

Ms. Pac-Man におけるモンテカルロ木探索を用いたゴーストチームの協力制御 Applying Monte-Carlo Tree Search to Collaboratively Controlling of a Ghost Team in Ms. Pac-Man

グエン・クアン・キエン†
Nguyen Quang Kien

ターウオンマツ・ラック†
THAWONMAS Ruck

1. はじめに

Ms. Pac-Man はデジタルゲームとしてはルールが単純であるが、優秀なプレイを行うためには複雑な戦略が必要である。これは AI 技術発展のための良質なテストベッドであると考えられる。

過去にミズパックマンの自動制御についての研究はあるが、ゴーストについての研究は例が少ない。理由の一つはゴーストを制御するのは Ms. Pac-Man の実機ゲームでできないため、別の標準ゲームシミュレータが必要なためである。幸い、2011 年の CEC で Ms. Pac-Man vs. Ghost Team Competition という大会が行われ、初めてゴーストチームの性能を競うことができるようになった。本研究はこの大会で使用したゲームシミュレータを用いた。

ゴースト制御の難題はゴーストの協力制御である。各ゴーストはミズパックマンと同じ速度で移動する一方で反転できず、エディブルの状態でも速度も半分になっているため、協力的な制御を実施しないとパックマンを効率的に追い詰めることができない。

本研究では Ms. Pac-Man におけるゴーストチームを効率的に制御すること（ゲームスコアが低ければ低いほど良い）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取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)法を用いた。UCT 法はシミュレーションに基づく木探索手法の一つで、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多くの探索を行う手法である。

2. 提案手法の概要

2.1 評価対象

迷路におけるクロスポイントとクロスポイントを結んだものを、「C 経路」と定義する。C 経路とは、その途中に分岐を含まない、一本道のことである。ゲームルールによって、ゴーストは自由に反転することができないので、ゴーストは C 経路に入れば、C 経路の他の端に行かなければならない。このルールのため、C 経路がゴーストの移動単位とみなすことができる。

2.2 提案手法

様々な戦略のミズパックマンと対戦できるゴーストを開発するためには、以下の二つの問題を解決しなければならない。

- (1) ミズパックマンの移動パターンを把握できる、あるいはミズパックマンの移動先を予測することができる。
- (2) (1)を基にして、ゴーストを効率的かつ協力的に制御

この二つの問題を解決するために、我々はモンテカルロ木探索とルールベースを組み合わせたシステムを提案する。このシステムの中で、Pinky, Inky, Sue は UCT 制御ゴーストであり、Blinky はルールベースによって移動する。UCT でランダムシミュレーション（シミュレーション中、パックマンとゴーストがランダム移動すると仮定する）を用いるが、ミズパックマンの移動パターンが把握できていれば、より短時間で効果的な戦略を導き出すことができる。そして、Ms. Pac-Man がリアルタイムゲームであるので、すべてのゴーストを UCT で制御すると負荷が高いため、一体をルールベースゴーストにし、その他を UCT 制御することで CPU の作業負荷も削減できる。それにより、リアルタイムで計算できるようになる。また、UCT ゴーストはルールベースゴーストの行動を反映した戦略をとる。

3. ゴースト制御

3.1 ゲーム木

各ゴーストに対して別のゲーム木（モンテカルロ木:MCT）が構築される。このゲーム木はゴーストの立場から見るゲーム状態を表す木である。ノードはクロスポイント、枝は C 経路に相当する。ルートノードはゴーストが次に到着するクロスポイント、子ノードはゴーストが親ノードから辿る最初のノードである。ゴーストが自由に反転できないので、直系親ノードがこのノードの直系子ノードになれない。

最初のゴーストの木はルートノードしか含まれない。このノードから、モンテカルロ木の構築法の流れは以下の通りである。

- ① ゴーストの現在位置から、次のクロスポイントに辿りついた状態を仮定する。この状態はゴーストとミズパックマンの移動ルール（3.2.1 と 3.2.2）に基づいて仮定される。
- ② ルートノードから、端ノードに降りる経路を選択する。
- ③ 端ノードの子ノードを展開し、一つの子ノードを選択する。
- ④ モンテカルロのランダムシミュレーションを開始する。このシミュレーション中ではゲーム木の所有ゴーストがルートノードから選択したノードまでの経路に沿って動かす。
- ⑤ モンテカルロシミュレーションの結果によって、来訪したノードに適切な報酬を与える。
- ⑥ ②から⑤までを何度も試行して、ルートノードの一番報酬が高い子ノードを選択し、このノードまでの方向を次のゴーストの方向とする。

ゲーム木が構築されると、UCT も採用される。ゲーム木の降下中には、各ノードにおいて、最も UCB1 値が高い子ノードが選択される。UCB1 値はノード i に対して以下のように計算される。

$$UCB1(i) = \frac{X_i}{T_i} + C \sqrt{\frac{\ln T}{T_i + \epsilon}}$$

$$T = \sum_j T_j + 1$$

ここで T_i はノード i の来訪回数であり、 T は親ノードの来訪回数であり、 C は定数、 ϵ は十分小さい定数である。上の式で用いる X_i はノード i の全ての報酬の総和である。

3.2 モンテカルロのシミュレーション

仮定されたゲーム状態からシミュレーションが開始される。シミュレーション途中で各ゴーストとミズパックマンは 3.2.1 と 3.2.2 のルールに従って動く。シミュレーションは次のいずれかの条件を満たせば終了する。

- ・ ミズパックマンが食べられる。
- ・ レベルクリア。
- ・ 制限ゲームサイクル（ゲーム中の時間の単位）の経過。

3.2.1 ゴースト移動ルール

ゲーム木の所有ゴーストがこの木によって動き、他の UCT ゴーストは自分の構築された木によって動く。このような移動ルールのおかげで、各 UCT ゴーストが互いに通信でき、協力制御をすることができる。Blinky がリアルゲーム中において常にミズパックマンの前のクロスポイントに注目して動くが、モンテカルロシミュレーションでランダムに動く。これはヒューリスティックに行われ、UCT ゴーストの攻撃性を高めるためである。さらに、ゲーム木のルートノードの来訪回数が制限されたので（ほぼ 1000 回）、木の深さも制限された。UCT ゴーストが端ノードを辿れば、ランダムに動く。

3.2.2 ミズパックマンの移動ルール

- ミズパックマンの行動は以下のルールに従う。
- ・ ミズパックマンの方向を決定するのはコーナー又はクロスポイントである。
 - ・ 同 C 経路で前方の一定距離に普通のゴーストがいれば反転する。
 - ・ 同 C 経路で前方の一定距離にパワーピルがあればそのまま進む。
 - ・ ミズパックマンの予測によって動くが、ある確率でランダムに動く。
 - ・ ある間隔でミズパックマンの予測によって動き、後はランダムに動く。

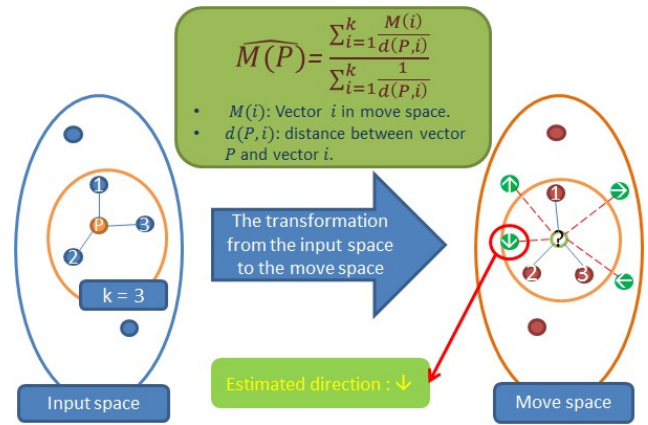


Figure 1 予測法の流れ

3.2.3 報酬

来訪したノードにモンテカルロシミュレーションの結果によって報酬が与えられる。報酬値は二つの基準に基づいて決定される。これはシミュレーションから出力したスコアとタイムの逆数である。そして、ゴーストが散り散りに動くとゴーストからミズパックマンまでの安全距離などを保持するために、ペナルティを報酬値に付ける。

4. ミズパックマンの移動先の予測

4.1 考慮すべき測定

ミズパックマンの行動を予測するため、いくつかの側面が考えられる。例えば、人間のプレイヤーがミズパックマンを制御する場合、ミズパックマンから最も近いゴーストと、一番近いパワーピルなどを考慮する。ここから、ミズパックマンの移動先を予測するためには以下の測定に注目すべきである。これらはゲーム状態を表すとみなす。

- ・ ミズパックマンから一番目と二番目に近いゴーストの状態。
- ・ そのゴーストからミズパックマンまでの距離。
- ・ ミズパックマンから一番近いパワーピルの距離。
- ・ ミズパックマンから一番近いピルの距離。
- ・ ミズパックマンから一番近いクロスポイントの距離。
- ・ ミズパックマンから一番近いパワーピルに一番近いゴーストの状態。
- ・ そのゴーストとミズパックマンの距離。
- ・ そのゴーストとバックマンの一番近いパワーピルの距離。

各距離測定はゲーム通路上の最短距離である。この測定があれば、K 近傍アルゴリズムを用いて、現在のゲーム状態に最も近い過去のゲーム状態を探して、ミズパックマンの次の行動がある程度予測できる。

4.2 ミズパックマン移動パターンの空間化

各ゲーム状態で注目した測定が入力空間という多次元空間の中の一つのベクトルと見なせる。各ゲーム状態に対してミズパックマンが狙うゴールがあると考えながら、このゴールを表すために、移動空間という多次元空間上の移動

ベクトルを生成する。例えば、ミズパックマンがこの方向に行けば、あるオブジェクトから離れる、近づくという意味を表すと考えられる。そのように、移動ベクトルは相応入力ベクトルの考慮するオブジェクトに対して、先のような意味を表すベクトルである。入力空間と移動空間から、ある程度ミズパックマンの動く方向を予測することができる。

4.3 予測法の流れ

パックマンの予測は以下のステップで行っている(Figure 1)。

- ① ゲームからミズパックマンの移動パターンのデータ(各ゲーム状態に対しての入力ベクトルと移動ベクトル)を集める。
- ② 取めたデータから、現在の入力ベクトルに対して K 個 ($K=3$) の最も近い入力ベクトルを選択する。
- ③ この K 個の入力ベクトルの相応移動ベクトルから、現在の移動ベクトルを判断する。
- ④ 現在の入力ベクトルから、ミズパックマンの移動できる方向に対して、移動ベクトルを生成する。
- ⑤ その中で判断された移動ベクトルの最も近いベクトルを選択して、このベクトルの移動方向がミズパックマンの移動方向とみなす。

ミズパックマンの移動先予測のために集めたデータが多すぎると計算に時間がかかるので、モンテカルロシミュレーションに適さない。そうでない場合、トレーニングデータが少なすぎると予測が全く違う可能性がある。しかし、適当なデータ数を使用すると、ミズパックマンの移動パターンが局所的に予測でき、随時変化するミズパックマンの戦略の予測に対し非常に有効である。

5. 結果と結論

表 1 : CEC'11 Ms. Pac-Man vs Ghosts の結果

	ICE gUCT	BruteForce	emgallar
James	16436	24158	21805
emgallar	16208	22599	17938
MsAriadne	19282	19031	20076
平均	17308.7	21939.3	19939.7

提案したゴーストは CEC'11 の Ms. Pac-Man vs. Ghost Team Competition において優勝という成績を取めた。表 1 は CEC'11 の結果の一部で、ミズパックマンとゴーストの上位三チームの対戦結果である。ICE gUCT が提案した手法のゴーストである。ICE gUCT と他のゴースト (BruteForce はルールベースのゴースト、emgallar は Ant Colony アルゴリズムを用いたゴースト) を比較すると、モンテカルロ木探索を用いたゴーストの方が優秀だとわかる。そのためミズパックマンの移動先の予測とモンテカルロ木探索を組み合わせることは、ゴースト制御において有効であると言える。このような手法は様々なミズパックマンの戦略に対応できると考えられる。さらに、この手法は一般的な AI エージェントを生成するポテンシャルを持っているとも考えられる。

参考文献

- [1] N. Ikehata and T. Ito, Monte Carlo Tree Search in Ms Pac-Man, in *The 15th Game Programming Workshop*, IPSJ Symposium Series Vol.2010/12, 2010. (in Japanese)
- [2] B.K. B. Tong and C. W. Sung, "A Monte-Carlo approach for ghost avoidance in the Ms. Pac-Man game," *Proc. IEEE GIC*, pp. 1-8, Hong Kong, Dec. 2010.
- [3] S. Samothrakis, D. Robles, and S. Lucas. "Fast approximate max-n Monte-Carlo Tree Search for Ms. Pac-Man". *IEEE Transactions on Computational Intelligence and AI in Games*, 2011.
- [4] www.pacman-vs-ghosts.net/cec11results/