5T-01

# ランダムな試行の学習によって構成された15パズルの評価関数について

伊藤康太 †　　山本修身 ‡

名城大学理工学研究科情報工学専攻 †　　名城大学理工学部情報工学科 ‡

## 1　はじめに

本稿では，ゴール状態からのランダムな試行で得られたステップ数と盤面の状態のペアを教師データとしたニューラルネットワーク（NN）を用いることで，15 パズルを IDA*アルゴリズムなどのヒューリスティックサーチで解くための効率的な評価関数を構成した．ただし，この評価関数を用いていつも最適解が得られるわけではない．この評価関数を，最適解を必ず出力するマンハッタン距離関数と単純に比較して IDA*アルゴリズムでは，平均探索ノード数約 4,200 分の 1，平均計算時間約 14 分の 1 に減少させることができた．

15 パズルとはスライディングブロックパズル（SBP）の一種である．$4 \times 4$ の枠に入っている 1 から 15 までのコマを，空きマスを利用して決められたゴールの配置に揃えることがこのパズルの目的である．本稿では図 1 の配置をゴール状態とする．その他の SBP に 24 パズルや箱入り娘パズルなどがある．

これらのパズルは最適解を求めようとするとサイズに関して PSPACE 完全であることが知られており [1]，解くことは困難である．ただし，15 パズル程度であれば，パズルの構造に関係したパターンデータベース（PDB）やギャップ集合 [2] などの手法で効率的に解くことができる．しかし，それ以上のサイズのパズルになると，PDB の構築に膨大なメモリと計算時間が必要となり構成することが困難になる．

本研究の最終的な目的は，一般の SBP のための高性能な評価関数を同様の手法で構成することである．そのため，まずは比較的容易に解くことのできる 15 パズルで基本的な手法を開発し，それをもっと複雑な 24 パズルなどに応用することを考えた．ただし，倉庫番などの SBP ではゴール状態から逆向きに状態変化させることが，パズルを解く動作と等価になっておらず，そのまま本稿で述べる手法を適用することはできない．ここでは，ゴール状態からランダムに状態変化させたときの累積のステップ数と状態を NN に学習させて，与えられた状態を入力としてステップ数を出力する関数を評価関数として利用する．

PDB などの評価関数はゴール状態までの最短手順数の下界値を出力するため，許容的であると言い，IDA*アルゴリズムなどに用いると最適解が得られることが知られている．しかし，本稿では NN を用いるため構成する評価関数は許容的ではない．そのため，最適解でなくても最適解に近い解が得られるような評価関数の構成を目指した．また，評価関数の性能は，PDB を用いるとマンハッタン距離関数と比較して探索ノード数は約 100〜10,000 分の 1 となるため，それを目安とした．

## 2　比較に用いる評価関数-マンハッタン距離関数

マンハッタン距離関数とは，$\mathrm{MD}(p) = \sum_{k=1}^{15} d(k)$ と定義される関数である．ただし，$p$ は 15 パズルのパターンであり，$d$ は，$d(k) = |k \bmod 4 - i| + |\lfloor k/4 \rfloor - j|$ と表すことができる．また，$(i, j)$ はコマ $k$ の座標とし，ここでは左上から右下へ順に 1 から 15 までコマを並べたパターンがゴール状態であると仮定している（左上隅は空きマスである）．この評価関数は常にゴール状態までの最短手順数の下界値を出力するため許容的である．簡単に構成できるため評価関数の比較に用いられることが多い．

## 3　NN に与える教師データの作成方法と学習手順

### 3.1　ゴールからのランダムな探索

15 パズルのような SBP では，ゴールからランダムに動かしたコマの動作の系列を逆回しすることによって，あるパターンからゴールに戻る動きを求めることができる．それぞれの動きに現れるパターンとゴール状態からそのパターンに至るまでのスライドの回数 $s$(ステップ数) を評価関数値として，パターンとともにテーブルに書き込む．すでにテーブルに入っているパターンが現れた場合，$s$ の最小値で更新する．実際には $s$ を記録するより $|s - \mathrm{MD}(p)|$ を記録した方がよりよい評価関数を与えるテーブルとなることがわかっているため，それを利用する [3]．本稿ではゴールから 200 手ランダムに動かす試行を $10^4$ 回繰り返し教師データを作成した．

### 3.2　よく通るパターンからの探索

前節のように，ゴールからランダムに何回も動かすことで，ゴール付近の状態についてはテーブルに最適な動

| | | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 |

図 1　本稿における 15 パズルのゴール状態．

On evaluation functions for the fifteen puzzle generated by learning random trials
† Kota Ito, Division of information Engineering, Graduate School of Science and Technorogy, Meijo University
‡ Osami Yamamoto, Department of Information Engineering, Faculty of Science and Technology, Meijo University



Copyright ©2019 Information Processing Society of Japan.
All Rights Reserved.

きが記録できるはずである．しかし，15 パズルには膨大なパターンが存在するため，ゴールから離れれば離れるほど，遠回りの経路を含むパターンが多く記録される．15 パズルでは，何度もゴールから様々なパターンを探索すると，あるステップ数で同じパターンにたどり着くことがあり，多くのパターンはこのよく通るパターンから伸びていると考えられる．そこで，その高い頻度で出現するパターンを見つけ出し，そこからランダムな探索を行うことでよりよい教師データを得ることができると考えた．本稿では，最適ステップ数が 20 または 22 のパターンを列挙しておき，ゴールから $10^7$ 回または $10^9$ 回それらのパターンにたどり着くまで探索した．そのうち頻度が上位 50，100，500 パターンを抽出し，それらをよく通るパターンと定義した．さらに，それらのパターンをゴール状態として，70 手ランダムに動かす試行を 200 回繰り返し教師データを構成した．ただし，19 または 21 ステップ目に戻らないようにした．

### 3.3 15 パズルにおける NN の構成

NN の構成は図 2 に示すように，入力層 1 層，中間層 2 層，出力層 1 層の 4 層とした．15 パズルの評価関数に対応した最適な NN の構造はわからないため，中間層のノード数を変化させることでよりよい NN の構築を目指した．入力は空きマス “0” と “1” から “15” の数字 1 コマを 4bit（0000〜1111）表現とした 64 ノードとする．中間層は IDA*アルゴリズムでは何回も NN の計算をしなければならないため，計算量を考慮して 2 層とした．さらに 2 層とも同じノード数で 96，128，192 ノードのいずれかとした．出力は 1 ノードで，ステップ数からマンハッタン距離の和を引いた値とした．

NN の学習手順は以下のとおり: 最初に 2.1 節のゴールからのランダムな探索のみで構成した教師データを NN に学習させる．この結果，教師データには遠回りのパターンを多く含んでいるため，NN は大きな値を出力する．この NN に 2.2 節で構成したよく通るパターンの教師データに含まれているパターンを一度入力として入れて，NN の出力が教師データの値より小さければ，その教師データは削除して，よりよい教師データのみを NN に学習させることで NN による評価関数を構築した[1]．

## 4 実験結果およびまとめ

ランダムに生成した 100 個のパターンについて IDA*アルゴリズム（C++）で解いた[2]．NN の学習は Python で行ったが，評価関数としての NN の計算は IDA*アルゴリズムと同様に C++で記述した．また，NN の出力は MD($p$) を引いた値なので後でそれを足している．

計算結果を表 1 に示す．よく通るパターン 50 個の step20-NN192 で最も平均探索ノード数が減少し約 4,200 分の 1 になり，計算時間は約 7 分の 1 となった．同じく step20-NN96 で最も計算時間が減少し約 14 分の 1 になり，平均探索ノード数は約 2,700 分の 1 となった．全ての評価関数において最適解はほとんど得られていないが，最適解から平均で約 4 から 6 手のみ離れていた．

本稿では，よく通るパターンを見つけ出し，そこからランダムな試行で教師データを構成した．さらにそれを用いた NN による，PDB に近い性能を持つが許容的に近い 15 パズルの評価関数を構成した．

[1] 活性化関数：LeRU，最適化器：Adam 法，損失関数：最小二乗誤差，学習率：0.001，学習回数：20，バッチサイズ：1,024

[2] CPU：Xeon E3-1240 v3，メモリ：16GByte，OS：Windows 7，言語処理系：Python 3.6.3，GCC 5.3.0（最適化オプション：-O3）

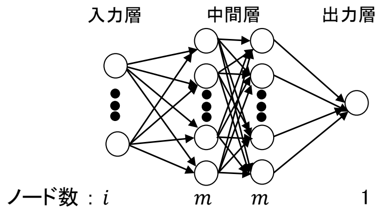


図 2 4 層からなる階層型 NN の構造．入力層 $i = 64$ ノード，中間層 $m = 96$ or 128 or 192 ノード．

表 1 よく通るパターンからの探索で得られた教師データで構成した NN を評価関数として 100 個のパターンを解いた結果．Node は平均探索ノード数，T は平均計算時間（秒），A は最適解が得られた個数．S は | 最適ステップ数 − ステップ数 | の平均，LT は NN の学習時間（分），LSum は教師データの個数を表す．step $k$-NN $m$ は $k$ ステップ目の高い頻度で出現するパターンからの探索を用いた NN であり，$m$ は中間層のノード数を示す．

| 評価関数 | よく通る経路の数 | Node | T(s) | A | S | LT(m) | LSum |
|---|---|---|---|---|---|---|---|
| MD | - | $6.07 \times 10^7$ | 21.7 | 100 | 0 | - | - |
| step20-NN96 | | $2.23 \times 10^4$ | 1.0 | 8 | 5.6 | | |
| step20-NN128 | 50 | $2.76 \times 10^4$ | 1.9 | 8 | 4.8 | 10 | 248,824 |
| step20-NN192 | | $1.45 \times 10^4$ | 2.0 | 5 | 5.6 | | |
| step20-NN96 | | $5.05 \times 10^4$ | 2.3 | 5 | 4.8 | | |
| step20-NN128 | 100 | $4.20 \times 10^4$ | 2.9 | 7 | 4.6 | 19 | 498,824 |
| step20-NN192 | | $4.90 \times 10^4$ | 6.7 | 5 | 4.5 | | |
| step20-NN96 | | $4.08 \times 10^4$ | 1.9 | 8 | 4.5 | | |
| step20-NN128 | 500 | $1.99 \times 10^4$ | 1.4 | 5 | 4.6 | 420 | 2,450,753 |
| step20-NN192 | | $1.68 \times 10^5$ | 22.1 | 7 | 4.7 | | |
| step22-NN96 | | $4.23 \times 10^4$ | 2.0 | 12 | 4.1 | | |
| step22-NN128 | 50 | $1.23 \times 10^5$ | 8.9 | 11 | 4.1 | 10 | 250,839 |
| step22-NN192 | | $1.71 \times 10^4$ | 2.4 | 4 | 5.6 | | |
| step22-NN96 | | $4.46 \times 10^4$ | 2.1 | 5 | 5.1 | | |
| step22-NN128 | 100 | $4.23 \times 10^4$ | 2.9 | 5 | 4.8 | 19 | 506,254 |
| step22-NN192 | | $1.99 \times 10^4$ | 2.8 | 9 | 4.7 | | |
| step22-NN96 | | $4.47 \times 10^4$ | 2.0 | 5 | 4.4 | | |
| step22-NN128 | 500 | $4.27 \times 10^4$ | 3.0 | 12 | 3.8 | 420 | 2,461,383 |
| step22-NN192 | | $8.80 \times 10^4$ | 12.6 | 8 | 4.1 | | |

### 参考文献

[1] Hearn, R. A. and Demaine, E. D.: ゲームとパズルの計算量. 上原隆平訳, 近代科学社 (2011)

[2] 山本修身, 佐藤根寛: ギャップ集合を用いた 15 パズルの最適解探索の高速化. 人工知能学会論文誌, Vol. 26, No. 2, pp. 419–426 (2011)

[3] 伊藤康太, 山本修身: ランダムな試行の学習による 15 パズルの評価関数の構成. 第 16 回情報学ワークショップ WiNF, D-4 (2018)



Copyright ©2019 Information Processing Society of Japan.
All Rights Reserved.