Vol.2017-GI-37 No.8
2017/3/7

ヒューリスティックス解法に基づく イラストロジックの難易度判定関数

金井 誠[1,a)]　伊藤 毅志[1]

概要：パズルゲームにおいて、問題の難易度表記の存在は、自分に合った問題を選ぶ上で有用な情報である。問題の見た目で分かる情報からは正確な難易度を推定することは難しく、問題を解く人間が解き始める前に正しい難易度を知るためには、問題作成者が予め難易度付けをしておくのが望ましい。しかし、人間が問題の難易度を付ける方法は様々あるものの、製作者以外の人間を用意したり、様々な要因を計測し、重み付けして要因を判断なければならないなど、そのどれもがコストの掛かる作業となっている。
そこで、本研究ではパズルゲームである「イラストロジック」を題材に、問題の難易度を複数の要因から自動的に判定するシステムの構築を行った。人間の用いるヒューリスティックスな解法に基づいたソルバーを実装し、問題の解答過程を分析することで、既存の手法に比べ人間の難易度判定に近い判定が行えるようシステムを設計した。判定実験では予め難易度付けがされた問題集を使い、既存の研究に比べ、元の難易度に近い値を判定出来ているか評価した。

Difficulty Evaluation Function of Illust-Logic Based on Heuristics Methods

Makoto Kanai[1,a)]　Takeshi Ito[1]

Abstract: In the puzzle game, presenting the degree of difficulty of problems is helpful for player in selecting a problem suitable for his level. It is difficult to estimate the degree of difficulty accurately by just looking information. In order to know the proper degree of difficulty before starting to solve the problem, it is required to prepare the difficulty for the person making the puzzle in advance. However, determining the degree of difficulty of problems manually is not only difficult but also expensive.
In this research, we constructed a system that automatically judges the difficulty level of problems from multiple factors using the puzzle "Illustration Logic". First, we implemented a solver based on heuristic solutions used by humans. Then, by analyzing the problem solving process, we designed a system that can make a judgment that is closer to human difficulty evaluation than the existing method. In the evaluation experiment, it showed that values close to the difficulty level of the original can be obtained than the existing research by using the problem collection attached the difficulty level in advance.

1. はじめに

パズルゲームにおいて、問題の難易度表記の存在は、自分に合った問題を選ぶ上で必要なものである。初級者にとって実力以上の高い難易度の問題は苦痛であるし、上級者にとって低い難易度の問題は退屈である。適切な難易度表記があれば、自分に合った問題を選ぶことが可能となる。多くのパズルゲームは問題の大きさ、あるいは要素数の様な数値が事前に提示されるが、その数値が問題の難しさにどの程度影響を及ぼすか判断するのは難しい。例えば、詰め将棋であれば、一般に手数が短いものほど易しい傾向があるが、手数は短くても難しいものや手数は長くても易しい問題もよくある。予め製作者から問題に難易度付けをしておくことは問題を解く人間（以下、解き手）が問題を解くことなく、真の難易度を知ることができる手段の1つで

[1] 電気通信大学
The University of Electro-Communications, Chofu, Tokyo 18–8585, Japan
[a)] kanai@minerva.cs.uec.ac.jp

あり、自分のスキルレベルにあった適切な問題を選ぶ上では重要な情報である。

しかし、人間が問題の難易度を付ける方法は様々あるものの、そのどれもがコストの掛かる作業となっている。主観的な方法としては人間が解いて値を設定する方法が考えられるが、その方法では作成者以外の人間を用意して難易度付けをさせる必要がある。また、客観的な方法では問題から難易度に関わる要因を抽出し、それを元に計算する方法が考えられるが、その方法では各要因の重みを設定する必要がある。判定する人の肩代わりになり、複数の要因の重み付けを自動で行うシステムがあれば、製作者が難易度付けを行うことのコストを軽減することができる。

そこでコンピュータに問題の難易度付けをさせることが考えられる。人間が作成した問題をコンピュータに読み込ませ、各問題の難易度を自動で判定することが出来れば、先ほど挙げた２つの方法が抱える問題を解消し、難易度判定による製作者の負担を軽減することができる。主観的な方法で見られる問題は解答を知らないコンピュータが問題を解いて判定することで回避でき、客観的な方法で見られる問題はコンピュータ上で定義された条件式を用いることで、手間を掛けることなく要因の計測、重要性の重みづけの自動化が可能である。

パズルゲームの問題に難易度を付けるにはソルバーが必要になる。その理由はパズルゲームの難しさは見た目よりも解く過程に強く現れることもあり、小さい問題が大きい問題よりも難しくなるためである。ソルバーには人間が用いる解法と解答手順に基づいたものと、論理的ルールに基づいたものなどがあるが、難易度判定に必要なのは前者である。人間と同じような解答手順を取ることで、人間が解答過程で感じる難しさを分析することが出来るようになり、人間が問題を選ぶための難易度判定が行えるようになると考えられる。

本研究ではパズルゲームである「イラストロジック」を題材に、問題の難易度を複数の要因から自動的に判定するシステムの構築を行った。人間の用いるヒューリスティックスな解法に基づいたソルバーを実装し、それを使って問題の解答過程を分析することで、論理的ルールに基づく手法に比べ人間の難易度判定に近い判定が行えるシステムを設計した。難易度判定システムには解答過程の特徴に基づく判定モデルを導入しているが、判定モデルのパラメータを難易度付けのされた問題集を学習することで決定している。システムの評価実験では予め難易度付けがされた問題集と被験者によって難易度が付けられた問題の２種類を使い、論理的解法に基づくシステムよりも人間の付けた難易度に近い値が推定出来るかどうかと、提案手法の汎化能力の高さを検証している。

2. イラストロジック

イラストロジックとはペンシルパズルの１種であり、格子状のマスを縦、横方向のヒント数字を元に白と黒で埋めていくことで最終的に絵が現れるというものである。白と黒の２色のものから、複数色存在するものもあり、問題の大きさも含め、今日までに多様な問題が作られてきている。本研究ではこの内、白黒２色の問題について注目する。

イラストロジックでは色を確定するための格子状の図形が書かれているが、本紙ではこれを**マトリクス**と呼ぶものとする。このマトリクスにおける色を確定する為の最小単位のことを**マス**、横方向に連なるマスの並びのことを**行 (row)**、縦方向に連なるマスの並びのことを**列 (column)**、行と列の総称を**ライン**と呼ぶものとする。ラインには黒で確定できるマス数を表す任意個数の数字が添えられているが、これを本紙では**ヒント**と呼ぶものとする。用語の対応を図 1 に示す。

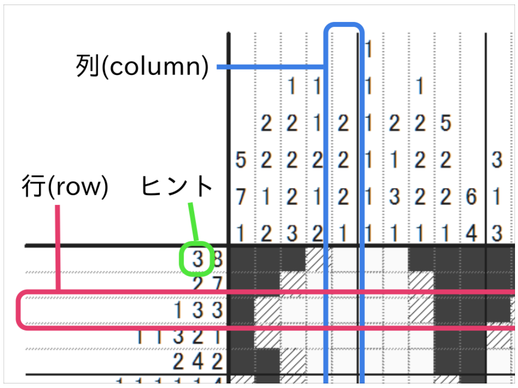

図 1　イラストロジックの用語の対応

3. 関連研究

3.1 人間の解答手順を元にした難易度判定

イラストロジックの難易度判定関数を作成した研究として、伊藤による人間の解答過程に基づいたソルバーから問題の難易度判定関数を構築した研究がある [1] 。

伊藤は被験者が問題を回答する過程を分析し、人間がどういった解法を用いるか、どういった解法から順に着手するかについて調べた。その結果、ヒューリスティックスな解法は６つのカテゴリに分けられ、それらには適用される順番があることが示された。伊藤が示したカテゴリは次の６つであり、人間は上から順に適用するものとされている。

カテゴリ１　問題中で大きなヒント数字の箇所を検討する解法

カテゴリ２　ラインの色が全て確定するかどうかを検討する解法

カテゴリ３ 端を考慮した配置から取りうるところを検討する解法
カテゴリ４ 配置から塗り得ないところを検討する解法
カテゴリ５ 既に塗られた大きさから、取りうる色が確定しているものを検討する解法
カテゴリ６ 上記の規則に適用されない解法

そして、６カテゴリを組み込んだソルバーを実装し、その解答過程に関するデータと難易度の関連性を調べた。そして、解答過程に関するデータを元に難易度を判定する関数を、問題のサイズ毎に構築した。

伊藤の研究では実際に人間がイラストロジックの問題を解く際の過程を調べており、それに基づいてソルバーや判定関数を構築しているため、導き出された判定関数はある程度の信憑性があるものと考えられる。しかし、伊藤が作成した判定関数は２種類の問題サイズのものしか作られておらず、他のサイズの問題に対しては適用できない。２つの関数の式は同じ構造をしているため、式中の定数値を何らかの方法で問題サイズを変数した形で置き換えることで対応できる可能性はあるが、それらの具体的な改善方法は示されていない。

伊藤と同様に、人間が用いる解法に基いて問題難易度の判定を行うシステムを構築した佐藤は、自身で作ったシステムに根本的に構造的な問題があることを述べている [2] 。佐藤は解法の適用順序をレベル付けという形で行い、使用した解法のレベルに応じて問題の難易度を判定している。そして、難易度付けがされた問題集を用いて、判定した難易度と問題集上の表記難易度の差異を検証したが、判定した難易度と表記難易度の相違に見られる傾向が、問題の大きさによって異なることを示している。これに基づき、佐藤は「問題サイズの違い」が人間コンピュータ（ソルバー）で異なる影響を与える仮説を立てている。このことから、人間の解答過程及び、問題の情報が人間の感性に与える影響を忠実に再現するためには、問題の大きさに関する情報も必要になると考えられるのである。

3.2 画一的数理操作に基づく難易度判定

機械的にイラストロジックの問題の難易度判定をした研究として、 Batenburg らによる画一的数理操作に基づく判定関数がある [3] 。論文中で Batenburg らは、イラストロジックの問題を画像から自動生成し、生成した問題の難易度判定の為に `Difficulty()` と呼ばれる難易度判定関数を構築した。

`Difficulty()` は settle 操作と呼ばれる数理操作を用いたソルバーを利用し、問題が完全に解かれるまでにソルバーが操作を適用した回数を問題の難易度としている。settle 操作は単一のラインの現在のマトリクスの状態とヒントから、そのライン中で確定できるマスを調べる操作である。settle 操作を全行、全列交互に適用していくことで仮定法を用いらずに解ける問題は全て解くことができる。`Difficulty()` が利用しているソルバーは全行、もしくは全列毎に settle 操作を試行することを操作の１単位としており、この回数が `Difficulty()` が示す問題の難易度となる。

Batenburg は `Difficulty()` によって、生成した複数の問題が異なる難易度を持つことを示した。`Difficulty()` は人間が問題を解いた際のデータを必要としないため、人間が問題を解くことなく即座に問題難易度を判定させることが可能である。しかし、 `Difficulty()` で用いられている解法は settle 操作のみであり、人間のように複数の解法を用いて解く方法とは異なった方法である。また、`Difficulty()` では「操作の適用回数」のみが難易度の基準として用いられているが、それだけが難易度を測る指標であるとは考えづらい。例えば、簡単な解法だけを用いて解ける問題は他の問題より簡単だろうし、他の問題より難しい解法を用いらなければ解けない問題はより難しく感じると考えられる。そのため、 `Difficulty()` は人間の感じる難易度の基準を十分に反映出来ていない可能性があり、人間が問題を選ぶための難易度の指標として用いるには疑問点が残る。

4. 提案手法

4.1 システムの概要

本研究では人間の使う解法に基づいたソルバーを実装し、それを使って問題の解答過程を分析することで、より人間の難易度判定に近いものを出力できるようにシステムを構築した。本システムではイラストロジックの問題の難易度を判定する際に、次のような手順を踏む。

(1) 解答過程の特徴に基づく判定モデルを構築
(2) 人間の使う解法に基づいたソルバーを使って問題を解き、解答過程に関する特徴量を得る
(3) 特徴量を使い、判定モデルから難易度を判定する

4.2 特徴に基づく判定モデル

本研究では式１のような一般線形モデルの判定関数を利用し、判定を行った。v は判定関数で判定された難易度値（スカラー）、 $\boldsymbol{\beta}$ は各特徴量の係数ベクトル、 $\boldsymbol{x}$ は解答過程の分析から得られた特徴ベクトルを表す。各特徴に対する重みを表す係数ベクトル $\boldsymbol{\beta}$ については、既に難易度付けのされた問題集を教師とした、回帰分析による学習で求めた。

$$v = \boldsymbol{\beta}^T \boldsymbol{x} \tag{1}$$

4.3 使用した解法

今回、判定モデルに用いる特徴はソルバーで利用する

ヒューリスティックス解法群から構築した。ヒューリスティックス解法群は伊藤の定めた６つのカテゴリを基に、不足していた解法を補う新たなカテゴリとして「全パターンを考慮する解法」を加えた、全７カテゴリとしたものを使用した。追加した解法については、本研究で用いた問題群において確かに使われたことを確認している。

伊藤が示したように、解法は特定の順で使われることが明らかになっている。本研究でもそれを反映し、確定できるマスがあるかどうか検討し始めたときは人間が早く適用する解法についてのみ検討し、高度な解法はなるべく使われないように実装した。なお、先述の新規に追加したカテゴリは、処理の関係上、最も後に使うべき解法と判断したため、一番最後に検討されるように実装した。

4.4 判定モデルに用いる特徴

実際に判定モデルに用いる特徴量は各解法カテゴリ毎の、①「解法の適用回数」と②「解法によって色が決定したマス数」で構成されている。①「解法の適用回数」を特徴として選んだ理由は、Batenburg らの `Difficulty()` の実装や、石田らによるナンバープレイスを題材とした研究において [4] 、最終回にたどり着くまでの解法の適用回数が問題の難易度判定に有効であることが示されていたためである。②「解法によって確定したマス数」を特徴として選んだ理由は、佐藤の研究において、解法のレベルだけでなく、問題の大きさなどの要素も難易度判定に必要であることが示されていたためである。問題の大きさを直接用いるのではなく解法カテゴリ毎に確定したマス数を用いた理由としては、「確定したマス数」が「難しい解法が必要になる箇所が多いことを表す」のを期待してである。イラストロジックは解答途中の段階でも視覚的な要素から最終結果を推測することが可能であり、より難しい解法で確定するマス数が多ければ、解答途中での推測が困難になるのではないかと考えたためである。

これらの特徴を全て説明変数として用いた場合、判定関数の表現力は上がるものの過学習になる恐れがある。そこで、全ての特徴を含む構成に加え、① と ② の２種類の特徴量の一方を全解法カテゴリで合計値を取ったものでも実験を行い、変数構成の違いによる汎化能力の比較を行った。用いた変数構成は以下の３つになる。

(1) 総適用回数＋確定したマス数（８変数）
(2) 適用回数＋確定した総マス数（８変数）
(3) 適用回数＋確定したマス数（１４変数）

4.5 係数の学習

学習ではまず学習用の問題集の難易度値を $[0, 1]$ の範囲で正規化し、その後に回帰分析によって係数ベクトルを求める。学習において、目的変数に該当する元の問題集の難易度の値については、最小のものが０、最大のものが１となるように正規化を行った。

学習用の問題群には DS パズラー [5] 収録の 15x15 の問題 159 題を利用した。問題の大きさとして 15x15 を選んだ理由は、用意した問題集の中ではこのサイズが比較的数が多く、学習や評価実験が容易であったためである。

5. 評価実験

実験では既に難易度付けのされた問題群を用いた実験と、被験者によって付けられた難易度との差異を計測する。

5.1 問題群の表記との差異を測る実験

5.1.1 目的

提案手法による判定関数が、 `Difficulty()` よりも元の問題集の表記難易度に近い値を判定できるかを評価する 。また、学習用問題群とは別の問題群に対応できるかどうかの汎化能力についても評価を行う。

5.1.2 実験方法

学習して求めた係数を使い、評価用の問題群を判定関数で判定し、判定した難易度値と元の問題集で表記されている難易度の差の絶対値の平均を計測した。判定用の問題群としては、以下の２パターンを用意した。それぞれ、元の難易度を表現できるかどうかと、各変数構成毎の汎化能力の高さを検証している。

パターン１　DS パズラー収録の 15x15 の問題 159 題
パターン２　ｉパズラー [6] 収録の 15x15 の問題 21 題

なお、これらの問題は提案手法のソルバー、`Difficulty()` のソルバーのどちらでも全問解けることを確認している。

5.1.3 実験結果

パターン１の実験結果が表 1 である。

表 1 学習時と同じ問題群を判定した難易度の差分

判定関数	絶対値平均	相関係数
`Difficulty()`	0.347	0.301
総適用回数＋確定したマス数	0.279	0.367
適用回数＋確定した総マス数	0.263	0.437
適用回数＋確定したマス数	0.244	0.547

表 1 の各問題の表記難易度と判定難易度の差分の大きさの平均、すなわち、絶対値平均を見ると、どの変数構成においても、提案手法によるものの方が小さい値を示している。また、相関係数についてはどうの変数構成においても、提案手法によるものの方が高い数値を示している。この内、最も絶対値平均が小さくなったのは「適用回数＋確定したマス数」の結果である。

次に、パターン２の実験結果が表 2 である。

表 2 から、「総適用回数＋確定したマス数」と「適用回数＋確定した総マス数」については、 `Difficulty()` よりも 絶対値平均が小さくなり、相関係数は大きくなった。一

表 2 学習時と異なる問題群を判定した難易度の差分

判定関数	絶対値平均	相関係数
Difficulty()	0.413	0.000
総適用回数＋確定したマス数	0.343	0.308
適用回数＋確定した総マス数	0.360	0.152
適用回数＋確定したマス数	0.412	-0.125

方、表 1 では最も絶対値平均が小さくなった「適用回数＋確定したマス数」の絶対値平均は、 `Difficulty()` によるものと殆ど変わらず、相関係数は小さくなった。

5.1.4 考察

表 1 から、提案手法による判定の方が差分の絶対値平均が小さくなり、 `Difficulty()` に比べてより人間が付けた難易度に近い値を判定できることが示された。表 2 からも「総適用回数＋確定したマス数」と「適用回数＋確定した総マス数」の変数構成からは、提案手法による判定関数の方が人間の判定に近い値を判定出来ることが示されており、一定の汎化能力も備わっていることが考えられる。

変数構成の違いに着目すると、「総適用回数＋確定したマス数」と「適用回数＋確定した総マス数」の変数構成については、 `Difficulty()` よりも小さい絶対値平均と高い相関係数を示しており、学習と評価で異なる問題集を用いた場合についても `Difficulty()` よりも良い結果が示されることがわかった。一方、「適用回数＋確定したマス数」の変数構成については、表 1 については他の構成よりも良い結果を示せたものの、表 2 では `Difficulty()` と殆ど変わらない値を示した。絶対値平均は `Difficulty()` と殆ど変わらず、相関係数については逆に小さくなってしまった。提案手法の他の変数構成は正の相関を示していたのに対し、「適用回数＋確定したマス数」は負の相関を示す結果になってしまったことから、この変数構成では上手く問題の難易度を表現できていないことになる。このことから、「適用回数＋確定したマス数」の変数構成については過学習が起きている可能性が考えられる。

相関係数について見てみると、提案手法によるものは学習時と同じ問題を用いたパターン 1 の結果（表 1）では「適用回数＋確定したマス数」が最も高い相関を示したが、パターン 2 の結果（表 1）では「総適用回数＋確定したマス数」が最も高い相関を示している。`Difficulty()` の相関係数よりも高いことからも、一定の有用性があるものと考えられる。

5.2 被験者による難易度付けとの差異を測る実験

5.2.1 目的

節 の実験と同様に、提案手法による判定関数が `Difficulty()` よりも元の問題集の表記難易度に近い値を判定できるかを評価する。ここでは、複数の被験者によって付けられた難易度と、判定関数によって付けられた難易度値が相関を持つかどうかを調べる。

5.2.2 実験方法

まず、被験者に問題を解かせ、それぞれに難易度を付けさせた。その後、節 の実験で学習した係数を用いて被験者に解かせた問題群を評価し、被験者に解かせた問題群と相関を持つかどうか調べた。

被験者に解かせた問題には「ロジックパラダイスミニ VOL.4」[7] に収録されている、15x15 の大きさの問題８問を利用した。この問題集では既に５段階の難易度評価が行われているが、殆どの難易度が問題サイズによって決まっているため、15x15 の大きさの問題については全て難易度が 1 となっている。被験者にこの８題に対して５段階の難易度を付けさせ集計し、全員の結果で平均を取った後に 1 が０、５が 1 となるように正規化を行った。

被験者の構成は以下の通りである。

- 男性 ５名
- 女性 １名

難易度判定に用いた関数は節 5.2.2 で用いたものと同じ４種類である。

5.2.3 実験結果

被験者に判定させた難易度の平均は表 3 のようになった。

表 3 被験者による難易度判定の結果

問題番号	難易度平均	正規化後
1	1.86	0.214
2	1.86	0.214
3	3.57	0.643
4	3.29	0.571
5	5.00	1.000
6	4.00	0.750
7	4.71	0.929
8	3.29	0.571

５問目と７問目の問題が最も難易度の高い部類の問題と判定されており、逆に 1 問目と 2 問目が最も低い難易度として判定されている。

次に難易度判定関数による判定結果を表 4 に示す。

表 4 各判定関数による難易度判定値

問題番号	1	2	3	4
1	0.231	0.322	0.265	0.141
2	0.077	0.063	0.033	0.551
3	0.269	0.583	0.459	1.606
4	0.269	0.473	0.475	0.357
5	1.000	0.800	0.870	1.100
6	0.385	0.614	0.439	0.451
7	0.923	0.776	0.800	0.946
8	0.231	0.518	0.529	0.543
相関係数	0.860	0.953	0.930	0.548

最上段の行の番号は次のように各判定関数に対応している。

(1) Difficulty()
(2) 総適用回数＋確定したマス数
(3) 適用回数＋確定した総マス数
(4) 適用回数＋確定したマス数

また、表 4 の結果を折れ線グラフにしたものを図 2 に示す。グラフを見ると黄色の「適用回数＋確定したマス数」の結果のグラフの 3 問目のところに大きな山があるのが目立つ。青の「Difficulty()」の結果について見てみると、黄色程では無いが、5 問目、7 問目の山が尖っていたり、8 問目が低く出ているなど、尖ったグラフになっていることが分かる。他の 2 つ（「総適用回数＋確定したマス数」と「適用回数＋総確定したマス数」）については先ほど示した 2 つに比べるとなだらかなグラフを示しており、3～8 問目間の差異が比較的小さいことが読み取れる。

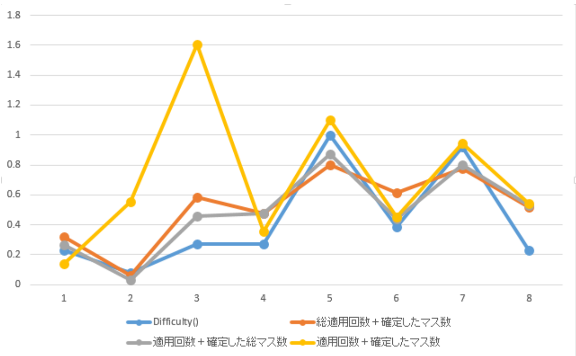

図 2　表 4 の各関数の平均のグラフ

次に、表 3 と平均と表 4 の各判定関数の判定値の差分を表 5 に示す。最上段の行の番号の対応は表 4 と同様である。

表 5　各判定関数による難易度判定値の差分

問題番号	1	2	3	4
1	-0.017	-0.108	-0.051	0.073
2	0.137	0.151	0.181	-0.337
3	0.374	0.060	0.184	-0.963
4	0.302	0.098	0.096	0.214
5	0	0.200	0.130	-0.099
6	0.365	0.136	0.311	0.299
7	0.006	0.153	0.129	-0.017
8	0.340	0.053	0.042	0.028
絶対値合計	1.541	0.959	1.124	2.030
相関係数	0.860	0.953	0.930	0.548

この差分は表 3 の値から表 4 の値を引いているため、負の値は表 4 の値の方が大きいことを示し、正の値は表 4 の値の方が小さいことを示している。また、絶対値合計は各差分を絶対値を取った後に足し合わせた値である。相関係数は差分を取る前の値を用いて、表 3 の正規化後の平均との相関係数を計算したものである。

表 4 中の 1 の結果と 2～4 の結果を比べてみると、相関係数が最も高くなったのは 2 であり、次いで 3 が高い数値を示した。2 と 3 は提案手法によるものであり、4 についても同様の結果が期待できるが、実際には最も低い結果を示しており、1 の結果と比べても明らかに低い相関係数を示している。絶対値合計が 2 が最も小さく、4 が最も大きい値を示していることからも、4 の判定関数が他の関数に比べてあまり良い結果を示せていないことがわかる。

5.2.4　考察

表 4 の相関係数を見てみると、提案手法によるものの結果の内、「総適用回数＋確定したマス数」と「適用回数＋確定した総マス数」の変数構成については、比較対象の Difficulty() のものよりも高い相関係数の値を示した。これは Difficulty() よりも提案手法の 2 つの変数構成の判定関数の結果の方が、問題間の難易度の関係性が被験者の付けたものに近いことを示しており、提案手法の判定関数がより人間の判定に近い難易度を表現できていることが示唆された。相関係数だけでなく表 5 の平均差分の絶対値合計も小さいことからも、提案手法の方が被験者の付けた難易度に近い値が示されていると考えられる。

一方、提案手法の変数構成の 1 つである、「適用回数＋確定したマス数」の判定結果の平均は Difficulty() よりも悪い結果を示している。表 4 の結果を見てみると相関係数が明らかに他の判定関数よりも悪く、表 5 の平均差分の絶対値も 4 つの関数の中では一番大きくなっている。

ここで、図 2 のグラフについて注目してみると、「適用回数＋確定したマス数」のグラフである黄色の線は、3 問目の難易度値が明らかに他の判定関数よりも大きな値を示している。被験者による難易度付けの平均や、他の判定関数による 3 問目の判定値を見る限り、この 3 問目の問題が他の問題に比べて頭抜けて難しい問題であるとは考えづらい。よって、「適用回数＋確定したマス数」による 3 問目の判定値は不当に高いものではないかと考えられる。

「適用回数＋確定したマス数」の判定関数は節 5.2.2 の結果でも汎化能力の低さが指摘されており、3 問目の判定値が不当に高いことについても、この判定関数の汎化能力が低いことに起因するのではないかと考えられる。ここで、提案手法による各判定関数の係数について確認したところ、一部の係数が他の判定関数よりも大きな値を示していることがわかった。このことから、「適用回数＋確定したマス数」の判定関数は懸念していた過学習になっているものと考えられ、節 5.2.2 の結果と合わせて、汎化能力が他の変数構成よりも劣っていることがわかった。

6.　おわりに

本研究では「イラストロジック」の為の問題難易度判定システムを作成し、それが論理的ルールに基づく手法よりも人間が付けたものに近い判定が行えることを示した。イ

ラストロジックの難易度判定では解答過程に基づいた判定をする必要があると考えられる。しかし、既存研究で提案されていた難易度判定関数は有用と思われる特定の情報が欠けていたり、人間の用いる解法とは大きく異なる解法を用いて問題を解いた解答過程に基づいて判定を行っているため、人間にとっての難易度を判定する関数としては問題がある。そこで、人間に近い解法で問題を解くソルバーを実装し、解答過程に関する複数の要因に基づいて難易度を判定できる関数を作成した後、難易度付けのされた問題を学習することで判定関数のパラメータを決定した。比較実験を行った結果、提案手法によって、理的ルールに基づく手法よりも人間の判定に近い難易度判定関数が作成できることが示された。

今後の展望としては、構築した判定関数の精度を向上させていくことが考えられる。そのためには現在の解法の構成が妥当であるか、それらの解法から導き出される特徴量が妥当であるか検討する必要がある。説明変数に用いる特徴については他にも考えられ、イラストロジックにおける人間の認知に関する特徴を用いることで、より人間の判定基準に近い難易度判定が行えることが考えられる。

参考文献

[1] 伊藤毅志：ヒューリスティックスを用いたロジックパズルの難易度の自動評価, ゲームプログラミングワークショップ 2005 論文集, No. 15, pp. 146-149, 2005
[2] 佐藤金吾:「イラストパズル」の難易度について, 法政大学多摩研究報告, Vol. 23, pp. 17-75, 2008
[3] K Joost Batenburg, Sjoerd Henstra, Walter A Kosters, Willem Jan Palenstijn : Constructing Simple Nonograms of Varying Difficulty, Pure Mathematics and Applications, Vol. 20, pp. 1-15, 2009
[4] 石田伸輔, 岩波拓, 高瀬治彦, 北英彦, 林照峯：数独問題を評価するための指標に関する一考察, 情報科学技術フォーラム一般講演論文集, Vol. 6 No. 2, pp. 437-438, 2007
[5] DS パズラー ナンプレファン＆お絵かきロジック, TDK コア, 2006
[6] お絵かきロジック by i パズラー, 世界文化社, 2013
[7] ロジックパラダイス・ミニ VOL.4, 学研プラス, 2016