Collaborative Allocation of Orthogonal Frequency Division Multiplex Sub-Carriers using the Swarm Intelligence

J.C. Dunat
Motorola MDB, Toulouse, France
Email: jcdunat@motorola.com

D. Grandblaise[1] and C. Bonnet[2]
[1]Motorola Labs, Gif sur Yvette, France - [2]Institut Eurecom, Sophia-Antipolis, France
Email: [1]david.grandblaise@motorola.com - [2]christian.bonnet@eurecom.fr

***Abstract*— Future generations of wireless systems require opportunistic spectrum access techniques to effectively detect and access temporarily unused spectrum bands. Cognitive radios, with their ability to learn and adapt to their environment, promise to possess such powerful capabilities. As a consequence, the spectrum allocation of a wireless system could quickly and appropriately auto-adapt to react to a sudden traffic variation.**
In this article, we propose an innovative and efficient distributed spectrum allocation algorithm, whose objective is to maximize the system UL capacity by exploiting multi-user diversity. The algorithm is capable of learning over time and of adapting the spectrum allocation when changes occur in the radio environment. Such an algorithm finds its application in the scope of future WLAN systems (e.g.: 802.11x).

***Index Terms*— swarm intelligence, distributed optimization, OFDMA, multi-user diversity, multi-channel opportunistic scheduling, cognitive radio**

I. Introduction

Several spectrum measurements campaigns have shown that in many countries, most of the spectrum bands are already allocated for use by wireless systems, but without being effectively used all the time and everywhere. Accordingly, to succeed in the introduction of future generations of wireless systems, we need to transition from a "spectrum scarcity" in the allocation to a model of "opportunistic spectrum availability". In collaboration with necessary modifications in the current rigid regulatory rules, technical progresses are also expected. Especially, intelligent Dynamic Spectrum Allocation (DSA) algorithms shall be developped with the capability to opportunistically identify and occupy "spectrum holes", and then release spectrum resources according to the needs.

Progresses in terminal equipments will soon make it possible to consider Cognitive Radio (CR)s [1] with the potential to observe their environment, orient their decisions, create plans, take decisions, and then act. CRs could cooperatively negotiate for spectrum access with others, thus creating a network of nodes dynamically using and releasing spectrum with no external guidance, except some spectrum policy rules.

A. Related Work

Assuming a pool of spectrum resources [2] (typically, a spectrum band shared by several entities), how can we best organize the transmission schedule and the assignment of spectrum channels to each user? We are dealing with researching intelligent Medium Access Control (MAC) algorithms, that would not waste spectrum resources by inappropriate reservations or by generating damaging interferences. We are interested in the study of Orthogonal Frequency Division Multiplexing (OFDM)-based standards (e.g.: IEEE 802.11a/g, etc) due to their possibility to slice a spectrum resource in closely packed elements not necessarily contiguous, and belonging to several different users (Orthogonal Frequency Division Multiple Access (OFDMA)). Previous works exist in trying to achieve a centralized OFDMA (for both the DownLink (DL) and the UpLink (UL)) [3] [4], but little work was found about distributed allocation of Sub-Carriers (SC)s.

We use biologically-inspired optimization and control methods offering a distributed design approach at the user level without external guidance. In this article, we design a multi-user cooperative algorithm achieving DSA in a Wireless Local Area Network (WLAN) cell. Specifically, our proposed algorithm achieves a dynamic and

Based on "Efficient OFDMA Distributed Optimization Algorithm Exploiting Multi-User Diversity", by J.C. Dunat, D. Grandblaise, and C. Bonnet which appeared in the Proceedings of the IEEE International Conference on DySPAN 2005, Baltimore, USA, November 2005.

flexible UL OFDMA SC allocation between users, while exploiting multi-user diversity. Users cooperate using a distributed approach, to maximize the UL cell sum capacity. As multi-user diversity is taken into account in the spectrum allocation, the system sum capacity increases with the number of nodes [5]. To our knowledge, no prior attempt of using the swarm intelligence meta-heuristic for SC allocation in this context was published, thus making it a new application of this very powerful generic method.

B. *Organization*

The remainder of this paper is organized as follows. The optimization problem is described in section II. Section III presents the advantages of a distributed optimization algorithm. Then, section IV describes the mechanism used to exchange information during the SC negotiation. Section V is dedicated to the variable threshold model description. Section VI contains the simulation scenarios and results. Finally, we conclude the article in section VII and discuss the future work.

II. Optimization Problem

First note that, in this article, we will refer to "users" as "nodes". Also, let us define as "central node" the infrastructure equipment (Access Point (AP)), and as "nodes" the user equipments (Mobile Terminal (MT)). Finally, we refer to "sum capacity" as the sum of the capacity per allocated SC over all the system SCs.

The optimization problem that we consider in this article can be described as follows (index i refers to the nodes and index j to the SCs). Given N available SCs to allocate among M nodes, each node i having a Signal to Noise Ratio (SNR) value γ_{ij} on SC j, finds the best UL allocation of nodes on SCs, to maximize the cell sum capacity, subject to some constraints described hereafter. Let us define c_{ij} as the Shannon capacity achievable by node i on SC j (for a unit bandwidth) which is: $c_{ij} = log_2(1 + \gamma_{ij})$. Let us define a_{ij} as the value used to indicate whether or not node i has selected SC j, with $a_{ij} = \{0; 1\}$. We are interested in the case of continous coding and modulation, independent between SCs. If we define n_i as the number of allocated SCs to node i, an "allocated node" has $n_i \neq 0$.

Using all these notations, the optimization problem can be formulated in mathematical terms as follows. The objective of the optimization problem is to:

$$\max \sum_i \sum_j a_{ij} c_{ij},$$
$$\text{s.t.:} \begin{cases} (1) \quad \{n_{min}, n_{max}\} \in [0; N], \\ (2) \quad \forall i,\, n_i = 0 \Leftrightarrow \displaystyle\sum_j a_{ij} = 0, \\ (3) \quad \forall i,\, n_i \neq 0 \Leftrightarrow \\ \qquad n_{min} \leq \displaystyle\sum_j a_{ij} \leq n_{max}, \\ (4) \quad \forall j,\; \displaystyle\sum_i a_{ij} \leq 1. \end{cases} \tag{1}$$

In other words, the objective is to find the final system allocation vector maximizing the system sum capacity, assuming all the constraints are respected.

Regularly, the central node updates the amount and position of the cell acquired SCs. This operation is based on the knowledge, at the central node as well as at the nodes, of the channel statistics for all the SCs in the band. To exchange spectrum resource (augmentation or reduction), each central node can negociate with other central nodes. Then, the central node broadcasts within the cell the positions of the new set of SCs to use. N is the spectrum size acquired/negociated by the central node. The values of n_{min} and n_{max} are controlled by the central node and can be tuned according to the entire available spectrum resource, channel statistics, etc.

We assume there exists link adaptation capabilities per SC such that nodes with good channel conditions will have a higher capacity than those with worst channel conditions. The n_i constraints (refer to equation 1) are requirements on the spectrum budget per allocated node with an objective to maintain a certain level of sum capacity per allocated node: n_{min} ensures a minimum sum capacity per allocated node, whereas n_{max} limits the overdimensioning of the necessary spectrum resource to run a given service in the best channel conditions. As such, each allocated node is ensured to have at least n_{min} SCs corresponding to a minimum sum capacity to run a given service. However, the algorithm does not try to equalize the capacity between the nodes.

As the considered optimization problem is NP-complete, there exists no algorithm to find the optimal solution in all cases in a polynomial time. Accordingly, to find a "good" sub-optimal solution we used an heuristic. In designing the algorithm, the compromise between the quality of the found solution (distance to the optimal solution) and the speed of convergence (time to obtain it) was taken into account.

III. Distributed Optimization

Distributed optimization methods possess many interesting properties including: scalability, robustness, adaptability to varying conditions. Also, distributed agents can be equiped with learning capabilites such as Reinforcement Learning (RL) [6] (see section V-B). Each agent runs the algorithm separately and by interacting with other agents, they all contribute to create a globally optimal solution, without the need for a central controller.

Solving the resource allocation problem with a distributed approach for the optimization has the advantage of parallelizing the optimization task among all the nodes. It reduces the complexity per node. Our algorithm is scalable with the number of nodes and able, without any prior knowledge of the number of nodes, to dynamically adapt to all the following scenarios: $N = M$, $N < M$ and $N > M$, as well as to any n_{min} and n_{max} values.

The scenarios envisaged consist in a single cell with a central node and several geographically dispersed nodes in the cell. The nodes have full responsibility in choosing their set of SCs to use for transmission, whereas the central node does not influence the nodes' allocation. In that sense, the algorithm is distributed and the power of decision is spread between the nodes. The optimization is achieved by cooperation between geographically separated nodes. Such a distributed approach is of particular interest when the number of attached nodes becomes large.

We assumed a low mobility of the nodes, a known channel at the receiver, a synchronized UL reception of the messages and a symmetric channel with not too rapid variations. This is reasonable for WLAN systems with small cell sizes.

A SC reallocation is achieved (if necessary) for every data packet, but the frequency of SC reallocation can be varied by the operator, depending on the rate of change of: the channel, the availability of new SCs, the cell traffic, etc. The optimization algorithm consists in an opportunistic spectrum access at the node level.

The time is decomposed into frames of variable duration $T_F = T_N + T_D$. Each frame comprises a negotiation part (variable duration T_N) and a data part (fixed duration T_D). The negotiation part consists in searching for the optimized allocation, whereas the data part consists in the transmission of data frames. We assume that SNR values do not change during T_F. Figure 1 presents the timeframe context of the optimization algorithm.

The negotiation part consists in a succession of UL (phase (1)) and DL (phase (2)) messages between the nodes and the central node. During phase (1), each node contending for a given SC transmits to the central node an UL burst message on that SC. All UL burst messages are simultaneously sent. Then, during phase (2), the central node broadcasts in the cell a DL feedback message (using a dedicated channel) solely containing the minimum and maximum capacity value found on each SC. These two phases are repeated during the negotiation part until the allocation is finished. Then, the data part takes place, during which a data frame can be transmitted by the nodes using their ON SCs.

The next section develops the wireless communication mechanism used to achieve an efficient distributed control.

IV. Mechanism of Information Exchange during Negotiation

This section explains the principles and the practical implementation of the inter-node communication mechanism, and the conflicts resolution.

A. Aggregation Mechanism

In a distributed system, the following questions have to be answered to exchange information between entities: (1) what information to include in the messages? and (2) how to communicate (inter-node communication mechanism)?

To answer question (1), a simple but efficient duration-coding technique was used: each c_{ij} value is coded as the duration T_{ij} of a burst message (transmitted in UL) such that: $T_{ij} = \frac{a}{c_{ij}} + b$. Thus, the greater c_{ij}, the shorter T_{ij}.

Node index	cij	Tij	Rx signal strength (RSS) at central node from node i
1	40	1/40	3
2	10	1/10	0.5
3	20	1/20	2

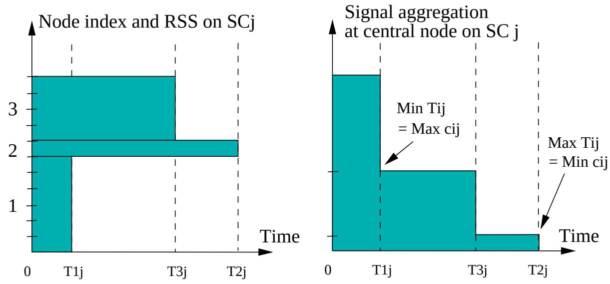

Fig. 2. Capacity Duration-Coding with 3 Nodes on SC j

To answer question (2), all contending nodes simultaneously (same Tx start time) use an UL transmission scheme. Even though figure 2 presents an example with a single SC, 3 nodes, $a = 1$, $b = 0$, and fictitious numerical values, the explanation suffers no loss of generality.

As shown in figure 2, 3 users contend for SC j, each with a different signal level at the receiver and c_{ij} value (refer to the table located at the top of the figure). The left part of the figure indicates the time-coding of c_{ij} into T_{ij}, and the Received Signal Strength (RSS). The right part of the figure shows the resulting signal aggregation at the receiver, with the detection of the minimum and maximum values.

All UL signals sent aggregate independently per SC. Then, the central node uses the aggregated signals to extract the minimum and maximum coded c_{ij} values. In practice, the central node easily finds the maximum value by detecting the first drop (simple gap detection rule) in the aggregated signals. The duration between the start of the UL reception and the first drop gives the corresponding transmitted value. By detecting the longest message duration, the central node finds the minimum value. As the optimization progresses, only the best values remain. A DL message is sent only after the end of the longest burst message. An advantage of our method is the simplicity of the coding and decoding, simultaneous and parallel exchange of information on the same radio resource, robustness against interference (if constant during the UL time slot) and collisions, and it does not require an orthogonal coding to separate nodes or capacity values.

To coordinate their allocation, the nodes do not communicate directly with each other, but only require to exchange information with the central node. The central node acts as a "filter" extracting the maximum and the minimum values out of all the received values, and sending it back to the nodes for information sharing.

WLAN systems with a distributed control (e.g.: IEEE 802.11 in DCF mode) suffer performance loss due to the hidden node problem. Out of range nodes are said to be hidden nodes. They have no possibility to communicate to avoid UL collisions on a shared channel. Our mechanism

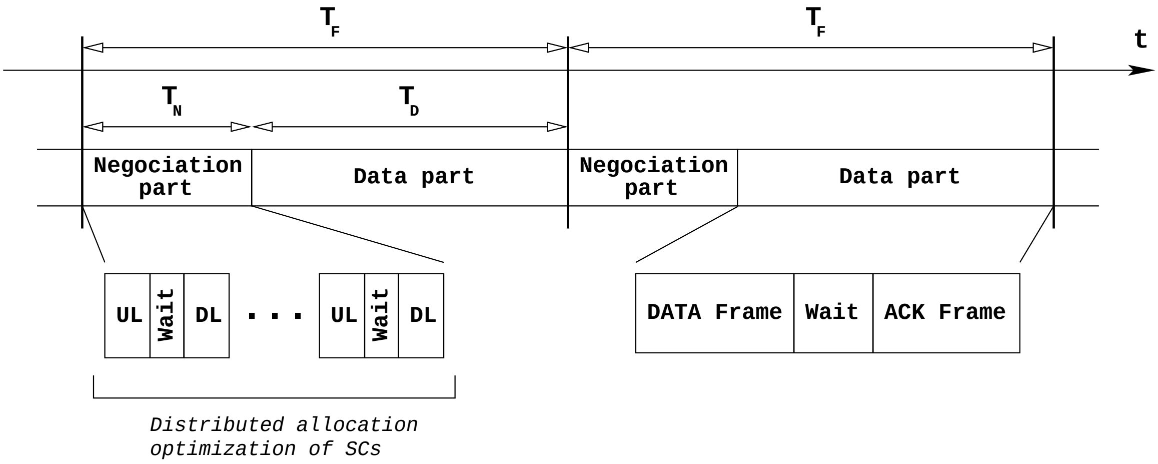

Fig. 1. Description of the Optimization Algorithm Parts

of inter-node exchange of information is built to encourage and manage message overlap (aggregation of signals) even between hidden nodes. The consequence is no more collision during the data transmission. Accordingly, the problem of hidden nodes is completely removed and there is no risk of performance degradation due to collisions.

This method has the advantage of being robust and simple, thus perfectly fitting in the swarm intelligence context.

B. Variable Quantification Step

We now show the practical implementation aspects of our communication mechanism. Before reaching the final SC allocation, several UL/DL time slots are usually necessary. The UL and DL negotiation time slots have a respective duration of T_{UL} and T_{DL}. To bound the UL burst messages duration, the c_{ij} values were quantified into a fixed number K of elements. However, the resulting precision is not sufficient to differentiate close values. Accordingly, to increase the precision, we used a variable quantification step per SC with a zoom window. The principle is explained in figure 3 (each triangle and cross represents the non-quantified c_{ij} value of node i on SC j).

Figure 3 represents the evolution over time (iteration after iteration) of the range of expressed quantified c_{ij} values (i.e. $q(c_{ij})$ values) in which the final best values will be located. At each iteration, the central node:

1) Collects the UL received quantified extreme values (minimum and maximum) per SC. In figure 3, the minimum (resp. maximum) value corresponds at each iteration to the left (resp. rigth) corner of the more at the left (resp. right) square. A square representing the range of different c_{ij} values having the same quantified value $q(c_{ij})$ due to the quantification step.
2) Adjusts the extreme window limits around these extreme values.

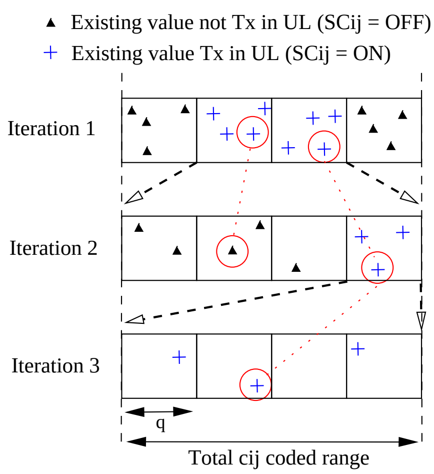

Fig. 3. Variable Quantification on SC j

3) Broadcasts to the nodes the new window limits and quantification step q. Indeed, q is changed after each zoom/dezoom operation.

This variable quantification process uses a zoom window to ensure a shorter negotiation duration, and to concentrate the precision of the quantification only for the current expressed values.

If after a period of time $T_{timeout}$ no final convergence was obtained, the central node imposes to stop the optimization process and forces all the nodes to apply a quick solution. Thus, this technique bounds the negotiation duration. After the timeout was reached, an acceptable solution can be found in two UL/DL time slots.

The next section describes the meta-heuristic used to solve the optimization problem.

V. Variable Threshold Model for Allocation

The objective of this part, the most important one, is to achieve the distributed optimal allocation of SCs among the nodes.

We assume SNR values do not change during the negotiation part, which is reasonable if the frame duration is not too long. It is to be noted that even if SNR values would change, the algorithm would still work properly, but the convergence time might be longer.

At the end of the optimization phase, there is only a single specialized node per SC, but a node can be specialized in several SCs. During the optimization process there is both an inter-node negotiation (to respect constraint 4) and an intra-node negotiation (constraints 2 and 3). Indeed, each node would want to use all the n_{max} SCs but by means of local interactions it must be optimally shared among all the nodes to globally maximize the cell UL sum capacity.

A. Introduction

To solve the considered optimization problem we implemented a general heuristic inspired from a model of division of labor used by social insects such as ants, wasps, etc. This heuristic comes from the domain of biology (studies of social insects) then modeled by the domain of artificial intelligence, improved and successfully applied to solve real world optimization problems, using a set of cooperating agents. This heurisitic was proven to be very efficient to optimize real life problems of dynamic and distributed tasks allocation in the industry [7], [8].

This method was found to give good results for the dynamic and distributed task allocation problem. Even though the obtained results on well-known optimization problems are very good, the underlying theory is still in its infancy [9] (theory of cooperating agents). All the agents use the same finite set of simple rules to interact with each other. The result of this cooperation is an auto-organization at the system level. A modelling challenge is to find the appropriate set of local rules that will globally create the desired behavior.

We managed to successfully adapt and use this heuristic in our problem with good results, thus showing its great potential for optimization and control of cooperating agents.

The method is inspired from social insects where the ability of an agent to perform a task depends on its ability for this task, as well as the number and ability of other agents willing to perform this task. Thus, agents "specialize" over time in the tasks where they are best fitted for, by means of a variable threshold, as well as by inter-agent conflicts (dominance contests).

This model of variable threshold is able to appropriately adapt to very dynamic environments. An improved version [10] uses a moving threshold with time of use. Agents receive stimuli from tasks in order to perform a task. They effectively perform the task according to a given probability, function of their threshold regarding this type of task.

The more an agent becomes specialized in a given type of task, the greater the probability to engage in doing it for future stimuli of this type. Each agent maintains and updates over time a threshold per type of job to be done.

B. Modelling and Description of the Heuristic

A node contains (1) a constraint controller and (2) a set of N cooperating agents (one by SC). The constraint controller ensures that the contraints in number of active SCs per node are always ensured. On the contrary, the main objective of an agent is to stay active (ON) at the end of the optimization process. Agents interact internally (with other SC-agents within the same node) and externally (with agents representing the same SC index but from other nodes). The rules of interactions of agents between them, are build to reach a global optimum by using local interactions, as in auto-organisation. While they interact, agents exchange information about capacity values (see section IV).

To become stronger and have better chances to stay active, agents organize in groups (within each node) and try to find the best other members of their group, for them to be ON at the end. Groups are an aggregation of each agent force: the group capacity is the sum of the capacities from all agents engaged in this group. The weakest agents are left apart and groups are reconstructed until the end of the optimization process. Groups are organized and re-organized dynamically and autonomously to reach the best final configuration for the system. Thus, groups have a variable size over time.

We assume that a node is able to simultaneously perform n_{max} tasks out of a total of N tasks. An agent is able to perform only a single task, and it can either perform it (ON status for that SC) or not.

In our model, an agent's threshold value changes over time and its value depends on (1) its previous history, (2) its ability to perform its task better than the agents from the same node (intra-node contests) and (3) its ability to perform its task better than the agents from other nodes for that same task (inter-node contests for the same SC).

The amplitude of change of an agent's activity threshold is variable and is a function of the agent's force compared to others agents for this task. The threshold value modifies the probability of an agent to perform its task in the future, such that: a low threshold leads to a high probability to perform its task, whereas a high threshold means a low probability to perform its task.

In optimization, there is an important trade-off between the exploitation of the current solution and the exploration of new solutions. The chosen heurisitic, owing to its variable threshold model, adapts well to a dynamic environment and leads to faster convergence (but not premature) than other heuristics. It is also very efficient in avoiding local optima. It uses a simple RL algorithm described hereafter. In addition, our model is able to adapt to a large set of constraints. As a result, the same algorithm solves the problem of SC allocation for a range of SCs per node from 1 up to N. The particular case where a single node can capture all N available SCs can be interpreted as a new allocation method for a single channel system.

The optimization starts with each node setting to ON its n_{max} SCs with the best capacity values and then, the algorithm tries to improve this solution. Then, we repeat Steps 1 to 4 (see below) until all the optimization constraints are enforced, and the found solution is considered as the final one. These steps are presented hereafter:

Step 1: The threshold variation $\Delta\theta_{ij}$ is updated after each DL time slot for each node i on SC j, as follows (using a modified Fermi-Dirac distribution).

$$\Delta\theta_{ij} = 2\varphi \cdot \left(\frac{1}{1 + e^{\beta_\theta (x_{ij} - x_{max\ j})}} - \frac{1}{2} \right). \qquad (2)$$

This function was chosen for its similar characteristics as in [10]. Either smooth or abrupt probability transitions can be flexibly generated by tuning β_θ (varies the transition's steep which impacts the amplitude of $\Delta\theta_{ij}$ variations) and φ (varies the user's memory depth which impacts his learning and forgetting speeds).

In DL, the central node broadcasts $c_{max\ j}$ (the maximum quantified capacity value extracted among all UL contenders on SC j) such that: $c_{max\ j} = \max_i\{q(c_{ij})\}$, where c_{ij} and $q(c_{ij})$ are respectively non-quantified and quantified capacity values. Only quantified values are transmitted in UL by the users, and returned in DL by the central node.

To transform $c_{max\ j}$ into $x_{max\ j}$ (see figure 4), a scaling operation is performed between $[c_l, c_u]$ and $[x_l, x_u]$ using: $x_{ij} = \frac{\Delta x}{\Delta c}(c - c_l) + x_l$, with $\Delta x = x_u - x_l$ and $\Delta c = c_u - c_l$. Note that $"."_l$ stands for "lower possible value" and $"."_u$ for "upper possible value" and applied to both x_{ij} and c_{ij} values.

To update threshold variations between the users, each SC uses its separate function (refer to equation 2), centered at the maximum returned value such that: $\Delta\theta_{\arg\{c_{max\ j}\}\ j} = 0$ corresponding to $x_{max\ j}$. Finally, each node transforms its c_{ij} value into x_{ij}, and then computes its threshold variation using equation 2. The total capacity range $[c_l, c_u]$ is given by the system and the $[x_l, x_u]$ x range is calculated.

This is best explained through an example. Thus, figure 4 will illustrate all these concepts. As an example, and without any loss of generality, we take the following numerical values: $\varphi = 5$, $\alpha = 1$, $\beta = 1$ and $\epsilon = 0.001$. Accordingly, we obtain: $[x_l : x_u] = [-9.21 : 9.21]$.

Figure 4 shows the variation of the thresholding function with $c_{max\ j}$, and presents an example of $\Delta\theta_{ij}$ calculation for $c_{max\ j} = 8$ and $c_{ij} = 7.5$. This figure also includes several curves corresponding to different α values.

Using a threshold variation based on the maximum value is a very efficient and scalable method for discriminating a large number of values. As described in [11], there can be an important gain in using an optimized exchange of information when using multi-user diversity. Accordingly, we obtain:

$$\forall i, \text{ on SC}j: \begin{cases} c_{ij} < c_{max\ j} & \Rightarrow \Delta\theta_{ij} > 0, \\ c_{ij} = c_{max\ j} & \Rightarrow \Delta\theta_{ij} = 0, \\ c_{ij} > c_{max\ j} & \Rightarrow \Delta\theta_{ij} < 0. \end{cases} \qquad (3)$$

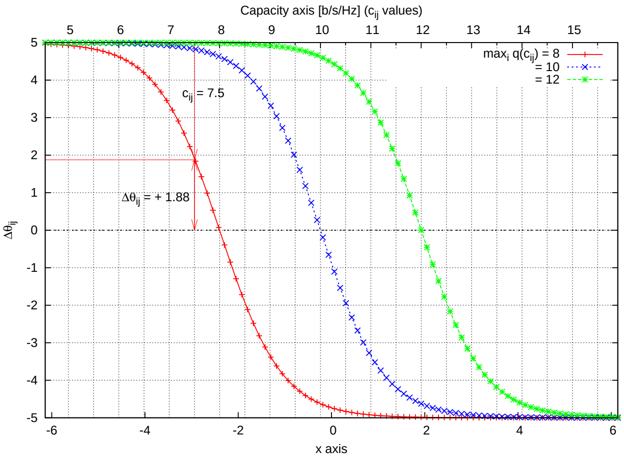

Fig. 4. $\Delta\theta_{ij}$ Variation for Node i on SC j

By properly adjusting the numerical values of φ and β_θ we can control the speed of convergence and the quality of the solution. As a result, Step 1 is responsible for performing an update of the immediate threshold value for SC j of user i.

Step 2: Once the threshold variation is known (amplitude and sign) it is aggregated with the previous threshold value such that:

$$\theta_{ij} \leftarrow \theta_{ij} + \Delta\theta_{ij}. \qquad (4)$$

Step 2 is responsible for updating and storing the history of each θ_{ij} value by compressing all the previous values by aggregation into a single new θ_{ij} value.

Step 3: Each agent updates its status (ON or OFF) as a function of its threshold θ_{ij} using a Fermi-Dirac distribution. The probability p_{ij} that an agent i performs its task j is given by equation 5:

$$p_{ij} = \frac{1}{1 + \alpha_s e^{\beta_s \theta_{ij}}}. \qquad (5)$$

with α_s and β_s some adjustable parameters.

Figure 5 presents some examples of resulting curves for various sets of combined $\{\alpha_s; \beta_s\}$ values. For each SC j, a node i draws a random value $z \in [0; 1]$. If $z \leq p_{ij}$, then SC j will be ON at the next negotiation iteration, otherwise it will be OFF. In other words, Step 3 acts as a gate for SC status: its decides whether or not a given SC will become or stay active. In the following studies good results were obtained with these values set to: $\alpha_s = 0.5$ and $\beta_s = 2.0$.

Having the possibility to reshape (by tuning some control parameters) both the function controlling the $\Delta\theta_{ij}$ amplitude variations (Step 1) and the function controlling the SC status change (Step 3) brings flexibility to the algorithm.

Step 4: The constraint controller controls if n_i is within the acceptable values. If $n_i < n_{min}$ and $n_i \neq 0$: all the agents in node i set their SCs to OFF. On the contrary, if

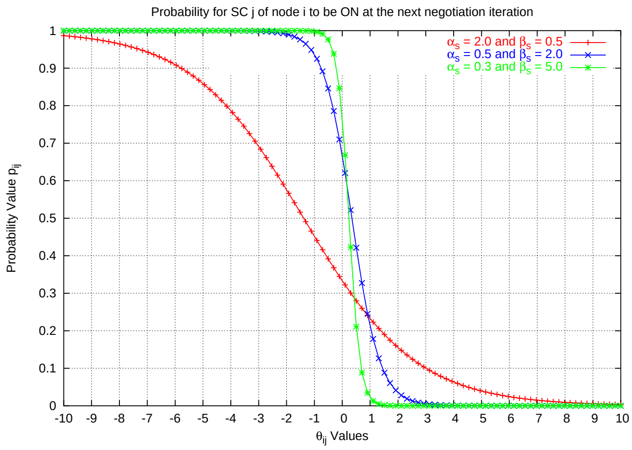

Fig. 5. Examples of curves with various $\{\alpha_s; \beta_s\}$ values

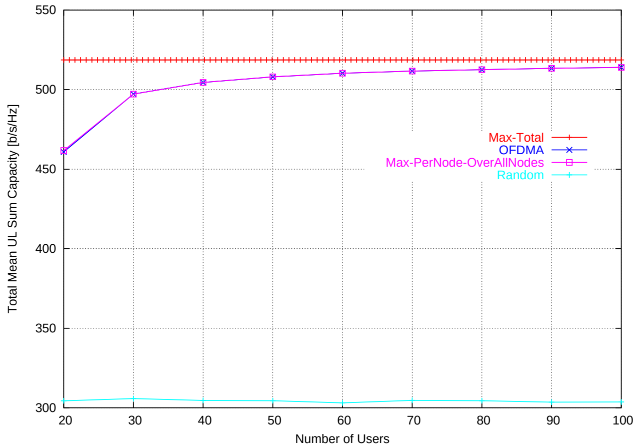

Fig. 6. Compared Total Sum Capacity as M Varies

$n_i > n_{max}$: node i chooses the best n_{max} SCs with the greater probability of being ON.

Then go back to Step 1 until the end of the optimization process. Our algorithm contains several global and easy to control parameters, allowing to tune the quality of the solution as needed. The variable threshold model offers a great dynamicity in the optimization process, even in changing conditions, and is well adapted to a distributed implementation.

VI. Simulation Scenarios and Results

The numerical values used to obtain the following examples are modified values taken from an IEEE 802.11a system. However, other values could be taken without altering the conclusions.

Taking as an example an 802.11a system [12] in the band $[5.25-5.35]$ GHz, we calculated the maximum range of SNR values for any cell node. The maximum range of capacity values per SC was deduced. After calculations and taking into account the maximum transmit power regulations, the SC bandwidth and the set of possible transmit modes, we obtained that all SNR values γ_{ij} $(\forall i, j)$ are within: $[\gamma_{min}; \gamma_{max}] = [13.5; 78.07]$ dB . In our study, only the continuous case is considered. We can now transform these SNR values to find out the maximum capacity range c_{ij} $(\forall i, j)$ are within: $[c_{min}; c_{max}] = [4.54; 25.9]$ b/s/Hz (for a SC bandwidth of 1 Hz). All c_{ij} values are uniformly drawn within $[c_{min}; c_{max}]$. Note, that the aim of this study was not to focus on the channel model, but rather to demonstrate the benefit of the proposed algorithm.

A. Comparison with Other Methods

Our method is compared with other methods in terms of total reached UL sum capacity after completing an allocation, for a varying M in the cell. The following methods are used for comparison:

- *Max-Total*: this method is intended to give an upper bound for the total sum capacity. Indeed, it is a fixed level of sum capacity corresponding to $N \times c_{max}$, as if you could allocate N nodes each one having the maximum allowable capacity value c_{max}. However, this level cannot be reached in reality.
- *OFDMA*: this is our distributed method.
- *Random*: for each SC, a random node is taken to be allocated to this SC. In other words, there is absolutely no intelligence in this choice. It is similar to doing a "blind" choice.
- *Max-PerNode-OverAllNodes*: for each SC we allocate the user having the maximum c_{ij} value over all the other users. Then, this user is removed from the future choices, and the decision goes on with the next SC until the complete allocation of all the SCs.

The following numerical values are used: $M = 20 : 10 : 100$, $N = 20$ and $n_{min} = n_{max} = 1$. A total of $K = 1000$ DATA frames are transmitted for each M value. For each M value, all the methods calculate the mean total UL sum capacity as a result of the K transmitted DATA frames.

Figure 6 presents the obtained results.

As expected, the *Random* method constantly gives the worst results (lower bound for the total sum capacity). In addition, both the *Max-PerNode-OverAllNodes* and the *OFDMA* methods result in an increasing capacity as M increases due to having more users with better channel conditions. These two methods asymptotically tend to the *Max-Total* level representing the best possible results. The methods *Max-PerNode-OverAllNodes* and *OFDMA* give comparable results in this scenario. However, the *Max-PerNode-OverAllNodes* method would correspond to a centralized method, whereas our *OFDMA* method is a distributed method in which users do not know about each other. Accordingly, our algorithm gives very good results when compared to some other methods and can be implemented in a distributed way.

B. Algorithm Adaptation to Changes in number of contending nodes

In this section we show the good adaptation capabilities of the algorithm to a load variation in terms of quality of

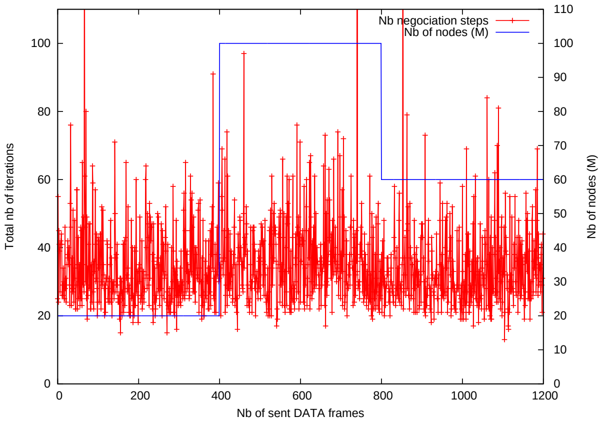

Fig. 7. Number of Iterations for $N = 100$

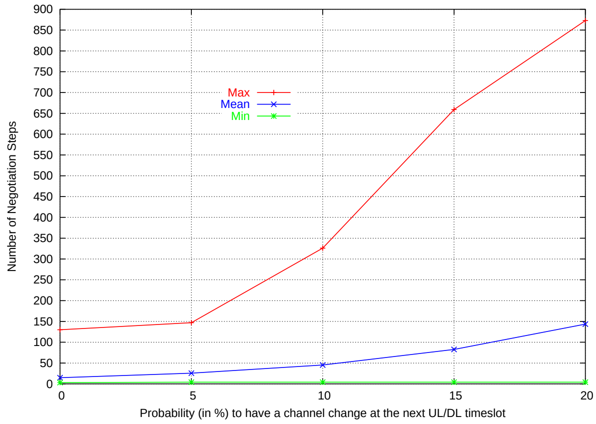

Fig. 9. Channel Variations within Negotiation Phases

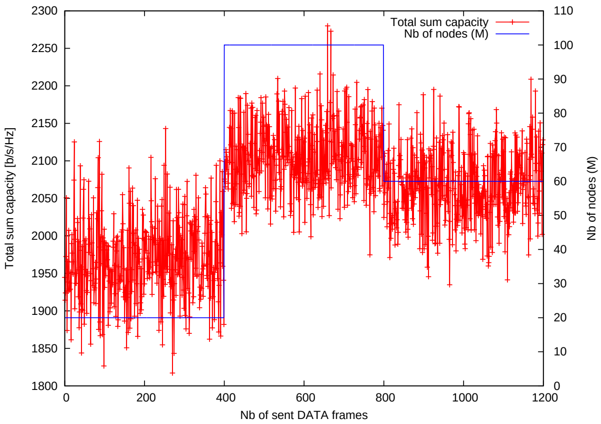

Fig. 8. Total Sum Capacity for $N = 100$

the reached solution and convergence time. The number of contending cell nodes M is abruptly changed during the simulation.

The following parameters were used: $N = 100$, $n_{min} = 1$ and $n_{max} = 10$. The total number of transmitted DATA frames over time was $D = 1200$. M was abruptly changed as follows: at $D = 0$, $M = 20$, then at $D = 400$, $M = 100$ and finally at $D = 800$, $M = 60$, figures 7 and 8 show the adaptation of the algorithm to the modified cell load (in number of nodes).

Several conclusions can be derived from the results presented in figures 7 and 8. The agents were successfully able to adapt to a modified (increase and decrease) number of cell nodes. The number of negotiation steps remained small (mean value = 34.8), considering the optimization problem size. Also, the number of negotiation steps remained almost independent from M variation and constant even for a population multiplied by 5 and then decreased by $\frac{5}{3}$. The number of negotiation steps does not appear to depend on M, thus proving the scalability of our distributed algorithm and its great interest.

In figure 8, the obtained system sum capacity was increased with an increased value of M, proving the multi-user diversity interest to increase the system sum capacity.

C. *Channel Variations within Negotiation Phases*

In this section we study the potential for the algorithm to adapt to channel variations at random instants during the negotiation phases, in terms of convergence speed. The following numerical values are used: $M = 100$, $N = 20$, $n_{min} = n_{max} = 1$, a total of $K = 1000$ DATA frames are transmitted.

We vary the probability to have a channel change at the next UL/DL timeslot. A channel change corresponds to an abrupt change for all SCs of all the users (contending or not). Some contending users might switch to non contending or the contrary. The entire system changes. In addition, the θ_{ij} are not reinitialized. Rather, we let the algorithm discover the change by itself and react to find the new best solution as quickly as possible before the next change occurs.

Figure 9 presents the maximum, mean, and minimum number of negotiation steps for each value of channel change probability.

As seen in figure 9, increasing the number of channel changes results in a longer mean negotiation phase duration. In fact, an increased number of changes increases the negotiation duration, which in turn increases the probability of having another change before convergence, etc. This is explains the important increase in maximum negotiation length. Thus, there is an avalanche effect. In case of an abrupt change, resetting the θ_{ij} values after a change would gain some time in negotiation, rather than trying to fully forget and readapt to the new input values. There is tradeoff to consider.

VII. Conclusions and Future Work

Our algorithm is very attractive in that it has shown the following properties: scalability with the number of competing nodes, agents auto-adaptation (self-learning) to react to changing conditions, robustness, simplicity of the implemented rules at the user level, ease for the operator to change the optimization constraints and thus the network characteristics. As the swarm intelligence is a metaheuristic, it can be used to solve diverse problems related to the wireless world. Especially, the swarm intelligence is very promising in the context of dynamic spectrum allocation. Spectrum wastefulness can thus be minimized while the fulfilling the users' needs. These algorithms are intended for implementation in future cognitive radios using next generations of WLAN systems, to opportunistically use the appropriate part of the spectrum according to the needs. Some practical design considerations have been presented to improve the algorithm's efficiency compared to existing solutions.

Accordingly, regulators should have no fear in putting more flexibility in their rules governing spectrum allocation. Indeed, technology will soon be offering sufficient garanties that a self-organizing spectrum planning can replace the rigid and under-optimized command-and-control model.

Including a more realistic channel model and provide the agents with the capability to anticipate the channel variations could help in reducing the convergence duration and improve the performance.

REFERENCES

[1] J. Mitola "Cognitive Radio - An Integrated Agent Architecture for Software Defined Radio", *PhD Thesis of the Royal Institute of Technology*, May 2000.

[2] F. Capar and al. "Comparison of Bandwidth Utilization for Controlled and Uncontrolled Channel Assignment in a Spectrum Pooling System", *IEEE VTC'02*, March 2002.

[3] I. Toufik "Channel Allocation Algorithms For Multi-Carrier Systems", *VTC*, 2004.

[4] F. Capar and F. Jondral "Resource Allocation in a Spectrum Pooling System for Packet Radio Networks Using OFDM/TDMA", *IST Mobile & Wireless Telecommunications Summit 2002*, June 2002.

[5] R. Knopp "Achieving Multiuser diversity under Hard Fairness Constraints", *Proceedings. 2002 International Symposium on Information Theory*, 2002, pp. 451.

[6] L.P. Kaelbling, M.L. Littman, and A.W. Moore "Reinforcement Learning: A Survey", *Journal of Artifical Intelligence Research*, pp. 237-285, May 1996.

[7] E. Bonabeau, G. Theraulaz and J.L. Deneubourg "Adaptive task allocation inspired by a model of division of labor in social insects", *In D. Lundh and B. Olsson, editors, Bio Computation and Emergent Computing*, pp.36-45, World Scientific, 1997.

[8] V.A. Cicirello and S.F. Smith "Improved routing wasps for distributed factory control", *In IJCAI-01 Workshop on Artificial Intelligence and Manufacturing: New AI Paradigms for Manufacturing*, August 2001.

[9] M. Fleischer "Foundations of Swarm Intelligence: From Principles to Practice", *Swarming Network Enabled C4ISR*, 2003.

[10] V.A. Cicirello and S.F. Smith "Distributed Coordination Resources via Wasp-like Agents", *In WRAC*, January 2002.

[11] D. Gesbert and M.-S. Alouini "How much feedback is Multi-User Diversity really worth?", *ICC 2004*.

[12] "IEEE 802.11a. High Speed Physical Layer in the 5 GHz band. Supplement to Standard IEEE 802.11", *IEEE*, New York, 1999.

J.C. Dunat was born in 1976 in France. In 2000 he received his engineering degree in Electrical and Computer Science Engineering from Ecole Supérieure d'Ingénieurs en Electronique et Electrotechnique in Paris, France.

He recently finished a PhD with Institut Eurecom and Motorola Labs. He is now working for Motorola MDB on the integration of new technologies in handsets.

Mr. Dunat's research interests cover wireless system performance modelling, distributed and cooperative systems, dynamic spectrum access algorithms, and wireless LANs systems. Also, he has contributed to the E2R Project on spectrum management. He has several publications pending in these areas.

D. Grandblaise received his engineering degree in Electrical Engineering and M.Sc degree in mobile radio communications from Ecole Spéciale de Mécanique et d'Electricité and Ecole Nationale Supérieure des Télécommunications (both in Paris, France) in 1998 and 1999, respectively.

In 2000, he joined Motorola European Communications Research Lab (Paris) where is currently Senior Research Engineer in the Seamless Radio Access Lab. After carrying out initial research activities on physical and system aspects of UMTS transmit diversity techniques, his research activities focus now for several years on cognitive radio and reconfigurable radio communication systems.

Mr. Grandblaise's research interests cover wireless system performance analysis for flexible spectrum management and radio resource management. He is actively contributing to IEEE 802.16 and 802.22 standards. Also, he has contributed to European IST research projects including TRUST, OverDRiVE. He is currently involved in IST E2R Project on spectrum management. He has several publications and patents pending in these areas.

C. Bonnet was born in Paris on September 17, 1955. He received an engineering degree from Ecole Nationale des Mines de Nancy in 1978.

He was a consultant in the GSI group for 9 years where he worked on different projects related to radio telecommuncations, value added networks and real-time networks. He participated to the European projects (ESPRIT) DESCARTES, DRAGON, REX. He was appointed Director of the Real-Time Department of GSI Tecsi in 1987. He joined Institut Eurecom as an associate professor in 1992. Since 1998 he is at the head of the Mobile Communications Department of Eurecom. He is participating in research projects related to UMTS in the field of QoS and IPv6: (IST) Mobydick, (RNRT) SAMU, PLATON, @IRS++. From 1992 to 1998, he has been the team leader of Eurecom for the European project (ACTS) WAND (Wireless ATM Network Demonstrator) and responsible for projects in wireless network simulations (GSM) and wireless LAN interconnection with ATM.

His teaching activities are distributed and real-time systems, mobile communication systems, wireless LANs and protocols, wireless access to IP networks and data communications in mobile networks including mobile ad-hoc networks.