ゲーム内エージェントの強化学習に用いる報酬値探索へのＧＡの適用

井上, 勇気 / INOUE, Yuhki

(出版者 / Publisher)
法政大学
(発行年 / Year)
2009-03-24
(学位授与年月日 / Date of Granted)
2009-03-24
(学位名 / Degree Name)
修士(理学)
(学位授与機関 / Degree Grantor)
法政大学 (Hosei University)

2009 年　修士論文

ゲーム内エージェントの強化学習に用いる報酬値探索への GA の適用

Applying GA for Searching Reward Value in Reinforcement Learning of Agents in Video Games

法政大学　情報科学研究科　情報科学専攻　佐藤　研究室

07T0001

井上　勇気

指導教授：佐藤　裕二

目次

ゲーム内エージェントの強化学習に用いる報酬値探索への GA の適用

Applying GA for Searching Reward Value in Reinforcement Learning of Agents in Video Games

井上　勇気
Inoue Yuhki
法政大学大学院情報科学研究科情報科学専攻
Email: yuhki.inoue.z2@gs-cis.hosei.ac.jp

概要

This paper describes our study applying GA to search the reward values for reinforcement learning in a soccer video game using learning classifier systems. In particular, we report the result of promotion of efficiency by dividing searched space that considered dependence between the reward values and searching the divided space alternately. We have already proposed that acquiring algorithms by using the event-driven hybrid learning classifier system. Moreover, we have proposed that using GA for setting the reward values which have no index for setting. As the result, a probability that the reward values can be set to appropriate value for learning was obtained. In prior studies, certain rewards became searching candidate. Meanwhile, if the kind of the success rewards increase, setting of success rewards is difficult in practical time because the search space of reward values become spread. To address this problem, in this paper, we propose that applying the technique of dividing the searched space that considered dependence between the reward values, and searching divided space alternately with exchange information. We show a possibility that proposal technique have effect to improve efficiency of learning the reward values.

1. はじめに

インターネットの普及に伴い,インターネットを介してプレーするビデオゲームが広がっている．一つのビデオゲームをゲームに慣れている人から，なれていない人まで，様々なレベルの人がプレーしている．一方，インターネットの普及によって，インターネットを介したビデオゲームの環境に変化が起きている．起きている変化として，多くの人が，インターネットを介したビデオゲームに参加しやすくなったこと，大人から子供まで，様々な熟練度の人が参加し，熟練度も変化することなどが上げられる．以上の変化によりビデオゲームの開発に問題が引き起こされている．

起きている問題として,プレイヤーの熟練度の変化によりアルゴリズムの寿命が短くなっていること，多くのプレイヤーの参加によって，プレイヤーの戦略に合わせた，エージェントの意思決定アルゴリズムが必要になり,ゲームの環境が複雑になっていることが上げられる．

ビデオゲーム開発の現場においては,ゲームアルゴリズムの開発は人手によって行なわれ,アルゴリズムの記述の方法として，IF-THEN 形式によるルールを用いることが一般的である．IF-THEN 形式を用いる利点として，デザイン，メンテナンスのしやすさがある．

上記の IF-THEN 形式によるアルゴリズムとの整合性のよさを考慮し,我々は前述の問題を解決する方法として，サッカービデオゲームを題材にし，クラシファイアシステム[1]を適用したゲーム内のエージェントに,バケツリレーアルゴリズムとイベントの出現頻度による学習を組込んだ強化学習を適用して行動アルゴリズムを獲得する手法[2]を提案した．また,学習に必要となる報酬値をゲーム内の役割によって偏りを持たせて効率的に学習させる方法[3,4]を提案した．一方で，報酬値の設定には指標が無く，経験に基づく試行錯誤によって決定しなくてはならないという問題がある．報酬値設定の問題を解決する手段として，遺伝的アルゴリズム（GA）による報酬値の探索を提案[5]した．上記提案では探索空間の広さの問題からエージェントの行動に対して与えられる 6 種類の報酬値だけについて探索を行い,効率的学習ができる報酬値を GA による探索によって見つけることができる可能性を示した．一方，先の方式では探索対象とする報酬値の種類が増加した場合，探索空間が広がるため,報酬値の学習が実用的な時間内で完了しないという問題点が残った．問題の解決のために本稿では,探索空間を 2 つに分割し,分割した探索空間を交互に探索しながら情報交換を行う手法を提案する．各プレーの最適な報酬値はそれぞれ依存関係が大きいものと，その依存関係が小さいものがあると考えたからである．以下では，2 章でサッカービデオゲームの概要と従来の研究について述べる．3 章では我々の提案である，報酬値を 2 つに分割して探索する手法について述べる．4 章では実験方法と実験結果を示し，5 章で考察を述べ，6 章でまとめを述べる．

2. サッカービデオゲームと問題点

2.1. サッカービデオゲーム

我々が扱うサッカービデオゲームは，コンピュータ上に，2 つのチームで点の取り合いを行なうサッカーを表現したソフトウェアである．サッカーゲームの概観を図 1 に示す．サッカーゲームでは 1 チーム 11 人のプレイヤーがおり，2 チーム合計で 22 人のプレイヤーがいる．ビデオゲームでは一般的に 22 人全員分のゲームプレイヤーがそろうことは無く，不足分をコンピュータが操作するプレイヤーで補われる．本稿では，22 人全てのプレイヤーをコンピュータによって操作する．

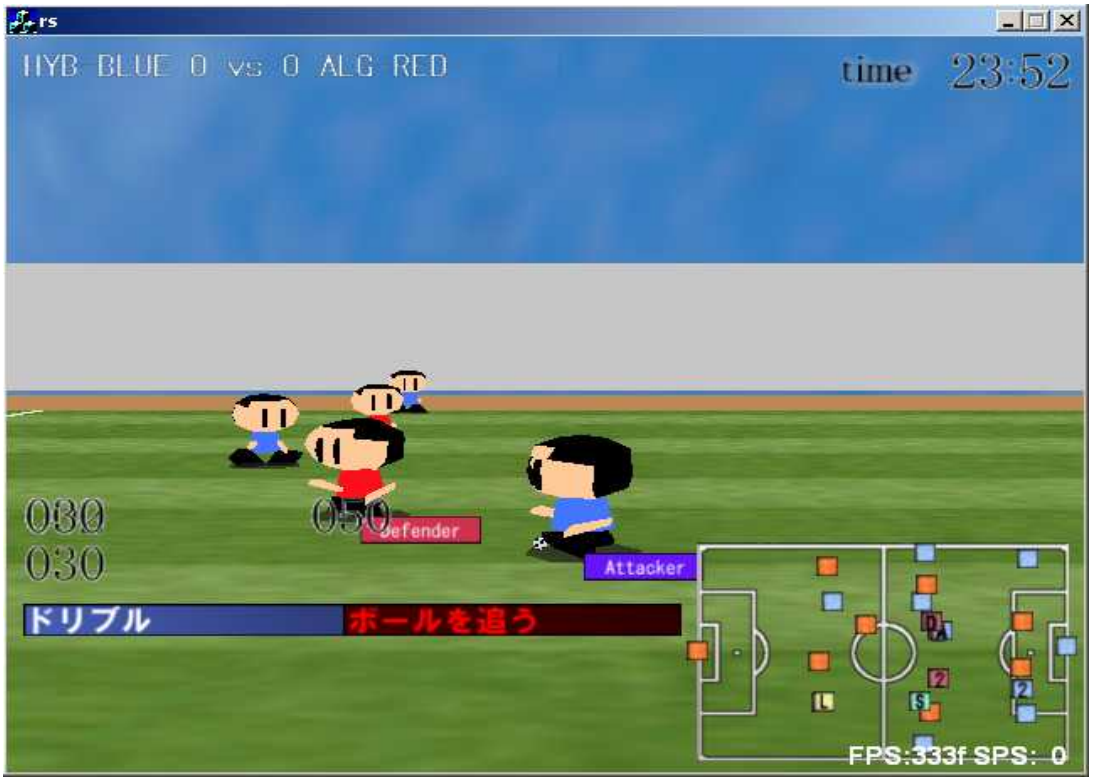

図 1 サッカーゲームの概観

2.2. エージェント

サッカービデオゲームにおけるエージェントは次の 3 つの部分から構成されている.

・認識器

認識器では, ボールの位置, 各エージェントの位置など, エージェントの置かれている環境を認識として認識器から取得する. 取得した状態を意思決定部へ送信する.

・意思決定部

意志決定部では, 認識器から取得した環境の状態に基づき, 環境の状態に適した行動を選択する. 意思決定部では, クラシファイアシステムを用いて行動を決定している.

・行動器

行動器では, 意志決定部で選択された行動（パスやドリブルなど）を実行する.

2.3. 従来の意思決定アルゴリズムと問題点

従来, サッカービデオゲームのエージェントの行動はビデオゲーム開発者の決めたアルゴリズムに従って選択される. アルゴリズムは IF-THEN 形式で書かれている. 図 2 に IF-THEN 形式で書かれたルールの適用例を示す. 図 2 の例では,「もし, 自分の右にボールがあり, かつ自分の前方にゴールがあり, かつ敵チームのプレイヤーがゴールと自分の間にいたならば, ボールをパスする.」というルールを記述している. アルゴリズムが IF-THEN 形式で書かれる利点として, 記述のしやすさがある. また, 新しいルールを追加する場合, 今までのルールが理解しやすく, すぐに新しいルールを追加できる.

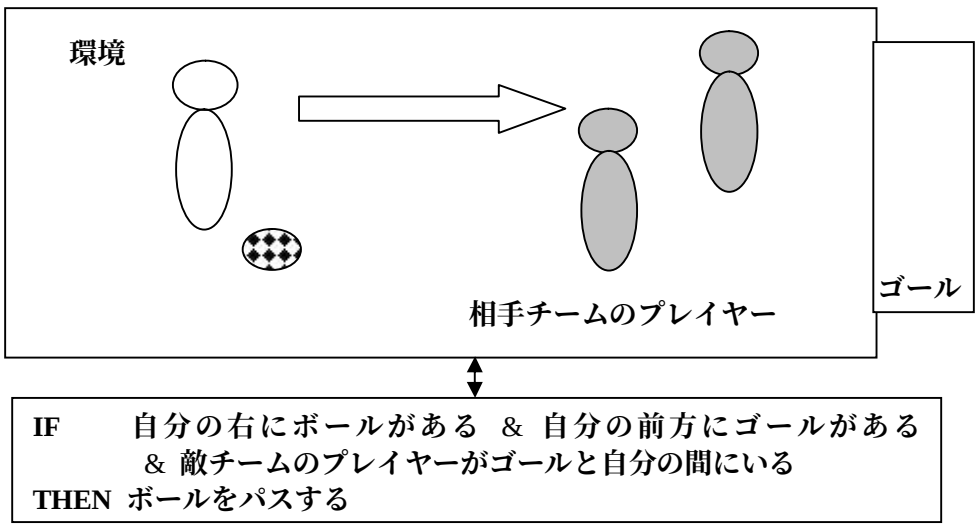

図 2 IF-THEN 形式で書かれたルールの例

一方で，多くの人がゲームに参加し，ゲーム環境が複雑化してきたことで，問題が出てきた．一つ目は，多くの人がゲームに参加しているため，それぞれに合わせたエージェントの戦略が必要となったことである．2 つ目に，熟練したプレイヤーに対してより難しいアルゴリズムを提供する必要が出てきたことである．3 つ目に，インターネットの広がりによって新しいプレイヤーが参加しやすくなったことで，新しいアルゴリズムを早いサイクルで提供，保守しなくてはならなくなったことである．以上のような理由から，アルゴリズムが短命化しており，従来の意思決定手法では複雑になっている環境に対して動的にアルゴリズムを変えることは難しいという問題がある．

3. 過去の研究

3.1. ハイブリッドシステム

上記の問題を解決するために，我々は機械学習を用いたアルゴリズム，ハイブリッド型意思決定システムを提案した．我々の提案は，機械学習を組み込むことで，ゲームプレイヤーの戦術を学習し，より良い戦略を自動的に獲得していくことを目的としたものである．提案によって，プログラムのバグを減らし，開発，メンテナンスに掛かる時間を減らすことができる．　実装する機械学習として，Q-学習[6]，遺伝的アルゴリズム，ニューラルネットワークなどが考えられるが，提案に用いる機械学習としてクラシファイアシステムを採用した．採用した理由は，RoboCup[7,8]のようなサッカーゲームにおいてロボットの意思決定に進化的計算を適用した例[9-11]が多くあり，その多く例を考慮し，従来のアルゴリズムによる意思決定システムと整合性がよいと考えたためである．クラシファイアシステムでは，アルゴリズムによる意思決定システムと同様に IF-THEN 形式でルールが書かれている．よって，アルゴリズムによる意思決定システムに書かれているルールを取り込みやすく，従来のクラシファイアシステムを少し変更するだけでハイブリッド型意思決定システムを作ることができる．また，行動の流れを学習するために，クラシファイア間で報酬の受け渡しを行う，バケツリレーアルゴリズム[12-17]を導入した．図 3 にハイブリッド型意思決定システムの概要を示す．図 3 では，エージェントは試合中に学習獲得する学習クラシファイアと，ゲーム製作者によって必要最低限の行動が記述された戦略クラシファイアの 2 種類を保持している．戦略クラシファイアは更新されず，優先的に実行される．

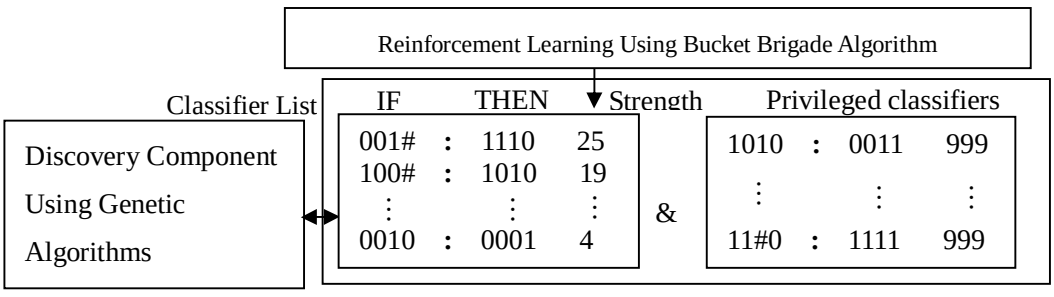

図 3 ハイブリッド意思決定システム(文献 2 より引用)

一方，クラシファイアの学習には多くの時間がかかることが知られている．そのため，ハイブリッド型意思決定システムでは 3 つの改善によって学習性能を強化している．1 つ目はイベント分析機能の導入である．認識した回数の多いイベントに対応するクラシファイアを学習することで環境への適応率を高めることができる．2 つめは，ルール生成において，遺伝的アルゴリズムの適用を行動部だけとしたことである．状況部の生成はイベント分析機能によって分析されたイベントを追加し，その状況に応じて行動部を生成する．3 つ目は，バケツリレーアルゴリズムによって出現頻度の高いイベントのクラシファイアの信頼度を更新する．図 4 に上記の改善を加えたイベント駆動型ハイブリッド意思決定システムの概要を示す．

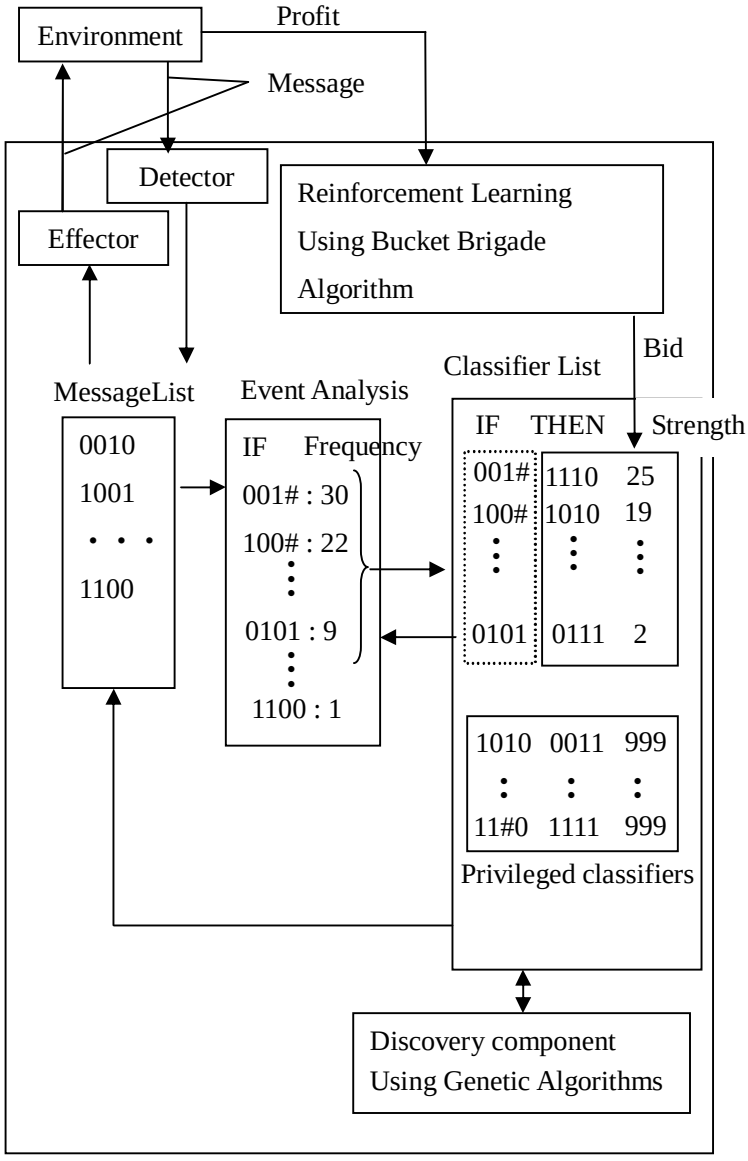

図 4 イベント駆動型ハイブリッド型意思決定システム(文献 2 より引用)

3.1.1. クラシファイアシステム

ハイブリッド型意思決定システムに用いているクラシファイアシステムとは, ルールベース型意思決定システムの一種である. クラシファイアシステムは, クラシファイアと呼ばれる IF（条件）—THEN（帰結部）形式で書かれたルールを持つ. このルールは, IF 部が一致するならば, THEN 部を実行せよというルールである. IF 部は 0, 1, ＃, THEN 部は 0, 1 を用いて記述される. 図 5 にクラシファイアシステムの概要を示す.

クラシファイアシステムは次の 3 つの部分から構成される.

・実行部

実行部では, 環境から入力された情報と適合するクラシファイア（IF 部）を探し, 適合したクラシファイアの THEN 部をメッセージリストに加える. メッセージリストに加えられた THEN 部は効果器（effecter）によって, 環境に対して実行される

・強化部

強化部は, クラシファイアの持っている信頼度という値を適切な値に調整する役割を持つ.

一般的に環境から入力された情報と適合するクラシファイアは複数ある. この場合, 適合したクラシファイアの中から実行するクラシファイアを決定する際に, 信頼度の値をもとに実行するクラシファイアを決定する. なお, 強化部では信頼度の調整にバケツリレーアルゴリズムが用いられている.

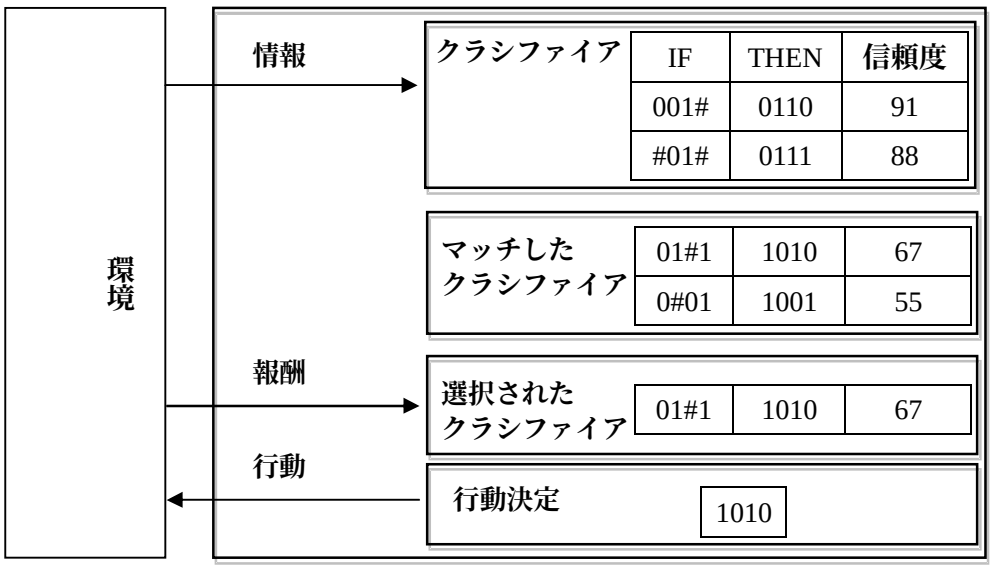

図 5 クラシファイアシステムの概要

・発見部

発見部では新しいクラシファイアを，遺伝的アルゴリズムを用いて発見し，古いクラシファイアの一部と交換する．

3.1.2. バケツリレーアルゴリズム

クラシファイアシステムでは，クラシファイアの信頼度の値の調整に，クラシファイア間で信頼度の受け渡しを行うバケツリレーアルゴリズムが用いられている．

クラシファイアの信頼度の調整を行うには，「成功したクラシファイアの信頼度を増やす」，「失敗したクラシファイアの信頼度を減らす」，ということが考えられる．一方で，信頼度の値が増減するのは，環境に作用し，成功，失敗が判定されたクラシファイアのみである．そのため，成否が判定されたクラシファイアに到るまでに実行されてきた別のクラシファイアでは信頼度が変化せず，一部のクラシファイアのみが強化されることになる．図 6 に例を示す．図 6 ではクラシファイア A から C まで順番に実行され，クラシファイア C で行動の成否が判定される．信頼度が増減するのはクラシファイア C のみであり，クラシファイア A，B の信頼度は変化しない．よって，クラシファイア C のみが強化され，クラシファイア C ばかりが選択されるようになる．また，クラシファイア A，B，C という一連の流れを学習することができない．

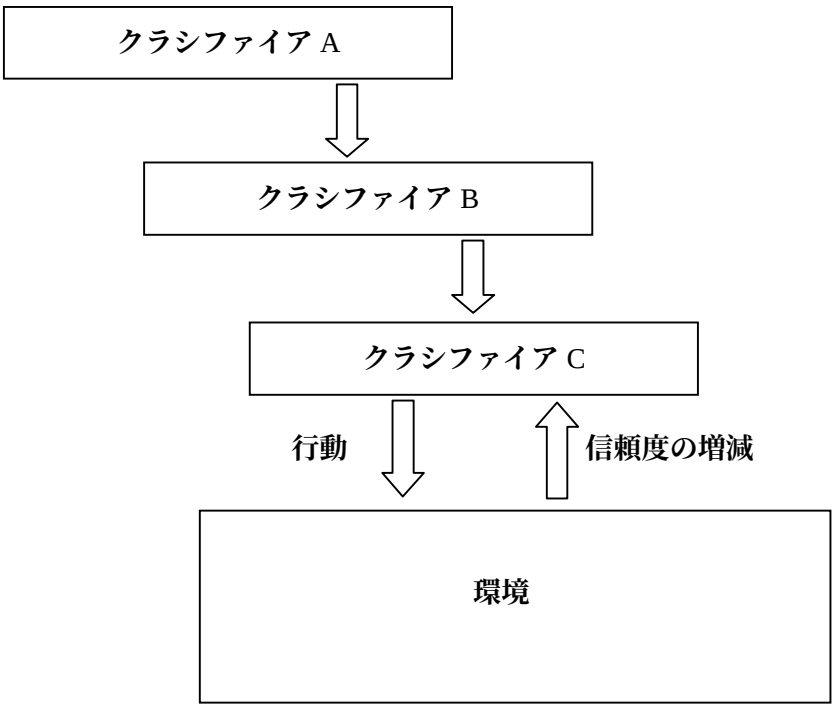

図 6 バケツリレーアルゴリズムが無い場合の実行例

上記の問題を解決するために, クラシファイア間で信頼度の受け渡しを行うバケツリレーアルゴリズムが考えられた. バケツリレーアルゴリズムはクラシファイア間で信頼度の受け渡しを行う. 図 7 に概要を示す. 図 7 ではクラシファイア A, B, C, D の順番で実行され, クラシファイア D で行動の成否が判定され, クラシファイア D の信頼度が増減する. このとき, それぞれのクラシファイアは, 一つ前に実行されたクラシファイアに信頼度の一部を分け与える. 各クラシファイアは次に実行されたクラシファイアから信頼度の一部を受取ることで信頼度を強化する.

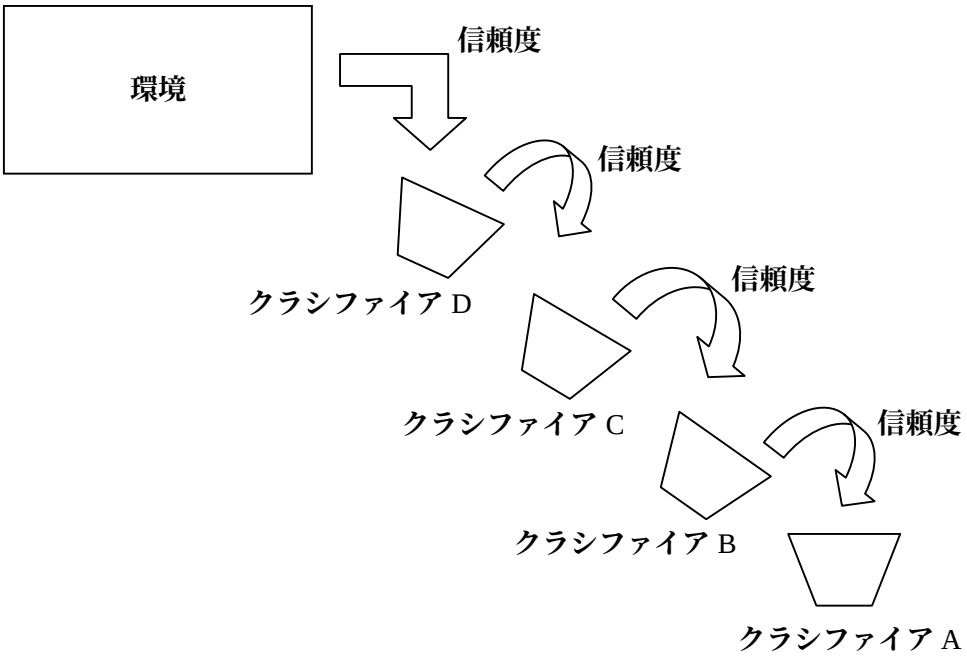

図 7 バケツリレーアルゴリズムの概要

3.1.3. クラシファイアシステムの問題点

クラシファイアシステムを用いた学習では，学習に環境からの報酬に頼って行っている．そのため，報酬の値が学習に適切な値に設定されていない場合，学習が効率よく行えず，多くの時間がかかる．一方で，報酬の値の設定には決定指標が無く，設計者の試行錯誤によって，エージェントの役割，学習の目的，などに合わせて設定する必要があるという問題がある．

3.2. 役割分担を考慮した報酬値設定

報酬値を試行錯誤によって設定しなくてはならないという問題の解決案の一つとして，ゲーム内のエージェントの役割を考慮した報酬値設定を行うことを提案した．

クラシファイアシステムをもちいたハイブリッド意思決定システムでは，報酬値を用いた強化学習によりシステムの持つアルゴリズムを変化させることで動的に変化する環境に対応することを可能にした．一方，強化学習に用いる報酬値は試行錯誤によって設定され，適切な値に設定されていない場合，学習効率を下げ，学習時間が大幅に増えることになる．適切な報酬値を設定し学習効率を高めるために，エージェントの役割を考慮した報酬値設定を行うことを提案した．

3.3. GA による報酬値設定

前述の改善により，エージェントの学習能力の向上を図れた．一方で，学習に用いる報酬値の設定をどのように行えばよいかという問題が残った．報酬値の設定には指標がなく，経験に基づく試行錯誤によって設定しなくてはならないという問題である．報酬値がエージェントの役割など，複数の要素を考慮して設定しなくてはならず，適切な設定でない場合エージェントの学習が効率的に行えない可能性がある．

我々は問題を解決するために，具体的な教師データがなくても，とりあえず強化学習の行えるＧＡの特徴に着目し，報酬値の設定にＧＡを用いることを提案した．結果，効率的な学習の行える報酬値を見つけることができ，報酬値設定を自動化できる見通しを得ることができた．上記提案では探索空間の広さの問題から，まずはサッカーゲームにおけるポジション，FW，MF，DF それぞれのパスの行動とドリブルの行動に与えられる成功報酬値である，各ポジションの PASS，DRIBBLE の 6 種類の報酬値について探索を行った．一方で，新たに探索対象となる報酬値を加えて探索を行う場合，探索空間が広くなるため探索効率が下がり，実用的な時間内で効率的な探索を行うことが困難となる問題があった．

上記問題を解決するために，集団を複数の集団に分割して解を探索する分散遺伝的アルゴリズム（DGA）から着想を得た，報酬値探索の方法を提案する．DGA については次項に記述する．

4. 分散遺伝的アルゴリズムとは

分散遺伝的アルゴリズム（DGA）[18]とは，集団全体を複数のサブ集団に分割して，それぞれのサブ集団に対して遺伝的アルゴリズムのオペレーションを実行する GA である. 各サブ集団のことを島とよび，島モデル GA とも言われる.

分散遺伝的アルゴリズムでは, 各サブ集団でそれぞれ独立に遺伝的アルゴリズムのオペレーションを実行し，一定世代ごとにサブ集団の間で，移住と呼ばれる情報交換を行う．移住の操作によって多様性を維持している. 移住の操作では，サブ集団からある割合の個体が別の集団に送られ，別のサブ集団から個体と同数の個体を受け取り，現在のサブ集団と置き換える操作をおこなう．図 8 に分散遺伝的アルゴリズムの基本概念の図を示す.

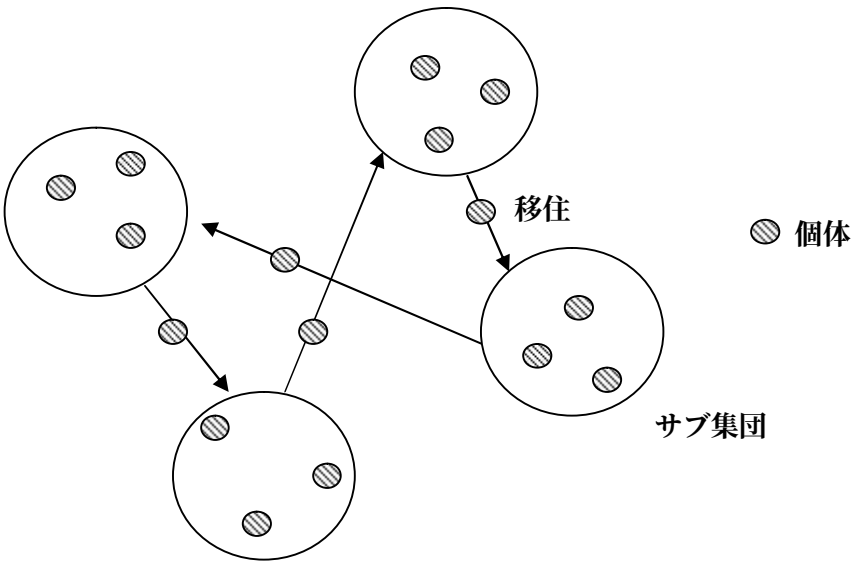

図 8 分散遺伝的アルゴリズムの基本概念

5. 探索空間を分割した報酬値探索の提案

5.1. 提案の概要

探索空間が広がるため探索効率が下がり，実用的な時間内で探索を行うことが困難になる問題の対策として，上記の分散 GA から着想を得て，探索を行う報酬値を 2 つのグループに分割し，分割した報酬値グループを交互に，GA を用いて探索することを提案する．分割の理由は，報酬値間の依存関係に着目したグループ分けを行う事で，探索空間を小さくし，効率的に探索を進めることができると考えたためである．また，分割した報酬値グループを交互に探索することで，それぞれのグループが相互に学習に影響を与えながら，報酬値全体でもエージェントの学習効率を向上させる方向に改善されると考えられる．

提案では複数の報酬値を，2 つのグループに分け GA によって探索する．GA による探索では，数世代の間，一方のグループを探索し（以下，探索部），もう一方のグループは探索をせず値を固定する（以下，固定部）．探索部では GA によって報酬値の探索を行い，固定部では，値をそれまでの探索によって設定された値で固定する．数世代探索を進めた後，探索部と固定部を切替えて，報酬値の探索を行う．図 9 に探索の例を示す．図 9 では探索を行う報酬値の範囲を α と β の 2 つの範囲に分け，α 部を探索部，β 部を固定部として探索をおこなう．その後，α 部を固定部，β 部を探索部に切替えて，β 部の探索を行なう．これを繰り返す．

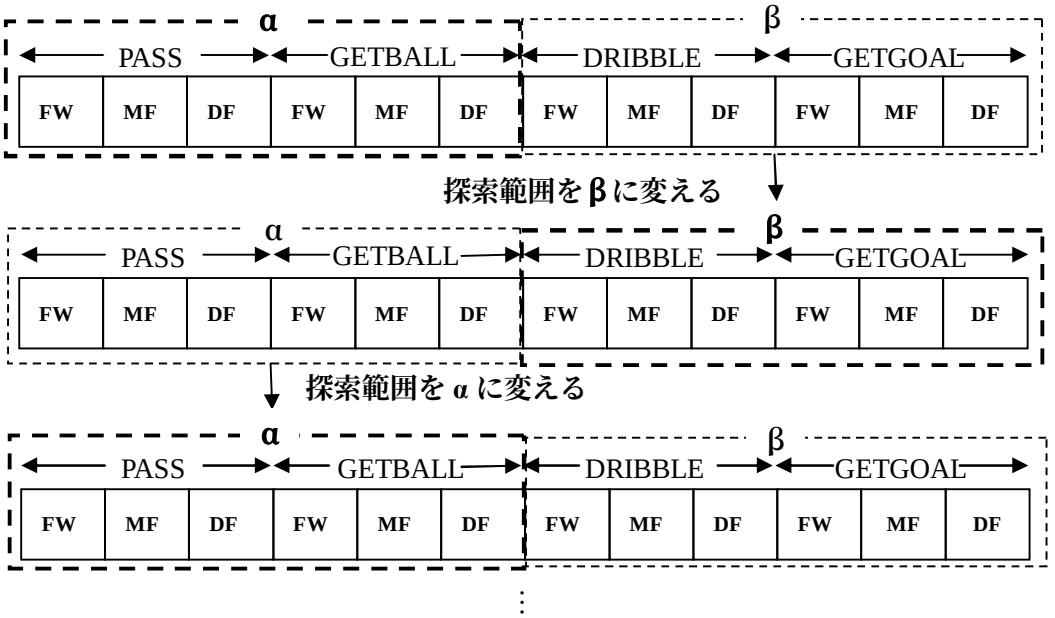

図 9 提案手法による探索の例

5.2. 報酬値の探索を行う GA の設計

本稿における染色体の定義を図 10 に示す．一番左がパスの行動を成功させたときに受け取る報酬値，PASS である．次がボールを奪ったときに受け取る報酬値，GETBALL である．次はドリブルが成功した時に受け取る報酬値，DRIBBLE である．最後がゴールを奪ったときに受取る報酬値，GETGOAL である．以上の報酬値から定義される染色体を個体とする．報酬値は全て整数値で与える．表 1 に我々の扱うサッカーゲームで用いる報酬値を示す．図 11 に勝率を基にした強化学習の構成を示す．次に，GA による報酬値探索の流れを説明する．まず，GA による探索を，どの報酬値グループに対して適応するかを選択する．本稿では PASS &GETBALL または DRIBBLE&GETGOAL のどちらのグループに適応するかを選ぶ．探索範囲に選択されなかった報酬値グループでは，報酬値の値を固定する．そのため探索範囲外の報酬値の値には，今までの探索により設定された値をそのまま用いる．一方，探索範囲に選択されたグループでは，個体の選択において，親個体を適応度に基づいてルーレット選択により選択する．適応度は，200 試合を 3 回行い，3 回の勝率の平均とする．次に選択された個体の間で，交叉率に基づいて交叉を行なう．交叉では，指定した探索範囲内でランダムに 0,1 を発生させマスクビットを作成し，0 なら親 1 の値を，1 なら親 2 の値を受け継ぐ．図 12 に交叉の例を示す．図 12 では，探索範囲を DRIBBLE と GETGOAL とし，マスクビットに基づいて MF の DRIBBLE の値と，FW の GETGOAL の値で交叉を行なっている．最後に突然変異を施す．突然変異は，突然変異率に基づいて，起こす報酬値を選択し，現在の値に対してランダムに 5 から 20 の値を加算，減算することで実現している．(1)に式を示す．

$$x_{n+1} = x_n \pm y \tag{1}$$

y は 5 から 20 のランダムな値

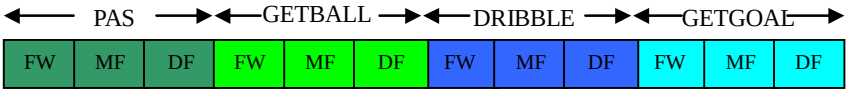

図 10 提案手法における染色体の定義

表 1 サッカーゲームで用いる報酬値

	FW	MF	DF
PASS	探索する報酬値		
DRIBBLE			
GETGOAL			
GETBALL			
LOSTBALL	-50		
GETGOALTEAM	20	LOSTGOALTEAM	-20
WINGAME	30	LOSTGAME	-30

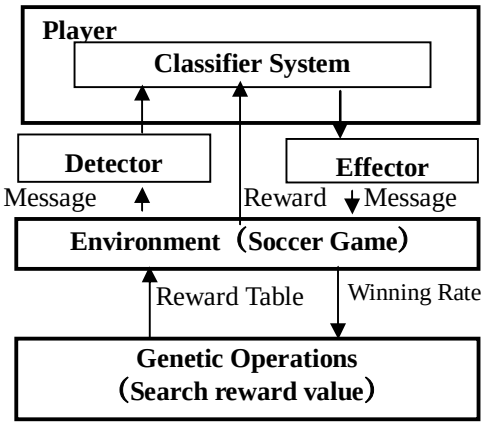

図 11 勝率を基にした強化学習の構成

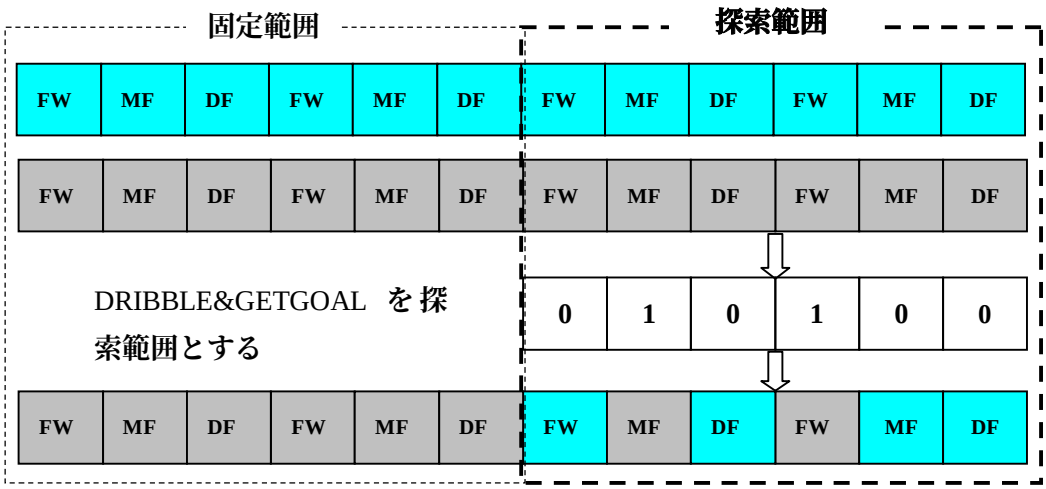

図 12 提案手法における交叉の例

5.2.1. 突然変異と交叉について

実験において，報酬値を設定する GA では，過去の実験結果から，突然変異率を 0.3 と高い値で設定している．一般的な突然変異率では，用いている個体の数が少ないため，探索を進めていくうちに各個体が同じ値に収束し個体が多様性を保てない可能性があるため高い値に設定している．また，交叉は，実験にかかる時間を考慮し過去の実験結果から，効率のよかった一様交叉を用いる．

6. 実験

6.1. 実験方法

実験では，サッカービデオゲームを用いて，2 つのチームを対戦させる．HOME チームにはイベント駆動型ハイブリッド意思決定システムを用いる．AWAY チームには，決まったアルゴリズムに沿って行動する，アルゴリズム型意思決定システムを用いる．探索によって設定される報酬値は HOME チームに適用する．探索対象となる報酬値は FW，MF，DF それぞれの PASS，DRIBBLE，GETBALL，GETGOAL の 12 種類の報酬値とした．この報酬値を PASS & GETBALL と DRIBBLE & GETGOAL の２つのグループに分け，探索を行う．図 13 に実験の流れを示す．

Phase 1 : 12 種類の報酬値を染色体とする個体を 4 つ用意し, 初期値をランダムに設定する．

Phase 2 : 各個体を用いて 200 試合を 3 回行い，3 回の平均勝率を個体の適応度とし，探索範囲を切替えながら 30 世代まで探索を行う．

Phase 3 : 初期世代，10 世代，20 世代，30 世代において 200 試合×3 回の平均勝率が最も良かった個体を用いて，200 試合を 10 回行い，平均勝率を出力する．

Phase 4 : Phase1 から 3 の実験を 3 回行い，その 3 回の平均データを結果として用いる．

予備実験の結果から,5 世代ごとに探索範囲を切替えて探索を進める.交叉率は 0.6 とし,突然変異率は 0.3 とした．実験結果に用いるデータは，探索した報酬値を用いて学習する，イベント駆動型ハイブリッド意思決定システムを適用した HOME チームのデータを用いる.勝率の計算は（2）で示す計算式によって出力する.

$$R_w = N_w /(N_t - N_d) \tag{2}$$

ここで N_t は総試合数である．N_d は引き分け数，N_w は勝利数を表す.

6.1.1. 対戦アルゴリズム

AWAY チームに用いるアルゴリズムとして以下の 3 つを用意した．1 つは攻撃と守備のバランスを取ったアルゴリズムである，アルゴリズム A．2 つ目は，攻撃に重点を置いたアルゴリズムである，アルゴリズム B．3 つ目は守備に重点を置いたアルゴリズムである，アルゴリズム C. 以上の 3 つを用意した．種類の違うアルゴリズムを用いることで多様性のある環境を実現する.

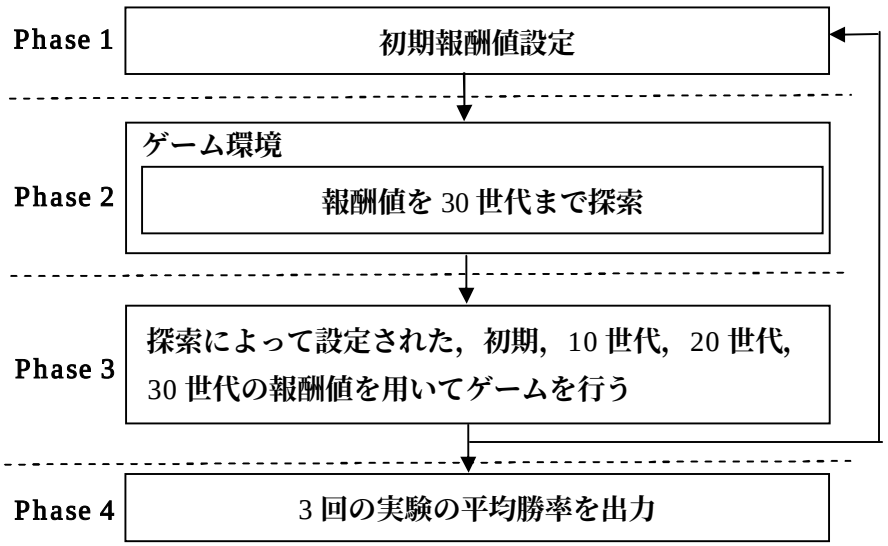

図 13 実験の流れ

6.2. 実験結果

6.2.1. 対アルゴリズム A

図 14 に攻守のバランスを取ったアルゴリズム, アルゴリズム A と対戦させた結果を示す. 初期から 30 世代に探索が進むにつれて勝率が上昇している．また，勝率の立ち上がりも探索が進むにつれ早くなっている．初期の報酬値を用いた場合，50 試合程度で 60%の勝率に達しており，その後は上昇していない．一方，10 世代での報酬値を用いた場合，初期より立ち上がりは早く，20 試合程度で 60%を超える勝率を示している．20 世代での報酬値を用いた場合，30 試合程度で 65%程度の勝率を示しており，最終的な勝率は 70%弱を示している．30 世代の報酬値を用いた場合，勝率の立ち上がりをみると 20 世代よりも早く，30 試合程度で 70%程度に達しており，70 試合程度で勝率は 70%を超える値を示している．

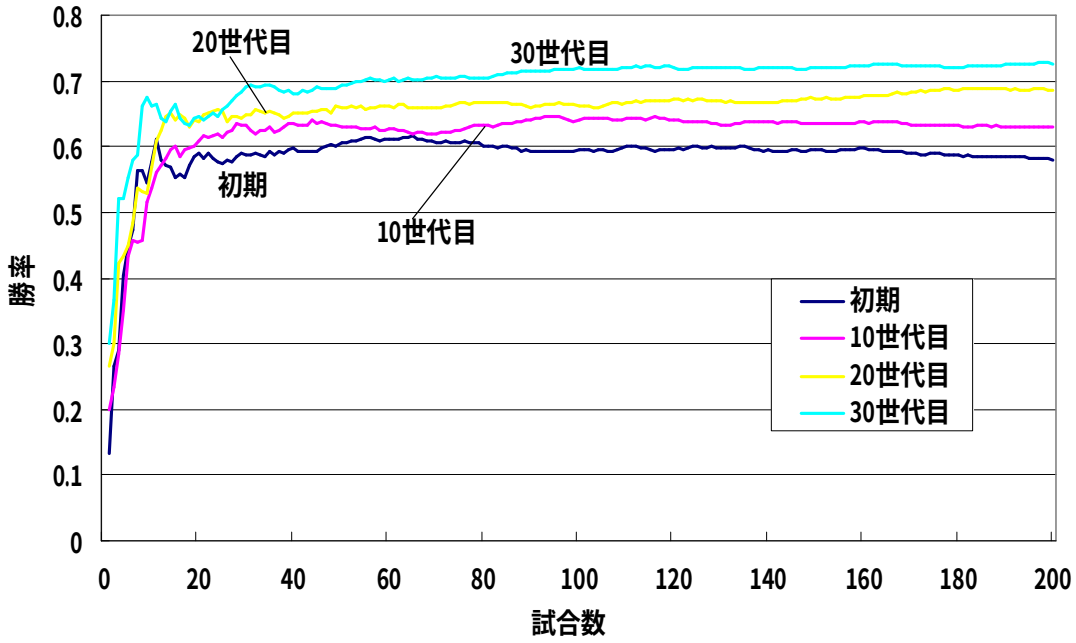

図 14 提案手法を用いて報酬値を探索した場合
（対アルゴリズム A）

6.2.2. 対アルゴリズム B

図 15 に攻撃に重点を置いたアルゴリズム，アルゴリズム B と対戦させた結果を示す．10 世代から 30 世代に探索が進むにつれて勝率が上昇している．勝率の立ち上がりでは世代が進むにつれて早くなっており，20 世代，30 世代では 20 試合までには 60%を超えている．初期の報酬値を用いた場合，勝率は大きな上昇も無く，最終的な勝率は 55%程度を示している．10 世代の報酬値を用いた場合，初期と同様に大きな変化は無く，最終的な勝率も初期とほぼ変わらず，60%弱を示している．一方，20 世代目の報酬値を用いた場合，勝率の立ち上がりが早く，20 試合から 30 試合で 60%を超えており，勝率は下がることなく，最終的な勝率は 65%程度を示している．30 世代での報酬値を用いた場合，20 世代と同等に勝率の立ち上がりは早く 20 試合程度で 65%程度の勝率を示し，勝率は下がることなく，最終的な勝率は 70%程度を示している．

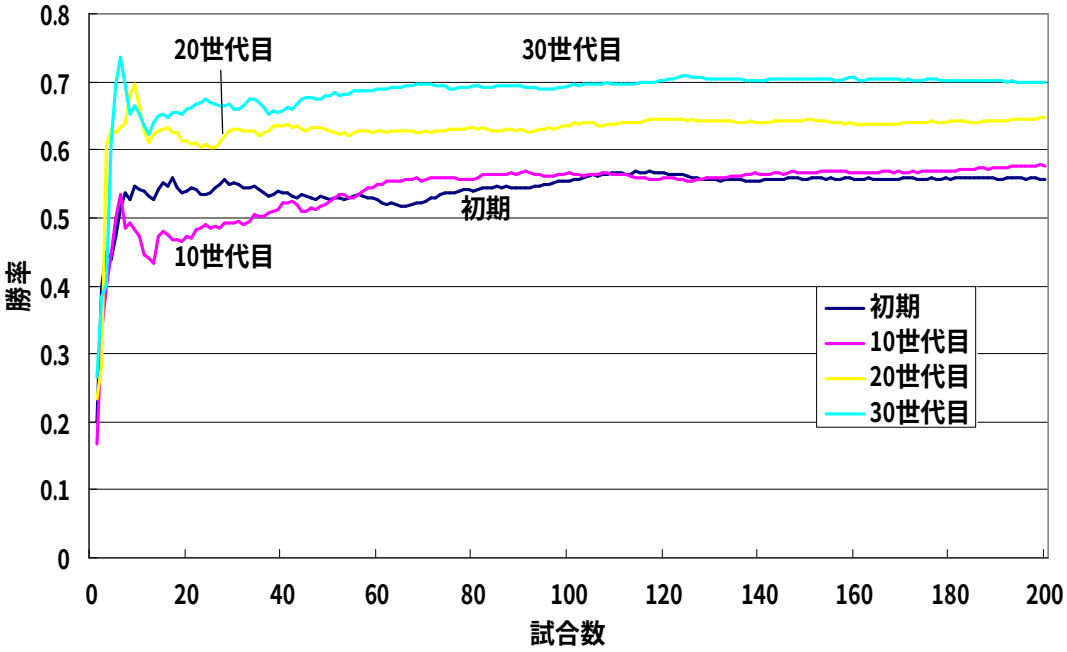

図 15 提案手法を用いて報酬値を探索した場合
（対アルゴリズム B）

6.2.3. 対アルゴリズム C

図 16 に守備に重点を置いたアルゴリズム，アルゴリズム C と対戦させた結果を示す．初期の報酬値を用いた場合，20 試合程度から徐々に下がっていき，勝率は 50 試合程度で 60%を下回り，勝率の上昇は見られず，最終的な勝率は 60%を超えることは無かった．10 世代目の報酬値を用いた場合，勝率は 30 試合程度で 60%程度に達し，80 試合程度で 70%弱の勝率になっている．20 世代目の報酬値を用いた場合，最終的な勝率は 70%弱の勝率を示しているが，20 世代目の立ち上がりを見てみると，10 世代目よりも早く，20 試合程度で 10%程度の差があり，20 世代目の勝率は 70%程度に達している．30 世代目の報酬値を用いた場合，立ち上がりは 20 世代目に劣るものの，60 試合程度で 70%程度の勝率に達し，最終的な勝率は 70%を超えている．

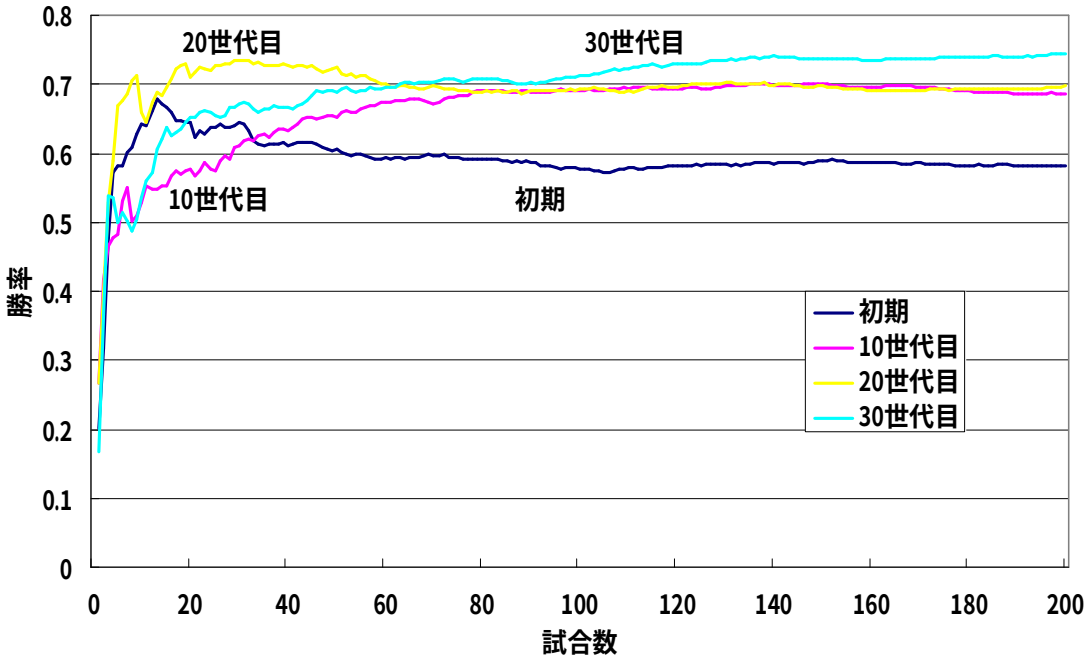

図 16 提案手法を用いて報酬値を探索した場合
（対アルゴリズム C）

6.2.4. 従来手法による探索を行った報酬値を用いた場合の勝率

図 17, 図 18, 図 19 に探索対象を一括で探索する従来手法によって探索を行った報酬値を用いた場合の対戦結果をしめす．対アルゴリズム A において，初期から探索を進めるにつれて，勝率はわずかではあるが上昇した．10 世代目，20 世代目，30 世代目の勝率は 60%を超える程度であり，勝率は 70%を超えることは無かった．勝率の立ち上がりを見ると，世代が進んでも提案手法を用いた時ほど早くなることは無く,ほとんど変わらない結果となった．

アルゴリズム B では，10 世代目の勝率は初期より低くなってしまったものの，20 世代目では 60%程度の勝率を示した．一方，30 世代目の報酬値を用いた場合でも，70%近くまで勝率は上昇したが，70%を超えることは無かった．勝率の立ち上がりを見ても初期，20 世代目，30 世代目の間に大きな差がなかった

アルゴリズム C では，10, 20, 30 世代目の勝率は初期よりわずかに高い値を示している．一方，10 世代目以降は探索を進めても勝率はほとんど上昇せず，10 世代目，20 世代目，30 世代目の勝率は，初期の勝率よりは高くても 60%程度であった．また，勝率の立ち上がりを見ても 10 世代目，20 世代目，30 世代目の間には大きな差は出ていない．

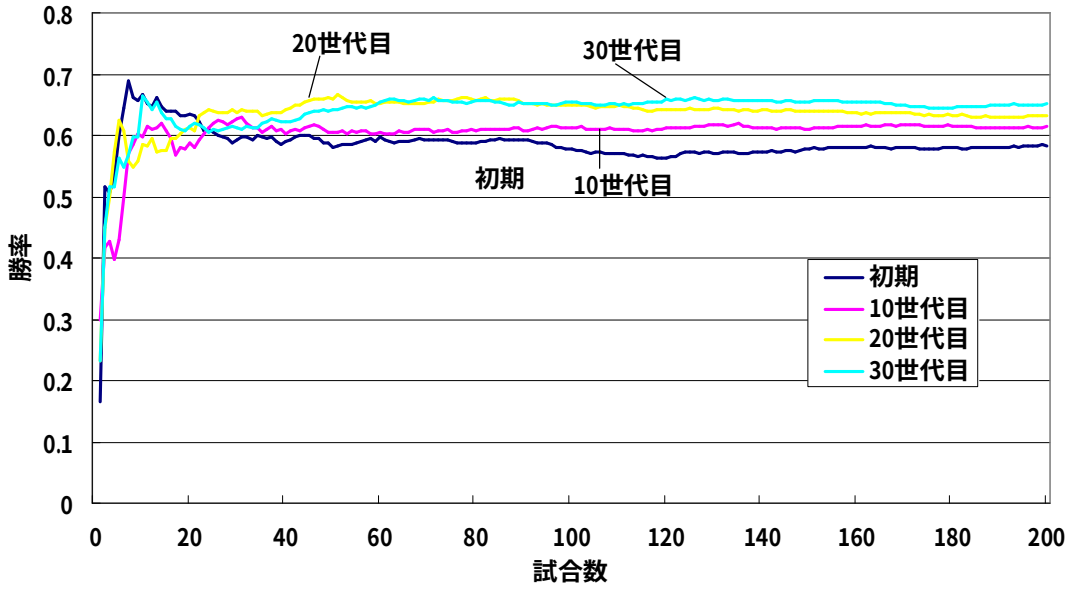

図 17 一括して探索を行う従来手法を用いて報酬値を探索した場合
（対アルゴリズム A）

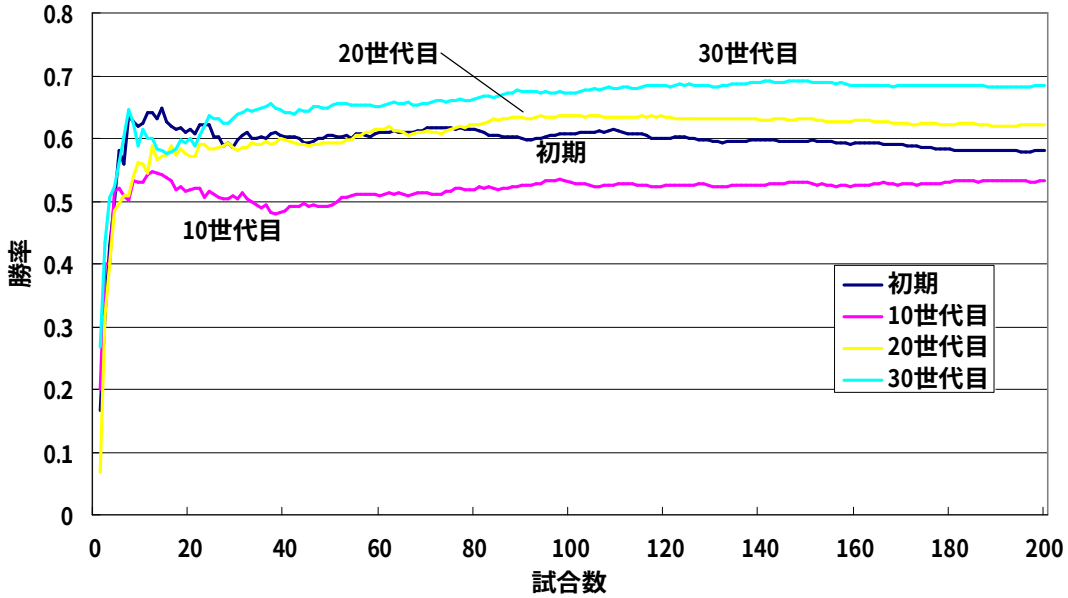

図 18 一括して探索を行う従来手法を用いて報酬値を探索した場合
（対アルゴリズム B）

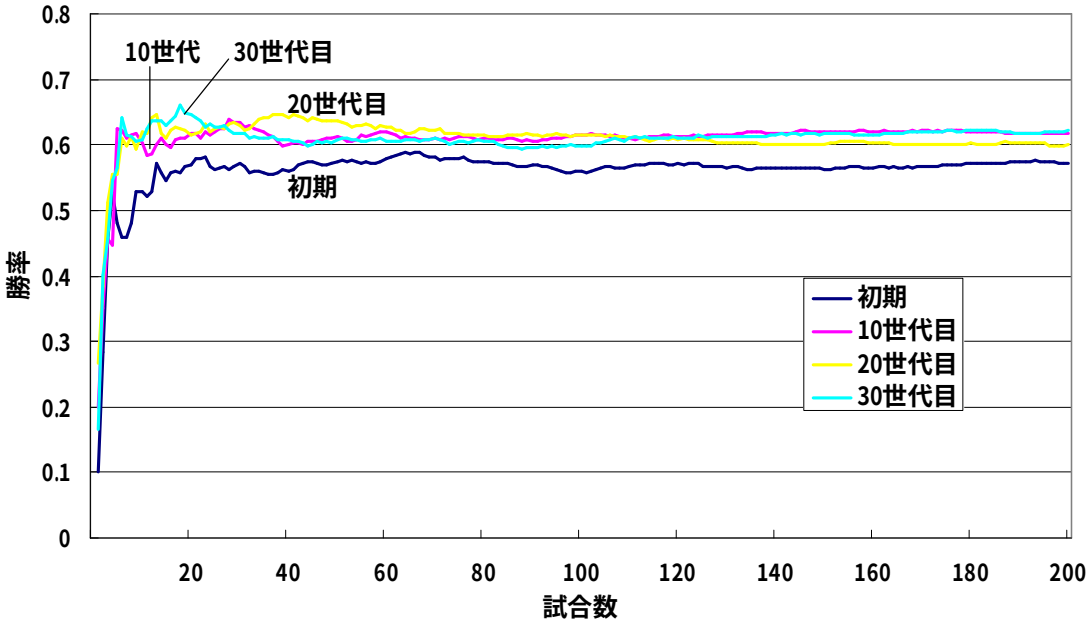

図 19 一括して探索を行う従来手法を用いて報酬値を探索した場合
（対アルゴリズム C）

6.2.5. 提案手法において探索範囲の切り替えを 10 世代ごとにした場合

図 20-22 に探索範囲の切り替えを，5 世代ごとから 10 世代ごとに変えた場合の勝率を示す．図 20 より，アルゴリズム A と対戦させた場合，10 世代から 30 世代にかけての探索により，わずかながら勝率が上昇しているが，60%をわずかに超える程度であった．勝率の立ち上がりを見ても，20 世代目，30 世代目の間に大きな差は出ていない．

図 21 より，アルゴリズム B と対戦させた場合，10 世代目の勝率より，20 世代目，30 世代目の勝率が上回っているが，30 世代目の勝率は 60%程度であった．勝率の立ち上がりでは，20 世代目の立ち上がりが早いが，その後試合を重ねると下がっていき，60%程度の勝率に下がっている．

図 22 より，アルゴリズム C と対戦させた場合，勝率は 60%程度で，60%を大きく超えることは無かった．勝率の立ち上がりを見ても大きな差は出ていない．また，10 世代目の勝率が，20 世代目の勝率を上回っている一方，30 世代目の勝率は 20 世代目よりも高い勝率を示しており，安定しない勝率となっている．

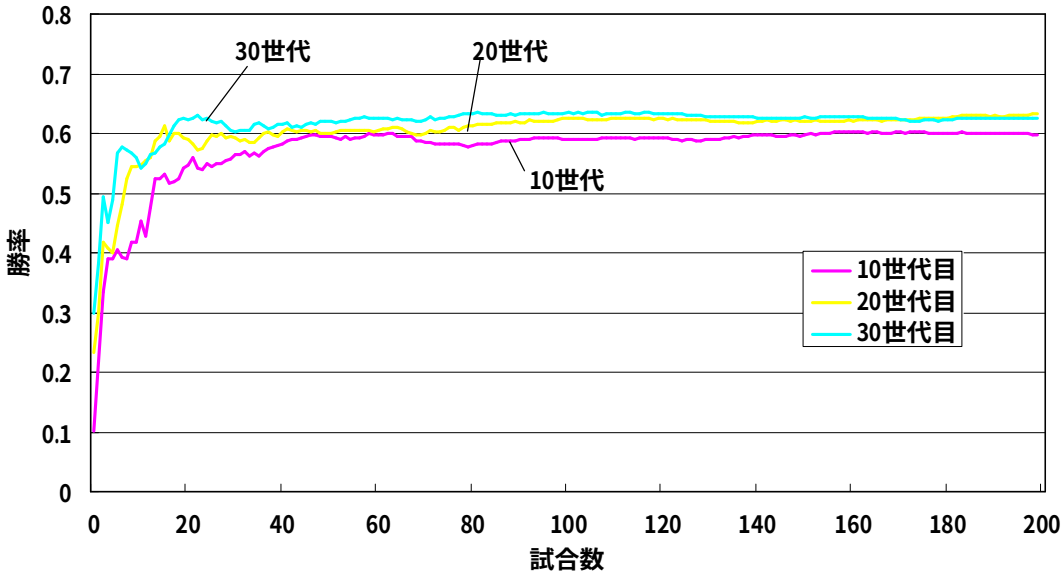

図 20 切替える世代数を 10 回ごとに変えた場合
（対　アルゴリズム A）

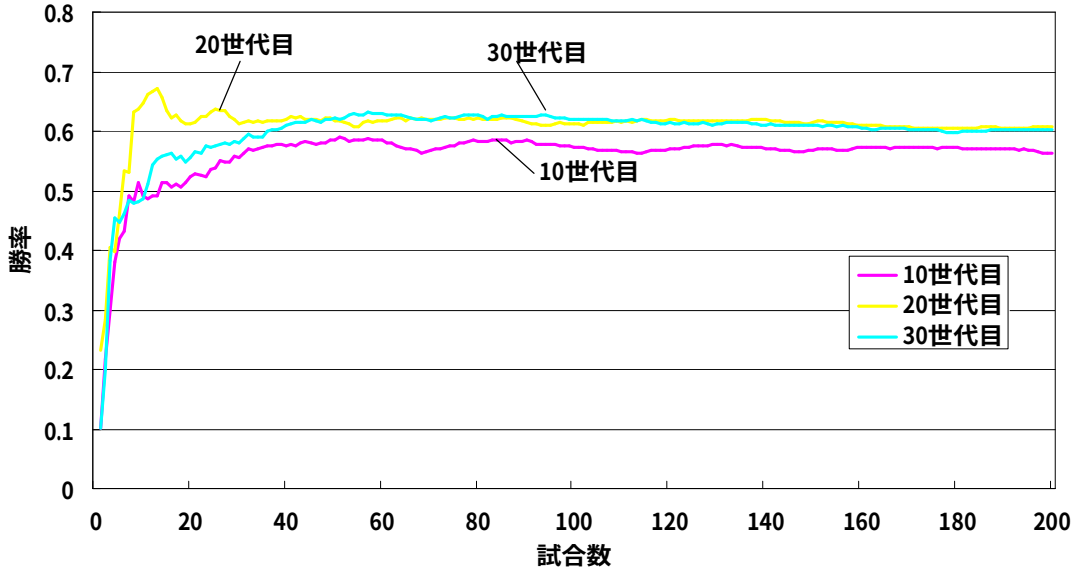

図 21 切替える世代数を 10 回ごとに変えた場合
（対　アルゴリズム B）

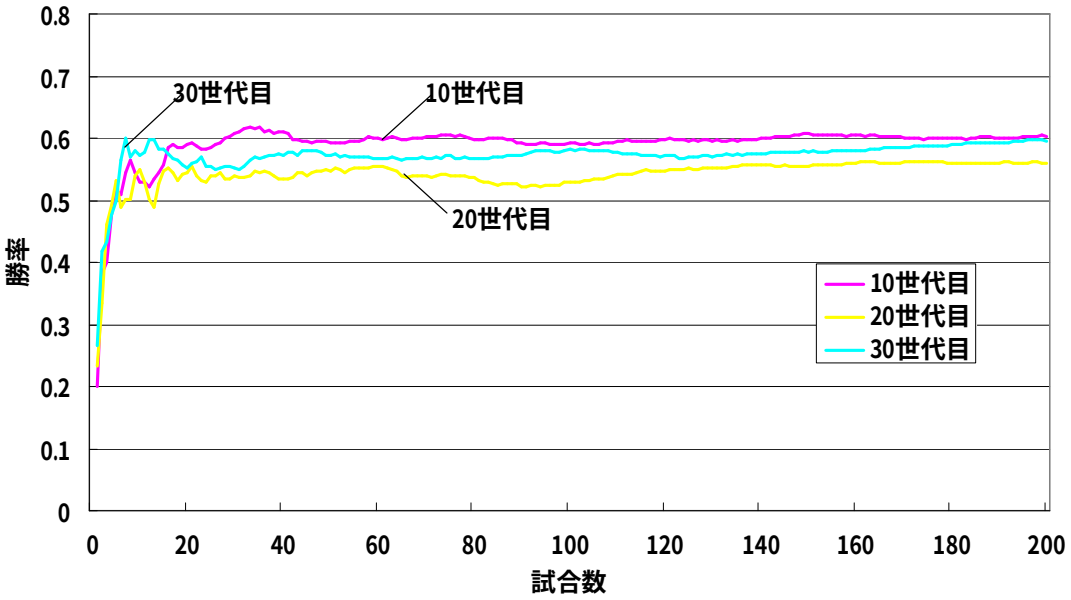

図 22 切替える世代数を 10 回ごとに変えた場合
（対　アルゴリズム C）

6.2.6. 探索の世代数を進めた場合

探索範囲の切り替えを 10 世代ごとに変えた場合，探索を進めても勝率の上昇があまり見られなかった．そこで，アルゴリズム C との対戦に用いた報酬値の探索をさらに進める実験を行った．図 23 に報酬値の探索をさらに進めた場合の勝率を示す．図 23 より，30 世代から 40 世代にかけての探索により勝率が上昇した．40 世代目の勝率は，20 試合程度で 60%を超えており，10 世代目，20 世代目，30 世代目の勝率より高くなっている．また立ち上がりが 10 世代目，20 世代目，30 世代目より早い．80 試合程度で 70%程度の勝率に達している．また，40 世代目の勝率は下がることは無く安定している．最終的な勝率は 70%程度を示している．

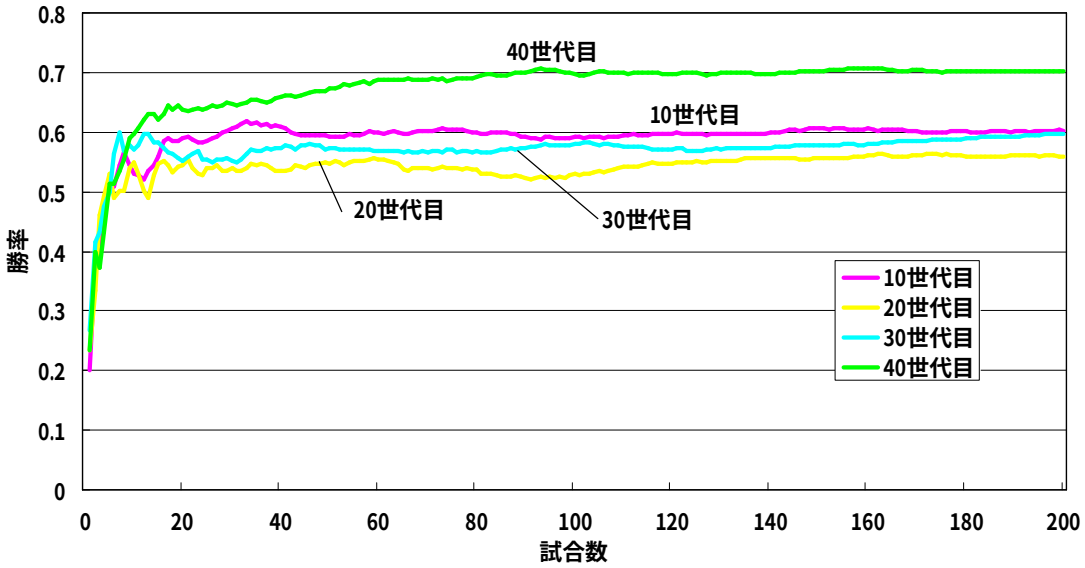

図 23 切替える世代数を 10 回ごとに変えた場合
（対　アルゴリズム C）

7. 考察

図 14,15,16 に示した，イベント駆動型ハイブリッド意思決定システムとアルゴリズム A，アルゴリズム B，アルゴリズム C との対戦結果より，提案手法を用いて探索した報酬値をエージェントのプレーの学習にもちいた場合，報酬値の探索が進むにつれて，勝率は上昇し，勝率の立ち上がりも早くなっている．また，一括で探索した場合の報酬値を学習に用いた場合と比較すると，提案手法を用いた場合は，一括で探索した場合よりも短い探索時間で，一括で探索した場合と同じ，もしくは一括で探索した場合よりも高い勝率を示している．

提案手法を用いて探索した報酬値を用いた場合において，アルゴリズム A と対戦した結果，図 14 より，探索を進めるにつれて勝率が上昇していることから，報酬値の改善ができていると考えられる．また勝率は下がることなく安定しており，最終的に 70%を超える勝率を示していることから安定した学習ができていると考えられる．

図 24 に示した，報酬値の探索に提案手法と一括して探索する手法を用いた場合の勝率の比較の図，及び図 17 に示した一括して探索を行う手法で探索した報酬値を用いた場合の結果と，提案手法を用いて探索した結果（図 14）との比較より，10 世代目の報酬値を用いた場合においては，提案手法によって探索した場合の勝率がわずかに高い．20 世代目の報酬値を用いた場合においては，提案手法によって探索した場合の勝率が，一括で探索した場合の 30 世代目の勝率を超えている．これは，一括で探索した場合に 30 世代かけて探索した報酬値と同程度の学習効率である報酬値を，提案手法を用いることにより 20 世代で設定できたと考えられる．つまり，一括で探索した場合よりも短い探索時間で報酬値を設定できている考えられる

提案手法を用いて探索した報酬値を用いた場合において，アルゴリズム B と対戦した結果，図 15 より，20 世代目，30 世代目の報酬値を用いた場合，勝率の立ち上がりが 10 世代目と比較して早くなっている．また 30 世代目の報酬値を用いた場合，20 試合から 30 試合で 20 世代目よりも高い勝率を示し始めていることから，探索を進めることによって報酬値の改善ができていると考えられる．

図 25 に示した，報酬値の探索に提案手法と一括して探索する手法を用いた場合の勝率の比較の図，及び図 18 に示した一括して探索を行う手法で探索した報酬値を用いた場合の結果と，提案手法を用いて探索した結果（図 15）との比較より，10 世代目の報酬値を用いた場合では，提案手法を用いて探索した場合の勝率が，一括で探索した場合の勝率を大きく上回っており，提案手法を用いて探索した報酬値は，一括で探索する場合より効率的に学習の行える値になっていると考えられる．また，20，30 世代目の報酬値を用いた場合，立ち上がりは提案手法を用いて探索した場合が早いことから，同じ探索時間でも提案手法を用いた場合はより効率的な学習を行える報酬値が設定できていると考えられる．

提案手法を用いて探索した報酬値を用いた場合において，アルゴリズム C と対戦させた結果，図 16 より，30 世代目に比べ 20 世代目の勝率の立ち上がりが早いが，最終的な勝率

は 30 世代目が良いことから，30 世代目は安定した学習ができる報酬値に設定ができていると考えられる．

図 26 に示した，報酬値の探索に提案手法と一括して探索する手法を用いた場合の勝率の比較の図，及び図 19 に示した一括して探索を行う手法で探索した報酬値を用いた場合の結果と，提案手法を用いた場合との比較より，提案手法によって探索した報酬値を用いた場合，10 世代目の報酬値を用いた場合において，一括で探索した場合の 10，20，30 世代目の勝率を超えている．一方で，一括で探索した場合は 30 世代探索しても勝率が 60%程度のままであり，提案手法を用いた場合の 10 世代目の勝率に届かない．このことから，提案手法を用いた報酬値の探索により，一括で探索した場合よりも短い探索時間で効率的に学習を行える報酬値を設定できていると考えられる．

探索範囲の切り替えを 10 世代ごとに変えて報酬値の探索をおこなった場合，学習に報酬値を用いてアルゴリズム A，B，C と対戦させた結果の図 20-22 より，どのアルゴリズムでも勝率の上昇があまり見られない．どのアルゴリズムにおいても 60%程度の勝率を示すにとどまっており，一括で探索した場合と大きく変わらない．報酬値の探索を進めても勝率があまり上昇していないことから，対戦しているアルゴリズムにより適応できるように学習を行わせる報酬値が 30 世代までの探索では設定できていないと考えられる．これは探索範囲の切り替えが 5 世代ごとから 10 世代ごとに変わったため，情報交換の回数が減り，30 世代までの報酬値の探索では，5 世代ごとに探索を行った報酬値のように効率的学習のできる報酬値が見つかりにくくなったため考えられる．

探索範囲の切り替えを 10 世代ごとに変えた場合の実験結果をふまえ，30 世代からさらに 10 世代探索を進めた結果図 23 より，探索をさらに進めることで勝率の上昇が見られた．対戦しているアルゴリズムに適応できており，エージェントのプレーの学習が，効率的に行われていると考えられる．勝率が上昇した理由として 30 世代までの探索では情報交換の回数が減った分，探索をさらに進めることで情報交換の回数を増やし，学習に適切な報酬値の値を設定することができたためと考えられる．

表 2，表 3，表 4 に提案手法と一括して探索を行う従来手法それぞれで探索を行った，30 世代目の報酬値を用いてアルゴリズム A，B，C と対戦させた時のシュート数とシュート成功数を示す．どのアルゴリズムにおいても，提案手法による探索を行った報酬値を用いた場合のシュート数が多くなっている．シュートに至るまでの行動の流れを効率よく学習できている結果と考えられる．よって提案手法を用いることで同じ探索時間で，一括で報酬値を探索する場合よりも行動の流れを効率よく学習できる報酬値を設定できているためと考えられる．

以上のことから，提案手法がエージェントの学習に用いる報酬値の効率的探索に有効であると考えられ，提案手法を用いることで，一括で報酬値の探索を行う場合よりも短い探索時間で学習に適切な報酬値を設定できる可能性を示した．

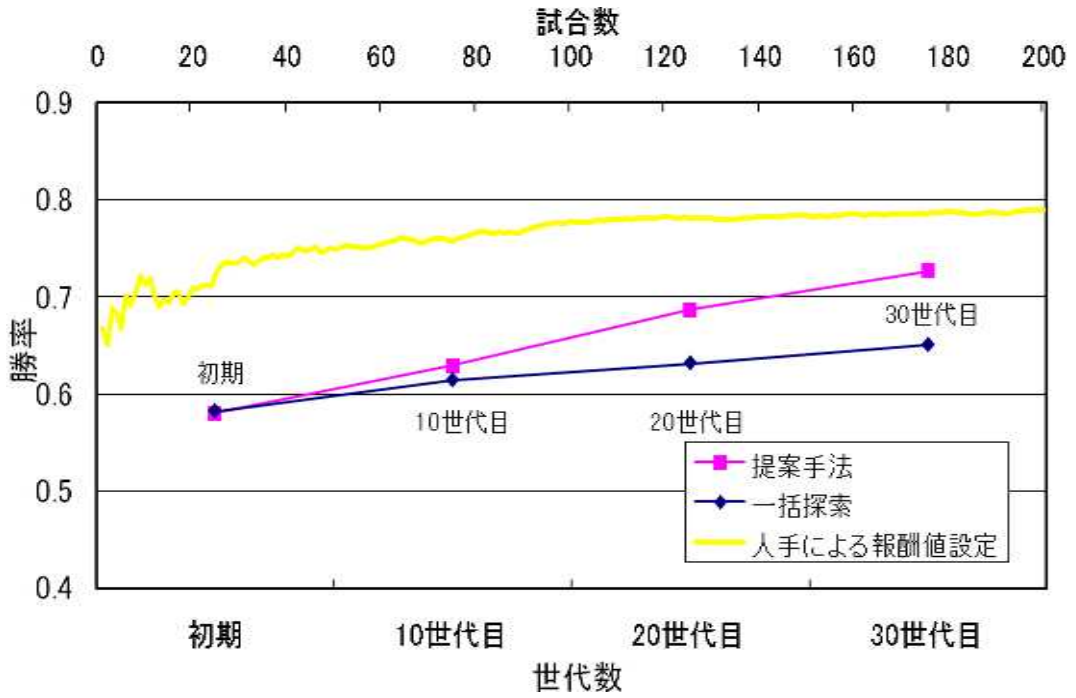

図 24 提案手法と一括による探索の比較

（対アルゴリズム A）

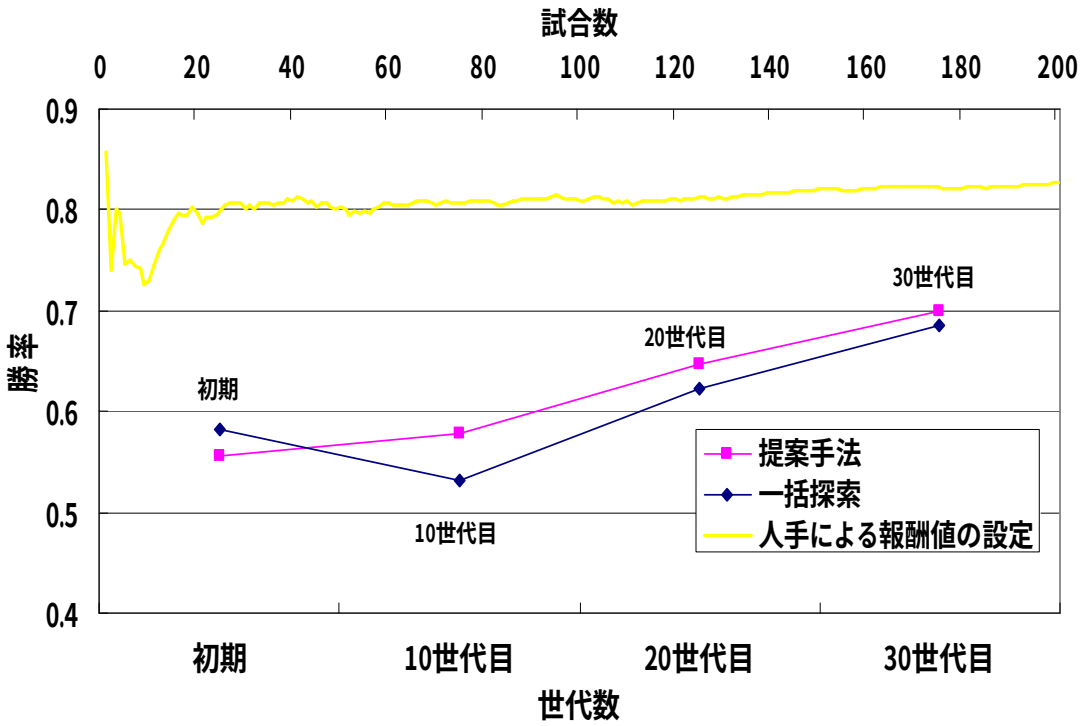

図 25 提案手法と一括による探索の比較

（対アルゴリズム B）

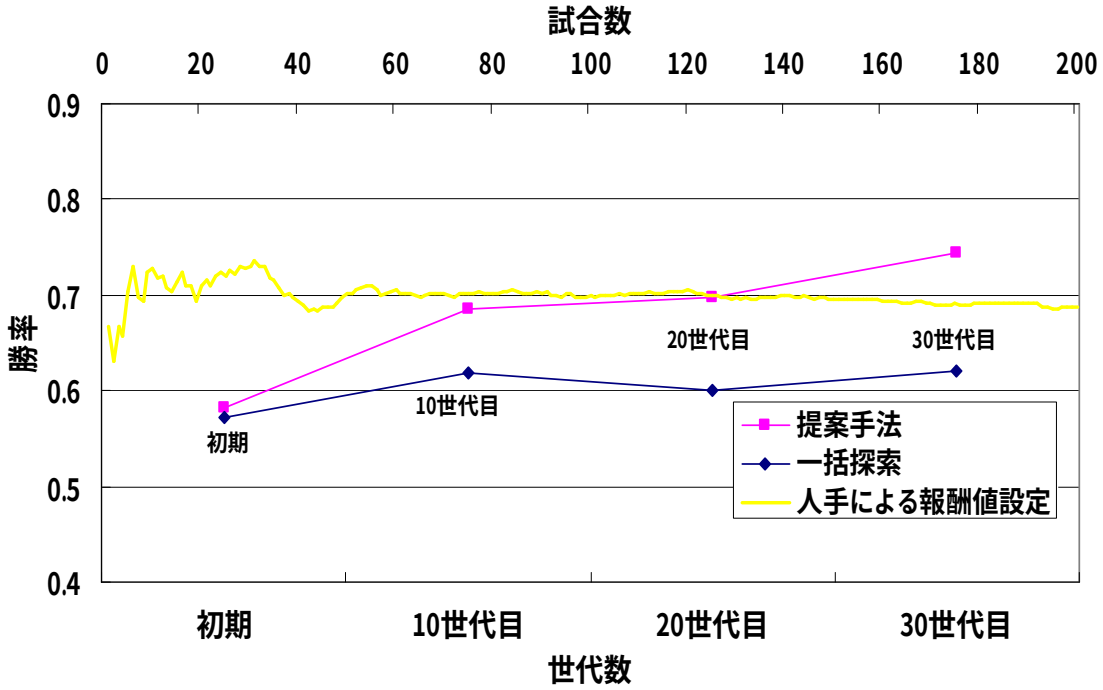

図 26 提案手法と一括による探索の比較
（対アルゴリズム C）

表 2　対アルゴリズム A における 30 世代目の報酬値を用いた場合のシュート数

	シュート数	シュート成功数
提案手法	408.6	105.4
従来手法	276.1	63.0

表 3　対アルゴリズム B における 30 世代目の報酬値を用いた場合のシュート数

	シュート数	シュート成功数
提案手法	371.6	91.9
従来手法	329.3	77.8

表 4　対アルゴリズム C における 30 世代目の報酬値を用いた場合のシュート数

	シュート数	シュート成功数
提案手法	326.7	102.5
従来手法	253.1	59.6

8. まとめ

本稿では，ビデオゲームのエージェントが，強化学習を行う際に用いる報酬値の探索に対し GA を適用した．特に報酬値を，依存関係を考慮した 2 つのグループに分割して交互に探索することを提案した．題材としてサッカーゲームを用いて実験を行い，その結果から，提案手法を用いて探索した報酬値を用いたチームの勝率が上昇し，学習効率が上昇していることが確認できた．また，本稿の提案手法と，報酬値を一括して探索する従来手法を比較した場合，提案手法は短い探索時間で，勝率が高く，勝率の立ち上がりが早くなる報酬値を見つけることができた．このことから，報酬値が多数あり探索空間が広い場合でも，報酬値のグループ分けを行って GA を適用した探索を行うことで，一括で探索を行う場合よりも効率よく学習に適切な報酬値を設定できる可能性を示した．

9. 参考文献

[1] J. H. Holland, “Adaptation in Natural and Artificial Systems” The MIT Press, Cambridge, MA, 1992. The University of Michigan Press Ann Arbor, 1975

[2] Y. Sato and R. Kanno, "Event-driven Hybrid Learning Classifier Systems for Online Soccer Games", In Proceedings of the 2005 IEEE Congress on Evolutionary Computation. IEEE Press, Edinburgh, 2005, pp. 2091-2098.

[3] Y. Sato and Y. Akatsuka and T. Nishizono, "Reward Allotment in an Event-driven Hybrid Learning Classifier System for Online Soccer Games", In Proceedings of the 2006 Genetic and Evolutionary Computation Conference. ACM Press, 2006, pp. 1753-1760.

[4] Y. Akatsuka, and Y. Sato, "Reward Allotment Considered Roles for Learning Classifier System for Soccer Video Games", Proc of the 2007 IEEE Symposium on Computational Intelligence and Game (2007)

[5] Y. Sato and Y. Inoue and Y. Akatsuka, “Applying GA to Self-allotment of Rewards in Event-driven Hybrid Learning Classifier Systems”, In Proceedings of the 2007 IEEE Congress on Evolutionary Computation. IEEE Press, 2007, pp. 1800-1087.

[6] Sutton. R. S, Barto. A. G. "Reinforcement Learning: An Introduction" The MIT Press, Cambridge, MA, 1998.

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

[8] Kitano. H, Asada. M, Kuniyoshi. Y, Noda. I, Osawa. E, and Matsubara. H, “RoboCup: A challenge problem for AI”, AI Magazine, Vol. 18, 1997, 73-85.

[9] S. M. Gustafson, and W. H. Hsu, “Layered Learning in Genetic Programming for a Cooperative Robot Soccer Problem”, In Proceedings of the Fourth European Conference on Genetic Programming, 2001, pp. 291-301.

[10] S. Luke, “Genetic Programming Produced Competitive Soccer Softbot Teams for RoboCup 97”, In Proceedings of the Third Annual Genetic Programming Conference. Morgan Kaufmann Publishers, San Francisco, CA, 1998, pp. 204-222.

[11] A. D. Pietro, L. While, and L. Barone, “Learning in RoboCup Keepaway using Evolutionary Algorithms”, In Proceedings of the Fourth Annual Genetic and Evolutionary Computation Conference. Morgan Kaufmann Publishers, San Francisco, CA, 2002, pp. 1065-1072.

[12] R. K. Belew, and M. Gherrity, “Back Propagation for the Classifier System”, In Proceedings of the Third International Conference on Genetic Algorithms. Morgan Kaufmann Publishers, CA, 1989, pp. 275-281.

[13] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA, 1989.

[14] J. H. Holland, “Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems”, In Michalski, R. S. et al. (eds.): Machine Learning II, Morgan Kaufmann Publishers, CA, 1986, pp. 593-623.

[15] R. L. Riolo, “Bucket brigade performance: I. Long sequences of classifiers, genetic algorithms and their application”, In Proceedings of the Second International Conference on Genetic Algorithms. Lawrence Erlbaum Associates, Publishers, 1987, pp. 184-195.

[16] R. L. Riolo, “Bucket brigade performance: II. Default hierarchies”, In Proceedings of the Second International Conference on Genetic Algorithms. Lawrence Erlbaum Associates, Publishers, 1987, pp. 196-201.

[17] R. L. Riolo, “The emergence of coupled sequences of classifiers”, In Proceedings of the Third International Conference on Genetic Algorithms. Morgan Kaufmann Publishers, CA, 1989, pp. 256-263.

[18] Tanese Reiko. “Distributed genetic algorithms”, Proc 3rd International Conference on Genetic Algorithms, 1989