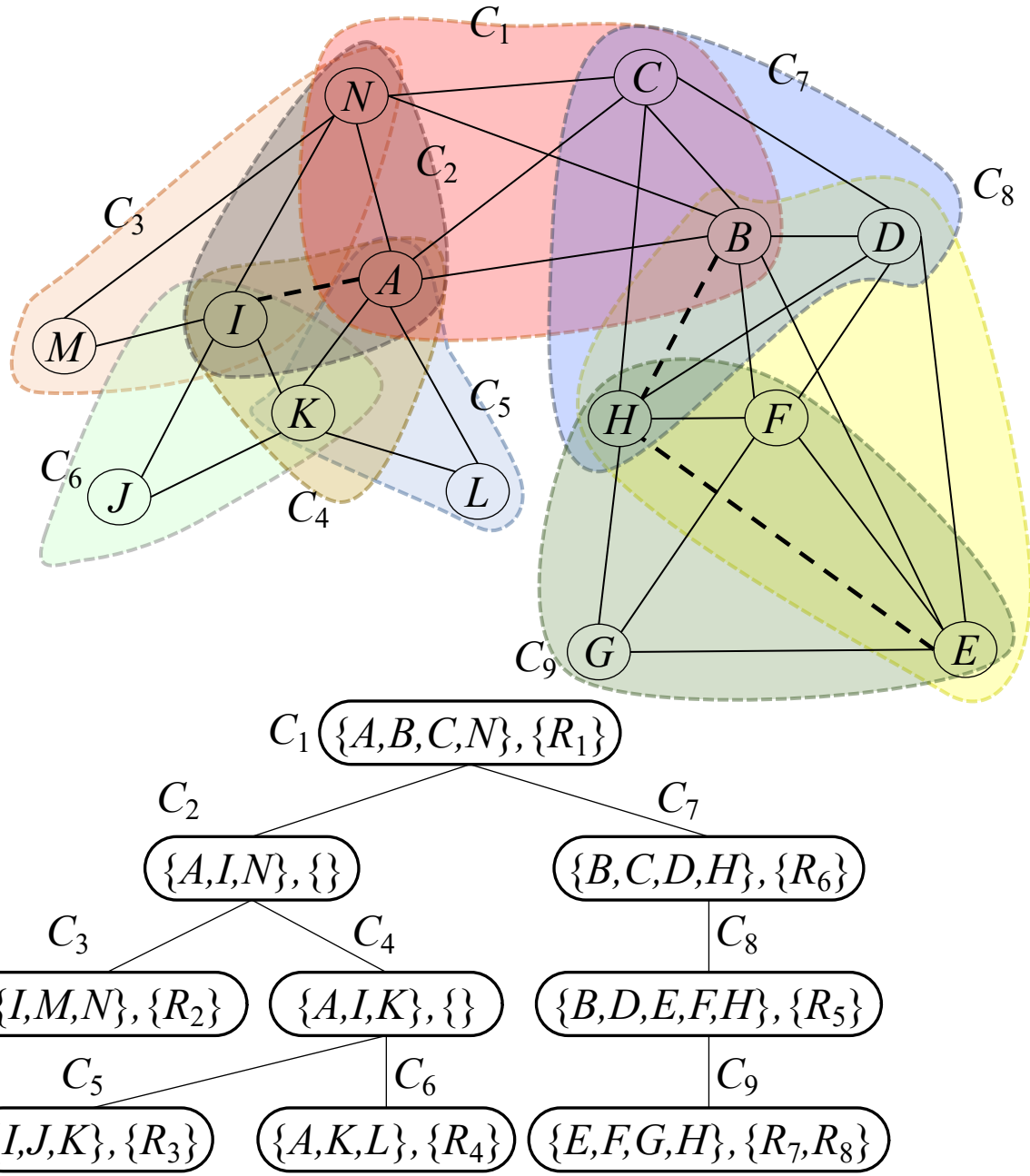
Homework/Mini projects: Four Ideas

April 6, 2020

Berthe Y. Choueiry

1-Tree Decomposition

- Goal: generate a tree decomposition and compute treewidth
- Tasks
  1. Implement the min-fill heuristic to triangulate a graph (report the number of fill-ins)
  2. Implement the Max Cardinality Ordering to get a PEO (or use the PEO provided by the min-fill heuristic)
  3. Implement the Max-Clique Algorithm for computing the maximal cliques (original algorithm available in Golumbic's Perfect Graph)
  4. Using the max-cliques, build the joing tree (Dechter, Fig. 9.4)
  5. Evaluation on binary CSPs, report:
     1. #minfill,
     2. umber of max cliques
     3. size of largest clique
     4. largest number of variables in seperators



2- AllDiff on Sudoku

- Implement GAC on All-Diffs
  - Value graph, maximum matching (one of 2 algorithms(, strongly connected components, depth first search (on oriented graph)
- Propagation across several All-Diffs
  - Keeping the value graphs, augment the maximum matching, iterate until a fix point
- Evaluation: Solve simple Sudoku instances, report CPU time

3- Dom/wdeg

- Implement MAC (strictly stronger than FC) using AC-3. Compare performance on benchmark problems against FC
- Implement MAC using AC-2001. Compare to above and to FC.
- Implement dom/wdeg: every time Revise(.,.) yields a domain wipe-out, update the weight of the constraint
- Evaluate on benchmark problems

4- Binary Branching and Last-Conflict

- Implement search with binary branching
- Implement MAC with AC-3
- Implement Last-Conflict as a dependency directed backatracking
- Evaluate on bechmarl