Journal of Network and Systems Management (2022) 30:33
https://doi.org/10.1007/s10922-022-09644-w

ALPACA: A PCI Assignment Algorithm Taking Advantage of Weighted ANR

Ensar Zeljković[1] · Dimitris Gogos[2] · Gerwin Dox[2] · Steven Latré[1] · Johann M. Marquez-Barja[1]

Received: 23 June 2021 / Revised: 19 November 2021 / Accepted: 2 January 2022

Abstract

The growth of mobile data traffic has led to the use of dense and heterogeneous networks with small cells in 4G and 5G. To manage such networks, dynamic and automated solutions for operation and maintenance tasks are needed to reduce human errors, save on Operating expense (OPEX) and optimize network resources. Self Organizing Networks (SON) are a promising tool in achieving this goal, and one of the essential use cases is Physical Cell Identity (PCI) assignment. There are only 504 unique PCIs, which inevitably leads to PCI reuse in dense and heterogeneous networks. This can create PCI collisions and confusions, but also a range of modulo PCI issues. These PCI issues can lead to use cases where User Equipments (UEs) cannot properly identify a cell or cells cannot properly identify UEs, especially during handovers, which all leads to radio communication failure. Therefore, a proper PCI assignment is crucial for network performance. In this paper, we first conduct a study on the impact of different PCI issues on the performance of the network by doing experiments on real-life hardware. Based on the finding from the experiments we create two SON algorithms: a Weighted Automatic Neighbor Relations (ANR) and a PCI assignment algorithm ALPACA. The Weighted ANR creates neighbor relations based on measurements from the network and calculates weights for cells and neighbor relations. ALPACA uses these weights to assign PCI values to cells in a way that avoids PCI issues or at least minimizes their effects on the network. ALPACA works in phases to allow it to adapt to dynamic network topology changes and continuously optimize the network. We validate and evaluate our approach using a simulator package that we have developed. The results show that ALPACA can resolve all collisions and confusion for up to 1000 cells in a highly dense topology, as well as minimize the effects of inevitable modulo PCI issues.

Keywords Dense networks · Heterogeneous networks · Centralized · Weighted ANR · PCI algorithm · Optimization

---

Extended author information available on the last page of the article

1 Introduction

Mobile data traffic has seen an increase of 4000 times in the period between 2007 and 2017, and has grown to an astonishing 30.6 exabytes per month by 2020 [1]. To meet this growing data demand, next generation networks propose the use of highly dense small cell deployments. Both Long Term Evolution (LTE) release 9 and LTE Advanced release 12 propose the use of small cells as a way to increase the network capacity in indoor and outdoor environments, to increase the coverage, reduce cell deployment cost, etc. [2]. Continuing in the same fashion, 5G also uses densely deployed small cells to increase the capacity of the network [3, 4]. Small cells are deployed and operate on the same frequency bands as the macro cells of a network operator to combat spectrum scarcity [5]. Because of the low coverage range, small cells are deployed in a dense manner. This leads to the creation of so called dense and heterogeneous networks. Furthermore, small cells are deployed on an ad-hoc manner [6, 7], which requires them to have plug and play capabilities. All this added complexity creates a need for more automated and dynamic solutions for operation and maintenance tasks that reduce human error and optimize the network [8, 9].

SON is a promising tool that is used to govern a mobile network and enables it to set itself up and manage the network resources to achieve optimum performance [10]. Its main objective is auto-configuration of new cells and small cells, and re-configuration of existing ones in an automated way [11]. SON aims to optimize the networks resources [12], minimize the OPEX costs [13], reduce human intervention and errors, and increase the network capacity [11]. Both 3rd Generation Partnership Project (3GPP) [11] and Next Generation Mobile Network (NGMN) [14] view SON as one of the crucial parts of next generation mobile networks. They have specified multiple SON use cases: (i) self-configuration which covers pre-operational phases of network deployment, such as planning and initial configuration), (ii) self-optimization which optimizes the network during the operational phase and (iii) self-healing for maintenance and recovery from errors [15]. One of the most relevant use cases of SON is the PCI assignment [11]. The idea of an automatic PCI assignment algorithm was first introduced in 3GPP TS 36.300 Release 10 [16], where it was part of the wider Operation and Management (OAM) entity.

The PCI is a low level cell identifier which is used by UEs to identify cells especially during mobility procedures such as handovers [17]. However, there is a limited number of unique PCI values, 504 to be precise [18]. In a dense and heterogeneous networks cells have to reuse PCI values that have already been taken by other cells. As shown on Fig. 1, this can result in PCI issues such as PCI collisions, confusions, and a range of PCI modulo issues , [19, 20]. These PCI issues can lead to use cases where UEs cannot properly identify a cell or cells cannot properly identify UEs, especially during handovers, which can all lead to radio communication failure [21]. Also, PCI issues can influence both the downlink and uplink control channels. This can result in wrong estimation of the channel quality and a wrong selection of the Modulation and Coding Scheme

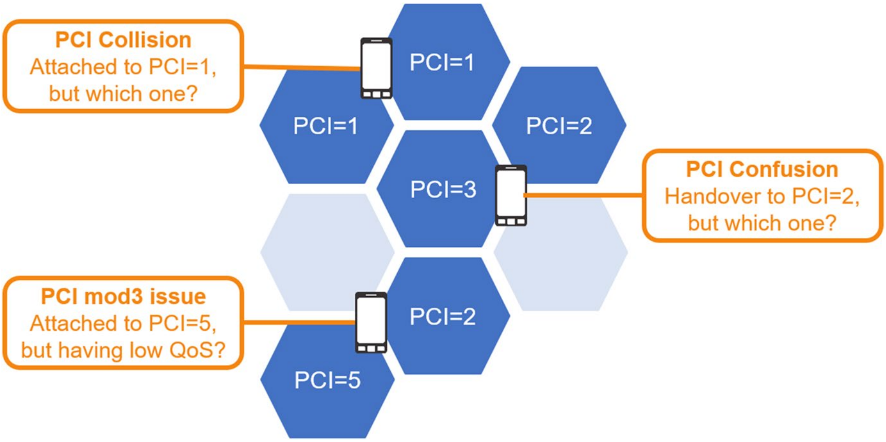

Fig. 1 PCI assignment problems including collisions, confusions and modulo 3 issues

(MCS), which leads to throughput degradation, [22]. Most PCI assignment algorithms focus only on the PCI collisions and confusions [15, 23, 24], not taking into account a range of modulo PCI issues. Some PCI algorithms focus only on the self-configuration aspect [18, 21, 22]. Even though there are PCI algorithms with self-optimisation and self-healing in mind [15, 25, 26], they mostly focus only on PCI collisions and confusions. The most popular metric to use for PCI assignment is the neighbor relations table, however it is mostly used very statically and does not reflect dynamic changes in a dense and heterogeneous network.

In this paper, we utilize dRAX [27], an OpenRAN compliant framework. dRAX is created by Accelleran [28], a company that also provides LTE small cells. dRAX [27] helps us create SON xApps in a centralized location. Using a global network overview and network wide metrics a more detailed neighbor relations table can be created. We add weights to the cells and neighbor relations to reflect dynamic changes happening in the network. To generate these weights, metrics are available such as cell load, number of UEs, UE throughput and their Reference Signal Receive Power (RSRP) and Reference Signal Received Quality (RSRQ) values. Such weights can be used in a PCI assignment algorithm that can resolve not just PCI collisions and confusions, but modulo PCI issues as well.

The contributions of this paper are four-fold. (I) First, we conduct a study on the impact of the different PCI issues on the performance of a network using real-life hardware and experimentation. (II) Based on those findings, we first introduce a Weighted ANR SON xApp on top of dRAX to determine neighbor relations. Weights are assigned to cells and neighbor relations using multiple metrics available in dRAX which have been seen as relevant during the PCI impact study. (III) Taking advantage of the weighted neighbor relations table, we introduce our PCI assignment algorithm AnneaLing Pci AlloCAtor (ALPACA). Using three phases, it adapts to dynamic network topology changes in dense and heterogeneous networks. Its main objective is to assign PCI values to cells in a way that solves PCI collisions and confusion, but also modulo 3,25,30 and 50 PCI issues. (IV) Finally, we create a simulator package with tools that help us simulate and emulate experiments with different network topologies to validate and evaluate our approach and algorithms.

2 Related Work

2.1 Centralized and Distributed Approaches

In this section we analyse the existing research on the topic of PCI assignment by conducting a literature study. A natural starting point is the 3GPP standard. 3GPP's technical specification 36.300 [16] introduces the idea of an automatic PCI assignment procedure which can be executed in two ways: (i) centralized or (ii) distributed. In both cases the PCI assignment procedure uses the OAM entity. According to the centralized approach, the OAM acts as the centralized entity that selects PCI values and communicates them to the cells. This happens during the pre-operational phase. The cell is responsible for then configuring itself with the specified PCI. The 3GPP specification, however, does not give details on the implementation itself.

The distributed approach, as defined by 3GPP [16], also makes use of the OAM entity. However, the OAM this time proposes a list of PCI values to the cell. The cell is then responsible to discard some of those PCI values based on local information gathered either through neighbor relations shared over the X2 interface or by making use of the UE measurements. The cell then randomly selects a PCI value from the remaining list and configures itself with it. The exact implementation details are again not specified.

As can be seen, 3GPP categorizes the PCI assignment into two approaches: (i) centralized or (ii) distributed. Therefore, we continue our literature study by examining these two approaches in other research works. There are a number of researchers that chose the centralized approach, such as [12, 23, 26, 29–33]. Khwandah et al. [26] use a centralized coordinator for configuring the PCI values for small cells. They claim that their approach avoids saturating the wireless medium with signalling information needed in a distributed approach. Bandh et al. [23] were one of the first to use the graph coloring approach for PCI assignment in a centralized manner. They use the OAM as the centralized entity which looks at cells as vertices and their X2 interfaces as edges between those vertices. Liu et al. [29] improve on the graph coloring approach by better re-use of the PCIs by the OAM. Amirijoo et al. [34], on the other hand, utilize the Operation Support System (OSS) as the centralized entity which has a global network overview. This entity gathers UE measurements to detect PCI issues. As can be seen, the centralized approach takes advantage of a centralized entity that has a global overview of the network and can therefore create network wide optimizations. These approaches are also simple to maintain since the centralized entity is responsible for the PCI assignment. However, they also present a single point of failure.

Another group of authors have chosen to use a distributed approach for PCI assignment, such as [24, 35–39]. Liu et al. [36] use a similar approach to the standard 3GPP distributed scheme, improving on the consultation mechanism over the X2 interface. Ahmed et al. [24] use a distributed graph coloring scheme which depends on local knowledge about the 1-hop and 2-hop neighbors of cells. Besides their centralized approach, Amirijoo et al. [37] have also developed a distributed approach in a similar fashion. They use UE measurements to detect PCI issues on the cell level. If a problem is detected, the cell needs to change its PCI and notify all its neighbors about the change. Oppolzer et al. [39] developed a distributed PCI assignment algorithm that uses radio scanning techniques to resolve PCI issues. Distributed approaches, as can be seen, depend on local knowledge of the cell. Even though there is no single point of failure in this case, the local knowledge can be limited and inaccurate, and achieving a global optimum is questionable. Also, the maintenance and management of such solutions become more complex.

2.2 Algorithms Used for PCI Assignment

We now take a more in depth look at the wide variety of algorithms used for PCI assignment. By far the most popular approach in literature has been to use graph coloring algorithms [9, 10, 12, 15, 21, 23, 24, 29, 40, 41]. Bandh et al. [23] use graph coloring theory to solve the PCI assignment problem by assigning each cell a color different from its neighbors. Diab et al. [13] propose a self-organized solution named Stable Graph Coloring (SGC) scheme which uses a temporal PCI sub-range for newly switched on cells. They claim to minimize the neighbor relations inaccuracies and maximize the re-use of PCIs. Abdullah et al. [42] and Panxing et al. [43] have taken a different approach by using genetic algorithms. Gui et al. [22] have used Binary Quadratic Programming for PCI assignment, while Yao et al. [25] have used the tree spinning algorithm. Shahab et al. [44] use neural networks to solve the PCI assignment problem. Zahran et al. [45] make use of the Extended Synchronization Signals (ESS) approach which eliminates the unavoidable PCI confusion problem in dense heterogeneous LTE deployments. Saxena et al. [32] use virtual cells and virtual cell clusters in a randomized algorithm to make a collision and confusion free network. Clustering of cells to determine the PCI values of cells and avoid PCI issue is another popular approach [19, 46]. Babadi et al. [47] use real-valued interference pricing algorithm. Abdulkareem et al. [31] make use of a matrix based graphic dyeing algorithm that improves the utilization rate of PCI values, however, it does not take into account the users Quality of Service (QoS). Mwanje et al. [48] look at same frequency networks. Using a super dense multi-layer technique they reduce the possibility of PCI conflicts and confusions. However, their approach has a relatively long PCI assignment time and is not suitable for real-life implementations. Using the fuzzy hierarchical clustering algorithm is another approach authors have used for the PCI assignment problem. Ning et al. [49] use fuzzy hierarchical clustering to classify cells based on their load. Cells with a higher load are assigned their PCI first. Similarly, Tu et al. [50] use dynamic fuzzy clustering to give higher priority of PCI assignment to cells with a higher load. Amirijoo et al. [37] use the local adjustment algorithm, where each cell adjusts their PCI based on the UE measurements. Wu et al. [51] use a multi-objective optimization model for PCI assignment with different constraints. Yang et al. [52] use a similar heuristic approach. Even with all the different algorithms proposed, network operators mostly use two

PCI assignment methods: (i) network roll-out planning and (ii) random selection. The first approach is unsustainable and can lead to an expensive and time consuming process. The second approach, on the other hand, can lead to unsolved PCI issues [53].

2.3 Metrics Used in PCI Assignment Algorithms

Beside the algorithm itself, it is also of interest to look at the different metrics used in such algorithm for PCI assignment. The majority of researchers focus on using the neighbor relations information as the main metric of the PCI assignment algorithm [15, 30, 48]. A convenient way to share the neighbor relations information is using the X2 interface, which a number of authors exploit [23, 24, 36, 39, 48]. The X2 interface is used to share the neighbor relations of small cells, where each small cell has local information about its neighbors only. Other authors [12, 13, 19, 54], have also taken a more enhanced approach where 2-hop or even 3-hop neighbors are shared among cells. Another approach used in literature for the PCI assignment algorithms is to use the similarity of cells as the metric. Ning et al. [49] for example use the angle cosine method to detect the similarities between cells based on which the algorithm can assign PCI values. Tu et al. [50] take a similar approach, but use the Euclidean distance method for analysing cell similarities. However, in their approach Tu et al. [50] use the number of users on a cell to calculate the similarities, which is often not a valid indicator. Xiao et al. [3] try to predict the cell status information and thereby predict handovers to specific cells. These cells then get a PCI from a reserved range to aid in the handover process. Abdullah et al. [53] on the other hand use subscriber information to get the location information of each cell. The PCI values are then assigned based on cell location. Liu et al. [29] use the Global Positioning System (GPS) information of cells as the metric in the PCI assignment algorithm, however this can be challenging for indoor small cells. Authors in [23, 26] also take this approach. Gui et al. [22] on the other hand take advantage of detecting electromagnetic interference based on which the PCI assignment algorithm works. Other authors such as Oppolzer et al. [39] and Wu et al. [55] use radio scanning to detect neighboring cells, however these approaches require the hardware to have that capability. Lee et al. have in [54, 56] tried to categorize cells and use the type of cell as the metric in the PCI assignment algorithm. They reserve a range of PCI values for each type to reduce the interference.

2.4 Summary

As can be seen, the PCI assignment problem has been explored by using different approaches, algorithms and metrics. However, a number of challenges can be seen which we now summarize.

(1) It can be seen that the PCI assignment algorithm falls into one of two categories: centralized or distributed. The main disadvantage of the distributed approach is that the algorithm received only local information, and cannot do global opti-

mizations. We therefore choose the path of a centralized entity that has a global overview of the network and can do global optimizations. This also adds the advantage of supporting heterogeneous networks like authors in [40, 45, 57], however most authors do not consider heterogeneous networks [23, 34, 37, 52, 56]. Specifically, we use an OpenRAN compliant dRAX Radio access network Intelligent Controller (RIC) [27] as the centralized controller entity. The dRAX RIC is capable of controlling the configuration of the cells and gathers a range of metrics from the network. A detailed description of the dRAX RIC is given in Sect. 3.

(2) The main challenge of a PCI assignment algorithm is to resolve the PCI issues. Most authors focus only on the PCI collisions and confusions [10, 13, 15, 23–25, 25, 29, 32, 34, 39, 41, 44, 48, 50, 53]. However, there are other PCI issues that can affect the performance of the network. Authors in [10, 18, 21, 22, 33] also look at another PCI issue, specifically the Reference Signal collision, which is also known as the PCI modulo 3 issue. Shen et al. [58] on the other hand only look at the modulo 3 issue, while Wu et al. [51] consider collisions, confusions, modulo 3 and modulo 30 PCI issue. In our approach we look at collisions and confusions, but also the modulo 3, modulo 25, modulo 30 and modulo 50 PCI issues. Detailed explanation of the different PCI issues and a study on the impact of different PCI issues on the performance of the network using real-life hardware and experiments are present in Sect. 4.

(3) An important aspect of 4G and 5G networks is SON [11], specifically self-configuration and self-optimization. However, some authors focus only on the self-configuration aspect [18, 21, 22, 44]. Even then, some authors do not take into account adding a new cell into the network in their algorithms [34]. Even though some authors do take this into account [13, 53], they do not consider removing a PCI from the network in an appropriate way. Other authors take into account also the self-optimization aspect [15, 25, 26, 32]. However, the self-optimizations are restricted to resolving collisions and confusions when new cells are added. The PCI modulo issues are not considered in the self-optimization phases. Our approach focuses on both aspects of SON. In the self-configuration phase we make sure that new cells are configured properly without collisions and confusions, while the self-optimizations phase continuously optimises the PCI assignments taking into account all the mentioned PCI issues. In certain cases, the PCI modulo issue cannot be avoided, therefore we optimize the network in a way that will minimize the impact of the modulo PCI issues. Our approach also takes into account the scenario of removing a cell from the network.

(4) As for the metrics used in the algorithms, authors in [39, 55] use radio scanning, however the hardware needs to support such functions. Others used GPS [23, 26], which is not always available. Most authors use the neighbor relations as their metric in the PCI assignment algorithms. However, they need the X2 interface to be present to be able to share such information [24, 39, 48]. Even then, these neighbor relations are very static, as in the cells just report who their neighbors are. There is no information about the load of the cells or the signal strength of the UEs towards the cells. This information becomes very useful in scenarios where the modulo PCI issues are unavoidable. Therefore, we develop

a Weighted ANR algorithm which takes multiple metrics to generate weights both for cells and their neighbor relations. Details are available in Sect. 5.

(5) Finally, we take advantage of the weighted ANR and input it into our PCI assignment algorithm ALPACA which uses simulated annealing [59] to assign PCI values to cells in a way that resolves collisions and confusions, and also minimizes the impact of the PCI modulo issues. We explain the details in Sect. 6, after which in Sect. 7 we show the results of its validation and evaluation. We discuss those results in Sect. 8.

3 dRAX Framework

OpenRAN aims to open the Radio Access Network (RAN) by creating a separation between hardware and software, and by creating open interfaces between the RAN components. On top of that, the RIC is used to update and control the network. The OpenRAN architecture is based on standards defined by the O-RAN Alliance, which are fully supporting and complimentary to standards promoted by 3GPP and other industry standards organizations. The O-RAN Alliance consists of almost 30 operator and 200 vendors.

dRAX is an OpenRAN [60] compliant framework. dRAX embodies a family of products that provide a multi-vendor, disaggregated and virtualised RAN Intelligent Control Plane, that includes a near real-time RIC, the RAN central unit, a Service Management and Orchestration (SMO) layer, as well as different xApps. Network virtualization can bring several benefits such as reducing network provisioning time, greater operational efficiency, placing workload independently of the physical topology, etc. dRAX is a Cloud Native solution that is managed and orchestrated through Kubernetes using Docker containers. The dRAX architecture is shown on Fig. 2. As can be seen, dRAX supports the Control User Plane Separation paradigm. First, there is a separation between the control and user plane. Second, the layer 3 of the stack has been moved out of the small cell itself and into the centralized dRAX framework. The layer 3 is containerized into a docker image and is orchestrated by Kubernetes. The dRAX RIC uses a distributed, microservice oriented architecture and serves, among other things, as a platform for creating so called xApps. xApps are additional modules that bring new functionality and services to dRAX. To do so, they can take advantage of a number of endpoints that the RIC exposes. One of them can be used to create, edit and delete the configuration of the small cells. This communications uses the NetConf protocol [61]. The configuration consists of a number of parameters, one of which is the PCI. It should be noted that changing the PCI requires a reboot of the small cell, which is taken into account in our ALPACA algorithm. On the other hand, the RIC also contains a databus, where different metrics from the network are aggregated. Among the many metrics exposed, we were interested in: (i) The measurements reports from UEs that contain the RSRP and RSRQ measurements from each UE to its serving cell, as well as to all the neighboring cells it can detect, and (ii) the throughput of the UE towards its serving cell. To be able to gather such metrics, we configure the small cells to report periodic measurements.

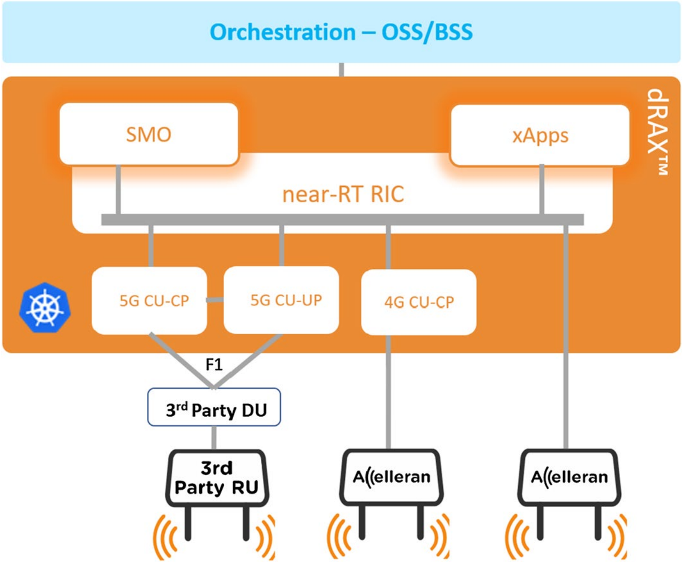

Fig. 2 Accelleran dRAX Framework showing the disagregated RAN layers with the control plane on the RIC, the dRAX Databus and the xApps

We exploit the dRAX RIC and create two SON xApps: (i) the Weighted ANR SON xApp and (ii) the ALPACA PCI SON xApp. The Weighted ANR xApp makes use of the different metrics available in the dRAX RIC databus, specifically the RSRP and RSRQ measurements, and the throughput report. By using the UEs measurements reports from its serving and neighbor cells, we have constructed an algorithm that generates the automatic neighbor relations. Using the values of the RSRP and RSRQ, we added weights to those relations, while using the UE throughput we were able to add weights to the small cells. This way, using a centralized approach we have a global overview of the network. On the other hand, by having the ability to create and edit the configuration of the small cells, we created an xApp that holds the ALPACA algorithm for PCI assignment and optimization. The weighted ANR information is shared on the dRAX RIC databus and then used by the ALPACA PCI xApp to create global optimizations of the PCI.

4 Study on PCI Issues Impact

In this section we conduct our own experimentation on real-life hardware to determine the effects of PCI issues on the network. We first introduce the PCI and explain why it is important. We then follow by describing the experimental setup and results of the study.

4.1 Why is PCI Important

The PCI is a low level cell signature [17]. It is used by the UEs during mobility procedures, such as handovers and cell re-selections, to identify the cells [20]. The PCI is made up of two signals [18]: (i) the Primary Synchronization Signal (PSS) and (ii) the Secondary Synchronization Signal (SSS). The PSS has three orthogonal sequences, while the SSS has 168, which leads to a total of 504 possible PCI values. Due to this limited number of available PCI values, in a dense and heterogeneous network this inevitably leads to PCI reuse [18, 31].

In mobile networks such as LTE, Orthogonal frequency-division multiple access (OFDMA) is used as a technique to eliminate intra-cell interference including inter symbol interference and inter code interference [22]. However, using the same frequency inevitably leads to inter-cell interference which significantly impacts the Signal-Interference-plus-to-Noise-Ratio (SINR) of UEs. Proper PCI assignment has been proved as one the most effective solutions for inter-cell interference [52]. This is because the PCI defines the positions of the Resource Element Groups (REGs) in the bandwidth. REGs are important because the location of the Physical Control Format Indicator Channel (PCFICH) is distributed equally among the four REGs in the frequency domain. The PCFICH is the channel that tells the UE the length of the control channel, so it is important to protect it against interference [26]. On the other hand, the PCI assignment serves as a self-coordination mechanism in the Physical Downlink Control Channel (PDCCH), the most important physical control channel [52]. PCI is also used by UEs to detect a cell during mobility procedures such as handovers [19].

The two most important issues with PCI assignment are [19, 32]: (i) PCI collisions and (ii) PCI confusions. They are shown on Fig. 3. A PCI collision occurs when two neighboring cells have the same PCI values. This can create problems during a handover, or when the UE scans for neighboring cells due to a low signal strength to the serving cell. In both cases, since its serving cell and a neighboring cell have the same PCI, the UE will not be able to distinguish between them and its connection may fail. This can lead to mayor disruptions in communication. The PCI confusion happens when one cell has two neighboring cells that have the same PCI value. Complications arise during handovers. When a handover is about to happen,

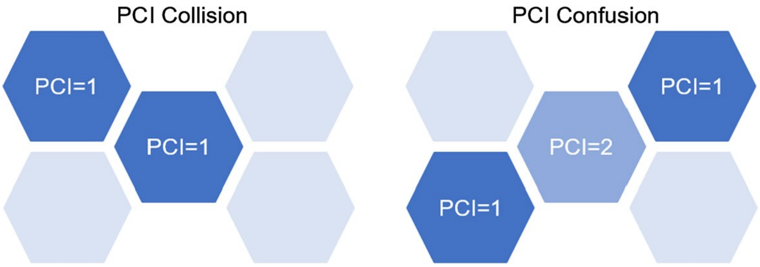

Fig. 3 Examples of the PCI collision (left) when two cells that are neighbors have the same PCI value and confusion (right) when two neighboring cells of a cell have the same PCI value

the cell checks its neighbor relations tables. In case there are two neighboring cells with the same PCI, the handover fails as it is not possible to distinguish to which cell the UE is suppose to be handed over to. Proper PCI assignment should eliminate PCI collisions and confusions. Failing to do so can create an issue where the UE cannot properly identify cells and radio communication fails. Also, improper PCI assignment can influence the SINR reported by the UEs, which in turn influence the MCS and so affects the throughput in the PDCCH [21].

PCI assignment also influences the uplink because it defines the demodulation reference signals which are used in a cell for channel estimation and coherent demodulation in the Physical Uplink Shared Channel (PUSCH) and Physical Uplink Control Channel (PUCCH). It can happen that two UEs on different cells use the same demodulation reference signals which will in turn degrade the decoding process in PUSCH or PUCCH. This happens with every 30th PCI and is often called the PCI modulo 30 issue. This issue may lead to the cell not being able to identify the UE properly [20]. Also, an issue can arise in the downlink where improper PCI assignment can degrade the SINR estimates and lead to wrong estimates of the Channel Quality Indicator (CQI), which can delay the transmissions in the downlink [22]. This superposition of the reference signals is whats called the PCI modulo 3 issue, because the two neighboring cells have PCI that have the same result of the modulo 3 operation [62].

Besides the modulo 3 PCI issue, there are also the modulo 25 and modulo 50 PCI issues. They are related to the location of the PCFICH channel which is determined by the PCI. Every 50th PCI will have the same PCFICH for a 20 (MHz) channel, while it is every 25th PCI for the 10 (MHz) channel. Having two neighboring cells with the same location of the PCFICH can lead to decoding failures and a high BLock Error Rate (BLER).

4.2 Experimental Setup

The setup consists of two small cells that are connected to the dRAX RIC. Using the RIC configuration endpoint, we are able to configure the small cells using NetConf. The two small cells are Time Division Duplex (TDD) units that operate on LTE band 43 with the E-UTRA Absolute Radio Frequency Channel Number (EARFCN) set to 44590 and the bandwidth set to 20 (MHz). The reference signal power is $-$ 10 (dBm). As the UE we use a Raspberry PI which is wired to the two small cells through a programmable attenuator. This gives us the ability to play around with the attenuation towards the two small cells. We use the programmable attenuator such that the UE sees the two cells with a similar signal strength. This way the UE sees the two small cells as neighbors. On the UE we use an iperf3 [63] client to generate 20 (Mbps) of downlink traffic and 10($Mbps$) of uplink traffic through one of the small cells. The experimental setup can be seen on Fig. 4.

We then create four experimental scenarios: (i) one where there are no PCI issues, (ii) the second where we intentionally create a modulo 3 PCI issue by configuring one small cell to have PCI equal to 2 and the other small cell to have PCI equal to 5; (iii) the third scenario where we intentionally create a modulo 30 PCI issue

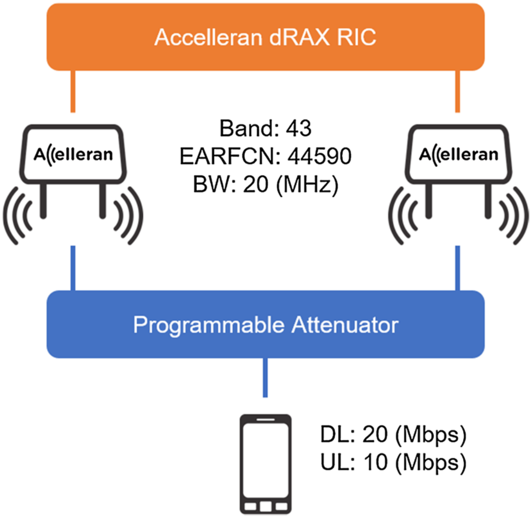

Fig. 4 Experimental setup showing 2 cells connected to dRAX and a UE connected to the two cells via a programmable attenuator

by configuring one small cell to use PCI 2 and the other one to use PCI 32, (iv) the fourth scenario where by creating a modulo 3 PCI issue, we analyse the impact of the RSRP and RSRQ values to neighboring cells on the QoS parameters of the UE. We run each experiment for 2 hours and using the RIC we monitor different Key Performance Indicators (KPIs) such as: (i) the downlink and uplink throughput, (ii) the wideband CQI value, (iii) the downlink and uplink BLER and iv) the RSRP and RSRQ values. Note that the BLER is the BLER reported after Hybrid Automatic Repeat Request (HARQ) in %.

4.3 Results

Table 1 compares the KPIs in the first and second scenario, that is without and with the PCI modulo 3 issue. After running the experiments for 2 h, for each KPI we analyse the average, median, standard deviation, percentile 90 and percentiles $99.\{999, 9999, 99999\}\%$ which is interesting for Ultra-Reliable Low Latency Communications (URLLC). We can see that on average the downlink throughput is around 49.14% higher when there is no PCI modulo3 issue compared to when there is. Looking at the 90th percentile its 57.15% higher. A significant difference can be seen with the wideband CQI value. Looking at the $90th$ percentile, the wideband CQI is 200% higher without the modulo 3 issue compared to the scenario with the PCI modulo 3 issue. The downlink BLER is also heavily degraded when the 2 small cells have a modulo 3 PCI issue. On average we get a 99.87% lower downlink BLER when the two small cells do not have a modulo 3 issue. It is worth noting that the actual numbers are 0.01% and 7.8% of downlink BLER without and with the modulo 3 PCI issue. Finally, the signal quality indicator RSRQ has been monitored. Looking at the $90th$ percentile of the experiment, we get a 17.65% higher RSRQ when there is no modulo 3 PCI issue.

Table 1 Statistics for KPIs in the downlink such as throughput, CQI, BLER and RSRQ with and without the modulo 3 PCI issue

KPI	DL Throughput (Mbps)		CQI		DL BLER (%)		RSRQ	
Experiment	No PCI issue	PCI mod 3 issue	No PCI issue	PCI mod 3 issue	No PCI issue	PCI mod 3 issue	No PCI issue	PCI mod 3 issue
Average	22.46	15.06	14.5	4.9	0.01	7.8	17.7	15.5
Median	22.42	15.10	15	5	0	7.6	18	16
Percentiles								
90%	23.73	15.10	15	5	0.1	13.2	20	17
99.999%	26.46	15.10	15	6	0.1	13.2	21	18
99.9999%	26.46	15.10	15	6	0,1	13.2	21	18
99.99999%	26.46	15.10	15	6	0.1	13.2	21	18
Standard deviation	1.74	0.71	0.6	0.3	0.02	4.21	2.4	1.5

Table 2 compares the KPIs in the uplink. We compared the scenario without any PCI issue to the one with the modulo 3 issue and to the one with the modulo 30 PCI issue. Since the modulo 30 PCI issue affects the uplink, we monitor the uplink KPIs. We can see that on average the uplink throughput is 30.77% and 21.43% higher when there is no PCI issue compared to when there is a modulo 3 and modulo 30 PCI issue respectively. Similar to the downlink, the wideband CQI value is also 200% higher when there are no PCI issues, compared to both the modulo 3 and 30 PCI issue scenarios. Uplink BLER is also heavily degraded when there are modulo 3 and modulo 30 PCI issues. On average, the uplink BLER is 99.87% and 99.81% lower when there are no modulo 3 and 30 PCI issues. We can also see higher BLER when looking at the $99.\{999, 9999, 99999\}\%$ percentiles. Furthermore, we show the Cumulative Distribution Function (CDF) of the uplink throughput on Fig. 5 and CQI in Fig. 6 in scenarios without PCI issues, with the mod 3 PCI issue and with the mod 30 PCI issue.

Figure 7 shows the results of the fourth experiment. Here we test how the signal strength, specifically RSRP and RSRQ, from neighboring cells impact the KPIs of the UE while there is a modulo 3 PCI issue. The figure shows the throughput of the UE, as well as the average throughput and wideband CQI to the serving cell for 3 different parts of the experimental scenario. Also, it shows the average RSRP and RSRQ values to the neighboring cell. In part 1 of the scenario we can see that the RSRP and RSRQ values to the neighboring cell are low, 44 and 7 respectively. With such conditions, the average throughput of the UE was around 33.50 (Mbps) and the wideband CQI was 8. In part 2 of the scenario, using the programmable attenuator, the RSRP and RSRQ values towards the neighboring cells are increased, while keeping the same RSRP and RSRQ values to the serving cell. We keep the RSRP and RSRQ to the serving cell approximately the same so that the radio conditions to the serving cell are equal in part 1 and 2. This way if the throughput suffers, we know its not because the radio conditions to the serving cell have degraded. In part 2, we can see that the RSRP and RSRQ values to the neighboring cell are now higher than in part 1 of the scenario, 52 and 19 respectively. However, due to this increase in the RSRP and RSRQ to the neighboring cell, the throughput suffers and goes down to around 10.43 (Mbps) on average. In part 3 of the scenario, we increase the signal strength to the serving cell, while keeping the same conditions for the neighboring cell. We can see that the RSRP to the neighboring cell did not change, but the RSRQ has decreased. This is because the signal strength to the serving cell is higher, creating more interference in the air. We can see, that because the RSRQ to the neighboring cell decreased, the throughput recovered slightly to around 14.90 (Mbps) on average. The wideband CQI was around the same value in part 2 and part 3.

By conducting this study, we prove that improper PCI assignment has a negative impact on the performance of the network. Both the uplink and downlink throughput get degraded, as well as the uplink and downlink BLER. We also saw that the reported CQI values get degraded when there is a PCI modulo issue. This confirms the conclusions of Sect. 4.1, where we conducted a literature study on the impact of PCI issues on the network. Adding to this the complete communication failure which are the result of PCI collisions and confusions, and the decoding failures due

Table 2 Statistics for KPIs in the uplink such as throughput, CQI and BLER in scenarios without PCI issue, with modulo 3 and with modulo 30 PCI issues

KPI	UL Throughput (Mbps)			CQI			UL BLER (%)		
Experiment	No PCI issue	PCI mod 3 issue	PCI mod 30 issue	No PCI issue	PCI mod 3 issue	PCI mod 30 issue	No PCI issue	PCI mod 3 issue	PCI mod 30 issue
Average	10.2	7.8	8.4	13.2	4.7	4.7	0.01	7.8	5.4
Median	10.2	7.8	8.3	14	5	5	0	7.6	5.1
Percentile									
90%	10.2	8.8	9.3	15	5	5	0,1	13.2	9,5
99.999%	10.2	9.8	10	15	6	6	0,1	22.2	19.1
99.9999%	10.2	9.8	10	15	6	6	0,1	22.2	19
99.99999%	10.2	9.8	10	15	6	6	0,1	22.2	19
Standard deviation	0.24	0.72	0.72	2.34	0.50	0.51	0.02	4.21	3.33

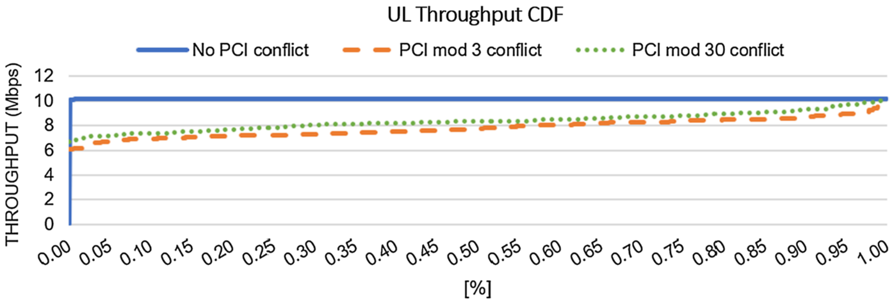

Fig. 5 CDF of the uplink throughput in scenarios without PCI issues, and with mod3 and mod 30 PCI issues

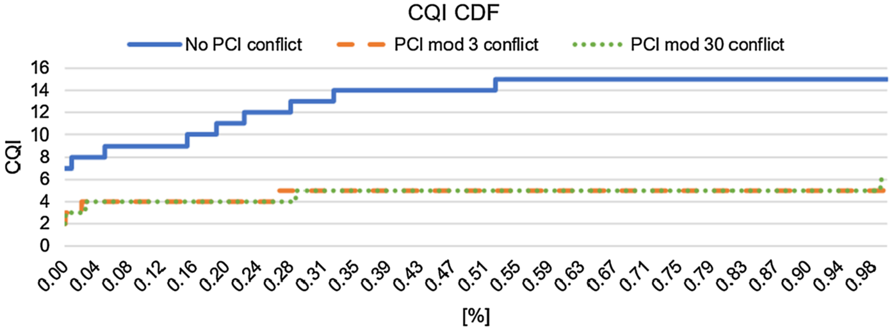

Fig. 6 CDF of the CQI in scenarios without PCI issues, and with mod3 and mod 30 PCI issues

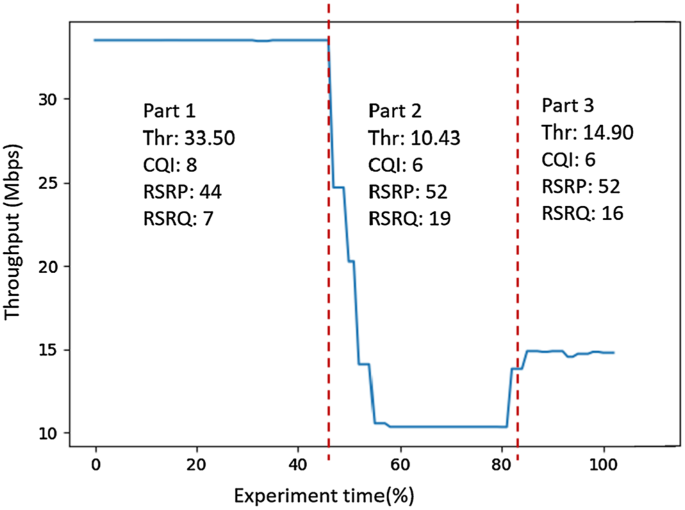

Fig. 7 How RSRQ and RSRP to the neighboring cell impacts throughput when there is a modulo 3 PCI issue, red dotted lines indicate when a new experiment started

to modulo 25 and 50 PCI issues, it becomes apparent why proper PCI assignment is of crucial importance. We also saw that the higher the RSRP and RSRQ are to a neighboring cell, the greater the impact this has on the QoS of the UE. In a dense and heterogeneous network, where small cells are added on an ad-hoc manner, this becomes and even greater challenge, as a more dynamic and automated solutions in needed. In such networks, some PCI issues, especially modulo 3, are sometimes unavoidable. Therefore, a solution that can at least minimize its effects on the network is necessary.

5 Weighted ANR Algorithm

Taking a look at the different PCI issues, its obvious that the most relevant metric is the cell neighbor relations table. However, it is often used in distributed approaches with local knowledge only. Most neighbor relation tables are also static. In most use cases this is more than enough, however, for the PCI assignment the neighbor relations could be supplemented with additional information such as cell load and how many UEs detect a neighbor relation. This is especially relevant for the PCI modulo issues, specifically the modulo 3 issue. In a scenario where there are more than three cells that are neighbors, the modulo 3 PCI issue is inevitable. In these cases the only thing that can be done is to minimize its effects. To do so, just knowing which cells are neighbors is not enough. Also, creating and updating the neighbor relations should be done in an automated way.

Therefore, we create the Weighted ANR SON xApp on top of the dRAX RIC. The objective of this SON xApp is to create and continuously update the neighbor relations table by gathering the available metrics in the dRAX RIC. Because the table is made using an algorithm in a centralized entity, it falls into the category of ANR. The neighbor relations are determined based on the UE measurement reports. Each small cell is configured to ask for periodic measurements from the UEs. These measurement reports contain RSRP and RSRQ values between the UE and its serving cell, as well as between the UE and all the cells it can detect. Because the UE was able to retrieve the RSRP and RSRQ values from those cells, it can be concluded that its serving cell and the reported cells are neighbors. Therefore, we create an algorithm that gathers the UE measurement reports from all of the network and generate a neighbor relations table.

Details on the Weighted ANR are given in Algorithm 1. Before the algorithm starts, the Weighted ANR SON xApp continuously gathers metrics from the dRAX RIC. This enables it to create a global overview of the network with information about the UE measurement reports, cells and neighbor relations. At the beginning of the Weigthed ANR algorithm, it takes a snapshot of the current network state, N. We then reset all the weights in the Weighted ANR table without deleting the existing cell relations. This is because the physical location of cells do not change, hence the neighbor relations do not change. What does change are the weights depending on the movement of the UEs and their usage of the network in terms of throughput. Next, we look at the serving and neighbor cell relation based on the UE measurement report as mentioned above, (s_cell, n_cell). If a UE measurement report comes in with a new cell relation, it is added to the Weighted ANR table. Next we calculate the weights of cells and cell relations. First for each cell, we calculate the cell load ,$cell_{load}$, as the sum of throughputs of all UEs that cell is serving. We then want to combine the information about each cells load and each cell number of UEs its serving into the cell weight. We therefore use the Weighted Sum Model Multi Decision Criteria Making (MCDM) technique [64]. For each cell we calculate its weight, $cell_{weight}$, as the result of MCDM with the following parameters: i) the number of UEs served by the cell, $cell_{num\_of\_ues}$, and ii) the load of the cell, $cell_{load}$.

Algorithm 1 Weighted ANR

```
N
resetWeights()
for $(s\_cell, n\_cell) \in ue\_measurement\_reports$ do
    if $(s\_cell, n\_cell) \notin WeightedANR$ then
        $(s\_cell, n\_cell) \rightarrow WeightedANR$
    end if
end for
for $cell \in cells$ do
    $cell_{load} = calculateCellLoad(cell)$
    $cell_{weight} = MCDM(cell_{load}, cell_{num\_of\_ues})$
end for
for $R \in cell\_relations$ do
    find UE with MAX RSRP to neighbor cell in R
    find UE with MAX RSRQ to neighbor cell in R
    $R_{weight} = MCDM(R_{num\_of\_ues}, RSRP_{max}, RSRQ_{max})$
end for
$sleep(wanr\_interval)$
```

Finally, we examine the cell relations. To explain the algorithm logic here, we use an example shown on Fig. 8. Each cell relation is defined by the UEs that have reported seeing both of the cells in a cell relation. On Fig. 8 we can see that UE 1 only sees Cell A. UEs 2,3 and 4 can see both Cell A and Cell B. This means the relation between cells A and B is defined by those three UEs. Cell A is the serving cell for UEs 2 and 3, while Cell B is the serving cell for UE 4. This is shown with full lines. Each UE also has dotted lines towards the neighbor cells it detected. Looking at all the UEs that are part of this cell relation, we first start by finding the one UE that has the highest reported RSRP towards the neighbor cell. Based on the

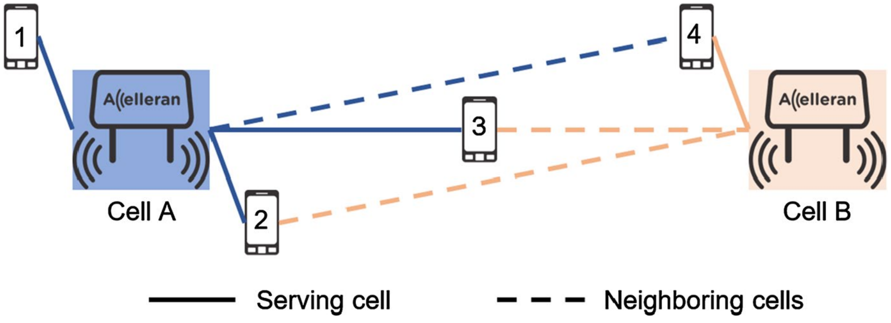

Fig. 8 Example of RSRQ to neighboring cell impact on calculating the weight of a neighbor relation

figure, lets assume this is UE 3. It should have the highest reported RSRP towards Cell B as the neighboring cell it detected. We also find the UE that has the highest RSRQ value towards the neighbor cell. The motivation for finding these UEs, and the respective RSRP and RSRQ values, is based on our conclusion from the study we conducted on the impact of PCI issues. We saw that the higher the RSRP and RSRQ values to the neighboring cell, the greater the impact this has on the QoS parameters of the UEs on those cells. In the context of PCI, this means that the higher the RSRP and RSRQ, the more related these cells are. That means if there is a PCI issue between the two cells, especially a modulo PCI issue, it can greatly impact the performance of those cells. Finding a relation with the lowest RSRP and RSRQ values means if the modulo 3 PCI issue was moved there, it would have less impact on the QoS parameters.

We then again combine multiple metrics to form the weight of the cell relations. For each cell relation, R, we calculate its weight R_{weight} using MCDM with the following parameters: (i) the RSRP value of the UE with the highest RSRP value to the neighbor cell in the relation, $RSRP_{max}$, (ii) the RSRQ value of the UE with the highest RSRQ value to the neighbor cell in the relation, $RSRQ_{max}$, and (iii) the number of UEs that see the cell relation $R_{num\_of\_ues}$.

The Weighted ANR SON xApp is configured with a periodic interval *wanr_interval* which controls how often the Weighted ANR table is calculated. Each time it is calculated, the table is published on the dRAX RIC so other xApps can take advantage of this information. The Weighted ANR xApp also has a control mechanism that is used for removing a cell from the table. As gathering UE measurements to create the neighbor relations depends on the UEs being there and reporting, each detected neighbor relation is stored indefinitely. However, we foresee situations where the network operator would physically remove a cell. In this case, the Weighted ANR SON xApp can be notified, and the specific cell can be removed from the Weighted ANR table.

6 The ALPACA Algorithm

The ALPACA algorithm has three main objectives: (i) to avoid PCI collisions and confusions, (ii) to avoid or at least minimize the effects of the modulo PCI issues, and (iii) to minimize the need for cell reboots on PCI change. In this section we introduce the PCI xApp where ALPACA is running and its different phases. We then focus on the algorithm itself.

6.1 The PCI xApp

The PCI SON xApp was developed on top of the dRAX RIC and is shown on Fig. 9. The input to the xApp comes from the dRAX Databus. Specifically, the Data Processor unit of the PCI xApp looks for the Weighted ANR messages that the Weighted ANR xApp periodically publishes, and the “cell_announce” message which is sent out by a cell when it powers on. The “cell_announce” message announces the cell

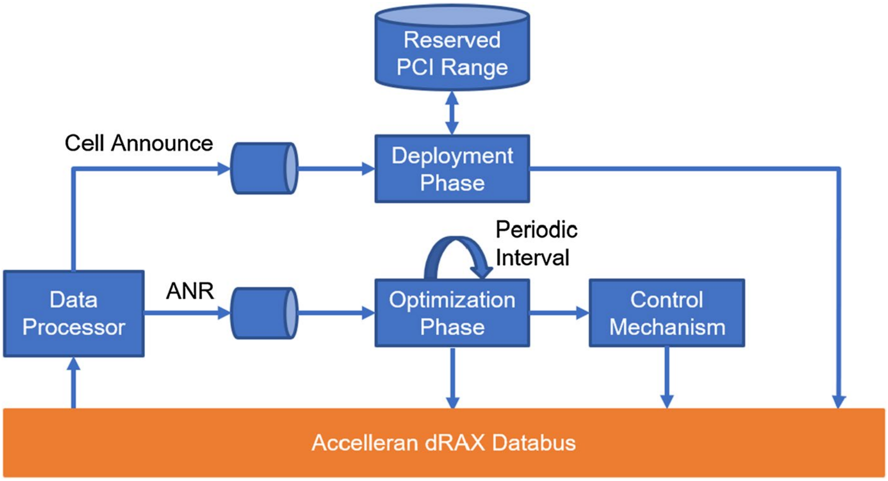

Fig. 9 PCI xApp Architecture showing data monitoring from the dRAX Databus, the deployment phase using the reserved PCI range and optimisation phase using Weighed ANR

and asks dRAX to configure it. The PCI xApp also has a number of configuration options, shown in Table 3 which we will define as we explain the working of the xApp. The PCI xApp consists of three phases: (i) deployment phase, (ii) monitoring phase and (iii) optimisation phase.

6.1.1 Deployment Phase

The deployment phase is essentially the self-configuration phase of SON and is used when a cell first announces itself and has no previous configuration, including a configured PCI. Therefore the cell sends a “cell_announce” message which is picked up by the PCI SON xApp. When this happens, the optimisation phase of the PCI SON xAPP is disabled until deployment is completed. When a new cell is deployed, we assume that we do not have any information about it, its location, its neighbors, etc. That means assigning a PCI value to it is a difficult problem. Therefore, we construct a so called reserved PCI range. The PCI values in this range are only used for newly deployed cells in the deployment phase and cannot be given out in the optimisation phase. One of the configuration parameters that a user is free to change is the size of this reserved PCI range. Based on that size, we can also know how many new

Table 3 PCI xApp configuration options

Name	Description
PCI Assignment Config	Controls how the PCI xApp sends commands to set the PCI value on cells
Size of reserved PCI values	How many reserved PCI values are used
Execution interval	How often is the PCI algorithm executed

cells can be brought into the network at the same time. In the deployment phase, the PCI xApp picks a random PCI value from the reserved range, however keeping track that no two newly deployed cells have the same PCI value. This way the PCI xApp makes sure that the newly deployed cells will not cause confusions and collisions, which are the most impactful PCI issues. This is because the other cells in the network cannot have a PCI that is from the reserved PCI range.

6.1.2 Monitoring Phase

Next, before we can activate the optimisation phase, we need to have information about the neighbors of the newly deployed cells. We get that information from the weighted ANR SON xApp. Therefore, we first have to let the weighted ANR xApp gather enough metrics to build the weighted ANR table. Since deploying a new cell requires physically installing a cell, we assume that the deployment technician can also connect to the network in the area near the newly deployed cell and by doing so start generating measurement reports. Once enough measurement reports are gathered, the technician can disable the deployment phase which will activate the optimisation phase.

6.1.3 Optimisation Phase

The optimisation phase corresponds to the self-optimisation phase of SON. Once the optimisation phase is activated, the PCI SON xApp starts to read the weighted ANR table that it receives from the weighted ANR SON xApp. The neighbor information, as well as the weights that are present in the weighted ANR table, are passed as input to the ALPACA algorithm. The output of the ALPACA algorithm is the assignment of PCIs for the cells in the network. This PCI assignment information is passed on to the Control mechanism. This control mechanism decides which cells get to be reconfigured with the new PCI value and when. The reason it controls which cells are because of the deployment phase. The newly deployed cells coming from the deployment phase will immediately receive their new PCI values. This will also free up the reserved PCI value they were using in the deployment phase, so that the deployment phase can use it on a new cell. After the deployment of a new cell is completed, the optimisation phase runs periodically. The execution interval is controlled by the Execution time configuration option from Table 3. The execution interval is set to 5 seconds, which was determined through empirical comparison of several different values. The algorithm also checks how much time has passed and adapts the “temperature” of the simulated annealing. While it is easier to run in a confined time frame, it makes it more difficult to predict the outcome, especially since older CPU’s or busy workstations might output worse results. Hence, as part of future work we plan to investigate how the execution interval impacts performance. Because changing the PCI value involves a reset of the cell radio, this can be a disruptive operation. However, in some cases, the network operator may want to do it either way. Because of this, the control mechanism has three options for when a PCI change should happen: (i) Automatic PCI assignment— Whenever the ALPACA algorithm finds a better PCI assignment, the cells are reconfigured,

(ii) Scheduled PCI assignment—When the ALPACA algorithm finds a better PCI assignment, the network operator may choose to schedule the PCI change at a later time, potentially during network low usage. (iii) Manual PCI assignment—The network operator manually chooses when the PCI change is going to happen.

6.2 The ALPACA Network Model

First, we model the network as a graph. Each cell forms a node, and two nodes are connected with an edge if a UE can see both of them. Taking advantage of the weights for cells and neighbor relations that are provided by the Weighted ANR xApp, we create a cost function C:

$$C = \sum_{n \in N} w_n \cdot c_n + \sum_{e \in E} w_e \cdot c_e \tag{1}$$

In Eq. 1, n is a node from the set of all nodes N, while e is an edge from the set of all edges E. The weights of nodes and edges are represented as w_n and w_e and they are received from the Weigthed ANR table. The cost functions of the nodes and edges are $c_n(P_n, P_c)$ and $c_e(P_a, P_b)$. This function C, which will be expanded in the next sections, describes the full cost (if positive) or benefit (if negative) of switching to a new network configuration, and is thus the function which we will need to minimize. We summarize the notations in Table 4.

6.3 Algorithm Cost Analysis

The cost of a node depends if a PCI change is required for that node. In that case, the cell needs to reboot disrupting all the UEs currently attached to that cell. Therefore, the node cost function can be represented as:

$$c_n(P_n, P_c) = \begin{cases} 0 & P_n = P_c \\ 1 & P_n \neq P_c \end{cases} \tag{2}$$

In Eq. 2, P_n and P_c are the new and current PCIs respectively. If the proposed new PCI is the same as the current one, then the cost is 0. On the other hand, if it is not

Table 4 Notation summary

Notation	Description
$n \in N$	n is a node from the set of all nodes N
$e \in E$	e is an edge from the set of all edges E
(N, E)	The network graph consisting of nodes and edges
w	The weight
P	The PCI value
p	Function from nodes N to PCIs P
c	The cost function
$d \in D$	d is a PCI mod issue from the set of all PCI mod issues D

then the cost function is 1 because a cell reboot would be required which disrupts the network.

The cost of an edge $c_e(P_a, P_b)$, which is an edge between 2 nodes $a \in N$ and $b \in N$, is defined based on the different PCI issues. First, the cost function avoids collisions when $P_a = P_b$, and confusions when $P_a = P_c \mid\mid P_b = P_c$ where c is a neighboring node to a or b. Next, it also looks at the different modulo PCI issues $d \in D$ where $D = \{3, 25, 30, 50\}$. So the cost functions becomes:

$$c_e(P_a, P_b) = \sum_{d \in D, c \in N : (c,a) \in E \vee (c,b) \in E} \begin{cases} 1 & P_a = P_b \vee P_b = P_c \\ 1 & P_a \equiv P_b \mod d \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Extending onto Eq. 1 and using (N, E) as the graph and p as a function from nodes N to PCIs P, where p_0 is the function from a node to its current PCI, we can rewrite it as:

$$\begin{aligned} C'(N, E, p, p_0) &= C_n(N, E, p, p_0) + C'_e(N, E, p) \\ &= \sum_{n \in N} w_n c_n\big(p(n), p_0(n)\big) \\ &\quad + \sum_{(n_1, n_2) \in E} w_{(n_1, n_2)} c_e\big(p(n_1), p(n_2)\big) \end{aligned} \tag{4}$$

Our job is now to find, given N, E and p_0, a p for which the given cost is as low as possible. Since minimizing the cost is invariant under adding a fixed constant, we can define $C_e(N, E, p, p_0) = C'_e(N, E, p) - C'_e(N, E, p_0)$ so that:

$$C(N, E, p, p_0) = C_n(N, E, p, p_0) + C_e(N, E, p, p_0) \tag{5}$$

We can deduce a few interesting equations:

$$C_n(N, E, p_a, p_a) = 0 \tag{6}$$

$$C_n(N, E, p_a, p_b) = C_n(N, E, p_b, p_a) \tag{7}$$

$$C_n(N, E, p_a, p_c) \leq C_n(N, E, p_a, p_b) + C_n(N, E, p_b, p_c) \tag{8}$$

In other words, $C_n(N, E)$ forms a metric space.

At the same time:

$$C_e(N, E, p_a, p_a) = 0 \tag{9}$$

$$C_e(N, E, p_a, p_b) = -C_e(N, E, p_b, p_a) \tag{10}$$

$$C_e(N, E, p_a, p_c) = C_e(N, E, p_a, p_b) + C_e(N, E, p_b, p_c) \tag{11}$$

Therefore $C_e(N, E)$ forms a potential space. As a result:

$$C(N,E,p_a,p_b) + C(N,E,p_b,p_a) = 2C_n(N,E,p_a,p_b) \tag{12}$$

$$\geq 0 \tag{13}$$

In Eq. 13 the equality only occurs when $c_n(p_a,p_b) = 0$, which only happens when $p_a = p_b$, therefore as long as no new information is given the algorithm will never prefer a state it has previously discarded. The proof for this is shown in Eq. 14. If $C(p_a,p_0) < C(p_b,p_0)$ which means the cost to move from the current PCI to PCI a is lower than the move to b, then $C(p_b,p_a)$ ends up to be larger than 0. This means the algorithm will not run into a situation where it goes from PCI a to PCI b, and then afterwards realize PCI a became better.

$$\begin{aligned}
&\text{If } C(p_a,p_0) < C(p_b,p_0),\\
&C(p_b,p_a) = C_e(p_b,p_a) + C_n(p_b,p_a)\\
&\quad = C_e(p_b,p_0) + C_e(p_0,p_a) + C_n(p_b,p_a)\\
&\quad = C_e(p_b,p_0) + C_e(p_0,p_a) + C_n(p_b,p_a)\\
&\quad + C_n(p_0,p_a) - C_n(p_a,p_0)\\
&\quad > C_e(p_b,p_0) - C_e(p_a,p_0) + C_n(p_b,p_0) - C_n(p_a,p_0)\\
&\quad = C(p_b,p_0) - C(p_a,p_0)\\
&\quad > 0
\end{aligned} \tag{14}$$

6.4 Optimizing the Cost

Trying to figure out which configuration is optimal is complicated. In the special case where changing a cell PCI is free and every conflict except mod 3 is unimportant, a perfect scoring solution exists when it is possible to color the given graph with 3 colors. The same applies for mod 25 conflicts and 25 colors, and so on. The only case where the problem would be P is when changing the cell PCI would be the only case to incur a penalty, in which case the problem is trivial. Therefore, comparing two possible solutions is much easier, and in fact can be used to find the minimum. Most optimization processes, ranging from simulated annealing and hill climbing to genetic algorithms, rely on being able to compare two possible solutions as well, so this operation is critical in order to go forward.

The difference between two solutions equals:

$$\begin{aligned}
&C(N,E,p_a,p_0) - C(N,E,p_b,p_0)\\
&\quad = C_n(N,E,p_a,p_0) - C_n(N,E,p_b,p_0)\\
&\qquad + C_e(N,E,p_a,p_0) - C_e(N,E,p_b,p_0)\\
&\quad = \sum_{n\in N} w_n\big(c_n\big(p_a(n),p_0(n)\big) - c_n\big(p_b(n),p_0(n)\big)\big)\\
&\qquad + \sum_{(n_1,n_2)\in E} w_{(n_1,n_2)}\\
&\qquad \big(c_e\big(p_a(n_1),p_0(n_2)\big) - c_e\big(p_b(n_1),p_0(n_2)\big)\big)
\end{aligned} \tag{15}$$

In this equation, it is clear that if $p_a(n) = p_b(n)$, then the corresponding entry in the first sum reduces to zero. Similarly, if $p_a(n_1) = p_b(n_1)$ and $p_a(n_2) = p_b(n_2)$, then the corresponding entry in the second sum reduces to zero as well. This means that in order to calculate the difference between two similar solutions we only have to take into account the nodes that have been changed and the edges connected to them, reducing the number of computations the ALPACA algorithm needs to do to $O(N)$.

6.5 Simulated Annealing

After reducing the problem statement to a simple optimization problem, we use simulated annealing [59] to solve it. In essence, simulated annealing mirrors the physical process of slowly cooling down a material (usually a metal) so that the atoms will form crystalline structures: slowly reducing the energy of the metal gives the atoms the time to arrange themselves into an optimal pattern. By replicating this behaviour on our model we hope to achieve a global minimum instead of a local minimum for our cost function. Simulated annealing has a few requirements and advantages.

6.5.1 Requirements

We have already defined our states (the PCI assignments) and our cost function C. Simulated annealing however, like hill climbing, also requires those states to be connected in a graph-like structure so that it can search through them. This requires us to define a neighbour relationship between them. We decided on the following transformations to map a state to its neighbour:

- Replace the PCI of a cell with an unused PCI
- Swap the PCIs of two cells

These transformations have a few useful properties.

First, they are cheap to compute. The cost function of the entire state is equal to the sum of partial costs of every cell, and all partial costs only depend on the PCI of the cell and its neighbour. For computing the cost function of a neighbouring state we only need to touch one or two cells and their neighbours instead of all cells present, greatly improving efficiency.

Second, simulated annealing benefits when the diameter of the state graph is small. This is actually the case, since for a network with N cells the diameter we can transition from one state to another in at most N steps.

6.5.2 Advantages

Simulated annealing is an improvement over hill climbing in that it will not get stuck in a local minimum. With enough resources it will always find the global maximum. Of course "enough resources" is a vague term and for large networks we won't have enough computing time at hands to find the global minimum, but we can get close.

Compared to other techniques, like tabu search or even genetic algorithms, simulated annealing has another advantage: it only actually keeps one state in memory at all times. Other methods need to keep track of which states have already been passed to e.g. forbid a previously passed state (tabu search) or join two states (genetic algorithms). Verifying this, no matter the method, takes at least $O(N + E)$ time since the full graph needs to be known, and $O(N + E)$ space per state if not resorting to bloom filters or similar. On the other hand, simulated annealing only needs to know the difference between the scores of two states, and since the two states considered only have one or two differing nodes, only $O(d)$ time is needed with d the degree of the concerned nodes. On networks with a large number of cells only having to check and modify one or two of them per iteration instead of all of them can lead to a massive improvement in performance.

7 Experimentation and Results

7.1 Experimental Setup

To be able to experiment with PCI assignments on a large scale, we have developed a simulator package that can simulate a network and feed the real Weighted ANR and PCI SON xApps with information. The simulator package consists of two python tools: (i) the topology generator and (ii) the topology simulator. The topology generator is a python, command line tool used to create any type of network topology. From the command line, the user can: (i) add a cell by specifying its ID, (ii) delete a cell by providing its ID, (iii) add a UE, which will get an automatic ID, and specify its throughput, serving cell ID, a list of neighboring cell IDs, and what the RSRP and RSRQ values are to those cells (iv) delete a UE by providing its auto generated ID, (v) save the topology in a .topo file (vi) load a .topo file for editing, (vii) list the UEs and for each one provide the serving cell and a list of neighboring cells, (viii) visualize the topology by creating a network topology graph.

The topology simulator tool is used to load the .topo file, including the information on cells and UEs. The tool then enables us to simulate turning on or off a cell. When a cell is turned on, a *cell_announce* message is sent on the dRAX Databus. This will trigger the deployment phase in the PCI xApp. The PCI xApp will then send the temporary PCI value, which the simulator will pick up and use as the cells PCI. Once the PCI is received, the simulator brings the cell on air. Once a cell is on air, any UEs found in the topology file that are attached to that cell are then turned on by the simulator. This means that the simulator starts generating and sending measurement reports of those UEs, including the throughput report, RSRP and RSRQ measurements to the serving and any on air neighboring cell. These reports are then picked up on the dRAX Databus by the Weighted ANR xApp and the Weighted ANR table is generated. In this way, we are able to replicate a real-life network, and validate and evaluate the PCI assignment of the ALPACA algorithm.

7.2 Scenarios

Two major scenarios are devised: (i) small scale validation scenario and (ii) large scale evaluation scenarios. The small scale validation scenario is used to validate the ALPACA algorithm by creating a number of specific topologies to showcase how PCI issues are avoided or their effect is minimized. Once we validate that the ALPACA algorithm does resolve PCI issues, we move on to evaluating the algorithm by testing it out on a larger scale with 250, 500, 1000 and 2000 cells scenarios. In these scenarios, we compare the PCI assignments of different PCI algorithms and the baseline network topology by evaluating the number of PCIs issue present in the network topology.

7.3 Data and Data Preparation

7.3.1 Validation scenario

To validate the ALPACA PCI assignment algorithm, we use our simulator package to create a number of specific use cases. Using the topology generator we generate topologies shown in Fig. 10. In topology A, we create 4 new cells that are turned on in the network. For each cell we describe its ID, current PCI value and the weight calculated by the Weighted ANR xApp. Here we want to show how the deployment phase of the PCI xApp works. We reserve 20 PCI values for the temporary PCI range [484–504]. Topology B in Fig. 10 shows the neighbor relations, depicted as lines between cell, based on the UEs in that topology once the cells come on air. The numbers on the lines represent the weights of the relations. Here we demonstrate how the ALPACA algorithm will assign PCIs without causing PCI issues. Topology C introduces new UEs in the network, which make all 4 cells neighbors to each other. We want to show here how the modulo 3 PCI issue is inevitable and motivate why ALPACA chose to put the PCI issue where it did. To show how the optimisation phase of ALPACA works and the modulo 3 impact is moved to minimize its effects we shuffle the UEs in the network to generate topology D. In topology E we then create two new cell, isolated from the previous four. We also show the neighbor relations of those 2 cells in topology F. Finally, to show a use case in which at some point new neighbor relations are discovered that combine the two islands of cells, we create topology G.

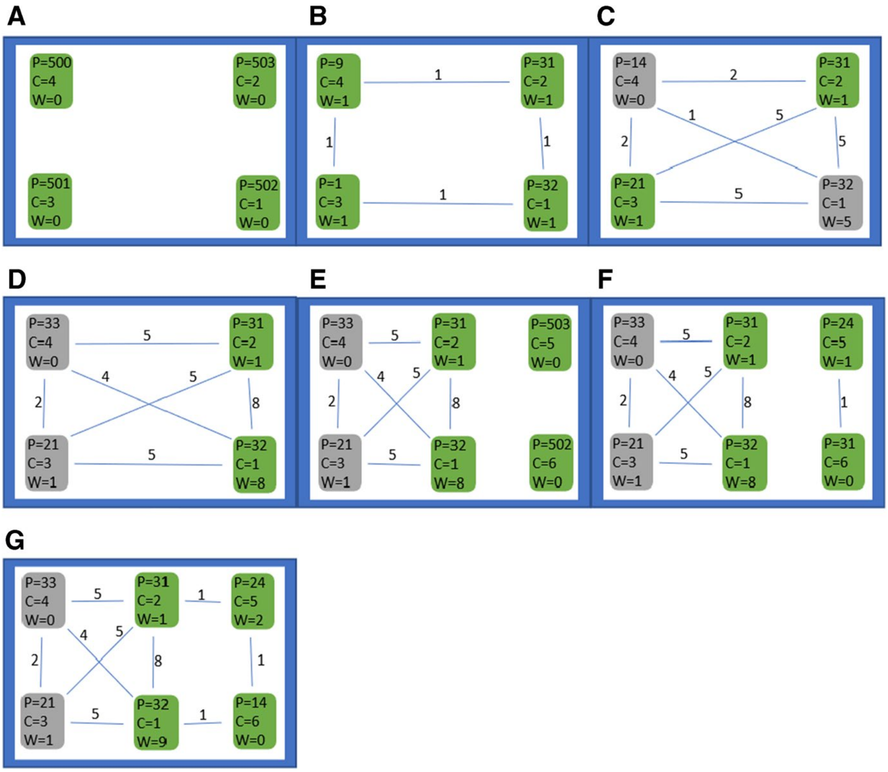

Fig. 10 Results of the validation scenario experiments showing different topologies and the assigned PCI values per cell

7.3.2 Evaluation Scenarios

Besides validating that the ALPACA algorithm works, we also evaluated the algorithm by testing it on a large scale. For this we also used our simulator. However, it would be difficult to come up with random network topologies. Therefore, we have used the Mozilla Location Service database [65]. The Mozilla Location Service is an open and crowd-sourced service which stores information about cells, Wi-Fi access points, etc. This data can be freely downloaded and is in a format that was developed in cooperation with the OpenCellID project [66], another open source service with the same objective. The data is formatted in a Comma-separated values (CSV) way, with fields that are of interest:

- radio—Network type (GSM, UMTS or LTE)
- unit—For LTE networks, this is the PCI
- lon—Longitude location of cell
- lat—Latitude location of cell
- range—Estimate of radio range, in meters.

- net—Mobile Network Code (MNC)

To be able to use this data, we created another tool in our simulator package: The Mozilla Location Service Parser. This tool takes as input the CSV file from the Mozilla Location Service and talks to the topology generator tool that outputs the .topo file based on the information in the CSV file. We then use this .topo file in the topology simulator tool to evaluate our ALPACA algorithm.

The Mozilla Location Service Parser has the ability to filter data based on the previously described fields. For the evaluation scenario we filter as follows:

1. radio—We only look at LTE cell.
2. unit—We only look at cells that have a reported PCI value.
3. lon and lat—We use the longitude and latitude information to locate the cells. For our experiments, we define the greater area of the city of Brussels, Belgium and filter for any reported cells. To be precise, we examine the area from 50.757841 to 50.939150 latitude, and from 4.233291 to 4.562366 longitude.
4. range—We use the range of cells to determine the coverage area. We take the simplified assumption that the coverage area is a circle with the radius equal to this field in meters.
5. net—Based on the data so far, we determined that there are three possible operators in the area of Brussels. In the data they are defined as operator 1, 10 and 20. We further filter the data on operator 20, as it had the most cells in the area. This has been done because different operators use different frequencies, while the PCI assignment is related to solving inter-cell interference [52].

To determine which cells are neighbors, taking our simplified assumption that the coverage area of a cell is a circle, we simply check if two coverage circles of two cells intersect. If they do, then the two cells are viewed as neighbors. This information is taken into account when generating the topology file. As in the evaluation we are interested in the number of PCI issues, and not where exactly they are, we further simplify the experiment by not taking into account the cell load, RSRP or RSRQ values. This is justified as the weights in that case are used to move the PCI modulo issues to parts of the network where they least impact the performance of the network. However, the PCI issues will remain, so the total number of PCI issues is not influenced by the weights. Also, minimizing the effects of the PCI issues is analysed in the validation scenario.

To compare the ALPACA algorithm, we also implemented a random PCI assignment algorithm in the PCI xApp, as its one of the most used algorithms by operators [53]. Therefore, we are able to compare the baseline network PCI assignment from the Mozilla Location service file, the random PCI assignment and the PCI assignment done by ALPACA. For the random PCI assignment algorithm, we run the algorithm 100 times, and pick the best outcome. By best outcome we define the outcome that first has the minimum number of collisions, then the minimum number of confusions, and then the minimum number of modulo issues. So for example, if two assignments had the same number of collisions, we pick the one that has a fewer number of PCI confusions. In future work, we plan to compare ALPACA to other PCI assignment algorithms such as graph coloring ones.

7.4 Results

In this subsection, we first present the raw results from our experiments. The discussion and analysis of the results are found in the next, Discussion section.

7.4.1 Validation Scenario Results

The results of the validation scenario experiments can be seen on Fig. 10. In topology A we see the results of the deployment phase of the PCI SON xApp. Four new cells announce themselves, and the PCI xApp gives the cells a PCI from the temporary PCI range, specifically 500, 501, 502 and 503. In topology B, the Weighted ANR xApp calculates the weights, which the PCI xAPP uses to assign PCI values. Each cell has a weight of 1 and each neighbor relation has a weight of 1. ALPACA assigns PCI values 9, 1, 31 and 32. In topology C, we add additional UEs which results in changing the weights of cells and neighbor relations. Also, two new neighbor relations are discovered by the Weighted ANR xApp. Now all four cells are neighbors to each other. This makes the modulo 3 PCI issue inevitable. Cell four has the lowest weight of 0, cells two and three have weight 1, and cell one has the weight of 5. On the other hand, neighbor relation weight between cells one and four is the lowest. ALPACA changes the PCI of cell three to 21, and the PCI of cell four to 14. This creates a modulo 3 PCI issue between cells one and four. Topology D shuffles the UEs around to generate different weights of cells and neighbor relations. Cell four still has the lowest weight of 0, but now the neighbor relation between cells four and three have the lowest weight. This triggers ALPACA to change the PCI of cell four to 33, moving the modulo 3 issue between cells four and three now. An interesting scenario appears in topology E, where two new cells announce themself. The PCI xApp deployment phase is again activated, and the two new cells get PCI values from the temporary range, 502 ad 503. In topology F, the two new cells get PCI assignment based on the neighbor relations and weights discovered by the Weighted ANR xApp. The two new cells form an isolated island of cells, that have no neighbor relation to the first four cells. ALPACA assigns them PCIs 24 and 31. Topology G, however, creates additional UEs which discover that the two new cells actually have neighbor relation with the first four cells. This creates a PCI confusion because cell one can see two cells, cell two and cell six, that have the same PCI value of 31. The weight of cell five is 2, while the weight of cell six is 0. ALPACA changes the PCI value of cell six to 14, which resolves the PCI confusion.

7.4.2 Evaluation Scenario Results

Running the Mozilla Location Service Parser from our simulator package, we were able to detect around 2000 cells in the greater area of Brussels, Belgium. First, we analyse the coverage range of the cells. Saxena et al. [32] define the range of different cells types in different environment. In an urban environment small cells have a coverage range of 20(m) while macro cells have 500 (m). In a suburban this is 40 (m) for small cells and 1000 (m) for macro cells, while in rural environments the coverage range of small cells is 100 (m) and for macro cells 2000 (m). Figure 11 show the number of cells with different coverage ranges for the 2000 cells in the greater Brussels region that we analyse. The majority of cells have a coverage range between 500 and 1000 (m), 875 cells to be precise. There are also also 101 cells with a coverage range of less than 100 (m). a great deal of cells, 484, have a coverage range of more than 1000 (m).

Next, we use the topology with 2000 cells and run it in our network simulator against our Weighted ANR and PCI xApps. We determine the number of PCI issues in the topology itself, which we call the baseline results. We then activate the random PCI algorithm which assigns random PCIs to cells. This experiment was rerun 100 time, and the best results was taken. Finally, we ran the topology against the ALPACA algorithm. The results can be seen on Fig. 12. The figure shows six columns, each corresponding to one category of PCI issues. The bars in the columns represent the number of PCI issues that the baseline, random and ALPACA algorithm had on the 2000 cells topology. We can see that the baseline topology has 372 PCI collisions. The random PCI assignment algorithm managed to drop this down to 144, while ALPACA only had 25. Similar results are seen for PCI confusions. For the modulo PCI issues, we can see that the baseline and random PCI algorithm had similar results, while ALPACA managed to reduce the number of PCI issues.

After analysing the results for 2000 cells, we also analyse the results for 250, 500 and 1000 cells. We do this by limiting more the area of interest in the Mozilla Location service database, specifically we zoom in more into the city of Brussels,

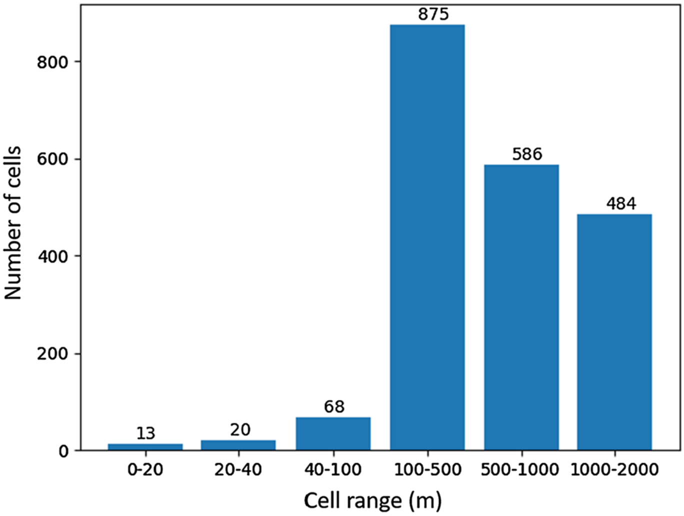

Fig. 11 Number of cells based on their range in the group of 2000 cells from the Mozilla Location Service database in the region of Brussels, Belgium

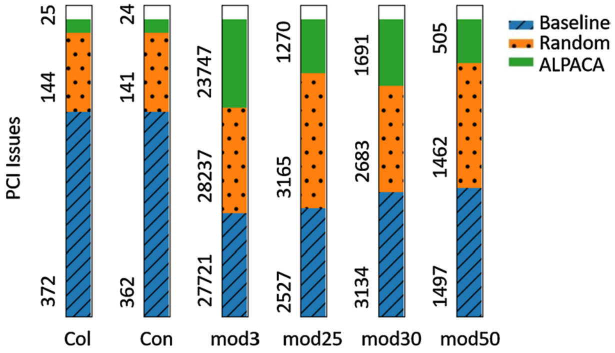

Fig. 12 Number of PCI Issues for 2000 cells from the Mozilla Location Service database

Belgium. Again, we compare the baseline with the random and ALPACA PCI algorithms. The results of the number of PCI issues are shown in Tables 5 and 6. For each algorithm we analyse the number of PCI issues. For 250 cells, we can see that the baseline has 41 collisions and 31 confusions, while even the random PCI algorithm managed to get 0 collisions and 0 confusions at some point in the 100 experiment. Similar results can be seen for 500 cells, except that the random PCI algorithm had 2 collisions. In both cases, ALPACA managed to resolve all collisions and confusions. For 1000 cells, ALPACA only had 1 collision and 1 confusion, while

Table 5 Comparing the number of PCI issues for different scenarios with 250 and 500 cells

Number of cells	250			500		
Algorithm	Base	Random	ALPACA	Base	Random	ALPACA
Collisions	41	0	0	40	2	0
Confusions	31	0	0	30	0	0
mod3	274	317	166	566	624	342
mod25	18	27	1	70	73	1
mod30	23	20	2	54	46	2
mod50	11	13	0	32	37	0

Table 6 Comparing the number of PCI issues for different scenarios with 1000 and 2000 cells

Number of cells	1000			2000		
Algorithm	Base	Random	ALPACA	Base	Random	ALPACA
Collisions	196	30	1	372	144	25
Confusions	187	28	1	362	141	24
mod3	6937	7358	5478	27,721	28,237	23,747
mod25	654	807	132	2527	3165	1270
mod30	811	662	192	3134	2683	1691
mod50	340	384	23	1497	1462	505

the random algorithm and baseline had a significant number. A similar observation can be made for the modulo PCI issues, where ALPACA manages to reduce the total number of PCI issues.

8 Discussion

8.1 Validation Scenario Results Discussion

The results of the validation scenario experiments can be seen on Fig. 10 and is summarized in Sect. 7.4.1. We saw from topology A and topology E that when new cells are announced in the network. Since the cells are still not on air, no UE can attach to them to generate measurement reports. Therefore, it is important to bring new cells into the network in a way that will avoid PCI issues, especially the PCI collisions and PCI confusions as they impact the network performance the most. The deployment phase therefore uses the temporary PCI range to make sure no other on air cells will have PCI collisions or confusion with the new cells being brought up. On the other hand, the deployment phase also makes sure that the collisions and confusions are avoided among the new cells as well. Also, we see how the temporary range is reused in topology E, once its freed in topology B. This shows how ALPACA using the deployment phase can cope with the dynamic nature of bringing new cells into the network. In Sect. 5 we also define how cells are removed from the network and when, specifically when a network operator manually chooses to remove a cell. This is because the data gathered during the monitoring phase for ALPACA is very valuable. A simple reboot of a cell should not delete all this data, and should not trigger ALPACA to find a better PCI assignment. Only when the operator is sure the cell will be taken off physically, for example, can they manually remove a cell from being taken into account by the algorithms.

Topology B and topology F show us that when the neighbors relations, as well as the cell and neighbor relation weights are discovered by the Weighted ANR, the ALPACA algorithm changes the temporary PCI values and assigns the cells PCI values in a way that avoid collisions, confusions and modulo PCI issues. Topology C intentionally creates four neighboring cells, which leads to an unavoidable modulo 3 PCI issue. The only question is which two cells will be affected. The neighbor relation with the lowest weight of 1 is between cells one and four. That mean only 1 UE will be affected if the modulo 3 issue was between these cells. Anywhere else would mean more UEs would be affected. ALPACA then takes into account the weights of the cells as well. Because changing a PCI requires a cell reboot, it pick the cell that has the lowest weight which in this case is cell four. Cell four's PCI changes to 14 and the modulo 3 problem appears between cells one and four. We can see how the weights of the cells and the weights of the neighbor relations influence where the modulo PCI issue will be to minimize its effect on the network. This can also be seen in topology D, where the UEs get shuffled to change the weights of the cells and their neighbor relations. This time, the neighbor relation with the lowest weight is between cells three and four, and still cell four has the lowest cell weight. There it changes PCI again to 33 moving the modulo 3 PCI issue between cells three and four now. Again, this minimizes the effect this PCI issue has on the network. This shows how the optimisation phase of ALPACA is designed to cope with incremental network growth and dynamic network topology changes in a dense and heterogeneous environment. Section 4 shows us why also taking modulo PCI issues into account is important, while in Sect. 5 we defined how the weights for ALPACA are calculated.

Topology F shows how ALPACA takes care of isolated islands of cells. Because cells are not related, PCI value can be reused. However, if at a later point a UE appears that discovers that the two islands of cells are related, ALPACA changes the PCIs in a way to avoid PCI issue. In this specific case it avoided a PCI confusion between cells two and six.

8.2 Evaluation Scenario Results Discussion

Looking at Fig. 11, we can see that the coverage range of the 2000 cells in the evaluation experimentation was diverse. Based on the categorization from Saxena et al. [32], we can say that in those 2000 cells we have both small cells and macro cells. This confirms that we conducted experiments on a heterogeneous mobile network.

Figure 12, and Tables 5 and 6 show the number of PCI issues for experiments with 2000, 1000, 500 and 250 cells, for the baseline topology, best out of 100 experiments for random PCI assignment, and the ALPACA algorithm. For the experiments with 250 and 500 cells, the ALPACA algorithm manages to resolve all PCI collisions and confusions. When compared to the ALPACA algorithm for 250 cells, the baseline topology had around 65% more modulo 3 PCI issues, while the random algorithm had around 90% more. In case of 500 cells, the baseline had 65% and random algorithm had 82% more modulo 3 PCI issues when compared to ALPACA. As for the other modulo issue, ALPACA only had 1 modulo 25 and 2 modulo 50 PCI issues in both cases, while the baseline and random algorithm had a significantly larger amount. In the experiment with 1000 cells, ALPACA managed to find a PCI assignment for all cells that ended up with only 1 collision and 1 confusion, while the baseline had 196 collisions and 187 confusions. The baseline topology also had 26% more modulo 3 PCI issues than ALPACA, while the random algorithm had 34% more than ALPACA. For other modulo PCI issues, the baseline and random had a drastically larger amount of PCI issues. A similar conclusion can be done for 2000 cells.

We can see that ALPACA manages to resolve all PCI collisions and confusions for less than 1000 cells in this topology. It also reduces the modulo 3 PCI issues, and drastically reduces modulo 25, 30 and 50 PCI issues in all experiments. We notice that the number of PCI collisions and confusions are high in the baseline topology. This can be due to the heterogeneous nature of the topology as seen from Fig. 11. Also, we approximate the coverage area as a circle with a radius that is defined as the approximated range of the cells from the Mozilla Location Service database. This, however, is an approximation and cells in the real world have different coverage shapes. One more reason is the limited number of available PCI values. Some operators may further limit the amount of PCIs they use for macro cells, and use a reserved range for small cells. However, even in these more difficult scenarios, the ALPACA algorithm manages to resolve and reduce the total number of PCI issues.

8.3 SON Discussion

As pointed out by Yunas et al. [67] most small cells deployed today have very limited communication between each other. This makes distributed solutions difficult. Hence, a more centralized solution is required like the Weighted ANR and PCI SON xApps in dRAX. Taking the advantage of having a global overview of the network, as well as gathering network wide metrics, the PCI xApp assigns PCI values to cells in a way that solves the PCI collisions and confusion, but also modulo PCI issues, unlike many research works in literature. As seen in Sect. 4, proper PCI assignment can have a great impact on the performance of the network. In addition to improved performance, the SON xApps can reduce the OPEX significantly as it eliminates the need for expensive skilled labour required for configuration, commissioning, optimisation, maintenance, troubleshooting and recovery of the system [6]. Especially in dense and heterogeneous networks, where the network topology changes dynamically. Here, not only is it important to be able to bring new cells into the network, but adapt to the network changes. Hence the PCI xApps' multiple phases.

9 Conclusion

SON is a tool in 4G and 5G mobile networks that is used to optimize the network. One of the crucial optimization parameters is the PCI. There are two critical PCI issues, collisions and confusions, that can lead to total communication failure. Other, modulo PCI issues, are less grave but can impact still impact the QoS of the network. Therefore, proper PCI assignment is one of the main goals of SON. In this paper, we first conduct a study on the impact of PCI issues using real-life hardware. These findings helped us in creating two SON algorithms: Weighed ANR and PCI assignment algorithm ALPACA. The Weighted ANR generates a neighbor relations table that is enhanced by adding weights to cells and their neighbor relations. Multiple metrics are taken into account, such as cell load, UE throughput, RSRP and RSRQ measurements. We use these weights to create the PCI optimisation algorithm ALPACA. ALPACA assigns PCI values in a way that avoids and minimizes the PCI issue effects. It operates in three phases, allowing it to cope with adding new cells in the network, but also adapting to network topology changes as UEs move around. This allows ALPACA to continuously optimize the network. Our validation and evaluation results show how ALPACA can drastically reduce the number of PCI issues in a network and minimize the effects of inevitable PCI issues. Future work will include a comparison of the number of PCI issues between ALPACA and other state of the art PCI assignment algorithms, as well as further investigation into the algorithm execution interval.

Funding No funding was received for conducting this study.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

1. Feng, G., Xia, F., Zhang, Y., Su, D., Lv, H., Wang, H., Lv, H.: Optimal cooperative wireless communication for mobile user data offloading. IEEE Access **6**, 16 224-16 234 (2018). https://doi.org/10.1109/access.2018.2804900
2. Chang, B.-J., Liou, S.-H.: Adaptive cooperative communication for maximizing reliability and reward in ultra-dense small cells LTE-a toward 5G cellular networking. Comput. Netw. **115**, 16–28 (2017). https://doi.org/10.1016/j.comnet.2017.01.014
3. Xiao, Z., Li, T., Ding, W., Wang, D., Zhang, J.: Dynamic PCI allocation on avoiding handover confusion via cell status prediction in LTE heterogeneous small cell networks. Wirel. Commun. Mob. Comput. **16**(14), 1972–1986 (2016). https://doi.org/10.1002/wcm.2662
4. Alsharif, M.H., Nordin, R.: Evolution towards fifth generation (5g) wireless networks: Current trends and challenges in the deployment of millimetre wave, massive MIMO, and small cells. Telecommun. Syst. **64**(4), 617–637 (2016). https://doi.org/10.1007/s11235-016-0195-x
5. Shakir, M.Z., Alouini, M.-S.: In the area spectral efficiency improvement of heterogeneous network by exploiting the integration of macro-femto cellular networks. In: *2012 IEEE International Conference on Communications (ICC)*. IEEE (2012) https://doi.org/10.1109/icc.2012.6364911
6. Heliot, F., Imran, M.A., Tafazolli, R.: Low-complexity energy-efficient resource allocation for the downlink of cellular systems. IEEE Trans. Commun. **61**(6), 2271–2281 (2013). https://doi.org/10.1109/tcomm.2013.042313.120516
7. Ghosh, D., Mohapatra, P.: Resource allocation using link state propagation in OFDMA femto networks. Comput. Commun. **46**, 3–9 (2014). https://doi.org/10.1016/j.comcom.2014.03.007
8. Ramiro, J.: Self-organizing Networks: Self-planning, Self-optimization and Self-healing for Gsm. Umts and Lte. Wiley, Hoboken (2011)
9. Krichen, M., Barth, D., Marce, O.: Performances evaluation of different algorithms for PCIs self configuration in LTE. In: *2012 18th IEEE International Conference on Networks (ICON)*. IEEE (2012), https://doi.org/10.1109/icon.2012.6506558
10. Acharya, S., Das, A.K., Mondal, A., Goswami, R.T.: Combined MST-graph coloring algorithm for PCI distribution of LTE-advanced self-organizing network. In: *Advances in Intelligent Systems and Computing*. Springer Singapore, pp. 261–270 (2018), https://doi.org/10.1007/978-981-10-7386-1_23
11. *Self-Configuring and Self-Optimizing Network (SON) Use Cases and Solutions (Release 9), TR 36.902 V9.3.1, Technical Report*, 3rd Generation Partnership Project Std. (2011). www.3gpp.org
12. Pratap, A., Misra, R., Gupta, U.: Randomized graph coloring algorithm for physical cell ID assignment in LTE-a femtocellular networks. Wirel. Pers. Commun. **91**(3), 1213–1235 (2016). https://doi.org/10.1007/s11277-016-3522-3
13. Diab, A., Mitschele-Thiel, A.: Development of distributed and self-organized physical cell identity assignment schemes for LTE-advanced systems. In: *Proceedings of the 8th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks - PM2HW2N '13*. ACM Press (2013), https://doi.org/10.1145/2512840.2512852
14. NGMN: Use cases related to self organising networks, overall description. (2007). www.ngmn.org
15. Chrost, L., Grochla, K.: Conservative graph coloring: A robust method for automatic PCI assignment in LTE. In: *Computer Networks*. Springer, Berlin, pp. 268–276 (2013), https://doi.org/10.1007/978-3-642-38865-1_28
16. *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 (Release 11); TS 36.300*, Std. (2015). www.3gpp.org
17. *Physical Channel and Modulation (Release 11), TR 36.211 V11.2.0, Technical Report*, 3rd Generation Partnership Project Std. (2013). www.3gpp.org

18. Gui, J.-H., Xiang Gao, S., Jiang, Z.: PCI planning based on local search in LTE network. *DEStech Transactions on Computer Science and Engineering*, no. msota (2019), https://doi.org/10.12783/dtcse/msota2018/27529
19. Pratap, A., Singhal, R., Misra, R., Das, S.K.: Distributed randomized-clustering based PCID assignment for ultra-dense femtocellular networks. IEEE Trans. Parallel Distrib. Syst. **29**(6), 1247–1260 (2018). https://doi.org/10.1109/tpds.2018.2800050
20. Acedo-Hernández, R., Toril, M., Luna-Ramírez, S., Fernández-Segovia, J.A., Úbeda, C.: Analysis of the influence of PCI planning on the physical uplink control channel in LTE. Wirel. Pers. Commun. **98**(1), 629–649 (2017). https://doi.org/10.1007/s11277-017-4887-7
21. Acedo-Hernández, R., Toril, M., Luna-Ramírez, S., Úbeda, C.: A PCI planning algorithm for jointly reducing reference signal collisions in LTE uplink and downlink. Comput. Netw. **119**, 112–123 (2017). https://doi.org/10.1016/j.comnet.2017.03.022
22. Gui, J., Jiang, Z., Gao, S.: PCI planning based on binary quadratic programming in LTE/LTE-a networks. IEEE Access **7**, 203–214 (2019). https://doi.org/10.1109/access.2018.2885313
23. Bandh, T., Carle, G., Sanneck, H.: Graph coloring based physical-cell-ID assignment for LTE networks. In: *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing Connecting the World Wirelessly - IWCMC '09*. ACM Press (2009), https://doi.org/10.1145/1582379.1582406
24. Ahmed, F., Tirkkonen, O., Peltomäki, M., Koljonen, J.-M., Yu, C.-H., Alava, M.: Distributed graph coloring for self-organization in LTE networks. J. Electr. Comput. Eng. **2010**, 1–10 (2010). https://doi.org/10.1155/2010/402831
25. WEI, Y., WANG, W., PENG, M., MIN, S.: Graph theory based physical cell identity self-configuration for LTE-a network. J. China Univ. Posts Telecommun. **20**(1), 101–107 (2013). https://doi.org/10.1016/s1005-8885(13)60015-1
26. Khwandah, S.A., Cosmas, J.P., Zaharis, Z.D., Lazaridis, P.I., Mihovska, A.: Self-organized scheme for unique cell ID assignment during femtocell deployment. In: *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE (2019), https://doi.org/10.1109/ieeeconf44664.2019.9049036
27. Accelleran: drax. https://www.accelleran.com/products/drax-2/
28. Accelleran: Accelleran. https://www.accelleran.com/
29. Liu, Y., Li, W., Zhang, H., Lu, W.: Graph based automatic centralized PCI assignment in LTE. In: *The IEEE symposium on Computers and Communications*. IEEE (2010), https://doi.org/10.1109/iscc.2010.5546812
30. Sedlacek, M., Bestak, R.: An automatic PCI assignment framework for femtocells in LTE networks. In: *Advances in Intelligent Systems and Computing*. Springer International Publishing (2017), pp. 103–110, https://doi.org/10.1007/978-3-319-65960-2_14
31. Abdulkareem, M.M., Yaseen, S.A., Abdullah, L.M.: Matrix based graph coloring algorithm for LTE-PCI assignment and reassignment reduction. In: *2017 IEEE 8th Control and System Graduate Research Colloquium (ICSGRC)*. IEEE (2017), https://doi.org/10.1109/icsgrc.2017.8070565
32. Saxena, N., Roy, A., Kim, H., Won, J.-J.: Auto-configuration of physical cell ID in LTE femtocellular systems using self organizing networks. Wirel. Netw. **20**(5), 1107–1120 (2013). https://doi.org/10.1007/s11276-013-0636-3
33. Wielgoszewska, A., Lopez-Perez, D., Claussen, H., Gacanin, H.: A centralized method for PCI assignment with common reference signal frequency shift control. In: *2016 IEEE International Conference on Communications (ICC)*. IEEE (may 2016), https://doi.org/10.1109/icc.2016.7511228
34. Amirijoo, M., Frenger, P., Gunnarsson, F., Kallin, H., Moe, J., Zetterberg, K.: Neighbor cell relation list and physical cell identity self-organization in LTE. In: *ICC Workshops - 2008 IEEE International Conference on Communications Workshops*. IEEE (2008) https://doi.org/10.1109/iccw.2008.12
35. Wei, Y., Peng, M., Wang, W., Min, S., Jiang, J.M., Huang, Y.: Automatic distributing schemes of physical cell identity for self-organizing networks. Int. J. Distrib. Sens. Netw. **8**(10), 973713 (2012). https://doi.org/10.1155/2012/973713
36. Liu, Y., Li, W., Zhang, H., Yu, L.: Distributed PCI assignment in LTE based on consultation mechanism. In: *2010 International Conference on Computational Intelligence and Software Engineering*. IEEE (2010), https://doi.org/10.1109/wicom.2010.5601210
37. Amirijoo, M., Frenger, P., Gunnarsson, F., Kallin, H., Moe, J., Zetterberg, K.: Neighbor cell relation list and measured cell identity management in LTE. In: *NOMS 2008 - 2008 IEEE Network Operations and Management Symposium*. IEEE (2008), https://doi.org/10.1109/noms.2008.4575129

38. Oppolzer, J., Bestak, R.: Physical cell identifier assignment in LTE-advanced networks. In: *2012 5th Joint IFIP Wireless and Mobile Networking Conference (WMNC)*. IEEE (2012), https://doi.org/10.1109/wmnc.2012.6416150
39. Oppolzer, J., Bestak, R.: Cell identifier assignment in two-tier femtocell networks. In: *2013 IEEE 77th Vehicular Technology Conference (VTC Spring)*. IEEE (2013), https://doi.org/10.1109/vtcspring.2013.6692470
40. Teyeb, O., Mildh, G., Furuskr, A.: Physical cell identity assignment in heterogeneous networks. In: *2012 IEEE Vehicular Technology Conference (VTC Fall)*. IEEE (2012) https://doi.org/10.1109/vtcfall.2012.6398941
41. Ahmed, F., Tirkkonen, O.: Self organized physical cell ID assignment in multi-operator heterogeneous networks. In: *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*. IEEE (2015), https://doi.org/10.1109/vtcspring.2015.7146077
42. Abdullah, L.M., Baba, M.D., Ali, S.G.A., Lim, A.O., Tan, Y.: New graph colouring algorithm for resource allocation in large-scale wireless networks. In: *2014 IEEE 5th Control and System Graduate Research Colloquium*. IEEE (2014), https://doi.org/10.1109/icsgrc.2014.6908728
43. Panxing, L.I., Wang, J.: PCI planning method based on genetic algorithm in lte network. Telecommun. Sci. **32**(3), 147 (2016). https://doi.org/10.11959/j.issn.1000-0801.2016082
44. Shahab, M.B., Bhatti, A.A.: Neural networks based physical cell identity assignment for self organized 3gpp long term evolution. In: *2012 35th International Conference on Telecommunications and Signal Processing (TSP)*. IEEE (2012), https://doi.org/10.1109/tsp.2012.6256276
45. Zahran, A.H.: Extended synchronization signals for eliminating PCI confusion in heterogeneous LTE. In: *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE (2012), https://doi.org/10.1109/wcnc.2012.6214236
46. Sedlacek, M., Bestak, R.: Analysis of neighbourhood relations for femtocell networks. Wirel. Pers. Commun. **96**(4), 5239–5252 (2016). https://doi.org/10.1007/s11277-016-3738-2
47. Babadi, B., Tarokh, V.: A distributed asynchronous algorithm for spectrum sharing in wireless ad hoc networks. In: *2008 42nd Annual Conference on Information Sciences and Systems*. IEEE (2008) https://doi.org/10.1109/ciss.2008.4558635
48. Mwanje, S.S., Ali-Tolppa, J.: Layer-independent PCI assignment method for ultra-dense multi-layer co-channel mobile networks. In: *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE (2017) https://doi.org/10.23919/inm.2017.7987298
49. Ning, L., Guo, Q., Zhang, H., Wang, Z.: Fuzzy layered physical cell identities assignment in heterogeneous and small cell networks. Electron. Lett. **52**(10), 879–881 (2016). https://doi.org/10.1049/el.2016.0048
50. Tu, S., Liu, M., Waqas, M., Rehman, S., Zhu, R., Liu, L.: FHC-PCIA: A physical cell identification allocation method based on fuzzy hierarchical clustering for heterogeneous cellular network. IEEE Access **6**, 46 976-46 987 (2018). https://doi.org/10.1109/access.2018.2867066
51. Wu, S., Ye, W., Shi, Q.: Heuristic optimizing of PCI assignment in real-life LTE networks. In: *11th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2015)*. Institution of Engineering and Technology (2015), https://doi.org/10.1049/cp.2015.0712
52. Yang, H., Huang, A., Gao, R., Chang, T., Xie, L.: Interference self-coordination: A proposal to enhance reliability of system-level information in OFDM-based mobile networks via PCI planning. IEEE Trans. Wirel. Commun. **13**(4), 1874–1887 (2014). https://doi.org/10.1109/twc.2014.030514.130447
53. Abdullah, L.M., Baba, M.D., Ali, S.G.A.: A novel scheme to resolve PCI conflicts and assignment problems in LTE-femtocell networks. In: *2013 IEEE 3rd International Conference on System Engineering and Technology*. IEEE (2013), https://doi.org/10.1109/icsengt.2013.6650153
54. Lee, P., Jeong, J., Saxena, N., Shin, J.: Dynamic reservation scheme of physical cell identity for 3gpp lte femtocell systems. J. Inform. Process. Syst. **5**(4), 207–220 (2009). https://doi.org/10.3745/JIPS.2009.5.4.207
55. Wu, Y., Jiang, H., Wu, Y., Zhang, D.: Physical cell identity self-organization for home eNodeB deployment in LTE. In: *2010 International Conference on Computational Intelligence and Software Engineering*. IEEE (2010) https://doi.org/10.1109/wicom.2010.5600778
56. Lee, P., Jeong, J., han Kim, Y., Shin, J.: Physical cell identity reservation for 3gpp LTE femtocell. In: *Proceedings of the 4th International Conference on Uniquitous Information Management and Communication - ICUIMC '10*. ACM Press (2010), https://doi.org/10.1145/2108616.2108619

57. Szilágyi, P., Bandh, T., Sanneck, H.: Physical cell ID allocation in multi-layer, multi-vendor LTE networks. In: *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Springer, Berlin pp. 156–168 (2013), https://doi.org/10.1007/978-3-642-37935-2_12
58. Shen, A., Guo, B., Gao, Y., Xie, T., Hu, X., Zhang, Y., Shen, J., Fang, Y., Wang, G., Liu, Y.: A novel PCI optimization method in LTE system based on intelligent genetic algorithm. In: *Lecture Notes in Electrical Engineering*. Springer, Singapore, pp. 350–355 (2017) https://doi.org/10.1007/978-981-10-7521-6_42
59. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983). https://doi.org/10.1126/science.220.4598.671
60. T.I. Project: Openran. https://telecominfraproject.com/openran/
61. Enns, R., Bjorklund, M., Schoenwaelder, J., Bierman, A.: Network configuration protocol (netconf). Internet Engineering Task Force (IETF), RFC 6241 (2011). https://tools.ietf.org/html/rfc6241
62. Li, N., Huang, C., Zhuang, M.: Performance optimization and simulation verification of LTE network planning based on micro coverage. In: *2016 10th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID)*. IEEE (2016) https://doi.org/10.1109/icasid.2016.7873932
63. iperf: www.iperf.fr
64. Marquez-Barja, J., Calafate, C.T., Cano, J.-C., Manzoni, P.: Performance trade-offs of a IEEE 802.21-based vertical handover decision algorithm under different network conditions. In: *2011 IEEE 10th International Symposium on Network Computing and Applications*. IEEE (2011) https://doi.org/10.1109/nca.2011.50
65. Mozilla location service: https://location.services.mozilla.com/
66. Opencellid project: https://opencellid.org/
67. Yunas, S.F., Ansari, W.H., Valkama, M.: Technoeconomical analysis of Macrocell and femtocell based HetNet under different deployment constraints. Mob. Inf. Syst. **2016**, 1–14 (2016). https://doi.org/10.1155/2016/6927678

Ensar Zeljković is currently working at Accelleran as the product manager of the dRAX RIC. He is also pursuing a PhD in Computer sciences as a PhD student at the University of Antwerp and the imec research center in Belgium. He holds a Master's degree in Electrical engineering, department of telecommunications from the Faculty of Electrical engineering, University of Sarajevo, Bosnia & Herzegovina. His research interests are in virtual slicing based management and SDN in heterogeneous wireless networks, as well as research in the domain of resource management in 4G and 5G mobile networks. His research resulted in 11 articles published in international peer-reviewed journals and conference proceedings, as well as in 2 submitted patent applications.

Dimitris Gogos is currently working at Accelleran as a software engineer on dRAX RIC, a cloud-native near real-time RAN Intelligent Controller. His interests lie in 5G mobile networks, O-RAN, network intelligence and automation. As a mobile telecom engineer he worked with various vendors and mobile operators in the past. Dimitris holds a Master's degree in Electrical Engineering, department of Electrical Engineering, KU Leuven, Belgium and a Diploma Degree in Computer and Electrical Engineering from the School of Electrical and Computer Engineering, National Technical University of Athens, Greece.

Gerwin Dox is currently working at Accelleran as a software engineer on dRAX RIC. He holds a Master's degree in computer science from the University of Ghent. While currently working on cloud-native technology, his experience with telecom stems mostly from his previous work at Nokia on embedded technology, and his interest lie in embedded systems, high performance software and applied mathematics.

Steven Latré is an associate professor at the University of Antwerp and director at the research centre imec, Belgium. He is leading the IDLab Antwerp research group (85+ members), which is performing applied and fundamental research in the area of communication networks and distributed intelligence.

His personal research interests are in the domain of machine learning and its application to wireless network optimization. He received a Master of Science degree in computer science from Ghent University, Belgium and a Ph.D. in Computer Science Engineering from the same university with the title "Autonomic Quality of Experience Management of Multimedia Services". He is author or co-author of more than 100 papers published in international journals or in the proceedings of international conferences. He is the recipient of the IEEE COMSOC award for best PhD in network and service management 2012, the IEEE NOMS Young Professional award 2014 and is a member of the Young Academy Belgium.

Johann M. Marquez-Barja currently is an Associate Professor at University of Antwerpen, as well as adjunct researcher in IMEC. He is the Head of the Wireless Cluster at imec IDLab UAntwerp. He is involved in several European research projects, being Principal and Co-Principal Investigator for many projects. He is a Senior member of the IEEE Communications Society as well as the IEEE Education Society, where he participates in the Standards Committee and contributes to the IEEE P1876 - Networked Smart Learning Objects for Online Laboratories standards working group. His main research interests are: 5G advanced architectures; elastic and flexible future wireless networks and its integration and impact on optical networks; IoT clustering; softwarization and virtualization; provisioning and dynamic resource allocation towards dynamic networks. He is also interested in vehicular networks, mobility and handovering within smart cities. He received a personal grant from the EU under the prestigious Alban Programme to purse a PhD degree. Prof. Marquez-Barja has been given several keynotes and invited talks in different major events. He is serving as Editor and Guest editor for four International Journals, as well as participating in several Technical Programme and Organizing Committees for different worldwide conferences/congresses.

Authors and Affiliations

Ensar Zeljković[1] · **Dimitris Gogos**[2] · **Gerwin Dox**[2] · **Steven Latré**[1] · **Johann M. Marquez-Barja**[1]

✉ Ensar Zeljković
ensar.zeljkovic@uantwerpen.be

Dimitris Gogos
Dimitris.Gogos@accelleran.com

Gerwin Dox
Gerwin.Dox@accelleran.com

Steven Latré
Steven.Latre@uantwerpen.be

Johann M. Marquez-Barja
Johann.Marquez-Barja@uantwerpen.be

[1] IDLab Research Group, University of Antwerp - imec, Sint-Pietersvliet 7, 2000 Antwerp, Belgium

[2] Accelleran, Kievitplein 20, 2018 Antwerp, Belgium