

# 設計問題の制約指向による改良解導出

上田 祐 彰    山 縣 敬 一  
広島大学 総合科学部

設計における改良解の導出を支援するプログラミング環境について検討している。設計に関わる変数や公式により制約ネットワークが構成され、設計者は設計意図に沿って変数値の変更を指示する。システムは、制約を充足するようにパラメータの変更を行い、設計者の意図を反映させることを試みる。充足解が見つからなければ、充足できない状況をシステムは提示する。このような設計過程でのシステムの役割について、提案を行なっている。

Obtaining Revised Solutions in Designing Process  
Using Constraint Network

Hiroaki Ueda and Keiichi Yamagata  
Faculty of Integrated Arts and Sciences, Hiroshima University  
Higashisenda, Nakaku, Hiroshima 730, JAPAN

This paper discusses a programming environment which supports designers to find revised solutions of design problems. A constraint network is constructed in the computer system by using variables and formulas necessary for a designed object. A designer indicates value changes of variables according to his intention. The system supports the designer by executing trials of parameter modification on the constraints. The roles and functions of the system in this interactive process are proposed.

## 1. はじめに

いろいろな設計の分野において、コンピュータ内部に形状を初めとして多様な属性を付加したモデルを構築し、それを介してマンマシンインタフェースをとり、また、解析プログラムもモデルに対して働かせる手法が確立しつつある。本研究で取り上げているのは、たとえ不満足なものであってもコンピュータ内部にモデルが存在するとき、その設計対象を設計者の意図するものにパラメータを変更して行く改良解の導出問題である。現在の状況においては、これを数理的な側面と人の持つ認識と判断能力との融合としてとらえる必要がある。本報告では、改良解の導出を制約ネットワーク上の充足問題として処理することにより1), 2), 解の探索過程の中に設計者の意図をどの様に反映させるかを検討している。本研究の課題は、より一般的なエキスパートシステム的环境中で議論すべきである3), 4)。また、制約充足の意味では、制約論理プログラミングの枠組みも考えるべきであるが、統合的な支援環境はまだ未完成であり、ここでは述べていない。以下で、2章においては考え方の背景を述べ、3章においては簡単な例で制約ネットワークに対して設計者の意図を反映させる様子を、そして、4章で設計者の意図に沿った変数値変更の方策を説明する。

## 2. 設計型の問題解決

一般に、設計問題において、暫定的なモデルが作られた後、改良しなければならない点が見つかったり、あるいは、より望ましい解をさらに見い出したいとき、パラメータをどの様に動かして行けばよいかというシンセシスの指針は、容易には得られない。このことは、非線形で変数の数が非常に多くなってくると、対象の大域的な性質がほとんど見えなくなってしまうこと、並びに、設計者の意図も多目的であって、簡単に定式化できないことによる。数理計画法の側からは、実際的な問題解決のための目標計画法や妥協計画法によるアプローチがある5)。また、エキスパートシステムによる設計支援では、ルールの適用によってこの問題を克服しようとしているが、一般的な解法を用意するほどにルールを整えることは困難である。制約充足によって解を導く手法は、数理計画法に比べて、離散的な制約条件が記述し易いとか探索の過程を制御できる利点はあるが、それ自身特別な方策を持っているわけではない。ここにいろいろな探索の手法を持ち込むとすれば、それは結局制約充足と数理計画法を結び付けることに他ならない。

本研究における制約充足型の設計支援システムは、素朴な形態をとっている。改良解の導出を完全に自動化すると言うよりは、あくまでも設計者の意図に沿って解の導出を支援する形式のものである。設計者の意図は、直接変更したい変数の値、あるいは増減の範囲を指定することによってシステムに伝えられる。システムは、設計者の意図に沿って一定の規則で変数値の変更を進めて行くが、充足解が見つからないと、何が充足できないかを提示して、設計者に別の入力を促す。設計者は、この手がかりから、いくらかでも設計対象についての性質を読みとることができるかも知れない。全体的な設計作業は、これの繰り返しで進められる。このような過程は、現実に実務に携わっている設計者が実際にとっている設計過程について、コンピュータによる支援を一段強化したものである。しかし、よりよい解を求めて行くことの判断は、設計者によっている。

### 3. システム概要

#### 3.1 解導出の流れ

一般に、複雑な問題の解あるいは改良解の導出は、既知の変数から未知の変数を推測し、また仮定されている値をいろいろ動かして全体の状況を把握するという、いわば試行錯誤的手法で行われる。その際、人は自分の持っている知識、経験を頼りに変数の決定、及びその値の推論を行っている。我々が開発しているシステムも、設計者の経験からくる要求を未知変数の推論、改良解導出の手がかりとして使用している。このため本システムは、図1に示すように、暫定解の導出、設計者の要求に基づくリファインを繰り返すことで満足解を得ている。

次節では各プロセスの詳細を示す。

#### 3.2 制約モデルの構築

制約式を逆ポーランド記法に変換し、その過程で制約ネットワークモデルをシステム内に構築する。制約ネットワークは、制約充足問題の数式モデルであり、制約を伝播する空間を記述したものである。制約ネットワークは、設計式を表す数式node、設計変数を表す変数node、設計式に含まれる変数nodeとその数式nodeをリンクするedgeの3つから構成される。図3は水圧シリンダ設計問題6) (図2) を制約ネットワークでモデル化したものである。

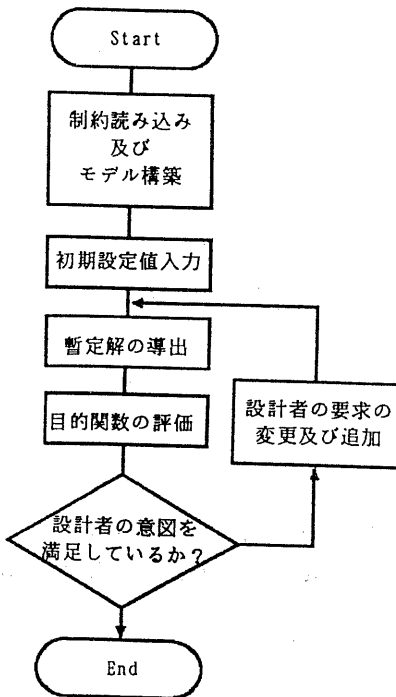
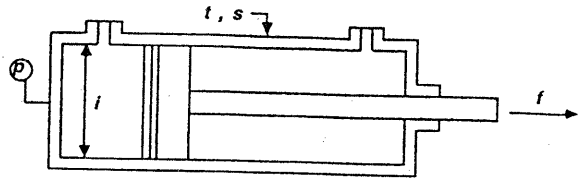


図1 フローチャート



$g1: f_{min} - f < 0$                        $f$ : output force  
 $g2: t_{min} - t < 0$                        $t$ : thickness  
 $g3: p - p_{max} < 0$                        $p$ : inside pressure  
 $g4: s - s_{max} < 0$                        $s$ : hoop stress  
 $h1: f - \pi i^2 p / 4 = 0$                        $i$ : inside diameter  
 $h2: s - ip / 2t = 0$   
 $f, p, s, i, t > 0$

図2 水圧シリンダ設計問題

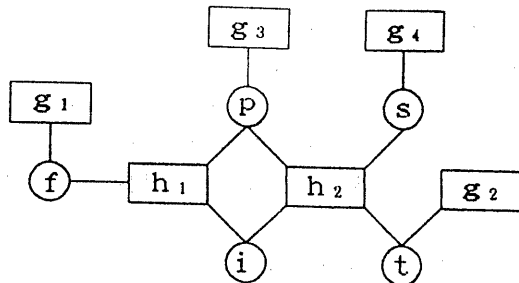


図3 制約ネットワーク

### 3. 3 制約伝播

初期設定値から未決定変数の値を決定するプロセスである。本システムでは、ネットワーク上で値の決定した変数の値を隣接する数式nodeに伝播させ、それにより導出可能になった変数の値を計算することによって制約伝播を行う。そして、この制約伝播が矛盾なくすべての制約に対して行われたとき、解が得られたことになる。制約伝播は等式を使用し、不等式は解の実行可能性の判定のみに用いる。

ここでは、簡単な機械設計の例を挙げたが、ここで制約伝播と設計意図の反映との対応を述べておく。コンピュータは、この例の数式の意味を知らないが、設計者は通常定性的な変数の意味を理解している。この例でいえば、設計者が明確に意識している変数は、出力としての力  $f$ 、大きさに対応する内部半径  $i$ 、材料に対応する応力  $s$ 、であろう。内部圧力  $p$  は、設計者の意図と直接結び付いてはいないと思われる。いま、各変数に暫定的な値が付与されているものとして、形状を小さくするため、内部半径  $i$  により小さい値を指示したとする。このとき、同じ出力を取り出すためには内部圧力を上げなければならない。そうすると応力  $s$  が材質に耐えられるかどうかが問題になる。また別の場合として、材質を落として許容応力  $s$  を小さくしたとき、期待した出力が取り出せるかどうかを調べたいこともある。

簡単な場合には設計者はこの様な伝播の振舞いを直感的に理解しているが、複雑な対象では不明になる。コンピュータによる制約伝播の機能は、この様な状況を支援してくれる事になる。また、同一の制約ネットワークがいろいろな目的に利用できることがわかる。一方において、コンピュータは上に言葉で述べたような物理的な意味は一切理解していないので、総合的に設計意図を実現して行く過程は設計者の判断に頼らなければならない。これは、機械設計の例であるが、一般にいろいろな分野の設計問題において、類似の考え方が成り立つ場合が多数あると思われる。次の4節では、一般的に通用すると思われる改良解導出の手順を説明する。

## 4. 改良解導出の過程

### 4. 1 設計者の要求

本システムで重要なのは、設計者の意図にそって改良解を求めることである。初期解または暫定解が得られると、設計者の要求の指標とするため、システムは解とともに目的関数、設計者の要求する等式及び不等式の増減表を提示する。その情報から、設計者は、過去の経験、知識を駆使し、以下の要求をシステムに入力する。

(1)ある変数の値を指定。

(2)ある変数の増減を指定。

ここで与えられた要求を満足することによって、設計者の希望する設計解を計算する。

(1)で得られた要求は新たな初期設定値、(2)で得られた要求は制約伝播を制御するために用いる。次節では(2)の使用法の詳細を述べる。

### 4. 2 暫定値と設計者の要求

新たな初期値として利用する(1)は全ての変数値を一意に決定しうるほど完全とは限らない。むしろ改良解導出の初期過程では(2)による要求がほとんどである。しかし、

制約伝播は等式を中心に行っており、等式は次数（式に含まれる未決定変数の個数）が1以下にならないと評価されないため、初期値が少ないと途中で制約伝播が止まってしまう。これは初期解導出の段階でもいえることだが、未決定変数に暫定値を与えることで制約伝播を再開させなければならない。その際（2）の要求を用い、設計者から増減の指定があった変数にその要求を満たす値を暫定値として与えることで、設計者の要求を制約伝播に反映する。設計者の指定する変数がなくなった場合は、最も多くの式に含まれる変数を選択し、前回の得られた解もしくは不等式制約を満たす値を暫定値として与えることにする。また、制約伝播の再開と同時に暫定値を与えたnode (a) から再び制約伝播がstopするまでに制約が伝播したnodeとegdeで構成されるネットワークを依存グラフとして記録する。図4に示す制約ネットワークに暫定値がX, Y, Zの順で与えられたとき、記録される依存グラフはA, B, Cとなる。

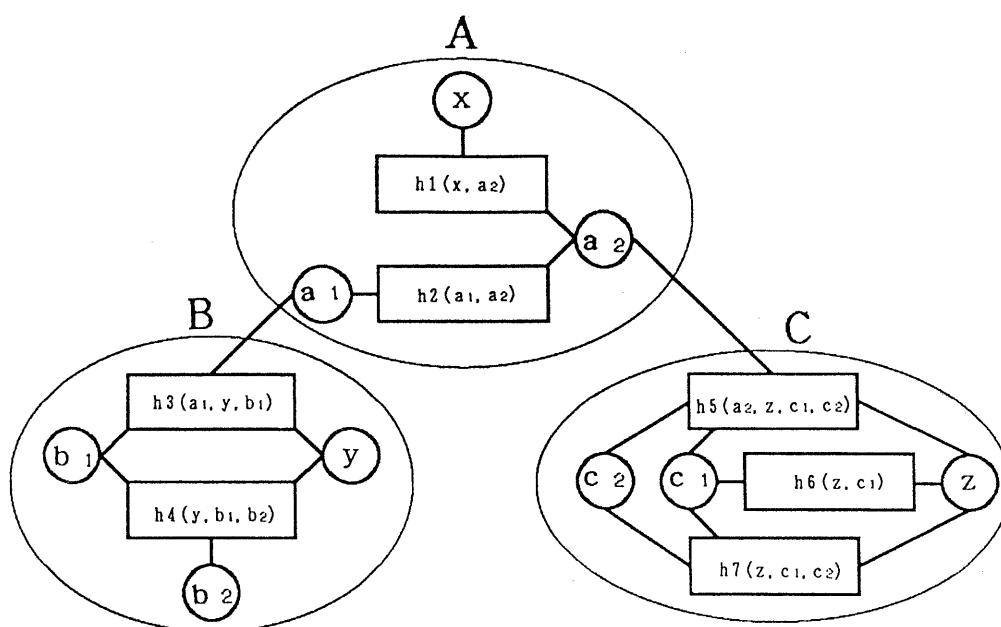


図4 依存グラフ

#### 4.3 制約不充足の解消

暫定値は必ずしも解空間に含まれているとは限らないので、制約不充足が起こり得る。また、制約伝播を行う段階で設計者の要求と異なる値が計算されることもある。制約不充足を解消するには、その原因となった暫定値を変更しなければならない。以下、変更すべき変数及び新たな暫定値の決定アルゴリズムについて述べる。

- [1] 制約不充足をおこした式を含む依存グラフを依存グラフ群とする。
- [2] 依存グラフ群で暫定値の変更を行っていない変数を選択し、前回の改良解探索で得られた解を新しい暫定値とする。
- [3] 新しい暫定値を与えた変数から制約伝播を再開し、それが不充足となった式に及ぼす影響を調べる。その際、不充足の改善につながる変数値の増減を記録する。

[4] 不充足の改善に必要な増減方向と設計者の増減要求が一致すれば、その方向に暫定値を変更し、依存グラフ全体の充足をはかる。

[5] 暫定値を変更しても制約不充足が解消されない、もしくは、不充足を解消する暫定値の変更が設計者の要求と異なる場合は、上位の依存グラフをその依存グラフ群に結合する。

[6] 上位の依存グラフは現在暫定値変更を行っている依存グラフ群に隣接し、その依存グラフ群より以前に形成されたもののうち最も新しく形成されたものを選択する。

例えば、図4の制約ネットワークの、変数 $x, y, z$ に暫定値がこの順序で与えられ、サブグラフC内で制約不充足が生じると、Cを依存グラフ群とし、Cで暫定値を与えた変数 $z$ の値を変更し、不充足を解消するために必要な $z$ の増減を計算する。不充足を解消するために必要な $z$ の増減が設計者の要求に反するとき、もしくは $z$ を変更してもC内の不充足が解消されないときは、上位依存グラフにあたるAと現在の依存グラフ群であるCを結合したものを新たな依存グラフ群とし、その中で暫定値を与えた変数 $x$ を変更して依存グラフ群全体の充足をはかる。

#### 4. 4 要求の変更

設計者の要求同志が互いに相反するとき、すなわち設計者の要求が解空間に含まれないときは、前節の方法で制約不充足を解消することはできない。設計者の要求に矛盾が存在すれば、制約不充足の解消段階で依存グラフ群に結合させるべき上位依存グラフがなくなるため、設計者の要求に沿った制約不充足の解消ができなくなるからである。この場合、前節で計算した制約不充足を解消する為に必要な暫定値増減方向と設計者の要求する変数増減方向が異なり、依存グラフ群内の変数に対して与えられている要求が制約不充足の原因となっている。システムは制約不充足の原因となる変数を設計者に提示し、設計者に要求の変更を求める。設計者はそれに基づき、新たな変更要求をシステムに入力し、その指定のあった変数から再び制約伝播を再開するすることで、要求の矛盾を解消する。

#### 5. おわりに

対話形式により設計者の要求を解探索に取り入れ、その要求に沿った改良解を導く設計支援システムについて述べた。本システムはまだ研究段階であるので、非常に原始的な形態をしており、不満な点も多々ある。しかし、多くのデータを蓄積することにより、より洗練されたシステムを構築することを目標としている。

#### 参考文献

- 1) 岸 義樹：ネットワークによる仕様充足問題解決の一方法，精密工学会誌，Vol. 57, No. 4, pp. 711-717 (1991).
- 2) Wm Leler : Constraint Programming Languages, Addison-Wesley (1988).
- 3) 長澤 勲，古川由美子：拘束条件リダクション法を用いた機械設計計算支援システム，情報処理学会論文誌，Vol. 27, No. 1, pp. 112-120 (1986).
- 4) 中島裕生，田原一徳，加藤 亨：機械設計支援エキスパートシステム構築シェル：MAGIC，情報処理学会論文誌，Vol. 31, No. 11, pp. 1572-1579 (1990).

5)馬場則夫, 坂和正敏: 数理計画法入門 4 章多目的計画法, 共立出版(1989).

6)A. M. Agogino and A. S. Almgren : Techniques for Integrating Qualitative Reasoning and Symbolic Computation in Engineering Optimization, Eng. Opt., Vol.12, p p. 117-135 (1987).