

# HEURISTIC METHODS FOR DESIGNING UNIMODULAR CODE SEQUENCES WITH PERFORMANCE GUARANTEES

Shankarachary Ragi<sup>†</sup>      Edwin K. P. Chong<sup>\*</sup>      Hans D. Mittelmann<sup>†</sup>

<sup>†</sup> School of Mathematical and Statistical Sciences  
Arizona State University  
Tempe, AZ 85287-1804, USA  
e-mail: {shankarachary.ragi, mittelmann}@asu.edu  
<sup>\*</sup> Department of Electrical and Computer Engineering  
Colorado State University  
Fort Collins, CO 80523-1373, USA  
e-mail: edwin.chong@colostate.edu

## ABSTRACT

We develop polynomial-time heuristic methods to solve unimodular quadratic programming (UQP) approximately, which is known to be NP-hard. In the UQP framework, we maximize a quadratic function of a vector of complex variables with unit modulus. Several problems in active sensing and wireless communication applications boil down to UQP. With this motivation, we present two new heuristic methods with polynomial complexity to solve the UQP approximately. The first method is called *dominant-eigenvector-matching*; here the solution is picked that matches the complex arguments of the dominant eigenvector of the Hermitian matrix in the UQP formulation. We also provide a performance guarantee for this method. The second heuristic method, a *greedy* strategy, is shown to provide a performance guarantee of  $(1 - 1/e)$  with respect to the optimal objective value given that the objective function possesses a property called *string submodularity*. We also present results from simulations to demonstrate the performance of these heuristic methods.

**Index Terms**— Unimodular codes, unimodular quadratic programming, heuristic methods, radar codes, string submodularity

## 1. INTRODUCTION

Unimodular quadratic programming (UQP) appears naturally in radar waveform-design, wireless communication, and active sensing applications [1]. To state the UQP problem in simple terms- a finite sequence of complex variables with unit modulus to be optimized maximizing a quadratic objective function. In the context of a radar system that transmits a

linearly encoded burst of pulses, the authors of [1] showed that the problems of designing the coefficients (or codes) that maximize the signal-to-noise ratio (SNR) [2] or minimize the Cramer-Rao lower bound (CRLB) lead to a UQP (see [1, 2] for more details). We also know that UQP is NP-hard from the arguments presented in [1, 3] and the references therein. In this study, we focus on developing tractable heuristic methods, to solve the UQP problem approximately that have polynomial complexity with respect to the size of the problem. We also provide performance bounds for these heuristic methods.

In this study, a bold uppercase letter represents a matrix and a bold lowercase letter represents a vector, and if not bold it represents a scalar. Let  $\mathbf{s}$  represent the unimodular code sequence of length  $N$ , where each element of this vector lies on the unit circle  $\Omega$  centered at the origin in the complex plane, i.e.,  $\Omega = \{x \in \mathbb{C}, |x| = 1\}$ . The UQP problem is stated as follows:

$$\underset{\mathbf{s} \in \Omega^N}{\text{maximize}} \quad \mathbf{s}^H \mathbf{R} \mathbf{s}, \quad (1)$$

where  $\mathbf{R} \in \mathbb{C}^{N \times N}$  is a given Hermitian matrix.

There were several attempts at solving the UQP problem (or a variant) approximately or exactly in the past; see references in [1]. For instance, the authors of [4] studied the discrete version of the UQP problem, where the unimodular codes to be optimized are selected from a finite set of points on the complex unit circle around the origin, as opposed to the set of all points that lie on this unit circle in our UQP formulation (as shown in (1)). Under the condition that the Hermitian matrix in this discretized UQP is rank-deficient and the rank behaves like  $\mathcal{O}(1)$  with respect to the dimension of the problem, the authors of [4] proposed a polynomial time algorithm to obtain the optimal solution. Inspired by these efforts, we propose two new heuristic methods to solve the UQP problem (1) approximately, where the computational complexity grows only polynomially with the size of the problem. In our

The work of S. Ragi and H. D. Mittelmann was supported in part by Air Force Office of Scientific Research under grant FA 9550-15-1-0351.

The work of E. K. P. Chong was supported in part by National Science Foundation under grant CCF-1422658.

study, we exploit certain properties of Hermitian matrices to derive performance bounds for these methods.

The rest of the paper is organized as follows. In Section 2, we present a heuristic method called *dominant-eigenvector-matching*. We also present a performance bound for this method in this section. In Section 3, we present a *greedy* strategy to solve the UQP problem approximately, which has polynomial complexity with respect to the size of the problem; we also present a performance bound (when  $\mathbf{R}$  satisfies certain conditions) for this method in this section. We also present application examples where our *greedy* method is guaranteed to provide the above-mentioned performance bound. In Section 4, we present numerical results to demonstrate the effectiveness of the above-mentioned heuristic methods. Section 5 provides a summary of the results and the concluding remarks. We omit the proofs in this paper for lack of space, but we provide key hints to derive these results.

## 2. DOMINANT EIGENVECTOR-MATCHING HEURISTIC

Let  $\lambda_1, \dots, \lambda_N$  be the eigenvalues of  $\mathbf{R}$  such that  $\lambda_1 \leq \dots \leq \lambda_N$ . We can verify that

$$\lambda_1 N \leq \max_{s \in \Omega^N} s^H \mathbf{R} s \leq \lambda_N N.$$

The above upper bound on the optimal solution ( $\lambda_N N$ ) will be used in the following discussions.

**Definition:** In this study, a complex vector  $\mathbf{a}$  is said to be *matching* a complex vector  $\mathbf{b}$  when  $\arg(\mathbf{a}(i)) = \arg(\mathbf{b}(i))$  for all  $i$ , where  $\mathbf{a}(i)$  and  $\mathbf{b}(i)$  are the  $i$ th elements of the vectors  $\mathbf{a}$  and  $\mathbf{b}$  respectively, and  $\arg(x)$  represents the argument of a complex variable  $x$ .

Without loss of generality, we can assume that the matrix  $\mathbf{R}$  is positive semi-definite. If  $\mathbf{R}$  is not positive semi-definite, we can turn it into one with diagonal loading technique without changing the optimal solution to UQP, i.e., we do the following  $\mathbf{R} = \mathbf{R} - \lambda_1 \mathbb{I}_N$ , where  $\lambda_1$  ( $< 0$  as  $\mathbf{R}$  is not semi-definite) is the smallest eigenvalue of  $\mathbf{R}$ . The matrix  $\mathbf{R}$  may be diagonalized as  $\mathbf{R} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^H$ , where  $\mathbf{\Lambda}$  is a diagonal matrix with eigenvalues  $(\lambda_1, \dots, \lambda_N)$  of  $\mathbf{R}$  as the diagonal elements, and  $\mathbf{U}$  is a unitary matrix with the corresponding eigenvectors as its columns. Let  $\mathbf{U} = [e_1 \dots e_N]$ , where  $e_i$  is the eigenvector corresponding to the eigenvalue  $\lambda_i$ . The UQP expression can be written as:  $s^H \mathbf{R} s = s^H \mathbf{U} \mathbf{\Lambda} \mathbf{U}^H s = \sum_{i=1}^N \lambda_i |t(i)|^2$ , where  $t(i)$  is the  $i$ th element of  $\mathbf{U}^H s$ , and  $|\cdot|$  is the modulus of a complex number. We know that  $\sum_{i=1}^N |t(i)|^2 = N$  for all  $s \in \Omega^N$ . Ideally, the UQP objective function would be maximum for an  $s$  such that  $|t(N)|^2 = N$  and  $|t(i)| = 0$  for all  $i < N$ ; but for any given  $\mathbf{R}$  such an  $s$  may not exist such that the above conditions hold true. Therefore, inspired by the above observation, we present the following heuristic method to solve the UQP problem approximately. We choose an  $s \in \Omega^N$  that maximizes the last term in the above summation  $|t(N)|$ . In other

words, we choose an  $s \in \Omega^N$  that “matches” (see the definition presented earlier)  $e_N$ , which is the dominant eigenvector of  $\mathbf{R}$ . But  $e_N$  may contain zero elements, and when this happens we set the corresponding entry in the solution vector to  $e^{j0}$ . Clearly, the above heuristic method has polynomial complexity as most eigenvalue algorithms (to find the dominant eigenvector) have a computational complexity of at most  $\mathcal{O}(N^3)$  [5], e.g., QR algorithm. We call this heuristic method *dominant-eigenvector-matching*. The following proposition provides a performance guarantee for this method. Hereafter, this heuristic method is represented by  $\mathcal{H}$ .

**Proposition 2.1** *Given a Hermitian and positive semi-definite matrix  $\mathbf{R}$ , if  $V_{\mathcal{H}}$  and  $V_{opt}$  represent the objective function values from the heuristic method  $\mathcal{H}$  and the optimal solution respectively for the UQP problem, then*

$$\frac{V_{\mathcal{H}}}{V_{opt}} \geq \frac{\lambda_N + (N-1)\lambda_1}{\lambda_N N},$$

where  $\lambda_1$  and  $\lambda_N$  are the smallest and the largest eigenvalues of  $\mathbf{R}$  of size  $N$ .

This result can be verified using the equalities  $V_{\mathcal{H}} = \mathbf{m}^H \mathbf{R} \mathbf{m} = \sum_{i=1}^N \lambda_i |e_i^H \mathbf{m}|^2$ , where  $\mathbf{m}$  is the solution obtained from  $\mathcal{H}$ .

## 3. GREEDY STRATEGY

In this section, we present a heuristic method with polynomial complexity (with respect to  $N$ ), which is a *greedy* strategy. We also explore the possibility of our objective function possessing a property called string submodularity [6, 7], which allows our *greedy* method to exhibit a performance guarantee. First, we describe the *greedy* method, and then explore the possibility of our objective function being string-submodular. Let  $\mathbf{g}$  represent the solution from this greedy strategy, which is obtained iteratively as follows:

$$\begin{aligned} \mathbf{g}(k+1) &= \arg \max_{x \in \Omega} [\mathbf{g}(1), \dots, \mathbf{g}(k), x]^H \mathbf{R}_k \\ &[\mathbf{g}(1), \dots, \mathbf{g}(k), x], \end{aligned} \quad (2)$$

where  $k = 1, \dots, N-1$  where  $\mathbf{g}(k)$  is the  $k$ th element of  $\mathbf{g}$  with  $\mathbf{g}(1) = 1$ ,  $[a, b]$  in the above expression represents a column vector with elements  $a$  and  $b$ , and  $\mathbf{R}_k$  is the principle sub-matrix of  $\mathbf{R}$  obtained by retaining the first  $k$  rows and the first  $k$  columns of  $\mathbf{R}$ . In other words, we optimize the unimodular sequence element-wise with a partitioned representation of the objective function as shown in (2), which suggests that the computational complexity grows as  $\mathcal{O}(N)$ . Let this heuristic method be represented by  $\mathcal{G}$ .

The greedy method  $\mathcal{G}$  is known to exhibit a performance guarantee of  $(1-1/e)$  when the objective function possesses a property called string-submodularity [6, 7, 8]. To verify if our objective function has this property, we need to re-formulate our problem, which requires certain definitions as described below.

We define a set  $A^*$  that contains all possible unimodular strings (finite sequences) of length up to  $N$ , i.e.,

$$A^* = \{(s_1, \dots, s_k) | s_i \in \Omega \text{ for } i = 1, \dots, k \text{ and } k = 1, \dots, N\},$$

where  $\Omega = \{x \in \mathbb{C}, |x| = 1\}$ . Notice that all the unimodular sequences of length  $N$  in the UQP problem are elements in the set  $A^*$ . For any given Hermitian matrix  $\mathbf{R}$  of size  $N$ , let  $f : A^* \rightarrow \mathbb{R}$  be a quadratic function defined as  $f(A) = A^H \mathbf{R}_k A$ , where  $A = (s_1, \dots, s_k) \in A^*$  for any  $1 \leq k \leq N$ , and  $\mathbf{R}_k$  is the principle sub-matrix of  $\mathbf{R}$  of size  $k \times k$  as defined before. We represent string concatenation by  $\oplus$ , i.e., if  $A = (a_1, \dots, a_k) \in A^*$  and  $B = (b_1, \dots, b_r) \in A^*$  for any  $k + r \leq N$ , then  $A \oplus B = (a_1, \dots, a_k, b_1, \dots, b_r)$ . A string  $B$  is said to be contained in  $A$ , represented by  $B \preceq A$  if there exists a  $D \in A^*$  such that  $A = B \oplus D$ . For any  $A, B \in A^*$  such that  $B \preceq A$ , a function  $f : A^* \rightarrow \mathbb{R}$  is said to be string-submodular [6, 7] if both the following conditions are true:

1.  $f$  is forward monotone, i.e.,  $f(B) \leq f(A)$ .
2.  $f$  has the diminishing-returns property, i.e.,  $f(B \oplus (a)) - f(B) \geq f(A \oplus (a)) - f(A)$  for any  $a \in \Omega$ .

Now, going back to the original UQP problem, the UQP quadratic function may not be a string-submodular function for any given Hermitian matrix  $\mathbf{R}$ . However, without loss of generality, we will show that we can transform the matrix  $\mathbf{R}$  to  $\overline{\mathbf{R}}$  (by manipulating the diagonal entries) such that the resulting quadratic function  $A_k^H \overline{\mathbf{R}}_k A_k$  for any  $1 \leq k \leq N$  and  $A_k \in A^*$  is string-submodular, where  $\overline{\mathbf{R}}_k$  is the principle sub-matrix of  $\overline{\mathbf{R}}$  of size  $k \times k$  as defined before. The following algorithm shows a method to transform  $\mathbf{R}$  to such a  $\overline{\mathbf{R}}$  that induces string-submodularity on the UQP problem.

1. First define  $\delta_1, \dots, \delta_N$  as follows:

$$\delta_k = \sum_{i=1}^{k-1} |r_{ki}|, \quad (3)$$

where  $k = 2, \dots, N$ ,  $\delta_1 = 0$ , and  $|r_{ki}|$  is the modulus of the entry in the  $k$ th row and the  $i$ th column of  $\mathbf{R}$ .

2. Define a vector with  $N$  entries  $(a_1, \dots, a_N)$ , where  $a_k = 2\delta_k + 4 \left( \sum_{i=1}^{N-k} \delta_{k+i} \right)$  for  $k = 1, \dots, N-1$ , and  $a_N = 2\delta_N$ .
3. Define  $\overline{\mathbf{R}}$  as follows:

$$\overline{\mathbf{R}} = \mathbf{R} - \text{Diag}(\mathbf{R}) + \text{diag}((a_1, \dots, a_N)), \quad (4)$$

where  $\text{Diag}(\mathbf{R})$  is a diagonal matrix with diagonal entries same as that of  $\mathbf{R}$  in the same order, and  $\text{diag}((a_1, \dots, a_N))$  is a diagonal matrix with diagonal entries equal to the array  $(a_1, \dots, a_N)$  in the same order.

Since we only manipulate the diagonal entries of  $\mathbf{R}$  to derive  $\overline{\mathbf{R}}$ , the following is true:

$$\arg \max_{A_N \in \Omega^N} A_N^H \overline{\mathbf{R}} A_N = \arg \max_{A_N \in \Omega^N} A_N^H \mathbf{R} A_N.$$

For any given Hermitian matrix  $\mathbf{R}$  and the derived  $\overline{\mathbf{R}}$  (as shown above), let  $F : A^* \rightarrow \mathbb{R}$  be defined as

$$F(A_k) = A_k^H \overline{\mathbf{R}}_k A_k, \quad (5)$$

where  $A_k \in A^*$ .

**Lemma 3.1** For a given  $\mathbf{R}$  and  $F : A^* \rightarrow \mathbb{R}$  as defined in (5) with the derived  $\overline{\mathbf{R}}$  from  $\mathbf{R}$ , and for any  $A, B \in A^*$  such that  $B \preceq A$ , with  $B = (b_1, \dots, b_k)$  and  $A = (b_1, \dots, b_k, \dots, b_l)$  ( $k \leq l \leq N$ ), the inequalities

$$4 \sum_{i=k+1}^l \sum_{j=1}^{N-i} \delta_{i+j} \leq F(A) - F(B) \leq 4 \sum_{i=k+1}^l \left( \delta_i + \sum_{j=1}^{N-i} \delta_{i+j} \right)$$

hold where  $\delta_i$  for  $i = 1, \dots, N$  are defined in (3).

The above result can be proved from equations (4) and (3).

**Lemma 3.2** Given any Hermitian matrix  $\mathbf{R}$  of size  $N$ , the objective function  $F : A^* \rightarrow \mathbb{R}$  defined in (5) is string sub-modular.

The above result can be proved using the results from Lemma 3.1 by showing that  $F$  is forward monotone and has the diminishing returns property. The above lemma shows that the function  $F$  in (5) is string submodular. Therefore, we know from [6, 7] that the performance of the heuristic method  $\mathcal{G}$  is at least  $(1 - 1/e)$  of the optimal value with respect to the function  $F$ , i.e., if  $\mathbf{g} \in A^*$  is the solution from the heuristic method  $\mathcal{G}$  and if  $\mathbf{o}$  is the optimal solution that maximizes the objective function  $F$  as in  $\mathbf{o} = \arg \max_{A_N \in \Omega^N} A_N^H \overline{\mathbf{R}} A_N$ , then

$$F(\mathbf{g}) \geq \left(1 - \frac{1}{e}\right) F(\mathbf{o}). \quad (6)$$

Although we have a performance guarantee for the greedy method with respect to  $F$ , we are more interested in the performance guarantee from this method with respect to the original UQP quadratic function with the given matrix  $\mathbf{R}$ . We explore this idea with the following result.

**Theorem 3.3** For a given Hermitian matrix  $\mathbf{R}$ , if  $\text{Tr}(\overline{\mathbf{R}}) \leq \text{Tr}(\mathbf{R})$  then  $\mathbf{g}^H \mathbf{R} \mathbf{g} \geq (1 - \frac{1}{e}) (\max_{\mathbf{s} \in \Omega^N} \mathbf{s}^H \mathbf{R} \mathbf{s})$ , where  $\mathbf{g}$  is the solution from the greedy method  $\mathcal{G}$ , and  $\overline{\mathbf{R}}$  is derived from  $\mathbf{R}$  as described earlier in this section.

The above result can be derived from (4) and (6).

### 3.1. Application Examples

A square matrix  $\mathbf{R} = [r_{ij}]_{N \times N}$  is said to be  $M$ -dominant if  $|r_{ii}| \geq M \left( \sum_{j=1, j \neq i}^N |r_{ij}| \right); \forall i$ . Using the results in [9], we can verify that if  $\mathbf{R}$  is Hermitian, non-singular, and  $M$ -dominant, then  $\mathbf{R}^{-1}$  is  $\sqrt{M}$ -dominant.

**Proposition 3.4** If a Hermitian matrix  $\mathbf{R}$  of size  $N$  is  $2N$ -dominant, then  $\text{Tr}(\overline{\mathbf{R}}) \leq \text{Tr}(\mathbf{R})$ , where  $\overline{\mathbf{R}}$  is derived from  $\mathbf{R}$  according to (4).

From the above proposition, it is clear that if the Hermitian matrix in the UQP of size  $N$  is  $2N$ -dominant, then the result in Theorem 3.3 holds true, i.e., our *greedy method* provides a performance guarantee of  $(1 - 1/e)$ .

In the case of a monostatic radar that transmits a linearly encoded burst of pulses (as described in [1]), the problem of optimizing the code elements that maximize the SNR boils down to UQP, where  $\text{SNR} = |a|^2 c^H \mathbf{R} c$ ,  $\mathbf{R} = \mathbf{M}^{-1} \odot (\mathbf{p}\mathbf{p}^H)^*$  ( $\odot$  represents the Hadamard product),  $\mathbf{M}$  is an error covariance matrix (size  $N$ ) corresponding to a zero-mean Gaussian vector,  $a$  represents channel propagation and backscattering effects,  $c$  represents the code elements, and  $\mathbf{p}$  is the temporal steering vector. See [2] for a detailed study on this application problem. Using the above arguments, we can verify that if  $\mathbf{M}$  is  $4N^2$ -dominant, which makes  $\mathbf{R}$   $2N$ -dominant, then our *greedy method* for this application is guaranteed to provide the performance of  $(1 - 1/e)$  of that of the optimal.

In the case of a linear array of  $N$  antennas, the problem of estimating the steering vector in adaptive beam-forming boils down to UQP as described in [1] [10], where the objective function is  $c^H \mathbf{R}^{-1} c$ , where  $\mathbf{R}$  is the sample covariance matrix (size  $N$ ), and  $c$  represents the steering vector; see [10] for details on this application problem. Again, we can verify that if the sample covariance matrix is  $4N^2$ -dominant, then our *greedy method* provides a performance guarantee of  $(1 - 1/e)$  (as the result in Theorem 3.3 holds true for this case).

#### 4. SIMULATION RESULTS

We test the performance of the heuristic method  $\mathcal{H}$  numerically for  $N = 20, 50, 100$ . We generate 500 Hermitian and positive semi-definite matrices randomly for each  $N$ , and for each matrix we evaluate  $V_{\mathcal{H}}$  (value from the heuristic method  $\mathcal{H}$ ) and the performance bound derived in Proposition 2.1. To generate a random Hermitian and positive semi-definite matrix, we use the following algorithm: 1) first we generate a random Hermitian matrix  $A$  using the function *rherm*, which is available at [11]; 2) second we replace the eigenvalues of  $A$  with values randomly (uniform distribution) drawn from the interval  $[0, 1000]$ . Figure 1 shows plots of  $\frac{V_{\mathcal{H}}}{\lambda_N N}$  (normalized objective function value) for each  $N$  along with the performance bounds for the heuristic, which also shows  $\frac{V_{\text{rand}}}{\lambda_N N}$ , where  $V_{\text{rand}}$  is the objective function value when the solution is picked randomly from  $\Omega^N$ . The numerical results clearly show that the heuristic method  $\mathcal{H}$  outperforms (by a good margin) random selection, and more importantly the performance of  $\mathcal{H}$  is close to the optimal strategy, which is evident from the simulation results, where the objective function value from  $\mathcal{H}$  is at least 90% (on average) of the upper bound on the optimal value for each  $N$ . The results clearly show that the lower bound is much smaller than the value we obtain from the heuristic method for every sample. In our future study, we will tighten the performance bound for  $\mathcal{H}$  as the results clearly show that there is room for improvement. Fig-

ure 2 shows the normalized objective function value from the greedy method, for each  $N$ , along with the bound  $(1 - 1/e)$ , supporting the result from Theorem 3.3.

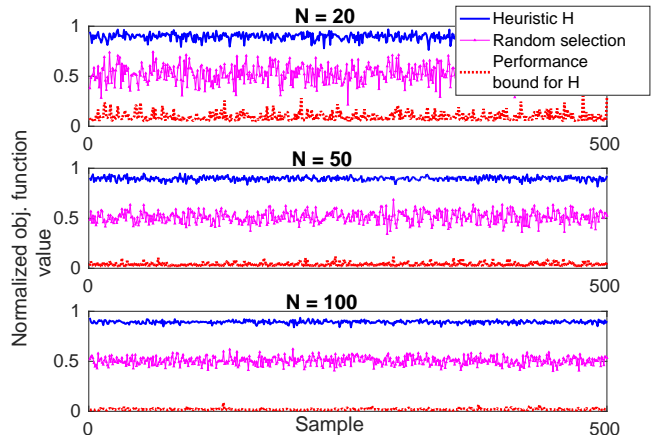


Fig. 1: Performance of  $\mathcal{H}$  for  $N = 20, 50, 100$

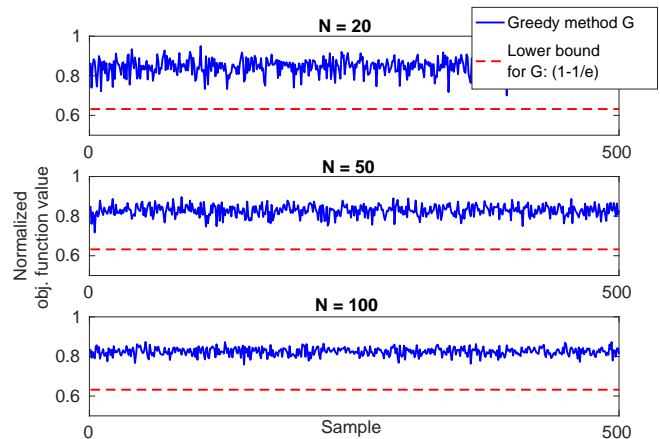


Fig. 2: Performance of  $\mathcal{G}$  for  $N = 20, 50, 100$

#### 5. CONCLUDING REMARKS

We presented two new heuristic methods to solve the UQP problem approximately, both with polynomial complexity with respect to the size of the problem. The first heuristic method was based on the idea of matching the unimodular sequence with the dominant eigenvector of the Hermitian matrix in the UQP formulation. The second heuristic method is a *greedy* strategy. We showed that under loose conditions on the Hermitian matrix, the objective function would possess a property called string submodularity, which then allowed this *greedy* method to provide a performance guarantee of  $(1 - 1/e)$  (a consequence of string-submodularity). Our numerical simulations demonstrated the performance of our heuristic methods. In our future study, we will explore other polynomial-time methods that can provide tighter performance bounds, and also tighten the bounds for the methods presented in this study. We will also study the effect of eigenvalue structure of the matrix  $\mathbf{R}$  on our heuristic methods.

## 6. REFERENCES

- [1] M. Soltanalian and P. Stoica, "Designing unimodular codes via quadratic optimization," *IEEE Trans. Signal Process.*, vol. 62, pp. 1221–1234, 2014.
- [2] A. D. Maio, S. D. Nicola, Y. Huang, S. Zhang, and A. Farina, "Code design to optimize radar detection performance under accuracy and similarity constraints," *IEEE Trans. Signal Process.*, vol. 56, pp. 5618–5629, 2008.
- [3] S. Zhang and Y. Huang, "Complex quadratic optimization and semidefinite programming," *SIAM J. Optim.*, vol. 16, pp. 871–890, 2006.
- [4] A. T. Kyrillidis and G. N. Karystinos, "Rank-deficient quadratic-form maximization over m-phase alphabet: Polynomial-complexity solvability and algorithmic developments," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process. (ICASSP)*, Prague, Czech Republic, 2011, pp. 3856–3859.
- [5] V. Y. Pan and Z. Q. Chen, "The complexity of the matrix eigenproblem," in *STOC '99 Proc. 31st Annu. ACM Symp. Theory of Computing*, Atlanta, GA, 1999, pp. 507–516.
- [6] Z. Zhang, Z. Wang, E. K. P. Chong, A. Pezeshki, and W. Moran, "Near optimality of greedy strategies for string submodular functions with forward and backward curvature constraints," in *Proc. 52nd IEEE Conf. Decision and Control*, Florence, Italy, 2013, pp. 5156–5161.
- [7] Z. Zhang, E. K. P. Chong, A. Pezeshki, and W. Moran, "String submodular functions with curvature constraints," *IEEE Trans. Autom. Control*, vol. 61, pp. 601–616, 2016.
- [8] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set function - I," *Mathematical Programming*, vol. 14, pp. 265–294, 1978.
- [9] A. George and K. D. Ikramov, "Gaussian elimination is stable for the inverse of a diagonally dominant matrix," *Math. Comput.*, vol. 73, pp. 653–657, 2003.
- [10] A. Khabbazibasmenj, S. A. Vorobyov, and A. Hasani, "Robust adaptive beamforming via estimating steering vector based on semidefinite relaxation," in *Proc. Conf. Signals, Syst. Comput. (ASILOMAR)*, Pacific Grove, CA, 2010, pp. 1102–1106.
- [11] Marcus, "Random hermitian matrix generator," <https://www.mathworks.com/matlabcentral/fileexchange/25912-random-hermitian-matrix-generator/content/rherm.m>, 2009.