

## スリザーリンク解答プログラムの開発と問題自動生成の検討

## Development of Slither Link Solver and Examination of Automatic Problem Generation

土井 佑香<sup>†</sup>  
Yuka Doi松本 実夕<sup>†</sup>  
Miyu Matsumoto金田 真輝<sup>†</sup>  
Maki Kaneda荒川 正幹<sup>†</sup>  
Masamoto Arakawa

## 1. はじめに

本研究ではペンシルパズルの一種であるスリザーリンクを対象とした。図 1 にスリザーリンクの問題例 (左) とその解答 (右) を示す。マスに書かれている数字は、周囲に引くことができる線の数を表す。この数字を手がかりとし、交差や枝分かれがないように線を引き、ひとつの大きなループを作ることがスリザーリンクの目的である。

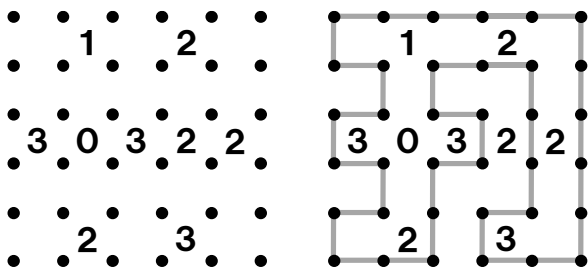


図 1 スリザーリンクの例

これまでに、スリザーリンクに関する研究がいくつか行われている [1 - 7]。コンピュータを用いてスリザーリンクを解く方法は、大きく 2 つに分けられる。

ひとつの方法は、スリザーリンクを制約充足問題や整数計画問題として定式化し、汎用ソルバを用いて解く方法である。Sugar[1]、ZDD[2]、GLPK[3]などを利用したスリザーリンクの解法が報告されている。

もうひとつの方法は、スリザーリンクに特化したアルゴリズムを開発することである。スリザーリンクでは、ひとつの大きなループを作ることが目標であり、これは全てのマスについてループの内側か外側かを決定することと等価である。Herting[4]は、マスのグループ分けにおいて、考えられるすべての組み合わせを列挙し、枝刈りのための規則を自動的に生成する手法を提案した。また Liu ら[5]は、各マスの頂点から引かれる線の数に着目し、ソルバを高速化する研究を行った。市岡[6]は、大域的なループの分割を効率的に発見する手法を提案した。

また、関連する研究がいくつか行われている。佐藤[7]は個々の規則について詳細に検討し、人間が解く場合の難易度を評価した。白井ら[8]は、独自に実装したソルバを利用し、スリザーリンクの問題を自動的に生成する方法を検討した。

本研究の目的は、スリザーリンクの問題を高速に解くことが可能なアルゴリズムを開発し実装することである。また、作成したソフトウェアを利用して自動あるいは半自動的に問題を生成する手順について検討した。

## 2. スリザーリンク解答アルゴリズム

既往の研究を参考に独自の問題解答アルゴリズムを考案し、プログラムを作成した。前述の通り、スリザーリンクは各マスをループの内側と外側のグループに分けることで解けるため、バックトラッキングを用いた。グループが決定していないマスについて、そのマスが内側か外側かを決定する処理を再帰的に行う。ルールに反しないように全てのマスのグループが決定されれば、その盤面が解となる。

どのような問題であってもバックトラッキングによって解答を求めることが可能であるが、計算時間短縮のため枝刈りによる効率化が必要である。枝刈りの手法は、あるマスの周囲の情報を用いる局所的解法と、全体でひとつのループになることを利用する大域的解法に分けられる。局所的解法とは、あるマスの周囲の状況を確認し、スリザーリンクのルールに従って、マスがどちらのグループに属するか決定する方法である。文献[5][6]などに記載されている規則の他、文献[4]に類する方法により独自に発見した規則を実装した。

## 3. 解答ソフトウェアの実装

考案したアルゴリズムに従いスリザーリンクを解くソフトウェアを実装した。実行画面例を図 2 に示す。問題の解を求める機能の他、ファイルへの入出力や盤面の編集機能などを備えている。このソフトウェアを利用することで、スリザーリンクの問題を作成することが可能である。

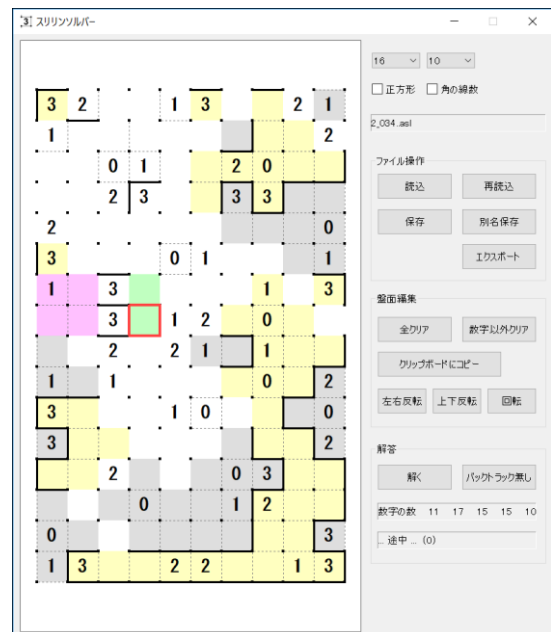


図 2 作成したソフトウェアの実行画面例

<sup>†</sup> 宇部工業高等専門学校

#### 4. 解答性能の評価

作成したプログラムの性能を評価するため、いくつかの問題について解答時間を測定した。使用した計算機の CPU は Intel core i5-8400、プログラミング言語は C++、コンパイラは Visual Studio 2017 である。

性能評価に用いた問題は、ペンパ全集スリザーリンク [9] に掲載されている全 288 問である。評価結果を図 3 に示す。横軸は解答時間であり、対数軸で表している。解答時間は問題のサイズに大きく依存するため、サイズごとの平均値と標準偏差を示す。サイズが大きいほど実行に時間がかかる傾向が確認されたが、最大サイズの問題であっても平均 8.6 ミリ秒と高速に解くことが可能であった。

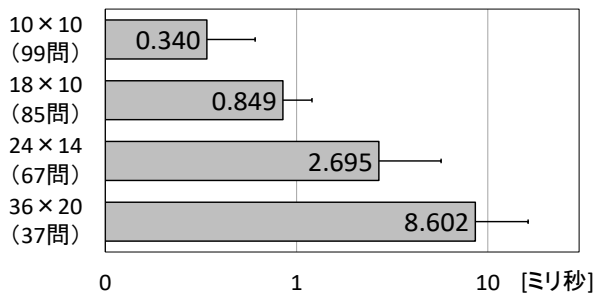


図 3 ペンパ全集の問題に対する解答時間

#### 5. 問題生成

開発したソフトウェアを利用し、スリザーリンクの問題を半自動で作成する手順は以下の通りである。

1. 数字の配置を決定する
2. 任意のマスに 0 から 3 の数字を設定する
3. 解答アルゴリズムによる探索を行う
4. 全マスに数字があり解が一意に定まれば問題完成
5. 必要に応じて任意の数字を削除する
6. 2.へ戻る

このアルゴリズムに従い、著者らが 10 × 10 の盤面について問題の生成を行った結果、平均 10 分程度で問題を作成することが可能であった。作成した問題の例を図 4 に示す。作成した 32 問について見た目や難易度、解き味を検証した結果、既存の問題と比較し遜色ないものであった。なお難易度は著者らの解答時間で評価した。

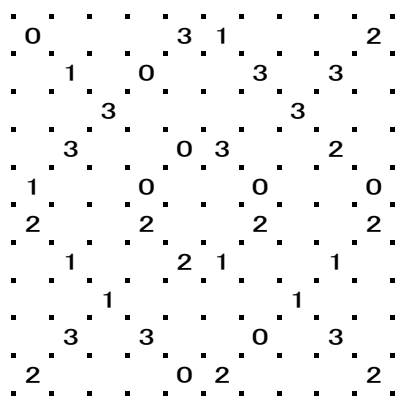


図 4 作成した問題の例

また、解答プログラムを用いて自動的に問題を生成するアルゴリズムを開発し性能を評価した。15 × 15 の盤面について様々な数字の配置を指定し問題を生成した結果、平均 2 分程度で作成可能であった。生成された問題の例を図 5 に示す。これらの問題を評価した結果、既存の問題と同程度の品質であることが確認された。

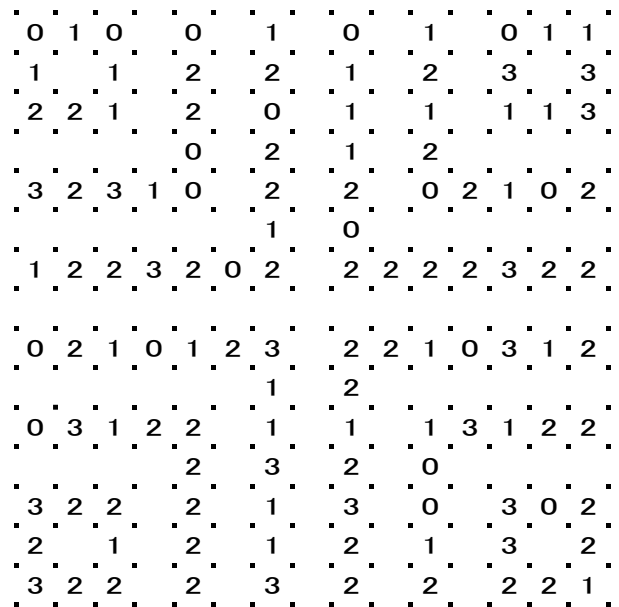


図 5 自動で作成された問題の例

#### 6. おわりに

本研究ではスリザーリンクの解答アルゴリズムを考案し、ソフトウェアとして実装した。性能について検証した結果、全ての問題を非常に高速に解くことが可能であった。また、作成したソフトウェアを利用して問題を生成するとともに、自動で問題を生成するプログラムを開発し自動問題生成を行った。その結果、既存の問題と遜色ない品質の問題を生成することに成功した。

今後の展開としては、枝刈りのための規則をより網羅的に探索することによる実行時間の短縮、問題の難易度を推定するアルゴリズムの開発と実装、提案手法の他分野への応用などがあげられる。

#### 参考文献

- [1] 田村直之, CSPSAT 研究会, 2008.
- [2] 吉仲亮 他, 種々のリンクパズルへの応用, オペレーションズ・リサーチ: 経営の科学, 57(11), 616-622, 2012.
- [3] 石濱友裕, 久野誉人, 整数計画法を用いた高速な Slitherlink パズルの解法, 情報処理学会論文誌, 54(8), 2103-2108, 2013.
- [4] S. Herting, A rule-based approach to the puzzle of Slither Link, 2005.
- [5] T. Y. Liu, et al., Solving the Slitherlink Problem, 2012 Conference on Technologies and Applications of Artificial Intelligence, 2012.
- [6] 市岡耕平, スリザーリンク自動解答アルゴリズム, <http://hp.vector.co.jp/authors/VA010341/slitherlink/>, 2003.
- [7] 佐藤金吾, パズル「スリザーリンク」の難易度について, 法政大学多摩研究報告, 18, 25-82, 2003.
- [8] 白井裕己 他, スリザーリンク解答システムと問題作成システム, ゲームプログラミングワークショップ 2006 論文集, 32-39, 2006.
- [9] ペンパ全集スリザーリンク 1, ニコリ, 2017.