Author’s version of:
G. Nardini, G. Stea, A. Virdis, A. Frangioni, L. Galli, D. Sabella, G.M. Dell'Aera, "Practical feasibility, scalability and effectiveness of coordinated scheduling algorithms in cellular networks towards 5G", Elsevier Journal of Network and Computer Applications, 106C pp. 1-16, DOI 10.1016/j.jnca.2018.01.007, 2018

Practical feasibility, scalability and effectiveness of coordinated scheduling algorithms in cellular networks towards 5G[*]

G. Nardini, G. Stea, A. Virdis
Dip. Ingegneria dell’Informazione
University of Pisa, Italy

A. Frangioni, L. Galli
Dip. di Informatica
University of Pisa, Italy

D. Sabella[§]
Intel Deutschland GmbH,
Munich, Germany

G.M. Dell’Aera
TIM (Telecom Italia Group),
Turin, Italy

Abstract*— Coordinated Scheduling (CS) is used to mitigate inter-cell interference in present (4G) and future (5G) cellular networks. We show that coordination of a *cluster* of nodes can be formulated as an optimization problem, i.e., placing the Resource Blocks (RB) in each node’s subframe with the least possible overlapping with neighboring nodes. We provide a clever formulation, which allows optimal solutions to be computed in clusters of ten nodes, and algorithms that compute good suboptimal solutions for clusters of tens of nodes, fast enough for a network to respond to traffic changes in real time. This allows us to assess the relationship between the *scale* at which CS is performed and its benefits in terms of network energy efficiency and cell-edge user rate. Our results, obtained using realistic power, radiation and Signal-to-Interference-and-Noise-Ratio (SINR) models, show that optimal CS allows a significant protection of cell-edge users. Moreover, this goes hand-in-hand with a reduction in the number of allocated RBs, which in turn allows an operator to reduce its energy consumption. Both benefits actually *increase* with the size of the clusters. The evaluation is carried out in both a 4G and a foreseen 5G setting, using different power models, system bandwidths and SINR-to-datarate mappings.

Keywords—Coordinated Scheduling, energy-efficiency, cellular networks, inter-cell interference, 5G

I. Introduction

Inter-cell Interference (ICI) is one of the major causes of performance degradation in the downlink of 4G cellular networks, where all neighboring cells share the same spectrum. 5G networks will be denser and with higher traffic demands, which will only exacerbate the problem. User Equipments (UEs) suffering interference from nearby eNodeBs (eNBs) will have a lower Signal-to-Interference-and-Noise Ratio (SINR), hence a lower Channel Quality Indicator (CQI). This means that an eNB will employ more robust modulations, carrying fewer bits per Resource Block (RB), to serve these UEs. Therefore, the network will be able to carry less traffic, and will consume more energy – which is proportional to the number of RB allocated per Transmission Time Interval (TTI) – to carry the same traffic. Moreover, energy efficiency is considered an important design goal for future 5G system [10]. Recent EU-funded research projects (e.g. METIS [12], Flex5Gware [11]), in fact, are considering energy-efficiency as a requirement and setting precise targets on it.

One of the techniques used to reduce ICI is *Coordinated Scheduling* (CS), by which neighboring eNBs agree to use different RBs, i.e., different frequencies, at the same TTI. CS techniques can be either *static* or *dynamic*. In static CS schemes (e.g., [2]-[3]), the partitioning of resources among neighboring eNBs is *fixed*, with a long-term perspective. Typical cases are *frequency reuse* schemes. A static partitioning is highly inflexible, especially when the traffic varies at a fast pace: in fact, no single cell is ever allowed to use the whole spectrum, even if the neighboring ones are unloaded, which leaves resources underutilized. A typical example is a single UE roaming through unloaded neighboring cells, no one of which is able to allocate its entire bandwidth to it. On the other hand, *dynamic* CS schemes have been proposed, e.g., [4]-[8]. Some of these are not standard-compatible, since they assume that the eNBs possess information which is not available in the current 4G standards, and would be costly to introduce in the next-generation 5G ones: for example, they assume that UEs can report the detail of the contributions of the single interferers to the SINR. Some dynamic scheme (e.g., [6],[8]), moreover, assumes that a central entity is in charge of a cluster of cells, and that it both receives per-UE information (i.e., buffer and CQI) *and* makes per-cell schedules *on each TTI*. Such schemes cannot scale with the number of UEs or cells, since both the amount of information to be conveyed and the algorithm complexity are infeasibly high. Under these settings, in fact, achieving an *optimal* result (i.e., a scheme that guarantees the maximum throughput on each TTI) is impossible in practice, since this requires solving to optimality an optimization problem that is too complex for a 1ms-timeframe [8], [9].

Between the two extremes of a static approach and a per-TTI centralized multicell scheduling lies a largely unexplored middle ground, where CS can still be run *dynamically*, but at *longer periods* than the TTI. The outcome of CS can then constrain the scheduling decisions of the coordinated nodes, taken on each TTI, for a whole period. This is the approach pursued

[*] This paper is a substantially revised and extended version of [1].
[§] This work was partially carried out while Dario Sabella was with TIM.

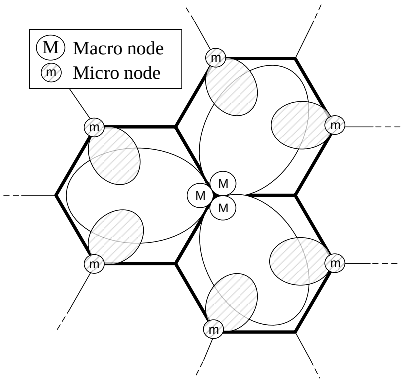


Figure 1 – Nodes in a hexagon tessellation.

in this paper, designed and prototyped within the framework of the Flex5Gware EU-5GPPP project [11]. More in detail, a *global scheduler* (GS), coordinating a *cluster* of nodes, runs a CS algorithm at periods of 100-1000 TTIs. The outcome of CS is an *allocation mask*, i.e. a list of RBs where each node in the cluster can schedule its UEs. That list is compiled so that UEs within a cell are protected from their highest interferers as much as possible. The individual nodes periodically record and send to the global scheduler the number of RBs that they need to carry all their traffic. Moreover, nodes are still in charge of per-TTI scheduling, unlike most Coordinated Multipoint (CoMP) solutions available in the literature, which makes the complexity of our CS independent of the number of UEs, hence more scalable.

A period in the range of 100-1000 TTIs is small enough for a network to be responsive to traffic changes. However, it is also large enough for the CS problem to be formulated as an optimization problem which can be solved at optimality, at fairly large scales (i.e., tens of nodes). While the natural way to formulate a CS problem would be as a Quadratic Semi-Assignment Problem (QSAP, [17]), which is inefficient, we devise instead a non-intuitive *pattern-based* formulation. The ensuing Integer Linear Program (ILP) can be solved at optimality in hundreds of milliseconds at scales of up to ten nodes. Larger scales can be reached (at comparable solution times) by adopting heuristic techniques, such as price-and-branch, where column generation can be handled in different ways, among which brute-force enumeration or the use of a general-purpose solver. Moreover, a second layer of coordination can easily be superimposed, working *among* neighboring clusters to mitigate cluster-border interference, which allows our CS to scale up to *hundreds* of nodes.

The benefits of our optimal dynamic CS are twofold: on one hand, it is effective in protecting cell-edge UEs from the interference of nearby cells. We show that their SINR increases considerably when CS is enforced. On the other hand, protecting cell-edge UEs actually *frees* a considerable amount of RBs at the nodes, namely those RBs that would otherwise be employed to guarantee a suitable data rate to UEs with poor channel conditions. This, in turn, increases the number of bits per RB in the whole network, making it more energy-efficient. An improved energy efficiency naturally translates to a reduced power consumption for the same network load. The above benefits are evident when the system is compared to both an uncoordinated network *and* one with static coordination, e.g. frequency reuse schemes. Moreover, they are confirmed in both a 4G and a foreseen 5G deployment, with increased data rates and improved power models, and in both a macro-only and a heterogeneous scenario, where micro cells are added to the coordinated scheduling problem. Last, but not least, it is worth noting that our dynamic CS framework is fully compliant with the current 4G standard. In fact, it has been implemented and demonstrated in a live prototype of an LTE cellular network [13]-[14].

The contributions of this paper are the following:

- A framework that makes dynamic CS possible, by splitting the scheduling between a GS and the individual nodes of a cluster;
- The design of an exact and two heuristic intra-cluster coordination algorithms to be run at the GS, and a heuristic inter-cluster coordination algorithm;
- An evaluation of the costs (in terms of running time and communication overhead) and benefits (in terms of reduced power consumption and improved SINR) of dynamic CS as a function of the cluster size, in 4G and towards-5G scenarios.

The rest of the paper is organized as follows: Section II describes the hypotheses of the system model and states the problem. Section III reviews the related work. In Section IV we describe our CS models, and in Section V we evaluate the CS performance at various scales. Section VI concludes the paper and highlights directions for future work.

II. System model and problem statement

We consider the downlink (DL) direction of a LTE-Advanced (LTE-A) cellular network, which is more critical than the uplink (UL) one in terms of both carried load and infrastructure power consumption. UEs are served by eNBs and transmissions are arranged in time slots of 1ms, called Transmission Time Intervals (TTIs). During a TTI, nodes allocate *subframes*, i.e. vectors of *M* RBs to its associated UEs. Each RB is a set of contiguous frequency resources allocated to one UE, which carry a fixed number of symbols. The latter translates to different amounts of bits according to the modulation used, which in turn depends on the quality of the air channel, i.e. on the SINR perceived by UEs. In order to allow the eNB to select the appropriate modulation for transmission, UEs report a quantized indication of their SINR (called a Channel

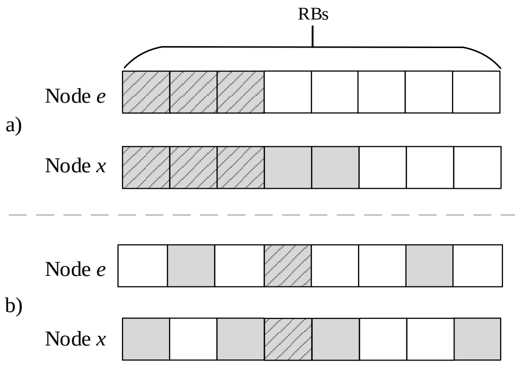


Figure 2 - Examples of first fit (a) and random (b) allocation

Quality Indicator, CQI) to the eNBs periodically. Since all nodes share the same spectrum, they can interfere with each other.

We consider a large-scale multicell cellular network, a portion of which is shown in Figure 1. Although the results in the paper do not depend on a particular network layout, hereafter we often represent cells as *hexagons* for simplicity, and without any loss of generality. Cells host a *macro node* that provides umbrella coverage within them. Moreover, they may have *micro nodes* as well to provide additional capacity. The term "micro" generally refers to a smaller cell, embedded within a macro, regardless of the actual transmission power.

An arbitrary number of UEs is deployed in the floorplan. Each one requests a certain *data rate* and is associated to *one* eNB. In particular, we assume that a UE associates to the node (either macro or micro) from which it perceives the highest SINR, among those covering the cell where they are deployed.

Since CS is intended to run over a timespan larger than the TTI, we are interested in computing the *average* SINR of UEs. Call $P_{x,u}$ the power received by u from node x (which depends on the distance and angle between them, the propagation model and the transmitting power of x). Then, the average SINR of UE u is:

$$SINR_u^e = \frac{P_{e,u}}{N_G + \sum_{x \neq e} P_{x,u} \cdot p_{e,x}^{ov}}, \qquad (1)$$

where N_G is the Gaussian noise and $p_{e,x}^{ov}$ is the *probability* that *u*, served by node *e*, suffers interference from *x* on a RB. This implies that each UE has the same probability of using any of the RBs allocated by its node, which is reasonable given the long timespan. The term $p_{e,x}^{ov}$ depends on how allocation has been performed by nodes *e* and *x*, i.e., which RBs have been allocated. It can be computed as follows. Call n_e and n_x the average *number* of RBs required to nodes *e* and *x* for serving their respective UEs, on each TTI. Define $\Delta_{e,x}$ as the number of *overlapping* RBs in the allocations of the two nodes. This value depends on the *allocation scheme* employed at each node. Two possible modes, which we call *first fit* (FF) and *random* (R) are exemplified in Figure 2, which represents the RB allocation of two arbitrary nodes *e* and *x*. Shaded blocks denote the allocated RBs and dashed ones represent overlapping blocks, i.e. those allocated by both nodes. With FF, RBs are allocated starting from the first position, hence the overlapping RBs are the maximum possible, i.e. $\Delta_{e,x} = \min(n_x, n_e)$. Although FF is the most inefficient approach from an interference perspective, practical implementations of eNBs often employ this strategy. For instance, OpenAirInterface nodes [15]-[16] work like this. On the other hand, a node implementing the R scheme selects RBs in a random fashion. With some straightforward computations, the average number of overlapping RBs is $\Delta_{e,x} = n_x \cdot n_e / M$. This quantity is smaller than with FF, especially at low network loads, i.e. when few RBs are allocated.

Given the number of overlapping RBs, the probability that a UE served by *e* suffers interference from *x* on a RB is $p_{e,x}^{ov} = \Delta_{e,x} / n_e$. Equation (1) becomes:

$$SINR_u^e = \frac{P_{e,u}}{N_G + \sum_{x \neq e} P_{x,u} \cdot \Delta_{e,x} / n_e}. \qquad (1)$$

Thus, the goal of CS is to reduce $\Delta_{e,x}$ for pairs of nodes that generate a high interference on each other's UEs.

Based on the above discussion, the obvious approach to CS would be to select RB placement in the subframes so as to maximize the sum of the average SINR across all UEs. However, this solution suffers from non-trivial *scalability* and *modeling* difficulties. In fact, a cluster of coordinated nodes may handle hundreds of UEs in practical cases, whereas equation (1) is non-linear and non-convex in variables $\Delta_{e,x}$ and n_e. This makes the CS problem hard to solve even for small-size clusters. Moreover, there is an even bigger obstacle: this approach requires that UEs report the received powers $P_{x,u}$ for *all* nodes *x* in the coordinated cluster. In real LTE networks, UEs' reporting is limited to the CQI value and there is no mean for the node to grasp *how* that number was obtained. Thus, (1) cannot be computed, except at the UEs themselves.

III. RELATED WORK

In the literature, ICI has been widely studied and several works have been proposed. They can be categorized in *static* and *dynamic* approaches.

Static schemes allow long-term, network-wide ICI management. Frequency Reuse (FR) schemes [2]-[3] are in this category: the available bandwidth is equally divided into RF portions (RF being the reuse factor) and one eNB is allowed to use $1/RF$ of the bandwidth. A tradeoff between interference and number of usable resources can be found by varying the value of RF. Although this approach can be used for very large scales, it is highly inflexible. In fact, if the load between neighboring cells is unbalanced, it occurs that one eNB may be overloaded, whereas its neighbors have unused RBs. Fractional Frequency Reuse (FFR) can be employed to reduce the amount of unemployed bandwidth, since a part of it is shared among all eNBs to serve cell-center UEs (i.e., with a reuse factor of one), whereas only the remaining part is partitioned into RF portions and exploited for serving cell-edge UEs. FFR mitigates, but does not solve entirely, the problem described above for FR. Soft Frequency Reuse (SFR) has been proposed to overcome the problem of bandwidth underutilization. Like FFR, SFR partitions the bandwidth into RF portions, one of which is reserved to cell-edge UEs. However, eNBs can also use the other subbands, at a lower power, to serve cell-center UEs. Although SFR allows each eNBs to use the whole available bandwidth, it provides less protection from interference than FFR, as observed in [40].

In [5], a dynamic FFR scheme is presented, where the size of cell-center and cell-edge subbands is set according to the cell load. It is based on a graph-coloring algorithm, where UEs are the vertices of the graph. An edge exists between two vertices if the corresponding UEs cannot be scheduled in the same RB (e.g., because they belong to cell-edge zones of neighboring cells). The algorithm assigns one *color* – i.e., one RB – to each node, preserving the interference constraints defined by the graph. However, the fact that only one RB is allocated to one UE might be inefficient and introduce unfairness among UEs, since it does not take into account the UEs' data rate requirement, or their different CQIs. For this scheme, as well, the size of the problem is proportional to the number of UEs. Moreover, the approach requires one to split UEs among cell-center and cell-edge UEs, which in turn requires a central controller to know the position and received power strength and interference for all UEs and cells. This requires additional, non-standard signaling overhead, and hampers scalability.

Dynamic ICI coordination in LTE-A networks is addressed by Coordinated MultiPoint (CoMP) techniques [27]. CoMP allows a cluster of eNBs to share UEs' information and perform coordination operations at fast time scales, typically on each TTI (i.e., one millisecond). CoMP techniques can be divided into Joint Processing (JP) and Coordinated Scheduling and Beamforming (CS/CB). In JP, UEs' data are stored at every eNBs in the cluster and transmissions can be performed by one or more eNBs simultaneously, where transmission points are possibly selected dynamically. On the other hand, CS/CB deals with selecting the best allocation of RBs and/or beam transmission patterns among coordinated eNBs. For our purposes, we consider only CS schemes in this section, [4]-[8]. Broadly speaking, the main problem of the schemes proposed in the above works is that they rely on *per-UE information*, which must be signaled, stored and processed. Thus, these schemes exhibit high signaling costs and limited scalability with the number of UEs and, indirectly, cells.

The authors of [4] propose a CS algorithm for Cloud Radio Access Networks (C-RAN) that jointly optimizes the UE association to the eNBs and their RB allocation. Using graph theory, the problem is first formulated as a maximum-weight clique problem, which is NP-hard. In the literature, some algorithms have been developed for solving more efficiently this problem, e.g. [28]. According to the results reported in the latter, solving the problem on a graph with 1000 vertices would require tens of seconds. In [4], the number of vertices of the graph is given by $U \times C \times M$, where U is the *total* number of UEs in a cluster and C is the number of nodes in the cluster. Considering a small cluster of 10 nodes, 20 UEs per node and 50 RBs, we obtain 10000 vertices, which makes it hardly possible at all to think of solving this problem at optimality in times comparable to our framework's. However [4] also reports a heuristic, whose complexity is linear with the size of the problem $U \times C \times M$. We believe that even this may not be affordable at the network sizes considered in this paper, where we coordinate up to 400 nodes, with a bandwidth of 250 RBs, and an arbitrarily large number of UEs per cell (we use 30 as a proof of concept, but our algorithms do not depend on the number of UEs). In any case, [4] shows no computational results to assess the scalability, or the optimality, of the proposed heuristic.

In [6], an algorithm for computing muting patterns of coordinated eNBs is proposed. Both centralized and distributed architectures are discussed. However, the algorithm assumes to know *two* CQIs for each UE: a "normal" CQI, where all interfering eNBs are assumed to be transmitting, and a "muted" CQI where the strongest interferer is muted. This is again non standard. The problem is then solved using a greedy search algorithm, whose complexity is $O(U^2)$.

Authors of [7] tackle the CS problem by allocating each RB independently based on a proportional fair criterion. This requires that the coordinator knows the channel state of all UEs in every RB.

In [8], the CS problem is solved using a two-layer approach, where large-scale coordination is added on top of a small-scale coordination scheme in order to reduce the signaling and algorithmic complexity. However, [8] requires UEs to convey to their serving nodes 2^{C-1}, different CQIs, obtained in all the possible muting conditions of the $C-1$ interferers, which limits the scalability of small-scale coordination to just three nodes, and the large-scale one to few cluster. The results in [8] are limited to an *overall* 21 cells, whereas our framework scales up by one order of magnitude or more.

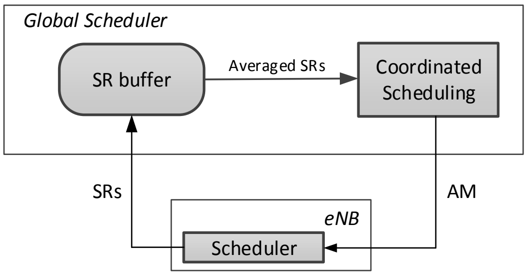
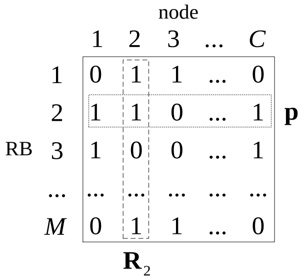


Figure 3 – Overview of Flex5Gware's CS solution



$$\begin{array}{cc|ccccc|c}
 & & & & \text{node} & & & \\
 & & 1 & 2 & 3 & \dots & C & \\
\hline
 & 1 & 0 & 1 & 1 & \dots & 0 & \\
 & 2 & 1 & 1 & 0 & \dots & 1 & \mathbf{p} \\
\text{RB} & 3 & 1 & 0 & 0 & \dots & 1 & \\
 & \dots & \dots & \dots & \dots & \dots & \dots & \\
 & M & 0 & 1 & 1 & \dots & 0 & \\
\hline
 & & & \mathbf{R}_2 & & & &
\end{array}$$

Figure 4 – Allocation masks (columns) and ownership vectors (rows)

Work [39] formulates CS with proportional fair scheduling as an ILP, assuming that each UE has a limited set of "strongest interferers", which should be muted on the RB allocated to that user, and then presents a heuristic that trades optimality for solving time. There are no computational results to show the relationship between cluster size and solving time, and the performance gains are evaluated in relatively small clusters of three to seven nodes.

Moreover, all the above schemes address the problem of allocating RBs to *single* UEs. Now, either this is done at each TTI, which is hardly feasible at all, given the complexity of the algorithms involved, or it is done at longer periods, in which case scheduling is progressively less effective and reactive: with long-term scheduling, fast-paced variations of channel quality cannot be taken into account, and traffic arriving within a period can only be scheduled in the *next* period, which increases delay. On the other hand, our framework handles the CS problem at timescales of hundreds of milliseconds, selecting which RBs can/cannot be used by coordinated eNBs. Allocation of UEs is still performed on each TTI by eNBs autonomously, while taking into consideration the constraints imposed by the CS algorithm.

Several recent works have addressed *placement of CoMP functions*, and the impact of non-ideal backhauling. An introduction to the topic can be found in [34]. Work [37] evaluates distributed and centralized CS deployments (the latter with the CS function placed either at the macro or at the edge cloud), with respect to communication latency and information overhead. Their findings are that a round-trip delay of up to 5 ms is expectable if CS is placed at the edge cloud, which is comparable to what we assume in this paper. Such delay can be added to the running time of our algorithms, which is analyzed in the next sections, when dimensioning the cluster size, based on a maximum period constraint.

A related avenue of research investigates *clustering* of eNBs for CoMP purposes. Several works have appeared lately on the subject (see, e.g., survey [38] and the many references therein). Two different approaches are *network-centric* clustering, whereby a set of eNB forms a cluster, and all the UEs attached to them are part of the same cluster, and *user-centric* clustering (see, e.g., [35]-[36]), whereby each UE may potentially have its own cluster of coordinated eNBs, e.g., for *joint transmission*. Interestingly, the above works point out that the main scalability limitation for CoMP is given by the amount of channel state information that needs be conveyed to make it effective, especially in the user-centric case. We use a network-centric approach in this paper, and address the scalability problem by involving the eNBs in the scheduling and limiting the amount of information sent to the controller to a couple of bytes per eNB per TTI.

IV. Optimization-based Coordinated Scheduling

In this section we discuss our approach to CS, proposed within the Flex5Gware EU project [11], which adopts a different perspective that does away with the problems described in the previous section. In Subsection *A*, we first describe the CS framework, shown in Figure 3, and the role of each node within it. Then we formulate the CS problem as an optimization problem, showing that a non-obvious *pattern-based* formulation is more efficient, but still has scalability problems (subsection *B*). In subsection *C* we discuss possible ways to trade CS optimality for an increase in scale. Subsection *D* shows that a second layer of *inter-cluster CS* can be superimposed to our architecture, and discusses efficient algorithms for it. Subsection *E* evaluates the overhead and optimality of our CS approach, relating time, communication and storage requirements to the network scale.

A. Overview of the CS architecture

The basic philosophy underlying our approach is that per-UE scheduling (i.e., understanding which RBs should be allocated to which UE) in a cell should be done by the cell eNB itself (see Figure 3). The latter, in turn, communicates with a *Global Scheduler (GS)*, that coordinates scheduling in a *cluster* of *C* adjacent cells. The size and membership of a cluster are communicated to the GS by a *Global Power Manager*, which decides which nodes are switched on at any time, using algorithms which are outside the scope of this paper. Nodes in a cluster send *Scheduling Requests* (SR) on each TTI. SRs report the number of RBs required to clear the node's backlog. An average of the latter, computed over a *period* of *T* TTIs (e.g., hundreds or more), is retrieved by the GS and used as an input.

In turn, the GS sends back to each node *i* an *Allocation Mask* (AM) on each period. The latter, shown in Figure 4, is a binary *M*-vector, $\mathbf{R}_i$, where $\mathbf{R}_i[x]=1$ means that node *i can* use RB *x* to schedule its UEs, and must not use it otherwise. Period *T* cannot be chosen arbitrarily. A constraint on its minimum value is the time that the GS employs to compute AMs for its cluster. That time will in turn depend on several factors, notably the size of the cluster itself. Therefore, a trade-off exists between the cluster size and the reactivity of the system. Hereafter, we describe several solutions, which strike different tradeoffs between the two. We refer the interested reader to [14] and [11] for more details on Flex5Gware's software framework.

B. Optimal Coordinated Scheduling

The GS runs an algorithm with the objective of minimizing the global interference in the cluster. The latter is computed as the sum of the overlapping RBs between all pairs of cells *i,j*, weighted by the respective *interference coefficients* (*ICs*) $\alpha_{i,j}$. These coefficients can be derived from live measurements of existing deployments, or possibly from ray-tracing-based simulations. IC $\alpha_{i,j}$ measures the interference that an average UE of cell *j* will hear from cell *i*. ICs form a cluster-wide *interference matrix* $\boldsymbol{\alpha}=\{\alpha_{i,j}\}$. Note that $\boldsymbol{\alpha}$ is not necessarily symmetric, since cells may be anisotropic. Call $\mathbf{C}$ the cluster, with $C=|\mathbf{C}|$, and let $\mathbf{A}$ be the *C*-vector including the SRs for cell *i*. A straightforward, though inefficient formulation of the CS problem is the following:

$$\begin{array}{llr} \min \sum_{i,j} \alpha_{i,j}\cdot\langle \mathbf{R}_i,\mathbf{R}_j\rangle & & \\ s.t.\ \sum_{x=1}^{M}\mathbf{R}_i[x]\ge \mathbf{A}[i], \quad i\in\mathbf{C} & (i) & (2)\\ \mathbf{R}_i[x]\in\{0,1\}, \quad i\in\mathbf{C}, 1\le x\le M & (ii) & \end{array}$$

The objective function minimizes the number of overlapping RBs, with the ICs acting as weights. Notation $\langle \mathbf{R}_i,\mathbf{R}_j\rangle$ represents the inner product of AMs $\mathbf{R}_i$ and $\mathbf{R}_j$. Constraint (2.*i*) forces the sum of RBs allocated to cell *i* to be at least equal to its SR $\mathbf{A}[i]$. Note that equality will hold in (2.*i*) at the optimum in any case, since this is a minimization problem. Coupled with the fact that problem variables are binary (constraint (2.*ii*)), this makes problem (2) a variant of the Quadratic Semi-Assignment Problem (QSAP) [17], which is notoriously hard to solve at optimality, in large part due to its nonlinear objective function. Its size is $O(M\cdot C)$. Problem (2) *can* be linearized by introducing *overlap vectors* $\mathbf{O}_{i,j}$, i.e. binary vectors such that $\mathbf{O}_{i,j}[x]=\mathbf{R}_i[x]$ AND $\mathbf{R}_j[x]$, as follows:

$$\begin{array}{llr} \min \sum_{i,j}\alpha_{i,j}\cdot\sum_{x=1}^{M}\mathbf{O}_{i,j}[x] & & \\ s.t. & & (3)\\ \mathbf{O}_{i,j}[x]\ge \mathbf{R}_i[x]+\mathbf{R}_j[x]-1 \quad i,j\in\mathbf{C}, j\ne i, 1\le x\le M & (i') & \\ \ldots & & \end{array}$$

In the above problem, constraint (3.*i'*) linearizes the logical AND between $\mathbf{R}_i[x]$ and $\mathbf{R}_j[x]$, and the rest remains equal to (2). Introducing overlap vectors, however, inflates the problem size to $O(M\cdot C^2)$: a cluster of $C=10$ cells, each one using $M=100$ RBs, generates a problem with 10^4 binary variables. Beside size, another major disadvantage is *symmetry*: any permutation of the rows of the matrix in Figure 4 yields the same objective. This is known to make it much harder to solve the model at optimality [29]. All the above concur to implying that the solving time of the above model is several orders of magnitude above our requirements (100s of TTIs). A better *formulation* can be found by acknowledging that it is the *ownership* of an RB that matters – i.e., which cells are allocating it – rather than its position in a subframe. In fact, only the former determines inter-cell interference.

Define the *ownership* of a generic RB as a *C*-vector of binaries: for instance $[0,1,1,0,...,0,1]$ means that this RB is allocated simultaneously in the AMs of cells 2, 3, and *C*. In Figure 4, where the AMs are represented as columns, rows are ownership vectors, also called *patterns*. Call $\mathbf{P}$ the set of *possible* patterns, hence $P=|\mathbf{P}|=2^C$. For a pattern $\mathbf{p}\in\mathbf{P}$, call $x_{\mathbf{p}}\ge 0$ the integer variable that counts the *occurrences* of $\mathbf{p}$ in an AM matrix. The *interference cost* of increasing $x_{\mathbf{p}}$ by one unit can be computed *statically*, as:

$$c_{\mathbf{p}}=\sum_{(i,j)\in\mathbf{C}\times\mathbf{C}}\alpha_{i,j}\cdot\mathbf{p}[i]\cdot\mathbf{p}[j]=\mathbf{p}^T\cdot\boldsymbol{\alpha}\cdot\mathbf{p}$$

Given costs $c_{\mathbf{p}}$, the model can be rewritten as follows:

$$\begin{array}{llllr} \min \sum_{\mathbf{p}\in\mathbf{P}} c_{\mathbf{p}}\cdot x_{\mathbf{p}} & & & & \\ s.t. & \sum_{\mathbf{p}\in\mathbf{P}}\mathbf{p}[i]\cdot x_{\mathbf{p}}\ge\mathbf{A}[i] & i\in\mathbf{C} & (i) & \\ & \sum_{\mathbf{p}\in\mathbf{P}} x_{\mathbf{p}}\le M & & (ii) & (4)\\ & x_{\mathbf{p}}\in\square & \mathbf{p}\in\mathbf{P} & (iii) & \end{array}$$

The objective, though formulated differently, is equal to the previous problem's. Constraint *(4.i)* states that the number of RBs in the AM of a node must not fall below its SR, whereas constraint *(4.ii)* caps the number of allocated RBs to the maximum *M*. Variables x_{p} are integer, and there are 2^C of them. This is therefore an Integer Linear Program (ILP), whose size is $O(2^C)$. This ILP is solvable at optimality by a general-purpose solver (such as CPLEX, [18]), and despite the fact that its size is exponential, it *can* be solved in split-second times for medium-sized clusters (e.g., up to 10 cells, which correspond to $2^{10} = 1024$ patterns). Once (4) is solved, the AMs can be found by placing $x_{\mathbf{p}}$ instances of each row $\mathbf{p}$ in *any* order. While (4) is considerably faster than (2) (or its linearized version) at medium scales (e.g., ~10 nodes), it goes without saying that its solving times will become prohibitive at larger scales, due to its exponential size. For this reason, we now discuss other algorithms which trade a little optimality for a reduction in solving time. The latter readily translates to an increase in *scale*, if the maximum solving time is fixed.

C. *Trading optimality for scalability*

ILP (4) can be solved to optimality in split-second times for small clusters. The standard solution algorithm for ILPs is *branch-and-bound* [30], which consists in iteratively solving the continuous relaxation, i.e., the model obtained by relaxing integrality constraints *(4.iii)* on variables $x_{\mathbf{p}}$, so as to compute *bounds*, and then *branching*. However, solving an LP with an exponential number of variables several times is too costly.

A well-known technique to solve LPs where the number of variables is too large is *column generation* (a.k.a. *variable pricing*) [19]. The idea is simple: one starts considering a model with a small subset of the variables, called *restricted master problem*, and generates the other variables only "if needed" in terms of optimality. More precisely, in a minimization problem, a column is needed if it has a *negative reduced cost*, because it can lead to an improvement in the objective function value. Let $\boldsymbol{\lambda}$ and μ be the dual variables associated to constraints (4.*i*) and (4.*ii*). The dual problem reads as follows:

$$\begin{array}{lll} \max & \boldsymbol{\lambda}^T\mathbf{A} + \mu \cdot M & \\ s.t. & \boldsymbol{\lambda}^T\mathbf{p} + \mu \le c_{\mathbf{p}} \quad \mathbf{p} \in \mathbf{P} & (i) \\ & \boldsymbol{\lambda} \ge 0, m \le 0 & (ii) \end{array} \tag{5}$$

Given a dual solution $(\boldsymbol{\lambda}^*, \mu^*)$, the *reduced cost* of variables $x_{\mathbf{p}}$ is given by:

$$c_{\mathbf{p}} - \boldsymbol{\lambda}^{*T}\mathbf{p} - \mu^* .$$

In order to find the pattern $x_{\mathbf{p}}$ with minimum negative reduced cost, or prove that none exist, we seek for a pattern $\mathbf{p} \in \mathbf{P}$ that minimizes $c_{\mathbf{p}} - \boldsymbol{\lambda}^{*T}\mathbf{p}$ (μ^* being a constant). The problem of finding one or more columns with negative reduced cost is called *pricing problem*. Our pricing problem for variables $x_{\mathbf{p}}$ has the following form:

$$\min\left\{\mathbf{p}^T \cdot \boldsymbol{\alpha} \cdot \mathbf{p} - \boldsymbol{\lambda}^* \cdot \mathbf{p} : \mathbf{p} \in \mathbf{P}\right\}, \tag{6}$$

We start by observing that the interference matrix $\boldsymbol{\alpha}$ is non negative, therefore pricing problem (6) could be solved in polynomial time as a *minimum cut* problem, were it not for the linear term $-\boldsymbol{\lambda}^* \cdot \mathbf{p}$. However, we can easily incorporate that term into the matrix as a diagonal term, since diagonal terms in $\boldsymbol{\alpha}$ are null by definition, and since $\mathbf{p}[i] \in \{0,1\}$ implies $\mathbf{p}[i]^2 = \mathbf{p}[i]$. Thus, define the modified interference matrix $\boldsymbol{\alpha} = \{\alpha_{i,j}\}$, where $\alpha_{i,j} = \alpha_{i,j}$ if $i \ne j$ and $\alpha_{i,i} = -\lambda_i^*$, and rewrite (6) as:

$$\min\left\{\mathbf{p}^T \cdot \boldsymbol{\alpha} \cdot \mathbf{p} : \mathbf{p} \in \mathbf{P}\right\}, \tag{7}$$

Problem (7) is an Unconstrained Boolean Quadratic Problem (UBQP) [20], and it is known to be NP-hard. In order to solve a continuous relaxation of (4), we need to solve (7). For this, we have two options:

- A *brute-force* enumeration of all the patterns in $\mathbf{P}$. This is fairly easy, because the UBQP is unconstrained, so the feasible set is simply given by all the vectors in $\mathbf{P}$. Moreover, the quadratic objective function for a given $\mathbf{p}$ can be evaluated in *linear time* if vectors are enumerated so that the hamming distance of consecutive vectors is equal to one, i.e., they only differ by one bit. Indeed, if the hamming distance is one, to evaluate the cost of $\mathbf{p}$ with respect to the previous pattern, we only need to consider the entries in $\boldsymbol{\alpha}$ corresponding to the one bit that has changed, which clearly are $O(C)$, so the cost update can be done in linear time, despite the objective function being quadratic.
- Rely on standard solvers like CPLEX, which can solve 0-1 quadratic programs (QPs).

The brute-force method will generally be fast enough until the number of variables reaches 20 or so. From that scale onward, solving the QP will be faster.

Once we establish that the LP relaxation of our ILP can be solved using column generation, if we then wanted to solve the original ILP to proven optimality, we would have to start branching and pricing *at each node* of the branch-and-bound tree, just in case more columns of negative reduced cost can be found. This method, called *branch-and-price*, is exact and guaranteed to find an optimal solution. However, its computing time is too large, hence we prefer to use a heuristic algorithm called *price-and-branch* (PB). PB is considerably faster, since it only involves pricing *at the root node*, rather than at each node of the branching tree. The final integer solutions that we find may not be optimal. However, we still get a *lower bound* to the optimum of (4) (obtained by solving its linear relaxation at optimality at the root node), hence we are able to bound from below the optimality gap of our heuristic solutions.

D. *Optimizing cluster borders*

Given that autonomous CS instances are run at each cluster, nodes of neighboring clusters can exert uncoordinated interference on cluster-border UEs, hence these will still have a worse SINR. Increasing the cluster dimension generally reduces the percentage of cluster-border UEs: this can be easily seen, for instance, by counting the percentage of cluster-border edges in a cluster as a function of its size in a hexagon deployment. However, our coordination framework leaves room for improving the conditions for cluster-border UEs, by exploiting the output of CS instances run at different clusters. In fact, our pattern-based modeling of CS leaves open the problem of *placing* RBs within a subframe. A solution to problem (4) is a set of non-zero integers $x_{\mathbf{p}}$, stating that a subframe will include $x_{\mathbf{p}}$ instances of pattern $\mathbf{p}$. Thus, a node in the cluster can place these instances at any of the *M* positions in the subframes (this is, in fact, the very expedient by which one avoids symmetry). This degree of freedom can be exploited to minimize the overlap of RB allocation at cluster-border nodes of adjacent clusters. A similar problem has been considered in [8], which shows that it can be formulated as a Quadratic Assignment Problem (QAP), whose size is $O\left(M \cdot K^2\right)$, *K* being the number of clusters. QAPs are NP-hard, and the solution times for a QAP of this scale are, again, orders of magnitude above our timing requirements, even for small values of *K*. For this reason, we employ a fast heuristic, proposed in [8], adapting it to our settings via some modifications.

Consider *K* clusters, each of which is running an autonomous instance of CS, and sort them according to some arbitrary order, for instance, starting from the innermost cluster and going towards the outer ones. Call T_k the set of patterns of cluster *k*. The basic idea is to consider clusters sequentially. The patterns of the first cluster are placed arbitrarily within the subframe. Then, an iterative procedure arranges the patterns of the remaining $K-1$ clusters. The patterns of each new cluster are placed in the subframe so as to minimize the *increase* in the total inter-cell interference, still measured as the weighted overlap of RBs between interfering nodes. In particular, at step *k*, the patterns belonging to T_k are placed according to the solution of the following optimization problem:

$$\begin{aligned}
&\min \sum\nolimits_{i=1}^{M} \sum\nolimits_{\mathbf{p}\in T_k} b_{i,\mathbf{p}} \cdot \sum\nolimits_{\mathbf{q}\ \mathit{active\ in}\ i} \beta_{\mathbf{p},\mathbf{q}} \\
&s.t. \qquad\qquad\qquad\qquad\qquad\qquad . \qquad\qquad (8) \\
&\sum\nolimits_{i=1}^{M} b_{i,\mathbf{p}} \geq x_{\mathbf{p}} \qquad \forall \mathbf{p} \in T_k \qquad (i) \\
&\sum\nolimits_{\mathbf{p}\in T_k} b_{i,\mathbf{p}} \leq 1 \qquad \forall i \qquad (ii) \\
&b_{i,\mathbf{p}} \in \{0,1\} \qquad \forall i, \forall \mathbf{p} \in T_k \qquad (iii)
\end{aligned}$$

The objective function minimizes the overall mutual interference with patterns belonging to T_j, $1 \leq j \leq k-1$. The term

$$\beta_{\mathbf{p},\mathbf{q}} = \sum\nolimits_{\mathbf{p}\in x, \mathbf{q}\in y} \alpha_{x,y}$$

represents the interference that nodes active in pattern **p** produce on UEs served by nodes active in pattern **q**, whereas binary variables $b_{i,\mathbf{p}}$ are set if pattern **p** is placed at RB *i*. As a result, the term

$$\sum\nolimits_{\mathbf{q}\ \mathit{active\ in}\ i} \beta_{\mathbf{p},\mathbf{q}}$$

in the objective accounts for the overall interference that **p** would produce if placed on RB *i*, knowing which patterns have already been placed in that RB during the previous iterations. Constraint (8.*i*) states that at least $x_{\mathbf{p}}$ instances of pattern **p** are allocated, whereas constraint (8.*ii*) avoids that two patterns of the same cluster are placed in the same position. The output will be taken into account for step *k*+1. Problem (8) is again an ILP, and $K-1$ instances of it need to be solved in sequence. However, it can be easily recognized to be a Linear Assignment Problem, i.e. one of the (few) ILPs that can be solved in polynomial time, e.g. via the Hungarian algorithm [21], which is $O\left(K^3\right)$.

E. *Overhead and optimality.*

It is worth mentioning that our CS framework has actually been prototyped and demonstrated [13]-[14], using eNB implementations based on OpenAirInterface and real 4G UEs. Now, the scale of a prototype can only be limited (ours incorporates three nodes), hence the latter is not the right tool to assess large-scale effects such as those we investigate in our paper. However, a prototype still allows one to measure the actual communication and storage overhead involved, and extrapolate the results to larger scales. The main findings in [14] are that the highest communication overhead is due to per-TTI SRs coming from the nodes, whose rate is in the order of 400kbps per node (at the Ethernet level). The network interface of a low-end server network hosting the GS will be able to manage scales of tens of nodes without any trouble. On the other hand, the SRs can be stored into circular buffers, which limits the storage required to perform CS to `sizeof(SR)`×*T* per node, *T* being the CS period in multiples of a TTI (reasonable values being 100-1000). Again, coordinating tens of nodes at a CS period of 1 second would require at most few kilobytes of storage. Therefore, the only possible limitation to scalability may be due to the running time of the CS algorithm.

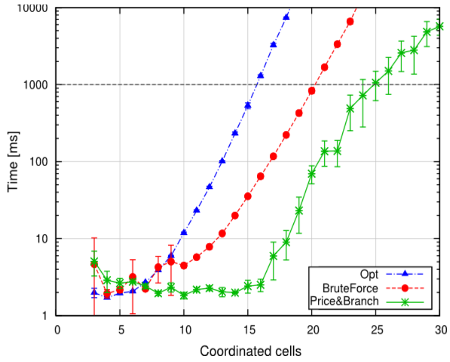


Figure 5 - Average solving times as a function of the cluster size

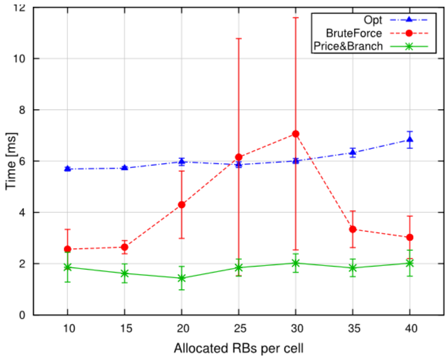


Figure 6 - Average solving times as a function of the cell load

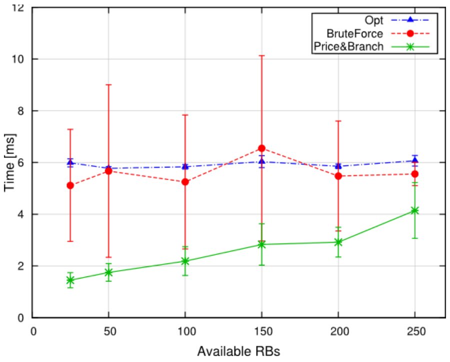


Figure 7 - Average solving times as a function of the available bandwidth

Figure 8 - Average solving time of inter-cluster coordination heuristic

Figure 9 - Optimality of CS heuristics

We then discuss the solving times of the CS algorithms and the optimality of the heuristics. Figure 5 shows the average solving time of (4) and the two heuristics based on column generation, with an increasing cluster size. Reported values are the average of measurements obtained running the CPLEX solver on ten network instances, on a machine equipped with an Intel(R) Core(TM) i7 CPU at 3.60 GHz, with 16 GB of RAM and a Linux Kubuntu 14.04 operating system. Assuming that the network manager requires CS to be run at a period of one second, (4) can be solved at optimality for cluster sizes of up to 15 cells. Larger scales can be achieved using the heuristics. In particular, it is possible to scale up to 20 and 25 nodes using the brute-force and PB approaches, respectively. While the size of (4) does not depend on either the system bandwidth M or the size of the SRs $\mathbf{A}[i]$, one may legitimately wonder whether its solution times are affected by the *values* of these parameters. However, Figure 6 and Figure 7 show that this is not the case. Figure 6 shows the CPLEX solving time for clusters of nine nodes, where coordinated cells request an increasing number of RBs, in a 10MHz-bandwidth deployment (i.e., M=50). Measured times are fairly constant. Figure 7 shows the behavior of CS with an increasing M (note that 50MHz deployments, i.e. M=250, are envisaged for 5G). The mean solving time of the CS problem remains constant, and reasonably small, in this case as well.

The solving times of the inter-cluster CS are in the range of few ms, hence the latter adds a negligible overhead on the CS problem itself. This is shown in Figure 8, which reports the average solving times of (8) as a function of both the number of clusters and their size. On one hand, solving times increase with the number of coordinated clusters. On the other hand, these depend weakly on the cluster size. Anyway, solving times are largely affordable for the intended CS periods. For instance, coordinating 19 clusters of 21 cells each requires less than 35ms. We recall that the solving time of inter-cluster CS is added to the *maximum* solving time among the coordinated clusters, since clusters can run their CS in parallel. Therefore, the above figure shows that it is actually feasible to coordinate an uber-cluster of 400 cells at sub-second periods. To the best of our knowledge, this is the first dynamic CS scheme to be tested at similar scales. To get an intuitive feeling of what this scale translates to, consider that *dense* 4G network deployments can be expected to have around 10 eNBs/km^2, and recent works anticipate *ultra-dense* 5G network to scale to 40-50 eNBs/km^2 [22]. Thus, even in dense scenarios, our CS could coordinate all the eNBs in medium- or large-sized cities, serving populations in the order of tens to hundreds of thousands. We remark that the above times have been obtained using a clever *problem formulation*, and relying on a general-purpose solver and off-the-shelf hardware. There are good reasons to believe that they could be further abated by employing ad hoc *solution algorithms* and more powerful, dedicated machines. Figure 9 shows the optimality gap of the two heuristics, up to a scale where the optimum can be computed within reasonable times. The figure shows that both are within few percentage points of the optimum, with PB faring worse when the scale increases.

Summing up, we provided a clever formulation for the CS problem, as well as three different solution strategies for it. An operator may thus choose the most appropriate strategy, trading cluster size (hence CS effectiveness, as we show in the next section) for solving time (hence reactivity).

V. Performance Evaluation

This section presents results showing the effects of CS on mean and cell-edge SINR, cell throughput, and the energy savings that it enables. Large-scale assessment requires proper tools. For this reason, we first describe the tool, and then present the results.

A. *Description of the simulation tool*

We use a flexible *snapshot simulator* that simulates the association and resulting SINR of UEs in a large-scale multicell deployment, evolved from the one in [23]. The term *snapshot* (as opposed to *discrete-event*) implies that time does not play a role here, and that the purpose of the simulator is to compute the steady-state regime given a cell deployment, a UE dropping, per-UE association rules and traffic requirements.

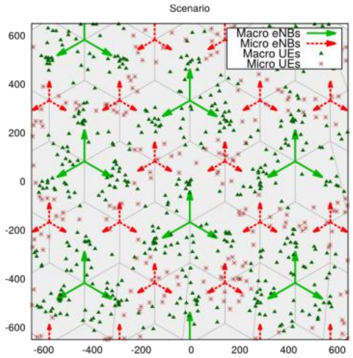


Figure 10 - Hexagon tessellation

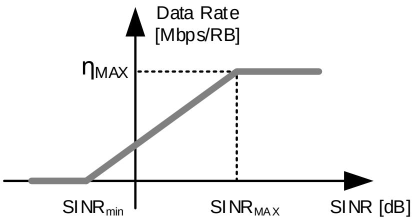


Figure 11 - Data rate as a function of SINR

Our CS simulator allows an arbitrary number of hexagons to be defined on a 2D-floorplan as a reference grid. Macro nodes are placed on the vertices of the hexagons (e.g., as shown in Figure 10) and transmit with an anisotropic pattern, whose attenuation is defined as $A(\theta) = \min\{12 \cdot (\theta/70^{\circ}), 25\}$ [25], where θ is the relative angle between the macro and the receiving UE. It is possible to simulate heterogeneous networks by placing low-power nodes (e.g., micro nodes) in the system, as shown in Figure 10. UEs can be dropped in the floorplan according to any pre-specified spatial distribution. UEs are associated sequentially, i.e., one by one, to their serving node. The latter is the macro within their hexagon, or - if micro nodes are present – the node with the highest SINR. Cell Range Expansion (CRE) can also be configured for micro nodes. For example, triangular markers in Figure 10 represent UEs associated to the macro eNB, whereas stars denote UEs associated with one of the micros.

We have already discussed that interference (hence SINR) depends on the RB occupancy at each node. Therefore, for each UE *u*, we need to compute the average RB utilization per TTI, which is obtained from its data rate D_u and its average SINR as:

$$RB_u = D_u / F\left(SINR_u^e\right). \tag{9}$$

Table 1 - Main simulation parameters

Parameter	***Value***
Inter-site distance	500 m
Carrier frequency	2 GHz
Path loss model	ITU Urban Macro [25]
UEs per hexagon	rand(25,35)
UEs deployment	Uniform
Number of snapshots	Five

In equation (9), $F\left(SINR_u^e\right)$ is the *data rate per RB* achievable by *u*, served by node *e*. The function is represented in Figure 11, where η_{MAX} is the maximum data rate that can be achieved on one RB, for values of SINR equal or above $SINR_{MAX}$. UEs whose SINR is below $SINR_{min}$ are considered out of range. The shape of the curve in Figure 11 is obtained by interpolating the results of link-level simulations of a 4G network (e.g., [24]). Parameters η_{MAX}, $SINR_{MAX}$ and $SINR_{min}$ will change with the onset of 5G cellular technologies, but the shape of the interpolated curve is unlikely to change. Note that RB_u may be non integer. This is not a problem, since RB_u is an *average* value obtained over the time span of a snapshot, which is large enough (hundreds of TTIs at least) to allow a fluid approximation.

```
1.  iteration = 0
2.  while(interfChanges or iteration > $N_{max}$)
3.     resetAllocation()
4.     for each hexagon i
5.        for each UE u in i
6.           if(iteration < N) // Association Ph.
7.              k = chooseEnb(u)
8.           else              // Convergence Ph.
9.              k = getServingEnb(u)
10.          allocateBlocks(u,k)
11.       end for
12.    end for
13.    interfChanges = updateInterference()
14.    iteration++;
15. end while
```

Figure 12 – Iterative allocation of RBs

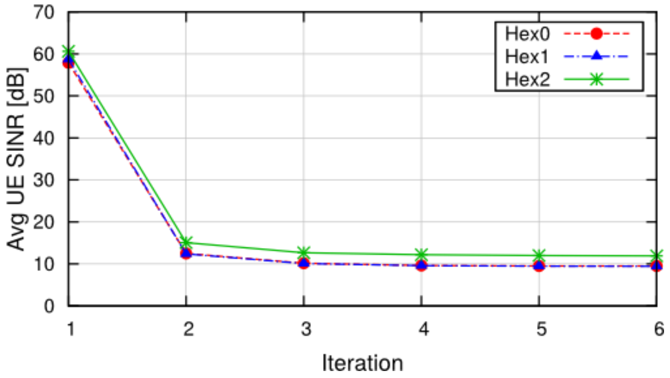


Figure 13 – Evolution of UE SINR across successive iterations in three hexagons

The alert reader will notice that, since the average SINR is computed through (1), there is a circular dependence between a UE's SINR and the RBs allocated to it. In fact, when node *e* allocates some RBs to its served UEs, $\Delta_{e,x}$ may increase for every node *x*, thus increasing the interference suffered by UEs attached to those nodes. This in turn reduces their SINR and increases their RB occupancy, and so on. This means that the average SINR must be computed *iteratively*, factoring in the varying interference of nearby cells every time, until a steady state is reached. The algorithm for doing this is shown in the pseudocode of Figure 12. The procedure is a loop repeated for up to $N_{\max}$ iterations or until convergence is reached. Each iteration cycles through every UE. We distinguish two phases:

1. *Association Phase:* for the first *N* iterations (lines 6-7) UEs are allowed to *select* the serving node, according to a best-SINR policy (line 7), possibly including cell-range expansion (CRE) biases. While doing so, the procedure also allocates RBs (line 10), according to the selected policy (e.g., FF, R, or CS). Note that, on the *first* iteration, no RBs have been allocated yet, hence the interference is null, hence the association is path loss-based rather than SINR-based. After the first iteration the interference is updated, hence the nearest node may not be the one with the best SINR anymore. This is why the *Association Phase* is repeated. However, a maximum number of re-associations has to be enforced, lest some UEs end up oscillating indefinitely between two or more serving nodes, typically when they are at cell edge.

2. *Convergence Phase:* for all the subsequent iterations, UEs do not change their serving node, and only the allocated RBs are updated according to the selected policy.

When CS is enabled, the allocation of RBs (line 10) includes the execution of the CS algorithm for all clusters. The solution of the CS problem is obtained using the CPLEX solver. In particular, CPLEX is given the number of RBs required by all nodes as an input, and returns the AMs for each node when a solution to the CS problem is obtained. At the end of each iteration, the interference is updated according to the allocation (line 13). With reference to (1), $\Delta_{e,x}$ is computed as described in Section II for FF and R. On the other hand, CS implies that $\Delta_{e,x} = \langle \mathbf{R}_e, \mathbf{R}_x \rangle$, where $\mathbf{R}_e$ and $\mathbf{R}_x$ are the AMs for nodes *e* and *x*. Moreover, the following value is computed:

$$Interf\,(n) = \sum\nolimits_u \left(\sum\nolimits_{x, x \neq servingEnb(u)} P^{RX}_{x,u} \right)$$

which represents the sum of the interference perceived by all UEs *u* from every non-serving eNB *x*, at iteration *n*. If:

$$\frac{\left| Interf\,(n) - Interf\,(n-1) \right|}{Interf\,(n-1)} > \delta\,,$$

δ being a configurable threshold, the *interfChanges* flag is set, to signal that convergence has not been reached yet. In our simulations, δ is set to 0.05, meaning that convergence is reached when interference variation is less than 5% w.r.t. the previous iteration. This condition is only tested after six iterations have been completed. Figure 13 shows how the SINR decreases with the iterations, quickly converging.

B. Simulation configuration

The simulation results presented in the following sections are obtained from a network deployment where trisectorial macro-nodes sites are placed at an inter-site distance of 500 m. The floorplan includes a total of 183 hexagons, each one served by a macro node. During the setup of each snapshot, a random number of UEs comprised between 25 and 35 are uniformly deployed within each hexagon. For simplicity, UEs request the same data rate. We vary that rate and measure their SINR according to the path loss model defined in [24] for urban scenarios. As far as data rate curves (Figure 11) are concerned, we consider $SINR_{\min} = -10dB$ and $SINR_{MAX} = 30dB$, whereas parameter η_{MAX} will be varied according to the considered technology (i.e., LTE-A or towards-5G). With reference to Figure 14, the power consumed by a node is modeled as an affine function of the number of transmitted RBs [26], i.e., $p = P_{base} + \rho \cdot n$, where P_{base} is a *baseline* power, $n \le M$ is the number of allocated RBs, and ρ is the consumption per

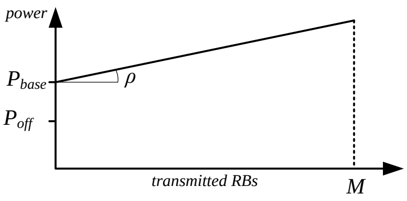


Figure 14 - Node power model

Table 2 - Power models for year 2016 (10MHz-bandwidth system)

Parameter	***Value***
Tx Power [dBm]	46
Antenna gain [dBi]	18
P_{base} [W]	279
ρ [W/RB]	15.08

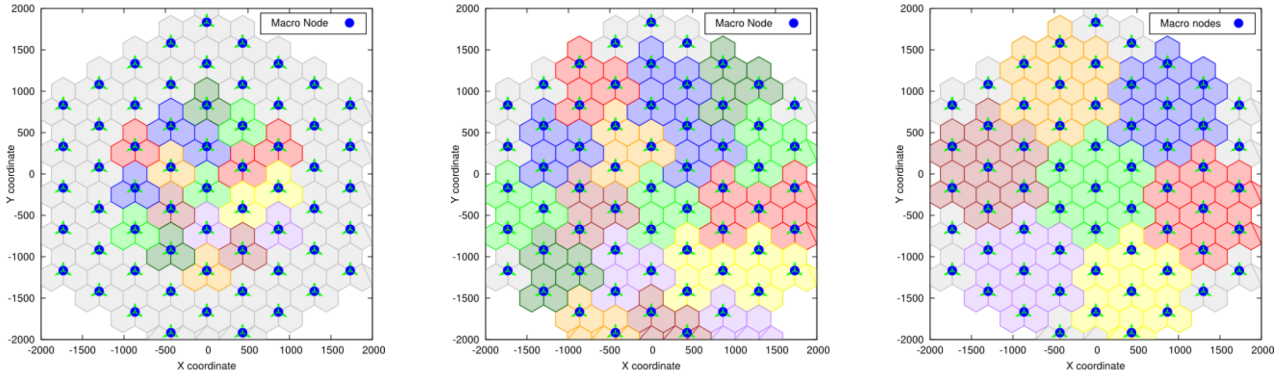


Figure 15 – Clustering of size three, nine and 21

RB. In our simulations, all nodes are active. Equipment manufacturers and operators [26] show that the *shape* of the power model has not changed so far, and it is unlikely that it will, whereas the above parameters have changed their value in the past years and will change it in the future, due to technology improvements and the onset of 5G. For this reason, we will specify the employed parameters for evaluating the consumption with both LTE-A and towards-5G technologies when needed.

In order to run the CS algorithm, we must compute interference coefficients (ICs) for each pair of nodes, according to the employed network configuration. Given two arbitrary cells *i,j*, we obtain IC $\alpha_{i,j}$ by measuring the average power received by (non-serving) cell *j* at three different locations in cell *i*, i.e. 100m of distance from the serving node, with a relative angle of -30°, 0° and 30° respectively.

Statistics are collected on the 21 central hexagons in the floorplan and on the UEs placed within them. Each measure is obtained as the average of five snapshots. A summary of the main simulation parameters is reported in Table 1.

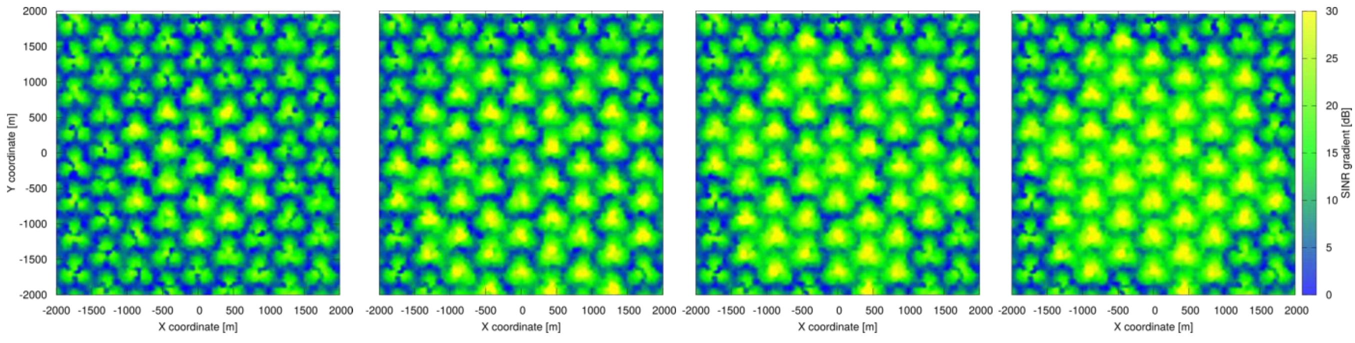
As far as CS is concerned, we use the optimal formulation of the CS for clusters up to 15 nodes. Then, we employ *bruteforce* heuristic for clusters of up to 21 nodes, whereas *price-and-branch* is used for larger clusters.

C. CS benchmarking

In this section, we assess the performance of CS at increasing scale, as well as the additional benefit deriving from inter-cluster coordination (ICC). We consider cluster sizes of three, nine and 21 (see Figure 15). When enabling ICC, we coordinate 19 clusters when the cluster size is three or nine, and seven clusters when the cluster size is 21. The remaining cells exert uncoordinated interference.

We first provide a network-wide representation of the benefits of CS from a channel-quality perspective, considering a scenario where only macro nodes are deployed, transmitting at 46dBm. Figure 16(a-d) show the distribution of SINR over the network area, obtained with an offered load of 36 Mbps per cell. The SINR value increases going from blue to yellow. Figure 16a shows the SINR when employing CS with *C*=3, i.e. coordinating cells located at the same site. UEs close to the intra-cluster borders obtain a good SINR, although interference from neighboring, uncoordinated nodes is still strong and large areas of the cells have a low SINR. In Figure 16b, CS coordinates clusters of size nine. In this case, more cell-border areas become greener. On the other hand, the improvement when scaling CS to 21 cells, shown in Figure 16c, is remarkable. Still, cluster-border areas with worse SINR are well visible. Figure 16d shows that ICC improves channel conditions for those areas as well.

Figure 17 shows the average power consumed by a node. In particular, the figure refers only to the power contribution due to the allocation of RBs, hence without considering the P_{base} terms, which only add a constant offset to the values. Without considering ICC, we can observe that power consumption decreases with the cluster size. Larger clusters imply fewer cluster-border UEs, which are those suffering most from uncoordinated interference. When ICC comes into play, interference at cluster borders is abated too and power consumed by nodes is



a) CS, cluster of 3 cells b) CS, cluster of 9 cells c) CS, cluster of 21 cells d) CS, 21-cells plus ICC

Figure 16 - SINR distribution, per-cell offered load=36Mbps

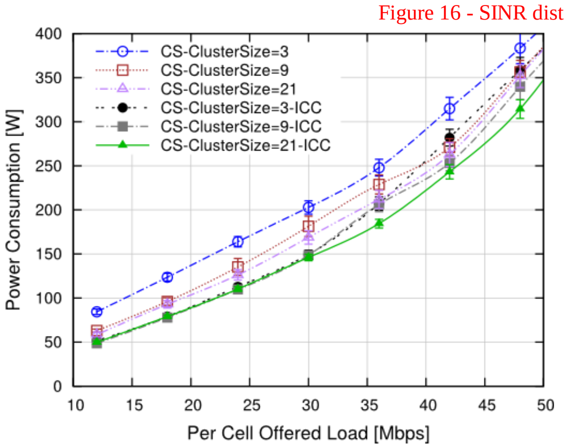


Figure 17 - Nodes' power consumption with different clustering

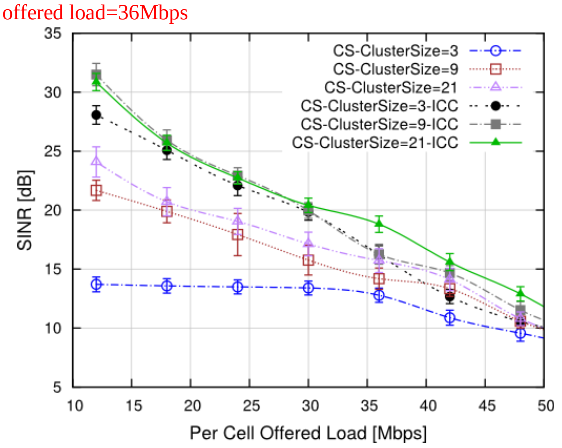


Figure 18 - Average SINR with different clustering

further reduced. At low loads, ICC levels the gaps between the performance of CS run at different cluster sizes. However, at higher loads, the increasing number of allocated RBs makes it more difficult to arrange RBs so as to minimize inter-cluster interference. This makes ICC less effective and employing larger clusters is again preferable.

The above power savings can be explained by looking at the average SINR perceived by the UEs, which is reported in Figure 18. The figure shows that larger clusters allow higher SINRs, and ICC provides additional improvements. Clearly, better channel conditions result in fewer RBs required to satisfy the same load, hence less consumed power.

Hereafter, we compare static CS schemes against our CS framework, with *C*=21 and ICC enabled.

D. Comparison against static CS schemes

We compare our CS approach against the FF and R baselines described in Section II, and against static CS schemes. We consider Frequency Reuse (FR) with reuse factors $RF = 3$ and $RF = 7$, where each cell can use $1/RF$ of the available bandwidth. FR schemes only determine which portion of the bandwidth can be used by a node. We assume that the latter allocates RBs within its portion using the *R* policy. We also simulate FFR, where the bandwidth is first halved in a cell-center and a cell-edge subbands: the former is shared among all the eNBs and used to serve UEs closer to the eNB, whereas the latter is partitioned in RF portions like FR and is reserved for cell-edge UEs.

1) Results for 4G (LTE-A) technology

In this section, we simulate a scenario that refers to the LTE-A technology, and is based on a release 10 deployment. In particular, we consider a 10MHz-bandwidth system (resulting in *M*=50 RBs) with macro nodes only. The maximum per-RB data rate is assumed to be $\eta_{MAX} = 4.5Mbps$. Power consumption is evaluated according to the parameters reported in Table 2. The latter are taken from [26] and refer to a 10MHz-bandwidth system for year 2016.

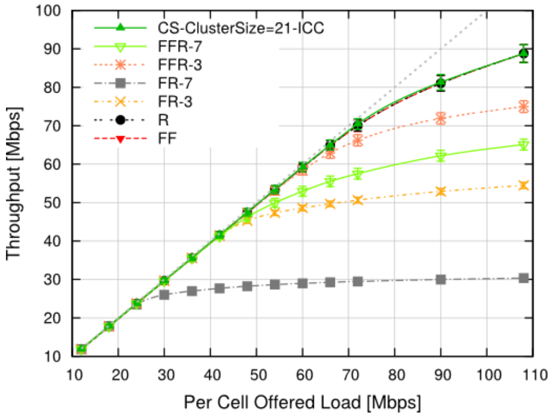


Figure 19 - Average cell throughput, LTE-A scenario

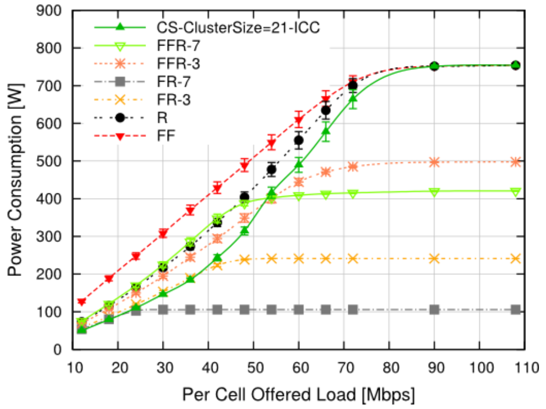


Figure 20 – Nodes' power consumption, LTE-A scenario

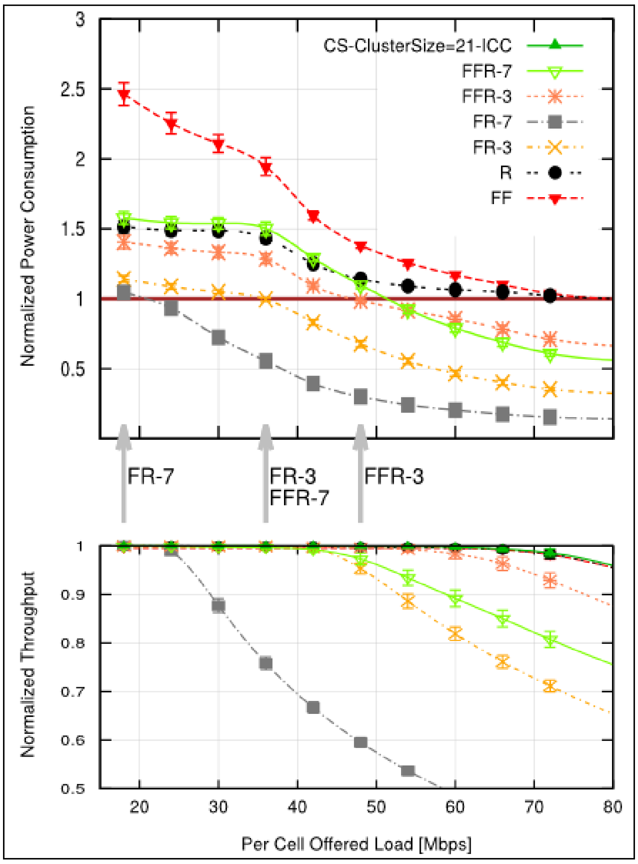


Figure 21 - Relationship between cell throughput and power consumption, LTE-A scenario

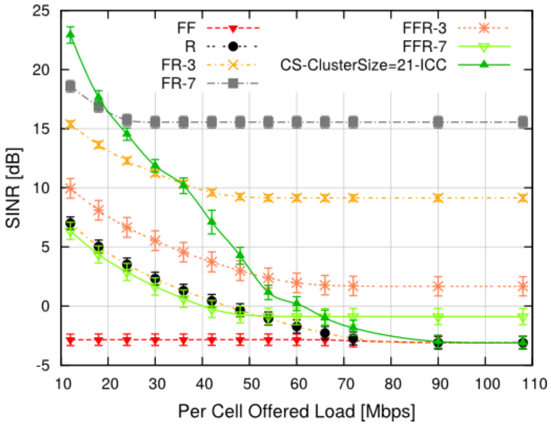


Figure 22 – 5th percentile of UEs' SINR, LTE-A scenario

Figure 19 reports the average cell throughput with increasing per-cell offered load. As the figure shows, static reuse schemes (i.e., both FR and FFR*)* saturate sooner than the others, with FFR faring better than FR for the same reuse factor[1]. In fact, reuse schemes cannot carry the offered load, even when it is quite low, since they are restricted to using only a limited portion of the available bandwidth. We remark that we simulated a scenario where UEs are uniformly distributed within the floorplan. This is the best condition for frequency reuse schemes that equally divide the bandwidth among coordinated cells. Even if the bandwidth partitioning can be done in order to accommodate a non-uniform distribution of UEs, static CS still fails to adapt to dynamic environment where UEs' position and/or datarate change at fast paces, e.g. connected vehicles on a highway. CS has the same throughput as FF and R, since CS

[1] Our simulations show that SFR fares considerably worse than FFR, hence results related to SFR are omitted for readability. This is coherent with findings in other papers, e.g. [40].

cannot bring benefits when the network is in saturation. However, they differ in terms of how efficiently the same traffic is handled. The average nodes' power consumption is reported in Figure 20. Recall that the figure refers only to the power contribution due to the allocation of RBs. The figure shows that CS consumes less power than the baselines, especially at low loads. This is because CS enhances the SINR perceived by the UEs, hence the number of RBs required to satisfy the offered load is reduced.

We now bring together the considerations for the throughput and power consumption. The top of Figure 21 shows the power consumed by baseline schemes, normalized w.r.t. the power consumed by our CS scheme with *C*=21 and ICC. The bottom of Figure 21 represents the fraction of carried load for *all* schemes (including our CS). For the reuse schemes, two regions can be distinguished, divided by a vertical arrow in the graphs: to the left, the corresponding scheme is able to carry the entire offered load, whereas to the right the throughput lags behind the offered load. We can observe that when the network is stable, the normalized power for the baselines is always *above* one, i.e. the baseline schemes consume *more* than our CS algorithm. On the other hand, (F)FR schemes *can* be more energy-efficient than our CS, but this only happens when they fall behind the offered load[2]. When the entire spectrum is allocated, i.e., with the FF and R schemes, the throughput is the same as CS's, but the power consumption can be up to 2.5 and 1.5 times higher than CS's. Note that 4G cells spend most of their time in very lightly loaded conditions, in practical deployments [31], hence schemes that allow a network to save energy when the load is low are going to make a remarkable difference in an operator's bill.

The improvements to the cell-edge UEs' channel quality are shown in Figure 22, which reports the 5th percentile of the SINR perceived by UEs with an increasing per-cell offered load. We note that, at low loads, CS improves the SINR of cell-edge UEs. At high loads, CS cannot perform better than uncoordinated schemes. In fact, the number of RBs required to satisfy the requested datarate increases and there is less space for coordination, i.e. it becomes hard to accommodate nodes' allocation so as to minimize interference. On the other hand, SINR values obtained with frequency reuse schemes stay higher than those obtained with CS at high loads. As already mentioned, this comes at the price of restricting the available bandwidth, hence achieving lower throughput.

2) *Foreseen results towards 5G*

In this section, we discuss what is expectable with the onset of 5G. 5G will evolve in at least three directions: higher data rates, due to higher-order modulations, larger spectra, and denser deployments. We keep all the above into account in the following experiments. As for data rate and spectra, we consider a configuration based on LTE-release 13, which is known under the commercial name of LTE-A Pro, and incorporates many new technologies and deployment characteristics that are foreseen to be used in 5G [41]. In particular, we assume both larger bandwidth with respect to LTE deployments, i.e. a minimum of 50MHz (*M*=250 RBs), and higher maximum per-RB data rate $\eta_{MAX} = 12 Mbps$. Moreover, it is likely that technology improvements will modify power consumption parameters. Table 3 reports the power model parameters foreseen for the year 2020 [26]. First, we evaluate a network with macro nodes only, as in the previous section. Figure 23 and Figure 24 represent the cell throughput and the power consumed by the eNBs. Figure 25 shows the 5th percentile of the UE SINR. Results are qualitatively similar to those in the previous section, albeit with a different scale. This supports our claim that increased data rates and larger spectra will not decrease the benefits of our CS framework.

Since the upcoming 5G technology will make use of dense deployment of heterogeneous cells to increase the system capacity, we now assess the performance of the CS algorithm in heterogeneous networks with different densities of small, low-power nodes. The latter introduce additional interference in the network, which needs to be managed. On the other hand, they can offload some capacity from the macro node. Association of UEs is performed according to a best-SINR criterion. In order to facilitate load balancing, we assume that low-power nodes exploit a CRE of 6 dBi. Our first heterogeneous scenario is reported in Figure 10, where two micro nodes per cell are placed on vertices of the hexagons. Since each macro comes with two additional micros, static reuse schemes, i.e. FR and FFR, employ reuse factors $RF = 3$ and $RF = 9$. For example, in FR with $RF = 3$, the available bandwidth is divided among the macro and the micro nodes of the same hexagon. For the same reason, we employ CS on clusters composed of nine hexagons, resulting in 27 nodes to be coordinated. Again, considering the solving times of the CS for large clusters, the price-and-branch approach is employed.

[2] The only exception appears to be for FR 3, with an offered load of 42Mbps, where the power consumption ratio is below 1. This is due to several factors: first of all, the power normalization is done w.r.t. the sample mean of CS's power consumption, which has statistical fluctuations. Second, at *C*=21, CS is being done suboptimally through brute-force. Third, once again, this is the best-case scenario for FR schemes, with uniform traffic and little differences from one cell to the next.

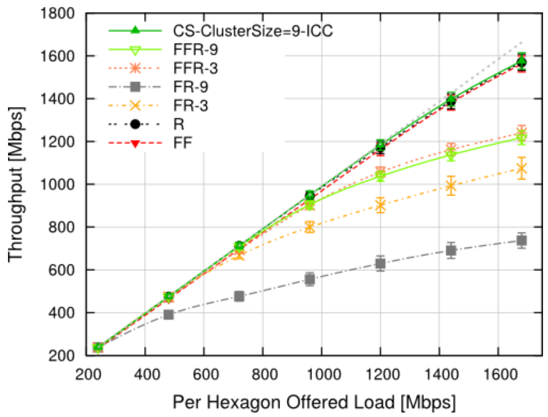


Figure 26 – Average per-hexagon throughput, foreseen 5G scenario, micros enabled

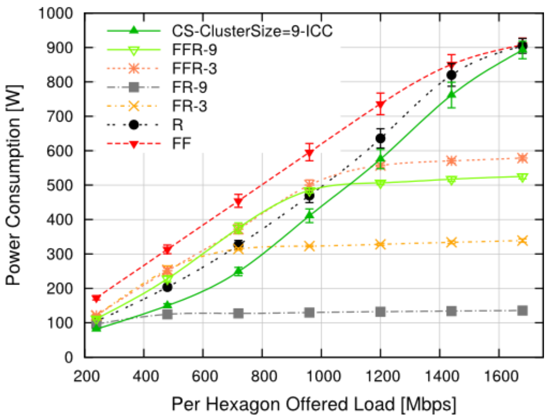


Figure 27 – Nodes' power consumption, foreseen 5G scenario, micros enabled

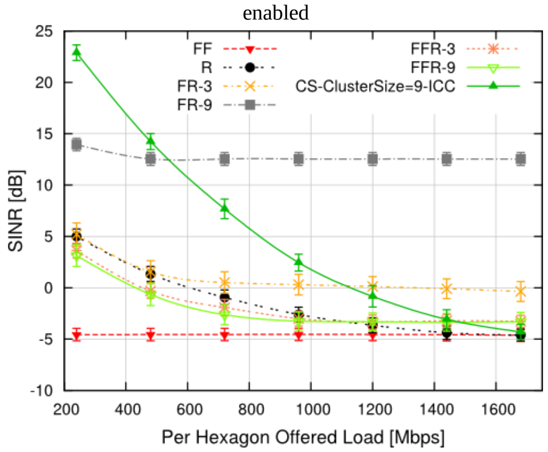


Figure 28 - 5th percentile of UE SINR, foreseen 5G scenario, micros enabled

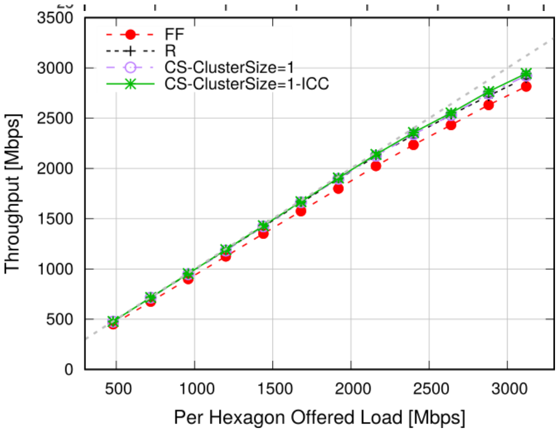


Figure 29 – Average per-hexagon throughput, dense deployment of pico nodes

The average cell throughput is reported in Figure 26. With respect to the macro-only scenario, micro nodes allow the network to postpone the saturation point of the cells. Nodes' power consumption is reported in Figure 27. In this case too CS allows the operator to save frequency resources and consume less power. For instance, considering an offered load of 720Mbps, CS with ICC saves about 75W per hexagon with respect to the R scheme, and about 200W with respect to the FF one. Figure 28 shows the 5th percentile of the SINR perceived by the UEs. By comparing this with Figure 25, we observe that adding micros actually improves the cell-edge throughput when CS is enabled, but it reduces it with (F)FR schemes. This is because reuse has to take into account micros as well, hence the distance between nodes (notably, macros) using the same bands is reduced for the same reuse factor.

We now evaluate our CS framework in a scenario where each macro node comes with 15 pico nodes randomly placed in the corresponding hexagon, resulting in a density of about 40 nodes/km^2 [22]. Pico cells are randomly deployed within the hexagon and transmit at 21dBm. Parameters for evaluating the power consumption of pico nodes are summarized in Table 3. In this case, a CS cluster is composed of the macro and its embedded picos, whereas ICC is done among single-macro clusters. Given the irregular deployment of nodes, static frequency reuse schemes, i.e. FR and FFR, can hardly be applied. In fact, a suitable frequency reuse plan cannot be done in dense deployments where distance between nodes is not uniform. Thus, we compare CS with and without ICC, against FF and R. Figure 29 and Figure 30 report the average per-hexagon throughput and the nodes' power consumption, respectively. Also in this case, CS achieves similar throughput as FF and R, albeit consuming less power. In such a dense scenario ICC largely contributes to improving the performance. In fact, macro nodes in adjacent hexagons are not part of the same cluster, hence their allocations are not coordinated when ICC is disabled and they would exert high interference. This is made more evident by the cell-edge SINR shown in Figure 31.

VI. Conclusions

In this paper, we have investigated how coordinated scheduling (CS) improves network performance, i.e. it allows a network to carry the same traffic employing fewer resources, and protects cell-edge users from excessive interference. To show this, we have first devised optimization models that *can* be solved in clusters of few tens of nodes in a sufficiently short time as to match the dynamics of current and future cellular

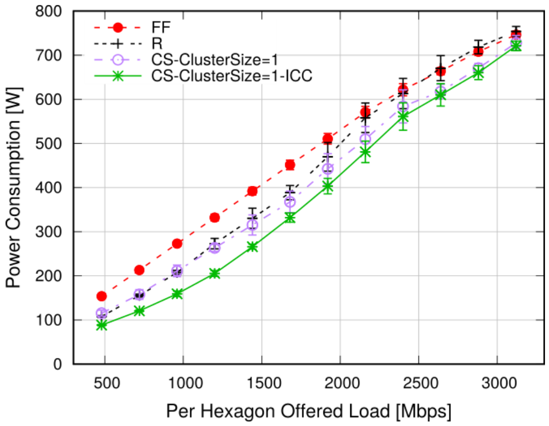


Figure 30 – Nodes' power consumption, dense deployment of pico nodes

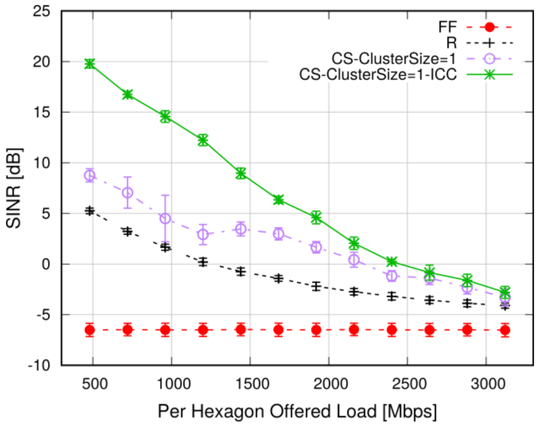


Figure 31 - 5th percentile of UE SINR, dense deployment of pico nodes

networks. Then, we have shown that clusters can be subject to a further level of inter-cluster coordination, to improve the conditions of cluster-border UEs, with a little extra overhead. This allows a network operator to coordinate up to 400 cells at sub-second timescales, with off-the-shelf hardware.

Using a large-scale snapshot simulator, we have shown that the above-mentioned benefits actually increase with the scale of coordination, up to the maximum allowed by our models, which confirm that there is a need to scale coordination up. The energy-efficiency benefits will be even more tangible in the near future, when next-generation base stations will be around, whose power consumption depends more on the number of allocated RBs. Moreover, the near future will witness *heterogeneous* and *denser* deployments, with both *macro* and *micro* (or pico) cells. In this case, protecting micro cells from the interference of the macros will be the key to reaping the benefits of having spatially-localized high-bandwidth hotspots. We have shown that these deployments benefit from our CS scheme as well.

The work reported in this paper can be extended in several directions. First, devising *optimal clustering algorithms*, so as to maximize the gains of CS given a maximum cluster size (or, equivalently, a constraint on the solving time of the optimization problem). This is especially important when the cell layout and antenna radiation pattern is irregular, as happens in practical deployments (and all the more with heterogeneous deployments). Second, devising *optimal power saving algorithms*, that leave the minimum set of nodes powered on for a given traffic demand, *assuming optimal CS is in place* within and/or among clusters. In fact, the resource saving obtained through CS may well translate to a *smaller* number of active nodes required for a given traffic demand (see, e.g. [32], [33]). This would further enhance the energy efficiency of the network.

ACKNOWLEDGMENT

This work was partially supported by the European Commission in the framework of the H2020-ICT-2014-2 project Flex5Gware (grant agreement no. 671563). The authors gratefully acknowledge the partial financial support from the University of Pisa, projects "*Modelli ed algoritmi innovativi per problemi strutturati e sparsi di grandi dimensioni*" (grant PRA_2017_05), and "*IoT e Big Data: metodologie e tecnologie per la raccolta e l'elaborazione di grosse moli di dati*" (grant PRA_2017_37).

The authors are grateful to Emanuele Guerrazzi, MSc of the University of Pisa, for his help with coding the optimization models.

REFERENCES

[1] G. Nardini, *et al*., "Scalability and energy efficiency of Coordinated Scheduling in cellular networks towards 5G", in proc. CLEEN 2017, Turin, June 21-22, 2017

[2] L. Fang, X. Zhang, "Optimal Fractional Frequency Reuse in OFDMA Based Wireless Networks", Proc. WiCOM '08, pp.1-4, 12-14 Oct. 2008.

[3] K. Hoon, H. Youngnam, J. Jayong, "Optimal subchannel allocation scheme in multicell OFDMA systems", Proc. of VTC Spring'04 pp.1821-1825 Vol.3, 17-19 May 2004.

[4] A. Douik, H. Dahrouj, T.Y. Al-Naffouri, M.S. Alouini, "Coordinated scheduling for the downlink of cloud radio-access networks", 2015 IEEE International Conference on Communications (ICC), London, 2015.

[5] R. Y. Chang, Z. Tao, J. Zhang, C.C. Kuo, "A Graph Approach to Dynamic Fractional Frequency Reuse (FFR) in Multi-Cell OFDMA Networks", 2009 IEEE International Conference on Communications (ICC), Dresden, 2009.

[6] X. Wang, B. Mondal, E. Visotsky, A. Ghosh, "Coordinated scheduling and network architecture for LTE Macro and small cell deployments", 2014 IEEE International Conference on Communications Workshops (ICC), Sydney, 2014.

[7] J. Liu, Y. Chang, Q. Pan, X. Zhang, D. Yang, "A Novel Transmission Scheme and Scheduling Algorithm for CoMP-SU-MIMO in LTE-A System", 2010 IEEE 71st Vehicular Technology Conference, Taipei, 2010.

[8] G. Nardini, G. Stea, A. Virdis, D. Sabella, M. Caretti, "Practical large-scale coordinated scheduling in LTE-Advanced networks", Wireless

Networks, DOI: 10.1007/s11276-015-0948-6, Volume 22, Issue 1, pp. 11-31, January 2016

[9] G. Accongiagioco, M. Andreozzi, D. Migliorini, G. Stea, "Throughput-optimal Resource Allocation in LTE-Advanced with Distributed Antennas" Computer Networks, vol. 57(2013), pp. 3997-4009, Dec. 2013

[10] Deliverable D1.1, Refined Scenarios and requirements, consolidated use cases, and qualitative techno-economic feasibility assessment, METIS-II Project, 31 Jan 2016

[11] Flex5Gware website: http://www.flex5gware.eu (accessed Apr. 2017)

[12] METIS II website: https:// metis-ii.5g-ppp.eu (accessed June 2017)

[13] N. Iardella, *et al.*, "Flexible dynamic Coordinated Scheduling in Virtualized-RAN deployments", FlexNets 2017, Paris, FR, 20th May 2017

[14] N. Iardella, *et al.* “A testbed for flexible and energy-efficient resource management with virtualized LTE-A nodes”, proc. of CLEEN 2017, Turin, Italy, 21-22 June 2017.

[15] R. Wang, *et al.*, "OpenAirInterface - An effective emulation platform for LTE and LTE-Advanced", Proc. ICUFN 2014, Shanghai, pp. 127–132.

[16] N. Iardella, *et al.*, "Statistically sound experiments with OpenAirInterface Cloud-RAN prototypes", Proc. of CLEEN 2016, Grenoble, FR, May 2016

[17] L. Pitsoulis "Quadratic Semi-assignment Problem", in Encyclopedia of Optimization, C.A. Floudas and P.M. Pardalos eds., p. 3170-3171, Springer, 2009

[18] ILOG CPLEX Software, http://www.ilog.com

[19] G. Desaulniers, J. Desrosiers and M.M. Solomon, “Column Generation”, Springer, 2005

[20] S. Burer and A. N. Letchford “Non-convex mixed-integer nonlinear programming: a survey”. Surveys in Oper. Res. and Mgmt. Sci., 17(2), 97-106, 2012

[21] H.W. Khun, “The Hungarian method for the assignment problem”, *Naval Research Logistic Quarterly*, vol.2, pp. 83-97, 1955

[22] X. Ge, S. Tu, G. Mao, C. X. Wang and T. Han, "5G Ultra-Dense Cellular Networks," in IEEE Wireless Communications, vol. 23, no. 1, pp. 72-79, February 2016. doi: 10.1109/MWC.2016.7422408

[23] A. Virdis, G. Stea, D. Sabella, M. Caretti, "A practical framework for energy-efficient node activation in heterogeneous LTE networks", *Mobile Information Systems*, vol. 2017, Article ID 2495934, June 2017, doi:10.1155/2017/2495934

[24] C. Mehlfhrer, M. Wrulich, J. C. Ikuno, D. Bosanska, M. Rupp, "Simulating the long term evolution physical layer," in Proc. 17th EUSIPCO, pp. 1471-1478, 2009

[25] 3GPP TR 36.814 v9.0.0, "Further advancements for E-UTRA physical layer aspects (Release 9)", March 2010

[26] D. Sabella, *et al.,* “Energy Management in Mobile Networks Towards 5G”, in M.Z. Shakir et al. (eds.), Energy Management in Wireless Cellular and Ad-hoc Networks, Studies in Systems, Decision and Control, Springer, DOI 10.1007/978-3-319-27568-0_17, 2016

[27] F. Qamara, *et al.*, "A comprehensive review on coordinated multi-point operation in LTE-A", Computer Networks, vol. 123, pp. 19-37, August 2017.

[28] K. Yamaguchi, S. Masuda, "A new exact algorithm for the maximum weight clique problem", in Proc. of 23rd ITC-CSCC 2008, Yamaguchi, JP, 2008

[29] J. Crawford, G. Luks, M. Ginsberg, A. Roy, “Symmetry breaking predicates for search problems”, in 5th Int. Conf. on Knowledge Representation and Reasoning, (KR ’96). (1996) 148–159.

[30] A. H. Land, A. G. Doig, "An automatic method of solving discrete programming problems". Econometrica. 28 (3). pp. 497–520, 1960. doi:10.2307/1910129.

[31] F. Xu, Y. Li, H. Wang, P. Zhang, D. Jin, "Understanding Mobile Traffic Patterns of Large Scale Cellular Towers in Urban Environment," in *IEEE/ACM Transactions on Networking*, vol. 25, no. 2, pp. 1147-1161, April 2017. doi: 10.1109/TNET.2016.2623950

[32] T. Sigwele, A. S. Alam, P. Pillai, Y. F. Hu, “Energy-efficient cloud radio access networks by cloud based workload consolidation for 5G”, *Journal of Network and Computer Applications*, Vol. 78, 15 Jan. 2017, pp. 1-8

[33] S. S. Soliman, B. Song, “Fifth generation (5G)cellular and the network for tomorrow: cognitive and cooperative approach for energy savings”, *Journal of Network and Computer Applications*, Vol. 85, 1 May 2017, pp. 84-93

[34] NGMN, "RAN evolution project CoMP evaluation and enhancement," Deliverable, Version 2.0, 31-March-2015

[35] L. Liu, V. Garcia, L. Tian, Z. Pan, "Joint clustering and inter-cell resource allocation for CoMP in ultra dense cellular networks," Proc. of IEEE ICC 2015, June 2015, pp:2560-2564.

[36] S. Bassoy, M. Jaber, M.A. Imran, P. Xiao, "Load Aware Self-Organizing User-Centric Dynamic CoMP Clustering for 5G Networks", *IEEE Access*,vol.4, 2016, pp.2895-2906.

[37] A. Marotta, *et al.*, "Impact of CoMP VNF placement on 5G Coordinated Scheduling performance", Proc. of EuCNC 2017, Oulu, 2017, pp. 1-6.

[38] S. Bassoy, M.A. Imran, A. Imran, “Coordinated Multi-Point Clustering Schemes: A Survey”, *IEEE Communications Surveys and Tutorials*, Vol 19, No. 2, 2nd quarter 2017, pp. 743-764.

[39] O.-D. Ramos-Cantor, J. Belschner, G. Hegde, M. Pesavento, “Centralized coordinated scheduling in LTE-Advanced networks”. *EURASIP Journal on Wireless Communications and Networking*, Dec. 2017

[40] T. Novlan, et al., "Comparison of Fractional Frequency Reuse Approaches in the OFDMA Cellular Downlink,", proc. of IEEE GLOBECOM 2010, Miami, FL.

[41] Qualcomm, “The essential role of Gigabit LTE & LTE Advanced Pro in a 5G World”, https://www.qualcomm.com/documents/essential-role-gigabit-lte-lte-advanced-pro-5g-world, accessed November 2017.