A Parallel Optimization Approach on the Infinity Norm Minimization Problem

Tianyi Liu*, Minh Trinh Hoang*, Yang Yang‡, Marius Pesavento*
* Communication Systems Group, Technische Universität Darmstadt, 64283 Darmstadt, Germany
‡ Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, L-1855 Luxembourg
Email: tliu@nt.tu-darmstadt.de, thminh@nt.tu-darmstadt.de, yang.yang@uni.lu, mpesa@nt.tu-darmstadt.de

***Abstract*—We consider the ℓ_∞-norm minimization problem, which has been investigated in various practical applications. Based on an equivalent problem reformulation we propose an efficient algorithm that is suitable for implementation on parallel hardware architectures. Simulation results show that when applied to the peak-to-average ratio reduction problem, the algorithm achieves the solution obtained by the *primal-dual hybrid gradient approach with proximal operator* while significantly reducing the required running time for convergence.**

***Index Terms*—ℓ_∞-norm minimization, peak-to-average power ratio (PAPR) reduction, successive convex approximation, parallel optimization**

I. Introduction

In this paper, we consider the ℓ_∞-norm minimization problem which consists in minimizing the peak value of a signal representation under the constraint that the representation error is bounded. This problem has recently been of great interest in various practical applications, including peak-to-average power ratio (PAPR) reduction in communication systems [1], vector quantization [2]–[4], approximate nearest neighbor search [5] and peak-force minimization in robotics and control [6]–[8]. PAPR reduction plays an important role in Massive MIMO communication systems where cost efficient power amplifiers are used. In this application, the ℓ_∞-norm minimization is treated as a convex approximation of a non-convex PAPR minimization problem.

The aforementioned problem can be expressed as a standard quadratically constrained quadratic program (QCQP), which can in principle be solved by popular interior-point-method-based solvers (e.g., SDPT3 [9], SeDuMi [10], MOSEK). However, those solvers are often inefficient for the problem sizes faced in large-scale MIMO systems. In the literature, therefore, more competent methods are proposed, which are suitable for large-dimensional optimization problems. In the special case where the variables are real-valued and the tolerance on mismatch is zero, the aforementioned problem can be reformulated into a linear program and solved using the simplex or interior point methods [11]. In [12], Cadzow proposed an algorithm to solve the dual problem of the ℓ_∞ problem, in which the computational complexity is reduced as compared to the linear programming approach. In the case of complex-valued variables, two iterative algorithms based on the primal-dual hybrid gradients with adaptive step-size are proposed in [1]. However, the two mentioned algorithms do not take advantage of the potent multiple parallel processing units typically equipped in large-scale MIMO systems.

Motivated by the aforementioned observations, in this paper we approach the ℓ_∞-norm minimization problem from a different perspective. We show an equivalent problem formulation for this problem, which allows us to adopt the successive convex approximation framework of [13] in an efficient bisection procedure. Our proposed approach takes full benefit of modern parallel hardware architectures and scales well with the number of optimizing variables.

The rest of the paper is organized as follows. The problem formulation is described in Section 2. Our proposed equivalent problem formulation and the solution approach based on a bisection procedure are provided in Section 3. Section 4 considers the parallel solution based on the successive convex approximation approach required in each bisection step. Simulation results are presented in Section V and we conclude in Section VI.

Notation: The symbol $\mathbb{C}$ denotes the set of complex numbers and $\Re\{\cdot\}$ denotes the real part of a complex number. We use x, $\mathbf{x}$ and $\mathbf{X}$ to denote a scalar, vector and matrix, respectively. x_n denotes the n-th element of $\mathbf{x}$. $\|\mathbf{x}\|_p$ denotes ℓ_p-norm of $\mathbf{x}$ and ℓ_∞-norm $\|\mathbf{x}\|_\infty = \max_i |x_i|$, where $|x_i|$ denotes the absolute value of x_i. $\mathbf{X}^\dagger = \mathbf{X}^H(\mathbf{X}\mathbf{X}^H)^{-1}$ stands for the right inverse of $\mathbf{X}$.

II. Problem formulation

Defining the function $g(\mathbf{x}) = \|\mathbf{x}\|_\infty$ the ℓ_∞-norm minimization problem is given by

$$\text{(P-INF}(\varepsilon)\text{)} \qquad \min_{\mathbf{x}\in\mathbb{C}^N} \quad g(\mathbf{x}) \qquad \text{subject to} \quad \|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2 \le \varepsilon,$$

where $\mathbf{H} \in \mathbb{C}^{M\times N}$ is an over-complete matrix (also referred to as frame) of full row-rank with $M < N$, the vector $\mathbf{y} \in \mathbb{C}^M$ is the signal to be represented, the vector $\mathbf{x} \in \mathbb{C}^N$ is the signal representation, and the non-negative parameter ε is the tolerance on representation error. In the case PAPR reduction in Massive MIMO networks, matrix $\mathbf{H}$ is the channel matrix, $\mathbf{x}$ is the signal vector transmitted from the antenna array, and y is the vector containing the information symbols for the different users [14]. In the case $\varepsilon \ge \|\mathbf{y}\|_2$, the all-zeros vector is optimal for problem P-INF(ε) and, hence, practically relevant choices of ε are in the range $0 \le \varepsilon < \|\mathbf{y}\|_2$.

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

Paper [1] analyzes the fundamental properties of the solutions of problem P-INF(ε) using the uncertainty principle (UP) introduced in [2], and then proposes the Convex Reduction of Amplitudes (CRAM) algorithm, in which problem P-INF(ε) is formulated as a saddle-point problem for the corresponding Lagrangian and the adaptive primal-dual hybrid gradient (PDHG) scheme [15] is employed. The proximal operator of the ℓ_∞-norm is used to update the primal variables $\mathbf{x}$. In PDHG the primal variables, the dual variables and the residual variables are sequentially updated and therefore implementation on parallel architectures is not well supported. In the following section we consider an equivalent problem formulation that can be solved in parallel using the successive convex approximation framework of [13].

III. Proposed algorithm for solving problem P-INF(ε)

In the following we introduce an equivalent problem formulation that can be efficiently solved using the successive parallel optimization framework of [13]. Towards this aim we consider the mismatch function $h(\mathbf{x}) = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2$ and introduce the following constrained least squares problem:

$$\begin{aligned} \text{(P-CLS}(\alpha)\text{)} \qquad & \min_{\mathbf{x}\in\mathbb{C}^N} \quad h(\mathbf{x}) \\ & \text{subject to} \quad \|\mathbf{x}\|_\infty \leq \alpha, \end{aligned}$$

which is also convex. Here, the constant α denotes a non-negative bound on the peak value of $\mathbf{x}$. Problem P-CLS(α), on the contrary to P-INF(ε), minimizes the representation error with the peak value of signal representation bounded by a non-negative parameter α.

Let $g^*(\varepsilon)$ and $h^*(\alpha)$ denote the optimal value of problem P-INF(ε) and P-CLS(α), respectively. Consider the problem P-CLS(α) and assume that $\alpha = g^*(\varepsilon)$, i.e., P-CLS($g^*(\varepsilon)$). The following lemma can be proven.

Lemma 1: Any optimal point of P-CLS($g^*(\varepsilon)$) is also optimal for problem P-INF(ε).[1]

Proof: To prove Lemma 1 we first remark that any optimal point $\mathbf{x}_\varepsilon^*$ of problem P-INF(ε), with corresponding function value $g^*(\varepsilon)$, is feasible for problem P-CLS($g^*(\varepsilon)$) as

$$g(\mathbf{x}_\varepsilon^*) = g^*(\varepsilon) = \alpha. \tag{1}$$

Let $\mathbf{x}_\alpha^*$ denote an optimal point of problem P-CLS($g^*(\varepsilon)$), with corresponding function value $h^*(g^*(\varepsilon))$. We conclude that:

$$h(\mathbf{x}_\alpha^*) \leq h(\mathbf{x}_\varepsilon^*) \leq \varepsilon. \tag{2}$$

From (2) we can also observe that $\mathbf{x}_\alpha^*$ is feasible for problem P-INF(ε). Hence,

$$g^*(\varepsilon) \leq g(\mathbf{x}_\alpha^*). \tag{3}$$

Furthermore, we remark that since $\mathbf{x}_\alpha^*$ is feasible for P-CLS($g^*(\varepsilon)$) it follows that $g(\mathbf{x}_\alpha^*) \leq \alpha$. Considering our assumption that $g^*(\varepsilon) = \alpha$ we conclude that (3) is satisfied with equality. Hence $\mathbf{x}_\alpha^*$ is optimal for P-INF(ε).■

[1]Remark: Similar results as in Lemma 1 have been used in literature for different problems, e.g., in the context of single group multicast beamforming [16].

Lemma 2: Any optimal point of P-INF($h^*(\alpha)$) is also optimal for problem P-CLS(α).

Proof: The proof follows the same line of arguments as the proof of Lemma 1 and is therefore omitted.■

From Lemma 1 we conclude that in order to solve problem P-INF(ε) we need to find a parameter α such that $\alpha = g^*(\varepsilon)$. For finding $g^*(\varepsilon)$, the following proposition is useful.

Proposition 1: Finding the optimal value $g^*(\varepsilon)$ of problem P-INF(ε) is equivalent to finding the minimum value of α such that the optimal value of problem P-CLS(α), i.e., $h^*(\alpha)$, is below ε.

Proof: We first remark the following properties of problems P-CLS(α) and P-INF(ε). Given two different values $\alpha_1 > \alpha_2 \geq 0$, the feasible set of problem P-CLS(α_2) is completely contained in the feasible set of problem P-CLS(α_1). Consequently, we have

$$h^*(\alpha_1) \leq h^*(\alpha_2), \quad \forall \alpha_1 > \alpha_2 \geq 0. \tag{4}$$

Therefore, $h^*(\alpha)$ is a non-increasing function in α. Similarly, we can prove for P-INF(ε) that $g^*(\varepsilon)$ is non-increasing in ε. Considering the non-increasing function property in (4) and making use of (2), we observe the following implication:

$$\alpha \geq g^*(\varepsilon) \Rightarrow \quad h^*(\alpha) \leq h^*(g^*(\varepsilon)) = h(\mathbf{x}_\alpha^*) \leq \varepsilon. \tag{5}$$

Similarly, from Lemma 2 and the non-increasing property of $g^*(\varepsilon)$ it follows that

$$h^*(\alpha) \leq \varepsilon \Rightarrow \quad \alpha \geq g^*(\varepsilon). \tag{6}$$

From (5) and (6) we see that the solution set of $h^*(\alpha) \leq \varepsilon$ is left-closed and bounded by $g^*(\varepsilon)$. Hence, Proposition 1 is proved.■

Based on Proposition 1, the bisection method can be employed to find the minimum α since $h^*(\alpha)$ is non-increasing. The continuity of $h^*(\alpha)$ is not required, as our target is not to find the point where $h^*(\alpha)$ is equal to a specific value. For the bisection method, the initial interval $[\alpha_l, \alpha_u]$ must be chosen such that $h^*(\alpha_l) > \varepsilon$ and $h^*(\alpha_u) < \varepsilon$. When $\alpha = 0$, the only feasible solution of problem P-CLS(0) is all-zeros vector and, hence, $h^*(0) = \|\mathbf{y}\|_2 > \varepsilon$. Therefore, a simple choice for α_l is 0. For the right endpoint α_u, it is straightforward to choose α_u such that $h^*(\alpha_u) = 0$, i.e., the optimal solutions are simply the solutions of linear equations $\mathbf{H}\mathbf{x} = \mathbf{y}$. One of them is $\mathbf{x}_{\text{LS}} = \mathbf{H}^\dagger\mathbf{y}$, which is the solution with minimum ℓ_2-norm. The smallest value α_u that includes $\mathbf{x}_{\text{LS}}$ in the feasible set is $\|\mathbf{x}_{\text{LS}}\|_\infty$ and, hence, $h^*(\alpha_u) = 0$. Therefore, the convergence of the bisection procedure is guaranteed. The details of the proposed bisection-based algorithm are presented in Algorithm 1. The task of solving problem P-CLS(α) within each iteration of the bisection procedure is done by an efficient parallel iterative algorithm, which is presented in Section IV and detailed in Algorithm 2.

IV. Iterative algorithm for solving problem P-CLS(α)

Problem P-CLS(α) can be reformulated as:

$$\min_{\mathbf{x}\in\mathbb{C}^N} \quad f(\mathbf{x}) = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 \tag{7a}$$

$$\text{subject to} \quad |x_n|^2 \leq \alpha^2, \quad \forall n = 1, ..., N, \tag{7b}$$

Algorithm 1: The proposed algorithm for problem P-INF(ε)

Input: $\mathbf{y} \in \mathbb{C}^M, \mathbf{H} \in \mathbb{C}^{M\times N}, \varepsilon \geq 0, \eta \geq 0$
Initialize $\mathbf{x}_l \leftarrow \mathbf{0}, \mathbf{x}_u \leftarrow \mathbf{x}_{\text{LS}}, \alpha_l \leftarrow 0, \alpha_u \leftarrow \|\mathbf{x}_u\|_\infty$;
while $\alpha_u - \alpha_l > \eta$ **do**
  Set $\alpha \leftarrow (\alpha_l + \alpha_u)/2$, and compute optimal value $h^*(\alpha)$ and solution $\mathbf{x}^*_\alpha$ of problem P-CLS(α) using Algorithm 2 with initial guess $\mathbf{x}_l$;
  if $h^*(\alpha) > \varepsilon$ **then**
    $\alpha_l \leftarrow \alpha, \mathbf{x}_l \leftarrow \mathbf{x}^*_\alpha$;
  else
    $\alpha_u \leftarrow \alpha, \mathbf{x}_u \leftarrow \mathbf{x}^*_\alpha$;
  end
end
return $\alpha_u, \mathbf{x}_u$

Algorithm 2: The successive convex approximation algorithm for problem P-CLS(α)

Input: $\mathbf{x}^0 \in \mathcal{X}, \mathbf{y} \in \mathbb{C}^M, \mathbf{H} \in \mathbb{C}^{M\times N}, \alpha, t \leftarrow 0$
while *not converged* **do**
  Compute $\hat{\mathbf{x}}^t$ using (12);
  Compute γ^t by the exact line search (14);
  Update $\mathbf{x}^t$ according to (10) and set $t \leftarrow t+1$;
end
return $\mathbf{x}^t$

which has the same formulation as the symbol detection problem investigated in [17] in the context of large-scale MIMO systems. In [17] an efficient parallel iterative algorithm is proposed for large-scale systems based on the successive convex approximation framework described in [13]. We briefly describe the algorithm in this section.

Similar to other gradient methods for convex problems [18], this algorithm starts from an feasible solution $\mathbf{x}^0$ (an initial guess) of problem (7) and iteratively updates the solution based on a computed descent direction and a selected step-size. In each iteration, the descent direction is obtained by solving an approximate convex problem and the step-size is computed using the exact line search method [18]. With the approximate problem and step-size presented in the following, the convergence to the optimal solution of problem (7) is guaranteed, which is explained in [13]. In the following, we first formulate the approximate problem. Then, the approximate problem is decomposed into N independent subproblems, which can be solved in parallel with closed form solutions. Computation of the optimal step-size, as required for the update phase, is presented afterwards.

The Approximate Problem: The Jacobi algorithm decouples the variables in the objective function, so that the problem can be decomposed. Let $\mathbf{x}^t$ denote the starting point in the t-th iteration. Employing the Jacobi algorithm, the approximate function of $f(\mathbf{x})$ in the t-th iteration around point $\mathbf{x}^t$ can be expressed as

$$\sum_{n=1}^{N} f(x_n, \mathbf{x}^t_{-n}) = \sum_{n=1}^{N} \|\mathbf{y} - \mathbf{H}_{-n}\mathbf{x}^t_{-n} - \mathbf{h}_n x_n\|_2^2, \quad (8)$$

where $\mathbf{H}_{-n}$ denotes the matrix obtained by eliminating the n-th column $\mathbf{h}_n$ from matrix $\mathbf{H}$, and $\mathbf{x}_{-n}$ denotes the vector containing all elements of $\mathbf{x}$ except x_n.

Defining $\mathbf{y}^t_n = \mathbf{y} - \mathbf{H}_{-n}\mathbf{x}^t_{-n}$ and $\mathcal{X}$ as the feasible set of problem (7), the approximate problem can be expressed as

$$\hat{\mathbf{x}}^t = \arg\min_{\mathbf{x}\in\mathcal{X}} \sum_{n=1}^{N} \|\mathbf{y}^t_n - \mathbf{h}_n x_n\|_2^2. \quad (9)$$

The vector $\hat{\mathbf{x}}^t - \mathbf{x}^t$ represents a descent direction of the objective function $f(\mathbf{x})$ in the domain of problem (7). Thus, the vector $\mathbf{x}^t$ is updated after each iteration, using the rule:

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \gamma^t(\hat{\mathbf{x}}^t - \mathbf{x}^t), \quad (10)$$

where $\gamma^t \in (0, 1]$ is a step-size. The algorithm has converged to the optimal solution of problem (7), when $\hat{\mathbf{x}}^t = \mathbf{x}^t$.

Decomposition of the Approximate Problem: The variables x_n are decoupled in both the objective function and the constraints in problem (9). Consequently, problem (9) can be decomposed into to N independent subproblems as

$$\hat{x}_n = \arg\min_{x_n} \|\mathbf{y}^t_n - \mathbf{h}_n x_n\|_2^2 \quad (11a)$$

$$\text{subject to } |x_n|^2 \leq \alpha^2, \quad (11b)$$

where $\hat{x}_n$ is the n-th element of vector $\hat{\mathbf{x}}^t$. Since each subproblem exclusively depends on a single variable, each subproblem can be solved independently and concurrently.

We can apply the Karush-Kuhn-Tucker (KKT) optimality conditions to solve each convex subproblem [19]. Then, a closed form expression for the optimal solution $\hat{x}_n$ of each convex subproblem can be derived as

$$\hat{x}_n = \min\left(\frac{|\mathbf{h}_n^H \mathbf{y}^t_n|}{\|\mathbf{h}_n\|^2}, \alpha\right)\left(\frac{\mathbf{h}_n^H \mathbf{y}^t_n}{|\mathbf{h}_n^H \mathbf{y}^t_n|}\right). \quad (12)$$

We recommend paper [17] for the detailed derivation of this closed form expression.

Optimal Step-size: The optimal update step-size γ^t is computed using the exact line search method, which searches on the line segment between $\mathbf{x}^t$ and $\hat{\mathbf{x}}^t$ for the solution that minimizes original objective function $f(\mathbf{x})$. Therefore, the computation of the optimal step-size is formulated by the following convex problem:

$$\gamma^t = \arg\min_{0\leq\gamma\leq 1} \underbrace{\|\mathbf{y} - \mathbf{H}\left(\hat{\mathbf{x}} + \gamma(\hat{\mathbf{x}}^t - \mathbf{x}^t)\right)\|^2}_{Z(\gamma)}. \quad (13)$$

Suppose the gradient of $Z(\gamma)$ with respect to γ vanishes at γ^*, then γ^t is the projection of γ^* onto the interval $[0, 1]$. Consequently, γ^t can be computed as

$$\gamma^t = \left[\frac{\Re\{\mathbf{p}^H\mathbf{q}\}}{\mathbf{q}^H\mathbf{q}}\right]_0^1, \quad (14)$$

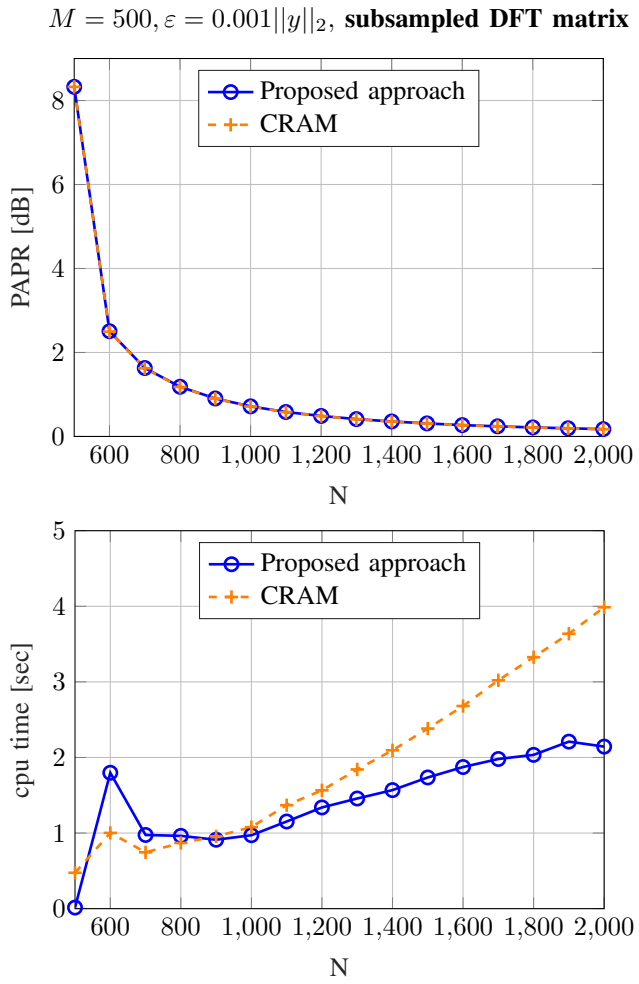

Fig. 1. Simulation results with sub-sampled DFT matrix

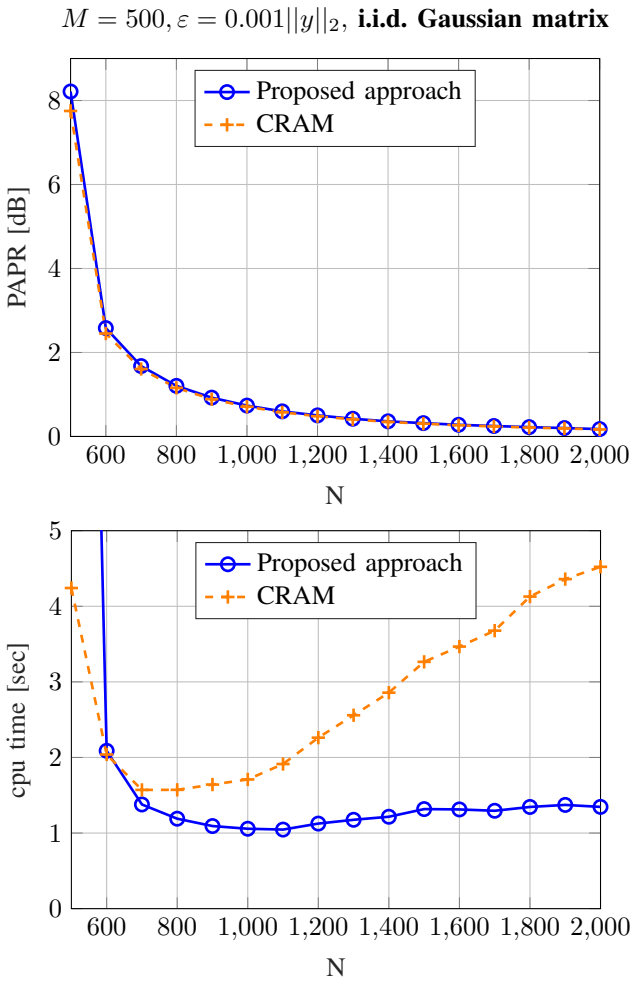

Fig. 2. Simulation results with i.i.d. complex Gaussian matrix

where $\mathbf{p} = \mathbf{y} - \mathbf{H}\mathbf{x}^t$, $\mathbf{q} = \mathbf{H}(\hat{\mathbf{x}}^t - \mathbf{x}^t)$, and $[\cdot]_0^1$ denotes the projection operation onto the interval $[0, 1]$.

Finite Termination: For an intermediate value of α, the exact solution of problem P-CLS(α) is not required by Algorithm 1. The objective is only to verify whether the optimal value $h^*(\alpha)$ is below the error tolerance ε. Therefore, in the case where the function value of the approximate solution $h(\mathbf{x}^t)$ has already reduced below ε, Algorithm 2 can be terminated regardless of the optimality gap, which is advantageous compared to other dual approaches in which each dual problem must be solved exactly to update the dual variable.

V. Simulation Results

In this section, we compare the performance of proposed bisection-based parallel optimization approach with that of CRAM algorithm[2]. We remark that since the CRAM algorithm does not fully support implementation on parallel computing architectures, when comparing the running time of both algorithms, we implemented our proposed parallel algorithm based on only sequential update. Thus, the reported results are based on a serial implementation in MATLAB. Additional gains can be expected when considering fully parallel implementation on corresponding architectures. All experiments are performed on a desktop computer with an AMD Ryzen 7 2700X eight-core 3.7 GHz CPU and 32 GB RAM. The convergence of the CRAM algorithm to the optimal solution of problem P-INF(ε) is proved by theoretical analysis and simulation results in [1]. In most of the applications, the original objective is to minimize the dynamic range. Hence, we compare the PAPR of the solutions returned by the algorithms and the required convergence time for two classes of frames, respectively. Fig. 1 shows the simulation results for frames whose rows are sub-sampled discrete Fourier transform (DFT) bases of dimension N and Fig. 2 for complex i.i.d. Gaussian frames. The PAPR of a nonzero vector $\mathbf{x} \in \mathbb{C}^N$ is defined as

$$\text{PAPR}(\mathbf{x}) = \frac{N\|\mathbf{x}\|_\infty^2}{\|\mathbf{x}\|_2^2}. \tag{15}$$

We fix $M = 500$ and vary N from 500 to 2000. For each dimension pair (M, N), we perform 100 Monte-Carlo trials, and for each trial we generate a frame $\mathbf{H}$ from each frame class

[2]An implementation of CRAM in MATLAB is provided on http://vip.ece.cornell.edu/software_papr.html

specified above. We then generate a complex i.i.d. zero-mean Gaussian vector $\mathbf{y}$ and the tolerance $\varepsilon = 0.001\|\mathbf{y}\|_2$.

For both two classes of frames, we can observe that the two algorithms converge to solutions with equivalent quality in terms of PAPR. In most applications, the region of high redundancy $r = N/M$ is of great importance as here the optimal solution of problem P-INF(ε) has low PAPR. In the region of high redundancy, our proposed algorithm significantly reduces the required running time for convergence .

VI. Conclusion

In this paper, we show an equivalent problem formulation for the ℓ_∞-norm minimization problem, which can be solved in parallel using the successive convex approximation framework in a bisection procedure. Simulation results show the convergence and efficiency of our proposed algorithm.

For future work, in order to further accelerate the bisection procedure, restricting the initial interval is of great interest and thus requires further investigation.

Acknowledgments

This work was supported by the EXPRESS project within the DFG priority program CoSIP (DFG-SPP 1798).

References

[1] C. Studer, T. Goldstein, W. Yin, and R. G. Baraniuk, "Democratic representations," *CoRR*, vol. abs/1401.3420, 2014. [Online]. Available: http://arxiv.org/abs/1401.3420

[2] Y. Lyubarskii and R. Vershynin, "Uncertainty principles and vector quantization," *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3491–3501, July 2010.

[3] M. Puschel and J. Kovacevic, "Real, tight frames with maximal robustness to erasures," in *Data Compression Conference*, March 2005, pp. 63–72.

[4] C. Novak, C. Studer, A. Burg, and G. Matz, "The effect of unreliable llr storage on the performance of mimo-bicm," in *2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*, Nov 2010, pp. 736–740.

[5] H. Jégou, T. Furon, and J. Fuchs, "Anti-sparse coding for approximate nearest neighbor search," *CoRR*, vol. abs/1110.3767, 2011. [Online]. Available: http://arxiv.org/abs/1110.3767

[6] J. Cadzow, "Algorithm for the minimum-effort problem," *IEEE Transactions on Automatic Control*, vol. 16, no. 1, pp. 60–63, February 1971.

[7] A. S. Deo and I. D. Walker, "Minimum effort inverse kinematics for redundant manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 5, pp. 767–775, Oct 1997.

[8] Y. Zhang, J. Wang, and Y. Xu, "A dual neural network for bi-criteria kinematic control of redundant manipulators," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 6, pp. 923–931, Dec 2002.

[9] R. H. Tütüncü, K. C. Toh, and M. J. Todd, "Solving semidefinite-quadratic-linear programs using sdpt3," *Mathematical Programming*, vol. 95, no. 2, pp. 189–217, Feb 2003. [Online]. Available: https://doi.org/10.1007/s10107-002-0347-5

[10] J. F. Sturm, "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones," 1998.

[11] J. Fuchs, "Spread representations," in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, Nov 2011, pp. 814–817.

[12] J. Cadzow, "A finite algorithm for the minimum l_∞ solution to a system of consistent linear equations," *SIAM Journal on Numerical Analysis*, vol. 10, no. 4, pp. 607–617, 1973. [Online]. Available: https://doi.org/10.1137/0710053

[13] Y. Yang and M. Pesavento, "A unified successive pseudoconvex approximation framework," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3313–3328, July 2017.

[14] C. Studer and E. G. Larsson, "Par-aware large-scale multi-user mimo-ofdm downlink," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 2, pp. 303–313, February 2013.

[15] T. Goldstein, M. Li, X. Yuan, E. Esser, and R. Baraniuk, "Adaptive Primal-Dual Hybrid Gradient Methods for Saddle-Point Problems," *ArXiv e-prints*, May 2013.

[16] A. B. Gershman, N. D. Sidiropoulos, S. Shahbazpanahi, M. Bengtsson, and B. Ottersten, "Convex optimization-based beamforming," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 62–75, May 2010.

[17] G. Hegde, Y. Yang, C. Steffens, and M. Pesavento, "Parallel low-complexity m-psk detector for large-scale mimo systems," in *2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, July 2016, pp. 1–5.

[18] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[19] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.