Qualitative Reasoning about Electrical Circuits using Series-Parallel-Star Trees

Jakob Mauss
DAIMLER-BENZ AG, Research F1M/EK
D-70546 Stuttgart, Tel.: +49 711 17 41247, Fax: +49 711 17 41717

Bernd Neumann
Labor für Künstliche Intelligenz (LKI)
Vogt-Koelln-Str. 30, 22527 Hamburg, Germany
E-Mail: {mauss,neumann}@informatik.uni-hamburg.de

Abstract

Qualitative reasoning about electrical circuits is often formalized as constraint propagation. We demonstrate a certain weakness of this reasoning scheme in deriving useful information for certain but not uncommon circuits and argue that the stated problem can be overcome by exploiting global structural information. Reasoning is not primary formalized as constraint propagation here but as a graph analysis task. We present an algorithm to derive and organize structural information based on series-parallel-star reduction of circuits and use the resulting sps tree to guide qualitative-algebraic reasoning about resistances, voltage drops and currents. We prove that the approach works for arbitrary resistive nets. The developed technique may be applied to model-based diagnosis, model aided fault tree development, failure mode and effect analysis or related application fields.

1 Introduction

Qualitative reasoning about electrical circuits or in general about attributed diagrams representing a technical device has many useful applications such as model based diagnosis, model aided fault tree development (cf. (Mauss & Neumann 1995)), or failure mode and effect analysis (FMEA, cf. (Lee & Ormsby 1992), (Struss et al. 1995)).

Reasoning about technical devices is often formalized as constraint propagation. The device model or set of constraints is often derived from a given device diagram by connecting generic component models.

A component model is a set of constraints relating local variables of a component with each other. A connection is an equation relating two variables of the two connected components with each other. Thus, the task of building a device model from given component models and a given topology is quite simple: It consists basically in instantiating predefined component constraints and generating a set of port equations.

However, this naive modeling approach has well known limitations. Consider e.g. the series connection of two resistances given in Figure 1. Propagation of the local component constraints will derive u1 = 0, and u4 = U, and nothing more. In particular, we cannot derive in this way that i0 = i1 = i2 = i3 = i4 = i5 = U/(R1 + R2). To derive this, we would need an additional constraint u4 - u1 = (R1 + R2) i1. Such missing information can be derived by taking equivalent circuits into account, sometimes called „grey boxes", or „slices" in (Sussmann & Steele 1980), by exploiting additional laws (e.g. Kirchhoff's voltage law, the sum of the voltage drops in a loop is zero), or - in some cases - by propagating connectivity information through the circuit (Struss et al. 1995). In any way, to derive the missing information we will have to take the structure of the given circuit into account in the one or the other way. For such a task, a global topology analysing processing scheme seems more natural to us than local constraint propagation using component models that have to observe the no-function-in-structure principle.

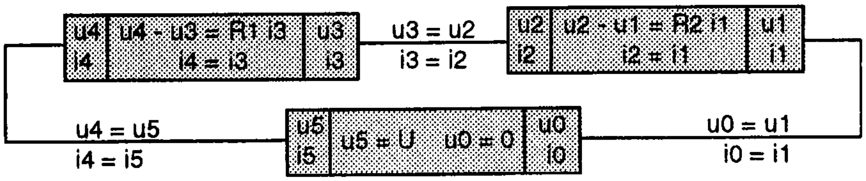


Figure 1: Two resistances R1 and R2 connected in series

In this paper we present a technique for qualitative circuit analysis that generates the above missing structural constraints for arbitrary resistive nets. Our method is a qualitative one since we exploit the topology of a given net and do not use any additional quantitative information.

In Section 2 we present an algorithm that reduces a given resitive net to a single total resistance with respect to a given source using series and parallel reduction and star-mesh conversion. We proof that the algorithm reduces arbitrary networks. In Section 3 we demonstrate, how the trees resulting from net reduction can be used to derive numeric, symbolic, or qualitative descriptions of the resistances, voltage drops, and currents observable in the given network.

2 SPS Reduction for Resistive Nets

In this paper we assume that the device under consideration can be represented as a resistive net. A resistive net consists of resistances and an ideal voltage source. Further we assume that all relevant assumptions about faulty or operational states of the device's components can be translated to assignments of certain values to the resistances and the source. For example, "wire broken" might translate to "r = ∞", and "switch closed" to "r = 0" in the net. As illustrated in Figure 2, a resistive net can be represented as a graph G = (E, V) where V is a finite set of nodes and E ⊆ V×V is a set of directed edges labeled with the resistance or source it represents. The direction of an edge e defines the sign of the current through e and of the voltage drop across e.

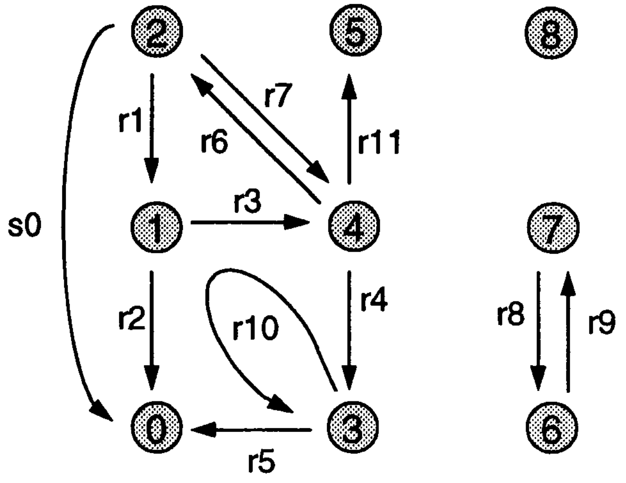


Figure 2: A resistive net represented as a directed graph

2.1 Some Definitions

Let e = (v0, v1) be an edge. The edge e is called directed from v0 to v1, v0 = start(e) is called the start node of e and v1 = end(e) is called the end node of e. Let nodes(e) be the set { start(e) } ∪ { end(e) }.

An edge e with start(e) = end(e) is called a <u>self-loop</u>. Two edges e0 and e1 with nodes(e0) = nodes(e1) and e0 is not a self-loop are called <u>parallel</u> with each other.

An edge set E containing no self-loops and no parallel edges is called <u>edge reduced</u>.

For every node v ∈ V of a graph G = (E, V), edges(v) is the set of edges starting or ending at v, and degree(v) is the number of edges contained in the edge set edges(v). A node v with degree(v) = 0 is called <u>isolated</u>.

Examples: In Figure 2, r6 and r7 are parallel edges, r10 is a self-loop, edges(v3) = {r4, r5, r10}, degree(v3) = 3, and v8 is isolated.

2.2 Basic Operations on Graphs

Edge removal An edge e can be removed from the graph. As a result, one or all of the nodes in nodes(e) may become isolated.

Node removal A node v can be removed from the graph by removing v and all the n = degree(v) edges starting or ending at v.

An edge can be removed from the graph by applying one of the two <u>edge removal rules</u>:

SELF-LOOP An edge that is a self-loop can be removed from the graph.

PARALLEL If an edge e0 = (v0, v1) is parallel to an edge e1 then e0 and e1 can be replaced by an edge e with e = (v0, v1) or e = (v1, v0), i.e. the direction of the replacing edge e can be chosen.

If a node v has a reduced edge set edges(v), v can be removed from the graph using one of the four <u>node removal rules</u> depending on the number n = degree(v) only:

ISOLATED NODE n = 0. Remove v.

END BRANCH n = 1. Because edges(v) = {e} is edge reduced, e is not a self-loop. Remove v. Following the definition of node removal, this will also remove e.

SERIES n = 2. Let edges(v) = {e1, e2} with node(e1) = {v1, v} and nodes e2 = {v2, v}. Because edges(v) is edge reduced, v1 and v2 are no self-loops, i.e. $v1 \neq v$ and $v2 \neq v$, and they are not parallel, i.e. $v1 \neq v2$. Remove v. This removes also e1 and e2. Add a new edge e with e = (v1, v2) or e = (v2, v1). Again, as for the PARALLEL removal rule, the direction of e can be chosen.

STAR n > 2. Let edges(v) = {e1, e2, ..., en} with nodes(ek) = {vk, v} for all k with $1 \leq k \leq n$. Because edges(v) is edge reduced we have $vk \neq v$ (no self-loops) and $vj \neq vk$ (no parallel) for all j and all k with $1 \leq j < k \leq n$. Remove v. This will also remove all the edges e1, e2, ..., en. Add n (n-1) / 2 directed edges ejk = (vj, vk) with $1 \leq j < k \leq n$.

Figure 3 gives some examples of the star removal rule, also known as star-mesh conversion or generalized star-delta conversion.

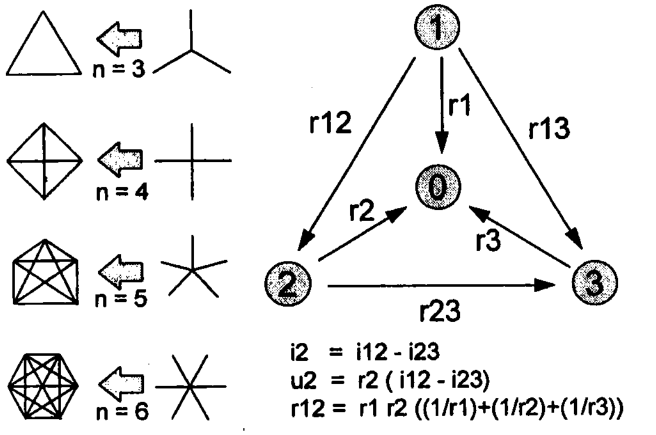


Figure 3: Some instances of the star removal rule and details for the case n = 3.

2.3 An Algorithm for SPS Reduction

We are now ready to present an algorithm that reduces any graph G = (E, V) representing a resistive net to a single edge e representing the total resistance of the net with respect to a given source $s \in E$. The reduction algorithm is given in Figure 4. The abbreviation sps stands for series-parallel-star because of the fact that every net can be reduced using series, parallel, and star reduction. We state this important property of the algorithm in the following theorem.

```
reduce-sps(E, V, s)
    for each edge $e \in E$, $e \neq s$
        apply an edge removal rule
        to e if possible
    end for

    for each node $v \in V$, $v \notin$ nodes(s)

        remove v by applying the node
        removal rule R(degree(v))

        for each new edge e added
        by the application of R
            apply an edge removal
            rule to e if possible
        end for

    end for
end reduce-sps
```

Figure 4: The reduction algorithm

Theorem: Let G = (E, V) be an arbitrary not neccessary connected graph that may contain isolated nodes, self-loops and parallel edges. Let $s \in E$ be an edge, called the source. Then the application of the procedure sps-reduce(E, V, s) as given above will reduce G to a graph G' = (E', V') with V' = nodes(s) and E' = {s} or E' = {s, e}.

Proof: It is not hard to see that if a graph G = (E, V) has a reduced edge set E, i.e. contains no self-loops and parallels, every node $v \in V$ can be removed from G using one of the node removal rules, resulting in a smaller node set V' = V \ {v}. Hence, every node $v \in V \setminus nodes\{s\}$ chosen in the for-each-node loop for removal can infact be removed by a node removal rule since the edge set E \ {s} has been edge reduced in the initial for-each-edge loop and remains edge reduced, since all edges added by a node removal rule are checked for self-loops and parallels in the for-each-new-edge loop. Edge removal cannot introduce new nodes. Therefore, after the for-each-node loop, all nodes except nodes(s) have been removed from V resulting in a reduced node set V' = nodes(s).

If the corresponding reduced edge set E' would contain more than two edges, say E' = {s, e1, e2}, e1 and e2 would have to be self-loops or parallel, since V' contains only one or two nodes. This would contradict the fact that E' \ {s} is edge-reduced. Hence, E' can contain only one or two edges, i.e. E' = {s} or E' = {s, e}.

To illustrate the reduction procedure, Figure 5 shows the reduction of the graph G0 given in Figure 2 to a graph G6 containing only two edges s0 and PA5.

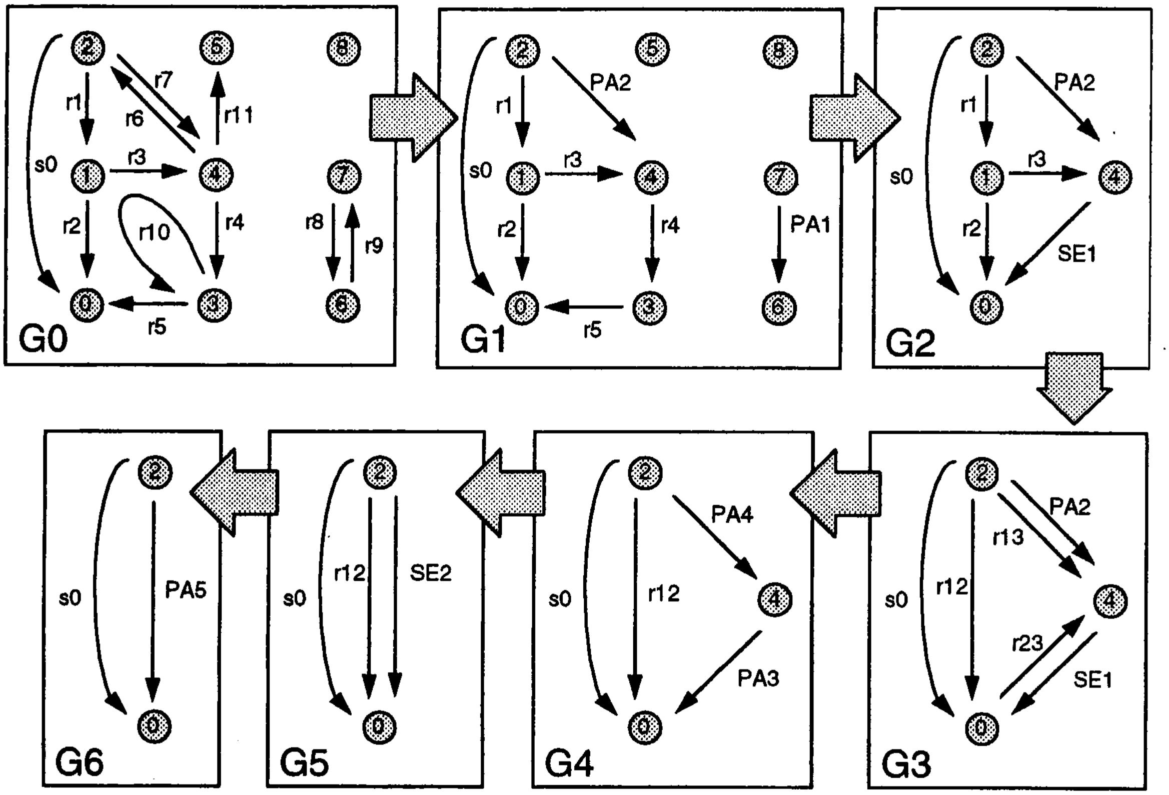


Figure 5: Sps reduction of a sample resistive network

The reduction algorithm does not define an ordering for removing nodes from the graph, any ordering is possible. However, it turns out that always choosing a node with minimal degree has some advantages: First, this strategy leads often to shorter reduction sequences.

Moreover, if a graph can be reduced using series-parallel reductions only the reduction algorithm will find such a reduction when minimal-degree ordering is applied. This avoids unneccessary star reductions. Minimal-degree ordering can be implemented by representing the node set V of a graph G = (E, V) as a list of nodes, sorted by the degree of the nodes and always choosing the first node of the list for removal. Adding and removing an edge will cause a reordering of that edge's start and end node.

3 Reasoning using SPS Trees

In the previous section we have presented an algorithm that decomposes an arbitrary resistive net using series, parallel and star-to-mesh reductions. In this section we show how the resulting sps trees - comparable to a tree resulting from syntax analysis in natural language processing - can be used to guide quantitative and qualitative reasoning about the decomposed net.

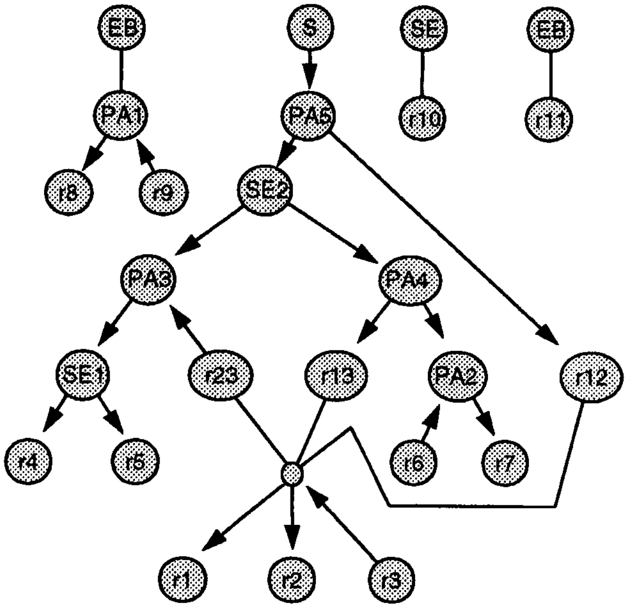


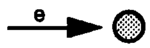
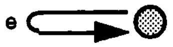
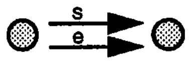
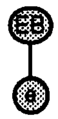
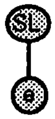
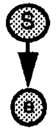
Figure 6: Sps trees resulting from the reduction given in Figure 5

3.1 Deriving Symbolic or Numeric Values

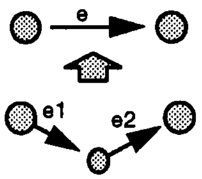
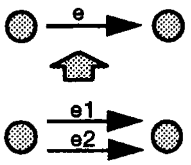
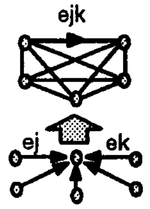
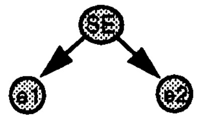
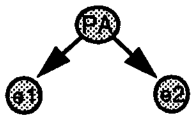
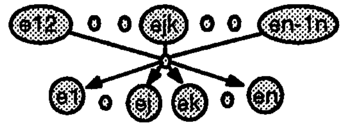
Each removal of an edge or node during sps reduction is represented by a non-leave node in the corresponding sps tree as illustrated in Figure 6. The arrows in the tree decode the sign of the currents and voltage drops. Downarrow represents a factor 1 and uparrow represents a factor -1.We are interested in the currents and voltage drops at the edges of the original unreduced graph. These edges are represented by the leaves of the sps trees.

The basic idea is now to propagate any request for a symbolic or numeric expression for a current or a voltage drop at a leave of a sps tree upwards to its root where the answer is known, since the root is e.g. given by the source edge. Propagation of a request is done using the algebraic relations given in Figure 7.

As one can see, the recursion for currents and voltages proceeds in a bottom-up manner from the leave of a sps tree to its root, while the recursion for a resistance leads in a top-down manner from the root to the leaves. To summarize, if we have generated the sps trees for a given resistive net, we can derive symbolic or numeric expressions for all voltage drops and currents in the net by recursively applying the algebraic relations in the table. The sps tree depends on the topology of the net only, i.e. has to be derived only once for a given net.

END BRANCH	SELF-LOOP	SOURCE
		
		

u(e) = 0
i(e) = 0
	u(e) = 0
	i(e) = 0
		u(e) = u(s)
		i(e) = u(s) / r(e)

SERIES	PARALLEL	STAR
		
		 e12 o o ejk o o en-1n e1 o ej ek o en

$1 \leq k \leq 2$:
i(ek) = i(SE)
u(ek) = i(SE) r(ek)
	$1 \leq k \leq 2$:
	i(ek) = u(PA) / r(ek)
	u(ek) = u(PA)
		$1 \leq k \leq n$:
		$i(ek) = \sum_{m=1}^{k-1} i(emk) - \sum_{m=k+1}^{n} i(ekm)$
		u(ek) = i(ek) r(ek)

r(SE) = r(e1) + r(e2)	$r(PA) = \frac{r(e1)\, r(e2)}{r(e1) + r(e2)}$	$1 \leq j < k \leq n$: $r(ejk) = r(ej)\, r(ek) \sum_{m=1}^{n} \frac{1}{r(em)}$
qr(SE)=Max(qr(e1),qr(e2))	qr(PA)=Min(qr(e1),qr(e2))	qr(ejk) = Min(qr(ej), qr(ek))

Figure 7: Graph reductions, corresponding sps trees and algebraic relations

3.2 Deriving Qualitative Values

Often, we are interested only in qualitative descriptions. For this purpose we define three sets of qualitative values, ordered with respect to < as indicated below and representing qualitative resistance and signed qualitative current and voltage drop.

```
R = (              ZERO, POS, INF)
I = (MINF, NEG,  ZERO, POS, INF)
U = (MMAX, NEG,  ZERO, POS, MAX)
```

Voltage drops occuring in a resistive net are bounded to a finite value MAX or minus MAX = MMAX, since we are assuming an ideal voltage source. As a consequence, the current through a ZERO resistance might be infinite, i.e. INF or MINF. We can now interprete the sps tree of a circuit as a constraint net. Every node of the tree holds three qualitative values qu, qr and qi. These values are related with each other by qualitative abstractions of the algebraic relations given in Figure 7. These abstractions turn out to be surprisingly simple. For example, in star-mesh conversion, the algebraic relation between rj, rk in the star and rjk in the mesh,

$$rjk = rj\, rk \sum_{i=1}^{n} (1 / r_i)$$

simplifies in its qualitative version to

$$qrjk = Min(qrj, qrk)$$

where Min(q1,q2) is the minimum of the qualitative values q1 and q2 with respect to the qualitative < relation defined above.

We have implemented sps reduction and constraint propagation for the resulting sps trees. Figure 8 shows the qualitative values for the currents derived by propagation where all resistances of the unreduced resistive net were initially set to POS and the voltage source was set to MAX. Qualitative analysis cannot derive a unique value for the current through the bridge resistor between nodes v1 and v4, i.e. the current can be NEG, ZERO, or POS. Only INF and MINF could be defintely excluded by the qualitative analysis. This is what we expect, since to determine the current through the bridge, we have to take the exact values of the resistances into account. However, if this quantitative information is known, we can derive a numeric value for the missing bridge current from the sps tree as described in Section 3.1.

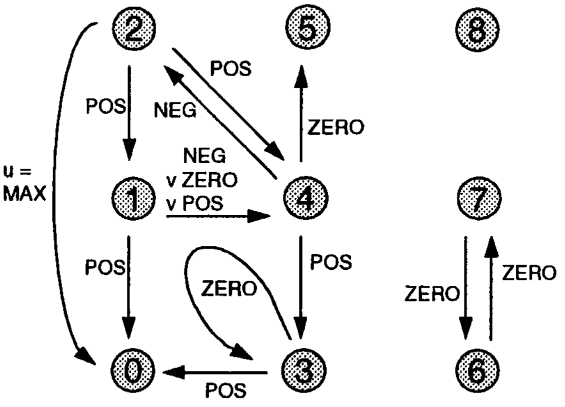


Figure 8: Qualitative currents derived from the sps trees

4 Related Work

In (Struss et al. 1995) the starting point is the same as in this paper: The authors demonstrate the limitations of local propagation with a three-resistance circuit concluding that global structural information should be exploited. For that purpose they introduce the notion of „connectivity of a component to sources and sinks of a circuit". They develop a component model that observes the no-function-in-structure principle but is able - to a certain degree - to derive the connectivity information for a component by local propagation only. The local connectivity information is used in turn to determine e.g. the sign of the current through the component. However, the method works only for circuits not containing certain loops and not containing too many series connections.

(Lee & Ormsby 1992) use series-parallel reduction of resistive nets in a way similar to the approach presented here. However, their approach is limited to series-parallel reducable networks, since they do not use star-mesh conversion. As a consequence, they cannot cope with resistive nets containing bridges, as for example the net given in Figure 8.

Bond graphs (cf. (Karnopp, Margolis, & Rosenberg 1990)) provide an elegant framework for representing dynamical systems. It turns out that a circuit's sps tree as used in this paper is isomorph to the circuit's bond graph if the circuit is series-parallel reducable, i.e. is reducable without star-mesh conversion. Otherwise the corresponding bond graph contains a loop, i.e. is not a tree.

(Chen 1976) presents an elaborated theory about the relations between graphs and linear systems representing e.g. general RLC-networks. These relations might proof to be useful for qualitative reasoning. However, it is still an open question - at least to the authors - how this work relates to the sps approach for resistive networks presented here.

5 Conclusion

We developed a method for reasoning about analog circuits or other devices whose behaviour can be modeled by resistive networks. The given network is first reduced to a total resistance using series- and parallel reduction, and star-to-mesh conversion. We proved that these three rules suffice to reduce arbitrary networks. We showed how the resulting sps tree (actually a dag - a directed acyclic graph) can be used to derive numeric, symbolic and qualitative values for the quantities occuring in the net.

Reasoning is not exclusively viewed as constraint propagation here but as a graph analysis task too. This enables us to exploit global structural information in a way that would be hard to achieve with traditional local propagation techniques only, i.e. without explicit graph analysis.

References

Chen, W. K. 1976. *Applied Graph Theory: Graphs and Electrical Networks*. North-Holland Publishing, Second Revised Edition.

Karnopp, D. C., Margolis, D. L., Rosenberg, R. C. 1990. *System Dynamics: A Unified Approach*. Second Edition, John Wiley & Sons.

Lee, M. H., Ormsby, A. R. T. 1992. Qualitative Modeling of Electrical Circuits, In QR'92: Sixth International Workshop on Qualitative Reasoning about Physical Systems, Heriot-Watt University, Edinburgh, Scotland, 155-169.

Mauss J., Neumann, B. 1995. Diagnosis by Algebraic Modeling and Fault-Tree Induction. In DX-95: Sixth International Workshop on Principles of Diagnosis, Goslar, Germany, 73-80.

Struss, P., Malik, A., Sachenbacher, M. 1995. Qualitative Modeling is the Key. In DX-95: Sixth International Workshop on Principles of Diagnosis, 99-106.

Sussmann, G. J., Steele, G. L. 1980. CONSTRAINTS - A Language for Expressing Almost-Hierarchical Descriptions, *Artificial Intelligence* 14, 1-39.