Abstract—Heterogeneous and small cell networks (HetSNets) have emerged as a promising mean to significantly improve coverage and performance of next generation cellular networks. However, the high density of base stations in such networks accentuates the harmful impact of interference on network performance. This paper considers a network of multiple small cells where the base stations seek to maximize a common objective by forming clusters and allocating power to their users. We formulate the joint clustering and power allocation problem as a mixed integer optimization problem. We show that such problems can be optimally solved only by performing an exhaustive search over all the possible clustering decisions. Furthermore, it is shown that even if the clustering is established the power allocation problem remains NP-hard. Due to the high computational complexity of the optimal solution, we propose three heuristic algorithms which perform greedy clustering and iterative power allocation. Simulations show that the proposed algorithms, and especially the neighbouring links first heuristic, provide a good computational complexity/performance tradeoff.

I. INTRODUCTION

Heterogeneous and small cell networks (HetSNets) are regarded, by researchers and industry leaders, as an enormous paradigm shift affecting the way cellular networks are designed [1]. Nowadays, base stations (BSs) of different sizes, carrier frequencies and transmit powers are added continuously to underlay the well-established macrocell BSs. This increase in the number of BSs may lead in the near future to cellular networks where each wireless device has its own BS [1]. In such networks, cooperation between BSs can play a major role for enhancing the users’ data rates. In fact, the high density of BSs gives rise to numerous research challenges. A major capacity limitation in wireless networks in general, and in HetSNets in particular, is interference. The latter can be mitigated, or reduced, through cooperative techniques [2].

Since the seminal work of Yoo and Goldsmith [3], zero forcing beamforming (ZFBF) has received a large research interest. In fact, the study in [3] showed that ZFBF can achieve the same sum rate as the optimal dirty paper coding while reducing significantly the implementation complexity. However, ZFBF suffers from high computational complexity when the number of active users in the system becomes large. Hence, most research efforts on ZFBF have focused on devising low complexity scheduling schemes for multiuser multi-input-multi-output (MIMO) systems [3], [4]. When the small cell base stations (SBSs) in a HetSNet are able to cooperate, they can use ZFBF in a distributed fashion. Two new constraints have to be considered: (i) a per BS power constraint instead of a sum power constraint and (ii) interference management between clusters.

Distributed linear precoding, and ZFBF in particular, and its integration in HetSNets has become a hot research topic that is attracting increasing interest [5]–[10]. Studying a per antenna power constraint instead of the extensively studied sum power constraint is done in [11]. The authors show that the power allocation problem under this new constraint is a convex problem that can be hence solved in polynomial time. The work in [5] tackles a problem with the same constraint for fully cooperating multicells. The authors designed the optimal precoder for weighted sum rate maximization. The authors in [6] proposed a new linear precoding scheme for sum rate maximization in cooperative multicell systems. They compared their scheme to other linear precoders and showed that it performs well for low and medium signal-to-noise ratio (SNR). More recently, [7] studies a system where the precoding is performed in a per cell basis. Two scenarios have been compared: a scenario where the BSs do not cooperate and a scenario where the BSs cooperate to compute precoding matrices that maximize a common objective. Another work tackling the same problem for sum rate maximization was presented recently in [8]. The works in [9] and [10] formulate the energy efficiency problem in multicell networks with ZFBF as a nonlinear fractional program and use the Dinckelbach method to solve it.

In addition to the work cited above where the decision is taken in a centralized fashion, coalitional game theory was used in order to design distributed algorithms to solve the coalition formation (or clustering) problem [12]–[15]. Such algorithms may suffer from a large information exchange overhead and cannot always converge to a social optimum.

Unlike the works described above, this work investigates a network of small cells where the SBSs are ready to form multiple clusters and cooperate to maximize a common objective, i.e. the overall sum rate. Therefore, we propose to use a multiuser MIMO precoding scheme in order to reduce the negative impact of interference in HetSNets. Hence, based on a centralized decision, the SBSs are disposed in several clusters and the interference in each cluster is totally suppressed using ZFBF. A cluster is hence regarded as a multiuser MIMO network where SBSs form a virtual large base station with multiple antennas serving multiple users. The clusters are formed and the power is allocated in such a way that inter-cluster interference is minimized.

The studied problem is first formulated as an optimization problem and the complexity of solving it optimally is briefly discussed. In fact, we find that the problem is NP-hard and hence cannot be solved in reasonable (polynomial) time even

Elmahdi Driouch, Wessam Ajib and Chadi Assi

Efficient Heuristics for Clustering and Power Allocation in Small Cell Networks

Elmahdi Driouch, Wessam Ajib and Chadi Assi

978-1-4799-8091-8/15/$31.00 ©2015 IEEE
for small system sizes unless $P = N P$. This motivates us to design suboptimal but computationally efficient algorithms. Three greedy heuristics are presented and their performances are assessed through simulations. We present also the computational complexity of the three algorithms and compare them to the complexity of finding the optimal solution.

The rest of the paper is organized as follows. Section II presents the system model. Section III formulates the problem and discusses its hardness. Section IV describes the proposed clustering and power allocation heuristics. Section V presents the numerical results and the computational complexity of the three algorithms and compares them to the complexity of finding the optimal solution.

II. SYSTEM MODEL

The network model, considered in this work, includes a set $\mathcal{N}$ of $N$ SBSs. Each SBS is associated with exactly one user. The link between SBS $n$ and its associated user $n$ is denoted by $\ell_n$, $n \in \mathcal{N}$. We assume that all the SBSs are transmitting over the same frequency band and that each SBS $n$ has a maximum transmit power constraint $P^\text{max}_n$. Without loss of generality, we are considering the downlink channel. All the SBSs are assumed to be connected to the core network via a fast and reliable backhaul network. Hence, SBSs can exchange information with small delays.

The SBSs are distributed randomly in the same geographical area. Therefore, their associated users will suffer from interference, from neighboring SBSs. In order to lower the impact of interference on their performance, SBSs seek to form clusters. Then, they use beamforming to eliminate interference inside the clusters. This form of interference is referred to as intra-cluster interference and can be totally prevented using ZFBF.

The received signal at user $n$, $n \in \mathcal{N}$, is given by:

$$y_n = h_n x + z_n,$$

where $h_n = [h_{nm}]_{m \in \mathcal{N}}$ is the $(1 \times N)$ channel vector which takes into account the short-term path loss propagation effect and the long-term fading effect between the $N$ SBSs and user $n$. $x$ is the $(N \times 1)$ transmitted vector of signals from all the SBSs and $z_n$ is an additive white Gaussian noise at user $n$ with zero mean and variance $\sigma^2_z$. Assuming that the SBSs are using ZFBF to eliminate intra-cluster interference, then the $n$th element of $x$ sent from SBS $n$ where $n \in S_k$ can be written as:

$$x_n = \sum_{m \in S_k} w_{nm} \sqrt{\gamma_m} a_m,$$

where $w_{nm} = [w_{nm}]_{m \in S_k}$ is the $(1 \times |S_k|)$ beamforming vector used by SBS $n$ to serve the users in the set $S_k$. $\gamma_m$ is the portion of power allocated to user $m$ and $a_m$ is the transmitted signal to user $m$ ([$\cdot$] is the cardinality of the set). Due to ZFBF, the beamforming vector $w_n$ is the column corresponding to SBS $n$ in matrix $W$ which is the pseudo-inverse of matrix $H_k$. Matrix $H_k$ is formed by the channel vectors of all users in the set $S_k$, i.e., $H_k = [h_{nm}]_{m \in S_k}$.

The received signal at user $n$, assuming $n \in S_k$, is:

$$y_n = \sqrt{\gamma_n} a_n + \sum_{p \in \mathcal{N} \setminus S_k} h_{pn} x_p + z_n.$$

The first term in the right hand side of (3) represents the desired signal whereas the second term represents the inter-cluster interference which can be reduced by adequately forming the clusters and assigning power portions. Note that the intra-cluster interference is not present in (3) thanks to ZFBF. Therefore, the achievable data rate at user $n \in \mathcal{N}$ when the inter-cluster interference is treated as noise is:

$$R_n = \log_2 \left(1 + \frac{\gamma_n}{\sigma^2_n + \sum_{p \in \mathcal{N} \setminus S_k} h_{pn} x_p} \right).$$

III. PROBLEM FORMULATION

In order to maximize the system performance in terms of sum rate, we face two main challenges: (i) the clustering problem, i.e., how the links are disposed into clusters and (ii) the power allocation problem, i.e., how each SBS allocates its available power among the users in its cluster. In the following, the joint clustering and power allocation problem is formulated as an optimization problem. In fact, the tackled problem can be formulated involving binary variables that represent the clustering decision and continuous variables that represent the power allocation.

We define the binary variables $x_{nm}$, for all $(n, m) \in \mathcal{N}^2$:

$$x_{nm} = \begin{cases} 1 & \text{if } \ell_n \text{ and } \ell_m \text{ are in the same cluster} \\ 0 & \text{otherwise}. \end{cases}$$

The achievable rate at user $n$, $n \in \mathcal{N}$, can be written as:

$$U_n = \log_2 \left(1 + \frac{\gamma_n}{\sigma^2_n + \sum_{m=1}^{N} (1 - x_{nm}) |h_{nm}|^2 s_m} \right),$$

where

$$s_m = \sum_{q=1}^{N} x_{mq} |w_{mq}|^2 \gamma_q.$$
where the first two constraints ensure the consistency of the definition of variables $x_{nm}$ while the third constraint ensures that no user obtains a negative amount of power.

Therefore, the problem can be written as follows:

\[
\text{Maximize } \sum_{n=1}^{N} U_n \quad (14)
\]
subject to \( (6) - (13) \).

The formulation given by (14) is computationally hard to solve since it involves both binary and continuous variables. Also, due to ZFBF, we cannot compute the matrix channel inverse before fixing the binary variables (i.e. before taking the clustering decision). Hence, the optimal solution to problem (14) can be found only by performing an exhaustive search over all the possible clustering decisions. For each possible clustering, we must perform a separate power allocation by solving the following nonlinear optimization problem:

\[
\text{Maximize } \sum_{k=1}^{K} \sum_{n \in S_k} R_n
\]
subject to \( \sum |w_{nm}|^2 \gamma_m \leq P_n^{\max} \forall k, \forall n \in S_k \quad (15) \)
\[
\gamma_n \geq 0 \quad \forall k, \forall n \in S_k
\]

where $R_n$ is given by (3) and $K$ is the number of clusters in the considered clustering decision.

Unfortunately, the number of possible clustering decisions grows exponentially with the number of small cells in the system. Hence, such optimal brute force solution cannot be implemented even for small number of SBS-user links. Furthermore, the power allocation problem can be proven to be NP-hard based on a polynomial time reduction from the maximum independent set problem in a similar way to the proof of Theorem 1 in [16]. The NP-hardness of the power allocation problem makes the problem of joint clustering and power allocation under the sum rate utility even more harder and an optimal brute force solution impossible to implement.

IV. HEURISTIC ALGORITHMS

Due to the NP-hardness of finding the optimal solution of the tackled problem, we present in this section low complexity greedy algorithms based on simple but efficient selection criteria. The proposed algorithms adopt the same main structure and differ only in the adopted selection criteria. Each algorithm starts by performing a clustering phase by selecting in a greedy fashion one link at each iteration and adding it to its “best” choice cluster. The superlative “best” refers to a local choice due to the greedy nature of the algorithms. Once the clusters are formed, the algorithms proceed to the power allocation by solving problem (15). In the following, we detail three simple heuristic algorithms for the phase of clustering.

A. Heuristic 1: Highest SNR First (HSF)

This first heuristic algorithm starts by putting in the first cluster the link that experiences the highest SNR among all the available links. It then adds the link that causes most interference to the firstly chosen link. If the cluster size limit is larger than two ($C_{\text{max}} > 2$), the algorithm adds the links which cause the most aggregate interference to the links already in the first cluster. The algorithm continues in the same fashion until the cluster is completely filled. It then proceeds to the formation of the next cluster by choosing the link with highest SNR from the remaining links (i.e. the ones not in cluster 1). The algorithm proceeds in the same way and terminates when all the links are assigned to clusters. The different steps of heuristic 1 are detailed in Algorithm 1.

B. Heuristic 2: Highest Interference First (HIF)

This heuristic starts by putting in the first cluster the two links which cause the highest interference to each other. Just like HSF, the algorithm adds the link that causes the most aggregate interference to the already added links. Then, it continues in the same way until the cluster is full. The algorithm forms the other clusters in the same fashion and terminates when no link remains unassigned. The structure of this heuristic is similar to HSF and thus Algorithm 1 can be modified to describe it.

C. Heuristic 3: Neighboring Links First (NFL)

The third heuristic is based only on the separating distances between the users and the interfering SBSs. The algorithm starts by putting in the first cluster the most neighboring links, i.e. the two links $a$ and $b$ where the distance between SBS $a$ and user $b$ or the distance between SBS $b$ and user $a$ is minimal. If $C_{\text{max}} > 2$, the algorithm adds another neighboring
Fig. 1. A network example of \( N = 6 \) links disposed in three clusters

link, i.e. the one having the nearest SBS to the users of the first two links or the one having the nearest user to the SBSs of the first two links. The algorithm continues adding links in the same fashion until no more places are available in the first cluster. Algorithm 1 can be modified to describe NLF.

D. Phase 2: Power Allocation

The second phase of the heuristic algorithms is the power allocation which is common for the three proposed heuristics. Once the clusters are formed, the power allocation problem (given by (15)) is solved. As discussed before, the problem is \( \mathcal{NP} \)-hard due especially to the interdependence between the clusters (i.e. making the objective function non-separable). Hence, we propose to solve the power allocation in an iterative fashion by computing the transmit powers of SBSs in one cluster in each new iteration.

In the first iteration, the algorithm starts by computing the transmit powers of a chosen cluster, \( j \in \{1 \ldots K\} \), (i.e. finding \( \gamma_n \) for all \( n \in S_j \)) by fixing the transmit power of the SBSs of the other clusters to their maximal values (e.g. putting \( P_{\text{max}}^n \) for all \( m \in S_k \) with \( k \neq j \)). Clearly, the problem to solve at each iteration is a convex optimization problem which can be solved using the well-known waterfilling method when \( |S_j| = 2 \) (as presented in [11]) or requiring a numerical optimization tool when \( |S_j| > 2 \). Each iteration involves solving a new convex optimization problem where the transmit power of the already considered clusters are taken equal to their computed values, while the other values of transmit power (not computed yet) are taken equal to their maximal values. When the algorithm finds the transmit powers of all the SBSs, it restarts taking as input these computed portions of power. The algorithm terminates when the utility improvement is not important. We have found through simulations that running the algorithm for only two iterations is enough to obtain a good performance improvement. Running more iterations adds only more computational complexity without significantly improving the system performance.

Fig. 2 compares the sum rate achieved using the three proposed heuristics to the optimal solution found by the highly complex exhaustive search and to the case where no clustering

V. Simulation Results

In this section, we investigate the performance of the proposed heuristic algorithms. We consider a small cell network where \( N \) links are randomly located according to a uniform distribution in a rectangular area as shown in Fig. 1. The size of this area is \( 200m \times 200m \). All the links have the same separation distance \( d_{\text{mm}} = 15m \). The channel coefficients are modeled as \( h_{mn} = K_0 \cdot (d_{mn}/d_0)^{-\alpha} \cdot \beta_{mn} \), where \( K_0 = 10^3 \) is a constant capturing system and transmission effects, \( d_{mn} \) is the distance between SBS \( m \) and user \( n \), \( d_0 = 1 \) m is the reference distance, \( \alpha = 4 \) is the path loss exponent and \( \beta_{mn} \) is a random Gaussian variable with zero mean and unit variance. The noise at each user is chosen to be \( N_0 = 10^{-10} \)W and \( P_{\text{max}}^m = P_{\text{max}}^n = 0.1 \)W. The sum rates are obtained by averaging over 2000 simulation runs.

Fig. 2 compares the sum rate achieved using the three proposed heuristics to the optimal solution found by the highly complex exhaustive search and to the case where no clustering
is performed. We notice that HIF always outperforms the two other heuristics. The HIF heuristic achieves sum rate close to the exhaustive search algorithm with a relatively tight gap between 2% and 2.7%. The HIF outperforms the no-clustering case with a gap approaching 25%. Also, the performance gap between HIF and NLF heuristics does not exceed 1%. NLF is based only on the information about the distances separating the SBSs to obtain the clustering decision. Hence, this heuristic is still interesting to implement thanks to its limited feedback information requirements. Finally, we notice that as $N$ increases, the performance of HSF degrades because the impact of interference becomes noticeable.

Fig. 3 compares the sum rate of the three heuristics when varying the value of $C_{\text{max}}$ for $N = 20$. We notice that the performance for the three heuristics increases with the increase of $C_{\text{max}}$. We also notice that for large values of $C_{\text{max}}$, the performance of the HSF heuristic approaches the performance of the other heuristics. This is due to the decrease of the impact of inter-cluster interference since when $C_{\text{max}}$ gets larger, the system forms less clusters.

We compare in Fig. 4 execution times of the three proposed heuristics to the exhaustive search algorithm. The execution time is the amount of time needed to perform the clustering and compute the power allocation. The four algorithms were all implemented in MATLAB and the simulations were performed in the same computer. We notice that the proposed algorithms have very similar execution times which are dominated by the time needed to perform the matrix channel inversion (which belongs to $O(N^3)$). We can also observe that even for a relatively small network with $N = 10$, the optimal algorithm suffers from a very high complexity. For $N = 10$, the proposed heuristics allow a complexity reduction of more than five orders of magnitude over the exhaustive search.

VI. Conclusion

In this paper, we proposed efficient heuristics for clustering and power allocation in HetSNetS. We have considered a network where SBSs try to form clusters in order to maximize the achievable sum rate. We have formulated the problem as optimization an problem and discussed its computational complexity and hardness. Since the optimal solution can be found only with a highly computationally complex exhaustive search (for clustering) coupled with an $NP$-hard optimization (for power allocation), we proposed three heuristic algorithms which perform clustering in polynomial time and find suboptimal power allocation using an iterative scheme. Simulation results show that the proposed algorithms (especially HIF) achieve a good tradeoff between computational complexity and performance compared to the complex optimal solution. In future work, we will design distributed clustering algorithms and investigate the clustering problem in the case where the SBSs can use cognitive capabilities to coexist with macrocells.

References