General Game Playingにおける 一般性の高い類似盤面利用手法の検討

上宮 佳晃[1]　横山 大作[1]

概要：General Game Playing(GGP) とは、初見の様々なゲームを上手くプレイすることができるプログラムの実現を目標とした問題カテゴリである。ゲーム固有の知識や技術に頼ることができないため、正確な評価関数を必要としないモンテカルロ機探索（MCTS）の活用が主流となっている。一方で、汎用の実装が求められるため、シミュレーション回数を容易に多くすることができない。この問題に対して、我々は、MCTS で展開するノードに探索経験から一致または類似する盤面の情報を付加することで、シミュレーション回数を増やすのと同等の効果を狙う手法を提案した。過去の検証の結果、類似盤面の活用によって一部のゲームでは勝率を上げることができた。本論文では、拡張する盤面と探索経験から取得する類似盤面のツリー上の距離を計測することで、本研究の手法が有効となるゲームの特徴を捉えた。そして、事前の準備時間で簡易的なシミュレーションを行い、扱うゲームによって類似盤面の利用をするか否かを判断することで、勝率を安定化させる仕組みを検討した。

Towards a universally effective utilization of board similarities in General Game Playing

Yoshiaki Uemiya[1]　Daisaku Yokoyama[1]

Abstract: General Game Playing (GGP) is a problem category that aims to achieve programs that can play a variety of games only with the rule definitions. Since it is not possible to rely on game-specific knowledge and techniques, Monte Carlo Tree Search (MCTS), which does not require an exact evaluation function, is widely used. On the other hand, the number of simulations cannot be easily increased due to the need for general-purpose implementation. With regard to this problem, we proposed a method to increase the number of apparent simulations by adding matched or similar board information from exploration experience to the expanded node in MCTS. Results of past verifications showed that using similar board information increased the win rate in some games. In this paper, we explored the characteristics of games for which our approach is effective by measuring the distance on the tree between the extended node and the similar board information obtained from the exploration experience. We also proposed a switching algorithm for our approach using a small number of simulations in the preparation period, which successfully stabilized the win rate.

1. はじめに

General Game Playing（GGP）は、初見の様々なゲームに対して高い精度でプレイするプログラムの実現を目標とした問題カテゴリである [1]。事前にプレイするゲームが決まっていないため、ゲーム固有の知識や技術に頼ることができないという課題が存在する。そのため、正確な評価関数を必要とせずにツリーを探索・構築することができる MCTS が GGP に特に適した手法となっている。

特定のゲームに対して研究される MCTS では、プレイするゲームに特化した専用の知識を用いるような最適化がなされている。一方で、GGP における MCTS では汎用の実装を利用するため、シミュレーション回数を多くすることができないという問題点が一般的に存在する。

この問題に対処するために、我々は、MCTS で展開する

[1] 明治大学大学院理工学研究科情報科学専攻
Department of Information Sciences Graduate School of Science and Technology, Meiji University

ノードに対して、以前訪れた類似盤面の報酬や訪問回数を加え、見かけ上のシミュレーション回数を増やすのと同等の効果を狙う探索手法を提案した [2]。

本手法では、探索中に新しい局面を訪問するたびに、その情報を記録する「探索経験」を作成して利用する。探索経験は、盤面表現の文字列をビット列にエンコードし、盤面の状態や報酬、訪問回数からなるハッシュ値を紐づけて作成した。そして、MCTS の拡張ステップにおいて、拡張する盤面に情報を付加するために、拡張する盤面と一致、または類似する盤面を探索経験から取得し活用した。過去の検証の結果、少ないシミュレーション回数でも類似盤面を活用することによって一部のゲームでは勝率が上がることを確認できた。

本研究では、勝率が上がるゲームと上がらなかったゲームに着目し、その違いを明らかにするとともに、多くの種類のゲームにおいて勝率を安定的に向上させる手法の構築を試みる。本論文の貢献は以下の 3 点である。

(1) 盤面のエンコード手法を複数検証し、盤面の出現頻度を考慮した手法が有効であることを明らかにした。
(2) 類似盤面の利用が有効でないゲームは、類似度が正しく設定できていない場合と、プレイごとに傾向が大きく変わる場合、の 2 通りがあることを確認した。
(3) ゲーム開始前の短い準備時間で、類似盤面の利用が有効であるか簡便に判定し、手法を切り替えることで勝率を安定化させる仕組みを検討し、有効性を検証した。

2. 関連研究

2.1 Game Description Language（GDL）

GGP では、論理プログラミング言語の一種である Game Description Language（GDL）[4] を用いて、ゲームルールや盤面状態を表現する。ゲームルールには、プレイヤごとの役割や、合法手によってどのようにゲームの状態が変化するのかといったことが記述されている。

ゲームの盤面状態には、主に駒ごとに種類や座標が記述されており、他にも手番やターン数といった情報が含まれている。図 1 は Tic-Tac-Toe のある盤面状態であり、図 2 のように GDL で記述することができる。各ターンごとに全てのプレイヤは与えられた盤面に対する合法手をゲームマスターに送信する。このとき、手番でないプレイヤはゲームの状態に影響しないという信号として noop（no operation）を送る。

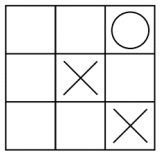

図 1 tic-tac-toe の盤面

```
( true ( cell 1 1 b ) ), ( true ( cell 1 2 b ) ),
( true ( cell 1 3 b ) ), ( true ( cell 2 1 b ) ),
( true ( cell 2 2 x ) ), ( true ( cell 2 3 b ) ),
( true ( cell 3 1 o ) ), ( true ( cell 3 2 b ) ),
( true ( cell 3 3 x ) ),  ( true ( control oplayer ) )
```

図 2 GDL で表現された盤面情報

2.2 MCTS

各ターンごとに与えられた現在の盤面状態をルートノードとして MCTS を行う。メモリ上に構成されるツリーの各ノードには GDL で表現される盤面状態や報酬, 訪問回数が記録される。本研究で扱う MCTS は以下の 4 つのステップから構成される。

(1) 選択：構築している木においてルートノードからリーフノードまでたどっていく。このとき、ノードの選択には最善手と新しい手の選択のバランスをとる必要があるため Upper Confidence bounds applied to Tree（UCT）[5] を用いる。UCT は、探索した手の内の最善手と探索回数が少ない手の選択のバランスをとるアルゴリズムである。ノード s において以下の式の値が最大となるような子ノード a を選択するようにする。

$$UCT = Q(s,a) + C\sqrt{\frac{\log N(s)}{N(s,a)}} \quad (1)$$

第一項の Q(s,a) は、ノード a が選択された場合の平均報酬を示している。第二項において、N(s) は s の訪問回数、N(s,a) は s においてノード a が選択された回数である。よって第一項は最善手の探索を、第二項は訪問回数が少ない子ノードの探索を示し、パラメータ C によって、最善手の選択と探索の調整を行う。また、全てのノードが一度は訪問されるように、未訪問のノードは UCT =∞となり選択されるようになる。

(2) プレイアウト：終局状態になるまでリーフノードの盤面からゲームをランダムにシミュレートする。

(3) 拡張：プレイアウトでリーフノードの次に選択した盤面を新たなリーフノードとして追加する。過剰にメモリを増やすことを防ぐために、追加するノードを 1 つにする。

(4) バックプロパゲーション：プレイアウトで得られた報酬をもとに、リーフノードからルートノードまでバックプロパゲーションを行い、累積報酬や訪問回数を更新する。

2.3 GGP に用いられてきた手法とその性能

本項では、過去の GGP の大会で優秀な成績を示したプレイヤに搭載されていた手法について紹介する。いずれも、過去のシミュレーションから得られる探索結果を活用して、プレイアウトステップにおいて、最善手の選択を行う手法となっている。これらの手法では、目的の盤面に対して、探索経験から一致する盤面情報を取得し活用しているが、類似盤面の情報を利用した手法とはなっていない。

2.3.1 Move-Average Sampling Technique (MAST)

MAST は、2008 年に AAAI の GGP の大会で優勝した Cadiaplayer に用いられていた探索制御手法である [6]。

バックプロパゲーションのステップにおいて、探索したノードに対して終局状態で得られた報酬と訪問回数を反映させていく際に、出現した各合法手 a に対する平均報酬 Q(a) も更新し記録しておく。よって、ゲームの状態に関係なく良い結果をもたらす合法手は、最終的に平均報酬が高くなる。そして、プレイアウトステップにおいて合法手の選択をする際には、式 2 のギブスサンプリングを用いて平均報酬が高い手を選択する確率が高くなるようにする。

$$P(a) = \frac{e^{Q(a)/\tau}}{\sum_{b=1}^{n} e^{Q(b)/\tau}} \quad (2)$$

合法手 a において、P(a) は、あるプレイアウトの盤面状態で a を選択する確率であり、Q(a) は平均報酬を示す。また τ はパラメータである。[7] では、ランダムにプレイアウトを行う MCTS プレイヤに対して MAST を用いた MCTS プレイヤは Checkers では 54.83%、Othello では 58.67% 、Breakthrough では 88.67% の勝率を示していた。

2.3.2 N-gram Slection Techinique (NST)

N-Gram selection Technique（NST）[9] は、出現した単語に基づいて次の単語を予測する統計モデルで有名な N-gram を活用したシミュレーション戦略であり、この手法を新たに搭載した Cadiaplayer は 2012 年に GGP の大会で優勝した。NST では、プレイアウトで次の手を決定する際に、平均報酬が記録された長さ 1,2,3 の連続した手のシーケンスを用いて推測を行う。これらのシーケンスは MCTS で終局状態に到達した後、シミュレーションで出現した長さ 1, 2, 3 のすべての手の組合せを抽出することで形成される。

図 3 のノード B から次のノード ci(i=1, 2, 3) を選択する際に、長さ L=1,2,3 のシーケンスを決定する。このとき L＝1 のシーケンスは、ci 単体、L＝2 のシーケンスは、ci と 1 個前のノードである B、L＝3 のシーケンスは、ci と 1 個前の B と 2 個前の A となる。それぞれのシーケンスには平均報酬が存在し、三つのシーケンスの平均報酬の非加重平均を用いて n のスコアを計算する。全ての合法手に対してこのスコアを計算し、これらのスコアを ϵ -greedy 法に適用してどの手を選択するか決定する。

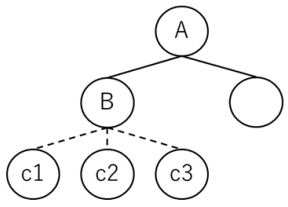

図 3 NST における次の盤面の選択

3. MCTS での類似盤面利用手法

GGP では、事前の短い準備時間内でコストの高い機械学習を用いたシミュレーション精度の向上などを行うことは困難である。そこで本研究では、MCTS の拡張ステップにおいて、拡張する盤面と類似している盤面を過去の探索経験ログから発見し、探索経験に含まれる累積報酬と訪問回数を拡張ノードに加算する、という方法を用いる。

3.1 盤面の類似度

本研究では、GDL で表現される盤面情報をエンコードしてハッシュ値化し、そのハッシュ値間の異なるビット数（ハミング距離）を、盤面の類似度とする。二つのハッシュ値が完全一致の場合は、異なるビット数は 0 であり、ハミング距離は 0 となる。よって、異なるビット数が少ないほどハミング距離の値が小さくなり、二つの盤面は似ていると判定される。任意の局面に対し、指定したハミング距離以下の類似盤面は一定の時間で検索できる。

図 1 と図 4 の場合であれば、○のコマが一マス分ズレている違いがある。それぞれ盤面の文字列を後述するハフマン符号を用いた 4.1(3) の手法でエンコードし、図 5 のようなビット列となったとする。二つの異なるビット列の数は 2 となるため、ハミング距離は 2 となる。

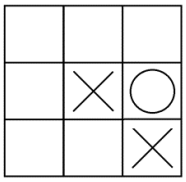

図 4 図 1 から一マスコマがズレた tic-tac-toe の盤面

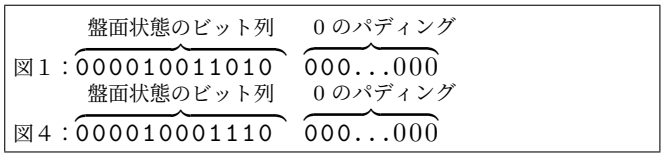

図 5 図 1 と図 4 の盤面の文字列をエンコードして得られるビット列

3.2 探索経験の作成

MCTS のバックプロパゲーションの過程で探索経験を作成していく。選択及びプレイアウトの過程で発生した盤面に対して、4.1 の方法でエンコードしたビット列をハッシュキーとして、そのシミュレーションで得られた報酬と訪問回数を紐付けてハッシュ値として、探索経験に保存する。すでに探索経験に同じ盤面情報がある場合は、報酬と訪問回数を累積していく。

4. 適切な符号化方法の検討

類似盤面をハミング距離で推定するときの精度は、ハッシュのエンコード方法に大きく影響されると考えられる。そこで、MCTS, UCT を元に構築した MCTS プレイヤに対して、3 種類のエンコード方法をそれぞれ実装したプレ

イヤを対戦させ、勝率を用いて比較することを試みる。

4.1 盤面のエンコード

4.1.1 盤面の前処理

GGP では、様々な未知のゲームに対して適用できる汎用的な手法が必要となる。GDL で表現される盤面情報はゲームによって異なり、空白マスの要素がない場合がある。そのため、盤面情報から駒とその位置を示す座標の要素を取り出し、空白マスの要素も含めて座標順にソートした文字列に変換する前処理を行う。GDL の文字列で表現された図 2 の盤面状態に対して前処理を行うと図 6 のようになる。

```
(1,1,#),(1,2,#),(1,3,#),(2,1,#),(2,2,x),(2,3,#),
(3,1,o),(3,2,#),(3,3,x)
```

図 6 図 2 の GDL で表現された盤面情報を前処理して得られる文字列

4.1.2 エンコード手法

前処理を行った盤面の文字列に対して、コマごとや座標の情報も含めたセルごとにビットを割り当てるエンコード方法を 3 種類実装した。本研究では、それぞれの方法について性能の検証を行う。

(1) ランダム：出現するセルごとにランダムなビットを割り当てて、ゾブリストハッシュの手法を用いて、盤面のエンコードを行う。このエンコード方法で 3 の手法を用いると、ハッシュのサイズによっては指定したハミングサイズの類似盤面を十分に取得できない場合がある。図 7 は Connect Four において、1 ターンあたりに拡張する盤面の数に対する探索経験からのハミング距離 2 の類似盤面の取得率である。ハッシュのサイズを 16 ビットから 24 ビットまで 2 ずつ変化させて割合を確認した。このときハッシュ長 20 の取得率が後述のハフマン符号のエンコード方法を用いたときの取得率と近かったため、本研究ではランダムのエンコード方法で作成するビット列はハッシュ長 20 とした。

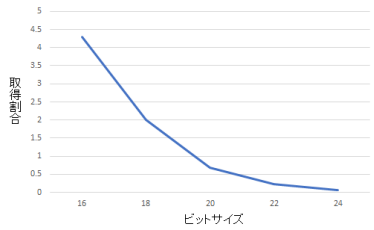

図 7 1 ターン当たりに拡張する盤面に対する探索経験から取得する類似盤面の割合

(2) 出現順：出現するコマごとに 0 から順に増えていくコードを割り当てて、盤面のエンコードを行なう手法。早く出現したコマほど短いコードが割り当てられるため、出現頻度を弱く反映すると期待できる。

(3) ハフマン：ハフマン符号の手法を用いて、コマの出現頻度を考慮した一意なビット列を作成する手法。ゲーム開始前にシミュレーションを 10 回行い、そこで発生した盤面情報からコマごとに出現頻度を計測し huffman 符号をそれぞれ割り当てる。huffman 符号に従ってエンコードしたビット列は 64 の倍数の長さになるように、適宜 0 でパディングを行う。ビット列の長さが 65 以上となった場合は、64 ビットごとに分割し、XOR 演算で複数のビット列を一つにまとめる。

例として 4.1.1 の前処理を行った図 6 の盤面のエンコードを示す。事前のシミュレーションによって huffman 符号が'x'=10, 'o'=11, '#'=0 と定められたとする。huffman 符号に従ってビット列に変換し、64 桁になるように 0 でパディングを行うことで、盤面のエンコーディングが完了となる。

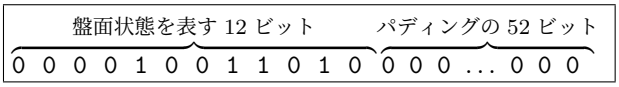

図 8 図 6 の文字列を huffman 符号でエンコードして得られるビット列

4.2 GGP プレイヤ

プレイヤの構築のためにスタンフォード大学のロジックグループが標準化した GGP のプラットフォーム [3] を使用した。GGP ライブラリを用いて MCTS 及び 3 の類似盤面利用手法を実装したプレイヤを作成し、4.1.2 のエンコード手法を変化させて対戦を行った。500 回のプレイアウト後に次の一手の選択を行う際には、累積訪問回数が多いノードを選択するのではなく、実際の選択回数が一番多いノードを次の一手で選ぶようにする。

4.3 実験設定

先行と後攻を交互に変えて 500 回対戦を行う。この一連の実験を 4.1 の 3 種類のエンコード方法それぞれに対して実施する。一手当たりの時間を考慮せずに全てのプレイヤのプレイアウト回数を統一して、構築する MCTS のツリーのノード数が同じになるように実験を行う。実験するゲームにおいて、一手当たりの処理時間を 10 秒から 1 分 30 秒に収めるような範囲として、シミュレーション回数は 500 回とする。類似盤面と判定するハミング距離の閾値は 0 から 2 まで変化させた。

4.4 使用するゲーム

二人零和確定完全情報ゲームとなる 10 種類のゲームを用いて実験を行う。報酬は勝者が 100、敗者が 0 となり、引き分けは互いに 50 となる。各ゲームの特徴を表 1 に示す。

平均終局ターン数、駒の増減、平均最大選択数はランダムなプレイヤー同士で 100 回対戦させて算出した。平均終局ターン数は、終局状態までかかったターン数の平均である。この値が大きいほど、MCTS のプレイアウトのシミュレーション時間は長い傾向にある。駒の種類が多いと、作成されるハフマン符号の種類も多くなる。よって、コマの場所が一マス違うことで似ている盤面でもハフマン符号により作成されるハッシュが大きく異なる可能性がある。駒の増減は、終局状態までに盤面上のコマが増減したターン数の割合である。平均最大選択数は、1 試合ごとの合法手の最大数を平均した値であり、この値が大きいゲームは構築される MCTS があまり深くならない。

表 1　実験を行う各ゲームの特徴

	平均終局ターン数	駒の種類	駒の増減	平均最大選択数
Connect Four	22.52	2	1.0	8
Connect Five	46.83	2	1.0	63.5
Break Through	27.49	2	0.19	19.65
Knight Through	31.45	2	0.13	41.73
Pawn hopping	34.8	2	0.14	16
TTCC4	43.14	8	0.29	11.96
Checkers	51.58	4	0.15	6.51
Sheep & Wolf	40.45	2	0	羊:4, 狼:7
Free For All	30	2	0.42	18.39
Golden Rectangle	29.87	2	1.0	7

4.5　実験結果

4.5.1　ランダムエンコードの評価

表 2　MCTS プレイヤに対してランダムのエンコード方法を実装したプレイヤの成績
(太字は統計的に有意な結果を示す)

	勝ち	負け	引き分け	符号検定 P 値
Connect Four	219	271	10	2.11e-02
Connect Five	**207**	**292**	**1**	1.64e-04
Break Through	**193**	**307**	**0**	2.11e-02
TTCC4	258	242	0	5.02e-01
Sheep & Wolf	265	235	0	1.94e-01

4.1.2 の (1) ランダムによるエンコード方法を用いたプレイヤの対戦結果を表 2 に示す。全てのゲームのおいて、有意差ありの勝ち越しとなるゲームは無かった。このことから、セルごとにランダムなビットを割り当て、駒ごとの座標関係の情報を落としたエンコード方法では、拡張する盤面に対して有意な情報を付加するような類似盤面を本研究の方法で取得することができないと考えられる。

4.5.2　出現頻度を考慮した 2 種類のエンコード方法の成績

4.1.2 の (2) 出現順と (3) ハフマンによるエンコード方法を用いたプレイヤの対戦結果を表 3 に示す。閾値 0 の時は、2 種類のエンコード方法によらずハッシュ値が完全一致する盤面のみを利用する手法であり、一定程度の勝率向上が見られる。異なるビット数が 1 となる二つの盤面のビット列の組み合わせはゲームを通してあまり出現しないため、探索経験からハミング距離が 1 となる盤面を取得することは稀であった。そのため、閾値 1 のときは勝率は閾値 0 とあまり変わらなかった。また、閾値 3 以上は探索時間増加の弊害が目立つ結果となった。本項ではそれらの結果は省略する。そして、類似盤面を利用する閾値 2 の時は、以下のような結果が得られた。

出現順：類似盤面を利用することで Break Through, Knight Through, Pawn Whopping は勝率が上がるが、他のゲームは勝率が下がってしまった。特に Connect Four, Free For All は顕著に勝率が下がった。

ハフマン：Break Through, Connect Five, Pawn Whopping は類似盤面を利用することで勝率が上がるが、他のゲームに関してはあまり効果が出なかった。

4.6　ゲーム毎の違いの検証

4.5.2 の実験結果より、本研究の手法がゲームによって有効な場合とあまり上手く作用しない場合があることを確認した。そこで、ハミング距離で類似盤面と判定された盤面が、探索木の中ではどの程度の距離にあったのか、を調べることで、ゲームごとの挙動の違いを理解することを試みる。ここでは、盤面間の距離の指標として、探索木の中で二つの盤面間にある枝の本数を用いることにする。

4.6.1　ランダムエンコードの検証

Connect Four において、ランダムのエンコード方法を利用した時の、類似盤面と判定された盤面の真の距離ごとの出現分布を図 9 に示す。横軸が真の距離、縦軸が 1 ターン分のシミュレーションでの出現回数である。各ゲームでの 10 回分の測定を折線で示している。距離 6 から 8 を中心とするように山形のグラフを形成した。このことから、本研究の類似盤面の取得方法では、ランダムに盤面を表現して、ハミング距離が 2 となる二つの盤面を取り出した時、多くは距離が 6 から 8 となることが考えられる。

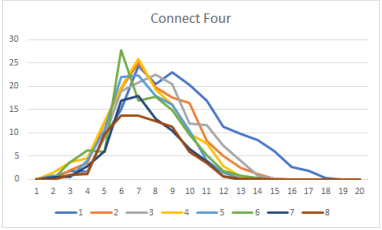

図 9　ランダムのエンコーディングを用いた
1 ターンあたりの類似盤面を取得した回数と拡張盤面との距離

表 3　MCTS プレイヤに対して拡張ノードに情報を付加する EE プレイヤの成績
(太字は統計的に有意な結果を示す)

	出現順								ハフマン							
	ハミング距離 0				ハミング距離 2				ハミング距離 0				ハミング距離 2			
	勝ち	負け	引き分け	符号検定 P 値	勝ち	負け	引き分け	符号検定 P 値	勝ち	負け	引き分け	符号検定 P 値	勝ち	負け	引き分け	符号検定 P 値
Break Through	**302**	**198**	**0**	3.80e-06	**377**	**123**	**0**	5.23e-31	**312**	**188**	**0**	3.24e-08	**326**	**174**	**0**	1.01e-11
Connect Five	**290**	**210**	**0**	4.04e-04	**275**	**225**	**0**	2.83e-02	272	228	0	5.44e-02	**287**	**213**	**0**	1.08e-03
Pawn Whopping	**269**	**213**	**18**	1.22e-02	**343**	**150**	**7**	1.93e-18	**346**	**147**	**7**	1.52e-19	**362**	**129**	**9**	1.25e-26
Knight Through	**290**	**210**	**13**	4.00e-04	**388**	**112**	**8**	1.17e-36	**292**	**208**	**0**	1.99e-04	267	233	0	1.40e-01
TTCC4	**307**	**190**	**0**	1.73e-07	255	243	0	6.22e-01	**308**	**192**	**2**	2.40e-07	**285**	**213**	**2**	1.44e-03
Connect Four	**288**	**199**	**13**	6.39e-05	**133**	**359**	**8**	4.96e-25	**309**	**183**	**8**	1.47e-08	**283**	**208**	**8**	9.79e-04
Free For All	**200**	**117**	**183**	3.64e-06	**28**	**409**	**63**	7.03e-88	**214**	**110**	**176**	7.89e-09	**139**	**279**	**82**	6.76e-12
Checkers	**185**	**66**	**249**	3.08e-14	**164**	**107**	**229**	6.43e-04	**186**	**73**	**241**	1.51e-12	**182**	**78**	**240**	9.52e-11
Golden Rectangle	**276**	**224**	**0**	2.25e-02	246	254	0	7.85e-01	**290**	**210**	**0**	4.00e-04	261	239	0	3.48e-01
Sheep and Wolf	255	245	0	6.87e-01	243	257	0	5.61e-01	**282**	**218**	**0**	4.79e-03	261	239	0	3.48e-01

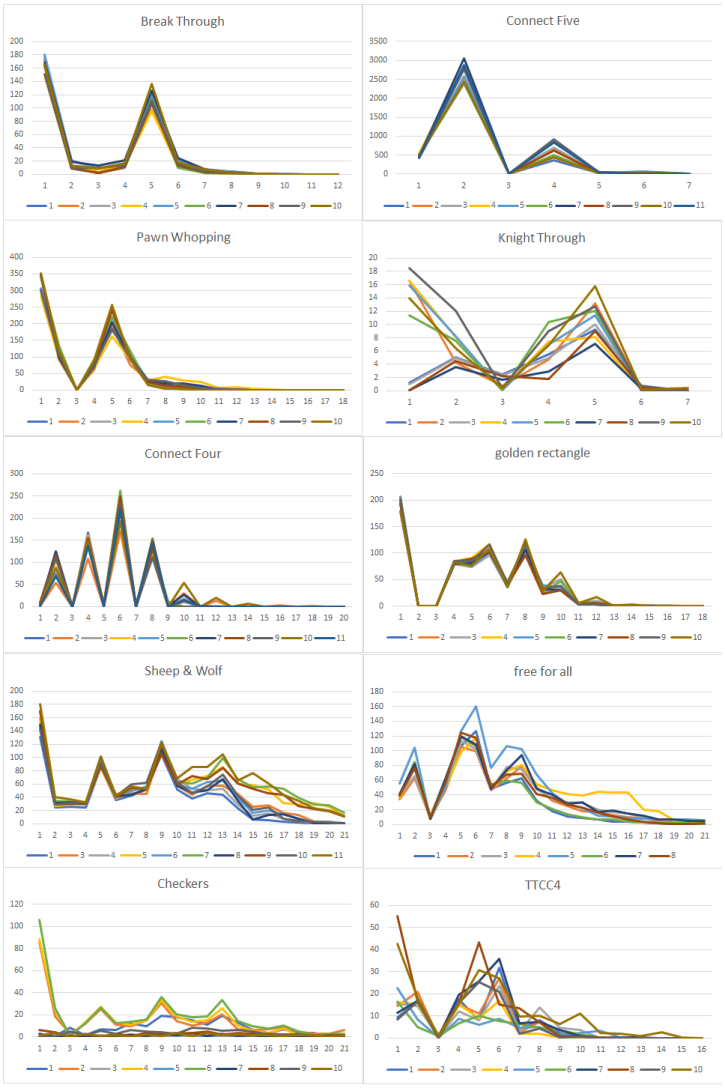

図 10　ハフマン符号のエンコーディングを用いた
1 ターンあたりの類似盤面を取得した回数と拡張盤面との距離

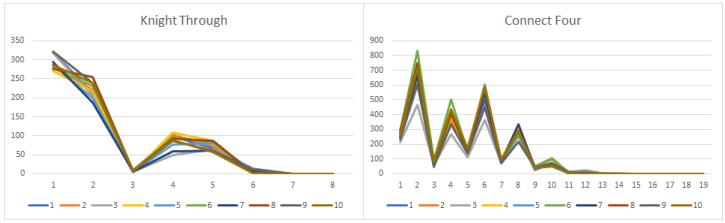

図 11　出現順のエンコーディングを用いた 2 ゲームの
1 ターンあたりの類似盤面を取得した回数と拡張盤面との距離

4.6.2　出現頻度を考慮したエンコードの検証

それぞれのゲームについて、ハフマン符号を利用した時、類似盤面と判定された盤面の真の距離ごとの出現分布を図 10 に示す。ハフマン符号のエンコーディング方法が有効となった Break Through, Connect Five, Pawn Whopping は、距離が 5 以下の範囲にある盤面を探索経験から多く取得していた。これは 4.6.1 のランダムな場合より短い。つまり、これらでは、ハミング距離によってある程度距離の近い盤面を正しく見つけられていたことがわかる。Knight Through も距離 5 以下の範囲で多く取得していたが、盤面取得の距離の傾向がプレイごとであまり安定していなかったことから、勝率の向上が望めなかったのだと考えられる。一方で、図 11 は出現順のエンコードにおける類似盤面の距離ごとの出現分布図であるが、Knight Through は、ハフマン符号と比べて Break Through や Pawn Whopping と同様に近傍での取得する距離の傾向が安定しているため、勝率が向上したのだと考えられる。

また、あまり効果がなかった他のゲームは、取得する盤面の距離が広範囲に渡っていたことや、探索経験から取得する類似盤面の距離の傾向が安定していないことがわかる。この結果から、ハミング距離が実際の距離をうまく反映できないゲームや、プレイごとに挙動が大きく変わるゲームがあることが示された。

5. 事前の準備時間で行うゲームの判別

GGP ではゲーム開始前に各プレイヤーに対してゲームルールと共に数十秒から数分の準備時間が与えられる。本研究では、この準備時間を用いて、提案手法である類似盤面の利用が有効であるか判定し、手法を切り替えることを試みる。

5.1 事前の準備時間での盤面距離の傾向

事前の短い準備時間では、膨大なシミュレーションを行うことでゲームを解析することは難しい。そこで、本研究では各ゲームに対して 10 手までのシミュレーションを複数回行うことで、簡易的にゲームの傾向を把握する。自己対戦の中で MCTS を実行することで、4.6 で行った検証方法と同様に、拡張する盤面に対して探索経験から取得できる類似盤面が構築しているツリー上でどれくらいの距離にあるかを計測した。プレイアウトを 500 回行うシミュレーションを 10 回計測した結果を図 12 に示す。4.6 の検証結果と比較すると、短時間でのシミュレーションでも対戦で計測する時と同じようなゲームの性質を捉えることができると考えられる。

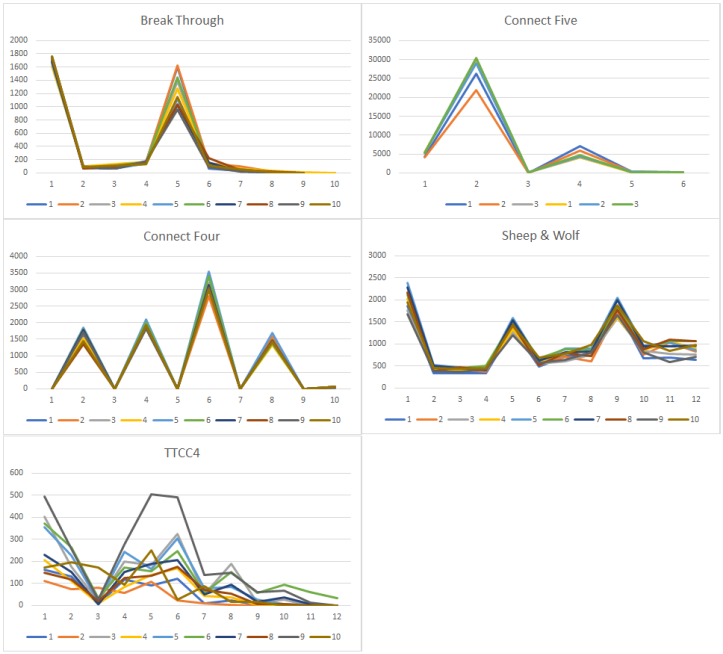

図 12 事前の準備時間で行う簡易的なシミュレーションによる類似盤面を取得した回数と拡張盤面との距離

5.2 類似盤面利用の判定方法

本研究の提案手法である類似盤面の活用が有効であったゲームは、拡張する盤面に対して構築しているツリー上で近い盤面を探索経験から多く取得していた。また上手く作用しなかったゲームの特徴の一つとして、対戦ごとの探索経験から取得する類似盤面の距離の傾向が安定していないことだった。これらのことから短時間でのシミュレーションを行いゲームの性質を把握することで、提案手法の類似盤面が有効であるかを判定することができる指標を考える。事前の準備時間で 10 手までの自己対戦を 10 回実施し、その過程で行うシミュレーションを 500 回と 250 回に設定して、拡張する盤面と探索経験から取得した類似盤面の距離を計測した。

5.2.1 拡張盤面と探索経験から取得した盤面との平均距離

表 4 事前の準備時間及び本番時で行うシミュレーションによる拡張する盤面に対する類似盤面の平均距離

プレイアウト回数	250	500	本番時 (図 10 より)
Connect Five	2.07	2.17	2.27
Knight Through	2.10	2.57	2.66
Break Through	2.23	2.94	2.96
Pawn hopping	2.85	3.27	3.42
TTCC4	3.59	4.06	4.42
Golden Rectangle	4.72	4.84	5.12
Connect Four	4.52	5.11	5.48
Sheep & Wolf	6.02	6.50	8.37
Free For All	5.81	6.60	7.21
Checkers	10.3	4.72	9.34

表 4 は、各ゲームごとに計測した距離の平均を算出したものである。図 10 の、本番で行う MCTS における 500 回のシミュレーションについての平均距離も算出した。出現順とハフマンのいずれかのエンコード方法による類似盤面の利用手法で勝率が向上した Connect Five, Knight Through, Break Through, Pawn hopping より、本番時のの平均距離から閾値を 3.5 と定め、それ以下の時にはハフマン符号化の距離 2 までを採用する方法、それ以外の時は距離 0 を用いる手法と切り替える方法を考える。表 4 の結果から本番時の平均距離は、250 回の時の平均 1.2 倍であるため、プレイアウト 250 回の時の閾値は 2.92 と設定することができる。表 4 より、プレイアウト 250 回の時の平均距離が 2.92 以下のゲームは類似盤面の利用で勝率が向上している。

5.2.2 探索経験から取得する盤面の距離の傾向

表 5 は、距離ごとの盤面の出現割合について 10 回分の計測から標準偏差を求め、それらの平均値を算出した結果である。図 10 の、本番で行う MCTS における 500 回のシミュレーションについての数値も算出した。値が 0 に近いほど、繰り返し同じゲームを行った際に、探索経験から取得する類似盤面の距離の傾向が安定していることを示す。図 10 から、取得する類似盤面の距離がほとんどの長さにおいて安定していなかったゲームは TTCC4, checkers, knight through であった。表 5 の本番時の標準偏差の平均値において、これらの安定していなかったゲームとそうで

表 5 事前の準備時間及び本番時で行うシミュレーションによる拡張する盤面に対する類似盤面の距離ごとの標準偏差の平均

プレイアウト回数	250	500	本番時 (図 10 より)
Golden Rectangle	3.55e-03	4.51e-03	4.10e-03
Connect Four	6.72e-03	4.93e-03	1.58e-03
Pawn hopping	8.50e-03	6.70e-03	2.63e-03
Sheep & Wolf	8.90e-03	6.39e-03	1.51e-03
Connect Five	8.94e-03	1.40e-02	6.73e-03
Break Through	1.62e-02	1.21e-02	3.34e-03
Free For All	1.74e-02	1.78e-02	1.64e-03
TTCC4	3.30e-02	2.10e-02	2.38e-02
Knight Through	3.44e-02	2.94e-02	2.44e-02
Checkers	3.94e-02	6.33e-02	2.68e-02

ないゲームとでは、算出した値は一桁違う結果となった。よって取得する類似盤面の距離の安定性を図る指標として、閾値を 1.0e-02 と定める。表 4 の結果から、プレイアウト 250 回の時の値は本番時の時の 2.6 倍であるため、プレイアウト 250 回の時の閾値は 2.6e-02 と設定することができると考える。表 4 より、プレイアウト 250 回の時の平均距離が 2.6e-02 以上のゲームは、いずれも取得する類似盤面の距離の傾向は安定していない。

5.2.3 類似盤面利用の切り替えを行うプレイヤの勝率

5.2.1 と 5.2.2 の指標から、ハフマン符号で盤面をエンコードする手法における類似盤面の利用に適したゲームは Break Through, Pawn hopping, Connect Five と判断できる。

表 6 ハミング距離を変化させた時とハミング距離を切り替えた場合における 10 種類のゲームの勝率の平均

ハミング距離 0	ハミング距離 2	ハミング距離の切り替え
0.622	0.571	0.631

ハフマン符号でエンコードを行う GGP プレイヤにおいて、表 3 の結果から、ハミング距離 0, 2 の時の勝率の平均値と、Break Through, Pawn hopping, Connect Five の時ハミング距離 2 の類似盤面の取得を行い、それ以外の時は一致する盤面のみの取得を行った際の勝率の平均値を 表 6 に示す。扱うゲームによってハミング距離の切り替えを行うことで、平均して高い勝率となることが示された。

6. おわりに

GGP において少ないシミュレーション回数で勝率を向上させるために、MCTS で展開するノードに対して、その盤面と類似する盤面情報を探索経験から取得し付加する手法を提案した。類似度の判定方法としては、文字列で表現された盤面情報をハッシュ値にエンコードし、ハッシュ値の異なるビット数が少ないほど類似度が高いとした。出現頻度を考慮したエンコード方法では、一部のゲームの勝率が上がることを確認したので、本研究では、3 種類の盤面のエンコード手法について勝率の比較を行い、エンコード手法の効果があったゲームと効かなかったゲームの違いについて検証を行った。その結果、拡張する盤面に対して、構築しているツリー上で近い盤面を探索経験から多く取得しているゲームでは勝率が向上することを明らかにした。

また、類似盤面を利用することで勝率が向上しないゲームは、探索経験から取得する盤面の距離が広範囲に渡っていたことや、試合ごとに取得する類似盤面の距離の傾向が安定していないことを確認した。

そして、事前の準備時間で、簡易的なシミュレーションを行い、拡張する盤面と探索経験から取得する類似盤面の距離を計測することで、実施するゲームに対して類似盤面の利用に有効であるかを判定する仕組みを検討した。類似盤面が有効であると判断する指標として、拡張する盤面と探索経験から取得する盤面とのツリー上の平均距離や、複数回計測した距離ごとの標準偏差の平均値を用いた。

今後の研究では、事前の準備時間で類似盤面の利用が有効でないと判断できた場合に、拡張する盤面から近い距離にある盤面を類似度が高いと判定するエンコード手法の改善ができるような仕組みを検討していく。

参考文献

[1] GGP.org - General Game Playing. http://www.ggp.org/. Accessed:.

[2] 上宮佳晃, 横山大作. General Game Playing における類似盤面を利用したモンテカルロ木探索性能向上の試み. ゲームプログラミングワークショップ 2021 論文集, Vol. 2021, pp. 172 – 178, 2021.

[3] GitHub - ggp-org/ggp-base: The General Game Playing Base Package https://github.com/ggp-org/ggp-base. Accessed:.

[4] N. Love, T. Hinrichs, D. Haley, E. Schkufza, and M. Genesereth.General game playing: Game description language specification, 2008.

[5] L. Kocsis and C. Szepesvári. Bandit based monte-carlo planning. In Proceedings of the European Conference on Machine Learning 2006, pp. 282 – 293, 2006.

[6] H. Finnsson and Y. Björnsson. Simulation-based approach to general game playing. In Proceedings of the 23rd AAAI Conference on Artifical Intelligence, pp. 259 – 264, 2008.

[7] H. Finnsson and Y. Björnsson. Simulation control in general game playing agents. In Proceedings of the International Joint Conference on Artificial Intelligence Workshop on General Game Playing, pp. 21 – 26, 2009.

[8] H. Finnsson and Y. Björnsson, Learning Simulation Control in General Game-Playing Agents, in Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence. AAAI Press, pp. 954 – 959, 2010.

[9] Mandy J.W. Tak, Mark H.M. Winands, and Yngvi Bjornsson “Decaying Simulation Strategies”IEEE Transactions on Computational Intelligence and AI in Games, vol. 6, pp. 395 – 406, 2014.