This paper appeared at CLEEN 2017, Turin, June 22 2017. The proceedings are published by IEEE

Scalability and energy efficiency of Coordinated Scheduling 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, G.M. Dell'Aera
TIM (Telecom Italia Group),
via G. Reiss Romoli 274,
Turin, Italy

***Abstract*— Coordinated Scheduling (CS) is one of the main techniques to control 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 in each node's subframe with the least possible overlapping with neighboring nodes. We provide a clever formulation, which allow optimal solutions to be computed in clusters of ten nodes, and algorithms that compute good suboptimal solutions for clusters of several 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 show that optimal CS allows a significant protection of cell-edge users. Moreover, this goes hand-in-hand with a significant reduction in the number of allocated Resource Blocks, which in turn allows an operator to reduce its energy consumption. Both benefits actually *increase* with the size of the clusters.**

Keywords—CoMP-CS, energy-efficiency, scheduling, mobile networks, optimization, simulation

I. Introduction

Inter-cell Interference (ICI) is one of the major causes of performance degradation in 4G cellular networks, where all neighboring cells share the same spectrum. 5G networks, if anything, will be denser and with higher traffic demands, which will only exacerbate the problem. User equipments (UEs) suffering interference from nearby eNodeBs (eNB) 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 Blocks (RBs), to serve these UEs. Thus, the network will be able to carry less traffic, and consume more energy – which is proportional to the number of RB allocated per Transmission Time Interval (TTI) – to carry the same traffic.

A technique 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]-[4]), resource partitioning 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. An example is a single UE roaming through unloaded neighboring cells, no one of which is able to allocate it its full bandwidth. On the other hand, *dynamic* CS schemes have been proposed, e.g., [5]-[8]. Some of these are not standard-compliant, since they assume that the eNBs know information which is not available in the current standards: for example, that UEs can report their channel coefficients. Some dynamic schemes (e.g., [7],[8]) assume instead that a central entity managing a cluster of cells 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 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 the ensuing optimization problems cannot be solved within a TTI [8].

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 will thus constrain the scheduling decisions for a whole period. This is the approach pursued in this paper, designed and prototyped within the framework of the Flex5Gware EU-5GPPP project [1]. A *global scheduler*, coordinating a *cluster* of nodes, runs CS on every *period* of 100-1000s 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. Single nodes are still in charge of per-TTI scheduling, keeping the complexity manageable, and they periodically record and send to the global scheduler the number of RBs that they need to carry all their traffic. A period in the range of 100-1000s 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 and solved at optimality, at fairly large scales (up to several tens of nodes). While the natural way to formulate a CS problem would be as a Quadratic Assignment Problem (QAP), which is inefficient, we devise instead a non-intuitive *pattern-based* formulation. The ensuing Mixed-Integer Linear Program (MIILP) can be solved at optimality in hundreds of milliseconds at scales of up to ten nodes. Larger scales can be reached by adopting heuristic techniques, such as price-and-branch, where column generation can be handled in different ways, e.g., brute-force enumeration or using a general-purpose solver.

The benefits of our optimal dynamic CS are twofold: on one hand, cell-edge UEs are protected from the interference of nearby cells. We show that their SINR increases remarkably when CS is activated. On the other hand, protecting cell-edge UEs actually *frees* a considerable amount of RBs at the nodes,

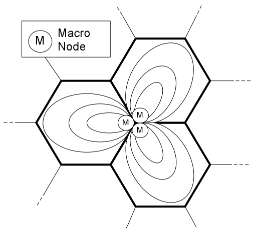


Figure 1 – Nodes in a hexagon tessellation.

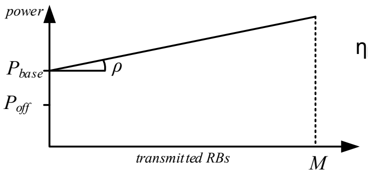


Figure 2 - Node power model.

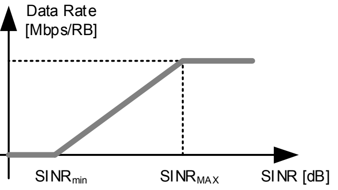


Figure 3 - Data rate vs. SINR.

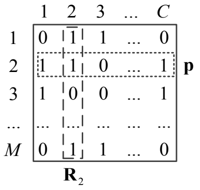


Figure 4 – Allocation masks (columns) and ownership vectors (rows)

namely those that would otherwise be employed to serve UEs with poorer channel conditions. This, in turn, increases the number of bits per RB in the whole network, making it more energy-efficient, and leading to reduced power consumption. We are thus able to assess the above benefits at scales at which, to the best of our knowledge, they have not been assessed so far. Both benefits actually *increase* with the coordination scale, all the way up to the maximum allowed by our algorithms. This calls for further research in the direction of increasing the CS scale. The above benefits are evident when the system is compared to both an uncoordinated cellular network *and* one with a static coordination, e.g. a frequency reuse of three or seven.

The rest of the paper is organized as follows: Section II describes the system model and states the problem. In Section III we describe our CS models, and in Section IV we evaluate their performance. Section V concludes the paper.

II. System model and problem statement

Transmissions in a cellular network are arranged in time slots called Transmission Time Intervals, (TTIs), whose duration is 1ms in the current LTE standard. We only consider the downlink (DL) direction, the most critical from the point of view of both carried load and infrastructure power consumption. In a TTI, nodes allocate *subframes* (SFs), i.e. vectors of RBs to its associated UEs. Each RB carries a fixed number of symbols, which translate to different amounts of bits depending on the quality of the air channel, i.e. on the SINR that UEs perceive. This varies over both time and frequency, hence UEs report a quantized indication of their SINR (called a Channel Quality Indicator, CQI) to the eNBs periodically. UEs are associated to *one* eNB at a time, and their association lasts longer than CS periods (minutes or more). In the DL, all the nodes share the same spectrum, hence interfere with each other.

We consider a large-scale multicell cellular network, a portion of which is shown in Figure 1. Cells are represented as *hexagons* for simplicity, and they host a *node* that covers them. UEs associate to the node from which they perceive the highest SINR. Figure 2 shows the power consumed by an active node, which is an affine function of the number of allocated RBs on each TTI, i.e., $p = P_{base} + \rho \cdot n$, where P_{base} is a *baseline* power, and $n \le M$ is the number of RBs. P_{off} is the power consumed by a switched-off node. Work [15] shows that, while the above parameters have changed and will be changing their values over the years, due to technology improvements and the onset of 5G, the *shape* of the power model has not changed so far, and it is unlikely that it will.

A UE u requests a certain *data rate* D_u from its serving node e. The number and data rates of UEs can be inferred from *load curves*, detailing how these two values evolve over time in each cell, with a time resolution of minutes or more. A method is required to infer the average RB occupancy per TTI (which in turn determines the power consumption) from the required data rate and SINR of a UE. The RBs allocated per UE are computed as follows. The *data rate per RB* achievable by u is a function of its *average* SINR, $F\left(SINR_u^e\right)$, represented in Figure 3. η_{MAX} is the maximum data rate that can be achieved for values of SINR equal or above $SINR_{MAX}$. UEs whose SINR is below $SINR_{min}$ are considered out of range. Again, parameters η_{MAX}, $SINR_{MAX}$ and $SINR_{min}$ will probably change with the onset of 5G cellular technologies, but the shape of the curve is likely to remain the same [15]. Thus, the average number of RBs that satisfy u 's demand are:

$$RB_u = D_u \big/ F\left(SINR_u^e\right). \tag{1}$$

Note that RB_u may not be integer. This is not a problem, since RB_u is an *average* value, and – as we will see later on – the time span of a CS period is large enough (hundreds of TTIs at least) as to allow a fluid approximation. For instance, an allocation of $RB_u = 2.5$ RBs per TTI can be enforced in practice by reserving two RBs on even TTIs and three on odd ones, etc.

In order to compute $SINR_u^e$, we need to quantify the signal received by u from node e and the interference from nodes $x \ne e$. Call n_e and n_x the number of RBs allocated by node e and x respectively, and call $\Delta_{e,x}$ the number of *overlapping* RBs in the two allocations. This quantity depends on the *allocation scheme* employed at each node. For instance:

- *First fit (FF)*: all nodes allocate RBs starting from the first position. This way, the overlapping RBs are the maximum possible, i.e. $\Delta_{e,x} = \min\left(n_x, n_e\right)$. Despite this being the most inefficient strategy from an interference standpoint, it is what happens in practice in several cases. For instance, OpenAirInterface nodes [11] actually do this.
- *Random (R)*: a node selects its RBs at random. This way, after some straightforward computations, $\Delta_{e,x} = n_x \cdot n_e \big/ M$, i.e., smaller than with FF, especially at *low* network loads.

The *probability* that x will interfere on a RB allocated by e is $\Delta_{e,x} \big/ n_e$. 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). Assuming that the scheduler at node e selects the RBs to be allocated to u randomly among its n_e RBs, u 's *average* SINR is:

$$SINR_u^e = \frac{P_{e,u}}{N_G + \sum_{x \ne e} P_{x,u} \cdot \Delta_{e,x} \big/ n_e}, \tag{1}$$

where N_G is the Gaussian noise. Expression (1) 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 aim of CS, of course, is to reduce $\Delta_{e,x}$ for couples 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 maximize the sum of the average SINR across all UEs. Besides the non-trivial *scalability* problem (nodes may handle hundreds of UEs in practical cases), and *modeling* difficulties (formula (1) is nonlinear and nonconvex in variables $\Delta_{e,x}$ and n_e), there is a stronger, practical impediment: UEs *do not report* the necessary information, i.e. the received powers $P_{x,u}$. Their reporting is limited to the CQI value, without any indication of *how* that number was obtained. Thus, (1) cannot be computed, except at the UEs themselves.

III. Optimization Approach to Coordinated Scheduling

The approach followed within Flex5Gware [1] adopts a different perspective, which does away with all the above problems. The basic philosophy 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 node itself. The latter communicates with a *Global Scheduler (GS)*, that coordinates scheduling in a *cluster* of cells. Nodes in a cluster send *Scheduling Requests* (SR) to the GS, which state the number of RBs required to clear the node's backlog, averaged over a *period* of *T* TTIs (e.g., hundreds or more). In turn, the GS sends back to each node i an *Allocation Mask* (AM) on each period. This, 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. The minimum period, determined by the time that the GS employs to compute AMs for its cluster, will depend on the size of the cluster. Hereafter, we describe several solutions, which achieve different tradeoffs between optimality and scale.

A. 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 symmetric, since cells are 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 simple, but inefficient formulation of the CS problem is the following:

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

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 (*i*) forces the sum of RBs allocated to cell *i* to be at least equal to its SR A_i. Note that equality will hold in (*i*) at the optimum in any case, since this is a minimization problem. Coupled with the fact that problem variables are binary (constraint (*ii*)), this makes problem (2) a slight variant of the Quadratic Semi-Assignment Problem (QSAP) [16], which is notoriously hard to solve at optimality, in large part due to its nonlinear objective function. It *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:

$$\begin{aligned} &\min \textstyle\sum_{i,j}\alpha_{i,j}\cdot\sum_{x=1}^{M}\mathbf{O}_{i,j}[x] \\ &s.t. \\ &\mathbf{O}_{i,j}[x]\geq \mathbf{R}_i[x]+\mathbf{R}_j[x]-1 \quad i,j\in\mathbf{C}, j\neq i, 1\leq x\leq M \quad (i') \\ &\ldots \end{aligned}$$

Constraint (*i'*) linearizes the logical AND between $\mathbf{R}_i[x]$ and $\mathbf{R}_j[x]$, and the rest stays as in (2). Introducing overlap vectors, however, brings 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 the size, another major disadvantage is *symmetry*: any permutation of the rows of the matrix in Figure 4 yields the same objective, which makes it much harder to solve the model at optimality. 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 SF. In fact, only the former determines inter-cell interference.

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

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

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

$$\begin{aligned} &\min \textstyle\sum_{\mathbf{p}\in\mathbf{P}}c_\mathbf{p}\cdot x_\mathbf{p} \\ &s.t.\quad \textstyle\sum_{\mathbf{p}\in\mathbf{P}}\mathbf{p}[i]\cdot x_\mathbf{p}\geq\mathbf{A}[i]\quad i\in\mathbf{C}\quad (i) \\ &\qquad \textstyle\sum_{\mathbf{p}\in\mathbf{P}}x_\mathbf{p}\leq M \qquad (ii) \\ &\qquad x_\mathbf{p}\in\mathbb{N}\qquad \mathbf{p}\in\mathbf{P}\quad (iii) \end{aligned} \qquad (3)$$

The objective, though formulated differently, is equal to the previous problem's. Constraint *(i)* states that the number of RBs in the AM to a node must not fall below its SR, whereas constraint *(ii)* caps the number of allocated RBs to the maximum *M*. The variables are $x_\mathbf{p}$, are integer, and there are 2^C of them, hence this is an Integer Linear Program (ILP). This ILP is solvable at optimality by a general-purpose solver (such as CPLEX, [9]) in split-second times for clusters up to 10 cells, i.e. $2^{10}=1024$ patterns. Once (3) is solved, the AMs can be found by placing $x_\mathbf{p}$ instances of each row $\mathbf{p}$ in *any* order. While (3) is considerably faster than (2) (or its linearized version) at medium scales (e.g., ~10 nodes), it goes without saying that its size is still $O(2^C)$, hence solving times become prohibitive at larger scales. Thus, we now discuss other algorithms which trade a little optimality for an increase in scale.

B. Trading optimality for scalability

ILP (3) can be solved to optimality in split-second times for small clusters. The standard solution algorithms for ILPs is *branch-and-bound*, which consists in iteratively solving the continuous relaxation, i.e., the model obtained by relaxing integrality constraints *(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*) [13]. 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 (*i*) and (*ii*) in (3). The dual problem reads as follows:

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

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:

$$\min\left\{\mathbf{p}^T \cdot \boldsymbol{\alpha} \cdot \mathbf{p} - \boldsymbol{\lambda}^* \cdot \mathbf{p} : \mathbf{p} \in \mathbf{P}\right\}, \qquad (4)$$

We start by observing that the interference matrix $\boldsymbol{\alpha}$ is non negative, therefore pricing problem (4) 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 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 $\tilde{\boldsymbol{\alpha}} = \{\tilde{\alpha}_{i,j}\}$, where $\tilde{\alpha}_{i,j} = \alpha_{i,j}$ if $i \ne j$ and $\tilde{\alpha}_{i,i} = -\lambda_i^*$, and rewrite (4) as:

$$\min\left\{\mathbf{p}^T \cdot \tilde{\boldsymbol{\alpha}} \cdot \mathbf{p} : \mathbf{p} \in \mathbf{P}\right\}, \qquad (5)$$

Problem (5) is an Unconstrained Boolean Quadratic Problem (UBQP) [12], and it is known to be NP-hard. In order to solve a continuous relaxation of (3), we need to solve (5). 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 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 up to 20 variables or so. From that scale onward, solving the QP will generally 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 guarantees 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 (3) (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.

Figure 5 shows the average solving time of (3) 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 12.04 operating system. Assuming that CS must be run at periods below 1s, (3) can be solved at optimality for cluster sizes of up to 15 cells. Larger clusters can be coordinated using the heuristics. In particular, it is possible to scale up to 20 and 25 nodes using the brute-force and PB approaches, respectively.

Given that autonomous CS instances are run at each cluster, cluster borders are subject to uncoordinated interference, hence cluster-border UEs will still have a worse SINR. The alert reader will have noticed that a pattern-based modeling of CS leaves open the problem of *placing* RBs within a SF. In theory, the $x_{\mathbf{p}}$ instances of $\mathbf{p}$ in a solution can be placed at any of the M positions in the SFs of the nodes in a cluster. This can be leveraged to improve conditions at cluster borders. We do this by leveraging a heuristic proposed in [8], which formulates the problem as a (polynomial) Linear Assignment Problem, that can be solved in few ms. using the Hungarian algorithm [14].

IV. Performance Evaluation

Macro nodes are placed on the vertices of hexagons (e.g., as shown in Figure 1) in a 2D floorplan and transmit at 36 dBm with an anisotropic pattern, whose attenuation is defined as $A(\theta) = \min\left\{12 \cdot \left(\theta / 70^\circ\right), 25\right\}$, where θ is the angle between the macro and the receiving UE. ICs $\alpha_{i,j}$ are computed by measuring the average power received by (non-serving) cell *j* at three different locations in cell *i* (100m of distance from the serving antenna, and an angle of -30°, 0° and 30° respectively). 30 UEs per hexagon are dropped uniformly in the floorplan and are associated sequentially, i.e., one by one, to the node serving that hexagon. The mean SINR of each UE is computed through (1). Note that there is a circular dependence between a UE's SINR and the RBs allocated to it: when a UE is associated to a node *e* and is allocated some RBs, in fact, $\Delta_{e,x}$ may increase, thus increasing the interference of UEs attached to *x* and reducing their SINR. This in turn increases their RB demand, and so on. This means that the average SINR must be computed *itera-*

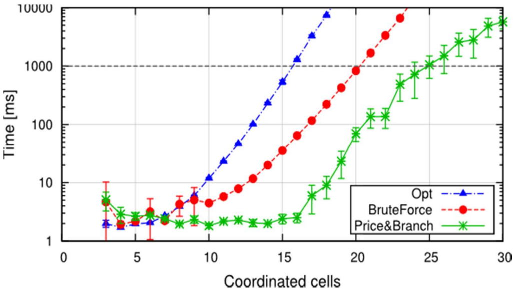


Figure 5 - Average solving times vs. cluster size

Table 1 - Main simulation parameters

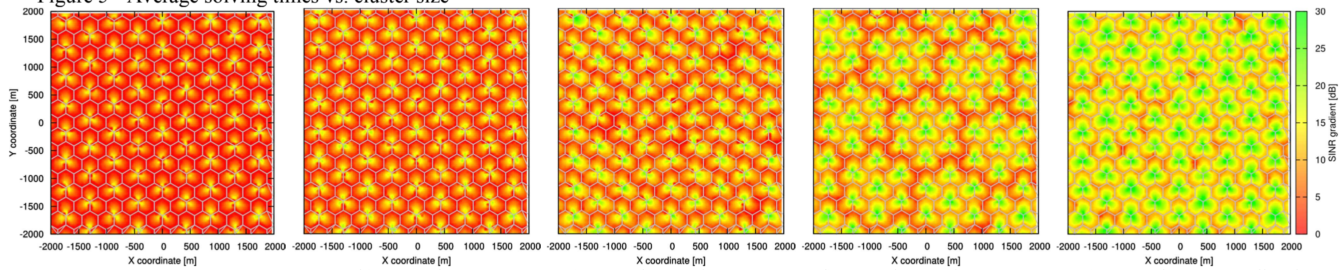
Parameter	*Value*
Inter-site distance	500 m
Carrier frequency	2 GHz
Bandwidth	10 MHz (*M*=50 RBs)
Path loss model	ITU Urban Macro [10]
Macro Node Tx Power	36 dBm
Antenna gain	18 dBm (Macro), 0 dBm
Noise figure	5 dBm
Cable loss	2 dBm

Table 2 - Power models parameters

Parameter	*2012*	*2016*	*2020*
P_{off} [W]	465	141	101
P_{base} [W]	590	279	200
ρ [W]	9.42	15.08	16.66

Table 3 - Max. spectral efficiency params.

Technology	η_{MAX}
LTE	1.5 Mbps
LTE-A	4.5 Mbps
LTE-A Pro	12 Mbps



a) FF b) Random c) CS, cluster size=3 d) CS, cluster size=21 e) Inter-cluster coordination

Figure 6 - SINR distribution, offered load=96Mbps

tively, factoring in the increasing interference of nearby cells every time, until convergence is reached. We repeat the procedure for up to $N_{\max}$ iterations or until convergence is reached, allowing a UE to change its serving node for the first $N < N_{\max}$ iterations. Convergence is assumed when the total interference over two successive iterations is below a configurable threshold δ. For CS, clusters of any size can be enforced by specifying the cluster-head hexagon and the number of hexagons in the cluster. We compare our CS approach against the *FF* and *R* baselines described in Section II, and against static CS schemes with frequency reuse factors of three and seven. With reference to (1), CS implies that $\Delta_{e,x} = \langle \mathbf{R}_e, \mathbf{R}_x \rangle$, where $\mathbf{R}$ is the AM when a solution to the CS problem has been obtained. The version of the CS that we use depends on the scale of the cluster. We vary the rate required by the UEs, and measure their SINR, and the number of allocated RBs and the power consumption of the network. Each measure is the average of 10 *snapshots* (i.e., random drops of UEs in the floorplan).

Simulation parameters are summarized in Table 1. The power model of Figure 2 is parametrized using the values reported in Table 2 [15], representing the power consumption for a 10MHz-bandwidth system for the years 2012, 2016 and 2020, respectively. As far as data rate curves (Figure 3) are concerned, we consider $SINR_{\min} = -10dB$ and $SINR_{MAX} = 30dB$, whereas η_{MAX} is reported in Table 3 for three different technologies. This way, we reflect the progressive shift towards more efficient modulations and transmission schemes, projecting our results towards a 5G scenario. Since results are qualitatively similar, hereafter we show only those for *LTE-A Pro* data-rate curves.

First, we provide a network-wide representation of the effects of CS. Figure 6(a-e) show the distribution of SINR over the network area, obtained with an offered load of 96 Mbps per cell. With no coordination, only UEs close to their serving node perceive a high SINR, especially when using FF. Figure 6c shows the SINR when coordinating clusters of size three, i.e. coordinating cells located at the same site. Intra-cluster cell borders become greener, although large areas of the cells have a low SINR, since interference from neighboring, uncoordinated cells is still strong. The improvement when scaling CS to 21 cells, shown in Figure 6d is remarkable. However, cluster-border UEs still have a low SINR. Their conditions improve by adding inter-cluster coordination, as shown in Figure 6e.

The improvements to the UEs' channel quality is shown in Figure 7 and 8, which report the average and the 5th percentile of the SINR, respectively, with an increasing offered load. We note that, at low loads, coordinating larger clusters improves the SINR of cell-edge UEs. Adding inter-cluster coordination achieves better results since it enhances SINR also for cluster-borders UEs. At high loads, the number of RBs required to satisfy the traffic increases and there is less space for coordination, i.e. it becomes hard to accommodate nodes' allocation so as to minimize interference. In fact, CS cannot perform better than uncoordinated schemes when saturation approaches. On the other hand, SINR values obtained with frequency reuse schemes are insensitive to the offered load, and the resulting SINR is even better than the one with CS at high loads. However, static reuse schemes cannot satisfy the requested traffic even at low loads, since they are restricted to use only a limited portion of the available bandwidth. This is shown in Figure 9, which reports the average number of overloaded RBs per cell, i.e. the number of additional RBs that would be necessary to a node to accommodate its load. As the figure shows, both static reuse schemes saturate much sooner than the others. Figure 10 reports the cumulative distribution function of the SINR for an offered load of 48Mbps. Comparing CS against FF and R scheme, improvements are evident. Moreover, it can be noted that curves of CS shift to the right when increasing the cluster size, whereas adding inter-cluster coordination provides further enhancements in the channel quality of all UEs. Having better SINR means allocating fewer RBs to satisfy the same amount of traffic. This clearly reflects on the average power consumption, which is reported in Figure 11. 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

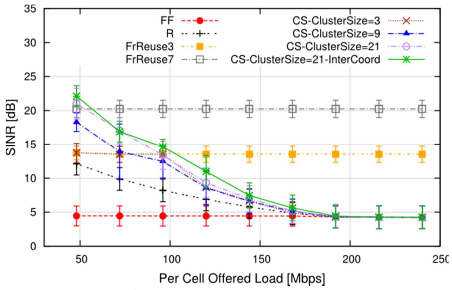


Figure 7 - Average SINR

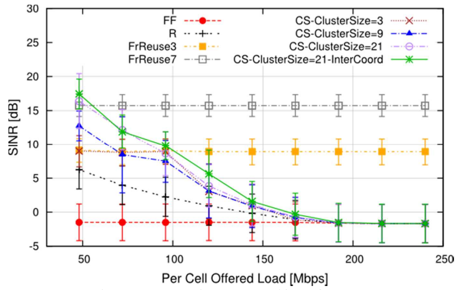


Figure 8 – 5th percentile of SINR

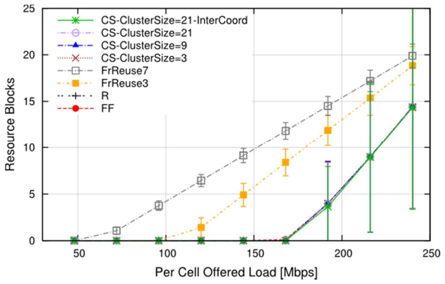


Figure 9 - Overloaded RBs

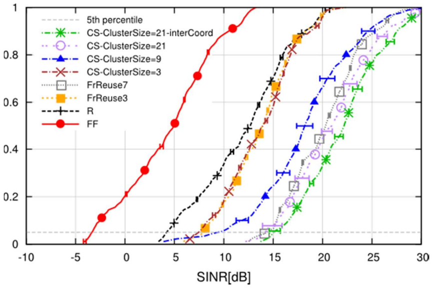


Figure 10 - CDF of SINR, offered load=48Mbps

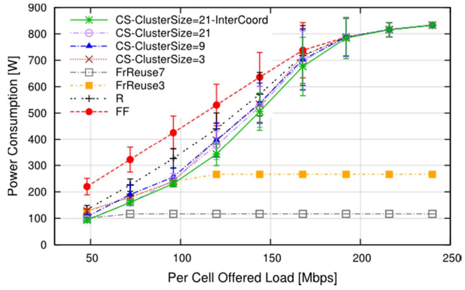


Figure 11 - Power consumption

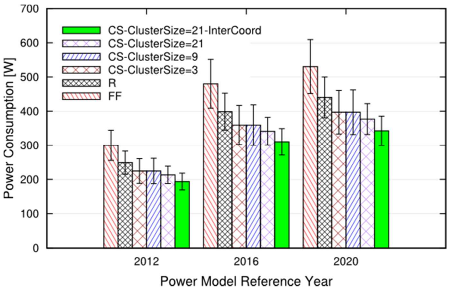


Figure 12 - Comparison of power models

adds a constant offset to the values. It is shown that CS consumes less power than the baselines. Figure 12 compares the depleted power that derives from the three different power models of Table 2. It is shown that moving towards 5G implies more power consumption. In fact, according to the power models, newer technologies require more power per RB, although the constant offset P_{base} is smaller. This implies that employing CS in the next years will allow operators to increase the power saving. As per Figure 12, CS with inter-cluster coordination saves about 100 W with respect to a FF scheme, using a power model of year 2012. On the other hand, with the year-2020 power model, the saving approaches 200 W.

V. Conclusions

In this paper, we have investigated how coordinated scheduling (CS) improves network performance, i.e. it allows it 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 at scales of tens of nodes in a sufficiently short time as to match the dynamics of current and future cellular networks. Then, using a large-scale fluid simulator, we have shown that the above-mentioned benefits of CS actually increase with the scale of coordination, up to the maximum allowed by our models. Moreover, the energy-efficiency benefits will be even more tangible in the near future, when next-generation base stations will be available, whose power consumption depends more on the number of allocated RBs.

Acknowledgment

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

References

[1] Flex5Gware website: http://www.flex5gware.eu (accessed March 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] S.H. Ali, V.C.M. Leung, "Dynamic frequency allocation in fractional frequency reused OFDMA networks", IEEE Transactions on Wireless Communications, vol.8, no.8, pp. 4286-4295, Aug. 2009.

[4] 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.

[5] G. Li, H. Liu, "Downlink Radio Resource Allocation for Multi-Cell OFDMA System", IEEE Transactions on Wireless Communications, vol.5, no.12, pp.3451-3459, Dec. 2006.

[6] C. Koutsimanis, G. Fodor, "A Dynamic Resource Allocation Scheme for Guaranteed Bit Rate Services in OFDMA Networks", Proc. ICC '08, pp.2524-2530, 19-23 May 2008.

[7] M.Y. Arslan, *et al.* "A Resource Management System for Interference Mitigation in Enterprise OFDMA Femtocells", IEEE/ACM Transactions on Networking, vol.21, no.5, pp.1447-1460, Oct. 2013

[8] G. Nardini, *et. al*, "Practical large-scale coordinated scheduling in LTE-Advanced networks", Wireless Networks, 22:(1), pp. 11-31, 2016

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

[10] 3GPP TR 36.814 v9.0.0, "Further advancements for E-UTRA physical layer aspects (Release 9)", March 2010

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

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

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

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

[15] Dario 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, Springer, 2016

[16] L. Pitsoulis "Quadratic Semi-assignment Problem", in Encyclopedia of Optimization, p. 3170-3171, Springer, 2009