

3C-1

知識ベースシステムによる多目的組合せ問題の解法

鶴島 彰

長谷川高志

斎藤宗昭

セコム IS 研究所 人工知能研究室

1 はじめに

数理計画法の分野では、一部の設計問題を組合せ最適化問題として、目的関数と制約条件からなる数学モデルを使って解いてきた。しかし、数学モデルは、定性的情報の表現が困難、設計変更等によるモデルの変更に対処できない等の問題があり、必ずしも設計問題向きではない。一方、知識工学では、柔軟で記述力の高い記号モデルを使った設計問題の解決が研究されている。しかし、設計においてはしばしば重要となる設計者の主観を含んだ目的については、これまで陽に扱ってこなかった。そこで我々は、多目的計画法と知識ベースシステムを組み合わせる方法により、設計問題の一つである警備機器選択配置問題の解決 [1] を試みる。

2 設計問題のモデル化

設計は、制約条件の下で、ある機能、ある目的を達成するような構成要素の組合せを決定する問題である。設計問題は、明確な解法が存在せず、意味のある状態空間の設定が困難かつ膨大、客観的な解の評価法がない、等の理由から代表的な悪構造問題として知られている。設計がある機能や目的の達成を目指した作業であることを考えると、設計者の中に解を評価するなんらかの視点が存在し、その視点から解を順序づけられるはずである。設計者の持つ目的は一般に複数存在するため、目的間に競合が起こることがあり、設計者は主観的な判断によってこの競合を解消している。我々は、この目的を問題記述の一部として扱う。

ここで以後議論する問題の範囲を、設計問題のうち、有限の要素の組合せとしてとらえることができ、定量的な評価が可能なものとする。さらに、制約条件という言葉、満たさなければならない義務的な制約の意味で使い、目的という言葉、ある評価項目に対し、こうあって欲しいと思う設計者の主観的な願望の意味で使う。

今、設計対象のある記述を表わす状態 s が与えられており、その記述が取り得る全ての状態の集合を S とする。設計に於て考慮しなければならない制約条件を C で表わし、 s が C を満足することを $C(s)$ と書く。制約条件を満たす状態の集合を $A = \{s \mid C(s), s \in S\}$ とし、 $a \in A$ を代替案と呼ぶ。ここで、設計の評価項目数を m とし、ある代替案を評価する写像を $\phi: A \rightarrow R^m$ とする。 $\phi(a) = (v_1(a), v_2(a), \dots, v_m(a))$ (この m 項ベクトルを便宜的に p で表わす) とし、 $v_i(a)$ は a の、 i 番目の評価項目に対する評価関数を表わすものとする。 p の全体を評価空間 P と呼ぶ。決定主体が各評価項目を最大化したいと考えたとすると、全ての i に対して $v_i(e) \leq v_i(e')$ かつ少なくともひとつの i に対して $v_i(e) < v_i(e')$ となるような $e' \in A$ が存在しないとき $e \in A$ をパレート最適解と呼ぶ。 A のうち、パレート最適解のみからなり、かつ全てのパレート最適解を含む集合を $E(\subseteq A)$ とする。さらに、 P から R への写像で、設計者の選好順序を実数の大小関係に置き換えるような関数 u を考える。設計問題を、 A の中から最大の $u(p)$ を持つようなひとつの要素 (e^* と呼ぶ) を求めることと定義する。(図 1)

3 設計問題解決過程

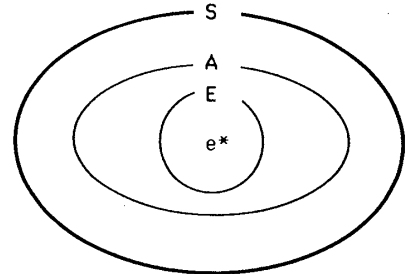


図 1 設計問題

設計問題は、 $S \rightarrow A \rightarrow E \rightarrow e^*$ というプロセスで解ける。これは、制約条件によって全ての状態 S を制約条件を満たす A に絞り、さらにパレート最適性テストによってパレート最適解の集合 E に絞り、その中から決定主体 (e.g. 設計者) の選好情報によって e^* を選ぶことを意味する。設計問題では、可能な状態の数が膨大なため、集合全体を絞って行くような方法は、特殊な場合を除いてとることはできない。我々が考えなければならないのは、代替案を構築しながら最も好ましいと思われる解を探索するような手続きである。

4 知識ベースシステムによるアプローチ

以上述べてきた問題を知識ベースシステムにより解くことを考える。ある状態を他の状態に変化させるようなヒューリスティックな手続きを h 、 h の適用条件を t とする。 t と h のペア、 $r = (t, h)$ をプロダクションルール、状態 s をワーキングメモリの内容としたプロダクションシステムを考える。設計問題を解くことは、初期状態 s_0 に一連のプロダクションルール、 $r_1, r_2, r_3, \dots$ を適用することによりワーキングメモリの内容を e^* に書き換えることである。 e^* の条件は、制約条件 C を満たしていること、パレート最適解であることさらに $u(p)$ が最大であること ($u(p)$ は定義域上で non-decreasing とする) である。ここで、 $u(p)$ の存在に対して次のふたつの場合に分けて考える。

4.1 $u(p)$ が明示的に与えられる場合

決定主体の大局的な選好情報である $u(p)$ が明示的に与えられている場合は、初期状態 s_0 から直接 e^* を求める。この場合、制約条件を満たさせるような構成要素の部分的な組合せを決定する手続きをプロダクションルールとした前向きプロダクションシステムを考える。つまり、初期状態に順次制約条件を満たすような部分的な組合せを加えて行くことによって解を構築する。このとき、どの組合せを受け入れどの組合せを無視するか、どういう順番で組合せを加えて行くかという推論の制御を行なうのが $u(p)$ であり、常に $u(p)$ を最大化する競合解消戦略を取ることで、決定主体の選好情報を反映した設計を行なう。

4.2 $u(p)$ の大ざっぱな推定 $u'(p)$ が与えられ、かつ決定主体とのインタラクションが可能な場合

効用理論に基づく方法において問題となるのが、決定主体の選好情報を正確に反映するような効用関数の決定が困難なことである。そこで、 $u(p)$ があらかじめ明示的に与えられない場合は、その大ざっぱな推定 $u'(p)$ を利用して近似的な解を求め、それをインタラクションによって改良して行く方法を取る。

*An Approach for Solving Multiple Objective Combinatorial Problems Using Knowledge Based Systems
Akira TSURUSHIMA, Takashi HASEGAWA, Muneaki SAITOU
SECOM IS Laboratory, Artificial Intelligence Dept.

s_0 から e^* を導く過程を、 s_0 から e を導く過程と e から e^* を導く過程のふたつに分けて考える。前者を構築過程、後者を改善過程と呼ぶことにする(図2)。我々は、それぞれの過程に対応した二種類の知識ベースシステムによるアプローチを考える。

4.2.1 構築過程

構築過程では、制約条件を満たし、パレート最適解を構築する。ここで、パレート最適解は相対的な概念であり、ある解がパレート最適であるかどうかは、その解が属する集合全体によってのみ決まる。初めに集合全体が与えられている場合は、パレート最適でない要素を全て除くことによってパレート最適解の集合を求めることが出来る。しかし、我々の問題のように順次代替案を生成しなければならない場合は、これまで生成してきた部分集合の中でのみパレート最適性が言えるに過ぎない。

4.1 で述べた知識ベースシステムをそのまま利用することによって、構築過程を実行する。このとき、推論の制御を行なう $u(p)$ が利用できないので、 $u'(p)$ をその代用として競合解消に使う。 $u'(p)$ を得ることも難しい場合は、各評価関数の重み付け総和

$$u'(p) = \sum_{i=1}^m w_i v_i(p) \quad (w_i > 0, i = 1, 2, \dots, m)$$

の最大化を競合解消戦略に取る。こうして得られた解は、Kuhn-Tucker 条件によって、少なくともパレート最適である。

4.2.2 改善過程

初期値として与えられた代替案を、なんらかの視点から改良、改善するのが改善過程である。この過程では、選考情報を与えることの出来る決定主体とのインタラクションが可能であることを前提とする。ここでは、既に存在する代替案の一部を変更することにより、別の代替案を生成するようなヒューリスティック手続きをプロダクションルールとして利用する。このプロダクションルールを次々に適用することにより、初期解のいろいろな部分に変更を加え、様々な代替案を生成する。

解改善過程は、概念的には評価空間 P においてパレート最適解が存在する境界面 (efficient frontier) 上の探索である。そのため、代替案の生成においてそれがパレート最適であるかどうか調べる必要がある。

我々はいくつかの条件を考慮して、対話型多目的 Gradient-Based-Method[3] に基づいた探索法による知識ベースシステムを試みている。この方法は、決定主体と対話により局所的に u を推定しながら探索を続ける。多目的 Gradient-Based-Method は、準凹形 (Quasi-Concave) な効用関数の存在を暗示的に仮定し、さらに決定主体が一对の代替案に対して選考関係とその強さを告げられることを仮定する。このアル

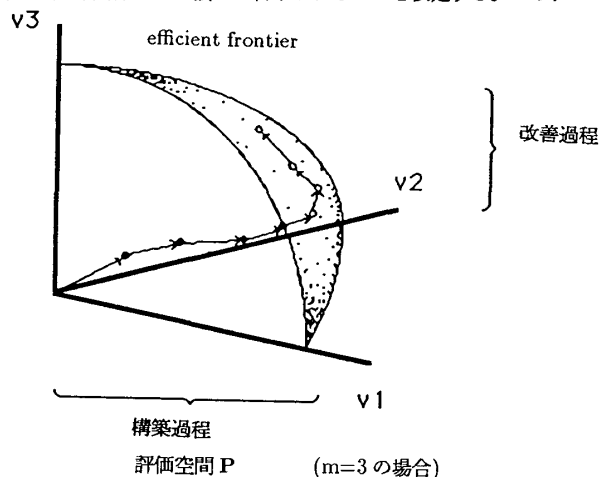


図2 構築過程と改善過程

ゴリズムは、非線形計画法の Steepest Ascent 法を多目的の場合に拡張したものであり、direction-finding 問題を解くことによって、探索の方向 (効用関数の Gradient) を求め、step-size 問題を解くことによって探索の距離を決定する。尚、改善過程では、効用関数を明示的に扱っていないので決定主体とのインタラクションを通じてふたつの問題の解を推定する。以下に、Gradient-Based-Method に基づいた推論の過程を簡単に述べる。

1. direction-finding 問題

- (a) ルールをワーキングメモリに適用し、現在の状態の一部のみを変更してできる代替案を生成するルールと事実の組を全て求める。
- (b) 各組に対し、複数の評価項目からそれが実行された場合得られる代替案を定量評価する。
- (c) 上で評価された代替案の内、パレート最適にならないものを全て取り除く。
- (d) 残った代替案を決定主体に提示し、ふたつ一組にして選考順序関係と選考の強さを聞き出す。
- (e) direction-finding 問題を解き [2, 3]、Gradient ベクトルを求め、各項を各評価項目の重みに使った、効用関数の局所的な線形近似式を求める。

2. step-size 問題

- (a) direction-finding 問題で得られた線形近似式を最大化することを競合解消戦略としたプロダクションシステムを考える。
- (b) このプロダクションシステムの各ステップで生成される代替案を、決定主体に提示し、最も好ましい状態を選ばせる。

このプロセスを順次繰り返すことにより、決定主体の望む解に近づけて行くことが出来る。このプロセスの停止条件は新たに生成されるどの代替案よりも、初めの代替案の方が好ましい場合である。尚、決定主体の判断は常に評価項目に基づくものであるとする。

5 おわりに

目的と制約条件によって問題を記述し、多目的計画法のアルゴリズムを推論制御に使った知識ベースシステムについて述べた。この方法は、設計問題のうち決定者の意志に関する部分を目的、外的な状況に依存する部分を制約条件として整理することにより問題を記述する。目的として扱いづらい示唆的な制約条件をどうするか等、いくつか問題を残すが、ある種の設計問題に対しては有効な方法である。

参考文献

- [1] 鶴島彰 斎藤宗昭, “機器選択配置エキスパートシステムの設計”, 情報処理学会第 37 回全国大会 (昭和 63 年後期)
- [2] B. Malakooti. “Assessment through strength of preference”, *Large Scale Systems: Theory and Applications*, vol. 8, pp.169-182, 1985
- [3] B. Malakooti and G. D'Souza. “Multiple objective programming for the quadratic assignment problem”, *International Journal of Production Research*, vol 25, no. 2, pp.285-300, 1987