Equity between secondary users in a cognitive radio network

[1] Abdellah IDRISSI and [2] Said LAKHAL
Research Computer Sciences Laboratory (LRI)
Computer Sciences Department, Faculty of Sciences
University Mohammed V - Agdal, Rabat
E-mail: [1] idriab@gmail.com , [2] said.lakhal.rech@gmail.com

Abstract: - Ensuring fairness in the allocation of free channels to Secondary Users (SUs) in a Cognitive Radio Network (CRN) is a big problem for researchers. In this work, we develop a new algorithm, explaining the parameters of channel assignment, by identifying the SU, the number of packets to send and the slots and channels to use. To calculate the effectiveness of the algorithm, we use the Equity Index Jain (EIJ) as an indicator of performance. The results show values of EIJ always close to 1, which proves the effectiveness of the algorithm.

Key-Words: - Cognitive Radio Network; Scheduling; Equity; Equity Index Jain; Transfert rate; Cantor's Bijection.

1 Introduction

Communication systems are currently experiencing a great development. Since their birth, several technologies have emerged. This diversity raises the question of heterogeneity and the possibility of coexistence. To solve this problem, the Software Defined Radio (SDR) approach provided an effective solution, crowned by the production of more and more intelligent communication systems, of which Cognitive Radio (CR) is the fruit [1].
In 1996, the first discussions on reconfigurable radio began in the SDR forum; the goal then was to add more intelligence and flexibility [2].

To understand the principles of the CR, we propose to restate the definitions given by the Federal Commission (FCC)[1], the IEEE 1900.1[2], and Joseph Mitola[3].
FCC: The CR is a system with a transmitter/receiver capable of understanding the environment, detecting unused spectrum by Primary Users (PUs), changing the transmission parameters, and adapting and sending packets of Secondary Users (SUs), while avoiding the collision with the PUs [2].
IEEE 1900.1: The CR is a type of radio which can detect the parameters of its environment and adapt dynamically and automatically [2, 5].
Joseph Mitola: In 1999, he defines the cognitive radio as follows: "A radio that can find, collect and learn from its environment and act to simplify the user's life" [1].

Based on these definitions, we would like to say that CR is a communication system with a transmitter/receiver capable of knowing its environment and automatically and dynamically adapting to transmit the data of SUs, while avoiding collisions and interferences with the PUs and among SUs themselves.

In the CR, there are two ways to access the spectra:
1) Dynamic access: the environment of the CR changes dynamically. To adapt to these changes, the SU must adapt and change as necessary its transmission parameters [4].
2) Opportunistic access: The SUs waits for the release of spectrum by PUs. CR provides all free spectra of SUs [14].

We are currently witnessing a major change in terms of diversity of services made possible by

[1] Federal Communications Commission

[2] Standard Definitions and Concepts for Dynamic Spectrum Access: Terminology Relating to Emerging Wireless Networks, System Functionality, and Spectrum Management

[3] The first to formally presented the idea of cognitive radio

the CRNs. The research focuses on the ideal use of available resources (spectra, energy, space and time, etc.) to achieve maximum throughput for SUs [13], avoid collisions and interference between the SUs [15, 16] and PUs and finally ensure equity between SUs [3, 6, 8].

To achieve these objectives, we think we should set up a scheduler able to deal effectively, automatically and dynamically with the momentary changes of the environment [17].

The scheduler must detect the free channels, select the channel to use, modify the parameters of the SUs for a better adaptation, and allocate these channels equally between the SUs which request the service [3, 6, 8].

Within this context, we propose here a Multi_Scheduler algorithm that responds to this kind of problems.

In the next section we expose related work. In section 3, we develop our approach. We present applications and experimentation results in the section 4 and conclude in section 5.

2 Related work

We haven't found convincing answers to the following questions in the literature: Which sends? How many packets? On what and how? This is to identify the sender, the number of packets being sent, and the channel and the procedure of transmission.

In [3], the authors have proposed a scheduling algorithm called Mono_Scheduler which consists to affect only a single channel to all Secondary Users (SUs). This algorithm returns a chain scheduling containing the order in which the packets will be sent. The algorithm guarantee some equity value between SUs, but it only treats the case of a MonoChannel CRN, the MultiChannel case was not treated. To calculate equity, the authors used the standard deviation as a performance indicator, such an indicator, admits a unit and depends on the size of the treated sample. By cons, the EIJ does not admit of unit and does not depend on the size of the selected sample. In [5], the authors study the coordination for the purpose of cohabitation of users while ensuring the QoS of SUs and PUs. QoS is expressed in terms of the absence of collision with PUs, improved throughput and fairness to the SUs. Nevertheless, the authors do not present the way the channels are assigned in each fraction.

In [6], the authors study the optimal energy consumed by each SU and distribution rates to ensure fairness on the basis of the current and previous CRN states, but they do not show the procedure for assigning channels to SUs. In [7], the authors present a scheme providing a dynamic spectrum sharing. By cons, they do not take into account fairness between SUs.

In [8], the authors propose an algorithm for allocating resources to maximize throughput and provide fairness between SUs. The work does not clearly express the procedure of allocating channels. In [9], the authors propose a scheme of scheduling an opportunistic spectrum. The proposed algorithm estimates the number of packets to send by each SU through each channel and maximizes the transfer rate. But they did not address equity between SUs and the problem of channel allocation based on the number of packets of each SU. In [10], the authors have developed a scheduling algorithm, to maximize the transfer rate of SUs, avoiding any collision with the PU. The problem of equity between SUs in terms of channel access was not studied, again the allocation process channels to SUs was not shown (during each slot, determine which among SU can transmit, and the number of packets to be transmitted). In [13], the authors studied the problem of maximizing transfer rate of SUs, wherein each SU can detect a limited number of channels. The authors have shown that the problem is NP-hard and proposed an algorithm of approximation. The channel assignment process was not reported. Also, the fairness was not mentioned despite it is an important criterion. In [17], the authors have developed a scheduling algorithm that estimates the number of packets to send during each slot. A central scheduler selects SUs who need to send their packets. We find that, the channel assignment procedure is implicit and the authors didn't present the problem of equity between SUs.

Our contributions include developing a new scheduling algorithm to allocate the free channels to SUs, adapting the Mono_Scheduler algorithm [3] on a MultiChannel and MultiPU in CRN, using the Cantor's Bijection (CB) to represent a vector with integer components with a single integer, and obtaining a significant equity between SUs based on the Equity Index Jain (EIJ).

3 Our approach

In this work, we study a CRN MultiChannel, MultiPU and MultiSU. Figure 1 illustrates this approach.

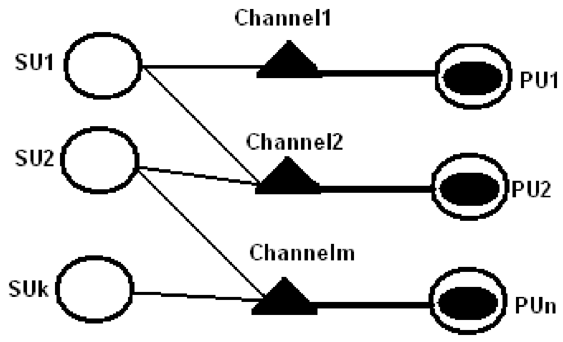

Fig.1: CRN with Multichannel, MultiPU and MultiSU.

The PUs have priority access to channels. If the SUs want to connect, they must await the release of channels by PUs.
Like this, the problem of collision between the PUs and SUs no longer arises. But there remains the problem of equity between SUs. To establish this equity, we define a new algorithm that we call Multi_Scheduler. It is based on the Mono_Scheduler algorithm which we proposed in [3].

3.1 Principle of the algorithm

As shown in [3], the Mono_Scheduler algorithm accepts two arguments: Column vectors G and P. It returns the Scheduling Chain (SC) and the vector of Carrying the Service (CS) as a result. This algorithm operates on a CRN with a single channel.

In this work, we consider a MultiChannel, MultiPU and MultiSU. Our algorithm should determine the state of each channel during each slot. These states appear in a matrix, called Matrix of the States, which we denote by MS. The rows of this matrix represent the channels and the columns represent the slots.

$$MS_{ij} = \begin{cases} 1 & \text{if channel i is free during the slot j} \\ 0 & \text{if not} \end{cases} \quad [10]$$

The Multi_Scheduler algorithm is based, both on the SC returned by the Mono_Scheduler algorithm and the MS. In fact, this algorithm runs through SC, element by element, and reserves the necessary channels required to send the current element. During this reservation, the algorithm increments a component of the CS vector[4].
The number of slots needed to transmit an element depends on two parameters: the size of the current element and the number of free channels during the current slot and the following slots.
The algorithm performs two passages: A passage through SC to determine the packets to send, and another through the columns of MS to identify fractions and free channels for each fraction. For each slot we assign two indices: one to indicate the number of the fraction and the other to indicate the amount of free channels during this fraction. Figure 2 shows the relation between the Mono_Scheduler algorithm which we proposed in [3] and the Multi_Scheduler algorithm that we propose in this paper.

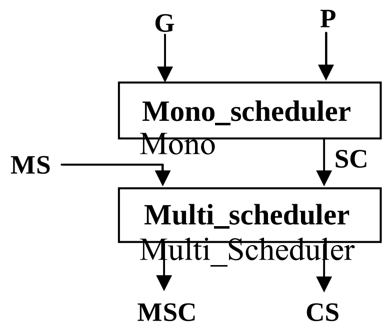

Fig.2: Relation between Mono_Scheduler and Multi_Scheduler.

3.2 System modeling

In this section, we present a set of parameters, relations and functions used in our approach.
R : The total number of groups
G : A vector containing the number of SUs in each group.

$$G = [g_1,.., g_i,.., g_R]^T \quad (1)$$

P : A vector containing the number of packets in each SU.

$$P = [p_1,.., p_i,.., p_R]^T \quad (2)$$

g : The total number of SUs

[4] Vector of Carrying the Service of all groups.

$$g = \sum_{i=1}^{R} g_i \quad (3)$$

p : The total number of packets:

$$p = \sum_{i=1}^{R} g_i p_i \quad (4)$$

CS_i : The number of free fractions passed, to send all packets of the group G_i.

CS : The vector of Carrying the Service to all groups.

$$CS = [CS_1,..,CS_R]^T \quad (5)$$

r_i : The transfert rate for G_i

$$r_i = \frac{p_i g_i}{CS_i} \quad (6)$$

r : The vector of rates

$$r = [r_1,..,r_R]^T \quad (7)$$

EIJ(r): The Equity Index Jain for a vector r [11]:

$$EIJ(r) = \frac{\left[\sum_{i=1}^{R} r_i\right]^2}{R\sum_{i=1}^{R} r_i^2} \quad (8)$$

TNC: Total Number of Channels.
TNF: Total Number of Fractions.
TNPU: Total Number of PUs.
TNPU and TNC must satisfy the following inequality:
$TNPU \leq TNC$, because for each PU, we reserve at least one channel.
VF: A row vector, with TNF columns, containing all fractions.

$$VF = [f_1,.., f_j,.., f_{TNF}] \quad (9)$$

V : A row vector, with TNF columns, containing the number of free channels in each fraction.

$$V = [L_1,.., L_j,.., L_{TNF}] \quad (10)$$

We consider that m is the number of necessary and sufficient fractions passed to send p packets (see equation (4)). So, m is the smallest integer satisfying the relation (11).

$$p \leq \sum_{j=1}^{m} L_j \quad (11)$$

m represents the fraction of Carrying the Service to all groups.
VC: is the CB (see section 4).
CB is a polynomial of the second degree, showing that the set of vectors with integer components and the set of integers are equipotent[5]. So, we can write: $N^k \approx N$

$$\begin{aligned} VC \; : \; & N^k \rightarrow N \quad (12) \\ & Y \rightarrow VC(Y) \quad [12] \end{aligned}$$

For more details on CB, see the section 4.
MS: Matrix of TNC rows and TNF columns, indicating the state of each channel during each fraction.
SC: The Scheduling Chain returned by the Mono_Scheduler algorithm [3].
MSC: Matrix channel assignment to the elements of SC.
Ind: Index of the current fraction.
Red: The remaining number of free channels during the fraction Ind.
InF: Index of the new fraction after channel assignment.
ReF: The remaining number of free channels during the InF fraction.
F: Storage Matrix of InF and ReF values obtained.

In addition to these parameters, we will use in our algorithm, a function, denoted f, which transforms a Conditional Instruction to an Assignment Instruction, for compacting algorithm. The desired function satisfies the condition below:

$$f(x,y) = \begin{cases} 1 & if \quad x=y \\ 0 & if \quad not \end{cases} \quad (13)$$

With x and y are two real numbers.

[5] We say that two sets A and B are equipotent if and only if there exists a bijection between them.

3.3 Algorithm

```
Algorithm  Multi_Scheduler (G, P, MS)
Begin
   SC←Mono_ Scheduler (G, P)[6]
   CS←0
   V←NbFreCha[7]
   Ind←1
   Red←V (Ind)
   i←1
 while i≤Nb_Rows_SC[8]
      k←NbCurrtPa[9]
      (InF, ReF) ←Affect (Ind, Red, v, k)[10]
      f1←f (Red, v(Ind))[11]
      f2←f (InF, Ind)
      f3←f (ReF,v(InF))
      u←Max (CS)[12] +f1
      CS (SC (i, 1)) ←u+ (InF-Ind-f3)(1-f2)[13]
       F(i) ← (InF, ReF)[14]
      (Ind, Red ) ← (InF, ReF)[15]
      i←i+1[16]
   End while
   MSC← [SC   F][17]
   For i=1 To R
      r(i)=P(i)*G(i)/ CS (i)
   End for
End
```

3.3.1 Applying the algorithm on an example.

We consider a CRN with 5 channels, 6 SUs and 4 PUs (Fig.3).

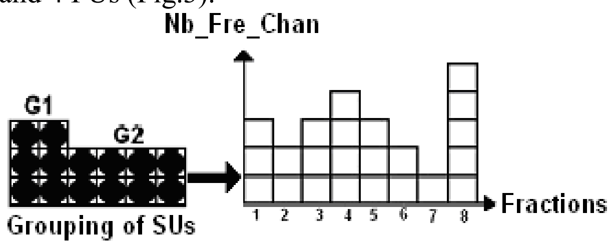

Fig.3: State of a CRN

The following table summarizes the organization of the SUs.

TABLE I. ORGANISATION OF SUS

Group	Nb_of_SUs_perGr	Nb_of_packets_ per SU
G1	2	3
G2	4	2

In the second table, we find the number of free channels for each fraction.

TABLE II. STATES OF CHANNELS.

SLOT	1	2	3	4	5	6	7	8
NB_FREE_CHAN	3	2	3	4	3	2	1	5

So we will have:

$$G=(2,4)^T \ , \quad P=(3,2)^T \quad \text{and then}$$

$$MS=\begin{pmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&1&1&0&0\\1&0&1&1&1&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&0&0&0&0\end{pmatrix}$$

When applying the Mono_Scheduler algorithm on the vectors G and P, we obtain the Scheduling Chain.

$SC=(G_1,1)\ (G_2,1)\ (G_1,1)\ (G_1,1)\ (G_2,1)$ [3].

To show the allocation of free channels to the elements of SC, we will assign a color to each element, as shown in the following table.

TABLE III. ALLOCATING CHANNELS TO ELEMENTS OF SC.

Element	Color	Packets
$(G_1,1)$		
$(G_2,1)$		
$(G_1,1)$		
$(G_1,1)$		
$(G_2,1)$		

[6] Establish SC, applying Mono_Scheduler algorithm [3].

[7] Number of free channels, during each fraction.

[8] i Index of the current element of SC.

[9] Number of packets of the current element.

[10] After channel assignment, find the new fraction and the remaining number of free channels during this fraction.

[11] f is the function that returns a conditional instruction in an assignment instruction.

[12] Find the maximum component of v.

[13] Increment a component of CS.

[14] Store InF and ReF in the matrix F.

[15] Reset Ind and Red.

[16] Advance by one step on SC.

[17] Concatenate SC and F in a single matrix, denoted MSC.

The following figure shows the allocation of free channels to the elements of SC.

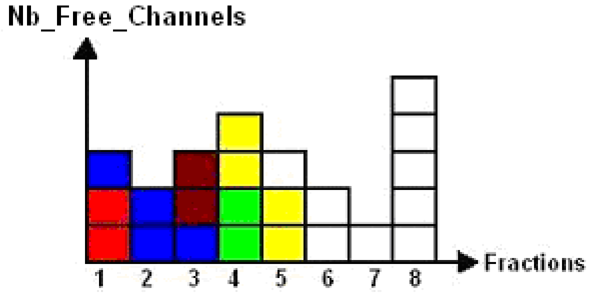

Fig.4: Allocation of free channels to the elements of SC

The first element of SC contains two packets (Table III, second row), they are sent in the first slot (Fig 4, red color).

The second element of SC contains four packets (Table III, third row), the first packet is sent during the first slot, the following two packets in the second slot and the last packet in the third slot (Fig 4, blue color).

The third element of SC contains two packets (Table III, fourth row), that are sent all over the third slot (Fig 4, Brown color).

The fourth element of SC contains two packets (Table III, fifth row), that are sent all over the fourth slot (Fig 4, green color).

The fifth element of SC contains four packets (Table III, sixth row), two packets are sent during the fourth slot and the other two for the fifth slot (Fig 4, yellow color).

After applying the Multi_Scheduler algorithm in the previous example (Fig 3), we find:

$$MSC = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & 3 & 2 \\ 1 & 1 & 4 & 4 \\ 1 & 1 & 4 & 2 \\ 2 & 1 & 5 & 1 \end{pmatrix}, CS = \begin{pmatrix} 4 \\ 5 \end{pmatrix}, r = \begin{pmatrix} 1.5 \\ 1.6 \end{pmatrix}$$

3.3.2 Reading MSC.

We recall that SC is the Scheduling Chain returned by the Mono_Scheduler algorithm [3] and MSC is the Matrix channel assignment to the elements of the Scheduling Chain (SC).

So, MSC matrix contains all information, for scheduling, channel assignment and the calculation of the transfer rate of each group.

- Rows of MSC
  Each row of MSC corresponds to an element of SC.
- Columns of MSC
  The first and the second column contain SC.
  The third column indicates the end slot of sends of the element.
  The fourth column indicates the number of free channels remaining, after the sending of the current element.

<u>Channel assignment to the first element of SC.</u>

The first element of SC is $(G_1,1)$. On the first component of the third column of MSC, we find the value 1 (fraction 1, slot of carrying the service of the current element). On the first component of the 4th column of MSC, we find the value 1 (remaining number of free channels). This means that, after sending the first element, there remains one free channel for the fraction 1 (Fig 4: first slot, red color).

<u>Channel assignment to the 2nd element of SC.</u>

The 2nd element of SC is $(G_2,1)$. On the 2nd component of the third column of MSC, we find the value 3 (fraction 3, slot of carrying the service of the current element). On the 2nd component of the 4th column of MSC, we find the value 2 (remaining number of free channels). This means that, after sending the 2nd element, there remains one free channel for the fraction 3 (Fig 4: first, second and third slot, blue color).

<u>Channel assignment for the 3rd element of SC.</u>

The 3rd element of SC is $(G_1,1)$. On the 3rd component of the 3rd column of MSC, we find the value 4 (fraction 4, slot of carrying the service of the current element). On the 3rd component of the 4th column of MSS, we find the value 4 (remaining number of free channels). This means that, after sending the 3rd element, there remain four free channels for the fraction 4 (Fig 4: third slot, brown color).

<u>Channel assignment for the 4th element of SC.</u>

The 4th element of SC is $(G_1,1)$. On the 4th component of the 3rd column of MSC, we find the value 4 (fraction 4, slot of carrying the service of the current element). On the 4td component of the 4th column of MSC, we find the value 2 (remaining number of free channels). This means that, after sending the 4th element, there remain two free channels for the fraction 4 (Fig 4: fourth slot, green color).

Channel assignment for the 5th element of SC.

The 5th element of SC is $(G_2,1)$. On the 5th component of the 3rd column of MSC, we find the value 5 (fraction 5, slot of carrying the service of the current element). On the 5th component of the 4th column of MSC, we find the value 1 (remaining number of free channels). This means that after sending the 5th element, there remains one free channel for the fraction 5 (Fig 4: fourth and fifth slot, yellow color).

4 Applications

4.1 Cantor's Bijection (CB)

Let N be the set of integer numbers and N^k the set of vectors with k integer components (k is an integer greater than or equal to 1). The Cantor's Bijection, denoted VC, maps each vector v of N^k to one and only one integer denoted VC(v):

$$VC : N^k \rightarrow N$$
$$v \rightarrow VC(v)$$

Since this is a bijection, then we will have the following equivalence:

$\forall\ v \in N^k, \forall\ v' \in N^{k'}\ \ VC(v) = VC(v')$ if and only if $v = v'$ [12].

4.2 Relation between Cantor's bijection and the Multi_Scheduler algorithm

The Multi_Scheduler algorithm admits three arguments:

1) Vector G: Containing the number of SUs in each group.
2) Vector P: Containing the number of packets in each SU.
3) Matrix MS: Binary matrix indicating the status of each channel during each slot.

We can do the concatenation of P and G, to obtain a new vector denoted GP. If we take for example $G = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$ *and* $G = \begin{pmatrix} 4 \\ 5 \end{pmatrix}$, the concatenation of G and P gives: $GP = (3,2,4,5)^T$.

The equity between the rates of groups, applying the algorithm is a function of GP and MS. If MS is fixed, the equity depend on GP only, since GP is a vector with integer components, then it can be replaced by VC(GP) (by Cantor [12]).

We applied the algorithm in two situations:

Situation 1: P and G each contain two components.

In this case we consider eight samples and we represent EIJ depending on the value of Cantor VC [12] corresponding to the vector:
GP= [G, P] (Fig 5).

Situation 2**:** same principle as Situation 1. The only difference is the size of G and P (number of components equal to 3) (Fig 6).

In both following figures, we find the representation of EIJ in terms of VC, reported previously.

On the horizontal axis we put VC and on the y-axis, we put EIJ mentioned in [11]. This index is, by definition, always between 0 and 1.

4.3 Representation of the results

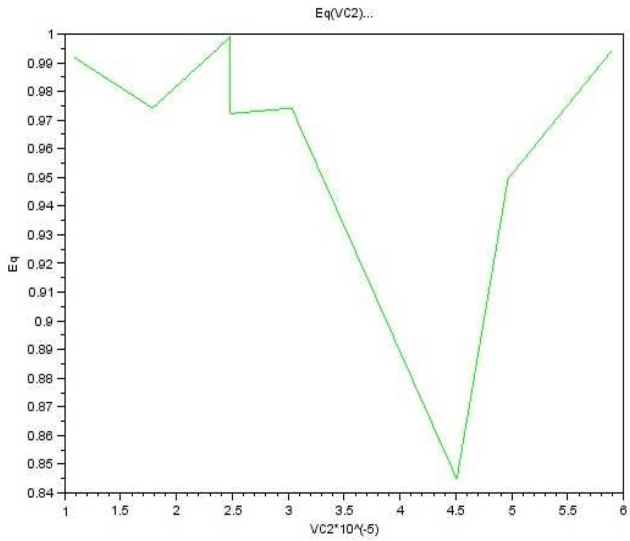

Fig.5: Variation of equity, based on the value of Cantor for eight samples of both groups of SUs.

We note that, for different values of VC, we always find equity greater than or equal to 0.84.

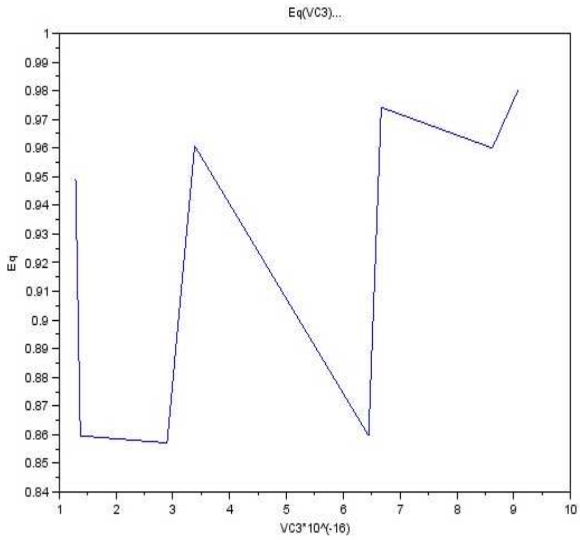

Fig.6: Variation of equity, based on the value of Cantor for eight samples of three groups of SUs.

In this case, the equity remains greater than or equal to 0.85, for different values of VC.

4.4 Interpretations

We chose the VC as a variable on the x-axis because each value of VC corresponds to one and only one vector GP with integer components and vice versa. It is a bijection [12].
We note, that the better performance in terms of fairness under the equity index jain, are characterized by values of EIJ too close to 1.
In both figures, we have represented the IJ in terms of VC, corresponding to the vector GP.
We always find a value of EIJ greater than 0.84, which shows that EIJ is close to 1. In other words, it can be concluded that the application of our algorithm on different samples of GP allows us to achieve better results in terms of equity transfer rates of SUs.

5 Conclusion

In this work, we proposed the Multi_Scheduler algorithm basing ourselves on the Mono_Scheduler algorithm [3]. The Multi_Scheduler algorithm can ensure equity between SUs, by identifying the SU, the number of packets to send, and the slots and channels to use.
After experiments performed on different samples of GP, we found values of EIJ always close to 1. This proves the effectiveness of the implemented algorithm.

References

[1] Mitola J., *Cognitive radio: an integrated agent architecture for software defined radio*, Ph.D. Dissertation. Royal Institute of Technology (KTH). Teleinformatics. Electrum 204. SE-164 40 Kista. Sweden, May 2000, pp 8-13
[2] James O'Daniell Neel, *Analysis and Design of Cognitive Radio Networks and Distributed Radio Resource Management Algorithms*, Dissertation submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the of Doctor of Philosophy in Electrical Engineering, September 2006, pp 23-25.
[3] Abdellah Idrissi and Said Lakhal, An algorithm for fairness between secondary users in cognitive radio network, *JATIT*, Vol. 58 No.1 December 2013, pp 230-236.
[4] Ammar S. Alshamrani, *Resource Management in Cognitive Radio Networks*, thesis, Waterloo, Ontario, Canada, 2010. pp. 25-28.
[5] Berk Canberk, Ian F. Akyildiz and Sema Ok tug, QoS-Aware User Cohabitation Coordinator in Cognitive Radio Networks, *IEEE, Globecom* 2012.
[6] Lutfa Akter and Balasubramaniam Natarajan. Modeling Fairness in Resource Allocation for Secondary Users in a Competitive Cognitive Radio Network. *IEEE,* 2010.
[7] Saleem Aslam, Waleed Ejaz, Najam Ul Hassan and Naveed Ejaz, Dynamic Intra–network Spectrum Sharing for Cognitive Radio Net. Works. IPCSIT, IACSIT Press, vol.3, 2011.
[8] Yongtao Ma, Liuji Zhou and Kaihua Liu, A Subcarrier-Pair Based Resource Allocation Scheme Using Proportional Fairness for Cooperative OFDM-Based Cognitive Radio Networks, *Sensors,* 2013.
[9] Bassem Zayen, Majed Haddad, Aawatif Hayar and Geir E. Øien, Binary power allocation for cognitive radio networks with cen tralized and distributed user selection strate gies, *ELSEVIER*, 2008.
[10] R. Urgaonkar, M. J. Neely, Opportunistic Scheduling with Reliability Guarantees in Cognitive Radio Net Works , *proc. ieee infocom*, *phoenix*, april 2008.
[11] Raj Jain, A Quantitative Measure of Fairness and Discrimination for Resource Allocation in shared computer system,

ACM Transaction on Computer Systems, September 28, 1984.

[12] *http://jm.davalan.org/divers/bij/index.html* visited in 29/10/2013

[13] Shuang Li, Zizhan Zheng, Eylem Ekici and Ness Shroff, Maximizing System Throughput by Cooperative Sensing in Cognitive Radio Networks, *arXiv.org > cs > arXiv:1211.3666*, November 2012.

[14] Dusit Niyato and Ekram Hossain, Optimal Price Competition for Spectrum Sharing in Cognitive Radio: A Dynamic Game-Theoretic Approach, *IEEE GLOBECOM*, 1930-529X/07/ 25.00, 2007.

[15] Yangyang Li, *Cognitive Interference Mangement in 4G Autonomous Femtocells*, Thesis. Department of Electrical and Computer Engineering University of Toronto, published in: http://hdl.handle.net/1807/24815, 2010.

[16] Abdoulaye Zana Bagayoko, *Spectrum Sharing under interfering Constraints*, Ph.D. Thesis presented at University of Cergy-Pontoise, Octobre 2010.

[17] Vamsi Krishna Tumuluru, Ping Wang, Member and Dusit Niyato. A Novel Spectrum Scheduling Scheme for Multi-channel Cognitive Radio Network and Performance Analysis, *IEEE transactions on vehicular technology*, may 2011.