

Modeling Circuits with Multiple Grounded Sources: An Efficient Clustering Algorithm

Juan J. Flores

Jaime Cerda

`juanf@zeus.ccu.umich.mx` `jcerda@zeus.ccu.umich.mx`

Facultad de Ingeniería Eléctrica

Universidad Michoacana de San Nicolás de Hidalgo

Morelia, Mich., Mexico

Abstract

Modeling a system is the first step in reasoning about physical devices. By restricting our domain to linear circuits, we can find an efficient algorithm to do this modeling.

The algorithm presented in this article is an efficient implementation of the star-mesh reductions used in Electrical Engineering. By choosing the right representation and based on simple data structures, we can reduce considerably the process of modeling a circuit.

The algorithm has three main sources of efficiency gain: An efficient cluster representation reduces the complexity of the produced model; a simple data structure reduces the search for parallel regions; in the last step, we generate a circuit model where the principle of superposition does not need to be applied. Those three points reduce dramatically both, the complexity of the modeling process and the size of the model. The reduction in the size of the model impacts favorably its use in any reasoning task to be performed. Finally, avoiding superposition will allow us to treat this class of circuits more efficiently.

1 Introduction

One of the main objectives of qualitative reasoning is to derive the behavior of a system from a description of its components and their interrelationships [de 84, For82, Kui85, Wil84]. Prediction of behavior of electric circuits has been achieved by electrical engineering at different levels. There are a number of numerical methods to analyze circuits of different kinds and under different conditions [Ker77, Wal87]. Those methods take as input a circuit topology and exact values for the parameters, perform some computation (mainly based on linear algebra or iterative methods to solve non-linear or differential equations), and return exact values for the variables representing the unknown quantities. In this process, causality and explanation are discarded; the only goals are precision and efficiency.

Several works have been developed to model and reason about circuits of different nature and in different application areas. Those works show that we need to achieve a more efficient implementation. There are several steps in the reasoning process where we can work towards more efficient algorithms and implementations. One such step is the modeling process.

In this paper we are presenting an efficient algorithm to model linear circuits with multiple grounded sources (i.e. all sources are connected to a reference node). Our algorithm is efficient in the time needed to produce the model and in the size of the resulting model. The size of the resulting model is important, since it will impact the reasoning tasks that follow, specially constraint propagation, which has been shown to be intractable [Dav84].

In section 2, we give the basic background that supports our algorithm. Section 3 outlines the proposed algorithms. Section 4 presents an application example. Section 5 presents some of the related work in this area. Section 6 concludes the work presenting our main contributions and discusses some possible extensions for future research.

2 Background

In this section we give some definitions to establish the proposed methodology. After that, we describe the star-mesh reduction [She47], then we show that every Circuit with Multiple Grounded Sources (MGSC) can be reduced to an equivalent circuit where all nodes are connected to sources. Finally, we show that superposition is redundant in this class of circuits.

2.1 Definitions

Let us define Kn as a complete graph (i.e. a complete mesh where every node is connected to every other node). A MGSC, can be represented by $C = (N, V, E, S, g)$, where

N	the set of nodes
$g \in N$	the grounded node
$V \subseteq (N \setminus \{g\})$	the source nodes,
E	the set of elements
S	the set of sources

Also, we define the following predicates:

$c(e, n)$	$\left\{ \begin{array}{l} \text{The element } e \text{ is connected} \\ \text{to node } n \end{array} \right.$
$b(e, m, n)$	$\left\{ \begin{array}{l} \text{The element } e \text{ is connected} \\ \text{between nodes } m, n \end{array} \right.$

Finally, Equations 1 define the functions: $nb(n)$ – the neighbors of node n ; $Kn(n)$ – a complete graph amongst n 's neighbors; and $PKn(m, n)$ – a partial mesh formed by m and n 's neighbors, not including m .

$$\begin{aligned}
nb(n) &\equiv \{m \in N \setminus \{n\} : b(e, n, m) \wedge e \in E\} \\
Kn(n) &\equiv \{e \in E : b(e, p, q) \wedge p, q \in nb(n)\} \\
PKn(m, n) &\equiv \{e \in E : p \in nb(n) \setminus \{m\} \wedge b(e, m, p)\}
\end{aligned} \tag{1}$$

Formally, MGSC circuits satisfy the next properties:

$$\begin{aligned}
&(\forall v \in V : (\exists s \in S : b(s, v, g))) \\
&(\forall e \in E : b(e, n, m) \wedge n, m \in N \wedge n \neq m)
\end{aligned}$$

So we can say that $V \cup \{g\}$, represents the active nodes (i.e. nodes connected to sources) and the set $N \setminus (V \cup \{g\})$ represents the passive nodes (i.e. nodes not connected to sources).

2.2 Generalized Star-Mesh Reduction

Star-Mesh reductions [She47] allow us to eliminate a node from the network, creating an equivalent circuit represented by a Kn that connects the neighbors of the eliminated node. See Figure 1.

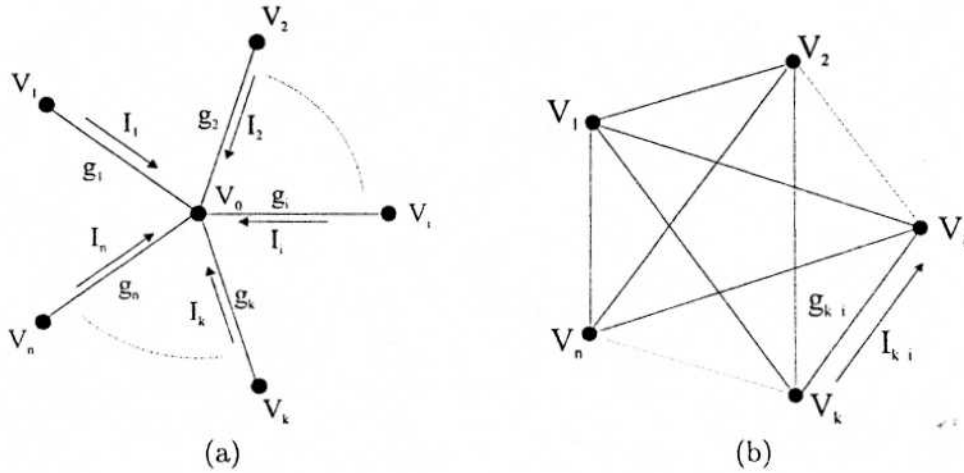


Figure 1: Star-Mesh Conversion

Note that the series reduction is a particular case when $n = 2$. When $n = 3$ we get the well known $Y \rightarrow \Delta$ transformation. For the general case, we get a K-mesh, with $\binom{n}{2} = \frac{n(n-1)}{2}$ elements.

Now we derive an expression for the admittance of the elements of the mesh as a function of the admittances of the elements of the star. Suppose we want to eliminate node 0 from figure 1 (a), and get an electrically equivalent circuit, of figure 1 (b), the rest of the circuit will not notice that the star has been replaced by a K-mesh. Let us define nodes i, k such that $i, k \in nb(0)$

By applying Ohm's law to each element of the star, we have:

$$I_k = (V_k - V_0)g_k \quad (2)$$

Now, applying Kirchoff's Current Law at node 0, we find

$$\sum I_i = \sum (V_i - V_0)g_i = 0 \quad (3)$$

This leads to write V_0 as

$$V_0 = \frac{\sum V_i g_i}{\sum g_i} \quad (4)$$

Substituting 4 into the right member of 2, we have

$$I_k = (V_k - \frac{\sum V_i g_i}{\sum g_i})g_k \quad (5)$$

We can write 5 as:

$$I_k = \frac{\sum (V_k - V_i)g_k g_i}{\sum g_i} \quad (6)$$

If we make $V_{ki} = V_k - V_i$ in 6, we have

$$I_k = \frac{\sum V_{ki} g_k g_i}{\sum g_i} \quad (7)$$

Now, to be electrically equivalent with the original star, we have to impose the following restriction to our reduced circuit:

$$I_k = \sum I_{ki} \quad (8)$$

This means that the sum of the currents of the newly generated elements must be the same as the current of the element eliminated from the star. Comparing 7 with 8, we find that they are equivalent, so we can write

$$I_{ki} = \frac{V_{ki} g_k g_i}{\sum g_j} \quad (9)$$

Finally, we deduce an expression for the admittance between the neighbors of the eliminated node.

$$g_{ki} = \frac{I_{ki}}{V_{ki}} = \frac{g_k g_i}{\sum g_j} \quad (10)$$

2.3 Reduction of MGSC

Let us consider a special class of circuits whose property is that $N = V \cup \{g\}$ (i.e. all nodes are connected to source nodes). The class of circuits that satisfies this property, will be called the reduced MGSC (**rMGSC**) class and the circuits that belong to this class will be called rMGSC circuits. We are ready to show that every circuit with N nodes and V nodes connected to grounded sources, can be reduced to a rMGSC circuit. This demonstration will be done by induction.

Proposition 1 *Every circuit $C = (N, V, E, S, g)$, $N \supseteq V$, can be reduced to a circuit of the rMGSC class.*

Proof.

BASIS. We have the circuit $C = (N, V, E, S, g)$, where $V = N \setminus \{g\}$, in this case the circuit satisfies the property of the rMGSC class per se, and is interconnected by at most a K -mesh among $V \cup \{g\}$.

INDUCTION. Suppose that a circuit $C = (N, V, E, S, g)$ where $N \supset (V \cup \{g\})$ can be reduced to a member of the rMGSC class. We will show that an augmented circuit $C' = (N \cup \{a\}, V, E \cup Y, S, g)$, where $\forall y \in Y : c(y, a)$, can be reduced to a rMGSC class circuit.

By applying the star-mesh reduction at node a , we are getting a reduced circuit with N nodes. The star formed by the elements of the set Y that were connected to this node, become a $Kn(a) \subseteq N$. This change does not affect the rMGSC reducibility property, and its effect is just to make a stronger connectivity in C . Finally, because of the induction hypothesis, C was a rMGSC reducible circuit and therefore so is circuit C' .

2.4 Superposition in the rMGSC Class.

Basically, superposition means that if the network contains several sources of excitation, it is possible to consider the effect of each source separately, independent of the others. At the end you add the partial solutions.

In previous research work[Flo97, Mau98], superposition has been applied to a circuit with s sources producing s models of the circuit, each with only one active source. This approach has several drawbacks. First, you need to produce s models, one for each source. Second, each time analysis is needed, we need to solve s circuits. Third, adding the partial results, adds s operations for each variable. An important detail is that each partial model is based on a totally different topology; this fact makes some reasoning tasks harder to implement.

With all these points in mind, we developed an algorithm to produce a single circuit reduction, where at the end, the effects of all sources on a given element can be computed directly, without the need of any further decomposition.

Superposition is an important property of linear networks, but as we will show it is not necessary in the rMGSC class.. Let us analyze what happens when superposition is applied to rMGSC circuits. Consider the example of figure 2, where $N = \{0, 1, 2, 3\}$, $V = \{1, 2, 3\}$, $S = \{S_1, S_2, S_3\}$, $E = \{g_{12}, g_{13}, g_{23}, g_{04}\}$ and $g = 0$.

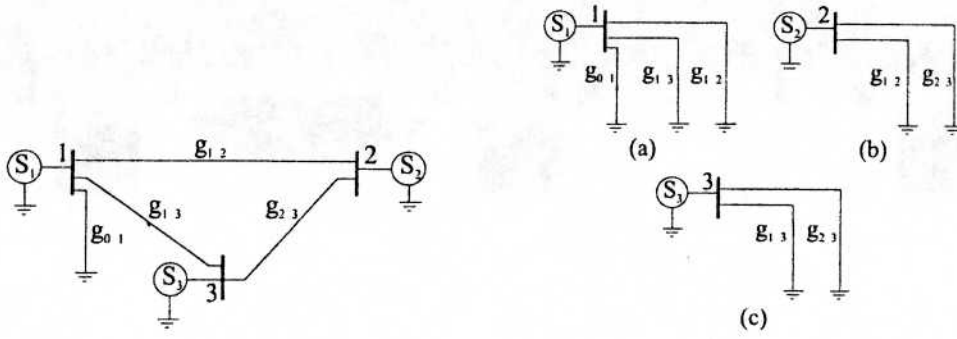


Figure 2: Example of Superposition

If we set S_1 , as the only active source and the rest of the sources are set to zero (i.e. voltage sources are in short circuit). The only elements of the circuit affected by the source are those connected to node 1, so we can picture this as a circuit with just one source, the topology of the resulting circuit is just a parallel array of elements. Figure 2 (a), (b), and (c) show how the original circuit is transformed when each of the sources becomes the only active source.

Finally, we find that in this class of circuits, the current in all of the elements can be determined by applying Ohm's Law, taking into account only the sources connected to it. The sources not connected to the element do not have any effect on it. So no decomposition of the circuits is needed; a single circuit model suffices to produce the solution. This property is the one we were looking for, since our method reduce any MGSC into a rMGSC circuit.

3 The Reduction Algorithm

In this section, we outline the main characteristics of our algorithm to reduce a MGSC to a rMGSC. The algorithm is based on a few key ideas: the elements are represented as admittances; parallel regions are represented as multiple clusters; reduction of parallel regions are detected at the time of inserting elements, and successive star mesh reductions are applied until only active nodes remain in the circuit.

3.1 Representation Choices

This algorithm represents the components of the set E as admittances. Each stage of the reduction we will perform two steps:

- 1.- Elimination of node n : Basically, this corresponds to variable elimination in the Gauss algorithm, and as deduced in 10, we need to calculate the summation of $nb(n)$, if we did not have used this representation, we would have to deal with a summation of inverses.

- 2.- Finding parallels: In admittance representation, parallel reduction can be performed by an addition; in impedance representation it is the quotient of the product and the addition of the involved elements. So we also save operations in this case.

Those are the reasons that lead us to choosing admittance over impedance in our representation.

The second choice was to represent several elements connected in parallel in a single cluster. This choice saves computational resources; for a parallel region with n elements, our representation needs $n + 1$ nodes, whereas a binary representation uses $2n - 1$.

3.2 Replacing a $PKn(n)$

When eliminating node n , we compare node $k \in nb(n) \setminus \{n\}$, against each element of $nb(m)$. We can do one of the following actions, based upon the value of k .

- $k = n \rightarrow$ Eliminate element
- $k \in nb(m) \rightarrow$ Parallel update
- $k \notin nb(m) \rightarrow$ Insert element

In one pass, we eliminate the elements of the star and insert the elements of the $PKn(n)$ to produce the equivalent circuit.

3.3 Reduction Rules

Basically our algorithm uses two types of reductions: parallel and star-mesh. Table 1 shows the algebraic constraints for each one of them.

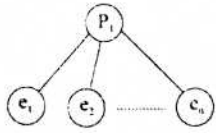
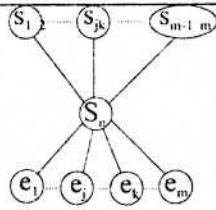
PARALLEL	STAR
	
$1 \leq j \leq n$ $i_j = V_{p1} g_j$ $V_j = V_{p1}$	$i, j \in nb(n)$ $g_n = \sum g_{in}$ $i_{in} = \sum_{k \in PKn(i,n)} i_{ik}$ $V_i = \frac{i_i}{g_i}$ $V_n = \frac{1}{g_n} \sum g_{in} V_i$
$g_{p1} = \sum_{i=1}^n g_i$	$g_{ij} = \frac{g_{in} g_{jn}}{g_n}$

Table 1: Algebraic Constraints

Once we have reduced the circuit to a rMGSC, we use these constraints to produce a constraint-based model of the circuit.

3.4 The Reduction Algorithm

Figure 3 shows the algorithm to reduce a MGSC $C = (N, V, E, S, g)$.

```

rMGSC( $C$ )
   $P = N \setminus (V \cup \{g\})$ 
  While  $P \neq \emptyset$ 
    Node_Min( $P$ )  $\rightarrow n$ 
    Replace_Y_by_Kn( $n$ )
     $P \setminus n \rightarrow P$ 
  Return

Node_Min( $P$ )
  ( $n \in P \wedge (\forall m \in P : |nb(n)| \leq |nb(m)|)$ )
  Return  $n$ 

Replace_Y_by_Kn( $n$ )
  For each  $m \in nb(n)$ 
    Replace_eY_by_PKn( $m, n$ )
  Return

Replace_eY_by_PKn( $m, n$ )
  For each  $k \in nb(n)$ 
    case
       $k = m : nb(m) \setminus n \rightarrow nb(m)$ 
       $k \in nb(m) : \text{Parallel Update}$ 
       $k \notin nb(m) : nb(m) \cup k \rightarrow nb(m)$ 
  Return

```

Figure 3: The reduction algorithm

The function $\text{rMGSC}(C)$, takes a circuit and starts to eliminate all the non-active nodes, by replacing the star associated with each of them by a mesh. $\text{Node_Min}(P)$, extracts the node with the minimal degree from P . $\text{Replace_Y_by_M}(n)$, replaces the star associated with node n by a Kn among its neighbors. Finally, $\text{Replace_eY_by_PKn}(m, n)$ replaces the partial Kn associated with node m when node n is eliminated.

4 An Illustrative Example

Let us consider the circuit $C=(N,V,E,S,g)$ of figure 4 (a), where $N=\{0,1,2,3,4,5,6\}$

$$V=\{1,2,3\}$$

$$E=\{G_1, G_2, \dots, G_{10}\}$$

$$S=\{S_1, S_2, S_3\}$$

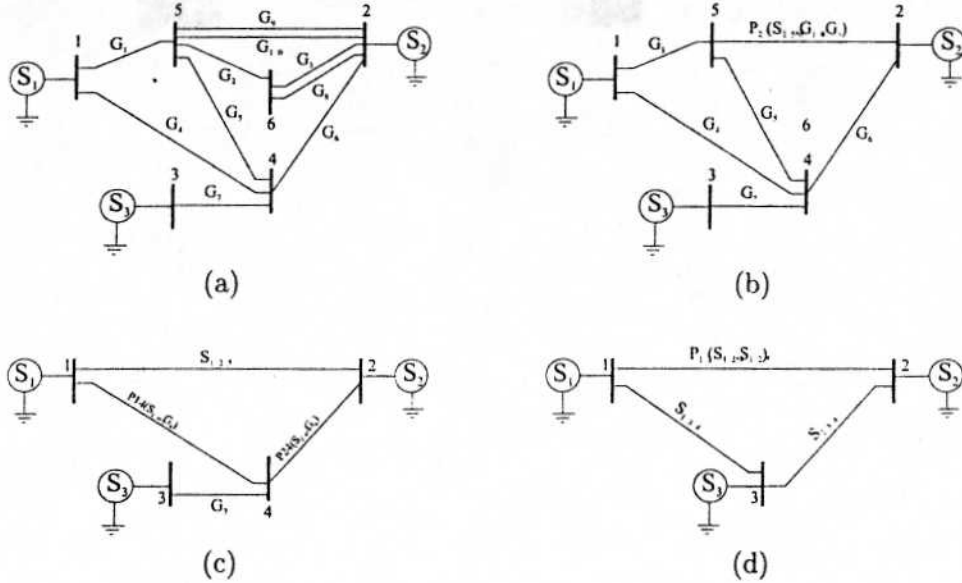


Figure 4: Example

When the nodes are inserted in the system, all parallel regions are immediately detected and made into clusters (first stage not shown in Figure 4).

For this circuit, $P = \{4, 5, 6\}$; we eliminate node 6 since it has the lowest degree. Elimination of node 6 in figure 4 (a) results in the circuit shown in figure 4 (b). P_{2-5} is the parallel cluster between nodes 2 and 5, containing elements S_{2-5-6} , G_{10} , and G_9 . S_{2-5-6} is one of the elements that result from the elimination of node 6 connecting nodes 2 and 5. This step shows how the series reduction is a special case of the star-mesh reduction.

In the next step, $P = \{4, 5\}$, the algorithm takes node 5; the star around it is replaced by $Kn(5)$. The resulting circuit is shown in figure 4 (c).

Finally, $P = \{4\}$, the algorithm takes node 4, as the one to be eliminated. The final reduced circuit is shown in figure 4 (d). As we can see, this circuit satisfies the rMGSC class property.

The reduction algorithm produces a graph containing a history of modifications applied to a given circuit. Figure 5 shows the graph corresponding to the reductions that transform the circuit of figure 4 (a) into the circuit of figure 4 (d). The leaf nodes of the same figure denote admittances. Internal nodes, marked with P denote parallel clusters. A Y node represents a star-mesh reduction, its children are the elements of the star, and its ancestors, S nodes, represent the $Kn(n)$ that replaces the star.

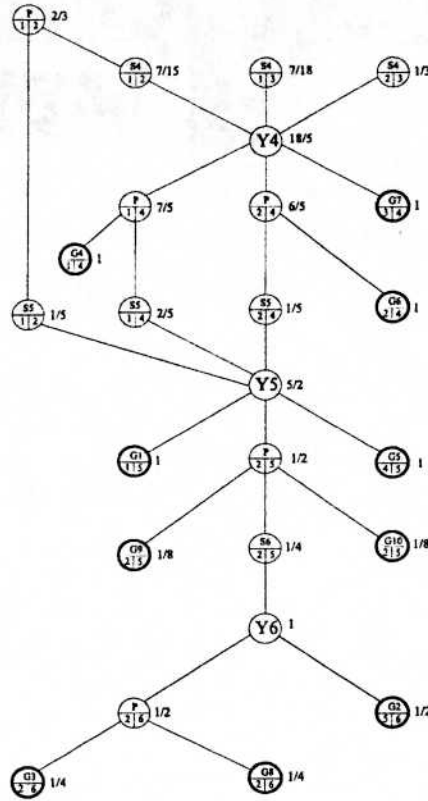


Figure 5: Final Reduction Graph for Circuit of Figure 4

5 Related Work

There have been many successful effort to perform model-based reasoning about circuits [SS77, SS80, de 84, Dav84, Gen84, Ham91, FF96, Flo97, Mau98]. The most recent work and more related to this one are the works of Flores and Farley, and Mauss.

The work of Flores and Farley can only cope with series-parallel reducible circuits, missing an important number of circuits and being of no much use for many practical applications. For instance, to validate a diagnosis the circuit is modified to reflect the fault. We can easily find an example where a short circuit can transform a series-parallel reducible circuit into a one that is not.

Both works, Flores and Farley's, and Mauss', represent parallel clusters as binary relations. This fact increases the size of the resulting graph. Also, they search for elements or clusters that share the same nodes to produce a parallel cluster. That is an expensive search. Our algorithm is based in a data structure that can determine whether there is an element in parallel with the inserted element in almost constant time.

Both works, Flores and Farley's, and Mauss', need to use superposition to analyze circuits with multiple sources in its natural representation. As men-

tioned above, they decompose a circuit with n sources in n sub-models, each with only one active source. This fact makes the production of the model and the analysis time more expensive.

The goal in both works was to reduce the circuit to a single resistance, without thinking that maybe it was not necessary and it would not provide more information, so they were making redundant clustering.

6 Conclusions and Future Work

We are presenting an algorithm capable of modeling circuits with multiple grounded sources. The reduction is accomplished by repeated applications of the star-mesh reduction rule. Our algorithm also gains in efficiency by eliminating the quadratic process to find parallel clusters. With the use of simple data structures, we reduce time in detection of parallel clusters to almost constant time.

By using an adequate representation, our modeling algorithm has proven to be efficient. Compared to Kron's reduction, a classical node-elimination methods used in Electrical Engineering [GS94, And73], our algorithm performs much more efficiently, since it only performs operations when they are necessary. Classical methods, on the other hand, use matrix representations and linear algebra, operating on complete rows and columns, therefore performing unnecessary operations in some cases.

To establish a comparison, let us define n as the number of nodes of the circuit and s the number of sources. Table 6 compares the number of operations needed to get the reduced matrix (without considering the inverse operation ($O((n-s)^3)$)), against the number of operations needed by our algorithm in the worst case (i.e. every node is connected to each other). Table 6 provides a numeric figure, for $n = 6$ and $1 \leq s < n$.

Method	Number of Operations
Kron	$\frac{s^2(1 + (n-s)^2) + s(n-s)(n-s-1)^2}{n^2 + s^2 - ns}$
rMGSC	$\sum_{k=s+1}^n \left[\frac{k(k+1)}{2} - 1 \right]$ $\sum_{k=s+1}^n [k(k-1)]$

Table 2: Number of Operations as a Function of n and s

On the other hand, by using multiple parallels clusters, we have optimized the graph, leading to a more efficient model.

Again, comparing our method to those used by classical circuit solvers, our algorithm produces a cluster graph that can be used to produce a constraint-based model. A constraint-based model can be derived from first principles,

s	Kron	rMGSC
1	256	120
2	332	116
3	288	105
4	184	84
5	80	50

Table 3: A Comparison Case

using the same ideas as in [FF96, Flo97]. Such a constraint-based model can be used to perform several reasoning tasks about the circuit, e.g. first-order reasoning, qualitative analysis, design, diagnosis, etc. (see [Flo97]).

Traditional circuit analysis is not that flexible, and does not provide the intuition behind the cluster graph. Circuit solvers just provide numeric solutions to precisely specified circuits. QR techniques enable us to provide solutions under the presence of incomplete or uncertain information; also, explanation can be generated, etc.

This algorithm was developed in an effort to make the work by Flores and Farley more efficient. Trying to apply QR methods to circuit analysis, design, and diagnosis, we found that efficiency was a crucial point to make a QR system applicable to larger circuits. This situation is present specially in diagnosis, where a fault has to be diagnosed, and a solution proposed in real time, in such a way as to shut-down or isolate part of the system.

Our next target in the process of making qualitative circuit analysis more efficient is constraint propagation. If we want to apply qualitative reasoning to practical problems, we need a constraint propagation algorithm that exploits the topology of the cluster graph.

References

- [And73] P. M. Anderson. *Computer methods in Power System Analysis*. Iowa State University Press, 1973.
- [Dav84] R. Davis. Diagnostic reasoning based on structure and behavior. *Artificial Intelligence*, 24:347–410, 1984.
- [de 84] Johan de Kleer. How circuits work. *Artificial Intelligence*, 24:205–280, 1984.
- [FF96] Juan J. Flores and Art M. Farley. Qualitative phasor analysis. In *Proc. 10th Int. Workshop on Qualitative Reasoning About Physical Systems*, Fallen Leaf Lake, CA, May 1996.
- [Flo97] Juan J. Flores. *Reasoning about Linear Circuits in Sinusoidal Steady State*. PhD thesis, University of Oregon, August 1997.

- [For82] Kenneth D. Forbus. Modeling motion with qualitative process theory. In *Proc. 2nd National Conf. on Artificial Intelligence (AAAI-82)*, pages 205–208, 1982.
- [Gen84] M. R. Genesereth. The use of design descriptions in automated diagnosis. *Artificial Intelligence*, 24:411–436, 1984.
- [GS94] John J. Grainger and William D. Stevenson. *Power System Analysis*. McGraw-Hill, New York, 1994.
- [Ham91] W.C. Hamscher. Modeling digital circuits for troubleshooting. *Artificial Intelligence*, 51:223–271, 1991.
- [Ker77] Robert B. Kerr. *Electrical Network Science*. Prentice-Hall, Englewood Cliffs, NJ, 1977.
- [Kui85] Benjamin J. Kuipers. The limits of qualitative simulation. In *Proc. 9th Int. Joint Conf. on Artificial Intelligence (IJCAI-85)*, pages 128–136, San Mateo, CA, 1985. Morgan Kaufmann.
- [Mau98] Jakob Mauss. Local analysis of linear networks by aggregation of characteristic lines. In *Proc. 9th International Workshop on Principles of Diagnosis (Dx98)*, Cape Cod, Massachusetts, USA, May 1998.
- [She47] D. W. C. Shen. Generalized star and mesh transformations. *Philosophical Magazine and Journal of Science*, 38(7):267–275(2), 1947.
- [SS77] R. Stallman and G. J. Sussman. Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit analysis. *Artificial Intelligence*, 9:135–196, 1977.
- [SS80] G. J. Sussman and G. L. Steele. CONSTRAINTS: a language for expressing almost-hierarchical descriptions. *Artificial Intelligence*, 14:1–39, 1980.
- [Wal87] Alan K. Walton. *Network Analysis and Practice*. Cambridge University Press, Cambridge MA, 1987.
- [Wil84] Brian Carter Williams. Qualitative analysis of MOS circuits. *Artificial Intelligence*, 24:281–346, 1984.