[6] To keep $\mathbf{T}(\mathbf{W}, t)$ as a tree, only one downstream node is chosen if a node can reach the destination through several downstream nodes with the same distance.

is because problem APPROX has different constraints from problem ORIG and a too greedy method may leave small search freedom for the successive REFINE problem. Finally, we need to specify initial and terminative μ values, ($\mu_{\text{init}} \geq \mu_{\text{end\_approx}} \geq \mu_{\text{end\_refine}}$), and maximum iteration number $\text{Iter}_{\text{approx}} \geq \text{Iter}_{\text{refine}}$. As shown in the next section, it is straightforward to specify these parameters.

Algorithm 6A. DEFT Solution.

1. $(\mu, \mathbf{W}) \leftarrow \text{DEFT\_IPOPT}(\mu_{\text{init}}, \mu_{\text{end\_approx}}, \text{Iter}_{\text{approx}}, \mathbf{nil})$
2. $\text{Initial\_Point} \leftarrow (\mathbf{W}, \text{DEFT\_FLOW}(\mathbf{W}))$
3. $(\mu, \mathbf{W}) \leftarrow \text{DEFT\_IPOPT}(\mu, \mu_{\text{end\_refine}}, \text{Iter}_{\text{refine}}, \text{Initial\_Point})$
4. Return $(\mathbf{W}, \text{DEFT\_FLOW}(\mathbf{W}))$

Algorithm 6B. DEFT IPOPT.

If Initial_Point $\neq$ **nil** Then
Initiate the problem with Initial_Point /*REFINE*/
End if
For **each** iteration $i \leq \text{Iter}_{\max}$ with $\mu_{\text{start}} \geq \mu \geq \mu_{\text{end}}$ do
$\mu_i \leftarrow$ current value for μ
$\mathbf{W}_i \leftarrow$ current values for all $w_{u,v}$
$\mathbf{Flow}_i \leftarrow \text{DEFT\_FLOW}(\mathbf{W}_i)$
end for
If Initial_Point = **nil** then
return $(\mu_i, \mathbf{W}_i)$ of the last iteration /*APPROX*/
else
return $(\mu_i, \mathbf{W}_i)$ of the iteration with the best $\mathbf{Flow}_i$ in terms of objective value /*REFINE*/
end if

5.5.4 Numerical Examples

We summarize some of the numerical results in [70] on various schemes under many practical scenarios. We employ the same cost function (5.45) as in [23]. The primary metric used is the optimality gap, in terms of total link cost, compared against the value achieved by optimal routing using CPLEX 9.1 [16] via AMPL [25]. The secondary metric used is the maximum link utilization. We do not reproduce the performance of some obvious link-weight-based traffic engineering approaches for OSPF, for example, UnitOSPF (setting all link weights to 1), RandomOSPF (choosing the weights randomly), InvCapOSPF (setting the weight of an link inversely proportional to its capacity as recommended by Cisco), or L2OSPF (setting the weight proportional to its physical Euclidean distance) [23], because none of them performs as well as

the state-of-the-art local search method proposed in [23]. In addition, because DEFT is always better than OSPF in terms of minimizing the maximum link utilization or the sum of link cost, we bypass the scenarios where OSPF can achieve near-optimal solution. Instead, we are particularly interested in those scenarios where OSPF does not perform well.

For fair comparisons, we use the same topology and traffic matrix as those in [23]. The 2-level hierarchical networks were generated using GT-ITM, which consists of two kinds of links: local access links with 200-unit capacity and long distance links with 1000-unit capacity. In the second type of topology, the random topologies, the probability of having a link between two nodes is a constant parameter and all link capacities are 1000 units.

Although AT&T's proprietary code of local search used in [23] is not publicly available, there is an open source software project with IGP weight optimization, TOTEM 1.1 [66]. It follows the same lines as [23], and has similar quality of the results. It is slightly slower due to the lack of implementation of the dynamic Dijkstra algorithm. We use the same parameter setting for local search as in [24, 23] where link weight is restricted as an integer from 1 to 20, initial link weights are chosen randomly, and the best result is collected after 5000 iterations.

To implement the proposed two-stage iterative method for DEFT, we modify another open source software, IPOPT 3.1 [34], and adjust its AMPL interface to integrate it into our test environment. We choose $\mu_{\text{init}} = 0.1$ for most cases except for $\mu_{\text{init}} = 10$ for the 100-node network with heavy traffic load. We also choose $\mu_{\text{end\_approx}} = 10^{-4}$, $\mu_{\text{end\_refine}} = 10^{-9}$, and maximum iteration number $\text{Iter}_{\text{approx}} = 1000$, $\text{Iter}_{\text{refine}} = 400$. The code terminates earlier if the optimality gap has been less than 0.1%.

Example 5.9. *DEFT and OSPF on 2-level topology.* The results for a 2-level topology with 50 nodes and 212 links with seven different traffic matrices are shown in Table 5.5. The results are also depicted graphically in Figure 5.15.

Table 5.5 Results of 2-level topology with 50 nodes and 212 links

Total Traffic Demand	1700	2000	2200	2500	2800	3100	3400
Ave Link Load-OPT	0.128	0.148	0.17	0.192	0.216	0.242	0.267
Max Link Load-OPT	0.667	0.667	0.667	0.9	0.9	0.9	0.9
Opt. Gap-OSPF	2.8%	4.4%	7.2%	9.4%	20.7%	64.2%	222.8%
Opt. Gap-DEFT	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%

In addition to the two metrics, optimality gap in terms of total link cost and maximum link utilization,[7] we also show the average link utilization under optimal routing as an indication of network load. From the results, we

[7] Note, however, maximum link utilization is not a metric as comprehensive as total link cost because it cannot indicate whether there are multiple overcongested links.

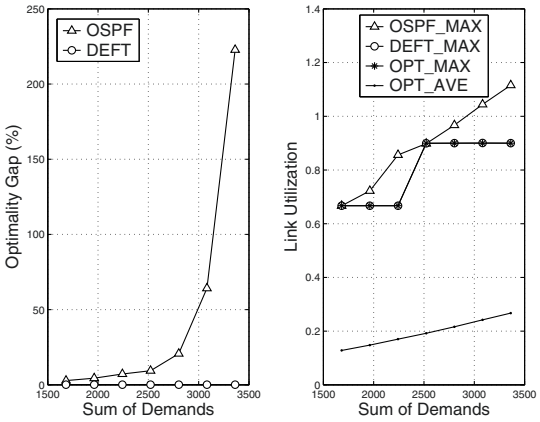

Fig. 5.15 Comparison of DEFT and local search OSPF in terms of optimality gap and maximum link utilization for a 2-level topology with 50 nodes and 212 links (Example 5.9).

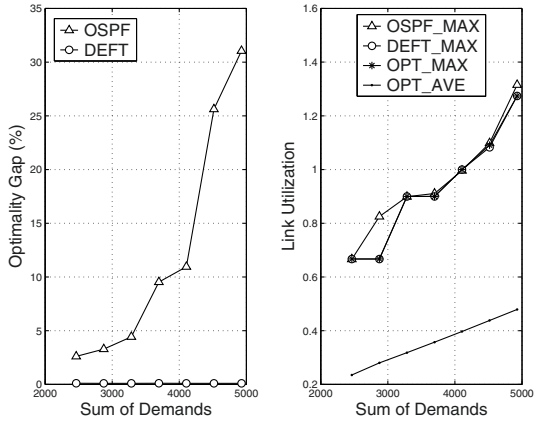

Fig. 5.16 2-level topology with 50 nodes and 148 links (Example 5.9).

can observe that the gap between OSPF and optimal routing can be very significant (up to 222.8%) for a practical network scenario, even when the average link utilization is low ($\leq$27%). In contrast, DEFT can achieve almost the same performance as the optimal routing in terms of both total link cost and maximum link utilization.

Example 5.10. *DEFT and OSPF on random topology.* Similar observations can be found for other scenarios, for example, as shown in Figure 5.17. Without exception, the curves of the DEFT scheme (the horizontal lines almost coinciding with x-axes) almost completely overlap those of optimal routing,

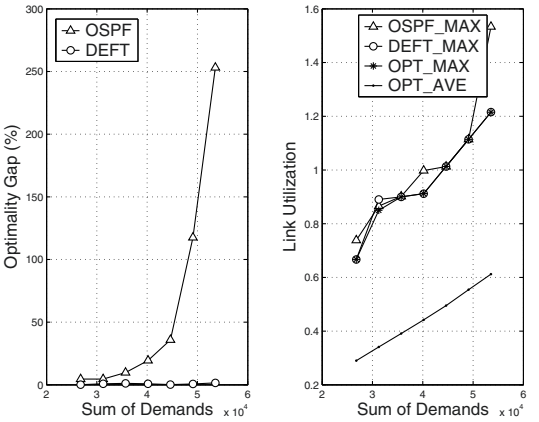

Fig. 5.17 Random topology with 50 nodes and 245 links (Example 5.10).

in terms of total link cost and maximum link utilization. Among these numerical experiments, the maximum optimality gap of OSPF is as high as 252% and that of DEFT is only at worst 1.5%. In addition, DEFT reduces the maximum link utilization compared to OSPF on all tests, and substantially on some tests.

Simulations on rate of convergence, as well as comparisons of computation and implementation complexity, can be found in [70].

5.5.5 Concluding Remarks and Future Directions

Network operators today try to alleviate congestion in their own network by tuning the parameters in IGP. Unfortunately, traffic engineering under OSPF or IS-IS to avoid networkwide congestion is computationally intractable, forcing the use of local-search techniques. While staying within the context of link-weight-based traffic engineering, we propose a new protocol called [70] distributed exponentially weighted flow splitting. DEFT significantly outperforms the state-of-the-art OSPF local search mechanisms in minimizing networkwide congestion. The success of DEFT can be attributed to two additional features. First, DEFT can put traffic on nonshortest paths, with an exponential penalty on longer paths. Second, DEFT solves the resulting optimization problem by integrating link weights and the corresponding traffic distribution together in the formulation. The novel formulation leads to a much more efficient way of tuning link-weight than the existing local search heuristic for OSPF.

DEFT is readily implementable as an extension to existing IGPs. It is provably always better than OSPF in minimizing the sum of link cost. DEFT retains the simplicity of having routers compute paths based on configurable link weights, while approaching the performance of the much more complex routing protocols that can split traffic arbitrarily over any paths. In summary, in terms of minimizing total link cost, performance of OSPF by local search heuristics is at best what is attained by solving the ILP, which is substantially outperformed by DEFT that comes very close to the optimal routing. In terms of a performance–complexity tradeoff, DEFT clearly exceeds OSPF.

In this section, we only address the link weighting under DEFT for a given traffic matrix. The next challenge would be to explore robust optimization under DEFT, optimizing to select a single weight setting that works for a range of traffic matrices and/or a range of link/node failure scenarios. Extension of the ideas behind DEFT to routing across different autonomous systems managed by different network operators is another interesting future direction.

In the larger picture of "design for optimizability", DEFT shows one case where by changing the underlying protocol, the resulting new optimization formulation becomes much more readily solvable or approximable. We expect this new approach to tackle nonconvex problems to bring many new results and insights to the engineering of communication networks. Indeed, in an extension of DEFT work [71], we have developed the first provably optimal link state routing protocol with hop by hop forwarding, called PEFT, which achieves optimal traffic engineering with polynomial time (and very fast in practice) computation of optimal link weights.

Acknowledgments The author would like to acknowledge collaborations with Raphael Cendrillon, Maryam Fazel, Prashanth Hande, Jianwei Huang, Daniel Palomar, Jennifer Rexford, Chee Wei Tan, and Dahai Xu while working on the five publications related to this survey [29, 19, 14, 7, 70], as well as very helpful discussions on these topics with Stephen Boyd, Rob Calderbank, John Doyle, David Gao, Jiayue He, David Julian, Jang-Won Lee, Ying Li, Steven Low, Marc Moonen, Daniel O'Neill, Asuman Ozdaglar, Pablo Parrilo, Ness Shroff, R. Srikant, Ao Tang, and Shengyu Zheng. The work reported in this chapter has been supported in part by the following grants: AFOSR FA9550-06-1-0297, DARPA W911NF-07-1-0057, NSF CNS-0519880 and CNS-0720570, and ONR N00014-07-1-0864.

References

1. M. Avriel, Ed., *Advances in Geometric Programming,* Plenum Press, New York, 1980.
2. N. Bambos, "Toward power-sensitive network architectures in wireless communications: Concepts, issues, and design aspects," *IEEE Pers. Commun. Mag.*, vol. 5, no. 3, pp. 50–59, 1998.
3. S. Boyd, S. J. Kim, L. Vandenberghe, and A. Hassibi, "A tutorial on geometric programming," *Optim. Eng.*, vol. 8, no. 1, pp. 67–127, 2007.
4. S. Boyd and L. Vandenberghe, *Convex Optimization,* Cambridge University Press, 2004.

5. L. Buriol, M. Resende, C. Ribeiro, and M. Thorup, "A memetic algorithm for OSPF routing," *Proc. 6th INFORMS Telecom*, 2002, pp. 187–188.
6. R. Cendrillon, G. Ginis, and M. Moonen, "Improved linear crosstalk precompensation for downstream VDSL," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004.
7. R. Cendrillon, J. Huang, M. Chiang, and M. Moonen, "Autonomous Spectrum Balancing (ASB) for digital subscriber loop," *IEEE Trans. Signal Process.,* vol. 55, no. 8, pp. 4241–4257, August 2007.
8. R. Cendrillon and M. Moonen, "Iterative spectrum balancing for digital subscriber lines," in *Proc. IEEE International Communications Conference*, 2005.
9. R. Cendrillon, W. Yu, M. Moonen, J. Verlinden, and T. Bostoen, "Optimal multi-user spectrum management for digital subscriber lines," *IEEE Trans. Commun.*, July 2006.
10. M. Chiang, "Balancing transport and physical layers in wireless multihop networks: Jointly optimal congestion control and power control," *IEEE J. Selected Areas Commun.,* vol. 23, no. 1, pp. 104–116, January 2005.
11. M. Chiang, "Geometric programming for communication systems," *Foundations Trends Commun. Inf. Theor.*, vol. 2, no. 1–2, pp. 1–156, August 2005.
12. M. Chiang, P. Hande, T. Lan, and C. W. Tan, "Power control in wireless cellular networks," *Foundations and Trends in Networking*, vol. 2, no. 4, pp. 381–533, July 2008.
13. M. Chiang, S. H. Low, R. A. Calderbank, and J. C. Doyle, "Layering as optimization decomposition," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 255–312, January 2007.
14. M. Chiang, C. W. Tan, D. Palomar, D. O'Neill, and D. Julian, "Power control by geometric programming," *IEEE Trans. Wireless Commun.*, vol. 6, no. 7, pp. 2640–2651, July 2007.
15. S. T. Chung, "Transmission schemes for frequency selective Gaussian interference channels," Ph.D. dissertation, Stanford University, 2003.
16. ILOG CPLEX, http://www.ilog.com/products/cplex/.
17. R. J. Duffin, E. L. Peterson, and C. Zener, *Geometric Programming: Theory and Applications*, Wiley, 1967.
18. M. Ericsson, M. Resende, and P. Pardalos, "A genetic algorithm for the weight setting problem in OSPF routing," *J. Combin. Optim.*, vol. 6, pp. 299–333, 2002.
19. M. Fazel and M. Chiang, "Nonconcave network utility maximization by sum of squares programming," *Proc. IEEE Conference on Decision and Control*, December 2005.
20. A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True, "Deriving traffic demands for operational IP networks: Methodology and experience," *IEEE/ACM Trans. Netw.*, June 2001.
21. J. H. Fong, A. C. Gilbert, S. Kannan, and M. J. Strauss, "Better alternatives to OSPF routing," *Algorithmica*, vol. 43, no. 1–2, pp. 113–131, 2005.
22. B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional IP routing protocols," *IEEE Commun. Mag.*, October 2002.
23. B. Fortz and M. Thorup, "Increasing internet capacity using local search," *Comput. Optim. Appl.*, vol. 29, no. 1, pp. 13–48, 2004.
24. B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," *Proc. IEEE INFOCOM*, May 2000.
25. R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, Thomson, Danvers, MA, 1993.
26. G. Foschini and Z. Miljanic, "A simple distributed autonomous power control algorithm and its convergence," *IEEE Trans. Vehicular Technol.*, vol. 42, no. 4, 1993.
27. G. Ginis and J. Cioffi, "Vectored transmission for digital subscriber line systems," *IEEE J. Selected Areas Commun.*, vol. 20, no. 5, pp. 1085–1104, 2002.
28. P. Hande, S. Rangan, M. Chiang, and X. Wu, "Distributed uplink power control for optimal SIR assignment in cellular data networks," To appear in *IEEE/ACM Trans. Netw.*, 2008.

29. P. Hande, S. Zhang, and M. Chiang, "Distributed rate allocation for inelastic flows," *IEEE/ACM Trans. Netw.*, vol. 15, no. 6, pp. 1240–1253, December 2007.
30. D. Handelman, "Representing polynomials by positive linear functions on compact convex polyhedra," *Pacific J. Math.*, vol. 132, pp. 35–62, 1988.
31. D. Henrion and J. B. Lasserre, "Detecting global optimality and extracting solutions in GloptiPoly," Research report, LAAS-CNRS, 2003.
32. J. Huang, R. Berry, and M. L. Honig, "A game theoretic analysis of distributed power control for spread spectrum ad hoc networks," *Proc. IEEE International Symposium of Information Theory*, July 2005.
33. J. Huang, C. W. Tan, M. Chiang, and R. Cendrillon, "Statistical multiplexing over DSL networks," *Proc. IEEE INFOCOM*, May 2007.
34. IPOPT, http://projects.coin-or.org/Ipopt.
35. D. Julian, M. Chiang, D. ONeill, and S. Boyd, "QoS and fairness constrained convex optimization of resource allocation for wireless cellular and ad hoc networks," *Proc. IEEE INFOCOM*, June 2002.
36. F. P. Kelly, "Models for a self-managed Internet," *Philosoph. Trans. Royal Soc.*, A358, 2335–2348, 2000.
37. 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, March 1998.
38. S. Kandukuri and S. Boyd, "Optimal power control in interference limited fading wireless channels with outage probability specifications," *IEEE Trans. Wireless Commun.*, vol. 1, no. 1, pp. 46–55, January 2002.
39. J. B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM J. Optim.*, vol. 11, no. 3, pp. 796–817, 2001.
40. J. B. Lasserre, "Polynomial programming: LP-relaxations also converge," *SIAM J. Optim.*, vol. 15, no. 2, pp. 383–393, 2004.
41. J. W. Lee, R. Mazumdar, and N. B. Shroff, "Non-convex optimization and rate control for multi-class services in the Internet," *IEEE/ACM Trans. Netw.*, vol. 13, no. 4, pp. 827–840, August 2005.
42. J. W. Lee, R. Mazumdar, and N. B. Shroff, "Downlink power allocation for multi-class CDMA wireless networks", *IEEE/ACM Trans. Netw.*, vol. 13, no. 4, pp. 854–867, August 2005.
43. J. W. Lee, R. Mazumdar, and N. B. Shroff, "Opportunistic power scheduling for multi-server wireless systems with minimum performance constraints," *IEEE Trans. Wireless Commun.*, vol. 5, no. 5, May 2006.
44. X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE J. Selected Areas Commun.*, August 2006.
45. S. H. Low, "A duality model of TCP and queue management algorithms," *IEEE/ACM Trans. Netw.*, vol. 11, no. 4, pp. 525–536, August 2003.
46. B. R. Marks and G. P. Wright, "A general inner approximation algorithm for nonconvex mathematical program," *Oper. Res.*, 1978.
47. D. Mitra, "An asynchronous distributed algorithm for power control in cellular radio systems," *Proc. 4th WINLAB Workshop*, Rutgers University, NJ, 1993.
48. Yu. Nesterov and A. Nemirovsky, *Interior Point Polynomial Methods in Convex Programming,* SIAM Press, 1994.
49. D. Palomar and M. Chiang, "Alternative distributed algorithms for network utility maximization: Framework and applications," *IEEE Trans. Autom. Control*, vol. 52, no. 12, pp. 2254–2269, December 2007.
50. P. A. Parrilo, Structured semidefinite programs and semi-algebraic geometry methods in robustness and optimization," PhD thesis, Caltech, May 2002.
51. P. A. Parrilo, "Semidefinite programming relaxations for semi-algebraic problems," *Math. Program.*, vol. 96, pp. 293–320, 2003.

52. S. Prajna, A. Papachristodoulou, and P. A. Parrilo, "SOSTOOLS: Sum of squares optimization toolbox for Matlab," available from `http://www.cds.caltech.edu/sostools`, 2002–04.
53. M. Putinar, "Positive polynomials on compact semi-algebraic sets," *Indiana Univ. Math. J.*, vol. 42, no. 3, pp. 969–984, 1993.
54. R. T. Rockafellar, *Network Flows and Monotropic Programming*, Athena Scientific, 1998.
55. C. Saraydar, N. Mandayam, and D. Goodman, "Pricing and power control in a multicell wireless data network", *IEEE J. Selected Areas Commun.*, vol. 19, no. 10, pp. 1883–1892, October 2001.
56. K. Schmüdgen, "The K-moment problem for compact semi-algebraic sets," *Math. Ann.*, vol. 289, no. 2, pp. 203–206, 1991.
57. S. Shenker, "Fundamental design issues for the future Internet," *IEEE J. Selected Areas Commun.*, vol. 13, no. 7, pp. 1176–1188, September 1995.
58. N. Z. Shor, "Quadratic optimization problems," *Soviet J. Computat. Syst. Sci.*, vol. 25, pp. 1–11, 1987.
59. A. Sridharan, R. Guérin, and C. Diot, "Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 2, pp. 234–247, 2005.
60. R. Srikant, *The Mathematics of Internet Congestion Control*, Birkhäuser 2004.
61. T. Starr, J. Cioffi, and P. Silverman, *Understanding Digital Subscriber Line Technology*, Prentice Hall, 1999.
62. G. Stengle, "A Nullstellensatz and a Positivstellensatz in semialgebraic geometry," *Math. Ann.*, vol. 207, no. 2, pp. 87–97, 1974.
63. C. Sung and W. Wong, "Power control and rate management for wireless multimedia CDMA systems," *IEEE Trans. Commun.*, vol. 49, no. 7, pp. 1215–1226, 2001.
64. C. W. Tan, D. Palomar and M. Chiang, "Distributed Optimization of Coupled Systems with Applications to Network Utility Maximization," *Proc. IEEE International Conference of Acoustic, Speech, and Signal Processing*, May 2006.
65. D. M. Topkis, *Supermodularity and Complementarity*, Princeton University Press, 1998.
66. TOTEM, http://totem.info.ucl.ac.be.
67. P. Tsiaflakis, Y. Yi, M. Chiang, and M. Moonen, "Throughput and delay of DSL dynamic spectrum management," *Proc. IEEE GLOBECOM*, December 2008.
68. A. Wächter and L. T. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, 2006.
69. M. Xiao, N. B. Shroff, and E. K. P. Chong, "Utility based power control in cellular wireless systems," *IEEE/ACM Trans. Netw.*, vol. 11, no. 10, pp. 210–221, March 2003.
70. D. Xu, M. Chiang, and J. Rexford, "DEFT: Distributed exponentially-weighted flow splitting," *Proc. IEEE INFOCOM*, May 2007.
71. D. Xu, M. Chiang, and J. Rexford, "Link state routing with hop by hop forwarding can achieve optimal traffic engineering," *Proc. IEEE INFOCOM*, April 2008.
72. R. Yates, "A framework for uplink power control in cellular radio systems," *IEEE J. Selected Areas Commun.*, vol. 13, no. 7, pp. 1341–1347, 1995.
73. Y. Yi, A. Proutiere, and M. Chiang, "Complexity of wireless scheduling: Impact and tradeoffs," *Proc. ACM Mobihoc*, May 2008.
74. W. Yu, G. Ginis, and J. Cioffi, "Distributed multiuser power control for digital subscriber lines," *IEEE J. Selected Areas Commun.*, vol. 20, no. 5, pp. 1105–1115, June 2002.
75. W. Yu, R. Lui, and R. Cendrillon, "Dual optimization methods for multiuser orthogonal frequency division multiplex systems," *Proc. IEEE GLOBECOM*, November 2004.