

Fast Convergence Algorithms for Coherence Optimization of Rank-1 Grassmannian Codebooks

Faisal AKRAM, Imran RASHID, Abdul GHAFOOR, Adil Masood SIDDIQUI

Dept. of Electrical Engineering, College of Signals, National University of Science and Technology (NUST), Pakistan

faisal.akram@mcs.edu.pk, irashid@mcs.edu.pk, abdulghafoor-mcs@nust.edu.pk, dradil@mcs.edu.pk

Submitted July 6, 2018 / Accepted March 4, 2019

Abstract. *Best Complex Antipodal Spherical Codes (BCASC) and Coherence Based Grassmannian Codebooks (CBGC) are state of the art algorithms to produce minimal coherence codebooks, however have high running time (relatively more in BCASC search than CBGC algorithm) which restricts their wider application. BCASC and CBGC algorithms are modified in a novel way by incorporating additional parameters adapted to three categories of codebook overcompleteness to check stagnance of optimization and divergent behaviour to achieve faster convergence. The proposed algorithms are compared with BCASC and CBGC algorithms to achieve similar coherence results in much less time.*

Keywords

Grassmannian codebooks, hybrid metaheuristics, numerical optimization

1. Introduction

A set of incoherent complex vectors forms a frame [1]. Compressed sensing uses a frame with low coherence amongst constituent vectors to achieve unique representation of sparse vectors with reliable recovery [2]. Such frames have numerous applications in signal processing, coding and communications [3]. A Grassmannian codebook is a set of vectors or matrices that span a corresponding Grassmannian manifold [4].

Difference sets are used to form Grassmannian codebooks analytically for certain configurations [5] and if analytical constructions are not possible then construction using numerical search method based on Lloyd's algorithm are used [5]. Sloane et al. [6] performed extensive computations for modest dimensions to find optimal packings in Grassmannian spaces. Sequential smooth optimization based flexible technique is employed in [4] to construct Grassmannian codebooks.

Gram matrix based optimization techniques include constructions using alternating projections in [7], [8] and iterative shrinkage with rank reduction in [9] and alternating

minimization penalty method in [10]. In [11], existing codebooks with low coherence or near orthogonal systems are used to construct sensing matrices. The work in [12] uses additive and multiplicative characters of finite fields to construct similar sensing matrices. In [13], frames with low coherence are constructed using convex optimization whereas in [14], sequential iterative decorrelation by convex optimization is used and in [15] weighted least squares solved by shifted power is used.

In [16], best spherical codes were formed and later in [17], best antipodal spherical codes with low coherence were formed in real domain. In [18], Best Complex Antipodal Spherical Codes (BCASC) are formed by a sequential procedure to optimize coherence, which aims at maximizing the Euclidean distance between complex vectors to reduce coherence. Antipodals of complex vectors are incorporated in the optimization process which increase the processing time. Adapted from [18], in [19], Coherence Based Grassmannian Codebook (CBGC) algorithm, aims to optimize coherence by directly reducing the coherence between complex vectors. The codebooks are produced with similar coherence values to [18] but with reduction in processing time owing to optimization without incorporating antipodals of complex vectors.

It is pertinent to mention that both BCASC and CBGC algorithms differ in goal function and the way in which they converge respectively. It is also worth mentioning that the CBGC algorithm which is an adaptation of BCASC algorithm doesn't always guarantee reduction in time as compared to BCASC algorithm. Moreover, BCASC algorithm has also been shown to achieve dictionary adaptation of sensing matrix [20] whereas the same for CBGC algorithm is to be attempted.

Alternatively, the optimization is performed using metaheuristic algorithms which are adapted for particular problems to achieve desired solution quality in reduced computing time [21]. Metaheuristics (include genetic/evolutionary algorithms, tabu search, simulated annealing, particle swarm optimization, ant colony optimization) do not exert additional demands on an optimization problem and are flexible in adapting to the specifics of a problem to achieve better performance [21].

Hybrid metaheuristics synergistically combine various optimization strategies (a combination of two metaheuristic or of an exact and metaheuristic algorithm) to achieve an optimum solution to the problem in a reduced amount of time [22]. Some hybrid metaheuristic schemes include [23–28].

In [23], vector quantization codebooks are generated using efficient and fast converging particle swarm optimization. In [24], the accelerated converging ant colony optimization process is used for vector quantization codebooks generation. In [25], fish school search algorithm based on swarm clustering algorithm is used for vector quantization codebook generation providing PSNR gains in comparison to the swarm clustering algorithm. In [26], codebooks for power control in distributed relay beamforming network are generated using Lloyd's algorithm and a metaheuristic general search algorithm. In [27], joint precoding and antenna selection is done in multiple-input multiple-output environment while using cross-entropy [28] method to maximize channel capacity.

In this work, BCASC and CBGC algorithms are analyzed and modified to produce rank-1 Grassmannian codebooks by tuning existing parameters and incorporating additional metaheuristics adapted to categories of codebook overcompleteness which reduce running time without compromising coherence.

2. Preliminaries

2.1 Coherence Bounds

Consider a matrix $\mathbf{A} \in \mathbb{C}^{M \times N}$ where $M < N$, its coherence is defined as $\mu(\mathbf{A}) = \max_{i \neq j} \frac{|\langle \mathbf{a}_i, \mathbf{a}_j \rangle|}{\|\mathbf{a}_i\| \|\mathbf{a}_j\|}$, where \mathbf{a}_i denotes the i^{th} column of matrix \mathbf{A} [18]. Lower bounds on achievable coherences for complex matrices [29–37], with their respective configurations (Category I, Category II, Category III) is:

$$\mu_{\text{bd}}(\mathbf{A}) = \begin{cases} \sqrt{\frac{N-M}{M(N-1)}}, & \text{if } N \leq M^2 \text{ [Category I]} \\ \max\left(\sqrt{\frac{1}{M}}, \sqrt{\frac{2N-M^2-M}{(M+1)(N-M)}}, 1 - 2N^{-\frac{1}{M-1}}\right), & \text{if } M^2 < N \leq 2(M^2 - 1) \text{ [Category II]} \\ \max\left(\sqrt{\frac{2N-M^2-M}{(M+1)(N-M)}}, 1 - 2N^{-\frac{1}{M-1}}\right), & \text{if } 2(M^2 - 1) < N \text{ [Category III]}. \end{cases} \quad (1)$$

2.2 BCASC Search Algorithm

BCASC algorithm [18], performs sequential Lagrange based optimization on an initial random matrix \mathbf{A} and maximizes the minimum Euclidean distance between any two vectors of the matrix. Optimization function for low coherence matrices is given as:

$$g(\mathbf{A}, \lambda) = \sum_{n=1}^N \sum_{n \neq l} \|\mathbf{a}_n - \mathbf{a}_l\|^{-(p-2)} + \sum_{n=1}^N \lambda_n (\|\mathbf{a}_n\|^2 - 1) \quad (2)$$

where $\lambda = \{\lambda_n\}_{n=1}^N$ are the Lagrange multipliers with unit radius constraint of spherical codes and $p \in \{2^1, 2^2, \dots, 2^9\}$ is the free parameter.

The sequential optimization process alters the location of points formed by initial random unit norm complex vectors on unit norm sphere under the mutual force of repulsion due to similar charge particles being assumed to be in place at the respective points. The force vector is derived by finding the necessary conditions for a global minimum in (2):

$$\left(\underline{\mathbf{f}}_n \leftarrow \int_{\kappa=0}^{2\pi} \sum_{l \neq n} \frac{\mathbf{a}_n - \mathbf{a}_l e^{i\kappa}}{\|\mathbf{a}_n - \mathbf{a}_l e^{i\kappa}\|^p} d\kappa \right)_{n=1}^N \quad (3)$$

where underlining represents unit normalization. BCASC algorithm involves infinite antipodals of each vector in each iteration. Force vector calculated using K -points approximation

$$\left(\underline{\mathbf{f}}_n \leftarrow \sum_{k=1}^K \sum_{l \neq n} \frac{\mathbf{a}_n - \mathbf{a}_l e^{i2\pi \frac{k}{K}}}{\|\mathbf{a}_n - \mathbf{a}_l e^{i2\pi \frac{k}{K}}\|^p} \right)_{n=1}^N, \quad (4)$$

reduces running time at the cost of slight coherence deviation [18]. After dampened application of the forces, the matrix is updated as: $\mathbf{A}^{(k)} = \mathbf{A}^{(k-1)} + \alpha \mathbf{F}^{(k)}$, where dampening factor α is related to free parameter p and causes coarse to fine adjustments. Optimization process doesn't progress to next value of p , until force vectors converge to codewords $\|\underline{\mathbf{f}}_n^{(k)} - \mathbf{a}_n^{(k)}\| < 10^{-10}$.

2.3 CBGC Algorithm

Adapted from BCASC algorithm [18], CBGC algorithm [19] directly optimizes the coherence of the random complex vectors set. It performs Lagrange based optimization of the objective function,

$$g(\mathbf{A}, \lambda) = \sum_{n \neq l} \left(|\langle \mathbf{a}_n, \mathbf{a}_l \rangle|^2 - \beta^2 \mu_{\text{bd}}^2 \right)^p + \sum_{n=1}^N \lambda_n (\|\mathbf{a}_n\|^2 - 1) \quad (5)$$

where $\beta = 0.5$ improves stability. The force vector is derived by finding the equilibrium conditions for (5) as:

$$\underline{\mathbf{f}}_n^{(k)} = -2 \sum_{n \neq l} \left(|\langle \mathbf{a}_n^{(k)}, \mathbf{a}_l^{(k)} \rangle|^2 - \beta^2 \mu_{\text{bd}}^2 \right)^{p-1} \langle \mathbf{a}_l^{(k)}, \mathbf{a}_n^{(k)} \rangle \cdot \mathbf{a}_l^{(k)} \quad (6)$$

where $\mathbf{a}_l^{(k)}$ represents unit norm vector at index l in k^{th} iteration.

After dampened application of the forces the matrix is updated. Optimization process moves to the next value of p , once condition $\|\mathbf{a}_n^{(k)} - \mathbf{a}_n^{(k-1)}\| < 10^{-10}$ is met.

3. Proposed Algorithms

3.1 Discussion on BCASC Algorithm

BCASC algorithm (adapted from [17]) is a sequential coherence optimization process based on complex vector relocations under the influence of a resultant force vector (calculated using numerical integration involving infinite antipodals (3)). Numerical integration increases complexity and processing time. K -point summation approximation based force vector calculation reduces processing time with coherence values deviation [18]. The algorithm convergence largely depends upon the quality of the initial random seed (orientation of the random complex vectors) and the level of over-completeness of the desired matrix configuration. Moreover, the convergence criteria of fixed point found i.e., $\|\mathbf{f}_n - \mathbf{a}_n\| < 10^{-10}$ is checked in each iteration. Fixed point found criteria aims at converging the force vectors to codewords and thereby attaining an equilibrium. During the course of algorithm progression, the matrix vectors at times reach a situation where force vectors based update of the complex matrix causes back and forth setting of vectors without significantly changing the orientation until next value of free parameter p takes effect. Such optimization stagnance wastes computation time. Furthermore, runtime coherence monitoring during the optimization process reveals a deteriorating coherence behaviour. The existing stopping criterion mostly doesn't stop at a point where worst case coherence is found to be at a desirable value.

3.2 Fast BCASC Algorithm (FBCASC)

BCASC algorithm does achieve its optimization goal but the algorithm progression highlights room for improvement in the convergence rate without compromising the quality of resultant matrix. The proposed FBCASC algorithm aims at modifying the K -point summation BCASC algorithm with the ability to terminate an interim optimization sub-problem once no significant improvement (stagnance in optimization) or worsening of goal function (divergence of optimization) occurs to allow reduction in overall computational time without deteriorating coherence achieved. It includes tuning of the existing parameters and introduction of a metaheuristics layer atop the existing algorithm in accordance with the codebook category to counter stagnance and divergence issues. FBCASC algorithm incorporates codebook category wise adaptive tuned parameters and metaheuristics (Algorithm 1, Line 5) to allow even finer reductions in time without compromising coherence.

Stagnance in coherence is assessed by monitoring inter-iteration coherence deviation compared with a pre-defined tolerance. The coherence of the interim-matrix is calculated after every 100 iterations and is denoted as $\mu^{(g)}$. The optimization process is found to be stagnant if condition $|\mu^{(g)} - \mu^{(g-1)}| < \mu_{\text{stol}}$ is satisfied where μ_{stol} is the stagnance coherence tolerance (Algorithm 1, Line 20) and pre-defined metaheuristic stagnant coherent count θ (initiated to θ_{opt} for a particular value of p) is decremented by a unit (Algorithm 1,

Line 21). Once the stagnant coherence count θ becomes zero, the optimization sub-problem is made to exit for the specific value of free parameter p .

Divergence of coherence is checked by decrementing the pre-defined metaheuristic divergence tolerance ξ (initiated to ξ_{opt} for a particular value of p) by a unit, if coherence value increases from the last observed coherence i.e., $\mu^{(g)} > \mu^{(g-1)}$ (Algorithm 1, Line 26–27). The optimization sub-problem proceeds to the next value of free parameter p once $\xi = 0$. The optimization process is stopped at the iteration where the condition $|\mu^{(g)} - \mu_{\text{bd}}(\mathbf{A})| < 10^{-6}$ is satisfied (Algorithm 1, Line 23–24) and the interim matrix is given as output.

3.3 Adapted Parameter Selection for FBCASC Algorithm

In order to get low coherence matrices in as less time as possible for a given configuration, the parameters error threshold ε , summation points K are tuned and optimum values for metaheuristic stagnant coherence tolerance μ_{stol} , stagnant coherence count θ and divergence tolerance ξ are adapted for each of three categories of codebooks (1).

Algorithm 1 FBCASC

Adaptive settings for configurations $\mathbb{C}^{M \times N}$			
Category I	Category II	Category III	Description
5: $\varepsilon = 10^{-4}$ $K \leftarrow 13$ $\mu_{\text{stol}} \leftarrow 10^{-2}$ $\theta_{\text{opt}} \leftarrow 1$ $\xi_{\text{opt}} \leftarrow 3$	$\varepsilon = 10^{-5}$ $K \leftarrow 18$ $\mu_{\text{stol}} \leftarrow 10^{-2}$ $\theta_{\text{opt}} \leftarrow 4$ $\xi_{\text{opt}} \leftarrow 6$	$\varepsilon = 10^{-5}$ $K \leftarrow 50$ $\mu_{\text{stol}} \leftarrow 10^{-5}$ $\theta_{\text{opt}} \leftarrow 12$ $\xi_{\text{opt}} \leftarrow 5$	▶ Random seed ▶ Dampening factor ▶ Max iterations and free parameter ▶ Coherence bound met and bound for $\mathbf{A} \in \mathbb{C}^{M \times N}$ (1)
6: while $p < p_{\text{max}}$ & $\eta = 0$ do	7: $\gamma \leftarrow 0$,	8: $\xi \leftarrow \xi_{\text{opt}}$, $\theta \leftarrow \theta_{\text{opt}}$,	▶ Fixed-point found ▶ Initialize optimum parameters
9: $i = 1$, $g = 1$, $\mu^{(g)} \leftarrow 0$,	10: while $i \leq i_{\text{max}}$ & $\gamma = 0$ & $\eta = 0$ & $\theta \neq 0$ & $\xi \neq 0$ do	11: for $k=1$ to N do	▶ Indices and coherence initiation
12: $\mathbf{f}_n \leftarrow \sum_{k=1}^K \sum_{l \neq n} \frac{\mathbf{a}_n - \mathbf{a}_l e^{i2\pi \frac{k}{K}}}{\ \mathbf{a}_n - \mathbf{a}_l e^{i2\pi \frac{k}{K}}\ ^p}$	13: end for	14: $\{\mathbf{a}_n\}_{n=1}^N \leftarrow \{\mathbf{a}_n + \alpha \mathbf{f}_n\}_{k=1}^N$	▶ Calculate forces (4) ▶ Normalized force application
15: if all $\ \mathbf{f}_k - \mathbf{a}_k\ < \varepsilon$ then	16: $\gamma \leftarrow 1$	17: end if	▶ Fixed point found
18: if $\text{mod}(i, 100) = 0$ then	19: $\mu^{(g)} = \arg \max_{r \neq l} \frac{ \langle \mathbf{a}_r, \mathbf{a}_l \rangle }{\ \mathbf{a}_r\ \ \mathbf{a}_l\ }$	20: if $ \mu^{(g)} - \mu^{(g-1)} < \mu_{\text{stol}}$ then	▶ Coherence calculation
21: $\theta = \theta - 1$	22: end if	23: if $ \mu^{(g)} - \mu_{\text{bd}}(\mathbf{A}) < 10^{-6}$ then	▶ Reduce stagnant coherence count
24: $\eta \leftarrow 1$	25: end if	26: if $\mu^{(g)} > \mu^{(g-1)}$ then	▶ Coherence bound met
27: $\xi = \xi - 1$	28: end if	29: $g = g + 1$	▶ Reduce divergence tolerance
30: end while	31: $i = i + 1$	32: end while	
33: $p \leftarrow 2p$, $\alpha \leftarrow \frac{\alpha_{\text{init}}}{(p-1)}$	34: end while		▶ Adjust free parameter and dampening factor

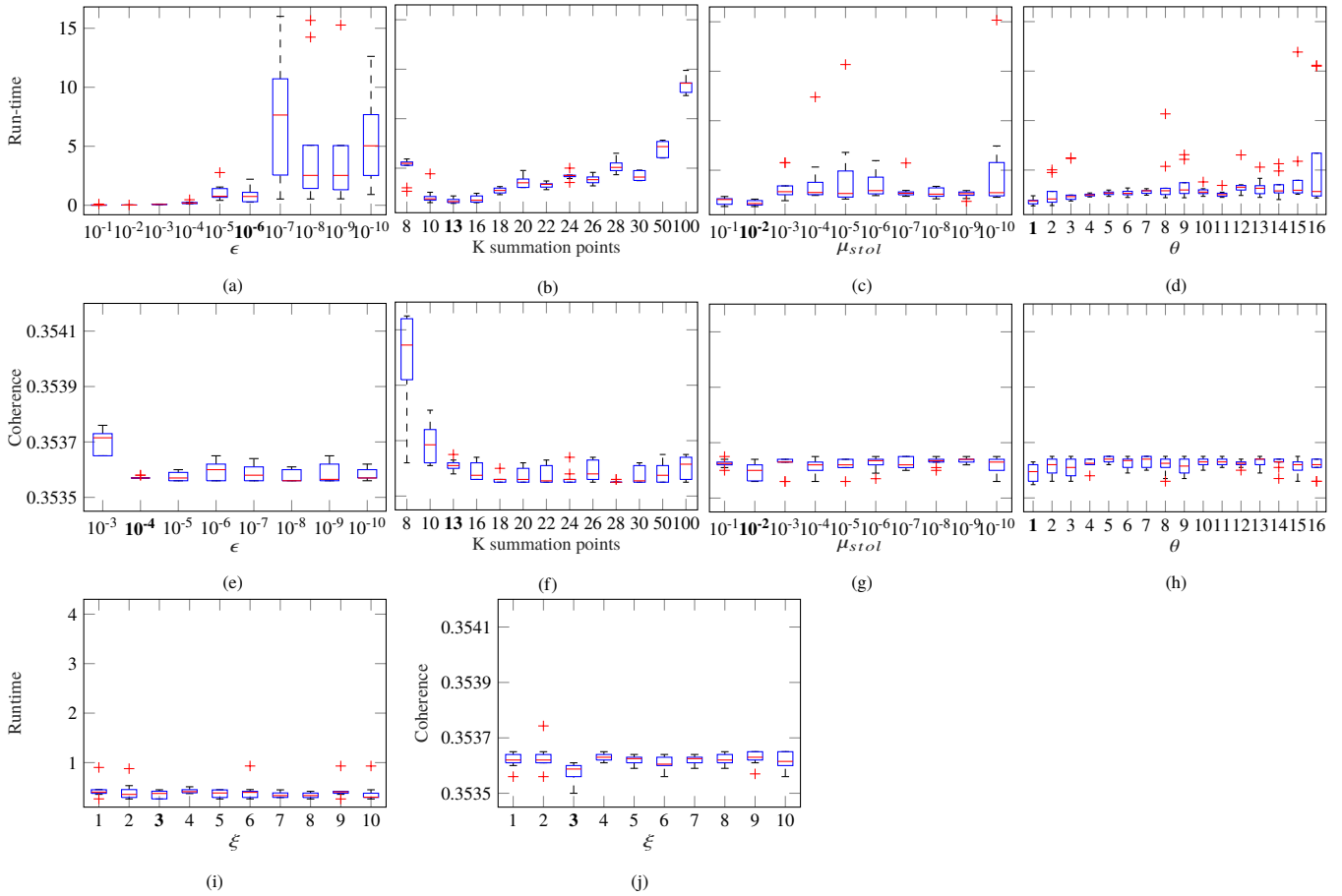


Fig. 1. FBCASC adaptive parameters selection for (4,7) using box plots (box encapsulates 25th to 75th percentile, central mark indicates median, whiskers extend to extreme data points and outliers are plotted using the '+' symbol): (a) and (e) $K = 15$, $\mu_{stol} = 10^{-2}$, $\xi = 5$, $\theta = 2$, (b) and (f) $\varepsilon = 10^{-4}$, $\mu_{stol} = 10^{-2}$, $\xi = 5$, $\theta = 2$, (c) and (g) $\varepsilon = 10^{-4}$, $K = 13$, $\xi = 5$, $\theta = 2$, (d) and (h) $\varepsilon = 10^{-4}$, $K = 13$, $\mu_{stol} = 10^{-2}$, $\xi = 5$, (i) and (j) $\varepsilon = 10^{-4}$, $K = 13$, $\mu_{stol} = 10^{-2}$, $\theta = 1$.

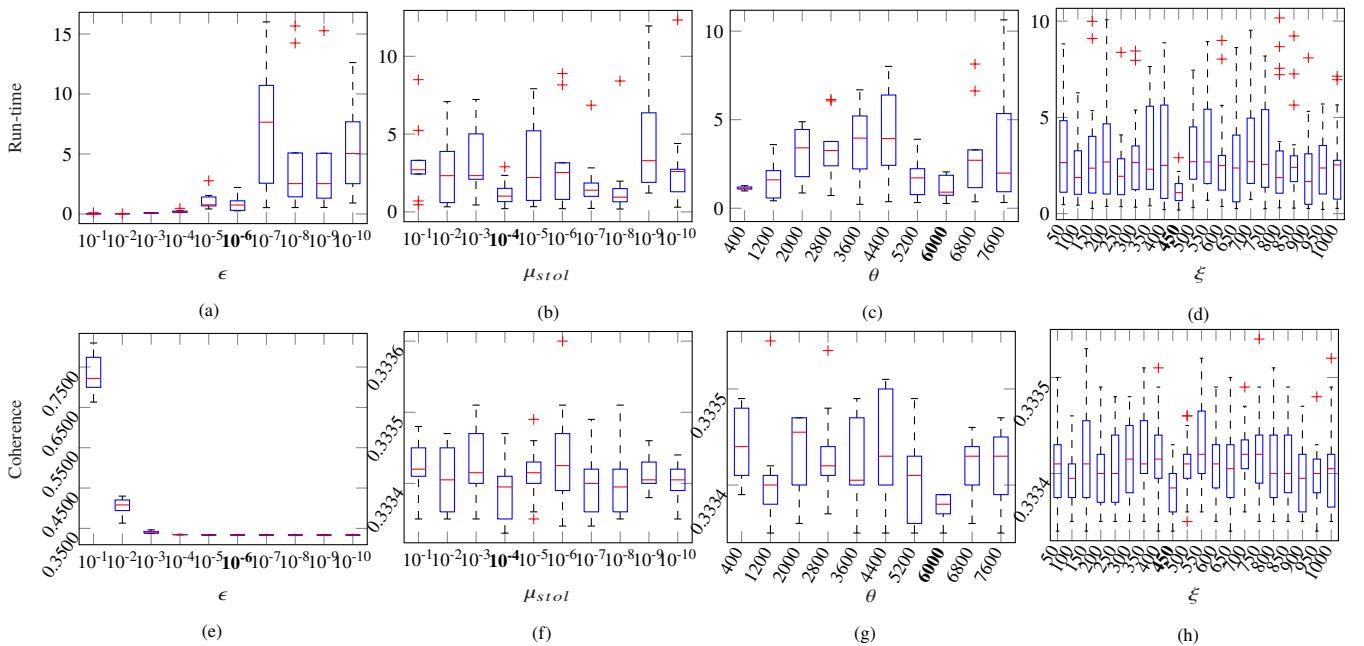


Fig. 2. FCBGC adaptive parameters selection for (5, 10) using box plots (box encapsulates 25th to 75th percentile, central mark indicates median, whiskers extend to extreme data points and outliers are plotted using the '+' symbol): (a) and (e) $\mu_{stol} = 10^{-4}$, $\xi = 100$, $\theta = 6000$, (b) and (f) $\varepsilon = 10^{-6}$, $\xi = 100$, $\theta = 6000$, (c) and (g) $\varepsilon = 10^{-6}$, $\mu_{stol} = 10^{-4}$, $\xi = 100$, (d) and (h) $\varepsilon = 10^{-6}$, $\mu_{stol} = 10^{-4}$, $\theta = 6000$.

Selection of tuned parameters and the metaheuristics for each category is done by considering one model configuration for each category to generalize. The step by step process looks for a suitable value to one existing parameter or a metaheuristic at a time while keeping other parameters constant at a preferable value or the one already found to be appropriate. The procedure assumes that the parameters and the metaheuristics are independent due to their respective nature. In Fig. 1(a) box plots of run-time variation and in Fig. 1(f) box plots of coherence deviation for varying ε error threshold while keeping other parameters constant for ten test trials for FBCASC algorithm configuration (4, 7) are shown. An appropriate value of parameter is chosen while giving equal weight to the lowest coherence achieved and low running time with minimal deviation. Suitable tuned value of $\varepsilon = 10^{-4}$ is chosen on the basis of low run-time as well as minimal deviation coupled with low coherence and low deviation. The remaining of Fig. 1 shows process of finding suitable tuned value for summation points K and optimum values for metaheuristic stagnant coherence tolerance μ_{stol} , stagnant coherence count θ and divergence tolerance ξ using box plots for coherence and runtime variation for ten test trials for FBCASC algorithm configuration (4, 7). Similar process is adopted to find suitable parameters for category II and III configurations (summary given in Algorithm 1 Line 5).

3.4 Discussion on CBGC Algorithm

CBGC algorithm [19] (an adaptation of BCASC algorithm), performs faster than BCASC algorithm due to absence of antipodals. CBGC algorithm convergence criteria i.e., $\|\mathbf{a}_n^k - \mathbf{a}_n^{k-1}\| \leq 10^{-10}$ checks equilibrium being achieved but it doesn't check unnecessary increase in optimization time. Moreover, monitoring of coherence for interim matrix at each iteration has shown that at times the optimization process on one hand doesn't fulfil the convergence criteria and on the other doesn't progress towards the attainment of optimization goal due to overly stringent convergence constraint. Worsening of the optimization process is also observed where coherence tends to increase or diverge. At times, even though the lowest possible coherence is attained with a desirable resolution but stopping criterion does not allow the process to halt.

3.5 Fast CBGC (FCBGC)

CBGC algorithm is broadly successful in its optimization goal but the algorithm progression highlights the need for improvement in the overall convergence rate. In the proposed FCBGC (Algorithm 2, a modification of CBGC), the existing parameters are tuned and a layer of metaheuristics is defined over the algorithm to overcome the issues of stagnant and divergent optimization. Stagnant coherence is overcome by incorporating metaheuristic stagnant coherence tolerance and its respective count. Divergence of coherence is countered by using metaheuristic divergence tolerance decrement. Owing to better convergence of CBGC, compared to BCASC, the additional constraints are made functional in every iteration.

Coherence is calculated at every iteration (Algorithm 2, Line 12) denoted as $\mu^{(i)}$ for the i^{th} iteration. Stagnant coherence is checked using $|\mu^{(i)} - \mu^{(i-1)}| < \mu_{\text{stol}}$ where μ_{stol} is the stagnant coherence tolerance, if found true then stagnance coherence count θ (initiated to θ_{opt} for a particular value of p) is decremented by a unit (Algorithm 2, Line 13–14). Once the stagnant coherence tolerance θ becomes 0, the optimization process progresses to the next value of p .

Coherence divergence is checked by decrementing divergence tolerance ξ (initiated to ξ_{opt} for a particular value of p) by a unit, provided the condition $\mu^{(i)} > \mu^{(i-1)}$ holds true (Algorithm 2, Line 19–20). When the divergence tolerance ξ becomes 0, the optimization process progresses to the next value of p . When $(|\mu^{(k)} - \mu_{\text{bd}}(\mathbf{A})| < 10^{-6})$, the optimization process is stopped and the interim-matrix is given as output (Algorithm 2, Line 16–17).

Algorithm 2 FCBGC

```

1:  $\mathbf{A} \in \mathbb{C}^{M \times N}$  ▷ Random seed
2:  $\alpha_{\text{init}} \leftarrow 0.04, \alpha \leftarrow \alpha_{\text{init}}$  ▷ Dampening factor
3:  $i_{\text{max}} \leftarrow 10^5, p_{\text{max}} \leftarrow 2^9, p \leftarrow 2$  ▷ Max iterations, free parameter
4:  $\eta \leftarrow 0, \mu_{\text{bd}}(\mathbf{A})$  ▷ Coherence bound met and bound for  $\mathbf{A} \in \mathbb{C}^{M \times N}$  (1)

```

Adaptive settings for configurations $\mathbb{C}^{M \times N}$

Category I	Category II	Category III	Description
5: $\varepsilon = 10^{-6}$	$\varepsilon = 10^{-8}$	$\varepsilon = 10^{-8}$	▷ Error threshold
$\mu_{\text{stol}} \leftarrow 10^{-4}$	$\mu_{\text{stol}} \leftarrow 10^{-4}$	$\mu_{\text{stol}} \leftarrow 10^{-5}$	▷ Stagnant coherence tolerance
$\theta_{\text{opt}} \leftarrow 6000$	$\theta_{\text{opt}} \leftarrow 5200$	$\theta_{\text{opt}} \leftarrow 6000$	▷ Stagnant coherence count
$\xi_{\text{opt}} \leftarrow 450$	$\xi_{\text{opt}} \leftarrow 650$	$\xi_{\text{opt}} \leftarrow 100$	▷ Divergence tolerance

```

6: while  $p \leq p_{\text{max}}$  &  $\eta = 0$ , do
7:    $i = 1, \mu^{(0)} \leftarrow 1$  ▷ Initialize index and coherence
8:    $\xi \leftarrow \xi_{\text{opt}}, \theta \leftarrow \theta_{\text{opt}}$  ▷ Initialize optimum parameters
9:   while  $i \leq i_{\text{max}}$  & any  $\|\mathbf{a}_n^{(i)} - \mathbf{a}_n^{(i-1)}\| \geq \varepsilon$  &  $\theta \neq 0$  &  $\xi \neq 0$  &  $\eta = 0$  do
10:    Calculate  $\mathbf{F}^{(i)} = [\mathbf{f}_1^{(i)} \mathbf{f}_2^{(i)} \dots \mathbf{f}_N^{(i)}]$  ▷ Force calculation (6)
11:     $\mathbf{A}^{(i)} = \mathbf{A}^{(i-1)} + \alpha \mathbf{F}^{(i)}$  ▷ Update and normalize A
12:     $\mu^{(i)} = \arg \max_{r \neq t} \frac{|\langle \mathbf{a}_r, \mathbf{a}_t \rangle|}{\|\mathbf{a}_r\| \|\mathbf{a}_t\|}$  ▷ Interim coherence calculation
13:    if  $|\mu^{(i)} - \mu^{(i-1)}| < \mu_{\text{stol}}$  then
14:       $\theta = \theta - 1$  ▷ Reduce stagnant coherence count
15:    end if
16:    if  $|\mu^{(i)} - \mu_{\text{bd}}(\mathbf{A})| < 10^{-6}$  then
17:       $\eta \leftarrow 1$  ▷ Coherence bound met
18:    end if
19:    if  $\mu^{(i)} > \mu^{(i-1)}$  then
20:       $\xi = \xi - 1$  ▷ Reduce divergence tolerance
21:    end if
22:     $i = i + 1$  ▷ Increase iteration
23:  end while
24:   $p \leftarrow 2p, \alpha \leftarrow \frac{\alpha_{\text{init}}}{(p-1)}$  ▷ Adjust free parameter and dampening factor
25: end while

```

3.6 Adapted Parameter Selection for FCBGC Algorithm

The adapted parameters for FCBGC algorithm are selected through a method similar to the one adopted for FBCASC algorithm (as discussed in Sec. 3.3). In Fig. 2(a) box plots of run-time variation and in Fig. 2(e) box plots of coherence deviation for varying ε error threshold while keeping other parameters constant for ten test trials for FCBGC algorithm configuration (5, 10) are shown. Suitable tuned value of $\varepsilon = 10^{-6}$ is chosen on the basis of low run-time as well as minimal deviation coupled with low coherence and low deviation. The remaining of Fig. 2 shows process of

finding suitable values for metaheuristic stagnant coherence tolerance μ_{stol} , stagnant coherence count θ and divergence tolerance ξ using box plots for coherence and runtime variation for ten test trials for FCBGC algorithm configuration (5,10). Similar process is adopted to find suitable tuned parameters and metaheuristics for category II and III configurations (summary given in Algorithm 2 Line 2).

4. Numerical Results

The algorithms are implemented in Matlab R2017a, and the machine used has Intel Core i7-6700 CPU @ 3.40 GHz octa-core 64-bit processor with 32 GB RAM. Minimum and maximum coherences achieved and running times required in ten test runs are compared for BCASC, FBCASC, CBGC and FCBGC algorithms in Tab. 1. BCASC algorithm is implemented using $K = 100$ summation approximation points for producing the results of all configurations considered. The adaptive settings of parameters (FBCASC Algorithm 1 Line 5 and FCBGC Algorithm 2 Line 5) as per the category of the configuration considered are used to produce the codebooks. The results generated are not simulation invariant due to deterministic optimization and variety of random seeds. Significant reduction in runtimes with comparable coherences are achieved using FBCASC and FCBGC algorithms respectively.

In Fig. 3, coherence achieved at each iteration versus number of iterations is plotted for category II configuration (4,20). Although the amount of processing involved in each iteration significantly differs in BCASC and CBGC algorithms and their subsequent modifications, yet in Fig. 3, the coherence optimization profile shows relative trend of convergence for BCASC, FBCASC, CBGC and FCBGC algorithms. In FBCASC and FCBGC algorithms, the stagnant behaviour and divergence is avoided. Significant drop in number of iterations required to achieve similar coherence results in FBCASC and FCBGC algorithms also reduces the processing time.

Gram-matrix ($A^H A$) off-diagonal values distribution for BCASC algorithm, FBCASC algorithm and for |FBCASC-BCASC| for $A \in \mathbb{C}^{4 \times 64}$ are shown in Fig. 4(a), (c) and (e) respectively. Similarly, in Fig. 4(b), (d) and (e), the distribution of Gram-matrix ($A^H A$) off-diagonal values for $A \in \mathbb{C}^{4 \times 64}$ for CBGC algorithm, FCBGC algorithm and for |FCBGC-CBGC| are shown. In Fig. 4 (e) and (f), it is observed that the absolute difference distributions of off-diagonal values for the proposed algorithms and principle algorithms shows minimal variation. This highlights that the matrices generated by the proposed algorithms are similar in quality to the ones generated by the principle algorithms.

Extensive experimentation reveals that some configurations of category I ((2,4), (4,16), (5,25), (6,36)) yield off-diagonal coherence distribution with the variation of ≤ 0.01 (See Fig. 5). However, for the configuration (3,9) the variation is ≤ 0.02 . This also confirms the existence of tight frames for the said configurations as given in [7].

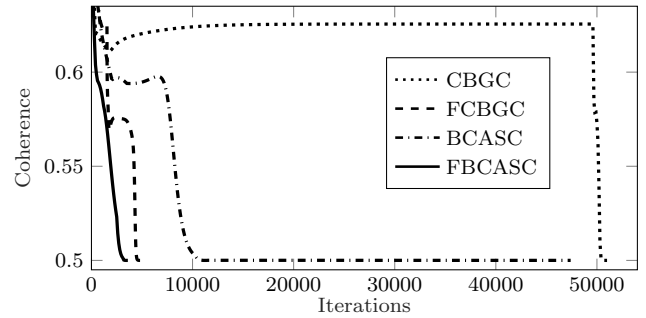


Fig. 3. Coherence optimization profile for (4, 20) configuration.

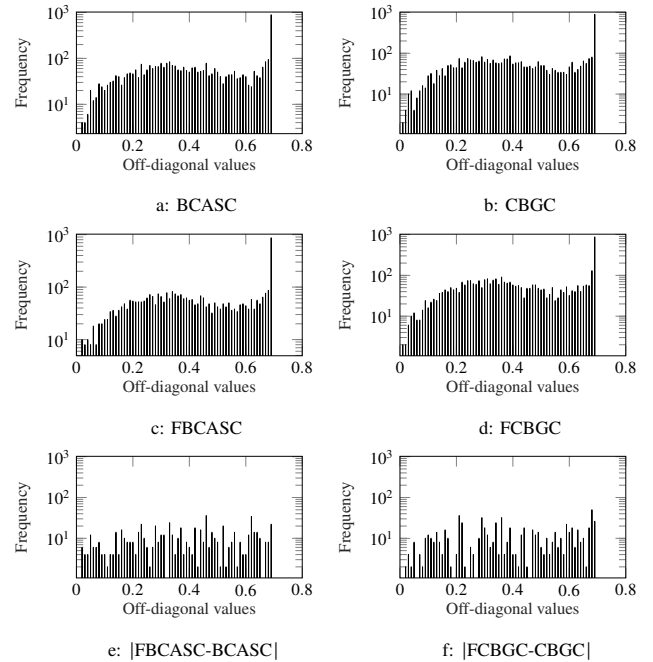


Fig. 4. Off-diagonal values distribution for (4, 64) configuration.

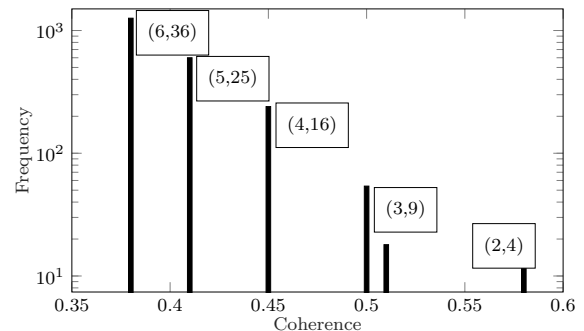


Fig. 5. Combined coherence distribution for $N = M^2$.

5. Conclusion

FBCASC and FCBGC algorithms are proposed to address the issue of processing time in BCASC and CBGC algorithms. Comprehensive comparison of proposed algorithms with the existing algorithms is given for numerous configurations from different categories of overcomplete complex matrices. The proposed algorithms using category-wise adaptive parameters significantly reduce the running time without deteriorating the coherence values.

- [23] TSAI, C., LIN, C., CHIANG, M. et al. A time-efficient particle swarm optimization-based codebook generation algorithm. In *Proceedings of the IEEE Congress on Evolutionary Computation*. Barcelona (Spain), 2010, p. 1–6. DOI: 10.1109/CEC.2010.5585985
- [24] TSAI, C., TSENG, S., YANG, C., et al. PREACO: A fast ant colony optimization for codebook generation. *Applied Soft Computing*, 2013, vol. 13, no. 6, p. 3008–3020. DOI:10.1016/j.asoc.2013.01.017
- [25] FONSECA, C. S., FERREIRA, F.A.B.S., MADEIRO, F. Vector quantization codebook design based on Fish School Search algorithm. *Applied Soft Computing*, 2018, vol. 73, p. 958–968. DOI: 10.1016/j.asoc.2018.09.025
- [26] ZHENG, M., DONG, M. On codebook design for distributed relay beamforming network. In *Proceedings of the IEEE International Conference on Communications in China (ICCC)*. Beijing (China), 2012, p. 323–328. DOI:10.1109/ICCCChina.2012.6356901
- [27] CHEN, J., CHIU, M., YANG, Y., et al. Efficient capacity-based joint quantized precoding and transmit antenna selection using cross-entropy method for multiuser MIMO systems. *International Journal of Antennas and Propagation*, 2012, vol. 2012, Article ID 965834, 7p. DOI 10.1155/2012/965834
- [28] RUBINSTEIN, R. The Cross-Entropy Method for Combinatorial and Continuous Optimization. *Methodology and Computing in Applied Probability*, 1999, vol. 7, no. 1, p. 127–190. DOI:10.1023/A:1010091220143
- [29] WELCH, L. Lower bounds on the maximum cross correlation of signals. *IEEE Transaction on Information Theory*, 1974, vol. 20, no. 3, p. 397–399. DOI: 10.1109/TIT.1974.1055219
- [30] SARWATE, D. V. *Meeting the Welch Bound with Equality*. Chapter in Ding, C., Helleseht, T., Niederreiter, H. *Sequences and their Applications*. Eds. Berlin, Germany: Springer-Verlag, 1999 ISBN: 978-1-4471-0551-0
- [31] DELSARTE, P., GOETHALS, J. M., SEIDEL, J. J. Bounds for systems of lines and Jacobi polynomials. *Philips Research Reports*, 1974, vol. 30, no. 3, p. 91–105. Available at: http://www.extra.research.philips.com/hera/people/aarts/_Philips_Bound_Archive/PRRep/PRRep-30-1975B-091.pdf
- [32] HENKEL, O. Sphere-packing bounds in the Grassmann and Stiefel manifolds. *IEEE Transactions on Information Theory*, 2005, vol. 51, no. 10, p. 3445–3456. DOI: 10.1109/TIT.2005.855594
- [33] PITAVAL, R.A., TIRKKONEN, O., BLOSTEIN, S. Low complexity MIMO precoding codebooks from orthoplex packings. In *Proceedings of the IEEE International Conference on Communication (ICC)*. Kyoto (Japan), 2011, p. 1–5. DOI: 10.1109/icc.2011.5963214
- [34] LEVENSHTAIN, V. I., KABATIANSKY, G. A. On bounds for packings on a sphere and in space. *Problemy Peredachi Informatsii*, 1978, vol. 14, no. 1, p. 3–25.
- [35] LEVENSHTAIN, V. I. Bounds for packings of metric spaces and some of their applications. *Problemy Kibernetiki*, 1983, vol. 40, p. 43–110.
- [36] DING, C., YIN, J. Signal sets from functions with optimum nonlinearity. *IEEE Transactions on Communication*, 2007, vol. 55, no. 5, p. 936–940. DOI: 10.1109/TCOMM.2007.894113
- [37] MUKKAVILLI, K., SABHARWAL, A., ERKIP, E., et al. On beamforming with finite rate feedback in multiple-antenna systems. *IEEE Transactions on Information Theory*, 2003, vol. 49, no. 10, p. 2562–2579. DOI: 10.1109/TIT.2003.817433

About the Authors . . .

Faisal AKRAM received his bachelors degree in Telecommunication Engineering from National University of Science and Technology (NUST), Pakistan, in 2005, the masters degree in Communication Technology from University of Ulm, Germany, in 2013. He is currently a Ph.D. candidate at the NUST, Pakistan. His research interests include compressed sensing, wireless communication, mm-wave hybrid MIMO systems and channel coding.

Imran RASHID received the B.E. degree in Electrical (Telecomm) Engineering from the NUST, Pakistan, in 1999, the M.Sc. degree in Telecomm Engineering (Optical Communication) from D.T.U. Denmark in 2004, and the Ph.D. degree in Mobile Communication from the University of Manchester, U.K., in 2011. He is currently the Chief Instructor at Engineering Wing, MCS, NUST, Pakistan. His research interests are in mobile and wireless communication, MIMO systems, compressed sensing for MIMO OFDM systems, massive MIMO systems, M2M for mobile systems, cognitive radio networks and cyber security.

Abdul GHAFOR obtained his Ph.D. in Control Systems in 2008 from University of Western Australia. Currently, he is Head of Information Security department at National University of Sciences and Technology, Pakistan. His research interests include model and controller reduction, image processing (enhancement, fusion, segmentation and watermarking), cognitive radios and ground penetration radar.

Adil Masood SIDDIQUI received his bachelor degree in Telecommunications Engineering from Military College of Signals, Rawalpindi Pakistan in 1994, masters degree in Electronics and Telecommunication and Ph.D. in Electrical from University of Engineering and Technology, Lahore Pakistan, in 2005 and 2009 respectively. He is on the faculty of Military College of Signals, NUST Since 2009. His research interest includes image registration, de-noising, image enhancement and defogging. He has number of research publication at his credit.