4Q-5

UCT 証明確率探索による詰碁解法プログラムの研究

松本　清吾†　　丸山　真佐夫†
木更津工業高等専門学校†

１．はじめに

囲碁の対局において，石の生死の判断は重要な要素である．UCT[1]は対局碁でよい成果を上げているが，詰碁は対局碁とは異なる問題であるため，詰碁に適した改良を UCT に加え，性能を向上させることが可能であると考えた．

一方，詰将棋を解くプログラムでは証明数探索[2]の有効性が示されている．

本研究では，証明数探索の考え方を UCT に取り入れた「UCT 証明確率探索」を提案し，その性能を評価することを目的とする．

２．UCT 証明確率探索

本章では，UCT のアルゴリズムをベースとして，証明数探索の考え方を取り入れた探索手法「UCT 証明確率探索」を提案する．

２．１　改善しようとする問題

UCT において，UCB１[3]値が同じノードは等価として扱われる．しかしそれらのノードが展開済みの場合，その子ノードの数やそれぞれの子ノードの勝率の分布によって，証明数探索の考え方を応用して新たな評価を行うことが可能である．

図 2.1 にある局面における UCT での探索の様子を示す．各ノードの上段の数値はその局面のプレイアウト数，下段は勝率を表している．ノードの右側の値は，そのノードの UCB1 値を表している．各ノードの左上の数字は，以降の説明のために便宜的に示したノードの番号である．UCT では，2 つの子ノード②，③は UCB1 値が同じであるため等価であると判断される．

ここで，子ノード②が証明される確率を考える．ノード②を証明するには，その子ノードである孫ノード④，⑤が共に証明されなければならない．ノード④，⑤がもつ勝率を，そのノードが証明できる確率と考えれば，ノード②が証明できる確率はノード④，⑤の証明できる確率（勝率）の掛け算で表すことができる．

ノード③についても同様に，ノード⑥，⑦の勝率の掛け算によってノード③が証明される確率を表すことができる．

それぞれの子ノード②，③の証明できる確率を計算した様子を図 2.1 に合わせて示す．

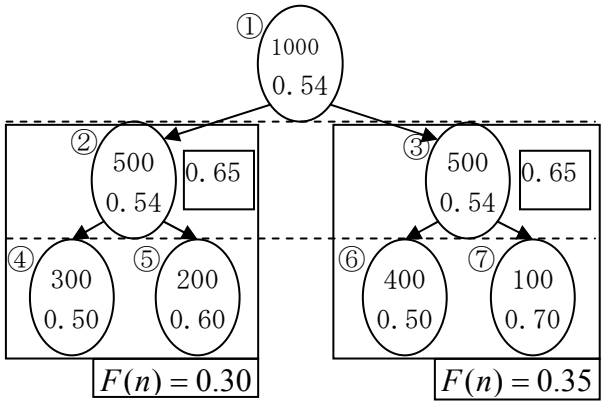

図 2.1　UCT で 2 ノードが等価と判断される例

２．２　UCT 証明確率探索

2.1 節で述べた考え方をまとめると，あるノードの評価式$F(n)$は次式のように表すことができる．

$$F(n)=\prod_{i=1}^{k}\overline{X}(n_i) \quad \cdots (2.1)$$

ここで，nは評価するノードを表し，ノードnの子ノードを（$n_1,n_2,\cdots,n_k$）と表す．$\overline{X}(n_i)$はノードn_iの勝率を表している．

次に末端ノードにおける評価式について考える．展開済みノードであれば式(2.1)によって評価ができるが，未展開ノードでは式(2.1)は適用できない．

しかし未展開ノードであっても，プレイアウトによって勝率の暫定値が求まっている．この勝率を未展開ノードの次の局面の勝率の平均値と考え，未展開ノードnの子ノード数の推測値をmとして，次のように評価する．

$$F(n)=\overline{X}(n)^m \quad \cdots (2.2)$$

これらのことをまとめると，UCT で用いていた UCB1 値に代わって次の式を用いて探索を行う．

$$E(n)=UCB1(n)+cF(n) \quad \cdots (2.3)$$

$$F(n)=\prod_{i=1}^{k}\overline{X}(n_i)$$（nが展開済みの場合）

$$F(n)=\overline{X}(n)^m$$（nが未展開の場合）

ここで$UCB1(n)$はノードnの UCB1 を表す．また，$F(n)$の係数cは未定の係数であり，以後の実験によって検討を行う．

A Study of Solving Tsume-Go by UCT-Proof-Number Search
† Seigo MATSUMOTO, Masao MARUYAMA
Kisarazu National College of Technology

3．実験・考察

3．1　UCT 証明確率探索の探索速度に関する実験

式(2.3)における係数cの値を 0，0.5，1.0 と変えて詰碁[4,5,6]を解いた．プレイアウト数を比較するため，非常に少ないプレイアウト数で解けてしまうものを除外し，22 問の詰碁の結果を提案手法と UCT の比較に用いた．UCT と比較して良い結果，悪い結果，同程度の結果の問題の内訳を表 3.1 に示す．

表 3.1　実験結果の内訳

	提案手法(c=0.5)	提案手法(c=1)
良い	10	10
同程度	11	10
悪い	1	2

総プレイアウト数に対する正解確率はそれぞれの係数cの値によって異なり，係数cによる探索効率への影響が観察できた．

90％の正解確率を得るのに必要なプレイアウト数を提案手法と UCT とで比較した．通常の UCT のプレイアウト数を 1 としたときのそれぞれの係数におけるプレイアウト数の比の平均値を図 3.1 に示す．

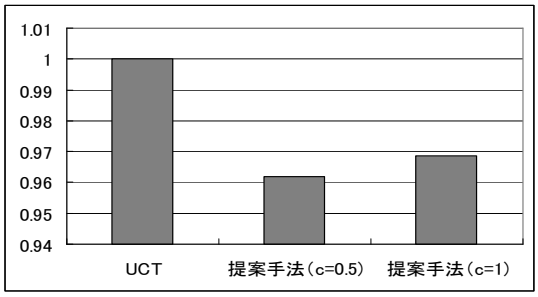

図 3.1　プレイアウト数の比較（UCT を 1 とした相対値）

図 3.2 に提案手法が良い結果を残した詰碁での正解確率の変化を示す．また，図 3.3 に UCT の方が悪い結果を残した詰碁での正解確率の変化を示す．

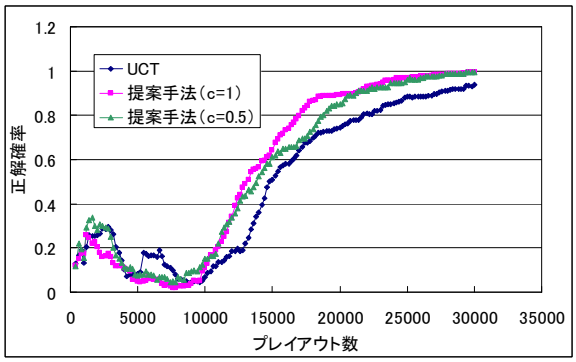

図 3.2　良い結果の詰碁の正解確率の変化

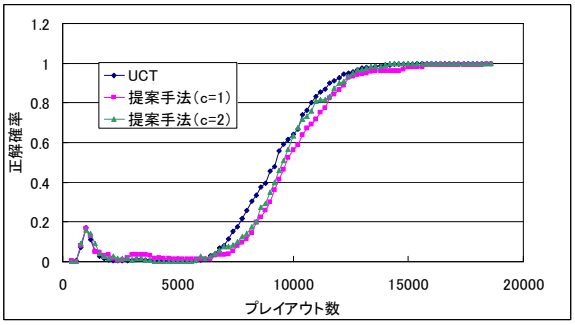

図 3.3　悪いの詰碁の正解確率の変化

3．2　探索速度に関する実験に対する考察

提案手法では，2.1 節で述べたように，最も証明しやすそうなノードを選択する要素を UCT に取り入れている．この予測が当たっていれば探索速度は速くなるが，当たらなければ遅くなる．

黒の正解手に対する白の応手のうち，難しい物が少なければ評価式(2.1)は大きな値を出力しやすくなり，正解の手が探索されやすくなる．

反対に，誤りである手に対して白が打つ応手に難しい手が多ければ，評価式(2.1)の値が小さくなるため選択されにくくなる．実際に提案手法が良い結果を残した詰碁では，前述の特徴が見られた．

提案手法が悪い結果を残した図 3.4 の詰碁では，誤りの手に対する白の難しい応手が少ない特徴が見られた．

4．おわりに

本研究では詰碁の問題を UCT を用いて解くために，証明数探索の概念を UCT に取り入れた UCT 証明確率探索を提案し，UCT と比較した．

実験の結果，提案手法（c=0.5）は UCT に対して平均約 96%のプレイアウト数で UCT と同程度の正解確率を得ることができた．

UCT では使われていなかった子ノードの勝率を用いることによって，UCT の探索速度を向上させることができた．

参考文献

[1] L. Kocsis and C. Szepesvari.: Bandit-based monte-carlo planning, Lecture Notes in Computer Science vol. 4212, pp.282-293, 2006.
[2] L.V.Allis, M. van der Meulen, and H. J. van den Herik: Proof-Number Search, Artificial Intelligence, Vol. 66, pp.91-124, 1994
[3] P. Auer, N. Cesa-Bianchi, and P. Fischer: Finite-time analysis of the multiarmed bandit problem, *Machine Learning*, 47(2/3), pp.234-256, 2002
[4] 藤沢一就：ひと目でわかる ポケット詰碁 200，日本棋院，1994
[5] 日本コンピュータ囲碁協会，テスト用の詰碁基本問題，http://www.geocities.jp/otto4yoshi/jcga/indexkp.html
[6] 石田芳夫：基本の詰碁，成美堂出版，2006