

## 物理世界の猫とネズミ — クラシファイア・システムによる学習

上田 雄悟 星野 力

筑波大学 構造工学系

〒305 茨城県 つくば市 天王台 1-1-1

E-mail : hoshino@kz.tsukuba.ac.jp

「猫とネズミ」と名付けたゲームを構築し、クラシファイア・システム (CS) の学習効果を調べた。従来の AI 的な手法によって記述されたルールが、if (餌がある) then (食べろ) のように明示的に、上位のレベルで表現されるのに対して、ここではルールを明示的に記述することを避け、if (自分の座標・相対速度・残り体力) then (動く方向と力・捕獲行動の有無) のように非明示的に、より下位の物理レベルの記述をしている。また、優秀なルールの系列を積極的に記憶する発火の実績を重視したアルゴリズムを考案し、常に安定して好成績を残すことに成功した。バケツリレー・アルゴリズム (BB) の検証を行うことで、Holland が主張する橋渡しクラシファイア (bridging classifier) の存在も確認された。さらにランダムに設定された初期ルールから、優秀なルールに進化させることで知識の獲得を行なった。これは、ルールを創発 (emerge) させるという意味で、非明示性をコンセプトとする人工生命のアプローチでもある。

## How cat and mouse acquire the physics-law ? — A learning by Classifier Systems

Yugo UEDA Tsutomu HOSHINO

Inst. of Engineering Mechanics, Univ. of Tsukuba

1-1-1 Ten-nodai, Tsukuba-shi, Ibaraki-ken 305

E-mail : hoshino@kz.tsukuba.ac.jp

Classifier Systems (CS) was applied to the learning of cat and mouse in the simulated "Cat & Mouse" pursuit-evasion game. The traditional explicit rules conventionally taken in the Artificial Intelligence research would be of the type: "if (food exists) then (eat it)". On the contrary, the rules here is not of this type but is subject to the lower-level descriptions, such as "if (coordinates, velocity, energy) then (direction and force to move or capture)". The simulation resulted in the rules improved from the initial random default rules: A better rules of a mouse which were able to flee from a cat. The level of the emergence is so low that the intelligent cat or mouse would have evolved in the frame work of the Artificial Life (AL).

## 1 はじめに

クラシファイア・システム (classifier systems 以下 CS) は、クラシファイアと呼ばれる if then 型のルール (プロダクション・ルール) の集合であり、バケツリレー・アルゴリズム (bucket brigade algorithm 以下 BB) によってクラシファイアの強度 (有効性を表す重み) を更新し、さらにクラシファイアを遺伝子とみなして突然変異や交叉を施す遺伝的アルゴリズム (genetic algorithms 以下 GA) によって、新しいクラシファイアを発生するシステムである [1]。

ここではランダムに設定された初期ルールから、優秀なルールに進化させることで、知識の獲得を行った。これは、ルールを創発 [2] (emerge) させるという意味で、非明示性をコンセプトとする人工生命のアプローチでもある。

### 1.1 物理法則に支配された環境

本研究では「猫とネズミ」と名付けた、猫とネズミが鬼ごっこをすることで点数を競うゲームを構築し、このゲームにおける CS の学習効果を調べた。従来の AI 的な手法によって記述されたルールが、if (餌がある) then (食べる) のように明示的に、上位のレベルで表現されるのに対して、ここではルールを明示的に記述することを避け、if (自分の座標・相対速度・残り体力) then (動く方向と力・捕獲行動の有無) のように非明示的に、より下位の物理レベルの記述をしている。格子世界に代表される環境では、自分が行きたいと思う方向に自由に移動できるのに対し、この猫とネズミの環境は物理法則に支配されているので、クラシファイアが出力した力ベクトルの方向へ、必ずしも移動できるとは限らない。格子世界に比べ、このようなより現実の世界に近い複雑な環境において、CS や GA がうまく適応できるかどうかは、とても興味深い問題である。

### 1.2 ゴールの分解

ここでゴールの分解という古典的問題について述べておこう。ゴールとは最終目的であり、ナビゲーションでは目的地へ到達することである。普通ゴールへは一気に到達できないので、サブゴールという部分問題が設定される。例えば、京の三条大橋へ到るには、まず品川宿へ到ることが最初のサブゴールであり、次には小田原、箱根、大井川、... というサブゴールが設定される。

強化学習というパラダイムでは、品川へ到着すれば、とりあえず 20 点を与え、箱根は 200 点という具合に、ご褒美を用意し、ちょっとでも京へ近づくエージェント (個体) を褒める (強化する)。

このときゴールの分解は誰がしたのであろう? 明らかに人間である。しかし猫やネズミは、自分で自分を褒めるしかない。人工知能の歴史をみれば、このゴールの分解については、サミュエルのチェッカープログラム以来、あまり進展はない。サミュエルのデルタ学習は、現代最適制御理論におけるハミルトニアンのパラダイムと奇妙に一致している。そこではハミルトニアンにあたるのは、局面の静的評価関数である。その後、ゲーム学習はこのパラダイムから大きく離れることはなかった。

一方これに匹敵するパラダイムは Holland による BB である。これは目的を達したときに得た報酬が 1 つ前の手に戻される。この戻し方は前回の手が次の手に賭け、勝てば報酬を得るという賭けのプロセスになっている。こうして、明示的なサブゴールを設けることなく、作戦のシーケンスが学習できると主張する [1]。

### 1.3 問題と実験の定義

本研究では、CS を (1) 強化学習として捉えた場合と (2) 人工生命的に考えた場合の 2 通りの実験を行ったので、それぞれの主旨・実験方法を明確にしておく。

#### 1. 強化学習として捉えた場合 (実験 1・実験 2)

**主旨** CS による学習を強化学習と捉えて、ゴール (overall goal) を部分的に分解したサブゴール (sub goal) を設定し、学習性能の向上を目指す。サブゴールを設定することで Holland の BB から逸脱することになるが、強化学習のパラダイムではむしろ自然な考え方である。

**実験方法** 猫とネズミを初期配置する際の距離を一定 ( $200 \pm 10$  [dot]) とする。この距離は学習の間を通して不変であるが、2 つのエージェント (猫とネズミ) 間の距離に応じたサブゴールを設定する。

#### 2. 人工生命的に考えた場合 (実験 3)

**主旨** 人間が明示的にサブゴールを設定することはせず、クラシファイアの強度の更新を BB のみで行い、その効果を検証す

る。また光・音・匂いなどの強さによってエージェントを誘導したり、壁に沿う行動計画 (wall following problem) を獲得するために壁の周囲にタイルを配置する [3] といった、暗黙のサブゴールも一切排除する。最終的な局面からしか過去の状況を評価できない問題が存在することを考えれば、サブゴールに頼るのは安易である。これは Holland の BB のオリジナルであるが、明示的な教師を設定しないとする人工生命の考え方に通じるものである。

**実験方法** 学習の初期からある段階まで、猫とネズミが配置される距離の初期値を徐々に伸ばし、その後は一定とする。これは、問題がやさしいうちに猫がネズミを捕獲すること覚え、時間を経て問題が難しくなった場合でも、猫がネズミを捕獲し続けることを期待するものである。最初に猫がネズミを捕まえることは偶然に期待するしかないので、その確率を上げるために猫の近くにネズミを置くが、サブゴールに相当する誘導は行わない。

## 2 ゲームの概要

図1にゲームの概要を示す。ゲームを管理するのはゲームマスター・プログラムであり、2人のプレイヤーはそれぞれ エントリ1、エントリ2 というエントリ・プログラムを用意する。

### 2.1 ゲームマスターとエントリの通信

- ゲームマスターが、2つのエントリ・プログラムと行う1回の通信を1ステップとし、36ステップを1セット、12セットを1ゲームとする。
- エントリ・プログラムは、1ステップごとにゲームマスターから環境の状況を表すデータを受けとり、それを元に自分が移動するための力  $f_x, f_y$  を計算して返す。

### 2.2 ゲームの規則

- エントリ1が猫、エントリ2がネズミとなり、鬼ごっこをする。
- 猫は、ネズミを捕まえると得点する。

- 両者ともに同量の初期設定された体力を持ち、 $f_x, f_y$  の大きさに比例して、またネズミを捕まえようとする動作に応じて減少する。
- 体力を使い切ると死んでしまい、生き残った方が得点する。
- どちらか一方が得点し1セット終了すると、野球の表裏のように立場を入れかえて、(猫はネズミに、ネズミは猫になり) 次のセットを行う。
- 36ステップ以内に勝負がつかないときは、猫から逃げ切ったと考えて、ネズミが得点する。

## 3 CSによる適応・学習モデル

本研究におけるCSの役目は、ゲームマスターから与えられる環境の状況を判断・分析し、その状況における有利な行動を決定する、いわゆる戦略の立案である。ここでいう環境の状況とは、ある時刻  $t$  における自分と相手の絶対座標・絶対速度・残り体力であり、行動とは、環境に出力する力の大きさ・方向である。猫やネズミは、“もし環境の状況が〇〇のときは、××の力ベクトルを出力しろ。” というようなクラシファイアの命令に従って、意思決定を行う。

### 3.1 座標系と行動部のフィードバック

ここではゲームマスターから与えられる座標を、絶対座標のまま使用、あるいは相対座標に変換してから使用する場合の2通りを考え、さらに、猫やネズミの体の向きを考慮する・しないを組み合わせた4通りを試した。

以上の座標系とは別に、時刻  $t$  のクラシファイアへ、時刻  $t-1$  に出力した力ベクトルをフィードバックするかしないか、という実験も行った。

### 3.2 情報のコーディング

CSの目や耳である検出器 (detector) は、環境からの入力 (自分と相手の座標・速度・残り体力) を、0, 1のビット列を用いて、メッセージ (自分の座標・残り体力・相対位置・相対速度) に変換する。クラシファイア [表1] の条件部の # は don't care と呼ばれるもので、0でも1でもマッチすることを許す記号である。表2に、このクラシファイアの  $a \sim 1$  の役割を示す。

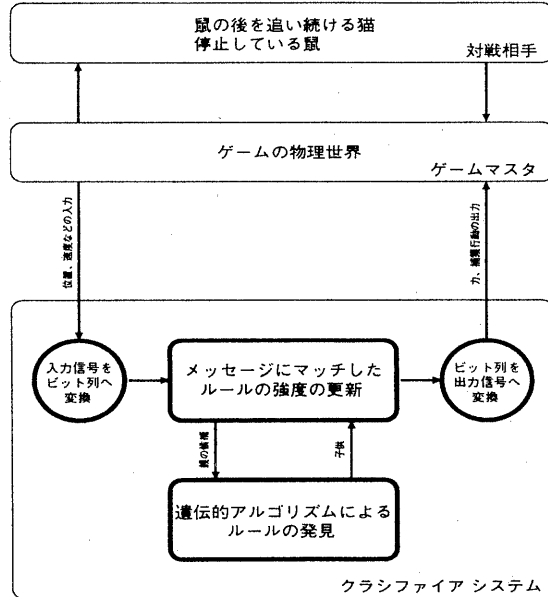


図 1: ゲームの概要

表 1: 猫のクラシファイアの例 (学習前・後)

	条件部	行動部	強度	背番号	発火実績メモリ
学習前	##1#1#111111	0111100	100.0	0.0000	.....
	##0#0#011100#	0010100	100.0	0.0001	.....
	100#0#0#1100#	0111010	100.0	0.0002	.....
	0#000#1##01#	0101110	100.0	0.0003	.....
	01##0##00##	0010100	100.0	0.0004	.....
学習後	100100100000	1110011	3192.9	200.0232	279.0395 246.0244 279.0395 246.0244 205.0387
	100100011000	1100101	2766.2	259.0247	269.0087 269.0087 174.0202 265.0112 278.0176
	011011011101	0010111	2714.4	253.0086	278.0436 274.0189 252.0351 278.0428 278.0428
	011100111101	1011111	2688.6	210.0332	276.0365 273.0135 277.0383 272.0154 273.0135
	011100100000	0100001	2652.8	204.0130	266.0048 269.0285 235.0293 259.0249 259.0249

表 2: クラシファイアの要素

条件部	相対 $x-y$ 座標の場合		極座標の場合		bit 数
	a				
a	自分の $x$ 座標			同左	3
b	自分の $y$ 座標			同左	3
c	自分の体力の残り			同左	3
d	相手との相対距離の $\Delta x$ 成分	d'	相手との相対距離の大きさ		3
e	相手との相対距離の $\Delta y$ 成分	e'	相手との相対距離の方向		3
f	相手との相対速度の $\Delta x$ 成分	f'	相手との相対速度の大きさ		3
g	相手との相対速度の $\Delta y$ 成分	g'	相手との相対速度の方向		3
h	1ステップ前の力の $x$ 成分	h'	1ステップ前の力の大きさ		3
i	1ステップ前の力の $y$ 成分	i'	1ステップ前の力の方向		3
行動部	j	移動のための力の $x$ 成分	j'	移動のための力の大きさ	3
	k	移動のための力の $y$ 成分	k'	移動のための力の方向	3
	l	捕獲の意思を示すフラグ		同左	1

## 4 実験に使用したモデル

### 4.1 実験 1

クラシファイアの強度を更新するメカニズムを説明する。下の項目の1はGoldbergのBB[4]に、2と3はHollandのBBに準じている。

1. まず、すべてのクラシファイアは、1ステップごとに強度に比例した「税金」を支払う。
2. さらに発火したクラシファイアは、強度と非#数の積に比例した「賭金」を支払う。
3. 発火したクラシファイアが複数あるときは、賭金に比例した確率で1つのクラシファイアが選択されて勝者となり、その勝者の「賭金」は、1ステップ前の勝者の「報酬」( $r[t-1]$ )となる。(パケツリレー・アルゴリズム)
4. またサブゴールを使用する場合は、クラシファイアの行動を1ステップごとに評価し、行動に対する「賞罰」( $e[t]$ )を与える。 $e[t]$ が正の場合は「報酬」を、 $e[t]$ が負の場合は「罰」を意味する。このサブゴールによって猫やネズミは、最終的な報酬を待たずに報酬を得ることができる。以下の式において、 $c_c$ 、 $c_d$ 、 $c_e$ は適当な定数、 $distance[t]$ 、 $energy[t]$ は、それぞれ時刻 $t$ における猫とネズミとの距離、自分の残り体力を表すものとする。

猫の場合

$$e[t] = \begin{cases} c_c & \text{if capture} \\ c_d \times \left(1 - \frac{distance[t]}{distance[t-1]}\right) & \text{otherwise} \end{cases}$$

ネズミの場合

$$e[t] = \begin{cases} c_e \times \frac{energy[t]}{energy[t-1]} & \text{if } (100[\text{dot}] < distance[t]) \\ c_d \times \left(\frac{distance[t]}{distance[t-1]} - 1\right) & \text{otherwise} \end{cases}$$

以上から(1)税金(2)賭金を支払って、(3)報酬(4)行動部の評価を得たクラシファイアの強度は次のようになる。ただし、 $s[t]$ は時刻 $t$ の強度であり、 $c_{tax}$ 、 $c_{bid}$ は実験によって異なる、強度に対するバランスを配慮して決めた0～1の係数である。

$$s[t+1] = \left(1 - (c_{tax} + c_{bid} \times \frac{\text{非#数}}{\text{条件部の遺伝子数}})\right) s[t] + r[t] + e[t]$$

### 4.2 実験 2

CSには大別して2つのアプローチがある。1つは単一のルールをGAにおける個体と考えるミシガン・アプローチであり、他方はルールセットを個体と考えるピッツ・アプローチである。それぞれ次に挙げるような利点・欠点を持つ。

- ミシガン・アプローチ

利点 遺伝的オペレータはクラシファイア単体に作用するので、ルールセットとしての動作は比較的安定

欠点 各クラシファイアに対する評価(BB)が難点

- ピッツ・アプローチ

利点 ルールセットを評価するので各クラシファイアに対する評価が不要

欠点 すべての個体を評価するための計算力が必要

実験2ではピッツ・アプローチよりも少ない計算量で、ネズミを捕まえたクラシファイアの時間的な系列を積極的に保存するために、発火実績メモリを持つクラシファイアを考案した[図2]。各

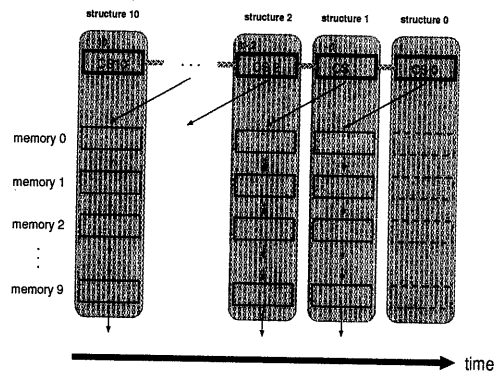


図2: 鎖でつながれたクラシファイアとその構造

クラシファイアには固有の背番号が付けられており、ネズミの捕獲に参加したクラシファイアは、自分の次の時刻(未来)に発火したクラシファイアの背番号を記憶する。この動作を予め設定された鎖の数繰り返すことで、優秀なクラシファイアの系列を保存する。つまり、未来のクラシファイアの背番号を発火実績メモリに記憶することに

よって、捕獲実績のあるクラシファイアのつながり(鎖)ができあがる。ただし鎖の数が9個のときは、つながれたクラシファイアは10個となる。

その後、同じような環境の状況が発生した場合、発火実績メモリに記憶されたクラシファイアを優先して選択することによって、安定した成績を残せるようになる。このアルゴリズムを図3に示す。

```

if (ある時刻 t に、猫がネズミを捕獲) {
  while (classifiers を結ぶ鎖の数) {
    時刻 t-1 に発火して勝利した classifier は、
    時刻 t に発火して勝利した classifier の背番号を記憶
  }
}

その後、対戦中に

if (発火実績メモリが空ではない classifier が発火) {
  システムは、その発火実績メモリの内容(背番号)を保存
}

環境(ゲームマスタ)と1回通信

if (時刻 t+1 に発火した classifier(s) の中に、
  システムが保存した背番号と同じ classifier(s) が存在) {
  背番号が同じ classifier(s) の中から、
  ルーレット法で選択
}
else {
  発火した classifier(s) の中から、
  # が最少のものをルーレット法で選択
}

```

図3: ルール系列を重視した選択アルゴリズム

この方法には、ミシガン・アプローチだけでは獲得しにくいクラシファイアの系列を、ピッツ・アプローチよりも少ない計算量で獲得し、単純なルーレット法によって、突飛な行動をするクラシファイアが発火してしまうのを抑制する効果がある。

### 4.3 実験3

実験3は、実験1・実験2と異なり、サブゴールを使用せずに Holland の BB のみによって強度の更新を行う。実験1・実験2では居候クラシファイアが淘汰されやすい機構を用いたが、実験3ではクラシファイアの総数の20%に限り、居候クラシファイアの存在を認める。実験3では問題の難易度が変化するので、必要となるクラシファイアの種類も変化する。その変化に対処できるようなクラシファイアを生成するため、GAが、居候クラシファイアを親として選択できるようにしている。

## 4.4 CSにおけるGAの役割

ここでクラシファイアがランダムに生成される場合の確率を考える。

$$\frac{5,000}{3^{12\text{bit}}(\text{条件部}) \times 2^{7\text{bit}}(\text{行動部})} \approx 0.0007$$

この式において、分子はクラシファイアの総数、分母は12 [bit](条件部)+7 [bit](行動部)で表現された比較的短いクラシファイアの0.1、#による数学的な組み合わせの総数を示す。

この確率の低さを克服するために、クラシファイアを遺伝子と見なし10,000ステップごとにGAを作用させる。優秀なクラシファイアから、さらに優秀なクラシファイアが誕生することが期待できるが、頻繁にGAを作用させると、遺伝子(クラシファイア)の多様性がなくなり、均一な集団に収束してしまうことで、かえってCSの性能が落ちることがある。CSにおいて、最適なクラシファイアの系列は存在するが、最適な唯一のクラシファイアは存在しない。

表3: GAのパラメータ

	実験1・実験2	実験3
すべてのルール数	5,000	—
親の候補となるルール数	2,500 ~ 4,750	—
GAで生成されるルール数	500	50 (1%)
交差点 [/ rule]	1	—
突然変異率 [/ bit]	0.001	0.001 以上 (可変)

## 5 実験と考察

ここでは、クラシファイア・システム(CS)の学習の指標として、ゲーム中に猫がネズミを捕まえた総回数やその捕獲効率を考える。総回数とは、学習開始から終了までを通して加算したものであり、捕獲効率とは10,000ステップあたりの捕獲回数、つまり総回数の差分である。ここで考える優れた学習結果とは、捕獲効率が相対的に高かった時点の学習結果(CSの強度・クラシファイアの系列など)を指す。

### 5.1 実験1・実験2の結果

座標系やフィードバック有無をいろいろ組合せて試した結果“体の向きを基準とした絶対極座標表現で、フィードバックなし”が最も良い成績を残した。図4のグラフは、その結果である。グラフ中の“09-05”という数字は、鎖の数が9個(結ばれるクラシファイアは10個)で、発火実績メモリの数が5個であることを意味し“00-00”は鎖も発火実績メモリも使用していないことを表す。

猫の成績は、09-01 → 00 00 → 09-05 → 09-10の順に向上している。09-10と09-05の成績はとても安定しており、今回実験した中で、最高の捕獲確率となった9/10以上を得ている。また、成績が振るわなかった09-01の試行回数を2,000,000ステップから、さらに3,000,000ステップまで伸ばしてみると、2,000,000ステップから2,100,000ステップにかけて、捕獲確率が4/10から7/10へ急激に上昇し、学習・適応モデルとしての大幅な進化が見られた[図5]。

発火実績メモリを使用することによって、何が改善されたのだろうか？発火実績メモリを使用しない(一般的な)CSによる猫が、ネズミを逃がしてしまう原因として、クラシファイアを選択する際に使用するルーレット法の確率的な挙動があげられる。環境が激しく変化するような場合では、その確率的な選択が、CSをロバストにする効果があるが、この猫とネズミのゲームでは、対戦中に突然、慣性質量や重力加速度が変化することはないので、確率的な選択はむしろ成績を下げてしまう。

発火実績メモリを使用することの意義は、その確率的な選択をできるだけ避けることにある。学習後のクラシファイアの発火実績メモリを観察すると、同じ背番号を複数記憶しているものもあれば、多様な背番号を記憶しているものもある[表1]。このことから発火実績メモリが複数ある方が、環境の状況や世代ごとに異なる個体集団に、柔軟に対応できることが推測される。

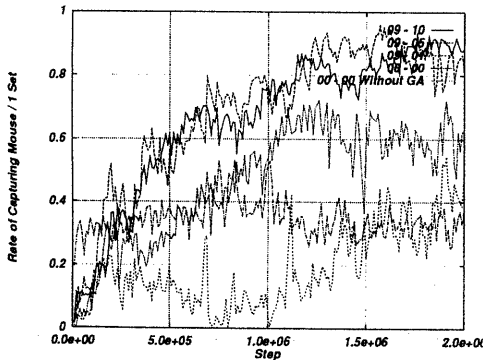


図 4: 発火実績メモリ数と捕獲効率の関係

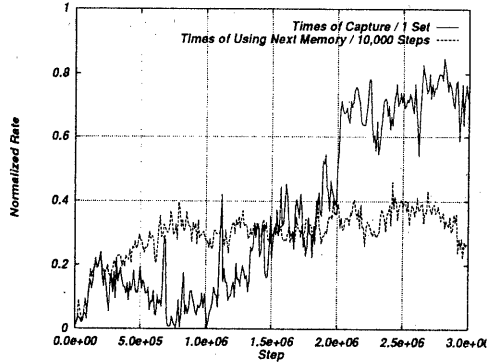


図 5: 捕獲効率と発火実績メモリの使用率

## 5.2 実験3の結果

最後に実験3の結果を図6に示す。居候クラシファイアの数に20%(1,000個体)に制限した場合(実線)は、問題の難易度が一定になる1,000,000ステップを越えた後も、コンスタントにネズミを捕まえ続けているのに対し、制限しない場合(点線)は、1,600,000ステップ付近で、ネズミを全く捕獲できなくなった。

この結果は、居候クラシファイアが少ないほうが良いことを示しているが、集団中の遺伝子の多様性が失われつつある場合、居候クラシファイアの遺伝子から優秀なクラシファイアが生成されることも考えられるので、居候クラシファイアをすべて淘汰すればいいというものではない。

最後に、フィードバックを加えた実験において「橋渡しクラシファイア」の存在が確認されたことを付け加えておく。表4は、ネズミを捕まえたクラシファイアの背番号の時系列を示す。この例では、最終的にネズミを捕まえたクラシファイア7.0487が得た報酬は、橋渡しクラシファイア5.0232によって、一気に0.4805まで戻されることを示しており、Hollandの主張を裏付けている。

表 4: 橋渡しクラシファイア (5.0232)

5.0379	→	0.4805	→	5.0232	→	5.0232	→	5.0232	→
5.0232	→	5.0232	→	5.0232	→	5.0232	→	5.0232	→
5.0232	→	5.0232	→	5.0232	→	5.0232	→	5.0232	→
5.0232	→	7.0487							

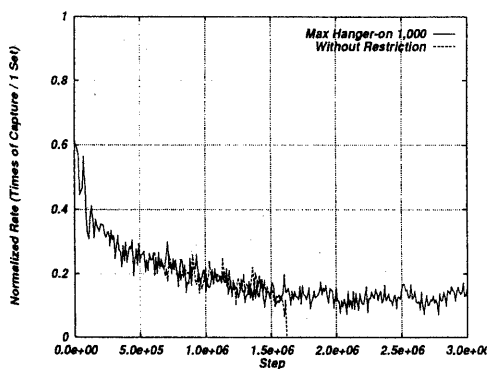


図 6: BB のみによる捕獲効率

## 6 結論

本研究では、「猫とネズミ」と名付けたゲームを構築し、その環境においてクラシファイア・システム (CS) の学習効果を調べた。このゲームの環境は物理法則に支配されているため、格子世界よりも難しい環境である。環境が難しいことに加え、さらに物理的な力や速度といったレベルの表現を用いてクラシファイアを記述し、ボトムアップ的に知識を獲得することを試みた。その結果として次のことが言える。

### 1. サブゴール付き学習について

- クラシファイアの発火系列を記憶する学習アルゴリズムを考案し実験した。このアルゴリズムはクラシファイアへのコーディング法や GA の不安定な挙動に影響されにくく、常に良い結果を残した。
- 遺伝的アルゴリズムによって、メッセージに対し特殊性が高い (より専門的な) クラシファイアを獲得できた。
- GA による特殊性の向上は、CS の性能を高めた。
- 学習の対象が変わらない場合、居候クラシファイアは無駄である。
- 学習の対象が変わる場合、居候クラシファイアは多様性を保つために必要であり、次の世代のクラシファイアの種 (たね) になり得る。
- サブゴールによる学習は、BB による学習よりも効率が高い。

### 2. BB のみによる学習について

- 問題のレベル次第では BB でも学習可能である。
- 獲得されるクラシファイアの系列は、世代を経るにつれて徐々に長くなる。
- Holland が主張する橋渡しクラシファイアの存在が確認された。
- クラシファイア (遺伝子) の多様性を保持する機構が必要である。

### 3. CS 全般について

- エージェントの体の向きを基準にとった座標系は有効である。
- 物理のレベルでクラシファイアを記述しても、知識は獲得される。
- 物理のレベルでクラシファイアを記述する場合、そのパラメータ調整は困難である。
- ランダムに設定された初期クラシファイアを優秀なクラシファイアに進化させることで知識獲得を行った。
- 人工生命的な手法によって、クラシファイアは物理法則を獲得した。

## 参考文献

- [1] Holland, J. H. : "Properties of the Bucket Brigade Algorithm", Proceedings of the First International Conference on Genetic Algorithms and Their Applications, 1985.
- [2] 星野 力 : 『はやわかりシステムの世界』, 共立出版, 1993.
- [3] Koza, J. R. : "Genetic Programming, on the Programming of Computers by Means of Natural Selection", The MIT Press, 1992.
- [4] Goldberg, D. E. : "Genetic Algorithms in Search, Optimization & Machine Learning", Addison-Wesley Publishing Company, 1989.
- [5] 上田 雄悟 : 「追跡ゲームにおける物理法則のクラシファイアによる獲得」, 平成 5 年度 筑波大学大学院 修士課程 理工学研究科 修士論文, 1994.