

# 2R-7 遺伝アルゴリズムによる動作ネットの学習法

桑原 敏  
NTT情報通信網研究所

## 1. はじめに

プロセス制御やロボット制御等の分野で、動作を複数の部分動作に分割、及び統合し、分散制御する方法が提案されている。[1][2] ユーザにとって、これらの分散制御パラダイムの使用は、オブジェクト指向的に制御機構を簡明に作成できる等の利点は図れるが、予め部分動作間の因果関係を厳密に定式化する必要があり、多様な並列動作が想定される制御対象に適用する場合は、この点が制御システム構築の難点と考えられる。即ちこのような技術の実現には人手による定式化の軽減が課題となり、機械自身が自動的に部分動作間の因果関係を獲得する等の機械学習法を確立することが期待される。

ここでは、分散制御パラダイムとして、複数の部分動作エージェントを活性化、あるいは非活性化信号で網状に相互接続した分散制御型の動作ネットについて、その学習法として遺伝アルゴリズム[3]の適用可能性について報告する。

## 2. 動作ネットの概要

各部分動作エージェントは、制御対象からのセンサー信号と他エージェントからの活性化(act)/非活性化(inact)信号を入力、制御対象への動作(operation)信号と他エージェントへの活性化/非活性化信号を出力、また活動状態(active)と非活動状態(inactive)の2つの内部状態を有する状態機械と考えられる。各エージェントは以下の2種の判定を行ない、その結果に基づき独立に状態遷移と出力を行なう。

### <動作可能判定>

エージェント固有の動作可能条件リストとセンサ信号との部分一致判定(p)

- ・一致の場合 ; p=true
- ・不一致の場合 ; p=false

### <起動可能判定>

受信した活性信号と非活性信号を時間積分した活性値が活性化しきい値を比較(c)

- ・活性値<しきい値 ; c=true
- ・活性値>しきい値 ; c=false

上記判定結果の論理積(p∧c)を内部入力とした時の各部分動作エージェントの状態遷移と出力を表1に示す。

表1 動作エージェントの状態遷移と出力

Current State	Output	Next State	
		p∧c=true	p∧c=false
active	front behave=on	active	inactive
inactive	behind behave=off	active	inactive

front :act/inact信号のactive状態での送出エージェント  
behind:act/inact信号のinactive状態での送出エージェント

上記frontとbehindはエージェント内ではそれぞれリストで保持し、両者を総称してネットリストと称する。動作ネットは上記で定義された部分動作エージェントを網状に相互接続したものであり、各エージェントは独立に状態遷移、及び出力を行い、全体として並列動作するものである。ここでc判定での活性値は受信されるact/inact信号の時間積分値をTとするとT/(T+1/ε)で算定する。活性値は動作が終了し、c判定がfalseになった時点でリセットする。

## 3. 遺伝アルゴリズムによる学習

動作ネットで学習すべきパラメータとしては、ここでは表2で示すように、ネットリストと活性化しきい値とする。なお、比較的記述可能な条件リストや動作については予め人が与えるものとする。

以下「板にハンマーで釘を打つ」の動作ネットを例に学習法を概説する。

### (1) 個体のコーディング

1動作ネットを1個体(individual)として、表2の学習パラメータを図1のような単純線形表記した。但し、各エージェント毎にしきい値εは小数2桁の実数値、また残りのネットリストは3値(f/b/d)からなる5つのコードである。それぞれの位置はact信号送出先エージェントであり、fはactive時、bはinactive時の送出に対応し、dはact信号の送出が無いことを示す。なお、この他にinact信号送出先のネットリストがあるが、図1では表記していない。

表2 学習パラメータ例

エージェント名	しきい値 ε	ネットリスト					
		0	1	2	3	4	5
0 Take-hammer	0.71	-	f	d	f	b	f
1 Take-Nail	0.50	f	-	b	d	f	b
2 Stand-Nail	0.54	b	b	-	f	b	b
3 Drive-Hammer	0.75	d	b	f	-	d	b
4 Put-Hammer	0.06	f	b	f	b	-	d
5 Put-Nail	0.01	d	f	f	d	f	-

f:front-act \* しきい値にしきい値を固定し  
b:behind-act 活性値=T/(T+1/ε)としたとき  
d:don't care のεを学習

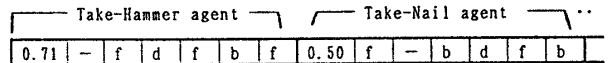


図1 コーディング

### (2) 適応度

適応度については制御対象や教示しようとする人の理想状態に依存するものであり、一般的に定められない。

ここでは動作の3つのバリエーションに対応して、3種の適応度を定義した。

・ 通常動作；

$$\text{Fitness} = a_0 * \text{正常動作数} + a_1 * 1 / (1 + \text{誤動作数}) + a_2 * 1 / (1 + \text{無効待時間})$$

・ 釘無し動作；

$$\text{Fitness} = b_1 * 1 / (1 + \text{誤動作数}) + b_2 * 1 / (1 + \text{無効待時間})$$

・ ハンマー無し動作；

$$\text{Fitness} = c_1 * 1 / (1 + \text{誤動作数}) + c_2 * 1 / (1 + \text{無効待時間})$$

誤動作は人が判断しておかした動作、例えば釘をとってすぐに釘を戻す等の検出回数である。無効時間は釘やハンマーを動作しないで、長く持つ時間等の累積である。a、b、cの係数は適応度を調整するためのものであり、正常動作の数を重視する時はa0を大きくし、誤動作回避を重視する場合はa1、b1、c1を大きくする等に使う。

(3) クロスオーバ

しきい値εとネットリストは独立にクロスオーバを行なう。しきい値εは通常のビットエンコーディングではなく、実数値単位で他の個体とクロスオーバする。

ネットリストについては3値(f/b/d)であり、一般的なクロスオーバと考えられる。

(4) ミューテーション

しきい値εについては、乱数探索手法に近く、子供の個体のしきい値に確率的に乱数値で置き替える。

ネットリストについては、3値間で確率的に変動を加える。

4. 実験

ソフトウェアシミュレータに基づく図2の実験システムを作成した。

(1) 実験システム概要

・ エージェント数：部分動作に対応して6種

- ① Take-Hammer：ハンマーを取る
- ② Take-Nail：釘をつかむ
- ③ Stand-Nail：釘を板に立てる
- ④ Drive-hammer：ハンマーで板に釘を打つ
- ⑤ Put-Hammer：ハンマーを置く
- ⑥ Put-Nail：釘を戻す

・ センサ数：10種

- ① 右手が空、② 左手が空、③ ハンマーが無い、④ 釘が無い、⑤ 板が無い、⑥ 右手にハンマー、⑦ 左手に釘、⑧ 板に釘、⑨ 右手が使用中、⑩ 左手が使用中

・ 動作時間：上限下限を定義し、その範囲で変動。  
・ inact信号は使用しない。

(2) 学習

以下の2方法で学習を試みた。

- ① 一括学習：6種エージェントを全部使って、通常動作、釘無し動作、ハンマー無し動作を繰り返す。
- ② 段階的学習：Take-Hammer、Take-Nail、Stand-Nail、Drive-Hammerで通常動作を学習した後、学習した関係パラメータを固定とし、Put-Hammer、Put-Nailを追加して、異常動作を含めた残りのパラメータを学習する。

(2) 結果

図3に一括学習での世代数(Generation)と適応度の変化について5回の試行結果を示す。これからほぼ100世代で理想状態に近付いているのが分かる。但し、理想状態を測る適応度の定義に依存し、理想状態になるパラメータが1つかどうかは保証できない。

図4は一括学習と同一条件での段階的学習の世代数と適応度の変化を示すものである。これから最初の通常

動作については20世代付近で理想状態に近付き、さらにエージェント追加後の計80世代付近で、異常動作も含めた理想状態に近付いているのがみられる。

5. 評価

遺伝アルゴリズムによるパラメータ学習を評価する手法は確立しておらず、特に最適解が得られるのか、到達解が複数ある場合の評価等について、他の方法と比較するのは難しい。今回の実験規模はパラメータ数で36個であり、全探索による単一解導出をベースに探索規模を推定すると、しきい値εの実数部を8ビットとすると一括学習では $2^6 * 3^{30} (1.3 * 10^{16})$ 、段階的学習では $2^6 * 3^{12} + 3^{18} (4.2 * 10^8)$ 程度である。複数解が想定されるが遺伝アルゴリズム学習では探索数の低減が期待される。段階的学習法は大規模な動作ネットを想定して、学習の大幅効率化を狙いとするものであるが、今回の実験では一括学習に対して、1割程度の低減に留まった。しかし、適切なエージェントの分割等により、学習効率向上の可能性は高いと考えられる。

6. おわりに

部分動作エージェントの階層的な分割と遺伝アルゴリズムによる段階的学習が実用的な時間で大規模な動作ネットを学習するのに有効であると考えており、今後さらに実験、評価を継続させたいと考えている。

<参考文献>

[1] Nilson, N. "Action Networks" Rochester Planning Workshop-1989  
 [2] Maes, P. "The Dynamic of Action Selection", IJCAI-1989  
 [3] Holland, J.H., "Adaptation in Natural and Artificial System", Univ. Michigan Press-1975

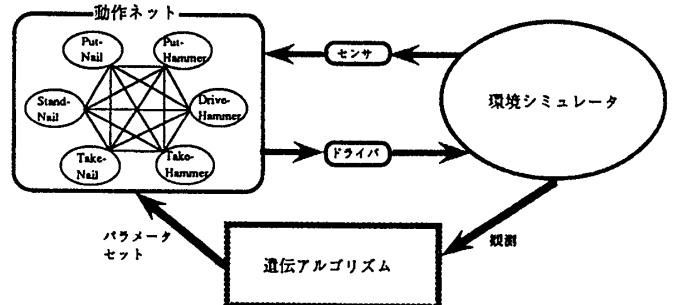


図2 実験システム構成

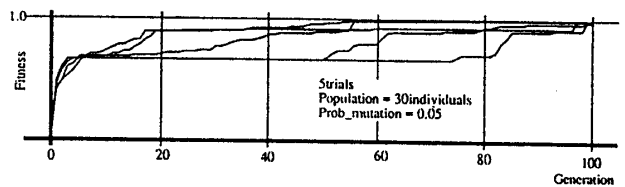


図3 学習回数と適応度 (一括学習)

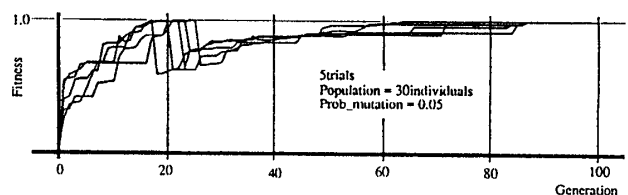


図4 学習回数と適応度 (段階的学習)