

Research Article

OBBP: An Efficient Burst Packing Algorithm for IEEE802.16e Systems

Omar M. Eshanta, M. Ismail, and Kasmiran Jumari

Faculty of Engineering, National University of Malaysia, 43600 Bangi Selangor, Malaysia

Correspondence should be addressed to Omar M. Eshanta, omarshanta@gmail.com

Received 1 December 2010; Accepted 6 January 2011

Academic Editor: C. Carbonelli

Copyright © 2011 Omar M. Eshanta et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mobile communications have witnessed a phenomenal increase in the amount of users, services, and applications. Orthogonal Frequency Division Multiple Access (OFDMA) targets to provide broadband connectivity to wide area coverage, in mobile environments, for Next Generation Networks (NGNs), which results in significant design challenges in the MAC (Medium Access Control) layer to provide an efficient resource allocation in a cost-effective manner. This paper proposes a two-dimensional (2D) Burst Packing (BP) algorithm for OFDMA downlink (DL) subframe that can provide the service providers with an efficient, fast, flexible, and high-spectral efficiency method to allocate system resources among the Mobile Stations (MSs). The proposed Orientation-Based Burst Packing (OBBP) algorithm uses the Orientation Factors (OFs) of the bursts as a criteria to achieve the challenging issues in the BP problem. The simulation results show that OBBP algorithm can achieve a high packing efficiency up to 99.2% when the burst size ratio (BSR) is 50%.

1. Introduction

The next generation of wireless communications have adopted OFDMA as an access technology for their mobile networks. The great benefit of OFDMA comes from its PHY layer features which were designed based on the Orthogonal Frequency Division Multiplexing (OFDM) technology. OFDM technique which has been adopted for Fixed Line-of-Sight (LOF) WiMAX networks, divides the available broad bandwidth into smaller narrow and orthogonal subbands called subcarriers. With this orthogonality, OFDM system can mitigate the intercarrier interference (ICI) efficiently and simply by using Fast Fourier Transform (FFT) and its Inverse (IFFT) techniques. These features with the robustness against intersymbol interference (ISI) give OFDM-based systems an added advantage over non-OFDM systems.

Beside these features, OFDMA technology divides the available physical subcarriers into logical groups called *subchannels*, where each subchannel contains a certain number of subcarriers that can be physically adjacent or distributed according to the subchannelization scheme. Adjacent scheme is suitable for fixed networks, where the channel has

slow fading and time invariant characteristics, while in mobile networks where the channel is time and frequency selective the distributed subchannelization scheme is more suitable. Two *permutation* schemes have been proposed in the standard [1, 2] for distributed subchannelization called Full Usage of Subchannels (FUSC) and Partial Usage of Subchannels (PUSC) and one for the adjacent scheme called Adaptive Modulation and Coding (AMC). All of these different schemes can be used in the same DL-subframe in different time slots as shown in Figure 1. The two-dimensional (time and frequency) OFDMA DL-subframe has a minimum resource unit called *slot*, each slot represents one subchannel by one or two symbols for FUSC or PUSC, respectively. The DL-subframe is the container that carries the user's data (*bursts*) by allocating a number of slots to each Mobile Station (MS). Some issues in the allocation and scheduling of users data units has been left unspecified by the IEEE 802.16 and 802.16e standards [1, 2] to give the operators a flexibility and variety to design their own efficient and optimal solutions. However, due to the limitations of the PHY layer some restrictions in the shape and the number of the data bursts have been stated in the standard. One

of these restrictions is the rectangular shape of the bursts which makes the problem of BP very complicated. In this paper we present an efficient and simple algorithm named Orientation-Based Burst Packing (OBBP) algorithm to pack the user's data bursts into the DL-Subframe container. This problem has been investigated in the literature from different viewpoints [3–16]. The authors in [3] presented the Less Flexibility First (LFF) algorithm as an efficient solution. The algorithm works by defining the flexibility of empty space and the flexibility of objects to be packed, then chooses the less flexible object first to be placed in the less flexible empty space. The main drawback of this algorithm is the computational complexity which can cause a notable delay in the real-time applications. A good proposal by Cohen and Katzir [4] combines the packets with the same Modulation and Coding Scheme (MCS) into one burst, so the bursts become larger and easier to be packed efficiently. But the problem is that mobile Stations have to decode the entire burst to obtain its own data, which leads to a higher power consumption of the MS's battery. A recent work by Zubow et al. [5] proposes a Greedy Scheduling Algorithm (GSA) as a solution to the BP problem and as a mechanism to mitigate the intercell interference taking into account the MS preferences in the burst shape. To the best of our knowledge there is no work in the state of the art that has simultaneously addressed as many performance aspects as [5] does. But the GSA algorithm did not bring much benefit to the main BP problem that they call "the core packing engine". The core packing engine of the GSA uses the same mechanism proposed by other related works which are mainly based on the search for the best location in the current available free space to allocate each burst independently, which makes the algorithm design complicated and time consuming. Our OBBP algorithm differs from other algorithms in that the packing is based on a common feature between the bursts which is the Orientation Factors (OF: defined as the most common orientation among all possible orientations of the variable dimensions rectangles). This makes the OBBP algorithm very simple, fast, and efficient as validated by our simulations. The rest of this paper is organized as follows: Section 2 presents a brief background about OFDMA MAC layer and DL-Subframe structure, while Section 3 presents the general BP and its challenges in OFDMA networks. The proposed OBBP algorithm will be detailed in Section 4. The performance evaluation, conclusion, and future work will be discussed on Sections 5 and 6.

2. OFDMA MAC Layer Background

OFDMA has been adopted by IEEE 802.16e [2] as the primary channel access mechanism for non-line-of-sight (NLOS) communications in the frequency bands below 11 GHz. Sections 2.1 and 2.2 describe the major components of OFDMA MAC layer.

2.1. OFDMA MAC Scheduler. The IEEE 802.16e MAC Scheduler (Figure 2) is the major component of resource management development in OFDMA system. The scheduling algorithm which runs in the Base Station (BS) receives

the classified Protocol Data Units (PDUs) from the classifier and creates bursts at the output according to the QoS or other utility criteria. There are two possible schemes used to pack the scheduled bursts into the DL-subframe [5], *online* scheme where the bursts packed independently one by one and *offline* scheme where the packed bursts queued first then packed all together using the packing algorithm. In this work we consider only the offline scheme which gives more flexibility in designing the algorithm and higher packing efficiency.

2.2. DL-Subframe Structure in IEEE 802.16e. As stated above the basic unit of resource allocation in OFDMA is the *slot*, which comprises a number of symbols in time domain, and one subchannel in frequency domain. The total frame duration divided with a certain ratio into two contiguous *subframes*, downlink (DL) and an uplink (UL) subframes. As illustrated in Figure 1, a DL subframe starts with a preamble, which helps mobile stations perform synchronization and channel estimation. The first two subchannels in the first *data* OFDMA symbol in the downlink is called frame control header (FCH), which shall be transmitted using QPSK rate 1/2 with 4 repetitions. The FCH specifies the length of the immediately succeeding downlink MAP (DL-MAP) message and the repetition coding used for DL-MAP. BS uses a downlink MAP (DL-MAP) and an uplink MAP (UL-MAP) message to notify MSs of the corresponding resources allocated to them in the downlink and uplink direction, respectively, within the current frame. Based upon the schedule received from the BS, each MS can determine when (i.e., which OFDMA symbols) and where (i.e., which subchannels) should receive from and transmit to BS. *Zone* is another key concept introduced in 802.16 standard [2], which refers to a number of contiguous OFDMA symbols in the downlink that use the same permutation scheme. The technique of *zoning* has been extensively used in 802.16e to accommodate a wide variety of antenna and physical layer configurations. BS would inform MSs of the location, format, and length of each zone by using certain *information element* (IE) in the DL-MAP and UL-MAP.

3. General BP Problem

The importance of designing an efficient OFDMA BP (OBP) algorithm arises from several facts; first, the resources in terms of data units (slots) are limited and should be used efficiently. Second, OFDMA system supports QoS provisioning which must be considered in designing OBP algorithm. Third, the OBP algorithm must work in compatible with the MAC scheduler to meet the QoS requirements. All of these issues make the OBP problem a big challenge for developers and service providers.

3.1. Major Issues in Designing BP Algorithm in OFDMA System. The major challenges in designing OBP algorithm can be summarized in three points.

- (i) Not like the conventional Bin Packing problem where the rectangle has a predetermined dimension, the

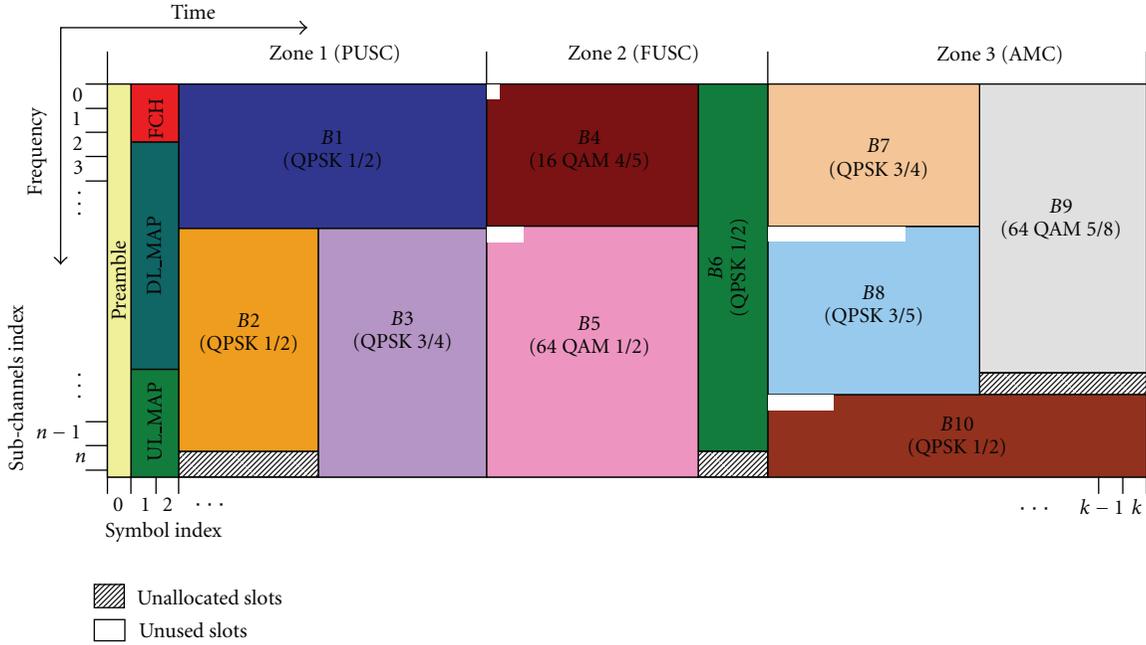


FIGURE 1: OFDMA DL-subframe structure.

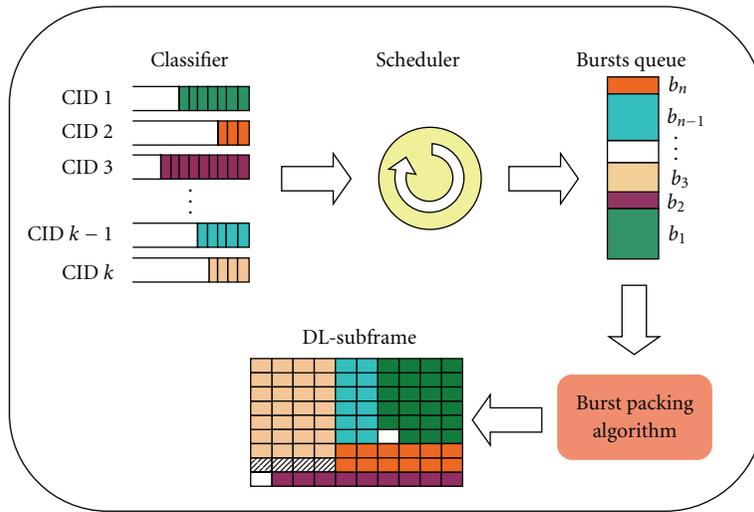


FIGURE 2: IEEE 802.16e MAC scheduler.

burst in OBP can have more than one dimension (more than one OF).

- (ii) In OBP each burst must have a rectangular shape, so some bursts must be padded to satisfy this condition, which reduces the packing efficiency.
- (iii) The number of slots used by DL_MAP is variable and depends on the number of bursts, which makes the unallocated slots (free space) in DL-Subframe in OBP unpredictable.

To fulfill these challenges the OBP algorithm must choose the optimum OF for each burst that allows the maximum number of bursts to be packed in the DL-subframe, the number of padded slots must be minimized and the number

of bursts also must be minimized (i.e., use *fragmentation* as little as possible).

4. Proposed OBBP Algorithm

The principle of OBBP algorithm is to group up the bursts based on the orientation factor (OF) and pack the bursts of each group in columnwise or row-wise in the DL-subframe. Since we deal with groups instead of independent single bursts we minimize the complexity and the processing time of the packing process and the packing efficiency improves as well. The algorithm divided into three stages: first stage is a prepacking stage to prepare and classify the bursts, second

stage packing the bursts using the common OF, and third stage packing the remaining bursts from second stage using the conventional best fit (BF) packing algorithm. Before we describe the details of these stages we need to clarify some assumptions.

- (i) The entire DL-subframe represents a single PUSC permutation zone.
- (ii) The communication channel has slow fading characteristics, that is, SINR is almost constant during the frame duration.

The details of OBBP algorithm are described next.

4.1. Prepacking Stage (Stage I). In this stage the algorithm receives a set of k bursts $B = \{b_1, b_2, b_3, \dots, b_{k-1}, b_k\}$ which has been assigned by the scheduler to be sent in the next frame. Note that the total size of bursts in B can be greater than the capacity of the DL subframe, (i.e., $\sum_{i=1}^k c_i > S$), where c_i is the size (number of slots) of burst b_i and S is the total number of available slots (free space) in DL subframe. In this stage the OBBP algorithms will perform the following operations:

- (a) *Priority Sorting.* The set of bursts B has been sorted by the scheduler based on their priorities and QoS preferences so that the highest priority burst is the first. Because the total number of slots in B can be greater than S , we take the first m bursts that has a total number of slots less than or equal to S , (i.e., $\sum_{i=1}^m c_i \leq S$), we call this new set of bursts B' and keep the remaining bursts from that, $B - B' = \{b_m, b_{m+1}, b_{m+2}, \dots, b_{k-1}, b_k\}$, to be allocated in the second stage if possible.
- (b) *OF Calculation.* We mean by the orientation factors the set of all possible orientations of each rectangle. For example, the orientation factors for a burst of size 14 slots is $\{1 \times 14, 14 \times 1, 2 \times 7, 7 \times 2\}$, (i.e., $\text{OF} = \forall i \leq m, c_i \bmod j = 0, j = 1, 2, 3, \dots, b_i/2$). The code of Algorithm 1 shows how OBBP calculates these OF.
- (c) *Burst Adaptation.* This step performed in two cases: (1) when the burst has one or more OF out of the frame range and (2) when the burst has no orientation factors within the frame range. In the first case if the burst has some OF out of the frame dimensions we remove those OF and keep only the in-range OF; an example for this is case when we need to allocate a burst of size 88 slots into a frame of size 60×28 (i.e., $N_{\text{Sch}} = 60, N_{\text{Symb}} = 28$ for channel BW 20 MHz and DL/UL ratio 28/19), the OFs of this burst are $\{1 \times 88, 88 \times 1, 4 \times 22, 8 \times 11\}$: the first two OFs are out of range and can not be used as OF. An example of the second case is a burst of size 87 slots which has only two OFs $\{87 \times 1, 1 \times 87\}$, both of these OFs are out of range and this burst cannot be allocated with this size. To solve the second case problem we increment the size of this burst to become 88 which has within range OFs $\{4 \times 22, 8 \times$

11}. The procedure of this step is described in the lines 8–10 of Algorithm 1.

- (d) *Constructing OF Matrix.* After calculating the possible OF in step (c), in this step we build a matrix of size $N_{\text{Sch}} \times N_{\text{Symb}} = S$ (in our example the matrix will be 60 rows and 28 columns), with all elements equal to zero, then according to the OF of each burst we set the corresponding elements of the matrix to the burst size. For example if $B' = \{10, 6, 8, 5\}$ OFs which has $\{10 \times 1, 1 \times 10, 2 \times 5, 5 \times 2\}$, $\{6 \times 1, 1 \times 6, 3 \times 2, 2 \times 3\}$, $\{8 \times 1, 1 \times 8, 4 \times 2, 2 \times 4\}$, and $\{5 \times 1, 1 \times 5\}$, respectively, will be allocated in the matrix as follows:

$$\text{OF}_{\text{Matrix}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 5 & 6 & 0 & 8 & 0 & 10 \\ 0 & 0 & 6 & 8 & 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (1)$$

We notice that the OF matrix is a symmetric matrix (i.e., the elements above and below the main diagonal are symmetric) for this reason OBBP algorithm can use the matrix to allocate the bursts vertically (columnwise) as well as horizontally (rows-wise) similarly. But we have to notice that if some bursts have been adapted in the previous step (c), this matrix will become asymmetric. For example if we have burst of size 62 slots which has OF set $\{62 \times 1, 1 \times 62, 31 \times 2, 2 \times 31\}$ after adaptation process we keep only the $\{31 \times 2\}$ which has no copy on the other side of the main diagonal of the OF matrix, and this is a reason why the two packing scenarios, vertical and horizontal, give different results.

4.2. Main Packing Algorithm (Stage II). As we mentioned above, the main idea of our algorithm is to minimize the complexity of finding the optimum location for each burst within the available free space. We achieve this by minimizing the irregularity of the free space (unallocated slots) shape produced after each burst packing. To understand the advantage of this strategy we use the two different scenarios shown in Figure 3. Figure 3(a) represents our strategy, where the Packing starts from the right-left corner (later we will discuss the reason behind this choice) and tries to fill up the N rows (total number of subchannels) with bursts which have common number of columns (symbols). On the other hand, the conventional method in Figure 3(b), allocates bursts independently one by one by finding the most suitable location for each burst. This strategy produces a deformed shape of the free space and divides the big rectangles into smaller ones that can not fit for big bursts without fragmentation.

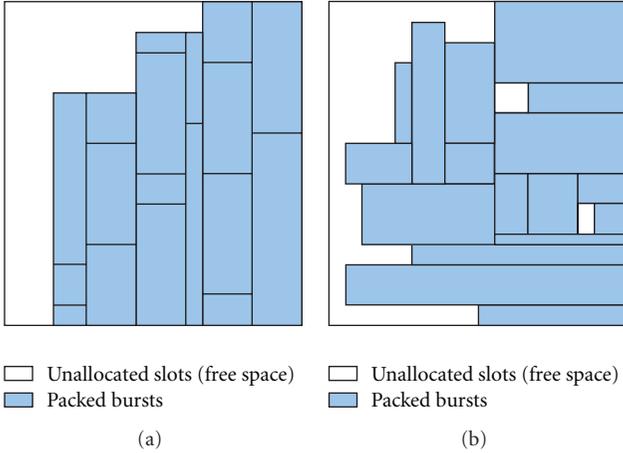


FIGURE 3: (a) OBBP strategy, (b) conventional strategy.

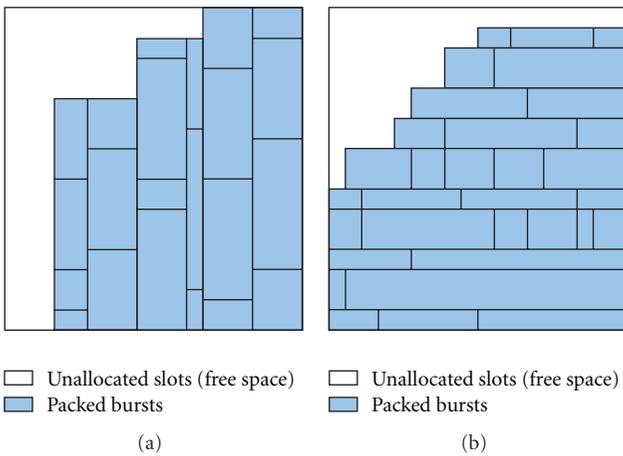


FIGURE 4: (a) Vertical scenario, (b) horizontal scenario.

For clarity of demonstration we divide the main algorithm into three sections: (a) packing set selection, (b) packing set arrangement, (c) packing set stuffing.

4.2.1. Packing Set Selection. From the OF matrix we can classify the bursts according to their common number of symbols (vertically) or subchannels (horizontally), for example, the bursts $\{6, 8, 10\}$ can be allocated vertically using the same number of symbols which is 2, and also horizontally using the same number of subchannels which is also 2.

To obtain the maximum efficiency of OBBP algorithm we start the packing with the maximum possible number of slots, this can be determined by calculating the maximum sum of columns or rows elements of the OF matrix. Note that the set of bursts that have the maximum sum can be many of small-size bursts, few of large-size bursts, or a combination of small and large. Allocating small-size bursts first has two disadvantages: firstly it does not support the QoS provisioned by OFDMA systems which usually give the highest priority to the large-size bursts (in heterogeneous traffic systems, large-size burst means high data rate service) and secondly it

minimizes the probability of finding free space for the big-size bursts in the next stage which minimizes the packing efficiency. On the other hand allocating large-size bursts first deforms the remaining free space. To compromise this problem, OBBP algorithm allocates the large-size bursts first and leaves the small-size bursts to be packed in the remaining free space of the current column (In all of our examples we assume that the allocation scenario is vertical.) if possible. The following example demonstrates the operation of this step.

- (a) From OF matrix, find the column that has the maximum sum, we call the subset of bursts in this column, B'_{\max} . Let us say that the column number of B'_{\max} is 2 and that $B'_{\max} = \{4, 100, 12, 88, 24, 48, 8, 80, 84, 40\}$ and $N_{\text{Sch}} = 60$.
- (b) Sort B'_{\max} in descending order, so $B'_{\max} = \{100, 88, 84, 80, 48, 40, 24, 12, 8, 4\}$.
- (c) The rectangles height of B'_{\max} , called $B'_{\max, h} = \{25, 22, 21, 20, 12, 10, 6, 3, 2, 1\}$.
- (d) Note that $(\sum B'_{\max, h} = 122) > N_{\text{Sch}}$, so we need to find the optimum subset of bursts $B'_{\max, \text{opt}}$ where $B'_{\max, \text{opt}} \subseteq B'_{\max}$ that satisfies the following condition:

$$\min \left(\sum B'_{\max, \text{opt}, h} - N_{\text{Sch}} \right). \quad (2)$$

In this example we have two possible optimum subsets from $B'_{\max, h}$ which has sum equals to N_{Sch} , $\{25, 22, 12, 1\}$ and $\{25, 22, 10, 2, 1\}$ and the corresponding subsets from B'_{\max} are $\{100, 88, 48, 4\}$ and $\{100, 88, 40, 8, 4\}$, respectively. As explained above, the decision of which subset should be allocated in this iteration of OBBP algorithm makes a notable effect on the packing efficiency.

After the optimum subset has been selected, we set the position of this subset in OF matrix to zero, and then repeat the steps from 1 to 4 using the new updated OF matrix until the end of bursts.

4.2.2. Packing Set Arrangement. After the packing sets has been selected from step (a), and according to our assumptions in Section 4, moving the burst from one position to another within the sub-frame will not affect the QoS provisioning. So, in this stage we rearrange the subsets assigned in step (a) by sorting them in descending order according to the condition in (2), so the most optimum subsets will be the first which forms a stairs-like shape as shown in Figure 3(a).

4.2.3. Packing Set Stuffing. Packing bursts starts from the bottom-right corner and grows in the two dimensions, so the unallocated slots (the free space) will be at the top-left corner where the DL_MAP can be extended if fragmentation used or extra bursts added to fill up the free space. This process continues until the end of subsets or no more columns in the subframe can fit for the remaining subsets. The remaining bursts, if any, will be transferred to the third stage.

```

(1) UnusedCount = [ ];
(2) for i = 1:length(B')
(3)   X = B'(i);
(4)   F = [ ];
(5)   for j = 1:X/2
(6)     F(j) = mod(X, j);
(7)   end
(8)   all_OF = [find(F==0)X];
(9)   OF{i} = all_OF(find(all_OF ≤ NSch & all_OF ≤ NSymb));
(10)  OF{i} = OF{i}(numel(all_OF)-numel(OF{i})+1:end);
(11) end
(12) if isempty(find(numel(OF{i})==0))==0
(13) // increment bursts which have out of range OF.
(14) B'(find(numel(OF{i})==0)) = B'(find(numel(OF{i})==0)) + 1;
(15) // counting unused slots.
(16) UnusedCount(find(numel(OF{i})==0)) = UnusedCount(find(numel(OF{i})==0)) + 1;
(17) end

```

ALGORITHM 1:

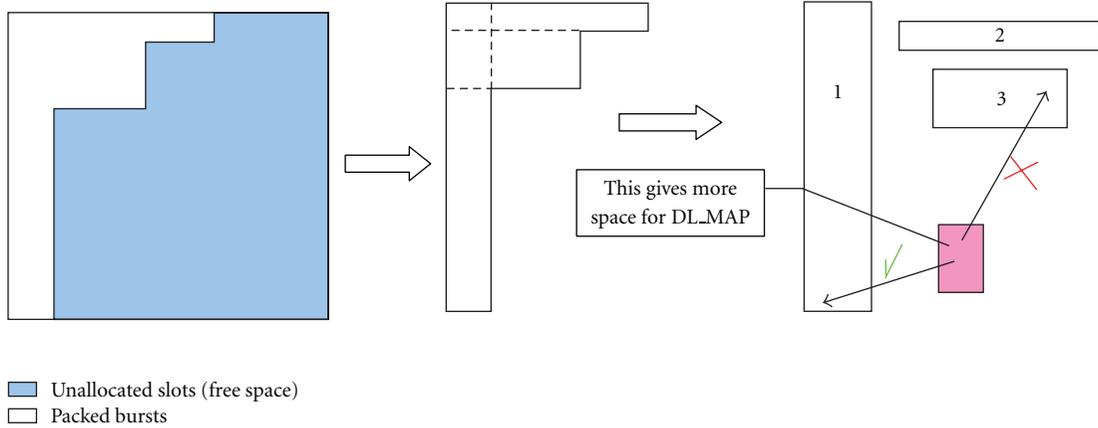


FIGURE 5: Free space rectangles selection.

Figure 4 shows an example of the output from stage II of OBBP algorithm. As we can see the free space is regular and easy to be allocated efficiently as we will see in next section.

4.3. Module II of OBBP Algorithm (Stage III). The remaining bursts from stage II which have OF unfit for the available regular free space (number of adjacent columns or rows) will be considered in this stage. In this stage the unallocated slots in the subframe (free space) divided into overlapped rectangles with its maximum possible dimensions as shown in Figure 5. In this figure, the possible rectangles that can be generated from the free space shape are three overlapped rectangles with its maximum possible dimensions. The following steps summarize the operation of this stage.

(a) Sort the remaining set of bursts B'_{Rem} in descending order.

(b) Find the optimum rectangle for the first burst in B'_{Rem} , the optimum means the rectangle that satisfies the following condition:

$$\min_{j=1}^q (r_j - c_i), \quad (3)$$

where r_j is the size of rectangle j , c_i is the size of burst i and q is the number of available rectangles. Notice that the first burst is the biggest burst, so in this stage also, we still give the highest priority to the big size bursts.

(c) If we success in step (b) and found a suitable rectangle j , we pack in the burst i in the bottom-right corner of the rectangle j , and if the burst have more than one OF we select the one which maximizes the dimensions of the remaining free space (i.e., try to

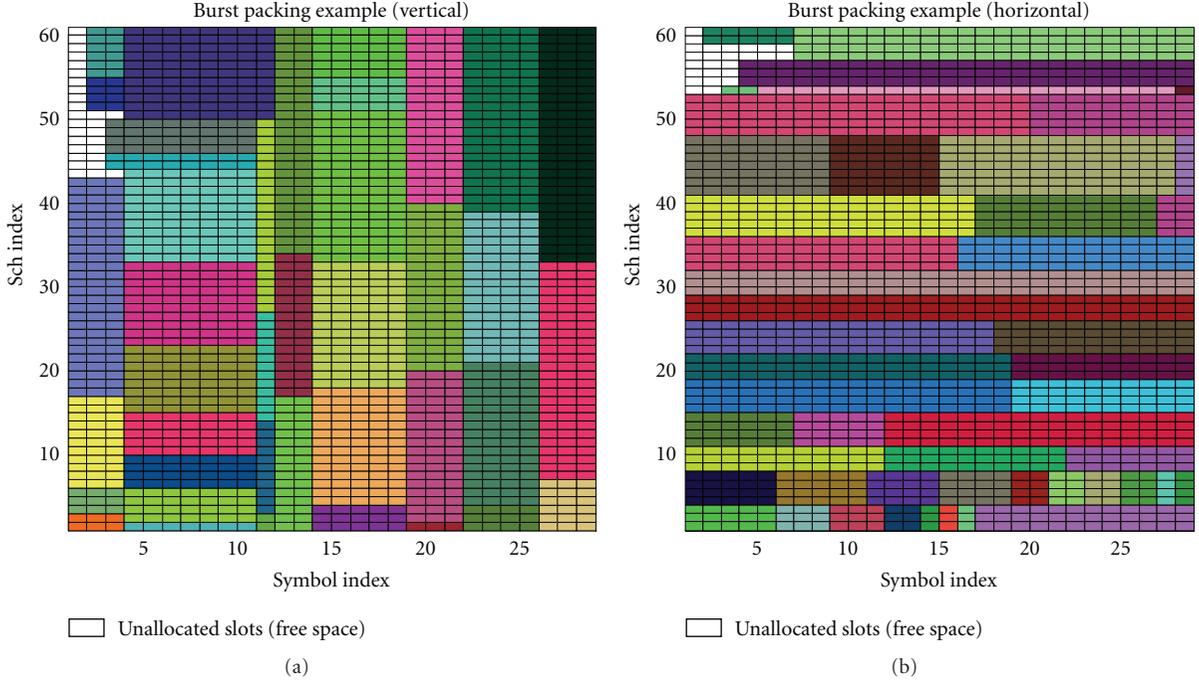


FIGURE 6: Examples from OBBP simulator.

maintain the regularity of the remaining free space as possible as we can), then update the OF matrix by setting the allocated burst to zero.

- (d) After every burst packing we remeasure the free space shape, generate the possible rectangles, and repeat steps from (b) to (c) until B'_{rem} becomes empty or the free space does not fit for any of the remaining bursts.

Figure 6 shows an example of the two modules of OBBP algorithm with the two possible scenarios, columnwise, and rows-wise. Notice that the colors of the bursts are generated randomly by the simulator, however the identical colors does not mean same burst, since that OBBP does not use burst fragmentation.

5. Performance Evaluation

In this section, we evaluate OBBP algorithm by means of simulations and taking the packing efficiency as the performance criteria. In the performance analysis we calculate the packing efficiency under different scenarios and compare it with the ideal algorithm which represents the upperbound of the packing efficiency. The bursts in ideal algorithm can be packed without any shape limitations and therefore no slot padding occurs. This is similar to the 1D packing algorithms [5].

The performance evaluation includes two packing scenarios, Vertical and Horizontal for various offered load and Burst Size Ratios (BSR). The parameters shown in Table 1 have been used in the simulations with frame duration of 5 ms and 20 MHz channel bandwidth.

The packing efficiency (η_{pk}) can be calculated using the following formula:

$$\eta_{pk} = \frac{S_{total} - S_{padded}}{S - OH}, \quad (4)$$

where, S_{total} is the total number of allocated slots (occupied by data bursts), S_{padded} is the total number of padded slots and OH is the overhead where

$$OH = \text{Preamble} + \text{FCH} + \text{DL_MAP} + \text{UP_MAP}. \quad (5)$$

The burst size ratio is the ratio of the number of big size bursts to the number of small size bursts, that is,

$$BSR = \frac{B'_{small}}{B'_{big}}, \quad (6)$$

where, $B'_{big} =$ for all $b_i > BS_{th}$, $b_i \in B'$, $B'_{small} =$ for all $b_i < BS_{th}$, $b_i \in B'$ and BS_{th} is an integer number represents the burst size threshold (in our simulations we used $BS_{th} = 20$ slots).

5.1. Packing Efficiency of OBBP Algorithm. Figure 7 demonstrates the behavior of OBBP algorithm under different offered loads. As shown in this figure, when the offered load is less than the frame capacity (i.e., less than 1%), the packing efficiency increases linearly with the offered load. This case is most likely in cellular mobile networks where the frames usually are sent with load less than the frame capacity which reduces the system throughput. So, we can improve the system throughput by dividing the DL-Subframe into variable size permutation zones in sectorized

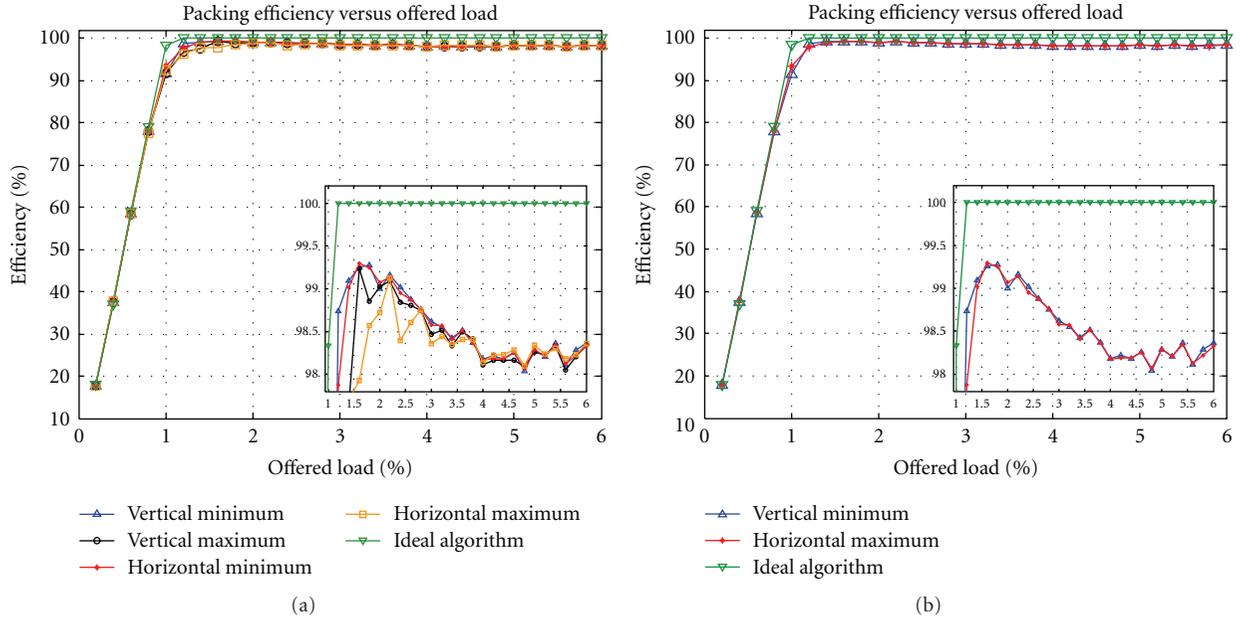


FIGURE 7: The Packing efficiency of vertical and horizontal scenarios.

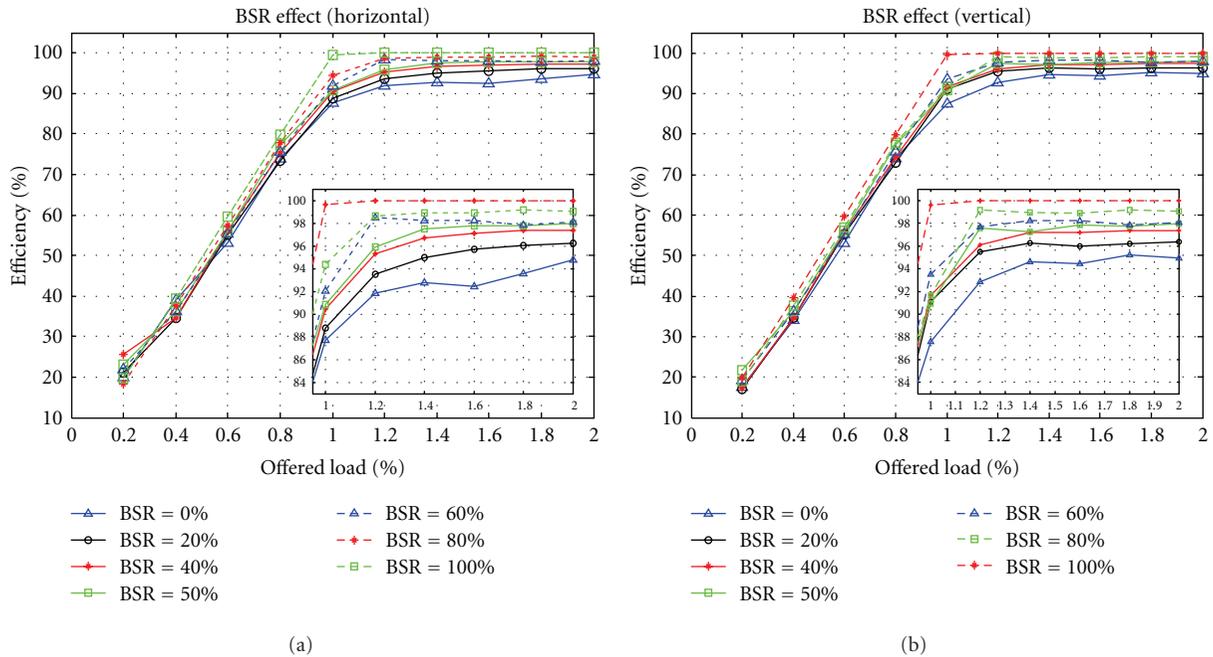


FIGURE 8: The effect of BSR for vertical and horizontal scenarios.

cells as we will discuss in the future work in Section 6. In Figure 7(a) we can see the effect of the burst set selection (discussed in Section 4.2.1 step (d)) either maximum or minimum number of bursts should be selected. This figure shows that the packing of subsets with a small number of large bursts first gives better packing efficiency than packing a big number of small bursts. According to our simulation results OBBP algorithm can achieve a packing efficiency up to 99.2% compared to the ideal algorithm for offered load greater than 1% and Burst Size Ratio (BSR) 50%.

5.2. *The Effect of Burst Size Ratio (BSR)*. Logically the efficiency of packing small rectangles is better than big rectangles and this is clearly presented by our simulation results. Figure 8 demonstrates the effect of BSR on the packing efficiency of OBBP algorithm. In this figure the packing efficiency increases as the BSR increase since we have more small bursts which can be packed efficiently. Both vertical and horizontal scenarios show the same performance with packing efficiency ranges from 89% to 100% for offered load 1% and from 95% to 100% for offered load 2%. The

TABLE 1: Simulation parameters.

Parameter	Value
Channel BW	20 MHz
Frame duration	5 ms
DL/UP ratio	28/19
Permutation scheme	PUSC
Number of permutation zones	1
Max burst size	95 slots
Total number of slots per DL-Subframe (S)	(60×28) 1680 slots

behavior of the algorithm for offered load below 1% is almost linearly increased as discussed in Section 5.1.

More work can be done to improve the performance of OBBP algorithm to meet the requirements of heterogeneous traffic with different QoS preferences.

6. Conclusion and Future Work

In this paper an efficient Orientation-Based Burst Packing (OBBP) algorithm for OFDMA DL-Subframe has been presented. The principle of common orientation factors (OF) gives OBBP algorithm flexibility and simplicity with high packing efficiency. According to our simulation results the algorithm can achieve a maximum of 99.2% in packing efficiency for full offered load (i.e., greater than 1%) and Burst Size Ratio (BSR) = 50%.

From our investigation to the Burst Packing problem in OFDMA networks through OBBP algorithm, we noted that the algorithm has very poor packing efficiency when the offered load is less than the frame capacity (full offered load). This feature can be employed to improve the packing efficiency and the system throughput in sectorized cellular networks.

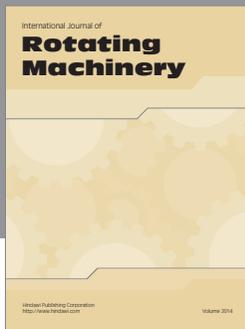
Acknowledgments

The authors would like to express their appreciation to the reviewers for their comments; this work was supported by the government of Malaysia, e-science fund under Grant no. 01-01-02-SF0471.

References

- [1] IEEE Standard for Local and Metropolitan Area Networks, "Part 16: Air Interface for Fixed Broadband Wireless Access Systems," IEEE 802.16-2004 Standard, October 2004.
- [2] IEEE Standard for Local and Metropolitan Area Networks, "Part 16: Air Interface for Fixed Broadband Wireless Access Systems—Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Band," IEEE 802.16e-2005 Standard, February 2006.
- [3] Y. L. Wu, W. Huang, S. C. Lau, C. K. Wong, and G. H. Young, "An effective quasi-human based heuristic for solving the rectangle packing problem," *European Journal of Operational Research*, vol. 141, no. 2, pp. 341–358, 2002.

- [4] R. Cohen and L. Katzir, "Computational analysis and efficient algorithms for micro and macro OFDMA scheduling," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 1184–1192, April 2008.
- [5] A. Zubow, D. C. Mur, X. P. Costa, and P. Favaro, "Greedy scheduling algorithm (GSA)—design and evaluation of an efficient and flexible WiMAX OFDMA scheduling solution," *Computer Networks*, vol. 54, no. 10, pp. 1584–1606, 2010.
- [6] J. Zhu, H. Kim, and H. H. Kwak, "A linear-complexity burst packing scheme for IEEE 802.16e OFDMA downlink frames," in *Proceedings of the 69th IEEE Vehicular Technology Conference (VTC '09)*, April 2009.
- [7] D. P. Malladi, F. Grilli, and S. D. Willenegger, "Method and apparatus for fast link setup in a wireless communication system," United States Patent Application 20090175236.
- [8] J. Kang and S. Park, "Algorithms for the variable sized bin packing problem," *European Journal of Operational Research*, vol. 147, no. 2, pp. 365–372, 2003.
- [9] T. Ohseki, M. Morita, and T. Inoue, "Burst construction and packet mapping scheme for OFDMA downlinks in IEEE 802.16 systems," in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM '07)*, pp. 4307–4311, November 2007.
- [10] P. Hosein, "Cross-Layer design for data burst construction in the downlink of IEEE 802.16 systems," in *Proceedings of the Annual IEEE Global Telecommunications Conference (GLOBECOM '08)*, pp. 4657–4661, December 2008.
- [11] C. So-In, R. Jain, and A. Al-Tamimi, "eOCSA: an algorithm for burst mapping with strict QoS requirements in IEEE 802.16e mobile WiMAX networks," in *Proceedings of the IEEE Wireless Communication and Networking Conference*, 2008.
- [12] J. Bae, A. Chindapol, J. Chui, and V. Marchenko, "Method for resource allocation with rectangular burst's," United States Patent Application 20080031190.
- [13] C. So-In, R. Jain, and A. K. Al Tamimi, "OCSA: an algorithm for burst mapping in IEEE 802.16e mobile WiMAX networks," in *Proceedings of the 15th Asia-Pacific Conference on Communications (APCC '09)*, pp. 52–58, October 2009.
- [14] M. C. Necker, M. Köhn, A. Reifert, J. Scharf, and J. Sommer, "Optimized frame packing for OFDMA systems," in *Proceedings of the 67th IEEE Vehicular Technology Conference (VTC '08)*, pp. 1483–1488, May 2008.
- [15] Y. Ben-Shimol, I. Kitroser, and Y. Dinitz, "Two-dimensional mapping for wireless OFDMA systems," *IEEE Transactions on Broadcasting*, vol. 52, no. 3, pp. 388–396, 2006.
- [16] C. Dessel and G. Lenoir, "WiMAX downlink OFDMA burst placement for optimized receiver duty-cycling," *IEEE Communication Society*, pp. 5149–5154, 2007.



Hindawi
Submit your manuscripts at
<http://www.hindawi.com>

