4S-06

認知的満足化アルゴリズムの木探索への応用

齋藤 建志 †　　高橋 達二 ‡
† 東京電機大学大学院　　‡ 東京電機大学

1 はじめに

知識を自ら選択、獲得しながらうまく振る舞えるような人工知能の需要は大きい。このような枠組みは強化学習と呼ばれ、モデル化によって状態遷移を木で表現できることがある。そのような場合、状態遷移を繰り返すごとに木は深く大きくなっていくため探索空間が膨大になりやすい。そのため、人工知能による木探索では UCB1 という価値関数を用いたモンテカルロ木探索 UCT が多く利用されてきた。しかし、膨大な試行回数が必要であったり、初期の振る舞いによる指標誤差の問題があった。　そこで、本研究では人間の意思決定法における特性の満足化を価値関数に取り入れた木探索に有用なアルゴリズムを提案し、シミュレーションによる評価を行う。

2 モンテカルロ木探索

モンテカルロ木探索 (Monte Carlo Tree Search:MCTS) とは乱数を用いたシミュレーションを繰り返すことで探索木を徐々に深くしていき、近似的に解を求める手法である。具体的な手順を Algorithm 1 に示した。Algorithm 1 の TREEPOLICY(v_0) は root ノードから最も有望か未探索なノードを返し、DEFAULTPOLICY(v_t) はノード v_t からランダムに手を打って得られる報酬を返す [2]。BACKUP(v_t, Δ) はノード v_t から root ノードまで得た報酬と試行回数を伝播させる。MCTS の中でも評価値として UCB1 を用いる UCT アルゴリズムは理論的に最善手である minimax アルゴリズムの解に収束する有用なアルゴリズムである [3]。

Algorithm 1 General MCTS

for $1, 2, \ldots,$ Maxstep **do**
　$v_t \leftarrow$ TREEPOLICY(v_0)
　$\Delta \leftarrow$ DFAULTPOLICY(v_t)
　$BACKUP(v_t, \Delta)$
end for

Risk-Sensitive Satisficing Algorithm for Tree Search
†Kenshi Saito　‡Tatsuji Takahashi
†Graduate School of Tokyo Denki University
‡Tokyo Denki University

3 満足化価値関数 RS

強化学習においては探索空間が大きすぎて探索が困難なことが多い。そこで、高橋らは人間の意思決定の特性である満足化を価値関数に取り入れて効率的な探索を行う RS アルゴリズムを提案した [1]。多腕バンディット問題での RS アルゴリズムで用いられる満足化価値関数は以下のように定義される。

$$RS_i = n_i(E_i - R) \tag{1}$$

ここで、E_i は獲得報酬平均で n_i は行動 a_i を試行した回数、R は基準値である。そしてエージェントは常に最大の RS_i を持つ行動 a_i を選択する。RS アルゴリズムでは、満足化価値関数の値に従って greedy に行動選択を行っていく。実験によって、多腕バンディット問題において基準値 R の設定によっては UCB1 よりも素早く学習できることが示されている [1]。

4 満足化木探索アルゴリズムの提案

RS アルゴリズムは基準値の設定によって UCB1 よりも多腕バンディット問題における学習が早い [1]。MCTS においても、ある状態での行動選択はランダムな試行によって得られた報酬に基づいてより有望な行動を選択していくため一種の多腕バンディット問題と捉えることができる。このことから、満足化価値関数 RS を MCTS に活用することでより枝切りをうまく行うアルゴリズムになると考えられる。(1) 式の満足化価値関数に従い、MCTS に用いる価値関数として (2) 式を定義した。そして UCT と同様に行動価値関数として RS を用いる木探索アルゴリズムを RST と呼ぶ。

$$RS(v) = n(v)(E(v) - R) \tag{2}$$

ここで、v は game tree 中のノードを表し、n(v) はノード v を通ってプレイアウトした回数、E(v) はプレイアウトで得られた報酬から計算する報酬確率である。多腕バンディット問題で RS は基準値が最適であれば UCB1 よりも優れている [1] ことから、RST における基準値 R をどのように設定するべきかという問題が生じる。そこで、あるノード v における行動選択を多腕バンディット問題と捉えた際に最適な基準値 R_{opt} を以下のように

定義する。

$$R_{opt}(v) = \frac{p_1 + p_2}{2} \tag{3}$$

ここで、p_1 は v の子ノード中で遷移していける終端状態の最大報酬平均で、p_2 は p_1 の次に大きい報酬平均である。

5 実験

満足化木探索アルゴリズム (RST) の性能を評価するために P-Game Tree を用いた実験を行う。実験で用いる P-Game Tree は深さ 16、葉以外の全てのノードは 2 つの子ノードをもつように設定した。この P-Game Tree では最初にプレーヤ MAX が行動を選択し、その次に MIN が行動を選択することを想定した。そのため、root ノードからその子ノードへの遷移は MAX が選択し、そのまた子ノードは MIN が選択することとなる。また各ノードへの遷移にはその行動を選択することでどれだけ MAX が有利になるかを表す評価値を設定する。MAX が選択する場合はその行動の評価値として一様分布から独立に範囲 [0, 127] の中から一つ割り当て、MIN が選択する場合は範囲 [−127, 0] として同様に割り当てる。ゲームの勝敗は葉ノードで決定され、root ノードからその葉ノードへ遷移した際の評価値の和が正であれば MAX の勝ち、負であれば MIN の勝ちとした。実験では、100 種類の P-Game Tree を生成しそれぞれについて満足化木探索アルゴリズムによる探索を行い、それらの結果を平均して満足化木探索アルゴリズムの評価を行った。また、Algorithm 1 の 1 ループを 1 playout として 10,000 playout を行った。最適切基準を用い場合は Ropt、R=0.9 を R0.9 のように表記した。

6 結果と結論

各 playout において Minimax アルゴリズムで得られる最適な選択をしたか否かを記録し、100 種類の木について平均して求めた各 playout での正解率を図 1 に示した。図 1 を見ると R_{opt} が最も高い平均正解率に達していることが分かる。さらに最適切基準を設定した RST について調べるため、5 つの木をランダムな評価値で生成してそれぞれについて Ropt を設定した RST による探索を 100 回行った際の MAX の第一行動における正解率を図 2 に示した。図 2 を見ると $(p_1, p_2) = (0.50, 0.31)$ の P-Game Tree で最適切基準を設定した RST が正解率約 0.8 で止まってしまっていることがわかる。したがって、RST における基準値の決定法として最適切基準を用いることでより正確に早く最適な行動を見つけ出す傾向があるが、最適でない行動を選択して満足してしまうことがあるとわかった。現状では基準値の計算に事前情報を用いているので、今後の課題として事前情報を用いずに各状態において有用な基準値決定法を見つけることが必要である。

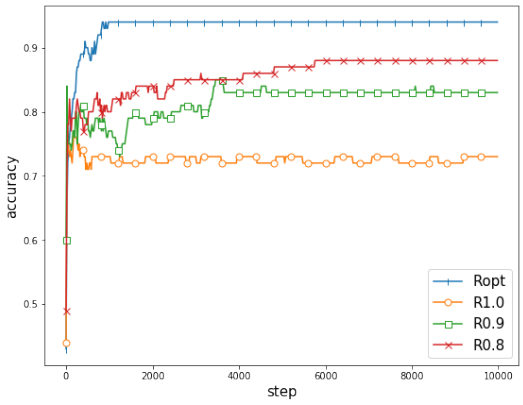

図 1: 満足化木探索実験正解率

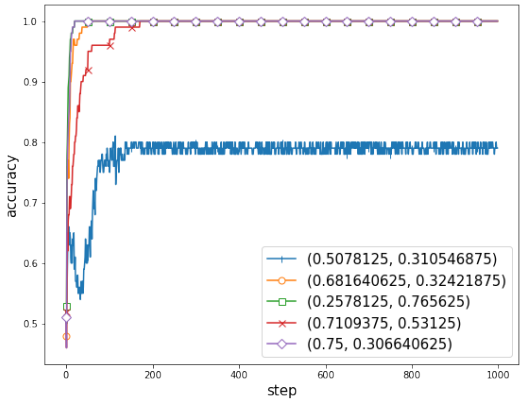

図 2: 最適切基準設定時正解率

参考文献

[1] 高橋 達二, 甲野 佑, 浦上 大輔, 認知的満足化, 人工知能学会論文誌, 2016, 31 巻, 6 号, p.AI30-M10

[2] C. B. Browne et al., "A Survey of Monte Carlo Tree Search Methods," in IEEE Transactions on Computational Intelligence and AI in Games, vol. 4, no. 1, pp. 1-43, March 2012. doi: 10.1109/TCIAIG.2012.2186810

[3] Kocsis, L. and Szepesvari, C.: Bandit based Monte-Carlo Planning, Machine Learning: ECML 2006 In Proceedings of the 17th European conference on Machine Learning, 4212, 282293 (2006).