AUTOMATIC GENERATION OF DECISION TREES FOR DIAGNOSIS: THE MAD-SYSTEM

Jörg Kahl, Lothar Hotz, Heiko Milde, Stephanie Wessel
Labor für künstliche Intelligenz, Universität Hamburg

Abstract

There are several computer-based approaches which support maintenance engineers in their diagnosis task. In industrial practice, the application of decision trees is the prevailing method due to its simplicity and efficiency. The major shortcomings of this technique are its time-consuming and error-prone manual generation. Model-based approaches usually overcome these deficiencies. However, adapting the current industrial diagnosis processes to model-based approaches is, in general, extremely costly. Furthermore, practice has shown that current modelling techniques are not sufficient for some technical systems relevant in industry. This paper presents an integration of both aforementioned techniques by exploiting device models to automatically generate decision trees usable for diagnosis. Thus, compatibility with existing industrial diagnosis processes is achieved. The approach is realized with MAD, a hybrid system that focuses on diagnosing analog circuits. Due to its relevance to current research, the modelling and simulation part of MAD is the main focus of this paper. In MAD, analog circuits are modeled as linear networks. Normal and faulty behavior is computed using a calculus based on qualitative deviations.

1. Introduction

Since most technical systems become increasingly complex their diagnosis becomes a more and more sophisticated task. There exist several computer-based approaches which support maintenance engineers in their work. Two of the approaches, decision trees and model-based diagnosis, are of relevance for this paper. They are further introduced in the following.

Usually, in diagnosis, experts manually put up structures of fault sets in form of *decision trees* that lead maintenance engineers through their task [15]. The diagnosis process starts at the root that represents the set of all considered (i.e. modeled) faults. Going down the tree, fault sets are split into subsets of faults differing in the value of a tested property of the diagnosed device. Depending on the values measured by a maintenance engineer, the tree is traversed until a leaf is reached. At this point, no further fault set discrimination can be performed. Decision trees are a simple and efficient tool to diagnose even complex technical systems. They are the prevailing diagnosis technique in industrial practice. However, the initial design of a decision tree is often very time-consuming and error-prone, and technical system changes often lead to an equally time-consuming and error-prone manual modification of the decision tree.

Model-based approaches explicitly represent the structure of a device and the behavior of its components. Behavior descriptions have to be purely local so that they can be used at any place in the model (the no-function-in-structure principle [5]). The models are evaluated to predict the normal and, often, the faulty system behavior, too. If discrepancies occur between the modeled behavior and the actual device, the approach assumes a fault in the device. In order to diagnose it, model-based approaches generally have the following basic algorithm in common (see [4] for an introduction):

- *Hypothesis generation:* Given the device structure and the set of behavior discrepancies, compute the components that account for them.
- *Hypothesis testing:* Given the candidates produced during hypothesis generation, select the candidates that explain all observed discrepancies. To do this, search for minimal assumptions about correctly working components that must be retracted to be consistent with the observed device behavior. This is called consistency-based diagnosis. Often, the search is supported by component fault models. With that, device simulations can be used to compute faulty device behavior.
- *Hypothesis discrimination:* Given the candidates that account for all observed discrepancies (often, more than one hypothesis survives the previous phase) propose additional measurements that efficiently find the faulty components.

However, adapting industrial diagnosis processes to model-based approaches is, in general, extremely costly. First, model-based approaches require expensive hardware due to intensive computation time. Second, maintenance engineers used to decision tree techniques would need schooling to cope with the new technique. Furthermore, model-libraries containing models for all relevant components are needed to make model-based approaches usable in industrial practice. This is not state of the art, yet. Practice has also shown that current modelling techniques are not sufficient for some technical systems relevant in industry. For these reasons, model-based techniques are not yet widespread in industrial diagnosis applications.

This paper introduces an integration of both aforementioned techniques by exploiting device models to automatically generate decision trees usable for diagnosis (see [6] for a similar approach). Thus, compatibility of the approach with existing industrial diagnosis processes is achieved.

Note that in our approach, costly modelling and simulation computations are carried out off-line, before the diagnosis task, while in classical model-based diagnosis and in [6] behavior simulation is carried out during each diagnosis. Thus, the often mentioned scaling problem [4] is partly alleviated. Additionally, our decision trees can be manually changed to overcome the shortcoming of model-based approaches that some technical devices cannot be appropriately modeled, yet. The approach is realized with the system MAD (Modelling, Analyzing, and Diagnosing System) that focuses on the treatment of analog circuits. Our basic idea is to transform the device model created by the MAD user to a linear network that can be described by a system of linear equations based on Kirchhoff's and Ohm's laws. Assuming that a fault has occurred, this system of equations can be exploited in order to determine qualitative values for voltage and current metering points. In our approach, we simulate the device behavior for all faults to be considered relevant.

Diagnosing analog circuits means to determine faults in circuits by their effects, mostly given by voltage and current measurements. This can be considered as a typical classification task with faults as classes and voltage and current measurements as attribute values. Our simulation results containing all modeled faults and measurements provide all information needed to automatically structure this classification task by generating decision trees.

Section 2 of this paper describes MAD's device independent modelling process. Section 3 outlines the simulation process comprising all modeled faults. Decision tree construction and decision tree adaption is presented in Section 4.

2. Modelling analog circuits with MAD

This section starts with an evaluation of previous approaches to model analog circuits (Section 2.1).

A description of the novel qualitative values used in MAD follows (Section 2.2). At last, a user's view of MAD's modelling process is given (Section 2.3).

2.1. Evaluation of previous modelling approaches

In the past, many different approaches to model-based diagnosis of analog circuits have been published. For instance, if the circuit parameters can be described by crisp quantitative values, a linear network can be analyzed by existing tools such as SPICE [1] or systems based on CLP(*R*) (e.g. [2]). In order to cope with tolerances and inaccuracies, the DIANA system [3] uses quantitative intervals to describe network parameters. The FLAMES system [13] proposes fuzzy intervals to describe inaccuracies more adequately.

While these systems can be used to simulate a large class of analog circuits by exploiting detailed component models, [17] argue that for diagnostic purposes more abstract models are advantageous. In particular, linear networks with qualitative parameter values have been investigated in the literature.

For instance, adhering to the no-function-in-structure principle, the Connectivity Method [17] basically propagates qualitative information that encodes which port of a circuit component is connected to source or sink. However, not all kinds of circuits can be handled adequately. In order to overcome these deficiencies, [11] have developed a qualitative method to analyze linear networks by exploiting the *structure* of networks. The so-called SPS method explicitly represents a network's series-parallel-star structure as a tree (sps tree). As a result of the network analysis, for all currents and voltages, *sign-based* qualitative values are determined. The Connectivity Method and the SPS method mainly focus on the detection of *structural faults*, e.g. broken wires or comparable component faults such as blown light bulbs etc. They do not address, however, several other diagnosis topics which may arise in applications:

- non-structural faults such as slight deviations from normal behavior,
- deviative effects of non-structural and structural faults on parameter values,
- specific circuit topologies,
- abstractions in diagnosis models,
- variants.

This paper addresses most of these topics and presents a qualitative modelling method that is realized in MAD. The problems solved by the approach are explained with an application example. In particular, we consider a field regulator that is a subcomponent of a motor. A schematic diagram of the field regulator circuit is presented in Figure 1. The components shown in the figure are abstractions of the real physical components. For instance, the control switches T1 to T4 are actually implemented with transistors and diodes but, for diagnostic purposes, such a fine-grained representation is not required.

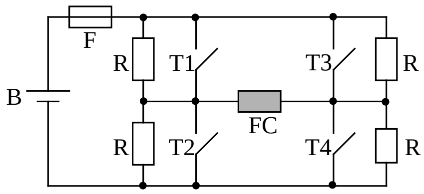

Figure 1: Field regulator comprising resistors (R), fuse (F), battery (B), field coil (FC), and controlled switches (T1 to T4).

This paper focuses on non-structural faults expressed by slight deviations from normal behavior, e.g. increased resistance values. Faults of this kind can neither be *modeled* by the SPS method nor by the Connectivity Method because of the sign-based qualitative values used by these methods (there are qualitative values, typically ″0″, ″-″, ″+″, representing a fixed landmark and values below and above, respectively). In analog circuits the occurrence of a fault affects all currents and voltages, i.e. the absolute values of parameters change. However, in most cases, the parameters do not change their signs (or reach certain fixed landmarks). Thus, it is hardly possible to adequately derive these fault *effects* using sign-based qualitative values. Furthermore, bridge circuit topologies are relevant in our domain (see Figure 1). These circuits pose a special problem for qualitative approaches because the direction of the current through the bridge resistor usually depends on the exact quantitative values of the component parameters.

Since we want to model abstractions of real components, *quantitative* modeling systems (see above) are not appropriate, either. In addition, we also have to deal with the “variants’ dilemma” [17]. This means that a certain model of the field regulator should cover several variants of this device. Variants differ only slightly concerning values of component parameters. Since qualitative values cover intervals of quantitative values, they are more suitable to cover several variants in the same model.

Based on the SPS method MAD realizes a new qualitative approach for reasoning about analog circuits for diagnostic purposes. The main features of the method are:

- The qualitative values represent deviations as well as sign information. With deviations we can describe non-structural faults such as “resistance too high” as well as structural faults such as blown light bulbs (“resistance too high and infinite”) even in bridge circuit topologies.
- The semantics of the qualitative values is grounded in the quantitative nature of landmarks. This way we can show soundness and completeness of the derivation algorithm for qualitative reasoning, cf. also [16].
- It is not necessary to specify the absolute quantitative values of landmarks. Thus, we can deal with abstract circuit components and the “variants’ dilemma”. The method presented in this paper derives simulation results that are sufficient for fault discrimination in our diagnosis application.

Qualitative values for expressing deviations to adequately describe faults and their effects are introduced in the next section.

2.2. The qualitative values of MAD

In a linear network, there are currents and voltages whose directions can be determined by the structure of the network (we will use ″voltage″ and ″current″ although these quantities could also be ″pressure″ and ″flow″ or other corresponding quantities in linear networks). These currents and voltages as well as resistances can be adequately described by a set of qualitative values that cover the extended *positive* real number line $[0, \infty]$ [1]. For some currents and voltages, the directions are not determined by the network structure. Thus, we also need qualitative values that cover the *whole* extended real number line $[-\infty, \infty]$. Distinguishing between two types of qualitative value sets is not merely a syntactic aspect but sharpens the reasoning about effects in linear networks.
To elaborate, the qualitative values for positive real numbers and their semantics are shown in Table 1. Although the landmark values in the semantics definition are not specified as fixed quantitative values, MAD does rely on the order between landmarks: $0 < min < max < \infty$.

---

[1] Note that although infinite voltages and currents do not occur in practice, it is useful to allow infinite qualitative values in our approach. See [10] for further details.

qualitative value	abbreviation	semantics
low_0	0	$[0, 0]$
low	l	$(0, min)$
normal	n	$[min, max]$
high	h	(max, ∞)
high_inf	∞	$[\infty, \infty]$

Table 1: Qualitative values expressed by quantitative intervals

Describing resistances, qualitative values can be used to model structural as well as non-structural faults. On the one hand, structural faults such as short circuits and broken wires can be described by the extreme deviations *low_0* and *high_inf*, respectively. On the other hand, non-structural faults such as partial short circuits in coils and corroded wiring points can be modeled by the qualitative values *low* and *high*, respectively. The faultless state of a component is represented by the qualitative value *normal* which represents an interval. This enables MAD to be tolerant with respect to differential deviations such as physical tolerances and temperature drifts. For details of current and voltage modelling, see [10].

2.3. The MAD user's view of the modelling process

Using MAD, structures of analog circuits are modeled with a special component editor called COMEDI (COMponent EDItor). Similar to CAD-tools, MAD users select items out of a predefined set of basic electrical and electronic components and define the connections between them. The existence of such a component library is highly important to guarantee reusability for a big variety of analog circuits. Within the library, every component is equipped with a set of fault models which is modifiable by COMEDI users. Additionally, users can define new device components. The faults are described with the above-mentioned qualitative values expressing deviations. Additionally, users have to determine where maintenance engineers can perform measurements during a diagnosis. In MAD, these so called metering points can be voltage or current measurements. If changes occur in device design or metering point positioning, users only have to adapt the COMEDI model. After that, the respective decision tree will be automatically regenerated.

Note that not all COMEDI users must have knowledge about the underlying fault models and device component definitions. If the problem is to define a model with existing device components and fault models, the task can be done by users only familiar with the device structure.

Due to the model-based approach, the behavior descriptions stored in MAD's component library have to be purely local. Since component behavior strongly depends on its place inside the device, most parameter values cannot be computed without a device structure analysis. MAD's analysis process is outlined in the next section.

3. Analyzing analog circuits with MAD

Section 3.1 gives an overview of how linear equations are derived from structures of linear networks and how the qualitative values of these equations are evaluated. Section 3.2 shows this process by means of our application example, the field regulator.

3.1. Qualitative analysis of linear networks

The approach realized in MAD to the qualitative analysis of a linear network consists of two main steps.

SDSP transformation: The linear network is transformed by series-parallel reductions, star-delta and delta-star conversions well known in electrical engineering into an SP-tree. The SP-tree is an explicit representation of the structure of the network that can be used to compute qualitative behavior for different kinds of faults by a local propagation method.

As a difference to the SPS method in [11], we would like to emphasize that we use star-delta as well as delta-star conversions (hence the name of our method: SDSP). This is advantageous because, in comparison to the SPS method, new classes of network topologies can be evaluated. As a further difference, we restrict stars and deltas to be transformed to those with three edges - with the purpose to obtain a fixed number of equation types. This cannot be ensured by the SPS method. As a disadvantage of this restriction to stars and deltas with three edges, we cannot treat some exceptional networks (e.g. networks consisting exclusively of four-edge stars without any delta and star transformations with three edges applicable). According to our experiences, these networks are hardly relevant in practice.

Local propagation of qualitative values: The second main step consists of local propagation of qualitative values in the SP-tree in order to simulate circuit behavior (i.e. the step can be carried out for each of the supplied fault models).

- Applying the star-delta transformation, values of transformation resistances are determined.
- Values of resistances are propagated from the leaves of the SP-tree to its root by exploiting two electrical laws describing compensation resistances of series and parallel groupings.
- Values for currents and voltages are propagated from the root of the SP-tree to its leaves. For that, four different electrical laws are evaluated, i.e. current divider, voltage divider, same voltages and same currents rule.
- Values of voltages and currents of the original network are determined by exploiting current and voltage transformations that are a part of the star-delta transformation.

Note that in qualitative reasoning, in general, the computation of qualitative values can be very problematic. In our approach, a well-founded qualitative calculus can be specified which is based on the assumption that in normal behavior all measurements have the qualitative value ˝normal˝. The algorithm is based on the interval calculus that is described in [10] and [12].

3.2. SDSP-analyzing the field regulator

In order to outline the strength of our approach we now show how to model and analyze the field regulator of our application domain. Faulty behavior of the field regulator is modeled by ‘low’ (e.g., R5_l in Figure 2) as an example.

The first step of the SDSP method is a star-delta conversion (see Figure 2). A subsequent SP-reduction leads to the SP-tree shown in Figure 3. Letters S and P indicate that nodes represent series and parallel groupings, respectively. For example, R12 and R4 are replaced by the compensation resistance P6 (parallel reduction), P6 and P7 are reduced by S8 (serial reduction), and so on.

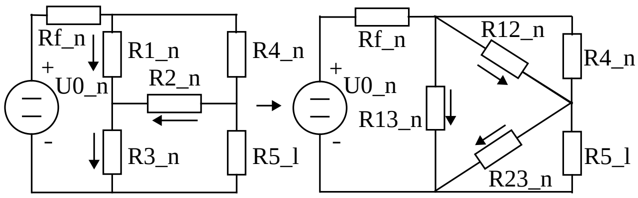

Figure 2: Star-delta conversion

The second step of the SDSP method is the local propagation of qualitative values. First, the transformed resistance values are determined (i.e., R12, R13, and R23, see Figure 2). Second, qualitative values of resistances, currents and voltages are propagated (see the labels of the arrows and the corresponding legend in Figure 3 to determine the direction and propagation rule of the respective propagation).

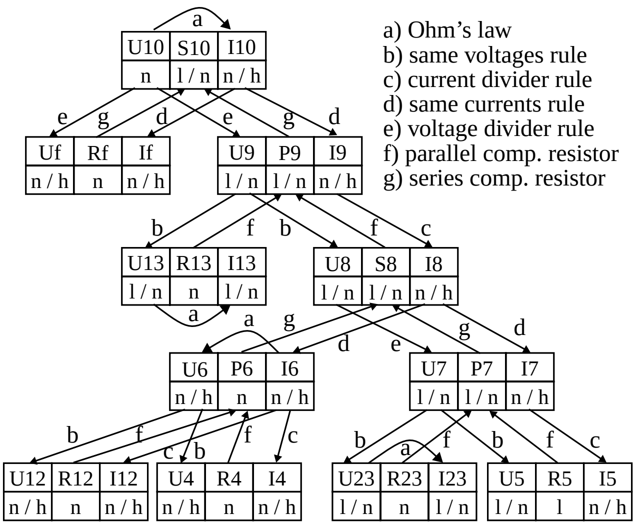

Figure 3: Propagation of qualitative values

The final step of the SDSP method is a voltage and current retransformation in order to obtain the qualitative values of the original network (i.e., compute the respective values for R1, R2, and R3, see Figure 2).

The implementation treats an SP-tree as a constraint net. The constraint propagation algorithm we apply is similar to the Waltz algorithm [18]. For more implementation details of this type of constraint systems, see [7] and [8].

The SDSP method is used to compute an exhaustive simulation for all faults which are considered relevant for the model (cf. Section 2.3), i.e. qualitative values that describe certain component faults are successively applied to compute all parameter values of the SP-tree. From these computations, we derive a data structure called fault relation that contains qualitative values for all defined metering points. For the field regulator example, consider the current and voltage metering points shown in Figure 4.

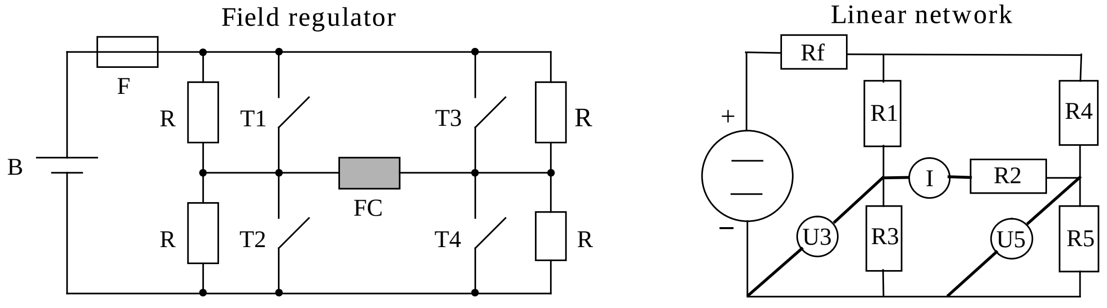

Figure 4: Linear network with three metering points (I, U3, U5)

Some modeled faults and the respective metering point values are shown in Table 2. They are computed by propagating the qualitative values through the SP-tree. Note that due to the qualitative reasoning process, metering points often comprise more than one qualitative value.

Faults \ Measurements	I	U3	U5
F1: Fuse blown	low_0	low_0	low_0
F2: Switch T2 high-impedance	low / normal	normal / high	normal / high
F3: Insulation damage in field coil	normal / high	normal / high	normal / high
F4: Battery low	low / normal	low / normal	low / normal
F5: Switch T3 high-impedance	low / normal	low / normal	low / normal

Table 2: Fault relation of field regulator

Fault relations are the input for the next step in our diagnosis system, the decision tree generation.

4. Generating and editing decision trees with MAD

Section 4.1. explains the automatic generation of decision trees with fault relations as basic data. Section 4.2 elaborates possibilities to adapt the decision tree in case of changing or additional requirements.

4.1. Generating decision trees

Several algorithms have been developed for decision tree construction [Quinlan 86]. Since the well-known ID3-algorithm often provides nearly optimal solutions with low computational costs, we decided to integrate this algorithm into MAD. To cope with applications with a bad ID3-performance, the computationally more costly but optimal A*-algorithm [9] can be used in MAD, alternatively. In this paper, we concentrate on our ID3-application.

Atypical for ID3 applications, MAD does not build up training and test sets. Instead, decision trees can be generated in one step since the fault relation contains all data required for this process. The idea of the algorithm is to select the most informative measurements for fault discrimination, proceeding from the decision tree root down to its leaves. For a detailed description of ID3, see [14]. Figure 5 shows the respective decision tree for the fault relation of Table 2. Note that F4 and F5 cannot be further discriminated due to the lack of discriminative measurement values in the fault relation.

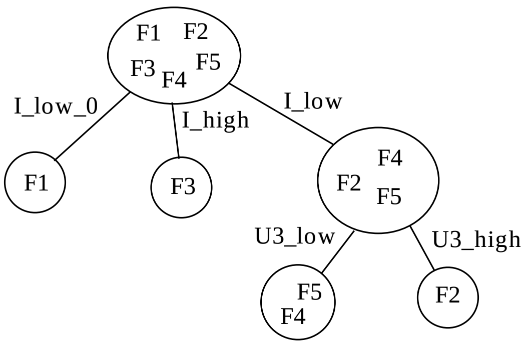

Figure 5: Decision tree of the field regulator

In industrial practice, in general, measurements are not equally costly which affects the optimal measurement order. MAD regards varying measurement costs in several forms. Due to space limitations, we omit a discussion of this aspect, here. For more information, see [19].

4.2. Editing decision trees

Since practice has shown that current modelling techniques are not sufficient for some technical systems relevant in industry, MAD offers the possibility to manually change the decision tree. Furthermore, the ability to adapt it to special needs and ease modifications is an important condition for acceptance in industry. The following basic manipulations are supported:

- Insert a new fault into the decision tree.
- Delete an existing fault out of the decision tree.
- Merge existing faults in different nodes into a new node.

In part, comprehensive adaptations of the value tests at the decision tree edges must be performed. Since decision trees do not contain all information needed to perform the adaptions mentioned above, the fault relation is an essential data structure to obtain missing information. Due to space limitations in this paper, a discussion of these aspects is omitted. For further information, see [19].

Typically, the data of the generated decision trees will be transferred to small portable computers in order to support maintenance personnel at the diagnoses site. Starting at the decision tree root, the computer asks values of measurements attached to the respective edges. One or more faults suspected for the faulty device behavior are the output when reaching a leaf of the decision tree. More than one fault can be found at a leaf if the metering points defined in the model are not sufficient to discriminate all faults.

5. Summary

MAD (Modelling, Analyzing, and Diagnosing System) is a hybrid system that integrates the advantages of model-based diagnosis approaches with the industrial practice of using decision trees. Similar to CAD-tools, a special component editor is provided that can be used to model devices and define metering points. Internally, MAD represents analog circuits as linear networks. Resistances, voltages, and currents are described by qualitative values which can express deviations from the faultless state. In order to compute parameter values for a certain circuit, MAD analyses the circuit structure with an algorithm called SDSP method. This algorithm exploits star, delta, serial, and parallel structures found in the model of the circuit. For each fault considered in the model, MAD computes values of currents and voltages, that is, a fault relation is generated. For each fault, the fault relation contains qualitative values for all metering points. An ID3 implementation uses the fault relation to create an optimal decision tree.

The work has reached the stage of a laboratory prototype which will be the basis for an industrial implementation and practical field tests.

References

[1] BANZHAF, W., Computer-aided circuit analysis using SPICE, 1989.

[2] BIASIZZO, A., NOVAK, F., Model-Based Diagnosis of Analog Circuits with CLP(R), Technical report CSD-TR-95-9, Jozef Stefan Institute, Ljubljana, Slovenia, 1995.

[3] DAGUE, P., JHEL, O., TALLIBERT, P., An Interval Propagation and Conflict Recognition Engine for Diagnosing Continues Dynamic Systems, in: Lecture Notes in AI, 462, 1990.

[4] DAVIS, R., HAMSCHER, W.: Model-based reasoning: Troubleshooting., in: H. E. Shrobe, (ed.), Survey Talks from the National Conferences on Artificial Intelligence, 1988.

[5] DE KLEER, J., BROWN, J. S.: A Qualitative Physics Based on Confluences. in: AI Journal, 1984.

[6] DVORAK, D., KUIPERS, B., Model-based monitoring of dynamic systems, in: Proc. 11th Int. Joint Conf. on Artificial Intelligence, 1989.

[7] GUELDEN, O., Entwurf und Implementierung eines Constraintsystems fuer das Werkzeugsystem KONWERK, PROKON-Memo Nr. 46, Univ. Hamburg, 1993.

[8] GUESGEN, H.-W., A System for Constraint Satisfaction, Thesis, Univ. Kaiserslautern, 1988.

[9] HART, P. E., NILSSON, N. J., RAPHAEL, B., A formal basis for the heuristic determination of minimum cost paths, in: IEEE Transactions on Systems, Science and Cybernetics, SSC-4(2), 1968.

[10] HOTZ, L., MILDE, L., MOELLER, R., NEUMANN, B.: Resistive Networks Revisited: Exploitation of Network Structures and Qualitative Reasoning about Deviation is the Key, in: M.-O. Cordier, (ed.), 8th International Workshop on Principles of Diagnosis (DX'97), 1997.

[11] MAUSS, J., NEUMANN, B., How to Guide Qualitative Reasoning about Electrical Circuits by Series-Parallel Trees, in: Proc. QR'96, 10th Int. Workshop on QR, 1996.

[12] MILDE, H., Untersuchung der SDSP-Analyse auf Vollständigkeit und Korrektheit, LKI-Report LKI-M-97/3, Artificial Intelligence Lab., Univ. Hamburg, 1997.

[13] MOHAMED, F., MARZOUKI, M., Test and Diagnosis of Analog Circuits: When Fuzziness can Lead to Accuracy, in: Journal of Electronic Testing: Theory and Applications, 9, 1-15 (1996), 1996.

[14] QUINLAN, J. R.: Induction of decision trees, in: Machine Learning, 1:81-106, 1986.

[15] QUINLAN, J. R.: C4.5: Programs for Machine Learning, 1993.

[16] STRUSS, P., Problems of Interval-Based Qualitative Reasoning, in: D. Weld, J. de Kleer, (ed.), Readings in Qualitative Reasoning about Physical Systems, 1990.

[17] STRUSS, P., MALIK, A., SACHENBACHER, M.: Qualitative Modelling is the Key. in: 6th International Workshop on Principles of Diagnosis (DX'95), 1995.

[18] WALTZ, D.: Understanding line drawings of scenes with shadows, in: P. H. Winston, (ed.), The Psychology of Computer Vision, 1975.

[19] WESSEL, S. HOTZ, L., KAHL, J., MILDE, H., Test Selection for Diagnosing Technical Systems: Automatic and Interactive Methods, LKI-Report LKI-M-98/1, Artificial Intelligence Lab., Univ. Hamburg,1998.