グラフ理論による制約解析および手順生成

永井 保夫
(株) 東芝 情報通信システム技術研究所

生駒 憲治
(財) 新世代コンピュータ技術開発機構

1 はじめに

我々は，制約指向知識コンパイラを適用した設計支援システム MECHANICOT [1] を開発し，知識 (制約) を有効に利用したシステムの構築支援ならびに問題解決の効率化について取り組んできた．しかしながら，現在のコンパイラの制約解析および手順生成処理では問題解決に関する知識の柔軟な記述に対して，十分な対処が困難な場合があり，検討が必要である [2].

本稿では，このような知識 (制約) の解析ならびに手順生成を容易にするために，グラフ表現の導入およびグラフ理論で用いられてきた効率的なアルゴリズムの適用について考察する．具体的には工作機械のギヤ・ユニット設計を例として説明する．なお，このような解析アルゴリズムは，与えられた問題に内在する並列性を抽出し，並列処理の効率化支援に利用することも期待できる．

2 制約解析および手順生成

制約解析は設計対象の制約間の依存関係を解析する．制約解析は，以下の手順に従って行われる．

(1) 制約表現から制約ネットワークを生成し，グラフ (ここでは，特に 2 部グラフ) を用いて表現する．

(2) 設計対象となるシステムが有する階層性に基づいて，グラフ表現された制約ネットワークを分割する．

(3) 分割された制約ネットワークに対して，グラフの構造情報を用いて制約間の依存関係を解析する．

(4) 各々の解析結果を用いて，制約ネットワーク全体の依存関係を解析する．

手順生成では上記の解析結果に基づきより効率的な制約問題解決 (解導出) が可能な手順の生成が行なわれる．具体的な手順生成は以下のように行なわれる．

(1) まず，対象システムの階層構造に従って，つまり，システム全体の依存関係の解析結果からおおまかな手順を決定する．

(2) 次に，各々の分割された制約ネットワークから求められた依存関係グラフに対して特定の問題解決機構を仮定して，方向付けを行ない計算順序を決定する．

(3) 最終的に，仮定された問題解決機構に基づいて実行可能な全体の手順を生成する．

2.1 グラフによる制約および制約ネットワークの表現

制約とは適用対象の構成要素およびその属性間で成立する関係を宣言的に記述したものである．

ここでは制約を表現するために，さまざまな応用分野に対して有用な表現である 2 部グラフを取り上げ，制約解析ならびに手順生成を行なう．なお，グラフ $G=(V,E)$ が 2 部グラフであるとは，V_1，V_2 が共に空集合とならない $V=V_1 \cup V_2$ において，$V_1 \cap V_2 = \phi$ であり，かつ各 $e \in E(G)$ がノード V_1 とノード V_2 につながっていると定義する．

2.2 制約解析

階層化された制約ネットワークにおける制約解析に対処するために，解析フェーズを二つに分割し，それぞれをフェーズ 1 とフェーズ 2 とした．なお，解析は，対象システムの部分・全体関係ならびに継承関係を示したトリー表現に基づき実行される．

フェーズ 1 は，ボトム・アップなアプローチをとり，トリー表現の葉から始まって，根に向かって処理を行う．このフェーズでは，各部品および機能ブロックにおいてグラフの構造情報を用いた依存関係解析ならびに解析情報の簡略化が行われる．

フェーズ 2 は，フェーズ 1 とは逆にトリーの根から葉に向かって処理を行なう．このフェーズでは，トップ・ダウンに，機能ブロック間，機能ブロックと部品間および部品間にまたがる依存関係を再解析することにより，制約ネットワーク全体の依存関係が求められる．

制約解析の処理概要は次のようなアルゴリズムにより与えられる．まず，制約ネットワークを分割し (部品ならびに機能ブロックへの分解に相当)，さらに，分割されたネットワーク (部品ならびに機能ブロック) の依存関係を解析し，最終的に，ネットワーク全体の依存関係を求める．

```
1  procedure main (制約解析)
2    begin
3      while (V ∈ 構成要素かつ V が終端である)
4        dependency_analysis_and_merge(V) ;
5      end;
6      for V ∈ 順序付けされた機能ブロックのリスト do
7        dependency_analysis_for_block (V) ;
8      for V ∈ 設計対象構造のトリー表現 do
9        top_down_dependency_analysis(V) ;
10   end;
```

以下では，特に分割されたネットワークの依存関係解析とネットワーク全体の依存関係解析について述べる．

2.2.1 分割された制約ネットワークの依存関係解析

制約ネットワーク全体の依存関係解析は，対象システムの階層構造に基づき分割された部分問題 (構成要素) に対して行なわれる．

はじめに，与えられた問題を部分問題に分割し，各部分問題をどのような順序で解いていけば最終的に解が求まるかを決定する．ここでは分割された制約ネットワークの依存関係解析を，問題を分割し，分割された部分問題の実行順序を半順序関係として求めることとみなす．

そのために，制約ネットワークを 2 部グラフとして表現し，これを部分グラフに分割し，各グラフの依存関係解析を行なうために DM (Dulmage Mendelsohn) 分解 [4] を導入する．

部品における (制約間) 依存関係解析と簡略化を行なうアルゴリズムは，対応する制約ネットワークを 2 部グラフ表現に変換し，得られた 2 部グラフ $G=(V^+,V^-,E)$ に対して DM 分解を行なうものである．次に，部品に定義された制約と実行文を解析して，2 部グラフ表現に変換する．最後に，生成された 2 部グラフに対して DM 分解を行ない，その結果に基づき依存関係を解析する．

機能ブロックにおける依存関係解析解析と簡略化を行なうアルゴリズムも，部品と同様である．

2.2.2 制約解析によるネットワーク全体の依存関係解析

制約解析のフェーズ 1 では，ボトムアップな処理 (局所的な解析) だけで全体の依存関係を求めた．しかしながら，設計知識 (制約) の与え方によっては，解析が十分に行なわれない場合も考えられる．

Constraint Analysis and Plan Generation Based on Graph Theory

Yasuo NAGAI　Kenji IKOMA
Toshiba Corp.　ICOT

そこで，制約解析のフェーズ2では，フェーズ1によって求められた依存解析結果に対して，システムレベルから部品レベルへとトップダウンに，フェーズ1で十分に解析されなかった大域的な設計知識の再解析を行い，全体のネットワークの依存関係を求める．

2.3 手順生成アルゴリズム

生成される手順が効率化に問題解決されるためには，得られた依存関係から，問題を解くのに必要となるタスクを同定し，その問題に適した問題解決戦略に基づく問題解決器を生成することが必要である．

2.3.1 求められた依存関係グラフのラベル付け

本節では，制約解析で得られた依存関係に対して，問題解決に関する知識を与えてラベル付けする手法について説明する．

手順を生成するためには制約解析で算出された（制約間の）依存関係に対してデータフロー情報を考慮したラベル付けが要求される．このラベル付けは制約間の依存関係を2部グラフ表現したエッジに対して次のような規則を用いて行われる．これらのエッジに対するラベルがデータフロー，すなわち変数の値が決定される順序を示している．

(1) 入力変数であるノードには外向き（出力方向）のエッジだけが存在するようにする．

(2) N変数からなる関係（制約，述語など）に対して，N-1本のエッジが入力方向となるようにし，1本のエッジのみが出力方向となるようにする．

(3) その他の各変数に対して，1本のエッジだけが入力方向に，残りのエッジが出力方向となるようにする．

(4) 問題解決において等式や不等式の役割があらかじめテスターであるとわかっているとき，すべてのエッジが入力方向になるようにする．

依存関係情報に対するラベル付けを上記の規則に従って行うときに，次の2点を考慮することが必要である．

(a) ラベル付けされた依存関係グラフが有効非循環グラフである場合

(b) ラベル付けされた依存関係グラフがループを構成している場合

2.3.2 ラベル付けに基づいた手順生成

ラベル付き依存関係グラフは中間コード形式になっており，最終的には実行可能なコードに変換されねばならない．従来，制約指向知識コンパイラが生成するコードとしてオブジェクト指向の概念を取り入れた論理型言語を設定していた．これは依存関係グラフが (a) のようにラベル付けされ，問題解決機構として生成検査法，局所的な制約伝搬法，問題分割法（分割統治法）を考慮して手順を生成する場合である．

本提案では，依存関係グラフが (a) のようにラベル付けされた場合だけでなく，(b) のような場合について大域的な問題解決機構を考慮した手順生成を対象とする．すなわち，依存関係グラフにおいてループ要素を構成している部分（制約型合に相当する）が等式制約用の問題解決器によって処理可能となるような手順を生成する．

その場合，コンパイラは最終的には制約問題解決器を組み込んだ論理型言語や制約論理型言語からなるコードを手順として生成する．

3 ギア・ユニット設計での制約解析および手順生成

パラメトリック設計は，設計仕様として機能および性能仕様が与えられ，ユーザの要求や制約を満足するように構成要素の属性値を決定する問題とみることができる．我々はギア・ユニット設計 [3] をパラメトリック設計の一種とみなし，以下では，制約指向知識コンパイラにおける制約解析および手順生成の具体例を示す．

3.1 制約解析および手順生成の具体例

はじめに，ギア・ユニットを構成する部品の関係が解析され，Fig. 1 の部分・全体関係を示すトリー表現が生成される．次にこのトリー表現に従って解析順序が決定され，フェーズ1の解析を経て，フェーズ2の解析が行なわれ，処理を終了する．

Fig. 2 は工作機械のギア・ユニット設計における制約解析の中間段階である2部グラフ形式の依存関係に対して，方向を表すラベルが付けられた結果を示しており，この結果に基づいて手順が生成される．

今後の方針としては，上記の制約解析および手順生成アルゴリズムのコンパイラへの実装，実問題での性能評価，ならびに並列化検討をおこなっていく予定である．

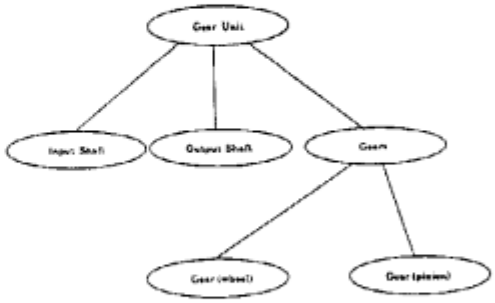

Fig.1 Hierarchical Tree

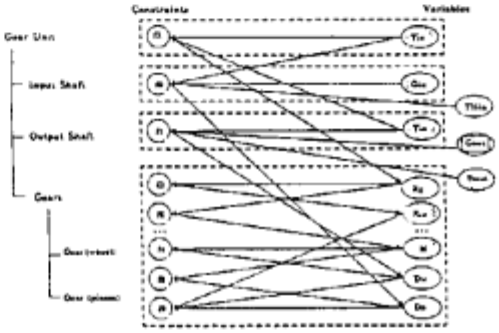

Fig.2 Labeled (Bipartite) Dependency Graph

4 むすび

以上、設計問題向き制約指向知識コンパイラにおいて，特に，グラフ理論に基づいた制約解析および問題解決機構があらかじめ設定されている場合の手順生成のアルゴリズムの概要について述べた．さらに、工作機械のギア・ユニット設計を例として説明を行ない，今後の方針について示した。

謝辞

本研究は第5世代コンピュータプロジェクトの一環として行なわれた．本研究の機会を与えてくださり，常にご指導いただいたICOTの渕一博所長、有益なコメントをいただいた古川康一研究次長ならびに長谷川隆三第一研究室長に深く感謝いたします．また，本研究を行なうにあたり，有益な助言を与えていただいた寺崎智氏をはじめとするICOT第5研究室の皆様に感謝いたします．

参考文献

[1] 永井保夫，尾 寛和，寺崎智，横山孝典，井上克巳：設計問題向けツール・アーキテクチャ，人工知能学会誌，Vol.4 No.3, (1989)

[2] 永井保夫，寺崎智：設計問題向け制約指向知識コンパイラにおける制約解析および手順生成について，1989年度人工知能学会全国大会（第3回），11−43,(1989)

[3] 井上克巳，永井保夫，藤井裕一，今村 聡，小島俊雄：工作機械の設計手法の解析 −設盤の回転機能部品の設計−，ICOT Technical Memorandumn, TM-494, ICOT (1988)

[4] 伊理正夫他，講座・数理計画法 7，グラフ・ネットワーク・マトロイド，産業図書 (1986)