*Article
Click here to view linked References

A PCI planning algorithm for jointly reducing reference signal collisions in LTE uplink and downlink

R. Acedo-Hernández*,[a], M. Toril[a], S. Luna-Ramírez[a], C. Úbeda[b]

[a] *Dpt. Ingeniería de Comunicaciones, Universidad de Málaga, Málaga, Spain*
[b] *Ericsson, E-28045 Madrid, Spain*

Abstract

In this paper, a novel automatic planning method is proposed for allocating Physical Cell Identifiers (PCI) to cells in a Long Term Evolution (LTE) system based on handover and cell load measurements. The method aims at avoiding PCI collision and confusion problems, while reducing Reference Signal (RS) collisions between neighbor cells. For this purpose, the problem is formulated to consider RS collisions both in DownLink (DL) and UpLink (UL). Then, a classical graph partitioning algorithm is adapted to solve the graph coloring problem behind PCI planning. The considered algorithm is the multi-level version of the Fiduccia and Mattheyses local refinement algorithm. Performance assessment is carried out on graphs constructed from data collected in live LTE network. Results show that the proposed algorithms can avoid PCI collision and confusion problems, while reducing DL RS collisions and almost eliminating UL RS collisions compared to a random plan.

1. Introduction

In recent years, the evolution of mobile networks has led to a significant increase in network system requirements and complexity. Therefore, network operators seek automated solutions for operation and maintenance tasks to minimize the workload and errors due to human interaction. This justifies that the new Long Term Evolution (LTE) standard is known for its

*Corresponding Author: R. Acedo-Hernández, rah@ic.uma.es, +34 952136311
Email addresses: rah@ic.uma.es (R. Acedo-Hernández), mtoril@ic.uma.es (M. Toril), sluna@ic.uma.es (S. Luna-Ramírez), carlos.ubeda@ericsson.com (C. Úbeda)

self-organization features, which include capabilities for self-planning, self-optimization and self-healing [1].

In LTE, Physical Cell Identifier (PCI) planning has been identified as an important use case for self-planning [2] [3]. A PCI is a low-level cell signature used to identify cells in mobility procedures, such as handovers or cell reselection [4]. The number of possible PCI values is limited, which forces several base stations to share the same PCI. As a result, a wrong assignment of PCIs may cause that two adjacent cells use the same PCI (problem referred to as *collision*) or a serving cell has two neighbors with the same PCI (referred to as *confusion*) [1]. Both issues prevent users from detecting target cells, causing that no radio communication is possible. Therefore, proper PCI allocation is essential for an adequate service performance. Such an allocation is a non-trivial task due to the limited number of PCIs.

At the same time, PCI determines the location in time and frequency of DownLink (DL) signaling channels, amongst which are DL Cell-specific Reference Signals (CRS), and the specific set of Demodulation Reference Signals (DM RS) in UpLink (UL).

In DL, CRSs are transmitted in different subcarriers depending on the PCI assigned to the cell. Thus, each cell has a specific pilot pattern corresponding to its cell identity. The number of possible pilot patterns depends on the antenna configuration, but it is always less than 6 [5], causing that CRSs of surrounding cells often collide. CRS collisions degrade Signal-to-Interference-plus-Noise Ratio (SINR) estimates, reported by the User Equipment (UE), which are used by the eNB to select the modulation and coding scheme (MCS) for DL transmission. Thus, an improper PCI planning may give inaccurate SINR estimates, which leads to inefficient data transmission in the DL [6].

In UL, PCI defines the set of DM RSs transmitted by users, used by the base station for channel estimation and coherent demodulation in the Physical Uplink Shared Channel (PUSCH) and Physical Uplink Control Channel (PUCCH) [5]. If two users of different cells use the same DM RS, the decoding process in PUCCH or PUSCH is degraded, which might cause that the base station could not identify the user correctly. As described in [7], the definition of DM RSs in PUSCH can be decoupled from PCI planning, and, thus, DM RS design can be optimized independently to the PCI plan design. However, this is not the case for PUCCH, where DM RSs can only be controled by PCI planning.

Then, the benefits of considering Reference Signals (RS) in PCI planning

are: a) in the DL, improved channel estimates, and, hence, better radio link efficiency, leading to higher DL user throughput and larger network capacity, and b) in the UL, a reduced PUCCH failure rate, which leads to lower UL delays due to less retransmissions and higher DL throughput due to a more robust Channel Quality Indicator (CQI) reporting.

In the literature, most PCI planning approaches are designed to avoid collision and confusion problems [8] [9] [10]. With these two objectives, PCI planning is usually formulated as a graph coloring problem, which can be solved by graph theoretic algorithms (e.g., greedy algorithm) [8] [9] or general-purpose discrete optimization algorithms (e.g., random search, simulated annealing, tabu search, genetic algorithms, constraint satisfaction algorithms, linear programming, ...) [11]. Preliminary studies considered centralized PCI planning schemes [12] and later studies evaluate distributed versions [13] [9] [14] [15]. Recent works have extended the analysis of PCI planning to heterogeneous LTE networks, consisting of several layers, by considering collision and confusion between cells of different layers [16] [17] [18] [19] [20] [21] [22] [23] [24]. Also, recent works have extended the PCI planning use case to emerging scenarios relevant to 5G, e.g., multi-operator [25] and ultra-dense network [26] scenarios. Moreover, considering UL RS collisions when assigning PCIs in 5G networks will be of the utmost importance, as UL performance is critical in machine type communications for Internet-of-Things applications [27]. However, none of these studies consider DL CRS or UL DM-RS collisions. Only in [28], a simple PCI planning method is proposed to reduce DL CRS collisions between cells of the same site. The method does not avoid DL CRS collisions between non-cosited adjacent cells, nor does it consider UL DM-RS collisions. Likewise, a centralized PCI planning scheme based on a genetic algorithm is proposed in [29] to avoid CRS collisions and thus improve network coverage and DL user throughput.

To the authors' knowledge, no PCI planning method has considered UL and DL RS collision together with PCI collision-confusion problems. As shown later, building a PCI plan for just one of these criteria leads to solutions with suboptimal performance in terms of the other criteria. Likewise, classical PCI planning methods often rely on heuristic graph coloring algorithms that must be designed from scratch with the subsequent developing effort. Moreover, previous PCI planning approaches are based on estimates built by radio network planning tools, and do not make the most of performance statistics in the network management system.

In this work, a novel formulation of the PCI planning problem is presented that considers UL and DL CRS collisions problems as well as PCI collision-confusion problems. In the proposed approach, problem instances are constructed from live network measurements. Unlike previous work, the graph coloring problem in PCI planning is re-formulated as a graph partitioning problem, for which very effective codes exist in the public domain (e.g., METIS [30], JOSTLE [31], Chaco [32]). Then, PCI planning is solved by a multi-level refinement graph partitioning algorithm, which is the common benchmark against which other graph partitioning methods are compared [33] [34] [35]. Finally, the method is validated with a problem instance taken from a real LTE network.

The rest of the paper is organized as follows. Section 2 presents the new formulation of the PCI planning problem that also considers RS collisions, which is the main contribution of this work. Section 3 presents the proposed multi-level refinement graph partitioning algorithm. Section 4 presents the results of the PCI planning method in a real problem instance. Finally, Section 5 presents the main conclusions of the study.

2. Problem formulation

This section begins with the basics of PCI assignment in cellular networks. Then, the proposed graph-theoretic formulation of the PCI planning problem is presented. Finally, the problem is re-formulated as a graph partitioning problem.

2.1. PCI planning in cellular networks

The PCI is obtained by the UE from the Primary Synchronization Signal (PSS) and Secondary Synchronization Signal (SSS). From SSS, the UE obtains the physical layer cell identity group index, $N_{ID}^{(1)}$ (ranging from 0 to 167). From PSS, the UE obtains the physical layer identity index, $N_{ID}^{(2)}$ (ranging from 0 to 2) [5]. Then, the PCI is defined as:

$$PCI(i) = 3 \cdot N_{ID}^{(1)}(i) + N_{ID}^{(2)}(i), \tag{1}$$

ranging from 0 to 503. The number of unique PCI values is limited to 504, so these have to be reused across the network. Such a reuse of PCIs leads to different problems.

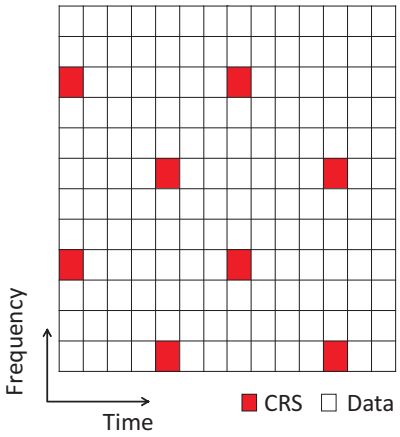
A first issue happens when a UE is in the coverage area of two cells sharing the same PCI. This UE may not be able to identify its serving cell. As a result, radio channel information may not be correctly decoded. This problem is referred to as a *collision* event. A second issue occurs when a UE is camped in a serving cell and two neighbor cells share the same PCI. In this case, the target cell would not be identified properly when a handover is executed. This problem is referred to as a *confusion* event. A good PCI plan should avoid collision and confusion problems assigning the same PCI value to quite distant cells.

In addition to avoiding PCI collision and confusion, a proper PCI plan can increase DL and UL transmission efficiency by reducing RS collisions.

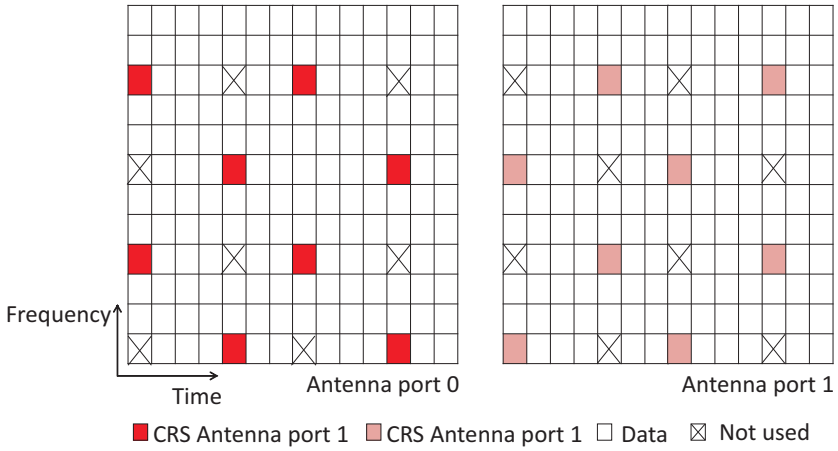
In DL, PCI defines the allocation of CRSs along the Physical Resources Blocks (PRBs) in time and frequency. PRB is the minimum unit of resource allocation in LTE, consisting of a bandwidth of 180 kHz and a time interval of 0.5 ms.

Fig. 1 shows the CRS pattern in time and frequency for 1 PRB and 1 Transmission Time Interval (TTI), with normal cyclic prefix. Squares in the figure represent the different Resource Elements (REs), each consisting of the combination of a subcarrier and an OFDM symbol. Dark squares correspond to REs reserved for CRSs, while white squares correspond to REs reserved for data transmission. In the time domain, CRSs are transmitted in the same Orthogonal Frequency Division Multiplexing (OFDM) symbol of the frame structure in all cells, regardless of the PCI value. However, in the frequency domain, CRSs are allocated to different subcarriers depending on the PCI value of the cell. The number of possible CRS allocation patterns depends on the antenna configuration. As observed in Fig. 1(a), when using one antenna port (i.e., Single Input Single Output -SISO- or Single Input Multiple Outputs -SIMO- schemes), 6 CRS allocation patterns are possible. The specific pattern in a cell is given by the PCI of the cell by the operation PCI mod 6. In case of two or four antenna ports (i.e., Multiple Inputs Single Output -MISO- or Multiple Inputs Multiple Outputs -MIMO- schemes), shown in Fig. 1(b), the CRS allocation pattern is given by the operation PCI mod 3, which coincides with the value of PSS in the cell.

When two cells have the same CRS allocation pattern, CRS collisions degrade DL Signal-to-Interference-plus-Noise Ratio (SINR) estimates. The reason is that the UE assumes that interference in data channel is as high as that experienced in REs where CRS are allocated, which is not necessarily true. SINR estimates are reported by the UE to the eNodeB (eNB) and used



(a) 1 antenna port



(b) 2 antenna ports

Figure 1: Cell-specific reference signal pattern in a PRB and TTI.

later to select the modulation and coding scheme (MCS) for DL transmission. Therefore, if a CRS collision occurs between neighbor cells as a result of a bad PCI assignment, data channel capacity is underestimated. The impact of these errors is larger for lightly loaded networks, where the actual DL SINR value in data resource elements is much higher due to the low interference caused by the absence of traffic in surrounding cells [36].

In UL, the performance of signaling channels is also affected by the PCI plan. PCI determines the set of DeModulation Reference Signals (DM RS) used in a cell for channel estimation and coherent demodulation in UL data and control channels (i.e., PUSCH and PUCCH). If two users of different cells use the same DM RS, the decoding process in PUCCH or PUSCH is degraded, which might cause that the eNB does not identify the user correctly and data symbols are not correctly decoded. As a result, retransmissions take place, which might end up with a transmission failure. PUSCH failures trigger retransmissions in higher-level protocols, which translates into a lower UL user throughput. In contrast, PUCCH failures translate into bad scheduling requests and DL channel quality reporting, which degrade DL throughput performance.

Specifically, DM RSs in a cell are constructed by the combination of base sequences and cyclic shift operations. The base sequence index, u, denoting a specific base sequence group, is calculated as

$$u = (f_{gh}(n_s) + f_{ss}) \bmod 30\,. \tag{2}$$

As observed in (2), the sequence group u assigned to a cell is determined by a sequence-group hopping pattern, f_{gh}, and a sequence group shift offset, f_{ss}, where n_s is the current time slot. f_{gh} is a pseudo-random sequence taken from a set of 17, and f_{ss} is a number between 0 and 29. Thus, there are 17 unique sequence-group hopping patterns, each of which can be shifted by any of the 30 sequence-group shifts offsets. In a cell, the sequence-group hopping pattern, f_{gh}, is the same for PUCCH and PUSCH, and is given by the PCI of the cell. Thus, up to 30 consecutive PCI values have the same f_{gh} (due to the modulus operation). In contrast, the sequence-group shift offset, f_{ss}, may be different for PUCCH and PUSCH. For PUCCH, fss is defined as

$$f_{ss} = PCI \bmod 30 \tag{3}$$

and, for PUSCH,

$$f_{ss} = PCI \bmod 30 + \delta_{ss} \,, \tag{4}$$

where δ_{ss} is a parameter for decoupling the assignment of sequence group from the assignment of PCI. If sequence-group hopping is disabled, $u = f_{ss}$. In the PUSCH case, the assignment of DM RSs in the UL can be decoupled from PCI values by using the shift parameter, δ_{ss}, so that DM-RS collisions between neighbor cells can be avoided easily. However, this is not true for PUCCH, where DM RSs only depend on PCI values, causing that PCI planning determines channel performance. When sequence-group hopping is enabled, a random component is introduced, leading to a negligible probability of collision among DM RSs. Thus, the impact of PCI planning on UL performance is limited.

From the previous explanation, and respecting RS collisions, it can be concluded that:

- an improper PCI planning where closely related cells collide in DL CRS may lead to inaccurate DL SINR estimates, and, as a consequence, to a DL throughput lower than possible. Thus, PCI plan should avoid equal PCI mod 6 (or PCI mod 3, depending on antenna configuration) in cells with high mutual interference levels, and

- an improper PCI plan where closely related cells share the same PUCCH DM RS may lead to PUCCH failures, which degrades DL throughput performance and large UL delays due to retransmissions. Thus, a proper PCI plan should avoid equal values of PCI mod 30 to avoid PUCCH DM-RS collisions in nearby cells with high mutual interference levels.

Both considerations must be additionally taken into account when building a new PCI plan.

2.2. Novel graph-theoretic formulation

The following formulation of PCI planning aims to avoid PCI confusion and collision, DL CRS collisions and UL DM RS collisions. For this purpose, the PCI planning problem is formulated as a multi-objective optimization problem.

First, the network is modeled as a graph, $G = (V, E)$. The vertices in the graph, V, are the cells in the analyzed network area. The complete

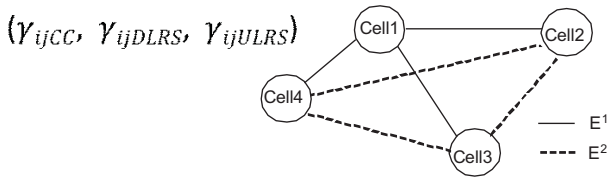


Figure 2: Network graph structure.

set of edges, E, is the union of two set of edges, E^1 and E^2. Edges in E^1 link a cell with its neighbors to model PCI collision problems and DL CRS and UL DM-RS collisions between adjacent cells, while edges in E^2 link neighbors of the same cell to model PCI confusion problems. In both cases, neighbor cells are defined as those cells toward or from which there exist handovers. Alternatively, neighbor cells could also be defined based on site coordinates. In such case, two cells would be neighbors of each other if the distance between their sites is below a certain threshold.

Fig. 2 shows an example of how the graph is constructed for a network consisting of a cell (cell 1) and its three neighbors (cells 2, 3 and 4). Cell 1, the main cell, is linked to neighbor cells 2, 3 and 4 by edges in E^1, while the latter are linked to each other as neighbors of the same cell by edges in E^2.

As shown in Fig 2, the weight of every edge is a triplet, consisting of a collision and confusion weight, $\gamma_{ij_{CC}}$, DL CRS collision weight, $\gamma_{ij_{DLRS}}$, and UL DM-RS collision weight, $\gamma_{ij_{ULRS}}$, representing the penalty associated to the different issues when PCI is not assigned properly between adjacent cells. The first one, $\gamma_{ij_{CC}}$, is calculated as

$$\gamma_{ij_{CC}} = \gamma_{ij_{Col}} + \gamma_{ij_{Con}} \,, \tag{5}$$

where

$$\gamma_{ij_{Col}} = \begin{cases} CollW & \text{if } (i,j) \in E^1 \,, \\ 0 & \text{otherwise} \,, \end{cases} \tag{6}$$

$$\gamma_{ij_{Con}} = \begin{cases} ConfW & \text{if } (i,j) \in E^2 \,, \\ 0 & \text{otherwise} \,, \end{cases} \tag{7}$$

In (6) and (7), the penalty of collision and confusion problems is given by the priority factors $CollW$ and $ConfW$, defined by the operator to prioritize among objectives.

The weight component associated to DL CRS collisions, $\gamma_{ij_{DLRS}}$, is calculated as

$$\gamma_{ij_{DLRS}} = \begin{cases} DLRSW \cdot \frac{HO_{ij}}{\sum_{i,j} HO_{ij}} \cdot (1 - \frac{LDL_i + LDL_j}{2}) & \text{if } (i,j) \in E^1 \,, \\ 0 & \text{otherwise} \,, \end{cases} \tag{8}$$

i.e., the product of the DL CRS collisions priority weight, $DLRSW$, the normalized number of handovers between neighbor cells i and j, $\frac{HO_{ij}}{\sum_{i,j} HO_{ij}}$ (i.e., handovers between cells i and j compared with the total number of handovers between any two cells in the network), and the average ratio of unused PRBs in the adjacency, $(1 - \frac{(LDLi + LDLj)}{2})$, where LDL_i and LDL_j are the DL PRB utilization ratios of cells i and j. The normalized handover factor is used as a measure of potential interference coupling between cells. It is assumed that two cells are generally closer when more handovers take place between them. Thus, the larger $\frac{HO_{ij}}{\sum_{i,j} HO_{ij}}$, the larger $\gamma_{ij_{DLRS}}$, reflecting that RS collisions problems are more severe when two cells are closer due to a higher potential interference level. Likewise, the PRB utilization factor reflects that the benefit from avoiding DL CRS collisions is larger for low network loads. Thus, if $LDL_i = LDL_j = 1$, there is no point in considering DL CRS collisions in the adjacency (i,j) when building a PCI plan. Hence, $\gamma_{ij_{DLRS}} = 0$ if $LDL_i = LDL_j = 1$.

Finally, the weight component associated to UL DM-RS collisions, $\gamma_{ij_{ULRS}}$, is calculated as

$$\gamma_{ij_{ULRS}} = \begin{cases} ULRSW \frac{HO_{ij}}{\sum_{i,j} HO_{ij}} (\frac{LUL_i + LUL_j}{2}) & \text{if } (i,j) \in E^1 \text{ and } f_{gh} \text{ is disabled}, \\ 0 & \text{otherwise} \,, \end{cases} \tag{9}$$

i.e., the product of the UL DM-RS priority weight, $ULRSW$, the normalized number of handovers between neighbor cells i and j, $\frac{HO_{ij}}{\sum_{i,j} HO_{ij}}$, and the UL average ratio of used PRBs in the adjacency, $\frac{(LULi + LULj)}{2}$. Again, the handover factor is used as a measure of potential interference coupling based on cell proximity. However, in this case, the PRB utilization factor reflects

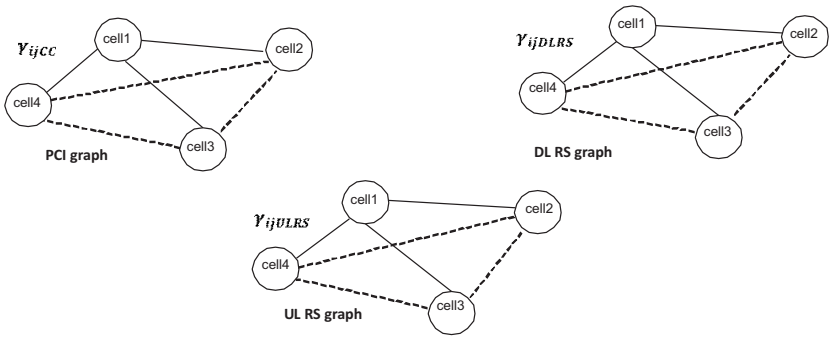


Figure 3: Separation in one graph per problem.

that interference level is higher for higher cell load. Hence, γij_{ULRS}=0 if $LDL_i = LDL_j = 0$, whereas the largest value of $\gamma_{ij_{ULRS}}$ is obtained for the maximum UL PRB utilization (i.e., $LDL_i = LDL_j = 1$).

For convenience, the graph with 3 weights per edge shown in Fig. 2 is broken down into 3 independent graphs with the same set of vertices V and edges E, but different edge weights, $\gamma_{ij_{CC}}$, $\gamma_{ij_{DLRS}}$, and $\gamma_{ij_{ULRS}}$, as shown in Fig. 3. These graphs are hereafter referred to as CC, DLRS and ULRS graphs.

The partitioning of the three graphs carried out by assigning vertices to a limited number of subdomains reflects the assignment of cells to PCIs, DL CRS patterns and DM RS base sequence groups. Specifically, the partitioning of the CC graph into 504 subdomains reflects the assignment of cells to PCIs, the partitioning of the DLRS graph into 3 (or 6 subdomains) reflects the assignment of cells to DL CRS patterns, and the partitioning of the ULRS graph into 30 subdomains reflects the assignment of cells to DM RS base sequence groups. Note that the three partitions obtained are interrelated, since the value of DL CRS pattern and DM RS base sequence group in a cell depends on its PCI. In particular, any partition of the CC graph can be represented as a vector $\Pi_{CC} = [p_1, p_2, \ldots, p_i, \ldots, p_{N_{cells}}]$, where p_i is the PCI assigned to cell i and N_{cells} is the number of cells in the scenario. The corresponding partition of the DLRS graph can be expressed as a vector $\Pi_{DLRS} = \mathrm{mod}(\Pi_{CC}, N)$, with N=3 or 6 depending on the antenna configuration, and that of the ULRS graph can be expressed as a vector $\Pi_{ULRS} = \mathrm{mod}(\Pi_{CC}, 30)$.

A proper PCI plan should be translated into a partition of the 3 graphs that minimizes the objective function:

$$F(\Pi_{CC}) = \begin{cases} F_1(\Pi_{CC}) \text{ if MIMO or MISO} \\ F_2(\Pi_{CC}) \text{ if SISO or SIMO}\,, \end{cases} \tag{10}$$

where

$$F_1(\Pi_{CC}) = \sum_{\substack{(i,j)\in E \\ \Pi_{CC}(i)=\Pi_{CC}(j)}} \gamma_{ij_{CC}} + \sum_{\substack{(i,j)\in E^1 \\ \text{mod}(\Pi_{CC}(i),3)= \text{ mod } (\Pi_{CC}(j),3)}} \gamma_{ij_{DLRS}} + \sum_{\substack{(i,j)\in E^1 \\ \text{mod } (\Pi_{CC}(i),30)= \text{ mod } (\Pi_{CC}(j),30)}} \gamma_{ij_{ULRS}}\,, \tag{11}$$

$$F_2(\Pi_{CC}) = \sum_{\substack{(i,j)\in E \\ \Pi_{CC}(i)=\Pi_{CC}(j)}} \gamma_{ij_{CC}} + \sum_{\substack{(i,j)\in E^1 \\ \text{mod } (\Pi_{CC}(i),6)= \text{ mod } (\Pi_{CC}(j),6)}} \gamma_{ij_{DLRS}} + \sum_{\substack{(i,j)\in E^1 \\ \text{mod } (\Pi_{CC}(i),30)= \text{ mod } (\Pi_{CC}(j),30)}} \gamma_{ij_{ULRS}}\,, \tag{12}$$

i.e., the total sum of the weight of edges joining vertices assigned to the same subdomain in the three graphs. The first addend is the total weight of edges joining vertices that share the same PCI value. The second addend is the total weight of edges joining vertices sharing the same PCI mod 6 (or PCI mod 3), representing cells with the same DL CRS pattern. The third addend is the total weight of edges joining vertices sharing the same PCI mod 30, representing cells with the same UL DM RS base sequence.

2.3. Re-formulation as a graph partitioning problem

To find an optimized PCI plan, any classical graph coloring algorithm can be adapted to deal with the three graphs (i.e., CC, DLRS and ULRS) in parallel. In this work, the graph coloring problem is re-formulated as a graph partitioning problem to take advantage of existing software. Thus, the problem stands for the partition of the three graphs such that the total edge cut (i.e., the sum of the weight of edges between vertices in different subdomains in the three graphs),

$$\sum_{(i,j)\in\delta(\Pi_{CC})} \gamma_{ij_{CC}} + \sum_{(i,j)\in\delta(\Pi_{DLRS})} \gamma_{ij_{DLRS}} + \sum_{(i,j)\in\delta(\Pi_{ULRS})} \gamma_{ij_{ULRS}}\,, \tag{13}$$

is maximized, where $\delta(\Pi_{CC})$, $\delta(\Pi_{DLRS})$ and $\delta(\Pi_{ULRS})$ are the set of edges joining vertices in different subdomains in the three graphs with partitions Π_{CC}, Π_{DLRS} and Π_{ULRS}.

In the classical version of the graph partitioning problem, the goal is to minimize the edge cut [37]. The minimization problem can easily be converted into a maximization problem just by changing the sign of edge weights in the graphs.

3. Graph partitioning algorithm

The graph partitioning problem is known to be $\mathcal{NP}$-hard [38]. For this reason, many heuristic graph partitioning algorithms have been proposed in the literature [39]. In this work, a classical multi-level refinement algorithm is adapted to solve the PCI planning problem as a multi-objective optimization problem [40].

Fig. 4 shows the structure of a classical multi-level refinement algorithm for graph partitioning, consisting of three stages: graph coarsening, initial partitioning and graph uncoarsening [39]. In the coarsening phase, the graph is simplified by collapsing vertices together to form single vertices in a higher level graph. A series of graphs, G^0, G^1, ...,G^m, is thus obtained as successive simplifications of the original graph, G^0. Coarsening continues until the number of vertices in the coarsened graph is slightly equal to the desired number of subdomains. In the simplest graph, a high-quality initial partitioning is built with a low computational load. For this purpose, any basic graph partitioning algorithm can be used (e.g., multiple randomized attempts [41], graph walking [42]). Then, the partition in the coarsest graph is projected back to the original graph in the uncoarsening stage by unfolding vertices. After each uncoarsening step, a local refinement algorithm (e.g., greedy or Fiduccia-Mattheyses [43]) is used to improve the quality of the partition by re-assigning vertices to other subdomains. For computational efficiency, refinement is only performed in the border of subdomains. Experiments shows that, in cellular network graphs, multi-level refinement outperforms other approaches in terms of both solution quality and execution time [37].

In this work, a classical multi-level graph partitioning algorithm is converted into a graph coloring algorithm for PCI planning. Fig. 5 shows the pseudocode of the algorithm.

In Stage 0 (*graph construction*), the graph structure is built as explained in Section 2.2. Then, in Stage 1 (*coarsening*), the original graph is progres-

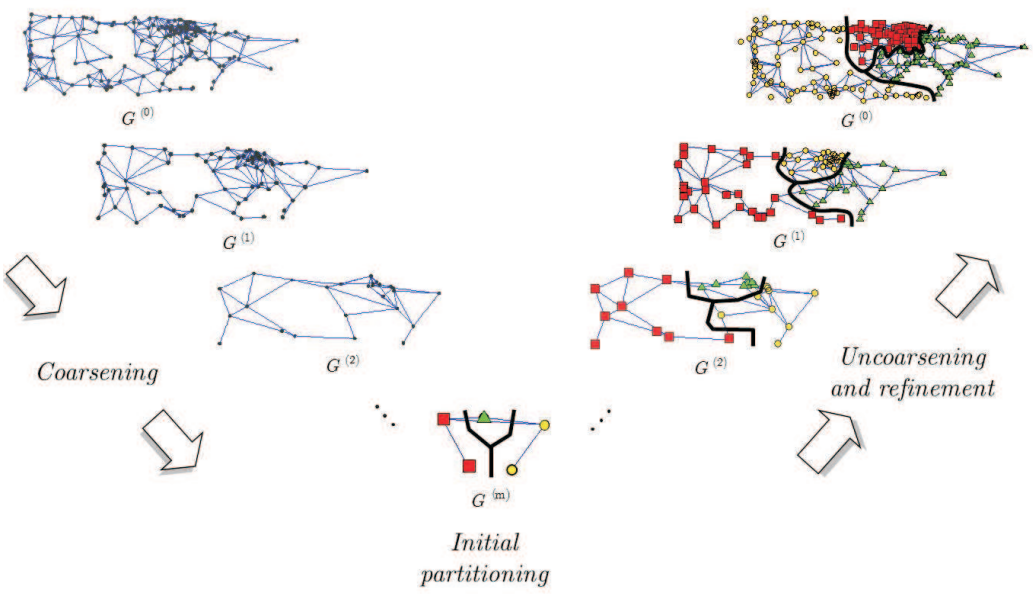


Figure 4: Classical multi-level refinement algorithm.

sively coarsened. Unlike classical approaches, the matching of vertices aims at grouping those vertices having the weakest relationship between them. In this work, the distance between any pair of vertices is calculated in terms of hops by the Floyd-Warshall algorithm [44]. Then, pairs of vertices are ranked based on their hop distance and those with the largest distance are matched. As a result of matching, a coarser version of the CC, DLRS and ULRS graphs is obtained, where the set of edges incident to each new vertex is the union of edges incident to the collapsed vertices. Coarsening continues until the number of vertices in the coarsest graph is equal to the number of available PCI values (504 max.).

In Stage 2 (initial partitioning), a direct assignment of PCI values to vertices in the coarsest CC graph is done. The partitions in DLRS and ULRS graphs are derived from the partitions in the CC graph, as

$$\Pi^n_{DLRS} = \text{mod}(\Pi^n_{CC}, 3 \text{ or } 6)\ , \text{ and} \tag{14}$$

$$\Pi^n_{ULRS} = \text{mod}(\Pi^n_{CC}, 30)\ , \tag{15}$$

where Π^n_{CC} is the partition of the coarsest CC graph (i.e., coarsening level

Function Graph_Coloring_Algorithm (C, HO_{ij}, P)
Inputs: $C \equiv$ list of cells, $HO_{ij} \equiv$ handover statistics, $P \equiv$ set of available PCI values,
Output: Partition of CC graph, Π_{CC} (i.e., PCI plan)

% Stage 0: Graph construction
Build basic network graph, $G = (V, E)$
$V \equiv$ cells, $E = E^1 U E^2$, $E^1 \equiv$ adjacencies with handovers, $E^2 \equiv$ adjacencies between neighbors of same cell
Compute edge weights, $\gamma_{ij} = (\gamma_{ij_{CC}}, \gamma_{ij_{DLRS}}, \gamma_{ij_{ULRS}})$
$\gamma_{ij_{CC}}$ [Eq. (5)-(7)], $\gamma_{ij_{DLRS}}$ [Eq. (8)], $\gamma_{ij_{ULRS}}$ [Eq. (9)]
Separate basic network graph into three graphs G_{CC}, G_{DLRS}, G_{ULRS}
$G_{CC} = (V, E) \backslash_{\gamma_{ij}=\gamma_{ij_{CC}}}$, $G_{DLRS} = (V, E) \backslash_{\gamma_{ij}=\gamma_{ij_{DLRS}}}$, $G_{ULRS} = (V, E) \backslash_{\gamma_{ij}=\gamma_{ij_{ULRS}}}$

% Stage 1: Coarsening stage
Repeat % Different coarsening levels
Compute distance in hops between each pair of vertices by Floyd-Warshall algorithm
Rank pair of vertices in terms of hop distance in descending order
Repeat % Different pair of vertices in the same graph
Select next pair of unmatched vertices with the largest hop distance
until all vertices are matched or the number of vertices in coarser graph is equal to the number of available PCI values
Build coarser version of CC, DLRS and ULRS graph by joining matched vertices
until the number of vertices in coarser graph is equal to the number of available PCI values

% Stage 2: Initial partitioning
Build partition for CC graph by direct assignment of subdomain index in the coarsest graph
$\Pi^n_{CC} = P$
Derive partitions for DLRS and ULRS graphs from partition in CC graph
$\Pi^n_{DLRS} = \text{mod}(\Pi^n_{CC}, 3 \text{ or } 6)$ [Eq. (14)], $\Pi^n_{ULRS} = \text{mod}(\Pi^n_{CC}, 30)$ [Eq. (15)]

% Stage 3: Uncoarsening stage
Repeat
Uncoarsen coarser version of CC, DL and ULRS graphs based on matching scheme
Uncoarsen coarser version of CC, DL and ULRS partitions based on matching scheme
Refine partition in CC graph by modified Fiduccia-Matheysses algorithm (Fig. 6)
until the finer graph is the original graph
Derive partitions for the original DLRS and ULRS graphs from partition in original CC graph
$\Pi^0_{DLRS} = \text{mod}(\Pi^0_{CC}, 3 \text{ or } 6)$ [Eq. (14)], $\Pi^0_{ULRS} = \text{mod}(\Pi^0_{CC}, 30)$ [Eq. (15)]

Figure 5: Pseudocode of the graph coloring algorithm for PCI planning.

n), and the value of 3 or 6 in (14) depends on antenna configuration.

In Stage 3 (*uncoarsening*), the three graphs and partitions are unfolded following the matching scheme obtained in the coarsening stage. After each uncoarsening operation, the CC graph partition is refined by a modified version of the *Fiduccia-Mattheyses* local refinement algorithm [43]. The pseudocode of the algorithm is shown in Fig. 6. The algorithm consists of a series of passes. In each pass, the algorithm first computes the gain from re-assigning every vertex (i.e., cell), i, from the current subdomain, p, to a different subdomain, p', in the CC graph, as

$$g(i,p') = \Delta F(\Pi_{CC}) = F(\Pi_{CC})|_{\Pi_{CC}(i)=p} - F(\Pi_{CC})|_{\Pi_{CC}(i)=p'} = \left(\sum_{\substack{(i,j)\in E \\ \Pi_{CC}(i)=\Pi_{CC}(j)}} \gamma_{ij_{CC}} \Big|_{\Pi_{CC}(i)=p} - \sum_{\substack{(i,j)\in E \\ \Pi_{CC}(i)=\Pi_{CC}(j)}} \gamma_{ij_{CC}} \Big|_{\Pi_{CC}(i)=p'} \right) + \left(\sum_{\substack{(i,j)\in E^1 \\ \Pi_{DLRS}(i)=\Pi_{DLRS}(j)}} \gamma_{ij_{DLRS}} \Big|_{\Pi_{CC}(i)=p} - \sum_{\substack{(i,j)\in E^1 \\ \Pi_{DLRS}(i)=\Pi_{DLRS}(j)}} \gamma_{ij_{DLRS}} \Big|_{\Pi_{CC}(i)=p'} \right) + \left(\sum_{\substack{(i,j)\in E^1 \\ \Pi_{ULRS}(i)=\Pi_{ULRS}(j)}} \gamma_{ij_{ULRS}} \Big|_{\Pi_{CC}(i)=p} - \sum_{\substack{(i,j)\in E^1 \\ \Pi_{ULRS}(i)=\Pi_{ULRS}(j)}} \gamma_{ij_{ULRS}} \Big|_{\Pi_{CC}(i)=p'} \right). \tag{16}$$

For each vertex, the best target subdomain, $p^*(i)$, is selected as the one with the maximum gain, $g^*(i)$. Then, the vertex with the largest potential gain is moved, and marked so that it is discarded for the rest of the pass. This process is repeated until there are no unmarked nodes. Note that any re-assignment of a vertex to a different subdomain in the CC graph requires updating the partition in the DLRS and ULRS graphs by (14) and (15). After each pass, the solution with the lowest cost, $F(\Pi_{CC})$, is selected. To reduce the computational load, refinement is only performed on vertices sharing subdomain with its adjacent vertices, i.e., vertices that have equal PCI, $\mathrm{mod}(PCI, 3 \text{ or } 6)$, or $\mathrm{mod}(PCI, 30)$. Likewise, vertices can be re-assigned to any subdomain when refining a partition (and not only to the subdomains of its adjacent vertices, as in classical graph partitioning).

Function Local_Refinement (G_{CC}, G_{DLRS}, G_{ULRS}, Π_{CC})

Inputs: CC/DLRS/ULRS graphs G_{CC}, G_{DLRS}, G_{ULRS}, CC partitition Π_{CC}

Output: Refined CC partition, Π'_{CC}

For a pre-determined number of iterations

Set the state of every node to unmarked

Derive DLRS and ULRS partitions from CC partition

$\Pi^n_{DLRS} = \text{mod}(\Pi^n_{CC}, 3 \text{ or } 6)$ [Eq. (14)], $\Pi^n_{ULRS} = \text{mod}(\Pi^n_{CC}, 30)$ [Eq. (15)]

For each vertex i

Compute largest potential gain from assigning vertex i to a different subdomain $P(i)$

$g(i,p') = \Delta F(\Pi_{CC}) = F(\Pi_{CC})|_{\Pi_{CC}(i)=p} - F(\Pi_{CC})|_{\Pi_{CC}(i)=p'}$ [Eq. (16)]

$g^*(i) = max_{p'} g(i,p')$

$p^*(i) = argmax_{p'} g(i,p')$

End

Rank unmarked vertices in terms of $g^*(i)$

Repeat % re-assignment of a vertex

Select unmarked vertex i with largest $g^*(i)$

Re-assign i to subdomain $p^*(i)$

Set the state of re-assigned node i to marked

Update partition in DLRS and ULRS graphs

$\Pi^0_{DLRS} = \text{mod}(\Pi^0_{CC}, 3 \text{ or } 6)$ [Eq. (14)], $\Pi^0_{ULRS} = \text{mod}(\Pi^0_{CC}, 30)$ [Eq. (15)]

Save intermediate CC partition and compute intermediate overall cost $F(\Pi_{CC})$

$$F(\Pi_{CC}) = \sum_{\substack{(i,j)\in E \\ \Pi_{CC}(i)=\Pi_{CC}(j)}} \gamma_{ij_{CC}} + \sum_{\substack{(i,j)\in E^1 \\ \Pi_{DLRS}(i)=\Pi_{DLRS}(j)}} \gamma_{ij_{DLRS}} + \sum_{\substack{(i,j)\in E^1 \\ \Pi_{ULRS}(i)=\Pi_{ULRS}(j)}} \gamma_{ij_{ULRS}} \text{ [Eq.(10)]}$$

Update potential gains and best target subdomains of unmarked vertices, $g^*(i)$ and $p^*(i)$

until there are no unmarked nodes or the number of consecutive movements without reducing the overall cost exceeds a certain limit

Restore intermediate solution with lowest overall cost

End

Figure 6: Pseudocode of modified Fiduccia-Mattheyses algorithm.

The theoretical worst-case time complexity of the proposed multi-refinement algorithm is that of the Floyd-Warshall algorithm for computing the distance between vertices, i.e., $O(|V|^3)$, where $|V|$ is the number of vertices in the graph [45]. In practice, the execution time is dominated by the computational load of the local refinement algorithm, which grows linear with the number of vertices in the graphs [43].

4. Performance analysis

The proposed method is tested in a problem instance constructed with data taken from a live LTE network. For clarity, the analysis set-up is first described and results are presented later.

4.1. Analysis set-up

The algorithm is tested with real data taken from 620 LTE cells, covering a seamless geographical area of 290 km^2. Fig. 7 shows the network graph of the area. Vertices in the graph, representing cells, are located in the location of the corresponding base stations. Edges in the graph represent adjacencies between cells. Two cells are considered adjacent if there are handovers between them, which is derived from handover statistics. In the figure, it is observed that some cells are isolated, since they do not have incoming or outgoing handovers. It is assumed that these cells do not cause collision or confusion problems, as they are isolated.

The inputs of the algorithm are collected from configuration parameters (i.e., neighbor cell list and antenna configuration) and performance measurements (i.e., handovers) in the network management system. From this data, the network graph is constructed. Edges are derived from handover statistics, while edge weights are computed from priority factor parameters and handover and cell load statistics by (5)-(9). For ease of analysis, it is assumed that the cell load (i.e., PRB utilization ratio) is the same for all cells in both DL and UL cases, i.e., $LDL = LUL = 50\%$. By eliminating the influence of cell load on CRS and DM RS collision probability, it is easier to show the impact of priority weights on the number of collisions. Likewise, it is assumed that base sequence group hopping is disabled in the UL, as it is normally the case in live networks.

Six different PCI planning approaches are tested. Four are obtained with the proposed algorithm by configuring internal parameters $CollW$, $ConfW$, $DLRSW$ and $ULRSW$ as described below:

- *Random plan.* PCI plan is built by assigning PCI to cells randomly. To improve solution quality with this option, 100 random plans are built and the one with the best quality is selected.

- *CollConf plan.* PCI plan is designed to avoid PCI collisions and confusion problems. For this purpose, the proposed solution algorithm is configured with $CollW = ConfW = 1$, $DLRSW = 0$, $ULRSW = 0$.

- *DL plan.* PCI plan is designed to minimize the number of DL RS collisions. As PRB utilization is the same for all cells, this plan aims to reduce the number of DL RS collisions in the network. For this purpose, the proposed algorithm is configured with $CollW = ConfW = 0$, $DLRSW = 1$, $ULRSW = 0$.

- *UL plan.* PCI plan is designed to minimize the number of UL RS collisions. For this purpose, the proposed algorithm is configured with $CollW = ConfW = 0$, $DLRSW = 0$, $ULRSW = 1$.

- *CollConfDLUL plan.* PCI plan is built taking into account all restrictions. For this purpose, the proposed algorithm is configured with $CollW = ConfW = 1$, $DLRSW = 1$, $ULRSW = 1$.

- Baseline plan. PCI plan currently implemented in the live network, which only takes PCI collision and confusion problems into account.

In the CollConf, DL, UL and CollConfDLUL plans, the best partition is constructed by executing the above-described multilevel refinement algorithm over the graphs built with different priority weights. By using the same heuristic algorithm in all four approaches, it is intended to show the impact of formulating the problem with different constraints. It is expected that the proposed multi-level refinement algorithm should give very high quality solutions, as reported in [40]. All these plans are deterministic, unlike the one from the random approach, which is the best of 100 randomized attempts.

The key performance indicators used to assess the quality of a PCI plan are: a) the number of adjacencies with PCI collision/confusion problems, and b) the total number of handovers between cells whose RSs collide in DL or UL. The aim of using handovers figures, instead of the raw number of adjacencies where collisions occur, is to quantify not only the number of RS collisions, but also that the severity of these collisions in terms of interference. Thus, a proper PCI plan should not only reduce the number

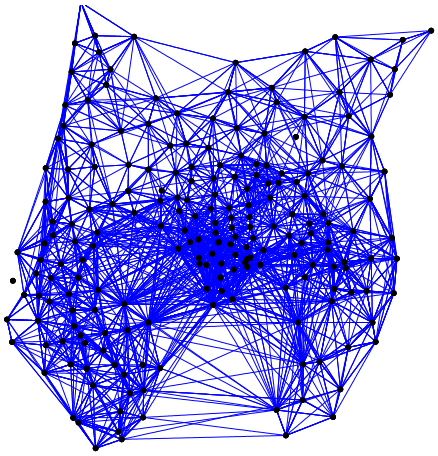


Figure 7: Network graph.

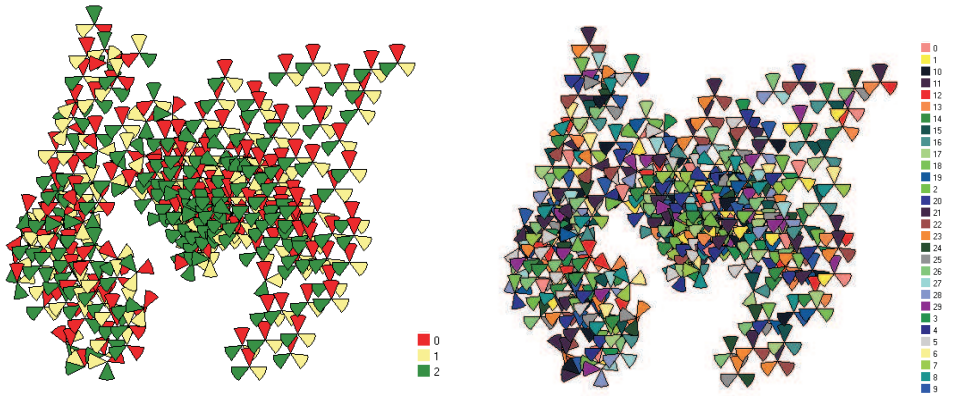
Table 1: Comparison between different PCI plans

Plans	PCI collisions [adjacencies]	PCI confusions [adjacencies]	DL RS collisions [handovers]	UL RS collisions [handovers]
Random	10	35	6243 (-3.22%)	567 (-77.18%)
Baseline	0	0	6048	320
CollConf plan	0	0	4248 (29.76%)	90 (71.87%)
DL plan	0	2	1634 (72.98%)	86 (73.125%)
UL plan	0	3	3788 (37.36%)	24 (92.5%)
CollConfDLUL plan	0	0	2338 (61.55%)	52 (83.75%)

of collisions between cells, but also avoid those with the largest interference levels. It is assumed here that those adjacencies with the largest number of handovers are normally geographically closer than others. The execution time is also used as a secondary performance criterion. For that purpose, the different routines were run on a Windows-based computer with a clock frequency of 2.6 GHz and 8 GB of RAM.

4.2. Analysis results

Table 1 shows the results of the different PCI plans, including the number of adjacencies with PCI collision and confusion problems, and the number of handovers between cells sharing DL CRS and PUCCH DM RS. For comparison purposes, the improvement obtained by each plan compared to the operator plan (baseline) is shown in parentheses. An inspection of the table



(a) PSS plan (PCI mod 3) (b) DMRS sequence plan (PCI mod 30)

Figure 8: DL PSS and UL DM RS plans obtained with the proposed method.

shows that PCIs cannot be assigned randomly, because that leads to many PCI collision and confusion problems (specifically, 45 for a cluster of 620 cells). In contrast, the baseline plan currently implemented in the real network manages to avoid collision and confusion problems, but the number of DL CRS and UL DM RS collisions is close to that obtained with the random PCI plan. The other plans, built with the proposed method, focus on the criterion prioritized by the selected internal parameter settings (either PCI collision/confusion, DL RS or UL RS). In particular, the CollConfplan plan reduces PCI collision/confusion events to 0 by forcing that all edges in the PCI graph have non-negative weights. However, DL RS and UL RS collisions are only decreased by 29.76% and 71.87% respectively. This is clear evidence of the need for considering UL and DL RS collisions when building a PCI plan. DL/UL plans reduce DL/UL RS collisions even further, at the expense of causing some PCI confusion problems. This result points out the need for also considering PCI collisions and confusions when optimizing DL and UL RS plans. The best tradeoff among all criteria is achieved by the CollConfDLUL plan, as it eliminates PCI collision/confusion problems, while obtaining DL and UL collision figures close to the ones obtained by plans specifically designed for these criteria.

Fig. 8 shows the DL PSS and UL base sequence plans built with the CollConfDLUL solution on a map. By eliminating collision problems between adjacent cells at different levels, the resulting plans would improve DL channel quality estimates and reduce the number of PUCCH failures, which

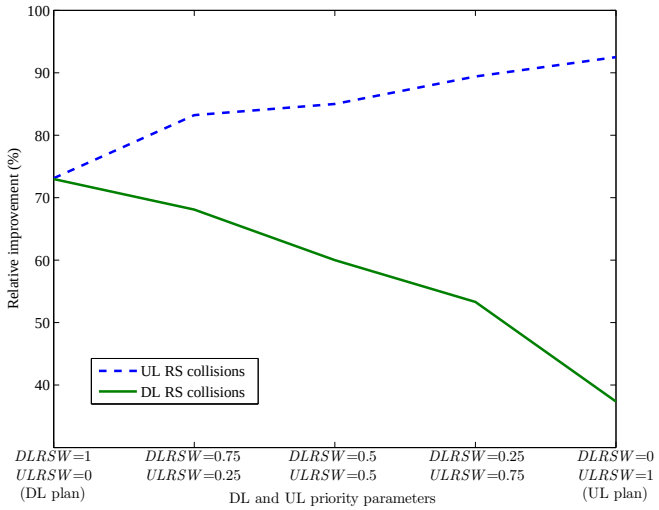


Figure 9: Tradeoff between UL and DL collision performance.

should translate into more robust mobility procedures and higher DL and UL user throughput.

To check the tradeoff between UL and DL performance, the algorithm is tested with different priority settings for $ULRSW$ and $DLRSW$. Fig. 9 shows the relative improvement in the number of DL CRS and UL DMRS collisions compared to the operator solution. Note that the combinations on the left and right of the figure correspond to the DL and UL plans, whose performance is shown in Table 1. As expected, the higher the priority given to the UL, the larger the reduction in DM-RS collisions. It is also observed that half of the maximum potential gain from a careful planning of UL DM RSs can be achieved by increasing DL CRS collisions only by 10% (for $DLRSW$=0.75 and $ULRSW$=0.25 settings).

Table 2 shows the execution times of the different approaches. It is observed that the randomized approach is executed almost instantly. In the 100 attempts tested, the average value is 0.0092 seconds with a 95-th percentile confidence interval of [0.002-0.0312] seconds. In contrast, the proposed approach, CollConfDLUL, takes 720 seconds. This time is short enough to use the method for network planning purposes.

Table 2: Execution time for different PCI planning methods

Plans	Random	CollConf plan	DL plan	UL plan	CollConfDLUL plan
Execution time [s]	0.0092 [0.002-0.0312]	124	660	320	720

5. Conclusions

In this paper, a heuristic self-planning method for Physical Cell Identifiers in an LTE system based on configuration and performance data in the network management system has been proposed. Unlike previous approaches, which were focused on isolated issues caused by PCIs, the proposed method manages to reduce PCI collision/confusion problems, DL CRS collisions and PUCCH DM RS collisions simultaneously. For this purpose, the assignment of PCIs has been formulated as a multi-objective graph partitioning problem, which is later solved by a multi-level refinement graph partitioning algorithm. Performance assessment has been carried out on graphs constructed from data collected in live LTE network. Results show that the algorithm can eliminate PCI collision and confusion problems, while reducing DL RS collisions by 62% and UL RS collisions by 84% compared to the solution currently implemented in the network. As already pointed out in [26], the need for the proposed algorithm will be larger in future ultra-dense heterogeneous scenarios, where the common strategy of defining PCI subsets for each layer reduces the number of available PCI values.

The proposed method can be implemented as a centralized or a distributed solution. In the centralized solution, base stations (enodeBs) take handover and cell load measurements. This data is periodically uploaded in the network management system, where a centralized version of the proposed PCI planning algorithm is executed. Then, the output of the algorithm (i.e., the PCI value assigned to each cell) is sent back to the enodeB and broadcasted to the user in a common control channel. In the distributed solution, each enodeB must exchange with surrounding enodeBs the information required to build a small portion of the graph, i.e., PCI list, handovers and cell load measurements. This information can be periodically exchanged through the X2 interface between adjacent cells. Then, a local refinement algorithm can be executed to decide the re-assignment of PCIs so as to reduce collision and confusion problems with surrounding cells.

Acknowledgement

This work has been funded by the Spanish Ministry of Economy and Competitiveness (TEC2015-69982-R) and Ericsson, Agencia IDEA (Consejería de Ciencia, Innovación y Empresa, Junta de Andalucía, ref. 59288) and FEDER.

References

[1] J. Ramiro, Self-Organizing Networks: Self-Planning, Self-Optimization and Self-Healing for GSM, UMTS and LTE, John Wiley & Sons, 2011.

[2] 3G Americas, The benefits of SON in LTE: Self-Optimizing and Self-Organizing Networks, Technical Report, 2009.

[3] 3rd Generation Parthnership Project, Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Self-Configuring and Self-Optimizing Network (SON) Use Cases and Solutions (Release 9), TR 36.902 V9.3.1, Technical Report, 2011.

[4] 3rd Generation Parthnership Project, Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access Network (E-UTRA); Physical Channel and Modulation (Release 11), TR 36.211 V11.2.0, Technical Report, 2013.

[5] S. Sesia, I. Toufik, M. Baker, LTE: The UMTS Long Term Evolution, From Theory to Practice, Wiley, 2009.

[6] J. Zyren, D. McCoy, Overview of the 3GPP Long Term Evolution Physical Layer, Technical Report, Freescale Semiconductor, 2007.

[7] H. Holma, A. Toskala, LTE for UMTS: Evolution to LTE-Advanced, John Wiley & Sons, UK, 2011.

[8] T. Bandh, G. Carle, H. Sanneck, Graph coloring based Physical-Cell-ID assignment for LTE networks, in: Wireless Communications and Mobile Computing, 2009, pp. 116–120.

[9] F. Ahmed, O. Tirkkonen, M. Peltomki, M. A. Juha-Matti Koljonen Chia-Hao Yu, Distributed graph Coloring for Self-Organization in LTE networks, Journal of Electrical and Computer Engineering (2010).

[10] H. Xu, X. W. Zhou, Y. Li, Model of hypergraph colouring for Self-configuration in LTE Networks, in: International Conference on Information Management, Innovation Management and Industrial Engineering (ICIII), 2011, volume 1.

[11] Z. Matloub, I. Kostaric, Methodology for the physical cell identity assignment in lte networks, International Journal of Networks and Communications 3 (5) (2013) 111–117.

[12] Y. Liu, W. Li, H. Zhang, W. Lu, Graph based automatic centralized PCI assignment in LTE, in: IEEE Symposium on Computers and Communications (ISCC), 2010, pp. 919 –921.

[13] Y. Wei, M. Peng, W. Wang, S. Min, J. M. Jiang, Y. Huang, Automatic Distributing Schemes of Physical Cell Identity for Self-Organizing Networks, International Journal of Distributed Sensor Networks (2012).

[14] Y. Liu, W. Li, H. Zhang, L.Yu, Distributed PCI assignment in LTE based on consultation mechanism, in: Wireless Communications Networking and Mobile Computing (WiCOM), (2010), pp. 1–4.

[15] J. Oppolzer, R. Bestak, Physical Cell Identifier assignment in LTE-advanced networks, in: 5th Joint IFIP Wireless and Mobile Networking Conference (WMNC), 2012, pp. 70–74.

[16] P. Lee, J. Jeong, N. Saxena, J. Shin, Dynamic reservation scheme of physical cell identity for 3gpp lte femtocell systems, Journal of Information Processing Systems 5 (2007) 207–220.

[17] T. Wu, L. Rui, A. Xiong, S. Guo, An Automation PCI allocation method for enodeB and home enodeB cell, in: 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), 2010.

[18] Y. Wu, H. Jiang, Y. Wu, D. Zhang, Physical Cell Identity Self-Organization for Home eNodeB Deployment in LTE, in: 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), pp. 1–6.

[19] X. Zhang, D. Zhou, Z. Xiao, E. Liu, J. Zhang, A. Glasunov, Dynamic group PCI assignment scheme, in: The 7th Int. Conference on Wireless and Mobile Communications, ICWMC, 2011, pp. 101–106.

[20] J. Lim, H. Daehyoung, Management of Neighbor Cell lists and Physical Cell Identifiers in Self-Organizing heterogeneous networks, Journal of Communications and Networks 13 (2011) pp. 367–376.

[21] A. Zahran, Extended Synchronization Signals for eliminating PCI confusion in heterogeneous LTE, in: IEEE Wireless Communications and Networking Conference (WCNC), 2012, pp. 2588–2592.

[22] M. Krichen, D. Barth, O.Marcé, Performances Evaluation of different Algorithms for PCIs Self Configuration in LTE, in: 18th IEEE International Conference onNetworks (ICON), 2012, pp. 197–203.

[23] O. Teyeb, G. Mildh, A. Furuskr, Physical cell identity assignment in heterogeneous networks, in: IEE Vehicular Technology Conference (VTC Fall), pp. 1–5.

[24] P. Szilagyi, T. Bandh, H. Sanneck, Physical cell id allocation in multi-layer, multi-vendor LTE networks, Mobile Networks and Management (Springer) (2013) 156–168.

[25] F. Ahmed, O. Tirkkonen, Self Organized Physical Cell ID Assignment in Multi-Operator Heterogeneous Networks, in: 2015 IEEE 81st Vehicular Technology Conference (VTC Spring), Glasgow, pp. 1–5.

[26] S. S. Mwanje, J. Ali-Tolppa, H. Sanneck, On the Limits of PCI Auto Configuration and Reuse in 4G/5G Ultra Dense Networks, in: 11th International Conference on Network and Service Management (CNSM), 2015, pp. 92–98.

[27] A. Gupta, R. K. Jha, A survey of 5g network: Architecture and emerging technologies, IEEE Access (2015) 1206–1232.

[28] E. Fredik Egrelius, Method and arrangement for PCI assignment, wo2014/120056 a1, 2014.

[29] H. Sun, N. Li, Y. Chen, J. Dong, N. Liu, Y. Han, W. Liu, A method of PCI planning in LTE based on genetic algorithm, Progress In Electromagnetics Research Symposium, Moscow, Russia (2012) pp. 19–23.

[30] G. Karypis, V. Kumar, A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices, Version 4.0, Technical Report, University of Minnesota, Departament of Computer Science / Army HPC Research Center, 1998.

[31] C. Walshaw, M. Cross, Jostle: Parallel multilevel graph-partitioning software - an overview, Mesh Partitioning Techniques and Domain Decomposition Techniques (2007) 27–58.

[32] Chaco, User guide, https://cfwebprod.sandia.gov/cfdocs/CompResearch/docs/guide.pdf, [Accessed November 2016].

[33] B. Hendrickson, R. Leland, A multilevel algorithm for partitions graphs, in: A. Press (Ed.), ACM/IEE Conference on Supercomputing.

[34] G. Karypis, V. Kumar, Multilevel k-way partitioning scheme for irregular graphs, Journal of Parallel and Distributed Computing 48 (1998) 96–129.

[35] C. Walshaw, M. Cross, Mesh partitioning: a multilevel balancing and refinement algorithm, SIAM Journal of Scientific Computing 22 (2000) 66–80.

[36] R. Acedo-Hernández, M. Toril, S. Luna-Ramírez, I. de la Bandera, N. Faour, Analysis of the impact of PCI planning on downlink throughput performance in LTE, Computer Networks vol. 76 (2015) pp. 42–54.

[37] M. Toril, I. Molina-Fernández, V. Wille, C. Walshaw, Analysis of Heuristic Graph Partitioning Method for the Assignment of Packet Control Unit in GERAN, Wireless Personal Communications vol. 60 (2011) pp. 611–633.

[38] M. R. Garey, D. S. Jonhson, Computers and Intractability: A Guide to the Theory of $\mathcal{NP}$-Completeness, Freeman, San Francisco, 1979.

[39] K. Schloegel, G. Karypis, V. Kumar, Graph partitioning for high performance scientific simulations, In CRPC Parallel Computing Handbook, Morgan Kaufmann, 2000.

[40] C. Walshaw, A Multilevel Approarch to the Graph Colouring Problem, Mathematics Research Report 01/IM/69, Computing and Mathematical Sciences, University of Greenwich, 2001.

[41] R. Martí, Multistart methods, Handbook on Metaheuristics (2000) 355–368.

[42] C. Farhat, A simple and efficient automatic fem domain descomposer, Computers and Structures 28 (1988) 579–602.

[43] C. Fiduccia, R. Mattheyses, A linear-time heuristic for improving network partitions, in: Proc 19th ACM/IEEE Design Automation Conference, pp. 175–181.

[44] C. E. Leiserson, T. H. Cormen, C. S. . R. Rivest, Introduction to Algorithm, MIT Press, 1990.

[45] T. H. Cormen, C. Stein, R. L. Rivest, C. E. Leiserson, Introduction to Algorithms, Mc. Graw-Hill Higher Education, 2001.