

## Star Trek ゲームプレーヤー意思決定モデルの進化

Alves Dias, Sonny

Université de Technologie de Belfort-Montbéliard | Graduate School of Design, Kyushu University

猪口, 裕香

九州大学芸術工学部

高木, 英行

九州大学大学院芸術工学研究院

<https://hdl.handle.net/2324/1434426>

---

出版情報：第2回進化計算学会研究会第8回進化計算フロンティア研究会合同研究会, pp.112-117, 2012-03. 進化計算学会

バージョン：

権利関係：

# Star Trek ゲームプレーヤ意思決定モデルの進化

## Evolving a Human Player Model for the Star Trek Game

Sonny ALVES DIAS<sup>12</sup> \* 猪口裕香<sup>3</sup> 高木英行<sup>4</sup>  
Sonny ALVES DIAS<sup>1\*</sup> Yuka INOKUCHI<sup>3</sup> Hideyuki TAKAGI<sup>4</sup>

<sup>1</sup> Université de Technologie de Belfort-Montbéliard

<sup>2</sup> 九州大学大学院芸術工学府

<sup>2</sup> Graduate School of Design, Kyushu University

<sup>3</sup> 九州大学芸術工学部

<sup>3</sup> Undergraduate School of Design, Kyushu University

<sup>4</sup> 九州大学大学院芸術工学研究院

<sup>4</sup> Faculty of Design, Kyushu University

**Abstract:** We make a Star Trek game player model with binary and fuzzy rules, parameterize the rules, and tune them using differential evolution (DE) and genetic algorithms (GA). First, we built a human player model player using binary and fuzzy rules that can fight with Star Trek game in a computer. Second, we parameterize the rules and apply DE and GA to them. This task was so difficult that DE and GA could not find optimum parameters that make the player model winner. To solve this situation, we apply new idea that changes the search environment gradually during the evolution process to find better solutions. We start from the simplest conditions of Star Trek game and gradually make them difficult according to the number of winnings of the human player model. This co-evolution-like strategy could find better rules parameters and the player model became stronger, while just applying DE and GA did not converge to its solution.

## 1 はじめに

シミュレーションゲームとはコンピュータ上で様々な状況を再現し、ゲームプレーヤがその状況に応じて自らの行動を決定するゲーム全般を指す。それらは (a) 戦争 (b) 経営や人材育成 (c) 飛行機や自動車の操縦、(d) リアルタイムストラテジー等に分類される。

ゲーム内に実装されているゲーム戦略は変化することはない。ゲームプレーヤが状況変化を感じるのは予めプログラムされた変化により生じている。ゲーム設計者の立場からは、ゲーム内で使用する戦略や行動のルール設計に多くの時間を必要であり、ゲームプレーヤの立場からは、自らの上達に合わせてゲームの難易度が上がって行くことが望ましい。

本研究の目的は2つある。第1の目的はゲームに対するプレーヤ意思決定モデルを設計しそのモデルを進化させること、第2の目的はゲーム戦略が進化する枠組みを導入し、第1目的で進化させたプレーヤ意思決定モデルを初期値に用いて、プレーヤ意思決定モデルと

ゲーム側の戦略ルールの双方を共に強くすることである。すなわち、本研究の第1段階では、妥当な強さのプレーヤ意思決定モデルを獲得し、第2段階ではゲーム戦略とプレーヤ意思決定モデルの共進化へと進む。

本論文の目的は、この第1段階を扱う事である。この目的を実現するために、2値論理ルールとファジィ論理ルールを用いて Star Trek ゲームに対するプレーヤ意思決定モデルを作成し、そのルールをパラメータ化し、進化計算を用いてそれらを最適化した。ここでは、いかに戦略を進化させるかが重要性であることを述べる。広い探索領域を探索するのは効率的ではなく、より最適領域を探索することに近づくためには、簡単な状態からゲームの進化をスタートさせ、敵「クリンゴン」の数を増やすことや対戦日数を減らすことによって、徐々にゲームの難易度（強さ）を上げていく必要がある。

本節に続いて、第2節ではプレーヤ意思決定モデルの作成に使用した Star Trek ゲームについて、第3節では Star Trek ゲームプレーヤ意思決定モデルの設計について解説する。第4節では設計したプレーヤ意思決定モデルを2つの異なる進化的戦略を用いて進化さ

\*連絡先：九州大学大学院芸術工学府  
〒815-8540 福岡市南区塩原4丁目9番1号  
E-mail: sonny.alvesdias@gmail.com

せ、第 5 節では実験結果について考察する。

## 2 Star Trek ゲームとその戦略

Star Trek ゲームは有名な宇宙対戦のテレビ番組に基づいて 1970 年代にアメリカで制作されたゲームである [5]。このゲームは一定対戦日数内に敵機をすべて撃破するもので、通常設定では、平均 17 機の敵機をゲーム内時間で 30 日以内に倒すことが求められる。

宇宙空間は  $8 \times 8$  象限の格子に分割されている。この小宇宙象限 (quadrant) 毎に名称 (Antares I, Antares II, Sirius I 等) が与えられ、プレーヤはこの名称によって現在どの小宇宙象限にいるかを確認することができる。また、各小宇宙象限はさらに  $8 \times 8$  区域 (section) の格子に分割されており、1 つの区域内には図 1 に示すようにプレーヤの機体 (エンタープライズ号) や宇宙基地、星、敵等の有無が表示される。

Now entering Vega II quadrant...

Combat Area Condition Red

```

-----
*      +K+
      *
      <*>
      *
-----
Stardate      3102
Condition     *RED*
Quadrant      4, 2
Sector        6, 6
Photon Torpedoes  10
Total Energy   2964
Shields       500
Klingons Remaining  17

```

図 1: 現在の小宇宙象限のスキャン結果を示すゲーム例。<\*> は自分の宇宙船エンタープライズ号、+K+ は敵機クリンゴン、\*は星を表す。

Star Trek ゲームに勝つための戦略の鍵は、自分のエンタープライズ号をどこに移動させるかやエンタープライズ号の資源の使い方にある。敵を倒すゲーム日数は限られており、その上移動すると与えられた対戦日数を消費してしまうため、プレーヤは最適な移動ルートを進む必要がある。プレーヤは長距離探索機能 (long range scan) を用いて、約 18 対戦日数を使って宇宙空間全体の情報を得ることができる。ある小宇宙象限から別の小宇宙象限に移動した場合、その移動距離に関わらず、ゲーム内時間で約 2 対戦日数を消費する。また、長距離探索機能で取得できるのは現在の小宇宙象限を中心とした  $3 \times 3$  の小宇宙象限の状況であるため、宇宙空間全体を探索 (図 2) するにはゲーム内時間で 18 対戦日数を消費し、移動を 9 回行う必要がある。

また、エネルギー量や光子魚雷数等にも限りがあるため、プレーヤはこれらの扱いにも注意しなければならない。光子魚雷は平均 17 機の敵に対し 10 発しか

Computer Record of Galaxy for Quadrant 7,2

	1	2	3	4	5	6	7	8
1	***	***	***	***	***	***	***	***
2	***	002	006	002	***	***	***	***
3	003	004	005	004	***	***	***	***
4	006	102	003	001	003	102	***	***
5	006	307	006	007	001	004	***	***
6	004	107	008	008	002	102	***	***
7	105	004	201	003	005	107	***	***
8	005	002	004	001	006	001	***	***

図 2:  $8 \times 8$  小宇宙象限に分割された銀河系地図例。数値の 100 の位の数値は敵機数、10 の位の数値は宇宙基地の数、1 の位の数値は星の数である。

い。エネルギーもまた貴重であり、これは移動や攻撃、機体の保護等あらゆることに用いられる。これらの資源は宇宙基地に戻ることで補充できるが、宇宙基地に行くには時間がかかる。効率的に戦うには機体の保護や攻撃に使用するエネルギーの総量と光子魚雷の総数等の資源の使い方が影響してくる。

ゲームプレーヤには、機体移動、2 種類の兵器による敵への攻撃、周辺環境の探索、機体やゲーム状況の情報取得、等の限られたコマンドが与えられている。機体の各部位はゲーム進行中に故障することもある。例えば、エンジン部分が故障した場合、プレーヤは通常で移動できなくなる。宇宙基地に戻った機体修理が可能であるが、自動的に修復するのを待つこともできる (しかしより長い時間がかかる)。

敵機クリンゴンの行動は非常に限られている。クリンゴンは他の小宇宙象限に移動することはできず、移動できるのは同じ小宇宙象限内だけである。また、クリンゴンは兵器を 1 つしか持たない。仮にプレーヤがクリンゴンに攻撃し、完全に倒さないまま別の小宇宙象限を移動した場合、プレーヤがその小宇宙象限に戻って来た時にはクリンゴンは損傷を完全修復して状態に戻っている。基本的にクリンゴンはプレーヤを見つけた時に攻撃するもので、エンタープライズ号が移動した場合にはクリンゴンも同じく移動して位置を変える。

ゲームのあらゆる状況に対応するため、人間のプレーヤは自宇宙船のすべての機能を使わなければならない。それ故に、移動ルートや資源をうまく活用して敵を見つけ出し、敵機を倒すための最良戦略を立てる能力が求められる。

### 3 Star Trek ゲームプレイヤーの意思決定モデル

人間のゲームプレイヤーを代替するプログラムはプレイヤー意思決定モデルのルールベース、推論エンジン、インタフェースの3つから構成される。プレイヤー意思決定モデルは2値論理ルール群とファジィ論理ルール群から成り、推論エンジンはこれらのルールを処理し Star Trek ゲームへのアクションを出力する。ゲームインタフェースは文字列のバッファを経て、Star Trek ゲームプログラムと、ゲームプレイヤープログラムとのコミュニケーションを行う。

Star Tre ゲームは命令と行動が交互に繰り返すターン制ストラテジー型のゲームなので、プレイヤー意思決定モデルが Star Trek ゲームプログラムとやり取りする1回の行動(1ターン)の行動論理をルールで記述すればよい。我々が設計した行動論理のフローチャートを図3に示す。1ターンでは敵クリンゴンへの攻撃やプレイヤーのエンタープライズ号が移動し、それに応じて敵の攻撃・移動や環境変化(損傷、エネルギーや対戦日数の変化)がある。

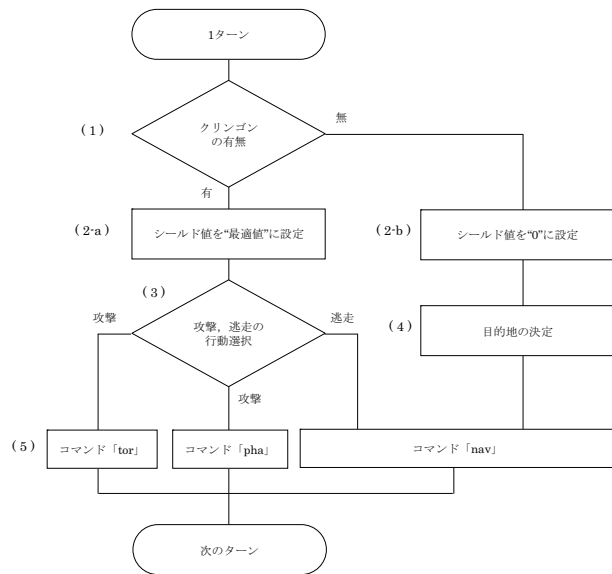


図3: プレーヤ意思決定モデルの行動フローチャート(1ターン)。図中の数字は更に複数のルールで構成される部分である。

作成したプレイヤー意思決定モデルのルールをパラメータ化し、進化計算でパラメータを最適化することでより強いゲームプレイヤーに仕上げる。最適化に必要なファジィ変数を抽出し、表1に示す。また、各変数に対して2~3個のメンバーシップ関数を割り当て、各メンバーシップ関数形状をパラメータ化して、次節の最適化実験で最適化する。

表1: ファジィ変数一覧。H, M, Lは「多い」「中位」, 「少ない」を, F, Nは「遠い」「近い」を表す

現在のエネルギー量	H / L
移動時の総エネルギー量	H / OK / L
残り戦闘日数	F / N
敵機がいる小宇宙象限への距離	F / N
宇宙基地までの距離	F / N
未開拓小宇宙象限への距離	F / N
敵機がない最も近い小宇宙象限への距離	F / N
光子魚雷の数	H / L
フェイザー砲使用時の総エネルギー量	H / M / L
シールド時の総エネルギー量	H / M / L

図3中の(3)の部分のファジィルールの一例を以下に示す。太字で示すファジィ数は三角形と台形のメンバーシップ関数で定義され、この形状を示すパラメータが進化計算で最適化される。

IF **小宇宙象限内の敵クリンゴン数が中位**で  
**残存エネルギーが高く**  
**残された対戦日数がOK**で  
**手持ち光子魚雷数が少ない**ならば,  
 THEN **光子魚雷を使う**

また、ファジィ数以外にもモデル最適化に必要なとする変数に、小宇宙象限の敵機クリンゴン数の「多い」, 「中位」, 「少ない」を決定するための閾値、その他の変数がある。これらで構成するルール数は1,800を超える。

### 4 プレーヤ意思決定モデルの進化

#### 4.1 進化実験条件

パラメータの最適化は差分進化(DE)と遺伝的アルゴリズム(GA)の2つのアルゴリズムを用いる。実験条件は10万世代まで行い、各世代では100回対戦し、その勝敗数を適応度で用いる。つまり、GAでは $10^7$ 回、DEでは $2 \times 10^7$ 回のゲーム対戦を行わせる。

フィットネス値は、勝敗数、撃墜した敵の数、勝敗時の消費した対戦日数の3つから評価される。したがって、まず第1の勝敗の評価値は次のように設定される。

$$\text{評価}_1 = \begin{cases} 10,000 & \text{勝った場合} \\ 0 & \text{負けた場合} \end{cases}$$

第2に最適化を強調するために、実際に倒した敵の数を用いる。

$$\text{評価}_2 = \text{撃墜敵機数} \times 100$$

これは3機の敵が登場するゲームよりも30機の敵が登場するゲームにおける勝利の方がより有益であることを示している。

そして、第3のプレイ時間の評価値は

$$\text{評価}_3 = \begin{cases} \frac{\text{残った対戦日数}}{\text{与えられた対戦日数}} \times 100 & \text{勝った場合} \\ \frac{\text{使った対戦日数}}{\text{与えられた対戦日数}} \times 100 & \text{負けた場合} \end{cases}$$

で与えられる。これは早々に敗北したゲームより長くプレイした末に敗北したゲームの方がより有益であり、時間をかけて勝利したゲームより素早く勝利したゲームの方がより有益であることを表している。

各世代ではフィットネス値の最高値が記録され、プログラムの実行結果を評価するために使用される。

## 4.2 通常設定のゲーム難易度によるプレーヤ意思決定モデルの進化

最初の評価試行用ゲームシーケンスでは収束しなかった。その後、進化時の戦略を変更したところ、図4に見られるように10万世代後にGAでは軽度の改善が見られたが、DEでは改善が見られなかった。

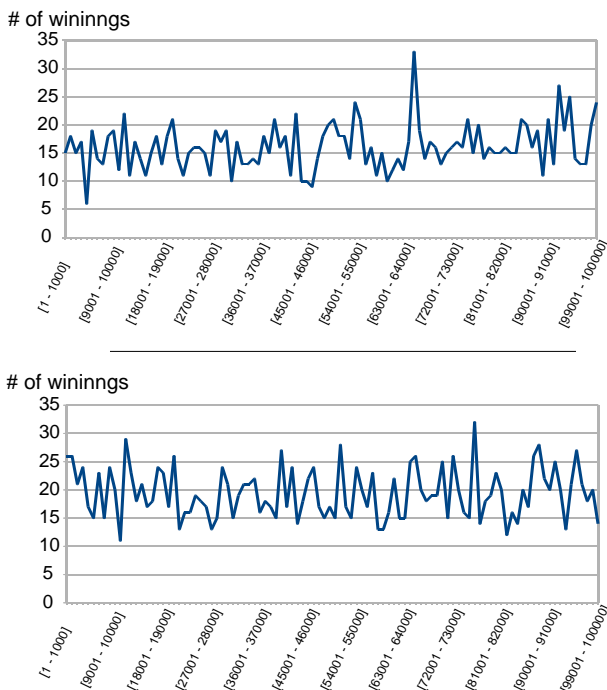


図4: 1000世代毎の勝利数。上段はGA, 下段はDE。横軸は世代数。

Star Trekゲームの標準設定時の敵の数を調べたところ、正規性乱数で生成されていた(図5参照)。これは時に敵の総数がとても少なくなり、時にとても多くなることを意味している。このような不規則な現象が進

化を阻む要因の1つである。例えば、3機の敵を倒すために導き出したプレーヤ意思決定モデルのルールは30機の敵を倒す場合には有効ではなかった、ということが進化の阻害要因として考えられる。同時に中央値は17機であるため、プレーヤ意思決定モデルが人並みに進化した暁には、17機の敵を倒すことが可能であり、勝率が50%であることが求められる。

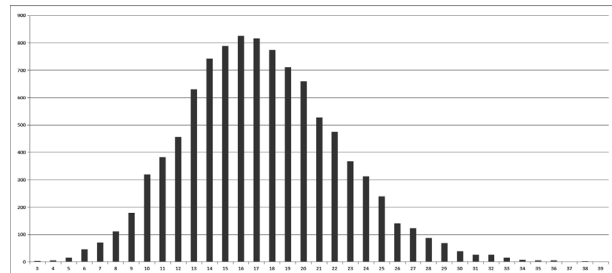


図5: 標準設定のゲームを10,000回実行した時の敵機数の度数分布。category of final peaks category of final peaks 平均17機, 最小/最大=3機/39機。

以上のことから、プレーヤ意思決定モデルを進化させる新しい戦略が明らかになった。すなわち、敵機数を操作しながら進化させるのである。3機の敵を倒すために導き出したプレーヤ意思決定モデルの戦略ルールは、30機のゲームに適用させながら進化させるより、むしろ4機の敵を倒す手段として有益であり、4機の敵を倒すようにプレーヤ意思決定モデルの戦略ルールを最適化することによって、今度は5機の敵を倒すために有用な戦略ルールを得る、という逐次進化、あるいは、共進化に通じる思想に基づく進化手法である。

## 4.3 共進化的思想に基づくプレーヤ意思決定モデルの進化

新たな進化方法はプレーヤ意思決定モデルとゲームの双方を世代ごとに強化していく共進化の手法に共通した思想に基づいている。実験では、敵の数を1~3機、対戦日数を118~128日の範囲で初期設定する。各世代では、図6のルールに基づいてゲームの難易度を増やす。プレーヤ意思決定モデルがStar Trekゲームに勝つと連続勝利数を1つ増やし、連続勝利数が10になると、敵の数を増加させるかあるいは対戦日数を減少させることでゲームの難度を上げる。勝利数を0にリセットして再度ゲームと対戦を続ける。

GAとDEのいずれでプレーヤ意思決定モデルを進化させた実験結果でも、簡単な対戦条件ではすばやく進化し、敵機数3~15機のStar Trekゲームと対戦で

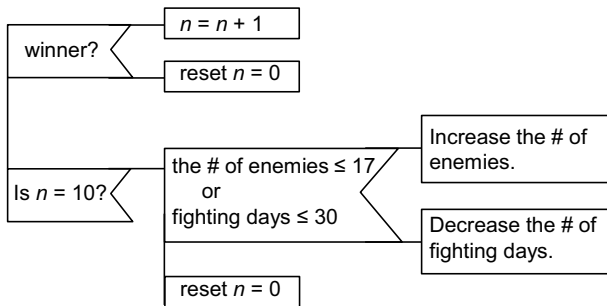


図 6: 連続 10 勝する毎に Star Trek ゲームの難易度を上げるルールのパラダイム図。n は累積勝利数。

きるまでに進化した。しかし、敵機数 17 機という通常設定のゲーム難易度で対戦できるまでは未だ進化しきれていない。図 7 は敵の数が少ないゲームにおいてプレーヤ意思決定モデルが簡単に勝利可能であるが、徐々に強くなるゲームにおいてはすぐにその限界に達していることを示している。しかし、第 4.2 節の通常設定のゲーム難易度で進化させるという戦略では勝率が 3.5% 以上に達することはできなかったが、本節の難易度を徐々に上げていく進化の戦略では、1000 世代毎の勝率を 20~50% にすることができた。なお、図 7 での勝利数が増えていないように見えるが、世代が上がり勝利数が増えるにつれてゲーム側の難易度を上げていることに注意して欲しい。

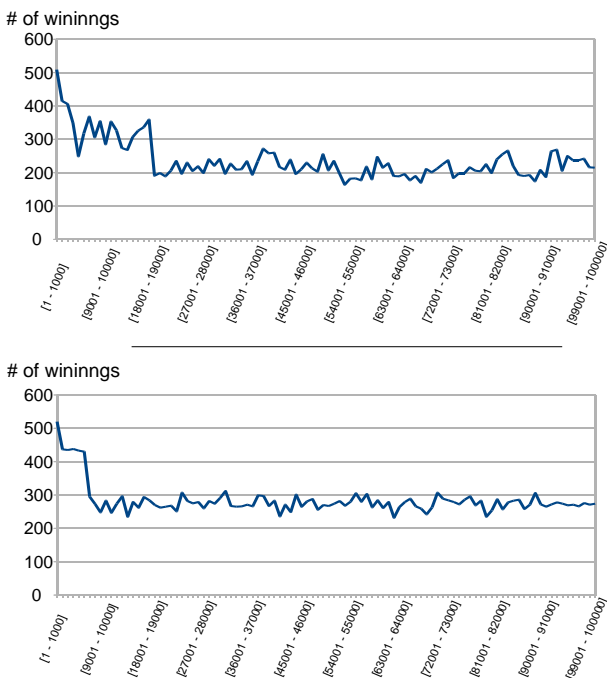


図 7: 1000 世代毎の勝利数。上段は GA，下段は DE。横軸は世代数。

## 5 考察

人間にとっても標準難易度設定の Star Trek ゲームは結構難しく、ゲームのランダム性がゲーム攻略を難しくしている。難易度の指標にはいろいろなパラメータが考えられる。敵機の合計数、ゲーム終了時に残った対戦日数、宇宙基地の数、星の数（エンタープライズ号の移動が遅く成り得るし、障害物となって光子魚雷が使えない場合もある）、ランダムに生じるエンタープライズ号のダメージ（エンジン故障、探索不能、兵器使用不可など）などである。これらによって非常に多くの異なるゲーム状況が生成されるため、ゲームプレーヤ意思決定モデルは、いかなる状況にも対処出来るよう完璧にすべきか、あるいは、モデルを進化させて最適化しやすいようにまったく自由にするか、のいずれかにならざるを得ない。

ゲームの複雑さとランダム性のために、Star Trek ゲームのプレーヤ意思決定モデルを進化させることは難しい。第 4.2 節のようにゲームの難易度を標準設定にした場合、探索解に大きなばらつきが生じ、一般解の生成が遅くなる。実際、10 万世代後でも適応度の収束は定常状態にならず、GA と DE では安定した解が探索できなかった。さらに世代を続けても多分同じ結果であったらうと思われる。このばらつきが非常に顕著であったため、進化計算はどのルールパラメータも安定化させることができなかった。

一方、共進化に似た戦略を取ってゲームを徐々に強くしていくと、ある敵機・対戦日数条件までは、勝てるプレーヤ意思決定モデルを獲得できるようになった。しかし、プレーヤ意思決定モデルの進化は限界に達してしまい、DE と GA では、敵機数 12 機と 13 機と共に大体 118 ゲーム日でこの状況に達してしまった（標準設定のゲーム難易度の場合、敵機数約 17、ゲーム日数 30 日以内）。

プレーヤ意思決定モデルのルールは、進化開始約 2 千世代で敵機数 10~11 機まで難易度を上げて勝つように進化し、2 万世代以降はその数が 13~14 機になったが、それ以降は 10 万世代まで続けても改善が見られなかった。もっと世代を重ねれば更に敵機を 1 機程度増やしても勝てるように多少進化したかもしれないが、敵機数  $x$  が  $x+1$  になるともゲーム難易度は本質的には些細なことであり、ゲーム状況を大きく変えるものではない。それよりも、今ここで重要なことは、敵機 14 機のゲーム難易度からプレーヤ意思決定モデルを進化させることが際立って難しいことなのか、それとも、我々の設計したゲームプレーヤ意思決定モデルに何らかの問題があるのかの、いずれであるかを明らかにすることである。

獲得したルールパラメータの値を観察すると、ゲームに勝った 2 つの世代のプレーヤ意思決定モデルパラ

メータ値間に相関が見られず、これらが安定している。これは最適解にはまだ辿りついていないと思われ、累積勝利数に基づいて共進化的な戦略で徐々にゲーム難易度を上げていく方法が本当によいのか、まだ不明である。

評価に用いた一連のゲームシーケンス（対戦させる複数ゲームを一セットとしたもの）は乱数の種をリセットすることで同じゲームシーケンスに対して進化させて解を見つけることができる、すなわち繰り返し可能な静的実験条件である。16種の乱数の種（ゲーム）を用意し、DEで10万世代進化させた結果を表2に示す。この実験での最優良個体は敵機9機に勝てるようになっただけである。全体の1/4は改善が見られず敵機数1~3機の初期状態のままである。最も簡単な条件でもいくつかのゲームシーケンスでは最適化できなかったということは、ゲームプレーヤ意思決定モデルが未だ弱く難があることを意味している。

表 2: 16 種類の乱数種で生成した Star Trek ゲームのシーケンスと DE で 10 万世代まで進化させた時、初期状態（敵機数 1~3 機）から難易度を上げるために増やした敵機数。0 は初期状態のゲーム難易度のままで進化できなかったことを意味する。\*は数日 PC で実行しても 10 万世代まで行き着かなかった実験を意味する。

Seed	1	2	3	4	5	6	7	8
Added	6	4	0	3	2	0	6	0

Seed	9	10	11	12	13	14	15	16
Added	4	*	1	*	1	9	8	0

## 6 結論

Star Trek ゲームに対戦する人間プレーヤの意思決定モデルを進化させる枠組み作りは困難であるが可能である、と結論付けられよう。我々が設計したプレーヤ意思決定モデルのルール群は一定レベルまでは進化できることを示した。静的な対戦ゲーム条件での実験で進化が可能であり、乱数アルゴリズムの種が制御できるので、結果を歪めるような変動はない。

しかし、進化の限界は未だ明らかではない。我々のプレーヤ意思決定モデルとそのルール群はまだ改良が必要であり、標準設定のゲーム難易度である敵機数 17 機でも勝てるようなルールには達していない。ルール群の改良をして完全にする必要はある。現在のルールパラメータではまだ十分ではないので、すべての対戦行動が実現できているわけではない。例えば、残存エネルギーが少なくなっても敵機が最後の 1 機ならば宇

宙基地に補給に帰らずに攻撃してゲームを終了させるという選択があり得るが、あるファジールールのパラメータでは、まだこのような行動が取り得ていない。引き続き、モデルルールを見直し拡張して、プレーヤ意思決定モデルの意思決定をもっと制御するように進化するようにしなければならない。

第 1 節で述べたように、今後は、ゲームプレーヤ意思決定モデル側のゲーム戦略とゲームプログラム側の戦略との間に本当の共進化を適用していきたい。その時には、ゲームプレーヤ意思決定モデルの観点から、今回の共進化的戦略とどのように進化効率・性能が異なるかを比較することが楽しみである。

## 謝辞

本研究は科学研究費（課題番号 23500279）の助成を受けたものである。

## 参考文献

- [1] デジタルゲームの教科書作成委員会「デジタルゲームの教科書 知っておくべきゲーム業界最新トレンド」ソフトバンククリエイティブ（2010年5月）。
- [2] Karr C., Freeman L., and Meredith D., “Improved Fuzzy Process Control of Spacecraft Autonomous Rendez-vous Using a Genetic Algorithm,” SPIE Conf. on Intelligent Control and Adaptive Systems, Philadelphia, PA, UAS, pp.274–283 (Nov., 1989).
- [3] Majumdar K. K. and Majumder D. D., “Fuzzy knowledge-based and model-based systems,” J. of Intelligent & Fuzzy Systems, vol.18, no.4, pp391–403 (2007).
- [4] Shill P. C., Akhand M. A. H., Das’ S. R., and Paul A., “Application of evolutionary algorithm in optimizing the fuzzy rule base for nonlinear system modeling and control,” 2010 Int. Conf. on Computer and Communication Engineering (ICCCE2010), Kuala Lumpur, Malaysia, pp.1–6 (May, 2010).
- [5] Star Trek: <http://www.dunnington.u-net.com/public/startrek/> (2005年3月)。