

## 論 文

## ノンプレイヤーキャラクタを伴うロールプレイングゲームの攻略法に関する一考察

前田 康成<sup>†</sup>      後藤文太郎<sup>†</sup>      升井 洋志<sup>†</sup>      梶井 文人<sup>†</sup>  
 鈴木 正清<sup>†</sup>      松嶋 敏泰<sup>††</sup>

## A Note on A Strategy for Role-Playing Game with Nonplayer Characters

Yasunari MAEDA<sup>†</sup>, Fumitaro GOTO<sup>†</sup>, Hiroshi MASUI<sup>†</sup>, Fumito MASUI<sup>†</sup>,  
 Masakiyo SUZUKI<sup>†</sup>, and Toshiyasu MATSUSHIMA<sup>††</sup>

あらまし 近年、遊び手である人間の負荷を軽減することを目的に、遊び手が操作するキャラクタであるプレイヤーキャラクタ (PC) 以外にコンピュータが操作するキャラクタであるノンプレイヤーキャラクタ (NPC) が導入されたロールプレイングゲーム (RPG) が増えてきた。従来からマルコフ決定過程 (MDP) を用いた RPG のモデル化が行われているが、NPC を伴う RPG の MDP を用いたモデル化はまだ行われていない。そこで、本研究では、MDP を用いて NPC を伴う RPG のモデル化を行う。更に、MDP の真のパラメータ未知の場合に相当する NPC を伴う RPG について、報酬の期待値をベイズ基準のもとで最大にする攻略法を算出するアルゴリズムを提案する。

キーワード ロールプレイングゲーム, ノンプレイヤーキャラクタ, マルコフ決定過程, 統計的決定理論, ベイズ基準

## 1. ま え が き

近年、遊び手である人間の負荷を軽減することを目的に、遊び手が操作するキャラクタであるプレイヤーキャラクタ (以下、PC と表記する) 以外にコンピュータが操作するキャラクタであるノンプレイヤーキャラクタ (以下、NPC と表記する) が導入されたロールプレイングゲーム (以下、RPG と表記する) が増えてきた [1]~[3]。

RPG にもいろいろな種類があるが、本研究では、マップモードにおいてマップ上の PC を移動させて、敵と遭遇すると戦闘モードになり、戦闘結果として敵を倒すと報酬 (お金、経験値などに相当) を得るような冒険型の RPG を対象とする。冒険型の RPG についても NPC が多く導入されている。NPC にもいろいろあるが、本研究ではマップモードでは PC の決定

(遊び手による行動選択) に従って PC と一緒に移動し、戦闘モードでは NPC 自身の決定 (コンピュータによる行動選択) に従って行動する NPC を想定する。

NPC を伴う RPG の従来研究 [1]~[3] では、開発者側の視点から NPC の設計 (攻撃重視, 安全重視など) や、NPC の行動選択の仕方に関する研究が行われている。しかし、人間の遊び手側の視点による、NPC を伴う RPG の攻略法 (本研究では、PC の行動選択と NPC の種類の選択 (当該時点でどの NPC を仲間にするか) の仕方に相当) に関する研究は行われていない。

NPC を伴わない RPG については、従来からマルコフ決定過程 (以下、MDP と表記する) [4]~[6] を用いたモデル化 [7]~[9] が行われており、従来研究 [9] では遊び手が PC を操作することによって得る報酬の期待値を統計的決定理論 [10], [11] に基づいてベイズ基準のもとで最大化するという点で最適な攻略法 (PC の行動選択の仕方) を算出するアルゴリズムを提案している。

そこで、本研究では、従来研究 [9] のモデルを拡張することによって、MDP を用いて NPC を伴う RPG

<sup>†</sup> 北見工業大学情報システム工学科, 北見市  
 Dept. of Computer Sciences, Kitami Institute of Technology,  
 165 Koen-cho, Kitami-shi, 090-8507 Japan

<sup>††</sup> 早稲田大学応用数理学科, 東京都  
 Department of Applied Mathematics, Waseda University,  
 Tokyo, 169-8555 Japan

のモデル化を行う。その上で、MDPの真のパラメータ未知の場合に相当するNPCを伴うRPGについて、遊び手が得る報酬の期待値を統計的決定理論に基づいてベイズ基準のもとで最大にするという点で最適な攻略法を算出するアルゴリズムを提案する。

従来研究[7]~[9]では、RPGをMDPを用いてモデル化することの目的として、RPGを遊び手がプレイする際に楽しいと感じる要素を工学的に把握することや、RPGの自動開発などが挙げられている。これらの従来研究は基礎研究であり、RPGの自動開発などは直近の目的ではなく、将来的な目的である。本研究も従来研究と同様に基礎研究であり、現時点で実際のRPGの開発現場に貢献することはできないが、将来的には実際の開発現場におけるコスト軽減などに貢献できる可能性がある。

例えば、RPGの開発現場では多数の被験者によるプレイを通して観測される事象を、データ収集することがある。本研究で算出されるような攻略法を組み込んだエージェントをバーチャルの遊び手としてシミュレーションを実施することにより、被験者によるコストを軽減できる可能性がある。

また、このようなエージェントが選択する行動を実際の人間の遊び手に例示することにより、初心者への支援機能という応用も考えられる。

## 2. MDPを用いたNPCを伴うRPGのモデル化

### 2.1 本研究で研究対象とするNPCを伴うRPG

以下で、本研究で研究対象とするNPCを伴うRPGについて説明する。なお、本研究で扱うNPCを伴うRPGは、従来研究[9]で扱っているRPGにNPCを追加して拡張したものに相当する。

PCはヒットポイント（以下、HPと表記する）と呼ばれる数値をもち、HPが0になると次の期にマップ上のスタート位置から再開する。再開時にはHPはスタート時と同じ最大値 $M_{hp}$ まで回復する。NPCはPCとともに行動するが、NPCはHPをもたず、敵から攻撃を受けた際のHPによる管理はPCのみとする。

$sm_i$ はマップ上の位置を示し、 $SM$ 、 $SM = \{sm_1, sm_2, \dots, sm_{|SM|}\}$ はマップ上の位置の集合である。ゲーム開始時のスタート位置を $sm_1$ とする。また、本研究では、スタート位置や現在のPCの位置は既知とする。NPCの位置はPCと同じとする。 $f_i$ はマップ上の地形の種類を示し、 $F$ 、 $F = \{f_1, f_2, \dots, f_{|F|}\}$

はマップ上の地形の種類の種類集合である。マップ上の各位置がどの地形に該当するかは、関数 $F(sm_i) \in F$ で分かる。

$e_i$ は敵の種類を示し、 $E$ 、 $E = \{e_1, e_2, \dots, e_{|E|}\}$ は敵の種類の種類集合である。 $M(e_i)$ は敵 $e_i$ 出現時の敵 $e_i$ のHPを示す。PCまたはNPCが敵を攻撃することによって敵のHPを0以下にすると、その敵を倒し、その敵に該当する報酬 $G(e_i)$ を得る。

$h_i$ はNPCの種類を示し、 $H$ 、 $H = \{h_1, h_2, \dots, h_{|H|}\}$ はNPCの種類の種類集合である。

PC（遊び手）が選択できる行動（コマンド）はマップモードと戦闘モードで異なり、マップモードでは $a_1$ から $a_4$ が選択可能で、戦闘モードでは $a_5$ と $a_6$ が選択可能である。また、戦闘モードでは遊び手はNPCの選択（NPCの種類指定）も行う。 $a_1$ 、 $a_2$ 、 $a_3$ 、 $a_4$ はそれぞれマップ上で右、左、上、下に移動するための行動である。 $mv(sm_i, a_j)$ はPCが位置 $sm_i$ で行動 $a_j$ を選択した際の移動先の位置である。PCの移動に際して、確率 $p(e_k|F(mv(sm_i, a_j)), \theta^*)$ で移動先 $mv(sm_i, a_j)$ に敵 $e_k$ が出現し戦闘モードになる。敵は同時に複数出現することはなく、確率 $1 - \sum_{e_k \in E} p(e_k|F(mv(sm_i, a_j)), \theta^*)$ で何も出現せずにマップモードが続く。

戦闘モードでは、遊び手はPCの行動選択とNPCの選択を行う。行動 $a_5$ は敵と戦うコマンドである。PCの行動として行動 $a_5$ が選択されると、確率 $p(C_1(e_i)|a_5, e_i, \theta^*)$ で敵 $e_i$ への攻撃に成功し、敵 $e_i$ のHPが $C_1(e_i)$ 減少する。PCは確率 $1 - p(C_1(e_i)|a_5, e_i, \theta^*)$ で敵 $e_i$ への攻撃に失敗する。行動 $a_6$ は敵から逃げるための行動で、確率 $p(map|a_6, \theta^*)$ で戦闘モードが終了し、次の期はマップモードになる。確率 $1 - p(map|a_6, \theta^*)$ でそのまま戦闘モードが続く。

PCの行動実施後に、NPCの行動選択が実施される。NPC $h_i$ は確率 $p(a_5|h_i, \psi^*)$ で攻撃行動 $a_5$ を選択し、確率 $p(a_7|h_i, \psi^*) = 1 - p(a_5|h_i, \psi^*)$ でPCを回復させるための行動 $a_7$ を選択する。NPCの攻撃行動 $a_5$ は確率1で成功して、敵 $e_j$ のHPが $C_2(h_i, e_j)$ 減少する。NPCの回復行動 $a_7$ は確率1で成功して、PCのHPを $M_{hp}$ に回復する。

NPCの行動実施後に敵の攻撃が実施される。確率 $p(B(e_i)|e_i, \theta^*)$ で敵 $e_i$ がPCへの攻撃に成功し、PCのHPが $B(e_i)$ 減少する。敵 $e_i$ は確率 $1 - p(B(e_i)|e_i, \theta^*)$ でPCへの攻撃に失敗する。

戦闘モードにおいて、PCの行動（戦う、または逃

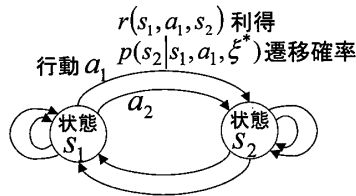


図1 MDP の例  
Fig. 1 An example of MDP.

げる), NPC の行動 (戦う, または回復), 敵の行動 (常に戦う) が実施される順番は, PC, NPC, 敵の順番である. また, PC の行動が実施されるのは, 遊び手による NPC の選択後である.  $\theta^*, \theta^* \in \Theta$  及び  $\psi^*, \psi^* \in \Psi$  は各種確率分布を支配する真のパラメータで未知とする.  $\Theta$  及び  $\Psi$  は連続のパラメータ集合である.

2.2 MDP と NPC を伴う RPG の対応

最初に一般的な MDP の概要について説明する.

MDP は, 状態  $s_i, s_i \in S, S = \{s_1, s_2, \dots, s_{|S|}\}$  ( $|S|$  は有限), 各状態で選択できる行動  $a_i, a_i \in A, A = \{a_1, a_2, \dots, a_{|A|}\}$  ( $|A|$  は有限), 状態  $s_i$  で行動  $a_j$  を選択したもとで状態  $s_k$  へ遷移する遷移確率  $p(s_k | s_i, a_j, \xi^*)$  ( $\xi^*$  は遷移確率分布を支配する真のパラメータ), 遷移に伴い発生する利得  $r(s_i, a_j, s_k)$  で構成される. 図 1 に状態数と行動数が 2 の MDP の例を示す. 状態  $s_1$  から 4 本の矢印が出ているが, これらは行動  $a_1$  を選択した場合の遷移確率  $p(s_1 | s_1, a_1, \xi^*)$  による状態  $s_1$  への遷移と遷移確率  $p(s_2 | s_1, a_1, \xi^*)$  による状態  $s_2$  への遷移と, 行動  $a_2$  を選択した場合の遷移確率  $p(s_1 | s_1, a_2, \xi^*)$  による状態  $s_1$  への遷移と遷移確率  $p(s_2 | s_1, a_2, \xi^*)$  による状態  $s_2$  への遷移の計 4 パターンの状態遷移を表現している. MDP を確率モデルとして採用した問題設定では, 行動を選び, 状態が遷移し, 利得を得るという一連のプロセスを繰り返しながら総利得を最大化することを目的とする場合が多い. プロセスの繰り返し回数が有限の場合には, 総利得の期待値 (期待総利得) を最大化する最適な決定関数 (選択する行動を決める関数) を動的計画法 (以下, DP と表記する) によって算出できる. 具体的には真のパラメータ  $\xi^*$  既知の場合であれば, 式 (1) を用いて,  $t$  期の状態が  $s_i$  という条件下における  $t$  期以降の期待総利得の最大値  $V(s_i, t)$  を逐次的に計算できる.

$$V(s_i, t) = \max_{a_j \in A} \sum_{s_k \in S} p(s_k | s_i, a_j, \xi^*) (r(s_i, a_j, s_k) + V(s_k, t + 1)). \quad (1)$$

表 1 マップモードの状態遷移  
Table 1 State transitions in map mode.

条件	遷移確率	状態 $x_{t+1}$
式 (3)	1	式 (5)
式 (4)	$p(e_i   F(mv(x_{t,2}, y_{t,1})), \theta^*)$	式 (6)
式 (4)	$1 - \sum_{e_i \in E} p(e_i   F(mv(x_{t,2}, y_{t,1})), \theta^*)$	式 (7)

次に, MDP と前節で説明した NPC を伴う RPG の対応について説明する. なお, MDP との対応についても NPC を追加して拡張した点以外は従来研究 [9] と同様である.

$x_t$  は MDP における  $t$  期の状態を示す変数で, 次式のように構成される.

$$x_t = (x_{t,1}, x_{t,2}, x_{t,3}, x_{t,4}), \quad (2)$$

ただし,  $x_{t,1}$  は  $t$  期における PC の HP,  $x_{t,2}$  は  $t$  期における PC のマップ上での位置,  $x_{t,3}$  は  $t$  期における敵の種類,  $x_{t,4}$  は  $t$  期における敵の HP を示し, マップモードの場合には敵は存在せず  $x_{t,3} = x_{t,4} = 0$  とする.

$A_{pc}(x_t)$  は状態  $x_t$  において PC が選択可能な MDP の行動集合を示し, 具体的には前述の  $a_1, a_2, a_3, a_4, a_5, a_6$  が要素となる.  $y_t$  は MDP における  $t$  期に選択した行動を示す変数である. 行動  $y_t$  はマップモードと戦闘モードで形式が異なる. マップモードでは,  $y_t = y_{t,1}$  で,  $y_{t,1}$  は PC の行動を示す変数である. 戦闘モードでは,  $y_t = (y_{t,1}, y_{t,2}, y_{t,3})$  で,  $y_{t,1}$  は PC の行動,  $y_{t,2} \in H$  は遊び手が選択した NPC の種類,  $y_{t,3} \in \{a_5, a_7\} = A_{npc}$  は NPC の行動を示す変数である.

次に  $t$  期の状態  $x_t$  で行動  $y_t$  を選択したときの状態遷移について説明する. マップモードにおける状態遷移 ( $a_1, a_2, a_3, a_4$  を選択したとき) を表 1 に示す. 戦闘モードにおいて PC の行動として  $a_5$  (戦う), NPC の種類として  $y_{t,2}$  を選択したときの状態遷移を表 2 に示す. 戦闘モードにおいて PC の行動として  $a_6$  (逃げる), NPC の種類として  $y_{t,2}$  を選択したときの状態遷移を表 3 に示す.

$$mv(x_{t,2}, y_{t,1}) = sm_1. \quad (3)$$

$$mv(x_{t,2}, y_{t,1}) \neq sm_1. \quad (4)$$

$$x_{t+1} = (M_{hp}, sm_1, 0, 0). \quad (5)$$

$$x_{t+1} = (x_{t,1}, mv(x_{t,2}, y_{t,1}), e_i, M(e_i)). \quad (6)$$

表 2 戦闘モードで PC の行動  $a_5$  と NPC  $y_{t,2}$  が選択されたときの状態遷移

Table 2 State transitions in battle mode when PC's action  $a_5$  and NPC  $y_{t,2}$  are selected.

遷移確率	状態 $x_{t+1}$
$P_1$	式 (9), $x_{t,4} \leq C_2(y_{t,2}, x_{t,3})$ 式 (10), $x_{t,4} > C_2(y_{t,2}, x_{t,3})$
$P_2$	式 (12)
$P_3$	式 (9), $x_{t,4} \leq C_2(y_{t,2}, x_{t,3})$ 式 (14), $x_{t,4} > C_2(y_{t,2}, x_{t,3})$ かつ $x_{t,1} > B(x_{t,3})$ 式 (5), $x_{t,4} > C_2(y_{t,2}, x_{t,3})$ かつ $x_{t,1} \leq B(x_{t,3})$
$P_4$	式 (16), $M_{hp} > B(x_{t,3})$ 式 (5), $M_{hp} \leq B(x_{t,3})$
$P_5$	式 (18), $x_{t,4} > C_1(x_{t,3}) + C_2(y_{t,2}, x_{t,3})$ 式 (9), $x_{t,4} \leq C_1(x_{t,3}) + C_2(y_{t,2}, x_{t,3})$
$P_6$	式 (20), $x_{t,4} > C_1(x_{t,3})$ 式 (9), $x_{t,4} \leq C_1(x_{t,3})$
$P_7$	式 (9), $x_{t,4} \leq C_1(x_{t,3}) + C_2(y_{t,2}, x_{t,3})$ 式 (22), $x_{t,4} > C_1(x_{t,3}) + C_2(y_{t,2}, x_{t,3})$ かつ $x_{t,1} > B(x_{t,3})$ 式 (5), $x_{t,4} > C_1(x_{t,3}) + C_2(y_{t,2}, x_{t,3})$ かつ $x_{t,1} \leq B(x_{t,3})$
$P_8$	式 (9), $x_{t,4} \leq C_1(x_{t,3})$ 式 (24), $x_{t,4} > C_1(x_{t,3})$ かつ $M_{hp} > B(x_{t,3})$ 式 (5), $x_{t,4} > C_1(x_{t,3})$ かつ $M_{hp} \leq B(x_{t,3})$

表 3 戦闘モードで PC の行動  $a_6$  と NPC  $y_{t,2}$  が選択されたときの状態遷移

Table 3 State transitions in battle mode when PC's action  $a_6$  and NPC  $y_{t,2}$  are selected.

遷移確率	状態 $x_{t+1}$
$P_9$	式 (9)
$P_{10}$	式 (10), $x_{t,4} > C_2(y_{t,2}, x_{t,3})$ 式 (9), $x_{t,4} \leq C_2(y_{t,2}, x_{t,3})$
$P_{11}$	式 (12)
$P_{12}$	式 (9), $x_{t,4} \leq C_2(y_{t,2}, x_{t,3})$ 式 (14), $x_{t,4} > C_2(y_{t,2}, x_{t,3})$ かつ $x_{t,1} > B(x_{t,3})$ 式 (5), $x_{t,4} > C_2(y_{t,2}, x_{t,3})$ かつ $x_{t,1} \leq B(x_{t,3})$
$P_{13}$	式 (16), $M_{hp} > B(x_{t,3})$ 式 (5), $M_{hp} \leq B(x_{t,3})$

$$x_{t+1} = (x_{t,1}, mv(x_{t,2}, y_{t,1}), x_{t,3}, x_{t,4}). \quad (7)$$

$$P_1 = (1 - p(C_1(x_{t,3}) | a_5, x_{t,3}, \theta^*)) p(a_5 | y_{t,2}, \psi^*) (1 - p(B(x_{t,3}) | x_{t,3}, \theta^*)). \quad (8)$$

$$x_{t+1} = (x_{t,1}, x_{t,2}, 0, 0). \quad (9)$$

$$x_{t+1} = (x_{t,1}, x_{t,2}, x_{t,3}, x_{t,4} - C_2(y_{t,2}, x_{t,3})). \quad (10)$$

$$P_2 = (1 - p(C_1(x_{t,3}) | a_5, x_{t,3}, \theta^*)) p(a_7 | y_{t,2}, \psi^*) (1 - p(B(x_{t,3}) | x_{t,3}, \theta^*)). \quad (11)$$

$$x_{t+1} = (M_{hp}, x_{t,2}, x_{t,3}, x_{t,4}). \quad (12)$$

$$P_3 = (1 - p(C_1(x_{t,3}) | a_5, x_{t,3}, \theta^*)) p(a_5 | y_{t,2}, \psi^*)$$

$$p(B(x_{t,3}) | x_{t,3}, \theta^*). \quad (13)$$

$$x_{t+1} = (x_{t,1} - B(x_{t,3}), x_{t,2}, x_{t,3}, x_{t,4} - C_2(y_{t,2}, x_{t,3})). \quad (14)$$

$$P_4 = (1 - p(C_1(x_{t,3}) | a_5, x_{t,3}, \theta^*)) p(a_7 | y_{t,2}, \psi^*) p(B(x_{t,3}) | x_{t,3}, \theta^*). \quad (15)$$

$$x_{t+1} = (M_{hp} - B(x_{t,3}), x_{t,2}, x_{t,3}, x_{t,4}). \quad (16)$$

$$P_5 = p(C_1(x_{t,3}) | a_5, x_{t,3}, \theta^*) p(a_5 | y_{t,2}, \psi^*) (1 - p(B(x_{t,3}) | x_{t,3}, \theta^*)). \quad (17)$$

$$x_{t+1} = (x_{t,1}, x_{t,2}, x_{t,3}, x_{t,4} - C_1(x_{t,3}) - C_2(y_{t,2}, x_{t,3})). \quad (18)$$

$$P_6 = p(C_1(x_{t,3}) | a_5, x_{t,3}, \theta^*) p(a_7 | y_{t,2}, \psi^*) (1 - p(B(x_{t,3}) | x_{t,3}, \theta^*)). \quad (19)$$

$$x_{t+1} = (M_{hp}, x_{t,2}, x_{t,3}, x_{t,4} - C_1(x_{t,3})). \quad (20)$$

$$P_7 = p(C_1(x_{t,3}) | a_5, x_{t,3}, \theta^*) p(a_5 | y_{t,2}, \psi^*) p(B(x_{t,3}) | x_{t,3}, \theta^*). \quad (21)$$

$$x_{t+1} = (x_{t,1} - B(x_{t,3}), x_{t,2}, x_{t,3}, x_{t,4} - C_1(x_{t,3}) - C_2(y_{t,2}, x_{t,3})). \quad (22)$$

$$P_8 = p(C_1(x_{t,3}) | a_5, x_{t,3}, \theta^*) p(a_7 | y_{t,2}, \psi^*) p(B(x_{t,3}) | x_{t,3}, \theta^*). \quad (23)$$

$$x_{t+1} = (M_{hp} - B(x_{t,3}), x_{t,2}, x_{t,3}, x_{t,4} - C_1(x_{t,3})). \quad (24)$$

$$P_9 = p(map | a_6, \theta^*). \quad (25)$$

$$P_{10} = (1 - p(map | a_6, \theta^*)) p(a_5 | y_{t,2}, \psi^*) (1 - p(B(x_{t,3}) | x_{t,3}, \theta^*)). \quad (26)$$

$$P_{11} = (1 - p(map | a_6, \theta^*)) p(a_7 | y_{t,2}, \psi^*) (1 - p(B(x_{t,3}) | x_{t,3}, \theta^*)). \quad (27)$$

$$P_{12} = (1 - p(map | a_6, \theta^*)) p(a_5 | y_{t,2}, \psi^*) p(B(x_{t,3}) | x_{t,3}, \theta^*). \quad (28)$$

$$P_{13} = (1 - p(map | a_6, \theta^*)) p(a_7 | y_{t,2}, \psi^*) p(B(x_{t,3}) | x_{t,3}, \theta^*). \quad (29)$$

表 2 で状態  $x_{t+1}$  が式 (9) で表現される状態遷移は, PC または NPC の攻撃によって敵  $x_{t,3}$  の HP が 0 以下になり, PC または NPC が敵を倒した遷移である. 表 3 で PC が逃げることに失敗したもとで状態  $x_{t+1}$  が式 (9) で表現される状態遷移も同様に NPC が敵を倒した遷移である. 表 2 及び表 3 で状態  $x_{t+1}$  が式 (5) で表現される状態遷移は, PC の HP が 0 以下になり, PC が倒されて初期状態  $x_1 = (M_{hp}, sm_1, 0, 0)$  と同じ状態に戻された遷移である. PC または NPC が敵  $x_{t,3}$  を倒した状態遷移に伴う利得は,  $r(x_t, y_t, x_{t+1}) = G(x_{t,3})$  である. その他の状態遷移に伴う利得は,  $r(x_t, y_t, x_{t+1}) = 0$  である. 各期の状態は観測可能とする. また, PC や NPC や敵の攻撃力  $C_1(e_i)$ ,  $C_2(h_i, e_j)$ ,  $B(e_i)$  及び敵を倒したときの報酬  $G(e_i)$  などは全て既知とし, 各種確率分布を支配する真のパラメータ  $\theta^*$ ,  $\psi^*$  のみ未知とする. このもとで,  $T$  期間のプレイを行って総利得  $\sum_{t=1}^T r(x_t, y_t, x_{t+1})$  の最大化を目的とする.

### 3. 定式化と提案アルゴリズム

統計的決定理論 [10], [11] に基づいて定式化し, 有限の  $T$  期間のプレイによる総利得をベイズ基準のもとで最大化するという点で最適な PC の行動選択及び NPC の選択の仕方 (決定関数) を求めるアルゴリズムを提案する. ベイズ最適な決定関数が NPC を伴う RPG の攻略法に相当する.

#### 3.1 統計的決定理論に基づく定式化

最初に効用関数を次式で定義する.

$$U(d_{pc}(\cdot, \cdot, \cdot), d_{npc}(\cdot, \psi^*), x^{T+1}y^T, \theta^*, \psi^*) = \sum_{t=1}^T r(x_t, y_t, x_{t+1}), \quad (30)$$

ただし,  $d_{pc}(\cdot, \cdot, \cdot)$  は当該の期の状態  $x_t$  と, 当該の期までの系列  $x^t y^{t-1} = x_1 y_1 \cdots x_{t-1} y_{t-1} x_t$ , 当該の期  $t$  (当該の期を示す整数) を受け取って, マップモードには PC の行動  $y_{t,1}$ , 戦闘モードには PC の行動  $y_{t,1}$  と NPC の種類  $y_{t,2}$  を返す決定関数を示す.  $d_{npc}(\cdot, \psi^*)$  は戦闘モードにおいて,  $d_{pc}(\cdot, \cdot, \cdot)$  によって選択された NPC の種類  $y_{t,2}$  を受け取って, 確率  $p(y_{t,3} | y_{t,2}, \psi^*)$  に従って NPC の行動  $y_{t,3}$  を返す決定関数を示す. 本研究では, 2 種類の決定関数が存在するが, 決定関数  $d_{npc}(\cdot, \psi^*)$  は未知パラメータ  $\psi^*$  によって定まる決定関数で固定である. 本研究では, なるべく総利得を大きくするような決定関数  $d_{pc}(\cdot, \cdot, \cdot)$  を探索する

問題を扱っている. ただし, 前述のとおり, 決定関数  $d_{npc}(\cdot, \psi^*)$  の引数である NPC の種類  $y_{t,2}$  を決定関数  $d_{pc}(\cdot, \cdot, \cdot)$  が決定するので, 決定関数  $d_{pc}(\cdot, \cdot, \cdot)$  が決定関数  $d_{npc}(\cdot, \psi^*)$  を制御する仕組みになっている. 式 (30) による効用関数は真のパラメータ  $\theta^*$ ,  $\psi^*$  のもとで, 決定関数  $d_{pc}(\cdot, \cdot, \cdot)$  と  $d_{npc}(\cdot, \psi^*)$  を用いて  $x^{T+1}y^T$  と遷移した場合の総利得である.

次に期待効用  $EU(d_{pc}(\cdot, \cdot, \cdot), d_{npc}(\cdot, \psi^*), \theta^*, \psi^*)$  を次式で定義する.

$$\begin{aligned} & EU(d_{pc}(\cdot, \cdot, \cdot), d_{npc}(\cdot, \psi^*), \theta^*, \psi^*) \\ &= \sum_{x^{T+1}y^T} p(x^{T+1}y^T | d_{pc}(\cdot, \cdot, \cdot), d_{npc}(\cdot, \psi^*), \theta^*, \psi^*) \\ & \quad U(d_{pc}(\cdot, \cdot, \cdot), d_{npc}(\cdot, \psi^*), x^{T+1}y^T, \theta^*, \psi^*) \\ &= \sum_{x_2} \bar{p}_1(x_2) \left( r(x_1, y_1, x_2) + \right. \\ & \quad \sum_{x_3} \bar{p}_1(x_3) \left( r(x_2, y_2, x_3) + \right. \\ & \quad \cdots \sum_{x_T} \bar{p}_1(x_T) \left( r(x_{T-1}, y_{T-1}, x_T) + \right. \\ & \quad \left. \left. \left. \sum_{x_{T+1}} \bar{p}_1(x_{T+1}) \left( r(x_T, y_T, x_{T+1}) \right) \cdots \right) \right) \right), \quad (31) \end{aligned}$$

ただし,  $x_t$  がマップモードの状態の場合,

$$\bar{p}_1(x_{t+1}) = p(x_{t+1} | x_t, y_t, \theta^*), \quad (32)$$

$x_t$  が戦闘モードの状態の場合,

$$\bar{p}_1(x_{t+1}) = \sum_{y_{t,3}} p(y_{t,3} | y_{t,2}, \psi^*) p(x_{t+1} | x_t, y_t, \theta^*); \quad (33)$$

である. 式 (31) による期待効用は真のパラメータ  $\theta^*$ ,  $\psi^*$  のもとで, 決定関数  $d_{pc}(\cdot, \cdot, \cdot)$  と  $d_{npc}(\cdot, \psi^*)$  を用いた場合の総利得の期待値である. 式 (32) が表 1 の遷移確率に相当し, 式 (33) の  $p(y_{t,3} | y_{t,2}, \psi^*) p(x_{t+1} | x_t, y_t, \theta^*)$  が表 2 及び表 3 の遷移確率に相当する.

真のパラメータ  $\theta^*$ ,  $\psi^*$  が未知なので, パラメータ  $\theta$ ,  $\psi$  の事前分布  $p(\theta)$ ,  $p(\psi)$  を導入し, 事前分布は既知とする. 事前分布に対してパラメータ空間で期待効用の期待値をとるベイズ期待効用を次式で定義する.

$$\begin{aligned} & BEU(d_{pc}(\cdot, \cdot, \cdot), d_{npc}(\cdot, \psi), p(\theta), p(\psi)) \\ &= \int_{\theta \in \Theta} \int_{\psi \in \Psi} p(\theta) p(\psi) \\ & \quad EU(d_{pc}(\cdot, \cdot, \cdot), d_{npc}(\cdot, \psi), \theta, \psi) d\psi d\theta \\ &= \sum_{x_2} \bar{p}_2(x_2) \left( r(x_1, y_1, x_2) + \right. \end{aligned}$$

$$\begin{aligned} & \sum_{x_3} \bar{p}_2(x_3) \left( r(x_2, y_2, x_3) + \right. \\ & \cdots \sum_{x_T} \bar{p}_2(x_T) \left( r(x_{T-1}, y_{T-1}, x_T) + \right. \\ & \left. \left. \sum_{x_{T+1}} \bar{p}_2(x_{T+1}) \left( r(x_T, y_T, x_{T+1}) \right) \cdots \right) \right), \quad (34) \end{aligned}$$

ただし,  $x_t$  がマップモードの状態の場合,

$$\bar{p}_2(x_{t+1}) = \int_{\theta \in \Theta} p(\theta | x^t y^{t-1}) p(x_{t+1} | x_t, y_t, \theta) d\theta, \quad (35)$$

$x_t$  が戦闘モードの状態の場合,

$$\begin{aligned} & \bar{p}_2(x_{t+1}) \\ & = \sum_{y_{t,3}} \int_{\psi \in \Psi} p(\psi | x^t y^{t-1}) p(y_{t,3} | y_{t,2}, \psi) d\psi \\ & \int_{\theta \in \Theta} p(\theta | x^t y^{t-1}) p(x_{t+1} | x_t, y_t, \theta) d\theta, \quad (36) \end{aligned}$$

$p(\theta | x^t y^{t-1})$  及び  $p(\psi | x^t y^{t-1})$  は事後分布である. 式 (34) によるベイズ期待効用はパラメータの事前分布  $p(\theta)$ ,  $p(\psi)$  のもとで, 決定関数  $d_{pc}(\cdot, \cdot, \cdot)$  と  $d_{npc}(\cdot, \psi)$  を用いた場合の総利得の期待値である.

本研究では, 前述のとおり, 決定関数  $d_{npc}(\cdot, \psi)$  は固定 ( $\psi^*$  は未知) で, なるべく総利得を大きくするような決定関数  $d_{pc}(\cdot, \cdot, \cdot)$  を探索する問題を扱っている. 本研究の問題設定において, NPC を伴う RPG の  $T$  期間のプレイによる総利得をベイズ基準のもとで最大化するという意味で最適な決定関数 (攻略法)  $d_{pc}^*(\cdot, \cdot, \cdot)$  は次式で定義される.

$$\begin{aligned} & d_{pc}^*(\cdot, \cdot, \cdot) = \arg \max_{d_{pc}(\cdot, \cdot, \cdot)} \\ & BEU(d_{pc}(\cdot, \cdot, \cdot), d_{npc}(\cdot, \psi), p(\theta), p(\psi)). \quad (37) \end{aligned}$$

式 (37) によって定義されたベイズ最適な決定関数を具体的に算出するアルゴリズムを次節で提案する.

### 3.2 提案アルゴリズム

式 (34) によるベイズ期待効用は与えられた事前分布を逐次的に事後分布に更新する,  $T$  期間の入れ子構造になっている. NPC を伴うように従来研究 [9] を拡張したため, 状態遷移の確率計算部分などが従来研究 [9] よりも複雑になっている. しかし, ベイズ最適な決定関数  $d_{pc}^*(\cdot, \cdot, \cdot)$  は従来研究 [9] と同様に, ベイズ期待効用の入れ子構造に DP を適用することによって算出できる.

以下で, DP を用いてベイズ期待効用を最大化する

という意味で最適な決定関数  $d_{pc}^*(\cdot, \cdot, \cdot)$  を求めるアルゴリズムを示す. 提案アルゴリズムでは, DP で  $T$  期からさかのぼりながら, 各期の各状態と 1 期からその期に至るまでの遷移系列と期 (当該の期を示す整数) の組に対して PC の行動選択 (マップモード及び戦闘モード) と NPC の種類の選択 (戦闘モード) を行う.

$T$  期目の状態  $x_T$  (全ての状態の候補) と遷移系列  $x^T y^{T-1}$  (遷移系列の全ての候補) の組に対する処理は以下のとおりである. 状態  $x_T$  がマップモードの状態の場合,

$$\begin{aligned} & d_{pc}^*(x_T, x^T y^{T-1}, T) = \arg \max_{y_{T,1} \in A_{pc}(x_T)} \\ & \sum_{x_{T+1}} \bar{p}_2(x_{T+1}) r(x_T, y_T, x_{T+1}), \quad (38) \end{aligned}$$

$$\begin{aligned} & V(x_T, x^T y^{T-1}, T) = \max_{y_{T,1} \in A_{pc}(x_T)} \\ & \sum_{x_{T+1}} \bar{p}_2(x_{T+1}) r(x_T, y_T, x_{T+1}), \quad (39) \end{aligned}$$

状態  $x_T$  が戦闘モードの状態の場合,

$$\begin{aligned} & d_{pc}^*(x_T, x^T y^{T-1}, T) = \arg \max_{y_{T,1} \in A_{pc}(x_T), y_{T,2} \in A_{npc}} \\ & \sum_{x_{T+1}} \bar{p}_2(x_{T+1}) r(x_T, y_T, x_{T+1}), \quad (40) \end{aligned}$$

$$\begin{aligned} & V(x_T, x^T y^{T-1}, T) = \max_{y_{T,1} \in A_{pc}(x_T), y_{T,2} \in A_{npc}} \\ & \sum_{x_{T+1}} \bar{p}_2(x_{T+1}) r(x_T, y_T, x_{T+1}), \quad (41) \end{aligned}$$

ただし,  $V(x_T, x^T y^{T-1}, T)$  は最後の 1 期間の期待利得の最大値である. 本研究の問題設定では,  $T$  期にマップモードの場合には最後の 1 回の状態遷移で敵を倒すことは不可能なため, 式 (39) の計算結果は必ず 0 になる. しかし, 問題設定を多少変更する場合にも適用できるように, ここでは  $T$  期のマップモードについても DP の処理を記載している.

$t, 1 \leq t \leq T-1$  期目の状態  $x_t$  (全ての状態の候補) と遷移系列  $x^t y^{t-1}$  (遷移系列の全ての候補) の組に対する処理は以下のとおりである. 状態  $x_t$  がマップモードの状態の場合,

$$\begin{aligned} & d_{pc}^*(x_t, x^t y^{t-1}, t) = \arg \max_{y_{t,1} \in A_{pc}(x_t)} \sum_{x_{t+1}} \bar{p}_2(x_{t+1}) \\ & (r(x_t, y_t, x_{t+1}) + V(x_{t+1}, x^{t+1} y^t, t+1)), \quad (42) \end{aligned}$$

$$V(x_t, x^t y^{t-1}, t) = \max_{y_{t,1} \in A_{pc}(x_t)} \sum_{x_{t+1}} \bar{p}_2(x_{t+1}) \\ (r(x_t, y_t, x_{t+1}) + V(x_{t+1}, x^{t+1} y^t, t+1)), \quad (43)$$

状態  $x_t$  が戦闘モードの状態の場合,

$$d_{pc}^*(x_t, x^t y^{t-1}, t) = \arg \max_{y_{t,1} \in A_{pc}(x_t), y_{t,2} \in A_{npc}} \\ \sum_{x_{t+1}} \bar{p}_2(x_{t+1}) (r(x_t, y_t, x_{t+1}) + \\ V(x_{t+1}, x^{t+1} y^t, t+1)), \quad (44)$$

$$V(x_t, x^t y^{t-1}, t) = \max_{y_{t,1} \in A_{pc}(x_t), y_{t,2} \in A_{npc}} \\ \sum_{x_{t+1}} \bar{p}_2(x_{t+1}) (r(x_t, y_t, x_{t+1}) + \\ V(x_{t+1}, x^{t+1} y^t, t+1)). \quad (45)$$

上記の提案アルゴリズムにより, NPC を伴う RPG の  $T$  期間のプレイによる総利得をベイズ基準のもとで最大化するという意味で最適な PC の行動選択と NPC の種類の選択が実施できる。

提案アルゴリズムに含まれる式 (35) 及び式 (36) には積分計算があり, 一般的に積分計算の計算量は大きい。しかし, 事前分布としてディリクレ分布を仮定すると, 積分計算は四則演算で実施できる [9], [12], [13]。四則演算の一例として, 式 (21) の状態遷移に対応する式 (36) 中の積分計算部分を以下に示す。

$$\int_{\psi \in \Psi} p(\psi | x^t y^{t-1}) p(y_{t,3} | y_{t,2}, \psi) d\psi \\ = \frac{N(y_{t,2}, a_5 | x^t y^{t-1}) + \alpha(a_5 | y_{t,2})}{\sum_{a_i \in A_{npc}} N(y_{t,2}, a_i | x^t y^{t-1}) + \alpha(a_i | y_{t,2})}, \quad (46)$$

$$\int_{\theta \in \Theta} p(\theta | x^t y^{t-1}) p(x_{t+1} | x_t, y_t, \theta) d\theta \\ = \frac{N(a_5, x_{t,3}, C_1(x_{t,3}) | x^t y^{t-1}) + \alpha_1}{N(a_5, x_{t,3} | x^t y^{t-1}) + \alpha_2} \\ \frac{N(x_{t,3}, B(x_{t,3}) | x^t y^{t-1}) + \alpha_3}{N(x_{t,3} | x^t y^{t-1}) + \alpha_4}, \quad (47)$$

ただし,

$$\alpha_1 = \alpha(C_1(x_{t,3}) | a_5, x_{t,3}), \quad (48)$$

$$\alpha_2 = \alpha(C_1(x_{t,3}) | a_5, x_{t,3}) + \alpha(- | a_5, x_{t,3}), \quad (49)$$

$$\alpha_3 = \alpha(B(x_{t,3}) | x_{t,3}), \quad (50)$$

$$\alpha_4 = \alpha(B(x_{t,3}) | x_{t,3}) + \alpha(- | x_{t,3}), \quad (51)$$

$N(y_{t,2}, a_i | x^t y^{t-1})$  は系列  $x^t y^{t-1}$  中で NPC  $y_{t,2}$  が行動  $a_i$  を選択した回数,  $\alpha(a_i | y_{t,2})$  は  $p(a_i | y_{t,2}, \psi)$  に対するディリクレ分布のパラメータを示す。 $N(a_5, x_{t,3}, C_1(x_{t,3}) | x^t y^{t-1})$  は系列  $x^t y^{t-1}$  中で PC が敵  $x_{t,3}$  に対して攻撃行動  $a_5$  を選択して成功した回数,  $\alpha(C_1(x_{t,3}) | a_5, x_{t,3})$  は  $p(C_1(x_{t,3}) | a_5, x_{t,3}, \theta)$  に対するディリクレ分布のパラメータを示す。 $N(a_5, x_{t,3} | x^t y^{t-1})$  は系列  $x^t y^{t-1}$  中で PC が敵  $x_{t,3}$  に対して攻撃行動  $a_5$  を選択した回数,  $\alpha(- | a_5, x_{t,3})$  は攻撃に失敗した場合の確率  $p(- | a_5, x_{t,3}, \theta)$  に対するディリクレ分布のパラメータを示す。 $N(x_{t,3}, B(x_{t,3}) | x^t y^{t-1})$  は系列  $x^t y^{t-1}$  中で敵  $x_{t,3}$  が攻撃に成功した回数,  $\alpha(B(x_{t,3}) | x_{t,3})$  は  $p(B(x_{t,3}) | x_{t,3}, \theta)$  に対するディリクレ分布のパラメータを示す。 $N(x_{t,3} | x^t y^{t-1})$  は系列  $x^t y^{t-1}$  中で敵  $x_{t,3}$  に攻撃の順番が回った回数,  $\alpha(- | x_{t,3})$  は敵  $x_{t,3}$  が攻撃に失敗した場合の確率  $p(- | x_{t,3}, \theta)$  に対するディリクレ分布のパラメータを示す。このように, 事前分布としてディリクレ分布を採用することにより, 積分計算を四則演算で置き換えることができる。

ディリクレ分布のパラメータの設定が事前分布の設定に相当するが, 事前に何も情報がない場合の設定の仕方についてはベイズ統計学やその応用分野でいろいろな方法が検討されている。本研究のシミュレーションの際には, 多くの分野で良好な性質が報告されているジェフリーズの事前分布 [9]~[13] を採用し, 具体的には各パラメータを 0.5 に設定した。

#### 4. 提案アルゴリズムの有効性の確認

提案アルゴリズムの有効性を確認するために行ったシミュレーション結果のうち, 2 パターンを報告する。有効性を分かりやすく確認するため, 従来研究 [9] 同様に小規模な問題設定でシミュレーションを行った。

図 2 がパターン 1 のマップである。パターン 1 の設定を表 4~表 8 に示す。

$T = 7$  でベイズ最適な PC の行動選択, NPC の種類の選択を算出した。NPC  $h_1$  の選択回数 3 のうち攻撃  $a_5$  の選択回数 3, NPC  $h_2$  の選択回数 0 という学習状況 (遷移系列中のその他の頻度情報は説明を省略する) の 6 期の状態 (4,  $sm_4, e_1, 10$ ) におけるベイズ最適な PC の行動は攻撃  $a_5$ , NPC の種類の選択は  $h_1$  となった。残り 2 期間に対して, 敵の HP がまだ 10

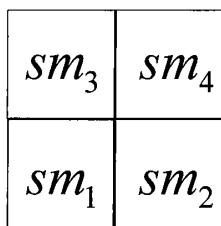


図 2 パターン 1 のマップ  
Fig. 2 A map of pattern1.

表 4 パターン 1 の地形の設定  
Table 4 Settings of the ground in pattern1.

$F(sm_1)$	$F(sm_2)$	$F(sm_3)$	$F(sm_4)$
$f_1$	$f_2$	$f_2$	$f_2$

表 5 パターン 1 の確率の設定 (その 1)  
Table 5 Settings of probabilities in pattern1 (the first part).

$p(e_1   f_2, \theta^*)$	$p(C_1(e_1)   a_5, e_1, \theta^*)$	$p(map   a_6, \theta^*)$
0.8	0.8	0.8

表 6 パターン 1 の確率の設定 (その 2)  
Table 6 Settings of probabilities in pattern1 (the second part).

$p(a_5   h_1, \psi^*)$	$p(a_5   h_2, \psi^*)$	$p(B(e_1)   e_1, \theta^*)$
0.9	0.1	0.6

表 7 パターン 1 のその他の設定 (その 1)  
Table 7 Othe settings in pattern1 (the first part).

$Mhp$	$E$	$H$	$M(e_1)$	$G(e_1)$
10	$\{e_1\}$	$\{h_1, h_2\}$	10	1

表 8 パターン 1 のその他の設定 (その 2)  
Table 8 Othe settings in pattern1 (the second part).

$C_1(e_1)$	$C_2(h_1, e_1)$	$C_2(h_2, e_1)$	$B(e_1)$
4	4	4	6

残っているため、積極的に攻撃しないと目の前の敵を倒せそうにない。そこで、6期までに攻撃の選択確率が高いことが学習できている NPC $h_1$  を選択している。また、NPC $h_1$  の選択回数 1 のうち攻撃  $a_5$  の選択回数 1, NPC $h_2$  の選択回数 2 のうち回復  $a_7$  の選択回数 2 という学習状況 (遷移系列中のその他の頻度情報は説明を省略する) の 6 期の状態 (4,  $sm_4, e_1, 2$ ) におけるベイズ最適な PC の行動は攻撃  $a_5$ , NPC の種類の選択は  $h_2$  となった。残り 2 期間に対して、敵の HP が 2 しか残っていない。また、残り 2 期間なので、目の前の敵を倒した後に更に新しい敵を倒すことはできない。そこで、6 期までに回復の選択確率が高いことが学習できている NPC $h_2$  を選択して、HP を回復して

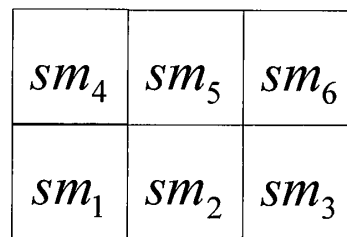


図 3 パターン 2 のマップ  
Fig. 3 A map of pattern2.

表 9 パターン 2 の地形の設定  
Table 9 Settings of the ground in pattern2.

$F(sm_1)$	$F(sm_2)$	$F(sm_3)$	$F(sm_4)$	$F(sm_5)$	$F(sm_6)$
$f_1$	$f_2$	$f_3$	$f_2$	$f_2$	$f_3$

表 10 パターン 2 の確率の設定 (その 1)  
Table 10 Settings of probabilities in pattern2 (the first part).

$p(e_1   f_2, \theta^*)$	$p(e_2   f_2, \theta^*)$	$p(e_1   f_3, \theta^*)$	$p(e_2   f_3, \theta^*)$
0.8	0.0	0.0	0.8

表 11 パターン 2 の確率の設定 (その 2)  
Table 11 Settings of probabilities in pattern2 (the second part).

$p(C_1(e_1)   a_5, e_1, \theta^*)$	$p(C_1(e_2)   a_5, e_2, \theta^*)$
0.9	0.7

表 12 パターン 2 の確率の設定 (その 3)  
Table 12 Settings of probabilities in pattern2 (the third part).

$p(map   a_6, \theta^*)$	$p(B(e_1)   e_1, \theta^*)$	$p(B(e_2)   e_2, \theta^*)$
0.8	0.4	0.6

表 13 パターン 2 の確率の設定 (その 4)  
Table 13 Settings of probabilities in pattern2 (the fourth part).

$p(a_5   h_1, \psi^*)$	$p(a_5   h_2, \psi^*)$
0.9	0.1

表 14 パターン 2 のその他の設定 (その 1)  
Table 14 Othe settings in pattern2 (the first part).

$Mhp$	$E$	$H$	$M(e_1)$	$M(e_2)$	$G(e_1)$	$G(e_2)$
10	$\{e_1, e_2\}$	$\{h_1, h_2\}$	10	10	1	10

表 15 パターン 2 のその他の設定 (その 2)  
Table 15 Othe settings in pattern2 (the second part).

$C_1(e_1)$	$C_1(e_2)$	$B(e_1)$	$B(e_2)$
6	4	4	6

より確実に目の前の敵を倒そうとしている。

図 3 がパターン 2 のマップである。パターン 2 の設定を表 9~表 16 に示す。



表 16 パターン 2 のその他の設定 (その 3)  
Table 16 Othe settings in pattern2 (the third part).

$C_2(h_1, e_1)$	$C_2(h_1, e_2)$	$C_2(h_2, e_1)$	$C_2(h_2, e_2)$
6	4	6	4

$T = 7$  でベイズ最適な PC の行動選択, NPC の種類の選択を算出した. NPC $h_1$  の選択回数 1 のうち攻撃  $a_5$  の選択回数 1, NPC $h_2$  の選択回数 1 のうち回復  $a_7$  の選択回数 1 という学習状況 (遷移系列中のその他の頻度情報は説明を省略する) の 5 期の状態  $(6, sm_5, e_1, 4)$  におけるベイズ最適な PC の行動は攻撃  $a_5$ , NPC の種類の選択は  $h_1$  となった. 残り 3 期間に対して, 敵  $e_1$  の HP が 4 残っているため, この期で目の前の敵を倒すことができれば, 残りの 2 期間で更に新しい敵を倒すことも可能である. そこで, 5 期までに攻撃の選択確率が高いことが学習できている NPC $h_1$  を選択している. また, NPC $h_1$  の選択回数 1 のうち攻撃  $a_5$  の選択回数 1, NPC $h_2$  の選択回数 1 のうち回復  $a_7$  の選択回数 1 という学習状況 (遷移系列中のその他の頻度情報は説明を省略する) の 5 期の状態  $(4, sm_3, e_2, 10)$  におけるベイズ最適な PC の行動は攻撃  $a_5$ , NPC の種類の選択は  $h_2$  となった. 残り 3 期間に対して, 敵  $e_2$  の HP が 10 残っている. HP が 10 の敵  $e_2$  を倒すためには最低でも 2 期間必要なので, 残り 3 期間では目の前の敵に加えて追加で新しい敵を倒すことはできない. そこで, 5 期までに回復の選択確率が高いことが学習できている NPC $h_2$  を選択して, HP を回復してより確実に目の前の敵を倒そうとしている.

少ない実験例ではあるが, NPC を伴う RPG におけるベイズ最適な PC の行動選択及び NPC の種類の選択方法により, 状況に応じた選択ができることを確認した.

## 5. むすび

NPC を伴う RPG をマルコフ決定過程を用いてモデル化し, 有限の  $T$  期間の期待総利得をベイズ基準のもとで最大化する PC の行動選択と NPC の種類の選択方法を DP を用いて算出する方法を提案した.

小規模かつ少数の実験例であるが, 提案方法によって, 状況に応じた PC の行動選択と NPC の種類の選択ができることを確認した.

従来研究 [9] 同様に本研究での実験例は小規模で, 仮定している RPG のモデルも実際のゲーム市場に存在

するものに比べてとてもシンプルなものである. そのため, NPC は本来は遊び手である人間の負荷を軽減する目的で導入されたものであるが, 本研究の RPG モデルでは NPC による人間の負荷軽減は十分に表現できていない. 実際のゲーム市場に存在するような規模の RPG を対象に検討するためには, 今後の課題として従来研究 [9] 同様に, 計算量の軽減が挙げられる.

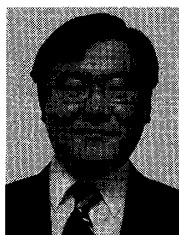
計算量が膨大になってしまうことは, MDP を RPG に適用する研究特有の課題ではなく, DP を扱う分野で共通の課題であり, 次元の呪い (curse of dimensionality) [14] と呼ばれている. 次元の呪いへの対処方法は大きく分けて, 最適性を保証したまま計算量を軽減する方法と, 近似によって計算量を軽減する方法に分かれる. 例えば, 従来研究 [15] では, 決定関数などの引数として状態と行動の系列ではなく, 本研究の式 (46) にもある各種頻度情報を利用することによって, 最適性を保証したまま計算量を  $T$  の指数オーダーから  $T$  の多項式オーダーに軽減する方法が提案されている. この方法は本研究にも適用可能である. また, 従来研究 [16] では,  $T$  期間の問題を解く代わりに  $T'$  ( $T' < T$ ) 期間の問題を逐次的に繰り返し解くことによる近似で計算量を軽減する方法が提案されている. この方法も本研究に適用可能である. 次元の呪いへの対処方法については多数の従来方法が存在し, 各方法で性質が異なる. よって, どの対処方法を RPG に適用すべきかの検討が今後の課題である.

## 文 献

- [1] 遠藤雅伸, “デジタルゲームにおける “AI” の役割,” 情報処理, vol.53, no.2, pp.146-152, Feb. 2012.
- [2] 張エイ, 椎塚久雄, “ゲームの面白さを演出する効果的な NPC の創出について,” 工学院大学研究報告第 109 号, pp.135-138, 2010.
- [3] 松原和也, 林 勇輔, 水久保勇記, 鶴身悠子, 垂水浩幸, “携帯電話向けインタラクティブエージェントと NPC への応用,” 情処学 EC 研報, vol.125, pp.71-78, Dec. 2005.
- [4] 金子哲夫, マルコフ決定理論入門, 槇書店, 1973.
- [5] L.P. Martin, Markov Decision Processes, John Wiley & Sons, 1994.
- [6] 森村英典, 高橋幸雄, マルコフ解析, 日科技連, 東京, 1979.
- [7] 高木幸一郎, 雨宮真人, “ロールプレイングゲーム (RPG) の戦闘におけるバランス自動調整システム開発のための基礎的考察,” 情処学 GI 研報, vol.28, pp.31-38, March 2001.
- [8] 高木幸一郎, 雨宮真人, “ロールプレイングゲーム (RPG) のバランスとは何か: 分析およびその調整に関する提案,” 情処学 GI 研報, vol.58, pp.67-74, June 2001.
- [9] 前田康成, 後藤文太郎, 升井洋志, 梶井文人, 鈴木正清,

“マルコフ決定過程のロールプレイングゲームへの適用,”  
 情処学論, vol.53, no.6, pp.1608-1616, June 2012.

- [10] J.O. Berger, *Statistical Decision Theory and Bayesian Analysis*, Springer-Verlag, New York, 1980.
  - [11] 繁榭算男, ベイズ統計入門, 東京大学出版会, 1985.
  - [12] T. Matsushima and H. Hirasawa, “A Bayes coding algorithm for Markov models,” *IEICE Technical Report*, IT95-1, 1995.
  - [13] 鈴木 譲, ベイジアンネットワーク入門, 培風館, 東京, 2009.
  - [14] D.P. Bertsekas, J.N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, Massachusetts, 1996.
  - [15] 前田康成, 浮田善文, 松嶋敏泰, 平澤茂一, “学習期間と制御期間に分割された強化学習問題における最適アルゴリズムの提案,” *情処学論*, vol.39, no.4, pp.1116-1126, April 1998.
  - [16] 前田康成, 松嶋敏泰, 平澤茂一, “未知パラメータを含むマルコフ決定過程に関する一考察,” *信学技報*, IT95-17, July 1995.
- (平成 24 年 10 月 30 日受付, 25 年 2 月 21 日再受付)



前田 康成 (正員)

平 7 早大・理工卒。平 9 同大学院理工学研究科修士課程了。日本電信電話 (株), 東日本電信電話 (株), 北見工大助手, 助教を経て平 22 同大学准教授, 現在に至る。博士 (工学)。統計的決定理論の学習問題への応用に関する研究に従事。情報処理学会等各会員。



後藤文太郎

昭 63 北大・工・電気卒。平 2 同大学院工学研究科電気工学専攻修士課程了。平 5 同博士後期課程単位修得退学。現在, 北見工業大学講師。博士 (工学)。論理型言語, インターネット応用, 観光情報学等の研究に従事。



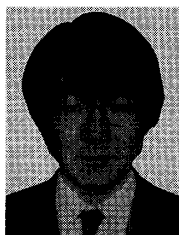
升井 洋志

1998 阪大・理・物理・博士後期課程了。博士 (理学)。2004 北見工業大学情報処理センター准教授 (当時は助教授)。情報処理学会, 日本物理学会, 日本原子力学会, 観光情報学会会員。



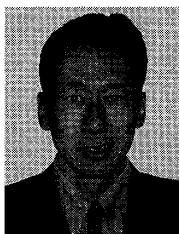
榭井 文人 (正員)

1990 岡山大・理・地卒。沖電気工業, 三重大・工・情報工助教を経て 2009 北見工大・工・情報システム工准教授。工博。自然言語処理, 観光情報学に興味をもつ。言語処理学会, 人工知能学会, 情報処理学会, 日本知能情報ファジィ学会, 観光情報学会各会員。



鈴木 正清 (正員)

昭 57 北大・工・電子卒。昭 62 同大学院博士課程了。同年同大応用電気研究所助手。平 5 北見工大・助教授, 平 8 北大・電子研助教授。センサアレー信号処理, 鮭追跡システムの開発, 国際会議運営支援システムの開発, 電子波包絡の回路モデルの研究に従事。平 13 より北見工大教授。工博。



松嶋 敏泰 (正員)

昭 53 早大・理工・工業経営卒。昭 55 同大学院修士課程了。同年, 日本電気 (株) 入社。昭 61 早大・理工学研究科・博士後期課程入学。現在, 早大・応用数理学科教授。知識情報処理及び情報理論とその応用に関する研究に従事。工博。IEEE 等各会員。