相手モデルを持つゲーム木探索法についての考察*

2N－6

徳田浩, 飯田弘之, 細江正樹, 久保田聡, 小谷善行
(東京農工大学　工学部　電子情報工学科)

1　はじめに

ゲーム木探索の手法として、我々の提案した相手モデルを考慮する探索戦略 [1] [2] は、ミニマックス戦略を越える結果を導き出すことが期待できる。ただ、この探索法はミニマックス法に比べると、単純に2倍の回数の静的評価が必要になる。そのため、この探索法のためのいくつかの枝刈りの手法が提案されている。本稿では、この探索法の実際のゲーム木探索における効率と評価について述べる。

2　OM-search

文献 [1] で相手モデルを考慮した探索法をOM-searchとして定義した。

OM-search は、各プレイヤーのモデルは独立している、自分は相手モデルに関する正確な知識を持っている、相手はミニマックス法にしたがってプレイする、という条件の下に行われる。OM-search がどのように子局面の値を繰り上げるかを以下の式に示す。

$$V_{om}(P)=\begin{cases}\max\limits_i V_{om}(P_i) & P\text{が max 局面}\\ V_{om}(P_j)\text{ for } j \text{ satisfying} & \\ \quad g(P_j)=\min\limits_i g(P_i) & P\text{が min 局面}\\ EV_f(P) & P\text{が葉局面}\end{cases}$$

$$g(P)=\begin{cases}\max\limits_i g(P_i) & P\text{が max 局面}\\ \min\limits_i g(P_i) & P\text{が min 局面}\\ EV_g(P) & P\text{が葉局面}\end{cases}$$

$V_{om}(P)$ はOM-search による局面 P の評価値を表す。$g(P)$ は相手モデルによるミニマックス法による評価値を表す。$EV_f(P)$ は自分の評価関数による静的評価値、$EV_g(P)$ は自分が持っている相手の評価関数による静的評価値である。

3　ゲーム木モデル

実験のために用いるゲーム木モデルは均質木であり、深さは d、各ノードでの分岐数は w であるとする。

自分と相手のそれぞれの観点からの、葉局面に対する評価値は以下のように計算される。いまある葉局面を P^d として、ルート局面 P^0 から P^d に至る経路の局面を $P^0, P^1, P^2, \cdots, P^d$ とする。

1. ルート局面 P^0 に2つの乱数値 $R(P^0)$、$r(P^0)$ を自分と相手のために発生させる。
2. $R(P)$、$r(P)$ を乱数の種としてそれぞれ乱数を発生させ、ルート局面 P^0 の各子局面に2つの乱数を順に割り当てる。
3. 1と2の手続きを再帰的に葉局面まで行う。
4. 自分の観点からのスコア $EV_f(P^d)$ を

$$EV_f(P^d)=\sum_{k=0}^{d} R(P^k)$$

によって与える。同様に、相手の観点からのスコア $EV_g(P^d)$ を

$$EV_g(P^d)=\gamma\times EV_f(P^d)+(1-\gamma)\times\sum_{k=0}^{d} r(P^k) \quad (1)$$

によって与える。ただし、γ は $0\le\gamma\le 1$ を満たす実数である。式 (1) において、$\gamma=1$ であることは、相手が自分とまったく同じように評価することを意図している。このγ値の変化によって、様々な相手のモデルが構築されると考えられる。

このようなゲーム木モデルを作ることによって、実際のゲームのように子局面が親局面の影響を受ける様子を表すことができる。

4　ゲーム木モデルでの実験

3節で述べたゲーム木モデルでOM-search の、ミスの測定、効率化の評価、得られた値の評価を行った。

OM-search が考える最善の値を$\alpha\beta$法が選択しなかった場合をミスという。ミスの割合は、γが0に近づくにしたがって大きくなった。また、木の深さと各ノードからの分岐数のそれぞれが大きくなるにしたがってミスの割合が大きくなった。結果の一部を表に示し、表中の値は木の内部の相手ノードに対するミスの回数を表す。

*Thoughts on Game Tree Search of Opponent-Model
Hiroshi TOKUDA, Hiroyuki IIDA, Masaki HOSOE,
Satoshi KUBOTA, Yoshiyuki KOTANI
Tokyo University of Agriculture and Technology

効率化の評価は、β枝刈りとルート値枝刈りという2種類の手法で静的評価の回数を測定した。β枝刈りは$\alpha\beta$法のβカットにあたる枝刈りの手法である。、ルート値枝刈りは、一回ミニマックス法で探索したときの評価値をルート値として保存し、再びOM-searchを行うときに枝刈りに利用するという手法である。その結果、両方の手法で木の深さと各ノードからの分岐数が増加するにしたがって、静的評価が不要となるノードの割合が増加した。また、ルート値枝刈りの場合はγが小さくなるほど静的評価の回数は減少した。結果の一部としてβ枝刈りによる効率化の表を示す。表中の値は、効率化なしのOM-searchでの静的評価回数に対するβ枝刈りOM-searchでの静的評価回数の割合である。

探索によって得られた値は、minimini法とmaxmax法という比較の基準を設定し、ミニマックス法、OM-search、ルート値枝刈りOM-searchとを比較した。結果として、深さ、分岐数が大きくなり、γが0に近くなるほど、得られる評価値は大きくなった。

表1: ミスの発生率(%): $\gamma = 0.5$

d\w	2	3	4	5	6
3	16.0	34.3	35.8	33.4	36.2
4	20.0	34.3	33.6	33.9	35.8
5	15.8	30.0	33.0	30.0	36.2
6	20.2	33.5	35.4	35.8	37.7

表2: β枝刈りによる効率化(%): $\gamma = 0.5$

d\w	2	3	4	5	6
3	82.6	69.8	63.0	57.7	53.5
4	67.8	53.1	44.4	38.8	34.8
5	66.6	46.4	36.1	29.8	25.5
6	56.8	37.5	26.9	20.7	17.3

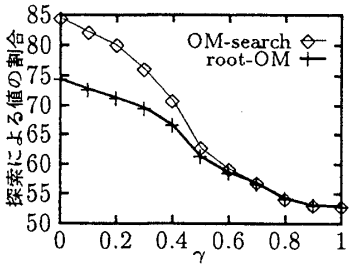

図1: 各探索法の値の割合

5 実際のゲームでの実験

5.1 TicTacToeでの実験

TicTacToeはゲームとしてかなり小さな部類に入り、両方のプレイヤーが最善を尽くすと、必ず引き分けになるという性質を持っている。

OM-searchと$\alpha\beta$法とを対戦させ、$\alpha\beta$の評価関数をOM-searchが正確に予想していた場合、OM-searchが勝ちになる手順が現れた。

5.2 将棋での実験

将棋において、OM-searchと$\alpha\beta$法を対戦させた。OM-searchが相手の評価関数を正確に予想できるとする。図2のような局面で、先手は$\alpha\beta$法に従って探索した場合、▲4八銀と指すが、OM-searchで探索した場合▲7七桂と指し△同馬▲同角となる手順を読んでいる。普通、△7七馬のように損な手を相手が指さないと考えて候補手から削除してしまうが、相手の評価関数が△7七馬を高く評価することを知っているためにこのような手を指すことができる。

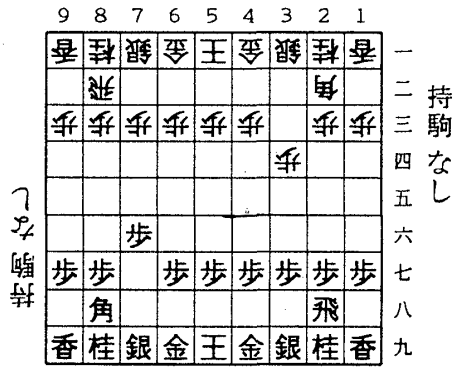

図2: 局面例

このように、相手の評価関数を正確に予想できる場合には相手のミスを引き出すような手を指すことができる。探索深さを同じにし、$\alpha\beta$法を用いたプレイヤーの評価関数をOM-searchが正確に予想できるという条件で対戦を行った結果OM-searchを用いたプレイヤーが19勝1敗で勝つという結果が出た。

6 まとめ

このように、OM-searchを実戦的なゲーム木とTicTacToeと将棋で実験した。それぞれの探索結果は相手モデルを正確に予想できたときには効果を発揮し、相手モデルにミスが多いほど効率化が図れることがわかった。

参考文献

[1] Iida, H., Uiterwijk, J.W.H.M. and Herik H.J.v.d., Opponent-Model Search, *Technical Reports in Computer Science*,Dept.of Computer Science, Univercity of Limburg,Maastricht,1993.

[2] Iida, H., Uiterwijk, J.W.H.M. and Herik H.J.v.d., Thoughts On Applying The Opponent Model Search, *Advances in Computer Chess 7*, 1993.