囲碁プログラムにおける UCB アルゴリズムの有効性に関する一検討

情報システムコース 山中 涼　指導教員：成瀬 正

1 はじめに

近年、コンピュータ囲碁の分野ではモンテカルロ碁が注目を集めている。モンテカルロシミュレーションの制御に UCB(Upper Confidence Bound)アルゴリズムが用いられる。

決められた時間内により良い手を選ぶには、試行回数の削減を行い、効率化をはかることが重要である。

UCB アルゴリズムにおいて、プレイアウト回数を削減する手法が提案されている[1]。その報告では、削減率が 5%程度と低い。

本研究では、この手法について数値実験により追試し、削減効果が上がらない理由について検討する。

2 UCB アルゴリズム

UCB アルゴリズムでは、ある局面から、次手を決定する際に以下の値が最も大きなものについて試行を行う。

$$\bar{x}_j + \sqrt{\frac{2\ln n}{n_j}} \tag{1}$$

ここで、$\bar{x}_j$:候補手 j の勝率の平均値
n:現時点での総プレイアウト回数
n_j:現時点における候補手 j のプレイアウト回数
このとき、以下の定理が成り立つ。

$$E\big(T_i(n)\big) \le \frac{8\ln n}{\Delta_i^2} + 1 + \frac{\pi^2}{3} \tag{2}$$

ここで、$E(T_i(n))$:n 回のうち候補手i がプレイアウトされる回数の期待値
$\Delta_i = \mu^* - \mu_i$
μ^*：最良候補手の勝率の期待値
μ_i：候補手 i の勝率の期待値

3 試行回数削減アルゴリズム

UCB アルゴリズムにおいて$\mu^* = \mu_j, \mu^* \ne \mu_i$とし、さらに$\bar{x}_i < \bar{x}_j$とする。このとき$i$に対する試行を重ねて$\bar{x}_j \le \bar{x}_i$となるために必要な回数を考えると、$i$は少なくとも以下の N 回は勝ちの試行を得なければならない。

$$N = \frac{(\bar{x}_j - \bar{x}_i)}{\left(\frac{1-\bar{x}_i}{n_i+1}\right)} = \frac{(\bar{x}_j - \bar{x}_i)}{(1-\bar{x}_i)}(n_i+1) \tag{3}$$

次にiが 1 回試行される間に j が試行される回数を考えると、以下の回数試行されることとなる。

$$\max\left(1, \frac{n_j}{F(\bar{x}_i)}\right) \tag{4}$$

ただし$F(x) = \frac{8\ln n}{(\bar{x}_j - x)^2} + 1 + \frac{\pi^2}{3}$とする。

$F(x)$は式(2)の右辺であるが、μ の真値はわからないので、μ の代わりに$\bar{x}$を用いている。

以上より、$\bar{x}_j \le \bar{x}_i$となるまでに、j は少なくとも以下の z 回試行されることとなる。

$$z = \sum_{k=1}^{N} \max\left(1, \frac{n_j}{F\left(\bar{x}_i + \frac{1-\bar{x}_i}{n_i+1}k\right)}\right) \tag{5}$$

最も試行回数の多い候補手が一定回数となった時点で着手が決定される回数を s とすると、

$$z > s - n_j \tag{6}$$

であれば、着手が決定される前に$\bar{x}_j \le \bar{x}_i$となることはないから、上記の条件を満たすi に関する試行はやらなくてよいこととなる。

4 実験

本研究では、勝率の分布を正規分布に従うと仮定し、勝ちの試行を得たら 1、負けの試行を得たら 0 として$\bar{x}$を求め、式(6)からプレイアウト回数がどの程度削減できるか実験した。

実験を行うにあたり、必要なパラメータを次のように設定した。

・候補手は 2 とし、s=5000 とする。
・UCB 削減法は、$z > s - n_i$となった時点で終了する。
・Δ=0.05,0.01,0.25,0.5 についてそれぞれ 20 回づつ実験した結果の平均値をとる。

結果を下表に示す。

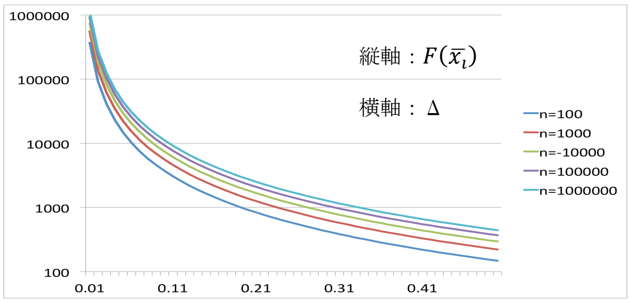
正規分布

Δ	UCB 法	UCB 削減法	削減率
0.05	6427.05	6271.3	2.42%
0.1	5709.35	5565.65	2.51%
0.25	5187.15	5023.75	3.15%
0.5	5059.65	4840.1	4.34%

5 考察

表から、Δの値が大きくなるにつれ削減率も上昇するが、大きな削減にはならなかった。その理由として、削減アルゴリズムは UCB アルゴリズムに付随する上限値に大きく依存しているからと考えられる。

以下のグラフは上限値を表す$F(\bar{x}_i)$である。



上限値は、Δに反比例するため、Δの小さい範囲では$n_j < F(\bar{x}_i)$であるので、$max\left(1, \frac{n_j}{F(\bar{x}_i)}\right) = 1$となり、削減法の効果が薄い。

すなわち、削減率が上がらない理由は$n_j < F(\bar{x}_i)$である範囲が広いためであり、また上限値が過大評価されているためである。

したがって、式(2)の上限値の精密化を行わなければならない。

6 おわりに

本研究では、UCB アルゴリズムの削減法を適用したシミュレーションにより、削減アルゴリズムがどのような範囲で有効であるかについて示した。

今後は、UCB アルゴリズムの上限値の引き下げを提案し、更なる試行回数の削減を進める予定である。

参考文献

[1]但馬 康宏，小谷 善行　「UCT アルゴリズムにおける確率的試行回数削減方法」情報処理学会研究報告. GI, [ゲーム情報学]2008(59), pp. 23-30, 2008-06-20