

やねうら王チーム WCSC33 終了後の PR 文書

・独自に工夫した点について

1. leaf node での df-pn の効率化について (b y やねうらお)

選手権前には PR 文書にも書いたように leaf node での df-pn の効率化をしようと思っていたのだが、簡単な予備実験によるとどうも効果が薄いようなので、わりと実装が大変だということもあり、今回は取り組むのを断念した。引き続き、やっていきたい。

2. 定跡作成の仕組み (当該項目における文責：たややん)

今回のやねうら王においては、以下のような定跡作成の仕組みを採用しました (2022 年末に行われた第 3 回電竜戦本戦における定跡作成手法をブラッシュアップしました)。

- (1) 指定局面から連続対局させ、指し手を全て定跡として登録する。
- (2) 任意の局面が定跡に登録されていた場合、ミニマックス法で定跡データ内を探索し、勝ちの枝があるか、負けの枝しかないか判別する。
- (3) 勝ちの枝があれば、その手を指し、負けの枝しかない、又は定跡に登録されていない局面であれば、探索エンジンで思考させる (探索エンジンにどれだけ思考させるかは、対局開始時にランダム (水匠 100 万ノードから 1 億ノードの範囲内) に決定する)。

このような仕組みを採れば、連勝を続ける限り、勝ちの枝は採用され続け、ノード数の異なる探索エンジンによって多数回その指し手が本当に勝ちか否かが挑戦され続けることになるため、定跡の精度は上がり続けます。

このような定跡作成ルーチンによって、作成された定跡は、2000 局面に満たない局面数しか登録されていないにもかかわらず、角換わり 37 手目基本図

(1r5n1/3g1kg2/2n1ppsp1/p1pps1p1p/1p5P1/P1PPSPP1P/1PS1P1N2/2GK1G3/LN5RL w Bb 38) から後手番水匠に対して 2000 連勝以上できるような定跡となりました。

疑似コードは以下のとおりです。

```

book = dict() #定跡保存用。
              #保存形式例：
              {"sfen lnsqkgsnl/1r5b1/ppppppppp/9/9/PPPPPPPPP/1B5R1/LNSGKGSNL b - 1": {"7g7f", "2g2f", "resign"}}
board = Board() #将棋盤。手を進めたり(push)、戻せたり(pop)する。

def minmax(board, book):
    #定跡内の指し手を簡易的なminmax法で探索し、末端局面において勝ち(+1)か、負け(-1)かを判定し、指し手とともに返す関数。
    #末端が"resign"で終わるようにbook登録されていることが前提
    #現局面のsfen文字列を受け取る
    #局面が定跡になければscore = 0, move = Noneを返す
    sfen = board.sfen()
    if sfen not in book:
        return 0, None
    for move in book[sfen]:
        if move == "resign":
            score = -1
        else:
            board.push(move)
            score = -minmax(board, book)[0]
            board.pop()
            if score == 1: #勝つ枝が見つければbreakしてよい(簡易的なアルファカット)
                break
    return score, move #scoreとその時の指し手を返す

board = ... #初期設定(連続対局開始局面をどうするか、投了値をどうするか等)
while True:
    #連続対局
    #定跡内ミニマックス探索
    score, move = minmax(board, book)
    if score == 1:
        #定跡内の指し手に勝つ枝があればそれを指す
        bestmove = move
    else:
        #勝つ枝がなければ探索して指す
        bestmove = ...
        book = ... #局面と指し手("resign"も含む)を定跡に登録
    if board == ...:
        #終局時処理。千日手の場合、後手勝ちとして、先手resignとしましたが、その他の実装もあると思います。
        ...
    else:
        board.push(bestmove) #終局していない場合、1手進める

```

・開発動機

近年、評価関数モデルの機械学習が札束による殴り合いになっているので、そこから逃走すべく定跡を整備することにした。

・開発過程

上記2. の疑似アルゴリズムで示されるコードで自動生成した。

・実験結果

思考エンジンに水匠を用い、定跡不使用のものと上記手法で自動生成した定跡を使用したものとを対局させると定跡を使用した側が **100%勝つ** ようにできた。また選手権では、定跡負けはほとんどしなかった。

・追試可能か

可能。

以上