

# Walsh 変換を用いた遺伝的プログラミング手法に基づく ファジィ交渉エージェント

松 村 幸 輝†

本稿では、ファジィ交渉エージェントの重要な構成要素であるメンバーシップ関数の最適化を目的として、進化システムの一つである遺伝的プログラミング手法を用いた自動生成システムについて検討している。具体的には、標準的に用いられる三角形の直線関数を基にして、その形状から進化させていくことによって最適メンバーシップ関数にチューニングする方法を提案した。この新規性は、基になる関数を Walsh 変換することによってスペクトル分析し、特徴抽出することによって進化過程を効率化する方法にある。この結果、ファジィ制御に必要な特徴を保ちながら、より事例に適合したメンバーシップ関数を創成することが可能となった。また、具体的な交渉支援の事例として、協調して自動車を購入する場合に合意を得るための交渉に応用することにより、この方法の有用性について検討した。

## Negotiation Support Agent Based on Fuzzy Decision Making by Genetic Programming with Walsh Transformation

KOKI MATSUMURA†

This paper describes a negotiation agent system based on the fuzzy decision making. The method of seeking appropriate membership functions and a reasonable agreement point was examined by means of the genetic programming technique with Walsh transformation, which is an intelligent principle. As a result, the following findings were obtained. This technique helps discover practicable membership functions in a vast search area, and achieve the solution search with high efficiency. This technique is also considered to be applied to the negotiation support easily.

### 1. はじめに

ファジィ制御に基づく交渉エージェントでは、メンバーシップ関数とルールはその制御精度を左右する重要な構成要素となる。一般に、ルールとしては、現象やモデルに適合すると考えられる数種類の制御規則を表したものが、またメンバーシップ関数としては三角形の一次関数が用いられることが多い。特に、このメンバーシップ関数は、最良ではないとしても、シンプルでかつ、ある程度信頼性もあるため、あまり精度を要求されない場合に標準的に用いられる。ところが、より精度の高いファジィ制御を実現させるためには、その事例に適合したルールおよびメンバーシップ関数を系統的に作り出すとともに、適切なものにチューンアップする必要がある。しかしながら、ルールおよびメンバーシップ関数ともに、その仕上がりは勘と経験など設計者の熟達度によるところが大きく、一般に最適なものを見出すことは容易ではない<sup>1)</sup>。

このことから、前稿で多くのファジィルールの中から事例に適合した最適ルールを効率的に見出す方法として、遺伝的アルゴリズム（以下、GA）を適用した方法を提案した<sup>2)</sup>。一方、メンバーシップ関数については多様な形態を持つため、GA の基本手続きだけではメンバーシップ関数の最適化には困難をとまう。

そのため本研究では、より適切なメンバーシップ関数を創成することを目的として、進化システムの一つである遺伝的プログラミング手法<sup>3)</sup>（以下、GP）を用いた自動生成システムについて検討している<sup>4)</sup>。ただ、メンバーシップ関数として形状的に優れるとともに、ファジィ制御の原理に基づいて有効に機能しうるものを作り出すことは簡単ではない。

そこで、本稿では、上記したような三角形の直線関数を基にして、その形状から進化させていくことによって最適メンバーシップ関数にチューニングすることを試みた。このための有効な手続きとして、Walsh 変換<sup>5)</sup>を用いて、基になる直線関数を直交変換することによってスペクトル分析し、その特徴的な要素を抽出することによって進化過程を効率化した。Walsh 変換を用いたのは、デジタル変換としての有用性によ

† 鳥取大学工学部

Faculty of Engineering, Tottori University

り理論的解析に効果的に適用することができる<sup>6)</sup> こと、処理手順が比較的容易で計算量が格段に少なくなること、さらに基になる関数が直線的であるので計算誤差が少ないことによる。そして、これによって得られた Walsh 関数の各係数 (Walsh 係数) を二分木構造の式の木で表現し、この木構造を遺伝的プログラミングによって進化させることによって、メンバーシップ関数の最適化を実現するのである。これにより、ファジィ制御に必要な特徴を保ちながら、より事例に適合したメンバーシップ関数を探索することが可能となる。

実際にこの手法を、ファジィ意思決定に基づく交渉エージェントの知能原理に適応してメンバーシップ関数の創成と交渉妥結案の提示を試み、有用性を検討した。このファジィ交渉エージェントは、意思決定の対象に対する選好の変動のあいまいさを考慮して、交渉相手の動きや自分の固執の度合いに応じて譲歩量を決定するという働きをするため、交渉支援のための有用な意思決定システムとして機能する<sup>1),2),7)</sup>。このことから、交渉過程において最も重要な行為である意思決定をエージェントに行わせるため、交渉者の内面的状態をできるだけ忠実に再現できるように効用理論に基づき効用関数で表現することを試みた。これによって、エージェントは交渉者の意思を反映した行動をとり、交渉者双方が互いに満足するような妥結案をシミュレーションにより導き出し、交渉支援に有益な解決案を提案することが期待できるのである。

具体的な交渉支援のための事例としては、協調して自動車を購入する場合に合意を得るための交渉<sup>8)</sup>を取り上げ、この場合の交渉支援のためのエージェントの構築と合意点の探索の方法を述べるとともに、この方法の有用性について検討する。

## 2. メンバーシップ関数の最適化

メンバーシップ関数として、図 1 に示すような三角形形状の関数が、簡便であるため一般によく使われる<sup>4)</sup>。この三角形形状のメンバーシップ関数を基に、以下に示す Walsh-Fourier 変換 (以下、Walsh 変換) によって各関数の係数 (Walsh-Fourier 係数、以下、Walsh 係数) を算出する。そして、これを GP によって進化させることによって事例に適したメンバーシップ関数に最適化する。

### 2.1 Walsh 変換

定義により Walsh 関数は、

$$\psi_k(x) = (-1)^{\langle k, x \rangle} \quad (1)$$

で表される<sup>5)</sup>。ここで、

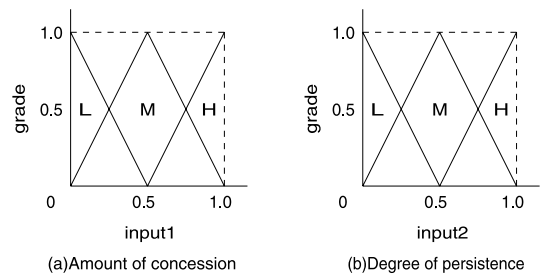


図 1 メンバーシップ関数の例

Fig. 1 Example of membership function.

$$\langle k, x \rangle = \sum_{i=0}^p k_i x_i \quad (2)$$

である。そして、区間  $[0, 1]$  上で定義された関数  $f$  を離散化するため、 $n \in \mathbf{N}$  ( $\mathbf{N} = \{1, 2, 3, \dots\}$ ) として、 $2^{-n}$  間隔でサンプリングし、 $f$  のサンプル値関数  $g_n$  を求める。

$$g_n(k) = f(k \cdot 2^{-n}) \quad (0 \leq k < 2^n) \quad (3)$$

これにより、サンプル値関数  $g_n$  の Walsh 変換は、上述の Walsh 関数を用いて、

$$\begin{aligned} \hat{g}_n(j) &= \frac{1}{2^n} \sum_{k=0}^{2^n-1} g_n(k) \psi_j \left( \frac{k}{2^n} \right) \\ &= \frac{1}{2^n} \sum_{k=0}^{2^n-1} f \left( \frac{k}{2^n} \right) \psi_j \left( \frac{k}{2^n} \right) \end{aligned} \quad (4)$$

で示される。ここで、 $\{k/2^n, 0 \leq k < 2^n\}$  は区間  $[0, 1]$  を等間隔に分割する点列である。このようにして得られた  $\hat{g}_n$  は、もとの関数  $f$  の Walsh-Fourier 係数 (Walsh 係数) の近似値となる。なお、サンプリング回数  $2^n$  を大きくすることによって、近似の精度も上がる。この係数  $\hat{g}_n$  を、次節以降で示すメンバーシップ関数の構成要素とする。

また、この係数から基の関数を求めるには、以下の Walsh 逆変換を用いる。

$$\tilde{g}(k) = \sum_{j=0}^{2^n-1} \hat{g}(j) \psi_k \left( \frac{j}{2^n} \right) \quad (0 \leq j < 2^n) \quad (5)$$

### 2.2 メンバーシップ関数のプログラム個体表現

GP を適用して生成するメンバーシップ関数は、後述するファジィ意思決定の場合では、相手の譲歩量と自分の固執の度合いを表す 2 つの入力関数とする。出力用のメンバーシップ関数は、図 1 と同型の簡略化された三角形形状の関数とし、2 つの入力に対して共用するものとした。なお、図中の L, M, H はそれぞれ、度合いが「少ない」を表す low 関数、「中程度」を表す middle 関数、「多い」を表す high 関数を示してい

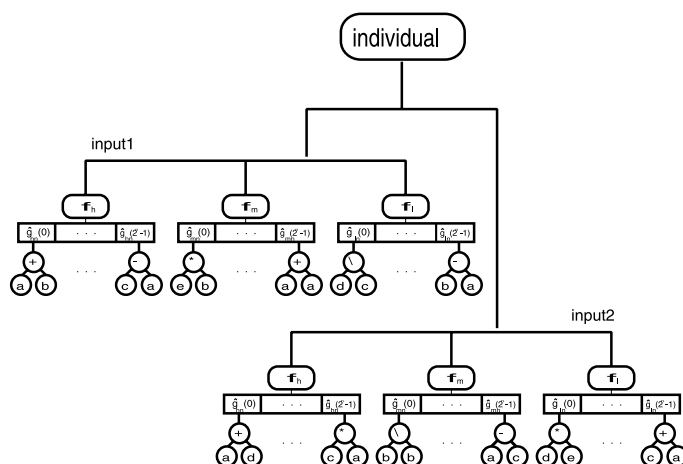


図 2 個体の構造

Fig. 2 Structure of individual.

る。よって、low 関数、middle 関数、high 関数を入力 2 系統分、計 6 つのメンバーシップ関数を 1 つの個体を含めることとする。

実際の GP 実行時の個体構造は、図 2 に示すように、6 つのメンバーシップ関数をルートノードの直下に並列に配し、さらに 1 つのメンバーシップ関数は複数の係数（Walsh 係数として本稿では 16 個）を表現する 16 本の式の木（二分木）を擁したものと扱う。係数を表す式の木ノードに割付け可能な要素としては、関数には四則演算子（+、-、\*、/）、端子には 0.0 から 9.0 までの 1.0 刻みの実数定数を用いることとする。ただし、除算については、実行中に 0 による除算が発生した場合の例外処理として返値に 1 を返す保護除算を用いた。

初期の木は図 2 に示されるように、ルートノードと 2 つの子ノードで構成された 2 階層の単純な二分木であり、これらの式の木に対して GP の遺伝操作が施され、進化過程で複雑な階層構造の木に成長する。また、本手法は、「はじめに」でも述べたように、三角形のメンバーシップ関数を基にして進化させるものなので、初期の式の木は、三角形の直線関数を Walsh 変換して得られる Walsh 係数に近い値になるように設定する。そして、GP による式を木の生成アルゴリズムとして、木構造の簡素化と処理の効率化のため、生成された式の木から実際に計算して得られる値を 100 で割った値が、その式の木として割り当てられるように設計した。

### 2.3 有効な個体の条件

探索の対象となる個体は、上述したような形式で扱うため、あらゆるブランチの構造とノードの組合せ

が可能であり、その探索領域は広大なものになる。しかし、個体はその構成要素としてメンバーシップ関数を擁するので、メンバーシップ関数が有する最低限の特徴は、探索条件に付加すべきと考えられる。これによって、探索の範囲を有効な範囲に制限できる。

その方法として、メンバーシップ関数の豊富な表現力を有する概形を探索条件に取り入れる。そのため、関数の基本的な性質として、low 関数が右下がり、middle 関数が凸型、high 関数が右上がりの形状となっていることが条件となり、そうでないものはメンバーシップ関数として低く評価する。このような関数としての形状の変化は、後節（2.5 節）に記述するように、メンバーシップ関数の要素である式の木に強く依存しており、式の木のある有効な範囲内にあるときに上述の条件を満たすことになる。

このことから、まず、メンバーシップ関数の要素である 16 個の式の木を評価する。そして、式の木が適切な範囲内にあるという条件の下で、初期集団の生成および遺伝操作による子孫の生成を進めていく。

具体的な作業として、初期集団生成時には、各々のメンバーシップ関数を構成する 16 個の式の木がそれぞれ Walsh 係数（後述の表 5 参照）に近い値になるように作成する。そのため、1 つの木を作成するのに、ランダムにノードを割り付ける作業をそれぞれ 20 回行い、そのうちで、対応する Walsh 係数の値に最も近い値を持った式の木を選定した。

また、遺伝操作による子孫生成時も同様に、交叉や突然変異によって生じた式の木が、上述したようなメンバーシップ関数としての基本的な性質に基づく概形と大きく相違するような要素となる場合、すなわち後

述する適切な値の範囲(表1)から外れた値を持つようなものである場合は、再度、交叉や突然変異の操作を行う。なお、このような操作を10回を限度に行った。大部分はこの10回の操作で上述の条件を満たす適切な木が生成されるが、希に、条件から外れる場合がある。そのときはその範囲に最も近い木を選定するようにしたが、このような木を要素に持つメンバーシップ関数は、全体の1%にも達しないが、次の関数の概形に関する評価でペナルティが課されて低く評価され、自然に淘汰される結果となる。

次に、関数の概形に関する評価を次のように行った。なお、下記の $i$  ( $i = 1, 2$ ) は入力関数1あるいは2のいずれかの入力系統番号を表す。

- (i) high 関数の場合、 $0 \leq x \leq 1$  の範囲で単調増加が望ましいので、 $x$  座標の最も小さいサンプル点から走査を始め、それまでに出現している最大値  $\max$  以上とならない場合には、所定の値のペナルティを課す。
  - (1)  $0 \leq g(x) - \max$  のとき、ペナルティなし、
  - (2)  $-0.05 \leq g(x) - \max < 0$  のとき、これを満たすサンプル点  $n_{hi1}$  個について1個につきペナルティ  $e_{hi1}$  だけ加点する、
  - (3)  $g(x) - \max < -0.05$  のとき、これを満たすサンプル点  $n_{hi2}$  個について1個につきペナルティ  $e_{hi2}$  だけ加点。
- (ii) low 関数の場合も同様に、 $0 \leq x \leq 1$  の範囲で単調減少が望ましいので、それまでに出現している最小値  $\min$  以下とならない場合には、所定の値のペナルティを課す。
  - (1)  $0 \leq \min - g(x)$  のとき、ペナルティなし、
  - (2)  $-0.05 \leq \min - g(x) < 0$  のとき、これを満たすサンプル点  $n_{li1}$  個について1個につきペナルティ  $e_{li1}$  だけ加点、
  - (3)  $\min - g(x) < -0.05$  のとき、これを満たすサンプル点  $n_{li2}$  個について1個につきペナルティ  $e_{li2}$  だけ加点。
- (iii) middle 関数の場合は、極大値が1つだけ存在するものが最適であるとし、そうでない場合は山と谷の数をそれぞれカウントして、(山の数 + 谷の数 - 1) の数  $n_{mi}$  だけのペナルティ  $e_{mi}$  を加点する。なお、山の数、連続する3点が上に凸の形状となっている数とし、また谷の数も同様にして調べた。

このように概形に関する評価においてその最良値は0で、最悪値は単調増加あるいは単調減少から外れたサンプル点数によって決まることになる。したがって、

6つのメンバーシップ関数から構成される1個体の総合的な評価値  $f_1$  は、それぞれの関数についての概形に関する評価値から、次式で計算される。

$$f_1 = \sum_{i=1}^2 \left\{ \sum_{j=1}^2 (e_{hi j} \times n_{hi j} + e_{li j} \times n_{li j}) + e_{mi} \times n_{mi} \right\} \quad (6)$$

ここで得られた概形に関する評価値  $f_1$  は、次章で求める事例に関する評価値  $f_2$  と合わせて、総合的な適応度  $F$  を算出する。なお、 $e_{hi1}$ ,  $e_{li1}$  および  $e_{hi2}$ ,  $e_{li2}$ ,  $e_{mi}$  の値は、 $f_2$  とのバランスを考慮して、それぞれ  $\frac{1}{300}$  および  $\frac{1}{150}$  程度の値とした。

## 2.4 基本的アルゴリズム

最適メンバーシップ関数の導出のための基本的アルゴリズムを以下に記述する。

- (i) 三角形メンバーシップ関数の Walsh 係数を求め、その係数1つ1つに対し、その係数の近似値を表すような式の木(二分木)を、2.2節で述べた手順に従って作成する。
- (ii) (i) の木の中から、各係数の式の木を1つずつ選び、係数16個分を束ねたものを1関数とする。これを6つ作成し、図2に示すように、それらの組を1個体とする。この操作を所定の個体数分だけ繰り返し、初期集団を生成する。
- (iii) 各関数の持つ式の木の値を用いて Walsh 逆変換をし、関数を復元し、概形に関する評価値を算出する。
- (iv) 各個体に対して具体的な応用事例(たとえば、後述する妥結点探索のための交渉モデル)に適用し、交渉結果に関する評価を行うとともに、総合的な適応度を求める。
- (v) ルーレット選択法により遺伝操作を施す対象となる個体を選び、式の木を表す二分木に対して、交叉、突然変異を行う。この場合、交叉回数は集団の個体数と同じ数だけ、また突然変異率は10%の個体数だけ、それぞれ行うものとする。そして、新たに生成された個体について、上述した概形に関する評価と後述する交渉結果の評価から総合的な適応度を算出する。なお、親個体と子個体はその世代が終わるまで淘汰させることなく残しておき、最後にエリート保存法でもとの親個体と同じ個体数だけ上位から選択して、次世代の集団を構成する。
- (vi) 手順(v)で得られた個体を次の世代の集団とし、この操作を繰り返し行う。

各遺伝操作は、親となる個体が選択されると、それに内包される 6 つのメンバーシップ関数に対して、同じ種類の関数の同じ係数に対応する木どうしについて施される。たとえば交叉の場合では、同じ入力系統で同じ種類の関数 (L, M, H) のメンバーシップ関数どうしで行われ、操作を受ける式の木はそれぞれの入力系統でランダムに選ばれる。この操作を 1 つの関数に対して 3 回行い、式の木を最大 3 つ変更するものとした。

このようにして作られた次の世代の集団は、淘汰により前の世代より全体的により適した集団へと進化していく。そして、これが繰り返し行われ、後述する適応度が十分大きな値に収束したときに、最適メンバーシップ関数が見出されるのである。

## 2.5 Walsh 係数の条件

2.3 節および 2.4 節で述べた式の木の数値の範囲について考察する。

Walsh 関数は、容易に Hadamard 行列の Kronecker 積によって再帰的に導出することができ、変換も基の関数との 1 あるいは  $-1$  の積に基づいて簡単に計算できる。しかも、変換の対象とする基になる関数が図 1 に示すような直線的な関数であるので、高次の行列で構成される Walsh 関数を用いなくても、誤差の小さい変換が可能となる。これは、多くの高調波成分を有する正弦波と余弦波で構成される Fourier 変換に比較して、計算量が極端に少ないという利点を有する。

さらに、Walsh 係数は多くの項で 0 となり、計算に関与する項は高次になるに従ってその増加の割合は減少するという特異な特徴を有する。たとえば、基の関数が図 1 で示されるものとする、係数が 0 でない項数は、サンプル点数  $N = 16$  の Walsh 関数  $H_{16}$  の場合で係数 16 個のうち、low 関数と high 関数でそれぞれ 8 個、middle 関数で 4 個だけとなり、 $H_{32}$  の場合では係数 32 個のうち、それぞれ 10 個、5 個、 $H_{64}$  の場合では係数 64 個のうち、それぞれ 12 個、6 個、また  $H_{128}$  の場合でも係数 128 個のうち、それぞれ 14 個、7 個のみとなる。このように係数が 0 でない項は個数としては高次になるに従って増えるが、その増加率はわずかで、増加率は極端に減少し、そのため処理の対象が限定されて、効率的な処理が可能となる。

これらのことを考慮して、メンバーシップ関数の要素である式の木としてとりうる値について調べた。図 3 は、チューニング過程にあるメンバーシップ関数の一例であるが、メンバーシップ関数としてはあまり望ましくない形状を呈する。このようなメンバーシップ関数を要素に持つ個体は適応度が低いので、結果的には進化過程において淘汰されるが、進化過程での成長を

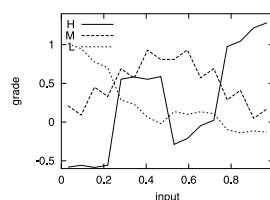


図 3 不適切なチューニング例

Fig. 3 Example of tuning's failing.

無駄に遅らせる要因となる。

このことから、メンバーシップ関数としての特性を失うことのないように、2.3 節で述べたように係数の大きさを探索条件に取り入れ、不必要な進化過程を省くことを試みたのである。その手続きとして、三角形メンバーシップ関数を基にした様々な関数を Walsh 変換し、その Walsh 係数から、各係数の有効範囲を定める。これにより、ファジィ制御に必要な概形を著しく損なうことなく、効率良くメンバーシップ関数が生成されることが期待できる。

これについて調べるため、基準となる様々なメンバーシップ関数の例を図 4 に示す。これらのメンバーシップ関数について、サンプル点数  $N$  を 16 とし、使用する Walsh 関数を  $H_{16}$  として Walsh 係数を算出した結果の一例を図 5 に示す。図より、ほとんど同じ値をとる項とばらつきのある項があることが分かる。なお、同図は high 関数パターン 1 (図 4 左上) の場合の結果であるが、他の関数についても同様の特徴が見られた。この結果から、係数にばらつきのある項のみを GP 操作の対象とし、ある程度の範囲内で変化させることで、おおよそ図 4 のテスト関数のような概形の関数を高速に作る事が可能であると考えられる。これらの係数の特徴を表 1 と表 2 にまとめた。

Walsh 係数がこれらの条件を満たせば、そのメンバーシップ関数はファジィ制御に必要な概形をある程度保持するものと考えられ、表 1 による制限を式の木生成の手順に加えるものとした。また、表 2 の符号パターンについては、概形に関する評価値を算出する際にあわせて判定し、評価した。これにより、メンバーシップ関数の特徴を保ちつつ、新たな概形を持った関数が効率良く生成されることが可能となる。

## 3. ファジィ交渉モデルへの適用

ファジィ交渉エージェントは、交渉者の目的と権限の範囲内で交渉当事者に代わって交渉に当たる代理人、あるいは仲介者としての働きを代行するもので、交渉過程において最も重要な行為である意思決定をエージェントに行わせるのである。このために、交渉当事

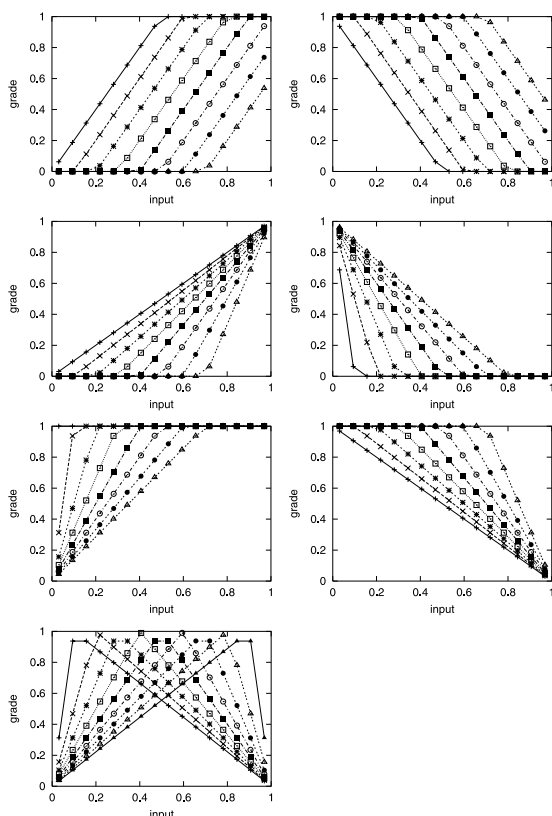


図 4 Walsh 係数の特徴調査のためのテスト関数

Fig. 4 Test functions for characteristic investigation of Walsh coefficient .

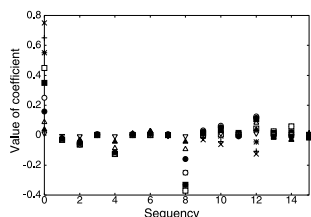


図 5 high 関数パターン 1 の Walsh 係数の特徴

Fig. 5 Characteristic of Walsh coefficient of high-function pattern 1.

者の習性や意思あるいは効用などを表現しうる交渉者モデルを意思決定原理に基づいて記述する。

### 3.1 エージェントの交渉モデル

自律的に交渉を進めることができるように、エージェントの意思決定原理として効用理論を用い、多次元効用理論の概念より個々の属性に対する効用値を合成して総合的な効用値を算出する。これによって、意思決定の対象の選好順序を定量的に表現する<sup>1),9)</sup>。

効用は、一般に望ましさの程度を表すので、評価項

表 1 テスト関数による Walsh 係数の範囲

Table 1 Range of Walsh coefficient by test functions.

| 項番号 | high 関数       | middle 関数       | low 関数        |
|-----|---------------|-----------------|---------------|
| 0   | 0.090~0.900   | 0.5             | 0.090~0.900   |
| 1   | -0.0315       | 0.0             | 0.0315        |
| 2   | -0.072~-0.034 | 0.0             | 0.034~0.072   |
| 3   | 0.0           | -0.0434~0.0     | 0.0           |
| 4   | -0.415~-0.085 | -0.0868~0.0868  | 0.085~0.415   |
| 5   | 0.0           | -0.0434~0.0     | 0.0           |
| 6   | -0.072~-0.048 | -0.0903~0.0     | -0.048~0.072  |
| 7   | 0.0           | -0.0434~0.0434  | 0.0           |
| 8   | -0.370~-0.090 | -0.2256~0.2256  | 0.090~0.370   |
| 9   | -0.033~-0.031 | -0.0625~0.0415  | -0.031~0.033  |
| 10  | -0.072~-0.066 | -0.1250~-0.0520 | -0.066~0.072  |
| 11  | 0.0           | -0.0434~0.0434  | 0.0           |
| 12  | -0.415~-0.415 | -0.2500~-0.0520 | -0.415~-0.415 |
| 13  | -0.033~-0.027 | -0.0434~0.0434  | -0.027~0.033  |
| 14  | -0.072~-0.058 | -0.0903~0.0903  | -0.058~0.072  |
| 15  | 0.0           | -0.0434~0.0162  | 0.0           |

表 2 テスト関数による Walsh 係数の符号パターン

Table 2 Sign pattern of Walsh coefficient by test functions.

| high 関数<br>項番号 | パターン |   |   |   |   |
|----------------|------|---|---|---|---|
|                | 1    | 2 | 3 | 4 | 5 |
| 6              | +    | + | + | - | - |
| 9, 10, 12      | -    | + | + | - | - |
| 13, 14         | +    | + | - | - | + |

| middle 関数<br>項番号 | パターン |   |   |   |
|------------------|------|---|---|---|
|                  | 1    | 2 | 3 | 4 |
| 4                | +    | - | + | - |
| 7                | -    | + | - | + |
| 8                | +    | + | - | - |
| 11, 13, 14       | -    | - | + | + |
| 15               | -    | + | + | - |

| low 関数<br>項番号 | パターン |   |   |   |   |
|---------------|------|---|---|---|---|
|               | 1    | 2 | 3 | 4 | 5 |
| 6             | -    | - | - | + | + |
| 9, 10, 12     | -    | - | - | + | + |
| 13, 14        | -    | - | + | + | - |

目の値(属性値)に対して効用値が単調増加あるいは単調減少するものと考え、また、あまり大きくない範囲内においては簡単のため、この変化は近似的に直線的と見なす。

このことに基づき、意思決定の対象に対する効用値は、次式で示される効用関数  $u_{ij}(y_j)$  で表す。

$$u_{ij}(y_j) = a_{ij} + k_{ij}y_j \quad (7)$$

ここで、 $i$  ( $i = 1, 2$ ) は 2 組のエージェント、 $j$  ( $j = 1, \dots, p$ ) は各属性、 $y_j$  は各属性値をそれぞれ表しており、また  $a_{ij}$ ,  $k_{ij}$  はそれぞれの属性に対する意思決定者の重みを含んだ効用関数に相応した係数である。

これにより、意思決定の対象に対する総合的な効用値は、多次元効用理論に基づき  $u_{ij}(y_j)$  の総和をとった総合評価の効用値  $u_i$  として次式で表される。

$$u_i = \sum_{j=1}^p u_{ij}(y_j) \quad (8)$$

次に、交渉における譲歩という行為が交渉者の効用

値の変化に基づいて行われるものであることを考慮すれば、エージェントの譲歩過程は、各属性に対する各エージェントの効用関数  $u_{ij}(y_j)$  を修正することで表現できる<sup>1),8)</sup>。したがって、交渉規則として、譲歩によって交渉を成立させるという共通の目標を前提として、上述の効用値を相手の値に近づけるように各効用関数の係数を修正し、この値に基づいて、上式で効用値を再計算する。

この係数を修正する操作において、各交渉者の譲歩の度合いを表す因子  $\alpha_{ij}$  ( $0 \leq \alpha_{ij} \leq 1$ ) を用いることで、修正した係数  $a'_{ij}$ ,  $k'_{ij}$  の値  $a'_{ij}$ ,  $k'_{ij}$  は次式のように表せる。

$$\begin{aligned} a'_{1j} &= (1 - \alpha_{1j})a_{1j} + \alpha_{1j}a_{2j} \\ k'_{1j} &= (1 - \alpha_{1j})k_{1j} + \alpha_{1j}k_{2j} \\ a'_{2j} &= \alpha_{2j}a_{1j} + (1 - \alpha_{2j})a_{2j} \\ k'_{2j} &= \alpha_{2j}k_{1j} + (1 - \alpha_{2j})k_{2j} \end{aligned} \quad (9)$$

また修正した効用関数  $u'_{ij}(y_j)$  は次式のようになる。

$$u'_{ij}(y_j) = a'_{ij} + k'_{ij}y_j \quad (10)$$

このことから、この交渉モデルで譲歩は因子  $\alpha_{ij}$  を用いて効用関数の修正幅を拡大していくという行為の過程であり、妥結は修正された総合的な効用値が一致した結果であって、そのときの効用値が最大となる対象が妥結案となる。

### 3.2 ファジィ意思決定

譲歩は、交渉相手の譲歩量と自分の固執の度合いに応じて譲歩量を決定するという交渉ルールに基づくものとする<sup>7)</sup>。そして、譲歩量を表現する割合  $\alpha_{ij}$  をファジィ制御によって求める。

そのため、制御に必要な制御対象に関する情報として、相手の譲歩量  $q_1$  と自分の固執の度合い  $q_2$  を入力とし、また制御量である自分の譲歩量  $r$  を出力とし、if (前件部) then (後件部) 型のファジィ制御規則によって、

IF  $q_1$  is small and  $q_2$  is big THEN  $r$  is big  
のように、言語表現で制御アルゴリズムを記述する。

上述したように、図 1 を例にとれば、入力値として、譲歩量および固執の度合いを、low 関数 (L), middle 関数 (M), high 関数 (H) の 3 つの関数で表す。

また、ルールは、その前件部が相手の譲歩量  $q_1$  と自分の固執の度合い  $q_2$  の組合せで構成すると、L, M, H の関数の組合せは、前件部だけで全部で 9 種類となる。そして、それぞれの前件部に対する後件部に、L, M, H の 3 つの関数のいずれかが組み合わせられるので、ルールとなりうる可能性のある組合せの全数は  $3^9 = 19683$  となる。

そのうち最適ルールとして、前稿<sup>2)</sup>で、本稿で述べ

るのと同じ交渉モデルに対して GA を適用することによって、以下のような 9 つのルールを導出している。

IF  $q_1$  is L and  $q_2$  is L THEN  $r$  is L  
IF  $q_1$  is L and  $q_2$  is M THEN  $r$  is L  
IF  $q_1$  is L and  $q_2$  is H THEN  $r$  is H  
IF  $q_1$  is M and  $q_2$  is L THEN  $r$  is L  
IF  $q_1$  is M and  $q_2$  is M THEN  $r$  is H  
IF  $q_1$  is M and  $q_2$  is H THEN  $r$  is H  
IF  $q_1$  is H and  $q_2$  is L THEN  $r$  is L  
IF  $q_1$  is H and  $q_2$  is M THEN  $r$  is L  
IF  $q_1$  is H and  $q_2$  is H THEN  $r$  is L

本稿では、交渉モデルが同じであることから、これらのルールを用いることにした。なお、推論結果である譲歩量  $r$  は、min-max 重心法 (マムダニの推論法) を用いて算出した。

### 3.3 適応度の設定

個体の適応度は遺伝操作において重要なパラメータである。この適応度を以下のように設定した。

適応度については、基本的に、提示案が交渉者双方で互いに納得できるものであれば、交渉結果として適切であると考ええる。そして、双方の譲歩の程度がほぼ等しくなる結果が望ましいものとする。このような妥結点を導きうるメンバーシップ関数が適切なものであって、そのようなメンバーシップ関数を要素に持つ個体の適応度は高いものとなる<sup>2),4)</sup>。

まず、交渉終了時に各エージェントの効用値がどれだけ近づいたかを次式で算出する。

$$T = \sum_{n=1}^h |u'_1(n) - u'_2(n)| / \sum_{n=1}^h |u_1(n) - u_2(n)| \quad (11)$$

ここで、意思決定の対象は複数あると考え、その個数を  $h$  としている。また、 $u_i(n)$  および  $u'_i(n)$  は、それぞれエージェント  $i$  の各対象  $n$  に対する総合的な効用値の交渉開始時の値および交渉終了時の値である。なお、 $T$  が 0 のとき、妥結案に到達したことを示す。

次に、各エージェント  $i$  の各評価項目  $j$  に対する固執の度合いを表す重み  $w_{ij}$  を次式を用いて算出する。

$$w_{ij} = \sum_{n=1}^h u_{ij}(n) / \sum_{n=1}^h u_i(n) \quad (12)$$

また同様にして、交渉終了時の重み  $w'_{ij}$  を算出し、総合的な固執の度合いの変化  $D_i$  を次式で求める。

$$D_i = \sqrt{\sum_{j=1}^p (w_{ij} - w'_{ij})^2} \quad (13)$$

表 3 評価項目と初期効用値  
Table 3 Attributes and initial utilities.

| 車の種類          | 属性            |                |                |                | 各属性に対する<br>$N_1$ の効用値 |          |          |          | 各属性に対する<br>$N_2$ の効用値 |          |          |          | 総合的な<br>効用値 |       | 選好順序  |       |
|---------------|---------------|----------------|----------------|----------------|-----------------------|----------|----------|----------|-----------------------|----------|----------|----------|-------------|-------|-------|-------|
|               | $y_1$<br>C120 | $y_2$<br>Space | $y_3$<br>Price | $y_4$<br>Speed | $u_{11}$              | $u_{12}$ | $u_{13}$ | $u_{14}$ | $u_{21}$              | $u_{22}$ | $u_{23}$ | $u_{24}$ | $u_1$       | $u_2$ | $N_1$ | $N_2$ |
| Opel Record   | 10.48         | 7.96           | 467            | 176            | 0.060                 | 0.170    | 0.285    | 0.140    | 0.260                 | 0.695    | 0.074    | 0.496    | 0.655       | 1.526 | 3     | 2     |
| Peugeot 505   | 10.01         | 7.88           | 495            | 173            | 0.071                 | 0.165    | 0.257    | 0.133    | 0.308                 | 0.676    | 0.067    | 0.471    | 0.627       | 1.522 | 4     | 3     |
| Peugeot 104   | 8.42          | 5.11           | 352            | 161            | 0.110                 | 0.000    | 0.398    | 0.106    | 0.470                 | 0.000    | 0.103    | 0.370    | 0.612       | 0.943 | 5     | 9     |
| Citroen Dyana | 6.75          | 5.81           | 248            | 117            | 0.150                 | 0.042    | 0.500    | 0.000    | 0.639                 | 0.171    | 0.130    | 0.000    | 0.692       | 0.940 | 1     | 10    |
| Citroen Visa  | 7.3           | 5.65           | 321            | 142            | 0.137                 | 0.032    | 0.428    | 0.060    | 0.583                 | 0.132    | 0.111    | 0.210    | 0.657       | 1.037 | 2     | 7     |
| VW Golf       | 9.61          | 6.15           | 392            | 140            | 0.081                 | 0.062    | 0.358    | 0.074    | 0.349                 | 0.254    | 0.093    | 0.261    | 0.575       | 0.956 | 6     | 8     |
| Mercedes 230  | 10.4          | 8.47           | 757            | 180            | 0.062                 | 0.200    | 0.000    | 0.150    | 0.268                 | 0.820    | 0.000    | 0.530    | 0.412       | 1.618 | 9     | 1     |
| Citroen CX    | 11.06         | 8.06           | 647            | 178            | 0.046                 | 0.176    | 0.108    | 0.145    | 0.202                 | 0.720    | 0.028    | 0.513    | 0.475       | 1.463 | 7     | 4     |
| Volvo 244     | 12.95         | 8.30           | 550            | 145            | 0.000                 | 0.195    | 0.203    | 0.067    | 0.009                 | 0.798    | 0.053    | 0.235    | 0.465       | 1.095 | 8     | 6     |
| BMW 520       | 12.26         | 7.81           | 686            | 182            | 0.017                 | 0.161    | 0.070    | 0.155    | 0.079                 | 0.659    | 0.018    | 0.547    | 0.402       | 1.303 | 10    | 5     |

同式は、 $D_i$  が小さければ、エージェントの希望どおりに進んだことを示すもので、また  $(|D_1 - D_2|)$  が少ないほど、双方が同程度譲歩したものと考えられる。

さらに、次式で示すように、各エージェント  $i$  が交渉開始時に重視していた属性  $t$  について、交渉開始時の重み  $w_{it}$  と交渉終了時の重み  $w'_{it}$  の差  $C_i$  を算出する。

$$C_i = |w'_{it} - w_{it}| \quad (14)$$

この  $C_i$  は、その値が小さいほど重視する属性についての譲歩が少なく、その属性へのこだわりが失われなかったことを示すものと考えられる。

このように、互いに納得できる妥結案を導きうるメンバーシップ関数を要素に持つ個体は、遺伝的プログラミングに関して、より適応した個体となる。このことから、各個体の交渉における以上の3項目の評価値の和  $f_2$  を次式で表す。

$$f_2 = T + |D_1 - D_2| + C_1 + C_2 \quad (15)$$

なお、この場合、 $f_2$  が小さい値をとるほど、適応度は高いと考える。また、妥結点探索のためのシミュレーションは、制限譲歩回数を10回までとし、その時点で各エージェントの総合評価の効用値の差が交渉開始時のその10%以下であるならば妥結案に到達したものと見なす。

ここで得られる交渉に関する評価値  $f_2$  と、前章で得られた概形に関する評価値  $f_1$  の2種類の評価値を総合して、以下の式によって個体の適応度  $F$  を算出する。

$$F = \frac{1}{1 + f_1 + f_2} \quad (16)$$

これは、 $f_1$ 、 $f_2$  がともに0を最良値としていることから、 $F$  は0~1の値をとり、また1に近い値をとるほど適応度は高いものとなる。

#### 4. 交渉シミュレーションの実行

具体的な事例についてエージェントによる交渉シミュレーションを実行し、メンバーシップ関数の最適

化を試みる。

##### 4.1 具体的事例

事例として、表3に示すような10種類の自動車の中から1台の自動車を選択して共同で購入する場合の交渉を考える。

同表は、意思決定の対象である車の種類とその評価項目と各エージェントの初期効用値や選好順序などを示す。なお、車の性能を表す評価項目として、(1)  $y_1$ : C120は時速120[km/h]で100[km]走行するのに必要な燃料の量[l]、(2)  $y_2$ : spaceは車内の広さ[m<sup>2</sup>]、(3)  $y_3$ : Priceは価格[100francs]、(4)  $y_4$ : speedは最高スピード[km/h]である。また、 $u_{ij}$  ( $i = 1, 2$ ,  $j = 1, 2, 3, 4$ ) は、式(7)に基づいて計算した各属性  $j$  の値  $y_j$  に対する各エージェント  $i$  の効用値で、 $u_1$  および  $u_2$  は、式(8)を用いて計算した各エージェントの各車種に対する総合評価としての効用値である。また、エージェント1 ( $N_1$ ) およびエージェント2 ( $N_2$ ) の選好順序もあわせて記している。

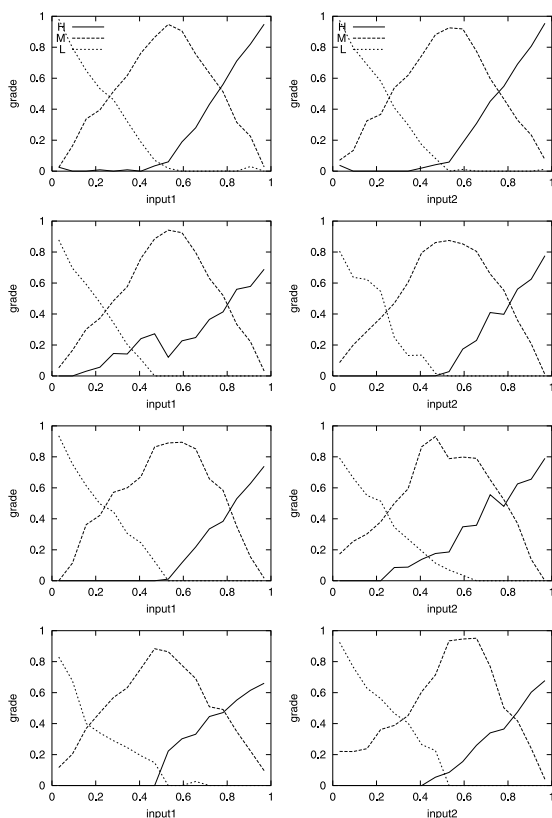
##### 4.2 メンバーシップ関数のチューニング

前節の事例に基づき、上述したような手続きに従ってメンバーシップ関数を作成した。図6は、集団にある個体数20個体(合計120個のメンバーシップ関数、1920本の式の木を含む)でGPを多数実行した結果で、メンバーシップ関数の生成過程を示す。また、同図は、代表的な世代(6, 24, 29, 49)での最優良個体の要素であるメンバーシップ関数(low関数, middle関数, high関数)を、上からこの世代の順に、2つの入力系統ごとに重ねて示している。同図において、世代交代が進むにつれてメンバーシップ関数の形が変化していく様子が観察される。

##### 4.3 交渉結果

前節で示したメンバーシップ関数のうち、最も適応度の良いメンバーシップ関数を用いて交渉シミュレーションを行った。この結果を表4に示す。また、このときの両交渉者の効用値の変化を図7に示す。





(a) Amount of concession (b) Degree of persistence

図 6 メンバシップ関数のチューニング

Fig. 6 Tuning of membership function.

表 4 は交渉での譲歩過程とその結果としての妥結案を示すものであり、これより 1 位は Opel Record, 2 位は Peugeot 505, 3 位は Mercedes 230 となっていることが分かる。この妥結案として提示された Opel Record は、エージェント 1 ( $N_1$ ) にとっては初期順位が 3 位, エージェント 2 ( $N_2$ ) にとっては初期順位が 2 位の車種であり、妥結案としては適当なものであると思われる。

図 7 は、上述の譲歩過程における各交渉者の効用値の変化を表すもののうち、選好順位が 1 位から 3 位までになった車の効用値の変化を示す。同図より、交渉が進むにつれて各交渉者がお互いに歩み寄るように効用値が変化の様子が観察される。

#### 4.4 考 察

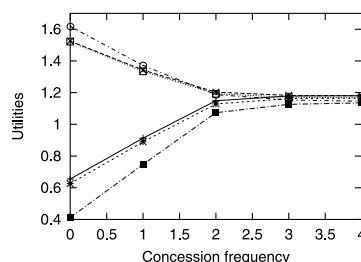
##### 4.4.1 個体の進化

図 8 は、世代交代が進むにつれて変化する適応度の推移を示したものである。同図において、各世代における最優秀個体の適応度と集団全体の平均の適応度は次第に大きくなっており、集団全体が少しずつ進化し

表 4 譲歩過程と妥結案

Table 4 Concession process and agreement.

|               |       | 初期値   | 1 回目  | 2 回目  | 3 回目  | 4 回目  |
|---------------|-------|-------|-------|-------|-------|-------|
| Opel Record   | $N_1$ | 0.655 | 0.913 | 1.149 | 1.179 | 1.180 |
|               | $N_2$ | 1.525 | 1.343 | 1.203 | 1.183 | 1.180 |
| Peugeot 505   | $N_1$ | 0.626 | 0.890 | 1.129 | 1.162 | 1.165 |
|               | $N_2$ | 1.522 | 1.335 | 1.193 | 1.171 | 1.167 |
| Peugeot 104   | $N_1$ | 0.613 | 0.724 | 0.779 | 0.791 | 0.793 |
|               | $N_2$ | 0.943 | 0.881 | 0.811 | 0.797 | 0.794 |
| Citroen Dyana | $N_1$ | 0.692 | 0.788 | 0.834 | 0.843 | 0.845 |
|               | $N_2$ | 0.940 | 0.892 | 0.852 | 0.848 | 0.847 |
| Citroen Visa  | $N_1$ | 0.657 | 0.785 | 0.858 | 0.873 | 0.877 |
|               | $N_2$ | 1.036 | 0.963 | 0.895 | 0.883 | 0.880 |
| VW Golf       | $N_1$ | 0.575 | 0.700 | 0.792 | 0.804 | 0.804 |
|               | $N_2$ | 0.957 | 0.881 | 0.813 | 0.804 | 0.803 |
| Mercedes 230  | $N_1$ | 0.412 | 0.747 | 1.073 | 1.126 | 1.136 |
|               | $N_2$ | 1.618 | 1.371 | 1.187 | 1.153 | 1.145 |
| Citroen CX    | $N_1$ | 0.475 | 0.755 | 1.025 | 1.064 | 1.069 |
|               | $N_2$ | 1.463 | 1.259 | 1.104 | 1.078 | 1.073 |
| Volvo 244     | $N_1$ | 0.465 | 0.653 | 0.856 | 0.874 | 0.870 |
|               | $N_2$ | 1.105 | 0.962 | 0.870 | 0.864 | 0.865 |
| BMW 520       | $N_1$ | 0.403 | 0.656 | 0.906 | 0.940 | 0.944 |
|               | $N_2$ | 1.303 | 1.119 | 0.974 | 0.950 | 0.946 |



+:  $N_1$ -Opel Record, \*:  $N_1$ -Peugeot 505, ■:  $N_1$ -Mercedes 230  
 ×:  $N_2$ -Opel Record, □:  $N_2$ -Peugeot 505, ○:  $N_2$ -Mercedes 230

図 7 譲歩過程での選好順位 3 位までの車種に対する効用値の変化  
 Fig. 7 Change of utilities in concession process.

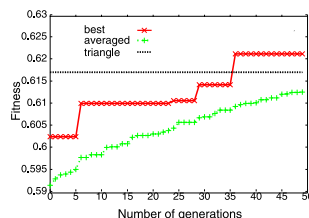


図 8 Walsh 変換を用いた場合の個体適応度の推移

Fig. 8 Transition of individual fitness in the case of Walsh transformation.

ていることが分かる。特に、各世代での最優秀個体の適応度が増大することは、それまでにないより優良な解が確実に生成されていることを示すものであり、最適解生成への進化が実現されているものと考えられる。

ここで、図 6 に示した最適メンバシップ関数は、49 世代目で出現した最優秀個体のものであり、その適応度は 0.621 と計算される。このうち、交渉過程での評価値  $f_2$  は 0.603 であり、両交渉者の効用値が近づ

表 5 代表的な式の木とその値および三角形形状メンバーシップ関数の Walsh 係数

Table 5 Typical trees of expression, evaluated values and Walsh coefficients.

(a = 1.0, b = 2.0, c = 3.0, d = 4.0, e = 5.0, f = 6.0, g = 7.0, h = 8.0, i = 9.0, j = 0.0).

| No. | high function                   |            |          | middle function                |            |         | low function |            |         |
|-----|---------------------------------|------------|----------|--------------------------------|------------|---------|--------------|------------|---------|
|     | S-expression                    | evaluation | Walsh    | S-expression                   | evaluation | Walsh   | S-expression | evaluation | Walsh   |
| 0   | (+ed)                           | 0.09       | 0.25     | (+(*gf)h)                      | 0.50       | 0.50    | (+c(*eh))    | 0.43       | 0.25    |
| 1   | (-jc)                           | -0.03      | -0.03125 | (/jf)                          | 0          | 0       | (+ad)        | 0.04       | 0.03125 |
| 2   | (-j(+ig))                       | -0.07      | -0.0625  | (/jf)                          | 0          | 0       | (g)          | 0.07       | 0.0625  |
| 3   | (j)                             | 0          | 0        | (-/(/j(+ab))(*ab))ij)          | 0          | 0       | (/(/jf)f)    | 0          | 0       |
| 4   | (-d(+gf))                       | -0.09      | -0.125   | (+bj)                          | 0.02       | 0       | (+(*ae)h)    | 0.14       | 0.125   |
| 5   | (j)                             | 0          | 0        | (/(*j(/jc)g)                   | 0          | 0       | (/(/bj)g))   | 0          | 0       |
| 6   | (-(/(+f(-(*e(+ad))j)(+fh)))g)g) | -0.03714   | 0        | (/(/jb)/a(*h(-ac))f))          | 0          | 0       | (j)          | 0          | 0       |
| 7   | (*cj)                           | 0          | 0        | (b)                            | 0.02       | 0       | (/bi)        | 0.00222    | 0       |
| 8   | (-ji)                           | -0.09      | -0.25    | (+jj)                          | 0          | 0       | (+ji)        | 0.09       | 0.25    |
| 9   | (b)                             | 0.02       | 0.03125  | (-jf)                          | -0.06      | -0.0625 | (-ja)        | -0.01      | 0.03125 |
| 10  | (- (+df)(+d))                   | 0          | 0.0625   | (- (/(*bj)(+bj)))(+f(/i(-jj))) | -0.06      | -0.0125 | (-hg)        | 0.01       | 0.0625  |
| 11  | (/bd)                           | 0.005      | 0        | (*j(+i(*ab)))                  | 0          | 0       | (/di)        | 0.0044     | 0       |
| 12  | (+h(/(*f)(+ee)))                | 0.134      | 0.125    | (-j(+i(*hj)))                  | -0.09      | -0.25   | (h)          | 0.08       | 0.125   |
| 13  | (-bj)                           | 0.02       | 0        | (*bb)                          | 0.04       | 0       | (j)          | 0          | 0       |
| 14  | (e)                             | 0.05       | 0        | (/(*ha)c)                      | 0.02666    | 0       | (i)          | 0          | 0       |
| 15  | (/jc)                           | 0          | 0        | (j)                            | 0          | 0       | (jf)         | 0          | 0       |

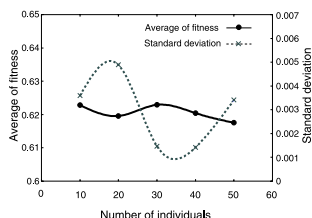


図 9 初期個体数による平均適応度と標準偏差の変化

Fig. 9 Changes in average fitness and standard deviation by number of individuals.

いた割合を示す  $T$  が 0.0 であることから、重みについての評価  $D_i$ ,  $C_i$  に関する値のみの和となっている。

このことから、図 8 より、個体の適応度は 0.621 程度の値で収束したと見なすことができ、49 世代目の個体を最優秀個体として最適メンバーシップ関数と考えることができる。

このような個体生成の再現性を調べるために、種々の条件の下でシミュレーションを行い、メンバーシップ関数の概形や適応度の変化を観察した。その一例として、初期個体の個体数を変えた場合の最終適応度の変化を調べた結果を図 9 に示す。同図は、個体数を 10, 20, 30, 40, 50 個体として、それぞれ 10 回ずつ解探索を行った場合の最終適応度の平均と標準偏差を示す。なお、探索条件としての個体数にかかわらず、全 50 個の標本の平均適応度は 0.621 で、標準偏差は 0.00384 であった。同図より、各個体数での適応度の平均はあまり変わらず、また標準偏差についても少しばらつきがみられるものの極端な変化はないものと思われた。ただ、両者ともに、個体数が多くなるに従ってわずかながら減少する傾向が観察される。このことから、個体数が多い方がより探索効率は良好と考えられるが、この条件の範囲内においては、概して本手法は安定した解探索が可能であるものと思われた。

また、個体数が 20 個体の場合に得られた最終世代の

代表的な個体のメンバーシップ関数 (high 関数, middle 関数, low 関数) の要素である式の木を S 式 (S-expression) で表現したものを表 5 に示す。同表には、これらの式の木から計算して得られた値 (evaluation), および進化開始時に基準とした三角形形状のメンバーシップ関数の Walsh 係数 (Walsh) も示す。これより、世代を経ることによって、階層が深くて、複雑な構造のものも出現するが、階層が浅くて単純な構造も観察される。前者については、進化過程においてさかんに交叉などの遺伝操作が行われた形跡を示すものであるが、後者については、(1) Walsh 係数が、前述したような比較的簡単な関数 (四則演算子) と実数定数で表現しやすい値である、(2) 進化初期の段階で生成されたものでも、式の木の値として優れていれば、世代を経て生き残ることができる、などによるものと考えられる。また、上でも述べたように、同表から、Walsh 変換で得られた係数は、基になるメンバーシップ関数が三角形形状の直線関数であるので、主要なものの以外の値は 0 となるという非常に特徴的な値をとっていることが分かる。このことは、同じような直交変換に用いられる Fourier 変換の場合と相違して、簡便な形式で計算することができ、この点において操作性や処理効率が優れているものと考えられる。

#### 4.4.2 三角形形状のメンバーシップ関数との比較

本手法で生成したメンバーシップ関数を用いた場合と三角形形状のメンバーシップ関数を用いた場合を比較する。

入出力関数のすべてに三角形形状のメンバーシップ関数を用いた場合も、選好順位 1 位が Opel Record, 2 位が Peugeot 505, 3 位が Mercedes 230 となり、選好順位が 3 位までは自動生成した最適メンバーシップ関数を用いた場合と同様の結果になった。そこで、この結果を 3.3 節で述べた GP 探索過程における個体の評価方法と同様に評価すると、交渉時の評価値  $f_2$  は

0.621 であり、また概形に関する評価値  $f_1$  は 0 なので、個体適応度  $F$  は 0.617 となった。

一方、本手法によって自動生成された最適メンバーシップ関数の適応度  $F$  は 0.621 であるが、これを純粋に交渉過程での精度を比較する意味で概形に関する評価値を除いて算出しなおすと 0.624 となる。この結果と比較すると、妥結案としては同様の結果ではあるが、両交渉者の譲歩量の平等さと、重要視する属性へのこだわりの尊重という観点から、より適切な交渉支援ができるものと考えられる。また、妥結点に到達するまでの譲歩回数を比較すると、三角形のメンバーシップ関数を用いた場合が 7 回であるのに対し、本手法で生成された関数では 4 回と減少している。これは、交渉に関する評価値が良くなっていることから、交渉が効率良く進められたことを示唆するものと考えられる。

このような結果を得た要因は、図 6 に見られるようにメンバーシップ関数の変化が交渉者の選好の変動を表現していることにあると思われる。たとえば、同図左側のグラフにおいて、middle 関数のピークが右にずれ、high 関数の傾きが緩やかになっている。これにより、自分の固執を少々譲るかたちとなり、このことがスムーズな交渉を可能にしたものと推察できる。

このような微妙な概形の変化を生み出すことができることも、Walsh 変換を用いた本手法の特長であると考えられる。

#### 4.4.3 Walsh 変換を用いない GP 手法による方法

本稿で提案する手法とは異なり、ルートノードに演算子 (+, -, \*, /), その左右の子ノードとしての終端子に変数  $x$  か定数項を配置した二分木で 1 つの初期メンバーシップ関数を構成し、これを GP で成長させていくという従来法<sup>10)</sup>と比較した。なお、この場合の条件として、入力関数 1, 2 の high 関数, middle 関数, low 関数からなる 6 個組を 1 個体とすることは同様であるが、これら 1 つのメンバーシップ関数はそれぞれ 1 つの木で表現されることが相違する。また、メンバーシップ関数としての概形について評価することは同様であるが、初期木は基準となるものを設定せずにランダムに作成するという方法をとっている。なお、個体数は 128 個体として、木 (メンバーシップ関数) は 768 個とした。また、適応度は上述とほぼ同様にして求めた。

この結果として、生成されたメンバーシップ関数を図 10 に示す。また、この場合の適応度は 0.51 程度となり、本稿で提案した手法や三角形メンバーシップ関数を上回る結果は得られていない。なお、この方式は、比較的簡便であるという利点を有するので、よ

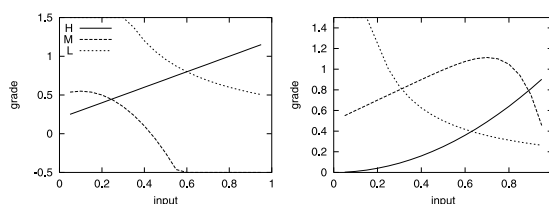


図 10 Walsh 変換を用いない GP 手法によるメンバーシップ関数の例

Fig. 10 Example of membership functions tuned by GP technique without the Walsh transformation.

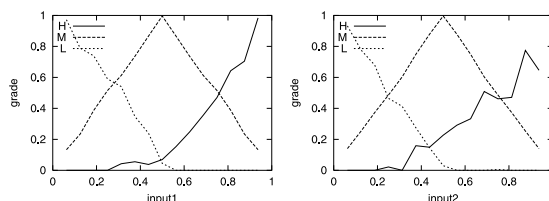


図 11 Fourier 変換を利用した GP 手法によるチューニングされたメンバーシップ関数の例

Fig. 11 Example of membership functions tuned by GP technique with Fourier transformation.

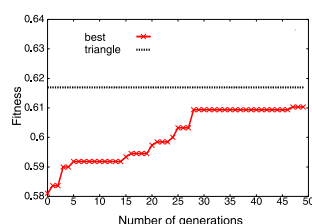


図 12 Fourier 変換を用いた場合の個体適応度の推移

Fig. 12 Transition of individual fitness in the case of Fourier transformation.

り深く検討することによって改善されるべきであると思われた。

#### 4.4.4 Fourier 変換を用いた GP 手法による方法

本手法で提案する Walsh 変換の代わりに、直交変換として一般によく知られる Fourier 変換を用いて、GP 手法によるメンバーシップ関数のチューニングを試みた結果を以下に示す。

項数  $N = 17$  ( $n \leq 8$ ), サンプル点数 16 というパラメータ以外はすべて、Walsh 変換を用いたときと同じ値に設定し、同じ事例で試してみたところ、図 11, 図 12 のような結果を得た。

交渉結果としては、三角形メンバーシップ関数や Walsh 変換を用いてチューニングされた関数の場合と同じ妥結案であった。適応度については、図 12 に見られるように、三角形メンバーシップの場合とほぼ同程度の値となった。これについては、図 11 の入力

系統 2 の high 関数が単調増加していないことによる概形に関する評価値の低さが影響していると思われる。

このようにメンバーシップ関数の概形がファジィ制御に適したものでなければ、提示された妥結案の信頼性は低い。このことから、Fourier 変換によるチューニング方法では、メンバーシップ関数の概形を改善する必要があるものと考えられ、今後の検討課題である。

## 5. おわりに

本研究は、ファジィ制御に基づく交渉エージェントの重要な構成要素であるメンバーシップ関数の最適化を目的として、進化システムの一つである遺伝的プログラミング手法に Walsh 変換を適用した自動生成システムについて検討した。

そのなかで、標準的に用いられる三角形の直線関数を基準にして、これに Walsh 変換を適用することによって、その形状から進化させて最適メンバーシップ関数にチューニングする新しい方法を提案した。この新規性は、基の関数を Walsh 変換することによってスペクトル分析し、特徴抽出することによって進化過程を効率化する方法にある。

また、本手法は、Walsh 変換で得られた係数が、基になるメンバーシップ関数が三角形の直線関数であるので、主要な係数以外の値は 0 となるという非常に有用性の高い特徴を有しており、同じような直交変換に用いられる Fourier 変換の場合と比較して、簡便な形式で計算することができ、操作性や、処理効率においてもきわめて優位性が高いものと考えられた。

実際にこの手法を、ファジィ制御を用いた妥結点探索のための交渉支援エージェントの構築に適用した。具体的な事例としては、協調して自動車を購入する場合に合意を得るための交渉を取り上げた。

この結果、本手法は、ファジィ制御に必要な特徴を保持しつつ、事例に適合するメンバーシップ関数へのチューニングを可能にするとともに、適切な妥結案を導くことができるものと思われた。

なお、本手法は、その適用分野として、本稿で取り扱った交渉エージェントだけではなく、ファジィ制御におけるチューニングや最適化を必要とする問題に拡張して適用することが可能であると考えられる。また、Walsh 変換がデジタル技術による特徴抽出が可能であるため、システム制御や画像認識の最適化にも効果的に利用可能であると考えており、現在、知能ロボット制御や生体認識の新しい方法として検討を進めている。今後の課題として、その他の様々な実用的な事例に応用できるように汎用性を高めることを試みる。

## 参考文献

- 1) 松村幸輝：間欠性カオス写像を用いたファジィ意思決定に基づく交渉エージェント、情報処理学会論文誌, Vol.39, No.5, pp.1203-1214 (1998).
- 2) 松村幸輝：遺伝的アルゴリズムによるチューニング手法を用いたファジィ意思決定に基づく交渉シミュレーションシステム、日本ファジィ学会論文誌, Vol.9, No.2, pp.223-230 (1997).
- 3) Koza, J.: *Genetic Programming*, MIT Press (1992).
- 4) 松村幸輝, 後藤道彦, 浜松芳夫：遺伝的プログラミングによるチューニング手法を用いたファジィ交渉エージェント, 電気学会論文誌, Vol.121-C, No.7, pp.1160-1170 (2001).
- 5) 遠藤 靖：ウォルシュ解析, 東京電気大学出版局 (1993).
- 6) 古谷博史：Walsh 変換による突然変異と交叉に対するスキーマ定理の導出, 情報処理学会論文誌, Vol.43, No.4, pp.1050-1060 (2002).
- 7) 松村幸輝：ファジィ意思決定に基づく交渉シミュレーションシステム, 日本ファジィ学会論文誌, Vol.8, No.5, pp.882-890 (1996).
- 8) Jarke, M., Jelassi, M.T. and Shakun, M.F.: MEDIATOR: Towards a negotiation support system, *European Journal of Operational Research*, No.31, pp.314-333 (1987).
- 9) 松村幸輝：遺伝的アルゴリズムに基づく交渉シミュレーションシステム, 電気学会論文誌, Vol.116-C, No.5, pp.524-533 (1996).
- 10) 松村幸輝, 後藤道彦, 浜松芳夫：カオス結合系を用いた遺伝的プログラミング手法に基づくファジィ交渉エージェント, 電気学会論文誌, Vol.124-C, No.10, pp.1501-1510 (2004).

(平成 16 年 4 月 6 日受付)

(平成 16 年 11 月 1 日採録)



松村 幸輝 (正会員)

1978 年大阪市立大学大学院博士課程修了。茨城大学工学部助教授、甲南大学理学部教授を経て、2002 年より鳥取大学工学部教授。工学博士。主として、人間の思考や意思決定を支援する知能情報科学の研究に従事。電気学会、電子情報通信学会、日本ファジィ学会、経営情報学会等各会員。