情報処理学会論文誌 Vol.53 No.6 1608–1616 (June 2012)

マルコフ決定過程のロールプレイングゲームへの適用

前田 康成[1,a)] 後藤 文太朗[1] 升井 洋志[1] 桝井 文人[1] 鈴木 正清[1]

受付日 2011年8月25日, 採録日 2012年3月2日

概要：従来からマルコフ決定過程（MDP）を用いたロールプレイングゲーム（RPG）のモデル化が行われている．従来研究では RPG が部分的にモデル化されている．本研究では，MDP を用いてより一般的な RPG のモデル化を行う．最初に MDP の真のパラメータ既知の場合に相当する RPG について，報酬の期待値を最大にするアルゴリズムを提案する．次に MDP の真のパラメータ未知の場合に相当する RPG について，ベイズ基準のもとで報酬を最大にするアルゴリズムを提案する．次に MDP の真のパラメータ未知の場合に相当する RPG について，学習データを用いて報酬を近似的に最大にするアルゴリズムを提案する．

キーワード：マルコフ決定過程，ロールプレイングゲーム，統計的決定理論，動的計画法，ベイズ基準

Applying Markov Decision Processes to Role-playing Game

Yasunari Maeda[1,a)] Fumitaro Goto[1] Hiroshi Masui[1] Fumito Masui[1]
Masakiyo Suzuki[1]

Received: August 25, 2011, Accepted: March 2, 2012

Abstract: In previous research a part of role-playing game (RPG) is represented with Markov decision processes (MDP). In this research we represent a more general PRG with MDP. We maximize an expected total reward under the condition that the true parameter of MDP is known in the first proposition. We maximize an expected total reward with respect to a Bayes criterion under the condition that the true parameter of MDP is unknown in the second proposition. We approximately maximize an expected total reward using learning data under the condition that the true parameter of MDP is unknown in the third proposition.

Keywords: Markov decision processes, role-playing game, statistical decision theory, dynamic programming, Bayes criterion

1. はじめに

近年，コンピュータの低価格化にともない，テレビゲーム機が広く普及し，ゲームの一分野としてロールプレイングゲーム（以下，RPG と表記する）も広く普及している．工学の分野においては，RPG を確率モデルを用いてモデル化して，ゲームを攻略する戦略について数理工学的に扱う研究 [1], [2] が行われている．

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プレイヤを移動させて，敵と遭遇すると戦闘モードになる冒険型の RPG を対象とする．従来研究 [1], [2] では，マルコフ決定過程（以下，MDP と表記する）[3], [4], [5] を用いて冒険型の RPG をモデル化しているが，その対象は戦闘モードのみである．

そこで，本研究ではマップ上でプレイヤを移動させるマップモードと戦闘モードを合わせて MDP を用いてモデル化する．このモデルを用いて，有限期間の報酬（お金や経験値など）の最大化問題を定式化し，報酬の期待値を最大化する戦略を動的計画法（以下，DP と表記する）を用いて求めるアルゴリズムを提案する．最後に小規模な例であるが，数値計算実験を行い，どのような戦略が求められるか検証する．

従来研究 [1], [2] では，RPG を MDP でモデル化する目的として自動開発があるが，本研究では将来的な目的を自動開発に限定しない．RPG を数理工学的に検討した場合

[1] 北見工業大学
Kitami Institute of Technology, Kitami, Hokkaido 090–8507, Japan
[a)] maeda@cs.kitami-it.ac.jp

の最適な戦略と実際の遊び手であるユーザの遊び方を比較することによって，ユーザの楽しみ方を工学的に解釈できる．ユーザの楽しみ方を工学的に解釈することの発展課題の 1 つが自動開発である．また，本研究では MDP の真のパラメータ既知の場合と未知の場合の両方を扱い，前者では開発者の立場で RPG の情報をすべて知っている場合の戦略を導出する．後者では，攻撃が成功する確率などの開発情報を知らないユーザの立場での戦略を導出する．

2. MDP を用いた RPG

2.1 本研究で研究対象とする RPG

以下で，本研究で研究対象とする RPG について説明する．

プレイヤはヒットポイント（以下，HP と表記する）と呼ばれる数値を持ち，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．

sm_i はマップ上の位置を示し，SM，$SM = \{sm_1, sm_2, \ldots, sm_{|SM|}\}$ 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プレイヤの位置はプレイヤにとって既知とする．f_i はマップ上の地形の種類を示し，F，$F = \{f_1, f_2, \ldots, f_{|F|}\}$ はマップ上の地形の種類の集合である．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，$E = \{e_1, e_2, \ldots, e_{|E|}\}$ は敵の種類の集合である．$M(e_i)$ は敵 e_i 出現時の敵 e_i の HP を示す．プレイヤは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．

プレイヤが選択できる行動（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．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．$mv(sm_i, a_j)$ はプレイヤが位置 sm_i で行動 a_j を選択した際の移動先の位置である．プレイヤの移動に際して，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．

戦闘モードの行動 a_5 はプレイヤが戦うための行動で，確率 $p(C(e_i)|a_5, e_i, \theta^*)$ で敵 e_i への攻撃に成功し，敵 e_i の HP が $C(e_i)$ 減少する．プレイヤは確率 $1 - p(C(e_i)|a_5, e_i, \theta^*)$ で敵 e_i への攻撃に失敗する．また，行動 a_5 の選択とは直接関係しないが，戦闘モードでは敵もプレイヤに対して攻撃し，確率 $p(B(e_i)|e_i, \theta^*)$ で敵 e_i がプレイヤへの攻撃に成功し，プレイヤの HP が $B(e_i)$ 減少する．攻撃はプレイヤがつねに先攻と仮定する．敵 e_i は確率 $1 - p(B(e_i)|e_i, \theta^*)$ でプレイヤへの攻撃に失敗する．行動 a_6 はプレイヤが敵から逃げるための行動で，確率 $p(map|a_6, \theta^*)$ でプレイヤは次の期にマップモードに移動し，確率 $1 - p(map|a_6, \theta^*)$ で戦闘モードが続く．行動 a_6 を選択した場合も，敵は攻撃してくる．よって，プレイヤが逃げることに失敗し，かつ敵が攻撃に成功するとプレイヤはダメージを受ける．θ^* は上記の各確率分布を支配する真のパラメータで本研究では，最初に既知の場合を検討し，次に未知の場合について検討する．

2.2 MDP と 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，$s_i \in S$，$S = \{s_1, s_2, \ldots, s_{|S|}\}$（$|S|$ は有限），各状態で選択できる行動 a_i，$a_i \in A$，$A = \{a_1, a_2, \ldots, a_{|A|}\}$（$|A|$ 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 $p(s_k \mid s_i, a_j, \xi^*)$（$\xi^*$ は遷移確率分布を支配する真のパラメータ），遷移にともない発生する利得 $r(s_i, a_j, s_k)$ 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．MDP の目的は，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である．プロセスの繰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を DP によって求めることができる．具体的には真のパラメータ ξ^* 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．決定関数は状態と期を受け取って，その期で選ぶべき行動を返す関数である．

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

次に，MDP と前節で説明した RPG の対応について説明する．

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 期におけるプレイヤの HP，$x_{t,2}$ は t 期におけるプレイヤのマップ上での位置，$x_{t,3}$ は t 期における

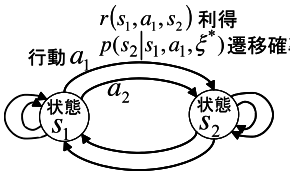

図 1 MDP の例
Fig. 1 An example of MDP.

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．

$A(x_t)$ は状態 x_t において選択可能な MDP の行動集合を示す．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．

次にマップモードの 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．$t+1$ 期には確率 $p(e_i|F(mv(x_{t,2}, y_t)), \theta^*)$ で敵 e_i が出現し，戦闘モードの状態 x_{t+1}，

$$\begin{aligned} x_{t+1} &= (x_{t+1,1}, x_{t+1,2}, x_{t+1,3}, x_{t+1,4}) \\ &= (x_{t,1}, mv(x_{t,2}, y_t), e_i, M(e_i)), \end{aligned} \tag{3}$$

に遷移する．ただし，ゲームのスタート位置である sm_1 が移動先 $mv(x_{t,2}, y_t)$ の場合には敵は出現しない（$p(e_i|F(mv(x_{t,2}, y_t)), \theta^*) = 0$）とする．また，確率 $1 - \sum_{e_i \in E} p(e_i|F(mv(x_{t,2}, y_t)), \theta^*)$ で敵が出現せずにマップモードの状態 x_{t+1} に遷移する．このときの状態 x_{t+1} は，移動先 $mv(x_{t,2}, y_t)$ が sm_1 の場合には，

$$\begin{aligned} x_{t+1} &= (x_{t+1,1}, x_{t+1,2}, x_{t+1,3}, x_{t+1,4}) \\ &= (M_{hp}, sm_1, x_{t,3}, x_{t,4}), \end{aligned} \tag{4}$$

移動先 $mv(x_{t,2}, y_t)$ が sm_1 以外の場合には，

$$\begin{aligned} x_{t+1} &= (x_{t+1,1}, x_{t+1,2}, x_{t+1,3}, x_{t+1,4}) \\ &= (x_{t,1}, mv(x_{t,2}, y_t), x_{t,3}, x_{t,4}), \end{aligned} \tag{5}$$

である．式 (4) の場合には，プレイヤがスタート位置 sm_1 に戻り，HP を最大値 M_{hp} まで回復した状態である．

次に戦闘モードの 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，行動 y_t が行動 a_5（戦う）の場合と，行動 a_6（逃げる）の場合に分けて説明する．ここでは，行動 a_5（戦う）の場合について説明する．確率 $(1 - p(C(x_{t,3})|a_5, x_{t,3}, \theta^*))(1 - p(B(x_{t,3})|x_{t,3}, \theta^*))$ でプレイヤと敵の両方が攻撃に失敗し，状態 x_{t+1}，

$$\begin{aligned} x_{t+1} &= (x_{t+1,1}, x_{t+1,2}, x_{t+1,3}, x_{t+1,4}) \\ &= (x_{t,1}, x_{t,2}, x_{t,3}, x_{t,4}), \end{aligned} \tag{6}$$

に遷移する．確率 $(1 - p(C(x_{t,3})|a_5, x_{t,3}, \theta^*))p(B(x_{t,3})|x_{t,3}, \theta^*)$ でプレイヤは攻撃に失敗し，敵は攻撃に成功し，状態 x_{t+1} へ遷移する．このときの状態 x_{t+1} は，$x_{t,1} > B(x_{t,3})$ の場合には，

$$\begin{aligned} x_{t+1} &= (x_{t+1,1}, x_{t+1,2}, x_{t+1,3}, x_{t+1,4}) \\ &= (x_{t,1} - B(x_{t,3}), x_{t,2}, x_{t,3}, x_{t,4}), \end{aligned} \tag{7}$$

で，$x_{t,1} \leq B(x_{t,3})$ の場合には，

$$x_{t+1} = (x_{t+1,1}, x_{t+1,2}, x_{t+1,3}, x_{t+1,4}) = (M_{hp}, sm_1, 0, 0), \tag{8}$$

である．式 (8) の場合には，プレイヤが敵に倒されて，ゲームのスタート位置 sm_1 からの再開である．確率 $p(C(x_{t,3})|a_5, x_{t,3}, \theta^*)(1 - p(B(x_{t,3})|x_{t,3}, \theta^*))$ でプレイヤは攻撃に成功し，敵は攻撃に失敗し，状態 x_{t+1} へ遷移する．このときの状態 x_{t+1} は，$x_{t,4} > C(x_{t,3})$ の場合には，

$$\begin{aligned} x_{t+1} &= (x_{t+1,1}, x_{t+1,2}, x_{t+1,3}, x_{t+1,4}) \\ &= (x_{t,1}, x_{t,2}, x_{t,3}, x_{t,4} - C(x_{t,3})), \end{aligned} \tag{9}$$

で，$x_{t,4} \leq C(x_{t,3})$ の場合には，

$$x_{t+1} = (x_{t+1,1}, x_{t+1,2}, x_{t+1,3}, x_{t+1,4}) = (x_{t,1}, x_{t,2}, 0, 0), \tag{10}$$

である．式 (10) の場合には，敵 $x_{t,3}$ を倒すことに成功しているので，この状態遷移にともない利得 $r(x_t, a_5, x_{t+1}) = G(x_{t,3})$ を得る．確率 $p(C(x_{t,3})|a_5, x_{t,3}, \theta^*)p(B(x_{t,3})|x_{t,3}, \theta^*)$ でプレイヤと敵の両方が攻撃に成功し，状態 x_{t+1} へ遷移する．このときの状態 x_{t+1} は，$x_{t,4} \leq C(x_{t,3})$ の場合には，

$$x_{t+1} = (x_{t+1,1}, x_{t+1,2}, x_{t+1,3}, x_{t+1,4}) = (x_{t,1}, x_{t,2}, 0, 0), \tag{11}$$

で，$x_{t,4} > C(x_{t,3})$ かつ $x_{t,1} > B(x_{t,3})$ の場合には，

$$\begin{aligned} x_{t+1} &= (x_{t+1,1}, x_{t+1,2}, x_{t+1,3}, x_{t+1,4}) \\ &= (x_{t,1} - B(x_{t,3}), x_{t,2}, x_{t,3}, x_{t,4} - C(x_{t,3})), \end{aligned} \tag{12}$$

で，$x_{t,4} > C(x_{t,3})$ かつ $x_{t,1} \leq B(x_{t,3})$ の場合には，

$$x_{t+1} = (x_{t+1,1}, x_{t+1,2}, x_{t+1,3}, x_{t+1,4}) = (M_{hp}, sm_1, 0, 0), \tag{13}$$

である．式 (11) の場合には，敵 $x_{t,3}$ を倒すことに成功しているので，この状態遷移にともない利得 $r(x_t, a_5, x_{t+1}) = G(x_{t,3})$ を得る．式 (13) の場合には，プレイヤが敵に倒されて，ゲームのスタート位置 sm_1 からの再開である．

次に戦闘モードの t 期の状態 x_t で行動 a_6（逃げる）を選択したときの状態遷移について説明する．確率 $p(map|a_6, \theta^*)$ でプレイヤが逃げることに成功し，状態 x_{t+1}，

$$x_{t+1} = (x_{t+1,1}, x_{t+1,2}, x_{t+1,3}, x_{t+1,4}) = (x_{t,1}, x_{t,2}, 0, 0), \tag{14}$$

に遷移する．確率 $(1 - p(map|a_6, \theta^*))(1 - p(B(x_{t,3})|x_{t,3}, \theta^*))$ でプレイヤが逃げることに失敗し，敵が攻撃に失敗し，状態 x_{t+1}，

$$\begin{aligned} x_{t+1} &= (x_{t+1,1}, x_{t+1,2}, x_{t+1,3}, x_{t+1,4}) \\ &= (x_{t,1}, x_{t,2}, x_{t,3}, x_{t,4}), \end{aligned} \tag{15}$$

に遷移する．確率 $(1 - p(map|a_6, \theta^*))p(B(x_{t,3})|x_{t,3}, \theta^*)$ で

プレイヤが逃げることに失敗し，敵が攻撃に成功し，状態 x_{t+1} へ遷移する．このときの状態 x_{t+1} は，$x_{t,1} > B(x_{t,3})$ の場合には，

$$\begin{aligned} x_{t+1} &= (x_{t+1,1}, x_{t+1,2}, x_{t+1,3}, x_{t+1,4}) \\ &= (x_{t,1} - B(x_{t,3}), x_{t,2}, x_{t,3}, x_{t,4}), \end{aligned} \tag{16}$$

で，$x_{t,1} \leq B(x_{t,3})$ の場合には，

$$x_{t+1}=(x_{t+1,1}, x_{t+1,2}, x_{t+1,3}, x_{t+1,4})=(M_{hp}, sm_1, 0, 0), \tag{17}$$

である．式 (17) の場合には，プレイヤが敵に倒されて，ゲームのスタート位置 sm_1 からの再開である．

前述したとおり，プレイヤが敵 $x_{t,3}$ を倒した状態遷移にともなう利得は，$r(x_t, a_5, x_{t+1}) = G(x_{t,3})$ である．その他の状態遷移にともなう利得は，$r(x_t, a_5, x_{t+1}) = 0$ である．本研究では，初期状態 x_1 が $x_1 = (M_{hp}, sm_1, 0, 0)$，各期の状態は観測可能とする．また，プレイヤや敵の攻撃力 $C(e_i)$，$B(e_i)$ および敵を倒したときの報酬 $G(e_i)$ などはすべて既知とする．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. 提案アルゴリズム

統計的決定理論 [6], [7] に基づいて定式化し，行動選択の仕方を求めるアルゴリズムを提案する．

3.1 真のパラメータ既知の場合

ここでは，真のパラメータ θ^* 既知の場合について検討する．真のパラメータが既知という仮定はゲームの開発者の立場になるので，この仮定のもとで求める最適な行動選択の仕方は，一般的に攻略法などと呼ばれているものに相当する．

最初に統計的決定理論に基づいて，効用関数 $U(d(\cdot,\cdot), x^{T+1}y^T, \theta^*)$ を次式で定義する．

$$U(d(\cdot,\cdot), x^{T+1}y^T, \theta^*) = \sum_{t=1}^{T} r(x_t, y_t, x_{t+1}), \tag{18}$$

ただし，$d(\cdot,\cdot)$ は引数が状態と期（期を示す整数）で，選択する行動を返す決定関数，$x^{T+1}y^T$ は系列 $x_1y_1x_2y_2\cdots x_Ty_Tx_{T+1}$ を示す．効用関数 $U(d(\cdot,\cdot), x^{T+1}y^T, \theta^*)$ は真のパラメータ θ^* のもとで，ある決定関数 $d(\cdot,\cdot)$ を用いて，$x^{T+1}y^T$ と遷移した場合の総利得である．

次に統計的決定理論に基づいて，期待効用 $EU(d(\cdot,\cdot), \theta^*)$ を次式で定義する．

$$\begin{aligned} &EU(d(\cdot,\cdot), \theta^*) \\ &= E\left(\sum_{t=1}^{T} r(x_t, y_t, x_{t+1})\right) \\ &= \sum_{x^Ty^T} p(x_2 \,|\, x_1, y_1, \theta^*)(r(x_1, y_1, x_2) \\ &\quad + p(x_3 \,|\, x_2, y_2, \theta^*)(r(x_2, y_2, x_3) \\ &\quad + \cdots + p(x_{T+1} \mid x_T, y_T, \theta^*) r(x_T, y_T, x_{T+1}))\cdots)), \end{aligned} \tag{19}$$

ただし，期待効用 $EU(d(\cdot,\cdot), \theta^*)$ は真のパラメータ θ^* のもとで，ある決定関数 $d(\cdot,\cdot)$ を用いた場合の総利得の期待値である．

真のパラメータ θ^* 既知の場合には，次式で定義される期待効用を最大にする決定関数が最適な決定関数（行動選択の仕方）である．

$$d^*(\cdot,\cdot) = \arg\max_{d(\cdot,\cdot)} EU(d(\cdot,\cdot), \theta^*). \tag{20}$$

以下で，DP を用いて期待効用を最大化するという点で最適な決定関数を求めるアルゴリズムを示す．

式 (20) で定義された最適な決定関数を DP を用いて求めるには，T 期間の MDP の問題を T 期目の末端から遡りながら解く．

T 期目の状態 x_T（すべての T 期目の状態の候補）に対する処理は以下のとおりである．

$$\begin{aligned} &d^*(x_T, T) \\ &= \arg\max_{y_T \in A(x_T)} \sum_{x_{T+1}} p(x_{T+1} \mid x_T, y_T, \theta^*) r(x_T, y_T, x_{T+1}), \end{aligned} \tag{21}$$

$$\begin{aligned} &V(x_T, T) \\ &= \max_{y_T \in A(x_T)} \sum_{x_{T+1}} p(x_{T+1} \,|\, x_T, y_T, \theta^*) r(x_T, y_T, x_{T+1}), \end{aligned} \tag{22}$$

ただし，$d^*(x_T, T)$ は T 期目の状態 x_T において選択すべき最適な行動で，$V(x_T, T)$ は T 期目の状態 x_T から $T+1$ 期への状態遷移にともなう最後の 1 期間の期待利得の最大値である．

t 期目（$1 \leq t \leq T-1$）の状態 x_t（すべての t 期目の状態の候補）に対する処理は以下のとおりである．

$$\begin{aligned} &d^*(x_t, t) \\ &\quad = \arg\max_{y_t \in A(x_t)} \sum_{x_{t+1}} p(x_{t+1} \,|\, x_t, y_t, \theta^*)(r(x_t, y_t, x_{t+1}) \\ &\qquad + V(x_{t+1}, t+1)), \end{aligned} \tag{23}$$

$$\begin{aligned} V(x_t, t) = \max_{y_t \in A(x_t)} \sum_{x_{t+1}} &p(x_{t+1} \mid x_t, y_t, \theta^*)(r(x_t, y_t, x_{t+1}) \\ &+ V(x_{t+1}, t+1)), \end{aligned} \tag{24}$$

ただし，$d^*(x_t, t)$ は t 期目の状態 x_t において選択すべき最適な行動で，$V(x_t, t)$ は t 期目の状態が x_t という条件のもとでの，t 期以降の期待総利得の最大値である．式 (21) から式 (24) を用いて $d^*(x_1, 1)$ まで求めることによって，1 期目から T 期目までのすべての状態に対して，期待総利得を最大にするという点で最適な行動選択の仕方を求めることができる．

3.2 真のパラメータ未知の場合

前節では，真のパラメータ既知の場合の最適な行動選択の仕方を DP を用いて求める方法を示したが，これはゲーム開発者が提供する攻略法の一種に相当する．しかし，一般的にゲームをプレイする側のユーザは真のパラメータは知らない．敵の攻撃力や報酬など，何がユーザにとって既知か未知かはいろいろなパターンが想定可能だが，本節では一例として，攻撃力や報酬などは既知で，真のパラメータのみ未知という仮定のもとで最適な行動選択の仕方について検討する．

真のパラメータ θ^* 未知の場合を検討するに際して，いくつか新たな定義を行う．$p(\theta)$ はパラメータ θ の事前分布で既知である．Θ はパラメータ空間で，$\theta^* \in \Theta$，$\theta \in \Theta$ である．$x^t y^{t-1}$ は t 期目の状態 x_t に至るまでの遷移系列で，$x^t y^{t-1} = x_1 y_1 \cdots x_t$ である．

効用関数と期待効用は基本的に前節と同様だが，本節では真のパラメータが未知なので，事前分布に対してパラメータ空間で期待値をとるベイズ期待効用を次式で定義する．

$$BEU(d_B(\cdot,\cdot,\cdot), p(\theta)) = \int_\Theta p(\theta) EU(d_B(\cdot,\cdot,\cdot),\theta) d\theta, \quad (25)$$

ただし，決定関数についてはその期に至るまでの系列も受け取るように引数を増やして拡張している．真のパラメータ θ^* 未知の場合には，次式で定義されるベイズ期待効用を最大にする決定関数が最適な決定関数（行動選択の仕方）である．

$$d_B^*(\cdot,\cdot,\cdot) = \arg \max_{d_B(\cdot,\cdot,\cdot)} BEU(d_B(\cdot,\cdot,\cdot), p(\theta)). \quad (26)$$

以下で，DP を用いてベイズ期待効用を最大化するという点で最適な決定関数を求めるアルゴリズムを示す．

前節の真のパラメータ既知の場合には，DP で T 期から遡りながら各期の各状態に対して行動選択を行った．本節の真のパラメータ未知の場合には，DP で T 期から遡りながら，各期の各状態と 1 期からその期に至るまでの各遷移系列の組に対して行動選択を行う．

T 期目の状態 x_T（すべての状態の候補）と遷移系列 $x^T y^{T-1}$（すべての遷移系列の候補）の組に対する処理は以下のとおりである．

$$\begin{aligned} &d_B^*(x_T, x^T y^{T-1}, T) \\ &= \arg \max_{y_T \in A(x_T)} \sum_{x_{T+1}} \int_\Theta p(\theta \mid x^T y^{T-1}) p(x_{T+1} \mid x_T, y_T, \theta) d\theta \\ &\quad r(x_T, y_T, x_{T+1}), \end{aligned} \quad (27)$$

$$\begin{aligned} &V_B(x_T, x^T y^{T-1}, T) \\ &= \max_{y_T \in A(x_T)} \sum_{x_{T+1}} \int_\Theta p(\theta \mid x^T y^{T-1}) p(x_{T+1} \mid x_T, y_T, \theta) d\theta \\ &\quad r(x_T, y_T, x_{T+1}), \end{aligned} \quad (28)$$

ただし，$p(\theta \mid x^T y^{T-1})$ は 1 期から T 期に遷移系列 $x^T y^{T-1}$ のように遷移した場合の事後分布である．

t 期目（$1 \le t \le T-1$）の状態 x_t（すべての状態の候補）と遷移系列 $x^t y^{t-1}$（すべての遷移系列の候補）の組に対する処理は以下のとおりである．

$$\begin{aligned} &d_B^*(x_t, x^t y^{t-1}, t) \\ &= \arg \max_{y_t \in A(x_t)} \sum_{x_{t+1}} \int_\Theta p(\theta \mid x^t y^{t-1}) p(x_{t+1} \mid x_t, y_t, \theta) d\theta \\ &\quad (r(x_t, y_t, x_{t+1}) + V_B(x_{t+1}, x^{t+1} y^t, t+1)). \end{aligned} \quad (29)$$

$$\begin{aligned} &V_B(x_t, x^t y^{t-1}, t) \\ &= \max_{y_t \in A(x_t)} \sum_{x_{t+1}} \int_\Theta p(\theta \mid x^t y^{t-1}) p(x_{t+1} \mid x_t, y_t, \theta) d\theta \\ &\quad (r(x_t, y_t, x_{t+1}) + V_B(x_{t+1}, x^{t+1} y^t, t+1)). \end{aligned} \quad (30)$$

式 (27) から式 (30) を用いて $d_B^*(x_1, x_1, 1)$ まで求めることによって，1 期目から T 期目までのすべての状態と遷移系列の組に対して，ベイズ基準のもとで総利得を最大にするという点で最適な行動選択の仕方を求めることができる．

式 (27) から式 (30) には積分計算が含まれており，一般的に積分計算の計算量は大きいが，二項分布（敵の出現以外の確率分布）の事前分布としてベータ分布，多項分布（敵の出現の確率分布）の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 [8]．四則演算の一例として，マップモードの t 期の状態 x_t において行動 y_t を選択したもとで，敵 e_i が出現し，戦闘モードの状態 x_{t+1} に遷移する場合の $\int_\Theta p(\theta \mid x^t y^{t-1}) p(x_{t+1} \mid x_t, y_t, \theta) d\theta$ の計算を以下に示す．

$$\begin{aligned} &\int_\Theta p(\theta \mid x^t y^{t-1}) p(x_{t+1} \mid x_t, y_t, \theta) d\theta \\ &= \int_\Theta p(\theta \mid x^t y^{t-1}) p(e_i | F(mv(x_{t,2}, y_t)), \theta) d\theta \\ &= \frac{N(F(mv(x_{t,2}, y_t)), e_i \mid x^t y^{t-1}) + \alpha_1}{N(F(mv(x_{t,2}, y_t)) \mid x^t y^{t-1}) + \alpha_2}, \end{aligned} \quad (31)$$

ただし，

$$\alpha_1 = \alpha(e_i \mid F(mv(x_{t,2}, y_t))), \quad (32)$$

$$\begin{aligned} \alpha_2 = &\sum_{e_j \in E} \alpha(e_j \mid F(mv(x_{t,2}, y_t))) \\ &+ \alpha(mv(x_{t,2}, y_t) \mid F(mv(x_{t,2}, y_t))), \end{aligned} \quad (33)$$

$N(F(mv(x_{t,2}, y_t)), e_i \mid x^t y^{t-1})$ は系列 $x^t y^{t-1}$ 中で地形の種類が $F(mv(x_{t,2}, y_t))$ の位置で敵 e_i が出現した回数，$N(F(mv(x_{t,2}, y_t)) \mid x^t y^{t-1})$ は系列 $x^t y^{t-1}$ 中で移動先の位置の地形の種類が $F(mv(x_{t,2}, y_t))$ だった回数，$\alpha(e_i \mid F(mv(x_{t,2}, y_t)))$ は $p(e_i \mid F(mv(x_{t,2}, y_t)), \theta)$ に対するディリクレ分布（事前分布）のパラメータ，$\alpha(mv(x_{t,2}, y_t) \mid F(mv(x_{t,2}, y_t)))$ は $1 - \sum_{e_i \in E} p(e_i \mid F(mv(x_{t,2}, y_t)), \theta)$ に対するディリクレ分布（事前分布）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やベータ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．ディリクレ分布やベータ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研究されている．本研究の実験の際には，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 [6], [7], [8], [9] を採用し，具体的には各パラメータを 0.5 に設定する．

事前分布にディリクレ分布やベータ分布を採用してジェフリーズの事前分布に設定し，式 (27) から式 (30) で処理することにより，真のパラメータ未知の場合にベイズ基準のもとで総利得を最大化できる．このベイズ最適な行動選択の仕方は，ユーザが真のパラメータについて何も知らない状況でプレイする際に最適な行動選択の仕方である．

なお，適用分野を特定せずに一般的な MDP を研究対象とした従来研究において，真のパラメータ未知の場合の MDP に関する検討も行われている [10]．よって，本節の内容は従来研究 [10] の検討結果を RPG に適用したと解釈できる．

3.3 真のパラメータ未知の場合の近似

前節では，真のパラメータ未知の場合のベイズ最適な行動選択の仕方を求めるアルゴリズムを提案した．事前分布としてディリクレ分布やベータ分布を採用することにより，積分計算を四則演算に置き換えた．しかし，ベイズ最適な行動選択の仕方を求めるためには大きな計算量が必要である．真のパラメータ既知の場合には，DP の各期ごとに式 (24) の処理を状態数の分だけ実施する．これに対し，真のパラメータ未知のベイズ最適の場合には，DP の各期ごとに式 (30) の処理を状態数と遷移系列の個数の積の分だけ実施する必要があり，処理の回数は期の数（t 期の t）に対する指数オーダになる．

そこで，本節では真のパラメータ未知の場合の近似の一例を提案する．ここで，学習データ L を新たに導入する．学習データは過去のゲームのプレイデータである遷移系列の集合であったり，敵の出現確率など個々の確率分布について真のパラメータの分布から発生させたサンプルデータであったり，いろいろな形態が考えられる．

前節のベイズ最適な方法では，各期ごとにその期までの遷移系列 $x^t y^{t-1}$ に対する事後分布によって，$\int_\Theta p(\theta \mid x^t y^{t-1}) p(x_{t+1} \mid x_t, y_t, \theta) d\theta$ を計算したが，近似アルゴリズムでは期に関係なく学習データ L による事後分布を用いて $\int_\Theta p(\theta \mid L) p(x_{t+1} \mid x_t, y_t, \theta) d\theta$ を計算する．具体的には，$\int_\Theta p(\theta \mid L) p(x_{t+1} \mid x_t, y_t, \theta) d\theta$ を真のパラメータ既知の場合の式 (21) から式 (24) に代入して行動選択の仕方を求める．

近似アルゴリズムにより，DP の処理の回数は真のパラメータ既知の場合と同じ回数に軽減できる．有限の学習データに対する理論的な精度保証はないが，漸近的には学習データによる事後分布を用いた推定値が真のパラメータに収束するので，求める行動選択の仕方も真のパラメータ既知の場合に収束する．

4. 実験例

4.1 真のパラメータ既知の場合の実験例

3.1 節で提案した真のパラメータ既知の場合の行動選択の仕方を求めるアルゴリズムについて，実験例を報告する．**図 2** に実験におけるマップを示す．実験結果の解釈がしやすいように 9 マスからなる小規模なマップにした．その他の設定を**表 1**，**表 2**，**表 3**，**表 4**，**表 5** に示す．

上記の設定で真のパラメータ既知の場合のアルゴリズムを適用して，10 期間の期待総利得を最大化するための各期における行動選択の仕方を求めた．結果の一部を紹介すると，3 期でプレイヤの HP が 6 で位置 sm_4 にいるマップモードの状態では，最適な行動選択は a_3 という上のマス sm_7 への移動だった．これは，プレイヤの HP にまだ余裕があるので，弱くて報酬の小さい敵が出現する右（a_1）や HP を回復する下（a_4）ではなく，強くて報酬の大きい敵が出現する上への移動を選択している．他方，同じ 3 期のマップモードでも HP が 1 で位置 sm_4 にいる状態では，行動 a_4 を選択して下の HP を回復してくれるスタート位置

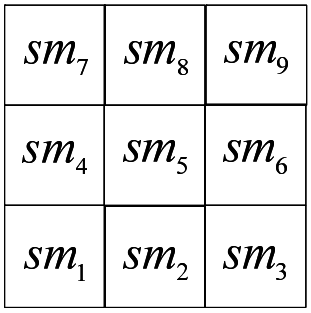

図 2 実験のマップ
Fig. 2 A map of experiment.

表 2 確率の設定（その 1）
Table 2 Settings of probabilities (the first part).

$p(e_1 \mid f_2, \theta^*)$	$p(e_2 \mid f_2, \theta^*)$	$p(e_1 \mid f_3, \theta^*)$	$p(e_2 \mid f_3, \theta^*)$
0.3	0.0	0.0	0.8

表 1 地形の設定
Table 1 Settings of the ground.

$F(sm_1)$	$F(sm_2)$	$F(sm_3)$	$F(sm_4)$	$F(sm_5)$	$F(sm_6)$	$F(sm_7)$	$F(sm_8)$	$F(sm_9)$
f_1	f_2	f_3	f_2	f_2	f_3	f_3	f_3	f_3

表 3 確率の設定（その 2）
Table 3 Settings of probabilities (the second part).

$p(C(e_1)\|a_5,e_1,\theta^*)$	$p(C(e_2)\|a_5,e_2,\theta^*)$	$p(B(e_1)\|e_1,\theta^*)$	$p(B(e_2)\|e_2,\theta^*)$	$p(map\|a_6,\theta^*)$
0.9	0.9	0.6	0.6	0.6

表 4 その他の設定（その 1）
Table 4 Othe settings (the first part).

Mhp	E	$M(e_1)$	$M(e_2)$	$G(e_1)$	$G(e_2)$
10	$\{e_1, e_2\}$	2	4	1	5

表 5 その他の設定（その 2）
Table 5 Othe settings (the second part).

$C(e_1)$	$C(e_2)$	$B(e_1)$	$B(e_2)$	T
3	3	1	3	10

sm_1 に移動している．これは，プレイヤの HP に余裕がないので回復するための行動選択である．また，9 期に HP が 1 でマップモードで sm_4 にいる状態の場合には，行動 a_1 を選択して，右の弱くて報酬の小さい敵が出現する sm_5 へ移動している．これは，プレイヤの HP に余裕はないが，残りの期間にも余裕がないので，弱い敵が出現する sm_5 への移動を選択している．

このように，真のパラメータ既知の場合のアルゴリズムを適用することにより，開発者であれば知っている真のパラメータの情報を利用して対象期間の期待総利得を最大にする行動選択の仕方を求めることができる．

4.2 真のパラメータ未知の場合の近似の実験例

前節と同じ設定のもとで，3.3 節で提案した真のパラメータが未知の場合の近似アルゴリズムを適用した．学習データとして，各確率分布ごとにサンプルデータを発生させた．そのもとで，近似アルゴリズムを適用し，各期の各状態における行動選択が前節の真のパラメータ既知の場合と比較して一致しているかどうかを調べた．各確率分布の学習データ数を 10，100，1,000 と変化させ，それぞれの学習データ数に対して 100 パターンの学習データを発生させて適用実験を行った．真のパラメータ既知の場合の行動選択との一致率は 100 パターンの平均で，学習データ数が 10 の場合が約 88%，学習データ数が 100 の場合が約 94%，学習データ数が 1,000 の場合が約 96%だった．

このように，学習データによる事後分布を利用する近似アルゴリズムを用いることにより，真のパラメータが未知の場合でも，DP に必要な計算量を真のパラメータ既知の場合と同程度で実施できる．また，少ない実験例ではあるが，学習データの増加にともない真のパラメータ既知の場合との行動選択の一致率が高くなることが確認できた．

5. 考察と今後の課題

本研究では，RPG の MDP によるモデル化とその行動選択の仕方を研究対象とした．従来研究でも，RPG の MDP によるモデル化は検討されていたが，従来研究では戦闘モードのみがモデル化されていた．本研究では，マップモードと戦闘モードを合わせたより一般化された RPG を MDP によってモデル化した．

本研究では，マップモードと戦闘モードを合わせた RPG 全体を MDP でモデル化したが，一般的な RPG に比べるととても簡素なものになっている．一般的な RPG ではマップモードや戦闘モード以外にも塔や城などの建造物内でのモードや洞窟内でのモードなども含まれることが多い．プレイヤや敵の攻撃についても，武器や魔法など多種多様である．しかし，これらの一般化については，MDP における状態空間，行動集合，遷移確率の拡張により対応可能である．

真のパラメータ未知の場合の近似アルゴリズムを提案したが，使用している学習データは，RPG の遊び手であるユーザが本格的にプレイする前に行う練習などによるユーザの経験に相当する．今回の実験では各確率分布ごとに均一にサンプルを発生させたが，実際のユーザは均一なサンプル収集とは異なる何らかの戦略に基づいてサンプル（学習データ）を得ていると想定される．このような，戦略に基づく学習データの獲得については学習理論における能動学習の分野で研究されており，一般的な MDP についても研究されている [11], [12], [13], [14]．RPG を対象とした場合の能動学習については今後の課題としたい．

状態空間などの拡張によるさらなる一般化の可能性について前述したが，アルゴリズム（DP）の計算量を考慮すると，モデル化は容易でも最適な行動選択の仕方の算出にはより大きな計算量が必要になる．状態空間の膨張に対する対処は一般的な MDP の分野でも研究されており，部分観測可能な MDP（POMDP）による対応 [15], [16] がよく行われている．RPG における POMDP による検討も今後の課題としたい．

将来的には，敵と戦うための行動の種類などを複数にした，さらにより一般的な RPG を表現する MDP に従って動く RPG を実装する．この RPG を実際にユーザにプレイしてもらい，プレイの履歴データを収集する．また，楽しかった部分についてアンケートに回答してもらう．プレイの履歴データとアンケート結果を照合して楽しかった部分での行動選択の仕方を抽出し，本研究の提案アルゴリズ

ムで算出可能な理論的に最適な行動選択の仕方と比較する．RPG の楽しさに関連する要素として，敵を倒したときの報酬の大きさ，敵の出現など各種イベントの発生確率，各種行動の名称（武器や魔法の名称など），敵やプレイヤを表現する画像，マップのデザインなどが考えられる．上記の比較結果から特に有効な要素を見つけ，ユーザの楽しみ方を解明したい．

参考文献

[1] 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，情報処理学会研究報告，GI, Vol.2001, No.28, pp.31–38 (2001).
[2] 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，情報処理学会研究報告，GI, Vol.2001, No.58, pp.67–74 (2001).
[3] 金子哲夫：マルコフ決定理論入門，槙書店 (1973).
[4] Martin, L.P.: *Markov Decision Processes*, John Wiley & Sons (1994).
[5] 森村英典，高橋幸雄：マルコフ解析，日科技連，東京 (1979).
[6] Berger, J.O.: *Statistical Decision Theory and Bayesian Analysis*, Springer-Verlag, New York (1980).
[7] 繁桝算男：ベイズ統計入門，東京大学出版会 (1985).
[8] Matsushima, T. and Hirasawa, S.: A Bayes coding algorithm for Markov models, TECHNICAL REPORT OF IEICE, IT95-1, pp.1–6 (1995).
[9] 鈴木　譲：ベイジアンネットワーク入門，培風館，東京 (2009).
[10] Martin, J.J.: *Bayesian Decision Problems and Markov Chains*, John Wiley & Sons (1967).
[11] 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，情報処理学会論文誌，Vol.39, No.4, pp.1116–1126 (1998).
[12] 宮崎和光，山村雅幸，小林重信：k-確実探査法強化学習における環境同定のための行動選択戦略，人工知能学会誌，Vol.10, No.3, pp.454–463 (1995).
[13] 宮崎和光，山村雅幸，小林重信：l-確実探査法エージェントによる環境同定のための行動選択戦略，人工知能学会誌，Vol.11, No.5, pp.804–808 (1996).
[14] 宮崎和光，山村雅幸，小林重信：MarcoPolo 報酬獲得と環境同定のトレードオフを考慮した強化学習システム，人工知能学会誌，Vol.12, No.1, pp.78–89 (1997).
[15] 木村　元，Kaelbling, L.P.：部分観測マルコフ決定過程下での強化学習，人工知能学会誌，Vol.12, No.6, pp.822–830 (1997).
[16] 宮崎和光，荒井幸代，小林重信：POMDPs 環境下での決定的政策の学習，人工知能学会誌，Vol.14, No.1, pp.148–156 (1999).

前田 康成 （正会員）

平成 7 年早稲田大学理工学部卒業．平成 9 年同大学院理工学研究科修士課程修了．日本電信電話（株），東日本電信電話（株），北見工業大学助手，助教を経て平成 22 年同大学准教授，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．電子情報通信学会等会員．

後藤 文太朗 （正会員）

昭和 63 年北海道大学工学部電気工学科卒業．平成 2 年同大学院工学研究科電気工学専攻修士課程修了．平成 5 年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 （正会員）

平成 10 年大阪大学大学院理工学研究科物理学専攻博士後期課程修了．博士（理学）．平成 16 年北見工業大学情報処理センター准教授（当時は助教授）．日本物理学会，日本原子力学会，観光情報学会各会員．

桝井 文人 （正会員）

平成 2 年岡山大学理学部地学科卒業．沖電気工業，三重大学工学部情報工学科助教を経て，平成 21 年北見工業大学工学部情報システム工学科准教授．工学博士．自然言語処理，観光情報学に興味を持つ．言語処理学会，人工知能学会，電子情報通信学会，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 （正会員）

昭和 57 年北海道大学工学部電子工学科卒業．昭和 62 年同大学大学院博士課程修了．同年同大学応用電気研究所助手．平成 5 年北見工業大学助教授，平成 8 年北海道大学電子科学研究所助教授．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．平成 13 年より北見工業大学教授．工学博士．