A Backtracking Correction Heuristic for Graph Coloring Algorithms

Sanjukta Bhowmick and Paul Hovland
Argonne National Laboratory
Funded by DOE
Distance-k Coloring

- Assignment of colors to each vertex so that no two distance-k neighbors are assigned the same color (adjacent vertices are distance 1 neighbors)
- Minimum color for distance-k coloring is the \textit{k-chromatic} number

Different Coloring Problems

- Distance 1 (k=1)
- Distance 2 (k=2)
- Partial Distance 2 (k=2 on a subset of vertices)
- Distance 3/2 (k=1 + every path of length 3 uses at least 3 colors)
- Acyclic Coloring (k=1 + every cycle uses at least 3 colors)

Applications

- Compiler Optimization (scheduling use of registers)
- Evaluation of Jacobians (detecting structurally orthogonal columns)
Role of Backtracking Heuristic

¢ Finding the least number of colors is NP-hard

¢ Different graph coloring heuristics
  * Smallest Last, Largest First, Saturation Degree,
  * Incidence Degree, Depth First, etc…

¢ Number of colors is often dependent on the order in which the vertices are colored

¢ Backtracking Heuristic
  * partially changes effects of vertex ordering
  * rearranges the color assignment
  * possibly lead to a lower number of colors
An Example

Minimum Color Required: 3
Backtracking Heuristic

* Set a threshold for expected number of colors
* Backtracking heuristic is invoked when number of colors exceed the threshold
* Only one vertex (*last vertex*) is colored higher than the threshold
* Assign a *pseudo-color* to the last vertex within the threshold
* Determine if there exists alternate color assignment to neighboring vertices; If alternate assignment found
  * Implement alternate assignment and last vertex retains pseudo-color
  * Else, threshold is increased by one
Backtracking Heuristic: An Example

Color Threshold: 3

Blue Conflicts
Green Conflicts
Red Works
Backtracking Heuristic

- **Advantages**
  - Easy to incorporate into other algorithms
  - Can be tuned to user specifications
  - Cost (execution time, memory) of backtracking at each vertex is proportional to its degree

- **Disadvantages**
  - Performance is limited by the top-level algorithm
  - It is based on a local greedy heuristic

*Coloring is as good (often better) than the top level algorithm*
Experiments

¢ A set of six matrices from molecular dynamics application
  l Vertices: 11414
  l Edges: [15K-2683K]

¢ Coloring Problems
  l Distance 1
  l Distance 2

¢ Algorithms Used
  l Natural, Largest First, Smallest Last, Incidence Degree,
    Saturation Degree, Depth First

¢ Correction Algorithms
  l Backtracking
  l Culberson’s Iterative Method

¢ Color Selection Strategy
  l Smallest First
Number of Colors Used by Different D1 Algorithms

- NT: Natural
- LF: Largest First
- SL: Smallest Last
- ID: Incidence Degree
- SD: Saturation Degree
- DF: Depth First
- B: Backtracking
- I: Culberson’s Iteration
Time Taken by Original Algorithms and Correction Methods

![Chart showing time taken by different algorithms and correction methods as a function of matrix size.](image-url)
Extensions to Backtracking

- Backtrack over Multiple Levels
  - Extend search for alternate color assignment to neighbor’s of neighbor’s and beyond.
    - Possibility of finding a better assignment
    - Time and memory requirement increases with levels

- Sorted Backtracking
  - While assigning pseudo-colors, start with the least used color among neighbors
    - Fewer neighbors with pseudo-color---possibility of fewer conflicts
    - Possibility of quickly finding alternate coloring
    - Extra time required for sorting
Number of Colors Used By Different Correction Algorithms

- N: Natural
- L: Largest First
- S: Smallest Last
- I: Incidence Degree
- T: Saturation Degree
- D: Depth First
- B: Backtracking
- Level-5
- S: Sorted Backtracking
- I: Culberson’s Iteration

Number of Colors used by Different Correction Algorithms

- 15K
- 64K
- 130K
- 412K
- 1655K
- 2683K
Number of Colors Used by Different D2 Algorithms

- NT: Natural
- LF: Largest First
- SL: Smallest Last
- ID: Incidence Degree
- SD: Saturation Degree
- DF: Depth First
- B: Backtracking
- I: Culberson's Iteration
Time Taken by Original D2 Algorithms and Correction Methods
Conclusions and Future Work

- **Summary**
  - Backtracking performs well for D1
  - Performance is not as good for D2--more constraints to be satisfied
  - Time taken is not significantly high in either case

- **Future Research**
  - Implementing parallel backtracking
  - A better reordering scheme for D2 (and others?)
  - Analytical results
Acknowledgement

- Backtracking heuristic was inspired by reading a preprint of a review of graph coloring algorithms by Assefaw Grebremdhin, Frederik S. Manne and Alex Pothen

- Thanks to Assefaw Grebremdhin and Rahmi Aksu for use of their graph coloring code. Backtracking heuristic was implemented within this code

- Thanks to DOE for funding this research