情報処理学会論文誌　Vol. 52　No. 12　3817–3827 (Dec. 2011)

Ms. Pac-Manにおけるモンテカルロ木探索

池畑　望[†1]　伊藤毅志[†2]

2007年よりIEEEのCIGシンポジウムの中でMs. Pac-Manの自動操作を競う大会が開かれている．この大会以来，Ms. Pac-Manは，デジタルゲームAIの研究対象として注目を集めつつある．これまでの大会では，知識ベースを用いた古典的な手法によるAIが最も良い成績を収めているが，その性能には限界が見え始めており，知識ベースに代わる新しいアプローチが求められている．そこで，本稿では囲碁で成功したモンテカルロ木探索によるMs. Pac-Manの自動操作システムを実現し，その有効性を検証した．モンテカルロ木探索は乱数によって生成された未来局面についてのシミュレーションを繰り返すことで，専門的知識に頼らずに期待値の高い次の手を求めることができる．性能評価実験ではモンテカルロ木探索による自動操作システムは過去にMs. Pac-Man Competitionに参加したすべてのプログラムよりも優秀な成績を示し，Ms. Pac-Manにおけるコンピュータの世界記録を上回る結果を得た．

Monte-Carlo Tree Search in Ms. Pac-Man

Nozomu Ikehata[†1] and Takeshi Ito[†2]

The competition of controller program for Ms. Pac-Man has been held in the CIG symposium of IEEE every year from 2007. Since this competition was held, Ms. Pac-Man has become to an attractive subject of research on digital game AI. In the competition by this year, the classical AI method by using the knowledge base has gotten the best result. But, since the improvement by this method is becoming a limit, a new approach is required. In this paper, we realized the Ms. Pac-Man controller by the using Monte-Carlo tree search which is effective on Go, and examined the effectiveness. By repeating the simulation about the future phase generated with the random number, the Monte Carlo tree search can select the next move with a high expected value, without depending on professional expertise. In an evaluation experiment, the controller by the Monte Carlo tree search showed results more excellent than all the programs which participated in Ms. Pac-Man Competition in the past.

1. はじめに

Ms. Pac-Man[⋆1]はデジタルゲームとしてはルールが単純である一方で，優秀なプレイを行うためには十分に複雑な知的戦略が必要となる．この特性はAI技術発展のための良質なテストベッドになると考えられている．Ms. Pac-ManのAI Competitionでは，自動操作によって人間のエキスパートが持つMs. Pac-Manの最高得点を塗り替えることを最終目標として，プログラムどうしに得点を競わせている[1)]．プログラムと人間のプレイ環境を平等にするために，プログラムはゲームの内部情報を利用することが禁止されており，動作画面のスクリーンキャプチャを通して状況認識・判断を行わなければならない．これはプログラムにとって非常に大きな制約となる．なぜなら，Ms. Pac-Manのようなデジタルゲームでは，画面に表示されていない情報（たとえば，ゲーム内の独自の重力加速度や床の滑り具合等，人間であれば，ゲームをプレイしながら感覚で把握していくような事柄）が多く，これらの正確な値をリアルタイムに獲得することは非常に困難であるからである．このような非明示的なルールが多数埋め込まれているのがデジタルゲームの特徴であり，AIがデジタルゲームの世界を単純にモデル化できない原因の1つとなっている．

2007年より行われているMs. Pac-ManのAI Competitionの歴史を振り返ると，これまでに様々なAIテクニックがMs. Pac-Manの自動操作システムに適用されてきたが，すべてのコンペティションにおいてルールベースの手法を採用したプログラムが優勝している．これは，高い安定性と高速な判断を利点とするルールベースの性質がMs. Pac-Manのゲーム性にうまく適応した結果であると考えられる．その反面，ここ数年，ルールベースによるMs. Pac-Manの自動操作システムの成績は伸び悩んでおり，ルールに頼った手法の限界が見え始めている．ルールベースが現在かかえている問題の1つに，未知の状況への対応力の低さがある．これは，条件式で記述することが可能な明瞭な状況には強いが，条件を記述し難い，あるいは記述不可能な曖昧な状況には非常に弱いという問題である．Ms. Pac-Manにおいて回避条件を記述することが困難な状況の1つに挟み撃ちがある．挟み撃ちとは，複数のゴーストにミズパックマンが移動可能なすべての経路を塞がれてしまうこと

†1 株式会社コナミデジタルエンタテインメント
Konami Digital Entertainment Co., Ltd.
†2 電気通信大学
The University of Electro-Communications
⋆1 文中でMs. Pac-Manと表記する場合にはゲームを，ミズパックマンと表記する場合にはキャラクタを示す．

であり，ミズパックマンの逃走経路が完全に絶たれた状態のことを指す．これは，将棋における詰みのような局面にあたり，どのような入力を与えたとしても，ゴーストとの接触を回避することができなくなってしまう．Ms. Pac-Man において，高水準での生存率を確保するためには，この挟み撃ちの状態は何よりも避けるべき事態である．挟み撃ちの発生を予測し回避するためには，状況の完全な先読みと評価が必要となるが，非明示的な情報が多い Ms. Pac-Man に対して，将棋等のチェスライクゲームで用いられているようなゲーム木探索を適用することは困難である．本研究では Ms. Pac-Man における挟み撃ち回避問題を解決し，ミズパックマンの生存率を高めることを目標とする．そのためのアプローチとして，コンピュータ囲碁で成功を収めたモンテカルロ木探索アルゴリズムである UCT（UCB applied to Trees）を用いた．UCT はシミュレーションをベースにした木探索手法の 1 つで，ランダムシミュレーションによる平均報酬とノードの探索回数を考慮して，見込みのある手に対して多くの探索を行う手法である．Ms. Pac-Man のようなリアルタイムゲームでは，不明瞭な情報の多さから，すべての状況を考慮した完全なゲーム木を形成することが困難であるが，単純化された類似局面のシミュレーションを繰り返すことで，それを補うことができると考えられる．

2. Ms. Pac-Man と Competition

Ms. Pac-Man は，1982 年にゼネラルコンピュータ社によって開発・販売された業務用ゲームソフトウェアである．その原型は，マサチューセッツ工科大学の卒業生が開発した Pac-Man のコピー作品であるが，完成したゲームの質が非常に高かったため，後にナムコに派生的著作物として許諾されることとなった．国内では未発売であるが，国外では Pac-Man を凌ぐ大ヒット商品となり世界的な知名度は非常に高い．

Ms. Pac-Man はドットイートと呼ばれるジャンルのゲームである．ドットイートとは，敵の追撃を回避しながら，迷路内に敷き詰められた目標（たいていはドットで表現される）を獲得することを目的とするリアルタイムアクションゲームの総称である．

Ms. Pac-Man のルールは，国内で有名な Pac-Man とほとんど同じであるが，迷路の形状や敵ゴーストの動きに決定的な違いがある．Pac-Man では，敵ゴーストの動きには規則性があり，こちらのパックマンの動きが一定であれば，つねに同じ動作をするが，Ms. Pac-Man では，敵ゴーストの動きは確率的にランダムに動く．また，その動きのアルゴリズムは公開されていない．その他，公開されていない情報が多く，キャラクタの移動速度や角を曲がる際の加速や減速，ドットイート時の減速の程度等，その正確な値を知ることができない．Ms. Pac-Man は，通常の Pac-Man に比べて，不確定性，不完全情報性の多いゲームであり，それがゲームとしての難しさを高めている．

Ms. Pac-Man Competition は，Ms. Pac-Man の自動操作を題材にした AI Competition である．参加者は，Ms. Pac-Man を自動操作するプログラムを開発して，その性能を競いあう．2007 年，2009 年には，CEC（IEEE Congress on Evolusionary Computation）で，2008 年には，WCCI（IEEE World Congress on Computational Intelligence）で，2009 年以降は CIG（IEEE COnference on Computational Intelligence and Games）で毎年行われている．Ms. Pac-Man Competition のルールは以下の 5 点である．

1. プログラムに Ms. Pac-Man を 10 回プレイさせた際の最高得点を競う．
2. プログラムは Ms. Pac-Man のスクリーンキャプチャからのみ情報を獲得する．
3. プログラムは Ms. Pac-Man の内部情報のいっさいの利用を禁ずる．
4. プログラムは約 15 FPS の頻度で Ms. Pac-Man に入力を送信する．
5. プログラムは Ms. Pac-Man のフレームレートの意図的な操作を禁ずる．

以上のルールは，コンピュータと人間のプレイ環境をできるかぎり平等にするために設定されたものである．近年の Ms. Pac-Man における自動操作プログラムのほとんどは，Ms. Pac-Man Competition の大会規約を前提として開発されたものであり，本研究においてもそれにならうものとする．

3. 先 行 研 究

Pac-Man の自動操作システムに関する最古の研究は Koza によって行われたものである．彼は自らの著書 “Genetic programming: on the programming of computers by means of natural selection” の一節，“TASKPRIORITIZATION” において，タスク優先順位についての説明の一例として，Pac-Man の行動制御について言及している[2)]．彼は “ゴーストが接近してきた場合に逆方向に進め”，“ゴーストが周囲にいない場合には最も近い餌に最短経路で進め” 等の有効と思われる条件と行動のセットを論理記述方式である S 式で記述することで，パックマンの行動を制御できると考えた．結果として，S 式が埋め込まれたエージェントは，単体のゴーストを回避するような行動を実現したものの，複数体のゴーストとの接触を回避することができない低性能なものとなった．彼はその原因について，自らが記述した S 式には不備があり，最適ではなかったためであると考察している．彼の研究は，単純なルールのみで Pac-Man を制御することが難しいことを示唆する結果となった．

2000 年代に入ると，ルールベースの AI は下火になり始め，評価関数型の手法が検討され

始めた．Simon はニューラルネットワークを用いて Ms. Pac-Man の位置評価関数を自動的に学習しようと試みた[3]．ゴーストまでの距離，最近傍交差点までの距離，最近傍餌までの距離等を入力とし，現在位置から数フレーム以内にパックマンが到達する可能性のある迷路上の座標を出力とする多層パーセプトロンを構成した．調整されたニューラルネットワークからは，ゴーストに囲まれたらパワー餌をとりに行く，いじけ状態のゴーストを追跡する等の人間プレイヤも行うような有効な戦略の自動的な発生が確認できたものの，いじけ状態でないゴーストに近づいてしまう等の不安定な動作も多く見られた．

2000 代も後半になり，Ms. Pac-Man Competition が開催されるようになると，Ms. Pac-Man を題材とした研究がさかんに行われるようになった．この頃になると，現在と同じく Ms. Pac-Man Competition のレギュレーションに則り，スクリーンキャプチャシステムを備えるエージェントが開発されるようになった．Wirth らはルールのみで Ms. Pac-Man を制御することは困難であると考える一方で，強化学習やニューラルネットワーク等の機械学習手法よりも安定な手法を追及して，パックマンの行動制御に影響マッピングを用いる制御方法を提案した[4]．これは，餌やゴースト等のオブジェクトが周辺領域に与える正負の影響度を定義し，影響度をもとに迷路状の各座標に対して評価値を設定するものであった．彼らのエージェントは，ミズパックマンが比較的安全な状況にいる場合には，ゴーストとの距離を適度に保ちつつ，効率的に餌を獲得するような効果的な判断を行うことができた．しかしながら，複数体のゴーストに囲まれてミズパックマンが危険な状況に陥った際には，ゴーストの接触を回避できない場面が多くあった．これは，ゴーストの動きに対する先読みがまったく行われていないために，影響マップにより正の評価値を与えられた座標であっても，数フレーム先での安全がまったく保障されていないためであった．

Robles らは，Wirth らの研究の結果から，現在の状況だけでなく，未来の局面も考慮に入れる必要があると考えた．彼らは，将棋やチェス等の静的思考ゲームで広く採用されているゲーム木探索によって，ミズパックマンの現在座標から移動可能なすべての経路の危険度を評価することを試みた[5]．迷路を一定サイズの升目で区切ったものをノードとし，ミズパックマンの現在位置をルート，ミズパックマンの移動経路を枝とする深さ 40 のゲーム木を構築した．そして，構築されたゲーム木にゴーストが含まれているかどうかを調査し，ゴーストがゲーム木に含まれているならば，分枝点（交差点）にミズパックマンがゴーストよりも先に到達できるかどうかを計算して，その経路が本当に安全であるかを評価した．しかしながら，探索の深さが固定であるため，ゲーム木に含まれないゴーストの動きがまったく考慮されないことや，アルゴリズムの精度が分枝点到達の計算の信頼性に依存してしまうこと等，いくつかの不安要素があった．

近年では，蟻コロニー最適化を用いて Ms. Pac-Man の経路選択を行った Emilio の研究[6]や，囲碁で成功したモンテカルロ法を Ms. Pac-Man に適用した Tong の研究がある[7]．しかしながら，これらの研究では依然としてルールベース型の AI の性能には届いてはおらず，前述したルールベース型 AI がかかえる問題の根本的な解決には至っていない．

4. 挟み撃ち

ルールベースでは困難であるが，高得点を獲得するために Ms. Pac-Man で解決するべき重要な課題として，挟み撃ちの回避という問題がある．1 章で述べたとおり，挟み撃ちとはミズパックマンが移動可能なすべての経路をゴーストに塞がれてしまい，ミズパックマンの逃走経路が完全に絶たれてしまう状態のことを指す．挟み撃ちを予測し回避することは，ミズパックマンの生存率を飛躍的に高めることにつながる．

非公開とされている情報が多い Ms. Pac-Man においては，完全なゲーム木を構成することが事実上不可能であるため，未来局面の予測を必要とする挟み撃ちの回避はきわめて難しい．しかしながら，挟み撃ちの完全な回避は困難であっても，挟み撃ちの発生確率を見積もり，その確率の高い局面を回避することで，ゲーム内での挟み撃ちの発生確率を低下させることができると考えられる．そこで，本研究では挟み撃ちの完全な回避を目標とするのではなく，挟み撃ちの発生確率が高い局面を回避することで生存率を高めるという方針でシステムを設計する．

5. UCT

2000 年代半ば，将棋やチェスのプログラムがアマチュア上級者と比較しても遜色のないレベルにまで達した一方で，コンピュータ囲碁の棋力はアマチュア級位者にも劣っているレベルで低迷していた．その背景には，囲碁には将棋やチェスとは異なり駒の損得のような概念がなく，1 つ 1 つの石の価値がすべて等価であるため，局面を評価する特徴要素を定義しにくいことや，探索空間が将棋やチェスに比べてきわめて膨大であることから，静的評価関数とゲーム木探索を用いた従来のアプローチでは強いプログラムを作ることが困難であるという問題があった．そんな折，2005 年頃からモンテカルロ法を採用したコンピュータ囲碁が成功を収めたという報告がなされ始めた．モンテカルロ法は乱数を用いたシミュレーション（プレイアウトと呼ぶ）を多数行い手を選択する手法の総称であり，解析的に解くことができない問題でも，十分に多くの回数のシミュレーションを繰り返すことで近似的な解

を求めることができる．囲碁におけるモンテカルロ法の最も基本的な流れは以下のようなものである．

1．着手可能な場所に乱数で石を配置する．
2．1．を白黒交互に繰り返す．打つ場所がなくなってパスが 2 回続けば終局．
3．終局局面に点数を付ける（代表的なものとしては勝ちで +1，負けで 0）．
4．1 から 3 を何度も繰り返して点数の平均値を求める．

これは，現在局面から終局までの大量のシミュレーションを行って，最も点数が高い手を選択するという手法である．囲碁には着手数が増えるに従い終局へと収束する性質があり，ほとんどの場合にシミュレーションを一定手数以内で終了させることができるため，シミュレーションベースの手法と非常に相性が良い．また，コンピュータ囲碁の棋力のボトルネックの 1 つは評価関数の作成の難しさであるが，モンテカルロ法では候補手の評価に評価関数を必要としないため，その問題は解決される．しかしながら，モンテカルロ法には深さが 2 以上の木に対して最善手を返す保証がないという欠点がある．たとえば，相手が悪手を打てば得であるが正しく応じられると損をするような局面があるとしたとき，相手にとっての正解の手が少なければシミュレーション中に正しい手が選択される確率は当然低くなるが，実際の対局では相手の棋力次第で正しく応じられる可能性は大いにありうる．このように，モンテカルロ法では，相手の棋力を考慮することができず，相手がミスをすることに期待して強引な手を打ってしまうという危険があった．

これを解決するために，2006 年に Coulom らによってモンテカルロ法を木探索に拡張したモンテカルロ木探索が提案された[8]．原始的なモンテカルロ法からの変更点は 2 つで，1 つが有利な手により多くのシミュレーションを割り当てることであり，もう 1 つが，シミュレーション回数が閾値を超えた際に木を成長させることである．この手法は，原始的なモンテカルロ法の問題を解決したように思われたが，ここで障害となったのが，各候補手に対するシミュレーションの割り振り方であった．そもそも，静的評価関数を作成することが困難なコンピュータ囲碁において，有利な手を判断すること自体が難しい問題であった．この問いに対する画期的な解答を与えたのが，同年 Kocsis らによって考案された UCT（UCB applied to Trees）であった[9]．UCT は多腕バンディッド問題における効率的な方策である UCB1 をゲーム木探索に拡張したものである．ゲーム木の各ノードにおいて以下の UCB1 式で求められる UCB（Upper Confidence Bounds）値が最大になる子ノードが探索される．

$$X_i + C\sqrt{\frac{\ln T}{T_i}} \tag{1}$$

X_i は子ノード i の平均報酬，T_i は子ノード i の探索回数，T は親ノードの探索回数，C はバランスパラメータである．したがって，未探索の候補や将来有望な候補により多くのシミュレーションが割り当てられる．UCT ではシミュレーションの終局時の結果を評価値として用いることができるため，評価関数を必要としない．そのため，戦略モデルや特徴要素を設計する必要がなく，新しいゲームや研究対象としての歴史が浅く知識の蓄積が少ないゲームにも有効に作用する．

性能の停滞に陥っていた囲碁プログラムは，UCT の登場によりその棋力を大きく向上させた．UCT は囲碁での成功を受けて，他の多くの静的思考ゲームに応用された．そして，そのほとんどで従来手法よりも有効な成果をあげ，ゲームを問わない汎用性の高さを示した．しかし，UCT のリアルタイムゲームへの応用は事例が非常に少なく，RTS（Real Time Strategy）に対する研究がいくつか報告されているのみであり[10),11)]，よりリアルタイム性の高い Ms. Pac-Man のようなアクションゲームに応用された研究はこれまでになかった．

6. 提案手法

6.1 評価対象

迷路におけるクロスポイント（T 字路と交差点）とクロスポイントをつないだものを，「C 経路」と定義する．C 経路とは，その途中に分岐点を持たない，迷路上の一本道のことである．Ms. Pac-Man では C 経路がきわめて重要な意味を持つ．Ms. Pac-Man においてプレイヤが行える操作は実質的にミズパックマンの移動のみであり，T 字路や交差点において，ミズパックマンが次に選択する C 経路次第では，複数体のゴーストの挟み撃ちによってミスが確定してしまう状況も多い．クロスポイントにおける C 経路の選択には細心の注意を払う必要がある．「ミズパックマンの最近傍クロスポイント 1（ミズパックマンの現在位置がクロスポイントだった場合には，これを含まない）」，「クロスポイント 1 と連結しているクロスポイント 2」を結んだ C 経路に対して，ミズパックマンの現在位置をつなげたものを「ミズパックマンと接続する経路」と定義する．これは，近い将来，ミズパックマンが通過する可能性のある短期的な移動経路を表したものである．

6.2 ゲーム木

図 1 は本研究で用いる Ms. Pac-Man のゲーム木である．ノードはクロスポイント，枝は C 経路に相当する．ルートノードはミズパックマンの現在位置，子ノードはミズパックマンが到達する最初のクロスポイント，孫ノードは 2 番目に到達するクロスポイントである．ルートノードから末端ノードまで木を降りると，ミズパックマンと接続する経路の 1 つを

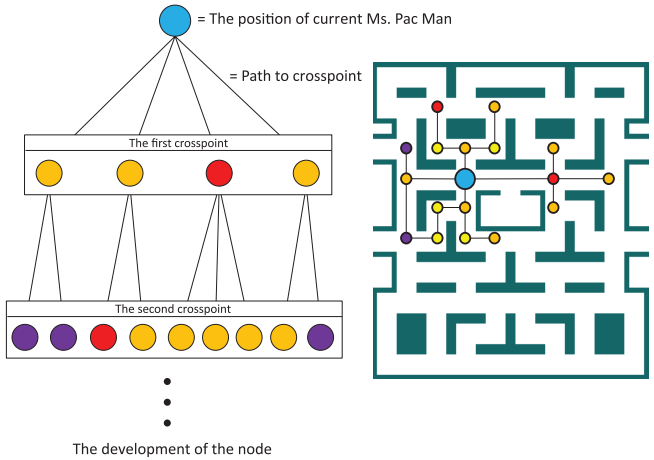

図 1　Ms. Pac-Man におけるゲーム木
Fig. 1　A game tree for Ms. Pac-Man.

ミズパックマンが選択したことになる．木を降りている間は，ミズパックマンとゴーストが交互に同じ距離を移動する．ミズパックマンの移動経路は通過ノードに従った決定的なものであるが，ゴーストの移動経路はゲーム木に束縛されず，非決定的に選択される．ゴーストの動きがゲーム木で考慮されないのは，ゴーストの移動経路までゲーム木に含めようとすると，状況の数が膨大になり，計算コスト上，浅い深さまでしか探索を行えなくなってしまうためである．すなわち，ルートノードから同じ枝やノードを経由して末端ノードに到達した場合，ミズパックマンの移動経路は同じであるが，ゴーストの移動経路は異なっている可能性がある．ゲーム木を降りる際のキャラクタの移動は，現実のゲーム空間上ではなく，次節で説明する離散空間上で行われる．ミズパックマンが移動した経路の餌は現実のゲームと同じく消費される．末端ノードの到達回数が閾値を超えると，その末端ノードが展開される．このとき，新たに生成される子ノードは，末端ノードから到達できる親ノードを除いたすべてのクロスポイントである．

ゲーム木が構築されると，UCT が開始される．UCT の最初のプロセスは，ゲーム木を降下していき，末端ノードまで到達することである．ゲーム木の降下中には，各節点において，最も UCB 値が高い子ノードが選択される．UCB 値はノード i に対して以下のように計算される．

$$UCB(i) = \begin{cases} n & (T_i = 0) \\ X_i + C\sqrt{\frac{\ln T}{T_i}} & (T_i > 0) \end{cases} \tag{2}$$

ノードの来訪回数 T_i が 0，すなわち初回訪問時には，十分に大きなランダム数 n が UCB 値に割り当てられる．それ以外の場合では，UCB1 式に基づいて，UCB 値が決定される．UCB1 式で用いられる平均報酬 X_i には，ミズパックマンの生存回数の平均値，すなわち生存率を用いる．

6.3　シミュレーション

ゲーム木降下中にミズパックマンがゴーストに接触することなく末端ノードに到達するとシミュレーションが開始される．シミュレーションの目的は，試行に対する報酬を獲得することである．シミュレーション中は，ゲーム木降下中と同じく離散化された空間上でミズパックマンとゴーストが交互に非決定的な移動を行う．

6.3.1　離散空間のルール

シミュレーション中には，ミズパックマンとゴーストは Ms. Pac-Man の実際のルールとは異なる独自のルール下において交互に行動する．これらのルールは，オリジナルの Ms. Pac-Man のキャラクタの挙動を，空間の離散化による利点を阻害しない範囲で考慮しつつ，思考アルゴリズムを実行するにともない，1 度の更新にかかる計算コストが十分に少なくなるように，ゲーム進行を単純化するためのものである．離散空間で適用されるルールは以下のようなものである（P：ミズパックマン，G：ゴースト）．

(1) 離散化空間は一辺 8 ピクセルの正方形を 1 つのグリッドとして取り扱うことで解像度を下げ，さらにゲームの挙動を単純化したものである．
(2) 離散化空間におけるキャラクタの移動単位はグリッドである．
(3) P・G は C 経路上では直進する（逆走しない）．
(4) P・G はクロスポイントでのみ移動方向を決定する．
(5) P は餌を一定個数獲得するたびに次サイクルの更新をスキップする．
(6) P はコーナーを一定回数曲がるたびに更新を連続して行う．
(7) P がパワー餌を獲得すると G はいじけ状態となり逆走する．
(8) いじけ状態の G は 2 回行動するたびに次サイクルの更新をスキップする．
(9) 1 度いじけ状態になった G のいじけ状態は解除されない．
(10) 1 度撃退された G は復活しない．
(11) 巣から新たな G は出現しない．
(12) ボーナスアイテムは出現しない．

6.3.2　ミズパックマンの移動

シミュレーション中には各キャラクタは一定の行動方針に基づいて，非決定的な移動を行う．ミズパックマンの行動方針は，クロスポイント上の経路選択において，ゴーストの回避を考えた場合に明らかに危険な C 経路を選択肢から排除し，C 経路上での単体のゴーストとの接触を避けることで，挟み撃ち以外の要因でゴーストと接触しないようにすることである．これを実現するために，次の 4 つのルールを設定する．

(1) クロスポイントでの経路選択時に，直前に移動していた C 経路へは戻らない．

(2) クロスポイントでの経路選択時に，移動可能な C 経路上にいじけ状態でないゴーストが存在し，かつ，ミズパックマンに接近する方向を向いている場合，その C 経路を移動経路の候補から排除する．

(3) (1)，(2) の方針により C 経路が一意に決定しない場合には，残された C 経路の中から一様な確率で C 経路を選択する．

(4) C 経路上で，ミズパックマンの前方一定グリッド数以内にいじけ状態でないゴーストが存在する場合，前述の離散化空間のルールの例外として，ミズパックマンの向きをその場で反転させる．

6.3.3　ゴーストの移動

Ms. Pac-Man では迷路上に残された餌の数に反比例してゴーストがミズパックマンに近づきやすくなる攻撃的な性質が設定されている．この性質を再現するために，離散空間におけるゴーストの移動経路選択の際に，ゴーストを確率的にミズパックマンに近づけることを考える．ゴーストがミズパックマンに近づく確率 P を以下の式で定義する．

$$P(G_{type}) = A(G_{type}) \times \frac{\mathrm{AP} - RP}{AP} \tag{3}$$

上記の式において，AP は迷路上に配置されたすべての餌の数，RP は迷路上に残された餌の数を表す．$A(G_{type})$ はゴーストの攻撃性であり，ゴーストの色に応じて**表 1** の値が代入される．クロスポイント上において，ゴーストは最短で目標ポイントに近づくような C 経路を P の確率で選択する．ゴーストが目標とするポイントは，ミズパックマンへの接近の仕方に応じて 2 種類設定されている．1 つが，「ミズパックマンの進行方向にある最近傍クロスポイント，またはコーナーポイント」である．このポイントを目標とすることで，ゴーストは，ミズパックマンの前方から回り込むような形の接近を行う．もう 1 つは，「ミズパックマンの後退方向にある最近傍クロスポイント，またはコーナーポイント」である．このポイントを目標とした場合には，ゴーストは，ミズパックマンを後方から追い込むような形で接近する．ゴーストの種類によってゴーストの接近の仕方が異なるのは，離散空間において，2 匹以上のゴーストによる挟み撃ちが発生する確率を高くするためである．

表 1　ゴーストの攻撃性と接近時の目標ポイント
Table 1　Aggressiveness of the ghosts and their target points.

ゴーストの色	攻撃性	目標ポイント
赤	0.9	パックマンの移動方向
オレンジ	0.8	パックマンの逆移動方向
ピンク	0.7	パックマンの移動方向
水色	0.6	パックマンの逆移動方向

6.3.4　シミュレーションの終了条件

以下のいずれかの条件が満たされるとシミュレーションは終了する．

- ミズパックマンがいじけ状態でないゴーストに接触した．
- ステージ上のすべての餌を獲得した．
- シミュレーションを開始してから一定数のサイクルが経過した．

囲碁における UCT では，終局（黒と白がお互いに打つ場所がなくなる）までシミュレーションが行われる．囲碁では，1 局にかかる手数は多くても 500 手ぐらいであり，終局まで試合を続けてもそれほどの時間はかからない．一方で，Ms. Pac-Man は 1 ステージを終了するまでに要する時間は数分程度であり，その時間に消費されるフレーム数は膨大であるため，シミュレーションを終局まで行うことは現実的に不可能である．そこで，一定数のサイクルでシミュレーションを打ち切る必要がある．

6.3.5　報　　酬

シミュレーションの結果から，以下の報酬を獲得する．

- 生存値（ミズパックマンが生存したかどうかの二値，0 または 1）
- 餌の獲得数（0 から 1）
- ゴーストの撃退数（0 から 1）

6.4　ゲーム木の更新

終了条件が満たされてシミュレーションが終了すると，獲得した報酬をもとにゲーム木の各ノードが持つ情報が更新される．ノードの更新は，通常のゲーム木探索と同様に末端ノードからルートノードに向かって伝播する．囲碁における UCT での親ノードの更新の仕方は，子ノードの情報の平均値を親ノードに渡すやり方が一般的であるが，提案手法の場合には，子ノードの中で最も生存率が高いノードが持つ情報を親ノードに伝播させるものとする．すなわち，子ノード A の生存率が 30%であり，子ノード B の生存率が 40%であったならば，親ノードには子ノード B の情報が与えられる．このような伝播方法をとった理由は，動作実験の段階で平均値を伝播させた場合に，3 本の C 経路のうち 2 本は危険であるが 1 本は安全であるようなケースで，生存率を平均化してしまうことで，安全な経路の存在が薄まってしまい，最終的に選択されなくなってしまう現象が確認されたからである．囲碁の場合では石 1 つの配置が異なったところで勝率にはそこまでの影響はないが，Ms. Pac-Man の場合には移動経路が少し異なるだけでも生存率は大幅に変わってしまう．複数のゴーストに囲まれた場合には，逃走のための正しい経路が 1 つしかない場合も少なくないため，最大生存率を伝播させる方法がより適切であると考えた．

6.5　経路の評価

シミュレーション回数が閾値を超えると，UCT は終了する．このとき，ゲーム木のミズパックマンと接続する経路の先端ノード（ルートノードの孫ノード）を比較して，最適な移動経路を決定する．ポイントとなるのは，本手法の離散空間が，4.3 節のルールと移動方針によって，ミズパックマンの死亡原因のほとんどを挟み撃ちに限定するように設計されていることである．そのため，「シミュレーション中の生存率 ≒ シミュレーション中で挟み撃ちが発生しなかった確率」という図式が成立している．すなわち，ある経路の生存率 $= 0.620$ であれば，挟み撃ちの発生率 $\fallingdotseq 0.380$ と考えることができ，求められた移動経路は，他の移動経路に比べて挟み撃ちが発生しにくい，安全な経路であると評価することができる．

6.6　戦　　略

Ms. Pac-Man では，報酬の最大化の過程において，囲碁の「勝利」のような単一の指標のみを考慮することは許されない．なぜなら，Ms. Pac-Man は相手に勝利することを目標とするゲームではないので，囲碁の「勝利」に該当するような絶対の指標がないからである．獲得得点の平均値を最大化することが一見良いように思えるが，ミズパックマンにおいては，獲得得点が高いことと，賢い動きを行えることは必ずしも同一ではない．瞬間的に高得点を獲得することができたとしても，そのすぐ後にゴーストと接触してしまっては意味がないのである．Ms. Pac-Man の主目標は高得点を獲得することで相違ないが，サブ目標としてより細かく「ゴーストと接触しない」「餌をすべて食べる」「ゴーストを多く撃退する」の 3 つの方針を考える必要がある．

本手法では，現在状況に対して，パックマンが何を考慮して行動すべきかを定めた「戦略」の概念を導入する．戦略はサブ目標を達成するためのものであり，ミズパックマンに戦略にあった行動をとらせるようにする．「生存」時にはゴーストに接触しないことを重視し，「食事」時には餌を多く食べることを重視する．また，「追跡」時にはよりゴーストを多く撃退することを重視する．

戦略は UCT の評価値とゴーストの状態に応じて切り替えられる．たとえば，現在の戦略が「生存」であるとき，UCT で経路を評価した結果，すべての経路の生存率が一定閾値以上であれば，ミズパックマンは比較的安全な状態であると考えられる．このとき，戦略を「食事」に変更して，餌をできる限りたくさん獲得することを重視する．また，ミズパックマンがパワー餌を獲得し，ゴーストがいじけ状態になったのならば，ゴーストを 1 匹でも多く撃退するために戦略を「追跡」に変更する．このように戦略は，ミズパックマンが置かれている状況が変化したときに別の戦略へと遷移する．戦略の遷移条件を以下に示す．

前フレームの生存率が一定閾値以上ならば，戦略を食事に変更する．
前フレームの生存率が一定閾値未満ならば，戦略を生存に変更する．
ゴーストがいじけ状態であり，かつ最近傍ゴーストが一定距離未満ならば，戦略を追跡に変更する．
ゴーストがいじけ状態であり，かつ最近傍ゴーストが一定距離以上ならば，戦略を食事に変更する．
すべてのゴーストがいじけ状態ではないならば，戦略を生存に変更する．

UCT による評価の有効性を高めるために，戦略に応じて，シミュレーション中のパワー餌やゴーストに触れた際の挙動を変更することを考える．たとえば，パワー餌の効果時間中にもかかわらずシミュレーション中で新たにパワー餌を獲得することは非効率的な行動である．そこで，戦略が「追跡」であるとき，ミズパックマンがパワー餌に接触した場合には，ゴーストと接触したときと同様に，ミス判定となるように設定する．このようにすることで，過剰にパワー餌を獲得するような経路の生存率が下がり，最終的に選択されないようになる．

表 2　移動経路の選択方法
Table 2　Selection of a movement path.

現在の戦略	選択方法
生存	最も生存率が高い経路を選択する
食事	生存率が閾値以上である経路のうち最も餌の獲得率が高い経路を選択する
追跡	生存率が閾値以上である経路のうち最もゴーストの撃退率が高い経路を選択する

生存：　パワー餌に接触可能．いじけゴーストに接触可能
食事：　パワー餌に接触可能．いじけゴーストに接触不可能
追跡：　パワー餌に接触不可能．いじけゴーストに接触可能

6.7　移動経路の選択

UCT は最終的に戦略に基づいてミズパックマンと接続する経路を選択する．ゴーストと接触しないことを最優先目標として，安全な経路のうち，戦略を遂行できるような経路を選択する．選択方法は**表 2** のようになる．

7. 性能評価実験

7.1　方　　法

実験では，実際に提案システムに Ms. Pac-Man を自動操作させて，その性能を評価する．これは，Ms. Pac-Man Competition の規約に則り，合計 20 回の試行を行うものである．それぞれの試行の際には，(1) ゲーム終了時の獲得得点，(2) ゲーム終了時のステージ到達数を記録する．実験は，CPU Core2 Duo 3 GHz，メモリ 2 GB のマシン環境で行った．提案システムの実装には C++ 言語を用いた．シミュレーション回数は 300，シミュレーション最大サイクル数は 30 とした．また，Ms. Pac-Man は Microsoft Games の Revenge of Arcade に収録されているものを使用した．

7.2　提案システムのパラメータ

実験時の提案システムのパラメータは以下のとおりである．これらの値はいくつかの値を試してみた結果，経験的に最も有効であったものを採用している．

(1) 離散空間のルール

C 経路上でミズパックマンが反転するかを決定するための，ゴーストとの距離 = 2（grid）
ミズパックマンの次サイクルでの更新をスキップするかを決定するための餌の獲得数 = 10（個）
ミズパックマンの更新が連続して行われるかを決定するためのコーナを曲がった数 = 2（回）

表 3　獲得得点
Table 3　Scores achieved.

p	min	max	mean	s. d.
提案システム	6,840	37,630	24,926.50	9,058.71
ICE Pambush3	8,620	30,660	20,574.50	6,213.21

(2) シミュレーション

UCT で実行するシミュレーションの回数 = 300（回）
1 回のシミュレーションにおける最大サイクル数 = 30（サイクル）

(3) 戦略

生存と食事の戦略を切り替えるための，生存率の閾値 = 0.20
ゴーストの追跡を行うかどうかを決定するための，最近傍ゴーストとの距離 = 120（pixel）

7.3　ICE Pambush3

提案システムの性能を既存のプログラムと比較するために，Ms. Pac-Man Competition CIG2009 の優勝プログラムである「ICE Pambush3」を用いる．ICE Pambush3 は立命館大学の Ruck THAWONMAS 教授の知能エンターテインメント研究室チームによって開発された，自動操作における Ms. Pac-Man の世界記録を保持する，現在，最も高性能なプログラムである．ICE Pambush3 はルールベースのアルゴリズムを使用しており，経路のコストを考慮した 9 つの条件式によって，ミズパックマンを制御する．

7.4　ゲーム終了時の獲得得点の比較

表 3 はゲーム終了時の獲得得点である．提案システムは最高得点，平均得点において ICE Pambush3 を上回り，総合的な得点獲得性能が ICE Pambush3 よりも優れている．特に提案システムが獲得した最高獲得得点である 37,630 点は，過去に Ms. Pac-Man Competition に参加したすべての Ms. Pac-Man エージェントのスコアを上回っており，事実上の世界記録となっている．一方で，ICE Pacmbush3 比較して，提案システムの最小得点が低く標準偏差が大きいことから，安定性の面から，提案システムは ICE Pambush3 に劣っている．その他の結果を加えて総合的に見ても，ICE Pambush3 の方が安定性の面では優位な結果となっている．

7.5　ゲーム終了時のステージ到達数の比較

表 4 はゲーム終了時のステージ到達数である．提案システムは，最大ステージ到達数，平均ステージ到達数，ともに ICE Pambush3 を上回っており，ステージ踏破性能の高さを示している．一方で，標準偏差は提案システムの方が大きく，ICE Pambush3 に対してやや

表 4　ステージ到達数
Table 4　Summarizes the number of stages reached.

p	min	max	mean	s.d.
提案システム	1	7	4. 4	1.497
ICE Pambush3	1	5	3. 4	0.970

安定性に欠ける．また，最小ステージ到達数は両プログラム 1 となっている．これは，提案システム，ICE Pambush3 ともに最初のステージをクリアすることができずにゲームオーバーとなってしまったケースが存在することを示している．

7.6 考　　察

提案システムがルールベース型の ICE Pambush3 に比べて生存能力に関して優秀な性能を示したのは，UCT による挟み撃ちの回避や戦略の切替えによる生存率・撃退率を重視したプレイが効果的に働いたためであるといえ，あらかじめ知識をほとんど与えていないにもかかわらず自動的に戦略が形成されたことを示している．これは UCT の適用が成功した他のゲームにも見られた成果であり，Ms. Pac-Man においても UCT が有効である可能性を示唆している．

一方で，動作の安定性に関して，提案システムは ICE Pambush3 に劣る結果となった．この原因として考えられるのが，特定の状況下において，提案手法が生存率を重視しすぎた結果，危険な場所に配置されている餌を獲得しに行かなくなってしまうことである．この現象が多く発生したのは，4 つのパワー餌をすべて消費してしまったステージ後半において，ステージ角や長い直線等のゴーストに挟み撃ちをされやすい場所に餌が残ってしまった状況である．餌の数が残り少なくなるとゴーストの動きが活発になり，ゴーストに接近される確率が高くなるため，ミズパックマンの生存率が極端に下がる．そのため，ミズパックマンは生存率重視のプレイを行うようになり，危険な場所に残された餌を獲得しに行かなくなってしまう．

8. 餌の獲得順序の効率化

考察のような問題が生じたのは，提案手法が餌の獲得順序をほとんど考慮していなかったために，ゴーストが活発でないステージ序盤のうちに，危険なエリアの餌を獲得しておかなかったことが原因であると考えられる．そこで，提案手法を改良し，迷路上の各地点の危険度を考慮した，餌の獲得順序の効率化を行う．ステージ終盤においてゴーストが活発になると，コーナや長い直線のような生存率が低い地点の餌を獲得することが難しくなる．そのような危険な地点に配置されている餌をステージ序盤に優先的に獲得することで，危険なエリアに残された餌を獲得することができなくなるという問題を解決する．具体的には，シミュレーションにおける報酬「餌の獲得数」を，餌が配置されていたグリッドの危険度を考慮した新しい報酬 X に置き換える．次式において，$W_{i,j}$ は k 番目に獲得した餌の配置グリッド (i,j) における危険度（1 − ミズパックマンの生存率）である．また，シミュレーション中にパワー餌を獲得できるような経路（$n_{ppill} > 0$）は危険度が皆無であり，無理に餌を獲得しに行く必要がないため，報酬の値を十分に小さな数 n としている．

$$X = \begin{cases} \sum_{k=1}^{n} W_{i,j} & (n_{ppill} = 0) \\ n & (n_{ppill} > 0) \end{cases} \tag{4}$$

表 5　獲得得点
Table 5　Scores achieved.

p	min	max	mean	s. d.
提案システム（改良前）	6,840	37,630	24,926.50	9,058.71
提案システム（改良後）	16,800	58,990	31,105.60	11,605.57

9. 性能評価実験 2

本実験の目的は，餌の獲得順序の最適化による提案システムの性能向上の検証である．実験条件は前回と同様のものを用いる．

9.1 ゲーム終了時の獲得得点

表 5 はゲーム終了時の獲得得点である．改良前と比較して獲得得点の大幅な向上が見られる．特に，改良手法の最高得点は 58,990 点であり，これは改良前に獲得した Ms. Pac-Man における世界記録をさらに大幅に更新している．また，最小獲得得点も改良前に比べて大きく上昇しており，最初のステージをクリアできずにゲームオーバーとなってしまい，点数がきわめて低くなってしまうような不安定な動作が軽減されている．一方で標準偏差は大きくなっており，依然として得点の獲得幅は大きい．

9.2 ゲーム終了時のステージ到達数

表 6 はゲーム終了時のステージ到達数である．ステージの平均到達数が改良前よりも高くなっており，ステージ踏破性能の向上が見られる．また改良前に見られた最初のステージでゲームオーバーとなってしまうケースが 1 度もないことや，標準偏差が改良前よりも低くなっていることから，改良前に比べて安定して餌を獲得することができている．

表 6　ステージ到達数
Table 6　Summarizes the number of stages reached in the game.

p	min	max	mean	s. d.
提案システム（改良前）	1	7	4. 4	1.497
提案システム（改良後）	2	7	4. 8	1.361

9.3 考　　察

餌の獲得順序の効率化により，提案システムの性能は飛躍的に向上した．活動しているゴーストの数が少なく，パワー餌をすべて消費していないような比較的安全なステージ序盤において，ステージ角や長い直線等の危険な地点に配置されていた餌を積極的に獲得しに行くような動きが見られた．これは，ICE Pambsuh3 や，改良前の提案手法には見られなかった所作である．加えて，パワー餌を獲得できるような経路を餌の獲得率が低い経路とみなすことで，パワー餌を獲得するタイミングが危険度の高いステージ後半に寄ったため，1 つのパワー餌で撃退できるゴーストの数は平均的に上昇した．また，同様の理由でミズパックマンの窮地をパワー餌で脱する機会が増えたため，生存率の向上にもつながる結果となった．

10. おわりに

本研究では，挟み撃ち回避問題を，コンピュータ囲碁で成功したモンテカルロ木探索アルゴリズムである UCT を用いて解決することを試みた．連続的ゲームである Ms. Pac-Man を離散化したシミュレーション空間において，挟み撃ちの発生確率を意図的に高めたランダムシミュレーションを大量に行うことで，実空間における挟み撃ちの発生確率を見積もり，発生確率が高い移動経路を回避させることでミズパックマンの生存率を向上させた．結果として，提案手法は過去の Ms. Pac-Man Competition に出場したすべての自動操作プログラムよりも高得点を獲得することができ，コンピュータが持つ世界記録を大きく上回ることができた．今後の課題としてはアルゴリズムの改良があげられる．特にシミュレーション精度の向上は最重要案件である．本研究においてあらかじめ手動で設定していた各種パラメータを，ゲーム画面から自動的に獲得できるようにすることで，シミュレータの挙動はより正確になり，本手法の性能を大いに引き上げることになることが期待される．

参 考 文 献

1) Lucas, S.M.: Ms. Pac-Man Competition, available from ⟨http://cswww.essex.ac.uk/staff/sml/pacman/PacManContest.html⟩.
2) Koza, J.R.: *Genetic Programming on the Programming of Computers by Means of Natural Selection*, MIT Press (1992).
3) Lucas, S.M.: Evolving a Neural Network Location Evaluator to Play Ms. Pac-Man, *IEEE Symposium on Computational Intelligence and Games 2005*, pp.203–210 (2005).
4) Wirth, N. and Gallagher, M.: An influence map model for playing Ms. Pac-Man, *IEEE Symposium on Computational Intelligence and Games 2008*, pp.228–233 (2008).
5) Robles, D. and Lucas, S.: A Simple Tree Search Method for Playing Ms. Pac-Man, *IEEE Symposium on Computational Intelligence and Games 2009*, pp.249–255 (2009).
6) Emilio, M., Moises, M., Gustavo, R. and Yago, S.: Pac-mAnt: Optimization Based on Ant Colonies Applied to Developing an Agent for Ms. Pac-Man, *Proc. 2010 IEEE Conference on Computational Intelligence and Games 2010*, pp.458–464 (2010).
7) Tong, B.K.B. and Sung, C.W.: A Monte-Carlo approach for ghost avoidance in the Ms. Pac-Man game, *Proc. IEEE GIC*, Hong Kong, pp.1–8 (Dec. 2010).
8) Coulom, R.: Efficient selectivity and backup operators in Monte-Calro tree search, *Proc. 5th International Conference on Computer and Games* (2006).
9) Kocsis, L. and Szepesvari, C.: Bandit based montecarlo planning, *European Conference on Machine Learning*, pp.282–293 (2006).
10) Chung, M., Buro, M. and Schaeffer, J.: Monte Carlo Planning in RTS Games, *IEEE Symposium on Computational Intelligence and Games* (2005).
11) Balla, R. and Fern, A.: UCT for tactical assault planning in real-time strategy games, *Proc. International Joint Conference on Artficial Intelligence 2009*, pp.40–45 (2009).
12) Matsumoto, H., Ashida, T., Ozasa, Y., Maruyama, T. and Thawonmas, R.: ICE Pambush 3, *2009 IEEE Symposium on Computational Intelligence and Games competition entry* (2009).
13) Lucas, S.M.: Ms. Pac-Man Competition (screen capture mode) IEEE CIG 2011 Results, available from ⟨http://dces.essex.ac.uk/staff/sml/pacman/CIG2011Results.html⟩.

付　　録

本研究で開発されたプログラムは 2011 年 9 月 1 日に開催された Ms. Pac-Man Competition（screen capture mode）IEE CIG 2011 に参加し，出場プログラム中最高得点（36,280）を獲得して優勝したことを最後に報告する．詳細は同コンペティションの公式ホームページを参考にしていただきたい[13]．

(平成 23 年 4 月 24 日受付)
(平成 23 年 9 月 12 日採録)

池畑　　望（正会員）

昭和 60 年生．平成 23 年電気通信大学電気通信学研究科情報工学専攻修士課程修了．同年株式会社コナミデジタルエンタテインメント入社．

伊藤　毅志（正会員）

1994 年名古屋大学大学院工学研究科情報工学専攻修了．工学博士．同年より，電気通信大学情報工学科助手．2007 年より，同助教．2010 年より，電気通信大学情報理工学研究科助教．人間の思考過程，学習過程に関する研究に従事．現在は特に思考ゲーム等を題材にした認知科学的研究に興味を持つ．著書に『先を読む頭脳』（新潮社，共著）ほか．日本認知科学会，人工知能学会等会員．コンピュータ将棋協会，コンピュータ囲碁フォーラム各理事．