On the Capacity Optimization for MIMO Ad Hoc Networks

Khalil Fakih, Jean-Francois Diouris, and Guillaume Andrieux
Ecole polytechnique de l’université de Nantes and IREENA, BP 50609, Nantes, France
Email: firstname.lastname@univ-nantes.fr

Abstract—Maximizing the system mutual information for MIMO ad hoc networks is considered in this paper. We first give a brief overview of the non-linear optimization methods and systematically we compare their performances. Then we propose a fast and distributed algorithm based on the quasi-Newton methods to give a lower bound of the system capacity of MIMO ad hoc networks. Our proposed algorithm solves the maximization problem while diminishing the amount of information in the feedback links needed in the cooperative optimization. Finally, an extensive set of simulations shows that our algorithm enjoys a provable convergence and presents the highest convergence rate.

Index Terms—MIMO ad hoc, Capacity optimization.

I. INTRODUCTION

Recently, MIMO ad hoc networks have attracted an increasing interest. The use of multiple antennas at both wireless link sides has shown a promising solution to boost up the spectral efficiency of the cellular communication systems [1]. In ad hoc networks, where the nodes operate without a central administration or underlying infrastructure, the MIMO links play an important role in overcoming some problems such as the lower system throughput and the higher energy consumption. However, a smart optimization signaling algorithm associated with sophisticated MAC/routing schemes have to be proposed in order to handle these benefits [12].

In the literature, extensive research is devoted to MIMO broadcast (MIMO BC) and to MIMO MAC systems. Recall that, in these systems either the transmitter or the receiver is common between the active wireless links. These researches issues concern the problem of the capacity optimization. In ad hoc networks, where each node can communicate with any other nodes located in its coverage zone, the optimization problem needs a careful study for two reasons: the first one is that we are in fully interfering environment and the second one is the sensitivity of the performance of ad hoc networks on the overheads introduced by the feedback link required for any cooperative optimization.

The optimum signaling in the case where the CSI (Channel State Information) is assumed only at the receiver, is considered in [10]. The authors demonstrate that putting all power into a single transmitting antenna is optimum in the case of weak interferences. Whereas, dividing the power equally between independent streams from the different antennas is optimum when strong interferences are expected. The great potential of MIMO communication in single link scenario is proven in [7]. However, evidencing this potentiality in the case of ad hoc networks, where the users mutually interfere is not a trivial problem. The same problem can be found in cellular communication system with cochannel interferences. In [11] the network spectral efficiency is quantified. In [6], the authors deal with the MIMO BC sum rate problem. They designed an algorithm based on the conjugate gradient (CG) method to solve the primal problem. The authors show that the proposed algorithm enjoys fast and provable convergence. However, the unconstrained problem emphasized in this work does not take into account the interference matrix as in the case of MIMO ad hoc networks. A modified version of this algorithm will be applied to our problem for comparison issues. In [8] a non-cooperative algorithm is proposed to solve for global problem. Resolving for optimum signaling for the non-interference and the fixed-interference cases is done with the traditional and generalized waterfilling procedures respectively in [7] and [13]. In real MIMO ad hoc networks, the interferences at each node depend on the transmit covariance matrices for all other nodes. Thus, the problem of maximizing the global capacity will be more complicated and this global maximization came usually at the cost of frequently feedback signaling which depends on the convergence rate of the proposed algorithm. In the literature and due to the non-concavity of this problem, only a suboptimum solution is found by using some non-linear programming methods. The gradient projection (GP) algorithm proposed in [2], maximizes the total system capacity subject to constant power constraint at each node in the network. In their work, the authors present centralized and distributed schemes to solve the problem. Although the proposed algorithm shows a provable convergence, its convergence rate slows down as it is approaching the solution. When performing cooperative and distributed optimization the nodes may share some data along the convergence process. The amount of information to be transmitted in the feedback link will grow with the number of iterations. Thus, reducing this number alleviates the unsupportable overheads. In this context, the Newton method becomes an intuitive candidate for such a problem. However, due the complexity of computing the inverse of the Hessian matrix, this solution will be excluded. As an intermediate solution, we propose to use the quasi-Newton (QN) methods that approximate the inverse of the Hessian matrix rather than computing the true one. The approximation of the inverse of the Hessian is based on the calculated gradient. These methods are motivated for the following reasons: i) A provable and super linear convergence can be achieved. ii) The complexity
of these methods is far from that of the Newton method and comparable to the gradient method.

In this work, we investigate the DFP (Davidon-Fletcher-Powell) method and the SSQN (Self Squalling Quasi Newton) method. For completeness, we implement also the CGP (Conjugate Gradient Projection) method. Moreover, all the aforementioned methods will be reviewed and systematically compared. Note that, we retain the projection method to fulfill the constant power constraint. An extensive set of simulations shows that the performance of the QN methods is close to that of the GP method while the convergence rate of the QN methods is much better.

The rest of this paper is organized as follows: in section II, the network model and the problem formulation are presented. In section III, the aforementioned iterative methods are overviewed. The cooperative and distributed optimization algorithm based on the QN methods for MIMO ad hoc networks is introduced in section IV. Numerical results are reported in section V and finally a conclusion is given in section VI.

The notation in this work will be as follows. The boldface denotes matrices and vectors. For a matrix $R$: $R^*$, $R^T$ and $R^H$ denotes the conjugate, the transpose, and the conjugate-transpose, respectively. $tr(R)$ is the trace. $R \geq 0$ represents a positive semi definite matrix.

II. System Model & Problem Formulation

In our work, we consider an ad hoc network formed by $N$ nodes. Each of which employs $M$ antenna-elements. The links in the network are assumed to be unicast predefined links. The nodes perform single user detection. We assume also that the channel and the noise are independent and identically distributed (iid) complex Gaussian distributed with zero mean. For such receiver the interfering signals are unknown. Thus we model them as Gaussian distributed [7]. Because we are based on the gradient projection method proposed in [2], we hold the same system model. That is, the received signal can be seen as the multiplication of the normalized weighted transmitted signal $x$ by the Signal to Noise Ratio ($SNR$) of this signal, namely $\rho$, and also multiplied by the correspondent channel. By focusing our attention on the node $i$, the baseband signal received by this node is given by: $y_i = \sqrt{\rho} H_{i} x + \sum_{j=1, j \neq i}^{N} \sqrt{\gamma_j} H_{i,j} x_j + n_i$, where $\gamma$ is the Interference to Noise Ratio ($INR$). The first term represents the desired signal intended to node $i$ while the second term regroups the total interference signal received by node $i$. The third term is the additive white Gaussian noise. Under the assumption that the channel and the noise are independent and that $E(n,n^H) = I$, the covariance matrix of interference plus noise is given by: $R_i = I + \sum_{j=1}^{N} \gamma_i \gamma_j H_{i,j} Q_j H_{i,j}^H$, where $Q_j = E(x_j x_j^H)$ represents the transmit covariance matrix. Finally the total system capacity can be written as:

$$C = \sum_{i=1}^{N} c_i = \sum_{i=1}^{N} E \left[ \log_2 \left( \det(I + \rho_i H_{i,i} Q_i H_{i,i}^H R_i^{-1}) \right) \right]$$

(1)

where, the expectation is taken over all the random channel matrices. The global optimization problem is, then:

$$\begin{cases}
\text{maximize} & C \\
\text{Subject to} & tr(Q_i) = 1 \quad i = 1...N.
\end{cases}$$

(2)

Due to the non-concavity of this problem, only a suboptimum solution can be found through non-linear optimization methods.

III. Mathematical Review

Let $f$ be a multi variable function defined on the subspace $E^n$. Without loss of generality, all the iterative descent/ascent methods are defined as an update of the solution at each iteration. A generic algorithm is given by the following equation:

$$x_{k+1} = x_k + \alpha_k g_k$$

(3)

where, $g_k$ is the step or the search direction and $\alpha_k$ is the step size.

**Determination of the step size**

Exact lines search is the evident and more accurate method in this context. With this method $\alpha_k$ can be computed as:

$$\alpha = \arg \min(x + \alpha g)$$

(4)

In practice, the exact line search may be hard to find. Inexact line search methods are more appropriate and easier to be implemented [4]. One of these methods, called back tracking line search, depends on two constants $a$, $b$ with $0 < a < 0.5$ and $0 < b < 1$ and it consists of iteratively increment the variable $t$ until fulfilling the following condition:

$$f(x_{k+1}) - f(x_k) \geq a t g^T g : \quad t = b^m t_0$$

(5)

**Determination of the step**

The general descending method can be written as $x_{k+1} = x_k + \alpha_k S g_k$. Almost all the methods defines $g_k$ as the gradient of $f$. The difference is only in the definition of the matrix $S$.

The method of steepest descent (referred later here as the gradient method) defines $S$ as the identity matrix. The idea behind this method is that the function $f$ is approximated locally by a linear function. This method is one of the widely used methods for minimizing a function of several variables. It is extremely motivated since it is very simple to be implemented and only the first partial derivatives of $f$ are required. However, the convergence rate of this method is very slow and is tightly depending on the initial point. This slowness can be interpreted by the fact that two consecutive search direction vectors are orthogonal. That is, $g_k^T g_{k+1} = 0$. 

1088
More careful examination on the convergence of this method can be found in [3].

The Newton’s method can achieve a super linear convergence by defining S as the inverse of the Hessian matrix of \( f \). Let \( F \) denotes the inverted matrix. Herein, the function \( f \) is approximated locally by a quadratic function and this approximate function is minimized exactly. Therefore, this method can eliminate efficiently the “jamming” or “zigzagging” phenomenon encountered by the gradient method. The order of convergence of this method is two if the initial point is closed to the solution. Although the Newton’s method is very attractive in terms of convergence properties, it requires a complex evaluation and inversion of the Hessian matrix at each iteration.

The CG method and theQN methods can be regarded as being some what intermediate between the method of steepest descent and Newton’s method.

The CG method is motivated to accelerate slow convergence of the steepest descent method while avoiding the evaluation and inversion of the Hessian matrix as required by the Newton’s method. This method is used in the context of MIMO BC [6], in order to maximize the global capacity under global power constraint.

TheQN methods use an approximation of the inverse of the Hessian matrix rather than the true inverse that is required in the Newton’s method. This approximated matrix can be build up on the base of information gathered along the convergence way. These methods offer the most simple, sophisticated, and fast algorithms for solving the unconstrained problems. In our work, we focus on these methods, explain their features and discuss their performances and benefits in the cooperative capacity optimization for MIMO ad hoc network.

IV. QUASI-NEWTON METHOD FOR MIMO AD HOC NETWORKS

In this work, we propose a fast and efficient algorithm based on the quasi Newton method to solve the global optimization problem (2). Our work is based on the gradient projection method proposed in [2] and detailed in [3]. As we have seen in the mathematical review section, the descent direction in the latter method is based essentially on the gradient of the total capacity \( C \). This gradient is calculated with respect to the transmit signaling matrix \( Q_i \) of the user \( i \). In our proposed algorithm, we deflect this gradient direction in order to achieve the most possible linear convergence rate. The deflection is done by approximating the inverse of the Hessian matrix by using the DFP and the SSQN methods. Along the convergence way, the gradient is calculated and the inverse of the Hessian is updated accordingly. The detailed procedure is illustrated in the Algorithm 1. Note that, for convenience we use the symbols vec and mat to convert the matrices into vectors and to concatenate the vectors into matrices, respectively.

According to our algorithm, the sub optimum is reached by a cooperative and distributed way which is the most suitable solution for ad hoc networks. More precisely, each user updates independently his covariance matrix with respect to the other updated and not-updated covariance matrices for other users. That is, when calculating the matrix \( Q_i(k+1) \) at the \( k^{th} \) iteration, the user \( i \) broadcasts the calculated matrix to other user, so by that they can proceed the calculation of their own matrices \( Q_{j/j\#} (k+1) \) successively. Clearly, the amount of information to be sent in the feedback link in order to reach the local optimum is straightforward depending on the rate convergence. Such a case in real ad hoc network may generate unsupportable overheads. Explicitly, the network will be saturated by the feed back information. Thus, we can see the utility to optimize the global capacity while keeping limited the amount of information to be transmitted in the feedback link. The direct solution of this problem is to minimize the number of iterations. Our proposed algorithm by using the quasi-Newton methods represent the most appropriate solution in this context. As we will see, it represents a trade-off between the capacity maximization, the convergence rate and the complexity.

Nevertheless, the proposed algorithm contains three embedded functions that need to be shown in explicit mathematical forms: 1) \( \text{find } \alpha_k \), 2) \( \text{projection(} Q \text{) } \) and 3) \( \text{find(} F_i(k+1) \text{)} \). In the following, we examine these three functions in details.

A. \( \text{find } \alpha_k \)

For determining \( \alpha_k \), we adopt the back tracking line search (also called Armijo rules) due to its simplicity in implementation. Herein, we do not suggest that this method is very accurate compared to the exact line search method. However, we believe that the value of \( \alpha_k \) will affect all the compared algorithms, similarly. According to this method, we choose fixed values of \( a \in [0, 0.1] \), \( b \in [0, 1] \) and \( t_0 \in [0, 1] \) and we find \( t \) according to the incremental procedure presented in the Algorithm 2.

B. \( \text{find(} F_i(k+1) \text{)} \)

In this work, we investigate the DFP and the SSQN quasi-Newton methods. According to these methods the inverse of

---

Algorithm 1: Distributed optimization (at the user \( i \))

*Initialization*:
- \( Q_i(0), F_i(0); \)
- \( k = 0; \)
- \( g_i(0) = \nabla Q_i C(Q_i(0),...,Q_i(0),...,Q_N(0)); \)

*Main*:
- While \( Q_i(k) - Q_i(k-1) - \epsilon \)
- \( d = F_i(k).\text{vec}(g_i(k)); \)
- \( Q'' = Q_i(k) + \text{mat}(d); \)
- \( Q' = \text{projection}(Q' \ onto \ the \ constraint's \ space); \)
- \( \text{find } \alpha_k; \)
- \( p_i(k) = \alpha_k.d; \)
- \( Q_i(k+1) = Q_i(k) + \alpha_k(Q'' - Q_i(k)); \)
- \( \text{broadcasting of } Q_i(k+1); \)
- \( g_i(k+1) = \nabla Q_i C(Q_i(k+1),...,Q_i(k+1),Q_{i+1}(k),...,Q_N(k)); \)
- \( q = \text{vec}(g_i(k+1) + g_i(k)); \)
- \( \text{find } (F_i(k+1)); \)
- \( k = k + 1; \)
the Hessian matrix can be computed iteratively according to the equation (6).
\[
\begin{cases}
    F^DFF_i(k + 1) = F_i(k) + v_1 - v_2 \\
    F^{SSQN}_i(k + 1) = (F_i(k) - v_2) c_1 + v_1
\end{cases}
\]
In which, \(c_1 = p^H_i(k)q > 0\) when \(\alpha_k\) is chosen appropriately, \(c_2 = q^H F_i(k)q\), \(v_1 = p_i(k)p^H_i(k)/c_1\) and \(v_2 = F_i(k)qq^H F_i(k)/c_2\).

C. projection(\(Q^\prime\))

From the subject function of (2), we know that the space of feasible solutions can be defined by the set \(S\) of positive semidefinite (PSD) matrices having unit trace. Then, the problem is how to project the matrix \(Q^\prime\) onto \(S\). For the sake of simplicity, let us introduce the concept of Hermitian vector. Assume now that \(b = vec(A)\) where \(A\) is a Hermitian matrix, then \(b\) is called a Hermitian vector. From this definition we have the following property:

\[\forall m, n \in [1, M] \quad b_{(m-1)M+n} = b^*_{(n-1)M+m}\]

For notation simplicity, we will refer to \(b_{(m-1)M+n}\) by \(b_{mn}\).

In the following, we give a theorem in order to demonstrate that \(Q\) is a Hermitian matrix and therefore the projection problem can be reduced to how to project a Hermitian matrix onto the set \(S\).

Theorem. The inverse of the Hessian matrix has the conjugacy property when interchanging the index of the column and the line simultaneously. That is \(F\) has the following property:

\[F^*_{mn,m'n'} = F^*_{nm,n'm'} \quad \forall m, n, m', n' \in [1, M]\]

Proof. The demonstration will be conducted recursively. Assume that this theorem is true for \(F_i(k) = A\) and demonstrate it for \(F_i(k + 1) = B\). Now we have \(A_{mn,m'n'} = A^*_{nm,n'm'}\). Note that, \(F_i(0)\) can be initialized appropriately, in order to verify the current theorem.

Mainly, the updating formula for the inverse of the Hessian considered in a previous section is based on three matrices: \(A, T = pp^H\) and \(R = Aqq^H A\). Now, if we prove that the last two matrices have the conjugacy property, then so for \(B\).

First we demonstrate that if \(b = vec(A)\), then \(c = Ab\) is a Hermitian vector. Herein, we have to demonstrate that \(c_{mn} = c^*_{nm} \forall m, n \in [1, M]\). Starting from the left side, we know that \(c_{mn} = A_{mn}b = (A_{mn})^*b\) where \(A_{mn}\) is the \((m-1)M + n\) line of the matrix \(A\). It follows directly that \(c_{mn} = \sum_{m'=1}^{M} \sum_{n'=1}^{M} A_{mn,m'n'}b_{m'n'} = \sum_{m'=1}^{M} \sum_{n'=1}^{M} A^*_{nm,n'm'}b^*_{m'n'} = c^*_{nm}\) where we used the fact that \(A\) has the conjugacy property as assumed before and \(b\) is a Hermitian vector. This latter property can be derived directly from the analytical form of the gradient matrix given in [2] and [6].

From the algorithm 1, we have that \(p = \alpha_k A A\) where \(A\) is a Hermitian vector(as shown above). That is \(p_{mn} = p^*_{mn}\). Thus, \(T_{mn,m'n'} = (pp^H)_{mn,m'n'} = p_{mn}p^*_{m'n'} = p_{nm}p^*_{n'm'} = ((pp^H)_{mn,m'n'})^* = T^*_{n'm',m'}\).

From [5], we recognize that \(A\) is a positive semidefinite matrix. Then we have \(A = A^H\) and \(R\) can be written as \(uu^H\) where \(u = Aq\), which has exactly the same form as \(T\) and therefore the demonstration will be the same.

Therefore, by summing the three components of \(B\), we have:

\[B_{mn,m'n'} = B^*_{n'm',m'} \quad \forall m, n, m', n' \in [1, M]\]

Consequently, \(d = F_i A\) is a Hermitian vector(same demonstration as \(c\) and finally \(Q\) is a Hermitian matrix.

As mentioned before, the problem now is how to project the Hermitian matrix \(Q^\prime\) onto the set \(S\). By using the Frobenius norm as the matrix distance criterion, it was shown that adjusting the eigenvalues appropriately and keeping the same eigenvectors solves for the projection problem. To be clearer, let \(Q' = V\Lambda V^H\) the eigenvalue decomposition of \(Q^\prime\). Therefore, to satisfy the constant power constraint we need to find \(\mu\) such that \(tr(\Lambda - \mu I)^{-1} = 1\). Once \(\mu\) is found \(Q^\prime\) can be constructed as follows: \(Q'' = V(\Lambda - \mu I)V^H\).

V. SIMULATION RESULTS

An extensive set of simulations was carried out in order to compare the performance of the four aforementioned algorithms: GP, CGP and our proposed algorithms, namely, DFP and SSQN. For fairness issues in our comparison, we plot in each figure: 1) the achievable per-user capacity, which stands for the local optimum in our problem. 2) The convergence rate represented by the number of iterations to reach this local optimum. Moreover and for more comparison fairness, we use the same common parameter used in [2] such as the symmetric rate represented by the number of iterations to reach this local optimum. For fairness issues in our comparison, we plot in each figure: 1) the achievable per-user capacity, which stands for the local optimum in our problem. 2) The convergence rate represented by the number of iterations to reach this local optimum. Moreover and for more comparison fairness, we use the same common parameter used in [2] such as the symmetric case where the SNR and the INR values are the same for all users. We note that, our results are averaged on 350 randomly generated channel matrices. For more simplicity, we use fixed number of antenna elements at each node in all the simulations (\(M = 2\)).

As a first result, we show in the figure 1, the performance with respect to the number of users. From the per-user capacity point of view, we notice that the four compared algorithms achieve almost the same performances. However, the DFP and the SSQN performs much better than the others in terms of the INR value. For the sake of clarity, we notice that the four compared algorithms achieve almost the same performances. However, the DFP and the SSQN performs much better than the others in terms of the INR value. For the sake of comparison fairness, we use the same common parameter used in [2] such as the symmetric case where the SNR and the INR values are the same for all users. We note that, our results are averaged on 350 randomly generated channel matrices. For more simplicity, we use fixed number of antenna elements at each node in all the simulations (\(M = 2\)).
given in [2] in terms of capacity and number of iterations. Recall that, the number of iterations of the GP method is less than 30 almost the time when the symmetric configuration is adopted (as suggested by the authors). However the DFP and the SSQN achieve a super linear convergence rate by reaching the local optimum in no more than 7 iterations, alleviating by that the amount of feedback information fourfold.

In the figure 2, the performances versus the INR values for a fixed SNR value are depicted. As shown in this figure, the convergence rate of the DFP and the SSQN methods is the best among the others. Basically, we observe that the convergence rate is independent from the interference level. Both the DFP and the SSQN reach the local optimum with less than 6 iterations. However, the GP and the CGP algorithms converge more quickly when low interference level is presented. Whereas, they shows a poor convergence rate when the interferences becomes strong. By comparing the proposed method and the old methods, we can get an improvement approximately from 200% to 400% on the convergence rate.

From the per-user capacity perspective, the simulations show that a small gap is presented between the proposed method and the old method. This gap is negligible in low and moderate interference environment. Whereas, when strong interferences are presented a small degradation on the DFP and the SSQN can be noticed. However, this degradation comes at the cost of the significant gain in the convergence rate that can be perceived.

Generally speaking, the performances of DFP and SSQN are much better then that of GP and CGP. In low interference environment the proposed algorithms enjoys a provable and fast convergence. However, in strong interference where the old algorithm shows a poorer convergence rate, a slight sacrifice on the capacity lead to higher convergence rate, which is the appropriate case for MIMO ad hoc networks.

VI. CONCLUSION

In this paper, the capacity optimization problem for MIMO ad hoc networks is emphasized. A novel, fast, cooperative and distributed algorithm is proposed in order to give an optimum solution without inundating the system by the feedback information. Our proposition is based on the quasi-Newton methods for solving non-linear optimization problems. Compared to other algorithms in this context and through an extensive simulation set, our algorithm presents the better convergence rate and enjoys a provable and satisfactory convergence quality.

REFERENCES