ISSN XXXX XXXX © 2018 IJESC

Research Article Volume 8 Issue No.4

Scheduling Algorithm for High Density MIMO Channels

C.Elayaraja[1], P.Manju[2], I.Parsana Begam[3]
Associate Professor[1], Student[2, 3]
Department of ECE
Dhaanish Ahmed College of Engineering, Padappai, Chennai, India

Abstract:
MIMO is a technology in which multiple antennas are used in both transmitter and receiver. MIMO is used for sending and receiving more than one data signal simultaneously over the same radio channel by exploiting multipath propagation. MIMO antennas achieve high data rate, for achieving this mathematical algorithms are used to recover original transmitted information with reduced BER.MIMO uses high power, has hardware complexity and the receiver will get confused while selecting the data. By using scheduling method the conflict in the receiver side will reduce and transmitted power also be reduced .By using learning automata and vertex multi coloring techniques scheduling is achieved. The graphical output of these two techniques is compared.

Keywords: Learning automata, Vertex multicoloring.

I. INTRODUCTION

Wireless link scheduling is one of the major challenging issues in multihop wireless networks when they need to be designed in distributed fashion. Scheduling in wireless network, users are in uncertain environments generated by fully stochastic network dynamics. The general randomized scheduling method by using learning automata based framework that allows through put optimal scheduling algorithms to be developed in a distributed fashion. A distributed scheduling algorithm that operates on more realistic conflict graph was proposed based on the physical interference model. In scheduling wireless links for simultaneous activation in such a way that all transmission is successfully decoded at the receivers and moreover network capacity is maximized is a computationally hardproblem.Usually it is tackled by heuristics whose output is a sequence of timeslots in which every link appears in exactly one timeslot. These problems in wireless scheduling method are reduced by using improved multi coloring and learning automata methods.

II. a. LEARNING AUTOMATA

A learning automation is an adaptive decision making unit situated in a random environment that learns the optimal action through repeated interactions with its environment. The actions are chosen according to a specific probability distribution which is updated based on the environment response the automation obtains by performing a particular action. With respect to the field of reinforcement learning, learning automata are characterized as policy iterators. In contrast to other reinforcement learns, policy iterators directly manipulate the policy π. Another example for policy iterators are evolutionary algorithms.

Stochastic automation consists of:

- ✓ A set of possible inputs,
- ✓ A set $\varphi = \{\varphi_1,\ldots\varphi s\}$ of possible internal states,
- ✓ A set $\alpha = \{\alpha 1, \ldots, \alpha_s\}$ of possible outputs, or actions, with r≤s,
- ✓ An initial state probability vector $p(0)=\ll p_1(0),\ldots,p_s(0)\gg$,
- ✓ A computable function A which after each tie step t generates p(t+1) from p(t), the current input, and the current state, and
- ✓ A function G:$\varphi \rightarrow \alpha$This generates the output at each time step.

b. VERTEX MULTICOLORING:

A multicoloring is an assignment where each vertex is assigned not just a single number (a "color") but a set of numbers. The numbers of colors assigned to the vertex is specified by the length parameter of that vertex in the input. As usual, adjacent vertices cannot receive the same color; thus here, the sets of colors they receive must be disjoint. In this approach every vertex gets exactly one color in the coloring of graph vertices. Here assigning of multiple colors to each vertex, so that in the resulting schedule, every link may appear more than once.

III. EXISTING SYSTEM:

Scheduling is very important in MIMO. Learning automata and vertex multi coloring is the best method compared to other methods.

LA: In this scheduling algorithm comparison between new schedule S^ (t) and old one S (t) to reach new schedule S(t+1).In the second part of compare algorithm, the constructed spanning tree was used, which is the output of the previous procedure for comparing total weights of two consecutive schedules. The link set S1, S_2, S_3.. S_t Output by a number of link activations given by one activation per link. If this schedule were to be repeated q times in a row for some q>1,the total number of link activations would grow by a factor of q and so would the number of timeslots used. That the same growth law should apply both to how many links are activated and to how many timeslots elapse indicates that the most basic scheduling unit is S_1,S_2,S_3............S_t itself ,not any number of repetitions thereof.

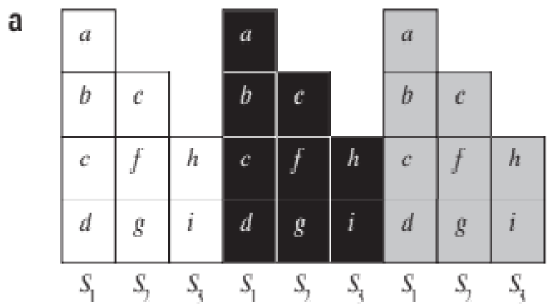


Figure.1. Based on continually repeating the single color schedule.

IV.PROPOSED SYSTEM:

4.1 ALGORITHM FOR LEARNING AUTOMATA:

STEP1: Enter the number of nodes
STEP2: Select the reference node.
STEP4: Find the shortest path based on time.
STEP5: Consider adjacent node as reference node.
STEP6: Repeat the STEP 3,4,5 until it reaches the destination.

4.2 ALGORITHM FOR VERTEX MULTICOLORING METHOD:

STEP1: Declare the number of nodes in consider network.
STEP2: Consider any one of the node as reference node.

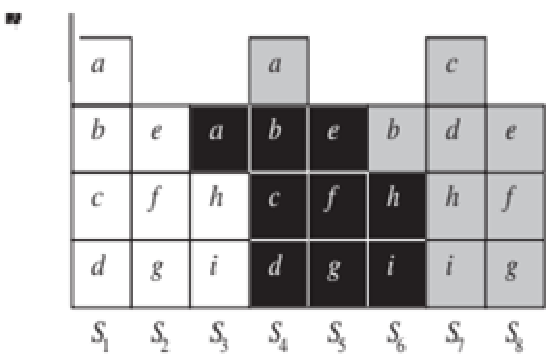


Figure.2. Based on continually repeating multi coloring scheduling

STEP3:Declare the adjacency matrix tothat particular reference node.
STEP4:Consider the coloring for required node based on the distance.

- ✓ Green-Shortest distance from the reference node
- ✓ Blue-Except shortest distance node all other nodes are appears in blue

STEP5:In the colored graph the reference nodes will transmits which is in Green color node.
STEP6:Then the new Green node act as reference node.
STEP7:Now new reference node will search for Green node if applicable it will transmit.
STEP8:The above STEP6,7 repeat until it reaches the destination.

V. PROGRAM FOR LA AND VM:

Java and Mat lab software is used for scheduling. Java-LA, Mat lab-VM.

5.1LEARNING AUTOMATA PROGRAM

```
importjava.util.Scanner;
importjava.io.PrintWriter;
importjava.io.IOException;
public class TestClass2 {
public static void main(String args[]) throws IOException {
Scanner s = new Scanner(System.in);
PrintWriterout = new PrintWriter("out1.txt");
p = new int[10];
pp = new int[10];
bt = new int[10];
w = new int[10];
t = new int[10];
System.out.print("Enter the number of process : ");
out.print("Enter the number of process : ");
System.out.print("\n\t Enter burst time : time priorities \n");
out.println("\n\t Enter burst time : time priorities \n");
for(i=0;i<n;i++) {
System.out.print("\nProcess["+(i+1)+"]:");
out.println("Process["+(i+1)+"]:");
pp[i] = s.nextInt();
p[i]=i+1;}
for(i=0;i<n-1;i++){
}}
for(i=0;i<n;i++)
{ System.out.print("\n   "+p[i]+"\t\t   "+bt[i]+"\t\t     "+w[i]+"\t\t
"+t[i]+"\t\t    "+pp[i]+"\n");
out.println("\n"+p[i]+"\t\t "+bt[i]+"\t\t  "+w[i]+"\t\t     "+t[i]+"\t\t
"+pp[i]+"\n"); }
System.out.print("\n Average Wait Time : "+awt);
System.out.print("\n Average Turn Around Time : "+atat);
out.println("\n Average Wait Time : "+awt);
out.close(); }}
```

5.2 VERTEX MULTICOLORING PROGRAM

```
functionvarargout = Main_GUI(varargin)
gui_Singleton = 1;
 'gui_OutputFcn',  @Main_GUI_OutputFcn, ...
ifnargin&&ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});
end
ifnargout
[varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
Else
gui_mainfcn(gui_State, varargin{:});
end
functionMain_GUI_OpeningFcn(hObject, eventdata, handles, va
varargout{1} = handles.output;
function L1_Callback(hObject, eventdata, handles)
SeedX=x1(Seed_Input);
SeedY=y1(Seed_Input);
D(Seed_Input)=0;
Data(1).Distance=D;
plot(handles.axes1,SeedX,SeedY,'go')

function L1_CreateFcn(hObject, eventdata, handle
function L2_Callback(hObject, eventdata, handles)
global x2 y2 SeedXSeedY Data
Data(2).XY=[x2,y2];
plot(handles.axes2,x2,y2,'o')
```

```
hold on
function L2_CreateFcn(hObject, eventdata,handles)
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
function L3_Callback(hObject, eventdata, handles)
Data(3).Distance=D ;
[~,ind]=min(D);
function L3_CreateFcn(hObject, eventdata, handles)
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
end
```

VI. SAMPLE OUTPUT:

The output of LA and VM is given below.

6.1 MULTI COLORING VERTEX

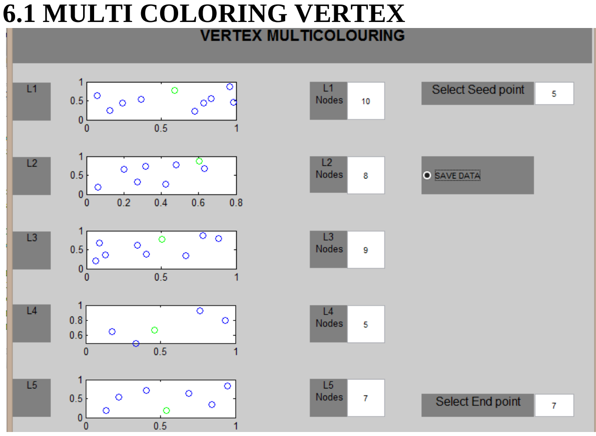
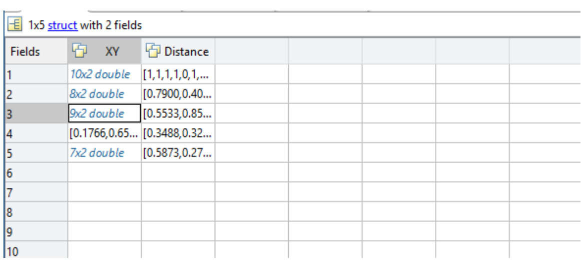


Figure3.a Vertex multicoloring GUI for upto 10 nodes



1x5 struct with 2 fields

Fields	XY	Distance
1	10x2 double	[1,1,1,1,0,1,...
2	8x2 double	[0.7900,0.40...
3	9x2 double	[0.5533,0.85...
4	[0.1766,0.65...	[0.3488,0.32...
5	7x2 double	[0.5873,0.27...
6		
7		
8		
9		
10		

Figure.4. b distance between the nodesupto 10 nodes

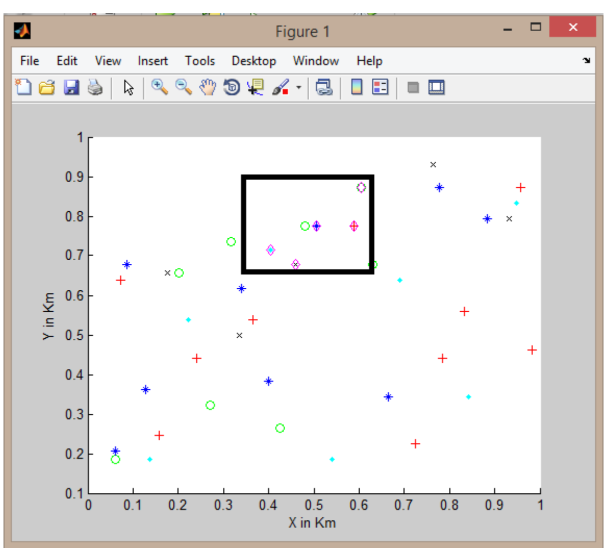


Figure.5. c connecting nodes

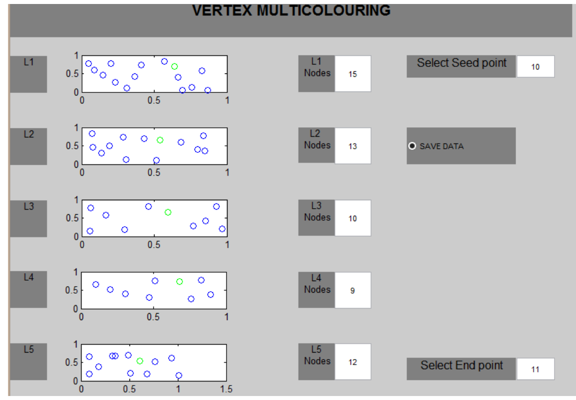
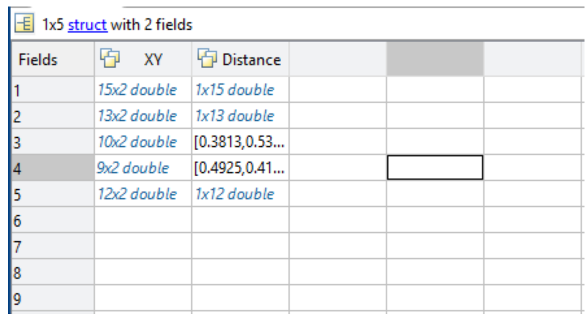


Figure.6. 2a Vertex multicoloring GUI for upto 15 nodes



1x5 struct with 2 fields

Fields	XY	Distance
1	15x2 double	1x15 double
2	13x2 double	1x13 double
3	10x2 double	[0.3813,0.53...
4	9x2 double	[0.4925,0.41...
5	12x2 double	1x12 double
6		
7		
8		
9		

Figure.7.2b distance between the nodes upto 15 node

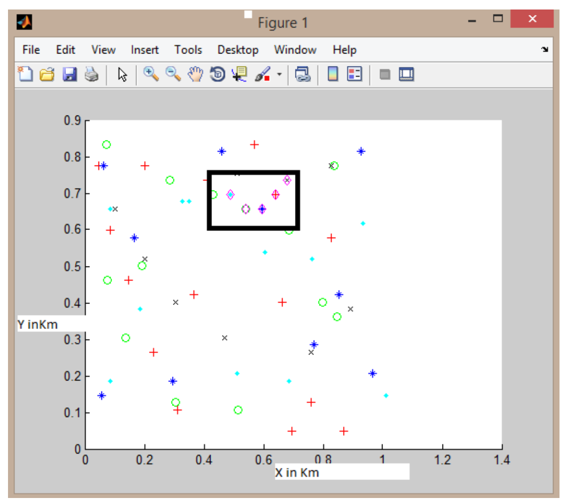


Figure.8.2c connecting nodes

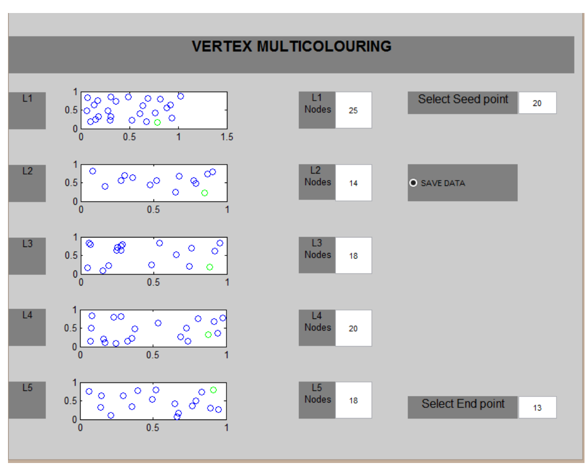


Figure.9. 3a Vertex multicoloring GUI for upto25nodes

Fields	XY	Distance
1	25x2 double	1x25 double
2	14x2 double	1x14 double
3	18x2 double	1x18 double
4	20x2 double	1x20 double
5	18x2 double	1x18 double
6		

Figure.10.3b distance between the nodes upto 25 nodes

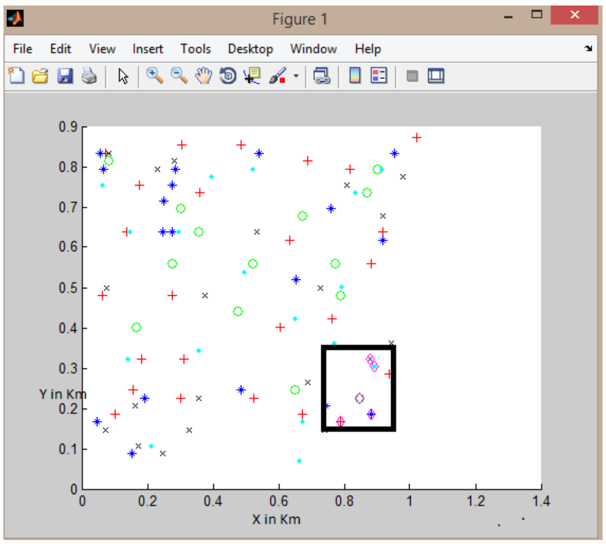


Figure.11. 3c connecting nodes

6.2 OUTPUT FOR LEARNING AUTOMATA

Process	Burst Time	Wait Time	Turn Around Time	Priority
10	36	0	36	43
6	20	36	56	42
8	45	56	101	35
5	13	101	114	32
3	7	114	121	30
2	14	121	135	26
9	8	135	143	22
4	3	143	146	19
1	18	146	164	17
7	16	164	180	5

Average Wait Time : 101
Average Turn Around Time : 119

Figure.12. 1a output for 10 processes

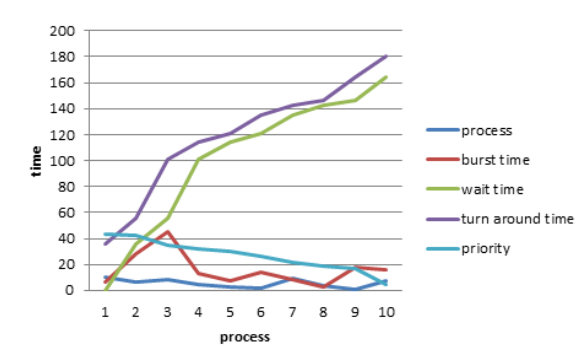


Figure.13.1b graphical output for 10 processes

Process	Burst Time	Wait Time	Turn Around Time	Priority
5	4	0	4	9
3	8	4	12	7
4	6	12	18	5
2	6	18	24	4
1	2	24	26	4

Average Wait Time : 11
Average Turn Around Time : 16

Figure.14. 2a output for 3 processes

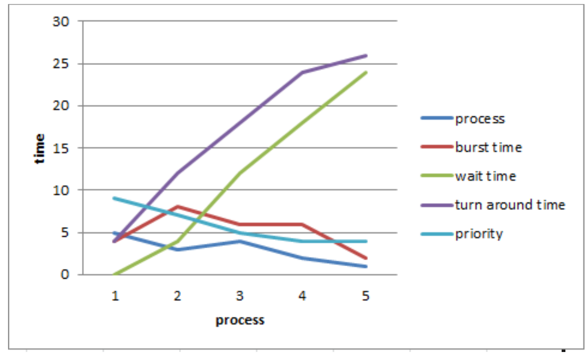


Figure.15. 2b graphical output for 3 processes

Process	Burst Time	Wait Time	Turn Around Time	Priority
2	5	0	5	8
7	5	5	10	8
6	3	10	13	7
3	9	13	22	7
8	9	22	31	5
5	2	31	33	4
1	3	33	36	4
4	6	36	42	1

Average Wait Time : 18
Average Turn Around Time : 24

Figure.16. 3a output for 8 processes

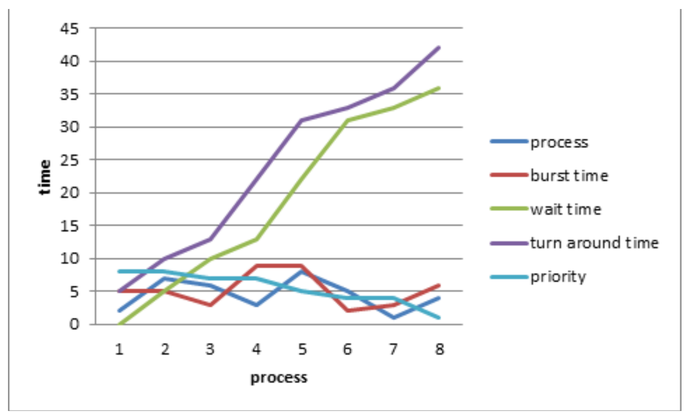


Figure.6. 3b graphical output for 8 processes

VII. CONCLUSION:

This paper presents a scheduling of high density MIMO channels by using learning automata and vertex multi coloring techniques. These techniques help to reduce the complexity of the MIMO channels and also reduce the BER in MIMO. By using Vertex multi coloring low power consumption is achieved. This technique is highly time efficient when compared to existing system.

VIII. REFERENCES:

[1]. Fabio R.J. Vieira,Jose F. de Rezende and ValmirC.Barbosa "Scheduling wireless links by vertex multicoloring in the physical interference model" IEEE Computer Networks 17 February 2016.

[2].Zaeddien Beheshtifard and Mohammed Reza Embody "Maximal throughput scheduling based on the physical interference model using learning automata" IEEE Ad Hoc Networks 3 March 2016.

[3]. Jeffrey G.Andrews ,Stefano Buzzi,Stephen V.Hanly, Anglo Lozano, Anthony C.K Soong, Jianzhongcharlie Zhang "What will be 5G be..?"IEEE June 2014.

[4]. BehrouzFarhang-Boroujeny and Hussein Moradi "OFDM inspired waveforms for 5G"IEEE 2016.

[5].M.A.Perkowski,R.Malavi,S.Grygiel,M.Burns,A.Mishchenko, "Graph coloring Algorithms for Fast Evaluation of Curties Decompositions," Proc. Of Design automation Conference, DAC '99, pp.225-230.

[6]. Appel.K, Haken.W andKoch.J, "Every Planar Map is Four Coloroble". March-21(1977), 429-567.

[7]. https://www.academia.edu/2479839/Applications of graph coloring in modern computer science.

[8]. http://www.geeksforgeeks.org/graph-coloring-set-2-greedy-algorithm/

[9]. Y.Xing, R.Chandramouli, Stochastic learning solution for distributed discrete power control game in wireless data networks, IEEE/ACM Trans. Netw. 16 (4) (2008) 932-944.

[10]. R. Barr, Z.J.Haas, R. Van Renesse, Scalable Wireless ad hoc network simulation, Handbook on Theoretical and Algorithic aspects of sensor, Ad hoc Wireless, and Peer-to-Peer networks, Auerbach Publications, 2005, pp, 297-311.