A FAST ALGORITHM FOR BEAMFORMING PROBLEMS IN DISTRIBUTED COMMUNICATION OF RELAY NETWORKS

Cong Sun, Yaxiang Yuan

State Key Lab. of Scientific and Engineering Computing, ICMSEC, AMSS,
Chinese Academy of Sciences, Beijing, 100190, China

ABSTRACT

A sequential quadratic programming (SQP) method is proposed to solve the distributed beamforming problem in multiple relay networks. The problem is formulated as the minimization of the total relay transmit power, subject to individual signal-to-interference-and-noise ratio constraints at each receiver, which is a nonconvex quadratic constraint quadratic programming. Rather than solving its semi-definite programming (SDP) relaxation, we apply the SQP method to solve its tightened form to replace its inequality constraints with equalities. Its global convergence is guaranteed. Simulations show that it not only runs much faster, but also performs as good as SDP for calculation results.

***Index Terms*—** relay networks, distributed beamforming, semi-definite programming, sequential quadratic programming

1. INTRODUCTION

Recently, multiuser cooperation diversity has been used to overcome the transmission loss due to poor channel conditions or severe signal interference. Such strategy allows users to act as relays of other users when they are free, thus to improve data rate and capacity with limited resources. There are several schemes to implement[1], in which amplify-and-forward (AF) is especially in hot research because of its simplicity.

A lot of work has been done on AF beamforming relay networks[2-7]. [2] and [3] discussed the problems in one source-destination pair model and brought in SDP method besides offering some analytical answers. Multiple peer-to-peer communication models with relays were considered in [5], [6] and [7]. To solve the proposed power minimization problem, [5] and [6] transform the quadratic constraint quadratic programming (QCQP) to an equal SDP, and solve it with primal and dual relaxation respectively. Based on their work, [7] proposed a new algorithm to relax the problem to a second order cone programming (SOCP) and solved it with low complexity while its calculation results are usually a few dBs more than those of SDP.

This work is partly supported by NSFC grant 10831006.

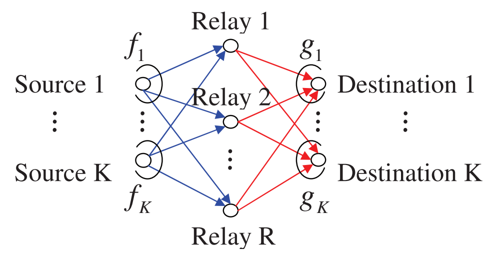

Fig.1 Multi-user relay network

In this paper, we construct a fast and efficient algorithm SQP for distributed peer-to-peer communication model in multiuser relay networks using AF protocol. Comparing with [5] and [7], its computational complexity is even smaller than that of SOCP thus much smaller than SDP. Our simulations show that its calculation result performs almost as good as SDP in terms of relay transmit power.

Notation: Lowercase and uppercase boldface represent for vectors and matrices respectively. We denote the complex conjugate, transpose and conjugate transpose as $(\cdot)^*$, $(\cdot)^T$ and $(\cdot)^H$ respectively.

2. PROBLEM MODEL

A relay network with K source-destination pairs and R relays(Fig.1) is considered. The kth source is assumed to transmit messages to the kth destination, $k = 1, 2, \dots, K$. All nodes are operated in a common frequency band and equipped with one antenna. No direct link is assumed. Also, we assume that the noise in the network is spatially white and all the noise and the transmit signals are mutually statistically independent.

Here we consider a two stage AF protocol. We only express one transmission process, thus omit the time index. In the first stage, the kth source transmits $\sqrt{P_k}s_k$ where P_k is its maximal transmit power and $E(|s_k|^2) = 1$. $\mathbf{r} = \sum_{k=1}^{K} \mathbf{f}_k \sqrt{P_k} s_k + \boldsymbol{\eta}$ is the receive vector of signals of the R relays, where $\mathbf{f}_k$ is the channels between the kth source and the R relays, and $\eta_p \sim N(0, \sigma_2^2)$ is the pth relay noise.

In the second stage, relays transmit $\mathbf{t} = \mathbf{W}^H\mathbf{r}$ to the destinations while w_p is the beamforming scalar of the pth relay,

$\mathbf{w} = (w_1, w_2, \ldots, w_R)^T$ and $\mathbf{W} = diag(\mathbf{w})$. The kth destination gets

$$y_k = \mathbf{g}_k^T \mathbf{t} + z_k = \sqrt{P_k}\mathbf{g}_k^T \mathbf{W}^H \mathbf{f}_k + \sum_{j=1, j\neq k}^{K} \sqrt{P_j}\mathbf{g}_k^T \mathbf{W}^H \mathbf{f}_j + (\mathbf{g}_k^T \mathbf{W}^H \boldsymbol{\eta} + z_k). \tag{1}$$

Here $z_k \sim N(0, \sigma_1^2)$ is the kth destination noise, and $\mathbf{g}_k$ the channels between relays and the kth destination. The three parts of (1) represent the desired signal, the interference and the noise components of the kth destination, and let P_S^k, P_I^k and P_N^k be their power, respectively.

Our aim is to minimize the relays' transmit power while keeping the destinations' quality of service (QoS) above certain predefined thresholds. Here we use signal-to-interference-and-noise ratio (SINR) as a measure of QoS. Thus we propose the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{C}^R} \quad & P_T \\ s.t. \quad & \text{SINR}_k = \frac{P_S^k}{P_I^k + P_N^k} \geq \gamma_k, \quad k = 1, 2, \ldots, K. \end{aligned} \tag{2}$$

The relay transmit power is expressed as $P_T = E(\mathbf{t}^H\mathbf{t}) = E(Tr(\mathbf{W}^H\mathbf{r}\mathbf{r}^H\mathbf{W})) = Tr(\mathbf{W}^H E(\mathbf{r}\mathbf{r}^H)\mathbf{W})$. Denote $R_s \triangleq E(\mathbf{r}\mathbf{r}^H)$, then $P_T = Tr(\mathbf{W}^H E(\mathbf{r}\mathbf{r}^H)\mathbf{W}) = \mathbf{w}^H\mathbf{D}\mathbf{w}$, where $\mathbf{D} = diag([\mathbf{R}_s]_{11}, [\mathbf{R}_s]_{22}, \ldots, [\mathbf{R}_s]_{RR})$ is a diagonal matrix, and $\mathbf{R}_s = E(\mathbf{r}\mathbf{r}^H) = \sum_{k=1}^K P_k E_{\mathbf{f}}(\mathbf{f}_k\mathbf{f}_k^H) + \sigma_2^2 I$.

Also we have

$$\begin{aligned} P_S^k &= E(|\sqrt{P_k}\mathbf{g}_k^T\mathbf{W}^H\mathbf{f}_k s_k|^2) = E(P_k\mathbf{g}_k^T\mathbf{W}^H\mathbf{f}_k\mathbf{f}_k^H\mathbf{W}\mathbf{g}_k^*) \\ &= E(P_k\mathbf{w}^H(\mathbf{g}_k \odot \mathbf{f}_k)(\mathbf{f}_k^H \odot \mathbf{g}_k^H)\mathbf{w}) = \mathbf{w}^H\mathbf{L}_k\mathbf{w}, \end{aligned}$$

where $\mathbf{l}_k \triangleq \mathbf{g}_k \odot \mathbf{f}_k = (f_{1k}g_{1k}, f_{2k}g_{2k}, \ldots, f_{Rk}g_{Rk})^T$ as their Schur product and $\mathbf{L}_k \triangleq E_{\mathbf{f},\mathbf{g}}(P_k\mathbf{l}_k\mathbf{l}_k^H)$.

Similarly, define $\mathbf{l}_k^j \triangleq \mathbf{g}_k \odot \mathbf{f}_j$ and

$$\mathbf{M}_k \triangleq E_{\mathbf{f},\mathbf{g}}(\sum_{j=1,j\neq k}^{K} P_j\mathbf{l}_k^j(\mathbf{l}_k^j)^H),$$

in simulations $\mathbf{L}_k$ and $\mathbf{M}_k$ is calculated directly without expectation assuming that relays and destinations require all channel coefficients. Then the interference power at the kth destination including two stages is given by

$$\begin{aligned} P_I^k &= E(|\sum_{j=1,j\neq k}^{K} \sqrt{P_j}\mathbf{g}_k^T\mathbf{W}^H\mathbf{f}_j s_j|^2) \\ &= E(\mathbf{w}^H(P_j \sum_{j=1,j\neq k}^{K} (\mathbf{g}_k \odot \mathbf{f}_j)(\mathbf{f}_j^H \odot \mathbf{g}_k^H))\mathbf{w}) = \mathbf{w}^H\mathbf{M}_k\mathbf{w}. \end{aligned}$$

With the assumptions we obtain the noise power

$$\begin{aligned} P_N^k &= E(\mathbf{g}_k^T\mathbf{W}^H\boldsymbol{\eta}\boldsymbol{\eta}^H\mathbf{W}\mathbf{g}_k^*) + \sigma_1^2 \\ &= \sigma_2^2 tr(\mathbf{W}^H\mathbf{W}E(\mathbf{g}_k^*\mathbf{g}_k^T)) + \sigma_1^2 = \mathbf{w}^H\mathbf{D}_k\mathbf{w} + \sigma_1^2, \end{aligned}$$

where $\mathbf{D}_k \triangleq \sigma_2^2 diag([\mathbf{R}_g^k]_{11}, [\mathbf{R}_g^k]_{22}, \ldots, [\mathbf{R}_g^k]_{RR})$ is a diagonal matrix and $\mathbf{R}_g^k = E_{\mathbf{g}}(\mathbf{g}_k\mathbf{g}_k^H)$.

Replacing P_T, P_S^k, P_I^k and P_N^k by their expression into (2), we can obtain a quadratic constraint quadratic programming as follows:

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{C}^R} \quad & \mathbf{w}^H\mathbf{D}\mathbf{w} \\ s.t. \quad & \mathbf{w}^H\mathbf{G}_k\mathbf{w} + c_k \leq 0, \quad k = 1, 2, \ldots, K, \end{aligned} \tag{3}$$

where $\mathbf{G}_k = \gamma_k(\mathbf{M}_k + \mathbf{D}_k) - \mathbf{L}_k$ and $c_k = \gamma_k\sigma_1^2$.

Here we consider that the channel obeys the complex Gaussian distribution. $\mathbf{G}_k$ are usually indefinite, so the optimization problem is nonconvex generally. Such problems can be solved by SDP relaxation for almost optimal solution[5, 6] or by SOCP for sub-optimal solution[7]. Although SOCP technique overcomes the disadvantage of SDP of high computational complexity, it suffers from a few dBs higher calculation results than SDP. To reduce the computational complexity and achieve good performances, we propose the following sequential quadratic programming algorithm.

3. SEQUENTIAL QUADRATIC PROGRAMMING

3.1. Basic Framework

In the above we have claimed that our problem can be formulated as a QCQP. The basic idea of sequential quadratic programming is to solve a series of quadratic programming subproblems to update the iterative point and corresponding Lagrange multipliers together. As its essential idea is Newton method, it achieves superlinear convergence rate under certain conditions. Noticing that the optimal solution is obtained at the equalities in almost all problems, we tighten the inequalities to equalities in constraints to speed up. Thus we try to solve a transformed QCQP with equalities in real domain:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^{2R}} \quad & f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^H\hat{\mathbf{G}}_0\mathbf{x} \\ s.t. \quad & C_k(\mathbf{x}) = \frac{1}{2}\mathbf{x}^H\hat{\mathbf{G}}_k\mathbf{x} + c_k = 0, \quad k = 1, 2, \ldots, K \end{aligned} \tag{4}$$

where $\mathbf{x} = (re(\mathbf{w}^T), im(\mathbf{w}^T))^T, \hat{\mathbf{G}}_0 = diag(2\mathbf{D}, 2\mathbf{D})$, $\hat{\mathbf{G}}_k = \begin{pmatrix} 2re(\mathbf{G}_k) & -2im(\mathbf{G}_k) \\ 2im(\mathbf{G}_k) & 2re(\mathbf{G}_k) \end{pmatrix}$.

For this type of equality QCQP, we propose a SQP algorithm, which framework is as follows:

SQP algorithm.

Step 1. Given any initial point $\mathbf{x}_0$ and Lagrange multiplier $\boldsymbol{\lambda}^0 > 0, i = 0$.

Step 2. Record the best point $\mathbf{x}_{best} = arg\min_{j=0}^{i} P(\mathbf{x}_j, u)$, where $P(\mathbf{x}, u)$ is the merit function and explained below. Solve

$$\begin{aligned} \min_{\mathbf{d}\in\mathbb{R}^{2R}} \quad & \frac{1}{2}\mathbf{d}^T\mathbf{B}_i\mathbf{d} + \tilde{\mathbf{g}}_i^T\mathbf{d} \\ s.t. \quad & \bar{\mathbf{C}}^i(\mathbf{d}) = \mathbf{A}^T\mathbf{d} + \mathbf{C} = 0, \end{aligned} \tag{5}$$

where the objective function is the Lagrange function of (4): $\mathbf{B}_i = \hat{\mathbf{G}}_0 + \sum_{k=1}^{K} \lambda_k^i \hat{\mathbf{G}}_k$, $\tilde{\mathbf{g}}_i = \hat{\mathbf{G}}_0 \mathbf{x}_i$. $\bar{\mathbf{C}}^i$ are the linear Taylor expansions of the constraints in (4). The *k*th column of $\mathbf{A}$, $\mathbf{A}_k = \hat{\mathbf{G}}_k \mathbf{x}_i$. And C_k means $C_k(\mathbf{x}_i)$.

Step 3. Get the search direction $\mathbf{d}_i$ and its corresponding Lagrange multipliers $\tilde{\boldsymbol{\lambda}}$ from Step 2. If $\|\mathbf{d}_i\| \leq \epsilon$, stop and output $\mathbf{x}_i$; else update $\boldsymbol{\lambda}^{i+1}$ and the merit function parameter u. If $\|\boldsymbol{\lambda}^{i+1}\|_\infty > \bar{\lambda}$ or we have gone for 5 forced steps, stop and claim infeasibility.

Step 4. If $\mathbf{x}_{best}$ has not been updated in n iterations, let $\mathbf{x}_i = \mathbf{x}_{best}$, find proper stepsize along the corresponding direction $\mathbf{d}_i = \mathbf{d}_{best}$ for a forced step; else find proper stepsize along $\mathbf{d}_i$ for a common step. $i \leftarrow i+1$, go back to Step 2.

3.2. Further explanation

In this part we provide some details for SQP algorithm.

3.2.1. Subproblem

Since the subproblem (5) itself may be nonconvex, our aim is to find a proper iteration direction through it. Suppose $\mathbf{A} = \mathbf{H}\begin{pmatrix} \mathbf{U} \\ \mathbf{0} \end{pmatrix} = \mathbf{YU}$ is its QR factorization, where $\mathbf{H} = (\mathbf{Y}, \mathbf{Z})$ is a $2R \times 2R$ orthogonal matrix and $\mathbf{U}$ is a $(2R-K) \times 2R$ upper triangle matrix. The solution of (5) can be written as

$$\mathbf{d}=\mathbf{Ydy}+\mathbf{Zdz}. \tag{6}$$

Together with $\mathbf{A}^T\mathbf{d} + \mathbf{C} = \mathbf{0}$ we have $\mathbf{dy} = -\mathbf{U}^{-T}\mathbf{C}$. To solve $\mathbf{dz}$, take (6) into (5) and we get a quadratic programming without constraints:

$$\min_{\mathbf{dz}\in\mathbb{R}^{2R-K}} \frac{1}{2}\mathbf{dz}^T\mathbf{Z}^T\mathbf{B}_i\mathbf{Zdz} + (\mathbf{B}_i\mathbf{Ydy} + \tilde{\mathbf{g}}_i)^T\mathbf{Zdz} \tag{7}$$

We can get an analytical solution at which point the function value of (7) is reduced. Through Schur decomposition $\mathbf{Z}^T\mathbf{B}_i\mathbf{Z} = \mathbf{Q}\tilde{\mathbf{D}}\mathbf{Q}^T$, where $\mathbf{Q}$ is an orthogonal matrix and $\tilde{\mathbf{D}}$ is a diagonal matrix with the eigenvalues of $\mathbf{Z}^T\mathbf{B}_i\mathbf{Z}$ on the diagonal, we have $\mathbf{dz} = (dz_1, dz_2, \ldots, dz_{2R-K})^T$, where

$$dz_j = \begin{cases} -\frac{(\mathbf{Q}^T\mathbf{Z}^T(\mathbf{B}_i\mathbf{Ydy}+\tilde{\mathbf{g}}_i))_j}{\tilde{D}_{jj}} & \tilde{D}_{jj} > 0 \\ -(\mathbf{Q}^T\mathbf{Z}^T(\mathbf{B}_i\mathbf{Ydy}+\tilde{\mathbf{g}}_i))_j & \tilde{D}_{jj} \leq 0 \end{cases} \quad j = 1, 2, \ldots, 2R-K \tag{8}$$

So we can take $\mathbf{dy}$ and $\mathbf{dz}$ into (6) to get the solution $\mathbf{d}_i$ of subproblem (5). With $\mathbf{d}_i$, we can solve $\tilde{\boldsymbol{\lambda}}$ through equations $\mathbf{B}_i\mathbf{d}_i + \tilde{\mathbf{g}}_i + \mathbf{A}\tilde{\boldsymbol{\lambda}} = 0$ according to its KKT condition. Computationally we let $\tilde{\boldsymbol{\lambda}} = -\mathbf{U}^{-1}\mathbf{Y}^T(\mathbf{W}\mathbf{d}_i + \tilde{\mathbf{g}}_i)$, which needs Schur decomposition of $\mathbf{B}_i$:

$$\mathbf{B}_i = \sum_{r=1}^{2R} \sigma_r \mathbf{v}_r \mathbf{v}_r^T, \mathbf{W} = \sum_{r=1,\sigma_r>0}^{2R} \sigma_r \mathbf{v}_r \mathbf{v}_r^T.$$

3.2.2. Merit function

To balance the function value $f(\mathbf{x})$ and constraint violation $\|\mathbf{C}(\mathbf{x})\|_1$, we bring in merit function $P(\mathbf{x}, u) = f(\mathbf{x}) + u\|\mathbf{C}(\mathbf{x})\|_1$. Here we require $u > \|\boldsymbol{\lambda}\|_\infty$ for convergence condition. Also, the value of merit function with linearized constraints $\bar{\mathbf{C}}^i(\mathbf{d})$ should be guaranteed to reduce, which means $(f(\mathbf{x}_i) + u\|\bar{\mathbf{C}}^i(\mathbf{0})\|_1) - (f(\mathbf{x}_i + \mathbf{d}_i) + u\|\bar{\mathbf{C}}^i(\mathbf{d}_i)\|_1) \geq \frac{1}{2}u\|\mathbf{C}(\mathbf{x}_i)\|_1$, that is, $u\|\mathbf{C}(\mathbf{x}_i)\|_1 \geq 2(f(\mathbf{x}_i+\mathbf{d}_i) - f(\mathbf{x}_i))$. The second condition forces u to be large enough to guarantee the reduction of constraint violation.

Quite related is to choose stepsize. There are several differences between forced and common step. Firstly, forced step starts back from the best point $\mathbf{x}_{best}$ while the common step goes from the current iteration point $\mathbf{x}_i$. Secondly, we only try a second order correction step $\mathbf{p}_i$ in common step to overcome Maratos effect which requires low computational complexity[9]. Let $\Delta P = P(\mathbf{x}_i + \mathbf{d}_i, u) - P(\mathbf{x}_i, u)$, if $P(\mathbf{x}_i + \mathbf{d}_i + \mathbf{p}_i, u) < P(\mathbf{x}_i, u) + \eta\Delta P$, we have $\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{d}_i + \mathbf{p}_i$ and finish the common step. Thirdly, the two kinds of steps have different stopping criterion. The forced step requires $P(\mathbf{x}_{best} + \alpha\mathbf{d}_{best}, u) \leq (1-0.01\alpha)P(\mathbf{x}_{best}) + 0.01\alpha f(\mathbf{x}_{best} + \mathbf{d}_{best})$ while the common step just requires $P(\mathbf{x}_i + \alpha\mathbf{d}_i, u) < P(\mathbf{x}_i, u) + \alpha\eta\Delta P$[9]. The former is used to guarantee convergence while the latter is to speed up the algorithm.

In both forced and common step, the stepsize α is searched by backtracking to be large enough.

3.2.3. Update strategy for $\boldsymbol{\lambda}$

Since the problem we want to solve is actually with inequalities, we expect the corresponding Lagrange multiplier $\boldsymbol{\lambda} \geq 0$ in the iterations. Also, we wish it not to change too fast, which coincides with the situation when $\mathbf{x}_i$ approaches the optimal point $\mathbf{x}^*$. Thus we update $\boldsymbol{\lambda}$ as below:

Update Strategy (for $k = 1, 2, \ldots, K$)

Step 1. If $C_k(\mathbf{x}_i) > 0$, go to step 2, else go to step 3;

Step 2. If $C_k(\mathbf{x}_i + \mathbf{d}_i) - C_k(\mathbf{x}_i) > f(\mathbf{x}_i + \mathbf{d}_i) - f(\mathbf{x}_i)$, $\lambda_k^{i+1} = \min\{\tilde{\lambda}_k, 2\lambda_k^i\}$, else $\lambda_k^{i+1} = \max\{\tilde{\lambda}_k, \frac{1}{2}\lambda_k^i\}$;

Step 3. If $f(\mathbf{x}_i + \mathbf{d}_i) > f(\mathbf{x}_i)$, $\lambda_k^{i+1} = \max\{\tilde{\lambda}_k, \frac{1}{2}\lambda_k^i\}$, else $\lambda_k^{i+1} = \min\{\tilde{\lambda}_k, 2\lambda_k^i\}$.

3.3. Further Analysis

From above, we can notice that the main computation of one iteration in SQP focuses on solving the subproblem (5), which mainly includes two Schur decompositions and one QR factorization. So the computational complexity is $O(R^3)$ in the hypothesis that $R \gg K$. This is smaller than SOCP's $O(R^3K^{1.5})$ and SDP's $O(R^6)$.

Since we have guaranteed the reduction of merit function in $n(<\infty)$ steps, we can get the global convergence result of our SQP algorithm similar to that of classical SQP method[8]. The specific theorem is omitted because of the limited space.

Although we can only guarantee a KKT point to be solved, in practice we usually get the optimal or almost optimal solution, which will be seen in simulations.

4. SIMULATIONS

In our numerical simulations, the transmit signal power is the same as the noise variation σ_1^2 and σ_2^2, both of which are set to be 0dB. All channel coefficients are supposed to be known at the relays and destinations. So the beamforming vector can be obtained optimally. In each simulation, components of $\mathbf{f}_k$ and $\mathbf{g}_k$ are generated as i.i.d complex Gaussian random variables with variations $\sigma_f^2 = \sigma_g^2 = 10dB$. All destination SINRs are required to be above the same threshold, $\gamma_k = \gamma, k = 1, 2, \ldots, K$. At each target SINR, channel coefficients are randomly generated for 100 times to calculate the normalized average relay transmit power. We consider 20 relays ($R = 20$) and compare the performances of SDP relaxation[5] and SQP at $K = 2, 3, 4$. Each problem is solved by both algorithms. In SQP, $\bar{\lambda} = 10^4, \eta = 0.1, n = 10$.

Fig.2 shows almost the same performance of SQP compared with SDP. In Fig.3, we compared the average running time including solving feasible ones and judging infeasibility. SQP outperforms SDP much when γ is moderate thus almost 100% problems are feasible. However when γ becomes too large that infeasible ones dominate in 100 examples, SQP is beaten by SDP. Our algorithm still needs improving for judging infeasibility.

In conclusion, simulations show that our proposed SQP algorithm runs much faster than SDP generally and acquire almost the same results. Thus, SQP algorithm is a better choice if we have a problem which is likely to be feasible.

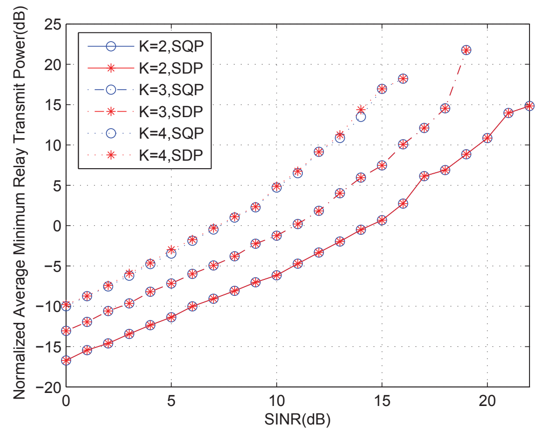

Fig.2 Total relay transmitted power versus minimal required SINR, $K = 2, 3, 4$

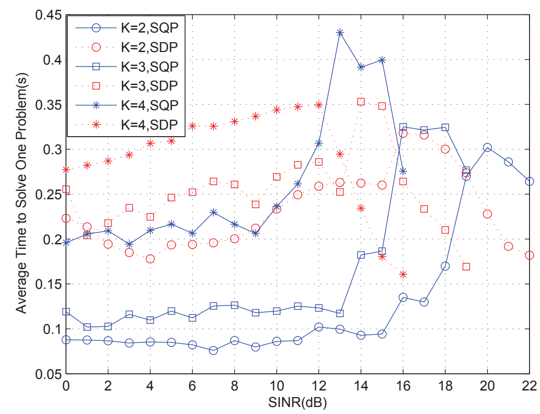

Fig.3 Average running time versus minimal required SINR, $K = 2, 3, 4$

5. REFERENCES

[1] J. N. Laneman, D. N. C. Tse and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," in *IEEE Trans. Inform. Theory*, vol. 50, pp. 3062-3080, Dec. 2004.

[2] Y. Jing and H. Jafarkhani, "Network beamforming using relays with perfect channel information," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Honolulu, HI, Apr. 2007, vol. 3.

[3] V. Havary-Nassab, S. Shahbazpanahi, A. Grami, and Z. -Q. Luo, "Distributed beamforming for relay networks based on second-order statistics of the channel state information," *IEEE Trans. Signal Process.*, vol. 56, no. 9, pp. 4306-4316, Sep. 2008.

[4] R. Zhang, C. C. Chai, and Y. C. Liang, "Joint Beamforming and Power Control for Multiantenna Relay Broadcast Channel With QoS Constraints," *IEEE Trans. Signal Process.*, vol.57, no.2, pp. 726-737, Feb. 2009.

[5] S. Fazeli-Dehkordy, S. Shahbazpanahi, and S. Gazor, "Multiple Peer-to-Peer Communications Using a network of Relays," *IEEE Trans. Signal Process.*, vol.57, no.8, pp. 3053-3062, Aug. 2009.

[6] D. H. N. Nguyen, H. H. Nguyen, H. D. Tuan, "Distributed Beamforming in Relay-Assisted Multiuser Communications," in *Proc. IEEE Int. Commun. Conf. (ICC)*, Dresden, Germany, Jun. 2009.

[7] H. -H. Chen, A. B. Gershman, S.Shahbazpanahi, "Distributed Peer-to-Peer Beamforming for Multiuser Relay Networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Taipei, Taiwan, Apr. 2009.

[8] Y.X. Yuan, *Computational Mehthod for Nonlinear Optimization (in Chinese)*. Beijing: Science Press, 2008.

[9] J. Nocedal, S. J. Wright, *Numerical Optimization*. Berlin: Springer, 1999.