Fairness-Driven Fast Resource Allocation for Interference-Free Heterogeneous Networks

Daniel Calabuig *Member, IEEE*, Jose F. Monserrat *Member, IEEE*, and Narcís Cardona *Member, IEEE*

Abstract—**Resource allocation in heterogeneous wireless networks is a complex problem. This letter addresses this problem maximizing the sum of logarithms of received rates. From this function, and using the Karush-Kuhn-Tucker (KKT) conditions, this paper proposes an iterative algorithm for allocating resources, which is much faster than conventional techniques. This fast response can be used to optimize larger regions, getting closer to the performance of global and centralized algorithms.**

I. Introduction

HETEROGENEOUS networks (HetNets) was the name given in 2009 by the Third Generation Partnership Project (3GPP) to advanced cell structures where additional nodes, like pico/femto-cells or relays, are deployed to improve overall capacity. HetNets pose significant conceptual and practical challenges to the state of the art. One of those challenges is the management of interference. Apart from the identification of white spaces with spectrum sensing, and the exchange of information between network nodes [1], power control is another evident means to reduce interference. In [2], the authors investigated this solution and provided a link budget analysis. In [3], Bernardo *et al.* proposed a joint self-optimization of spectrum assignment and transmission power for femto-cells. Finally, Park *et al.* studied a beam subset selection strategy to reduce interferences between macro and femto-cells [4]. In general, different methods can self-configure the HetNet in such a way that there is no interference to/from the different nodes. This paper focuses on such interference-free scenarios for HetNets.

A different and less studied problem is dynamic resource allocation. Due to the inherent complexity of resource allocation in HetNets, most of the authors agree on the need of new distributed mechanisms [5]. A common approach consists in clustering network nodes that may coordinate in the usage of resources with different techniques, like [6], [7]. In [6], Xiang *et al.* proposed a two-phase method that uses an heuristic and convex optimization to distribute resources, and, then, allocates power to the nodes in the HetNet. However, this kind of approach is simplistic since it assumes that users are attached to one node. Moreover, the scheduler only aims at increasing the system throughput, not considering any fairness between users. Lee *et al.* [7] addressed intercell fairness proposing an iterative water-filling mechanism. However, again, this work attaches users to one node.

This paper addresses the optimization of the resource allocation process in terms of fairness. Moreover, users are not attached to any node, and, hence, the node —or nodes— that communicates with each user is optimized too. The formulation presented in Section II results in a convex problem (CP), which can be solved using interior-point methods. In Section III, we present an alternative algorithm based on successive Linear Programs (LPs), which can obtain the optimum in a significantly shorter amount of time, compared with interior-point methods. Section IV proves that this efficient algorithm reaches the optimum solution of the original problem. Finally, Section V presents a performance evaluation of the proposed algorithm with respect to an interior-point method.

The advantage of fast convergence algorithms is that larger clusters can be used; which, at the end, means that the distributed algorithm approaches the performance of centralized solutions.

II. System model and optimization

Let I be the number of users in the system, and K the number of nodes. Moreover, let T_{ik} be the effective throughput per resource unit (r.u.) that user i can obtain from node k with its current channel. The objective of this work is to maximize the sum of the rate logarithms,

$$\begin{aligned} \text{Maximize} \quad & f(\boldsymbol{\rho}) = \sum_{i=1}^{I} \log\left(\sum_{k=1}^{K} \rho_{ik} T_{ik}\right), && \text{(1a)}\\ \text{subject to} \quad & \rho_{ik} \geq 0, \text{ if } T_{ik} > 0, \ \forall i,k, && \text{(1b)}\\ & \rho_{ik} = 0, \text{ if } T_{ik} = 0, \ \forall i,k, && \text{(1c)}\\ & g_k(\boldsymbol{\rho}) = \sum_{i=1}^{I} \rho_{ik} - \rho_{\max k} \leq 0, \ \forall k, && \text{(1d)} \end{aligned}$$

where ρ_{ik} is the quantity of resources allocated to the i-th user in the k-th node, $\boldsymbol{\rho}$ is the vector of all ρ_{ik}, and $\rho_{\max k} > 0$ is the maximum quantity of resources of the k-th node. Note that $\sum_{k=1}^{K} \rho_{ik} T_{ik}$ is the bit rate allocated to the i-th user. The optimization variable of (1) is the vector $\boldsymbol{\rho}$. This formulation assumes that it is possible to serve users from several nodes at the same time. Fairness is given by the logarithmic law.

For the rest of the paper, we assume that all users are in the coverage area of at least one node and all nodes have at least one user in their coverage areas. Note that, otherwise, we could always eliminate that user or node from the optimization.

The problem (1) is convex, since we are maximizing a concave function subject to linear constraints. Moreover, the duality gap is zero, since (1) satisfies Slater's condition. Therefore, the Karush-Kuhn-Tucker (KKT) conditions are necessary

Manuscript received March 30, 2012; revised May 3, 2012. This work was supported by a Marie Curie International Outgoing Fellowship of the European Commission under the project COMIC. Authors are with the Universidad Politécnica de Valencia.

and sufficient for optimality. The KKT conditions of (1) are

$$\text{(1b)-(1d)}, \tag{2a}$$
$$\lambda_{ik} \geq 0, \text{ if } T_{ik} > 0, \ \forall i, k, \tag{2b}$$
$$\mu_k \geq 0, \ \forall k, \tag{2c}$$
$$\lambda_{ik}\rho_{ik} = 0, \ \forall i, k, \tag{2d}$$
$$\mu_k g_k(\boldsymbol{\rho}) = 0, \ \forall k, \tag{2e}$$
$$-G_{ik}(\boldsymbol{\rho}) - \lambda_{ik} + \mu_k = 0, \ \forall i, k, \tag{2f}$$

where λ_{ik} are the dual variables associated with constraints (1b) and (1c), μ_k are the dual variables associated with constraints (1d), and $G_{ik}(\boldsymbol{\rho}) = \partial f(\boldsymbol{\rho})/\partial \rho_{ik} = T_{ik}/(\sum_{l=1}^{K} \rho_{il} T_{il})$ is the gradient of $f(\boldsymbol{\rho})$. If we find some $\hat{\rho}_{ik}$, $\hat{\lambda}_{ik}$ and $\hat{\mu}_k$ such that they satisfy (2), then $\hat{\boldsymbol{\rho}}$, defined as the vector of all $\hat{\rho}_{ik}$, is the optimum solution of (1).

This problem can be solved numerically using interior-point techniques. However, the complexity of interior-point methods for general convex problems has not been bounded yet. Therefore, the resolution of (1) may require a non-viable amount of time. This would mean the reduction of the cluster size, with small K and I. However, the KKT conditions can give us very useful hints to find a more efficient algorithm. More specifically, let us assume $T_{ik} > 0$, which is true for at least one user in the k-th node. Then, $G_{ik}(\hat{\boldsymbol{\rho}}) > 0$. Since $\hat{\lambda}_{ik}$ is non-negative, then, necessarily, $\hat{\mu}_k > 0$. This means that all nodes have to be using all their resources. What is even more interesting is that the gradient of all users consuming resources in the k-th node must be equal to $\hat{\mu}_k$, and, hence, equal to each other. This property highlights a relevant structure of optimum solutions that can be exploited to design an efficient algorithm.

III. Optimum and efficient resource allocation

The proposed resource allocation algorithm is based on the resolution of several optimization problems that can be solved as LPs. This section presents these problems and how to formulate them. Next section proves the optimality of the final solution. Let us define $\mathcal{I}_k^{(1)}, \cdots, \mathcal{I}_k^{(r)}, \cdots$ as a users sets succession for the k-th node, where $\mathcal{I}_k^{(r)} \subseteq \{1, \cdots, I\}$ and r is the iteration index. Moreover, let $\Phi(\mathcal{I}_1^{(r)}, \cdots, \mathcal{I}_K^{(r)})$ be the following *partial* optimization problem:

$$\text{Maximize} \quad \sum_{i=1}^{I}\sum_{k=1}^{K} \rho_{ik} T_{ik}, \tag{3a}$$
$$\text{subject to} \quad \rho_{ik} \geq 0, \ \forall i \in \mathcal{I}_k^{(r)}, k, \tag{3b}$$
$$\rho_{ik} = 0, \ \forall i \notin \mathcal{I}_k^{(r)}, k, \tag{3c}$$
$$g_k(\boldsymbol{\rho}) \leq 0, \ \forall k, \tag{3d}$$
$$G_{ik}(\boldsymbol{\rho}) = G_{jk}(\boldsymbol{\rho}), \ \forall i, j \in \mathcal{I}_k^{(r)}, k, \tag{3e}$$
$$G_{ik}(\boldsymbol{\rho}) \geq G_{jk}(\boldsymbol{\rho}), \ \forall i \in \mathcal{I}_k^{(r)}, j \notin \mathcal{I}_k^{(r)}, k \tag{3f}$$

Note that (3) is not a LP, since (3e) and (3f) are not linear functions of $\boldsymbol{\rho}$. However, these constraints can be easily transformed into linear functions, since

$$G_{ik}(\boldsymbol{\rho}) = G_{jk}(\boldsymbol{\rho}) \Leftrightarrow \frac{\sum_{l=1}^{K} \rho_{il} T_{il}}{T_{ik}} = \frac{\sum_{l=1}^{K} \rho_{jl} T_{jl}}{T_{jk}}, \tag{4}$$

$$G_{ik}(\boldsymbol{\rho}) \geq G_{jk}(\boldsymbol{\rho}) \Leftrightarrow \frac{\sum_{l=1}^{K} \rho_{il} T_{il}}{T_{ik}} \leq \frac{\sum_{l=1}^{K} \rho_{jl} T_{jl}}{T_{jk}}, \tag{5}$$

for all $T_{ik} > 0$ and $T_{jk} > 0$. For the other cases, where $T_{ik} = 0$ and/or $T_{jk} = 0$, note that, if $T_{jk} = 0$, (3f) and (5) reduce to $G_{ik}(\boldsymbol{\rho}) \geq 0$, which is ensured by (3b). Thus, these constraints are redundant. Finally, the users sets are going to be selected in such a way that, if $i \in \mathcal{I}_k^{(r)}$, then $T_{ik} > 0$. Consequently, although (3) is not a LP, it can be solved with an equivalent LP.

The users in $\mathcal{I}_k^{(r)}$ are those allowed to consume resources in the k-th node. The objective of (3) is to maximize the sum rate, subject to certain constraints. Constraints (3b)-(3d) are related with (1b)-(1d). (3e) reflects the fact that all users must have the same gradient, if they consume resources in the same node. Finally, the relevance of (3f) will be shown in Section IV.

We define the *partial* solution $\tilde{\rho}_{ik}^{(r)}$ as the solution of $\Phi(\mathcal{I}_1^{(r)} \cdots \mathcal{I}_K^{(r)})$, $\tilde{\boldsymbol{\rho}}^{(r)}$ as the vector of all $\tilde{\rho}_{ik}^{(r)}$, and the distance between two resource allocation vectors $\boldsymbol{\rho}$ and $\boldsymbol{\sigma}$ as

$$d(\boldsymbol{\rho}, \boldsymbol{\sigma}) = \sqrt{\sum_{i=1}^{I}\sum_{k=1}^{K} \left(G_{ik}(\boldsymbol{\rho}) - G_{ik}(\boldsymbol{\sigma})\right)^2}. \tag{6}$$

Theorem 1: There is a succession of users sets for each node such that

$$\lim_{r \to \infty} d\left(\tilde{\boldsymbol{\rho}}^{(r)}, \hat{\boldsymbol{\rho}}\right) = 0. \tag{7}$$

Proof: This paper is devoted to prove this theorem. ■

Note that if $d(\boldsymbol{\rho}, \boldsymbol{\sigma}) = 0$ then gradients are equal, which means that data rates allocated to the users are the same. Consequently $f(\boldsymbol{\rho}) = f(\boldsymbol{\sigma})$. This theorem states that, with the correct succession of sets, we can approach to the optimum solution of (1) as much as desired. We are going to prove Theorem 1 finding the succession of users sets that satisfy (7).

A. Users sets updating

After solving the r-th problem $\Phi(\mathcal{I}_1^{(r)} \cdots \mathcal{I}_K^{(r)})$, the user sets can be updated following

$$\mathcal{I}_k^{(r+1)} = \mathcal{A}_k^{(r+1)} \cup \mathcal{B}_k^{(r+1)} \cup \mathcal{C}_k^{(r+1)}, \tag{8}$$

where $\mathcal{A}_k^{(r+1)} = \{i : i \in \mathcal{I}_k^{(r)}, g_k(\tilde{\boldsymbol{\rho}}^{(r)}) < 0\}$, $\mathcal{B}_k^{(r+1)} = \{i : \tilde{\rho}_{ik}^{(r)} > 0\}$, and $\mathcal{C}_k^{(r+1)} = \{i : i \notin \mathcal{I}_k^{(r)}, G_{ik}(\tilde{\boldsymbol{\rho}}^{(r)}) = G_{jk}(\tilde{\boldsymbol{\rho}}^{(r)}), j \in \mathcal{I}_k^{(r)}\}$. Although the formulation of (8) might seem complex, it is very intuitive. The set $\mathcal{A}_k^{(r+1)}$ includes all users that were previously in the user set, but only if the k-th node has some free resources, otherwise $\mathcal{A}_k^{(r+1)}$ is empty. The set $\mathcal{B}_k^{(r+1)}$ includes all users that strictly consume resources in the k-th node, i.e. the users whose $\tilde{\rho}_{ik}^{(r)}$ is strictly greater than zero. It is easy to see that, if the node has some free resources, then $\mathcal{B}_k^{(r+1)} \subseteq \mathcal{A}_k^{(r+1)}$. With this updating, we are maintaining the users in the k-th user set as long as the node has free resources. If saturation is reached, then we eliminate those users that are not actually using node k. Moreover, this ensures that, if $\mathcal{I}_k^{(1)}$ is not empty, then no set $\mathcal{I}_k^{(r)}$ will be empty. The set $\mathcal{C}_k^{(r+1)}$ adds those users that were not in the k-th user set and have the same gradient as the users in it.

B. Initial users sets

Before describing the initial users sets, let us introduce some definitions that are going to be used throughout this section. Let $\hat{k}_i$ be the best node for the i-th user, that is, $T_{i\hat{k}_i} \geq T_{ik}$ for all k. Moreover, let $\mathcal{K}$ be the set of nodes that are the best ones for at least one user, that is, $\mathcal{K} = \{k : k = \hat{k}_i \text{ for at least one } i\}$. Then, the initial users sets are chosen as

$$\mathcal{I}_k^{(1)} = \{i : \hat{k}_i = k\},\ \forall k \in \mathcal{K}, \tag{9}$$

$$\mathcal{I}_k^{(1)} = \left\{ \arg\max_i \frac{T_{ik}}{T_{i\hat{k}_i}} \right\},\ \forall k \notin \mathcal{K}. \tag{10}$$

These sets have some important properties. First, all users belong to one set at least. Second, all users sets are non-empty. Third, if $k \notin \mathcal{K}$, then $\mathcal{I}_k^{(1)}$ has just one user. Forth, if $i \in \mathcal{I}_k^{(1)}$, then $T_{ik} > 0$. Finally, the feasible region of $\Phi(\mathcal{I}_1^{(1)}, \cdots, \mathcal{I}_K^{(1)})$ is non-empty, and, hence, it is a feasible problem. In order to proof the last statement, we are going to build a feasible resource allocation. More specifically, make $\rho_{ik} = \alpha$ if $k = \hat{k}_i$, and $\rho_{ik} = 0$ otherwise, where $\alpha > 0$ has to be selected to satisfy (3d). Then,

$$G_{ik}(\boldsymbol{\rho}) = \frac{T_{ik}}{\alpha T_{i\hat{k}_i}} \leq \frac{1}{\alpha} = G_{i\hat{k}_i}(\boldsymbol{\rho}). \tag{11}$$

This means that (3e) and (3f) are satisfied for the nodes in $\mathcal{K}$, since $G_{i\hat{k}_i}(\boldsymbol{\rho}) = G_{j\hat{k}_j}(\boldsymbol{\rho}) \geq G_{j\hat{k}_i}(\boldsymbol{\rho})$. Moreover, if $k \notin \mathcal{K}$ and $i \in \mathcal{I}_k^{(1)}$, then $T_{ik}/T_{i\hat{k}_i} \geq T_{jk}/T_{j\hat{k}_j}$, which at the end means that $G_{ik}(\boldsymbol{\rho}) \geq G_{jk}(\boldsymbol{\rho})$. (3e) is not a problem for nodes $k \notin \mathcal{K}$, since they just have one user.

IV. Proof of optimality

The proof presented in this section is divided into three steps. In the first step, we show that all partial optimization problems are feasible, which is important to ensure that they have a solution. In the second step, we show that, if $\tilde{\boldsymbol{\rho}}^{(r)}$ is not optimum for (1), then there is at least one k such that $\mathcal{I}_k^{(r)} \neq \mathcal{I}_k^{(r+1)}$. In the third step, we show that $G_{ik}(\tilde{\boldsymbol{\rho}}^{(r)}) \geq G_{ik}(\tilde{\boldsymbol{\rho}}^{(r+1)})$ for all i, k and r. Moreover, we show that, if $\tilde{\boldsymbol{\rho}}^{(r)}$ is not optimum for (1), then there is at least one i and one k such that $G_{ik}(\tilde{\boldsymbol{\rho}}^{(r)}) > G_{ik}(\tilde{\boldsymbol{\rho}}^{(r+1)})$. Assuming that these three steps are true, the gradients continuously decrease if the partial solutions are not optimum for (1), and hence, they are different. Consequently, there must be some difference between the users sets of two consecutive algorithm iterations. Since all the combinations of user sets are finite —the number of subsets of a finite set is finite too,— then the algorithm must converge, and we must reach an iteration $r_0 < \infty$ in which $G_{ik}(\tilde{\boldsymbol{\rho}}^{(r)}) = G_{ik}(\tilde{\boldsymbol{\rho}}^{(r+1)})$ for all i, k and $r \geq r_0$. However, this convergence is not possible if $\tilde{\boldsymbol{\rho}}^{(r)}$ is not an optimum solution of (1). As a consequence, the algorithm presented in Section III must converge to an optimum solution of (1) in a finite number of iterations. This proves Theorem 1.

A. First step

We have already shown the feasibility of the first partial problem. For any other iteration, $\tilde{\boldsymbol{\rho}}^{(r)}$ is a feasible point of $\Phi(\mathcal{I}_1^{(r+1)} \cdots \mathcal{I}_K^{(r+1)})$. In order to see this, we have to look at the changes performed in the user sets in (8), and see if the solution is still feasible with those changes. First, let us assume that $i \in \mathcal{I}_k^{(r)}$ and $i \notin \mathcal{I}_k^{(r+1)}$. This means that the node ran out of resources and the user was not consuming any of them. For the $r+1$-th partial problem, $\tilde{\rho}_{ik}^{(r)}$ satisfies (3c), and $G_{ik}(\tilde{\boldsymbol{\rho}}^{(r)})$ satisfies (3f) with equality. Let us assume now that $i \notin \mathcal{I}_k^{(r)}$ and $i \in \mathcal{I}_k^{(r+1)}$. Then, $G_{ik}(\tilde{\boldsymbol{\rho}}^{(r)}) = G_{jk}(\tilde{\boldsymbol{\rho}}^{(r)})$ for any $j \in \mathcal{I}_k^{(r)}$. For the $r+1$-th partial problem, $\tilde{\rho}_{ik}^{(r)}$ satisfies (3b) with equality, and $G_{ik}(\tilde{\boldsymbol{\rho}}^{(r)})$ satisfies (3e). Therefore, despite the changes performed to the users sets, the r-th partial solution is a feasible point of the $r+1$-th partial problem.

B. Second step

The objective function (3a) is strictly increasing and unbounded. Therefore, the partial solutions are always reached in the hull of the feasible set. This means that the objective (3a) cannot be increased because of some inequalities that are satisfied as equalities.

Let us assume that, for a given k, $g_k(\tilde{\boldsymbol{\rho}}^{(r)}) < 0$. Hence, $\tilde{\boldsymbol{\rho}}^{(r)}$ is not optimum for (1), as stated in Section II. In this case, the constraints (3b)-(3d) are not preventing the variation of $\tilde{\boldsymbol{\rho}}^{(r)}$. Since $\tilde{\boldsymbol{\rho}}^{(r)}$ is the partial solution, this leads to two cases: $\tilde{\boldsymbol{\rho}}^{(r)}$ cannot be further modified because of (3e), case (a); or because of (3f), case (b). In case (a), we can see that there is another node l, different from k, such that $i \in \mathcal{I}_k^{(r)}$ and $i \in \mathcal{I}_l^{(r)}$, and another user $j \in \mathcal{I}_l^{(r)}$ cannot increase its gradient because it has no access to more resources. This is the reason why it is not possible to increase $\tilde{\rho}_{ik}^{(r)}$. That would increase $G_{il}(\tilde{\boldsymbol{\rho}}^{(r)})$, which is not possible because of (3e). This directly leads to $g_l(\tilde{\boldsymbol{\rho}}^{(r)}) = 0$ and $\tilde{\rho}_{il}^{(r)} = 0$. Consequently, $i \notin \mathcal{I}_l^{(r+1)}$ because of (8), which produces a change in the user set l. In case (b), there is a node l, which might be the same k, and a user j, which cannot increase its gradient, such that $i \in \mathcal{I}_l^{(r)}$, $j \notin \mathcal{I}_l^{(r)}$, and $G_{il}(\tilde{\boldsymbol{\rho}}^{(r)}) = G_{jl}(\tilde{\boldsymbol{\rho}}^{(r)})$. Then, from (8), $j \in \mathcal{I}_l^{(r+1)}$, which produces a change in the user set l too.

Let us assume now that $g_k(\tilde{\boldsymbol{\rho}}^{(r)}) = 0$, for all k. In this situation $\tilde{\boldsymbol{\rho}}^{(r)}$ is an optimum solution of (1). In order to prove that, we are going to find some λ_{ik} and μ_k such that they satisfy (2). Let us make $\lambda_{ik} = 0$ for all $i \in \mathcal{I}_k^{(r)}$, and $\mu_k = G_{ik}(\tilde{\boldsymbol{\rho}}^{(r)})$, for any $i \in \mathcal{I}_k^{(r)}$ since all of them are equal. The users that are not in $\mathcal{I}_k^{(r)}$ are not consuming any resource of this node. Therefore, $\tilde{\rho}_{ik} = 0$, and we can set any positive value for λ_{ik}. Let us make $\lambda_{ik} = \mu_k - G_{ik}(\tilde{\boldsymbol{\rho}}^{(r)}) \geq 0$, for all $i \notin \mathcal{I}_k^{(r)}$, which is non-negative thanks to (3f). As a consequence, these dual variables satisfy (2), and the resource allocation vector $\tilde{\boldsymbol{\rho}}^{(r)}$ is optimum for (1).

C. Third step

Let us prove that $G_{ik}(\tilde{\boldsymbol{\rho}}^{(r)}) \geq G_{ik}(\tilde{\boldsymbol{\rho}}^{(r+1)})$ for all i, k and r. Let $\boldsymbol{\rho}$ be a certain resource allocation vector that is feasible for the $r+1$-th partial problem, and let $G_{ik}(\boldsymbol{\rho}) > G_{ik}(\tilde{\boldsymbol{\rho}}^{(r)})$, for a certain i and k. If that is true, the rest of users consuming resources in the k-th node must have a greater gradient too, due to (3e). Moreover, if the gradients in the k-th node are greater, the gradients of these users in other nodes they are in coverage will be greater too. If any of these users is in another user set, this effect is propagated to the users of that set. A user is not affected only if it is completely isolated from the user and node where this increment was originated. However, if the user is isolated, it could not use the resources freed by other users. Therefore, the increment of the gradients does not benefit anyone, and it reduces the value of the objective (3a). Then, $\boldsymbol{\rho}$ cannot be the solution of the $r+1$-th partial problem.

We now prove that, if $\tilde{\boldsymbol{\rho}}^{(r)}$ is not optimum for (1), there is one i and one k such that $G_{ik}(\tilde{\boldsymbol{\rho}}^{(r)}) > G_{ik}(\tilde{\boldsymbol{\rho}}^{(r+1)})$. It is important to recall that, if $\tilde{\boldsymbol{\rho}}^{(r)}$ is not optimum for (1), then there is a k such that $g_k(\tilde{\boldsymbol{\rho}}^{(r)}) < 0$. Let $i \in \mathcal{I}_k^{(r)}$, then $\tilde{\rho}_{ik}^{(r)}$ has not a higher value because there is an l and another user j that cannot use more resources, such that $i \in \mathcal{I}_l^{(r)}$ and either $j \in \mathcal{I}_l^{(r)}$ and $g_l(\tilde{\boldsymbol{\rho}}^{(r)}) = 0$, previous case (a), or $j \notin \mathcal{I}_l^{(r)}$ and $G_{il}(\tilde{\boldsymbol{\rho}}^{(r)}) = G_{jl}(\tilde{\boldsymbol{\rho}}^{(r)})$, previous case (b). The changes performed in the user sets ensure that $i \notin \mathcal{I}_l^{(r+1)}$ in case (a) and $j \in \mathcal{I}_l^{(r+1)}$ in case (b). Thus, the i-th user will strictly increase its gradient from the r-th to the $r+1$-th iteration, since it has access to free resources at least in node k, and the constraints that avoided that increment in the r-th partial problem do not prevent it in the $r+1$-th partial problem.

V. Performance evaluation

This section presents a performance study of the Optimum and Efficient Resource Allocation (OERA) mechanism of Section III in terms of computation time. As a reference, the CP (1) was solved using an Interior-Point Algorithm (IPA). The partial problems of OERA were solved using the simplex algorithm for LPs. Although simplex has a non-polynomial worst case cost, in average, it is quite efficient. Both techniques were implemented in MATLAB, where we used the interior-point solver of `fmincon.m`, and the simplex solver of `linprog.m`.

The simulation scenario was a squared area with a variable side length, which was wrapped around like a torus. Nodes and users were randomly distributed, with 10 nodes/km^2 and 150 users/km^2. The system had a power of 1 W/km^2, which was randomly split into the nodes. The noise power at the user terminals was set to 10^{-13} W, and the path loss was computed as $150+45\log_{10}(r_{ik})$ dB, where r_{ik} is the distance, in km, between the i-th user and the k-th node. The available resources were the bandwidth, which was set to 3 MHz for each node, and the r.u. was 1 Hz. For the sake of simplicity, the effective throughput per r.u. was calculated using the Shannon limit but assuming a sensitivity of -113 dBm. No shadowing or fast-fading was considered. Results were averaged for a total of 100 seeds. All simulations were carried out in an Intel Core2 Quad Processor Q8200, and with equal conditions.

The main result is shown in Figure 1, where OERA was always faster than IPA. The relevant points are those of areas greater than 0.1 km^2, since, for smaller areas, there is just one node, and it is possible to find a closed formula for the solution of (1). In any case, those points are interesting because they establish the starting points of the posterior evolution. The time required by IPA increases much faster than the time required by OERA. The slope of OERA is always around 2 in logarithmic scale, which means a quadratic law with respect to the area. However, the slope of IPA always increases, and it goes beyond 4 for areas greater than 1 km^2.

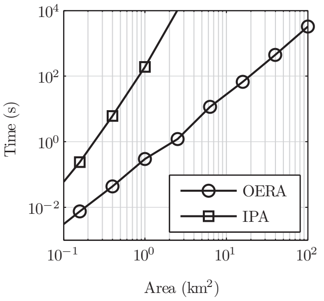

Fig. 1. Average time needed to find the solution of (1).

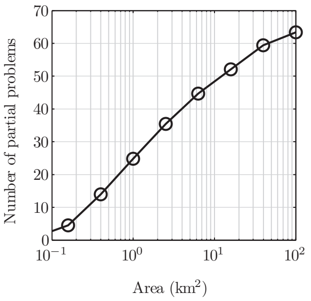

Fig. 2. Average number of partial problems of the OERA.

The weakness of the OERA is the need for several partial problems. We know that the number of problems to be solved is bounded, but this bound is quite big, and not tight. Figure 2 shows the average number of partial problems required to find the optimum solution. As it can be seen, the number of problems increases very slowly with the area.

VI. Conclusion

This paper has proposed a new algorithm to fairly allocate resources in interference-free heterogeneous networks. The key advantage is its significantly lower complexity, in terms of computation time. This reduced time can be used to optimize larger regions, achieving a better and global optimization.

References

[1] J. Huang, P. Xiao and X. Jing,“A Downlink ICIC Method Based on Region in the LTE-Advanced System,” in *Proc. IEEE Personal, Indoor and Mobile Radio Communications*, Istambul, Turkey, 2010.

[2] V. Chandrasekhar, J.G. Andrews, T. Muharemovic, Z. Shen and A. Gatherer, “Power Control in Two-Tier Femtocell Networks,” *IEEE Trans. Wireless Commun.*, vol 8, no. 8, pp. 43164328, 2009.

[3] F. Bernardo, R. Agustí, J. Cordero and C. Crespo, “Self-optimization of spectrum assignment and transmission power in OFDMA femtocells,” in *Proc. Advanced Int. Conf. on Telecommunications*, Barcelona, Spain, 2010.

[4] S. Park *et alt.*, “Beam Subset Selection Strategy for Interference Reduction in Two-Tier Femtocell Networks,” *IEEE Trans. on Wireless Commun.*, vol.9, no.11, pp.3440-3449, 2010.

[5] A. Ladanyi, D. Lopez-Perez, A. Juttner, C. Alpar, X. Chu, J. Zhang, “Distributed Resource Allocation for Femtocell Interference Coordination via Power Minimisation,” in *Proc. IEEE Global Communications Conf.*, pp.744-749, 2011.

[6] J. Xiang, Y. Zhang, T. Skeie and X. Lang, “Downlink Spectrum Sharing for Cognitive Radio Femtocell Networks,” *IEEE Syst. J.*, vol.4, no.4, pp.524-534, 2010.

[7] K. Lee, O. Jo and D. Cho, “Cooperative Resource Allocation for Guaranteeing Intercell Fairness in Femtocell Networks,” *IEEE Commun. Letters*, vol. 15, no. 2, pp. 214-216, 2011.