Feasible Point Pursuit and Successive Approximation of Non-Convex QCQPs

Omar Mehanna, *Student Member, IEEE*, Kejun Huang, *Student Member, IEEE*,
Balasubramanian Gopalakrishnan, *Student Member, IEEE*, Aritra Konar, *Student Member, IEEE*, and
Nicholas D. Sidiropoulos, *Fellow, IEEE*

Abstract—Quadratically constrained quadratic programs (QCQPs) have a wide range of applications in signal processing and wireless communications. Non-convex QCQPs are NP-hard in general. Existing approaches relax the non-convexity using semi-definite relaxation (SDR) or linearize the non-convex part and solve the resulting convex problem. However, these techniques are seldom successful in even obtaining a feasible solution when the QCQP matrices are indefinite. In this letter, a new feasible point pursuit successive convex approximation (FPP-SCA) algorithm is proposed for non-convex QCQPs. FPP-SCA linearizes the non-convex parts of the problem as conventional SCA does, but adds slack variables to sustain feasibility, and a penalty to ensure slacks are sparingly used. When FPP-SCA is successful in identifying a feasible point of the non-convex QCQP, convergence to a Karush-Kuhn-Tucker (KKT) point is thereafter ensured. Simulations show the effectiveness of our proposed algorithm in obtaining feasible and near-optimal solutions, significantly outperforming existing approaches.

Index Terms—Feasible point pursuit, linearization, multicast beamforming, non-convex QCQP, semi-definite relaxation, successive convex approximation.

I. INTRODUCTION

QUADRATICALLY constrained quadratic programs (QCQPs) are an important class of optimization problems that have a wide spectrum of applications ranging from transmit beamforming in wireless networks, to portfolio risk management in financial engineering [1], [2]. A QCQP can be expressed as

$$\Pi_1 \min_{\mathbf{x} \in \mathbb{C}^n} \quad \mathbf{x}^H \mathbf{A}_0 \mathbf{x}$$
$$\text{s.t.} \quad \mathbf{x}^H \mathbf{A}_m \mathbf{x} \leq c_m, \quad m = 1, \dots, M \tag{1}$$

where $\mathbf{A}_0 \succeq 0$, i.e., positive semi-definite, and $\mathbf{A}_m \in \mathbb{C}^{n \times n}$ are Hermitian matrices for all $m \in \{1, \dots, M\}$. In the special case when $\mathbf{A}_m \succeq 0$ for all $m \in \{1, \dots, M\}$, the QCQP Π_1 becomes a convex optimization problem which can be efficiently solved to optimality using interior point methods [3]. For general indefinite $\mathbf{A}_m$, this problem is non-convex and NP-hard [4], except for special cases, such as when $M \leq 3$ [5]–[9].

Several methods have been proposed to approximate non-convex QCQPs, including (a) the (prevailing) *semi-definite relaxation* (SDR) approach [3]; (b) the *reformulation linearization technique* (RLT) [10], [11]; and (c) *successive convex approximation* (SCA) [12]–[15]. RLT consists of a reformulation step and a linearization step. The reformulation step creates redundant nonlinear constraints involving pairwise product combinations of the individual scalar variables, by multiplying different constraint pairs. The linearization step then substitutes a continuous variable for each distinct product of variables. The resulting convex optimization problem is solved to obtain an approximate solution of the non-convex problem. The main issue with RLT is that the solution of the linear program is seldom feasible for the non-convex problem. Furthermore the size of the linear program approximation is much larger than the original problem, thereby making it computationally involved.

The SCA approach is a more general scheme to deal with non-convex problems, and its application to non-convex QCQPs is sometimes called the convex-concave approach [4]: each quadratic term is separated into convex and concave parts, and the latter is replaced by a convex (usually linear) approximation around a feasible point. The resulting convex problem is solved to obtain the next iterate, which also serves as the approximation point for the next iteration. Scutari *et al.* recently proposed a parallel and distributed SCA framework to obtain stationary points for non-convex optimization problems [15]. The algorithm starts from an initial feasible point, the non-convex constraints are approximated by a strictly convex function around this point, and the resulting convex problem is solved to obtain the next iterate. This procedure is repeated until convergence to a stationary point.

The drawback with approaches (b) and (c) is that they need a feasible point as initialization, which is difficult to obtain in general. Constraint approximation about a feasible point yields a nonempty set that contains at least the given point, whereas constraint approximation about an infeasible point tends to yield an empty set, even if the original problem is feasible. Existing convergence results for SCA depend on a feasible initialization, e.g., [12].

The most popular among the above is the SDR approach [1], [3], where the original problem is reformulated by introducing

$\mathbf{X} := \mathbf{x}\mathbf{x}^H$ and solving the semi-definite program (SDP) $\mathbf{\Pi}_{1r}$ obtained after relaxing the rank-1 constraint.

$$\mathbf{\Pi}_{1r} \quad \min_{\mathbf{X} \in \mathbb{C}^{n \times n}} \quad \mathrm{Trace}(\mathbf{A}_0 \mathbf{X})$$
$$\text{s.t.} \quad \mathrm{Trace}(\mathbf{A}_m \mathbf{X}) \leq c_m, m = 1, \ldots, M \qquad (2)$$
$$\mathbf{X} \succeq 0$$

Because of the rank relaxation, the solution to $\mathbf{\Pi}_{1r}$ gives a lower bound on the optimal value of the cost function of $\mathbf{\Pi}_1$. Note that the SDR $\mathbf{\Pi}_{1r}$ is the Lagrange bi-dual of $\mathbf{\Pi}_1$. If the solution $\mathbf{X}^*$ to problem $\mathbf{\Pi}_{1r}$ is rank-1, the optimal solution $\mathbf{x}^*$ to $\mathbf{\Pi}_1$ is the principal eigenvector of $\mathbf{X}^*$, scaled by the square root of the maximum eigenvalue of $\mathbf{X}^*$; otherwise, randomization techniques are used [1]. If the matrices $\{\mathbf{A}_m\}_{m=1}^M$ are all negative semidefinite, then any randomly generated point can be scaled up to satisfy the constraints of the QCQP $\mathbf{\Pi}_1$; finding a feasible solution using randomization is easy in this case, and the challenge is to find one that is close to optimum, see [1], [16].

In the general setting where $\{\mathbf{A}_m\}_{m=1}^M$ are all indefinite, or when one deals with two-sided positive semidefinite constraints such as in [17], SDR with randomization often fails to find a point that satisfies the constraints in $\mathbf{\Pi}_1$. That is why it is important to develop an alternate approach (instead of SDR followed by randomization) that has a high probability of finding a feasible solution to the NP-hard QCQP $\mathbf{\Pi}_1$, when one exists.

In this letter we propose an iterative algorithm for obtaining good feasible solutions to general QCQPs, where we approximate the feasible region through a linear restriction of the non-convex parts of the constraints. In order to guarantee feasibility of the modified problem, slack variables are added, and a penalty is used to ensure that slacks are sparingly used. The solution of the resulting optimization problem is then used to compute a new linearization, and the procedure is repeated until convergence. The proposed *Feasible Point Pursuit - Successive Convex Approximation* (FPP-SCA) algorithm differs from the conventional SCA approach [12]–[14] in that the latter requires the starting point to be in the feasible region of the original problem. Finding a feasible point is easy in some cases, such as when all the constraints $\mathbf{x}^H \mathbf{A}_m \mathbf{x} \leq c_m$ involve negative semi-definite $\mathbf{A}_m$, as considered in [14]. For general QCQPs, however, finding an initial feasible point is hard. The performance of the proposed algorithm is compared with the conventional SDR followed by randomization, and simulations show that the proposed algorithm attains a feasible solution for a much larger percentage of problem instances. Furthermore, the feasible solution obtained using the proposed algorithm is much closer to the SDR lower bound than SDR followed by randomization.

II. THE FPP-SCA APPROACH

Problem $\mathbf{\Pi}_1$ may or may not be feasible, and establishing (in)feasibility is generally NP-hard. When infeasible, one may instead seek a compromise that minimizes constraint violations in some sense - this is common in engineering applications. In order to account for potential infeasibility, consider adding slack variables $\mathbf{s} \in \mathbb{R}^M$ and a slack penalty to $\mathbf{\Pi}_1$

$$\mathbf{\Pi}_2 \quad \min_{\mathbf{x} \in \mathbb{C}^n, \mathbf{s} \in \mathbb{R}^M} \quad \mathbf{x}^H \mathbf{A}_0 \mathbf{x} + \lambda \|\mathbf{s}\|$$
$$\text{s.t.} \quad \mathbf{x}^H \mathbf{A}_m \mathbf{x} \leq c_m + s_m,$$
$$s_m \geq 0, \quad m = 1, \ldots, M, \qquad (3)$$

where λ trades off the original objective function and the slack penalty term, and $\| \cdot \|$ can be any vector norm. Problem $\mathbf{\Pi}_2$ is always feasible, and if $(\mathbf{x}_o, \mathbf{s}_o)$ is an optimal solution of $\mathbf{\Pi}_2$ and it so happens that $\mathbf{s}_o = 0$, then $\mathbf{x}_o$ is an optimal solution of $\mathbf{\Pi}_1$; else using the l_1 norm of $\mathbf{s}$ in $\mathbf{\Pi}_2$ (which reduces to the sum of the slacks, due to the non-negativity constraints) promotes sparsity in terms of constraint violations. The difficulty though is that problem $\mathbf{\Pi}_2$ remains non-convex and NP-hard in general.

Successive convex approximation (SCA). Using eigen-decomposition, the matrix $\mathbf{A}_m$ can be expressed as $\mathbf{A}_m = \mathbf{A}_m^{(+)} + \mathbf{A}_m^{(-)}$, where $\mathbf{A}_m^{(+)} \succeq 0$ and $\mathbf{A}_m^{(-)} \preceq 0$ (negative semi-definite). For any $\mathbf{z}, \mathbf{x} \in \mathbb{C}^{n \times 1}$, $(\mathbf{x} - \mathbf{z})^H \mathbf{A}_m^{(-)} (\mathbf{x} - \mathbf{z}) \leq 0$. Expanding the left-hand side of the inequality, we obtain

$$\mathbf{x}^H \mathbf{A}_m^{(-)} \mathbf{x} \leq 2\mathrm{Re}\left\{ \mathbf{z}^H \mathbf{A}_m^{(-)} \mathbf{x} \right\} - \mathbf{z}^H \mathbf{A}_m^{(-)} \mathbf{z}. \qquad (4)$$

Therefore, using the linear *restriction* (4) around the point $\mathbf{z}$, we may replace the m-th (non-convex) constraint of $\mathbf{\Pi}_2$ with the convex constraint

$$\mathbf{x}^H \mathbf{A}_m^+ \mathbf{x} + 2\mathrm{Re}\left\{ \mathbf{z}^H \mathbf{A}_m^{(-)} \mathbf{x} \right\} \leq c_m + \mathbf{z}^H \mathbf{A}_m^{(-)} \mathbf{z} + s_m. \qquad (5)$$

This leads us to propose the following algorithm.

Algorithm 1 Feasible Point Pursuit Successive Convex Approximation (FPP-SCA) Algorithm

---

Initialization: Set $k = 0$ and randomly generate an initial point $\mathbf{z}_0$.

Repeat
1) Solve

$$\mathbf{\Pi}_3 \quad \min_{\mathbf{x}, \mathbf{s}} \quad \mathbf{x}^H \mathbf{A}_0 \mathbf{x} + \lambda \sum_{m=1}^M s_m$$
$$\text{s.t.} \quad \mathbf{x}^H \mathbf{A}_m^{(+)} \mathbf{x} + 2\mathrm{Re}\left\{ \mathbf{z}_k^H \mathbf{A}_m^{(-)} \mathbf{x} \right\}$$
$$\leq c_m + \mathbf{z}_k^H \mathbf{A}_m^{(-)} \mathbf{z}_k + s_m$$
$$s_m \geq 0, \quad m = 1, \ldots, M$$
$$(6)$$

2) Let $\mathbf{x}_k$ denote the optimal $\mathbf{x}$ obtained by solving $\mathbf{\Pi}_3$ at the k-th iteration, and set $\mathbf{z}_{k+1} = \mathbf{x}_k$.
3) Set $k = k + 1$.

until convergence.

Some important remarks and claims are in order.
- We first relaxed the constraints in $\mathbf{\Pi}_1$ by adding slacks, then tightened the relaxed constraints via partial linear restriction of their non-convex parts. We could instead first tighten the original constraints (risking turning a feasible original problem into an infeasible one) then relax by adding slacks to make the restriction feasible–the net result turns out being $\mathbf{\Pi}_3$ in both cases, and it is always feasible.
- FPP-SCA yields a non-increasing cost sequence, i.e., the optimal cost of $\mathbf{\Pi}_3$ is non-increasing in k. This is because the cost function is independent of k, and the solution of the k-th iteration is also feasible for the $(k + 1)$-th iteration. To see this, note that $\mathbf{z}_{k+1} = \mathbf{x}_k$ is the optimal solution of $\mathbf{\Pi}_3$ at the k-th

iteration, so it satisfies the restriction $\mathbf{z}_{k+1}^H \mathbf{A}_m^{(+)} \mathbf{z}_{k+1} + 2\mathrm{Re}\{\mathbf{z}_k^H \mathbf{A}_m^{(-)} \mathbf{z}_{k+1}\} \leq c_m + \mathbf{z}_k^H \mathbf{A}_m^{(-)} \mathbf{z}_k + s_m$, and therefore *a fortiori* also the non-convex quadratic constraint $\mathbf{z}_{k+1}^H \mathbf{A}_m^{(+)} \mathbf{z}_{k+1} + \mathbf{z}_{k+1}^H \mathbf{A}_m^{(-)} \mathbf{z}_{k+1} \leq c_m + s_m$. Looking at the corresponding constraint at the next iteration, $\mathbf{x}^H \mathbf{A}_m^{(+)} \mathbf{x} + 2\mathrm{Re}\{\mathbf{z}_{k+1}^H \mathbf{A}_m^{(-)} \mathbf{x}\} \leq c_m + \mathbf{z}_{k+1}^H \mathbf{A}_m^{(-)} \mathbf{z}_{k+1} + s_m$, plugging in $\mathbf{x} = \mathbf{z}_{k+1}$ we obtain $\mathbf{z}_{k+1}^H \mathbf{A}_m^{(+)} \mathbf{z}_{k+1} + \mathbf{z}_{k+1}^H \mathbf{A}_m^{(-)} \mathbf{z}_{k+1} \leq c_m + s_m$, i.e., feasibility of $(\mathbf{x}_k = \mathbf{z}_{k+1}, \mathbf{s})$ for the k-th iteration of $\mathbf{\Pi}_3$ implies feasibility of $(\mathbf{z}_{k+1}, \mathbf{s})$ for the same $\mathbf{s}$ for the $(k+1)$-th iteration of $\mathbf{\Pi}_3$.

- Problem $\mathbf{\Pi}_3$ is convex and can be easily formulated as a second-order cone program (SOCP). Worst-case complexity of solving $\mathbf{\Pi}_3$ is $\mathcal{O}([n+M]^{3.5}) \ll \mathcal{O}(n^{6.5})$ for SDR[1]. FPP-SCA usually takes just a few iterations to converge.
- FPP-SCA can be run using different starting points $\mathbf{z}_0$, and the best solution can be taken. Simulations suggest that SDR can provide a good initialization for FPP-SCA, if the extra complexity of SDR is acceptable. Otherwise random initialization(s) can be used.
- We propose using $\lambda \gg 1$ to force the slack variables toward zero, thereby pushing the iterates towards the feasible region of $\mathbf{\Pi}_1$ when this is non-empty. Higher λ also helps ensure that if a feasible point of $\mathbf{\Pi}_1$ is found, subsequent iterates will remain in the feasible region of $\mathbf{\Pi}_1$, although there are no analytical guarantees for this. Note that the main advantage of FPP-SCA over conventional SCA is the ability to find a feasible point with high probability. Once the slacks are all 0, one can simply switch to conventional SCA without the slacks. Our simulations show that the subsequent iterates are almost identical between these two schemes (with or without the slack variables).
- If FPP-SCA converges, it converges to a KKT point for problem $\mathbf{\Pi}_2$, according to Beck *et al.* [12]. If the converged slack variables turn out being all zero, then it is easy to show that the remaining variables satisfy the KKT conditions for the original problem $\mathbf{\Pi}_1$.

Illustrative example. To get a better understanding of the approximations used in the FPP-SCA algorithm and how the solution evolves after each iteration, Fig. 1 considers a setup in $\mathbb{R}^2$ with $M = 3$, where $\mathbf{A}_0 = I$, $\mathbf{A}_1 = \begin{bmatrix} -1.48 & 0.68 \\ 0.68 & -0.52 \end{bmatrix}$, $\mathbf{A}_2 = \begin{bmatrix} -0.93 & -0.07 \\ -0.07 & -1.07 \end{bmatrix}$, $\mathbf{A}_3 = \begin{bmatrix} 1.59 & -0.17 \\ -0.17 & 0.41 \end{bmatrix}$, $c_1 = c_2 = -1$, and $c_3 = 1$. Note that $\mathbf{A}_1$ and $\mathbf{A}_2$ are negative semidefinite, whereas $\mathbf{A}_3$ is positive semidefinite. The ellipsoids that correspond to $\mathbf{x}^H \mathbf{A}_m \mathbf{x} = c_m$ for $m = 1, 2$ are plotted in red in Fig. 1, while the ellipsoid for $m = 3$ is plotted in green. Two different initializations for $\mathbf{z}_0$ are considered, both are chosen with relatively large scale so that the first two constraints of the QCQP are satisfied, but not the third one. Therefore, only the slack variables that correspond to the third constraint can be nonzero. The FPP-SCA algorithm is run for 3 iterations using $\lambda = 10$, and the figure shows the solution after each iteration. The point
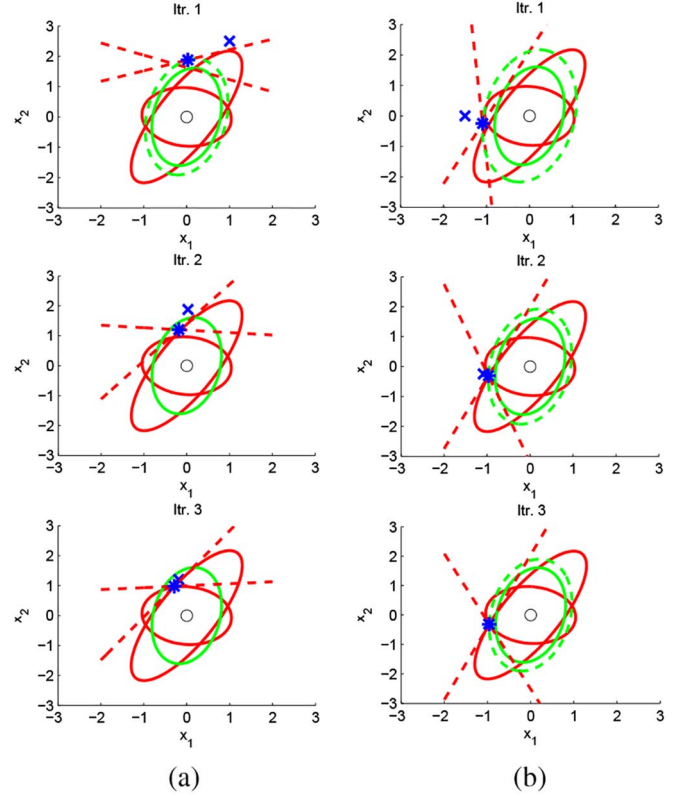

Fig. 1. Illustration of FPP-SCA algorithm in $\mathbb{R}^2$, $M = 3$: 3 iterations of successful (left) and unsuccessful (right) FPP (a) Successful (b) Unsuccessful.

about which the non-convex constraints are linearized in each iteration is denoted by a blue cross, while the solution obtained in each iteration is denoted by a blue star. For the non-convex constraints that correspond to $m = 1, 2$, the linear approximation (restriction) of each ellipsoid at each iteration is plotted in a dashed red line. The (extended) ellipsoid after adding the necessary slack to the convex constraint $m = 3$, in order to obtain a feasible solution for $\mathbf{\Pi}_3$, is plotted in dashed green.

As shown in Fig. 1, for the first initialization point (case (a) - left panels), a feasible solution to the original problem is obtained after the second iteration, and the optimum solution is achieved after 3 iterations. On the other hand, for the second initialization point (case (b) - right panels), the algorithm is stuck at a point that is not feasible for the original problem. Note that FPP-SCA converges in both cases, albeit to an undesirable point in the second case.

III. NUMERICAL RESULTS

To test the performance of the FPP-SCA algorithm, a problem with $n = 8$ complex dimensions is considered, with $M \in \{16, 24, 32\}$. The entries of the matrices $\{\mathbf{A}_m\}_{m=1}^M$ are randomly and independently generated from a complex Gaussian distribution (with zero-mean and variance 2), then symmetrized. An initial point $\mathbf{x}_{\mathrm{init}}$ is randomly generated, and the values of $\{c_m\}_{m=1}^M$ are randomly generated from a Gaussian distribution $c_m \sim \mathcal{N}(\mathbf{x}_{\mathrm{init}}^H \mathbf{A}_m \mathbf{x}_{\mathrm{init}}, 1)$. If $\mathbf{x}_{\mathrm{init}}^H \mathbf{A}_m \mathbf{x}_{\mathrm{init}} > c_m$, the matrix $\mathbf{A}_m$ and c_m are multiplied by -1 to get $\leq$ inequalities. The matrix $\mathbf{A}_0$ is set to the identity. To solve the SDR $\mathbf{\Pi}_{1\mathrm{r}}$ and the SCA $\mathbf{\Pi}_3$, the modeling language YALMIP [19] is used and the generic conic programming solver SeDuMi [20] is chosen as the

---

[1]The primal-dual method converges in $\mathcal{O}(n^{0.5})$ iterations [18, Theorem 5.1], each requiring $\mathcal{O}(m^2 n^2)$, where m is the number of equality constraints, and the PSD matrix is $n \times n$ [18, Section 7.6]. To convert $M = \mathcal{O}(n)$ inequalities to equalities, introduce M auxiliary nonnegative variables and $m = \mathcal{O}(M^2) = \mathcal{O}(n^2)$ equalities, yielding $\mathcal{O}(n^6)$ per iteration.

TABLE I
RESULTS USING THE SDR APPROACH FOR $n = 8$

M	16	24	32
Rank-1 solution	45%	16%	5.5%
No feasible sol. after randomization	42%	80%	94.2%
Feasible sol. after randomization	13%	4%	0.3%
Avg. loss (dB)	1.3	1.5	-

TABLE II
RESULTS USING THE FPP-SCA APPROACH FOR $n = 8$

M	16	24	32
Feasible solution	100%	99.5%	92.8%
Avg. itrs. for feasibility	3.207	4.871	7.893
Avg. itrs. for convergence	10.97	11.6371	12.1352
Avg. loss (dB)	0.942	1.5684	1.9256

TABLE III
RESULTS USING THE SDR APPROACH FOR $n = 20$

M	32	40	48
Rank-1 solution	11.5%	2.9%	0.4%
No feasible sol. after randomization	84.7%	96.4%	99.5%
Feasible sol. after randomization	3.80%	0.7%	0.1%

TABLE IV
RESULTS USING THE FPP-SCA APPROACH FOR $n = 20$

M	32	40	48
Feasible solution	100%	100%	100%
Avg. itrs. for feasibility	4.3640	5.0927	5.9840
Avg. itrs. for convergence	16.2130	16.5368	17.2805
Avg. loss (dB)	0.4570	0.4881	0.5618

solver for both approaches. The results reported in Tables I–IV are averaged over 1000 Monte-Carlo simulation runs.

In Table I, we consider solving the QCQP $\boldsymbol{\Pi}_1$ using SDR $\boldsymbol{\Pi}_{1r}$ followed by a randomization (and scaling) technique that is similar to the one used in [17], if the solution to $\boldsymbol{\Pi}_{1r}$ is not rank-1. For the randomization step, 10^4 random points are generated for each simulation run. The table reports the average number of simulation runs where a rank-1 solution was obtained, the average number of simulation runs where no feasible solution was obtained after the randomization step, the average number of simulation runs where a feasible solution was obtained with randomization, and the average difference between the solution obtained with randomization and the lower bound obtained from the (higher-rank) solution of $\boldsymbol{\Pi}_{1r}$. The table shows that as M increases (i.e., the set of constraints becomes more stringent), the percentage of feasible solutions that can be obtained using the SDR approach (either directly from rank-1 solutions or after the randomization step) diminishes quickly.

In Table II, we consider solving the QCQP $\boldsymbol{\Pi}_1$ using the FPP-SCA algorithm which solves $\boldsymbol{\Pi}_3$ in each iteration (setting $\lambda = 10$). The maximum number of iterations was set to 30 and convergence was declared if $\|\mathbf{x}_k^H \mathbf{A}_0 \mathbf{x}_k - \mathbf{x}_{k-1}^H \mathbf{A}_0 \mathbf{x}_{k-1}\| \leq 10^{-4}$, for $k \geq 1$. The vector $\mathbf{z}_0$ used to initialize the FPP-SCA algorithm in each simulation run was randomly drawn from an i.i.d. complex circularly symmetric zero mean Gaussian distribution of variance 2. The table reports the average number of simulation runs where a feasible solution was obtained (i.e., $\mathbf{s} = 0$), the average number of iterations until a feasible solution was obtained, the average number of iterations until convergence is declared, and the average difference between the solution obtained by FPP-SCA and the lower bound obtained

from the (higher-rank) solution of $\boldsymbol{\Pi}_{1r}$. The table shows that a feasible solution can be obtained from FPP-SCA with very high probability even for large M, unlike the SDR approach. With $M = 32$ for example, it is almost impossible to find a solution using SDR followed by randomization if the solution is not rank-1, whereas in 92.8% of the cases FPP-SCA managed to find a feasible point. This percentage can even be increased further if multiple starting points are considered for the non-feasible cases. The table also shows that few iterations are required for the algorithm to converge, and much fewer iterations are required to reach a feasible point. Finally, the table shows that the solutions obtained using the FPP-SCA algorithm are very close to the generally unattainable relaxation lower bound provided by the SDR. Tables III and IV show similar results for a higher dimension $n = 20$.

Multicast Beamforming under Interference Constraints. We further illustrate the advantage of FPP-SCA using a wireless communication design problem, namely *secondary multicast beamforming* as considered in [17], which can be posed as the following non-convex QCQP:

$$\begin{aligned} \min_{w \in \mathbb{C}^n} \quad & \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & |\mathbf{w}^H \mathbf{h}_i|^2 \geq \tau, \ i = 1, \ldots, M \\ & |\mathbf{w}^H \mathbf{g}_k|^2 \leq \eta, \ k = 1, \ldots, K \end{aligned} \quad (7)$$

which describes a system comprising a secondary transmitter with n antennas, M secondary single-antenna receivers interested in the same multicast, and K primary single-antenna receivers. The M secondary receivers should be provided with signal power no less than some threshold, while the K primary receivers should be protected from excessive interference. The channel gains from the transmit antennas to the i-th secondary user are denoted as $\mathbf{h}_i$, and those to the k-th primary user as $\mathbf{g}_k$. We assume i.i.d. Rayleigh fading, i.e., the channels are drawn from an i.i.d. zero-mean complex Gaussian distribution with $\sigma^2 = 1$.

For smaller problem dimensions, like the ones simulated in [17], a feasible point is easy to find using SDR and randomization. However, it becomes very hard to find a feasible point when the problem size becomes higher. We conducted simulations for $n = 8, M \in \{12, \cdots, 24\}, K = 4, \tau = 10$ and $\eta = 1$. We simulated 1000 random problem instances having feasible SDR. After SDR, we drew 10^4 randomization points for each problem instance. None of them was (or could be scaled to be) feasible. However, FPP-SCA initialized with an SDR randomization point managed to find a feasible solution in all problem instances, with only minor average power increase (ranging from 1 to 2.2 dB, for M ranging from 12 to 24, respectively) compared to the generally unattainable relaxation lower bound provided by SDR.

IV. CONCLUSIONS

FPP-SCA is a new iterative approach for approximately solving general QCQPs. FPP-SCA was compared with conventional SDR followed by randomization, and it was observed that FPP-SCA was successful in obtaining good feasible solutions for a much higher percentage of problem instances than SDR plus randomization, at a lower worst-case complexity, and smaller gap to the relaxation lower bound. The results suggest that FPP-SCA holds promise for a broad range of applications in engineering design problems that can be cast as non-convex QCQPs.

REFERENCES

[1] Z.-Q. Luo, W. Ma, A. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, May 2010.

[2] Y. Huang and D. P. Palomar, "Randomized algorithms for optimal solutions of double-sided QCQP with applications in signal processing," *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1093–1108, Mar. 2014.

[3] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[4] A. d'AspermontS. Boyd, "Relaxations and randomized methods for nonconvex QCQPs," Stanford Univ., Stanford, CA, USA, 2003, EE392 Lecture Notes.

[5] G. Pataki, "On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues," *Math. Oper. Res.*, vol. 23, no. 2, pp. 339–358, 1998.

[6] Y. Ye and S. Zhang, "New results on quadratic minimization," *SIAM J. Optim.*, vol. 14, no. 1, pp. 245–267, 2003.

[7] Y. Huang and S. Zhang, "Complex matrix decomposition and quadratic programming," *Math. Oper. Res.*, vol. 32, no. 3, pp. 758–768, 2007.

[8] Y. Huang and D. P. Palomar, "Rank-constrained separable semidefinite programming with applications to optimal beamforming," *IEEE Trans. Signal Process.*, vol. 58, no. 2, pp. 664–678, Feb. 2010.

[9] Y. Huang, A. De Maio, and S. Zhang, "Semidefinite programming, matrix decomposition, and radar code design," in *Convex Optimization in Signal Processing and Communications*, D. P. Palomar and Y. Eldar, Eds. Cambridge, U.K.: Cambridge Univ. Press,, 2010, ch. 6.

[10] K. M. Anstreicher, "On convex relaxations for quadratically constrained quadratic programming," *Math. Program.*, vol. 136, no. 2, pp. 233–251, Dec. 2012.

[11] H. D. Sherali and W. P. Adams, *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Norwell, MA, USA: Kluwer, 1998.

[12] A. Beck, A. Ben-Tal, and L. Tetruashvili, "A sequential parametric convex approximation method with applications to nonconvex truss topology design problems," *J. Global Optim.*, vol. 47, no. 1, pp. 29–51, 2010.

[13] B. R. Marks and G. P. Wright, "A general inner approximation algorithm for nonconvex mathematical programs," *Oper. Res.*, vol. 26, no. 4, pp. 681–683, 1978.

[14] L. N. Tran, M. F. Hanif, and M. Juntti, "A conic quadratic programming approach to physical layer multicasting for large-scale antenna arrays," *IEEE Signal Process. Lett.*, vol. 21, no. 1, pp. 114–117, Jan. 2014.

[15] G. Scutari, F. Facchinei, L. Lampariello, and P. Song, "Parallel and distributed methods for nonconvex optimization," in *Proc. ICASSP*, May 2014.

[16] N. D. Sidiropoulos, T. Davidson, and Z.-Q. Luo, "Transmit beamforming for physical layer multicasting," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2239–2251, Jun. 2006.

[17] K. T. Phan, S. A. Vorobyov, N. D. Sidiropoulos, and C. Tellambura, "Spectrum sharing in wireless networks via QoS-aware secondary multicast beamforming," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2323–2335, Jun. 2009.

[18] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Rev.*, vol. 38, pp. 49–95, 1996.

[19] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in MATLAB," in *Proc. CACSD*, Taipei, Taiwan, Sep. 4, 2004.

[20] J. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optim. Meth. Softw.*, vol. 11, no. 1, pp. 625–653, 1999.