Solving complex-valued ℓ_0 minimization problems with constant modulus constraints

Frederic Matter

Joint work with
Tobias Fischer (Fraunhofer ITWM Kaiserslautern),
Ganapati Hegde (TU Darmstadt),
Marius Pesavento (TU Darmstadt),
Marc Pfetsch (TU Darmstadt) and
Andreas Tillmann (RWTH Aachen)

Overview

Motivation and Introduction

Solving CMPs

A heuristic

Application

Implementation

Conclusion and Outline

Introduction

TECHNISCHE UNIVERSITÄT DARMSTADT

We want to solve the following problem

Constant Modulus Program (CMP)

$$\min_{\mathbf{x}\in\mathbb{C}^N} \|\mathbf{x}\|_0$$
$$\text{s. t. } \|\mathbf{b}-\mathbf{A}\mathbf{x}\|_2 \leq \sqrt{\delta},$$
$$|x_n| \in \{0, c\}, \quad \forall n \in [N],$$

with $\mathbf{A} \in \mathbb{C}^{K\times N}$, $\mathbf{b} \in \mathbb{C}^K$, $\delta \in \mathbb{R}$, $c \in \mathbb{R}$, $[N] = \{1, \dots, n\}$ and $\|\mathbf{x}\|_0$ denoting the number of nonzero entries in $\mathbf{x}$.

- In the problem we could also allow additional constraints, such as linear (in-)equalities.
- Throughout this talk, we assume w.l.o.g. that $c = 1$.

Discrete Optimization

Motivation

- Generalization of compressed sensing: search sparse solutions of an optimization problem $\Rightarrow$ use ℓ_0-term in objective function.
- Note that the constant modulus constraint is nonconvex.
- We want to solve this problem exactly.
- constant modulus often appears in signal processing (e.g., phase retrieval).

MINLP–Reformulation

TECHNISCHE UNIVERSITÄT DARMSTADT

- auxiliary binary variables $\mathbf{y} = [y_1, y_2, \dots, y_N]^\mathsf{T} \in \{0,1\}^N$.
- $w_n = \mathrm{Re}[x_n]$ and $z_n = \mathrm{Im}[x_n]$ as real and imaginary part of x_n, respectively.
- $\mathbf{w} = [w_1, w_2, \dots, w_N]^\mathsf{T}$ and $\mathbf{z} = [z_1, z_2, \dots, z_N]^\mathsf{T}$.
- $\mathbf{A}^T = [\mathbf{a}_1, \dots, \mathbf{a}_K] \in \mathbb{C}^{N \times K}$.

MINLP–Reformulation

- auxiliary binary variables $\mathbf{y} = [y_1, y_2, \dots, y_N]^\mathrm{T} \in \{0,1\}^N$.
- $w_n = \mathrm{Re}[x_n]$ and $z_n = \mathrm{Im}[x_n]$ as real and imaginary part of x_n, respectively.
- $\mathbf{w} = [w_1, w_2, \dots, w_N]^\mathrm{T}$ and $\mathbf{z} = [z_1, z_2, \dots, z_N]^\mathrm{T}$.
- $\mathbf{A}^T = [\mathbf{a}_1, \dots, \mathbf{a}_K] \in \mathbb{C}^{N \times K}$.

$$
\begin{aligned}
\min_{\mathbf{w},\mathbf{z}\in\mathbb{R}^N,\mathbf{y}} \sum_{n=1}^{N} y_n \quad \text{s.t.} \quad & \sum_{k=1}^{K} \Big(\mathrm{Re}[b_k] - \big(\mathrm{Re}[\mathbf{a}_k]^T\mathbf{w} - \mathrm{Im}[\mathbf{a}_k]^T\mathbf{z}\big) \Big)^2 \\
& + \Big(\mathrm{Im}[b_k] - \big(\mathrm{Re}[\mathbf{a}_k]^T\mathbf{z} + \mathrm{Im}[\mathbf{a}_k]^T\mathbf{w}\big) \Big)^2 \leq \delta, \\
& w_n^2 + z_n^2 \leq y_n, \quad \forall\, n \in [N], \\
& w_n^2 + z_n^2 \geq y_n, \quad \forall\, n \in [N], \\
& y_n \in \{0,1\}, \quad \forall\, n \in [N].
\end{aligned}
$$

Overview

Motivation and Introduction

Solving CMPs

A heuristic

Application

Implementation

Conclusion and Outline

Solving the resulting MINLP

TECHNISCHE UNIVERSITÄT DARMSTADT

- Apply spatial branch-and-cut framework.
- Error bound (ℓ_2-norm) constraint is convex, even in the complex case, as it can be recast as a second-order cone constraint.
  - second-order cone: $C_k = \left\{ [x_1, \dots, x_k, x_{k+1}]^T \in \mathbb{R}^{k+1} \;:\; \|[x_1, \dots, x_k]^T\|_2 \leq x_{k+1} \right\}$.
  - second-order cone constraint: $\|A_i x + b_i\|_2 \leq c_i^T x + d_i$, $A_i \in \mathbb{R}^{(n_i-1)\times n}, b_i \in \mathbb{R}^{n_i-1}, c_i \in \mathbb{R}^n, d_i \in \mathbb{R}$.
- Upper bound parts of modulus constraints are convex quadratic constraints.
- Lower bound parts of modulus constraints are nonconvex quadratic constraints.

Solving the resulting MINLP

TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Apply spatial branch-and-cut framework.
- Error bound (ℓ_2-norm) constraint is convex, even in the complex case, as it can be recast as a second-order cone constraint.
  - second-order cone: $C_k = \left\{ [x_1, \ldots, x_k, x_{k+1}]^T \in \mathbb{R}^{k+1} \; : \; \|[x_1, \ldots, x_k]^T\|_2 \leq x_{k+1} \right\}$.
  - second-order cone constraint: $\|A_i x + b_i\|_2 \leq c_i^T x + d_i$,
    $A_i \in \mathbb{R}^{(n_i-1)\times n}$, $b_i \in \mathbb{R}^{n_i-1}$, $c_i \in \mathbb{R}^n$, $d_i \in \mathbb{R}$.
- Upper bound parts of modulus constraints are convex quadratic constraints.
- Lower bound parts of modulus constraints are nonconvex quadratic constraints.

$\Rightarrow$ exploit structure of modulus constraints to develop specialized branching and propagation (and some very easy separation) to enforce modulus constraints.

The LP relaxation

Another problem: poor linear relaxation.

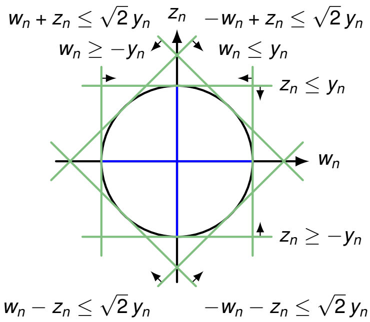
$\Rightarrow$ Use additional linear inequalities that strengthen the LP relaxation.

The LP relaxation

TECHNISCHE UNIVERSITÄT DARMSTADT

Another problem: poor linear relaxation.

$\Rightarrow$ Use additional linear inequalities that strengthen the LP relaxation.

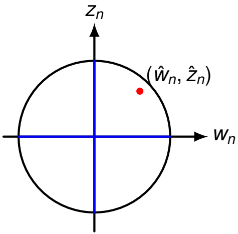


$$
\begin{aligned}
-y_n \leq w_n &\leq y_n, \\
-y_n \leq z_n &\leq y_n, \\
w_n + z_n &\leq \sqrt{2}\, y_n, \\
w_n - z_n &\leq \sqrt{2}\, y_n, \\
-w_n + z_n &\leq \sqrt{2}\, y_n, \\
-w_n - z_n &\leq \sqrt{2}\, y_n.
\end{aligned}
$$

Enforcing Modulus Constraints

TECHNISCHE UNIVERSITÄT DARMSTADT

If the solution $(\hat{\mathbf{w}}, \hat{\mathbf{z}}, \hat{\mathbf{y}})$ of the LP relaxation violates $w_n^2 + z_n^2 \geq y_n$ for some $n \in [N]$:

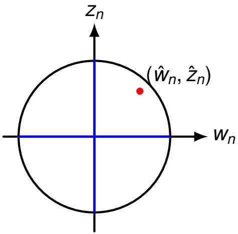


Discrete Optimization

Enforcing Modulus Constraints

If the solution $(\hat{\mathbf{w}}, \hat{\mathbf{z}}, \hat{\mathbf{y}})$ of the LP relaxation violates $w_n^2 + z_n^2 \geq y_n$ for some $n \in [N]$:

- If binary variable $\hat{y}_n$ fixed to zero: set $\hat{w}_n, \hat{z}_n$ to zero as well.

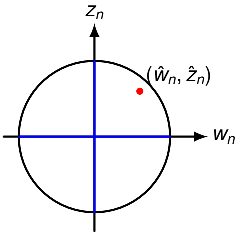


Enforcing Modulus Constraints

TECHNISCHE UNIVERSITÄT DARMSTADT

If the solution $(\hat{\mathbf{w}}, \hat{\mathbf{z}}, \hat{\mathbf{y}})$ of the LP relaxation violates $w_n^2 + z_n^2 \geq y_n$ for some $n \in [N]$:

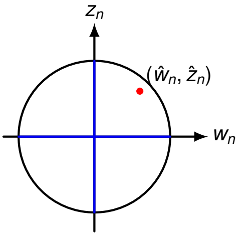
- If binary variable $\hat{y}_n$ fixed to zero: set $\hat{w}_n, \hat{z}_n$ to zero as well.
- If bounds of the continuous variables w_n and z_n are not yet restricted to one of the four orthants: create four branching nodes, one for each orthant.



Enforcing Modulus Constraints

If the solution $(\hat{\mathbf{w}}, \hat{\mathbf{z}}, \hat{\mathbf{y}})$ of the LP relaxation violates $w_n^2 + z_n^2 \geq y_n$ for some $n \in [N]$:

- If binary variable $\hat{y}_n$ fixed to zero: set $\hat{w}_n, \hat{z}_n$ to zero as well.
- If bounds of the continuous variables w_n and z_n are not yet restricted to one of the four orthants: create four branching nodes, one for each orthant.
- Otherwise:
  1. Propagation,
  2. Separation,
  3. Branching.

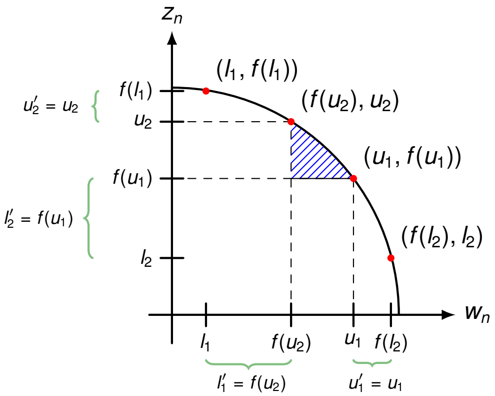


Propagation

If y_n is fixed to one:
l_1, u_1, l_2, u_2 current lower and upper bounds of w_n and z_n.

$$l_1' = \max\{l_1, f(u_2)\},$$
$$u_1' = \min\{u_1, f(l_2)\},$$
$$l_2' = \max\{l_2, f(u_1)\},$$
$$u_2' = \min\{u_2, f(l_1)\},$$

where $f(x) = \sqrt{1 - x^2}$.
$w_n^2 + z_n^2 \geq y_n$ needs to be fulfilled.



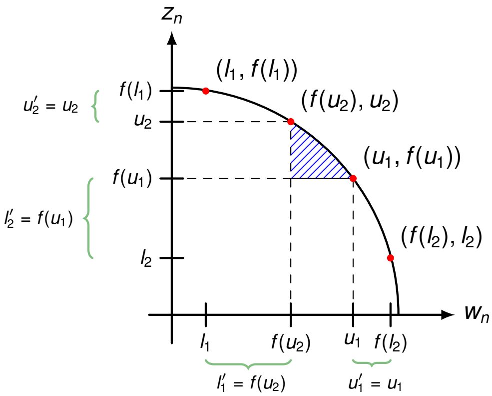
Propagation

TECHNISCHE UNIVERSITÄT DARMSTADT

If y_n is fixed to one:
l_1, u_1, l_2, u_2 current lower and upper bounds of w_n and z_n.

$$l_1' = \max\{l_1, f(u_2)\},$$
$$u_1' = \min\{u_1, f(l_2)\},$$
$$l_2' = \max\{l_2, f(u_1)\},$$
$$u_2' = \min\{u_2, f(l_1)\},$$

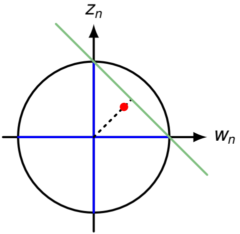
where $f(x) = \sqrt{1 - x^2}$.
$w_n^2 + z_n^2 \geq y_n$ needs to be fulfilled.



⇒ Use l_1', u_1', l_2' and u_2' as new lower and upper bounds of w_n and z_n.

Separation and Branching

If $\hat{w}_n + \hat{z}_n < \hat{y}_n$, add the cut $w_n + z_n \geq y_n$ to the LP relaxation.

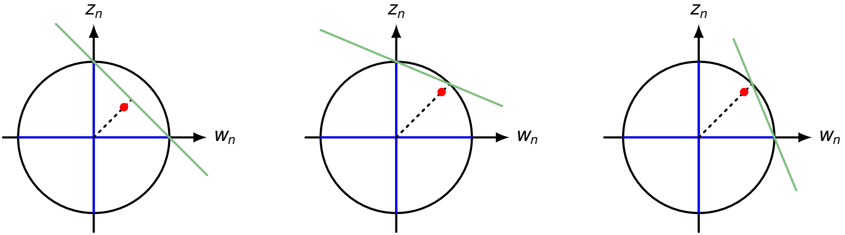


Separation and Branching

If $\hat{w}_n + \hat{z}_n < \hat{y}_n$, add the cut $w_n + z_n \geq y_n$ to the LP relaxation.
If $\hat{w}_n + \hat{z}_n \geq \hat{y}_n$, create two branching nodes with the inequalities

$$-\frac{\hat{z}_n{}' - 1}{\hat{w}_n{}'} \cdot w_n + z_n = 1, \qquad \frac{\hat{z}_n{}'}{\hat{w}_n{}' - 1} \cdot w_n - z_n = \frac{\hat{z}_n{}'}{\hat{w}_n{}' - 1},$$

where $(\hat{w}_n, \hat{z}_n)$ is the current LP solution, and $\hat{w}_n{}' = \frac{\hat{w}_n}{\sqrt{\hat{w}_n{}^2 + \hat{z}_n{}^2}}$, $\hat{z}_n{}' = \frac{\hat{z}_n}{\sqrt{\hat{w}_n{}^2 + \hat{z}_n{}^2}}$.



Specialized Approach – Some remarks

TECHNISCHE UNIVERSITÄT DARMSTADT

- Enforcement of binary variables, second-order cone (SOC) constraint and the (convex) upper bound part of the quadratic modulus constraints is prioritized over the (nonconvex) lower bound part of the modulus constraints.
- A modulus constraint for enforcing is selected by a "most infeasible" rule: Enforce a modulus constraint $w_{\bar{n}}^2 + z_{\bar{n}}^2 \geq y_{\bar{n}}$, $\bar{n} \in [N]$, with largest violation

$$\rho(n) = \hat{y}_n - (\hat{w}_n^2 + \hat{z}_n^2).$$

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

Motivation and Introduction

Solving CMPs

A heuristic

Application

Implementation

Conclusion and Outline

Discrete Optimization

A heuristic

Recall the problem

$$\min \; \|\mathbf{x}\|_0 \;\text{ s. t. }\; \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2 \leq \sqrt{\delta}, \quad |x_n| \in \{0, 1\}, \; \forall n \in [N].$$

Basic idea: fix sparsity level M, try iteratively to create an M-sparse solution by guessing and updating. If not successful, increase M by one and restart.

A heuristic

Recall the problem

$$\min \; \|\mathbf{x}\|_0 \;\text{ s.t. }\; \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2 \leq \sqrt{\delta}, \quad |x_n| \in \{0, 1\}, \; \forall n \in [N].$$

Basic idea: fix sparsity level M, try iteratively to create an M-sparse solution by guessing and updating. If not successful, increase M by one and restart.

- guess solution: draw at random M indices and randomly choose their values on the unit circle, set the remaining values to zero.
- update solution: For two randomly chosen indices, one with non-zero value, one with zero value, recompute their values with a least squares problem where all but these two indices are fixed. Keep the smaller of the two new values, possibly swap the zero and non-zero index.

A heuristic

TECHNISCHE UNIVERSITÄT DARMSTADT

Recall the problem

$$\min \|\mathbf{x}\|_0 \text{ s. t. } \|\mathbf{b} - \mathbf{Ax}\|_2 \leq \sqrt{\delta},$$
$$|x_n| \in \{0, 1\}, \forall n \in [N].$$

- guess 1000 starting solutions, update each 1000 times.
- If desired error bound $\sqrt{\delta}$ is reached: break.
- Use solution as initial primal bound in the branch-and-cut tree.

Overview

Motivation and Introduction

Solving CMPs

A heuristic

Application

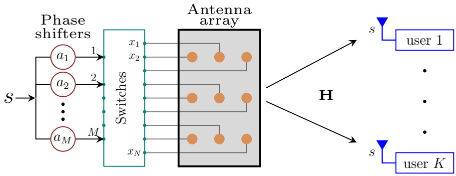
Implementation

Conclusion and Outline

Application example: Joint Antenna Selection and Phase-only Beamforming

TECHNISCHE UNIVERSITÄT DARMSTADT

- Sensor array with N antenna elements, M phase shifters, $M \ll N$.
- $\mathbf{x} \in \mathbb{C}^N$ is the transmit signal vector, $x_n = |x_n| e^{i\psi_n}$.
- channel matrix $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_K] \in \mathbb{C}^{N \times K}$.
- Analog beamformers $\mathbf{a} = [a_1, a_2, \dots, a_M]^T \in \mathbb{C}^M$, assume $|a_m| = c$ for $m \in [M]$.
- if nth antenna element connected to mth phase shifter: $x_n = a_m$, else: $x_n = 0$.

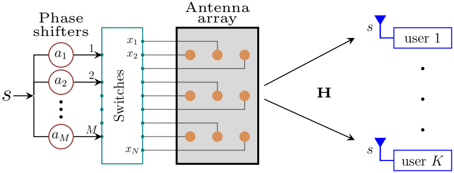


Discrete Optimization

Application example: Joint Antenna Selection and Phase-only Beamforming

- Sensor array with N antenna elements, M phase shifters, $M \ll N$.
- $\mathbf{x} \in \mathbb{C}^N$ is the transmit signal vector, $x_n = |x_n| e^{i\psi_n}$.
- channel matrix $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_K] \in \mathbb{C}^{N\times K}$.
- Analog beamformers $\mathbf{a} = [a_1, a_2, \dots, a_M]^T \in \mathbb{C}^M$, assume $|a_m| = c$ for $m \in [M]$.
- if nth antenna element connected to mth phase shifter: $x_n = a_m$, else: $x_n = 0$.

- K single antenna users that need to be served.
- desired output at users: $\mathbf{s} = [s_1, s_2, \dots, s_K]^\mathrm{T}$.
- actual output at users: $\hat{\mathbf{s}} = \mathbf{H}^\mathrm{T}\mathbf{x} + \mathbf{n}$.

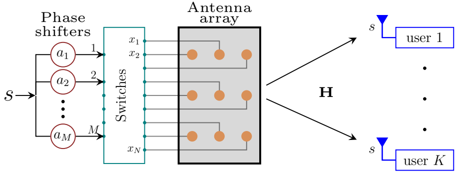


Application example: Joint Antenna Selection and Phase-only Beamforming

TECHNISCHE UNIVERSITÄT DARMSTADT

Our goal

- Design the optimal phase values of the ABF **a**.
- Assign appropriate antenna elements to the phase shifters to serve users in the network.
- Minimize the number of active antennas.
- Keep the root-mean-square error between the desired and actual output at the user below $\sqrt{\delta}$.



Mathematical model

$$\min_{\mathbf{x}\in\mathbb{C}^N} \|\mathbf{x}\|_0$$

$$\text{s. t. } \left\|\mathbf{s} - \mathbf{H}^T\mathbf{x}\right\|_2 \leq \sqrt{\delta},$$

$$|x_n| \in \{0, c\}, \quad \forall n \in [N].$$

Overview

Motivation and Introduction

Solving CMPs

A heuristic

Application

Implementation

Conclusion and Outline

Implementation details

TECHNISCHE UNIVERSITÄT DARMSTADT

We have implemented the following parts:

- A reader that sets up the problem. It builds the error bound constraint as
  - either general quadratic constraint,
  - or as SOC constraint: auxiliary variables that represent $\mathbf{Ax} - \mathbf{b}$ are needed.
  - SOC-representation in SCIP: $\sqrt{\gamma + \sum_{i=1}^{n} (\alpha_i (x_i + \beta_i))^2} \leq \alpha_{n+1} (x_{n+1} + \beta_{n+1})$
- A probdata structure to save **A**, **b** and δ.
- A constraint handler for the branching, propagation and the separation methods for constant modulus constraints.
- The constraint handler has a propagation and enforcement method, the branching and separation is done in the enforcement method.
- A heuristic plugin.

Some computational results

TECHNISCHE UNIVERSITÄT DARMSTADT

First testset: 48 instances, $N \in \{16, 32, 48, 64\}$, $K \in \{2, 3, 4\}$, $\delta^2 \in \{0.1, 0.2\}$
Second testset: 81 instances, $N \in \{32, 48, 64\}$, $K \in \{4, 5, 6\}$, $\delta^2 \in \{0.1, 0.2, 0.3\}$

First testset:

	SCIP 4.0.1			SCIP 5.0.1		
Setting	#opt	#nodes	time (s)	#opt	#nodes	time (s)
default	48	3309	36.4	46	2787	43.1
default + SOC	48	2575	26.0	48	2478	29.1
default + heur	48	396	21.5	47	448	21.9
default + SOC + heur	48	323	21.1	48	364	18.8
mod handling	48	3283	28.3	47	3398	35.5
mod handling + SOC	48	2762	18.4	48	2313	19.7
mod handling + heur	48	427	19.3	48	507	18.0
mod handling + SOC + heur	48	317	17.9	48	361	15.2

Second testset:

	SCIP 4.0.1			SCIP 5.0.1		
Setting	#opt	#nodes	time (s)	#opt	#nodes	time (s)
default	47	126710	1058.8	35	38884	1232.2
default + SOC	47	149482	812.0	40	60565	984.2
default + heur	56	29122	444.4	45	15310	590.2
default + SOC + heur	54	33296	455.9	51	18000	472.0
mod handling	43	154338	1070.4	34	85619	1257.9
mod handling + SOC	55	167485	666.0	49	138182	746.2
mod handling + heur	55	35912	433.6	54	28203	461.2
mod handling + SOC + heur	59	40334	360.2	58	36938	306.9

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

Motivation and Introduction

Solving CMPs

A heuristic

Application

Implementation

Conclusion and Outline

Discrete Optimization

Summary and further research

Conclusion:

- Specialized methods for handling modulus constraints result in speed-up.
- A good primal bound has a (very) high impact on the solution process, as significantly less nodes are needed.
- Explicitly building the convex error bound term as SOC greatly fastens the solution process.

Summary and further research

Conclusion:

- Specialized methods for handling modulus constraints result in speed-up.
- A good primal bound has a (very) high impact on the solution process, as significantly less nodes are needed.
- Explicitly building the convex error bound term as SOC greatly fastens the solution process.

Outlook:

- Let the heuristic also run in deeper nodes of the branch-and-cut tree and initialize it with current LP solution.
- Closely connected: vary the two parameters for the number of guesses and updates.
- Apply this framework to phase retrieval: For a measured modulus $|\mathbf{x}|$ of a complex signal $\mathbf{x} = |\mathbf{x}|e^{i\psi}$ find the phase ψ satisfying a set of constraints.

TECHNISCHE UNIVERSITÄT DARMSTADT

Thanks for your attention!

Discrete Optimization

References

[WSA'18] T. Fischer, G. Hegde, F. Matter, M. Pesavento, M. E. Pfetsch, A. M. Tillmann, "Joint Antenna Selection and Phase-only Beamforming using Mixed-Integer Nonlinear Programming", in *22nd International ITG Workshop on Smart Antennas (WSA)*, Bochum, Germany, March 2018. [accepted]