

ゲームプログラミングのアルゴリズム

98146 山本 諭 (非線型物理学講座)

ゲームプログラミングの基本的なアルゴリズムであるミニマックス法とアルファベータ法の比較を行う。

1 ゲームプログラミングのアルゴリズム

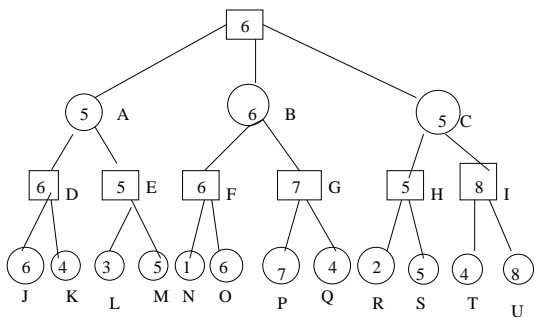


図 1: ゲーム木

1. ミニマックス法

まず現在の局面から数手先 (図 1 では 3 手) の局面をすべて調べ評価値として表します。そして自分は自分にとって最良の手を選ぶと仮定し自分の手番の時は評価値を最大にするように選択し、相手は自分にとってもっとも不利になるように手を選ぶと仮定し相手の手番の時は評価値を最小にするように選択します。これをミニマックス法と呼びます。

2. アルファベータ法

ミニマックス法では 1 局面で n 手選べると仮定すれば m 手進めた時の局面の数は n の m 乗となり深く探索すれば、調べないといけない局面の数は爆発的に増えてしまいます。

そこで考えられたのがアルファベータ法です。アルファベータ法では図 1 の接点 P を探索した後、接点 Q は探索する必要がなくなります。なぜなら接点 P の評価値が 7 なので接点 G の評価値は 7 以上になります。接点 F の評価値は 6 で 7 より

小さいので接点 B は 6 に決まります。この様にある接点の評価値がある値以上になるため探索を打ち切る事をベータカットと言います。

次に接点 H を探索した時の事を考えます。この時接点 C の評価値は 5 以下になります。すると接点 B の評価値が 6 なので、接点 C の局面になるような手は採用されない事が決まり、接点 I, T, U は探索する必要がなくなります。この様にある接点の評価値がある値以下になるため探索を打ち切る事をアルファカットと言います。

アルファカットおよびベータカットによって無駄な探索をしないようにする方法をアルファベータ法と呼びます。

2 実験

オセロゲームでコンピューターとコンピューター (共に深さ 1 のミニマックス法) を対戦させた時の手数推移を調べた。

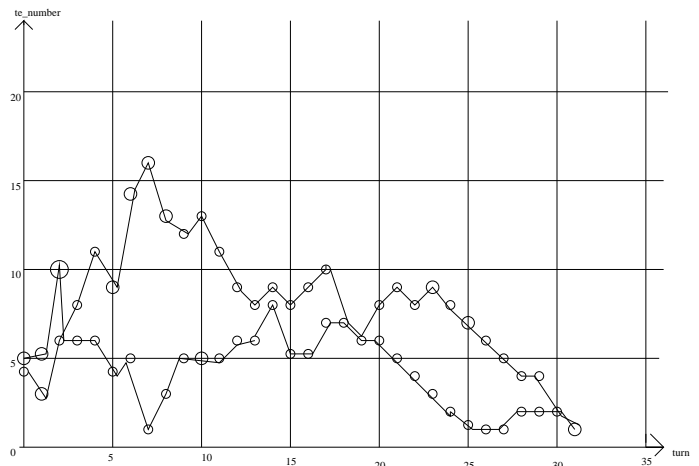


図 2: 横軸は局面で縦軸は手数

中盤に手数が増えた。中盤では平均大体 8 手くらいになった。これで 10 手先まで読むには、8 の 10 乗くらいの大きな数になる。

今後はアルファベータ法ではどのくらい節約されるのかを調べ発表する。