Solving Delete-Relaxed Planning Tasks by Using Cut Sets

BACHELOR THESIS PRESENTATION

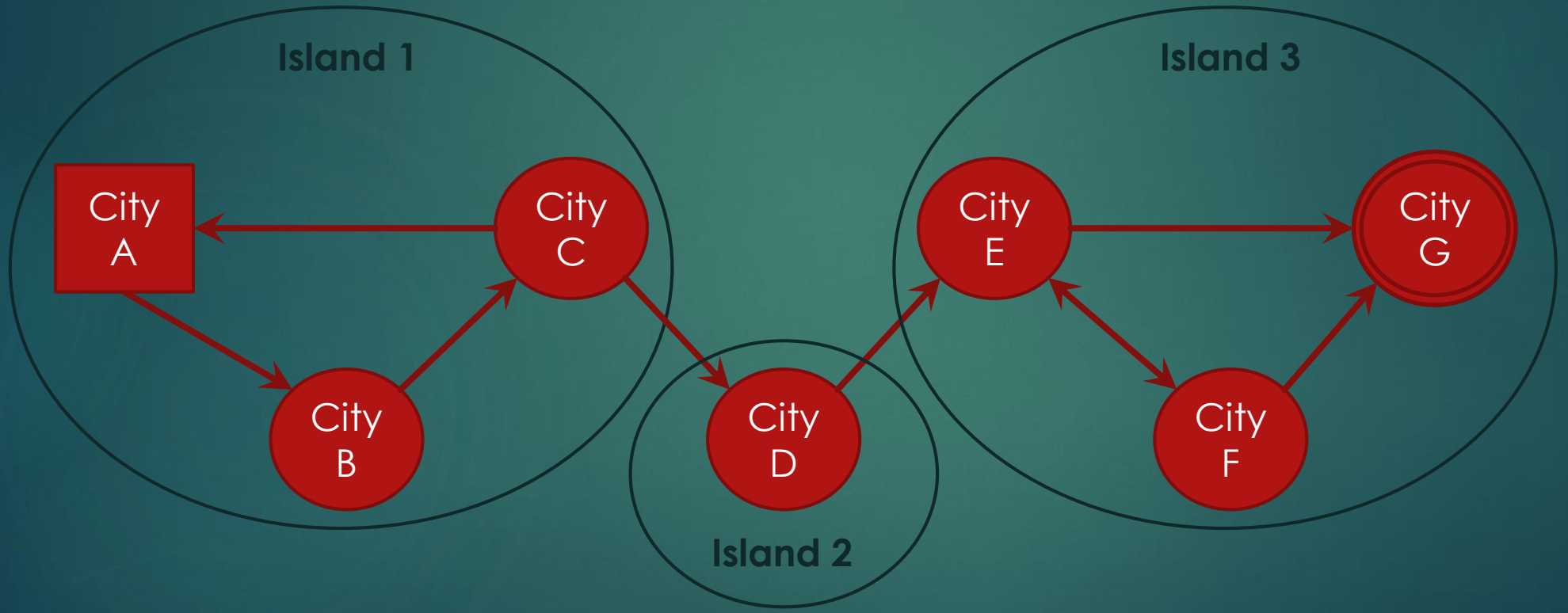
BY MARVIN BUFF

Overview

- Classical Planning
- What is the Problem?
- The Flow-Cut Algorithm
- Experiments /Results
- Conclusion

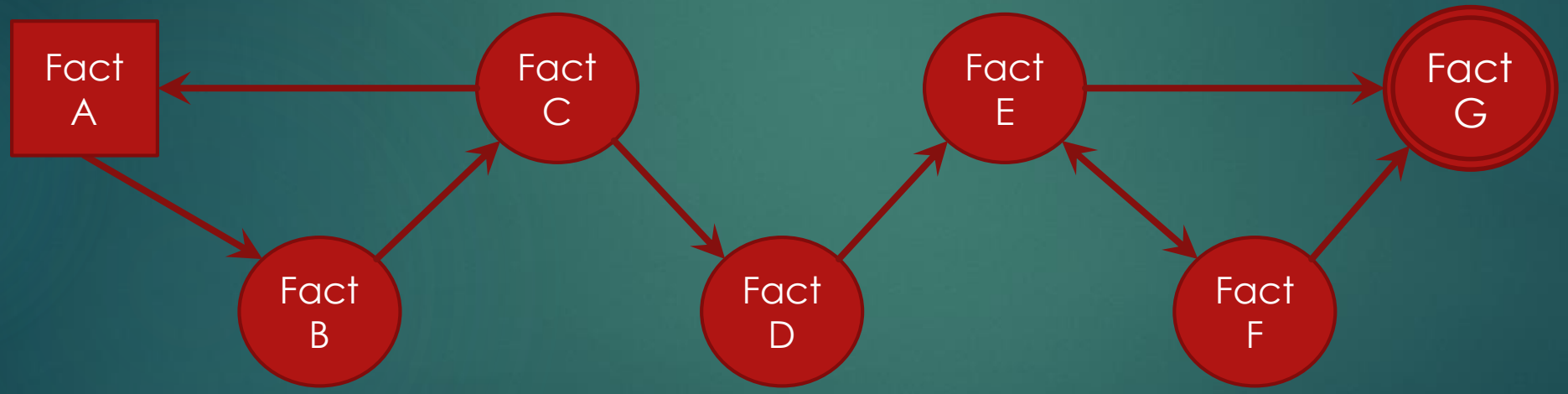
Classical Planning

Island Problem



Classical Planning

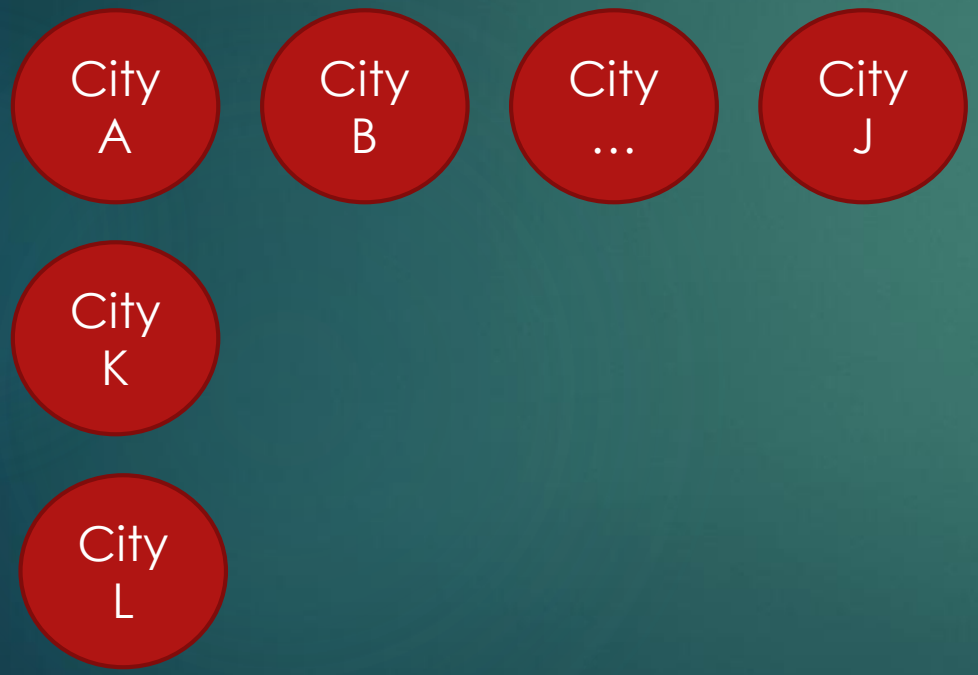
Causal Graph



What is the Problem?

Exponential Growth!

Island Problem – 10 Cities



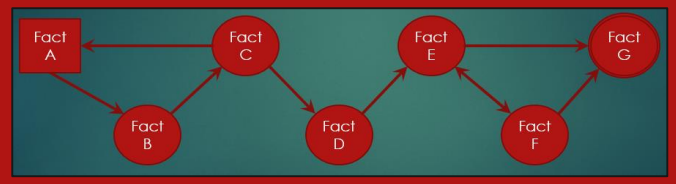
- ~**3 mio** expanded states
- ~**40 mio** expanded states
- ~**500 mio** expanded states

Flow-Cut Algorithm

The Idea

Island Problem

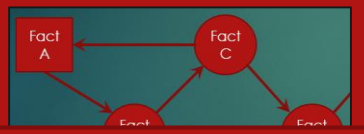
$O(2^n)$



Splitting

Island Problem 1

$O(2^{n/2})$

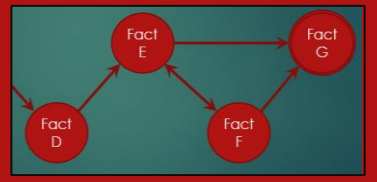


Split Problem

$O(2 * 2^{n/2})$

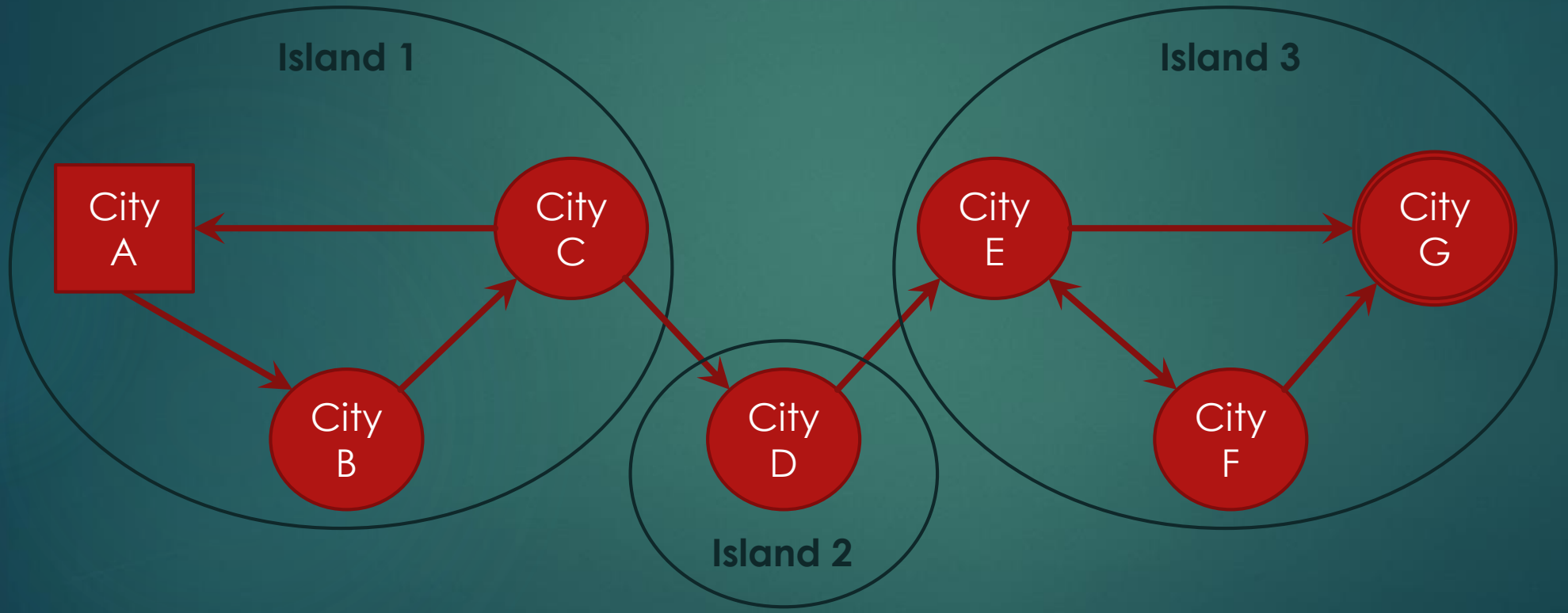
Island Problem 2

$O(2^{n/2})$



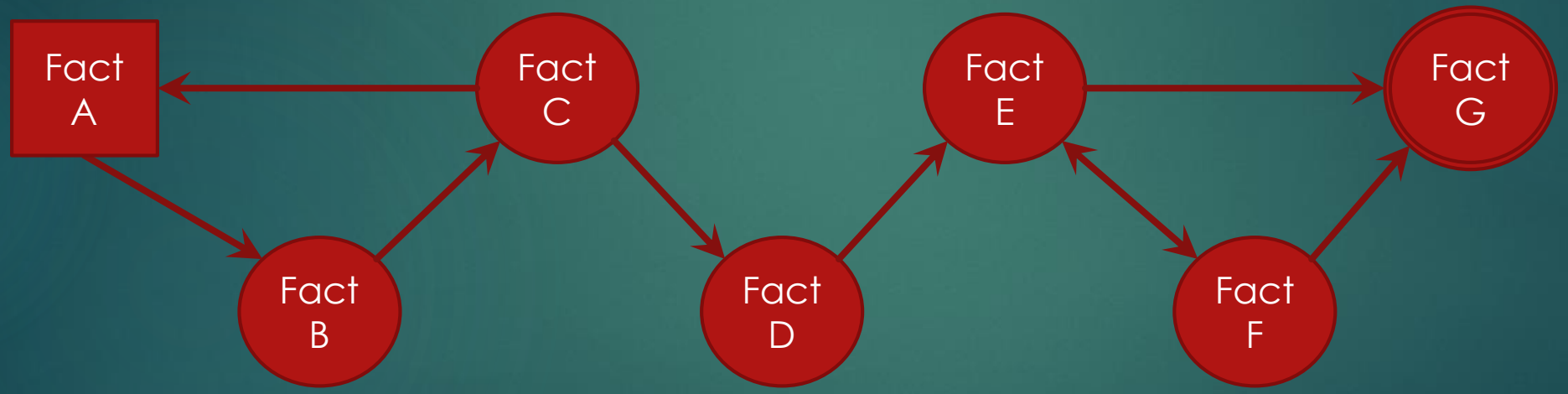
Flow-Cut Algorithm

Given Problem



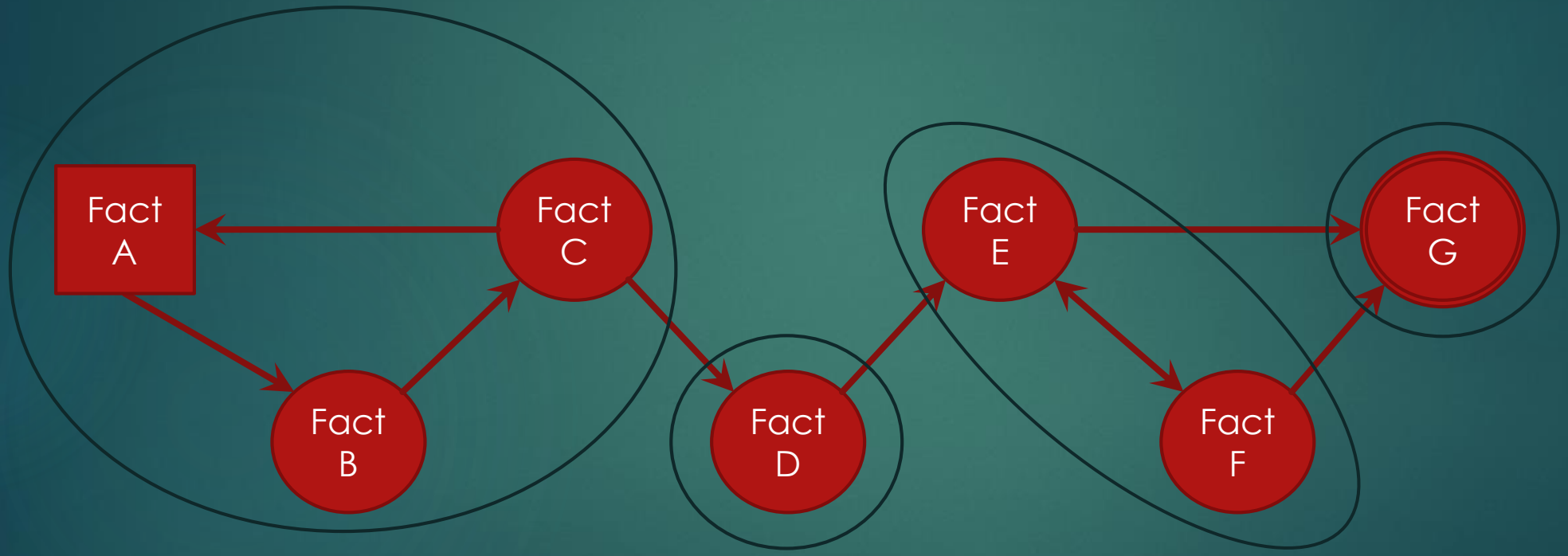
Flow-Cut Algorithm

Derive Causal Graph



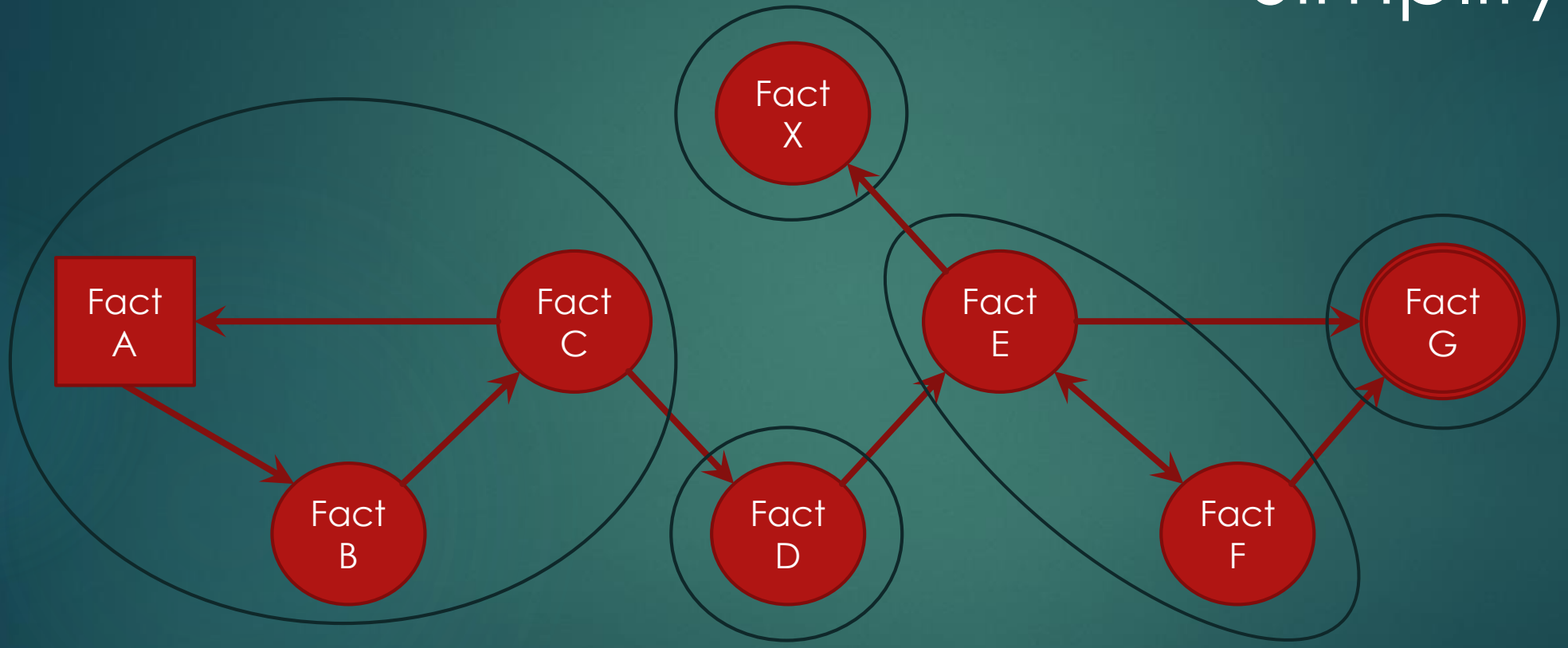
Flow-Cut Algorithm

Determine SCC's



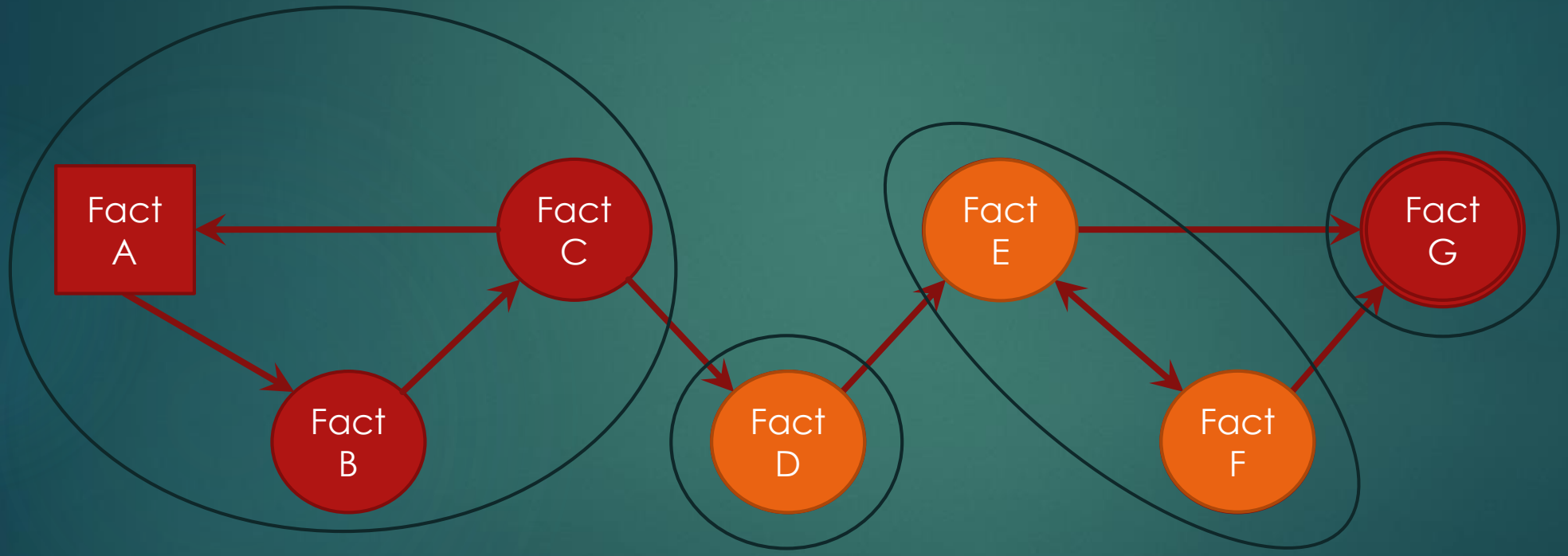
Flow-Cut Algorithm

Simplify



Flow-Cut Algorithm

Initial Cut



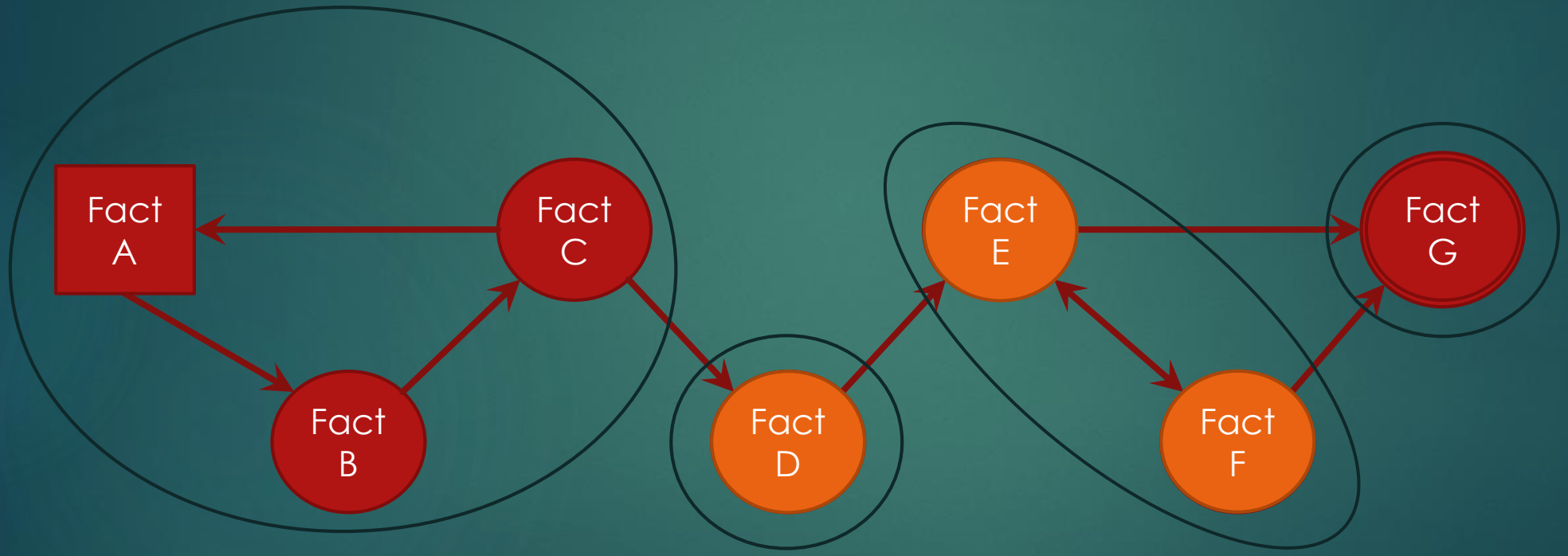
Left SCC

Middle SCCs = Initial Cut

Right SCC

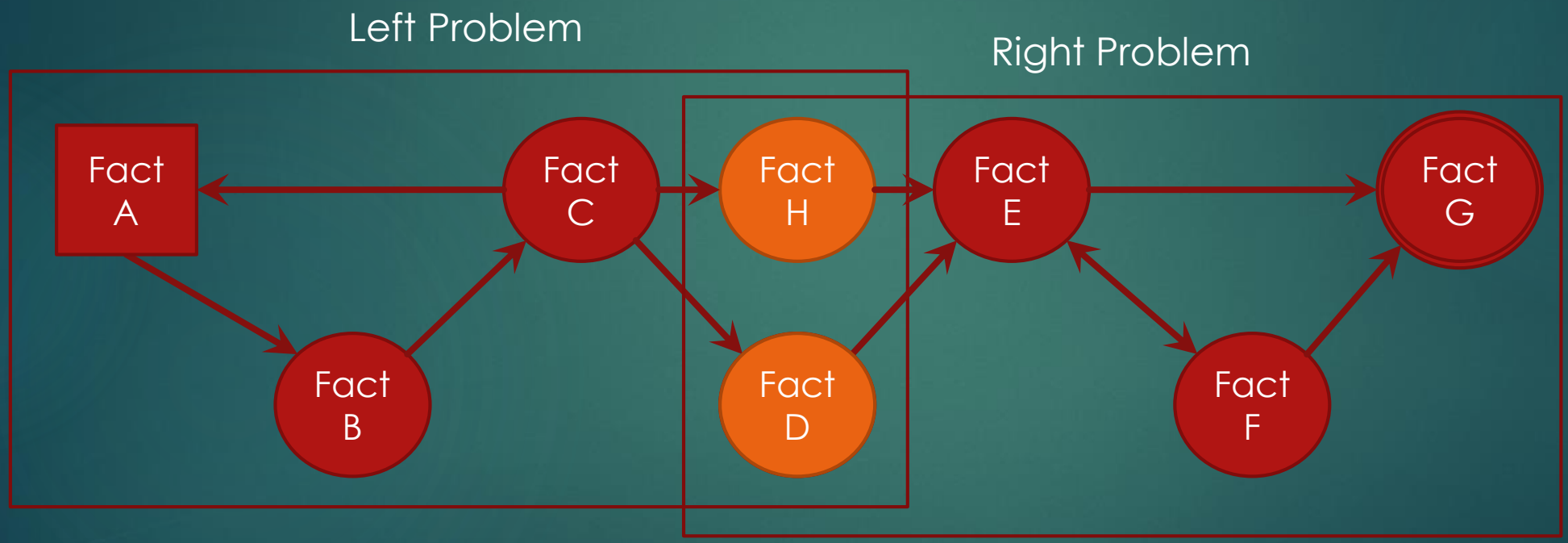
Flow-Cut Algorithm

Minimize Cut



Flow-Cut Algorithm

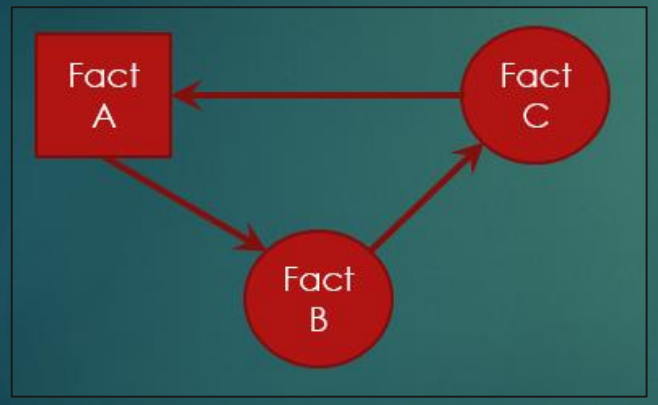
Create Sub-Problems



Flow-Cut Algorithm

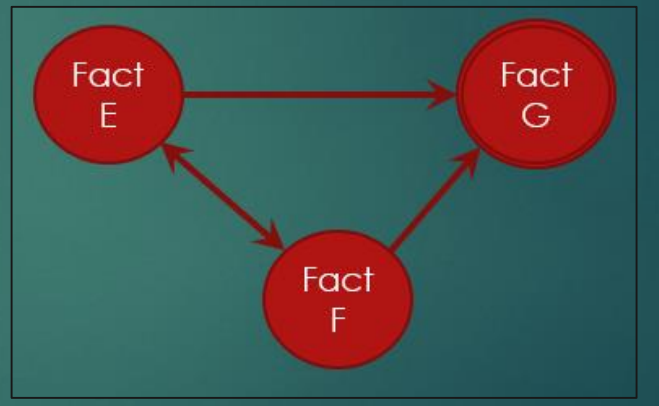
Create Sub-Problems

Left Problem 1



Cut = {}

Right Problem 1



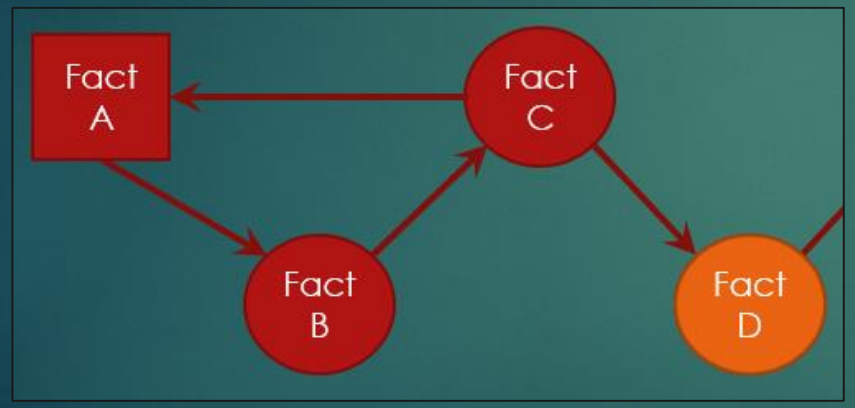
Flow-Cut Algorithm

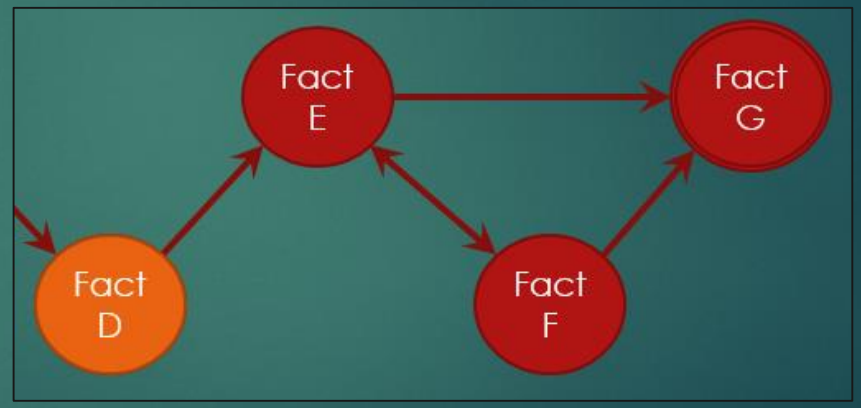
Create Sub-Problems

Left Problem 2

Cut = {D}

Right Problem 2





Solution = 3 + 2

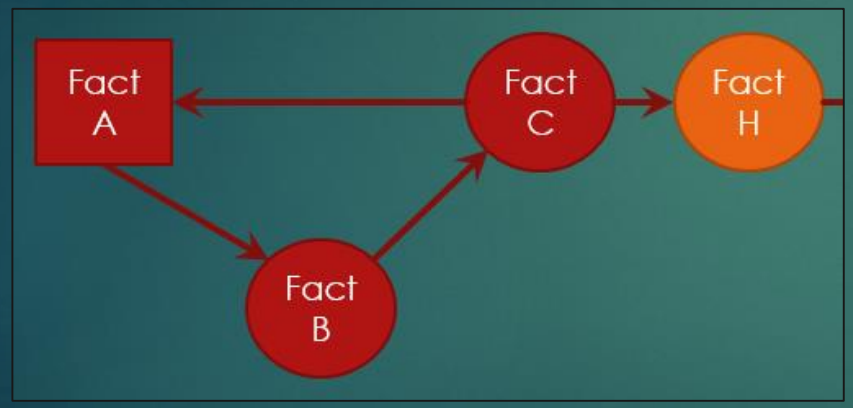
Flow-Cut Algorithm

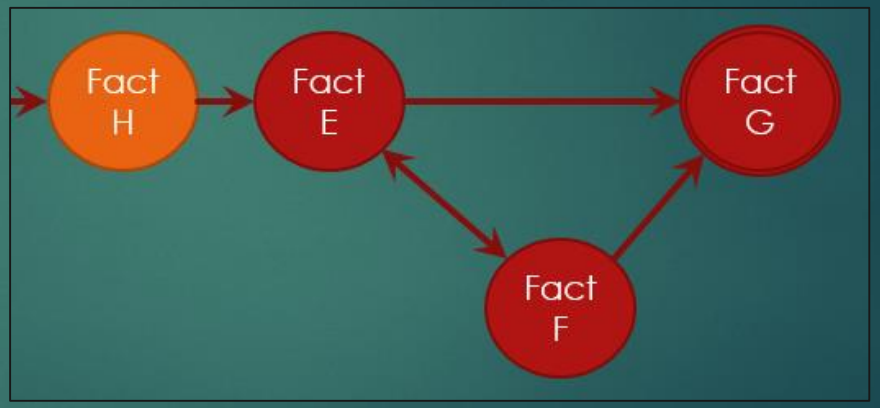
Create Sub-Problems

Left Problem 3

Cut = {H}

Right Problem 3





Solution = 3 + 2

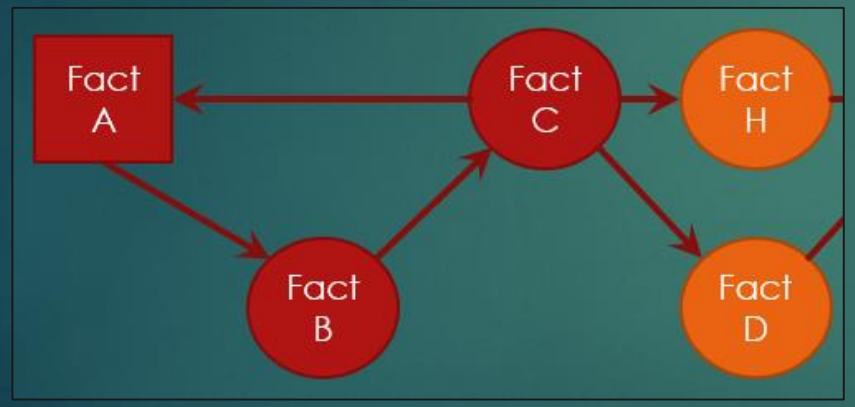
Flow-Cut Algorithm

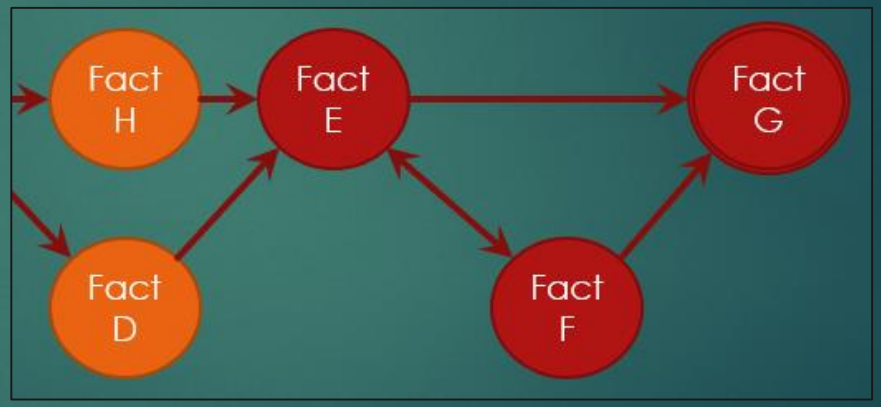
Create Sub-Problems

Left Problem 4

Cut = {D,H}

Right Problem 4





Solution = 4 + 2

Experiments

Setup

- Implemented in C++
- Tested on IPC Benchmark
- Run over 60 minutes per problem

Results

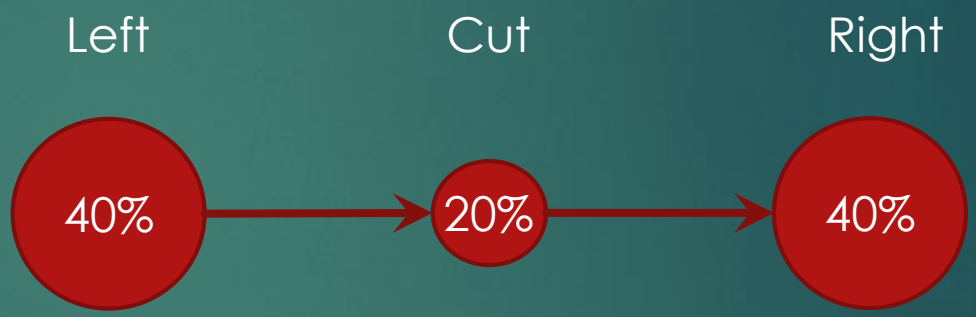
- Type A: Solvable
- Type B: Big Cut
- Type C: No Cut

Experiments

Type A: Solvable

Suitable Domains

- miconic
- pathways
- satellite

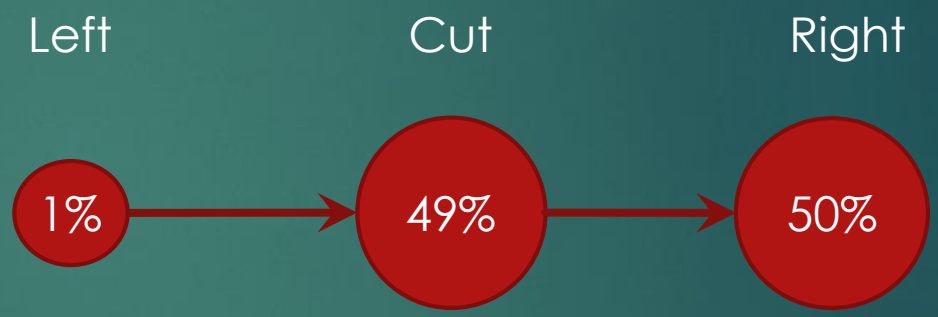


Experiments

Type A: Solvable

Unbalanced Domains

- Mystery
- Rovers
- trucks-strips

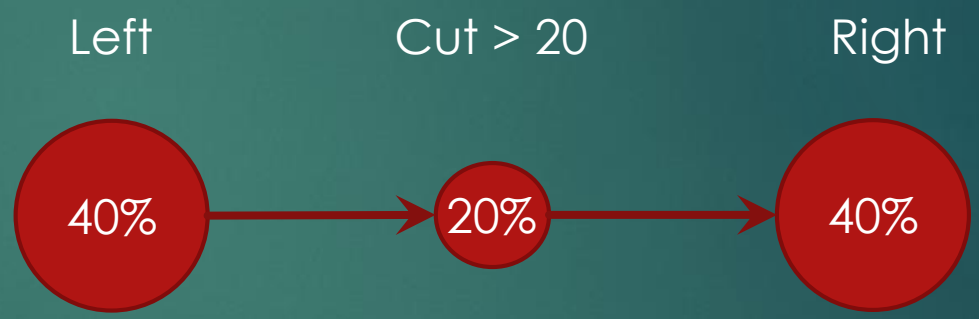


Experiments

Type B: Big Cut

Domains with Big Cut

- childsnack
- no-mystery
- parcprinter
- tidybot

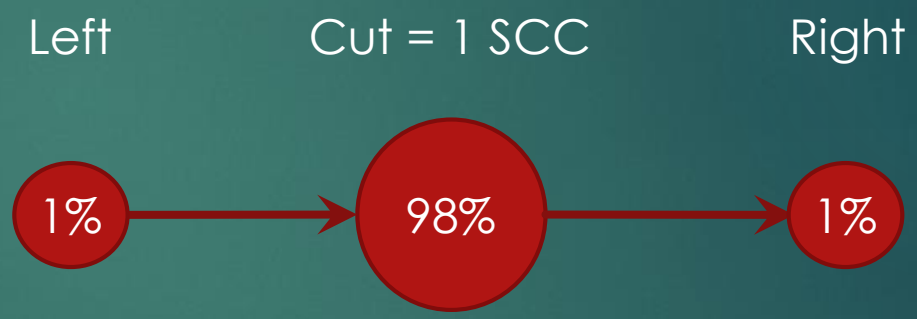


Experiments

Type B: Big Cut

Domains with Big Cut

- childsnack
- no-mystery
- parcprinter
- tidybot

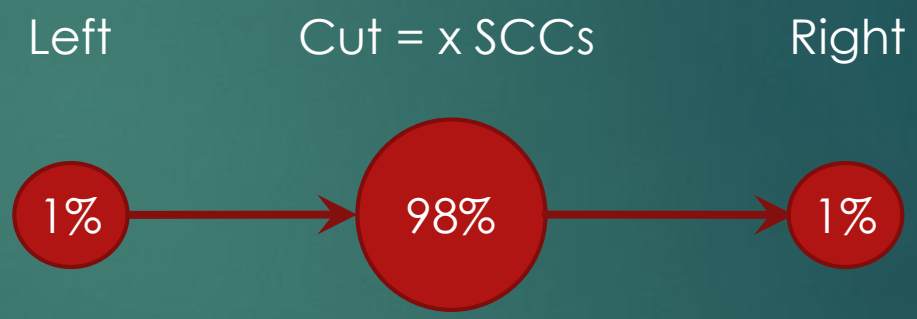


Experiments

Type B: Big Cut

Domains with Big Cut

- childsnack
- no-mystery
- parcprinter
- tidybot

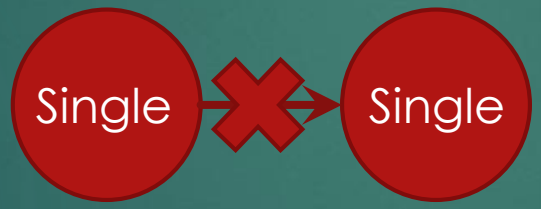


Experiments

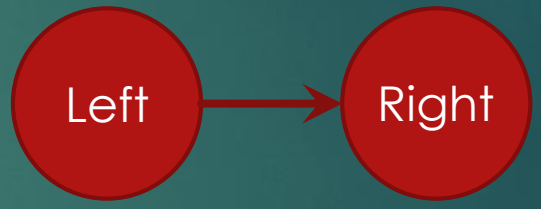
Type C: No Cut

Only One SCCs

Single SCCs



No Middle SCC's



Conclusion

- What did we do?
  - Algorithm to compute h^+
- What could be done different?
  - No SCCs
  - Undirected Structure
  - Multiple Left- /Right-Parts

Questions?