
NP-Completeness of Multicast Beamforming in Wireless Communication

Sagar Shrestha

Department of Electrical Engineering and Computer Science
Oregon State University
shressag@oregonstate.edu

Abstract

In this work, the classical problem of multi-cast beamforming in wireless communication is reconsidered. [1] showed that the multi-cast beamforming problem is NP-hard. In this project, we show that the corresponding decision problem for real channel matrices and beamformers is NP-complete. Finally, we carry out simulations to reveal the computational complexity of solving the NP-complete beamforming problem in various setting using SAT/SMT solvers.

1 Introduction

In wireless communication, beamforming refers to a technique by which a transmitter with multiple antennas can focus wireless signal towards a receiver. Multi-cast beamforming refers to focusing a common wireless signal towards multiple receivers that may be spatially distributed. This technique is crucial for efficient audio and video streaming in the multiuser setting [2]. In this work, we focus on single group multicast beamforming, which is of great interest in the wireless communication [2, 3].

Single group multicast beamforming was shown to be NP-hard in [1]. Specifically, [1] showed that any instance of the partition problem [4], which is known to be NP-complete, can be reduced to a special instance of the single group multicast beamforming problem. This showed that such special instance can be reduced to any NP-complete problem, as there exist a one to one mapping with a NP-complete problem. However, the special instance used in [1] does not cover any cases interesting or realistic in wireless communication. In this work, we show that a broader subset of the beamforming problem is NP-complete, mainly the case where the entries of channel matrix and beamformer are real numbers instead of complex numbers. Although in practice, the channel matrix are almost always complex, real channel matrix can approximate channel conditions with negligible phase shifts. Similarly, real precoder/beamformer vectors would correspond to discarding quadrature component of modulation.

Finally, we analyse the size of Boolean formula that will be required to encode the beamforming problem, and offer synthetic data simulations using SMT solvers to find solutions of the considered NP-complete problem under various settings.

Notation. x , \mathbf{x} , and \mathbf{X} denote a scalar, a vector, and a matrix, respectively. \mathbf{x}_n denotes the n th column of matrix \mathbf{X} . $x(n)$ denotes the n th element of the vector \mathbf{x} . $\|\cdot\|_2$ denotes the vector ℓ_2 norm. \mathbf{I}_N , $\mathbf{1}_N$, and $\mathbf{0}_N$ denote an identity matrix of size N , N dimensional vector of ones, and N dimensional vector of zeros. $\text{Diag}(\cdot)$ returns a diagonal matrix with the input vector on the diagonal. \mathbf{X}^\top and \mathbf{X}^H denote the transpose and hermitian of \mathbf{X} .

2 Background

In a wireless communication setting, the transmitted signal suffers attenuation or even amplification, along with phase shift due to environmental factors. A channel matrix is used to model the such effect. To that end, we use the following signal model which is experimentally verified and physics backed [5]. Let $s \in \mathbb{C}$ denote the common signal to be transmitted from a transmitter with N antennas to M receivers with single antenna each. Suppose that $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_M] \in \mathbb{C}^{N \times M}$ denote the channel matrix where \mathbf{h}_m is the wireless channel from the transmitter to the m th user. Then the received signal $y_m \in \mathbb{C}^M$ by the m th user can be written as:

$$y_m = \mathbf{h}_m^H \mathbf{w} s + \eta_m,$$

where $\eta_m \in \mathbb{C}$ is a zero mean circularly symmetric random Gaussian noise of variance σ^2 and $\mathbf{w} \in \mathbb{C}^N$ is the precoding/beamforming vector computed by the transmitter. The problem of beamforming is to design the beamformer \mathbf{w} such that some utility function is maximized. A popular and useful utility function studied widely in the literature is the *quality of service* (QoS) formulation. Under the QoS formulation, the transmitter tries to meet a specified *signal to noise ratio* SNR at the receiving end, while using as small power as possible at the transmitter. This can be formulated as follows:

$$\begin{aligned} & \underset{\mathbf{w} \in \mathbb{C}^N}{\text{minimize}} \quad \|\mathbf{w}\|_2 \\ & \text{subject to} \quad \frac{|\mathbf{h}_m^H \mathbf{w}|}{\sigma^2} \geq \gamma_m, \forall m \in [M], \end{aligned} \quad (1)$$

where $\frac{|\mathbf{h}_m^H \mathbf{w}|}{\sigma^2}$ is the SNR achieved at receiver m , and $\gamma_m \in \mathbb{R}$ is the minimum specified SNR for receiver m . Problem (1) can be re-written as

$$\begin{aligned} & \underset{\mathbf{w} \in \mathbb{C}^N}{\text{minimize}} \quad \|\mathbf{w}\|_2 \\ & \text{subject to} \quad |\tilde{\mathbf{h}}_m^H \mathbf{w}| \geq 1, \forall m \in [M], \end{aligned} \quad (2)$$

where $\tilde{\mathbf{h}}_m = \frac{\mathbf{h}_m}{\gamma_m \sigma_m^2}$. Hence without loss of generality, we assume $\sigma_m^2 \gamma_m = 1, \forall m \in [M]$. Problem (1) was shown to be NP-hard in [1]. In this report, we show that when \mathbf{w} and \mathbf{H} are constrained to be real, the resulting beamforming problem is NP-complete. Therefore, we can use SMT solvers to find solutions to the problem under various setting.

3 NP completeness of the beamforming problem

Since irrational numbers cannot be represented by a finite size symbols, we restrict $\mathbf{H} \in \mathbb{Q}^{N \times M}$, i.e., rational numbers. Then the beamforming problem can be written as

$$\begin{aligned} & \underset{\mathbf{w} \in \mathbb{R}^N}{\text{minimize}} \quad \|\mathbf{w}\|_2 \\ & \text{subject to} \quad |\mathbf{h}_m^\top \mathbf{w}| \geq 1, \quad \forall m \in [M], \end{aligned} \quad (3)$$

Problem (3) is not a decision problem. We first consider the corresponding decision problem where we ask the question whether $\|\mathbf{w}\|_2 \leq \kappa, \kappa \in \mathbb{Q}_+$ exists such that $|\mathbf{h}_m^\top \mathbf{w}| \geq 1, \forall m \in [M]$. Let BF denote the language that consists of all yes instances of such decision problem, i.e.,

$$\text{BF} = \{(\mathbf{H}, \kappa) \mid \exists \mathbf{w} : \|\mathbf{w}\|_2 \leq \kappa, |\mathbf{h}_m^\top \mathbf{w}| \geq 1, \forall m \in [M]\},$$

At first glance, it appears obvious that $\text{BF} \in \text{NP}$ by taking \mathbf{w} to be the witness, and the conditions can be checked in polynomial time. However, for \mathbf{w} to be a valid witness, we have to show that the number of digits in \mathbf{w} is polynomial in that of the input (\mathbf{H}, κ) . To clarify, if the solution of (3) is \mathbf{w}^* and $\|\mathbf{w}^*\|_2 \leq \kappa$, then there exists a witness \mathbf{w} such that the following holds:

- \mathbf{w} satisfies the constraint in (3), and $\|\mathbf{w}\|_2 \leq \kappa$,
- \mathbf{w} is a rational number with number of digits polynomial in size of the total number of digits in (\mathbf{H}, κ)

Next, consider the following proposition:

Proposition 1. *BF is NP-complete.*

Proof. To prove the NP-completeness of BF, we first use the fact that quadratic programs are NP-complete [4, 6], and show that BF is in NP. Then we use the idea from [1] to establish that BF is NP-hard. Combining the two results, we conclude that BF is NP-complete.

3.1 BF \in NP

First, we show that BF is in NP by reducing BF to a quadratic program. We represent the decision problem associated with quadratic program as QP, which is defined as follows:

$$\text{QP} = \{(\mathbf{A}, \mathbf{b}, \mathbf{c}, \mathbf{D}, \kappa) \mid \exists \mathbf{w} : \mathbf{w}^\top \mathbf{A} \mathbf{w} + \mathbf{c}^\top \mathbf{w} \leq \kappa, \mathbf{D} \mathbf{w} \geq \mathbf{b}\}$$

where $\mathbf{A} \in \mathbb{S}^{P \times P}$, $\mathbf{D} \in \mathbb{Q}^{R \times P}$, $\mathbf{b} \in \mathbb{Q}^R$, $\mathbf{c} \in \mathbb{Q}^P$, $\kappa \in \mathbb{Q}_+$. [6] showed that $\text{QP} \in \text{NP}$. This implies that there exists a polynomial time TM T_{QP} such that:

$$\text{QP} = \{(\mathbf{A}, \mathbf{b}, \mathbf{c}, \mathbf{D}, \kappa) \mid \exists \mathbf{y} : T_{\text{QP}}((\mathbf{A}, \mathbf{b}, \mathbf{c}, \mathbf{D}, \kappa), \mathbf{y}) = 1\}.$$

where the witness \mathbf{y} is polynomial in the size of the input.

Let us define a variable $\mathbf{z} \in \{+1, -1\}^M$. Define $\tilde{\mathbf{A}}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}}, \tilde{\mathbf{D}}$ as follows:

$$\begin{aligned} \tilde{\mathbf{A}} &= \mathbf{I}_N \\ \tilde{\mathbf{D}}(\mathbf{z}) &= \text{Diag}(\mathbf{z}) \mathbf{H}^\top \\ \tilde{\mathbf{b}} &= \mathbf{1}_M \\ \tilde{\mathbf{c}} &= \mathbf{0}_N. \end{aligned}$$

Then one can write rewrite problem (3) as follows:

$$\begin{aligned} &\underset{\mathbf{w} \in \mathbb{Q}^N, \mathbf{z} \in \{+1, -1\}^M}{\text{minimize}} && \mathbf{w}^\top \tilde{\mathbf{A}} \mathbf{w} + \tilde{\mathbf{c}}^\top \mathbf{w} \\ &\text{subject to} && \tilde{\mathbf{D}}(\mathbf{z}) \mathbf{w} \geq \tilde{\mathbf{b}} \end{aligned} \tag{4}$$

It is straightforward to see that the objective of (3) is equivalent to that of (4). To see how the constraints are equivalent, observe that the m th constraint of (4) is

$$z(m) \mathbf{h}_m^\top \mathbf{w} \geq 1.$$

Since $z(m)$ can take the value $\{+1, -1\}$, the above inequality corresponds to

$$\text{either } \mathbf{h}_m^\top \mathbf{w} \geq 1 \text{ or } -\mathbf{h}_m^\top \mathbf{w} \geq 1,$$

which is equivalent to $|\mathbf{h}_m^\top \mathbf{w}| \geq 1$.

Note that problem (4) is not a quadratic program because $\tilde{\mathbf{D}}(\mathbf{z})$ is not a constant. The constraints are in fact non-convex, whereas for a quadratic program the constraints need to be affine. However, we can use existential quantifier to write the BF problem as follows:

$$\text{BF} = \{(\mathbf{H}, \kappa) \mid \exists \mathbf{z} \exists \mathbf{y} : T_{\text{QP}}((\tilde{\mathbf{A}}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}}, \tilde{\mathbf{D}}(\mathbf{z}), \kappa), \mathbf{y}) = 1\}.$$

Hence the concatenated string \mathbf{zy} take the role of a witness. Since, we know that for any value of $\mathbf{z} \in \{-1, 1\}^N$, $(\tilde{\mathbf{A}}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}}, \tilde{\mathbf{D}}(\mathbf{z}), \kappa)$ defines a quadratic program, the witness of QP, \mathbf{y} , is polynomial in size of the input due to [6]. Finally, as \mathbf{z} can be represented by N bits, it is polynomial in size of the input. Hence \mathbf{zy} is polynomial in size of the input.

Thus, write

$$\text{BF} = \{(\mathbf{H}, \kappa) \mid \exists \mathbf{zy} : T_{\text{BF}}((\mathbf{H}, \kappa), \mathbf{zy}) = 1\}$$

for a polynomial time TM T_{BF} detailed in Algorithm ??, where \mathbf{zy} is polynomial in the size of the input.

Finally, since the construction of the variables $\tilde{\mathbf{A}}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}}, \tilde{\mathbf{D}}(\mathbf{z})$ can be done in polynomial time, and T_{QP} is a polynomial time TM, T_{BF} is a polynomial time TM. Hence $\text{BF} \in \text{NP}$.

Algorithm 1 $T_{BF}((H, \kappa), yz)$

$\tilde{A} = I_N$
 $\tilde{D} = \text{Diag}(z)H^\top$
 $\tilde{b} = \mathbf{1}_M$
 $\tilde{c} = \mathbf{0}_N$
return $T_{QP}((\tilde{A}, \tilde{b}, \tilde{c}, \tilde{D}, \kappa), y)$

3.2 PARTITION \leq_p BF

PARTITION [4] can be Karp reduced to BF by using ideas from [1].

First, the problem PARTITION asks whether a list of integers a_1, \dots, a_N can be partitioned into two sets such that the sum of the two sets are equal. In other words, we search for variables $y_1, \dots, y_N \in \{+1, -1\}^N$ such that

$$\sum_{n=1}^N y_n a_n = 0.$$

Let $\mathbf{p} = [p_1, \dots, p_N]^\top \in \mathbb{R}^N$. The PARTITION problem is equivalent to asking whether the following problem as a solution of N :

$$\begin{aligned} & \underset{\mathbf{p} \in \mathbb{R}^N}{\text{minimize}} \quad p_1^2 + p_2^2 + \dots + p_N^2 + \left(\sum_{n=1}^N p_n a_n \right)^2 \\ & \text{subject to} \quad p_n^2 \geq 1, \quad n \in [N] \end{aligned} \tag{5}$$

If the above problem has solution N , then $p_n \in \{+1, -1\}$ and $\sum_{n=1}^N p_n a_n = 0$, i.e., a_1, \dots, a_n can be partitioned. Therefore, there is one to one correspondence of the above problem and PARTITION. However, the above problem is a special instance of BF. To see this, we can write the objective of (5) as $\mathbf{p}^\top \mathbf{Q} \mathbf{p}$, where $\mathbf{Q} = \mathbf{I}_N + \mathbf{a} \mathbf{a}^\top$, where $\mathbf{a} = [a_1, \dots, a_N]^\top$. Since \mathbf{Q} is a positive definite matrix, we can write $\mathbf{Q} = \mathbf{V}^\top \mathbf{V}$, for some full rank \mathbf{V} . Then we make a change of variable to $\tilde{\mathbf{w}} = \mathbf{V} \mathbf{p}$. This implies that $\mathbf{y} = \mathbf{V}^{-1} \mathbf{w}$. We represent $\tilde{\mathbf{h}}_n = \mathbf{V}^{-1}(n, :)^T$. Hence we can equivalently write (5) as follows:

$$\begin{aligned} & \underset{\tilde{\mathbf{w}} \in \mathbb{R}^n}{\text{minimize}} \quad \|\tilde{\mathbf{w}}\|_2^2 \\ & \text{subject to} \quad |\tilde{\mathbf{h}}_n| \geq 1, \quad n \in [N]. \end{aligned} \tag{6}$$

Therefore, $\mathbf{a} \in \text{PARTITION}$ if and only if $((\mathbf{V}^{-1})^\top, N) \in BF$, where \mathbf{V} is obtained by square root decomposition of $\mathbf{I}_N + \mathbf{a} \mathbf{a}^\top$. Since the square root decomposition via SVD, matrix inversion and multiplication can be done in polynomial time, PARTITION \leq_p BF. Hence BF is NP-hard

Since BF \in NP and BF is NP-hard, BF is NP-complete. \square

4 Encoding an instance of BF as a Boolean formula

In the previous section, we showed that BF is NP-complete. In order to encode an instance of BF as a Boolean formula, we restrict our attention to a fixed size representation (e.g., 64 bits) for each rational numbers, i.e., the input and the witness for BF. The reason is that if we merely assume that the witness is polynomial in size with respect to the input string, it is difficult to quantify exactly the number of variables and clauses. Secondly, the SMT solver that we will be using also represents the real numbers using a fixed size.

Let Q be the number of bits used to represent a single scalar. Since the number of scalar variables is N and Boolean variables is M , we have $NQ + M$ Boolean variables of interest in total. Note, however, that the total number of Boolean variables in the final formula can be far greater than

this, mainly due to the need for representing intermediate results as variables (e.g., each constraint will be converted into a Boolean formula with its own set of variables, and later, variables from all constraints are enforced to be equivalent).

In order to solve problem BF in practice, we will provide the following linear and quadratic inequalities to the SMT solver.

$$w(1)^2 + w(2)^2 + \cdots + w(N)^2 \leq \kappa \quad (7a)$$

$$z(1)(w(1)h_m(1) + w(2)h_m(2) + \cdots + w(N)h_m(N)) \geq 1, \forall m \in [M] \quad (7b)$$

The solver attempts to find a solution \mathbf{w}, \mathbf{z} that satisfy the above inequalities. In order to convert the above inequalities into a Boolean formula needed to represent the above inequalities, the following steps are followed:

1. obtain Boolean circuit blocks for summation, multiplication, and comparison,
2. compose the circuit blocks in order to obtain individual inequalities
3. convert the resulting Boolean circuit into Boolean formula, and
4. add clauses for ensuring variable consistency
5. count the size of the final Boolean formula.

Let $C_{\text{sum}}, C_{\text{mult}}$, and C_{comp} denote the size of the Boolean formula corresponding to the Boolean circuits for the summation, multiplication, and comparison of two scalar values represented by Q bits. For (7a), we require N multiplication blocks, $N - 1$ summation blocks, and 1 comparison block. For one inequality of (7b), we require N multiplication blocks, $N - 1$ summation blocks, 1 comparison block and one Boolean circuit for changing the sign of $\mathbf{w}_m^T \mathbf{h}$ according to the value of z_m . Let C_{sign} be the size of the Boolean formula associated with the Boolean circuit for changing the sign of a scalar represented by Q bits according to the value of another Boolean variable. In total, (7b) requires NM multiplications, $(N - 1)M$ summations, M sign determination, and M comparisons.

Finally, we need to ensure that the Boolean variables in the Boolean formula for each of the inequalities are consistent with each other. This can be done by adding extra clauses. For example, if $x^{(1)}, x^{(2)}, x^{(3)}$ denote the variables used in the 3 inequalities to represent the same variable x , then consistency between them can be enforced by adding the following clause

$$\left(\left(x^{(1)} \wedge x^{(2)} \wedge x^{(3)} \right) \vee \left(\bar{x}^{(1)} \wedge \bar{x}^{(2)} \wedge \bar{x}^{(3)} \right) \right).$$

Let C_{consist} denote the size of the boolean formula required to ensure consistency between two scalars taking Q bits each. The number of consistency checks required is $M - 1$.

From the above discussion, the total size of the boolean formula, C_{total} is approximately

$$C_{\text{total}} \approx (NM + N)C_{\text{mult}} + (N - 1)(M + 1)C_{\text{sum}} + (M + 1)C_{\text{comp}} + MC_{\text{sign}} + (M - 1)C_{\text{consist}}.$$

The above is an approximation because we have ignored the increase in size due to concatenation of different inequalities, and the additional consistency clauses that may be required to compose different blocks.

If we fix one of M and N constant, we can see that the size of the Boolean formula grows linearly with respect to the other parameter.

5 Simulations

In this section, we run simulations to observe the performance of problem BF for various problem setting. We use SMT solver, Z3 [7], to solve problem BF. Specifically, for different realizations of \mathbf{H} and κ , we observe whether $(\mathbf{H}, \kappa) \in \text{BF}$. Further, we observe relation between the various

problem parameters and the run-time of the SMT solver. The code used in the following simulations is available online¹.

Optimal Solution: In order to verify the correctness of the implementation, we obtain an optimal solution to the given problem with enumeration. The enumeration method works as follows. If we fix z to be $\hat{z} \in \{-1, +1\}^N$, problem (4) is a quadratic program, which is a convex program. Hence any convex programming solver can be used to obtain a solution arbitrarily close to the optimal. Therefore, we can evaluate problem (4) for all 2^M possible values of z and select the solution x^* and z^* that attains the minimum objective value v^* .

Data Generation: M and N can be varied arbitrarily. Note that the problem is feasible for any value of M, N . All entries of \mathbf{H} are sampled independently from a zero mean Gaussian distribution with variance 1. This is consistent with Rayleigh fading channel [5], which is a widely adopted channel model in wireless communications research. In Rayleigh fading channel the entries of \mathbf{H} is sampled from circularly symmetric Gaussian distribution. In the following experiments, σ_m^2 and γ_m are set to 1 for all $m \in [M]$.

5.1 Sanity Check

First, we verify the solution provided by the SMT solver by comparing it to the optimal solution. Table 1 shows the solution obtained for a particular realization of \mathbf{H} with $(N, M) = (2, 3)$ while κ is varied relative to the optimal solution v^* obtained by using the enumeration procedure. One can see that when $(\mathbf{H}, \kappa) \notin \text{BF}$, the solver takes much longer time to return failure. when $\kappa \geq v^*$, the solver returns a solution and successful set membership status in relatively small time.

Table 1: Set membership, objective value and time for various κ relative to the optimal objective value v^* .

κ	$(\mathbf{H}, \kappa) \in \text{BF}$	Objective	Time(seconds)
$0.5v^*$	False	-	0.29
$0.99v^*$	False	-	72.51
v^*	True	v^*	0.05
$1.1v^*$	True	$1.07v^*$	0.08
$1.5v^*$	True	$1.31v^*$	0.06
$2v^*$	True	$1.23v^*$	0.03

5.2 Further Experiments

Now, we observe solver time for various sizes of problem parameters, i.e., the number of antennas and users. Figure 1 shows the time taken by the solver for various parameter settings. The results are averaged over 15 randomly sampled \mathbf{H} .

In figure 1[Left] shows the time taken by the solver for varying number of antennas when the number of users is fixed to 4. Small scale scenarios such as with WiFi can be reflected by this problem size. However, large scale scenarios (e.g., a 5G base station serving many users can contain 10s of antennas) can not be covered. Here, we have set κ to $100v^*$. We can see that the time taken by the solver grows exponentially with the number of antennas. This is expected as the solving time for Boolean formula grows exponentially with the number of variables.

Figure 1[Right] shows the time taken by the solver for varying number of users when the number of antennas is fixed to 4. We can see that, similar to the previous case, the solving time increases exponentially with the number of users. However, time consumed seems to grow much faster with increase in N than M . The reason could be that the size of the constraint $\|\mathbf{w}\|_2^2 \leq \kappa$ grows only with the increase in N .

¹<https://github.com/shresthasagar/beamforming-np-completeness-toc.git>

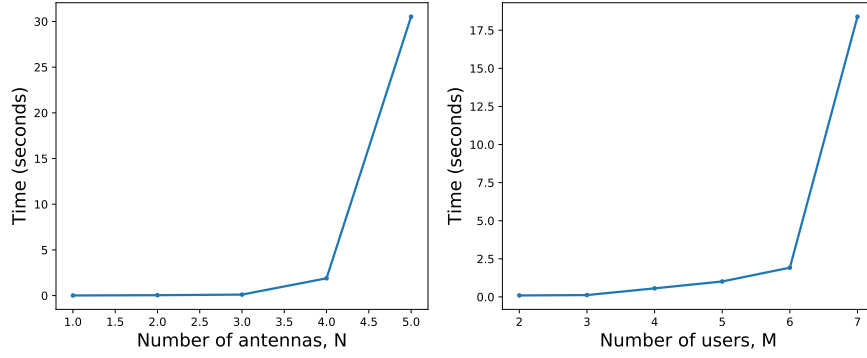


Figure 1: [Left] Time taken by the solver for various number of antennas, when $M = 4$. [Right] Time taken by solver for various number users, when $N = 4$.

6 Conclusion

In this project, we studied single group multicast beamforming problem in wireless communication in the case where both channel matrix and beamforming vector is real. We showed that, in such a case, the resulting decision problem is NP-complete. Using this knowledge, we first analysed the size of the Boolean formula needed to represent the problem using finite precision for each scalar. Finally, we observed in simulations the dependence of the running time on various parameters of the beamforming problem.

We should remark that experiments with larger problem size compared to the ones shown in Figure 1 seem to be infeasible (some problem instances take more than 5 minutes to solve) in general.

References

- [1] N. D. Sidiropoulos, T. N. Davidson, and Z.-Q. Luo, "Transmit beamforming for physical-layer multicasting," *IEEE transactions on signal processing*, vol. 54, no. 6, pp. 2239–2251, 2006.
- [2] A. Konar and N. D. Sidiropoulos, "A fast approximation algorithm for single-group multicast beamforming with large antenna arrays," in *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2016, pp. 1–5.
- [3] C. Lu and Y.-F. Liu, "An efficient global algorithm for single-group multicast beamforming," *IEEE Transactions on Signal Processing*, vol. 65, no. 14, pp. 3761–3774, 2017.
- [4] M. R. Garey and D. S. Johnson, *Computers and intractability*. freeman San Francisco, 1979, vol. 174.
- [5] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.
- [6] S. A. Vavasis, "Quadratic programming is in np," *Information Processing Letters*, vol. 36, no. 2, pp. 73–77, 1990.
- [7] L. d. Moura and N. Bjørner, "Z3: An efficient smt solver," in *International conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2008, pp. 337–340.