

フリーセルの着手を学習するディープラーニングのためのデータセットの作成

Construction of Data Set for Deep Learning of Moves in FreeCell

玉井 慎悟[†]
Shingo Tamai

今村 奨[†]
Sho Imamura

上野 一樹[†]
Kazuki Ueno

神保 秀司[†]
Shuji Jimbo

1. はじめに

質の良い着手を求めるニューラルネットと従来型の探索部分の組み合わせによる小さい探索木をもつフリーセルソルバーの開発を試みているが、ニューラルネットに短時間で十分な学習をさせるには、A*アルゴリズムや深さ優先探索などの従来型のアルゴリズムを使って十分な質と量をもつディープラーニングのためのデータセットを作成し、ニューラルネットに教師有り学習をさせる必要がある。本研究における質の良いデータセットの作成のための初期局面の選別と最短手順に近い解の求解のための戦略について述べる。

2. フリーセルの求解とデータセットの作成

2.1 フリーセルとは

フリーセルは一人用のトランプゲームであり、ジョーカーを除く 52 枚のカードをホームセルと呼ばれる場所にスートごとにエースからキングの順番で移動させることを目的としている。初期盤面は図 1 のようにカードを 8 つのタブローと呼ばれる山にランダムに分割し 7 枚のタブローを 4 つと 6 枚のタブローを 4 つ用意する。移動させることの可能なカードは各タブローの天（一番上、図上の一番下）のカードかフリーセルに置かれているカードである。フリーセルとは 4 つ用意されている空のセルで、フリーセルに置くことの可能なカードは各セルに 1 枚までである。タブローの上に置くことの可能なカードは各タブローの天のカードと異なる色の数字が一つ小さいカードのみである(例: ♠7 の上に♥6, ♦K の上に♣Q など)。以上のルールにのっとり初期盤面からすべてのカードをスートごとにエースからキングの順にホームセルに移動させれば勝利となる。

2.2 フリーセルソルバ

今回はランダムに初期盤面を作成し、その盤面をフリーセルソルバに与えて解を求め、それを元にデータセットを作成する。使用するソルバではヒューリスティック関数で探索空間を制限し深さ優先探索を行っている。関数に使用しているパラメータは“現在までの手数(T)”，“ホームセルに移動していないカードの数(N)”，“封鎖辺となっているカードの数

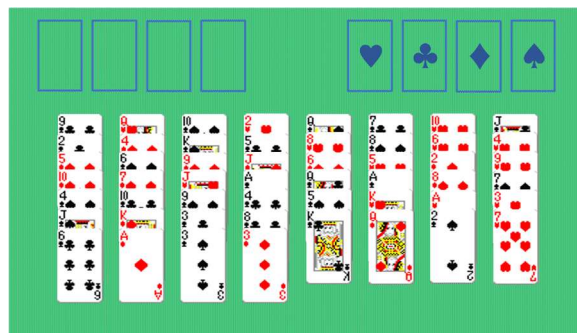


図 1: フリーセルの初期盤面の一例。左上にフリーセル, 右上にホームセルがありすべてのカードをホームセルに移動させると勝利

(B)”，“動かしていないカードの数(M)”の 4 つである。ヒューリスティック関数を次のように定義した。

$$1500 * T + \text{alph1} * N + \text{alph2} * B + \text{alph3} * M$$

ソルバを動かす際に alph1 , alph2 , alph3 , delt (公差), 手数の上限を与え, コストがあらかじめ定めたコストの上限を超えないような着手の探索を行っていく。コストの上限の初期値は初期盤面のコスト以上に定める。コストの上限以内のコストで解が見つからなければ一定数(公差)ずつ上限を上げていき範囲内で解くことの可能な解を探索する。ある程度上限を上げてても(今回は 20 回とした)手数の上限より短い手数の解が見つからない場合は解くことを諦める。次に打てる手を見つげるときの高速化手法として 48×2 の 2 次元配列の表を使用している。表を初期局面について設定した後, 各手毎に微修正を繰り返し, 空でないタブローの一番上のカードそれぞれについて表を引くだけでタブローからタブローへの移動の手を見付けることが可能なようにしている。また同一局面の排除のためにハッシュ表を利用している。登録内容は 63 ビットのハッシュ値の他に, 16 ビットのその局面までの最短手数を含めている(合計 10 バイト)。解は盤面に対する着手を 16 進数で表記する。一つの着手を 16 進数表現で各フリーセルを 0 から 3, 各タブローを 4 から b, 各ホームセルを c から f とし, 移動元と移動先で表す(例: 6b とあれば 6 の位置のカードを b の位置に移動することを表す)。

[†]岡山大学大学院自然科学研究科

Graduate School of Natural Science and Technology,
Okayama University

2.3 学習データ

目的は十分な質と量をもつディープラーニングのためのデータセットを作成し、ニューラルネットに教師有り学習をさせることであり、ニューラルネットは現在の盤面に対してどのような着手が有効かを学習する。そのため与えるデータは“現在の盤面+着手”とする。

2.4 実験

データセットの作成にかかる時間について述べる。ここでは 2500 通りの初期盤面に対する解のデータセットの作成を行った例を示す。今回は表 1 に表す条件で実験を行った。

表 1 : 与えた変数

alph1	alph2	alph3	delt	手数の上限	n
2000	1000	1000	1000	120	12

2500 通りの初期盤面の解を求めるにあたって、まずは何通りかの変数の組み合わせを試した。同一の 50 通りの初期盤面に対して様々な組合せで実験をしたところ表 1 の組み合わせが最も早く解を求めることができたため今回はこの値を採用している。ハッシュ表のサイズは実行時に引数として n を与えた時に $10 * 2^{n+12}$ バイトとなるようにしている (例: $n=14$ の時は $10 * 2^{26} \approx 6.4 * 10^8$ バイト=640MB)。使用したマシンの仕様を以下に示す。

表 2 : 実験に使用したマシンのスペック

CPU:	Intel(R) Xenon(R) CPU E5-2630 v2 @ 2.60GHz を 2 個搭載
Memory:	256GB
OS:	Cent OS Linux release 7.5.1804
コンパイラ:	gcc 4.8.5

上記のマシンで 125 個の初期盤面に対しての解を求める操作を 20 並列で行い、合計 2500 通りの初期盤面の解を求めるのににかかった時間は約 12 分となった。

2.5 より良い解を得るために

学習の精度を上げるためには、作成するデータセットが可能な限り最短手順に近いことが望ましい。現在使用しているソルバは深さ優先探索に基づいているため、最長手順を指定するとその範囲の中で早く求めることの可能な長い解を求めることになりがちである。しかし、学習に与えるデータは最短手順に近いものの方が良質と考えられる。以下に良質な解を得るために 2 つの手法を紹介する。

方法 1 : 最短手順に近い解を得るために二分探索を行う

方法 2 : 得られた着手から無駄な手を削除する

方法 1 は、デメリットとしてソルバを複数回動かす必要があるため計算に時間がかかってしまうことがあげられる(数倍になると予測する)。学習に使うデータ量は膨大になる可能性があるため、この方法で

は時間が掛かり過ぎてしまう可能性がある。方法 2 であげられている無駄な手というのは、直接移動可能だが別のフリーセルやタブローを経由して移動している場合である(例: 95 と動かした直後に 5d と動かしたような場合)。学習のためのデータを作っているため、この場合 95 と動かしているデータを無駄な着手として除外し 5d と動かしているデータは動きとしては問題の無い手であるためそのまま残す。この方法はソルバの出力に対する処理であり、その出力をより短い手数に加工することも検討中である。学習の質と量の向上のため、現在これらの方法を取り入れたデータセットの作成方法の開発に取り組んでいる。

3. まとめ

今回はフリーセルの初期盤面からソルバを用いて解を探索し、その時点での局面と着手をデータとして大量に生産するための手法について述べた。その過程でフリーセルをなるべく使わないようにする方法などの前節に述べていない手法を検討したが、その手法の有効性は不明である。今後はフリーセルソルバの性能の向上と 2.5 節で取り上げた方法の実装に取り組んでいきたい。

謝辞

本研究の一部は、株式会社アースライズカンパニー支援により実施された。

参考文献

- [1] Elyasaf, Achiya, Ami Hauptman, and Moshe Sipper. "GA-FreeCell: Evolving solvers for the game of FreeCell." Proceedings of the 13th annual conference on Genetic and evolutionary computation. ACM, 2011.
- [2] Chan, Chris. "Helping Human Play Freecell."
- [3] Elyasaf, Achiya, Ami Hauptman, and Moshe Sipper. "Evolutionary design of freecell solvers." IEEE Transactions on Computational Intelligence and AI in Games 4.4 (2012): 270-281.
- [4] Paul, Gerald, and Malte Helmert. "Optimal solitaire game solutions using A* search and deadlock analysis." Ninth Annual Symposium on Combinatorial Search. 2016.