

|     |         |            |       |
|-----|---------|------------|-------|
| 部 門 | 競 技 部 門 | No. 1 登録番号 | 30033 |
|-----|---------|------------|-------|

|           |                              |        |    |        |    |    |    |     |
|-----------|------------------------------|--------|----|--------|----|----|----|-----|
| No.2      | 1) 予定開発期間：半年間(競技内容公開直後～本選直前) |        |    |        |    |    |    |     |
|           | 2) 予定開発人数：6人程度               |        |    |        |    |    |    |     |
|           |                              | 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 | <p>実現方法</p> <p>1) 公開フィールドへの事前対策</p> <p>今年の競技ではフィールドの一部が事前に公開される事から、各公開フィールドに対してそれぞれ特有の対策を行うことにする。パラメータを様々な形で調整した複数のアルゴリズムを用いて数万回以上の対戦を行い、それらの対戦データを活用して序盤の定石や典型的な行動パターン、膠着状態に持ち込まれる状態などをそれぞれ事前に予測する。また、それぞれの盤面に対して最適なアルゴリズムとパラメータの組も事前に調べ、本選ではそれを使用する。</p> <p>さらに、現在の盤面状態からの勝率の評価を行うようなネットワークを深層学習によって実装し、それを評価基準に組み込んだアルゴリズムを実装する。</p>   |
|      | <p>2) 公開フィールドでの戦略アルゴリズム</p> <p>フィールド内容が事前に公開されていることから、(1)で記載したような情報を事前に予測することができると考えている。また、盤面が固定されているので盤面の取りうる状態数が去年と比べてかなり小さくなるため、強化学習などを用いたアプローチや盤面からの勝率評価も可能になり、それらを局所的に組み込んだアルゴリズムを採用する。</p> <p>序盤の動きに関しては事前に予測されているパターンを優先するような形でエージェントの行動を決定する。相手が予測と大幅に異なる行動をした場合は適宜別のアルゴリズムに切り替える事で対処をする。</p> <p>中盤以降は基本的に BeamSearch や Min-Max 探索のアルゴリズムを使用する。また、盤面に対するパラメータの最適化を事前に行う。アルゴリズムに使用する評価関数には、現在の得点やタイル情報だけでなく深層学習によって評価された現在の勝率の予測値や膠着状態に持ち込まれる確率なども用いる。</p> |
|      | <p>3) 非公開フィールドでの戦略アルゴリズム</p> <p>非公開フィールドでの対戦では BeamSearch を主としたアルゴリズムを使用する。非公開フィールドに関して盤面の状況を分析して、エージェントが集中しやすいと考えられる場所や、連続して得点が得られると考えられるパスを簡単な方法で計算し、それらを考慮して行動を決定させる。</p> <p>アルゴリズム内で使用するパラメータの最適値は盤面の点数配置やタイルの状態によって異なるため、対戦中の各ターンで使用するべきパラメータの値を再計算し、それぞれのターンの行動は再計算されたパラメータを用いて決定する。</p> <p>非公開フィールドでの対戦は公開フィールドでの対局後に行われるため、状況によっては失点数を一定以下にすればいい場合が発生する。そのような場合は、積極的に膠着状態を発生させるようなアルゴリズムを使用する。</p>  |
|      | <p>4) その他 (独創的なところ)</p> <p>UI の描画部分や盤面の更新処理、領域ポイントの計算といった時間のかかる処理やアルゴリズムの実装については高速化のために C++ を用いて実装をするが、機械学習による勝率の予測などでは Python の機械学習フレームワークを用いる事になる。冗長な処理を省いたまま C++ 側から Python で実装された関数などを呼び出すために、Boost.Python を用いて複数言語間の処理を行う。また、アルゴリズムには抽象化されたラッパークラスを用意し、複数アルゴリズムの切り替えや自己対戦でのお互いのアルゴリズムの変更を簡単に行えるように実装している。</p>  |

|      |  |
|------|--|
| No.4 | <p>開発環境</p> <p>OS: Arch Linux</p> <p>使用言語: C++17, Python</p> <p>使用するライブラリ, フレームワーク: Qt, Boost, Chainer 等</p> |
|------|--|