Foundations and Trends® in
Networking
Vol. 6, Nos. 1-2 (2011) 1–163

DOI: 10.1561/1300000036

now
the essence of knowledge

Weighted Sum-Rate Maximization in Wireless Networks: A Review

By Pradeep Chathuranga Weeraddana, Marian Codreanu, Matti Latva-aho, Anthony Ephremides and Carlo Fischione

Contents

Foundations and Trends® in
Networking
Vol. 6, Nos. 1-2 (2011) 1–163

DOI: 10.1561/1300000036

now
the essence of knowledge

Weighted Sum-Rate Maximization in Wireless Networks: A Review

Pradeep Chathuranga Weeraddana[1], Marian Codreanu[2], Matti Latva-aho[3], Anthony Ephremides[4] and Carlo Fischione[5]

[1] *Automatic Control Lab, KTH Royal Institute of Technology, Stockholm, 100-44, Sweden, chatw@kth.se*
[2] *Centre for Wireless Communications, University of Oulu, Oulu, 90014, Finland, codreanu@ee.oulu.fi*
[3] *Centre for Wireless Communications, University of Oulu, Oulu, 90014, Finland, matti.latva-aho@ee.oulu.fi*
[4] *University of Maryland, College Park, MD, 20742, USA, etony@ece.umd.edu*
[5] *Automatic Control Lab, KTH Royal Institute of Technology, Stockholm, 100-44, Sweden, carlofi@kth.se*

Abstract

A wide variety of resource management problems of recent interest, including power/rate control, link scheduling, cross-layer control, network utility maximization, beamformer design of multiple-input multiple-output networks, and many others are directly or indirectly reliant on the weighted sum-rate maximization (WSRMax) problem. In general, this problem is very difficult to solve and is NP-hard. In this review, we provide a cohesive discussion of the existing solution

methods associated with the WSRMax problem, including global, fast local, as well as decentralized methods. We also discuss in depth the applications of general optimization techniques, such as branch and bound methods, homotopy methods, complementary geometric programming, primal decomposition methods, subgradient methods, and sequential approximation strategies, in order to develop algorithms for the WSRMax problem. We show, through a number of numerical examples, the applicability of these algorithms in various application domains.

1

Introduction

Consider a general wireless network with L interfering links. The achievable rate of each link is a scalar function and is denoted by r_l. Then the general weighted sum-rate maximization (WSRMax) problem has the form:

$$\begin{array}{ll} \text{maximize} & \sum_{l=1}^{L} \beta_l r_l(\mathbf{y}) \\ \text{subject to} & \mathbf{y} \in \mathcal{Y}. \end{array} \tag{1.1}$$

Here $\mathbf{y} = (y_1, \ldots, y_n)$ is the optimization variable of the problem, positive scalar β_l is the weight associated with link l, and the (possibly nonconvex) set $\mathcal{Y}$ is the feasible set of the problem. In general, r_l is not convex in $\mathbf{y}$. Therefore, problem (1.1) is surprisingly *difficult* to solve, though it appears to be very simple.

In this section, we first provide a discussion that emphasizes the importance of WSRMax problem (1.1) in wireless networks. Next we discuss the importance of global, fast local, as well as distributed solution methods for WSRMax problem. Finally, the existing key literature that address the problem is presented. Applications of optimization techniques for developing algorithms for the problem will be covered in later sections, with more technical detail.

1.1 Motivation

Among various resource management policies, the WSRMax for an arbitrary set of interfering links plays a central role in many network control and optimization methods. For example, the problem is encountered in network utility maximization (NUM) [88], the resource allocation (RA) subproblem in various cross-layer control policies [43, 81], MaxWeight link scheduling in multihop wireless networks [120], power/rate allocation in wireless networks, as well as in wireline networks [115, 124], joint optimization of transmit beamforming patterns, transmit powers, and link activations in multiple-input multiple-output (MIMO) networks [34], and finding achievable rate regions of singlecast/multicast wireless networks [87], among others.

1.1.1 Network Utility Maximization (NUM)

In the late nineties, Kelly et al. [59, 60] introduced the concept of NUM for fairness control in wireline networks. It was shown therein that maximizing the sum-rate under the fairness constraint is equivalent to maximizing certain network utility functions and different network utility functions can be mapped to different fairness criteria. For a useful discussion of many aspects of the NUM concept in the case of wireless network, see [88] and the references therein. In this context, the WSRMax problem appears as a part of the Lagrange dual problem of the overall NUM problem; see [89] and the references therein.

1.1.2 Cross-layer Control Policies for Wireless Networks

For useful discussions of cross-layer control policies, see [40, 43, 68, 71, 81, 90, 142] and the references therein. Many of these policies are essentially identical. It has been shown that an optimal cross-layer control policy, which achieves data rates arbitrarily close to the optimal operating point, can be decomposed into three subproblems that are normally associated with different network layers. Specifically, *flow control* resides at the transport layer, *routing and in-node scheduling*[1] resides

[1] *in-node scheduling* refers to selecting the appropriate commodity and it is not to be confused with the links scheduling mechanism which is handled by the resource allocation subproblem [43].

at the network layer, and *resource allocation* (or RA) is usually associated with the medium access control and physical layers [43]. The first two subproblems are convex optimization problems and can be solved relatively easily. It turns out that under reasonably mild assumptions, the RA subproblem can be cast as a general WSRMax problem over the instantaneous achievable rate region [43]. The weights of the links are given by the differential backlogs and the policy resembles the well-known *backpressure algorithm* introduced by Tassiulas and Ephremides in [120, 121] and further extended by Neely to dynamic networks with power control; see [81] and the references therein.

1.1.3 MaxWeight Link Scheduling for Wireless Networks

Maximum weighted link scheduling for wireless networks [41, 68, 105, 120, 121, 138] is a place, in which the problem of WSRMax is directly used. Note that, for networks with fixed link capacities, the maximum weighted link scheduling problem reduces to the classical maximum weighted matching problem and can be solved in polynomial time [68]. However, no solution is known for the general case when the link rates depend on the power allocation of all other links.

1.1.4 Power/rate Control Policies

We see sometimes that the WSRMax problem is directly used as the basis for the power/rate control policy in wireless, as well as in wireline networks [115, 124]. For example, in DSL networks, there is considerable research on resource management policies, which rely directly on the WSRMax problem for multiuser spectrum balancing [3, 27, 28, 39, 74, 94, 97, 127, 128, 129, 139, 140]. Direct application of WSRMax as an optimization criterion can also been seen extensively in joint power control and subcarrier assignment algorithms for OFDMA networks [7, 50, 54, 67, 104, 106, 145].

1.1.5 Resource Management in MIMO Networks

There are also a number of resource management algorithms in multiuser MIMO networks, which rely on the problem of WSRMax. For example, the methods proposed in [34, 35, 111, 122] rely on WSRMax

for joint design of linear transmit and receive beamformers. In addition, many references have applied WSRMax directly as an optimization criterion for beamformer design in MIMO networks, e.g., [45, 151].

1.1.6 Finding Achievable Rate Regions in Wireless Networks

In multiuser systems many users share the same network resources, for example, time, frequency, codes, space, etc. Thus, there is naturally a tradeoff between the achievable rates of the users. In other words, one may require to reduce its rate if another user wants a higher rate. In such multiuser systems, the achievable rate regions are important since they characterize the tradeoff achievable by any resource management policy [124]. By invoking a time sharing argument, one can always assume that the rate region is convex [124]. Therefore, any boundary point of the rate region can be obtained by using the solution of a WSRMax problem for some weights.

Thus, WSRMax is a central component in many network design problems as we discussed above. Unfortunately, the general WSRMax problem is not yet amendable to a convex formulation [76]. In fact, it is NP-hard [75]. Therefore, we must rely on global optimization approaches [5, 52] for computing an exact solution of the WSRMax problem. Such global solution methods are increasingly important because they can be used to provide performance benchmarks by back-substituting them into any network design method, which relies on WSRMax. They are also very useful for evaluating the performance loss encountered by any heuristic algorithm for the WSRMax problem.

Although global methods find the solution of the WSRMax problem, they are typically slow. Even small problems, with a few tens of variables, can take a very long time to solve WSRMax. Therefore, it is natural to seek suboptimal algorithms for WSRMax that are efficient enough, and still close to optimal; the compromise is optimality [22]. Such algorithms are of central importance since they can be fast and widely applicable in large-scale network control and optimization methods.

Due to the explosion of problem size and the signal overhead required in centralized network control and optimization methods, it is highly desirable to develop decentralized variants of those algorithms. Therefore, finding distributed methods for the WSRMax problem is of crucial importance from a theoretical, as well as from a practical perspective for decentralized implementation of many network control and optimization methods, such as those investigated in [81, 120].

1.2 Global Methods for WSRMax in Wireless Networks

The general WSRMax problem is NP-hard [75]. It is therefore natural to rely on global optimization approaches [5, 52] for computing an exact solution. One straightforward approach is based on exhaustive search in the variable space [28]. The main disadvantage of this approach is the prohibitively expensive computational complexity, even in the case of very small problems. A better approach is to apply branch and bound techniques [52], which essentially implement the exhaustive search in an intelligent manner; see [3, 39, 57, 97, 129, 135, 139]. Branch and bound methods based on difference of convex functions (DC) programming [52] have been proposed in [3, 39, 139] to solve (a subclass of) WSRMax. Although DC programming is the core of their algorithms, it also limits the generality of their method to the problems in which the objective function cannot easily be expressed as a DC [52]. For example, in the case of multicast wireless networks, expressing the objective function as a DC cannot be easily accomplished, even when Shannon's formula is used to express the achievable link rates. Another branch and bound method has been used in [129] in the context of DSL bit loading, where the search space is discretized in advance. As a result of discretization, this method does not allow a complete control of the accuracy of the solution. An alternative optimal method was proposed in [97], where the WSRMax problem is cast as a generalized linear fractional program [96] and solved via a polyblock algorithm [48]. The method works well for small scale problems, but as pointed out in [5, chap. 2, pp. 40–41] and [96, sec. 6.3], it may show much slower convergence than branch and bound methods as the problem size increases. A special form of the WSRMax problem is presented

in [23, p. 78] [128], where the problem data and the constraints must obey certain properties and, consequently, the problem can be reduced to a convex formulation. However, these required properties correspond to very unlikely events in wireless/wireline networks, and therefore the method has a very limited applicability.

1.3 Local Methods for WSRMax in Wireless Networks

The worst case computational complexity for solving the general WSRMax problem by applying global optimization approaches can increase more than polynomially with the number of variables. As a result, these methods are prohibitively expensive, even for off line optimization of moderate size networks. Therefore, the problem of WSRMax deserves efficient algorithms, which even though suboptimal, perform well in practice.

Several approximations have been proposed for the case when all links in the network operate in certain signal-to-interference-plus-noise ratio (SINR) regions. For example, the assumption that the achievable rate is a linear function of the SINR (i.e., low SINR region) is widely used in the ultra-wide-band systems, e.g., [98]. Other references, which provide solutions for the power and rate control in low SINR regions include [37, 69, 99]. The high SINR region is treated in [29, 58, 86]. However, at the optimal operating point different links correspond to different SINR regions, which is usually the case with multihop networks. Therefore, all methods mentioned above that are based on either the low or the high SINR assumptions can fail to solve the general problem.

One promising method is to cast the WSRMax problem into a signomial program (SP) formulation [20, sec. 9] or into a complementary geometric program (CGP) [6, 30], where a suboptimal solution can be obtained efficiently; we can readily convert an SP to a CGP and vice versa [30, sec. 2.2.5]. Applications of SP and CGP, or closely related solution methods, have been demonstrated in various signal processing and digital communications problems, e.g., [30, 31, 34, 35, 78, 94, 122, 127]. There are a number of other important papers proposing suboptimal solution methods for the WSRMax

problem, such as [2, 27, 33, 45, 68, 70, 74, 105, 111, 116, 131, 151], among others.

Though the suboptimal methods mentioned above, including SP/CGP based algorithms, can perform reasonably well in many cases,[2] it is worth pointing out that not all of them can handle the general WSRMax problem. The reason is the self-interference problem, which arises when a node transmits and receives simultaneously in the same frequency band. Since there is a huge imbalance between the transmitted signal power and the received signal power of nodes, the transmitted signal strength is typically few orders of magnitude larger than the received signal strength. Thus, when a node transmits and receives simultaneously in the same channel, the useful signal at the receiver of the incoming link is overwhelmed by the transmitted signal of the node itself. As a result, the SINR values at the incoming link of a node that simultaneously transmits in the same channel is very small. Therefore, the self-interference problem plays a central role in WSRMax in general wireless networks [133].

Thus, in the case of general multihop wireless networks, the WSRMax problem must also cope with the self-interference problem. Under such circumstances SP/CGP cannot be directly applicable even to obtain a better suboptimal solution, since initialization of the algorithms is critical. One approach to dealing with self interference consists of adding supplementary combinatorial constraints, which prevent any node in the network from transmitting and receiving simultaneously [13, 14, 24, 38, 46, 61, 71, 138]. This is sometimes called the node-exclusive interference model; only subsets of mutually exclusive links can simultaneously be activated in order to avoid the large self interference encountered if a node transmits and receives in the same frequency band. Of course, such approaches induce a combinatorial nature for the WSRMax problem in general. The combinatorial nature is circumvented in [134], where *homotopy methods* (or continuation methods) [4] together with complementary geometric programming [6] are adopted to derive efficient algorithms for the general WSRMax

[2] For example, when a node does not transmits and receives simultaneously in the same frequency band.

problem. Here, the term "efficient" can mean faster convergence, or convergence to a point with better objective value.

1.4 Distributed WSRMax in Wireless Networks

The emergence of large scale communication networks, as well as accompanying network control and optimization methods with huge signalling overheads triggered a considerable body of recent research on developing distributed algorithms for resource management, see [21, 79, 141] and the references therein. Such distributed algorithms rely only on local observations and are carried out with limited access to global information. These algorithms essentially involve coordinating many local subproblems to find a solution to a large global problem. It is worth emphasizing that the convexity of the problems is crucial in determining the behavior of the distributed algorithms [21, chap. 9]. For example, in the case of nonconvex problems such algorithms need not converge, and if they do converge, they need not converge to an optimal point, which is the case with the WSRMax problem. Nevertheless, finding even a suboptimal yet distributed method is crucial for deploying distributively many network control and optimization methods, e.g., [41, 68, 81, 83, 84, 105, 118, 120, 138], which rely on WSRMax.

Distributed implementation of the WSRMax problem has been investigated in [27, 93, 94, 127, 144] in the context of digital subscriber loops (DSL) networks. Those systems are inherently consisting of single-input and single-output (SISO) links. Related algorithms for SISO wireless ad hoc networks and SISO orthogonal frequency division multiple access cellular systems are found in [53, 117, 147, 146]. However, in the case of multi antenna cellular systems, the decision variables space is, of course, larger, for example, joint optimization of transmit beamforming patterns, transmit powers, and link activations is required. Therefore, designing efficient distributed methods for WSRMax is a more challenging task due to the extensive amount of message passing required to resolve the coupling between variables. In the sequel, we limit ourselves to basic, but still very important, results that develop distributed coordinated algorithms for resource management in networks with multiple antennas.

Several distributed methods for WSRMax in multiple-input and single-output (MISO) cellular networks have been proposed in [10, 11, 72, 95, 130, 132, 136]. Specifically, in [95] a two-user MISO interference channel (IC)[3] is considered and a distributed algorithm is derived by using the commonly used high SINR approximation [29]. Moreover, another approximation, which relies on zero forcing (ZF) beamforming is introduced in [95] to address the problem in the case of multiuser MISO IC.

The methods proposed in [10, 11, 130] derived the necessary (but not sufficient) optimality conditions for the WSRMax problem and used it as the basis for their distributed solution. However, many parameters must be selected heuristically to construct a potential distributed solution and there is, in general no systematic method for finding those parameters. In particular, the algorithms in [10, 11] are designed for systems with very limited backhaul signaling resources and do not consider any iterative base station (BS) coordination mechanism to resolve the out-of-cell interference coupling. Even though the method proposed in [130] relies on stringent requirements on the message passing between BSs during each iteration of the algorithm, their results show that BS coordination can provide considerable gains compared to uncoordinated methods. An inexact cooperate descent algorithm for the case where each BS is serving only one cell edge user has been proposed in [72]. The method proposed in [66] is designed for sum-rate maximization and uses high SINR approximation. A cooperative beamforming algorithm is proposed in [152] for MISO IC, where each BS can transmit only to a single user. Their proposed method employs an iterative BS coordination mechanism to resolve the out-of-cell interference coupling. However, the convexity properties exploited for distribution of the problem are destroyed when more than one user is served by any BS. Thus, their proposed method is not directly applicable to the WSRMax problem. Recently, an interesting distributed algorithm for WSRMax is proposed by Shi et al. [110], which exploits a nontrivial equivalence between the WSRMax problem and a weighted sum mean

[3] K-user MISO IC means that there are K transmitter–receiver pairs, where the transmitters have multiple antennas and the receivers have single antennas.

squared error minimization problem. This algorithm relies on user terminal assistance, such as signal covariance estimations, computation and feedback of certain parameters form user terminals to BSs over the *air interface*. In practice, performing perfect covariance estimation and perfect feedback during each iteration can be very challenging. In the presence of user terminal imperfections, such as estimation and feedback errors, the algorithms performance can degrade and its convergence can be less predictable.

Algorithms based on game theory are found in [49, 55, 65, 107, 108, 109]. Their proposed methods are restricted to interference channels, for example, MISO IC, MIMO IC. The methods often require the coordination between receiver nodes and the transmitter nodes during algorithm's iterations.

Many optimization criteria other than the weighted sum-rate have been considered in references [12, 113, 123, 143, 148, 149, 150] to distributively optimize the system resources (e.g., beamforming patterns, transmit powers, etc.) in multi antenna cellular networks. In particular, the references [12, 148, 149, 150] used the characterization of the Pareto boundary of the MISO interference channel [56] as the basis for their distributed methods. Their proposed methods do not employ any BS coordination mechanism to resolve the out-of-cell interference coupling. In [113, 123, 143] distributed algorithms have been derived to minimize a total (weighted) transmitted power or the maximum per antenna power across the BSs subject to SINR constraints at the user terminals.

1.5 Outline of the Volume

Section 2 presents a solution method, based on the branch and bound technique, which solves globally the nonconvex WSRMax problem with an optimality certificate. Efficient analytic bounding techniques are introduced and their impact on convergence is numerically evaluated. The considered link-interference model is general enough to model a wide range of network topologies with various node capabilities, for example, single- or multipacket transmission

(or reception), simultaneous transmission and reception. Diverse application domains of WSRMax are considered in the numerical results, including cross-layer network utility maximization and maximum weighted link scheduling for multihop wireless networks, as well as finding achievable rate regions for singlecast/multicast wireless networks.

Section 3 presents fast suboptimal algorithms for the WSRMax problem in multicommodity, multichannel wireless networks. First, the case where all receivers perform singleuser detection[4] is considered and algorithms are derived by applying complementary geometric programming and homotopy methods. Here we apply the algorithms within a general cross-layer utility maximization framework to examine quantitative impact of gains that can be achieved at the network layer in terms of end-to-end rates and network congestion. In addition, we show, through examples, that the algorithms are well suited for evaluating the gains achievable at the network layer when the network nodes employ self interference cancelation techniques with different degrees of accuracy. Finally, a case where all receivers perform multiuser detection is considered and solutions are presented by imposing additional constraints, such as that only one node can transmit to others at a time or that only one node can receive from others at a time.

Section 4 presents an easy to implement distributed method for the WSRMax problem in a multicell multiple antenna downlink system. The algorithm is based on primal decomposition and subgradient methods, where the original nonconvex problem is split into a master problem and a number of subproblems (one for each base station). A sequential convex approximation strategy is used to address the nonconvex master problem. Unlike the recently proposed minimum weighted mean-squared error based algorithms, the method presented here *does not* rely on any user terminal assistance. Only base station to base station synchronized signalling via backhaul links. All the necessary computation is performed at the BSs. Numerical experiments

[4] That is, a receiver decodes each of its intended signals by treating all other interfering signals as noise.

are provided to examine the behavior of the algorithm under different degrees of BS coordination.

Finally, in Section 5, we present our conclusions. The detailed work presented in this volume is based on the research performed by the authors that led to several recent journal and conference publications.

2

A Branch and Bound Algorithm

In this section we present a branch and bound method for solving globally the general WSRMax problem for a set of interfering links. At each step, the algorithm computes upper and lower bounds for the optimal value. The algorithm terminates when the difference between the upper and the lower bounds is within a pre-specified accuracy level. Efficient analytic bounding techniques are introduced and their impact on the convergence is numerically evaluated. The considered link-interference model is general enough to model a wide range of network topologies with various node capabilities, for example, single- or multipacket transmission (or reception), simultaneous transmission and reception. In contrast to the other branch and bound based techniques [3, 39, 139], the method presented here does not rely on the convertibility of the problem into a DC problem. Therefore, it applies to a broader class of WSRMax problems (e.g., WSRMax in multicast wireless networks). Moreover, the method discussed here is not restricted to WSRMax; it can also be used to maximize *any* system performance metric that can be expressed as a Lipschitz continuous and increasing function of SINR values.

The branch and bound method given in this section shows some analogy to the one proposed in [129] in terms of the initial search domain and the basic bounding techniques. However, the two methods are fundamentally different in terms of branching techniques, as the algorithm proposed in [129] is designed specifically to search over a discrete space whilst the method here is optimized for a continuous search space. We also provide improved bounding techniques which substantially improve the convergence speed of the algorithm.

Given its generality, the algorithm can be adapted to a wide range of network control and optimization problems. Performance benchmarks for various network topologies can be obtained by back-substituting it into any network design method which relies on WSRMax. Several applications, including cross-layer network utility maximization and maximum weighted link scheduling for multihop wireless networks, as well as finding achievable rate regions for singlecast/multicast wireless networks, are presented. As suboptimal but low-complex algorithms are typically used in practice, the given algorithm can also be used for evaluating their performance loss.

2.1 System Model and Problem Formulation

The network considered consists of a collection of nodes which can send, receive, and relay data across a set of links. The set of all nodes is denoted by $\mathcal{N}$ and we label the nodes with the integer values $n = 1, \ldots, N$. A link is represented as an ordered pair (i, j) of distinct nodes. The set of all links is denoted by $\mathcal{L}$ and we label the links with the integer values $l = 1, \ldots, L$. We define $\text{tran}(l)$ as the transmitter node of link l, and $\text{rec}(l)$ as the receiver node of link l. The existence of a link $l \in \mathcal{L}$ implies that a direct transmission is possible from node $\text{tran}(l)$ to node $\text{rec}(l)$. Note that, in the most general case, $\mathcal{L}$ may consist of a combination of wireless and wireline links, for example, in the case of hybrid networks. We define $\mathcal{O}(n)$ as the set of links that are outgoing from node n, and $\mathcal{I}(n)$ as the set of links that are incoming to node n. Furthermore, we denote the set of transmitter nodes by $\mathcal{T}$ and the set of receiver nodes by $\mathcal{R}$, i.e., $\mathcal{T} = \{n \in \mathcal{N} | \mathcal{O}(n) \neq \emptyset\}$ and $\mathcal{R} = \{n \in \mathcal{N} | \mathcal{I}(n) \neq \emptyset\}$.

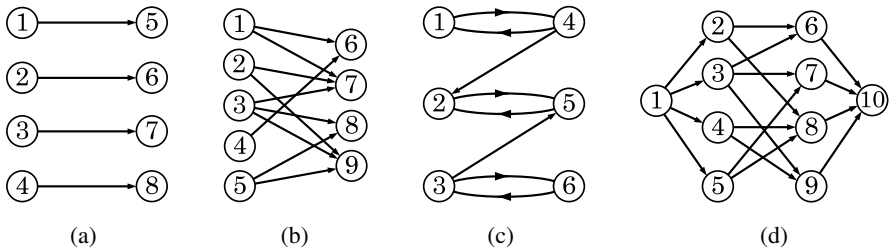


Fig. 2.1 Various network topologies: (a) Bipartite network, $\mathcal{T} = \{1,2,3,4\}$, $\mathcal{R} = \{5,6,7,8\}$, degree 1; (b) Bipartite network, $\mathcal{T} = \{1,2,3,4,5\}$, $\mathcal{R} = \{6,7,8,9\}$, degree 3; (c) Nonbipartite singlehop network, $\mathcal{T} = \mathcal{R} = \mathcal{N} = \{1,2,3,4,5,6\}$, degree 3; (d) Nonbipartite multihop network, $\mathcal{T} = \{1,2,\ldots,9\}$, $\mathcal{R} = \{2,3,\ldots,10\}$, $\mathcal{T} \cap \mathcal{R} = \{2,3,\ldots,9\}$, degree 4.

The model above covers a wide range of network topologies from very simple ones to more complicated ones, as shown in Figure 2.1. A particular class of network topologies is the one for which the set of transmitters $\mathcal{T}$ and the set of receivers $\mathcal{R}$ are disjoint and we refer to these networks as *bipartite* networks. Figures 2.1(a) and 2.1(b) show two examples of bipartite networks. In Figure 2.1(a) each transmitter node has only one outgoing link and each receiving node has only one incoming link, i.e., $|\mathcal{O}(n)| = 1$ for all $n \in \mathcal{T}$ and $|\mathcal{I}(n)| = 1$ for all $n \in \mathcal{R}$. Borrowing terminology from graph theory, we say this network has *degree* one.[1] In contrast, the network shown in Figure 2.1(b) has degree three, since all nodes $n \in \{3,7,9\}$ have degree 3. A network for which $\mathcal{T} \cap \mathcal{R} \neq \emptyset$ is referred to as a *nonbipartite* network. Examples of nonbipartite networks are shown in Figures 2.1(c) and 2.1(d). Note that all bipartite networks are necessarily singlehop networks whilst the nonbipartite networks can be either singlehop [e.g., Figure 2.1(c)] or multihop [e.g., Figure 2.1(d)] networks. Furthermore, all networks with degree one are necessarily bipartite and all nonbipartite networks have degrees larger than one.

In general, depending on the complexity limitations and the transceiver techniques employed at different nodes of the network, some

[1] In graph theory, the degree of a vertex is the number of edges incident on it and the degree of a graph is the maximum degree of any vertex. By associating the network's nodes with vertices and the network's links with (oriented) edges, we say that the degree of node n is given by $\deg(n) = |\mathcal{I}(n)| + |\mathcal{O}(n)|$ and the degree of the network is given by $\max_{n \in \mathcal{N}} \deg(n)$.

nodes may have restricted transmit and receive capabilities. For example, certain nodes may have only singlepacket receive and/or transmit capabilities[2] and some nodes may not be able to transmit and receive simultaneously. These limitations create subsets of mutually exclusive links and induce a combinatorial nature for the power and rate optimization in the case of networks with degree larger than one [9, 25, 26, 62, 68, 112, 138]. An example is the maximum weighted link scheduling for multihop wireless networks [120].

We assume that all links share a common channel and the interference is controlled via power allocation. We denote the channel gain from the transmitter of link i to the receiver of link j by h_{ij}. For any pair of distinct links $i \neq j$, we denote the interference coefficient from link i to link j by g_{ij}. In the case of nonadjacent links (i.e., links i and j do not have a common node), g_{ij} represents the power of the interference signal at the receiver node of link j when one unit of power is allocated to the transmitter node of link i, i.e., $g_{ij} = |h_{ij}|^2$. When links i and j are adjacent (i.e., links i and j do have a common node), the value of g_{ij} also depends on the transmit and receive capabilities of the common node. Specifically, we set $g_{ij} = \infty$ if links i and j are mutually exclusive and $g_{ij} = |h_{ij}|^2$ if links i and j can be simultaneously activated. Thus, $g_{ij} = g_{ji} = \infty$ for any pair of mutually exclusive links. Figure 2.2 illustrates three examples of choosing the value of the interference coefficient in the case of adjacent links. Note that in the case of nonbipartite networks, when $i \in \mathcal{O}(n)$ and $j \in \mathcal{I}(n)$, the term g_{ij} represents the power gain within the same node from its transmitter to its receiver, and is referred to as the self-interference coefficient [see Figure 2.2(c)]. In the case of wireless networks, these gains can be several orders of magnitude larger than the power gains between distinct nodes. References [47, 100, 102, 119] discuss various self interference cancelations techniques that provide different degrees of accuracy. When such schemes are employed, g_{ij} models the residual self-interference coefficient after a certain (imperfect) self interference cancelation technique was performed.

[2] We say that a node has singlepacket receive capability if it can only receive from a single incoming link at a time. Similarly, we say that a node has singlepacket transmit capability if it can transmit only through a single outgoing link at a time.

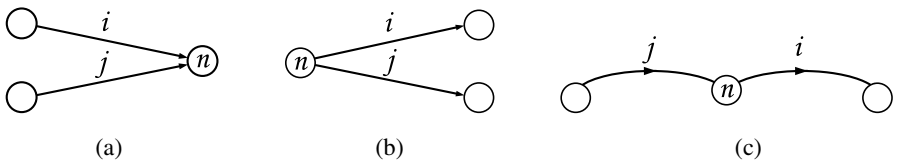


Fig. 2.2 Choosing the value of interference coefficient in the case of adjacent links: (a) $i,j \in \mathcal{I}(n)$, $g_{ij} = g_{ji} = \infty$ if node n has singlepacket receive capability or $g_{ij} = |h_{ii}|^2$, $g_{ji} = |h_{jj}|^2$ if node n has multipacket receive capability; (b) $i,j \in \mathcal{O}(n)$, $g_{ij} = g_{ji} = \infty$ if node n has singlepacket transmit capability or $g_{ij} = |h_{jj}|^2$, $g_{ji} = |h_{ii}|^2$ if node n has multipacket transmit capability; (c) $i \in \mathcal{O}(n)$, $j \in \mathcal{I}(n)$, $g_{ij} = g_{ji} = \infty$ if node n can not transmit and receive simultaneously or $g_{ij} = |h_{ij}|^2$ and $g_{ji} = |h_{ji}|^2$ if node n can transmit and receive simultaneously.

It is worth noting that the interference model described previously can easily be extended to accommodate different multiple access techniques by appropriately reinterpreting the interference coefficients. For example, in the case of wireless CDMA networks, the interference coefficient g_{ij} would model the residual interference at the output of the despreading filter of node rec(j) [124]. Similarly, in the case of wireless SDMA networks, where nodes are equipped with multiple antennas, g_{ij} represents the equivalent interference coefficient measured at the output of the antenna combiner of node rec(j) [124]. Extensions to a multichannel scenario (e.g., FDMA or FDMA-SDMA networks) are also possible by introducing multiple links between nodes, one link for each available spectral channel, and by setting $g_{ij} = 0$ if links i and j correspond to orthogonal channels. However, many such extensions are beyond the main scope of this volume.

We consider the case where all receiver nodes are using *singleuser detection* (i.e., a receiver decodes each of its intended signals by treating all other interfering signals as noise) and assume that the achievable rate of link l is given by

$$r_l = \log\left(1 + \frac{g_{ll}p_l}{\sigma^2 + \sum_{j \neq l} g_{jl}p_j}\right), \tag{2.1}$$

where p_l is the power allocated to link l, σ^2 represents the power of the thermal noise at the receiver, and g_{ll} represents the power gain

of link l, i.e., $g_{ll} = |h_{ll}|^2$. The use of the Shannon formula[3] for the achievable rate in (2.1) is common practice (see, e.g., [115, 124]) but it must be noted that this is not strictly correct in the case of finite length packets. However, as the packet length increases, it is asymptotically correct.

Let us first consider the case of singlecast networks, where all links carry different information. Let β_l denote an arbitrary nonnegative number which represents the weight associated with link l. Assuming that the power allocation is subject to a maximum power constraint $\sum_{l\in\mathcal{O}(n)} p_l \leq p_n^{\max}$ for each transmitter node $n \in \mathcal{T}$,[4] the problem of weighted sum-rate maximization can be expressed as

$$\begin{array}{ll} \text{maximize} & \sum_{l\in\mathcal{L}} \beta_l \log\left(1 + \dfrac{g_{ll}p_l}{\sigma^2 + \sum_{j\neq l} g_{jl}p_j}\right) \\ \text{subject to} & \sum_{l\in\mathcal{O}(n)} p_l \leq p_n^{\max},\ n \in \mathcal{T} \\ & p_l \geq 0,\ l \in \mathcal{L}, \end{array} \tag{2.2}$$

where the variable is $(p_l)_{l\in\mathcal{L}}$.

In the case of multicast networks, a transmitter can simultaneously send common information to multiple receiver nodes. We consider the general case where each transmitter node can have several multicast transmissions. Thus, for each $n \in \mathcal{T}$ we partition $\mathcal{O}(n)$ into M_n disjoint subsets of links, i.e., $\mathcal{O}(n) = \cup_{m=1}^{M_n} \mathcal{O}^m(n)$, where M_n is the number of multicast transmissions from node n and the set $\mathcal{O}^m(n)$ contains all links associated with the mth multicast transmission of node n (see Figure 2.3). Let p_n^m and β_n^m be the power and the nonnegative weight allocated to the mth multicast transmission of node n. Moreover, let $\mathbf{p} = (p_n^m)_{n\in\mathcal{T},\ m=1,\ldots,M_n}$ and denote the SINR of the lth link belongs to

[3] The algorithm presented in this section can be used for any other rate versus SINR dependence. The only restriction is that the rate must be a nondecreasing and Lipschitz continuous function of SINR.

[4] For the sake of clarity we only consider the case of sum-power constraints for each transmitter node. However, supplementary sum-power constraints can be also handled by the given algorithm. For example, in the case of a cellular downlink employing the cooperation of several multiantenna base stations, sum-power constraints per subsets of nodes (one subset of nodes corresponds to a base station) should be also considered [122].

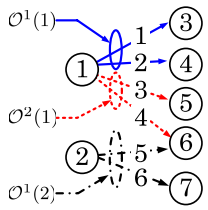


Fig. 2.3 Multicast network: Different line styles represent different multicast transmissions. $\mathcal{T}=\{1,2\}$, $M_1=2$, $M_2=1$, $\mathcal{O}^1(1)=\{1,2\}$, $\mathcal{O}^2(1)=\{3,4\}$, and $\mathcal{O}^1(2)=\{5,6\}$.

the mth multicast transmission of the nth node by $\mathrm{SINR}_n^{ml}(\mathbf{p})$, where

$$\mathrm{SINR}_n^{ml}(\mathbf{p})=\frac{g_{ll}p_n^m}{\sigma^2+\sum_{j\in\mathcal{T},j\neq n}\sum_{k=1}^{M_j}p_j^k\max\limits_{i\in\mathcal{O}^k(j)}g_{il}+\sum_{k=1,k\neq m}^{M_n}p_n^k\max\limits_{i\in\mathcal{O}^k(n)}g_{il}} \quad \text{for all } n\in\mathcal{T},\ m=1,\ldots,M_n. \tag{2.3}$$

Clearly, for any link in the mth multicast transmission of node n, i.e., $l\in\mathcal{O}^m(n)$, interference at $\mathrm{rec}(l)$ is created by the other multicast transmissions of node n itself and by multicast transmissions of other nodes. The $\max(\cdot)$ operator in the denominator of SINR expressions is used to impose mutually exclusive multicast transmissions, for example, if node 6 in Figure 2.3 has singlepacket reception capability, then $\mathcal{O}^2(1)$ and $\mathcal{O}^1(2)$ are mutually exclusive.

Thus, by noting that the maximum rate achievable by all links in $\mathcal{O}^m(n)$ is given by $r_n^m=\min_{l\in\mathcal{O}^m(n)}r_l$, the weighted sum-rate maximization problem can be expressed as

$$\begin{aligned}\text{maximize}\quad & \sum_{n\in\mathcal{T}}\sum_{m=1}^{M_n}\beta_n^m\min_{l\in\mathcal{O}^m(n)}\log\left(1+\mathrm{SINR}_n^{ml}(\mathbf{p})\right)\\ \text{subject to}\quad & \sum_{m=1}^{M_n}p_n^m\le p_n^{\max},\ n\in\mathcal{T}\\ & p_n^m\ge 0,\ n\in\mathcal{T},\ m=1,\ldots,M_n,\end{aligned} \tag{2.4}$$

where the variable is $(p_n^m)_{n\in\mathcal{T},\ m=1,\ldots,M_n}$.

2.2 Algorithm Derivation

For the sake of clarity, let us first address the case of singlecast networks. Extension to multicast case is presented separately in Section 2.4. We start by equivalently reformulating the original

problem (2.2) as minimization of a nonconvex function over an L-dimensional rectangle. Then, we present the algorithm based on a branch and bound technique [16] to minimize the nonconvex function over the L-dimensional rectangle.

By introducing auxiliary variables $\gamma_l,\ l \in \mathcal{L}$ we first reformulate problem (2.2) in the following equivalent form:

$$\begin{array}{ll} \text{minimize} & \sum_{l\in\mathcal{L}} -\beta_l \log(1+\gamma_l) \\ \text{subject to} & \gamma_l \leq \dfrac{g_{ll}p_l}{\sigma^2 + \sum_{j\neq l} g_{jl}p_j},\ l \in \mathcal{L} \\ & \sum_{l\in\mathcal{O}(n)} p_l \leq p_n^{\max},\ n \in \mathcal{T} \\ & p_l \geq 0,\ l \in \mathcal{L}, \end{array} \tag{2.5}$$

where the variables are $(p_l)_{l\in\mathcal{L}}$ and $(\gamma_l)_{l\in\mathcal{L}}$. The equivalence between problems (2.2) and (2.5) follows from the monotone increasing property of the $\log(\cdot)$ function. Clearly, any feasible $\gamma_l,\ l \in \mathcal{L}$ in problem (2.5) represents an achievable SINR value for link l. Let us denote the objective function of problem (2.5) by $f_0(\boldsymbol{\gamma}) = \sum_{l\in\mathcal{L}} -\beta_l \log(1+\gamma_l)$ and the feasible set for the variables $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_L)$ (or the achievable SINR values) by $\mathcal{G}$, i.e.,

$$\mathcal{G} = \left\{ \boldsymbol{\gamma} \left| \begin{array}{ll} \gamma_l \leq \dfrac{g_{ll}p_l}{\sigma^2 + \sum_{j\neq l} g_{jl}p_j}, & l \in \mathcal{L} \\ \sum_{l\in\mathcal{O}(n)} p_l \leq p_n^{\max}, & n \in \mathcal{T} \\ p_l \geq 0, & l \in \mathcal{L} \end{array} \right. \right\}. \tag{2.6}$$

The optimal value of problem (2.5) can be expressed compactly as $t^\star = \inf_{\boldsymbol{\gamma}\in\mathcal{G}} f_0(\boldsymbol{\gamma})$.

For clarity, let us define a new function $\tilde{f} : \mathbb{R}_+^L \to \mathbb{R}$ as

$$\tilde{f}(\boldsymbol{\gamma}) = \begin{cases} f_0(\boldsymbol{\gamma}) & \boldsymbol{\gamma} \in \mathcal{G} \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

and note that for any $\mathcal{D} \subseteq \mathbb{R}_+^L$ such that $\mathcal{G} \subseteq \mathcal{D}$, we have

$$\inf_{\boldsymbol{\gamma}\in\mathcal{D}} \tilde{f}(\boldsymbol{\gamma}) = \inf_{\boldsymbol{\gamma}\in\mathcal{G}} f_0(\boldsymbol{\gamma}) = t^\star, \tag{2.8}$$

where the first equality follows from that for any $\boldsymbol{\gamma} \in \mathbb{R}_+^L$ we have $f_0(\boldsymbol{\gamma}) \leq 0$. It is also worth noting that the function $\tilde{f}$ is nonconvex over $\mathcal{D}$ and f_0 is a global lower bound on $\tilde{f}$, i.e., $f_0(\boldsymbol{\gamma}) \leq \tilde{f}(\boldsymbol{\gamma})$ for all $\boldsymbol{\gamma} \in \mathcal{D}$.

Let us now define the L-dimensional rectangle

$$\mathcal{Q}_{\text{init}} = \left\{\boldsymbol{\gamma} \middle| 0 \leq \gamma_l \leq \sigma^{-2} g_{ll} p_{\text{tran}(l)}^{\max},\ l \in \mathcal{L}\right\},$$

which encloses the set of all achievable SINR values, i.e., $\mathcal{G} \subseteq \mathcal{Q}_{\text{init}}$. By using (2.8), it follows that

$$t^\star = \inf_{\boldsymbol{\gamma} \in \mathcal{Q}_{\text{init}}} \tilde{f}(\boldsymbol{\gamma}).$$

Thus, we have reformulated problem (2.2) equivalently as a minimization of the nonconvex function $\tilde{f}$ over the rectangle $\mathcal{Q}_{\text{init}}$. In what follows we show how the branch and bound technique is used to minimize $\tilde{f}$ over $\mathcal{Q}_{\text{init}}$.

Let $\mathcal{Q}$ be a L-dimensional rectangle defined as

$$\mathcal{Q} = \left\{\boldsymbol{\gamma} \middle| \gamma_{l,\min} \leq \gamma_l \leq \gamma_{l,\max},\ l \in \mathcal{L}\right\},$$

where $\gamma_{l,\min}$ and $\gamma_{l,\max}$ are real numbers such that $\gamma_{l,\min} \leq \gamma_{l,\max}$ for all $l \in \mathcal{L}$. For any L-dimensional rectangle $\mathcal{Q} \subseteq \mathcal{Q}_{\text{init}}$, let us now define the following function:

$$\phi_{\min}(\mathcal{Q}) = \inf_{\boldsymbol{\gamma} \in \mathcal{Q}} \tilde{f}(\boldsymbol{\gamma}). \tag{2.9}$$

Note that

$$\phi_{\min}(\mathcal{Q}_{\text{init}}) = \inf_{\boldsymbol{\gamma} \in \mathcal{Q}_{\text{init}}} \tilde{f}(\boldsymbol{\gamma}) = t^\star. \tag{2.10}$$

The key idea of the branch and bound method is to generate a sequence of asymptotically tight upper and lower bounds for $\phi_{\min}(\mathcal{Q}_{\text{init}})$. At each iteration k, the lower bound L_k and the upper bound U_k are updated by partitioning $\mathcal{Q}_{\text{init}}$ into smaller rectangles. To ensure convergence, the bounds should become tight as the number of rectangles in the partition of $\mathcal{Q}_{\text{init}}$ grows. To do this, the branch and bound method uses two functions $\phi_{\text{ub}}(\mathcal{Q})$ and $\phi_{\text{lb}}(\mathcal{Q})$, defined for any rectangle $\mathcal{Q} \subseteq \mathcal{Q}_{\text{init}}$ such that the following conditions are satisfied [16].

C1 : The functions $\phi_{\text{lb}}(\mathcal{Q})$ and $\phi_{\text{ub}}(\mathcal{Q})$ compute a lower bound and an upper bound respectively on $\phi_{\min}(\mathcal{Q})$, i.e.,

$$\forall \mathcal{Q} \subseteq \mathcal{Q}_{\text{init}} \text{ we have } \phi_{\text{lb}}(\mathcal{Q}) \leq \phi_{\min}(\mathcal{Q}) \leq \phi_{\text{ub}}(\mathcal{Q}). \tag{2.11}$$

C2 : As the maximum half length of the sides of $\mathcal{Q}$ $\big($i.e., $\text{size}(\mathcal{Q})=\frac{1}{2}\max\limits_{l\in\mathcal{L}}\{\gamma_{l,\max}-\gamma_{l,\min}\}\big)$ goes to zero, the difference between the upper and lower bounds uniformly converges to zero, i.e.,

$$\forall\epsilon>0\ \exists\delta>0\ \text{ s.t. }\ \forall\mathcal{Q}\subseteq\mathcal{Q}_{\text{init}},\text{size}(\mathcal{Q})\leq\delta \\ \Rightarrow\phi_{\text{ub}}(\mathcal{Q})-\phi_{\text{lb}}(\mathcal{Q})\leq\epsilon. \quad (2.12)$$

For the sake of clarity, the definition and computation of ϕ_{lb} and ϕ_{ub} are described in Section 2.3. In the remainder of this section we will present the branch and bound method in more detail.

Let ϵ be an *a priori* specified tolerance. The algorithm starts by computing $\phi_{\text{ub}}(\mathcal{Q}_{\text{init}})$ and $\phi_{\text{lb}}(\mathcal{Q}_{\text{init}})$. If $\phi_{\text{ub}}(\mathcal{Q}_{\text{init}})-\phi_{\text{lb}}(\mathcal{Q}_{\text{init}})\leq\epsilon$, the algorithm terminates and **C1** in (2.11) confirms that we have an upper bound $\phi_{\text{ub}}(\mathcal{Q}_{\text{init}})$, which is at most ϵ-away from the optimal value $t^\star$. Otherwise, we start partitioning $\mathcal{Q}_{\text{init}}$ into smaller rectangles. At the kth partitioning step, $\mathcal{Q}_{\text{init}}$ is split into k rectangles such that $\mathcal{Q}_{\text{init}}=\mathcal{Q}_1\cup\mathcal{Q}_2\cup\ldots\cup\mathcal{Q}_k$ and $\phi_{\text{ub}}(\mathcal{Q}_k)$ and $\phi_{\text{lb}}(\mathcal{Q}_k)$ are computed. Then the lower bound L_k and upper bound U_k are updated as follows:

$$L_k=\min_{i\in\{1,2,\ldots,k\}}\phi_{\text{lb}}(\mathcal{Q}_i)\leq\phi_{\min}(\mathcal{Q}_{\text{init}})=t^\star\leq\min_{i\in\{1,2,\ldots,k\}}\phi_{\text{ub}}(\mathcal{Q}_i)=U_k. \quad (2.13)$$

Note that the lower bound L_k and the upper bound U_k are refined at each step and they represent the best lower and upper bounds obtained so far. If the difference between new bounds become smaller than ϵ, then the algorithm terminates. Otherwise, further partitioning of $\mathcal{Q}_{\text{init}}$ is required until the difference between U_k and L_k is less than ϵ. The condition **C2** in (2.12) ensures that, the difference U_k-L_k eventually becomes smaller than ϵ for some finite k. The algorithm based on the branch and bound method can be summarized as follows:

The first step initializes the algorithm and the upper and lower bounds are computed over the initial rectangle $\mathcal{Q}_{\text{init}}$. The second step checks the difference between the best upper and lower bounds found so far [bounds U_k and L_k are given by (2.13)]. The algorithm repeats steps 3 to 6 until $U_k-L_k<\epsilon$.

Algorithm 2.1 Branch and bound method for WSRMax.

1. Initialization; given tolerance $\epsilon > 0$. Set $k = 1$, $\mathcal{B}_1 = \{\mathcal{Q}_{\text{init}}\}$, $U_1 = \phi_{\text{ub}}(\mathcal{Q}_{\text{init}})$, and $L_1 = \phi_{\text{lb}}(\mathcal{Q}_{\text{init}})$.
2. Stopping criterion; if $U_k - L_k > \epsilon$ go to step 3, otherwise STOP.
3. Branching;
   (a) pick $\mathcal{Q} \in \mathcal{B}_k$ for which $\phi_{\text{lb}}(\mathcal{Q}) = L_k$ and set $\mathcal{Q}_k = \mathcal{Q}$.
   (b) split $\mathcal{Q}_k$ along one of its longest edge into $\mathcal{Q}_I$ and $\mathcal{Q}_{II}$.
   (c) form $\mathcal{B}_{k+1}$ from $\mathcal{B}_k$ by removing $\mathcal{Q}_k$ and adding $\mathcal{Q}_I$ and $\mathcal{Q}_{II}$.
4. Bounding;
   (a) set $U_{k+1} = \min_{\mathcal{Q} \in \mathcal{B}_{k+1}} \{\phi_{\text{ub}}(\mathcal{Q})\}$.
   (b) set $L_{k+1} = \min_{\mathcal{Q} \in \mathcal{B}_{k+1}} \{\phi_{\text{lb}}(\mathcal{Q})\}$.
5. Pruning;
   (a) pick all $\mathcal{Q} \in \mathcal{B}_{k+1}$ for which $\phi_{\text{lb}}(\mathcal{Q}) \geq U_{k+1}$.
   (b) update $\mathcal{B}_{k+1}$ by removing all $\mathcal{Q}$ obtained in the above step 5-(a).
6. Set $k = k + 1$ and go to step 2.

Step 3 is the *branching* mechanism of the algorithm. Here we adopt the following branching rule: select from the current partition of $\mathcal{Q}_{\text{init}}$ (i.e., $\mathcal{B}_k$) the rectangle with the smallest lower bound and split it in two smaller rectangles along its longest edge. Splitting the chosen rectangle along its longest edge ensures the convergence of the algorithm [16]. At step 4 the best upper bound U_k and the best lower bound L_k are updated according to (2.13).

Step 5 is used to eliminate (or prune) rectangles for which the lower bound is larger than the best upper bound found so far, since those rectangles can never contain a minimizer of the function $\tilde{f}$. Note that *pruning* does not affect the speed of the main algorithm since none of the rectangles that were pruned will be selected later in the branching

step 3 for further splitting. The advantage of pruning is the release of the memory otherwise used for storing unnecessary rectangles.

The convergence of the above algorithm is established by the following theorem.

Theorem 2.1. For any $\mathcal{Q} \subseteq \mathcal{Q}_{\text{init}}$ with $\mathcal{Q} = \big\{\boldsymbol{\gamma} \big| \gamma_{l,\min} \leq \gamma_l \leq \gamma_{l,\max}, \; l \in \mathcal{L}\big\}$, if the functions $\phi_{\text{ub}}(\mathcal{Q})$ and $\phi_{\text{lb}}(\mathcal{Q})$ satisfy the conditions **C1** and **C2**, then Algorithm 2.1 converges in a finite number of iterations to a value arbitrarily close to $t^\star$, i.e., $\forall \epsilon > 0, \; \exists K > 0$ s.t. $U_K - t^\star \leq \epsilon$.

Proof. The proof is given in [16, 8] and is not reproduced here for the sake of brevity. □

2.3 Computation of Upper and Lower Bounds

Note that the main challenge in designing a global optimization algorithm based on the branch and bound method is to find cheaply computable functions $\phi_{\text{ub}}(\mathcal{Q})$ and $\phi_{\text{lb}}(\mathcal{Q})$ such that the conditions given in (2.11) and (2.12) are satisfied. The essence of the branch and bound method is based on that for any $\mathcal{Q} \subseteq \mathcal{Q}_{\text{init}}$, the bounds $\phi_{\text{ub}}(\mathcal{Q})$ and $\phi_{\text{lb}}(\mathcal{Q})$ are substantially easier to compute than the true minimum $\phi_{\min}(\mathcal{Q})$ [16].

In this section we present several candidates for $\phi_{\text{lb}}(\mathcal{Q})$ and $\phi_{\text{ub}}(\mathcal{Q})$ in Algorithm 2.1. To simplify the presentation, first we describe two basic lower and upper bound functions, prove that they satisfy the conditions **C1** and **C2** [see (2.11) and (2.12)] and present efficient methods for computing them. Computationally efficient better bounds are presented later in this section.

2.3.1 Basic Lower and Upper Bounds

Recall that $\mathcal{Q} = \{\boldsymbol{\gamma} \,|\gamma_{l,\min} \leq \gamma_l \leq \gamma_{l,\max}, \; l \in \mathcal{L}\}$. We now define the functions $\phi_{\text{lb}}^{\text{Basic}}(\mathcal{Q})$ and $\phi_{\text{ub}}^{\text{Basic}}(\mathcal{Q})$ as

$$\phi_{\text{lb}}^{\text{Basic}}(\mathcal{Q}) = \begin{cases} f_0(\boldsymbol{\gamma}_{\max}) & \boldsymbol{\gamma}_{\min} \in \mathcal{G} \\ 0 & \text{otherwise} \; ; \end{cases} \tag{2.14}$$

$$\phi_{\text{ub}}^{\text{Basic}}(\mathcal{Q}) = \tilde{f}(\boldsymbol{\gamma}_{\min}) = \begin{cases} f_0(\boldsymbol{\gamma}_{\min}) & \boldsymbol{\gamma}_{\min} \in \mathcal{G} \\ 0 & \text{otherwise,} \end{cases} \tag{2.15}$$

where $\boldsymbol{\gamma}_{\max} = (\gamma_{1,\max}, \ldots, \gamma_{L,\max})$, $\boldsymbol{\gamma}_{\min} = (\gamma_{1,\min}, \ldots, \gamma_{L,\min})$, and $\mathcal{G}$ is defined in (2.6). Note that the most computationally expensive part of evaluating $\phi_{\text{lb}}^{\text{Basic}}(\mathcal{Q})$ and $\phi_{\text{ub}}^{\text{Basic}}(\mathcal{Q})$ is to check the condition $\boldsymbol{\gamma}_{\min} \in \mathcal{G}$. An efficient method for checking this condition is provided soon after establishing the following important properties of $\phi_{\text{lb}}^{\text{Basic}}$ and $\phi_{\text{ub}}^{\text{Basic}}$.

Lemma 2.2. The functions $\phi_{\text{lb}}^{\text{Basic}}(\mathcal{Q})$ and $\phi_{\text{ub}}^{\text{Basic}}(\mathcal{Q})$ satisfy the condition **C1**.

Proof. If $\boldsymbol{\gamma}_{\min} \notin \mathcal{G}$, then $\phi_{\text{lb}}^{\text{Basic}}(\mathcal{Q}) = \phi_{\min}(\mathcal{Q}) = \phi_{\text{ub}}^{\text{Basic}}(\mathcal{Q}) = 0$, and therefore the inequalities in **C1** hold with equalities. If $\boldsymbol{\gamma}_{\min} \in \mathcal{G}$, then we have

$$\phi_{\min}(\mathcal{Q}) = \inf_{\boldsymbol{\gamma} \in \mathcal{Q}} \tilde{f}(\boldsymbol{\gamma}) \leq \tilde{f}(\boldsymbol{\gamma}_{\min}) = f_0(\boldsymbol{\gamma}_{\min}) = \phi_{\text{ub}}^{\text{Basic}}(\mathcal{Q}). \tag{2.16}$$

The first equality follows from (2.9), the inequality follows since $\boldsymbol{\gamma}_{\min} \in \mathcal{Q}$, and the second equality follows from (2.7). Moreover, we have

$$\phi_{\min}(\mathcal{Q}) = \inf_{\boldsymbol{\gamma} \in \mathcal{Q}} \tilde{f}(\boldsymbol{\gamma}) \geq \inf_{\boldsymbol{\gamma} \in \mathcal{Q}} f_0(\boldsymbol{\gamma}) = f_0(\boldsymbol{\gamma}_{\max}) = \phi_{\text{lb}}^{\text{Basic}}(\mathcal{Q}), \tag{2.17}$$

where the inequality follows from that $\tilde{f}(\boldsymbol{\gamma}) \geq f_0(\boldsymbol{\gamma})$ and the second equality is from that $\mathcal{Q}$ is a rectangle and $f_0(\boldsymbol{\gamma})$ is monotonically decreasing in each variable γ_l, $l \in \mathcal{L}$. From (2.16) and (2.17) we conclude that $\phi_{\text{lb}}^{\text{Basic}}(\mathcal{Q}) \leq \phi_{\min}(\mathcal{Q}) \leq \phi_{\text{ub}}^{\text{Basic}}(\mathcal{Q})$. □

Lemma 2.3. The functions $\phi_{\text{lb}}^{\text{Basic}}(\mathcal{Q})$ and $\phi_{\text{ub}}^{\text{Basic}}(\mathcal{Q})$ satisfy the condition **C2**.

Proof. We first show that the function $f_0(\boldsymbol{\gamma}) = \sum_{l \in \mathcal{L}} -\beta_l \log(1 + \gamma_l)$ is Lipschitz continuous on $\mathbb{R}_+^L$ with the constant $D = \sqrt{\sum_{l \in \mathcal{L}} \beta_l^2}$, i.e.,

$$|f_0(\boldsymbol{\mu}) - f_0(\boldsymbol{\nu})| \leq D \, ||\boldsymbol{\mu} - \boldsymbol{\nu}||_2 \tag{2.18}$$

for all $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}_+^L$. We start by noting that $f_0(\boldsymbol{\gamma})$ is convex. Therefore, for all $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}_+^L$ we have [22, sec. 3.1.3]

$$f_0(\boldsymbol{\mu}) - f_0(\boldsymbol{\nu}) \leq \nabla f_0(\boldsymbol{\mu})^{\mathrm{T}}(\boldsymbol{\mu} - \boldsymbol{\nu}). \tag{2.19}$$

Without loss of generality, we can assume that $f_0(\boldsymbol{\mu}) - f_0(\boldsymbol{\nu}) \geq 0$. Otherwise, we can obtain exactly the same results by interchanging $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ in (2.19), i.e., $f_0(\boldsymbol{\nu}) - f_0(\boldsymbol{\mu}) \leq \nabla f_0(\boldsymbol{\nu})^{\mathrm{T}}(\boldsymbol{\nu} - \boldsymbol{\mu})$. Thus, we have

$$\begin{aligned}
|f_0(\boldsymbol{\mu}) - f_0(\boldsymbol{\nu})| &\leq |\nabla f_0(\boldsymbol{\mu})^{\mathrm{T}}(\boldsymbol{\mu} - \boldsymbol{\nu})| && (2.20)\\
&\leq ||\nabla f_0(\boldsymbol{\mu})||_2 ||(\boldsymbol{\mu} - \boldsymbol{\nu})||_2 && (2.21)\\
&\leq \max\nolimits_{\boldsymbol{\gamma} \in \mathbb{R}_+^L} ||\nabla f_0(\boldsymbol{\gamma})||_2 ||(\boldsymbol{\mu} - \boldsymbol{\nu})||_2 && (2.22)\\
&= \max_{\boldsymbol{\gamma} \in \mathbb{R}_+^L} \sqrt{\sum_{l \in \mathcal{L}} \frac{\beta_l^2}{(1+\gamma_l)^2}}\ ||(\boldsymbol{\mu} - \boldsymbol{\nu})||_2 && (2.23)\\
&= D||(\boldsymbol{\mu} - \boldsymbol{\nu})||_2, && (2.24)
\end{aligned}$$

where (2.20) follows from (2.19), (2.21) follows from the Cauchy–Schwarz inequality, (2.22) follows from the maximization operation, (2.23) follows by noting that $[\nabla f_0(\gamma)]_l = \beta_l/(1+\gamma_l)$, $l \in \mathcal{L}$, and (2.24) follows by setting $\gamma_l = 0$ for all $l \in \mathcal{L}$.

Now we can write the following relations:

$$\begin{aligned}
\phi_{\mathrm{ub}}^{\mathrm{Basic}}(\mathcal{Q}) - \phi_{\mathrm{lb}}^{\mathrm{Basic}}(\mathcal{Q}) &\leq f_0(\boldsymbol{\gamma}_{\min}) - f_0(\boldsymbol{\gamma}_{\max}) && (2.25)\\
&\leq D\,||\boldsymbol{\gamma}_{\min} - \boldsymbol{\gamma}_{\max}||_2 && (2.26)\\
&= D\left\|\sum_{l \in \mathcal{L}} (\gamma_{l,\max} - \gamma_{l,\min})\mathbf{e}_l\right\|_2 && (2.27)\\
&\leq D\textstyle\sum_{l \in \mathcal{L}} (\gamma_{l,\max} - \gamma_{l,\min}) && (2.28)\\
&\leq 2DL\ \mathrm{size}(\mathcal{Q}). && (2.29)
\end{aligned}$$

The first inequality (2.25) follows from (2.14) and (2.15) by noting that f_0 is nonincreasing, (2.26) follows from (2.18), (2.27) follows clearly by noting that $\mathbf{e}_l$ is the lth standard unit vector, (2.28) follows from triangle inequality, and (2.29) follows from the definition of size($\mathcal{Q}$) (see **C2**). Thus, for any given $\epsilon > 0$, we can select δ such that $\delta \leq \epsilon/2DL$, which in turns implies that condition **C2** is satisfied. □

In the sequel, we present a computationally efficient method of checking the condition $\boldsymbol{\gamma}_{\min} \in \mathcal{G}$, which is central in computing $\phi_{\text{lb}}^{\text{Basic}}(\mathcal{Q})$ and $\phi_{\text{ub}}^{\text{Basic}}(\mathcal{Q})$ efficiently. Without loss of generality, we can assume that $\boldsymbol{\gamma}_{\min} > \mathbf{0}$. Note that the method can be extended to the case where there are links l for which $\gamma_{l,\min} = 0$ in a straightforward manner; then, checking the original condition $\boldsymbol{\gamma}_{\min} \in \mathcal{G}$ is equivalent to checking a modified condition $\breve{\boldsymbol{\gamma}}_{\min} \in \breve{\mathcal{G}}$, where $\breve{\boldsymbol{\gamma}}_{\min}$ and $\breve{\mathcal{G}}$ are obtained by eliminating the dimensions (or link indexes) for which $\gamma_{l,\min} = 0$ and thus, we have $\breve{\boldsymbol{\gamma}}_{\min} > \mathbf{0}$.

Let us first consider the first set of inequalities in the description of $\mathcal{G}$, i.e.,

$$\gamma_l \leq \frac{g_{ll} p_l}{\sigma^2 + \sum_{j \neq l} g_{jl} p_j}, \; l \in \mathcal{L}. \tag{2.30}$$

Let $\mathbf{p} = (p_1, \ldots, p_L)$. By rearranging the terms and by using $\geq$ to denote componentwise inequalities, (2.30) is equivalent to [114, 35]

$$(\mathbf{I} - \mathbf{B}(\boldsymbol{\gamma})\mathbf{G})\,\mathbf{p} \geq \sigma^2 \mathbf{B}(\boldsymbol{\gamma})\mathbf{1}, \tag{2.31}$$

where the matrices $\mathbf{B}(\boldsymbol{\gamma}) \in \mathbb{R}_+^{L \times L}$ and $\mathbf{G} \in \mathbb{R}_+^{L \times L}$ are defined by

$$\mathbf{B}(\boldsymbol{\gamma}) = \text{diag}\left(\frac{\gamma_1}{g_{11}}, \ldots, \frac{\gamma_L}{g_{LL}}\right) \; ; \; [\mathbf{G}]_{i,j} = \begin{cases} g_{ji} & i \neq j \\ 0 & \text{otherwise} \end{cases}. \tag{2.32}$$

Here $\text{diag}(\mathbf{x})$ denotes the diagonal matrix with the elements of vector $\mathbf{x}$ on the main diagonal. For the notational simplicity, let

$$\mathbf{A}(\boldsymbol{\gamma}) = \mathbf{I} - \mathbf{B}(\boldsymbol{\gamma})\mathbf{G} \;\text{ and }\; \mathbf{b}(\boldsymbol{\gamma}) = \sigma^2 \mathbf{B}(\boldsymbol{\gamma})\mathbf{1}. \tag{2.33}$$

Thus, (2.30) can be compactly expressed as $\mathbf{A}(\boldsymbol{\gamma})\mathbf{p} \geq \mathbf{b}(\boldsymbol{\gamma})$. Let us denote the spectral radius [51, p. 5] of matrix $\mathbf{B}(\boldsymbol{\gamma})\mathbf{G}$ by $\rho(\mathbf{B}(\boldsymbol{\gamma})\mathbf{G})$. The following theorem helps us to check if $\boldsymbol{\gamma} \in \mathcal{G}$.

Theorem 2.4. For any $\boldsymbol{\gamma} > \mathbf{0}$, the following implications hold:

1. $\rho(\mathbf{B}(\boldsymbol{\gamma})\mathbf{G}) \geq 1 \;\Rightarrow\; \boldsymbol{\gamma} \notin \mathcal{G}$.
2. $\rho(\mathbf{B}(\boldsymbol{\gamma})\mathbf{G}) < 1$ and $\sum_{l \in \mathcal{O}(n)} p_l \leq p_n^{\max}$ for all $n \in \mathcal{T}$, where $\mathbf{p} = \mathbf{A}^{-1}(\boldsymbol{\gamma})\mathbf{b}(\boldsymbol{\gamma}) \Rightarrow \boldsymbol{\gamma} \in \mathcal{G}$.
3. $\rho(\mathbf{B}(\boldsymbol{\gamma})\mathbf{G}) < 1$ and $\exists n \in \mathcal{T}$ s.t. $\sum_{l \in \mathcal{O}(n)} p_l > p_n^{\max}$, where $\mathbf{p} = \mathbf{A}^{-1}(\boldsymbol{\gamma})\mathbf{b}(\boldsymbol{\gamma}) \;\Rightarrow\; \boldsymbol{\gamma} \notin \mathcal{G}$.

Proof. The proof is similar to the one provided in [135, app. A] and it is not reproduced here for the sake of brevity. □

Based on Theorem 2.4, the condition $\boldsymbol{\gamma}_{\min} \in \mathcal{G}$ can be checked as follows:

Algorithm 2.2 Checking for condition $\boldsymbol{\gamma}_{\min} \in \mathcal{G}$

1. Construct $\mathbf{B}(\boldsymbol{\gamma}_{\min})$ and $\mathbf{G}$ according to (2.32).
2. If $\rho(\mathbf{B}(\boldsymbol{\gamma}_{\min})\mathbf{G}) \geq 1$, then $\boldsymbol{\gamma}_{\min} \notin \mathcal{G}$ and STOP. Otherwise, let
   $\mathbf{p} = \mathbf{A}^{-1}(\boldsymbol{\gamma}_{\min})\mathbf{b}(\boldsymbol{\gamma}_{\min})$.
3. If $\sum_{l \in \mathcal{O}(n)} p_l \leq p_n^{\max}$ for all $n \in \mathcal{T}$, then $\boldsymbol{\gamma}_{\min} \in \mathcal{G}$ and STOP. Otherwise, $\boldsymbol{\gamma}_{\min} \notin \mathcal{G}$ and STOP.

2.3.2 Improved Lower and Upper Bounds

Finding tighter bounds is very important as they can substantially increase the convergence speed of Algorithm 2.1. By exploiting the monotonically nonincreasing property of f_0 [i.e., $\boldsymbol{\gamma}_1 \leq \boldsymbol{\gamma}_2 \Rightarrow f_0(\boldsymbol{\gamma}_1) \geq f_0(\boldsymbol{\gamma}_2)$], one improved lower bound and two improved upper bounds are presented in this subsection. Efficient methods of computing them are provided as well.

Note that, in the case of $\boldsymbol{\gamma}_{\min} \notin \mathcal{G}$ [i.e., $\mathcal{Q} \cap \mathcal{G} = \emptyset$, see Figure 2.4(a)], $\tilde{f}(\boldsymbol{\gamma}) = 0$ for any $\boldsymbol{\gamma} \in \mathcal{Q}$. Thus, both the basic lower bound (2.14) and the basic upper bound (2.15) are trivially zero and no further improvement is possible since they are tight. Consequently, tighter bounds can be found only in the case $\boldsymbol{\gamma}_{\min} \in \mathcal{G}$ [i.e., $\mathcal{Q} \cap \mathcal{G} \neq \emptyset$, see Figure 2.4(b)]. Thus, we consider only this case in the sequel, unless otherwise specified.

2.3.2.1 Improved Lower Bound

Roughly speaking, a tighter lower bound can be obtained as follows. We first construct the smallest rectangle $\bar{\mathcal{Q}}^{\star} \subseteq \mathcal{Q}$, which encloses the intersection $\mathcal{Q} \cap \mathcal{G}$ [see Figure 2.4(b)]. Let us denote this rectangle as

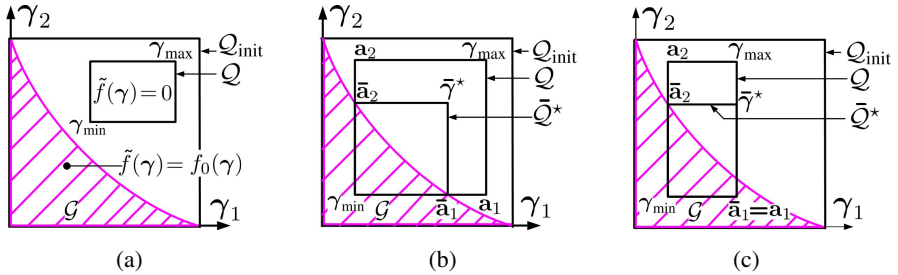


Fig. 2.4 Illustration of the sets $\mathcal{G}$, $\mathcal{Q}_{\text{init}}$, $\mathcal{Q}$, and $\bar{\mathcal{Q}}^\star$ in a 2-dimensional space.

$\bar{\mathcal{Q}}^\star = \{\boldsymbol{\gamma} \,|\gamma_{l,\min} \le \gamma_l \le \bar{\gamma}_l^\star,\ l \in \mathcal{L}\}$. The improved lower bound is given by $f_0(\bar{\gamma}_1^\star, \ldots, \bar{\gamma}_L^\star)$.[5]

Recall that $\mathcal{Q} = \{\boldsymbol{\gamma} \,|\gamma_{l,\min} \le \gamma_l \le \gamma_{l,\max},\ l \in \mathcal{L}\}$. For any $\mathcal{Q} \subseteq \mathcal{Q}_{\text{init}}$, the improved lower bound can be formally expressed as

$$\phi_{\text{lb}}^{\text{Imp}}(\mathcal{Q}) = \begin{cases} f_0(\bar{\boldsymbol{\gamma}}^\star) & \boldsymbol{\gamma}_{\min} \in \mathcal{G} \\ 0 & \text{otherwise,} \end{cases} \tag{2.34}$$

where $\bar{\boldsymbol{\gamma}}^\star = (\bar{\gamma}_1^\star, \ldots, \bar{\gamma}_L^\star)$ and $\bar{\gamma}_i^\star$ is the optimal value of the following optimization problem:

$$\begin{aligned} \text{maximize} \quad & \frac{g_{ii}p_i}{\sigma^2 + \sum_{j\neq i} g_{ji}p_j} \\ \text{subject to} \quad & \frac{g_{ii}p_i}{\sigma^2 + \sum_{j\neq i} g_{ji}p_j} \le \gamma_{i,\max} \\ & \gamma_{l,\min} = \frac{g_{ll}p_l}{\sigma^2 + \sum_{j\neq l} g_{jl}p_j},\ l \in \mathcal{L}\setminus\{i\} \\ & \textstyle\sum_{l\in\mathcal{O}(n)} p_l \le p_n^{\max},\ n \in \mathcal{T} \\ & p_l \ge 0,\ l \in \mathcal{L}, \end{aligned} \tag{2.35}$$

where the variable is $(p_l)_{l\in\mathcal{L}}$. The first inequality constraint ensures that $\bar{\mathcal{Q}}^\star \subseteq \mathcal{Q}$, and it is active if and only if the corner point

[5] Further improvement can be obtained by constructing an outer polyblock approximation [96] for $\bar{\mathcal{Q}}^\star \cap \mathcal{G}$ that lies inside $\bar{\mathcal{Q}}^\star$. If $\{\hat{\mathbf{v}}_i\}_{i\in\hat{\mathcal{V}}}$ are the proper vertices of the polyblock, it is easy to see that an improved bound is given by $\min_{i\in\hat{\mathcal{V}}} f_0(\hat{\mathbf{v}}_i)$. Though interesting, in this volume we do not consider these possible extensions, which can be carried out in a straightforward manner. But we refer the reader to [5, chap. 2, sec. 7], where similar bound improving techniques are discussed in the context of (difference of) monotonic optimization problems.

$\mathbf{a}_i = \boldsymbol{\gamma}_{\min} + (\gamma_{i,\max} - \gamma_{i,\min})\mathbf{e}_i$ lies inside $\mathcal{G}$, i.e., $\mathbf{a}_i \in \mathcal{G}$ [see $\mathbf{a}_1$ in Figure 2.4(c)]. Therefore, when $\mathbf{a}_i \in \mathcal{G}$, $\bar{\gamma}_i^\star = \gamma_{i,\max}$. Otherwise (i.e., $\mathbf{a}_i \notin \mathcal{G}$), $\bar{\gamma}_i^\star$ is limited by the power constraints. In this case, the first constraint of problem (2.35) can be safely dropped and the resulting problem can be readily converted into a standard geometric program (or GP) [22] so that the solution can be obtained numerically by using a GP solver, for example, GGPLAB, GPPOSY, GPCVX [15]. However, it turns out that the particular structure of problem (2.35) allows us to analytically find the optimal value. This provides a more computationally efficient way to compute $\phi_{\rm lb}^{\rm Imp}(\mathcal{Q})$ without relying on a GP solver. This method is described soon after the following important property of $\phi_{\rm lb}^{\rm Imp}(\mathcal{Q})$ is established.

Lemma 2.5. For any $\mathcal{Q} \subseteq \mathcal{Q}_{\rm init}$, the lower bound $\phi_{\rm lb}^{\rm Imp}(\mathcal{Q})$ (2.34) is better than the basic lower bound $\phi_{\rm lb}^{\rm Basic}(\mathcal{Q})$ (2.14), i.e., $\phi_{\min}(\mathcal{Q}) \geq \phi_{\rm lb}^{\rm Imp}(\mathcal{Q}) \geq \phi_{\rm lb}^{\rm Basic}(\mathcal{Q})$.

Proof. If $\boldsymbol{\gamma}_{\min} \notin \mathcal{G}$, we have $\phi_{\min}(\mathcal{Q}) = \phi_{\rm lb}^{\rm Imp}(\mathcal{Q}) = \phi_{\rm lb}^{\rm Basic}(\mathcal{Q}) = 0$. Otherwise, i.e., when $\boldsymbol{\gamma}_{\min} \in \mathcal{G}$ we obtain

$$\begin{aligned}\phi_{\min}(\mathcal{Q}) = \inf_{\boldsymbol{\gamma}\in\mathcal{Q}} \tilde{f}(\boldsymbol{\gamma}) = \inf_{\boldsymbol{\gamma}\in\mathcal{G}\cap\mathcal{Q}} \tilde{f}(\boldsymbol{\gamma}) = \inf_{\boldsymbol{\gamma}\in\mathcal{G}\cap\mathcal{Q}} f_0(\boldsymbol{\gamma}) \\ \geq f_0(\bar{\boldsymbol{\gamma}}^\star) = \phi_{\rm lb}^{\rm Imp}(\mathcal{Q}) \geq f_0(\boldsymbol{\gamma}_{\max}) = \phi_{\rm lb}^{\rm Basic}(\mathcal{Q}), \quad (2.36)\end{aligned}$$

where the first equality is from (2.9), the second equality follows from the fact that $\mathcal{G} \cap \mathcal{Q}$ is nonempty and $\tilde{f}(\boldsymbol{\gamma}) = 0$ for all $\boldsymbol{\gamma} \in \mathcal{Q} \setminus (\mathcal{G} \cap \mathcal{Q})$, the third equality follows from $\tilde{f}(\boldsymbol{\gamma}) = f_0(\boldsymbol{\gamma})$ for all $\boldsymbol{\gamma} \in \mathcal{G} \cap \mathcal{Q}$, the first inequality follows by noting that $\bar{\boldsymbol{\gamma}}^\star \geq \boldsymbol{\gamma}$ for all $\boldsymbol{\gamma} \in \mathcal{Q} \cap \mathcal{G}$ and f_0 is monotonically decreasing in each dimension, and the last inequality follows since $\boldsymbol{\gamma}_{\max} \geq \bar{\boldsymbol{\gamma}}^\star$ and f_0 is monotonically decreasing. □

We describe now an efficient method to find $\bar{\gamma}_i^\star$ by solving problem (2.35) when $\boldsymbol{\gamma}_{\min} \in \mathcal{G}$ and $\mathbf{a}_i \notin \mathcal{G}$. We can assume without loss of generality that $\gamma_{l,\min} > 0$ for all $l \in \mathcal{L} \setminus \{i\}$; the method can be extended to the case where there are links for which $\gamma_{l,\min} = 0$ for some $l \in \mathcal{L} \setminus \{i\}$. In such cases the original problem (2.35) is equivalent to

a modified problem obtained by eliminating the dimensions $l \in \mathcal{L} \setminus \{i\}$ (i.e., link indexes) for which $\gamma_{l,\min} = 0$.

The method can be summarized as follows. By using the equality constraints we eliminate the $L-1$ variables $(p_l)_{l\in\mathcal{L}\setminus\{i\}}$ and transform problem (2.35) into a single-variable optimization problem (with the variable p_i). This facilitates finding the optimal power $p_i^\star$ (and implicitly $\bar{\boldsymbol{\gamma}}_i^\star$), in an efficient and straightforward manner.

For a detailed description of the above method it is useful to introduce a virtual network obtained from the original network by removing the ith link. Such a network is referred to as *reduced* network. For notational convenience let us define the following vectors and matrices associated to the reduced network: $\bar{\mathbf{p}}_i$ and $\boldsymbol{\gamma}_{\min,i}$ are obtained from $\mathbf{p}$ and $\boldsymbol{\gamma}_{\min}$ by removing the ith entries, i.e., $\bar{\mathbf{p}}_i = (p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_L)$ and $\bar{\boldsymbol{\gamma}}_{\min,i} = (\gamma_{1,\min}, \ldots, \gamma_{i-1,\min}, \gamma_{i+1,\min}, \ldots, \gamma_{L,\min})$; similarly, $\bar{\mathbf{B}}_i(\bar{\boldsymbol{\gamma}}_{\min,i})$ and $\bar{\mathbf{G}}_i$ are obtained from $\mathbf{B}(\boldsymbol{\gamma}_{\min})$ and $\mathbf{G}$ [see (2.32)] by removing the ith rows and the ith columns. It is important to note that if SINR vector $\boldsymbol{\gamma}_{\min}$ is achievable in the original network then $\bar{\boldsymbol{\gamma}}_{\min,i}$ is also achievable in the reduced network.

Now we turn to problem (2.35). By rearranging the terms, the equality constraints can be expressed compactly as

$$[\mathbf{I} - \bar{\mathbf{B}}_i(\bar{\boldsymbol{\gamma}}_{\min,i})\bar{\mathbf{G}}_i]\bar{\mathbf{p}}_i + \mathbf{d}_i(\bar{\boldsymbol{\gamma}}_{\min,i})p_i = \sigma^2\bar{\mathbf{B}}_i(\bar{\boldsymbol{\gamma}}_{\min,i})\mathbf{1}, \tag{2.37}$$

where

$$\mathbf{d}_i(\bar{\boldsymbol{\gamma}}_{\min,i}) = -\left(\frac{g_{i1}}{g_{11}}\gamma_{1,\min}, \ldots, \frac{g_{ii-1}}{g_{i-1i-1}}\gamma_{i-1,\min}, \frac{g_{ii+1}}{g_{i+1i+1}}\gamma_{i+1,\min}, \ldots \right.$$
$$\left. \ldots, \frac{g_{iL}}{g_{LL}}\gamma_{L,\min}\right).$$

Similarly to (2.33), let us denote

$$\bar{\mathbf{A}}_i(\bar{\boldsymbol{\gamma}}_{\min,i}) = \mathbf{I} - \bar{\mathbf{B}}_i(\bar{\boldsymbol{\gamma}}_{\min,i})\bar{\mathbf{G}}_i \;;\; \bar{\mathbf{b}}_i(\bar{\boldsymbol{\gamma}}_{\min,i}) = \sigma^2\bar{\mathbf{B}}_i(\bar{\boldsymbol{\gamma}}_{\min,i})\mathbf{1} \tag{2.38}$$

and rewrite (2.37) equivalently as

$$\bar{\mathbf{A}}_i(\bar{\boldsymbol{\gamma}}_{\min,i})\bar{\mathbf{p}}_i + \mathbf{d}_i(\bar{\boldsymbol{\gamma}}_{\min,i})p_i = \bar{\mathbf{b}}_i(\bar{\boldsymbol{\gamma}}_{\min,i}). \tag{2.39}$$

Since $\boldsymbol{\gamma}_{\min} \in \mathcal{G}$ it follows that the SINR vector $\bar{\boldsymbol{\gamma}}_{\min,i} > \mathbf{0}$ is achievable in the reduced network. Thus, Theorem 2.4 (applied to the reduced network) implies that the spectral radius of the matrix $\bar{\mathbf{B}}_i(\bar{\boldsymbol{\gamma}}_{\min,i})\bar{\mathbf{G}}_i$ is strictly smaller than one, i.e., $\rho(\bar{\mathbf{B}}_i(\bar{\boldsymbol{\gamma}}_{\min,i})\bar{\mathbf{G}}_i) < 1$. This, in turn, ensures that matrix $\bar{\mathbf{A}}_i(\bar{\boldsymbol{\gamma}}_{\min,i})$ is invertible and its inverse has nonnegative entries, i.e., $\bar{\mathbf{A}}_i^{-1}(\bar{\boldsymbol{\gamma}}_{\min,i}) \geq \mathbf{0}$ [51, thm. 2.5.3, items 2 and 17]. Therefore, we can parameterize all solutions of (2.37), using p_i as a free parameter [22, sec. C.5, p. 681]. Thus, we obtain

$$\begin{aligned}\begin{bmatrix}\bar{\mathbf{p}}_i \\ p_i\end{bmatrix} &= \begin{bmatrix}-\bar{\mathbf{A}}_i^{-1}(\bar{\boldsymbol{\gamma}}_{\min,i})\mathbf{d}_i(\bar{\boldsymbol{\gamma}}_{\min,i}) \\ 1\end{bmatrix} p_i + \begin{bmatrix}\bar{\mathbf{A}}_i^{-1}(\bar{\boldsymbol{\gamma}}_{\min,i})\bar{\mathbf{b}}_i(\bar{\boldsymbol{\gamma}}_{\min,i}) \\ 0\end{bmatrix} \\ &= \begin{bmatrix}\bar{\mathbf{q}}_i \\ q_i\end{bmatrix} p_i + \begin{bmatrix}\bar{\mathbf{s}}_i \\ s_i\end{bmatrix}, \end{aligned} \tag{2.40}$$

where $q_i = 1$, $s_i = 0$, $\bar{\mathbf{q}}_i = -\bar{\mathbf{A}}_i^{-1}(\bar{\boldsymbol{\gamma}}_{\min,i})\mathbf{d}_i(\bar{\boldsymbol{\gamma}}_{\min,i})$, and $\bar{\mathbf{s}}_i = \bar{\mathbf{A}}_i^{-1}(\bar{\boldsymbol{\gamma}}_{\min,i})\bar{\mathbf{b}}_i(\bar{\boldsymbol{\gamma}}_{\min,i})$. The vectors $\bar{\mathbf{q}}_i$ and $\bar{\mathbf{s}}_i$ are introduced for notational simplicity and they have the following structure:

$$\bar{\mathbf{q}}_i = (q_1, \ldots, q_{i-1}, q_{i+1}, \ldots, q_L) \; ; \; \bar{\mathbf{s}}_i = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_L).$$

Furthermore, since $\bar{\mathbf{A}}_i^{-1}(\bar{\boldsymbol{\gamma}}_{\min,i}) \geq \mathbf{0}$ and by noting that $\mathbf{d}_i(\bar{\boldsymbol{\gamma}}_{\min,i}) \leq \mathbf{0}$ and $\mathbf{b}_i(\bar{\boldsymbol{\gamma}}_{\min,i}) \geq \mathbf{0}$ [see (2.38)], we can see that all entries in vectors $\bar{\mathbf{q}}_i$ and $\bar{\mathbf{s}}_i$ are nonnegative, $\bar{\mathbf{q}}_i \geq \mathbf{0}$ and $\bar{\mathbf{s}}_i \geq \mathbf{0}$. Finally, we can rewrite parametrization (2.40) as

$$p_j = q_j p_i + s_j, \; j \in \mathcal{L}, \tag{2.41}$$

where $q_j \geq 0$, $s_j \geq 0$ for all $j \in \mathcal{L}$, and $q_i = 1$, $s_i = 0$.

Next we use parametrization (2.41) to convert problem (2.35) (with L power variables) into an equivalent one with a single-power variable p_i. To do this, we first express the objective function of problem (2.35) $g_i(\mathbf{p})$ as a function of single variable p_i, i.e.,

$$g_i(\mathbf{p}) = \frac{g_{ii}p_i}{\sigma^2 + \sum\limits_{j \neq i} g_{ji}p_j} = \frac{g_{ii}p_i}{\sigma^2 + \sum\limits_{j \neq i} g_{ji}\,(q_j p_i + s_j)} = \bar{g}_i(p_i). \tag{2.42}$$

The sum-power constraints of problem (2.35) (i.e., $\sum_{l \in \mathcal{O}(n)} p_l \leq p_n^{\max}$, $n \in \mathcal{T}$) can be expressed as

$$p_i \;\; \leq \;\; \frac{p_n^{\max} - \sum_{l \in \mathcal{O}(n)} s_l}{\sum_{l \in \mathcal{O}(n)} q_l}, \; n \in \mathcal{T}. \tag{2.43}$$

Furthermore, since $q_j \geq 0$, $s_j \geq 0$, all L nonnegativity power constraints of problem (2.35) can be replaced by $p_i \geq 0$, i.e., $p_i \geq 0$ in parametrization (2.41) implies that $p_j \geq 0$ for all $j \in \mathcal{L}$. Recall that we consider the nontrivial case $\mathbf{a}_i \notin \mathcal{G}$, and therefore the first inequality constraint of problem (2.35) can be safely dropped, and therefore problem (2.35) can be expressed equivalently as

$$\begin{aligned} \text{maximize} \quad & \bar{g}_i(p_i) \\ \text{subject to} \quad & p_i \leq \frac{p_n^{\max} - \sum_{l \in \mathcal{O}(n)} s_l}{\sum_{l \in \mathcal{O}(n)} q_l}, \ n \in \mathcal{T} \\ & p_i \geq 0, \end{aligned} \tag{2.44}$$

where the variable is p_i. By recalling that $s_l \geq 0$ for all $l \in \mathcal{L}$, it is easy to see that the first derivative of the objective function $\bar{g}_i(p_i)$ is strictly positive. Hence, the maximum $\bar{g}_i(p_i)$ can be found by increasing p_i until one power constraint become active. Thus, in the case of $\mathbf{a}_i \notin \mathcal{G}$, we have

$$p_i^\star = \min_{n \in \mathcal{T}} \frac{p_n^{\max} - \sum_{l \in \mathcal{O}(n)} s_l}{\sum_{l \in \mathcal{O}(n)} q_l} \tag{2.45}$$

and we can express the optimal $\bar{\gamma}_i^\star$ as $\bar{\gamma}_i^\star = \bar{g}_i(p_i^\star)$. Hence, the general solution of problem (2.35) can be expressed as

$$\bar{\gamma}_i^\star = \begin{cases} \gamma_{i,\max} & \mathbf{a}_i \in \mathcal{G} \\ \bar{g}_i(p_i^\star) & \text{otherwise.} \end{cases} \tag{2.46}$$

Note that, the method presented for checking $\boldsymbol{\gamma}_{\min} \in \mathcal{G}$ (i.e., Algorithm 2.3.1) can be readily applied to check the condition $\mathbf{a}_i \in \mathcal{G}$ in (2.46) as well.

2.3.2.2 Improved Upper Bound

Based on monotonicity of f_0, L tighter upper bounds can be easily obtained by evaluating f_0 at the vertices of $\bar{\mathcal{Q}}^\star$ adjacent to $\boldsymbol{\gamma}_{\min}$. Specifically, they are given by $f_0(\bar{\mathbf{a}}_l)$, $l \in \mathcal{L}$, where $\bar{\mathbf{a}}_l = \boldsymbol{\gamma}_{\min} + (\bar{\gamma}_l^\star - \gamma_{l,\min})\mathbf{e}_l$ [see $\bar{\mathbf{a}}_1$ and $\bar{\mathbf{a}}_2$ in Figures 2.4(b) and 2.4(c)]. Note that the values $\bar{\gamma}_l^\star, l \in \mathcal{L}$ have already been found for computing the improved lower bound $\phi_{\text{lb}}^{\text{Imp}}(\mathcal{Q})$ (2.34). Let $l^\star$ be the index of the vertex which provide

the best (smallest) upper bound, i.e., $l^\star = \arg\min_{l\in\mathcal{L}} f_0(\bar{\mathbf{a}}_l)$. Thus, the first improved upper bound is given by

$$\phi_{\text{ub}}^{\text{Imp}}(\mathcal{Q}) = \begin{cases} f_0(\bar{\mathbf{a}}_{l^\star}) & \boldsymbol{\gamma}_{\min} \in \mathcal{G} \\ 0 & \text{otherwise.} \end{cases} \tag{2.47}$$

The following lemma ensures that $\phi_{\text{ub}}^{\text{Imp}}(\mathcal{Q})$ is tighter than the basic upper bound $\phi_{\text{ub}}^{\text{Basic}}(\mathcal{Q})$.

Lemma 2.6. For any $\mathcal{Q} \subseteq \mathcal{Q}_{\text{init}}$ and $\check{\boldsymbol{\gamma}} \in \mathcal{G} \cap \mathcal{Q}$ we have $\phi_{\min}(\mathcal{Q}) \leq f_0(\check{\boldsymbol{\gamma}}) \leq f_0(\boldsymbol{\gamma}_{\min}) = \phi_{\text{lb}}^{\text{Basic}}(\mathcal{Q})$.

Proof. First note from (2.36) that, $\phi_{\min}(\mathcal{Q}) = \inf_{\boldsymbol{\gamma}\in\mathcal{G}\cap\mathcal{Q}} f_0(\boldsymbol{\gamma})$. Moreover, by noting that $\check{\boldsymbol{\gamma}} \in \mathcal{G} \cap \mathcal{Q}$, we have $\inf_{\boldsymbol{\gamma}\in\mathcal{G}\cap\mathcal{Q}} f_0(\boldsymbol{\gamma}) \leq f_0(\check{\boldsymbol{\gamma}})$ and since $\boldsymbol{\gamma}_{\min} \leq \check{\boldsymbol{\gamma}}$ and f_0 is monotonically decreasing in each dimension, we have $f_0(\check{\boldsymbol{\gamma}}) \leq f_0(\boldsymbol{\gamma}_{\min})$. Thus, we can combine these relations together and the result follows. □

We can further improve the previously obtained bound by using efficient local optimization techniques. Specifically, we can use as an initial point $\boldsymbol{\gamma} = \bar{\mathbf{a}}_{l^\star}$ and (locally) minimize $f_0(\boldsymbol{\gamma})$ subject to $\boldsymbol{\gamma} \in \mathcal{G} \cap \mathcal{Q}$, i.e.,

$$\begin{array}{ll} \text{minimize} & f_0(\boldsymbol{\gamma}) \\ \text{subject to} & \boldsymbol{\gamma} \in \mathcal{G} \cap \mathcal{Q}, \end{array} \tag{2.48}$$

where the variable is $\boldsymbol{\gamma}$. Let us denote the obtained local optimum by $\boldsymbol{\gamma}_{\text{ImpCGP}}$. Thus, the second improved upper bound is given by

$$\phi_{\text{ub}}^{\text{ImpCGP}}(\mathcal{Q}) = \begin{cases} f_0(\boldsymbol{\gamma}_{\text{ImpCGP}}) & \boldsymbol{\gamma}_{\min} \in \mathcal{G} \\ 0 & \text{otherwise.} \end{cases} \tag{2.49}$$

One simple approach to efficiently compute $\boldsymbol{\gamma}_{\text{ImpCGP}}$ via complementary geometric programming (or CGP) [6] is presented in Appendix A.

Since all improved bounds are tighter than the basic ones (see Lemmas 2.5 and 2.6), any possible combination of a lower and an upper bound pair must also satisfy the conditions **C1** and **C2**. This ensures the convergence of the Algorithm 2.1.

2.4 Extensions to Multicast Networks

In this section we consider the problem of WSRMax in multicast networks [i.e., problem (2.4)] and show how Algorithm 2.1 can be adapted to find the solution of problem (2.4). By noting the monotonically increasing property of $\log(\cdot)$ function, problem (2.4) can be expressed in the following equivalent form:

$$\begin{array}{ll} \text{maximize} & \sum_{n\in\mathcal{T}}\sum_{m=1}^{M_n}\beta_n^m \log\big(1+\min\limits_{l\in\mathcal{O}^m(n)} \mathrm{SINR}_n^{ml}(\mathbf{p})\big) \\ \text{subject to} & \sum_{m=1}^{M_n} p_n^m \le p_n^{\max},\ n\in\mathcal{T} \\ & p_n^m \ge 0,\ n\in\mathcal{T},\ m=1,\ldots,M_n, \end{array} \tag{2.50}$$

where the variable is $(p_n^m)_{n\in\mathcal{T},\ m=1,\ldots,M_n}$. By introducing auxiliary variables $\gamma_n^m,\ n\in\mathcal{T}, m=1,\ldots,M_n$, we can equivalently express problem (2.50) as

$$\begin{array}{ll} \text{minimize} & \sum_{n\in\mathcal{T}}\sum_{m=1}^{M_n} -\beta_n^m \log(1+\gamma_n^m) \\ \text{subject to} & \gamma_n^m \le \mathrm{SINR}_n^{ml}(\mathbf{p}), \quad \begin{array}{l} n\in\mathcal{T},\ m=1,\ldots,M_n, \\ l\in\mathcal{O}^m(n) \end{array} \\ & \sum_{m=1}^{M_n} p_n^m \le p_n^{\max},\ n\in\mathcal{T} \\ & p_n^m \ge 0,\ n\in\mathcal{T},\ m=1,\ldots,M_n, \end{array} \tag{2.51}$$

where the variables are $(p_n^m)_{n\in\mathcal{T},\ m=1,\ldots,M_n}$ and $(\gamma_n^m)_{n\in\mathcal{T},\ m=1,\ldots,M_n}$. A close comparison of problems (2.51) and (2.5) reveals that they have a very similar structure. Therefore, the branch and bound method (i.e., Algorithm 2.1) can be applied to solve problem (2.51) by redefining appropriately the following sets and functions.

1. $\boldsymbol{\gamma}=(\gamma_1,\ldots,\gamma_L)$ is replaced by $\boldsymbol{\gamma}=(\gamma_n^m)_{n\in\mathcal{T},\ m=1,\ldots,M_n}$.
2. $f_0(\boldsymbol{\gamma})$ is replaced by $\tilde{f}_0(\boldsymbol{\gamma})$, where $\tilde{f}_0(\boldsymbol{\gamma}) = \sum_{n\in\mathcal{T}}\sum_{m=1}^{M_n} -\beta_n^m\log(1+\gamma_n^m)$.
3. $\mathcal{G}$ is replaced by $\tilde{\mathcal{G}}$, where

$$\tilde{\mathcal{G}} = \left\{ \boldsymbol{\gamma} \left| \begin{array}{ll} \gamma_n^m \le \mathrm{SINR}_n^{ml}(\mathbf{p}), & n\in\mathcal{T},\ m=1,\ldots,M_n, \\ & l\in\mathcal{O}^m(n) \\ \sum_{m=1}^{M_n} p_n^m \le p_n^{\max}, & n\in\mathcal{T} \\ p_n^m \ge 0, & n\in\mathcal{T},\ m=1,\ldots,M_n \end{array} \right. \right\}.$$

4. $\mathcal{Q}_{\text{init}}$ is replaced by $\tilde{\mathcal{Q}}_{\text{init}}$, where

$$\tilde{\mathcal{Q}}_{\text{init}} = \left\{ \boldsymbol{\gamma} \;\middle|\; 0 \leq \gamma_n^m \leq \min_{l \in \mathcal{O}^m(n)} g_{ll} \frac{p_n^{\max}}{\sigma^2},\ n \in \mathcal{T},\ m = 1, \ldots, M_n \right\}.$$

5. $\mathcal{Q}$ is replaced by $\tilde{\mathcal{Q}}$, where

$$\tilde{\mathcal{Q}} = \left\{ \boldsymbol{\gamma} \,\middle|\, \gamma_{n,\min}^m \leq \gamma_n^m \leq \gamma_{n,\max}^m,\ n \in \mathcal{T},\ m = 1, \ldots, M_n \right\}.$$

Note that the definitions of the lower and upper bound functions provided in the case of singlecast networks [i.e., (2.14), (2.15), (2.34), and (2.47)] are applicable in the case of multicast networks as well. However, instead of the given efficient methods based on M-matrix theory [51, p. 112] for checking $\boldsymbol{\gamma} \in \mathcal{G}$ (see Algorithm 2.3.1) and for evaluating $\bar{\gamma}_i^\star$ [see (2.46)], in the case of multicast networks, we have to rely on a linear programming (LP) or a GP solver.

2.5 Numerical Examples

In this section we first compare the impact of the considered lower bounds and upper bounds (Section 2.3) on the convergence of the branch and bound method (Algorithm 2.1 in Section 2.2). Next, we provide various applications of Algorithm 2.1 and numerical examples for the considered applications. In summary, those applications include: sum-rate maximization in singlecast wireless networks, the problem of maximum weighted link scheduling for wireless multihop networks [120, sec. III-B,V-A], [68, sec. 4], cross-layer control policies for network utility maximization (NUM) in multihop wireless networks [43, sec. 5], finding achievable rate regions in singlecast, as well as in multicast, wireless networks.

To simplify the presentation we use the abbreviations: LB_{Basic} for the basic lower bound given in (2.14), UB_{Basic} for the basic upper bound given in (2.15), LB_{Imp} for the improved lower bound given in (2.34), UB_{Imp} for the improved upper bound given in (2.47), and $\text{UB}_{\text{ImpCGP}}$ for the improved upper bound given in (2.49).

2.5.1 Impact of Different Lower Bounds and Upper Bounds on BB

To gain insight into the impact of the lower and upper bounds on the convergence of Algorithm 2.1, we focus first on the problem of sum-rate maximization in a simple bipartite network of degree 1 [see Figure 2.5(a)]. The channel power gain between distinct nodes are modeled as

$$|h_{ij}|^2 = \mu^{|i-j|} c_{ij},\ i,j \in \mathcal{L}, \tag{2.52}$$

where c_{ij}s are small-scale fading coefficients and the scalar $\mu \in [0,1]$ is referred to as the interference coupling index, which parameterizes the interference between direct links. The fading coefficients are assumed to be exponentially distributed independent random variables to model Rayleigh fading. An arbitrarily generated set $\acute{\mathcal{C}}$ of fading coefficients, where $\acute{\mathcal{C}} = \{c_{ij} \mid\ i,j \in \mathcal{L}\}$ is referred to as a *single fading realization*; we use a discrete argument t sometimes, to indicate the fading realization index. For example $\acute{\mathcal{C}}(t)$ represents the tth fading realization. We define the signal-to-noise ratio (SNR) operating point as ($p_n^{\max} = p_0^{\max}$ for all $n \in \mathcal{T}$)

$$\text{SNR} = \frac{p_0^{\max}}{\sigma^2}. \tag{2.53}$$

We consider first the nonfading case, i.e., $c_{ij} = 1, i,j \in \mathcal{L}$, and Algorithm 2.1 was run with all possible combinations of the lower and upper bound pairs. Figure 2.6 shows the evolution of the upper and

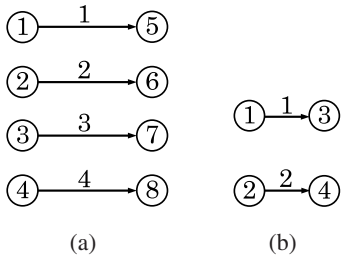


Fig. 2.5 (a) Bipartite network, degree 1, $N = 8$, $L = 4$; (b) Bipartite network, degree 1, $N = 4$, $L = 2$.

lower bounds for the optimal value of problem (2.5)[6] for SNR = 15 dB, $\mu = 0.25$, and $\beta_l = 0.25$ for all $l \in \mathcal{L}$. Specifically in Figure 2.6(a), we used the basic lower bound $\mathrm{LB}_{\mathrm{Basic}}$ in conjunction with all upper

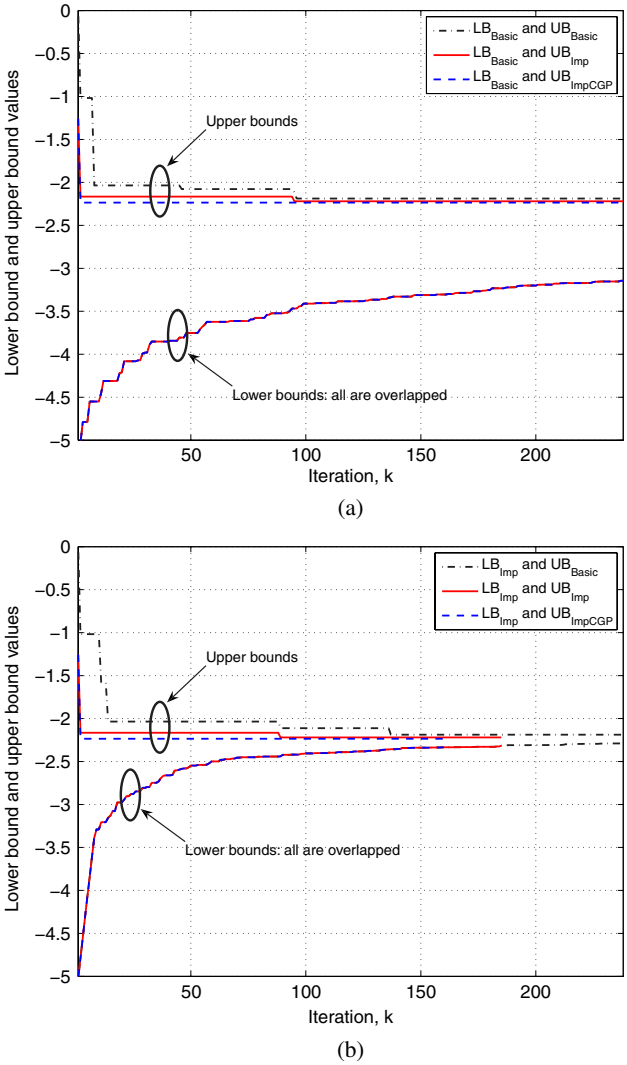


Fig. 2.6 Evolution of lower and upper bounds: (a) Basic lower bound in conjunction with all upper bounds; (b) Improved lower bound in conjunction with all upper bounds.

[6] The optimal value of problem (2.5) is the negative of the optimal value of problem (2.2).

bounds and in Figure 2.6(b) we used the improved lower bound $\mathrm{LB}_{\mathrm{Imp}}$ in conjunction with all upper bounds. The results show that the convergence speed of Algorithm 2.1 can be substantially increased by improving the lower bound whilst the tightness of the upper bound has a much reduced impact. Note that this is in general the behavior of a branch and bound method, where an approximative solution can be found relatively fast but certifying it typically takes a much larger number of iterations [16]. Note that in both Figures 2.6(a) and 2.6(b) the evolution of lower bounds is independent of the upper bound used. This is due to the fact that in each iteration the branching mechanism depends only on the lower bound.

In order to provide a statistical description of the speed of convergence we turn to the fading case and run Algorithm 2.1 for a large number of fading realizations. For each one we store the number of iterations and the total CPU time required to find the optimal value of problem (2.5) within an accuracy of $\epsilon = 10^{-1}$ for $\mathrm{SNR} = 15$ dB, $\mu = 0.25$, and $\beta_l = 0.25$ for all $l \in \mathcal{L}$. Figure 2.7 shows the empirical cumulative distribution function (CDF) plots of the total number of iterations [Figure 2.7(a)] and the total CPU time [Figure 2.7(b)] for all possible combinations of lower and upper bounds pairs. Figure 2.7(a) shows that, irrespective of the upper bound we use, the improved lower bound $\mathrm{LB}_{\mathrm{Imp}}$ provides a remarkable reduction in the total number of iterations when compared to $\mathrm{LB}_{\mathrm{Basic}}$. Results further show that, even though the improved upper bound $\mathrm{UB}_{\mathrm{ImpCGP}}$ makes use of advanced optimization techniques, such as complementary geometric programming (see Algorithm A.0.1, Appendix A), the benefits from $\mathrm{UB}_{\mathrm{ImpCGP}}$ over the improved upper bound $\mathrm{UB}_{\mathrm{Imp}}$ is marginal in terms of the total number of iterations. In terms of the total CPU time [Figure 2.7(b)], significant improvements are often achieved by using the lower and upper bound pairs ($\mathrm{LB}_{\mathrm{Imp}}$, $\mathrm{UB}_{\mathrm{Imp}}$) and ($\mathrm{LB}_{\mathrm{Imp}}$, $\mathrm{UB}_{\mathrm{Basic}}$). Interestingly, the lower and upper bound pair ($\mathrm{LB}_{\mathrm{Imp}}$, $\mathrm{UB}_{\mathrm{ImpCGP}}$) performs very poorly. This behavior is due to the complexity of step 2 of Algorithm A.0.1, where we have to rely on a GP solver.

Therefore, in all of the following numerical examples, Algorithm 2.1 is run with the lower and upper bound pair ($\mathrm{LB}_{\mathrm{Imp}}$, $\mathrm{UB}_{\mathrm{Imp}}$), unless otherwise specified.

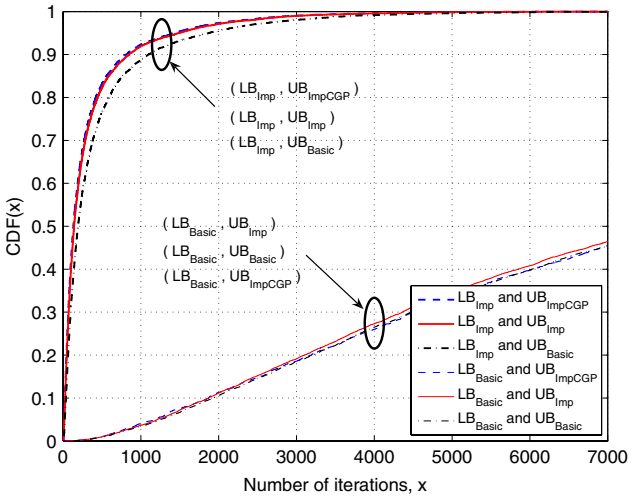


Fig. 2.7 Empirical CDF plots of: (a) Total number of iterations; (b) Total CPU time.

2.5.2 Sum-rate Maximization in Singlecast Wireless Networks

Let us now consider the problem of sum-rate maximization in a bipartite singlecast network. To evaluate the benefits from multipacket transmit/receive capabilities of nodes, we chose a network setup with

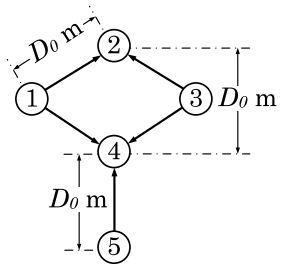


Fig. 2.8 Bipartite network, degree 3, $N = 5$, $L = 5$.

degree 3, as shown in Figure 2.8. The network is symmetric and the distances between nodes are chosen as shown in the figure. We assume an exponential path loss model, where the channel power gains between distinct nodes are given by

$$|h_{ij}|^2 = \left(\frac{d_{ij}}{d_0}\right)^{-\eta} c_{ij}, \tag{2.54}$$

where d_{ij} is the distance from the transmitter of link i to the receiver of link j, d_0 is the *far field reference distance* [64], η is the path loss exponent, and c_{ij} are defined as in (2.52). Note that the interference coefficients g_{ij} s are chosen as we discussed in Section 2.1. The first term of (2.54) represents the path loss factor and the second-term models Rayleigh small-scale fading. The SNR operating point is defined as ($p_n^{\max} = p_0^{\max}$ for all $n \in \mathcal{T}$)

$$\text{SNR} = \frac{p_0^{\max}}{\sigma^2}\left(\frac{D_0}{d_0}\right)^{-\eta}. \tag{2.55}$$

In the following simulations we set $D_0/d_0 = 10$ and $\eta = 4$.

Figure 2.9(a) shows the dependence of average sum-rate (i.e., $\beta_l = 1$ for all $l \in \mathcal{L}$) on the SNR. Results show that the average sum-rate, in the case of multipacket transmission/reception, is always better than, or equal to, the case of singlepacket transmission/reception and the performance gap increases as SNR decreases. However, as expected for practical SNR values, the benefits of multipacket transmission/reception are negligible when the receivers perform singleuser detection [124]. For comparison, we also plot the result obtained from a suboptimal solution method based on complementary geometric programming [6, 20, 30].

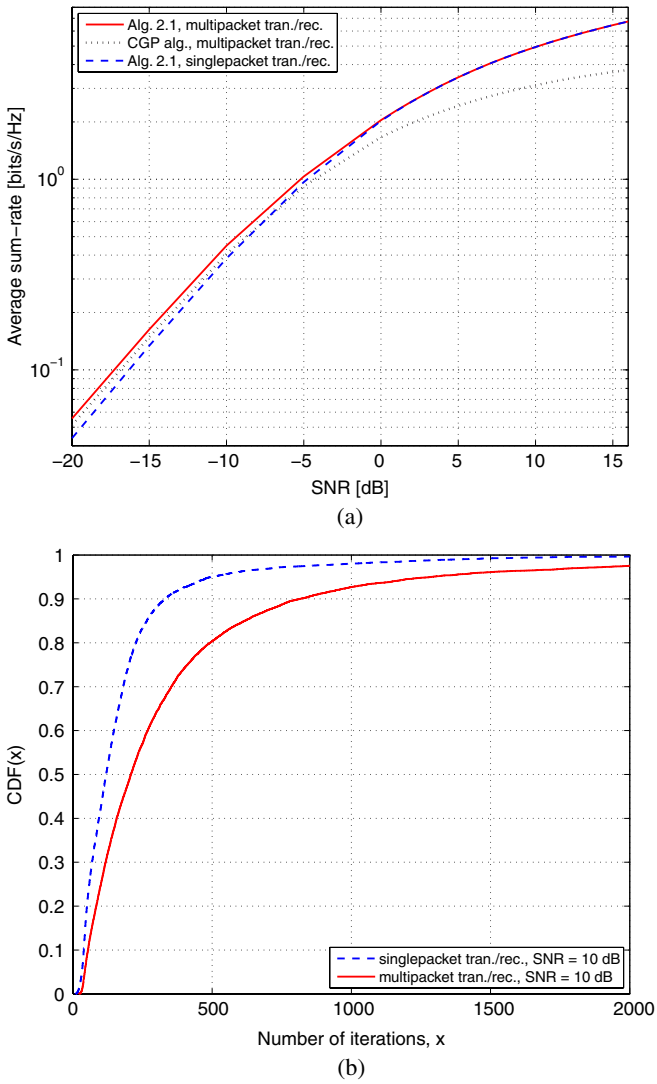


Fig. 2.9 (a) Dependence of the average sum-rate on SNR; (b) Empirical CDF of the total number of iterations.

We refer to this suboptimal method as *CGP algorithm* for the rest of the section. Note that, CGP algorithm is equivalent to running Algorithm A.0.1 (Appendix A) with $\mathcal{Q} = \mathcal{Q}_{\text{init}}$ and a proper initialization $\hat{\boldsymbol{\gamma}}$. Specifically, we found the initial $\hat{\gamma}_l,\ l \in \mathcal{L}$ according to (2.30) by

using a uniform feasible power allocation, which will be referred to as uniform initialization in the rest of the section. Let us first focus on CGP performance in the case of multipacket transmission/reception. Results show that there is a significant performance loss due to the suboptimality of CGP algorithm, especially for SNR > 0 dB. In the case of singlepacket transmission/reception, the average sum-rate that is obtained by using CGP algorithm is almost zero, irrespective of the SINR and not plotted in Figure 2.9(a) to preserve clarity. Results confirm that CGP algorithm cannot handle the huge imbalance between interference coefficient values.[7]

Figure 2.9(b) shows the empirical CDF plots of the total number of iterations required to find the sum-rate by using Algorithm 2.1, which gives insight into the complexity of Algorithm 2.1. The plots are for the case of SNR $= 10$ dB and $\epsilon = 10^{-3}$. Roughly speaking, results show that the total number of iterations required in the case of singlepacket transmission/reception is smaller compared to the case of multipacket transmission/reception.

2.5.3 Maxweight Scheduling in Multihop Wireless Networks

Next, we consider a multihop wireless network, where the nodes have only singlepacket transmit/receive capability and no node can transmit and receive simultaneously. In such setups the WSRMax problem is equivalent to the maximum weighted matching[8] (MWM) problem [92]. Polynomial time algorithms are available for the problem in the case of fixed link rates [68, sec. 4.2], [92]. When the link rates depend on the power allocation of all other links, it is worth noting that the given algorithm is able to find the MWM.

To show this, we use the symmetric multihop wireless network shown in Figure 2.10(a). The channel power gains, between nodes are given by (2.54) and the SNR operating point is given by (2.55). In the following simulations we set $D_0/d_0 = 10$ and $\eta = 4$.

[7] Recall from Figures 2.2(a) and 2.2(b) that, if nodes have singlepacket transmitter/receiver capabilities, then some of the interference coefficients are infinite.

[8] Borrowing terminology from graph theory, a matching is a set of links, no two of which share a node [92].

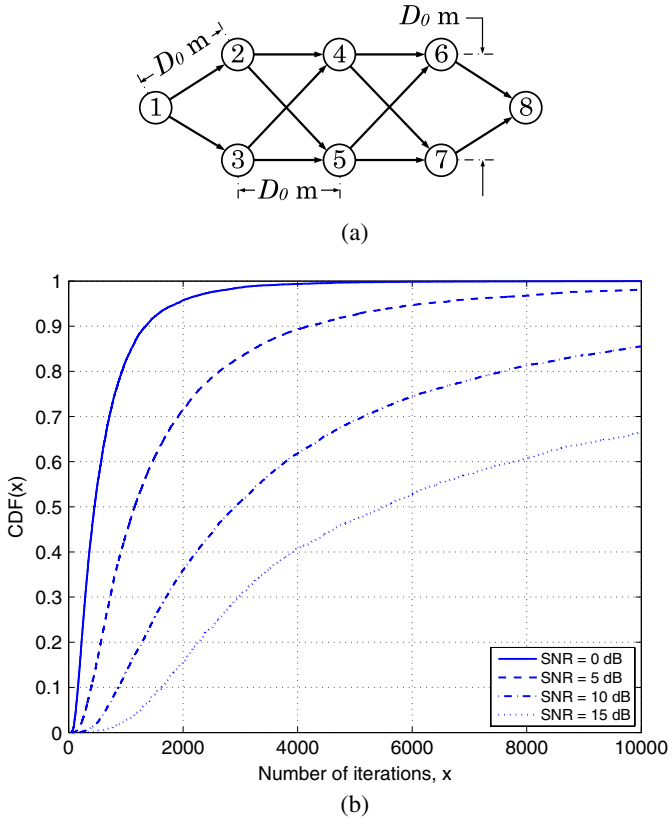


Fig. 2.10 (a) Multihop network, $N = 8$, $L = 12$; (b) Empirical CDF of the total number of iterations.

Table 2.1 shows MWMs obtained for different link weights (see the left most column) and the SNR combinations. Here we consider a non-fading scenario (i.e., $c_{ij} = 1,\ i,j \in \mathcal{L}$) and an accuracy of $\epsilon = 10^{-10}$. Results show that the smaller the SNR, the larger the number of links that are activated simultaneously in the maximum weighted matching. This is intuitively expected since, at low SNR values, node transmission power is small, and therefore the interference generated is very small so that many links are activated simultaneously.

To gain some insight into the computational complexity of the algorithm we plot the CDF of the total number of iterations by running the algorithm for a large number of fading realizations.

Table 2.1. Maximum weighted matchings.

Associated weights of the links \ SNR [dB]	−10	0	5	10

Figure 2.10(b) shows the empirical CDF plots of the total number of iterations required to terminate Algorithm 2.1 (or to find the MWM). Plots are drawn for the cases of SNR $= 0,\ 5,\ 10,$ and 15 dB, $\beta_l = 1$ for all $l \in \mathcal{L}$, and $\epsilon = 10^{-2}$. Results show that the smaller the SNR, the smaller the total number of iterations required to find the MWM. For example, in the case of SNR $= 0$ dB, with probability 0.9, the MWM is found in less than 1500 iterations. However, in the case of SNR $= 5$ dB, with the same probability 0.9, the MWM is found in less than 4000 iterations.

2.5.4 Cross-layer Control Policies for NUM

In this section we specifically consider the problem of network utility maximization subject to stability constraints [43, sec. 5]. Let us first revisit briefly the commodity description of the network. Exogenous data arrives at the source nodes and they are delivered to the destination nodes over several, possibly multihop, paths. The data is identified by their destinations, i.e., all data with the same destination are considered as a single commodity, regardless of their source. We label the commodities with integers $s = 1, \ldots, S$ $(S \leq N)$. For every node, we define $\mathcal{S}_n \subseteq \{1, \ldots, S\}$ as the set of commodities, which can arrive exogenously at node n. The network is time slotted and at each source node, a set of flow controllers decides the amount of each commodity data admitted in every time slot in the network. Let $x_n^s(t)$ denote the amount of data commodity s admitted in the network at node n during time slot t. It is assumed that the data that is successfully delivered to its destination exits the network layer. Associated with each node-commodity pair $(n, s)_{s \in \mathcal{S}_n}$ we define a concave and nondecreasing *utility function* $u_n^s(y)$, representing the "reward" received by sending data of commodity s from node n to node d_s at a long term average rate of y [bits/slot]. Thus, the NUM problem under stability constraints can be formulated as [43, sec. 5]

$$\begin{array}{ll} \text{maximize} & \sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}_n} u_n^s(y_n^s) \\ \text{subject to} & (y_n^s)_{n \in \mathcal{N}, s \in \mathcal{S}_n} \in \mathbf{\Lambda}, \end{array} \tag{2.56}$$

where the variable is $(y_n^s)_{n \in \mathcal{N}, s \in \mathcal{S}_n}$ and $\mathbf{\Lambda}$ represents the *network layer capacity region* [43, def. 3.7].

An arbitrarily close to optimal solutions for problem (2.56) is achieved by a cross-layer control policy, which consists of solving three subproblems: (1) flow control, (2) next-hop routing and in-node scheduling, and (3) RA, during each time slot [43]. The RA subproblem exactly resembles the WSRMax problem (2.2), where the weights are given by the maximum differential backlogs of network links [43]. Here, we implement the cross-layer control algorithm in [43] and, in the third step, we use Algorithm 2.1 to solve the RA subproblem. The cross-layer control algorithm is simulated for at least $\acute{T} = 10{,}000$ time slots, and the average rates $\bar{x}_n^s$ are computed by averaging the last $t_0 = 3000$ time slots, i.e., $\bar{x}_n^s = 1/t_0 \sum_{t=\acute{T}-t_0}^{\acute{T}} x_n^s(t)$. We assume that the rates corresponding to all node-commodity pairs $(n,s)_{s\in\mathcal{S}_n}, n \in \mathcal{N}$ are subject to proportional fairness, and therefore we select the utility functions $u_n^s(y) = \log_e(y)$. For a detailed description of the cross-layer control policy [43] the reader may refer to Section 3.1.2.

Two fully connected multihop wireless network setups, as shown in Figure 2.11 are considered, where all nodes have multipacket transmit/receieve capability and no node can transmit and receive simultaneously. Each of the networks consist of four nodes (i.e., $N = 4$) and two commodities, which arrive exogenously at source nodes. In the case of the first network setup, shown in Figure 2.11(a), commodity 1 arrives exogenously at node 1, and is intended for node 4; commodity 2 arrives exogenously at node 4, and is intended for node 1. Nodes are located in a square grid such that the horizontal and the vertical distance between adjacent nodes are D_0 meters [m]. In the case of the second network

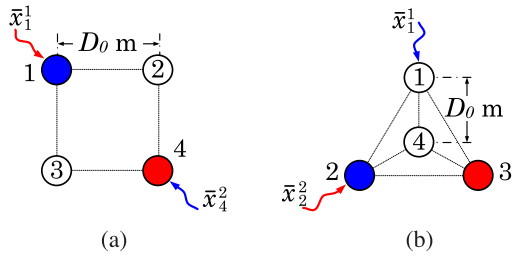


Fig. 2.11 (a) Multihop network 1, $N = 4$, fully connected, $S = 2$; (b) Multihop network 2, $N = 4$, fully connected, $S = 2$.

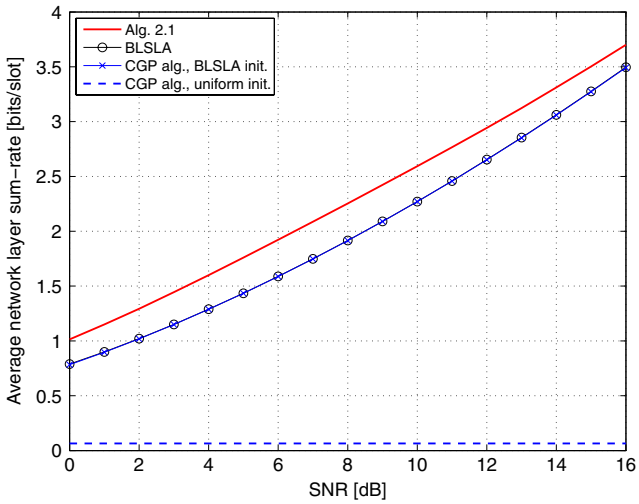
setup, shown in Figure 2.11(b), commodity 1 arrives exogenously at node 1, and is intended for node 2; commodity 2 arrives exogenously at node 2, and is intended for node 3. Nodes are located such that three of them form an equilateral triangle and the fourth one is located at its center [see Figure 2.11(b)]. It is assumed that the distance from the middle node to any other is D_0 m. The channel power gains are given by (2.54) and SNR operating point is given by (2.55). We set $D_0/d_0 = 10$ and $\eta = 4$ in the following simulation.

Figure 2.12 shows the dependence of the average network layer sum-rate on the SNR for the considered network setups. As a reference, we first consider a suboptimal and more restrictive RA policy, where only one link can be activated during each time slot. This policy is called base line single link activation (BLSLA); BLSLA policy consists of activating, during each time slot, only the link that achieves the maximum weighted rate. Other suboptimal RA policy is based on CGP algorithm (see Section 2.5.2). Specifically, we use two initialization methods for CGP algorithm: (1) the initial $\hat{\gamma}_l,\ l \in \mathcal{L}$ is found according to (2.30) by using BLSLA power allocation, (2) the uniform initialization, as discussed in Section 2.5.2.

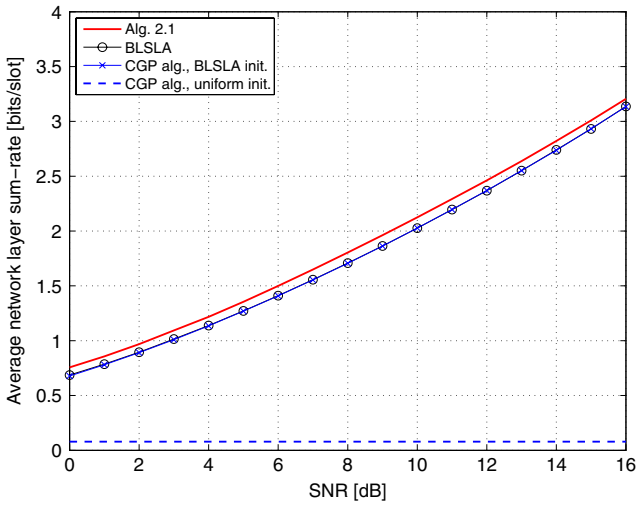
Results show that the gains obtained by using Algorithm 2.1 are always larger compared to other suboptimal methods. The relative gains achieved by Algorithm 2.1 in the case of network setup 1 [Figure 2.12(a)] are more significant than in the case of network setup 2 [Figure 2.12(b)]. Results further show that, the suboptimal CGP algorithm is very sensitive to initialization. For example, in the case of uniform initialization, CGP algorithm performs extremely poorly compared to the case of BLSLA based initialization. Moreover, in the case of BLSLA based initialization, the suboptimal CGP algorithm can not perform beyond the limits that are achieved by simple BLSLA RA policy.

2.5.5 Achievable Rate Regions in Singlecast Wireless Networks

In this section we illustrate how Algorithm 2.1 can be used to find the achievable rate region in singlecast wireless networks. Recall that we consider the case where all receiver nodes perform singleuser detection,



(a) Average networkl ayer sum-rate $\bar{x}_1^1 + \bar{x}_4^2$



(b) Average network layer sum-rate $\bar{x}_1^1 + \bar{x}_2^2$

Fig. 2.12 (a) Dependence of average network layer sum-rate on SNR for network 1; (b) Dependence of average network layer sum-rate on SNR for network 2.

and therefore the achievable rate regions we are referring to are different from the information theoretic capacity regions [1, 126, 137]. Note that the information theoretic capacity region is not known, even in the simple case of two interfering links [36].

To facilitate the graphical illustration, we consider a simple bipartite singlecast network of degree 1, as shown in Figure 2.5(b). The channel power gains are given by (2.52) and the SNR operating point is given by (2.53).

We start by defining the *directly achievable rate region*, the *instantaneous rate region*, and the *average rate region* for singlecast wireless networks. Let $\mathcal{R}^{\mathrm{DIR-SC}}(\mu, \acute{\mathcal{C}}(t), p_1^{\max}, p_2^{\max})$ denote the directly achievable rate region for a given interference coupling index μ, a given fading realization[9]

$$\acute{\mathcal{C}}(t) = \{c_{11}(t), c_{12}(t), c_{22}(t), c_{21}(t)\}, \tag{2.57}$$

and maximum node transmission power $p_1^{\max}$ and $p_2^{\max}$, i.e.,

$$\begin{aligned}
&\mathcal{R}^{\mathrm{DIR-SC}}(\mu, \acute{\mathcal{C}}(t), p_1^{\max}, p_2^{\max}) \\
&\qquad = \left\{ (R_1, R_2) \left| \begin{array}{l} R_1 \le \log\left(1 + \dfrac{c_{11}(t)p_1}{\sigma^2 + \mu c_{21}(t)p_2}\right) \\ R_2 \le \log\left(1 + \dfrac{c_{22}(t)p_2}{\sigma^2 + \mu c_{12}(t)p_1}\right) \\ 0 \le p_1 \le p_1^{\max}, \quad 0 \le p_2 \le p_2^{\max} \end{array} \right. \right\}. \qquad (2.58)
\end{aligned}$$

By invoking a time sharing argument, one can obtain the instantaneous rate region $\mathcal{R}^{\mathrm{INS-SC}}(\mu, \acute{\mathcal{C}}(t), p_1^{\max}, p_2^{\max})$; the convex hull of $\mathcal{R}^{\mathrm{DIR-SC}}(\mu, \acute{\mathcal{C}}(t), p_1^{\max}, p_2^{\max})$. That is,

$$\mathcal{R}^{\mathrm{INS-SC}}(\mu, \acute{\mathcal{C}}(t), p_1^{\max}, p_2^{\max}) = \mathrm{conv}\left\{ \mathcal{R}^{\mathrm{DIR-SC}}(\mu, \acute{\mathcal{C}}(t), p_1^{\max}, p_2^{\max}) \right\},$$

where $\mathrm{conv}\{\mathcal{R}\}$ denotes the convex hull of the set $\mathcal{R}$. As noted in [48], since the instantaneous rate region $\mathcal{R}^{\mathrm{INS-SC}}(\mu, \acute{\mathcal{C}}(t), p_1^{\max}, p_2^{\max})$ is convex, any boundary point of the rate region can be obtained by using the solution of an optimization problem in the form of (2.2) with $\beta_1 = \alpha$, $\beta_2 = (1 - \alpha)$ for some $\alpha \in [0, 1]$.

Finally, we define the average rate region $\mathcal{R}^{\mathrm{AVE-SC}}(\mu, p_1^{\max}, p_2^{\max})$ for a given interference coupling index μ and a maximum node

[9] The argument t is used to indicate the fading realization index.

transmission power $p_1^{\max}$ and $p_2^{\max}$ as $\mathcal{R}^{\text{AVE}-\text{SC}}(\mu, p_1^{\max}, p_2^{\max}) = \frac{1}{\acute{T}} \sum_{t=1}^{\acute{T}} \mathcal{R}^{\text{INS}-\text{SC}}(\mu, \acute{\mathcal{C}}(t), p_1^{\max}, p_2^{\max})$, where addition and scalar multiplication of sets is used.[10] The nonnegative integer $\acute{T}$ is the total number of fading realizations we used in averaging. Note that, any boundary point $(R_1^{\text{b}}, R_2^{\text{b}})$ of $\mathcal{R}^{\text{AVE}-\text{SC}}(\mu, p_1^{\max}, p_2^{\max})$ is obtained by using the following steps for some $\alpha \in [0,1]$: (1) solve problem (2.2) with $\beta_1 = \alpha$ and $\beta_2 = 1 - \alpha$ for $\acute{T}$ fading realizations, (2) for each fading realization $t \in \{1, \ldots, \acute{T}\}$, evaluate the rate of link 1 and 2 denoted by $r_1(t), r_2(t)$ according to (2.1), and (3) average $r_1(t)$ and $r_2(t)$ over all $\acute{T}$ fading realizations to obtain $R_1^{\text{b}} = \frac{1}{\acute{T}} \sum_{t=1}^{\acute{T}} r_1(t)$ and $R_2^{\text{b}} = \frac{1}{\acute{T}} \sum_{t=1}^{\acute{T}} r_2(t)$.

Figure 2.13(a) shows $\mathcal{R}^{\text{INS}-\text{SC}}(\mu, \acute{\mathcal{C}}(t), p_1^{\max}, p_2^{\max})$, the instantaneous rate regions for different values of μ and for an arbitrary chosen fading realization in the case of $\text{SNR} = 15$ dB. Specifically, the fading coefficients are $c_{11}(t) = 0.4185$, $c_{12}(t) = 0.3421$, $c_{22}(t) = 0.3700$, and $c_{21}(t) = 1.299$. As a reference, we also plot the directly achievable rate regions $\mathcal{R}^{\text{DIR}-\text{SC}}(\mu, \acute{\mathcal{C}}(t), p_1^{\max}, p_2^{\max})$ for all the scenarios considered. Note that the problem of finding any boundary point of $\mathcal{R}^{\text{DIR}-\text{SC}}(\mu, \acute{\mathcal{C}}(t), p_1^{\max}, p_2^{\max})$ can be easily cast as a GP, or as a problem of the form (2.35). Results show that the smaller the μ, the larger the rate regions. This is intuitively explained by noting that the smaller the μ, the smaller the interference coefficients, g_{ij} between links, and therefore the higher the rates. When $\mu \geq 0.2$, the directly achievable rate regions become nonconvex, whereas the instantaneous rate region is a triangle referred to as time division multiple access (TDMA) rate region, obtained by time sharing between the maximum rates of R_1 and R_2. Moreover, when $\mu < 0.2$, the instantaneous rate region expands beyond the TDMA rate region and for $\mu \leq 0.01$, the directly achievable rate region almost overlaps with the instantaneous rate region.

Figure 2.13(b) shows the average rate region $\mathcal{R}^{\text{AVE}-\text{SC}}(\mu, p_1^{\max}, p_2^{\max})$ for different values of μ in the case of $\text{SNR} = 15$ dB. As a reference, we also plot the region obtained by using CGP algorithm to problem (2.2). Results show that the region obtained by CGP algorithm is always worse than the average rate region. The

[10] For vector sets $\mathcal{A}_1$ and $\mathcal{A}_2$ and scalars α_1, α_2, the set $\alpha_1 \mathcal{A}_1 + \alpha_2 \mathcal{A}_2$ is defined as $\{\alpha_1 \mathbf{a}_1 + \alpha_2 \mathbf{a}_2 \mid \mathbf{a}_1 \in \mathcal{A}_1, \mathbf{a}_2 \in \mathcal{A}_2\}$ [22, p. 38].

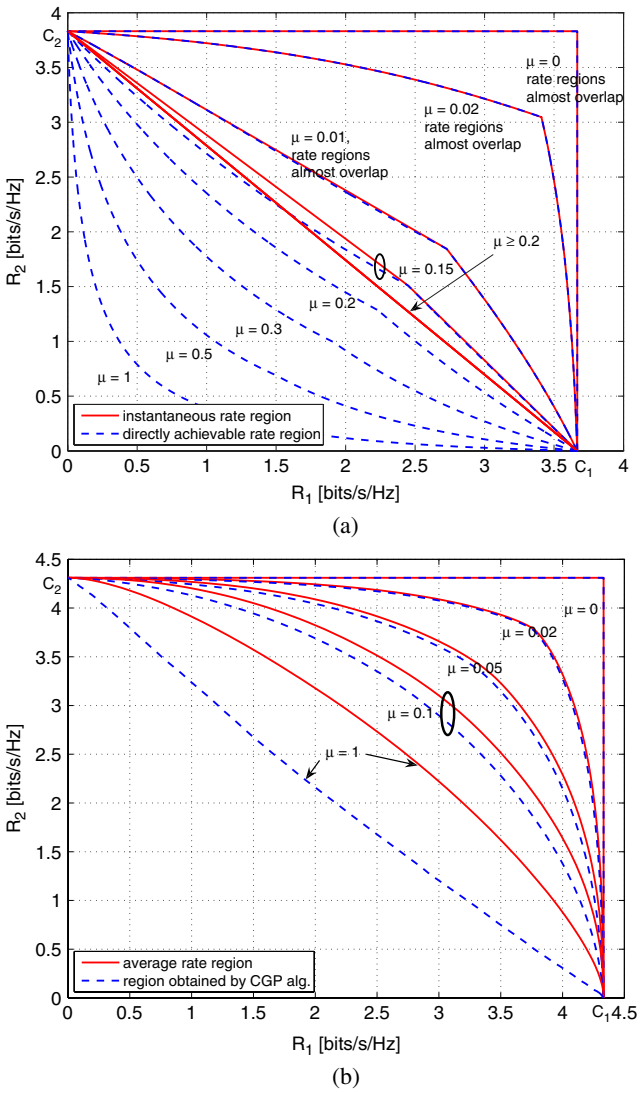


Fig. 2.13 Rate regions: (a) Directly achievable and instantaneous rate regions; (b) Average rate regions.

gap in performance is more pronounced in the case of larger values of μ. Note that, even in the case of $\mu = 1$, the average rate region is bounded by a concave function with end points C_1 and C_2, although the corresponding instantaneous rate regions used in the averaging

are triangles [see Figure 2.13(a)] in general. This phenomenon is due to the property of the set addition used in the definition of $\mathcal{R}^{\mathrm{AVE-SC}}(\mu, p_1^{\max}, p_2^{\max})$. Results also show that the smaller the μ, the larger the average rate region.

2.5.6 Achievable Rate Regions in Multicast Wireless Networks

We finally show the applicability of Algorithm 2.1 for finding the rate regions in a multicast wireless networks. A multicast with only two multicast transmissions [see Figure 2.14(a)] is considered for the sake

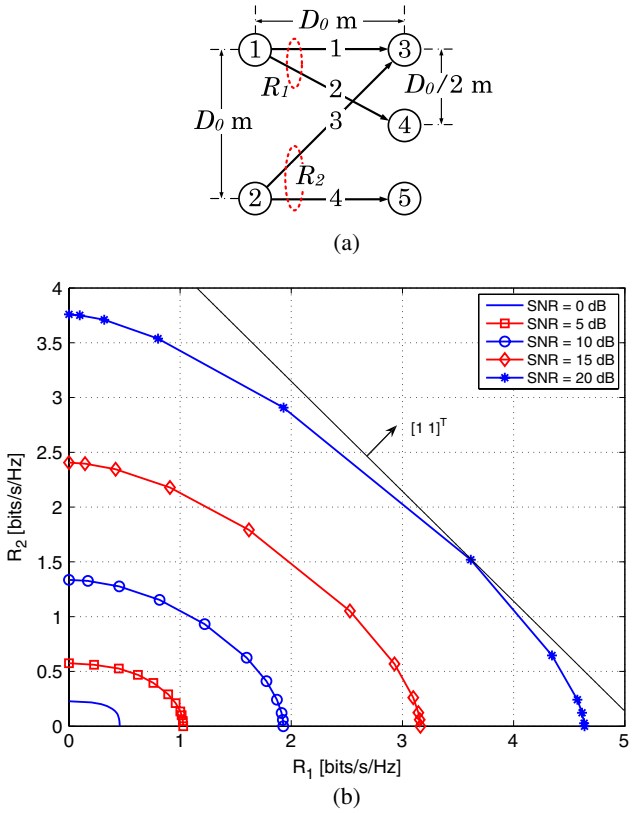


Fig. 2.14 (a) Multicast network, $\mathcal{T} = \{1,2\}$, $M_1 = 1$, $M_2 = 1$, $\mathcal{O}^1(1) = \{1,2\}$, $\mathcal{O}^1(2) = \{3,4\}$; (b) Average rate region.

of graphical illustration of the rate regions. Node 1 has common information to send to nodes 3 and 4, whereas node 2 has common information to send to nodes 3 and 5. We assume that node 3 has multipacket receiver capability. The channel power gains are given by (2.54) and SNR operating point is given by (2.55). Moreover, we set $D_0/d_0 = 10$ and $\eta = 4$.

As in the case of singlecast wireless networks, we first define the directly achievable rate region, instantaneous rate region, and the average rate region for multicast wireless networks. Particularized to the network setup in Figure 2.14(a), for a given set of interference coefficients/power gains

$$\acute{G}(t) = \{g_{11}(t), g_{22}(t), g_{33}(t), g_{44}(t), g_{14}(t), g_{32}(t)\} \tag{2.59}$$

and maximum node transmission power $p_1^{\max}$ and $p_2^{\max}$, the instantaneous rate region $\mathcal{R}^{\mathrm{INS-MC}}(\acute{G}(t), p_1^{\max}, p_2^{\max})$ is defined as

$$\mathcal{R}^{\mathrm{INS-MC}}(\acute{G}(t), p_1^{\max}, p_2^{\max}) = \mathrm{conv}\left\{\mathcal{R}^{\mathrm{DIR-MC}}(\acute{G}(t), p_1^{\max}, p_2^{\max})\right\}, \tag{2.60}$$

where $\mathcal{R}^{\mathrm{DIR-MC}}(\acute{G}(t), p_1^{\max}, p_2^{\max})$ denotes the directly achievable rate region for multicast wireless networks, i.e.,

$$\begin{aligned}
&\mathcal{R}^{\mathrm{DIR-MC}}(\acute{G}(t), p_1^{\max}, p_2^{\max}) \\
&= \left\{ (R_1, R_2) \left| \begin{array}{l} R_1 \le \log\left(1 + \dfrac{g_{11}(t)p_1^1}{\sigma^2 + g_{33}(t)p_2^1}\right) \\ R_1 \le \log\left(1 + \dfrac{g_{22}(t)p_1^1}{\sigma^2 + g_{32}(t)p_2^1}\right) \\ R_2 \le \log\left(1 + \dfrac{g_{33}(t)p_2^1}{\sigma^2 + g_{11}(t)p_1^1}\right) \\ R_2 \le \log\left(1 + \dfrac{g_{44}(t)p_2^1}{\sigma^2 + g_{14}(t)p_1^1}\right) \\ 0 \le p_1^1 \le p_1^{\max}, \quad 0 \le p_2^1 \le p_2^{\max} \end{array} \right. \right\}.
\end{aligned} \tag{2.61}$$

Finally, for a given maximum node transmission power $p_1^{\max}$ and $p_2^{\max}$, the average rate region $\mathcal{R}^{\mathrm{AVE-MC}}(p_1^{\max}, p_2^{\max})$ is defined as $\mathcal{R}^{\mathrm{AVE-MC}}(p_1^{\max}, p_2^{\max}) = \frac{1}{\acute{T}} \sum_{t=1}^{\acute{T}} \mathcal{R}^{\mathrm{INS-MC}}(\acute{G}(t), p_1^{\max}, p_2^{\max})$.

Figure 2.14(b) shows the average multicast rate region for different SNR values. Results show that, when the weights associated with rates R_1 and R_2 are same, the resulting R_1 is always greater than R_2. For example, in the case of SNR = 20 dB, we have $R_1 = 3.71$ bits/sec/Hz and $R_2 = 1.50$ bits/sec/Hz. Roughly speaking, this observation can be explained as follows: R_1 is determined by the rate of link 2 (the weakest of link 1 and link 2), R_2 is determined by the rate of link 3 (the weakest of link 3 and link 4) and rate of link 2 is larger than that of link 3 due to path losses.

2.6 Summary and Discussion

We considered the general WSRMax problem for a set of interfering links. In fact, this problem is NP-hard. A solution method, based on the branch and bound technique, was presented for solving the nonconvex WSRMax problem globally with an optimality certificate. Efficient and analytic bounds were given and their impact on convergence was numerically evaluated. The convergence speed of the given algorithm can be substantially increased by improving the lower bound, whilst the tightness of the upper bound has a much reduced impact. Numerical results showed that the algorithm converges fairly fast in all considered setups. Nevertheless, since the problem is NP-hard, the worst case complexity can be exponential in the number of variables.

The considered link-interference model is fairly general so that it can model a wide range of network topologies with various node capabilities, such as single- or multipacket transmission (or reception) and simultaneous transmission and reception. Unlike other branch and bound based solution methods for WSRMax, the method given does not require the problem to be convertible into a DC (difference of convex functions) problem. Therefore, the given method applies to a broader class of WSRMax problems (e.g., WSRMax in multicast wireless networks). Moreover, the method can also be used to maximize *any* system performance metric that can be expressed as a Lipschitz continuous and increasing function of SINR values and is not restricted to WSRMax. Given its generality, the algorithm can be adapted to address a wide range of network control and optimization

problems. Performance benchmarks for various network topologies can be obtained by back-substituting it into any network design method which relies on WSRMax. Several applications, including cross-layer network utility maximization and maximum weighted link scheduling for multihop wireless networks, as well as finding achievable rate regions for singlecast/multicast wireless networks, have been presented. Since there are a number of suboptimal but low-complex algorithms are typically used in practice, the algorithm can also be used for evaluating their performance loss.

3

Low Complexity Algorithms

In this section we first present efficient, low-complexity algorithms for the WSRMax problem in multicommodity, multichannel wireless networks by using homotopy methods [4] and complementary geometric program (or CGP) [6]. The considered problem formulation is fairly general and it allows frequency reuse by activating multiple links in the same channel simultaneously. Here, the interference is solely resolved via power control. Furthermore, the formulation allows the possibility of exploiting multichannel diversity via dynamic power allocation across the available channels. The gains that can be achieved at upper layers in terms of end-to-end rates and network congestion are quantitatively analyzed by incorporating the algorithms within Neely's cross-layer utility maximization framework [43, 80].

The algorithm based on homotopy methods, also handles the self-interference problem in such a way that the combinatorial nature of the problem is circumvented. Here the imperfect self interference cancelation is modeled as a variable power gain from the transmitter to the receiver at all nodes. This simple model gives insight into the behavior of different network topologies when self interference cancelation is employed in network nodes. A similar approach can be used in a straightforward manner to model a wide range of network topologies

with various node capabilities as well, for example, singlepacket transmission, singlepacket reception, and many others. The method can also be used to find the required level of accuracy for the self interference cancelation such that certain gains are achieved at the network layer. In addition, it provides a simple mechanism to evaluate the impact of scaling the distance between network nodes on the accuracy level of the self interference cancelation. Thus, from a network design perspective, the method presented here can be very useful.

Recall that WSRMax problem is NP-hard and one has to rely on exponentially complex global optimization techniques [5, 16, 52] to obtain the optimal solution. Nevertheless, the numerical results show that the algorithms presented in this section perform close to global optimization methods. We further test the algorithms by carrying them out on large-scale problems, where global optimization methods [28, 97, 129, 135, 139] cannot be used, due to prohibitive computational complexity. Results show that the given algorithms can provide significant gains at the network layer, in terms of end-to-end rates and network congestion, by exploiting efficiently the available multichannel diversity. We also evaluate the potential gains achievable at the network layer when the network nodes employ self interference cancelation techniques with different degrees of accuracy.

Finally, we consider different receiver capabilities and examine the effect of the use of multiuser detectors.

3.1 System Model and Problem Formulation

3.1.1 Network Model

The wireless network consists of a collection of nodes that can send, receive and relay data across wireless links. The set of all nodes is denoted by $\mathcal{N}$ and we label the nodes with the integer values $n = 1, \ldots, N$. A wireless link is represented as an ordered pair (i, j) of distinct nodes. The set of links is denoted by $\mathcal{L}$ and we label the links with the integer values $l = 1, \ldots, L$. We define $\text{tran}(l)$ as the transmitter node of link l, and $\text{rec}(l)$ as the receiver node of link l. The existence of a link $l \in \mathcal{L}$ implies that direct transmission is possible from $\text{tran}(l)$ to $\text{rec}(l)$. We assume that each node can be equipped with

multiple transceivers, that is, any node can simultaneously transmit to, or receive from, multiple nodes. We define $\mathcal{O}(n)$ as the set of links that are outgoing from node n, and $\mathcal{I}(n)$ as the set of links that are incoming to node n. Furthermore, we denote the set of transmitter nodes by $\mathcal{T}$ and the set of receiver nodes by $\mathcal{R}$, i.e., $\mathcal{T} = \{n \in \mathcal{N} | \mathcal{O}(n) \neq \emptyset\}$ and $\mathcal{R} = \{n \in \mathcal{N} | \mathcal{I}(n) \neq \emptyset\}$.

The network is assumed to operate in slotted time with slots normalized to integer values $t \in \{1, 2, 3, \ldots\}$. All wireless links are sharing a set $\mathcal{C}$ of orthogonal channels, labeled with integers $c = 1, \ldots, C$. When there are many channels which fade independently, at any one time there is a high probability that one of the channels will be strong. Thus, the main motivation for considering multiple channels is exploitation of the diversity that results from unequal links' behavior across a given wide band.

Let $h_{ijc}(t)$ denote the channel gain from the transmitter of link i to the receiver of link j in channel c during time slot t. We assume that $h_{ijc}(t)$ are constant for the duration of a time slot and are independent and identically distributed over the time slots, links as well as over the channels. Let $g_{iic}(t)$ represent the power gain of link i in channel c during time slot t, i.e., $g_{iic}(t) = |h_{iic}(t)|^2$ (see Figure 3.1). For any pair of *distinct* links $i \neq j$, we denote the interference coefficient from link i to link j in channel c by $g_{ijc}(t)$. In the case of nonadjacent links (i.e., links i and j do not have common nodes), g_{ijc} represents the power of the interference signal at the receiver node of link j in channel c when one unit of power is allocated to the transmitter node of link i in channel c, i.e., $g_{ijc} = |h_{ijc}|^2$. When links i and j are adjacent, the value of g_{ijc} represents the power gain in channel c within the same node from its transmitter to its receiver, and is referred to as the

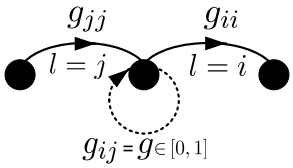


Fig. 3.1 Choosing the value of interference coefficients, i.e., g_{ij} for $i \neq j$ and link power gains, i.e., g_{ii} and g_{jj} (channel index c and time index t are omitted for clarity): $\mathcal{A} = \{(i, j)\}$, $g_{ij} = g$, $g_{ji} = |h_{ji}|^2$, $g_{ii} = |h_{ii}|^2$, and $g_{jj} = |h_{jj}|^2$.

self-interference coefficient (see Figure 3.1). For notational convenience let $\mathcal{A}$ denote the set of all link pairs (i,j) such that links i and j are adjacent. In other words, $\mathcal{A}$ represents the set of all link pairs (i,j) for which the transmitter of link i and the receiver of link j coincide, i.e., $\mathcal{A} = \{(i,j)_{i,j\in\mathcal{L}} |\ \text{tran}(i) = \text{rec}(j)\}$ (see Figure 3.1). Specifically, for all $(i,j) \in \mathcal{A}$, we set $g_{ijc}(t) = g$ to model the residual self-interference gains after a certain self interference cancelation technique was employed at the network's nodes in channel c, where $g \in [0,1]$ is a scalar. We refer to g as the self-interference gain (see Figure 3.1). A value $g = 1$ means that no self interference cancelation technique is used and models the very large self interference that would affect the incoming links of a node if it simultaneously transmitted and received in the same channel. On the other hand, a value $g = 0$ corresponds to a perfect self interference cancelation. Note that, according to relative distances between network's nodes, $g_{ijc}(t)$ for all $(i,j) \in \mathcal{A}$ (i.e., the self-interference coefficients) can be several orders of magnitude larger than $g_{ijc}(t)$ for all $(i,j) \notin \mathcal{A}$. The particular class of network topologies, for which $\mathcal{A} = \emptyset$ (i.e., $\mathcal{T} \cap \mathcal{R} = \emptyset$) is referred to as *bipartite* networks. On the other hand, the class of network topologies, for which $\mathcal{A} \neq \emptyset$ (i.e., $\mathcal{T} \cap \mathcal{R} \neq \emptyset$) is referred to as *nonbipartite* networks. Note that all multihop networks are necessarily nonbipartite.

In every time slot a network controller decides the power and rates allocated to each link in every channel. We denote the power allocated to each link l in channel c during time slot t by $p_{lc}(t)$. The power allocation is subject to a maximum power constraint $\sum_{c\in\mathcal{C}} \sum_{l\in\mathcal{O}(n)} p_{lc}(t) \leq p_n^{\max}$ for each node n.

We consider first the case where all receivers perform *singleuser detection*, i.e., any receiver decodes each of its intended signals by treating all other interfering signals as noise. Extensions to more advanced *multiuser detection* techniques will be addressed in Section 3.3. Suppose that the achievable rate of link l during time slot t is given by

$$r_l(t) = \sum_{c=1}^{C} W_c \log\left(1 + \frac{g_{llc}(t)p_{lc}(t)}{N_l W_c + \sum_{j\neq l} g_{jlc}(t)p_{jc}(t)}\right), \tag{3.1}$$

where W_c represents the bandwidth of channel c and N_l is the power spectral density of the noise at the receiver of link l. Note that for

any link l, interference at $\mathrm{rec}(l)$ $\left[\text{i.e., the term } \sum_{j\neq l} g_{jlc}(t)p_{jc}(t)\right]$ is created by self transmissions $\left[\text{i.e., } \sum_{j\in\mathcal{O}(\mathrm{rec}(l))} g_{jlc}(t)p_{jc}(t)\right]$, as well as by the other node transmissions $\left[\text{i.e., } \sum_{j\in\mathcal{L}\setminus\{\mathcal{O}(\mathrm{rec}(l))\cup\{l\}\}} g_{jlc}(t)p_{jc}(t)\right]$. To simplify the presentation, we assume in the sequel that all channels have equal bandwidths and that the noise power density is the same at all receivers (i.e., $W_c = W$ for all $c \in \mathcal{C}$ and $N_l = N_0$ for all $l \in \mathcal{L}$). The extension to the case of unequal bandwidths W_c and noise power spectral densities N_l is straightforward. Let $\sigma^2 = N_0 W$ denote the noise power, which is constant for all receivers in all channels. Furthermore, we denote by $\mathbf{P}(t) \in \mathbb{R}_+^{L\times C}$ the overall power allocation matrix, i.e., $p_{lc}(t) = [\mathbf{P}(t)]_{l,c}$. The use of the Shannon formula for the achievable rate in (3.1) is approximate in the case of finite length packets and is used to avoid the complexity of rate-power dependence in practical modulation and coding schemes. This is common practice but it must be noted that this is not strictly correct. However, as the packet length increases it is asymptotically correct.

3.1.2 Network Utility Maximization

Exogenous data arrive at the source nodes and they are delivered to the destination nodes over several, possibly multihop, paths. We identify the data by their destinations, i.e., all data with the same destination are considered as a single commodity, regardless of their source. Actually, formulation considered also permits the anycast case, in which any data exits the network as soon as any of particular destination set of nodes receives the data successfully. We label the commodities with integers $s = 1, \ldots, S$ $(S \leq N)$ and the destination node of commodity s is denoted by d_s. For every node, we define $\mathcal{S}_n \subseteq \{1, \ldots, S\}$ as the set of commodities that can arrive exogenously at node n.

A network utility maximization, or NUM, framework similar to the one given in [43, sec. 5.1] is considered. Specifically, exogenously arriving data is not directly admitted to the network layer. Instead, the exogenous data is first placed in the transport layer storage reservoirs. To avoid complications that may arise that are extraneous to the main focus, we assume that all commodities have infinite demand at the transport layer. Nevertheless, the resource allocation (or RA)

algorithms given in this section are still applicable when this assumption is relaxed. At each source node, a set of flow controllers decides the amount of each commodity data admitted during every time slot in the network. Let $x_n^s(t)$ denote the amount of data of commodity s admitted in the network at node n during time slot t. At the network layer, each node maintains a set of S internal queues for storing the current backlog (or unfinished work) of each commodity. Let $q_n^s(t)$ denote the current backlog of commodity s data stored at node n. We formally let $q_{d_s}^s(t) = 0$, i.e., it is assumed that data, which is successfully delivered to its destination, exits the network layer. Associated with each node-commodity pair $(n, s)_{s \in \mathcal{S}_n}$ we define a concave and nondecreasing *utility function* $u_n^s(y)$, representing the "reward" received by sending data of commodity s from node n to node d_s at a long term average rate of y [bits/slot].

The NUM problem under stability constraints is [43, sec. 5]

$$\begin{array}{ll} \text{maximize} & \sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}_n} u_n^s(y_n^s) \\ \text{subject to} & (y_n^s)_{n \in \mathcal{N}, s \in \mathcal{S}_n} \in \mathbf{\Lambda}, \end{array} \tag{3.2}$$

where the variable is $(y_n^s)_{n \in \mathcal{N},\ s \in \mathcal{S}_n}$ and $\mathbf{\Lambda}$ represents the *network layer capacity region*. In particular, the network layer capacity region $\mathbf{\Lambda}$ is the closure of the set of all admissible arrival rate vectors that can be stably supported by the network, considering all possible strategies for choosing the control variables that affect routing, scheduling, and resource allocation (including those with perfect knowledge of future events) [43, p. 28].

A dynamic cross-layer control algorithm, which achieves a utility that is arbitrarily close to the optimal value of problem (3.2), has been introduced in [43, sec. 5]. Specifically, the algorithm's performance can be characterized as follows:

$$\sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}_n} u_n^s(y^{\star s}_n) - \liminf_{T \to \infty} \sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}_n} u_n^s\left(\frac{1}{T} \sum_{t=1:T} \mathrm{E}\{x_n^s(t)\}\right) \leq \frac{B}{V}, \tag{3.3}$$

where $(y^{\star s}_n)_{n \in \mathcal{N}, s \in \mathcal{S}_n}$ is the optimal solution of problem (3.2), $B > 0$ is a well defined constant, and $V > 0$ is an algorithm parameter that can be used to control the tightness of the achieved utility to the optimal value [43, sec. 5.2.1]. The details are extraneous to the central objective

of this section. Particularized to the considered network model, in every time slot t, the algorithm performs the following steps:

Algorithm 3.1 Dynamic cross-layer control algorithm [43, sec. 5.2].

1. Flow control; each node $n \in \mathcal{N}$ solves the following problem:

$$\begin{array}{ll} \text{maximize} & \sum_{s\in\mathcal{S}_n} V u_n^s(x_n^s) - x_n^s q_n^s(t) \\ \text{subject to} & \sum_{s\in\mathcal{S}_n} x_n^s \le R_n^{\max},\ x_n^s \ge 0, \end{array} \tag{3.4}$$

   where the variable is $(x_n^s)_{s\in\mathcal{S}_n}$. Set $(x_n^s(t) = x_n^s)_{s\in\mathcal{S}_n}$. The parameter $V > 0$ is a chosen parameter that affects the algorithm performance [see (3.3)] and $R_n^{\max} > 0$ is used to control the burstiness of data delivered to the network layer.
2. Routing and in-node scheduling; for each link l, let

$$\begin{aligned} \beta_l(t) &= \max_s \left\{ q^s_{\text{tran}(l)}(t) - q^s_{\text{rec}(l)}(t), 0 \right\} \\ c_l^\star(t) &= \arg\max_s \left\{ q^s_{\text{tran}(l)}(t) - q^s_{\text{rec}(l)}(t), 0 \right\}. \end{aligned} \tag{3.5}$$

   If $\beta_l(t) > 0$, the commodity that maximizes the differential backlog, i.e., $c_l^\star(t)$, is selected for potential routing over link l. This is the well known rule of next-hop transmission under the backpressure algorithm [120].
3. Resource allocation; the power allocation $\mathbf{P}(t)$ is given by $\mathbf{P}$ whose entries p_{lc} solve the following problem

$$\begin{array}{ll} \text{maximize} & \sum\limits_{l\in\mathcal{L}} \beta_l(t) \sum\limits_{c\in\mathcal{C}} \log\left(1 + \dfrac{g_{llc}(t)p_{lc}}{\sigma^2 + \sum\limits_{j\neq l} g_{jlc}(t)p_{jc}}\right) \\ \text{subject to} & \sum\limits_{c\in\mathcal{C}} \sum\limits_{l\in\mathcal{O}(n)} p_{lc} \le p_n^{\max},\ n \in \mathcal{N} \\ & p_{lc} \ge 0,\ l \in \mathcal{L},\ c \in \mathcal{C}. \end{array} \tag{3.6}$$

   Once the optimal power allocation $\mathbf{P}(t)$ is determined, compute rate allocation $r_l(t)$ for all $l \in \mathcal{L}$ by using (3.1). The resulting rate $r_l(t)$ is offered to the data of commodity $c_l^\star(t)$.

In the first step, each node n determines the amount of data of commodity s $\big($i.e., $x_n^s(t)$ for all $s \in \mathcal{S}_n\big)$ that are admitted in the network,

based on the current backlogs $\big($i.e., $q_n^s(t)$ for all $s \in \mathcal{S}_n\big)$. In the second step, each node n computes β_l and the corresponding commodity $c_l^\star(t)$ for all $l \in \mathcal{O}(n)$. The commodity $c_l^\star(t)$ is selected for potential routing over link l during time slot t. Recall that *in-node scheduling* refers to selecting the appropriate commodity and it is not to be confused with the links scheduling mechanism, which is handled by the RA subproblem, i.e., step 3. The third step is the most difficult part of Algorithm 3.1, which computes the power allocation $\mathbf{P}(t)$ in each link l. Of course, the RA subproblem maximizes the sum of weighted rates, i.e., WSRMax. The solution $\mathbf{P}(t)$ determines implicitly the links/channels that should be activated in every time slot t. The power allocation $\mathbf{P}(t)$ is used to determine $r_l(t)$ [see (3.1)] and the resulting link rate $r_l(t)$ is offered to the data of commodity $c_l^\star(t)$. Since thw main focus resides in problem (3.6), extensive explanations of Algorithm 3.1 are avoided. However, we refer the reader to [43, sec. 5] for more details.

3.2 Algorithm Derivation: CGP and Homotopy Methods

In this section we focus on resource allocation problem (3.6). By using standard reformulation techniques, we first show that problem (3.6) is equivalent to a CGP [6]. Then we obtain a successive approximation algorithm for problem (3.6) in *bipartite* networks. Next we explain the challenges of the problem in nonbipartite networks (e.g., multihop networks), due to the *self-interference problem*; when a node simultaneously transmits and receives in the same channel, its incoming links are affected by very large self interference levels. Finally, we present a method based on *homotopy methods* [4], together with CGP, which circumvents the aforementioned difficulties.

3.2.1 CGP for WSRMax

Let us denote the negative of the objective function of problem (3.6) by $f_0\big(\mathbf{P}\big)$. It can be expressed as

$$f_0\big(\mathbf{P}\big) = -\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}} \log\left(1+\frac{g_{llc}p_{lc}}{\sigma^2+\sum_{j\neq l} g_{jlc}p_{jc}}\right)^{\beta_l} \tag{3.7}$$

$$= \log\prod_{l\in\mathcal{L}}\prod_{c\in\mathcal{C}}\big(1+\gamma_{lc}\big)^{-\beta_l}, \tag{3.8}$$

where the time index t was dropped for the sake of notational simplicity, and γ_{lc} represents the SINR of link l in channel c, i.e.,

$$\gamma_{lc} = \frac{g_{llc}p_{lc}}{\sigma^2 + \sum_{j\neq l} g_{jlc}p_{jc}},\ l \in \mathcal{L},\ c \in \mathcal{C}. \tag{3.9}$$

Since $\log(\cdot)$ is an increasing function, problem (3.6) can be reformulated equivalently as

$$\begin{aligned} \text{minimize} \quad & \textstyle\prod_{c\in\mathcal{C}} \prod_{l\in\mathcal{L}} \left(1+\gamma_{lc}\right)^{-\beta_l} \\ \text{subject to} \quad & \gamma_{lc} = \frac{g_{llc}p_{lc}}{\sigma^2 + \sum_{j\neq l} g_{jlc}p_{jc}},\ l \in \mathcal{L},\ c \in \mathcal{C} \\ & \textstyle\sum_{c\in\mathcal{C}} \sum_{l\in\mathcal{O}(n)} p_{lc} \leq p_n^{\max},\ n \in \mathcal{N} \\ & p_{lc} \geq 0,\ l \in \mathcal{L},\ c \in \mathcal{C}, \end{aligned} \tag{3.10}$$

where the variables are $(p_{lc})_{l\in\mathcal{L},c\in\mathcal{C}}$ and $(\gamma_{lc})_{l\in\mathcal{L},c\in\mathcal{C}}$. Now we consider the related problem

$$\begin{aligned} \text{minimize} \quad & \textstyle\prod_{c\in\mathcal{C}} \prod_{l\in\mathcal{L}} \left(1+\gamma_{lc}\right)^{-\beta_l} \\ \text{subject to} \quad & \gamma_{lc} \leq \frac{g_{llc}p_{lc}}{\sigma^2 + \sum_{j\neq l} g_{jlc}p_{jc}},\ l \in \mathcal{L},\ c \in \mathcal{C} \\ & \textstyle\sum_{c\in\mathcal{C}} \sum_{l\in\mathcal{O}(n)} p_{lc} \leq p_n^{\max},\ n \in \mathcal{N} \\ & p_{lc} \geq 0,\ l \in \mathcal{L},\ c \in \mathcal{C}, \end{aligned} \tag{3.11}$$

with the same variables $(p_{lc})_{l\in\mathcal{L},c\in\mathcal{C}}$ and $(\gamma_{lc})_{l\in\mathcal{L},c\in\mathcal{C}}$. Note that the equality constraints of problem (3.10) have been replaced with inequality constraints. We refer to these inequality constraints as SINR constraints for simplicity. Since the objective function of problem (3.11) is decreasing in each γ_{lc}, we can guarantee that at any optimal solution of problem (3.11), the SINR constraints must be active.[1] Therefore we solve problem (3.11) instead of problem (3.10).

[1] Otherwise, we can show by contradiction that the current solution is not optimal.

Finally, by introducing the auxiliary variables $v_{lc} \leq 1 + \gamma_{lc}$ and rearranging the terms, problem (3.6) can be further reformulated as

$$\begin{aligned} \text{minimize} \quad & \textstyle\prod_{c\in\mathcal{C}} \prod_{l\in\mathcal{L}} v_{lc}^{-\beta_l} \\ \text{subject to} \quad & v_{lc} \leq 1 + \gamma_{lc},\ l \in \mathcal{L},\ c \in \mathcal{C} \\ & \sigma^2 g_{llc}^{-1} p_{lc}^{-1} \gamma_{lc} + \textstyle\sum_{j\neq l} g_{llc}^{-1} g_{jlc} p_{jc} p_{lc}^{-1} \gamma_{lc} \leq 1,\ l \in \mathcal{L},\ c \in \mathcal{C} \\ & \textstyle\sum_{c\in\mathcal{C}} \sum_{l\in\mathcal{O}(n)} \left(p_n^{\max}\right)^{-1} p_{lc} \leq 1,\ n \in \mathcal{N} \\ & p_{lc} \geq 0,\ l \in \mathcal{L},\ c \in \mathcal{C}, \end{aligned} \tag{3.12}$$

where the variables are $(p_{lc})_{l\in\mathcal{L},c\in\mathcal{C}}$, $(\gamma_{lc})_{l\in\mathcal{L},c\in\mathcal{C}}$, and $(v_{lc})_{l\in\mathcal{L},c\in\mathcal{C}}$. Problem (3.12) is a CGP.

3.2.2 Successive Approximation Algorithm for WSRMax in Bipartite Networks

In this section we consider the case of bipartite networks. Recall from Section 3.1.1 that for such networks we have $\mathcal{A} = \emptyset$. By inspecting problem (3.12), we notice the following: (1) the objective is a monomial function [20, sec. 2.1], (2) the right-hand side (RHS) terms of the first inequality constraints (i.e., $1 + \gamma_{lc}$) are posynomial functions, and (3) the left-hand side terms of all the inequality constraints are either monomial or posynomial functions. Note that if the RHS terms of the first inequality constraints were monomial functions (instead of posynomial ones), problem (3.12) would become a geometric program (or GP) in standard form. GPs can be reformulated as convex problems and they can be solved very efficiently, even for large scale problems [20, sec. 2.5]. These observations suggest that by starting from an initial point, one can search for a close local optimum by solving a sequence of GPs, which locally approximate the original problem (3.12). At each step, the GP is obtained by replacing the posynomial functions in the RHS of the first inequality constraints with their best local monomial approximations near the the solution obtained at the previous step. The solution methods achieved by monomial approximations [6, 20] can be considered as a subset of a broader class of mathematical optimization problems, known in mathematical literature as *inner approximation algorithms for nonconvex problems* [77]. The monomial approximation

for the RHS terms of the first inequality constraints in problem (3.12) is described in the following lemma.

Lemma 3.1. Consider the function $s(\gamma) = 1 + \gamma$, $\gamma > 0$. Let $m(\gamma) = k\gamma^a$, $\gamma > 0$, be a monomial function used to approximate $s(\gamma)$ near an arbitrary point $\hat{\gamma} > 0$. Then,

1. the parameters a and k of the best monomial local approximation are given by
$$a = \hat{\gamma}(1 + \hat{\gamma})^{-1}, \quad k = \hat{\gamma}^{-a}(1 + \hat{\gamma}). \tag{3.13}$$
2. $s(\gamma) \geq m(\gamma)$ for all $\gamma > 0$.

Proof. To show the first part we note the following: the monomial function m is the best local approximation of s near the point $\hat{\gamma}$ if

$$m(\hat{\gamma}) = s(\hat{\gamma}), \quad m'(\hat{\gamma}) = s'(\hat{\gamma}). \tag{3.14}$$

By replacing the expressions of m and s in (3.14) we obtain the following system of equations:

$$\begin{cases} k\hat{\gamma}^a = 1 + \hat{\gamma} \\ ka\hat{\gamma}^{a-1} = 1, \end{cases} \tag{3.15}$$

which has the solution given by (3.13).

The second part follows from (3.14) and noting that $s(\gamma)$ is affine and $m(\gamma)$ is concave on $\mathbb{R}_+$; concavity of $m(\gamma)$ follows from the fact that $k > 0$ and $0 < a < 1$ [22, sec. 3.1.5]. □

The approximation given in Lemma 3.1 turns out to be equivalent to the lower bound approximation used in [94, sec. III-B] for dynamic spectrum management in digital subscriber lines.

Let us now turn to the GP obtained by using the local approximation given by Lemma 3.1. The posynomial functions $1 + \gamma_{lc}$ of the first inequality constraints of problem (3.12) are approximated near the point $\hat{\gamma}_{lc}$. Consequently the approximate inequality constraints become

$$v_{lc} \leq k_{lc}\gamma_{lc}^{a_{lc}}, \quad l \in \mathcal{L},\ c \in \mathcal{C}, \tag{3.16}$$

where a_{lc} and k_{lc} have the forms given in (3.13). Since the objective function of problem (3.12) is a decreasing function of $v_{lc},\ l \in \mathcal{L}, c \in \mathcal{C}$, it can be easily verified that all of these modified inequality constraints will be active at the solution of the GP. Therefore, we can eliminate the auxiliary variables v_{lc} and rewrite the objective function of problem (3.12) as

$$\prod_{l\in\mathcal{L}}\prod_{c\in\mathcal{C}} v_{lc}^{-\beta_l} = \prod_{l\in\mathcal{L}}\prod_{c\in\mathcal{C}} k_{lc}^{-\beta_l}\gamma_{lc}^{-\beta_l a_{lc}} = K \prod_{l\in\mathcal{L}}\prod_{c\in\mathcal{C}} \gamma_{lc}{}^{-\beta_l \frac{\hat{\gamma}_{lc}}{1+\hat{\gamma}_{lc}}}, \tag{3.17}$$

where K is a multiplicative constant which does not affect the problem solution.

In the following subsections, we examine computationally efficient algorithms to obtain a suboptimal solution for problem (3.12). For notational convenience it is useful to define the overall SINR matrices $\boldsymbol{\gamma}, \hat{\boldsymbol{\gamma}} \in \mathbb{R}_+^{L\times C}$ as $[\boldsymbol{\gamma}]_{l,c} = \gamma_{lc}$ and $[\hat{\boldsymbol{\gamma}}]_{l,c} = \hat{\gamma}_{lc}$, respectively.

A very brief outline of the successive approximation algorithm is as follows. It solves an approximated version of problem (3.12) in every iteration and the algorithm consists of repeating this step until convergence.

The first step initializes the algorithm and an initial feasible SINR guess $\hat{\boldsymbol{\gamma}}^{(i)}$ is computed. For bipartite networks, there is no *self-interference problem*, and a simple uniform power allocation can be used.

The second step solves an equivalent GP approximation of problem (3.12) around the current guess $\hat{\boldsymbol{\gamma}}^{(i)}$ [see problem (3.18)]. Note that the auxiliary variables $(v_{lc})_{l\in\mathcal{L},c\in\mathcal{C}}$ of problem (3.12) are eliminated and the objective function of problem (3.12) is replaced by using the monomial approximation at $\hat{\boldsymbol{\gamma}}^{(i)}$ given in (3.17); $K^{(i)}$ is a multiplicative constant which does not influence the solution of problem (3.18). These monomial approximations are sufficiently accurate only in the closer vicinity of the current guess $\hat{\boldsymbol{\gamma}}^{(i)}$. Therefore, the first set of inequality constraints are added to confine the domain of variables $\boldsymbol{\gamma}$ to a region around the current guess $\hat{\boldsymbol{\gamma}}^{(i)}$ [18]. The first set of inequality constraints of problem (3.18) are sometimes called trust region constraints [18, 20], which are not originally introduced in [6]. Therefore, Algorithm 3.2 is a slightly modified version of the solution method proposed in [6]. The

Algorithm 3.2 Successive approximation algorithm for WSRMax

1. Initialization; given tolerance $\varepsilon > 0$, a feasible power allocation $\mathbf{P}_0$. Set $i = 1$. The initial SINR guess $\hat{\boldsymbol{\gamma}}^{(i)}$ is given by (3.9).
2. Solving the GP:

$$\begin{aligned} \text{minimize} \quad & K^{(i)} \prod_{l\in\mathcal{L}} \prod_{c\in\mathcal{C}} \gamma_{lc}^{-\beta_l \frac{\hat{\gamma}_{lc}^{(i)}}{1+\hat{\gamma}_{lc}^{(i)}}} \\ \text{subject to} \quad & \alpha^{-1}\hat{\gamma}_{lc}^{(i)} \le \gamma_{lc} \le \alpha\hat{\gamma}_{lc}^{(i)},\ l \in \mathcal{L},\ c \in \mathcal{C} \\ & \sigma^2 g_{llc}^{-1} p_{lc}^{-1}\gamma_{lc} + \sum_{j\neq l} g_{llc}^{-1} g_{jlc} p_{jc} p_{lc}^{-1}\gamma_{lc} \le 1,\ l \in \mathcal{L},\ c \in \mathcal{C} \\ & \sum_{c\in\mathcal{C}} \sum_{l\in\mathcal{O}(n)} \left(p_n^{\max}\right)^{-1} p_{lc} \le 1,\ n \in \mathcal{N}, \end{aligned} \tag{3.18}$$

   with the variables $(p_{lc})_{l\in\mathcal{L},c\in\mathcal{C}}$ and $(\gamma_{lc})_{l\in\mathcal{L},c\in\mathcal{C}}$. Denote the solution by
   $(p_{lc}^\star)_{l\in\mathcal{L},c\in\mathcal{C}}$ and $(\gamma_{lc}^\star)_{l\in\mathcal{L},c\in\mathcal{C}}$.
3. Stopping criterion; if $\max_{(l,c)\in\mathcal{L}\times\mathcal{C}} \left|\gamma_{lc}^\star - \hat{\gamma}_{lc}^{(i)}\right| \le \varepsilon$ STOP; otherwise go to step 4.
4. Set $i = i + 1$, $\left(\hat{\gamma}_{lc}^{(i)} = \gamma_{lc}^\star\right)_{l\in\mathcal{L},c\in\mathcal{C}}$ and go to step 2.

parameter $\alpha > 1$ controls the desired approximation accuracy. However, it also influences the convergence speed of Algorithm 3.2. At every step, each entry of the current SINR guess $\hat{\boldsymbol{\gamma}}^{(i)}$ can be increased or decreased at most by a factor α. Thus, a value of α close to 1 provides good accuracy for the monomial approximations at the cost of slower convergence speed, while a value much larger than 1 improves the convergence speed at the cost of reduced accuracy. In most practical cases, a fixed value $\alpha = 1.1$ offers a good speed/accuracy tradeoff [20]. Though we have trust region constraints for problem (3.18), it is not mandatary to include those here and Algorithm 3.2 can still be carried out.

The third step checks whether the SINRs $(\gamma_{lc}^\star)_{l\in\mathcal{L},c\in\mathcal{C}}$ obtained from the solution of problem (3.18) have been significantly changed compared to the entries of the current guess $\hat{\boldsymbol{\gamma}}^{(i)}$. If there are no substantial changes, then the algorithm terminates and the link

rate $r_l(t) = \sum_{c=1}^{C} W \log(1 + \gamma_{lc}^{\star})$ is offered to the data of commodity $c_l^{\star}(t)$ [given by (3.5)]. Otherwise, the solution $(\gamma_{lc}^{\star})_{l\in\mathcal{L},c\in\mathcal{C}}$ is taken as the current guess and the algorithm repeats steps 2 to 4 until convergence.

Note that the auxiliary variables $(v_{lc})_{l\in\mathcal{L},c\in\mathcal{C}}$ were only used to reformulate problem (3.11) as a CGP [6] [i.e., problem (3.12)], but they do not appear in Algorithm 3.2. In fact, an identical algorithm results if, at each step, the objective function of problem (3.11) is locally approximated by a monomial function (see [34, lem. 4.2.2]). This alternative derivation, known in optimization literature as *signomial programming* [20]. Careful comparisons reveal that the algorithm recently proposed in [78, p. 3034] is almost identical to Algorithm 3.2 for single channel case with no trust region constraints, i.e., $C = 1$ and $\alpha = \infty$.

The convergence of the Algorithm 3.2 to a Kuhn–Tucker solution of the original nonconvex problem (3.12) is guaranteed [77, thm. 1], since the algorithm falls into the broader class of mathematical optimization problems, *inner approximation algorithms for nonconvex problems* [77].

One interesting and important remark is that the objective function of the approximated problem (3.18) in each iteration i yields a upper bound on the objective function of the original problem (3.11), i.e.,

$$K^{(i)} \prod_{l\in\mathcal{L}} \prod_{c\in\mathcal{C}} \gamma_{lc}^{-\beta_l \frac{\hat{\gamma}_{lc}^{(i)}}{1+\hat{\gamma}_{lc}^{(i)}}} \geq \prod_{l\in\mathcal{L}} \prod_{c\in\mathcal{C}} \left(1 + \gamma_{lc}\right)^{-\beta_l} \tag{3.19}$$

for $(\gamma_{lc} > 0)_{l\in\mathcal{L},\ c\in\mathcal{C}}$, with equality when $\boldsymbol{\gamma} = \hat{\boldsymbol{\gamma}}^{(i)}$. This follows directly from the second statement of Lemma 3.1. By using (3.19), we can show immediately that Algorithm 3.2 is monotonically decreasing; the monotonicity is established by the following theorem.

Theorem 3.2. Let i and $i + 1$ be any consecutive iterations of Algorithm 3.2 and $\hat{\boldsymbol{\gamma}}^{(i)}$ and $\hat{\boldsymbol{\gamma}}^{(i+1)}$ be the SINR guesses at the beginning of each iteration respectively. Then

$$\prod_{l\in\mathcal{L}} \prod_{c\in\mathcal{C}} \left(1 + \hat{\gamma}_{lc}^{(i)}\right)^{-\beta_l} \geq \prod_{l\in\mathcal{L}} \prod_{c\in\mathcal{C}} \left(1 + \hat{\gamma}_{lc}^{(i+1)}\right)^{-\beta_l}. \tag{3.20}$$

Proof. To show this we write the following relations:

$$\prod_{l\in\mathcal{L}}\prod_{c\in\mathcal{C}}\left(1+\hat{\gamma}_{lc}^{(i)}\right)^{-\beta_l} = K^{(i)}\prod_{l\in\mathcal{L}}\prod_{c\in\mathcal{C}}\left(\hat{\gamma}_{lc}^{(i)}\right)^{-\beta_l\frac{\hat{\gamma}_{lc}^{(i)}}{1+\hat{\gamma}_{lc}^{(i)}}} \tag{3.21}$$

$$\geq K^{(i)}\prod_{l\in\mathcal{L}}\prod_{c\in\mathcal{C}}\left(\hat{\gamma}_{lc}^{(i+1)}\right)^{-\beta_l\frac{\hat{\gamma}_{lc}^{(i)}}{1+\hat{\gamma}_{lc}^{(i)}}} \tag{3.22}$$

$$\geq \prod_{l\in\mathcal{L}}\prod_{c\in\mathcal{C}}\left(1+\hat{\gamma}_{lc}^{(i+1)}\right)^{-\beta_l}, \tag{3.23}$$

where (3.21) follows from (3.19), (3.22) follows since $\hat{\boldsymbol{\gamma}}^{(i+1)}$ is the solution of problem (3.18), and (3.23) follows again from (3.19). □

Therefore we see immediately that Algorithm 3.2 always yields a solution, which is at least as good as the one in the previous iteration. This is important in the context of practical implementations, since in practice, one can always stop the algorithm within few iterations, before it terminates.

3.2.3 The Self-interference Problem in Nonbipartite Networks

Let us now consider the nonbipartite networks; networks for which $\mathcal{A}\neq\emptyset$. In other words, the set of nodes cannot be divided into two distinct subsets, $\mathcal{T}$ and $\mathcal{R}$, i.e., $\mathcal{T}\cap\mathcal{R}\neq\emptyset$ (e.g., multihop wireless networks). For example see Figures 3.1 and 3.2. For such network topologies, there is the *self-interference problem* and, consequently, the WSRMax problem must also cope with the self-interference problem. The difficulty comes from that the self-interference gains g^2 can typically be a few orders of magnitude larger than the power gains between distinct network nodes $\{g_{jjc}\}_{j\in\mathcal{L}}$, for example, when there is no self interference cancelation. Therefore there can be a huge imbalance between some entries of $\{g_{ijc}\}_{i,j\in\mathcal{L}}$, especially when g is large. Roughly speaking, this can destroy the smoothness of the functions associated with the WSRMax problem [e.g., the objective function of

[2] Recall that $g_{ijc}=g$ for all $(i,j)\in\mathcal{A}$.

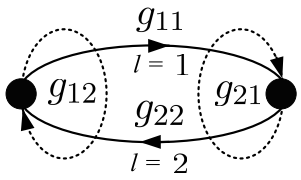


Fig. 3.2 Two node network (channel index c and time index t are omitted for clarity): $\mathcal{A} = \{(1,2),\ (2,1)\}$, $g_{12} = 1$, $g_{21} = 1$, $g_{11} = |h_{11}|^2$, and $g_{22} = |h_{22}|^2$.

problem (3.6)] and can ruin the reliability and the efficiency of Algorithm 3.2 that solves it, at least suboptimally. In other words, there can be many highly suboptimal Kuhn–Tucker solutions for problem (3.12) at which Algorithm 3.2 can terminate by returning an undesirable suboptimal solution. Moreover, the SINR values at the incoming links of a node that simultaneously transmits in the same channel are very *small* and the convergence of Algorithm 3.2 can be very slow if it starts with an initial SINR guess $\hat{\boldsymbol{\gamma}}$ containing entries with nearly zero values. Therefore, the direct application of Algorithm 3.2 can perform very poorly and further improvements are necessary.

A standard way to deal with the self-interference problem consists of adding a supplementary combinatorial constraint into the WSRMax problem that does not allow any node in the network to transmit and receive simultaneously in the same channel [13, 14, 46]. We will refer to a power allocation, which satisfies this constraint as *admissible*. Note that this approach would require solving a power optimization problem (by using Algorithm 3.2) for each possible subset of links that can be simultaneously activated. This results in a combinatorial nature for the WSRMax problem in the case of nonbipartite networks [9, 25, 26, 62, 68, 112, 138]. Of course, since the complexity of this approach grows exponentially with the number of links and number of channels, this solution method becomes quickly impractical.

3.2.4 Successive Approximation Algorithm for WSRMax in Nonbipartite Networks: A Homotopy Method

To avoid difficulties pointed out in Section 3.2.3, we present an algorithm inspired from *homotopy methods* [4] that can be traced back to late 80s; see [63] and the references therein. In fact, the well known

interior-point methods [22, sec. 11],[85] for convex optimization problems also fall into this general class of homotopy methods.

The underlying idea is to first introduce a parameterized problem that approximates the original problem (3.11). Specifically, we construct the parameterized problem from the original problem (3.11) by setting $g_{ijc} = g_v$ for all $(i,j) \in \mathcal{A}$, where $g_v \in \mathbb{R}_+$ is referred to as the homotopy parameter. Indeed, in the original problem (3.11) we have $g_{ijc} = g$ for all $(i,j) \in \mathcal{A}$. Note that the quality of the approximation improves as g_v reaches g; the true self-interference gain. Of course, when g_v is small (e.g., g_v and g_{jjc} are roughly in the same order), Algorithm 3.2 can be used reliably to find a suboptimal solution for the parameterized problem. Thus, a sequence of parameterized problems are solved, starting at a very small g_v and increasing the parameter g_v (thus the accuracy of the approximation) at each step until g_v reaches the true self-interference gain g. Moreover, in each step, when solving the parameterized problem for the current value of g_v, the initial guess for Algorithm 3.2 is obtained by using the solution (power) of the parameterized problem for the previous value of g_v.

The algorithm based on homotopy methods can be summarized as follows:

Algorithm 3.3 Successive approximation algorithm for WSRMax in the presence of self interferers.

1. Initialization; given an initial homotopy parameter $g_0 < g$, $\rho > 1$, a feasible power allocation $\mathbf{P}_0$. Let $g_v = g_0$, $\mathbf{P} = \mathbf{P}_0$.
2. Set $g_{ijc} = g_v$ for all $(i,j) \in \mathcal{A}$. Find the SINR guess $\hat{\boldsymbol{\gamma}}$ by using (3.9).
3. Solving parameterized problem; let $\hat{\boldsymbol{\gamma}}^{(1)} = \hat{\boldsymbol{\gamma}}$, perform steps 2 to 4 of Algorithm 3.2 until convergence to obtain the power $(p^\star_{lc})_{l\in\mathcal{L},c\in\mathcal{C}}$ and SINR values $(\gamma^\star_{lc})_{l\in\mathcal{L},c\in\mathcal{C}}$. Let $(p_{lc} = p^\star_{lc})_{l\in\mathcal{L},c\in\mathcal{C}}$.
4. If $\exists (i,j) \in \mathcal{A}$ and $c \in \mathcal{C}$ such that $p_{ic}p_{jc} > 0$ (i.e., $\mathbf{P}$ is not admissible), then set $g_v = \min\{\rho g_v, g\}$ and go to step 5. Otherwise (i.e., $\mathbf{P}$ is admissible) STOP.
5. If $g_v < g$, go to step 2, otherwise STOP.

The first step initializes the algorithm; the homotopy parameter g_v is initialized by g_0, where g_0 is chosen in the same range of values as the power gains between distinct nodes. Specifically, in the simulations we select $g_0 = \max_{j\in\mathcal{L},c\in\mathcal{C}}\{g_{jjc}\}$. Step 2 updates the problem data for the parameterized problem and a feasible SINR guess is computed. The third step finds a suboptimal solution for the parameterized problem. The algorithm terminates in step 4 if $\mathbf{P}$ is admissible (thus none of nodes in the network are transmitting and receiving simultaneously in the same channel). On the other hand, if $\mathbf{P}$ is not admissible, then the homotopy parameter g_v is increased. If g_v reaches its extreme allowed value (i.e., the actual self-interference gain value of g), the algorithm terminates. Otherwise (i.e., $g_v < g$), it returns to step 2 and continues. Terminating Algorithm 3.3 if the solution is *admissible* is intuitively obvious for the following reason. The data associated with the parameterized problem that is solved in step 3 of Algorithm 3.3 becomes independent of the homotopy parameter g_v, and therefore a further increase in g_v after having an *admissible* solution has no effect on the results. Computational experience suggests that Algorithm 3.3 yields an *admissible* solution way before g_v reaches value g (e.g., in the case of no self interference cancelation, i.e., $g = 1$, an admissible power allocation is achieved in about $1-4$ iterations with $\rho = 2$).

Since Algorithm 3.3 runs a finite number of instances of Algorithm 3.2, its computational complexity *does not increase* more than polynomially with the problem size. Clearly, Algorithm 3.3 can converge to a Kuhn-Tucker solution of the last parameterized problem (the one just before the termination of Algorithm 3.3).

As a specific example to illustrate self interference, consider the simple network shown in Figure 3.2 and suppose that no self interference cancelation technique is employed at the network's nodes, i.e., $g = 1$. Here, $N = 2$, $L = 2$, and $C = 1$. Note that $\mathcal{A} = \{(1,2),(2,1)\}$ and let $\beta_1, \beta_2 \neq 0$. Suppose that $g_{12} \gg g_{22}$ and $g_{21} \gg g_{11}$, which is often the case due to path losses. Since the gains $g_{12} = 1$ and $g_{21} = 1$ are very large compared to g_{22} and g_{11}, for any nonzero power allocation $p_1, p_2 = p_0$ the initial SINR guess $\hat{\gamma}_1, \hat{\gamma}_2$ will have nearly zero values. This results in difficulties in using Algorithm 3.2 directly.

In Algorithm 3.3 this problem is circumvented by initializing the gains g_{12} and g_{21} by a parameter g_0 (e.g., $g_0 = \max\{g_{11}, g_{22}\}$) and executing Algorithm 3.2 repeatedly, increasing incrementally the parameter g_v until it reaches 1, the true values of g_{12} and g_{21}.

Regarding the complexity of the algorithm we make the following remarks. The computational complexity of a GP depends on the number of variables and constraints, as well as on the sparsity pattern of the problem [20]. Unfortunately, it is difficult to quantify precisely the sparsity pattern, and therefore a general complexity analysis is not available. To give a rough idea, let us consider a fully connected network with $N = 9$ nodes and $C = 8$ channels. The number of variables in problem (3.18) is $2LC = 1152$ and the number of constraints is $3LC + N = 1737$, and it was solved in about 12 seconds on desktop computer. The number of iterations depends on the starting point, $p_n^{\max}$ and channel gains g_{ijc}, but typically Algorithm 3.2 required around 100 iterations to converge.

Nevertheless, with some slight modifications it is possible to dramatically decrease the average complexity per iteration, which is very important in the context of practical implementations. Two simple modifications are as follows:

1. Use a large values for the parameter α in Algorithm 3.2: as we discussed in Section 3.2.2, large α can improve the convergence speed of Algorithm 3.2 at the cost of reduced accuracy of the monomial approximation.
2. Eliminate (relatively) insignificant variables; we can eliminate the power variables p_{lc} and the associated SINR variables γ_{lc} from problem (3.18) when they have relatively very small contributions to the overall objective value of (3.18). Specifically, the exponent term $\beta_l \hat{\gamma}_{lc}^{(i)} / \left(1 + \hat{\gamma}_{lc}^{(i)}\right)$ in the objective of (3.18) is evaluated for all $l \in \mathcal{L}, c \in \mathcal{C}$ and if

$$\beta_l \hat{\gamma}_{lc}^{(i)} / \left(1 + \hat{\gamma}_{lc}^{(i)}\right) \ll \max_{\bar{l} \in \mathcal{L}, \bar{c} \in \mathcal{C}} \beta_{\bar{l}} \hat{\gamma}_{\bar{l}\bar{c}}^{(i)} / \left(1 + \hat{\gamma}_{\bar{l}\bar{c}}^{(i)}\right),$$

   then p_{lc}s and the associated γ_{lc}s are eliminated in successive GPs.

3.2.5 Impact of Scaling the Node Distances on the Accuracy of Self Interference Cancelation

Based on a simple exponential path loss model, in this section we discuss the impact of scaling the distance between network nodes on the accuracy level of the self interference cancelation.

For simplicity, we focus on the single-channel case (i.e., $C = 1$). Suppose an exponential path loss model, where the channel power gains $|h_{ij}(t)|^2$, between distinct nodes are given by

$$|h_{ij}(t)|^2 = \left(\frac{d_{ij}}{d_0}\right)^{-\eta} c_{ij}(t). \tag{3.24}$$

Here d_{ij} is the distance from the transmitter of link i to the receiver of link j, d_0 is the *far field reference distance* [64], η is the path loss exponent, and $c_{ij}(t)$ are exponentially distributed random variables with unit mean, independent of the time slots and links. The first term of (3.24) represents the path loss factor and the second term models Rayleigh small-scale fading.

Suppose $p_n^{\max} = p_0^{\max}$ for all $n \in \mathcal{N}$. For all $l \in \mathcal{L}$ we define the signal-to-noise ratio (or SNR) of link l as

$$\mathrm{SNR}_l = \frac{p_0^{\max}}{\sigma^2}\left(\frac{d_{ll}}{d_0}\right)^{-\eta}. \tag{3.25}$$

It represents the average SNR at $\mathrm{rec}(l)$ when $\mathrm{tran}(l)$ allocates all its transmission power to link l and all the other nodes are silent. Let $\mathbf{p}(t) \in \mathbb{R}_+^L$ denote the overall power allocation matrix, i.e., $p_l(t) = [\mathbf{p}(t)]_l$ (note that the channel index is dropped for simplicity, since $C = 1$).

Let us consider a network that is obtained from another one, by scaling the distance between distinct nodes and the maximum node transmission power such that all link's SNRs [see (3.25)] are conserved. In the sequel, we show that, in order to preserve the achievable rate region, the accuracy level of the self interference cancelation techniques must also be scaled appropriately.

We start by defining two matrices, which will be useful for later reference. Let $\mathbf{D} \in \mathbb{R}_+^{L\times L}$ denote the node distance matrix defined as $[\mathbf{D}]_{i,j} = d_{ij}$ and $\mathbf{G}(t) \in \mathbb{R}_+^{L\times L}$ denote the interference coefficient and the power gain matrix during time slot t, defined as $[\mathbf{G}(t)]_{i,j} = g_{ij}(t)$.

The achievable rate region with singleuser detection at receivers for a given $\mathbf{G}(t)$ and a maximum node transmission power $p_0^{\max}$ can be expressed as

$$\mathcal{R}(\mathbf{G}(t), p_0^{\max}) = \left\{ (r_l)_{l\in\mathcal{L}} \middle| \begin{array}{ll} r_l \le \log\left(1 + \dfrac{g_{ll}(t)p_l}{\sigma^2 + \sum_{j\neq l} g_{jl}(t)p_j}\right), & l\in\mathcal{L} \\ \sum_{l\in\mathcal{O}(n)} p_l \le p_0^{\max}, & n\in\mathcal{N} \\ p_l \ge 0, & l\in\mathcal{L} \end{array} \right\}. \tag{3.26}$$

From (3.26), it follows that if the matrix $\mathbf{G}(t)$ is scaled by a factor of $1/\kappa$, and the maximum node transmission power $p_0^{\max}$ is scaled by a factor of κ, then the achievable rate region is unchanged, i.e.,

$$\mathcal{R}(\mathbf{G}(t), p_0^{\max}) = \mathcal{R}(\mathbf{G}(t)/\kappa, \kappa p_0^{\max}). \tag{3.27}$$

Let $\kappa = \theta^\eta$. According to the exponential path loss model given in (3.24), the scaling of $\mathbf{G}(t)$ by a factor of $1/\kappa$ (or $1/\theta^\eta$) is equivalent to the scaling of node distance matrix $\mathbf{D}$ by a factor of θ and the scaling of self-interference gains g by a factor of $1/\theta^\eta$. Therefore, with a slight abuse of notation, we rewrite (3.27) as

$$\mathcal{R}(\mathbf{D}, g, p_0^{\max}) = \mathcal{R}(\theta\mathbf{D}, g/\theta^\eta, \theta^\eta p_0^{\max}). \tag{3.28}$$

To interpret the relation in (3.28), we consider a network characterized by D, g, and $p_0^{\max}$. If we construct another network by scaling D by a factor of θ and by scaling $p_0^{\max}$ by a factor of θ^η, then to preserve the achievable rate region, the accuracy level of the self interference cancelation should be improved to g/θ^η. This is intuitively obvious since, the larger the distance between network nodes, the larger the power levels required to preserve the link SINRs, and therefore the higher the accuracy level required by the self interference cancelation techniques to remove the increased transmit power at nodes. Based on (3.28) we can establish similar equivalences in terms of network layer performance metrics as well. Roughly speaking, relation (3.28) suggests that in networks where the nodes are located far apart (e.g., cellular type of wireless networks), the accuracy of self interference cancelation is more stringent compared to that in networks where the nodes are located in close vicinity (e.g., a wireless network setup in an office).

3.3 Extensions to Wireless Networks with Advanced Transceivers

Until now the receiver structure was basically assumed to be equivalent to a bank of match filters, each of which attempts to decode one of the signals of interest at each node while treating the other signals as noise. This is a suboptimal detector structure that is commonly assumed. In this section, we investigate the possible gains achievable by using more advanced receiver structures. For clarity, we discuss only the single-channel case; see [134, app. B] for the extension to the multichannel case. We assume that at every node $n \in \mathcal{N}$ the transmitter performs superposition coding over its outgoing links $\mathcal{O}(n)$, and the receiver decodes the signals of incoming links $\mathcal{I}(n)$ by using a multiuser receiver based on successive interference cancelation (SIC) strategy. One may of course assume other detector structure, including the optimum one that implements maximum likelihood. The largest set of achievable rates is obtained when the SIC receiver at every node $n \in \mathcal{N}$ is allowed to decode and cancel out the signals of all its incoming links $\mathcal{I}(n)$ and any subset of the remaining links in its complement set $\mathcal{L} \setminus \mathcal{I}(n)$. Let $\mathcal{D}(n)$ denote the set of links, which are decoded at the node n, i.e., $\mathcal{D}(n) = \mathcal{I}(n) \cup \mathcal{U}(n)$ for some $\mathcal{U}(n) \subseteq \mathcal{L} \setminus \mathcal{I}(n)$. Furthermore, let $\mathcal{R}^{\mathrm{SIC}}(\mathcal{D}(1),\ldots,\mathcal{D}(N),p_1^{\max},\ldots,p_N^{\max})$ denote the achievable rate region for given $\mathcal{D}(1),\ldots,\mathcal{D}(N)$ and maximum node transmission power $p_1^{\max},\ldots,p_N^{\max}$. We denote by $\mathcal{R}^{\mathrm{SIC}}(p_1^{\max},\ldots,p_N^{\max})$ the achievable rate region, which is obtained as a union of all $\mathcal{R}^{\mathrm{SIC}}(\mathcal{D}(1),\ldots,\mathcal{D}(N),p_1^{\max},\ldots,p_N^{\max})$ over all possible $2^{\sum_{n\in\mathcal{N}}(L-|\mathcal{I}(n)|)}$ combinations of sets $\mathcal{D}(1),\ldots,\mathcal{D}(N)$, i.e.,

$$\mathcal{R}^{\mathrm{SIC}}(p_1^{\max},\ldots,p_N^{\max}) = \bigcup_{\mathcal{D}(1),\ldots,\mathcal{D}(N)|\forall n\in\mathcal{N}\ \exists\mathcal{U}(n)\subseteq\mathcal{L}\setminus\mathcal{I}(n)\ \text{s.t.}\ \mathcal{D}(n)=\mathcal{I}(n)\cup\mathcal{U}(n)} \mathcal{R}^{\mathrm{SIC}}(\mathcal{D}(1),\ldots,\mathcal{D}(N),p_1^{\max},\ldots,p_N^{\max}). \tag{3.29}$$

The receiver of each node $n \in \mathcal{N}$ is allowed to perform SIC in its own order. Let $\boldsymbol{\pi}_n = \big(\pi_n(1),\ldots,\pi_n(|\mathcal{D}(n)|)\big)$ be an arbitrary permutation of the links in $\mathcal{D}(n)$, which describes the decoding and

cancelation order at node n. Specifically, the signal of link $\pi_n(l)$ is decoded after all codewords of links $\pi_n(j)$, $j < l$ have been decoded and their contribution to the signal received at node n has been canceled. Thus, only the signals of the links $\pi_n(j)$, $j > l$ act as interference. The rate region $\mathcal{R}^{\mathrm{SIC}}(\mathcal{D}(1),\ldots,\mathcal{D}(N),p_1^{\max},\ldots,p_N^{\max})$ is obtained by considering all possible combinations of decoding orders for all nodes, i.e., all possible $\prod_{n\in\mathcal{N}}(|\mathcal{D}(n)|\,!)$ combinations $\boldsymbol{\pi} \stackrel{\Delta}{=} \boldsymbol{\pi}_1 \times \boldsymbol{\pi}_2 \times \cdots \times \boldsymbol{\pi}_N$. Thus, $\mathcal{R}^{\mathrm{SIC}}(\mathcal{D}(1),\ldots,\mathcal{D}(N),p_1^{\max},\ldots,p_N^{\max})$ can be expressed as

$$\mathcal{R}^{\mathrm{SIC}}(\mathcal{D}(1),\ldots,\mathcal{D}(N),p_1^{\max},\ldots,p_N^{\max}) = \bigcup_{\boldsymbol{\pi}} \left\{ (r_1,\ldots,r_L) \left| \begin{array}{l} r_{\pi_n(l)} \le \log\left(1+\dfrac{G_{\pi_n(l)n}(t)\ p_{\pi_n(l)}}{\sigma^2+\sum_{j>l} G_{\pi_n(j)n}(t)\ p_{\pi_n(j)}}\right), \\ \qquad\qquad \forall (n,l) \text{ s.t. } n\in\mathcal{N},\ l\in\{1,\ldots,|\mathcal{D}(n)|\} \\ \sum_{l\in\mathcal{O}(n)} p_l \le p_n^{\max},\ n\in\mathcal{N} \\ p_l \ge 0,\ l\in\mathcal{L} \end{array} \right. \right\}, \tag{3.30}$$

where G_{ln}, $l\in\mathcal{L}$, $n\in\mathcal{N}$ represents the power gain from the transmitter of link l to the receiver at *node* n, and p_l represents the power allocated for the signal of link l. Clearly, the computational complexity experiences a formidable increase. Nevertheless, the RA subproblem at the third step of Dynamic Cross-Layer Control Algorithm 3.1 can be written as[3]

$$\begin{array}{ll} \text{maximize} & \sum_{l\in\mathcal{L}} \beta_l(t) r_l \\ \text{subject to} & (r_l)_{l\in\mathcal{L}} \in \mathcal{R}^{\mathrm{SIC}}(p_1^{\max},\ldots,p_N^{\max}), \end{array} \tag{3.31}$$

where the variable is $(r_l)_{l\in\mathcal{L}}$.

The combinatorial description of $\mathcal{R}^{\mathrm{SIC}}(p_1^{\max},\ldots,p_N^{\max})$ implies that solving problem (3.31) requires optimization over all possible combinations of decoding sets $\mathcal{D}(1),\ldots,\mathcal{D}(N)$ and decoding orders $\boldsymbol{\pi}$. This is intractable, even for off line optimization of moderate size networks.

[3] Note that $\mathcal{R}^{\mathrm{SIC}}(p_1^{\max},\ldots,p_N^{\max})$ represents the set of *directly achievable* rates. By invoking a time sharing argument, one can extend the achievable rate region to the convex hull of $\mathcal{R}^{\mathrm{SIC}}(p_1^{\max},\ldots,p_N^{\max})$. However, this would not affect the optimal value of problem (3.31) because the objective function is linear [69].

Therefore, in the following we present two alternatives to find the solution of a more constrained version of problem (3.31) instead of solving problem (3.31) itself. The first alternative limits the access protocol so that only one node can transmit in all its outgoing links in each time slot. The second alternative adopts a similar view by assuming that only one node can receive from all its incoming links in each time slot. The main advantage of the alternatives above is their simplicity. As a result, a cheaply computable lower bound on the optimal value of problem (3.31) can be obtained. Moreover, these simple access protocols can be useful in practical applications with more advanced communication systems.

3.3.1 Single Node Transmission Case

By imposing the additional constraint that only one node can transmit during each slot, problem (3.31) is reduced to a problem where the optimal power and rate allocation can be computed via convex programming. Specifically, problem (3.31) is reduced to N WSRMax problems for the scalar broadcast channel, one for each possible transmitting node.

For any node $n \in \mathcal{N}$, let $\boldsymbol{\rho}_n = \big(\rho_n(1), \ldots, \rho_n(|\mathcal{O}(n)|)\big)$ be a permutation of the set of outgoing links $\mathcal{O}(n)$ such that

$$g_{\rho_n(1)\rho_n(1)}(t) \leq g_{\rho_n(2)\rho_n(2)}(t) \leq \ldots \leq g_{\rho_n(|\mathcal{O}(n)|)\rho_n(|\mathcal{O}(n)|)}(t),$$

where $g_{ii}(t)$ denotes the power gain from the transmitter of link i to the receiver of link i during time slot t. Now we consider the case where node n is the transmitter. This results in a scalar Gaussian broadcast channel with $|\mathcal{O}(n)|$ users. Thus for all $i \in \{1, \ldots, |\mathcal{O}(n)|\}$, the optimal decoding and cancelation order at the receiver node of links $\rho_n(i)$ is specified by $\boldsymbol{\rho}_n$ [124, sec. 6]. Specifically, the receiver of the link $\rho_n(i)$ decodes its own signal after all the codewords of links $\rho_n(j)$, $j < i$ have been decoded and their contribution to the received signal has been canceled. Thus, only the signals of the links $\rho_n(j)$, $j > i$ act as interference at the receiver of the link $\rho_n(i)$. Now we can rewrite problem (3.31) by using the capacity region descriptions of the scalar Gaussian broadcast

channels [125] as

$$\begin{aligned}
\text{maximize} \quad & \textstyle\sum_{l\in\mathcal{O}(n)} \beta_l r_l \\
\text{subject to} \quad & n \in \mathcal{N} \\
& r_{\rho_n(i)} \le \log\left(1+\frac{g_{\rho_n(i)\rho_n(i)}\ p_{\rho_n(i)}}{\sigma^2+g_{\rho_n(i)\rho_n(i)}\sum_{j=i+1}^{|\mathcal{O}(n)|}\ p_{\rho_n(j)}}\right), \\
& \qquad\qquad i \in \{1,\ldots,|\mathcal{O}(n)|\} \\
& \textstyle\sum_{l\in\mathcal{O}(n)} p_l \le p_n^{\max} \\
& p_l \ge 0,\ l \in \mathcal{O}(n) \\
& p_l = 0,\ l \notin \mathcal{O}(n),
\end{aligned} \tag{3.32}$$

where the variables are n, $(p_l)_{l\in\mathcal{L}}$, and $(r_l)_{l\in\mathcal{L}}$. Note that the time index t is dropped for notational convenience. The solution of problem (3.32) is obtained in two steps. First, we solve N independent subproblems (one subproblem for each possible transmitting node $n \in \mathcal{N}$). Then we select the solution of the subproblem with the largest objective value. The subproblem can be expressed as

$$\begin{aligned}
\text{maximize} \quad & \textstyle\sum_{i=1}^{|\mathcal{O}(n)|} \beta_{\rho_n(i)} r_{\rho_n(i)} \\
\text{subject to} \quad & r_{\rho_n(i)} = \log\left(1+\frac{g_{\rho_n(i)\rho_n(i)}\ p_{\rho_n(i)}}{\sigma^2+g_{\rho_n(i)\rho_n(i)}\sum_{j=i+1}^{|\mathcal{O}(n)|}\ p_{\rho_n(j)}}\right), \\
& \qquad\qquad i \in \{1,\ldots,|\mathcal{O}(n)|\} \\
& \textstyle\sum_{l\in\mathcal{O}(n)} p_l \le p_n^{\max} \\
& p_l \ge 0,\ l \in \mathcal{O}(n),
\end{aligned} \tag{3.33}$$

where the variables are $(p_l)_{l\in\mathcal{O}(n)}$ and $(r_l)_{l\in\mathcal{O}(n)}$. Problem (3.33) represents the WSRMax over the capacity region of a scalar Gaussian broadcast channel [125, sec. 2] with $|\mathcal{O}(n)|$ users. The *barrier method* [22, sec. 11.3.1], or the explicit greedy method proposed in [125, sec. 3.2], can be used to efficiently solve this problem. Here we use the barrier method and refer the reader to Appendix B for more details. Let $g^{(n)}$, $p_l^{(n)}$, and $r_l^{(n)}$ denote the optimal objective value and the corresponding optimal solution (i.e., power and rate) respectively. Then

the rate/power relation can be expressed as

$$r_{\rho_n(i)}^{(n)} = \log\left(1+\frac{g_{\rho_n(i)\rho_n(i)}\ p_{\rho_n(i)}^{(n)}}{\sigma^2+g_{\rho_n(i)\rho_n(i)}\sum_{j=i+1}^{|\mathcal{O}(n)|}p_{\rho_n(j)}^{(n)}}\right),\quad i\in\{1,\ldots,|\mathcal{O}(n)|\} \tag{3.34}$$

and the optimal solution of problem (3.32) is given by

$$\begin{aligned} n^\star &= \arg\max_{n\in\mathcal{N}} g^{(n)}\ ;\\ p_l^\star &= \begin{cases} p_l^{(n^\star)} & l\in\mathcal{O}(n^\star)\\ 0 & \text{otherwise}\ ; \end{cases}\\ r_l^\star &= \begin{cases} r_l^{(n^\star)} & l\in\mathcal{O}(n^\star)\\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{3.35}$$

Next we consider the case where only one node can receive during each slot.

3.3.2 Single Node Reception Case

In this case, the associated problem (3.31) is reduced to a simpler form where the optimal power and rate allocation can be computed very efficiently by considering N WSRMax problems for the Gaussian multiple access channel, one for each possible receiving node.

We start by considering the capacity region descriptions of the Gaussian multiaccess channel with $|\mathcal{I}(n)|$, $n\in\mathcal{N}$ users [126],[124, sec. 6]. For any receiving node $n\in\mathcal{N}$, the capacity region of a the $|\mathcal{I}(n)|$ user Gaussian multiaccess channel, with power constraints $p_l,\ l\in\mathcal{I}(n)$, is given by the set of rate vectors that lie in the intersection of the constraints

$$\sum_{l\in\mathcal{V}(n)} r_l \le \log\left(1+\frac{\sum_{l\in\mathcal{V}(n)} g_{ll}p_l}{\sigma^2}\right) \tag{3.36}$$

for every subset $\mathcal{V}(n)\subseteq\mathcal{I}(n)$. Thus, we can rewrite problem (3.31) as

$$\begin{aligned} \text{maximize}\quad & \textstyle\sum_{l\in\mathcal{I}(n)}\beta_l r_l\\ \text{subject to}\quad & n\in\mathcal{N}\\ & \textstyle\sum_{l\in\mathcal{V}(n)} r_l \le \log\left(1+\frac{\sum_{l\in\mathcal{V}(n)} g_{ll}p_l}{\sigma^2}\right),\ \mathcal{V}(n)\subseteq\mathcal{I}(n)\\ & 0\le p_l\le p_{\text{tran}(l)}^{\max},\ l\in\mathcal{I}(n)\\ & p_l=0,\ l\notin\mathcal{I}(n), \end{aligned} \tag{3.37}$$

where the variables are n, $(p_l)_{l\in\mathcal{L}}$, and $(r_l)_{l\in\mathcal{L}}$. Again, the solution is obtained in two steps. First, we solve N independent subproblems (one subproblem for each possible receiving node $n \in \mathcal{N}$). Then we select the solution of the subproblem with the largest objective value. The subproblem has the form:

$$\begin{aligned} \text{maximize} \quad & \textstyle\sum_{l\in\mathcal{I}(n)} \beta_l r_l \\ \text{subject to} \quad & \textstyle\sum_{l\in\mathcal{V}(n)} r_l \le \log\left(1+\frac{\sum_{l\in\mathcal{V}(n)} g_{ll}p_l}{\sigma^2}\right),\ \mathcal{V}(n)\subseteq\mathcal{I}(n) \\ & 0 \le p_l \le p_{\text{tran}(l)}^{\max},\ l\in\mathcal{I}(n), \end{aligned} \tag{3.38}$$

where the variables are $(p_l)_{l\in\mathcal{I}(n)}$ and $(r_l)_{l\in\mathcal{I}(n)}$. Problem (3.38) is equivalent to the WSRMax over the capacity region of the Gaussian multiaccess channel with $|\mathcal{I}(n)|$ users [124, sec. 6]. The solution is readily obtained by considering the *polymatroid* structure of the capacity region [126, lem. 3.2]. Again we denote by $g^{(n)}$, $p_l^{(n)}$, and $r_l^{(n)}$ the optimal objective value and the optimal solution of problem (3.38) respectively. Thus, the solution of problem (3.38) can be written in closed form as $p_l^{(n)} = p_{\text{tran}(l)}^{\max}$ for all $l \in \mathcal{I}(n)$ and

$$r_{\varrho_n(i)}^{(n)} = \log\left(1+\frac{g_{\varrho_n(i)\varrho_n(i)}\ p_{\varrho_n(i)}^{(n)}}{\sigma^2+\sum_{j=i+1}^{|\mathcal{I}(n)|} g_{\varrho_n(j)\varrho_n(j)}\ p_{\varrho_n(j)}^{(n)}}\right),\quad i\in\{1,\ldots,|\mathcal{I}(n)|\}, \tag{3.39}$$

where $\boldsymbol{\varrho}_n = (\varrho_n(1),\ldots,\varrho_n(|\mathcal{I}(n)|))$ is a permutation of the set of incoming links $\mathcal{I}(n)$ such that

$$\beta_{\varrho_n(1)} \le \beta_{\varrho_n(2)} \le \cdots \le \beta_{\varrho_n(|\mathcal{I}(n)|)}. \tag{3.40}$$

One can in fact identify $\boldsymbol{\varrho}_n$ as the SIC order at the receiving node $n \in \mathcal{N}$. Finally, the optimal solution of problem (3.37) can be expressed as

$$\begin{aligned} n^\star &= \arg\max_{n\in\mathcal{N}} g^{(n)}\ ; \\ p_l^\star &= \begin{cases} p_l^{(n^\star)} & l\in\mathcal{I}(n^\star) \\ 0 & \text{otherwise}\ ; \end{cases} \\ r_l^\star &= \begin{cases} r_l^{(n^\star)} & l\in\mathcal{I}(n^\star) \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{3.41}$$

3.4 Numerical Examples

In this section, we use the algorithms of the preceding sections to identify the solutions to the selected NUM problem and their properties, so as to get insight into network design and provisioning methods. Specifically, in every time slot t, the rate allocation at step 3 of the Dynamic Cross-Layer Control Algorithm (i.e., Algorithm 3.1, sec. 3.1) is obtained by using the algorithms presented for WSRMax described in Sections 3.2 and 3.3.

We assume a block fading Rayleigh channel model where the channel gains are constant during each time slot and change independently from slot-to-slot. The small-scale fading components of the channel gains are assumed to be independent and identically distributed over the time slots, links, and channels. Recall that we consider equal power spectral density for all receivers, i.e., $N_l = N_0$ for all $l \in \mathcal{L}$ and equal channel bandwidths, i.e., $W_c = W$ for all $c \in \mathcal{C}$. Furthermore, the maximum power constraint is assumed to be the same for all nodes, i.e., $p_n^{\max} = p_0^{\max}$ for all $n \in \mathcal{N}$ (independent of the number of channels C). For a fair comparison between cases with different numbers of channels, we have assumed that the total available bandwidth is constant regardless of C, i.e., $\sum_{c=1}^{C} W_c = W_{\text{tot}}$. In all the simulations we have selected the total bandwidth to be normalized to one, i.e., $W_{\text{tot}} = 1$ Hz.

For comparing different algorithms, we consider the following two performance metrics: (1) *the average sum-rate* $\sum_{n\in\mathcal{N}} \sum_{s\in\mathcal{S}_n} \bar{x}_n^s$ and (2) *the average network congestion* $\sum_{n\in\mathcal{N}} \sum_{s=1}^{S} \bar{q}_n^s$. For each network instance, the Dynamic Cross-Layer Control Algorithm (i.e., Algorithm 3.1) is simulated for at least $\acute{T} = 10{,}000$ time slots and the average rates $\bar{x}_n^s$ and queue sizes $\bar{q}_n^s$ are computed by averaging the last $t_0 = 3000$ time slots, i.e., $\bar{x}_n^s = 1/t_0 \sum_{t=\acute{T}-t_0}^{\acute{T}} x_n^s(t)$ and $\bar{q}_n^s = 1/t_0 \sum_{t=\acute{T}-t_0}^{\acute{T}} q_n^s(t)$. We assume that the rates corresponding to all node-commodity pairs $(n,s)_{s\in\mathcal{S}_n}, n \in \mathcal{N}$ are subject to proportional fairness, and therefore we select the utility functions $u_n^s(y) = \log_e(y)$. In all considered setups, we selected $V = 100$ [see (3.4)] and the parameters $R_n^{\max}$ [see (3.4)] were chosen such that all conditions presented in [82, sec. III-D] were satisfied.

We start with a simple network instance (Section 3.4.1); a bipartite network where no self interferers exist (i.e., $\mathcal{A} = \emptyset$) and the successive approximation algorithm, Algorithm 3.2 (Section 3.2.2) is used in resource allocation. The associated results show important consequences on upper layers due to the successive approximation algorithm. We then consider more general networks (Section 3.4.2), with the presence of self interferers (i.e., $\mathcal{A} \neq \emptyset$) and no self interference cancelation at network's nodes (i.e., $g = 1$). Here Algorithm 3.3 (Section 3.2.4) is used in resource allocation. The gains achievable at the network layer, due to different degrees of the self interference cancelation performed at the network nodes, are investigated quantitatively in Section 3.4.3. By changing g in the interval $[0,1]$, the results are able to capture the effect of self interference cancelation performed with different levels of accuracy. Finally, we look at the multiuser receiver scenario, again using the same network instance as in Section 3.4.2. The associated results (Section 3.4.4) show impacts in upper layer performance due to advanced receiver architectures.

3.4.1 NUM for Bipartite Networks with Singleuser Detection at Receivers

A bipartite network, as shown in Figure 3.3, is considered. There are $N = 8$ nodes, $L = 4$ links, and $S = 4$ commodities. One distinct commodity arrives exogenously at every node n from the subset $\{1,2,3,4\} \subseteq \mathcal{N}$. Without loss of generality we assume that the nodes and commodities are labeled such that commodity i arrives at node i for

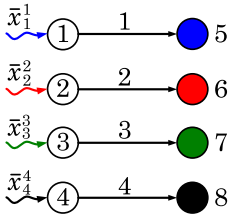


Fig. 3.3 Bipartite wireless network with $N = 8$ nodes, $L = 4$ links, and $S = 4$ commodities.

any $i \in \{1,2,3,4\}$. The destination nodes are specified by the following commodity-destination node pairs $(s,d_s) \in \{(1,5),(2,6),(3,7),(4,8)\}$.

The channel power gains between distinct nodes are given by

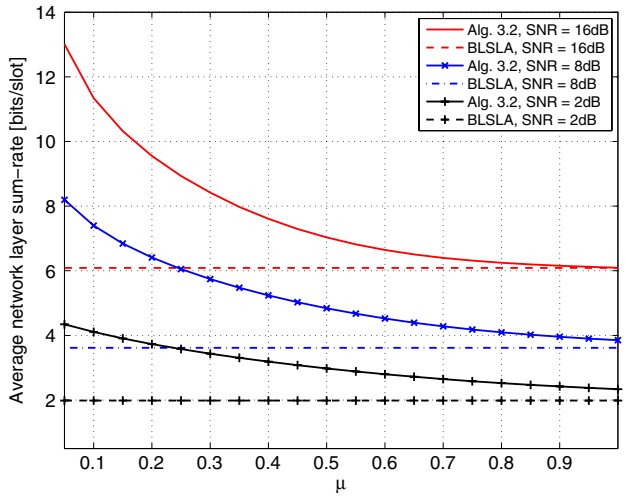
$$|h_{ijc}(t)|^2 = \mu^{|i-j|} c_{ijc}(t),\ i,j \in \mathcal{L},\ c \in \mathcal{C}, \tag{3.42}$$

where $c_{ijc}(t)$ are exponentially distributed independent random variables with unit mean used to model Rayleigh small-scale fading and the scalar $\mu \in [0,1]$ is referred to as the interference coupling index, which parameterizes the interference between direct links. For example, if $\mu = 0$, transmissions of links are interference free. The interference between transmissions increases as the parameter μ grows. Similar channel gain models for bipartite networks has also been used in [91]. Of course, this simple hypothetical model provides useful insights into the performance of presented algorithms in bipartite networks (e.g., cellular networks). We define the SNR operating point as

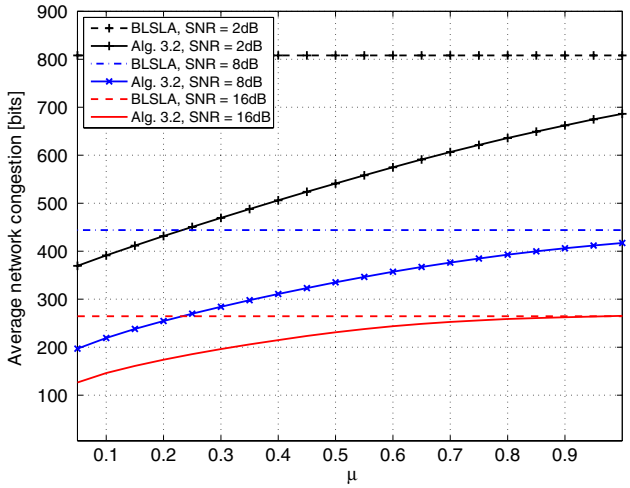
$$\text{SNR} = \frac{p_0^{\max}}{N_0 W_{\text{tot}}}. \tag{3.43}$$

Figure 3.4 shows the dependence of the average sum-rate [Figure 3.4(a)] and the average network congestion [Figure 3.4(b)] on the interference coupling index μ for Algorithm 3.2 and for the optimal base line single link activation (or BLSLA) policy.[4]. We consider the single-channel case $C = 1$ operating at three different SNR values 2, 8, and 16 dB. The initial power allocation $\mathbf{P}_0$ for Algorithm 3.2 is chosen such that $[\mathbf{P}_0]_{l,1} = p_0^{\max}$ unless otherwise specified. Here we can make several observations. First, Algorithm 3.2 provides substantial gains, both in the average sum-rate, as well as in the average network congestion, especially for small and medium values of the interference coupling index. The gains diminish as interference between direct links become significant. This is intuitively expected since for large SNR values the BLSLA policy becomes optimal when the interference coupling index

[4] A channel access policy where, during each time slot, only one link is activated in each channel is called BLSLA policy. Finding the optimal BLSLA policy that solves problem (3.6) is a combinatorial problem with exponential complexity in C. Thus, it quickly becomes intractable, even for moderate values of C. However, for the case $C = 1$ the optimal BLSLA policy can be easily found and it consists of activating, during each time slot, only the link that achieves the maximum weighted rate [84, sec. IV-B].



(a) Average network layer sum-rate $\sum_{s=1}^{4} \bar{x}_s^s$



(b) Average network congestion $\sum_{s=1}^{4} \bar{q}_s^s$

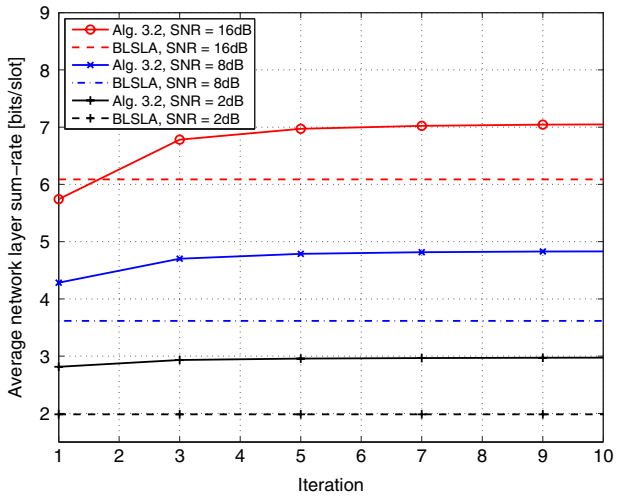
Fig. 3.4 Dependence of the average sum-rate and the average network congestion on the interference coupling index μ; $C = 1$ and SNR $= 2, 8, 16$ dB.

μ approaches 1. It is interesting to note that at small SNR values the network can still benefit from scheduling multiple links per slot, even for the case $\mu = 1$. This gain comes from the fact that the channel gains between interfering links are also affected by fading. Thus, links that

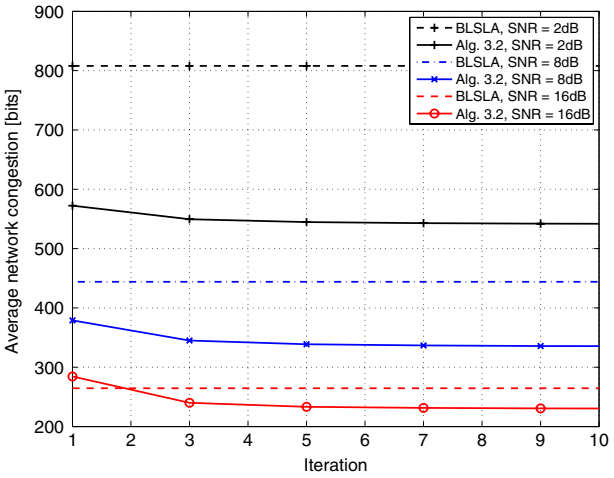
experience low instantaneous interference levels can be simultaneously scheduled. Results suggest that, especially for small and medium values of the interference coupling index, the given method often yields designs that are far superior to those obtained by BLSLA.

Figure 3.5 shows the dependence of the average sum-rate [Figure 3.5(a)] and of the average network congestion [Figure 3.5(b)] on the number of iterations of Algorithm 3.2. We consider the single-channel case $C = 1$ with interference coupling index $\mu = 0.5$ and SNR values 2, 8, and 16 dB. To facilitate faster convergence, Algorithm 3.2 is run without considering the trust region constraints; to do this, we can simply set the parameter α in Algorithm 3.2 to a very large positive number, e.g., $\alpha = 10^{100}$ [see problem (3.18)]. As a reference, we consider the optimal BLSLA policy. Results show that the incremental benefits are very significant for the first few iterations and are marginal later. For example, in the case of SNR = 16 dB, when the numbers of iterations changes from 1 to 3, the improvement in the average sum-rate is around 18.1%, whereas when it changes from 7 to 9, the improvement is around 0.30%. Therefore, by running Algorithm 3.2 for a few iterations (e.g., 5 iterations) we can yield performance levels that are almost indistinguishable from those that would have been obtained by running Algorithm 3.2 until it terminates (see the stopping criterion in step 3). This observation can be very useful in practice, since we can terminate Algorithm 3.2 when the incremental improvements between consecutive iterations become negligible.

Figure 3.6 shows the dependence of the average sum-rate [Figure 3.6(a)] and of the average network congestion [Figure 3.6(b)] on the SNR for Algorithm 3.2 and optimal BLSLA policy. We have considered the case $C = 1$ and $\mu = 0.3$. For comparison, we also plot the results due to a commonly used high SINR approximation [29] where the achievable rates $\log(1 + \gamma_{lc})$ are approximated by $\log(\gamma_{lc})$. In particular, the objective function of problem (3.11) is approximated by $\prod_{c\in\mathcal{C}} \prod_{l\in\mathcal{L}} \gamma_{lc}^{-\beta_l}$. Recall that γ_{lc} represents the SINR of link l in channel c and β_l represents the differential backlog of link l. This results in a convex approximation (i.e., a GP) of problem (3.11). One should not confuse high SINR with high SNR, since those are fundamentally different and a high SNR value does not ensure high



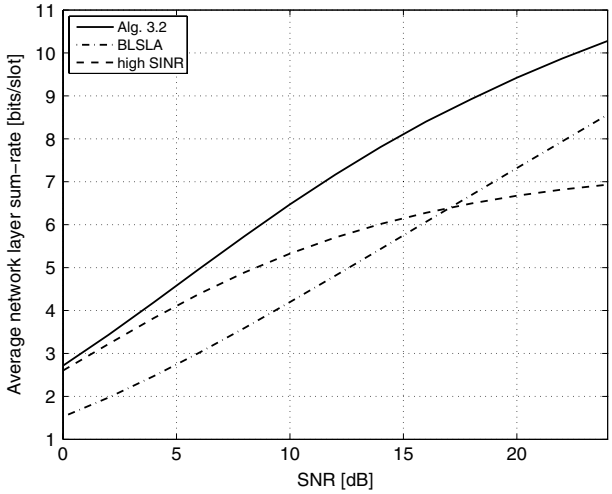
(a) Average network layer sum-rate $\sum_{s=1}^{4} \bar{x}_s^s$



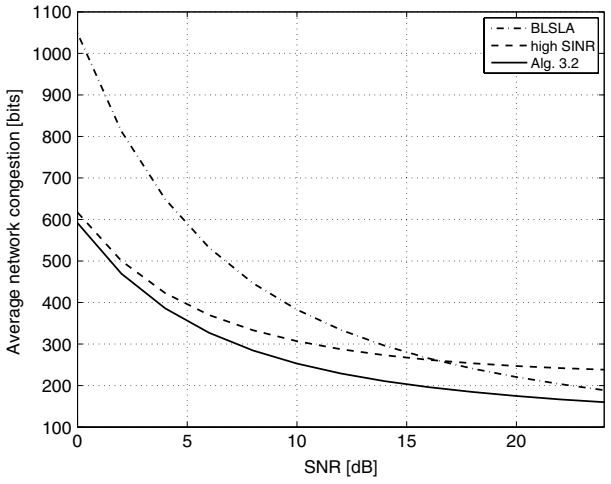
(b) Average network congestion $\sum_{s=1}^{4} \bar{q}_s^s$

Fig. 3.5 Dependence of the average sum-rate and the average network congestion on the iteration; $\mu = 0.5$, $C = 1$, and SNR $= 2, 8, 16$ dB.

SINR values in all links. Results show that, when compared with other methods, RA based on Algorithm 3.2 offers larger average sum-rate as well as reduced average network congestion. The relative gains of Algorithm 3.2 reduce, compared to the BLSLA at high SNR, for example, the relative gain offered by Algorithm 3.2 in the average



(a) Average network layer sum-rate $\sum_{s=1}^{4} \bar{x}_s^s$



(b) Average network congestion $\sum_{s=1}^{4} \bar{q}_s^s$

Fig. 3.6 Dependence of the average sum-rate and the average network congestion on the SNR; $C = 1$ and $\mu = 0.3$.

sum-rate changes from 40% to 17% [Figure 3.6(a)] and the relative gain in the average network congestion changes from 23% to 15% [Figure 3.6(b)] when the SNR value is increased from $\gamma = 16$ dB to $\gamma = 24$ dB respectively. This observation is consistent with that at high SNR it is very likely the optimal RA has a BLSLA structure. As a

result, at the optimal RA, different links correspond to different SINR regions, and therefore the high SINR approximation is, of course, unreasonable and suffers a large penalty, especially at high SNR values. This poor performance is qualitatively consistent with intuition: the solution obtained by employing high SINR approximation in RA must contain all nonzero entries (i.e., nonzero γ_{lc}) to drive the approximated objective $\left(\text{i.e., } \prod_{c\in\mathcal{C}}\prod_{l\in\mathcal{L}}\gamma_{lc}^{-\beta_l}\right)$ into a nonzero value, and therefore never yields a solution to the form of BLSLA.

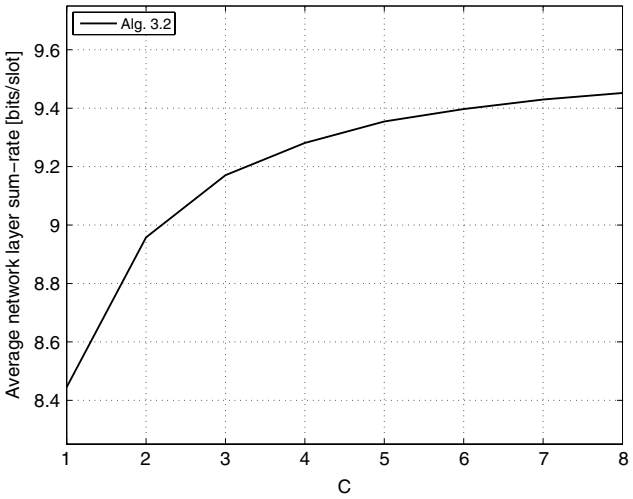
Figure 3.7 shows the dependence of the average sum-rate [Figure 3.7(a)] and of the average network congestion [Figure 3.7(b)] on the number of channels C for Algorithm 3.2. We consider the case $\text{SNR} = 16$ dB and $\mu = 0.3$ and the initial power allocation $\mathbf{P}_0$ for Algorithm 3.2 is simply chosen such that $[\mathbf{P}_0]_{l,c} = p_0^{\max}/C$. The plots illustrate that increasing the number of channels will yield better performance in both the average sum-rate and the average network congestion (e.g., when the number of channels C changes from 1 to 8, the improvement in the average sum-rate and the reduction in average network congestion is around 12% and 12.4%, respectively). We stress that the benefits are solely achieved by opportunistically exploiting the available multichannel diversity in the network via Algorithm 3.2 without any supplementary bandwidth or power consumption. Moreover, the incremental benefits are very significant for small C, for example, when the number of channels C changes from 1 to 2, the improvement in the average sum-rate is around 6%, whereas when C changes from 7 to 8, the improvement is around 0.25%. The plots give much insight into *why* multichannel designs are important and beneficial compared to the single-channel counterpart.

3.4.2 NUM for Nonbipartite Networks with Singleuser Detection at Receivers

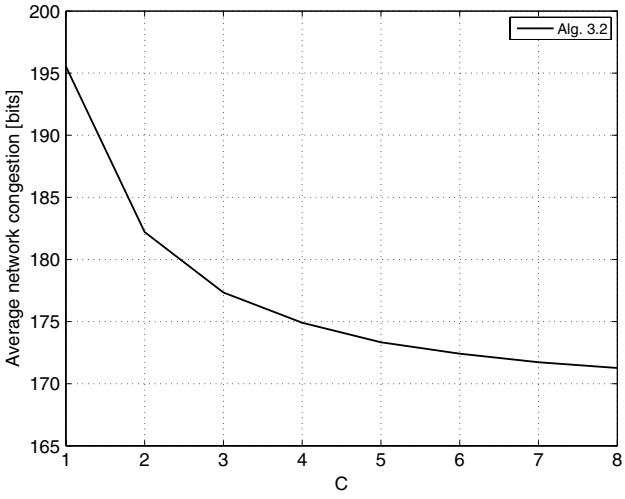
First, two small fully connected multihop wireless network setups, which are identical to the once shown in Figure 2.11 are considered.

We assume an exponential path loss model; the channel power gains $|h_{ijc}(t)|^2$ between distinct nodes are given by

$$|h_{ijc}(t)|^2 = \left(\frac{d_{ij}}{d_0}\right)^{-\eta} c_{ijc}(t), \tag{3.44}$$



(a) Average network layer sum-rate $\sum_{s=1}^{4} \bar{x}_s^s$



(b) Average network congestion $\sum_{s=1}^{4} \bar{q}_s^s$

Fig. 3.7 Dependence of the average sum-rate and the average network congestion on the number of channels C; SNR = 16 dB and $\mu = 0.3$.

where d_{ij} is the distance from the transmitter of link i to the receiver of link j, d_0 is the *far field reference distance* [64], η is the path loss exponent, and $c_{ijc}(t)$ are exponentially distributed random variables with unit mean, independent over the time slots, links, and channels.

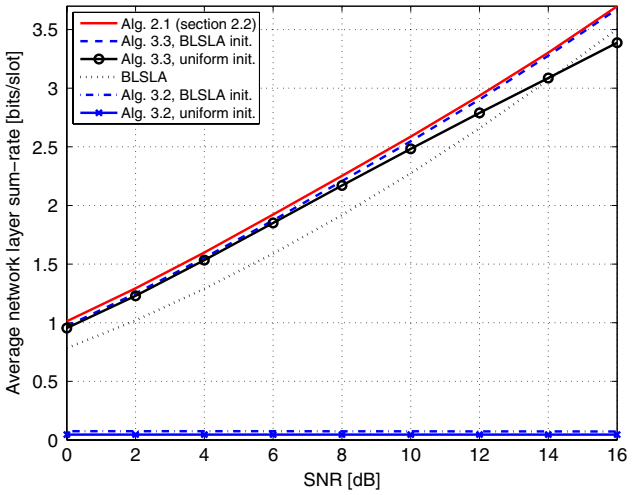
The first term of (3.44) represents the path loss factor and the second term models Rayleigh small-scale fading. The SNR operating point is defined as

$$\mathrm{SNR} = \frac{p_0^{\max}}{N_0 W_{\mathrm{tot}}} \left(\frac{D_0}{d_0} \right)^{-\eta}. \tag{3.45}$$

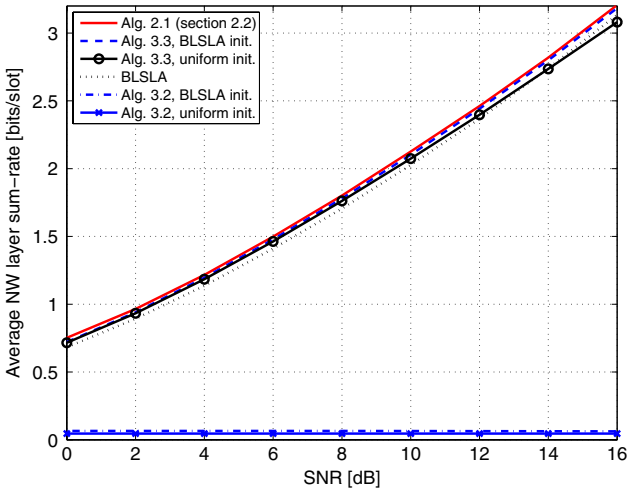
In the following simulations we set $D_0/d_0 = 10$ and $\eta = 4$.

Figure 3.8 shows the dependence of the average network layer sum-rate on the SNR for the considered network setups, where we use $C = 1$. As a benchmark, we first consider the branch and bound algorithm presented in Section 2.2 to optimally solve the RA subproblem. It should be stressed that the optimality of the algorithm given in Section 2.2 is achieved at the expense of prohibitive computational complexity, even in the case of very small problem instances. We then consider the optimal BLSLA policy and Algorithm 3.3 with two initialization methods: (1) Uniform initialization and (2) BLSLA based initialization. In the case of uniform initialization the initial power allocation $\mathbf{P}_0$ is chosen such that $[\mathbf{P}_0]_{l,1} = p_0^{\max}/(|\mathcal{O}_{\mathrm{tran}(l)}|)$. In the case of BLSLA based initialization the initial power allocation $\mathbf{P}_0$ is chosen such that $[\mathbf{P}_0]_{l^\star,1} : [\mathbf{P}_0]_{j,1} = P : 1$ for all $j \in \mathcal{L},\ j \neq l^\star$ where $l^\star$ is the index of the active link obtained based on the optimal BLSLA policy and $P \gg 1$ is a real number. For comparison, we also plot the results for Algorithm 3.2 with uniform and BLSLA initializations.

Results show that the performance of Algorithm 3.3 is very close to the optimal branch and bound algorithm. Specifically, Algorithm 3.3 with BLSLA initialization is almost indistinguishable from the optimum and at least as good as the optimal BLSLA, for all considered cases. In contrast, Algorithm 3.3 with uniform initialization exhibits significant deviations from both the optimal branch and bound algorithm and BLSLA, specially at high SNR values. This behavior is not surprising since Algorithm 3.3 is a local method for the nonconvex problem (3.6). Therefore, the initialization point of the algorithm can influence the resulting solution [22, sec. 1.4.1]. Nevertheless, a carefully selected initialization point can improve the performance of Algorithm 3.3 to very close to the optimum. For example, at high SNR values, the performance of Algorithm 3.3 with BLSLA initialization is



(a) Average network layer sum-rate $\bar{x}_1^1 + \bar{x}_4^2$



(b) Average network layer sum-rate $\bar{x}_1^1 + \bar{x}_2^2$

Fig. 3.8 (a) Dependence of the average network layer sum-rate on SNR for network 1 [Figure 2.11(a)]; (b) Dependence of the average network layer sum-rate on SNR for network 2 [Figure 2.11(b)].

almost identical to the optimum, whereas the performance with uniform initialization deviates a bit from the optimum. It is important to remark that at low and moderate values of SNR, results derived from Algorithm 3.3 are not significantly affected by the initialization method.

Results also show that, in the presence of self interferers, Algorithm 3.2 cannot perform well and it can converge to a very bad suboptimal point, as we pointed out in Section 3.2.4. Therefore, even though the computational complexity of Algorithm 3.3 *does not increase* more than polynomially with the problem size, results show that Algorithm 3.3 with a proper initialization performs close to the optimum.

Next, a larger network; a fully connected multihop, multicommodity wireless network as shown in Figure 3.9 is considered. There are $N = 9$ nodes and $S = 3$ commodities. The commodities arrive exogenously at different nodes in the network as described in Table 3.1. Thus we have $\mathcal{S}_1 = \{2\}, \mathcal{S}_2 = \{3\}, \mathcal{S}_3 = \{3\}, \mathcal{S}_5 = \{2\}, \mathcal{S}_7 = \{1,3\}$, and $\mathcal{S}_i = \emptyset$ for all $i \in \{4,6,8,9\}$. The nodes are located in a rectangular grid such that the horizontal and vertical distances between adjacent nodes are D_0 m. The channel power gains, between nodes, are given by (3.44) and the SNR operating point is given by (3.45). Moreover, we set $D_0/d_0 = 10$ and $\eta = 4$.

Figure 3.10 shows the dependence of the average sum-rate [Figure 3.10(a)] and of the average network congestion [Figure 3.10(b)]

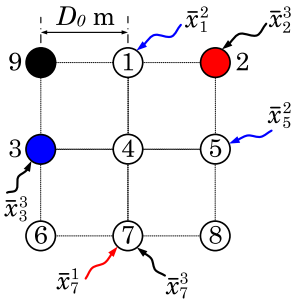
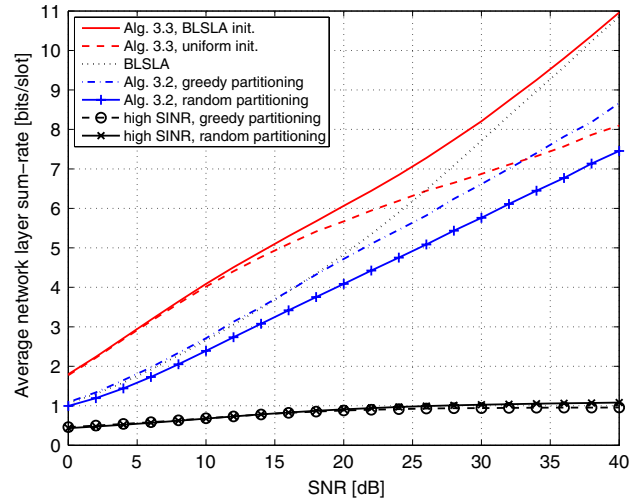


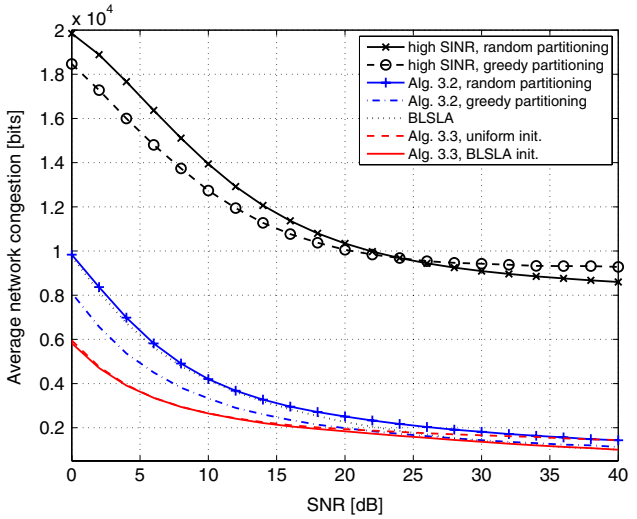
Fig. 3.9 Multihop wireless network with $N = 9$ nodes and $S = 3$ commodities.

Table 3.1. Network commodities, destination nodes, and source nodes.

Commodity (s)	Destination node (d_s)	Source nodes
1	2	7
2	3	1,5
3	9	2,3,7



(a) Average network layer sum-rate $\sum_{n=1}^{9} \sum_{s \in \mathcal{S}_n} \bar{x}_n^s$



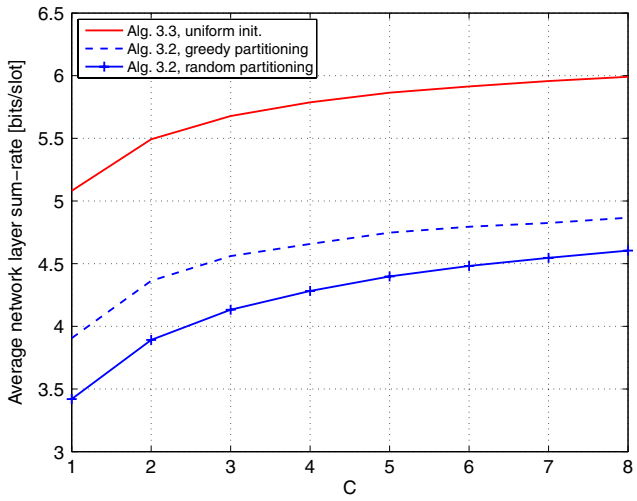
(b) Average network congestion $\sum_{n=1}^{9} \sum_{s=1}^{3} \bar{q}_n^s$

Fig. 3.10 Dependence of the average sum-rate and the average network congestion on the SNR; $C = 1$.

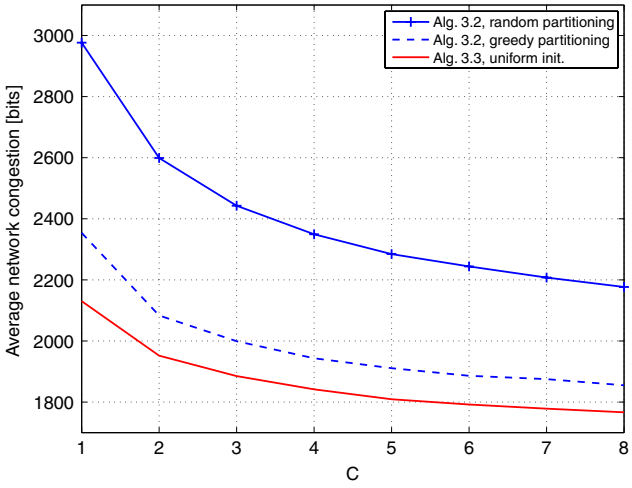
on the SNR for several algorithms, where we use $C = 1$. First we have considered the optimal BLSLA policy and Algorithm 3.3 with the two initialization methods: (1) Uniform initialization and (2) BLSLA based initialization (the same initializations as used when plotting Figure 3.8).

For comparison, we also plot the results for the low-complex approaches where the set of nodes $\mathcal{N}$ is first partitioned into two disjoint subsets, the set of transmitting nodes $\mathcal{T}$ and the set of receiving nodes $\mathcal{R}$ and then Algorithm 3.2 and high SINR approximation are used in RA. The partitioning of the set of nodes $\mathcal{N}$ into two disjoint subsets is performed using two simple methods: (1) random partitioning and (2) greedy partitioning based on differential backlogs. In random partitioning, each node is allocated either to $\mathcal{T}$ or to $\mathcal{R}$ with equal probabilities. Greedy partitioning is performed as follows. We start with an empty set of links $\bar{\mathcal{L}} = \emptyset$. At each step, the link $l^\star$ from the set $\mathcal{L} \setminus \bar{\mathcal{L}}$ which has the largest differential backlog β_l (i.e., $l^\star = \arg\max_{l \in \mathcal{L} \setminus \bar{\mathcal{L}}} \beta_l$) is added to the set $\bar{\mathcal{L}}$. Then all links outgoing from $\mathrm{rec}(l^\star)$ and all links incoming to $\mathrm{tran}(l^\star)$ are deleted from $\mathcal{L}$. This procedure continues until there are no links left in $\mathcal{L} \setminus \bar{\mathcal{L}}$. The sets $\mathcal{T}$ and $\mathcal{R}$ can be found as $\mathcal{T} = \{\mathrm{tran}(l) | l \in \bar{\mathcal{L}}\}$ and $\mathcal{R} = \{\mathrm{rec}(l) | l \in \bar{\mathcal{L}}\}$.

From Figure 3.10 we make the following observations. First, Algorithm 3.3 with BLSLA based initialization yields results better than any other counterpart. In contrast, Algorithm 3.3 with uniform initialization shows significant deviations from the BLSLA solution at high SNR, especially in the terms of average sum-rate [Figure 3.10(a)]. Moreover, it is important to observe again that at low and moderate values of SNR, results derived from Algorithm 3.3 are not substantially affected by the initialization method. These observations are almost the same as those we saw in Figure 3.8. We also observe that Algorithm 3.3 with a proper initialization, can significantly outperform Algorithm 3.2 in conjunction with either random or greedy partitioning. This elaborates the importance of gradual self-interference gain increments (i.e., step 4 of Algorithm 3.3) in finding a better RA compared to the direct application of Algorithm 3.2 with heuristic partitioning. In most cases there is no advantage to using high SINR approximation. These observations are very useful in practice since they illustrate that Algorithm 3.3 often works well when initialized with a reasonable starting point (e.g., BLSLA based initialization). In addition, we note that even with a very simple initialization, for example, uniform initialization, Algorithm 3.3 yields substantial gains, especially at small and moderate SNR values (e.g., 0 dB–20 dB).



(a) Average network layer sum-rate $\sum_{n=1}^{9} \sum_{s \in \mathcal{S}_n} \bar{x}_n^s$



(b) Average network congestion $\sum_{n=1}^{9} \sum_{s=1}^{3} \bar{q}_n^s$

Fig. 3.11 Dependence of the average sum-rate and the average network congestion on the number of channels C; SNR = 16 dB.

Figure 3.11 shows the dependence of the average sum-rate [Figure 3.11(a)] and of the average network congestion [Figure 3.11(b)] on the numbers of channels C for Algorithm 3.3. We have considered the case SNR = 16 dB and have considered a uniform initialization for

Algorithm 3.3 where the initial power allocation $\mathbf{P}_0$ is chosen such that $[\mathbf{P}_0]_{l,c} = p_0^{\max}/(C.|\mathcal{O}_{\text{tran}(l)}|)$. For comparison, we also plot the results for Algorithm 3.2 with random and greedy partitioning of nodes $\mathcal{N}$. The results are consistent with the previous observations in Figure 3.7, that is, as the number of channels increases better performance in both the average sum-rate and the average network congestion is achieved. These benefits are again obtained by opportunistically exploiting the available multichannel diversity in the network via the algorithms presented. Moreover, the results suggest that using Algorithm 3.3 in the RA can increase the gains very significantly, compared to the RA based on simple extensions to Algorithm 3.2, which runs with either random or greedy partitioning of nodes. For example, the relative gains in the average sum-rate are above 23% [Figure 3.11(a)] and the relative gains in the average network congestion are above 4.7% [Figure 3.11(b)] over the range of interest, $C = 1$ to $C = 8$.

3.4.3 Effect of Self Interference Cancelation

For the considered network setups in this section, the channel power gains between nodes are given by (3.44) and the SNR operating point is given by (3.45). For illustration purposes we consider a single-channel case (i.e., $C = 1$). Moreover, we set $D_0/d_0 = \sqrt{10}$ and $\eta = 4$. In all the simulations in this subsection, Algorithm 3.3 with BLSLA based initialization is considered.

Inspired by the Gaussian two-way channel [36], we first consider a simple two node wireless network as shown in Figure 3.12(a). There are two commodities, the first one arrives at node 1, and is intended for node 2; the second commodity arrives at node 2, and is intended for

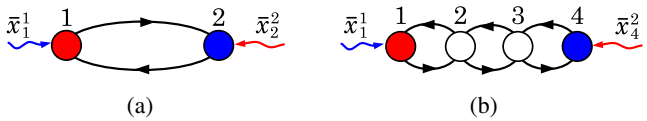


Fig. 3.12 (a) Two node wireless network with $N = 2$ nodes, $L = 2$ links, and $S = 2$ commodities. Different commodities are represented by different color; (b) Tandem wireless network with $N = 4$ nodes and $S = 2$ commodities. Different commodities are represented by different color.

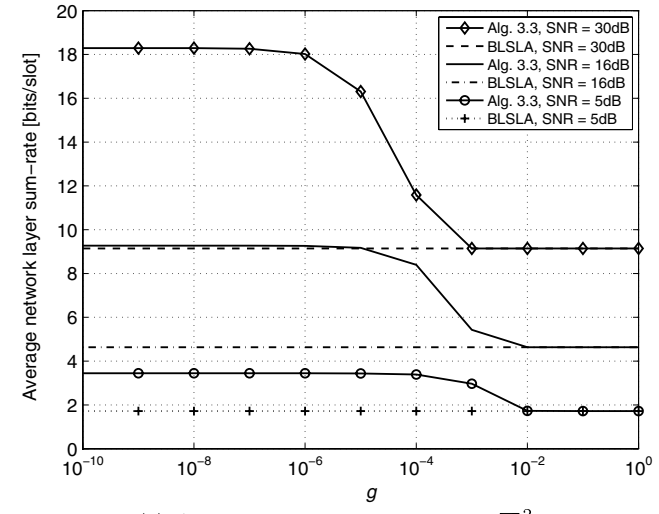
node 1. As explained in [36], a Gaussian two-way channel is equivalent to two independent Gaussian channels where perfect self interference cancelation is realized (i.e., $g = 0$). As a result, the sum-capacity of the symmetric Gaussian two-way channel becomes twice the capacity of either of the equivalent Gaussian channels. The considered two node network allows us to illustrate similar behavior in terms of the network layer average sum-rate.

Figure 3.13 shows the dependence of the average sum-rate [Figure 3.13(a)] and of the average network congestion [Figure 3.13(b)] on the self-interference gain g. We consider three link SNR values, $5, 16$, and 30 dB, which correspond to low, medium, and high data rate systems respectively. The results show that the average sum-rate with perfect self interference cancelation (i.e., $g = 0$) is increased by a factor of 2 and the average network congestion reduced significantly, as compared to no self interference cancelation (i.e., $g = 1$); see Figure 3.13(a). Similar gains are achieved in terms of average network congestion as well; see Figure 3.13(b). The results also reveal that, even with an imperfect self interference cancelation technique, we can achieve the performance limits guaranteed by perfect self interference cancelation. For example, a decrease of the self-interference gain up to a value $g = 10^{-4}$ is enough to double the average sum-rate for link SNR = 5 dB.

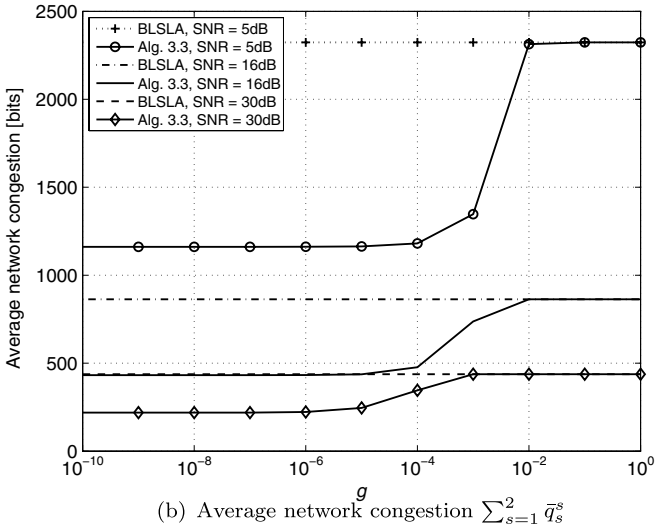
Let us now consider a tandem wireless network, as shown in Figure 3.12(b). There are two commodities, the first one arrives at node 1, and is intended for node 4; the second commodity arrives at node 4, and is intended for node 1. Thus we have $\mathcal{S}_1 = \{1\}, \mathcal{S}_4 = \{2\}$, and $\mathcal{S}_n = \emptyset$ for all $n \in \{2, 3\}$.

Figure 3.14 shows the dependence of the average sum-rate and of the average network congestion on the self-interference gain g for SNR values $5, 16$, and 30 dB. The behavior is very similar to the case of Figure 3.13. For example, in the case SNR = 5 dB, the results show that by decreasing the self-interference gain from $g = 10^{-1}$ to $g = 10^{-4}$ the average sum-rate is increased by a factor of around 1.82 [see Figure 3.14(a)] and the average network congestion has reduced significantly as well [see Figure 3.14(b)].

Let us next consider a fully connected multihop, multicommodity wireless network as shown in Figure 3.9. Figure 3.15 shows the



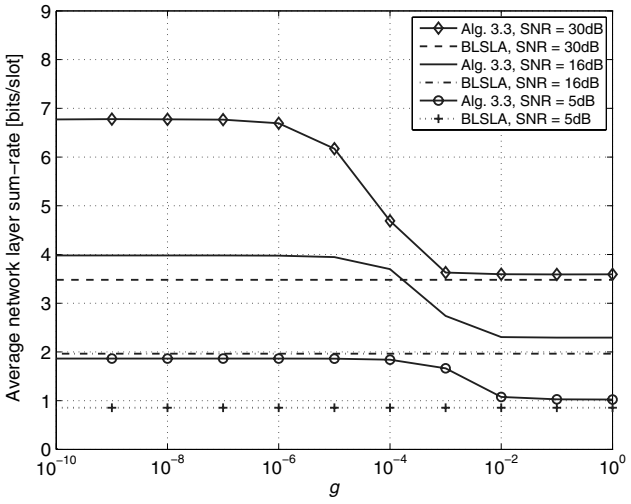
(a) Average network layer sum-rate $\sum_{s=1}^{2} \bar{x}_s^s$



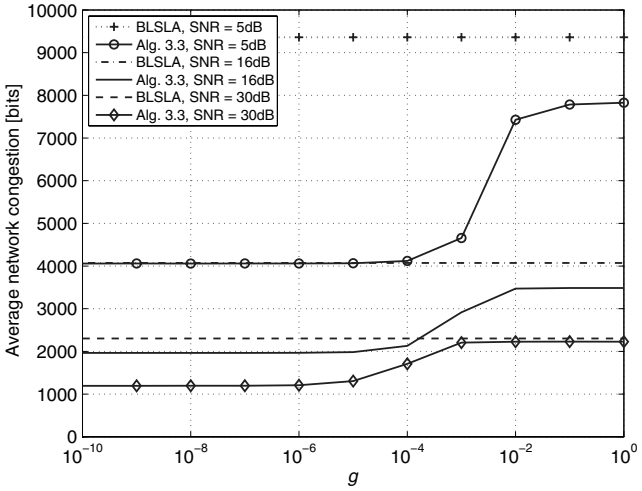
(b) Average network congestion $\sum_{s=1}^{2} \bar{q}_s^s$

Fig. 3.13 Dependence of the average sum-rate and of the average network congestion on the self-interference gain g in the case of the two node wireless network.

dependence of the average sum-rate and of the average network congestion on the self-interference gain g for SNR values $5, 16$, and 30 dB. Let us first consider the case of a low SNR value, i.e., SNR $= 5$ dB. The results show that by decreasing the self-interference gain from $g = 10^{-1}$



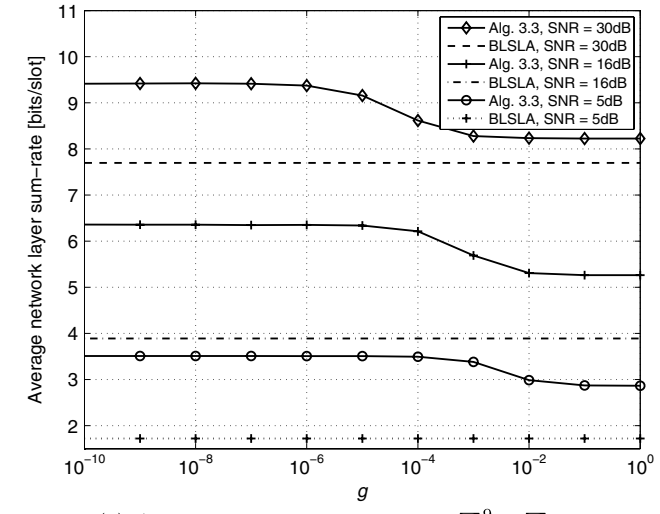
(a) Average network layer sum-rate ($\bar{x}_1^1 + \bar{x}_4^2$)



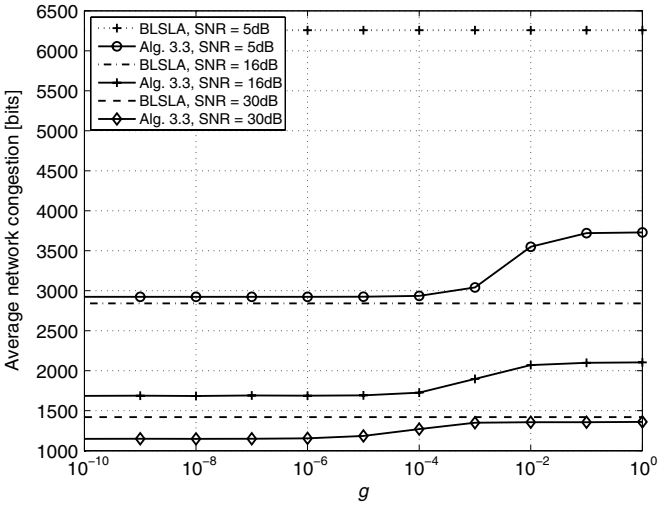
(b) Average network congestion $\sum_{n=1}^{4} \sum_{s=1}^{2} \bar{q}_n^s$

Fig. 3.14 Dependence of the average sum-rate and of the average network congestion on the self-interference gain g in the case of the four node tandem wireless network.

to $g = 10^{-4}$ the average sum-rate is increased by a factor of about 1.22 [see Figure 3.15(a)]. From Figure 3.15(b), we see reductions in the average network congestion as well. The network performance remains the same as in the case of perfect self interference cancelation for all



(a) Average network layer sum-rate $\sum_{n=1}^{9} \sum_{s \in \mathcal{S}_n} \bar{x}_n^s$



(b) Average network congestion $\sum_{n=1}^{9} \sum_{s=1}^{3} \bar{q}_n^s$

Fig. 3.15 Dependence of the average sum-rate and of the average network congestion on the self-interference gain g in the case of the nine node multihop wireless network.

values of $g < 10^{-4}$. In this region the network performance is limited by the interference between distinct nodes, and no further improvement is possible by only increasing the accuracy of the self interference cancelation. On the other hand, no gain in the network performance is

achieved by using an imperfect self interference cancelation technique which leads to $g > 10^{-1}$. In this region the RA solution provided by Algorithm 3.3 is always *admissible* (i.e., no node transmits and receives simultaneously).

In each considered network setup (i.e., Figures 3.12(a), 3.12(b), and 3.9) a similar behavior of the results holds for medium and high SNR values as well (i.e., SNR = 16 and 30 dB). Moreover, as we change SNR from low values to high values, the accuracy level required by the self interference cancelation becomes more stringent. For example, in the case of the fully connected multihop, multicommodity wireless network in figure 3.9, if the SNR operating point is changed from 5 to 30 dB, then the accuracy level required by the self interference cancelation should be improved from $g = 10^{-1}$ to $g = 10^{-3}$ to start gaining in network layer performances. This is intuitively expected since, the larger the SNR operating point, the larger the power levels of the nodes, and therefore the higher the accuracy level required by the self interference cancelation techniques to remove the increased transmit power at nodes.

Note that the relative gains due to self interference cancelation in the considered fully connected multihop network [Figure 3.9] are smaller compared to the relative gains experienced in the tandem wireless network [Figure 3.12(b)]. This behavior is intuitively explained by looking in to the network topology. When the self interference is significantly canceled, the resultant interference at the receiver node of any link in the case of the tandem multihop wireless network in Figure 3.12(b) is smaller on average to that of the multihop wireless network in Figure 3.9; note that any receiver node of the fully connected multihop network has many adjacent interfering nodes. Thus, with zero self interference, links in the tandem network can operate at larger rates, and therefore yields larger relative gains.

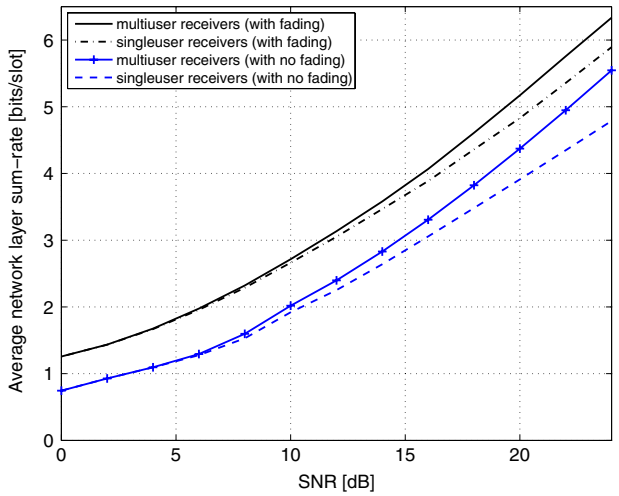
Finally, we show by an example, how to apply rate-equivalence (3.28) to find the required value of self-interference gain g in order to preserve network layer performances if the distance between nodes is scaled. Let us construct a new network by scaling the distances between the nodes of the original network (see Figure 3.9) by

a factor of $\theta = \sqrt{10}$ and the maximum node transmission power $p_0^{\max}$ by a factor of $\theta^\eta = 100$ (note that $\eta = 4$). We refer to this new network as the *scaled network*. To illustrate the idea let us consider the case SNR = 5 dB in Figure 3.10 and focus to the point $g = 10^{-4}$ for which the average sum-rate is 3.5 [bits/slot]. The value of g at this point can be considered as the minimum required accuracy level of self interference cancelation to achieve an average sum-rate of 3.5 bits/slot in the original network. Now we ask what is the required self-interference gain g_{new} that would result in the same average sum-rate value (i.e., 3.5 bits/slot) in the scaled network. From (3.28) it follows that the required accuracy level of self interference cancelation should be improved at least to a level of $g_{\text{new}} = g/\theta^\eta = 10^{-4}/100 = 10^{-6}$.

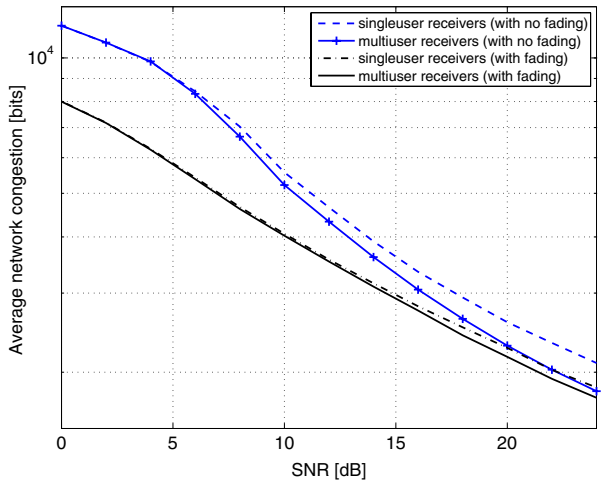
3.4.4 NUM for Networks with Multiuser Detection at Receivers

The network in Figure 3.9 is considered. The assumptions and the simulation parameters are exactly the same as in Section 3.4.2.

Figure 3.16 shows the dependence of the average sum-rate [Figure 3.16(a)] and of the average network congestion [Figure 3.16(b)] on the SNR for RA where only one node is allowed to transmit in each slot and receivers perform multiuser detection. For illustration purposes we consider the single-channel case (i.e., $C = 1$). We also show the results for a nonfading case [i.e., by having $c_{ijc}(t) = 1$ in (3.24)] for comparison. Here we can make several observations. Fading can significantly improve the overall performance in average sum-rate and average network congestion. This observation has an analogy with multiuser diversity in downlink fading channels [124, sec. 6.6]. Intuition suggests that when there are many links which fade independently, at any time slot there is a high probability that the resulting rate and power allocation yields a better schedule (see [43, sec. 4.7]) compared to the nonfading case. There are significant advantages to having multiuser detection, especially for high SNR values. At low SNR, gains are marginal. Thus, multiuser detectors have a practical advantage over singleuser detectors, especially in a high SINR regime. For example,



(a) Average network layer sum-rate $\sum_{n=1}^{9} \sum_{s \in \mathcal{S}_n} \bar{x}_n^s$



(b) Average network congestion $\sum_{n=1}^{9} \sum_{s=1}^{3} \bar{q}_n^s$

Fig. 3.16 Dependence of the average sum-rate and the average network congestion on the SNR; $C = 1$.

in a fading environment, at SNR $= 24$ dB we obtain around a 7.5% increase in the average sum-rate and a 5% decrease in the average network congestion. In a nonfading environment multiuser detectors offer around a 16% increase in the average sum-rate and a 13.5% decrease in the average network congestion.

3.5 Summary and Discussion

We considered the power and rate control problem in a wireless network in conjunction with the next-hop routing/scheduling and the flow control problem. Thus, although the focus lies on the so-called resource allocation problem that is confined to the physical/medium access control layers, its formulation captures interactions with the higher-layers in a manner similar to the one employed in [82]. The result is a cross layer formulation. The problem, unfortunately, is NP-hard, and therefore there are no polynomial time algorithms to solve it. Our presentation has been to consider first a general access operation, but with a relatively simple form of receiver structure (bank of match filters), and then to limit the access operation to a single node at a time (either transmitting or receiving) but allowing for increased multiuser detector complexity at the receiver.

In the first case, we presented a new optimization methodology based on homotopy methods and complementary geometric programming solution methods. Numerical results showed that the presented algorithms perform close to exponentially complex optimal solution methods. In addition, they are in fact fast and are capable of handling large-scale problems.

The considered method was also used to evaluate the gains achievable at the network layer when the network nodes employ self interference cancelation techniques with different degrees of accuracy. Numerical results showed that the self interference cancelation requires a certain level of accuracy to obtain quantifiable gains at the network layer. The gains saturate after a certain cancelation accuracy. The level of accuracy required by the self interference cancelation techniques depends on many factors, such as the distances between the network nodes and the operating power levels of the network nodes. For the considered network setups, the numerical results showed that a self interference reduction in the range $20-60$ dB leads to significant gains at the network layer. We emphasize that this level of accuracy is practically achievable, for example, the recent proposals [32, 100, 101, 102] provide cost effective mechanisms for an up to 55 dB reduction in the self-interference coefficient. These observations are indeed important in

the context of certain future cellular systems. For example, as the trend in cellular systems is to increase the data rates of users by introducing very small cells (e.g., femtocells), the difference between transmitted and received power might not be very large. Therefore, self interference cancelation strategies can be of great benefit to such systems. Numerical results further showed that the topology of the network has a substantial influence on performance gains. For example, in the case of tandem multihop wireless networks, the benefits due to self interference cancelation are more pronounced when compared to those of a multihop network, in which the nodes are located in a square grid.

In the second case, we obtained a complete solution and illustrate numerically the performance gain due to multiuser detector capability. The main benefit here is the simplicity of the solution methods. As a result, these simple access protocols can be potentially useful in practical applications with more advanced communication systems.

4

A Distributed Approach

This section present a distributed method for WSRMax problem in a multicell MISO downlink system. Unlike the WMMSE algorithm[1] proposed in [110], the method presented in this section does not rely on user terminals' assistance such as estimations, computations, and feedback information to base stations over the air interface. Only base station to base station (BS) synchronized communication is required, where all the signalling overhead is exchanged through reliable backhaul links (e.g., fiber and microwave links). All the necessary computation can be carried out *asynchronously* at each BS without any involvement of the user terminals. Thus, the algorithm presented in this section is well suited for systems where the user terminal support is not allowed or not desirable. The algorithm is based on primal decomposition methods and subgradient methods [17, 19]. Specifically, primal decomposition techniques [17] are applied to split the problem into a master problem and many subproblems. In the case of master problem, a sequential convex approximation strategy [18] together with a subgradient method [19] that relies on BS coordinations is adopted.

[1] We use the same acronym WMMSE to refer to this algorithm as suggested in [110].

In the case of subproblems, an existing algorithm originally proposed in [34, sec. 4.3], which is based on second-order cone programming (SOCP) [73] and geometric programming (or GP) [20] is adopted. The subproblems (or BS optimizations) can be carried out in a fully asynchronous manner. With appropriate choice of stopping criteria, the monotonic convergence of the algorithm is guaranteed. Practical stopping criteria, which are favorable for implementing the algorithm, but at the expense of a sacrificing the monotone convergence are also presented. Numerical results are provided to compare this method with WMMSE algorithm [110], the GP/SOCP based algorithm proposed in [34, sec. 4.3], and the distributed algorithm proposed in [10, 11]. The behavior of the algorithm under different degrees of BS coordination is also discussed and numerically illustrated.

4.1 System Model and Problem Formulation

A multicell MISO downlink system, with N BSs each equipped with T transmit antennas is considered. The set of all BSs is denoted by $\mathcal{N}$ and we label them with the integer values $n = 1, \ldots, N$. The *transmission region* of each BS is modeled as a disc with radius R_{BS} centered at the location of the BS. Single data stream is transmitted for each user. We denote the set of all data streams in the system by $\mathcal{L}$ and label them with the integer values $l = 1, \ldots, L$. The transmitter node (i.e., the BS) of lth data stream is denoted by $\mathrm{tran}(l)$ and the receiver node of lth data stream is denoted by $\mathrm{rec}(l)$. We have $\mathcal{L} = \cup_{n \in \mathcal{N}} \mathcal{L}(n)$, where $\mathcal{L}(n)$ denotes the set of data streams transmitted by nth BS. Note that the users of the data streams transmitted by each BS are necessarily located inside the transmission region of the BS (see Figure 4.1).

The antenna signal vector transmitted by nth BS is given by

$$\mathbf{x}_n = \textstyle\sum_{l \in \mathcal{L}(n)} \sqrt{p_l} d_l \mathbf{v}_l, \tag{4.1}$$

where $p_l \in \mathbb{R}_+$ denotes the power, $d_l \in \mathbb{C}$ represents the information symbol, and $\mathbf{v}_l \in \mathbb{C}^T$ is the beamformer, all associated to lth data stream. We assume that d_l and $\mathbf{v}_l$ are normalized such that $\mathrm{E}|d_l|^2 = 1$ and $\|\mathbf{v}_l\|_2 = 1$. Moreover, we assume independent data streams, i.e., $\mathrm{E}\{d_l d_j^*\} = 0$ for all $l, j \in \mathcal{L}$, where $l \neq j$.

The signal received at rec(l) is given by

$$y_l = \mathbf{h}_{ll}^{\mathrm{H}}\sqrt{p_l}d_l\mathbf{v}_l + \sum_{j\in\mathcal{L}(\mathrm{tran}(l)),j\neq l}\mathbf{h}_{jl}^{\mathrm{H}}\sqrt{p_j}d_j\mathbf{v}_j + \sum_{j\in\mathcal{L}\setminus\mathcal{L}(\mathrm{tran}(l))}\mathbf{h}_{jl}^{\mathrm{H}}\sqrt{p_j}d_j\mathbf{v}_j + z_l \tag{4.2}$$

$$= \mathbf{h}_{ll}^{\mathrm{H}}\sqrt{p_l}d_l\mathbf{v}_l + \sum_{j\in\mathcal{L}(\mathrm{tran}(l)),j\neq l}\mathbf{h}_{jl}^{\mathrm{H}}\sqrt{p_j}d_j\mathbf{v}_j + \sum_{i\in\mathcal{N}\setminus\{\mathrm{tran}(l)\}}\sum_{j\in\mathcal{L}(i)}\mathbf{h}_{jl}^{\mathrm{H}}\sqrt{p_j}d_j\mathbf{v}_j + z_l, \tag{4.3}$$

where $\mathbf{h}_{jl}^{\mathrm{H}} \in \mathbb{C}^{1\times T}$ is the channel matrix between tran(j) and rec(l), and z_l is circular symmetric complex Gaussian noise with variance σ_l^2. Note that the second term in (4.3) represents the intra-cell interference. The third term represents the out-of-cell interference. The received SINR of lth data stream is given by

$$\gamma_l = \frac{p_l|\mathbf{h}_{ll}^{\mathrm{H}}\mathbf{v}_l|^2}{\sigma_l^2 + \sum_{j\in\mathcal{L}(\mathrm{tran}(l)),j\neq l} p_j|\mathbf{h}_{ll}^{\mathrm{H}}\mathbf{v}_j|^2 + \sum_{i\in\mathcal{N}\setminus\{\mathrm{tran}(l)\}} z_{il}}, \tag{4.4}$$

where $z_{il} = \sum_{j\in\mathcal{L}(i)} p_j|\mathbf{h}_{jl}^{\mathrm{H}}\mathbf{v}_j|^2$ represents the out-of-cell interference power from ith BS to $rec(l)$, which is typically known as *interference temperature* [152].

The out-of-cell interference term in (4.4) (i.e., $\sum_{i\in\mathcal{N}\setminus\{\mathrm{tran}(l)\}} z_{il}$) prevents resource allocation (RA) on an intra-cell basis and demands centralized RA methods. To facilitate potential distributed algorithms for RA, we make the following assumption: transmissions from ith BS *do interfere* the lth data stream transmitted by BS $n \neq i$, if the distance between ith BS and $rec(l)$ is smaller than a threshold R_{int}.[2] The disc with radius R_{int} centered at the location of any BS is referred to as the *interference region* of the BS (see Figure 4.1). Thus, if ith BS is located at a distance larger than R_{int} to $rec(l)$, then the associated z_{il} components are set to zero.[3] Based on the assumption above, we can

[2] Similar assumptions are made in [44] in the context of arbitrary wireless networks.

[3] The value of R_{int} is chosen such that the power of the interference term is below the noise level and this commonly used approximation is made to avoid unnecessary coordinations between distant BSs. The effect of nonzero z_{il} terms can be accurately modeled by changing the statistical characteristics of noise z_l at $rec(l)$. However, those issues are extraneous to the main focus of the section.

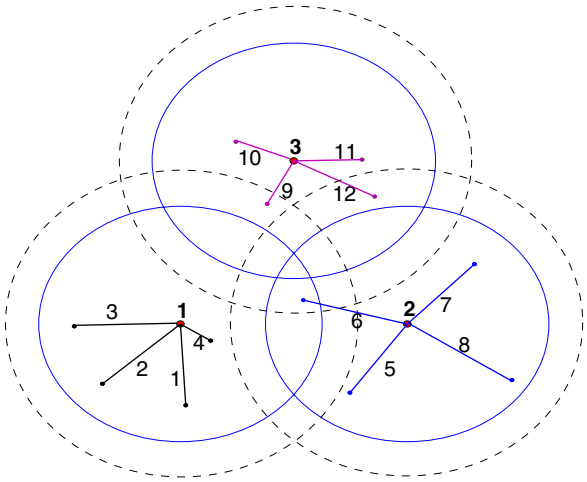


Fig. 4.1 Multicell network, $\mathcal{N} = \{1,2,3\}$, $\mathcal{L} = \{1,\ldots,12\}$, $\mathcal{L}(1) = \{1,\ldots,4\}$, $\mathcal{L}(2) = \{5,\ldots,8\}$, $\mathcal{L}(3) = \{9,\ldots,12\}$. The area inside solid-lined circles around BS 1,2, and 3 represent the associated transmission regions of each BS and the area inside dash-lined circles around BSs represent the associated interference regions of each BS.

express γ_l as

$$\gamma_l = \frac{p_l |\mathbf{h}_{ll}^{\mathrm{H}} \mathbf{v}_l|^2}{\sigma_l^2 + \sum\limits_{j \in \mathcal{L}(tran(l)), j \neq l} p_j |\mathbf{h}_{ll}^{\mathrm{H}} \mathbf{v}_j|^2 + \sum\limits_{i \in \mathcal{N}_{\mathrm{int}}(l)} z_{il}}, \tag{4.5}$$

where $\mathcal{N}_{\mathrm{int}}(l) \subseteq \mathcal{N} \setminus \{tran(l)\}$ is the set of out-of-cell interfering BSs that are located at a distance less than R_{int} to $rec(l)$. For example, in Figure 4.1 we have $\mathcal{N}_{\mathrm{int}}(9) = \{1\}$, $\mathcal{N}_{\mathrm{int}}(12) = \{2\}$, $\mathcal{N}_{\mathrm{int}}(6) = \{1,3\}$, and $\mathcal{N}_{\mathrm{int}}(l) = \emptyset$ for all $l \in \mathcal{L} \setminus \{6,9,12\}$. Finally, it is useful to define the set $\mathcal{L}_{\mathrm{int}}$ of data streams that are subject to out-of-cell interference, i.e., $\mathcal{L}_{\mathrm{int}} = \{l \mid l \in \mathcal{L}, \mathcal{N}_{\mathrm{int}}(l) \neq \emptyset\}$. For example, in Figure 4.1 we have $\mathcal{L}_{\mathrm{int}} = \{6,9,12\}$.

Let β_l be an arbitrary positive weight associated with lth data stream. We consider the case where all receivers are using *single-user detection* (i.e., a receiver decodes its intended signal by treating all other interfering signals as noise). Assuming that the power allocation is subject to a maximum power constraint $\sum_{l \in \mathcal{L}(n)} p_l ||\mathbf{v}_l||_2 \leq p_n^{\max}$ for

each BS $n \in \mathcal{N}$, the problem of WSRMax can be expressed as

$$\begin{aligned} \text{maximize} \quad & \sum_{n\in\mathcal{N}} \sum_{l\in\mathcal{L}(n)} \beta_l \ln\left(1+\frac{p_l|\mathbf{h}_{ll}^{\mathrm{H}}\mathbf{v}_l|^2}{\sigma_l^2+\sum\limits_{j\in\mathcal{L}(n),j\neq l} p_j|\mathbf{h}_{ll}^{\mathrm{H}}\mathbf{v}_j|^2+\sum\limits_{i\in\mathcal{N}_{\mathrm{int}}(l)} z_{il}}\right) \\ \text{subject to} \quad & z_{il}=\textstyle\sum_{j\in\mathcal{L}(i)} p_j|\mathbf{h}_{jl}^{\mathrm{H}}\mathbf{v}_j|^2,\ l\in\mathcal{L}_{\mathrm{int}},\ i\in\mathcal{N}_{\mathrm{int}}(l) \\ & \textstyle\sum_{l\in\mathcal{L}(n)} p_l||\mathbf{v}_l||_2^2 \le p_n^{\max},\ n\in\mathcal{N} \\ & ||\mathbf{v}_l||_2=1, \quad p_l\ge 0,\ l\in\mathcal{L}, \end{aligned} \tag{4.6}$$

where the variables are $(p_l, \mathbf{v}_l)_{l\in\mathcal{L}}$ and $(z_{il})_{l\in\mathcal{L}_{\mathrm{int}},i\in\mathcal{N}_{\mathrm{int}}(l)}$ and $\ln(\cdot)$ is the natural logarithm.

4.2 Problem Decomposition, Master Problem, and Subproblems

In this section, we present the main building blocks required to derive the distributed algorithm for problem (4.6), namely, the master problem and the subproblems. To do this, we first break problem (4.6) into a master problem and N subproblems (one for each BS), by treating out-of-cell interference powers $\{z_{il}\}_{l\in\mathcal{L}_{\mathrm{int}},i\in\mathcal{N}_{\mathrm{int}}(l)}$ as complicating variables. In the case of the master problem, a sequential convex approximation strategy to circumvent the difficulties due to the inherent nonconvexity of problem (4.6) is presented. In the case of the subproblem, we adopt the method originally proposed in [34, sec. 4.3], which is essentially based on SOCP and GP techniques.

4.2.1 Primal Decomposition

We start by first reformulating problem (4.6) as

$$\begin{aligned} \text{minimize} \quad & -\sum_{n\in\mathcal{N}} \sum_{l\in\mathcal{L}(n)} \beta_l \ln\left(1+\frac{p_l|\mathbf{h}_{ll}^{\mathrm{H}}\mathbf{v}_l|^2}{\sigma_l^2+\sum\limits_{j\in\mathcal{L}(n),j\neq l} p_j|\mathbf{h}_{ll}^{\mathrm{H}}\mathbf{v}_j|^2+\sum\limits_{i\in\mathcal{N}_{\mathrm{int}}(l)} z_{il}}\right) \\ \text{subject to} \quad & z_{il}\ge\textstyle\sum_{j\in\mathcal{L}(i)} p_j|\mathbf{h}_{jl}^{\mathrm{H}}\mathbf{v}_j|^2,\ l\in\mathcal{L}_{\mathrm{int}},\ i\in\mathcal{N}_{\mathrm{int}}(l) \\ & \textstyle\sum_{l\in\mathcal{L}(n)} p_l||\mathbf{v}_l||_2^2 \le p_n^{\max},\ n\in\mathcal{N} \\ & ||\mathbf{v}_l||_2=1, \quad p_l\ge 0,\ l\in\mathcal{L}, \end{aligned} \tag{4.7}$$

where the variables are $(p_l, \mathbf{v}_l)_{l\in\mathcal{L}}$ and $(z_{il})_{l\in\mathcal{L}_{\text{int}}, i\in\mathcal{N}_{\text{int}}(l)}$. Problem (4.6) and (4.7) are equivalent, since (1) function $\ln(\cdot)$ is increasing and (2) the objective function of problem (4.7) is increasing in z_{il}, and therefore the first set of constraints holds with equality at the optimal point.

Let $\mathcal{L}_{\text{int}}(n)$ denote the set of links for which base station n acts as an out-of-cell interferer. In particular, $\mathcal{L}_{\text{int}}(n) = \{l | l \in \mathcal{L}_{\text{int}}, n \in \mathcal{N}_{\text{int}}(l)\}$. By noting that the sets $\{(l,i) | l \in \mathcal{L}_{\text{int}}, i \in \mathcal{N}_{\text{int}}(l)\}$ and $\{(l,n) | n \in \mathcal{N}, l \in \mathcal{L}_{\text{int}}(n)\}$ are identical, we can rewrite the first inequality constraint of problem (4.7) as

$$z_{nl} \geq \textstyle\sum_{j\in\mathcal{L}(n)} p_j |\mathbf{h}_{jl}^{\text{H}} \mathbf{v}_j|^2, \ n \in \mathcal{N}, \ l \in \mathcal{L}_{\text{int}}(n). \tag{4.8}$$

Now we treat z_{nl} as complicating variables and use primal decomposition techniques to split problem (4.7) into a master problem and N subproblems (one for each BS). The master problem updates the complicating variables $(z_{nl})_{n\in\mathcal{N}, l\in\mathcal{L}_{\text{int}}(n)}$ to maximize the overall weighed sum rate [i.e., to maximize the objective of original problem (4.6)]. To express the master problem compactly, let us denote the vector $(z_{nl})_{n\in\mathcal{N}, l\in\mathcal{L}_{\text{int}}(n)}$ of out-of-cell interference components by $\mathbf{z}$. The master problem is given by

$$\begin{array}{ll} \text{minimize} & \sum_{n\in\mathcal{N}} f_n(\mathbf{z}) \\ \text{subject to} & \mathbf{z} \succeq \mathbf{0}, \end{array} \tag{4.9}$$

where the variable is $\mathbf{z}$ and $f_n(\mathbf{z})$ is the optimal value of the nth subproblem given by

$$\begin{array}{ll} \text{minimize} & -\displaystyle\sum_{l\in\mathcal{L}(n)} \beta_l \ln\left(1+\frac{p_l|\mathbf{h}_{ll}^{\text{H}}\mathbf{v}_l|^2}{\sigma_l^2 + \displaystyle\sum_{j\in\mathcal{L}(n), j\neq l} p_j|\mathbf{h}_{ll}^{\text{H}}\mathbf{v}_j|^2 + \sum_{i\in\mathcal{N}_{\text{int}}(l)} z_{il}}\right) \\ \text{subject to} & z_{nl} \geq \sum_{j\in\mathcal{L}(n)} p_j |\mathbf{h}_{jl}^{\text{H}}\mathbf{v}_j|^2, \ l \in \mathcal{L}_{\text{int}}(n) \\ & \sum_{l\in\mathcal{L}(n)} p_l ||\mathbf{v}_l||_2^2 \leq p_n^{\max} \\ & ||\mathbf{v}_l||_2 = 1, \quad p_l \geq 0, \ l \in \mathcal{L}(n), \end{array} \tag{4.10}$$

with variables $(p_l, \mathbf{v}_l)_{l\in\mathcal{L}(n)}$. To simplify the presentation, it is also useful to introduce the following equivalent reformulation of

problem (4.10):

$$\begin{aligned}
\text{minimize} \quad & -\textstyle\sum_{l\in\mathcal{L}(n)} \beta_l \ln(1+\gamma_l) \\
\text{subject to} \quad & \gamma_l \leq \frac{p_l |\mathbf{h}_{ll}^{\mathrm{H}} \mathbf{v}_l|^2}{\sigma_l^2 + \sum\limits_{j\in\mathcal{L}(n), j\neq l} p_j |\mathbf{h}_{ll}^{\mathrm{H}} \mathbf{v}_j|^2 + \sum\limits_{i\in\mathcal{N}_{\mathrm{int}}(l)} z_{il}}, \; l \in \mathcal{L}(n) \\
& z_{nl} \geq \textstyle\sum_{j\in\mathcal{L}(n)} p_j |\mathbf{h}_{jl}^{\mathrm{H}} \mathbf{v}_j|^2, \; l \in \mathcal{L}_{\mathrm{int}}(n) \\
& \textstyle\sum_{l\in\mathcal{L}(n)} p_l ||\mathbf{v}_l||_2^2 \leq p_n^{\max} \\
& ||\mathbf{v}_l||_2 = 1, \quad p_l \geq 0, \; l \in \mathcal{L}(n),
\end{aligned} \tag{4.11}$$

where the variable is $(p_l, \gamma_l, \mathbf{v}_l)_{l\in\mathcal{L}(n)}$. The equivalence of problem (4.10) and (4.11) follows since the objective function of problem (4.11) is decreasing in γ_l, and therefore the first set of constraints holds with equality at the optimal point.

4.2.2 Master Problem

Computing the objective value of the master problem (4.9) requires the solution of each subproblem (4.10), which is NP-hard [75]. Moreover, even if we would be able to *solve* the subproblems, we *cannot* apply standard subgradient methods to solve the master problem (4.9) since it is *not convex*. To address these difficulties, we present a method that solves successive approximated variants of the original master problem (4.9). Each approximated problem can be transformed into a convex problem by a change of variables. To solve the resulting convex problems, we use a subgradient method. It is important to note that, the approximations and variable transformations mentioned above are such that we can always rely on subproblems (4.10) (i.e., BS optimizations) to compute a subgradient. Details of the subproblem solution method are deferred to Section 4.2.3.

We start by approximating the objective function of problem (4.9) with an upper bound function, which in turn is used to obtain the approximation of the master problem. We refer to the resulting approximation as *the approximated master* problem. Next, we derive an equivalent convex form of the approximated master problem, followed by the subgradient methods to solve it.

4.2.2.1 Derivation of an Upper Bound Function for the Master Problem

The key idea is as follows: we first carry out partial minimization of problem (4.11) to yield an initial upper bound on $f_n(\mathbf{z})$.[4] Then the initial upper bound is further modified by using a well known monomial approximation, so that convex optimization techniques can be readily employed.

To simplify the presentation, let $\mathcal{H}$ denote the feasible set of problem (4.11). For some fixed normalized $\check{\mathbf{v}}_l$, let $\check{\mathcal{H}}((\check{\mathbf{v}}_l)_{l\in\mathcal{L}(n)}) = \big\{(p_l,\gamma_l)_{l\in\mathcal{L}(n)} \big| (p_l,\gamma_l,\check{\mathbf{v}}_l)_{l\in\mathcal{L}(n)} \in \mathcal{H}\big\}$. Now we can write the following relations:

$$f_n(\mathbf{z}) = \inf_{(p_l,\gamma_l,\mathbf{v}_l)_{l\in\mathcal{L}(n)}\in\mathcal{H}} -\textstyle\sum_{l\in\mathcal{L}(n)} \beta_l \ln(1+\gamma_l) \tag{4.12}$$

$$\leq \inf_{(p_l,\gamma_l)_{l\in\mathcal{L}(n)}\in\check{\mathcal{H}}((\check{\mathbf{v}}_l)_{l\in\mathcal{L}(n)})} -\textstyle\sum_{l\in\mathcal{L}(n)} \beta_l \ln(1+\gamma_l) \tag{4.13}$$

$$= \inf_{(p_l,\gamma_l)_{l\in\mathcal{L}(n)}\in\check{\mathcal{H}}((\check{\mathbf{v}}_l)_{l\in\mathcal{L}(n)})} \ln\big(\textstyle\prod_{l\in\mathcal{L}(n)}(1+\gamma_l)^{-\beta_l}\big) \tag{4.14}$$

$$\leq \inf_{(p_l,\gamma_l)_{l\in\mathcal{L}(n)}\in\check{\mathcal{H}}((\check{\mathbf{v}}_l)_{l\in\mathcal{L}(n)})} \ln\left(\prod_{l\in\mathcal{L}(n)}\left(\check{\gamma}_l^{-\frac{\check{\gamma}_l}{1+\check{\gamma}_l}}(1+\check{\gamma}_l)\ \gamma_l^{\frac{\check{\gamma}_l}{1+\check{\gamma}_l}}\right)^{-\beta_l}\right) \tag{4.15}$$

$$= \ln\Bigg(\underbrace{\inf_{(p_l,\gamma_l)_{l\in\mathcal{L}(n)}\in\check{\mathcal{H}}((\check{\mathbf{v}}_l)_{l\in\mathcal{L}(n)})} \prod_{l\in\mathcal{L}(n)}\left(\check{\gamma}_l^{-\frac{\check{\gamma}_l}{1+\check{\gamma}_l}}(1+\check{\gamma}_l)\ \gamma_l^{\frac{\check{\gamma}_l}{1+\check{\gamma}_l}}\right)^{-\beta_l}}_{\check{f}_n(\mathbf{z})}\Bigg) \tag{4.16}$$

$$= \ln\big(\check{f}_n(\mathbf{z})\big). \tag{4.17}$$

The first equality follows from the definition of $f_n(\mathbf{z})$ and from the equivalence of problems (4.10) and (4.11), (4.13) follows from partial minimization of the function over $(p_l,\gamma_l)_{l\in\mathcal{L}(n)}$ while $(\mathbf{v}_l)_{l\in\mathcal{L}(n)}$ being fixed such that $(\mathbf{v}_l = \check{\mathbf{v}}_l)_{l\in\mathcal{L}(n)}$, (4.14) follows trivially from the properties of $\ln(\cdot)$ function, (4.15) follows from the monomial lower bound on

[4] The minimum value of a function with respect to the all set of variables is always better than the minimum value of the function with respect to a subset of variables while others being fixed.

$1+\gamma_l$, i.e., $1+\gamma_l \geq \check{\gamma}_l^{-\frac{\check{\gamma}_l}{1+\check{\gamma}_l}}(1+\check{\gamma}_l)\gamma_l^{\frac{\check{\gamma}_l}{1+\check{\gamma}_l}}$, where $\check{\gamma}_l$ is an arbitrary positive number[5][134, lem. 1], (4.16) follows from the monotonic properties of $\ln(\cdot)$, and $\check{f}_n(\mathbf{z})$ is the optimal value of the following problem[6]:

$$\begin{aligned}
\text{minimize} \quad & \textstyle\prod_{l\in\mathcal{L}(n)}\left(\check{\gamma}_l^{-\check{\gamma}_l/(1+\check{\gamma}_l)}(1+\check{\gamma}_l)\right)^{-\beta_l}\prod_{l\in\mathcal{L}(n)}\gamma_l^{-\beta_l\frac{\check{\gamma}_l}{1+\check{\gamma}_l}} \\
\text{subject to} \quad & \gamma_l \leq \frac{p_l|\mathbf{h}_{ll}^{\mathrm{H}}\check{\mathbf{v}}_l|^2}{\sigma_l^2+\sum\limits_{j\in\mathcal{L}(n),j\neq l}p_j|\mathbf{h}_{ll}^{\mathrm{H}}\check{\mathbf{v}}_j|^2+\sum\limits_{i\in\mathcal{N}_{\mathrm{int}}(l)}z_{il}},\ l\in\mathcal{L}(n)\setminus\mathcal{L}_{\mathrm{local}}(n) \\
& \gamma_l \leq \frac{p_l|\mathbf{h}_{ll}^{\mathrm{H}}\check{\mathbf{v}}_l|^2}{\sigma_l^2+\sum\limits_{j\in\mathcal{L}(n),j\neq l}p_j|\mathbf{h}_{ll}^{\mathrm{H}}\check{\mathbf{v}}_j|^2},\quad l\in\mathcal{L}_{\mathrm{local}}(n) \\
& z_{nl} \geq \textstyle\sum_{j\in\mathcal{L}(n)}p_j|\mathbf{h}_{jl}^{\mathrm{H}}\check{\mathbf{v}}_j|^2,\ l\in\mathcal{L}_{\mathrm{int}}(n) \\
& \textstyle\sum_{l\in\mathcal{L}(n)}p_l \leq p_n^{\max} \\
& p_l \geq 0,\ l\in\mathcal{L}(n),
\end{aligned} \tag{4.18}$$

where the variable is $(p_l,\gamma_l)_{l\in\mathcal{L}(n)}$ and $\mathcal{L}_{\mathrm{local}}(n)$ is the subset of data streams transmitted by nth BS, which are not interfered by any out-of-cell interference, i.e., $\mathcal{L}_{\mathrm{local}}(n)=\{l\,|\,l\in\mathcal{L}(n),\mathcal{N}_{\mathrm{int}}(l)=\emptyset\}$. Note that, the inequality (4.13) holds with equality if the *optimal* normalized beamforming directions of problem (4.11) is identical to $(\check{\mathbf{v}}_l)_{l\in\mathcal{L}(n)}$ and the inequality (4.15) holds with equality if $(\gamma_l=\check{\gamma}_l)_{l\in\mathcal{L}(n)}$.

From (4.12)–(4.17) we have $f_n(\mathbf{z}) \leq \ln\left(\check{f}_n(\mathbf{z})\right)$, which holds for all $n\in\mathcal{N}$. Thus we have

$$\textstyle\sum_{n\in\mathcal{N}}f_n(\mathbf{z}) \leq \sum_{n\in\mathcal{N}}\ln\left(\check{f}_n(\mathbf{z})\right), \tag{4.19}$$

which gives an upper bound on the objective function of (4.9). The approximated master problem is obtained by replacing the objective function of the original master problem (4.9) by the upper bound function given in (4.19), i.e.,

$$\begin{aligned}
\text{minimize} \quad & \textstyle\sum_{n\in\mathcal{N}}\ln\left(\check{f}_n(\mathbf{z})\right) \\
\text{subject to} \quad & \mathbf{z}\succeq\mathbf{0},
\end{aligned} \tag{4.20}$$

[5] This bound is typically used in conjunction with an iterative method, which uses local approximations. The parameter $\check{\gamma}_l$ is usually the point at which the approximation is made.

[6] Here we have explicitly characterized the constraint $(p_l,\gamma_l)_{l\in\mathcal{L}(n)}\in\check{\mathcal{H}}((\check{\mathbf{v}}_l)_{l\in\mathcal{L}(n)})$.

where the variables is $\mathbf{z}$. Though Problem (4.20) is not convex in its current form, it can be equivalently reformulated into a convex problem via a variable transformation as shown in the next section.

4.2.2.2 Convex Reformulation of the Approximated Master Problem

Let us first transform problem (4.20) by the logarithmic change of variables $\bar{z}_{il} = \ln z_{il}$ (so $z_{il} = e^{\bar{z}_{il}}$). This yields the problem

$$\text{minimize} \quad \textstyle\sum_{n\in\mathcal{N}} \ln\left(\check{f}_n\left(e^{\bar{\mathbf{z}}}\right)\right), \tag{4.21}$$

where the variable is $\bar{\mathbf{z}} = (\bar{z}_{il})_{l\in\mathcal{L}_{\text{int}}, i\in\mathcal{N}_{\text{int}}(l)}$. Here we use the notation $e^{\mathbf{y}}$, where $\mathbf{y}$ is a vector, to mean componentwise exponentiation: $[e^{\mathbf{y}}]_k = e^{y_k}$.

Next we show that problem (4.21) is convex. To see this, we capitalize on perturbation and sensitivity analysis results for convex optimization problems [22, 42, 89].[7] In particular, we apply perturbation results to the convex form of GP (4.18). To do this, let us first perform the logarithmic change of variables $\bar{p}_l = \ln p_l$, $\bar{\gamma}_l = \ln \gamma_l$, logarithmic change of parameters $\bar{z}_{il} = \ln z_{il}$, and a logarithmic transformation of the objective and constraint functions of GP (4.18) to get its convex form:

$$\begin{aligned}
\text{minimize} \quad & \textstyle\sum_{l\in\mathcal{L}(n)} \frac{\beta_l \check{\gamma}_l}{1+\check{\gamma}_l}\, \bar{\gamma}_l + \ln\left(\prod_{l\in\mathcal{L}(n)} \left(\check{\gamma}_l^{-\frac{\check{\gamma}_l}{1+\check{\gamma}_l}} (1+\check{\gamma}_l)\right)^{-\beta_l}\right) \\
\text{subject to} \quad & \ln\left(g_{ll}^{-1} e^{\bar{\gamma}_l - \bar{p}_l}\left(\sigma_l^2 + \sum_{j\in\mathcal{L}(n), j\neq l} g_{jl} e^{\bar{p}_j} + \sum_{i\in\mathcal{N}_{\text{int}}(l)} e^{\bar{z}_{il}}\right)\right) \leq 0, \\
& \qquad\qquad l \in \mathcal{L}(n) \setminus \mathcal{L}_{\text{local}}(n) \\
& \ln\left(g_{ll}^{-1} e^{\bar{\gamma}_l - \bar{p}_l}\left(\sigma_l^2 + \textstyle\sum_{j\in\mathcal{L}(n), j\neq l} g_{jl} e^{\bar{p}_j}\right)\right) \leq 0,\ l \in \mathcal{L}_{\text{local}}(n) \\
& \ln\left(\textstyle\sum_{j\in\mathcal{L}(n)} g_{jl} e^{-\bar{z}_{nl}} e^{\bar{p}_j}\right) \leq 0, \quad l \in \mathcal{L}_{\text{int}}(n) \\
& \ln\left(\textstyle\sum_{l\in\mathcal{L}(n)} (p_n^{\max})^{-1} e^{\bar{p}_l}\right) \leq 0,
\end{aligned} \tag{4.22}$$

[7] Basic sensitivity results are documented in [22, sec. 5.6] and more general results can be found in [42, chap. 2] and [89, sec. 5.6].

where the variable is $(\bar{p}_l, \bar{\gamma}_l)_{l\in\mathcal{L}(n)}$ and $g_{jl} = |\mathbf{h}_{jl}^{\mathrm{H}}\check{\mathbf{v}}_j|^2$. Problem (4.22) possesses the following key features:

1. Since the optimal value of GP (4.18) is $\check{f}_n(\mathbf{z})$, the optimal value of problem (4.22) is given by $\ln(\check{f}_n(e^{\bar{\mathbf{z}}}))$.
2. Objective function of problem (4.22) is jointly convex in $(\bar{p}_l, \bar{\gamma}_l)_{l\in\mathcal{L}(n)}$ *and* $\bar{\mathbf{z}}$.
3. The constraint functions of problem (4.22) become jointly convex in $(\bar{p}_l, \bar{\gamma}_l)_{l\in\mathcal{L}(n)}$ *and* $\bar{\mathbf{z}}$.

By using the perturbation and sensitivity result given in [89, lem. 1] it follows that $\ln\left(\check{f}_n\left(e^{\bar{\mathbf{z}}}\right)\right)$ is convex in $\bar{\mathbf{z}}$. Consequently, problem (4.21) is convex.

4.2.2.3 Subgradient Method to Solve the Convex Form of the Approximated Master Problem

In this subsection, we derive the subgradient method for solving problem (4.21). By invoking [89, lem. 1], we can compute a subgradient of $\sum_{n\in\mathcal{N}}\ln\left(\check{f}_n\left(e^{\bar{\mathbf{z}}}\right)\right)$ at $\bar{\mathbf{z}}$. Specifically, a subgradient is given by $\sum_{n\in\mathcal{N}}(d_{il}^n(\bar{\mathbf{z}}))_{l\in\mathcal{L}_{\mathrm{int}},i\in\mathcal{N}_{\mathrm{int}}(l)}$ and

$$d_{il}^n(\bar{\mathbf{z}}) = \begin{cases} \dfrac{\lambda_l^\star(e^{\bar{\mathbf{z}}})e^{\bar{z}_{il}}}{\sigma_l^2 + \sum\limits_{j\in\mathcal{L}(n),j\neq l} g_{jl}e^{\bar{p}_j^\star(e^{\bar{\mathbf{z}}})} + \sum\limits_{m\in\mathcal{N}_{\mathrm{int}}(l)} e^{\bar{z}_{ml}}} & l\in\mathcal{L}(n)\setminus\mathcal{L}_{\mathrm{local}}(n),\\ & i\in\mathcal{N}_{\mathrm{int}}(l)\\ -\mu_l^\star(e^{\bar{\mathbf{z}}}) & l\in\mathcal{L}_{\mathrm{int}}(n),\ \ i=n\\ 0 & \text{otherwise,}\end{cases} \tag{4.23}$$

where $(\lambda_l^\star(e^{\bar{\mathbf{z}}}))_{l\in\mathcal{L}(n)\setminus\mathcal{L}_{\mathrm{local}}(n)}$ denotes the optimal Lagrange multipliers associated with the first set of constraints of problem (4.22), $(\mu_l^\star(e^{\bar{\mathbf{z}}}))_{l\in\mathcal{L}_{\mathrm{int}}(n)}$ denotes the optimal Lagrange multipliers associated with the third set of constraints of (4.22), and $(\bar{p}_l^\star(e^{\bar{\mathbf{z}}}), \bar{\gamma}_l^\star(e^{\bar{\mathbf{z}}}))_{l\in\mathcal{L}(n)}$ denotes the optimal solution of problem (4.22). Each BS n can compute $(d_{il}^n(\bar{\mathbf{z}}))_{l\in\mathcal{L}_{\mathrm{int}},i\in\mathcal{N}_{\mathrm{int}}(l)}$ independently, which in turn are used to construct the subgradient of $\sum_{n\in\mathcal{N}}\ln\left(\check{f}_n\left(e^{\bar{\mathbf{z}}}\right)\right)$ at $\bar{\mathbf{z}}$ via BS–BS coordination. Note that the *zero* in Equation (4.23) are used to simplify the presentation, and these zeros need *not* be exchanged between BSs during their coordinations.

The subgradient method for problem (4.21) is given by [19]

$$\bar{z}_{il}^{(j+1)} = \bar{z}_{il}^{(j)} - \theta^{(j)} \textstyle\sum_{n\in\mathcal{N}} d_{il}^{n}(\bar{\mathbf{z}}^{(j)}),\ l \in \mathcal{L}_{\text{int}},\ i \in \mathcal{N}_{\text{int}}(l) \tag{4.24}$$

$$= \bar{z}_{il}^{(j)} - \theta^{(j)} \left(d_{il}^{i}(\bar{\mathbf{z}}^{(j)}) + d_{il}^{tran(l)}(\bar{\mathbf{z}}^{(j)}) \right),\ l \in \mathcal{L}_{\text{int}},\ i \in \mathcal{N}_{\text{int}}(l). \tag{4.25}$$

where j is the current iteration index of the subgradient method and $\theta^{(j)} \in \mathbb{R}_{+}$ is a step size.[8] The second equality (4.25) follows from (4.23) after ignoring the zero elements. This suggest that, for computing the (i,l)th component of the subgradient, only *two* BSs [i.e., i and tran(l)] need to coordinate.

4.2.3 Subproblem: BS Optimization

Note that subproblem (4.11) is NP-hard [75], and therefore any solution method is reliant on approximations. The subproblem solution method presented in this section is essentially based on Algorithm 4.3.1 originally proposed in [34, sec. 4.3]. Here we briefly discuss the key idea of this algorithm for the sake of completeness.

The key idea of the algorithm is to carry out the optimization with respect to different subsets of variables by considering others fixed [34, sec. 4.3]. First, by fixing the beamformers $(\mathbf{v}_l)_{l\in\mathcal{L}(n)}$, a GP of the form (4.18) is solved which locally approximates the original subproblem (4.11). This is a decent step. Then, for fixed $(\gamma_l)_{l\in\mathcal{L}(n)}$ values, a maximum power reduction factor $t^\star$ is found such that the SINR values are preserved. The maximum power reduction factor is given by the optimum $t^\star$ that solves the following problem:

$$\begin{aligned}
\text{minimize} \quad & t \\
\text{subject to} \quad & \gamma_l \leq \frac{p_l |\mathbf{h}_{ll}^{\mathrm{H}} \mathbf{v}_l|^2}{\sigma_l^2 + \sum\limits_{j\in\mathcal{L}(n), j\neq l} p_j |\mathbf{h}_{ll}^{\mathrm{H}} \mathbf{v}_j|^2 + \sum\limits_{i\in\mathcal{N}_{\text{int}}(l)} z_{il}},\ l \in \mathcal{L}(n) \\
& t^2 z_{nl} \geq \textstyle\sum_{j\in\mathcal{L}(n)} p_j |\mathbf{h}_{jl}^{\mathrm{H}} \mathbf{v}_j|^2,\ l \in \mathcal{L}_{\text{int}}(n) \\
& \textstyle\sum_{l\in\mathcal{L}(n)} p_l ||\mathbf{v}_l||_2^2 \leq t^2 p_n^{\max} \\
& ||\mathbf{v}_l||_2 = 1,\ p_l \geq 0,\ l \in \mathcal{L}(n),
\end{aligned} \tag{4.26}$$

[8] We chose diminishing nonsummable step lengths (i.e., $\theta^{(j)} = 1/j$), that guarantees the asymptotic convergence of the subgradient method [19].

where the variables are t and $(p_l, \mathbf{v}_l)_{l\in\mathcal{L}(n)}$.[9] Note that, we always have $t^\star \leq 1$, and, hence, the saved power can be used to decrease the objective of original problem (4.11) by (1) setting $(\mathbf{v}_l = \mathbf{v}_l^\star)_{l\in\mathcal{L}(n)}$ and $(p_l = p_l^\star/t^\star)_{l\in\mathcal{L}(n)}$ and (2) increasing $(\gamma_l)_{l\in\mathcal{L}(n)}$ until the SINR constraints become tight. The result is again a descent step. The discussion above leads to the following descent algorithm which can be asynchronously solved by nth BS:

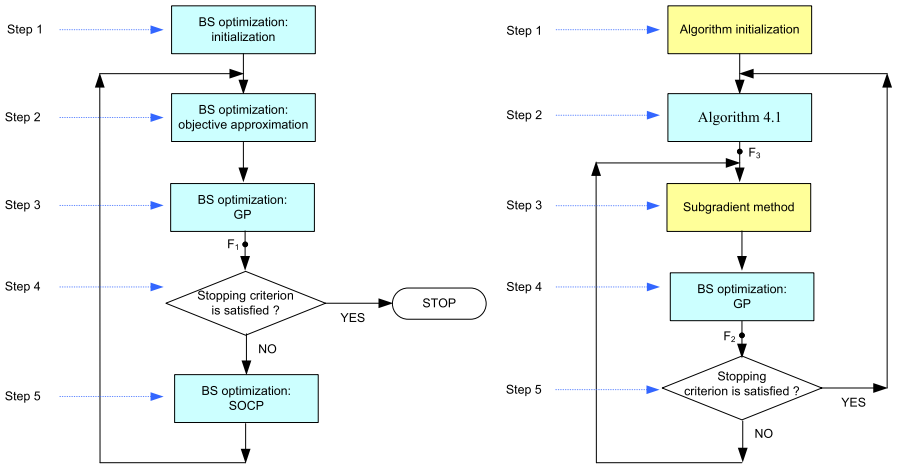
Algorithm 4.1 Finding a suboptimal solution for BS optimization problem (4.11) [34, sec. 4.3]

1. Initialization; given a feasible beamformer configuration $\left(\mathbf{v}_l^{(0)}\right)_{l\in\mathcal{L}(n)}$, a feasible power allocation $\left(p_l^{(0)}\right)_{l\in\mathcal{L}(n)}$, and $\mathbf{z}$. Set iteration index $i = 0$.
2. By setting $p_l = p_l^{(i)}$ and $\mathbf{v}_l = \mathbf{v}_l^{(i)}$, compute $\check{\gamma}_l$ for all $l \in \mathcal{L}(n)$ from (4.5).
3. By setting $\check{\mathbf{v}}_l = \mathbf{v}_l^{(i)}$ for all $l \in \mathcal{L}(n)$, solve problem (4.18). Denote the solution by $(p_l^\star, \gamma_l^\star)_{l\in\mathcal{L}(n)}$ and the optimal Lagrange multipliers by $(\lambda_l^\star)_{l\in\mathcal{L}(n)\setminus\mathcal{L}_{\text{local}}(n)}$ and $(\mu_l^\star)_{l\in\mathcal{L}_{\text{int}}(n)}$.
4. Stopping criterion; if the stopping criterion is satisfied STOP by returning $d_{il}^n(\cdot)$ by using (4.23) and the suboptimal solution $(\check{p}_l, \check{\gamma}_l, \check{\mathbf{v}}_l)_{l\in\mathcal{L}(n)}$, where $\check{p}_l = p_l^\star$. Otherwise, update achieved SINR values $\gamma_l^{\text{tmp}} = \gamma_l^\star$ for all $l \in \mathcal{L}(n)$.
5. By setting $\gamma_l = \gamma_l^{\text{tmp}}$ for all $l \in \mathcal{L}(n)$, solve problem (4.26). Denote the solution by $t^\star$ and $(p_l^\star, \mathbf{v}_l^\star)_{l\in\mathcal{L}(n)}$. Update $p_l^{(i+1)} = p_l^\star/(t^\star)^2$ and $\mathbf{v}_l^{(i+1)} = \mathbf{v}_l^\star$ for all $l \in \mathcal{L}(n)$. Set $i = i + 1$ and go to step 2.

The block diagram shown in Figure 4.2(a) summarizes Algorithm 4.1. It is a descent algorithm and we refer the reader to [34] for more details.

Note that, step 3 of Algorithm 4.1 solves problem (4.18) for some normalized $\check{\mathbf{v}}_l$. This is *the problem* that should be solved to find $d_{il}^n(\bar{\mathbf{z}})$ given in (4.23), which is then used to compute a subgradient $\sum_{n\in\mathcal{N}}(d_{il}^n(\bar{\mathbf{v}}))_{l\in\mathcal{L}_{\text{int}},i\in\mathcal{N}_{\text{int}}(l)}$ for the objective of the approximated

[9] It is well known that problem (4.26) is equivalently formulated as a SOCP (see [34, sec. 4.3])



(a) Algorithm 4.1 (§ 4.2.3) (b) Algorithm 4.2 (§ 4.3)

Fig. 4.2 Block diagrams of the algorithms.

master problem (4.21). The observations above suggest that the local BS optimizations (i.e., Algorithm 4.1) can be employed to compute the subgradient in a distributed fashion. Specifically, the dual variables and the optimal solutions required to compute the subgradient elements $d_{il}^n(\bar{\mathbf{z}})$ are obtained as a by-product of the BS optimization process. These are, of course, desirable and favorable features that are exploited in the distributed WSRMax algorithm discussed in Section 4.3.

4.3 Distributed Algorithm

In this section we blend (1) the subgradient method, which solves an approximation of the master problem (4.9) (see Section 4.2.2) and (2) Algorithm 4.1, which finds a suboptimal solution to subproblem (4.10) (see Section 4.2.3). The result is an algorithm, which solves a series of approximated variants of the original master problem (4.9) via a subgradient method. Subgradients for the subgradient method are computed by coordinating the subproblems or the BS optimizations.

The main skeleton of the distributed algorithm is depicted in Figure 4.2(b), which is a smooth integration of the subgradient

method (4.24) and Algorithm 4.1 in an iterative manner. The detailed algorithm is as follows (see Figure 4.2(b) for a concise block diagram).

Algorithm 4.2 Distributed algorithm for WSRMax

1. Initialization; given the globally agreed initial out-of-cell interference $\mathbf{z}$, a feasible beamformer configuration $\left(\mathbf{v}_l^{(0)}\right)_{l\in\mathcal{L}(n)}$, and a feasible power allocation $\left(p_l^{(0)}\right)_{l\in\mathcal{L}(n)}$. Set subgradient iteration index $j=0$.
2. for $n=1$ to N
   — perform Algorithm 4.1 and return the subgradient contribution $(d_{il}^n(\bar{\mathbf{z}}))_{l\in\mathcal{L}_{\text{int}},i\in\mathcal{N}_{\text{int}}(l)}$ and the suboptimal solution $(\check{p}_l,\check{\gamma}_l,\check{\mathbf{v}}_l)_{l\in\mathcal{L}(n)}$.
3. Set $(\bar{z}_{il}^{(j)}=\ln z_{il})_{l\in\mathcal{L}_{\text{int}},i\in\mathcal{N}_{\text{int}}(l)}$ and perform (4.24) to yield $(\bar{z}_{il}^{(j+1)})_{l\in\mathcal{L}_{\text{int}},i\in\mathcal{N}_{\text{int}}(l)}$ and set $\mathbf{z}=\left(e^{\bar{z}_{il}^{(j+1)}}\right)_{l\in\mathcal{L}_{\text{int}},i\in\mathcal{N}_{\text{int}}(l)}$.
4. for $n=1$ to N
   — solve problem (4.18). Denote the solution by $(p_l^\star(\mathbf{z}),\gamma_l^\star(\mathbf{z}))_{l\in\mathcal{L}(n)}$ and the optimal Lagrange multipliers by $(\lambda_l^\star(e^{\bar{\mathbf{z}}}))_{l\in\mathcal{L}(n)\setminus\mathcal{L}_{\text{local}}(n)}$ and $(\mu_l^\star(e^{\bar{\mathbf{z}}}))_{l\in\mathcal{L}_{\text{int}}(n)}$.
   — Compute $d_{il}^n(\bar{\mathbf{z}})$ by using (4.23).
5. Stopping criterion; if the stopping criterion is satisfied, reset subgradient iteration index j, i.e., $j=0$, set $\left(\mathbf{v}_l^{(0)}=\check{\mathbf{v}}_l\right)_{l\in\mathcal{L}(n)}$, $\left(p_l^{(0)}=p_l^\star(\mathbf{z})\right)_{l\in\mathcal{L}(n)}$, and go to step 2. Otherwise increment subgradient iteration index j, i.e., $j=j+1$ and go to step 3.

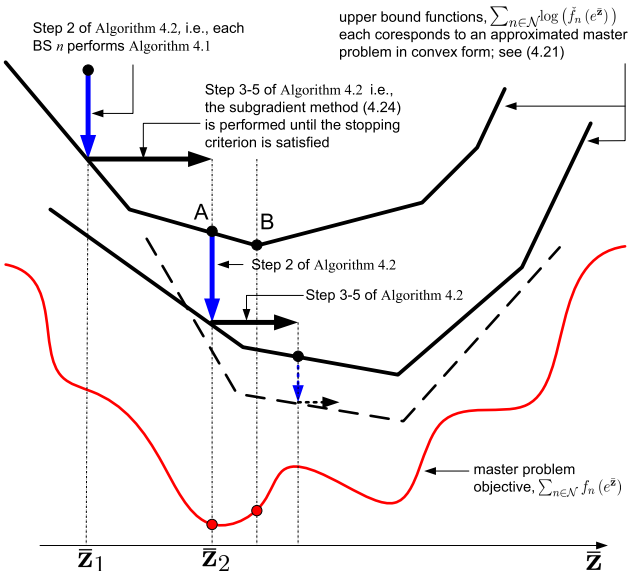
The first step initializes Algorithm 4.2. Steps 2 represent the BS optimizations that are performed asynchronously in a decentralized fashion by each BS for fixed out-of-cell interference $\mathbf{z}$. BS optimizations terminate after the per BS stopping criterion is satisfied; see step 4 of Algorithm 4.1. At this stage each BS has its own solution and the subgradient part $(d_{il}^n(\bar{\mathbf{z}}))_{l\in\mathcal{L}_{\text{int}},i\in\mathcal{N}_{\text{int}}(l)}$. BS coordination is initiated at step 3. For example each BSs coordinate to construct a subgradient $\sum_{n\in\mathcal{N}}(d_{il}^n(\bar{\mathbf{z}}))_{l\in\mathcal{L}_{\text{int}},i\in\mathcal{N}_{\text{int}}(l)}$ and perform subgradient method (4.24). This updates global out-of-cell interference variable $\mathbf{z}$. At step 4, each

BS performs their own GP to compute $(d_{il}^n(\bar{\mathbf{z}}))_{l\in\mathcal{L}_{\text{int}},i\in\mathcal{N}_{\text{int}}(l)}$ for the next subgradient iteration. Step 5, is the stopping criterion for the subgradient method. If the stopping criterion is satisfied, Algorithm switches back to BS optimizations, i.e., step 2. Otherwise, the subgradient method is performed until the stopping criterion is satisfied. The algorithm continues in an iterative manner.

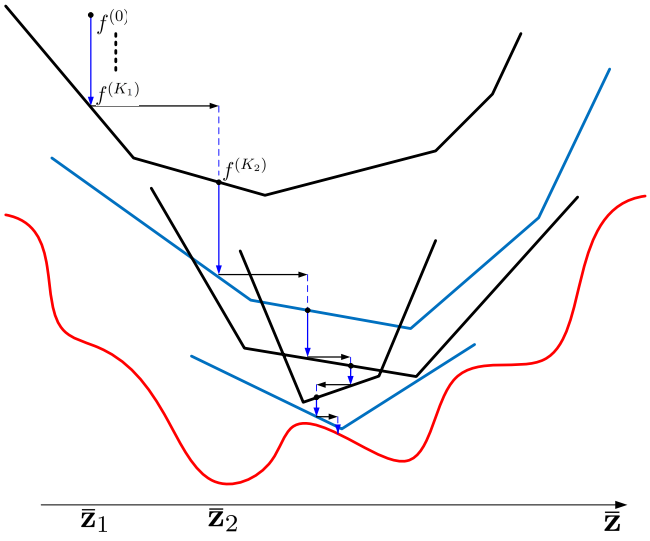
Figure 4.3(a) depicts graphically the behavior of Algorithm 4.2. The nonconvex curve is the objective function of the master problem (4.9) after the logarithmic change of variables $\bar{z}_{il} = \ln z_{il}$. The convex curves are the objective functions of approximated master problems of the form (4.21), which are essentially parameterized by the current beamforming directions. The vertical arrows correspond to asynchronous per BS optimizations, i.e., step 2 depicted in Figure 4.2(b). The horizontal arrows correspond to the subgradient method, i.e., step 3–5 depicted in Figure 4.2(b). Figure 4.3(a) shows that the algorithm switches between the per BS optimizations and the subgradient method. For example, by fixing out-of-cell interference at $\bar{\mathbf{z}}_1$, the algorithm performs per BS optimizations. Once a specified stopping criterion is satisfied, the algorithm stops BS optimizations and performs the subgradient method until a specified stopping criterion is satisfied. As a result, the out-of-cell interference values are changed from $\bar{\mathbf{z}}_1$ to $\bar{\mathbf{z}}_2$. The algorithm continues in an iterative manner.

The algorithm presented in this section has following features, which simplify its practical implementation:

1. *Asynchronism*: All the subproblems or BS optimizations can be carried out in a fully asynchronous fashion until a stopping criterion is satisfied.
2. *Fast Local optimization*: Each subsystem need to solve convex problems, which can be performed very fast provided the significant computing power available at each BS.
3. *Thin protocol*: Each BS does *not* need to reveal the entirety of its own subproblem during the BS coordination; only a little communication is needed, and therefore the protocol between BSs can be very light.



(a) Upper bounds



(b) monotonic convergence

Fig. 4.3 The behavior of Algorithm 4.2; the objective function of problem (4.9) and (4.21) are shown in the domain of $\bar{\mathbf{z}}$.

4. *Reliability*: To carry out the algorithm, only BS to BS synchronized signalling is required. This signalling can be carried out via reliable backhaul communication links such as microwave and fibre links.
5. *No user terminal involvement*: The user terminals do not require performing any processing associated with algorithm iterations and user to BS signalling is not required.

4.3.1 Monotonic Convergence of Algorithm 4.2

In this section we first show that Algorithm 4.2 can generate a monotonically nonincreasing sequence of objective values, with appropriate choice of stopping criteria. In particular, we measure the objective value given by the algorithm just after each GP; see point "F_1" of Figure 4.2(a) and point "F_2" of Figure 4.2(b). Then we show the monotonic convergence of Algorithm 4.2.

Algorithm 4.2 starts with Algorithm 4.1 (see step 2). Let $f^{(0)}, f^{(1)}, \ldots, f^{(K_1)}$ denote the sequence of objective values obtained during Algorithm 4.1 iterations. Here K_1 is the number of Algorithm 4.1 iterations until its stopping criterion is satisfied. Natural stopping criteria includes (1) running Algorithm 4.1 for a fixed number of iterations or (2) running Algorithm 4.1 until the objective value decrement between two successive iterations is below a certain predefined threshold. Since Algorithm 4.1 contains nonascent steps (see Section 4.2.3) we have

$$f^{(0)} \geq f^{(1)} \geq \cdots \geq f^{(K_1)}, \tag{4.27}$$

as depicted in Figure 4.3(b).

Next, Algorithm 4.2 switches to the subgradient method (4.24) (see step 3). Note that, the subgradient method is not a descent algorithm. Therefore, in order to obtain a monotonically nonincreasing sequence of objective values, we consider the following stopping criterion: running subgradient method until an objective value $f^{(K_2)}$ is achieved, such that $f^{(K_1)} \geq f^{(K_2)}$ [see Figure 4.3(b)], where K_2 is the number of

subgradient iterations.[10] Thus, we have

$$f^{(0)} \geq f^{(1)} \geq \ldots \geq f^{(K_1)} \geq f^{(K_2)}. \tag{4.28}$$

The switching between Algorithm 4.1 and the subgradient method is done in an iterative manner. The result is a monotonically nonincreasing sequence of objective values $f^{(0)}, f^{(1)}, f^{(2)}, \ldots$ such that $f^{(i)} \geq f^{(i+1)}$, $i = 0, 1, 2, \ldots$. Moreover, note that the optimal objective value of problem (4.7) is *bounded*. This guarantees the monotonic convergence of Algorithm 4.2 to an equilibrium point [103, thm. 3.14].

Note that the development of Algorithm 4.2 is not based on Karush-Kuhn-Tucker (KKT) optimality conditions for the nonconvex problem (4.7). As a result, characterizing completely the solution structure of the algorithm is a difficult task. For example, the (suboptimal) solution after the convergence of Algorithm 4.2 may not necessarily be a locally optimal point of problem (4.7).

4.3.2 A Practical Stopping Criterion/Signalling Strategy

The stopping criteria discussed in Section 4.3.1 are, of course, important to ensure the monotonic convergence of the algorithm. However, it is desirable to seek for stopping criteria, which are favorable for practical implementations of the algorithm, but with a violation of the monotonic convergence.

In the sequel, such an example strategy is explained. The key idea is to define time barriers; i.e., system checkpoints at which all BS must start their local optimizations (i.e., Algorithm 4.1) and system checkpoints at which all BS start coordination (i.e., the subgradient method). In particular, each BS transmissions are synchronized and the data transmission phase of each BS is preceded by a signalling phase, in which the rate/power allocation of each BS is determined via WSR-Max; see Figure 4.4. The signalling phase consists of three types of time

[10] In fact, the subgradient method, with diminishing nonsummable step lengths, ensures asymptotic convergence [19]. However, the requirement here is to iterate until a better objective value (compared to the initial objective value $f^{(K_1)}$) is obtained.

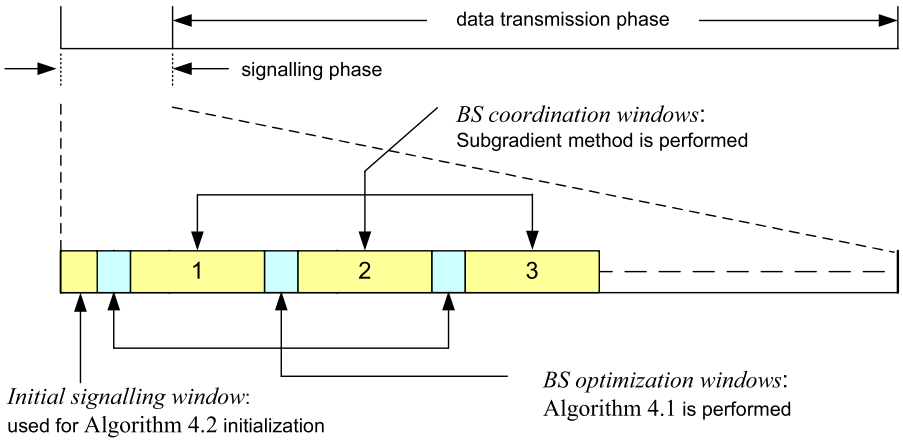


Fig. 4.4 An example signalling frame structure.

slots called *initial signalling window*, *BS optimization window*, and *BS coordination window*. The initial signalling window is used for step 1 of Algorithm 4.2, i.e., the initialization step. The latter two types of windows (i.e., BS optimization window and BS coordination window) are repeated until the data transmission phase is reached as shown in Figure 4.4. We define the BS optimization windows to be the time periods where Algorithm 4.1 is performed asynchronously. Therefore, during BS optimization windows, step 2 of Algorithm 4.2 is carried out. The width of the window is determined by the maximum number of Algorithm 4.1 iterations. The BS coordination windows are defined to be the time periods where the subgradient method is performed. Therefore, during any BS coordination window, step 3, step 4, and step 5 of Algorithm 4.2 are carried out repeatedly. The width of the BS coordination window is determined by the maximum number of subgradient iterations. Typically, we may assume that the time period of any BS optimization window is significantly *smaller* compared to the time period of any BS coordination window because of the following reasons: (1) significant computing power available at BSs so that the BS optimization can be performed very fast, (2) BS coordination require backhaul message exchanges between BSs, which in turn demand stringent time requirements.

4.4 Numerical Examples

In this section we run Algorithm 4.2 (Section 4.3) in multiuser multicell environments and the benefits due to different degrees of BS coordination are numerically evaluated. As benchmarks, we consider three algorithms: (1) distributed WMMSE algorithm [110], (2) GP-SOCP based centralized algorithm proposed in [34, sec. 4.3], and (3) the distributed algorithm proposed in [10, 11], which is based on a ZF beamforming strategy. To emphasize the practical relevance of the algorithm, we consider only the stopping criterion discussed in Section 4.3.2, which is based on time barriers or system checkpoints as shown in Figure 4.4.

We consider an exponential path loss model, where the channel gains between BSs and users are given by

$$\mathbf{h}_{ij} = \sqrt{\left(\frac{d_{ij}}{d_0}\right)^{-\eta}}\, \mathbf{c}_{ij}, \tag{4.29}$$

where d_{ij} is the distance from the transmitter of ith data stream to the receiver of jth data stream, d_0 is the *far field reference distance* [64], η is the path loss exponent, and $\mathbf{c}_{ij} \in \mathbb{C}^T$ such that $\mathbf{c}_{ij} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$ (i.e., frequency-flat fading with uncorrelated antennas). The first term of (4.29) represents the path loss factor and the second term models the Rayleigh small-scale fading. An arbitrarily generated set $\mathcal{C}$ of fading coefficients where $\mathcal{C} = \{\mathbf{c}_{ij} \mid \ i, j \in \mathcal{L}\}$ is referred to as a *single fading realization*. The variance of the noise is considered equal for all data streams, i.e., $\sigma_l^2 = N_0$ for all $l \in \mathcal{L}$ and the maximum power constraint is assumed the same for all nodes, i.e., $p_n^{\max} = p_0^{\max}$ for all $n \in \mathcal{N}$. We define the signal-to-noise ratio (or SNR) operating point at a distance d [distance units] as

$$\mathrm{SNR}(d) = \begin{cases} \dfrac{p_0^{\max}}{N_0} & d \le d_0 \\ \dfrac{p_0^{\max}}{N_0}\left(\dfrac{d}{d_0}\right)^{-\eta} & \text{otherwise.} \end{cases} \tag{4.30}$$

In all the simulations we set $d_0 = 1$, $\eta = 4$, $p_0^{\max}/N_0 = 45$ dB, $\mathrm{SNR}(R_{\mathrm{int}}) = 0$ dB, where R_{int} is the radius of the interference regions of

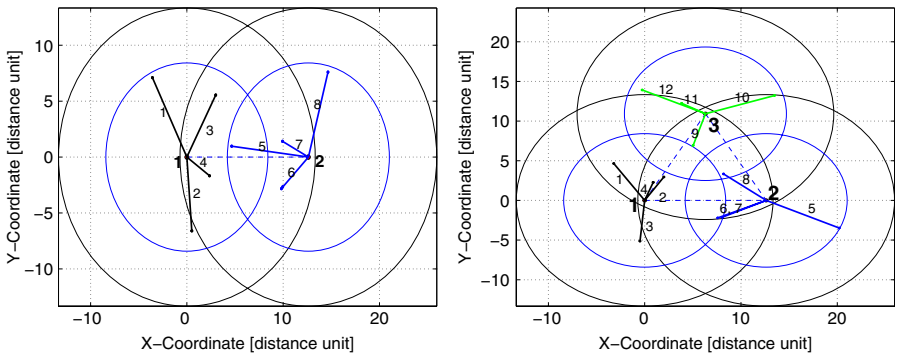


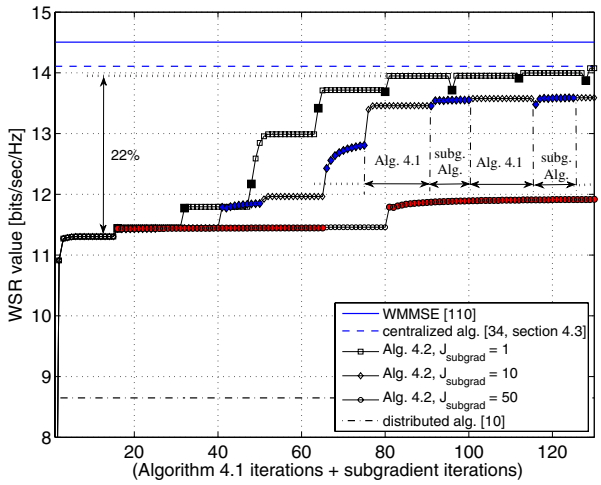
Fig. 4.5 (a) Multicell network 1, $\mathcal{N}=\{1,2\}$, $\mathcal{L}=\{1,\ldots,8\}$, $\mathcal{L}(1)=\{1,\ldots,4\}$, $\mathcal{L}(2)=\{5,\ldots,8\}$, $\mathcal{L}_{\text{int}}=\{3,\ldots,7\}$; (b) Multicell network 2, $\mathcal{N}=\{1,2,3\}$, $\mathcal{L}=\{1,\ldots,12\}$, $\mathcal{L}(1)=\{1,\ldots,4\}$, $\mathcal{L}(2)=\{5,\ldots,8\}$, $\mathcal{L}(3)=\{9,\ldots,12\}$, $\mathcal{L}_{\text{int}}=\{1,2,4,6,\ldots,11\}$.

each BS,[11] and $\text{SNR}(R_{\text{BS}})=8$ dB, where R_{BS} is the radius of the transmission regions of each BS.

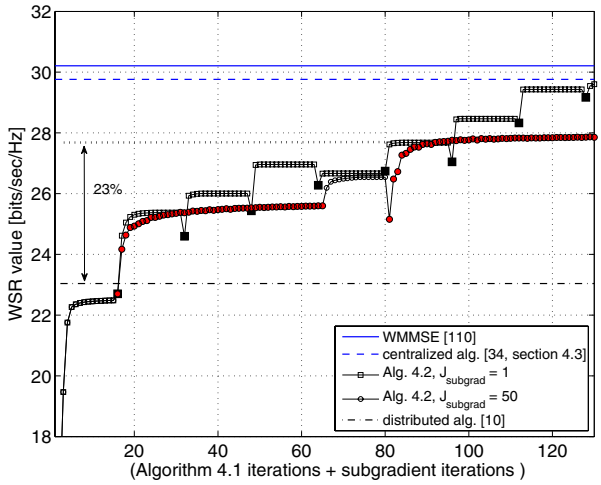
In the simulations two multicell multiuser wireless cellular networks as shown in Figure 4.5 are considered. In the case of first network [i.e., Figure 4.5(a)], there are $N=2$ BSs with $T=4$ antennas at each one. The BSs are located such that the distance between the two BSs is $D_{\text{BS}}=1.5\times R_{\text{BS}}$. In the case of second network [i.e., Figure 4.5(b)], there are $N=3$ BSs with $T=4$ antennas at each one. Moreover, the BSs are located such that they form an equilateral triangle and the distance between any two BSs is $D_{\text{BS}}=1.5\times R_{\text{BS}}$. There are four users per each BS located inside the transmission region of the BS. The locations of users associated with BSs are arbitrarily chosen as shown in Figure 4.5. A single data stream is transmitted for each user.

To see the behavior of Algorithm 4.2, we first consider a nonfading case and run the algorithm in both networks shown in Figure 4.5. Figure 4.6 shows the objective value of problem (4.6) computed at points "F_1" and "F_2" [see Figures 4.2(a) and 4.2(b)] for arbitrary generated fading realizations. Here the X-axis of Figure 4.6 represents

[11] Signal strength of BS's transmitted signal at a distance R_{int} is at most on the order of noise, Therefore, as we modeled in Section 4.1, it is reasonable to consider that the interference cased by the BS outside the interference region is negligible.



(a)



(b)

Fig. 4.6 Objective value versus GP iteration: (a) Multicell network 1; (b) Multicell network 2.

combined Algorithm 4.1 iterations and subgradient iterations. For simplicity, we denote the number of Algorithm 4.1 iterations carried out during the BS optimization window by $J_{\mathrm{BS-opt}}$ and denote the number of subgradient iterations performed during the BS coordination window by J_{subgrad}.

Plots are drawn for the cases of $J_{\text{BS}-\text{opt}} = 15$ and $J_{\text{subgrad}} = 1, 10, 50$. Note that J_{subgrad} is a measure of the degree of BS coordination. For example, $J_{\text{subgrad}} = 1$ means that the subgradient method is performed only once during any BS coordination window and $J_{\text{subgrad}} = 50$ means that the subgradient method is carried out 50 consecutive times during any BS coordination window. Weights β_l of each data stream is arbitrarily chosen from the interval $(0, 1]$. In step 1 of Algorithm 4.2, the components of initial out-of-cell interference vector $\mathbf{z}$ are chosen on the order of noise variance N_0 (e.g., 0.5 N_0). Moreover, the normalized initial beamformers $(\mathbf{v}_l^{(0)})_{l \in \mathcal{L}(n)}$ are randomly generated and a feasible uniform initial beamformer power allocation is chosen, i.e., $(p_l^{(0)} = \alpha p_0^{\max}/T)_{l \in \mathcal{L}(n)}$, where $\alpha \in (0, 1]$ is chosen to ensures the feasibility of problem (4.18).

In order to describe the algorithm's behavior, let us first focus to Figure 4.6(a), the case of $J_{\text{subgrad}} = 1$. To distinguish Algorithm 4.1 iterations from the subgradient iterations, we use two types of squares; transparent squares and solid squares. Specifically, the transparent squares correspond to the Algorithm 4.1 iterations and the solid squares correspond to the subgradient iterations. Since $J_{\text{subgrad}} = 1$, only a *single* subgradient iteration is performed during any BS coordination window. Furthermore, each BS perform 15 Algorithm 4.1 iterations during any BS optimization window, since we have $J_{\text{BS}-\text{opt}} = 15$. Note that the BS optimizations (Algorithm 4.1) are always nondecreasing steps.[12] The flattening of these line segments means that BS optimizations cannot further improve the system objective. Violation of overall monotonic behavior is inevitable since the subgradient method is not a descent algorithm in general [19]. Results show that BS coordination can gracefully resolve the out-of-cell interference (i.e., $\mathbf{z}$) via subgradient method. For example, the plot in the case of $J_{\text{subgrad}} = 1$, shows a 22% increase in the weighted sum-rate (WSR), after having 5 subgradient iterations.

Figure 4.6(a) further shows that the value of J_{subgrad}, which parameterizes the degree of BS coordination has a significant effect on the

[12] Nondecreasing because we have plotted the positive weighted sum-rate value instead of the negative value of it.

overall WSR value. It is interesting to note that, a smaller number of *consecutive* subgradient iterations (e.g., $J_{\text{subgrad}} = 1, 10$) can perform better compared to a larger number of consecutive subgradient iterations (e.g., $J_{\text{subgrad}} = 50$). Such a behavior is very important in practice to reduce significantly the backhaul message exchanges during any BS coordination window. We can intuitively explain the behavior by considering the two points "A" and "B" in Figure 4.3(a). In particular, point "A" corresponds to a smaller J_{subgrad}, where the (convex form) approximated master problem (4.21) is solved to a low accuracy. Point "B" corresponds to a larger J_{subgrad}, where the (convex form) approximated master problem is solved to a high accuracy. Of course, point "B" is better than point "A" for the *approximated* master problem, but not necessarily for the original master problem (4.9); see the master objective depicted in Figure 4.3(a). This suggest that one need not solve each approximation to a high accuracy. Refining the approximation more often (which corresponds to a smaller J_{subgrad}), rather than solving some approximated master problem to a high accuracy (which corresponds to a larger J_{subgrad}) is more beneficial.

Figure 4.6(b) shows the algorithm behavior in the case of network setup 2 in Figure 4.5(b). The behavior is very similar to the previous plots in Figure 4.6(a). The network can yield substantial gains by performing just one subgradient iterations during any BS coordination window, that is, less backhaul message exchanges between BSs. For example, the plot in the case of $J_{\text{subgrad}} = 1$, shows a 23% increase in the WSR, after having 5 subgradient iterations; see Figure 4.6(a). Figure 4.6 also shows the performance of the considered benchmark algorithms *after* their convergence. In both networks, for the considered channel realizations, the performance of the distributed algorithm in [10] is significantly low. It is intuitively clear that the performance drop is due to the lack of degrees of freedom available at BS transmissions to avoid interference. Results further show that the distributed WMMSE algorithm outperforms Algorithm 4.2 in both scenarios. Such results are intuitively expected because WMMSE algorithm *do* rely on user terminal assistance during algorithm's iterations compared to Algorithm 4.2. The good performance of the centralized algorithm compared to Algorithm 4.2 agrees with the intuition that methods

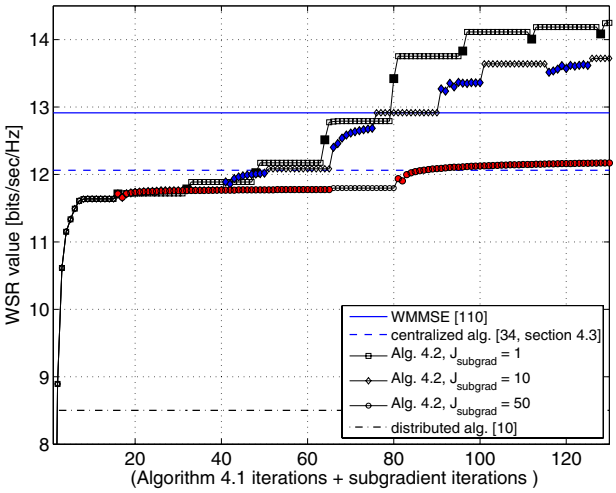
with a centralized controller can always outperform decentralized methods.

It is important to note, however, that all the considered algorithms are suboptimal methods to problem (4.6), and therefore their optimality is not guaranteed. As a result, they may experience different performance ranking for different channel realizations. One such case is illustrated in Figure 4.7. The algorithms' parameters are same as in Figure 4.6 except the fading realizations. Results show that Algorithm 4.2 can outperforms WMMSE and the centralized algorithms.

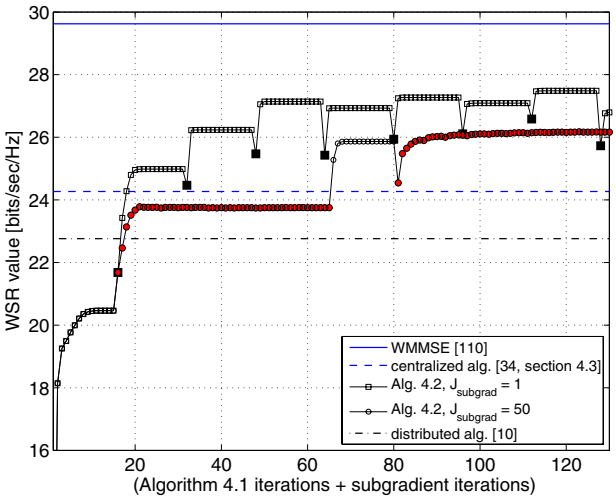
In order to see the average behavior of the algorithm, we consider a fading case. Here, we run Algorithm 4.2 for 500 fading realization with $J_{\text{subgrad}} = 1$ and $J_{\text{BS}-\text{opt}} = 15$. Recall that the algorithm parameter $J_{\text{subgrad}} = 1$ means that during any BS coordination window only one subgradient iteration is performed. These are the only operations that require message exchanging between BSs via backhaul links. Moreover, subgradient iterations are the main implementation-level bottleneck, provided significant computing power at BSs, where Algorithm 4.1 iterations can be performed fast and efficiently. Thus, it is interesting to see the average WSR value of problem (4.6) achieved at point 'F_3' of Algorithm 4.2 [see Figure 4.2(b)] after m $(= 0, 1, \ldots)$ subgradient iteration. In other words, we examine the evolution of average WSR versus the number of BS coordinations.

Figure 4.8 shows the dependence of the average WSR value on the number of subgradient iterations in the case of considered network 1 and network 2. Note that, we have used the same figure to plot the dependence of the average objective value of WMMSE algorithm on the number of iterations.[13] Results show that the BS coordination plays a critical role in the performance of Algorithm 4.2. For clarity, we denote the situation where the subgradient iterations $= 0$ as *noncoordinating* case. In the case of network 1 [see Figure 4.8(a)], more than 12% improvement in the average objective value is achieved within five BS

[13] The subgradient iterations are analogous to WMMSE iterations in the following sense: both the subgradient iterations and the WMMSE algorithm iterations require message exchanges between nodes. Specifically, the subgradient method requires BS–BS message exchanges and WMMSE requires BS-user terminal as well as user terminal-BS message exchanges.



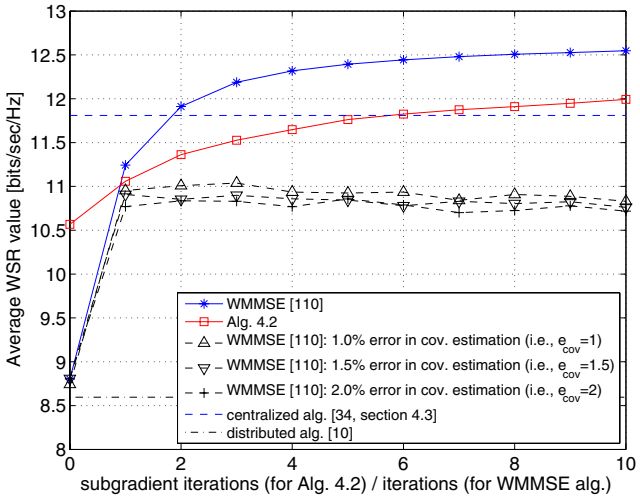
(a)



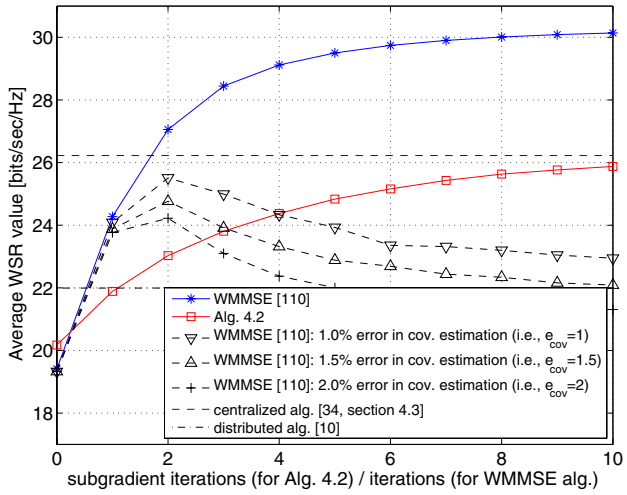
(b)

Fig. 4.7 Objective value versus GP iteration: (a) Multicell network 1; (b) Multicell network 2.

coordinations compared to the noncoordinating case. For network 2 [see Figure 4.8(b)], within five BS coordinations, more than 24% improvement in the average objective value is achieved as compared to the noncoordinating case.



(a)



(b)

Fig. 4.8 Average objective value versus number of BS coordinations: (a) Multicell network 1; (b) Multicell network 2.

Figure 4.8 also shows that the average performance of WMMSE algorithm is better compared to that of Algorithm 4.2. This behavior is intuitively expected since, unlike Algorithm 4.2, the WMMSE algorithm benefits from user terminal assistance. Recall that, during each

iterations, WMMSE algorithm requires user terminals assistance such as signal covariance estimations, computations, and feedback information to BSs over the air interface. In contrast, the given method require only BS-level synchronized communication and all the necessary computation is concentrated at the BSs. The result is naturally a trade off between performance gains and the implementation-level simplicity. For a fair comparison of Algorithm 4.2 and WMMSE algorithm, we examine the sensitivity of WMMSE algorithm to imperfections on the signal covariance estimations at user terminals. Specifically, during each WMMSE iteration, we randomly perturb the error free signal covariance matrix J_l (which is a scalar in the case of MISO) at each user terminal l as follows: $J_l = J_l + J_l(xe_{\text{cov}}/100)$, where x is a random variable with 2 equiprobable outcomes $-1, 1$ and e_{cov} is the amount of covariance perturbation. Results show that such small estimation errors have a significant effect on the performance of WMMSE algorithm. Moreover, in such situations, the convergence of the WMMSE method becomes less predictable. Thus, the algorithm presented is well suited for systems where the user terminal assistance is not desirable due to potential errors such as estimation errors and feedback errors.

Figure 4.8 further shows that, the performance of Algorithm 4.2 within several BS coordinations is comparable with that of the centralized algorithm [34, sec. 4.3]. For example, in the case of network 1, Algorithm 4.2 achieves around 99% of the average WSR value given by the centralized algorithm [34, sec. 4.3]. Moreover, in the case of network 2, Algorithm 4.2 yields around 94% of the average WSR value given by the considered centralized algorithm. Finally, we see that there is a substantial performance gap between Algorithm 4.2 and the distributed algorithm in [10]. The main reason for such a performance drop of algorithm in [10] is the insufficient degree of freedom available at BS transmissions to cancel the interference it causes to the user terminals.

4.5 Summary and Discussion

The weighted sum-rate maximization problem in a multicell multiple-input and single-output downlink system was considered. The problem

is nonconvex; in fact it is NP-hard. A distributed solution method for the problem was presented. The main advantage of the given distributed algorithm is its implementation-level simplicity. Unlike the minimum weighted mean-squared error based algorithms, the method presented does not demand user terminal assistance during each iteration. It essentially requires base station to base station (BS) communication, which are reasonably realizable, provided reliable backhaul links (e.g., fibre and microwave links) and significant computing power at BSs. As a result, a good trade-off between the performance gains and the implementation-level simplicity was achieved. The algorithm was based on primal decomposition and subgradient methods. In particular, the main problem was split into a master problem and many subproblems (one for each base station). A sequential convex approximation strategy together with a subgradient method were blent to address the nonconvex master problem. Master problem solution relies on synchronous BS coordinations. A descent algorithm based on second-order cone programming and a geometric programming were adopted in the case of subproblems. The subproblems can be performed in a fully *asynchronous* manner. The monotonic convergence of the algorithm was established, with appropriate choice of stopping criteria at intermediate steps. Practical stopping criteria were also presented. Numerical experiments were presented to compare the method with other state-of-the-art algorithms. Results suggest that the algorithm is well suited for systems where the user terminal assistance is not allowed or not desirable. Results further showed that the algorithm could significantly improve the overall system performance with a small amount of BS coordinations. These observations are indeed important for deriving simple signalling protocols in the context of large-scale practical cellular communication systems.

5

Conclusions

In this volume a greater emphasis was placed on the general weighted sum-rate maximization (WSRMax) problem, which is NP-hard; it plays an important role in various problems of recent interest in wireless communication, including network utility maximization, cross-layer design, link scheduling, and many others.

The first section was mainly intended to highlight the motivation for the WSRMax problem and to review earlier and parallel work. In Section 2, a global optimization approach based on the branch and bound technique was presented to solve the nonconvex WSRMax problem with an optimality certificate. Efficient analytic bounding methods were given and their effect on the convergence of the branch and bound algorithm was analyzed numerically. Though the convergence speed was dramatically increased by improving the lower bound, the benefits of improving the upper bounding methods are imperceptible. This suggests that a grater emphasis should be placed on exploring better lower bounding techniques. Unlike other branch and bound based algorithms for WSRMax, the method presented does not rely on the convertibility of the problem into a DC (difference of convex functions) problem. Therefore, the method applies to a broader class of WSRMax problems

(e.g., WSRMax in multicast wireless networks). More importantly, it can address *any* system performance metric that can be expressed as a Lipschitz continuous and increasing function of signal-to-interference-plus-noise ratio (SINR) values (not restricted to WSRMax).

The worst case complexity of the branch and bound algorithm increases exponentially with the problem size for a given accuracy. Even a small-scale problems (e.g., one with few tens of variables) can take a long time, and therefore such global methods are applicable when the computing time requirements are not critical. This means that, in the case of wireless networks, where the computing time is often crucial, any global optimization method is hardly applicable. However, it is indeed useful to provide performance benchmarks, e.g., for evaluating the performance loss encountered by any suboptimal method for WSRMax.

The link-interference model considered in Section 2 is very general; it accommodates a wide variety of network topologies. Moreover, the considered link-interference model supports different node capabilities, including single- or multipacket transmission, single- or multipacket reception, and many others. Numerical examples of diverse application domains of WSRMax were presented to demonstrate the branch and bound algorithm.

Fast, suboptimal algorithms for the WSRMax problem in general multicommodity, multichannel wireless networks were considered in Section 3. The algorithms were used within a general cross-layer utility maximization framework and the quantitative impact of gains that can be achieved at the network layer in terms of end-to-end rates and network congestion was numerically evaluated.

First, a general access operation with a relatively simple form of receivers structure (a bank of match filters) was considered; a receiver decodes each of its intended signals by treating all other interfering signals as noise. The presented algorithms were based on complementary geometric programming and homotopy methods. The algorithm based on homotopy methods handled the self-interference problem without combinatorial constraints to enforce simultaneous transmissions and receptions in the same frequency band; thus the combinatorial nature of the problem has been circumvented. Numerical results showed that

the algorithms could exploit the multichannel diversity via dynamic power allocation across the available channels. It was interesting to see numerically that they performed closely to the exponentially complex optimal branch and bound algorithm. The methods also provide a mechanism, based on which the gains achievable at the network layer could be evaluated when the network nodes employ self interference cancelation techniques with different degrees of accuracy. Numerical results showed that quantifiable gains at the network layer have been achieved after a certain level of the self interference cancelation accuracy, e.g., a self interference reduction in the range $20-60$ dB led to significant gains at the network layer.

The latter part of Section 3 considered a case where all receivers perform multiuser detection and the gains that can be achieved at the network layer were numerically evaluated. The solution methods here were obtained by imposing additional constraints, such as that only one node can transmit to others at a time or that only one node can receive from others at a time. The main benefit of such constraints was the problem tractability and simplicity of the solution methods. Consequently, these simple access protocols can be potentially useful in practice with more advanced communication systems.

Note that all the algorithms presented in Section 3 are reliant on the well known interior-point methods. Therefore, the algorithms are fast, compared to the optimal branch and bound algorithm presented in Section 2 and they can be deployed in relatively large-scale problems (e.g., one with few hundreds of variables), which are not, of course, apparently handled by exponentially complex optimal algorithms. However, to facilitate the use of the algorithms in real-time applications further improvements in the efficiency are required. There are many important issues and methods in linear algebra that can be used to improve the efficiency of the algorithms. For example, the structure of the problem can be heavily exploited. We may assume in a wireless network that, the problem variables (e.g., power and SINRs) are not fully coupled due to exponential path losses. Such assumptions, which are reasonable in practice, can result desirable sparsity patterns of the involved problem data (e.g., sparse matrices), and can dramatically increase the efficiency of the methods.

An easy to implement distributed method for WSRMax problem in a multicell multiple antenna downlink system was presented in Section 4. Unlike the recently proposed minimum weighted mean-squared error based algorithms, where at each iteration all mobile terminals needs to estimate the covariance matrices of their received signals, compute and feedback over the air certain parameters to the base stations (BS), the presented algorithm operates without any user terminal assistance. It requires only BS to BS signalling via reliable backhaul links (e.g. fiber, microwave links) and all required computation is performed at the BSs. The algorithm is based on primal decomposition and subgradient methods, where the original nonconvex problem is split into a master problem and a number of subproblems (one for each BS). A sequential convex approximation strategy was used to address the nonconvex master problem. In the case of subproblems, an existing iterative approach based on second-order cone programming and geometric programming was adopted. The subproblems were coordinated to find a (possibly suboptimal) solution to the master problem. Subproblems can be solved by BSs in a fully asynchronous manner, though the coordination between subproblems should be synchronous. Numerical results were provided to see the behavior of the algorithm under different degrees of BS coordination. They showed that the algorithm could yield a good tradeoff between the implementation-level simplicity and the performance.

A

Improving Upper Bound for Branch and Bound via Complementary Geometric Programming

We show in this appendix how to compute efficiently $\boldsymbol{\gamma}_{\mathrm{ImpCGP}}$ via CGP [6], when $f_0(\boldsymbol{\gamma}) = \sum_{l\in\mathcal{L}} -\beta_l \log(1+\gamma_l)$. Note that this is the only place where the exact expression of the rate function (2.1) has been explicitly taken into account. In the derivation of all other bounds only the monotonicity property has been used. We start by equivalently reformulating problem (2.48) as

$$\begin{array}{ll} \text{minimize} & \prod_{l\in\mathcal{L}} \left(1+\gamma_l\right)^{-\beta_l} \\ \text{subject to} & \gamma_{l,\min} \leq \gamma_l \leq \gamma_{l,\max},\ l\in\mathcal{L} \\ & \gamma_l \leq \dfrac{g_{ll}p_l}{\sigma^2 + \sum_{j\neq l} g_{jl}p_j},\ l\in\mathcal{L} \\ & \sum_{l\in\mathcal{O}(n)} p_l \leq p_n^{\max},\ n\in\mathcal{T} \\ & p_l \geq 0,\ l\in\mathcal{L}, \end{array} \tag{A.1}$$

where the variables are $(p_l)_{l\in\mathcal{L}}$ and $(\gamma_l)_{l\in\mathcal{L}}$. The equivalence between problem (2.48) and problem (A.1) follows from the monotonically increasing property of $\log(\cdot)$ function and the explicit description of the

constraints. To obtain a suboptimal solution, we adopt Algorithm 3.2 (see Section 3.2.2) in a straightforward manner as follows:

Algorithm A.0.1 CGP based algorithm for finding $\boldsymbol{\gamma}_{\mathrm{ImpCGP}}$

---

1. Given tolerance $\varepsilon > 0$. Let $\hat{\boldsymbol{\gamma}} = \bar{\mathbf{a}}_{l^\star}$.
2. Solve the following GP:

$$\begin{array}{ll} \text{minimize} & \prod_{l\in\mathcal{L}} \gamma_l^{-\beta_l \frac{\hat{\gamma}_l}{1+\hat{\gamma}_l}} \\ \text{subject to} & \gamma_{l,\min} \le \gamma_l \le \gamma_{l,\max},\ l \in \mathcal{L} \\ & \gamma_l \le \dfrac{g_{ll}p_l}{\sigma^2 + \sum_{j\neq l} g_{jl}p_j},\ l \in \mathcal{L} \\ & \sum_{l\in\mathcal{O}(n)} p_l \le p_n^{\max},\ n \in \mathcal{T}, \end{array} \tag{A.2}$$

   with the variables $(p_l)_{l\in\mathcal{L}}$ and $(\gamma_l)_{l\in\mathcal{L}}$. Denote the solution by $(p_l^\star)_{l\in\mathcal{L}}$ and $(\gamma_l^\star)_{l\in\mathcal{L}}$.
3. If $\max_{l\in\mathcal{L}} |\gamma_l^\star - \hat{\gamma}_l| > \varepsilon$ set $(\hat{\gamma}_l = \gamma_l^\star)_{l\in\mathcal{L}}$ and go to step 2. Otherwise set $\boldsymbol{\gamma}_{\mathrm{ImpCGP}} = \hat{\boldsymbol{\gamma}}$ and STOP.

B

The Barrier Method

In this appendix we outline the basic steps involved in solving problem (3.33) using the barrier method [22, sec. 11.3.1]. For the sake of notational simplicity let us define $\bar{\sigma}_{\rho_n(k)} = \sigma^2/g_{\rho_n(k)\rho_n(k)}$ for $k = 1, \ldots, |\mathcal{O}(n)|$ and $\bar{\sigma}_{\rho_n(|\mathcal{O}(n)|+1)} = 0$. Furthermore, let $\mathbf{u}_i$ be the ith column of the upper triangular matrix $\mathbf{U} \in \mathbb{R}_+^{|\mathcal{O}(n)|\times|\mathcal{O}(n)|}$ with all nonzero entries being equal to 1.

By denoting the feasible set of rate allocation vector $\mathbf{r}_n = (r_{\rho_n(1)} \ldots r_{\rho_n(|\mathcal{O}(n)|)})$ [125, sec. 3], problem (3.33) can be equivalently expressed as

$$\begin{array}{ll} \text{maximize} & \sum_{l\in\mathcal{O}(n)} \beta_l r_l \\ \text{subject to} & u_i(\mathbf{r}_n) \leq 0, \quad i = 1, \ldots, |\mathcal{O}(n)| + 1, \end{array} \tag{B.1}$$

where the variable is $\mathbf{r}_n$. The function $u_i(\mathbf{r}_n)$ can be compactly expressed as

$$u_i(\mathbf{r}_n) = \begin{cases} -\mathbf{e}_i^{\mathrm{T}}\mathbf{r}_n & 1 \leq i \leq |\mathcal{O}(n)| \\ \displaystyle\sum_{j=1}^{|\mathcal{O}(n)|} b_n(j) e^{\mathbf{u}_j^{\mathrm{T}}\mathbf{r}_n} - \bar{\sigma}_{\rho_n(1)} - p_n^{\max} & i = |\mathcal{O}(n)| + 1, \end{cases}$$

where $b_n(j) = \bar{\sigma}_{\rho_n(j)} - \bar{\sigma}_{\rho_n(j+1)}$. Problem (B.1) is a convex optimization problem [22], and therefore can be solved efficiently. It is worth noting that, given any feasible $\mathbf{r}_n$ the corresponding power variables $p_{\rho_n(k)}$, $k = 1, \ldots, |\mathcal{O}(n)|$ are given by [125]

$$p_{\rho_n(k)} = (e^{r_{\rho_n(k)}} - 1) \sum_{i \geq k} b_n(i) e^{\sum_{k<j\leq i} r_{\rho_n(j)}}.$$

The barrier method [22, sec. 11.3.1] can be used to solve problem (B.1). The gradient and the Hessian of the function $u_i(\mathbf{r}_n)$ are given by

$$\nabla u_i(\mathbf{r}_n) = \begin{cases} -\mathbf{e}_i & 1 \leq i \leq |\mathcal{O}(n)| \\ \sum_{j=1}^{|\mathcal{O}(n)|} b_n(j) e^{\mathbf{u}_j^{\mathrm{T}} \mathbf{r}_n} \mathbf{u}_j & i = |\mathcal{O}(n)| + 1 \end{cases}$$

and

$$\nabla^2 u_i(\mathbf{r}_n) = \begin{cases} \mathbf{0} & 1 \leq i \leq |\mathcal{O}(n)| \\ \sum_{j=1}^{|\mathcal{O}(n)|} b_n(j) e^{\mathbf{u}_j^{\mathrm{T}} \mathbf{r}_n} \mathbf{u}_j \mathbf{u}_j^{\mathrm{T}} & i = |\mathcal{O}(n)| + 1. \end{cases}$$

The expressions above are used to evaluate the gradient and the Hessian of the logarithmic barrier function [22, sec. 11.2.1].

Acknowledgments

The work of P.C. Weeraddana and C. Fischione was supported by the Swedish Research Council and EU projects Hydrobionets and Hycon2. Most of the results have been obtained during Dr. Weeraddan's doctoral studies at the University of Oulu, Finland. M. Codreanu and M. Latva-aho were supported by the Finnish Funding Agency for Technology and Innovation (Tekes), Academy of Finland, Nokia, Nokia Siemens Networks, Renesas Mobile Europe, and Elektrobit. A. Ephremides was supported by the NSF grant CCF-0728966, CCF-0905209, and US Army Research Office grant W911NF-08-1-0238.

Notations and Acronyms

$\mathrm{E}\{\cdot\}$	Expectation
$\mathbf{e}_i$	ith standard unit vector
$\mathbf{I}$	Identity matrix; size of the matrix is implicit
$\mathbf{1}$	Vector of 1s, i.e., $(1,\ldots,1)$; size of the vector is implicit
$\lvert x\rvert$	Absolute value of the complex number x
$\lvert\lvert\mathbf{x}\rvert\rvert_2$	ℓ_2-norm of complex vector $\mathbf{x}$
$\mathbf{X}^{\mathrm{H}}$	Conjugate transpose (Hermtian) of matrix $\mathbf{X}$
$\mathbf{X}^{\mathrm{T}}$	Transpose of matrix $\mathbf{X}$
$\mathbf{X}^{-1}$	Inverse of matrix $\mathbf{X}$
$[\mathbf{X}]_{i,j}$	Element at the ith row and the jth column of matrix $\mathbf{X}$
$\lvert\mathcal{X}\rvert$	Cardinality of set $\mathcal{X}$
$\nabla f(\mathbf{x})$	Gradient of function f at $\mathbf{x}$
$\nabla^2 f(\mathbf{x})$	Hessian matrix of function f at $\mathbf{x}$
$\mathbb{C}$	Set of complex numbers
$\mathbb{C}^n$	Set of complex n-vectors
$\mathbb{C}^{m\times n}$	Set of complex $m \times n$ matrices

$\mathbb{R}$,$\mathbb{R}_+$,$\mathbb{R}_{++}$	Set of real, and nonnegative real numbers
$\mathbb{R}_+^n$	Cone of nonnegative, real n-vectors
$\mathbb{R}_+^{m\times n}$	Set of nonnegative, real $m \times n$ matrices
$\mathcal{CN}(\mathbf{m},\mathbf{K})$	Circularly symmetric complex Gaussian vector distribution with mean $\mathbf{m}$ and covariance matrix $\mathbf{K}$
$\sim$	Distributed according to
$\geq$	Greater than or equal operator; between real matrices, it represents componentwise inequality
CGP	Complementary Geometric Programming
GP	Geometric Programming
MWM	MaxWeight Matching
NUM	Network Utility Maximization
RA	Resource Allocation
SINR	Signal-To-Interference-Plus-Noise Ratio
SNR	Signal-To-Noise Ratio
SOCP	Second-Order Cone Programming
SP	Signomial Programming
WSR	Weighted Sum-Rate
WSRMax	Weighted Sum-Rate Maximization

References

[1] I. C. Abou-Faycal, M. D. Trott, and S. Shamai, "The capacity of discrete-time memoryless Rayleigh-fading channels," *IEEE Transaction on Information Theory*, vol. 47, no. 4, pp. 1290–1301, May 2001.

[2] R. Agarwal and J. M. Cioffi, "Beamforming design for the MIMO downlink for maximizing weighted sum-rate," in *Proceedings on IEEE International Symposium on Information Theory and its Applications*, pp. 1–6, Auckland, New Zealand, December 7–10 2008.

[3] H. Al-Shatri and T. Weber, "Optimizing power allocation in interference channels using D.C. programming," in *Proceedings on Workshop on Resource Allocation in Wireless Networking*, pp. 367–373, Avignon, France, June 4 2010.

[4] E. L. Allgower and K. Georg, *Introduction to Numerical Continuation Methods*. Philadelphia, PA: SIAM, 2003.

[5] C. Audet, P. Hansen, and G. Savard, *Essays and Surveys in Global Optimization*. 233 Spring street, NY 10013, USA: Springer Science + Business Media, Inc, 2005.

[6] M. Avriel and A. C. Williams, "Complementary geometric programming," *SIAM Journal on Appllied Mathematics*, vol. 19, no. 1, pp. 125–141, July 1970.

[7] C. Bae and D. Cho, "Fairness-aware adaptive resource allocation scheme in multihop OFDMA systems," *IEEE Communications Letters*, vol. 11, no. 2, pp. 134–136, February 2007.

[8] V. Balakrishnan, S. Boyd, and S. Balemi, "Branch and bound algorithm for computing the minimum stability degree of parameter-dependent linear systems," *International Journal on Robust and Nonlinear Control*, vol. 1, no. 4, pp. 295–317, October–December 1991.

[9] D. E. Barreto and S. S. Chiu, "Decomposition methods for cross-layer optimization in wireless networks," in *Proceedings on IEEE Wireless Communications and Networking Conference*, pp. 270–275, Kawloon, Hong Kong, March 11–15 2007.

[10] E. Björnson, M. Bengtsson, and B. Ottersten, "Optimality properties and low-complexity solutions to coordinated multicell transmission," in *Proceedings on IEEE Global Telecommunications Confernce*, pp. 1–6, Miami, Florida, USA, December 6–10 2010.

[11] E. Björnson, N. Jaldén, M. Bengtsson, and B. Ottersten, "Optimality properties, distributed strategies, and measurement-based evaluation of coordinated multicell OFDMA transmission," *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 6086–6101, December 2011.

[12] E. Björnson, R. Zakhour, D. Gesbert, and B. Ottersten, "Cooperative multicell precoding: Rate region characterization and distributed strategies with instantaneous and statistical CSI," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4298–4310, August 2010.

[13] S. A. Borbash and A. Ephremides, "The feasibility of matchings in a wireless network," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2749–2755, June 2006.

[14] S. A. Borbash and A. Ephremides, "Wireless link scheduling with power control and SINR constraints," *IEEE Transactions on Information Theory*, vol. 52, no. 11, pp. 5106–5111, November 2006.

[15] S. Boyd, "GGPLAB: A simple matlab toolbox for geometric programming," [Online]. Available: http://www.stanford.edu/~boyd/ggplab/, 2006.

[16] S. Boyd, "Branch-and-bound methods," [Online]. Available: http://www.stanford.edu/class/ee364b/lectures/bb_slides.pdf, 2007.

[17] S. Boyd, "Primal and dual decomposition," [Online]. Available: http://www.stanford.edu/class/ee364b/lectures/decomposition_slides.pdf, 2007.

[18] S. Boyd, "Sequential convex programming," [Online]. Available: http://www.stanford.edu/class/ee364b/lectures/seq_slides.pdf, 2007.

[19] S. Boyd, "Subgradient methods," [Online]. Available: http://www.stanford.edu/class/ee364b/lectures/subgrad_method_slides.pdf, 2007.

[20] S. Boyd, S. J. Kim, L. Vandenberghe, and A. Hassibi, "A tutorial on geometric programming," *Optimization and Engineering*, vol. 8, no. 1, pp. 67–127, 2007.

[21] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[22] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.

[23] S. Boyd and L. Vandenberghe, "Additional exercises for convex optimization," [Online]. Available: http://www.stanford.edu/~boyd/cvxbook/bv_cvxbook_extra_exercises.pdf, May 2010.

[24] L. Bui, A. Eryilmaz, R. Srikant, and X. Wu, "Asynchronous congestion control in multi-hop wireless networks with maximal matching-based scheduling," *IEEE/ACM Transactions on Networking*, vol. 16, no. 4, pp. 826–839, August 2008.

[25] M. Cao, V. Raghunathan, S. Hanly, V. Sharma, and P. R. Kumar, "Power control and transmission scheduling for network utility maximization in wireless networks," in *Proceedings on IEEE International Conference on Decision and Control*, pp. 5215–5221, New Orleans, LA, USA, December 12–14 2007.

[26] M. Cao, X. Wang, S. Kim, and M. Madihian, "Multi-hop wireless backhaul networks: A cross-layer design paradigm," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 4, pp. 738–748, May 2007.

[27] R. Cendrillon, J. Huang, M. Chiang, and M. Moonen, "Autonomous spectrum balancing for digital subscriber lines," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4241–4257, August 2007.

[28] R. Cendrillon, W. Yu, M. Moonen, J. Verlinden, and T. Bostoen, "Optimal multiuser spectrum balancing for digital subscriber lines," *IEEE Transactions on Communications*, vol. 54, no. 5, pp. 922–933, May 2006.

[29] M. Chiang, "Balancing transport and physical layers in wireless multihop networks: Jointly optimal congestion control and power control," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 104–116, January 2005.

[30] M. Chiang, "Geometric programming for communication systems," vol. 2, no. 1-2, pp. 1–154, July 2005.

[31] M. Chiang, C. W. Tan, D. P. Palomar, D. O'Neill, and D. Julian, "Power control by geometric programming," *IEEE Transactions on Wireless Communications*, vol. 6, no. 7, pp. 2640–2651, July 2007.

[32] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti, "Achieving single channel, full duplex wireless communication," in *Proceedings on ACM International Conference on Mobile Computing and Networking*, pp. 1–12, Chicago, Illinois, USA, September 20–24 2010.

[33] S. S. Christensen, R. Agarwal, E. Carvalho, and J. Cioffi, "Weighted sum-rate maximization using weighted MMSE for MIMO-BC beamforming design," *IEEE Transactions on Wireless Communications*, vol. 7, no. 12, pp. 4792–4799, December 2008.

[34] M. Codreanu, "Multidimensional adaptive radio links for broadband communications," Ph.D. thesis, Centre for Wireless Communications, University of Oulu, Oulu, Finland: Acta Universitatis Ouluensis, [Online]. Available: http://herkules.oulu.fi/isbn9789514286223, November 2007.

[35] M. Codreanu, A. Tölli, M. Juntti, and M. Latva-aho, "Joint design of Tx-Rx beamformers in MIMO downlink channel," *IEEE Transactions on Signal Processing*, vol. 55, no. 9, pp. 4639–4655, September 2007.

[36] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, USA: John Wiley, second Edition, 2006.

[37] R. L. Cruz and A. V. Santhanam, "Optimal routing, link scheduling and power control in multi-hop wireless networks," in *Proceedings on IEEE INFOCOM*, vol. 1, pp. 702–711, San Diego, CA, April 2003.

[38] T. ElBatt and A. Ephremides, "Joint scheduling and power control for wireless ad hoc networks," *IEEE Transactions on Wireless Communications*, vol. 3, no. 1, pp. 74–85, January 2004.

[39] K. Eriksson, "Dynamic resource allocation in wireless networks," Examensarbete utfört i Kommunikationssystem vid Tekniska högskolan i Linköping, [Online]. Available: http://liu.diva-portal.org/smash/get/diva2:326290/FULLTEXT01, 2010.

[40] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control," *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, pp. 1333–1344, December 2007.

[41] A. Eryilmaz, R. Srikant, and J. R. Perkins, "Stable scheduling policies for fading wireless channels," *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, pp. 411–424, April 2005.

[42] A. V. Fiacco, *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, vol. 165. 111 Fifth Avenue, New York, New York 10003: Academic Press, Inc., 1983.

[43] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends in Networks*, vol. 1, no. 1, pp. 1–144, April 2006.

[44] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, March 2000.

[45] C. Guthy, W. Utschick, R. Hunger, and M. Joham, "Efficient weighted sum rate maximization with linear precoding," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2284–2297, April 2010.

[46] B. Hajek and G. Sasaki, "Link scheduling in polynomial time," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 910–917, September 1988.

[47] D. Halperin, T. Anderson, and D. Wetherall, "Taking the sting out of carrier sense: Interference cancellation for wireless LANs," in *Proceedings on ACM International conference on Mobile Computing and Networking*, pp. 339–350, San Francisco, California, USA, September 14–19 2008.

[48] S. V. Hanly and D. N. C. Tse, "Multiaccess fading channels — Part II: Delay-limited capacities," *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 2816–2831, November 1998.

[49] Z. K. M. Ho and D. Gesbert, "Balancing egoism and altruism on the MIMO interference channel," *IEEE Journal on Selected Areas in Communications, submitted*, 2010, [Online]. Available: http://arxiv.org/PS_cache/arxiv/pdf/0910/0910.1688v1.pdf.

[50] L. M. C. Hoo, B. Halder, J. Tellado, and M. Cioffi, "Multi user transmit optimization for multiuser broadcast channels: Asymptotic fdma capacity region and algorithms," *IEEE Transactions on Communications*, vol. 52, no. 6, pp. 922–930, June 2004.

[51] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, UK: Cambridge University Press, 1994.

[52] R. Horst, P. Pardolos, and N. Thoai, *Introdiction to Global Optimization*, vol. 48. Dordrecht, Boston, London: Kluwer Academic Publishers, second Edition, 2000.

[53] J. Huang, R. Berry, and M. Honig, "Distributed interference compensation for wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 5, pp. 1074–1084, May 2006.

[54] J. Jang and K. B. Lee, "Transmit power adaptation for multiuser OFDM systems," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 2, pp. 171–178, February 2003.

[55] E. A. Jorswieck and E. G. Larsson, "The MISO interference channel from a game-theoretic perspective: A combination of selfishness and altruism achieves pareto optimality," in *Proceedings on IEEE International Conference on Acoustics, Speech, Signal Processing*, pp. 5364–5367, Las Vegas, Nevada, USA, March 31–April 4 2008.

[56] E. A. Jorswieck, E. G. Larsson, and D. Danev, "Complete characterization of the pareto boundary for the MISO interference channel," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 5292–5296, October 2008.

[57] S. Joshi, P. C. Weeraddana, M. Codreanu, and M. Latva-aho, "Weighted Sum-Rate Maximization for MISO Downlink Cellular Networks via Branch and Bound," *IEEE Transactions on Signal Processing*, vol. 60, no. 4, pp. 2090–2095, April 2012.

[58] D. Julian, M. Chiang, D. ONeil, and S. Boyd, "QoS and fairness constrained convex optimization of resource allocation for wireless cellular and ad-hoc networks," in *Proceedings on IEEE INFOCOM*, vol. 2, pp. 477–486, New York, USA, June 2002.

[59] F. P. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, pp. 33–37, 1997.

[60] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness, and stability," *Journal of the Operations Research Society*, vol. 49, pp. 237–252, 1998.

[61] T. S. Kim, Y. Yang, J. C. Hou, and S. V. Krishnamurthy, "Joint resource allocation and admission control in wireless mesh networks," in *Proceedings on Int. Symp. on Modelling and Opt. in Mobile, Ad-hoc and Wireless Networks*, pp. 1–10, Seoul, Korea, June 23–27 2009.

[62] M. Kodialam and T. Nandagopal, "Characterizing achievable rates in multi-hop wireless mesh networks with orthogonal channels," *IEEE/ACM Transactions on Networking*, vol. 13, no. 4, pp. 868–880, August 2005.

[63] M. Kojima, N. Megiddo, and T. Noma, "Homotopy continuation methods for nonlinear complementarity problems," *Mathematics of Operational Research*, vol. 16, pp. 754–774, 1989.

[64] A. Kumar, D. Manjunath, and J. Kuri, *Wireless Networking*. Burlington, MA, USA: Elsevier Inc., 2008.

[65] E. G. Larsson and E. A. Jorswieck, "Competition Versus Cooperation on the MISO Interference Channel," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 7, pp. 1059–1069, September 2008.

[66] B. O. Lee, H. W. Je, I. Sohn, O.-S. Shin, and K. B. Lee, "Interference-aware decentralized precoding for multicell MIMO TDD systems," in *Proceedings on IEEE Global Telecommunications Conference*, pp. 1–5, New Orleans, LA, USA, November 30–December 4 2008.

[67] G. Li and H. Liu, "On the Optimality of the OFDMA networks," *IEEE Communications Letters*, vol. 9, no. 5, pp. 438–440, May 2005.

[68] X. Lin and N. B. Shroff, "Joint rate control and scheduling in multihop wireless networks," Technical Reports, Purdue University, [Online]. Available: http://cobweb.ecn.purdue.edu/~linx/papers.html, 2004.

[69] X. Lin and N. B. Shroff, "Joint rate control and scheduling in multihop wireless networks," in *Proceedings on IEEE International Conference on Decision and Control*, vol. 5, pp. 1484–1489, Atlantis, Paradise Island, Bahamas, December 14–17 2004.

[70] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer rate control in wireless networks," in *Proceedings on IEEE INFOCOM*, vol. 3, pp. 1804–1814, Miami, USA, March 13–17 2005.

[71] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1452–1463, August 2006.

[72] Y.-F. Liu, Y.-H. Dai, and Z.-Q. Luo, "Coordinated beamforming for MISO interference channel: Complexity analysis and efficient algorithms," *IEEE Transactions on Signal Processing*, vol. 59, no. 3, pp. 1142–1157, March 2011.

[73] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra and Applications*, vol. 284, pp. 193–228, November 1998.

[74] R. Lui and W. Yu, "Low-complexity near-optimal spectrum balancing for digital subscriber lines," in *Proceedings on IEEE International Conference on Communications*, pp. 1947–1951, Seoul, Korea, May 16–20 2005.

[75] Z. Luo and S. Zhang, "Dynamic spectrum management: Complexity and duality," *IEEE Journal on Selected Areas in Communications*, vol. 2, no. 1, pp. 57–73, February 2008.

[76] Z. Q. Luo and W. Yu, "An introduction to convex optimization for communications and signal processing," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1426–1438, August 2006.

[77] B. R. Marks and G. P. Wright, "A general inner approximation algorithm for nonconvex mathematical programs," *Operational Research*, vol. 26, no. 4, pp. 681–683, July–August 1978.

[78] E. Matskani, N. D. Sidiropoulos, and L. Tassiulas, "Convex approximation algorithms for back-pressure power control of wireless multi-hop networks," in *Proceedings on IEEE International Conference on Acoustics, Speech, Signal Processing*, pp. 3032–3035, Prague, Czech Republic, May 22–27 2011.

[79] A. Nedić and A. Ozdaglar, "Cooperative distributed multi-agent optimization," in *Convex Optimization in Signal Processing and Communications*, (D. P. Palomar and Y. C. Eldar, eds.), pp. 340–386, Cambridge University Press, 2011.

[80] M. J. Neely, "Dynamic power allocation and routing for satellite and wireless networks with time varying channels," PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 2003.

[81] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*, vol. 7 of *Synthesis Lectures on Communication Networks*. San Rafael, CA: Morgan & Claypool, 2010.

[82] M. J. Neely, E. Modiano, and C. Li, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 396–409, April 2008.

[83] M. J. Neely, E. Modiano, and C. E. Rohrs, "Power allocation and routing in multibeam satellites with time-varying channels," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 138–152, February 2003.

[84] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time varying wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 89–103, January 2005.

[85] Y. Nesterov and A. Nemirovsky, *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, 1994.

[86] D. C. Oneill, D. Julian, and S. Boyd, "Optimal routes and flows in congestion constrained ad-hoc networks," in *Proceedings on IEEE Vehicle Technology Conference*, pp. 702–711, Los Angeles, CA, September 26–29 2004.

[87] D. P. Palomar, "A unified framework for communications through MIMO channels," PhD thesis, Department of Signal Theory and Communications, Technical University of Catalonia, Barcelona, Spain, May 2003.

[88] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1439–1451, August 2006.

[89] D. P. Palomar and M. Chiang, "Alternative distributed algorithms for network utility maximization: Framework and applications," *IEEE Transactions on Automatic Control*, vol. 52, no. 12, pp. 2254–2269, December 2007.

[90] A. Pantelidou and A. Ephremides, "A cross-layer view of wireless multicasting under uncertainty," in *Proceedings on IEEE Information Theory Workshop*, pp. 110–114, Volos, Greece, June 10–12 2009.

[91] A. Pantelidou and A. Ephremides, "A cross-layer view of optimal scheduling," *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5568–5580, November 2010.

[92] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs NJ: Prentice-Hall, 1982.

[93] J. Papandriopoulos and J. S. Evans, "SCALE: A low-complexity distributed protocol for spectrum balancing in multiuser DSL networks," *IEEE Transactions on Information Theory*, vol. 55, no. 8, pp. 3711–3724, August 2009.

[94] J. Papandriopoulos and S. Evans, "Low-complexity distributed algorithms for spectrum balancing in multi-user DSL networks," in *Proceedings on IEEE International Conference on Communications*, pp. 3270–3275, Istanbul, Turkey, June 11–15 2006.

[95] S.-H. Park, H. Park, and I. Lee, "Distributed beamforming techniques for weighted sum-rate maximization in MISO interference channels," *IEEE Communications Letters*, vol. 14, no. 12, pp. 1131–1133, December 2010.

[96] N. T. H. Phuong and H. Tuy, *A Unified Monotonic Approach to Generalized Linear Fractional Programming*. Kluwer Academic Publishers, 2003.

[97] L. Qian, Y. J. A. Zhang, and J. Huang, "MAPEL: Achieving global optimality for a non-convex wireless power control problem," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1553–1563, March 2009.

[98] B. Radunovic and J. L. Boudec, "Optimal power control, scheduling, and routing in UWB networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 7, pp. 1252–1270, September 2004.

[99] B. Radunovic and J. L. Boudec, "Power Control is not Required for Wireless Networks in the Linear Regime," in *Proceedings on IEEE International Symposium World of Wireless, Mobile and Multiple Networks*, vol. 5, pp. 417–427, Taormina, Giardini Naxos, Italy, June 13–16 2005.

[100] B. Radunović, D. Gunawardena, P. Key, A. Proutiere, N. Singh, V. Balan, and G. Dejean, "Rethinking Indoor Wireless: Low Power, Low Frequency, Full-duplex," Technical Reports MSR-TR-2009-148, Microsoft Research, July 2009, [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=104950.

[101] B. Radunović, D. Gunawardena, P. Key, A. Proutiere, N. Singh, V. Balan, and G. Dejean, "Rethinking indoor wireless mesh design: Low power, low frequency, full-duplex," in *Proceedings on IEEE Works on Wireless Mesh Networks*, pp. 1–6, Boston, USA, June 21 2010.

[102] B. Radunović, D. Gunawardena, A. Proutiere, N. Singh, V. Balan, and P. Key, "Efficiency and fairness in distributed wireless networks through self-interference cancellation and scheduling," Technical Reports MSR-TR-2009-27, Microsoft Research, [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=79933, March 2009.

[103] W. Rudin, *Principles of Mathematical Analysis*. USA: McGraw-Hill, Inc., 3 Edition, 1976.

[104] K. Seong, M. Mohseni, and M. Cioffi, "Optimal resource allocation for OFDMA downlink systems," in *Proceedings on IEEE International Symposium on Information Theory*, pp. 1394–1398, Seattle, USA, July 9–14 2006.

[105] G. Sharma, R. R. Mazumdar, and N. B. Shroff, "On the complexity of scheduling in wireless networks," in *Proceedings on ACM International Conference on Mobile Computation and Networking*, pp. 227–238, Los Angeles, CA, USA, September 24–29 2006.

[106] Z. Shen, J. G. Andrews, and B. L. Evans, "Adaptive resource allocation in multiuser OFDM systems with proportional rate constraints," *IEEE Transactions on Wireless Communications*, vol. 4, no. 6, pp. 2726–2737, November 2005.

[107] C. Shi, R. A. Berry, and M. L. Honig, "Distributed interference pricing with MISO channels," in *Proceedings on Annual Allerton Conference Communications, Cont, Computing*, pp. 539–546, Urbana-Champaign, USA, September 23–26 2008.

[108] C. Shi, R. A. Berry, and M. L. Honig, "Monotonic convergence of distributed interference pricing in wireless networks," in *Proceedings on IEEE International Symposium on Information Theory*, pp. 1619–1623, Seoul, Korea, June 28–July 3 2009.

[109] C. Shi, R. A. Berry, and M. L. Honig, "Local interference pricing for distributed beamforming in MIMO networks," in *Proceedings on IEEE Military Communications Conference*, pp. 1–6, Boston, MA, October 18–21 2009.

[110] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, September 2011.

[111] S. Shi, M. Schubert, and H. Boche, "Weighted sum-rate optimization for multiuser MIMO systems," in *Proceedings of the Conference on Information Sciences Systems (CISS)*, pp. 425–430, Baltimore, MD, USA, March 14–16 2007.

[112] Y. Shi and Y. T. Hou, "A distributed optimization algorithm for multi-hop cognitive radio networks," in *Proceedings on IEEE INFOCOM*, pp. 1292–1300, Phoenix, AZ, USA, April 13–18 2008.

[113] B. Song, R. L. Cruz, and B. D. Rao, "A simple joint beamforming and power control algorithm for multi-user MIMO wireless networks," in *Proceedings on IEEE Vehicles Technology Conference*, pp. 247–251, Los Angeles, CA, September 26–29 2004.

[114] B. Song, R. L. Cruz, and B. D. Rao, "Network duality and its application to multi-user MIMO wireless networks with SINR constraints," in *Proceedings on IEEE International Conference on Communications*, vol 4, pp. 2684–2689, Seoul, Korea, May 16–20 2005.

[115] T. Starr, M. Sorbara, J. Cioffi, and P. J. Silverman, *DSL Advances.* Upper Saddle River, NJ 07458: Prentice-Hall, 2003.

[116] M. Stojnic, H. Vikalo, and B. Hassibi, "Rate maximization in multi-antenna broadcast channels with linear preprocessing," *IEEE Transactions on Wireless Communications*, vol. 5, no. 9, pp. 2338–2342, September 2006.

[117] A. Stolyar and H. Viswanathan, "Self-organizing dynamic fractional frequency reuse for best-effort traffic through distributed inter-cell coordination," in *Proceedings on IEEE INFOCOM*, pp. 1287–1295, Rio De Janeiro, Brazil, April 19–25 2009.

[118] A. L. Stolyar, "Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm," *Queing Systems*, vol. 50, no. 4, pp. 401–457, August 2005.

[119] H. Suzuki, K. Itoh, Y. Ebin, and M. Sato, "A booster configuration with adaptive reduction of transmitter-receiver antenna coupling for pager systems," in *Proceedings on IEEE Vehicles Technology Conference* , vol. 3, pp. 1516–1520, Amsterdam, Netherlands, September 19–22 1999.

[120] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1949, December 1992.

[121] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Transactions on Information Theory*, vol. 39, no. 2, pp. 466–478, March 1993.

[122] A. Tölli, M. Codreanu, and M. Juntti, "Cooperative MIMO-OFDM cellular system with soft handover between distributed base station antennas," *IEEE Transactions on Wireless Communications*, vol. 7, no. 4, pp. 1428–1440, April 2008.

[123] A. Tölli, H. Pennanen, and P. Komulainen, "Decentralized minimum power multi-cell beamforming with limited backhaul signaling," *IEEE Transactions on Wireless Communications*, vol. 10, no. 2, pp. 570–580, February 2011.

[124] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, UK: Cambridge University Press, 2005.

[125] D. N. Tse, "Optimal power allocation over parallel gaussian broadcast channels," *Unpublished, a short summary published in Proceedings of International Symposium on Information Theory, Ulm, Germany*, 1997.

[126] D. N. C. Tse and S. V. Hanly, "Multiaccess fading channels. I. Polymatroid structure, optimal resource allocation and throughput capacities," *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 2796–2815, November 1998.

[127] P. Tsiaflakis, M. Diehl, and M. Moonen, "Distributed spectrum management algorithms for multiuser DSL networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4825–4843, October 2008.

[128] P. Tsiaflakis, W. Tan, Y. Yi, M. Chiang, and M. Moonen, "Optimality certificate of dynamic spectrum management in multi-carrier interference channels," in *Proceedings on IEEE International Symposium on Information Theory*, pp. 1298–1302, Toronto, Canada, July 6–11 2008.

[129] P. Tsiaflakis, J. Vangorp, M. Moonen, and J. Verlinden, "A low complexity optimal spectrum balancing algorithm for digital subscriber lines," *Elseviser Signal Processing*, vol. 87, no. 7, July 2007.

[130] L. Venturino, N. Prasad, and X. Wang, "Coordinated linear beamforming in downlink multi-cell wireless networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 4, pp. 1451–1461, April 2010.

[131] N. Vucic, S. Shi, and M. Schubert, "DC programming approach for resource allocation in wireless networks," in *Proceedings on Workshop on Resource Allocation in Wireless Networking*, pp. 360–366, Avignon, France, June 4 2010.

[132] P. C. Weeraddana, M. Codreanu, and M. Latva-aho, "Multicell downlink weighted sum-rate maximization: A distributed approach," in *Proceedings on Annual Asilomar Conference on Signals, Systems and Computers, to appear*, Pacific Grove, CA, USA, November 6–9 2011.

[133] P. C. Weeraddana, M. Codreanu, M. Latva-aho, and A. Ephremides, "On the effect of self-interference cancelation in multihop wireless networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, Article ID 513952, 10 pages, 2010, doi:10.1155/2010/513952.

[134] P. C. Weeraddana, M. Codreanu, M. Latva-aho, and A. Ephremides, "Resource allocation for cross-layer utility maximization in wireless networks," *IEEE Transactions on Vehicles Technology*, vol. 60, no. 6, pp. 2790–2809, July 2011.

[135] P. C. Weeraddana, M. Codreanu, M. Latva-aho, and A. Ephremides, "Weighted sum-rate maximization for a set of interfering links via branch and bound," *IEEE Transactions on Signal Processing*, vol. 59, no. 8, pp. 3977–3996, August 2011.

[136] P. C. Weeraddana, M. Codreanu, M. Latva-aho, and A. Ephremides, "Multicell downlink weighted sum-rate maximization: A distributed approach," *IEEE Transactions on Signal Processing, submitted*, 2012.

[137] H. Weingarten, Y. Steinberg, and S. Shamai, "The capacity region of the Gaussian multiple-input multiple-output broadcast channel," *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 3936–3964, September 2006.

[138] X. Wu and R. Srikant, "Regulated maximal matching: A distributed scheduling algorithm for multi-hop wireless networks with node-exclusive spectrum sharing," in *Proceedings on IEEE Conference on Decision and Control, and the European Control Conference*, pp. 5342–5347, Seville, Spain, December 12–15 2005.

[139] Y. Xu, T. Le-Ngoc, and S. Panigrahi, "Global concave minimization for optimal spectrum balancing in multi-user DSL networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 2875–2885, July 2008.

[140] Y. Xu, S. Panigrahi, and T. Le-Ngoc, "A concave minimization approach to dynamic spectrum management for digital subscriber lines," in *Proceedings on IEEE International Conference on Communications*, pp. 84–89, Istanbul, Turkey, June 11–15 2006.

[141] B. Yang and M. Johansson, "Distributed optimization and games: A tutorial overview," *Networked Control Systems, Lecture Notes in Control and Information Sciences, London: Springer-Verlag*, vol. 406, pp. 109–148, 2011.

[142] Y. Yi and M. Chiang, "Stochastic network utility maximization: A tribute to Kelly's paper published in this journal a decade ago," *European Transactions on Telecommunications*, vol. 19, no. 4, pp. 421–442, June 2008.

[143] H. D. W. Yu, "Coordinated beamforming for the multicell multi-antenna wireless system," *IEEE Transactions on Wireless Communications*, vol. 9, no. 5, pp. 1748–1759, May 2010.

[144] W. Yu, "Multiuser water-filling in the presence of crosstalk," San Diego, USA, pp. 414–420, January 29–February 2, 2007.

[145] W. Yu and M. Cioffi, "FDMA capacity of Gaussian multiple-access channels with ISI," *IEEE Transactions on Communications*, vol. 50, no. 1, pp. 102–111, January 2002.

[146] W. Yu, T. Kwon, and C. Shin, "Joint scheduling and dynamic power spectrum optimization for wireless multicell networks," in *Proceedings on Conference on Information Sciences Systems (CISS)*, pp. 1–6, Princeton, NJ, USA, March 21–23 2010.

[147] J. Yuan and W. Yu, "Distributed cross-layer optimization of wireless sensor networks: A game theoretic approach," in *Proceedings on IEEE Global Telecommunications Conference*, San Francisco, California, November 27–December 1 2006.

[148] R. Zakhour and D. Gesbert, "Coordination on the MISO interference channel using the virtual SINR framework," in *Proceedings on ITG Workshop Smart Antennas*, Berlin, Germany, February 16–18 2009.

[149] R. Zakhour and D. Gesbert, "Distributed multicell-MISO precoding using the layered virtual SINR framework," *IEEE Transactions on Wireless Communications*, vol. 9, no. 8, pp. 2444–2448, August 2010.

[150] R. Zakhour, Z. Ho, and D. Gesbert, "Distributed beamforming coordination in multicell MIMO channels," in *Proceedings on IEEE Vehicles Technology Conference*, pp. 1–5, Barcelona, Spain, April 26–29 2009.

[151] L. Zhang, Y. Xin, Y. C. Liang, and H. V. Poor, "Cognitive multiple access channels: Optimal power allocation for weighted sum rate maximization," *IEEE Transactions on Communications*, vol. 57, no. 9, pp. 2754–2762, September 2009.

[152] R. Zhang and S. Cui, "Cooperative interference management with MISO beamforming," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5450–5458, October 2010.