Cross-layer Packet-dependant OFDM Scheduling Based on Proportional Fairness

Hua Hou[i *] and Gen-xuan Li[ii]

Received 08 Jun 2011; received in revised 3 March 2012; accepted 13 May 2012

ABSTRACT

This paper assumes each user has more than one queue, derives a new packet-dependant proportional fairness power allocation pattern based on the sum of weight capacity and the packet's priority in users' queues, and proposes 4 new cross-layer packet-dependant OFDM scheduling schemes based on proportional fairness for heterogeneous classes of traffic. Scenario 1, scenario 2 and scenario 3 lead respectively artificial fish swarm algorithm, self-adaptive particle swarm optimization algorithm and cloud adaptive particle swarm optimization algorithm into sub-carrier allocation in packet-dependant proportional fairness scheduling, and use respectively new power allocation pattern, self-adaptive particle swarm optimization algorithm and population migration algorithm to allocate power. Scenario 4 uses greedy algorithm concerning fairness to allocate sub-carriers, and uses new power allocation pattern to allocate power. Simulation indicates scenario 1,scenario 2 and scenario 3 raise the system's total rate on the basis of undertaking the fairness among users' rates and average packet delay; scenario 4 not only meets users' rates and average packet delay demands, but also improve the fairness among users' rates.

KEYWORDS

Multi-user OFDM, Scheduling; Proportional fairness, Swarm Intelligence Algorithm, Cross-layer, Resource allocation, Particle swarm algorithm, Population migration algorithm, Artificial fish swarm algorithm, Packet-dependant

1. INTRODUCTION

OFDM is one of the main techniques in the future telecommunication system. Scheduling is a technique which supports multi-user data transferring, which provides a mechanism among users competing resource, make users visit fairly the shared resource, and improve the total rate of system according to the requirement of users' QoS in base station.

Hopfield neural network algorithm is used to solve the problem of minimizing the transmitted power in [1]. Game theory is used to solve the problem of maximizing the total system's rate in [2]. Hopfield neural network algorithm is used to solve the problem of maximizing the

i *Corresponding Author Hua Hou is with school of Information Science and Electrical Engineering, Hebei University of Engineering, Handan , P.R.China (Email: hh110040@gmail.com).

ii Gen-xuan Li is with School of Information Science and Electrical Engineering, Hebei University of Engineering, Handan , P.R.China (Email: hh110040@gmail.com).

total system's rate in [3]. A packet-dependant scheduling scenario which is for multi-queues and multi-users under heterogeneous traffic in [4]; it arranges weight for each packet in users' queues, and maximizing the total rate of system. However, the above document does not take the fairness among users' rates into consideration. The problem of maximizing the total rate of system which premises keeping the fairness among users' rates is solved in [5].

Intelligence algorithm is a new evolutionary computation technique which includes Hopfield neural network algorithm, game theory, particle swarm algorithm, artificial fish swarm algorithm and population migration algorithm etc. The resource scheduling of OFDM is the problem of multi-knapsack. Artificial fish swarm algorithm is used to solve the problem of multi-knapsack in [6]. Artificial fish swarm algorithm is used to solve the problem of adaptive resource allocation in multi-user OFDM system in [7]. The particle swarm algorithm is introduced to the sub-carrier allocation in OFDMA system in [8], which minimizes the total transmitted power. The particle swarm algorithm is used to optimize the power allocation in OFDMA system in [9]. Self-adaptive particle swarm algorithm in [10], cloud particle swarm algorithm in [11] and population migration algorithm in [12] are not yet used to solve the problem of resource scheduling in OFDM system.

We suppose that each user has multiple queues in the paper; learn from the sum of weight capacity, the packets weight in users' queues and the proportional fairness among users' rates; infer a new packet-dependant proportional power allocation pattern; and propose 4 new cross-layer packet-dependant OFDM scheduling schemes based on proportional fairness for heterogeneous classes of traffic. These schemes provide packets of different type in users' queues different weight; they maximize the sum of weight capacity on the condition that keeping the proportional fairness among users' rates. In packet-dependant OFDM scheduling based on proportional fairness, scenario 1, scenario 2 and scenario 3 lead, respectively, artificial fish swarm algorithm, self-adaptive particle swarm optimization algorithm and cloud adaptive particle swarm optimization algorithm into sub-carrier allocation in packet-dependant proportional fairness scheduling, and use respectively new power allocation pattern, self-adaptive particle swarm optimization algorithm and population migration algorithm to allocate power; scenario 4 uses greedy algorithm concerning fairness to allocate sub-carriers, and uses new power allocation pattern to allocate power. Simulation indicates scenario 1,scenario 2 and scenario 3 raise the system's total rate on the basis of undertaking the fairness among users' rates and average packet delay; scenario 4 not only meets users' rates and average packet delay demands, but also improve the fairness among users' rates.

2. SYSTEM BLOCK DIAGRAM

Multi-user OFDM system block diagram of downlink scheduling is shown in Figure 1. Each user has three queues, each queue holds separately up voice traffic, video traffic, data traffic. At the beginning of each scheduling, the scheduler first collects the requirements of users' QoS. At the interval of each scheduling, the scheduler in base station collects accurately channel state information (CSI) by means of uplink feedback channel and acquires queue state information (QSI) through observing backlogged packets in users' queues. Then, the scheduler in base station makes up corresponding sub-carrier and power allocation strategy according to the information, and passes the strategies to mobile stations. All users' state information's updating and the scheduling decision are carried out once in each time slot. We assume that the system passed accurately the sub-carrier and power allocation strategy to each mobile station. System distinguishes each user by Frequency Division Multiple Access (FDMA).

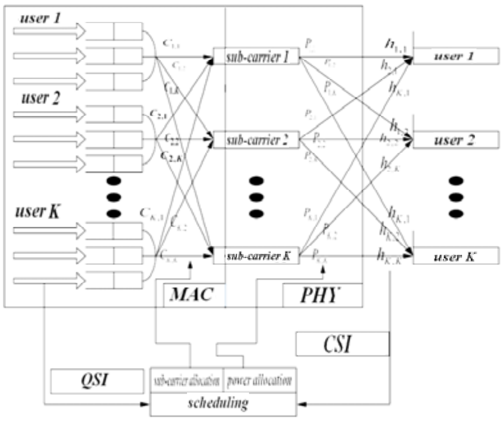

Fig.1 the block diagram of multi-user OFDM system.

3. PACKET-DEPENDANT SCHEDULING BASED ON PROPOR -TIONAL FAIRNESS SYSTEM MODEL

We assume that the total bandwidth of system is *B*, which is divided into N sub-carriers and is shared by *K* users, the total transmitted power is P_T .The channel gain of system is quasi-static in each time slot and the time slot is of length T_{slot} .Let $C_{k,n}$ denote the allocation indicator of user *k* on sub-carrier *n*. $C_{k,n}=0$ indicates that sub-carrier *n* is not allocated to user *k*. $C_{k,n}=1$ indicates that sub-carrier *n* is allocated to user *k* and $C_{k,n}=0$ indicates that sub-carrier *n* is not allocated to user *k*. Define Ω_k as the index set of sub-carriers allocated to user *k*. Let $P_{k,n}$ denote the power allocated user *k* on sub-carrier *n* ($n\in\Omega_k$), $h_{k,n}$ the corresponding channel gain, and N_0 the single-sided power spectral density of additive white Gaussian noise (AWGN). Assuming perfect channel estimation, the achievable instantaneous data rate of user *k* on sub-carrier *n* is expressed as $R_{k,n}=B/N*\log_2(1+P_{k,n}H_{k,n}/\Gamma)$, where $H_{k,n}=h_{k,n}^2/(N_0*B/N)$ is the channel-to-noise power ratio for user *k* on sub-carrier *n*, and Γ is channel-to-noise gap which is expressed as $-\ln(5*p_e)/1.5$,where p_e is *BER*. The total achievable instantaneous data rate of user *k* is given by $R_k=\sum_{n\in\Omega_k}R_{k,n}$.

Define $W_{k,i}'$ as the *i*th queue's weight of user *k* which has something to do with the QoS priority of packets in the *i*th queue of user *k*, the length of packets and the time when packets stay in the queue. We assume that $U_{k,i,f}$ is the maximum time delay of packet *f* in the *i*th queue of user *k*. $\beta_{k,i,f}$ is the QoS priority of packet *f* in *i*th queue of user *k*, which is of length $D_{k,i,f}$. The packet *f* arrives at time t_f and the current time is t_c . So, the time interval when the packet *f* stays is $S_{k,f}=t_c-t_f$ .A guard interval $G_{k,i}$ in *i*th queue of user *k* is introduced to reduce the packet drop rate. The relationship of $U_{k,i,f}$, $S_{k,i,f}$ and G_k is shown in Fig.2.

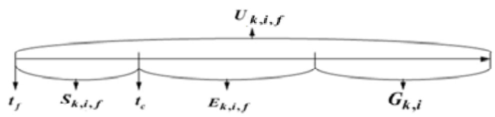

Fig.2. The relationship of various time delays in the system.

In Fig.2, the *fth* packet's $E_{k,i,f}$ in the *fth* queue of user *k* is $U_{k,i,f}-S_{k,i,f}-G_{k,i}$ .when $E_{k,i,f}$ is less than 0, the *fth* packet in the *ith* queue of user *k* is in the state of emergency, whose weight is $W_{k,i,f}'=\beta_{k,i,f}D_{k,i,f}$; unless the *fth* packet in the *ith* queue of user *k* is in the state of non-emergency, whose weight is $W_{k,i,f}'=\beta_{k,i,f}D_{k,i,f}/(E_{k,i,f}+1)$ .In short, we can acquire the following equation:

$$W_{k,i,f}'=\begin{cases}\beta_{k,i,f}D_{k,i,f} & (E_{k,i,f}<0)\\ \beta_{k,i,f}D_{k,i,f}/(E_{k,i,f}+1) & (E_{k,i,f}\geq 0)\end{cases} \tag{1}$$

Through the above analysis, the packets in the *ith* queue of user *k* can be divided into two parts in the current time slot. One part is in the state of emergency, which can be defined as L_k^U ; the other part is in the state of non- emergency, which can be defined as $\overline{L_k^U}$. Therefore the weight of the *ith* queue of user *k*:

$$W_{k,i}'=\sum_{f\in L_k^U}\beta_{k,i,f}*D_{k,i,f}+\sum_{f\in \overline{L_k^U}}\beta_{k,i,f}*D_{k,i,f}/(E_{k,i,f}+1) \tag{2}$$

The weight of user:

$$W_k'=\sum_{i=1}^{3}W_{k,i}' \tag{3}$$

Define Q_k as user *k*'s actual queue length. The mathematical model of packet-dependant scheduling based on proportional fairness can be indicated as:

$$\max J \tag{4}$$
$$= \max \sum_{k=1}^{K} W_k' R_k$$
$$= \max \sum_{k=1}^{K} W_k' \sum_{n=1}^{N} (\frac{B}{N} \log_2 (1 + P_{k,n} \frac{H_{k,n}}{\Gamma}))$$

s.t.

(C1): $P_{k,n} \geq 0 \quad \forall k, n$

(C2): $C_{k,n} = \{0,1\} \quad \forall k, n$

(C3): $\sum_{k=1}^{K} C_{k,n} = 1 \quad \forall k, n$

(C4): $\sum_{k=1}^{K} \sum_{n=1}^{N} (C_{k,n} * P_{k,n}) \leq P_T \quad \forall k, n$

(C5): $R_1 : R_2 : \cdots : R_K = \theta_1 : \theta_2 : \cdots : \theta_K$

(C6): $R_k * T_{slot} \leq Q_k \quad \forall k$

(C4) ensures the total transmitted power constraint of the system. (C5) ensures various users' rates' proportional fairness. (C6) ensures the queue length constraint of user *k*.

The allocation indicator $C_{k,n}$ of user *k* on sub-carrier *n* and the power $P_{k,n}$ allocated user *k* on sub-carrier *n* ($n \in \Omega_k$) in equation (4) are unknown variables. Solving them at the same time is NP-hard. So, sub-optimal solution is used to solve them. Generally, sub-optimal solution is first solving the allocation indicator $C_{k,n}$ of user *k* on sub-carrier *n* (namely sub-carrier allocation), and then solving the power $P_{k,n}$ allocated user *k* on sub-carrier *n* ($n \in \Omega_k$) (namely power allocation). Equation (C6) is used as a judgmental condition in sub-carriers allocation algorithm, which isn't listed in the following derivation.

4. CROSS-LAYER SCHEDULING SCHEME

Scenario 1

First part: sub-carrier allocation

The artificial fish swarm algorithm in the sub-carrier allocation of the scenario simulating the behavior of fish swarm search optimal solution in the solution space. In the algorithm, each artificial fish selects one of the behaviors which are foraging, clustering and so on to execute according to the changing situation of its target function. In the end, artificial fish will gather around in a few local extremes; and then the algorithm selects the global optimal solution from these local extremes.

1) Initialize the iterative times gen, artificial fish swarm size M, the size of each artificial fish vector is the number of sub-carrier *N*, the perception distance of artificial fish *visual*, retry times *try_number*, the crowding factor δ.

2) Coding: there are *K* users in the system, and generate randomly $M-1$ artificial fish vectors $F_i^1 (i = 1, \cdots, M-1)$ which is N dimensions, where the element in the vector is between 1 and *K*. The sub-carrier allocation result in packet-dependant scheduling [4] is coded as the *M*th artificial fish vector which is *N* dimensions. So, if the *j*th element of the vector is a, the system allocates the *j*th sub-carrier to user a. When the scenario computes the target function, it translates the vector into the corresponding sub-carrier allocation matrix. Assume the situation of artificial fish vector ***i*,*j*** is separately F_i^t, F_j^t; define the distance of artificial fish vector ***i*,*j*** $d_{i,j} = \left| F_i^t - F_j^t \right|$.

3) The computation of food concentration: $\{\theta_1, \theta_2, \cdots, \theta_K\}$ is a group of predefined numbers. $f = \sum_{k=1}^{K} (R_k - \theta_k (\sum_{k=1}^{K} R_k / \sum_{k=1}^{K} \theta_k))^2$ indicates the fairness degree among users. The greater f is, the poorer the fairness degree among users is; the lower f is, the better the fairness degree among users is. The sub-carrier allocation in the scenario define $J2 = \sum_{k=1}^{K} W_k' R_k - \zeta 2 \sum_{k=1}^{K} (R_k - \theta_k (\sum_{k=1}^{K} R_k / \sum_{k=1}^{K} \theta_k))^2$ as food concentration, where weighting coefficient $\zeta 2 \geq 0$

4) Compute the food concentration of each artificial fish, record the artificial fish vector *F_best* which has the global greatest food concentration.

5)

5.1) Evaluate each artificial fish; select its behavior to execute, which includes foraging, clustering and chasing.

5.2) The definition of single artificial fish's behavior

5.2.1) Foraging: assume that the current situation of artificial fish *i* is F_i^t, select *try_number* times $F_m^{t+1} = ceil(F_i^t + visual * rand())$ within the scope of its perception ($d_{i,j}$ <*visual*), where *rand()* generates the number which is between 0 and 1. If $J2_m > J2_i$,

$$F_i^{t+1} = ceil(F_i^t + \left[\frac{(F\_best - F_i^t) + (F_m^t - F_i^t)}{\left\| (F\_best - F_i^t) + (F_m^t - F_i^t) \right\|} \right] * rand());$$

if $J2_i \ge J2_m$ after trying *try_number* times, $F_i^{t+1} = ceil(F_i^t + visual * rand())$; and F_i^{t+1} must make $R_k * T_{slot} \le Q_k \quad \forall k$.

5.2.2) Clustering: assume that the current situation of artificial fish i is F_i^t , search other artificial fish F_m^t within the scope of its perception ($d_{i,j}$ <*visual*), define the number of artificial fish within the scope of its perception ($d_{i,j}$ <*visual*) as the number of friends nf . If nf isn't equal to 0, search the center of the artificial fish F_{center} , and compute the food concentration of the center J_{center} .

If $J_{center} / (\delta * nf) > J_i$,

$$F_i^{t+1} = ceil(F_i^t + \left[\frac{(F\_best - F_i^t) + (F_{center} - F_i^t)}{\left\| (F\_best - F_i^t) + (F_{center} - F_i^t) \right\|} \right] * rand());$$

or else execute foraging behavior, and F_i^{t+1} must make $R_k * T_{slot} \le Q_k \quad \forall k$.

5.2.3) Chasing: assume that the current situation of artificial fish i is F_i^t , search other artificial fish F_m^t within the scope of its perception ($d_{i,j}$ <*visual*), define the number of artificial fish within the scope of its perception ($d_{i,j}$ <*visual*) as the number of friends nf . If nf is not equal to 0, search the artificial fish $F_{\max}$ which has maximum food concentration, and compute the food concentration of the center $J_{\max}$.

If $J_{\max} / (\delta * nf) > J_i$, $F_i^{t+1} = ceil(F_i^t + [(F\_best$

$$-F_i^t) + (F_{\max} - F_i^t)] \,/ \left\| (F\_best - F_i^t) + (F_{\max} - F_i^t) \right\| * rand());$$

or else execute foraging behavior, and F_i^{t+1} must make $R_k * T_{slot} \le Q_k \quad \forall k$.

6) Execute the behavior which it selects, update the global optimal the artificial fish *F_best*. Translate the vector into corresponding sub-carrier allocation matrix *suballo*, and according to *suballo* update $C_{k,n} (k = 1, \cdots, K; n \in \Omega_k)$.

7) If the iterative times are enough, translate the vector into corresponding sub-carrier allocation matrix *suballo* , and according to *suballo* update $C_{k,n} (k = 1, \cdots, K; n \in \Omega_k)$. Or else return 4).

Second part: power allocation

While sub-carrier allocation is finished, the allocation index $C_{k,n}$ of user k on sub-carrier n is solved. Equation (4) translates into:

$$\max J \tag{5}$$
$$= \max \sum_{k=1}^{K} W_k' R_k$$
$$= \max \sum_{k=1}^{K} W_k' \sum_{n \in \Omega_k} (\frac{B}{N} \log_2 (1 + P_{k,n} \frac{H_{k,n}}{\Gamma}))$$

s.t.

(C1): $P_{k,n} \ge 0 \quad \forall k, n$

(C4): $\sum_{k=1}^{K} \sum_{n \in \Omega_k} P_{k,n} \le P_T \quad \forall k, n$

(C5): $R_1 : R_2 : \cdots : R_K = \theta_1 : \theta_2 : \cdots : \theta_K$

According to Karush-Kuhn-Tucker(KKT)[6] condition, equation (5) translates into:

$$\max J1 \tag{6}$$
$$= \max \sum_{k=1}^{K} W_k' R_k$$
$$= \max \sum_{k=1}^{K} W_k' \sum_{n \in \Omega_k} (\frac{B}{N} \log_2 (1 + P_{k,n} \frac{H_{k,n}}{\Gamma}))$$

s.t.

(C7): $\partial J1 / \partial P_{j,n} = 0 \; (j = 1, \cdots, K; n \in \Omega_j)$

(C8): $\zeta \ge 0$

(C9): $\zeta * (\sum_{k=1}^{K} \sum_{n \in \Omega_k} P_{k,n} - P_T) = 0$

(C10): $R_j : R_k = \theta_j : \theta_k (j = 1, \cdots, K; k = 1, \cdots, K)$

where ζ is lagrange factor.

Substitute (C10) into (6):

$$\max J2 \tag{7}$$
$$= \max [\sum_{k=1}^{K} W_k' \theta_k / \theta_j * R_j - \zeta (\sum_{k=1}^{K} \sum_{n \in \Omega_k} P_{k,n} - P_T)] (j = 1, \cdots, K)$$

s.t.

(C7): $\partial J1 / \partial P_{j,n} = 0 \; (j = 1, \cdots, K; n \in \Omega_j)$

(C8): $\zeta \ge 0$

(C9): $\zeta * (\sum_{k=1}^{K} \sum_{n \in \Omega_k} P_{k,n} - P_T) = 0$

Differentiate $J2$ to $P_{j,n}$ $(j = 1, \cdots, K; n \in \Omega_j)$:

$$\partial J2/\partial P_{j,n}=B/(N*\ln 2)*(\frac{H_{j,n}}{\Gamma}/(1+\frac{H_{j,n}}{\Gamma}*P_{j,n}))*\sum_{k=1}^{K}W_k\theta_k/\theta_j-\zeta=0 \quad (8)$$

$$(j=1,\cdots,K;n\in\Omega_j)$$

s.t.

(C8): $\zeta \geq 0$

(C9): $\zeta *(\sum_{k=1}^{K}\sum_{n\in\Omega_k}P_{k,n}-P_T)=0$

According to (8) and (C4) infer:

The power of user k on sub-carrier n :

$$P_{j,n}=\{\frac{[P_T+\sum_{k=1}^{K}\sum_{n\in\Omega_k}\Gamma/H_{k,n}]}{\sum_{j=1}^{K}|\Omega_j|/\theta_j^2}-\Gamma/H_{j,n}\}^+ \ (j=1,\cdots,K;n\in\Omega_j) \quad (9)$$

According to the sub-carrier allocation result *suballo* in first part add up the number of sub-carriers each user obtaining $|\Omega_j|(j=1,\cdots,K)$, and then use equation (9) to compute $P_{j,n}(j=1,\cdots,K;n\in\Omega_j)$; if $P_{j,n}<0$, set $P_{j,n}=0$。

Scenario 2

First part: sub-carrier allocation

Particle swarm algorithm simulates the law of birds feeding. In solution space, the individual in each generation flies towards the area which has high fitness on the basis of the optimal area which it and its yellows pass; thus the next generation has higher fitness than the current generation. The particle of adaptive particle swarm algorithm in the scenario has individual inertia weight and the weight can dynamically adjust.

1)Initialization: the particle swarm size is M , the iterative times are $MaxDT$, the dynamic adjustment part of inertia weight is ω_1 , fixed part is ω_2 , cognitive factor is c_1 ,social factor is c_2 , current generation $t=1$.

2) Coding: generate randomly $M-1$ particle vectors $F_i^1(i=1,\cdots,M-1)$ which is N dimensions, and again generate randomly M velocity vectors $V_i^1(i=1,\cdots,M)$ which is N dimensions, where the element in vectors is between 1 and K ; the sub-carrier allocation results of packet-dependant scheduling[4] are coded as the Mth particle vector which is N dimensions.

3) Fitness computation:

the sub-carrier allocation in the scheme define $J2=\sum_{k=1}^{K}W_k'R_k-\zeta 2\sum_{k=1}^{K}(R_k-\theta_k(\sum_{k=1}^{K}R_k/\sum_{k=1}^{K}\theta_k))^2$ as fitness, where weighting coefficient $\zeta 2\geq 0$。

4) Initialize the individual extreme vector P_i^1 of particle $i(i=1,\cdots,M)$ in particle swarm, the particle vector pg^1 which has maximum fitness is global extreme vector.

5) Compute the inertia weight $\omega=(\omega_1-t)*(\omega_1-\omega_2)/MaxDT$ of particle in the tth generation, and again compute the velocity vector $V_i^t=\omega(i)*V_i^{t-1}+c_1*rand()*(P^{t-1}-F_i^{t-1})+c_2*rand()*(pg^{t-1}-F_i^{t-1})$ of particle in particle swarm in tth generation. Finally, compute the new location $F_i^t=ceil(F_i^{t-1}+V_i^t)$ of particle $i(i=1,\cdots,M)$ where R_k*T_{slot} of F_i^t mustn't be more than Q_k $\forall k$.

6) Compute the fitness l of the tth generation new location F_i^t of particle $i(i=1,\cdots,M)$. If l is higher than the fitness of the (t-1)th generation individual extreme vector P_i^{t-1} , update the tth generation individual extreme vector $P_i^t=F_i^t$; or else P_i^{t-1} keeps unchanged. Update the tth generation global extreme vector pg^t .

7) Judge whether the iterative times are enough. If it is not enough, return 5). Or else finish iteration. Assign result pg^{MaxDT} to *suballo* and according to *suballo* update $C_{k,n}(k=1,\cdots,K;n\in\Omega_k)$.

Second part: power allocation

The scenario use adaptive particle swarm algorithm to allocate power.

1)Initialization: the particle swarm size is M , the iterative times are $MaxDT$, the dynamic adjustment part of inertia weight is ω_1 , fixed part is ω_2 , cognitive factor is c_1 ,social factor is c_2 , current generation $t=1$.

2) Coding: generate randomly $M-1$ particle vectors $F_i^1(i=1,\cdots,M-1)$ which is N dimensions, and again generate randomly M velocity vectors $V_i^1(i=1,\cdots,M)$ which is N dimensions, where the element in vectors is between 1 and 2/N (N is the number of sub-carriers); Power allocation result of new power allocation pattern obtained by means of *suballo* in first part is coded

the Mth population vector.

3) Fitness computation: the sub-carrier allocation in the scheme define $J2=\sum_{k=1}^{K}W_k'R_k-\zeta 2\sum_{k=1}^{K}(R_k-\theta_k(\sum_{k=1}^{K}R_k/\sum_{k=1}^{K}\theta_k))^2$ as fitness, where weighting coefficient $\zeta 2\geq 0$。

4) Initialize the individual extreme vector P_i^1 of particle $i(i=1,\cdots,M)$ in particle swarm, the particle vector pg^1 which has maximum fitness is global extreme vector.

5) Compute the inertia weight $\omega=(\omega_1-t)*(\omega_1-\omega_2)/MaxDT$ of particle in the tth generation, and again compute the velocity vector $V_i^t=\omega(i)*V_i^{t-1}+c_1*rand()*(P^{t-1}-F_i^{t-1})+c_2*rand()*(pg^{t-1}-F_i^{t-1})$ of particle in particle swarm in tth generation. Finally, compute the new location $F_i^t=ceil(F_i^{t-1}+V_i^t)$ of particle $i(i=1,\cdots,M)$ where R_k*T_{slot} of F_i^t mustn't be more than Q_k $\forall k$.

6) Compute the fitness l of the tth generation new location F_i^t of particle $i(i=1,\cdots,M)$. If l is higher than the fitness of the (t-1)th generation individual extreme vector P_i^{t-1}, update the tth generation individual extreme vector $P_i^t=F_i^t$; or else P_i^{t-1} keeps unchanged. Update the tth generation global extreme vector pg^t .

7) Judge whether the iterative times are enough. If it is not enough, return 5). Or else finish iteration. Assign result pg^{MaxDT} to *powerallo* and according to *powerallo* update $P_{k,n}(k=1,\cdots,K;n\in\Omega_k)$.

Scenario 3

First part: sub-carrier allocation

In sub-carrier allocation of the scheme, inertia weight ω of cloud adaptive particle swarm algorithm is different from adaptive particle swarm algorithm.

1)Initialization: the particle swarm size is M , the iterative times are $MaxDT$, the dynamic adjustment part of inertia weight is ω_1 , fixed part is ω_2 , cognitive factor is c_1 ,social factor is c_2 , current generation $t=1$.

2) Coding: generate randomly $M-1$ particle vectors $F_i^1(i=1,\cdots,M-1)$ which is N dimensions, and again generate randomly M velocity vectors $V_i^1(i=1,\cdots,M)$ which is N dimensions, where the element in vectors is between 1 and K ; the sub-carrier allocation results of packet-dependant scheduling[4] are coded as the Mth particle vector which is N dimensions.

3) Fitness computation: the sub-carrier allocation in the scheme define $J2=\sum_{k=1}^{K}W_k'R_k-\zeta 2\sum_{k=1}^{K}(R_k-\theta_k(\sum_{k=1}^{K}R_k/\sum_{k=1}^{K}\theta_k))^2$ as fitness, where weighting coefficient $\zeta 2\geq 0$。

4) Initialize the individual extreme vector P_i^1 of particle $i(i=1,\cdots,M)$ in particle swarm, the particle vector pg^1 which has maximum fitness is global extreme vector.

5) According to the following equation:

$Ex=\overline{f}$ //average fitness of particle in particle swarm

$En=(Ex-f\max)/2.9$

$He=En/10$

$En'=randn(En,He)$ // generate the Gaussian distribution whose average value is En and whose standard deviation is He .

$$p(i)=\begin{cases}\dfrac{-(Ex-f(i))^2}{2(En')^2} & f(i)<Ex\\ 0.9 & f(i)\geq Ex\end{cases}$$

$\omega(i)=0.9*\omega(i)$

, compute the inertia weight $\omega(i)$ of particle i in the tth generation, and again compute the velocity vector $V_i^t=\omega(i)*V_i^{t-1}+c_1*rand()*(P^{t-1}-F_i^{t-1})+c_2*rand()*(pg^{t-1}-F_i^{t-1})$ of particle in particle swarm in tth generation. Finally, compute the new location $F_i^t=ceil(F_i^{t-1}+V_i^t)$ of particle $i(i=1,\cdots,M)$ where R_k*T_{slot} of F_i^t mustn't be more than Q_k $\forall k$.

6) Compute the fitness l of the tth generation new location F_i^t of particle $i(i=1,\cdots,M)$. If l is higher than the fitness of the (t-1)th generation individual extreme vector P_i^{t-1}, update the tth generation individual extreme vector $P_i^t=F_i^t$; or else P_i^{t-1} keeps unchanged. Update the tth generation global extreme vector pg^t .

7) Judge whether the iterative times are enough. If it is not enough, return 5). Or else finish iteration. Assign

result pg^{MaxDT} to *suballo* and according to *suballo* update $C_{k,n}(k=1,\cdots,K;n\in\Omega_k)$.

Second part: power allocation

In the power allocation of the scheme, population migration algorithm simulates the discipline of population movement. It uses objective function to measure the attraction of areas which population moves into, calls the area whose objective function value is high favorable area. It defines four patterns searching the optimal solution in solution space, which are people's flowing in origin, people's migration attracted by favorable areas, people flowing in favorable areas until population pressure reaches a certain limit, and people moving from favorable areas.

1) Initialize the iterative times $\max DT$, $radius$, population pressure index $alpha$, Contraction coefficient $delta$, and population vectors scale M.

2) Coding: generate randomly $M-1$ population vector $F_i^1(i=1,\cdots,M-1)$ which is N dimensions as initial population group, where the element in the vector is between 0 and 2/*N* (*N* is the number of sub-carriers). Power allocation result of new power allocation pattern obtained by means of *suballo* in first part is coded the *M*th population vector.

3) Attraction computation: define $f=\sum_{k=1}^{K}W_k'R_k-\zeta 2\sum_{k=1}^{K}(R_k-\theta_k(\sum_{k=1}^{K}R_k/\sum_{k=1}^{K}\theta_k))^2$ as the attraction, where the weighting coefficient $\zeta 2\geq 0$.

4) Compute the attraction $f(F_i^1)(i=1,\cdots,M)$ of the population vector $i(i=1,\cdots,M)$ residence. Record the population vector $p\max^1$ which has global maximum attraction.

5) Population vector $i(i=1,\cdots,M)$ flows in its area. Generate its *t*th generation vector $F_i^t=2*radius*rand()+(F_i^{t-1}-radius)$, and then compute the attraction of the vector residence $f(F_i^t)(i=1,\cdots,M)$. Record the population vector $p\max^t$ which has global maximum attraction.

6) Population migration: generate M population vector $F_{i\text{new}}^t(i=1,\cdots,M)$ in the areas whose center is $p\max^t$ and whose radius is *radius* to replace the previous vector $F_i^t(i=1,\cdots,M)$.

7) Compute the attraction of the vector $F_{i\text{new}}^t(i=1,\cdots,M)$ residence $f(F_i^t)(i=1,\cdots,M)$. Record the population vector $p\max_{new}^t$ which has global maximum attraction.

8) According to $radius=(1-delta)*radius$ contract the favorable area, define the area whose center is $p\max_{new}^t$ and whose radius is *radius* as the favorable area. In the area generate randomly M population vector $F_{i\text{new}1}^t(i=1,\cdots,M)$ to replace the vector $F_{i\text{new}}^t(i=1,\cdots,M)$. Record the population vector $p\max_{new1}^t$ which has current global maximum attraction.

9) Repeat 8) until $radius\leq alpha$.

10) Population spreading: generate randomly $M-1$ population vector $F_i^{t+1}(1,\cdots,M-1)$ which is N dimensions as initial population group, where the element in the vector is between 0 and 2/*N* (*N* is the number of sub-carriers).Power allocation result of new power allocation pattern obtained by means of *suballo* in first part is coded the *M*th population vector. Compute the attraction $f(F_i^{t+1})(i=1,\cdots,M)$ of the population vector $i(i=1,\cdots,M)$ residence. Record the population vector $p\max^{t+1}$ which has global maximum attraction.

11) The number of current generation $t=t+1$, if t is lower than $\max DT$, return 5).Or else power allocation result $powerallo=p\max_{new1}^t$ and according to *powerallo* update $P_{k,n}(k=1,\cdots,K;n\in\Omega_k)$.

Scenario 4

First part: sub-carrier allocation

The scenario uses the greedy algorithm considering fairness to allocate sub-carrier.

1) Compute each user's weight W_k'; the number of sub-carrier initially allocated to each user $N_k=\left\lfloor\frac{\theta_k}{\sum_{k=1}^{K}\theta_k}*N\right\rfloor$;

while sub-carrier is initially allocated, the number of remaining unallocated sub-carriers is $N^*=N-\sum_{k=1}^{K}N_k$;

User set is $\Phi=\{1,2,\cdots,K\}$, sub-carrier set is $N'=\{1,2,\cdots,N\}$;for

$\forall k \in \Phi, n \in N'$ $R_k = 0, \Omega_k = \phi, C_{k,n} = 0, P_{k,n} = \frac{P_T}{N}$.

2) Allocate the sub-carrier whose $(H_{k,n}/\Gamma)^{W_k'}(n=1,\cdots,N)$ is highest to the user $k(k=1,\cdots,K)$, and update $C_{k,n}$, N', R_k. If $R_k * T_{slot}$ of user k isn't less than its length of queue Q_k, user k finishes allocating, delete user k from user set Φ.

3) Allocate the sub-carrier whose $(H_{k,n}/\Gamma)^{W_k'}(n=1,\cdots,N)$ is highest to the user $k(k=1,\cdots,K)$ whose $R_k/\theta_k(k=1,\cdots,K)$ is least. and update $C_{k,n}$, N', R_k. If $R_k * T_{slot}$ of user k isn't less than its length of queue Q_k, user k finishes allocating, delete user k from user set Φ; the process loops until N' becomes empty set.

4) Allocate the remaining N^* sub-carriers to the user k whose $(H_{k,n}/\Gamma)^{W_k'}$ is highest on the sub-carrier, and assign the final sub-carrier allocation result to $suballo$.

Second part: power allocation

According to the sub-carrier allocation result $suballo$ in first part add up the number of sub-carriers each user obtaining $|\Omega_j|(j=1,\cdots,K)$, and then use equation (9) to compute $P_{j,n}(j=1,\cdots,K; n\in\Omega_j)$; if $P_{j,n}<0$, set $P_{j,n}=0$.

5. SIMULATION ANALYSIS

The system's bandwidth is 1 MHz; it is divided into 128 sub-carriers. Wireless channel is six-path frequency selective fading channel, the envelope of power delay is e^{-g}, where multi-path index is g. The total transmitted power of system is 1W, proportional rate constraint is $\theta_1:\theta_2:\cdots:\theta_K = 1:1:\cdots:1$. In user's queues, voice traffic packet parameters are $U_{k,i,f}=100$ ms, $\beta_{k,i,f}=1024$, $D_{k,i,f}=500$ bits; video traffic packet parameters are $U_{k,i,f}=400$ ms, $\beta_{k,i,f}=512$, $D_{k,i,f}=239$ bits; data traffic packet parameters are $U_{k,i,f}=1000$ ms, $\beta_{k,i,f}=1$, $D_{k,i,f}=64$ bits. The guard interval of user's queues is 10 ms, the weighting coefficient $\zeta 2 = 5$. In scenario 1, the iterative times are $gen=100$, artificial fish population size is $M=31$, the perception distance of artificial fish is $visual=5$, the tried times are $try\_number=5$, the crowding factor is $\delta = 0.2$. In scenario 2, the particle swarm size is $M=31$, the iterative times are $MaxDT=70$, the dynamic adjustment part of inertia weight is $\omega_1 = 1.4$, fixed part is $\omega_2 = 0.4$, cognitive factor is $c_1 = 2$, social factor is $c_2 = 1$. In the sub-carrier allocation of scenario 3, the particle swarm size is $M=31$, the iterative times are $MaxDT=70$, the dynamic adjustment part of inertia weight is $\omega_1 = 1.4$, fixed part is $\omega_2 = 0.4$, cognitive factor is $c_1 = 2$, social factor is $c_2 = 1.5$. In the power allocation of scenario 3, maximum iterative times are $\max DT = 50$, the radius of population migration algorithm is $radius = 0.00078$, the population pressure index is $alpha = 1e-5$, the contraction coefficient is $delta = 0.4$, the size of population vector swarm is $M=31$. In simulation, the algorithm in [4] is called packet-dependant scheme, the algorithm in [5] is called proportional fairness scheme.

The impact of number of users and *SNR* on the system's total rate is separately expressed in Fig.3 and Fig.4. As number of users and *SNR* increase, the function of multi-user density enhances, the system's total rate of all algorithms shows increasing trend. Packet-dependant scheme does not take proportional fairness among users' rates into consideration, the user whose rate and weight is high may obtain the sub-carrier whose *SNR* is high, and its system's total rate is high. Proportional fairness scheme, while allocating resource, does not take the fairness among users into consideration; the user whose rate is low may obtain the sub-carrier whose *SNR* is high; its system's total rate is low. Scenario 1, scenario 2 and scenario 3, while allocating resource, don't take users' weight and the fairness degree $f = \sum_{k=1}^{K}(R_k - \theta_k(\sum_{k=1}^{K} R_k / \sum_{k=1}^{K}\theta_k))^2$ into consideration; in the solution selected by swarm intelligence algorithm, the user whose proportional rate and whose weight and whose rate is high may obtain the sub-carrier whose *SNR* is high, their system's total rate is higher than packet-dependant scheme. Scenario 4 also takes users' weight and the fairness among users into consideration, its system's total rate is between packet-dependant scheme and proportional fairness scheme.

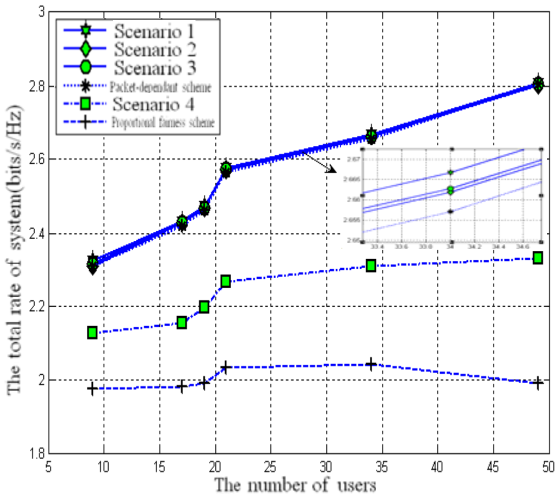

Fig.3 The total rate of system versus the number of users.

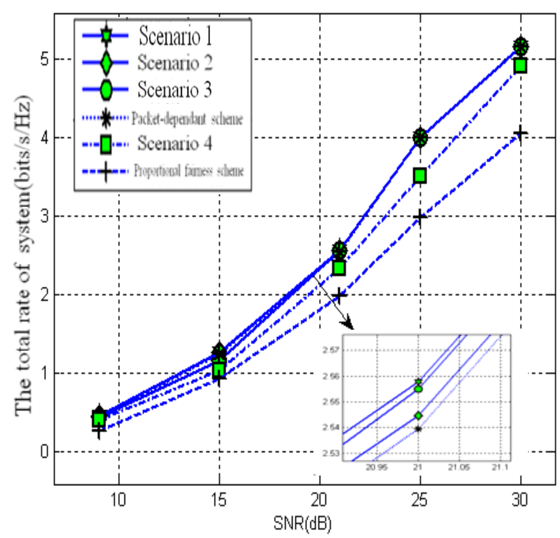

Fig.4 The total rate of system versus *SNR*.

The impact of number of users on average packet delay is expressed in Fig.5. The system's total rate of packet-dependant scheme is high, its average packet delay is low; the system's total rate of proportional fairness scheme is low, its average packet delay is high; the system's total rate in scenario 1, scenario 2 and scenario 3 is higher than packet-dependant scheme; their average packet delay is lower than packet-dependant scheme; the system's total rate in scenario 4 is between packet-dependant scheme and proportional fairness scheme, its average packet delay is between them.

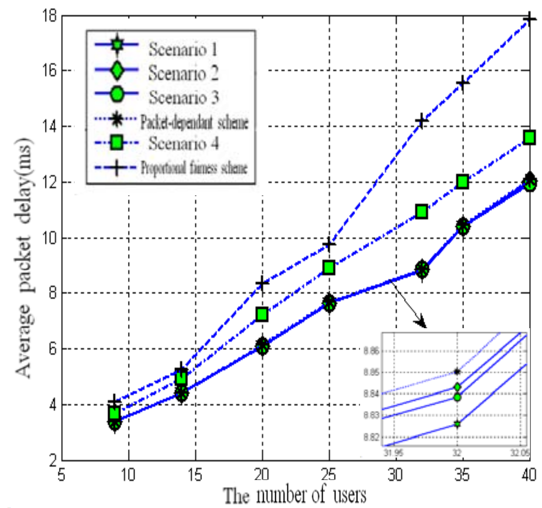

Fig.5 Average packet delay versus the number of users.

Jain's fairness index [5] is defined as $F = \dfrac{(\sum_{k=1}^{K} R_k)^2}{K\sum_{k=1}^{K} R_k^2}$.

The impact of number of users and *SNR* is separately expressed in Fig.6 and Fig.7. The higher Jain's fairness index is, the better the fairness among users is. Packet-dependant scheme does not consider fairness among users, the user whose rate and whose weight is low may obtain sub-carrier whose *SNR* is low, the system's capacity does not distribute uniformly among users, and its Jain's fairness index is low. In proportional fairness scheme, the user whose *SNR* is low may obtain sub-carrier whose *SNR* is high, the system's capacity distributes uniformly among users, its Jain's fairness index is high. Scenario 2 and scenario 3 take users' weight and the fairness degree among users $f = \sum_{k=1}^{K}(R_k - \theta_k(\sum_{k=1}^{K} R_k / \sum_{k=1}^{K}\theta_k))^2$ into consideration; while swarm intelligence algorithm selects the optimal solution, the solution where the fairness degree among users is low, the system's capacity distribute uniformly among users, the system's total rate is high is firstly considered; its Jain's fairness index is between packet-dependant scheme and proportional fairness scheme. In scenario 4, the user whose rate is low may obtain the sub-carrier whose *SNR* is high; the system's capacity distributes uniformly among users; new power allocation pattern raises the power allocated to each user in comparison with proportional fairness scheme; thus its Jain's fairness index is higher than proportional fairness scheme.

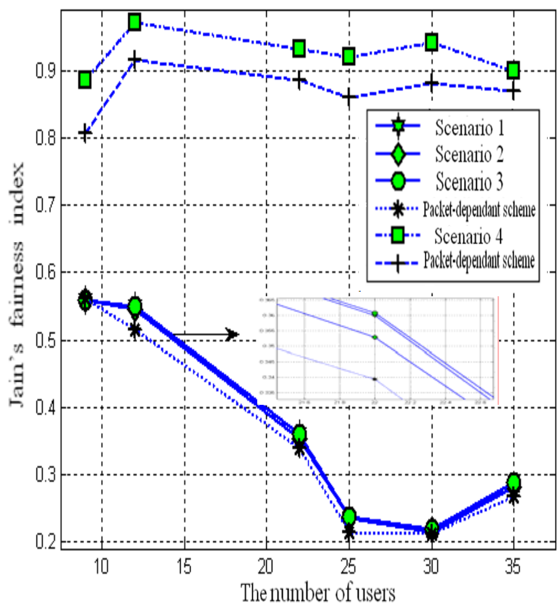

Fig.6 Jain's fairness index versus the number of users.

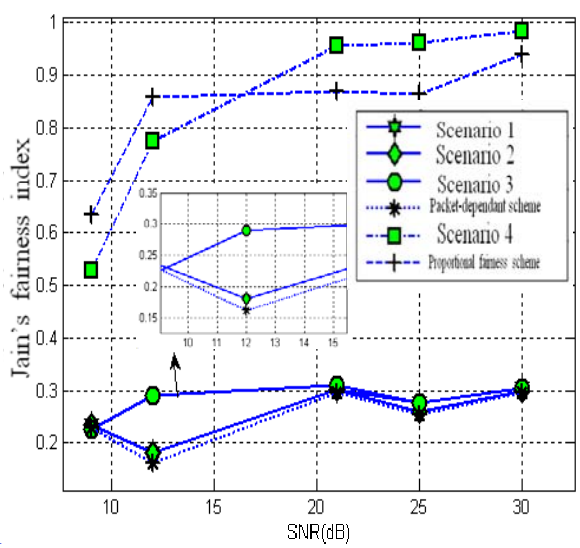

Fig.7 Jain's fairness index versus *SNR*.

Maximum proportional rate deviation is defined as as $dv=\max(R_k/\theta_k)-\min(R_k/\theta_k)(k=1,\cdots,K)$. The impact of number of users and *SNR* on maximum proportional rate deviation is separately expressed in Fig.8 and Fig.9.

The lower maximum proportional rate deviation is, the better the fairness among user is. Jain's fairness index of packet-dependant scheme is low; its maximum proportional rate deviation is high. Jain's fairness index of proportional fairness scheme is high; its maximum proportional rate deviation is low. Jain's fairness index in scenario 1, scenario 2 and scenario 3 is between packet-dependant scheme and proportional fairness scheme, their maximum proportional rate deviation is between them. Jain's fairness index in scenario 4 is higher than proportional fairness scheme; its maximum proportional rate deviation is lower than it.

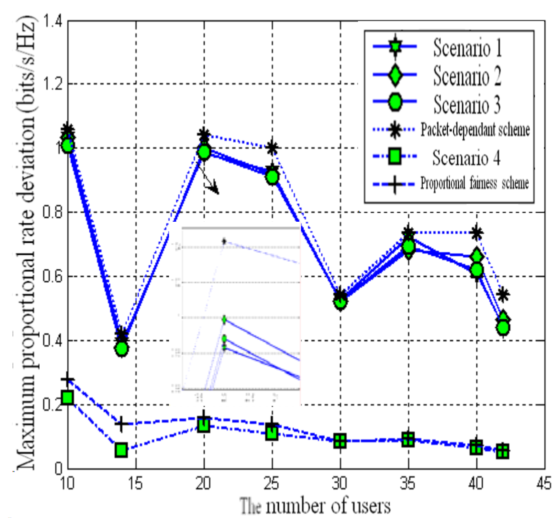

Fig.8 Maximum proportional rate deviation versus the number of users.

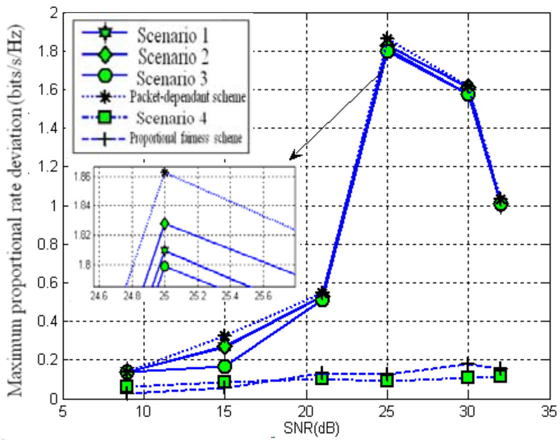

Fig.9 Maximum proportional rate deviation versus *SNR*.

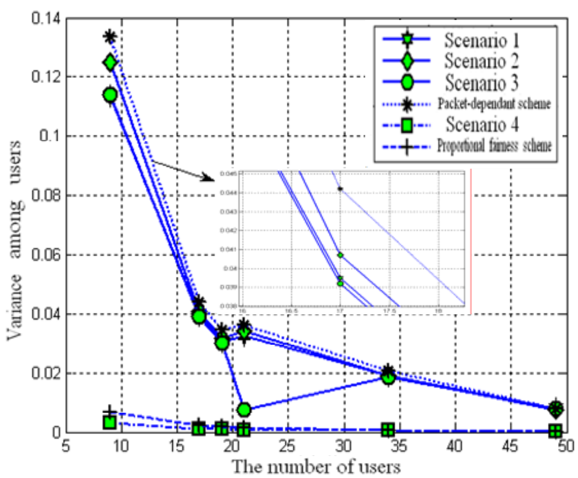

Fig.10 Variance among users versus the number of users.

The impact of number of users on variance among users is expressed in Fig.10. The lower variance among users is, the better the fairness among user is. Maximum proportional fairness rate in scenario 1, scenario 2 and scenario 3 is lower than packet-dependant scheme; variance among users is lower than packet-dependant scheme. Maximum proportional rate deviation of proportional fairness scheme is lower than scenario 1, scenario 2 and scenario 3; variance among users is lower than them. Maximum proportional rate deviation in scenario 4 is lower than proportional fairness scheme; variance among users is lower than proportional fairness scheme.

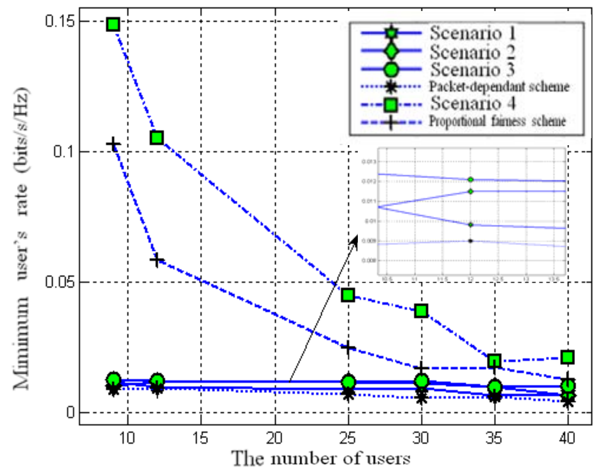

Fig.11 Minimum user's rate versus the number of users.

Minimum user's rate is also used as an indicator of measuring the fairness among users in the paper. The higher minimum user's rate is, the better the fairness among user is. As the number of users increases, the probability of that a certain sub-channel is deep fading for all users becomes lower [5], and minimum user's rate decreases in the figure. In the packet-dependant scheme, the user whose rate and whose weight is low may not obtain sub-carrier, so minimum user's rate is low. The system capacity of proportional fairness scheme scatters among all users, its minimum user's rate is higher than packet-dependant scheme. The total rate of system in scenario 1 to scenario 3 is higher than packet-dependant scheme, the maximum proportional rate deviation in scenario 1 to scenario 3 is between packet-dependant scheme and proportional fairness scheme, and so the minimum user's rate in scenario 1 to scenario 3 is between packet-dependant scheme and proportional fairness scheme. The system's total rate in scenario 4 is higher than proportional fairness scheme, the maximum proportional rate deviation in scenario 4 is lower than it, and minimum user's rate is higher than it.

6. CONCLUSIONS

The paper supposes that each user has multiple queues, derives a new packet-dependant proportional fairness power allocation pattern, and finally proposes 4 new cross-layer packet-dependant proportional fairness scheduling schemes in multi-user OFDM system which are available for heterogeneous traffics. Their target is to maximize the system's weight capacity in the premise of maintaining users' rates' proportional fairness. Scenario 1, scenario 2 and scenario 3 lead respectively artificial fish swarm algorithm, self-adaptive particle swarm optimization algorithm and cloud adaptive particle swarm optimization algorithm into sub-carrier allocation in packet-dependant proportional fairness scheduling, and use respectively new power allocation pattern, self-adaptive particle swarm optimization algorithm and population migration algorithm to allocate power. Scenario 4 uses greedy algorithm concerning fairness to allocate sub-carriers, and uses new power allocation pattern to allocate power. Simulation indicates scenario 1,scenario 2 and scenario 3 raise the system's total rate on the basis of undertaking the fairness among users' rates and average packet delay; scenario 4 not only meets users' rates and average packet delay demands, but also improve the fairness among users' rates. The four schemes have high application value.

7. REFERENCES

[1] Y.L. Liu, M.Y. Jiang.Adaptive resource allocation in multiuser OFDM system based on hopfield neural networks. JOURNAL OF CIRCUITS AND SYSTEMS,2010,15(2):47-51.

[2] D.X. Yu, Y.M. Cai,D. Wu,W. Zhong.Subcarrier and Power Allocation Based on Game Theory in Uplink OFDMA Systems. Journal of Electronics and Information Technology,2010,32(4):775-779.

[3] L. Peng, M.Y. Jiang. Adaptive cross-layer resource allocation scheme resisting delay sensibility. Application Research of Computers, 2010,27(3):1122-1125.

[4] N.Zhou, X.zhu, Y.Huang, H.Lin. Low Complexity Cross-Layer Design with Packet Dependent Scheduling for Heterogeneous Traffic in Multiuser OFDM Systems. Wireless Com.,IEEE,Jun.2010,9(6):1912– 1923.

[5] Z.K.Shen, J.G. Andrews, B.L. Evans. Adaptive Resource Allocation in Multiuser OFDM Systems With Proportional Rate Constraints. IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, NOVEMBER 2005,4(6): 2726-2737.

[6] X. Ma,Q. Liu. Artificial fish swarm algorithm for multiple knapsack problem. Journal of Computer Applications, 2010,30(2):469-471.

[7] Y.M. Cheng, M.Y. Jiang. Adaptive resource allocation in multiuser OFDM system based on improved artificial fish swarm algorithm. Application Research of Computers, 2009,26(6):2092-2094.

[8] X.J. Bi, W.W. Cao. Adaptive sub-carrier allocation for an orthogonal frequency division multiple access system based on a particle swarm optimization algorithm. Journal of Harbin Engineering University,2010,32(4):775-779.

[9] K. Niu, W.W. Sun, W.J. Xu, Z.Q. He.The Distributed Power Allocation used in OFDMA systems Based on Particle Swarm Optimization Algorithm. China,201010033918.8,2010.

[10] J. Li, Ch. Wang. A modified self-adaptive particle swarm optimization.Journal of Huazhong University of Science and Technology(Natural Science Edition),2008,36(3):118-121.

[11] X.Q. Wei,Y.Q. Zhou,H.J. Huang,D.X. Luo. Adaptive particle swarm optimization algorithm based on cloud theory.Computer Engineering and Application, 2009,45(1):48-50.

[12] A.J.OuYang, W.W. Zhang, Y.Q. Zhou. Hybrid global optimization algorithm based on simplex and population migration.Computer Engineering and Applications, 2010,46(4):29-31.