

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6103683号
(P6103683)

(45) 発行日 平成29年3月29日 (2017.3.29)

(24) 登録日 平成29年3月10日 (2017.3.10)

(51) Int. Cl. F 1
A 6 3 F 13/60 (2014.01) A 6 3 F 13/60
A 6 3 F 13/822 (2014.01) A 6 3 F 13/822

請求項の数 4 (全 19 頁)

<p>(21) 出願番号 特願2012-128918 (P2012-128918) (22) 出願日 平成24年6月6日 (2012.6.6) (65) 公開番号 特開2013-252247 (P2013-252247A) (43) 公開日 平成25年12月19日 (2013.12.19) 審査請求日 平成27年5月26日 (2015.5.26)</p>	<p>(73) 特許権者 504238806 国立大学法人北見工業大学 北海道北見市公園町165番地 (74) 代理人 100081271 弁理士 吉田 芳春 (74) 代理人 100159628 弁理士 吉田 雅比呂 (74) 代理人 100162189 弁理士 堀越 真弓 (72) 発明者 前田 康成 北海道北見市公園町165番地 国立大学 法人北見工業大学内 審査官 古川 直樹</p>
--	--

最終頁に続く

(54) 【発明の名称】 ロールプレイングゲームの攻略法算出装置、算出方法、算出プログラム及びこのプログラムを記録した記録媒体

(57) 【特許請求の範囲】

【請求項1】

ロールプレイングゲームのプレイヤーのゲーム開始時点における初期状態と制御期間の長さと確率分布を支配する真のパラメータ * 又は近似のための学習データLとが与えられた際に、該制御期間におけるマルコフ決定過程問題を動的計画法で求めて該制御期間中に得られる総報酬に相当する期待総利得を最大にする政策を出力する最適政策算出部と、前記プレイヤーの状態及び時点と確率分布を支配する真のパラメータ * 又は近似のための学習データLとが与えられると当該時点の当該状態においてマルコフ決定過程問題のそれ以降の期待総利得を最大にする最適行動及び期待総利得の最大値を出力する行動決定部とを備えており、

前記プレイヤーの初期状態及び制御期間の長さと確率分布を支配する真のパラメータ * 又は近似のための学習データLとに対して制御期間におけるマルコフ決定過程問題を動的計画法で求めて該制御期間における期待総利得を最大にすることが保証された最適政策を出力することを特徴とするロールプレイングゲームの攻略法算出装置。

【請求項2】

ロールプレイングゲームのプレイヤーのゲーム開始時点における初期状態と制御期間の長さと確率分布を支配する真のパラメータ * 又は近似のための学習データLとが与えられた際に、該制御期間におけるマルコフ決定過程問題を動的計画法で求めて該制御期間中に得られる総報酬に相当する期待総利得を最大にする政策を出力する最適政策算出工程と

前記プレイヤーの状態及び時点と確率分布を支配する真のパラメータ * 又は近似のための学習データLとが与えられると当該時点の当該状態においてマルコフ決定過程問題のそれ以降の期待総利得を最大にする最適行動及び期待総利得の最大値を出力する行動決定工程とを備えており、

前記プレイヤーの初期状態及び制御期間の長さ^と確率分布を支配する真のパラメータ * 又は近似のための学習データLとに対して制御期間におけるマルコフ決定過程問題を動的計画法で求めて該制御期間における期待総利得を最大にすることが保証された最適政策を出力することを特徴とするロールプレイングゲームの攻略法算出方法。

【請求項3】

ロールプレイングゲームのプレイヤーのゲーム開始時点における初期状態と制御期間の長さ^と確率分布を支配する真のパラメータ * 又は近似のための学習データLとが与えられた際に、該制御期間におけるマルコフ決定過程問題を動的計画法で求めて該制御期間中に得られる総報酬に相当する期待総利得を最大にする政策を出力する最適政策算出手順と

10

前記プレイヤーの状態及び時点と確率分布を支配する真のパラメータ * 又は近似のための学習データLとが与えられると当該時点の当該状態においてマルコフ決定過程問題のそれ以降の期待総利得を最大にする最適行動及び期待総利得の最大値を出力する行動決定手順とをコンピュータで実行させ、

前記プレイヤーの初期状態及び制御期間の長さ^と確率分布を支配する真のパラメータ * 又は近似のための学習データLとに対して制御期間におけるマルコフ決定過程問題を動的計画法で求めて該制御期間における期待総利得を最大にすることが保証された最適政策を出力することを特徴とするロールプレイングゲームの攻略法算出プログラム。

20

【請求項4】

ロールプレイングゲームのプレイヤーのゲーム開始時点における初期状態と制御期間の長さ^と確率分布を支配する真のパラメータ * 又は近似のための学習データLとが与えられた際に、該制御期間におけるマルコフ決定過程問題を動的計画法で求めて該制御期間中に得られる総報酬に相当する期待総利得を最大にする政策を出力する最適政策算出手順と

前記プレイヤーの状態及び時点と確率分布を支配する真のパラメータ * 又は近似のための学習データLとが与えられると当該時点の当該状態においてマルコフ決定過程問題のそれ以降の期待総利得を最大にする最適行動及び期待総利得の最大値を出力する行動決定手順とをコンピュータで実行させる攻略法算出プログラムを記録したコンピュータ読み取り可能な記録媒体であり、

30

前記プレイヤーの初期状態及び制御期間の長さ^と確率分布を支配する真のパラメータ * 又は近似のための学習データLとに対して制御期間におけるマルコフ決定過程問題を動的計画法で求めて該制御期間における期待総利得を最大にすることが保証された最適政策を出力することを特徴とするロールプレイングゲームの攻略法算出プログラムを記録した記録媒体。

【発明の詳細な説明】

【技術分野】

40

【0001】

本発明は、ロールプレイングゲーム(RPG)をモデル化してゲームの攻略法(行動選択の仕方)を算出するための攻略法算出装置、算出方法、算出プログラム及びこのプログラムを記録した記録媒体に関する。

【背景技術】

【0002】

近年、コンピュータの低価格化に伴い、テレビゲーム機が広く普及し、ゲームの一分野として、RPGも広く普及している。このようなRPGの開発を行う場合、従来は以下のような問題点を有していた。

(1) プレイヤーが遊ぶ際にどのようにプレイするかを把握するためには、多くの被験者

50

に遊んでもらいプレイヤーの体験データを取得することが行われる。しかしながら、これは、多くの被験者を雇う必要があるため、コストが大幅に高くなるのみならず、データを
得るのに多大な時間を要する。

(2) 日々販売されるRPGは多数あるが、その中でヒットするRPGは非常に少数である。そのようにヒットしたRPGには、人間が楽しいと感じる要素が多数含まれていると
考えられるが、どのような要素が楽しいと感じる要素であるのが工学的には未だ把握され
ていない。

(3) 最近では、プレイヤーの補佐を行うキャラクタ(お仲間キャラクタ)をコンピュータ
が操作するRPGも多いが、このようなお仲間キャラクタの賢い行動(コマンド)選択の
仕方をプログラミングすることはかなり難しい。

10

【0003】

一方、ゲーム情報学等の工学分野においては、RPGをマルコフ決定過程(MDP)等
の確率モデルを用いてモデル化し、ゲームを攻略する戦略について数理工学的に扱う研究
がなされ、報告されている(例えば、非特許文献1及び2)。

【先行技術文献】

【非特許文献】

【0004】

【非特許文献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、2001(28)、pp.31-38(2001)

20

【非特許文献2】高木幸一郎、雨宮真人、「ロールプレイングゲーム(RPG)の balan
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、200
1(58)、pp.67-74(2001)

【発明の概要】

【発明が解決しようとする課題】

【0005】

非特許文献1及び2に開示されているとき、従来のモデル化は、RPGの一部のみの
モデル化であった。即ち、プレイヤーがマップ上を移動するマップモードと、マップモ
ードにおいて敵と遭遇した際に開始される戦闘モードとからなる冒険型のRPGにおいて、
戦闘モードのみのモデル化であり、これでは、RPG全体の攻略法(行動選択の仕方)を
算出することができず、前述した(1)~(3)のごとき開発支援上の問題点を解決する
ことができなかった。

30

【0006】

従って本発明の目的は、被験者の体験データを取得することなく、RPGの攻略法(行
動選択の仕方)を算出することができるRPGの攻略法算出装置、算出方法、算出プログラ
ム及びこのプログラムを記録した記録媒体を提供することにある。

【0007】

本発明の他の目的は、人間が楽しいと感じるゲーム要素を工学的に把握することができ
るRPGの攻略法算出装置、算出方法、算出プログラム及びこのプログラムを記録した記
録媒体を提供することにある。

40

【0008】

本発明のさらに目的は、コンピュータによって操作されるキャラクタの行動選択の仕方
を容易にプログラミングすることができるRPGの攻略法(行動選択の仕方)を算出する
ことができるRPGの攻略法算出装置、算出方法、算出プログラム及びこのプログラムを
記録した記録媒体を提供することにある。

【課題を解決するための手段】

【0009】

本発明によれば、RPGのプレイヤーのゲーム開始時点における初期状態と制御期間の
長さとは与えられた際に、この制御期間中に得られる総報酬に相当する期待総利得を最大
にする政策を出力する最適政策算出部と、プレイヤーの状態及び時点が与えられるとその

50

時点のその状態においてそれ以降の期待総利得を最大にする最適行動及び期待総利得の最大値を出力する行動決定部とを備えており、プレイヤーの初期状態及び制御期間の長さに対して制御期間における期待総利得を最大にすることが保証された最適政策を出力するRPGの攻略法算出装置が提供される。

【0010】

RPGの自動開発に資する研究開発としてMDPなる確率モデルを用いた定式化は従来より行われていたが、従来技術では、RPGの一部のイベントを定式化するのみの不十分なものであった。

【0011】

本発明によれば、RPG全体をMDPで定式化している。これにより、
(A) 開発者のみが知っている情報を既知と仮定したもとの攻略法の算出を行い、
(B) 開発者のみが知っている情報を未知と仮定した(一般のプレイヤーと同様の立場を仮定した)もとの攻略法の算出を行っている。

10

【0012】

これは、ゲーム情報学分野における学際的な知見を与えるのみならず、ゲーム産業においても以下のごとく有用である。

【0013】

第1に、本発明で算出される目的(お金の獲得等)のもとの攻略法(行動選択の仕方)をコンピュータにシミュレーションさせることによって、その目的におけるプレイヤーのゲーム結果をシミュレートできる。このシミュレーションを利用することによって、マップ上に隠されたアイテムやイベントに遭遇する割合(仮にプレイヤーが1万人いた場合に何人が遭遇できるか)等を把握でき、その割合を見ながら適切な隠し場所を設定することができる。

20

【0014】

第2に、本発明で算出される数理工学的に最適な攻略法と実際のプレイヤーによる攻略法との比較を行うことにより、人間が楽しいと感じるゲーム要素を工学的に把握できる可能性がある。人間が楽しいと感じるゲーム要素を確率モデル上のパラメータ設定のある種のパターンとして把握できれば、そのような要素を多く含むゲーム開発を行うことができる。

【0015】

第3に、近年では、ゲーム中にプレイヤーに協力するコンピュータ操作のキャラクタの登場が多い。このようなコンピュータによって操作されるキャラクタの行動選択の仕方をプログラミングするのは難しかったが、本発明により種々の目的毎の攻略法を算出することによって、各目的に適した(プレイヤーに協力する)キャラクタの行動選択の仕方(攻略法)をプログラミングすることが可能となる。

30

【0016】

最適政策算出部は、制御期間におけるMDP問題を動的計画法(DP)で求めるように構成されていることが好ましい。

【0017】

本発明によれば、さらに、RPGのプレイヤーのゲーム開始時点における初期状態と制御期間の長さとは与えられた際に、この制御期間中に得られる総報酬に相当する期待総利得を最大にする政策を出力する最適政策算出工程と、プレイヤーの状態及び時点が与えられるとその時点のその状態においてそれ以降の期待総利得を最大にする最適行動及び期待総利得の最大値を出力する行動決定工程とを備えており、プレイヤーの初期状態及び制御期間の長さに対して制御期間における期待総利得を最大にすることが保証された最適政策を出力するRPGの攻略法算出方法が提供される。

40

【0018】

最適政策算出工程は、制御期間におけるMDP問題をDPで求めるように構成されていることが好ましい。

【0019】

50

本発明によれば、さらにまた、RPGのプレイヤーのゲーム開始時点における初期状態と制御期間の長さとは与えられた際に、この制御期間中に得られる総報酬に相当する期待総利得を最大にする政策を出力する最適政策算出手順と、プレイヤーの状態及び時点が与えられるとその時点のその状態においてそれ以降の期待総利得を最大にする最適行動及び期待総利得の最大値を出力する行動決定手順とをコンピュータで実行させ、プレイヤーの初期状態及び制御期間の長さに対して制御期間における期待総利得を最大にすることが保証された最適政策を出力するRPGの攻略法算出プログラムが提供される。

【0020】

最適政策算出手順は、制御期間におけるMDP問題をDPで求めるように構成されていることが好ましい。

10

【0021】

本発明によれば、さらに、RPGのプレイヤーのゲーム開始時点における初期状態と制御期間の長さとは与えられた際に、この制御期間中に得られる総報酬に相当する期待総利得を最大にする政策を出力する最適政策算出手順と、プレイヤーの状態及び時点が与えられるとその時点のその状態においてそれ以降の期待総利得を最大にする最適行動及び期待総利得の最大値を出力する行動決定手順とをコンピュータで実行させる攻略法算出プログラムを記録したコンピュータ読み取り可能な記録媒体であり、プレイヤーの初期状態及び制御期間の長さに対して制御期間における期待総利得を最大にすることが保証された最適政策を出力するRPGの攻略法算出プログラムを記録した記録媒体が提供される。

【0022】

最適政策算出手順は、制御期間におけるMDP問題をDPで求めるように構成されていることが好ましい。

20

【発明の効果】

【0023】

本発明によれば、制御期間における期待総利得を最大にすることが保証された政策が出力されるので、プレイヤーの初期状態と制御期間長とに対して制御期間における期待総利得を最大にする政策を出力することが可能となり、そのRPGに関する攻略法（行動選択の仕方）を算出することができる。

【0024】

即ち、本発明によれば、攻略法をコンピュータにシミュレーションさせることによって、被験者の体験データを取得することができ、算出される数理工学的に最適な攻略法と実際のプレイヤーによる攻略法との比較を行うことにより、人間が楽しいと感じるゲーム要素を工学的に把握することができ、さらに、人間が楽しいと感じるゲーム要素を確率モデル上のパラメータ設定のある種のパターンとして把握できれば、そのような要素を多く含むゲーム開発を行うことができる。また、種々の目的毎の攻略法を算出することによって、各目的に適したプレイヤーに協力するキャラクタの攻略法を容易にプログラミングすることが可能となる。

30

【図面の簡単な説明】

【0025】

【図1】本発明の第1の実施形態として、RPGの開発支援に用いる攻略法算出装置の全体構成を概略的に示すブロック図である。

40

【図2】図1の実施形態における攻略法算出装置の主要部の動作を説明するフローチャートである。

【図3】図1の実施形態における攻略法算出装置の主要部の構成を概略的に示すブロック図である。

【図4】図1の実施形態におけるDPグラフの一例を示す図である。

【図5】図1の実施形態における攻略法算出装置の行動決定部の動作を説明するフローチャートである。

【図6】第1の実施形態及び第2の実施形態の実施例におけるマップの構成例を示す図である。

50

【発明を実施するための形態】

【0026】

図1は本発明の第1の実施形態としてRPGの開発支援に用いる攻略法算出装置の全体構成を概略的に示しており、図2は本実施形態における攻略法算出装置の主要部の動作を説明しており、図3は本実施形態における攻略法算出装置の主要部の構成を概略的に示しており、図4は本実施形態におけるDPグラフの一例を示しており、図5は本実施形態における攻略法算出装置の行動決定部の動作を説明している。この第1の実施形態は、各種確率分布を支配する真のパラメータ*が既知の場合である。

【0027】

図1に示すように、本実施形態における攻略法算出装置は、バス10を介して互いに接続された中央処理装置(CPU)11と、リードオンリメモリ(ROM)12と、ランダムアクセスメモリ(RAM)13と、ハードディスク駆動装置(HDD)14と、サウンド処理部15と、画像処理部16と、ブルーレイディスク/デジタルバーサタイルディスク(BR/DVD)駆動装置17と、入出力インタフェース18と、通信インタフェース19とを備えたコンピュータ及びこれを作動させるプログラムから構成される。

10

【0028】

サウンド処理部15はスピーカ20に接続されており、画像処理部16は表示ディスプレイ21に接続されている。BR/DVD駆動装置17はブルーレイディスク/デジタルバーサタイルディスク/コンパクトディスク(BR/DVD/CD)22が装着可能となっており、入出力インタフェース18にはコントローラ23、キーボード24及びマウス25が接続されている。

20

【0029】

CPU11は、ROM12に記憶されているオペレーションシステム(OS)やブートプログラム等の基本プログラムに従ってRAM13に記憶されているプログラムを実行して本実施形態の処理を行う。また、CPU11は、RAM13、HDD14、音声処理部15、画像処理部16、BR/DVD駆動装置17、入出力インタフェース18、及び通信インタフェース19の動作を制御する。

【0030】

RAM13は攻略法算出装置のメインメモリとして使用され、HDD14やBR/DVD駆動装置17から転送されたプログラムやデータを記憶する。また、RAM13は、プログラム実行時の各種データが一時的に記憶されるワークエリアとしても使用される。

30

【0031】

HDD14は、プログラム及びデータがあらかじめ記憶されているか、又は通信インタフェース19を介して外部のネットワーク取り込んだプログラム及びデータが記憶される。

【0032】

サウンド処理部15は、CPU11の指示に従ってゲームの背景音や効果音等のサウンドデータを再生するための処理を行い、スピーカ20へその音声信号を出力する。

【0033】

画像処理部16は、CPU11の指示に従って2次元又は3次元グラフィック処理を行い、画像データを生成する。生成された画像データは、表示ディスプレイ21に出力される。表示ディスプレイ21ではなく図示しないTVの画面に表示する場合には、同期信号を付加したビデオ信号を出力する。

40

【0034】

BR/DVD駆動装置17は、CPU11の指示に従って、セットされたBR/DVD/CD22からゲームに関連するプログラムやデータを読み出し、RAM13へ転送する。また、セットされたBR/DVD/CD22へプログラムやデータの書き込みをすることも可能である。

【0035】

入出力インタフェース18は、コントローラ23、キーボード24及びマウス25とC

50

P U 1 1 又は R A M 1 3 との間のデータのやり取りを制御する。コントローラ 2 3 には、ゲームを行う際に操作される方向キーやボタン等を備えている。

【 0 0 3 6 】

通信インタフェース 1 9 は、通信回線を介して外部ネットワークに接続されており、C P U 1 1 の指示に従って、外部ネットワークとの間でプログラムやデータのやり取りが可能となっている。

【 0 0 3 7 】

このような構成の攻略法算出装置において、C P U 1 1 は、作動時は、まず、R A M 1 3 内にプログラム記憶領域、データ記憶領域及びワークエリアを確保し、H D D 1 4 又は外部からプログラム及びデータを取り込んで、プログラム記憶領域及びデータ記憶領域に格納する。次いで、このプログラム記憶領域に格納されたプログラムに基づいて、図 2 に示す処理を実行する。C P U 1 1 がプログラムを実行することによって、図 3 に概略的に示すとき攻略法算出装置が構築される。

10

【 0 0 3 8 】

即ち、図 3 に示すように、本実施形態の攻略法算出装置は、最適政策算出部 3 0 と、行動決定部 3 1 とを備えるように構築される。ここで、最適政策算出部 3 0 は D P グラフ作成器 3 0 a と、D P 実施器 3 0 b とを備え、行動決定部 3 1 は行動決定器 3 1 a と、遷移確率テーブル 3 1 b と、利得テーブル 3 1 c とを備えるように構築される。

【 0 0 3 9 】

この攻略法算出装置の各部説明を行う前に、本実施形態で用いるマルコフ決定過程 (M D P) を利用したロールプレイングゲーム (R P G) の数理モデルと、その数理モデルで表現される R P G の仕様とについて説明する。

20

【 0 0 4 0 】

まず、本実施形態における R P G の仕様について説明する。

【 0 0 4 1 】

プレイヤーはヒットポイント (H P) と呼ばれる数値を持ち、H P が 0 となると、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、H P は、スタート時と同じ最大値 M_{h_p} まで回復する。

【 0 0 4 2 】

s_{m_i} はマップ上の位置を示し、 S_M 、 $S_M = \{ s_{m_1}, s_{m_2}, \dots, s_{m_{|S_M|}} \}$ は、マップ上の位置の集合である。ここで、ゲーム開始時のスタート位置を s_{m_1} とする。なお、本実施形態においては、スタート位置及び現在のプレイヤーの位置は、プレイヤーによって既知であるとする。 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 $F(s_{m_i})$ F で分かる。

30

【 0 0 4 3 】

e_i は、敵の種類を示し、 E 、 $E = \{ e_1, e_2, \dots, e_{|E|} \}$ は敵の種類の集合である。 $M(e_i)$ は、敵 e_i の出現時のこの敵 e_i の H P を示す。プレイヤーは、敵を攻撃することによって敵の H P 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。

40

【 0 0 4 4 】

プレイヤーが選択できる行動 (コマンド) は、マップモードと戦闘モードとでは異なり、マップモードでは a_1 から a_4 が選択可能であり、戦闘モードでは a_5 及び a_6 が選択可能である。 a_1, a_2, a_3, a_4 は、マップ上でそれぞれ右、左、上、下に移動するための行動である。 $mv(s_{m_i}, a_j)$ はプレイヤーが位置 s_{m_i} で行動 a_j を選択した際の移動先位置である。プレイヤーの移動に際して、確率 $p(e_k | F(mv(s_{m_i}, a_j)))$ 、 $*$) で移動先 $mv(s_{m_i}, a_j)$ 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い、確率 $1 - \sum_{e_k \in E} p(e_k | F(mv(s_{m_i}, a_j)))$ 、 $*$) で何も出現せずにマップモードが続く。

【 0 0 4 5 】

50

戦闘モードの行動 a_5 はプレイヤーが戦うための行動であり、確率 $p(C(e_i) | a_5, e_i, \xi^*)$ で敵 e_i への攻撃に成功し、その場合、敵 e_i のHPが $C(e_i)$ だけ減少する。プレイヤーは、確率 $1 - p(C(e_i) | a_5, e_i, \xi^*)$ で敵 e_i への攻撃に失敗する。また、行動 a_5 の選択とは直接的に関係しないが、戦闘モードでは敵もプレイヤーに対して攻撃し、確率 $p(B(e_i) | e_i, \xi^*)$ で敵 e_i がプレイヤーへの攻撃に成功し、その場合、プレイヤーのHPが $B(e_i)$ だけ減少する。攻撃は、プレイヤーが常に先攻すると仮定する。敵 e_i は、確率 $1 - p(B(e_i) | e_i, \xi^*)$ でプレイヤーへの攻撃に失敗する。行動 a_6 は、プレイヤーが敵から逃げるための行動であり、確率 $p(\text{map} | a_6, \xi^*)$ でプレイヤーは次の期にマップモードで移動し、確率 $1 - p(\text{map} | a_6, \xi^*)$ で戦闘モードが続く。行動 a_6 を選択した場合にも、敵は攻撃してくる。よって、プレイヤーが逃げることに失敗し、かつ敵が攻撃に成功すると、プレイヤーはダメージを受ける。 ξ^* は、上述の各確率分布を支配する真のパラメータであり、本実施形態では既知であるとする。

10

【0046】

次に、確率システムの動的な最適化問題を定式化する優れた能力を有する数理モデルであるMDPについてその概要を説明する。

【0047】

MDPについては、例えば、金子哲夫、「マルコフ決定理論入門」、槇書店(1973)や森村英典、高橋幸雄、「マルコフ解析」、日科技連、東京(1979)等に記載されている。

20

【0048】

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 を選択したもとの、状態 s_k へ遷移する遷移確率 $p(s_k | s_i, a_j, \xi^*)$ (ξ^* は遷移確率分布を支配する真のパラメータ)、遷移に伴って発生する利得 $r(s_i, a_j, s_k)$ で構成される。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を動的計画法(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^* が既知の場合であれば、下記の式(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、状態と期とを受け取って、その期で選ぶべき行動を返す関数である。

30

【0049】

【数1】

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

【0050】

次に、MDPと本実施形態におけるRPGとの対応について説明する。

40

【0051】

x_t は、MDPにおける t 期の状態を示す変数であり、式(2)のように構成される。

【0052】

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

ただし、 $x_{t,1}$ は t 期におけるプレイヤーのHP、 $x_{t,2}$ は t 期におけるプレイヤー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をそれぞれ示し、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。

【0053】

$A(x_t)$ 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

50

における t 期に選択した行動を示す変数である。

【 0 0 5 4 】

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 $p(e_i | F(mv(x_t, 2, y_t)), *)$ で敵 e_i が出現し、戦闘モードの状態 x_{t+1} 、

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

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 ($p(e_i | F(mv(x_t, 2, y_t)), *) = 0$) とする。また、確率 $1 - p(e_i | F(mv(x_t, 2, y_t)), *)$ 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、

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

移動先 $mv(x_t, 2, y_t)$ が sm_1 以外の場合には、

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

である。式 (4) の場合は、プレイヤーがスタート位置 sm_1 に戻り、HP を最大値 M_{hp} まで回復した状態である。

【 0 0 5 5 】

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) の場合とに分けて説明する。

【 0 0 5 6 】

まず、行動 a_5 (戦う) の場合について説明する。確率 $1 - p(C(x_t, 3) | a_5, x_t, 3, *) (1 - p(B(x_t, 3) | x_t, 3, *))$ でプレイヤーと敵との両方が攻撃に失敗し、状態 x_{t+1} 、

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

に遷移する。確率 $(1 - p(C(x_t, 3) | a_5, x_t, 3, *)) p(B(x_t, 3) | x_t, 3, *))$ でプレイヤーは攻撃に失敗し、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、

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

で、 $x_t, 1 \leq B(x_t, 3)$ の場合には、

$$\begin{aligned} x_{t+1} &= (x_{t+1, 1}, x_{t+1, 2}, x_{t+1, 3}, x_{t+1, 4}) \\ &= (M_{hp}, sm_1, 0, 0) \end{aligned} \quad (8)$$

である。式 (8) の場合には、プレイヤーが敵に倒されて、ゲームのスタート位置 sm_1 からの再開である。確率 $p(C(x_t, 3) | a_5, x_t, 3, *) (1 - p(B(x_t, 3) | x_t, 3, *))$ でプレイヤーは攻撃に成功し、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、

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

で、 $x_t, 4 \leq C(x_t, 3)$ の場合には、

$$\begin{aligned} x_{t+1} &= (x_{t+1, 1}, x_{t+1, 2}, x_{t+1, 3}, x_{t+1, 4}) \\ &= (x_t, 1, x_t, 2, 0, 0) \end{aligned} \quad (10)$$

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、この状態遷移に伴い、利得 $r(x_t, a_5, x_{t+1}) = G(x_t, 3)$ を得る。確率 $p(C(x_t, 3) | a_5, x_t, 3, *) p(B(x_t, 3) | x_t, 3, *))$ でプレイヤー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 $C(x_{t,3})$ の場合には、

$$\begin{aligned} x_{t+1} &= (x_{t+1,1}, x_{t+1,2}, x_{t+1,3}, x_{t+1,4}) \\ &= (x_{t,1}, x_{t,2}, 0, 0) \quad (11) \end{aligned}$$

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

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

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

$$\begin{aligned} x_{t+1} &= (x_{t+1,1}, x_{t+1,2}, x_{t+1,3}, x_{t+1,4}) \\ &= (M_{hp}, sm_1, 0, 0) \quad (13) \end{aligned}$$

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、この状態遷移に伴い、利得 $r(x_t, a_5, x_{t+1}) = G(x_{t,3})$ を得る。式(13)の場合はプレイヤーが敵に倒されて、ゲームのスタート位置 sm_1 からの再開である。

【0057】

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 $p(\text{map} | a_6, *)$ でプレイヤーが逃げることに成功し、状態 x_{t+1} 、

$$\begin{aligned} x_{t+1} &= (x_{t+1,1}, x_{t+1,2}, x_{t+1,3}, x_{t+1,4}) \\ &= (x_{t,1}, x_{t,2}, 0, 0) \quad (14) \end{aligned}$$

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

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

に遷移する。確率 $(1 - p(\text{map} | a_6, *)) p(B(x_{t,3}) | x_{t,3}, *)$ でプレイヤー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、

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

)

で、 $x_{t,1} \leq B(x_{t,3})$ の場合には、

$$\begin{aligned} x_{t+1} &= (x_{t+1,1}, x_{t+1,2}, x_{t+1,3}, x_{t+1,4}) \\ &= (M_{hp}, sm_1, 0, 0) \quad (17) \end{aligned}$$

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

【0058】

前述した通り、プレイヤーが敵 $x_{t,3}$ を倒した状態遷移に伴う利得は、 $r(x_t, a_5, x_{t+1}) = G(x_{t,3})$ である。その他の状態遷移に伴う利得は、 $r(x_t, a_5, x_{t+1}) = 0$ である。本実施形態では、初期状態 x_1 が $x_1 = (M_{hp}, sm_1, 0, 0)$ であり、各期の状態は観測可能である。また、プレイヤーや敵の攻撃力 $C(e_i)$ 、 $B(e_i)$ 及び敵を倒したときの報酬 $G(e_i)$ 等は全て既知であるとする。このもとで、 T 期間のプレイを行って総利得

【数2】

$$\sum_{t=1}^T r(x_t, y_t, x_{t+1})$$

の最大化を目的とする。

【0059】

次に、図2に示されたフローチャート、図3に示されたブロック図、図4に示された図、及び図5に示されたフローチャートを参照して、本実施形態の攻略法算出装置の動作を説明する。

【0060】

10

20

30

40

50

図2及び図3に示すように、まず、構築された最適政策算出部30におけるDPグラフ作成器30aにプレイヤーの初期状態 x_1 と制御期間長 T とが入力される(ステップS1)。この初期状態 x_1 と制御期間長 T とはHDD14に格納されているデフォルト値又は前回のプレイ結果値をRAM13のデータ記憶領域に格納したものであっても良いし、キーボード24から入力した値であっても良い。

【0061】

プレイヤーの初期状態 x_1 と制御期間長 T とが入力されると、DPグラフ作成器30aは、 T 期間の期待総利得を最大化するための動的計画法(DP)の問題を解くためのDPグラフを作成する(ステップS2)。例えば、想定されるプレイヤーの全状態を要素とする状態集合 S が、 $S = \{s_1, s_2, s_3, s_4\}$ で $x_1 = s_1$ の場合であれば、図4の

10

。

【0062】

次いで、DP実施器30bがDPによって T 期間のMDP問題を解くことによって、 T 期間の期待総利得を最大化する最適政策が求められる(ステップS3)。DP実施器30bは、DPグラフの末端の各ノードから順にそのノードでの最適な行動(RPGにおけるプレイヤーのコマンド選択)とそのノード以降の期待総利得の最大値を、行動決定部31

20

【0063】

即ち、各ノード毎にそのノードの時点 t (何時点目かを示す自然数)とプレイヤーの状態 x_t (t 時点目のプレイヤーの状態)とを行動決定器31aへ送ると、そのノードにおける最適な行動とそのノード以降の期待総利得の最大値とが求められて行動決定器31aから送り返される。

【0064】

その後、DPグラフの1時点目のノードまで全て解き終わったかどうか判断され(ステップS5)、解き終わっていればDPグラフの全ノードにおける最適な行動とそのノード以降の期待総利得の最大値が最適政策として出力される(ステップS6)。

30

【0065】

次に、図3及び図5に示されたフローチャートを参照して行動決定部31の動作、即ち図2におけるステップS4の動作を説明する。

【0066】

まず、最適政策算出部30のDP実施器30bから行動決定部31の行動決定器31aへ、時点 t (何時点目かを示す自然数)とプレイヤーの状態 x_t (t 時点目のプレイヤーの状態)とが入力される(ステップS41)。

【0067】

次いで、入力された時点 t とプレイヤーの状態 x_t とに応じてそのノード以降の期待総利得の最大値とそのノードにおける最適な行動とが算出される(ステップS42)。 $t = T$ の場合には、式(18)で、そのノード以降の期待総利得の最大値が求められる。

40

【0068】

【数3】

$$V(x_T, T) = \max_{y_T \in A(x_T)} \sum_{x_{T+1}} p(x_{T+1} | x_T, y_T, \theta^*) r(x_T, y_T, x_{T+1}), \quad (18)$$

ただし、 $V(x_T, T)$ は T 時点目の状態 x_T から時点 $T+1$ への状態遷移に伴う最後の1期間の期待総利得の最大値である。 $p(x_{T+1} | x_T, y_T, \theta^*)$ は遷移確率テーブル31bから読み取ったものである。 $r(x_T, y_T, x_{T+1})$ は利得テーブル31cから読み取ったものである。 $1 \leq t \leq T-1$ の場合には次の式(19)でそのノード以

50

降の期待総利得の最大値が求められる。

【 0 0 6 9 】

【 数 4 】

$$V(x_t, t) = \max_{y_t \in A(x_t)} \sum_{x_{t+1}} p(x_{t+1} | x_t, y_t, \theta^*) (r(x_t, y_t, x_{t+1}) + V(x_{t+1}, t+1)), \quad (19)$$

ただし、 $V(x_t, t)$ は t 時点目の状態が x_t という条件のもとでの、 t 時点以降の期待総利得の最大値である。本実施形態では DP を利用しているため、このように部分最適解を再利用している。 $t = T$ の場合には次の式 (20) でそのノードにおける最適な行動が求められる。

【 0 0 7 0 】

【 数 5 】

$$d^*(x_T, T) = \arg \max_{y_T \in A(x_T)} \sum_{x_{T+1}} p(x_{T+1} | x_T, y_T, \theta^*) r(x_T, y_T, x_{T+1}), \quad (20)$$

ただし、 $d^*(x_T, T)$ は T 時点目の状態 x_T において選択すべき最適な行動である。 $t = T - 1$ の場合には次の式 (21) でそのノードにおける最適な行動が求められる。

【 0 0 7 1 】

【 数 6 】

$$d^*(x_t, t) = \arg \max_{y_t \in A(x_t)} \sum_{x_{t+1}} p(x_{t+1} | x_t, y_t, \theta^*) (r(x_t, y_t, x_{t+1}) + V(x_{t+1}, t+1)), \quad (21)$$

ただし、 $d^*(x_t, t)$ は t 時点目の状態 x_t において選択すべき最適な行動である。

【 0 0 7 2 】

その後、そのノードにおける最適な行動とそのノード以降の期待総利得の最大値が最適政策算出部 30 の DP 実施器 30 b へ出力され (ステップ S 43)、前述の最適政策が出力されるのである。

【 0 0 7 3 】

以上説明したように、第 1 の実施形態によれば、DP を用いて各時点のプレイヤーの各状態において、その時点以降の期待総利得を最大化し、最終的に制御期間における期待総利得を最大にすることが保証された政策が出力されるので、プレイヤーの初期状態と制御期間長とに対して制御期間における期待総利得を最大にする政策を出力することが可能となり、その RPG に関する攻略法 (行動選択の仕方) を算出することができる。

【 0 0 7 4 】

即ち、本実施形態によれば、攻略法をコンピュータにシミュレーションさせることによって、被験者の体験データを取得することができる。例えば、プレイヤーのゲーム結果をシミュレーションすることによって、マップ上に隠されたアイテムやイベントに遭遇する割合等を把握でき、その割合を見ながら適切な隠し場所を設定する等のゲーム開発支援を行うことができる。また、算出される数理工学的に最適な攻略法と実際のプレイヤーによる攻略法との比較を行うことにより、人間が楽しいと感じるゲーム要素を工学的に把握することができる。人間が楽しいと感じるゲーム要素を確率モデル上のパラメータ設定のある種のパターンとして把握できれば、そのような要素を多く含むゲーム開発を行うことができる。また、種々の目的毎の攻略法を算出することによって、各目的に適したプレイヤーに協力するキャラクタの攻略法を容易にプログラミングすることが可能となる。

【 0 0 7 5 】

次に、各種確率分布を支配する真のパラメータ θ^* が未知である、本発明を拡張した第 2 の実施形態について説明する。

【 0 0 7 6 】

本実施形態における攻略法算出装置の構成は基本的には、第 1 の実施形態の場合と同様

10

20

30

40

50

であり、従ってその構成の説明は省略する。

【 0 0 7 7 】

真のパラメータ未知の場合を説明するために、いくつかの新たな定義を行う。 $p(\cdot)$ はパラメータの事前分布であり、既知であるとする。 Θ はパラメータ空間であり、 $\theta \in \Theta$ 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あり、 $x^t y^{t-1} = x_1 y_1 \cdots x_t$ である。

【 0 0 7 8 】

真のパラメータ既知の場合には、DPでT時点から遡りながら各時点の各状態に対して行動選択を行うが、真のパラメータ未知の場合には、DPでT時点から遡りながら各時点の各状態と1時点からその時点に至るまでの各遷移系列の組に対して行動選択を行う。

10

【 0 0 7 9 】

T時点目の状態 x_T (全ての状態の候補) と遷移系列 $x^T y^{T-1}$ (全ての遷移系列の候補) の組に対する処理は以下の通りである。

【 0 0 8 0 】

【 数 7 】

$$d_B^*(x_T, x^T y^{T-1}, T) = \arg \max_{y_T \in A(x_T)} \sum_{x_{T+1}} \int_{\Theta} p(\theta | x^T y^{T-1}) p(x_{T+1} | x_T, y_T, \theta) d\theta r(x_T, y_T, x_{T+1}), \quad (22)$$

20

$$V_B(x_T, x^T y^{T-1}, T) = \max_{y_T \in A(x_T)} \sum_{x_{T+1}} \int_{\Theta} p(\theta | x^T y^{T-1}) p(x_{T+1} | x_T, y_T, \theta) d\theta r(x_T, y_T, x_{T+1}), \quad (23)$$

ただし、 $p(\cdot | x^T y^{T-1})$ は1時点からT時点に遷移系列 $x^T y^{T-1}$ のように遷移した場合の事後分布である。

【 0 0 8 1 】

t 時点目 ($1 \leq t \leq T-1$) の状態 x_t (全ての状態の候補) と遷移系列 $x^t y^{t-1}$ (全ての遷移系列の候補) の組に対する処理は以下の通りである。

【 0 0 8 2 】

【 数 8 】

$$d_B^*(x_t, x^t y^{t-1}, t) = \arg \max_{y_t \in A(x_t)} \sum_{x_{t+1}} \int_{\Theta} p(\theta | x^t y^{t-1}) p(x_{t+1} | x_t, y_t, \theta) d\theta (r(x_t, y_t, x_{t+1}) + V_B(x_{t+1}, x^{t+1} y^t, t+1)). \quad (24)$$

30

$$V_B(x_t, x^t y^{t-1}, t) = \max_{y_t \in A(x_t)} \sum_{x_{t+1}} \int_{\Theta} p(\theta | x^t y^{t-1}) p(x_{t+1} | x_t, y_t, \theta) d\theta (r(x_t, y_t, x_{t+1}) + V_B(x_{t+1}, x^{t+1} y^t, t+1)). \quad (25)$$

40

式(22)から式(25)を用いて $d_B^*(x_1, x_1, 1)$ まで求めることによって、1時点目からT時点目までの全ての状態と遷移系列の組とに対して、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。

【 0 0 8 3 】

式(22)から式(25)には積分計算が含まれており、一般的に、積分計算は計算量が多いが、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を、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それぞれ仮定すると、積分計算は四則演算で実施することができる(Matsushima, T., Hirasawa, S., A Bayes coding algorithm for Markov models, TECHNICAL REPORT OF IEICE, IT95-1

50

, pp. 1 - 6 (1995)). 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 を選択したもとの、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

【数 9】

$$\int_{\Theta} p(\theta | x^t y^{t-1}) p(x_{t+1} | x_t, y_t, \theta) d\theta$$

の計算を以下に示す。

【0084】

【数 10】

$$\begin{aligned} \int_{\Theta} p(\theta | x^t y^{t-1}) p(x_{t+1} | x_t, y_t, \theta) d\theta &= \int_{\Theta} p(\theta | x^t y^{t-1}) p(e_i | F(mv(x_{t,2}, y_t)), \theta) d\theta \\ &= \frac{N(F(mv(x_{t,2}, y_t)), e_i | x^t y^{t-1}) + \alpha_1}{N(F(mv(x_{t,2}, y_t)) | x^t y^{t-1}) + \alpha_2}, \end{aligned} \quad (26)$$

10

ただし、

$$\alpha_1 = \alpha(e_i | F(mv(x_{t,2}, y_t))), \quad (27)$$

$$\alpha_2 = \sum_{e_j \in E} \alpha(e_j | F(mv(x_{t,2}, y_t))) + \alpha(mv(x_{t,2}, y_t) | F(mv(x_{t,2}, y_t))), \quad (28)$$

20

ここで、 $N(F(mv(x_{t,2}, y_t)), e_i | x^t y^{t-1})$ 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、 $(e_i | F(mv(x_{t,2}, y_t)))$ は $p(e_i | F(mv(x_{t,2}, y_t)), \theta)$ に対するディリクレ分布 (事前分布) のパラメータ、 $(mv(x_{t,2}, y_t) | F(mv(x_{t,2}, y_t)))$ は

【数 11】

$$1 - \sum_{e_i \in E} p(e_i | F(mv(x_{t,2}, y_t)), \theta)$$

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、事前に何も情報が無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種々の方法が研究されている。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が有名であり、本実施形態に適用する場合は、各パラメータを 0.5 に設定することに相当する。例えば、Berger, J. O., *Statistical Decision Theory and Bayesian Analysis*, Springer-Verlag, New York (1980)、繁樹算男, *ベイズ統計入門*, 東京大学出版会 (1985)、Matsushima, T., Hirasawa, S., *A Bayesian coding algorithm for Markov models*, TECHNICAL REPORT OF IEICE, IT95-1, pp. 1-6 (1995)、鈴木讓, *ベシアンネットワーク入門*, 培風館, 東京 (2009) を参照。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 (22) から式 (25) 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することができる。

30

40

【0085】

以上の説明では、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について述べた。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。しかしながら、ベイズ最適な行動選択の仕方を求めるためには、多大な計算量が必要となる。真のパラメータ既知の場合には、DPの各時点毎に式 (19) の処理を状態数分だけ実施すればよい。一方、真のパラメータ未知

50

のベイズ最適の場合には、DPの各時点毎に式(25)の処理を状態数と遷移系列の個数との積分だけ実施する必要があり、処理の回数は時点の数(t期のt)に対する指数オーダーとなる。

【0086】

そこで、本実施形態の望ましい態様として、真のパラメータ未知の場合に近似を行う例を説明する。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、敵の出現確率などの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、種々の形態が適用可能である。

【0087】

前述したベイズ最適な方法では、各時点毎にその時点までの遷移系列 $x^t y^{t-1}$ に対する事後分布によって、

【数12】

$$\int_{\Theta} p(\theta | x^t y^{t-1}) p(x_{t+1} | x_t, y_t, \theta) d\theta$$

を計算したが、近似アルゴリズムでは、時点に関係なく学習データLによる事後分布を用いて

【数13】

$$\int_{\Theta} p(\theta | L) p(x_{t+1} | x_t, y_t, \theta) d\theta$$

を計算する。具体的には、

【数14】

$$\int_{\Theta} p(\theta | L) p(x_{t+1} | x_t, y_t, \theta) d\theta$$

を真のパラメータ既知の場合の式(18)から式(21)に代入して行動選択の仕方を求める。

【0088】

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することができる。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。

【実施例】

【0089】

真のパラメータ*が既知の場合の行動選択の仕方を求めるアルゴリズムに関する第1の実施形態について実験を行った実施例を説明する。図6にこの実施例におけるマップを示す。実験結果が理解しやすいように9マスからなる小規模のマップとした。以下の表1~表5は地形の設定、確率の設定及びその他の設定を示している。

【0090】

【表1】

表1 地形の設定

$F(s m_1)$	$F(s m_2)$	$F(s m_3)$	$F(s m_4)$	$F(s m_5)$
f_1	f_2	f_3	f_2	f_2

$F(s m_6)$	$F(s m_7)$	$F(s m_8)$	$F(s m_9)$
f_3	f_3	f_3	f_3

【0091】

10

20

30

40

【表 2】

表 2 確率の設定 (その 1)

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

【0092】

【表 3】

表 3 確率の設定 (その 2)

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

10

$p(B(e_2) e_2, \theta^*)$	$p(\text{map} a_6, \theta^*)$
0.6	0.6

【0093】

【表 4】

表 4 その他の設定 (その 1)

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

20

【0094】

【表 5】

表 5 その他の設定 (その 2)

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

【0095】

以上の設定で、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。

30

【0096】

結果の一部について述べると、例えば、時点 3 (3 期) でプレイヤーの HP が 6 であり位置 s_{m_4} にいるマップモードの状態では、最適の行動選択は a_3 という上のマス s_{m_7} への移動であった。これは、プレイヤーの 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 (a_3) への移動を選択しているのである。他方、同じ時点 3 (3 期) のマップモードであっても HP が 1 であり位置 s_{m_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 s_{m_1} へ移動した。これは、プレイヤーの 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、時点 9 (9 期) に HP が 1 であり位置 s_{m_4} にいるマップモードの状態では、行動 a_1 を選択して、弱くて報酬の小さい敵が出現する右の s_{m_5} へ移動した。これは、プレイヤーの HP に余裕はないが、残りの期間にも余裕がないため、弱い敵が出現する s_{m_5} への移動を選択しているのである。

40

【0097】

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。

【0098】

次に、真のパラメータが未知の場合の行動選択の仕方を求めるアルゴリズムに関する第 2 の実施形態について実験を行った実施例を説明する。真のパラメータ既知の場合の実施例と同じ設定のもとで、真のパラメータ未知の場合の近似アルゴリズムを適用した。学習

50

データとして、各確率分布毎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、各時点の各状態における行動選択が真のパラメータ既知の場合と比較して一致するかどうか調べた。各確率分布の学習データ数を10、100、1000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 of 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の場合で約88%、学習データ数が100の場合で約94%、学習データ数が1000の場合で約96%であった。

【0099】

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の計算量と同程度とすることができる。また、少ない実験例ではあるが、学習データの増加に伴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。

10

【0100】

以上述べた実施形態は全て本発明を例示的に示すものであって限定的に示すものではなく、本発明は他の種々の変形態様及び変更態様で実施することができる。従って本発明の範囲は特許請求の範囲及びその均等範囲によってのみ規定されるものである。

【符号の説明】

【0101】

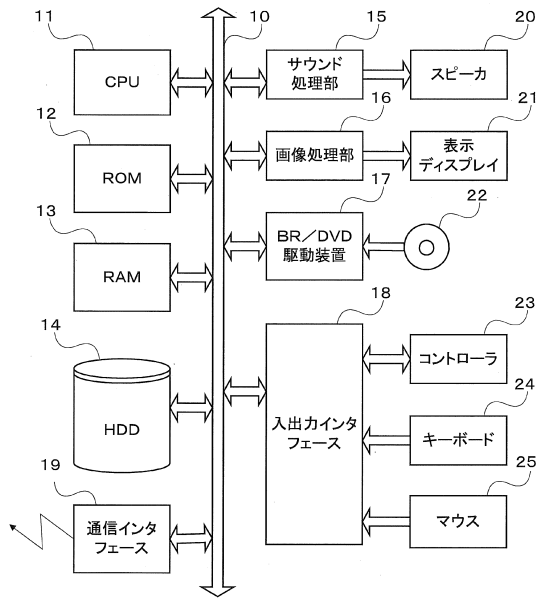
- 10 バス
- 11 CPU
- 12 ROM
- 13 RAM
- 14 HDD
- 15 サウンド処理部
- 16 画像処理部
- 17 BR/DVD駆動装置
- 18 入出力インタフェース
- 19 通信インタフェース
- 20 スピーカ
- 21 表示ディスプレイ
- 22 BR/DVD/CD
- 23 コントローラ
- 24 キーボード
- 25 マウス
- 30 最適政策算出部
- 30a DPグラフ作成器
- 30b DP実施器
- 31 行動決定部
- 31a 行動決定器
- 31b 遷移確率テーブル
- 31c 利得テーブル

20

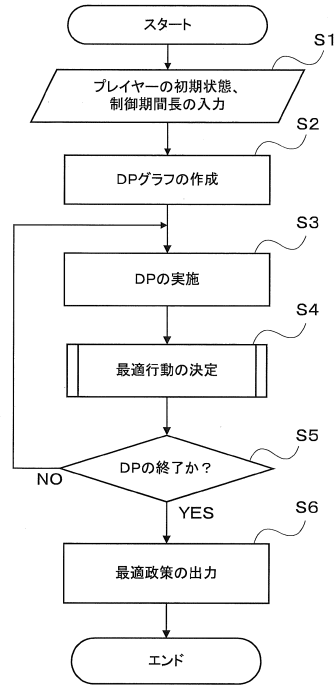
30

40

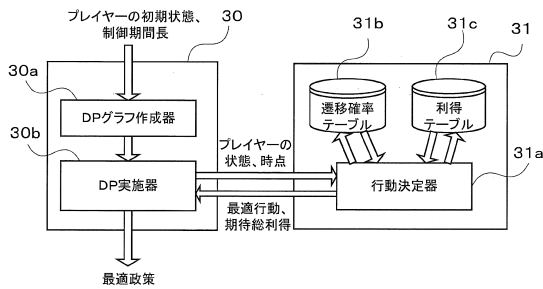
【図1】



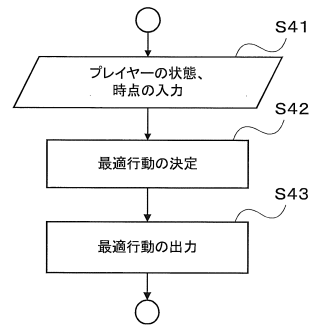
【図2】



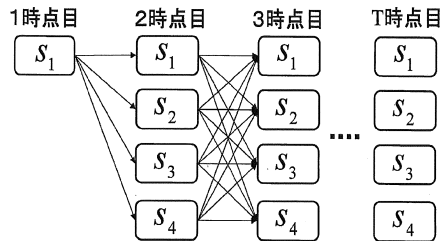
【図3】



【図5】



【図4】



【図6】

sm_7	sm_8	sm_9
sm_4	sm_5	sm_6
sm_1	sm_2	sm_3

フロントページの続き

(56)参考文献 [INTERNET ARCHIVE WayBack Machine] 遺伝的アルゴリズムの応用, [o n l i n e] , 2 0 0 9 年 5 月 2 7 日, 検索日: 2 0 1 6 年 7 月 1 5 日, U R L , <https://web.archive.org/web/20090527023150/http://www6.plala.or.jp/mnagaku/cmaga/ac19999/ga3.html>

高木 幸一郎, 他, ロールプレイングゲーム (R P G) 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, 日本, 社団法人情報処理学会, 2 0 0 1 年 3 月 1 6 日, V o l . 2 0 0 1 N o . 2 8 , 第 3 1 - 3 8 頁

Chen Haoyang et. al, Searching Optimal Combat Strategies of On-line Action Role-playing Game using Discrete Competitive Markov Decision Process, ゲームプログラミングワークショップ 2 0 1 1 論文集, 日本, 情報処理学会, 2 0 1 1 年 1 0 月 2 8 日, 第 2 0 1 1 巻, 第 6 号, p . 1 2 0 - 1 2 7

(58)調査した分野(Int.Cl., DB名)

A 6 3 F 1 3 / 0 0 - 1 3 / 9 8