

# 計算機を用いた数独解法アルゴリズムに関する一考察

Consideration concerning the Number Place method algorithm with computer

CS06 石川 達也  
指導教員 米山 秋文

## 1. 緒言

この研究を始めたきっかけは、数独の問題をインターネット上で聞き、数独における問題の作成や解を導くためにどのような方法を使用しているかなどに疑問を持ったからである。数独のマス目に応じた解法の考察を行い、最適アルゴリズムを構築し、プログラムに知能を付け加えることで、数独の問題をすばやく解くようにすることを研究目的とする。

## 2. 研究のアプローチ

### 2-1 バックトラック法

バックトラック法とは、ある解を求めるときに、可能性のある手順を順に試していき、その手順では解が求められないと判明した時点で一つ前の状態に戻って別の手順を試す、という方法である。

すべての可能性をしらみつぶしに試していく方法に近いが、計算途中で絶対に解が出ないと分かっていた場合は最後まで計算せずにその場で諦めて引き返すため、純粋な総当りよりは効率がよい。

解を求めるのにかかる時間はばらつきが大きく一定しないが、もっとも単純で記述しやすい方法であるため、広く用いられている[1]。

### 2-2 知能

知能 1:ある枠内で、確定している数字は他のマスには入らない。

知能 2:ある枠内で、ある数の候補が1 マスの場合、そのマスにその数字が入る。

という2つの知能を付け加え、高速化を計る。

## 3. 結果

今回の結果の検証に用いたプログラムはレベル:普通でヒント数:20 のものである。レベルとは、問題の難易度を表すものであり、ヒント数とは初期に配置されている数のことである。

問題	時間/秒
1	55.35
2	30.63
3	176.45
4	141.62
5	73.17
6	183.95
7	201.17
8	58.27
9	410.96
10	360.09
平均	169.166

表 1.バックトラック法のための計算結果

問題	時間/秒
1	47.99
2	16.77
3	145.96
4	112.18
5	56.42
6	176.67
7	194.45
8	42.75
9	360.09
10	307.80
平均	146.108

表 2.知能を付け加えた場合の計算結果

## 4. 結論

バックトラック法でのヒント数と解を解く時間の関係を検証した結果、レベルが普通でも難問でもヒント数が多い場合、計算時間が短いことがわかった。次に、ヒント数が徐々に少なくなっていくにつれ(28, 25, ..., 20)計算時間が長くなることがわかった。

また、レベルやヒント数に関係なく計算時間がかかる場合があることがわかった。これは、ヒントの初期位置によって計算時間が変わっている事、つまり、問題の先頭の方にヒント数が少ないと処理に時間がかかるためである。今回の結果の中には無いが、ヒントの初期位置によっては難問でも普通より速い時間で計算される場合があることが判明した。

知能を付け加えることで、結果として計算時間が短縮されたことが表を見てわかる。しかし、計算時間がとても短い時間になると知能を持たせても持たせなくてもかわらない結果となってしまった。また、ヒントの初期位置の関係は解消できなかった。

## 5. 今後の発展

今後の課題としては、ヒントの初期位置に左右されないような新たな知能を付け加え高速化を計る。もしくは、別のアルゴリズムで数独のプログラムの作成をする。

## 文献

- [1] IT 用語辞典 e-Words
- [2]高橋 麻奈, やさしいC 第2版
- [3]Sudoku 数独 数独パズル