部　　門	競　技　部　門	No.1 登録番号	30032

No.2

1) 予定開発期間：5ヶ月
2) 予定開発人数：3人

	4月	5月	6月	7月	8月	9月	10月
問題分析	←	→					
設計		←	→	→			
実装			←	→	→		
試用・トレーニング						←	→

No.3

実現方法

1) ボードを最終盤面にするアルゴリズム

各行を順番に最も上の行で揃え、それを23番の抜き型を用いて最も下の行に退避させるという操作を繰り返すことを考える。この操作を行数と同じ回数だけ繰り返すことで、ボードを最終盤面にすることができる。

各行を揃える際は順番に左から揃えていくことを考える。順番に揃える際は、現在の盤面から2手を全探索し、現在揃っている場所を崩さずに揃っている部分の長さを最大にできるような操作を適用することを考える。

盤面が大きく2手の全探索に長時間かかってしまう場合は、定型抜き型の特定の場合のみを考えることで高速化する。特にタイプⅡの定型抜き型を用いることで特定の連続部分列を最も上の行に移動させることができるので、この操作を中心に探索していく。

2) 不一致ピース・手数を少なくする工夫

上で説明したアルゴリズムに対してビームサーチを適用する。2手未満で揃っている部分が長くなるような操作を全て試し、最も長く伸ばせた盤面以外にも複数個の盤面を持つようにする。

しかし、ビームサーチのみでは制限時間を十分に使った探索となるようにパラメータを調整するのが難しい。そこで、ビームサーチをchokudaiサーチに変更したアルゴリズムも用意する。このchokudaiサーチを用いることで制限時間を十分に用いた探索を行うことが可能で、制限時間が事前に分からない状況下で柔軟に対応させることが可能である。

chokudaiサーチはビームサーチと比べてメモリを非常に多く利用するという欠点もあるため、制限時間や盤面の大きさに対応して用いるアルゴリズムを本番で自由に変更できるようにする。

参考文献：ゲームで学ぶ探索アルゴリズム実践入門（青木栄太）、技術評論社、2023年2月

3) その他（独創的なところ）

ウェブブラウザ上で動作するビジュアライザを作成した。テストケースの生成と、操作を可視化したGIFファイルを生成することができる。これによりチーム内で情報を共有しやすくなる。図はこのビジュアライザの動作画面である。

また、競技中に使用するビジュアライザをOpenSiv3Dを用いて作成する。このビジュアライザで盤面とソルバーの情報の表示を行う。さらに、メインプログラムとプロセス間通信をすることによって、初期盤面に応じて最適なアルゴリズムを選択することができるようにする。サーバーとの通信もこのビジュアライザが担当する。

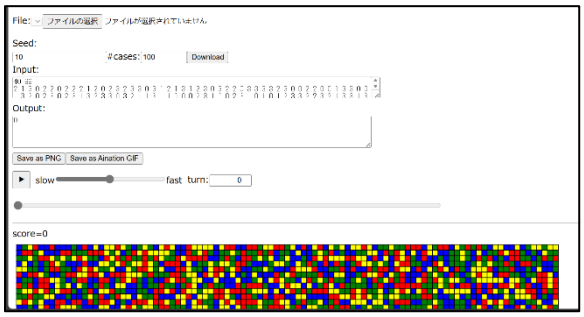

図1 ビジュアライザの外観

No.4

開発環境
- C++ (OpenSiv3D 0.6.14, AtCoder Library)
- Python3 (PyTorch)
- Rust (visualizer-template-public)