A Game-Theoretic Approach for Cognitive Radio Networks using Machine Learning Techniques

S. Mangairkarasi, Rooppesh Sarankapani and Deivasegamani Arivudainambi

January 4, 2020

A Game-Theoretic Approach for Cognitive Radio Networks using Machine Learning Techniques

S. Mangairkarasi [1], Rooppesh Sarankapani [2], D.Arivudainambi [3]
[1]Department of Mathematics, Anna University, Chennai, India
[2]Department of Computer Science & Engineering, Anna University, Chennai, India
[3]Department of Mathematics, Anna University, Chennai, India

Abstract. Cognitive Radio has been viewed as a promising technology to enhance spectrum utilization significantly. In this work, we propose a model for Dynamic Spectrum Allocationin Cognitive Radio Networks using Game Theory. Furthermore, in order to accommodate for all cases, we have put to good use of Preemptive Resume Priority M|M|1 Queuing Model. To supplement it we introduce a priority-based scheduling algorithm called Incremental Weights-Decremental Ratios (IW-DR). As a means to ameliorate the efficiency,we have made use of Regression Models.

Keywords: Cognitive Networks, Spectrum Allocation, Queuing Theory, Game Theory, Regression.

1 Introduction

With the advent of the digital age, there has been a critical deficit of unlicensed spectrum, as a consequence of rising demands for wireless spectrum. There are an exponentially increasing number of applications and devices that singularly depend on the availability of the unlicensed bands. Such applications and devices make the unlicensed bands congested, contrarily; preliminary studies have shown that a significant portion of the licensed bands is being underutilized. To ensure the prospective growth of wireless services, it is vital to increase the efficient usage of these channels.

Cognitive Radio (CR) [1] & [2] Networks have been proposed as the novel solution to alleviate the deficiency problem of the limited radio spectrum. The CR Network is composed of intelligent spectrum-agile devices that are competent of adjusting their configurations based on the spectrum environment [3] & [4]. A CR Network typically has two types of users: Primary Users (PUs) who are obligatory licensed users of the spectrum and Secondary Users (SUs) who try to opportunistically access the unused licensed spectrum, this feature is called Dynamic Spectrum Access [5]. The system has to attain a way to ensure that these networks are able to peacefully and harmoniously coexist without any loss in Quality of Service. In [6], [7],

[8] & [9], game theory, auctions, leasing etc. have been proposed to aid dynamic spectrum allocation.

However, game theory has been used as a robust tool developed to model the interactions of players with contradicting interests. However, while implementing game theory, there is a premise that each player in the game is rational. Being rational players in the game, SUs intent to individually maximize their own payoffs. In our case, the payoff being the allotment of a channel.

This work proposes the use of a non-cooperative dynamic game wherein the SUs (players) compete for the available channels, relinquished by the PUs. Their strategy is to switch or stay between the available networks in such a fashion that they dodge collisions with other SUs. The game reaches an equilibrium point once all the SUs have acquired an accessible channel.

In [10], [11] & [12] a similar, game-theoretic environment was set up for dynamic spectrum access. In [11] & [12], the Nash Equilibrium for the game is formulated. In [10] regression techniques are implemented to simulate the game. However, in [10] there was a setback, the system fails when the traffic surpasses the number of available channels.

In this work, we have amended this setback. In order to relieve the congestion among the arriving SUs [10] we make use of Preemptive Resume Priority (PRP) M|M|1 queuing network model [13] & [14].

Another premise to consider while implementing Game Theory in Spectrum Allocation is that all SUs might not be guaranteed the same levels of performance [15]. Thus, in this work we introduce a customized scheduling algorithm named Incremental Weights-Decremental Ratios (IW-DR). Where, in order to bolster the delay-sensitive secondary user applications and achieve the quality of service between different classes of users, we prioritize them based on their application type and order of sensitivity.

To realize learning in the game, we have analysed various regression algorithms as the datasets would be in a continuous fashion. In this paper, we have employed Linear Regression, Polynomial Regression, Support Vector Regression, Decision Tree Regression and Random Forest Regression in order to predict the optimal probability for a given (N_N, N_C) tuple, where N_N is the number of available networks and N_C is the number of active channels competing for the networks.

2 System model

Assume a game environment with dynamically changing components, let X_T be the number of available channels at a given point of time and Y_T be the number of networks or SUs competing for X_T channels. Assume the time period of arrival into the system $(1/\lambda_T)$ is greater than the time taken to accommodate the channels (T_{EQ}). Based on X_T and Y_T the system can be divided into two sub-cases.

Table 1. Payoff Matrix/Game Strategy

A/B	Switch	Stay
Switch	(C,C)	(C,0)
Stay	(0,C)	(0,0)

Case 1: $X_T \geq Y_T$

In this case, at equilibrium (T_{EQ}) all the Y_T networks would be accommodated in either of the X_T channels. All the Y_T networks are given an equal opportunity to all the X_T channels. But, if a conflict of interest occurs (i.e) two or more networks competing for the same channel, then their strategy is to switch or stay between the channels to avoid collisions depending on the most favourable option as depicted in Table.1, where C is the cost of switching.

We illustrate this system with Fig.1 where the squares represent the channels. In our case, we have 8 networks competing for 10 channels. (SU_3, SU_1, SU_7), (SU_2, SU_4) & (SU_5, SU_6) in Step 1 (SU_2, SU_1) & (SU_4, SU_6) in Step 2 and (SU_1, SU_4) tuples in Step 3 have a collision. However, in each case the problem is dealt in a different manner, depending upon the optimal probability.

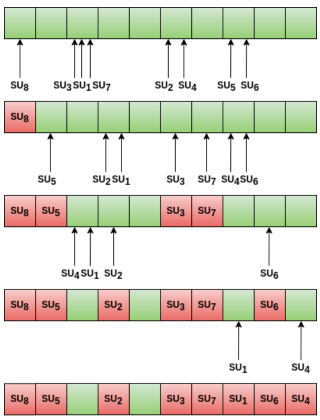

Fig. 1. Ilustration of the non-cooperative dynamic game with 8 networks and 10 channels

From the above example, we can comprehensively establish the fact that the optimal probability of switching can neither be 0 nor be 1. As neither would lead to a state of equilibrium. If p is the probability of switching, the probability of staying would be $1-p$. Hence, the probability tuple would be $(p,1-p)$.

Case 2: $X_T < Y_T$

In a practical scenario, we would be dealing with different application types in CR networks, one being the real-time applications and the other being non-real time applications. The real-time application types are more sensitive to transmission delay than the non-real-time applications. According to their sensitivity, they are pushed into either of the Y queues with descending order of priority.

Additionally, to enhance the user experience and Quality of Service, an interrupted SU must be given higher priority than the newly arrived SU. This typical case of traffic congestion provides a good application for the use of a Preemptive Resume Priority (PRP) M|M|1 queuing network model. As depicted in Fig.2, the interrupted SUs are given a higher priority.

To provide an opportunity for Lower Priority Queue members to access the networks and to reduce the waiting time for the Lower Priority Queue members we devised a scheduling algorithm called Incremental Weights - Decremental Ratios (IW - DR).

***IW - DR Scheduling Algorithm**:*

```
N -> Number of Active Queues
if(Q1)
   Allow Users (Q1, N)
   Continue
if(Q2)
   Allow Users (Q2, N-1)
   Continue
 . . . . .
if(Qn)
   Allow Users(Qn,1)
End
```

Assume, we have a total of N active queues $(Q_1, Q_2, ...Q_n)$ at a given point of time such that $Q_p >_{\text{Priority}} Q_{p+1}$.

In this paper, in order to dynamically schedule the active queues, the ratio of the number of users permitted from $Q_1 : Q_2 : ....Q_n$ would be $N : N-1 : N-1 : ....N$. Hence, N users of Q_1 are given the highest priority followed by $N-1$ users from Q_2 and so on. In this manner, we are able to provide equality.

For example, if we there were five active queues namely Q_1, Q_2, Q_3, Q_4& Q_5 in order of priority. A maximum of five users would be permitted from Q_1, followed by four users from Q_2 and so on. In this manner we are able to reduce the average waiting time of the Lower Priority Queue members.

The first X_T networks of highest priority are then selected from the IW-DR Scheduling Algorithm, to compete for X_T channels, which boils down as a sub-case of Case 1 ($X_T = Y_T$).

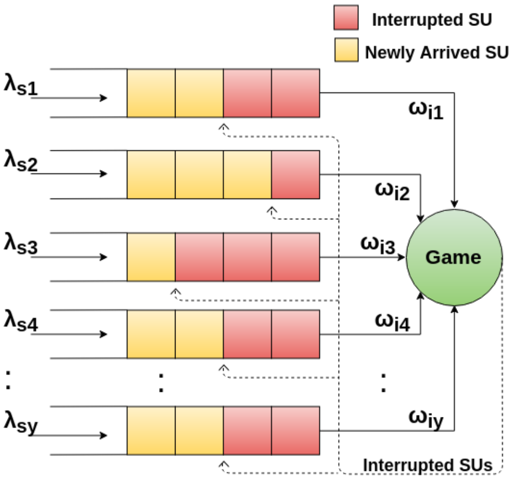

Fig. 2. Illustration of PRP M|M|1 Queuing Network Model with y levels of priority

As depicted in Fig.2, the interrupted SUs are given a higher priority as compared to the newly arrived SUs. The selected channels are then entitled to compete among themselves for a possible network, based on the Game Algorithm.

Game Algorithm:

M -> Number of Networks
N -> Number of Channels
for*(P = 0.01 -> 0.99)*
StartTime = CurrentTime
(P, 1 - P) -> (SwitchProbability, StayProbability)
while*(!EquilibriumState)*
SimulateGame(M, N, P)
EndTime = CurrentTime
Time[P] = EndTime – StartTime
for*(P = 0.01 -> 0.99)*
if*(Time[P] = min(Time[ ]))*
OptimizedProbability = P
end

Game Algorithm has been used to experimentally simulate a game similar to our scenario, we then calculate the time taken to reach equilibrium for each value of P, ranging from 0.01 to 0.99.

This process is repeated 100 times for each (M, N, P) tuple and then the mean equilibrium time is calculated. The optimized probability for a corresponding (M, N) tuple is the one for which the equilibrium time is the least. Thus an (M, N, P_0) tuple is the output for a given input tuple (M, N).

3 Proposed Algorithms

In order to be really cognitive, a Cognitive Radio Network should be equipped with abilities of learning and reasoning. In our paper we have used the following regression techniques:

3.1 Simple Linear Regression

It is called a Simple Linear Regression if there is only one independent variable and is called a Multiple Linear Regression if it has more than one independent variable. Mathematically it is denoted as:

$$f(x) = w_0 + w_1 x_1 + w_2 x_2 + ...w_d x_d = w_0 + \sum_{j-1}^{d} w_i k_i \tag{1}$$

It is called a linear regression since it it a linear function of parameters $w = (w_0, w_1, w_2, \dots w_d)$ and input variables $x = (x_1, x_2, \dots x_d)$.. The parameter w_0 allows for any fixed offset in the data. We extend the class of models by considering linear combinations of fixed nonlinear functions of the input variables, of the form:

$$f(x) = w_0 + \sum_{j-1}^{m-1} w_j \phi_j(x) \tag{2}$$

where $\phi_j(x)$ is known as basic functions. In the case of Linear Regression, $\phi_j(x) = 1$. By denoting the maximum value of the index j by $M-1$, the total number of parameters in this model will be M .

3.2 Polynomial Regression

Polynomial Regression is a more versatile algorithm as compared to Linear Regression, however, it is quite similar to too, the primary difference being that the basis function would be of the form:

$$\phi_j(x) = x^j \qquad (3)$$

Where the degree of the polynomial is $M-1$. Depending on the value of M, we can have a Constant Polynomial $(M=0)$, First Order Polynomial $(M=1)$, Second Order Polynomial $(M=2)$ and so on. We choose the degree that best fits our training dataset.

3.3 Support Vector Regression

Support Vector Regression uses the same principles as the SVM for classification that is to find a hyperplane that separates the data in a multidimensional space with as maximal separation between the data points and hyperplane as possible. In Support Vector Regression, our goal is to find a $f(x)$ such that it has a deviation of at most ε, that is the errors are fine as long as they are within the limits of ε. We define our linear function as:

$$f(x) = wx + b \qquad (4)$$

The main concern is to reduce the error. Which can be modelled as an optimization problem:

$$\min imixe \frac{1}{2}\|w^2\| \qquad (5)$$

Such that,

$$y_i - wx_i - b \leq \varepsilon \qquad (6)$$

$$wx_i + b - y_i \leq \varepsilon \qquad (7)$$

3.4 Decision Tree Regression

Decision Tree Regression uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). The most popular algorithm to build the decision tree is the CART (Classification and Regression Trees) which uses the Gini Index as the metric:

$$Gini\ Index = 1 - \sum_{i=1}^{c} (p_i)^2 \tag{8}$$

Where C, is the various classes and Pi is the probability of each class.

3.5 Random Forest Regression

Random Decision Forest is an ensemble learning method for regression, where a multitude of decision trees are constructed at training time and the mean prediction of the individual trees is outputted. It acts as a solution for the overfitting problem sometimes faced in Decision Tree Regression. The importance of each feature on a decision tree is calculated as:

$$norm\ fi_i = \frac{fi_i}{\sum_{j\ \varepsilon all\ features} fi_j} \tag{9}$$

Where $normfi$is the normalized importance of feature i and $fi$is the importance of feature i. Then feature importance values from each tree are normalized:

$$RFfi_i = \frac{\sum_j norm\ fi_{ij}}{\sum_{j\ \varepsilon\ all\ features\ k\ \varepsilon\ all\ trees} norm\ fi_{jk}} \tag{10}$$

Where $RFfij$ is the importance of feature i calculated from all trees in the Random Forest model.

4 Performance Evaluation

For all simulation purposes, we have used Python in Spyder (Scientific Python Development Environment), which is an open source integrated development environment (IDE) that is included with the Anaconda framework.

In order to stand by its definition, cognitive radio networks need to be armed with the capabilities of learning and reasoning. Moreover, it is not feasible for the channels to sense and search for channels, hence the need for self-learning arises. In order to couple learning in games, we need an effective dataset. In our work, we have simulated such a game environment utilizing which, we obtained the optimal probability for a given number of networks (N_N) and available channels (N_C).

We simulate the Game Algorithm for various possible combinations of M and N ranging from 10-50 where $M \geq N$, which gives us a total of 820 datasets. Out of which 656 datasets are used for training and 164 are used for testing. The dataset is stored in a database with 3 columns, Number of Channels, Number of Networks and Optimal Probability of Switching (M, N, P_0).

In this work, we have proposed the use of five different regression algorithms: Simple Linear Regression, Polynomial Regression, Support Vector Regression, Decision Tree Regression, and Random Forest Regression.

Fig.3 illustrates the experimental values of the probability as obtained from the Game Algorithm. As read from the graph, the probability of switching drops as the number of competitors raises in the system.

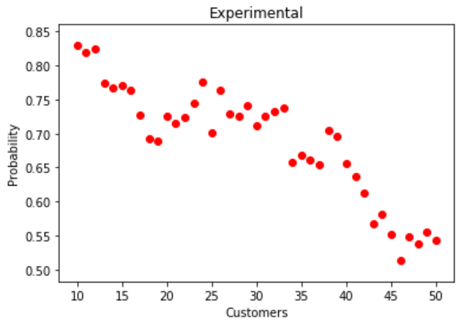

Fig. 3. Experiment – Optimal Probability vs Customers

Fig.4 depicts the results obtained from Linear Regression on Optimal Probability. The nature of the result is right however it is not the most optimal solution.

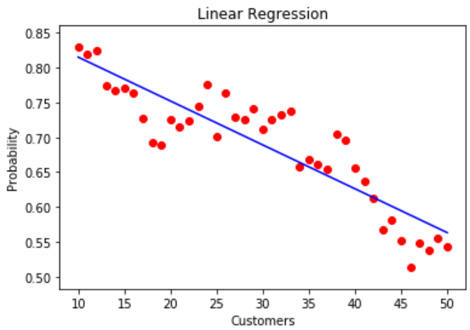

Fig. 4. Linear Regression – Optimal Probability vs Customers

Alternatively, we implement Polynomial Regression - an algorithm quite similar to Linear Regression, but a bit more versatile. In this case, we have used a polynomial regression of degree 3 as it most advantageously fits our dataset. Fig.5 depicts the results obtained on performing Polynomial Regression.

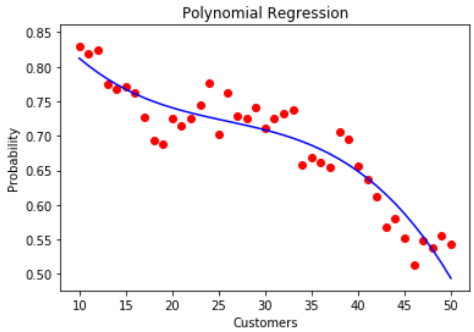

Fig. 5. Polynomial Regression – Optimal Probability vs Customers

Fig.6 depicts the Optimal Probability Vs Customers graph as obtained from Support Vector Regression using the ‘rbf’ kernel.

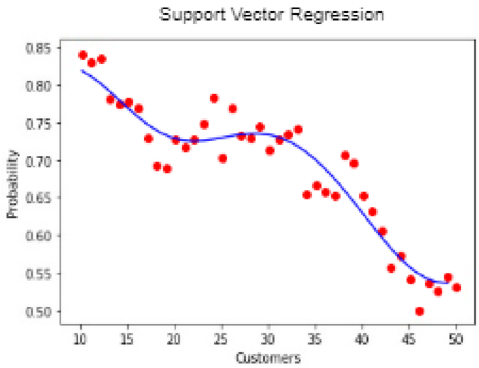

Fig. 6. Support Vector Regression – Optimal Probability vs Customers

The results obtained from Decision Tree Regression are presented in Fig.7. For unsurpassed results we have set the parameter ‘max_depth’ to 3 and ‘random_state’ to 0.

Fig. 7. Decision Tree Regression – Optimal Probability vs Customers

In Fig.8 we represent the results obtained on performing Random Forest Regression. In order to fit the data accurately, we have set parameters 'n_estimators' to 100, and 'min_samples_leaf' to 2.

In Table 2 we have portrayed the numerical values of the optimal probability of switching for 10 networks retrieved from the experiment as well as using the regression algorithms, with an interval of 5 customers. Furthermore, in order to analyse the efficiency of the five regression algorithms: Simple Linear Regression (SLR), Polynomial Regression (PR), Support Vector Regression (SVR), Decision Tree Regression (DTR), and Random Forest Regression (RFR), we calculate their Mean Square Error using our dataset. This has been depicted in Table.3.

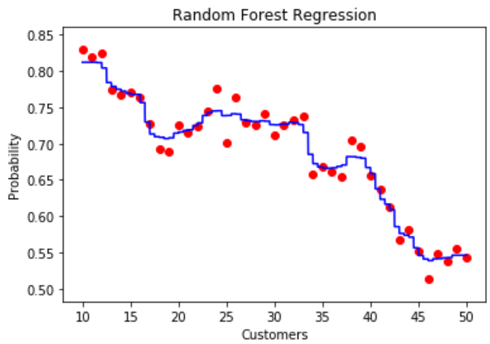

Fig. 8. Random Forest Regression – Optimal Probability vs Customers

Table 2. Numerical Comparison between the probabilities as obtained from different strategies

	Exp	SLR	PR	SVR	DTR	RFR
10	0.8295	0.818	0.818	0.81	0.819	0.809
15	0.7708	0.788	0.759	0.743	0.77	0.7708
20	0.7247	0.758	0.742	0.721	0.727	0.715
25	0.7019	0.728	0.721	0.7291	0.727	0.7404
30	0.7114	0.698	0.71	0.7291	0.727	0.7254
35	0.669	0.668	0.689	0.7054	0.678	0.669
40	0.6557	0.638	0.662	0.637	0.678	0.6557
45	0.5519	0.608	0.578	0.548	0.535	0.5519
50	0.5433	0.578	0.492	0.551	0.535	0.5561

Table 3. Mean square errors of the proposed five regression algorithms

SLR	PR	SVR	DTR	RFR
0.00078	0.00052	0.00044	0.00024	0.00022

Table 4. Root mean square of the proposed five regression algorithms

SLR	PR	SVR	DTR	RFR
0.0280	0.0228	0.0210	0.0157	0.0150

Table 5. Comparison of mean quare error of various prediction algorithms

Existing		Proposed	
Linear	SVR	Linear	SVR
0.03345	0.181004	0.00078	0.00044

In [10] Linear Regression, Support Vector Regression and Elastic Net Regression were used as the predictive algorithms. However, in this work we have implemented Regression, Support Vector Regression, Decision Tree Regression, and Random Forest Regression. Moreover, as depicted in Table.4, this work has enhanced the results of Linear Regression by 97.7% and that of Simple Vector Regression by 99.8% when compared to [10].

In addition, [10] didn't support the case where the traffic of SUs surpasses the number of available channels. However, in this work with the use PRP M|M|1 Queuing Networks and IW-DR Scheduling Algorithm, this issue has been resolved.

5 Conclusion

In this paper, a dynamic non-cooperative game was implemented to help improve Dynamic Spectrum Allocation. In order to steer the traffic of Secondary Users, this work proposes the use of a PRP M|M|1 Queuing Network. Additionally, IW-DR Scheduling Algorithm was introduced to provide equality among the various classes of users in the queues. Further, in order to induce self-learning in our non-cooperative game we propose the use of five different regression algorithms: Simple Linear Regression, Polynomial Regression, Support Vector Regression, Decision Tree Regression, and Random Forest Regression, and analyse and compare their results along with the existing work.

References

1. J. Mitola III : Cognitive Radio: An integrates Agent Architecture for Software Defined Radio, Doctoral Dissertation, KTH, Stockholm, Sweden, (2000).
2. J.Mitola III, G. Q. Maguire, Jr.: Cognitive radio: making software radios more personal, IEEE Personal Communication, vol. 6, no. 4, pp. 13 –18, (1999).
3. T.Costlow : Cognitive radios will adapt to users, IEEE Intelligent Systems, vol. 18, no. 3, (2003).
4. S.K.Jayaweera, C.G.Christodoulou: Radiobots: Architecture, algorithms and real time reconfigurable antenna designs for autonomous, self-learning future cognitive radios, University of New Mexico, Technical Report EECE-TR-11-0001, (2011).
5. C.Song, Q.Zhang : Intelligent Dynamic Spectrum Access Assisted by Channel Usage Prediction, IEEE Communications Society, (2010).
6. V. Krishnamurthy: Decentralized spectrum access amongst cognitive radios - An interacting multivariate global game-theoretic approach, IEEE Transactions on Signal Processing, vol. 57, no. 10, pp. 3999 –4013, (2009).
7. Z.Han, R. Zheng, H.Poor: Repeated auctions with Bayesian nonparametric learning for spectrum access in cognitive radio networks, IEEE Transactions on Wireless Communication, vol. 10, no. 3, pp. 890 –900, (2011).
8. S.K.Jayaweera, T.Li: Dynamic spectrum leasing in cognitive radio networks via primary-secondary user power control games, IEEE Transactions on Wireless Communication, vol. 8, no. 6, pp. 3300–3310, (2009).
9. S.K.Jayaweera, G.Vazquez-Vilar, C.Mosquera: Dynamic spectrum leasing: A new paradigm for spectrum sharing in cognitive radio networks, IEEE Transactions on Vehicular Technology, vol. 59, no. 5, pp. 2328– 2339, (2010).
10. A.Thapaliya, S.Sengupta: Understanding the feasibility of Machine Learning Algorithms in a Game Theoretic Environment for Dynamic Spectrum Access, SumerSim-SPECTS, Bellevue, Washington, USA, (2017).

11. S.Sengupta, R.Chandramouli, S.Brahma, M.Chatterjee : A game theoretic framework for distributed self-coexistence among IEEE 802.22 networks, Global Telecommunications (GLOBECOM), IEEE Conference and Exhibition, pp. 1-6, (2008).
12. D.Huang, C.Miao, Y.Miao, Z.I.Shen : A Game Theory Approach for Self-Coexistence Analysis among IEEE 802.22 Networks, 7th International Conference onInformation, Communications and Signal Processing, pp 1-5, (2009)
13. Ahmed Y. Zakariya, Sherif I. Rabia : Analysis of an Interruption-Based Priority for Multi-Class Secondary Users in Cognitive Radio Networks, IEEE ICC 2016 - Cognitive Radio and Networks Symposium, pp. 1-6, (2016)
14. Y. Zhang, T. Jiang, L. Zhang, D.Qu,W.Peng: Analysis on the Transmission Delay of Priority-Based Secondary Users in Cognitive Radio Networks, International Conference on Wireless Communications and Signal processing, pp. 1-6, (2013)
15. James Neel, Jeffrey H. Reed, Robert P. Gilles : The role of Game theory in Analysis of Software radio Networks, Proceeding of the SDR 02 Technical Conference and Product Exposition, pp. 1-7, (2002).