

FAST UNIT-MODULUS LEAST SQUARES WITH APPLICATIONS IN TRANSMIT BEAMFORMING

John Tranter*, Nicholas D. Sidiropoulos*, Xiao Fu*, and Ananthram Swami†

*Department of ECE, University of Minnesota, Minneapolis, MN, USA

†Army Research Laboratory, Adelphi, MD, USA

ABSTRACT

This paper considers the Unit-modulus Least Squares (ULS) problem, which is commonly seen in signal processing applications, e.g., phase-only beamforming, phase retrieval and radar code design. ULS formulations are easily reformulated as Unit-modulus Quadratic Programs (UQPs), to which Semi-Definite Relaxation (SDR) can be applied, and is often the state-of-the-art approach. SDR has the drawback of squaring the number of variables, which lifts the problem to much higher dimension and renders SDR ill-suited for large-scale ULS/UQP. In this work, we propose first-order algorithms that meet or exceed SDR performance in terms of (approximately) solving ULS problems, and also exhibit much more favorable runtime performance relative to SDR. We specialize to phase-only beamformer design, which entails additional degrees of freedom that we point out and exploit in two custom algorithms that build upon the general first-order algorithm for ULS/UQP. Simulations are used to showcase the effectiveness of the proposed algorithms.

1. INTRODUCTION

There exist many applications in which Unit-modulus Least Squares (ULS) problems arise, notably in signal processing, wireless communications, and radar [1–5]. ULS problems can be algebraically transformed into Unit-modulus Quadratic Programs (UQPs), and vice-versa; therefore, algorithms designed for ULS can be used for UQP, and vice-versa. Unfortunately, the general ULS/UQP problem is non-convex and NP-hard [6]. Arguably the most popular approach is to pose ULS/UQP as a Quadratically Constrained Quadratic Program (QCQP), to which Semi-Definite (rank) Relaxation (SDR) can be applied [7]. After finding an optimal solution to the relaxed convex problem, a feasible solution to the original UQP problem is constructed following some standard procedures, e.g., randomization [7].

Computational complexity is the main challenge in applying SDR to ULS/UQP. If the original ULS/UQP problem has N optimization variables, SDR requires lifting the optimization problem to N^2 dimensions. This lifting results

in $\mathcal{O}(N^{6.5})$ complexity if the relaxed problem is solved by interior-point methods, which may render the problem impractical for large problem size. In addition, squaring the number of variables is memory-inefficient – storing N^2 variables is not an easy task when N is large. On the other hand, large-scale ULS/UQP arises increasingly often in signal processing. For example, massive Multiple-Input-Multiple-Output (MIMO) communication systems have recently drawn much attention, as employing large-scale antenna arrays enables multi-faceted performance improvements [8]. Phase retrieval and radar code design are also likely to require high-dimensional optimization. Therefore, it is well-motivated to find alternatives to SDR.

In this paper, we propose to employ a gradient projection (GP)-based algorithm to tackle the ULS/UQP problem. GP remains in the original problem domain without lifting the number of variables, and thus is considered more efficient in terms of both memory and simplicity of computation. Unlike the classical GP algorithm that in general works on convex constraint sets, the proposed algorithm works with the non-convex element-wise unit modulus constraint, i.e., every scalar variable is constrained to lie on the unit circle in the complex plane. Nevertheless, since projection onto the unit modulus constraint is computationally easy, the simplicity of the classical GP is maintained in our case. The difficulty is that non-convex constraints are in general considered not suitable for GP, since projection onto such sets may increase the cost function. Interestingly, without resorting to monotonic decrease, we still show that the proposed algorithm converges to a Karush-Kuhn-Tucker (KKT) point. We also specialize the GP idea to a phase-only array beamforming problem [4, 5, 9] that has significance in massive MIMO and radar design, and provide two different modified algorithms that aim at enhancing performance in this specific application. Simulations are performed to showcase the effectiveness of the algorithms.

2. PROBLEM STATEMENT

What is known as *phased-array* beamforming generally assumes control of both magnitude and phase for each antenna, but here we are interested in *phase-only* beamforming, which

*E-mail: (trant004,nikos,xfu)@umn.edu. Supported in part by ARO STIR W911NF-15-1-0384. †E-mail: a.swami@ieee.org.

only controls the phase of each antenna. This allows using only one power amplifier that is common to all antennas (instead of separate per-antenna amplifiers), and also avoids the high peak-to-average power ratio (PAPR) problem across antennas, which is important in massive MIMO communication systems [8]. For simplicity of exposition, let us consider a Uniform Linear Array (ULA) containing N antennas with equidistant spacing ($\lambda/2$, where λ denotes the carrier frequency). Let $\boldsymbol{\theta} = [0, \frac{2\pi}{M}, \frac{4\pi}{M}, \dots, \frac{2(M-1)\pi}{M}]^T$. In a ULA scenario, the $N \times 1$ steering vectors have Vandermonde structure $\mathbf{a}(\theta_i) = [1, e^{-j\theta_i}, e^{-j2\theta_i}, \dots, e^{-j(N-1)\theta_i}]^T$. The phase-only beamformer design problem can be formulated as the ULS

$$\begin{aligned} \min_{\mathbf{w} \in \mathbb{C}^N} \quad & \|\mathbf{y} - \mathbf{A}\mathbf{w}\|_2^2 \\ \text{subject to} \quad & |w_i| = 1, i = 1, \dots, N, \end{aligned} \quad (1)$$

where \mathbf{w} is a prescribed spatial response pattern (see Sec. 3.2),

$$\mathbf{A} = [\mathbf{a}(\theta_1), \dots, \mathbf{a}(\theta_M)]^H, \quad (2)$$

with $(\cdot)^H$ denoting Hermitian (conjugate transpose), \mathbf{w} is the sought beamforming vector, and w_i the i -th element of \mathbf{w} .

2.1. ULS and UQP Preliminaries

Although the beamformer design problem is expressed as ULS in (1), any ULS can be recast as a UQP, and vice versa. Consider the ULS optimization problem in (1), whose cost function can be expressed as

$$\begin{bmatrix} \mathbf{w}^H & 1 \end{bmatrix} \begin{bmatrix} \mathbf{A}^H \mathbf{A} & -\mathbf{A}^H \mathbf{y} \\ -\mathbf{y}^H \mathbf{A} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ 1 \end{bmatrix} \quad (3)$$

where the (constant) $\mathbf{y}^H \mathbf{y}$ term has been discarded. Hence, we may equivalently rewrite (1) as

$$\begin{aligned} \min_{\tilde{\mathbf{w}} \in \mathbb{C}^{N+1}} \quad & \tilde{\mathbf{w}}^H \mathbf{R} \tilde{\mathbf{w}} \\ \text{subject to} \quad & |\tilde{w}_i|^2 = 1, i = 1, \dots, N; \tilde{w}_{N+1} = 1, \end{aligned} \quad (4)$$

where $\tilde{\mathbf{w}} = [\mathbf{w}, 1]^T$ and \mathbf{R} denotes the matrix in the middle of (3). We may in fact relax the last constraint from $\tilde{w}_{N+1} = 1$ to $|\tilde{w}_{N+1}|^2 = 1$ without loss of generality, and thus Problem (4) becomes a standard UQP problem. Note that although \mathbf{R} as defined above may not be positive semi-definite (PSD), we can easily render the cost function convex via diagonal loading without changing the optimization problem. Conversely, any UQP can be cast as a ULS; for PSD \mathbf{R} in (4), we have

$$\begin{aligned} \tilde{\mathbf{w}}^H \mathbf{R} \tilde{\mathbf{w}} &= [\mathbf{w}^H e^{-j\theta} \quad e^{-j\theta}] \begin{bmatrix} \mathbf{A}^H \mathbf{A} & -\mathbf{A}^H \mathbf{y} \\ -\mathbf{y}^H \mathbf{A} & c \end{bmatrix} \begin{bmatrix} \mathbf{w} e^{j\theta} \\ e^{j\theta} \end{bmatrix} \\ &= \mathbf{w}^H \mathbf{A}^H \mathbf{A} \mathbf{w} - \mathbf{y}^H \mathbf{A} \mathbf{w} - \mathbf{w}^H \mathbf{A}^H \mathbf{y} + c. \end{aligned} \quad (5)$$

Setting $c = \mathbf{y}^H \mathbf{y}$, the problem can be expressed as in (1).

2.2. Semidefinite Relaxation for UQP/ULS

To deal with UQP (and thus ULS by the equivalence), the most popular approach is arguably semidefinite relaxation (SDR) [7]. The SDR procedure can be described as follows. Consider the UQP problem

$$\begin{aligned} \min_{\mathbf{w} \in \mathbb{C}^N} \quad & \mathbf{w}^H \mathbf{R} \mathbf{w} \\ \text{subject to} \quad & |w_i|^2 = 1, i = 1, \dots, N, \end{aligned} \quad (6)$$

where \mathbf{R} is positive semi-definite, which ensures that the cost function is convex. This problem is still non-convex due to the element-wise Unit-modulus constraints on \mathbf{w} [10]. Writing $\mathbf{w}^H \mathbf{R} \mathbf{w} = \text{Trace}(\mathbf{w}^H \mathbf{R} \mathbf{w}) = \text{Trace}(\mathbf{R} \mathbf{w} \mathbf{w}^H)$, we can define a matrix $\mathbf{W} := \mathbf{w} \mathbf{w}^H$ and equivalently write (6) as

$$\begin{aligned} \min_{\mathbf{W}} \quad & \text{Trace}(\mathbf{R} \mathbf{W}) \\ \text{subject to} \quad & \mathbf{W}_{ii} = 1, i = 1, \dots, N \\ & \mathbf{W} = \mathbf{w} \mathbf{w}^H. \end{aligned} \quad (7)$$

The constraint $\mathbf{W} = \mathbf{w} \mathbf{w}^H$ is equivalent to $\mathbf{W} \succeq \mathbf{0}$ (positive semi-definite) and $\text{rank}(\mathbf{W}) = 1$. The $\text{rank}(\mathbf{W}) = 1$ constraint is non-convex; dropping it yields a relaxation form that is convex. However, the optimal \mathbf{W}_o of the relaxed problem will (in general) not be rank one. To extract a feasible solution of (6) from \mathbf{W}_o , we follow the randomization procedure in [11]: We first extract the principal component of \mathbf{W}_o ; then, we generate L candidate random vectors via $\boldsymbol{\zeta}_\ell = \mathbf{U} \boldsymbol{\Sigma}^{1/2} \mathbf{v}_\ell$, for $\ell = 1, \dots, L$, where each \mathbf{v}_ℓ is independently chosen from a circularly symmetric zero-mean complex Gaussian distribution of unit variance, with $\mathbf{W}_o = \mathbf{U} \boldsymbol{\Sigma} \mathbf{U}^H$ signifying the eigendecomposition of \mathbf{W}_o . Then each $\boldsymbol{\zeta}_\ell$ is projected onto the feasible set via $\hat{\mathbf{w}}_\ell = e^{j\angle(\boldsymbol{\zeta}_\ell)}$, for $\ell = 1, \dots, L$, where $\angle(\cdot)$ denotes the element-wise angle of the argument, and the cost of (6) is evaluated at this $\hat{\mathbf{w}}_\ell$. Finally, the $\hat{\mathbf{w}}_\ell$ that yields the smallest cost is chosen as the approximate solution.

3. PROPOSED APPROACH

3.1. Gradient Projection for ULS

Although the ULS problem in (1) can be approximated using SDR, SDR is computationally expensive and memory-inefficient. Previous works on phase-only beamforming [4, 9] instead considered an explicit $w_i = e^{j\theta_i}$ parametrization of the unit modulus constraint, and proposed using unconstrained derivative-based methods, such as gradient descent and Newton's method to address the optimization problem. However, such a parametrization awkwardly changes the cost function from a nice quadratic convex function to a non-convex one, and does not seem to work well for this problem. Different from earlier attempts, we propose keeping the unit modulus constraint and using *projected* gradient descent instead of unconstrained gradient descent. The details are provided in Algorithm 1.

Algorithm 1 Gradient Projection

- 1: Initialization: Set $k = 0$, $\alpha = \frac{1}{\lambda_{\max}(\mathbf{A}^H \mathbf{A})}$, $\mathbf{w}_0 = \mathbf{A}^\dagger \mathbf{y}$
 - 2: **Repeat**
 - 3: $\boldsymbol{\zeta}_{k+1} = \mathbf{w}_k + \alpha \mathbf{A}^H (\mathbf{y} - \mathbf{A} \mathbf{w}_k)$ (gradient descent);
 - 4: $\mathbf{w}_{k+1} = e^{j\angle(\boldsymbol{\zeta}_{k+1})}$ (projection);
 - 5: $k = k + 1$;
 - 6: **until convergence**
-

Algorithm 1 is nothing but a GP algorithm – what is special is that the projection step involves a non-convex set – the element-wise unit modulus constraint, in particular. In Algorithm 1, $\lambda_{\max}(\mathbf{X})$ denotes the principal eigenvalue of \mathbf{X} , and α is the step size along the opposite direction of the gradient. The motivation of Algorithm 1 is simple: gradient descent has the advantage of scalability, and is able to explore data sparsity. In addition, it is much more memory-efficient relative to SDR. These traits are well-suited for large-scale problems. In addition, projection onto a unit modulus constraint admits a closed-form solution (cf. line 4 in Algorithm 1), and the entire procedure can be carried out very efficiently. On the other hand, the concern is that projection onto a non-convex set may in fact increase the cost value, and thus tends to be problematic in terms of optimization. Nonetheless, regarding convergence, we have shown that

Proposition 1 For $\alpha \leq \frac{1}{\lambda_{\max}(\mathbf{A}^H \mathbf{A})}$, every limit point of $\{\mathbf{w}_k\}_{k=1,2,\dots}$ produced by Algorithm 1 is a Karush-Kuhn-Tucker (KKT) point of Problem (1).

We relegate the proof to a forthcoming journal version [12] due to space limitations. We remark that the convergence result is of broad interest: Since ULS/UQP has a large variety of applications such as multiuser detection [1], phase retrieval [3], and source localization [2], GP can be applied to these applications when the problem dimension grows large – and convergence to a KKT point can be guaranteed.

3.2. GP with Automatic Scaling for Beam Pattern Design

Algorithm 1 is simple and effective in addressing general ULS/UQP problems, as we will show in Sec. 4. However, in the beamformer design problem, there is a subtle factor that greatly affects the performance, namely, the scaling of \mathbf{y} . To explain, consider the following choice of \mathbf{y} for receive beamformer design:

$$y_i = \begin{cases} 1 & \text{if } i \in \mathcal{J}, \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

where \mathcal{J} denote the set containing the indices i corresponding to the direction(s) of interest in $\boldsymbol{\theta}$. The \mathbf{y} in (8) specifies the beampattern that we want to produce. However, note that $|\mathbf{a}^H(\theta_i) \mathbf{w}| = |\sum_{j=1}^N a_j^*(\theta_i) w_j| \leq \sum_{j=1}^N |a_j^*(\theta_i) w_j| = N$, and, from the Cauchy-Schwartz inequality, the maximum is

achieved if and only if $\mathbf{w} = e^{j\phi} \mathbf{a}(\theta_i)$. Therefore, setting y_i to N seems to be a better choice when one is interested in beamforming to only one angle. In practice, it is more common that multiple angles are of interest, and the ‘optimal scaling’ of \mathbf{y} is unclear under such circumstances. To compensate for this, we can introduce an additional scaling variable $s \in \mathbb{C}$ to obtain

$$\begin{aligned} \min_{\mathbf{w} \in \mathbb{C}^N, s \in \mathbb{C}} \quad & \|\mathbf{y} - s \mathbf{A} \mathbf{w}\|_2^2 \\ \text{subject to} \quad & |w_i| = 1, i = 1, \dots, N. \end{aligned} \quad (9)$$

Variable s can be regarded as an ‘automatic normalization’ factor, which fixes the aforementioned scaling issue. Note that, by separability, we may compute and substitute the optimal s as a function of \mathbf{w} :

$$s_{opt} = \frac{\mathbf{w}^H \mathbf{A}^H \mathbf{y}}{\|\mathbf{A} \mathbf{w}\|_2^2}, \quad (10)$$

Using (10), we can ‘embed’ the tuning of s in each iteration of Algorithm 1, leading to Algorithm 2. This modified algorithm can be considered as an alternating optimization (AO) with respect to (w.r.t.) \mathbf{w} and s . For the subproblem w.r.t. \mathbf{w} , we solve it inexactly by taking a GP step. Since s can be easily computed, Algorithm 2 has only marginal complexity increase relative to Algorithm 1, but can better address the scaling problem in beam pattern design.

Algorithm 2 GP with AO on (\mathbf{w}, s)

- 1: Initialization: Set $k = 0$, $\mathbf{w}_0 = \mathbf{A}^\dagger \mathbf{y}$
 - 2: **Repeat**
 - 3: $s_{k+1} = \mathbf{w}_k^H \mathbf{A}^H \mathbf{y} / \|\mathbf{A} \mathbf{w}_k\|_2^2$;
 - 4: $\alpha_{k+1} = 1 / \lambda_{\max}(|s_{k+1}|^2 \mathbf{A}^H \mathbf{A})$;
 - 5: $\boldsymbol{\zeta}_{k+1} = \mathbf{w}_k + \alpha_{k+1} s_{k+1}^* \mathbf{A}^H (\mathbf{y} - s_{k+1} \mathbf{A} \mathbf{w}_k)$;
 - 6: $\mathbf{w}_{k+1} = e^{j\angle(\boldsymbol{\zeta}_{k+1})}$;
 - 7: $k = k + 1$;
 - 8: **until convergence**
-

3.3. Exploring Additional Degrees of Freedom

At this point it is worth highlighting an important difference between transmit and receive beamforming. In the receive beamforming scenario, we may be required to control the phase response as a function of θ , e.g., for phase coherence or constructive combining of specular multipath components. In transmit (including multicast) beamforming, however, it is sufficient to specify a desired magnitude for each direction or general channel vector of interest, as the receiver will have to perform any necessary phase estimation/correction anyway, due to local oscillator phase mismatch. We can mathematically model this situation by considering the modified optimization problem

$$\begin{aligned} \min_{\mathbf{w} \in \mathbb{C}^N, \boldsymbol{\theta} \in \mathbb{R}^M, s \in \mathbb{C}} \quad & \|\mathbf{y} \circledast e^{j\boldsymbol{\theta}} - s \mathbf{A} \mathbf{w}\|_2^2 \\ \text{subject to} \quad & |w_i| = 1, i = 1, \dots, N, \end{aligned} \quad (11)$$

where \otimes denotes the Hadamard (element-wise) product, and $\boldsymbol{\theta}$ represents the additional degrees of (phase response) freedom. Let $\mathbf{u} = e^{j\boldsymbol{\theta}}$; we can equivalently express (11) as

$$\begin{aligned} \min_{\mathbf{w} \in \mathbb{C}^N, \mathbf{u} \in \mathbb{C}^M, s \in \mathbb{C}} \quad & \|\mathbf{Y}\mathbf{u} - s\mathbf{A}\mathbf{w}\|_2^2 \\ \text{subject to} \quad & |w_i| = 1, i = 1, \dots, N, \\ & |u_i| = 1, i = 1, \dots, M, \end{aligned} \quad (12)$$

where $\mathbf{Y} = \text{Diag}(\mathbf{y})$. Performing alternating optimization on \mathbf{w} , \mathbf{u} and s (projecting \mathbf{w} and \mathbf{u} onto the Unit-modulus space after each iteration), we obtain the following algorithm:

Algorithm 3 GP with AO on $(\mathbf{w}, \mathbf{u}, s)$

- 1: Initialization: Set $k = 0$, obtain initial \mathbf{w}_0 from Algorithm 2, initialize $\mathbf{u}_0 = \boldsymbol{\xi}_0 = \mathbf{1}$, $\beta = 1/\lambda_{\max}(\mathbf{Y}^H \mathbf{Y})$
 - 2: Let $\mathcal{J} = \{i : y_i \neq 0\}$, $\mathbf{Y} = \text{Diag}(\mathbf{y})$, $\tilde{\mathbf{Y}} = \text{Diag}(\mathbf{y}(\mathcal{J}))$
 - 3: **Repeat**
 - 4: $s_{k+1} = \mathbf{w}_k^H \mathbf{A}^H \mathbf{Y} \mathbf{u}_k / \|\mathbf{A} \mathbf{w}_k\|_2^2$;
 - 5: $\alpha_{k+1} = 1/\lambda_{\max}(\|\mathbf{s}_{k+1}\|^2 \mathbf{A}^H \mathbf{A})$;
 - 6: $\boldsymbol{\zeta}_{k+1} = \mathbf{w}_k + \alpha_{k+1} s^* \mathbf{A}^H (\mathbf{Y} \mathbf{u}_k - s \mathbf{A} \mathbf{w}_k)$;
 - 7: $\mathbf{w}_{k+1} = e^{j\angle(\boldsymbol{\zeta}_{k+1})}$;
 - 8: $\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k$;
 - 9: $\boldsymbol{\xi}_{k+1}(\mathcal{J}) = \mathbf{u}_k(\mathcal{J}) - \beta \tilde{\mathbf{Y}}^H (\tilde{\mathbf{Y}} \mathbf{u}_k(\mathcal{J}) - s \mathbf{A}(\mathcal{J}, :) \mathbf{w}_{k+1})$;
 - 10: $\mathbf{u}_{k+1} = e^{j\angle(\boldsymbol{\xi}_{k+1})}$;
 - 11: $k = k + 1$;
 - 12: **until convergence**
-

Note that for the \mathbf{u} update, only the elements corresponding to the non-zero values of \mathbf{y} are of interest. As such, the set \mathcal{J} has been defined as the set of indices where $y_i \neq 0$ and $\mathbf{A}(\mathcal{J}, :)$ denotes a matrix comprised of the rows of \mathbf{A} corresponding to the indices in \mathcal{J} . Thus, only the elements of $\mathbf{u}(\mathcal{J})$ are updated (steps 9 and 10 of the algorithm).

4. SIMULATIONS

We first compare the performance of Algorithm 1 and SDR for solving a general ULS problem. Then, we use simulations to demonstrate the effectiveness of Algorithms 2-3 in the phase-only beamforming problem. In the simulations, the benchmarked algorithm is an alternating optimization-based algorithm for solving the SDR form of UQP [13]. This algorithm exhibits much more favorable runtime performance compared to interior-point based solvers, and thus will be referred to as *FastSDR*. We stop the GP algorithms when $\|\mathbf{w}_k - \mathbf{w}_{k-1}\|_2 / \|\mathbf{w}_k\|_2 < 10^{-6}$. The results are all averaged over 100 independent Monte Carlo trials.

Fig. 1 compares the GP algorithm for general ULS/UQP, i.e., Algorithm 1 with FastSDR, for $N = 2, \dots, 200$. For testing purposes, we draw $\mathbf{y} = \mathbf{A}\mathbf{w} + \mathbf{n}$, where \mathbf{n} is circularly symmetrical zero-mean unit-variance i.i.d. Gaussian noise. Here, we set $M = 144$ and SNR = 10dB, and observe the mean-squared-error (MSE) of the estimated $\hat{\mathbf{w}}$ as performance measure. The Cramér-Rao bound (CRB) is also

presented as a baseline. We see that GP exhibits better MSE performance relative to FastSDR in this simulation, which is rather encouraging. Furthermore, GP is more than 10 times faster than FastSDR for all N considered, see Fig. 2. Note that FastSDR is known to be a surprisingly fast algorithm for approximating UQP, and our results show that GP is even more promising in dealing with ULS/UQP.

Figs. 3-4 present a simulation of phase-only beamforming for a ULA. \mathbf{A} is specially structured (Vandermonde and unit modulus) in this case. We compare the modified versions of GP, i.e., Algorithms 2-3 with FastSDR in this simulation. The angle space is discretized into $M = 36$ regions, and $N = 2, \dots, 200$ antennas are considered. For each problem instance, we randomly draw $K = 2$ angles from $\boldsymbol{\theta}$, and then construct \mathbf{y} in accordance with (8). As shown in Fig. 3, Algorithm 2 performs comparably with FastSDR for $N \leq M$, and increasingly outperforms FastSDR for $N > M$ in terms of least squares cost. Note that it is challenging to incorporate an automatic scaling factor in SDR, and this may explain the performance of FastSDR – it also serves as evidence that adding s to the formulation is much helpful in this special ULS problem. Algorithm 3 yields even slightly lower costs compared to that of Algorithm 2 since it explores additional degrees of freedom. The difference becomes significant when \mathbf{y} contains many non-zeros, as in sector beamforming, because the number of additional degrees of freedom is equal to the number of nonzero elements of \mathbf{y} (since the phase response can only be set where the target magnitude response is non-zero). We also see that the cost values obtained by the proposed algorithms are much closer to the SDR lower bound (which is obtained from the relaxed UQP problem without converting the solution \mathbf{W}_o to a rank-one matrix).

Fig. 4 shows the runtime performance of the algorithms. We see that both Algorithm 2 and Algorithm 3 significantly outperform FastSDR.

5. CONCLUSION

In this paper we considered a simple gradient projection-based algorithmic framework for addressing the ULS / UQP problem, and specialized the algorithm to the application of phase-only beamforming that is motivated by in large-scale radar code design and massive MIMO applications. Many other applications exist, such as phase retrieval, multiuser detection, and sensor network localization, see [1, 2, 10] and references therein. The proposed GP algorithm has been proven to converge to a KKT point of the original NP-hard problem (see the journal version [12]). This is interesting because of the associated projection onto a *non-convex* set, and no analogous convergence result was previously available. The proposed algorithms were carefully compared against state-of-art methods based on SDR, and were found to perform at least as well in terms of least squares cost / mean squared error, and even better in several scenarios at significantly lower runtime complexity.

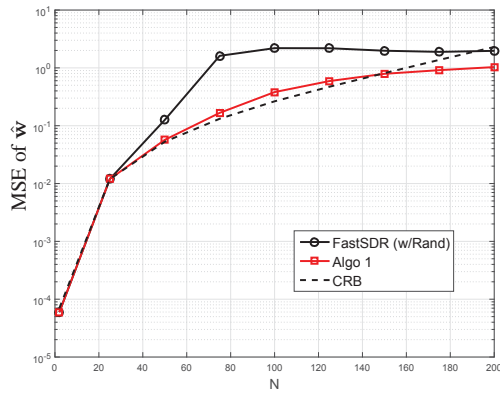


Fig. 1. The MSEs of Algorithm 1 and FastSDR under various N 's; SNR= 10dB.

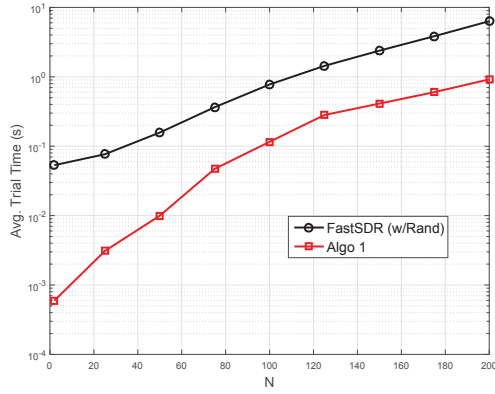


Fig. 2. Run times of Algorithm 1 and FastSDR under various N 's; SNR= 10dB.

REFERENCES

- [1] W.-K. Ma, T. N. Davidson, K. M. Wong, Z.-Q. Luo, and P.-C. Ching, "Quasi-maximum-likelihood multiuser detection using semi-definite relaxation with application to synchronous CDMA," *IEEE Trans. Signal Process.*, vol. 50, no. 4, pp. 912–922, 2002.
- [2] X. Fu, K.-W. Chan, W.-K. Ma, and H.-C. So, "A complex-valued semidefinite relaxation approach for two-dimensional source localization using distance measurements and imperfect receiver positions," in *IEEE ICSP 2012*, 2012, vol. 2, pp. 1491–1494.
- [3] E. J. Candes, T. Strohmer, and V. Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Communications on Pure and Applied Mathematics*, vol. 66, no. 8, pp. 1241–1274, 2013.
- [4] P.A. Thompson, "Adaptation by direct phase-shift adjustment in narrow-band adaptive antenna systems," *IEEE Trans. Antennas Propagat.*, vol. 24, no. 5, pp. 756–760, 1976.
- [5] C.-J. Lu, W.-X. Sheng, Y.-B. Han, and X.-F. Ma, "A novel adaptive phase-only beamforming algorithm based on semidefinite relaxation," in *Proc. IEEE International Symposium on Phased Array Systems Technology*, Oct. 2013, pp. 617–621.
- [6] S. Zhang and Y. Huang, "Complex quadratic optimization and semidefinite programming," *SIAM Journal on Optimization*, vol. 16, no. 3, pp. 871–890, 2006.
- [7] Z.-Q. Luo, W.-K. Ma, A.M.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, May 2010.
- [8] J. Pan and W.-K. Ma, "Constant envelope precoding for single-user large-scale MISO channels: efficient precoding and optimal designs," *IEEE J. Select. Topics Signal Process.*, vol. 8, no. 5, pp. 982–995, 2014.
- [9] S. Smith, "Optimum phase-only adaptive nulling," *IEEE Trans. Signal Process.*, vol. 47, no. 7, pp. 1835–1842, July 1999.
- [10] M. Soltanalian and P. Stoica, "Designing unimodular codes via quadratic optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1221–1234, March 2014.
- [11] N.D. Sidiropoulos and Z.-Q. Luo, "A semidefinite relaxation approach to MIMO detection for high-order QAM constellations," *IEEE Signal Process. Lett.*, vol. 13, no. 9, pp. 525–528, September 2006.
- [12] J. Tranter, N.D. Sidiropoulos, X. Fu, and A. Swami, "Fast unit-modulus least squares with applications in beamforming and phase retrieval," *submitted*, 2016.
- [13] Z. Wen, D. Goldfarb, S. Ma, and K. Scheinberg, "Roy by row methods for semidefinite programming," *Dept. of IEOR, Columbia University, Tech. Rep.*, April 2009.

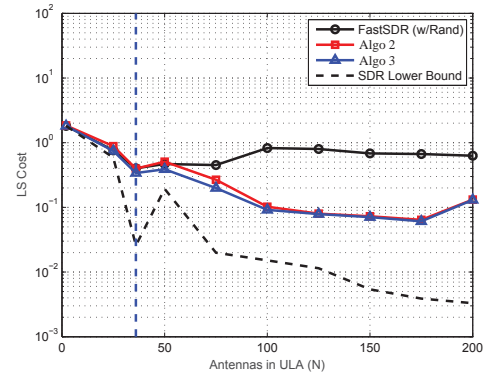


Fig. 3. Least Squares cost for Vandermonde $\mathbf{A} \in \mathbb{C}^{36 \times N}$, $N = 2, \dots, 200$. Vertical line shows the point where $N = M$.

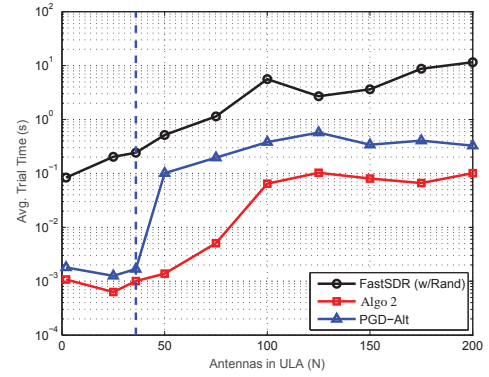


Fig. 4. Run times for Vandermonde $\mathbf{A} \in \mathbb{C}^{36 \times N}$, $N = 2, \dots, 200$. Vertical line shows the point where $N = M$.