Optimal Resource Allocation in Coordinated Multi-Cell Systems: Matlab Code

Supplement to a Research Book in Foundations and Trends in Communications and Information Theory

Vol. 9, no. 2-3, pp. 113-381, 2013, DOI: http://dx.doi.org/10.1561/0100000069

Version 1.2, July 2014

by Emil Björnson[⋆†] and Eduard Jorswieck[‡]

[⋆]ACCESS, Dept. of Signal Processing, KTH Royal Institute of Technology, Osquldas väg 10, SE-100 44 Stockholm, Sweden

[†]Alcatel-Lucent Chair on Flexible Radio, Supélec, Plateau du Moulon, 3 rue Joliot-Curie, 91192, Gif-sur-Yvette cedex, France

[‡]Communications Theory, Communications Laboratory, Dresden University of Technology, Dresden 01062, Germany

Contents

1 Introduction

This is the documentation of the Matlab code supplement to the book "Optimal Resource Allocation in Coordinated Multi-Cell Systems" by Emil Björnson and Eduard Jorswieck; see [1] for the full publication details. The abstract of this monograph reads:

Abstract of the book: The use of multiple antennas at base stations is a key component in the design of cellular communication systems that can meet high-capacity demands in the downlink. Under ideal conditions, the gain of employing multiple antennas is well-recognized: the data throughput increases linearly with the number of transmit antennas if the spatial dimension is utilized to serve many users in parallel. The practical performance of multi-cell systems is, however, limited by a variety of nonidealities, such as insufficient channel knowledge, high computational complexity, heterogeneous user conditions, limited backhaul capacity, transceiver impairments, and the constrained level of coordination between base stations.

This tutorial presents a general framework for modeling different multi-cell scenarios, including clustered joint transmission, coordinated beamforming, interference channels, cognitive radio, and spectrum sharing between operators. The framework enables joint analysis and insights that are both scenario independent and dependent.

The performance of multi-cell systems depends on the resource allocation; that is, how the time, power, frequency, and spatial resources are divided among users. A comprehensive characterization of resource allocation problem categories is provided, along with the signal processing algorithms that solve them. The inherent difficulties are revealed: (a) the overwhelming spatial degrees-of-freedom created by the multitude of transmit antennas; and (b) the fundamental tradeoff between maximizing aggregate system throughput and maintaining user fairness. The tutorial provides a pragmatic foundation for resource allocation where the system utility metric can be selected to achieve practical feasibility. The structure of optimal resource allocation is also derived, in terms of beamforming parameterizations and optimal operating points.

This tutorial provides a solid ground and understanding for optimization of practical multi-cell systems, including the impact of the nonidealities mentioned above. The Matlab code is available online for some of the examples and algorithms in this tutorial.

This documentation is distributed along with the code package mentioned above. The package contains Matlab implementations of many of the algorithms described in [1]. The use of these algorithms is exemplified by Matlab scripts (m-files) that generate some of the figures shown in the book. The algorithms are briefly described in Section 7 and the selected example figures are described and shown in Section 8. Please note that the all channel vectors are generated randomly as Rayleigh fading in these examples, thus this code package is not able to reproduce exactly the same curves as was shown in the book.

1.1 Problem Formulation: Optimal Resource Allocation

Resource allocation in multi-cell multi-antenna systems can be described as the maximization of a system utility function by allocating transmit power among users and spatial directions. The allocation should satisfy a set of power constraints that have physical, regulatory, and economic implications. The book provides an optimization framework for downlink resource allocation in different multi-cell scenarios with K_t multi-antenna transmitters, which have a total of N transmit antennas. There are K_r single-antenna receiving users. The framework can describe different transmitter cooperation scenarios, including global joint transmission, dynamic cooperation clusters, interference channels, coordinated beamforming, and cognitive radio. These scenarios are characterized by the design parameters $\mathbf{D}_k \in \mathbb{C}^{N \times N}$, which are diagonal matrices with zeros and ones on the main diagonal. One means that the corresponding antenna might send data to the user k; see [1] for details and examples.

Multi-cell resource allocation can be formulated as the single-objective optimization problem

$$\begin{aligned} \underset{\mathbf{v}_1,\ldots,\mathbf{v}_{K_r}}{\text{maximize}} \;\; & f\left(g_1(\mathrm{SINR}_1),\ldots,g_{K_r}(\mathrm{SINR}_{K_r})\right) \\ \text{subject to } \; & \mathrm{SINR}_k = \frac{|\mathbf{h}_k^H \mathbf{D}_k \mathbf{v}_k|^2}{1+\sum\limits_{i\neq k}|\mathbf{h}_k^H \mathbf{D}_i \mathbf{v}_i|^2} \quad \forall k, \\ & \sum_{k=1}^{K_r} \mathbf{v}_k^H \mathbf{Q}_l \mathbf{v}_k \leq q_l \quad \forall l. \end{aligned} \tag{1}$$

The optimization variables are the beamforming vectors $\mathbf{v}_k \in \mathbb{C}^{N\times 1}$ for $k = 1,\ldots,K_r$. The channel from transmitter j to user k is described by a channel vector $\mathbf{h}_{jk}$ and the collective channel vector from all transmitters is denoted $\mathbf{h}_k = [\mathbf{h}_{1k}\ \ldots\ \mathbf{h}_{K_t k}]^T \in \mathbb{C}^{N\times 1}$. The channels are fixed during the transmission. The performance of user k is measured by a continuous and strictly monotonically increasing function $g_k(\cdot)$ of the signal-to-interference-and-noise ratio (SINR).

The beamforming vectors satisfy a set of L power constraints: $\sum_{k=1}^{K_r} \mathbf{v}_k^H \mathbf{Q}_l \mathbf{v}_k \leq q_l$ for $l = 1,\ldots,L$. The weighting matrices $\mathbf{Q}_l \in \mathbb{C}^{N\times N}$ are Hermitian positive semi-definite and the limits q_l are positive for all l. These design parameters are given in advance by the multi-cell scenario under study. The power constraint can model any combination of per-antenna limitations, per-array limitations, soft-shaping constraints, regulatory constraints, etc.

The system utility is a weighted combination of the performance of each user, described by the system utility function $f(\cdot)$. This function is assumed to be Lipschitz continuous and monotonically increasing in the performance $g_k(\mathrm{SINR}_k)$ of each user. The following are common examples.

- Weighted arithmetic mean: $f(g_1,\ldots,g_{K_r}) = \sum_k w_k g_k$ (also known as weighted sum rate);
- Weighted geometric mean: $f(g_1,\ldots,g_{K_r}) = \prod_k g_k^{w_k}$ (also known as weighted proportional fairness);
- Weighted max-min fairness: $f(g_1,\ldots,g_{K_r}) = \min_k \frac{g_k}{w_k}$ (also known as weighted worst-user rate).

The weights $w_1,\ldots,w_{K_r}$ are positive parameters in these examples. The system utility functions are defined such that it is favorable for the system to allocate many resources to users with large weights.

The choice of system utility has two main consequences: (a) it defines the balance between aggregate data throughput and user fairness in the optimal resource allocation; and (b) it determines the computational complexity of solving (1). System utilities based on arithmetic/geometric means generally give non-convex monotonic optimization problems and the computational complexity scales exponentially with the number of users K_r.[1] On the other hand, system utilities based on max-min fairness give quasi-convex problems, thus the optimal resource allocation can be computed with polynomial computational complexity.

This package contains code for solving (1) for any of the aforementioned system utilities. The Polyblock Outer Approximation (PA) algorithm and Branch-Reduce-and-Bound (BRB) algorithm maximize the weighted arithmetic and geometric means, while the fairness-profile optimization (FPO) algorithm maximizes the weighted max-min fairness. All three algorithms solve a sequence of convex feasibility problems, which check if a certain combination of $(\mathrm{SINR}_1,\ldots,\mathrm{SINR}_{K_r})$ is achievable under the power constraints—this is equivalent to checking if $(g_1(\mathrm{SINR}_1),\ldots,g_{K_r}(\mathrm{SINR}_{K_r}))$ is inside of the so-called achievable rate region. Implementations of this feasibility problem under ideal and non-ideal conditions are provided as functions.

[1] There are two important special cases: single-antenna transmitters and zero-forcing transmission. In these special cases, most system utilities give convex resource allocation problems and the computational complexity can therefore be greatly reduced. See Table 2.1 in [1] for further details on this. Please note that the code in this package does not identify and exploit the structure of these special cases, thus we recommend alternative implementations for these cases.

2 Differences From the Book

The implementations in this code package are based on some simplifying assumptions as compared to the general problem formulations described in the book. Firstly, we only consider the information rate $g_k(\mathrm{SINR}_k) = \log_2(1+\mathrm{SINR}_k)$ as user performance function, but it is relatively easy to consider other user performance functions as well. Moreover, all the matrices $\mathbf{C}_k$ in the book are set to identity matrices (i.e., $\mathbf{C}_k = \mathbf{I}_N$) and are thus not found in the code or in this documentation. This means that the signals from all transmitters will reach all receivers. This is not really a limitation, because one can model a silent communication link between a transmit antenna and a receiver by setting the corresponding element in $\mathbf{h}_k$ to zero. In addition, the noise powers have been normalized as $\sigma_k^2 = 1$ for each user k. Hence, one can think of $\mathbf{h}_k$ in the implementations as actually being $\frac{1}{\sigma_k}\mathbf{h}_k$. Finally, the weighting matrices $\mathbf{Q}_l \in \mathbb{C}^{N\times N}$ also had a user index in the book, while this code assumes that it is the same for all users.

3 Software Requirements

The code requires Matlab—a high-level language and interactive environment for numerical computation, visualization, and programming from Mathworks Inc. We have tried the code with many different versions of Matlab, but the final testing was performed on Matlab R2012b. The command `rng('shuffle')`, which assigns a stochastic seed to the random number generators, is not be supported by older Matlab versions. The commands `randn('state',sum(100*clock)); rand('state',sum(100*clock));` can be used instead.

The code package solves a variety of convex, quasi-convex, and non-convex optimization problems in the area of multi-cell multi-antenna wireless communications. An important part of the algorithms is the solution to a certain optimization problem: *minimize the transmit power while satisfying quality-of-service (QoS) constraints and power constraints*. This convex optimization problem is solved by `CVX`—a Matlab-based modeling system for convex optimization from [3]. `CVX` and the bundled free solvers, `SeDuMi` [4] and `SDPT3` [5], are distributed as open source and are free to download. There are also commercial solvers that might be faster. This code package has been used together with several version of `CVX`, but the final testing of the original code release was performed on the `CVX` 2.0 beta. Version 1.2 of the code package was tested on `CVX` 2.1 (June 2014, Build 1085, BYOS).

4 License

The *m*-files in this package are entirely free for use in both academic and commercial settings. We hope that the code will be useful to (a) understand the algorithms described in the book [1]; (b) simplify implementation and evaluation of new algorithms for resource allocation in multi-cell systems; and (c) encourage reproducibility and comparability of future research results.

If our code is used in research work that is published, please include an explicit mention of it in the publication. This serves both as an acknowledgment of the effort behind making this code package available and to promote reproducible research by spreading the word about the existence of this code package. The book and this document can be cited as follows:

- E. Björnson and E. Jorswieck, "Optimal resource allocation in coordinated multi-cell systems," *Foundations and Trends in Communications and Information Theory*, vol. 9, no. 2-3, pp. 113–381, 2013
- E. Björnson and E. Jorswieck, "Optimal resource allocation in coordinated multi-cell systems: Matlab code," Tech. Rep., July 2014. doi: 10.1561/0100000069_supp, version 1.2

Please note that `CVX` [3] and the solvers that it uses (e.g., `SeDuMi` [4] and `SDPT3` [5]) also deserve recognition. We refer to their respective web pages for suitable formulations of such citations.

5 Comments and Bug Reports

Comments, questions, and bug reports can be submitted by e-mail to `emil.bjornson@liu.se`. We will try keep the code package up-to-date, in case any bugs are found. We might also add additional examples or algorithms from the book if we receive such requests.

Please note that the code package is *not* exactly the same as was originally used to generate the figures in the book—some features have been improved and new bugs might have appeared.

6 Acknowledgments

The authors would like to thank Mats Bengtsson and Jinghong Yang for giving feedback on the implementations in this code package.

7 Algorithms

This section describes the different algorithms in the code package.

7.1 Solve Feasibility Problem with QoS Constraints

The function `functionFeasibilityProblem_cvx.m` solves the feasibility problem with quality-of-service (QoS) constraints in (2.29) of [1]. The implementation is based on the alternative formulation in (2.30): power minimization under QoS requirements.

The power minimization problem under QoS requirements of $\text{SINR}_k \geq \gamma_k$, for $k = 1, \ldots, K_r$, is

$$\begin{aligned} \underset{\mathbf{v}_k \, \forall k, \, \beta}{\text{minimize}} \;\; & \beta \\ \text{subject to} \;\; & |\mathbf{h}_k^H \mathbf{D}_k \mathbf{v}_k|^2 \geq \gamma_k \Big(1 + \sum_{i \neq k} |\mathbf{h}_k^H \mathbf{D}_i \mathbf{v}_i|^2 \Big) \quad \forall k, \\ & \sum_{k=1}^{K_r} \mathbf{v}_k^H \mathbf{Q}_l \mathbf{v}_k \leq \beta q_l \quad \forall l. \end{aligned} \tag{2}$$

If this optimization problem is feasible and $\beta^* \leq 1$ for the optimal $\beta = \beta^*$, then the feasibility problem with QoS constraints is also feasible. This optimization problem in (2) is convex.[2] The computational complexity is therefore polynomial in the number of users, antennas, and power constraints. The implementation uses `CVX` and can, at least, handle $K_r = 30$ users, $N = 50$ transmit antennas, and $L = 50$ power constraints.

7.2 Solve Feasibility Problem with QoS Constraints under Impairments

The function `functionFeasibilityProblem_Impairments_cvx.m` solves a generalized version of the feasibility problem with quality-of-service (QoS) constraints described in the previous section. The generalization consists of having physical transceiver impairments that distort the transmitted and received signals at each antenna. A theoretical impairment model is described in Corollary 4.3 in [1]. This model supports general distortion functions, but this implementation only considers linear distortion functions.

The optimization problem in (2) is generalized as

$$\begin{aligned} \underset{\mathbf{v}_k \, \forall k, \, \beta}{\text{minimize}} \;\; & \beta \\ \text{subject to} \;\; & \text{EVM}_{(t)} \| \mathbf{T}_n [\mathbf{D}_1 \mathbf{v}_1 \; \ldots \; \mathbf{D}_{K_r} \mathbf{v}_{K_r}] \|_F \leq t_n \quad \forall n, \\ & \frac{|\mathbf{h}_k^H \mathbf{D}_k \mathbf{v}_k|^2}{\Big(1 + \text{EVM}_{(r)}^2 |\mathbf{h}_k^H \mathbf{D}_k \mathbf{v}_k|^2 + (1 + \text{EVM}_{(r)}^2) \sum_{i \neq k} |\mathbf{h}_k^H \mathbf{D}_i \mathbf{v}_i|^2 + \sum_{n=1}^{N} t_n^2 \mathbf{h}_k^H \mathbf{T}_n \mathbf{h}_k \Big)} \geq \gamma_k \quad \forall k, \\ & \sum_{k=1}^{K_r} \mathbf{v}_k^H \mathbf{Q}_l \mathbf{v}_k \leq \beta q_l \quad \forall l, \end{aligned} \tag{3}$$

where $\text{EVM}_{(t)}, \text{EVM}_{(r)}$ are the error vector magnitudes (EVMs) that describe the level of impairments at the transmitter and receiver, respectively. Note that $\text{EVM}_{(t)}^2, \text{EVM}_{(r)}^2$ act as proportionality constants between the signal powers and distortion powers; see Chapter 4.3 in [1] for details. The diagonal matrix $\mathbf{T}_n$ has one at the nth diagonal element and zero elsewhere.

If this optimization problem is feasible and $\beta \leq 1$ for the optimal $\beta = \beta^*$, then the corresponding feasibility problem with QoS constraints is also feasible. This optimization problem is convex. Note that this function can be used instead of `functionFeasibilityProblem_cvx.m` in any of the algorithms in this code package, although we have only added explicit support in the FPO algorithm.

[2] The SINR constraints in (2) are not convex, but can be rewritten as convex second-order cone constraints without loss of generality. See the book for details.

7.3 Fairness-Profile Optimization (FPO) Algorithm

The function `functionFairnessProfile_cvx.m` solves the FPO problem in Example 2.8 in [1]. This is a generalization of max-min fairness optimization. The FPO problem searches on a line between $\mathbf{a} = [a_1, \ldots, a_{K_r}]^T$ and $\mathbf{b} = [b_1, \ldots, b_{K_r}]^T$ and finds the intersection between the line and the Pareto boundary of the rate region. The line search is based on bisection and solving optimization problems of the type in (2) to decide if a point on the line is inside of the rate region or outside.

The FPO problem is equivalent to solving

$$\begin{aligned} \underset{\mathbf{v}_k \, \forall k}{\text{maximize}} \quad & \min_k \frac{g_k(\text{SINR}_k) - a_k}{b_k - a_k} \\ \text{subject to} \quad & |\mathbf{h}_k^H \mathbf{D}_k \mathbf{v}_k|^2 \geq g_k^{-1}(a_k) \Big(1 + \sum_{i \neq k} |\mathbf{h}_k^H \mathbf{D}_i \mathbf{v}_i|^2 \Big) \quad \forall k, \\ & \sum_{k=1}^{K_r} \mathbf{v}_k^H \mathbf{Q}_l \mathbf{v}_k \leq q_l \quad \forall l. \end{aligned} \tag{4}$$

This problem is quasi-convex, meaning that it can be solved as a short sequence of convex optimization problems. These subproblems are solved by `functionFeasibilityProblem_cvx.m` for the ideal case or `functionFeasibilityProblem_Impairments_cvx.m` for the case with transceiver impairments. The computational complexity is therefore polynomial in the number of users, transmit antennas, and power constraints. The implementation can, at least, handle $K_r = 30$ users, $N = 50$ transmit antennas, and $L = 50$ power constraints.

7.4 Distributed Algorithm for the FPO problem

The function `functionFairnessProfile_Distributed_cvx.m` solves the FPO problem in Example 2.8 in [1] using the distributed Algorithm 5. This is the same optimization problem as in the previous section, but formulated slightly differently: search on a line from $\mathbf{a} = [a_1, \ldots, a_{K_r}]^T$ in the direction of $\mathbf{w} = [w_1, \ldots, w_{K_r}]^T$ and find the intersection between the line and the Pareto boundary of the rate region.

The FPO problem is equivalent to solving

$$\begin{aligned} \underset{\mathbf{v}_k \, \forall k}{\text{maximize}} \quad & \min_k \frac{g_k(\text{SINR}_k) - a_k}{w_k} \\ \text{subject to} \quad & |\mathbf{h}_k^H \mathbf{D}_k \mathbf{v}_k|^2 \geq g_k^{-1}(a_k) \Big(1 + \sum_{i \neq k} |\mathbf{h}_k^H \mathbf{D}_i \mathbf{v}_i|^2 \Big) \quad \forall k, \\ & \sum_{k=1}^{K_r} \mathbf{v}_k^H \mathbf{Q}_l \mathbf{v}_k \leq q_l \quad \forall l. \end{aligned} \tag{5}$$

The distributed algorithm moves step-by-step along the line using the fixed step size $\tau > 0$. The feasibility of a point on the line is evaluated by solving the feasibility problem (2.29) in [1] in a distributed manner. Dual decomposition is applied to divide the original problem (2.29) into the K_r distributed subproblems (one per user)

$$\begin{aligned} \underset{\mathbf{v}_k, \{\Theta_{ki}\}_{\forall i}, \{\widetilde{\Theta}_{ik}\}_{\forall i}, \{q_{lk}\}_{\forall l}}{\text{minimize}} \quad & \sum_{i \neq k} \Big(y_{ki} \Theta_{ki} - y_{ik} \widetilde{\Theta}_{ik} \Big) + \sum_{l=1}^{L} z_l q_{lk} \\ \text{subject to} \quad & \frac{|\mathbf{h}_k^H \mathbf{D}_k \mathbf{v}_k|^2}{1 + \sum_{i \neq k} \widetilde{\Theta}_{ik}^2} \geq \gamma_k \quad \forall k, \\ & \mathbf{v}_k^H \mathbf{Q}_l \mathbf{v}_k \leq q_{lk}, \quad q_{lk} \leq q_l \quad \forall l, \\ & |\mathbf{h}_i^H \mathbf{D}_k \mathbf{v}_k|^2 \leq \Theta_{ki}^2 \quad i \neq k \end{aligned} \tag{6}$$

and a master dual problem

$$\underset{\{y_{ik}\}_{\forall k,i},\{z_l\}_{\forall l}}{\text{maximize}} \sum_{k=1}^{K_r} \sum_{i \neq k} y_{ik} \left(\Theta_{ik}^* - \widetilde{\Theta}_{ik}^* \right) + \sum_{l=1}^{L} z_l \left(\sum_{k=1}^{K_r} q_{lk}^* - q_l \right) \tag{7}$$

where $\Theta_{ik}^*, \widetilde{\Theta}_{ik}^*, q_{lk}^*$ are the current solutions of the subproblems.

The subproblems are convex optimization problems while the master problem is solved by moving along the subgradient with a step size of $\xi > 0$. The parameters y_{ik} and z_l are Lagrange multipliers of the interference consistency constraints $\Theta_{ik} \leq \widetilde{\Theta}_{ik}$ and power consistency constraints $\sum_{i=1}^{K_r} q_{li} \leq q_l$, respectively.

7.5 Branch-Reduce-and-Bound (BRB) Algorithm

The BRB algorithm was developed in [7] for solving any monotonic optimization problem. The BRB algorithm in Algorithm 3 in [1] is adapted for solving any optimization problem of the form in (1). The implementation in `functionBRBalgorithm_cvx.m` is written to maximize the weighted sum rate or weighted proportional fairness, but can easily be generalized for other system utility functions.

Observe that the maximization of the weighted sum rate or the weighted proportional fairness are non-convex problems and NP-hard in general, thus the computational complexity scales exponentially with the number of user K_r. This implementation is not recommend for more than $K_r = 6$ users. The numerical evaluation in Section 2.4 of [1] shows that the BRB algorithm is relatively good at maximizing the weighted sum rate, while the PA algorithm is better at maximizing the weighted proportional fairness.

The algorithm terminates when an ϵ-optimal solution is found, where $\epsilon > 0$ is an input parameter. It also terminates when (a) the number of iterations in the algorithm meets a predefined input value or (b) the number of convex feasibility problems of the type solved by `functionFeasibilityProblem_cvx.m` meets a certain predefined input number. Since the run time can be several hours, it is recommended to try the algorithm with a small number of iterations before solving the actual problem. It is also important to note that the algorithm typically finds an ϵ-optimal solution long before it is able to validate that the solution actually satisfies the ϵ-optimality condition.

7.6 Polyblock Outer Approximation (PA) Algorithm

The PA algorithm was developed in [6] for solving any monotonic optimization problem. The PA algorithm in Algorithm 2 in [1] is adapted for solving any optimization problem of the form in (1). The implementation in `functionPAalgorithm_cvx.m` is written to maximize the weighted sum rate or weighted proportional fairness, but can easily be generalized for other system utility functions.

Observe that the maximization of the weighted sum rate or the weighted proportional fairness are non-convex problems and NP-hard in general, thus the computational complexity scales exponentially with the number of user K_r. This implementation is not recommend for more than $K_r = 6$ users. The numerical evaluation in Section 2.4 of [1] shows that the PA algorithm is relatively good at maximizing the weighted proportional fairness, while the BRB algorithm is better at maximizing the weighted sum rate.

The algorithm terminates when an ϵ-optimal solution is found, where $\epsilon > 0$ is an input parameter. It also terminates when (a) the number of iterations in the algorithm meets a predefined input value or (b) the number of convex feasibility problems of the type solved by `functionFeasibilityProblem_cvx.m` meets a certain predefined input number. Since the run time can be several hours, it is recommended to try the algorithm with a small number of iterations before solving the actual problem. It is also important to note that the algorithm typically finds an ϵ-optimal solution long before it is able to validate that the solution actually satisfies the ϵ-optimality condition.

7.7 Heuristic Beamforming Algorithms

Three heuristic beamforming strategies are also provided in the code package:

- Maximum ratio transmission (MRT) in `functionMRT.m`. This beamforming strategy tries to maximize the received signal power.
- Zero-forcing beamforming (ZFBF) in `functionZFBF.m`. This beamforming strategy tries to minimize the received interference power.
- Signal-to-leakage-and-noise ratio maximizing (SLNR-MAX) beamforming in `functionSLNRMAX.m`. This beamforming strategy tries to balance between maximizing the received signal power and minimizing the interference power. This strategy is also known as transmit Wiener filtering, virtual SINR beamforming, and regularized zero-forcing; see Remark 3.2 in [1] for a historical background to this popular heuristic beamforming strategy.

These functions compute (normalized) beamforming directions. These directions are generally suboptimal, but MRT can be asymptotically optimal at low SNR and ZFBF can be asymptotically optimal at high SNR. SLNR-MAX beamforming combines the asymptotic properties of MRT/ZFBF in the low/high SNR regimes with relatively good performance at intermediate SNRs. See Section 3.4 in [1] for definitions and further details on these beamforming strategies.

The code package also includes a heuristic power allocation that can be used along with MRT, ZFBF, and SLNR-MAX: `functionHeuristicPowerAllocation.m`. This function calculates the power allocation in Theorem 3.5 in [1] for any fixed beamforming directions, weighted sum rate maximization, and a total power constraint per base station. The power allocation is optimal for coordinated beamforming with ZFBF. Otherwise, the power allocation is suboptimal.

8 Examples & Figures

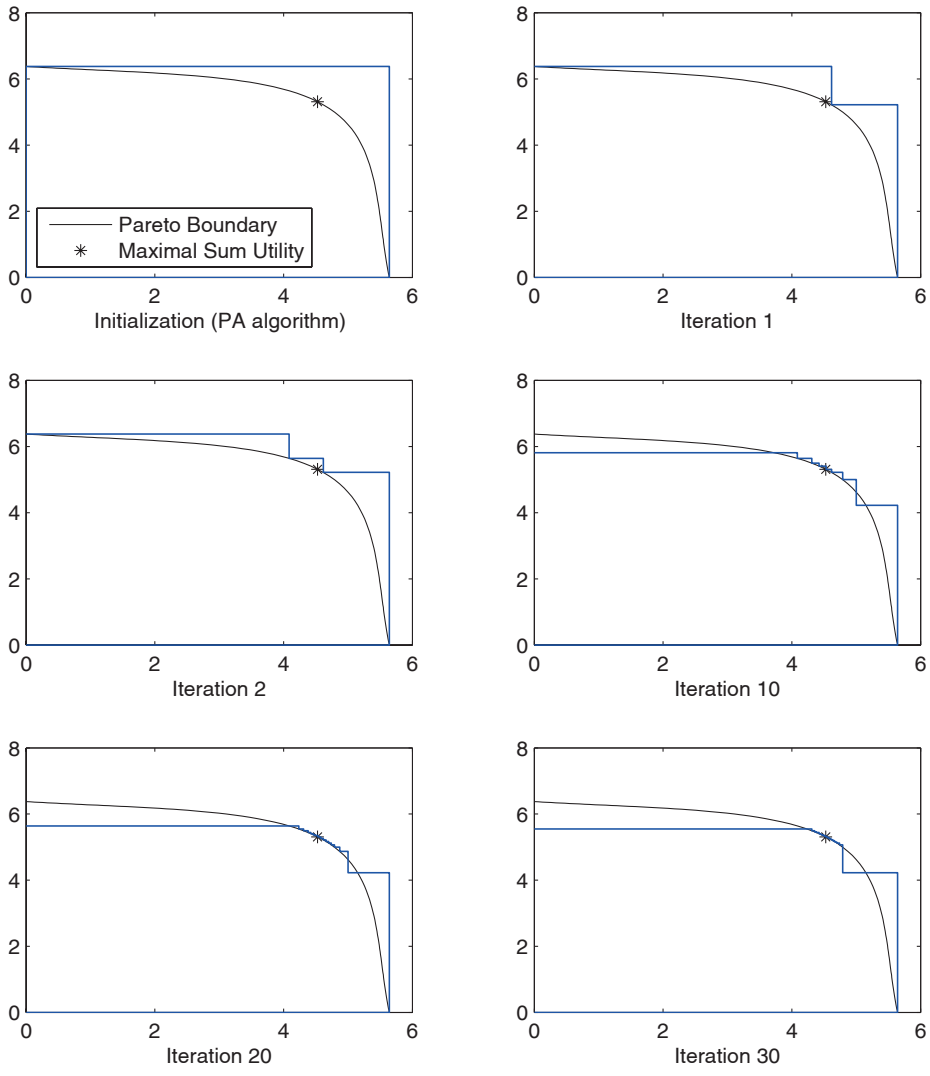
8.1 Figure 2.8

This figure illustrates the Polyblock Outer Approximation (PA) algorithm. The sum information rate is maximized by approximating the rate region (around the optimal point) from above using a polyblock. The approximation is iteratively refined and parts that cannot contain the optimal point are removed.

We consider a downlink scenario with $K_t = 2$ transmitters that serves $K_r = 2$ users by global joint transmission. Each transmitter has 2 antennas and per-array power constraints of $q_l = 10$ dB. Each user is "close" to one of the transmitters. The figure considers a single random channel realization where the channel $\mathbf{h}_{jk}$ between transmitter j and user k is generated as uncorrelated Rayleigh fading. The average single-user SNR $\mathbb{E}\{q_l \|\mathbf{h}_{jk}\|_2^2\}$ is $2q_l$ for the user close to transmitter j and $\frac{2}{3}q_l$ for the other user.

The implementation runs a few iterations of the PA algorithm using `functionPAalgorithm_cvx.m`. The line-search accuracy is $\delta = 0.001$. For illustrative purposes, the Pareto boundary of the rate region is also sampled and showed. The samples are achieved by applying Theorem 3.4 in [1] to search for the Pareto boundary in 101 equally spaced directions from the origin. This search uses the function `functionFairnessProfile_cvx.m`. The sample point that maximizes the sum rate is also indicated.

The progress of the PA algorithm after different number of iterations is shown at bottom of the page. Note that Figure 2.10 on the next page shares code with this example and shows the same channel realization.



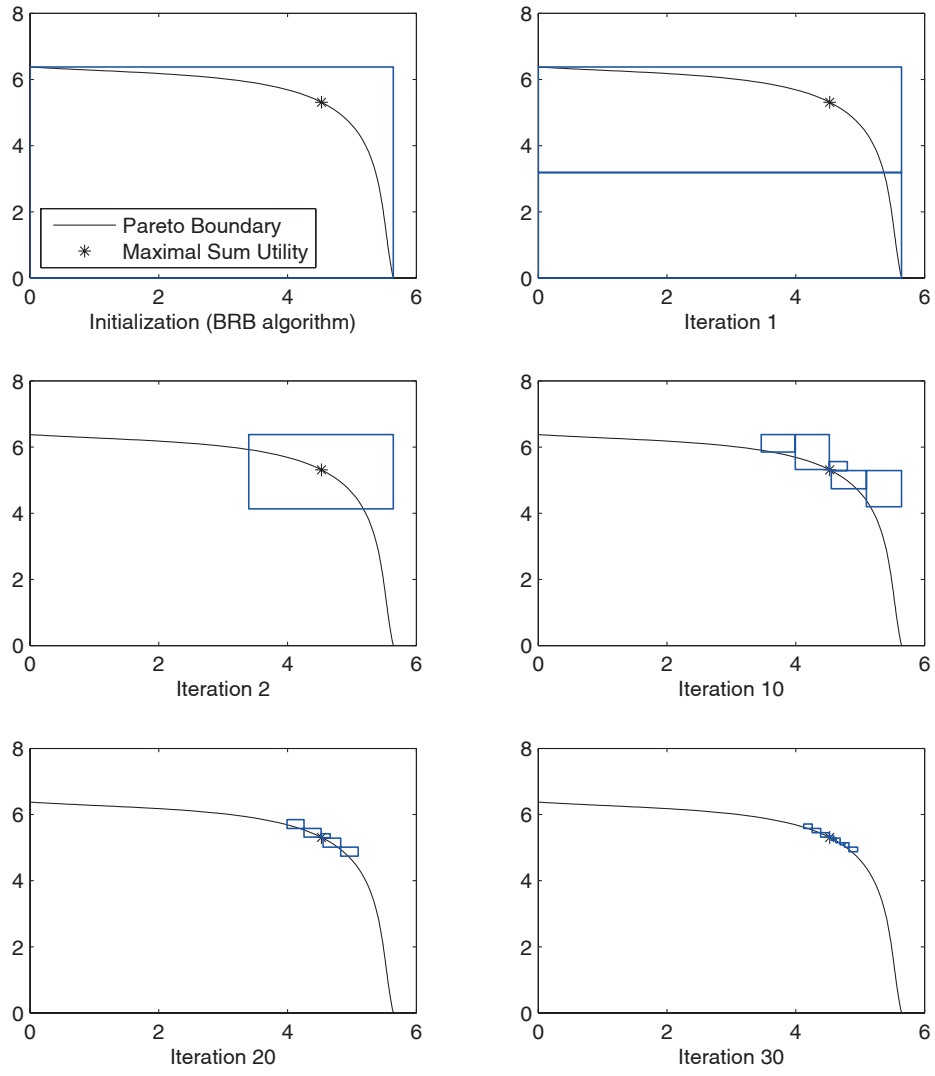
8.2 Figure 2.10

This figure illustrates the Branch-Reduce-and-Bound (BRB) algorithm. The sum information rate is maximized by approximating the Pareto boundary of the rate region (around the optimal point) using a set of disjoint boxes. The approximation is iteratively refined and parts that cannot contain the optimal point are removed.

We consider a scenario with $K_t = 2$ transmitters that serves $K_r = 2$ users by global joint transmission. Each transmitter has 2 antennas and per-array power constraints of $q_l = 10$ dB are used. Each user is "close" to one of the transmitters. The figure considers a single random channel realization where the channel $\mathbf{h}_{jk}$ between transmitter j and user k is generated as uncorrelated Rayleigh fading. The average single-user SNR $\mathbb{E}\{q_l \|\mathbf{h}_{jk}\|_2^2\}$ is $2q_l$ for the user close to transmitter j and $\frac{2}{3}q_l$ for the other user.

The implementation runs a few iterations of the BRB algorithm using `functionBRBalgorithm_cvx.m`. The line-search accuracy is $\delta = 0.001$. For illustrative purposes, the Pareto boundary of the rate region is also sampled and showed. The samples are achieved by applying Theorem 3.4 in [1] to search for the Pareto boundary in 101 equally spaced directions from the origin. This search uses the function `functionFairnessProfile_cvx.m`. The sample point that maximizes the sum rate is also indicated.

The progress of the BRB algorithm after different number of iterations is shown at bottom of the page. Note that Figure 2.8 on the previous page shares code with this example and shows the same channel realization.



8.3 Figure 3.1

This figure illustrates the achievable rate region in a downlink scenario with $K_t = 2$ transmitters that serve $K_r = 2$ users by global joint transmission. Each transmitter has 2 antennas and per-array power constraints of $q_l = 10$ dB are used. Each user is "close" to one of the transmitters.

The figure considers a single random channel realization where the channel $\mathbf{h}_{jk}$ between transmitter j and user k is generated as uncorrelated Rayleigh fading. The average single-user SNR $\mathbb{E}\{q_l \|\mathbf{h}_{jk}\|_2^2\}$ is $2q_l$ for the user close to transmitter j and $\frac{2}{3}q_l$ for the other user.

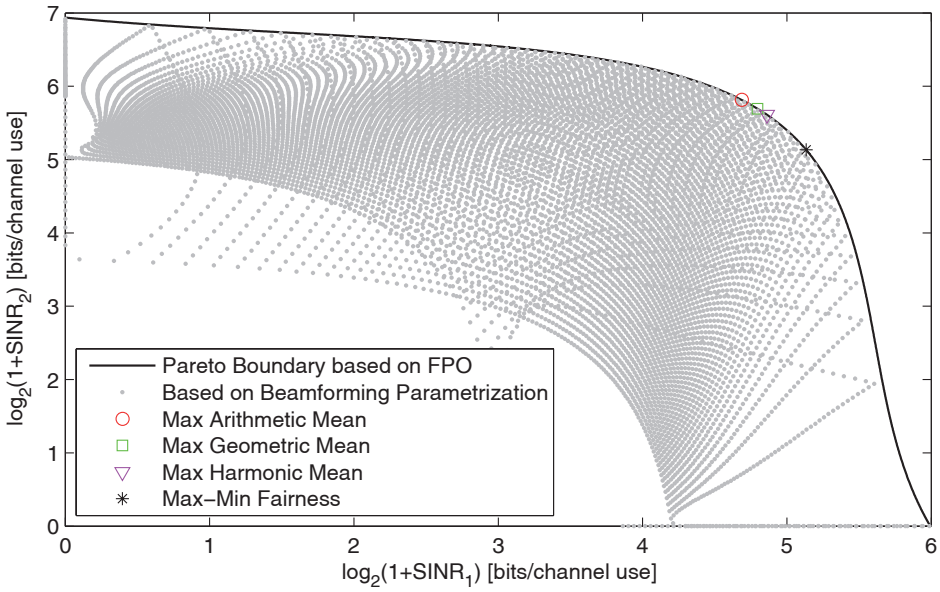
The Pareto boundary of the achievable rate region is approximated in two ways:

1. Calculate sample points on the Pareto boundary by applying Theorem 3.4 in [1] on a fine grid of weighting vectors. The implementation can be described as searching for the Pareto boundary in 1001 equally spaced directions from the origin and draw a line between these points. The implementation uses the function `functionFairnessProfile_cvx.m` to find the Pareto boundary in each sample direction; this is equivalent to solving the weighted max-min fairness problem for a grid of different weights.

2. Calculate sample points of the whole rate region by using the beamforming parametrization in Section 3.2.3 of [1]. The positive parameters $\lambda_1, \lambda_2, \mu_1, \mu_2$ in this beamforming parametrization are varied between 0 and 1 in 101 equally spaced steps. Note that Corollary 3.2 shows that the parameters satisfy $\lambda_1 + \lambda_2 = 1$ and $\mu_1 + \mu_2 = 1$, thus we only need to vary λ_1, μ_1 since λ_2, μ_2 are given as $\lambda_2 = 1 - \lambda_1$ and $\mu_2 = 1 - \mu_1$.

The sample points (from Approach 1) that maximize the arithmetic mean rate, geometric mean rate, harmonic mean rate, and max-min fairness are also indicated in the figure.

An example figure is shown at the bottom of the page. Please note that the rate region looks different for each channel realization.



8.4 Figure 3.2

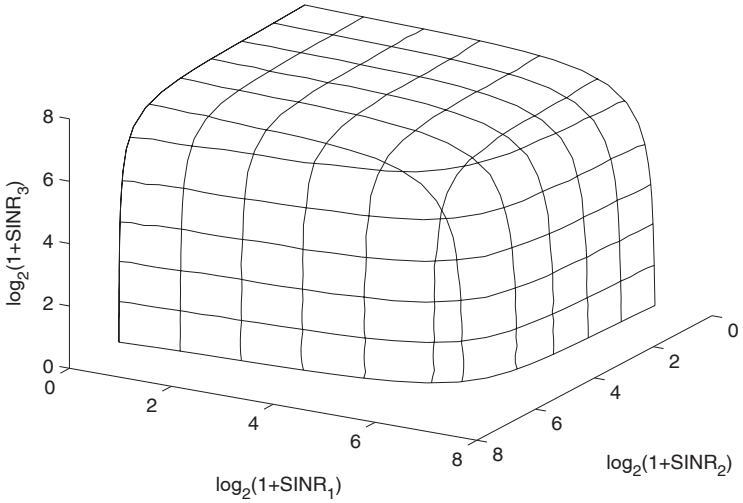
This figure illustrates the achievable rate region in a downlink scenario with $K_t = 3$ transmitters that serve $K_r = 3$ users by global joint transmission. Each transmitter has 3 antennas and per-array power constraints of 10 dB are used. The channels are generated as in Figure 3.1 on the previous page.

The Pareto boundaries are approximated in two ways:

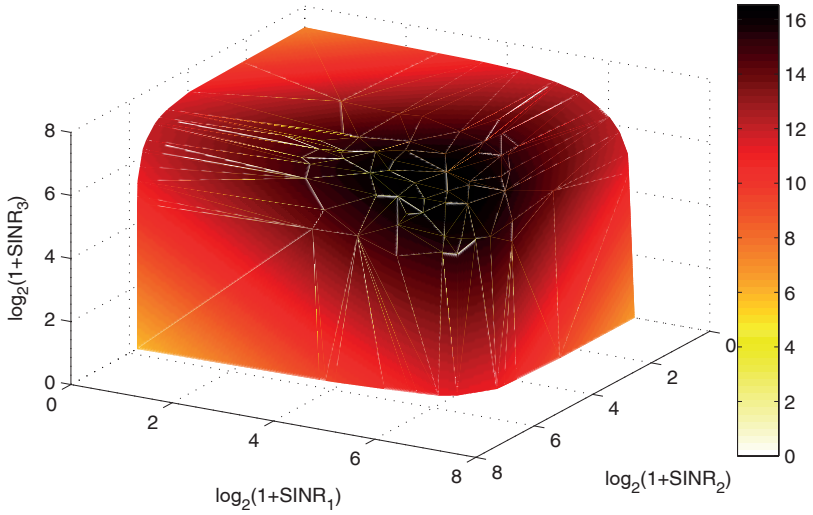
1. Calculate sample points on the Pareto boundary by applying Theorem 3.4 in [1] on a fine grid of weighting vectors. The grid is generated by fixing the performance of one of the users (at 6 different values) and searching for the Pareto boundary in 31 different directions by varying the search direction for the remaining two users. The function `functionFairnessProfile_cvx.m` is used.

2. Calculate sample points of the whole rate region by using the beamforming parametrization in Section 3.2.3 of [1]. The positive parameters $\lambda_1, \lambda_2, \lambda_3, \mu_1, \mu_2, \mu_3$ in this beamforming parametrization are varied between 0 and 1 in 31 equally spaced steps. Note that Corollary 3.2 shows that the parameters satisfy $\lambda_1 + \lambda_2 + \lambda_3 = 1$ and $\mu_1 + \mu_2 + \mu_3 = 1$, thus λ_3, μ_3 are given by the other parameters.

Example figures are shown at the bottom of the page. Please note that the rate region looks different for each channel realization. The color bar in (b) shows the sum rate.



(a) Approach 1: Sample the Pareto boundary using Theorem 3.4 in [1].



(b) Approach 2: Sample the whole rate region using the beamforming parametrization in Section 3.2.3 of [1].

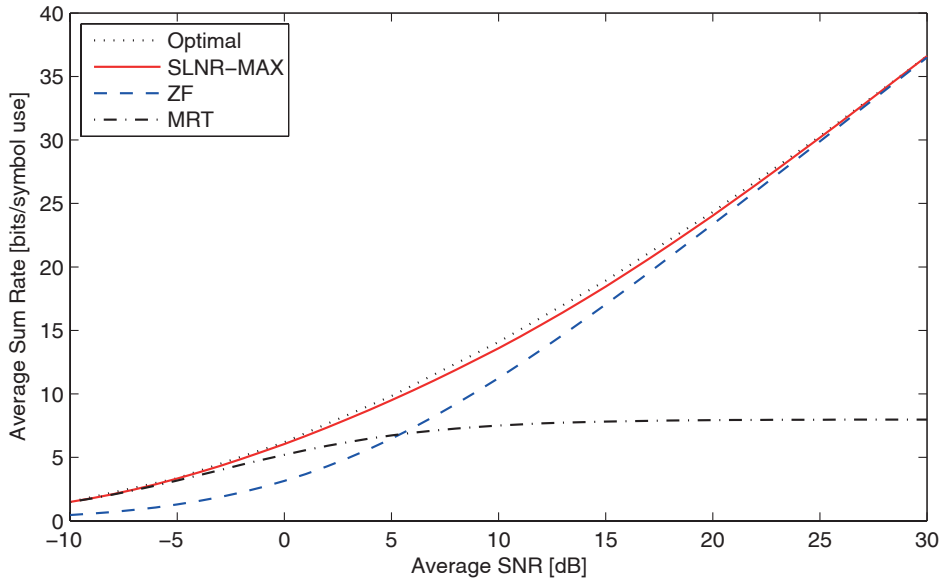
8.5 Figure 3.6

This figure compares the sum rate maximizing linear beamforming with three heuristic beamforming strategies: SLNR-MAX, MRT, and ZFBF. The average sum information rate (over channel realizations) in a 4-user MISO interference channel is shown as a function of the transmit power. The figure illustrates that MRT and ZFBF are good beamforming directions at low and high SNR, respectively, while SLNR-MAX shows good performance in the entire SNR range.

We consider a scenario with $K_t = 4$ transmitters with 4 antennas. Each transmitter serves one of the $K_r = 4$ users. The channel $\mathbf{h}_{jk}$ between transmitter j and user k is generated as uncorrelated Rayleigh fading and the average channel gains $\mathbb{E}\{\|\mathbf{h}_{jk}\|_2^2\}$ equals 4 for the serving transmitter and 2 for all interfering transmitters.

The beamforming that maximizes the sum information rate is computed using the BRB algorithm, implemented in `functionBRBalgorithm_cvx.m`. The heuristic beamforming strategies are generated using the functions `functionMRT.m`, `functionZFBF.m`, and `functionSLNRMAX.m`. These functions give the normalized beamforming directions, while the power allocation is computed using the heuristic waterfilling algorithm in `functionHeuristicPowerAllocation.m`. We refer to Section 3.4 of [1] for definitions and further details on these heuristic strategies.

An example figure is shown at the bottom of the page. Please note that it looks different depending on the channel realizations. This figure shows the average over 10 channel realizations. The heuristic beamforming strategies are very efficient to compute, while the BRB algorithm converges slowly. If hundreds of channel realizations and a wide SNR range are needed, then it will take days or weeks to run the algorithm.



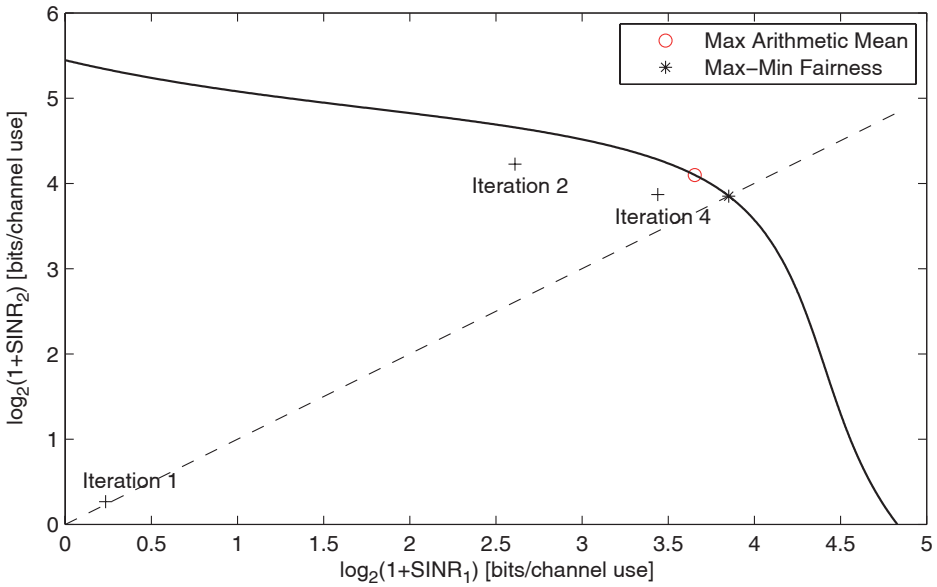
8.6 Figure 4.5

This figure illustrates the the convergence of the distributed resource allocation algorithm in Algorithm 5 in a downlink scenario with $K_t = 2$ transmitters that serve $K_r = 2$ users by global joint transmission. Each transmitter has 2 antennas and per-array power constraints of $q_l = 10$ dB are used. Each user is "close" to one of the transmitters. Max-min fairness is the system utility function and a large percentage of the optimal utility is achieved after a few iterations.

The figure considers a single random channel realization where the channel $\mathbf{h}_{jk}$ between transmitter j and user k is generated as uncorrelated Rayleigh fading. The average single-user SNR $\mathbb{E}\{q_l \|\mathbf{h}_{jk}\|_2^2\}$ is $2q_l$ for the user close to transmitter j and $\frac{2}{3}q_l$ for the other user.

The implementation uses the function `functionFairnessProfile_Distributed_cvx.m` to achieve max-min fairness in a distributed manner, based on dual decomposition. Steps of size $\tau = 0.1$ are taken along the search line, $\xi = \frac{5}{\sqrt{n}}$ is the step size of the subgradient method in the master problem (n is the iteration number in the current QoS problem), and $\epsilon = 0.01$ is the required accuracy in the consistency constraints. For illustrative purposes, sample points on the Pareto boundary of the rate region are also generated. Based on Theorem 3.4 in [1], the samples points are obtained by searching for the Pareto boundary in 1001 equally spaced directions from the origin and draw a line between these points.

An example figure is shown at the bottom of the page. The sample points that maximize the arithmetic mean rate and max-min fairness are also indicated. Please note that the rate region looks different for each channel realization. The code might not provide the same convergence speed for every channel realization; the convergence can be improved by having adaptive step sizes in the algorithm.



8.7 Figure 4.9

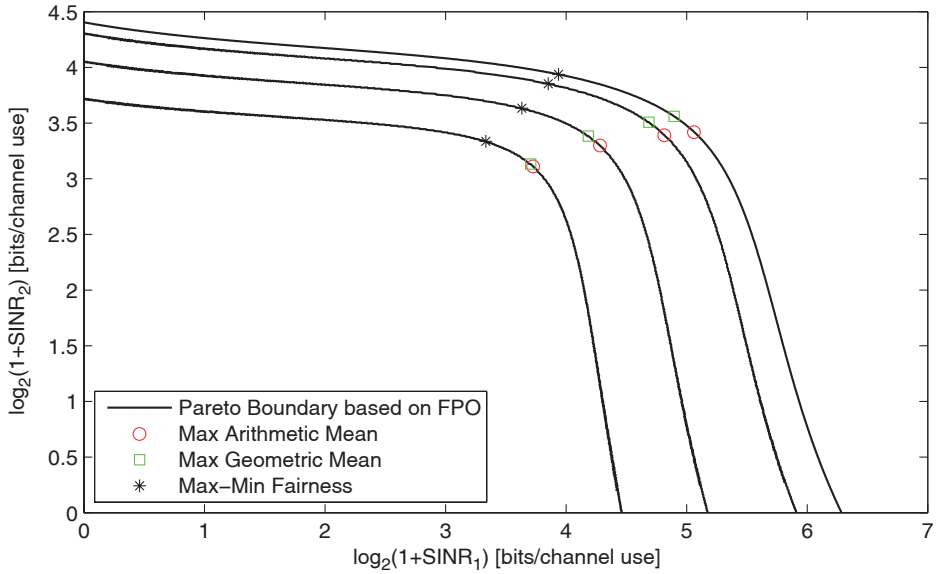
This figure illustrates the achievable rate region in a downlink scenario with transceiver impairments. Apart from that, the scenario is the same as in Figure 3.1. The main conclusion is that the rate region becomes smaller as the level of impairments is increased.

The figure considers $K_t = 2$ transmitters that serve $K_r = 2$ users by global joint transmission. Each transmitter has 2 antennas and per-array power constraints of $q_l = 10$ dB are used. Each user is "close" to one of the transmitters. The rate region is generated for a single random channel realization where the channel $\mathbf{h}_{jk}$ between transmitter j and user k is generated as uncorrelated Rayleigh fading. The average single-user SNR $\mathbb{E}\{q_l \|\mathbf{h}_{jk}\|_2^2\}$ is $2q_l$ for the user close to transmitter j and $\frac{2}{3}q_l$ for the other user.

The transceiver impairments are modeled with linear distortion functions, which is a special case of Corollary 4.3 in [1]. This means that the distortion power is proportional to the signal power. The proportionality constants are $\mathrm{EVM}_{(t)}^2, \mathrm{EVM}_{(r)}^2$ at the transmit antennas and receive antennas, respectively.

The simulation considers $\mathrm{EVM}_{(t)} = \mathrm{EVM}_{(r)} \in \{0, 0.05, 0.1, 0.15\}$. The Pareto boundaries for different levels of impairments are generated by applying Theorem 3.4 in [1] on a fine grid of weighting vectors. The implementation can be described as searching for the Pareto boundary in 1001 equally spaced directions from the origin and draw a line between these points. The implementation uses the function `functionFairnessProfile_cvx.m` to find the Pareto boundary in each sample direction.

An example figure is shown at the bottom of the page. The sample points that maximize the arithmetic mean rate, geometric mean rate, and max-min fairness are also indicated. Please note that the rate region looks different for each channel realization.



References

[1] E. Björnson and E. Jorswieck, "Optimal resource allocation in coordinated multi-cell systems," *Foundations and Trends in Communications and Information Theory*, vol. 9, no. 2-3, pp. 113–381, 2013.

[2] E. Björnson and E. Jorswieck, "Optimal resource allocation in coordinated multi-cell systems: Matlab code," Tech. Rep., July 2014. doi: 10.1561/0100000069_supp, version 1.2.

[3] CVX Research, Inc., "CVX: Matlab software for disciplined convex programming, version 2.0 beta," `http://cvxr.com/cvx`, Sep. 2012.

[4] J. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11-12, pp. 625–653, 1999.

[5] R. Tütüncü, K. Toh, and M. Todd, "Solving semidefinite-quadratic-linear programs using SDPT3," *Mathematical Programming*, vol. 95, no. 2, pp. 189–217, 2003.

[6] H. Tuy, "Monotonic optimization: Problems and solution approaches," *SIAM J. Optim.*, vol. 11, no. 2, pp. 464–494, 2000.

[7] H. Tuy, F. Al-Khayyal, and P. Thach, "Monotonic optimization: Branch and cut methods," in *Essays and Surveys in Global Optimization*, (C. Audet, P. Hansen, and G. Savard, eds.), Springer US, 2005.