S. Ross

Today we will...

- Discuss graph theory in the context of electric circuits
- Argue that nodal analysis results in an independent set of equations
- Discuss time complexity of problems in this course
- Discuss approximate algorithms for reducing complexity

EECS 40 Spring 2003 Lecture 30	S. Ross	EECS 40 Spring 2003 Lecture 30	S. Ross
Hooray for the CalBot contest winners!			
Best Functionality:			
Nathan Sentz—1st			
Terrance Shih and Rita Chu—2nd			
John Breneman, Ian Robertson and Liona Fan-Chia	ang—3rd		
Kevin Jones—Honorable Mention			
Speed Contest:			
Dan Tocchini and Waley Liang			
Hooray for all the participants and their awesome (if unl	ucky!) bots!		
Hooray for the last day of EE 40!			

## **GRAPH THEORY IN CIRCUIT ANALYSIS**

When I enter a circuit like this into PSpice,



it stores the information "Node 4 and node 5 are connected, with a resistor."

It creates a graph, with elements as branches.



EECS 40 Spring 2003 Lecture 30

S. Ross

EECS 40 Spring 2003 Lecture 30



A **tree** is defined as any set of connections in a graph that connects every node to every other node *without forming any closed paths.* 

Also known as Hamiltonian path!

All connections not in a tree are in the **co-tree** for that tree.



A **cut set** is a minimal set of connections that, when broken, breaks the graph into two completely separate parts.

Minimal means that the cut set cannot contain another cut set as a subset that would break the graph into the same two parts. S. Ross

EECS 40 Spring 2003 Lecture 30

S. Ross

S. Ross

EECS 40 Spring 2003 Lecture 30

# a



Suppose I am given a tree for a graph.

A **fundamental cut set** is a cut set that only contains one branch of the tree.

HOW TO FIND FUNDAMENTAL CUT SETS
 1. Redraw the graph with the tree in a straight line.
 2. For each tree branch,

 a) place the tree branch in a new cut set
 b) see what two node groups are formed by that cut branch
 b) place all links in the co-tree that connect these two groups in that cut set.

#### THE ARGUMENT FOR NODAL ANALYSIS

- Each cut set defines two groups of nodes: 2 supernodes.
- The current flowing out of one supernode must sum to zero.
- Thus, the current in branches of a cut set must sum to zero.
- Suppose we do this KCL for each fundamental cut set.
- Each time we KCL a new fundamental cut set, we introduce a tree branch which is not in any of the other KCL equations.
- So each set of KCL equations is independent!\*
- There are n-1 tree branches, where n is the number of nodes.
- That's how many fundamental cut sets there are to do KCL.
- We get the right number of KCL equations for nodal analysis!

EECS 40 Spring 2003 Lecture 30

#### \*CONFLICTS

For some circuits on paper (not in real life) we can't solve the KCL equations.

If there are the right number of independent equations, how can this happen?

Answer: This graph theory analysis did not address the information that the elements themselves will contribute to the equations.

Sometimes, the elements can give conflicting information (e. g., voltage source in parallel with wire) or redundant information and this leads to no solution.

Remember, this doesn't happen in real life!

EECS 40 Spring 2003 Lecture 30

S Ross

S. Ross





We all know resistors have tolerances, 5%, 10%, etc.

Suppose I want to find the maximum  $V_{OUT}$  possible with given resistor values and tolerances.



A simple calculus proof shows that when  $V_{OUT}$  is maximum, each resistor is at its highest or lowest possible value (inbetween values cannot maximize  $V_{OUT}$ ).

So we could just try each possible combination to find max: 6 resistors, each has 2 possible values: 2<sup>6</sup> combinations.

S. Ross

EECS 40 Spring 2003 Lecture 30

If we tried to do this maximization for a circuit with n resistors, we would have 2<sup>n</sup> resistor combinations to test.

Not good if n is even somewhat large!

This problem has exponential time complexity.

The number of computations it takes to solve this problem is exponential in the size of the problem, n.

Prof. Ross's research: Are there ways around this, such as approximate algorithms or probabilistic algorithms?

EECS 40 Spring 2003 Lecture 30

## **CIRCUIT SATISFIABILITY**

Here is another problem with exponential time complexity.

Suppose we have a Boolean function in n variables:

F = AB + BCD + ADE + BE (Here, n = 5)

```
Is there some combination of inputs that will make F = 1?
```

We could test all of the possible inputs: 2<sup>n</sup> of them.

Again, exponential complexity!

S. Ross

EECS 40 Spring 2003 Lecture 30

## THE CLASS NP

If someone gives you a **certificate** that a Boolean function produces a 1, namely, the inputs that make F = 1, then it is easy to **verify** their claim: it takes just one computation of a digital circuit.

So an answer to this problem can be **verified in polynomial time**, but we need **exponential time to decide** the problem from scratch.

This describes the famous class of problems known as NP.

Think of NP as the description of a good exam:

- Exponentially difficult for the student to solve
- Polynomially easy for the instructor to check

EECS 40 Spring 2003 Lecture 30

## **APPROXIMATE ALGORITHMS**

The problem of finding a Hamiltonian path (tree) is also in NP.

So it could be computationally challenging to come up with the tree that leads to the fundamental cut sets that lead to the KCL in our circuit analysis.

What if I just went from node to node in numerical order,

