Received: 20 December 2017 | Revised: 3 November 2018 | Accepted: 13 March 2019

DOI: 10.1002/dac.4066

RESEARCH ARTICLE

WILEY

# QoS-aware LTE-A downlink scheduling algorithm: A case study on edge users

Osman Gokhan Uyan | Vehbi Cagri Gungor

Electric and Computer Engineering, Abdullah Gul University, Kayseri, Turkey

**Correspondence**
Vehbi Cagri Gungor, Electric and Computer Engineering, Abdullah Gul University, Sumer Campus, Kayseri, Turkey.
Email: cagri.gungor@agu.edu.tr

**Funding information**
BAGEP and AGU Foundation; AGU Foundation; BAGEP

**Summary**

4G/LTE-A (Long-Term Evolution—Advanced) is the state of the art wireless mobile broadband technology. It allows users to take advantage of high Internet speeds. It makes use of the OFDM technology to offer high speed and provides the system resources both in time and frequency domain. A scheduling algorithm running on the base station holds the allocation of these resources. In this paper, we investigate the performance of existing downlink scheduling algorithms in two ways. First, we look at the performance of the algorithms in terms of throughput and fairness metrics. Second, we suggest a new QoS-aware fairness criterion, which accepts that the system is fair if it can provide the users with the network traffic speeds that they demand and evaluate the performance of the algorithms according to this metric. We also propose a new QoS-aware downlink scheduling algorithm (QuAS) according to these two metrics, which increases the QoS-fairness and overall throughput of the edge users without causing a significant degradation in overall system throughput when compared with other schedulers in the literature.

**KEYWORDS**

4G, LTE-A, QoS aware fairness, resource allocation, scheduling

## 1 | INTRODUCTION

Since the introduction of the first generation Mobile Telecommunication Systems, mobile telecommunication technology has been developed rapidly. From the first analogue systems which were introduced in the early 1980s to the latest broadband technology we use today, highly increasing data transmission speeds have added many new features to the mobile networks and provided the users with new multimedia applications. These developments caused mobile data traffic to grow 4000-fold over the past 10 years and almost 400 million-fold over the past 15 years. It is also expected to grow another tenfold until the year 2020.[1]

Third Generation Partnership Project (3GPP), the global mobile communication standards developing organization, has been working on new technologies to meet this traffic demand and presented the standard (4G/LTE) with Release 8 in the year 2008. LTE simply consists of two subnetworks: Evolved-Universal Terrestrial Radio Access Network (E-UTRAN) and Evolved Packet Core. E-UTRAN is introduced with LTE, and it is the interface between the base station (eNodeB) and user equipment (UE). It employs Orthogonal Frequency Division Multiple Access (OFDMA) for downlink connections which can allow reaching high data speeds with low latencies. OFDMA is based on Orthogonal Frequency Division Multiplexing (OFDM). In OFDM, a large number of closely spaced orthogonal subcarrier signals are used to carry data on several parallel data streams or channels.[2]

It lends resources in both time and frequency domains. In time domain, a 10-ms radio resource unit is called a frame, and it consists of 10 subframes which are all 1 ms long. On the frequency domain side, there are multiple subcarriers each of which have 15 KHz bandwidth. Half of a subframe (0.5 ms) from time domain and 12 subcarriers from frequency domain form a Resource Block (RB). These Resource Blocks are allocated to users every 1 ms which we call a Transmission Time Interval (TTI). The process of this allocation of resources is named scheduling. Scheduling is executed on MAC layer using an appropriate algorithm.

3GPP Organization has not defined a standard algorithm for the scheduling mechanism in LTE specifications, which means that a service provider is free to choose a suitable one among a variety of scheduling algorithms. This freedom has been an inspiration for both scientists, mobile network corporations, and mobile operators to bring about several different scheduling algorithms. Since scheduling has a serious effect on the operation of the system, success of the scheduling algorithm is an important issue for system management.

In this paper, we propose a new QoS-aware downlink scheduling algorithm, QuAS, to enhance the QoS experience of mobile network users. Unlike the existing studies, we especially concentrate on the performance of the edge users as they gain meager throughput and delay in the cell. The important challenge that the edge users face is the poor channel conditions they experience because of the distance and the obstacles between the UEs and the eNodeB. The primary goal of the proposed algorithm is to enhance the QoS experience of the edge users while avoiding a significant loss in overall system throughput and QoS. For this purpose, the scheduler uses packet size and delay information of the users to define the allocation of the RBs. The simulations for evaluating performance of the algorithms are held under various scenarios, such as static and mobile users, Coordinated Multi Point (CoMP) structure, with various parameters, like carrier frequency, different number of users and eNodeBs. Here, CoMP represents a combination of distinct methods that endorse coordination of transmission and reception dynamically over different eNodeBs. Its aim is to upgrade overall throughput of the users, particularly at the cell edges. Moreover, when a user is connected to multiple eNodeBs, its data can be transmitted through the least busy eNodeB, or through the best quality channel among the connections which is expected to decrease delivery delays and increase capacity.

The remainder of this paper is organized as follows. Section 2 discusses the related work in the literature. The QuAS algorithm is explained in detail in Section 3. Performance evaluations are presented in Section IV. Lastly, the paper is concluded in Section 5.

## 2 | RELATED WORK

Scheduling is a very popular subject in the area of LTE, and it has attracted many researchers and corporations to put on some effort for designing new algorithms. This is the reason that there are several studies about scheduling algorithms in the literature. The motivation of each algorithm changes commonly around system throughput and fairness. The important issue about these two metrics is that there is a trade-off between them.

In general, the existing algorithms in the literature attend to improve one of these classical metrics while trying to keep the degradation in the other metric as limited as possible. The classification of well-known existing scheduling algorithms, such as best CQI, proportional fair, round-robin (RR), and their performance metrics are shown in Figure 1.

One of the well-known algorithms is RR, which is very basic and easy to implement. The algorithm is channel-blind, which means that it simply lends RBs one by one to the users consecutively until all of the users are assigned a resource

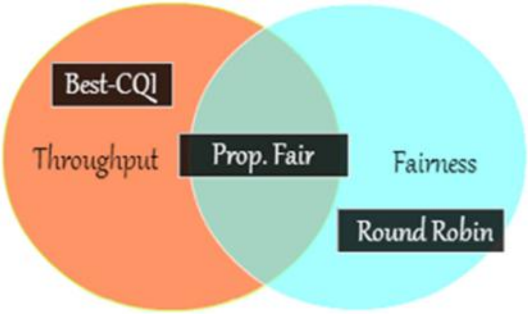

**FIGURE 1** Objectives of scheduling algorithms

without taking their channel conditions into consideration. After this process, the algorithm starts over from the beginning of the user equipment (UE) list and repeats this sequence.

Figure 2 depicts a sample resource allocation using RR. The x-coordinate shows time, and y-coordinate shows channel quality. Each vertical box shows a Transmission Time Interval (TTI). The thicker lines depict the UEs which are allocated a RB in each TTI. Since RR is channel-blind, it sometimes allocates to UEs who are on fading channels and this causes a decrease in the system throughput as a result of low transmission rates under a bad channel condition. Although it seems to be a fair algorithm, the fairness it provides is in terms of the number of RBs assigned to each user rather than throughput manner.

Another widely used algorithm is best-CQI. In every TTI, best-CQI algorithm strives to allocate RBs to the UEs which have the finest channel conditions for each RB. It is a channel-aware algorithm, and it guarantees the largest throughput for a cell because it always allocates a RB to the UE with the best channel quality for that RB. However, based on the fairness-throughput trade-off, this results in a deficient fairness index for the network.

Best-CQI grants RBs only to the UEs with best channel conditions. Oppositely, UEs with poor channel conditions, especially the ones who are close to the cell edges, may never be able to use the network with this scheme. The metric of the best-CQI algorithm is rather simple:

$$k = \arg\max_j \left(R_j\right), \tag{1}$$

where $R_j$ is the momentary transmission rate for $j$th user and it is calculated from the Channel Quality Index (CQI) values sent by each UE to the eNodeB.

Figure 3 illustrates a sample resource allocation with best-CQI algorithm. The x-coordinate shows time, and y-coordinate shows channel quality. Each vertical box shows a TTI. It always allots UEs who have the best channel conditions at each TTI and this results in a high overall throughput. Adversely, this type of allocation produces an unfair system, peculiarly for the UEs close to the cell edges that are challenging poor channel conditions.

Yildiz and Sokullu have proposed an algorithm, MAS, which is a hybrid of RR and best-CQI algorithms.[3] They first assign to users with best channel conditions, then they apply an RR among other users to increase the fairness index of the scheduling.

The most acclaimed algorithm in the literature is the Proportional Fair (PF) algorithm. It was first designed for CDMA systems to be used on time-domain scheduling only. Kim et al[4] expanded this algorithm so that it can be used with OFDM in both frequency and time domain. Notwithstanding, this algorithm was computationally complex, and it

**FIGURE 2** Round-robin scheduling example

**FIGURE 3** Best-CQI scheduling example

was hard to use in a real-time system. Based on this algorithm, Sun et al [5] offered a low-complexity PF algorithm. This new algorithm reduced the computational complexity while showing similar performance with the former version. The goal of the PF algorithm is to achieve high fairness values by using channel quality indices of the users. Being a channel-aware algorithm, it calculates instantaneous achievable throughput for each user from the CQI values and divides it by the average achieved throughput of the user in a past window to obtain the decisive metric to allocate resources to users. The PF algorithm redesigned by Sun et al is given below.

```
Proportional Fair Algorithm [5].
Input: CQI feedback of users
for each k
    compute: $R_k(n)$ from CQIs and $T_k(n)$
    evaluate: $k^*(n) = argmax\frac{R_k(n)}{T_k(n)}$
Output: Resource allocation matrix (N_RB x N_UE).
```

One more famous algorithm is called Blind Equal Throughout (BET) algorithm which is proposed by Toseef et al.[6] This algorithm uses a memory to store the average throughput achieved by each user in the past window, and it uses this information as a metric for calculating the weight of each user for allocating resources. BET maintains fairness among all users without taking their channel conditions into consideration; thence, it is called "blind." Weight of a user for next TTI is evaluated as the inverse of its average throughput up to then $M_i = 1/R_i(t)$, where $R_i(t)$ is the prior average throughput of the $i$th user.

Sudheep and Rebekka[7] introduced another algorithm named Proportional Equal Throughput (PET), which is a hybrid of PF and BET algorithms. PET allocates the RBs in a TTI with a fraction to the users; instead of giving all RBs in a bandwidth to one user, they divide the RBs into proportions so that they can be given to other users whose weights follow the user with the maximum weight. Their simulation results show that the PET algorithm gives good performances about fairness compared to BET without causing a considerable decrease in system throughput.

AlQahtani and Alhassany[8] came up with a novel algorithm. It behaves like classical RR as far as all users share same number of RBs. Subsequently, it starts acting like best-CQI algorithm and allocates the remaining RBs to the users with topmost CQI values. It performs better than best-CQI in terms of fairness but causes a decrease in the overall system throughput oppositely as expected.

Liu and Lee[9] propound Earliest Deadline First (EDF), a QoS-aware algorithm, aiming at avoiding headline expiration. In Internet services, guaranteed delay needs that a packet must be delivered before an assured time limit to fend dropping off packets. EDF schedules the packets with the impending deadlines. Nonetheless, besides being QoS-aware, EDF is channel-unaware, that is, it does not take CQI feedback of the users into account. As a consequence of this feature, it is not very suitable to use with mobile networks because channel characteristics may change rapidly in a wireless broadband connection and a packet still might not be delivered on a bad quality channel on time. To cope with this issue, Bin et al[10] suggested a combined version of EDF and PF, which is more convenient to be used in mobile networks. The proposed M-EDF-PF algorithm is both channel aware and QoS aware, as it takes fairness characteristic of PF and limited delay guaranteed characteristic of EDF. It is suitable to be used with real-time services like video broadcasting or VoIP.

Trabelsi and Selem[11] proposed a Decoupled-Level QoS aware scheduling algorithm, which tries to guarantee QoS for different traffic types by keeping reasonable values of throughput and fairness. In the first step, the algorithm checks if a UE has a packet in buffer, and if so, it separates the users into two groups: Guaranteed Bit Rate (GBR) and non-GBR. After the selection, the scheduler serves the GBR list using best-CQI approach, then moves to non-GBR list, and serves the users according to highest priority packet.

Akyildiz and Akkuzu[10] have come up with a QoS algorithm that works in a similar manner with M-EDF-PF. This scheduler also divides the UEs into two groups according to their traffic type. If a user has a UDP traffic, it is placed into primary list, and if it has a TCP traffic, it is placed into the secondary list. After the separation of the users into two lists, the scheduler works as the best-CQI algorithm and gives resources first to the primary list and then the secondary list according to this approach.[12]

Zaki and Weerawardane[13] proposed another QoS-aware algorithm. Their algorithm, LTE MAC, categorizes the incoming packets into five different QoS classes. The top two QoS classes are accepted as GBR, and the other three classes are accepted as non-GBR bearers. The algorithm applies strict scheduling with giving priority to the GBR bearers and then starts with scheduling of non-GBR bearers.

Ferdosian and Othman[14] has proposed a new scheme which again divides the mobile traffic into GBR and non-GBR groups. There are four services grouped as GBR which are conversational voice, conversational video (live-streaming), online gaming, and nonconversational video (buffered-stream). On the other hand, there are five services grouped as non-GBR which are IMS signaling, TCP-based video, voice-video (live-streaming), and voice-video (buffered-streaming). They design a mathematical utility function to evaluate the ranks of the bearers about their desired performance targets. After classifying the bearers, they use the same manner with the Proportional Fair algorithm to assign the RBs to the UEs. Another algorithm, FQB, proposed by Ferdosian and Othman also uses the same GBR–non-GBR grouping to increase fairness index of the users when the users' demands are higher than the available system capacity.[15]

Al-Shuraifi and Al-Zayadi[16] propose a scheduling method, which is again based on the best-CQI algorithm. The scheduler first collects data about network and channel conditions of the users. Then it separates the users into two groups according to their Signal to Noise Ratio (SNR) values. And then the algorithm uses the best-CQI method to allocate resources to both groups according to the priorities of the groups.

All the above-mentioned studies provide valuable efforts for LTE downlink scheduling both about fairness and throughput as listed in Table 1. However, none of them presents a detailed evaluation about satisfying the Quality of Service requirements of the system users, and more specifically, edge users. In our previous study, this gap is partially filled considering basic user metrics comprising average cell throughput, edge throughput, Lain's fairness index, and QoS fairness index introduced in Gungor and Uyan.[17] The objective of this paper is to extend our previous study further by simulating the performance of the QuAS algorithm using more advanced user metrics, such as cell peak throughput, mobility, carrier frequency, and antenna configuration (MIMO). With these further simulations, the efficiency of QuAS algorithm is analytically quantified in more details. Furthermore, for mobility, carrier frequency, and antenna configurations, parameters mostly used in real life are chosen to be able to demonstrate a prospect of QuAS algorithm in familiar conditions. These simulation parameters can be seen in Table 3.

Tian et al[18] proposes two new algorithms for a constrained QoS in wireless interference limited networks. One of the algorithms is designed for homogeneous user traffic scenario while the other one is designed for heterogeneous user traffic. The algorithms are designed with respect to the optimization of medium access probability (MAP) while taking the delay sensitivity of the user applications into consideration.

**TABLE 1** Comparison of the related work

| Related Work | Algorithms | Number of Users | Mobility | Antenna Config. | CF | Performance Metric |
|---|---|---|---|---|---|---|
| Sun[5] | PS, PF, SC-PF | 20 | N/A | 1 × 1 | SF | CT, MT |
| Toseef[6] | BET, PF, Adp. Fair, RR | 10 | Static | 1 × 1 | SF | MT |
| Sudheep[7] | PET, BET, PF, BCQI, RR | 10 | Static | 2 × 2 | SF | MT |
| AlQahtani[8] | PS, RR, BCQI | 10-50 | N/A | 1 × 1 | SF | CT, fairness |
| Bin[10] | EXP-RULE, EXP/PF, LOG-RULE, M-LWDF, M-EDF-PF | 10-80 | N/A | 1 × 1 | SF | CT, fairness, PLR |
| Trabelsi[11] | LWDF, RR, EDF, PF, M-LWDF, FIFO | 10-250 | N/A | 1 × 1 | SF | CT, Avg. delay, PLR |
| Akyildiz[12] | BCQI | 80-150 | 5 km/h | 1 × 1 | SF | CT, Avg. delay |
| Zaki[13] | LTE MAC | 5, 20, 40 | N/A | 1 × 1 | SF | CT, Avg. delay, response time |
| Soni[19] | Optimal, VToD, sub-optimal (proposed in[18]) | 10-100 | N/A | 1 × 1 | SF | CT, fairness, complexity |
| Wu[20] | EXP/PF, MLWDF, ZBQoS, RLBS | 5-60 | 3 km/h | 1 × 1 | SF | Avg. TP, delay, PLR |
| Jiang[17] | M-LWDF, EXP/PF, | 5-25 | N/A | 1 × 1 | SF | PLR |
| Wang[21] | BCQI, M-LWDF, QFS | 10-60 | N/A | 1 × 1 | SF | Avg. TP, delay, PLR |
| QuAS algorithm | PS, PF, BCQI, RR, CoMP RR, QuAS | 20, 40, 60, 80, 100 | 5, 50, 100 km/h | 1 × 1, 2 × 2, 4 × 4 | MF | ET, PT, CT, fairness, QoS fairness |

Zhong et al[19] offer a new approach for analyzing the delay in heterogeneous cellular networks. They propose the notion of delay outage to evaluate the performance of diverse algorithms such as random scheduling, FIFO, and RR. The work states that RR runs better for heavy loaded networks while FIFO performs better for light network traffic.

Tian et al[20] propose a cross-layer scheme to maximize the average received video quality by considering the network transmission strategy. They formulate the maximization of video transmission quality as a cross-layer optimization problem, and they suggest a distributed algorithm based on game theory to solve the optimization problem. They compare the proposed solution with optimal-beta scheme and optimal-rate scheme in terms of peak signal-to-noise ratio (PSNR) and show that the proposed solution improves the system performance.

Soni and Tyagi[21] proposed an algorithm, which uses the same classification method with the algorithm defined in Ferdosian et al.[14] It divides the users into two group, GBR and non-GBR, according to their traffic information. The algorithm tries to maximize the throughput of the non-GBR users who increase the cell spectral efficiency. For allocation, the algorithm uses the metric of the Proportional Fair algorithm and multiplies it with the CQI index parameter to define the priority of the users among them. As with previously defined algorithms, this method also allocates the GBR users first to fulfill their delay constraints. After that, it starts allocation of non-GBR users and provides an opportunistic scheduling to increase the fairness of the system.

Wu and Han[22] proposed a Rate-Level–Based scheduling algorithm with the aim of supporting heterogeneous traffic in LTE downlink. The scheduler tries to minimize the packet loss ratio of the real-time traffic while guaranteeing QoS requirements. The algorithm calculates the priority of the users with pending transmissions according to their packet delay budget and Head of Line (HOL) packet delay along with the average spectrum efficiency of each user. After calculation of the priority of the users, the scheduler uses an RR-type process to schedule the users, where it allocates the user with highest priority first and the user with lower priority the last.

Jiang and Zhang[23] propose an algorithm to enhance the capacity of the network. It tries to allocate more resources to the users with poor channel conditions while supporting QoS requirements of the users with good channel conditions. For the users with good channel conditions, the algorithm allocates only the RBs with the instant throughput rate close to the peak rate to them, restricting the number of allocated RBs. This allows the algorithm to preserve more RBs to the users with bad channel conditions.

Wang and Huang[24] have proposed another classification-based algorithm. However, instead of having two groups of users, they divide the users into three groups, which are GBR, non-GBR, and Urgent. Urgent queue is given with the highest priority. If the RBs are allocated to all of the UEs in the Urgent queue, then scheduler starts allocation of the second priority group, which is the GBR users. After the allocation process of Urgent and GBR users, non-GBR users are allocated if there is still empty RBs awaiting to be allocated in the system. Table 1 shows the comparison of algorithms described in Section 2.

## 3 | QuAS ALGORITHM

In mobile networks, users are spread in the covering area of a base station. The quality of communication channel of a user depends on the communication distance and obstacles between the user and the base station. This affects the signal-to-noise ratio (SNR), bit error rate (BER), transmission delay, and achieved throughput. As the communication distance increases, that is, the user is closer to the cell edge, SNR and throughput tend to decrease while BER and delay increase.

The Proportional Fair (PF) algorithm,[5] which disclosed in the previous section, allocates network resources to the users according to the following metric:

$$k^{*}(n) = argmax\frac{R_k(n)}{T_k(n)}, \tag{2}$$

where $R_k(n)$ is the current achievable throughput for the $k$th user on $n$th resource block and $T_k(n)$ is the average throughput of the $k$th user in a predefined past window. The PF algorithm provides high performance in terms of both fairness and throughput. However, it lacks of a mechanism to deal with the quality of service (QoS) requirements of the users to maintain a guaranteed rate of data transmission for different user applications.

In this study, we propose a new QoS-aware downlink-scheduling algorithm, QuAS, to enhance the QoS experience of mobile network users based on the metric of PF. Unlike the existing studies, we especially concentrate on the performance of the edge users as they gain meager throughput and higher delays in the cell. The primary goal of the proposed algorithm is to enhance the QoS experience of the users, especially users closer to the cell edges, while avoiding a significant loss in overall system throughput and QoS. For this purpose, the scheduler uses packet size and delay information of the users to define the allocation of the RBs.

In first place, we introduce a new fairness metric named QoS fairness. This metric takes delay needs of each user's packets into consideration. The definition of the metric is as follows: If a packet is delivered in time, the fairness index of the user is incremented, else it is increased by the amount of transmitted data divided by the packet size.

$$f_{k,i} = \begin{cases} 1 & \text{if } B_r = 0 \\ (B_t/P) & \text{if } B_r > 0 \end{cases}, \tag{3}$$

where $f_{k,i}$ is the fairness value of $k$th user in $i$th TTI, $B_r$ is the amount of remaining data bits at the end of requested delay, $B_t$ is the amount of successfully transmitted data bits, and $P$ is the packet size. After estimating $f_{k,i}$ for each user, its average is calculated to find the eventual fairness $F$ of the system, where N_UE is the number of users in a cell.

$$F = \frac{\sum_{k=1}^{N\_UE} f_{k,i}}{N\_UE}. \tag{4}$$

The aim of QuAS is both maintaining good QoS fairness results when compared with other schedulers and increasing average throughput of edge users without causing a significant decrease in overall cell throughput. To attain this, QuAS algorithm uses the requested delay for each packet, instantaneous throughput rate of a user for each RB, packet size, and requested delivery time of the user in each Transmission Time Interval (TTI):

$$D_k(n) = \frac{P}{R_k(n)}, \tag{5}$$

where the time needed to transmit a packet is $D_k(n)$, and it is calculated by dividing packet size $P$ of a user by current achievable throughput $R_k(n)$ of that user. If $D_k(n)$ of a user's packet is smaller than requested delay $Q_k$, it is better to increase the user's chance of getting resources because the packet has a chance to be delivered in time. Here, we use the metric (2) of the PF algorithm, but we temporarily manipulate the CQI feedback input of the user by adding or subtracting it with a coefficient $c$, where $c$ can be modified during scheduling process to reach better fairness or throughput values. By increasing CQI of a user temporarily, the instant achievable throughput of the user is calculated to be higher, the metric of the user increases, and its chance to get a resource increases accordingly.

```
QuAS Algorithm.
Input: CQI feedback of the users, $CQI_{k,n}$, Requested Delay, $Q_{k,n}$, Packet Size, $P_{k,n}$.
Estimate: Average throughput $T_k(n)$ and Necessary delivery time $D_k(n)$ for each user.
for each user $k$
    if $D_k(n)$ smaller than $Q_k$
        $TCQI_{k,n}$ is equal to $CQI_{k,n}+c$
    else
        $TCQI_{k,n}$ is equal to $CQI_{k,n} - c$
    Estimate: Instant achievable throughput $R_k(n)$ using $TCQI_{k,n}$
obtain $(k^*, n^*) = argmax\frac{R_k(n)}{T_k(n)}$
Return: Resource allocation matrix (N_RB x N_UE)
```

CQI is used to obtain modulation and spectral efficiency of a user for each channel, and instantaneous throughput for each user is calculated using spectral efficiency. CQI is an indicator based on the SNR value of the user, and it is sent to eNodeB at the end of each TTI to inform it about the channel quality.

In the QuAS algorithm, CQI is manipulated to increase or decrease the calculated instant throughput of a user for each resource block, and this manipulation increases or decreases the chance of a user being allocated a resource block according to the metric (2). Table 2 shows the MCS index and spectral efficiency values corresponding to CQI values for LTE-A network.

**TABLE 2** CQI-efficiency table

| CQI | Modulation (LTE) | Modulation (LTE-A) | Code Rate (× 1024) | Spectral Eff. (LTE) | Spectral Eff. (LTE-A) |
|---|---|---|---|---|---|
| 0 | Out of range | | | | |
| 1 | QPSK | QPSK | 78 | 0.1523 | 0.1523 |
| 2 | QPSK | QPSK | 120 | 0.2344 | 0.3770 |
| 3 | QPSK | QPSK | 193 | 0.3770 | 0.8770 |
| 4 | QPSK | 16QAM | 308 | 0.6016 | 1.4766 |
| 5 | QPSK | 16QAM | 449 | 0.8770 | 1.9141 |
| 6 | QPSK | 16QAM | 602 | 1.1758 | 2.4063 |
| 7 | 16QAM | 64QAM | 378 | 1.4766 | 2.7305 |
| 8 | 16QAM | 64QAM | 490 | 1.9141 | 3.3223 |
| 9 | 16QAM | 64QAM | 616 | 2.4063 | 3.9023 |
| 10 | 64QAM | 64QAM | 466 | 2.7305 | 4.5234 |
| 11 | 64QAM | 64QAM | 567 | 3.3223 | 5.1152 |
| 12 | 64QAM | 256QAM | 666 | 3.9023 | 5.5547 |
| 13 | 64QAM | 256QAM | 772 | 4.5234 | 6.2266 |
| 14 | 64QAM | 256QAM | 873 | 5.1152 | 6.9141 |
| 15 | 64QAM | 256QAM | 948 | 5.5547 | 7.4063 |

## 4 | PERFORMANCE EVALUATIONS

QuAS algorithm is designed to provide sufficient edge throughput and maintain good QoS fairness. The simulations for evaluating performance of the algorithms are held under various scenarios with various parameters like mobility, carrier frequency, antenna configuration, number of cell users, and eNodeBs. The scenarios are designed to show performances of some well-known scheduling algorithms as well as QuAS algorithm. Simulation parameters are given in Table 3. Throughout the simulations, The Vienna LTE System Level Simulator[25] has been used.

### 4.1 | Edge throughput

The average edge throughput results are shown in Figure 4 according to different number of users in a cell. As expected, average edge throughput decreases as the number of users increases because the resources become scarce. Meanwhile, edge throughput is always 0 for best-CQI algorithm as it allocates resources to users with best channel quality, and edge users do not get service as they have poor channels because of fading channels problem in mobile networks.

As shown in Figure 4, the QuAS algorithm performs best, and it is 9.2% better than the second best algorithm (PF) in terms of edge user throughput. Moreover, these two algorithms outperform RR, best-CQI, and CoMP with RR algorithms especially if the number of users are smaller in a cell.

**TABLE 3** Simulation parameters

| Number of eNodeBs | 3 |
|---|---|
| Number of users per eNodeB | 20-100 |
| Simulation duration | 50TTI |
| Bandwidth | 20 MHz |
| Carrier frequency | 800, 1800, 2100 MHz |
| UE speeds | 5, 50, 100 km/h |
| Performance metrics | Edge, peak, cell Avg. TPs, fairness, QoS fairness |
| Algorithms | PF, RR, B-CQI, CoMP with RR, proposed Sch. |

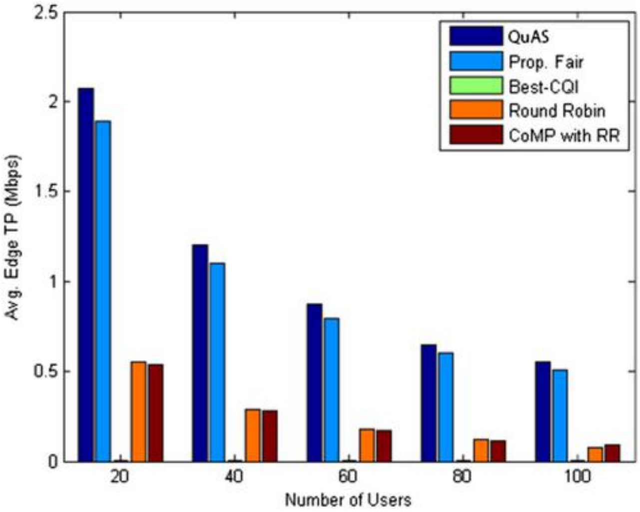

**FIGURE 4** Average Edge Throughput

## 4.2 | Peak throughput

Figure 5 shows the approximate locations of the users in the cell area. Peak throughput is the value calculated for those users who are closest to the cell centers and it mainly affects the overall cell throughput.

The users who are closer to the cell center suffer least from fading channels, and they have the best channel quality when compared with other users. This leads to a better communication between central users and the eNodeB. Figure 6 shows the simulation results for central users scheduled by diverse algorithms.

Best-CQI algorithm performs well as it allocates the resources to the users with the best channel conditions. However, this causes very poor fairness results according to Jain's fairness metric[26] and QoS fairness metric suggested in this paper (3), (4), (5). QuAS algorithm provides very similar results when compared with PF algorithm about peak throughput and it performs better compared with PF when the number of users in a cell increases. RR and CoMP with RR provide similar results to each other which are outperformed by best-CQI algorithm. However, the coordinated structure increases average peak throughput about 10% compared with RR.

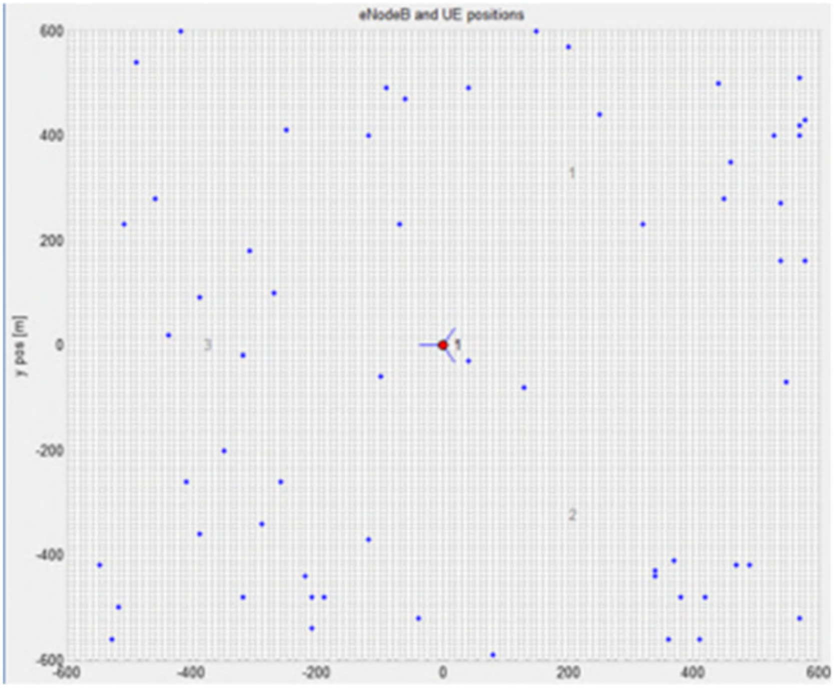

**FIGURE 5** Positions of the users and eNodeBs on the simulation

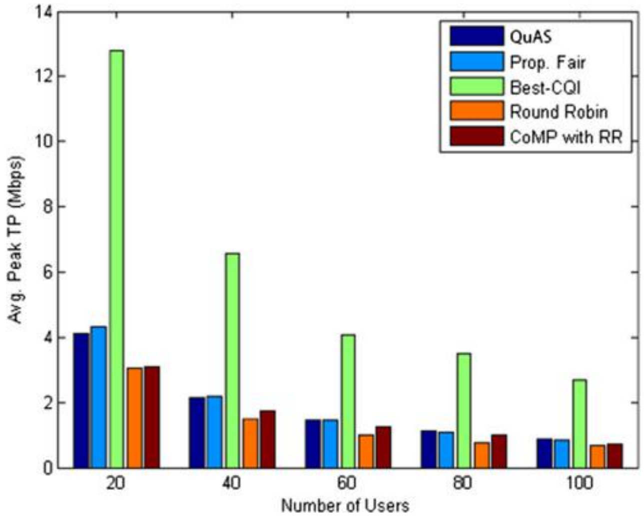

**FIGURE 6** Avg. peak TP performance of schedulers

In CoMP, a certain number of transmitters arranges coordinated transmission in the downlink, and some of the receivers provide coordinated reception in the uplink. CoMP is a combination of distinct methods that endorse coordination of transmission and reception dynamically over different eNodeBs. Its aim is to upgrade overall throughput for users, particularly at the cell edges. Moreover, when a user is connected to multiple eNodeBs, its data can be transmitted through the least busy eNodeB, or through the best quality channel among the connections which is expected to decrease delivery delays and increase capacity. Figure 7 depicts a sample CoMP environment.

## 4.3 | Average cell throughput

Average cell throughput is one of the most important criteria in resource scheduling of LTE systems. High throughput means that users are served better, having better experience from the network. However, there is another criterion, fairness, which is in a trade-off with throughput. The network has to serve all of its users without ignoring service requests of any user. A user with poor channel qualities might also request higher data rates. Therefore, the scheduling process has to take both of these criteria into consideration to meet user application requirements.

Figure 8 depicts average cell throughput achieved by the simulated schedulers. Best-CQI algorithm reaches high throughput rates on the cell average; however, its fairness results are not as good as its throughput values. QuAS algorithm provides similar results with PF in terms of average cell throughput and it is better than RR, and CoMP with RR algorithms according to the simulation results.

## 4.4 | Jain's fairness metric

Jain's fairness metric estimates how impartial an algorithm is, about giving equal throughput to all the users being served. It estimates the fairness values for n users and $x_i$ is the throughput value gained on the *i*th channel.

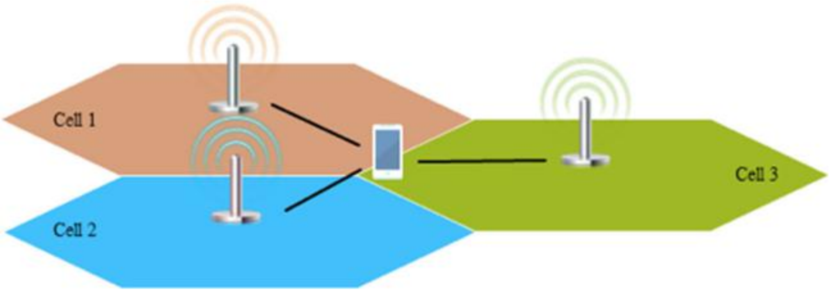

**FIGURE 7** Coordinated multi point (CoMP)

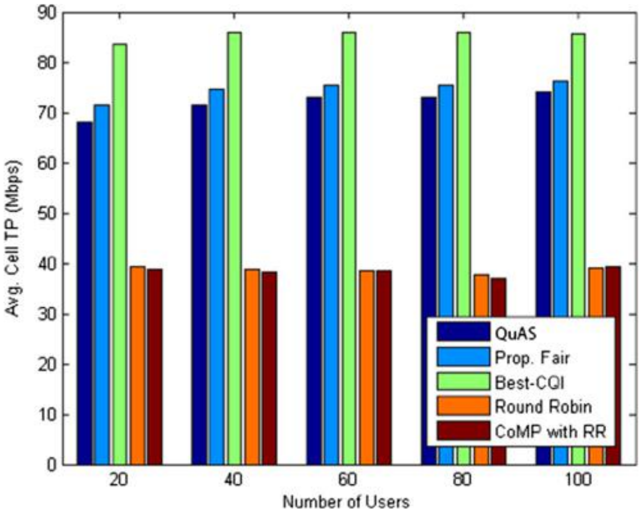

**FIGURE 8** Avg. cell TP performance of schedulers

$$J(x_1, x_2, \ldots, x_n) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \cdot \sum_{i=1}^{n} {x_i}^2}. \tag{6}$$

Although best-CQI algorithm provides good results about throughput, it shows very poor performance about Jain's fairness. QuAS algorithm shows the best performance among all algorithms. PF, RR, and CoMP algorithms also provide reasonable results about fairness, which can be seen from Figure 9.

The reason of QuAS algorithm performing the best in terms of Jain's fairness index is that, while trying to fulfill QoS requirements of all the users inside a cell, it allocates more resources to edge users and shares the resources more equally among the users.

## 4.5 | QoS fairness metric

QoS fairness is a novelty introduced in this paper to examine performances of scheduling algorithms about users' service requests and network experiences. QoS metric is defined in Part III, and it is calculated as given in (3), (4), and (5). QuAS algorithm uses delay needs of users' packets which is very important for services like video streaming or online gaming.

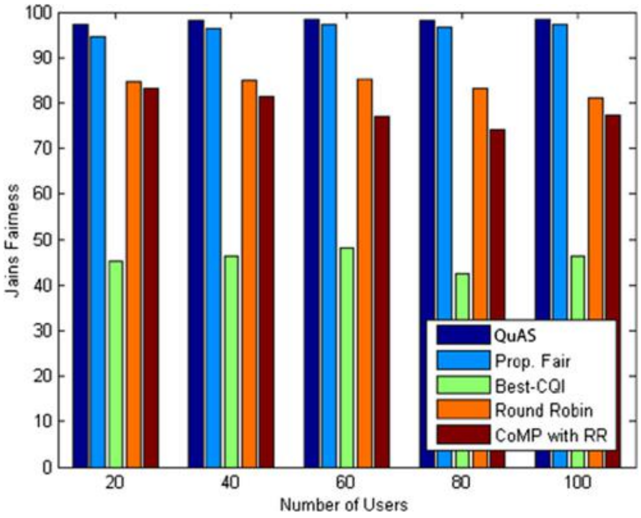

**FIGURE 9** Jain's fairness index of schedulers

As it can be seen from Figure 10, QoS fairness index decreases as the number of users increase. QuAS algorithm provides the highest results and outperforms best-CQI, RR, and CoMP algorithms. It also maintains about 5.5% higher results than PF algorithm, and it helps users to get a continuous experience from the system.

## 4.6 | Mobility

LTE network is developed to perform well under a range of diverse user speeds from about 5 to 120 km/h. In the simulations, three level of user speeds are chosen to test out the performance of scheduling algorithms about mobility: 5 km/h as average human walking speed, 50 km/h as urban driving speed, and 100 km/h as highway driving speed.

It can be observed from Figure 11 that the average peak throughput supplied by each scheduler decreases as the speed of the users increase. This is an expected result of mobility, because the more speed of a user increases, the harder is it to maintain good channel quality between the user and the eNodeB.

The performance of best-CQI decreases dramatically according to increasing user speed. Other algorithms including QuAS are more robust against mobility and they do not cause a significant throughput loss which becomes about 2% smaller as the user speed increases. QuAS algorithm and PF produce similar results; QuAS algorithm is 3% better than

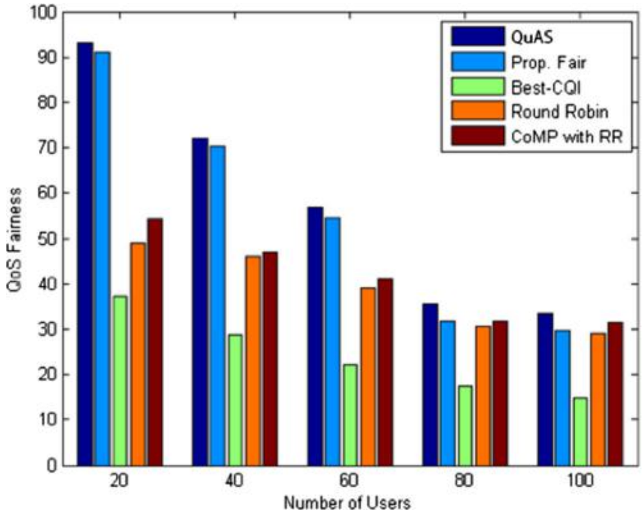

**FIGURE 10** QoS fairness index of schedulers

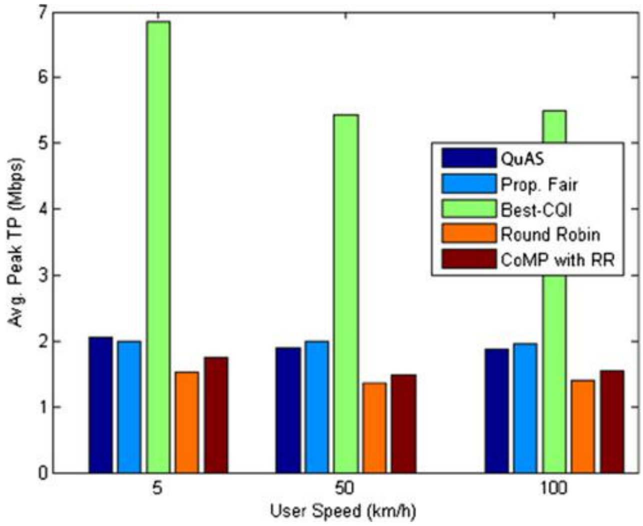

**FIGURE 11** Avg. peak TP with mobility

PF when user speed is 5 km/h, while PF provides 2% better results on the average. CoMP with RR gives about 10% better results than RR but they provide linear throughput results.

The average edge throughput results under different user speeds are shown in Figure 12. The average edge throughput supplied by each scheduler tends to increase as the speed of the users increase. This is also a natural result of mobility, because as the speed of a user increases, the harder is it to maintain good channel quality between the user and the eNodeB. This means there are going to be more users acting like edge users if the speed increases. It can be seen from the Figure 12 that QuAS algorithm and PF algorithm again outperform other three algorithms when edge throughput is considered. QuAS algorithm provides the best results in terms of edge throughput, which is about 9.2% higher than PF algorithm.

Since best-CQI algorithm provides highest peak throughput rates, it also produces the best cell throughputs. QuAS algorithm performs acceptable results when compared to PF algorithm. It causes a decrease about 3.5% in the overall cell throughput, but instead it allows a significant increase in edge throughput.

Average cell throughput results are another important measure to show the effects of mobility of the users. Figure 13 depicts average cell throughput results for three speed levels along with performance of scheduling algorithms.

Observing the Jain's fairness results along with mobility shows that QuAS algorithm performs the best, at a ratio of 1.8% better than PF and outperforming other three algorithms. The fairness results can be seen in Figure 14.

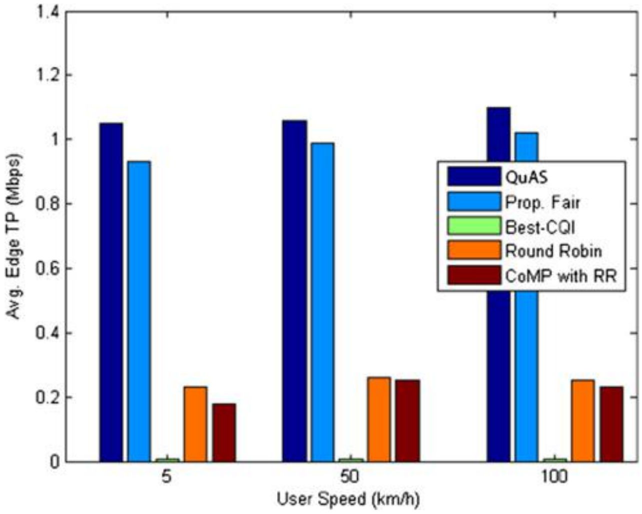

**FIGURE 12** Avg. edge TP with mobility

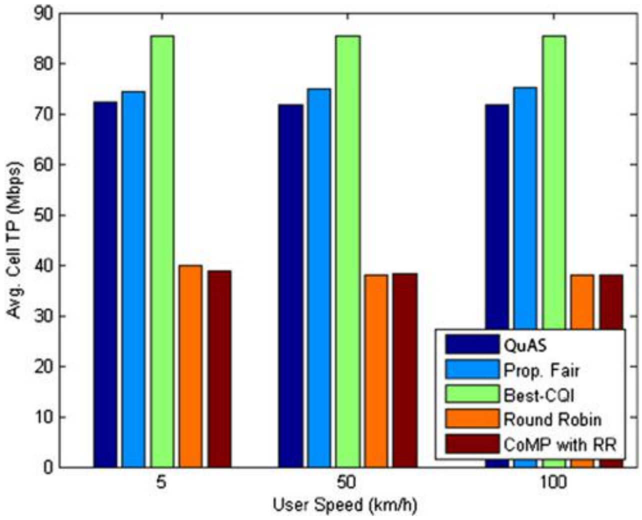

**FIGURE 13** Avg. cell TP with mobility

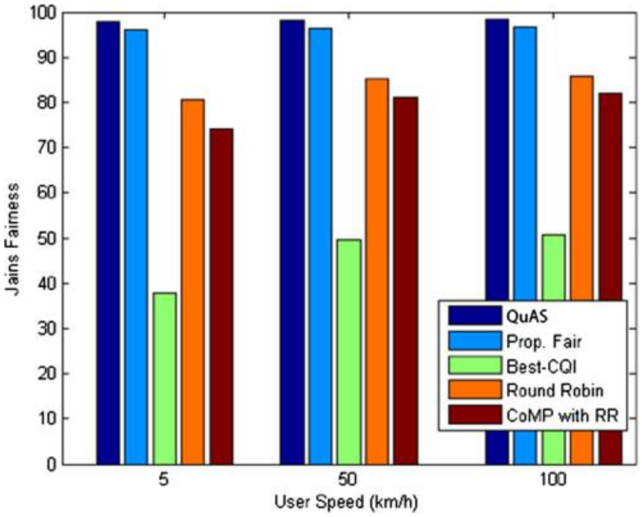

**FIGURE 14** Jain's fairness with mobility

Note that the main goal of QuAS algorithm of increasing QoS experience of the edge users also means allocating more resources to them. This is the main reason behind higher fairness results produced by QuAS algorithm and this also causes a decrease in the overall throughput as expected by the trade-off between fairness and throughput. As being a well-known algorithm for providing good fairness results, PF comes second after QuAS algorithm.

It can be observed that all of the algorithms tend to produce better fairness results as the user speeds increase. Increasing speed means experiencing worse channel conditions for users, and as mentioned above, this means more users are starting to act as edge users if speed increases. Allocating more resources to edge users allows the fairness index to increase.

As the user speeds increase, quality of the channel conditions decreases oppositely. This is why providing users with necessary packet delivery times becomes harder according to increasing speeds. As with Jain's fairness index, best-CQI gives the lowest results which is outperformed by QuAS algorithm. PF, RR, and CoMP with RR provide acceptable results about QoS fairness.

From the simulations about mobility, it can be observed that user channel conditions tend to decrease when increasing speeds. As a result, average peak throughput and average cell throughput decrease while average edge throughput increases.

Figure 15 depicts average QoS fairness values for the scheduling algorithms conjointly with mobility. It can be seen that QuAS algorithm again performs the best, with a 5% higher ratio than the PF algorithm and outperforming all of the algorithms.

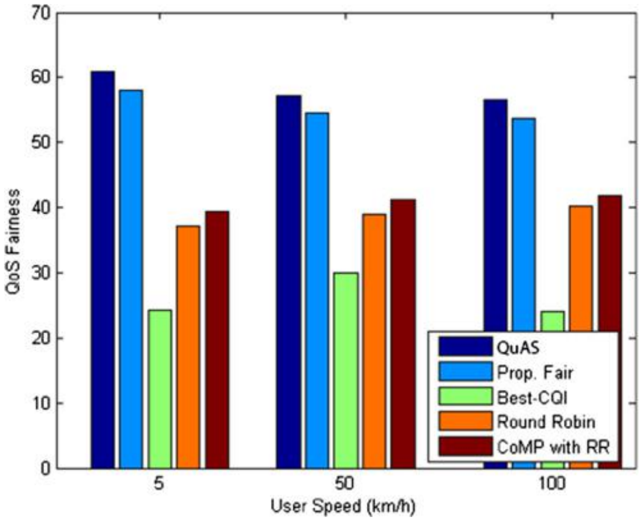

**FIGURE 15** QoS fairness with mobility

## 4.7 | Carrier frequency

LTE networks deployed in various countries work in different carrier frequency bands ranging between 700 to 3500 MHz. In the simulations, 800, 1800, and 2100 MHz frequency bands, which are used in European countries for LTE networks, were chosen to demonstrate the effects of carrier frequency on throughput, fairness, and QoS fairness performances of the scheduling algorithms.

The simulation results show that average peak throughput tends to increase according to the increase of carrier frequency bands for best-CQI algorithm. It stays stable for PF and QuAS algorithm and tends to decrease for RR and CoMP with RR algorithms. Figure 16 depicts the average peak throughput results of the five evaluated scheduling algorithms.

It can be observed that users having better channel conditions get better throughput from the network when carrier frequency becomes higher. As being the fairest algorithm, QuAS algorithm tries to allocate the resources more equally to the users, and this results in short changes about peak throughput. RR and CoMP with RR algorithms do not take channel conditions into account while allocating resources and this causes peak throughput to decrease as the carrier frequency bandwidth increases.

Figure 17 shows that the best edge throughput results are achieved at 1800 MHz carrier frequency among the three frequencies occupied in simulations. QuAS algorithm provides the best results which are 10% higher than its closest follower, PF algorithm. It also outperforms RR and CoMP with RR algorithms with over 330% better results.

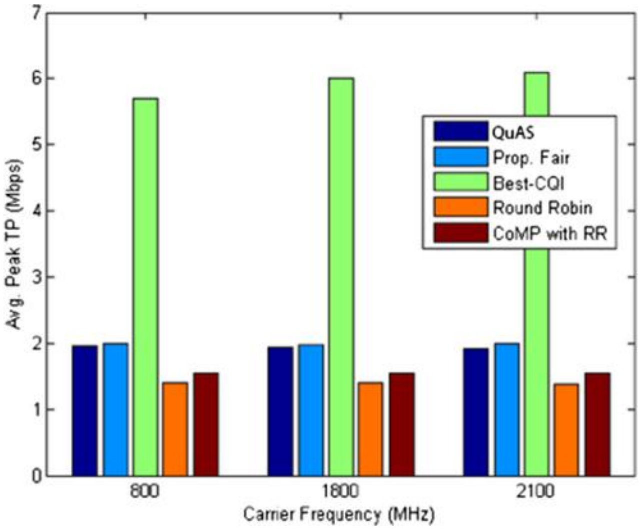

**FIGURE 16** Avg. peak TP with carrier frequency

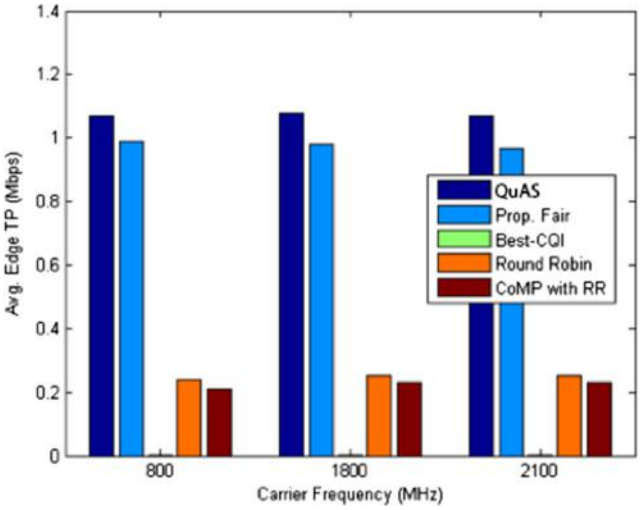

**FIGURE 17** Avg. edge TP with carrier frequency

The simulations show that the average cell throughput stays similar and is not affected too much by changing carrier frequency bands. Figure 18 represents the average cell throughput values for evaluated algorithms under 800, 1800 and 2100 MHz carrier frequencies.

The investigation of results for Jain's fairness index under different carrier frequencies show that QuAS algorithm generates the best results on all of the simulations. As can be seen from Figure 19, PF algorithm provides second best results, about 2% lower than QuAS algorithm on the average. QuAS algorithm and PF algorithm generates their best fairness results on 1800 MHz.

The simulation results presenting QoS fairness results along with carrier frequencies are show in Figure 20. QuAS algorithm performs the best with about 5.5% higher results than the PF algorithm on the average. PF algorithm becomes the second with 55% QoS fairness results on the average. RR and CoMP with RR algorithms perform their highest results on 800 MHz frequency. On the average, QuAS algorithm outperforms best-CQI, RR, and CoMP with RR algorithms by generating about 59% QoS fairness results on the average.

## 4.8 | Antenna configuration (MIMO)

In the design phase of LTE network, Multiple Input–Multiple Output (MIMO) has been proposed to develop the throughput of the system by using two or more antennas to transmit and receive two or more different data flows

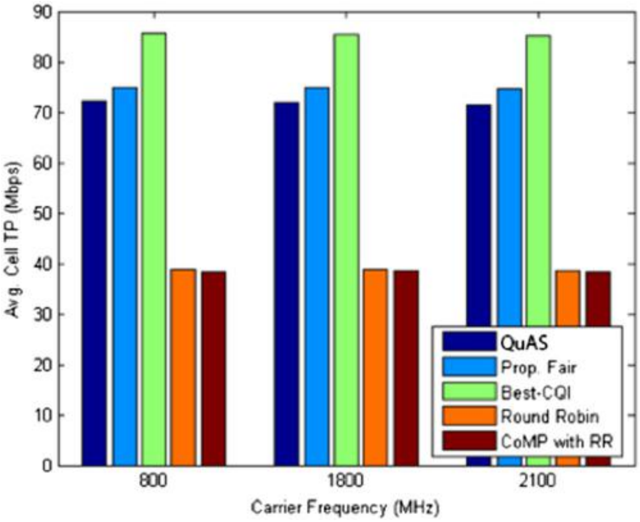

**FIGURE 18** Avg. cell TP with carrier frequency

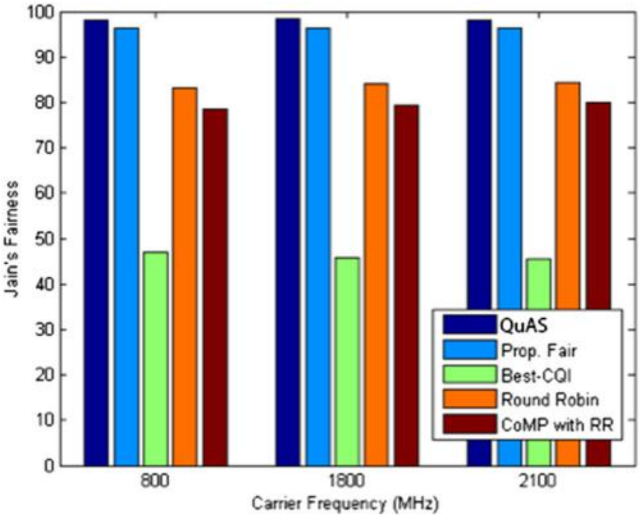

**FIGURE 19** Jain's fairness with carrier frequency

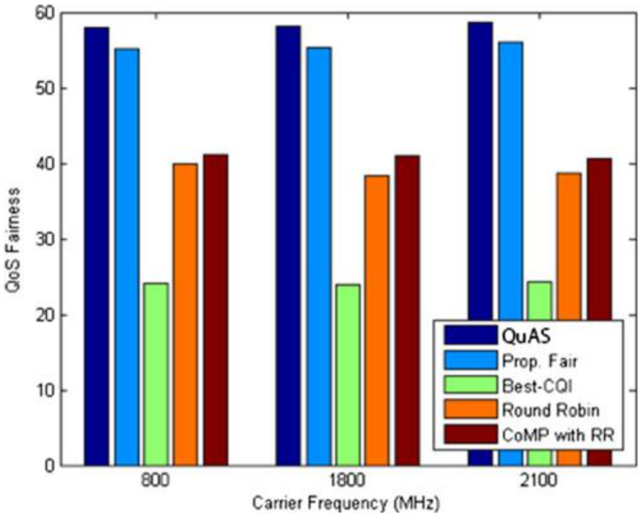

**FIGURE 20** QoS fairness with carrier frequency

simultaneously both in UE side and the eNodeB side. During the simulations, we have employed SISO (1 × 1) and MIMO (2 × 2 and 4 × 4) antenna configurations to observe the effects of multiple input multiple output on the network.

The effects of MIMO on the average edge throughput of the network can be found in Figure 21. MIMO affects edge throughput results dramatically.

The biggest improvement of edge throughput is observed with the QuAS algorithm with about 23.5% increase from 1 × 1 to 2 × 2 and 12.5% increase from 2 × 2 to 4 × 4 antenna configurations. CoMP with RR provides its highest results and passes RR with 4 × 4 MIMO. PF algorithm also increases edge throughput with MIMO; however, the rise is limited when compared with QuAS algorithm.

The results of simulations, which depict the average cell throughput values according to the antenna configuration, can be found in Figure 22. From the average edge throughput values examined above, it is expected to occur a serious change in the average cell throughput values.

The average cell throughput tends to change highly when moving from 1x1 to 2x2 antenna configuration. For the QuAS, RR and CoMP with RR algorithms, average cell throughput increases. On the other hand, cell throughput decreases for best-CQI and PF algorithms.

Figure 23 demonstrates the performances of examined algorithms about Jain's fairness index along with different antenna configurations. Moving from 1 × 1 SISO to 2 × 2 MIMO brings a valuable increase to the Jain's fairness index results. On the other hand, moving from 2 × 2 MIMO to 4 × 4 MIMO does not provide a notable gain, although it still generates higher results on the fairness results.

The simulation results demonstrating the QoS fairness results of the inspected algorithms are shown in Figure 24. Different from Jain's fairness index, QoS fairness takes the delay needs of the users into consideration, that is, how

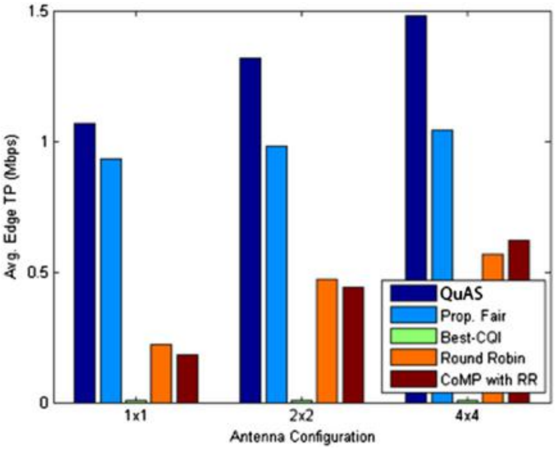

**FIGURE 21** Average edge throughput with MIMO

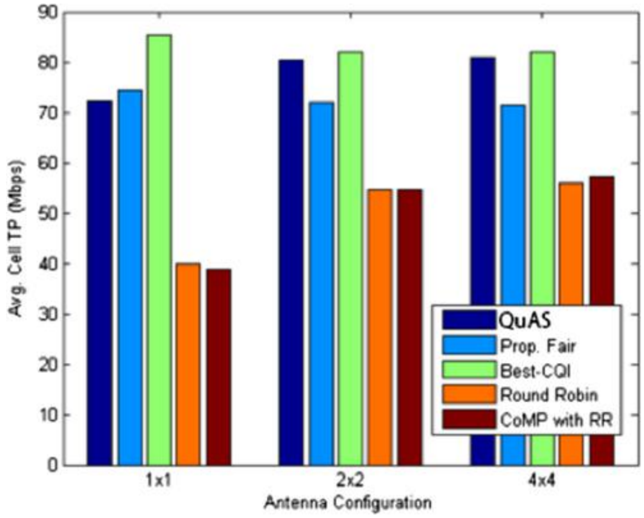

**FIGURE 22** Average cell throughput with MIMO

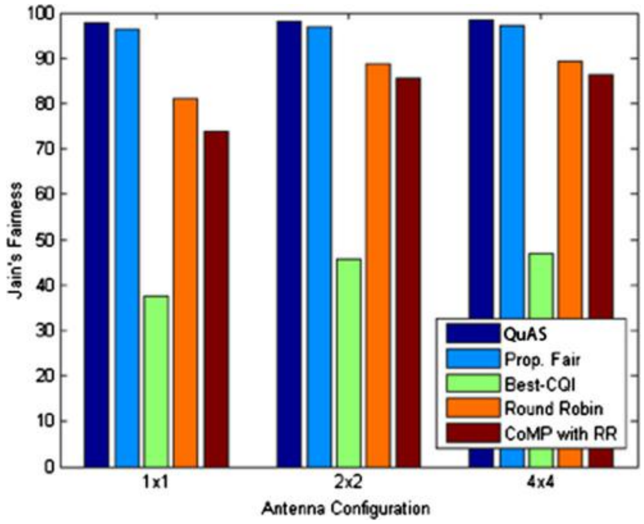

**FIGURE 23** Jain's fairness index results with MIMO

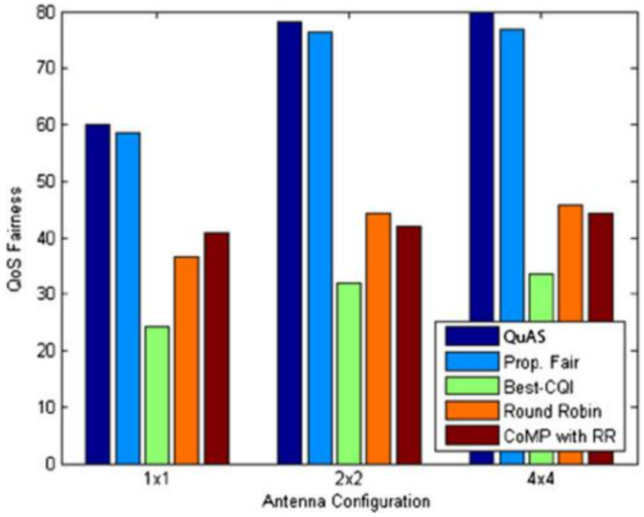

**FIGURE 24** QoS fairness index results with MIMO

**TABLE 4** Proposed algorithm vs proportional fair

| Advantages | Disadvantages |
|---|---|
| Edge TP increase 10% | Peak TP decrease 1.8% |
| Jain fairness increase 2% | Avg. cell TP decrease 3.5% |
| QoS fairness increase 6% | |

many packets of a user could have been delivered on time is the main focus of QoS fairness. With respect to this paradigm, since MIMO is designed to increase the overall throughput of the network, it can also be expected to increase the QoS fairness by providing more throughput to the users.

Moving from 1 × 1 to 2 × 2 antenna configuration generates a serious increase of the QoS fairness index for all of the inspected algorithms. On the other hand, the change about QoS fairness increases lesser with 4 × 4 MIMO. It can be observed from the figure that QuAS algorithm generates the highest results with all of the antenna configurations.

## 5 | CONCLUSION

In this study, we propose a new QoS-aware downlink scheduling algorithm QuAS to enhance the QoS experience of mobile network users. Unlike the existing studies, we especially concentrate on the performance of the edge users as they gain meager throughput and high delay in the cell. The primary goal of the QuAS algorithm is to enhance the QoS experience of the edge users while avoiding a significant loss in overall system throughput and QoS. For this purpose, the scheduler uses packet size and delay information of the users to define the allocation of the RBs. The simulations for evaluating performance of the algorithms are held under various scenarios, such as static and mobile users, Coordinated Multi Point (CoMP) structure, with various parameters, like carrier frequency, number of users and eNodeBs.

Simulation results indicate that QuAS algorithm provides very good results about edge throughput, Jain's fairness and QoS fairness, especially when the number of users is smaller. QuAS algorithm specifically aims at providing better QoS results than Proportional Fair algorithm, which is taken as reference since it is the algorithm that provides best fairness values about Jain's fairness. The simulations show that allocating the network resources according to the delay needs and packet sizes of the users brings several advantages over standard PF algorithm. As shown in Table 4, the QuAS algorithm results in 10% higher edge throughput, 2% higher fairness, and 6% higher QoS fairness when compared to PF algorithm. However, there is still a 1.8% decrease in peak throughput. The reason of this decrease is giving some more of the resources to the edge users instead of the users closer to eNodeBs in order to fulfill their QoS needs. Since channel quality is not very good for edge users, they can get less throughput from the eNodeB, and this brings an expected but limited decrease in peak and overall cell throughput.

The rearrangement of the resources according to delay and packet sizes of the users brings notable advantages about edge throughput and QoS fairness over Proportional Fair algorithm. On the other hand, QuAS algorithm does not cause a significant decrease in peak throughput and cell average throughput.

## ACKNOWLEDGEMENT

The work of V.C. Gungor was supported by BAGEP and AGU Foundation in Turkey.

## ORCID

*Vehbi Cagri Gungor* https://orcid.org/0000-0003-0803-8372

## REFERENCES

1. Barnett TJ, Sumits A, Jain S, Andra U. “Cisco visual networking index (VNI) update global mobile data traffic forecast,” Cisco VNI White Pap., 2015.
2. Akan A, Onen E, Chaparro LF. “Time-frequency based robust ofdm channel equalization,” 2007, No Eusipco, pp. 493–496.
3. Yildiz O, Sokullu R. “A novel mobility aware downlink scheduling algorithm for LTE-A networks,” ICUFN, pp. 300–305, 2017.

4. Kim H, Kim K, Han Y, Yun S. A proportional fair scheduling for multicarrier transmission systems. *Veh Technol Conf*. 2004;1:409-413.
5. Sun Z, Yin C, Yue G. Reduced-complexity proportional fair scheduling for OFDMA systems. *Int Conf Commun Circuits Syst*. 2006;0(60472070):1221-1225, 2006.
6. Toseef U, Weerawardane T, Timm-Giel A, Gorg C, Kroner H. “Adaptive fair radio interface scheduling for LTE networks,” 2012 9th Int. Conf. High Capacit. Opt. Networks Enabling Technol. HONET 2012, pp. 21–26, 2012.
7. Sudheep S, Rebekka B. “Proportional equal throughput scheduler—a very fair scheduling approach in LTE downlink,” 2014 Int Conf Inf Commun Embed Syst ICICES 2014, no. 978, 2015.
8. Alqahtani SA, Alhassany M. “Performance modeling and evaluation of novel scheduling algorithm for LTE networks,” *2013* IEEE 12th Int Symp Netw Comput Appl, pp. 101–105, 2013.
9. Liu D, Lee YH. “An efficient scheduling discipline for packet switching networks using earliest deadline first round robin,” Proc - Int Conf Comput Commun Networks, ICCCN, vol. 2003–Janua, no C, pp. 5–10, 2003.
10. Bin L, Tian H, Xu L. “An efficient downlink packet scheduling algorithm for real time traffics in LTE systems,” 2013 IEEE 10th Consum Commun Netw Conf CCNC 2013, pp. 364–369, 2013.
11. Trabelsi S, Belghith A, Zarai F. “Performance evaluation of a decoupled-level with QoS-aware downlink scheduling algorithm for LTE networks,” *2015 IEEE Int. Conf. Data Sci. Data Intensive* Syst, pp. 696–704, 2015.
12. Akyildiz HA, Akkuzu B, Cirpan HA. “A QoS-aware reconfigurable LTE MAC scheduler,” 2014 *IEEE 22nd Signal Process. Commun. Appl. Conf. (SIU* 2014*)*, no. Siu, pp. 1467–1470, 2014.
13. Zaki Y, Weerawardane T, Görg C, Timm-giel A. “Multi-QoS-aware fair scheduling for LTE,” Veh. Technol. Conf. (VTC Spring), 2011 IEEE 73rd, 2011.
14. Ferdosian N, Othman M, Mohd B, Yeah K. Throughput-aware resource allocation for QoS classes in LTE networks. *Procedia - Procedia Comput Sci*. 2015;59, no. Iccsci:115-122.
15. Ferdosian N, Othman M. Fair-QoS broker algorithm for overload-state downlink resource scheduling in LTE networks. *IEEE Syst J*. 2017;99:1-12.
16. Al-Shuraifi M, Al-Zayadi H, Lavriv O, Klymash M. “Improving QoS in MAX C/I scheduling using resource allocation type 1 of LTE,” Exp des Appl CAD Syst Microelectron (CADSM)*,* 2015 *13th Int. Conf.*, pp. 12–14, 2015.
17. Gungor VC, Uyan OG. “LTE Ağları için Servis Kalitesi Farkında Aşağı Yönlü Çizelgeleme Algoritması: Kenar Kullanıcıları Üzerine İnceleme QoS-Aware Downlink Scheduling Algorithm for LTE Networks: A Case Study on Edge Users,” in SIU, 2017.
18. Tian J, Zhang H, Wu D, Yuan D. QoS-constrained medium access probability optimization in wireless interference-limited networks. *IEEE Trans Commun*. 2017;66(3):1064-1077.
19. Zhong Y, Quek TQS, Member S, Ge X, Member S. Spatio-temporal traffic: delay analysis and scheduling. *IEEE J Sel Areas Commun*. 2017;35(6):1373-1386.
20. Tian J, Zhang H, Wu D, Yuan D. Interference-aware cross-layer design for distributed video transmission in wireless networks. *IEEE Trans Circuits Syst Video Technol*. 2015;8215, no. c:1.
21. Soni K, Tyagi A. “A suboptimal QoS aware multiuser scheduling for 3GPP LTE network,” 2015 Second Int. Conf. Adv. Comput. Commun. Eng., 2015.
22. Wu X, Han X, Lin X. “QoS oriented heterogeneous traffic scheduling in LTE downlink,” IEEE ICC 2015 - Mob Wirel Netw Symp, pp. 3088–3093, 2015.
23. Jiang Z, Zhang W, Gao S. An improving strategy for LTE downlink scheduling algorithms based on QoS requirement. *Commun Qual Reliab (CQR)*. 2015 *IEEE Int. Work. Tech. Comm.*;(1):2015.
24. Wang Y, Huang H. “A QoS-based fairness-aware downlink scheduling in LTE-advanced,” *2014 Int. Conf. Network-Based* Inf Syst, pp. 470–475, 2014.
25. Rupp M. *The Vienna LTE-Advanced Simulators*. 1st ed. Singapore: Springer; 2016.
26. Jain R, Durresi A, Babic G. Throughput fairness index: an explanation. *ATM Forum Contrib*. 1999;45:1-13.

**How to cite this article:** Uyan OG, Gungor VC. QoS-aware LTE-A downlink scheduling algorithm: A case study on edge users. *Int J Commun Syst*. 2019;e4066. https://doi.org/10.1002/dac.4066