www.ietdl.org

Published in IET Communications
Received on 9th July 2010
Revised on 10th October 2010
doi: 10.1049/iet-com.2010.0616

ISSN 1751-8628

Adaptive second-order control of transmitter power in wireless communication systems

C.-H. Wu   C.-A. Lin

Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu, Taiwan
E-mail: chwu.ece92g@nctu.edu.tw

Abstract: The authors consider transmitter power control of wireless communication systems. They propose an adaptive second-order distributed power control algorithm in which the relaxation factors are adaptively adjusted to improve the rate of convergence. The algorithm updates power using a weighted combination of the distributed power control algorithm and the second-order power control (SOPC) algorithm. Simulation results show performance improvement over distributed constrained power control and constrained SOPC.

1 Introduction

Transmitter power control has been considered as an effective means for resource allocation in wireless communication systems. In direct-sequence code-division multiple-access (DS-CDMA) systems, power control is indispensable for achieving the required level of quality of service (QoS), which is usually measured in terms of the signal-to-interference ratio (SIR). Over the past decade, many power control algorithms have been proposed to maintain QoS of communication systems. Utility maximisation under power constraints and minimum SIR requirement is discussed in [1–4] while power and rate control under outage constraints is discussed in [5]. The works in [6, 7] address time-varying link gain systems and propose algorithms to maintain SIR requirement. For quasi-stationary link gains, first-order algorithms, whose power updates use only current power level, for improving the rate of convergence over the distributed power control (DPC) algorithm in [8] are proposed in [9–11]; second-order algorithms, whose power updates used the current and the immediate past power levels, are proposed in [12] and compared with DPC.

In this paper, we consider a quasi-stationary link gain system and study a distributed power control algorithm aiming at fast convergence. Our work is motivated by that [12] in which a second-order power control (SOPC) algorithm derived from the successive over-relaxation (SOR) method [13] was proposed. The second-order algorithm uses power levels of current and previous steps, and can potentially achieve better convergence performance than that in [8] provided the relaxation factor is properly chosen. However, the non-adaptive choice of the relaxation factor in [12] may lead to a situation in which the powers of some users are adjusted unnecessarily low at the first few steps owing to large power levels of the previous step. In this situation, the required SIR of those users can no longer be supported, resulting in large transient outage probability and system performance deterioration. Hence, we propose an adaptive SOPC algorithm in which the relaxation factor is adaptively adjusted based on the ratio of DPC power level and the power level of the previous step. Simulation results show performance improvement in SIR convergence in the sense that the convergence to steady-state outage probability is faster.

This paper is organised as follows. In Section 2, we describe the system model and review the power control algorithms in [12] and [8]. In Section 3, we describe the proposed adaptive SOPC algorithm. Simulation results are given in Section 4. Section 5 is a brief conclusion.

2 Power control schemes

We consider a generic case where there are K active mobile station–base station pairs in a wireless communication network. In a cellular CDMA system, K users are assigned to a base station; in others the K pairs represent co-channel cells using the same frequency band. We consider uplink power control. The power received at the ith base station, y_i, is the sum of power from user i (the intended transmitter), interference power and noise

$$y_i = g_{ii}p_i + \sum_{j=1, j\neq i}^{K} g_{ij}p_j + u_i \tag{1}$$

where p_i is the power transmitted by user i, g_{ij} is the power link gain from user j to base i and u_i is the receiver noise power at base i. We assume that $g_{ij} \geq 0$ and $g_{ii} > 0$. The SIR at the ith base station, γ_i, is defined as

$$\gamma_i = \frac{g_{ii}p_i}{\sum_{j\neq i} g_{ij}p_j + u_i} \tag{2}$$

The goal of power control is to maintain a target SIR threshold for each user by adjusting the transmitter power p_i, that is, to achieve

$$\gamma_i = \gamma_i^{\mathrm{t}}, \quad i = 1, 2, \ldots, K \tag{3}$$

where γ_i^{t} is the target SIR for user i. If there are $p_i \geq 0$, $i = 1, \ldots, K$ such that (3) holds, the power control problem is said to be feasible, otherwise it is infeasible.

To analyse the problem, we define the normalised link gain $h_{ij} = \gamma_i^{\mathrm{t}} g_{ij}/g_{ii}$ for $j \neq i$ and the normalised noise power $\nu_i = \gamma_i^{\mathrm{t}} u_i/g_{ii}$, and rewrite (3) as

$$p_i = \sum_{j \neq i} h_{ij} p_j + \nu_i \tag{4}$$

In matrix form, (4) becomes

$$\boldsymbol{p} = \boldsymbol{H}\boldsymbol{p} + \boldsymbol{\nu} \tag{5}$$

where $\boldsymbol{p} = [p_1\ p_2\ \cdots\ p_K]^{\mathrm{T}} \in \mathbb{R}^K$, $\boldsymbol{\nu} = [\nu_1\ \nu_2\ \cdots\ \nu_K]^{\mathrm{T}} \in \mathbb{R}^K$, and $\boldsymbol{H} \in \mathbb{R}^{K\times K}$ is a non-negative matrix with zero diagonal entries and ijth entry h_{ij} for $i \neq j$. We assume that the problem is feasible, that is, there is a non-negative power vector $\boldsymbol{p}^* \geq 0$ that satisfies (5).

One way to solve (5) iteratively is to use the Jacobi iteration [14, p. 353]

$$\boldsymbol{p}(n+1) = \boldsymbol{H}\boldsymbol{p}(n) + \boldsymbol{\nu} \tag{6}$$

We know that a sufficient and necessary condition for stability and hence convergence is that all the eigenvalues of $\boldsymbol{H}$ lie inside the unit circle or equivalently the spectral radius of $\boldsymbol{H}$, $\rho(\boldsymbol{H})$, is strictly less than 1. Small $\rho(\boldsymbol{H})$ implies fast convergence and if $\rho(\boldsymbol{H}) \simeq 1$, the convergence can be very slow. The iteration (6) in its present form, however, cannot be implemented in practice, since to compute $\boldsymbol{p}(n+1)$ knowledge of the link gain matrix $\boldsymbol{H}$, the current power $\boldsymbol{p}(n)$ and the noise power $\boldsymbol{\nu}$ is required. The distributed power control algorithm proposed by Foschini and Maljanic [8] is basically an implementation (6) so that each user adjusts its power based only on information available locally. Rewriting (6) for user i, we have

$$p_i(n+1) = \sum_{j\neq i} h_{ij} p_j(n) + \nu_i = \gamma_i^{\mathrm{t}} \left(\sum_{j \neq i} \frac{g_{ij} p_j(n) + u_i}{g_{ii}} \right)$$

It follows that the DPC is

$$p_i(n+1) = \frac{\gamma_i^{\mathrm{t}}}{\gamma_i(n)} p_i(n), \quad i = 1, \ldots, K \tag{7}$$

Note the power $p_i(n+1)$ depends only on $\gamma_i(n)$, the SIR of user i at the nth instant, and $p_i(n)$, both are available to user i. It is proved in [8] that if the problem is feasible, then the algorithm (7) guarantees the convergence of transmitter power and SIR for each user.

Another way to solve (5) is to use the SOR method [13], which modifies (6) to include a relaxation parameter. Based on the SOR method, Jäntti and Kim [12] proposed the iteration

$$\boldsymbol{p}(n+1) = \omega(\boldsymbol{H}\boldsymbol{p}(n) + \boldsymbol{\nu}) + (1-\omega)\boldsymbol{p}(n-1) \tag{8}$$

which can be implemented as a distributed algorithm, that is, for user i, the power update becomes

$$p_i(n+1) = \omega \frac{\gamma_i^{\mathrm{t}}}{\gamma_i(n)} p_i(n) + (1-\omega) p_i(n-1) \tag{9}$$

Iteration (9) defines an SOPC algorithm since to compute $p_i(n+1)$ we need both $p_i(n)$ and $p_i(n-1)$ as well as the SIR $\gamma_i(n)$. It is shown that the optimal relaxation factor ω^* is given by Young [13] as

$$\omega^* = \frac{2}{1 + \sqrt{1 - \rho^2(\boldsymbol{H})}} \tag{10}$$

and for $\rho(\boldsymbol{H}) < 1$, $1 < \omega^* < 2$. However, since $\boldsymbol{H}$ is not available, there is no way of knowing the exact value of ω^*. Note also that (9) reduces to DPC when $\omega = 1$ and for an arbitrary ω, the convergence of (9) may be slower than that of DPC.

3 Adaptive SOPC

Since the optimal relaxation factor ω^* is unknown and cannot be determined from the available local information, it cannot be used in (9). In this section, we consider a modification of algorithm (9). Instead of using a fixed ω for all users, we allow each user to have an individual $\omega_i(n)$ which may vary from step to step. Rewrite the second-order power control (SOPC) algorithm (9) as

$$p_i(n+1) = \frac{\gamma_i^{\mathrm{t}}}{\gamma_i(n)} p_i(n) + \Delta\omega_i(n) \left[\frac{\gamma_i^{\mathrm{t}}}{\gamma_i(n)} p_i(n) - p_i(n-1) \right] \tag{11}$$

where $\Delta\omega_i(n) = \omega_i(n) - 1$. Since the optimal relaxation factor satisfies $1 < \omega^* < 2$, we confine each $\Delta\omega_i(n)$ to between 0 and 1. The first term on the right-hand side of (11) is the DPC update, and the second correction term is the weighted difference between the DPC update and the power of the previous step. The algorithm is completely specified by the choice of $\Delta\omega_i(n)$. If $\Delta\omega_i(n) = 0$, the algorithm reduces to DPC. The SOPC in [12], with $\Delta\omega_i(n) = 1/a^n$ for all i and for some $a > 1$, is non-adaptive in the sense that $\Delta\omega_i(n)$ is fixed a priori. If a is chosen large, the algorithm approaches rapidly to DPC. If $a > 1$ is chosen small then, as observed in [12], the power update in the first few steps may result in negative power levels (which are then set to the minimal transmitter power). As a result, a significant number of users may have SIR below the target level. This is likely to happen when $p_i(n-1)$ is large compared to $(\gamma_i^{\mathrm{t}}/\gamma_i(n))p_i(n)$. Note that to the power update (11), we need to add the minimal and maximal power constraint

$$p_{\min} \leq p_i(n) \leq p_{\max} \tag{12}$$

where $p_{\min} \geq 0$ and $p_{\max} > p_{\min}$ for practical implementation. The algorithms in (7) and in (11) with constraint (12) are called the distributed constrained power control (DCPC) algorithm and the constrained second-order power control (CSOPC) algorithm, respectively.

From (11), we see that the power update $p_i(n+1)$ may become negative if the difference in square brackets is negative and large (in magnitude) and $\Delta\omega_i(n)$ is not small. Although it will be adjusted to $p_{\min}$, this unnecessarily low

doi: 10.1049/iet-com.2010.0616

power level is then likely to decrease $\gamma_i(n+1)$ and cause outage. To discuss the problem, we define

$$A_i(n) = \frac{\gamma_i^t}{\gamma_i(n)} \frac{p_i(n)}{p_i(n-1)} \tag{13}$$

the ratio between the DPC update and the power level of the previous step. Since (13) can be rewritten as $A_i(n) = \gamma_i^t \eta_i(n)/(g_{ii}p_i(n-1))$, where $\eta_i(n) = \sum_{j\neq i} g_{ij}p_j(n) + u_i > 0$ is the interference of user i, we have $A_i(n) > 0$. The following proposition gives a sufficient and necessary condition under which $p_i(n+1) \leq 0$.

Proposition 1: Consider algorithm (11) with $\Delta\omega_i(n) \geq 0$. Suppose $p_i(n-1) > 0$. Then $p_i(n+1) \leq 0$ if and only if

(a) $A_i(n) < 1$ and
(b) $\Delta\omega_i(n) \geq A_i(n)/1 - A_i(n)$.

Proof: Since $p_i(n-1) > 0$, we divide (11) by $p_i(n-1)$ and obtain

$$\begin{aligned}\frac{p_i(n+1)}{p_i(n-1)} &= \frac{\gamma_i^t}{\gamma_i(n)} \frac{p_i(n)}{p_i(n-1)} + \Delta\omega_i(n)\left[\frac{\gamma_i^t}{\gamma_i(n)} \frac{p_i(n)}{p_i(n-1)} - 1\right] \\ &= A_i(n) + \Delta\omega_i(n)[A_i(n) - 1] \\ &= [A_i(n) - 1]\left[\frac{A_i(n)}{A_i(n) - 1} + \Delta\omega_i(n)\right]\end{aligned}$$

Since $\Delta\omega_i(n) \geq 0$ and $A_i(n) > 0$, it is seen that $p_i(n+1) \leq 0$ if and only if $A_i(n) - 1 < 0$ and $(A_i(n)/A_i(n) - 1) + \Delta\omega_i(n) \geq 0$. The result follows. □

By the choice $\Delta\omega_i(n) = 1/a^n$ in [12], $\Delta\omega_i(n)$ is large when n is small. Thus, from Proposition 1, the power may update to a negative level in the first few steps. The basic idea of the proposed adaptive algorithm is to prevent the power update in the first few steps from being unnecessarily low. From Proposition 1, this means that when $A_i(n) < 1$ or equivalently $(\gamma_i^t/\gamma_i(n))p_i(n) - p_i(n-1) < 0$, we must have

$$\Delta\omega_i(n) < \frac{A_i(n)}{1 - A_i(n)} \tag{14}$$

so that $p_i(n+1) > 0$. To satisfy (14), one choice of the relaxation factor is

$$\Delta\omega_i(n) = \min\{1, \varepsilon A_i(n)\} \tag{15}$$

where $0 \leq \varepsilon \leq 1$. By the adaptive adjustment in (15), we expect the performance of the algorithm to be better than that in [12] during the first few iterations since (a) or (b) in Proposition 1 is avoided. Moreover, if $\Delta\omega_i(n) = \varepsilon A_i(n)$, we can substitute (15) into (11) and obtain

$$p_i(n+1) = (1-\varepsilon)p_i(n-1)A_i(n) + \varepsilon p_i(n-1)A_i^2(n) \tag{16}$$

Note that if $\varepsilon = 0$, then from (15) we have $\Delta\omega_i(n) = 0$ and (16) reduces to DPC algorithm. Comparing algorithm (16) with the DPC algorithm, if $A_i(n) > 1$, then $A_i^2(n)p_i(n-1) > A_i(n)p_i(n-1)$ and thus $p_i(n+1) > (\gamma_i^t/\gamma_i(n))p_i(n)$. Also, we have $p_i(n+1) < (\gamma_i^t/\gamma_i(n))p_i(n)$ for $A_i(n) < 1$ and $p_i(n+1) = (\gamma_i^t/\gamma_i(n))p_i(n)$ for $A_i(n) = 1$. The parameter ε can be regarded as the weighting factor between DPC update and the term $A_i^2(n)p_i(n-1)$. For a large ε, the gap between $p_i(n+1)$ and $(\gamma_i^t/\gamma_i(n))p_i(n)$ becomes large if $A_i(n) \neq 1$.

The second-order algorithm CSOPC asymptotically approaches the first-order algorithm DCPC. For example, with $a = 1.5$, then in (11) the relaxation factor $\Delta\omega_i(n) = 1/a^n < 0.1$ for $n = 6$. So effectively the algorithm is of second order only in the first few iterations; this is to avoid the undesirable consequence of fast power convergence to $p_{\min}$. In the proposed adaptive algorithm, the power update switches to DCPC if a certain condition is satisfied. To discuss the condition, let us suppose $p_i(n) = p_{\min}$ and $\gamma_i(n) > \gamma_i^t$ at the nth instant; then we have $g_{ii}p_{\min}/\eta_i(n) > \gamma_i^t$ or, equivalently, $(g_{ii}/\gamma_i^t)p_{\min} > \eta_i(n)$. In addition, from (16), since $(\gamma_i^t/\gamma_i(n))p_{\min} \leq p_{\min}$, we obtain $p_i(n+1) < p_{\min}$, which is set to $p_{\min}$ because of the power constraint (12). This situation is undesirable in the sense that user i cannot decrease its transmitter power, which results in the increase of the interference $\eta_j(n+1)$, for $j \neq i$. Therefore we switch algorithm (16) to the DCPC algorithm before $\eta_i(n)$ reaching $(g_{ii}/\gamma_i^t)p_{\min}$ to slow down the power convergence to $p_{\min}$, that is, we set $\Delta\omega_i(n) = 0$ if $\eta_i(n) \leq \zeta$, where $\zeta > (g_{ii}/\gamma_i^t)p_{\min}$ is a positive value. Although we know that the users with large g_{ii} will converge to $p_{\min}$ faster than those with small g_{ii}, they have no link gain information about each other and we need to use an estimated value. One appropriate choice of the switching level is

$$\zeta = \beta \frac{\hat{g}}{\gamma_i^t} p_{\min} \tag{17}$$

where $\hat{g} = \sqrt{\bar{g}^2 + \mathrm{var}(g)}$ is estimated value of the large link gain with $\bar{g}$ denoted the expected link gain and $\mathrm{var}(g)$ denoted the link gain variance and β is a positive value. With proper choice of β, we can obtain good performance. Note that if β is large, the convergent behaviour of the algorithm, named adaptive second-order power control (ASOPC) in Fig. 1 is similar to that of the DCPC algorithm because of switching early.

Remark 1: The parameter ε in (15) determines the relative weighting of DPC update and the term $A_i^2(p)p_i(n-1)$. From the simulation example in Section 4, it is seen that $\varepsilon \in [0.4, 0.6]$ yields good performance.

Remark 2: From (17), the parameter ζ is determined by β. For a large β, the algorithm switches to DPC early. From the simulation examples, $\beta \in [0.3, 3]$ is a good choice.

4 Simulation results

In this section, we use examples to compare the proposed ASOPC algorithm with the DCPC algorithm and the CSOPC algorithm. The DCPC and CSOPC power updates are, respectively

$$p_i(n+1) = \min\left\{p_{\max},\ \max\left\{p_{\min},\ \frac{\gamma_i^t}{\gamma_i(n)}p_i(n)\right\}\right\}$$

and

$$p_i(n+1) = \min\left\{p_{\max},\ \max\left\{p_{\min},\ \left(1 + \frac{1}{a^n}\right)\frac{\gamma_i^t}{\gamma_i(n)}p_i(n) - \frac{1}{a^n}p_i(n-1)\right\}\right\}$$

We summarise our adaptive second-order power control (ASOPC) algorithm as follows:

- Initialisation: Given target SIR γ_i^t and initial power $p_i(0)$, $i = 1, \cdots, K$. Choose parameters $0 < \varepsilon < 1$ and ζ according to (17).
- For $n = 0$, measure $\gamma_i(0)$ and set

$$p_i(n+1) = \min\{p_{\max}, \max\{p_{\min}, \frac{\gamma_i^t}{\gamma_i(n)} p_i(n)\}\}$$

- For $n = 1, 2, 3, \cdots$

(i) Measure $\gamma_i(n)$. Compute $\eta_i(n) = g_{ii} p_i(n)/\gamma_i(n)$ and $A_i(n) = \gamma_i^t p_i(n)/(\gamma_i(n) p_i(n-1))$.

(ii) Set $\triangle\omega_i(n) = \begin{cases} 0 & , \text{ if } \eta_i(n) \le \zeta \\ \min\{1, \varepsilon A_i(n)\} & , \text{ if } \eta_i(n) > \zeta \end{cases}$

(iii) Compute power

$$p_i(n+1) = \min\{p_{\max}, \max\{p_{\min}, (\gamma_i^t/\gamma_i(n)) p_i(n) + \triangle\omega_i(n)[(\gamma_i^t/\gamma_i(n)) p_i(n) - p_i(n-1)]\}\}.$$

Fig. 1 *Adaptive second-order power control algorithm*

where $p_{\min}$ and $p_{\max}$ are, respectively, the minimum and maximum power that can be used by each user. In simulations, the target SIR for each user is chosen to be the same fixed value. In each example we compare the convergence of the SIR, which is measured in terms of the outage probability. The outage probability at each iteration is defined to be the number ratio of users whose SIR is below a required SIR, a value slightly lower than the target SIR. When the number of users is large, say a few hundreds, the ratio is close to the probability that the achieved SIR is below the required SIR.

4.1 Example 1

We consider the same IS-95 system example as in [12], in which the processing gain is set to 21 dB. The omnibase is assumed to locate at the centre of a hexagonal cell. In each simulation run, 190 mobiles uniformly distributed over the 19 hexagonal cells are generated. We assume that the minimum distance between any two base stations is $\sqrt{3}$ km. The link gain is modelled as $g_{ij} = s_{ij} \cdot d_{ij}^{-4}$, where s_{ij} has log-normal distribution with $E[s_{ij}] = 0$ and $\sigma_{s_{ij}} = 8$ dB and d_{ij} is the distance between base i and mobile j. The target SIR γ_i^t is set to 8 dB, the required SIR is set to 7 dB, the maximum power $p_{\max} = 1$ W, the minimum power $p_{\min} = 4 \times 10^{-6}$ W and the receiver noise is taken to be 10^{-12} W for all mobiles. We consider 11 000 independent instances and take the 'feasible' ones in each instance for the computation of the average outage probability. The initial power for each mobile is randomly chosen from the interval $[p_{\min}, 1]$.

Fig. 2 shows the the outage probability of ASOPC with $\varepsilon = 0.5$ and $\zeta = 0.01$, CSOPC with $a = 1.5$, and DCPC. The outage probability for the CSOPC algorithm has an initial surge and remains high for the first six steps. This is because the power level of a significant number of users are adjusted unnecessarily low. The ASOPC prevents this from happening at the first few steps with larger power adjustment steps compared to DCPC, thus yielding the lowest outage probability. After the initial surge, the outage probability of CSOPC drops rapidly and compares favourably with DCPC. In addition, the outage probability of ASOPC converges to 1.5×10^{-4}, which is regarded as the steady state of the outage probability, at the 15th iteration on average. CSOPC takes 30 iterations on average to reach the steady-state outage probability while DCPC needs more than 30 iterations.

There are two parameters ε and ζ in the ASOPC algorithm, one specifies the relative weighting between DPC update and the term $A_i^2(n)p_i(n-1)$, and the other specifies the condition under which to switch to DCPC. We consider the effect of these two parameters. Fig. 3 plots the outage probability for different ε and a fixed $\zeta = 0.01$. For small ε, the behaviour of the algorithm approaches the DCPC algorithm and as ε increases, $A_i^2(n)p_i(n-1)$ term is weighted more. For ε too small ($\varepsilon \simeq 0$) and ε too large ($\varepsilon \simeq 1$), the algorithm results in a large outage probability and a reasonable choice is $\varepsilon \in [0.4, 0.6]$.

Fig. 4 plots the outage probability for a fixed $\varepsilon = 0.5$ and different ζ (in logarithm scale). Algorithm (16) switches to DCPC when $\eta_i(n) \le \zeta$. As ζ is large, the iteration switches to DCPC early and the performance of the algorithm is similar to that of DCPC. When $\log(\zeta)$ decreases to -1.3 ($\zeta = 5 \times 10^{-2}$ in linear scale), the outage probability improves. When $\log(\zeta) = -2$, the outage probability of the 10th iteration is better than other choice of $\log(\zeta)$. As $\log(\zeta)$

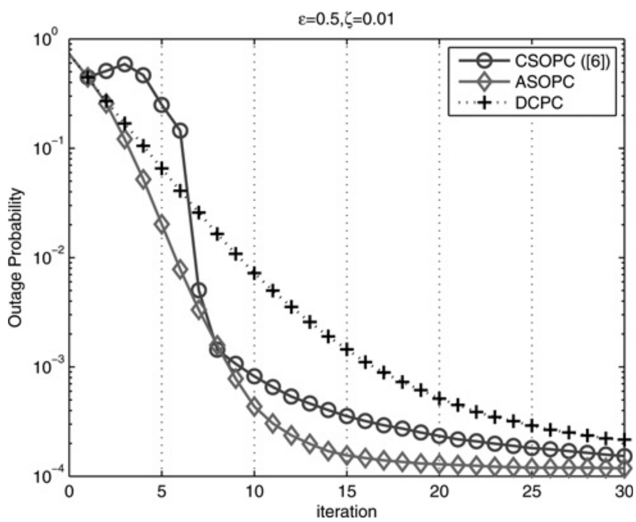

Fig. 2 *Outage probability of DCPC, CSOPC with a = 1.5, and ASOPC with ε = 0.5 and ζ = 0.01*

doi: 10.1049/iet-com.2010.0616

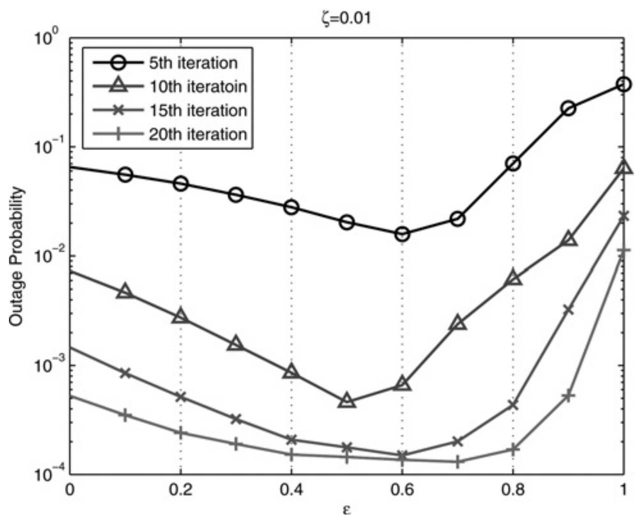

Fig. 3 *Outage probability for different ε and ζ = 0.01*

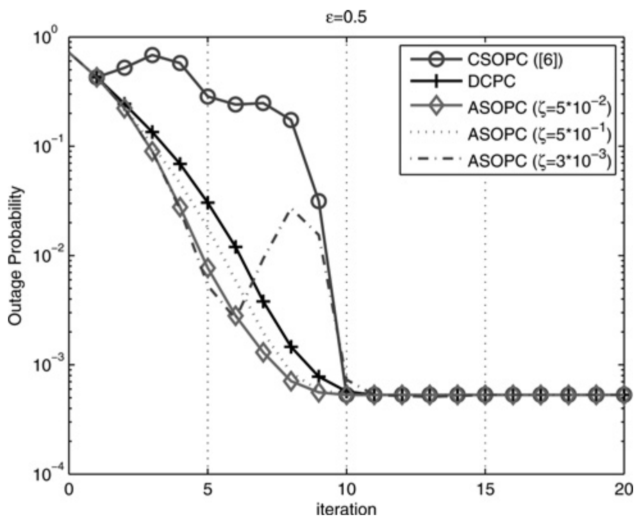

Fig. 5 *Outage probability of 10 users' case*

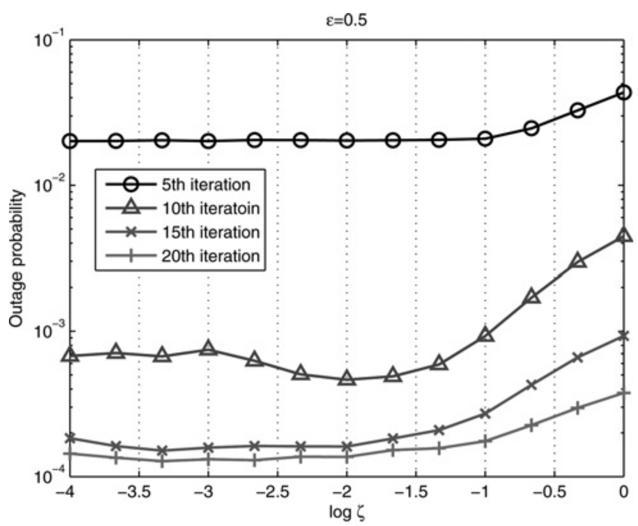

Fig. 4 *Outage probability for different ζ and ε = 0.5*

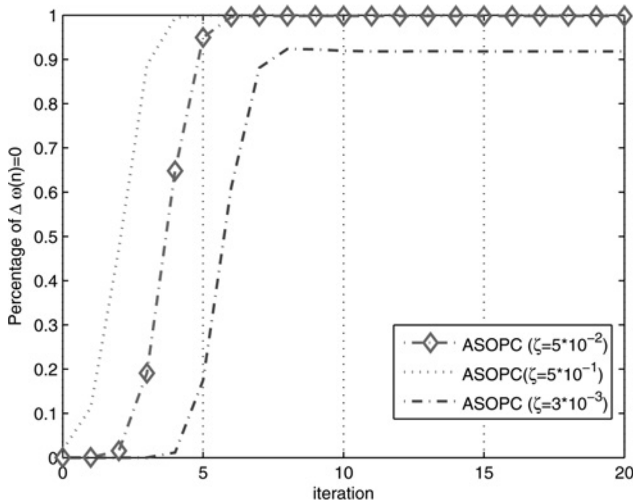

Fig. 6 *Percentage of users switching to DCPC*

decreases further, the outage probability dose not change significantly.

4.2 Example 2

We use another example to see the effect of the parameter ζ. Consider a system in which 10 mobiles are uniformly distributed in a cell. The cell size, processing gain and link gain are set the same as in Example 1. Also, the target SIR is 8 dB, the required SIR is 7 dB, $p_{\max} = 1$ W and $p_{\min} = 4 \times 10^{-6}$ W. The initial power for each mobile is randomly chosen from the interval $[p_{\min}, 1]$. Fig. 5 shows the outage probability by taking 10 000 independent instances. It is clear that when $\zeta = 5 \times 10^{-2}$, the outage probability of ASOPC is better on average than that of CSOPC and DCPC. For $\zeta = 5 \times 10^{-1}$, the outage probability of ASOPC approaches to that of CSOPC owing to early switching. If $\zeta = 3 \times 10^{-3}$, we can see that the outage probability of ASOPC has a slight increase from the seventh iteration to ninth iteration. This is because some users have reached $p_{\min}$ before the switch occurs. In this figure, we see again that with proper choice of power adjustment step between DCPC and CSOPC, the ASOPC performs better than the other two algorithms. Fig. 6 shows the percentage of users switching to DCPC. It is obvious that when $\zeta = 5 \times 10^{-2}$, all users use DCPC to update power after the fifth iteration; when $\zeta = 5 \times 10^{-1}$, all users switch to DCPC update after the fourth iteration; while $\zeta = 10^{-4}$, only about 92% of users switches to DCPC after the eighth iteration.

5 Conclusion

We consider uplink power control in a wireless communication network. The control objective is to achieve a preset target SIR for each user in the network. We modify the SOPC algorithm proposed in [12] to include an adaptive adjustment of the relaxation factor in each iteration. The choice of the relaxation factor is based on the ratio between the DPC update and the power level of the previous step. The adaptive algorithm makes sure that the power updates in the first few steps do not become unnecessarily low, which then results in better performance in terms of outage probability. Simulation results show performance improvement over DCPC and CSOPC.

6 Acknowledgments

We thank the reviewers for their helpful suggestions that improve the paper. This research was sponsored by National Science Council under grant NSC 97-2221-E009-046-MY3.

7 References

1 Li, C., Wang, X.: 'Adaptive opportunistic fair scheduling in power-controlled code division multiple access systems', *IET Commun.*, 2008, **2**, (4), pp. 573–586

2 Han, S.-W., Kim, H., Han, Y.: 'Distributed utility-maximization using a resource pricing power control in uplink DS-CDMA', *IEEE Commun. Lett.*, 2008, **12**, (4), pp. 286–288

3 Ahmed, M.H., Yanikomeroglu, H.: 'Throughput fairness and efficiency of link adaptation techniques in wireless networks', *IET Commun.*, 2009, **3**, (7), pp. 1227–1238

4 Tsiropoulou, E.E., Katsinis, G.K., Papavassiliou, S.: 'Utility-based power control via convex pricing for the uplink in CDMA wireless networks'. Conf. European Wireless, Tuscany, Italy, April 2010, pp. 200–206

5 Fischione, C., Butussi, M., Johansson, K.H., D'Angelo, M.: 'Power and rate control with outage constraints in CDMA wireless networks', *IEEE Commun.*, 2009, **57**, (8), pp. 2225–2229

6 Uykan, Z., Koivo, H.N.: 'Proportional power control algorithm for time varying link gains in cellular radio systems', *IEEE Trans. Veh. Technol.*, 2006, **55**, (1), pp. 341–349

7 Zhang, H., Chen, C.S., Wong, W.S.: 'Distributed power control for time varying systems: performance and convergence analysis', *IEEE Trans. Veh. Technol.*, 2005, **54**, (5), pp. 1896–1904

8 Foschini, G., Miljanic, Z.: 'A simple distributed autonomous power control algorithm and its convergence', *IEEE Trans. Veh. Technol.*, 1993, **42**, (4), pp. 641–646

9 Uykan, Z., Koivo, H.N.: 'Sigmoid-basis nonlinear power-control algorithm for mobile radio systems', *IEEE Trans. Veh. Technol.*, 2004, **53**, (1), pp. 265–270

10 Elmusrati, M., Jäntti, R., Koivo, H.N.: 'Multiobjective distributed power control algorithm for CDMA wireless communication systems', *IEEE Trans. Veh. Technol.*, 2007, **56**, (3), pp. 779–788

11 Gross, T.J., Abrão, T., Jeszensky, P.J.E.: 'Distributed power control algorithm for multiple access systems based on verhulst model', *AEU-Int. J. Electron. Commun.*, available at http://dx.doi.org/10.1016/j.aeue.2010.04.010

12 Jäntti, R., Kim, S.-L.: 'Second-order power control with asymptotically fast convergence', *IEEE J. Sel. Areas Commun.*, 2000, **18**, (3), pp. 447–457

13 Young, D.M.: 'Iterative solution of large linear systems' (Academic, New York, 1971)

14 Golub, G.H., Van Loan, C.F.: 'Matrix computations' (John Hopkins Unviversity Press, 1983)

doi: 10.1049/iet-com.2010.0616