Review on Graph Coloring Algorithms

Under the Guidance of

Mr. Krishna Kumar Singh
Department of Computer Science and Engineering
RGUKT-NUZVID

By

Ms. Sravani Murakonda
ID Number :N091869
Computer Science and Engineering

Contents

Abstract

✓ To study and present the already existing graph coloring algorithms and analyzing the heuristics, properties and time complexities of different algorithms like Greedy coloring, Sequential coloring, Largest Degree coloring, Incidence Degree Coloring, Saturation Degree Coloring, Independent Set Coloring and Graph coloring by Domination Vertex Covering(DOM). Comparing the output chromatic number($\chi(G)$) given by the algorithms for the various benchmark graphs. To discuss different applications of graph coloring in real time problems and their implementation.

- Abstract
- Introduction
- Related Works
  - Greedy Algorithm[1]
  - Sequential Ordering[3]
  - SDO[3]
  - IDO[16]
  - LDO[15]
  - Independent Set[13]
  - DOM[18]
- Comparison
- Computational Results
- Applications
  - Scheduling Exams[15]
  - Register Allocation[14]
  - Sudoku[17]
  - 8 Queen Problem
  - Traffic Signaling [2]
- Conclusion
- References

Introduction

Vertex coloring:

Function f: V $\rightarrow$ C, such that for all $\{v,w\} \in E$:

$$f(v) \neq f(w)$$

Chromatic number of G: $\chi(G)$: Minimum size of C such that there is a vertex coloring to C.

Vertex coloring problem:

1.Finding the optimal number of colors to color a given graph ?

2.Given: graph G, integer k

Question: Is there a vertex coloring of G with k colors?

Cont….

- ✓ The problem of coloring a graph with the minimum number of colors is well known to be NP-hard, even restricted to k-colorable graphs for constant $k \geq 3$.
- ✓ This review explores the approximation problem of coloring worst-case graphs with as few additional colors as possible in polynomial time.
- ✓ Graph coloring is also closely related to other combinatorial problems such as finding the maximum independent set in a graph (the largest set of vertices such that no two have an edge between them),which is a NP-hard problem.
- ✓ Thus, researchers attempting to find good fast algorithms must consider issues of approximation.

Related works

✓ **A simple graph coloring algorithms.**

Greedy Coloring(G, n)

Input: Graph G(V,E)

2. **for** i = 1 **to** n **do**

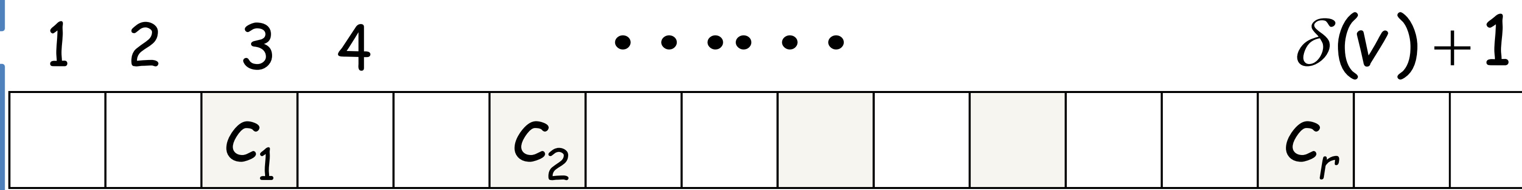
Select v∈ V(G)

Color the vertex with a color min index and different from its neighbor's color.

Output: number of colors required to color the graph G(V,E).

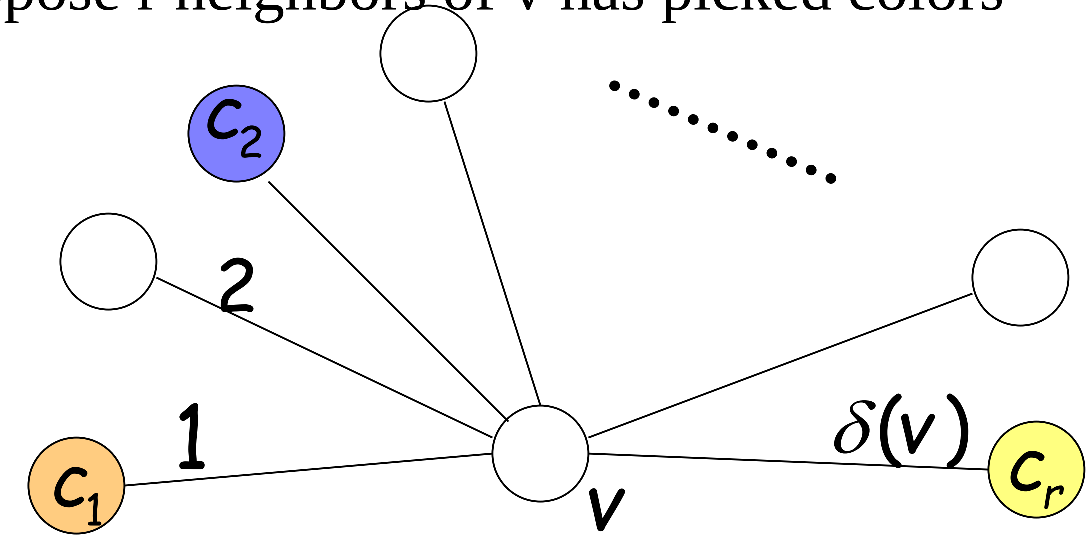
Abstract
Introduction
Related Works
•Greedy Algorithm[1]
•Sequential Ordering[3]
•SDO[3]
•IDO[16]
•LDO[15]
•Independent Set[13]
•DOM[18]
Comparison
Computational Results
Applications
•Scheduling Exams[15]
•Register Allocation[14]
•Sudoku[17]
•8 Queen Problem
•Traffic Signaling [2]
Conclusion
References

Sequential Algorithm:



Color palette of a node v

Suppose r neighbors of v has picked colors



Sequential Coloring Algorithm :

Input: Graph G(V,E)

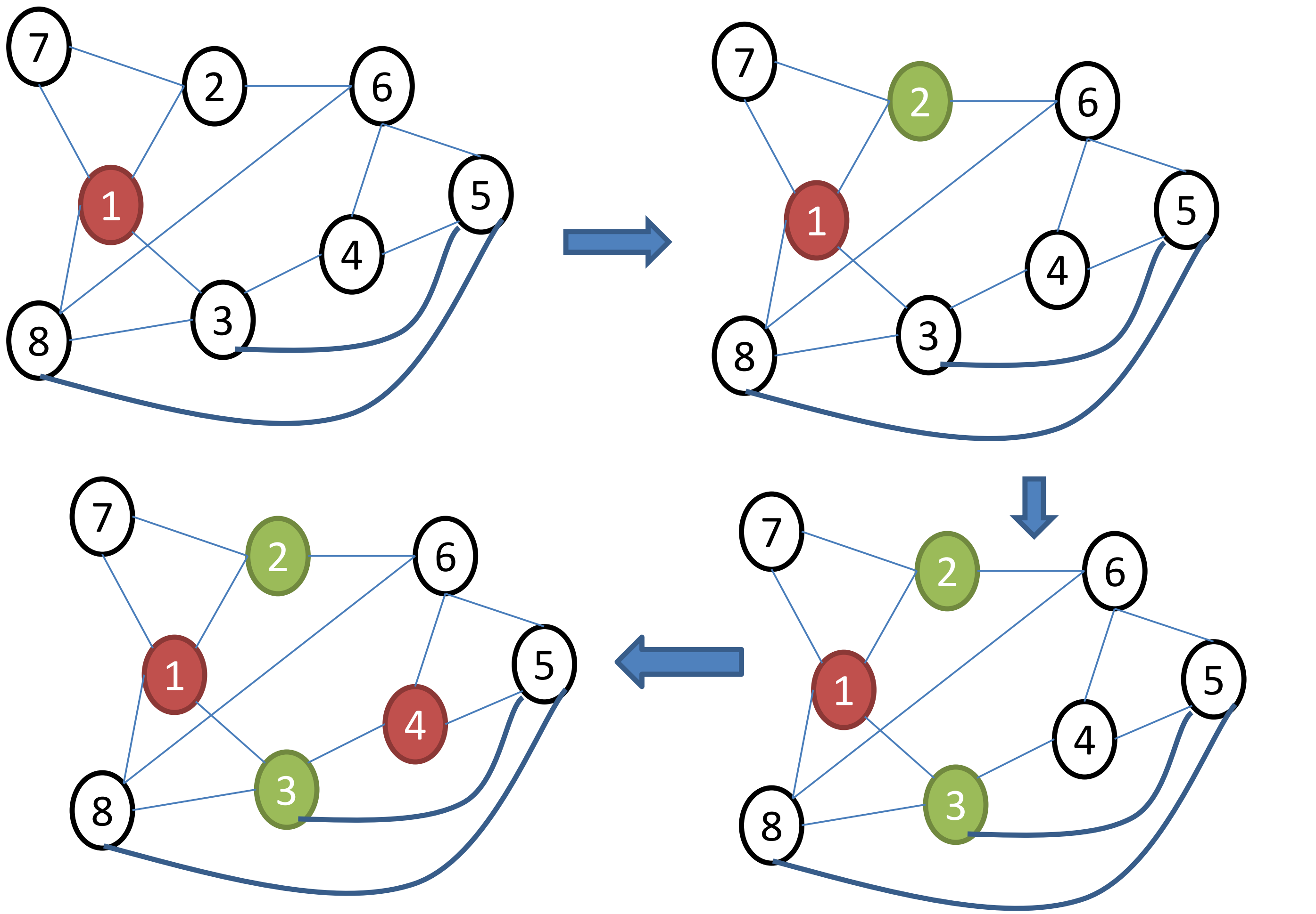
Algorithm:

1. for i=1 to n do
2. Color the node in the order of vertex number
3. Remove the node from the graph G'[V,E]
4. Update the adjacency matrix Adj[n][n] and degree array D[n]
5. return #colors

Output: number of colors required to color the graph G(V,E) in O(n^2) time complexity

Sequential Coloring Algorithm :

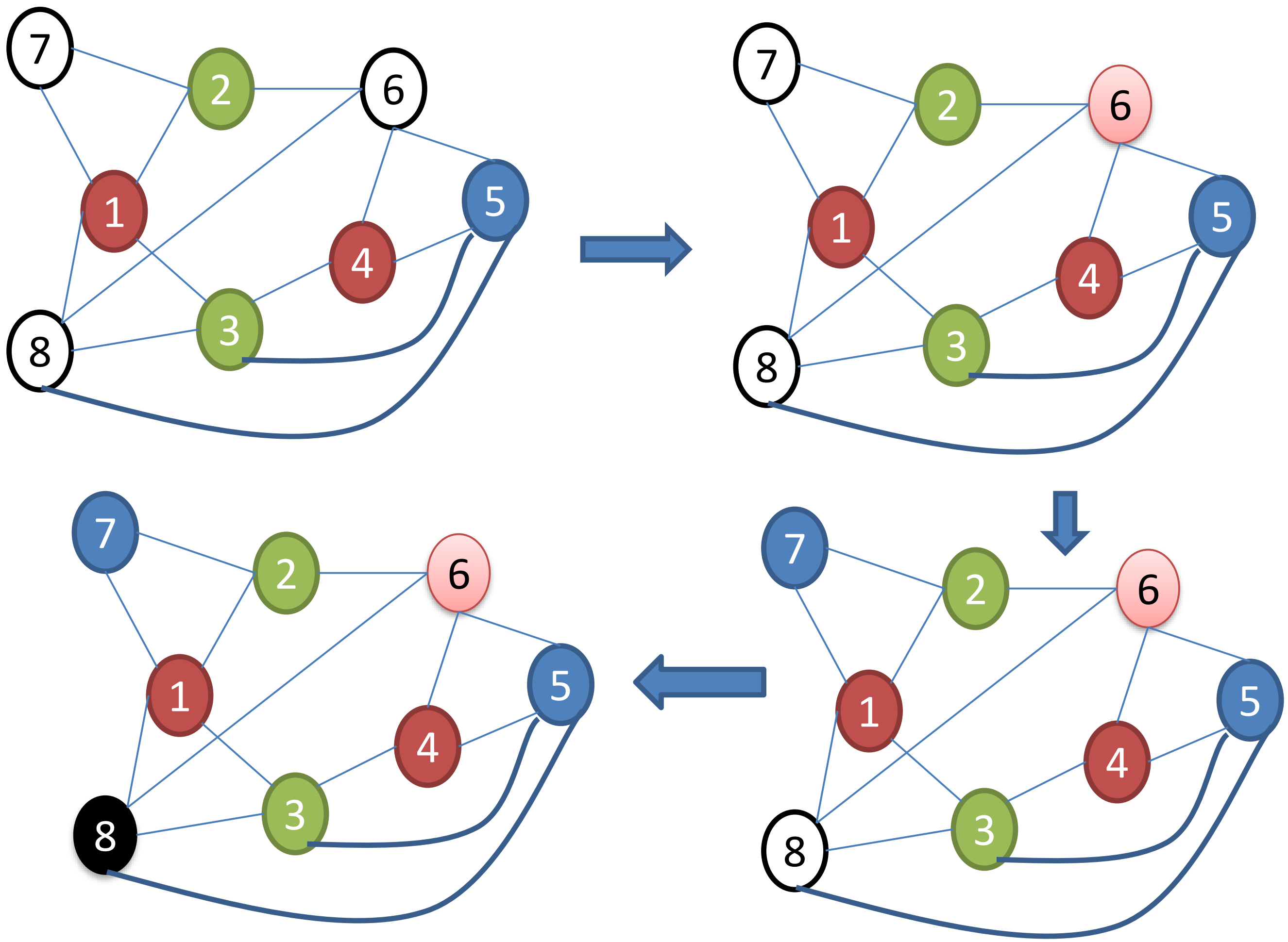
Selects each node in the sequence of nodes



- Abstract
- Introduction
- Related Works
  - Greedy Algorithm[1]
  - Sequential Ordering[3]
  - SDO[3]
  - IDO[16]
  - LDO[15]
  - Independent Set[13]
  - DOM[18]
- Comparison
- Computational Results
- Applications
  - Scheduling Exams[15]
  - Register Allocation[14]
  - Sudoku[17]
  - 8 Queen Problem
  - Traffic Signaling [2]
- Conclusion
- References

Sequential Coloring Algorithm :

Chromatic number is equal to 5.



Saturation Degree Ordering :

Input: Graph G(V,E)

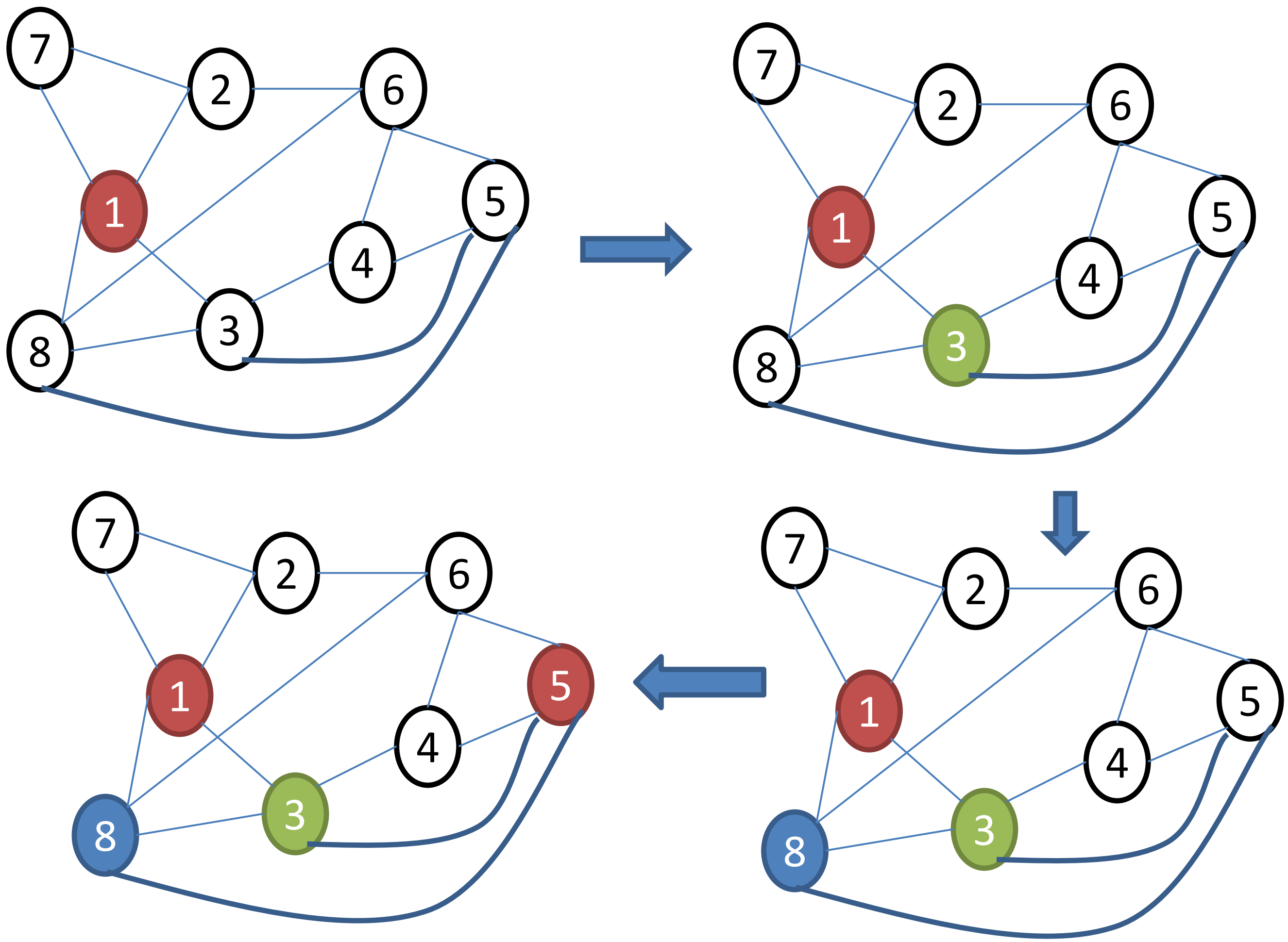
Algorithm:

1. Sort the nodes based on degree of the vertex in descending order
2. for i=1 to n do
3. Color the node with maximum saturation degree with least available index color and if ambiguity is there then color the node with maximum degree
4. Remove the node from the graph G'(V,E)
5. Update the adjacency matrix Adj[n][n] and degree array D[n]
6. Find the maximum saturation degree vertex in the residual graph G'(V,E)
7. return #colors

Output: number of colors required to color the graph G(V,E) in O(n^2) time complexity

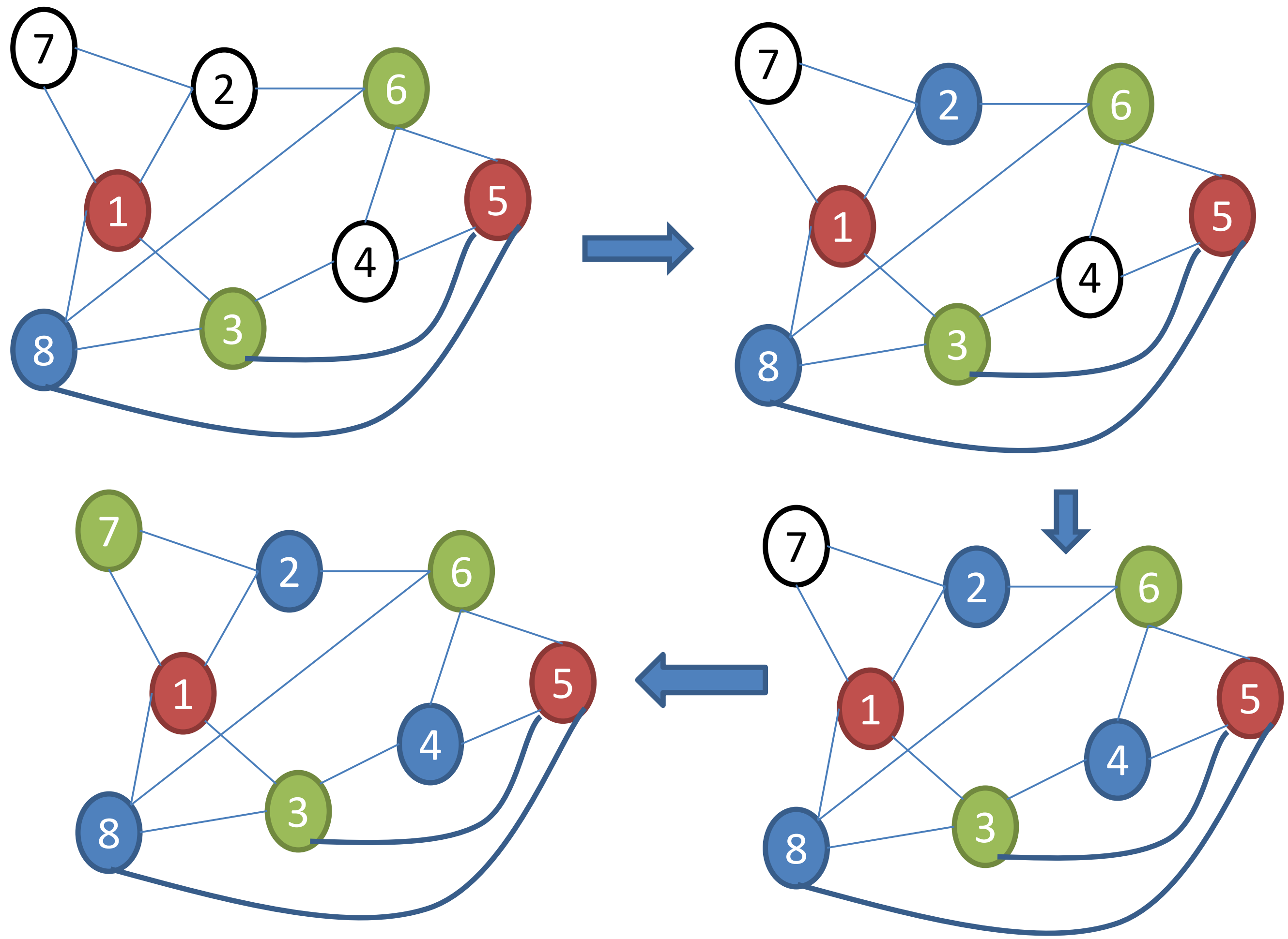
Saturation Degree Ordering :

Selects each node that has max # different colors



Saturation Degree Ordering :

Chromatic number is 3.



Incidence Degree Ordering :

Input: Graph G(V,E)

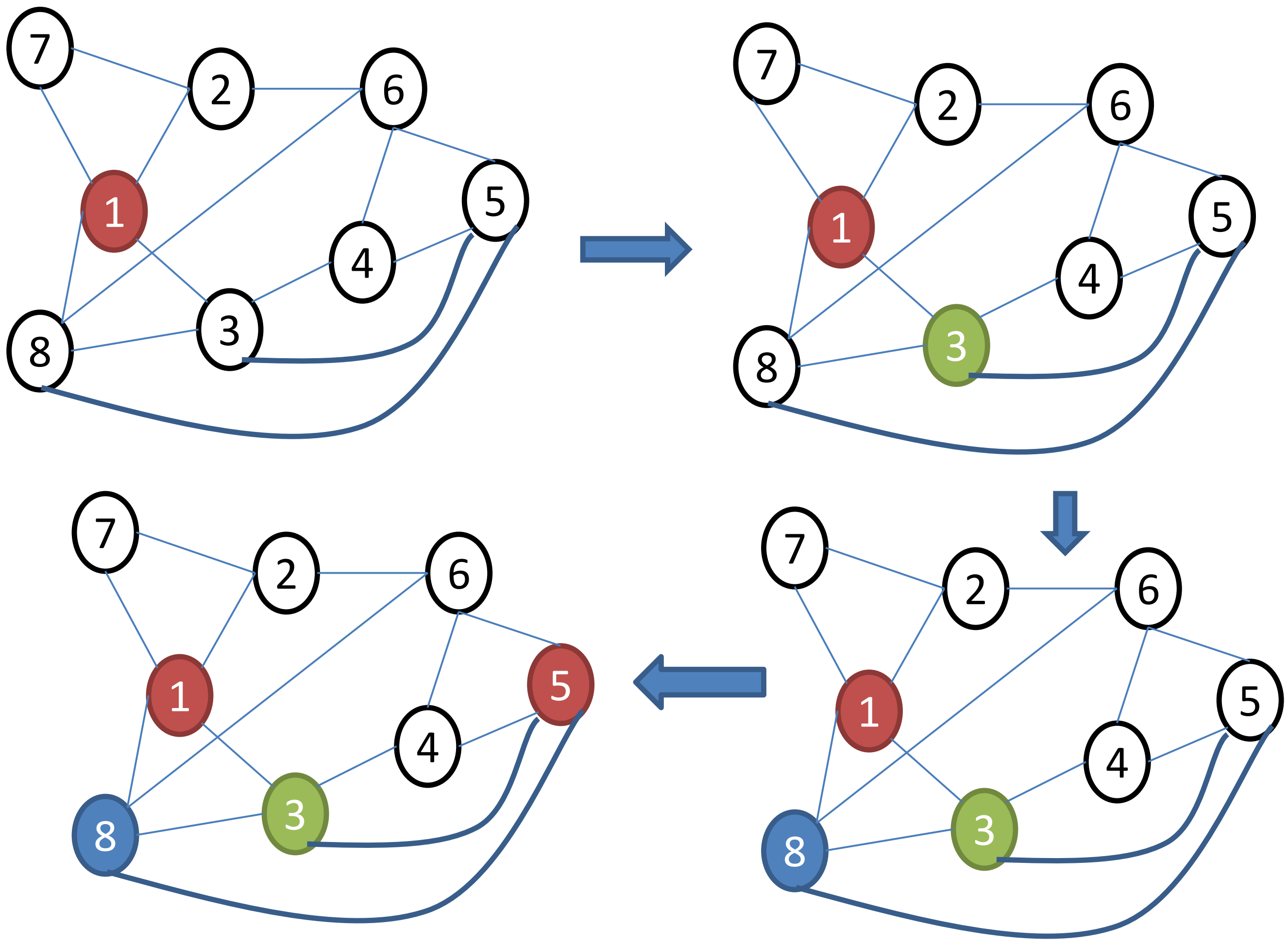
Algorithm:

1. Sort the nodes based on degree of the vertex in descending order
2. for i=1 to n do
3. Color the node with maximum incidence degree with least availables index color and if ambiguity is there then color the node with maximum degree
4. Remove the node from the graph G'(V,E)
5. Update the adjacency matrix Adj[n][n] and degree array D[n]
6. Find the maximum incidence degree vertex in the residual graph G'(V,E)
7. return #colors

Output: number of colors required to color the graph G(V,E) in O(n^2) time complexity.

Incidence Degree Ordering :

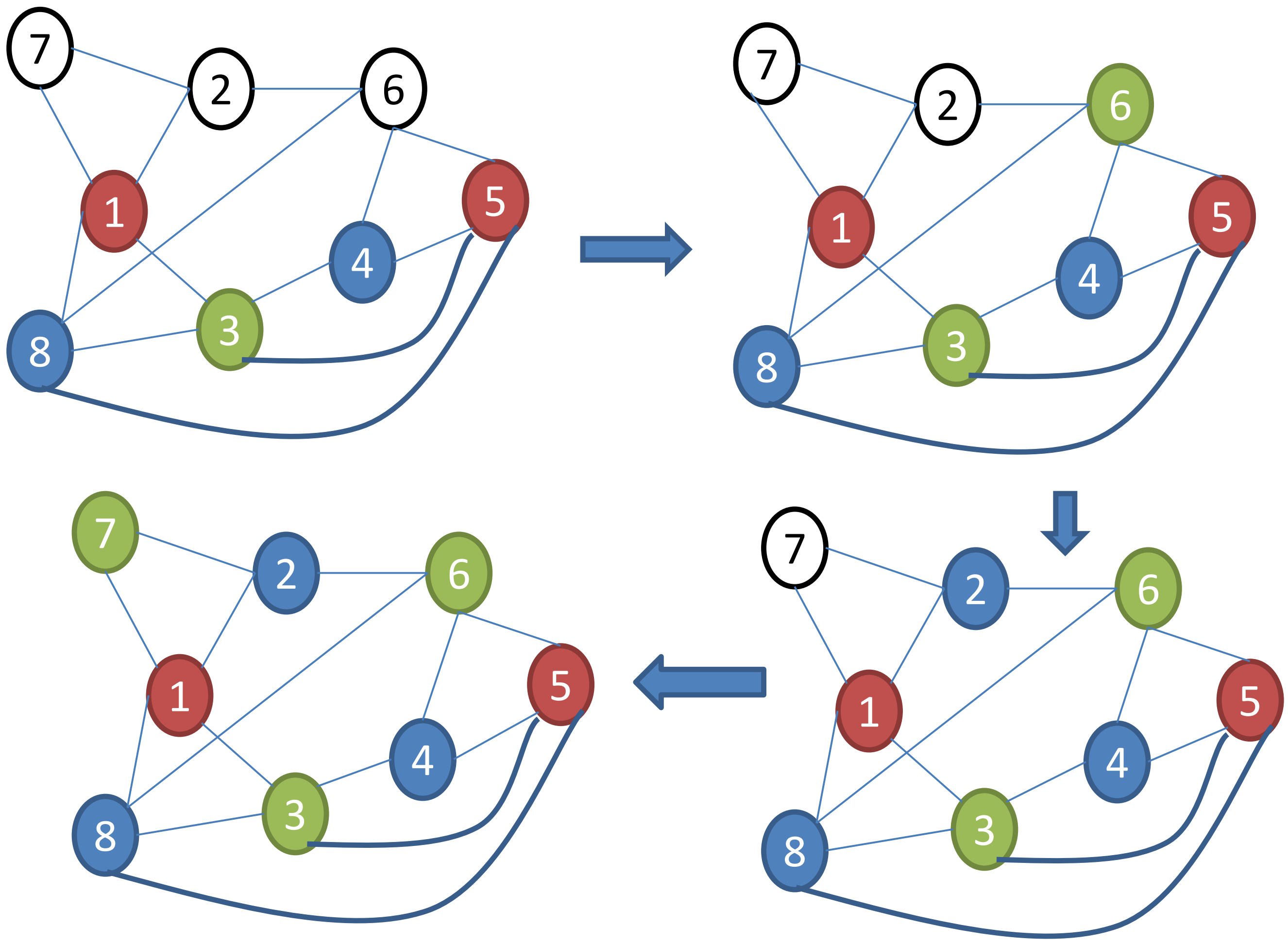
Selects each node that has max colored neighbors



- Abstract
- Introduction
- Related Works
  - [Greedy Algorithm[1]]
  - [Sequential Ordering[3]]
  - [SDO[3]]
  - [IDO[16]]
  - [LDO[15]]
  - [Independent Set[13]]
  - [DOM[18]]
- Comparison
- Computational Results
- Applications
  - [Scheduling Exams[15]]
  - [Register Allocation[14]]
  - [Sudoku[17]]
  - [8 Queen Problem]
  - [Traffic Signaling [2]]
- Conclusion
- References

Incidence Degree Ordering :

Chromatic number is 3.



Largest Degree Ordering :

Input: Graph G(V,E)

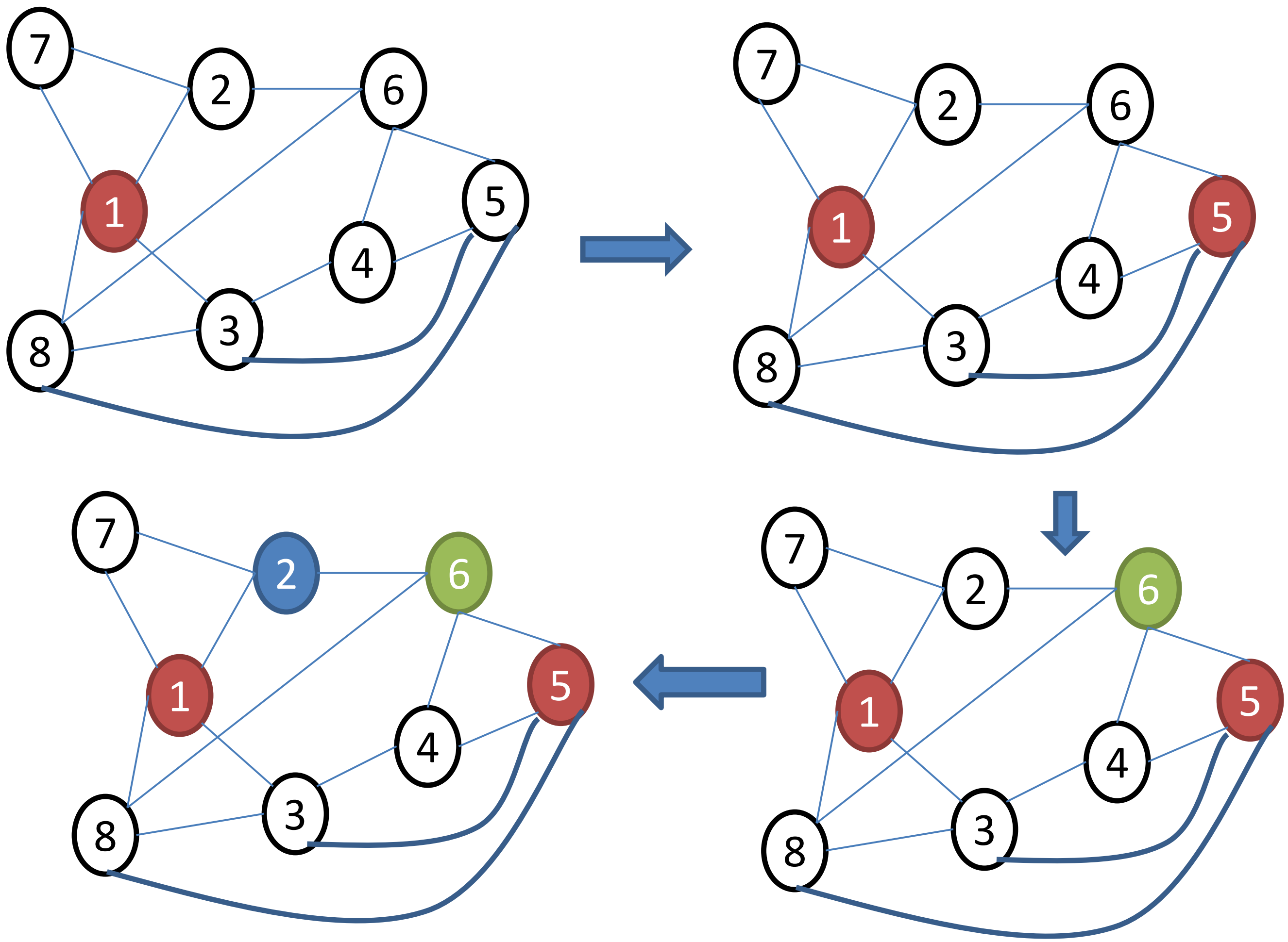
Algorithm:

1. Sort the nodes based on degree of the vertex in descending order
2. for i=1 to n do
3. Color the node with maximum degree with least available index color and if ambiguity is there then color the node with maximum saturation degree
4. Remove the node from the graph G'(V,E)
5. Update the adjacency matrix Adj[n][n] and degree array D[n]
6. Find the maximum degree vertex in the residual graph G'(V,E)
7. return #colors

Output: No.of colors required to color the graph G(V,E) in O(n^2) time complexity.

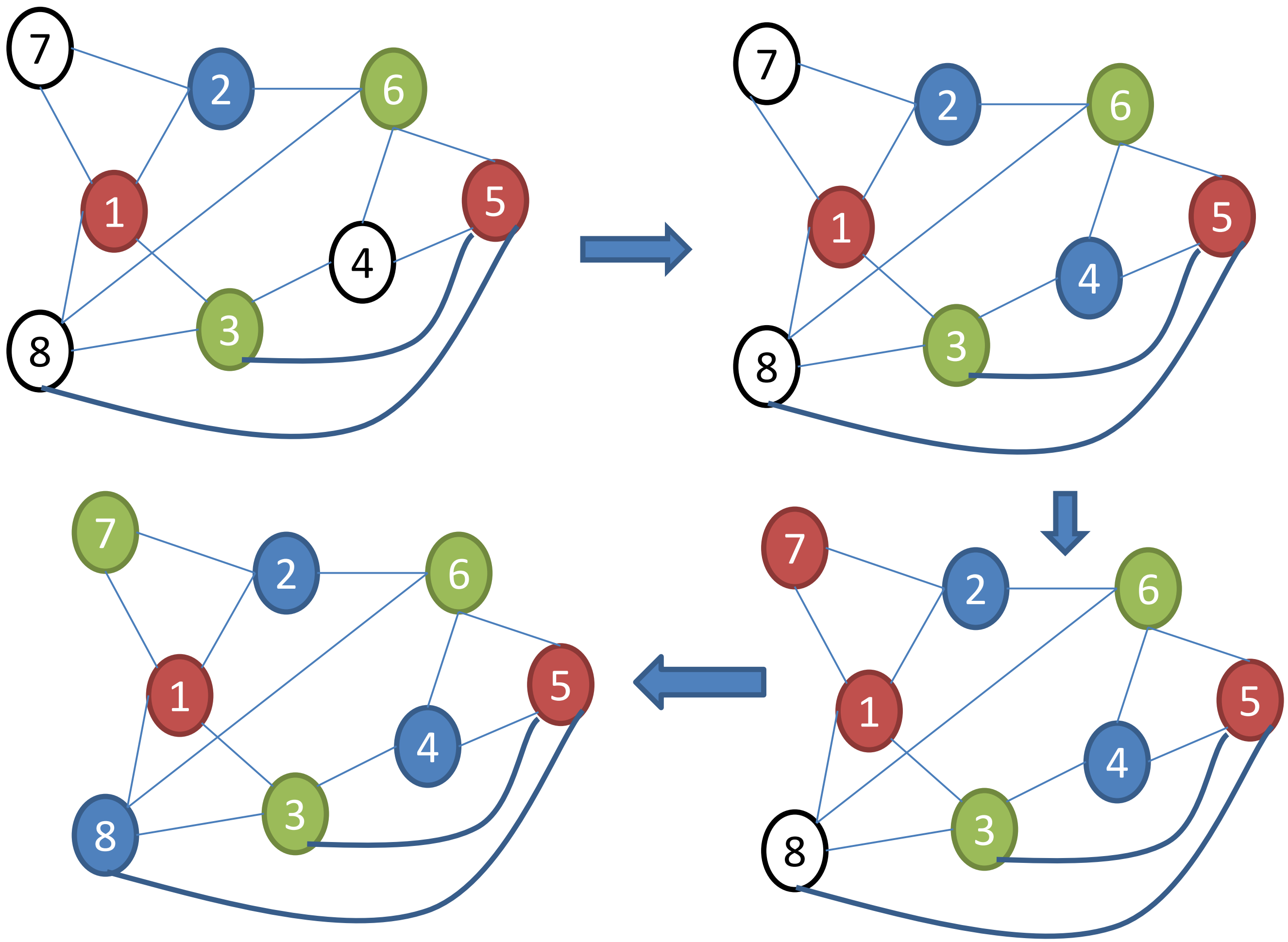
Largest Degree Ordering :

Selects each node that has max degree



Largest Degree Ordering :

Chromatic number is 3



Independent Set :

Input: Graph G(V,E)

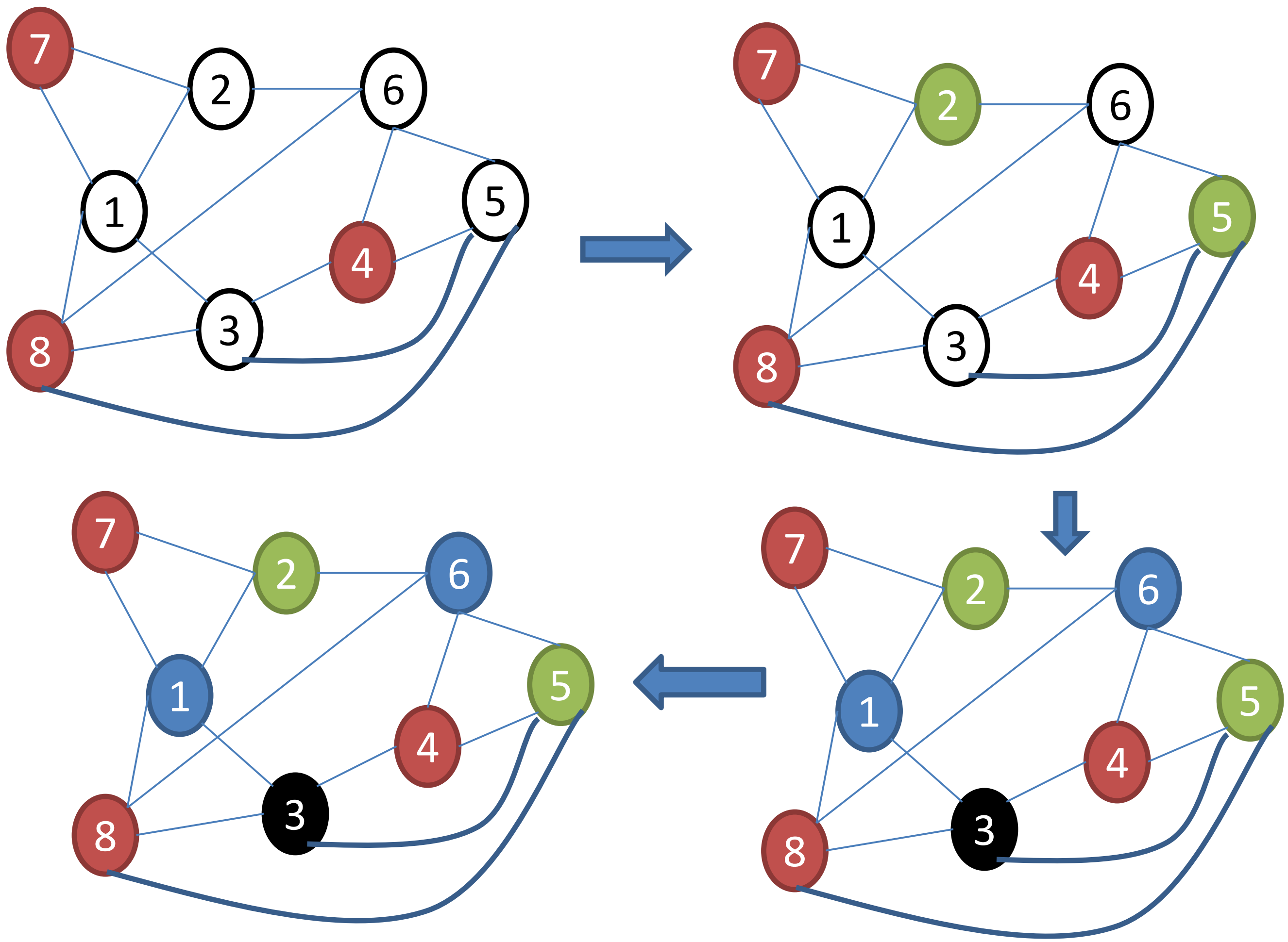
Algorithm:

1. While(G(V,E)!=null) do
2. if(i is not belongs to any list I1,I2…..Im)
3. select all the nodes not connected to i and to other nodes in the selected list with each other
4. Remove the nodes of independent set from the graph G'(V,E)
5. Update the adjacency matrix Adj[n][n]
6. Color each independent set I1,I2…..In with different color
7. return #colors

Output: No. of colors required to color the graph G(V,E) in O(n^3) time complexity

Independent Set :

Selects an independent set in the uncolored graph and assigns with a new color and chromatic number is 4.



Domination Covering:

Definition 1: A node "A" in an incompatibility graph covers some other node "B" in the graph if all of the following are satisfied:

1) Node "A" and node "B" have no common edge.

2) Node "A" has edges with all the nodes that node "B" has edges with.

3) Node "A" has at least one more edge than node "B".

When two nodes have a covering then both the nodes can be colored with the same color.

Definition 2: If conditions 1) and 2) for coverings are satisfied and node "A" has the same number of edges as node "B", then it is called a pseudo-covering.

Theorem 1: If any node "A" in a graph covers any other node "B" in the graph, node "B" can be removed from the graph, and in a pseudo-covering any one of the nodes "A" or "B" can be removed.

Domination Covering:

Definition3: A complete graph is one in which all pairs of vertices are connected.

In a complete graph total edges =n*(n-1)/2, where total edges is the sum of all the edges in a graph. In a complete graph no covering or pseudo coverings can be found and all nodes must have unique color.

Definition4: A non-reducible graph is a graph that is not complete and has no covered or pseudo-covered node(s).

Theorem 2: If a graph is reducible and can be reduced to a complete graph by successive removing of all its covered and pseudo-covered nodes, then Algorithm DOM finds the coloring with the minimum number of colors (the exact coloring).

DOM Algorithm:

Input: Graph G(V,E)

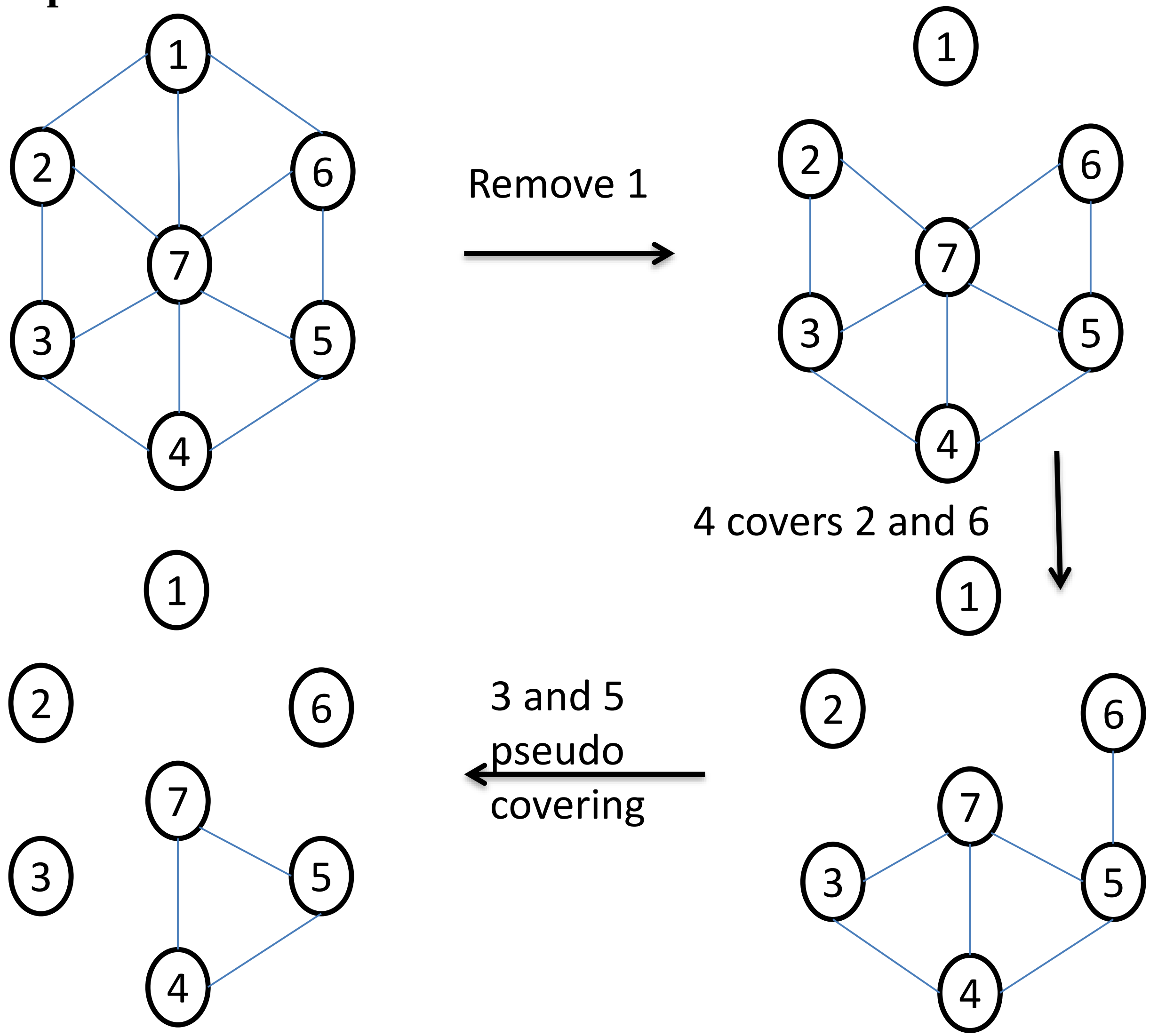
Algorithm:

```
for i=1 to N do
    If(node i is not colored)
        for j=1 to N do
            if (node j is not colored and i!=j and i,j are not connected)
                if(i covers/pseudo covers j)
                    Then remove j and keep it in i th bucket
    if (there is any complete sub graphs as a component in the residual graph G'(V,E) )
        Color the complete graph and its covered and pseudo covered nodes
    else
        Remove a node n randomly from the residual graph G'(V,E) and continue till all the nodes are colored
    Make ith node and its covered nodes as colored
    return #colors
```

Output: number of colors required to color the graph G(V,E) in O(n^3) time complexity.

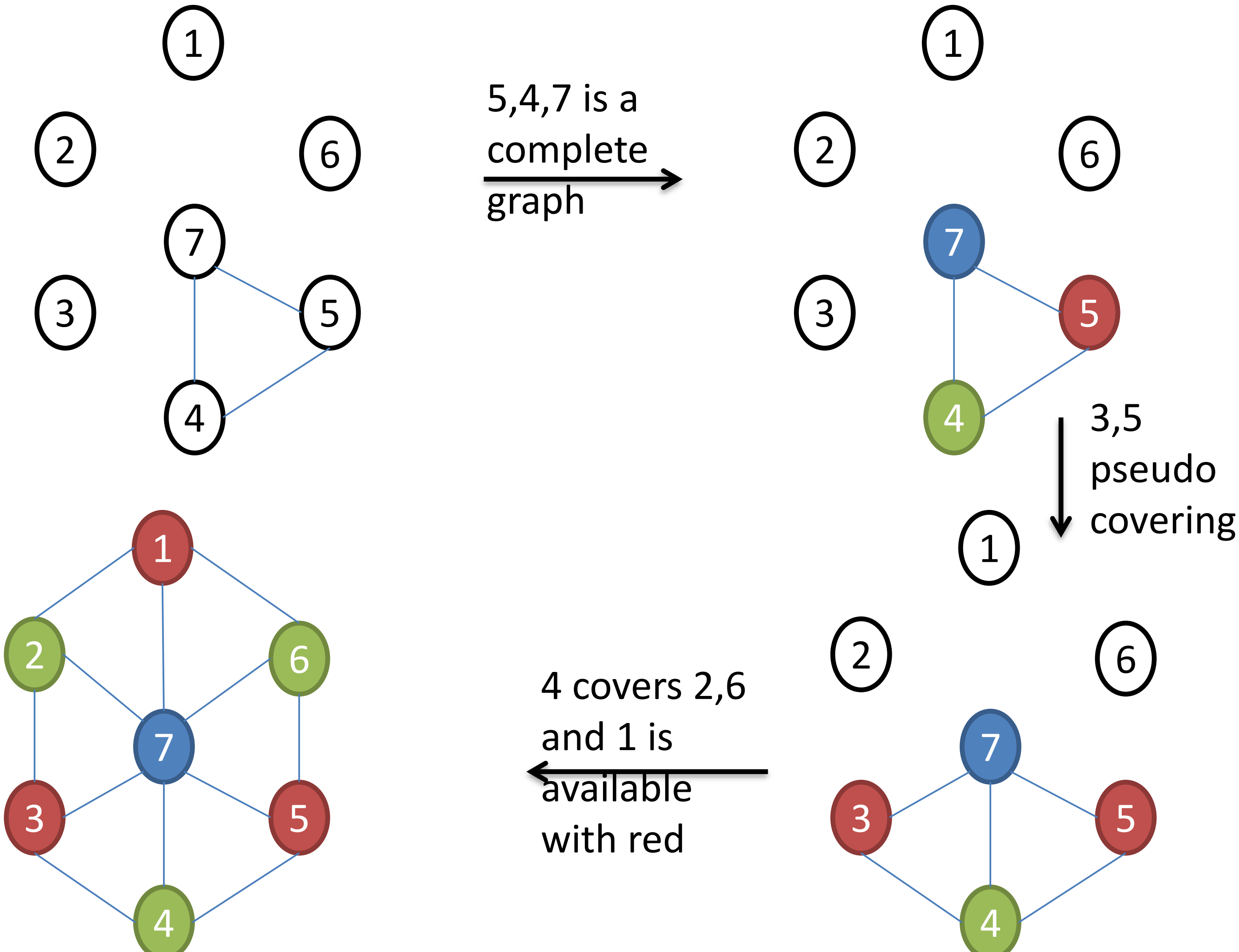
Domination Covering:

Example:



Domination Covering:

Example:



Comparison

Year	Author	Methodology	Remark on Performance
1967	D. J. A Welsh and M. B. Powel	LDO[15]	Output is not optimal
1979	Daniel Brelaz	Greedy Algorithm[3]	Doesn't depend on past or future problems
1979	Daniel Brelaz	SDO[3]	Output is not Optimal always
1983	T. Coleman and J. More	IDO[16]	Output is not optimal always
1986	N. Alon, L. Babai, and A.Itai	Independent Set[13]	Finding Independent set is NPC
1999	Marek Perkowski, Rahul Malvi, Stan Grygiel, Mike Burns, and Alan Mishchenko	Domination Covering for Graph Coloring[18]	Gives optimal solution for some type graphs only
2000	Dimitris A. Fotakis, Spiridon D. Likothanassis, and Stamatis K. Stefanakos	Evolutionary Annealing Approach	Wont give better results if instance size increases

Comparison

Year	Author	Methodology	Remark on Performance
2006	Dr. Hussein Al-Omari and Khair Eddin Sabri	LDO with IDO[3]	Output is not optimal, but it gives better output than LDO
2006	Dr. Hussein Al-Omari and Khair Eddin Sabri	SDO with IDO[3]	Output is not optimal, but it gives better output than SDO
2011	Qinghua Wu and Jin-Kao Hao	EXTRACOL[19]	Gives optimal solution for some graphs only and independent set extraction is also a NPC
2011	Linda Ouerfelli, Hend Bouziri	DSATUR	Not giving optimal solution for dense graph
2012	Hilal ALMARA'BEH, Amjad SULEIMAN	LDO[20],SDO[20],Min-Max[20]	Giving better output for Sparse Graphs

DIMAC Graphs

NAME	No.of vertices	No.of Edges	#colors	LDO	IDO	Ind Set	Domination Vertex Covering (#colors)	
DIMAC GRAPHS				#colors	#colors	#colors	LDO	IDO
abb313GPIA.col	1557	65390	?	12	15	18	13	15
ash331GPIA	662	4185	?	6	8	10	6	7
ash608GPIA	1216	7844	?	6	8	10	6	9
ash958GPIA	1916	12506	?	7	10	10	7	7
DSJC1000.1	1000	49629	20	28	30	29	47	48
DSJC1000.5	1000	499652	83	121	127	121	121	127
DSJC1000.5	1000	249826	85	121	127	121	121	127
DSJC1000.9	1000	449449	223	300	300	313	306	317
DSJC250.1	250	3218	8	11	11	11	11	13
DSJC250.5	250	15668	28	39	41	41	39	41
DSJC250.9	250	27897	72	89	95	93	95	95
DSJC500.1	500	12458	12	18	20	18	17	18
DSJC500.5	500	125248	48	69	71	71	69	71
DSJC500.5	500	62624	48	69	71	71	69	71
DSJC500.9	500	224874	126	166	177	169	166	177
DSJR500.1	500	3555	12	18	20	13	13	13
DSJR500.1c	500	121275	84	96	103	100	93	94
DSJR500.5	500	58862	122	136	129	134	136	129
latin_square_10.col	900	307350	?	148	184	213	148	184
le450_15a.col	450	8168	15	18	19	18	18	17
le450_15b.col	450	8169	15	18	18	18	18	18
le450_15c.col	450	16680	15	26	27	26	26	27
le450_15d.col	450	16750	15	26	26	26	26	26
le450_25b.col	450	8263	25	25	25	25	25	25
le450_25c.col	450	17343	25	31	31	29	30	30
le450_25d.col	450	17425	25	31	29	30	31	31
le450_5a.col	450	5714	5	12	11	11	11	11
le450_5b.col	450	5734	5	15	15	12	11	11
le450_5c.col	450	9803	5	10	12	12	10	13
le450_5d.col	450	9757	5	10	14	14	11	12

- Abstract
- Introduction
- Related Works
  - •Greedy Algorithm[1]
  - •Sequential Ordering[3]
  - •SDO[3]
  - •IDO[16]
  - •LDO[15]
  - •Independent Set[13]
  - •DOM[18]
- Comparison
- Computational Results
- Applications
  - •Scheduling Exams[15]
  - •Register Allocation[14]
  - •Sudoku[17]
  - •8 Queen Problem
  - •Traffic Signaling [2]
- Conclusion
- References

Name	No.of Vertices	No.of Edges	LDO(#colors)			IDO(#colors)				Domination Vertex Covering(#colors)	
			Least Color	Random	Half Random	Least Color	Random	Half Random	Ind-Set	LDO	IDO
brock2001_3.cq.b	200	12048	39	39	39	43	47	45	42	41	41
brock2001_4.cq.b	200	13089	44	46	45	45	52	48	46	44	49
brock4001_1.cq.b	400	59723	92	95	92	98	104	98	98	92	98
brock4001_2.cq.b	400	59786	96	98	97	101	104	98	100	96	101
brock4001_3.cq.b	400	59681	95	98	94	100	108	103	97	95	100
brock4001_4.cq.b	400	59765	95	98	98	101	105	99	96	96	101
brock8001_1.cq.b	800		139	145	139	140	154	145	142	139	140
brock8001_2.cq.b	800		138	145	141	143	154	145	142	138	143
brock8001_3.cq.b	800		137	143	137	143	156	143	141	137	143
brock8001_4.cq.b	800		138	149	144	142	152	143	142	138	142
C-fat200-1.clq	200	3235	13	13	13	15	12	15	14	15	13
C-fat200-2.clq	200	3235	24	24	24	24	25	24	24	24	24
C-fat200-5.clq	200	8473	82	82	82	84	84	84	84	84	84
C-fat500-1.clq	500	4459	18	18	18	14	15	16	14	17	14
C-fat500-10.clq	500	46627	127	127	127	126	126	126	126	127	126
C-fat500-2.clq	500	9139	33	33	33	26	27	26	26	33	26
C-fat500-5.clq	500	23191	78	79	79	64	66	64	64	78	64
gen200_p0.9_44.b	200	17910	68	72	69	62	70	66	68	66	60
gen200_p0.9_55.b	200	17910	71	74	73	73	78	74	74	71	72
gen400-p0.9-55	400		120	128	126	94	104	101	129	122	87
gen400-p0.9-65	400		130	136	130	87	96	91	129	131	84
gen400-p0.9-75	400		129	138	139	101	115	101	128	132	101
hamming6-4.clq	64	704	8	8	8	9	9	10	8	8	9
keller4.clq.b	171	9435	32	32	32	35	34	31	37	31	31
Keller5.clq.b	776		107	116	106	118	123	114	175	123	108
keller6.clq.b	3361		260	312	285	380	400	350	781	380	260
MANN_a27.clq	378	70551	141	141	141	144	144	144	144	126	126
MANN_a45.clq	1035		372	372	372	375	375	375	375	338	338
r200.5	200	10036	34	35	34	36	35	36	35	34	36
r400.5	400	40061	57	58	57	59	58	59	58	57	59
r500.5	500	62161	68	70	68	69	68	69	69	68	69

- Abstract
- Introduction
- Related Works
  - Greedy Algorithm[1]
  - Sequential Ordering[3]
  - SDO[3]
  - IDO[16]
  - LDO[15]
  - Independent Set[13]
  - DOM[18]
- Comparison
- Computational Results
- Applications
  - Scheduling Exams[15]
  - Register Allocation[14]
  - Sudoku[17]
  - 8 Queen Problem
  - Traffic Signaling [2]
- Conclusion
- References

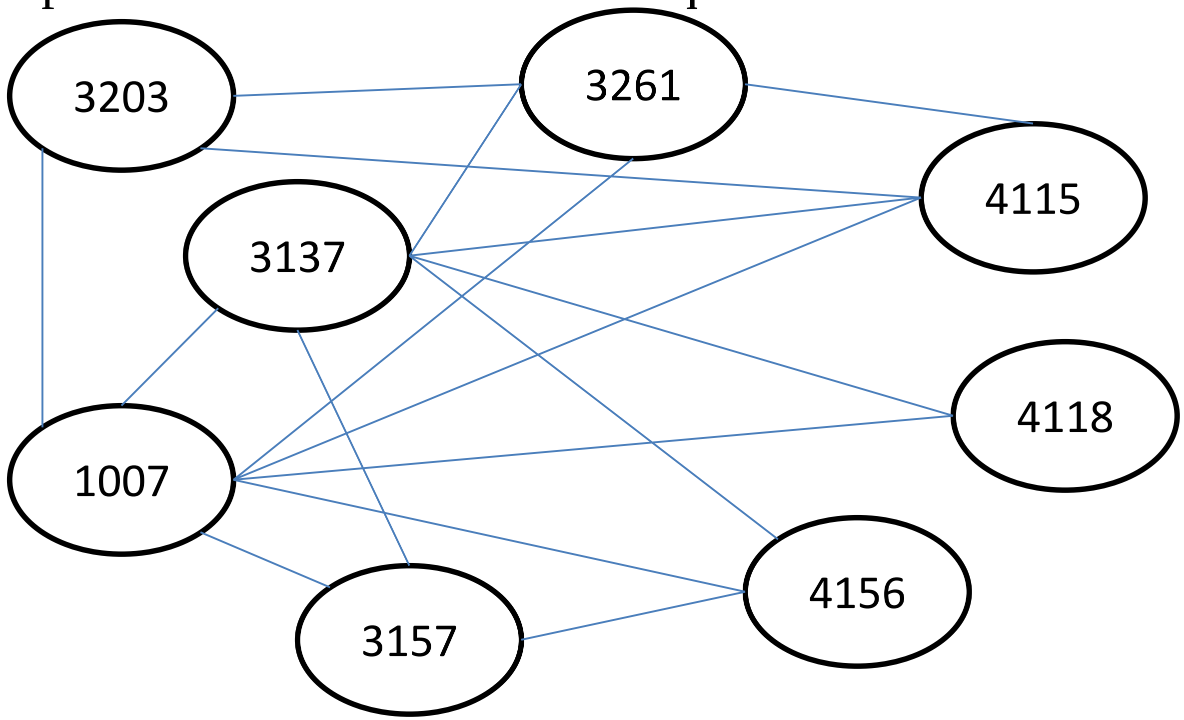
Applications

- ✓ Scheduling Final Exams
- ✓ Traffic signal design
- ✓ Sudoku
- ✓ 8 Queen Problem
- ✓ Register allocation
- ✓ VLSI channel routing
- ✓ Testing printed circuit boards (Garey, Johnson, & Hing)

Scheduling Exams:

- Final Exam Example
- Suppose want to schedule some final exams for CS courses with following course numbers:
  1007, 3137, 3157, 3203, 3261, 4115, 4118, 4156
- Suppose also that there are no students in common taking the following pairs of courses:
  - 1007-3137
  - 1007-3157, 3137-3157
  - 1007-3203
  - 1007-3261, 3137-3261, 3203-3261
  - 1007-4115, 3137-4115, 3203-4115, 3261-4115
  - 1007-4118, 3137-4118
  - 1007-4156, 3137-4156, 3157-4156
- How many exam slots are necessary to schedule exams?

- Convert problem into a graph coloring problem.
- Courses are represented by vertices.
- Two vertices are connected with an edge if the corresponding courses have a student in common.
- One way to do this is to put edges down where students mutually excluded and then compute the complementary graph and chromatic number is equal to 3.



Register Allocation:

- *L1: li a, 0*
- *loop: addi b, a, 1*
- *L2:add c, c, b*
- *L3:muli a, b, 2*
- *L4: blt a, n, loop*
- *L5:print c*

Label	Live-i
L1	*c, n*
loop	*a, c, n*
L2	*b, c, n*
L3	*b, c, n*
L4	*a, c, n*
L5	*c*

Steps:

1. Make a node for each register
2. Are V1 and V2 ever live together, if so keep an edge between both the vertices.
3. Do the coloring for the graph.

Sudoku:

- ✓ Each cell is a vertex
- ✓ Each integer label is a “color”
- ✓ A vertex is adjacent to another vertex if one of the following hold:
  - Same row
  - Same column
  - Same 3x3 grid

- ✓ Vertex-coloring solves Sudoku

8 Queen Problem:

- ✓ Each cell is a vertex
- ✓ Each integer label is a "color"
- ✓ A vertex is adjacent to another vertex if one of the following hold:
  - Same row
  - Same column
  - Same Diagonal Element
- ✓ Vertex-coloring solves Sudoku

Abstract

Introduction

Related Works
- Greedy Algorithm[1]
- Sequential Ordering[3]
- SDO[3]
- IDO[16]
- LDO[15]
- Independent Set[13]
- DOM[18]

Comparison

Computational Results

Applications
- Scheduling Exams[15]
- Register Allocation[14]
- Sudoku[17]
- 8 Queen Problem
- Traffic Signaling [2]

Conclusion

References

Traffic signal design:

✓ At an intersection of roads, we want to install traffic signal lights which will periodically switch between green and red.

✓ The goal is to reduce the waiting time for cars before they get green signal.

✓ This problem can be modeled as a coloring problem.

✓ Each path that crosses the intersection is a node. If two paths intersect each other, there is an edge connecting them. Each color represents a time slot at which the path gets a green light.

Conclusion

✓ Using all the mentioned algorithms we can find the chromatic number of the given graph and the solution is around optimal solution with polynomial time complexity using all the algorithms and its properties.

Abstract

Introduction

Related Works

- Greedy Algorithm[1]
- Sequential Ordering[3]
- SDO[3]
- IDO[16]
- LDO[15]
- Independent Set[13]
- DOM[18]

Comparison

Computational Results

Applications

- Scheduling Exams[15]
- Register Allocation[14]
- Sudoku[17]
- 8 Queen Problem
- Traffic Signaling [2]

Conclusion

References

References

[1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms,* Third Edition. MIT Press/McGraw-HillHigher Education, 2009.

[2] Narsingh Deo, *Graph Theory with Application to Engineering and Computer Science*, Prentice-Hall, Englewood Cliffs, N.J., 1974.

[3] Daniel Brelaz, *New Methods to Color the Vertices of a Graph*,, ACM New York, Vol 22 Issue4,1979.

[4] Dr. Hussein Al-Omari and Khair Eddin Sabri, *New Graph Coloring Algorithms*, Computer Science Department, Applied Science University, Amman, Jordan, 2006.

[5] Desislava Kukova, *Chromatic Number* , 30 Car Osvoboditel STR, ap. 7, 9000 Varna, Bulgaria, desi_kukova@abv.bg

[6] Marco Chiarandini and Thomas Stutzle, *An Analysis of Heuristics for Vertex Coloring,* University of Southern Denmark, Campusvej 55,Odense, Denmark,2010.

[7] J. Randall-Brown, “Chromatic scheduling and the chromatic number problems,” Management Science 19(4), Part I, pp. 456-463, 1972.

[8] P. Cheeseman, B. Kenefsky, and W. Taylor, Where the really hard problems are, In J.Mylopoulos and R. Reiter (Eds.), Proceedings of 2th International Joint Conference on AI (IJCAI-91), Volume 1, pp. 331–337., 1991.

[9] S. Vishwanathan, Randomized online graph coloring, Journal of Algorithms,1992.

[10] A. Miller, Online graph colouring, Canadian Under graduate Mathematics Conference, 2004.

[11] [http://en.wikipedia.orglwiki/Greedy_algorithm]

[12] [http://www.cse.ohiostate.edul-gurari/course/cis680/cis680ChI7.html#QQ 1-49-1 07]

[13] N. Alon, L. Babai, and A.Itai, A fast and simple randomized parallel algorithm for the maximal independent set problem, J. Algorithms , 1986.

[14] P. Briggs. Register allocation via graph coloring. PhD thesis, Rice University , 1992.

[15] D. J. A Welsh and M. B. Powell. An upper bound for the chromatic number of a graph and its application to time tabling problems. The computer Journal, 1967.

[16] T. Coleman and J. More. Estimation of sparse Jacobian matrices and graph coloring problems. SIAM J. Numer. Anal., 1983.

[17] Vlastimil Chytry. Sudoku Game solution based on Graph Theory and Suitable for School Mathematics. 2014

[18] Marek Perkowski, Rahul Malvi, Stan Grygiel, Mike Burns, and Alan Mishchenko. Graph Coloring Algorithms for Fast Evaluation of Curtis Decompositions, Portlan University, USA, mperkows@ee.pdx.edu.2010.

[19] Qinghua Wu and Jin-Kao Hao, Coloring Large Graphs based on Independent Set Extraction, LERIA,France ,2011.

[20] Hilal ALMARA’BEH, Amjad SULEIMAN, Heuristic Algorithm for Graph Coloring Based On Maximum Independent Set, *Journal of Applied Computer Science & Mathematics, no. 13 (6) , Suceava*, Saudi Arabia,2012.

Thank You