DYNAMIC SUBCARRIER ALLOCATION FOR SINGLE CARRIER - FDMA SYSTEMS

Obilor Nwamadi, Xu Zhu and Asoke Nandi

Department of Electrical Engineering and Electronics
University of Liverpool
Liverpool, L69 3GJ, UK
Email: {o.nwamadi, xuzhu, a.nandi}@liv.ac.uk

ABSTRACT

This paper explores and compares different dynamic subcarrier allocation (SA) schemes for single carrier (SC) FDMA systems. A so-called maximum greedy algorithm is proposed, which outperforms the existing greedy algorithm. We also provide an optimum solution for SA by using the so-called Hungarian algorithm. Furthermore, all the algorithms presented in this paper are very general and can be extended for OFDMA systems easily. Simulation results show that the investigated dynamic SA schemes significantly outperform the case with fixed SA. With the increase of the number of users, dynamic SA provides improved bit error rate (BER) performance, benefiting from multiuser diversity.

1. INTRODUCTION

Single-carrier frequency division multiple access (SC-FDMA) [10], an alternative to orthogonal FDMA (OFDMA) [2], is a promising transmission technology for future wireless communication systems such as 3GPP-LTE [2]. Unlike OFDMA where each data symbol occupies a single frequency band (subcarrier), the energy of each data symbol in SC-FDMA is spread over several subcarriers, which combats the situations where there are deep fades on certain subcarriers. Therefore, SC-FDMA has a lower peak-to-average power ratio (PAPR), and a higher frequency diversity than OFDMA [10].

Dynamic subcarrier allocation (SA) [2] to different users plays an important role in wireless multi-carrier systems, where the channel environment is time-varying. In [12] adaptive SA was proposed, based on maximizing the utility(rate) of the users for OFDMA systems. However it did not consider fairness of SA, leading to the scenario where a single user with good channel gains on all subcarriers obtains much more resources than others. In [4] an iterative greedy algorithm [7] based SA scheme was proposed for OFDMA systems to maximize the system capacity. In [6] and [13] greedy SA methods were employed, which guaranteed fairness by using the prior knowledge of how many subcarriers are needed by each user. However, all these methods performed SA by allocating one subcarrier at a time, whose complexity becomes prohibitive with a large number of subcarriers [4]. In [5], a cluster-based fair greedy SA scheme was proposed for OFDMA, where each user is allocated an equal-size cluster or chunk of subcarriers each time. Most previous work on SA was based on OFDMA systems, however in [8] a greedy SA algorithm was proposed for the uplink localized SC-FDMA system, using the same cluster scheme proposed in [5], but as mentioned in [8], it is neither necessary nor sufficient for optimality, and an optimum solution was not provided.

In this paper, we investigate dynamic SA for SC-FDMA systems. Our work is different in that we focus on SC-FDMA systems and provide an extensive study of variant SA schemes. We propose a novel so called maximum greedy algorithm which selects the best result from different implementations of the greedy algorithm and it outperforms the existing greedy algorithm [8]. We also provide an optimum solution for SA by using the Hungarian algorithm [9], which was originally employed to find the optimum matching in graph theory [1]. To the best of our knowledge, this is the first application of the Hungarian algorithm for dynamic SA in wireless communications. Furthermore, the proposed maximum greedy and Hungarian algorithms presented in this paper are very general and can be extended for OFDMA systems easily. Simulation results show that the three investigated dynamic SA algorithms (greedy, maximum greedy and Hungarian algorithms) significantly outperform the case with fixed SA. With the increase of the number of users, dynamic SA methods provide improved bit error rate (BER) performance, benefiting from multiuser diversity. Complexity analysis is also provided which compares the computational complexity of the algorithms presented.

The paper is organized as follows, Section 2 presents the system model. Section 3 explains all the dynamic subcarrier allocation algorithms investigated in this paper, Section 4 looks into the computational complexity of the presented algorithms and in Section 5 we present the simulation setup and results. Finally Section 6 concludes the paper.

2. SYSTEM MODEL

A localized SC-FDMA [10] system in the uplink is considered, with U users, as illustrated in Figure 1. Each user has a data block of M symbols which are transformed into the frequency domain by the Fast Fourier Transform (FFT) and then mapped onto the whole subcarrier set of size N ($N = MU$). The mapped data are then transferred back into the time domain by N-point inverse FFT (IFFT), which is denoted by $\mathbf{x}_u$ ($u = 1, \cdots, U$). Each block of N symbols is prepended with a cyclic prefix (CP) before transmission, which is discarded at the receiver to remove the inter-block interference and to make the channel appear to be circular [3]. The received signals are transferred into the frequency domain by N-point FFT, which is followed by subcarrier de-mapping. The frequency domain equalization is performed for each user, and the equalized signals are transferred back into the time do-

This work was supported by the Engineering and Physical Sciences Research Council(Grant No. EP/E026915/1),UK

main by M-point IFFT.

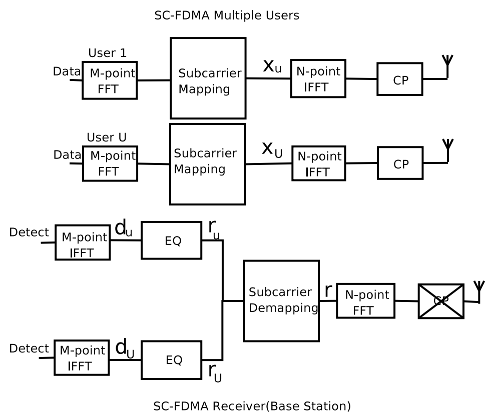

Figure 1: SC-FDMA System

Equations (1) to (3) describe the system in the frequency domain. $\mathbf{r}$ in (1) is the recieved signal for all users.

$$\mathbf{r} = \sum_{u=1}^{U} H_u \mathbf{x}_u + \mathbf{n}_u \tag{1}$$

$\mathbf{H}_u$, $\mathbf{x}_u$ and $\mathbf{n}_u$ represent the channel frequency response matrix, transmitted data and complex AWGN with variance $N_0/2$ per dimension, respectively for a particular user u. $\mathbf{H}_u$ is a $N \times N$ diagonal matrix with $[h_{u1}, h_{u2} \cdots h_{uN}]$ on its diagonal. All others are vectors, with size $N \times 1$. The MMSE equalizer matrix $\mathbf{W}_u$ for each user is given by (2) below

$$\mathbf{W}_u = \mathbf{h}_u^H (\mathbf{h}_u \mathbf{h}_u^H + N_0 \mathbf{I}_M)^{-1} \tag{2}$$

where H is the Hermitian transpose operation, $\mathbf{I}_M$ the $M \times M$ identity matrix and $\mathbf{h}_u$ is the $M \times M$ de-mapped channel matrix for each user, because each user is equalized differently as shown in Figure 1. The equalized data $\mathbf{d}_u$ is given by (3)

$$\mathbf{d}_u = \mathbf{W}_u \mathbf{r}_u \tag{3}$$

where $\mathbf{r}_u$ is each users' $M \times 1$ de-mapped data from $\mathbf{r}$.

Note that in this frequency domain model, the user transmits a null value on the subcarriers that are not allocated to it, leaving it to be used by other users. The subcarrier mapping function performs this operation in the system [8]. Subcarrier mapping for Localized SC-FDMA is shown in Figure 2, where the two shaded subcarriers for each users' row represent the cluster of subcarriers allocated to a particular user.

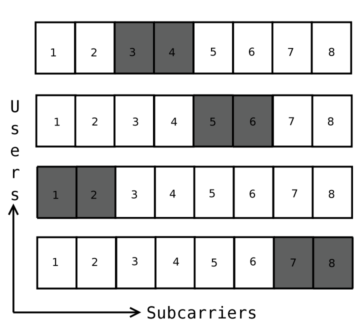

Figure 2: Localized SC-FDMA with $N = 8$ subcarriers, $U = 4$ users and $M = 2$

3. DYNAMIC SUBCARRIER ALLOCATION SCHEMES

The aim of dynamic allocation is to give the user the best resource that will optimize the performance of the whole system. Each user is matched to exactly one resource (cluster, made up of 1 or more subcarriers). The number of subcarriers assigned to each user is given by $K = N/M$ - total number of subcarriers divided by the data length. The clusters for all the users form the matrix $\mathbf{C}$ given below,

$$\mathbf{C} = \begin{bmatrix} c_{11} & \cdots & c_{1K} \\ c_{21} & \cdots & c_{2K} \\ \vdots & \vdots & \vdots \\ c_{U1} & \cdots & c_{UK} \end{bmatrix}$$

where U and K are the number of users and clusters respectively. Each user's subcarrier gain values are kept on the rows of $\mathbf{C}$, cluster by cluster. Each cluster is defined by,

$$c_{uk} = \frac{1}{M} \sum_{n=kM-M+1}^{kM} ||h_{un}|| \ \forall \ k = 1:K \ and \ u = 1:U \tag{4}$$

where k is the cluster number, u is the user number and h_{un} are the diagonal elements from each user's channel matrix $\mathbf{H}_u$.

In this paper, it is assumed that all the users require the same rate and the subcarriers are evenly distributed between them. It is also assumed that all the users have the same power requirements. Furthermore, the base station implements the SA algorithms and the results are fed-back to the users, perfectly and instantaneously. Our $\mathbf{C}$ matrix of subcarrier gains is also viewed as a cost matrix that we are trying to maximize according to the cost function below

$$J_c = \sum_{u=1}^{U} \sum_{k=1}^{K} c_{uk} s_{uk} \tag{5}$$

where c_{uk} are the elements of the cost (cluster) matrix $\mathbf{C}$ and s_{uk} are elements of the choice matrix S. The choice matrix is a binary matrix that specifies the final assigned cluster for each user. It has a 1 on each row which identifies the cluster assigned to a user, and it has a 1 on each column which implies that only one cluster can be assigned to a certain user. Examples of the cost and choice matrices are given below:

$$\mathbf{C} = \begin{bmatrix} 0.7 & \mathbf{4.8} & 2.1 & 8.2 \\ 8.1 & 2.6 & \mathbf{6.2} & 0.5 \\ \mathbf{7.8} & 7.5 & 4.4 & 1.1 \\ 3.2 & 0.2 & 1.1 & \mathbf{7.6} \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

There is a factorial growth of choices, when there are lots of users in the system, the complexity of finding the optimum solution by direct search is highly complex, for example if there are 64 users, there are 64! choices from which the best has to be found.

3.1 Greedy method

A greedy algorithm has a myopic view of the solution space. It analyzes only a local part of the solution space (subspace or subgraph) at a time [7]. In this scenario the greedy method emulates a situation, where a user chooses the best part of the spectrum based on only his/her channel information, rendering that part of the spectrum in-accessible to subsequent users[5]. The following steps show how the greedy algorithm is implemented in this paper:

1. For a certain user, corresponding to a row in cost matrix $\mathbf{C}$ find the cluster of subcarriers with highest gain. Assign it to this user.
2. Remove this cluster of subcarriers from service.
3. Move on to the next user and perform steps 1 and 2 again, until all the users have been allocated.

Notice from the example below that there is a bias inherent in this algorithm, the available clusters reduce by 1 after each iteration, therefore the last user is not left with any choice.

$$\begin{bmatrix} 0.7 & 4.8 & 2.1 & \mathbf{8.2} \\ 8.1 & 2.6 & 6.2 & 0.5 \\ 7.8 & 7.5 & 4.4 & 1.1 \\ 3.2 & 0.2 & 1.1 & 7.6 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ \mathbf{8.1} & 2.6 & 6.2 & 0 \\ 7.8 & 7.5 & 4.4 & 0 \\ 3.2 & 0.2 & 1.1 & 0 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \mathbf{7.5} & 4.4 & 0 \\ 0 & 0.2 & 1.1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1.1 & 0 \end{bmatrix}$$

$$\mathbf{S}_g = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

S_g is the final choice matrix, which is clearly not the optimum result.

3.2 Maximum Greedy method

In this paper we propose a Maximum Greedy algorithm for dynamic SA. This algorithm selects the best or maximum from various implementations of the greedy algorithm that is run on different local areas of the total solution space. This is clearly more complex than the greedy method explained above, but a trade-off between the quality of the solution desired and the complexity is required. The steps in this algorithm are given below:

1. Determine the different suitable local areas of the solution space.
2. For each unique area, perform the greedy algorithm above and save the solution.
3. Find the best from saved greedy solutions.

The example below emphasizes the point that changing the order in which the greedy algorithm is performed for the cost matrix $\mathbf{C}$, gives a different result. In this case, the greedy algorithm is run starting from the last user to the first, note that $\mathbf{S}_{mg}$ is optimum. The orders represent different local areas of the solution space and can be initialized either randomly or in a specific way if the structure of the solution space is known.

$$\begin{bmatrix} 0.7 & 4.8 & 2.1 & 8.2 \\ 8.1 & 2.6 & 6.2 & 0.5 \\ 7.8 & 7.5 & 4.4 & 1.1 \\ 3.2 & 0.2 & 1.1 & \mathbf{7.6} \end{bmatrix} \rightarrow \begin{bmatrix} 0.7 & 4.8 & 2.1 & 0 \\ 8.1 & 2.6 & 6.2 & 0 \\ \mathbf{7.8} & 7.5 & 4.4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} 0 & 4.8 & 2.1 & 0 \\ 0 & 2.6 & \mathbf{6.2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 4.8 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{S}_{mg} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3.3 Hungarian Method

The Hungarian algorithm [9] was initially used to find the optimum matching of a *bipartite graph*. It is also used to solve *The Assignment Problem* in graph theory [9][1]. It employs the use of iterative row and column reductions to find the minimum cost of a complete matching given a certain cost matrix, the algorithm is briefly explained with the following steps:

1. Find minimum of each row in cost matrix $\mathbf{C}$ and subtract it from the corresponding row. Zeros should appear on each row.
2. Check if there are zeros on each column also, if yes, jump to step 3, if no, perform step 1 on the columns. Now there should be zeros on the row and columns.
3. Try to cover the zeros with the *minimum* number of lines (horizontal or vertical) in the reduced cost matrix. If minimum number of lines equals K (size of square cost matrix), then final solution is reached.
4. Else find the minimum cost in the uncovered part of the cost matrix, and subtract it from the uncovered rows, then add it to the covered columns. Repeat steps 3 and 4 until solution is found.

If finding the maximum is required, find the minimum of the $-\mathbf{C}$. An example is given below, using $\mathbf{C}$ given earlier,

$$\begin{bmatrix} -0.7 & -4.8 & -2.1 & -8.2 \\ -8.1 & -2.6 & -6.2 & -0.5 \\ -7.8 & -7.5 & -4.4 & -1.1 \\ -3.2 & -0.2 & -1.1 & -7.6 \end{bmatrix} \rightarrow \begin{bmatrix} 7.5 & 3.4 & 6.1 & 0 \\ 0 & 5.5 & 1.9 & 7.6 \\ 0 & 0.3 & 3.4 & 6.7 \\ 4.4 & 7.4 & 6.5 & 0 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} 7.5 & 3.1 & 4.2 & 0 \\ 0 & 5.2 & 0 & 7.6 \\ 0 & 0 & 1.5 & 6.7 \\ 4.4 & 7.4 & 4.6 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 4.4 & 0 & 1.1 & -3.1 \\ 0 & 5.2 & 0 & 7.6 \\ 0 & 0 & 1.5 & 6.7 \\ 1.3 & 4.3 & 1.5 & -3.1 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} 4.4 & \mathbf{0} & 1.1 & 0 \\ 0 & 5.2 & \mathbf{0} & 10.7 \\ \mathbf{0} & 0 & 1.5 & 9.8 \\ 1.3 & 4.3 & 1.5 & \mathbf{0} \end{bmatrix} \qquad \mathbf{S}_H = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Observe that after step 1, there are 2 columns that don't have any zero on it. Perform step 2. Two more zeros are added to the system, but all the zeros can be covered by 3 lines (the 2 middle rows and the 4th column) which is less than 4. The minimum cost, 3.1 is subtracted from the top and bottom rows (uncovered rows), this introduces negative numbers on the covered column. Adding 3.1 to the covered column (4th column) removes the negative numbers and adds more zeros to the system. For each iteration of steps 3 and 4, more zeros will be added to the system, with the last set of zeros guaranteed to still be in place [9]. It is worth noting, from the example above, there might be more than K zeros in the final matrix, but only the important zeros are considered - a single zero on a row or a column or on both a row and column. After selecting this zero, the next zero is found from the cofactor matrix (matrix remaining after the row and column corresponding to the first zero are removed), this continues until the last zero is assigned. Finally $\mathbf{S_H}$ in the example above, has 1's in the positions of chosen zeros.

4. COMPLEXITY ISSUES

The computational complexity of the Hungarian algorithm cannot be exactly estimated every-time because of the randomness of the cost matrices and the heuristic nature of the algorithm, however in [9] the estimated upper-bound on the number of operations is $(11K^3 + 12K^2 + 31K)/6$ where K is size of the square cost matrix. Based on this, the complexity order for the Hungarian algorithm is approximated by $O(K^3)$ as shown in Table 1. As expected, the greedy algorithm is less complex than the Hungarian algorithm, because the search space for the greedy algorithm constantly reduces after each iteration. In [5] the number of operations for the greedy algorithm is estimated as $\frac{1}{2}K(K-1)$. So for large K, the complexity order of the greedy algorithm is approximated by $O(K^2)$. The novel maximum greedy algorithm proposed is more complex than the greedy algorithm by a factor of a, where a is determined by how many greedy solutions searched. Care should be taken in choosing a so that the complexity does not become very large.

In Table 2 the normalized numerical complexity values are estimated using the number of operations formulae and normalized by the greedy algorithm. A 128 user system is assumed with cluster/cost matrix size 128×128. All the methods presented in this paper are still less complex than a direct search of $K!$ solutions, especially when K is large. In [11] a list of optimum methods are highlighted and some of them have complexities comparable to greedy algorithm, an example is the Edmonds and Karp's algorithm with $O(K^2 log\, K)$. Optimized data structures [7] and faster computers currently available, can be used to improve the efficiency of these algorithms.

5. SIMULATION RESULTS

Our simulations use a fixed amount of subcarriers $N = 256$, that are equally divided among the users. The normalized delay spread is approximately 1 and the basic SC-FDMA model explained in Section 2 is used, the users have a single antenna and the base station has a single antenna. All data are QPSK modulated (baseband equivalent), and Minimum Mean Square Equalization (MMSE) is used for detection. $a = K$ is used for the maximum greedy algorithm simulations. A thousand errors are averaged for each SNR level, Monte Carlo style.

Algorithm	complexity	# of operations
Hungarian	$O(K^3)$	$(11K^3 + 12K^2 + 31K)/6$
Greedy	$O(K^2)$	$\frac{1}{2}K(K-1)$
max. Greedy	$O(aK^2)$	$\frac{a}{2}K(K-1)$

Table 1: Complexity Order for the presented SA algorithms

Algorithm	Normalized Complexity
Hungarian	2863
Greedy	1
Max.Greedy $a = 128$	128
Direct Search	128! = ∞

Table 2: Normalized complexities for the presented SA algorithms

It is easily observable in Figure 3 that at all user average BER of 10^{-4} the Hungarian algorithm has over 20dB gain when compared to the fixed SA case. It also outperforms the greedy and maximum greedy algorithms by about 6dB and 2dB respectively, while the maximum greedy algorithm outperforms the greedy algorithm by about 4dB. Furthermore, for better BER performance(10^{-6}), the Hungarian algorithm outperforms the greedy and maximum greedy algorithms by about 5dB and 15dB respectively, with the maximum greedy algorithm having a gain of about 10dB over the greedy algorithm. So for different applications a trade-off can be made on which SA scheme to use, for speech applications where low BER performance is suitable, the greedy algorithm can be employed, but for text or video applications, a better SA scheme may be used.

Figures 4 and 5 show the positive impact (BER improvement) with the increase of the number users in the system, in contrast to the fixed case where the BER degrades with increasing users. These figures also show that the Hungarian algorithm outperforms the greedy and maximum greedy algorithms, whereas the maximum greedy algorithm outperforms the greedy algorithm. Comparing these two figures, it is observable that there is difference in performance between the algorithms when the SNR is greater - the gain as users increase is greater in Figure 5 for 15dB than in Figure 4 for 10dB.

6. CONCLUSION

In this paper, we proposed dynamically allocating subcarriers to different users in a Single Carrier - FDMA uplink system. We have shown that using the Hungarian algorithm far out performs existing greedy algorithms in terms of BER, but with a higher computational complexity. We also proposed a novel maximum greedy algorithm for SA, that outperforms

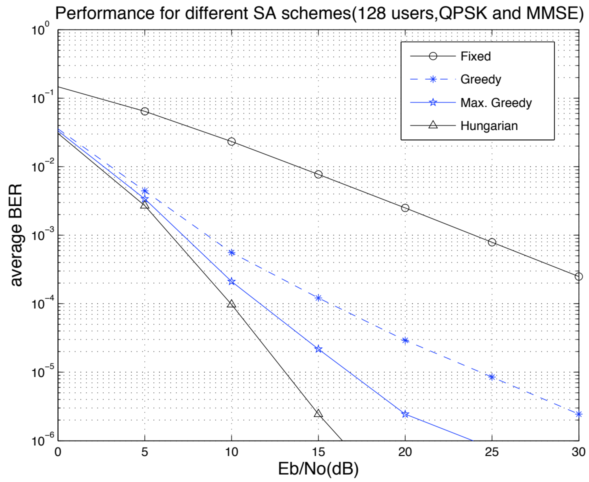

Figure 3: Different SA schemes for $U = 128$ users, $N = 256$ subcarriers and $a = 128$ for max. greedy algorithm

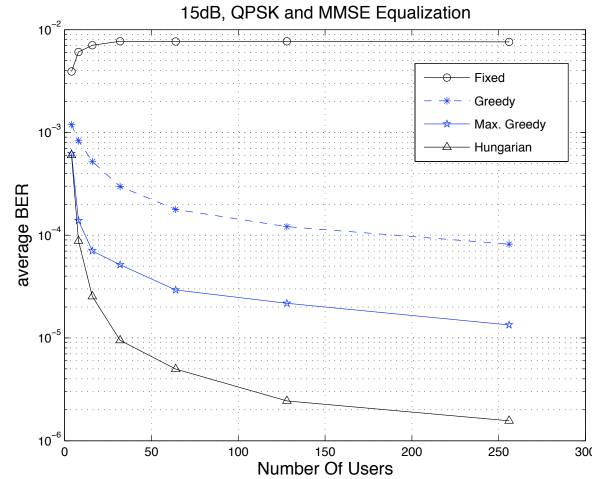

Figure 5: Effects of increasing users for different SA schemes with $SNR = 15dB$

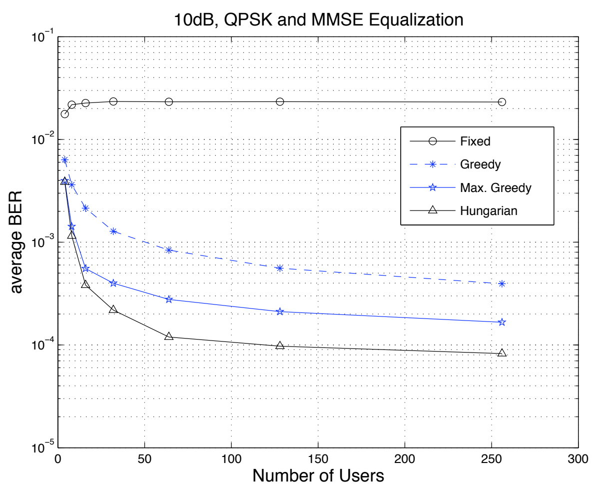

Figure 4: Effect of increasing users for different SA schemes with $SNR = 10dB$

the existing greedy SA algorithm. Finally we conclude that effectively carrying out dynamic SA greatly increases the efficiency of multi-user communication systems, thus more effort should be put into these areas of research.

REFERENCES

[1] Bondy, J., A. and Murty, U., S., R., *Graph Theory with Applications*, Macmillan press, London, 1976.

[2] Dahlman, E., Parkvall, S., Skold, J. and Beming, P., *3G Evolution: HSPA and LTE for Mobile Broadband*, Academic Press Elevier, 2007.

[3] Haykin, S. and Moher, M., *Modern Wireless Communications*, Pearson Prentice Hall, New Jersey, 2005.

[4] Keunyoung K., Hoon, K., Youngnam, H. and Seong-Lyun, K., "Iterative and greedy resource allocation in an uplink OFDMA system" in proc. *15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, PIMRC, Barcelona, Spain, September, 2004, pp. 2377-2381.

[5] Kim, Y. and Kim, J., "A 2-D subcarrier allocation scheme for capacity enhancement in a clustered OFDM system," *IEICE transactions on communications*, vol. E90-B, issue 7, pp. 1880–1883, July 2007.

[6] Kivanc, D. and Liu H., "Subcarrier Allocation and Power Control for OFDMA" in proc. *34th Asilomar conference on signals, systems and computers*, Pacific Groove, CA, 29 Oct. - 1 Nov. 2000, pp. 147–151.

[7] Klienberg, J. and Tardos, E., *Algorithm Design*, Pearson Education Inc. Boston, 2006.

[8] Lim, J., Myung, H. G., Oh, K. and Goodman, D. J., "Channel-Dependent Schedulling of Uplink single carrier FDMA systems" in proc. IEEE *Vehicular technology conference*, Montreal Canada, VTC fall 2006, pp. 1–5.

[9] Munkres, J., "Algorithms for the assignment and transportation problems," *Society for Industrial and Applied Mathematics*, Vol. 5 issue 1, pp. 32–38, March 1957.

[10] Myung, H. G., *Single carrier orthogonal multiple access technique for broadband wireless communications*, Dissertation, Polytechnic University, NY, 2007

[11] Saip, H., A., B. and Lucchesi, C., L., *Matching Algorithms for Bipartite Graphs*, Relatorio Tecnico DCC-03, 1993.

[12] Song, G and Li, Y., "Adaptive subcarrier and power allocation in OFDM based on maximazing utility " in proc. IEEE *Vehicular technology conference*, Jeju, Korea, VTC spring 2003, pp. 905–909.

[13] Teo, K. A. D., Otani, Y. and Ohno, S., "Adaptive subcarrier allocation for Multi-user OFDM system," *IEICE transactions fundamentals*, vol. E89-A, issue 11, pp. 3131–3137, November 2006.