Bho Matthiesen

Efficient Globally Optimal Resource Allocation in Wireless Interference Networks

TECHNISCHE UNIVERSITÄT DRESDEN

Efficient Globally Optimal Resource Allocation in Wireless Interference Networks

Bho Matthiesen

von der Fakultät Elektrotechnik und Informationstechnik
der Technischen Universität Dresden

zur Erlangung des akademischen Grades

DOKTORINGENIEUR
(Dr.-Ing.)

genehmigte Dissertation

Vorsitzender:	Prof. Dr.-Ing. habil. Leon Urbas
Gutachter:	Prof. Dr.-Ing. Eduard A. Jorswieck
	Prof. Dr.-Ing. Robert Schober
	Prof. Dr. Markku Juntti
Tag der Einreichung:	08. 08. 2019
Tag der Verteidigung:	14. 11. 2019

Bho Matthiesen
Technische Universität Dresden
Communications Laboratory
Dresden, Germany

Published on Qucosa, Sächsische Landesbibliothek — Staats- und Universitätsbibliothek Dresden, Dresden, Germany with Uniform Resource Name (URN) urn:nbn:de:bsz:14-qucosa2-362878
Available for download at https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa2-362878
Source code is available for download at https://github.com/bmatthiesen

Abstract

Radio resource allocation in communication networks is essential to achieve optimal performance and resource utilization. In modern interference networks the corresponding optimization problems are often nonconvex and their solution requires significant computational resources. Hence, practical systems usually use algorithms with no or only weak optimality guarantees for complexity reasons. Nevertheless, asserting the quality of these methods requires the knowledge of the globally optimal solution. State-of-the-art global optimization approaches mostly employ Tuy's monotonic optimization framework [125] which has some major drawbacks, especially when dealing with fractional objectives or complicated feasible sets.

In this thesis, two novel global optimization frameworks are developed. The first is based on the successive incumbent transcending (SIT) scheme to avoid numerical problems with complicated feasible sets. It inherently differentiates between convex and nonconvex variables, preserving the low computational complexity in the number of convex variables without the need for cumbersome decomposition methods. It also treats fractional objectives directly without the need of Dinkelbach's algorithm. Benchmarks show that it is several orders of magnitude faster than state-of-the-art algorithms.

The second optimization framework is named mixed monotonic programming (MMP) and generalizes monotonic optimization. At its core is a novel bounding mechanism accompanied by an efficient BB implementation that helps exploit partial monotonicity without requiring a reformulation in terms of difference of increasing (DI) functions. While this often leads to better bounds and faster convergence, the main benefit is its versatility. Numerical experiments show that MMP can outperform monotonic programming by a few orders of magnitude, both in run time and memory consumption.

Both frameworks are applied to maximize throughput and energy efficiency (EE) in wireless interference networks. In the first application scenario, MMP is applied to evaluate the EE gain rate splitting might provide over point-to-point codes in Gaussian interference channels. In the second scenario, the SIT based algorithm is applied to study throughput and EE for multi-way relay channels with amplify-and-forward relaying. In both cases, rate splitting gains of up to 4.5 % are observed, even though some limiting assumptions have been made.

Contents

Acknowledgments

Firstly, I am deeply grateful towards my supervisor Professor Eduard Jorswieck for supporting me during these past years. He is a constant source of interesting ideas, research topics, and solution approaches. Most importantly, he sparked my interest in optimization and gave me the necessary freedom and guidance to pursue my own research goals.

My work was funded by the Deutsche Forschungsgemeinschaft (DFG) in the collaborative research center 912 "Highly Adaptive Energy-efficient Computing" (HAEC). Besides the DFG, I would also like to thank Professor Gerhard Fettweis for heading this project and his inspiring keynotes during project meetings, our project manager Nicolle Seifert for her commitment, and the whole HAEC team for interesting discussions, and fruitful collaborations. I also thank the Center for Information Services and High Performance Computing (ZIH) at TU Dresden for their generous allocations of computer time.

I would like to thank Professor Robert Schober and Professor Markku Juntti for serving as reviewers of this dissertation, their encouraging comments, and the insightful discussions during the defense. I also thank Professor Leon Urbas for heading my promotion committee and Karl Besser for writing the protocol.

I am very grateful to Professor Alessio Zappone who is one of my closest collaborators and has taught me a great deal about research and scientific writing. My gratitude extents to Christoph Hellings for developing the mixed monotonic optimization framework with me and his constant critical thinking. I thank Professor Ayding Sezgin, Professor Bruno Clerckx, Professor Tim Davidson, Professor Petar Popovski, Professor Armin Dekorsy, Professor Wolfgang Utschick, Professor Moe Win, and Meik Dörpinghaus for interesting and encouraging discussions regarding my work. I am thankful for successfully collaborating and co-authoring papers with Professor Mérouane Debbah, Professor Wolfgang Utschick, Professor Wolfgang Nagel, Professor Frank Fitzek, Professor Thorsten Strufe, Professor Silvia Santini, Lars Thiele, Yang Yang, Osman Aydin, Stefan Pfennig, Thomas Ilsche, Ievgenii Tsokalo, Giang Nguyen, Elke Franz, Andrew Lonnstrom, Tao Li, Martin Kurras, Mario Bielert, and Juan Cabrera.

I would like to thank my colleagues at TU Dresden for stimulating discussions and their support. I am especially grateful to Martin Mittelbach for teaching me mathematical thinking and information theory, Anne Wolf and Christian Scheunert for their organizational support, Thomas Ilsche for reviewing my branch-and-bound implementation, and Andrew Lonnstrom for proofreading.

Finally, I am deeply grateful to my wife Liz and my family for their substantial support and continuous encouragement. This dissertation would not have been possible without them.

Chapter 1

Introduction

Radio resource management (RRM) is an integral part of communication systems design. Its goal is the optimal allocation of scarce resources, e.g., power and spectrum, with regard to heterogeneous Quality of Service (QoS) demands. Feasible resource allocation methods provide satisfying service quality to all users within the limits of the network. In commercial networks, this ensures customer loyalty and is therefore integral in sustaining the network operator's revenue. Moreover, not being subject to explicit standardization, these methods are one of the key factors for radio equipment manufacturers to differentiate their product from competitors. In the era of 5G and beyond, the progressive move towards more heterogeneous network equipment and QoS demands [10], especially fostered by new concepts like network slicing [91, 102] and tactile internet [33, 113], leads to ever increasing RRM challenges.

Common to most radio resource allocation tasks is that, at its core, a mathematical optimization problem is solved. In real-world systems, a resource allocation usually has to be obtained in very short time frames rendering the optimal solution computationally infeasible. Thus, practical systems mostly employ heuristic algorithms. Nevertheless, from a research and development perspective, obtaining the true solution, at least within a reasonable tolerance, is very desirable. Applications of the optimal solution include evaluating theoretical performance bounds during system and protocol design, feasibility studies, and standardization; benchmarking and verification of heuristic algorithms; and insights on the solution structure that lead to better heuristic algorithms or even to computationally cheap optimal algorithms, e.g., water filling based procedures.

The challenges in solving resource allocation problems vary greatly among the different tasks. Mathematical programs can be categorized in several ways. The most important differentiation is made among the domain of the variables which is either continuous, combinatorial, or integer. Another criterion is based on the properties of the objective function and feasible set, which are, e.g., linear or convex. Of course, combinations of these properties are also possible, e.g., an optimization problem can have continuous and integer variables, or be linear in some variables and nonconvex in other variables. For example, power control and rate adaption are continuous optimization problems, while scheduling is a combinatorial problem and spectrum allocation is a (mixed) integer problem.

This thesis is mainly motivated by power control in wireless interference networks. Depending on the employed interference mitigation techniques, QoS constraints, and utility function this results in a continuous optimization problem with either convex or nonconvex variables. Convex optimization theory is well developed with most problems being solvable in polynomial time. Our focus is mainly on nonconvex optimization problems that are not solvable in polynomial time, also known as global optimization problems. Indeed, many resource allocation problems are known to be NP-hard, meaning that a polynomial time solution algorithm for these problems would settle the $\mathrm{P} = \mathrm{NP}$ question affirmatively. This is considered unlikely and, thus, we focus our attention on developing solution algorithms that are efficient despite their exponential computational complexity.

Any efficient optimization algorithm heavily exploits structural properties of the problem at hand. For example, convex programs are usually categorized into different problem classes like quadratic, geometric, semidefinite, or conical programs to exploit as much structure as possible. Naturally, nonconvex optimization requires very different algorithmic approaches than linear or convex programming. Helpful structural properties that can be exploited to great benefit are partial convexity and monotonicity. Over the past three decades two major global optimization frameworks relying on these properties have emerged from the operations research community. The first was DC programming which depends on the objective and constraints being representable as difference of convex (DC) functions [50]. The second is monotonic programming which came up around the turn of the millennium and, instead of convexity, exploits monotonicity properties in the objective and constraints [125]. Similar to DC programming, both the objective and constraint functions need to be representable as difference of increasing (DI) functions. From a theoretical point of view, both frameworks cover a large class of functions. In particular, it can be shown that they are dense in the space of continuous real-valued functions with supremum norm and, thus, include many nonconvex optimization problems of interest.

Especially monotonic optimization has been widely adopted for the solution of resource allocation problems. In general, global optimization algorithms can be roughly classified into two groups, outer approximation algorithms and branch-and-bound (BB) procedures. Along with the initial paper by Tuy [125], the polyblock outer approximation algorithm was developed to solve monotonic optimization problems. Although it suffers from major impairments it is by far the most widely used algorithm for this framework. Later, a BB method was proposed in [130] that improves upon the polyblock algorithm in almost all aspects. However, the necessity to obtain DI representations for all involved functions limits the usefulness of monotonic programming in practice, and lots of ingenuity went into transforming problems into DI representations. Consider, for example, the maximization of a fractional objective which occurs, e.g., when optimizing the global energy efficiency (GEE) in a communication system. In this case, the numerator is often a DI function while the denominator is increasing. Although the monotonicity in this program is apparent, the fraction renders it almost impossible to obtain the DI representation. For this reason, monotonic programming is usually paired with fractional programming theory, and Dinkelbach's algorithm in particular, resulting in an iterative algorithm that solves several monotonic optimization problems. One of the main contributions of this thesis is to show that this approach is unnecessarily complex. A novel optimization framework exploiting partial monotonicity in a much more general way than classical monotonic programming is proposed. With this mixed monotonic programming (MMP) approach, fractional functions with a DI numerator and increasing denominator can be solved directly without involving Dinkelbach's algorithm. Resource allocation problems that can be cast directly into a monotonic program will most likely also benefit from MMP due to its faster convergence speed.

Common to all these frameworks is that the standard procedures accompanying them can only guarantee convergence to the approximately optimal solution in an infinite number of iterations when the feasible set is complicated.[1] In practice, these procedures usually still converge in finite time but there is no theoretical guarantee. Indeed, one of the key defining properties of an algorithm is its finiteness, i.e., that it always terminates after a finite number of steps [66, §1.1]. Knuth proposes to call "a procedure that has all the properties of an algorithm except that it possibly lacks finiteness [a] computational method."[2] We do not make such fine distinctions and mostly use the words algorithm, procedure,

[1]This statement might call for clarification in several regards. First, most optimization algorithms are unable to obtain the exact optimal solution due to numerical reasons within a finite number of steps. Thus, the usual approach is to accept any point that achieves an objective value sufficiently close to the true optimum as a solution. A notable exception to this are linear programs over the rational numbers [133]. Please refer to the introduction of Chapter 3 for more details. Second, the notion of "complicated feasible sets" will be made precise in Section 3.2.

[2]Source: [66, p. 5].

and method interchangeably. However, the finiteness of each algorithm is stated precisely.

The usual solution to this convergence issue is to relax the feasible set by some small $\varepsilon > 0$ and accept any approximately feasible point that maximizes the objective as a solution. The general assumption is that, for small enough ε, the obtained approximate solution is sufficiently close to the real solution. The issue is that it is impossible to know beforehand how small ε has to be chosen for this to work. In particular, it has been observed in [126] that this approach easily leads to completely wrong "solutions" and, possibly, to numerical problems. One remedy, presently not widely known in the signal processing community is the successive incumbent transcending (SIT) scheme [127]. The core idea is to exchange objective and constraints to obtain a problem with a simple feasible set and then solve the original problem as a sequence of feasibility problems. This scheme can be directly incorporated into a BB procedure resulting in an efficient joint algorithm with finite convergence. Another benefit of the SIT scheme is that it facilitates primal decomposition. This enables the joint treatment of convex and nonconvex variables in a single optimization problem without losing polynomial computational complexity in the convex variables and without the need of cumbersome and problem specific decomposition approaches.

1.1 Overview and Contributions

This thesis is organized into four parts. Important preliminaries are established in Part I. The main contributions are made in Parts II and III, with the exception of Chapter 4, which, although covering preliminaries, is better described as an "optimization framework" and is, therefore, covered in Part II. In particular, two novel global optimization frameworks are developed in Chapters 5 and 6, and subsequently applied to study two different aspects of interference networks in Chapters 7 and 8. Conclusions are given in Part IV. Lists of notation, acronyms and algorithms are provided in the appendix.

Chapter 2 motivates the resource allocation problems that are solved in this thesis. Section 2.1 introduces K-user Gaussian interference channels and discusses some achievability and capacity results. The corresponding resource allocation problem and common utility functions are stated in Section 2.2. Although this thesis is motivated by the optimization problems in this chapter, the developed methods are much more general and applicable to a wide range of engineering and economic problems.

Chapter 3 establishes important global optimization concepts and algorithms. In Section 3.1, we develop a prototype BB algorithm that is the basis for most algorithms in Part II. The treatment is based on

> [128] H. Tuy, *Convex Analysis and Global Optimization*, 2nd ed., ser. Springer Optim. Appl. New York; Berlin, Germany; Vienna, Austria: Springer-Verlag, 2016, vol. 110.

but generalizes where necessary with results from

> [51] R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*, 3rd, rev. and enl. ed. New York; Berlin, Germany; Vienna, Austria: Springer-Verlag, 1996.

The exposition in [51] is very general and probably unsuitable for a first exposure to BB methods. It also requires a considerable amount of adaptation to apply it to a specific problem. Instead, [128] presents the standard approach to BB and focuses mainly on different variants of the bounding procedure. The main contributions of this section are the generalization to different selection procedures, a more modular approach than in [128], and extended, easier to follow, convergence proofs.

Section 3.2 introduces the reader to the SIT scheme and ε-essential feasibility. Again, the treatment follows [128] but, hopefully, is easier to understand. The reader should be advised that this section does not stand on its own and Chapter 5 is required for a complete understanding of the SIT scheme.

In Section 3.3 the Polyblock algorithm for monotonic programming is introduced. We only discuss its most basic properties because it is mainly used for benchmarking and to explain state-of-the-art approaches to globally optimal resource allocation. Nevertheless, the version in Algorithm 3 includes several improvements from the literature to reflect the implemented version employed for numerical experiments in this thesis.

This chapter is concluded by a short treatment of fractional programming and Dinkelbach's algorithm in Section 3.4.

Chapter 4 treats the monotonic optimization framework and its application to radio resource allocation problems. We start with the essentials, mostly following [128], along with easy to grasp resource allocation examples directly explaining how the framework is applied.

In Section 4.2, we derive a BB procedure supported by the material in Section 3.1 to solve generic monotonic optimization problems. Although a similar procedure with infinite convergence was derived in

> H. Tuy, F. Al-Khayyal, and P. T. Thach, "Monotonic optimization: Branch and cut methods", in *Essays and Surveys in Global Optimization*, C. Audet, P. Hansen, and G. Savard, Eds., New York; Berlin, Germany; Vienna, Austria: Springer-Verlag, 2005, ch. 2, pp. 39–78,

we take a slightly more general approach and also discuss the finite case. This section *could* also include the Polyblock algorithm from Section 3.3 but there is good reason not to do so. First, although the Polyblock algorithm is invariably connected to the monotonic optimization framework, the optimization problem considered by the framework is much more general than that solved by this algorithm. Cumbersome transformations discussed in Section 4.1 are necessary to apply it to general DI problems. Second, the BB procedure in Section 4.2 is actually a better fit for DI problems and should therefore be emphasized. Finally, all basic algorithms are discussed in Chapter 4 and there is no reason the Polyblock algorithm should be an exception.

In Section 4.3, fractional monotonic programming, the merger of fractional programming and monotonic optimization theory, is explained. To the best of the author's knowledge, this approach was first applied in

> B. Matthiesen, A. Zappone, and E. A. Jorswieck, "Resource allocation for energy-efficient 3-way relay channels", *IEEE Trans. Wireless Commun.*, vol. 14, no. 8, pp. 4454–4468, Aug. 2015

and subsequently developed by Zappone *et al.* into the framework presented in this section.

In Section 4.4, optimization over rate regions is discussed. Primal decomposition is employed to keep the polynomial computational complexity in the rate variables. The material in this section has been published in

> B. Matthiesen and E. A. Jorswieck, “Weighted sum rate maximization for non-regenerative multi-way relay channels with multi-user decoding”, in *Proc. IEEE Int. Workshop Comput. Adv. Multi-Sensor Adaptive Process. (CAMSAP)*, Curaçao, Dutch Antilles, Dec. 2017.

Chapter 5 develops an easy-to-use resource allocation framework based on the SIT scheme that inherently differentiates between convex and nonconvex variables, preserving the low computational complexity in the number of convex variables. Further highlights of this algorithm are its numerical stability, that it inherently treats fractional objectives without the need of Dinkelbach’s algorithm, and that it is several orders of magnitude faster than state-of-the-art algorithms. To the best of the author’s knowledge, this is the first application of the SIT scheme in the communication systems and signal processing communities. This chapter is mostly based on

> B. Matthiesen and E. A. Jorswieck, “Efficient global optimal resource allocation in non-orthogonal interference networks”, *IEEE Trans. Signal Process.*, vol. 67, no. 21, pp. 5612–5627, Nov. 2019

Chapter 6 presents the MMP framework. The notion of mixed monotonic (MM) functions, a versatile generalization of DI functions, and their properties are introduced in Section 6.1. Then, a BB algorithm based on these MM functions is developed in Sections 6.2 and 6.3. The reduction procedure for DI functions from

> H. Tuy, *Convex Analysis and Global Optimization*, 2nd ed., ser. Springer Optim. Appl. New York; Berlin, Germany; Vienna, Austria: Springer-Verlag, 2016, vol. 110, §11.2.1

is extended to MM functions in Section 6.4. This chapter is concluded by benchmarks of MMP against several monotonic programming approaches in Section 6.5.

Chapter 7 evaluates point-to-point (PTP) codes against rate splitting in Gaussian interference channels. This first application example employs the MMP framework to show that Han-Kobayashi coding might provide better GEE than PTP codes in the moderate interference regime, even under the restricting assumptions of Gaussian codebooks and no time sharing. This chapter is based on

> B. Matthiesen, C. Hellings, and E. A. Jorswieck, "Energy efficiency: Rate splitting vs. point-to-point codes in Gaussian interference channels", in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Cannes, France, Jul. 2019.

Chapter 8 considers multi-way relay channels with amplify-and-forward (AF) relaying. After motivating the model and the use of AF relaying, achievable rate regions with and without rate splitting for discrete memoryless channels are derived in Section 8.1. These results are adapted to Gaussian channels in Section 8.2 and the corresponding resource allocation problems are discussed in Section 8.3. Numerical results that include benchmarks of the SIT based resource allocation framework from Chapter 5 against the state-of-the-art are discussed in Section 8.4. This chapter includes material from

> B. Matthiesen and E. A. Jorswieck, "Efficient global optimal resource allocation in non-orthogonal interference networks", *IEEE Trans. Signal Process.*, vol. 67, no. 21, pp. 5612–5627, Nov. 2019.
>
> B. Matthiesen and E. A. Jorswieck, "Instantaneous relaying for the 3-way relay channel with circular message exchanges", in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Nov. 2015, pp. 475–479
>
> B. Matthiesen, A. Zappone, and E. A. Jorswieck, "Resource allocation for energy-efficient 3-way relay channels", *IEEE Trans. Wireless Commun.*, vol. 14, no. 8, pp. 4454–4468, Aug. 2015
>
> B. Matthiesen and E. A. Jorswieck, "Optimal resource allocation for non-regenerative multiway relaying with rate splitting", in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Kalamata, Greece, Jun. 2018
>
> B. Matthiesen and E. A. Jorswieck, "Weighted sum rate maximization for non-regenerative multi-way relay channels with multi-user decoding", in *Proc. IEEE Int. Workshop Comput. Adv. Multi-Sensor Adaptive Process. (CAMSAP)*, Curaçao, Dutch Antilles, Dec. 2017

Parts of the material in this thesis were published previously in IEEE transactions and conferences. It is reprinted, with permission, from the following publications.

- B. Matthiesen and E. A. Jorswieck, "Efficient global optimal resource allocation in non-orthogonal interference networks", *IEEE Trans. Signal*

Process., vol. 67, no. 21, pp. 5612–5627, Nov. 2019, ©2019 IEEE, primarily in Sections 2.2 and 3.2 and Chapters 5 and 8.

- B. Matthiesen and E. A. Jorswieck, "Global energy efficiency maximization in non-orthogonal interference networks", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Brighton, U.K., May 2019, ©2019 IEEE, primarily in Sections 3.2 and 8.4 and Chapter 5.
- B. Matthiesen and E. A. Jorswieck, "Optimal resource allocation for non-regenerative multiway relaying with rate splitting", in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Kalamata, Greece, Jun. 2018, ©2018 IEEE, primarily in Section 3.2 and Chapters 5 and 8.
- B. Matthiesen and E. A. Jorswieck, "Weighted sum rate maximization for non-regenerative multi-way relay channels with multi-user decoding", in *Proc. IEEE Int. Workshop Comput. Adv. Multi-Sensor Adaptive Process. (CAMSAP)*, Curaçao, Dutch Antilles, Dec. 2017, ©2017 IEEE, primarily in Sections 4.4 and 8.4.
- B. Matthiesen, A. Zappone, and E. A. Jorswieck, "Resource allocation for energy-efficient 3-way relay channels", *IEEE Trans. Wireless Commun.*, vol. 14, no. 8, pp. 4454–4468, Aug. 2015, ©2015 IEEE, primarily in Chapter 8.
- B. Matthiesen, C. Hellings, and E. A. Jorswieck, "Energy efficiency: Rate splitting vs. point-to-point codes in Gaussian interference channels", in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Cannes, France, Jul. 2019, ©2019 IEEE, primarily in Chapters 6 and 7.
- B. Matthiesen and E. A. Jorswieck, "Instantaneous relaying for the 3-way relay channel with circular message exchanges", in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Nov. 2015, pp. 475–479, ©2015 IEEE, primarily in Chapter 8.
- B. Matthiesen, Y. Yang, and E. A. Jorswieck, "Optimization of weighted individual energy efficiencies in interference networks", in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Barcelona, Spain, Apr. 2018, ©2018 IEEE, primarily in Section 2.1.
- B. Matthiesen, O. Aydin, and E. A. Jorswieck, "Throughput and energy-efficient network slicing", in *Proc. Int. ITG Workshop Smart Antennas (WSA)*, Bochum, Germany, Mar. 2018, primarily in Section 2.2.

1.2 State-of-the-Art

Radio resource management is essential in most communication systems and any attempt to a complete literature review of this vast research field is likely to fail. Being aware of this fact, we discuss just a few contributions in this area to obtain a general overview. The text books [40, 52] focus on classical resource allocation techniques mostly for convex problems. A general treatment of convex problems in communications systems is given in [72]. Examples of convex problems are weighted sum rate (WSR) maximization for the multiple-access channel (MAC) in [122], for multi-antenna broadcast channels (BCs) in [55], and for multiple-input multiple-output (MIMO) MACs and BCs in [57]. An overview of nonconvex resource allocation problems is given in [18]. Global optimization for interference limited networks is, e.g., topically covered in the books [9, 138, 154]. Monotonic optimization as a method for resource allocation in interference networks is discussed in [154]. This is extended into a framework for optimal resource allocation in coordinated multicell systems in [9], and a BB approach to WSR maximization is presented in [138]. Further contributions in this area are [21], where joint beamformer design in MIMO MACs is considered. In [92], the WSR in multicell MACs with interference coordination is maximized, and [115] discusses joint power and subcarrier allocation in non-orthogonal multiple access (NOMA) systems.

Energy-efficient resource allocation has gained a lot of attention in the past few years. A good starting point is [149] which focuses on the application of fractional programming theory to energy-efficient resource allocation in wireless networks. The first publications to consider the GEE and the corresponding fractional programming problems for centralized resource allocation are [53], where the energy efficiency (EE) for multicarrier transmission links is maximized, and [90] where the EE in multicell orthogonal frequency-division multiple access (OFDMA) systems with limited backhaul capacity is considered. Further works consider EE maximization in distributed relay-assisted interference networks [148], in MIMO BCs [140], for heterogeneous services in OFDMA downlinks [142], in two-hop MIMO interference channels (ICs) [150], in three-way relay channels [83], in MIMO under- and overlay cognitive radio systems [151], and for 5G wireless technologies [152]. Energy-efficient beamforming and precoding in multicell multi-user systems is considered in [41, 42], respectively. Beamforming in a similar scenario but a with realistic power consumption model is studied in [119].

Monotonic optimization, especially in combination with the Polyblock algorithm, is possibly the most widely employed global optimization framework for resource allocation in wireless systems. It was first used in [98] for power control in interference networks. Almost simultaneously it was employed for beamforming in the two-user multiple-input single-output (MISO) IC in [58, 59]. In [132], the polyblock algorithm is used for coordinated beamforming in multicell networks. The combination of fractional programming and monotonic optimization was first proposed in [83]. Later, this approach was generalized to Gaussian interference networks and named fractional monotonic programming in [147]. Optimal joint power and subcarrier allocation for full-duplex multicarrier NOMA systems is considered in [115]. This work is extended to secure MISO systems under channel state information uncertainty in [116].

BB algorithms are one of the most versatile and widely used tools in global optimizations. Possibly the first BB algorithm is proposed in [68] for integer programming and a computationally successful implementation of the traveling salesman problem is presented in [70]. Today, virtually all commercial mixed-integer linear programming solvers rely on BB algorithms [99, 139]. The first continuous BB method is developed in [32, 114] for separable nonconvex functions relying on convex envelopes for bounding. It is generalized in [8, 46, 47] and successfully applied to concave minimization in [45]. All these early approaches suffer from a computationally costly bounding procedure based on subfunctionals of the objective and other algorithmic problems. These are fixed in [48] which develops a fairly general BB concept with convergence conditions that allow for a wide range of bounding approaches. It also incorporates the conical BB variants developed in [121, 131]. Together with [49, 129], [48] builds the foundation of modern BB algorithms for continuous optimization problems as they are taught in textbooks [51, 128]. Applications of BB for radio resource allocation are in [101], where a nonconvex utility maximization problem for Gaussian MISO BCs and ICs is solved by transforming the problem such that the bounding procedure is a convex program; [44], where energy-efficient rate balancing in BCs is solved with a bounding procedure relying on the principles behind MMP; [120], where energy-efficient transmit beamforming in multi-user MISO downlinks is solved with a monotonic optimization inspired BB procedure; [43] to optimize treating interference as noise (TIN) transmission strategies in MISO BCs; and [137], where the WSR is maximized for a set of interfering links.

The SIT approach was developed and successively refined by Hoang Tuy in [126–128, 130]. To the best of the author's knowledge, it has not been adopted for resource allocation problems before. However, the importance of robust feasible sets has been noted in [97] where beamforming in a cognitive radio network is solved with DC programming. In [141], the basic principle of the SIT approach is used to solve a multi-objective optimization problem (MOP).

Primal decomposition approaches [96] are widely used. For example, in [95] the design of linear transceivers for multicarrier MIMO channels is considered. This challenging nonconvex problem is solved by primal decomposition into a convex outer problem and inner problems with closed-form solutions. The authors of [61] combine successive convex approximation and primal decomposition to solve the sum rate maximization problem with QoS constraints for interfering broadcast channels with first order optimality. A distributed algorithm for coordinated beamforming in multicell multigroup multicast systems is developed in [118] based on primal decomposition and semidefinite relaxation.

This is just a very short and, naturally, very incomplete overview of some publications related to this thesis. Nevertheless, it should provide interested readers with a good starting point to get involved in the covered topics.

Part I

Preliminaries

Chapter 2

System Model & Resource Allocation

2.1 Gaussian Interference Networks

Consider the K-user discrete-time Gaussian interference channel (GIC) illustrated in Fig. 2.1 with input-output relation

$$y_k = \sum_{j=1}^{K} h_{kj} x_j + z_k \tag{2.1}$$

where $h_{kj} \in \mathbb{C}$ is the channel coefficient from transmitter j to receiver k, x_j is the complex-valued channel input of transmitter j, y_k is the received symbol at receiver k, and z_k is circularly-symmetric complex Gaussian noise with zero mean and power σ_k^2. We assume quasi-static block flat fading, i.e., the channel coefficients are constant over n symbols. Thus, the channel output in transmission time $i \in \{1, 2, \dots, n\}$ is

$$y_{ki} = \sum_{j=1}^{K} h_{kj} x_{ji} + z_{ki}$$

where $\{z_{ki}\}_i$ is a circularly-symmetric complex white Gaussian noise process with average power σ_k^2, independent of the channel inputs and other noise processes. Channel inputs are subject to an average power constraint

$$\sum_{i=1}^{n} |x_{ki}|^2 \le n\bar{P}_k$$

for all $k = 1, 2, \dots, K$.

A $(2^{nR_1}, 2^{nR_2}, \dots, 2^{nR_K}, n)$ code for the K-user GIC consists of

- K message sets $\mathcal{M}_k = \{1, \dots, 2^{nR_k}\}$, one for each user $k = 1, 2, \dots, K$,
- K encoders, where encoder k assigns a codeword $\boldsymbol{x}_k(m_k)$ of length n to each message $m_k \in \mathcal{M}_k$,
- K decoders, where decoder k assigns an estimate $\hat{m}_k \in \mathcal{M}_k$ or an error message e to each pair $(m_k, \boldsymbol{y}_k)$ where $\boldsymbol{y}_k$ is a length-n vector.

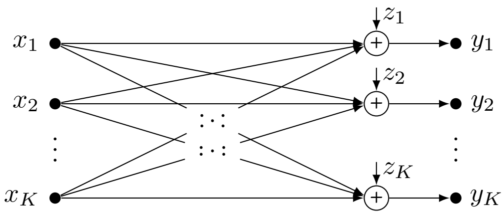


Figure 2.1: Illustration of a K-user Gaussian interference channel.

We assume that the message tuple $(M_1, M_2, \dots M_K)$ is uniformly distributed over $\mathcal{M}_1 \times \mathcal{M}_2 \times \cdots \times \mathcal{M}_K$. The average probability of error is defined as

$$P_e^{(n)} = \Pr\left\{\hat{M}_k \neq M_k \text{ for some } k = 1, 2, \dots, K\right\}.$$

A rate tuple $(R_1, R_2, \dots, R_K)$ is said to be achievable if there exists a sequence of $(2^{nR_1}, 2^{nR_2}, \dots, 2^{nR_K}, n)$ codes such that $\lim_{n\to\infty} P_e^{(n)} = 0$. The capacity region of the K-user GIC is the closure of the set of achievable rates and unknown except for some special cases.[3] One such special case is the PTP capacity region where codes are restricted to Gaussian-PTP (G-PTP) codes.

A $(2^{nR_1}, 2^{nR_2}, \dots, 2^{nR_K}, n)$ G-PTP code is defined as a $(2^{nR_1}$, $2^{nR_2}, \dots$, 2^{nR_K}, $n)$ code where the codewords $\boldsymbol{x}_k(m_k) = (x_{k1}(m_k), \dots, x_{kn}(m_k))$ are independent and identically distributed (i.i.d.) $\mathcal{CN}(0, P_k)$ sequences for some $P_k \in [0, \bar{P}_k]$ and all $k = 1, 2, \dots, K$, where $\mathcal{CN}(\mu, \sigma^2)$ denotes the circularly-symmetric complex Gaussian distribution with mean μ and variance σ^2. Consider the kth receiver and assume it jointly decodes a subset $\mathcal{S} \subseteq \{1, 2, \dots, K\}$ of codewords and treats the remaining x_j, $j \in \mathcal{S}^c = \{1, 2, \dots, K\} \setminus \mathcal{S}$, as noise. This is a MAC with capacity region

$$\mathcal{A}_k(\mathcal{S}) = \left\{ (R_j)_{j\in\mathcal{S}} \geq \mathbf{0} \,\middle|\, \sum_{i\in\mathcal{T}} R_i \leq \mathrm{C}\left(\frac{P_k(\mathcal{T})}{\sigma_k^2 + P_k(\mathcal{S}^c)}\right) \text{ for every } \mathcal{T} \subseteq \mathcal{S} \right\} \tag{2.2}$$

where $\mathrm{C}(x) = \log(1 + x)$ and $P_k(\mathcal{S}) = \sum_{j\in\mathcal{S}} |h_{kj}|^2 P_j$. The following proposition from [4] establishes the G-PTP capacity region of the GIC.

[3]Please refer to Chapter 7 for capacity results of the 2-user GIC.

2.1 Proposition ([4, Thm. 1]). The capacity region of the K-user GIC with G-PTP codes is

$$\mathcal{C}_{\mathrm{ptp}} = \bigcap_{k=1}^{K} \bigcup_{\mathcal{S} \in \mathscr{S}_k} \{ \boldsymbol{R} \,|\, (R_j)_{j \in \mathcal{S}} \in \mathcal{A}_k(\mathcal{S}) \} \tag{2.3}$$

where $\mathscr{S}_k$ contains all $\mathcal{S}$ such that $k \in \mathcal{S}$, i.e.,

$$\mathscr{S}_k = \{ \mathcal{S} \cup \{k\} \,|\, \mathcal{S} \subseteq \{1, 2, \dots, K\} \setminus \{k\} \}. \qquad \triangleleft$$

Observe that this definition of G-PTP codes does not include time sharing. However, Proposition 2.1 is strengthened in [5, Thm. 2] where the authors show that $\mathcal{C}_{\mathrm{ptp}}$ is also the capacity region when the choice of codebooks is limited to random PTP codes with coded time sharing. Thus, given that a PTP codebook is employed, the optimal interference mitigation strategy is to either jointly decode an interfering message or treat it as additional noise. In information theory, this is known as simultaneous non-unique decoding (SND) [5].

Combinations of arbitrary ratio between those two interference mitigation techniques are possible by applying rate splitting (RS) and superposition coding. For $K = 2$ transmitter-receiver pairs, this is known as Han-Kobayashi (HK) coding [19, 39] and, again, SND is the best possible strategy for random code ensembles constrained to superposition coding and coded time sharing [5]. For Gaussian codebooks, the achievable rate region takes the form

$$\left\{ \boldsymbol{R} \geq \boldsymbol{0} \,\middle|\, \boldsymbol{a}_i^T \boldsymbol{R} \leq \sum_{j=1}^{l_i} \lambda_j \,\mathrm{C}\!\left(\frac{\boldsymbol{b}_{ij}^T \boldsymbol{p}}{\boldsymbol{c}_{ij}^T \boldsymbol{p} + \sigma_{ij}^2} \right), \quad i = 1, \dots, m \right\} \tag{2.4}$$

with nonnegative, nonzero vectors $\boldsymbol{a}_i$ and $\boldsymbol{b}_{ij}$, nonnegative vector $\boldsymbol{c}_{ij}$, positive constants σ_{ij}^2, and real λ_i for all i and j. Inequalities between vectors are element-wise and $\boldsymbol{0}$ is an all-zero vector of appropriate dimension. Observe that $\mathcal{C}_{\mathrm{ptp}}$ in (2.3) can be constructed as a finite union of these rate regions, since

$$\mathcal{C}_{\mathrm{ptp}} = \bigcap_{k=1}^{K} \bigcup_{\mathcal{S} \in \mathscr{S}_k} \{ \boldsymbol{R} \,|\, (R_j)_{j \in \mathcal{S}} \in \mathcal{A}_k(\mathcal{S}) \} \tag{2.5a}$$

$$= \bigcup_{\substack{\mathcal{S}_1, \dots, \mathcal{S}_K \\ \in \mathscr{S}_1 \times \dots \times \mathscr{S}_K}} \bigcap_{k=1}^{K} \{ \boldsymbol{R} \,|\, (R_j)_{j \in \mathcal{S}} \in \mathcal{A}_k(\mathcal{S}) \}. \tag{2.5b}$$

and $\bigcap_{k=1}^{K}\{\boldsymbol{R} \,|\, (R_j)_{j\in\mathcal{S}} \in \mathcal{A}_k(\mathcal{S})\}$ is an instance of (2.4).

The achievable rate region in (2.4) (respectively a finite union of rate regions) is fairly general and represents the achievable rate regions of a wide range of interference networks with Gaussian codebooks. However, Gaussian codebooks are not necessarily the optimal choice. Indeed, they often perform poorly if interference is treated as noise. This is due to the competitive nature of the problem. On the one hand, Gaussian inputs maximize the differential entropy and constitute capacity achieving PTP codes, while, on the other hand, Gaussian noise is the worst noise under Gaussian inputs [1]. The optimal codebook for TIN is still subject to ongoing research. For example, in [28] mixed Gaussian codebooks are shown to perform well but these mixed distributions are hard to analyze. Instead, Gaussian codebooks are a common assumption, achieve the (sum) capacity of the 2-user GIC in all cases where it is established, and are of practical interest. Hence, we limit the exposition to these codebooks.

State-of-the-art communication systems mostly employ single-user decoders that treat all interference as noise and operate on PTP codes. The achievable rate region for such a systems is

$$\bigcap_{k=1}^{K}\{\boldsymbol{R} \,|\, R_k \in \mathcal{A}_k(\{k\})\} = \left\{\boldsymbol{R} \,\middle|\, R_k \leq \mathrm{C}\left(\frac{|h_{kk}|^2 P_k}{\sigma_k^2 + \sum_{j\neq k} |h_{kj}|^2 P_j}\right)\right\}. \tag{2.6}$$

The main reasons for only employing single user codes and decoders in modern communication systems are their maturity and commercial availability, ease of implementation, and fairly moderate requirements on the coordination and cooperation among transmitters. Instead, a direct implementation of joint decoding (JD) requires costly multi-user sequence detection. However, this is subject to ongoing research, e.g., in [134] commercially available PTP codes and sliding-window decoding are shown to achieve the whole SND region asymptotically.

Gaussian interference networks are a suitable model for many modern communication systems. Indeed, even systems involving relays often have rate regions of the form (2.4) and sometimes are even adequately modelled by (2.1). One such example is the Gaussian multi-way relay channel (MWRC) with multiple unicast transmissions and AF relaying that will be analyzed in Chapter 8. Some applications are discussed below.

Downlink MISO multicell system

Consider a downlink MISO multicell system with K single antenna terminals and L cells with one base station (BS) each. BS l has n_t antennas and serves the terminals in the nonempty set $\mathcal{C}_l \subset \{1, \dots, K\}$ satisfying $|\mathcal{C}_l| \leq n_t$ and $\mathcal{C}_l \cap \mathcal{C}_{l'} = \emptyset$ for all $l \neq l'$. Without loss of generality (WLOG) we assume that every terminal is associated with exactly one BS, i.e., $\bigcup_{l=1}^{L} \mathcal{C}_l = \{1, \dots, K\}$. Denote the channel vector from BS l to terminal k as $\boldsymbol{h}_{kl}$. BS l employs linear precoding and transmits the signal $\boldsymbol{x}_l = \sum_{k \in \mathcal{C}_l} \sqrt{p_k} \boldsymbol{w}_k s_k$, where s_k is the symbol intended for terminal k, $\boldsymbol{w}_k$ is the corresponding beamforming vector satisfying $\|\boldsymbol{w}_k\| = 1$, and p_k is the allocated power. Each BS is subject to a sum power constraint, i.e., $\sum_{k \in \mathcal{C}_l} p_k \leq \bar{P}_l$ for all l. Let ℓ_k denote the home cell of terminal k. The received signal at this terminal is

$$
\begin{aligned}
y_k &= \boldsymbol{h}_{k\ell_k}^T \boldsymbol{x}_{\ell_k} + \sum_{\substack{l=1 \\ l \neq \ell_k}}^{L} \boldsymbol{h}_{kl}^T \boldsymbol{x}_l + z_k \\
&= \underbrace{\sqrt{p_k} \gamma_{k\ell_k}^{k} s_k}_{\text{desired signal}} + \underbrace{\sum_{\kappa \in \mathcal{C}_{\ell_k} \setminus \{k\}} \sqrt{p_\kappa} \gamma_{k\ell_k}^{\kappa} s_\kappa}_{\text{intra-cell interference}} + \underbrace{\sum_{l \neq \ell_k,\, \kappa \in \mathcal{C}_l} \sqrt{p_\kappa} \gamma_{kl}^{\kappa} s_\kappa}_{\text{inter-cell interference}} + \underbrace{z_k}_{\substack{\text{sink} \\ \text{noise}}}
\end{aligned}
$$

where $\gamma_{kl}^{\kappa} = \boldsymbol{h}_{kl}^T \boldsymbol{w}_\kappa$ is the effective channel and z_k is i.i.d. $\mathcal{CN}(0, \sigma_k^2)$ noise. Due to the linear precoding, this is effectively a K-user IC.

Relay-assisted Multicell Communication

Consider a wireless multicell scenario where K users located near the cell edges desire to communicate among each other. The common method of routing this communication through the respective BSs and over the backhaul network will suffer from high attenuation and near-far effects. This can result in lower data rates, higher transmit powers, or higher spectrum utilization. Instead, a small-cell based on a layer-1 relay close to the cell edges is a simple and cost-efficient method to overcome these issues and improve the QoS for these users. This relay-assisted multicell scenario is an instance of the MWRC discussed in Chapter 8 and illustrated in Fig. 2.2.

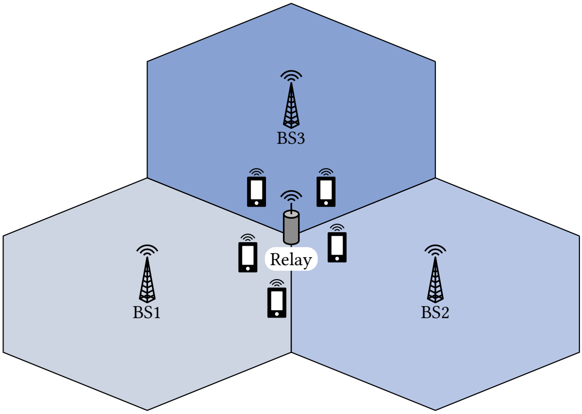


Figure 2.2: Relay-assisted multicell communication [82]. ©2018 IEEE

Digital Subscriber Lines

Digital subscriber line (DSL) systems enable high-speed data transfers over the copper wires used in traditional telephone systems. Crosstalk between adjacent lines has long been identified as the main impairment in theses systems. Thus, the DSL environment is an instance of the GIC. However, in contrast to most wireless systems where flat fading is often a suitable assumption, the DSL channel is highly frequency selective and the model in (2.1) might be an oversimplification [144]. Practical systems use wireline discrete multitone modulation, a multicarrier modulation similar to orthogonal frequency-division multiplexing. In that case, the achievable rate region is of the form (2.4) with l_i being the subcarrier index. Since the number of subcarriers can be as large as 4096 and the duality gap in the corresponding resource allocation problem is asymptotically zero in the number of subcarriers, the resource allocation is usually done in the dual domain [145].

2.2 Resource Allocation Problems

Radio resource allocation is essential to best utilize the available resources in a communication network, e.g., spectrum and energy. We have seen in the previous section that many practical communication systems are modelled as Gaussian interference networks with rate region (2.4). WLOG, the corresponding optimization problem for some utility function $f(\boldsymbol{p}, \boldsymbol{R})$ is

$$\begin{cases} \max\limits_{\boldsymbol{p},\boldsymbol{R}} & f(\boldsymbol{p}, \boldsymbol{R}) \\ \text{s. t.} & \boldsymbol{a}_i^T \boldsymbol{R} \leq \log\left(1 + \dfrac{\boldsymbol{b}_i^T \boldsymbol{p}}{\boldsymbol{c}_i^T \boldsymbol{p} + \sigma_i^2}\right), \quad i = 1, \dots, n \\ & \boldsymbol{R} \geq \boldsymbol{0}, \quad \boldsymbol{R} \in \mathcal{Q} \\ & \boldsymbol{p} \in [\boldsymbol{0}, \bar{\boldsymbol{P}}] \cap \mathcal{P} \end{cases} \tag{2.7}$$

where the sets $\mathcal{Q}, \mathcal{P}$ are usually linear and, often, not present at all, and $[\boldsymbol{0}, \bar{\boldsymbol{P}}]$ is the generalized closed interval defined as

$$[\boldsymbol{0}, \bar{\boldsymbol{P}}] = \{\boldsymbol{p} \,|\, \boldsymbol{0} \leq \boldsymbol{p} \leq \bar{\boldsymbol{P}}\}.$$

The right-hand side (RHS) of the first inequality constraint is a nonconvex function if $\boldsymbol{c}_i \neq \boldsymbol{0}$. Except for some special cases, this is a nonconvex optimization problem and known to be NP-hard for several of the utility functions discussed below. This implies that in these cases obtaining the optimal solution of (2.7) is a challenging computational problem with worst-case computational complexity scaling exponentially in the number of variables.

Problem (2.7) also covers optimizing over $\mathcal{C}_{\text{ptp}}$ in Proposition 2.1. Recall from (2.5) that intersection and union in the definition of $\mathcal{C}_{\text{ptp}}$ can be exchanged by properly adjusting the index sets. The following, almost trivial, lemma connects (2.7) to resource allocation problems over $\mathcal{C}_{\text{ptp}}$.

2.2 Lemma. Let $f : \mathcal{X} \to \mathbb{R}$ and $\mathcal{X} = \bigcup_{i \in \mathcal{I}} \mathcal{X}_i$ for a finite index set $\mathcal{I}$. Then

$$\sup_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}) = \max_{i \in \mathcal{I}} \sup_{\boldsymbol{x} \in \mathcal{X}_i} f(\boldsymbol{x}). \qquad \triangleleft$$

Proof (adapted from [109]). Let $\alpha = \sup_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x})$ and $\alpha_i = \sup_{\boldsymbol{x} \in \mathcal{X}_i} f(\boldsymbol{x})$, $i \in \mathcal{I}$. Clearly, $\alpha \geq \alpha_i$ for all $i \in \mathcal{I}$. Hence, $\alpha \geq \max_{i \in \mathcal{I}} \alpha_i$.

On the other hand, for every $\boldsymbol{x} \in \mathcal{X}$ there exists an $i \in \mathcal{I}$ such that $\boldsymbol{x} \in \mathcal{X}_i$. Thus, $\alpha_i \geq f(\boldsymbol{x})$. This implies that there exists an $i' \in \mathcal{I}$ such that $\alpha_{i'} \geq \sup_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x})$. Since $\max_{i \in \mathcal{I}} \alpha_i \geq \alpha_{i'}$, we have $\max_{i \in \mathcal{I}} \alpha_i \geq \alpha$. $\square$

Hence, maximizing a utility function f over $\mathcal{C}_{\text{ptp}}$ is equivalent to solving a finite number of instances of (2.7). However, the union in (2.5b) is over $2^{K(K-1)}$ elements and optimizing over $\mathcal{C}_{\text{ptp}}$ is not only challenging due to the exponential complexity of (2.7) but also poses an additional combinatorial optimization problem. In this thesis, we focus on the efficient solution of (2.7) and leave the combinatorial problem open for future work.

Besides finding the best operating point with respect to $f(\boldsymbol{p}, \boldsymbol{R})$, solving (2.7) is also necessary to characterize the outermost boundary of the achievable rate region $\mathcal{R}$ in (2.4), i.e.,

$$\partial^+\mathcal{R} = \{\boldsymbol{R} \in \mathcal{R} \,|\, \nexists \boldsymbol{R}' \in \mathcal{R} : (\forall i : R_i' \geq R_i) \wedge (\exists i : R_i' > R_i)\} .$$

In optimization theory, this is known as the *strict Pareto boundary* [146], which is defined as the set of all Pareto-optimal points. A *Pareto-optimal point* is one of an infinite number of noninferior solutions to the MOP (2.7) with vector objective $f(\boldsymbol{p}, \boldsymbol{R}) = (R_1, R_2, \dots, R_K)$. It has the property that it is impossible to increase one of the objectives without decreasing at least one of the others. Multi-objective programming theory provides several approaches to convert the vector objective of a MOP into a scalar one whose maximization results in a Pareto-optimal point. The two most widely used are the scalarization[4] and utility profile approach [153].

Next, we discuss some widely used utility functions $f(\boldsymbol{p}, \boldsymbol{R})$.

2.2.1 Weighted Sum Rate

The WSR $f(\boldsymbol{p}, \boldsymbol{R}) = \boldsymbol{w}^T \boldsymbol{R}$ is possibly the most widely used utility function for power control in wireless communication systems. Its operational meaning is apparent: with all weights $w_k = 1$, $f(\boldsymbol{p}, \boldsymbol{R})$ is the total *throughput* in the network. Without QoS constraints, though, this will often result in very bad performance for some users. Varying the weights between 0 and 1 with $\sum_k w_k = 1$ characterizes the convex hull of the achievable rate region's Pareto boundary. This is known as the scalarization approach from multi-objective programming theory. Moreover, in several communication networks this condition leads to the characterization of the stability region. In this queueing theoretic setting, choosing the weights proportional to the queue lengths prioritizes longer queues and stabilizes the network [87, 117].

While many instances of the WSR maximization problem are convex [55, 57, 122], this is not the case for (2.7). Instead, the problem is known to be NP-hard for TIN [73], and it is a safe assumption that it is also NP-hard for most other instances of (2.7) with nonzero $\boldsymbol{c}_i$. Despite the computational challenges, WSR

[4] This approach is also known as rate balancing [56].

maximization for TIN interference networks is studied widely in the literature, e.g., [9, 21, 59, 92, 98, 115]. A nice property of the TIN case is that the rate region is a hypercube, i.e., WLOG $\boldsymbol{a}_i = \boldsymbol{e}_i$, for all i, where $\boldsymbol{e}_i$ is the ith canonical unit vector. Thus, the rates are easily eliminated from (2.7) and the optimization is only over $\boldsymbol{p}$. This is not the case if, e.g., JD is involved. Then, the rate region is more involved and the optimization is jointly over $\boldsymbol{p}$ and $\boldsymbol{R}$. However, close scrutiny reveals that (2.7) is linear in $\boldsymbol{R}$. Thus, variable reduction techniques should be employed to avoid treating $\boldsymbol{R}$ as global variables and keep the computational complexity at a reasonable level. This is done in Section 4.4 and Chapter 5.

2.2.2 Rate Profile Approach

The rate profile approach is an instance of the utility profile approach and finds the intersection of a ray in the direction $\boldsymbol{w}$ and the Pareto boundary of the achievable rate region, i.e.,

$$\begin{cases} \max\limits_{\boldsymbol{p},\boldsymbol{R},t} & t \\ \text{s.t.} & \boldsymbol{a}_i^T \boldsymbol{R} \leq \log\left(1 + \dfrac{\boldsymbol{b}_i^T \boldsymbol{p}}{\boldsymbol{c}_i^T \boldsymbol{p} + \sigma_i^2}\right), \quad i = 1, \dots, n \\ & \boldsymbol{R} \geq t\boldsymbol{w}, \quad \boldsymbol{R} \in \mathcal{Q} \\ & \boldsymbol{p} \in [\boldsymbol{0}, \bar{\boldsymbol{P}}] \cap \mathcal{P} \end{cases} \tag{2.8}$$

for some $\boldsymbol{w} \geq \boldsymbol{0}$. WLOG one can assume $\|\boldsymbol{w}\| = 1$. By varying the direction of the ray, the complete Pareto boundary can be characterized.

Problem (2.8) is the epigraph form of (2.7) with $f(\boldsymbol{p}, \boldsymbol{R}) = \min_k \frac{R_k}{w_k}$ as long as $\boldsymbol{w} \neq \boldsymbol{0}$. Besides reducing the number of variables, this reformulation also makes it apparent that the rate profile approach is equivalent to the *weighted max-min fairness* performance function. For $\boldsymbol{w} = \boldsymbol{1}$, it puts very strong emphasis on the weakest user and often results in low spectral efficiency. However, especially for the characterization of the rate region's Pareto boundary this approach might be computationally less challenging than the scalarization approach. For example, in the TIN case (2.8) is equivalent to a generalized linear fractional program and can be solved as a sequence of linear programs with polynomial complexity [73, §III-A-2]. Another example is discussed below in Section 2.2.4.

2.2.3 Global Energy Efficiency

The GEE is the most widely used metric to measure the network EE, a key performance metric in 5G networks and beyond [53, 149, 152]. It is defined as the benefit-cost ratio of the total network throughput in a time interval T and the energy necessary to operate the network during this time, i.e.,

$$\mathrm{GEE} = \frac{TW\sum_k R_k}{T\left(\boldsymbol{\phi}^T\boldsymbol{p} + P_c\right)} = \frac{W\sum_k R_k}{\boldsymbol{\phi}^T\boldsymbol{p} + P_c} \qquad \left[\frac{\mathrm{bit}}{\mathrm{J}}\right],$$

where W is the bandwidth, $\phi_k \geq 1$ are the inverses of the power amplifier efficiencies and P_c is a constant modelling the static part of the circuit power consumption. Using the GEE as a performance metric results in a fractional programming problem [17, 105, 106] that is commonly solved by Dinkelbach's algorithm [26]. This algorithm solves the GEE problem as a sequence of auxiliary problems (2.7) with objective $\sum_k R_k - \lambda(\boldsymbol{\phi}^T\boldsymbol{p} + P_c)$ and parameter λ (cf. Section 3.4). Since the objective includes the sum rate, we conjecture that this problem is also NP-hard if $\boldsymbol{c}_i \neq \boldsymbol{0}$ for at least one i.

2.2.4 Energy Efficiency Region

Similar to the GEE, the individual EE of link k is defined as the benefit-cost ratio of the link's throughput divided by the link's power consumption, i.e., $\mathrm{EE}_k = \frac{R_k}{\phi_k p_k + P_{c,k}}$. Analogue to the achievable rate region, there is also an achievable EE region whose Pareto boundary is the solution to the MOP (2.7) with objective $f(\boldsymbol{p}, \boldsymbol{R}) = (\mathrm{EE}_1, \dots, \mathrm{EE}_K)$ [149]. The same approaches as in Sections 2.2.1 and 2.2.2 can be used to transform this MOP into a scalar optimization problem. In this case, the utility profile approach should be favored because the scalarization approach leads to the sum-of-ratios problem

$$\begin{cases} \max\limits_{\boldsymbol{p},\boldsymbol{R}} & \displaystyle\sum_{k=1}^{K} \frac{w_k R_k}{\phi_k p_k + P_{c,k}} \\ \mathrm{s.\,t.} & \boldsymbol{a}_i^T\boldsymbol{R} \leq \log\left(1 + \dfrac{\boldsymbol{b}_i^T\boldsymbol{p}}{\boldsymbol{c}_i^T\boldsymbol{p} + \sigma_i^2}\right), \quad i = 1, \dots, n \\ & \boldsymbol{R} \geq \boldsymbol{0}, \quad \boldsymbol{R} \in \mathcal{Q} \\ & \boldsymbol{p} \in [\boldsymbol{0}, \boldsymbol{P}] \cap \mathcal{P}. \end{cases}$$

The sum-of-ratios problem in general is one of the most difficult fractional programs and known to be NP-complete [35, 107]. While for other EE problems (e.g., the GEE) first-order optimal solutions are usually observed to be globally optimal [147] this does not hold for the sum-of-ratios case [82].

Instead, with the utility profile approach the objective in (2.7) is the weighted minimum EE $f(\boldsymbol{p}, \boldsymbol{R}) = \min_k \frac{\mathrm{EE}_k}{w_k}$. This is, as is the sum-of-ratios problem, a multi-ratio optimization problem but numerically less challenging. Another benefit of using the utility profile approach over the scalarization approach is that the EE region is nonconvex and the scalarization approach only obtains the convex hull of the Pareto region [147]. The general approach to solving such an optimization problem is to use the generalized version of Dinkelbach's algorithm in [25, 147, 149].

2.2.5 Proportional Fairness

Another well known performance function is proportional fairness where the objective is the product of the rates, i.e., $\prod_k R_k$ [63]. The operating point achieved by this metric is usually almost as fair as the one obtained by max-min fairness but achieves significantly higher throughput. Observe that the objective is log-concave. Thus, we can determine the proportional fair operating point by solving (2.7) with concave objective $f(\boldsymbol{p}, \boldsymbol{R}) = \sum_k \log(R_k)$. For TIN and no QoS constraints, it is shown in [73, Thm. 5] that this problem can be transformed into a convex optimization problem and, thus, be solved in polynomial time.

Chapter 3

Global Optimization

Consider the optimization problem

$$\max_{\boldsymbol{x}\in\mathcal{D}} \; f(\boldsymbol{x}) \tag{3.1}$$

where $f : \mathcal{D} \to \mathbb{R}$ is a continuous function and $\mathcal{D}$ is a compact,[5] nonempty subset of $\mathbb{R}^n$. A point $\boldsymbol{x}^* \in \mathcal{D}$ satisfying $f(\boldsymbol{x}^*) \geq f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{D}$ is called a *global maximizer* of f on $\mathcal{D}$. If $\boldsymbol{x}^*$ only satisfies this condition in an open ε-neighborhood for some $\varepsilon > 0$, i.e., for all $\boldsymbol{x} \in \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x} - \boldsymbol{x}^*\| < \varepsilon\} \cap \mathcal{D}$, it is called a *local maximizer*. When (3.1) is a convex optimization problem, i.e., if f is a concave function and $\mathcal{D}$ a convex set, then every local maximizer of (3.1) is also a global maximizer [100, §27].

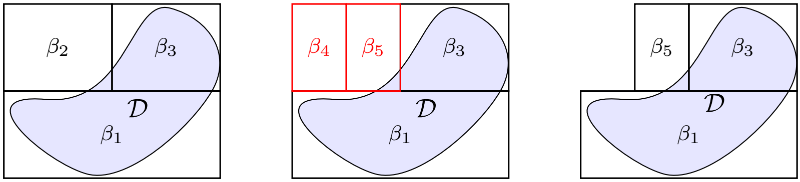
Algorithms designed for convex optimization problems locate at most a local optimizer of (3.1). In fact, most of these algorithms obtain Karush-Kuhn-Tucker (KKT) points. Because the KKT conditions are not only necessary but also sufficient for convex optimization problems, such a point solves Problem (3.1) if it is convex. In general, however, KKT points are not even locally optimal in most instances of (3.1)[6] and provide, at best, candidate solutions for the global solution of (3.1).[7] Indeed, it is shown in [86] that even verifying local optimality in an unconstrained nonconvex optimization problem is NP-hard.

Non-convex optimization problems, henceforth mostly referred to as *global optimization problems*, require quite different algorithmic approaches than (generalized) convex programs. Many global optimization problems are known to be NP-hard and all available solution algorithms have exponential worst-case computational complexity in the number of optimization variables. Although this implies that large-scale problems cannot be solved with global optimality within reasonable time, exploiting structural properties of (3.1) might still lead to efficient algorithms for small- to medium-scale problems. These algorithms can be roughly grouped into BB and outer approximation algorithms. Contin-

[5] A subset of $\mathbb{R}^n$ is compact if it is closed and bounded [103, Thm. 2.41]

[6] Please refer to [128, p. 128] for an example.

[7] Murty and Kabadi [86, §5] make an argument for the case that low complexity algorithms for nonconvex optimization problems should not only converge to a KKT point but also generate an increasing sequence of objective values.



(a) After 2 iterations (b) In iteration 3 (c) After iteration 3

Figure 3.1: Illustration of rectangular subdivision in a branch-and-bound (BB) procedure. The feasible set is denoted as $\mathcal{D}$, the bounds are β_i, and in Fig. 3.1b the upper left box is selected for branching.

uous BB methods are introduced in Section 3.1 and extended to complicated feasible sets in Section 3.2. A very popular outer approximation algorithm for instances of (3.1) with certain monotonic properties, known as the *polyblock algorithm*, is presented in Section 3.3. A notable exception among the class of nonconvex optimization problems are fractional programs which are discussed in Section 3.4.

Determining an exact globally optimal solution to (3.1) is computationally infeasible because, in general, the solution is an irrational number that cannot be represented in finite precision.[8] This is usually not an issue in practice since an approximate solution within a tolerance $\eta > 0$ around the optimal value is often sufficient. We say that a feasible point $\bar{\boldsymbol{x}}$ is a global *η-optimal solution* of (3.1) if it satisfies

$$f(\bar{\boldsymbol{x}}) \geq f(\boldsymbol{x}) - \eta$$

for all $\boldsymbol{x} \in \mathcal{D}$. Most algorithms discussed in the remainder of this chapter obtain such an η-optimal solution.

3.1 Branch-and-Bound

BB algorithms are one of the most versatile and widely used tools in global optimization. The core idea is to relax the feasible set $\mathcal{D}$ and subsequently partition it such that upper bounds on the objective value can be determined easily. During the course of the algorithm, available or easily computable feasible points are evaluated, and the partition of $\mathcal{D}$ is successively refined until each

[8] This is in contrast to, e.g., linear programming where rational input data guarantees that the solution is also a rational number [133, p. 33].

partition set is proven to contain no feasible points or no points with an objective value greater than the current best known feasible solution. Part of this process is illustrated in Fig. 3.1.

A prototype BB algorithm is stated in Algorithm 1. It requires an initial partition set $\mathcal{M}_0$ that contains the feasible set $\mathcal{D}$. During the course of the algorithm, in iteration k, $\mathscr{R}_{k-1}$ holds the partition of the feasible set, $\bar{\boldsymbol{x}}^k$ and γ_k are the current best solution and value, respectively, $\mathscr{S}_k$ holds the partition element(s) selected for branching, and $\mathscr{P}_k$ is an intermediate set that holds the new subdivision obtained from the elements in $\mathscr{S}_k$. We say that this procedure

Algorithm 1 Prototype BB Algorithm

Step 0 **(Initialization)** Choose $\mathcal{M}_0 \supseteq \mathcal{D}$ and $\eta > 0$. Let $k = 1$ and $\mathscr{R}_0 = \{\mathcal{M}_0\}$. If available or easily computable, find $\bar{\boldsymbol{x}}^0 \in \mathcal{D}$ and set $\gamma_0 = f(\bar{\boldsymbol{x}}^0)$. Otherwise, set $\gamma_0 = -\infty$.

Step 1 **(Selection)** Select a subset $\mathscr{S}_k \subseteq \mathscr{R}_{k-1}$ for branching.

Step 2 **(Branching)** For each $\mathcal{M} \in \mathscr{S}_k$, subdivide $\mathcal{M}$ and collect resulting sets in $\mathscr{P}_k$.

Step 3 **(Reduction)** For each $\mathcal{M} \in \mathscr{P}_k$, replace $\mathcal{M}$ by $\mathcal{M}'$ such that $\mathcal{M}' \subseteq \mathcal{M}$ and

$$(\mathcal{M} \setminus \mathcal{M}') \cap \{\boldsymbol{x} \in \mathcal{D} \,|\, f(\boldsymbol{x}) > \gamma_k\} = \emptyset. \tag{3.2}$$

Step 4 **(Bounding)** For each $\mathcal{M} \in \mathscr{P}_k$, compute $\beta(\mathcal{M}) \geq \sup_{\boldsymbol{x} \in \mathcal{M} \cap \mathcal{D}} f(\boldsymbol{x})$. Find $\boldsymbol{x} \in \mathcal{M} \cap \mathcal{D}$ and set $\alpha(\mathcal{M}) = f(\boldsymbol{x})$. If $\mathcal{M} \cap \mathcal{D} = \emptyset$, set $\alpha(\mathcal{M}) = -\infty$.

Step 5 **(Incumbent)** Let $\alpha_k = \max\{\alpha(\mathcal{M}) \,|\, \mathcal{M} \in \mathscr{P}_k\}$. If $\alpha_k > \gamma_{k-1}$, set $\gamma_k = \alpha_k$ and let $\bar{\boldsymbol{x}}^k \in \mathcal{D}$ such that $\alpha_k = f(\bar{\boldsymbol{x}}^k)$. Otherwise, let $\gamma_k = \gamma_{k-1}$ and $\bar{\boldsymbol{x}}^k = \bar{\boldsymbol{x}}^{k-1}$.

Step 6 **(Pruning)** Delete every $\mathcal{M} \in \mathscr{P}_k$ with $\mathcal{M} \cap \mathcal{D} = \emptyset$ or $\beta(\mathcal{M}) \leq \gamma_k + \eta$. Let $\mathscr{P}'_k$ be the collection of remaining sets and set $\mathscr{R}_k = \mathscr{P}'_k \cup (\mathscr{R}_{k-1} \setminus \mathscr{S}_k)$.

Step 7 **(Termination)** Terminate if $\mathscr{R}_k = \emptyset$ or, optionally, if $\{\mathcal{M} \in \mathscr{R}_k \,|\, \beta(\mathcal{M}) > \gamma_k + \eta\} = \emptyset$. Return $\bar{\boldsymbol{x}}^k$ as a global η-optimal solution. Otherwise, update $k \leftarrow k + 1$ and return to Step 1.

is *convergent* if it can be infinite only when $\eta = 0$ and then

$$\lim_{k \to \infty} \gamma_k = \max\{f(\boldsymbol{x}) \,|\, \boldsymbol{x} \in \mathcal{D}\},$$

i.e., the current best value γ_k approaches the global optimum [128, §6.2.1]. In this section, we establish the convergence of Algorithm 1 under some very general assumptions to be discussed next.

First, consider the selection in Step 1 of Algorithm 1.

3.1 Definition ([51, Def. IV.6]). A selection operation is said to be *bound-improving* if, at least each time after a finite number of steps, $\mathscr{S}_k$ satisfies

$$\mathscr{S}_k \cap \operatorname{argmax}\{\beta(\mathcal{M}) \mid \mathcal{M} \in \mathscr{R}_{k-1}\} \neq \emptyset,$$

i.e., at least one partition element where the actual upper bound is attained is selected for further partitioning in iteration k of Algorithm 1. ◁

The most obvious and widely used bound-improving selection criterion is the *best-first* rule

$$\mathscr{S}_k \subseteq \operatorname{argmax}\{\beta(\mathcal{M}) \mid \mathcal{M} \in \mathscr{R}_{k-1}\}, \quad |\mathscr{S}_k| = 1. \tag{3.3}$$

Another option is to always select one of the oldest elements in $\mathscr{R}_{k-1}$, i.e., define for every $\mathcal{M}$ by $\sigma(\mathcal{M})$ the iteration index of its creation and select

$$\mathscr{S}_k \subseteq \operatorname{argmin}\{\sigma(\mathcal{M}) \mid \mathcal{M} \in \mathscr{R}_{k-1}\}, \quad |\mathscr{S}_k| = 1, \tag{3.4}$$

which is named *oldest-first* rule. Due to the finiteness of $\mathscr{R}_k$, every set $\mathcal{M} \in \mathscr{R}_k$ will be deleted or selected after finitely many iterations [51, p. 130]. Hence, the oldest-first rule is bound-improving.

We say that $\{\mathcal{M}_k\}$ is a *decreasing sequence of sets* if, for all k, $\mathcal{M}_{k+1} \subset \mathcal{M}_k$, i.e., $\mathcal{M}_{k+1}$ is a descendent of $\mathcal{M}_k$. The subdivision and bounding procedures in Algorithm 1 are required to satisfy the next definition.

3.2 Definition ([51, Def. IV.4]). A bounding operation is said to be *consistent with branching* (or, simply *consistent*) if at every step any element in $\mathscr{R}_k$ can be further refined, and if any infinitely decreasing sequence $\{\mathcal{M}_{k_q}\}_q$ of successively refined partition elements satisfies

$$\lim_{q\to\infty} (\beta(\mathcal{M}_{k_q}) - \alpha(\mathcal{M}_{k_q})) = 0. \tag{3.5}$$

◁

With these convergence criteria in place we are in the position to formally prove the convergence of Algorithm 1.

3.3 Proposition. If the selection is bound-improving and the bounding operation is consistent, then Algorithm 1 is convergent and the final $\bar{\boldsymbol{x}}^k$ is a global η-optimal solution of (3.1). ◁

Proof. In Step 3, let $\mathcal{D}' = \{\boldsymbol{x} \in \mathcal{D} \mid f(\boldsymbol{x}) > \gamma_k\} \subseteq \mathcal{D}$ and observe that $\mathcal{D} \setminus \mathcal{D}'$ does not contain any solutions better than the current best solution. Thus, if $\mathcal{M}'$ satisfies (3.2), no solutions better than the current incumbent are lost and the reduction does not affect the solution of (3.1).

Observe that $\beta(\mathcal{M}) > -\infty$ for all $\mathcal{M} \in \mathscr{R}_k$ and k since $\mathcal{M} \cap \mathcal{D} \neq \emptyset$ for all $\mathcal{M} \in \mathcal{R}_k$. Thus, if the algorithm terminates in Step 7 and iteration K, then $\beta(\mathcal{M}) \leq \gamma_K + \eta$ for all $\mathcal{M} \in \mathscr{R}_K$ and $\gamma_K > -\infty$. Hence, $\bar{\boldsymbol{x}}_K$ is feasible and

$$\gamma_K = f(\bar{\boldsymbol{x}}_K) \geq \beta(\mathcal{M}) - \eta \geq \sup_{\boldsymbol{x} \in \mathcal{M} \cap \mathcal{D}} f(\boldsymbol{x}) - \eta \tag{3.6}$$

for every $\mathcal{M} \in \mathscr{R}_K$. Now, for every $\boldsymbol{x} \in \mathcal{D}$, either $\boldsymbol{x} \in \mathcal{D} \cap \bigcup_{\mathcal{M} \in \mathscr{R}_K} \mathcal{M}$ or $\boldsymbol{x} \in \mathcal{D} \setminus \bigcup_{\mathcal{M} \in \mathscr{R}_K} \mathcal{M}$. In the first case, $f(\boldsymbol{x}) - \eta \leq f(\bar{\boldsymbol{x}}_K)$ due to (3.6). In the latter case, $\boldsymbol{x} \in \mathcal{M}'$ for some $\mathcal{M}' \in \mathscr{P}_{k'} \setminus \mathscr{P}'_{k'}$ and some k' since $\mathcal{M}_0 \supseteq \mathcal{D}$. Because $\{\gamma_k\}$ is nondecreasing and due to Step 6, $f(\boldsymbol{x}) \leq \beta(\mathcal{M}') \leq \gamma_{k'} + \eta \leq \gamma_K + \eta$. Hence, for every $\boldsymbol{x} \in \mathcal{D}$, $f(\boldsymbol{x}) \leq f(\bar{\boldsymbol{x}}_K) + \eta$.

It remains to be shown that Algorithm 1 is finite. Suppose this is not the case. Then, due to the bound-improving selection, there exists an infinite decreasing subsequence of sets $\{\mathcal{M}_{k_q}\}_q$ such that $\mathcal{M}_{k_q} \in \operatorname{argmax}\{\beta(\mathcal{M}) \mid \mathcal{M} \in \mathscr{R}_{k_q - 1}\}$. Due to consistency, $\beta(\mathcal{M}_{k_q}) - \alpha(\mathcal{M}_{k_q}) \to 0$ as $q \to \infty$. Since $\alpha(\mathcal{M}) \leq \gamma_{k_q} \leq \beta\left(\mathcal{M}_{k_q}\right)$ for all $\mathcal{M} \in \mathscr{R}_{k_q-1}$, this implies $\beta(\mathcal{M}_{k_q}) \to \gamma_{k_q}$. Hence, $\beta(\mathcal{M}_{k_q}) = \gamma_{k_q} + \delta_{k_q}$ with $\delta_{k_q} \geq 0$ and $\lim_{q \to \infty} \delta_{k_q} = 0$. Thus, there exists a $\tilde{K}$ such that $\delta_k \leq \eta$ for all $k > \tilde{K}$, and $\beta(\mathcal{M}) \leq \gamma_k + \eta$ for all $\mathcal{M} \in \mathscr{R}_{k-1}$ and $k > \tilde{K}$. Then, either the algorithm is directly terminated in Step 7 or the remaining sets in $\mathscr{R}_k$ are successively pruned in finitely many iterations until $\mathscr{R}_k = \emptyset$.[9] □

3.4 Remark (Optional reduction). The reduction in Step 3 is optional since choosing $\mathcal{M}' = \mathcal{M}$ satisfies (3.2). On the one hand, a reduction generally results in tighter bounds and, thus, reduces the number of iterations. On the other hand, it increases the computation time per iteration. Thus, it depends heavily on the problem at hand, especially the structure of $\mathcal{D}$, and the implementation of the reduction procedure whether this step speeds up the convergence or not. ◁

3.5 Remark (Relative Tolerance). Algorithm 1 determines an η-optimal solution of (3.1), i.e., a feasible solution $\bar{\boldsymbol{x}}$ of (3.1) that satisfies $f(\bar{\boldsymbol{x}}) \geq f(\boldsymbol{x}) - \eta$ for all $\boldsymbol{x} \in \mathcal{D}$. Instead, by replacing all occurrences of "$\gamma_k + \eta$" in Algorithm 1

[9] This proof incorporates ideas from the proofs of [128, Prop. 6.1], [51, Thm. IV.3], and [124, Prop. 5.6].

with "$(1+\eta)\gamma_k$", the tolerance η becomes relative to the optimal value and the algorithm terminates if the solution satisfies $(1+\eta)f(\bar{\boldsymbol{x}}) \geq f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{D}$. The necessary modifications of Proposition 3.3 are straightforward. ◁

3.1.1 Rectangular subdivision

Subdivision procedures partition an initial set into several subsets of the same type. There are three basic types of subdivisions: simplicial, conical, and rectangular. We focus on rectangular subdivision in this thesis. The interested reader is referred to [128] for information on the other two.

A rectangle (or box) in $\mathbb{R}^n$ is a set

$$\mathcal{M} = [\boldsymbol{r}, \boldsymbol{s}] = \{\boldsymbol{x} \in \mathbb{R}^n \,|\, r_i \leq x_i \leq s_i, (i = 1, 2, \dots, n)\}.$$

There are several ways to partition $\mathcal{M}$ into sub-rectangles. In nonparallel global optimization, the most common method is to partition $\mathcal{M}$ along a hyperplane parallel to one of the facets of $\mathcal{M}$. Specifically, given a point $\boldsymbol{v} \in \mathcal{M}$ and index $j \in \{1, 2, \dots, n\}$, divide $\mathcal{M}$ along the hyperplane $x_j = v_j$. The resulting partition sets are the sub-rectangles

$$\mathcal{M}^- = \{\boldsymbol{x} \in \mathbb{R}^n \,|\, r_j \leq x_j \leq v_j, r_i \leq x_i \leq s_i \ (i \neq j)\} \tag{3.7a}$$

$$\mathcal{M}^+ = \{\boldsymbol{x} \in \mathbb{R}^n \,|\, v_j \leq x_j \leq s_j, r_i \leq x_i \leq s_i \ (i \neq j)\}. \tag{3.7b}$$

This is referred to as a *partition via* $(\boldsymbol{v}, j)$ of $\mathcal{M}$ and illustrated in Fig. 3.2.

Exhaustive Rectangular Subdivision

Possibly the most widely used rectangular subdivision procedure is the standard *bisection*, where a rectangle $\mathcal{M} = [\boldsymbol{r}, \boldsymbol{s}]$ is partitioned via $(\boldsymbol{v}, j)$ with $j \in \operatorname{argmax}_j s_j - r_j$ and $\boldsymbol{v} = \frac{1}{2}(\boldsymbol{s} + \boldsymbol{r})$. The bisection belongs to the larger class of *exhaustive* subdivision procedures. We formally prove this fact after establishing convergence criteria for Algorithm 1 with exhaustive subdivision. Note that exhaustiveness does not necessarily require a rectangular subdivision.

3.6 Definition ([128, §6.1.1]). A subdivision procedure is said to be *exhaustive* if it results in an infinite decreasing sequence of sets $\{\mathcal{M}_k\}$ whose limit $\mathcal{M}_\infty = \bigcap_{k=1}^{\infty} \mathcal{M}_k$ is a singleton, i.e., $\lim_{k\to\infty} \operatorname{diam} \mathcal{M}_k = 0$. ◁

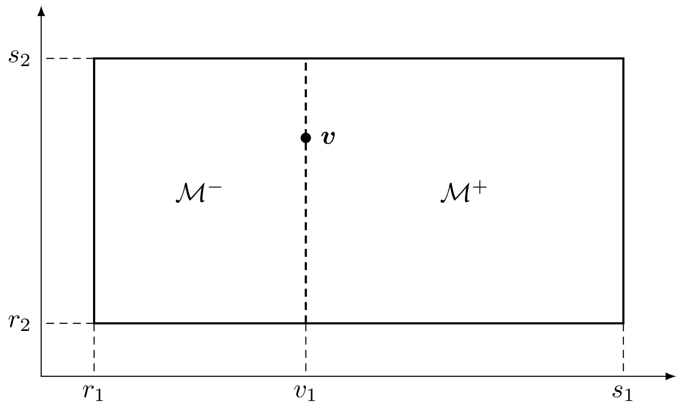


Figure 3.2: Rectangular subdivision in $\mathbb{R}^2$ with $j = 1$ [128, Fig. 6.3].

3.7 Proposition. If the selection is bound-improving, the subdivision is exhaustive and the bounding operation satisfies

$$\beta(\mathcal{M}) - \max\{f(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{M} \cap \mathcal{D}\} \to 0 \tag{3.8}$$

as $\operatorname{diam}(\mathcal{M}) \to 0$, then Algorithm 1 is convergent and the final $\bar{\boldsymbol{x}}^k$ is a global η-optimal solution of (3.1). ◁

Proof. Let $\boldsymbol{x}^{k_q} \in \mathcal{M}_{k_q} \cap \mathcal{D}$ be such that $\alpha(\mathcal{M}_{k_q}) = f(\boldsymbol{x}^{k_q})$. By virtue of the extreme value theorem,[10] there exists a $\boldsymbol{z}^{k_q} \in \mathcal{M}_{k_q} \cap \mathcal{D}$ such that $f(\boldsymbol{z}^{k_q}) = \sup\{f(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{M}_{k_q} \cap \mathcal{D}\}$. The exhaustive subdivision ensures that $\operatorname{diam} \mathcal{M}_{k_q} \to 0$ as $q \to \infty$. Thus, $\left\|\boldsymbol{z}^{k_q} - \boldsymbol{x}^{k_q}\right\| \to 0$ and, due to continuity, $\alpha(\mathcal{M}_{k_q}) - f(\boldsymbol{z}^{k_q}) \to 0$. Hence, condition (3.8) ensures that (3.5) is satisfied. Then, the proposition holds by virtue of Proposition 3.3. □

Exhaustiveness of the bisection procedure follows from the following lemma. Slightly simplifying matters, it basically states that for a sequence of sets obtained by rectangular subdivision via $(\boldsymbol{v}^k, j_k)$, the subdivision points $\{\boldsymbol{v}^k\}$ approach a corner of the resulting rectangles under the condition that j_k is chosen as the index of the dimension with maximum distance between $\boldsymbol{v}^k$ and the corner of the rectangle to be partitioned closest to $\boldsymbol{v}^k$.

[10] The extreme value theorem states that both the maximum and minimum of any continuous mapping on a compact metric space exist [103, Thm. 4.16].

3.8 Lemma ([128, Thm. 6.3]). Let $\{\mathcal{M}_k = [\boldsymbol{r}^k, \boldsymbol{s}^k]\}_{k\in\mathcal{K}}$ be a decreasing sequence of rectangles such that $\mathcal{M}_k$ is partitioned via $(\boldsymbol{v}^k, j_k)$, where $j_k \in \operatorname{argmax}_{j=1,\dots,n} \mu_j^k$ and $\mu_j^k = \min\{v_j^k - r_j^k, s_j^k - v_j^k\}$. Then at least one accumulation point of the sequence of subdivision points $\{\boldsymbol{v}^k\}$ coincides with a corner of the limit rectangle $\mathcal{M}_\infty = \bigcap_{k=1}^\infty \mathcal{M}_k$. ◁

3.9 Corollary ([128, Cor. 6.2]). Let $\{\mathcal{M}_k = [\boldsymbol{r}^k, \boldsymbol{s}^k]\}_{k\in\mathcal{K}}$ be a decreasing sequence of rectangles such that $\mathcal{M}_{k+1}$ is a child of $\mathcal{M}_k$ in a bisection. Then the subdivision procedure is exhaustive, i.e., the diameter $\operatorname{diam}(\mathcal{M}_k)$ of $\mathcal{M}_k$ tends to zero as $k \to \infty$. ◁

Proof. Lemma 3.8 is applicable since, for $\boldsymbol{v}^k = \frac{1}{2}(\boldsymbol{s}^k + \boldsymbol{r}^k)$, we have

$$\begin{aligned}\mu_j^k &= \min\{\frac{1}{2}(s_j^k + r_j^k) - r_j^k, s_j^k - \frac{1}{2}(s_j^k + r_j^k)\}\\ &= \min\{\frac{1}{2}(s_j^k - r_j^k), \frac{1}{2}(s_j^k - r_j^k)\} = \frac{1}{2}(s_j^k - r_j^k).\end{aligned}$$

Thus, $j_k \in \operatorname{argmax}_j s_j^k - r_j^k$ satisfies the condition in Lemma 3.8. Then, by virtue of that lemma, there exists an accumulation point of $\{\boldsymbol{v}^k\}$ that coincides with one corner of the limit rectangle. This is only possible if $\left\|\boldsymbol{s}^k - \boldsymbol{r}^k\right\| \to 0$ as $k \to \infty$, i.e., $\operatorname{diam}(\mathcal{M}^k) \to 0$. □

Adaptive Rectangular Subdivision

Exhaustive rectangular subdivision procedures, especially the bisection, are very easy to implement because they are independent of the optimization problem at hand. They are also slow since they aim to reduce each sequence of rectangles towards a singleton.

Given that for each $\mathcal{M}_k$, two points $\boldsymbol{u}^k, \boldsymbol{v}^k \in \mathcal{M}_k$ are known such that

$$\boldsymbol{u}^k \in \mathcal{M}_k \cap \mathcal{D} \quad \text{and} \quad \beta(\mathcal{M}_k) - f(\boldsymbol{v}^k) \to 0 \text{ as } \left\|\boldsymbol{u}^k - \boldsymbol{v}^k\right\| \to 0, \tag{3.9}$$

a faster method can be devised. Specifically, bisect $\mathcal{M}_k$ via $(\boldsymbol{w}^k, j_k)$ where $\boldsymbol{w}^k = \frac{1}{2}(\boldsymbol{u}^k + \boldsymbol{v}^k)$ and $j_k \in \operatorname{argmax}_j \left|u_j^k - v_j^k\right|$. This is referred to as an *adaptive subdivision via* $(\boldsymbol{u}^k, \boldsymbol{v}^k)$. Then, if $\boldsymbol{u}^k$ and $\boldsymbol{v}^k$ satisfy (3.9), Algorithm 1 is convergent and called an *adaptive BB algorithm*. The following lemma is a consequence of Lemma 3.8 and essential to proving the convergence.

3.10 Lemma ([128, Thm. 6.4]). For each $k = 1, 2, \ldots$, let $\boldsymbol{u}^k$, $\boldsymbol{v}^k$ be two points in the rectangle $\mathcal{M}_k = [\boldsymbol{r}^k, \boldsymbol{s}^k]$ and let $\boldsymbol{w}^k$ and j_k be defined as above. If for every $k = 1, 2, \ldots$, $\mathcal{M}_{k+1}$ is a child of $\mathcal{M}_k$ in the subdivision via $(\boldsymbol{w}^k, j_k)$, then there exists an infinite subsequence $\mathcal{K} \subseteq \{1, 2, \ldots\}$ such that $\left\|\boldsymbol{u}^k - \boldsymbol{v}^k\right\| \to 0$ as $k \in \mathcal{K}$, $k \to \infty$. ◁

For example,[11] let $\beta(\mathcal{M}_k)$ be an upper bound of $f(\boldsymbol{x})$ depending on some approximation point $\boldsymbol{v}^k$ and tight at this point, i.e., $\beta(\mathcal{M}_k)(\boldsymbol{v}^k) = f(\boldsymbol{v}^k)$. Instead of shrinking $\mathcal{M}^k$ until $\boldsymbol{v}^k$ becomes feasible, the adaptive subdivision drives $\boldsymbol{v}^k$ towards a known feasible point $\boldsymbol{u}^k$, resulting in much faster convergence.

3.11 Proposition. If the selection is bound-improving, the subdivision is adaptive, $\boldsymbol{u}^k$ is such that $f(\boldsymbol{u}^k) = \alpha(\mathcal{M}_k)$ and (3.9) is satisfied, then Algorithm 1 is convergent and the final $\bar{\boldsymbol{x}}^k$ is a global η-optimal solution of (3.1). ◁

Proof. By Lemma 3.10 there exists a subsequence $\{k_q\}_q$ such that $\left\|\boldsymbol{u}^{k_q} - \boldsymbol{v}^{k_q}\right\| \to 0$ as $q \to \infty$. Hence, there exists a limit point $\boldsymbol{x}^* \in \mathcal{D}$ such that $\lim_{q\to\infty} \boldsymbol{u}^{k_q} = \lim_{q\to\infty} \boldsymbol{v}^{k_q} = \boldsymbol{x}^*$. From (3.9) it follows that $\beta(\mathcal{M}_{k_q}) \to f(\boldsymbol{x}^*)$, and, due to continuity, $f(\boldsymbol{x}^*) \to \alpha(\mathcal{M}_{k_q})$. Hence, $\beta(\mathcal{M}_{k_q}) \to \alpha(\mathcal{M}_{k_q})$ and the proposition follows from Proposition 3.3.[12] □

3.1.2 Branch selection

The most widely employed branch selection operation is the best-first strategy in (3.3). Rigorous complexity analyses for BB methods, especially in the continuous case, are scarce. However, intuitively it seems obvious that the best-first selection minimizes the number of bound evaluations and leads to fastest convergence of Algorithm 1. This intuition is verified by numerical experiments in Section 6.5, at least when comparing with the oldest-first rule. From a practical perspective, the convergence speed in terms of iterations is an indirect performance measure. More important are the run time and memory consumption of Algorithm 1. Especially in terms of memory consumption, the oldest-first rule (3.4) often outperforms the best-first selection.

In this section, we identify the optimal data structures to implement $\mathcal{R}_k$ with best-first and oldest-first selection and lay the theoretical foundation to understand why the better memory efficiency of the oldest-first rule might lead to shorter average run times despite requiring more bound evaluations.

[11] Based on [128, Ex. 6.1].

[12] This proof follows [128, Prop. 6.2].

Priority Queue

In each iteration of Algorithm 1, one top access, one deletion, and at most two insertion operations on $\mathcal{R}_k$ are necessary. With best-first selection, the top access operation obtains the element in $\mathcal{R}_k$ with the largest bound value. This can be described as a largest in, first out list and is known as a *priority queue* in computer science. A naive implementation of a priority queue $\mathcal{R}$ would be to keep the elements of $\mathcal{R}$ in an ordered list. In that case, accessing the largest element has complexity $O(1)$ but insertion costs $O(n)$ where $n = |\mathcal{R}|$. An alternative approach is storing the elements of $\mathcal{R}$ in arbitrary order and searching for the largest element on every selection. This clearly has insertion complexity $O(1)$ but requires $O(n)$ operations to select the largest element. Instead, heaps are much better suited allowing insertion and deletion with complexity $O(\log n)$, top access with constant complexity $O(1)$, and use minimum memory [67, pp. 148–52], [22, §6.5].[13] Then, the total complexity of operations on $\mathcal{R}_k$ per iteration of Algorithm 1 is $O(\log n)$. This is also the theoretical minimum complexity of implementing (3.3).

3.12 Proposition. Best-first selection has a complexity of $O(\log n)$ per iteration of Algorithm 1, where n is the number of elements in $\mathcal{R}_k$. ◁

Proof. The complexity of $O(\log n)$ is achieved by implementing $\mathcal{R}_k$ as a heap. Now, suppose there exists an algorithm that implements the best-first selection with complexity less than $O(\log n)$. Then, this algorithm could be used to sort n elements with complexity less than $O(n \log n)$, a contradiction to the fundamental result that every comparison sort algorithm has a worst case lower bound on the number of comparisons of $\Omega(n \log n)$ [22, Thm. 8.1]. □

A benefit of implementing $\mathcal{R}_k$ as a priority queue is that the optional termination criterion in Step 7 of Algorithm 1 can be used at virtually no cost since the top access operation has constant complexity.[14] However, the gain is small in practice and often outweighed by its major drawback. The best-first selection rule tends to accumulate a lot of boxes that stay undecided until a solution is found. This requires a large amount of memory. Moreover, due to the exponential complexity of Algorithm 1, the number of accumulated boxes

[13] The C++ standard template library (STL) provides the `priority_queue` container implemented on top of a binary heap [60, §11.9.4, §12.3].

[14] The C++ STL `priority_queue` uses a `vector` or `deque` to store the heap structure. Accessing the first element of these containers comes at the cost of very few pointer dereferences.

also tends to grow exponentially in the number of optimization variables. The result is that deletion and insertion operations on the priority queue become quite costly for a large number of elements in $\mathcal{R}_k$. At some point, this turns into the performance bottleneck of Algorithm 1. Also, even with the large amounts of random access memory in contemporary computer systems, memory usage for larger optimization problems is a major issue. A remedy, especially if the bounding operation is computationally cheap, is the oldest-first selection rule.

Queue

The implementation of the oldest-first selection rule in (3.4) is a first in, first out list commonly known as *queue*. Similarly to the priority queue, it provides insertion, deletion, and access to the front element. These operations all require constant time $O(1)$ [22, §10.1]. Thus, large queues do not suffer from the same performance impairments as priority queues do.

However, the real benefit of the oldest-first rule is that, in practice, the number of elements in $\mathcal{R}_k$ does not grow nearly as fast as with best-first selection. The downside is that more bound evaluations are necessary and the optional termination criterion in Step 7 of Algorithm 1 has prohibitive complexity.

These theoretical observations can only provide a rough guidance on the optimal selection rule. Ultimately, the answer to this question is problem specific and can only be obtained by numerical experiments. One such study is performed in Section 6.5, where we benchmark oldest-first selection against the best-first rule for typical resource allocation problems.

3.2 Successive Incumbent Transcending Scheme

Consider the optimization problem

$$\max_{\boldsymbol{x}\in[\boldsymbol{a},\boldsymbol{b}]} \; f(\boldsymbol{x}) \quad \text{s.t.} \quad g_i(\boldsymbol{x}) \leq 0,\; i = 1, 2, \dots, m \tag{3.10}$$

where $f, g_1, g_2, \dots, g_m$ are continuous real-valued functions and $\boldsymbol{a}, \boldsymbol{b}$ are real-valued vectors satisfying $\boldsymbol{a} \leq \boldsymbol{b}$. This is an instance of (3.1) where the feasible set $\mathcal{D}$ is specified by $[\boldsymbol{a}, \boldsymbol{b}]$ and $g_i(\boldsymbol{x})$, $i = 1, \dots, m$.

Problem (3.10) can be solved with Algorithm 1. This requires that for each box $\mathcal{M} \subseteq [\boldsymbol{a}, \boldsymbol{b}]$, the feasibility problem

$$\text{find } \boldsymbol{x} \in \mathcal{M} \cap \mathcal{D} \tag{3.11}$$

is solved, preferably with reasonably low computational complexity. Unless $\mathcal{D}$ has "nice" structural properties, this is a challenging problem. For example, let g_i be convex (affine) functions, then $\mathcal{D}$ is a convex (linear) set and (3.11) can be solved with polynomial complexity by standard tools from convex optimization [11, 89].[15] Another example of nice feasible sets are normal (conormal) sets where g_i are increasing (decreasing) functions for all $i = 1, \dots, m$ (cf. Lemmas 4.4 and 4.5). A generalization of (co-)normal sets is discussed in Section 6.3.

In many other cases, neither (3.11) nor the test $\mathcal{M} \cap \mathcal{D} \stackrel{?}{=} \emptyset$ can be solved in every iteration of Algorithm 1. Instead, assume there is a procedure available that is able to solve the feasibility problem as $\lim_{k\to\infty} \operatorname{diam} \mathcal{M}_k = 0$. Then, we can alter Algorithm 1 such that, in Step 5, a feasible point is only required if available, and, in Step 6, boxes are only pruned if $\mathcal{M} \cap \mathcal{D}$ is definitely empty. The necessary adjustments to the proof of Proposition 3.3 require a concrete implementation of the feasibility test and are discussed for mixed monotonic programming in Section 6.3.1. For now, the important aspect to note is that these modifications prevent finite convergence of Algorithm 1.

3.2.1 ε-Essential Feasibility

While, practically, an infinite algorithm *might* converge in finite time, there are no theoretical guarantees. A widely accepted method to obtain a finite algorithm is to accept an η-optimal point that is approximately feasible as solution, i.e., a point $\bar{\boldsymbol{x}}(\varepsilon)$ satisfying $f(\bar{\boldsymbol{x}}(\varepsilon)) \geq f(\boldsymbol{x}) - \eta$ for all $\boldsymbol{x} \in \mathcal{D}$ and $g_i(\bar{\boldsymbol{x}}(\varepsilon)) \leq \varepsilon$ for all $i = 1, \dots, m$ and some small $\varepsilon > 0$. Such a point is called *(ε, η)-approximate optimal solution.* Since $\bar{\boldsymbol{x}}(\varepsilon)$ is almost feasible for small $\varepsilon > 0$ it converges to a feasible solution as $\varepsilon \to 0$. Because $f(\bar{\boldsymbol{x}}(\varepsilon))$ also tends to $v(3.10)$, which denotes the optimal value of (3.10), $f(\bar{\boldsymbol{x}}(\varepsilon))$ should be close to $v(3.10)$ for a sufficiently small $\varepsilon = \varepsilon_0$. The problem with this approach is that ε_0 is, in general, unknown and hard to determine. Thus, the obtained solution and $f(\bar{\boldsymbol{x}}(\varepsilon))$ can be quite far away from the true optimal value even for a small ε [126]. This becomes apparent from the following example.

3.13 Example (Issues with $\boldsymbol{\varepsilon}$-approximate solutions). Consider a Gaussian 2-user single-input single-output MAC with channel coefficients h_1 and h_2, transmit powers $p_1 \leq P_1$ and $p_2 \leq P_2$, and a minimum total throughput of Q, i.e., $\log(1 + |h_1|^2 p_1 + |h_2|^2 p_2) \geq Q$. Transmitter i, $i = 1, 2$, is eavesdropped

[15] In the convex (linear) case, (3.10) may also contain equality constraints $h_j(\boldsymbol{x}) = 0$, $j = 1, \dots, l$, with h_i affine.

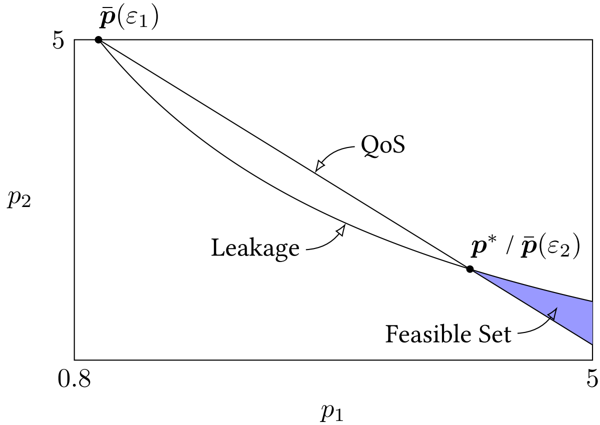


Figure 3.3: Feasible set of Example 3.13 with optimal solution $\boldsymbol{p}^*$ and ε-approximate solutions $\bar{\boldsymbol{p}}(\varepsilon_1)$ and $\bar{\boldsymbol{p}}(\varepsilon_2)$ for $\varepsilon_1 = 10^{-3}$ and $\varepsilon_2 = 10^{-4}$. The ε_1-approximate solution is quite far away from the true optimum [81]. ©2019 IEEE

by a single antenna adversary over a channel g_i. The eavesdroppers are only able to overhear one of the transmitters and do not cooperate with each other [6]. The total information leakage to the eavesdroppers is limited by L, i.e., $\log(1 + |g_1|^2 p_1) + \log(1 + |g_2|^2 p_2) \leq L$. The transmit power of transmitter 1 should be minimized without violating these constraints. The resulting feasible set for $|h_1|^2 = |h_2|^2 = 10$, $|g_1|^2 = \frac{1}{2}$, $|g_2|^2 = 1$, $Q = \log(61)$, and $L = \log(8.99)$ is shown in Fig. 3.3. The true optimum solution is $\boldsymbol{p}^* = (4.00665, 1.99335)$ with $f(\boldsymbol{p}^*) = 4.00665$, while the ε-approximate solution for $\varepsilon_1 = 10^{-3}$ is $\bar{\boldsymbol{p}}(\varepsilon_1) = (0.995843, 5)$ with $f(\bar{\boldsymbol{p}}(\varepsilon_1)) = 0.995843$. This is, obviously, quite far away from both, the optimal solution and value. Instead, the ε-approximate solution obtained for $\varepsilon_2 = 10^{-4}$ is $\bar{\boldsymbol{p}}(\varepsilon_2) = (4.00541, 1.99417)$ with $f(\bar{\boldsymbol{p}}(\varepsilon_2)) = 4.00541$. $\triangleleft$

Even worse, if the optimal solution is an isolated point, i.e., a point at the center of a ball containing no other feasible points, any change in the tolerances ε, η might lead to drastic changes in the (ε, η)-approximate optimal solution. A numerical example illustrating this issue can be found in [127, §4]. The following example picks up the scenario from Example 3.13 and illustrates that isolated feasible points may exist even in simple resource allocation problems. Therefore, the outlined feasibility test should be avoided in general and the reader is well advised not to employ the (ε, η)-approximate feasibility approach.

3.14 Example (Isolated optimal solution). Consider Example 3.13 again. With $L = \log(9)$ there exists an isolated feasible point $\boldsymbol{p} = (1, 5)$ (close to $\bar{\boldsymbol{p}}(\varepsilon_1)$ in Fig. 3.3) that also happens to be the optimal solution. However, apart from the algorithmic difficulties in computing it, it might also be unstable under small perturbations of the data and is, thus, quite impractical from an engineering point of view. $\triangleleft$

A common approach to these numerical problems is to assume that the feasible set is robust, i.e., it satisfies $\mathcal{D}^* = \operatorname{cl}(\operatorname{int} \mathcal{D})$, where cl and int denote the closure and interior of a set, respectively. This assumptions is also a prerequisite for several global optimization algorithms to ensure convergence. Unfortunately, this condition is generally very hard to check, so that, in practice, we have to deal with feasible sets where we do not know a priori whether they are robust or not.

This motivates the concept of ε-essential optimality developed by Tuy [126–128]. A solution $\boldsymbol{x}^* \in \mathcal{D}^*$ is called *essential optimal solution* of (3.10) if $f(\boldsymbol{x}^*) \geq f(\boldsymbol{x})$ for all $x \in \mathcal{D}^*$. A point $\boldsymbol{x} \in [\boldsymbol{a}, \boldsymbol{b}]$ satisfying $g_i(\boldsymbol{x}) \leq -\varepsilon$ for all i and some $\varepsilon > 0$ is called *ε-essential feasible* and a solution of (3.10) is said to be *essential (ε, η)-optimal* if it satisfies

$$f(\boldsymbol{x}^*) + \eta \geq \sup\{f(\boldsymbol{x}) | \boldsymbol{x} \in [\boldsymbol{a}, \boldsymbol{b}], \forall i : g_i(\boldsymbol{x}) \leq -\varepsilon\},$$

for some $\eta > 0$. Apparently, for $\varepsilon, \eta \to 0$ an essential (ε, η)-optimal solution is a nonisolated feasible point which maximizes $f(\boldsymbol{x})$ over $\mathcal{D}^*$. The next section introduces an algorithm that implements the essential optimality approach.

3.2.2 Successive Incumbent Transcending Scheme

Solving (3.10) with Algorithm 1 involves two major challenges: finding suitable bounds for the objective and implementing the feasibility check. The SIT scheme provides an effective method to deal with the second task. The main idea is to solve a sequence of easily implementable feasibility problems. Specifically, given a real number γ, the core problem of the SIT algorithm is to check whether (3.10) has a nonisolated feasible solution $\boldsymbol{x}$ satisfying $f(\boldsymbol{x}) \geq \gamma$, or, else, establish that no such ε-essentially feasible $\boldsymbol{x}$ exists. In this manner, a sequence of nonisolated feasible points ("incumbents") with increasing objective value is generated until no point with greater objective value than the current best solution γ exists.

Given that the feasibility subproblem is solved within finitely many steps, this algorithm converges to the global essential optimal solution of (3.10) in finitely many iterations. This algorithm is formally stated in Algorithm 2.

Algorithm 2 SIT Algorithm [128, §7.5.1].

Step 0 Initialize $\bar{\boldsymbol{x}}$ with the best known nonisolated feasible solution and set $\gamma = f(\bar{\boldsymbol{x}}) + \eta$; otherwise do not set $\bar{\boldsymbol{x}}$ and choose $\gamma \leq f(\boldsymbol{x})\ \forall \boldsymbol{x} \in \mathcal{D}$.

Step 1 Check if (3.10) has a nonisolated feasible solution $\boldsymbol{x}$ satisfying $f(\boldsymbol{x}) \geq \gamma$; otherwise, establish that no such ε-essential feasible $\boldsymbol{x}$ exists and go to Step 3.

Step 2 Update $\bar{\boldsymbol{x}} \leftarrow \boldsymbol{x}$ and $\gamma \leftarrow f(\bar{\boldsymbol{x}}) + \eta$. Go to Step 1.

Step 3 Terminate: If $\bar{\boldsymbol{x}}$ is set, it is an essential (ε, η)-optimal solution; else Problem (3.10) is ε-essential infeasible.

The most important part to make this algorithm work is an efficient implementation of the feasibility problem in Step 1. Consider the optimization problem

$$\min_{\boldsymbol{x} \in [\boldsymbol{a}, \boldsymbol{b}]} \max_{i=1,2,\dots,m} g_i(\boldsymbol{x}) \quad \text{s.t.} \quad f(\boldsymbol{x}) \geq \gamma \tag{3.12}$$

where we interchanged the objective and constraints of (3.10). Given that $f(\boldsymbol{x})$ has structural properties allowing for "easy" feasibility tests, (3.12) can be solved efficiently with Algorithm 1. Otherwise, (3.10) can always be reformulated such that $f(\boldsymbol{x})$ is a linear function by transforming it into its epigraph form [11, p. 134]. The following proposition, which is an adapted version of [128, Prop. 7.13], establishes a duality between (3.10) and (3.12) in the sense that the feasibility problem in Step 1 of Algorithm 2 is equivalent to solving (3.12).

3.15 Proposition. For every $\varepsilon > 0$, the ε-essential optimal value of (3.10) is less than γ if the optimal value of (3.12) is greater than $-\varepsilon$. ◁

Proof (adapted from [128, Prop. 7.13]). If the optimal value of (3.12) is greater than $-\varepsilon$, then any $\boldsymbol{x} \in [\boldsymbol{a}, \boldsymbol{b}]$ such that $g_i(\boldsymbol{x}) \leq -\varepsilon$, for all $i = 1, 2, \dots, m$, must satisfy $f(\boldsymbol{x}) < \gamma$. Hence, by the compactness of (3.10)'s feasible set, $\max\{f(\boldsymbol{x}) | \boldsymbol{x} \in [\boldsymbol{a}, \boldsymbol{b}], \forall i : g_i(\boldsymbol{x}) \leq -\varepsilon\} < \gamma$. □

Observe that every point $\boldsymbol{x}'$ in the feasible set of (3.12) with an objective value less than or equal to zero is also a feasible point of (3.10) with an objective value greater than γ. Thus, we can solve (3.10) sequentially by solving (3.12) with a BB method. Each time this BB algorithm finds a feasible point $\boldsymbol{x}'$ with an objective value less than or equal to zero (in (3.12)), the current best value γ is updated with the objective value of $\boldsymbol{x}'$ in the original Problem (3.10) (plus the tolerance η). Then, the BB solver continues solving (3.12) with the updated feasible set until it either finds a new point to update γ or establishes that no solution to (3.12) with an objective value less than or equal to $-\varepsilon$ exists. By virtue of Proposition 3.15, the last feasible point $\boldsymbol{x}'$ that was used to update γ is an (ε, η)-optimal solution of (3.10). This observation will be formalized later in Proposition 5.9.

The convergence of Algorithm 2 is evident from the discussion above. Apart from the improved numerical stability and convergence, Algorithm 2 has another very desirable feature: it provides a good nonisolated feasible (but possibly suboptimal) point even if terminated prematurely. Instead, many conventional algorithms[16] usually outer approximate the solution rendering intermediate solutions almost useless.

3.3 Polyblock Algorithm

Outer approximation algorithms are one of the earliest nonconvex optimization approaches. The basic idea is to solve a sequence of easier relaxed problems that converge to a solution of the original problem [128, §6.3]. A popular instance is the polyblock algorithm that solves the optimization problem

$$\max_{\boldsymbol{x}\in[\boldsymbol{a},\boldsymbol{b}]} \; f(\boldsymbol{x}) \quad \text{s.t.} \quad g(\boldsymbol{x}) \leq 0 \leq h(\boldsymbol{x}), \tag{3.13}$$

where f, g, h are increasing functions. The feasible set of (3.13) is the intersection of the normal set

$$\mathcal{G} = \{\boldsymbol{x} \in [\boldsymbol{a}, \boldsymbol{b}] \,|\, g(\boldsymbol{x}) \leq 0\}$$

and the conormal set

$$\mathcal{H} = \{\boldsymbol{x} \in [\boldsymbol{a}, \boldsymbol{b}] \,|\, h(\boldsymbol{x}) \geq 0\}.$$

A *polyblock* $\mathcal{P}$ is the normal hull of a finite vertex set $\mathcal{V} \subseteq [\boldsymbol{a}, \boldsymbol{b}]$, i.e., $\mathcal{P} = \bigcap_{\boldsymbol{v}\in\mathcal{V}} [\boldsymbol{a}, \boldsymbol{v}]$. A vertex set is called *proper* if for every $\boldsymbol{v} \in \mathcal{V}$, no $\boldsymbol{v}' \in \mathcal{V}$ exists such that $\boldsymbol{v}' \geq \boldsymbol{v}$. Clearly, a polyblock is fully defined by its proper vertex set [128, p. 394]. The algorithm is designed around the following two properties.

[16] A popular example is the polyblock algorithm in the next section.

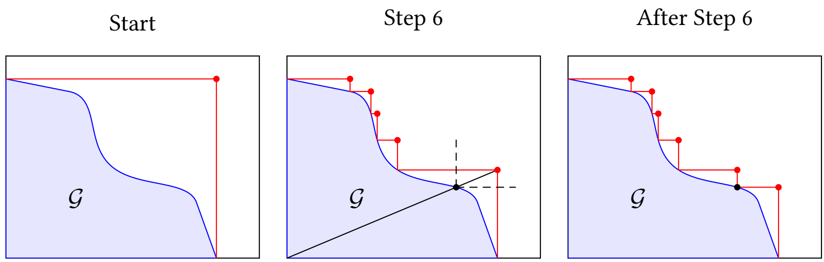


Figure 3.4: Illustration of polyblock feasible set approximation.

1. At least one optimal solution of (3.13) is a Pareto point of $\mathcal{G}$, i.e., $\boldsymbol{x}^* \in \partial^+\mathcal{G}$, if $\mathcal{G} \cap \mathcal{H} \neq \emptyset$ [128, Prop. 11.11].
2. Any compact normal set can be approximated as closely as desired by the intersection of a family of polyblocks [128, Cor. 11.1].

The main idea is to outer approximate the normal part $\mathcal{G}$ of the feasible set by successively refined polyblocks $\mathcal{P}_k \subseteq [\boldsymbol{a}, \boldsymbol{b}]$ until a feasible point $\bar{\boldsymbol{x}} \in \mathcal{G} \cap \mathcal{H}$ is found such that $\max_{\boldsymbol{x} \in \mathcal{P}_k} f(\boldsymbol{x}) \leq f(\bar{\boldsymbol{x}}) + \eta$. The optimization problem $\max_{\boldsymbol{x} \in \mathcal{P}_k} f(\boldsymbol{x})$ is easily solved since, due to Property 1, it is equal to $\max_{\boldsymbol{x} \in \mathcal{V}_k} f(\boldsymbol{x})$, where $\mathcal{V}_k$ is the vertex set of $\mathcal{P}_k$. The refined polyblock $\mathcal{P}_{k+1}$ is obtained from $\mathcal{P}_k$ by projecting the vertex with maximum objective value onto a point $\boldsymbol{z}^k \in \partial^+\mathcal{G}$ and removing the set $(\boldsymbol{z}^k, \boldsymbol{b}]$ from $\mathcal{P}_k$. This procedure is illustrated in Fig. 3.4, where $\mathcal{P}_k$ is displayed in red, its (proper) vertex set $\mathcal{V}_k$ are the red points, and the projection and $\boldsymbol{z}^k$ are shown in black.

The polyblock algorithm is stated in Algorithm 3. It contains several refinements of the basic procedure which are not mentioned in this section. Please refer to [128, §11.1–11.2], [154, §3] for more details. An important prerequisite for convergence of Algorithm 3 is that $\mathcal{H}$ does not contain the origin $\mathbf{0}$, i.e., that the feasible set of (3.13) lies completely in the positive orthant $\mathbb{R}^n_{>0}$. By shifting the origin of (3.13) towards the negative orthant if necessary, this condition is always satisfied. Convergence of Algorithm 3 is formally stated below.

3.16 Proposition ([128, Prop. 11.16]). When infinite Algorithm 3 with $\eta = 0$ generates an infinite sequence $\{\bar{\boldsymbol{x}}^k\}$, every cluster point of this sequence is a globally optimal solution of (3.13). ◁

3.17 Remark (Convergence). With $\eta > 0$, Algorithm 3 is finite only if no conormal constraint $h(\boldsymbol{x})$ is present. Otherwise, even for $\eta > 0$, the generated $\bar{\boldsymbol{x}}^k$ can be infeasible and Algorithm 3 may not provide an η-optimal solution in finitely many steps. Instead, an (ε, η)-approximate optimal solution can be computed in finitely many iterations by modifying Algorithm 3 accordingly. However, then the numerical issues discussed in Section 3.2.1 might occur [128, Remark 11.3]. $\triangleleft$

Algorithm 3 Polyblock Algorithm [128, Alg. POA in §11.2.3]

Step 0 **(Initialization)** Let $\mathcal{P}_1 \supseteq \mathcal{G} \cap \mathcal{H}$ be the initial polyblock and $\mathcal{V}_1$ its proper vertex set. If available, let $\bar{\boldsymbol{x}}^1$ be the best feasible solution available and $\gamma_1 = f(\bar{\boldsymbol{x}}^1)$. Otherwise, let $\gamma_1 = -\infty$. Set $k = 1$.

Step 1 **(Reduction)** Let $\bar{\mathcal{P}}_k$ be a reduced version of $\mathcal{P}_k$ still containing all feasible solutions and $\bar{\mathcal{V}}_k$ be the corresponding proper vertex set. Specifically, discard every $\boldsymbol{z} \in \mathcal{V}_k$ that satisfies $\min\{g(\boldsymbol{z}), h(\boldsymbol{z}), f(\boldsymbol{z}) - \gamma - \eta\} < 0$ and replace every other $\boldsymbol{z} \in \mathcal{V}_k$ with $\boldsymbol{z}' = \boldsymbol{a} + \sum_{i=1}^n \alpha_i (z_i - a_i)\boldsymbol{e}_i$ where

$$\alpha_i = \sup\{\alpha \,|\, 0 \leq \alpha \leq 1, g(\boldsymbol{a} + \alpha(z_i - a_i)\boldsymbol{e}_i) \leq 0\} \quad \text{for all } i = 1, \dots, n.$$

Step 2 **(Termination)** If $\bar{\mathcal{V}}_k = \emptyset$ terminate: if $\gamma_k = -\infty$, the problem is infeasible; otherwise, the current best solution $\bar{\boldsymbol{x}}^k$ is an optimal solution.

Step 3 **(Selection)** Select $\boldsymbol{z}^k \in \operatorname{argmax}_{\boldsymbol{x} \in \bar{\mathcal{V}}_k} f(\boldsymbol{x})$. If $g(\boldsymbol{z}^k) \leq 0$, terminate: $\boldsymbol{z}^k$ is an optimal solution.

Step 4 **(Incumbent)** Compute $\boldsymbol{x}^k = \pi_{\mathcal{G}}(\boldsymbol{z}^k)$, the last point of $\mathcal{G}$ in the line segment from $\boldsymbol{a}$ to $\boldsymbol{z}^k$. Determine the new current best value as $\gamma_{k+1} \leftarrow \max\{\gamma_k, f(\boldsymbol{x}^k)\}$ and set the current best feasible point $\bar{\boldsymbol{x}}^{k+1}$ accordingly.

Step 5 **(Refine Polyblock)** Set $\mathcal{P}_{k+1} = \bar{\mathcal{P}}_k \setminus (\boldsymbol{x}^k, \boldsymbol{b}]$ and compute its vertex set as

$$\mathcal{T} = (\mathcal{V}_k \setminus \mathcal{V}_*) \cup \{\boldsymbol{z}^i = \boldsymbol{z} + (x_i^k - z_i)\boldsymbol{e}_i \,|\, \boldsymbol{z} \in \mathcal{V}_*, i = 1, \dots, n\}$$

where $\mathcal{V}_* = \{\boldsymbol{z} \in \mathcal{V}_k \,|\, \boldsymbol{x}^k < \boldsymbol{z}\}$. Obtain from $\mathcal{T}$ the proper vertex set $\mathcal{V}_{k+1}$ of $\mathcal{P}_{k+1}$ by removing all improper vertices: For every pair $\boldsymbol{z} \in \mathcal{V}_*$, $\boldsymbol{y} \in \mathcal{V}_*^+ = \{\boldsymbol{z} \in \mathcal{V}_k \,|\, \boldsymbol{x}^k < \boldsymbol{z}\}$ compute $J(\boldsymbol{z}, \boldsymbol{y}) = \{j \,|\, z_j > y_j\}$ and if $J(\boldsymbol{z}, \boldsymbol{y}) = \{i\}$ then $\boldsymbol{z}^i$ is an improper element of $\mathcal{T}$.

Step 6 **(Repeat)** Increment k and return to Step 1.

3.4 Dinkelbach's Algorithm

Consider the fractional optimization problem

$$\max_{\boldsymbol{x}\in\mathcal{D}} \frac{f(\boldsymbol{x})}{g(\boldsymbol{x})} \tag{3.14}$$

where $\mathcal{D}$ is a compact and connected subset of $\mathbb{R}^n$, f, g are continuous real-valued functions, and $g(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} \in \mathcal{D}$. This is a nonconvex optimization problem even for affine f, g. Several solution methods exist, like the Charnes-Cooper transformation [17] for linear fractional programs. Possibly the most general and widely known is Dinkelbach's algorithm which solves (3.14) as a sequence of auxiliary problems. Specifically, consider the function

$$\Lambda(\lambda) = \max_{\boldsymbol{x}\in\mathcal{D}} f(\boldsymbol{x}) - \lambda g(\boldsymbol{x}),$$

which is continuous, strictly decreasing and convex in λ. Thus, $\Lambda(\lambda) = 0$ has a unique solution [26, Lemmas 1–4]. The main result of [26] is that $\lambda^* = \frac{f(\boldsymbol{x}^*)}{g(\boldsymbol{x}^*)}$ is the optimal value of (3.14) if and only if $\Lambda(\lambda^*) = f(\boldsymbol{x}^*) - \lambda^* g(\boldsymbol{x}^*) = 0$. Together with the fact that $\Lambda(\lambda^+) \geq 0$ for all $\lambda^+ = \frac{f(\boldsymbol{x}^+)}{g(\boldsymbol{x}^+)}$ with $\boldsymbol{x}^+ \in \mathcal{D}$ [26, Lem. 5], this forms the foundation for Dinkelbach's algorithm stated in Algorithm 4. Convergence is established in [26, §3]. From this convergence proof it can be seen that the algorithm is finite for $\delta > 0$ and the convergence speed is super-linear [149, pp. 241–42]. Thus, with f concave and g convex Problem (3.14) can be solved with polynomial complexity.

Algorithm 4 Dinkelbach's Algorithm

Step 0 Initialize $\delta > 0$, $k = 0$, and $\lambda_0 = \frac{f(\boldsymbol{x}^0)}{g(\boldsymbol{x}^0)}$ if any $\boldsymbol{x}^0 \in \mathcal{D}$ is available and $\lambda_0 = 0$ otherwise.

Step 1 Solve $\Lambda(\lambda_k)$ and let $\boldsymbol{x}^k$ be a solution to this problem.

Step 2 Terminate if $\Lambda(\lambda_k) \leq \delta$. If $\Lambda(\lambda_k) = 0$, $\boldsymbol{x}^k$ is an optimal solution of (3.14).

Step 3 Update $\lambda_{k+1} \leftarrow \frac{f(\boldsymbol{x}^k)}{g(\boldsymbol{x}^k)}$. Increment k and go to Step 1.

A major drawback of Dinkelbach's algorithm is that the termination criterion is unrelated to the optimal value of (3.14), i.e., it does not find a δ-optimal solution of (3.14). Either, the solution is exact, i.e., $\Lambda(\lambda_k) = 0$, and $\boldsymbol{x}^k$ solves

(3.14), or $\boldsymbol{x}^k$ is assumed to be a reasonably good approximate solution. Thus, theoretically, the obtained solution is arbitrarily far away from the solution of (3.14) if δ is not chosen sufficiently small. Another issue is that the auxiliary problem, which is often nonconvex, needs to be solved globally several times. This increases the computational complexity quite unnecessarily as we will discuss in the following chapters.[17]

[17] The convergence proof of Algorithm 4 in [26] relies on $\Lambda(\lambda)$ being strictly decreasing. This property, however, does not hold if the auxiliary problem is only solved approximately. Thus, theoretically, convergence of Algorithm 4 does require an *exact* solution of the auxiliary problem. This is, of course, computationally infeasible. However, experience shows that this is not an issue in practice.

Part II

Optimization Frameworks

Chapter 4

Monotonic Optimization

Monotonic optimization is concerned with the solution of global optimization problems where the objective and constraints are difference of increasing (DI) functions, i.e., the solution of

$$\max_{\boldsymbol{x}\in[\underline{\boldsymbol{x}},\bar{\boldsymbol{x}}]} \quad f^+(\boldsymbol{x}) - f^-(\boldsymbol{x}) \quad \text{s. t.} \quad g_i^+(\boldsymbol{x}) - g_i^-(\boldsymbol{x}) \leq 0, \;\; i = 1, \dots, n, \tag{4.1}$$

where all functions are continuous and increasing for all $\boldsymbol{x} \in [\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}]$. Over the past ten years, it has become the most widely used global optimization framework for resource allocation problems in wireless communications [9, 58, 83, 98, 115, 116, 132, 147, 149, 154]. This is because the rate functions in interference networks (cf. Chapter 2) often have an easily obtainable DI representation.

4.1 Example (Throughput maximization for TIN). Consider maximizing the throughput in a K-user GIC where interference is treated as noise and each user requires a minimum rate $R_{k,\min} \geq 0$. The achievable rate region is given in (2.6) and the optimization problem is an instance of (2.7) with objective $f(\boldsymbol{p}, \boldsymbol{R}) = \sum_k R_k$ and $\mathcal{Q} = \{\boldsymbol{R} \,|\, R_k \geq R_{k,\min},\, k = 1, \dots, K\}$. Due to the rectangular structure of (2.6), the rates can be eliminated from the optimization problem and we obtain

$$\begin{cases} \max\limits_{\boldsymbol{p}\in[\boldsymbol{0},\boldsymbol{P}]} & \sum\limits_k \mathrm{C}\left(\dfrac{|h_{kk}|^2 p_k}{\sigma_k^2 + \sum_{j\neq k} |h_{kj}|^2 p_j}\right) \\ \text{s. t.} & \mathrm{C}\left(\dfrac{|h_{kk}|^2 p_k}{\sigma_k^2 + \sum_{j\neq k} |h_{kj}|^2 p_j}\right) \geq R_{k,\min},\, k = 1, \dots, K. \end{cases} \tag{4.2}$$

This is easily transformed into a monotonic optimization problem of the form (4.1), since the rate functions have the DI representation

$$\mathrm{C}\left(\frac{\boldsymbol{b}_i^T \boldsymbol{p}}{\boldsymbol{c}_i^T \boldsymbol{p} + \sigma_i^2}\right) = \log\Big((\boldsymbol{b}_i + \boldsymbol{c}_i)^T \boldsymbol{p} + \sigma_i^2\Big) - \log\big(\boldsymbol{c}_i^T \boldsymbol{p} + \sigma_i^2\big)\,. \tag{4.3}$$

◁

From a theoretical perspective, (4.1) covers a large class of global optimization problems as stated in the following proposition.

4.2 Proposition ([128, Prop. 11.9]). The following functions are DI functions on $[\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}] \subset \mathbb{R}^n_{\geq 0}$

1. any polynomial, and more generally, any function $\boldsymbol{x} \mapsto \sum_{k=1}^{K} c_k x_1^{a_{k1}} \cdots x_n^{a_{kn}}$ with real-valued, nonnegative exponents a_{ki} and real constants c_k, for all $k = 1, \dots, K$ and $i = 1, \dots, n$.[18]
2. any convex function $f : [\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}] \to \mathbb{R}$.
3. any Lipschitz function $f : [\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}] \to \mathbb{R}$ with Lipschitz constant K.
4. any continuously differentiable function $f : [\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}] \to \mathbb{R}$. $\triangleleft$

Due to the Weierstrass approximation theorem, which states that any continuous real-valued function defined on the interval $[\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}]$ can be uniformly approximated with arbitrary precision by a real-valued polynomial [103, Thm. 7.26], and Proposition 4.2, Property 1, the class of DI function is dense in the space $C[\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}]$ of continuous real-valued functions on $[\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}]$ equipped with the supremum norm. That is, every function in $C[\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}]$ can be approximated arbitrarily closely by a DI function and, thus, monotonic optimization is applicable to all functions in $C[\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}]$.

Practically, however, not all functions covered by Proposition 4.2 have a readily available DI representation. The proof of Proposition 4.2[19] is constructive in the sense that the necessary transformations to obtain a DI representation of a function covered by Proposition 4.2 are stated explicitly, but these are often analytically challenging and may result in very slow convergence.

4.3 Example. Consider GEE maximization in the K-user GIC from Example 4.1. The objective

$$f(\boldsymbol{p}) = \frac{\sum_k \mathrm{C}\left(\frac{|h_{kk}|^2 p_k}{\sigma_k^2 + \sum_{j \neq k} |h_{kj}|^2 p_j}\right)}{\boldsymbol{\phi}^T \boldsymbol{p} + P_c}$$

is a continuously differentiable function on $[\boldsymbol{0}, \bar{\boldsymbol{P}}]$, and, thus, a DI function by virtue of Proposition 4.2. However, there obviously does not exist an easily obtainable DI representation as in Example 4.1. Following the proof of Proposition 4.2 in [128, Prop. 11.9], we first need a constant $M > \sup_{\boldsymbol{x} \in [\boldsymbol{0}, \bar{\boldsymbol{P}}]} \|\nabla f(\boldsymbol{x})\|$.

[18] Some sources call this a *synomial* but this term is not used consistently in the literature.
[19] Please refer to [128, Prop. 11.9].

Then, $g(\boldsymbol{x}) = f(\boldsymbol{x}) + M \sum_{k=1}^{K} p_k$ is an increasing function. Hence, a DI representation of $f(\boldsymbol{x})$ is

$$\underbrace{\frac{\sum_k \mathrm{C}\left(\frac{|h_{kk}|^2 p_k}{\sigma_k^2 + \sum_{j\neq k} |h_{kj}|^2 p_j}\right)}{\boldsymbol{\phi}^T \boldsymbol{p} + P_c} + M \sum_{k=1}^{K} p_k}_{f^+(\boldsymbol{p})} - \underbrace{M \sum_{k=1}^{K} p_k}_{f^-(\boldsymbol{p})}.$$

Clearly, determining M is a challenging analytical task. Moreover, convergence of the chosen solution algorithm will be slow. For example, if (4.1) is solved with a BB algorithm (please refer to Section 4.2.1 below) the bound over the box $[\boldsymbol{r}, \boldsymbol{s}]$ is computed as

$$f^+(\boldsymbol{s}) - f^-(\boldsymbol{r}) = \frac{\sum_k \mathrm{C}\left(\frac{|h_{kk}|^2 s_k}{\sigma_k^2 + \sum_{j\neq k} |h_{kj}|^2 s_j}\right)}{\boldsymbol{\phi}^T \boldsymbol{s} + P_c} + M \sum_{k=1}^{K} s_k - M \sum_{k=1}^{K} r_k.$$

Comparing this to other bounds obtained for this problem, especially in Chapter 6, it is apparent that this bound is not very tight. Moreover, the quality of this bound decreases with increasing M. Thus, M should be as close as possible to the true supremum, rendering the analytical task of determining M even harder. A similar argument can be made for the polyblock algorithm.

Instead, fractional programs with DI numerator and denominator are usually solved by a combination of Dinkelbach's algorithm (cf. Section 3.4) and monotonic optimization. This is known as fractional monotonic programming and discussed in Section 4.3. ◁

4.1 Canonical Monotonic Optimization Problem

The optimization problem in (4.1) with DI objective and constraints is the most general form of a monotonic optimization problem. However, the solution algorithms are mostly designed to solve the *canonical monotonic optimization problem*, which is

$$\max_{\boldsymbol{x} \in [\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}]} \; f(\boldsymbol{x}) \quad \text{s.t.} \quad g(\boldsymbol{x}) \leq 0 \leq h(\boldsymbol{x}), \tag{4.4}$$

with f, g, h being increasing functions. By virtue of [128, Thm. 11.1], any monotonic optimization problem can be transformed into the canonical form

(4.4). First, the system $g_i^+(\boldsymbol{x}) - g_i^-(\boldsymbol{x}) \leq 0$ is equivalent to a single inequality $g(\boldsymbol{x}) = \max_i\{g_i^+(\boldsymbol{x}) - g_i^-(\boldsymbol{x})\} \leq 0$ with $g(\boldsymbol{x})$ being DI (cf. [128, Prop. 11.8]). Thus, (4.1) is equivalent to

$$\max_{\boldsymbol{x}\in[\underline{\boldsymbol{x}},\bar{\boldsymbol{x}}]} \; f^+(\boldsymbol{x}) - f^-(\boldsymbol{x}) \quad \text{s. t.} \quad g^+(\boldsymbol{x}) - g^-(\boldsymbol{x}) \leq 0, \tag{4.5}$$

with $g^+(\boldsymbol{x}) = \max_i\left\{g_i^+(\boldsymbol{x}) + \sum_{j\neq i} g_j^-(\boldsymbol{x})\right\}$ and $g^-(\boldsymbol{x}) = \sum_{i=1}^n g_i^-(\boldsymbol{x})$. Next, introduce two auxiliary variables z and t and write (4.5) as

$$\begin{cases} \max\limits_{\boldsymbol{x},z,t} & f^+(\boldsymbol{x}) + z \\ \text{s. t.} & f^-(\boldsymbol{x}) + z \leq 0 \\ & g^+(\boldsymbol{x}) + t \leq 0 \leq g^-(\boldsymbol{x}) + t \\ & \boldsymbol{x} \in [\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}], \quad z \in [-f^-(\bar{\boldsymbol{x}}), -f^-(\underline{\boldsymbol{x}})], \quad t \in [-g^-(\bar{\boldsymbol{x}}), g^-(\underline{\boldsymbol{x}})] \end{cases}$$

Finally, due to [128, Prop. 11.1], we can replace the inequalities $f^-(\boldsymbol{x}) + z \leq 0$ and $g^+(\boldsymbol{x}) + t$ by the single inequality $\max\{f^-(\boldsymbol{x}) + z,\, g^+(\boldsymbol{x}) + t\} \leq 0$ and obtain a monotonic optimization problem in canonical form (4.4).

4.2 Algorithms

Tuy introduced monotonic optimization theory in his seminal paper [125] along with the polyblock outer approximation algorithm (cf. Section 3.3) to solve the canonical problem (4.4). Apart from the issues discussed in Section 3.3, the major downside is that up to two auxiliary variables are required to transform a general monotonic optimization problem (4.1) into the canonical form. Because the polyblock algorithm has exponential complexity in the number of variables, the increase in computational complexity due to extra variables is significant. Subsequently, Tuy introduced a BB algorithm to solve (4.1) directly in [130, §7], and a SIT approach in [126, 127, 130]. Both will be discussed in the following subsections. However, despite its drawbacks, the polyblock algorithm is the most widely employed algorithm to solve monotonic optimization problems.

4.2.1 Branch-and-Bound

Recall from Section 3.1 that the implementation of Algorithm 1 requires, for each box $\mathcal{M} \subseteq \mathcal{M}_0$ encountered during the algorithm, a bound $\beta(\mathcal{M}) \geq$

$\sup_{\boldsymbol{x}\in\mathcal{M}\cap\mathcal{D}} f(\boldsymbol{x})$, a feasibility check $\mathcal{M}\cap\mathcal{D} \stackrel{?}{=} \emptyset$, and a method to obtain some feasible point $\boldsymbol{x}' \in \mathcal{M}\cap\mathcal{D}$.

First, consider a box $\mathcal{M} = [\boldsymbol{r}, \boldsymbol{s}]$ and the objective of (4.1). Since $f^+(\boldsymbol{x})$ and $f^-(\boldsymbol{x})$ are both increasing functions, a bound over $\mathcal{M}$ is easily computed as

$$\begin{aligned}\sup_{\boldsymbol{x}\in\mathcal{M}\cap\mathcal{D}} \left(f^+(\boldsymbol{x}) - f^-(\boldsymbol{x})\right) &\leq \sup_{\boldsymbol{x}\in\mathcal{M}} f^+(\boldsymbol{x}) - \inf_{\boldsymbol{x}\in\mathcal{M}} f^-(\boldsymbol{x}) \\ &\leq f^+(\boldsymbol{s}) - f^-(\boldsymbol{r}) = \beta(\mathcal{M}). \qquad (4.6)\end{aligned}$$

For the feasibility problem, we start with the case where either $g_i^-(\boldsymbol{x}) = 0$ for all i or $g_i^+(\boldsymbol{x}) = 0$ for all i. In the first case, we have the following result.

4.4 Lemma. Let $\mathcal{D} = \{\boldsymbol{x} \,|\, g_i^+(\boldsymbol{x}) \leq 0, i = 1, \dots, n\}$ with increasing functions $g_i^+(\boldsymbol{x})$, and $\mathcal{M} = [\boldsymbol{r}, \boldsymbol{s}]$. Then, $\mathcal{D}\cap\mathcal{M} \neq \emptyset$ if and only if $g_i^+(\boldsymbol{r}) \leq 0$ for all $i = 1, \dots, n$. ◁

Proof. We prove the negation, i.e., $\mathcal{D}\cap\mathcal{M} = \emptyset \Leftrightarrow \exists i : g_i^+(\boldsymbol{r}) > 0$. First,

$$\mathcal{D}\cap\mathcal{M} = \emptyset \Rightarrow \forall \boldsymbol{x} \in \mathcal{M} : \exists i : g_i^+(\boldsymbol{x}) > 0 \Rightarrow \exists i : g_i^+(\boldsymbol{r}) > 0.$$

Conversely,

$$\begin{aligned}\exists i : g_i^+(\boldsymbol{r}) > 0 &\Rightarrow \exists i : \forall \boldsymbol{x} \geq \boldsymbol{r} : g_i^+(\boldsymbol{x}) \geq g_i^+(\boldsymbol{r}) > 0 \\ &\Rightarrow \{\boldsymbol{x} \in [\boldsymbol{r}, \boldsymbol{s}] \,|\, g_i^+(\boldsymbol{x}) \leq 0 \text{ for all } i\} = \emptyset.\end{aligned}$$ □

Similarly, for the second case we have

4.5 Lemma. Let $\mathcal{D} = \{\boldsymbol{x} \,|\, g_i^-(\boldsymbol{x}) \geq 0, i = 1, \dots, n\}$ with increasing functions $g_i^-(\boldsymbol{x})$, and $\mathcal{M} = [\boldsymbol{r}, \boldsymbol{s}]$. Then, $\mathcal{D}\cap\mathcal{M} \neq \emptyset$ if and only if $g_i^-(\boldsymbol{s}) \geq 0$ for all $i = 1, \dots, n$. ◁

The proof is equivalent to the previous one and therefore omitted.

For these two special cases, we obtain a convergent BB algorithm from Algorithm 1 with bound as in (4.6) and feasibility test as in Lemmas 4.4 and 4.5. If $\mathcal{D}\cap\mathcal{M} \neq \emptyset$, a feasible point is $\boldsymbol{r}$ if $g_i^-(\boldsymbol{x}) = 0$ for all i or $\boldsymbol{s}$ if $g_i^+(\boldsymbol{x}) = 0$ for all i. Convergence is established with the help of Proposition 3.7.

4.6 Proposition. Algorithm 1 solves (4.1) with η-optimality if either $g_i^-(\boldsymbol{x}) = 0$ for all i or $\boldsymbol{s}$ if $g_i^+(\boldsymbol{x}) = 0$ for all i, the selection is bound-improving, the bounding operation and feasibility test are as described as above, and exhaustive rectangular subdivision is employed. ◁

Proof. The feasibility test in Lemmas 4.4 and 4.5 is conclusive for every box $\mathcal{M} \subseteq \mathcal{M}_0$ and, thus, satisfies the requirements of Algorithm 1. Moreover, a feasible point can be found for every box such that $\mathcal{M} \cap \mathcal{D} \neq \emptyset$. By virtue of Proposition 3.7, Algorithm 1 is convergent if the subdivision is exhaustive and the bounding operation satisfies

$$\beta(\mathcal{M}) - \max\{f(\boldsymbol{x}) \,|\, \boldsymbol{x} \in \mathcal{M} \cap \mathcal{D}\} \to 0 \tag{4.7}$$

as $\operatorname{diam}(\mathcal{M}) \to 0$. The exhaustiveness of simple rectangular subdivision (i.e., bisection) is established in Corollary 3.9.

Finally, let $\bar{\boldsymbol{x}} \in \mathcal{M}$ be a maximizer of $f(\boldsymbol{x})$ over $\mathcal{M} \cap \mathcal{D}$. Clearly, as $\operatorname{diam}(\mathcal{M}) \to 0$, $\boldsymbol{r}, \boldsymbol{s} \to \bar{\boldsymbol{x}}$. Thus,

$$\beta(\mathcal{M}) = f^+(\boldsymbol{s}) - f^-(\boldsymbol{r}) \to f^+(\bar{\boldsymbol{x}}) - f^-(\bar{\boldsymbol{x}}) = f(\bar{\boldsymbol{x}})$$

and condition (4.7) is satisfied. □

If neither $g_i^-(\boldsymbol{x}) = 0$ nor $g_i^+(\boldsymbol{x}) = 0$ for all i, feasibility of a box $\mathcal{M}$ is not always easily determined. Consider the following lemma, which is part of [130, Prop. 7.3].

4.7 Lemma. Let $\mathcal{M} = [\boldsymbol{r}, \boldsymbol{s}]$ and $\mathcal{D}$ be the feasible set of (4.1), i.e., $\mathcal{D} = \{\boldsymbol{x} \,|\, g_i^+(\boldsymbol{x}) - g_i^-(\boldsymbol{x}) \leq 0,\ i = 1, \ldots, n\}$ with increasing functions g_i^+, g_i^-. Then, $\mathcal{D} \cap \mathcal{M} = \emptyset$ if $g_i^+(\boldsymbol{r}) - g_i^+(\boldsymbol{s}) > 0$ for some $i = 1, \ldots, n$. ◁

Proof. First, observe that, due to the monotonicity of g_i^+ and g_i^-,

$$\begin{aligned} g_i^+(\boldsymbol{r}) - g_i^+(\boldsymbol{s}) &\leq g_i^+(\boldsymbol{x}) - g_i^+(\boldsymbol{s}) && \text{for all } \boldsymbol{x} \geq \boldsymbol{r} \\ &\leq g_i^+(\boldsymbol{x}) - g_i^+(\boldsymbol{x}) && \text{for all } \boldsymbol{r} \leq \boldsymbol{x} \leq \boldsymbol{s}. \end{aligned}$$

Thus,

$$\begin{aligned} &\exists i : g_i^+(\boldsymbol{r}) - g_i^-(\boldsymbol{s}) > 0 \Rightarrow \exists i : \forall \boldsymbol{x} \in \mathcal{M} : g_i^+(\boldsymbol{x}) - g_i^-(\boldsymbol{x}) > 0 \\ &\qquad \Rightarrow \mathcal{M} \cap \mathcal{D} = \{\boldsymbol{x} \in \mathcal{M} \,|\, g_i^+(\boldsymbol{x}) - g_i^-(\boldsymbol{x}) \leq 0, i = 1, \ldots, n\} = \emptyset. \end{aligned}$$ □

Lemma 4.7 provides a sufficient condition for the infeasibility of a box $\mathcal{M}$, i.e., that $\mathcal{M} \cap \mathcal{D} = \emptyset$. Instead, Algorithm 1 requires a necessary and sufficient condition. Of course, we can modify Algorithm 1 as outlined in Section 3.2 to obtain an infinite algorithm, i.e., in Step 5, only perform an update of the incumbent if new feasible points are available and, in Step 6, only prune boxes that can be identified as infeasible.

4.8 Proposition. If Algorithm 1 is modified as described above, the selection is bound-improving, the bounding operation is as in (4.6), the modified feasibility test is implemented with Lemma 4.7, and exhaustive rectangular subdivision is employed, then the algorithm is infinite and, whenever it creates an infinite sequence $\{\boldsymbol{r}^k\}$, every accumulation point of this sequence is a globally optimal solution of (4.1). ◁

Proof sketch. Proposition 4.6 establishes the applicability of Proposition 3.7. From the proofs of Propositions 3.3 and 3.7 it follows that there exists an infinite decreasing sequence of sets $\{\mathcal{M}_{k_q}\}_q$ such that $g_i^+(\boldsymbol{r}^{k_q}) - g_i^+(\boldsymbol{s}^{k_q}) \leq 0$ for all i and $q = 1, 2, \ldots$. Since diam $\mathcal{M}_{k_q} \to 0$, $\boldsymbol{r}^{k_q}$ and $\boldsymbol{s}^{k_q}$ approach a common limit point $\boldsymbol{x}$. Due to continuity of g_i^+, g_i^-, this point satisfies $g_i^+(\boldsymbol{x}) - g_i^+(\boldsymbol{x}) \leq 0$ for all i. Altering Proposition 3.3's proof accordingly, it can be shown that the above proposition holds.[20] □

4.9 Remark. A similar result to Proposition 4.8 is established in [130, Thm. 7.1]. That algorithm incorporates the feasibility check into a reduction procedure which is intended to speed up the convergence. This optional reduction step is omitted here for brevity. Please refer to [130, §7.4] or to Section 6.4 for a more general procedure. ◁

4.10 Remark. While Proposition 4.8 is a significantly weaker statement than Proposition 4.6, this modified algorithm works surprisingly well for many problems despite its theoretically infinite convergence. ◁

4.2.2 Successive Incumbent Transcending

Following the discussion in Section 3.2, there are two methods to solve (4.1) with finite convergence. The first is to relax the constraints by some small $\varepsilon > 0$ and use the modified algorithm discussed above Proposition 4.8 to obtain an (ε, η)-approximate optimal solution. However, this very popular approach might result in a solution quite far away from the true optimal value and should, therefore, be avoided (cf. Section 3.2.1).

The other option is to employ the SIT approach from Section 3.2.2. This was first done for monotonic optimization in [126] and subsequently refined in [127, 128, 130]. Except for [127], all of these algorithms are designed for the canonical

[20] Please refer to Theorem 6.11 for a complete proof of a more general algorithm.

monotonic optimization Problem (4.4) and require up to two auxiliary variables to solve the general form (4.1). Instead, [127] solves

$$\max_{\boldsymbol{x}\in[\underline{\boldsymbol{x}},\bar{\boldsymbol{x}}]} \ f(\boldsymbol{x}) \quad \text{s.t.} \quad g^+(\boldsymbol{x}) - g^-(\boldsymbol{x}) \leq 0$$

where f, g^+, g^- are increasing functions of $\boldsymbol{x}$. This approach requires at most one auxiliary variable if the objective is DI.[21] However, more than one constraint requires the transformation given below (4.5) in Section 4.1. This might negatively affect the convergence speed of the algorithm since it results in increasingly loose bounds as the number of constraints grows.

In Chapter 5, we develop a SIT algorithm that works directly with more than one DI constraint.

4.3 Fractional Monotonic Programming

Consider the fractional program

$$\max_{\boldsymbol{x}\in[\underline{\boldsymbol{x}},\bar{\boldsymbol{x}}]} \frac{p^+(\boldsymbol{x}) - p^-(\boldsymbol{x})}{q^+(\boldsymbol{x}) - q^-(\boldsymbol{x})} \quad \text{s.t.} \quad g_i^+(\boldsymbol{x}) - g_i^-(\boldsymbol{x}) \leq 0, \ \ i = 1, \dots, n, \tag{4.8}$$

where $p^+, p^-, q^+, q^-, g_i^+, g_i^-, i = 1, \dots, n$, are all increasing functions. We have shown in Example 4.3 that, even though the objective might be a DI function,[22] the DI representation required to transform (4.8) into a canonical monotonic optimization problem can be very hard to find. Instead, the state-of-the-art approach to solve (4.8) is to combine Dinkelbach's algorithm (cf. Section 3.4) with monotonic optimization. This technique was, to the best of the author's knowledge, first used in [83] and subsequently stated as a framework in [147, 149].

Recall from Section 3.4 that Dinkelbach's algorithm (cf. Algorithm 4) solves a fractional program as a sequence of auxiliary optimization problems that need to be solved with global optimality. For (4.8) and some $\lambda \in \mathbb{R}$, this auxiliary problem is

$$\begin{cases} \max\limits_{\boldsymbol{x}\in[\underline{\boldsymbol{x}},\bar{\boldsymbol{x}}]} & p^+(\boldsymbol{x}) - p^-(\boldsymbol{x}) - \lambda(q^+(\boldsymbol{x}) - q^-(\boldsymbol{x})) \\ \text{s.t.} & g_i^+(\boldsymbol{x}) - g_i^-(\boldsymbol{x}) \leq 0, \ \ i = 1, \dots, n. \end{cases} \tag{4.9}$$

[21] If f is decreasing the algorithm in [127] works with minor modifications.

[22] Recall that a sufficient condition for the objective to be DI is that it is continuously differentiable.

By identifying

$$f^+(\boldsymbol{x}) = \begin{cases} p^+(\boldsymbol{x}) + \lambda q^-(\boldsymbol{x}) & \text{if } \lambda \geq 0 \\ p^+(\boldsymbol{x}) - \lambda q^+(\boldsymbol{x}) & \text{otherwise} \end{cases}$$

and

$$f^-(\boldsymbol{x}) = \begin{cases} p^-(\boldsymbol{x}) + \lambda q^+(\boldsymbol{x}) & \text{if } \lambda \geq 0 \\ p^-(\boldsymbol{x}) - \lambda q^-(\boldsymbol{x}) & \text{otherwise} \end{cases}$$

it is apparent that this is an instance of the monotonic optimization Problem (4.1). Thus, we can solve it with global optimality as described in Section 4.2.

The main drawback of this approach is that the inner monotonic problem is solved several times until convergence of the outer algorithm. This dramatically increases the computational complexity. Another shortcoming is the termination criterion. Recall that Dinkelbach's algorithm iteratively solves (4.9) and updates λ until the optimal value of (4.9) is close to zero. If it is exactly zero, the final $\boldsymbol{x}^k$ in Algorithm 4 optimally solves (4.8). Instead, if Algorithm 4 is terminated because $v(4.9) \leq \delta$, where $v(4.9)$ is the optimal value of (4.9) and δ is the tolerance in Algorithm 4, the final $\boldsymbol{x}^k$ only solves (4.8) approximately. The quality of this approximation is expected to improve as δ decreases but there is no strict mathematical relation between δ and the distance of the obtained solution to the true optimum. Thus, Dinkelbach's algorithm cannot guarantee an η-optimal solution.

In Chapters 5 and 6, two novel frameworks are proposed which are not limited in this way. In particular, both approaches obtain an η-optimal solution of (4.8).

4.4 Primal Decomposition for Resource Allocation over Rate Regions

Consider the general GIC resource allocation problem in (2.7) with WSR objective, i.e.,

$$\begin{cases} \max\limits_{\boldsymbol{p},\boldsymbol{R}} & \sum\limits_k w_k R_k \\ \text{s. t.} & \boldsymbol{a}_i^T \boldsymbol{R} \leq \log\left(1 + \dfrac{\boldsymbol{b}_i^T \boldsymbol{p}}{\boldsymbol{c}_i^T \boldsymbol{p} + \sigma_i^2}\right), \quad i = 1, \dots, n \\ & \boldsymbol{R} \geq \boldsymbol{0}, \quad \boldsymbol{R} \in \mathcal{Q}, \quad \boldsymbol{p} \in [\boldsymbol{0}, \bar{\boldsymbol{P}}] \cap \mathcal{P}, \end{cases} \tag{4.10}$$

where we assume $\mathcal{Q}$ linear and $\mathcal{P}$ normal. If the matrix $[\boldsymbol{a}_1, \boldsymbol{a}_2, \dots, \boldsymbol{a}_n]$ is equal to the identity matrix, the achievable rate region is a hypercube, the rates are easily eliminated from (4.10), and (4.10) is equivalent to (4.2) in Example 4.1. Instead, if the rate region is more involved, the rates are usually not eliminated easily from (4.10) and we have to solve it without prior simplification.

With (4.3), Problem (4.10) is identified as a general monotonic optimization problem. Since the objective is an increasing function, the transformation into canonical form requires a single auxiliary variable for the constraints. Thus, (4.10) is a monotonic optimization problem with $\dim \boldsymbol{p} + \dim \boldsymbol{R} + 1$ variables, where $\dim \boldsymbol{p}$ is the dimension of vector $\boldsymbol{p}$, i.e., the number of elements in $\boldsymbol{p}$. However, observe that (4.10) is a linear program for fixed $\boldsymbol{p}$. It is a well established fact that every linear program can be solved with polynomial computational complexity in the number of variables [62, 64]. Instead, all algorithms that solve (4.10) as a monotonic optimization problem have exponential computational complexity in the number of variables. Thus, treating the rates $\boldsymbol{R}$ as global variables leads to an unnecessary increase in computational complexity.

We can exploit the structural properties in $\boldsymbol{R}$ by transforming (4.10) into an outer monotonic optimization problem

$$\begin{cases} \max\limits_{t,\boldsymbol{p}} & \rho(t,\boldsymbol{p}) \\ \text{s.t.} & t + g_\Sigma(\boldsymbol{p}) \leq g_\Sigma(\bar{\boldsymbol{P}}) \\ & 0 \leq t \leq g_\Sigma(\bar{\boldsymbol{P}}) - g_\Sigma(\boldsymbol{0}) \\ & \boldsymbol{p} \in [\boldsymbol{0}, \bar{\boldsymbol{P}}] \cap \mathcal{P} \end{cases} \tag{4.11}$$

where $g_\Sigma(\boldsymbol{p}) = \sum_{i=1}^n \log(\boldsymbol{c}_i^T \boldsymbol{p} + \sigma_i^2)$ and $\rho(t,\boldsymbol{p})$ is either $-\infty$ or the solution to the linear program

$$\begin{cases} \max\limits_{\boldsymbol{R}} & \sum\limits_k w_k R_k & \text{(4.12a)} \\ \text{s.t.} & \boldsymbol{a}_i^T \boldsymbol{R} < f_i(\boldsymbol{p}) + \sum\limits_{j\neq i} g_j(\boldsymbol{p}) + t - g_\Sigma(\bar{\boldsymbol{P}}), \quad i = 1, \dots, n & \text{(4.12b)} \\ & \boldsymbol{R} \geq \boldsymbol{0}, \quad \boldsymbol{R} \in \mathcal{Q} & \text{(4.12c)} \end{cases}$$

if it is feasible, where $f_i(\boldsymbol{p}) = \log((\boldsymbol{b}_i + \boldsymbol{c}_i)^T \boldsymbol{p} + \sigma_i^2)$ and $g_i(\boldsymbol{p}) = \log(\boldsymbol{c}_j^T \boldsymbol{p} + \sigma_j^2)$. We first establish that (4.11) is indeed a monotonic optimization problem and then formally prove the equivalence of (4.10) and (4.11).

4.11 Proposition. (4.11) is a monotonic optimization problem and has at least one optimal solution. $\triangleleft$

Proof. Let $\mathcal{D}(\boldsymbol{p}, t)$ be the constraint set of (4.12) for fixed $(\boldsymbol{p}, t)$. Since the RHS of (4.12b) is an increasing function, $\mathcal{D}(\boldsymbol{p}, t) \subseteq \mathcal{D}(\boldsymbol{p}', t')$ whenever $0 \leq (\boldsymbol{p}, t) \leq (\boldsymbol{p}', t')$. This implies $\max\{\sum_k w_k R_k : \boldsymbol{R} \in \mathcal{D}(\boldsymbol{p}, t)\} \leq \max\{\sum_k w_k R_k : \boldsymbol{R} \in \mathcal{D}(\boldsymbol{p}', t')\}$ since $w_k > 0$. Thus, $\rho(\boldsymbol{p}, t)$ is an increasing function. Since $g_\Sigma(\boldsymbol{p})$ is increasing and continuous for $\boldsymbol{p} \geq \boldsymbol{0}$ and $g_\Sigma(\boldsymbol{p})$ is constant, the constraint set of (4.11) is closed and normal [125, Prop. 5]. This proves that (4.11) is a monotonic optimization problem.

Moreover, the feasible set of (4.11) is nonempty, $g_\Sigma(\boldsymbol{p})$ is a continuous function for $\boldsymbol{p} \geq \boldsymbol{0}$ and $\rho(t, \boldsymbol{p})$ is upper semi-continuous (u.s.c.) on the relevant interval. Thus, (4.11) has at least one optimal solution [128, Prop. 11.11]. $\square$

4.12 Proposition. Let $(\boldsymbol{R}^*, \boldsymbol{p}^*, t^*)$ be a solution of (4.11). Then, $(\boldsymbol{R}^*, \boldsymbol{p}^*)$ solves (4.10). $\triangleleft$

Proof. First, consider the optimization problem

$$\begin{cases} \max\limits_{t, \boldsymbol{R}, \boldsymbol{p}} & \sum\limits_k w_k R_k & (4.13a) \\ \text{s.t.} & \boldsymbol{a}_i^T \boldsymbol{R} < f_i(\boldsymbol{p}) + \sum\limits_{j \neq i} g_j(\boldsymbol{p}) + t - g_\Sigma(\bar{\boldsymbol{P}}), \quad i = 1, \dots, n & (4.13b) \\ & t + g_\Sigma(\boldsymbol{p}) \leq g_\Sigma(\bar{\boldsymbol{P}}) & (4.13c) \\ & 0 \leq t \leq g_\Sigma(\boldsymbol{p}) - g_\Sigma(\boldsymbol{0}) & (4.13d) \\ & \boldsymbol{R} \geq \boldsymbol{0}, \quad \boldsymbol{R} \in \mathcal{Q}, \quad \boldsymbol{p} \in [\boldsymbol{0}, \bar{\boldsymbol{P}}] \cap \mathcal{P}. & (4.13e) \end{cases}$$

We know from Proposition 4.11 that (4.11) has a solution. Hence, $\rho(t^*, \boldsymbol{p}^*)$ takes the value $\sum_k w_k R_k^*$ for every optimal solution $(\boldsymbol{R}^*, \boldsymbol{p}^*, t^*)$ satisfying the constraints in (4.12). Hence, (4.11) yields the maximum of $\sum_k w_k R_k^*$ under the constraints in (4.11) and (4.12). These are exactly the objective and constraints of (4.13). Thus, $(\boldsymbol{R}^*, \boldsymbol{S}^*, t^*)$ solves (4.13).

Next, observe that any optimal solution of (4.13) satisfies (4.13c) with equality because the RHS of (4.13b) is increasing in t, and (4.13d) is always inactive since g_Σ is an increasing function and $\boldsymbol{p} \geq \boldsymbol{0}$. Then, for (4.13b), we have

$$\boldsymbol{a}_i^T \boldsymbol{R}^* < f_i(\boldsymbol{p}^*) + \sum_{j \neq i} g_j(\boldsymbol{p}^*) + t^* - g_\Sigma(\bar{\boldsymbol{P}})$$

$$= f_i(\boldsymbol{p}^*) + \sum_{j \neq i} g_j(\boldsymbol{p}^*) + g_\Sigma(\bar{\boldsymbol{P}}) - g_\Sigma(\boldsymbol{p}^*) - g_\Sigma(\bar{\boldsymbol{P}})$$
$$= f_i(\boldsymbol{p}^*) + \sum_{j \neq i} g_j(\boldsymbol{p}^*) - g_\Sigma(\boldsymbol{p}^*).$$

Thus, constraints (4.13b)–(4.13d) are equal to the first constraint of (4.10) for any optimal solution of (4.13). Because (4.10) and (4.13) also have the same objective and all other constraints are the same, any solution to (4.13) is also a solution of (4.10). □

Proposition 4.12 establishes that (4.10) can be decomposed into (4.11) and (4.12). Proposition 4.11 proves that (4.11) is a monotonic optimization problem and the algorithms discussed in Section 4.2 are applicable. Numerical experiments prior to the publication of this approach in [77] have shown that the polyblock algorithm outperforms the BB algorithm in Section 4.2.1. The most likely reason for this is that the polyblock algorithm tends to evaluate the objective less often than a BB algorithm.

Apparently, this decomposition approach is highly problem specific and cumbersome. A significantly more versatile SIT based algorithm is developed in Chapter 5. It differentiates between convex and nonconvex variables and preserves the low computational complexity in the number of convex variables. Benchmark results in Section 8.4.1 show that it is up to four orders of magnitude faster than the decomposition approach presented here.

Chapter 5

SIT for Global Resource Allocation

In the previous chapter we have seen that optimization problems with DI constraints can be solved only in infinite time with BB, at least from a theoretical point of view. Moreover, the application of monotonic optimization theory to problems with several inequality constraints requires cumbersome transformations that often lead to slower convergence. The SIT scheme introduced in Section 3.2 is not subject to those limitations. Recall that the core idea is to exchange objective and constraints to obtain an optimization problem that is considerably easier to solve than the original problem. This requires some assumptions on the objective so that the dual feasible set is robust and has favorable properties that facilitate an easy implementation of the feasibility checks. Another benefit of this approach is the improved numerical stability due to the inherent exclusion of isolated feasible points. It also enables us to directly treat fractional objectives without Dinkelbach's algorithm.

Another key design objective for the algorithm in this chapter is to differentiate between convex and nonconvex variables without explicit primal decomposition as in Section 4.4. It is not necessary, however, to draw the line between convex and nonconvex variables. For example, consider the fractional programming problem (3.14) and assume that numerator and denominator are affine functions. In that case, (3.14) is a nonconvex optimization problem and all variables would be considered nonconvex, although it is solvable in polynomial time with Dinkelbach's algorithm or the Charnes-Cooper transformation. Hence, we introduce the notion of global and nonglobal variables. Consider (3.1), i.e., $\max_{\boldsymbol{x}\in\mathcal{D}} f(\boldsymbol{x})$, and split $\boldsymbol{x}$ into two vectors $\boldsymbol{y} = (x_i)_{i\in\mathcal{I}}$ and $\boldsymbol{z} = (x_i)_{i\in\{1,2,\dots,n\}\setminus\mathcal{I}}$. Modify (3.1) such that the optimization is only over $\boldsymbol{y}$ for some fixed $\boldsymbol{z}$, i.e., $\max_{\boldsymbol{y}\in\mathcal{D}_{\boldsymbol{z}}} f(\boldsymbol{y}, \boldsymbol{z})$ where $\mathcal{D}_{\boldsymbol{z}}$ is the $\boldsymbol{z}$-section of $\mathcal{D}$. If there exists an algorithm to solve this optimization problem with computational complexity significantly less than that required for solving the global part of (3.1),[23] the variables $\boldsymbol{y}$ and $\boldsymbol{z}$ are denoted as *nonglobal* and *global variables,* respectively.

[23]Ideally, the computational complexity for solving the nonglobal part of (3.1) grows polynomially in the number of nonglobal variables. However, there are optimization problems with exponential complexity that still have very efficient solution algorithms, e.g., certain kinds of convex optimization problems [7, §5.4], [88].

5.1 Problem Statement

Consider the global optimization problem

$$\begin{cases} \max\limits_{(\boldsymbol{x},\boldsymbol{\xi})\in\mathcal{C}} & \dfrac{f^+(\boldsymbol{x},\boldsymbol{\xi})}{f^-(\boldsymbol{x},\boldsymbol{\xi})} \\ \text{s.t.} & g_i^+(\boldsymbol{x},\boldsymbol{\xi}) - g_i^-(\boldsymbol{x}) \leq 0, \quad i = 1,2,\dots,m, \end{cases} \tag{5.1}$$

with global variables $\boldsymbol{x}$ and nonglobal variables $\boldsymbol{\xi}$. To enable the efficient solution of this problem with the SIT scheme and a BB algorithm, some technical assumptions on (5.1) are required. First, the functions $\{g_i^-\}$ must have a common maximizer over every box $[\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}] \subseteq \mathcal{M}_0$ with $\mathcal{M}_0$ being a box enclosing the $\boldsymbol{x}$ dimensions of $\mathcal{C}$, i.e., $\mathcal{M}^0 \supset \operatorname{proj}_{\boldsymbol{x}} \mathcal{C}$, where $\operatorname{proj}_{\boldsymbol{x}} \mathcal{C}$ is the projection of $\mathcal{C}$ onto the $\boldsymbol{x}$ coordinates. A sufficient condition for the existence of this common maximizer is that the functions $g_i^-(\boldsymbol{x})$ are increasing in the variables x_j, $j \in \mathcal{I}$, and decreasing in the remaining variables x_k, $k \in \{1,2,\dots,\dim \boldsymbol{x}\} \setminus \mathcal{I}$, for all i. This includes the case where all g_i^- are either increasing or decreasing in $\boldsymbol{x}$. Further, we assume the functions f^-, g_i^+, $i = 1,\dots,m$, to be lower semi-continuous (l.s.c.), the functions f^+, g_i^-, $i = 1,\dots,m$, to be u.s.c., and, WLOG, $f^-(\boldsymbol{x},\boldsymbol{\xi}) > 0$.

Further requirements can be grouped into two different sets. Both of these cases contain conditions that depend on a constant γ which will hold the current best known value in the developed algorithm. We will discuss the domain of γ after the definition of both cases.

5.1 Case (DC problems). If $\mathcal{C}$ is a closed convex set and $\gamma f^-(\boldsymbol{x},\boldsymbol{\xi}) - f^+(\boldsymbol{x},\boldsymbol{\xi})$, $g_1^+(\boldsymbol{x},\boldsymbol{\xi})$, ..., $g_m^+(\boldsymbol{x},\boldsymbol{\xi})$ are jointly convex in $(\boldsymbol{x},\boldsymbol{\xi})$ for all γ, Problem (5.1) resembles a DC optimization problem but with fractional objective and non-DC variables. ◁

5.2 Case (Separable problems). Let $\mathcal{C} = \mathcal{X} \times \Xi$ be such that $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{\xi} \in \Xi$ with Ξ being a closed convex set, and let each function of $(\boldsymbol{x},\boldsymbol{\xi})$ be separable in the sense that $h(\boldsymbol{x},\boldsymbol{\xi}) = h_x(\boldsymbol{x}) + h_\xi(\boldsymbol{\xi})$. Further, let the functions $\gamma f_\xi^-(\boldsymbol{\xi}) - f_\xi^+(\boldsymbol{\xi})$, $g_{1,\xi}^+(\boldsymbol{\xi})$, ..., $g_{m,\xi}^+(\boldsymbol{\xi})$ be convex in $\boldsymbol{\xi}$ for all γ, and let the functions $\gamma f_x^-(\boldsymbol{x}) - f_x^+(\boldsymbol{x})$, $g_{1,x}^+(\boldsymbol{x})$, ..., $g_{m,x}^+(\boldsymbol{x})$ have a common minimizer over $\mathcal{X} \cap \mathcal{M}$ for every box $\mathcal{M} \subseteq \mathcal{M}_0$ and all γ. Finally, let the function $\gamma f_x^-(\boldsymbol{x}) - f_x^+(\boldsymbol{x})$ be either increasing for all γ with $\mathcal{X}$ being a closed normal set in some box, or decreasing for all γ with $\mathcal{X}$ being a closed conormal set in some box. ◁

5.3 Remark. Separable problems often lead to linear auxiliary optimization problems with less variables instead of, typically, convex problems for Case 5.1. They usually have lower computational complexity than DC problems and, thus, if the problem at hand falls into both cases, it is usually favorable to consider it as a separable problem. ◁

With both cases defined we can take a closer look at γ. First, observe that γ only appears as a factor to f^-. Thus, its value is only relevant for the technical requirements above if (5.1) is a fractional program, i.e., if f^- is not constant. In that case, the only relevant property of γ is its sign and whether it may change during the algorithm. For example, in Case 5.1 the function $\gamma f_\xi^-(\boldsymbol{\xi}) - f_\xi^+(\boldsymbol{\xi})$ is convex if f_ξ^+ is concave and γf_ξ^- is convex. The latter is the case if $\gamma \geq 0$ and f_ξ^- is convex or if $\gamma \leq 0$ and f_ξ^- is concave. Thus, in most applications we should ensure that the sign of γ is constant. In general, γ may take values between some γ_0 and $v(5.1) + \eta$ for some small $\eta > 0$, where γ_0 is either the objective value of (5.1) for some preliminary known nonisolated feasible point $(\boldsymbol{x}, \boldsymbol{\xi})$ or an arbitrary value satisfying $\gamma_0 \leq \frac{f^+(\boldsymbol{x},\boldsymbol{\xi})}{f^-(\boldsymbol{x},\boldsymbol{\xi})}$ for all feasible $(\boldsymbol{x}, \boldsymbol{\xi})$. This implies, e.g., that γ is nonnegative if f^+ is nonnegative. Otherwise, it might be necessary to find a nonisolated feasible point such that $f^+(\boldsymbol{x}, \boldsymbol{\xi}) \geq 0$ or transform the problem.

Interchanging the objective and constraints in (5.1) leads to the dual problem

$$\begin{cases} \min\limits_{(\boldsymbol{x},\boldsymbol{\xi})\in\mathcal{C}} & \max\limits_{i=1,2,\dots,m} \left(g_i^+(\boldsymbol{x}, \boldsymbol{\xi}) - g_i^-(\boldsymbol{x})\right) \\ \text{s. t.} & \dfrac{f^+(\boldsymbol{x}, \boldsymbol{\xi})}{f^-(\boldsymbol{x}, \boldsymbol{\xi})} \geq \gamma \end{cases}$$

or, equivalently,

$$\begin{cases} \min\limits_{(\boldsymbol{x},\boldsymbol{\xi})\in\mathcal{C}} & \max\limits_{i=1,2,\dots,m} \left(g_i^+(\boldsymbol{x}, \boldsymbol{\xi}) - g_i^-(\boldsymbol{x})\right) \\ \text{s. t.} & \gamma f^-(\boldsymbol{x}, \boldsymbol{\xi}) - f^+(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0 \end{cases} \tag{5.2}$$

since $f^-(\boldsymbol{x}, \boldsymbol{\xi}) > 0$ by assumption. To ease notation, we define

$$g(\boldsymbol{x}, \boldsymbol{\xi}) = \max_{i=1,2,\dots,m} \left(g_i^+(\boldsymbol{x}, \boldsymbol{\xi}) - g_i^-(\boldsymbol{x})\right). \tag{5.3}$$

The feasible set $\mathcal{D}$ of (5.2) is robust and enables an efficient feasibility test as long as the conditions in Case 5.1 or 5.2 are satisfied. This is formally established in the following proposition.

5.4 Proposition. The feasible set of (5.2) does not contain any isolated points if the conditions in Case 5.1 or 5.2 are satisfied. ◁

Proof. First consider Case 5.1. The level set $\mathcal{F} = \{\gamma f^-(\boldsymbol{x}, \boldsymbol{\xi}) - f^+(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\}$ is closed and convex [100, Thms. 4.6 & 7.1]. Hence, $\mathcal{D} = \mathcal{C} \cap \mathcal{F}$ is also closed and convex. Every nonempty closed convex set is robust [100, Thm. 6.3]. Thus, $\mathcal{D}$ is robust or (5.2) is infeasible.

Now for Case 5.2. The feasible set of (5.2) is $\mathcal{D} = \{\boldsymbol{x} \in \mathcal{X}, \boldsymbol{\xi} \in \Xi : f_\xi(\boldsymbol{\xi}) + f_x(\boldsymbol{x}) \leq 0\}$ with $f_\xi(\boldsymbol{\xi}) = \gamma f_\xi^-(\boldsymbol{\xi}) - f_\xi^+(\boldsymbol{\xi})$ and $f_x(\boldsymbol{x}) = \gamma f_x^-(\boldsymbol{x}) - f_x^+(\boldsymbol{x})$. By assumption, $f_\xi(\boldsymbol{\xi})$ is an l.s.c. convex function and $f_x(\boldsymbol{x})$ an l.s.c. increasing (decreasing) function. Further, $\mathcal{X}$ is normal (conormal) within a box and Ξ is convex. Observe that $\mathcal{D}$ is a convex set in $\boldsymbol{\xi}$ since for fixed $\boldsymbol{x}$, $f_x(\boldsymbol{x})$ is a constant, $\tilde{f}_\xi(\boldsymbol{\xi}) = f_\xi(\boldsymbol{\xi}) + \text{const}$ is a convex function, and $\{\boldsymbol{\xi} : \tilde{f}_\xi(\boldsymbol{\xi}) \leq 0\}$ is a closed convex set [100, Thms. 4.6 & 7.1]. By the same argument, $\tilde{f}_x(\boldsymbol{x})$ is an increasing (decreasing) function and $\{\boldsymbol{x} : \tilde{f}_x(\boldsymbol{x}) \leq 0\}$ is a closed normal (conormal) set [128, Prop. 11.2]. Thus, $\mathcal{D}$ is normal (conormal) in a box in $\boldsymbol{x}$. Neither closed convex nor closed (co-)normal sets have any isolated feasible points [127]. Since $\mathcal{D}$ is either convex or (co-)normal in each coordinate the proposition is proven. □

Before we continue to solve (5.1), we shortly revisit some of the example resource allocation problems from Section 2.2 to discuss how they fit into (5.1) and Cases 5.1 and 5.2.

5.5 Example (Weighted Sum Rate). First consider the generic power allocation problem for GICs in (2.7) with objective $f(\boldsymbol{p}, \boldsymbol{R}) = \boldsymbol{w}^T \boldsymbol{R}$ and simple QoS constraints, i.e.,

$$\begin{cases} \max\limits_{\boldsymbol{p}, \boldsymbol{R}} & \boldsymbol{w}^T \boldsymbol{R} & (5.4a) \\ \text{s.t.} & \boldsymbol{a}_i^T \boldsymbol{R} \leq \log\left(1 + \dfrac{\boldsymbol{b}_i^T \boldsymbol{p}}{\boldsymbol{c}_i^T \boldsymbol{p} + \sigma_i^2}\right), \quad i = 1, \dots, n & (5.4b) \\ & \boldsymbol{R} \geq \boldsymbol{R}_{\min}, \quad \boldsymbol{p} \in [\boldsymbol{0}, \bar{\boldsymbol{P}}] & (5.4c) \end{cases}$$

for some $\boldsymbol{R}_{\min} \geq \boldsymbol{0}$. We have already identified $\boldsymbol{R}$ as linear variables (cf. Section 4.4) and, thus, we have $\boldsymbol{\xi} = \boldsymbol{R}$, $\boldsymbol{x} = \boldsymbol{p}$, $f^+(\boldsymbol{x}, \boldsymbol{\xi}) = \boldsymbol{w}^T \boldsymbol{\xi}$, $f^-(\boldsymbol{x}, \boldsymbol{\xi}) = 1$, and $\mathcal{C} = [\boldsymbol{0}, \boldsymbol{P}] \times [\boldsymbol{R}_{\min}, \infty)$. In (4.3), we have established that constraint (5.4b) is equivalent to

$$\boldsymbol{a}_i^T \boldsymbol{R} - \log\left((\boldsymbol{b}_i + \boldsymbol{c}_i)^T \boldsymbol{p} + \sigma_i^2\right) + \log\left(\boldsymbol{c}_i^T \boldsymbol{p} + \sigma_i^2\right) \leq 0$$

and we can identify

$$g_i^+(\boldsymbol{x}, \boldsymbol{\xi}) = \boldsymbol{a}_i^T \boldsymbol{\xi} - \log\big(\big(\boldsymbol{b}_i^T + \boldsymbol{c}_i^T\big)\, \boldsymbol{x} + \sigma_i^2\big) \tag{5.5a}$$

$$g_i^-(\boldsymbol{x}) = -\log\big(\boldsymbol{c}_i^T \boldsymbol{x} + \sigma_i^2\big)\,. \tag{5.5b}$$

Then, the functions g_i^- are decreasing, and, thus, are jointly maximized over the box $[\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}]$ by $\underline{\boldsymbol{x}}$. Further, $g_i^+(\boldsymbol{x}, \boldsymbol{\xi})$ is separable in $\boldsymbol{x}$ and $\boldsymbol{\xi}$ with $g_{i,\xi}^+(\boldsymbol{\xi})$ being linear and $g_{i,x}^+(\boldsymbol{x})$ decreasing and convex in $\boldsymbol{x}$. With $f(\boldsymbol{x}, \boldsymbol{\xi}) = f_\xi(\boldsymbol{\xi})$ being a linear function in $\boldsymbol{\xi}$, Problem (5.4) qualifies for both, Cases 5.1 and 5.2. ◁

5.6 Example (Global Energy Efficiency). Again, consider (2.7) but with GEE utility function $f(\boldsymbol{p}, \boldsymbol{R}) = \frac{\sum_k R_k}{\boldsymbol{\phi}^T \boldsymbol{p} + P_c}$. Thus, we identify $f^+(\boldsymbol{x}, \boldsymbol{\xi}) = \sum_k \xi_k$ and $f^-(\boldsymbol{x}, \boldsymbol{\xi}) = \boldsymbol{\phi}^T \boldsymbol{x} + P_c$, and observe that both are linear functions. Hence, $\gamma f^-(\boldsymbol{x}, \boldsymbol{\xi}) - f^+(\boldsymbol{x}, \boldsymbol{\xi})$ is convex for all γ and, with the remaining identifications as in the previous example, Case 5.1 is satisfied. Of course, f^+ and f^- are also separable, and because the GEE is nonnegative, $\gamma \geq 0$ and γf_x^- is increasing. But since $g_{i,x}^+$ is decreasing, the functions γf_x^-, $g_{1,x}^+$, …, $g_{m,x}^+$ do not have a common minimizer over $[\boldsymbol{0}, \bar{\boldsymbol{P}}] \cap \mathcal{M}$ and Case 5.2 does not apply. However, observe that instead of the choice in (5.5), we can also identify g_i^+ and g_i^- as

$$g_i^+(\boldsymbol{x}, \boldsymbol{\xi}) = \boldsymbol{a}_i^T \boldsymbol{\xi} + \log\big(\boldsymbol{c}_i^T \boldsymbol{x} + \sigma_i^2\big) \tag{5.6a}$$

$$g_i^-(\boldsymbol{x}) = \log\big(\big(\boldsymbol{b}_i^T + \boldsymbol{c}_i^T\big)\, \boldsymbol{x} + \sigma_i^2\big)\,. \tag{5.6b}$$

The functions $g_i^+(\boldsymbol{x}, \boldsymbol{\xi})$ are separable in $\boldsymbol{x}$ and $\boldsymbol{\xi}$, and $g_{i,x}^+(\boldsymbol{x})$ and $g_i^-(\boldsymbol{x})$ are increasing in $\boldsymbol{x}$. Hence, γf_x^-, $g_{1,x}^+$, …, $g_{m,x}^+$ have a common minimizer over $[\boldsymbol{0}, \bar{\boldsymbol{P}}] \cap \mathcal{M}$ for every box $\mathcal{M}$ and Case 5.2 applies. In general, (5.5) will result in tighter bounds and, thus, faster convergence. On the other hand, Case 5.2 has significantly lower numerical complexity than Case 5.1 (cf. Remark 5.3) which should compensate for this drawback. However, a final assessment is only possible through numerical experimentation which will be carried out for an example application in Section 8.4.2. ◁

Next, we design an adaptive BB procedure to solve (5.2) and then embed this algorithm within the SIT scheme to solve (5.1).

5.2 A Branch-and-Bound Procedure to Solve the Dual

We design a rectangular BB algorithm to solve (5.2). Recall from Section 3.1 that the core idea of BB is to relax the feasible set and subsequently partition it such that lower bounds[24] on the objective value can be determined easily. The technical requirements on (5.1) are chosen such that bounds are easily computed over hyperrectangles. In particular, this is enabled by the existence of a common maximizer of $\{g_i^-\}$ over every box.

5.7 Proposition. Let $\bar{\boldsymbol{x}}^*_{\mathcal{M}}$ be a common maximizer of $\{g_i^-\}$ over the box $\mathcal{M}$. Then, Problem (5.2)'s objective g is lower bounded over $\mathcal{M}$ by

$$\max_{i=1,2,\dots,m} \left\{g_i^+(\boldsymbol{x}, \boldsymbol{\xi}) - g_i^-(\bar{\boldsymbol{x}}^*_{\mathcal{M}})\right\}$$

This bound is tight at $\bar{\boldsymbol{x}}^*_{\mathcal{M}}$. ◁

Proof. For all $\boldsymbol{x} \in \mathcal{M}$, $g_i^-(\bar{\boldsymbol{x}}^*) \geq g_i^-(\boldsymbol{x})$ and hence also $g_i^+(\boldsymbol{x}, \boldsymbol{\xi}) - g_i^-(\bar{\boldsymbol{x}}^*) \leq g_i^+(\boldsymbol{x}, \boldsymbol{\xi}) - g_i^-(\boldsymbol{x})$ with equality at $\boldsymbol{x} = \bar{\boldsymbol{x}}^*$. It remains to be shown that for all real-valued functions $h_1, h_2, \dots, \underline{h}_1, \underline{h}_2, \dots$ satisfying $h_i \geq \underline{h}_i$ and $h_i(\boldsymbol{y}) = \underline{h}_i(\boldsymbol{y})$ for all i and some point $\boldsymbol{y}$, $\max_i\{h_i\} \geq \max_i\{\underline{h}_i\}$ and $\max_i\{h_i(\boldsymbol{y})\} = \max_i\{\underline{h}_i(\boldsymbol{y})\}$ holds.

Consider the case with two functions and assume that $\max\{h_1, h_2\} < \max\{\underline{h}_1, \underline{h}_2\}$. Since $h_i \geq \underline{h}_i$, this can only hold if $\max\{h_1, h_2\} = h_1$ and $\max\{\underline{h}_1, \underline{h}_2\} = \underline{h}_2$ or vice versa. This implies $h_1 \geq h_2 \geq \underline{h}_2$ which contradicts the assumption. The generalization to arbitrarily many functions follows by induction. Finally, if $\underline{h}_i(\boldsymbol{y}) = h_i(\boldsymbol{y})$ for all i, then $\min_i\{\underline{h}_i(\boldsymbol{y})\} = \min_i\{h_i(\boldsymbol{y})\}$.□

With Proposition 5.7 we can determine the lower bound $\beta(\mathcal{M}_i)$ in Algorithm 1 as the optimal value of

$$\begin{cases} \min\limits_{\boldsymbol{x}, \boldsymbol{\xi}} & \max\limits_{i=1,2,\dots,m} \left\{g_i^+(\boldsymbol{x}, \boldsymbol{\xi}) - g_i^-(\bar{\boldsymbol{x}}^*_{\mathcal{M}_i})\right\} \\ \text{s. t.} & \gamma f^-(\boldsymbol{x}, \boldsymbol{\xi}) - f^+(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0 \\ & (\boldsymbol{x}, \boldsymbol{\xi}) \in \mathcal{C}, \ \boldsymbol{x} \in \mathcal{M}_i. \end{cases} \tag{5.7}$$

This is a convex optimization problem if the conditions in Case 5.1 or 5.2 are satisfied and can be solved in polynomial time under very mild assumptions using standard tools [89]. For Case 5.1, the objective of (5.7) is a convex function because $g_i^-(\bar{\boldsymbol{x}}^*_{\mathcal{M}_i})$ is constant and the feasible set is convex (cf. Proposition 5.4).

[24] The prototype BB algorithm in Section 3.1 is designed to solve the maximization Problem (3.1) and requires an upper bound on the objective. Instead, (5.2) is a minimization problem and, thus, requires a lower bound on the objective.

Thus, (5.7) is convex given Case 5.1. For Case 5.2, (5.7) can be written as

$$\begin{cases} \min\limits_{\boldsymbol{x},\boldsymbol{\xi}} & \max\limits_{i=1,2,\dots,m} \left\{ g_{i,\xi}^{+}(\boldsymbol{\xi}) + g_{i,x}^{+}(\boldsymbol{x}) - g_i^{-}(\bar{\boldsymbol{x}}_{\mathcal{M}_i}^{*}) \right\} \\ \text{s. t.} & \gamma f_{\xi}^{-}(\boldsymbol{\xi}) - f_{\xi}^{+}(\boldsymbol{\xi}) + \gamma f_{x}^{-}(\boldsymbol{x}) - f_{x}^{+}(\boldsymbol{x}) \le 0 \\ & \boldsymbol{\xi} \in \Xi, \ \boldsymbol{x} \in \mathcal{X} \cap \mathcal{M}_i. \end{cases} \tag{5.8}$$

Let $\underline{\boldsymbol{x}}_{\mathcal{M}_i}^{*}$ be the common minimizer of $\gamma f_{x}^{-}(\boldsymbol{x}) - f_{x}^{+}(\boldsymbol{x})$, $g_{1,x}^{+}(\boldsymbol{x})$, ..., $g_{m,x}^{+}(\boldsymbol{x})$ over $\mathcal{X} \cap \mathcal{M}_i$. Then (5.8) is equivalent to

$$\begin{cases} \min\limits_{\boldsymbol{\xi}\in\Xi} & \max\limits_{i=1,2,\dots,m} \left\{ g_{i,\xi}^{+}(\boldsymbol{\xi}) + g_{i,x}^{+}(\underline{\boldsymbol{x}}_{\mathcal{M}_i}^{*}) - g_i^{-}(\bar{\boldsymbol{x}}_{\mathcal{M}_i}^{*}) \right\} \\ \text{s. t.} & \gamma f_{\xi}^{-}(\boldsymbol{\xi}) - f_{\xi}^{+}(\boldsymbol{\xi}) + \gamma f_{x}^{-}(\underline{\boldsymbol{x}}_{\mathcal{M}_i}^{*}) - f_{x}^{+}(\underline{\boldsymbol{x}}_{\mathcal{M}_i}^{*}) \le 0. \end{cases} \tag{5.9}$$

It is easy to see that $\underline{\boldsymbol{x}}_{\mathcal{M}_i}^{*}$ is the optimal solution of (5.8) since it jointly minimizes the objective and the first constraint. Problem (5.9) is convex due to the assumptions on Ξ and the remaining functions of $\boldsymbol{\xi}$ made in Case 5.2.

This bounding procedure generates two points, namely $\boldsymbol{v}^k = \bar{\boldsymbol{x}}_{\mathcal{M}^k}^{*}$ and the optimal solution $(\boldsymbol{x}^k, \boldsymbol{\xi}^k)$ of (5.7), that satisfy the convergence criterion (3.9) for the adaptive rectangular subdivision procedure in Section 3.1.1.

5.8 Lemma. Algorithm 1[25] with bound (5.7) (or (5.9)), adaptive bisection via $(\boldsymbol{x}^k, \boldsymbol{v}^k)$, feasible value $\alpha(\mathcal{M}^k) = \min_{\boldsymbol{\xi}\in\mathcal{D}_{\boldsymbol{v}^k}} g(\boldsymbol{v}^k, \boldsymbol{\xi})$, and bound-improving selection is convergent and the final $\bar{\boldsymbol{x}}^k$ is a global η-optimal solution of (5.2).◁

Proof. By virtue of Proposition 3.11, this lemma holds if the bound satisfies the convergence criterion in (3.9). Specified to the problem at hand, the bounding procedure needs to generate two points $\boldsymbol{x}^k, \boldsymbol{v}^k \in \mathcal{M}^k$ satisfying

$$(\boldsymbol{x}^k, \boldsymbol{\xi}^k) \in \mathcal{D}, \quad \beta(\mathcal{M}^k) - \min_{\boldsymbol{\xi}\in\mathcal{D}_{\boldsymbol{v}^k}} g(\boldsymbol{v}^k, \boldsymbol{\xi}) \to 0 \text{ as } \left\| \boldsymbol{x}^k - \boldsymbol{v}^k \right\| \to 0$$

for some $\boldsymbol{\xi}^k$. Neither (5.7) nor (5.9) relax the feasible set $\mathcal{D}$ of (5.2). Hence, the first condition is always satisfied. Further,

$$\beta(\mathcal{M}^k) = \min_{\boldsymbol{\xi}\in\mathcal{D}_{\boldsymbol{x}^k}} \max_{i=1,2,\dots,m} \left\{ g_i^{+}(\boldsymbol{x}^k, \boldsymbol{\xi}) - g_i^{-}(\bar{\boldsymbol{x}}_{\mathcal{M}_i}^{*}) \right\}$$

[25] With minor modifications such that the objective is minimized instead of maximized.

and

$$\min_{\boldsymbol{\xi}\in\mathcal{D}_{\boldsymbol{v}^k}} g(\boldsymbol{v}^k,\boldsymbol{\xi}) = \min_{\boldsymbol{\xi}\in\mathcal{D}_{\boldsymbol{v}^k}} \max_{i=1,2,\dots,m} \left(g_i^+(\bar{\boldsymbol{x}}^*_{\mathcal{M}^k},\boldsymbol{\xi}) - g_i^-(\bar{\boldsymbol{x}}^*_{\mathcal{M}^k})\right)$$

Thus, $\beta(\mathcal{M}^k) \to \min_{\boldsymbol{\xi}\in\mathcal{D}_{\boldsymbol{v}^k}} g(\boldsymbol{v}^k,\boldsymbol{\xi})$ as $\left\|\boldsymbol{x}^k - \boldsymbol{v}^k\right\| = \left\|\boldsymbol{x}^k - \bar{\boldsymbol{x}}^*_{\mathcal{M}^k}\right\| \to 0$. □

BB algorithms typically require a large amount of bound evaluations. This requires that (5.7) (or (5.9)) is solved as efficiently as possible. Moreover, the selection procedure should be chosen as the best-first rule in (3.3) because it results in fastest convergence.

5.3 The SIT Algorithm

The BB procedure from the previous section solves (5.2), but that is not exactly what is required by the SIT scheme. Instead, Algorithm 2 requires the implementation of

Step 1 Check if (5.1) has a nonisolated feasible solution $\boldsymbol{x}$ satisfying $f(\boldsymbol{x}) \geq \gamma$; otherwise, establish that no such ε-essential feasible $\boldsymbol{x}$ exists and go to Step 3.

This is accomplished by the adaptive BB algorithm for solving (5.2) from the previous section but with pruning criterion $\beta(M) > -\varepsilon$ and termination criterion $\min_{\boldsymbol{\xi}\in\mathcal{D}_{\boldsymbol{x}^k}} g(\boldsymbol{x}^k,\boldsymbol{\xi}) < 0$. The following proposition together with Lemma 5.8 establishes convergence of this procedure.

5.9 Proposition ([128, Prop. 7.14]). Let $\varepsilon > 0$ be given, $g(\boldsymbol{x},\boldsymbol{\xi})$ as in (5.3), and (5.2) be solved by the BB procedure from Section 5.2 with best-first selection rule. Either $g(\boldsymbol{x}^k,\boldsymbol{\xi}^*) < 0$ for some k and $\boldsymbol{\xi}^*$ or $\beta(\mathcal{M}^k) > -\varepsilon$ for some k. In the former case, $(\boldsymbol{x}^k,\boldsymbol{\xi}^*)$ is a nonisolated feasible solution of (5.1) satisfying $\frac{f^+(\boldsymbol{x}^k,\boldsymbol{\xi}^*)}{f^-(\boldsymbol{x}^k,\boldsymbol{\xi}^*)} \geq \gamma$. In the latter case, no ε-essential feasible solution $(\boldsymbol{x},\boldsymbol{\xi})$ of (5.1) exists such that $\frac{f^+(\boldsymbol{x},\boldsymbol{\xi})}{f^-(\boldsymbol{x},\boldsymbol{\xi})} \geq \gamma$. ◁

Proof. If $\min_{\boldsymbol{\xi}\in\mathcal{D}_{\boldsymbol{x}^k}} g(\boldsymbol{x}^k,\boldsymbol{\xi}) < 0$, then $\boldsymbol{x}^k$ is a feasible point of (5.1) with objective value $\frac{f^+(\boldsymbol{x}^k,\boldsymbol{\xi}^*)}{f^-(\boldsymbol{x}^k,\boldsymbol{\xi}^*)} \geq \gamma$. By continuity, $\min_{\boldsymbol{\xi}\in\mathcal{D}_{\boldsymbol{x}}} g(\boldsymbol{x}^k,\boldsymbol{\xi}) \leq 0$ for all $\boldsymbol{x}$ in a neighborhood of $\boldsymbol{x}^k$. Thus, $\boldsymbol{x}^k$ is an essential feasible point of (5.1).

Observe that

$$\mathcal{M}^k \in \operatorname{argmin}\{\beta(\mathcal{M}) \,|\, \mathcal{M} \in \mathscr{R}_k\} \Rightarrow \forall \mathcal{M} \in \mathscr{R}_k : \beta(\mathcal{M}) \geq \beta(\mathcal{M}^k) \quad (5.10)$$

and, thus, $\beta(\mathcal{M}^k) \leq v(5.2)$ where $v(5.2)$ is the optimal value of (5.2). Hence, if $\beta(\mathcal{M}^k) > -\varepsilon$, then also $v(5.2) > -\varepsilon$ and, by Proposition 3.15, no ε-essential feasible solution of (5.1) with objective value greater or equal than γ exists.

It remains to show that either $g(\boldsymbol{x}^k, \boldsymbol{\xi}^*) < 0$ for some k and $\boldsymbol{\xi}^*$ or $\beta(\mathcal{M}^k) > -\varepsilon$ for some k. Suppose the contrary, i.e.,

$$\forall k: \quad \min_{\boldsymbol{\xi} \in \mathcal{D}_{\boldsymbol{x}^k}} g(\boldsymbol{x}^k, \boldsymbol{\xi}) \geq 0, \;\; \beta(\mathcal{M}^k) \leq -\varepsilon. \tag{5.11}$$

By virtue of Proposition 3.11 and Lemmas 3.10 and 5.8, the sequence $\{\boldsymbol{x}^k, \boldsymbol{v}^k\}$ generated by the adaptive BB procedure satisfies, for some infinite subsequence $\{k_q\} \subseteq \{1, 2, \ldots\}$,

$$\lim_{q\to\infty} \left(\beta(\mathcal{M}^{k_q}) - \min_{\boldsymbol{\xi} \in \mathcal{D}_{\boldsymbol{v}^{k_q}}} g(\boldsymbol{v}^{k_q}, \boldsymbol{\xi}) \right) = 0 \tag{5.12}$$

$$\lim_{q\to\infty} \boldsymbol{x}^k = \lim_{q\to\infty} \boldsymbol{v}^k = \boldsymbol{x}'$$

with $\frac{f^+(\boldsymbol{x}', \boldsymbol{\xi})}{f^-(\boldsymbol{x}', \boldsymbol{\xi})} \geq \gamma$ for some $\boldsymbol{\xi}$. By (5.11),

$$\min_{\boldsymbol{\xi} \in \mathcal{D}_{\boldsymbol{x}'}} g(\boldsymbol{x}', \boldsymbol{\xi}) \geq 0. \tag{5.13}$$

On the other hand, (5.12) implies

$$\lim_{q\to\infty} \beta(\mathcal{M}^{k_q}) = \lim_{q\to\infty} \min_{\boldsymbol{\xi} \in \mathcal{D}_{\boldsymbol{v}^{k_q}}} g(\boldsymbol{v}^{k_q}, \boldsymbol{\xi}) = \min_{\boldsymbol{\xi} \in \mathcal{D}_{\boldsymbol{x}'}} g(\boldsymbol{x}', \boldsymbol{\xi}).$$

Thus, because $\beta(\mathcal{M}^k) \leq -\varepsilon$, for all k, by assumption (5.11),

$$\min_{\boldsymbol{\xi} \in \mathcal{D}_{\boldsymbol{x}'}} g(\boldsymbol{x}', \boldsymbol{\xi}) \leq -\varepsilon,$$

which is a contradiction of (5.13).[26] □

We can directly call this BB procedure from Step 1 of the SIT scheme in Algorithm 2 to solve (5.1). Of course, this means we have to solve (5.2) several times and the computational complexity of this approach would be prohibitively

[26] Proposition 5.9 is the same as [128, Prop. 7.14] but adapted to (5.1). The proof is a more verbose version of the original in [128] and contains verbatim copies of text passages from the original proof. It is repeated here because it is an important cornerstone of the developed algorithm.

high. Instead, we can integrate both procedures into a single algorithm. The outlined BB procedure is started with some initial γ_0 and interrupted whenever a primal feasible point is encountered, i.e., a point with objective value less than zero in the dual problem (5.2). Then, γ from Algorithm 2, which holds the current best objective value of the primal problem (5.1), is updated and the BB procedure is continued with the updated feasible set.

The complete algorithm is stated in Algorithm 5. It is initialized in Step 0 where an initial box $\mathcal{M}^0 = [\boldsymbol{p}^0, \boldsymbol{q}^0]$ is required that contains the $\boldsymbol{x}$-dimensions of $\mathcal{C}$, i.e.,

$$p_i^0 \leq \min_{(\boldsymbol{x},\boldsymbol{\xi})\in\mathcal{C}} x_i \qquad\qquad q_i^0 \geq \max_{(\boldsymbol{x},\boldsymbol{\xi})\in\mathcal{C}} x_i. \tag{5.14}$$

The set $\mathscr{P}_k$ contains new boxes to be examined in Step 1, γ holds the current best value adjusted by the tolerance η, and $\mathscr{R}_k$ holds all boxes that are not yet eliminated. In Step 1 the bound is computed for each box in $\mathscr{P}_k$. If it is less than $-\varepsilon$, the box may contain a nonisolated feasible solution with objective value greater than γ and is added to $\mathscr{R}_k$. Then, in Step 3, the box with the smallest bound is taken out of $\mathscr{R}_k$. If the point $\boldsymbol{x}^k$ attaining the bound is feasible in the original Problem (5.1), it is a nonisolated feasible point and needs to be examined further: if the objective value for $\boldsymbol{x}^k$ is greater than the current best value $\gamma - \eta$, $\boldsymbol{x}^k$ is the new current best solution and γ is updated accordingly in Step 4. Irrespective of $\boldsymbol{x}^k$'s feasibility, the box selected in Step 3 is bisected via $(j_k, \boldsymbol{v}^k)$ adaptively. These new boxes are then passed to Step 1 and the algorithm repeats until $\mathscr{R}_k$ holds no more boxes which is checked in Step 2. Convergence of the algorithm is established in the theorem below.

5.10 Theorem. Algorithm 5 converges in finitely many steps to the (ε, η)-optimal solution of (5.1) or establishes that no such solution exists. ◁

Proof. Convergence of the BB procedure is established in Lemma 5.8 and Proposition 5.9. It remains to show that continuing the BB procedure after updating γ preserves convergence.

Let $\{\gamma_k\}$ be the sequence of updated gammas, and observe that this sequence is increasing, i.e., $\gamma_{k+1} \geq \gamma_k$. Thus, the feasible sets of (5.7) form a decreasing sequence of sets

$$\left\{\boldsymbol{x} \in \tilde{\mathcal{C}} \,\middle|\, \frac{f^+(\boldsymbol{x},\boldsymbol{\xi})}{f^-(\boldsymbol{x},\boldsymbol{\xi})} \geq \gamma_{k+1}\right\} \subseteq \left\{\boldsymbol{x} \in \tilde{\mathcal{C}} \,\middle|\, \frac{f^+(\boldsymbol{x},\boldsymbol{\xi})}{f^-(\boldsymbol{x},\boldsymbol{\xi})} \geq \gamma_k\right\}$$

with $\tilde{\mathcal{C}} = \{(\boldsymbol{x},\boldsymbol{\xi}) \,|\, (\boldsymbol{x},\boldsymbol{\xi}) \in \mathcal{C},\, \boldsymbol{x} \in \mathcal{M}\}$. Therefore, the optimal value of the bound (5.7) is increasing with k

$$\begin{aligned}&\min_{\boldsymbol{x},\boldsymbol{\xi}} \left\{ \tilde{g}(\boldsymbol{x},\boldsymbol{\xi}) \;\middle|\; \frac{f^+(\boldsymbol{x},\boldsymbol{\xi})}{f^-(\boldsymbol{x},\boldsymbol{\xi})} \geq \gamma_{k+1},\, (\boldsymbol{x},\boldsymbol{\xi}) \in \tilde{\mathcal{C}} \right\} \\ \geq &\min_{\boldsymbol{x},\boldsymbol{\xi}} \left\{ \tilde{g}(\boldsymbol{x},\boldsymbol{\xi}) \;\middle|\; \frac{f^+(\boldsymbol{x},\boldsymbol{\xi})}{f^-(\boldsymbol{x},\boldsymbol{\xi})} \geq \gamma_{k},\, (\boldsymbol{x},\boldsymbol{\xi}) \in \tilde{\mathcal{C}} \right\}\end{aligned} \tag{5.15}$$

where $\tilde{g}(\boldsymbol{x},\boldsymbol{\xi}) = \max_{i=1,2,\ldots,m} \left\{ g_i^+(\boldsymbol{x},\boldsymbol{\xi}) - g_i^-(\bar{\boldsymbol{x}}^*_{\mathcal{M}_i}) \right\}$. Hence, every box eliminated due to the deletion criterion $\beta(\mathcal{M}) > -\varepsilon$ in a BB procedure with γ_k, would also be eliminated in a procedure with γ_{k+1}. It follows that the set holding the boxes $\mathscr{R}_k$ remains valid after updating γ. In particular, no box is eliminated prematurely. Thus, restarting the BB procedure after updating γ is not necessary. Convergence of Algorithm 5 is finite since the underlying BB procedure is finite and $\{\gamma_k\}$ is bounded. □

Observe that (5.16) is a convex feasibility problem due to the assumptions in Cases 5.1 and 5.2. It also has the same feasible set as (5.17). Because $\boldsymbol{\xi}$ are nonglobal variables, Problem (5.17) is solvable with reasonable computational complexity by assumption. For example, if $f^+(\boldsymbol{x},\boldsymbol{\xi})$ is nonnegative and concave in $\boldsymbol{\xi}$, and $f^-(\boldsymbol{x},\boldsymbol{\xi})$ is convex in $\boldsymbol{\xi}$ it can be solved with Dinkelbach's algorithm (cf. Section 3.4). Note that in many circumstances it will be faster to solve (5.17) directly instead of first considering the feasibility problem (5.16).

5.11 Remark. Proposition 5.9 suggests that "$\beta(\mathcal{M}^k) < -\varepsilon$" should be used as the termination criterion. Instead, it is employed as pruning rule in the outlined BB procedure and Algorithm 5. However, with best-first selection and "$\beta(\mathcal{M}^k) < -\varepsilon$" pruning rule, the termination criterion "$\beta(\mathcal{M}^k) < -\varepsilon$" is equivalent to the usual termination rule "$\mathscr{R}_k = \emptyset$" (cf. [128, p. 208]). Thus, the proposed BB procedure does not violate Proposition 5.9 but might have slower convergence than a procedure with this additional stopping rule.

However, when the bounding procedure is changed during the algorithm, as is done in Algorithm 5, the "$\beta(\mathcal{M}^k) < -\varepsilon$" termination criterion is impractical. This is because, in an implementation of Algorithm 5, the bound of each box is only calculated once and then stored together with the box in $\mathscr{R}_k$. Thus, after updating γ, the precomputed bounds in $\mathscr{R}_k$ are actually obtained from different bounding procedures and the argmin box selection in Step 3 is not identical to the theoretical argmin operation in (5.10). Since the inequality in (5.15) might

Algorithm 5 SIT Algorithm for (5.1)

Step 0 Initialize $\varepsilon, \eta > 0$ and $\mathcal{M}_0 = [\boldsymbol{p}^0, \boldsymbol{q}^0]$ as in (5.14), $\mathscr{P}_1 = \{\mathcal{M}_0\}$, $\mathscr{R}_1 = \emptyset$, and $k = 1$. Initialize $\bar{\boldsymbol{x}}$ with the best known nonisolated feasible solution and set γ as described in Step 4; otherwise do not set $\bar{\boldsymbol{x}}$ and choose $\gamma \leq \frac{f^+(\boldsymbol{x},\boldsymbol{\xi})}{f^-(\boldsymbol{x},\boldsymbol{\xi})}$ for all feasible $(\boldsymbol{x}, \boldsymbol{\xi})$.

Step 1 For each box $\mathcal{M} \in \mathscr{P}_k$:

- Compute $\beta(\mathcal{M})$. Set $\beta(\mathcal{M}) = \infty$ if (5.7) (or (5.9)) is infeasible.
- Add $\mathcal{M}$ to $\mathscr{R}_k$ if $\beta(\mathcal{M}) \leq -\varepsilon$.

Step 2 Terminate if $\mathscr{R}_k = \emptyset$: If $\bar{\boldsymbol{x}}$ is not set, then (5.1) is ε-essential infeasible; else $\bar{\boldsymbol{x}}$ is an essential (ε, η)-optimal solution of (5.1).

Step 3 Let $\mathcal{M}^k = \operatorname{argmin}\{\beta(\mathcal{M}) \mid \mathcal{M} \in \mathscr{R}_k\}$. Let $\boldsymbol{x}^k$ be the optimal solution of (5.7) for the box $\mathcal{M}^k$ (or $\underline{\boldsymbol{x}}^*_{\mathcal{M}^k}$ if (5.9) is employed for bounding), and $\boldsymbol{y}^k = \bar{\boldsymbol{x}}^*_{\mathcal{M}^k}$. Solve the feasibility problem

$$\begin{cases} \text{find} & \boldsymbol{\xi} \in \mathcal{C}_{\boldsymbol{x}^k} \\ \text{s. t.} & g_i^+(\boldsymbol{x}^k, \boldsymbol{\xi}) - g_i^-(\boldsymbol{x}^k) \leq 0, \quad i = 1, 2, \ldots, m. \end{cases} \tag{5.16}$$

If (5.16) is feasible go to Step 4; otherwise go to Step 5.

Step 4 $\boldsymbol{x}^k$ is a nonisolated feasible solution satisfying $\frac{f^+(\boldsymbol{x}^k,\boldsymbol{\xi})}{f^-(\boldsymbol{x}^k,\boldsymbol{\xi})} \geq \gamma$ for some $\boldsymbol{\xi} \in \mathcal{C}_{\boldsymbol{x}^k}$. Let $\boldsymbol{\xi}^*$ be a solution to

$$\begin{cases} \min\limits_{\boldsymbol{\xi} \in \mathcal{C}_{\boldsymbol{x}^k}} & \dfrac{f^+(\boldsymbol{x}^k, \boldsymbol{\xi})}{f^-(\boldsymbol{x}^k, \boldsymbol{\xi})} \\ \text{s. t.} & g_i^+(\boldsymbol{x}^k, \boldsymbol{\xi}) - g_i^-(\boldsymbol{x}^k) \leq 0, \quad i = 1, 2 \ldots, m. \end{cases} \tag{5.17}$$

If $\bar{\boldsymbol{x}}$ is not set or $\frac{f^+(\boldsymbol{x}^k,\boldsymbol{\xi}^*)}{f^-(\boldsymbol{x}^k,\boldsymbol{\xi}^*)} > \gamma - \eta$, set $\bar{\boldsymbol{x}} = \boldsymbol{x}^k$ and $\gamma = \frac{f^+(\boldsymbol{x}^k,\boldsymbol{\xi}^*)}{f^-(\boldsymbol{x}^k,\boldsymbol{\xi}^*)} + \eta$.

Step 5 Bisect $\mathcal{M}^k$ via $(\boldsymbol{v}^k, j_k)$ where $j_k \in \operatorname{argmax}_j\{|y_j^k - x_j^k|\}$ and $\boldsymbol{v}^k = \frac{1}{2}(\boldsymbol{x}^k + \boldsymbol{y}^k)$ (cf. (3.7)). Let $\mathscr{R}_{k+1} = \mathscr{R}_k \setminus \{\mathcal{M}^k\}$. Let $\mathscr{P}_{k+1} = \{\mathcal{M}^k_-, \mathcal{M}^k_+\}$. Increment k and go to Step 1.

be strict, the implication in (5.10) does not hold anymore and a "$\beta(\mathcal{M}^k) < -\varepsilon$" termination criterion might stop Algorithm 5 prematurely. ◁

5.12 Remark (Reduction). As in any other BB algorithm, we could add an additional reduction step before computing the bound of a box. This step is optional and should only be included if it speeds up the algorithm. A direct application of the reduction procedure of the original SIT algorithm in [128] requires the solution of several convex optimization problems. This slowed down the algorithm in preliminary numerical experiments and is thus omitted. The design of an efficient reduction procedure is left open for future work. ◁

5.13 Remark (Pointwise Minimum). Algorithm 5 is easily extended to the case were the objective is the pointwise minimum of several functions, i.e.,

$$\max_{(\boldsymbol{x},\boldsymbol{\xi})\in\mathcal{C}} \min_j \left\{ \frac{f_j^+(\boldsymbol{x},\boldsymbol{\xi})}{f_j^-(\boldsymbol{x},\boldsymbol{\xi})} \right\} \quad \text{s. t.} \quad g_i^+(\boldsymbol{x},\boldsymbol{\xi}) - g_i^-(\boldsymbol{x}) \leq 0,\ \ i = 1, 2, \dots, m.$$

In that case, all fractions in the minimum need to be greater or equal than γ and the dual Problem (5.2) becomes

$$\begin{cases} \displaystyle\min_{(\boldsymbol{x},\boldsymbol{\xi})\in\mathcal{C}} & \displaystyle\max_{i=1,2,\dots,m} \left(g_i^+(\boldsymbol{x},\boldsymbol{\xi}) - g_i^-(\boldsymbol{x})\right) \\ \text{s. t.} & \gamma f_j^-(\boldsymbol{x},\boldsymbol{\xi}) - f_j^+(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, \quad j = 1, 2, \dots \end{cases}$$

The necessary modifications to Algorithm 5 are straightforward. ◁

5.4 Benchmark

The goal of this section is to assess how Algorithm 5 scales with an increasing number of global and nonglobal variables. This is done by means of an idealized example. Namely, consider the K-user GIC from Section 2.1 and its PTP capacity region in Proposition 2.1.

For the benchmark, assume that receiver i jointly decodes its own message and the interfering message of transmitter $i + 1 \mod K$. All other messages are treated as noise. Moreover, assume that $h_{ij} = 0$ for $i \neq j, j \neq i+1 \mod K$, $j > \kappa$, and some positive κ. These assumptions allow to precisely control the complexities in the global and nonglobal variables. Thus, $\mathcal{S}_k = \{k, k + 1$

mod $K\}$ for all k and $\mathcal{S}_k^c = \{1, \dots, \kappa\} \setminus \mathcal{S}_k$. Maximizing the throughput in this system requires the solution of

$$\begin{cases} \max\limits_{\boldsymbol{p}, \boldsymbol{R}} & \sum\limits_{k=1}^{K} R_k \\ \text{s.t.} & \boldsymbol{p} \in [\boldsymbol{0}, \bar{\boldsymbol{P}}], \quad \boldsymbol{R} \in \mathcal{A}_k(\mathcal{S}_k), \text{ for all } k, \end{cases} \tag{5.18}$$

where $\mathcal{A}_k(\mathcal{S}_k)$ is as defined in (2.2). First, observe that this is a nonconvex optimization problem due to $P_k(\mathcal{S}_k^c)$ in the denominators of the log-terms in (2.2). Further, observe that $\mathcal{S}_k^c \subseteq \{1, \dots, \kappa\}$ for all $k = 1, \dots, K$. Hence, we can identify the global variables as $(p_k)_{k=1}^{\kappa}$ and the nonglobal variables as $(p_k)_{\kappa+1}^{K}$ and $\boldsymbol{R}$. Moreover, the powers $(p_k)_{\kappa+1}^{K}$ only occur in the numerators of the log-terms in (2.2) which are increasing functions in $(p_k)_{\kappa+1}^{K}$. Thus, $p_k = \bar{P}_k$ is the optimal solution for all $k = \kappa + 1, \dots, K$. Plugging-in this partial solution in (5.18), we obtain a simplified version of (5.18) with κ global variables $(p_k)_{k=1}^{\kappa}$, K linear variables $\boldsymbol{R}$, and $3K$ inequality constraints. Thus, we can independently control the number of global and nonglobal variables. Since the number of constraints grows linearly with the number of global variables and the bounding problem is linear, the complexity of solving (5.18) should grow polynomially with the number of nonglobal variables.

This rationale is illustrated in Fig. 5.1 where (5.18) is solved for a fixed number of κ global variables and an increasing number of K total users in the system. We employ Algorithm 5 with Gurobi [38] as inner solver and choose $\bar{P}_k = 1$ and $N_k = 0.01$ for all $k = 1, \dots, K$, i.e., a transmit signal-to-noise ratio (SNR) of 20 dB, and $h_{kj} \sim \mathcal{CN}(0,1)$. Each data point is obtained by averaging over 1000 i.i.d. channel realizations. Algorithm 5 was started with parameters $\eta = 0.01$, $\varepsilon = 10^{-5}$, and $\gamma = 0$. Since the abscissa is directly proportional to the number of nonglobal variables ($K - \kappa$) and the number of inequality constraints ($3K$), we expect to observe a polynomial growth of the run time. Indeed, Fig. 5.1 displays a roughly linear slope expect for small K. This is because, for big K, the optimal power allocation converges to zero for all global variables. Instead, for small K these powers are generally nonzero and the global part of the optimization problem requires more time to converge to the optimal solution.

Finally, Fig. 5.2 illustrates the exponential complexity in the number of global variables. With the same parameters as in Fig. 5.1, Fig. 5.2 displays the run time for a fixed number of $K = 7$ users over the number of global variables κ. Due to the logarithmic y-axis, exponential growth corresponds to (at least) an affine function in Fig. 5.2.

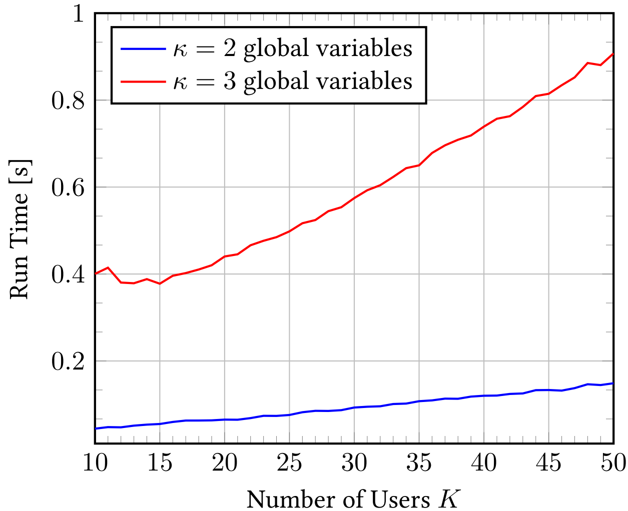


Figure 5.1: Average run time of Algorithm 5 to solve (5.18) as a function of the total number of users for fixed number of κ global variables. The abscissa is proportional to the number of nonglobal variables $(K - \kappa)$. Averaged over 1000 i.i.d. channel realizations [81]. ©2019 IEEE

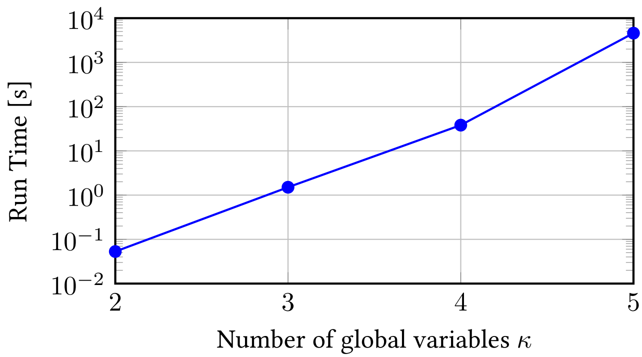


Figure 5.2: Average run time of Algorithm 5 to solve (5.18) as a function of the number of global variables κ for fixed number of $K = 7$ users. Averaged over 1000 i.i.d. channel realizations [81]. ©2019 IEEE

Chapter 6

Mixed Monotonic Programming

We have seen in Chapter 4 that the monotonic optimization framework's requirement to reformulate every function as the difference of increasing functions can be quite limiting in practice. A good example for this is given in Section 4.3 where a fractional objective with DI numerator and denominator is maximized. A likely reason for this is that the initial design of the monotonic optimization framework was guided by the requirements of the polyblock algorithm. In particular, the polyblock algorithm solves the canonical monotonic optimization problem (3.13) with increasing objective and two constraints defined by increasing functions. Instead, BB procedures are not limited in this way and only require that a suitable upper bound can be computed. In this chapter, we develop a novel optimization framework that exploits monotonicity properties of the optimization problem in a much more general way than monotonic programming.

Consider the optimization problem

$$\max_{\boldsymbol{x}\in\mathcal{D}} \; f(\boldsymbol{x}) \tag{6.1}$$

with continuous objective function $f : \mathbb{R}^n \to \mathbb{R}$ and compact feasible set $\mathcal{D} \subseteq \mathbb{R}^n$. For now, we do not need any further assumptions on $\mathcal{D}$ and postpone the discussion of its structure to Section 6.3. If the objective function has a mixed monotonic programming (MMP) representation, Problem (6.1) can be solved efficiently with a BB procedure.

6.1 Definition. Let $F : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be a continuous function such that

$$F(\boldsymbol{x},\boldsymbol{y}) \leq F(\boldsymbol{x}',\boldsymbol{y}) \qquad \text{if } \boldsymbol{x} \leq \boldsymbol{x}', \tag{6.2}$$

$$F(\boldsymbol{x},\boldsymbol{y}) \geq F(\boldsymbol{x},\boldsymbol{y}') \qquad \text{if } \boldsymbol{y} \leq \boldsymbol{y}'. \tag{6.3}$$

Then, F is a *mixed monotonic (MM) function.* ◁

6.2 Definition. Let $\mathcal{M}_0 = [\boldsymbol{r}^0, \boldsymbol{s}^0] \supseteq \mathcal{D}$ be a box enclosing $\mathcal{D}$. Assume there exists an MM function F on $\mathcal{M}_0$ such that

$$F(\boldsymbol{x}, \boldsymbol{x}) = f(\boldsymbol{x}), \tag{6.4}$$

for all $\boldsymbol{x} \in \mathcal{M}_0$. Then, (6.1) is a *MMP problem* and F is denoted as the *MMP representation of f*. ◁

Note that, in general, there can exist more than one MMP representation for a function f. In the next section, we explore properties of MM functions and consider some examples. Then, in Sections 6.2 to 6.4, we design a BB procedure to solve (6.1) based on the MMP representation of f. This chapter is concluded with performance benchmarks in Section 6.5.

6.1 Mixed Monotonic Functions

In this section we discuss some properties and examples of MM functions. We start with the following proposition which lists some operations that preserve the MM properties of functions.

6.3 Proposition. Let $F_i(\boldsymbol{x}, \boldsymbol{y})$ be MM functions for $i = 1, \dots, K$, $g(x)$ a real valued, nondecreasing function, and $h(x)$ a real, nonincreasing function. Then,

$$(\boldsymbol{x}, \boldsymbol{y}) \mapsto \sum_{i=1}^{K} F_i(\boldsymbol{x}, \boldsymbol{y}), \tag{6.5}$$

$$(\boldsymbol{x}, \boldsymbol{y}) \mapsto \max_{i=1,\dots,K} F_i(\boldsymbol{x}, \boldsymbol{y}), \qquad (\boldsymbol{x}, \boldsymbol{y}) \mapsto \min_{i=1,\dots,K} F_i(\boldsymbol{x}, \boldsymbol{y}), \tag{6.6}$$

$$(\boldsymbol{x}, \boldsymbol{y}) \mapsto g(F_i(\boldsymbol{x}, \boldsymbol{y})), \qquad (\boldsymbol{x}, \boldsymbol{y}) \mapsto h(F_i(\boldsymbol{y}, \boldsymbol{x})). \tag{6.7}$$

are MM functions. If in addition $F_i(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ for all $i = 1, \dots, K$ and $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ for some $\mathcal{X} \in \mathbb{R}^n$, then

$$(\boldsymbol{x}, \boldsymbol{y}) \mapsto \prod_{i=1}^{K} F_i(\boldsymbol{x}, \boldsymbol{y}) \tag{6.8}$$

is an MM function on $\mathcal{X}$. ◁

Proof. Equations (6.5) and (6.6) are trivially verified from Definition 6.1. For example, take the pointwise maximum and let $\boldsymbol{x}' \geq \boldsymbol{x}$. Then,

$$\max_{i=1,\dots,K} F_i(\boldsymbol{x}, \boldsymbol{y}) \geq \max_{i=1,\dots,K} F_i(\boldsymbol{x}', \boldsymbol{y})$$

because, for all i, $F_i(\boldsymbol{x}, \boldsymbol{y}) \geq F_i(\boldsymbol{x}', \boldsymbol{y})$. Similarly, with $\boldsymbol{y}' \geq \boldsymbol{y}$,

$$\max_{i=1,\dots,K} F_i(\boldsymbol{x}, \boldsymbol{y}) \leq \max_{i=1,\dots,K} F_i(\boldsymbol{x}, \boldsymbol{y}')$$

since, for all i, $F_i(\boldsymbol{x}, \boldsymbol{y}) \leq F_i(\boldsymbol{x}, \boldsymbol{y}')$ by definition. Moreover, for (6.8) if all factors in the product are positive, the MM properties of the resulting function are straightforwardly verified from Definition 6.1.

For (6.7), consider the first partial derivatives. Due to the chain rule

$$\nabla_{\boldsymbol{x}}\, g(F_i(\boldsymbol{x}, \boldsymbol{y})) = \left.\frac{dg(z)}{dz}\right|_{z=F_i(\boldsymbol{x},\boldsymbol{y})} \nabla_{\boldsymbol{x}} F_i(\boldsymbol{x}, \boldsymbol{y}). \tag{6.9}$$

Since $g(x)$ is a nondecreasing function, $g'(x) \geq 0$ and the RHS of (6.9) has the same sign as $\frac{\partial}{\partial \boldsymbol{x}} F_i(\boldsymbol{x}, \boldsymbol{y})$. Thus, the monotonicity properties of $F(\boldsymbol{x}, \boldsymbol{y})$ are preserved. Similarly, for the second property, $h'(x) \leq 0$ and the monotonicity properties are inversely preserved from $F(\boldsymbol{x}, \boldsymbol{y})$. Thus, the resulting function is increasing in the second argument and decreasing in the first argument of the original MM function F. □

In particular, it follows from (6.7) that $(\boldsymbol{x}, \boldsymbol{y}) \mapsto -F(\boldsymbol{y}, \boldsymbol{x})$ and $(\boldsymbol{x}, \boldsymbol{y}) \mapsto 1/F(\boldsymbol{y}, \boldsymbol{x})$ are MM if $F(\boldsymbol{x}, \boldsymbol{y})$ is a positive MM function. The next example highlights the connection between MMP and monotonic programming.

6.4 Example (Monotonic Programming). Consider the monotonic optimization problem

$$\max_{\boldsymbol{x} \in \mathcal{D}} \; f^+(\boldsymbol{x}) - f^-(\boldsymbol{x})$$

where f^+ and f^- are increasing functions. A MMP representation of that objective is $(\boldsymbol{x}, \boldsymbol{y}) \mapsto f^+(\boldsymbol{x}) - f^-(\boldsymbol{y})$. Thus, monotonic optimization is a special case of MMP. ◁

Since DI functions have a straightforward MMP representation, it follows that Proposition 4.2 also holds for MM functions. Hence, the following corollary.

6.5 Corollary. For a function f, there exists an MMP representation F such that $f(\boldsymbol{x}) = F(\boldsymbol{x}, \boldsymbol{x})$ on $[\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}] \subset \mathbb{R}^n_{\geq 0}$, if f is

1. a polynomial, and more generally, a function $\boldsymbol{x} \mapsto \sum_{k=1}^{K} c_k x_1^{a_{k1}} \cdots x_n^{a_{kn}}$ with real-valued, nonzero exponents a_{ki} and real constants c_k, for all $k = 1, \dots, K$ and $i = 1, \dots, n$, or
2. a convex function $f : [\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}] \to \mathbb{R}$, or
3. a Lipschitz function $f : [\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}] \to \mathbb{R}$ with Lipschitz constant K, or

4. a continuously differentiable function $f : [\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}] \to \mathbb{R}$. ◁

Proof. For the function in Property 1, construct the MM function as follows: For each k and i, the x_i in the term $c_k x_1^{a_{k1}} \cdots x_n^{a_{kn}}$ is replaced by y_i if $c_k > 0$ and $a_{ki} < 0$ or if $c_k < 0$ and $a_{ki} > 0$. The resulting function is increasing in $\boldsymbol{x}$ and decreasing in $\boldsymbol{y}$.

Properties 2–4 follow directly from Proposition 4.2 since every DI function has an MMP representation. □

However, the class of functions with MMP representation does not only include DI functions but is generally much more versatile. The following example shows how to treat fractional objectives of DI functions without the need of Dinkelbach's algorithm (cf. Section 4.3).

6.6 Example (Fractional Monotonic Programming). Consider the fraction of a DI function and an increasing function, i.e.,

$$\frac{p^+(\boldsymbol{x}) - p^-(\boldsymbol{x})}{q(\boldsymbol{x})} \tag{6.10}$$

where p^+, p^-, q are increasing functions. A MMP representation of (6.10) is

$$(\boldsymbol{x}, \boldsymbol{y}) \mapsto \frac{p^+(\boldsymbol{x}) - p^-(\boldsymbol{y})}{q(\boldsymbol{y})}.$$

Thus, maximizing (6.10) is an MMP problem and can be solved directly with the MMP algorithm in Section 6.2. Instead, maximizing (6.10) with the monotonic optimization framework requires the combination of Dinkelbach's algorithm with monotonic programming as explained in Section 4.3.

This bounding approach is easily extended beyond the capabilities of fractional monotonic programming. For example, consider the sum of ratios

$$\sum_i \frac{p_i^+(\boldsymbol{x}) - p_i^-(\boldsymbol{x})}{q_i(\boldsymbol{x})}. \tag{6.11}$$

Neither Dinkelbach's algorithm nor its generalization [24, 25] are able to solve sum-of-ratios problems.[27] Instead, in [147] it is proposed to reformulate (6.11) as

$$\frac{\sum_i \left[(p_i^+(\boldsymbol{x}) - p_i^-(\boldsymbol{x})) \prod_{j \neq i} q_j(\boldsymbol{x}) \right]}{\prod_j q_j(\boldsymbol{x})}$$

[27] A Dinkelbach-type algorithm to solve the sum-of-ratios problem has been proposed in [2] but it was shown in [31] that it does not converge to the globally optimal solution.

which is a single ratio fractional program and can be solved by the fractional monotonic programming framework. However, it is shown in [82] that this approach has very poor convergence behavior. Instead, from (6.5) it is apparent that an MMP representation of (6.11) is

$$(\boldsymbol{x}, \boldsymbol{y}) \mapsto \sum_i \frac{p_i^+(\boldsymbol{x}) - p_i^-(\boldsymbol{y})}{q_i(\boldsymbol{y})}. \qquad \triangleleft$$

We conclude this section with a more concrete example that highlights how the MMP approach eases the solution of a well known resource allocation problem. In particular, note that no reformulation of the objective is necessary.

6.7 Example (Treating Interference as Noise). Consider the WSR in a K-user GIC where interference is treated as noise. The achievable rate region is given in (2.6) and the maximization of this function with monotonic optimization is discussed in Example 4.1. In particular, we have

$$\sum_k w_k \, \mathrm{C}\left(\frac{|h_{kk}|^2 p_k}{\sigma_k^2 + \sum_{j \neq k} |h_{kj}|^2 p_j}\right) \tag{6.12}$$

with nonnegative weights w_k. A MMP representation for each term can be obtained directly from (6.12) as

$$(\boldsymbol{x}, \boldsymbol{y}) \mapsto \mathrm{C}\left(\frac{|h_{kk}|^2 x_k}{\sigma_k^2 + \sum_{j \neq k} |h_{kj}|^2 y_j}\right).$$

Instead, monotonic optimization operates on the DI representation

$$(\boldsymbol{x}, \boldsymbol{y}) \mapsto \log\left(|h_{kk}|^2 x_k + \sum_{j \neq k} |h_{kj}|^2 x_j + \sigma_k^2\right) - \log\left(\sum_{j \neq k} |h_{kj}|^2 y_j + \sigma_k^2\right) \tag{6.13}$$

which is also an MM function. This example will be continued in Section 6.5.1. An important aspect discussed there is that the precise choice of F directly impacts the convergence speed of the MMP algorithm. $\triangleleft$

6.2 MMP Algorithm

We design a BB algorithm to determine a global η-optimal solution of (6.1). The core idea of BB is to relax the feasible set $\mathcal{D}$ and subsequently partition it such that upper bounds on the objective value can be determined easily. This is where the MMP representation F of the objective function f comes in handy, since it is especially well suited to compute upper bounds over rectangular sets. Let $\mathcal{M} = [\boldsymbol{r}, \boldsymbol{s}]$ be a box in $\mathbb{R}^n$. Then,

$$\max_{\boldsymbol{x}\in\mathcal{M}\cap\mathcal{D}} f(\boldsymbol{x}) \leq \max_{\boldsymbol{x}\in\mathcal{M}} F(\boldsymbol{x}, \boldsymbol{x}) \leq \max_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{M}} F(\boldsymbol{x}, \boldsymbol{y}) = F(\boldsymbol{s}, \boldsymbol{r}) \tag{6.14}$$

is an upper bound on the optimal value of $f(\boldsymbol{x})$ on $\mathcal{M} \cap \mathcal{D}$. Thus, rectangular subdivision is an excellent choice to partition $\mathcal{D}$. In particular, exhaustive bisection is employed because the bounding procedure is not suitable for adaptive bisection.[28]

With these observations, we can specialize Algorithm 1 to the MMP algorithm stated in Algorithm 6. Its convergence is formally established in the theorem below.

6.8 Theorem. Algorithm 6 converges towards a global η-optimal solution of (6.1) if it is an MMP problem and the selection is bound-improving. $\triangleleft$

Proof. By virtue of Proposition 3.7 and Corollary 3.9, Algorithm 6 converges to the η-optimal solution of (6.1) if the selection is bound-improving and bounding is consistent with branching, i.e.,

$$F(\boldsymbol{s}, \boldsymbol{r}) - \max\{f(\boldsymbol{x}) \,|\, \boldsymbol{x} \in [\boldsymbol{r}, \boldsymbol{s}] \cap \mathcal{D}\} \to 0 \tag{6.15}$$

as $\|\boldsymbol{s} - \boldsymbol{r}\| \to 0$. As $\|\boldsymbol{s} - \boldsymbol{r}\| \to 0$, $\boldsymbol{r}, \boldsymbol{s}$ converge to a common limit point $\boldsymbol{y}$. Then, (6.15) approaches

$$F(\boldsymbol{y}, \boldsymbol{y}) - f(\boldsymbol{y}) = f(\boldsymbol{y}) - f(\boldsymbol{y}) = 0$$

and the convergence criterion is satisfied. $\square$

6.3 The Feasibility Check

Algorithm 6 requires in every iteration k and for each box $\mathcal{M} \in \mathscr{P}_k$ a feasible point $\boldsymbol{x} \in \mathcal{M} \cap \mathcal{D}$ in Step 3 and a conclusive rule to prune infeasible boxes, i.e.,

[28] Recall that adaptive bisection requires the bounding procedure to generate two points $\boldsymbol{u}^k, \boldsymbol{v}^k$ satisfying (3.9). Bounding as in (6.14) "generates" only the two points $\boldsymbol{r}^k, \boldsymbol{s}^k$ which result in an adaptive bisection that is equivalent to exhaustive bisection.

Algorithm 6 MMP BB Algorithm

Step 0 **(Initialization)** Choose $\mathcal{M}_0 \supseteq \mathcal{D}$ and $\eta > 0$. Let $k = 1$ and $\mathscr{R}_0 = \{\mathcal{M}_0\}$. If available or easily computable, find $\bar{\boldsymbol{x}}^0 \in \mathcal{D}$ and set $\gamma_0 = f(\bar{\boldsymbol{x}}^0)$. Otherwise, set $\gamma_0 = -\infty$.

Step 1 **(Branching)** Select a box $\mathcal{M}_k = [\boldsymbol{r}^k, \boldsymbol{s}^k] \in \mathscr{R}_{k-1}$ and bisect $\mathcal{M}_k$ via $(\frac{1}{2}(\boldsymbol{s}^k + \boldsymbol{r}^k), j)$ with $j \in \operatorname{argmax}_j s_j^k - r_j^k$. Let $\mathscr{P}_k = \{\mathcal{M}_k^-, \mathcal{M}_k^+\}$ with $\mathcal{M}_k^-, \mathcal{M}_k^+$ as in (3.7).

Step 2 **(Reduction)** For each $\mathcal{M} \in \mathscr{P}_k$, replace $\mathcal{M}$ by $\mathcal{M}'$ such that $\mathcal{M}' \subseteq \mathcal{M}$ and

$$(\mathcal{M} \setminus \mathcal{M}') \cap \{\boldsymbol{x} \in \mathcal{D} \,|\, F(\boldsymbol{x}, \boldsymbol{x}) > \gamma_k\} = \emptyset. \tag{6.16}$$

Step 3 **(Incumbent)** For each $\mathcal{M} \in \mathscr{P}_k$, find $\boldsymbol{x} \in \mathcal{M} \cap \mathcal{D}$ and set $\alpha(\mathcal{M}) = f(\boldsymbol{x})$. If $\mathcal{M} \cap \mathcal{D} = \emptyset$, set $\alpha(\mathcal{M}) = -\infty$. Let $\alpha_k = \max\{\alpha(\mathcal{M}) \,|\, \mathcal{M} \in \mathscr{P}_k\}$. If $\alpha_k > \gamma_{k-1}$, set $\gamma_k = \alpha_k$ and let $\bar{\boldsymbol{x}}^k \in \mathcal{D}$ such that $\alpha_k = f(\bar{\boldsymbol{x}}^k)$. Otherwise, let $\gamma_k = \gamma_{k-1}$ and $\bar{\boldsymbol{x}}^k = \bar{\boldsymbol{x}}^{k-1}$.

Step 4 **(Pruning)** Delete every $\mathcal{M} = [\boldsymbol{r}, \boldsymbol{s}] \in \mathscr{P}_k$ with $\mathcal{M} \cap \mathcal{D} = \emptyset$ or $F(\boldsymbol{s}, \boldsymbol{r}) \leq \gamma_k + \eta$. Let $\mathscr{P}_k'$ be the collection of remaining sets and set $\mathscr{R}_k = \mathscr{P}_k' \cup (\mathscr{R}_{k-1} \setminus \mathscr{S}_k)$.

Step 5 **(Termination)** Terminate if $\mathscr{R}_k = \emptyset$ or, optionally, if $\{[\boldsymbol{r}, \boldsymbol{s}] \in \mathscr{R}_k \,|\, F(\boldsymbol{s}, \boldsymbol{r}) > \gamma_k + \eta\} = \emptyset$. Return $\bar{\boldsymbol{x}}^k$ as a global η-optimal solution. Otherwise, update $k \leftarrow k + 1$ and return to Step 1.

$\mathcal{M} \cap \mathcal{D} = \emptyset$, in Step 4. This requires some assumptions on the feasible set $\mathcal{D}$. In particular, assume that $\mathcal{D}$ is given by m inequalities, i.e.,

$$\mathcal{D} = \{\boldsymbol{x} \,|\, g_i(\boldsymbol{x}) \leq 0,\ i = 1, \dots, m\}.$$

The most obvious approach in the context of this chapter is to further assume that each function $g_i(\boldsymbol{x})$ has an MMP representation $G_i(\boldsymbol{x}, \boldsymbol{x}) = g_i(\boldsymbol{x})$, i.e.,

$$\mathcal{D} = \{\boldsymbol{x} \,|\, G_i(\boldsymbol{x}, \boldsymbol{x}) \leq 0,\ i = 1, \dots, m\} \tag{6.17}$$

where G_i satisfies (6.2) and (6.3). These properties lead to the following sufficient conditions for (in-)feasibility of $\mathcal{M}$.

6.9 Proposition. Let $\mathcal{M} = [\boldsymbol{r}, \boldsymbol{s}]$ and $\mathcal{D}$ as in (6.17). Then,

$$\forall i \in \{1, \dots, m\} : G_i(\boldsymbol{s}, \boldsymbol{r}) \leq 0 \Rightarrow \mathcal{M} \cap \mathcal{D} = \mathcal{M} \neq \emptyset \tag{6.18a}$$

$$\exists i \in \{1, \dots, m\} : G_i(\boldsymbol{r}, \boldsymbol{s}) > 0 \Rightarrow \mathcal{M} \cap \mathcal{D} = \emptyset. \tag{6.18b}$$

◁

Proof. From (6.2) and (6.3), $G_i(\boldsymbol{x}, \boldsymbol{x}) \leq G_i(\boldsymbol{s}, \boldsymbol{r})$ and $G_i(\boldsymbol{x}, \boldsymbol{x}) \geq G_i(\boldsymbol{r}, \boldsymbol{s})$ for all $\boldsymbol{x} \in \mathcal{M}$. Thus, if $G_i(\boldsymbol{s}, \boldsymbol{r}) \leq 0$ for all $i = 1, \dots, l$, then $G_i(\boldsymbol{x}, \boldsymbol{x}) \leq 0$ for all i and $\boldsymbol{x} \in \mathcal{M}$. Hence, (6.18a). Similarly, if $G_i(\boldsymbol{r}, \boldsymbol{s}) > 0$ for some $i = 1, \dots, l$, then also $G_i(\boldsymbol{x}, \boldsymbol{x}) > 0$ for this i and all $\boldsymbol{x} \in \mathcal{M}$. Thus, $\boldsymbol{x} \notin \mathcal{D}$ and (6.18b) holds. □

It is easy to construct a box $\mathcal{M}$ and feasible set $\mathcal{D}$ such that neither (6.18a) nor (6.18b) holds. In that case, it is impossible to decide the feasibility of $\mathcal{M}$ with Proposition 6.9. Thus, without further assumptions on $\mathcal{D}$, a conclusive feasibility test is not available for (6.1) and Algorithm 6 is not applicable. However, we can alter it as proposed in Sections 3.2 and 4.2.1 to obtain an infinite version of Algorithm 6. We postpone this to Section 6.3.1 and first discuss two sets of additional assumptions on $\mathcal{D}$ that lead to sufficient feasibility tests for Algorithm 6.

Suppose $\mathcal{D}$ is defined by the system

$$g_i(\boldsymbol{x}) \leq 0,\ i = 1, \dots, m, \quad h_j(\boldsymbol{x}) = 0,\ j = 1, \dots, l.$$

with g_i convex and h_j affine. Then, $\mathcal{D}$ is a closed convex set and the feasibility check can be performed by standard tools from convex optimization [11, 89]. However, since this check is performed up to two times per iteration it is still quite costly.

Instead, let $\mathcal{D}$ be defined by MM functions as in (6.17) where G_i not only satisfies (6.2) and (6.3) but also, for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$,

$$G_i\left(\sum_{j\in\mathcal{I}} x_j \boldsymbol{e}_j, \sum_{k\in\mathcal{I}^c} y_k \boldsymbol{e}_k\right) = G_i(\boldsymbol{x}, \boldsymbol{y}) \tag{6.19}$$

for some index set $\mathcal{I} \subseteq \{1, \dots, n\}$ and all $i = 1, \dots, m$ where $\mathcal{I}^c = \{1, \dots, n\} \setminus \mathcal{I}$. That is, each function $g_i(\boldsymbol{x}) = G_i(\boldsymbol{x}, \boldsymbol{x})$ is increasing in the variables x_j, $j \in \mathcal{I}$, and decreasing in the remaining variables x_k, $k \in \mathcal{I}^c$. Then, the following proposition holds.

6.10 Proposition. Let $\mathcal{M} = [\boldsymbol{r}, \boldsymbol{s}]$ and $\mathcal{D}$ be defined as in (6.17) by MM functions $G_i(\boldsymbol{x}, \boldsymbol{y})$ satisfying (6.2), (6.3) and (6.19). Then, $\mathcal{M} \cap \mathcal{D} \neq \emptyset$ if and only if $G_i(\boldsymbol{r}, \boldsymbol{s}) \leq 0$ for all $i = 1, \dots, m$. In that case, $\sum_{j\in\mathcal{I}} r_j \boldsymbol{e}_j + \sum_{k\in\mathcal{I}^c} s_k \boldsymbol{e}_k \in \mathcal{M} \cap \mathcal{D}$ with $\mathcal{I}$ and $\mathcal{I}^c$ as in (6.19). ◁

Proof. Let $\boldsymbol{\xi} = \sum_{j\in\mathcal{I}} r_j \boldsymbol{e}_j + \sum_{k\in\mathcal{I}^c} s_k \boldsymbol{e}_k$. Then, for all i and due to (6.19), $G_i(\boldsymbol{\xi}, \boldsymbol{\xi}) = G_i(\boldsymbol{r}, \boldsymbol{s})$. Thus, if $G_i(\boldsymbol{r}, \boldsymbol{s}) \leq 0$, then $\boldsymbol{\xi} \in \mathcal{D}$. Since, trivially, $\boldsymbol{\xi} \in \mathcal{M}$, $\boldsymbol{\xi} \in \mathcal{D} \cap \mathcal{M} \neq \emptyset$. Finally, from (6.18b) follows $\mathcal{M} \cap \mathcal{D} \neq \emptyset \Rightarrow G_i(\boldsymbol{r}, \boldsymbol{s}) \leq 0$. □

Observe, that we can obtain Lemmas 4.4 and 4.5 as corollaries from Proposition 6.10 by setting $\mathcal{I} = \{1, \dots, n\}$ and $\mathcal{I} = \emptyset$, respectively.

Proposition 6.10 provides a powerful tool to solve a wide array of feasibility problems. However, note that this property is not necessary as long as the feasibility check in Algorithm 6 is somehow implemented (efficiently).

6.3.1 Infinite MMP Algorithm

In this section we alter Algorithm 6 such that it works with a feasible set defined by MM functions as in (6.17). In particular, no assumptions apart from G_i being MM functions for all i are required. That is, we design an algorithm to obtain an η-optimal global solution of

$$\max_{\boldsymbol{x} \in [\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}]} \; F(\boldsymbol{x}, \boldsymbol{x}) \quad \text{s.t.} \quad G_i(\boldsymbol{x}, \boldsymbol{x}) \leq 0, \; i = 1, 2, \dots, m \tag{6.20}$$

with $F, G_1, \dots, G_m$ MM functions. The devised procedure will have infinite convergence in theory, but, in practice, it often converges in reasonable time.

The necessary changes to Algorithm 6 are discussed in Section 3.2. Specifically, in Step 3, a feasible point is only required if available, and, in Step 4, boxes are only pruned if (6.18b) is met. The modified procedure is stated in Algorithm 7. In Step 3, a point $\boldsymbol{x} \in \mathcal{M}$ is required. This can be any easily obtainable point in $\mathcal{M}$, e.g., $\boldsymbol{r}$, $\boldsymbol{s}$, or any other corner point of $\mathcal{M}$. The actual choice is quite arbitrary but depending on the feasible set there might be points that are more often feasible than others. Thus, choosing the "right" point for the specific problem at hand might speed up convergence.

Convergence of Algorithm 7 is established in the theorem below. We prove convergence from first principles because infinite BB procedures are not covered in detail by Chapter 3.

6.11 Theorem. For a bound-improving selection, if Algorithm 7 terminates (6.20) is either infeasible or $\bar{\boldsymbol{x}}^k$ is an η-optimal solution of (6.20). If it is infinite, at least one accumulation point of the infinite sequence $\{\boldsymbol{r}^k\}$ is a globally optimal solution of (6.20). ◁

Proof. For the same reasons as in Proposition 3.3, the reduction in Step 2 does not negatively affect the solution of (6.20), and upon termination of Algorithm 7, either $\gamma_K = \gamma_0$ and (6.20) is infeasible or $\bar{\boldsymbol{x}}_K$ satisfies $F(\boldsymbol{x}, \boldsymbol{x}) \leq F(\bar{\boldsymbol{x}}_K, \bar{\boldsymbol{x}}_K) + \eta$ for every $\boldsymbol{x} \in \mathcal{D}$.

Algorithm 7 Infinite MMP BB Algorithm

Step 0 **(Initialization)** Choose $\mathcal{M}_0 \supseteq \mathcal{D}$ and $\eta > 0$. Let $k = 1$ and $\mathscr{R}_0 = \{\mathcal{M}_0\}$. If available or easily computable, find $\bar{\boldsymbol{x}}^0 \in \mathcal{D}$ and set $\gamma_0 = F(\bar{\boldsymbol{x}}^0, \bar{\boldsymbol{x}}^0)$. Otherwise, set $\gamma_0 = -\infty$.

Step 1 **(Branching)** Select a box $\mathcal{M}_k = [\boldsymbol{r}^k, \boldsymbol{s}^k] \in \mathscr{R}_{k-1}$ and bisect $\mathcal{M}_k$ via $(\frac{1}{2}(\boldsymbol{s}^k + \boldsymbol{r}^k), j)$ with $j \in \operatorname{argmax}_j s_j^k - r_j^k$. Let $\mathscr{P}_k = \{\mathcal{M}_k^-, \mathcal{M}_k^+\}$ with $\mathcal{M}_k^-, \mathcal{M}_k^+$ as in (3.7).

Step 2 **(Reduction)** For each $\mathcal{M} \in \mathscr{P}_k$, replace $\mathcal{M}$ by $\mathcal{M}'$ such that $\mathcal{M}' \subseteq \mathcal{M}$ and

$$(\mathcal{M} \setminus \mathcal{M}') \cap \{\boldsymbol{x} \in \mathcal{D} \,|\, F(\boldsymbol{x}, \boldsymbol{x}) > \gamma_k\} = \emptyset.$$

Step 3 **(Incumbent)** For each $\mathcal{M} \in \mathscr{P}_k$, select a point $\boldsymbol{x} \in \mathcal{M}$. If $G_i(\boldsymbol{x}, \boldsymbol{x}) \leq 0$ for all $i = 1, \dots, m$, set $\alpha(\mathcal{M}) = F(\boldsymbol{x}, \boldsymbol{x})$. Otherwise, set $\alpha(\mathcal{M}) = -\infty$. Let $\alpha_k = \max\{\alpha(\mathcal{M}) \,|\, \mathcal{M} \in \mathscr{P}_k\}$. If $\alpha_k > \gamma_{k-1}$, set $\gamma_k = \alpha_k$ and let $\bar{\boldsymbol{x}}^k \in \mathcal{D}$ such that $\alpha_k = F(\bar{\boldsymbol{x}}^k, \bar{\boldsymbol{x}}^k)$. Otherwise, let $\gamma_k = \gamma_{k-1}$ and $\bar{\boldsymbol{x}}^k = \bar{\boldsymbol{x}}^{k-1}$.

Step 4 **(Pruning)** Delete every $\mathcal{M} = [\boldsymbol{r}, \boldsymbol{s}] \in \mathscr{P}_k$ with $F(\boldsymbol{s}, \boldsymbol{r}) \leq \gamma_k + \eta$ or $G_i(\boldsymbol{r}, \boldsymbol{s}) > 0$ for some $i = 1, \dots, m$. Let $\mathscr{P}'_k$ be the collection of remaining sets and set $\mathscr{R}_k = \mathscr{P}'_k \cup (\mathscr{R}_{k-1} \setminus \mathscr{S}_k)$.

Step 5 **(Termination)** Terminate if $\mathscr{R}_k = \emptyset$ or, optionally, if $\{[\boldsymbol{r}, \boldsymbol{s}] \in \mathscr{R}_k \,|\, F(\boldsymbol{s}, \boldsymbol{r}) > \gamma_k + \eta\} = \emptyset$. Return $\bar{\boldsymbol{x}}^k$ as a global η-optimal solution. Otherwise, update $k \leftarrow k + 1$ and return to Step 1.

If Algorithm 7 is infinite, it generates an infinite decreasing subsequence of sets $\{\mathcal{M}_{k_q}\}_q$ such that $G_i(\boldsymbol{r}^{k_q}, \boldsymbol{s}^{k_q}) \leq 0$, for all i and $q = 1, 2, \dots$, and $\mathcal{M}_{k_q} \in \operatorname{argmax}\{F(\boldsymbol{s}, \boldsymbol{r}) \,|\, [\boldsymbol{r}, \boldsymbol{s}] \in \mathscr{R}_{k_q - 1}\}$ due to the bound-improving selection. By virtue of Corollary 3.9, the bisection in Step 1 is exhaustive and, thus, by Proposition 3.7, $\operatorname{diam} \mathcal{M}_{k_q} \to 0$. Thus, $\boldsymbol{r}^{k_q}$ and $\boldsymbol{s}^{k_q}$ approach a common limit point $\boldsymbol{x}$. Due to the continuity of G_i, this point satisfies $G_i(\boldsymbol{x}, \boldsymbol{x}) \leq 0$ for all i. Moreover, $F(\boldsymbol{s}^{k_q}, \boldsymbol{r}^{k_q}) \to F(\boldsymbol{x}, \boldsymbol{x})$ and $F(\boldsymbol{s}^{k_q}, \boldsymbol{r}^{k_q}) \geq \operatorname{argmax}\{F(\boldsymbol{s}, \boldsymbol{r}) \,|\, [\boldsymbol{r}, \boldsymbol{s}] \in \mathscr{R}_{k_q - 1}\} \geq v(6.20)$. □

A popular method to obtain a finite procedure from Algorithm 7 is to relax the constraints by some small $\varepsilon > 0$. This results in an approximately feasible η-optimal solution of (6.20). However, as discussed in Section 3.2.1 this approach is prone to numerical issues and might result in "solutions" that are quite far away from the true optimal value. Instead, the SIT approach avoids these problems and should be employed to solve (6.20) if a finite algorithm is required. This is left open for future work.

6.4 A Reduction Procedure for MM Feasible Sets

The reduction procedure in Step 2 of Algorithm 6 replaces each box $\mathcal{M} \in \mathscr{P}_k$ with a smaller box $\mathcal{M}'$ that still contains all feasible points that might improve the current best known solution. Because smaller boxes result in tighter bounds, this step speeds up the convergence in general. However, it also increases the computation time per iteration and, thus, slows down the algorithm. Ultimately, it depends on the problem at hand, especially the structure of the feasible set $\mathcal{D}$, and the implementation of the reduction procedure whether it should be applied.

For $\mathcal{D}$ convex (linear), the reader is referred to the vast literature on convex (linear) optimization theory regarding possible implementations of the reduction. The most straightforward approach would be to solve the convex (linear) optimization problems

$$r_i' = \min_{\boldsymbol{x} \in \mathcal{M} \cap \mathcal{D}} x_i \qquad\qquad s_i' = \max_{\boldsymbol{x} \in \mathcal{M} \cap \mathcal{D}} x_i$$

for all $i = 1, \dots, n$, and let $\mathcal{M}' = [\boldsymbol{r}', \boldsymbol{s}']$. However, this approach might be computationally too complex to provide any gain.

For an MM feasible set as defined in (6.17), the reduction can be carried out in a similar fashion as for monotonic programming problems [128, §11.2.1].

6.12 Proposition. Let $\mathcal{M} = [\boldsymbol{r}, \boldsymbol{s}]$ and (6.1) be an MMP with MM feasible set, i.e., $\mathcal{D} = \{\boldsymbol{x} \,|\, G_i(\boldsymbol{x}, \boldsymbol{x}) \leq 0,\, i = 1, \dots, m\}$. Then, a reduced set $\mathcal{M}' \subseteq \mathcal{M}$ satisfying (6.16) is

$$\mathcal{M}' = \begin{cases} [\boldsymbol{r}', \boldsymbol{s}'] & \text{if } F(\boldsymbol{s}, \boldsymbol{r}) \geq \gamma_k \text{ and } G_i(\boldsymbol{r}, \boldsymbol{s}) \leq 0 \text{ for all } i \\ \emptyset & \text{otherwise} \end{cases} \tag{6.21}$$

with

$$\boldsymbol{r}' = \boldsymbol{s} - \sum_{i=1}^{n} \alpha_i (s_i - r_i) \boldsymbol{e}_i, \qquad \boldsymbol{s}' = \boldsymbol{r}' + \sum_{i=1}^{n} \beta_i (s_i - r_i') \boldsymbol{e}_i \tag{6.22}$$

and, for all $i = 1, \dots, n$,

$$\begin{aligned} \alpha_i = \sup \Bigg\{ \alpha \in [0,1] \,\Bigg|\, & F(\boldsymbol{s} - \alpha(s_i - r_i)\boldsymbol{e}_i, \boldsymbol{r}) > \gamma_k, \\ & G_j(\boldsymbol{r}, \boldsymbol{s} - \alpha(s_i - r_i)\boldsymbol{e}_i) \leq 0,\, j = 1, \dots, m \Bigg\} \end{aligned} \tag{6.23}$$

$$\beta_i = \sup \left\{ \beta \in [0,1] \,\middle|\, F(\boldsymbol{s}, \boldsymbol{r}' + \beta(s_i - r'_i)\boldsymbol{e}_i) > \gamma_k, \right.$$
$$\left. G_j(\boldsymbol{r}' + \beta(s_i - r'_i)\boldsymbol{e}_i, \boldsymbol{s}) \le 0, j = 1, \dots, m \right\}. \tag{6.24}$$

◁

Proof. Observe from (6.18b) that if, for some $i = 1, \dots, m$, $G_i(\boldsymbol{r}, \boldsymbol{s}) > 0$, then $\mathcal{M} \cap \mathcal{D} = \emptyset$ and $\mathcal{M}' = \emptyset$. Moreover, if $F(\boldsymbol{s}, \boldsymbol{r}) \le \gamma_k$, then $\{\boldsymbol{x} \in \mathcal{M} \mid F(\boldsymbol{x}, \boldsymbol{x}) > \gamma_k\} = \emptyset$ and $\mathcal{M}' = \emptyset$ satisfies (6.16). This establishes the second case in (6.21).

For the other case, we extend [128, Lem. 11.1]. Let $\mathcal{D}' = \{\boldsymbol{x} \in \mathcal{D} \mid F(\boldsymbol{x}, \boldsymbol{x}) > \gamma_k\} \subseteq \mathcal{D}$. Then, condition (6.16) is equivalent to showing that $\mathcal{M} \cap \mathcal{D}' = \mathcal{M}' \cap \mathcal{D}'$ holds. Consider the left equation in (6.22) and (6.23). Since $r'_i = \alpha_i r_i + (1 - \alpha_i) s_i$ for some $\alpha_i \in [0, 1]$, we have $r_i \le r'_i \le s_i$. Thus, $\mathcal{M}' \subseteq \mathcal{M}$ and $\mathcal{M} \cap \mathcal{D}' \supseteq \mathcal{M}' \cap \mathcal{D}'$. For the reverse inclusion, first observe that, for every $\boldsymbol{x} \in \mathcal{M}$ and because of (6.2) and (6.3), $G_i(\boldsymbol{x}, \boldsymbol{x}) \le 0$ implies $G_i(\boldsymbol{r}, \boldsymbol{x}) \le 0$ and $F(\boldsymbol{x}, \boldsymbol{x}) > \gamma_k$ implies $F(\boldsymbol{x}, \boldsymbol{r}) > \gamma_k$. Hence, $\mathcal{M} \cap \mathcal{D}' \subseteq \mathcal{M} \cap \bar{\mathcal{D}}$ with

$$\mathcal{M} \cap \bar{\mathcal{D}} = \{\boldsymbol{x} \in \mathcal{M} \mid F(\boldsymbol{x}, \boldsymbol{r}) > \gamma_k, G_i(\boldsymbol{r}, \boldsymbol{x}) \le 0 \text{ for all } i = 1, \dots, m\}.$$

Now, any $\boldsymbol{x} \in \mathcal{M} \cap \bar{\mathcal{D}}$ is equal to $\boldsymbol{s} - \sum_{k=1}^{K} (s_i - x_i)\boldsymbol{e}_i$. For any such $\boldsymbol{x}$, the box $[\boldsymbol{x}, \boldsymbol{s}] \subseteq \mathcal{M} \cap \bar{\mathcal{D}}$ because if $\boldsymbol{x}' \ge \boldsymbol{x}$, then $F(\boldsymbol{x}', \boldsymbol{r}) \ge F(\boldsymbol{x}, \boldsymbol{r}) > \gamma_k$ and $G_i(\boldsymbol{r}, \boldsymbol{x}') \le G_i(\boldsymbol{r}, \boldsymbol{x}) \le 0$. Thus, also $\boldsymbol{x}^i = \boldsymbol{s} - (s_i - x_i)\boldsymbol{e}_i \in \mathcal{M} \cap \bar{\mathcal{D}}$. Because $x_i \ge r_i$, we have $\boldsymbol{x}^i = \boldsymbol{s} - \alpha(s_i - r_i)\boldsymbol{e}_i$ for some $\alpha \in [0, 1]$. Since $\boldsymbol{x}^i \in \bar{\mathcal{D}}$, this and (6.23) implies $\alpha \le \alpha_i$. Hence, $\boldsymbol{x}^i \ge \boldsymbol{r}'$ and $\boldsymbol{x} \ge \boldsymbol{r}'$. Thus, every $\boldsymbol{x} \in \mathcal{M} \cap \mathcal{D}'$ is also in $\mathcal{M}'$ and, $\mathcal{M} \cap \mathcal{D}' \subseteq \mathcal{M}' \cap \mathcal{D}'$. The second reduction step for $\boldsymbol{s}'$ in (6.22) and (6.24) can be proved analogously. □

Equations (6.23) and (6.24) can be implemented efficiently by a low precision bisection. It is important though that the obtained solutions are greater (or equal) than the true α_i, β_i. Otherwise, feasible solutions might be lost.

6.5 Benchmarks

In this section, we evaluate the performance of MMP against the state-of-the-art. Consider the K-user GIC from Section 2.1 where all receivers treat interference as noise. The achievable rate region for Gaussian codebooks is stated in (2.6). In particular, user k can transmit at most with rate

$$r_k = \mathrm{C}\left(\frac{|h_{kk}|^2 p_k}{\sigma_k^2 + \sum_{j \neq k} |h_{kj}|^2 p_j} \right). \tag{6.25}$$

In Example 6.7, we have established the MMP representation of (6.25) as

$$R_k(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{C}\left(\frac{|h_{kk}|^2 x_k}{\sigma_k^2 + \sum_{j \neq k} |h_{kj}|^2 y_j}\right). \tag{6.26}$$

We first continue Example 6.7 and consider the WSR. Afterwards, we compute the GEE and compare the performance of MMP against fractional monotonic programming.

6.5.1 Weighted Sum Rate Maximization

The WSR maximization problem with minimum rate constraints is

$$\max_{\boldsymbol{p} \in [\boldsymbol{0}, \bar{\boldsymbol{P}}]} \sum_{k=1}^{K} w_k r_k \quad \text{s. t.} \quad r_k \geq R_{\min,k}, \ k = 1, \dots, K. \tag{6.27}$$

This is an MMP problem since the objective has the MMP representation

$$F(\boldsymbol{x}, \boldsymbol{y}) = \sum_{k=1}^{K} w_k R_k(\boldsymbol{x}, \boldsymbol{y})$$

due to (6.5), (6.7) and (6.26). The feasible set is as in (6.17) with

$$G_k(\boldsymbol{x}, \boldsymbol{y}) = R_{\min,k} - R_k(\boldsymbol{y}, \boldsymbol{x}), \, k = 1, \dots, K.$$

As already observed in Example 6.7, Problem (6.27) is also a monotonic optimization problem. It can be solved either by the polyblock algorithm which requires the transformation into canonical form (cf. Sections 3.3, 4.1 and 4.2), or by the BB approach in Section 4.2.1 which is equivalent to using the MMP representation (6.13). Both methods are evaluated and labeled as "PA" and "BRB DM", respectively, in the figures below.

Other approaches from the literature rely mostly on the polyblock algorithm. In the MAPEL framework [98, 154], Problem (6.27) is parametrized in terms of the signal to interference plus noise ratio (SINR) as

$$\max_{\boldsymbol{\gamma} \in \mathcal{G}} \sum_{k=1}^{K} w_k \, \mathrm{C}(\gamma_k) \quad \text{s. t.} \quad \gamma_k \geq \gamma_{\min,k}, \ k = 1, \dots, K$$

where the set of possible SINR combinations $\mathcal{G}$ is approximated from the outside by means of the polyblock algorithm (cf. Section 3.3) until the globally optimal solution is found. Since the objective is an increasing function of $\boldsymbol{\gamma}$, this method has the theoretical advantage of not requiring any auxiliary variables that are necessary if optimizing directly over the powers. However, the feasibility check $\boldsymbol{\gamma} \in \mathcal{G}$ requires the solution of a linear system.

Instead, the authors of [71] formulate Problem (6.27) as

$$\max_{\boldsymbol{\rho} \in \mathcal{R}} \sum_{k=1}^{K} w_k \rho_k \quad \text{s.t.} \quad \rho_k \geq R_{\min,k},\, k = 1, \dots, K$$

where $\mathcal{R}$ is the achievable rate region defined in (2.6) (or by (6.25)). This rate region is then approximated by the polyblock method. This can be considered as a special case of the framework in [12] and will be labeled as “Ratespace PA” in the numerical results below. Similarly to the MAPEL framework, this method requires the solution of a linear system in every iteration to check $\boldsymbol{\rho} \in \mathcal{R}$ and project points from outside the feasible set onto its boundary.

The average convergence speed of BB methods depends strongly on the quality of the bounds, i.e., tighter bounds lead, in general, to faster convergence. Recall that the bounds in the MMP framework are computed from the MMP representation of the objective, i.e., to bound the objective over a box $[\boldsymbol{r}, \boldsymbol{s}]$, $F(\boldsymbol{s}, \boldsymbol{r})$ is computed. To show that (6.26) provides tighter bounds than the DI approach, consider the difference of a single rate term in the MMP representations (6.13) and (6.26), i.e.,

$$\begin{aligned}
\Delta F_k(\boldsymbol{x}, \boldsymbol{y}) &= R_k(\boldsymbol{x}, \boldsymbol{y}) - R_{k,\mathrm{DM}}(\boldsymbol{x}, \boldsymbol{y}) \\
&= \mathrm{C}\left(\frac{|h_{kk}|^2 x_k}{\sigma_k^2 + \sum_{j \neq k} |h_{kj}|^2 y_j}\right) - \log\left(|h_{kk}|^2 x_k + \sum_{j \neq k} |h_{kj}|^2 x_j + \sigma_k^2\right) \\
&\qquad + \log\left(\sum_{j \neq k} |h_{kj}|^2 y_j + \sigma_k^2\right) \\
&= \log\left(\frac{|h_{kk}|^2 x_k + \sigma_k^2 + \sum_{j \neq k} |h_{kj}|^2 y_j}{|h_{kk}|^2 x_k + \sigma_k^2 + \sum_{j \neq k} |h_{kj}|^2 x_j}\right) - \log(1)
\end{aligned}$$

Thus, the difference between the bounds obtained by (6.13) and (6.26) when bounding the objective over the box $[\boldsymbol{r}, \boldsymbol{s}]$ is

$$\Delta F_k(\boldsymbol{s}, \boldsymbol{r}) = \log\left(\frac{|h_{kk}|^2 s_k + \sigma_k^2 + \sum_{j \neq k} |h_{kj}|^2 r_j}{|h_{kk}|^2 s_k + \sigma_k^2 + \sum_{j \neq k} |h_{kj}|^2 s_j}\right) \leq 0$$

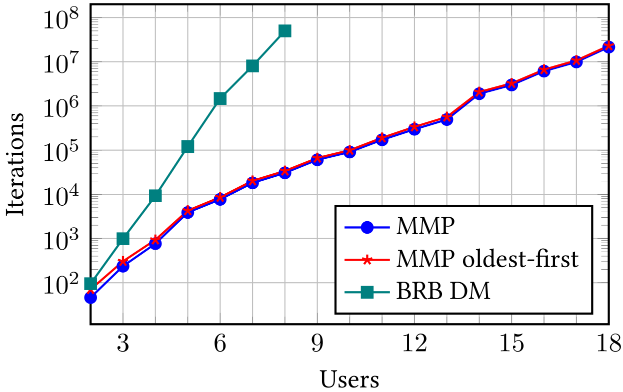


Figure 6.1: Average number of iterations required by Algorithm 6 to solve (6.27) with bounds obtained by 1) (6.26) and best-first selection; 2) (6.26) and oldest-first selection; and 3) (6.13) and best-first selection. Results are averaged over 100 different i.i.d. channel realizations.

where the inequality holds because $\boldsymbol{r} \leq \boldsymbol{s}$.

The results in Fig. 6.1 illustrate this theoretical intuition numerically. It displays the average number of iterations required by Algorithm 6 to solve (6.27) versus the number of variables. Each data point is averaged over 100 i.i.d. channel realizations with $h_{ij} \sim \mathcal{CN}(0, 1)$ for all $i, j = 1, \dots, K$. Further, $\eta = 0.01$, $\sigma^2 = 0.01$, $\bar{P}_k = 1$, $w_k = 1$, $R_{\min,k} = 0$ for all k, and no reduction is used. It can be observed that, for $K = 8$ variables, MMP with best-first selection and MMP function (6.26), labeled as "MMP", requires three orders of magnitude less iterations to solve (6.27) than the same algorithm with MMP function obtained from (6.13), named "BRB DM". From a practical perspective, this means that the MMP framework is able to solve (6.27) with 18 variables in the same time that state-of-the-art monotonic programming requires to solve problems with 8 variables. A further observation from Fig. 6.1 is that the oldest-first selection rule ("MMP oldest-first") requires only slightly more iterations to converge than the best-first rule. The benefits of oldest-first selection will be further evaluated below.

A comparison based on iterations works well for algorithms with similar computational complexity per iteration. However, when evaluating algorithms as different as the polyblock and BB algorithms, comparing the number of iterations is meaningless: the polyblock algorithm typically requires much

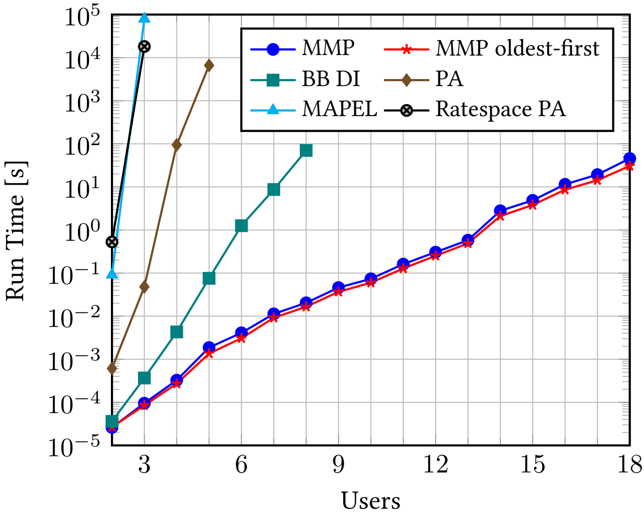


Figure 6.2: Average run time required to solve (6.27) with different algorithms. Results are averaged over 100 different i.i.d. channel realizations.

fewer iterations but each iteration takes much longer than in a BB algorithm. Thus, we resort to comparing the average run time of the algorithms. This and the memory consumption of all discussed approaches is displayed in Figs. 6.2 and 6.3, respectively. The same parameters as in the computation of Fig. 6.1 were used.

First, observe that all algorithms scale both in run time and memory consumption exponentially with the number of variables. Since Problem (6.27) is NP-hard [73, Thm. 1], better asymptotic complexity is not achievable. However, it is obvious that the computational complexity still may have very different slope and some algorithms are significantly more efficient than others. Specifically, the proposed MMP framework solves Problem (6.27) in considerably less time and memory requirements than all other state-of-the-art methods. The polyblock based methods all consume more memory than the BB based algorithms starting from three optimization variables. In terms of run time they are already outperformed by at least 1.5 orders of magnitude for 2 variables and soon reach the run time limit of 8 h. For the BB methods, the observations from Fig. 6.1 continue to hold for Fig. 6.2. It can be observed from the memory consumption that good bounds are not only critical for fast convergence but also for memory

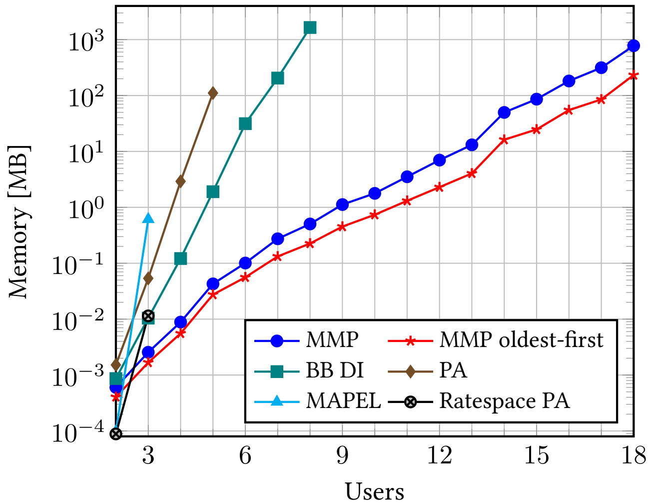


Figure 6.3: Average memory consumption of different algorithms to solve (6.27). Results are averaged over 100 different i.i.d. channel realizations.

efficiency. In this example, the MMP algorithm was able to solve problems more than twice the size of the DI BB algorithm within a memory limit of 2.5 GB.

Finally, observe that the best-first approach consumes considerably more memory than the oldest-first rule, e.g., 3.4 times or 546 MB more at 18 variables. Further, observe that while requiring fewer iterations the best-first rule has longer run times than the oldest-first rule, e.g., 1.5 times or 14.9 s at 18 variables. This can be explained from Fig. 6.3 since the memory consumption is directly proportional to the number of boxes in $\mathscr{R}_k$. Recall from Section 3.1.2 that the best-first rule is equivalent to a priority queue. While accessing the top-element in a priority queue has complexity $O(1)$, insertion has worst-case complexity $O(\log n)$, where n is the number of elements in the data structure. Compared to the other operations during each iteration of the algorithm, which have polynomial complexity in the number of variables, $O(\log n)$ is extremely small except when the size of the queue is very large. Instead, the implementation of the oldest-first rule is a queue with with constant time insertion, deletion, and top access complexity.

6.5.2 Energy Efficiency Maximization

Recall from Section 2.2.3 that the GEE is a key performance metric for 5G networks and beyond measuring the network EE. It is defined as the benefit-cost ratio of the total network throughput in a time interval T and the energy necessary to operate the network during this time, i.e.,

$$\mathrm{GEE} = \frac{TW\sum_{k=1}^{K} r_k}{T\left(\boldsymbol{\phi}^T\boldsymbol{p} + P_c\right)} = \frac{W\sum_{k=1}^{K} r_k}{\boldsymbol{\phi}^T\boldsymbol{p} + P_c} \quad \left[\frac{\text{bit}}{\text{J}}\right], \tag{6.28}$$

where r_k is the achievable rate of link k, W is the bandwidth, $\boldsymbol{\phi} \geq 1$ are the inverses of the power amplifier efficiencies and P_c is a constant modelling the static part of the circuit power consumption.

Maximizing the GEE for interference networks with TIN, i.e., where r_k is as in (6.25), results in the nonconvex fractional programming problem

$$\max_{\boldsymbol{p}\in[\boldsymbol{0},\bar{\boldsymbol{P}}]} \quad \frac{\sum_{k=1}^{K} r_k}{\boldsymbol{\phi}^T\boldsymbol{p} + P_c}, \tag{6.29}$$

where we have omitted the inessential constant W and minimum rate constraints that are already discussed in Section 6.5.1.

The state-of-the-art approach to solve (6.29) is fractional monotonic programming introduced in Section 4.3 where Dinkelbach's algorithm is combined with monotonic programming to solve a sequence of auxiliary problems

$$\max_{\boldsymbol{p}\in[\boldsymbol{0},\bar{\boldsymbol{P}}]} \quad \sum_{k=1}^{K} r_k - \lambda\left(\boldsymbol{\phi}^T\boldsymbol{p} + P_c\right), \tag{6.30}$$

with nonnegative parameter λ. While most works use the polyblock algorithm to solve (6.30) (e.g., [83, 147]), we have already established above that the BB procedure with DI bounds from Section 4.2.1 outperforms the classical polyblock algorithm [125] in terms of computational complexity.

Instead, the MMP framework allows to solve (6.29) without the need of Dinkelbach's algorithm. A MMP representation of (6.28) can be obtained similar to Example 6.6. Specifically, with (6.26) and the identities in (6.5), (6.7) and (6.8), we obtain

$$F(\boldsymbol{x}, \boldsymbol{y}) = \frac{W\sum_{k=1}^{K} R_k(\boldsymbol{x}, \boldsymbol{y})}{\boldsymbol{\phi}^T\boldsymbol{y} + P_c}$$

with $R_k(\boldsymbol{x}, \boldsymbol{y})$ as in (6.26).

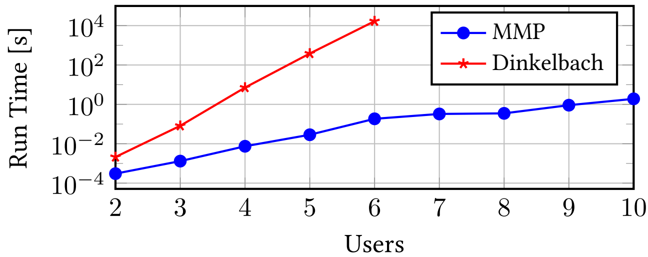


Figure 6.4: Average run time to solve (6.29) with MMP and Dinkelbach's algorithm where the inner problem is solved by Algorithm 6 with DI identification as in (6.13). Results are averaged over 100 different i.i.d. channel realizations.

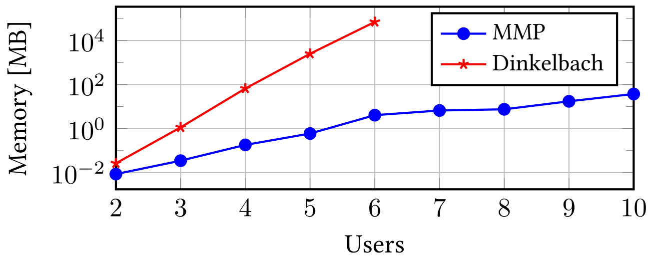


Figure 6.5: Average memory consumption during the solution of (6.29) with MMP and Dinkelbach's algorithm where the inner problem is solved by Algorithm 6 with DI identification as in (6.13). Results are averaged over 100 different i.i.d. channel realizations.

The run time performance of both algorithms is evaluated in Fig. 6.4 where $\phi_k = 5$, for all k, and $P_c = 1$. The remaining parameters were chosen as in Section 6.5.1. It can be observed that MMP requires significantly less time to solve (6.29) than the legacy approach employing Dinkelbach's algorithm. For example, with $K = 6$ variables, MMP is, on average, almost five orders of magnitude faster than fractional monotonic programming. The memory consumption in Fig. 6.5 scales almost identically to the run time with MMP using four orders of magnitude less memory for six variables. Besides showing much better performance, the MMP algorithm obtains an η-optimal solution. Instead, Dinkelbach's algorithm does not provide any strict guarantees on the solution quality as already discussed in Section 4.3.

Part III

Applications

Chapter 7

Interference Channel

We have seen in Chapter 2 that GICs are a suitable model for many modern communication systems. Despite four decades of research, the complete characterization of the capacity region, i.e., the fundamental limits of reliable communication, remains an open problem even for channels with only two transmitters. Focusing on the sum rate (or throughput) of channels with two transmitter-receiver pairs, the situation is much more satisfying with sum capacity results being available for most interference regimes. The goal of this chapter is to evaluate the absolute limits of energy-efficient communication over two user ICs in terms of the GEE, the most important resource allocation metric to bring down the total energy consumption in communication networks. We will see in Section 7.2 that the sum capacity plays a prominent role in maximizing the GEE.

7.1 System Model & Capacity Results

Consider the 2-user GIC in standard form [14] that is illustrated Fig. 7.1 and defined by

$$y_1 = x_1 + a_2 x_2 + z_1, \qquad\qquad y_2 = a_1 x_1 + x_2 + z_2, \tag{7.1}$$

where, for $i = 1, 2$, a_i is the complex-valued channel crosstalk coefficient, z_i is i.i.d. zero-mean circularly symmetric complex Gaussian noise with unit power, and the transmitted signal x_i is subject to an average power constraint P_i. Comparing (7.1) to (2.1), we can identify

$$a_1 = \frac{h_{21}\sigma_1}{h_{11}\sigma_2} \qquad\qquad a_2 = \frac{h_{12}\sigma_2}{h_{22}\sigma_1} \qquad\qquad P_i = \frac{|h_{ii}|^2}{\sigma_i^2}\bar{P}_i,$$

where the variables are as defined in Section 2.1.

7.1.1 Point-to-Point Codes

The capacity region of this channel when constrained to capacity-achieving PTP codes is, according to Proposition 2.1, the union of the achievable rate regions

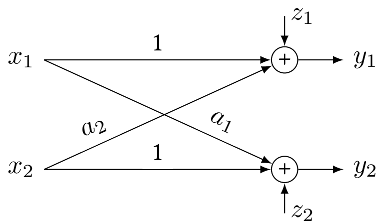


Figure 7.1: Illustration of a 2-user Gaussian interference channel in standard form.

with proper Gaussian codebooks where each receiver uses either JD or TIN. Then, the PTP capacity region of the GIC is

$$\mathcal{C}_{\text{ptp}} = \bigcup_{\boldsymbol{d} \in \{\text{TIN,JD}\}^2} \mathcal{R}_{1,d_1}(\boldsymbol{P}) \cap \mathcal{R}_{2,d_2}(\boldsymbol{P}). \tag{7.2}$$

with

$$\mathcal{R}_{1,\text{TIN}}(\boldsymbol{p}) = \left\{ \boldsymbol{R} \,\middle|\, 0 \le R_1 \le \mathrm{C}\left(\frac{p_1}{1 + |a_2|^2 p_2} \right) \right\} \tag{7.3}$$

and

$$\mathcal{R}_{1,\text{JD}}(\boldsymbol{p}) = \left\{ \boldsymbol{R} \,\middle|\, \begin{array}{l} 0 \le R_1 \le \mathrm{C}(p_1)\,, \\ R_1 + R_2 \le \mathrm{C}\Big(p_1 + |a_2|^2 p_2\Big) \end{array} \right\}.$$

This is also the capacity region under the constraint of random codebook ensembles and coded time sharing (cf. Section 2.1).

Unconstrained capacity results depend mainly on the channel conditions. Most notably, if $|a_1|^2 \ge 1$ and $|a_2|^2 \ge 1$, the channel is said to have *strong interference* and the capacity region is

$$\bigcap_{i=1}^{2} \mathcal{R}_{i,\text{JD}} = \left\{ \boldsymbol{R} \,\middle|\, \begin{array}{c} 0 \le R_i \le \mathrm{C}(p_i), \quad i = 1, 2 \\ R_1 + R_2 \le \min\{\mathrm{C}(p_1 + |a_2|^2 p_2),\ \mathrm{C}(|a_1|^2 p_1 + p_2)\} \end{array} \right\} \tag{7.4}$$

which is achieved by proper Gaussian codewords and jointly decoding x_1 and x_2 at each receiver [104, 111].[29] The sum capacity, i.e., the maximal sum rate

[29] The references [104] and [3, 85, 112] to be cited below consider only real-valued channels, while [111] generalizes these results to complex-valued channels.

$R_1 + R_2$ at which reliable communication is possible, is readily obtained from (7.4) as

$$C_\Sigma = \min\{\mathrm{C}(P_1) + \mathrm{C}(P_2), \mathrm{C}(P_1 + |a_2|^2 P_2), \mathrm{C}(|a_1|^2 P_1 + P_2)\}.$$

Apart from the strong interference case, only the sum capacity is known. The channel has *mixed interference* if only one receiver observes strong interference, i.e., $|a_1|^2 \geq 1$ and $0 < |a_2|^2 \leq 1$, or $0 < |a_1|^2 \leq 1$ and $|a_2|^2 \geq 1$. The sum capacity for the first case, i.e., $|a_1|^2 \geq 1$, is

$$C_\Sigma = \min\left\{\mathrm{C}(|a_1|^2 P_1 + P_2), \mathrm{C}(P_2) + \mathrm{C}\left(\frac{P_1}{1 + |a_2|^2 P_2}\right)\right\}. \tag{7.5}$$

It is achieved by proper Gaussian codewords, jointly decoding x_1 and x_2 at receiver 2, and treating x_2 as noise at receiver 1. Likewise, the second case is obtained by exchanging the indices in (7.5) [85, 111]. For $|a_1|^2 < 1$ and $|a_2|^2 < 1$, the channel has *weak interference*. This regime is subdivided by the condition

$$|a_1|\,(1 + |a_2|^2 P_2) + |a_2|\,(1 + |a_1|^2 P_1) \leq 1.$$

If it holds, the interference is *noisy* and the sum capacity is

$$C_\Sigma = \mathrm{C}\left(\frac{P_1}{1 + |a_2|^2 P_2}\right) + \mathrm{C}\left(\frac{P_2}{1 + |a_1|^2 P_1}\right), \tag{7.6}$$

which is achievable by proper Gaussian inputs and TIN [3, 85, 111, 112]. Otherwise, the interference is said to be *moderate* and no capacity results are available. Thus, from a throughput perspective, the moderate interference regime is the only one in which more complicated codes might improve upon PTP codes. Figure 7.2 visualizes the different interference regimes and summarizes the quality of the available capacity results.

7.1.2 Rate Splitting

Recently, RS has been shown to increase spectral and energy efficiency in practical wireless networks [20, 143]. The concept is much older though, being one main ingredient of the famous HK coding scheme [19, 39] for the IC. Its advantage over simpler PTP codes is that decoders are not forced to decide between completely treating an interfering message as noise or decoding it. Instead, HK

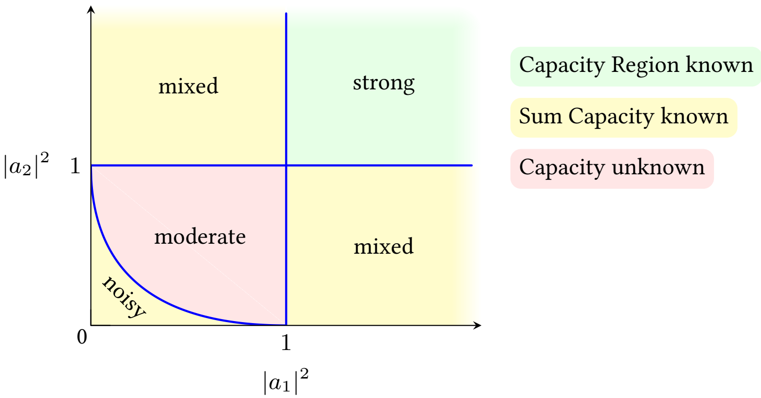


Figure 7.2: Interference regimes and capacity results for the 2-user GIC (adapted from [110]).

coding, and especially the RS part, softly bridges the gap between both interference mitigation approaches by allowing to decode interfering messages partially and treat the remaining part as noise. It leads to the largest known achievable rate region for the IC and is within one bit per user of the GIC's capacity region [30]. It also achieves the (sum) capacity in all cases where it is established. Thus, the only interference regime where HK coding might provide a throughput gain over PTP codes is moderate interference, an important scenario for multicell wireless networks.

The HK codebook of each transmitter is generated by first splitting its message into a common and private part. These are then combined into a single codeword by superposition coding [23], [29, §5.3] where the common message is the cloud center and the private message is the satellite codeword. The common message is decoded by both receivers, while the private is only decoded by the intended receiver and considered as noise by the other. An error occurs at each receiver if either its common or private message is decoded incorrectly, while the common message of the other receiver needs not be decoded correctly (this is known as "simultaneous nonunique decoding"). Together with coded time sharing, the HK achievable rate region is obtained as [19, 29, 39] the set of all nonnegative rate tuples (R_1, R_2) satisfying

$$\begin{aligned} R_i &\leq I(X_i; Y_i | U_j, Q) \\ R_i + R_j &\leq I(X_i, U_j; Y_i | Q) + I(X_j; Y_j | U_1, U_2, Q) \end{aligned}$$

$$2R_i + R_j \leq I(X_i, U_j; Y_i|Q) + I(X_i; Y_i|U1, U2, Q) + I(X_j, U_i; Y_j|U_j, Q)$$
$$R_1 + R_2 \leq I(X_1, U_2; Y_1|U_1, Q) + I(X_2, U_1; Y_2|U_2, Q)$$

for all $i, j = 1, 2$, $j \neq i$, and some probability mass function (pmf) $p(q)$ $p(u_1, x_1|q)$ $p(u_2, x_2|q)$, where $|\mathcal{U}_i| \leq |\mathcal{X}_i| + 4$ and $|\mathcal{Q}| \leq 6$. It is established in [5, 135] that this is the largest rate region achievable when constrained to random code ensembles, RS, superposition coding, and coded time sharing. Optimizing this region is hard due to its complicated structure, and the optimal input distribution remains an open problem to date.

Indeed, even for TIN and PTP codes, the optimal codebook is still subject to ongoing research. This is due to the competitive nature of the problem discussed in [1] and Section 2.1. It appears that mixed inputs, consisting of the superposition of a Gaussian and a discrete random variable, are a favorable choice for TIN [27, 28], and, hence, might also be a good choice for the private message in the HK scheme. Surprisingly, TIN with such a codebook shows a resemblance of HK coding in the sense that, even though interference is not decoded, the discrete part behaves as a common message and can be removed from the received signal without rate loss. The downside is that mixed Gaussian inputs complicate analysis significantly and are not (yet) widely used.

However, proper Gaussian codebooks without coded time sharing are common assumptions and achieve the sum capacity of the GIC in all cases with established sum capacity. Evaluating the general HK region above under these assumptions for the channel model in (7.1) results in an achievable rate region consisting of all nonnegative (R_1, R_2) such that

$$R_i \leq \mathrm{C}\left(\frac{p_i + \bar{p}_i}{1 + |a_2|^j \bar{p}_j}\right)$$
$$R_i + R_j \leq \mathrm{C}\left(\frac{p_i + \bar{p}_i + a_j^2 p_j}{1 + a_j^2 \bar{p}_j}\right) + \mathrm{C}\left(\frac{\bar{p}_j}{1 + a_i^2 \bar{p}_i}\right)$$
$$2R_i + R_j \leq \mathrm{C}\left(\frac{p_i + \bar{p}_i + a_j^2 p_j}{1 + a_j^2 \bar{p}_j}\right) + \mathrm{C}\left(\frac{\bar{p}_i}{1 + a_j^2 \bar{p}_j}\right) + \mathrm{C}\left(\frac{a_i^2 p_i + \bar{p}_j}{1 + a_i^2 \bar{p}_i}\right)$$
$$R_1 + R_2 \leq \mathrm{C}\left(\frac{\bar{p}_1 + |a_2|^2 p_2}{1 + |a_2|^2 \bar{p}_2}\right) + \mathrm{C}\left(\frac{\bar{p}_2 + |a_1|^2 p_1}{1 + |a_1|^2 \bar{p}_1}\right)$$

for all $i, j = 1, 2$, $j \neq i$ and with $p_i + \bar{p}_i \leq P_i$, $i = 1, 2$. We denote this region as $\mathcal{R}_{\mathrm{HK}}(\boldsymbol{p})$. The sum rate of this rate region, i.e., $R_\Sigma(\boldsymbol{p}) = \max\{R_1 + R_2 \mid \boldsymbol{R} \in$

$\mathcal{R}_{\mathrm{HK}}(\boldsymbol{p})\}$, is [123]

$$\begin{aligned}
&\mathrm{C}\left(\frac{\tilde{p}_1}{1+|a_2|^2\,\tilde{p}_2}\right)+\mathrm{C}\left(\frac{\tilde{p}_2}{1+|a_1|^2\,\tilde{p}_1}\right)+\\
&\quad\min\Bigg\{\mathrm{C}\left(\frac{(p_1-\tilde{p}_1)+|a_2|^2\,(p_2-\tilde{p}_2)}{1+\tilde{p}_1+|a_2|^2\,\tilde{p}_2}\right),\ \mathrm{C}\left(\frac{|a_1|^2\,(p_1-\tilde{p}_1)+(p_2-\tilde{p}_2)}{1+|a_1|^2\,\tilde{p}_1+\tilde{p}_2}\right),\\
&\qquad\qquad\mathrm{C}\left(\frac{|a_1|^2\,(p_1-\tilde{p}_1)}{1+|a_1|^2\,\tilde{p}_1+\tilde{p}_2}\right)+\mathrm{C}\left(\frac{|a_2|^2\,(p_2-\tilde{p}_2)}{1+\tilde{p}_1+|a_2|^2\,\tilde{p}_2}\right),\\
&\qquad\qquad\qquad\mathrm{C}\left(\frac{p_1-\tilde{p}_1}{1+\tilde{p}_1+|a_2|^2\,\tilde{p}_2}\right)+\mathrm{C}\left(\frac{p_2-\tilde{p}_2}{1+|a_1|^2\,\tilde{p}_1+\tilde{p}_2}\right)\Bigg\},
\end{aligned}\tag{7.7}$$

where $p_i \in [0, P_i]$ is the total transmit power of user i and $\tilde{p}_i \in [0, p_i]$ is the power allocated to the private message. From a throughput perspective, allocating all available transmit power, i.e., $p_i = P_i$, is optimal. This is not necessarily true for the GEE.

7.2 Global Energy Efficiency

Recall from Section 2.2.3 that the GEE of a communication network is defined as the benefit-cost ratio of the total network throughput and the total power necessary to operate the network, i.e., for a network with two transmitters,

$$\mathrm{GEE} = W\frac{R_1+R_2}{\boldsymbol{\phi}^T\boldsymbol{p}+P_c},$$

where W is the bandwidth, $\boldsymbol{\phi}$ are the power amplifier inefficiencies, and P_c is the total static power consumption of the network. The goal of this chapter is to maximize the GEE, i.e., solve the problem

$$\max_{\boldsymbol{p},\boldsymbol{R}}\ \frac{R_1+R_2}{\boldsymbol{\phi}^T\boldsymbol{p}+P_c}\quad\text{s.t.}\quad \boldsymbol{R}\in\mathcal{R}(\boldsymbol{p}),\ \boldsymbol{p}\in\mathcal{P}\tag{7.8}$$

where $\mathcal{R}(\boldsymbol{p})$ is the achievable rate region of the chosen coding scheme, and $\mathcal{P}$ are the power constraints. For PTP codes, these are $\mathcal{P} = [\mathbf{0}, \boldsymbol{P}]$, while for HK coding $\mathcal{P} = \{(p_1, p_2, \tilde{p}_1, \tilde{p}_2) \in \mathbb{R}^4_{\geq 0} : p_i \leq P_i, \tilde{p}_i \leq p_i, \text{for } i = 1, 2\}$ and $\boldsymbol{\phi} = (\phi_1, \phi_2, 0, 0)^T$.

Problem (7.8) is a global optimization problem that poses two challenges, the fractional objective and the nonconcave RHSs of the rate expressions characterizing $\mathcal{R}(\boldsymbol{p})$ when interfering messages are treated as noise. For example, the

RHS of (7.3) is easily verified to be nonconcave. However, with the sum capacity results from Section 7.1 we can identify (7.8) as an MMP problem.

Observe that any optimization problem can be decomposed into an outer and an inner problem by first optimizing over some variables and then over the others [11, p. 133]. Thus, Problem (7.8) is equivalent to

$$\max_{\boldsymbol{p}\in\mathcal{P}} \max_{\boldsymbol{R}\in\mathcal{R}(\boldsymbol{p})} \frac{R_1 + R_2}{\boldsymbol{\phi}^T\boldsymbol{p} + P_c} = \max_{\boldsymbol{p}\in\mathcal{P}} \frac{\max_{\boldsymbol{R}\in\mathcal{R}(\boldsymbol{p})} R_1 + R_2}{\boldsymbol{\phi}^T\boldsymbol{p} + P_c} = \max_{\boldsymbol{p}\in\mathcal{P}} \frac{R_\Sigma(\boldsymbol{p})}{\boldsymbol{\phi}^T\boldsymbol{p} + P_c} \tag{7.9}$$

where the first equality is due the objective's denominator being independent of $\boldsymbol{R}$. The numerator of (7.9) is the sum rate which is limited above by the sum capacity C_Σ. Thus, for constant P_c, the maximum achievable GEE is the solution of (7.9) for $R_\Sigma = C_\Sigma$. In the next section, we discuss the solution of the rightmost form of (7.9) with the MMP framework from Chapter 6.

Recall from Section 7.1 that the sum capacity is known for all interference regimes except moderate interference. In all cases, it is achievable by PTP codes, which require less complex decoding than HK codes. More complex decoding also requires more energy and, thus, P_c is likely to increase for HK coding. Thus, in cases where both PTP codes and HK coding achieve the sum capacity, HK coding has a lower GEE than PTP codes. However, for moderate interference, where the sum capacity is unknown, HK coding might offer a benefit despite its increased decoding complexity. One of the aims of this chapter is to evaluate numerically whether such GEE gains due to HK coding can occur.

The capacity region of GICs constrained to capacity-achieving PTP codes is given in (7.2) as $\mathcal{C}_{\text{ptp}}$. Then, the maximum achievable GEE when constrained to capacity-achieving PTP codes is the solution of (7.8) with $\mathcal{R} = \mathcal{C}_{\text{ptp}}$. Because $\sup_{\boldsymbol{x}\in\bigcup_i\mathcal{D}_i} f(\boldsymbol{x}) = \max_i \sup_{\boldsymbol{x}\in\mathcal{D}_i} f(\boldsymbol{x})$ by Lemma 2.2, we can split (7.8) into four individual optimization problems, each equivalent to (7.8) with $\mathcal{R} = \mathcal{R}_{1,d_1} \cap R_{2,d_2}$ for all different combinations of $d_1, d_2 \in \{\text{TIN}, \text{JD}\}$. These rate regions are exactly the achievable rate regions of the sum capacity achieving schemes in the four other interference regimes, and, hence, the solution of $(7.8)|_{\mathcal{R}=\mathcal{C}_{\text{ptp}}}$ is the maximum of the achievable GEEs in the four other interference regimes.[30] After computing the absolute GEE limit of PTP codes using the algorithms proposed in the following, we can be sure that any gain observed by RS is definitely not achievable by these codes.

[30] These are strong, noisy, and two cases of mixed interference.

7.3 Optimal Resource Allocation

In the previous section we have shown that (7.8) is equivalent to

$$\max_{\boldsymbol{p}\in\mathcal{P}} \frac{R_\Sigma(\boldsymbol{p})}{\boldsymbol{\phi}^T\boldsymbol{p} + P_c}$$

All sum rate expressions in Section 7.1, except for the strong interference case, are nonconcave functions of $\boldsymbol{p}$. The strong interference case will be solved with Dinkelbach's algorithm in Section 7.3.1. For all other interference regimes, we apply MMP from Chapter 6. This requires an initial box $\mathcal{M}_0 \supseteq \mathcal{P}$, an MMP representation of the objective and a feasibility test. First, observe that for all coding schemes except HK coding, every box $\mathcal{M}$ generated by Algorithm 6 is $\mathcal{M} \subset \mathcal{P}$. Thus, $\mathcal{M} \cap \mathcal{P} = \mathcal{M} \neq \emptyset$ and obtaining a feasible point is trivial. Instead, for HK coding let the optimization variables be ordered as $\boldsymbol{x} = (\boldsymbol{p}, \tilde{\boldsymbol{p}})$. Then, for a box $\mathcal{M} = [\boldsymbol{r}, \boldsymbol{s}] \times [\tilde{\boldsymbol{r}}, \tilde{\boldsymbol{s}}]$, $\mathcal{M} \cap \mathcal{P} \neq \emptyset$ if $\boldsymbol{s} \geq \tilde{\boldsymbol{r}}$. In this case, the point $(\boldsymbol{s}, \tilde{\boldsymbol{r}})$ is always feasible.

For the MM representation of the objective, assume there exists an MMP representation of $R_\Sigma(\boldsymbol{p})$ and denote it as $F_\Sigma(\boldsymbol{x}, \boldsymbol{y})$. Then, similar to Example 6.6, an MMP representation of $f(\boldsymbol{p})$ is

$$F(\boldsymbol{x}, \boldsymbol{y}) = \frac{F_\Sigma(\boldsymbol{x}, \boldsymbol{y})}{\boldsymbol{\phi}^T\boldsymbol{y} + P_c}. \tag{7.10}$$

Thus, we require MMP representations of (7.5)–(7.7) to solve (7.9) with Algorithm 6. These are

$$\begin{aligned} F_\Sigma(\boldsymbol{x}, \boldsymbol{y}) = {} & \mathrm{C}\left(\frac{\tilde{x}_1}{1 + |a_2|^2 \tilde{y}_2}\right) + \mathrm{C}\left(\frac{\tilde{x}_2}{1 + |a_1|^2 \tilde{y}_1}\right) + \\ & \min\left\{ \mathrm{C}\left(\frac{(x_1 - \tilde{y}_1) + |a_2|^2 (x_2 - \tilde{y}_2)}{1 + \tilde{y}_1 + |a_2|^2 \tilde{y}_2}\right), \mathrm{C}\left(\frac{|a_1|^2 (x_1 - \tilde{y}_1) + (x_2 - \tilde{y}_2)}{1 + |a_1|^2 \tilde{y}_1 + \tilde{y}_2}\right), \right. \\ & \qquad \mathrm{C}\left(\frac{|a_1|^2 (x_1 - \tilde{y}_1)}{1 + |a_1|^2 \tilde{y}_1 + \tilde{y}_2}\right) + \mathrm{C}\left(\frac{|a_2|^2 (x_2 - \tilde{y}_2)}{1 + \tilde{y}_1 + |a_2|^2 \tilde{y}_2}\right), \\ & \qquad \left. \mathrm{C}\left(\frac{x_1 - \tilde{y}_1}{1 + \tilde{y}_1 + |a_2|^2 \tilde{y}_2}\right) + \mathrm{C}\left(\frac{x_2 - \tilde{y}_2}{1 + |a_1|^2 \tilde{y}_1 + \tilde{y}_2}\right) \right\} \end{aligned}$$

for HK coding in (7.7),

$$F_\Sigma(\boldsymbol{x}, \boldsymbol{y}) = \min\left\{ \mathrm{C}(|a_1|^2 x_1 + x_2), \mathrm{C}(x_2) + \mathrm{C}\left(\frac{x_1}{1 + |a_2|^2 y_2}\right) \right\}$$

for mixed interference in (7.5), and

$$F_\Sigma(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{C}\left(\frac{x_1}{1 + |a_2|^2 y_2}\right) + \mathrm{C}\left(\frac{x_2}{1 + |a_1|^2 y_1}\right)$$

for noisy interference in (7.6). In all cases, verifying the properties in Definition 6.1 and (6.4) is straightforward with the help of Proposition 6.3. Of course, these MMP representations can also be used directly for throughput maximization instead of combining them with (7.10) for GEE maximization.

7.3.1 Strong Interference

In contrast to the other interference regimes, the sum capacity $C_\Sigma(\boldsymbol{p})$ of the GIC with strong interference is a concave function and, thus, the fractional objective of (7.9) is pseudo-concave. This problem is solved efficiently by Dinkelbach's algorithm as a sequence of convex optimization problems. Specifically, we identify the auxiliary problem $\Lambda(\lambda)$ from Section 3.4 as

$$\begin{cases} \max\limits_{t, \boldsymbol{p} \in [\boldsymbol{0}, \boldsymbol{P}]} & t - \lambda\left(\boldsymbol{\phi}^T \boldsymbol{p} + P_c\right) \\ \text{s. t.} & t \leq \mathrm{C}(p_1) + \mathrm{C}(p_2), \quad t \leq \mathrm{C}(p_1 + |a_2|^2 p_2), \\ & t \leq \mathrm{C}(|a_1|^2 p_1 + p_2). \end{cases}$$

Since the objective is linear and the inequality constraints form a closed convex set, this is a convex optimization problem that can be solved by any state-of-the-art convex optimization solver, e.g., Mosek [84].

7.4 Numerical Evaluation

We consider a wireless interference network in which two single-antenna senders are placed randomly with uniform distribution in a 1 km$\times$2 km rectangular area and communicate with two single-antenna receivers placed at coordinates $(-0.5, 0)$ km and $(0.5, 0)$ km. The path-loss is modeled according to [13], with power decay factor 3.5 and carrier frequency 1.8 GHz, while small-scale fading effects are modeled as i.i.d. proper Gaussian random variates. The communication bandwidth is 180 kHz and the noise spectral density is −174 dBm/Hz. Each receiver has a 3 dB noise figure and power amplifier inefficiency $\phi = 4$. Static power consumption is $P_c = 1$ W and all senders have the same maximum transmit power $\bar{P}$.

Table 7.1: Empirical probability of interference regimes [75] ©2019 IEEE

	$\bar{P}$	Strong	Mixed	Weak	
				Moderate	Noisy
Arbitrary	0 dBm	34.8 %	31.5 %	11.0 %	22.7 %
	15 dBm			21.6 %	12.1 %
	30 dBm			32.3 %	1.4 %
Multicell	0 dBm	0.3 %	9.3 %	11.3 %	79.1 %
	15 dBm			44.1 %	46.3 %
	30 dBm			85.3 %	5.1 %

Two different scenarios are considered: One where the transmitters are dropped in the whole area and always associated with the same receiver, and one where each transmitter is placed such that it is geographically closer to its respective receiver. The latter resembles an uplink multicell scenario where the receivers represent BSs. Table 7.1 shows the empirical probability of the resulting channel being in a specific interference regime. It is noteworthy that, while with arbitrary user placement the distribution is approximately uniform,[31] the weak interference regime clearly dominates for the multicell scenario. Especially for realistic transmit powers, moderate interference clearly dominates. Thus, the only interference regime where RS might provide a benefit over PTP codes is definitely nonnegligible.

Numerical results for both user placements obtained by Algorithms 4 and 6 with tolerances $\eta = \delta = 0.01$ are displayed in Fig. 7.3. In addition, the GEEs for the individual interference regimes are shown. It can be observed that weak interference achieves the highest GEE, followed by mixed interference and strong interference. As could be expected from Table 7.1, the multicell scenario achieves a similar GEE as weak interference, and the arbitrary scenario is approximately in the middle of the three interference regimes. The slightly better performance of the multicell scenario compared to the weak interference regime is due to statistical effects from not completely overlapping sets of channel realizations.

The maximum GEE in the moderate interference regime is unknown. In all other regimes, it is achieved by PTP codes. To study whether this is also the case under moderate interference, we compare the maximum GEE achievable with

[31]From this point of view, the moderate and noisy interference regimes should be considered together since the differentiation between these two depends on the transmit power.

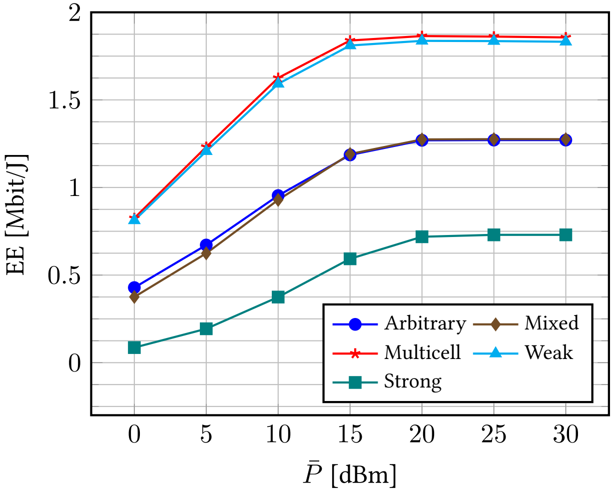


Figure 7.3: GEE of GIC with arbitrary and multicell user placement. Averaged over 1,000+ i.i.d. channel realizations [75]. ©2019 IEEE

PTP codes with the GEE of HK coding with proper Gaussian codebooks without coded time sharing. In 9 out of 5390 analyzed channel realizations that fell within the moderate interference regime, HK coding achieves a numerically significant gain when ignoring the different circuit power consumptions P_c. This implies that there can still be gains if P_c is modeled as being higher for HK than for PTP as long as the difference in P_c is not too large. The maximal observed gain over PTP codes is 0.11 bit/J/Hz (4.44 %) which amounts to 19.71 kbit/J. Having observed that HK coding can be beneficial in terms of EE, it should be studied in future research whether more general HK coding with coded time sharing and codebooks other than proper Gaussian can bring higher gains and/or gains in a larger number of channel realizations.

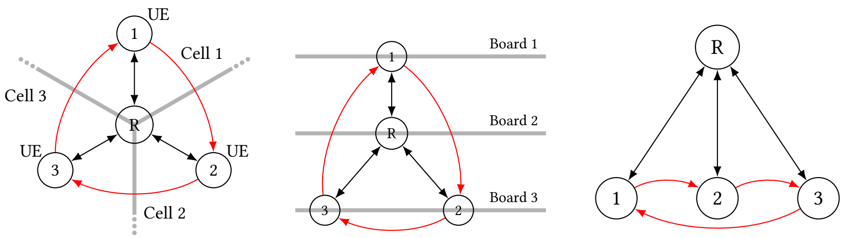
Chapter 8

Multi-Way Relay Channel

Relays are fundamental building blocks in wireless networks. Deploying relays in areas affected by significant shadowing such as tunnels or the inside of buildings, or in areas that are far away from the transmitter, allows one to extend cell-coverage and increase the network's reliability and throughput. Due to the high attenuation in mmWave communications, this is even more important in the era of 5G than it was in classical and current wireless systems. Thus, relaying is a key technology in emerging wireless technologies and its study is important to understand the fundamental limits of modern and future wireless communication systems [54].

An integral part to advance the understanding of relaying in networks is the MWRC. It models relay-aided communication across several nodes with no direct links between the users. Applications of this model are, e.g., heterogeneous dense small cell networks in modern and future wireless networks (cf. Fig. 8.1a), wireless board-to-board communication in highly adaptive computing [34] where multiple chips exchange data with the help of another chip acting as a relay (cf. Fig. 8.1b), wireless sensor networks, Industry 4.0 where several production stations pass messages along with the product, or communication of several ground stations over a satellite (cf. Fig. 8.1c).

The MWRC was first introduced in [37] as a model for clustered communication over a relay, where the terminals in each cluster exchange information among each other with the help of a relay. In each cluster, every terminal transmits a broadcast message to all other terminals in that cluster. Most subsequent works drop the multiple cluster assumption and consider a single cluster scenario with different message exchanges. In [94] the common-rate capacity of the additive white Gaussian noise MWRC with multiple broadcast messages is derived and it is shown that for three or more users this capacity is achieved by decode-and-forward (DF) for SNRs below 0 dB and compute-and-forward otherwise. The same authors present in [93] the capacity region of the finite field MWRC. In [15] a constant gap approximation of the capacity region of the Gaussian 3-user MWRC is derived. The authors consider the most general message exchange possible, where each terminal broadcasts a message to all other terminals and, in addition, transmits unicast messages to each individual



(a) Heterogeneous dense small-cell networks

(b) Wireless board-to-board communication

(c) Industry 4.0 & Satellite Communications

Figure 8.1: Illustration of different applications for MWRCs. Physical channels are drawn in black while message paths are displayed in red [79].

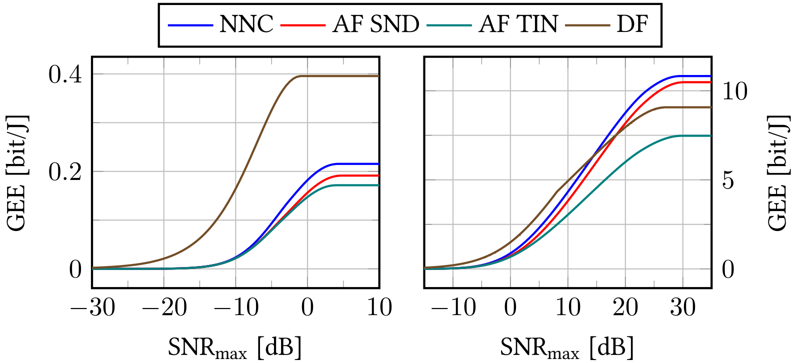
terminal. For a more complete discussion of achievability and capacity results please refer to [16].

In contrast to most other works, we focus on an MWRC with three users and multiple unicast transmissions such that

1. each user has a message to transmit which is intended for at least one other user.
2. each user desires at most one message.

From these two properties it follows immediately that each message is only required at one other user. One such message exchange is illustrated by the red arrows in Fig. 8.1. In a preliminary study in [83], the GEE of noisy network coding (NNC), AF, and DF relaying under the assumption of symmetric channels was evaluated. These relaying schemes impose very different hardware requirements on the relay and receivers and, thus, also have a different hardware energy consumption. For example, an AF relay does not decode any messages, so no digital signal processing (DSP) hardware is necessary, and, thus, no analog-to-digital converter (ADC) and digital-to-analog converter (DAC) is necessary either. In comparison, a DF relay needs all those components and has to decode all three messages, thus, having much higher circuit power consumptions.

Figure 8.2 displays the GEE of these systems for two different choices of the noise power. It illustrates that the energy-efficient performance of the different transmission and relaying scheme highly depends on the saturation point of the GEE function, which in turn is heavily affected by the system parameters. Thus, for a sensible comparison of the different schemes, it is necessary to



(a) 1 W noise power

(b) 0.1 mW noise power

Figure 8.2: GEE in the 3-user MWRC with 1) NNC, 2) AF-SND, 3) AF-TIN, and 4) DF as a function of the maximum SNR ($\mathrm{SNR}_{\max} = \frac{\bar{P}}{\sigma^2}$) for fixed circuit power $P_c = 1$ W and different noise powers [83]. ©2015 IEEE

choose physically meaningful values for the noise variance and the circuit power consumption. Furthermore, assuming the same circuit power consumption for all schemes is unfair since they have different hardware complexity.

For this reason, a specific communication system is considered in [83, §VI-A]. A simple model for its energy consumption is developed which allows a fair comparison of the relaying schemes. In particular, a wireless board-to-board communication system as illustrated in Fig. 8.1b and operating at 200 GHz carrier frequency with 25 GHz bandwidth is evaluated. The circuit power consumption P_c is modeled such that it reflects the different hardware complexities of the transmission schemes. It is assumed that P_c depends neither on the power allocation nor the transmission rate. The only component whose energy consumption is modeled to scale with the transmission power is the power amplifier with an efficiency of 6.2 % [136]. The model for a DF relay is shown in Fig. 8.3. Transmitter and receiver are modeled similarly. The DSP power is assumed to scale discretely with the number of messages to decode: TIN uses one single user decoder, SND two single user decoders, and a DF relay has to decode all three messages. The NNC relay requires barely any DSP power, and the AF relay directly amplifies the analog signal and, thus, requires neither DSP nor ADC and DAC. Please refer to Table 8.1 and [83] for more details on the power consumptions model.

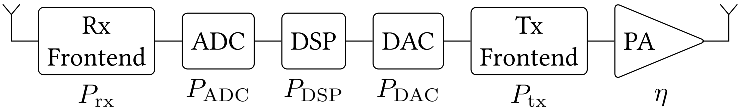


Figure 8.3: Energy consumption model for the relay.

Table 8.1: Parameters for circuit power modeling of board-to-board communications [83]. ©2015 IEEE

Component	Symbol	Value	Unit
Rx Frontend	P_{rx}	345	mW
Analog-to-digital converter	P_{ADC}	812	mW
Single User Decoder	P_{dec}	300	mW
Digital-to-analog converter	P_{DAC}	800	mW
Tx Frontend	P_{tx}	57	mW
Power Amplifier	η	6.2	%

Figure 8.4 displays the GEE with regard to this power consumption model. It can be seen that in the low SNR regime DF performs best, but starting from a transmit power limit of approximately 6.3 dBm, which corresponds to a transmit SNR of approximately 10 dB, AF-SND achieves the best GEE. This continues to hold for higher SNRs where DF is outperformed by all other schemes. Observe that AF performs much better than in Fig. 8.2 due to the low hardware complexity. This is our main motivation to focus on AF relaying for the remainder of this chapter. Other benefits of AF include very short relaying delay, especially compared to DF, and low hardware costs.

8.1 Discrete Memoryless Channels

In this section, we study the 3-user discrete memoryless multi-way relay channel (DM-MWRC) with instantaneous relaying and multiple unicast transmissions as described above. Instantaneous relaying [65] is a generalization of AF relaying. It restricts the relay to operate deterministically, memoryless and symbol-wise, i.e., the current output signal depends only on the currently observed input symbol. The AF relaying scheme [69, 108] is a special case of instantaneous relaying and implemented by a linear relaying function. However, AF is not necessarily the optimal instantaneous relaying technique and other relaying functions might achieve higher throughput without surrendering the benefits of AF. For example,

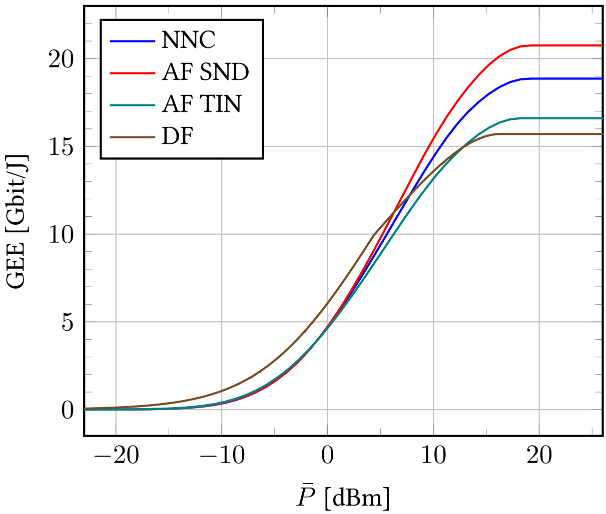


Figure 8.4: GEE in the 3-user MWRC of 1) NNC, 2) AF-SND, 3) AF-TIN, and 4) DF as a function of the SNR for wireless board-to-board communication at 200 GHz [83]. ©2015 IEEE

it is shown in [65] that a sawtooth relaying function outperforms AF in Gaussian relay channels with orthogonal receive components.

Restricting the relay to operate instantaneously has two main reasons: First, the processing delay at the relay is significantly less than in other schemes, e.g., DF. Second, the relay does not need power hungry analog-to-digital conversion and DSP which results in reduced hardware complexity and better EE. The downside is that noise is not cancelled at the relay and, thus, processed (e.g., amplified) in the same way as the signal. However, we have shown in the previous section that AF can outperform DF and similar relaying schemes in terms of EE due to the reduced circuit power consumption.

Because of the message exchange, each node only wants to recover one message and, hence, has to deal with interference. Some of this interference is self-induced and can be removed, but the remaining interference needs to be addressed. Common approaches are TIN and JD. With RS and superposition coding a combination of these two techniques is possible which, in general, results in higher transmission rates. This coding scheme is known as HK coding.

In this section, we first show that the DM-MWRC with instantaneous relaying is, for every fixed relaying function, equivalent to a 3-user IC with receiver message side information and feedback. Then, we extend the HK coding scheme to this channel and derive an achievable rate region, which, to the best of the author's knowledge, is the largest achievable rate region for this channel known to date. In addition, we also derive the SND rate region without RS.

8.1.1 System Model

We consider the 3-user DM-MWRC illustrated in Fig. 8.5. Recall that the message exchange has two defining properties:

1. Each user has a message to transmit which is intended for at least one other user.
2. Each user desires at most one message.

From these two properties it follows immediately that each message is only required at one other user. We denote the message of user k, $k \in \mathcal{K} = \{1, 2, 3\}$, as m_k and the node receiving it as $q(k)$. Furthermore, the user not interested in m_k is denoted by $l(k)$. Also, it follows from the properties listed above that user k, $k \in \mathcal{K}$, desires the message sent by user $l(k)$. Further, since m_k and $m_{l(k)}$ are to be decoded by users $q(k)$ and k, respectively, $m_{q(k)}$ is the desired message at node $l(k)$. This leaves us with two possible message exchanges, either clockwise as in Fig. 8.1b or counter-clockwise, depending on whether $q(1)$ is 2 or 3, respectively. Furthermore, it follows immediately that

$$\begin{aligned} q(q(k)) &= l(k) & l(q(k)) &= k \\ q(l(k)) &= k & l(l(k)) &= q(k) \\ q^{-1}(k) &= l(k) & l^{-1}(k) &= q(k). \end{aligned}$$

The messages of the users are assumed to be independent. They are communicated in n channel uses with the help of the relay. Each message m_k, $k \in \mathcal{K}$, is encoded into a codeword $\boldsymbol{x}_k^n$ of length n and transmitted over the channel. Upon receiving $\boldsymbol{y}_{q(k)}^n$, receiver $q(k)$, $k \in \mathcal{K}$, finds an estimate $\hat{m}_k$ of message m_k using its own message $m_{q(k)}$ as side information. We assume full-duplex operation at all nodes.

The 3-user DM-MWRC $(\mathcal{X}_0 \times \mathcal{X}_1 \times \mathcal{X}_2 \times X_3, p(y_0, y_1, y_2, y_3 | x_0, x_1, x_3, x_4),$ $\mathcal{Y}_0 \times \mathcal{Y}_1 \times \mathcal{Y}_2 \times \mathcal{Y}_3)$ consists of four finite input sets $\mathcal{X}_0$, $\mathcal{X}_1$, $\mathcal{X}_2$, $\mathcal{X}_3$, four finite output sets $\mathcal{Y}_0$, $\mathcal{Y}_1$, $\mathcal{Y}_2$, $\mathcal{Y}_3$, and a collection of conditional pmfs $p(y_0, y_1, y_2, y_3 | x_0,$ $x_1, x_3, x_4)$ on $\mathcal{Y}_0 \times \mathcal{Y}_1 \times \mathcal{Y}_2 \times \mathcal{Y}_3$.

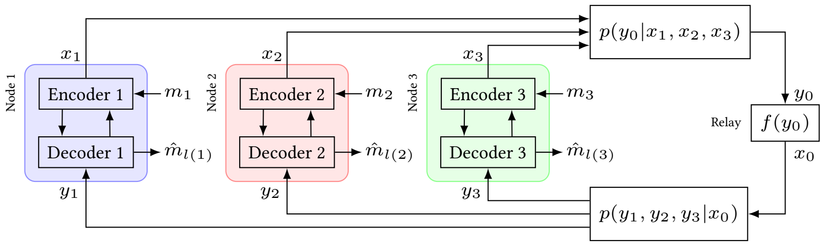


Figure 8.5: Block diagram of the 3-user DM-MWRC with circular message exchange and instantaneous relaying [76]. ©2015 IEEE

A $(2^{nR_1}, 2^{nR_2}, 2^{nR_3}, n)$ code for the 3-user DM-MWRC consists of

- three message sets $\mathcal{M}_k = \{1, \ldots, 2^{nR_k}\}$, one for each user $k \in \mathcal{K}$,
- three encoders, where encoder $k \in \mathcal{K}$ assigns a symbol $x_{ki}(m_k, \boldsymbol{y}_k^{i-1})$ to each message $m_k \in \mathcal{M}_k$ and received sequence $\boldsymbol{y}_k^{i-1}$ for $i = 1, 2, \ldots, n$,
- a relay encoder that assigns a symbol $x_{0i}(\boldsymbol{y}_0^{i-1})$ to every past received sequence $\boldsymbol{y}_0^{i-1}$ for $i = 1, 2, \ldots, n$, and
- three decoders, where decoder $q(k) \in \mathcal{K}$ assigns an estimate $\hat{m}_k \in \mathcal{M}_k$ or an error message e to each pair $(m_{q(k)}, \boldsymbol{y}_{q(k)}^n)$.

We assume that the message triple (M_1, M_2, M_3) is uniformly distributed over $\mathcal{M}_1 \times \mathcal{M}_2 \times \mathcal{M}_3$. The average probability of error is defined as $P_e^{(n)} = \Pr\left\{\hat{M}_k \neq M_k \text{ for some } k \in \mathcal{K}\right\}$. A rate triple (R_1, R_2, R_3) is said to be achievable if there exists a sequence of $(2^{nR_1}, 2^{nR_2}, 2^{nR_3}, n)$ codes such that $P_e^{(n)} \to 0$ as $n \to \infty$. The capacity region of the 3-user DM-MWRC is the closure of the set of achievable rates.

The MWRC does not include direct user-to-user links. This implies that the channel decomposes into an upstream channel $p(y_0|x_1, x_2, x_3)$ from the users to the relay, and a downstream channel $p(y_1, y_2, y_3|x_0)$ from the relay back to the users. Further, we constrain the relay to operate instantaneously on the received symbol y_{0i} using a deterministic mapping

$$\begin{aligned} f &: \mathcal{Y}_0 \to \mathcal{X}_0 \\ x_{0i} &= f(y_{0i}), \quad i = 1, \ldots, n. \end{aligned}$$

Then, for every fixed f, the channel effectively is a 3-user discrete memoryless interference channel (DM-IC) with receiver message side information and causal feedback with inputs (x_1, x_2, x_3), outputs (y_1, y_2, y_3) and a collection of conditional pmfs $p(y_1, y_2, y_3|x_1, x_2, x_3)$ given as

$$p(y_1, y_2, y_3|x_1, x_2, x_3) = \sum_{y_0 \in \mathcal{Y}_0} p(y_0|x_1, x_2, x_3)\, p_{Y_1 Y_2 Y_3|X_0}(y_1, y_2, y_3|f(y_0)).$$

From a practical perspective, the assumption of instantaneous relaying is problematic. However, assume that the relay has a delay of d symbols. In this case, the relaying function is

$$x_{0i} = \begin{cases} 0 & \text{if } i < d \\ f(y_{0,i-d}) & \text{otherwise.} \end{cases}$$

This implies that the messages in blocks $n-d+1, \cdots, n$ cannot be communicated. Thus, the maximum achievable rates are decreased by a factor $\frac{n-d}{n}$. Since

$$\lim_{n\to\infty} \frac{n-d}{n} = 1$$

this does not impair the achievable rates for asymptotically error free communication and we can use the instantaneous relay assumption WLOG.

Next, we carefully adapt results for the classical DM-IC to the considered channel with receiver message side information.

8.1.2 Simultaneous Non-Unique Decoding

Recent results show that SND is an optimal decoder for general interference networks under the restriction to random codebooks with superposition coding and time sharing [5]. Before, common wisdom was that neither TIN nor SND dominates the other rate-wise, with TIN generally better in noise limited scenarios and SND superior when interference is the limiting factor. This misconception is due to a longstanding oversight in the SND proof that was clarified in [5]. Using PTP codes and SND, we obtain the following achievable rate region that contains TIN as a special case.

8.1 Theorem (Simultaneous non-unique decoding). A rate triple (R_1, R_2, R_3) is achievable for the DM-MWRC $p(y_0, y_1, y_2, y_3|x_0, x_1, x_3, x_4)$ using SND if, for all $k \in \mathcal{K}$,

$$R_k \leq I\big(X_k; Y_{q(k)}|X_{q(k)}, Q\big)\,, \tag{8.1}$$

or

$$R_k \le I\big(X_k; Y_{q(k)} | X_{l(k)}, X_{q(k)}, Q\big)\,, \tag{8.2a}$$

$$R_k + R_{l(k)} \le I\big(X_k, X_{l(k)}; Y_{q(k)} | X_{q(k)}, Q\big)\,, \tag{8.2b}$$

for some pmf $p(q)p(x_1|q)p(x_2|q)p(x_3|q)$, and some deterministic mapping $f(y_0)$ at the relay. ◁

Proof. Codebook generation: Fix $p(q)\prod_{k\in\mathcal{K}} p(x_k|q)$ and $f(y_0)$. Generate a sequence $\boldsymbol{q}^n$ according to $\prod_{i=1}^n p_Q(q_i)$. For each $k \in \mathcal{K}$, randomly and conditionally independently generate 2^{nR_k} sequences $\boldsymbol{x}_k^n(m_k)$, $m_k \in \{1,\ldots,2^{nR_k}\}$, each according to $\prod_{i=1}^n p_{X_k|Q}(x_{ki}|q_i)$.

Encoding: To send m_k, encoder $k \in \mathcal{K}$ transmits $\boldsymbol{x}_k^n(m_k)$.

Decoding: We use SND. Suppose that receiver $q(k)$ observes $\boldsymbol{y}_{q(k)}^n$. Then it finds the unique $\hat{m}_k$ such that

$$\big(\boldsymbol{q}^n, \boldsymbol{x}_k^n(\hat{m}_k), \boldsymbol{x}_{l(k)}^n(m_{l(k)}), \boldsymbol{x}_{q(k)}^n(m_{q(k)}), \boldsymbol{y}_{q(k)}^n\big) \in \mathcal{T}_\varepsilon^{(n)}$$

for some $m_{l(k)} \in \{1,\ldots,2^{nR_{l(k)}}\}$ using its own message $m_{q(k)}$ as side information. Otherwise it declares an error.

Analysis of the probability of error: To bound the average probability of error for decoder $q(k)$, $k \in \mathcal{K}$, we assume WLOG that $(M_1, M_2, M_3) = (1,1,1)$ is sent. Then, the decoder makes an error if and only if one or more of the following events occur:

$$\begin{aligned}
\mathcal{E}_{q(k),0} &= \{(\boldsymbol{Q}^n, \boldsymbol{X}_k^n(1), \boldsymbol{X}_{l(k)}^n(1), \boldsymbol{X}_{q(k)}^n(1), \boldsymbol{Y}_{q(k)}^n) \notin \mathcal{T}_\varepsilon^{(n)}\},\\
\mathcal{E}_{q(k),1} &= \{(\boldsymbol{Q}^n, \boldsymbol{X}_k^n(m_k), \boldsymbol{X}_{l(k)}^n(m_{l(k)}), \boldsymbol{X}_{q(k)}^n(1), \boldsymbol{Y}_{q(k)}^n) \in \mathcal{T}_\varepsilon^{(n)}\\
&\qquad\qquad\qquad\qquad\qquad \text{for some } m_k \neq 1 \text{ and some } m_{l(k)}\}.
\end{aligned}$$

Due to the union bound, the average probability of error for decoder $q(k)$ is bounded above as $\Pr(\mathcal{E}_{q(k)}) \le \sum_{i=0}^2 \Pr(\mathcal{E}_{q(k),i})$. By the law of large numbers (LLN) [36, Thm. 1.7.4], $\Pr(\mathcal{E}_{q(k),0})$ tends to zero as $n \to 0$.

In [5, §II-A] it was observed that there are two ways to bound $\mathcal{E}_{q(k),1}$. First, the joint typicality of $(\boldsymbol{Q}^n, \boldsymbol{X}_k^n(m_k), \boldsymbol{X}_{l(k)}^n(m_{l(k)}), \boldsymbol{X}_{q(k)}^n(1), \boldsymbol{Y}_{q(k)}^n)$ for every $m_{l(k)}$ implies that $(\boldsymbol{Q}^n, \boldsymbol{X}_k^n(m_k), \boldsymbol{X}_{q(k)}^n(1), \boldsymbol{Y}_{q(k)}^n)$ is also jointly typical for each $m_{l(k)}$. Hence,

$$\mathcal{E}_{q(k),1} \subseteq \{(\boldsymbol{Q}^n, \boldsymbol{X}_k^n(m_k), \boldsymbol{X}_{q(k)}^n(1), \boldsymbol{Y}_{q(k)}^n) \in \mathcal{T}_\varepsilon^{(n)} \text{for some } m_k \neq 1\} = \mathcal{E}'_{q(k),1}$$

and $\Pr(\mathcal{E}_{q(k),1}) \le \Pr(\mathcal{E}_{q(k),1'})$. By the packing lemma [29, §3.2], $\Pr(\mathcal{E}_{q(k),1})$ tends to zero as $n \to \infty$ if $R_k \le I\big(X_k; Y_{q(k)} | X_{q(k)}, Q\big) - \delta(\varepsilon)$.

The second way to bound $\Pr(\mathcal{E}_{q(k),1})$ is the "classical" method. First, split the error event $\mathcal{E}_{q(k),1}$ into two individual events

$$\begin{aligned}
\mathcal{E}_{q(k),11} &= \{(\boldsymbol{Q}^n, \boldsymbol{X}_k^n(m_k), \boldsymbol{X}_{l(k)}^n(1), \boldsymbol{X}_{q(k)}^n(1), \boldsymbol{Y}_{q(k)}^n) \in \mathcal{T}_\varepsilon^{(n)} \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{for some } m_k \neq 1\}, \\
\mathcal{E}_{q(k),12} &= \{(\boldsymbol{Q}^n, \boldsymbol{X}_k^n(m_k), \boldsymbol{X}_{l(k)}^n(m_{l(k)}), \boldsymbol{X}_{q(k)}^n(1), \boldsymbol{Y}_{q(k)}^n) \in \mathcal{T}_\varepsilon^{(n)} \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{for some } m_k \neq 1, m_{l(k)} \neq 1\}.
\end{aligned}$$

Second, use the packing lemma [29, §3.2] to bound each of these events, i.e., $\Pr(\mathcal{E}_{q(k),11})$, and $\Pr(\mathcal{E}_{q(k),12})$, tend to zero as $n \to 0$ if the conditions $R_k \le I\big(X_k; Y_{q(k)} | X_{l(k)}, X_{q(k)}, Q\big) - \delta(\varepsilon)$, and

$$R_k + R_{l(k)} \le I\big(X_k, X_{l(k)}; Y_{q(k)} | X_{q(k)}, Q\big) - \delta(\varepsilon)$$

are satisfied, respectively. The bound of $\Pr(\mathcal{E}_{q(k),1})$ again follows from the union bound, i.e., $\Pr(\mathcal{E}_{q(k),1}) \le \Pr(\mathcal{E}_{q(k),11}) + \Pr(\mathcal{E}_{q(k),12})$. Finally, taking $\varepsilon \to 0$ concludes the proof. □

8.2 Remark. Let $\mathcal{R}_{k,\text{TIN}}$ and $\mathcal{R}_{k,\text{JD}}$ be the regions defined by (8.1) and (8.2), respectively. Then, similarly to (2.5), the rate region in Theorem 8.1 is

$$\mathcal{R} = \bigcap_{k \in \mathcal{K}} (\mathcal{R}_{k,\text{TIN}} \cup \mathcal{R}_{k,\text{JD}}) = \bigcup_{\boldsymbol{d} \in \{\text{TIN},\text{JD}\}^{|\mathcal{K}|}} \bigcap_{k \in \mathcal{K}} \mathcal{R}_{k,d_k}. \qquad \triangleleft$$

Under the additional assumption of restricted encoders, i.e., the encoders $x_{ki}(m_k, \boldsymbol{y}_k^{i-1})$ are not allowed to use the available feedback and are equal to $x_{ki}(m_k)$, the rate region Theorem 8.1 is the largest region achievable with random coding and the given codebook construction [5, Thm. 2]. Nevertheless, interactive channel coding that employs the available channel observations $\boldsymbol{y}_k^{i-1}$ might improve this region.

8.1.3 Rate Splitting and Superposition Coding

The achievable rate region in Theorem 8.1 requires each decoder to decide between treating the interfering message as noise or jointly decoding it. We can improve upon this scheme by introducing RS and superposition coding.

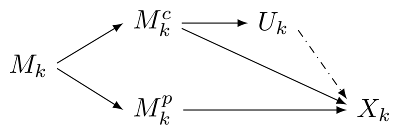


Figure 8.6: Codebook construction for HK coding. Solid lines show a direct relationship while dashed lines indicate statistical dependence.

This allows arbitrary combinations of JD and TIN at each decoder. In particular, each message m_k, $k \in \mathcal{K}$, is represented by an independent common message m_k^c at rate R_k^c and a private message m_k^p at rate R_k^p. These messages are sent via superposition coding with cloud center $u_k(m_k^c)$ and satellite codeword $x_k(m_k^c, m_k^p)$. Receiver $q(k)$, $k \in \mathcal{K}$, recovers all common messages and its desired private message m_k^p using its own message as side information. For the 2-user IC, this is known as HK coding. The codebook construction here follows the modern version in [19] instead of the classical one from [39]. It is illustrated in Fig. 8.6.

For notational convenience, we define $R_\Sigma = R_1 + R_2 + R_3$. The achievable rate region is stated in the theorem below.

8.3 Theorem. A rate triple (R_1, R_2, R_3) is achievable for the DM-MWRC $p(y_0, y_1, y_2, y_3|x_0, x_1, x_3, x_4)$ if, for all $k \in \mathcal{K}$,

$$\begin{aligned} R_k &\le B_k, \\ R_k + R_{q(k)} &\le A_k + D_{q(k)}, \\ R_\Sigma &\le A_k + C_{q(k)} + D_{l(k)}, \\ R_k + R_\Sigma &\le A_k + C_{q(k)} + C_{l(k)} + D_k, \\ R_\Sigma &\le C_1 + C_2 + C_3, \end{aligned}$$

where

$$A_k = I\big(X_k; Y_{q(k)}|U_k, U_{l(k)}, X_{q(k)}, Q\big) \tag{8.3a}$$

$$B_k = I\big(X_k; Y_{q(k)}|U_{l(k)}, X_{q(k)}, Q\big) \tag{8.3b}$$

$$C_k = I\big(X_k, U_{l(k)}; Y_{q(k)}|U_k, X_{q(k)}, Q\big) \tag{8.3c}$$

$$D_k = I\big(X_k, U_{l(k)}; Y_{q(k)}|X_{q(k)}, Q\big) \tag{8.3d}$$

for some pmf $p(q)p(u_1, x_1|q)p(u_2, x_2|q)p(u_3, x_3|q)$, and some deterministic mapping $f(y_0)$ at the relay, where $|\mathcal{U}_k| \le |\mathcal{X}_k| + 3$, $k \in \mathcal{K}$, and $|\mathcal{Q}| \le 9$. $\triangleleft$

Before we proof Theorem 8.3, we need the following lemma.

8.4 Lemma. A rate triple (R_1, R_2, R_3) is achievable for the DM-MWRC $p(y_0, y_1, y_2, y_3|x_0, x_1, x_3, x_4)$ if, for all $k \in \mathcal{K}$,

$$\begin{aligned} R_k &\leq \min\{B_k,\, A_k + C_{q(k)}\} \\ R_k + R_{q(k)} &\leq A_k + \min\left\{D_{q(k)},\, C_{q(k)} + C_{l(k)}\right\}, \\ R_\Sigma &\leq A_k + C_{q(k)} + D_{l(k)}, \\ R_k + R_\Sigma &\leq A_k + C_{q(k)} + C_{l(k)} + D_k, \\ R_\Sigma &\leq C_1 + C_2 + C_3, \end{aligned}$$

for some pmf $p(q)p(u_1, x_1|q)p(u_2, x_2|q)p(u_3, x_3|q)$, and some deterministic mapping $f(y_0)$ at the relay, where A_k, ..., D_k are as defined in Theorem 8.3. $\triangleleft$

Proof. We first show that $(R_1^c, R_2^c, R_3^c, R_1^p, R_2^p, R_3^p)$ is achievable if, for every $k \in \mathcal{K}$,

$$\begin{aligned} R_k^p &\leq I\big(X_k; Y_{q(k)}|U_k, U_{l(k)}, X_{q(k)}, Q\big), \\ R_k^c + R_k^p &\leq I\big(X_k; Y_{q(k)}|U_{l(k)}, X_{q(k)}, Q\big), \\ R_k^p + R_{l(k)}^c &\leq I\big(X_k, U_{l(k)}; Y_{q(k)}|U_k, X_{q(k)}, Q\big), \\ R_k^c + R_k^p + R_{l(k)}^c &\leq I\big(X_k, U_{l(k)}; Y_{q(k)}|X_{q(k)}, Q\big), \end{aligned}$$

for some pmf $p(q)p(u_1, x_1|q)p(u_2, x_2|q)p(u_3, x_3|q)$. Then, we use the Fourier-Motzkin procedure [29, Appendix D] to project this rate region onto (R_1, R_2, R_3).

Codebook generation: Fix $p(q)\prod_{k\in\mathcal{K}} p(u_k, x_k|q)$ and $f(y_0)$. Generate a sequence $\boldsymbol{q}^n$ according to $\prod_{i=1}^n p_Q(q_i)$. For each $k \in \mathcal{K}$, randomly and conditionally independently generate $2^{nR_k^c}$ sequences $\boldsymbol{u}_k^n(m_k^c)$, $m_k^c \in \{1, \ldots, 2^{nR_k^c}\}$, each according to $\prod_{i=1}^n p_{U_k|Q}(u_{ki}|q_i)$. For each m_k^c, randomly and conditionally independently generate $2^{nR_k^p}$ sequences $\boldsymbol{x}_k^n(m_k^c, m_k^p)$, $m_k^p \in \{1, \ldots, 2^{nR_k^p}\}$, each according to $\prod_{i=1}^n p_{X_k|U_k,Q}(x_{ki}|u_{ki}(m_k^c), q_i)$.

Encoding: To send $m_k = (m_k^c, m_k^p)$, encoder $k \in \mathcal{K}$ transmits $\boldsymbol{x}_k^n(m_k^c, m_k^p)$.

Decoding: We use SND. Suppose that receiver $q(k)$ observes $\boldsymbol{y}_{q(k)}^n$. Then it finds the unique $(\hat{m}_k^c, \hat{m}_k^p)$ such that

$$\begin{aligned} \big(\boldsymbol{q}^n, \boldsymbol{u}_k^n(\hat{m}_k^c), \boldsymbol{u}_{l(k)}^n(m_{l(k)}^c), \boldsymbol{x}_k^n(\hat{m}_k^c, \hat{m}_k^p), \\ \boldsymbol{u}_{q(k)}^n(m_{q(k)}^c), \boldsymbol{x}_{q(k)}^n(m_{q(k)}^c, m_{q(k)}^p), \boldsymbol{y}_{q(k)}^n\big) \in \mathcal{T}_\varepsilon^{(n)} \end{aligned}$$

for some $m^c_{l(k)} \in \{1,\dots,2^{nR^c_{l(k)}}\}$ using its own messages $(m^c_{q(k)}, m^p_{q(k)})$ as side information. Otherwise it declares an error.

Analysis of the probability of error: To bound the average probability of error for decoder $q(k)$, $k \in \mathcal{K}$, we assume WLOG that $(M^c_k, M^p_k) = (1,1)$ is sent for all $k \in \mathcal{K}$. Then, the decoder makes an error if and only if one or more of the following events occur:

$$\begin{aligned}
\mathcal{E}_{q(k),0} &= \{(\boldsymbol{Q}^n, \boldsymbol{U}^n_k(1), \boldsymbol{U}^n_{l(k)}(1), \boldsymbol{X}^n_k(1,1), \boldsymbol{U}^n_{q(k)}(1), \\
&\qquad\qquad\qquad \boldsymbol{X}^n_{q(k)}(1,1), \boldsymbol{Y}^n_{q(k)}) \notin \mathcal{T}^{(n)}_\varepsilon \}, \\
\mathcal{E}_{q(k),1} &= \{(\boldsymbol{Q}^n, \boldsymbol{U}^n_k(1), \boldsymbol{U}^n_{l(k)}(1), \boldsymbol{X}^n_k(1,m^p_k), \boldsymbol{U}^n_{q(k)}(1), \\
&\qquad \boldsymbol{X}^n_{q(k)}(1,1), \boldsymbol{Y}^n_{q(k)}) \in \mathcal{T}^{(n)}_\varepsilon \text{ for some } m^p_k \neq 1\}, \\
\mathcal{E}_{q(k),2} &= \{(\boldsymbol{Q}^n, \boldsymbol{U}^n_k(m^c_k), \boldsymbol{U}^n_{l(k)}(1), \boldsymbol{X}^n_k(m^c_k,m^p_k), \boldsymbol{U}^n_{q(k)}(1), \\
&\qquad \boldsymbol{X}^n_{q(k)}(1,1), \boldsymbol{Y}^n_{q(k)}) \in \mathcal{T}^{(n)}_\varepsilon \text{ for some } m^c_k \neq 1, m^p_k\}, \\
\mathcal{E}_{q(k),3} &= \{(\boldsymbol{Q}^n, \boldsymbol{U}^n_k(1), \boldsymbol{U}^n_{l(k)}(m^c_{l(k)}), \boldsymbol{X}^n_k(1,m^p_k), \boldsymbol{U}^n_{q(k)}(1), \\
&\qquad \boldsymbol{X}^n_{q(k)}(1,1), \boldsymbol{Y}^n_{q(k)}) \in \mathcal{T}^{(n)}_\varepsilon \text{ for some } m^c_{l(k)} \neq 1, m^p_k \neq 1\}, \\
\mathcal{E}_{q(k),4} &= \{(\boldsymbol{Q}^n, \boldsymbol{U}^n_k(m^c_k), \boldsymbol{U}^n_{l(k)}(m^c_{l(k)}), \boldsymbol{X}^n_k(m^c_k,m^p_k), \boldsymbol{U}^n_{q(k)}(1), \\
&\qquad \boldsymbol{X}^n_{q(k)}(1,1), \boldsymbol{Y}^n_{q(k)}) \in \mathcal{T}^{(n)}_\varepsilon \text{ for some } m^c_k \neq 1, m^c_{l(k)} \neq 1, m^p_k\}.
\end{aligned}$$

Due to the union bound, the average probability of error for decoder $q(k)$ is bounded above as $\Pr(\mathcal{E}_{q(k)}) \leq \sum_{i=0}^4 \Pr(\mathcal{E}_{q(k),i})$.

By the LLN [36, Thm. 1.7.4] $\Pr(\mathcal{E}_{q(k),0})$ tends to zero as $n \to 0$. By the packing lemma [29, §3.2] and since

$$U_k - (Q, X_k) - \big(Y_1, Y_2, Y_3, U_{q(k)}, X_{q(k)}, U_{l(k)}, X_{l(k)}\big) \tag{8.4}$$

form a Markov chain for every $k \in \mathcal{K}$, $\Pr(\mathcal{E}_{q(k),1})$, $\Pr(\mathcal{E}_{q(k),2})$, $\Pr(\mathcal{E}_{q(k),3})$, and $\Pr(\mathcal{E}_{q(k),4})$ tend to zero as $n \to 0$ if the conditions

$$\begin{aligned}
R^p_k &\leq I\big(X_k; Y_{q(k)} | U_k, U_{l(k)}, U_{q(k)}, X_{q(k)}, Q\big) - \delta(\varepsilon) \\
R^c_k + R^p_k &\leq I\big(X_k; Y_{q(k)} | U_{l(k)}, U_{q(k)}, X_{q(k)}, Q\big) - \delta(\varepsilon) \\
R^p_k + R^c_{l(k)} &\leq I\big(X_k, U_{l(k)}; Y_{q(k)} | U_k, U_{q(k)}, X_{q(k)}, Q\big) - \delta(\varepsilon)
\end{aligned}$$

and

$$R^c_k + R^p_k + R^c_{l(k)} \leq I\big(U_k, X_k, U_{l(k)}; Y_{q(k)} | U_{q(k)}, X_{q(k)}, Q\big) - \delta(\varepsilon)$$

are satisfied, respectively, where $\delta(\varepsilon)$ is a functions that tends to zero as $\varepsilon \to 0$.

Finally, by substituting $R_k^p = R_k - R_k^c$ and using Fourier-Motzkin we obtain Lemma 8.4. □

Proof of Theorem 8.3

We show that the rate regions in Theorem 8.3 and Lemma 8.4 are equal. For that, we use the same approach as in [19].

Let $\mathcal{P}$ be the set of pmfs on $\mathcal{Q} \times \mathcal{U}_1 \times \mathcal{X}_1 \times \mathcal{U}_2 \times \mathcal{X}_2 \times \mathcal{U}_3 \times \mathcal{X}_3$ that factor as $p(q)p(u_1, x_1|q)p(u_2, x_2|q)p(u_3, x_3|q)$. Fix an input pmf $P \in \mathcal{P}$ and let $\mathcal{R}^o(P)$ be the achievable rate region from Lemma 8.4 evaluated for P. For a fixed $\kappa \in \mathcal{K}$, this rate region is given below in explicit form:

$$R_\kappa \leq B_\kappa \tag{8.5a}$$
$$R_{q(\kappa)} \leq B_{q(\kappa)} \tag{8.5b}$$
$$R_{l(\kappa)} \leq B_{l(\kappa)} \tag{8.5c}$$
$$R_\kappa \leq A_\kappa + C_{q(\kappa)} \tag{8.5d}$$
$$R_{q(\kappa)} \leq A_{q(\kappa)} + C_{l(\kappa)} \tag{8.5e}$$
$$R_{l(\kappa)} \leq A_{l(\kappa)} + C_\kappa \tag{8.5f}$$
$$R_\kappa + R_{q(\kappa)} \leq A_\kappa + D_{q(\kappa)} \tag{8.5g}$$
$$R_{q(\kappa)} + R_{l(\kappa)} \leq A_{q(\kappa)} + D_{l(\kappa)} \tag{8.5h}$$
$$R_{l(\kappa)} + R_\kappa \leq A_{l(\kappa)} + D_\kappa \tag{8.5i}$$
$$R_\kappa + R_{q(\kappa)} \leq A_\kappa + C_{q(\kappa)} + C_{l(\kappa)} \tag{8.5j}$$
$$R_{q(\kappa)} + R_{l(\kappa)} \leq A_{q(\kappa)} + C_{l(\kappa)} + C_\kappa \tag{8.5k}$$
$$R_{l(\kappa)} + R_\kappa \leq A_{l(\kappa)} + C_\kappa + C_{q(\kappa)} \tag{8.5l}$$
$$R_\kappa + R_{q(\kappa)} + R_{l(\kappa)} \leq A_\kappa + C_{q(\kappa)} + D_{l(\kappa)} \tag{8.5m}$$
$$R_{q(\kappa)} + R_{l(\kappa)} + R_\kappa \leq A_{q(\kappa)} + C_{l(\kappa)} + D_\kappa \tag{8.5n}$$
$$R_{l(\kappa)} + R_\kappa + R_{q(\kappa)} \leq A_{l(\kappa)} + C_\kappa + D_{q(\kappa)} \tag{8.5o}$$
$$R_\kappa + R_{q(\kappa)} + R_{l(\kappa)} \leq C_\kappa + C_{q(\kappa)} + C_{l(\kappa)} \tag{8.5p}$$
$$2R_\kappa + R_{q(\kappa)} + R_{l(\kappa)} \leq A_\kappa + C_{q(\kappa)} + C_{l(\kappa)} + D_\kappa \tag{8.5q}$$
$$2R_{q(\kappa)} + R_{l(\kappa)} + R_\kappa \leq A_{q(\kappa)} + C_{l(\kappa)} + C_\kappa + D_{q(\kappa)} \tag{8.5r}$$
$$2R_{l(\kappa)} + R_\kappa + R_{q(\kappa)} \leq A_{l(\kappa)} + C_\kappa + C_{q(\kappa)} + D_{l(\kappa)}. \tag{8.5s}$$

Then, $\mathcal{R}^o = \bigcup_{P \in \mathcal{P}} R^o(P)$ is the rate region in Lemma 8.4 for a fixed f.

Further, define $\tilde{\mathcal{R}}^c(P)$ as the set of all rate tuples (R_1, R_2, R_3) satisfying (8.5a) to (8.5c) and (8.5g) to (8.5s), and $\mathcal{R}^c(P)$ as the set of all rate tuples (R_1, R_2, R_3) satisfying (8.5a) to (8.5c), (8.5g) to (8.5i) and (8.5m) to (8.5s). Let

$$\tilde{\mathcal{R}}^c = \bigcup_{P \in \mathcal{P}} \tilde{\mathcal{R}}^c(P) \qquad \text{and} \qquad \mathcal{R}^c = \bigcup_{P \in \mathcal{P}} \mathcal{R}^c(P).$$

Thus, $\mathcal{R}^c$ is the rate region in Theorem 8.3 for a fixed f. Our goal is to show that $\mathcal{R}^c = \mathcal{R}^o$.

Obviously, $\mathcal{R}^o \subseteq \tilde{\mathcal{R}}^c \subseteq \mathcal{R}^c$ since, for every $P \in \mathcal{P}$, $\mathcal{R}^o(P) \subseteq \tilde{\mathcal{R}}^c(P) \subseteq \mathcal{R}^c(P)$. Thus, for equality to hold, we have to show that $\mathcal{R}^c \subseteq \tilde{\mathcal{R}}^c \subseteq \mathcal{R}^o$. We first show that the second inclusion holds and proceed with a distinction of cases to show that, for every $P \in \mathcal{P}$, $\tilde{\mathcal{R}}^c(P) \subseteq \mathcal{R}^o$. First, if $R_k \leq A_k + C_{q(k)}$ for all $k \in \mathcal{K}$, then, obviously, $\tilde{\mathcal{R}}^c(P) \subseteq \mathcal{R}^o(P)$. Next, consider the case

$$R_\kappa > A_\kappa + C_{q(\kappa)}. \tag{8.6}$$

We construct a new input pmf $P_\kappa \in \mathcal{P}$ from P that results in a $\mathcal{R}^o(P_\kappa)$ such that $\tilde{\mathcal{R}}^c(P) \subseteq \mathcal{R}^o(P_\kappa)$. For this inclusion to hold, we have to show that every inequality in $\mathcal{R}^o(P_\kappa)$ is implied by $\tilde{\mathcal{R}}^c(P)$ and (8.6).

Let $P_\kappa = \sum_{u_\kappa \in \mathcal{U}_\kappa} P$ be the marginal pmf of P where U_κ has been marginalized out and set $U_\kappa = \emptyset$. Then, $\mathcal{R}^o(P_\kappa)$ consists of all nonnegative rate tuples (R_1, R_2, R_3) such that

$$R_\kappa \leq B_\kappa \tag{8.7a}$$
$$R_{q(\kappa)} \leq B^*_{q(\kappa)} \tag{8.7b}$$
$$R_{q(\kappa)} \leq A^*_{q(\kappa)} + C_{l(\kappa)} \tag{8.7c}$$
$$R_{l(\kappa)} \leq B_{l(\kappa)} \tag{8.7d}$$
$$R_\kappa + R_{l(\kappa)} \leq A_{l(\kappa)} + D_\kappa \tag{8.7e}$$
$$R_{q(\kappa)} + R_{l(\kappa)} \leq A^*_{q(\kappa)} + D_{l(\kappa)} \tag{8.7f}$$
$$R_\kappa + R_{q(\kappa)} + R_{l(\kappa)} \leq D_\kappa + A^*_{q(\kappa)} + C_{l(\kappa)} \tag{8.7g}$$

where

$$A^*_{q(\kappa)} = I\big(X_{q(\kappa)}; Y_{l(\kappa)} | U_{q(\kappa)}, X_{l(\kappa)}, Q\big)\,,$$
$$B^*_{q(\kappa)} = I\big(X_{q(\kappa)}; Y_{l(\kappa)} | X_{l(\kappa)}, Q\big)\,.$$

Equations (8.7a), (8.7d), and (8.7e) are the same as (8.5a), (8.5c), and (8.5i), respectively. From (8.4), (8.5g) and (8.6), we obtain

$$R_{q(\kappa)} < I\big(U_{q(\kappa)}; Y_{l(\kappa)} | X_{l(\kappa)}, Q\big) \leq B^*_{q(\kappa)},$$

from (8.5j) and (8.6), we obtain

$$R_{q(\kappa)} < C_{l(\kappa)} \leq A^*_{q(\kappa)} + C_{l(\kappa)},$$

from (8.5m) and (8.6), we obtain

$$R_{q(\kappa)} + R_{l(\kappa)} < D_{l(\kappa)} \leq D_{l(\kappa)} + A^*_{q(\kappa)},$$

and, finally, from (8.5q) and (8.6), we obtain

$$R_\kappa + R_{q(\kappa)} + R_{l(\kappa)} < C_{l(\kappa)} + D_\kappa \leq C_{l(\kappa)} + D_\kappa + A^*_{q(\kappa)}.$$

The proof for the cases $R_{q(\kappa)} \leq A_{q(\kappa)} + C_{l(\kappa)}$ and $R_{l(\kappa)} \leq A_{l(\kappa)} + C_\kappa$ follow exactly along the same lines due to symmetry. Hence, for every $P \in \mathcal{P}$, $\tilde{\mathcal{R}}^c(P) \subseteq \mathcal{R}^o(P) \cup \big(\bigcup_{k\in\mathcal{K}} \mathcal{R}^o(P_k)\big)$ with $P_k = \sum_{u_k \in \mathcal{U}_k} P$. Thus, $\tilde{\mathcal{R}}^c = \mathcal{R}^o$.

Next, we show that $\mathcal{R}^c \subseteq \tilde{\mathcal{R}}^c$ using similar steps as before. First, if $R_k + R_{q(k)} \leq A_k + C_{q(k)} + C_{l(k)}$ for all $k \in \mathcal{K}$, then $\mathcal{R}^c(P) \subseteq \tilde{\mathcal{R}}^c(P)$. Further, consider the case

$$R_\kappa + R_{l(\kappa)} > A_{l(\kappa)} + C_\kappa + C_{q(\kappa)}. \tag{8.8}$$

Let P_κ be as before. Then, $\tilde{\mathcal{R}}^c(P_\kappa)$ consists of all nonnegative rate tuples (R_1, R_2, R_3) such that

$$R_\kappa \leq B_\kappa \tag{8.9a}$$
$$R_{q(\kappa)} \leq B^*_{q(\kappa)} \tag{8.9b}$$
$$R_{l(\kappa)} \leq B_{l(\kappa)} \tag{8.9c}$$
$$R_\kappa + R_{l(\kappa)} \leq A_{l(\kappa)} + D_\kappa \tag{8.9d}$$
$$R_{q(\kappa)} + R_{l(\kappa)} \leq A^*_{q(\kappa)} + D_{l(\kappa)} \tag{8.9e}$$
$$R_\kappa + R_{q(\kappa)} \leq B_\kappa + A^*_{q(\kappa)} + C_{l(\kappa)} \tag{8.9f}$$
$$R_\kappa + R_{q(\kappa)} + R_{l(\kappa)} \leq D_\kappa + A^*_{q(\kappa)} + C_{l(\kappa)} \tag{8.9g}$$

Equations (8.9a), (8.9c), and (8.9d) are the same as (8.5a), (8.5c), and (8.5i), respectively. From (8.4), (8.5o) and (8.8), we obtain

$$R_{q(\kappa)} < I\big(U_{q(\kappa)}; Y_{l(\kappa)} | X_{l(\kappa)}, Q\big) \leq B^*_{q(\kappa)},$$

from (8.5s) and (8.8), we obtain

$$R_{q(\kappa)} + R_{l(\kappa)} < D_{l(\kappa)} \leq A^*_{q(\kappa)} + D_{l(\kappa)},$$

from (8.4), (8.5q) and (8.8), we obtain

$$R_\kappa + R_{q(\kappa)} < A_\kappa + D_\kappa - C_\kappa + C_{l(\kappa)} - A_{l(\kappa)} \leq B_\kappa + A^*_{q(\kappa)} + C_{l(\kappa)},$$

and, finally, from (8.5i), (8.5p) and (8.8), we obtain

$$R_\kappa + R_{q(\kappa)} + R_{l(\kappa)} < C_{l(\kappa)} + D_\kappa \leq D_\kappa + A^*_{q(\kappa)} + C_{l(\kappa)}.$$

The proof for the remaining cases $R_{q(\kappa)} + R_\kappa \leq A_\kappa + C_{q(\kappa)} + C_{l(\kappa)}$ and $R_{l(\kappa)} + R_{q(\kappa)} \leq A_{q(\kappa)} + C_{l(\kappa)} + C_\kappa$ is similar. Hence, $\mathcal{R}^c(P) \subseteq \tilde{\mathcal{R}}^c(P) \cup \left(\bigcup_{k\in\mathcal{K}} \tilde{\mathcal{R}}^c(P_k)\right)$. Thus, $\mathcal{R}^c = \tilde{\mathcal{R}}^c = \mathcal{R}^o$.

Cardinality Bounds Regarding the bound on $\mathcal{Q}$, from the Support Lemma in [29, Appendix C] and (8.3) follows $|\mathcal{Q}| \leq 12$. However, this bound can be tightened. Consider the following functions of Q for some $\kappa \in \mathcal{K}$:

$$\begin{array}{lll}
f_1(Q) = B_\kappa & f_5(Q) = A_{l(\kappa)} - C_{l(\kappa)} + D_{l(\kappa)} & f_8(Q) = A_\kappa + D_{q(\kappa)} \\
f_2(Q) = B_{q(\kappa)} & f_6(Q) = D_\kappa + C_{l(\kappa)} - D_{l(\kappa)} & f_9(Q) = A_{q(\kappa)} + D_{l(\kappa)} \\
f_3(Q) = B_{l(\kappa)} & f_7(Q) = C_{q(\kappa)} - D_{q(\kappa)} + D_{l(\kappa)} & \\
\multicolumn{3}{l}{f_4(Q) = C_\kappa + D_{q(\kappa)} + C_{l(\kappa)} - D_{l(\kappa)}}
\end{array}$$

Then, $\mathcal{R}^c$ is

$$\begin{array}{cc}
R_\kappa \leq f_1(Q) & R_\kappa + R_{q(\kappa)} \leq f_8(Q) \\
R_{q(\kappa)} \leq f_2(Q) & R_{q(\kappa)} + R_{l(\kappa)} \leq f_9(Q) \\
R_{l(\kappa)} \leq f_3(Q) & R_{l(\kappa)} + R_\kappa \leq f_5(Q) + f_6(Q)
\end{array}$$

$$\begin{aligned}
R_\kappa + R_{q(\kappa)} + R_{l(\kappa)} &\leq f_7(Q) + f_8(Q) \\
R_{q(\kappa)} + R_{l(\kappa)} + R_\kappa &\leq f_6(Q) + f_9(Q) \\
R_{l(\kappa)} + R_\kappa + R_{q(\kappa)} &\leq f_4(Q) + f_5(Q) \\
R_\kappa + R_{q(\kappa)} + R_{l(\kappa)} &\leq f_4(Q) + f_7(Q) \\
2R_\kappa + R_{q(\kappa)} + R_{l(\kappa)} &\leq f_6(Q) + f_7(Q) + f_8(Q) \\
2R_{q(\kappa)} + R_{l(\kappa)} + R_\kappa &\leq f_4(Q) + f_9(Q) \\
2R_{l(\kappa)} + R_\kappa + R_{q(\kappa)} &\leq f_4(Q) + f_5(Q) + f_7(Q).
\end{aligned}$$

Thus, $|\mathcal{Q}| \leq 9$.

The bounds on $\mathcal{U}_k$, $k \in \mathcal{K}$, also follow from the Support Lemma. First, write the RHSs of (8.3) as differences of entropies for some $\kappa \in \mathcal{K}$:

$$\begin{aligned}
A_\kappa &= H\big(Y_{q(\kappa)}|U_\kappa, U_{l(\kappa)}, X_{q(\kappa)}, Q\big) - H\big(Y_{q(\kappa)}|X_\kappa, U_{l(\kappa)}, X_{q(\kappa)}, Q\big)\\
B_\kappa &= H\big(Y_{q(\kappa)}|U_{l(\kappa)}, X_{q(\kappa)}, Q\big) - H\big(Y_{q(\kappa)}|X_\kappa, U_{l(\kappa)}, X_{q(\kappa)}, Q\big)\\
C_\kappa &= H\big(Y_{q(\kappa)}|U_\kappa, X_{q(\kappa)}, Q\big) - H\big(Y_{q(\kappa)}|X_\kappa, U_{l(\kappa)}, X_{q(\kappa)}, Q\big)\\
D_\kappa &= H\big(Y_{q(\kappa)}|X_{q(\kappa)}, Q\big) - H\big(Y_{q(\kappa)}|X_\kappa, U_{l(\kappa)}, X_{q(\kappa)}, Q\big)
\end{aligned}$$

$$\begin{aligned}
A_{q(\kappa)} &= H\big(Y_{l(\kappa)}|U_{q(\kappa)}, U_\kappa, X_{l(\kappa)}, Q\big) - H\big(Y_{l(\kappa)}|X_{q(\kappa)}, U_\kappa, X_{l(\kappa)}, Q\big)\\
B_{q(\kappa)} &= H\big(Y_{l(\kappa)}|U_\kappa, X_{l(\kappa)}, Q\big) - H\big(Y_{l(\kappa)}|X_{q(\kappa)}, U_\kappa, X_{l(\kappa)}, Q\big)\\
C_{q(\kappa)} &= H\big(Y_{l(\kappa)}|U_{q(\kappa)}, X_{l(\kappa)}, Q\big) - H\big(Y_{l(\kappa)}|X_{q(\kappa)}, U_\kappa, X_{l(\kappa)}, Q\big)\\
D_{q(\kappa)} &= H\big(Y_{l(\kappa)}|X_{l(\kappa)}, Q\big) - H\big(Y_{l(\kappa)}|X_{q(\kappa)}, U_\kappa, X_{l(\kappa)}, Q\big)
\end{aligned}$$

$$\begin{aligned}
A_{l(\kappa)} &= H\big(Y_\kappa|U_{l(\kappa)}, U_{q(\kappa)}, X_\kappa, Q\big) - H\big(Y_\kappa|X_{l(\kappa)}, U_{q(\kappa)}, X_\kappa, Q\big)\\
B_{l(\kappa)} &= H\big(Y_\kappa|U_{q(\kappa)}, X_\kappa, Q\big) - H\big(Y_\kappa|X_{l(\kappa)}, U_{q(\kappa)}, X_\kappa, Q\big)\\
C_{l(\kappa)} &= H\big(Y_\kappa|U_{l(\kappa)}, X_\kappa, Q\big) - H\big(Y_\kappa|X_{l(\kappa)}, U_{q(\kappa)}, X_\kappa, Q\big)\\
D_{l(\kappa)} &= H(Y_\kappa|X_\kappa, Q) - H\big(Y_\kappa|X_{l(\kappa)}, U_{q(\kappa)}, X_\kappa, Q\big)
\end{aligned}$$

Note that only 5 different entropies depend upon U_κ. The remaining entropies are preserved as long as $p(x_\kappa|q)$ is preserved. However, since in $\mathcal{R}^c$ we only have the combinations $A_\kappa + C_{q(\kappa)}$, $A_\kappa + D_{q(\kappa)}$, $C_\kappa + C_{q(\kappa)}$, and $C_\kappa + D_{q(\kappa)}$, we only have the following four functions of U_κ:

$$\begin{aligned}
g_1(U_\kappa) &= H\big(Y_{q(\kappa)}|U_\kappa, U_{l(\kappa)}, X_{q(\kappa)}, Q\big) - H\big(Y_{l(\kappa)}|X_{q(\kappa)}, U_\kappa, X_{l(\kappa)}, Q\big)\,,\\
g_2(U_\kappa) &= H\big(Y_{q(\kappa)}|U_\kappa, X_{q(\kappa)}, Q\big) - H\big(Y_{l(\kappa)}|X_{q(\kappa)}, U_\kappa, X_{l(\kappa)}, Q\big)\,,
\end{aligned}$$

and $A_{q(\kappa)}$ and $B_{q(\kappa)}$. Then, by the Support Lemma, we can find a U'_κ taking at most $|\mathcal{X}_\kappa| + 3$ values such that

$$\begin{aligned}
I\big(X_{q(\kappa)}; Y_{l(\kappa)}|U_{q(\kappa)}, U_\kappa, X_{l(\kappa)}, Q\big) &= I\big(X_{q(\kappa)}; Y_{l(\kappa)}|U_{q(\kappa)}, U'_\kappa, U_{l(\kappa)}, X_{l(\kappa)}, Q\big)\\
I\big(X_{q(\kappa)}; Y_{l(\kappa)}|U_\kappa, X_{l(\kappa)}, Q\big) &= I\big(X_{q(\kappa)}; Y_{l(\kappa)}|U'_\kappa, X_{l(\kappa)}, Q\big)\\
g_1(U_\kappa) &= g_1(U'_\kappa)\\
g_2(U_\kappa) &= g_2(U'_\kappa)\\
p(x_\kappa|q) &= \sum\nolimits_{u'_\kappa} p(u'_\kappa|q)p(x_\kappa|u'_\kappa, q)
\end{aligned}$$

for all $x_\kappa = 1, \ldots, |\mathcal{X}_\kappa| - 1$ and hence for $x_\kappa = |\mathcal{X}_\kappa|$. Thus, the achievable rate region in Theorem 8.3 does not change when we substitute U_κ by U'_κ.

This completes the proof of Theorem 8.3.

8.2 Gaussian Channels

Consider a continuous version of the DM-MWRC in Section 8.1 with 3-users, AF relaying, and circularly-symmetric complex Gaussian noise with zero mean. Under the assumption of quasi-static block flat fading and with single transmit and receive antennas, the channel outputs in transmission time $i \in \{1, 2, \dots, n\}$ at the relay and users are

$$y_{0i} = \sum_{k \in \mathcal{K}} h_k x_{ki} + z_{0i}, \qquad y_{ki} = g_k x_{0i} + z_{ki}, \; k \in \mathcal{K}, \tag{8.10}$$

where $h_k, g_k \in \mathbb{C}$ are the up- and downstream channel coefficients of user k, x_0 and x_k, $k \in \mathcal{K}$, is the complex-valued channel input of the relay and transmitter k, respectively, having an average transmit power $P_k = \sum_{i=1}^{n} |x_{ki}|^2$, and $\{z_{ki}\}_i$, $k \in \mathcal{K} \cup \{0\}$, is a circularly-symmetric white Gaussian noise process with zero mean and average power σ_k^2, independent of the channel inputs and other noise processes. The channel inputs are subject to an average power constraint

$$\sum_{i=1}^{n} |x_{ki}|^2 \leq n\bar{P}_k, \quad k \in \mathcal{K} \cup \{0\}.$$

Amplify-and-forward relaying is a special case of instantaneous relaying where the relaying function f is a linear function, i.e.,

$$x_{0i} = \alpha y_{0i}. \tag{8.11}$$

The constant α must be chosen such that x_{0i} satisfies the average power constraint. A sufficient condition follows directly from the power constraint.

8.5 Lemma. The probability of violating the relay power constraint approaches zero as the block length $n \to \infty$ if codewords are generated as mutually independent i.i.d. random sequences, and

$$\alpha \leq \sqrt{\frac{\bar{P}_0}{\sigma_0^2 + \sum_{k \in \mathcal{K}} |h_k|^2 P_k}}. \qquad \triangleleft$$

Proof. Consider the average power constraint at the relay, i.e.,

$$n\bar{P}_0 \geq \sum_{i=1}^{n} |\alpha y_{0i}|^2 = \alpha^2 \sum_{i=1}^{n} \left| \sum_{k \in \mathcal{K}} h_k x_{ki} + z_{0i} \right|^2.$$

Thus, α needs to satisfy $\alpha \leq \bar{\alpha}$ with

$$\frac{1}{\bar{\alpha}^2} = \frac{1}{n\bar{P}_0} \sum_{i=1}^{n} \Big[\sum_k |h_k|^2 |x_{ki}|^2 + |z_{0i}|^2 + \sum_{k,j\neq k} h_k h_j^* x_{ki} x_{ji}^* + \sum_k (h_k^* z_{0i} x_{ki}^* + h_k z_{0i}^* x_{ki}) \Big]$$

By the LLN [36, Thm. 1.7.4],

$$\frac{1}{n} \sum_{i=1}^{n} |z_{0i}|^2 \to \sigma_0^2 \quad \text{in probability as} \quad n \to \infty.$$

By assumption, the symbols x_{ki} are generated as i.i.d. random variables, independent of x_{ji} for all $j \neq k$ and i. Thus, by the LLN,

$$\frac{1}{n} \sum_{i=1}^{n} |x_{ki}|^2 \to P_k \quad \text{in probability as} \quad n \to \infty,$$

where P_k is the power of X_k, and

$$\frac{1}{n} \sum_{i=1}^{n} x_{ki} x_{ji}^* \to 0 \quad \text{in probability as} \quad n \to \infty,$$

for $j \neq k$. Thus,

$$\frac{1}{\bar{\alpha}^2} \to \frac{\sigma_0^2 + \sum_k |h_k|^2 P_k}{\bar{P}_0} \quad \text{in probability as} \quad n \to \infty$$

and the probability of violating the relay power constraint approaches zero as $n \to \infty$ if

$$\alpha \leq \sqrt{\frac{\bar{P}_0}{\sigma_0^2 + \sum_k |h_k|^2 P_k}}. \qquad \square$$

8.2.1 Achievable Rate Regions

The achievability results for discrete channels in Section 8.1 can be adapted to Gaussian channels in a two step process:

1. Modify the rate region for discrete memoryless channels to include a cost function as in [29, §3.3].
2. Adapt to continuous channels with the standard discretization argument in [29, §3.4.1].

For well-behaved continuous channels, and especially Gaussian channels, this procedure is just a technicality and extending it beyond PTP channels does not provide any insight. Hence, formal proofs are not provided and the interested reader is referred to [29, §3.3–3.4].

Simultaneous Non-Unique Decoding

Define the transmit SNR $S_k = \frac{P_k}{\sigma_0^2}$, the maximum transmit SNR $\bar{S}_k = \frac{\bar{P}_k}{\sigma_0^2}$, and the corresponding vectors $\boldsymbol{S} = (S_1, S_2, S_3)$ and $\bar{\boldsymbol{S}} = (\bar{S}_1, \bar{S}_2, \bar{S}_3)$.

8.6 Proposition (SND). A rate triple (R_1, R_2, R_3) is achievable for the Gaussian MWRC with AF and SND if, for each $k \in \mathcal{K}$,

$$R_k \leq \mathrm{C}\left(\frac{|h_k|^2 S_k}{\left|h_{l(k)}\right|^2 S_{l(k)} + \delta_k(\boldsymbol{S})}\right) \tag{8.12}$$

or

$$R_k \leq \mathrm{C}\left(\frac{|h_k|^2 S_k}{\delta_k(\boldsymbol{S})}\right) \tag{8.13a}$$

$$R_k + R_{l(k)} \leq \mathrm{C}\left(\frac{|h_k|^2 S_k + \left|h_{l(k)}\right|^2 S_{l(k)}}{\delta_k(\boldsymbol{S})}\right) \tag{8.13b}$$

where $S_k \leq \bar{S}_k$, and

$$\delta_k(\boldsymbol{S}) = 1 + \frac{\sigma_{q(k)}^2}{\left|g_{q(k)}\right|^2 \bar{P}_0}\left(1 + \sum_{i \in \mathcal{K}} |h_i|^2 S_i\right). \qquad \triangleleft$$

Proof. Assume Gaussian channel inputs, i.e., let $X_k \sim \mathcal{CN}(0, P_k)$ for all $k \in \mathcal{K}$, and adapt Theorem 8.1 to continuous channels using the standard discretization procedure in [29, §3.4.1]. Evaluate the adapted rate region in Theorem 8.1 for (8.10), (8.11), $Q = \emptyset$, and Gaussian codebooks to obtain (8.12) and (8.13) but with

$$\delta_k(\boldsymbol{S}) = 1 + \frac{\sigma_{q(k)}^2}{\left|g_{q(k)}\right|^2} \frac{1}{\sigma_0^2} \alpha^{-2}.$$

This is a decreasing function of α. Hence, the RHSs of the rate expressions are all increasing with α. Thus, the largest rate region is obtained by setting α to its upper limit. With Lemma 8.5, we obtain

$$\begin{aligned}\delta_k(\boldsymbol{S}) &= 1 + \frac{\sigma_{q(k)}^2}{\left|g_{q(k)}\right|^2} \frac{1}{\sigma_0^2} \left(\frac{\sigma_0^2 + \sum_k |h_k|^2 P_k}{\bar{P}_0} \right) \\ &= 1 + \frac{\sigma_{q(k)}^2}{\left|g_{q(k)}\right|^2} \left(\frac{1 + \sum_k |h_k|^2 S_k}{\bar{P}_0} \right). \qquad \square\end{aligned}$$

Rate Splitting and Superposition Coding

We first evaluate Theorem 8.3 under the assumption of Gaussian codebooks in the next lemma. Subsequently, we specialize this result to the case without coded time sharing.

8.7 Lemma. A rate triple (R_1, R_2, R_3) is achievable for the Gaussian MWRC with AF relaying if

$$R_k \leq \mathbb{E}_Q \left[\mathrm{C} \left(\frac{|h_k|^2 (\lambda_{kQ} + \bar{\lambda}_{kQ}) \bar{P}_k}{\left|h_{l(k)}\right|^2 \lambda_{l(k)Q} \bar{P}_{l(k)} + \sigma_0^2 + \sigma_{q(k)}^2 \left|g_{q(k)}\right|^{-2} \alpha^{-2}} \right) \right], \tag{8.14a}$$

$$\begin{aligned} R_k + R_{q(k)} \leq \mathbb{E}_Q \Bigg[&\mathrm{C} \left(\frac{|h_k|^2 \lambda_{kQ} \bar{P}_k}{\left|h_{l(k)}\right|^2 \lambda_{l(k)Q} \bar{P}_{l(k)} + \sigma_0^2 + \sigma_{q(k)}^2 \left|g_{q(k)}\right|^{-2} \alpha^{-2}} \right) \\ &+ \mathrm{C} \left(\frac{\left|h_{q(k)}\right|^2 (\lambda_{q(k)Q} + \bar{\lambda}_{q(k)Q}) \bar{P}_{q(k)} + |h_k|^2 \bar{\lambda}_{kQ} \bar{P}_k}{|h_k|^2 \lambda_{kQ} \bar{P}_k + \sigma_0^2 + \sigma_{l(k)}^2 \left|g_{l(k)}\right|^{-2} \alpha^{-2}} \right) \Bigg] \end{aligned} \tag{8.14b}$$

$$
\begin{aligned}
R_\Sigma \le \mathbb{E}_Q\Bigg[&\mathrm{C}\left(\frac{|h_k|^2\lambda_{kQ}\bar{P}_k}{\left|h_{l(k)}\right|^2\lambda_{l(k)Q}\bar{P}_{l(k)}+\sigma_0^2+\sigma_{q(k)}^2\left|g_{q(k)}\right|^{-2}\alpha^{-2}}\right)\\
&+\mathrm{C}\left(\frac{\left|h_{q(k)}\right|^2\lambda_{q(k)Q}\bar{P}_{q(k)}+|h_k|^2\bar{\lambda}_{kQ}\bar{P}_k}{|h_k|^2\lambda_{kQ}\bar{P}_k+\sigma_0^2+\sigma_{l(k)}^2\left|g_{l(k)}\right|^{-2}\alpha^{-2}}\right)\\
&+\mathrm{C}\left(\frac{\left|h_{l(k)}\right|^2(\lambda_{l(k)Q}+\bar{\lambda}_{l(k)Q})\bar{P}_{l(k)}+\left|h_{q(k)}\right|^2\bar{\lambda}_{q(k)Q}\bar{P}_{q(k)}}{\left|h_{q(k)}\right|^2\lambda_{q(k)Q}\bar{P}_{q(k)}+\sigma_0^2+\sigma_k^2|g_k|^{-2}\alpha^{-2}}\right)\Bigg]
\end{aligned}
\tag{8.14c}
$$

$$
\begin{aligned}
R_k + R_\Sigma \le \mathbb{E}_Q\Bigg[&\mathrm{C}\left(\frac{|h_k|^2\lambda_{kQ}\bar{P}_k}{\left|h_{l(k)}\right|^2\lambda_{l(k)Q}\bar{P}_{l(k)}+\sigma_0^2+\sigma_{q(k)}^2\left|g_{q(k)}\right|^{-2}\alpha^{-2}}\right)\\
&+\mathrm{C}\left(\frac{\left|h_{q(k)}\right|^2\lambda_{q(k)Q}\bar{P}_{q(k)}+|h_k|^2\bar{\lambda}_{kQ}\bar{P}_k}{|h_k|^2\lambda_{kQ}\bar{P}_k+\sigma_0^2+\sigma_{l(k)}^2\left|g_{l(k)}\right|^{-2}\alpha^{-2}}\right)\\
&+\mathrm{C}\left(\frac{\left|h_{l(k)}\right|^2\lambda_{l(k)Q}\bar{P}_{l(k)}+\left|h_{q(k)}\right|^2\bar{\lambda}_{q(k)Q}\bar{P}_{q(k)}}{\left|h_{q(k)}\right|^2\lambda_{q(k)Q}\bar{P}_{q(k)}+\sigma_0^2+\sigma_k^2|g_k|^{-2}\alpha^{-2}}\right)\\
&+\mathrm{C}\left(\frac{|h_k|^2(\lambda_{kQ}+\bar{\lambda}_{kQ})\bar{P}_k+\left|h_{l(k)}\right|^2\bar{\lambda}_{l(k)Q}\bar{P}_{l(k)}}{\left|h_{l(k)}\right|^2\lambda_{l(k)Q}\bar{P}_{l(k)}+\sigma_0^2+\sigma_{q(k)}^2\left|g_{q(k)}\right|^{-2}\alpha^{-2}}\right)\Bigg]
\end{aligned}
\tag{8.14d}
$$

$$
\begin{aligned}
R_\Sigma \le \mathbb{E}_Q\Bigg[&\mathrm{C}\left(\frac{|h_1|^2\lambda_{1Q}\bar{P}_1+\left|h_{l(1)}\right|^2\bar{\lambda}_{l(1)Q}\bar{P}_{l(1)}}{\left|h_{l(1)}\right|^2\lambda_{l(1)Q}\bar{P}_{l(1)}+\sigma_0^2+\sigma_{q(1)}^2\left|g_{q(1)}\right|^{-2}\alpha^{-2}}\right)\\
&+\mathrm{C}\left(\frac{|h_2|^2\lambda_{2Q}\bar{P}_2+\left|h_{l(2)}\right|^2\bar{\lambda}_{l(2)Q}\bar{P}_{l(2)}}{\left|h_{l(2)}\right|^2\lambda_{l(2)Q}\bar{P}_{l(2)}+\sigma_0^2+\sigma_{q(2)}^2\left|g_{q(2)}\right|^{-2}\alpha^{-2}}\right)\\
&+\mathrm{C}\left(\frac{|h_3|^2\lambda_{3Q}\bar{P}_3+\left|h_{l(3)}\right|^2\bar{\lambda}_{l(3)Q}\bar{P}_{l(3)}}{\left|h_{l(3)}\right|^2\lambda_{l(3)Q}\bar{P}_{l(3)}+\sigma_0^2+\sigma_{q(3)}^2\left|g_{q(3)}\right|^{-2}\alpha^{-2}}\right)\Bigg]
\end{aligned}
\tag{8.14e}
$$

hold for all $k \in \mathcal{K}$, where

$$
\alpha^{-2} = \frac{\sigma_0^2 + \sum_{k\in\mathcal{K}} |h_k|^2 \bar{P}_k \mathbb{E}_Q[\lambda_{kQ} + \bar{\lambda}_{kQ}]}{\bar{P}_0},
$$

for some $\lambda_{kQ}, \bar{\lambda}_{kQ} \ge 0$, $k \in \mathcal{K}$, such that $\mathbb{E}_Q[\lambda_{kQ} + \bar{\lambda}_{kQ}] \le 1$ and pmf $p(q)$ with $|\mathcal{Q}| \le 9$. ◁

Proof. Using the standard procedure [29, §3.4.1] we can adapt Theorem 8.3 to the Gaussian MWRC with power constraint. This yields the same inner bounds as in Theorem 8.3 evaluated for a product input distribution with $\mathbb{E}[|X_k|^2] \le \bar{P}_k$, $k \in \mathcal{K} \cup \{0\}$.

Let $U_{kq} \sim \mathcal{CN}(0, \bar{\lambda}_{kq}\bar{P}_k)$ and $V_{kq} \sim \mathcal{CN}(0, \lambda_{kq}\bar{P}_k)$ for all $kq \in \mathcal{K} \times \mathcal{Q}$ with $\lambda_{kQ}, \bar{\lambda}_{kQ} \geq 0$. Then $X_{kq} = U_{kq} + V_{kq}$ is also circularly symmetric complex Gaussian with zero mean and variance $(\lambda_{kq} + \bar{\lambda}_{kq})\bar{P}_k$. With these choices, evaluation of the mutual information terms in Theorem 8.3 is straightforward and results in (8.14a)–(8.14e).

The power constraint at transmitter $k \in \mathcal{K}$ is met as long as $\mathbb{E}[|X_{kQ}|^2] = \bar{P}_k \mathbb{E}_Q[\lambda_{kQ} + \bar{\lambda}_{kQ}] \leq \bar{P}_k$, or, equivalently, $\mathbb{E}_Q[\lambda_{kQ} + \bar{\lambda}_{kQ}] \leq 1$. By virtue of Lemma 8.5, the relay amplification factor α must satisfy

$$\alpha^2 \leq \frac{\bar{P}_0}{\sigma_0^2 + \sum_{k\in\mathcal{K}} |h_k|^2 P_k}$$

with $P_k = \mathbb{E}_Q[|X_{kQ}|^2] = \bar{P}_k \mathbb{E}_Q[\lambda_{kQ} + \bar{\lambda}_{kQ}]$. Observe that the RHSs of the rate constraints (8.14a)–(8.14e) are all increasing with α^{-2}. Thus, the rate region is maximized for

$$\alpha^{-2} = \frac{\sigma_0^2 + \sum_{k\in\mathcal{K}} |h_k|^2 \bar{P}_k \mathbb{E}_Q[\lambda_{kQ} + \bar{\lambda}_{kQ}]}{\bar{P}_0}. \qquad \square$$

Resource allocation over the rate region in Lemma 8.7 involves $|\mathcal{Q}|\,(1+2\,|\mathcal{K}|) + |\mathcal{K}| = 66$ variables. A common assumption to simplify this problem is to neglect time sharing. Similar to Proposition 8.6, this rate region does not depend on the absolute values of P_k^c, P_k^p and σ_0^2 but on their ratio. Hence, we state the result below in terms of the SNR.

Define the cloud and satellite transmit SNRs as $S_k^c = \frac{P_k^c}{\sigma_0^2}$ and $S_k^p = \frac{P_k^p}{\sigma_0^2}$, respectively, their sum $S_k = S_k^c + S_k^p$, the maximum transmit SNR $\bar{S}_k = \frac{\bar{P}_k}{\sigma_0^2}$, and the corresponding vectors $\boldsymbol{S}^c = (S_1^c, S_2^c, S_3^c)$, $\boldsymbol{S}^p = (S_1^p, S_2^p, S_3^p)$, $\boldsymbol{S} = (\boldsymbol{S}^c, \boldsymbol{S}^p)$, and $\bar{\boldsymbol{S}} = (\bar{S}_1, \bar{S}_2, \bar{S}_3)$.

8.8 Proposition (Rate Splitting). A rate triple (R_1, R_2, R_3) is achievable for the Gaussian MWRC with AF relaying if, for all $k \in \mathcal{K}$,

$$\begin{aligned}
R_k &\leq B_k, \\
R_k + R_{q(k)} &\leq A_k + D_{q(k)}, \\
R_\Sigma &\leq A_k + C_{q(k)} + D_{l(k)}, \\
R_k + R_\Sigma &\leq A_k + C_{q(k)} + C_{l(k)} + D_k, \\
R_\Sigma &\leq C_1 + C_2 + C_3,
\end{aligned}$$

where

$$A_k = \mathrm{C}\left(\frac{|h_k|^2 S_k^p}{\gamma_k(\boldsymbol{S})}\right) \tag{8.15}$$

$$B_k = \mathrm{C}\left(\frac{|h_k|^2 (S_k^p + S_k^c)}{\gamma_k(\boldsymbol{S})}\right) \tag{8.16}$$

$$C_k = \mathrm{C}\left(\frac{|h_k|^2 S_k^p + \left|h_{l(k)}\right|^2 S_{l(k)}^c}{\gamma_k(\boldsymbol{S})}\right) \tag{8.17}$$

$$D_k = \mathrm{C}\left(\frac{|h_k|^2 (S_k^p + S_k^c) + \left|h_{l(k)}\right|^2 S_{l(k)}^c}{\gamma_k(\boldsymbol{S})}\right) \tag{8.18}$$

where $S_k^c + S_k^p \leq \bar{S}_k$ and

$$\gamma_k(\boldsymbol{S}) = 1 + \left|h_{l(k)}\right|^2 S_{l(k)}^p + \frac{\sigma_{q(k)}^2}{\left|g_{q(k)}\right|^2 \bar{P}_0}\left(1 + \sum_{i\in\mathcal{K}} |h_i|^2 (S_i^c + S_i^p)\right). \qquad \triangleleft$$

Proof. In Lemma 8.7, remove time sharing, i.e., set $|\mathcal{Q}| = 1$, and identify $P_k^c = \bar{\lambda}_k \bar{P}_k$ and $P_k^p = \lambda_k \bar{P}_k$. □

8.9 Remark. For the case of a linear relaying function, each terminal can subtract its self-interference from the received signal. The resulting effective channel is equivalent to a 3-user cyclic IC. The achievable rate region with HK coding for such a channel is evaluated in [155]. A similar result to Theorem 8.3 is derived in [155, Thm. 1], but only for AF relaying and without cardinality bounds. This region is then subsequently specialized to Gaussian channels but with a fixed power allocation that is sufficient to show that this region is within 1.5 bit of the capacity region for 3 users. ◁

8.3 Optimal Resource Allocation

In this section, we discuss the application of the global optimization algorithms from Part II to resource allocation in Gaussian MWRCs. Specifically, we consider maximization throughput and EE over the achievable rate regions computed in Section 8.2.

8.3.1 Simultaneous Non-Unique Decoding

The SND rate region in Proposition 8.6 is nonconvex in $\boldsymbol{S}$ since both the numerator and denominator in the logarithms on the RHSs of the rate constraints are functions of $\boldsymbol{S}$.

Recall from Remark 8.2 that this rate region is equivalent to

$$\mathcal{R}_{\mathrm{SND}} = \bigcup_{\boldsymbol{d} \in \{\mathrm{TIN}, \mathrm{JD}\}^{|\mathcal{K}|}} \bigcap_{k \in \mathcal{K}} \mathcal{R}_{k, d_k}.$$

where $\mathcal{R}_{k,\mathrm{TIN}}$ and $\mathcal{R}_{k,\mathrm{JD}}$ are the regions defined by (8.12) and (8.13), respectively. By virtue of Lemma 2.2,

$$\begin{aligned} &\max\{f(\boldsymbol{R}, \boldsymbol{S}) \mid \boldsymbol{R} \in \mathcal{R}_{\mathrm{SND}}(\boldsymbol{S})\} \\ &\qquad = \max_{\boldsymbol{d} \in \{\mathrm{TIN}, \mathrm{JD}\}^{|\mathcal{K}|}} \max \left\{ f(\boldsymbol{R}, \boldsymbol{S}) \mid \boldsymbol{R} \in \textstyle\bigcap_{k \in \mathcal{K}} \mathcal{R}_{k, d_k}(\boldsymbol{S}) \right\}, \end{aligned}$$

i.e., resource allocation problems over $\mathcal{R}_{\mathrm{SND}}$ can be split into $2^3 = 8$ individual global optimization problems. Each of these is easily identified as an instance of (2.7) with $\mathcal{P} = \emptyset$, $\mathcal{Q} = [\boldsymbol{R}_{\min}, \infty)$ and solvable with Algorithm 5 using the initial box $\mathcal{M}_0 = [\boldsymbol{0}, \bar{\boldsymbol{S}}]$. Example 5.5 discusses the application of Algorithm 5 to WSR maximization, and Example 5.6 to GEE optimization. For the WSR problem, the monotonic optimization based approach in Section 4.4 is also applicable.

Global Energy Efficiency Maximization

Maximizing the GEE as in Example 5.6, i.e., computing the solution to

$$\begin{cases} \max\limits_{\boldsymbol{R}, \boldsymbol{S}} & \dfrac{\sum_k R_k}{\boldsymbol{\phi}^T \sigma_0^2 \boldsymbol{S} + P_c} \\ \text{s.t.} & R_k \leq \mathrm{C}\left(\dfrac{|h_k|^2 S_k}{\delta_k(\boldsymbol{S}) + [d_k = \mathrm{TIN}] \left|h_{l(k)}\right|^2 S_{l(k)}} \right), \ k \in \mathcal{K} \\ & R_k + R_{l(k)} \leq \mathrm{C}\left(\dfrac{|h_k|^2 S_k + \left|h_{l(k)}\right|^2 S_{l(k)}}{\delta_k(\boldsymbol{S})} \right), \ \forall k : d_k = \mathrm{JD} \\ & \boldsymbol{R} \geq \underline{\boldsymbol{R}}, \quad \boldsymbol{S} \in [\boldsymbol{0}, \bar{\boldsymbol{S}}] \end{cases} \tag{8.19}$$

where $[\cdot]$ is the Iverson bracket defined as

$$[A] = \begin{cases} 1 & \text{if } A \text{ is true} \\ 0 & \text{otherwise,} \end{cases}$$

neglects the relay power and, thus, does not reflect the true network GEE. This user-centric approach to the GEE makes sense in many scenarios where the relay power consumption is irrelevant, e.g., in satellite networks (cf. Fig. 8.1c) where the satellite is typically powered by solar energy and does not contribute to the CO_2-footprint of the communication system. However, there are at least as many scenarios where it is sensible to include the relay into the GEE. For this case, first consider the SINR expression in the single-user constraint of (8.19),

$$\begin{aligned}
&\frac{|h_k|^2 S_k}{1 + \frac{\sigma^2_{q(k)}}{|g_{q(k)}|^2 \bar{P}_0} \big(1 + \sum_{i\in\mathcal{K}} |h_i|^2 S_i\big) + [d_k = \text{TIN}] |h_{l(k)}|^2 S_{l(k)}} \\
&\quad = \frac{|h_k|^2 S_k}{\frac{1}{\bar{P}_0}\Big(\bar{P}_0 + \frac{\sigma^2_{q(k)}}{|g_{q(k)}|^2} \big(1 + \sum_{i\in\mathcal{K}} |h_i|^2 S_i\big) + [d_k = \text{TIN}] |h_{l(k)}|^2 S_{l(k)} \bar{P}_0\Big)} \\
&\quad = \frac{|h_k|^2 S_k \bar{P}_0}{\bar{P}_0 + \frac{\sigma^2_{q(k)}}{|g_{q(k)}|^2} \big(1 + \sum_{i\in\mathcal{K}} |h_i|^2 S_i\big) + [d_k = \text{TIN}] |h_{l(k)}|^2 S_{l(k)} \bar{P}_0}.
\end{aligned}$$

Identifying

$$\delta'_k(\boldsymbol{S}, P_0) = \bar{P}_0 + \frac{\sigma^2_{q(k)}}{|g_{q(k)}|^2} \big(1 + \sum_{i\in\mathcal{K}} |h_i|^2 S_i\big)$$

the GEE maximization problem is

$$\left\{\begin{aligned}
&\max_{\boldsymbol{R}, \boldsymbol{S}, P_0} && \frac{\sum_k R_k}{\boldsymbol{\phi}^T \sigma_0^2 \boldsymbol{S} + \mu P_0 + P_c} \\
&\text{s.t.} && R_k \leq \mathrm{C}\left(\frac{|h_k|^2 S_k P_0}{\delta'_k(\boldsymbol{S}, P_0) + [d_k = \text{TIN}] \left|h_{l(k)}\right|^2 S_{l(k)} P_0} \right), \ k \in \mathcal{K} \\
& && R_k + R_{l(k)} \leq \mathrm{C}\left(\frac{(|h_k|^2 S_k + |h_{l(k)}|^2 S_{l(k)}) P_0}{\delta'_k(\boldsymbol{S}, P_0)} \right), \ \forall k : d_k = \text{JD} \\
& && \boldsymbol{R} \geq \underline{\boldsymbol{R}}, \quad \boldsymbol{S} \in [\boldsymbol{0}, \bar{\boldsymbol{S}}], P_0 \in [0, \bar{P}_0].
\end{aligned}\right. \tag{8.20}$$

This is an instance of (5.1) and solvable with Algorithm 5. We identify

$$f^+(\boldsymbol{R}) = \sum_k R_k \qquad f^-(\boldsymbol{S}, P_0) = \boldsymbol{\phi}^T \sigma_0^2 \boldsymbol{S} + \mu P_0 + P_c$$

$$g_k^+(\boldsymbol{R}, \boldsymbol{S}, P_0) = R_k + \log\Big(\delta'_k(\boldsymbol{S}, P_0) + [d_k = \text{TIN}] \left|h_{l(k)}\right|^2 S_{l(k)} P_0\Big)$$

$$g_k^-(\boldsymbol{S}, P_0) = \log\Big(|h_k|^2 S_k P_0 + \delta_k'(\boldsymbol{S}, P_0) + [d_k = \text{TIN}] \left|h_{l(k)}\right|^2 S_{l(k)} P_0\Big)$$
$$g_{|\mathcal{K}|+k}^+(\boldsymbol{R}, \boldsymbol{S}, P_0) = \left(R_k + R_{l(k)} + \log(\delta_k'(\boldsymbol{S}, P_0))\right) [d_k = \text{JD}]$$
$$g_{|\mathcal{K}|+k}^-(\boldsymbol{S}, P_0) = \Big(\log\Big((|h_k|^2 S_k + |h_{l(k)}|^2 S_{l(k)}) P_0 + \delta_k'(\boldsymbol{S}, P_0)\Big)\Big) [d_k = \text{JD}].$$

The global variables $\boldsymbol{x}$ are $(\boldsymbol{S}, P_0)$ and the nonglobals $\boldsymbol{\xi} = \boldsymbol{R}$. Clearly, each of the functions above is separable in the global and nonglobal variables and the domain $\mathcal{C} = [\boldsymbol{0}, \bar{\boldsymbol{S}}] \times [0, \bar{P}_0] \times [\boldsymbol{R}_{\min}, \infty)$. All functions are linear in the nonglobal variables $\boldsymbol{R}$. Moreover, since $\delta_k'(\boldsymbol{S}, P_0)$ is an increasing function and the objective is bounded below by zero, $\gamma f^-(\boldsymbol{S}, \boldsymbol{P}_0), g_{k,\boldsymbol{x}}^+(\boldsymbol{S}, \boldsymbol{P}_0), g_{|\mathcal{K}|+k,\boldsymbol{x}}^+(\boldsymbol{S}, P_0)$ are increasing functions and have a common minimizer over every box $\mathcal{M}$. The functions $g_k^-(\boldsymbol{S}, P_0), g_{|\mathcal{K}|+k}^-(\boldsymbol{S}, P_0)$ are also increasing and, thus, have a common maximizer over every box $\mathcal{M}$. Hence, Problem (8.20) with the above identifications is a Case 5.2 problem.

8.3.2 Rate Splitting and Superposition Coding

Globally optimal resource allocation for the achievable rate region in Proposition 8.8 is a straightforward extension of (2.7):

$$\begin{cases} \max\limits_{\boldsymbol{S},\boldsymbol{R}} & f(\boldsymbol{S}, \boldsymbol{R}) \\ \text{s.t.} & \boldsymbol{a}_i^T \boldsymbol{R} \leq \sum\limits_j \mathrm{C}\left(\dfrac{\boldsymbol{b}_{i,j}^T \boldsymbol{S}}{\gamma_{\kappa(i,j)}(\boldsymbol{S})}\right), \quad i = 1, \dots, n \\ & S_k^c + S_k^p \leq \bar{S}_k,\ k \in \mathcal{K}, \quad \boldsymbol{R} \geq \boldsymbol{0}, \quad \boldsymbol{S} \geq \boldsymbol{0} \end{cases} \tag{8.21}$$

where $\kappa(i,j)$ maps from (i, j) to the correct $k \in \mathcal{K}$ and $\boldsymbol{a}_i, \boldsymbol{b}_{i,j} \geq \boldsymbol{0}$ are easily identified from Proposition 8.8. From the discussion in Section 2.2 it is apparent that (8.21), depending on the identification of $g_i^+(\boldsymbol{S}, \boldsymbol{R})$ and $g_i^-(\boldsymbol{S})$, satisfies both, Cases 5.1 and 5.2, and is solvable with Algorithm 5 where the global and nonglobal variables are $\boldsymbol{S}$ and $\boldsymbol{R}$, respectively.

However, we can reduce the number of global variables in (8.21) from six to four: When using the identification $g_i^-(\boldsymbol{S}) = -\sum_j \log(\gamma_{\kappa(i,j)}(\boldsymbol{S}))$, the nonconvexity due to $\boldsymbol{S}^c$ stems only from the sum $\sum_{k\in\mathcal{K}} |h_k|^2 S_k^c$ in $\gamma_k(\boldsymbol{S})$. Thus, if we replace this sum by an auxiliary variable y, the variables $\boldsymbol{S}^c$ can be treated as nonglobal. The resulting problem is

$$\begin{cases} \max\limits_{\boldsymbol{S},\boldsymbol{R},y} & f(\boldsymbol{S},\boldsymbol{R}) \\ \text{s.t.} & \boldsymbol{a}_i^T\boldsymbol{R} \le \sum\limits_j \mathrm{C}\left(\dfrac{\boldsymbol{b}_{i,j}^T\boldsymbol{S}}{\gamma_{\kappa(i,j)}(\boldsymbol{S}^p, y)}\right), \quad i = 1,\dots,n \\ & y \ge \sum\limits_{k\in\mathcal{K}} |h_k|^2 S_k^c \\ & S_k^c + S_k^p \le \bar{S}_k,\ k\in\mathcal{K}, \quad \boldsymbol{R}\ge\boldsymbol{0}, \quad \boldsymbol{S}\ge\boldsymbol{0} \end{cases} \tag{8.22}$$

where $(\boldsymbol{S}^p, y)$ are the global variables and $(\boldsymbol{S}^c, \boldsymbol{R})$ are the nonglobal variables. Since the computational complexity grows exponentially with the number of global variables, solving (8.22) instead of (8.21) reduces the complexity considerably. The drawback of this approach is that Case 5.2 is no longer satisfied and the bounding problem becomes convex instead of linear. However, the performance gain significantly outweighs these, comparatively small, performances losses. The proposition below formally states the equivalence of (8.21) and (8.22).

8.10 Proposition. Problems (8.21) and (8.22) are equivalent in the sense that $v(8.21) = v(8.22)$ if $f(\boldsymbol{S},\boldsymbol{R})$ is increasing with $\boldsymbol{R}$ for fixed $\boldsymbol{S}$. $\triangleleft$

Proof. Let $\mathcal{F}$ be the feasible set of (8.21) without the first constraint. Then, we have

$$\begin{aligned} &\max_{(\boldsymbol{S},\boldsymbol{R})\in\mathcal{F}} \left\{ f(\boldsymbol{S},\boldsymbol{R}) \,\middle|\, \forall i : \boldsymbol{a}_i^T\boldsymbol{R} \le \sum_j \mathrm{C}\left(\frac{\boldsymbol{b}_{i,j}^T\boldsymbol{S}}{\gamma_{\kappa(i,j)}(\boldsymbol{S})}\right)\right\} \\ = &\max_{\substack{(\boldsymbol{S},\boldsymbol{R})\in\mathcal{F}\\ y=\sum_{k\in\mathcal{K}}|h_k|^2 S_k^c}} \left\{ f(\boldsymbol{S},\boldsymbol{R}) \,\middle|\, \forall i : \boldsymbol{a}_i^T\boldsymbol{R} \le \sum_j \mathrm{C}\left(\frac{\boldsymbol{b}_{i,j}^T\boldsymbol{S}}{\gamma_{\kappa(i,j)}(\boldsymbol{S}^p, y)}\right)\right\} \\ \le &\max_{\substack{(\boldsymbol{S},\boldsymbol{R})\in\mathcal{F}\\ y\ge\sum_{k\in\mathcal{K}}|h_k|^2 S_k^c}} \left\{ f(\boldsymbol{S},\boldsymbol{R}) \,\middle|\, \forall i : \boldsymbol{a}_i^T\boldsymbol{R} \le \sum_j \mathrm{C}\left(\frac{\boldsymbol{b}_{i,j}^T\boldsymbol{S}}{\gamma_{\kappa(i,j)}(\boldsymbol{S}^p, y)}\right)\right\}, \end{aligned}$$

since relaxing a constraint does not decrease the optimal value. Conversely, $\gamma_k(\boldsymbol{S}^p, y)$ is increasing with y and, thus, the RHSs of the constraints are decreasing in y. Since $f(\boldsymbol{S},\boldsymbol{R})$ and $\boldsymbol{a}_i^T\boldsymbol{R}$ are increasing with $\boldsymbol{R}$, the RHS should be as large as possible. Thus, the optimal y is as small as possible and

$$\max_{\substack{(\boldsymbol{S},\boldsymbol{R})\in\mathcal{F}\\ y\geq\sum_{k\in\mathcal{K}}|h_k|^2 S_k^c}} \left\{ f(\boldsymbol{S},\boldsymbol{R}) \,\middle|\, \forall i: \boldsymbol{a}_i^T\boldsymbol{R} \leq \sum_j \mathrm{C}\left(\frac{\boldsymbol{b}_{i,j}^T\boldsymbol{S}}{\gamma_{\kappa(i,j)}(\boldsymbol{S}^p, y)}\right)\right\}$$

$$\leq \max_{\substack{(\boldsymbol{S},\boldsymbol{R})\in\mathcal{F}\\ y=\sum_{k\in\mathcal{K}}|h_k|^2 S_k^c}} \left\{ f(\boldsymbol{S},\boldsymbol{R}) \,\middle|\, \forall i: \boldsymbol{a}_i^T\boldsymbol{R} \leq \sum_j \mathrm{C}\left(\frac{\boldsymbol{b}_{i,j}^T\boldsymbol{S}}{\gamma_{\kappa(i,j)}(\boldsymbol{S}^p, y)}\right)\right\}.$$

This establishes Proposition 8.10. □

Finally, note that all constraints in (8.22) except the first are linear and, thus, $\mathcal{C}$ is a convex set. The initial box $\mathcal{M}_0$ required by Algorithm 5 is easily identified as $[\mathbf{0}, \bar{\boldsymbol{S}}] \times [0, \sum_{k\in\mathcal{K}} |h_k|^2 \bar{S}_k]$.

8.4 Numerical Evaluation

We employ Algorithm 5 to maximize the throughput for SND and HK coding. Results are benchmarked against the monotonic optimization solution based on primal decomposition in Section 4.4. Then, Algorithm 5 is applied to GEE maximization for SND and both identifications in Example 5.6 are benchmarked against Dinkelbach's algorithm.

8.4.1 Throughput

We employ Algorithm 5 to compute the throughput optimal resource allocation for the rate regions in Propositions 8.6 and 8.8. Because the objective $f(\boldsymbol{S},\boldsymbol{R}) = \sum_i R_i$ is increasing with $\boldsymbol{R}$ it satisfies the condition in Proposition 8.10 and we can solve (8.22) instead of (8.21). We assume equal maximum power constraints and noise power at the users and the relay, and no minimum rate constraint, i.e. $\underline{\boldsymbol{R}} = \mathbf{0}$. Channels are assumed reciprocal and chosen i.i.d. with circular symmetric complex Gaussian distribution, i.e., $h_k \sim \mathcal{CN}(0,1)$ and $g_k = h_k^*$. Results are averaged over 1000 channel realizations. The precision of the objective value is $\eta = 0.01$, ε is chosen as 10^{-5}, and the algorithm is started with $\gamma = 0$.

Figure 8.7 displays the maximum throughput for SND as in Proposition 8.6 and RS as in Proposition 8.8. Recall that before [5], common wisdom was that neither TIN nor SND dominates the other rate-wise. Figure 8.7 also includes

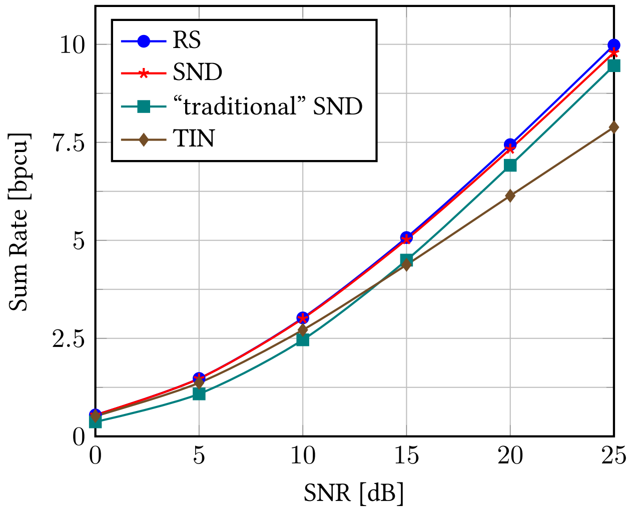


Figure 8.7: Throughput in the MWRC with AF relaying and 1) RS 2) SND; 3) "traditional" SND; and 4) TIN. Averaged over 1000 i.i.d. channel realizations [81]. ©2019 IEEE

results for "traditional" SND and TIN which are defined by the rate regions obtained from (8.12) and (8.13), respectively. First, observe that, in accordance with conventional wisdom, neither "traditional" SND nor TIN dominates the other. Instead, "extended" SND clearly dominates the other two where the gain is solely due to allowing each receiver to either use TIN or "traditional" SND. The average gain of SND over the other two is approximately 11 % and 22 % at 10 dB, respectively, or 0.29 bpcu and 0.54 bpcu. Note that this gain is only achieved by allowing each receiver to choose between TIN and JD which does not result in higher decoding complexity than "traditional" SND. The average gain observed for RS over single message SND is rather small, e.g., at 25 dB it is only 2 % (or 0.2 bpcu). However, depending on the channel realization gains of up to half a bit (or 4.6 %) at 20 dB have been observed. With spectrum being an increasingly scarce and expensive resource this occasional gain might very well justify the slightly higher complexity.

We use the primal decomposition approach from Section 4.4 as state-of-the-art reference for a performance comparison. The inner linear problem is solved with Gurobi 8 [38] and the monotonic program with the polyblock

Table 8.2: Mean and median run times of throughput maximization for SND and "traditional" SND. For "traditional" SND, both the SIT and the polyblock algorithm (PA) with primal decomposition as in Section 4.4 are employed [81]. ©2019 IEEE

		SNR		
		0 dB	15 dB	30 dB
SND	Mean	385.78 s	351.92 s	5335.48 s
	Median	75.51 s	29.16 s	2494.54 s
"trad." SND	Mean	19.77 s	6.23 s	3217.21 s
	Median	9.50 s	1.84 s	831.71 s
PA	Mean	192 426.53 s	77 670.36 s	10 602.06 s
	Median	46 210.42 s	7091.09 s	35.67 s

algorithm[32] in Algorithm 3 with a tolerance of 0.01. Results match those obtained with Algorithm 5 for "traditional" SND and, thus, are not displayed in Fig. 8.7. Run times for both algorithms are reported in Table 8.2. About 5.4 % of the computations with the polyblock algorithm did not complete within one week. For these, a run time of one week is assumed for the computation of the mean in Table 8.2. Thus, the reported run times for the polyblock algorithm are underestimated. Nevertheless, we observe, on average, roughly 10 000 times faster convergence for the SIT algorithm for low to medium SNRs. Interestingly, the median run time for high SNRs is lower for the polyblock algorithm while SIT is clearly faster on average.

8.11 Remark. If all users treat interference as noise (i.e., $d_k = \text{TIN}$ for all $k \in \mathcal{K}$), the rate variables can be eliminated from the optimization problem and MMP could be used. While the results in Fig. 8.7 would not change, the computation times in the first row of Table 8.2 would be even smaller. ◁

8.4.2 Energy Efficiency

The user-centric GEE, i.e., the GEE of only the users and without taking the relay energy consumption into account, as considered in (8.19) is computed with the same parameters as in Section 8.4.1 except for the precision which is chosen as

[32]The polyblock algorithm was implemented in C++ with similar techniques as in the implementation of Algorithm 5. Thus, performance differences should be mostly due to algorithmic differences.

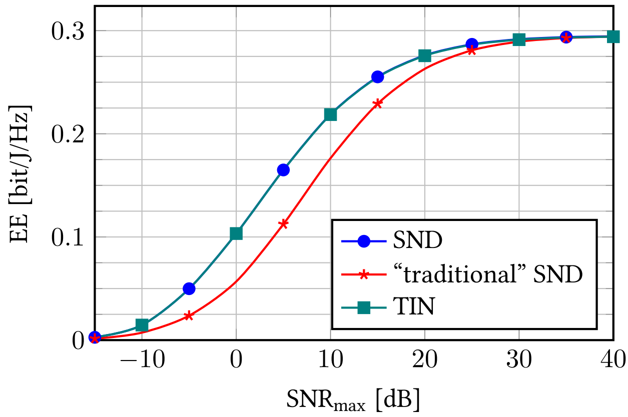


Figure 8.8: Energy efficiency in the MWRC with AF relaying and 1) SND; 2) "traditional" SND; and 3) TIN. Averaged over 1000 i.i.d. channel realizations and computed with a precision of $\eta = 10^{-3}$ [81]. ©2019 IEEE

$\eta = 10^{-3}$ for Fig. 8.8 and $\eta = 10^{-2}$ for Table 8.3. Additionally, we assume the static circuit power $P_c = 1$ W, the power amplifier inefficiencies $\phi_i = 4$, and that the relay always transmits at maximum power. Results for SND, "traditional" SND, and TIN are displayed in Fig. 8.8. First, observe that the curves saturate starting from 30 dB as is common for EE maximization. In this saturation region, all three approaches achieve the same EE. However, for lower SNRs, TIN and, thus also SND, outperform "traditional" SND by 24 % on average at 10 dB. Of course, as we have seen in the beginning of this chapter, EE performance depends quite a lot on the choice of real-world simulation parameters, so further work is necessary to draw final conclusions in this regard.

The main point of this subsection, however, is a performance comparison of different computational approaches to the EE computation. Recall from Example 5.6 that there are two possible identifications of g_i^+ and g_i^- that result in the problem either belonging to Case 5.1 or Case 5.2. As discussed before, identification (5.5) (resp. Case 5.1) leads to tighter bounds than (5.6) but requires the solution of a convex optimization problem in the bounding step. Instead, (5.6) (resp. Case 5.2) leads to a linear bounding problem which is considerably easier to solve. Table 8.3 summarizes mean and median computation times for both approaches. Each bounding problem is solved with the fastest solver available: the convex problem that stems from Case 5.1 with Mosek [84] and the linear from

Table 8.3: Mean and median run times of EE computation for "traditional" SND and different solvers, all with precision $\eta = 0.01$ [81]. ©2019 IEEE

		SNR		
		0 dB	20 dB	40 dB
Gurobi	Mean	5.1438 s	0.1771 s	0.1550 s
	Median	3.2781 s	0.0762 s	0.0600 s
Mosek	Mean	15.0453 s	0.1368 s	0.0323 s
	Median	10.2756 s	0.0219 s	0.0190 s
Dinkelbach	Mean	377.1501 s	145.4181 s	36.9690 s
	Median	162.8110 s	23.0270 s	16.9229 s

Case 5.2 with Gurobi [38]. Observe, that the Case 5.2 identification, labelled as "Gurobi", is, on average, three times as fast as the Case 5.1 identification ("Mosek") at 0 dB and Mosek is almost five times faster than Gurobi at 40 dB. However, these are only relative numbers. The average total computation time per channel realization is 8 s for Gurobi and 19 s for Mosek. So, despite the faster convergence speed of Case 5.1 (i.e., Mosek), Case 5.2 is much faster due to the lower computational complexity of the linear bounding problem.

The state-of-the-art approach to compute the EE is to combine fractional programming with monotonic optimization. This could be achieved by combining Dinkelbach's algorithm with the primal decomposition approach in Section 4.4. This generalizes the fractional monotonic programming framework in Section 4.3. However, we have already established in the previous section that the SIT approach is much faster than the primal decomposition in Section 4.4. Since the auxiliary problem to be solved in every step of Dinkelbach's algorithm is very similar to the throughput maximization problem, the benchmark results in Table 8.2 are also applicable to Dinkelbach's auxiliary problem. Instead, we solve this inner problem with the fastest method available: the SIT algorithm developed in this paper (cf. Table 8.2). Hence, the differences in the run time are solely due to the use of Dinkelbach's algorithm. It can be observed from Table 8.3 that the inherent treatment of fractional objectives in the developed algorithm is always significantly faster (up to 800 times on average at 20 dB) than Dinkelbach's algorithm. Moreover, the obtained result is guaranteed to lie within an η-region around the true essential optimal value, which is not guaranteed for Dinkelbach's algorithm (cf. p. 57).

Part IV

Conclusions

Chapter 9

Open Topics

The goal when developing an optimization framework is that it finds application beyond the motivational problems. Apart from transmit power allocation in wireless systems, the frameworks developed in this thesis are also suitable for other resource allocation tasks, e.g., beamforming and precoding, energy harvesting, flight path planning for unmanned aerial vehicles (UAVs), or ultra-reliable low-latency communication. Of course, applications are not limited to resource allocation problems. In particular, every program solvable by monotonic optimization can benefit from the methods in this thesis.

Besides applications, there are several algorithmic improvements and extensions that should be considered for future work.

- Following the discussion in Remark 5.11, integrating the BB procedure with the SIT scheme requires a weaker termination criterion than could be used otherwise. An alternative would be to restart the BB part of the algorithm after every update of the incumbent γ and employ the stronger stopping rule. Although unlikely from a theoretical perspective, this might lead to faster overall convergence speed. This question is best answered through numerical experiments.

- We left the design of an efficient reduction procedure for Algorithm 5 open for future work. Depending on the implementation and problem at hand, it might speed up the convergence (cf. Remark 5.12).

- The bounding method of the MMP framework is also applicable to the SIT based algorithm in Chapter 5 and will most likely increase the convergence speed. Conversely, applying the SIT approach to the MMP framework would result in an algorithm with finite convergence that can handle generic MM constraints (cf. Section 6.3.1).

- While most optimization algorithms are limited by their computational complexity, the major bottleneck in the MMP algorithm is its memory consumption. By thoroughly optimizing the data structures that store the partition of the feasible set it should be possible to reduce the memory consumption by at least 50 %.

- While exploiting monotonicity in scalar variables is straightforward with MMP, this is not the case for vectors and matrices. Further work in this direction is necessary.
- Supporting integer and continuous variables in the same algorithm would help to solve, e.g., dynamic spectrum allocation problems. Since BB is the most widely employed method for discrete optimization problems it should be possible to combine MMP with integer programming.

Some other research directions left unexplored in this thesis and not directly related to the two developed frameworks are the following.

- Evaluating the PTP capacity region for K-user Gaussian ICs is challenging for two reasons: the individual power allocation problems and the combinatorial problem outlined below Lemma 2.2. While we have solved the first, it is safe to say that the second is considerably more demanding.
- In Chapter 7, we have shown that HK coding might provide better EE than PTP codes even under the limiting assumptions of proper Gaussian codebooks and no coded time sharing. It is expected that relaxing these assumptions leads to more significant gains than those obtained by pure Gaussian inputs.

Chapter 10

Conclusions

We have developed two novel global optimization frameworks that rely on partial monotonicity of the objective and constraints. They are applied to solve two different radio resource allocation problems in interference networks. Although we focused on power control problems, the developed methods are much more general, both in the context of RRM and more general engineering problems. In particular, every optimization problem that falls within the class of monotonic optimization problems is also solvable by our algorithms, most likely with faster convergence speed.

In the presentation of the underlying mathematical concepts we have put an emphasis on understandability. While the treatment often follows [128], it is more general and the concepts are arranged in a modular fashion. A notable contribution is the focus on numerical problems that might arise from complicated feasible sets, an issue currently not widely known in the communication systems and signal processing communities. We established ε-essential feasibility and the SIT approach as a remedy and important concept towards numerical stable global optimization algorithms. Based on this theory, we have built a novel global optimization framework tailored to resource allocation problems that inherently supports fractional objectives, avoids numerical problems with nonrobust feasible sets, and is several orders of magnitude faster than state-of-the-art algorithms. It also preserves the computational complexity in the number of nonglobal variables without the need for explicit primal decomposition. This is a major benefit, for example, when optimizing over rate regions where the rate variables are linear.

The second optimization framework, named mixed monotonic programming (MMP), generalizes monotonic optimization. At its core is a novel bounding mechanism accompanied by an efficient BB implementation that helps exploit partial monotonicity without requiring a DI reformulation. This often leads to tighter bounds and faster convergence. It also facilitates bounding of functions that are hard or impossible to express as DI functions. For example, fractional monotonic programs are usually solved by a combination of Dinkelbach's iterative algorithm and monotonic programming. This approach requires the solution of several global optimization problems and results in unnecessarily long computation times. Instead, the MMP bounding method makes it easy to obtain suitable bounds for fractional monotonic programs without the need for Dinkelbach's algorithm. Benchmark results show that MMP outperforms monotonic programming by several orders of magnitude, both in run time and memory consumption.

We have considered two detailed application scenarios, one for each framework. In the first, MMP is applied to evaluate the GEE gain RS might provide over PTP codes in 2-user GICs. The basis for this study is a comprehensive overview of sum capacity results for the GIC. These are subsequently used to establish that RS can only provide a benefit under moderate interference, a common scenario in multicell systems. The numerical results obtained with the MMP framework show that RS might offer a gain over PTP even under restrictive assumptions. The second scenario employs the SIT framework for WSR and GEE maximization in a 3-user MWRC. After motivating the use of AF relaying with some preliminary results from [83], we derived achievable rate regions for discrete memoryless channels with SND and HK coding under the assumption of instantaneous relaying. These results are then applied to Gaussian channels and the corresponding optimization problems are identified. The numerical results show that HK coding might provide a throughput benefit in certain circumstances and that TIN is the most energy-efficient transmission scheme. These experiments are also employed to benchmark the algorithm against the state-of-the-art. It turns out that it is four orders of magnitude faster than primal decomposition with monotonic optimization and has almost three orders of magnitude of additional speed-up over Dinkelbach's algorithm for fractional programs without a concave-convex structure.

With 5G being rolled out and 6G on the horizon, the developed methods are an important step towards solving the RRM challenges to come. In particular, heterogeneous systems and QoS requirements will lead to ever more complicated feasible sets that require numerically stable algorithms. Another important aspect is the ability to deal with an increasing number of variables due to more efficient algorithms. This enables researchers to solve more challenging problems and gain new insights.

Part V

Appendices

Appendix A

Implementation Details

All algorithms were implemented in C++17 and compiled with GCC 7.3. The code relies heavily on templates and STL functions. Linear optimization problems are solved with Gurobi 8 [38] and nonlinear problems with Mosek 8.1 [84] using the kindly provided academic licenses. Great care was taken to implement all algorithms with the same rigour to ensure fair benchmarks. Most of the source code is online available at `https://github.com/bmatthiesen`.

All computations were done on TU Dresden's Bull HPC-Cluster Taurus. Reported performance results were obtained on Intel Haswell nodes with Xeon E5-2680 v3 CPUs running at 2.50 GHz. The author thanks the Center for Information Services and High Performance Computing (ZIH) at TU Dresden for generous allocations of computer time.

List of Algorithms

Notation

Vectors

$\boldsymbol{a}$	Column vector
$(a_i)_{i=1}^n$	$(a_1, a_2, \ldots, a_n)^T$
$\mathbf{0}$	Null vector
$\boldsymbol{e}_i$	ith canonical unit vector
$\boldsymbol{a}^T$	Transpose of $\boldsymbol{a}$
$\boldsymbol{a} \leq \boldsymbol{b}$	Element-wise vector inequality: $\boldsymbol{a} \leq \boldsymbol{b} \Leftrightarrow \forall i : a_i \leq b_i$
$\boldsymbol{a} < \boldsymbol{b}$	Strict element-wise vector inequality: $\boldsymbol{a} < \boldsymbol{b} \Leftrightarrow \forall i : a_i < b_i$
$\|a\|$	ℓ_2-norm of $\boldsymbol{a}$
$\dim \boldsymbol{a}$	Dimension of $\boldsymbol{a}$, i.e., the number of elements in $\boldsymbol{a}$

Sets

$\mathcal{A}$	Set
$\mathscr{A}$	Set of sets
$\mathbb{R}$	Real numbers
$\mathbb{R}_{\geq 0}$	Non-negative real numbers
$\mathbb{C}$	Complex numbers
$\emptyset$	Empty set
$\{a_i\}_{i \in \mathcal{I}}$	Sequence of a_i, $i \in \mathcal{I}$
$[\boldsymbol{a}, \boldsymbol{b}]$	Closed interval. A hypercube (box) if $\dim \boldsymbol{a} > 1$. Defined as $[\boldsymbol{a}, \boldsymbol{b}] = \{\boldsymbol{x} \mid \boldsymbol{a} \leq \boldsymbol{x} \leq \boldsymbol{b}\}$
$(\boldsymbol{a}, \boldsymbol{b})$	Open interval. An open hypercube (box) if $\dim \boldsymbol{a} > 1$. Defined as $(\boldsymbol{a}, \boldsymbol{b}) = \{\boldsymbol{x} \mid \boldsymbol{a} < \boldsymbol{x} < \boldsymbol{b}\}$

Set Operations

$\|\mathcal{A}\|$	Cardinality of $\mathcal{A}$
$\operatorname{cl} \mathcal{A}$	Closure of $\mathcal{A}$
$\operatorname{int} \mathcal{A}$	Interior of $\mathcal{A}$
$\mathcal{A}^*$	Robust version of $\mathcal{A}$ (cf. Section 3.2.1)
$\operatorname{diam} \mathcal{A}$	Diameter of $\mathcal{A}$: maximum distance between two points in $\mathcal{A}$
$\partial^+ \mathcal{A}$	Pareto boundary of $\mathcal{A}$ (cf. p. 22)
$\operatorname{proj}_{\boldsymbol{x}} \mathcal{A}$	Projection of $\mathcal{A}$ onto the $\boldsymbol{x}$ coordinates: $\operatorname{proj}_{\boldsymbol{x}} \mathcal{A} = \{\boldsymbol{x} \mid (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{A} \text{ for some } \boldsymbol{y}\}$
$\mathcal{A}_{\boldsymbol{x}}$	$\boldsymbol{x}$-section of $\mathcal{A}$: $\mathcal{A}_{\boldsymbol{x}} = \{\boldsymbol{y} \mid (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{A}\}$

Mathematical Operators and Functions

$\lvert a\rvert$	Absolute value of a
$x \to y$	x approaches y
$v(1.1)$	Optimal value of optimization problem in equation (1.1)
$\mathrm{C}(x)$	$\log(1+x)$
$O(g(n))$	Asymptotically upper bounded by $g(n)$: $O(g(n)) = \{f(n) \mid \exists c > 0 : \forall n \geq n_0 : 0 \leq f(n) \leq cg(n)\}$
$\Omega(g(n))$	Asymptotically lower bounded by $g(n)$: $\Omega(g(n)) = \{f(n) \mid \exists c > 0 : \forall n \geq n_0 : 0 \leq cg(n) \leq f(n)\}$
$[A]$	Iverson bracket: 1 if A is true, 0 otherwise

Functions and Derivatives

$f : \mathcal{A} \to \mathcal{B}$	Function f from $\mathcal{A}$ to $\mathcal{B}$
$\boldsymbol{x} \mapsto f(\boldsymbol{x})$	Function that maps $\boldsymbol{x}$ to $f(\boldsymbol{x})$
$\frac{d}{dx} f$	Derivative of f with respect to x
∇f	Gradient of f
$C[\boldsymbol{a}, \boldsymbol{b}]$	Space of continuous real-valued functions on $[\boldsymbol{a}, \boldsymbol{b}]$ equipped with supremum norm

Probability

$\Pr\{\mathcal{A}\}$	Probability of event $\mathcal{A}$
$\mathbb{E}_{\boldsymbol{X}}[g(\boldsymbol{X})]$	Expected value of $g(\boldsymbol{X})$ with respect to random vector $\boldsymbol{X}$
$\mathbb{E}[g(\boldsymbol{X})]$	Short version of $\mathbb{E}_{\boldsymbol{X}}[g(\boldsymbol{X})]$
$p(\boldsymbol{x})$	Pmf of random vector $\boldsymbol{X}$ with argument $\boldsymbol{x}$: $p(\boldsymbol{x}) = \Pr\{\boldsymbol{X} = \boldsymbol{x}\}$
$p(\boldsymbol{x} \mid \boldsymbol{y})$	Collection of conditional pmfs of $\boldsymbol{X}$, one for every $\boldsymbol{y}$ in the support of the random vector $\boldsymbol{Y}$
$\mathcal{CN}(\mu, \sigma^2)$	Circularly-symmetric complex Gaussian distribution with mean μ and variance σ^2

Information Theory

$\mathcal{T}_{\varepsilon}^{(n)}(\boldsymbol{X})$	Typical set of $\boldsymbol{X}$. Please refer to [29, §2.4–2.5]
$H(X)$	Entropy of X (X discrete)
$H(X\lvert Y)$	Conditional entropy of X given Y (X and Y discrete)
$I(X;Y)$	Mutual information between X and Y
$I(X;Y\lvert Z)$	Conditional Mutual information between X and Y given Z

Abbreviations

ADC	analog-to-digital converter
AF	amplify-and-forward
BB	branch-and-bound
BC	broadcast channel
bpcu	bit per channel use
BS	base station
DAC	digital-to-analog converter
DC	difference of convex
DF	decode-and-forward
DI	difference of increasing
DM-IC	discrete memoryless interference channel
DM-MWRC	discrete memoryless multi-way relay channel
DSL	digital subscriber line
DSP	digital signal processing
EE	energy efficiency
GEE	global energy efficiency
GIC	Gaussian interference channel
G-PTP	Gaussian point-to-point
HK	Han-Kobayashi
IC	interference channel
i.i.d.	independent and identically distributed
JD	joint decoding
KKT	Karush-Kuhn-Tucker
LLN	law of large numbers
l.s.c.	lower semi-continuous

MAC	multiple-access channel
MIMO	multiple-input multiple-output
MISO	multiple-input single-output
MM	mixed monotonic
MMP	mixed monotonic programming
MOP	multi-objective optimization problem
MWRC	multi-way relay channel
NNC	noisy network coding
NOMA	non-orthogonal multiple access
OFDMA	orthogonal frequency-division multiple access
pmf	probability mass function
PTP	point-to-point
QoS	Quality of Service
RHS	right-hand side
RRM	radio resource management
RS	rate splitting
SINR	signal to interference plus noise ratio
SIT	successive incumbent transcending
SND	simultaneous non-unique decoding
SNR	signal-to-noise ratio
s. t.	subject to
STL	standard template library
TIN	treating interference as noise
UAV	unmanned aerial vehicle

u.s.c.	upper semi-continuous
WLOG	without loss of generality
WSR	weighted sum rate

Bibliography

[1] E. Abbe and L. Zheng, “A coordinate system for Gaussian networks”, *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 721–733, Feb. 2012.

[2] Y. Almogy and O. Levin, “A class of fractional programming problems”, *Operations Res.*, vol. 19, no. 1, pp. 57–67, Feb. 1971.

[3] V. S. Annapureddy and V. V. Veeravalli, “Gaussian interference networks: Sum capacity in the low-interference regime and new outer bounds on the capacity region”, *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3032–3050, Jul. 2009.

[4] F. Baccelli, A. El Gamal, and D. N. C. Tse, “Interference networks with point-to-point codes”, *IEEE Trans. Inf. Theory*, vol. 57, no. 5, pp. 2582–2596, Apr. 2011.

[5] B. Bandemer, A. El Gamal, and Y.-H. Kim, “Optimal achievable rates for interference networks with random codes”, *IEEE Trans. Inf. Theory*, vol. 61, no. 12, pp. 6536–6549, Oct. 2015.

[6] J. Barros and M. R. D. Rodrigues, “Secrecy capacity of wireless channels”, in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Seattle, WA, USA, Jul. 2006.

[7] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, ser. MPS-SIAM Ser. Optim. Philadelphia, PA, USA: SIAM, 2001.

[8] H. P. Benson, “On the convergence of two branch-and-bound algorithms for nonconvex programming problems”, *J. Optim. Theory Appl.*, vol. 36, no. 1, pp. 129–134, Jan. 1982.

[9] E. Björnson and E. A. Jorswieck, *Optimal Resource Allocation in Coordinated Multi-Cell Systems*, ser. FnT Commun. Inf. Theory 2–3. Boston, MA, USA: Now, 2013, vol. 9, pp. 113–381.

[10] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, “Five disruptive technology directions for 5G”, *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 74–80, Feb. 2014.

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

[12] J. Brehmer, *Utility Maximization in Nonconvex Wireless Systems*, ser. Found. Signal Process., Commun., Netw. New York; Berlin, Germany; Vienna, Austria: Springer-Verlag, 2012, vol. 5.

[13] G. Calcev, D. Chizhik, B. Goransson, *et al.*, “A wideband spatial channel model for system-wide simulations”, *IEEE Trans. Veh. Technol.*, vol. 56, no. 2, pp. 389–403, Mar. 2007.

[14] A. B. Carleial, “Interference channels”, *IEEE Trans. Inf. Theory*, vol. 24, no. 1, pp. 60–70, Jan. 1978.

[15] A. Chaaban and A. Sezgin, “The approximate capacity region of the Gaussian Y-channel via the deterministic approach”, *IEEE Trans. Inf. Theory*, vol. 61, no. 2, pp. 939–962, Feb. 2015.

[16] ——, *Multi-way Communications: An Information Theoretic Perspective*, ser. FnT Commun. Inf. Theory 3–4. Boston, MA, USA: Now, 2015, vol. 12, pp. 113–381.

[17] A. Charnes and W. W. Cooper, “Programming with linear fractional functionals”, *Nav. Res. Logistics Quart.*, vol. 9, no. 3–4, pp. 181–186, Sep. 1962.

[18] M. Chiang, “Nonconvex optimization for communication networks”, in *Advances in Applied Mathematics and Global Optimization: In Honor of Gilbert Strang*, ser. Adv. Mech. Math. D. Y. Gao and H. D. Sherali, Eds., vol. 17, New York; Berlin, Germany; Vienna, Austria: Springer-Verlag, 2009, ch. 5, pp. 137–196.

[19] H.-F. Chong, M. Motani, H. K. Garg, and H. El Gamal, “On the Han-Kobayashi region for the interference channel”, *IEEE Trans. Inf. Theory*, vol. 54, no. 7, pp. 3188–3195, Jul. 2008.

[20] B. Clerckx, H. Joudeh, C. Hao, M. Dai, and B. Rassouli, “Rate splitting for MIMO wireless networks: A promising PHY-layer strategy for LTE evolution”, *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 98–105, May 2016.

[21] M. Codreanu, A. Tölli, M. Juntti, and M. Latva-aho, “Joint design of Tx-Rx beamformers in MIMO downlink channel”, *IEEE Trans. Signal Process.*, vol. 55, no. 9, pp. 4639–4655, Sep. 2007.

[22] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press, 2009.

[23] T. M. Cover, “Broadcast channels”, *IEEE Trans. Inf. Theory*, vol. 18, no. 1, pp. 2–14, Jan. 1972.

[24] J.-P. Crouzeix and J. A. Ferland, “Algorithms for generalized fractional programming”, *Math. Program.*, vol. 52, no. 1–3, pp. 191–207, May 1991.

[25] J.-P. Crouzeix, J. A. Ferland, and S. Schaible, “An algorithm for generalized fractional programs”, *J. Optim. Theory Appl.*, vol. 47, no. 1, pp. 35–49, Sep. 1985.

[26] W. Dinkelbach, “On nonlinear fractional programming”, *Manage. Sci.*, vol. 13, no. 7, pp. 492–498, Mar. 1967.

[27] A. Dytso, D. Tuninetti, and N. Devroye, “On the two-user interference channel with lack of knowledge of the interference codebook at one receiver”, *IEEE Trans. Inf. Theory*, vol. 61, no. 3, pp. 1257–1276, Mar. 2015.

[28] ——, “Interference as noise: Friend or foe?”, *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3561–3596, Jun. 2016.

[29] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[30] R. H. Etkin, D. N. C. Tse, and H. Wang, “Gaussian interference channel capacity to within one bit”, *IEEE Trans. Inf. Theory*, vol. 54, no. 12, pp. 5534–5562, Dec. 2008.

[31] J. E. Falk and S. W. Palocsay, “Optimizing the sum of linear fractional functions”, in *Recent Advances in Global Optimization*, ser. Princeton Ser. Comput. Sci. C. A. Floudas and P. M. Pardalos, Eds., Princeton, NJ, USA: Princeton Univ. Press, 1992, pp. 221–258.

[32] J. E. Falk and R. M. Soland, “An algorithm for separable nonconvex programming problems”, *Manage. Sci.*, vol. 15, no. 9, pp. 550–569, May 1969.

[33] G. P. Fettweis, “The tactile internet: Applications and challenges”, *IEEE Veh. Technol. Mag.*, vol. 9, no. 1, pp. 64–70, Mar. 2014.

[34] G. P. Fettweis, M. Dörpinghaus, J. Castrillon, *et al.*, “Architecture and advanced electronics pathways toward highly adaptive energy- efficient computing”, *Proc. IEEE*, vol. 107, no. 1, pp. 204–231, Jan. 2019.

[35] R. W. Freund and F. Jarre, “Solving the sum-of-ratios problem by an interior-point method”, *J. Global Optim.*, vol. 19, no. 1, pp. 83–102, Jan. 2001.

[36] R. G. Gallager, *Stochastic Processes: Theory for Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[37] D. Gündüz, A. Yener, A. Goldsmith, and H. V. Poor, “The multiway relay channel”, *IEEE Trans. Inf. Theory*, vol. 59, no. 1, pp. 51–63, Jan. 2013.

[38] Gurobi Optimization. (2018). Gurobi Optimizer 8.0.1, [Online]. Available: `http://gurobi.com`.

[39] T. Han and K. Kobayashi, “A new achievable rate region for the interference channel”, *IEEE Trans. Inf. Theory*, vol. 27, no. 1, pp. 49–60, Jan. 1981.

[40] Z. Han and K. J. R. Liu, *Resource Allocation for Wireless Networks: Basics, Techniques, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[41] S. He, Y. Huang, S. Jin, and L. Yang, “Coordinated beamforming for energy efficient transmission in multicell multiuser systems”, *IEEE Trans. Commun.*, vol. 61, no. 12, pp. 4961–4971, Dec. 2013.

[42] S. He, Y. Huang, L. Yang, and B. Ottersten, “Coordinated multicell multiuser precoding for maximizing weighted sum energy efficiency”, *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 741–751, Feb. 2014.

[43] C. Hellings, “Reduced-entropy signals in MIMO communication systems”, PhD thesis, Technische Universität München, Jun. 2017.

[44] C. Hellings and W. Utschick, “Energy-efficient rate balancing in vector broadcast channels with linear transceivers”, in *Proc. Int. Symp. Wireless Commun. Syst. (ISWCS)*, Paris, France, Aug. 2012, pp. 1044–1048.

[45] R. Horst, “A new branch and bound approach for concave minimization problems”, in *Optimization Techniques Modeling and Optimization in the Service of Man Part 2*, ser. Lecture Notes Comput. Sci. J. Cea, Ed., vol. 41, New York; Berlin, Germany; Vienna, Austria: Springer-Verlag, Sep. 1976, pp. 330–336.

[46] ——, “An algorithm for nonconvex programming problems”, *Math. Program.*, vol. 10, no. 1, pp. 312–321, Dec. 1976.

[47] ——, “A note on the convergence of an algorithm for nonconvex programming problems”, *Math. Program.*, vol. 19, no. 1, pp. 237–238, Dec. 1980.

[48] ——, “A general class of branch-and-bound methods in global optimization with some new approaches for concave minimization”, *J. Optim. Theory Appl.*, vol. 51, no. 2, pp. 271–291, Nov. 1986.

[49] ——, “Deterministic global optimization with partition sets whose feasibility is not known: Application to concave minimization, reverse convex constraints, DC-programming, and Lipschitzian optimization”, *J. Optim. Theory Appl.*, vol. 58, no. 1, pp. 11–37, Jul. 1988.

[50] R. Horst and N. V. Thoai, “DC programming: Overview”, *J. Optim. Theory Appl.*, vol. 103, no. 1, pp. 1–43, Oct. 1999.

[51] R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*, 3rd, rev. and enl. ed. New York; Berlin, Germany; Vienna, Austria: Springer-Verlag, 1996.

[52] E. Hossain, M. Rasti, and L. B. Le, *Radio Resource Management in Wireless Networks: An Engineering Approach.* Cambridge, U.K.: Cambridge Univ. Press, 2017.

[53] C. Isheden, Z. Chong, E. A. Jorswieck, and G. P. Fettweis, “Framework for link-level energy efficiency optimization with informed transmitter”, *IEEE Trans. Wireless Commun.*, vol. 11, no. 8, pp. 2946–2957, Aug. 2012.

[54] M. Iwamura, H. Takahashi, and S. Nagata, “Relay technology in LTE-Advanced”, *NTT DOCOMO Tech. J.*, vol. 12, no. 2, pp. 29–36, Sep. 2010.

[55] N. Jindal, W. Rhee, S. Vishwanath, S. A. Jafar, and A. Goldsmith, “Sum power iterative water-filling for multi-antenna Gaussian broadcast channels”, *IEEE Trans. Inf. Theory*, vol. 51, no. 4, pp. 1570–1580, Apr. 2005.

[56] E. A. Jorswieck and H. Boche, “Rate balancing for the multi-antenna Gaussian broadcast channel”, in *Proc. IEEE Int. Symp. Spread Spectr. Techn. Appl. (ISSSTA)*, Prague, Czech Republic, Sep. 2002.

[57] ——, “On the performance optimization in multiuser MIMO systems”, *Trans. Emerg. Telecommun. Technol.*, vol. 18, no. 3, pp. 287–304, Apr. 2007.

[58] E. A. Jorswieck and E. G. Larsson, “Monotonic optimization framework for the MISO IFC”, in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 3633–3636.

[59] ——, “Monotonic optimization framework for the two-user MISO interference channel”, *IEEE Trans. Commun.*, vol. 58, no. 7, pp. 2159–2168, Jul. 2010.

[60] N. M. Josuttis, *The C++ Standard Library: A Tutorial and Reference*, 2nd ed. Reading, MA, USA: Addison-Wesley, 2012.

[61] J. Kaleva, A. Tölli, and M. Juntti, "Decentralized sum rate maximization with QoS constraints for interfering broadcast channel via successive convex approximation", *IEEE Trans. Signal Process.*, vol. 64, no. 11, pp. 2788–2802, Jun. 2016.

[62] N. Karmarkar, "A new polynomial-time algorithm for linear programming", *Combinatorica*, vol. 4, no. 4, pp. 373–395, Dec. 1984.

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

[64] L. G. Khachiyan, "A polynomial algorithm in linear programming", *Dokl. Akad. Nauk SSSR*, vol. 244, no. 5, pp. 1093–1096, 1979.

[65] M. N. Khormuji and M. Skoglund, "On instantaneous relaying", *IEEE Trans. Inf. Theory*, vol. 56, no. 7, pp. 3378–3394, Jul. 2010.

[66] D. E. Knuth, *The Art of Computer Programming: Fundamental Algorithms*, 3rd ed. Reading, MA, USA: Addison-Wesley, 1997, vol. 1.

[67] ——, *The Art of Computer Programming: Sorting and Searching*, 2nd ed. Reading, MA, USA: Addison-Wesley, 1997, vol. 3.

[68] A. H. Land and A. G. Doig, "An automatic method for solving discrete programming problems", *Econometrica*, vol. 28, no. 3, pp. 497–520, Jul. 1960.

[69] J. N. Laneman, D. N. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior", *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[70] J. D. C. Little, K. G. Murty, D. W. Sweeney, and C. Karel, "An algorithm for the traveling salesman problem", *Operations Res.*, vol. 11, no. 6, pp. 863–1025, Dec. 1963.

[71] L. Liu, R. Zhang, and K.-C. Chua, "Achieving global optimality for weighted sum-rate maximization in the K-user Gaussian interference channel with multiple antennas", *IEEE Trans. Wireless Commun.*, vol. 11, no. 5, pp. 1933–1945, May 2012.

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

[73] Z.-Q. Luo and S. Zhang, "Dynamic spectrum management: Complexity and duality", *IEEE J. Sel. Areas Commun.*, vol. 2, no. 1, pp. 57–73, Feb. 2008.

[74] B. Matthiesen, O. Aydin, and E. A. Jorswieck, "Throughput and energy-efficient network slicing", in *Proc. Int. ITG Workshop Smart Antennas (WSA)*, Bochum, Germany, Mar. 2018.

[75] B. Matthiesen, C. Hellings, and E. A. Jorswieck, "Energy efficiency: Rate splitting vs. point-to-point codes in Gaussian interference channels", in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Cannes, France, Jul. 2019.

[76] B. Matthiesen and E. A. Jorswieck, "Instantaneous relaying for the 3-way relay channel with circular message exchanges", in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Nov. 2015, pp. 475–479.

[77] ——, "Weighted sum rate maximization for non-regenerative multi-way relay channels with multi-user decoding", in *Proc. IEEE Int. Workshop Comput. Adv. Multi-Sensor Adaptive Process. (CAMSAP)*, Curaçao, Dutch Antilles, Dec. 2017.

[78] ——, "Optimal resource allocation for non-regenerative multiway relaying with rate splitting", in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Kalamata, Greece, Jun. 2018.

[79] ——, "Optimal resource allocation for non-regenerative multiway relaying with rate splitting", Poster, presented at *IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Kalamata, Greece, Jun. 2018. [Online]. Available: `http://sigport.org/3282` (visited on 07/31/2019).

[80] ——, "Global energy efficiency maximization in non-orthogonal interference networks", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Brighton, U.K., May 2019.

[81] ——, "Efficient global optimal resource allocation in non-orthogonal interference networks", *IEEE Trans. Signal Process.*, vol. 67, no. 21, pp. 5612–5627, Nov. 2019.

[82] B. Matthiesen, Y. Yang, and E. A. Jorswieck, "Optimization of weighted individual energy efficiencies in interference networks", in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Barcelona, Spain, Apr. 2018.

[83] B. Matthiesen, A. Zappone, and E. A. Jorswieck, “Resource allocation for energy-efficient 3-way relay channels”, *IEEE Trans. Wireless Commun.*, vol. 14, no. 8, pp. 4454–4468, Aug. 2015.

[84] MOSEK ApS. (2017). MOSEK Optimizer 8.1.0.34, [Online]. Available: `http://mosek.com`.

[85] A. S. Motahari and A. K. Khandani, “Capacity bounds for the Gaussian interference channel”, *IEEE Trans. Inf. Theory*, vol. 55, no. 2, pp. 620–643, Feb. 2009.

[86] K. G. Murty and S. N. Kabadi, “Some NP-complete problems in quadratic and nonlinear programming”, *Math. Program.*, vol. 39, no. 2, pp. 117–129, Jun. 1987.

[87] M. J. Neely, E. Modiano, and C. E. Rohrs, “Power allocation and routing in multibeam satellites with time-varying channels”, *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 138–152, Feb. 2003.

[88] A. Nemirovski and D. B. Yudin, *Problem complexity and method efficiency in optimization*. Hoboken, NJ, USA: Wiley, 1983.

[89] Y. Nesterov and A. Nemirovski, *Interior-Point Polynomial Algorithms in Convex Programming*, ser. SIAM Stud. Appl. Numer. Math. Philadelphia, PA, USA: SIAM, 1994, vol. 13.

[90] D. W. K. Ng, E. S. Lo, and R. Schober, “Energy-efficient resource allocation in multi-cell OFDMA systems with limited backhaul capacity”, *IEEE Trans. Wireless Commun.*, vol. 11, no. 10, pp. 3618–3631, Oct. 2012.

[91] NGMN Alliance, “Description of network slicing concept”, Tech. Rep., version 1.0, Jan. 2016.

[92] D. H. N. Nguyen and T. Le-Ngoc, “Sum-rate maximization in the multicell MIMO multiple-access channel with interference coordination”, *IEEE Trans. Wireless Commun.*, vol. 13, no. 1, pp. 36–48, Jun. 2014.

[93] L. Ong, S. J. Johnson, and C. M. Kellett, “The capacity region of multiway relay channels over finite fields with full data exchange”, *IEEE Trans. Inf. Theory*, vol. 57, no. 5, pp. 3016–3031, May 2011.

[94] L. Ong, C. M. Kellett, and S. J. Johnson, “Capacity theorems for the AWGN multi-way relay channel”, in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Austin, TX, USA, Jun. 2010, pp. 664–668.

[95] D. P. Palomar, “Convex primal decomposition for multicarrier linear MIMO transceivers”, *IEEE Trans. Signal Process.*, vol. 53, no. 12, pp. 4661–4674, Dec. 2005.

[96] D. P. Palomar and M. Chiang, “A tutorial on decomposition methods for network utility maximization”, *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.

[97] A. H. Phan, H. D. Tuan, and H. H. Kha, “D.C. iterations for SINR maximin multicasting in cognitive radio”, in *Proc. Int. Conf. Signal Process. Commun. Syst. (ICSPCS)*, Gold Coast, Australia, Dec. 2012.

[98] L. P. Qian, Y. J. Zhang, and J. Huang, “MAPEL: Achieving global optimality for a non-convex wireless power control problem”, *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1553–1563, 2009.

[99] T. K. Ralphs, “Parallel branch and cut”, in *Parallel Combinatorial Optimization*, ser. Wiley Ser. Parallel Distrib. Comput. E.-G. Talbi, Ed., Hoboken, NJ, USA: Wiley, 2006, ch. 3, pp. 53–102.

[100] R. T. Rockafellar, *Convex Analysis*, ser. Princeton Math. Ser. Princeton, NJ, USA: Princeton Univ. Press, 1970, vol. 28.

[101] M. Rossi, A. M. Tulino, O. Simeone, and A. M. Haimovich, “Non-convex utility maximization in gaussian MISO broadcast and interference channels”, in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Prague, Czech Republic, May 2011.

[102] P. Rost, C. Mannweiler, D. S. Michalopoulos, *et al.*, “Network slicing to enable scalability and flexibility in 5G mobile networks”, *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 72–79, May 2017.

[103] W. Rudin, *Principles of Mathematical Analysis*, 3rd ed., ser. Int. Ser. Pure Appl. Math. New York, NY, USA: McGraw-Hill, 1976.

[104] H. Sato, “The capacity of the Gaussian interference channel under strong interference (corresp.)”, *IEEE Trans. Inf. Theory*, vol. 27, no. 6, pp. 786–788, Nov. 1981.

[105] S. Schaible, “Fractional programming”, in *Handbook of Global Optimization*, ser. Nonconvex Optim. Appl. R. Horst and P. M. Pardalos, Eds., vol. 2, Norwell, MA, USA: Kluwer, 1995, pp. 495–608.

[106] S. Schaible and T. Ibaraki, “Fractional programming”, *Eur. J. Operational Res.*, vol. 12, no. 4, pp. 325–338, Apr. 1983.

[107] S. Schaible and J. Shi, "Fractional programming: The sum-of-ratios case", *Optim. Methods Softw.*, vol. 18, no. 2, pp. 219–229, 2003.

[108] B. Schein and R. G. Gallager, "The Gaussian parallel relay network", in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Sorrento, Italy, Jun. 2000.

[109] B. M. Scott, *Infimum of a union*, Mathematics Stack Exchange, Jun. 14, 2012. [Online]. Available: `https://math.stackexchange.com/q/158164` (visited on 08/06/2019).

[110] X. Shang and B. Chen, *Two-User Gaussian Interference Channels: An Information Theoretic Point of View*, ser. FnT Commun. Inf. Theory 3. Boston, MA, USA: Now, Dec. 2013, vol. 10, pp. 113–381.

[111] X. Shang, B. Chen, G. Kramer, and H. V. Poor, "Capacity regions and sum-rate capacities of vector Gaussian interference channels", *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5030–5044, Oct. 2010.

[112] X. Shang, G. Kramer, and B. Chen, "A new outer bound and the noisy-interference sum-rate capacity for Gaussian interference channels", *IEEE Trans. Inf. Theory*, vol. 55, no. 2, pp. 689–699, Feb. 2009.

[113] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. P. Fettweis, "5G-enabled tactile internet", *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 460–473, Mar. 2016.

[114] R. M. Soland, "An algorithm for separable nonconvex programming problems, II: Nonconvex constraints", *Manage. Sci.*, vol. 17, no. 11, pp. 661–786, Jul. 1971.

[115] Y. Sun, D. W. K. Ng, Z. Ding, and R. Schober, "Optimal joint power and subcarrier allocation for full-duplex multicarrier non-orthogonal multiple access systems", *IEEE Trans. Commun.*, vol. 65, no. 3, pp. 1077–1091, Mar. 2017.

[116] Y. Sun, D. W. K. Ng, J. Zhu, and R. Schober, "Robust and secure resource allocation for full-duplex MISO multicarrier NOMA systems", *IEEE Trans. Commun.*, vol. 66, no. 9, pp. 4119–4137, Sep. 2018.

[117] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks", *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[118] O. Tervo, H. Pennanen, D. Christopoulos, S. Chatzinotas, and B. Ottersten, “Distributed optimization for coordinated beamforming in multicell multigroup multicast systems: Power minimization and SINR balancing”, *IEEE Trans. Signal Process.*, vol. 66, no. 1, pp. 171–185, Jan. 2018.

[119] O. Tervo, A. Tölli, M. Juntti, and L.-N. Tran, “Energy-efficient beam coordination strategies with rate dependent processing power”, *IEEE Trans. Signal Process.*, 2017.

[120] O. Tervo, L.-N. Tran, and M. Juntti, “Optimal energy-efficient transmit beamforming for multi-user MISO downlink”, *IEEE Trans. Signal Process.*, vol. 63, no. 20, pp. 5574–5588, Oct. 2015.

[121] N. V. Thoai and H. Tuy, “Convergent algorithms for minimizing a concave function”, *Math. Operations Res.*, vol. 5, no. 4, pp. 556–566, Nov. 1980.

[122] D. N. C. Tse and S. V. Hanly, “Multiaccess fading channels. I. polymatroid structure, optimal resource allocation and throughput capacities”, *IEEE Trans. Inf. Theory*, vol. 44, no. 7, pp. 2796–2815, Nov. 1998.

[123] D. Tuninetti and Y. Weng, “On the Han-Kobayashi achievable region for Gaussian interference channels”, in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Toronto, ON, Canada, Jul. 2008.

[124] H. Tuy, *Convex Analysis and Global Optimization*, ser. Nonconvex Optim. Appl. Norwell, MA, USA: Kluwer, 1998, vol. 22.

[125] ——, “Monotonic optimization: Problems and solution approaches”, *SIAM J. Optim.*, vol. 11, no. 2, pp. 464–494, Feb. 2000.

[126] ——, “Robust solution of nonconvex global optimization problems”, *J. Global Optim.*, vol. 32, no. 2, pp. 307–323, Jun. 2005.

[127] ——, “$\mathcal{D}(\mathcal{C})$-optimization and robust global optimization”, *J. Global Optim.*, vol. 47, no. 3, pp. 485–501, Oct. 2009.

[128] ——, *Convex Analysis and Global Optimization*, 2nd ed., ser. Springer Optim. Appl. New York; Berlin, Germany; Vienna, Austria: Springer-Verlag, 2016, vol. 110.

[129] H. Tuy and R. Horst, “Convergence and restart in branch-and-bound algorithms for global optimization. application to concave minimization and D.C. optimization problems”, *Math. Program.*, vol. 41, no. 1–3, pp. 161–183, May 1988.

[130] H. Tuy, F. Al-Khayyal, and P. T. Thach, “Monotonic optimization: Branch and cut methods”, in *Essays and Surveys in Global Optimization*, C. Audet, P. Hansen, and G. Savard, Eds., New York; Berlin, Germany; Vienna, Austria: Springer-Verlag, 2005, ch. 2, pp. 39–78.

[131] H. Tuy, T. V. Thieu, and N. Q. Thai, “A conical algorithm for globally minimizing a concave function over a closed convex set”, *Math. Operations Res.*, vol. 10, no. 3, pp. 498–514, Aug. 1985.

[132] W. Utschick and J. Brehmer, “Monotonic optimization framework for coordinated beamforming in multicell networks”, *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1899–1909, Apr. 2012.

[133] S. A. Vavasis, “Complexity issues in global optimization: A survey”, in *Handbook of Global Optimization*, ser. Nonconvex Optim. Appl. R. Horst and P. M. Pardalos, Eds., vol. 2, Norwell, MA, USA: Kluwer, 1995, pp. 27–41.

[134] L. Wang, Y.-H. Kim, C.-Y. Chen, H. Park, and E. Şaşoğlu, “Sliding-window superposition coding: Two-user interference channels”, submitted to *IEEE Trans. Inf. Theory*, Jan. 2017, [Online]. Available: `https://arxiv.org/abs/1701.02345`.

[135] L. Wang, E. Şaşoğlu, B. Bandemer, and Y.-H. Kim, “A comparison of superposition coding schemes”, in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Turkey, Jul. 2013.

[136] Z. Wang, P.-Y. Chiang, P. Nazari, C.-C. Wang, Z. Chen, and P. Heydari, “A CMOS 210-GHz fundamental transceiver with OOK modulation”, *IEEE J. Solid-State Circuits*, vol. 49, no. 3, pp. 564–580, Mar. 2014.

[137] 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 Trans. Signal Process.*, vol. 59, no. 8, pp. 3977–3996, Aug. 2011.

[138] P. C. Weeraddana, M. Codreanu, M. Latva-aho, A. Ephremides, and C. Fischione, *Weighted Sum-Rate Maximization in Wireless Networks: A Review*, ser. FnT Netw. 1–2. Boston, MA, USA: Now, 2012, vol. 6, pp. 1–163.

[139] L. A. Wolsey, *Integer Programming*, ser. Wiley-Intersci. Ser. Discrete Math. Optim. Hoboken, NJ, USA: Wiley, 1998.

[140] J. Xu and L. Qiu, "Energy efficiency optimization for MIMO broadcast channels", *IEEE Trans. Wireless Commun.*, vol. 12, no. 2, pp. 690–701, Feb. 2013.

[141] J. Xu, Z. Miao, and Q. Liu, "New method to get essential efficient solution for a class of D.C. multiobjective problem", in *Proc. Int. Conf. Comput., Mechatronics, Control, Electron. Eng. (CMCE)*, Changchun, China, Oct. 2010.

[142] Q. Xu, X. Li, H. Ji, and X. Du, "Energy-efficient resource allocation for heterogeneous services in OFDMA downlink networks: Systematic perspective", *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2071–2082, Jun. 2014.

[143] S. Yang, M. Kobayashi, D. Gesbert, and X. Yi, "Degrees of freedom of time correlated MISO broadcast channel with delayed CSIT", *IEEE Trans. Inf. Theory*, vol. 59, no. 1, pp. 315–328, Jan. 2013.

[144] W. Yu, G. Ginis, and J. Cioffi, "Distributed multiuser power control for digital subscriber lines", *IEEE J. Sel. Areas Commun.*, vol. 20, no. 5, pp. 1105–1115, Jun. 2002.

[145] W. Yu and R. Lui, "Dual methods for nonconvex spectrum optimization of multicarrier systems", *IEEE Trans. Commun.*, vol. 54, no. 7, pp. 1310–1322, Jul. 2006.

[146] L. A. Zadeh, "Optimality and non-scalar-valued performance criteria", *IEEE Trans. Autom. Control*, vol. 8, no. 1, pp. 59–60, Jan. 1963.

[147] A. Zappone, E. Björnson, L. Sanguinetti, and E. A. Jorswieck, "Globally optimal energy-efficient power control and receiver design in wireless networks", *IEEE Trans. Signal Process.*, vol. 65, no. 11, pp. 2844–2859, Jun. 2017.

[148] A. Zappone, Z. Chong, E. A. Jorswieck, and S. Buzzi, "Energy-aware competitive power control in relay-assisted interference wireless networks", *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1860–1871, Apr. 2013.

[149] A. Zappone and E. A. Jorswieck, *Energy Efficiency in Wireless Networks via Fractional Programming Theory*, ser. FnT Commun. Inf. Theory 3–4. Boston, MA, USA: Now, 2015, vol. 11, pp. 113–381.

[150] A. Zappone, E. A. Jorswieck, and S. Buzzi, “Energy efficiency and interference neutralization in two-hop MIMO interference channels”, *IEEE Trans. Signal Process.*, vol. 62, no. 24, pp. 6481–6495, Dec. 2014.

[151] A. Zappone, B. Matthiesen, and E. A. Jorswieck, “Energy efficiency in MIMO underlay and overlay device-to-device communications and cognitive radio systems”, *IEEE Trans. Signal Process.*, vol. 65, no. 4, pp. 1026–1041, Feb. 2017.

[152] A. Zappone, L. Sanguinetti, G. Bacci, E. A. Jorswieck, and M. Debbah, “Energy-efficient power control: A look at 5G wireless technologies”, *IEEE Trans. Signal Process.*, vol. 64, no. 7, pp. 1668–1683, Apr. 2016.

[153] R. Zhang and S. Cui, “Cooperative interference management with MISO beamforming”, *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5450–5458, Oct. 2010.

[154] Y. J. Zhang, L. P. Qian, and J. Huang, *Monotonic Optimization in Communication and Networking Systems*, ser. FnT Netw. 1. Boston, MA, USA: Now, 2012, vol. 7, pp. 1–75.

[155] L. Zhou and W. Yu, “On the capacity of the K-user cyclic Gaussian interference channel”, *IEEE Trans. Inf. Theory*, vol. 59, no. 1, pp. 154–165, Jan. 2013.