

# エージェントのルール学習における GA と GP の特性比較と融合化による性能向上

西村 崇征<sup>†</sup>      岩沼 宏治<sup>‡</sup>      鍋島 英知<sup>‡</sup>

<sup>†</sup> 松下ソリューションテクノロジー株式会社

<sup>‡</sup> 山梨大学 大学院 医学工学総合研究部

{iwanuma, nabesima}@iw.media.yamanashi.ac.jp

## 概要

本研究の目的は、エージェントの行動学習問題において、GA と GP のどちらの手法がより有効であるかを明らかにすることである。我々は、GA と GP を制約付き経路問題に対する行動ルール学習に適用し、その特性を実験的に比較した。その結果、不完全情報に対する頑健性や多段階学習の結果など全体的に GP の方が優れているが、解の近傍における局所的探索では GA の方が良いという結果が得られた。この結果を基に、両手法を融合化した新手法を 2 つ提案し、性能の向上を実証的に確認した。

**キーワード:** 遺伝的アルゴリズム, 遺伝的プログラミング, エージェント, ルール学習

## 1 はじめに

近年、最適化問題に対して遺伝的探索手法である遺伝的アルゴリズム [伊庭 94](Genetic Algorithm: GA) 及び、遺伝的プログラミング [伊庭 96](Genetic Programming: GP) が有効であると認められ、数多くの研究がなされている。最適化問題として有名な巡回セールスマン問題などには GA が適しており [伊庭 94, Nagata 97, Valenzuela 99], 関数合成などには GP が考案されよく用いられている [関岡他 00].

一方、現在エージェントに行動を学習させる研究も盛んに行われている。行動学習には強化学習 [荒井他 98][伊藤他 01] など幾つかの方法が存在するが、エージェントが認識する状態数が爆発的に増加してしまうような環境では、GA や GP は非常に有効な行動学習の手法である [山田 98].

[山田 98] では、移動ロボットの行動学習に GA を用いている。一方、[片上他 01] では GP を用いてロボットの行動学習が行われている。また、マルチエージェントの学習でも、様々な手法が用いられている [Iba 98, Luke 98]. このように、これまで行動学習には GA と GP の両手法が用いられてきている。一般には行動決定には決定木を用い、GP によって学習を行った方が自然かつ効率的であると予想される。しかし、GA と GP のどちらがこれらの問題に対し有効な手法であるかという点について直接的に比較した研究は、これまでのところ筆者の知る限りでは無い。

GA と GP の特性、能力を直接的に比較することは極めて難しく、殆んど行われていない。当然、対象とする問題領域における適性が異なることから、一般的な比較は無理であり、またその意義も薄い。本研究は、エージェントの行動ルール学習問題において、GA と GP の特性を解明し、両者の長所を組み合わせ、より性能の良い遺伝的探索手法を提案することを目的とする。

本研究では GA と GP を制約付き経路問題に対する行動ルール学習に適用し、その特性を実験的に比較した。その結果、不完全情報に対する頑健性や多段階学習の結果など全体的に GP の方が優れているが、解の近傍における局所的探索では GA の方が良いという結果が得られた。この結果を基に、両手法を融合化した新手法を 2 つ提案し、性能の向上を実証的に確認した。

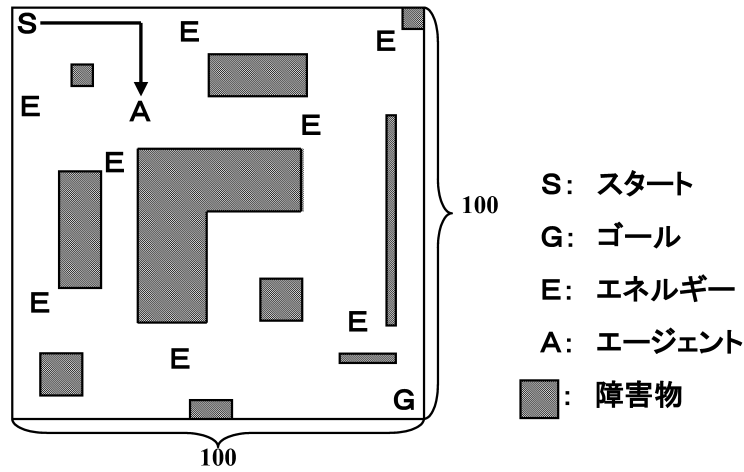


図 1: 制約付き経路問題の例

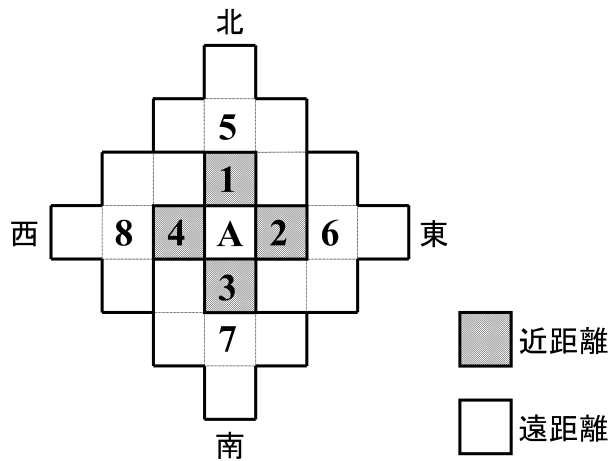


図 2: 視界内の領域分割

## 2 問題設定

### 2.1 制約付き経路問題

本研究で扱う制約付き経路問題とは、 $100 \times 100$  の格子状のフィールド上をエージェントがある行動ルールの下で、スタートからゴールに到達可能な経路を求める問題である (図 1)。フィールド上での方角は、上下左右の順に北南西東とし、東の方角を X 軸、南の方角を Y 軸とする。フィールド上の各マスの取り得る値は 'スタート', 'ゴール', 'エネルギー', '障害物', '何ものなし' の 5 値である。このような環境でエージェントは行動ルールに基づき行動し、ゴールを目指す。但し、制約としてエージェントはエネルギーを持ち、移動する毎に消費する。また初期エネルギーだけではゴールに到達できず、途中でエネルギーを補給する必要がある。本研究では、エージェントがゴールに到達可能な行動ルールを GA と GP を用いて学習する。

表 1: 視界と周囲の状態総数の関係

視界の距離	視界内のマス数	周囲の状態総数
1	4	297
2	12	4,782,969
3	24	$4.8 \times 10^{12}$
4	40	$3.3 \times 10^{20}$
5	60	$1.7 \times 10^{30}$
圧縮時	8	41,553

表 2: 代表値を決定するルールと優先順位

優先順位	代表値決定ルール
1	領域内に'ゴール'があれば値は'ゴール'
2	領域内に'スタート'があれば値は'スタート'
3	領域内に'エネルギー'があれば値は'エネルギー'
4	領域内で'障害物'の数が'何もし'の数以上ならば 値は'障害物', '障害物'の数が'何もし'の数より 小さければ値は'何もし'

## 2.2 エージェントの能力

エージェントは周囲の状態を認識するためのセンサとして視界を持っており、視界の距離はタスク実行前に2~5の範囲で任意に設定可能である。例えば、視界3ではエージェントは半径3マス以内の範囲を見ることができる。しかし、視界が広がるに連れて視界内のマス数が増加するので、エージェントが認識する周囲の状態総数は爆発的に増加してしまう。各マスの取り得る値は5値であるが、タスクの性質上スタートとゴールはフィールドに1つずつしか存在せず、且つスタートとゴールは同時にエージェントの視界に入ることは無いため、周囲の状態総数は式(1)で与えられる。式中の $S$ は周囲の状態総数を表し、 $N$ は視界内のマス数を表している。

$$S = 2 \times N \times 3^{(N-1)} + 3^N \quad (1)$$

視界の拡大に伴う視界内のマス数と周囲の状態総数を表1に示す。視界の距離2の場合において、周囲の状態総数は既に400万通りを越えており、取り扱いが非常に困難である。そこで、視界内を東西南北の4方向、更に各方向において遠近を区別し、計8領域に分割(図2)し、各領域をその領域内の支配的な状態で代表させ、エージェントが認識すべき状態総数を圧縮する。領域の代表値決定方法を表2に示す。このため、エージェントの状態認知能力は不完全となり、視界が広いほど不確実性が増す。圧縮時の周囲の状態総数は表1に示す通りである。

## 2.3 遺伝子のコーディングと進化的学習

GA 遺伝子は、図3の通りである。エージェントが認識し得る周囲の状態総数は表1より41553通りであるため、遺伝子配列長を41553とした。エージェントは状態エンコード表から現在の視界内の状態を状態番号に変換し、その後、GA 遺伝子配列において、現在の状態番号の行動コードを調べて、行動を決定する。

一方、GP 遺伝子は図4のような木構造をしており、現在の視界内の状態を入力として受け取り、それに対する行動を決定する決定木になっている。非終端記号には条件文番号が格納されており、番号毎

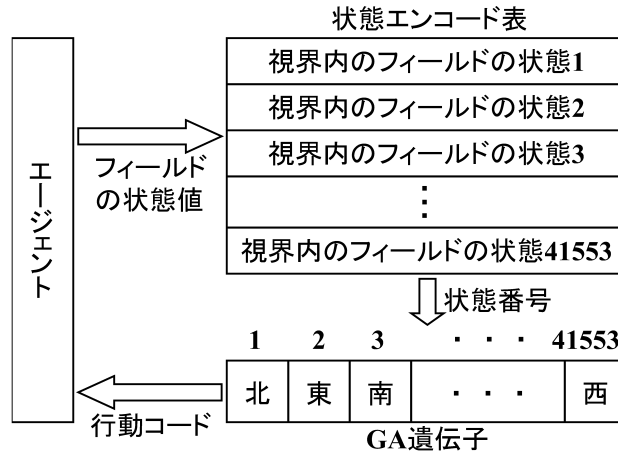


図 3: GA 遺伝子と行動決定方法

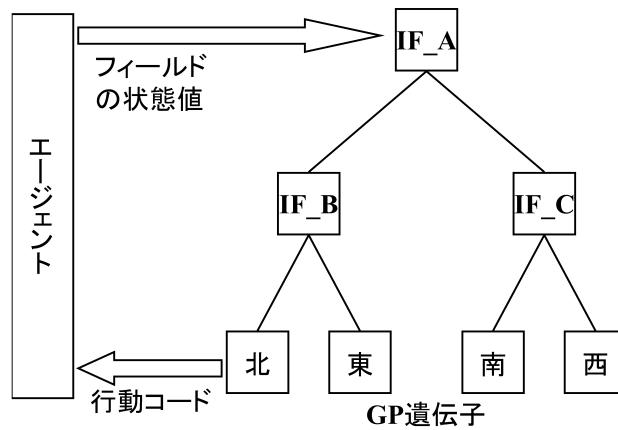


図 4: GP 遺伝子と行動決定方法

に条件文が一意に決定される。条件文は、視界内の領域 A が値 B を取っているか否かで表され、A が 8 通りで B が 5 通りであることから、条件文は 40 種類からなる。

本研究は GA と GP の比較を目的としているため、両方で用いる手法に可能な限り差が出ないようにしたい。そのため、GP の交叉は部分木の交叉とし、GA では GP の部分木交叉に相当すると考えられる 2 点交叉を用いた。

また、既存のエージェントの行動学習の研究では、エージェントの任意の状態に対して「動作列」を返すコーディングが多数見受けられる [片上他 01] が、本研究では GA と GP とともに、任意の状態に対して「単一の行動」を持つようにコーディングを行った。

以上に述べたコーディング方法では、GA 遺伝子の配列は固定長であるが、GP 遺伝子の決定木は固定長ではない。そのため、GP 遺伝子が大きく成長してしまうと、GA と GP の遺伝子の持つ情報量に格差が生じてしまう。これを防ぐために、GP 遺伝子の大きさに制限を設け、決定木が一定の深さ以上に成長しないようにした。この深さの制限値は、完全二分木をヒープ木の要領で配列に格納したとき、GA 遺伝子配列 (長さ 41553) に納まる最大の完全二分木の深さ 14 とした。

遺伝的探索の手続きを図 5 に示す。エージェントがゴールに到達する、もしくは規定世代数まで進化することが終了条件である。次世代遺伝子の選択方法にはルーレット選択とエリート保存戦略を用いる。

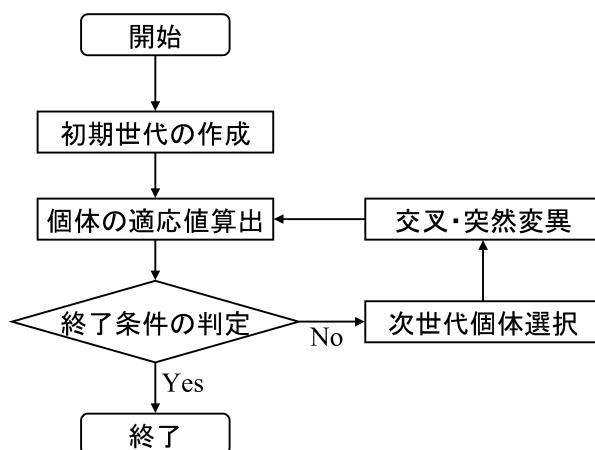


図 5: 遺伝的探索手法の手続き

エリート保存戦略によって現世代からエリート上位 10 個体を次世代に引き継ぎ、残りの個体はルーレット選択で選出する。

遺伝子の評価をする  $fitness$  関数は式 (2) とする。  $X_A, Y_A$  はエージェントの行動終了位置の座標で、  $X_G, Y_G$  はゴールの座標である。 制約付き経路問題ではゴール到達がタスク完了であるので、ゴールに近いほど適応値が高くなる。 また、次世代遺伝子の選択にルーレット選択を行う際、適応値をそのままルーレットの領域面積として用いるため、  $fitness$  値がマイナスの値を取らないように、エージェントとゴール間のマンハッタン距離を 200 から引いた値を  $fitness$  値として用いた。

$$fitness = 200 - (|X_G - X_A| + |Y_G - Y_A|) \quad (2)$$

### 3 GA と GP の比較

#### 3.1 実験設定

GA と GP の能力の比較を行うために、前述の制約付き経路問題のフィールドを 5 つ用意し、3 つの実験を行う。 5 つのフィールドは人手で作成した。 各フィールドのスタートとゴールの座標は同じであり、エネルギーの配置と障害物の配置と向きが異なっている。 実験結果は 10 回の試行の平均値である。 実験の各種パラメータの設定は表 3 の通りである。

実験 1 は、エージェントの視界情報の不確実性が探索に及ぼす影響を調べるため、フィールド 1 においてエージェントの視界の距離を 2 から 5 まで変化させて実験を行う。

実験 2 は、環境の変化に対する探索能力の頑健性を調べるために、5 つのフィールドを用いて実験を行う。 視界の距離は 5 に固定する。

実験 3 は、訓練事例を用いて多段階学習を行ったとき、同様の学習結果が得られるか調べるため、GA と GP にクロスオーバー検定を適用する。 テスト事例がフィールド 1 から 5 の場合までの 5 つのパターンをそれぞれテスト事例とし、残りの 4 つのフィールドを訓練事例として実験を行う。 視界の距離は 5 に固定する。

表 3: 各種パラメータの値

パラメータ	値
遺伝子数	100
規定世代数	200
交叉率	0.8
突然変異率	0.05
エージェントの初期エネルギー量	100
エネルギー拾得時のエネルギー増加量	50
GP 遺伝子の初期の木の深さ	7
GP 遺伝子の木の深さの最大長	14

### 3.2 比較実験の考察

図 6 と図 8 は、実験 1 での各世代における最大適応値の平均を示したグラフである。GA では視界の範囲が広がるに連れてゴールに到達するまでの世代数が増加していることが分かる。一方、GP では視界が変化してもほぼ同一なグラフを描いている。この結果から、GP は GA に比べ素早い進化をしていることと、GA より視界の拡大に伴う入力情報の雑音に強いことが分かる（GP が頑健な理由の考察は 4.2 節の脚注を参照して頂きたい）。また、図 7 と図 9 は規定世代まで進化させた 100 個の遺伝子の探索終了位置の分布の例を示している。GA ではフィールドの北と東に分布が偏在しているが、GP ではフィールド全体に分布し、フィールド全体を探索していることが見て取れる。これより、GP はフィールドを大域的に探索していることが分かる。

次に、実験 2 の結果を表 4 と表 5 に示す。この結果より、GP は GA に比べてどのテストフィールドでもほぼ一定の値を取っているため、環境の変化に対する頑健性が高いといえる。また、ゴール発見世代数とゴール到達世代数ともに GP の方が GA より良いことが分かる。しかし、ゴールを発見してからゴールに到達するまでに要した世代数を比べると、GA の方が GP より世代数が少ない結果となった。これは、GA 遺伝子と GP 遺伝子が構造的に異なるためである。GP では、視界内にゴールがあるかを判定する条件文はエージェントがゴール発見前では適応値に影響を及ぼさないため、ゴール発見後においてゴールの有無を判定する条件文が Building Block の一部として構築されている確率は高くないと予想される。つまり、ゴールを発見した後にゴール到達に必要な Building Block を構築するためある程度の世代を要してしまうと考えられる。一方の GA では、ゴールを発見可能な遺伝子が複数いた場合、そのいずれかにゴールへ向かう行動コードが GA オペレータにより発現されれば適応値は更新されるため、GP ほど世代を必要としないと考えられる。

続いて実験 3 の結果を図 10 から図 13 に示す。図 10 と図 12 は 4 つの訓練事例を用いて学習した遺伝子と、全く訓練していない遺伝子を 5 つのテスト事例に適用し、第一世代での最良の遺伝子の適応値を比較したグラフである。GA と GP を相対的に比較すると、両手法ともに全てのテスト事例で適応値が改善され、平均で 50 も上昇した。これより、どちらの手法においても学習により適応値が改善されていることが分かる。しかし、両手法の適応値を絶対的に比較すると、GA では学習で適応値は上昇しているが、第一世代でゴールに到達するところまでは学習できていない。一方の GP では、学習した場合の適応値は最大適応値の 200 に近い値を示しており、ゴールに到達するための行動ルールが確実に学習されていることが分かる。

また、図 11 と図 13 は、図 10 と図 12 の実験結果を導き出した遺伝子をそのままテスト事例でも学習させた場合の、ゴール到達に要した世代数とその内訳である。GA ではフィールド 2 をテスト事例とした結果において、学習無しの結果より学習有りの結果の方が悪くなったことから、訓練フィールドによる多段階学習が失敗に終わっていることが分かる。しかし、それ以外のテスト事例では結果が改善され

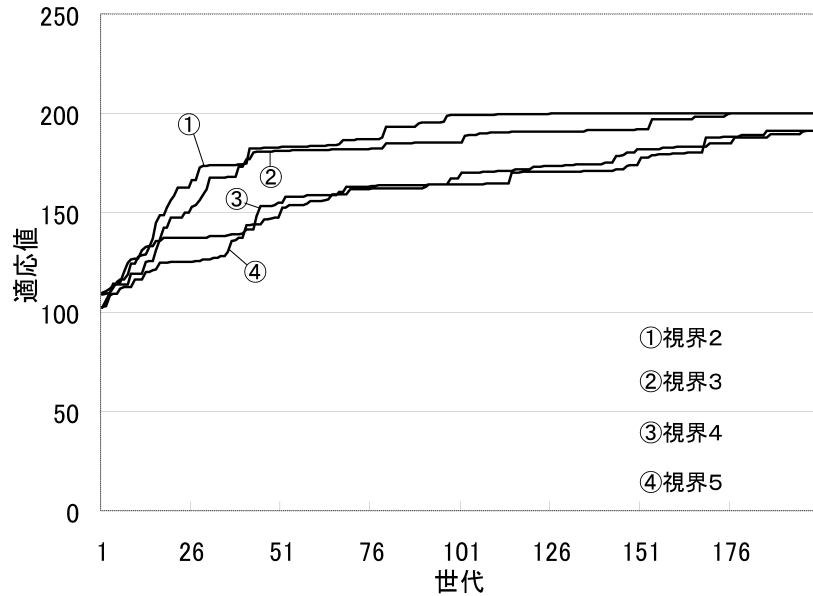


図 6: GA における視界毎の適応値

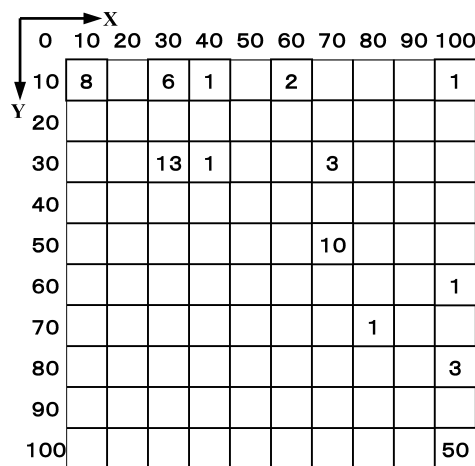


図 7: 200 世代目における GA 遺伝子の探索終了位置の分布

ており、平均では訓練有りの場合は、訓練無しの場合の半分の世代数で探索を終了している。減少した世代は主にスタートしてからゴールを発見するまでの大域的探索に要した世代である。多段階学習により大域的探索に必要な行動ルールが学習されているといえる。

一方、GP では訓練することにより、全てのテスト事例で結果が改善されている。また、GA に比べて学習結果が良く、テスト事例がフィールド 3 と 5 の場合では、どちらも 10 回の試行の全てにおいて第 1 世代でゴールに到達している。更に平均では、訓練無しの結果の 6 分の 1 の世代数で探索を終了している。減少した世代は主にゴールを発見してからゴールに到達するまでの局所的探索に要した世代で、多段階学習により GP の欠点である局所的探索に必要な行動ルールが学習されていることが分かる。以上より、GP は GA より訓練による学習がスムーズに進み、多段階学習において頑健であるといえる。

以上 3 つの実験結果を総合的に評価すると、GA より GP の方が性能が良いが、ゴールの近傍での探索能力は GA の方が良いことから、両者ともに一長一短があることが分かった。本研究で用いた問題は、

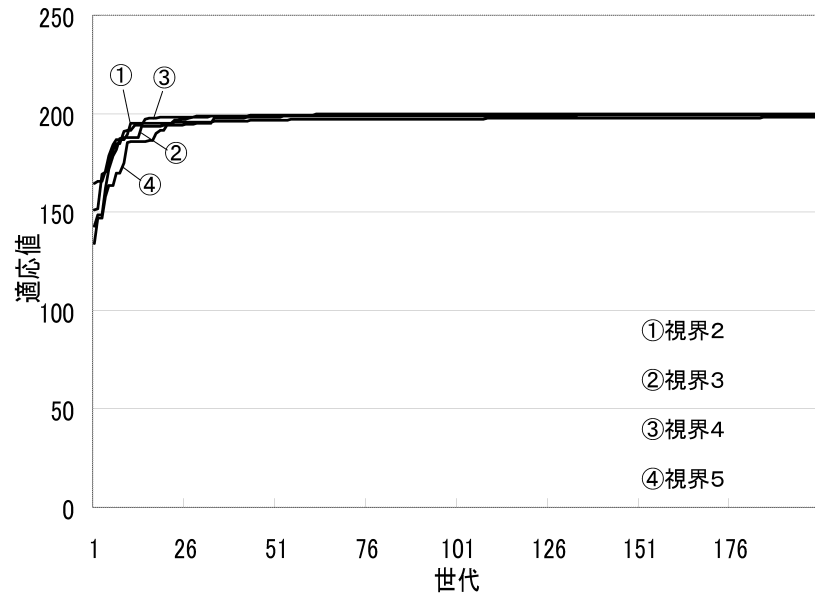


図 8: GP における視界毎の適応値

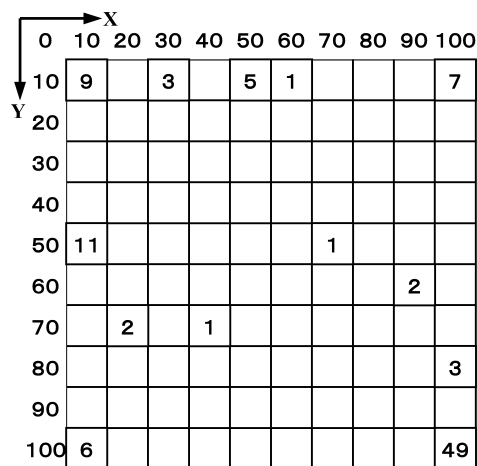


図 9: 200 世代目における GP 遺伝子の探索終了位置の分布

探索序盤のゴールの大体の位置を探る大域的探索と、探索終盤のゴールの正確な位置を探る局所的探索の 2 つの側面を持ち合わせている。本論文の結果から GA は局所的探索、GP は大域的探索に向いていることが分かった。しかし、クロスオーバー検定の結果では、両手法ともに各々の欠点となっている部分の探索を訓練により克服しており、特に GP では探索能力が大きく改善された。

#### 4 新手法の提案

前章までの結果を踏まえ、より高性能な学習を目指して、以降に GA と GP の単純ハイブリッド法と初期 GA 遺伝子のブロック化法という 2 つの新しい手法を提案する。



表 4: GA の結果

field No	ゴール発見時の世代 (1)	ゴール到達時の世代 (2)	(2)-(1)	ゴール到達率
1	92.60	106.40	13.80	1.00
2	72.50	91.50	19.00	0.40
3	44.90	65.20	20.30	1.00
4	62.70	70.60	7.90	0.90
5	42.00	54.00	12.00	1.00
平均	62.94	77.54	14.60	0.86

表 5: GP の結果

field No	ゴール発見時の世代 (1)	ゴール到達時の世代 (2)	(2)-(1)	ゴール到達率
1	5.60	30.10	24.50	0.80
2	17.10	44.10	27.00	1.00
3	6.20	39.90	33.70	0.90
4	9.40	47.00	37.60	0.80
5	8.50	38.90	30.40	1.00
平均	9.36	40.00	30.64	0.90

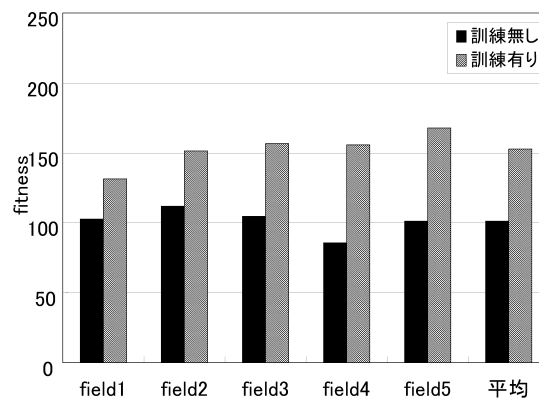


図 10: GA におけるクロスオーバー検定の *fitness* の比較

#### 4.1 GA と GP の単純ハイブリッド法

前述の GA と GP の比較実験の結果において、ゴールの近傍では GP より GA の探索能力が高いことに注目し、GA と GP のハイブリッド化を試みた。ゴール発見前は GP で探索し、ゴール発見後は GP 遺伝子を GA 遺伝子に変換し、その後 GA で探索する手法を取った。GP 遺伝子から GA 遺伝子への変換は、視界内の取り得る全状態を GP 遺伝子に入力し、出力として得られた行動コードを GA 遺伝子配列に格納すれば可能である。

表 6 に単純ハイブリッド法を 5 つのフィールドに適用した結果を示す。各フィールドの結果は全て 10 回の試行の平均値である。

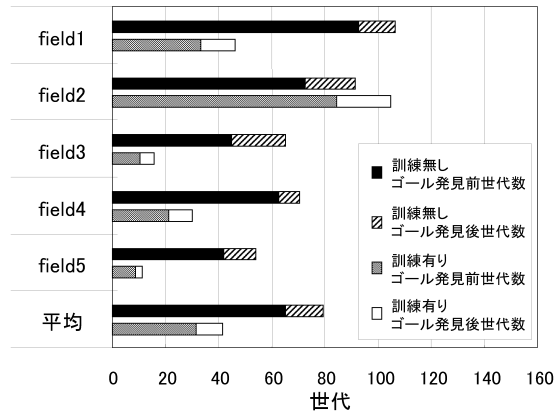


図 11: GA におけるクロスオーバー検定の世代数の比較

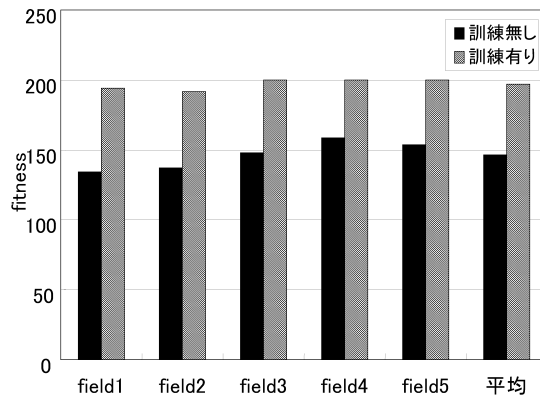


図 12: GP におけるクロスオーバー検定の fitness の比較

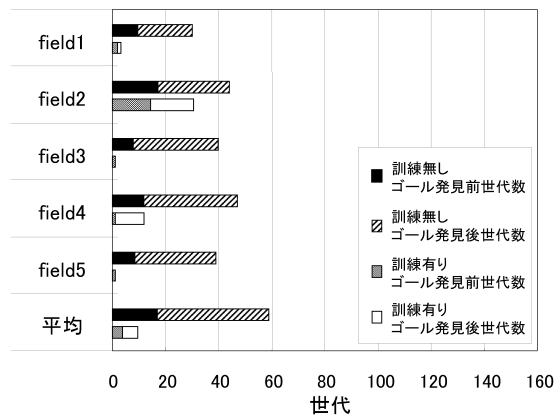


図 13: GP におけるクロスオーバー検定の世代数の比較

## 4.2 初期 GA 遺伝子のブロック化法

単純ハイブリッド法は、GP を主とした手法であるが、それとは逆に GA を主とした手法が、この初期 GA 遺伝子のブロック化法である。これは初期世代 GA 遺伝子の作成時に、遺伝子配列を大小 64 個の

表 6: ハイブリッド法の結果

field No	ゴール発見時の世代 (1)	ゴール到達時の世代 (2)	(2)-(1)	ゴール到達率
1	10.30	35.60	25.30	1.00
2	18.60	37.50	18.90	1.00
3	13.20	25.70	12.50	1.00
4	10.90	28.30	17.40	1.00
5	8.00	20.60	12.60	1.00
平均	12.20	29.54	17.34	1.00

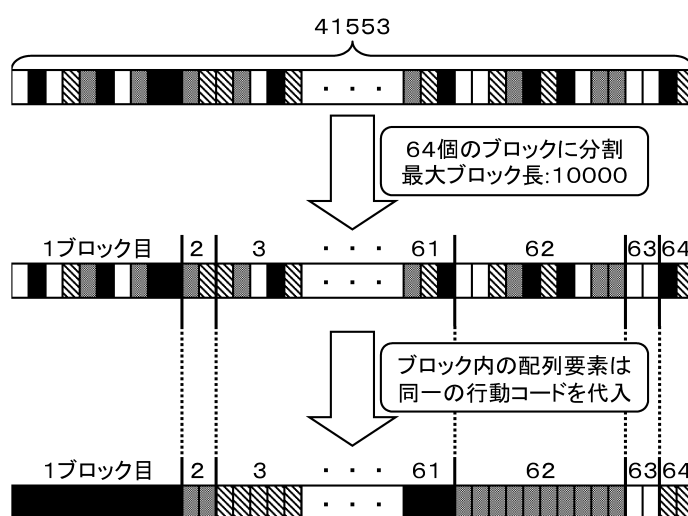


図 14: 初期 GA 遺伝子のブロック化

ブロックにランダムに分け、各ブロック内での行動コードは単一の値を取るようにしたものである (図 14)。ブロックの大きさは最小のブロックで 1, 最大のブロックでは 10000 とした。

GP を用いて探索を行った実験結果 (図 8) において, MDL 基準などを導入していないにも関わらず, 進化が進むにつれて遺伝子の行動決定木が小さくなるという傾向が認められた。初期世代の遺伝子は表 3 より深さ 7 の完全二分木であるので葉の数は 128 である。それが, ゴール到達時には平均 104 まで減少し, 規定世代まで進化させた場合は平均 29 まで減少した。このような小さな木では周囲の状態を詳細に分類し, エージェントの行動を変化させるのは無理である。しかし, GA より GP の方が明らかにゴールを発見するまでの世代数が少ないことから, 各葉の使用率に大きな差があり, 使用率の高い葉にゴール方向へ向かう行動が割り当てられていると考えられる。我々はこの行動をデフォルト行動と考えた<sup>1</sup>。GP 遺伝子が持つデフォルト行動を GA 遺伝子にも持たせるため, 初期 GA 遺伝子をブロック化することを考案した。これにより, GA にも強制的にデフォルト行動を持たせ, 行動コードの使用率に偏りを持たすことができると考えた。表 7 に実験の結果を示す。

<sup>1</sup>デフォルト行動の存在が GP を不完全情報に対して頑健にしていると考えられる。

表 7: 初期 GA 遺伝子ブロック化法の結果

field No	ゴール発見時の世代 (1)	ゴール到達時の世代 (2)	(2)-(1)	ゴール到達率
1	15.00	26.70	11.70	1.00
2	69.56	89.67	20.11	0.90
3	19.60	34.10	14.50	1.00
4	8.20	22.20	14.00	1.00
5	8.00	21.50	13.50	1.00
平均	24.07	38.83	14.76	0.98

### 4.3 新手法の考察

表 6 から単純ハイブリッド法の結果を考察する。GP ではゴール発見からゴール到達までに約 30 世代も要しているが、ゴール発見後に GA で探索を進めることにより約 17 世代まで減少した。GA ではゴール発見から平均 14.6 世代でゴールに到達していることから、純粋に GA の局所探索能力が発揮されていると考えられる。また、全ての試行においてゴールに到達していることから、フィールドの変化に対し頑健であるといえる。

次に、表 7 から初期 GA 遺伝子のブロック化法の結果を考察する。GA ではスタートからゴール発見までに約 63 世代もかかっているが、初期遺伝子をブロック化することにより約 24 世代まで減少させることができた。この手法は初期世代においてのみ遺伝子をブロック化するため、交叉や突然変異を繰り返すことにより次第にブロックが細分化され、探索の後半では単純 GA に近づく。これにより、探索終盤の局所探索でも GA と同様な探索能力を発揮することが可能である。また、ゴール到達率も上昇し、頑健性も向上している。

ここで、2つの新手法の結果を GA と GP におけるクロスオーバー検定の結果と比較してみると、多段階学習を行った単純 GP の結果が圧倒的に良い。つまり、計算機によるシミュレーションが可能な問題で、多段階学習という非常に大きな学習コストをかけても良い状況では、単純 GP はかなり有効であると考えられる。だが、計算機によるシミュレーションが不可能な問題、即ち直接実環境にて実験を行わなければならない場合には、多段階学習は多大なコストがかかり無理である。このような場合においては、本稿で提案した新手法が有効な手段になると考えられる。

ここで、両手法を他の問題へ適用することを考える。ハイブリッド法では GP から GA への変更ポイントをどこに置くかという問題や、GP 遺伝子から GA 遺伝子へ変換可能なコーディングを考慮しなくてはならず、他の問題への適用は難しい点が多い。[伊藤他 01] では、前者の問題に対して「学習残エントロピー」等の指標の利用を提案しているが、本問題にそのまま適用することは難しい。一方、初期 GA 遺伝子のブロック化法では、初期世代の遺伝子をブロック化する以外は単純 GA と何ら変わりはない。よって、本研究で用いた制約付き経路問題と同じくデフォルト行動や決定が存在すると思われる問題、例えばロボットのエスケープ問題やクラスタリングの問題、決定木の合成等には容易に適用可能であり、効果が期待できる。

また、初期 GA 遺伝子のブロック化法は GP 遺伝子の葉の使用率に着目して考案された手法であるが、節点や葉の使用率に着目した研究は [片上他 01] でも報告されており、行動学習ではノード等の使用率を考慮することが重要である。

## 5 結論

本研究で用いた遺伝子は、GA は全状態に対し行動コードを持つ密な構造、GP は小さな決定木で分類される状態にしか行動コードを持たない粗な構造である。一般的に、本研究で用いたような問題では遺伝子構造が密でないと問題解決できないと考えがちであるが、そうではなく、探索序盤は遺伝子構造が粗であることが探索速度を向上させる 1 つの要因であることが分かった。

探索問題の基本は、探索序盤の大域的探索と終盤の局所的探索にあることは既によく知られている事実であるが、単純な GA や GP ではその特性から双方を行うことは難しい。今回の実験では、初期世代の GA 遺伝子をブロック化するような単純な方法で大域的探索能力の改善が可能であり、全体の探索速度が改善された点が重要である。

## 参考文献

- [荒井他 98] 荒井幸代, 宮崎和光, 小林重信: マルチエージェント強化学習の方法論-Q-Learning と Profit Sharing による接近-, 人工知能学会論文誌, Vol.13, No.4, pp.609-618, 1998
- [伊藤他 01] 伊藤 昭, 金淵 満: 知覚情報の粗視化によるマルチエージェント強化学習の高速化-ハンターゲームを例に-, 電子情報通信学会論文誌, D-I Vol.J84-D-I No.3 pp.285-293, 2001
- [伊庭 94] 伊庭齊志: 遺伝的アルゴリズムの基礎, オーム社, 1994
- [伊庭 96] 伊庭齊志: 遺伝的プログラミング, 東京電気大学出版局, 1996
- [Iba 98] Hitoshi Iba: Multi-Agent Reinforcement Learning with Genetic Programming, Genetic Programming 98, pp.167-172, 1998
- [片上他 01] 片上大輔, 山田誠二: ノード使用頻度に依存した交叉による進化ロボティクスの高速化, 人工知能学会誌, Vol.16, No.4, pp.392-399, 2001
- [関岡他 00] 関岡哲也, 船曳信生, 東野輝夫: 遺伝的プログラミングを用いた関数合成アルゴリズムの改良法の提案, 電子情報通信学会論文誌, D-I Vol.J83-D-I No.4 pp.407-417, 2000
- [Nagata 97] Y.Nagata and S.Kobayashi: Edge Assembly Crossover: A High-power Genetic Algorithm for the Traveling Salesman Problem, 7th Inter. Conf. on Genetic Algorithms, pp.450-457, 1997
- [Valenzuela 99] C.L.Valenzuela: Evolutionary Divide and Conquer (II) for the TSP, Genetic and Evolutionary Computation Conference'99, pp.1744-1749, 1999
- [Luke 98] S.Luke: Genetic Programming Produced Competitive Soccer Softbot Teams for RoboCup97, Genetic Programming 1998, pp.214-222, 1998
- [山田 98] 山田誠二: 行為に基づく環境モデリングのための移動ロボットの進化的行動獲得, 人工知能学会誌, Vol.14, No.5, pp.870-878, 1998