On linear constraints[*] with uncertain coefficients

Extended abstract for the CP2001 Doctoral Programme

Frank Seelisch[†]
Knowledge-based Engineering
Research & Technology
DaimlerChrysler AG

May 14, 2001

1 Introduction

Many engineering tasks can be reformulated as CSPs. There exist digital libraries that contain mathematical models for a great variety of physical components. In an electrics library those are models for Ohmic resistors, diodes, voltage sources, etc. These component models "know" their associated mathematical constraints, and when plugging them together (i.e. connecting them by wires, pipes, etc.), one obtains a mathematical description of the entire artifact modeled (see [MMT]). In practice, the generated constraint models happen to be very large, and consist of many different types of constraints, such as linear, non-linear, finite domain, or even procedural constraints.
Usually, one has to consider alternative variants for single components (in design), or distinct behavioural modes of certain components (in diagnosis), which will introduce disjunctions and might produce many different CSP instances.[1] In this work, we will focus only on single CSP instances.
Moreover, the constraints of interest will stem from a special class of algebraic constraints over real-valued variables, namely a subclass of quadratic equations. As guiding examples we are going to use mathematical models of electrical resistive networks, although the class of applications we are interested in, and our formalism can handle, is much larger than that.
For solving problems composed of such constraints, we will present a generalization of *Gauss's Elimination Algorithm* which can be applied to this class of non-linear equations. Other existing methods will then be compared to the new approach on the basis of the guiding examples.

2 Investigated CSP instances

2.1 CSPs arising from resistive networks

Figure 1a shows the constraints associated to four components from an electrics library. As a convention and to avoid ambiguities, currents always point inwards w.r.t. the component. The pairs (v, c) are *voltage current pairs* associated with each *port*.[2]
In Figure 1b, a small electric circuit is depicted. The grey box contains the corresponding CSP instance, which is easily obtained by instantiating the constraint sets of Figure 1a. (Note that

[*]The constraints we are going to talk about here are, unless stated otherwise, defined over real-valued variables only. We will also use the term *Constraint Satisfaction Problem (CSP)* for such constraints, even though, in the literature, it seems to be used almost exclusively for finite-domain constraints.

[†]email: Frank.Seelisch@DaimlerChrysler.Com phone: +49 (030) 39982-391 fax: -107

[1]The described disjunctions of CSPs contribute significantly to the complexity of the underlying engineering task. In many cases, however, the CSP instances are very similar, a feature which can be efficiently utilized.

[2]A port can be thought of as an interface "pin".

wires just add two constraints to the CSP, an equality for voltages and one of the form $c_A + c_B = 0$, relating currents.)

Any such resistive network induces a system of equations over real-valued variables. The crucial

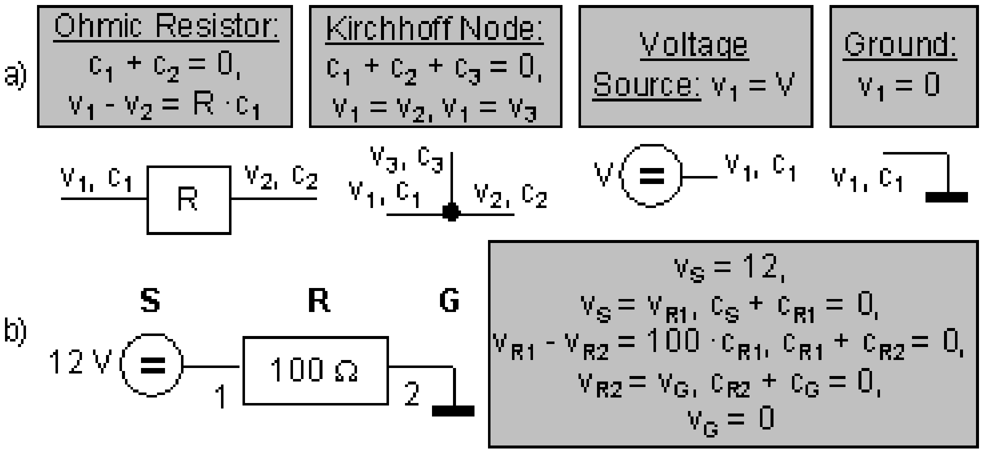

Figure 1: *component models from an electrics library, and a small circuit*

point of this paper is to consider these CSPs under the assumption of *uncertain parameter knowledge* about the voltage (provided by the ideal voltage source) and all occuring Ohmic resistances.

2.2 Benchmark CSP instances

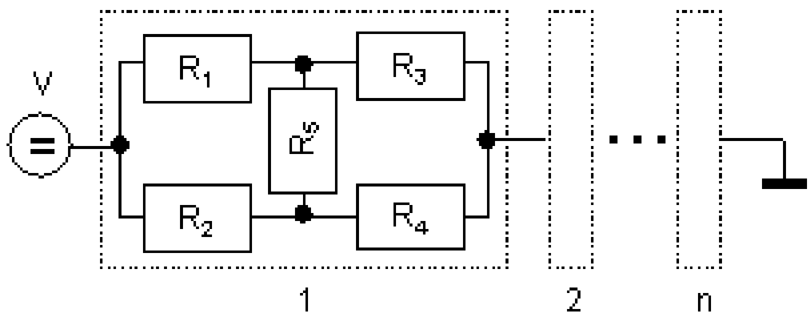

Figure 2: *the benchmark circuit family*

Figure 2 shows the family of benchmark circuits that will be used later on to compare alternative methods described in section 4. n denotes the number of repetitions (in series) of the left-most box consisting of five resistors. Note that none of the circuits is serial-parallel reducable. The parameters $R_i, i \in \{1, 2, 3, 4, 5\}$, which are the same in each box, and V will be provided by six *uncertainty conditions* of the form *variable* $\in S$, where S denotes a bounded, closed, convex real subset[3].

How many variables and constraints are there? Voltage source and ground emit one constraint each. Kirchhoff nodes give three constraints each, and Ohmic resistors two. Per box, there are 10 wires. This provides us with 42 constraints per box. Also, one has $n + 1$ wires connecting the

[3] S is thus either a bounded, closed interval or a singleton set with exactly one real element.

source, the boxes and ground in series, yielding $42n + 2(n + 1) + 2 = 44n + 4$ constraints in total. We also have $22n + 2$ ports, meaning $2(22n + 2) = 44n + 4$, i.e. just as many variables.
The set of solutions to the problem depends on the extra conditions $R_k \in I_k$, $k \in \{1, 2, 3, 4, 5\}$, and $V \in J$. So let us denote a problem instance by

$$\Pi(n, I_1, I_2, I_3, I_4, I_5, J).$$

In the case of fixed real values for the $R_k's$ and V, the $I_k's$ and J become singleton sets, and the instance is just a regular system of linear equations. It is straightforward to verify that

$$\begin{aligned} i_G &= \frac{V}{n} \cdot \frac{R_1R_3 + R_1R_4 + R_1R_5 + R_2R_3 + \ldots}{R_1R_2R_3 + R_1R_2R_4 + R_1R_2R_5 + R_1R_3R_4 + \ldots} \\ &\quad \frac{\ldots + R_2R_4 + R_2R_5 + R_3R_5 + R_4R_5}{\ldots + R_1R_4R_5 + R_2R_3R_4 + R_2R_3R_5 + R_3R_4R_5}, \end{aligned} \tag{1}$$

where i_G denotes the current into ground.

3 Formalisation

Definition 1 (Linear equation with uncertain coefficients)
Let $n \in \mathbb{N}$; $x_i, i \in \{1, 2, \ldots, n\}$, and $y_j, j \in \{0, 1, 2, \ldots, n\}$, be mutually distinct real-valued variables. The quadratic constraint

$$y_1 \cdot x_1 + y_2 \cdot x_2 + \ldots + y_n \cdot x_n = y_0 \tag{2}$$

is termed a **linear equation with uncertain coefficients** *iff for each $j \in \{0, 1, 2, \ldots, n\}$ there is a pair of real numbers a_j, b_j such that $a_j \le y_j \le b_j$. A vector $(s_1, s_2, \ldots, s_n) \in \mathbb{R}^n$ is called a* **solution** *iff for each $j \in \{0, 1, 2, \ldots, n\}$ there exist $t_j \in [a_j, b_j]$ such that $t_1 \cdot s_1 + t_2 \cdot s_2 + \ldots + t_n \cdot s_n = t_0$. We may also denote the constraint by*

$$[a_1, b_1] \cdot x_1 + [a_2, b_2] \cdot x_2 + \ldots + [a_n, b_n] \cdot x_n = [a_0, b_0].$$

Example 1
Let us take Ohm's law from the Ohmic resistor in Figure 1a. Suppose R is known to lie within $[100 - \delta, 100 + \delta]\,\Omega$ for some (small) $\delta > 0$. Then, we can rewrite the conjunction $v_1 - v_2 = R \cdot c_1 \;\wedge\; 100 - \delta \le R \le 100 + \delta$ as the *linear equation with uncertain coefficient(s)* $v_1 - v_2 = [100 - \delta, 100 + \delta] \cdot c_1$, omitting occurences of R.

4 Alternative methods for solving linear equations with uncertain coefficients

A solution to a CSP, derived from a resistive network with uncertain voltage and resistance parameters, is a vector of voltages and currents that assign values to all the unknowns[4] such that all constraints simultaneously hold. Solving such a CSP means to find all solution vectors. In practice, these can be infinitely many, so that an implementation should return an implicit representation of the solution set that allows an efficient retrieval.

[4]We have one (voltage, current) pair of unknowns per port.

4.1 Non-linear optimization ($\mathbb{NO}$)

As an example, let us go back to Formula (1). Suppose, the R_k's and V are only known to lie within some (small) bounded, closed interval. Then finding the strongest restriction on i_G is not at all trivial. It involves finding the minimum and maximum values of a rational function[5] f on a bounded, closed subset of $\mathbb{R}^m$, where m denotes the arity of f. Apart from that, we can only hope to apply *non-linear optimization techniques* here, if we have symbolically processed our original constraints, in order to obtain the rational functions to be optimized. In this case, we note that the entire solving algorithm will be complete if and only if the non-linear optimization is.[6]

4.2 Groebner Bases ($\mathbb{GB}$)

Applying algorithms based on *Groebner Bases* means to treat *linear equations with uncertain coefficients* as quadratic polynomials in several variables, as stated in Formula (2). This guarantees complete results but is expected to perform rather poorly, w.r.t. runtime, on large CSP instances, as e.g. $\Pi(n, I_1, I_2, I_3, I_4, I_5, J)$ for large n.

4.3 A new approach based on Gauss's Elimination Algorithm

The following concepts have - at least partly - been implemented, and our first test results proof them to be a very valuable alternative compared to existing methods. As we will point out next, these concepts allow us to trade computational completeness for runtime.

4.3.1 Gauss's Elimination + naiv interval calculus ($\mathbb{GN}$)

Definition 2 (Naiv interval calculus[7])
Let I, J be bounded, closed real intervals; and $\circ$ be any of the arithmetic operations $+, -, \cdot, \div$. Then **naiv interval arithmetic** *is defined according to*

$$I \circ J \stackrel{\text{def}}{=} \{\, i \circ j \;:\; i \in I, j \in J \,\}, \tag{3}$$

where $\div$ is defined only if $0 \notin J$.

It is easy to see that the resulting subsets of $\mathbb{R}$ are again bounded, closed real intervals.
Now, having defined the four basic arithmetic operations also for uncertain coefficients, *Gauss's Elimination Algorithm* (see [BS2]) can immediately be applied to *linear equations with uncertain coefficients*. Runtimes have shown to be similar to runtimes obtained for ordinary linear equations. The price for good runtime results is incompleteness of the computational results, due to the incompleteness of naiv interval calculus. Figure 3 illustrates the effect of naiv interval calculus along the elimination of two variables.[8] The strategy of choosing an arbitrary path in the tree will, in this example, find the complete solution $z = 1$ only in 2 of 7 cases. This leads us to the next improvement.

---

[5]For resistive networks, this function is the quotient of two determinantes as can be seen applying Cramer's rule (see [BS1]); an example is the term in Formula (1).

[6]For non-linear optimization in connection with interval computations, see [RR] and [KLRK].

[7]Obviously, this definition can be extended to arbitrary subsets of $\mathbb{R}$.

[8]The arrow annotations tell us which variable had been eliminated using which equation for producing a substitution term. If a coefficient contains 0, the equation cannot emit a substitute for that variable.

4.3.2 $\mathbb{GN}$ + elimination heuristic ($\mathbb{GNH}$)

Since traversing the entire elimination tree will not be a practicable option for large CSPs, we can try to follow the most promising paths, in terms of preserved completeness. This can be controlled by locally informed heuristics which choose the best variable to be eliminated next, and the best equation for providing a substitute.[9] For instance, in the example of Figure 3, it is better to eliminate y first, since its coefficients are all singletons. Also, in view of later eliminations, coefficients of other variables will become important and must be considered in powerful local heuristics.

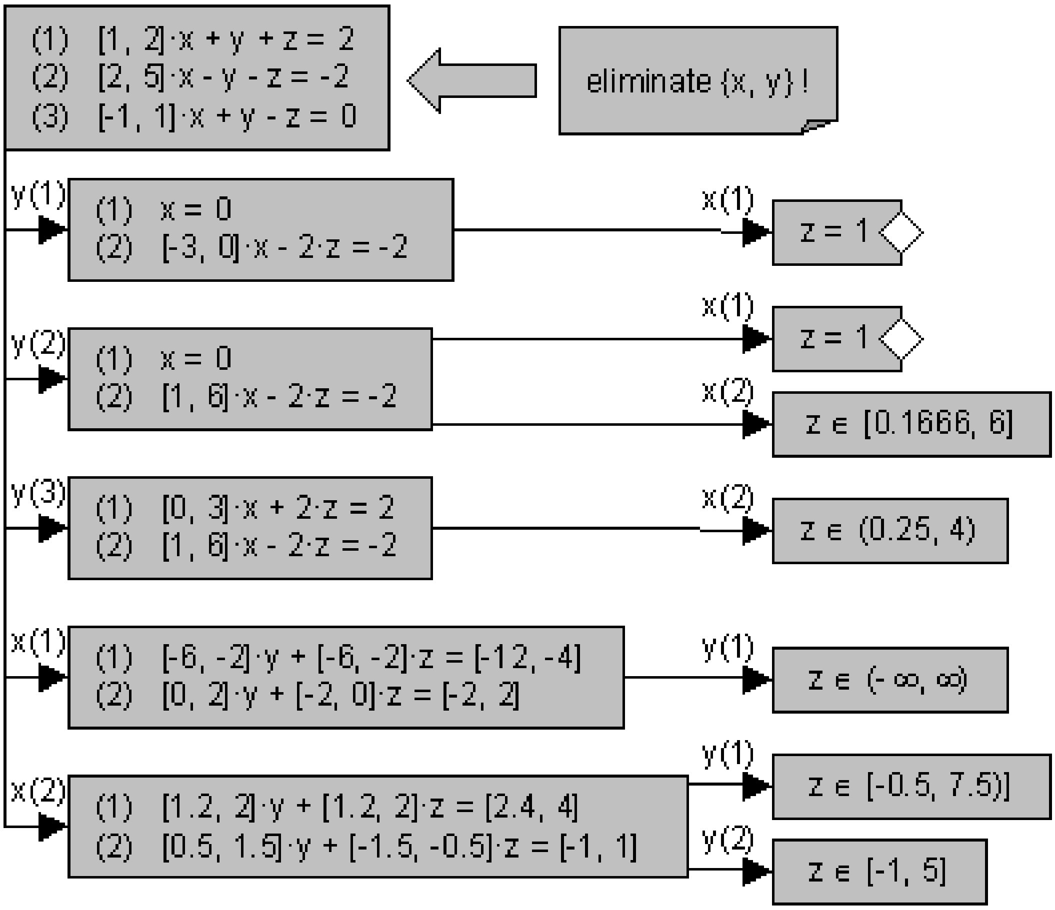

Figure 3: *an elimination tree with two complete computational paths*

4.3.3 Gauss Elimination + enhanced interval calculus + elimination heuristic ($\mathbb{GEH}$)

Heuristics alone are not expected to resolve the problem of incompleteness. We should furthermore try to replace naiv interval calculus by more sophisticated methods. This is what $\mathbb{GEH}$ attempts.

$\mathbb{GEH}$ is based on the postponement of naiv interval operations which are stored, together with their arguments, in arithmetic trees, as depicted in Figure 4.[10] This enables us to make symbolic

[9]We are currently investigating how structural analysis on constraint hypergraphs can contribute to determining "best" elimination sequences (see [GLS1] and [GLS2]).

[10]Interval coefficients are stored together with their (hidden) variables, in the sense of the $y_j' s$ from Definition 1.

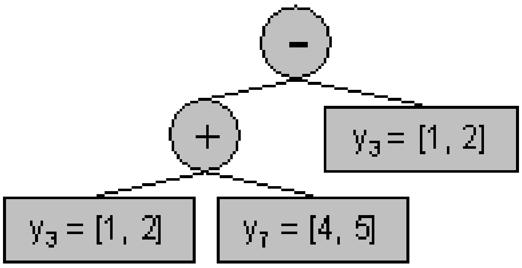

Figure 4: *an arithmetic tree of depth 2*

simplifications in order to reduce the complexity of the terms: The term in Figure 4 naively evaluates to $([1,2]+[4,5])-[1,2]=[5,7]-[1,2]=[3,6]$. But utilizing the fact that both occurences of $[1,2]$ are due to the *same* coefficient, yields the complete, i.e. most restrictive, solution $[4,5]$. $\mathbb{GEH}$ can be implemented to be adjustable to bounded tree depths. Exceeding the depth then leads to partial naiv evaluations until the depth is re-established. Also, the power of $\mathbb{GEH}$ depends strongly on the set of implemented simplification rules.

5 Conclusions

We have presented a family of new methods for solving systems of linear equations with uncertain coefficients. Test results obtained from a first Smalltalk implementation prove them to be a considerable alternative to existing methods.[11] These new methods allow the user to adjust the desired degree of computational completeness, trading it of against expected runtime.

References

[BS1] Bronstein-Semendjajew, *Taschenbuch Der Mathematik*, Verlag Nauka, Moskau, BSB B. G. Teubner Verlagsgesellschaft, Leipzig, pp. 159, 1989

[BS2] Bronstein-Semendjajew, *Taschenbuch Der Mathematik*, Verlag Nauka, Moskau, BSB B. G. Teubner Verlagsgesellschaft, Leipzig, pp. 157-158, 1989

[GLS1] G. Gottlob, N. Leone, F. Scarcello, *Hypertree Decompositions and Tractable Queries*, Proc. of the Conference on Database and Expert Systems Applications (DEXA'99), Florence, 1999, LNCS 1677, pp. 1-15, Springer

[GLS2] G. Gottlob, N. Leone, F. Scarcello, *A Comparison of Structural CSP Decomposition Methods*, Artificial Intelligence, 124(2): 243-282, 2000

[KLRK] Vladik Kreinovich, Anatoly Lakeyev, Jiři Rohn and Patrick Kahl, *Computational Complexity And Feasibility Of Data Processing And Interval Computations*, Applied Optimization, Kluwer Academic Publishers, 1998

[MMT] J. Mauss, V. May, M. Tatar, *Towards Model-based Engineering: Failure Analysis with MDS*, ECAI-2000 Workshop on Knowledge-Based Systems for Model-Based Engineering, ECAI 2000, Berlin

[RR] H. Ratschek and J. Rokne, *Computer Methods For The Range Of Functions*, Ellis Horwood series in mathematics and its applications, Ellis Horwood Limited, 1984

[11] $\mathbb{GN}$ has been found to have almost identical runtimes for singleton and interval coefficients.