共進化によるマルチ移動ロボット環境における協調行動の獲得

内部 英治
浅田 稔
中村 理輝

1 はじめに

近年，複数ロボットを同時に学習させるための方法論として共進化現象が注目されている．共進化は，膨大な探索領域の中から適切な戦略を発見することができることで，マルチエージェント環境における複数エージェントの同時学習の問題に対する一つのアプローチとして注目されている．

典型的な共進化として，獲物と捕食者の競争的なタスク設定による例がしばしば議論されている．Cliff and Miller [3] はこの問題について，共進化現象を実験的に解析し，獲物と捕食者でセンサの配置が異なる進化を遂げる結果を得た．また，Floreano and Nolfi [4] は，小型ロボットを使って実環境で獲物と捕食者の共進化実験を行い，ある環境条件下での相互の技術向上を確認した．このタスクを扱った研究では，異種のエージェントが他方の戦略に打ち勝つために現在の行動戦略を変更する場合が多い．

また，別のタスクとして，協調行動ついても幾つか研究がなされている．伊庭 [5] は GP を用いて，複数のエージェントがどのように協調行動を獲得できるかについて，タイルワールドでの幾つかの実験を通して議論している．この手法では，学習者間で共有できる部分集団が仮定されており，敵対者を含む共進化を扱うことは想定していない．また，Luke [10] は 11 台のサッカーエージェントによるチームの共進化実験を行い，チーム内での協調行動を実現する個体を獲得することに成功した．しかし，チームとしての共進化しか扱っておらず，協調行動するエージェント自身の共進化実験，すなわち個体の適合度関数の設計問題は扱われていない．

我々の興味は，競争と協調を含むマルチエージェント環境である RoboCup [7, 14] において，個々のロボットが敵との競争行動，味方との協調行動をどのように実現するかにある．そこで本研究は，各ロボットに遺伝的プログラミング (Genetic Programming, 以下 GP) [8, 9] を適用し，適合度関数や初期条件を設定することで，協調行動が獲得できることをシミュレーション結果によって示し，各ロボットを適切に共進化させるための条件について考察する．ここで，各個体は既に強化学習によって獲得したビヘービアを下位のビヘービアとして持ち，そのビヘービアを状況に応じて使い分けることを学習する．このことにより，下位のビヘービア部分を実環境でも微調整することが可能 [19] であり，更に GP による解の探索を軽減できる．また，個々のロボットは各自がそれぞれ個体集合を持つが，個体集合間に対して移民は行わない．共進化は他の進化ロボットなどを含むことで，変化しつづける環境

との相互作用により実現される．この過程の中で，望みの行動獲得を試みる．問題の複雑さは以下の二つにまとめられる

1) 協調行動のための共進化は，相互の進化の正確な同期を必要とする．
2) 3 台の共進化実験では，単純なものからより複雑な状況を含む広範な探索領域を提供可能な，適度に複雑な環境設定が必要である．

以下では，協調タスクにおける共進化について最初に説明する．次に我々が扱ったタスクと環境を述べ，進化的手法で実現した共進化実験の詳細を述べる．次に，GP で必要な関数及び終端集合，適合度関数などを説明し，2 台のロボットの場合の協調行動と 3 台の場合の協調・競争行動の共進化実験のシミュレーションおよび実ロボットへの適用結果を示す．最後に，これらの結果に対する考察と今後の課題を示す.

2　協調タスクにおける共進化

マルチエージェント環境において，複数の学習者が同時に学習する際には，一般に以下の問題に直面する．

1．他者の政策が不明：学習ロボットは，他のロボットの行動政策を前もって知ることはできない． そのため，行動と観測を通じて推定する必要がある．但し，相手も学習過程で行動政策を変更する可能性があり，問題はより困難となる．

2．学習の非同期性：相互に学習するロボットは，同時に互いのスキルを改善させなければならない．相手のロボットが自分より早く学習が収束し，スキルアップした場合，学習中のロボットとっての環境は，相手の行動政策が固定するものの，非常に困難な状況 (相手が単独でタスクを達成する) ことになり，戦略を改善することは容易ではなくなる．

3．報酬の割り当て：もし，報酬がチーム全体に対するものだった場合，その報酬を適切に学習者に割り当てる必要がある．全ての学習者に均等に与えた場合，通常期待される協調行動は生まれず，1 台のロボットが自分で目標を達成し，他のロボットは無関係な行動が強化され，無意味な行動を獲得する可能性がある．逆に個体に対する報酬のみでは，報酬の奪い合いとなり，結果的に全体のパフォーマンスは悪化する．

著者らのこれまでの研究では，ロボットによるサッカーの世界競技大会 RoboCup [7, 14] を取り上げ，ドリブル，シュートなどの個々のロボットの行動，パス，センタリングなどの複数ロボット間の協調行動，さらにはブロックなどの競争行動を視覚に基づいた学習によって獲得するために，これまで強化学習に関連した種々の手法を開発してきた．はじめに一台のロボットの学習問題について，ボールをゴールにシュートするタスクを取り上げた．自分の知覚空間 (ここでは視覚) と運動空間を経験を通じて抽象化し，状態行動

空間を形成する手法として，オフライン学習 [2] とオンライン学習 [17] を提案した．更に，複数ロボットの協調行動を強化学習で実現するために，状態ベクトルをシステム同定の手法で推定する手法を提案し，それらと相互学習するための学習スケジュールによりパッサーとシューターの協調行動を実現した [19]．この手法では，相互作用を通して学習者の行動と他の学習者との行動の関係を同定し，他者の行動政策を推定することにより，1. の問題に対処した．また，3 の問題に対しても環境設定と適合度関数の工夫により解決可能と考えられる．しかし，推定や学習を安定化させるために設計者が学習する順番を指定する必要があった．つまり，ある学習者に対する他のロボットの行動政策は単一であった．しかしながら，順番によっては望まれる適切な行動が獲得されない可能性がある．

一方，共進化は膨大な探索領域の中から，適切な戦略を発見することができることで，上記の問題 1 に対する方法として近年注目されている．一般に，共進化のパターンは以下の三つにまとめられる [12]．

A. **固定した政策のスイッチングの繰り返し：**
このパターンは，獲物と捕食者の場合によく見られ，現在の他者の戦略にだけ対処できる行動政策を獲得し，それ以外の戦略には対処できない．このため，同じ政策が何度と繰り返され，双方で改善が見られない．

B. **局所解へのはまりこみ：**
一方のロボットが他方を圧倒し，そのロボットの戦略だけが改善され，それ以外は低い評価に収束する．双方が安定状態に早く収束するが，共にスキルレベルは低く，その後変化が起きない．

C. **相互のスキル向上：**
ある条件下で，互いに向上し，結果として変化する環境内で各ロボットが戦略を改善する場合を指す．すべてのロボットが効率的に進化するので本来の共進化と呼べる．

一般のロボット間には，競合だけでなく協調や無視といった関係も存在し，様々な関係が共進化の過程を通してどのように獲得されるかを議論する必要がある．我々は 1 対 1 やチームとしての競争行動の共進化以外に，協調行動を含む他の共進化のケースがありえないか，更にその場合，タスク，環境，適合度関数の複雑さがどのように関係するか，などの事項を検討してきた．

複数ロボットを共進化させる場合，**パターン C** が期待されるが，多くの場合は **パターン A, B** であり，**パターン C** を実現するためには，適切な適合度関数と適度なタスクの複雑さが必要になる．しかし，一般的な適合度関数とタスクの複雑さが共進化に及ぼす影響を議論するのは困難であるため，ここでは特定のタスク，特に協調を含む問題について，様々な実験結果を示し，一般的な議論への足掛かりとしたい．以下では，

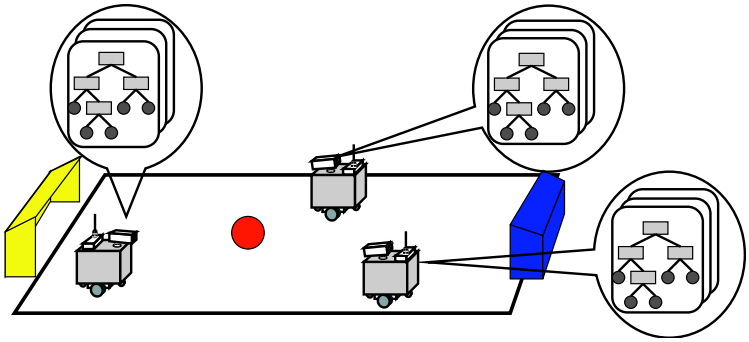
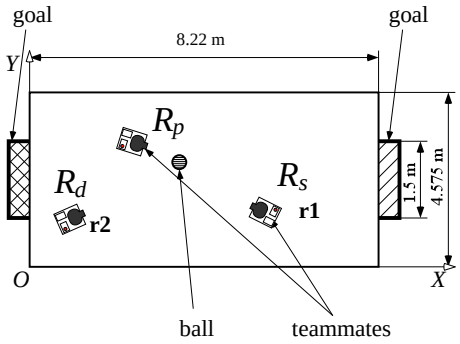
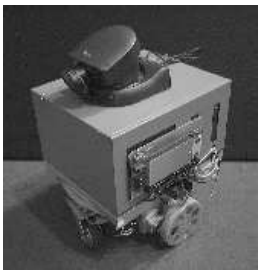


図 1: 本手法のアプローチ



(a) フィールドの大きさ



(b) 想定する実ロボット

図 2: 3 台のロボットによるサッカーゲーム

1. 各ロボットに与えられたタスクの複雑さの関係,
2. 適合度関数における個体としての評価と全体としての評価のトレードオフ,

について多くの実験をおこない，それらに基づいて協調行動を含む共進化について考察する．図 1 に我々の提案するアプローチのイメージを示す.

3　タスク及び仮定

3.1　環境とロボット

マルチエージェントを研究する例題として，ロボットサッカーの世界競技会および研究会議 (RoboCup) が提案され，世界中の多くの研究者がこの問題に取り組んでいる [7, 14]．RoboCup は広範な最新技術が集積され，試さ

れる標準問題を提供することにより，人工知能とロボティクスのあらたな標準問題を生み出す試みである．この問題は，敵，味方に分かれた多くのロボット群が存在するという環境の中で，他の個々のロボットの行動理解及びチームとしての行動戦略パターンの理解などといった高度な視覚認識の問題を含んでいる．

そこで，図 2 (a) に示す環境で，2 ないし 3 台のロボットに簡単なサッカーゲームをおこなわせ，共進化による協調行動について考察する．便宜上，各ロボットを R_p, R_s, R_d として区別する[1]．環境はボールと二つのゴールから構成されている．また，周囲は壁で覆われており，ボールがフィールドからはみ出すことはない．フィールドは RoboCup の中型リーグのサイズを想定した．また，使用するロボットは我々がこれまで用いてきたロボット (図 2 (b) 参照) を想定した．各ロボットは一つのカメラを中心部に搭載し，カメラから得られた視覚情報をもとに行動する．また，行動系は PWS (Power Wheeled Steering) を採用し，左右輪を独立に制御できる．つまり，各ロボットは直進，旋回，およびその組み合わせで行動することが可能である．

3.2 関数集合と終端集合の設計

GP を実際の問題に適用する際，関数・終端集合をどのように設計するか，ということは非常に重要な問題である．一つの実装方法として，状態空間 $\boldsymbol{X}$ から行動空間 $\boldsymbol{A}$ への写像を GP によって獲得させることが考えられる．しかし，GP 単独でこの写像関数を求めることは非常に効率が悪い．また，これまでに獲得された幾つかの基本ビヘービア (シュート，衝突回避，パスなど) を再利用したい．

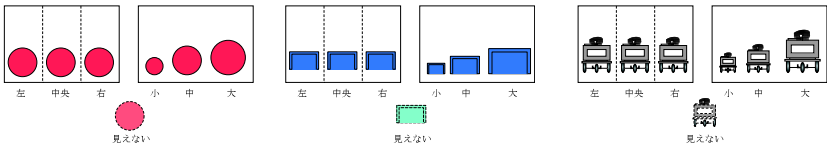
そのため，ここでは基本ビヘービアの状況による切り替えを GP によって学習させることを考える．ここでビヘービアとは，ある目標を達成するための行動のシーケンスのことであり，

$$f_i : \boldsymbol{X}_i \rightarrow \boldsymbol{A},$$

という形式で与えられる．今回の実験では，以下の 4 種類のビヘービア

1. `shoot`： ボールを敵陣のゴールにシュートするビヘービア．このとき，他のロボットは無視している．
2. `pass`：味方のいる方向にボールを蹴るビヘービア．
3. `avoid`：他のロボットとの衝突を回避するビヘービア．ただし，敵味方の区別はしない．
4. `search`：ボールをその場で回転することによって探索するビヘービア．

[1]それぞれ，パッサー (passer)，シューター (shooter)，ディフェンダー (defender) の役割が期待されているとの意味合いで，p, s, d の添え字を付けている．



(a) ボールの分類　(b) ゴールの分類　(c) 他のロボットの分類

図 3: 学習者が観測できる情報

を準備した．これらのビヘービアは，筆者らによって既に提案された強化学習法 [1, 19] によって獲得されたものであり，概要を付録 A に示す．これらのビヘービアの名前は便宜的なものである．例えば，ある状況において全てのビヘービアが出力する行動が同一である場合も存在する．また，`shoot` ビヘービアはボールが見えないからといって停止するわけではなく，何らかの行動 (状況によってはボールを探索する行動) を生成することに注意されたい．

関数部分では，状況による場合分けが必要になる．ここでは，基本ビヘービアの切り替えの条件として，各ロボットが観測しうる画像情報を用いる．図 3 に，画像情報の分類を示す．関数セットとしては，単純な制約付き条件分岐関数 “IF a b, then c, else d” を準備した．ここで a はボール，二つのゴール，2 台の他ロボットの計 5 種類である．また，b は右，中央，左，小，中，大，見えない，の 7 種類である．a が b に属する場合には c を評価し，そうでなければ d を評価する．

図 4 は獲得される木の例である．このような形状の木を獲得する手法として，C4.5 などの決定木を用いる方法 [15] が提案されている．この方法では，ある状況で事前に全てのデータをオフラインで収集した後で木を作成するため，GP を用いた方法よりも効率的な木が作成されると考えられる．しかし，(1) 評価が成功/失敗の 2 値でしか表現されないこと，(2) 決定木を用いてビヘービアを選択した後は，選んだビヘービアを取り続けるだけで，様々な状況で適切に行動するためには，新たに木を作成するか，別の方法が必要になる．一方，GP を用いた場合，木は常に評価され[2]，様々な状況を経験することで，ゲーム全体に適用できる木を獲得できる，といった違いがある．

3.3 適合度関数の設計

進化的手法を適用する際のもう一つの重要な点として，適合度関数の設計があげられる．今回，0.0 が最善となるように適合度関数を設計した．また，味方同士の協調や敵との競争行動を評価するために，以下の要素を考慮した．

- 得点数 $G(i)$: ロボット i が所属するチームの総得点数．多いほうが良い．

[2]今回の場合は 1/30 [msec] ごとに評価される．

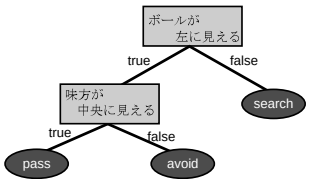


図 4: 獲得される木の例

- 失点数 $L(i)$: ロボット i が所属するチームの総失点数，少ないほうが良い.

ここで，各個体が体験する試行回数を T_{max} とすると，

$$G(i) + L(i) \leq T_{max}, \tag{1}$$

が成り立つ.

しかし，これらのパラメータだけで実験をおこなったところ，自分自身だけで目標を達成する 1 台のロボットを除いて，他のロボットは邪魔しない程度の消極的な協調行動しか発現しなかった．そこで，より個体レベルに近い評価 (衝突をできるだけ避けながら，他のロボットと相互作用することを薦める) を考慮するために以下の要素を与えた.

- キック数 $K(i)$: ロボット i がボールをキックした回数.
- 衝突回数 $C(i)$: ロボット i と他のロボットとの衝突回数．少ないほうが良い.

また，ロボットがゴールにより早く到達できるように，タスク達成に要した時間に関する評価を導入した.

- ステップ数 $steps$: 一試行が終了するまでのステップ数．少ないほうが良い．一ステップは，センサ情報に対してモータコマンドを送信する繰り返しのインターバルで，実ロボットの場合で 1/30 [msec] である.

ここで，一試行はボールがゴールに入った時，もしくは事前に決めたステップ数を超過した時に終了するとしている.

これらのパラメータの線形結合で適合度関数が構成される．最終的にロボット i が受け取る適合度は,

$$\begin{aligned} f_s(i) &= \alpha_k h(K(i), \beta) + \alpha_g h(G(i), T_{max}) + \alpha_l * L(i) \\ &\quad + \alpha_c * C(i) + \alpha_s * steps \\ h(x, y) &= \begin{cases} y - x & \text{if } x < y \\ 0 & \text{otherwise} \end{cases}, \end{aligned} \tag{2}$$

ここで $\alpha_k \sim \alpha_s$, β は重みパラメータである．以下の実験では，予備的な実験に基づき $\alpha_k = \alpha_g = 1$, $\alpha_l = 5.0 \times 10^{-1}$, $\alpha_c = 5.0 \times 10^{-2}$, $\alpha_s = 1.0 \times 10^{-4}$, $\beta = 10$ と設定した．もし，複数のロボットが等価な適合度を持った場合，より短い木を持つほうを良い個体とする．

3.4 GP の実装

具体的な GP の実装について述べる．GP における遺伝的オペレータは次のものを採用する．図 5 に現在の世代から次の世代の個体集合の生成方法を示す.

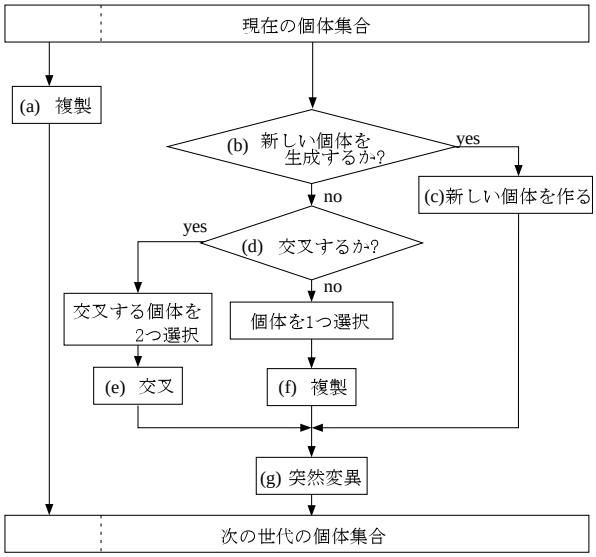
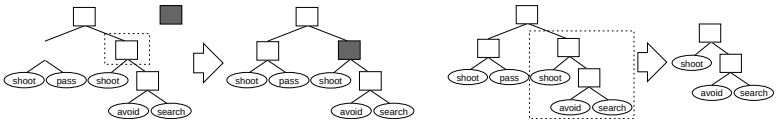


図 5: 世代交代の方法

集団の大きさ : 各ロボットは別々に個体集団を持つ．個体集団のサイズを $Num(= 80)$ とする．
初期集団の生成 : 木の深さが 10 になるように一様に初期個体を生成する．
(a) 選択 : エリート保存戦略 [6] を採用する．つまり，現在の世代で最善の個体は無条件で次の世代に残す．他の個体についてはサイズ 10 のトーナメント戦略を用いる．
(b) 探索 : 確率 $p_{cr}(= 0.05)$ で新規に個体を生成する．この時の木の深さは，初期集団を生成する時と同じ 10 である．
(d), (e) 交叉 : 確率 $p_c(= 0.95)$ で交叉させる．集団からトーナメント戦略により二つの個体 (親) を選択した後，1 点交叉により二つの子を生成し，次の個体とする．ここでメモリの制約から，交叉によって生成される木の深さ



(a) 交換による突然変異　　(b) 縮退による突然変異

図 6: 突然変異の方法

は 25 を超えないように制限する．

(d), (f) 複製：確率 $p_r = 1 - p_c$ で個体を無条件に次の世代へコピーする．

(g) 突然変異：確率 $p_m(= 0.05)$ で，選択された個体を突然変異させる．突然変異の方法としては，図 6 (a) に示す条件分岐文を交換するタイプと，(b) に示す部分木を取り出すタイプのどちらかを用いる．

各個体はそれぞれ $T_{max}(= 20)$ 回ずつゲームに参加する．結果として，1 つの世代を評価するためには，$Num \times T_{max}(= 1600)$ 試行を要することになる．また，世代交代の回数は 60 回と限定した．シミュレーションは DEC VT Alpha 600 上で実施し，一回の実験を終了するために約 16 時間を要した．

4　シミュレーション結果

4.1　2 台の学習者の場合

最初に協調関係にあるロボットだけが存在する環境で，協調行動が獲得されるかどうかを実験した．つまり，図 2 (a) において，ロボット R_d を取り除いた R_p, R_s だけで実験をおこなった．関数の個数は

$$7\ (\text{ボール}) + 2 \times 7\ (\text{ゴール二つ}) + 7\ (\text{他のロボット}) = 28$$

である．図 7 に 2 台のロボットによる共進化実験の結果を示す．ここで，各値は個体数で平均したものを示している．この実験では，20 試行中に約 17 回ほど得点することができた ((a) 参照) が，設計者が想定したパス行動は観測されず，図 8 のような行動が得られた．図 8 のゲームの解析結果を以下に示す．

1. R_p は初期配置でボールが近くにあるにもかかわらず，`search` ビヘービアと `avoid` ビヘービアを交互に繰り返し，その場回転している行動が観測された．
2. その間に R_s がボールに近づき，R_p にパスする行動をとった．結果的にボールは R_p と衝突してゴール前に転がった．
3. R_p は `shoot` ビヘービアを実行しているが，R_s が先にボールに近づき `shoot` ビヘービアによってボールをゴールにシュートした．

このことから R_p は得点に直接的には貢献しなかったと言える．実際，図 7 (b) に示した通り，ボールを蹴る回数は R_p の方が少なかった．

R_s の行動だけが改善された原因として，以下のことが考えられる．

- R_p が R_s にパスして R_s がシュートした時点で始めて協調行動が成立する．そのためには，ロボット間の行動が正確に同期しなければならない．このような正確な同期を探索することは，非常に困難な問題である．
- R_p のパス行動とはあまり関係なく，R_s は自分で得点できる．すなわち，R_s は R_p の助けを必要としない．つまり，R_s のタスクが R_p に比べて簡単なためである．

結果として，R_p の技術が向上しない間に R_s がこのタスクを支配することになってしまった．つまり，2 節で説明した 2 番目のパターンに落ち着いたと考えられる．

図 9 に，2 台の学習者による進化結果を実ロボット上への実装した場合の行動を示す．当然だが，1 台のロボットのみがシュート行動をとり，他は静止しており，積極的な協調行動は生まれていない．これは**パターン B** に相当すると考えられる．

4.2　2 台の学習ロボットと 1 台の静止ロボットの場合

4.1 節で述べた通り，協調関係にある 2 台だけで共進化させようとしても，設計者が考えるところの協調行動は発生しなかった．そこで，静止障害物として R_d を追加して R_s がシュートしにくい状況 (シュートコースを塞ぐ) を設定し，4.1 節と同様の実験をした．用いた関数の個数は

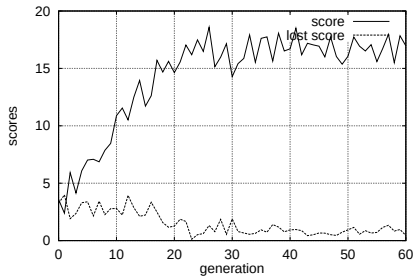
$$7\ (\text{ボール}) + 2 \times 7\ (\text{ゴール二つ}) + 7\ (\text{他のロボット 2 台}) = 35$$

である．

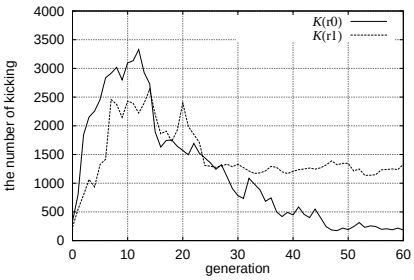
図 10 に共進化実験の結果を示す．先の 2 台だけの実験と異なり，R_s は静止ロボット R_d の存在により，4.1 節で獲得したような木ではタスク遂行が困難となり，R_p との同期した行動が必要となる．R_s のタスクの複雑さが増し，R_p のタスクの複雑さに近づいたといえる．

進化の歴史を次に示す．

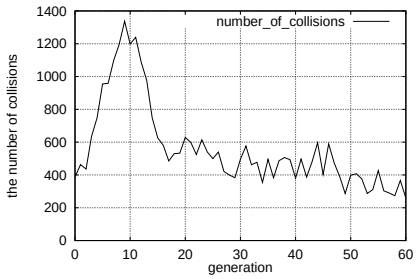
- 第 4 世代まで：双方にボールを蹴る個体が発生する．R_p の方の個体集合に R_s の方へパスする個体が幾つか観測される．しかしながら，そのパスがあまり正確でないため，R_s はシュートできない．
- 第 15 世代まで：R_s は壁に沿って相手ゴールまで行く行動を獲得した (図 11 参照)．このビヘービアが世代を経るごとにに支配的になる．



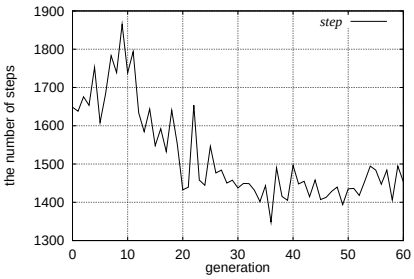
(a) 得失点の推移



(b) ボールを蹴った回数の推移



(c) 衝突回数の推移



(d) ゲーム終了までのステップ数の推移

図 7: 2 台の学習者による共進化の実験結果

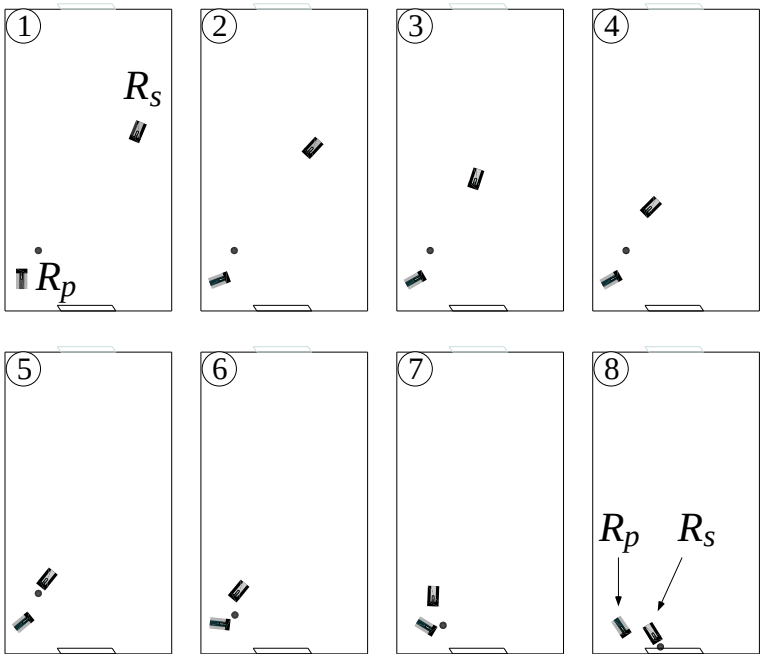
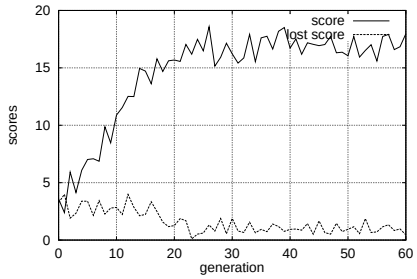
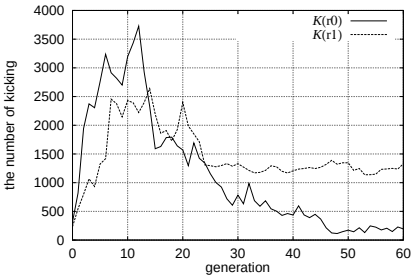


図 8: 2 台の学習者による共進化実験で得られた行動の例

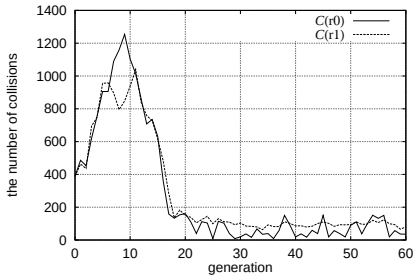
図 9: 2 台の学習者による進化結果の実ロボット上への実装



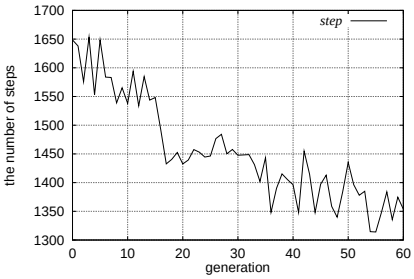
(a) 得失点の推移



(b) ボールを蹴った回数の推移



(c) 衝突回数の推移



(d) ゲーム終了までのステップ数の推移

図 10: 2 台の学習ロボットと 1 台の静止ロボットによる共進化の実験結果

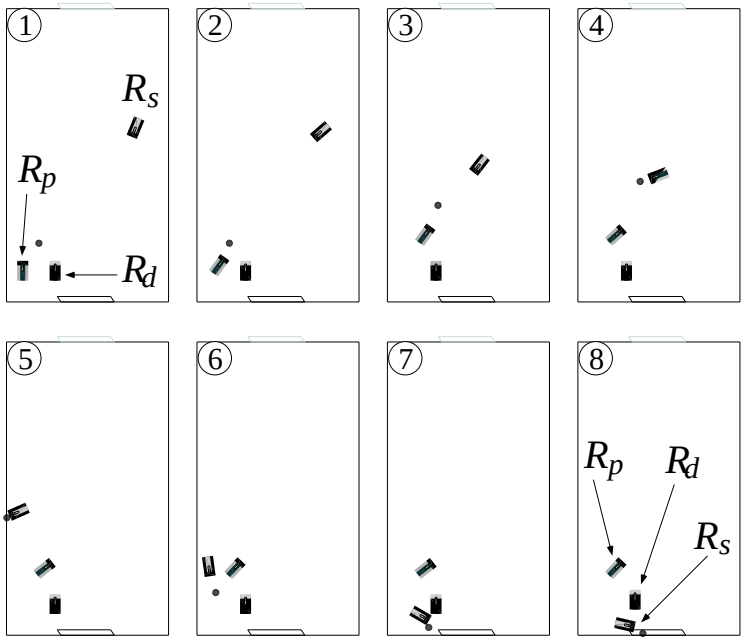


図 11: R_s が壁沿いにシュートする行動の例

- 最終結果：R_p のパスの精度が上がった協調行動が発現する．つまり，R_p が R_s にパスし，R_s がシュートする行動が得られた．

結果として，R_p と R_s が同期して技術を向上しており，これは2 節で説明した 3 番目のパターンのひとつと考えられる．

図 12，13，14 に，実ロボット上への実装結果の例を示す．図 12 では，タイミングが合わずパスがうまくいかなかった[3]．一方，図 13 では静止ロボットに邪魔されてパスせざるを得ない状況が発生し，うまくパスした様子がうかがえる．図 14 は同様の別のシーンでゴール後ろから撮影したものである．

4.3 3 台の学習者の場合

最後に，3 台のロボット R_p, R_s, R_d を同時に学習させた場合について述べる．これは 4.2 節で追加された静止ロボット R_d が他の 2 台と同時に進化するケースである．4.1 節や 4.2 節との違いは，ロボット R_d と ロボット R_p, R_s との競争が含まれることである．関数の個数は 4.2 節と同じである．また，R_d に対しても R_p, R_s と同一の適合度関数 (式 (2)) を用いて実験した．そのときの結果を図 15 に示す．

この実験では，次に示す二つのタイプの行動パターンが見られた．一つは，4.2 節と同じで、R_p がボールを R_s の方向にキックし，R_s が R_d との衝突を避けながらシュートする場合である (図16 参照)．もう一つは，R_d がボールをインターセプトし，ボールを奪って敵ゴールにシュートする場合である (図17 参照)．その割合は，図15 (a) に示す通り，約 1:3 となり，台数的に有利な R_p, R_s の組が負けるという結果が得られた．

1 台対 2 台という不利な条件にもかかわらず，R_d がゲームに勝つ確率が多い原因として，次のことが考えられる．R_d は最初からボールと相手ゴールを同時に視野に収め，自分だけでシュートを決めることが可能であり，`shoot` ビヘービアはこのような状況に適していた．一方，R_p は単独でシュートするか R_s にパスするかの選択があるが，必要なステップ数や R_d にインターセプトされる可能性が高いことから，R_s にパスせざるを得ない．そのタスクの複雑さが，R_d より複雑であるためと予想される．

最後に，この実験で各ロボットが獲得した木を解析した結果について述べる．図 18 に獲得された木の一部を示す．要約すると，

- R_d の場合：R_d がゲームに勝った時は，オフェンス行動 (図 18 (a) 参照) が多くの状況下で選択されていた．`shoot` や `pass` といった準備したビヘービアが同様な条件で獲得されており，このことが GP によってコンパクトな表現を獲得できたことにつながっていると考えられる．

[3]計算機上でのシミュレーションでは，フィールドが壁で囲われているため，リバウンドボールを処理することが可能であったが，実環境は壁がないため，そのままフィールド外にボールが出た時点で試行を中断している．

時刻 1

時刻 2

時刻 3

時刻 4

時刻 5

時刻 6

図 12: 2 台の学習者と 1 台の静止障害物を含む場合の結果の実ロボット上への実装 (I)

図 13: 2 台の学習者と 1 台の静止障害物を含む場合の結果の実ロボット上への実装 (II)

時刻 1

時刻 2

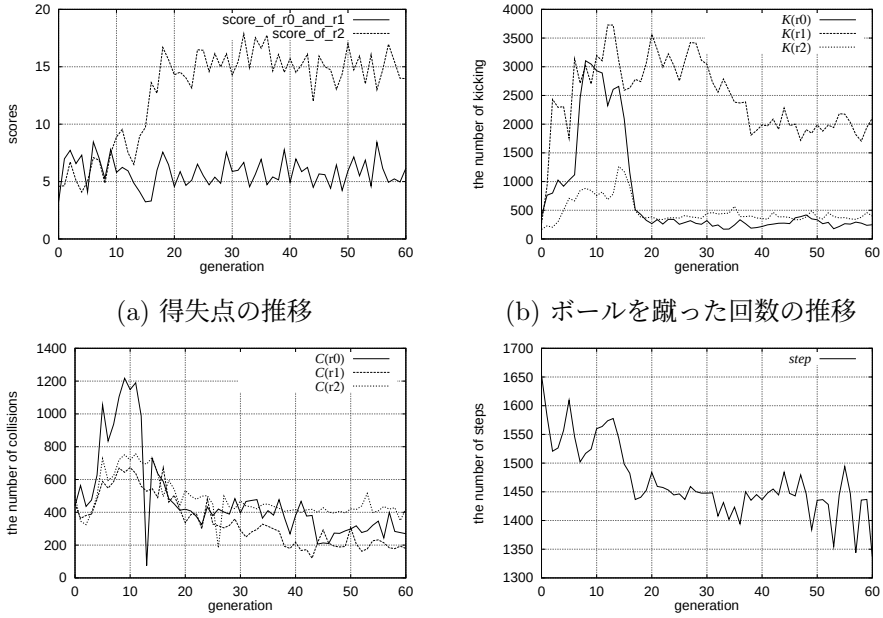
時刻 3

時刻 4

時刻 5

時刻 6

図 14: 2 台の学習者と 1 台の静止障害物を含む場合の結果の実ロボット上への実装 (III)



(a) 得失点の推移　(b) ボールを蹴った回数の推移

(c) 衝突回数の推移　(d) ゲーム終了までのステップ数の推移

図 15: 3 台の学習者による共進化の実験結果

- R_p の場合：図 18 (b) から，R_p から R_s へのパス行動が成功したときに，pass ビヘービアが選択されていることがわかる．この理由として，今回準備した pass ビヘービアはこのタスクに類似した状況で獲得されていたためと考えられる．また，R_p は R_s が視野内に無い場合，ボールを捜してシュート行動を選択する行動を獲得している．

しかしながら，R_s の獲得した木には R_p や R_d のような構造は見られなかった．言い換えると，今回準備した終端セットは R_s が利用するには適切でなかったと考えられる．

5　考察

5.1　タスクの複雑さについて

前節までの実験で，各ロボットに与えられたタスクの複雑さが，適切な共進化 (相互のスキル向上) を導く上で重要であることを実験的に示した．しかしながら，一般的なタスクの複雑さを定義することは極めて困難である．事前に考えられる複雑さの指標として，次のようなことが考えられる．

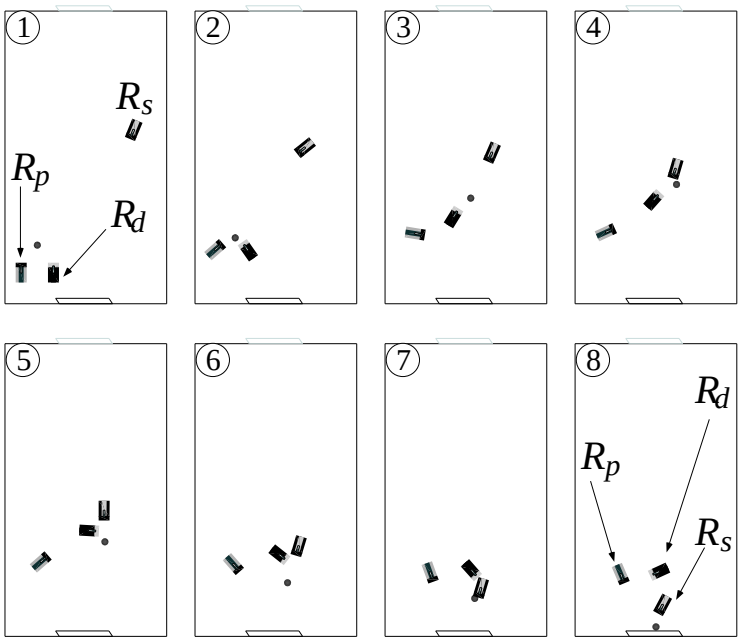


図 16: 獲得されたビヘービアの例 (I)：R_p と R_s が R_d を相手にシュートを決めた場合

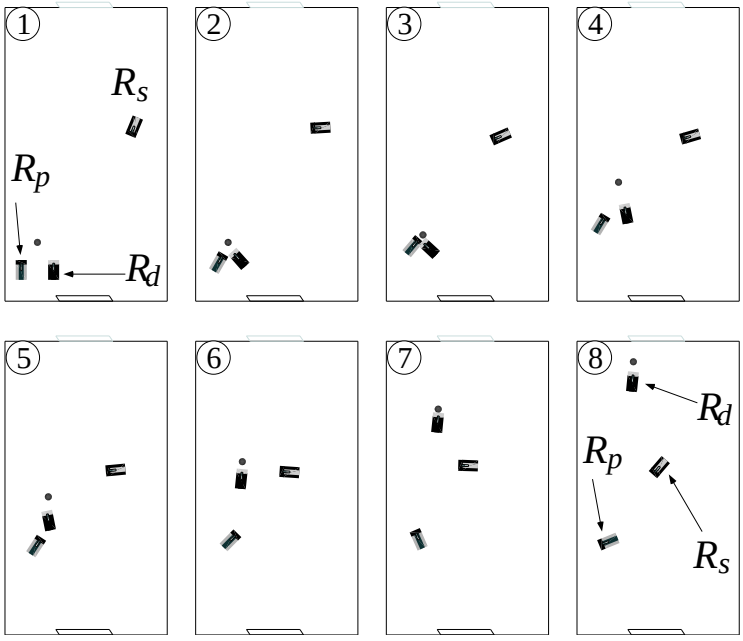
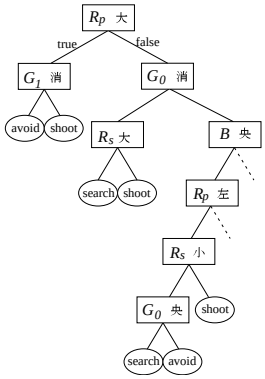
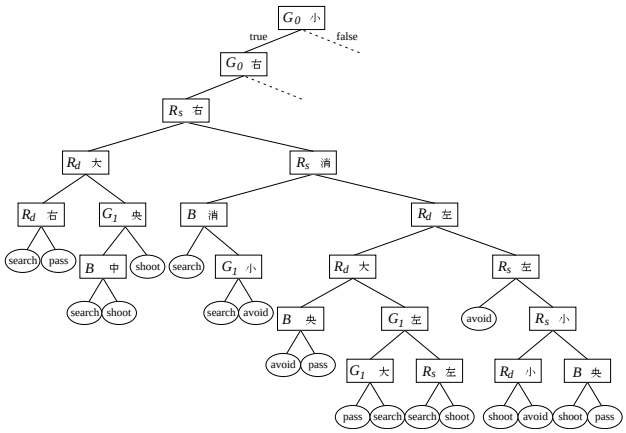


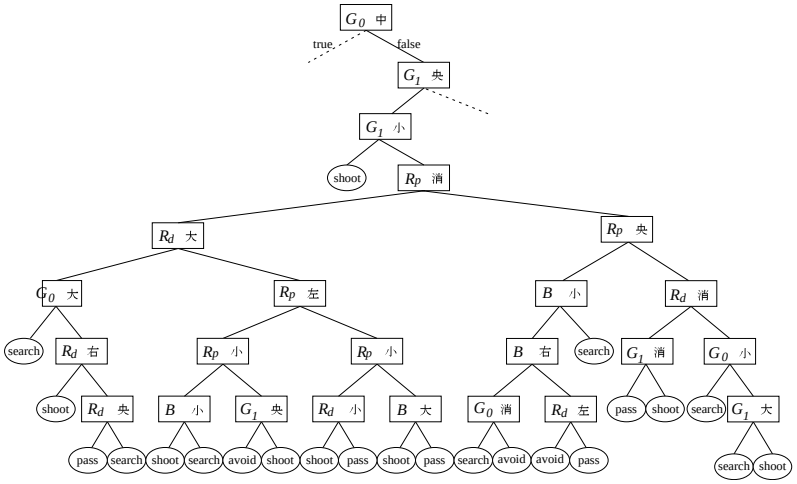
図 17: 獲得されたビヘービアの例 (II)：R_d が相手ゴールにシュートを決めた場合



(a) R_d の獲得した木



(b) R_p の獲得した木



(c) R_s の獲得した木

左	左に見えている	小	小さく見えている
央	中央に見えている	中	中ぐらいに見えている
右	右に見えている	大	大きく見えている
消	見えない		
G_0	守るゴール (自陣ゴール)	G_1	攻めるゴール (敵陣ゴール)
B	ボール		

(d) 図中の記号

図 18: 3 台の学習者が獲得した木の一部．ここで四角は条件分岐部，楕円は 3.2 節で述べたビヘービアである．

状態空間について　状態空間の次元が増大 (要素数が増加) すれば，それだけ探索空間も増大する．逆に，次元が小さい (要素数が少ない) 場合，必要な状態が観測されない状況が生じ，結果的に複雑さは増大する．また，タイルワールドなどの理想的な状況では，コンパクトな状態表現が利用できるのに比べ，実環境を想定した今回のタスクでは，状態空間の構成問題は非常に重要な問題である．

行動空間について　今回用いたロボットは 2 自由度であったが，より多くの自由度を持ったロボット，例えば脚式ロボットを用いた場合には，問題は更に複雑になる．また，2 自由度であっても，我々の用いたロボットは非ホロノミックの制約[4]を受ける点において，タイルワールドで 4 方向に移動できるロボットと比べて複雑である．

評価関数について　目標値との誤差といった形で評価値が常にフィードバックできる時には，問題は比較的簡単になるが，評価が遅延している場合には，それだけ問題は複雑になる．また，5.2 節でも述べるが，達成すべき目的が多くなれば，そのトレードオフを考慮しなければならない．

初期配置について　目標状態に近い状態から学習を始めた場合，学習に必要な時間が短縮されることが報告されている [1, 13].

5.2　適合度関数について

最適化問題として考えた時，このタスクは 3.3 節に示した通り 5 つの目的の多目的最適化問題とみなすことができる．多目的最適化の問題を扱う方法は幾つか提案されており，それらの比較もなされている [18]．今回は実装の容易さから，式 (2) のような複数の関数の重み付け線形和を適合度関数として用いている．重みの決定方法は極めて重要であり，今回は全世代を通して重みは一定の値を取るように設定した．

実ロボットに適用する際に問題となるのはロボット間での衝突であり，衝突回数 $C(i)$ の最小化が最も優先される．しかし，$C(i)$ に対する重み α_c を最初から大きな値に設定すると，全てのロボットがその場で回転する行動が得られた．つまり，全てのロボットが静止していれば衝突することもなく，この目的は達成されるが，これは我々が望む行動ではない．そのため，α_c は他の重みと比べて小さ目の値を設定したが，実験結果に示す通り，$C(i)$ は 0 にはなっていない．

以上のことから，より適切な行動を獲得するために，

[4]今回の実験では，ロボットへの制御入力は 2 次元であるのに対し，平面内の位置と姿勢の 3 自由度を制御しなければならない．例えば，ロボットが横に移動するためには，何回かの切り返しを必要とする．

- 進化の初期段階では，ロボット間の衝突をある程度容認して，ボールを蹴ることを優先させる．
- ある程度得点することができるようになってから，ロボット間の衝突を考慮する

といった方針が必要になる．そのため，重みを進化の途中で変更するような枠組みを現在開発中である．

6　おわりに

本研究では，競合タスクだけでなく協調タスクにおいても，複数ロボットを共進化できることを，2 ないし 3 台のロボットによる簡単なサッカーゲームによって示した．協調エージェントを共進化させるためには，ロボットが正確に同期して進化する必要があることである．これは，環境自身も単純な状況からより複雑なものへと共進化することで，協調エージェントの共進化を助けることができる点を示唆する．

望みの共進化を導くためには，何が必要で十分な条件であるかを明確にするためのシステマティックな理解が必要と考えられる．また，今回は計算機シミュレーション上で獲得した結果を実ロボットに適用しただけであり，シミュレーションと実環境とのギャップは考慮されていない．終端として用いたビヘービアは実環境での改善 (微調整) が可能である [19] が，多大な試行回数を要する GP の場合に，ギャップをどう扱うかは今後の課題である．

謝辞　本研究を始めるにあたり，多くの議論，助言を頂いた Maryland 大学の Sean Luke 氏に深く感謝致します．また，本研究は日本学術振興会未来開拓学術研究推進授業，知能情報・高度情報処理研究分野「分散協調視覚による動的3次元環境理解」プロジェクト (プロジェクト番号 : JSPS–RFTF 96P00501) の援助を受けた．ここに感謝の意を表します．

付録 A 終端に用いた基本ビヘービアについて

ここでは，GP を適用する際に終端ノードとして用いた基本ビヘービアの獲得方法について簡単に述べる．詳細は他の文献 [1, 19] を参照されたい．

強化学習の基礎

強化学習 [16] とは教師なし学習の一種である．ここでは，もっとも代表的な学習法であり，本研究でも用いている Q 学習 [20] について簡単に説明する．Q 学習では学習者を含めた環境がマルコフ過程としてモデル化できる時，環境との相互作用を繰り返すことで最適な行動を獲得できる．

現在の状態 x_t において行動 u_t を選択し，環境が次の状態 x_{t+1} に遷移し，報酬 r を受け取った時，行動価値関数 $Q(x,u)$ は

$$Q_{t+1}(x_t,u_t) = (1-\alpha)Q_t(x_t,u_t) + \alpha(r + \gamma \max_{b \in \boldsymbol{A}} Q_t(x_{t+1},b)), \qquad (3)$$

と更新される．ここで，α は学習率，γ は減衰係数である．また，$\boldsymbol{A}$ は行動集合である．式 (3) を用いて状態が遷移するたびに $Q(x,u)$ を更新することにより，最適な行動政策を獲得できる．学習が収束した後は，最大の行動価値を持つ行動，つまり，

$$u^* = \arg\max_{b \in \boldsymbol{A}} Q(x_t,b), \qquad (4)$$

となる行動 u^* を選択すれば良い．

基本ビヘービアの獲得

強化学習を適用する際には，状態空間 $\boldsymbol{X}$ と行動空間 $\boldsymbol{A}$ を定義する必要がある．行動空間は全てのビヘービアについて同一であり，各ロボットへの制御入力は

$$\boldsymbol{u}^T = \begin{bmatrix} v & \omega \end{bmatrix}, \quad v, \phi \in \{-1, 0, 1\}, \qquad (5)$$

として表現できる．ここで，v は台車の移動速度であり，ω は角速度である．結局，実際に選択できる行動の合計は 9 通りである．状態空間は，4 種類のビヘービアそれぞれに関して設計している．

今回用いた 4 種類のビヘービアのうち，`shoot`, `avoid`, `pass` ビヘービアについては局所予測モデル [19] によって状態ベクトルを推定した後で，強化学習を適用することにより獲得した．以下に，学習した時の条件について簡単に示す．

`shoot` について　環境には学習者一台とボールが存在している．学習者が観測するのはボールとゴール二つだけである．ボールがゴールに入った時に正の報酬が学習者に与えられる．

`avoid` について　環境中には学習者一台と，固定政策に従って行動するロボットが存在しており，ボールは存在しない．学習者が観測するのは他のロボットだけである．ロボット間で衝突した場合に負の報酬が学習者に与えられる．

`pass` について　環境には学習者一台と `shoot` ビヘービアを再現しているロボット，更にボールが存在する．学習者は味方にボールをパスした時に正の報酬が与えられる．

search について　ボールを搭載したカメラ画像中心に観測させるような単純な制御則をロボットに実装した.

参考文献

[1] M. Asada, S. Noda, S. Tawaratumida, and K. Hosoda. Purposive Behavior Acquisition for a Real Robot by Vision-Based Reinforcement Learning. *Machine Learning*, 23:279–303, 1996.

[2] M. Asada, S. Noda, and K. Hosoda. Action based Sensor Space Segmentation for Soccer Robot Learning. *Applied Artificial Intelligence*, 12(2-3):149–164, 1998.

[3] D. Cliff and G. F. Miller. Co-evolution of pursuit and evasion II : Simulation methods and results. In *Proc. of the 4th International Conference on Simulation of Adaptive Behavior: From Animals to Animats 4*, pages 506–515, 1996.

[4] D. Floreano and S. Nolfi. Adaptive behavior in competeing co-evolving species. In *Fourth European Conference on Artificial Life (ECAL97)*, pages 378–387, 1997.

[5] 伊庭斉志. 遺伝的プログラミングによるマルチエージェント学習. 北野 (編), 遺伝的アルゴリズム 3, 第 12 章, pages 299–335, 産業図書, 1997.

[6] 北野宏明 編. 遺伝的アルゴリズム. 産業図書, 1993.

[7] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, and H. Matsubara. "RoboCup: A challenge problem for AI". *AI Magazine*, Vol. 18, pp. 73–85, 1997.

[8] John R. Koza. *Genetic Programming I : On the Programming of Computers by Means of Natural Selection.* MIT Press, 1992.

[9] John R. Koza. *Genetic Programming II : Automatic Discovery of Reusable Programs.* MIT Press, 1992.

[10] S. Luke. Genetic Programming Produced Competitive Soccer Softbot Teams for RoboCup97. In *Proc. of the Third Annual Genetic Programming Conference (GP98)*, pp. 204–222. Morgan Kaufmann, 1998.

[11] T. Matsuyama. Cooperative Distributed Vision – Dynamic Integration of Visual Perception, Action, and Communication –. In *Proc. of Image Understanding Workshop*, Monterey CA, 11 1998.

[12] S. Nolfi and D. Floreano. Co-evolving predator and prey robots: Do 'arm races' arise in artificial evolution? *Artificial Life*, 4:311–335, 1998.

[13] 小俣，松平，井上．力学的補助による段階的な学習方法．日本ロボット学会誌，16(8):1069–1075, 1998.

[14] RoboCup web page. http://www.robocup.org/

[15] P. Stone and M. Veloso. Using Machine Learning in the Soccer Server. In *Proc. of IROS-96 Workshop on RoboCup*, 1996.

[16] R. S. Sutton and A. G. Barto. *Reinforcement Learning*. MIT Press/Bradford Books, March 1998.

[17] 高橋，浅田．実ロボットによる行動学習のための状態空間の漸次的構成．日本ロボット学会誌，17(1):118–124, 1999.

[18] 玉置．遺伝的アルゴリズムと多目的最適化．北野 (編)，遺伝的アルゴリズム 2，第 3 章，pages 71–87, 産業図書，1995.

[19] E. Uchibe, M. Asada, and K. Hosoda. Cooperative Behavior Acquisition in Multi Mobile Robots Environment by Reinforcement Learning based on State Vector Estimation. In *Prof. of IEEE International Conference on Robotics and Automation*, pages 1558–1563, 1998.

[20] C. J. C. H. Watkins and P. Dayan. Technical note: Q-learning. *Machine Learning*, pp. 279–292, 1992.