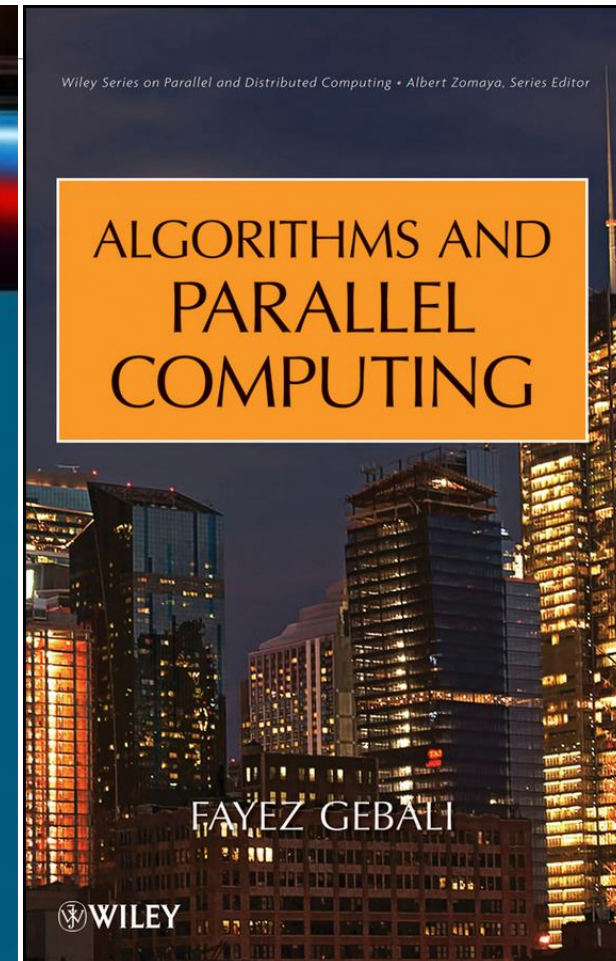
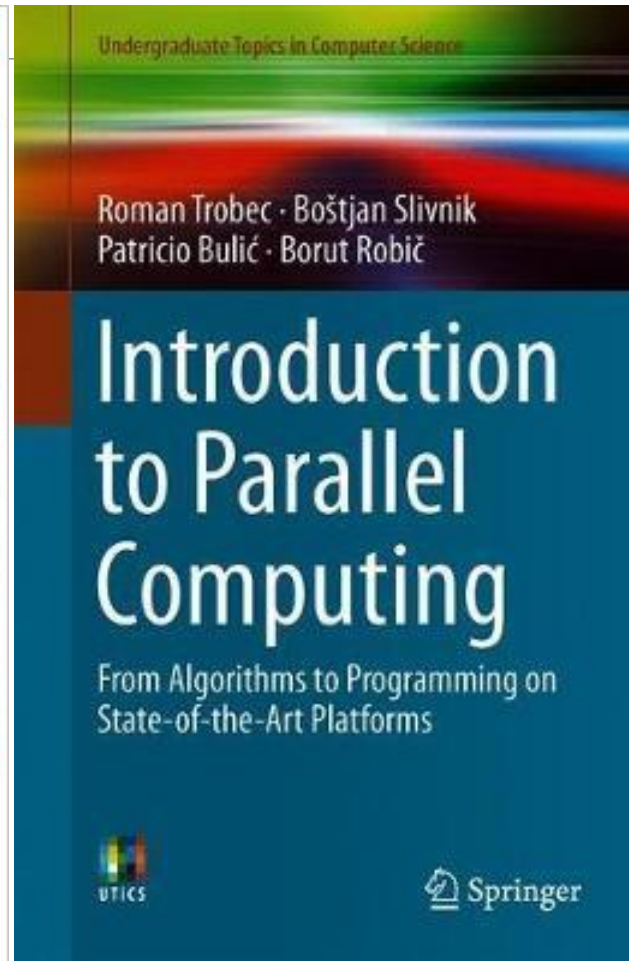
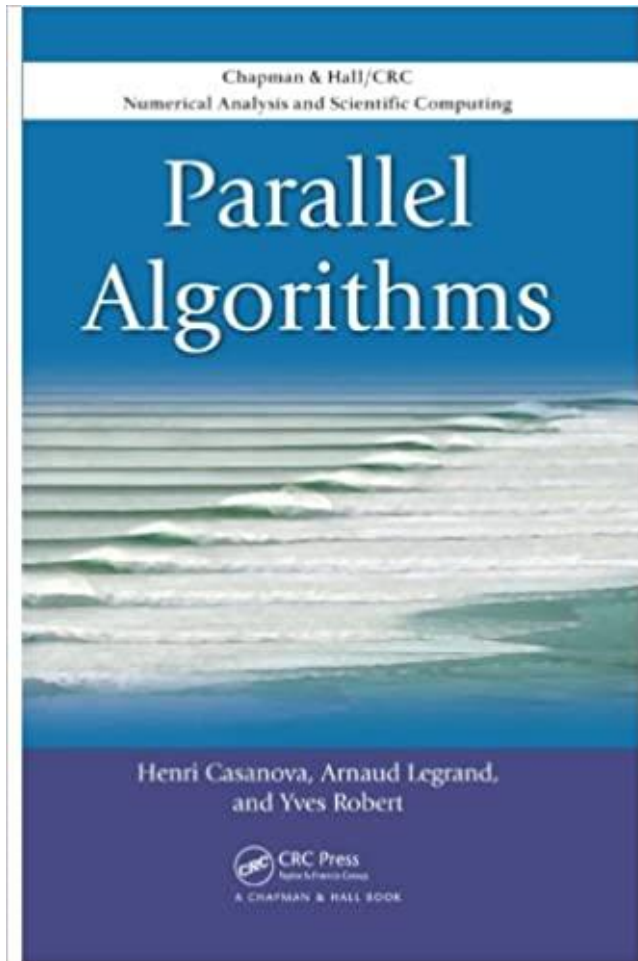


Parallel Programming

Lec 8

Books



PowerPoint

<http://www.bu.edu.eg/staff/ahmedaboalatah14-courses/14779>

The screenshot displays a web interface for Benha University. At the top, the university logo and name are on the left, and the user's name 'Ahmed Hassan Ahmed Abu El Atta' with a 'Log out' link is on the right. A navigation menu on the left lists various university services. The main content area shows the user's current location ('Home/Courses/Compilers') and the course title 'Ass. Lect. Ahmed Hassan Ahmed Abu El Atta :: Course Details: Compilers'. Below this, there are two tables: one for course details and another for management actions.

Benha University Staff Search: **Welcome: Ahmed Hassan Ahmed Abu El Atta (Log out)**

You are in: [Home/Courses/Compilers](#) [Back To Courses](#)

Ass. Lect. Ahmed Hassan Ahmed Abu El Atta :: Course Details: Compilers [add course](#) | [edit course](#)

Course name	Compilers
Level	Undergraduate
Last year taught	2018
Course description	Not Uploaded

Course password

Course files	add files
Course URLs	add URLs
Course assignments	add assignments
Course Exams & Model Answers	add exams

Navigation menu (left): Benha University, Home, النسخة العربية, My C.V., About, Courses, Publications, **Inlinks(Competition)**, Theses, Reports, Published books, Workshops / Conferences, Supervised PhD, Supervised MSc, Supervised Projects, Education, Language skills, Academic Positions, Administrative Positions

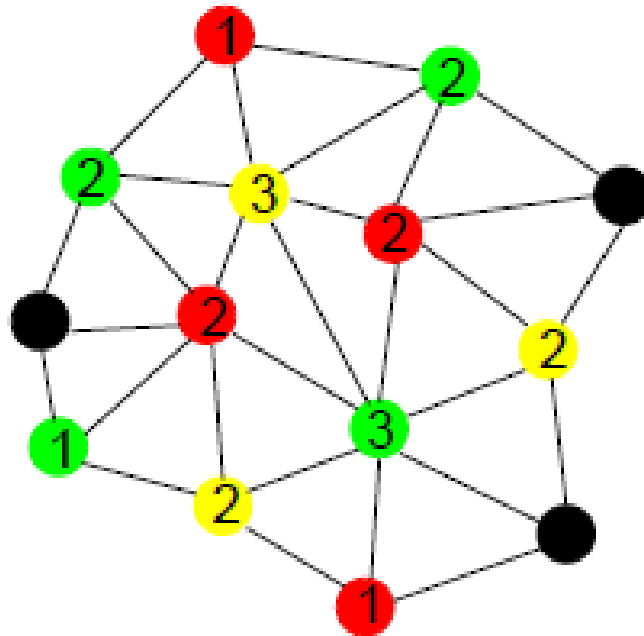
Social media icons (right): Google, Benha University, RG, in, f, Twitter, g+, YouTube, W, Instagram, RSS, Z, (edit)

Graph Coloring Algorithms for Shared Memory Architecture

Problem Statement

Given a simple graph $G = (V; E)$.

Assign colors to the vertices of the graph such that no two adjacent vertices are assigned the same color.



Sequential Algorithms

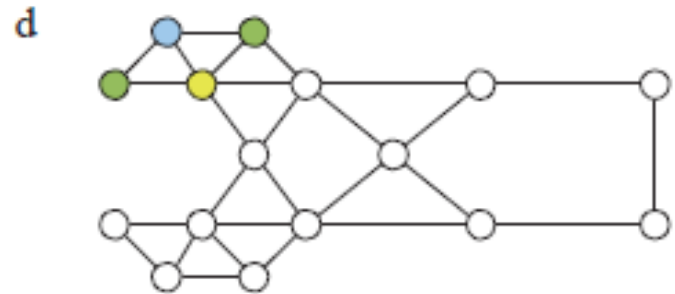
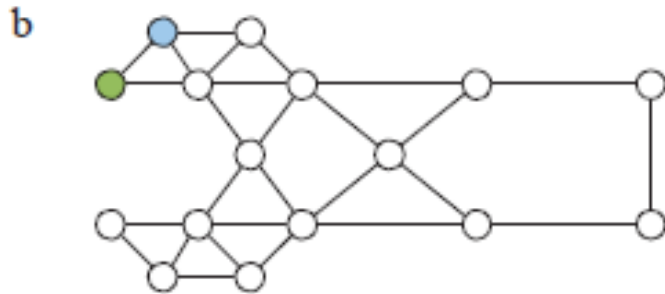
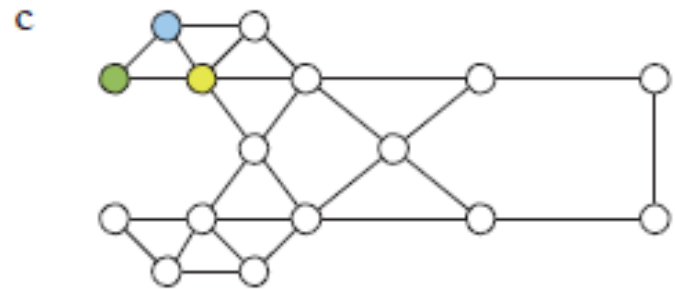
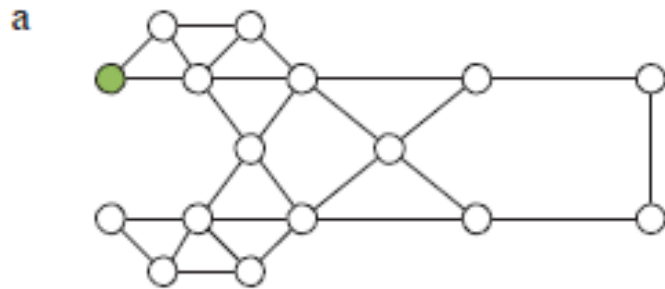
Greedy Sequential

Greedy Sequential

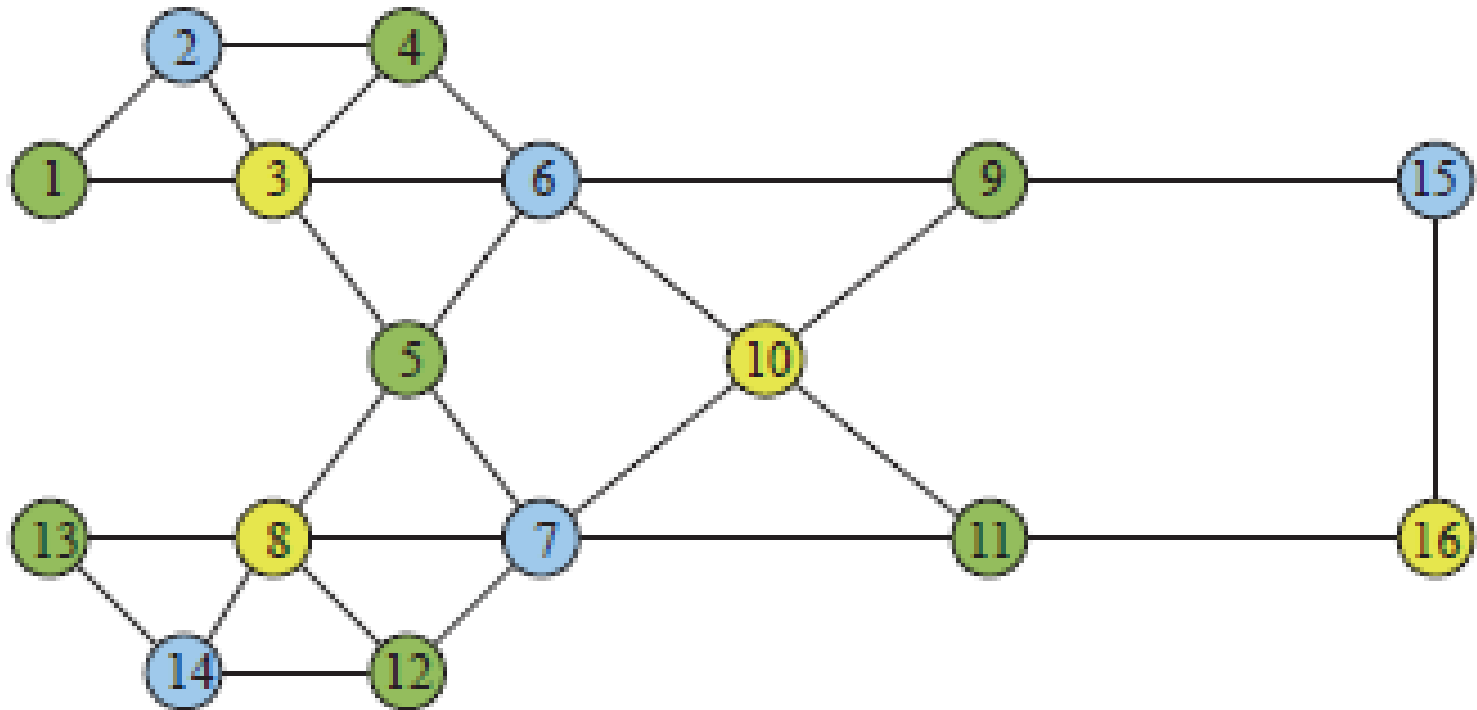
- 1: while \exists uncolored vertex v do
 - 2: color v with the minimal color (number) that does not conflict with the already colored neighbors
- 3: end while

$T(n) = O(n^2)$ in complete graphs and with $|V| = n$.

Greedy Sequential



Greedy Sequential



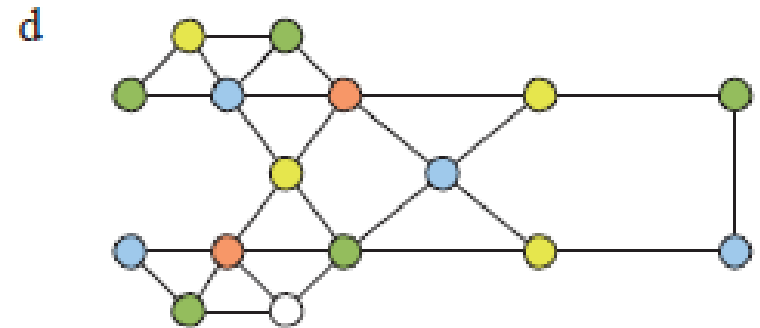
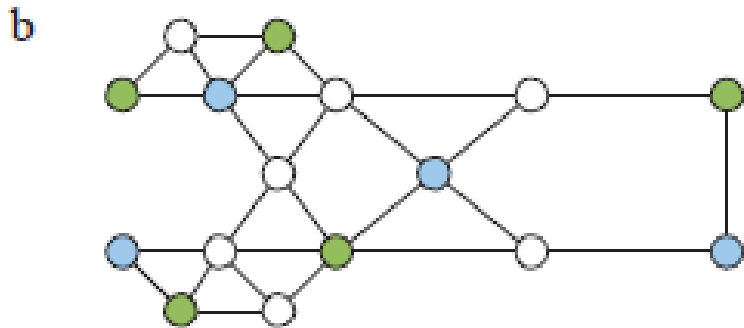
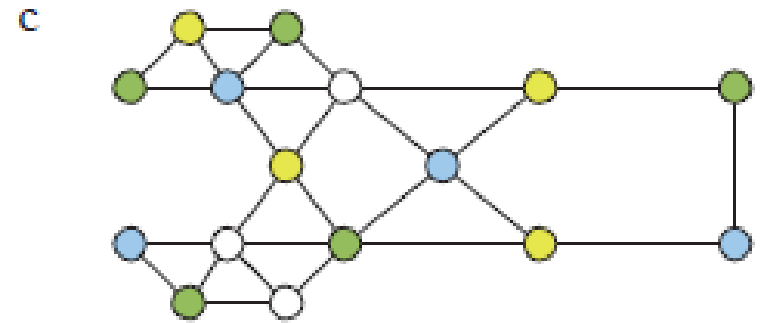
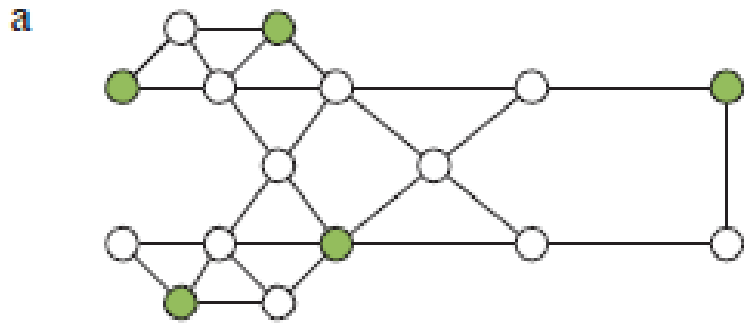
Luby Jones Algorithm

Luby Jones Algorithm

```
color = 0
 $\forall V_i$  assign random numbers
while  $\exists \text{color}(V_i) = 0$  do
  for  $V_i$  do
    for  $V_j$  adjacent to  $V_i$  do
      if  $\text{color}(V_i) = 0$  and  $\text{color}(V_j) = 0$  and  $\text{random}(V_i) > \forall \text{random}(V_j)$ 
         $\text{color}(V_i) = \text{color}$ 
      end if
    end for
  end for
  color++
end while
```

$T(n) = O(n^2)$ in complete graphs and with $|V| = n$.

Luby Jones Algorithm



Parallel Algorithms

Jones Plassman Algorithm

Jones Plassman Algorithm

Assign a random priority to each vertex given by $w(v)$

$U := V$

while ($|U| > 0$) **do**

for all vertices $v \in U$ **do in parallel**

$I := \{v \text{ such that } w(v) > w(u) \forall \text{ neighbors } u \in U\}$

for all vertices $v' \in I$ **do in parallel**

$S := \{\text{colors of all neighbors of } v'\}$

$c(v') := \text{minimum color not in } S$

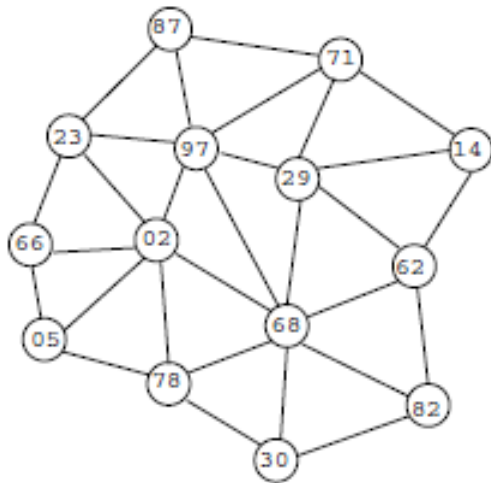
end do

end do

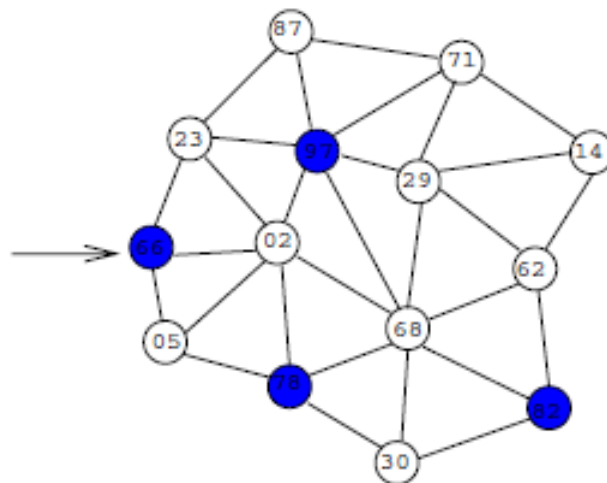
$U := U - I$

end do

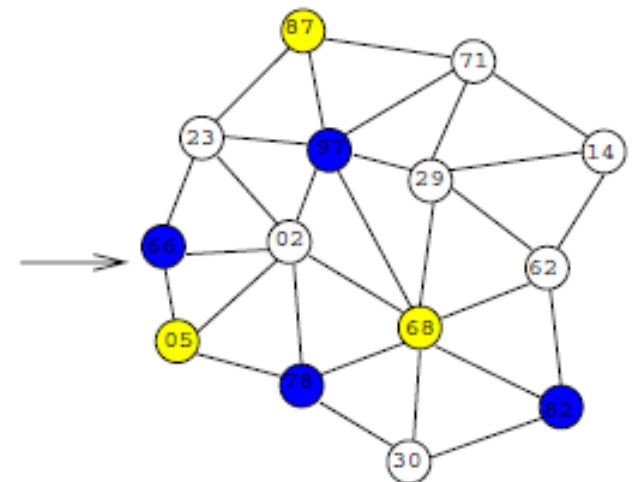
Jones Plassman Algorithm



Initially Assigned
Random Numbers



Each vertex looks at its
neighbors and if it has
the highest number gets
assigned the lowest
available color



Each uncolored vertex looks
at its uncolored neighbors
and gets colored with the
lowest available color if
it has the highest number,
and so on...

Jones Plassman Algorithm Drawback

If the input graph is a chain of vertices and the numbering of vertices correspond to their priorities, then there is no parallelism exhibited by this algorithm

Parallel Multi- Coloring Algorithm

Parallel Multi-Coloring Algorithm

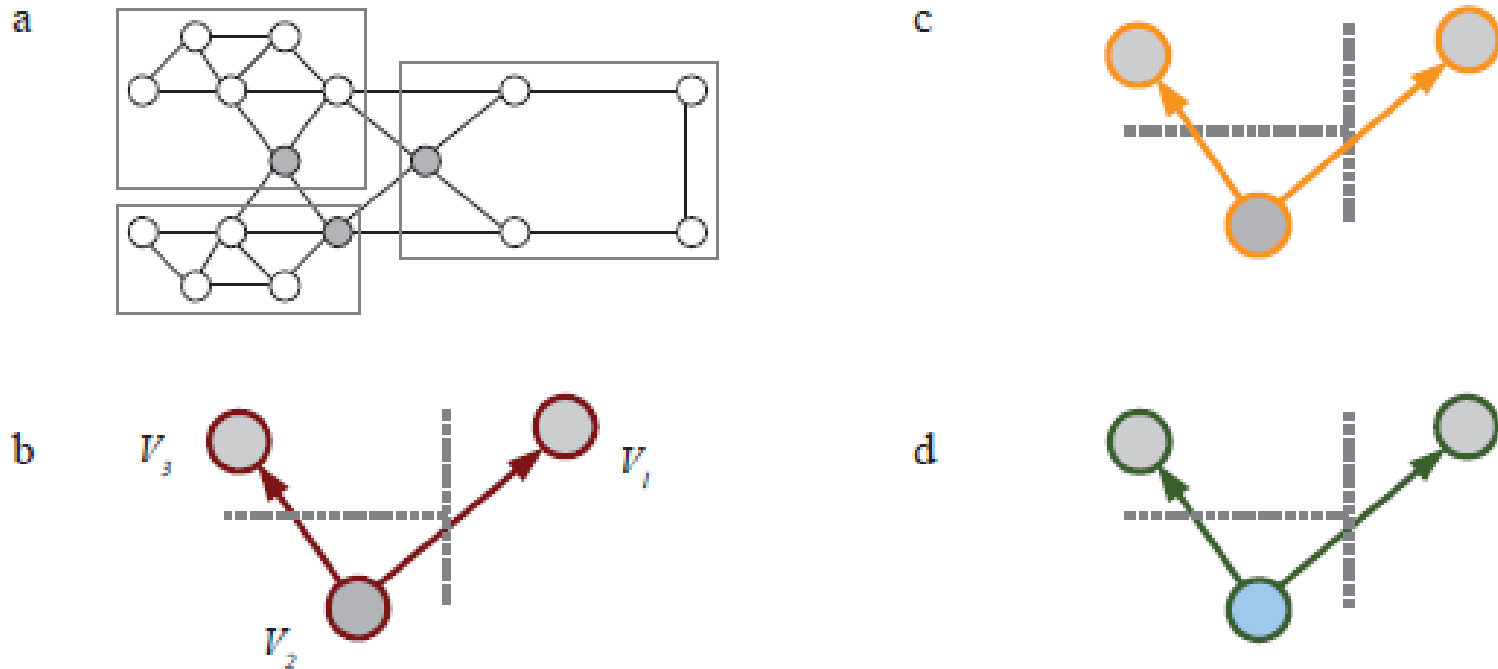


Figure 2.4 The colors depicts: red = lock, orange = check and green = unlock. The dotted line is the interface between the mesh parts. a) A graph is divided. b) Locks are applied on sensitive vertices around V_2 . c) V_2 reads colors in adjacent vertices. d) When V_2 has been colored the locks on adjacent vertices are released.

Parallel Multi-Coloring Algorithm

Initiation

Divide Matrix M intervals of I

for V_i **do in parallel**

$\text{color}(V_i) = 0$

end for

Parallel Multi-Coloring Algorithm

coloring phase

a. invoking

for V_i **do in parallel**

for V_j adjacent to V_i **do**

if $((V_j \text{ adjacent to } V_i) \in D)$

$\text{color}(V_i) = \min\{ m > 0 \mid m \neq \text{color}(V_j), \forall V_j \text{ adjacent to } V_i \}$

else if $((V_j \text{ adjacent to } V_i) \notin D)$

 lock(dependent V_j adjacent to V_i)

 dependent = true

end for

b. thread safe

if(dependent)

for V_j adjacent to V_i **do**

$\text{color}(V_i) = \min\{ m > 0 \mid m \neq \text{color}(V_j), \forall V_j \text{ adjacent to } V_i \}$

 unlock(V_j)

end for

end for

Parallel Multi-Coloring Algorithm

Strength

Weakness

Parallel Multi-Coloring

Low number of colors

Dependent on locks

not much parallelism exploited

