

## 1 ゲーム木の探索

代表的なゲーム木の探索アルゴリズムには、ミニマックス法とアルファベータ法がある。アルファベータ法は、ミニマックス法に分枝限定法の考え方を取り入れ、無駄な探索を極力避ける方法である。

## 2 ミニマックス法のアルゴリズム

```
minimax(節点  $n$ , 深さ  $d$ )
{
  if( $d=0$  ||  $n$  が終端節点) return( $n$  の評価値);
   $n$  の子節点を  $n_i$  とする
  if( $n$  が Max 節点) /* (先手番) */
    return  $\max_i\{\text{minimax}(n_i, d-1)\}$ ;
  else /*  $n$  が Min 節点 (後手番) */
    return  $\min_i\{\text{minimax}(n_i, d-1)\}$ ;
}
```

## 3 アルファベータ法のアルゴリズム

```
(節点  $n$ , 深さ  $d$ ,  $\alpha$ ,  $\beta$ )
{
  if( $d=0$  ||  $n$  が終端節点) return( $n$  の評価値);
   $\alpha_n = -$  ;  $\beta_n = +$  ;
  if( $n$  が Max 節点) {
    for(全ての  $n$  の子節点  $n_i$ ) {
       $v =$  ( $n_i, d-1, \max\{\alpha_n, \alpha\}, \beta$ );
       $\alpha_n = \max\{\alpha_n, v\}$ ;
      if( $\alpha_n \geq \beta$ ) return  $\alpha_n$ ;
    }
    return  $\alpha_n$ ;
  }
  else { //  $n$  が Min 節点
    for(全ての  $n$  の子節点  $n_i$ ) {
       $v =$  ( $n_i, d-1, \alpha, \min\{\beta_n, \beta\}$ );
       $\beta_n = \min\{\beta_n, v\}$ ;
      if( $\beta_n \leq \alpha$ ) return  $\beta_n$ ;
    }
    return  $\beta_n$ ;
  }
}
```