

TR-0738

複数の制約ソルバーを結合した
非線形不等式制約ソルバー

大木 優、土岐 尚子、磯川 澄江
大平 栄二、新庄 広、川口 俊治
阿部 正博 (口立)

February, 1992

© 1992, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

複数の制約ソルバーを結合した 非線形不等式制約ソルバー

Nonlinear Inequalities Constraint Solver combined multiple constraint solvers

大木優^{*1}, 上岐尚子^{*2}, 磯川澄江^{*2},
大平栄二^{*1}, 新庄広^{*1}, 川口俊治^{*2}, 阿部正博^{*1}

* 1 (株) 日立製作所中央研究所

* 2 (株) 日立情報システムズ

概要

定性推論は、量を定性的に取り扱い、動的な系の状態とその時間的変化を推論する技術である。しかし、量を定性的に取り扱うために、推論結果が曖昧になるという問題が生じていた。本論文では、この問題を解決するために量を不等式で取り扱う非線形不等式制約ソルバーConsortについて述べる。Consortは、定性推論による電子回路の設計支援システムで使用するため量を不等式で扱う以外に、①非線形連立不等式を解くことができること、②負帰還の検出および正帰還とその変化方向の検出ができること、が必要となる。Consortでは、より広い範囲の非線形連立不等式を解くために、区間法による制約ソルバーおよびシンプレックス法による制約ソルバー、グレブナーベースによる制約ソルバーの3つの制約ソルバーを結合し、さらに、それらの間で、新しくわかった変数の値、上限下限値、線形化された等式を交換する。その結果、個々の制約ソルバーが解ける範囲の和よりもより広い範囲の非線形不等式を解くことが可能となった。帰還の解析は、区間法による制約ソルバーで変数の上限値と下限値を計算する過程で行い、正帰還であるかの判定は、上限値と下限値の変化に注目し、同じ変数が発散的に変化しているかどうかを調べることにより行う。発散的に変化している場合は、正帰還と見なし、収束する場合は負帰還と見なす。このようにすることにより、負帰還や正帰還を含む回路の制約も扱うことが可能となった。なお、Consortは並列論理型言語KL1で記述されている。

Keywords : Constraint Solver, Nonlinear Inequalities, Qualitative Reasoning, Feedback

1. まえがき

定性推論は、量を定性的に取り扱い、動的な系の状態とその時間的変化を推論する技術である。これまで、動的な系の挙動の説明、故障診断、あるいは設計などへの応用が試みられてきている〔西田 88a, 西田 88b, 西田 91, 大木 88a〕。しかし、量を正、負、ゼロと定性的に取り扱うために、推論結果が曖昧になるという問題が生じていた〔西田 88a〕。この問題を解決するために、極めて大きい量と極めて小さい量の導入〔西田 85〕、項書き換えシステムによる不等式の利用〔大木 86〕、区間法の利用〔Simmons 86, 綱谷 87〕、量の大きさの程度に関する推論〔Raiman 86, 綱谷 87〕、〔大木 86〕を拡張した非線形不等式制約ソルバーの利用〔大木 88b〕、などの研究が行なわれてきた。〔大木 86〕あるいは〔大木 88b〕では、量の取り扱いに不等式を使うことを提案し、量の定性的取り扱いと定量的取り扱いが可能である。この方法では、定量的情報がわかっている場合は、定性化せずに定量情報をそのまま扱い、定性的情報は不等式を使って表現する。その結果、定量情報がわかっている場合は、それを定性化することによる情報の欠落を防ぐことができ、推論結果が曖昧になることを減らすことができる。

〔大木 86〕では、項書き換えシステムを利用して不等式を解いていたが、〔大木 88b〕では、複数の制約ソルバーを結合し、その間で情報を交換することにより、より広い範囲の非線形不等式を解く不等式制約ソルバーが使われた。

本論文では、〔大木 88b〕で使われていた不等式制約ソルバーの方式を基本にして作られた不等式制約ソルバーConsort (Constraint solver for non linear inequalities) の機能と実現方式について述べる。Consortは、電子回路の設計支援システムDesq (Design support system based on qualitative reasoning) 〔大木 91, Ohki 92a〕の制約ソルバーとして使うことを目的として開発された。Desqは、値を決めたい設計パラメータの値を未定にして、電子回路の構造と回路の動作仕様が与えられると、動作仕様を満たすために必要な設計パラメータの範囲を求める設計支援システムである。Desqによる電子回路の設計支援では、回路のモデルが非線形不等式となったり、回路に帰還が起きるため、制約ソルバーに対して次のような能力が必要となる。

- (1) 非線形連立不等式を解くことができること。
- (2) 負帰還の検出および正帰還とその変化方向の検出ができること。

Consortでは、これら能力を実現するために、以下の機能を開発した。

- (1) 複数の制約ソルバーの結合とそれらの制約ソルバー間での情報交換機能。

区間法による制約ソルバー〔Simmons 86, 綱谷 87〕およびシンプレックス法による制約ソルバー〔今野 87〕、グレブナーベースによる制約ソルバー〔佐々木 86, Aiba 88, Sakai 89, Hawley 91〕を結合し、それらの制約ソルバー間で新しく判明した制約を交換し、単独の制約ソルバーに比べてより広い範囲の非線形不等式を解く。

- (2) 区間法による制約ソルバーにおける上限値と下限値の収束あるいは発散の検出機能。

区間法による制約ソルバーにおいて変数の上限値と下限値の変化を調べ、上

限値あるいは下限値が収束しているか、発散しているかを調べる。収束する場合は、負帰還と判断し、発散する場合は正帰還と判断する。

非線形連立不等式を解く効率的なアルゴリズムは知られていない。そこで、Consortでは、複数の制約ソルバーを結合し、各制約ソルバーで新しく判明した制約の情報交換を行う。その結果、個々の制約ソルバーで取り扱える範囲を単純に合わせたものより、広い範囲の非線形不等式を解くことが可能となった。しかし、3つの制約ソルバーはどれも非線形連立不等式を完全に解くことはできないため、Consortでも非線形不等式を完全に扱うことはできない。

電子回路では、負帰還や正帰還を含むことは特別なことではない。負帰還は、線形不等式の範囲ではシンプレックス法による制約ソルバーで解くことができ、非線形等式の範囲では、グレブナーベースによる制約ソルバーで一部解くことができる（高次方程式を解く必要がある場合は解くことができない）。しかし、シュミットトリガーやマルチバイブレータなどのように正帰還を含む回路を対象にすると、この2つの制約ソルバーは間違った答えを出す可能性がある。一方、区間法による制約ソルバーは非線形不等式を完全には解くことはできないが、変数の上限値あるいは下限値の更新に基づいて非線形不等式を解くため、区間法による制約ソルバーが解ける範囲であるが、負帰還および正帰還を検出することができる。

正帰還の研究の一つとして、因果ストリームによる正帰還の解析方法がある〔西田 8 9〕。因果ストリームによる正帰還の解析は、変数間の因果関係を解析して、フィードバック係数を計算するものである。ただし、取り扱える範囲は線形微分方程式に制限されている。

区間法による制約ソルバーを使った正帰還の検出は、変数の上限値と下限値を計算する過程で同時に正帰還の解析を行う。正帰還の判定は、上限値と下限値の変化に注目し、同じ変数が発散的に変化しているかどうかを調べることにより行う。発散している場合は、正帰還と見なす。本方法は、区間法による制約ソルバーの計算過程で行っているため、非線形不等式で表わされている帰還でも扱うことができる。Desqによる電子回路の設計支援では、電子回路のモデルが非線形になるため、非線形不等式に対して正帰還かどうか判断できることが望まれる。しかし、区間法による制約ソルバーは連立方程式（不等式）を完全に解くことができないので、すべての正帰還を解析することはできない。

本研究は、定性推論の研究の一環で行なわれたが、制約ソルバーの研究としても位置付けることができる。これまで、不等式を扱う制約ソルバーの研究として多くの研究がなされているが、大部分のものは、独立したアルゴリズムを使ったものである〔Jaffar 8 6, 川村 8 7, 大木 8 8 c〕。Consortは、複数の制約ソルバーを結合して、互いに情報交換をしながら、非線形不等式を解く。なお、Consortは、並列推論マシンPIM上で動作可能なように並列論理型言語KL1で記述されている。

本論文では、第2章で定性推論システムにおける制約ソルバーの位置づけを述べ、第3章では3つの制約ソルバーを組み合わせて協調させた制約ソルバーConsortについて述べる。

2. 定性推論システムにおける制約ソルバーの位置づけ

本章では、本制約ソルバーConsortの定性推論システムにおける位置づけを述べる。Consortは定性推論システムを使った設計支援システムDesqにおける非線形不等式制約ソルバーとして使用する。

Desqのシステム構成を図1に示す。Desqは電子回路の構成が決まった後、回路が望ましい挙動をするようにいくつかの回路パラメータを決定する設計支援システムである。Desqは設計したいパラメータが未定義として含まれる回路構成情報を初期データとして入力すると、知識ベースにある回路部品の知識や電子回路に関する物理法則の知識を使いながら、回路のモデルである非線形不等式を生成する。Desqでの定性推論の基本的部分はQupras [大木 86, 大木 88b, Ohki 92b] と同じである。

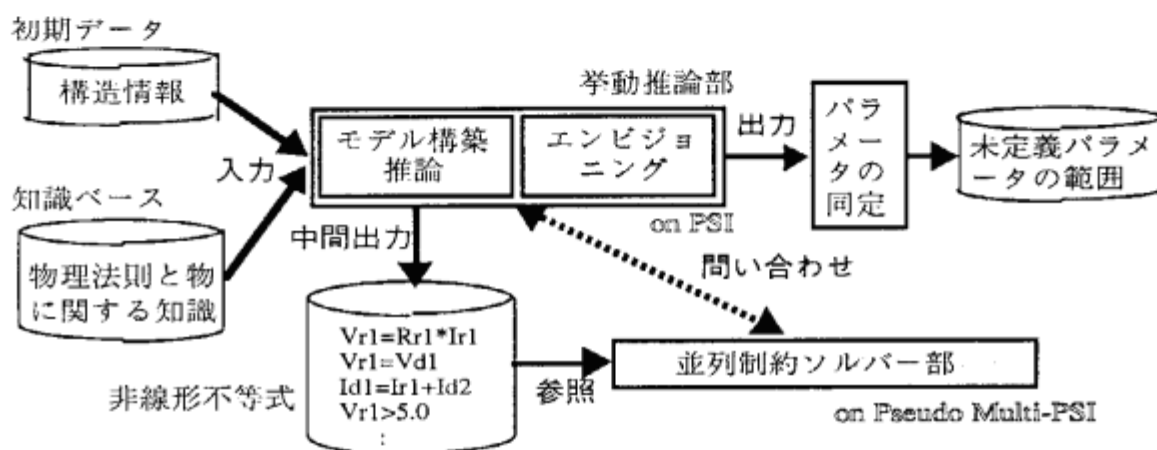


図1 Desqのシステム構成

次に、ダイオードの定義を図2に示す。ダイオードがONである状態の条件（state onのconditions）は、

$$\text{ダイオードのベース電圧 (V@Di)} \geq 0.7$$

である。Consortは、現在の不等式群でこの関係が満足されるか満足されないかを調べる。満足されていることがわかれば、ダイオードがONの状態であることがわかり、ダイオードがONである状態を表わす関係式（state onのrelations以下の式）がConsortに追加される。満足されるか満足されないかが決められない時は、評価は保留され、後で、再度、評価される。Desqによる抵抗値等の設計支援では、モデルが非線形不等式になるため、Consortは非線形不等式を扱う必要がある。

DTLのような電子回路では正帰還が起きないが、シュミットトリガーやマルチバイブレータ等の回路では、正帰還が起きる。正帰還になっている式は、シンプレックス法による制約ソルバーかグレブナーベースによる制約ソルバーで解くことができる可能性がある。しかし、それは、正帰還の挙動とは異なった結果を生成する。例えば、次のような式は正帰還を表わしている式である。

```

object terminal:Terminal
  attributes
    v ;
    i ;
end.

object two_terminal_device:T2D
  parts_of
    t1-terminal ;
    t2-terminal ;
end.

object diode:Di
  supers
    two_terminal_device;
  attributes
    v ;
    i ;
    resistance-constant ;
  initial_relations
    v@Di=v@t1!Di-v@t2!Di ;
  state on
    conditions
      v@Di >= 0.7 ;
    relations
      v@Di= 0.7 ;
      i@Di >= 0.0 ;
  state off
    condition
      v@Di < 0.7 ;
    relations
      resistance@Di=100000.0 ;
      v@Di=resistance@Di*i@Di ;
end.

```

図 2 ダイオードの定義

$$Z > 0.0$$

$$Z = Y + 1.0$$

$$Y = 2.0 * Z$$

これは、シンプレックス法による制約ソルバーでは矛盾となってしまいます。しかし、これは、電子回路での正帰還においては誤りで、上の式はYとZとも正の方向に発散する正帰還を表わしている。Desqでは、正帰還を含む可能性がある回路も対象としているため、Consortでは、正帰還を解析する機能が必要となってくる。

3. 非線形不等式制約ソルバーConsort

3. 1 制約ソルバーの結合

非線形不等式を効率良く解くアルゴリズムがないので、Consortでは、単独の制約ソ

ルバーの解ける範囲を合わせたものより広い範囲の非線形連立不等式を解くことを目的としている。そのために、次の3つの制約ソルバーを結合し、それぞれの制約ソルバーで解けた情報をお互いに交換する。

- (1) 区間法による制約ソルバー (本章では、ICS(Interval Constraint Solver)と呼ぶ。) [Simmons 86, 綱谷 87]

2変数の四則演算に対して2変数の上限値と下限値がわかっている場合、その演算における上限値と下限値を求める区間法の演算規則に従い、変数の上限値と下限値を与えられた式を使って次々に求める制約ソルバーである。非線形不等式を受け付けることができるが、完全に解くことはできない。ある変数の上限と下限を決めることが自分自身の上限と下限を決めることを含んでいるような(変数の依存関係がループになっているような)式は解くことができない場合がある。典型的な例は、 $X + Y = 1$ 、 $X - Y = 3$ のような連立方程式である。

- (2) シンプレックス法による制約ソルバー (本章では、SCS(Simplex Constraint Solver)と呼ぶ。) [今野 87]

シンプレックス法に従って線形不等式を解く制約ソルバーである。線形等式と線形不等式を完全に解くことができる。

- (3) グレブナーベースによる制約ソルバー (本章では、GCS(Grobner based Constraint Solver)と呼ぶ) [佐々木 86, Aiba 88, Sakai 89, Hawley 91]

グレブナーベースによる簡約化によって非線形等式を解く制約ソルバーである。ただし、 $X^2 = 1$ のような高次方程式の根の公式を使う必要があるものは解くことができない。また、不等式は取り扱いえない。

Consortは、図3に示すように、3つの制約ソルバーと制約ソルバー間の情報を交換する制御処理部、トランスレータ部からなる。制御処理部は、3つの制約ソルバー間の情報交換を司る。トランスレータ部は、以下のような規則に従って、入力された式を各制約ソルバーが処理できるような式に変形し、各制約ソルバーに分配する。

- (1) 入力された式が線形等式であれば、そのまま、すべての制約ソルバーに渡す。
- (2) 入力された式が線形不等式であれば、そのまま、ICSとSCSに渡す。
- (3) 入力された式が非線形等式であれば、式の中の個々の非線形部分を独立な変数で置き換えることによって非線形等式を線形化し、その線形等式を、すべての制約ソルバーに渡す。非線形部分と独立な変数の等号関係は、ICSとGCSに渡す。
- (4) 入力された式が非線形不等式であれば、基本的に(3)と同じであるが、線形化した線形不等式が、SCSとICSのみに渡されるのが異なる。

非線形等式と非線形不等式を非線形部と線形部に変形する理由は、制約ソルバー間で非線形の情報交換をさせるためである。制約ソルバー間の情報交換の利点については次に述べる。

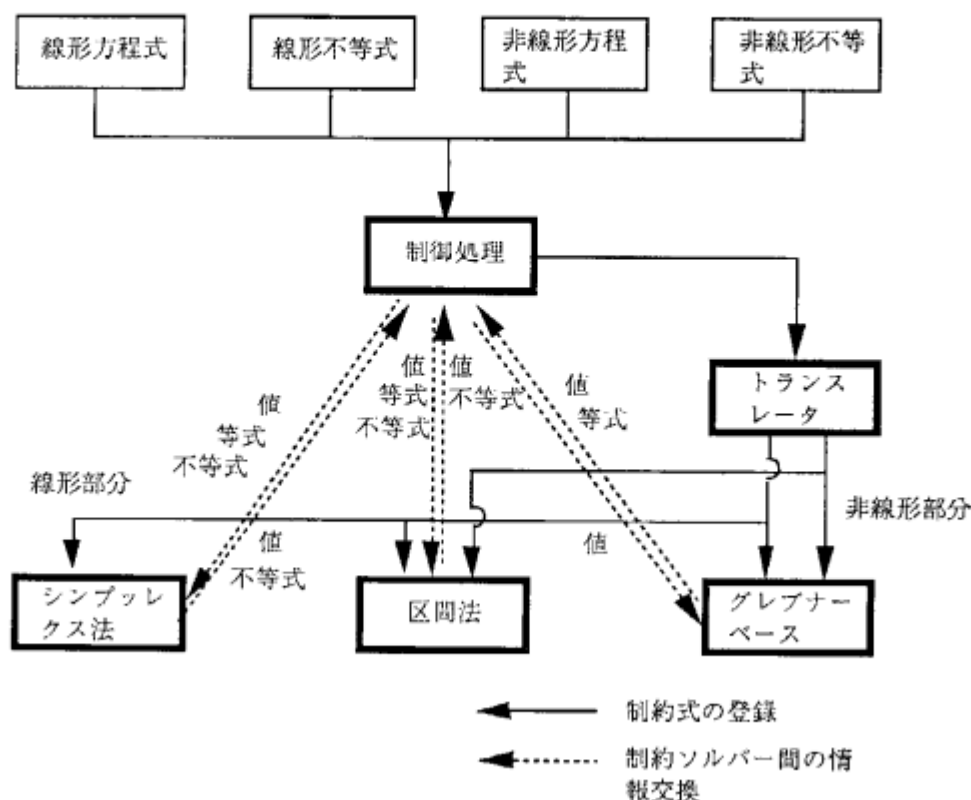


図3 Consortのシステム構成

3. 2 制約ソルバー間の情報交換

3つの制約ソルバー間で交換する情報としては、以下の情報が考えられる。

(1) 新しく求めた値

3つの制約ソルバーは与えられた制約（制約ソルバーに与えられる式を等式か不等式かを区別しない場合、制約と呼ぶ）から新しく値を求めることができる。別の制約ソルバーは、その値を渡されるとさらに制約を解く可能性がある。

(2) 新しく厳しく求めた変数の上限値あるいは下限値

ICSあるいはSCSで変数の上限値あるいは下限値を求めることができる。別の制約ソルバーに上限値あるいは下限値が伝えられと、さらに制約を解く可能性がある。

(3) 簡約された式

SCSとGCSは式を簡約しながら、制約を解く。その過程において、簡約化された式を生成する。別の制約ソルバーは、その式が渡されるとさらに制約を解く可能性がある。

Consortでは、(1)と(2)及び(3)の一部の情報を交換することにした。(3)のすべてを交換すると、制約ソルバーに入力される情報が冗長になる可能性がある。例えば、線形の等式はICSとSCS、GCSに直接入力されるが、GCSでは別な式に簡約

化されることがあり、それがICSやSCSに入力されると冗長な式になり、計算時間が増大する可能性がある。そこで、GCSで変数の同値関係がわかった場合かあるいは非線形部分が線形化された等式のみをICSとSCSに伝える。まとめると、図3に示すように、3つの制約ソルバーの間で、以下のような情報を交換する。

- (1) 変数の値
- (2) 変数の上限値あるいは下限値（不等式として表現する）
- (3) 変数の同値関係あるいは非線形部分が線形化された等式（等式として表現する）

3つの制約ソルバーで交換される情報の流れについて図4を用いて説明する。縦軸と横軸はそれぞれ、制約が等式であるか不等式であるかと制約が非線形であるか線形あるかを分けるものである。以下、交換される情報の流れについて説明する。

- (1) GCSからSCSとICS（それぞれ、図4の(1)と(2)）

非線形不等式の中の非線形の部分がGCSによって線形化されたか変数同士の同値関係がわかったならば、SCSとICSに線形化された情報を渡す。非線形部分がGCSによって線形に変換できる部分を使うことによりSCSやICSはさらに制約を解くことができる。

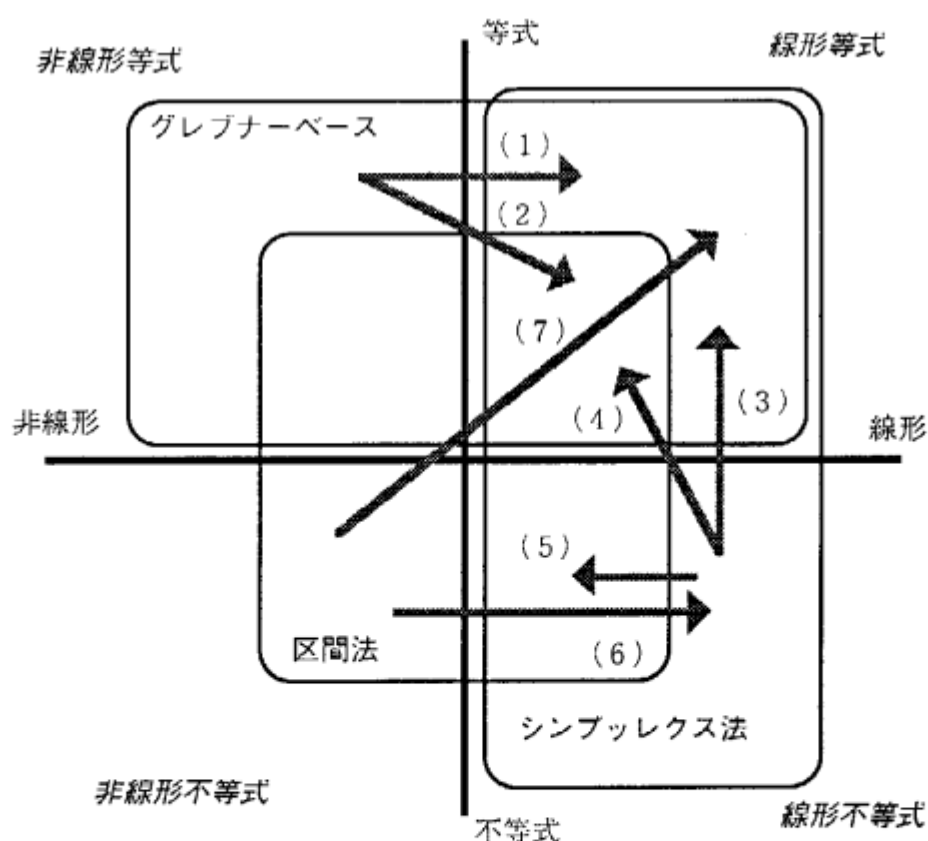


図4 Consortの各ソルバー間の情報交換

(2) SCSからGCSとICS（それぞれ、図4の(3)と(4)）

SCSによって線形不等式（線形等式も含めて）から変数の値が求まった場合はGCSとICSにその値を渡す。

(3) SCSからICS（図4の(5)）

SCSによって変数のより厳しい上限値あるいは下限値が求まった場合は、不等式の形でICSに渡される。これは、変数の依存関係がループになっているような線形連立不等式の場合は、ICSは解けない場合があるが、SCSは解くことができるためである。

(4) ICSからSCS（図4の(6)）

ICSは区間法の演算規則を使って非線形不等式から変数についてより厳しい上限値あるいは下限値を求めることができる場合がある。その場合は、不等式の形でそれらをSCSに渡す。

(5) ICSからGCSとSCS（図4の(7)）

ICSは区間法の演算規則を使って非線形不等式から変数の値を見つけることができる場合がある。その場合は、その値がGCSやSCSに渡される。例えば、以下のような非線形連立不等式から

$$X = Y * Z$$

$$X \geq 1$$

$$Y \leq 1$$

$$Z \leq 1$$

$X = 1$ を見つけることができる。

このように、各ソルバーが新しく判明した情報を他の制約ソルバーに渡し、さらに、その情報を使ってさらに計算を進めることができる。この結果、3つの制約ソルバーが解ける範囲を合わせたものより広い範囲の非線形連立不等式を解くことができる。

図5に単独の制約ソルバーでは解けないが、制約ソルバー間が情報を交換することによって、より厳しい上限値と下限値を求めることができた例について説明する。これは、DesqにおいてDTLの中の一つの抵抗値を設計支援する際に出てきた非線形不等式である。全体の式は非線形であるため、SCSでは扱うことはできない。また、連立不等式を含むため、ICSでは、連立不等式部分の変数の上限値と下限値を求めることができない。しかし、制約ソルバー間で情報を交換すると、図に示すように、2つの制約ソルバー間で新しく求められた変数の上限値や下限値を交換しながら、より厳しい上限値と下限値を求めることができる場合がある。

非線形式を線形式及び非線形部と変数の等式に変換する理由は、GCSが非線形部分を線形等式に変換する可能性があるからである。例えば、

$$V = R1 * I + R2 * I$$

$$V1 = R1 * I$$

$$V1 > 0$$

$$V2 = R2 * I$$

$$V2 > 0$$

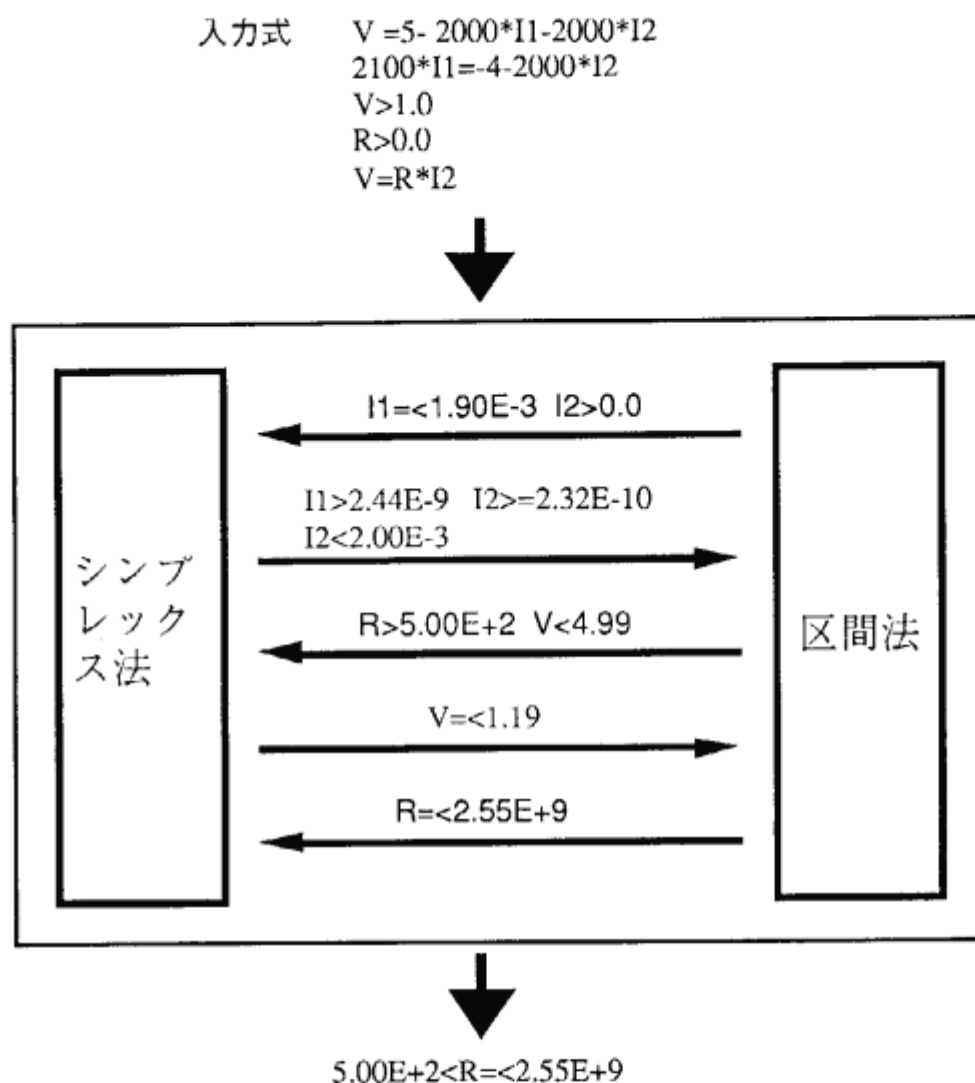


図5 制約ソルバー間の情報交換の例

のような式は，トランスレータ部によって非線形部分がそれぞれ X_1 , X_2 , X_3 , X_4 に置き換えられ，次のように変形される。

$$V = X_1 + X_2$$

$$X_1 = R_1 \cdot I$$

$$X_2 = R_2 \cdot I$$

$$V_1 = X_3$$

$$X_3 = R_1 \cdot I$$

$$V_1 > 0$$

$$V_2 = X_4$$

$$X_4 = R_2 \cdot I$$

これらの式から，Consortは

$$V > 0$$

であることを見つける。これは、次のようにして行われる。まず、GCSが、

$$X4 = -V1 + V$$

であることを見つける。この式は、GCSが簡約した式の中で、非線形部を置き換えた変数を持つ線形な式で、かつGCSに入力された式でないものを選ぶことによって見つけられる。そのような式は、元の非線形の式がGCSによって線形になったものと期待できる。元の非線形式をそのままGCSに入力すると、どれが線形化された式かを判定することが難しいと言う問題が生じる。次に、この式がSCSとICSに渡されて、 $V > 0$ であることが求められる。

3. 3 実現方式

3. 3. 1 Consortの並列実現方式

Consortは並列論理型言語KL1 [ICOT 9 1 a] で記述されている。その構成は図1に示したとおりである。各制約ソルバーは制約が加えられると、以下の処理を並列に実行する。

(1) ICS：制約に影響を受ける変数の上限値と下限値を計算する。

(2) SCS：すべての変数の上限値と下限値を計算する。

(3) GCS：加えられた制約を使って、今まで与えられた制約を簡約化する。

これらの処理の結果、新しく情報が求まったならば、その情報がConsortの制御処理部に伝えられ、残りの制約ソルバーに伝播される。これは、一つの問題の部分問題を色々な解き方で解いていることに対応しており、それぞれの制約ソルバーでは、OR並列で解くことができる。ここでは、Consortで使用した制約ソルバーの中でICSの並列実現方式について述べる。GCSは、ICOTで開発されたものを利用した。SCSはICOTでESP [ICOT 8 8] で開発されたものを基本的にKL1に変更したものを利用した*1。

3. 3. 2 区間法による制約ソルバーの実現方式

区間法による制約ソルバー（ICS）をKL1で開発するに当たって、非線形不等式と変数をオブジェクトとし、メッセージ交換によって変数の上限値と下限値を計算する方式を採用した。これは、KL1のプロセスと適応性が良く、KL1の並列性を生かせると考えられたからである。オブジェクトとして次の2種類のプロセスを作った。

(1) 式プロセス

非線形不等式一つづつごとにプロセスを作り、その式を使って変数の上限値と下限値を計算する。

*1

グレブナーベースによる制約ソルバーとシンプレクス法による制約ソルバーを借用させていただき、また使い方など教えていただいた第4研究室相場室長代理、坂井主任研究員、寺崎研究員、川岸研究員、毛受研究員に感謝いたします。

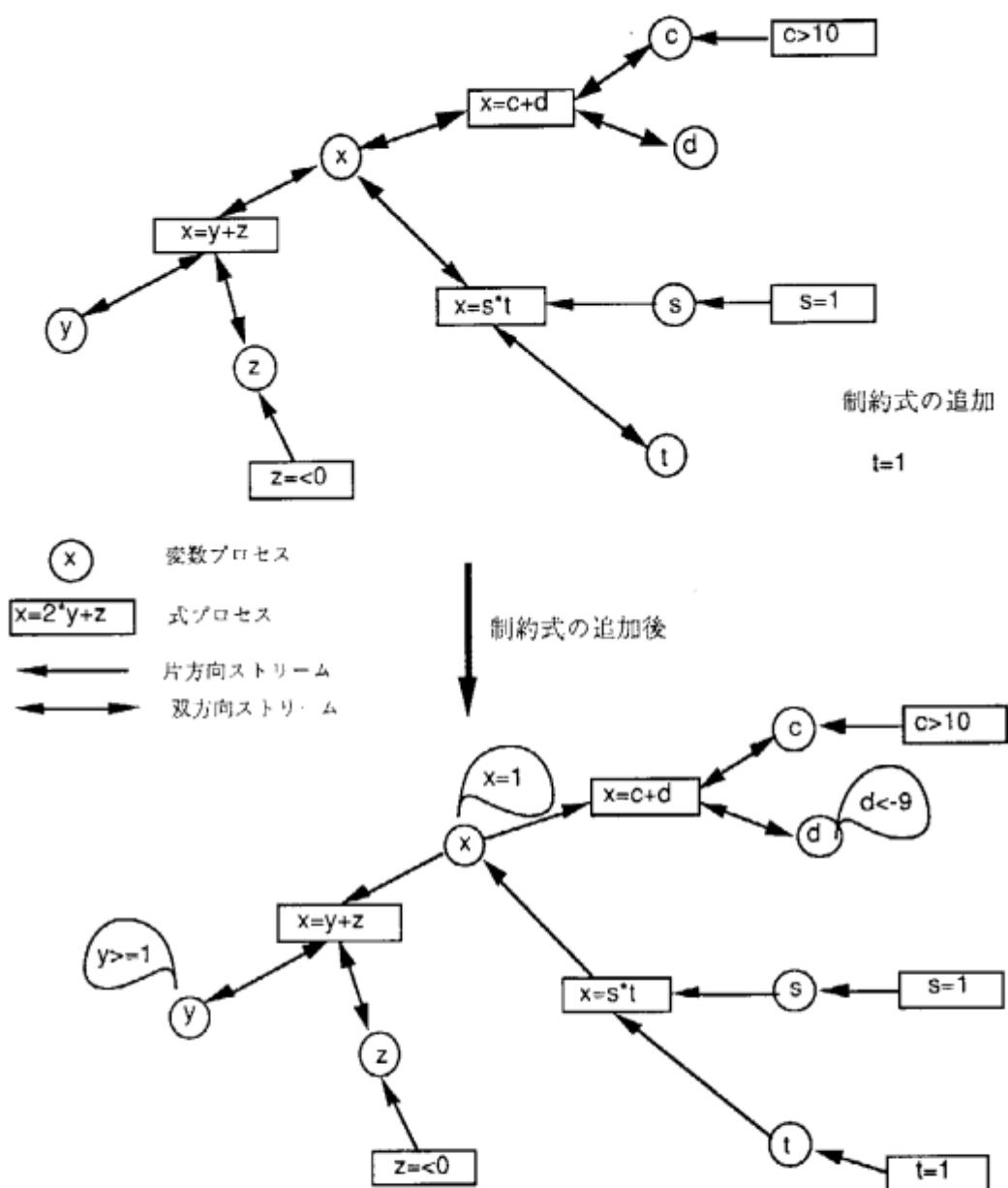


図6 区間法による制約ソルバーの処理概要図

(2) 変数プロセス

変数の上限値と下限値を持ち、それらが更新された場合、関係する式プロセスに変数の上限値と下限値が更新されたことを伝える。

ICSの実行経過を図式化したものを図6に示す。図に示すように、制約（図中では、 $t=1$ ）が加えられると、更新された内容が更新メッセージとして伝えられる。プロセス間のメッセージは一般には双方向であるが、変数の伝播方向が確定した場合は、片方向となる。処理中に矛盾を検知すると、矛盾であることを報告する。

ここで示した方法だけでは、一つの式プロセスに色々なルートからの上限値と下限値の更新メッセージがくる可能性がある。このような場合、後からくる更新メッセー

ジの上限値あるいは下限値の方が前の更新メッセージよりも制約が厳しい可能性がある。このような場合、前の更新メッセージの処理は無意味になり、前の更新メッセージに関する処理を棄却するほうが効率上望ましい。複数の更新メッセージの受信を受け取る例を図7に示す。メッセージ①が原因である更新メッセージ④がプロセス5にきた後に、より厳しい制約（ここでは、上限値あるいは下限値も不等式と見なせるので制約と呼ぶ）を持つ更新メッセージ⑨が来たならば、更新メッセージ④によって生じた⑤から⑧までの更新メッセージの処理は棄却するほうが望ましい。ICSでは、無意味な更新メッセージを棄却するために、処理の棄却フラグをメッセージを送る毎に付加する。

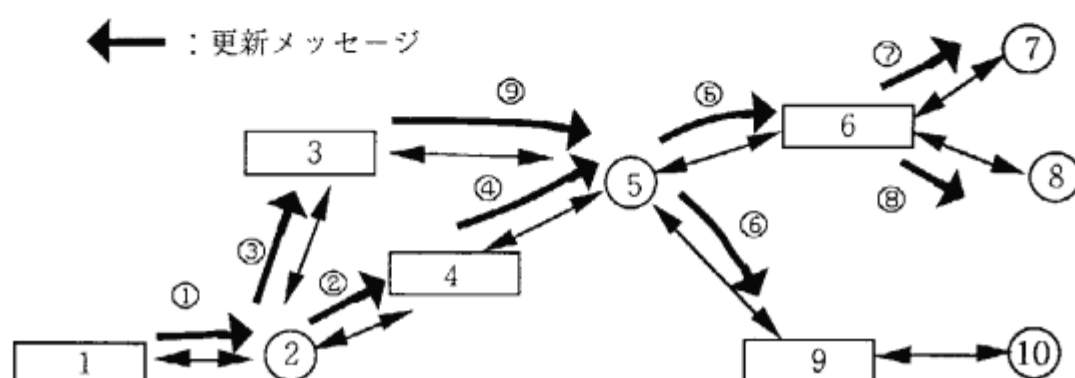


図7 複数の更新メッセージの受信

更新メッセージの形式は、

new_constraint(新しく求まった制約, 棄却フラグ, 履歴)

である。履歴は、同じ変数プロセスに複数回メッセージが送られてきた場合、更新の収束性について調べるために使用する。棄却フラグが設定されていない場合は、更新メッセージの処理を続行し、棄却フラグが設定されている場合は処理を中断する。受け取った更新メッセージを処理した結果、他のプロセスに更新メッセージを送る必要がでてきた場合、単純に、受け取った更新メッセージの棄却フラグを送り出す更新メッセージの棄却フラグにすることはできない。もし、そうしたならば、あるプロセスで棄却フラグを設定すると、それが、そのプロセスから送り出した更新メッセージ以外に、メッセージの流れとは関係なく棄却フラグの設定が伝播されてしまうため、そのプロセスに更新メッセージを送ったプロセスが別なプロセスへ送った更新メッセージも棄却してしまうことになる。この状況を避けるために、受け取った更新メッセージの棄却フラグが設定されたならば、送出した更新メッセージの棄却フラグを棄却に設定する棄却フラグ伝播プロセスを生成する。このプロセスは、KL1で簡単に生成できる。これにより、棄却フラグの設定が更新メッセージの送られた方向しか伝播しないようにすることができる。あるプロセスにより厳しい制約が伝わったならば、そのプロセスから既に送り出した更新メッセージの棄却フラグを設定すると、棄却フラグ

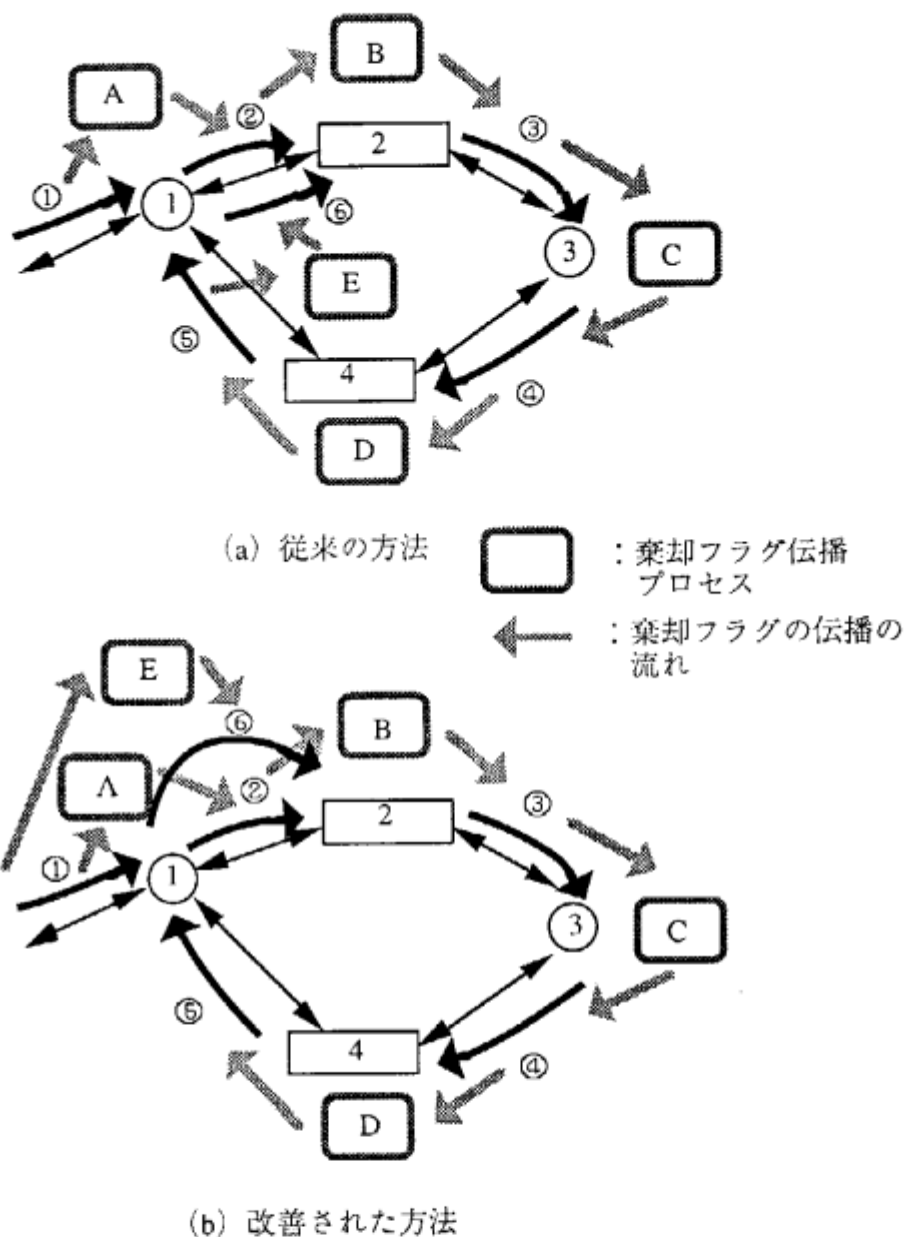


図8 ループがある場合の棄却フラグ伝播プロセスのつながり方

がつぎつぎに伝播され、送り出した更新メッセージによって作られて、まだ処理されていない更新メッセージがあれば、そのメッセージの処理が中断される。

しかし、この方法だけでは、帰還があり、プロセスのネットワークがループになっている場合は、自分自身のメッセージを自分自身で棄却する可能性がある。その例を図8の(a)で示す。(a)では、プロセス1からプロセス4がループになっている。メッセージ①がプロセス1に伝えられたためプロセス1がプロセス2に更新メッセージ②を送出したとすると、メッセージ①の棄却フラグとメッセージ②の棄却フラグが棄却フラグ伝播プロセスAによってつながれる。プロセス1から4までがループして

いるため、更新メッセージ②が原因となった更新メッセージ⑤がプロセス1に伝わる可能性がある。更新メッセージ⑥の制約が更新メッセージ②の制約より厳しい場合は、更新メッセージ②の棄却フラグを設定し、更新メッセージ⑤の制約を基にして、更新メッセージ⑥をプロセス2に転送する。そして、更新メッセージ⑤の棄却フラグと更新メッセージ⑥の棄却フラグをつなぐ棄却フラグ伝播プロセスEを作成する。しかし、更新メッセージ②の棄却フラグが設定されてたため、それが、棄却フラグ伝播プロセスBからDを経由して、棄却フラグ伝播プロセスEに伝わり、更新メッセージ⑥を棄却してしまう。このように必要な更新メッセージまで棄却してしまうことを防ぐために、(b)に示すように、自分が起因した更新メッセージによって棄却フラグ伝播プロセスを作る場合、自分が起因したものではない更新メッセージである更新メッセージ①の棄却フラグと更新メッセージ⑥の棄却フラグを棄却フラグ伝播プロセスEによってつなぎ、更新メッセージ⑥の棄却がそもそもの更新の原因である更新メッセージ①の棄却フラグによって行なわれるようにする。このようにすることにより、ループがあっても、棄却フラグを使った更新メッセージの棄却を行なうことが可能となる。

3. 4 帰還の解析

Consortでは、ICSの処理中に、変数の上限値あるいは下限値の収束性を調べることによって、発散する正帰還か、収束する正帰還あるいは負帰還かの検出を行うことができる。これは、図8の(b)のプロセス1のように、変数プロセス自身が送出した更新メッセージが原因でその変数プロセスに更新メッセージが伝えられたときに、次のようにして調べる。

- (1) 履歴の中に格納されている更新された上限値あるいは下限値が収束していれば、収束する正帰還あるいは負帰還と判断し、収束値を結果とする。
- (2) 更新された上限値あるいは下限値が発散していれば、正帰還と見なし、発散の方向から正帰還による変数の発散方向を求める。結果として、正帰還による発散が起きていることと、その変化方向を報告する。
- (3) 発散していないが、一定の収束範囲に入っていない場合は、新たな更新メッセージを伝播する。ただし、一定回数以上、同じ変数に更新メッセージが伝わっていれば、処理を終了する。結果として、収束しなかったことを返す。

発散は、更新メッセージの履歴の中から連続した更新された上限値あるいは下限値を3つ取り出し、最初の更新値から2番目の更新値への変化を1とした場合、2番目の更新値から3番目の更新値への変化が1.0倍以上であるならば、発散と判断する^{*2}。収束については、更新値の変化が更新値の一定比率以内に入っているならば、収束と判断する。

制御処理部では、ICSの結果が正帰還であれば、GCSやSCSの結果を棄却する。これは、正帰還であれば、GCSやSCSでは、矛盾を検知するか、正帰還による結果と異なる結果を出すことがあるからである。

簡単な非線形不等式で表わされた正帰還による下限値の変化の例を以下に示す。以下のような式が与えられると、

$$Z = Y + X$$

$$Y = R * Z$$

$$R > 2.0$$

$$R < 3.0$$

$$X = 1.0$$

$$Z > 0$$

ICSでは、YとZの下限值が表1のように更新される。表からわかるように、YとZとも更新される度に、下限値が2倍ずつ大きくなっていることがわかり、発散していることがわかる。これから、YとZが正の方向に正帰還で発散していると判断できる。

現在の方法では、ICSで正帰還部分を正帰還と判断できない時点で、先にGCSやSCSが解くことができる場合には、解いた結果をICSに渡してしまうので、ICSで正帰還を見つけられなくなったり、矛盾になってしまったりすることがある。この解決方法の一つに、他の制約ソルバーと情報交換をしない正帰還検知用のICSを別に設けることが考えられる。現在、主に逐次計算機上のPseudo Multi-PSI (Pseudo Multi-Personal Sequential Inference Machine) [ICOT 9 1 b] を使用しているため性能を悪くするた

表1 正帰還内の変数の変化

回数	Yの下限值	Yの変化率	Zの下限值	Zの変化率
1	0		1	
2	2	2	3	2
3	6	4	7	4
4	14	8	15	8
5	30	16	31	16

* 2

次の式をフィードバックを表わしている式と見なし、

$$Z = X + Y$$

$$Y = K * Z$$

時間遅れを考慮すると、

$$Y_n = Y_{n-1} + K^n \Delta X$$

となる。これから $n-2$ 時刻から $n-1$ 時刻への変化と $n-1$ 時刻から n 時刻への変化を求めると、

$$Y_n - Y_{n-1} = K^n \Delta X$$

$$Y_{n-1} - Y_{n-2} = K^{n-1} \Delta X$$

$$(Y_n - Y_{n-1}) / (Y_{n-1} - Y_{n-2}) = K$$

となり、更新された値の差が帰還係数 K となる。 K が 1 以上であれば、 n が無限大になれば Y_n は発散する。

め設けていない，Multi-PSIやPIM（Parallel Inference Machine）[ICOT 9 1 b]などの並列計算機上では，余分な計算機を正帰還検知用のICSに割り振ることが考えられる。

4. むすび

本論文では，電子回路の設計支援システムDesqで用いられるために開発された非線形不等式を解く制約ソルバーConsortについて述べた。Consortは，区間法による制約ソルバーおよびシンプレクス法による制約ソルバー，グレブナーベースによる制約ソルバーを結合し，それらの制約ソルバー間で新しくわかった情報を交換する。その結果，より広い範囲の非線形不等式を解くことが可能となった。そして，区間法による制約ソルバーで上限値あるいは下限値の発散を調べることにより，電子回路の解析でしばしば必要となる負帰還や正帰還の解析が一部可能となった。ただし，現在，Desqでは，Consortの正帰還の解析機能の開発が遅れたため，設計支援推論側で数式処理を使って正帰還の解析処理を行っている[大平 9 2]。

残された課題は，以下のとおりである。

(1) 健全性の向上

非線形不等式を完全に取り扱うことができないために，結果に対する健全性がない。しかし，これは，Desqが設計支援であるため，ConsortがDesqの制約ソルバーとして使われる限り大きな問題とはならないと考えられる。しかし，純粋に制約ソルバーとして使われる時は問題になると思われる。非線形不等式を完全に解くことは難しいので，健全性を保証することは難しい。部分的な健全性の保証を検討する必要がある。

(2) 高速化

シンプレクス法による制約ソルバーは制約が追加される度に，すべての変数を計算している。これは効率的でないため，制約によって関係付けられている変数にグループ化して計算量を減らすことや，通常は制約が追加されてもすべての変数を計算しないで，アプリケーションプログラムが必要なときにすべての変数を計算するような制御機構を組み込むことによって，高速化を図りたい。

(3) 並列性

区間法による制約ソルバーは式プロセスや変数プロセスをネットワークで結合しているため，PIMのような密結合計算機で並列実行しやすい形態と考えている。ただし，Multi-PSIのような疎結合計算機では，計算機間のメッセージの交換が頻繁になるため並列性がでない可能性がある。PIMが一般公開されたならば，並列性の評価を行ないたい。

謝辞

本研究を進めるに当たって，ICOT第7研究室新田克巳室長，市吉伸行室長代理，第4研究室相場亮室長代理，坂井公主任研究員（現在，筑波大学），寺崎智研究員，川岸太郎研究員，毛受哲研究員，および京都大学西田豊明助教授に有益な指導・議論をしていただいたことを感謝いたします。なお，本研究は，新世代コンピュータ技術

開発機構 (ICOT) からの委託により行なったものである。

参考文献

- [Aiba 88] Aiba, A., Sakai, K., Sato Y., and Hawley, D. J.: Constraint Logic Programming Language CAL, pp. 263-276, Proc. of FGCS, ICOT, Tokyo, 1988.
- [Hawley 91] Hawley, D.: The Concurrent Constraint Language GDCC and Its Parallel Constraint Solver, KLI Programming Workshop, pp. 155-168 (1991).
- [ICOT 88] ICOT: 小型化PSI ESP説明書 (SIMPOS 3, 1版), ICOT (1988).
- [ICOT 91a] ICOT: PIMOSマニュアル (第3, 0版) -プログラミング編-, ICOT (1991).
- [ICOT 91b] ICOT: 第五世代コンピュータ研究開発の概要, 第9回第五世代コンピュータに関するシンポジウム, ICOT (1991).
- [Jaffar 86] Jaffar, J. and Michylov, S.: METHODOLOGY and IMPLEMENTATION of a CLP SYSTEM., Proc. of the Fourth International Conference of Logical Programming, pp.196-218, 1986.
- [川村 87] 川村十志夫, 溝口文雄: 制約論理プログラミング言語とその適用例, 日本ソフトウェア科学会第4回大会, D-2-3, pp215-218 (1987).
- [今野 87] 今野浩: 線形計画法, 日科技連 (1987).
- [西田 85] 西田豊明, 川村正, 堂下修司: 定性推論における曖昧性と不連続性の取り扱いについて, 情報処理学会知識情報処理シンポジウム論文集, pp. 67-76 (1985).
- [西田 88a] 西田豊明: 定性推論に関する最近の研究動向 (I) 基礎技術の進歩, 情報処理, Vol. 29, No. 9, pp. 1009-1022 (1988).
- [西田 88b] 西田豊明: 定性推論に関する最近の研究動向 (II) 新しい研究分野・応用, 情報処理, Vol. 29, No. 11, pp. 1322-1333 (1988).
- [西田 89] 西田豊明: 電子回路の定性解析, 定性推論, 溝口文雄, 古川康一, 安西祐一郎共編, 共立出版, pp. 175-206 (1989).
- [西田 91] 西田豊明: 定性推論の考え方とその知的問題解決への応用, 情報処理, Vol.32, No.2, pp. 105-117 (1991).
- [大平 92] 大平栄二, 大木優, 新庄広, 阿部正博: 定性推論を用いた設計支援システムにおける不連続変化を伴う回路のための推論方法の検討, (1992) (電子情報通信学会論文誌投稿中).
- [大木 86] Ohki, M., Furukawa, K.: Towards Qualitative Physics, 1986年度日本認知科学会シンポジウム, 日本認知科学会, pp. 20-24 (1986), あるいはICOT TR-221.
- [大木 88a] 大木優: 定性推論とその応用, 精密工学会誌, Vol.54, No.8, pp. 1418-1422 (1988).
- [大木 88b] 大木優, 藤井裕一, 古川康一: 物理法則に基づいた定性推論, 情報

- 処理学会論文誌, Vol.29, No.7, pp. 694-702(1988).
- [大木 88c] 大木優, 澤本潤, 坂根清和, 藤井裕一: SupInf法に基づいた制約論理型プログラミング言語, 日本ソフトウェア科学会, 第5回大会, pp. 49-52 (1988).
- [大木 91] 大木優, 大平栄二, 新庄広, 阿部正博: 定性推論に基づいた設計支援システム, 第43回情報処理学会全国大会講演集, 4D-9(1991).
- [Ohki 92a] Ohki, M., Oohira, E., Shinjo, H., and Abe, M.: Range Determination of Design Parameters by Qualitative Reasoning and its Application to Electronic Circuits, FGCS'92 (1992) (to appear).
- [Ohki 92b] Ohki, M., Sakane, J., Sawamoto, K., and Fujii, Y.: Enhanced Qualitative Physical Reasoning System: Qupras, New Generation Computing, Vol. 10, pp. 223-253 (1992).
- [Raiman 86] Raiman, O.: Order of Magnitude Reasoning Proc. of AAAI-86, pp. 100-104 (1986).
- [Sakai 89] Sakai, K. and Aiba, A.: CAL: A Theoretical Background of Constraint Logic Programming and its Applications, J of Symbolic Computation 8, (1989).
- [佐々木 86] 佐々木建昭, 元吉文男, 渡辺隼郎, 数式処理システム, 昭晃堂 (1986) .
- [Simmons 86] Simmons, S.: Commonsense Arithmetic Reasoning, Proc. of AAAI-86, pp. 118-128, 1986.
- [網谷 87] 網谷勝俊, 西田豊明, 堂下修司, 量に関する不十分な情報に基づく推論, 情報処理学会知識工学と人工知能研究会52-5(1987).