証明数と反証数を用いた λ 探索

副田俊介[†1] 美添一樹[†2] 岸本章宏[†1]
金子知適[†3] 田中哲朗[†4] マーティン ミュラー[†5]

本論文では脅威度と証明数・反証数の双方を利用する df-pn λ 探索を提案する．脅威度と証明数・反証数は，両者とも AND/OR 木探索を効率的に行うための指標であり，脅威度を利用する λ 探索と，証明数・反証数を利用する df-pn 探索は，どちらも優れた探索アルゴリズムである．脅威度と証明数・反証数の双方を用いることで，探索をさらに効率的に行うことは自然なアイデアである．脅威度と証明数・反証数の関係を結び付けるために，本論文では各節点ごとに，各脅威度に対応する疑似節点をモデル化する．そのうえで，疑似節点上の証明数・反証数を用いて，元の節点の証明数・反証数を定義することを提案し，それにより探索が制御されることを示す．さらに，性質の異なる複数のゲームを対象に，df-pn λ 探索と df-pn 探索の性能を比較する実験を行った．その結果，将棋や囲碁において df-pn λ 探索は df-pn 探索よりも性能が良いことを確認した．

λ Search Based on Proof and Disproof Numbers

Shunsuke Soeda,[†1] Kazuki Yoshizoe,[†2] Akihiro Kishimoto,[†1]
Tomoyuki Kaneko,[†3] Tetsuro Tanaka[†4] and Martin Müller[†5]

We present the df-pn λ search algorithm that combines threats with proof and disproof numbers. λ search is a promising method based on threats. Df-pn is an efficient algorithm that employs the notion of proof and disproof numbers. However, λ search uses neither proof nor disproof numbers, whereas df-pn incorporates no information on threat levels. Integrating threats with proof and disproof numbers is a natural extension to further enhance the search performance. We introduce pseudo-nodes for various threat levels at each node, to represent a node searched with a specific threat level. Then the proof and disproof numbers of the original node are defined using pseudo-nodes, which provides a model that can be searched with df-pn. We compared df-pn λ with df-pn on games with different properties. The results showed that df-pn λ is better than df-pn in Shogi and Go.

1. はじめに

囲碁や将棋などの複雑なゲームで強いプログラムを作るには，将棋の詰みや囲碁の石の捕獲など，部分目標の探索の活用が重要である．一般に，2 人ゲームのプログラムでは，ゲーム木探索を用いて次に指す手を決定する．一方，目的を部分目標の実現に絞った場合には，ある問題（局面）に対する真偽値（成功か否か）を求める AND/OR 木探索の利用によって，効率的な探索が可能である．これまで AND/OR 木探索は，ゲームを題材にして著しい進歩をとげてきている．

df-pn 探索[8] を代表とする証明数と反証数[2] を用いた手法は，最も成功を収めている．この手法では，展開途中の木の形から，局面の解きやすさ（解きにくさ）を推定する証明数（反証数）を利用して，解きやすい節点を優先的に展開し，効率良く解を得る．

λ 探索[11] を代表とする脅威度に基づく探索は，五目並べを解くことに成功した DB-search[1] を一般化した手法である．脅威度は，片方のプレイヤがパスした後の情報を利用して，局面がどれだけ解に近いか（λ 次数）を推測したものである．この手法では，局面が解に近い状態ほど指し手が限定されるという，多くのゲームに共通する性質に基づいて効率良く探索を行う．

†1 公立はこだて未来大学
Future University-Hakodate
†2 中央大学研究開発機構
Center of Research and Development Initiative, Chuo University
†3 東京大学大学院総合文化研究科
Graduate School of Arts and Sciences, The University of Tokyo
†4 東京大学情報基盤センター
Information Technology Center, The University of Tokyo
†5 アルバータ大学コンピューティング・サイエンス学科
Department of Computing Science, University of Alberta

両者は，簡単に解けそうな節点の優先的な展開によって性能を上げる点は共通であるが，解への近さに対する指標（脅威度と証明数・反証数）が異なる．片方の指標では同じ優先度を持つ節点でも，もう片方の指標では優先度が異なることがあり，片方の指標のみを使った場合はもう片方の優先度が低い節点を展開してしまうことがある．

そこで，本論文では，df-pn 探索と λ 探索の長所を組み合わせた，df-pn λ 探索を提案する．両者の組合せによって，脅威度と証明数・反証数の片方のみに基づく探索の欠点を補える．この組合せ自体は，副田[10),19)]と Yoshizoe[12)] の研究に基づいているが，本論文では両者をまとめた枠組みを提案し，さらに AND 節点における証明数の計算について複数の戦略で実験を行い議論する．実験には，性質の異なるゲームとして，囲碁と将棋とシンペイを用いた．将棋で扱った詰将棋と必至は df-pn 探索が有効な例題であり，囲碁の石の捕獲問題は λ 探索の対象として研究されている．また，シンペイは完全に解かれたゲームであるので，アルゴリズムの振舞いを調べるのに理想的な題材である．

本論文の構成は以下のとおりである．次章で，関連研究について紹介した後，3 章で df-pn λ 探索について説明する．4 章で各ゲームでの実装の詳細を説明した後，5 章で実験結果を示し，最後に結論を述べる．

2. 関 連 研 究

2.1 λ 探 索

多くのゲームには，プレイヤが達成したい目標がある．この目標を**ゴール**と呼ぶ．ゴールには，ゲームに勝つという大局的なものから，局面を有利にするための部分目標など様々なものがある．AND/OR 木探索の利用によって，あるゴールを達成できるかどうかを調べられる．本論文では，ゴールを達成したいプレイヤを**攻め方**，阻止したいプレイヤを**受け方**と定義する．

攻め方がゴールを達成しようとする手を指したとき，受け方が正しい応手を選ばなければ，攻め方はすぐにゴールを達成してしまう．つまり，受け方は，攻め方のゴールを阻止する手を選ばなければならない．このように，受け方の手を制限する要素を**脅威度**と呼ぶ．

脅威度の利用によって，AND/OR 木探索が考慮すべき指し手の集合を減らすことができる．脅威度の概念は，Allis らの DB-search に利用され[2)]，その後，Thomsen や Cazenave らによって一般化された[3),11)]．

Thomsen の λ 探索では，受け方が何回パスすれば攻め方がゴールを達成できるかで，脅威度を定義している．λ 探索における受け方がパス可能な回数 n を，**λ 次数 n** と呼ぶ．n が小さければ小さいほど，受け方にとってより脅威の大きい局面である．

λ 次数が n のときにゴールが達成された場合は $\lambda^n=1$，達成されない場合には $\lambda^n=0$ で表す．λ 次数 n における λ 探索を，**λ^n-手**と **λ^n-木**を利用して定義する．ここで，攻め方の λ^n-手を **λ^n_a-手**，受け方の λ^n-手を **λ^n_d-手**とする．また，攻め方の手番の節点を根節点とする λ^n-木を **λ^n_a-木**とする．λ^n-手および λ^n-木は次のように定義される[11)]：

(1) λ^0_a-手は，直接ゴールが達成される攻め方の手である．また，λ^0_a-木は λ^0_a-手のみで構成される木である．λ^0_a-手を持つ λ^0_a-木は $\lambda^0=1$ であり，それ以外の λ^0_a-木は $\lambda^0=0$ である．

(2) **λ^n-木（$n\geq 1$）の終端節点**の値は，攻め方が手番のときには $\lambda^n=0$，受け方が手番のときには $\lambda^n=1$ である．

(3) **λ^n_a-手**と **λ^n_d-手**（$n\geq 1$）は，λ^i_a-木（$0\leq i\leq n-1$）を用いて，次のように定義される：
λ^n_a-手は，λ^n_a-手の直後に受け方がパスをすれば，$\lambda^i_a=1$ となる λ^i_a-木が存在する手である．つまり，受け方がパスをすれば，攻め方は λ^n_a-手によって，λ 次数 n 未満でゴールを達成できることを保証している．一方，λ^n_d-手は，λ^n_d-手の直後に $\lambda^i_a=1$ となる λ^i_a-木が存在しない手である．λ^n_d-手によって，受け方は n 未満の λ 次数でゴールを達成する脅威を防いだことになる．

(4) **λ^n-木**（$n\geq 1$）は λ^n-手で構成される AND/OR 木である．

将棋における λ^n-手の例を示す．λ^0_a-手は，攻め方が受け方の玉を取る手に対応する．λ^1_a-手は，もし受け方がパスをすれば，攻め方は受け方の玉を取ることができるという定義である．よって，λ^1_a-手は攻め方が受け方に王手をかける手であり，λ^1_d-手は王手に対する受け手である．同様に，λ^2_a-手は攻め方が受け方に詰めろ（受け方がパスをすれば詰むこと）をかける手であり，λ^2_d-手は詰めろから逃れる手である．

λ-木では，通常の AND/OR 木の性質に加えて，次の優越関係が成り立つ：

$$\lambda^n=1 \Longrightarrow \lambda^i=1 \quad (n\leq i) \qquad (1)$$

$$\lambda^n=0 \Longrightarrow \lambda^j=0 \quad (1\leq j\leq n) \qquad (2)$$

λ 探索の結果は，パスが合法手である，またはパスが最善手となる局面がない場合には，つねに正しい．つまり，$\lambda^n=1$ ならば，つねにゴールが達成でき，$\lambda^n=0$ ならば，ゴールは達成できないか，より高い λ 次数の脅威度でのみ達成できるかのどちらかである．

実際の λ 探索は，λ-木を構築しながら行う．そこ

で，部分的に構築された λ-木のどこを展開し，いくつの λ 次数で探索するかについては様々な戦略が存在しうる．効率的な探索のためにはこれらの制御が重要であるが，これらの研究はなされていない．

2.2 df-pn 探索

Nagai の df-pn 探索[8] では，最良優先探索である Allis の証明数探索[2] を，反復深化を利用して効率化している．df-pn は，証明数探索と比べ，内部節点の再展開数が少なく，探索した局面をすべて持っておく必要がないという優れた性質を持つ．df-pn は，詰将棋[8] や詰碁[7]，チェッカ[9] など，様々なゲームで有効である．

df-pn は，Allis の**証明数・反証数**を利用している．本論文では，攻め方を OR 節点，受け方を AND 節点とする．さらに，ゴールが達成された場合を**証明された**，阻止された場合を**反証された**呼ぶことにする．節点 n の証明数 $pn(n)$（反証数 $dn(n)$）は，n を証明（反証）するために展開しなければならない葉節点数の最小値として定義される．$n_1, \cdots, n_k$ を n の子節点とすると $pn(n)$ と $dn(n)$ は，次のように計算できる．

(1) n が証明された終端節点のとき，$pn(n) = 0$, $dn(n) = \infty$.

(2) n が反証された終端節点のとき，$pn(n) = \infty$, $dn(n) = 0$.

(3) n が未展開節点のとき，$pn(n) = dn(n) = 1$.

(4) n が内部 OR 節点のとき，

$$pn(n) = \min_{i=1,\cdots,k} pn(n_i),\ dn(n) = \sum_{i=1}^{k} dn(n_i).$$

(5) n が内部 AND 節点のとき，

$$pn(n) = \sum_{i=1}^{k} pn(n_i),\ dn(n) = \min_{i=1,\cdots,k} dn(n_i).$$

証明数（反証数）が小さな節点ほど，証明（反証）しやすいと予測できる．df-pn では，OR 節点では最小の証明数を持つ子節点を，AND 節点では最小の反証数を持つ子節点を選択して到達できる葉節点を展開する．この手順を根節点が解けるまで繰り返す．

なお，上記のように計算された証明数·反証数は，合流やループのあるゲーム木では本来の定義の証明数・反証数とは異なる．しかし，この値を用いても探索は進行し，正しい結果が得られることが示されている[5]．

3. Df-pn λ 探索

提案する df-pn λ 探索は，df-pn と λ 探索の長所である証明数・反証数と λ 次数の双方を考慮することで，より効率的な探索を実現する．なお，λ 次数を利用するため，根節点においてどの λ 次数で探索を行うべきかを与える必要がある．たとえば，λ 次数 2 では根節点の証明ができず，λ 次数 3 で証明ができる場合には，根節点以下を λ 次数 3 で探索すれば効率的である．

本章では，最初に**疑似節点**と OR 節点での探索制御について説明する．次に，合計戦略[10] と最高次戦略[12] という AND 節点での探索制御について説明する．

3.1 疑似節点の利用

df-pn では節点ごとに 1 つの証明数と反証数を定義していた．df-pn λ 探索では，各節点 n の各 λ 次数 i に対する証明数 $pn^i(n)$ と反証数 $dn^i(n)$ を用いる．

$pn^i(n)$ と $dn^i(n)$ の定義は**疑似節点**に基づくため，まずそれらについて説明する．節点 n の疑似節点 c_l とは，λ 次数 l の探索を行うために，n 内に仮想的に作られた節点である．c_l は，n と同じ局面であるが，λ^l-木を選択すること☆を明示的に表している．n 内には様々な λ 次数の疑似節点が存在する．

λ 次数が 3 の場合の n の疑似節点を図 **1** に示す．この例では，OR 節点には λ 次数が 0 から 3 の疑似節点があり，AND 節点には λ 次数が 2 と 3 の疑似節点が存在する．各節点における疑似節点の構成方法につ

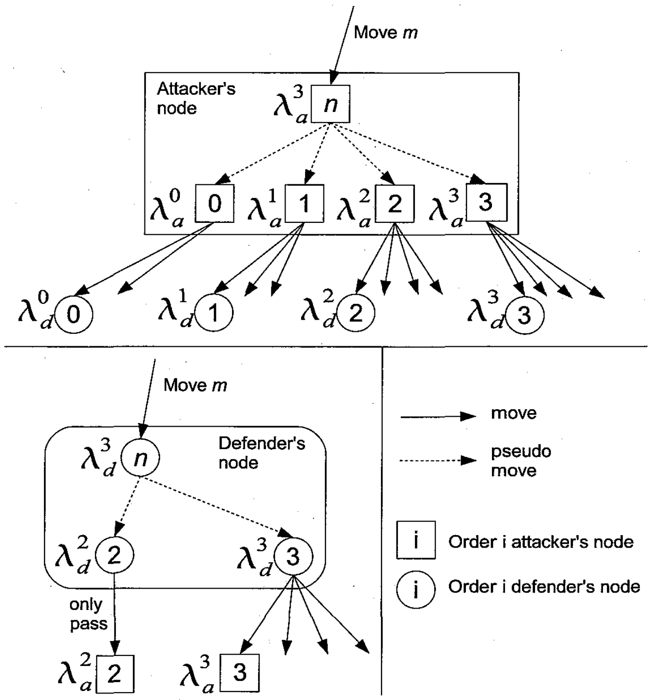


図 1 OR 節点と AND 節点
Fig. 1 OR-node and AND-node.

☆ 実装上は，c_l 用に n の局面情報をあらためて持つ必要はなく，トランスポジションテーブル内に λ 次数 l と証明数や反証数を保持しておけばよい．

いては，3.2 節および 3.3 節で取り扱う．

3.2 OR 節点の取扱い

OR 節点 n において探索の λ 次数を l とする．df-pn λ 探索では，OR 節点では，λ 次数 0 から l までの疑似節点を用意する（λ 次数 3 の例として，図 1 を参照）．

2 節の式 (1) より，λ^i-木（$i < l$）が証明されれば，n の証明は終了する．一般的には，λ^i-木は，λ^l-木よりも探索空間が小さいことが多いため，OR 節点では，できるだけ低い λ 次数で探索を行えば，高速に探索が終了する可能性がある．しかし，λ^i-木が反証されたときには，λ^l-木も探索する必要がある．この場合には，λ^i-木の探索は，n の証明には不要な探索である．さらに，多くのゲーム木では，λ^i-木の反証の方が，λ^l-木の証明よりも難しいことが多い．

そこで，λ^i-木（$i < l$）の探索をほどよく行うために，df-pn λ 探索では**証明数拡幅戦略**[10] を用いて，証明数と反証数を定義し，探索を制御する．$n_1, \cdots, n_k$ を λ 次数 l の OR 節点 n の子節点としたとき，n と疑似節点 c_i の証明数と反証数は以下のように定義される：

$$
\begin{aligned}
pn^i(c_i) &= \min_{j=1,\cdots,k} pn^i(n_j),\\
dn^i(c_i) &= \sum_{j=1}^{k} dn^i(n_j),\\
pn^l(n) &= \min_{i=1,\cdots,l} pn^i(c_i),\\
dn^l(n) &= dn^l(c_l)
\end{aligned}
$$

証明数拡幅戦略では，$pn^i(c_i)$ と $dn^i(c_i)$ の計算は従来の証明数と反証数と同様である．一方，$pn^l(n)$ には最小の証明数を持つ疑似節点の証明数を，$dn^l(n)$ には最大の λ 次数を持つ疑似節点の反証数を用いる．これは，λ 次数 l で節点 n を証明するためには最も証明しやすい λ 次数から探索を行えばよく，反証するためには λ 次数 l で反証しなければならないことに基づく．

n の子節点の選択は，証明数に基づく．疑似節点のうち，最も小さな証明数を持つ疑似節点を先に展開する．

3.3 AND 節点の取扱い

λ 次数 l の AND 節点 n では，通常の指し手に加え，パスも合法手であるとする．パス後の探索は，λ^{l-1}-木に移行する（λ 次数 3 の例として，図 1 を参照）．受け方が反証する手段は，λ 次数 l の指し手の 1 つを反証するか，またはパス後の節点の脅威度が $l-1$ より高いことを示すかの 2 通りである．一般には，λ^{l-1}-木は，λ^l-木よりも小さいが，λ^{l-1}-木が証明されたときには，λ^l-木の結果を調べる必要がある．どちらから反証するかにはトレードオフが存在する．一方，n を証明する際には，λ^l-木を証明する必要がある．

$n_1, \cdots, n_k$ を n の子節点としたとき，n の反証数，疑似節点 c_i の証明数 $pn^i(c_i)$ および反証数 $dn^i(c_i)$ は以下のように定義される：

$$
\begin{aligned}
dn^l(n) &= \min(dn^{l-1}(c_{l-1}), dn^l(c_l)),\\
pn^i(c_i) &= \sum_{j=1}^{k} pn^i(n_j),\\
dn^i(c_i) &= \min_{j=1,\cdots,k} dn^i(n_j).
\end{aligned}
$$

n の子節点の選択は，反証数に基づく．λ^{l-1}-木と λ^l-木の中で，反証の容易そうな節点を先に展開する．

n の証明数に関しては，パスの取扱い方によって，2 つの手法が考えられる．合計戦略[10] では，パス以下の局面を実際に探索するという事実より，パスを通常の指し手と同一に取り扱う．最高次戦略[12] では，パスは証明には無関係な指し手であるので，パス後の証明数を無視する．両者の $pn^l(n)$ は次のとおりである：

$$
\begin{aligned}
pn^l(n) &= pn^{l-1}(c_{l-1}) + pn^l(c_l) &&\text{（合計戦略）}\\
pn^l(n) &= pn^l(c_l) &&\text{（最高次戦略）}
\end{aligned}
$$

合計戦略と最高次戦略には，それぞれ利点と欠点がある．合計戦略では，$pn^{l-1}(c_{l-1})$ の加算によって n の証明数が大きくなるため，パス以下の探索が証明に有効でない場合には n の探索を後回しにできるという利点がある．その一方で，証明数を大きく見積もることで必要な探索が後回しになる場合もある．一方，最高次戦略では，合計戦略とは逆のことが生じる．両者の性能は，探索空間の性質に依存するので，どちらが優れた方法であるかは一概にはいえない．4 章では，各ゲームについて，どちらの戦略を選択するべきかをさらに議論する．

4. df-pn λ 探索の各ゲームへの適用

提案する手法の有効性を調べるために囲碁，将棋，シンペイの 3 つのゲームを対象として，df-pn λ 探索と df-pn 探索アルゴリズムを実装した．将棋では合計戦略を，囲碁とシンペイでは最高次戦略を利用している．各ゲームでの実装の詳細を以下に述べる．主に対象とするゴールと現実的な λ 次数，およびゲームに依存した指し手生成が可能かどうかが異なっている．

各探索アルゴリズムおよびすべてのゲームに共通する効率化として，証明数のヒューリスティックな初期化[8] を利用している．通常の df-pn 探索において末端

節点の証明数・反証数を1とするが，1に代えてゲーム依存の知識などを利用して予測した値を用いることで探索を効率化する．

4.1 将棋における実装の詳細

ゴールとしては必至（λ 次数が2）と詰（λ 次数が1）を用いた．それ以上の λ 次数は2手すき以上の手を含む必至が対応するが，探索空間が広すぎて現実的でないためである．指し手生成は，λ^1-手は盤面から効率良く生成することが可能であり，そのように実装した．λ^2-手は探索を行わない限り判定することができないため，解の正しさを保証するために全合法手を生成した．このため，必至を解く場合には分岐数が100を超える局面が数多く存在する．また，将棋ではパスは合法手ではないが，手番が移動すると考えて扱う．

必至を解く際に生成される手の大部分は λ^2-手ではない．そこで，パスを先に探索する手法[10]を用いて探索性能の向上を図る．AND 節点では，まずパスを探索し，パスの後の局面の探索結果が判明するまでは，他の手の探索を保留する．パスの後の局面が証明された場合は，シミュレーション[4]を利用して，パス以外の指し手が受けになっているかを判定することで，受け方の手の枝刈りができる．また，パス後の局面が反証された場合は，元の節点は λ^2-木に含まれないため即座に枝刈りが可能となる．なお，パスを先に探索する手法と最高次戦略と組み合わせた場合，探索が進まないことが生じるため，将棋では合計戦略を用いる．

また，局面の優越関係や合駒処理などの将棋特有の手法[4),8),15]を用い，末端局面で3手詰め[16]の判定を行う．

4.2 囲碁における実装の詳細

囲碁では，石の捕獲問題を対象にしている．生成する着手は，捕獲対象の石の周囲の知識を利用して制限している．ダメおよびダメの1路周囲の空点，攻め合いに関与する連（*surround blocks*[11]）のダメなどからなる候補手を生成した．攻め合いに関与する連は λ 次数に応じて増えるため，高い次数での探索では候補手の数も増える．また，このように生成された手の多くは脅威度を持つため，合法手すべてを使う場合と比べて，パス以降の探索の有効性が少ないと予想される．そこで，囲碁では最高次戦略を用いた．

また，λ-木の定義に用いられるパスとは別に，通常の合法手としてもパスを探索する．ただし，そのようなパスはほとんどの局面では最善手とならないため，他の手の結果が判明するまで探索を保留する．

囲碁のシチョウは，捕獲問題で λ 次数が1の場合である．シチョウ判定のための着手生成は，非常に簡単であるので，通常探索よりも10倍程度高速なシチョウ探索ルーチンを実装した．df-pn λ 探索では，λ^1-木探索にこのシチョウ探索ルーチンを利用している．これは現在探索する λ 次数が明示的に分かっているために利用できる工夫である．

さらに，AND 節点において，ある手の後の局面が証明されたときには，シミュレーションを利用して，他の着手の証明の判定を行う．

4.3 シンペイ

シンペイではゲームに勝つことをゴールとし，全合法手を用いた．また，λ 次数を求めるためにパスを行う必要があるが，シンペイのルールではパスが許されない局面が多いため，パスを次のように扱った[19]：手元に駒がなければ，単に手番を交替する．手元に駒がある場合は駒を1つ捨てその駒は2度と利用できない．また，その後の局面で，盤面の自分の駒が4個未満でも手元の駒がない場合は盤面の駒を動かす．

予備実験で最高次戦略と合計戦略を比較した結果，シンペイでは最高次戦略の方が性能が良かった．そこで，本論文でもシンペイは最高次戦略で実験を行う．

5. 実験

将棋[10]，囲碁[12]およびシンペイ[17),19]を用いて，3章および4章で説明した df-pn λ 探索と，最新の最適化手法[4),6),7]を取り込んだ df-pn 探索の性能比較実験を行った．

なお，各実験で探索する λ 次数の上限を**探索 λ 次数**，問題を解くのに必要な最小の λ 次数を**実際の λ 次数**と呼ぶ．

5.1 将棋

将棋では，次の必至問題と詰問題を実験対象とした．[必至] は長井の研究[8]で利用された32題[13]で，λ 次数1では反証となり，λ 次数2で証明可能な問題である．[詰] は実戦の棋譜[14]から抽出した777題[16]で，λ 次数1で証明可能な問題である．実験には，Opteron 280（2.4 GHz）のマシンを用いた．また，置換表のサイズは，100万に制限した．

df-pn は，探索 λ 次数が2の場合には4.1節で説明した df-pn λ 同様に，全合法手を生成し，AND 節点ではパスを初めに探索することで枝刈りを行う．

まず，実際の λ 次数を2に設定して，解答能力を比較した．各手法の解答数の比較を**表1**に，両手法で解けた問題の解答時間の比較を**図2**に示す．[必至]，[詰] ともに df-pn λ の方が df-pn よりも解答能力が高い．また，多くの問題で，df-pn λ の方が高速に解けた．特に，[詰] の中には，df-pn が df-pn λ に比

表 1 将棋における正解数の比較
Table 1 Number of problems solved in Shogi.

	df-pn λ	df-pn
詰	702	646
必至	24	21

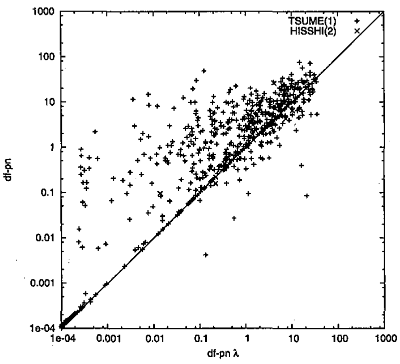


図 2 将棋における探索時間の比較
Fig. 2 Comparison of the search time in Shogi.

表 2 将棋における探索 λ 次数ごとの探索時間（sec）の比較
Table 2 Search time (sec) for each λ order in Shogi.

実際の λ 次数	—	1	2
1（詰（642 問））	df-pn λ	0.0399	2.78
1（詰（642 問））	df-pn	0.0399	4.91
2（必至（20 問））	df-pn λ	(1.17)	3.91
2（必至（20 問））	df-pn	(1.17)	5.41

べ，1,000 倍程度の時間を要するものも存在した．

次に，探索 λ 次数を 1 と 2 に設定し，探索の性質を調べた．**表 2** は df-pn λ と df-pn の双方で解けた問題について，性能を平均解答時間で示したものである．

探索 λ 次数が 1 の場合には，両手法は詰将棋探索を行うので，表中の実行時間が同じになる．また，探索 λ 次数が 1 のときには，両手法は，［必至］に対して反証を返す．この場合の平均実行時間を括弧内に記述した．探索 λ 次数を 2 としたとき，df-pn λ は df-pn と比べ，［詰］は半分強の時間で，［必至］も 7 割強の時間で解答し，より良い性能を示した．

これらの実験より，将棋では実際の λ 次数があらかじめ判明している問題でも不明な問題でも，df-pn λ 探索が df-pn 探索と比べて優れているといえる．

5.2 囲　　碁

囲碁の実験には問題集「攻め合いの達人」[18] から選んだ捕獲問題 110 問を用いた．攻め合いの達人は全 148 問からなる問題集で，難易度は級位者向けの簡単なものから，アマ高段者向けの難問まで幅広い．その中から 9 路盤に入る問題 110 問を選択した．実験には，Opteron 870（2.0 GHz）のマシンを用いた．また，置換表のサイズは，100 万に制限した．

なお，df-pn は λ 次数のような捕獲を打ち切る条件

表 3 囲碁における解けた問題数の比較
Table 3 Number of problems solved in Go.

—	df-pn λ	df-pn
解答数	77	75

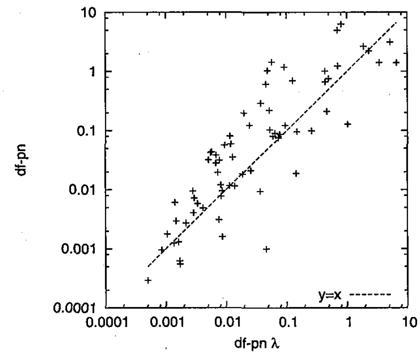


図 3 囲碁における探索時間の比較
Fig. 3 Comparison of the search time in Go.

表 4 囲碁における λ 次数ごとの探索時間（sec）の比較
Table 4 Search time (sec) for each λ order in Go.

実際の λ 次数	—	2	3	4	5
2（35 問）	df-pn λ	0.0118	0.0112	0.00223	0.237
2（35 問）	df-pn	0.180	0.182	0.364	0.428
3（16 問）	df-pn λ	—	0.00415	0.290	0.295
3（16 問）	df-pn	—	0.148	0.612	0.864
4（7 問）	df-pn λ	—	—	1.77	1.84
4（7 問）	df-pn	—	—	0.792	1.32

を持たないため，捕獲対象のダメ数が対応する λ 次数 +2 に達したら捕獲不能と見なすようにしている（df-pn λ は根節点では，λ 次数 +1 のダメ数の連まで，捕獲を試みる性質がある）．

まず，探索 λ 次数を 5 に設定し，df-pn λ と df-pn の解答能力を比較した．解けた問題数を**表 3** に，df-pn λ，df-pn 双方で解けた問題についての解答時間を**図 3** に示す．df-pn λ の方が多くの問題を解いた．また，全体的に，df-pn λ の方が速い傾向にある．

続いて，将棋での実験と同様に，探索 λ 次数ごとの，df-pn λ と df-pn の性能を比較した．その結果を**表 4** に示す．実際の λ 次数が低い場合は，たとえ無駄な探索をする場合でも df-pn λ は df-pn を上回る性能を示している．逆に，実際の λ 次数が高い問題では df-pn が良い性能を示した．

これらの実験により，囲碁では実際の λ 次数の低い問題では df-pn λ 探索の方が，実際の λ 次数の高い問題では df-pn 探索の方が優れているといえる．

5.3 シ ン ペ イ

シンペイでは，全局面の 1,000 分の 1 の局面をランダムに取り出したうち，勝ちとなる局面を実験対象にした．実験には，Opteron 280（2.4 GHz）のマシン

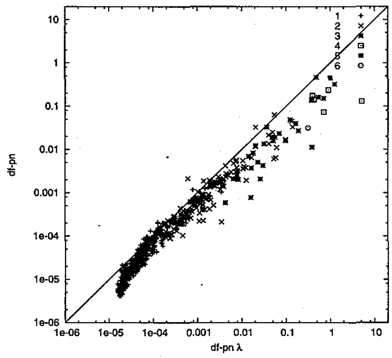


図 4 シンペイにおける探索時間の比較
Fig. 4 df-pn λ vs. df-pn in Simpei.

表 5 シンペイにおける λ 次数ごとの探索時間の比較
Table 5 Search time (sec) for each λ order in Simpei.

実際の λ 次数	df-pn	df-pn λ
1 (3,470 問)	6.97×10^{-5}	1.08×10^{-4}
2 (1,115 問)	1.09×10^{-2}	9.52×10^{-3}
3 (258 問)	6.63×10^{-2}	1.80×10^{-1}
4 (35 問)	1.41×10^{-1}	1.16
5 (1 問)	4.50×10^{-1}	1.01
6 (1 問)	3.19×10^{-2}	3.11×10^{-1}

表 6 シンペイにおける λ 手の平均
Table 6 The average number of λ-moves in Simpei.

λ 次数	1	2	3	4	5 以上	平均合法手
手の数	7.718	8.227	3.416	0.875	5.435	25.673

を用いた．置換表のサイズは 100 万節点に制限した．

探索 λ 次数を 7 に設定した df-pn λ 探索（df-pn λ）と df-pn 探索（df-pn）を比較した．

この制限のもとで，df-pn λ と df-pn は実際の λ 次数が 1 から 6 までの合計 4,880 局面を解くことができた．これらの局面について，各問題の解答時間を，df-pn λ と df-pn で比較したものを**図 4** に示す．また，実際の λ 次数ごとに分類した各手法の平均解答時間を**表 5** に示す．全体的に df-pn の方が df-pn λ よりも速い傾向にあり，実際の λ 次数が 2 の問題でのみ df-pn λ の方が性能が良い．

df-pn λ の性能低下については，λ 次数による選択的探索と，脅威度を利用するコストのトレードオフが原因であると考えられる．**表 6** に，手番プレイヤが勝ちとなる全局面☆について λ^i-手の平均数を示す．8 割弱の手の λ 次数が 3 以下であるため，λ 次数 3 以上で探索した場合には実質的には全幅探索に近い探索となり，λ 次数を考慮するメリットがなかったと考えられる．

☆ ただし，直接勝つ手が存在する局面（次数が λ 0 の局面）は除いてある．

これらの実験より，合法手の分布が偏っていないところでは λ 次数を考慮する価値があることを確認した．

6. 結 論

本論文では脅威度と証明数・反証数の双方を利用する df-pn λ 探索を提案した．脅威度を利用した探索アルゴリズムには λ 探索が，証明数・反証数を利用した優れた探索アルゴリズムとしては df-pn 探索が存在するが，df-pn λ 探索は，脅威度と証明数・反証数の双方を利用することで，さらに効率的に探索を行うことができる．

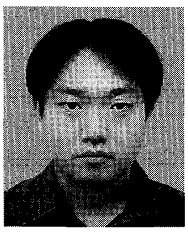
性質の異なる複数のゲーム（将棋，囲碁，シンペイ）を対象に df-pn λ 探索を実装して，df-pn 探索と性能を比較する実験を行った．その結果，現実的なゲームプログラミングで重要となる条件での df-pn λ 探索の有効性を確認できた．具体的には将棋では必至問題で，囲碁では低い λ 次数での探索で有効であった．また，シンペイの実験では λ 次数の分布を調査し，λ 次数が大きい手は少ないことが分かった．そのため，根節点の λ 次数の閾値が大きくなるにつれて df-pn λ の効果が低減し，df-pn の方が勝るという知見を得た．

参 考 文 献

1) Allis, L.V.: Searching for Solutions in Games and Artificial Intelligence, Ph.D. thesis, University of Limburg, Maastricht, the Netherlands (Sep. 1994).
2) Allis, L.V., van der Meulen, M. and van den Herik, H.J.: Proof-number search, *Artif. Intell.*, Vol.66, No.1, pp.91–124 (1994).
3) Cazenave, T.: A generalized threats search algorithm, *Computers and Games*, Vol.2883 of Lecture Notes in Computer Science, pp.75–87, Springer (2002).
4) Kawano, Y.: Using similar positions to search game trees, *Games of No Chance*, pp.193–202, Cambridge University Press (1996).
5) Kishimoto, A.: Correct and Efficient Search Algorithms in the Presence of Repetitions, Ph.D. thesis, University of Alberta (2005).
6) Kishimoto, A. and Müller, M.: Df-pn in Go: An application to the one-eye problem, *Advances in Computer Games 10*, pp.125–141 (2003).
7) Kishimoto, A. and Müller, M.: Search versus knowledge for solving life and death problems in Go, *20th National Conference on Artificial Intelligence* (*AAAI-05*), pp.1374–1379, AAAI Press (2005).
8) Nagai, A.: Df-pn Algorithm for Searching

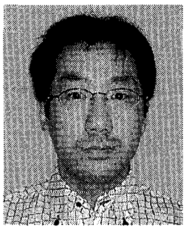
AND/OR Trees and Its Applications, Ph.D. thesis, Department of Information Science, University of Tokyo, Japan (2002).
9) Schaeffer, J., Björnsson, Y., Burch, N., Kishimoto, A., Müller, M., Lake, R., Lu, P. and Sutphen, S.: Solving checkers, *19th International Joint Conference on Artificial Intelligence* (*IJCAI-05*), pp.292–297 (2005).
10) Soeda, S.: Game Tree Search Algorithms based on Threats, Ph.D. thesis, The University of Tokyo (2006).
11) Thomsen, T.: Lambda-search in game trees – with application to Go, *Computers and Games* (*CG 2000*), Vol.2063 of Lecture Notes in Computer Science, pp.19–38, Springer (2002).
12) Yoshizoe, K., Kishimoto, A. and Müller, M.: Lambda depth-first proof number search and its application to Go, *20th International Joint Conference on Artificial Intelligence* (*IJCAI-07*), pp.2404–2409 (2007).
13) 金子タカシ：詰みより必至，毎日コミュニケーションズ (2003).
14) 久米　宏：将棋倶楽部 24 万局集，ナイタイ出版 (2002).
15) 脊尾昌宏：詰将棋を解くアルゴリズムにおける優越関係の効率的な利用について，第 5 回ゲーム・プログラミング ワークショップ，pp.129–135 (1999).
16) 金子知適，田中哲朗，山口和紀，川合　慧：新規節点で固定深さの探索を併用する df-pn アルゴリズム，第 10 回ゲーム・プログラミング ワークショップ，pp.1–8 (2005).
17) 田中哲朗：ボードゲーム「シンペイ」の完全解析，情報処理学会ゲーム情報学研究会資料，第 15-9 巻，pp.65–72 (2006).
18) 日本棋院（編）：攻め合いの達人，日本棋院 (2002). ISBN: 4818204722.
19) 副田俊介，美添一樹，田中哲朗：証明数を用いた λ 探索の効率的な実装，第 11 回ゲームプログラミング ワークショップ，pp.48–55 (2006).

(平成 19 年 1 月 29 日受付)
(平成 19 年 9 月 3 日採録)



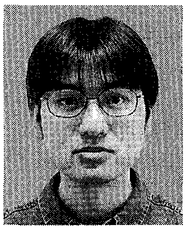
副田　俊介（正会員）

1977 年生．公立はこだて未来大学 CREST ポストドクター研究員．2006 年東京大学総合文化研究科博士課程単位取得退学．博士（学術）．人工知能，ゲーム情報学に興味を持つ．



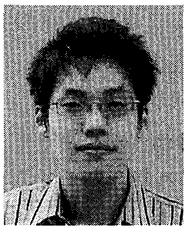
美添　一樹（正会員）

1974 年生．2003 年東京大学大学院理学系研究科博士課程単位取得退学．（株）富士通研究所，東京大学大学院研究生等を経て，2006 年中央大学研究開発機構助手．人工知能に興味を持つ．



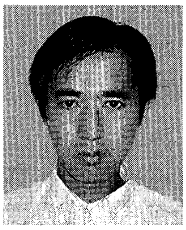
岸本　章宏（正会員）

公立はこだて未来大学システム情報科学部情報アーキテクチャ学科助教．2005 年アルバータ大学コンピューティング・サイエンス学科博士課程修了．博士（理学）．人工知能に興味を持つ．東大将棋の開発に従事．



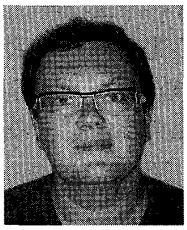
金子　知適（正会員）

1997 年東京大学教養学部卒業．2002 年東京大学大学院総合文化研究科博士課程修了．博士（学術）．2002 年東京大学院総合文化研究科助手．思考ゲーム，知識処理に興味を持つ．



田中　哲朗（正会員）

1965 年生．1992 年東京大学大学院博士課程修了．博士（工学）．東京大学工学部助手，東京大学教育用計算機センター助教授を経て，現在は東京大学情報基盤センター准教授．



マーティン　ミュラー

アルバータ大学コンピューティング・サイエンス学科准教授．組合せゲーム理論，コンピュータ囲碁の研究に興味を持つ．