

複数の制約ソルバーを統合した制約ソルバー

2H-8

大木 優*, 大平栄二*, 新庄 広*, 阿部正博*,
土岐尚子**, 磯川澄江**, 川口俊治**

* (株) 日立製作所 中央研究所, ** (株) 日立情報システムズ

1. はじめに

本報告では、複数の制約ソルバーを統合し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、以下の機能を持つ。

(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。本報告では、この機能の実現方式については省略する。

Consortでは、複数の制約ソルバーで情報交換を行なうため、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。

なお、Consortは、電子回路の設計支援システムDesq (Design support system based on qualitative reasoning) [Ohki 92] 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。

2. 非線形不等式制約ソルバー-Consort

2.1 制約ソルバーの結合

Consortでは、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 (ICS(Interval Constraint Solver)と呼ぶ。) [Simmons 86, 網谷 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。

(2) シンプレックス法による制約ソルバー (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。

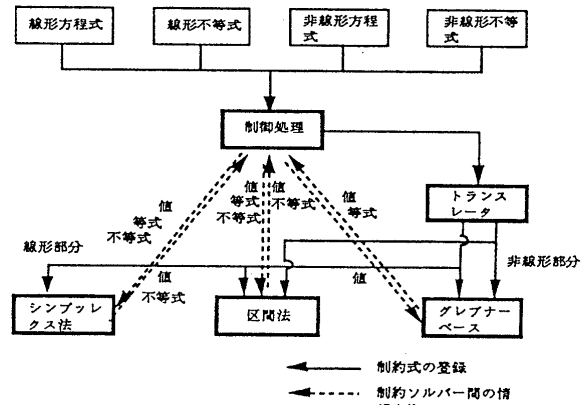


図1 Consortのシステム構成

(3) グレブナーベースによる制約ソルバー (GCS(Grobner based Constraint Solver)と呼ぶ) [Aiba 88]

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、不等式は取り扱えない。

Consortは、図1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。

2.2 制約ソルバー間の情報交換

Consortでは、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)

Nonlinear Inequalities Constraint Solver combined multiple constraint solvers
Masaru OHKI, Eiji OHHIRA, Hiroshi SHINJO, Masahiro ABE
HITACHI, Ltd.
Naoko TOKI, Sumie ISOKAWA, Toshiharu KAWAGUTI
Hitachi Information Systems

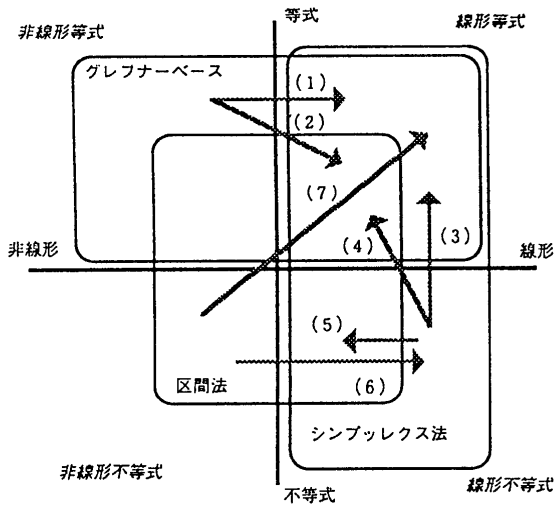


図2 Consortの各ソルバー間の情報交換

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2に示す。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2の(1)と(2))
非線形不等式の中の非線形部分が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。
- (2) SCSからGCSとICS (それぞれ、図2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2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2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2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

$$\begin{aligned} X &= Y * Z \\ X &\geq 1 \\ Y &\leq 1 \\ Z &\leq 1 \end{aligned}$$

$X = 1$ 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。

非線形形式を線形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、

$$\begin{aligned} V &= R1 * I + R2 * I \\ V1 &= R1 * I \\ V1 &> 0 \\ V2 &= R2 * I \\ V2 &> 0 \end{aligned}$$

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 $X1, X2, X3, X4$ に置き換えられ、次のように変形される。

$$\begin{aligned} V &= X1 + X2 \\ X1 &= R1 * I \\ X2 &= R2 * I \\ V1 &= X3 \\ X3 &= R1 * I \\ V1 &> 0 \\ V2 &= X4 \\ X4 &= R2 * I \end{aligned}$$

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、元の非線形式がGCSによって線形になったものと期待できる。一方、元の非線形式をそのままGCSに入力すると、非線形式のどれが線形化された式かを見つけることは難しくなる。線形化された式はSCSとICSに渡されて、 $V > 0$ であることが求められる。

3. おわりに

本報告では、複数の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制約ソルバーConsortについて述べた。

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.
- [今野 87] 今野浩: 線形計画法, 日科技連 (1987).
- [Ohki 92] 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).
- [Simmons 86] Simmons, S.: Commonsense Arithmetic Reasoning, Proc. of AAAI-86, pp. 118-128, 1986.
- [綱谷 87] 綱谷勝俊, 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).