

# Mixed Monotonic Programming for Fast Global Optimization

Bho Matthiesen, *Member, IEEE*, Christoph Hellings, *Member, IEEE*,  
Eduard A. Jorswieck, *Fellow, IEEE*, and Wolfgang Utschick, *Senior Member, IEEE*

**Abstract**—While globally optimal solutions to many convex programs can be computed efficiently in polynomial time, this is, in general, not possible for nonconvex optimization problems. Therefore, locally optimal approaches or other efficient suboptimal heuristics are usually applied for practical implementations. However, there is also a strong interest in computing globally optimal solutions of nonconvex problems in offline simulations in order to benchmark the faster suboptimal algorithms. Global solutions often rely on monotonicity properties. A common approach is to reformulate problems into a canonical monotonic optimization problem where the monotonicity becomes evident, but this often comes at the cost of nested optimizations, increased numbers of variables, and/or slow convergence. The framework of mixed monotonic programming (MMP) proposed in this paper avoids such performance-deteriorating reformulations by revealing hidden monotonicity properties directly in the original problem formulation. By means of a wide range of application examples from the area of signal processing for communications (including energy efficiency for green communications, resource allocation in interference networks, scheduling for fairness and quality of service, as well as beamformer design in multi-antenna systems), we demonstrate that the novel MMP approach leads to tremendous complexity reductions compared to state-of-the-art methods for global optimization. However, the framework is not limited to optimizing communication systems, and we expect that similar speed-ups can be obtained for optimization problems from other areas of research as well.

**Index Terms**—Resource allocation, global optimization, interference networks, monotonic optimization, branch-and-bound

## I. INTRODUCTION

In point-to-point communication systems without interference, the optimization of various performance metrics can be

B. Matthiesen is with the Department of Communications Engineering, University of Bremen, 28359 Bremen, Germany (e-mail: matthiesen@uni-bremen.de). C. Hellings is with ETH Zürich, Department of Physics, 8093 Zürich, Switzerland (e-mail: chellings@phys.ethz.ch). E. A. Jorswieck is with Technische Universität Braunschweig, Institut für Nachrichtentechnik, 38106 Braunschweig, Germany (e-mail: jorswieck@ifn.ing.tu-bs.de). W. Utschick is with Technische Universität München, Professur für Methoden der Signalverarbeitung, 80290 München, Germany (e-mail: utschick@tum.de). Part of this research was conducted while B. Matthiesen and E. A. Jorswieck were with Technische Universität Dresden and C. Hellings was with Technische Universität München.

This work is supported in part by the German Research Foundation (DFG) in the Collaborative Research Center 912 “Highly Adaptive Energy-Efficient Computing,” under Germany’s Excellence Strategy (EXC 2077 at University of Bremen, University Allowance), and under grant number JO 801/24-1. We thank the Center for Information Services and High Performance Computing (ZIH) at TU Dresden for generous allocations of computer time. B. Matthiesen and C. Hellings contributed equally to this work.

Conference versions of two application examples can be found in [1], [2]. This journal version gives a more general perspective on the framework, gives more details on algorithmic aspects, and discusses many further application examples.

formulated as convex programs such as in rate maximization [3] or mean square error minimization [4]. More complicated objective functions in the context of energy efficiency optimization can be shown to be pseudoconvex or quasiconvex [5]. Even in advanced scenarios with multiple antennas or parallel transmission on orthogonal carriers, these optimization problems can be solved with efficient methods from convex optimization [4] or fractional programming [5], and sometimes even in closed form [3]. However, in multi-terminal scenarios with interfering users, performance optimization typically involves nonconvex problems. This is often due to interference terms that make the rate equations nonconcave or due to product operations contained in multiuser utility functions.

Apart from special cases where efficient solutions exist,<sup>1</sup> performance optimization in interference networks is, thus, usually tackled by locally optimal approaches or suboptimal heuristics. Examples are gradient ascent algorithms [8]–[11], successive allocation methods [12], [13], successive (pseudo-)convex approximation [14]–[16], alternating optimization [17]–[21], distributed interference pricing [22], or game-theoretic methods [23]–[26]. Such heuristics are good candidates for practical implementation due to their low computational complexity and/or the possibility of distributed implementation. However, there is also a strong interest in globally optimal solutions to assess the fundamental limits of the considered multiuser communication systems and to have benchmarks for the heuristic methods.

In order to obtain such global solutions, researchers have applied methods from the field of monotonic optimization [27]–[30] to optimization problems in various communication systems. For instance, monotonic programming was applied in interference channels [7], [31]–[38], in broadcast channels with linear transceivers [11], [39]–[42], in interfering broadcast channels [43], in relaying scenarios [25], [36], and in satellite systems [44] with the aim of maximizing weighted sum rates [7], [31]–[33], [35], fairness-based performance metrics [34], [35], [38]–[40], [43], or the energy efficiency [25], [36], [37], [42] as well as minimizing the required sum transmit power [11], [41], [44]. Some of these applications include solutions for multi-antenna systems [7], [11], [31], [32], [38]–[41], [43], allow to average data rates over several time slots [34], [35], [38], [39], [41], and/or incorporate additional robustness considerations [43]. Moreover, monotonic optimization can also be applied on the medium access control layer. One example is optimizing

<sup>1</sup>E.g., in multiple-input/multiple-output (MIMO) broadcast channels with dirty paper coding [6] or for rate balancing problems in multiple-input/single-output (MISO) interference channels with interference treated as noise [7].

the transmit probabilities in the slotted ALOHA protocol [35]. A wider overview with further application examples can be found in [35], [36], [45].

A common approach is to reformulate the objective function<sup>2</sup> as a difference  $f^+(\mathbf{x}) - f^-(\mathbf{x})$  of nondecreasing functions  $f^+$  and  $f^-$ . The resulting difference-of-monotonic (DM) problem can be reformulated further into a canonical monotonic optimization problem where a nondecreasing function is maximized over a normal set.<sup>3</sup> For instance, instead of maximizing  $f^+(\mathbf{x}) - f^-(\mathbf{x})$  over a box  $[\mathbf{r}, \mathbf{s}]$ , we can maximize the nondecreasing function  $f^+(\mathbf{x}) + t$  under the additional constraints  $f^-(\mathbf{x}) + t \leq 0$  and  $t \in [-f^-(\mathbf{s}), -f^-(\mathbf{r})]$  which form a normal set.<sup>4</sup> The resulting canonical monotonic optimization problem can then be solved with the so-called Polyblock Algorithm (PA) [30, Sec. 11.2] as was done, e.g., in [31], [32]. An important drawback is that the number of optimization variables is increased by introducing the auxiliary variable  $t$ . This negatively affects the convergence speed because the PA has exponential worst-case complexity in the number of variables [45].

As an alternative, DM problems can be solved by means of branch-and-bound (BB) techniques as described in [29]. This approach, which was pursued in [11], [41], [43], [44], avoids the overhead of the additional optimization variable  $t$ , but still suffers from drawbacks that will be observed in Section IV-A. Just like the PA, BB methods rely on calculating utopian bounds to the objective function, and their convergence speed depends heavily on the quality of these bounds. Unfortunately, DM bounds are, in general, not very tight.

Therefore, several authors have proposed to improve the speed of convergence by reparameterizing an optimization problem in terms of a new set of monotonic variables. For instance, [33]–[35] use the signal-to-interference-plus-noise ratio (SINR) values of the users as optimization variables instead of their transmit powers, while [7], [11], [38]–[41] use the achievable rates, and [44] uses the received interference powers. The resulting monotonic or DM problems can then be solved by means of the PA [33]–[35], [38]–[40] or a BB algorithm [11], [41], [44]. However, the change of variables usually makes the evaluation of the objective and constraint functions more costly. For instance, the SINR values and achievable rates can be calculated analytically when the transmit powers are used as optimization variables, but a fixed point iteration is necessary to calculate the transmit powers if the SINR values or the achievable rates are used as variables (see, [33]–[35] and [7], [11], [38]–[41], respectively). Thus, a change of variables might reduce the number of iterations required in the monotonic programming method, but comes at the cost of increasing the computational complexity of each iteration.

Moreover, not all optimization problems can be conveniently rewritten in terms of monotonic functions or DM functions. For instance, in the context of energy-efficient communications, we encounter objective functions that can be written as fractions of DM functions. For this type of problems, the fractional

monotonic programming method proposed in [25], [36], [37] uses a monotonic programming approach as an inner solver inside Dinkelbach's method for fractional programs. This combination has the drawback that a highly complex monotonic programming algorithm has to be executed not only once but repeatedly in each iteration of the outer algorithm. Moreover, it is no longer possible to obtain a rigorous guarantee that the obtained solution is indeed  $\eta$ -optimal, i.e., that it is no more than a given constant  $\eta$  away from the exact globally optimal solution.

In this paper, we propose the framework of mixed monotonic programming (MMP) which avoids all these drawbacks since it neither requires a reformulation of the objective function nor a change of variables. Instead, the main idea is that a function defined by several terms might have different monotonicity properties in each term and variable. Thus, the MMP approach does not consider whether the whole function is monotonic in a variable, but takes the monotonicity for each occurrence of a variable separately into account by formulating a so-called mixed monotonic (MM) function. If such an MM function can be constructed for a given optimization problem, the problem can be solved by a BB algorithm as discussed in Section III.<sup>5</sup> In Section IV, we show that a wide variety of optimization problems (including the difficult fractional monotonic problems mentioned above) can be solved with the MMP approach, and we demonstrate significant advantages compared to state-of-the-art solutions using the C++ implementation available at [46].

Note that there are several existing approaches that can be considered as special cases of the MMP framework, the most prominent being DM formulations. However, the MMP approach is much more general and can be used to find solution methods that are faster than the DM approach. This will become clear after the formal definition of an MMP problem in Section II. Moreover, some specialized solution methods developed for particular optimization problems can be identified to fall into the more general MMP framework. For instance, [42] exploited a structure with a fraction of nonnegative nondecreasing functions of a scalar variable, and [47], [48] consider an optimization problem in a two-user interference channel that can be identified as a two-dimensional special case of the MMP framework. An implementation of the BRB algorithm for MMP problems can, thus, be readily applied to any of these special cases.

*Notation:* We use  $\mathbf{0}$  for the zero vector,  $\mathbf{1}$  for the all-ones vector, and  $\mathbf{I}_L$  for the identity matrix of size  $L$ . Vectors are written in bold-face lowercase and matrices in bold-face uppercase. Inequalities between vectors are meant component-wise, i.e.,  $\mathbf{x} \geq \mathbf{y}$  if and only if  $x_i \geq y_i$  for all  $i$ , and  $[\mathbf{r}, \mathbf{s}] = \{\mathbf{x} \mid \mathbf{r} \leq \mathbf{x} \leq \mathbf{s}\}$  denotes a box (hyperrectangle). We use shorthand notations of the form  $(\bullet_k)_{\forall k} = (\bullet_1, \dots, \bullet_K)$ , and we write  $\mathcal{CN}(0, 1)$  for the circularly symmetric Gaussian distribution with zero mean and unit variance.

<sup>2</sup>Similar reformulations can be applied to the constraints if needed.

<sup>3</sup>A set  $\mathcal{G} \subset \mathbb{R}_0^+$  is called normal if  $[\mathbf{0}; \mathbf{x}] \subseteq \mathcal{G}$  for all  $\mathbf{x} \in \mathcal{G}$  [28].

<sup>4</sup>Please refer to [30, Thm. 11.1] for more details.

<sup>5</sup>A branch-reduce-and-bound (BRB) algorithm is a special kind of BB algorithm that includes a reduction step to speed up the convergence.

## II. MIXED MONOTONIC PROGRAMMING

Consider the optimization problem

$$\max_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x}) \quad (\text{P})$$

with continuous objective function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  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 III-A. Let  $\mathcal{M}_0 = [r^0, s^0]$  be a box in  $\mathbb{R}^n$  enclosing  $\mathcal{D}$ , i.e.,  $\mathcal{M}_0 \supseteq \mathcal{D}$ . Assume there exists a continuous function  $F : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  such that

$$F(\mathbf{x}, \mathbf{x}) = f(\mathbf{x}) \quad (1)$$

for all  $\mathbf{x} \in \mathcal{M}_0$  and

$$F(\mathbf{x}, \mathbf{y}) \leq F(\mathbf{x}', \mathbf{y}) \quad \text{if } \mathbf{x} \leq \mathbf{x}', \quad (2a)$$

$$F(\mathbf{x}, \mathbf{y}) \geq F(\mathbf{x}, \mathbf{y}') \quad \text{if } \mathbf{y} \leq \mathbf{y}'. \quad (2b)$$

for all  $\mathbf{x}, \mathbf{x}', \mathbf{y}, \mathbf{y}' \in \mathcal{M}_0$ . We call such a function a *mixed monotonic (MM)* function. The optimization problem (P) is said to be a *mixed monotonic programming (MMP)* problem if its objective has an MMP representation, i.e., if  $f$  satisfies (1) for some MM function  $F$ . In the following section, we will show that MMP problems are especially well suited for solution by a BB procedure.

As mentioned before, some well established problem formulations can be identified as special cases of this novel MMP framework. The most prominent among them are DM programs [28], i.e.,

$$\max_{\mathbf{x} \in \mathcal{D}} f^+(\mathbf{x}) - f^-(\mathbf{x}) \quad (3)$$

where  $f^+$  and  $f^-$  are nondecreasing functions. A MMP representation of that objective is  $F(\mathbf{x}, \mathbf{y}) = f^+(\mathbf{x}) - f^-(\mathbf{y})$ .

However, the MMP approach is much more versatile. For example, consider the fraction

$$\frac{p^+(\mathbf{x}) - p^-(\mathbf{x})}{q(\mathbf{x})} \quad (4)$$

with nondecreasing  $p^+$ ,  $p^-$ , and  $q$ , where we assume  $p^+(\mathbf{x}) - p^-(\mathbf{x}) \geq 0$  and  $q(\mathbf{x}) > 0$  for all  $\mathbf{x}$ . Maximizing this function with monotonic programming requires the combination of Dinkelbach's algorithm [49] as outer and monotonic programming as inner solver [37]. This approach has the drawbacks that the inner global optimization problem needs to be solved several times and the stopping criterion does not guarantee an  $\eta$ -optimal solution. Instead, (4) can be optimized directly by the algorithm proposed in Section III since it is easily verified that

$$F(\mathbf{x}, \mathbf{y}) = \frac{p^+(\mathbf{x}) - p^-(\mathbf{y})}{q(\mathbf{y})} \quad (5)$$

is an MMP representation of (4).

It is important to note that the MMP representation of  $f$  is never unique. This can be observed in the following simple example. Let  $F$  be an MMP representation of  $f$ . Then, it is easy to verify that

$$\tilde{F}(\mathbf{x}, \mathbf{y}) = F(\mathbf{x}, \mathbf{y}) + \sum_{i=1}^N (x_i - y_i) \quad (6)$$

fulfills the requirements in (1) and (2) as well. Intuitively,  $\tilde{F}$  can be understood as an MMP version of

$$\tilde{f}(\mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^N (x_i - x_i). \quad (7)$$

However, while we obviously have  $\tilde{f}(\mathbf{x}) = f(\mathbf{x})$ , the difference between the MMP representations  $\tilde{F}$  and  $F$  is crucial. As we will see later,  $F$  leads to tighter bounds than  $\tilde{F}$  which, in turn, leads to faster convergence of the BB algorithm.

Another, practically more relevant, example for the non-uniqueness of  $F$  is throughput maximization in wireless interference networks [37]

$$\max_{\mathbf{0} \leq \mathbf{p} \leq \mathbf{P}} \sum_{i=1}^K \log \left( 1 + \frac{\alpha_i p_i}{\sigma_i^2 + \sum_{j=1}^K \beta_{i,j} p_j} \right) \quad (8)$$

with positive constants  $\alpha_i$ ,  $\sigma_i$ , and nonnegative  $\beta_{i,j}$ . Conventionally, (8) is converted into a DM program (3) with  $f^+(\mathbf{x}) = \sum_{i=1}^K \log(\alpha_i x_i + \sigma_i^2 + \sum_{j=1}^K \beta_{i,j} x_j)$  and  $f^-(\mathbf{x}) = \sum_{i=1}^K \log(\sigma_i^2 + \sum_{j=1}^K \beta_{i,j} x_j)$ . This yields the MMP representation in the text below (3). A more direct approach to obtain  $F$  from (8) is

$$F(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^K \log \left( 1 + \frac{\alpha_i x_i}{\sigma_i^2 + \beta_{i,i} x_i + \sum_{j \neq i} \beta_{i,j} y_j} \right). \quad (9)$$

This example will be continued in Section IV-A. An important aspect discussed there is how the precise choice of  $F$  directly impacts the convergence speed of the developed algorithm.

To conclude this section, we state some useful properties of MM functions. Let  $F_i(\mathbf{x}, \mathbf{y})$  be MM functions for  $i = 1, \dots, K$ . Then,

$$(\mathbf{x}, \mathbf{y}) \mapsto \sum_{i=1}^K F_i(\mathbf{x}, \mathbf{y}), \quad (10)$$

$$(\mathbf{x}, \mathbf{y}) \mapsto \max_{i=1, \dots, K} F_i(\mathbf{x}, \mathbf{y}), \quad (\mathbf{x}, \mathbf{y}) \mapsto \min_{i=1, \dots, K} F_i(\mathbf{x}, \mathbf{y}) \quad (11)$$

are MM functions, i.e., the properties of MM functions are preserved by summation and by taking the pointwise minimum or maximum of several MM functions. Moreover, if  $g(\mathbf{x})$  is a real-valued, nondecreasing function, and  $h(\mathbf{x})$  is a real-valued, nonincreasing function, the composed functions

$$(\mathbf{x}, \mathbf{y}) \mapsto g(F_i(\mathbf{x}, \mathbf{y})), \quad (\mathbf{x}, \mathbf{y}) \mapsto h(F_i(\mathbf{y}, \mathbf{x})) \quad (12)$$

are MM functions as well. Note that the nondecreasing and non-increasing variables are swapped in case of a composition with a nonincreasing function, i.e., to ensure that the composition of  $h$  and  $F_i$  is nondecreasing in its first argument and nonincreasing in the second one, the first argument of the composition has to be plugged into  $F_i$  as the second argument and vice versa. In particular, it follows from (12) that  $(\mathbf{x}, \mathbf{y}) \mapsto -F(\mathbf{y}, \mathbf{x})$  and  $(\mathbf{x}, \mathbf{y}) \mapsto 1/F(\mathbf{y}, \mathbf{x})$  are MM if  $F(\mathbf{x}, \mathbf{y})$  is a positive MM function. If in addition  $F_i(\mathbf{x}, \mathbf{y}) \geq 0$  for all  $i = 1, \dots, K$  and  $\mathbf{x}, \mathbf{y} \in \mathcal{X}$  for some  $\mathcal{X} \subseteq \mathbb{R}^n$ , then

$$(\mathbf{x}, \mathbf{y}) \mapsto \prod_{i=1}^K F_i(\mathbf{x}, \mathbf{y}) \quad (13)$$

is an MM function on  $\mathcal{X}$ , i.e., the product of nonnegative MM functions is MM as well.

### III. GLOBAL OPTIMAL SOLUTION OF (P)

We design a BRB algorithm to determine a global  $\eta$ -optimal solution of (P), i.e., a feasible point  $\bar{x} \in \mathcal{D}$  such that  $f(\bar{x}) \geq f(x) - \eta$  for all  $x \in \mathcal{D}$ . The core idea of any BB algorithm, including the considered BRB variant, is to relax the feasible set  $\mathcal{D}$  and subsequently partition it such that upper bounds on the objective value can be determined easily.<sup>6</sup> This is where the MMP representation  $F$  of the objective function  $f$  comes in handy as it is well suited to compute upper bounds over rectangular sets. Let  $\mathcal{M} = [r, s]$  be a box in  $\mathbb{R}^n$ . Then,

$$\max_{x \in \mathcal{M} \cap \mathcal{D}} f(x) \leq \max_{x \in \mathcal{M}} F(x, x) \leq \max_{x, y \in \mathcal{M}} F(x, y) = F(s, r) \quad (14)$$

gives an upper bound  $U(\mathcal{M}) = U([r, s]) = F(s, r)$  on the optimal value of  $f(x)$  on  $\mathcal{M} \cap \mathcal{D}$ . Thus, rectangular subdivision [30, Sec. 6.1.3], where a box  $\mathcal{M}$  is partitioned along a hyperplane parallel to one of its facets, is an excellent choice to partition  $\mathcal{D}$ . Given a point  $v \in \mathcal{M}$  and index  $j \in \{1, 2, \dots, n\}$ , we divide  $\mathcal{M}$  along the hyperplane  $x_j = v_j$ . The resulting partition sets are the subrectangles

$$\mathcal{M}^- = \{x \mid r_j \leq x_j \leq v_j, r_i \leq x_i \leq s_i \ (i \neq j)\} \quad (15a)$$

$$\mathcal{M}^+ = \{x \mid v_j \leq x_j \leq s_j, r_i \leq x_i \leq s_i \ (i \neq j)\}. \quad (15b)$$

This is referred to as a *partition via*  $(v, j)$  of  $\mathcal{M}$ . A partition of  $\mathcal{M}$  via  $(\frac{1}{2}(s + r), j)$  where  $j \in \arg \max_j s_j - r_j$  is called a *bisection* of  $\mathcal{M}$ .

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 following proposition is an important property for the convergence of BB methods.

*Lemma 1 ([30, Corollary 6.2]):* Let  $\{\mathcal{M}_k\}$  be a decreasing sequence of sets such that  $\mathcal{M}_{k+1}$  is a descendent of  $\mathcal{M}_k$  in a bisection along a longest side of  $\mathcal{M}_k$ . Then, the diameter  $\text{diam}(\mathcal{M}_k)$  of  $\mathcal{M}_k$  tends to zero as  $k \rightarrow \infty$ .

Besides the subdivision procedure and computation of bounds, the selection of the next box (or branch) for further partitioning is crucial for the convergence and implementation of a BB procedure. A widely used selection criterion is

$$\mathcal{M}_k \in \arg \max \{U(\mathcal{M}) \mid \mathcal{M} \in \mathcal{R}_{k-1}\}. \quad (16)$$

where  $U(\mathcal{M})$  is the upper bound chosen for the BB method — in our case the MMP bound defined below (14) — and  $\mathcal{R}_{k-1}$  holds all undecided boxes from the previous iteration [30, Sec. 6.2], i.e., all boxes for which it is not yet clear whether or not they contain the global optimum. However, this selection might not be the best choice from an implementation point of view. To guarantee convergence, it suffices if the selection satisfies the following condition.

*Definition 1 ([27, Def IV.6]):* A selection operation is said to be *bound improving* if, at least each time after a finite number of steps,  $\mathcal{M}_k$  satisfies (16).

By construction, (16) satisfies Definition 1. An alternative is to select one of the oldest elements in  $\mathcal{R}_{k-1}$ , i.e., define for every  $\mathcal{M}$  by  $\sigma(\mathcal{M})$  the iteration index of its creation and select

$$\mathcal{M}_k \in \arg \min \{\sigma(\mathcal{M}) \mid \mathcal{M} \in \mathcal{R}_{k-1}\}. \quad (17)$$

<sup>6</sup>Please refer to [50, Chapter 3], [27, Chapter 4], or [30, Chapter 6] for a thorough introduction to BB methods.

Due to the finiteness of  $\mathcal{R}_k$  every set  $\mathcal{M} \in \mathcal{R}_k$  will be deleted or selected after finitely many iterations [27, p. 130].

The final BRB procedure is stated in Algorithm 1. It is initialized in Step 0 where an initial box  $\mathcal{M}_0 = [r^0, s^0]$  containing the feasible set  $\mathcal{D}$  is required, i.e.,

$$r_i^0 \leq \min_{x \in \mathcal{D}} x_i \quad s_i^0 \geq \max_{x \in \mathcal{D}} x_i \quad (18)$$

for all  $i = 1, 2, \dots, n$ . In Step 1, a box is selected for partitioning with any bound improving selection rule, e.g., (16) or (17), and then bisected along one of its longest dimensions. Step 2 is optional and discussed separately in Section III-B. For each newly constructed box, a feasible value is computed in Step 3. If necessary, the current best known feasible solution  $\bar{x}^k$  (the ‘‘incumbent’’) and current best known value  $\gamma_k$  are updated. Infeasible boxes, i.e., new boxes that do not contain any feasible points, are deleted (pruned) in Step 4. Note that the box selected in Step 1 is replaced by the new boxes and, thus, removed from the partition  $\mathcal{R}_k$ . In Step 5, the algorithm is terminated if the partition is empty or if none of the remaining boxes can contain any better solution. Otherwise, the algorithm continues in Step 1. Convergence of Algorithm 1 to an  $\eta$ -optimal solution

#### Algorithm 1 BRB Algorithm for MMP Problems

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

**Step 1 (Branching)** Select a box  $\mathcal{M}_k = [r^k, s^k] \in \mathcal{R}_{k-1}$  and bisect  $\mathcal{M}_k$  via  $(\frac{1}{2}(s^k + r^k), j)$  with  $j \in \arg \max_j s_j^k - r_j^k$ . Let  $\mathcal{P}_k = \{\mathcal{M}_k^-, \mathcal{M}_k^+\}$  with  $\mathcal{M}_k^-, \mathcal{M}_k^+$  as in (15).

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

$$(\mathcal{M} \setminus \mathcal{M}') \cap \{x \in \mathcal{D} \mid F(x, x) > \gamma_k\} = \emptyset. \quad (19)$$

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

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

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

of (P) is established below<sup>7</sup> for any  $F$  satisfying (1) and (2).

*Theorem 1:* Algorithm 1 converges towards a global  $\eta$ -optimal solution of (P) if the selection is bound improving.

*Proof:* In Step 2, let  $\mathcal{D}' = \{x \in \mathcal{D} \mid F(x, x) > \gamma_k\} \subseteq \mathcal{D}$  and observe that  $\mathcal{D} \setminus \mathcal{D}'$  does not contain any solutions better than the current best solution. Thus, if  $\mathcal{M}'$  satisfies (19), no solutions better than the current incumbent are lost and the reduction does not affect the solution of (P).

If the algorithm terminates in Step 5 and iteration  $K$ , then  $F(s, r) \leq \gamma_K + \eta$  for all  $[r, s] \in \mathcal{R}_K$  and, since  $F(x, y) > -\infty$ ,  $\gamma_K > -\infty$ . Hence,  $\bar{x}_K$  is feasible and

$$\gamma_K = f(\bar{x}_K) \geq F(r, s) - \eta \geq \max_{x \in \mathcal{M} \cap \mathcal{D}} f(x) - \eta \quad (20)$$

<sup>7</sup>We combine the convergence proof from [51, Prop. 5.6], [30, Prop. 6.1] with the idea of a general selection criterion from [27, Thm. IV.3].

for every  $\mathcal{M} \in \mathcal{R}_K$ . Now, for every  $\mathbf{x} \in \mathcal{D}$ , either  $\mathbf{x} \in \mathcal{D} \cap \bigcup_{\mathcal{M} \in \mathcal{R}_K} \mathcal{M}$  or  $\mathbf{x} \in \mathcal{D} \setminus \bigcup_{\mathcal{M} \in \mathcal{R}_K} \mathcal{M}$ . In the first case,  $f(\mathbf{x}) - \eta \leq f(\bar{\mathbf{x}}_K)$  due to (20). In the latter case,  $\mathbf{x} \in \mathcal{M}'$  for some  $\mathcal{M}' = [\mathbf{r}', \mathbf{s}'] \in \mathcal{P}_{k'} \setminus \mathcal{P}'_{k'}$  and some  $k'$  since  $\mathcal{M}_0 \supseteq \mathcal{D}$ . Because  $\{\gamma_k\}$  is nondecreasing and due to Step 4,  $f(\mathbf{x}) \leq F(\mathbf{s}', \mathbf{r}') \leq \gamma_{k'} + \eta \leq \gamma_K + \eta$ . Hence, for every  $\mathbf{x} \in \mathcal{D}$ ,  $f(\mathbf{x}) \leq f(\bar{\mathbf{x}}_K) + \eta$ .

It remains to show 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  $\mathcal{M}_{k_q} \in \arg \max\{F(\mathbf{s}, \mathbf{r}) \mid [\mathbf{r}, \mathbf{s}] \in \mathcal{R}_{k_q}\}$ . Because  $\mathcal{M}_{k_q} \cap \mathcal{D} \neq \emptyset$ , there exists an  $\mathbf{x}^{k_q} \in \mathcal{M}_{k_q} \cap \mathcal{D}$ . Due to Lemma 1,  $\text{diam } \mathcal{M}_{k_q} \rightarrow 0$  as  $q \rightarrow \infty$ . Thus,  $\mathbf{x}^{k_q}$ ,  $\mathbf{s}^{k_q}$ , and  $\mathbf{r}^{k_q}$  all converge towards a common limit, and, together with (1),  $F(\mathbf{s}^{k_q}, \mathbf{r}^{k_q}) \rightarrow F(\mathbf{x}^{k_q}, \mathbf{x}^{k_q}) = f(\mathbf{x}^{k_q})$ . Since  $\alpha([\mathbf{r}, \mathbf{s}]) \leq \sup_{\mathbf{x} \in [\mathbf{r}, \mathbf{s}] \cap \mathcal{D}} f(\mathbf{x}) \leq F(\mathbf{s}, \mathbf{r}) \leq F(\mathbf{s}^{k_q}, \mathbf{r}^{k_q})$  for all  $[\mathbf{r}, \mathbf{s}] \in \mathcal{R}_{k_q}$ ,  $F(\mathbf{s}^{k_q}, \mathbf{r}^{k_q}) \rightarrow \gamma_{k_q}$  and, hence,  $F(\mathbf{s}^{k_q}, \mathbf{r}^{k_q}) = \gamma_{k_q} + \delta_{k_q}$  with  $\delta_{k_q} \geq 0$  and  $\lim_{q \rightarrow \infty} \delta_{k_q} = 0$ . Thus, there exists a  $\tilde{K}$  such that  $\delta_k \leq \eta$  for all  $k > \tilde{K}$ , and  $F(\mathbf{s}, \mathbf{r}) \leq \gamma_k + \eta$  for all  $[\mathbf{r}, \mathbf{s}] \in \mathcal{R}_k$  and  $k > \tilde{K}$ . Then, either the algorithm is directly terminated in Step 5 or the remaining sets in  $\mathcal{R}_k$  are successively pruned in finitely many iterations until  $\mathcal{R}_k = \emptyset$ . ■

*Remark 1 (Relative Tolerance):* Algorithm 1 determines an  $\eta$ -optimal solution of (P), i.e., a feasible solution  $\bar{\mathbf{x}}$  of (P) that satisfies  $f(\bar{\mathbf{x}}) \geq f(\mathbf{x}) - \eta$  for all  $\mathbf{x} \in \mathcal{D}$ . Instead, by replacing all occurrences of “ $\gamma_k + \eta$ ” in Algorithm 1 with “ $(1 + \eta)\gamma_k$ ”, the tolerance  $\eta$  becomes relative to the optimal value and the algorithm terminates if the solution satisfies  $(1 + \eta)f(\bar{\mathbf{x}}) \geq f(\mathbf{x})$  for all  $\mathbf{x} \in \mathcal{D}$ . The necessary modifications of Theorem 1 are straightforward. ◇

*Remark 2 (Non-Uniqueness of MMP Representations):* The proof of Theorem 1 is valid for any  $F$  satisfying (1) and (2). Thus, the non-uniqueness of the MMP representation  $F$  does not impair the convergence proof of Algorithm 1. However, the actual choice of  $F$  has an impact on the tightness of the obtained bounds and, thus, on the convergence speed. For the example in (6), we can calculate  $\tilde{F}(\mathbf{s}, \mathbf{r}) - F(\mathbf{s}, \mathbf{r}) = \sum_{i=1}^N (s_i - r_i) \geq 0$  to see that the MMP representation  $F$  never leads to worse bounds than the alternative  $\tilde{F}$ . A practical example in which the influence of the choice of the MMP representation on the convergence speed can be observed is studied in detail in Section IV-A, and a more general discussion of this important aspect is provided in Section V-A. ◇

### A. Properties of $\mathcal{D}$ and Implementation of the Feasibility Check

To implement the BRB method as described in Algorithm 1, it is necessary to have means to perform the feasibility check in Step 3 (and Step 4). Let us first discuss cases in which this can be easily done. Afterwards, we comment on workarounds that can be used if no conclusive feasibility check is available.

A conclusive feasibility test based solely on the properties of MM functions is not possible. Consider a feasible set

$$\mathcal{D} = \{\mathbf{x} \mid G_i(\mathbf{x}, \mathbf{x}) \leq 0, i = 1, \dots, m\} \quad (21)$$

where  $G_i$  satisfies (2a) and (2b). These properties lead to the following sufficient conditions for (in-)feasibility of  $\mathcal{M}$ .

*Proposition 1:* Let  $\mathcal{M} = [\mathbf{r}, \mathbf{s}]$  and  $\mathcal{D}$  as in (21). Then,

$$\forall i \in \{1, \dots, m\} : G_i(\mathbf{s}, \mathbf{r}) \leq 0 \Rightarrow \mathcal{M} \cap \mathcal{D} = \mathcal{M} \neq \emptyset \quad (22a)$$

$$\exists i \in \{1, \dots, m\} : G_i(\mathbf{r}, \mathbf{s}) > 0 \Rightarrow \mathcal{M} \cap \mathcal{D} = \emptyset. \quad (22b)$$

*Proof:* From (2a) and (2b),  $G_i(\mathbf{x}, \mathbf{x}) \leq G_i(\mathbf{s}, \mathbf{r})$  and  $G_i(\mathbf{x}, \mathbf{x}) \geq G_i(\mathbf{r}, \mathbf{s})$  for all  $\mathbf{x} \in \mathcal{M}$ . Thus, if  $G_i(\mathbf{s}, \mathbf{r}) \leq 0$  for all  $i = 1, \dots, m$ , then  $G_i(\mathbf{x}, \mathbf{x}) \leq 0$  for all  $i$  and  $\mathbf{x} \in \mathcal{M}$ . Hence, (22a). Similarly, if  $G_i(\mathbf{r}, \mathbf{s}) > 0$  for some  $i = 1, \dots, m$ , then also  $G_i(\mathbf{x}, \mathbf{x}) > 0$  for this  $i$  and all  $\mathbf{x} \in \mathcal{M}$ . Thus,  $\mathbf{x} \notin \mathcal{D}$  and (22b) holds. ■

In general, there exist boxes for which neither (22a) nor (22b) holds, so that it remains open whether  $\mathcal{M}$  contains a feasible point. However, we could consider the special case where

$$G_i \left( \sum_{j \in \mathcal{I}} x_j \mathbf{e}_j, \sum_{k \in \mathcal{I}^c} y_k \mathbf{e}_k \right) = G_i(\mathbf{x}, \mathbf{y}), \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n \quad (23)$$

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  $g_i(\mathbf{x}) = G_i(\mathbf{x}, \mathbf{x})$  is nondecreasing in the variables  $x_j, j \in \mathcal{I}$ , and nonincreasing in the remaining variables  $x_k, k \in \mathcal{I}^c$ . In this case, the following proposition is a simple feasibility test based on MM properties.

*Proposition 2:* Let  $\mathcal{M} = [\mathbf{r}, \mathbf{s}]$  and  $\mathcal{D}$  be defined as in (21) by MM functions  $G_i(\mathbf{x}, \mathbf{y})$  satisfying (2) and (23). Then,  $\mathcal{M} \cap \mathcal{D} \neq \emptyset$  if and only if  $G_i(\mathbf{r}, \mathbf{s}) \leq 0$  for all  $i = 1, \dots, m$ . In that case,  $\sum_{j \in \mathcal{I}} r_j \mathbf{e}_j + \sum_{k \in \mathcal{I}^c} s_k \mathbf{e}_k \in \mathcal{M} \cap \mathcal{D}$  with  $\mathcal{I}$  and  $\mathcal{I}^c$  as in (23).

*Proof:* Let  $\boldsymbol{\xi} = (r_1, \dots, r_{\kappa}, s_{\kappa+1}, \dots, s_K)^T$ . Then, for all  $i$  and due to (23),  $G_i(\boldsymbol{\xi}, \boldsymbol{\xi}) = G_i(\mathbf{r}, \mathbf{s})$ . Thus, if  $G_i(\mathbf{r}, \mathbf{s}) \leq 0$ , 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 (22b) follows  $\mathcal{M} \cap \mathcal{D} \neq \emptyset \Rightarrow G_i(\mathbf{r}, \mathbf{s}) \leq 0$ . ■

*Corollary 1:* Let  $\mathcal{M} = [\mathbf{r}, \mathbf{s}]$  and  $\mathcal{D}$  be a normal set, i.e.,

$$\mathcal{D} = \{\mathbf{x} \mid g_i(\mathbf{x}) \leq 0, i = 1, \dots, m\} \quad (24)$$

with  $g_i$  being nondecreasing functions. Then,  $\mathcal{D} \cap \mathcal{M} \neq \emptyset$  if and only if  $g_i(\mathbf{r}) \leq 0$  for all  $i = 1, \dots, m$ .

*Corollary 2:* Let  $\mathcal{M} = [\mathbf{r}, \mathbf{s}]$  and  $\mathcal{D}$  be a conormal set, i.e.,

$$\mathcal{D} = \{\mathbf{x} \mid h_i(\mathbf{x}) \geq 0, i = 1, \dots, m\} \quad (25)$$

with  $h_i$  being nondecreasing functions. Then,  $\mathcal{D} \cap \mathcal{M} \neq \emptyset$  if and only if  $h_i(\mathbf{r}) \geq 0$  for all  $i = 1, \dots, m$ .

Proposition 2 and Corollaries 1 and 2 cover a wide range of feasible sets. However, none of these properties is necessary as long as we have other means to perform a feasibility check. For instance, consider the case where we can express  $\mathcal{D}$  by

$$g_i(\mathbf{x}) \leq 0, i = 1, \dots, m, \quad h_j(\mathbf{x}) = 0, j = 1, \dots, l \quad (26)$$

where  $g_i$  are convex functions and  $h_j$  are affine functions. In this case,  $\mathcal{D}$  is a closed convex set and the feasibility check can be solved with polynomial complexity by standard tools from convex optimization [52], [53]. In particular, (26) includes polyhedral sets where  $g_i$  are affine functions.

Let us now discuss workarounds for cases where a feasibility test as described above is not available, but the constraints can be written as MM functions as in (21). The first possible workaround is to alter Algorithm 1 such that, in Step 3, a feasible point is only required if available, and, in Step 4, boxes are only pruned if (22b) is met. Due to this modification, we can

use Proposition 2 instead of a fully conclusive feasibility test. Then, by a similar argument as in [29, Prop. 7.4] and according to the proof of Theorem 1, there exists an infinite decreasing sequence of sets  $\{\mathcal{M}_{k_q}\}_q$  such that  $G_i(\mathbf{r}^{k_q}, \mathbf{s}^{k_q}) \leq 0$ , for all  $i$  and  $q = 1, 2, \dots$ . Since  $\text{diam } \mathcal{M}_{k_q} \rightarrow 0$ ,  $\mathbf{r}^{k_q}$  and  $\mathbf{s}^{k_q}$  approach a common limit point  $\mathbf{x}$ . Due to the continuity of  $G_i$ , this point satisfies  $G_i(\mathbf{x}, \mathbf{x}) \leq 0$  for all  $i$ . Altering Theorem 1's proof accordingly, it can be shown that the modified algorithm is infinite and, whenever it generates an infinite sequence  $\{\mathbf{r}^k\}$ , every accumulation point of this sequence is a global optimum. Please refer to [50, Sec. 6.3.1] for more details.

In practice, an infinite algorithm often converges in finite time (see the numerical example in Section IV-F), but there are no theoretical guarantees for this, and for some problem instances, the resulting algorithm can have very slow convergence.

Another widely accepted workaround is to accept an  $\eta$ -optimal point that is approximately feasible as solution, i.e., a point  $\bar{\mathbf{x}}$  satisfying  $f(\bar{\mathbf{x}}) \geq f(\mathbf{x}) - \eta$  for all  $\mathbf{x} \in \mathcal{D}$  and  $G_i(\bar{\mathbf{x}}, \bar{\mathbf{x}}) \leq \varepsilon$  for all  $i = 1, \dots, m$  and some small  $\varepsilon > 0$ . Such a point is called  $(\varepsilon, \eta)$ -approximate optimal solution.

This second method restores finite convergence, but gives rise to numerical problems. If  $\varepsilon$  is not chosen sufficiently small, the  $(\varepsilon, \eta)$ -approximate optimal solution might be far from the true optimum. The issue is that it is usually unclear how small is "sufficient" to guarantee a good approximate solution [30, Sec. 7.5]. Even worse, if the true optimum is an isolated point,<sup>8</sup> any change in the tolerances  $\varepsilon, \eta$  can lead to drastic changes in the  $(\varepsilon, \eta)$ -approximate optimal solution [55, Sec. 4]. We thus generally do not recommend the  $(\varepsilon, \eta)$ -approximate approach.

A more suitable method for optimization problems with such "hard" feasible sets is the successive incumbent transcending scheme from [55], which algorithmically excludes all isolated feasible points and provides an elegant solution to the feasibility check issues. An optimization framework based on this scheme is published in [54] along with source code, and could be combined with the MMP concept. Besides its numerical stability, this scheme also improves efficiency for problems that are only nonconvex due to some of their variables [54].

We stress the fact that none of the above workarounds is required if a fully conclusive feasibility test can be implemented (preferably with low computational complexity), so that the unmodified algorithm as stated in Algorithm 1 can be used.

### B. The Reduction Procedure in Step 2 of Algorithm 1

In Step 2 of Algorithm 1, each box  $\mathcal{M} \in \mathcal{P}_k$  is replaced by a smaller box  $\mathcal{M}'$  that still contains all feasible points that might improve the current best known solution. This step speeds up the convergence since smaller boxes result in tighter bounds. 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 Step 2 speeds up Algorithm 1 or not. Hence, an important observation is that Step 2 is entirely optional since choosing

<sup>8</sup>A feasible point is called isolated if it is at the center of a ball containing no other feasible points. Please refer to [54] for a numerical example showing the existence of isolated feasible points in a radio resource allocation problem.

$\mathcal{M}' = \mathcal{M}$  satisfies the above condition. Moreover, note that (19) is also satisfied if  $\mathcal{M}'$  satisfies  $(\mathcal{M} \setminus \mathcal{M}') \cap \mathcal{D} = \emptyset$ .

For  $\mathcal{D}$  convex (or even linear), we refer the reader to the vast literature on convex (or linear) optimization regarding possible implementations of the reduction. Here, we just mention the most straightforward approach, namely to solve the convex (linear) optimization problems

$$r'_i = \min_{\mathbf{x} \in \mathcal{M} \cap \mathcal{D}} x_i \quad s'_i = \max_{\mathbf{x} \in \mathcal{M} \cap \mathcal{D}} x_i \quad (27)$$

for all  $i = 1, \dots, n$ , and let  $\mathcal{M}' = [\mathbf{r}', \mathbf{s}']$ .

For a feasible set defined by MM constraints as in (21), the reduction can be carried out in a similar fashion as for DM programming problems [30, Sec. 11.2.1]. Let  $\mathcal{M} = [\mathbf{r}, \mathbf{s}]$  and observe from (22b) that if, for some  $i = 1, \dots, m$ ,  $G_i(\mathbf{r}, \mathbf{s}) > 0$ , then  $\mathcal{M} \cap \mathcal{D} = \emptyset$  and  $\mathcal{M}' = \emptyset$ . Moreover, if  $F(\mathbf{s}, \mathbf{r}) \leq \gamma_k$ , then  $\{\mathbf{x} \in \mathcal{M} \mid F(\mathbf{x}, \mathbf{x}) > \gamma_k\} = \emptyset$  and  $\mathcal{M}' = \emptyset$  satisfies (19). Otherwise, i.e., if  $G_i(\mathbf{r}, \mathbf{s}) \leq 0$  for all  $i$  and  $F(\mathbf{s}, \mathbf{r}) \geq \gamma_k$ , let  $\mathcal{M}' = [\mathbf{r}', \mathbf{s}']$  with

$$\mathbf{r}' = \mathbf{s} - \sum_{i=1}^n \alpha_i (s_i - r_i) \mathbf{e}_i, \quad \mathbf{s}' = \mathbf{r}' + \sum_{i=1}^n \beta_i (s_i - r'_i) \mathbf{e}_i \quad (28)$$

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

$$\alpha_i = \sup \left\{ \alpha \in [0, 1] \mid F(\mathbf{s} - \alpha(s_i - r_i) \mathbf{e}_i, \mathbf{r}) > \gamma_k, \quad (29a)$$

$$G_j(\mathbf{r}, \mathbf{s} - \alpha(s_i - r_i) \mathbf{e}_i) \leq 0, j = 1, \dots, m \right\}$$

$$\beta_i = \sup \left\{ \beta \in [0, 1] \mid F(\mathbf{s}, \mathbf{r}' + \beta(s_i - r'_i) \mathbf{e}_i) > \gamma_k, \quad (29b)$$

$$G_j(\mathbf{r}' + \beta(s_i - r'_i) \mathbf{e}_i, \mathbf{s}) \leq 0, j = 1, \dots, m \right\}.$$

The proof that (19) holds for this reduction procedure is an extension of [30, Lem. 11.1] and can be found in [50, Sec. 6.4]. Equations (29a) and (29b) can be implemented efficiently by a low precision bisection. It is important though that the obtained solutions are greater (or equal) than the true  $\alpha_i, \beta_i$ . Otherwise, feasible solutions might be lost.

## IV. APPLICATION EXAMPLES

To demonstrate the usefulness and exceptional performance of the proposed MMP approach, we consider examples of various applications in the area of signal processing for communications. Where available, existing globally optimal approaches are discussed and compared to the proposed framework. Run time comparisons show tremendous gains over the state-of-the-art solutions. For the example considered in Section IV-A, we even provide an analytical justification why the proposed method outperforms previous DM formulations.

The complete source code is available on GitHub [46]. All reported performance results were obtained on Intel Haswell nodes with Xeon E5-2680 v3 CPUs running at 2.50 GHz.

The presented applications are only meant as examples: we are aware of further optimization problems for which the MMP framework can be useful, and we are convinced that further applications can also be identified in other research areas.

### A. Weighted Sum Rates in the $K$ -User Interference Channel

As a first application example, we consider weighted sum rate maximization in a  $K$ -user interference channel (IC) under the assumption that the input signals are proper Gaussian and that interference is treated as noise. Letting  $\alpha_k$  denote the gains of the intended channels, and  $\beta_{kj}$  with  $j \neq k$  the gains of the unintended channels, we can write the achievable rates as

$$r_k = \log_2 \left( 1 + \frac{\alpha_k p_k}{\sigma^2 + \sum_{j=1}^K \beta_{kj} p_j} \right) \quad (30)$$

where  $\sigma^2$  is the noise variance and  $p_k$  is the transmit power of user  $k$ . Due to the possibility of modeling self-interference or hardware impairments by choosing  $\beta_{kk} \neq 0$ , this formulation is more general than in some of the previous works mentioned below. Note that there are several other system models for which the rate expressions can be brought to a form equivalent to (30), e.g., certain massive MIMO, cellular, and relay-aided scenarios [14], [56].

The weighted sum rate maximization problem with minimum rate constraints is

$$\max_{0 \leq p \leq P} \sum_{k=1}^K w_k r_k \quad \text{s. t.} \quad r_k \geq R_{\min,k}, k = 1, \dots, K. \quad (31)$$

For this problem, various approaches can be found in the literature. In the MAPEL framework [33], [35], the problem is parametrized in terms of SINRs as

$$\max_{(\gamma_k)_{\forall k} \in \mathcal{G}} \sum_{k=1}^K w_k \log_2(1 + \gamma_k) \quad \text{s. t.} \quad \gamma_k \geq \gamma_{\min,k}, \forall k \quad (32)$$

where the set of possible SINR combinations  $\mathcal{G}$  is approximated from the outside by means of the PA [28] until the global optimal solution is found. Instead, the authors of [7] formulate the problem as

$$\max_{\rho \in \mathcal{R}} \sum_{k=1}^K w_k \rho_k \quad \text{s. t.} \quad \rho_k \geq R_{\min,k}, k = 1, \dots, K \quad (33)$$

where  $\mathcal{R}$  is the achievable rate region defined by (30) and the power constraints. The rate region is then approximated by the PA. This special case of the framework in [38], [39] is termed as ‘‘Ratespace PA’’ in the numerical results below. A disadvantage of both methods is that in every iteration an inner problem with considerable computational complexity has to be solved to project points from outside the feasible set onto its boundary.

Another approach to apply the monotonic optimization framework [28] is to rewrite the rates as DM functions

$$r_k = \log_2 \left( \alpha_k p_k + \sigma^2 + \sum_{j=1}^K \beta_{kj} p_j \right) - \log_2 \left( \sigma^2 + \sum_{j=1}^K \beta_{kj} p_j \right). \quad (34)$$

This problem can either be solved via the PA by introducing an auxiliary variable [28], [57] (termed as ‘‘PA’’) or directly via the BB method for DM problems [29] (‘‘BB DM’’).<sup>9</sup>

<sup>9</sup>The BB algorithm in [29] includes a reduction step similar to the one described in Section III-B. However, for the problem under consideration, omitting the reduction step leads to faster performance.

Among the state-of-the-art, this BB approach is most closely related to the proposed MMP framework. Indeed, finding an MMP representation of (34) is straightforward as explained in Section II: all powers in the first log-term of (34) are replaced by nondecreasing variables  $x_i$ , and all powers in the second log-term are replaced by nonincreasing variables  $y_i$ . However, we will show below in (38) that this leads to looser bounds and, thus, slower average convergence speed than the new MMP representation proposed below.

By calculating the partial derivatives of  $r_k$  in (30), it is easy to verify that  $r_k$  is nondecreasing in  $p_k$  (regardless of the value of  $p_j$ ) and nonincreasing in  $p_j$  for  $j \neq k$  (regardless of the value of  $p_k$ ). Thus, the MM function

$$R_k(\mathbf{x}, \mathbf{y}) = \log_2 \left( 1 + \frac{\alpha_k x_k}{\sigma^2 + \beta_{kk} x_k + \sum_{j \neq k} \beta_{kj} y_j} \right) \quad (35)$$

is an MMP representation of (30). Using (10) and (12), an MMP representation of the objective of (31) is obtained as

$$F(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^K w_k R_k(\mathbf{x}, \mathbf{y}). \quad (36)$$

An MMP formulation of the feasible set is given by (21) with

$$G_k(\mathbf{x}, \mathbf{y}) = R_{\min,k} - R_k(\mathbf{y}, \mathbf{x}), k = 1, \dots, K. \quad (37)$$

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 [29]. Consider a single rate  $r_k$  and the bounds obtained by (34) and (35) evaluated for a box  $[\mathbf{x}, \mathbf{y}]$ . The difference between (35) and (34) is

$$R_k(\mathbf{x}, \mathbf{y}) - \log_2 \left( \alpha_k x_k + \sigma^2 + \sum_{j=1}^K \beta_{kj} x_j \right) - \log_2 \left( \sigma^2 + \sum_{j=1}^K \beta_{kj} y_j \right) \quad (38a)$$

$$= \log_2 \left( \frac{\alpha_k x_k + \sigma^2 + \beta_{kk} x_k + \sum_{j \neq k}^K \beta_{kj} y_j}{\alpha_k x_k + \sigma^2 + \beta_{kk} x_k + \sum_{j \neq k}^K \beta_{kj} x_j} \right) + \log_2 \left( \frac{\sigma^2 + \beta_{kk} y_k + \sum_{j \neq k}^K \beta_{kj} y_j}{\sigma^2 + \beta_{kk} x_k + \sum_{j \neq k}^K \beta_{kj} y_j} \right) \leq 0 \quad (38b)$$

where we have exploited that the numerator is greater or equal than the denominator in both fractions due to  $\mathbf{y} \geq \mathbf{x}$ . This shows that the MMP bound is always tighter than the DM bound.<sup>10</sup>

The results in Fig. 1 illustrate this theoretical intuition numerically. The plot displays the average number of iterations required by Algorithm 1 to solve (31) versus the number of variables. Each data point is averaged over 100 independent and identically distributed (i.i.d.) channel realizations with  $\alpha_i = |\alpha'_i|^2$  and  $\beta_{ij} = |\beta'_{ij}|^2$  where  $\alpha'_i, \beta'_{ij} \sim \mathcal{CN}(0, 1)$  for all  $i$  and  $j \neq i$ . Further,  $\eta = 0.01$ ,  $\sigma^2 = 0.01$ ,  $P_k = 1$ ,  $w_k = 1$ ,  $\beta_{kk} = 0$ ,  $R_{\min,k} = 0$  for all  $k$ , and no reduction is used. It can be observed that MMP with best-first selection (16) and MM function (35), labeled as ‘‘MMP,’’ requires three orders of

<sup>10</sup>For problems without self-interference, the above comparison simplifies as the second logarithm in (38b) vanishes, but the conclusion remains the same.

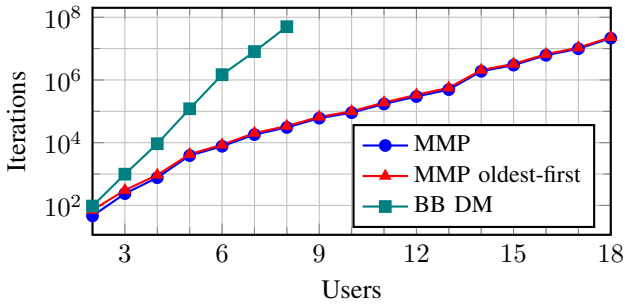


Fig. 1. Average number of iterations in Algorithm 1 to solve (31) with bounds obtained by (35) with best- and oldest-first selection, respectively, and (34) with best-first selection. Results are averaged over 100 i.i.d. channel realizations.

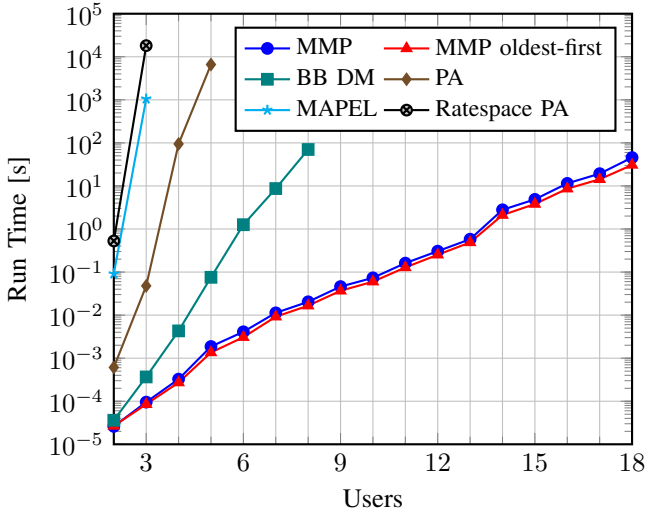


Fig. 2. Average run time required to solve (31) with different algorithms. Results are averaged over 100 different i.i.d. channel realizations.

magnitude less iterations than the same algorithm with MM function obtained from (34), named “BB DM,” to solve (31) with  $K = 8$  variables. From a practical perspective, this means that the MMP framework is able to solve (31) with 18 variables in the same time that state-of-the-art monotonic programming requires for 8 variables. A further observation from Fig. 1 is that the oldest-first selection rule (17), labeled as “MMP oldest-first,” requires only slightly more iterations to converge than the best-first rule. The benefits of the 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 BB algorithms and the PA, comparing the number of iterations is meaningless: the PA typically requires much less iterations but each iteration takes much longer than in a BB algorithm. Thus, we resort to measuring the average run time of the algorithms in the C++ implementation available at [46]. We have taken great care to implement the state-of-the-art algorithms with the same rigor and amount of code optimization as the proposed method to make this benchmark as fair as possible. The average run time and memory consumption of all discussed approaches is displayed in Figs. 2 and 3, respectively. The same parameters as in the computation of Fig. 1 were used.

First, observe that all algorithms scale both in run time and

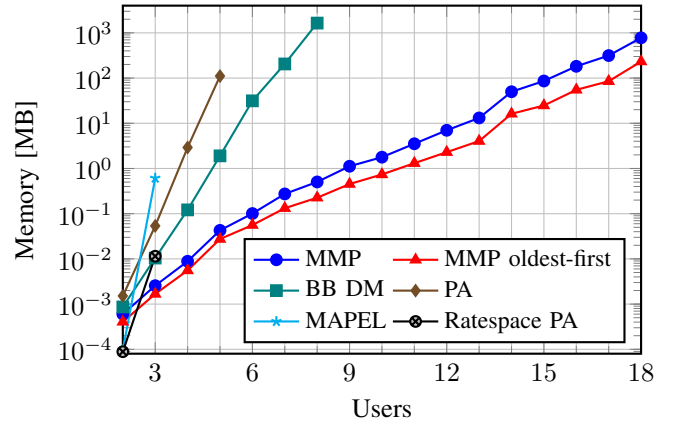


Fig. 3. Average memory consumption of different algorithms to solve (31). Results are averaged over 100 different i.i.d. channel realizations.

memory consumption exponentially in the number of variables. Since problem (31) is NP hard [58, Thm. 1], better asymptotic complexity is most likely 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. The proposed MMP framework solves problem (31) in considerably less time and memory requirements than all other state-of-the-art methods. The PA based methods all consume more memory than the BB based methods 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 our run time limit of 8 h. For the BB methods, the observations from Fig. 1 continue to hold in Fig. 2. From the memory consumption, it can be observed that good bounds are not only critical for fast convergence but also for memory efficiency. In this example, the MMP method was able to solve problems more than twice the size of the DM BB method 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× or 546 MB more at 18 variables. Further, observe that while requiring less iterations, the best-first rule has longer run times than the oldest-first rule. This can be explained from Fig. 3 since the memory consumption is directly proportional to the number of boxes in  $\mathcal{R}_k$ . The best-first rule is the mathematical description of a priority queue. While accessing the top-element in a priority queue has complexity  $O(1)$ , insertion has worst-case complexity  $O(\log n)$  [59, pp. 148–152], 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, i.e., a first in, first out (FIFO) list. Here, the insertion, deletion, and access to the front element all require constant time  $O(1)$  and do not grow with the number of stored elements.

### B. Energy Efficiency Optimization

The global energy efficiency (GEE) is a key performance metric for 5G and beyond networks measuring the network energy efficiency [5], [14], [36]. 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:

$$\text{GEE} = \frac{TB \sum_{k=1}^K r_k}{T(\phi^T \mathbf{p} + P_c)} = \frac{B \sum_{k=1}^K r_k}{\phi^T \mathbf{p} + P_c} \left[ \frac{\text{bit}}{\text{J}} \right], \quad (39)$$

where  $r_k$  is the achievable rate of link  $k$ ,  $B$  is the bandwidth,  $\phi \geq \mathbf{1}$  contains the inverses of the power amplifier efficiencies and  $P_c$  is a constant modeling the constant part of the circuit power consumption.

Maximizing the GEE for interference networks with treating interference as noise, i.e., where  $r_k$  is as in (30), results in the nonconvex fractional programming problem [36], [60]

$$\max_{\mathbf{0} \leq \mathbf{p} \leq \mathbf{P}} \frac{\sum_{k=1}^K r_k}{\phi^T \mathbf{p} + P_c} \quad (40)$$

where we have omitted the inessential constant  $B$  and minimum rate constraints that are already discussed in Section IV-A. As the objective includes the sum rate as a special case for  $\phi = \mathbf{0}$  and  $P_c = 1$ , this problem is also NP-hard due to [58, Thm. 1]. As already mentioned below (4), the state-of-the-art approach to solve (40) is to combine Dinkelbach's Algorithm [36], [49] with monotonic programming. This was first proposed in [25] and subsequently developed into the fractional monotonic programming framework in [37]. Dinkelbach's Algorithm solves (40) as a sequence of auxiliary problems

$$\max_{\mathbf{0} \leq \mathbf{p} \leq \mathbf{P}} \sum_{k=1}^K r_k - \lambda \phi^T \mathbf{p} + P_c \quad (41)$$

with non-negative parameter  $\lambda$ . Problem (41) can be solved by monotonic programming much in the same way as discussed in Section IV-A. While most works use the PA to solve (41) (e.g., [25], [37]), we have already demonstrated above that BB with DM bounds [29] outperforms the classical PA [28].

The MMP framework even allows to solve (40) without the need of Dinkelbach's Algorithm. An MMP representation of (39) can be obtained similar to (4). Specifically, with (35) and the identities in (10), (12) and (13), we obtain

$$F(\mathbf{x}, \mathbf{y}) = \frac{B \sum_{k=1}^K R_k(\mathbf{x}, \mathbf{y})}{\phi^T \mathbf{y} + P_c} \quad (42)$$

with  $R_k(\mathbf{x}, \mathbf{y})$  as in (35).

The run time performance of both algorithms is evaluated in Fig. 4 where  $\phi_k = 5$ , for all  $k$ , and  $P_c = 1$ . The remaining parameters were chosen as in Section IV-A. It can be observed that MMP requires significantly less time to solve (40) 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 (not displayed) scales almost identically to the run time with MMP using four orders of magnitude less memory than Dinkelbach's Algorithm for six variables. Besides showing much better run time and memory performance, the MMP method also guarantees an  $\eta$ -optimal solution. By contrast, Dinkelbach's Algorithm does not provide any guarantees on the solution quality since an inaccuracy of  $\eta$  in the inner solver might propagate to larger inaccuracies in the overall results.

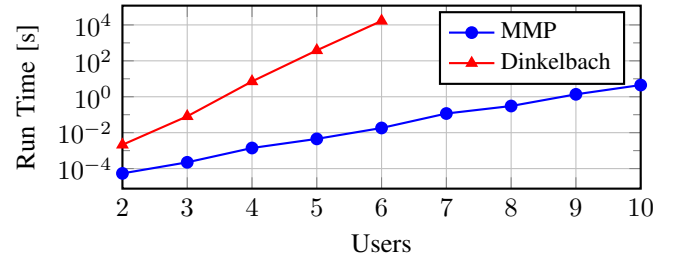


Fig. 4. Average run time to solve (40) with MMP and Dinkelbach's Algorithm where the inner problem is solved by Algorithm 1 with DM rate expressions (34). Results are averaged over 100 different i.i.d. channel realizations.

Other energy efficiency (EE) metrics can be maximized with the MMP framework in a similar manner. For example, in interference networks with rate function (30), the weighted minimum EE (WMEE) has the objective [37]

$$\text{WMEE} = \min_{k=1, \dots, K} w_k \frac{B r_k}{\phi_k p_k + P_{c,k}} \quad (43)$$

with nonnegative weights  $w_1, \dots, w_K$  and MMP representation

$$F_{\text{WMEE}}(\mathbf{x}, \mathbf{y}) = \min_{k=1, \dots, K} w_k \frac{B R_k(\mathbf{x}, \mathbf{y})}{\phi_k y_k + P_{c,k}}, \quad (44)$$

and the weighted sum EE (WSEE) has the objective [37]

$$\text{WSEE} = \sum_{k=1}^K w_k \frac{B r_k}{\phi_k p_k + P_{c,k}} \quad (45)$$

with MMP representation

$$F_{\text{WSEE}}(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^K w_k \frac{B R_k(\mathbf{x}, \mathbf{y})}{\phi_k y_k + P_{c,k}}. \quad (46)$$

The WMEE can be maximized similarly to the GEE with a combination of the Generalized Dinkelbach Algorithm and monotonic programming [37]. Instead, optimizing the WSEE with monotonic optimization is much more challenging since neither Dinkelbach's Algorithm nor its generalization are applicable. In [37], it is proposed to transform (45) into a single fractional program, i.e.,

$$\frac{\sum_{k=1}^K w_k B r_k \prod_{i \neq k} (\phi_i p_i + P_{c,i})}{\prod_{k=1}^K (\phi_k p_k + P_{c,k})} \quad (47)$$

and then apply fractional monotonic programming. While this works in theory, it is shown in [56] that this approach has very poor convergence. Instead, the MMP framework allows to directly optimize both metrics without cumbersome transformations and without using Dinkelbach's Algorithm or its generalized version.

### C. Proportional Fair Rate Optimization with Scheduling

The weighted sum rate utility in (31) can also be replaced by a utility function that accounts for the fairness between users, such as the proportional fair utility

$$U(r_1, \dots, r_K) = \sum_{k=1}^K \ln r_k. \quad (48)$$

In this context, a common approach (e.g., [34], [35], [38], [39]) is to increase the flexibility in the optimization by scheduling

different transmit strategies in multiple time slots and averaging the data rates, i.e.,

$$\max_{\substack{(0 \leq p_k^{(\ell)} \leq P_k)_{\forall k \forall \ell} \\ L \in \mathbb{N}, \tau \geq \mathbf{0}: \mathbf{1}^T \tau = 1}} U(\bar{r}_1, \dots, \bar{r}_K) \quad \text{s. t. } \bar{r}_k \geq R_{\min, k}, \forall k \quad (49)$$

with

$$\bar{r}_k = \sum_{\ell=1}^L \tau_\ell r_k^{(\ell)}, \quad r_k^{(\ell)} = \log_2 \left( 1 + \frac{\alpha_k p_k^{(\ell)}}{\sigma^2 + \sum_{j \neq k} \beta_{kj} p_j^{(\ell)}} \right). \quad (50)$$

In this application example, we restrict ourselves to the proportional fair utility (48) since this problem was shown to be NP hard [58] for all  $L \geq 3$  even though the utility is concave in the per-user rates.

In [34], [35], an algorithm called S-MAPEL for nondecreasing utility functions was proposed. The approach is based on the PA and makes use of the reformulation (32) as well as of the observation that the rates are nondecreasing functions of the time fractions  $\tau_\ell$ . By arguing that no more than  $L = K + 1$  strategies are necessary due to the Carathéodory theorem, the approach from [34], [35] uses  $LK = (K + 1)K$  optimization variables in total.<sup>11</sup> This leads to a significant computational complexity. A second disadvantage of this approach is as follows. The optimizer of (49) is not unique since any re-indexing of the time index  $\ell$  leads to an optimal solution as well, but when directly solving (49) this inherent symmetry is not exploited. The authors of [34], [35] thus proposed an accelerated algorithm called A-S-MAPEL which employs a heuristic (with an additional tolerance parameter  $\varepsilon_{\text{tol}}$ ) to exploit the symmetry, but the resulting strategy is no longer guaranteed to be  $\eta$ -optimal.

We focus on the following alternative method for concave utility functions from [38], [39], which avoids increasing the number of variables at the cost of having to solve a series of monotonic optimization problems. To obtain an efficient algorithm, we combine this approach with the MMP framework.

We rewrite problem (49) as

$$\max_{\substack{(0 \leq p_k^{(\ell)} \leq P_k)_{\forall k \forall \ell} \\ (\rho_k \geq R_{\min, k})_{\forall k} \\ L \in \mathbb{N}, \tau \geq \mathbf{0}: \mathbf{1}^T \tau = 1}} U(\rho_1, \dots, \rho_K) \quad \text{s. t. } \bar{r}_k \geq \rho_k \quad \forall k \quad (51)$$

and consider the Lagrangian dual problem

$$\min_{\mu \geq \mathbf{0}} \max_{\substack{(0 \leq p_k^{(\ell)} \leq P_k)_{\forall k \forall \ell} \\ (\rho_k \geq R_{\min, k})_{\forall k} \\ L \in \mathbb{N}, \tau \geq \mathbf{0}: \mathbf{1}^T \tau = 1}} U(\rho_1, \dots, \rho_K) + \sum_{k=1}^K \mu_k (\bar{r}_k - \rho_k) \quad (52)$$

where  $\bar{r}_k$  depends on the optimization variables via (50). Since averaging the rates can be interpreted as optimizing over the convex hull of the achievable rate region, (51) can be rewritten as a convex program to show that strong duality holds [38], [39], i.e., (52) has the same optimal value as (51).

<sup>11</sup>In fact, the number of strategies can be reduced to  $L = K$  due to an extension to the Carathéodory Theorem discussed in [61], yielding a total number of  $LK = K^2$  variables.

We note that  $\mathbf{p}^{(\ell)}$  can be optimized separately for each  $\ell$ , and that these inner problems are all equivalent, i.e.,

$$\max_{\substack{(0 \leq p_k^{(\ell)} \leq P_k)_{\forall k \forall \ell} \\ L \in \mathbb{N}, \tau \geq \mathbf{0}: \mathbf{1}^T \tau = 1}} \mu^T \sum_{\ell=1}^L \tau_\ell \mathbf{r}^{(\ell)} \quad (53a)$$

$$= \max_{L \in \mathbb{N}, \tau \geq \mathbf{0}: \mathbf{1}^T \tau = 1} \underbrace{\sum_{\ell=1}^L \tau_\ell}_{=1} \max_{(0 \leq p_k^{(\ell)} \leq P_k)_{\forall k}} \mu^T \mathbf{r}^{(\ell)} \quad (53b)$$

which implies that the choice of  $L$  and  $\tau$  in the dual problem is arbitrary. Thus, the dual problem (52) can be rewritten as

$$\min_{\mu \geq \mathbf{0}} u_\mu(\boldsymbol{\rho}^*(\boldsymbol{\mu})) + v_\mu(\mathbf{p}^*(\boldsymbol{\mu})) \quad (54)$$

where

$$\boldsymbol{\rho}^*(\boldsymbol{\mu}) = \arg \max_{(\rho_k \geq R_{\min, k})_{\forall k}} u_\mu(\boldsymbol{\rho}), \quad u_\mu(\boldsymbol{\rho}) = U(\boldsymbol{\rho}) - \boldsymbol{\mu}^T \boldsymbol{\rho} \quad (55a)$$

$$\mathbf{p}^*(\boldsymbol{\mu}) = \arg \max_{(0 \leq p_k \leq P_k)_{\forall k}} v_\mu(\mathbf{p}), \quad v_\mu(\mathbf{p}) = \boldsymbol{\mu}^T \mathbf{r}. \quad (55b)$$

In total, we have to solve three optimization problems in (54). The outer minimization is a convex problem in the dual variables  $\boldsymbol{\mu}$  and can be solved by the cutting plane method [62], [63] which successively refines outer approximations

$$\min_{\mu \geq \mathbf{0}, z \in \mathbb{R}} z \quad (56a)$$

$$\text{s. t. } z \geq u_\mu(\boldsymbol{\rho}^{(\ell)}) + v_\mu(\mathbf{p}^{(\ell)}) \quad \forall \ell \in \{1, \dots, L\}. \quad (56b)$$

For given constant vectors  $\boldsymbol{\rho}^{(\ell)}$  and  $\mathbf{r}^{(\ell)}$ , this is a linear program in  $\boldsymbol{\mu}$  and  $z$ . By solving for the optimal  $\boldsymbol{\mu}^*$ , setting  $(\boldsymbol{\rho}^{(L+1)}, \mathbf{p}^{(L+1)}) = (\boldsymbol{\rho}^*(\boldsymbol{\mu}^*), \mathbf{p}^*(\boldsymbol{\mu}^*))$ , and incrementing  $L$ , a refined approximation is obtained. In every iteration, a feasible approximate solution to the primal problem (51) can be recovered by solving the dual linear program of (56). These solutions converge from below to the global optimum [62, Sec. 6.5]. Note that primal recovery implicitly performs the convex hull operation corresponding to the rate averaging in (50) if needed [39, Sec. 3.3.2]. In addition, each iteration delivers a feasible value of the dual problem in (54), which acts as an upper bound to the global optimum of (51). As a termination criterion, we thus check whether the difference of these values is below a predefined accuracy threshold  $\varepsilon_{\text{CP}}$ .

In each iteration of the cutting plane method, evaluating  $\boldsymbol{\rho}^*$  and  $\mathbf{p}^*$  requires solving the inner problems (55). The first maximization (55a) is a convex program due to the assumption of a concave utility. In the special case of the proportional fair utility (48) it can even be solved in closed form.

The challenging nonconvex problem (55b) is a weighted sum rate maximization, which can be tackled by any of the methods discussed in Section IV-A. In [38], [39], it was proposed to apply the PA with the rates as optimization variables. Motivated by the run time comparison from Section IV-A, we instead use Algorithm 1 together with the reduction procedure from Section III-B. Combining this approach with the cutting plane method for the outer problem, we get the guarantee that the obtained solution lies at most  $\eta + \varepsilon_{\text{CP}}$  away from the global optimum, i.e., it is  $(\eta + \varepsilon_{\text{CP}})$ -optimal.

For a run time comparison using the implementation in [46], we reconsider the example from [34] with  $K = 4$  interfering links and channel gains derived from a path loss model for the network topology given in [34, Fig. 5]. As in [34], we maximize the proportional fair utility (48) without any constraints on the per-user rates. Since the original S-MAPEL algorithm did not converge within a reasonable amount of time, we use the A-S-MAPEL heuristic with accuracy  $\eta = 10^{-2}$  and  $\varepsilon_{\text{tol}} = 10^{-3}$ . Unlike S-MAPEL, this accelerated heuristic cannot give a rigorous guarantee for the quality of the obtained solutions [34], but we can use its run time of 3146 seconds as a (very loose) lower bound for the actual run time of S-MAPEL. Instead, the proposed combination of the cutting plane algorithm and the MMP framework with total tolerance of  $\eta + \varepsilon_{\text{CP}} = 9 \cdot 10^{-3} + 1 \cdot 10^{-3} = 10^{-2}$  converged in only 1.77 seconds.

#### D. Coded Time-Sharing and Rate Balancing

The combination of a Lagrangian dual approach and the MMP framework can be extended to solve several other problems. For instance, we can consider coded time-sharing [64] where not only the rates but also the transmit powers are averaged. In this case, we have to dualize the resulting average power constraints  $\sum_{\ell=1}^L \tau_{\ell} p_k^{(\ell)} \leq P_k$  in addition to the rate constraints. Moreover, we could replace the fairness optimization by a so-called rate balancing problem, which can be used to characterize the Pareto boundary of the rate region [65] and to guarantee the quality of service of all users.

As an example, let us combine both mentioned modifications in an IC under the assumptions of Gaussian inputs, coded time-sharing, and treating interference as noise. The resulting rate balancing problem with coded time-sharing is

$$\max_{\substack{(p_k^{(\ell)} \geq 0)_{\forall k, \forall \ell} \\ L \in \mathbb{N}, R \in \mathbb{R} \\ \tau \geq \mathbf{0}: \mathbf{1}^T \tau = 1}} R \quad \text{s. t.} \quad \sum_{\ell=1}^L \tau_{\ell} r_k^{(\ell)} \geq \rho_k R, \quad \forall k \quad (57a)$$

$$\sum_{\ell=1}^L \tau_{\ell} p_k^{(\ell)} \leq P_k, \quad \forall k \quad (57b)$$

for given relative rate targets  $\rho_k$ ,  $k = 1, \dots, K$ , and with  $r_k^{(\ell)}$  from (50). After introducing dual variables  $\boldsymbol{\mu}$  and  $\boldsymbol{\lambda}$  for the rate constraints and power constraints, respectively, and performing some reformulations similar to the ones in Section IV-C, the dual problem of (57) can be written as [48]

$$\min_{\substack{\boldsymbol{\mu} \geq \mathbf{0}, \boldsymbol{\lambda} \geq \mathbf{0} \\ \boldsymbol{\rho}^T \boldsymbol{\mu} = 1}} \left( \sum_{k=1}^K \lambda_k P_k + \max_{(p_k \geq 0)_{\forall k}} (\boldsymbol{\mu}^T \mathbf{r} - \boldsymbol{\lambda}^T \mathbf{p}) \right). \quad (58)$$

The inner maximization is no longer a pure weighted sum rate problem, but using the MM function

$$F(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^K (\mu_k R_k(\mathbf{x}, \mathbf{y}) - \lambda_k y_k) \quad (59)$$

it can still be solved via the MMP framework.

In [48], this problem was considered for the special case of a two-user single-input/single-output IC, and a BB solution was proposed for the inner maximization. In fact, this solution can be considered as a special case of the MMP framework with

MM function (59). For further details and numerical simulations of the rate balancing problem, the reader is referred to [48].

#### E. Multiantenna Interference Channels

The MMP framework can also be used in multiantenna scenarios. In a single-input/multiple-output (SIMO) IC

$$\mathbf{y}_k = \sum_{j=1}^K \mathbf{h}_{kj} x_j + \sigma_k^2 \quad (60)$$

the achievable rates with Gaussian codebooks, interference treated as noise, and without self-interference can be expressed as [8]

$$r_k = \log_2 \left( 1 + p_k \mathbf{h}_{kk}^H \left( \mathbf{I}_{M_k} + \sum_{j \neq k} p_j \mathbf{h}_{kj} \mathbf{h}_{kj}^H \right)^{-1} \mathbf{h}_{kk} \right) \quad (61)$$

where  $M_k$  is the number of antennas at receiver  $k$  and  $\mathbf{I}_{M_k}$  is the identity matrix of this size. By replacing the rate (30) by (61), we can formulate the weighted sum rate maximization (Section IV-A), the energy efficiency optimizations (Section IV-B), the scheduling problem (Section IV-C), and the rate balancing problem (Section IV-D) for the SIMO IC.

By calculating the partial derivatives with respect to  $\mathbf{x}$  and  $\mathbf{y}$  in order to study monotonicity, it can be verified that

$$R_k(\mathbf{x}, \mathbf{y}) = \log_2 \left( 1 + x_k \mathbf{h}_{kk}^H \left( \mathbf{I}_{M_k} + \sum_{j \neq k} y_j \mathbf{h}_{kj} \mathbf{h}_{kj}^H \right)^{-1} \mathbf{h}_{kk} \right) \quad (62)$$

is a MMP representation of (61). We can thus directly apply the MMP framework to solve all of the above-mentioned problems in the SIMO IC.

For the MISO IC with multiple antennas at the transmitter side, the optimization is more involved since transmit covariance matrices or beamforming vectors need to be designed instead of transmit powers. In the following, we present a beamformer-based method for the two-user MISO IC

$$y_k = \mathbf{h}_{kk}^H \mathbf{x}_k + \mathbf{h}_{kj}^H \mathbf{x}_j + \eta_k \quad (63)$$

with  $j = 3 - k$ . The transmit signals  $\mathbf{x}_k = \sqrt{p_k} \mathbf{b}_k s_k$  are generated from scalar Gaussian inputs  $s_k \sim \mathcal{CN}(0, 1)$ , where  $\mathbf{b}_k$  is a normalized beamforming vector with  $\|\mathbf{b}_k\| = 1$ .

The approach is based on [32], which uses parameters  $\zeta_k$  to construct a convex combination of the maximum ratio transmission (MRT) beamformer and the zero-forcing (ZF) beamformer, which is provably sufficient to parameterize all Pareto-optimal transmit strategies in the considered scenario. We thus use the beamforming vectors

$$\mathbf{b}_k = \tilde{\mathbf{b}}_k \|\tilde{\mathbf{b}}_k\|^{-1}, \quad \tilde{\mathbf{b}}_k = \zeta_k \mathbf{b}_k^{\text{MRT}} + (1 - \zeta_k) \mathbf{b}_k^{\text{ZF}}, \quad (64a)$$

$$\mathbf{b}_k^{\text{MRT}} = \mathbf{h}_{kk} \|\mathbf{h}_{kk}\|^{-1}, \quad \mathbf{b}_k^{\text{ZF}} = \mathbf{\Pi}_{\mathbf{h}_{jk}}^{\perp} \mathbf{h}_{kk} \|\mathbf{\Pi}_{\mathbf{h}_{jk}}^{\perp} \mathbf{h}_{kk}\|^{-1} \quad (64b)$$

where  $\mathbf{\Pi}_{\mathbf{h}_{jk}}^{\perp} = \mathbf{I}_{M_k} - \frac{\mathbf{h}_{jk} \mathbf{h}_{jk}^H}{\mathbf{h}_{jk}^H \mathbf{h}_{jk}}$  is the orthogonal projection onto the orthogonal complement of the span of  $\mathbf{h}_{jk}$ , and  $\zeta_k$ ,  $k = 1, 2$  are auxiliary variables that need to be optimized. The achievable rates with Gaussian codebooks, interference treated

as noise, and without self-interference can then be expressed as [32]

$$r_k = \log_2 \left( 1 + \frac{p_k |\mathbf{h}_{kk}^H \mathbf{b}_k|^2}{\sigma_k^2 + p_j |\mathbf{h}_{kj}^H \mathbf{b}_j|^2} \right) = \log_2 \left( 1 + \frac{p_k \alpha_k(\boldsymbol{\zeta})}{\sigma^2 + p_j \beta_j(\boldsymbol{\zeta})} \right) \quad (65)$$

where

$$\alpha_k(\boldsymbol{\zeta}) = |\mathbf{h}_{kk}^H \mathbf{b}_k|^2 = \frac{(\zeta_k \gamma_{kk} + (1 - \zeta_k) \gamma_{kj})^2}{1 - 2\zeta_k(1 - \zeta_k)(1 - \frac{\gamma_{kj}}{\gamma_{kk}})} \geq 0, \quad (66a)$$

$$\beta_k(\boldsymbol{\zeta}) = |\mathbf{h}_{kj}^H \mathbf{b}_j|^2 = \frac{\zeta_k^2 \delta_{kj}^2 \gamma_{kk}^{-2}}{1 - 2\zeta_k(1 - \zeta_k)(1 - \frac{\gamma_{kj}}{\gamma_{kk}})} \geq 0 \quad (66b)$$

with  $\gamma_{kk} = \|\mathbf{h}_{kk}\|$ ,  $\gamma_{kj} = \|\mathbf{\Pi}_{\mathbf{h}_{jk}}^\perp \mathbf{h}_{kk}\|$ , and  $\delta_{kj} = |\mathbf{h}_{kk}^H \mathbf{h}_{kj}|$  [2], [32]. Since  $\alpha_k(\boldsymbol{\zeta})$  and  $\beta_k(\boldsymbol{\zeta})$  are nondecreasing in both components of  $\boldsymbol{\zeta}$  (see [32] for a proof), we can use (10), (12) and (13) to establish the MMP rate expression

$$R_k \left( \begin{bmatrix} \zeta \\ \mathbf{p} \end{bmatrix}, \begin{bmatrix} \xi \\ \mathbf{q} \end{bmatrix} \right) = \log_2 \left( 1 + \frac{p_k \alpha_k(\boldsymbol{\zeta})}{\sigma^2 + q_j \beta_j(\boldsymbol{\xi})} \right). \quad (67)$$

We can thus optimize the global energy efficiency in the two-user MISO IC by replacing  $R_k$  in (42) by (67) with  $\mathbf{x} = [\boldsymbol{\zeta}^T, \mathbf{p}^T]^T$  and  $\mathbf{y} = [\boldsymbol{\xi}^T, \mathbf{q}^T]^T$ . This means that we apply Algorithm 1 in a four-dimensional space.

In a similar manner, all other optimization problems for the single-antenna interference channel that could previously be formulated by means of the MMP rate expression  $R_k$  from (35) can be easily extended to the two-user MISO interference channel by using the MMP rate expression (67) instead. An example for this is the rate balancing problem (57) in the MISO IC which we considered in [2]. For the special case of weighted sum rate maximization without minimum rate constraints, the problem can be simplified since, in this case, it is optimal for both users to exploit their full power budget [23, Proposition 1]. Hence, Algorithm 1 has to be applied only for the two auxiliary variables  $\boldsymbol{\zeta}$ .

The MMP framework can also be applied to nonconvex optimization problems in other multiantenna scenarios, such as the  $K$ -user MISO broadcast channel with linear transceivers. An example is the method in [66, Sec. 7.3.1.2], which is in fact a special case of the MMP framework.

#### F. Probability Optimization for Slotted ALOHA

To demonstrate that the proposed MMP framework can also be useful for solving problems on the medium access control layer, we study the problem from [35, Ch. 7] where the transmission probabilities in the slotted ALOHA protocol with  $K$  users were optimized, i.e.,

$$\max_{\mathbf{0} \leq \boldsymbol{\theta} \leq \mathbf{1}} U(r_1(\boldsymbol{\theta}), \dots, r_K(\boldsymbol{\theta})) \quad (68a)$$

$$\text{s. t. } r_k(\boldsymbol{\theta}) \geq R_{\min,k} \quad \forall k \quad (68b)$$

with an increasing (not necessarily concave) utility function  $U$ , and average per-user throughput

$$r_k(\boldsymbol{\theta}) = c_k \theta_k \prod_{j \in \mathcal{I}(k)} (1 - \theta_j). \quad (69)$$

Here,  $\boldsymbol{\theta} = [\theta_1, \dots, \theta_K]^T$  contains the probabilities  $\theta_k$  that user  $k$  attempts to transmit a packet in any time-slot, and  $\mathcal{I}(k)$

contains the indices of all users that cause interference to receiver  $k$ . The data rates  $r_k$  are given by the product of the data rate  $c_k$  of a successful transmission and the probability of a collision-free transmission.

The first solution approach in [35, Ch. 7] transforms the problem to a canonical monotonic optimization problem

$$\max_{\boldsymbol{\theta} \geq \mathbf{0}, \hat{\boldsymbol{\theta}} \geq \mathbf{0}} U(\hat{r}_1(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}), \dots, \hat{r}_K(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})) \quad (70a)$$

$$\text{s. t. } \hat{r}_k(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) \geq R_{\min,k} \quad \forall k \quad (70b)$$

$$\boldsymbol{\theta} + \hat{\boldsymbol{\theta}} \leq \mathbf{1} \quad (70c)$$

with

$$\hat{r}_k(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = c_k \theta_k \prod_{j \in \mathcal{I}(k)} \hat{\theta}_j \quad (71)$$

and solves it by means of the PA. As an alternative, this problem could also be solved with the BRB algorithm for DM problems from [29, Sec. 7]. However, no matter which algorithm is applied, the formulation in (70) suffers from the doubled dimensionality of the optimization problem, which has drastic consequences [35, Ch. 7] since the worst-case complexities of the PA and BB algorithm grows exponentially in the number of variables [45].

Therefore, a second approach

$$\max_{\mathbf{v} \geq \mathbf{0}} U(c_1 v_1, \dots, c_K v_K) \quad \text{s. t. } \mathbf{v} \in \mathcal{Y} \quad (72)$$

was proposed in [35, Ch. 7], where

$$\mathcal{Y} = \{ \mathbf{v} \mid c_k v_k \geq R_{\min,k}, \forall k \text{ and } \exists (\mathbf{0} \leq \boldsymbol{\theta} \leq \mathbf{1}) : c_k v_k = r_k(\boldsymbol{\theta}), \forall k \}. \quad (73)$$

As a result, the PA algorithm can be implemented with only  $K$  variables, but this comes at the cost that a geometric program (for details see [35, Ch. 7]) has to be solved to perform the projection to  $\mathbf{v} \in \mathcal{Y}$  in each iteration of the PA.

To avoid the drawbacks of both methods, we reformulate (68) in terms of the MMP framework with MM objective and MM constraint functions given as

$$F(\mathbf{x}, \mathbf{y}) = U(R_1(\mathbf{x}, \mathbf{y}), \dots, R_K(\mathbf{x}, \mathbf{y})), \quad (74a)$$

$$G_k(\mathbf{x}, \mathbf{y}) = R_{\min,k} - R_k(\mathbf{y}, \mathbf{x}), \quad (74b)$$

$$\text{with } R_k(\mathbf{x}, \mathbf{y}) = c_k x_k \prod_{j \in \mathcal{I}(k)} (1 - y_j). \quad (74c)$$

Note that these constraints do not fulfill the additional requirements in (23). Thus, Proposition 2 is not applicable and (68) with MMP representations (74) needs to be solved with the modified, infinite version of Algorithm 1 described in Section III-A. However, although the algorithm is infinite in theory, it turns out to have very fast convergence in practice.

It is important to note that the auxiliary variables  $\mathbf{y}$  in the MMP method are used only as a vehicle to compute bounds, without considering them as additional optimization variables. Thus, unlike the canonical monotonic reformulation (70), the MMP method does not increase the dimensionality of the problem. Moreover, the MMP formulation avoids an additional inner solver as needed in the geometric programming based formulation (72).

For the numerical results in Table I, we have used a three-user system with proportional fair utility (48), full

TABLE I  
 MEAN AND MEDIAN RUN TIMES OF VARIOUS SOLUTION METHODS FOR (68).

	3 Users		4 Users	
	Mean	Median	Mean	Median
(70) & PA	> 23 h		—	—
(70) & DM BB (no reduction)	88.250 s	21.305 s	—	—
(70) & DM BRB (reduction)	23.182 s	5.919 s	—	—
(72) & PA	3.629 s	0.961 s	13.4 h	22.935 s
MMP (74) (no reduction)	0.769 s	0.172 s	150.1 s	4.838 s
MMP (74) (reduction)	1.413 s	0.364 s	256.0 s	8.139 s

interference  $\mathcal{I}(k) = \{j \mid j \neq k\}$ , and  $c_k = \log_2(1 + |\alpha'_k|^2)$  with i.i.d.  $\alpha'_k \sim \mathcal{CN}(0, 1)$  for all  $k$ . To create a variety of challenging scenarios in which some of the minimum rate constraints are active, we have generated  $R_{\min,k} = c_k \chi_k$  with  $\chi_k \sim \mathcal{N}(\frac{(K-1)^2}{K^2}, .05^2)$  since  $R_{\min,k} = \frac{(K-1)^2}{K^2} c_k, \forall k$  would be the boundary to infeasibility in case of full interference. All infeasible scenarios among the generated ones have been discarded, and the results are averaged over 100 feasible scenarios. As all algorithms have fundamentally different per-iteration complexities, it does not make sense to count iterations. We thus again fall back to comparing computation times of the C++ implementations [46].

In addition to the significantly lower run time of the MMP method, another remarkable aspect can be observed. The reduction step (28) reduces the run time of the DM approach while it increases the run time of the MMP approach. As stated before, it depends on the problem under consideration whether or not performing a reduction leads to an overall gain in computation time. An example where the reduction step proves to be very helpful in combination with the MMP method is the rate balancing problem with time-sharing in [2].

## V. DISCUSSION

The mixed monotonic programming (MMP) framework that we propose in this article can directly exploit hidden monotonicity of single terms in a function expression even if the function as a whole is neither monotonic nor a difference of monotonic functions. This allows us to derive bounds that are tighter than previously used DM bounds, leading to faster convergence in BB algorithms. Moreover, the MMP framework enables us to derive bounds even for a wide range of problems for which no DM reformulation exists, so that we can avoid previously proposed nested algorithms, e.g., for fractional monotonic problems. Due to these advantages, solutions based on the new MMP framework achieve tremendous reductions of run time and memory consumption compared to state-of-the-art solutions in all numerical examples that we considered. These examples come from the area of signal processing for communications, but we are convinced that the proposed framework can help to speed up global optimization in many other areas of research as well.

An interesting theoretical aspect of the MMP framework is that it can be considered as a generalization of the DM approach and of other special cases previously studied in the literature. From a practical perspective, we have discussed the oldest-first selection rule and a reduction method for MMP problems as additional methods to speed up the implementation

in specific scenarios. In the code repository [46], we provide a C++ implementation of the proposed BRB algorithm for MMP, which can be easily adapted to arbitrary MMP problems, as well as the simulation code for all numerical examples discussed in this paper.

### A. Convergence Speed and the Optimal Choice of $F$

We have established convergence of the BRB algorithm for MMP problems (Algorithm 1) to an  $\eta$ -optimal solution of (P) within a finite number of iterations in Theorem 1. Establishing this kind of convergence is an important theoretical result as pointed out by Donald Knuth [67, Sect. 1.1]. In case of Algorithm 1, it holds for any MMP representation  $F$  of  $f$ , i.e., any continuous function  $F$  that satisfies (1) and (2). Thus, from a theoretical perspective, the non-uniqueness of MMP representations is not an issue for the convergence of Algorithm 1.

However, we have seen in Section IV that the actual convergence speed depends strongly on the precise choice of  $F$ . It would be beneficial to have means of obtaining the optimal  $F$  for a given objective function  $f$ . This raises the question of how to define *optimality* in terms of a bounding function. From a theoretical perspective, the tightest bound leads to fastest convergence. Practically, however, this bound can be costly to compute and might lead to longer run times than less tight bounds. In some problem instances, a tighter bound might even lead to slower convergence because the branching is performed in a different order and better feasible points are encountered earlier.

Even when leaving all these considerations aside and simply assuming that the optimal  $F$  is the MMP representation of  $f$  that leads to the tightest bound, it is still challenging to obtain general quantitative statements that are not limited to a particular problem instance. Indeed, obtaining tight (not even the tightest) bounds is not only relevant for MMP, or monotonic optimization in general, but also in other fields of nonconvex optimization, both for global optimal solutions and heuristic algorithms [68]. For example, every successive convex approximation algorithm benefits from tight bounds, but in many cases it is already a noticeable achievement to obtain *some* suitable bound. This is where one of our main contributions lies: We provide the theoretical and algorithmic framework for a novel bounding methodology that enables the derivation of powerful bounds for functions that were previously intractable (cf. Section IV-B).

Another aspect is comparing the convergence behavior of different algorithms (e.g., PA and MMP) and bounding schemes (e.g., DM and MMP) analytically. The common methodology is to examine the rate of convergence defined as the number  $p \in \mathbb{N}$  such that

$$U(\mathcal{M}_k) - \gamma_k \leq C \text{diam}(\mathcal{M}_k)^p \quad (75)$$

for some fixed constant  $C > 0$ . This is an active area of research in the operations research community and leads to worst-case bounds on the number of required iterations. However, supported by the results in [69], we would expect a rate of convergence  $p \leq 2$  for DM and MMP bounds. Moreover,

we expect this rate to be equal for both bounding schemes since DM bounds are a special case of MMP bounds. Thus, supporting the experimental results in Section IV by theoretical results would require an approach that goes beyond the usual rate of convergence analysis and determines the constant  $C$  in (75) analytically. Again, the challenge would be to obtain quantitative statements that are not specific to a particular problem instance. Moreover, since such results will only be worst-case bounds without direct implications for the average run time, it is not clear whether this approach is a viable method to quantify the experimentally verified performance gain of MMP. For all these reasons, such an analysis goes beyond the scope of this paper and is left open for future research.

Apart from these theoretical questions, we have demonstrated that the MMP framework helps to find much faster globally optimal solution methods for many relevant problems. This can be observed numerically from the broad selection of examples in Section IV, it can be justified analytically in some cases such as in Section IV-A, and it is obvious from the fact that the MMP framework helps to avoid nested optimization in other cases. Especially in cases where a DM representation is not available but an MMP representation can be found, MMP leads to tremendous speed-ups over the state-of-the-art, even though we might not have a theoretical guarantee to have found the optimal MMP representation. Moreover, in some of these cases it even enables the solution of previously intractable problems.

#### REFERENCES

- [1] 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.
- [2] C. Hellings, B. Matthiesen, E. A. Jorswieck, and W. Utschick, "Globally optimal TIN strategies with time-sharing in the MISO interference channel," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, A Coruña, Spain, Sep. 2019.
- [3] I. E. Telatar, "Capacity of multi-antenna Gaussian channels," *Eur. Trans. Telecommun.*, vol. 10, no. 6, pp. 585–595, Nov./Dec. 1999.
- [4] D. Palomar, J. Cioffi, and M. Laganas, "Joint Tx-Rx beamforming design for multicarrier MIMO channels: a unified framework for convex optimization," *IEEE Trans. Signal Process.*, vol. 51, no. 9, pp. 2381–2401, Sep. 2003.
- [5] C. Isheden, Z. Chong, E. Jorswieck, and G. Fettweis, "Framework for link-level energy efficiency optimization with informed transmitter," *IEEE Trans. Wireless Commun.*, vol. 11, no. 8, pp. 2946–2957, Aug. 2012.
- [6] 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.
- [7] 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.
- [8] S. Ye and R. S. Blum, "Optimized signaling for MIMO interference systems with feedback," *IEEE Trans. Signal Process.*, vol. 51, no. 11, pp. 2839–2848, Nov. 2003.
- [9] R. Böhnke and K.-D. Kammeyer, "Weighted sum rate maximization for MIMO-OFDM systems with linear and dirty paper precoding," in *Proc. Int. ITG Conf. Source, Channel Coding (SCC)*, Ulm, Germany, Jan. 2008.
- [10] R. Hunger, D. Schmidt, and M. Joham, "A combinatorial approach to maximizing the sum rate in the MIMO BC with linear precoding," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Oct. 2008, pp. 316–320.
- [11] C. Hellings, W. Utschick, and M. Joham, "Power minimization in parallel vector broadcast channels with separate linear precoding," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Barcelona, Spain, Sep. 2011, pp. 1834–1838.
- [12] T. Yoo and A. Goldsmith, "On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 3, pp. 528–541, Mar. 2006.
- [13] C. Guthy, W. Utschick, R. Hunger, and M. Joham, "Efficient weighted sum rate maximization with linear precoding," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2284–2297, Apr. 2010.
- [14] 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.
- [15] M. Hong, Q. Li, and Y.-F. Liu, "Decomposition by successive convex approximation: A unifying approach for linear transceiver design in heterogeneous networks," vol. 15, no. 2, pp. 1377–1392, Feb. 2016.
- [16] Y. Yang and M. Pesavento, "A unified successive pseudoconvex approximation framework," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3313–3328, Jul. 2017.
- [17] M. Codreanu, A. Tölili, 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.
- [18] S. Shi, M. Schubert, and H. Boche, "Rate optimization for multiuser MIMO systems with linear processing," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 4020–4030, Aug. 2008.
- [19] S. S. Christensen, R. Agarwal, E. D. Carvalho, and J. M. Cioffi, "Weighted sum-rate maximization using weighted MMSE for MIMO-BC beamforming design," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 4792–4799, Dec. 2008.
- [20] Y.-F. Liu, Y.-H. Dai, and Z.-Q. Luo, "Max-min fairness linear transceiver design for a multi-user MIMO interference channel," *IEEE Trans. Signal Process.*, vol. 61, no. 9, pp. 2413–2423, May 2013.
- [21] S. Lagen, A. Agustin, and J. Vidal, "Coexisting linear and widely linear transceivers in the MIMO interference channel," *IEEE Trans. Signal Process.*, vol. 64, no. 3, pp. 652–664, Feb. 2016.
- [22] D. A. Schmidt, C. Shi, R. A. Berry, M. L. Honig, and W. Utschick, "Pricing algorithms for power control and beamformer design in interference networks," *IEEE Signal Process. Mag.*, vol. 26, no. 5, pp. 53–63, Sep. 2009.
- [23] E. G. Larsson and E. A. Jorswieck, "Competition versus cooperation on the MISO interference channel," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 7, pp. 1059–1069, Sep. 2008.
- [24] G. Scutari, D. P. Palomar, and S. Barbarossa, "Optimal linear precoding strategies for wideband noncooperative systems based on game theory—part II: Algorithms," *IEEE Trans. Signal Process.*, vol. 56, no. 3, pp. 1250–1267, Mar. 2008.
- [25] 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.
- [26] F. Baccelli, N. Bambos, and N. Gast, "Distributed delay-power control algorithms for bandwidth sharing in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 19, no. 5, pp. 1458–1471, Oct. 2011.
- [27] R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*, 3rd ed. New York; Berlin, Germany; Vienna, Austria: Springer-Verlag, 1996.
- [28] H. Tuy, "Monotonic optimization: Problems and solution approaches," *SIAM J. Optim.*, vol. 11, no. 2, pp. 464–494, Feb. 2000.
- [29] 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.
- [30] H. Tuy, *Convex Analysis and Global Optimization*, 2nd ed., ser. Springer Optim. Appl. New York; Berlin, Germany; Vienna, Austria: Springer-Verlag, 2016, vol. 110.
- [31] E. A. Jorswieck and E. G. Larsson, "Linear precoding in multiple antenna broadcast channels: Efficient computation of the achievable rate region," in *Proc. Int. ITG Workshop Smart Antennas (WSA)*, Darmstadt, Germany, Feb. 2008, pp. 21–28.
- [32] —, "Monotonic optimization framework for the two-user MISO interference channel," *IEEE Trans. Commun.*, vol. 58, no. 7, pp. 2159–2168, Jul. 2010.
- [33] 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, Mar. 2009.
- [34] L. P. Qian and Y. J. Zhang, "S-MAPEL: Monotonic optimization for non-convex joint power control and scheduling problems," *IEEE Trans. Wireless Commun.*, vol. 9, no. 5, pp. 1708–1719, May 2010.

- [35] Y. J. Zhang, L. P. Qian, and J. Huang, *Monotonic Optimization in Communication and Networking Systems*, ser. FnT Netw. Boston, MA, USA: Now, 2012, vol. 7, no. 1.
- [36] A. Zappone and E. A. Jorswieck, *Energy Efficiency in Wireless Networks via Fractional Programming Theory*, ser. FnT Commun. Inf. Theory. Boston, MA, USA: Now, 2015, vol. 11, no. 3–4.
- [37] 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.
- [38] 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.
- [39] J. Brehmer, *Utility Maximization in Nonconvex Wireless Systems*, ser. Found. Signal Process., Commun., Netw., W. Utschick, H. Boche, and R. Mathar, Eds. New York; Berlin, Germany; Vienna, Austria: Springer-Verlag, 2012, vol. 5.
- [40] J. Brehmer and W. Utschick, “Utility maximization in the multi-user MISO downlink with linear precoding,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Dresden, Germany, Jun. 2009.
- [41] C. Hellings, M. Joham, M. Riemensberger, and W. Utschick, “Minimal transmit power in parallel vector broadcast channels with linear precoding,” *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1890–1898, Apr. 2012.
- [42] 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.
- [43] E. Björnson, G. Zheng, M. Bengtsson, and B. Ottersten, “Robust monotonic optimization framework for multicell MISO systems,” *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2508–2523, Jan. 2012.
- [44] A. Gründinger, M. Joham, and W. Utschick, “Feasibility test and globally optimal beamformer design in the satellite downlink based on instantaneous and ergodic rates,” in *Proc. Int. ITG Workshop Smart Antennas (WSA)*, Dresden, Germany, Mar. 2012, pp. 217–224.
- [45] E. Björnson and E. A. Jorswieck, *Optimal Resource Allocation in Coordinated Multi-Cell Systems*, ser. FnT Commun. Inf. Theory. Boston, MA, USA: Now, 2013, vol. 9, no. 2–3.
- [46] B. Matthiesen and C. Hellings. (2019) Accompanying source code. [Online]. Available: <https://github.com/bmatthiesen/mixed-monotonic>
- [47] C. Hellings and W. Utschick, “Proper time-sharing as a baseline for studying improper signaling in interference channels,” in *Proc. Int. ITG Workshop Smart Antennas (WSA)*, Bochum, Germany, Mar. 2018.
- [48] —, “Improper signaling versus time-sharing in the two-user Gaussian interference channel with TIN,” 2018, in review *IEEE Trans. Inf. Theory*. [Online]. Available: <https://arxiv.org/abs/1808.01611>
- [49] W. Dinkelbach, “On nonlinear fractional programming,” *Manage. Sci.*, vol. 13, no. 7, pp. 492–498, Mar. 1967.
- [50] B. Matthiesen, “Efficient globally optimal resource allocation in wireless interference networks,” Ph.D. Thesis, Technische Universität Dresden, Dresden, Germany, Nov. 2019. [Online]. Available: <https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa2-362878>
- [51] H. Tuy, *Convex Analysis and Global Optimization*. Kluwer, 1998.
- [52] 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.
- [53] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [54] B. Matthiesen and E. A. Jorswieck, “Efficient global optimal resource allocation in non-orthogonal interference networks,” *IEEE Trans. Signal Process.*, 2019.
- [55] H. Tuy, “ $\mathcal{D}(C)$ -optimization and robust global optimization,” *J. Global Optim.*, vol. 47, no. 3, pp. 485–501, Oct. 2009.
- [56] 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.
- [57] B. Matthiesen and E. A. Jorswieck, “Global sum rate optimal resource allocation for non-regenerative 3-way relay channels,” in *Workshop Broadband Wireless Commun. Comput. Boards (Atto-Nets), IEEE Int. Conf. Ubiquitous Wireless Broadband (ICUWB)*, Salamanca, Spain, Sep. 2017.
- [58] 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.
- [59] D. E. Knuth, *The Art of Computer Programming: Sorting and Searching*, 2nd ed. Reading, MA, USA: Addison-Wesley, 1997, vol. 3.
- [60] S. Schaible, “Fractional programming,” in *Handbook of Global Optimization*, ser. Nonconvex Optim. Appl., R. Horst and P. M. Pardalos, Eds. Norwell, MA, USA: Kluwer, 1995, vol. 2, pp. 495–608.
- [61] O. Hanner and H. Rådström, “A generalization of a theorem of Fenchel,” *Proc. Amer. Math. Soc.*, vol. 2, no. 4, pp. 589–593, 1951.
- [62] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, 3rd ed. Hoboken, NJ, USA: Wiley, 2006.
- [63] J. E. Kelley, Jr., “The cutting-plane method for solving convex programs,” *J. Soc. Indust. Appl. Math.*, vol. 8, no. 4, pp. 703–712, 1960.
- [64] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2011.
- [65] M. Mohseni, R. Zhang, and J. Cioffi, “Optimized transmission for fading multiple-access and broadcast channels with multiple antennas,” *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1627–1639, Aug. 2006.
- [66] C. Hellings, “Reduced-entropy signals in MIMO communication systems,” Ph.D. Thesis, Professur für Methoden der Signalverarbeitung, Technische Universität München, 2017. [Online]. Available: <http://mediatum.ub.tum.de/?id=1362885>
- [67] D. E. Knuth, *The Art of Computer Programming: Fundamental Algorithms*, 3rd ed. Reading, MA, USA: Addison-Wesley, 1997, vol. 1.
- [68] Y. Sun, P. Babu, and D. P. Palomar, “Majorization-minimization algorithms in signal processing, communications, and machine learning,” *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 794–816, Feb. 2017.
- [69] A. Schöbel and D. Scholz, “The theoretical and empirical rate of convergence for geometric branch-and-bound methods,” *J. Global Optim.*, vol. 48, no. 3, pp. 473–495, Dec. 2009.