MERIT: A MONOTONICALLY ERROR-BOUND IMPROVING TECHNIQUE FOR UNIMODULAR QUADRATIC PROGRAMMING

Mojtaba Soltanalian* and Petre Stoica

Dept. of Information Technology, Uppsala University, Uppsala, Sweden

ABSTRACT

The NP-hard problem of optimizing a quadratic form over the unimodular vector set arises in radar code design scenarios as well as other active sensing and communication applications. To tackle this problem, a monotonically error-bound improving technique (MERIT) is proposed to obtain the global optimum or a local optimum of UQP with good sub-optimality guarantees. The provided sub-optimality guarantees are case-dependent and may outperform the \( \pi/4 \) approximation guarantee of semi-definite relaxation.

Index Terms— Code design, radar codes, unimodular codes, quadratic programming, peak-to-average-power ratio (PAR)

1. INTRODUCTION

Unimodular codes are used in many active sensing and communication systems mainly as a result of their optimal (i.e. unity) peak-to-average-power ratio (PAR). In a variety of applications, such as receiver signal-to-noise ratio (SNR) optimization, synthesizing cross ambiguity functions, steering vector estimation, and maximum likelihood (ML) detection, the code approximation can be formulated as the optimization of a quadratic form [1]-[8]. Therefore, we will study the problem

\[
\text{UQP: } \max_{s \in \Omega^n} s^H Rs
\]

where \( R \in \mathbb{C}^{n \times n} \) is a given Hermitian matrix, \( \Omega \) represents the unit circle, i.e. \( \Omega = \{ s \in \mathbb{C} : |s| = 1 \} \) and UQP stands for Unimodular Quadratic Programming.

In [9], the NP-hardness of UQP is proven by employing a reduction from an NP-complete matrix partitioning problem. Studies on polynomial-time (or efficient) algorithms for UQP have been extensive (e.g. see [9]-[22] and the references therein). In particular, the semi-definite relaxation (SDR) technique has been one of the most appealing approaches to the researchers. We refer the interested reader to the survey of the rich literature on SDR in [14].

Throughout the paper, we assume that \( R \) is positive semi-definite (if \( R \) is not positive semi-definite, one can make it so using a diagonal loading without changing the solution to (1)). We present the existing (analytically derived) sub-optimality guarantee for SDR. Let \( v_{\text{SDR}} \) be the expected value of the UQP objective at the obtained randomized solution. Let \( v_{\text{opt}} \) represent the optimal value of the UQP objective. We have

\[
\gamma v_{\text{opt}} \leq v_{\text{SDR}} \leq v_{\text{opt}}
\]

with the sub-optimality guarantee coefficient \( \gamma = \pi/4 \) [9][15]. For the sake of brevity, in the sequel the abbreviation SDR will be used for semidefinite relaxations followed by the randomization procedure.

Besides SDR, the literature does not offer many other numerical approaches to tackle UQP. In this paper, we propose a monotonically error-bound improving technique (called MERIT) that obtains the global optimum or a local optimum of UQP with generally good sub-optimality guarantees. MERIT provides real-time case-dependent sub-optimality guarantees (\( \gamma \)) during its iterations. To the best of our knowledge, such guarantees for UQP were not known prior to this work. Using MERIT one may obtain better performance guarantees compared to the analytical worst-case guarantees (such as \( \gamma = \pi/4 \) for SDR). The provided case-dependent sub-optimality guarantees are of practical importance in decision making scenarios.

2. MERIT

To help the formulation of MERIT, Theorem 1 presents a bijection among the set of matrices leading to the same solution.

**Theorem 1.** Let \( \mathcal{K}(s) \) represent the set of matrices \( R \) for which a given \( s \in \Omega^n \) is the global optimizer of UQP. Then

1. \( \mathcal{K}(s) \) is a convex cone.
2. For any two vectors \( s_1, s_2 \in \Omega^n \), the one-to-one mapping

\[
R \in \mathcal{K}(s_1) \iff R \odot (s_0 s_0^H) \in \mathcal{K}(s_2)
\]

(where \( s_0 = s_1^* \odot s_2 \)) holds among the matrices in \( \mathcal{K}(s_1) \) and \( \mathcal{K}(s_2) \).
sub-optimality guarantee for a solution of UQP based on the above $\mathcal{K}(s)$ approximation.

Using the previous results, we build a sequence of matrices (for which the UQP global optima are known) whose distance from the given matrix $R$ is decreasing. The proposed iterative approach can be used to solve for the global optimum of UQP or at least to obtain a local optimum (with an upper bound on the sub-optimality of the solution). We know from Theorem 2 that if $s$ is a stable point of the UQP associated with $R$ then there exist matrices $Q_s \in \mathcal{C}_s$, $P_s \in \mathcal{C}(V_s)$ and a scalar $\alpha_0 \geq 0$ such that $R + \alpha_0 ss'^H = Q_s + P_s$. The latter equation can be rewritten as

$$R + \alpha_0 ss'^H = (Q_1 + P_1) \odot (ss'^H)$$  \hspace{1cm} (5)$$

where $Q_1 \in \mathcal{C}_1$, $P_1 \in \mathcal{C}(V_1)$. We first consider the case of $\alpha_0 = 0$ which corresponds to the global optimality of $s$. Consider the optimization problem:

$$\min_{s \in \Omega, Q_1 \in \mathcal{C}_1, P_1 \in \mathcal{C}(V_1)} \| R - (Q_1 + P_1) \odot (ss'^H) \|_F$$  \hspace{1cm} (6)$$

Note that, as $\mathcal{C}_1 \odot \mathcal{C}(V_1)$ is a convex cone, the global optimizers $Q_1$ and $P_1$ of (6) for any given $s$ can be easily found. On the other hand, the problem of finding an optimal $s$ for fixed $R_1 = Q_1 + P_1$ is non-convex and hence more difficult to solve globally (see below for details).

In the following, we introduce a suitable diagonal loading of $R$ that is necessary to tackle (6). Next the optimization of the function in (6) is discussed through a separate optimization over the three variables of the problem. The detailed derivations can be found in [1].

- **Diagonal loading of $R$**: Let $\overline{R} = R \odot (ss'^H)^*$. We can compute $Q_1$ and $P_1$ (hence $R_1 = Q_1 + P_1$) for any initialization of $s$. In order to guarantee the positive definiteness of $R_1$, define $\varepsilon_0 \triangleq \| \overline{R} - R_1 \|_F$. Then we diagonally load $R$ with $\lambda > \lambda_0 = -\sigma_{n}(R) + \varepsilon_0$:

$$R \leftarrow R + \lambda I.$$  \hspace{1cm} (7)$$

- **Optimization with respect to $Q_1$**: Let $R_Q = R \odot (ss'^H)^* - P_1$, $H = 1_n \odot R_Q 1_n \odot 1_n$, and

$$Q_1(\rho) \triangleq \rho I_n + (I_n - \frac{1_n \odot 1_n}{n})(R_Q - \rho I_n)(I_n - \frac{1_n \odot 1_n}{n}).$$

Also let $\rho_0$ denote the maximal eigenvalue of $Q_1(0)$ corresponding to an eigenvector other than $1_n \odot 1_n/\sqrt{n}$. Then for fixed $P_1$ and $s$, the optimal solution $Q_1$ to (6) is given by

$$Q_1 = Q_1(\rho_*)$$  \hspace{1cm} (8)$$

where

$$\rho_* = \begin{cases} \frac{H}{\rho_0} & \frac{H}{\rho_0} \geq \rho_0, \\ \rho_0 & \text{otherwise}. \end{cases}$$  \hspace{1cm} (9)$$
Moreover, it can be verified that (17) and observe that \( R \) obtain a global optimum of UQP in such cases. However, it not converge to zero. As a result, the proposed method cannot based on the above results is summarized in Table 1-A. There
tions
local optimum of the problem can be obtained by the itera-
tions introduced in the Appendix to decre ase

\[
P_1(k, l) = \begin{cases} R'_p(k, l) & R'_p(k, l) \geq 0 \text{ or } k = l, \\ 0 & \text{otherwise,} \end{cases}
\]

where \( R'_p = \Re\{R_P\} \).

- Optimization with respect to \( s \): Define \( R_1 = Q_1 + P_1 \)
and observe that \( \|R - R_1 \odot (ss^H)\|_F^2 \) can be rewritten as

\[
\|R - \text{Diag}(s) R_1 \text{Diag}(s^*)\|_F^2 = \text{tr}(R^2 + R_1^2) - 2\Re\{\text{tr}(R \text{Diag}(s) R_1 \text{Diag}(s^*))\}.
\]

Moreover, it can be verified that (17)

\[
\text{tr}(R \text{Diag}(s) R_1 \text{Diag}(s^*)) = s^H(R \odot R_1^*)s.
\]

As \( R \odot R_1^* \) is positive definite, we can employ the power
method-like iterations introduced in the Appendix to decrease
the criterion in (6), i.e. starting from the current \( s = s^{(0)} \), a
local optimum of the problem can be obtained by the iterations

\[
s^{(t+1)} = e^{j \arg((R \odot R_1^*)s^{(t)})}.
\]

Finally, the proposed algorithmic optimization of (6)
based on the above results is summarized in Table 1-A. There
exist examples for which the objective function in (6) does
not converge to zero. As a result, the proposed method cannot
obtain a global optimum of UQP in such cases. However, it
is still possible to obtain a local optimum of UQP for some
\( \alpha_0 > 0 \). To do so, we solve the optimization problem,

\[
\min_{s \in \Omega, Q_1, P_1 \in C(V_1)} \|R' - (Q_1 + P_1) \odot (ss^H)\|_F
\]

with \( R' = R + \alpha_0 s s^H \), for increasing \( \alpha_0 \). It is worth pointing
out that achieving a zero value for the criterion in (15) implies
\( R + \alpha_0 s s^H \in K(s) \). As a result, there exists a non-negative
\( v \in \mathbb{R}^n \) such that \( (R + \alpha_0 s s^H)s = v \odot s \). Consequently,
\( R s = (v - n\alpha_0 I) \odot s \) which implies \( s \) is a stationary point
of the UQP associated with \( R \).

The optimization problem in (15) can be tackled using
the same tools as proposed for (6). In particular, note that
increasing \( \alpha_0 \) decreases (15), see [1]. The obtained solution
\( (s, Q_1, P_1) \) of (6) can be used to initialize the corresponding
variables in (15). In effect, the solution of (15) for any \( \alpha_0 \)
can be used for the initialization of (15) with an increased \( \alpha_0 \).

Based on the above discussion and the fact that small val-
ues of \( \alpha_0 \) are of interest, a bisection approach can be used to
obtain \( \alpha_0 \). The proposed method for obtaining a local opti-
mum of UQP along with the corresponding \( \alpha_0 \) is described in
Table 1-B. Using the proposed algorithm, the task of finding
the minimal \( \alpha_0 \) can be accomplished within a finite number
of steps [1].

### Table 1. The MERIT Algorithm

(A) The case of \( \alpha_0 > 0 \)

**Step 0:** Initialize the variables \( Q_1 \) and \( P_1 \) with \( I \). Let \( s \) be a random
vector in \( \Omega^n \).

**Step 1:** Perform the diagonal loading of \( R \) as in (7) (note that this
diagonal loading is sufficient to keep \( R_1 = Q_1 + P_1 \) always positive definite).

**Step 2:** Obtain the minimum of (6) with respect to \( Q_1 \) as in (8).

**Step 3:** Obtain the minimum of (6) with respect to \( P_1 \) using (11).

**Step 4:** Minimize (6) with respect to \( s \) using (14).

**Step 5:** Goto step 2 until a stop criterion is satisfied, e.g. \( \|R - (Q_1 + P_1) \odot (ss^H)\|_F \leq \epsilon_0 \) (or if the number of iterations exceeded a prede-
fined maximum number).

(B) The case of \( \alpha_0 > 0 \)

**Step 0:** Initialize the variables \( (s, Q_1, P_1) \) using the results obtained by
the optimization of (6) as in Table 1-A.

**Step 1:** Set \( \delta \) (the step size for increasing \( \alpha_0 \) in each iteration). Let \( \delta_0 \), be
the minimal \( \delta \) to be considered and \( \alpha_0 = 0 \).

**Step 2:** Let \( \alpha_0^{\text{opt}} = \alpha_0, \alpha_0^{\text{new}} = \alpha_0 + \delta \) and \( R' = R + \alpha_0^{\text{new}} s s^H \).

**Step 3:** Solve (15) using the steps 2-5 in Table 1-A.

**Step 4:** If \( \|R' - (Q_1 + P_1) \odot (ss^H)\|_F \leq \epsilon_0 \) do:

1. **Step 4-1:** If \( \delta > \delta_0 \), let \( \delta \leftarrow \delta/2 \) and initialize (15) with the
previously obtained variables \( (s, Q_1, P_1) \) for \( \alpha_0 = \alpha_0^{\text{opt}} \). Goto
step 2.

2. **Step 4-2:** If \( \delta < \delta_0 \), stop.

Else, let \( \alpha_0 = \alpha_0^{\text{new}} \) and goto step 2.

### 2.1. Sub-optimality Analysis

In this sub-section, we show how the proposed method can
provide real-time sub-optimality guarantees and bounds
during its iterations. Let \( \alpha_0 = 0 \) (as a result \( R' = R \)) and define

\[
E \triangleq R' - (Q_1 + P_1) \odot (ss^H)
\]

with \( Q_1 \in C_1 \) and \( P_1 \in C(V_1) \). By construction, the
global optimum of the UQP associated with \( Rs = s \). We have
that

\[
\max_{s' \in C_1} s'^H Rs' \leq \max_{s' \in C_1} s'^H R s' + \max_{s' \in C_1} s'^H E s'
\]

\[
\leq \max_{s' \in C_1} s'^H R s' + n\sigma_1(E)
\]

\[
= s^H R_\sigma s + n\sigma_1(E)
\]

Furthermore,

\[
\max_{s' \in C_1} s'^H Rs' \geq \max_{s' \in C_1} s'^H R s' + \min_{s' \in C_1} s'^H E s'
\]

\[
\geq \max_{s' \in C_1} s'^H R s' + n\sigma_1(E)
\]

\[
= s^H R_\sigma s + n\sigma_1(E)
\]

As a result, an upper bound and a lower bound on the objective
function for the global optimum of (6) can be obtained at each
termination. Next, suppose that we have to increase
The provided case-dependent sub-optimality guarantee is thus given by

$$\gamma = \frac{s^H R s}{s^H R_s s} = 1 - \frac{\alpha_0 n^2}{s^H R_s s} = \frac{s^H R s}{s^H R_s s + \alpha_0 n^2}. \quad (20)$$

### 3. NUMERICAL EXAMPLES AND DISCUSSION

In order to examine the performance of the proposed method, several numerical examples will be presented. In all cases, we stopped the iterations when $\|E\|_F \leq 10^{-9}$. Moreover, $R$ is a random positive (semi)definite matrix; see [1] for details. We use the power method-like iterations discussed in Appendix A, and MERIT, as well as the curvilinear search of [18] with Barzilai-Borwein (BB) step size, to solve an UQP (with $n = 10$) based on the same initialization. The resultant UQP objectives along with required times (in sec) versus iteration number are plotted in Fig. 2. It can be observed that the power method-like iterations approximate the UQP solution much faster than the curvilinear search of [18]. On the other hand, both methods are much faster than MERIT. This type of behavior, which is not unexpected, is due to the fact that MERIT is not designed solely for local optimization; indeed, MERIT relies on a considerable over-parametrization in its formulation which is the cost paid for easily derivable sub-optimality guarantees. In general, one may employ the power method-like iterations to obtain a fast approximation of the UQP solution (e.g. by using several initializations), whereas for obtaining sub-optimality guarantees one can resort to MERIT.

Next, we approximate the UQP solutions for 20 full-rank random positive definite matrices of sizes $n \in \{8, 16\}$. Inspired by [20] and [21], we also consider rank-deficient matrices $R$ with rank $d \ll n$. The performance of MERIT for different values of $d$ is shown in Table 2. Note that, in general, the provided sub-optimality guarantees $\gamma$ are considerably larger than $\pi/4$ of SDR. We also employ SDR [22] to solve the same UQPs. In this example, we continue the randomization procedure of SDR until reaching the same UQP objective as for MERIT. A comparison of the computation times of SDR and MERIT can also be found in Table 2.

### A. APPENDIX: POWER METHOD FOR UQP

Let $\{s(t+1)\}_{t=0}^{\infty}$ be a sequence of unimodular codes where $s(t+1)$ is the minimizer of the criterion $\|s(t+1) - Rs(t)\|_2$ for $s(t+1) \in \Omega^n$. The minimizing vector $s(t+1)$ of the latter criterion is simply given by the following power method-like iteration:

$$s(t+1) = e^j \text{arg}(Rs(t)) \quad (21)$$

Note that $\|s(t+1) - Rs(t)\|_2^2 = \text{const} - 2\text{Re}\{s(t+1)^H R s(t)\}$. As a result, $s(t+1)$ is also the maximizer of the criterion $\text{Re}\{s(t+1)^H R s(t)\}$. Moreover, if $s(t+1) \neq s(t)$ we have that

$$s(t+1)^H R s(t+1) - s(t)^H R s(t) \quad (22)$$

$$= (s(t+1) - s(t))^H R \overline{s(t+1) - s(t)} + 2 \text{Re}\{s(t+1)^H R s(t)\} - 2 s(t)^H R s(t) > 0$$

as $\text{Re}\{s(t+1)^H R s(t)\} > s(t)^H R s(t)$. Therefore, the UQP objective is increasing through the power method-like iterations in (21). It can also be shown that the sequence $\{s(t+1)\}_{t=0}^{\infty}$ obtained by (21) converges to a local optimum (or a saddle point) of the UQP; see [1].

### Table 2. Comparison of the performance of MERIT (see Table 1) and SDR [22] when solving the UQP for 20 random positive definite matrices of different sizes $n$ and ranks $d$.

<table>
<thead>
<tr>
<th>n</th>
<th>Rank (d)</th>
<th>#problems for which $\gamma = 1$</th>
<th>Average $\gamma$</th>
<th>Average SDR time</th>
<th>Average MERIT time</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>2</td>
<td>15</td>
<td>0.9841</td>
<td>1.08</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>9</td>
<td>0.9912</td>
<td>0.81</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>15</td>
<td>0.9789</td>
<td>2.08</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>13</td>
<td>0.9773</td>
<td>0.95</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>13</td>
<td>0.9610</td>
<td>0.92</td>
<td></td>
</tr>
</tbody>
</table>
B. REFERENCES


