

Array Spatial Thinning for Interference Mitigation by Semidefinite Programming

Hamed Nosrati

School of Electrical and

Telecommunications Engineering

University of New South Wales Australia

Data61,CSIRO

Email: hamed.nosrati@unsw.edu.au

Elias Aboutanios

School of Electrical and

Telecommunications Engineering

University of New South Wales Australia

Email: elias@unsw.edu.au

David B. Smith

Data61,CSIRO

Australian National University

Email:David.Smith@data61.csiro.au

Abstract—We study the problem of interference mitigation in a phased array, where a subset containing k out of a total of N receivers creates a virtual spatial null for an incoming interference. The signal-of-interest and interference are represented by their corresponding steering vectors, and an optimum subarray is chosen such that the two vectors are as orthogonal as possible. This optimization is a binary quadratic non-convex minimization. We propose a semidefinite programming method to find suboptimal solutions using an optimal randomized sampling strategy. We show that the proposed method provides solutions as good as an exhaustive search with a cubic computational complexity. Furthermore, the proposed algorithm outperforms existing methods by solving the problem in a higher dimensionality.

Index Terms—Array thinning, antenna selection, binary quadratic constrained programming, semidefinite programming, convex optimization.

I. INTRODUCTION

Assuming a large phased array, a thinned array is a subset containing some elements of the full large array that are usually chosen to give a desirable performance. Spatial array thinning provides three important advantages. First, in a large phased array system with substantial hardware requirements, a thinned array reduces the hardware cost in terms of required blocks at the front-end, such as low noise amplifiers (LNA), phase shifters, and frequency mixers. Second, in dense arrays where power dissipation is a matter of concern, a well-designed thinned array can provide enough sparsity to alleviate heating problems [1]. Third, from a signal processing perspective, the dimensionality and the computational cost of the system are significantly reduced by a smaller subarray.

In addition to the above advantages, array thinning can be revisited in an interference cancellation strategy. Such a strategy eliminates undesirable radio interference from the origin by creating spatial nulls. Wang et al. proposed a reconfigurable array for interference mitigation [2], [3]. Employing spatial correlation coefficient (SCC) [4] as a cost function, they formulated the thinning problem to make the steering vectors of the signal and interference as orthogonal as possible. The corresponding problem proposed in [2], is a constrained binary quadratic non-convex optimization which is solvable in polynomial time. The computational complexity of these

methods, which enumerate all possible binary points within a box or ellipsoid, is exponential in N [5]. Therefore, many heuristic approximation methods have been proposed to locate the solution at a lower cost. With growing dimensionality, approximation algorithms become vital as the problem can not be solved in reasonable time.

A first attempt at approximation can be done by relaxation and finding lower bounds. More specifically, a simple lower bound is achieved for binary optimization by relaxing the binary constraint within $[0, 1]$. The lower bound is then obtained by rounding the optimum solution to the nearest binary value. However, the obtained solution is, in general, neither optimal nor feasible. An effective way to relax the binary programming problem is semidefinite programming (SDP), which has been proven to provide the tightest bound [6], [7]. In SDP, the optimization is rearranged by introducing a rank-1 matrix variable. Dropping the rank-1 constraint, the problem is then relaxed to one of SDP.

Although the quality of the obtained lower bound is guaranteed, our primary goal is to find a feasible solution. Goemans and Williamson proved in their well-known paper that it is possible to find acceptable values for the max-cut case using the bounds provided by SDP [8]. They reformulated the NP-hard max-cut problem as a Boolean non-convex quadratic minimization. After relaxing the problem by SDP, they proposed a randomized sampling algorithm to find the optimal values using a solution of the relaxed problem.

The randomized rounding algorithm proposed in [8] has been applied in different applications, e.g. [9]–[11]. Park and Boyd presented a randomized algorithm for finding good suboptimal solutions for integer convex quadratic minimization by a probabilistic interpretation of SDP [12]. They showed the effectiveness of their algorithm for the NP-hard integer programming case. Nonetheless, the binary nonconvex case has not been studied. *In this paper we investigate a quadratic minimization containing both binary and non-convex constraints.* Similar problems such as clique, graph coloring, and constrained max-cut in the field of graph optimizations are challenging and command an active research profile [13], [14].

Our main contribution is reformulating the quadratic SCC minimization problem as an SDP and proposing a new ran-

domized rounding algorithm inspired by [12]. After reviewing the spatial array thinning framework in Section II, two relevant existing algorithms are studied in Sections III and IV. The new method based on SDP is proposed in Section V and the computational complexity of the algorithms is discussed in Section VI. Simulation results using a rectangular phased array are given in VII to demonstrate the effectiveness of the proposed algorithm. Finally, some conclusions are drawn in Section VIII.

II. SPATIAL ARRAY THINNING FRAMEWORK

Let matrix \mathbf{P} be the locations of a set of N receivers in a two dimensional plane,

$$\mathbf{P} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_N & y_N \end{bmatrix}. \quad (1)$$

Given the normal vector pointing to a plane wave, coming from elevation and azimuth (θ_i, ϕ_i)

$$\mathbf{u}_i = [\sin \theta_i \cos \phi_i \quad \sin \theta_i \sin \phi_i]^T, \quad (2)$$

the spatial steering vector corresponding to the signal of interest and interference are respectively

$$\mathbf{v}_s = e^{j \frac{2\pi}{\lambda} \mathbf{P} \mathbf{u}_s}, \mathbf{v}_j = e^{j \frac{2\pi}{\lambda} \mathbf{P} \mathbf{u}_j}. \quad (3)$$

To characterize the spatial separability of the signal and interference in the assumed configuration the correlation between the two steering vectors are measured by the spatial correlation coefficient (SCC) [4],

$$\alpha_{js} = \frac{\mathbf{v}_j^H \mathbf{v}_s}{\|\mathbf{v}_j\| \|\mathbf{v}_s\|} = \frac{\mathbf{v}_j^H \mathbf{v}_s}{\sqrt{\mathbf{v}_j^H \mathbf{v}_j} \sqrt{\mathbf{v}_s^H \mathbf{v}_s}} = \frac{\mathbf{v}_j^H \mathbf{v}_s}{N^2}. \quad (4)$$

Now we aim to find a subset containing k elements. That makes the steering vectors as orthogonal as possible and minimizes the SCC. This is an optimization problem, where the quadratic cost function is given by the weighted norm of a selection vector, \mathbf{c}

$$|\alpha_{js}|^2 = \frac{\|\mathbf{c}\|_{\mathbf{W}_r}^2}{k^2} = \frac{\mathbf{c}^T \mathbf{W}_r \mathbf{c}}{k^2}. \quad (5)$$

Here \mathbf{c} is a vector comprising entries that are 1 or 0, with 1 indicating that the corresponding element is selected and 0 that it is not. The matrix \mathbf{W}_r is defined as

$$\mathbf{W}_r = \text{real}(\mathbf{v}_{js} \mathbf{v}_{js}^H), \quad (6)$$

and

$$\mathbf{v}_{js} = \mathbf{v}_s \odot \mathbf{v}_j^T. \quad (7)$$

For the vector \mathbf{v} , we use \mathbf{v}^H and \mathbf{v}^T to denote the complex Hermitian transpose and transpose respectively. The notation \odot represents Hadamard product and $\text{real}(\cdot)$ denotes the real

Algorithm 1 Correlation Measurement Algorithm [2]

Set all candidate antenna selected i.e. $\mathbf{c} = \mathbf{1}^N$,

In each iteration n :

- 1: Let $i := \text{argmax}_{l=1, \dots, N} \sum_{j=1}^N \mathbf{W}_{r,j,l}$
 - 2: Delete antenna i , i.e. $\mathbf{c}(i) = 0$,
Set the i_{th} column and i_{th} row of \mathbf{W}_r to zero,
Put $n := n + 1$,
 - 3: If $n = N - K$, terminate, otherwise go back to step 1.
-

part of a complex matrix or vector. The problem can be cast as a binary quadratic programming [2]:

$$\begin{aligned} \min_{\mathbf{c}} \quad & |\alpha_{js}|^2 \\ \text{s.t.} \quad & c_i(c_i - 1) = 0 \quad i = 1 \dots N, \\ & \text{and } \mathbf{c}^T \mathbf{c} = k. \end{aligned} \quad (8)$$

This is a non-convex problem where we are minimizing a quadratic form over the vertices of an N -dimensional hypercube ($c_i \in \{0, 1\}$), that coincides with the surface of a hypersphere ($\mathbf{c}^T \mathbf{c} = k$). Problem (8) can be solved in factorial time by a simple exhaustive search. However, the solution time increases rapidly as the dimensionality grows. Therefore, approximation algorithms are required to find suitable suboptimal solutions.

III. CORRELATION MEASUREMENT

Correlation measurement (CM) is a greedy approach that decreases the number of elements iteratively [15]. The square of the SCC, specified in (5), is actually the sum of the entries of matrix \mathbf{W}_r corresponding to selected elements in vector \mathbf{c} ,

$$|\alpha_{js}|^2 = \frac{\mathbf{c}^T \mathbf{W}_r \mathbf{c}}{k^2} = \frac{1}{k^2} \sum_{i,j=1}^N c_i c_j \mathbf{W}_{r,i,j}. \quad (9)$$

As shown in Algorithm 1, the CM method starts with a full array and finds in each step the element that gives the maximum total correlation. This element is then removed and the process repeated until a set of k elements with the least total correlation remains. While, correlation measurement is a heuristic method and it is difficult to study its performance, some performance bounds can be established [2]:

- 1) The CM algorithm finds the global solution if $|\mathbf{u}_s - \mathbf{u}_j| \leq \frac{1}{N-1} \frac{\lambda}{2d}$, where d and λ denote the inter-element spacing and wavelength respectively.
- 2) When $|\mathbf{u}_s - \mathbf{u}_j|$ is sufficiently large, the distance between the provided solution and the global minimum is maximally $\frac{1}{k^2}$.

IV. DCS PROGRAMMING

The non-convex binary quadratic programming specified by (8) can be viewed from another perspective. Using the manifold of difference of two convex sets (DCS) we can relax the binary constraints and formulate an iterative linear programming (LP) problem that yields an SCC lower than a certain value [16]. Wang *et al.* used this property to control the

SCC at different points and avoid the high sidelobes caused by non-uniformly spaced elements [2]. Thus, in addition to the original constraints, another set of constraints derived by sampling the \mathbf{u} -space, permits the SCC to be controlled. Let the SCC corresponding to the original DOAs of the signal and interference be less than some value δ , and sample the \mathbf{u} -space interval $[-1, 1]$ at L points with correlation vectors $\mathbf{v}_{j_s}^i$ and desired SCCs δ^i , for $i = 1, \dots, L$. Then the binary vector satisfying these constraints is found by running Algorithm 2 for a sufficient number of steps.

V. SEMIDEFINITE PROGRAMMING

The binary constraints can be relaxed by semidefinite programming. We approach this method by finding the lower bound and subsequently computing the solution. Introducing a new matrix variable $\mathbf{C} \in \mathbb{R}^{N \times N}$, we reformulate the minimization problem (8) as

$$\begin{aligned} \min_{\mathbf{C}, \mathbf{c}} \quad & \frac{1}{k^2} \text{Tr}(\mathbf{C}\mathbf{W}_r) \\ \text{s.t.} \quad & \text{diag}(\mathbf{C}) = \mathbf{c} \\ & \mathbf{C} = \mathbf{c}\mathbf{c}^T \\ & \text{Tr}(\mathbf{C}) = k. \end{aligned} \quad (10)$$

The binary constraint $c_i(c_i - 1) = 0$ is satisfied by requiring that, $\text{diag}(\mathbf{C}) = \mathbf{c}$, and $\mathbf{C} = \mathbf{c}\mathbf{c}^T$. Also, $\text{Tr}(\mathbf{C}) = k$, ensures that the number of active elements is equal to k and makes the non-convex constraint $(\mathbf{c}^T \mathbf{c} = k)$ convex. The only remaining non-convexity arises from $\mathbf{C} = \mathbf{c}\mathbf{c}^T$. This is relaxed by a positive semidefinite constraint, $\mathbf{C} \succeq \mathbf{c}\mathbf{c}^T$. Using the Schur complement, the semidefinite convex programming is recast as,

$$\begin{aligned} \min_{\mathbf{C}, \mathbf{c}} \quad & \frac{1}{k^2} \text{Tr}(\mathbf{C}\mathbf{W}_r) \\ \text{s.t.} \quad & \text{diag}(\mathbf{C}) = \mathbf{c} \\ & \text{Tr}(\mathbf{C}) = k \\ & \begin{bmatrix} \mathbf{C} & \mathbf{c} \\ \mathbf{c}^T & 1 \end{bmatrix} \succeq 0. \end{aligned} \quad (11)$$

It has been proven in [17] that the optimum value obtained by semidefinite convex optimization, (11), gives a lower bound for the primal problem in (8).

Algorithm 2 Controlled Quiescent Algorithm [2]

In each iteration n :

$$\begin{aligned} 1: \quad & \min_{\mathbf{c}_n} \mathbf{1}^T \mathbf{c}_n \\ \text{s.t.} \quad & \frac{|\mathbf{c}_n^T \mathbf{v}_{j_s}|}{k} \leq \delta \\ & \frac{|\mathbf{c}_n^T \mathbf{v}_{j_s}^i|}{k} \leq \delta^i, i = 1, \dots, L \\ & \mathbf{c}_n \in [0, 1]^N \\ & 2\mathbf{c}_{n-1}^T \mathbf{c}_n - \mathbf{c}_{n-1}^T \mathbf{c}_{n-1} = k. \end{aligned}$$

2: If $\|\mathbf{c}_n - \mathbf{c}_{n-1}\| \leq \epsilon$ terminate, otherwise
 Put $n := n + 1$,
 Go back to step 1.

The reformulation provided in the set of equations (11) can be interpreted as a nonlinear lifting to a higher dimensional space. In fact, by introducing matrix \mathbf{C} , we are solving the problem in a $N \times N$, rather than N , dimensional space. The rank-1 constraint ($\mathbf{C} = \mathbf{c}\mathbf{c}^T$) in (10) was relaxed by a positive definiteness constraint. If we find an optimal solution for (11) that has rank-1 and trace k , then we would have solved the primal problem. However, in general the optimal solutions do not meet the relaxed constraints. Nevertheless, if the optimal SDP solution does not comply with the primary constraints, it is possible to achieve a suboptimal solution by an appropriate randomized rounding strategy [8].

Let $(\mathbf{C}^*, \mathbf{c}^*)$ be the optimal value obtained by semidefinite relaxation. Also, assume $\mathbf{z} \in \mathbb{R}^N$ is a normal Gaussian variable with distribution $\mathbf{z} \sim \mathcal{N}(\mu, \Sigma)$. Then, the following minimization problem optimizing the expected value of a quadratic form

$$\begin{aligned} \min_{\mathbf{z}} \quad & \mathbb{E} \left(\frac{1}{k^2} (\mathbf{z}^T \mathbf{W}_r \mathbf{z}) \right) \\ \text{s.t.} \quad & \mathbb{E}(z_i(z_i - 1)) = 0 \\ & \mathbb{E}(\mathbf{z}^T \mathbf{z}) = k, \end{aligned} \quad (12)$$

is solved by \mathbf{z} for $\mu = \mathbf{c}^*$ and $\Sigma = \mathbf{C} - \mathbf{c}^{\star T} \mathbf{c}^*$ [12]. In short, the idea here is to solve the SDP relaxed problem (11), obtain the optimal solution $(\mathbf{C}^*, \mathbf{c}^*)$ and then sample the suboptimal solution from \mathbf{z} . The randomized sampling does not immediately provide a feasible point for the original problem (8). Hence, we need to round it to the nearest binary point that satisfies the sum constraint ($\mathbf{c}^T \mathbf{c} = k$). The constrained rounding meeting all the above criteria is another form of binary programming that is not preferred. Thus, we heuristically design an algorithm for rounding based on optimal rounding under integer constraints (ORIC) proposed in [18]. Given a vector of N positive real numbers \mathbf{x} with a total sum of M ($\sum_{i=1}^N x_i = M$), ORIC finds the closest integer vector \mathbf{y} , by solving the following minimization problem,

$$\begin{aligned} \min_{\mathbf{y}} \quad & \|\mathbf{x} - \mathbf{y}\| \\ \text{s.t.} \quad & \sum_{i=1}^N y_i = M. \end{aligned}$$

Our heuristic approach employs controlled random sampling. After sampling \mathbf{z} , we first check whether the sum of elements is within a tolerable bound, i.e. close enough to k , so that the following ORIC algorithm gives a vector with total sum of k . Then, we feed the collected random vector into ORIC and inspect the output to determine if it meets the binary constraint. The proposed method is summarized in Algorithm 3.

VI. COMPUTATIONAL COMPLEXITY

In this section we briefly study the computational complexity of the methods listed in Algorithms 1 to 3. Algorithm 1 sums the columns of \mathbf{W}_r in each step and takes $\mathcal{O}(N^2)$ operations subsequently. Algorithm 2 solves a linear program with

$L + N + 2$ constraints in each step. Assuming that LP requires $\mathcal{O}(n^2m)$ operations for m constraints [17], the overall computational complexity of Algorithm 2 is $\mathcal{O}(N^2(L + N + 2))$ or $\mathcal{O}(N^3)$. Solving the SDP problem in step 1 of the Algorithm 3 using an interior point method with constant iteration requires $\mathcal{O}(N^3)$ operations. By employing Cholesky decomposition, the random sampling in step 4 takes $\mathcal{O}(N^3/3)$ once, followed by $\mathcal{O}(N^2)$ for each iteration. Moreover, the ORIC algorithm used in step 6 has a complexity of $\mathcal{O}(N \log(N))$. Hence, the computational complexity in M iterations is $\mathcal{O}(N^3 + MN^2)$ or $\mathcal{O}(N^3)$. Noting that the relaxed problem is solved in an augmented dimension of $N \times N$, the complexity becomes $\mathcal{O}(N^6)$ compared to the primal problem.

VII. SIMULATION

In order to illustrate the performance of the algorithms presented above, we run a simulation in this section. We consider a rectangular array comprising 4×4 grid of antennas. The elevation and azimuth angles of the signal of interest are fixed at $\phi_s = 0.15\pi, \theta_s = 0.25\pi$. The azimuth of the interference, ϕ_j , varies from 0 to $\frac{\pi}{2}$, whereas the elevation is fixed at $\theta_j = 0.4\pi$. We select $k = 10$ antennas out of a total 16 elements. In addition to Algorithms 1-3, we include the results of the exhaustive search. To make the algorithms comparable, the additional L SCC constraints are eliminated from Algorithm 2, and δ is assumed to be the minimum value obtained by exhaustive search in each DOA scenario. Also, we limit the number of iterations to 50 for Algorithm 2, and set the number of randomization iterations for Algorithm 3 to 1000. For each scenario, we run a Monte Carlo simulation with 1000 trials and then calculate the mean value of $|\alpha_{js}|^2$. As shown in Fig. 1, the exhaustive search outperforms all other algorithms as it enumerates all possible candidates. The SDP method (Algorithm 3) exhibits the closest performance to the exhaustive search. The SCC squared value achieved by the correlation measurement listed in Algorithm 1 is notably large confirming that it finds a suboptimum solution. All three curves coincide in the sequence centered at 0.15π (≈ 0.47), which is the azimuth of the signal of interest. It is due to the fact that in this region the signal and interference come from directions that are very close to each other. This also confirms the first performance bound mentioned in Section

Algorithm 3 Constrained Randomized Algorithm

- 1: Solve SDP relaxed problem (11) to get $(\mathbf{C}^*, \mathbf{c}^*)$
- 2: Compute the covariance matrix $\Sigma = \mathbf{C} - \mathbf{c}^* \mathbf{c}^{*T}$
- 3: Initialize the best point $\mathbf{c}^{\text{best}} := \mathbf{0}$ and $f^{\text{best}} = |\alpha_{js}|^2(\mathbf{c}) := 1$
In each iteration n :
- 4: Random sampling $\mathbf{z}^{(n)} \sim \mathcal{N}(\mu, \Sigma)$
- 5: If $\left| \sum_{i=1}^N z_i^{(n)} - k \right| < 0.5$ continue, otherwise go to step 4
- 6: Compute $\mathbf{y} = \text{ORIC}(\mathbf{z}^{(n)})$
- 7: If $y_i \in \{0, 1\}$ for $i = 1, \dots, N$ then $\mathbf{c}^{(n)} = \mathbf{y}$ and continue, otherwise go to step 4
- 8: If $f(\mathbf{c}^{(n)}) < f^{\text{best}}$ update $\mathbf{c}^{\text{best}} := \mathbf{c}^{(n)}$ and $f^{\text{best}} := f(\mathbf{c}^{(n)})$

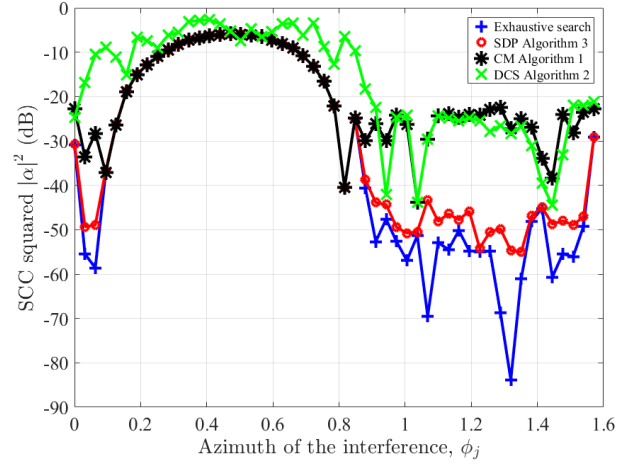


Fig. 1. Optimum SCC squared value achieved by different optimization algorithms versus exhaustive search in different interference scenarios.

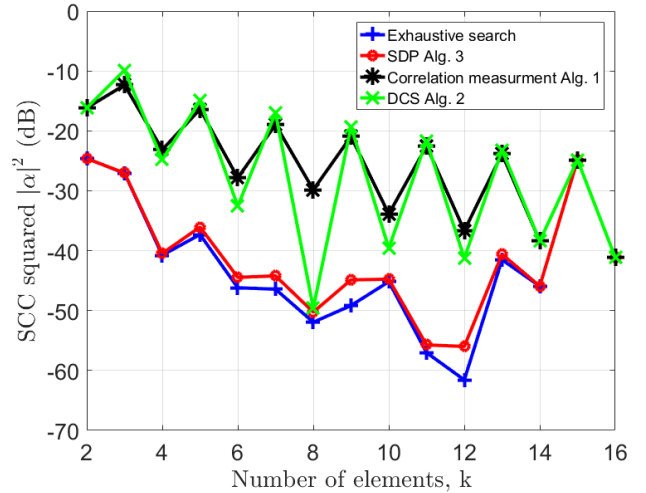


Fig. 2. Optimum SCC squared value achieved by different optimization algorithms versus exhaustive search for different number of antenna elements.

III. Beyond the mentioned region, the controlled quiescent algorithm (Algorithm 2) performs better than Algorithm 1 but worse than Algorithm 3, however, its advantage lies in its ability to control the array response in different directions.

The azimuth of the interference is then fixed at $\phi_j = 0.45\pi$ for the next simulation, while the algorithms are studied for selecting different number of elements. As shown in Fig. 2, the proposed algorithm based on SDP relaxation provides an optimum SCC squared value that is close to the exhaustive search. While both Algorithms 1 and 2 exhibit a worse performance, with Algorithm 2 giving a better results than Algorithm 1.

Let $|\alpha_{js}|^2$ be the objective function, $f(\mathbf{c})$, for the selection vector \mathbf{c} , and define an approximation ratio, ρ as follows

REFERENCES

$$\rho = \frac{f(\mathbf{c}^{appr})}{f(\mathbf{c}^{opt})}, \quad (13)$$

where \mathbf{c}^{appr} and \mathbf{c}^{opt} denote the selection vector achieved by approximation and optimal solution obtained by exhaustive search respectively. The maximum value, mean value, and standard deviation of the approximation ratio achieved by each algorithms are reported in Table I. Algorithm 1 approximates the solution with the largest ratio for both simulation scenarios. It also has a high standard deviation implying a lack of robustness. Algorithm 3 exhibits very good reliability and performance, closely approximating the solution. While, Algorithm 2 occupies the middle ground between the other two.

Finally, we make some observations regarding the performance and computational complexity. Firstly, although Algorithm 1 has the best computational complexity, it gives a relatively large SCC value and lacks any closed form that would enable further theoretical analysis. Secondly, Algorithm 2 is capable of controlling the array response, with a complexity of $\mathcal{O}(N^3)$ and has a closed form solution. Thirdly, Algorithm 3 achieves an objective value close to that returned by the exhaustive search method, while requiring a complexity of $\mathcal{O}(N^6)$. Moreover, extensive literature on the theory and analysis of Algorithm 3 exists, allowing it to be studied in terms of approximation ratio and making it suitable to be extended using the stochastic programming [19].

VIII. CONCLUSION

We have proposed semidefinite programming, followed by an optimal randomized sampling, for radio interference mitigation. Our approach exploits the probabilistic features of SDP to solve, as non-convex, the problem of interference cancellation by array thinning. Employing an optimal rounding method, we proposed a heuristic randomization comprising binary and box constraints. Finally, we provided simulation results confirming the effectiveness of the proposed algorithm compared to exhaustive search as well as its superiority in terms of approximation ratio compared to algorithms that solve the problem in a lower dimensional space.

TABLE I

COMPUTATIONAL COMPLEXITY, MAXIMUM, MEAN AND STANDARD DEVIATION OF THE APPROXIMATION RATIO FOR DIFFERENT ALGORITHMS AND SIMULATION SCENARIOS.

Alg.	Op.	ρ , sim #1			ρ , sim #2		
		max	mean	std	max	mean	std
1	$\mathcal{O}(N^2)$	1600	241	528	2800	325	718
2	$\mathcal{O}(N^3)$	733	126	263	107	21	31
3	$\mathcal{O}(N^6)$	4.41	1.46	1.06	7	1.65	1.51

- [1] D. G. Leeper, "Isophoric arrays - massively thinned phased arrays with well-controlled sidelobes," *IEEE Trans. Antennas Propag.*, vol. 47, no. 12, pp. 1825–1835, 1999.
- [2] X. Wang, E. Aboutanios, M. Trinkle, and M. G. Amin, "Reconfigurable Adaptive Array Beamforming by Antenna Selection," *IEEE Trans. Signal Process.*, vol. 62, no. 9, pp. 2385–2396, 2014.
- [3] X. Wang, E. Aboutanios, and M. G. Amin, "Thinned array beampattern synthesis by iterative soft-thresholding-based optimization algorithms," *IEEE Trans. Antennas Propag.*, vol. 62, no. 12, pp. 6102–6113, 2014.
- [4] H.-C. Lin, "Spatial correlations in adaptive arrays," *IEEE Trans. Antennas Propag.*, vol. 30, no. 2, pp. 212–223, 1982.
- [5] C. Buchheim, A. Caprara, and A. Lodi, "An effective branch-and-bound algorithm for convex quadratic integer programming," *Math. Program.*, vol. 135, no. 1-2, pp. 369–395, 2012.
- [6] C. Buchheim and A. Wiegele, "Semidefinite relaxations for non-convex quadratic mixed-integer programming," *Math. Program.*, vol. 141, no. 1-2, pp. 435–452, 2013.
- [7] Z.-q. Luo, W.-k. Ma, A. So, Y. Ye, and S. Zhang, "Semidefinite Relaxation of Quadratic Optimization Problems," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, 2010.
- [8] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, vol. 42, no. 6, pp. 1115–1145, 1995.
- [9] Y. Ye, "A .699-approximation algorithm for Max-Bisection," *Math. Program.*, vol. 90, no. 1, pp. 101–111, 2001.
- [10] S. He, Z.-Q. Luo, J. Nie, and S. Zhang, "Semidefinite Relaxation Bounds for Indefinite Homogeneous Quadratic Optimization," *Bernoulli*, vol. 19, no. 2, pp. 1–23, 2007.
- [11] Z. Xu, M. Hong, and Z.-Q. Luo, "Semidefinite Approximation for Mixed Binary Quadratically Constrained Quadratic Programs," *SIAM J. Optim.*, vol. 24, no. 3, pp. 1265–1293, 2014.
- [12] J. Park and S. Boyd, "A Semidefinite Programming Method for Integer Convex Quadratic Minimization," no. 1, p. 24, 2015.
- [13] I. Dukanovic and F. Rendl, "Semidefinite programming relaxations for graph coloring and maximal clique problems," *Math. Program.*, vol. 109, no. 2-3, pp. 345–365, jan 2007.
- [14] C. Wu, D. Du, and D. Xu, "An improved semidefinite programming hierarchies rounding approximation algorithm for maximum graph bisection problems," *J. Comb. Optim.*, vol. 29, no. 1, pp. 53–66, 2015.
- [15] Y. Selen, H. Tullberg, and J. Kronander, "Sensor Selection for Cooperative Spectrum Sensing," *2008 3rd IEEE Symp. New Front. Dyn. Spectr. Access Networks*, no. i, pp. 1–11, 2008.
- [16] H. Tuy, *Convex Analysis and Global Optimization*, ser. Springer Optimization and Its Applications. Springer International Publishing, 2016, vol. 110.
- [17] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Univ Press, 2010, vol. 25, no. 3.
- [18] R. Cont and M. Heidari, "Optimal rounding under integer constraints," *arXiv preprint arXiv:1501.00014*, 2014.
- [19] B. Gaujal and P. Mertikopoulos, "A stochastic approximation algorithm for stochastic semidefinite programming," *Probab. Eng. Information Sci.*, vol. 30, no. 03, pp. 431–454, 2016.