# Semi-Static Cell Differentiation And Integration With Dynamic BBU-RRH Mapping In Cloud Radio Access Network

M.Khan, *Student Member*, IEEE, Zainab H. Fakhri, H.S. Al-Raweshidy, *Senior Member*, IEEE

***Abstract*—In this paper, a Self-Organising Cloud Radio Access Network is proposed, which dynamically adapt to varying network capacity demands. A load prediction model is considered for provisioning and allocation of Base Band Units (BBUs) and Remote Radio Heads (RRHs). The density of active BBUs and RRHs is scaled based on the concept of cell differentiation and integration (CDI) aiming efficient resource utilisation without sacrificing the overall QoS. A CDI algorithm is proposed in which a semi-static CDI and dynamic BBU-RRH mapping for load balancing are performed jointly. Network load balance is formulated as a linear integer-based optimisation problem with constraints.The semi-static part of CDI algorithm selects proper BBUs and RRHs for activation/deactivation after a fixed CDI cycle, and the dynamic part performs proper BBU to RRH mapping for network load balancing aiming maximum Quality of Service (QoS) with minimum possible handovers. A Discrete Particle Swarm Optimisation (DPSO) is developed as an Evolutionary Algorithm (EA) to solve network load balancing optimisation problem. The performance of DPSO is tested based on two problem scenarios and compared to Genetic Algorithm (GA) and the Exhaustive Search (ES) algorithm. The DPSO is observed to deliver optimum performance for small-scale networks and near optimum performance for large-scale networks. The DPSO has less complexity and is much faster than GA and ES algorithms. Computational results of a CDI-enabled C-RAN demonstrate significant throughput improvement compared to a fixed C-RAN, i.e., an average throughput increase of 45.53% and 42.102%, and an average blocked users reduction of 23.149%, and 20.903% is experienced for Proportional Fair (PF) and Round Robin (RR) schedulers, respectively.**

***Index Terms* - Base Band Unit (BBU), Cloud Radio Access Network (C-RAN), Particle Swarm Optimisation (PSO), Remote Radio Head (RRH), Self-Organising Network (SON).**

## I. INTRODUCTION

The surging volume of mobile data traffic witnessed in the recent years is triggered by the dramatic growth in smart mobile devices, diverse mobile internet enabled applications, and ever increasing wireless access demands. It is predicted that the global mobile devices and connections will increase up to 20 billion or even 50 billion by 2020 [1]. One promising direction is to densify the access networks (especially at data traffic hot-spots) with small cells. However, network densification comes with even bigger challenges, i.e., the considerable increase in capital (CAPEX) and operational (OPEX) expenditure, a greater number of un necessary handovers among small cells, traffic load imbalances, and under-utilised network resources. Mobile Network Operators (MNOs) require innovative methods and solutions beyond traditional performance upgrades to extract maximum returns on investment while maintaining high levels of network QoS.

In general, a network may not function at its best performance levels if its resources are utilised inefficiently. Network resources are often under-utilised during unbalanced traffic situations, particularly when some network cells may suffer from heavy loads causing a high number of blocked users, while others remain lightly loaded with their resources underutilised. Therefore, it is crucial to achieving self-optimisation in the network on varying traffic environment. Inter-cell optimisation is a critical optimisation problem in Self Organising Networks (SON) for the Third Generation Partnership Project (3GPP) [2], [3].

Moreover, SON architectures can be divided into three types - a) centralised, b) decentralised and, c) hybrid. In the decentralised and hybrid SON architectures, the SON algorithm partially runs on the network management level and partially in the network elements. Coordination of different SON functions, possibly having conflicting goals and operating on various time scales, is more challenging than in the centralised architecture [4]. In the centralised structure, a central Network Management System (NMS) or a SON server decides the network optimisation algorithms and the eNodeB parameter configuration [5]. The centralised SON architecture is more manageable regarding the implementation of SON algorithms compared to distributed and Hybrid SON architectures. It enables the SON algorithms to jointly optimise multiple network parameters, therefore, allowing a globally tuned system. However, the centralised SON server in this approach requires strict latency and delay requirements regarding system KPIs and UE measurements for SON parameter updates, which restricts the applicability of a purely centralised SON architecture.

However, Cloud Radio Access Network (C-RAN) is a promising centralised network architecture that can support super-dense small cells deployment. C-RAN is considered to meet the challenges mentioned above and has attracted considerable attention by both academia and MNOs and is a key enabler of Next Generation Mobile Networks (5G) [6]–[8]. The C-RAN architecture consists of the following main parts:

- Several BBUs aggregated into a BBU cloud/pool for centralised management and processing.
- Distributed RRHs in a given geographical area.
- The connection between BBUs and RRHs (also refered to as *front-haul*) via an optical transport network

Unlike conventional cellular networks, where the base sta-

tions are not always in peak time and often work in idle states with their resources not fully utilised, in C-RAN, suitable resource allocation schemes can dynamically adjust the logical connection between BBUs and RRHs. It is necessary for a system to optimise its resources according to varying traffic environments. In C-RAN the problem of resource wastage is overcome by dynamically allocating the shared and centralised BBUs resources to the RRHs. Moreover, significant cost and energy savings can be achieved by dynamically scaling the BBUs concerning varying traffic caused by uneven user distribution in the network [9]. C-RAN is feasible to realise the coordinated control between multiple cells by centralised management.

Although the main features in SONs include self-configuration, self-optimisation, and self-healing. However, this paper emphasises on self-optimisation technique in C-RAN concerning network performance improvement. The primary focus is to model a multi-objective optimisation problem along with several other criteria necessary to tailor the optimisation objective according to specific system requirements. C-RAN combined with Self-optimising ability can provide MNOs with a flexible network regarding network dimensioning, adaptation to non-uniform traffic and efficient utilisation of network resources.. However, before a full commercial C-RAN deployment, several challenges need to be addressed. Firstly, the front-haul technology used must support enough bandwidth for delivering delay sensitive signals (i.e., the 1 ms physical layer processing requirement of LTE). Secondly, the proper BBU-RRH assignment in C-RAN to not only support collaboration technology like Cooperative Multipoint Processing (CoMP) but also enabling load balancing in the network. Moreover, significant energy savings can be achieved if the RRHs and BBUs are turned on/off in such a way that the QoS of the network is not degraded.

In this context, a two-stage design is proposed in this paper for efficient resource utilisation in a self-optimised C-RAN with real time BBU-RRH mapping. Network resources are utilised based on the concept of Cell Differentiation and Integration (CDI) which allows a cell(s) to split into multiple small cells and vice versa in response to a measured load information in one or more cells in the network. CDI allows C-RAN to adapt to varying capacity demands through resource provisioning and allocation. Resource provisioning not only resizes the number of BBUs in the pool to meet the fluctuating traffic demands but also scales the density of active RRHs required to serve a given geographical area. In the first stage, the optimum number of BBUs is computed to serve the load demand, and the RRHs are activated or deactivated based on the concept of CDI to handle network traffic load. In this paper, the number of BBUs required to serve the system load at a given time is computed based on a prediction model called Wiener process [10]. In the second phase, the proper BBU-RRH mapping is identified to avoid unbalanced network scenarios while maintaining high levels of QoS. The second stage in this paper is modelled as an integer based linear optimisation problem with constraints.

The rest of the paper is organised as follows: Section II presents a survey of related work; Section III presents the Self-Optimising C-RAN framework; Section IV presents the system model; Section V presents a model of load prediction and BBU estimation; Section VI illustrates the formulation for dynamic RRH-sector allocation problem. Section VII defines the CDI algorithm. Computational results are discussed in Section VIII. Finally, the paper is concluded in Section IX.

## II. Related Work

Numerous studies and methods on self-optimisation have suggested addressing the problem of load balancing in cellular networks via SON. When a traffic imbalance is detected among cells, operation parameters are autonomously adjusted such as antenna angle (Antenna tilt) [11] and/or handover parameters [12] to reduce the coverage area to achieve Mobility Load Balancing (MLB) [13]. In MLB, the handover thresholds are adjusted following traffic conditions which result in expansion or contraction of virtual transfer areas among adjacent cells and thereby reducing or increasing users in the cells. However, incorrect handover parameter adjustment can cause additional handovers in the network which often leads to handover ping-pongs/delays and radio link failures. Mobility Robustness Optimisation (MRO) [14] is a SON function which aims to eliminate link failures and reduce unnecessary handovers caused by incorrect handover parameters. Power adaptation for load balancing is another technique to effectively change the cell coverage area which in return changes the association of all users in the coverage area. In LTE, Cell Range Expansion (CRE) [15] is a technique which allows Low Power Nodes (LPN) to expand their coverage area and take in users from the Macro Cell. Usually, users associate to the cell which provides the strongest signal. However, in CRE users connect to the LPNs despite receiving the strongest signal from the Macro cell. A comprehensive survey on self-organisation in future cellular networks, which includes a detailed description of the schemes mentioned above along with hybrid approaches and other existing SON load balancing methods in the literature are provided in [16].

Moreover, the benefits of Artificial Intelligence (AI) techniques while designing load balancing SON algorithms are inevitable. Among numerous AI techniques, the Genetic Algorithm (GA) [17], [18] and Swarm intelligence [19] are the most embraced learning algorithms inspired the process of gene evolution and the natural actions of swarms of ants, a shoal of fish, a flock of birds etc, respectively.Many algorithms have been designed to mimic the behaviour of natural organisms, however, Particle Swarm Optimisation (PSO) [20] remains the backbone of swarm intelligence on which all other algorithms are built. Both GA and PSO are widely discussed in studies related to network planning, interference management, routing and coverage optimisation problems [21]–[24].

On the other hand, a number of research studies on enabling technologies for C-RAN exist. Here, some related studies on BBU-RRH mapping along with RRH-UE association are briefly described. In [25], the authors propose a cross-layer framework for downlink multi-hop C-RAN to improve throughput performance by optimising both physical and network layer resources. Also, RRHs beamforming vectors, user

RRH association, and network coding based routing are optimised in an overall design. In [26], the authors attempt to solve a joint RRH and precoding optimisation problem which aims to minimise network power consumption in a MIMO based user-centric C-RAN. In line with this work, the authors of [27] propose a weighted minimum mean square error (WMMSE) approach to solving the network-wide beam-forming vector optimisation problem for RRH-UE clusters formation. The BBU scheduling is then formulated as a bin packing problem for energy efficient BBU utilisation in a heterogeneous C-RAN environment. A dynamic BBU-RRH mapping scheme is proposed in [28] using a borrow-and-lend approach in C-RAN. Overloaded BBUs switch their supported RRHs to underutilised BBUs for a balanced network load and enhanced throughput. The authors of [29] proposed a lightweight, scalable framework that utilises optimal transmission strategies via BBU-RRH reconfiguration to cater dynamic user traffic profiles. [30] describes the traffic adaptation and energy saving potential of TDD-based heterogeneous C-RAN by adjusting the logical connections between BBUs and RRHs. The authors of [31] recently investigated an RRH clustering design and proposed a spectrum allocation genetic algorithm (SAGA) to improve network QoS via efficient resource utilisation.

Regarding other related work, research initiatives are taken to develop Network Function Virtualisation (NFV) and Software Defined Network (SDN) solutions for C-RAN [32]–[34]. NFV is an architectural framework that provides a virtualised network infrastructure, functions and NFV orchestrator for control and management [35]. However, SDN is a concept related to NFV. SDN decouples data and control plane to enable directly programmable control plane while abstracting underlying physical infrastructure from applications and services [36]. Although SDN and NFV are not the prime focus of this paper, they are presented in this section for completeness of the C-RAN introduction. Moreover, [8], [37] provides a comprehensive survey on C-RAN and highlights the challenges, advantages, and implementation issues regarding different deployment scenarios. Also, an in-depth review of the principles, technologies and applications of C-RAN describing innovative concepts regarding many physical layer, resource allocation, and network challenges together with their potential solutions are highlighted in [38].

To sum up, the existing resource allocation mechanisms in C-RAN does not take full advantage of the concept of centralised BBU pool. This paper extends the scope of C-RAN by introducing a concept of CDI with dynamic BBU-RRH mapping for load balancing and efficient resource utilisation. The system model in this article allows combining SON and C-RAN for a more centrally managed network operations. The proposed model is suitable for the framework of software defined front-haul with optical switching for C-RAN [29]. However, this paper only focuses on the centralised-SON aspect of the structure.

## III. Self-Optimising Cloud Radio Access Network Framework

In this paper, the self-organising framework proposed in the author's previous work [39] is utilised. The framework is applicable for short and long term dynamics of C-RAN and maximises the overall QoS of the system while considering a network load balance. Network QoS is maximised based on desired system KPIs. Note that, several performance indicators (KPIs) can be considered to measure the network QoS, so the framework is modelled as a general multi-objective optimisation problem including several criteria. Many other criteria may be included to tailor other optimisation objectives subject to specific operator policy requirements.

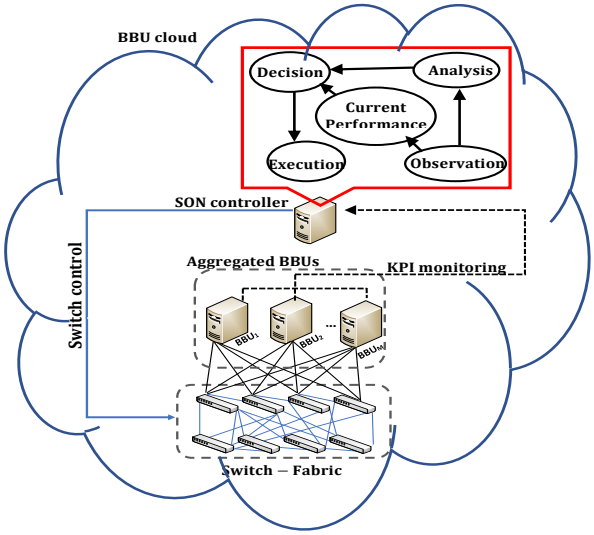



Fig. 1: Genetric concept of a Self organisation in C-RAN

Fig. 1 shows a generic concept of a self-organisation in C-RAN. The BBU cloud consists of aggregated BBUs pool and is connected to the RRHs via an optical transport network which may include a switch fabric (including a network of switches, optical splitters, multiplexers) [40] and low latency, high bandwidth fibre optic links as shown in Fig. 1. Note that, there are various possibilities of front-haul deployment in C-RAN [41]. The self-organisation concept is explained in phases as shown in Fig. 1, where a SON server/controller inside the BBU pool realises the self-organising concept. The observation and analysis phases are utilised to detect the performance of current network deployment (BBU-RRH configuration), and then an optimal implementation is identified for performance comparison. KPIs are used to monitor network status for current and optimal deployment settings. Based on the chosen KPIs, an algorithm decides the best system configuration, and finally, the new topology (BBU-RRH setting) is enforced in the execution phase (if necessary). The primary objective of SON server/controller in the proposed C-RAN architecture is as follows: (i) to compile required metric by discovering the status of each KPI and (ii) to produce a decision and enforce it. The BBUs feed the system KPIs to the main multi-objective decision-making algorithm hosted by SON server/controller. The weights or priority levels are then applied to each KPI for decision making. The corresponding weight of a KPI defines its preference value and is set according to network operator's preferences.

Fig. 2 provides a logical block diagram for a multiple objective decision making performed by the SON server/controller.

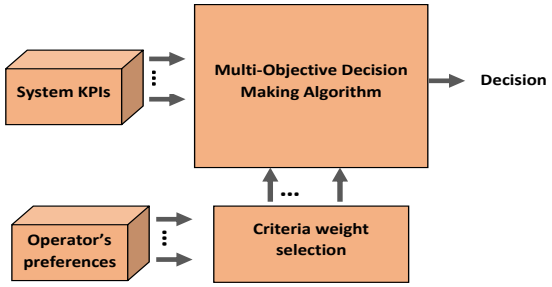



Fig. 2: Block diagram of multi-objective decision making logic for SON server

The BBUs feed the system KPIs to the main multi-objective decision-making algorithm hosted by SON server/controller. The weights or priority levels are then applied to each KPI for decision making. The corresponding weight of a KPI defines its preference value and is set according to network operator's preferences.

## IV. SYSTEM MODEL

### A. C-RAN Architecture

A self-optimised C-RAN (SOCRAN) architecture is presented in Fig. 3. The BBUs are decoupled from the RRH and migrated to a centralised BBU-pool, whereas the RRHs with simple RF transmission functions are left on the cell sites. A SON controller inside the BBU cloud monitors the BBU-pool resource utilisation as well as controls the front-haul. Each RRH is connected to the BBU pool via optical transport network (front-haul). Fig. 3 shows that each cell is covered by seven RRHs. In contrast, the same geographical area is served by a single high-power base station at extreme low load traffic conditions. As the traffic load reaches the resource limitation of the high-power base station, the geographical area differentiates into **C** equally sized small cells by activating the RRHs deployed. Furthermore, each of the **C** cells can further differentiate into **c** more small cells by activating the RRHs deployed within the cell to accommodate capacity demands. The term cell and RRH are used interchangeably throughout this paper since it is assumed that an RRH can serve only one cell at a particular time t. Moreover, each RRH can be served by only one BBU at a given time instance. The actual number of RRHs required in the network is determined by the coverage area, users density, and other environment-related factors, however, in this paper, both **C** and **c** are considered to be seven as a reasonable example.

In this paper, the concept of CDI is supported by considering three tiers of RRHs deployment in the network as shown in Fig. 4. Tier-3 RRH deployment imitates a single high-power Base station serving a Macro cell as in traditional cellular systems. Tier-2 and 1 represents a structure with universal frequency reuse, where each cell is surrounded by a continuous tier of $6 + \mathrm{E}$ and $6 \times [1 + \mathrm{j}] + \mathrm{E}$ cells, respectively. Where E represents the number of other external macro cells, and j accounts for the level of differentiation, e.g., level 1 represents any one of the tier-2 cells further differentiated; level 2 shows any two of tier-2 cells are further differentiated and so on as

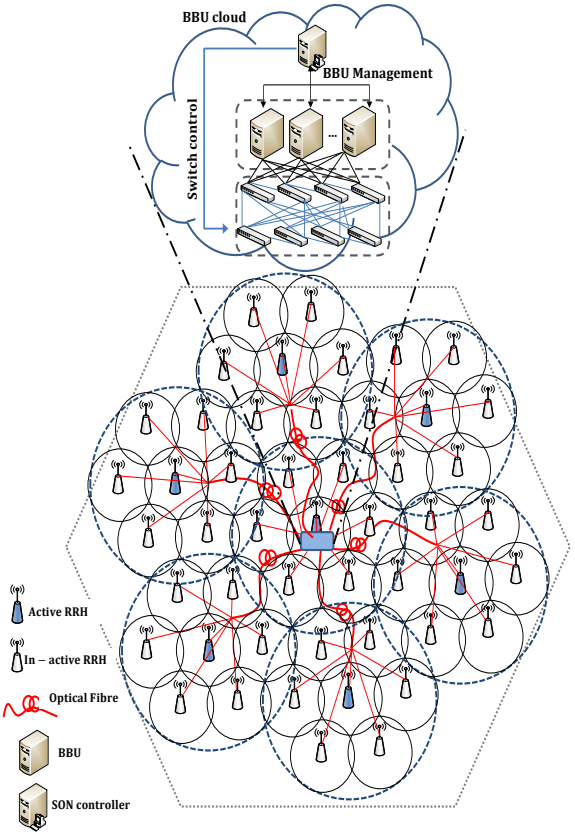



Fig. 3: Structure of a Cloud Radio Access Network represented as SON

shown in Fig. 4. A set $\mathbf{S}_i = \{\mathrm{RRH}_{i1}, \mathrm{RRH}_{i2}, ..., \mathrm{RRH}_{ic}\}$ is maintained for each cell $C_i$ in tier-2 RRH deployment, which contains a group of RRHs responsible for differentiating cell $C_i$ into $c$ small cells provided that the sum of transmit powers of all RRHs covers $C_i$ coverage area. The transmission power of the upper tier RRHs is split between the lower tier RRHs (including the original RRH). The SON server monitors all BBUs in the pool for traffic information and is responsible for cell differentiation and integration with proper BBU-RRH configurations, whereas the optical switch is in charge of realising the settings via server commands.

### B. System model constraints

This paper presents a centralised-SON architecture for C-RAN, a central server/controller collects reports about the BBUs and their users and informs the BBUs about the configuration along with switching the BBU-RRH configuration via server commands. Fine time-scale operations that happen within a few milliseconds, such as user scheduling in the frame, are executed by the BBUs themselves since they cannot be realised in the same report-configure-inform cycle due to latency limitations posed by the interfaces to/from the controller. Since the complete deployment of C-RAN has to pass through certain stages to extract its full potential [42], the final stage involves BBU resource pooling via NFV. In this sense, centralised SON can be seen as a direct extension of

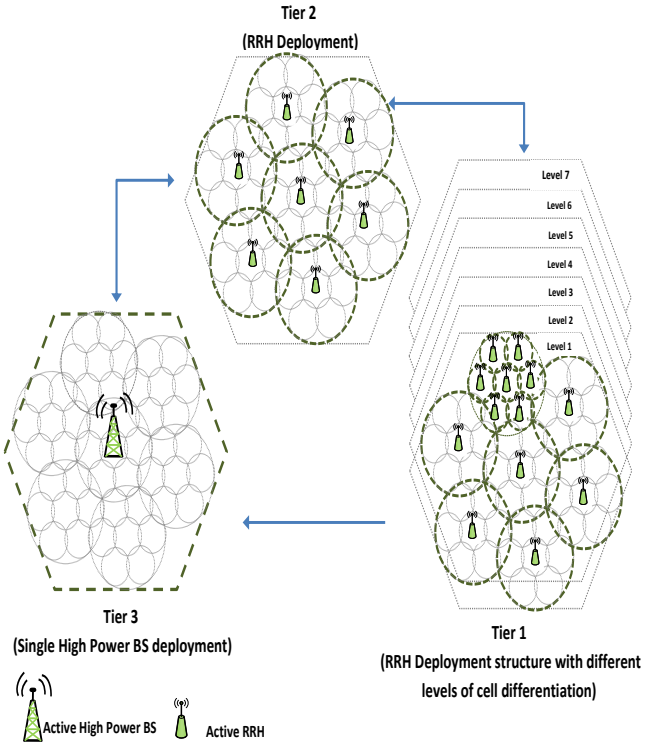



Fig. 4: Cell differentiation and integration with multiple tiers of RRH deployment.

Software Define Networks control/data plane separation principle, where resources can be virtualised thereby leveraging full benefits of C-RAN. The proposed System model allows for more efficient resource utilisation through centralised control across aggregated BBU resources. However, the model is constrained in the following ways:

- Since the SON server/controller is in charge of monitoring the BBU-cloud, the whole network may collapse in case of server/controller failure.
- Coarse time-scales may limit the optimisation process due to interface-latency between SON controller and the BBUs, along with the front-haul latency.
- The front-haul must support enough bandwidth for delivering delay sensitive signals, and the switching elements used to effect the BBU-RRH configurations must not affect the sub-frames time scale (i.e., 1ms)
- The sub-frame processing delay on a link between RRHs and BBU should be kept below 1 ms, to meet HARQ requirements [42]

Potential solutions to the challenges/limitations mentioned above are discussed in [43] and [44]. Furthermore, the strategic deployment of RRHs to support the proposed concept may or may not be feasible in real-time environments. The inclusion of random RRHs distribution with different cell sizes and shapes and the indefinite network extension in all directions is a more realistic approach. However, such an approach gives rise several other challenges such as RRHs being located very close to each other, increased inter-cell interference at cell edges, irregular cell shapes and sizes, and coverage holes within a geographical area among others. Such challenges need to be addressed within the CDI algorithm if a random RRH deployment is considered. Especially by considering all possibilities during cell differentiation and integration.

### C. Channel model

In this paper, Guaranteed Bit Rate (GBR) users with QoS requirements are considered. The frequency reuse factor is 1, and the time-frequency resources are equal for all BBUs. The basic unit of time-frequency resources that can be allocated to users is known as the Physical Resource Block (PRB). Let **M** and **N** represent the number of BBUs and RRHs in the network, respectively, such that $\mathrm{K}_{in}$ represents the total number of users in cell $i$ served by RRH $n$. Each user reports Channel Quality Information (CQI) to its serving BBU every two subframes (i.e., 2 milli-seconds) for proper PRB assignment. The channel model considered in this paper is a composite fading channel which involves path-loss and both small and large scale fading, given as:

$$\mathrm{H}_{k_{in}} = \mathrm{h}^{*}_{k_{in}} l_{k_{in}} \left[\mathrm{AD}^{-\delta}_{k_{in}}\right] \tag{1}$$

where $\mathrm{h}^{*}_{k_{in}}$ and $l_{k_{in}}$ represent the small and large scale fading channel between the RRH $n$ and user $k$ in cell $i$, respectively. The small scale fading is assumed to be a Rayleigh random variables with a distribution envelop of zero-mean and unity-variance Gaussian process. $\mathrm{AD}^{-\delta}_{k_{in}}$ reflects the path-loss between RRH $n$ and user $k$ in cell $i$, where A is a constant which depends on the carrier frequency $f_c$ and $\mathrm{D}_{k_{in}}$ is the distance between user $k$ and RRH $n$ in cell $i$ and a path-loss exponent of $\delta$. The large scale fading is assumed to be lognormal random variable with a standard deviation of 10dB and is typically modelled with a probability density function of [45]:

$$\rho(l) = \frac{\zeta}{\sqrt{2\pi}\sigma_l l} \exp\left[-\frac{(10\log_{10} l - \mu_l)^2}{2\sigma_l^2}\right] \tag{2}$$

where $\zeta = 10/\ln 10$, and $\mu_l$ and $\sigma_l$ are the mean and the standard deviation of $l$, both expressed in decibels.

The instantaneous Signal-to-Interference-and-Noise-Ratio $\gamma$ based on CQI received from user $k$ in cell $i$ served by RRH $n$ at the $\mathrm{p}^{\mathrm{th}}$ PRB of subframe $\tau$ is expressed as

$$\gamma_{k_{in},p}(\tau) = \frac{\mathrm{H}_{k_{in},p}(\tau)\mathrm{P}_{in,p}(\tau)}{\mathrm{N}_0 + \sum_{j\in C}\sum_{a\in c, a\neq n}\mathrm{H}_{k_{ja},p}(\tau)\mathrm{P}_{ja,p}(\tau)} \tag{3}$$

where $\mathrm{P}_{in,p}(\tau)$ and $\mathrm{H}_{k_{in},p}(\tau)$ are the transmit power and channel gain between the serving RRH $n$ of user $k$ in cell $i$ at $\mathrm{p}^{\mathrm{th}}$ PRB of subframe $\tau$. $\mathrm{N}_0$ is the power of Additive White Gaussian Noise per PRB and $\sum_{j\in C}\sum_{a\in c, a\neq n}\mathrm{H}_{k_{ja,p}}(\tau)\mathrm{P}_{ja,p}(\tau)$ represents the interference power from all other RRHs $a$ in cells $j$ except the serving RRH $n$ of user $k$ on subframe $\tau$ in cell $i$. The average spectral efficiency of a user $k$ served by RRH $n$ in cell $i$ at any time instance $t$ is given as

$$\vartheta_{k_{in}}(t) = \frac{1}{\mathrm{RB}.\mathrm{N}_\tau}\left\{\sum_{\tau\in[t-1,t]}\sum_{p\in\mathrm{RB}}\log_2\left[1+\gamma_{k_{in},p}(\tau)\right]\right\} \tag{4}$$

where RB is the total number of PRBs assigned to a BBU and $N_\tau$ is the number of sub-frames in a load balancing cycle. We assume that the dynamic load balancing cycle is 2 seconds (i.e., 2000 subframes in 1 cycle). To keep up with the requirement throughput $\phi_k$, the number of PRBs required by user $k_{in}$ at a time period $t$ can be calculated by multiplying the achievable user throughput to the PRB bandwidth (i.e., 180 KHz per PRB)

$$N_{RB}^{k}(t) = \left\lceil \frac{\phi_k(t)}{p_{BW}.\vartheta_{k_{in}}(t)} \right\rceil \tag{5}$$

where $p_{BW}$ represents the bandwidth of a PRB and $\lceil . \rceil$ is the ceil function.

## V. Load Prediction and BBU estimation

This paper proposes a semi-static cell differentiation and integration scheme, which determines the number of BBUs and RRHs to be activated/deactivated depending on load distribution across the network to accommodate capacity demands. The BBUs and RRHs needed to handle traffic load is determined at the end of each cell differentiation or integration (CDI) cycle. The CDI cycle is a constant time interval after which a decision on required number of active BBUs and RRHs is made. A CDI cycle of 60 seconds is considered in this paper. However, to avoid constant scaling of BBUs due to load fluctuations, future load demand change is predicted to calculate the number of required BBUs. A decreasing predicted load might require scaling down the BBUs and integration of multiple cells into a single cell (i.e., from lower to higher tier RRH structure). However, an increasing predicted load might scale up the BBUs and further differentiation of cells (i.e., from higher to lower tier RRH structure).

### A. Load Prediction Based on Wiener Processes

Let $\eta_m(t)$ be the load of a BBU at time period (t), which is represented as

$$\eta_m(t) = \frac{\sum_{k=1}^{K} I_{m,k}(t) N_{RB}^{k}(t)}{RB} \tag{6}$$

Where $I_{m,k}$ is a binary indicator such that $I_{m,k} = 1$ if user $k$ is served by $BBU_m$. However, an important constraint $\sum_{m=1}^{M} I_{m,k} = 1, \forall k$ defines that each user k is served by only one BBU at time period $t$. Note that, all BBUs are assigned the same number of PRBs. Another important constraint is that $\sum_{k=1}^{K} I_{m,k}(t) N_{RB}^{k}(t) \leq RB, \forall m$, which states that the number of PRBs assigned to users served by the same BBU should not exceed the BBU PRB limitation. The total actual load on the network at time $t$ is represented as the aggregated load on each active BBU at time $t$, which is given by

$$\eta(t) = \sum_{m=1}^{M} \eta_m(t) \tag{7}$$

The predicted total load on the network at time $t + \Delta t$ is modelled as a stochastic process based on Wiener Processes. The idea is to utilise accurate records of current and past load values to predict future load demands [10]. Let the total load on the network at time $t$ and $t + \Delta t$ is given by $\eta(t)$ and $\eta(t + \Delta t)$ respectively, then

$$\eta(t + \Delta t) = \eta(t) + \Delta\eta(t) \tag{8}$$

where $\Delta\eta(t)$ represents the load variation between time $t$ and $t + \Delta t$ and can be modelled as

$$\Delta\eta(t) = \mu\Delta t + \epsilon\delta\sqrt{\Delta t} \tag{9}$$

where $\epsilon$ is standard normal random variable with zero mean and standard 1. $\Delta\eta(t)$ is a normally distributed random variable with mean $\mu\Delta t$ and standard deviation $\delta\sqrt{\Delta t}$. The $\mu$ and $\delta$ in (9) are refered to as *expected drift-rate* and *standard deviation rate* of $\Delta\eta(t)$ respectively, since for any arbitrary time interval $\Delta t$ the mean $\mu\Delta t$ and standard deviation $\delta\sqrt{\Delta t}$ are directly calculated from $\mu$ and $\delta$.

The values of $\mu$ and $\delta$ in (9) can be estimated by taking previous $s$ sample values of $\eta$ i.e., $[t, t - \check{\tau}], [t - \check{\tau}, t - 2\check{\tau}], ..., [t - (s-1)\check{\tau}, t - s\check{\tau}]$. Where $\check{\tau}$ is the sampling time interval. The estimator $\hat{\mu}$ of $\mu$ and $\hat{\delta}$ of $\delta$ are given as:

$$\hat{\mu} = \frac{\sum_{i=0}^{s} \left(\eta\left(t - i\check{\tau}\right) - \eta\left(t - i\check{\tau} - \check{\tau}\right)\right)}{s\check{\tau}} \tag{10}$$

$$\hat{\delta} = \frac{1}{\sqrt{\check{\tau}}} \sqrt{\frac{\sum_{i=1}^{s} \left(\eta\left(t - i\check{\tau}\right) - \eta\left(t - i\check{\tau} - \check{\tau}\right) - \hat{\mu}\check{\tau}\right)^2}{s}} \tag{11}$$

From (10) and (11), $\Delta\eta(t)$ in (9) is calculated and the total network load $\eta(t + \Delta t)$ in (8) is predicted for the next time $t + \Delta t$.

### B. Number of BBUs required in the network

The required number of BBUs to serve the offered traffic load at a particular time $t$ can be calculated using predicted network load $\eta(t + \Delta t)$ and actual load $\eta(t)$, such that:

$$\text{No. of BBUs} = \begin{cases} \lceil \eta(t + \Delta t) \rceil & \text{if } \eta(t) \leq \eta(t + \Delta t) < M \\ |M| & \text{if } \eta(t + \Delta t) > M \\ \lceil \eta(t) \rceil & \text{if } \eta(t + \Delta t) < \eta(t) < M \end{cases} \tag{12}$$

where M is the total number of BBUs in the BBU pool and the notation $\lceil . \rceil$ is the ceil function. Moreover, the load contributed by an active $RRH_n$ in the network is given by

$$\eta_{RRH_{in}}(t) = \sum_{k=1}^{K} I_{k,in}(t) N_{RB}^{k}(t) \tag{13}$$

where $I_{k,in}(t)$ is a binary variable which indicates the association of user $k$ with RRH $n$ of cell $i$ at time $t$. $I_{k,in}(t)$ equals to 1 if user $k$ is associated with RRH $n$ at time $t$ in cell $i$.

## VI. Dynamic BBU-RRH configuration and formulation

For a SOCRAN architecture shown in Fig. 3, it is essential to balance the network load amongst the active BBUs by proper BBU-RRH configuration. After each CDI cycle, the network may reconfigure itself by scaling the BBUs and RRHs with respect to traffic load. However, during the process, the RRH to BBU mapping might not satisfy the QoS requirement.

Therefore, If the BBU-RRH configuration at time $t$ is known then it is necessary to adjust the BBU-RRH configuration at time $t+1$ to adaptively balance the variance in traffic demands. Note that, the time between $t$ and $t+1$ is longer than that of a subframe (i.e., one millisecond) and is called the load balancing cycle. A user location indicator vector $\mathbf{u} = \{\mathrm{u}_1, \mathrm{u}_2, ..., \mathrm{u}_\mathrm{K}\}$ is defined which shows users association with RRHs such that $\mathrm{u}_\mathrm{k} = \{\mathrm{r}_{in} | \mathrm{r}_{in} \in \mathbb{Z}^+ : i, n = 1, 2, 3, ...C\}$, where $\mathrm{u}_\mathrm{k} = r_{in}$ if user $k$ is associated with RRH $n$ of cell $i$. To indicate RRHs association with BBUs, a vector $\mathbf{r} = \{\mathrm{r}_{11}, \mathrm{r}_{12}, ..., \mathrm{r}_{in}\}$ is defined, where $\mathrm{r}_{in} \in \{1, 2, ..., \mathrm{M}\}$ and $\mathrm{r}_{in} = m$ indicates RRH $n$ of cell $C_i$ is being served by BBU $m$. Whereas, $\mathrm{r}_{in} = 0$ indicates that RRH $n$ of cell $C_i$ is not active. If the user location indicator vector $\mathbf{u}$ is given, then the problem is to identify the new RRH allocation vector $\mathbf{r}$.

Network performance determined by Key Performance Indicators (KPIs) indicates its QoS. Based on these KPIs, the SON server identifies optimum BBU-RRH conguration by utilising the existing number of active BBUs and RRHs, to achieve a highly stable network with highest achievable QoS with respect to load demand. Following are the important KPIs considered for BBU-RRH mapping problem;

### A. Key Performance Indicator for Load Fairness Index

In this paper a Jains fairness index $\psi$ is monitored, which determines the level of load balancing in the network at a particular time and is evaluated by using the load distribution in all cells. The Jains fairness index [46] at time $t$ can be defined as

$$\psi(t) = \frac{\left(\sum_{m=1}^{\mathrm{M}} \eta_m(t)\right)^2}{|\mathrm{M}| \left(\sum_{m=1}^{\mathrm{M}} \eta_m^2(t)\right)} \tag{14}$$

where $|\mathrm{M}|$ is the required number of active BBUs computed in (12). The range of $\psi$ is in interval $[\frac{1}{M}, 1]$, with higher value representing a highly balanced load distribution amongst all active BBUs. Therefore, maximising $\psi$ is one of the objectives of this work to achieve a highly balanced load in the C-RAN.

### B. Key Performance Indicator for Network Throughput

To compute an optimal BBU-RRH setting with higher system capacity, maximising network throughput is considered as a second objective in this paper. The practical capacity of user $k$ served by RRH $n$ at in cell $i$ at time $t$ is given as

$$\mathrm{C}_{k_{in}}(t) = \sum_{p \in \mathrm{N}_{\mathrm{RB}}^k} \mathrm{p}_{\mathrm{BW}} . \log_2 \left(1 + a\gamma_{k_{in},p}(\tau)\right) \tag{15}$$

where $p_{BW}$ denotes the bandwidth per PRB (i.e., 180 KHz) and $a$ denotes the bit error rate (BER) and is defined by $a = -1.5/\ln(5\mathrm{BER})$ and BER is set to $10^{-6}$. The overall network throughput at time $t$ can be expressed as

$$\xi(t) = \sum_{m=1}^{\mathrm{M}} \sum_{k=1}^{\mathrm{K}} \mathrm{I}_{m,k}(t) \mathrm{C}_{k_{in},p}(t) \tag{16}$$

where $\mathrm{I}_{m,k}(t)$ notifies if the user $k$ is being served by BBU $m$ at time (t) i.e., $\mathrm{I}_{m,k} = 1$ if user $k$ is served by BBU $m$. Note that, serving a user $k$ at a given sub-frame $\tau$ by BBU $m$ depends on the scheduler employed. This paper considers Proportional Fair (PF) AND Round Robin (RR) scheduling for the optimisation process. Moreover, the overall throughput $\xi$ at time $t$ is normalised before reusing it in the optimisation procedure.

### C. Key Performance Indicator for Handovers

Network transition to a new BBU-RRH configuration may require significantly forced handovers. An increased number of forced handovers in the system is undesirable and leads to performance degradation. Allocating an RRH to a new BBUs at a particular time results in forced handovers of all users associated with the RRH. Since inter-BBU handovers not only involves BBUs but a signalling overhead between the Serving Gateway (S-GW) and Mobility Management Entity (MME), therefore, it is desirable to achieve a new optimum BBU-RRH configuration with a minimum required handovers. A handover index $\mathrm{h}(t)$ is monitored as a third objective for load balancing problem and is given as

$$\mathrm{h}(t) = \frac{1}{2}\left(\frac{\sum_{m=1}^{\mathrm{M}} \sum_{k=1}^{\mathrm{K}} |\mathrm{I}_{m,k}(t) - \mathrm{I}^{\circ}_{m,k}(t)|}{\mathrm{K}}\right) \tag{17}$$

where $\mathrm{I}^{\circ}_{m,k}(t)$ is a binary variable that indicates a user's association in previous BBU-RRH configuration i.e., $\mathrm{I}^{\circ}_{m,k}(t)$=1, if user $k$ is served by BBU $m$ in previous BBU-RRH configuration.

For proper BBU-RRH configuration, a QoS function is needed which is the weighted combination of KPIs defined in (14), (16), and (17). The multiple objectives are combined into a single QoS objective function. This paper represents QoS as the following maximisation problem with constraints:

$$\begin{aligned} \mathrm{Max} \quad & \mathrm{QoS}(t) = \alpha\psi(t) + \beta\xi(t) - (1-\alpha-\beta)\mathrm{h}(t) \\ s.t. \quad & \mathrm{C}_1 : \sum_{k=1}^{\mathrm{K}} \mathrm{I}_{m,k}(t) \mathrm{N}_{\mathrm{RB}}^k \leq \mathrm{RB}, \forall m \in \{1, 2, ...\mathrm{M}\} \\ & \mathrm{C}_2 : \sum_{m=1}^{\mathrm{M}} \mathrm{I}_{m,k}(t) = 1, \forall k \in \{1, 2, ..., \mathrm{K}\} \end{aligned} \tag{18}$$

Both $\alpha$ and $\beta$ are control parameters of the QoS function. The main objective is to maximise the QoS function.

## VII. Cell Differentiation and Integration (CDI) Algorithm

According to the intuitive analysis above, a CDI algorithm is proposed in this section. The CDI algorithm executes proper cell differentiation, and integration of a geographical area and provides a balanced network by identifying correct BBU-RRH configuration. The block diagram of CDI algorithm is shown in Fig. 5. Network information is collected in the first step and analysed for proper cell differentiation and integration. The information includes, user location indicator $\mathbf{u}$, load contributed by each RRH, BBU-RRH mapping vector $\mathbf{r}$, actual network load, predicted load, and the required number

of BBUs. The algorithm seeks to utilise the network resources efficiently by calculating the necessary number of BBUs and RRHs to serve capacity demands at the end of each CDI cycle. Apart from a single BBU required to serve load requirements, proper BBU-RRH configuration is adjusted at the end of optimisation step by comparing the analysed and optimised QoS values. For the optimisation part of the algorithm, a Discrete Particle Swarm Optimisation (DPSO) algorithm is developed as an Evolutionary Algorithm (EA) to solve the BBU-RRH configuration problem and is explained in the next section. The optimisation process continues until the CDI cycle is completed. Note that, the CDI algorithm shown in Fig. 5 is triggered at the beginning of each CDI cycle.

The pseudo-codes for semi-static cell differentiation and integration are given in Algorithm 1 and Algorithm 2, respectively.Both algorithms execute the scaling of RRHs and BBUs with respect to network load distribution. However, an important consideration is the first association of RRHs to the required number of BBUs during cell differentiation and integration. Algorithm 3 and 4 are supporting algorithms for Algorithm 1, and 2, respectively, which covers all possible cases of initial BBU-RRH assignment during differentiation or integration along with cases where the number of BBUs are increased, decreased or remain unchanged. The initial BBU-RRH mapping is important for efficiently utilising the available BBU resources so as to prevent high blocking rate before a proper BBU-RRH mapping is identified in the optimisation step. Therefore, the blocking rate of the network at time $t$ can be measured as

$$\text{Blocking rate} = \left[1 - \frac{\left(\sum_{m=1}^{\mathrm{M}} \sum_{k=1}^{\mathrm{K}} \mathrm{I}_{m,k}(t)\right)}{\mathrm{K}}\right] \times 100 \quad (19)$$

where $\mathrm{I}_{m,k}(t)$ as discussed earlier, is a binary indicator such that $\mathrm{I}_{m,k} = 1$, if user $k$ is served by BBU $m$ at time $t$. Note that, users are served based on the choice of scheduler used by a BBU. Moreover, the amount of resource shortage (or PRB shortage) in the network based on users PRB demand can be estimated as follows

$$\text{Resource Shortage} = \sum_{m=1}^{\mathrm{M}} \max\Big[(\eta_m(t) - 1), 0\Big] \times 100 \quad (20)$$

where $\eta_m(t)$ is the load on BBU $m$ at time $t$ defined in (6). Note that, the CDI algorithm triggers Algorithm 1 and Algorithm 2 sequentially, i.e., Algorithm 2 is triggered immediately after the Algorithm 1 is executed. In the interest of simplicity and understanding, the CDI algorithm is separated into different pseudo-codes.

### A. Discrete Particle Swarm Optimisation (DPSO)

Particle Swarm Optimisation (PSO) is a robust optimisation technique inspired by social behaviour of flocking organisms. PSO method uses *Swarm Intelligence* for solving global optimisation problems [47]. PSO utilises a population (or swarm) of particles, where each particle represents a solution, namely

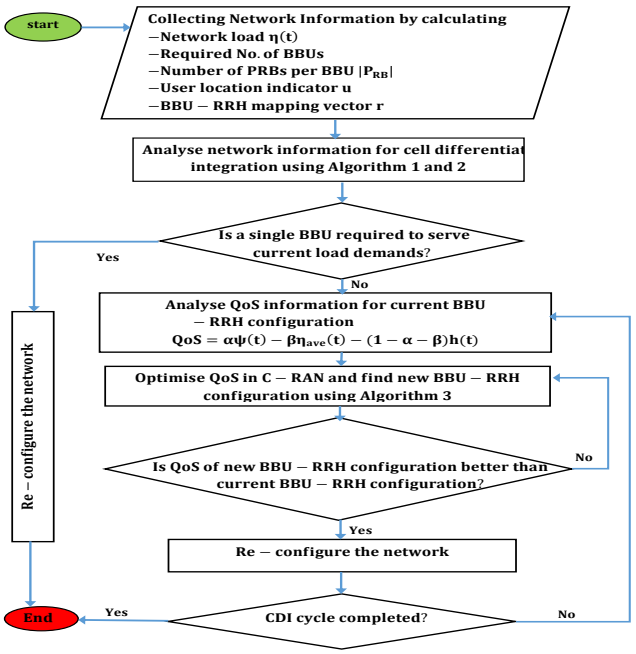



Fig. 5: Block Diagram of CDI Algorithm for one CDI cycle

a BBU-RRH association vector $\mathbf{r}$. As the QoS represented in (18) is considered as the main objective function (or fitness function), PSO seeks to maximise the QoS function by finding the best solution vector $\{\mathrm{r}_{11}, \mathrm{r}_{12}, ..., \mathrm{r}_{in}\}$. PSO operates on a group of particles (or solutions) to probe the solution space in a random way with different velocities. The vector $\{\mathrm{r}_{11}, \mathrm{r}_{12}, ..., \mathrm{r}_{in}\}$ is viewed as the particle position in the n-dimensional solution space while discovering the optimal solution can be viewed as particles probing the solution to search for the optimum position. To direct the particles to their best fitness values, the velocity of each particle is changed stochastically at each iteration. The velocity update of each particle $j$ depends on the historical best position experience (pbest) of the particle itself and the best position experience of neighbouring particles, i.e., the global best position (gbest). Therefore, every particle in the swarm tends to direct itself towards the best solution at each iteration. Since the solution vector $\mathbf{r}$ is real-valued, the standard PSO algorithm can not be applied directly to solve the discrete optimisation problem. Therefore, a Discrete PSO is developed to solve the QoS maximisation problem defined in (18). The DPSO algorithm is described in Fig. 6 and the following steps:

***Step 1:*** Generate initial population $\mathbf{R}^0$ with population/swarm size of $|\Delta|$. Where $\mathbf{R}^0$ consists of N-bit particles (BBU-RRH mapping solutions). Where $\mathbf{N}$ is taken according to the number of active RRHs in the network and the superscript 0 represents the initial iteration number $\mathbf{I} = 0$. The best position of each particle $\mathbf{pbest}_{\mathrm{j}}^0 = \mathbf{r}_{\mathrm{j}}^0, 1 \leq \mathrm{j} \leq |\Delta|$ are initialised with a random velocity of $\mathbf{V}_{\mathrm{j}}^0$ for each particle.

***Step 2:*** Calculate the fitness values for each particle (BBU-RRH mapping solution) in the current swarm/population using the fitness function F defined as QoS in equation (18) and

**Algorithm 1:** Pseudo-code for Semi-static Cell Differentiation with number of BBU requirement

**Input :** Current network load $\eta(t)$ from (7)
Predicted load $\eta(t+\Delta t)$ from (8)
BBU-RRH mapping vector **r**
Required number of BBUs from (12)

**if** *No. of active BBUs =1* **then**
  **if** $\eta(t+\Delta t) \geq |P_{\text{RB}}|$ **then**
    -Activate required No. of BBUs
    -Differentiate cell into tier-2 RRH structure by BBU-RRH mapping using Algorithm 3
    -Update BBU-RRH mapping vector **r**
    **for** *i=1 to C* **do**
      -Select set $S_i$
      -Compute $\eta_{\text{RRH}_{ij}}(t)$ from (13)
      **if** $\eta_{\text{RRH}_{i1}}(t) > |P_{\text{RB}}|$ **then**
        - R$\leftarrow S_i$ {Add $S_i$ to R}
        -Diffentiate cell $C_i$ according to Algorithm 3.
        -Update BBU-RRH mapping vector **r**.
      **end**
    **end**
  **else**
    -No cell differentiation required.
    -Tier-3 RRH structure remains.
  **end**
**else**
  **if** *No. of active BBUs* $\leq$ *No. of required BBUs* **then**
    **if** *All RRHs deployed in the network are active* **then**
      -Activate the required No. of BBUs.
      -Update BBU-RRH mapping vector **r**
    **else**
      -Activate required number of BBUs
      **for** *i= 1 to C* **do**
        -Select set $S_i$
        -Compute $\eta_{\text{RRH}_{i1}}(t)$ from (13)
        **if** $\eta_{\text{RRH}_{i1}}(t) > |P_{\text{RB}}|$ **then**
          - R$\leftarrow S_i$ {Add $S_i$ to List R}
          -Differentiate cell $C_i$ using Algorithm 3
          -Update BBU-RRH mapping vector **r**
        **end**
      **end**
    **end**
  **end**
**end**

identify the global best position achieved i.e., $\mathbf{gbest}^0 = \underset{1\leq j\leq|\Delta|}{\text{argmax}}\, \text{F}(\mathbf{pbest}_j^{\text{I}})$.

***Step 3:*** Update particle j position by updating its velocity. The velocity update equation is given as

$$\mathbf{v}_j^{\text{I}} = w\mathbf{v}_j^{\text{I}-1} + c_1\varepsilon_1\left(\mathbf{pbest}_j^{\text{I}} - \mathbf{x}_j^{\text{I}}\right) + c_2\varepsilon_2(\mathbf{gbest}_j^{\text{I}} - \mathbf{x}_j^{\text{I}}) \quad 1 \leq j \leq |\Delta| \tag{21}$$

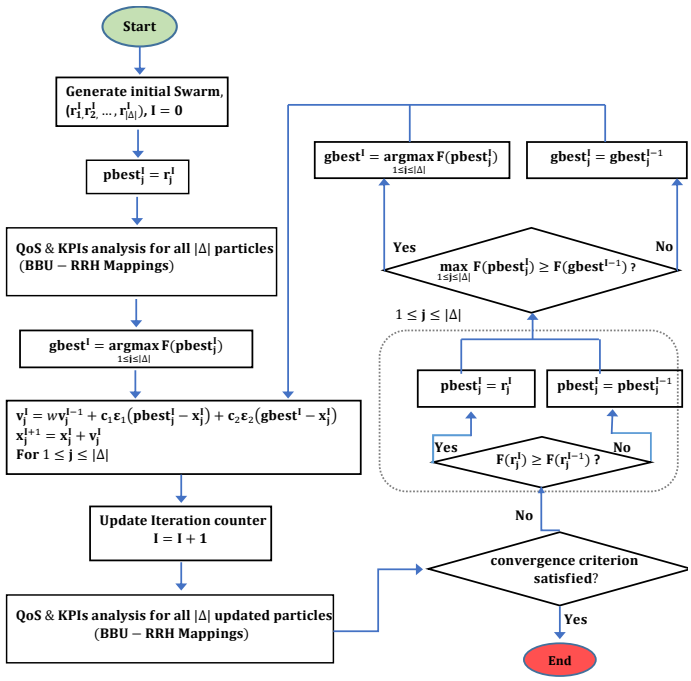



Fig. 6: Block Diagram of DPSO algorithm

where $\mathbf{x}_j^{\text{I}}$ is the current position of particle $j$ in iteration I and $\varepsilon_1, \varepsilon_2$ are random numbers chosen between the range $[0-1]$. Both $c_1$ and $c_2$ are acceleration constants that pulls the particle towards best position. Values in the range 0-5 are chosen for $c_1$ and $c_2$. The inertial weight $w$ represents the effect of preceding velocity on the updated velocity. Larger and smaller value of $w$ are used for global exploration and local search expedition in the search-space, respectively. However, choosing an optimum value for $w$ can assist a balanced proportion between global and local exploration of the search space. Usually values between 0-1 are selected for $w$ [48]. A value of 0.9 for $w$ is selected in this paper. The new position of particle j for the next iteration $\text{I}+1$ will be:

$$\mathbf{x}_j^{\text{I}+1} = \mathbf{x}_j^{\text{I}} + \mathbf{v}_j^{\text{I}} \tag{22}$$

***Step 4:*** Update the iteration counter $(\text{I} = \text{I}+1)$. If the convergence criteria is satisfied then end else go to step 5.

***Step 5:*** Update particle j's personal best position as

$$\mathbf{pbest}_j^{\text{I}} = \begin{cases} \mathbf{pbest}_j^{\text{I}-1} & \text{if} \quad \text{F}(\mathbf{r}_j^{\text{I}}) \leq \text{F}(\mathbf{pbest}^{\text{I}-1}) \\ \mathbf{r}^{\text{I}-1} & \text{if} \quad \text{F}(\mathbf{r}_j^{\text{I}}) > \text{F}(\mathbf{pbest}_j^{\text{I}-1}) \end{cases} \tag{23}$$

***Step 6:*** Update global best position achieved by:

$$\mathbf{gbest}^{\text{I}} = \begin{cases} \underset{1\leq j\leq|\Delta|}{\text{argmax}}\, \text{F}(\mathbf{pbest}_j^{\text{I}}) & \text{if F}(\mathbf{pbest}_j^{\text{I}}) > \text{F}(\mathbf{gbest}^{\text{I}-1}) \\ \mathbf{gbest}^{\text{I}-1} & \text{otherwise} \end{cases} \tag{24}$$

***Step 7:*** Repeat all steps starting from step 1.

## VIII. Computational Results and Analysis

Before going to a more thorough analysis of the proposed CDI concept, the performance of DPSO is tested in BBU-RRH configuration problems.

**Algorithm 2:** Pseudo-code for Semi-static Cell Integration with number of BBU requirement

**Input :** Current network load $\eta(t)$ from (7)
Predicted load $\eta(t+\Delta t)$ from (8)
BBU-RRH mapping vector **r**
Required number of BBUs from (12)

**if** *No. of active BBUs =1* **then**
  -No cell integration required.
  -A high-power BS serves the geographical area.
**else**
  **if** *No.of required BBUs=1* **then**
    -Integrate all cells into tier-3 RRH structure, i.e., a high power BS should serve the geographical area.
    -Switch-off remaining BBUs.
    -Update BBU-RRH mapping vector **r**.
  **else**
    **for** *i=1 to C* **do**
      -Select set $S_i$
      **for** *j=1 to end of* $S_i$ **do**
        -Compute load $\eta_{\mathrm{RRH}_{ij}}(t)$ from (13)
        -Sum=Sum+$\eta_{\mathrm{RRH}_{ij}}(t)$
      **end**
      **if** *Sum* $\leq P_{\mathrm{RB}}$ **then**
        -Integrate all cells by switching-off all RRHs in set $S_i$ except $\mathrm{RRH}_{i1}$.
        -Offload RRHs to required number of BBUs according to Algorithm 4.
        -Update BBU-RRH mapping vector **r**.
      **end**
    **end**
    -Run Algorithm 4 {Case of BBU reduction and no integration}
  **end**
**end**

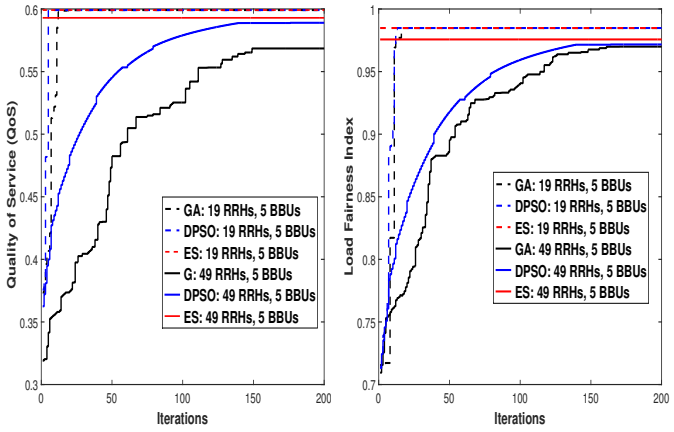
### A. Performance of DPSO algorithm

The performance of DPSO is demonstrated over two different problem scenarios, $P_1$, $P_2$, and compared with the known Genetic algorithm (GA) [49] and ES algorithm. $P_1$ network scenario consists of 5 active BBUs and 19 active RRHs including two differentiated cells (Tier 1, level 2, RRH structure). Whereas, $P_2$ scenario includes 5 active BBUs and 49 active RRHs (Tier 1, level 7, RRH structure). The aim is to analyse the performance DPSO optimisation algorithm for small and large networks. User distribution within each cell is uniform where 6 and 25 users are considered for non-dense and high dense cells, respectively. One fifth of the RRH supported cells are chosen randomly to have high-density users.

For Monte Carlo analysis, the DPSO, GA and ES algorithms are repeated 50 times with 50 different initial BBU-RRH settings for both problem scenarios and the results obtained are averaged. The load fairness index, normalised network throughput, and handover index are represented in Figs 7b, 8b and 8a, respectively, over 200 iterations for both $P_1$ and $P_2$. The optimum values shown in the figures and Table.I, are achieved by exhaustively searching for all possible solutions ($N^M$) using ES algorithm, which helps in demonstrating the improvement in each iteration of the DPSO and GA algorithms. Note that, ES algorithm is independent of the number of iterations.

TABLE I: Computational Results for DPSO, GA and ES

| | | $P_1$ (19 RRH) | $P_2$ (49 RRH) |
|---|---|---|---|
| Quality of Service | DPSO | 0.599142 | 0.588970 |
| | GA | 0.599142 | 0.568473 |
| | ES | 0.599142 | 0.592940 |
| Load Fairness Index | DPSO | 0.984797 | 0.971681 |
| | GA | 0.984797 | 0.969936 |
| | ES | 0.984797 | 0.975680 |
| Network throughput | DPSO | 0.958 | 0.9515 |
| | GA | 0.958 | 0.941499 |
| | ES | 0.958 | 0.95598 |
| Handover Index | DPSO | 0.380957 | 0.387010 |
| | GA | 0.380957 | 0.387021 |
| | ES | 0.380957 | 0.383659 |




(a) (b)

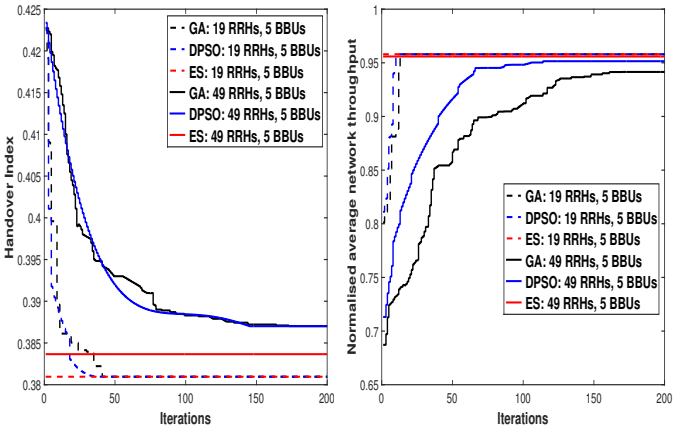
Fig. 7: (a) Quality of Service and (b) Load Fairness Index for DPSO, GA, and ES

Fig. 7a shows that both DPSO GA algorithm converges to the optimum solution in $P_1$ with a Convergence Rate (CR) of 0.970 and 0.8150, respectively. Where CR is defined as the number of times, the DPSO finds a best or optimum solution during the entire number of iterations. This implies that over 200 iterations, the optimum solution is achieved 195 times by DPSO and 163 times by GA for $P_1$. For $P_2$, the CR of DPSO and GA algorithm is 0.2750 and 0.2550, respectively. However, the optimum solution is not achieved by both algorithms over 200 generations. DPSO algorithm achieves the best value 56 times whereas GA achieves the best value 52 times i.e., after 145 iterations and $145\times|\Delta|$ fitness evaluations for DPSO, and after 149 iterations and $149\times|\Delta|$ fitness evaluations for GA. Both DPSO and GA still achieve 99.3012% and 95.8735% of the optimum value found by ES algorithm after an enormous $5^{49}$ ($M^N$) fitness evaluations.

Fig. 7b shows that the DPSO and GA algorithm converge to the optimum load fairness index value after $13^{\mathrm{th}}$ and $16^{\mathrm{th}}$ iterations in $P_1$. However, in $P_2$, the optimum value can not be

found over 200 iterations. The best load fairness index value achieved by DPSO is after 164 iterations and 164×|Δ| fitness evaluations. Whereas 170 iterations and 170×|Δ| fitness evaluations by GA. Which is 99.59% and 99.4113% of the optimum value found by ES algorithm. ES algorithm performs $5^{49}$ fitness evaluations to find the optimum value which is a considerable amount of fitness evaluations.

Figs 8b and 8a displays the convergence of DPSO and GA algorithms to the optimum value for normalised network throughput and handover index, respectively, for both $P_1$ and $P_2$ . In $P_1$, the optimum value for handover index and network throughput is achieved after 38 and 10 iterations by DPSO and after 41 and 13 iterations by GA, respectively. For $P_2$, both DPSO and GA could not find the optimum value over 200 iterations. However, the best possible value achieved by DPSO and GA for network throughput is 99.537% and 98.49361% of the optimum value found by ES algorithm, respectively. The best value achieved for handover index by DPSO and GA are 99.1343% and 99.1315% of the optimum value, respectively. The ES algorithm finds the optimum value after performing $5^{49}$ enormous fitness evaluations whereas the DPSO algorithm performs 114×|Δ| and 145×|Δ| to find the best value for network throughput and handover index, respectively. Note that, the $\alpha$ and $\beta$ control parameters in (18) are selected by performing an exhaustive search (ES) algorithm to identify the optimal BBU-RRH setting for $P_1$. Both $\alpha$ and $\beta$ values are orderly set to 0, 0.1, ..., 1 with a constraint $\alpha+\beta \leq 1$ as shown in Fig. 9. An optimal BBU-RRH setting is found using ES algorithm for each pair of $\alpha$ and $\beta$. It is observed that setting a higher value for load fairness index (until $\alpha = 0.8$) not only reduces the resource shortage but also improves network balance. Setting values for $\alpha > 0.8$ results into improper BBU-RRH mapping which implies that maximising network load balance is overly considered compared to maximising network throughput and minimising handovers, resulting into an increased resource shortage. This paper considers $\alpha = 0.8$ and $\beta = 0.1$ which means assigning a 10% weight to handover minimisation.




(a) (b)

Fig. 8: (a) Average handovers and (b) average network throughput for DPSO, GA and ES.

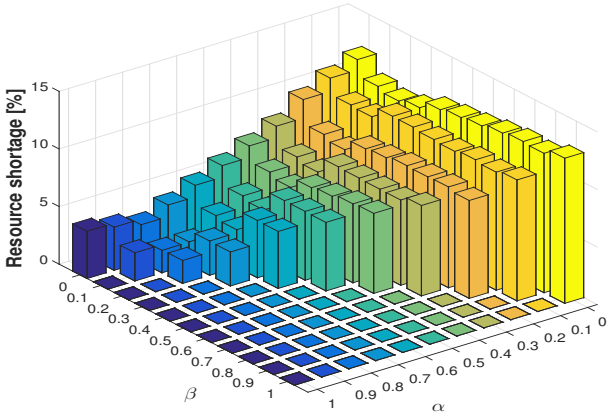



Fig. 9: Resource shortage for different $\alpha$ and $\beta$.

## B. Complexity comparison

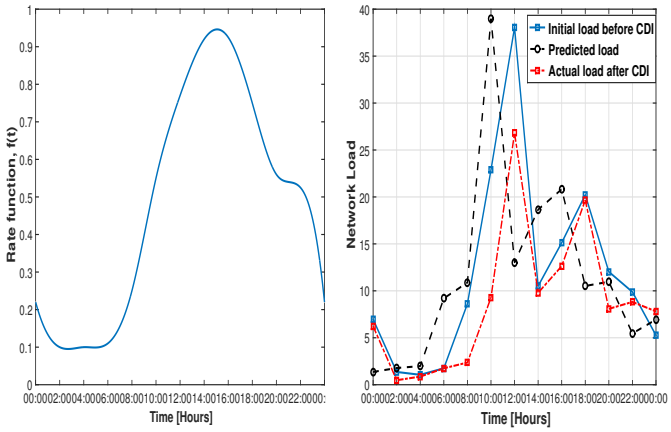
The computation complexity of DPSO and GA algorithm compared to ES algorithm is presented by comparing the number of fitness evaluations carried out by both algorithms. For ES algorithm, the computational complexity is $\boldsymbol{O}\left(|N|^{|M|}\right)$ where **N** and **M** are the number of RRHs and BBUs, respectively. The required fitness evaluations for **N** RRHs and **M** BBUs in the network for ES algorithm is $|N|^{|M|}$. However, the computational complexity of DPSO and GA is $\boldsymbol{O}\left(|\Delta|\mathrm{I}\right)$. The number of fitness evaluations at each iteration of DPSO and GA algorithm depends on the swarm/population size $|\Delta|$. Therefore, the required fitness evaluations for DPSO are $|\Delta\mathrm{I}|$. Note that, the ES algorithm finds the optimum value after evaluating an enormous number of fitness functions which is $|\mathbf{N}|^{|\mathbf{M}|} - |\Delta|\mathrm{I}$ times more than the fitness evaluations of DPSO and GA at the $\mathrm{I}^{\mathrm{th}}$ iteration/generation.

In $P_1$, the optimum BBU-RRH configuration is achieved after $37 \times 2 \times 10^2$ ($\mathrm{I} \times |\Delta|$) fitness evaluations by GA and $5 \times 2 \times 10^2$ ($\mathrm{I} \times |\Delta|$) whereas the ES algorithm finds the optimum BBU-RRH configuration after $5^{19}$ ($|\mathrm{N}|^{|\mathrm{M}|}$) fitness evaluations. The ES algorithm performs $1.9073 \times 10^{16}$ ($|\mathbf{N}|^{|\mathbf{M}|} - |\Delta|\mathrm{I}$) extra fitness function evaluations than the number of fitness evaluations performed at $5^{\mathrm{th}}$ iteration of DPSO and $14.114{\times}10^{16}$ more fitness function evaluations than the number of fitness evaluations performed at $37^{\mathrm{th}}$ generation of GA.

In. $P_2$, the near optimum BBU-RRH configuration is found after $144 \times 2 \times 10^2$ ($\mathrm{I} \times |\Delta|$) fitness evaluations by DPSO and after $148 \times 2 \times 10^2$ ($\mathrm{I} \times |\Delta|$) fitness evaluations by GA. The ES however, performs $5^{49}$ fitness evaluations to find the optimum solution. Note that, the optimum solution found by ES algorithm required $1.7764 \times 10^{34}$ ($|\mathbf{N}|^{|\mathbf{M}|} - |\Delta|\mathrm{I}$) more fitness evaluations than DPSO and GA, which is too enormous.

## C. Performance analysis of CDI algorithm

To make the simulation more realistic, the user arrivals in Fig. 10b follows a Poisson process with rate $\lambda$. However, due to the dynamic spatial and temporal nature of user traffic,




(a) (b)

Fig. 10: (a) Rate function for time in-homogeneous user arrivals and (b) Actual vs Predicted network load with respect to time

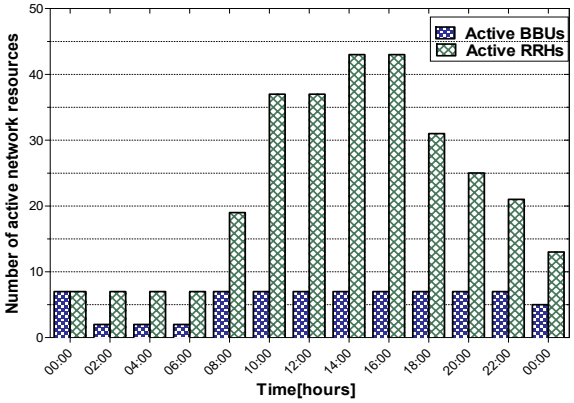



Fig. 11: Number of active BBUs and RRHs with respect to network load/time

the user arrival is modelled as a time-inhomogeneous process. This is achieved by multiplying the time-homogeneous Poisson process with traffic intensity parameter $\lambda$ and the rate function $f(t)$ shown in Fig. 10a. The rate function is unit-less and reshapes the traffic from constant intensity to an analogous time varying profile that reflects typical traffic patterns in a real cellular network. If users arrive in the system following a Poisson process with intensity $\lambda$ users/min, with a constant service time of $h$ (60 sec), then the number of users at time $t$ is calculated as $\mathrm{K}(t) = \chi h f(t)$. Where $\chi \sim \mathrm{Poiss}(\lambda)$ is a random variable with mean $\lambda$ (i.e., $\lambda = 200$). Moreover, different data rate requirements are assumed for end users based on 3GPP standard simulation parameters [50] i.e., 4-25 kbps for audio, 32-384 kbps for video, 28.8 kbps for data, and 60 kbps for real-time gaming services. A reliable prediction of network load is achieved if the collected load samples are 25 or larger [51]. In this paper, network load prediction is performed every $\Delta t = 60\,\mathrm{sec}$. However, the actual network load is sampled every $\breve{\tau} = 2.4\,\mathrm{sec}$. The initial load prediction starts after a 'training period' of 1 minute. This allows the prediction model to have enough first samples to estimate $\hat{\mu}$ and $\hat{\delta}$ for $\Delta\eta$. The CDI algorithm proposed in this paper determines the number of BBUs and RRHs required to handle traffic demands. Fig. 11 shows an actual number of active BBUs and RRHs with respect to time, based on uniform user distribution and network load shown in Fig. 10b. The path- loss models for micro and pico-user at a distance $D$ from transmitter and frequency 2GHz are given as $\mathrm{PL_{mic}(dB)} = 34.53 + 38\log_{10}D$, and $\mathrm{PL_{pico}(dB)} = 140.7 + 36.7\log_{10}D$ [52], [53], respectively. It is considered that the RRHs support micro cells in tier-2 RRH structure.

The RRH-BBU association vector $\mathbf{r} = \{\mathrm{r}_{11}, \mathrm{r}_{12}, ..., \mathrm{r}_{in}\}$ is maintained and updated after each CDI cycle only if the number of BBUs or RRHs are scaled and also if the BBU-RRH mapping is re-configured. Newly activated RRHs and BBUs in the network are mapped according to Algorithm 3 and 4. In this paper, a maximum of 49 RRHs are deployed in the network to support semi-static cell differentiation and integration. The BBU pool consists of 5 BBUs that can be activated and deactivated according to load demand. The initial BBU-RRH mapping at the beginning of a CDI cycle might degrade the network QoS with unnecessarily blocked users. Therefore, dynamic BBU-RRH mapping is proposed as an optimisation problem to identify proper BBU-RRH mapping. The CDI algorithm utilises DPSO to find optimum BBU-RRH configuration to overcome network QoS degradation and minimising unnecessarily blocked users.

For a more thorough analysis, the proposed CDI enabled C-RAN (CDI-CRAN) concept is compared to a fixed C-RAN scenario (F-CRAN). The BBU cloud holds five BBUs in both cases. However, the F-CRAN scenario does not support cell differentiation or integration, and only 7 RRHs serves the entire macrocell coverage area. However, the dynamic BBU-RRH mapping is enabled in the F-CRAN scenario which shows $5^7$ possible BBU-RRH mapping solutions to choose from at the beginning of each CDI cycle. However, the number of possible BBU-RRH mapping solutions for CDI scenario at the start of each CDI cycle is $\mathrm{M^N}$, where M and N represents the number of active BBUs and RRHs, respectively. Moreover, an increasing user arrival is considered in the network with random data rates requirement as explained earlier. However, a Monte-Carlo analysis is performed, where uniformly distributed users are envisaged for each instance, and the average of all distributions are taken into account regarding network load, throughput, blocked users, and resource shortage analysis. Figs 12a and 12b shows the comparative performances regarding average blocked users and average network throughput for F-CRAN and CDI-CRAN with Proportional Fair (PF) and Round Robin (RR) scheduling techniques. The CDI algorithm includes 2 phases, i.e., the cell integration/differentiation phase and the BBU-RRH optimisation phase. The optimisation phase in both F-CRAN and CDI-C-RAN is tested with both GA and DPSO algorithm in the second phase and the results achieved are compared.

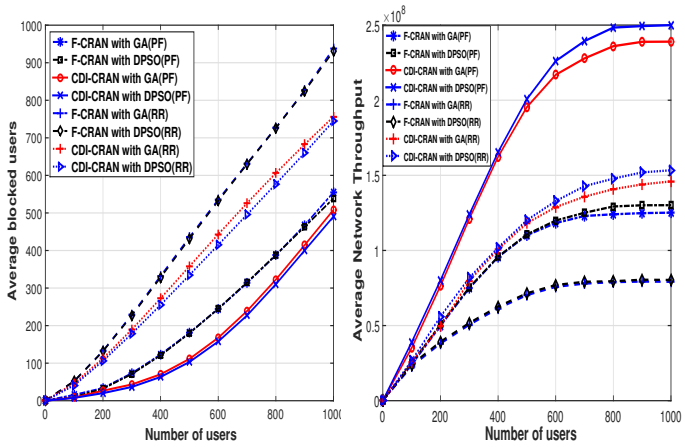
The simulation results demonstrate the advantage of using CDI-CRAN instead of an F-CRAN setting. When an F-

**Algorithm 3:** Initial RRH association to active BBUs during cell differentiation.

**Input :** List A of newly activated BBUs
List R containing sets of RRHs supporting cell differentiation

**if** *A is not empty* **then**
  **for** *m=1 to No. of active BBUs* **do**
    -Compute $\eta_m(t)$ from (6)
    **if** $\eta_m(t) \leq$ *lower limit* **then**
      A$\leftarrow$ BBU$_m$\{Add BBU$_m$ to List A\}
    **end**
  **end**
  I=1;
  **while** *not the end of List R* **do**
    -Select I$^{th}$ set from list R
    $m = 1$;
    **for** *j=1 to end of set* $S_i$ **do**
      **if** $m > |A|$ **then**
        $m = 1$
      **end**
      $BBU_m \leftarrow RRH_{ij}$\{Map $RRH_{ij}$ to $BBU_m$ except $R_{1j}$\}
      $m = m + 1$;
    **end**
    I=I+1;
  **end**
**else**
  **for** *m=1 to No. of active BBUs* **do**
    -Compute $\eta_m(t)$ from (6)
    **if** *lower limit* $\leq \eta_m(t) \leq$ *Upper limit* **then**
      A$\leftarrow$ BBU$_m$\{Add BBU$_m$ to A\}
    **end**
  **end**
  **if** *A is still empty* **then**
    A $\leftarrow$ All active BBUs
  **end**
  -Sort A in increasing order of BBU loads
  I=1;
  **while** *not the end of List R* **do**
    -Select I$^{th}$ set from List R
    $m = 1$;
    **for** *j=1 to end of set* $S_i$ **do**
      **if** $m > |A|$ **then**
        m=1;
      **end**
      BBU$_m \leftarrow$ RRH$_{ij}$\{Map RRH$_{ij}$ to BBU$_m$ except RRH$_{1j}$\}
    **end**
    I=I+1;
  **end**
**end**

CRAN is considered, the average blocked users in the network are much higher with significantly lower average throughput, using any scheduling technique, as shown in Figs 12a and 12b, provided that the dynamic BBU-RRH mapping is enabled




(a) (b)

Fig. 12: (a) Average blocked users and (b) Average network throughput for fixed and CDI-enabled C-RAN.

in both scenarios. However, an interesting observation is the significant drop in the averaged blocked users and the necessary increase in average network throughput in CDI-CRAN compared to F-CRAN. This indicates that during cell differentiation, an overloaded cell divides into multiple smalls cells. This not only reduces the user to RRH distances but also the PRB demands resulting from high SINR. Note that, cell differentiation increases the number of RRH interferers in the network. However, RRHs served by the same BBU does not contribute to the overall interference experienced by users served by the same BBU.

From the results shown in Fig. 12b, it is observed that the average network throughput increases by 44.56% in the CDI-CRAN compared to F-CRAN, both enabled with GA (as an optimisation algorithm) and PF schedulers. Whereas with DPSO algorithm and PF schedulers, an increase of 45.53% is observed. However, with RR schedulers the average network throughput increases by 40.68% and 42.102% in CDI-CRAN with GA and DPSO, respectively. Moreover, the average throughput difference between GA and DPSO algorithm in a CDI-CRAN, with PF and RR scheduling is 4.022% and 4.12%, respectively.

Fig. 12a shows that with PF scheduling in both CDI-CRAN and F-CRAN, about 52.29% and 23.149% reduction in the average number of blocked users is observed in CDI-CRAN hosting GA and DPSO, respectively. However, with RR scheduler, the average blocked users decline seen in CDI-C-RAN compared to F-CRAN is 17.489% and 20.903% for GA and DPSO, respectively. Table.II shows that the average resource shortage drastically decreases in the CDI CRAN, compared to F-CRAN provided that both F-CRAN and CDI-CRAN have an equal amount of available resources (i.e., 5 BBUs, 5$\times$100 PRBs). A 76.4% decrease in average PRB shortage is estimated with CDI-enabled C-RAN compared to fixed C-RAN for both GA and DPSO. Moreover, the overall blocking rate experienced during the course of simulation is also highlighted in Table.II.

TABLE II: Comparison results for fixed and CDI-enabled C-RAN

| | Blocking Rate[%] | | | | Resource Shortage | |
|---|---|---|---|---|---|---|
| | GA | | DPSO | | GA | DPSO |
| | PF | RR | PF | RR | | |
| F-CRAN | 35.99 | 81.66 | 35.34 | 81.10 | $16.44 \times 10^3$ | $16.42 \times 10^3$ |
| CDI-CRAN | 26.87 | 67.33 | 25.809 | 62.13 | $38.79 \times 10^2$ | $38.47 \times 10^2$ |

## IX. CONCLUSION

The concept of cell differentiation and integration in C-RAN is examined with an objective to utilise network resources efficiently without degrading the overall network QoS. A load prediction model is considered for pro-active network scaling of BBUs and RRHs. Network load balance is maintained via proper BBU-RRH mapping and is formulated as a linear constrained integer programming problem. In this regard, a CDI algorithm is developed for C-RAN and tested for comparison with an F-CRAN setting. Computational results show a significant increase in average network throughput and a noticeable decrease in average blocked users and average resource (PRBs) shortage. The CDI algorithm hosts a DPSO algorithm which is developed to find optimum BBU-RRH configuration dynamically. The performance of DPSO is tested and compared to GA and ES. Using two benchmark problems, the DPSO delivered noticeably faster convergence compared to GA and ES, which makes the CDI algorithm more reliable for a self-organised C-RAN.

**Algorithm 4:** Initial RRH association to active BBUs during cell integration.

**Input :** List A of No. of active BBUs
No. of required BBUs
BBU-RRH mapping vector r

**if** *No. of required BBUs* $<$ $|A|$ **then**
  **for** *m=1 to* $|A|$ **do**
    **for** *i=1 to C* **do**
      **for** *j=1 to c* **do**
        -Select $\text{RRH}_{ij}$ from BBU-RRH vector **r**
        **if** $RRH_{ij} = m$ **then**
          $Z_m \leftarrow \text{RRH}_{ij}$
          {$Z_m$ is a List of RRHs handled by $\text{BBU}_m$}
        **end**
      **end**
    **end**
  **end**
  -Sort List A in decreasing order of BBU loads
  **for** *m=1 to end of List A* **do**
    **if** $m \neq |$*No. of required BBUs*$|$ **then**
      $\overline{A} \leftarrow \text{BBU}_m\{\overline{A}$ is a List of required BBUs$\}$
    **else**
      $\overline{B} \leftarrow \text{BBU}_m$
      {$\overline{B}$ is a List of BBUs to be switched off}
    **end**
  **end**
  -Sort List $\overline{A}$ in increasing order of BBU loads
  **for** *i=1 to end of List* $\overline{B}$ **do**
    -Select $i^{th}$ BBU from List $\overline{B}$
    -Select List $Z_m$ of the $i^{th}$ BBU
    m=1;
    **for** *j=1 to end of* $Z_m$ **do**
      **if** $m > |\overline{A}|$ **then**
        m=1;
      **end**
      -Select RRH at $\text{j}^{th}$ index in List $Z_m$
      -Select BBU at $\text{m}^{th}$ index of List $\overline{A}$
      $-\text{BBU}^m \leftarrow \text{RRH}^j$
      m++
    **end**
  **end**
  -Switch off all BBUs in List $\overline{B}$
**end**

## REFERENCES


[1] B. Haberland, F. Derakhshan, H. Grob-Lipski, R. Klotsche, W. Rehm, P. Schefczik, and M. Soellner, "Radio base stations in the cloud," *Bell Labs Technical Journal*, vol. 18, no. 1, pp. 129–152, 2013.

[2] E. U. T. R. A. Network, "Self-configuring and self-optimizing network (SON) use cases and solutions," *Third Generation Partnership Project (3GPP) specification TR 36.902 V9.3.1*, vol. 36, 2011.

[3] F. Ahmed, A. A. Dowhuszko, and O. Tirkkonen, "Network optimization methods for self-organization of future cellular networks: Models and algorithms," *Self-Organized Mobile Communication Technologies and Techniques for Network Optimization*, pp. 35–65, 2016.

[4] O. Østerbø and O. Grøndalen, "Benefits of Self-Organizing Networks (SON) for mobile operators," *Journal of Computer Networks and Communications*, vol. 2012, 2012. [Online]. Available: http://dx.doi.org/10.1155/2012/862527

[5] L. Jorguseski, A. Pais, F. Gunnarsson, A. Centonza, and C. Willcock, "Self-organizing networks in 3GPP: standardization and future trends," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 28–34, December 2014.

[6] I. Chih-Lin, C. Rowell, S. Han, Z. Xu, G. Li, and Z. Pan, "Toward green and soft: a 5G perspective," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 66–73, 2014.

[7] C. Liu, L. Zhang, M. Zhu, J. Wang, L. Cheng, and G.-K. Chang, "A novel multi-service small-cell cloud radio access network for mobile backhaul and computing based on radio-over-fiber technologies," *Journal of Lightwave Technology*, vol. 31, no. 17, pp. 2869–2875, 2013.

[8] J. Wu, Z. Zhang, Y. Hong, and Y. Wen, "Cloud radio access network (C-RAN): a primer," *Network, IEEE*, vol. 29, no. 1, pp. 35–41, Jan 2015.

[9] M. Khan, R. Alhumaima, and H. Al-Raweshidy, "Reducing energy consumption by dynamic resource allocation in C-RAN," in *Networks and Communications (EuCNC), 2015 European Conference on*. IEEE, 2015, pp. 169–174.

[10] T. Zhang, E. van den Berg, J. Chennikara, P. Agrawal, J.-C. Chen, and T. Kodama, "Local predictive resource reservation for handoff in multimedia wireless IP networks," *IEEE Journal on selected areas in Communications*, vol. 19, no. 10, pp. 1931–1941, 2001.

[11] N. Dandanov, H. Al-Shatri, A. Klein, and V. Poulkov, "Dynamic Self-Optimization of the Antenna Tilt for Best Trade-off Between Coverage and Capacity in Mobile Networks," *Wireless Personal Communications*, vol. 92, no. 1, pp. 251–278, 2017.

[12] P. Muoz, R. Barco, and I. de la Bandera, "On the Potential of Handover Parameter Optimization for Self-Organizing Networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 1895–1905, Jun 2013.

[13] J. M. Ruiz-Avils, M. Toril, S. Luna-Ramrez, V. Buenestado, and M. A. Regueira, "Analysis of Limitations of Mobility Load Balancing in a Live LTE System," *IEEE Wireless Communications Letters*, vol. 4, no. 4, pp. 417–420, Aug 2015.

[14] M. Huang and J. Chen, "A conflict avoidance scheme between mobility load balancing and mobility robustness optimization in self-organizing networks," *Wireless Networks*, pp. 1–11, 2016. [Online]. Available:

http://dx.doi.org/10.1007/s11276-016-1331-y
[15] Y. Dhungana and C. Tellambura, "Multichannel Analysis of Cell Range Expansion and Resource Partitioning in Two-Tier Heterogeneous Cellular Networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 3, pp. 2394–2406, March 2016.
[16] O. G. Aliu, A. Imran, M. A. Imran, and B. Evans, "A survey of self organisation in future cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 336–361, 2013.
[17] M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," *computer*, vol. 27, no. 6, pp. 17–26, 1994.
[18] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32 – 49, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2210650211000058
[19] Z. Zhang, K. Long, J. Wang, and F. Dressler, "On Swarm Intelligence Inspired Self-Organized Networking: Its Bionic Mechanisms, Designing Principles and Optimization Approaches," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 513–537, First 2014.
[20] J. Kennedy, "Particle Swarm Optimization," in *Encyclopedia of machine learning*. Springer, 2011, pp. 760–766.
[21] A. Vasilakos, M. P. Saltouros, A. F. Atlassis, and W. Pedrycz, "Optimizing QoS routing in hierarchical ATM networks using computational intelligence techniques," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 33, no. 3, pp. 297–312, Aug 2003.
[22] N. Phan, T. Bui, H. Jiang, P. Li, Z. Pan, and N. Liu, "Coverage optimization of LTE networks based on antenna tilt adjusting considering network load," *China Communications*, vol. 14, no. 5, pp. 48–58, May 2017.
[23] Y. Adediran, H. Lasisi, and O. Okedere, "Interference management techniques in cellular networks: A review," *Cogent Engineering*, vol. 4, no. 1, p. 1294133, 2017.
[24] R. Sachan, T. J. Choi, and C. W. Ahn, "A Genetic Algorithm with Location Intelligence Method for Energy Optimization in 5G Wireless Networks," *Discrete Dynamics in Nature and Society*, vol. 2016, 2016.
[25] L. Liu and W. Yu, "Cross-Layer Design for Downlink Multihop Cloud Radio Access Networks With Network Coding," *IEEE Transactions on Signal Processing*, vol. 65, no. 7, pp. 1728–1740, April 2017.
[26] C. PAN, H. Zhu, N. Gomes, and J. Wang, "Joint Precoding and RRH selection for User-centric Green MIMO C-RAN," *IEEE Transactions on Wireless Communications*, vol. PP, no. 99, pp. 1–1, 2017.
[27] K. Wang, W. Zhou, and S. Mao, "On Joint BBU/RRH Resource Allocation in Heterogeneous Cloud-RANs," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.
[28] Y. S. Chen, W. L. Chiang, and M. C. Shih, "A Dynamic BBU-RRH Mapping Scheme Using Borrow-and-Lend Approach in Cloud Radio Access Networks," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–12, 2017.
[29] K. Sundaresan, M. Y. Arslan, S. Singh, S. Rangarajan, and S. V. Krishnamurthy, "FluidNet: a flexible cloud-based radio access network for small cells," *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 915–928, 2016.
[30] Z. Yu, K. Wang, H. Ji, X. Li, and H. Zhang, "Dynamic resource allocation in TDD-based heterogeneous cloud radio access networks," *China Communications*, vol. 13, no. 6, pp. 1–11, June 2016.
[31] K. Lin, W. Wang, Y. Zhang, and L. Peng, "Green Spectrum Assignment in Secure Cloud Radio Network with Cluster Formation," *IEEE Transactions on Sustainable Computing*, vol. PP, no. 99, pp. 1–1, 2017.
[32] W. He, J. Gong, X. Su, J. Zeng, X. Xu, and L. Xiao, "SDN-Enabled C-RAN? An Intelligent Radio Access Network Architecture," in *New Advances in Information Systems and Technologies*. Springer, 2016, pp. 311–316.
[33] C. Yang, Z. Chen, B. Xia, and J. Wang, "When ICN meets C-RAN for HetNets: an SDN approach," *IEEE Communications Magazine*, vol. 53, no. 11, pp. 118–125, November 2015.
[34] O. Simeone, A. Maeder, M. Peng, O. Sahin, and W. Yu, "Cloud radio access network: Virtualizing wireless access for dense heterogeneous systems," *Journal of Communications and Networks*, vol. 18, no. 2, pp. 135–149, April 2016.
[35] J. G. Herrera and J. F. Botero, "Resource Allocation in NFV: A Comprehensive Survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 518–532, Sept 2016.
[36] R. Cziva, S. Jout, D. Stapleton, F. P. Tso, and D. P. Pezaros, "SDN-Based Virtual Machine Management for Cloud Data Centers," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 212–225, June 2016.
[37] M. Peng, Y. Sun, X. Li, Z. Mao, and C. Wang, "Recent advances in cloud radio access networks: System architectures, key techniques, and open issues," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2282–2308, 2016.
[38] T. Q. Quek, M. Peng, O. Simeone, and W. Yu, *Cloud radio access networks: Principles, technologies, and applications*. Cambridge University Press, 2017.
[39] M. Khan, R. S. Alhumaima, and H. S. Al-Raweshidy, "QoS-Aware Dynamic RRH Allocation in a Self-Optimised Cloud Radio Access Network with RRH Proximity Constraint," *IEEE Transactions on Network and Service Management*, vol. PP, no. 99, pp. 1–1, 2017.
[40] K. Sundaresan, M. Y. Arslan, S. Singh, S. Rangarajan, and S. V. Krishnamurthy, "FluidNet: a flexible cloud-based radio access network for small cells," *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 915–928, 2016.
[41] M. Peng, C. Wang, V. Lau, and H. V. Poor, "Fronthaul-constrained cloud radio access networks: Insights and challenges," *IEEE Wireless Communications*, vol. 22, no. 2, pp. 152–160, 2015.
[42] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud RAN for Mobile Networks; A Technology Overview," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 405–426, Firstquarter 2015.
[43] M. Y. Arslan, K. Sundaresan, and S. Rangarajan, "Software-defined networking in cellular radio access networks: potential and challenges," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 150–156, January 2015.
[44] K. Tsagkaris, G. Poulios, P. Demestichas, A. Tall, Z. Altman, and C. Destré, "An open framework for programmable, self-managed radio access networks," *IEEE Communications Magazine*, vol. 53, no. 7, pp. 154–161, 2015.
[45] M. K. Simon and M.-S. Alouini, *Digital communication over fading channels*. John Wiley & Sons, 2005, vol. 95.
[46] S. Sheikh, R. Wolhuter, and H. A. Engelbrecht, "An Adaptive Congestion Control and Fairness Scheduling Strategy for Wireless Mesh Networks," in *Computational Intelligence, 2015 IEEE Symposium Series on*. IEEE, 2015, pp. 1174–1181.
[47] A. Kaveh, "Particle swarm optimization," in *Advances in Metaheuristic Algorithms for Optimal Design of Structures*. Springer, 2017, pp. 11–43.
[48] J. Vesterstrom and R. Thomsen, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems," in *Evolutionary Computation, 2004. CEC2004. Congress on*, vol. 2. IEEE, 2004, pp. 1980–1987.
[49] M. Srinivas and L. M. Patnaik, "Genetic algorithms: a survey," *Computer*, vol. 27, no. 6, pp. 17–26, June 1994.
[50] 3GPP, "Technical Specification Group Services and System Aspects;Services and service capabilities (Release 13) ," 3rd Generation Partnership Project (3GPP), TS V13.0.0, Dec. 2015.
[51] T. Zhang, E. van den Berg, J. Chennikara, P. Agrawal, J.-C. Chen, and T. Kodama, "Local predictive resource reservation for handoff in multimedia wireless IP networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 1931–1941, Oct 2001.
[52] 3GPP, "Technical Specification Group Radio Access Network;Spatial channel model for Multiple Input Multiple Output (MIMO) simulations (Release 13)," 3rd Generation Partnership Project (3GPP), TR V13.1.0, Dec. 2016.
[53] ——, "Technical Specification Group Radio Access Network;Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancement for E-UTRA physical layer aspects (Release 9) ," 3rd Generation Partnership Project (3GPP), TS V9.1.0, Dec. 2016.