

# 遺伝的アルゴリズムによる数独の解法

井上 はづき 佐藤 裕二

法政大学情報科学部

## Abstract

現在, 世界中で広く普及しているペンシルパズルのひとつである数独は, 大規模組合せ最適化問題のひとつと考えられる. 確率的探索手法のひとつである遺伝的アルゴリズム(GA)は, 組合せ最適化問題に強いと考えられており, 数独に GA を適用した研究例がいくつかある. しかし, 数独は GA の基本的操作のひとつである交叉が, 有益なビルディングブロックを非常に壊しやすいという特性を持っている. 本研究では, ビルディングブロックの破壊を避ける交叉の提案, また, GA が不得手なローカルサーチの性能を向上させることにより, GA 単独での数独解法の精度向上を試みる.

## 1. はじめに

現在の数独は 1979 年にアメリカ人建築家 Howard Garns が匿名で考案したペンシルパズルのひとつで, 2005 年にイギリスでブームになったのをきっかけに, 日本を含め世界中で流行している. 数独は大規模組合せ最適化問題に属すると考えられる. また, 従来手法では容易に解が求まらないような大規模かつ複雑な問題に対し, GA は比較的容易に最適解あるいは近似解を得られるために, 組合せ最適化問題に強いとされている. そのため, 数独に GA を適用した研究例はいくつかあり, 初期配置数の多い簡単な数独は GA で解けることが証明されているが, 最適解に至るまでのステップ数(世代数)が, 難解な問題では特に莫大になっている[1]. 原因は局所解が多数存在することと, GA のメインオペレーションである交叉が, 数独の場合では有効な解の一部(ビルディングブロック)を破壊しやすいことなどが挙げられる. 従来研究では GA と Grammatical Evolution[2]や Cultural Algorithm[3]を組み合わせることで, 問題に対処しているが, 本稿では, GA の適用方法を工夫することで, GA 単独での数独解法の精度向上を試みる.

## 2. 数独のルール

数独の問題は一般的に  $9 \times 9$  のマスから構成され,  $1 \sim 9$  の数字がいくつかのマスに予め与えられている. 出題の段階で与えられた数字を初期配置と呼び, その数字の数や配置によって難易度が変化する.

数独は全てのマスを  $1 \sim 9$  の数字で埋めることが解答となるが, その条件として, 縦・横の各列, 太枠で囲われた  $3 \times 3$  の小ブロック(リージョン)の全てにおいて,  $1 \sim 9$  の数字が重複していないことが条件となる. Fig. 1 に数独の問題・解答例を示す.

Solving and Analyzing Sudoku with Genetic Algorithms  
Hazuki Inoue, Yuji Sato  
Department of Computer Science,  
Faculty of Computer and Information Sciences,  
Hosei University.

	4			1		9		8
8		5				7		
							1	
	2			5				4
		1	6					
	3			8				2
							6	
3		4				8		
	8			9		4		3

6	4	3	5	1	7	9	2	8
8	1	5	3	2	9	7	4	6
2	9	7	8	6	4	3	1	5
9	2	8	1	7	5	6	3	4
4	7	1	6	3	2	5	8	9
5	3	6	9	4	8	1	7	2
7	5	9	4	8	3	2	6	1
3	6	4	2	5	1	8	9	7
1	8	2	7	9	6	4	5	3

Fig.1 数独の問題と解答例(初期配置数:24).

## 3. GA による数独解法の精度向上案

GA を数独に適用した従来例を見ると, 初期配置数の少ない問題の求解にかかる世代数が, 初期配置数が多い問題に対して莫大になっている[1]. 初期配置数が少ない, すなわち, 解の探索範囲が広い問題に対して, 一般に GA は有効であるとされるが, 今回の数独の問題では GA が上手く機能していない. その原因として, GA のメインオペレーションである交叉が有効な解の一部(ビルディングブロック)を破壊しやすい, GA はローカルサーチの性能が低い, 局所解に陥りやすいなどの問題が挙げられる. これらの問題点を改善するための工夫を, 以下で提案する.

### 3.1. ビルディングブロック問題への対処法の提案

#### 3.1.1. 初期設定

初期値は初期配置以外のマスをランダムな数字で埋めていく. その際, リージョン毎に数字が重複しないようにすることで, 数独の「全てのリージョン内で数字が重複しない」という条件を初めから満たし, 各操作をリージョン単位で行うことによって, 上記の条件を満たし続け, リージョン内でのビルディングブロックを崩され難くする.

#### 3.1.2. 突然変異

突然変異はリージョン毎に行う. リージョン内で初期配置以外の 2 つの数字をランダムに選び, その位置を交換する. それによってリージョン内で数字が重複することを避ける.

#### 3.1.3. 交叉

交叉では 2 つの親個体から 2 つの子個体を生成するが, 1 つは 3 行毎に評価値を出し比較し, 評価値の高い方を子個体に継承し, もう 1 つは同様に列毎で比較し, 子個体に継承する.

Fig. 2 に交叉を図で示す.

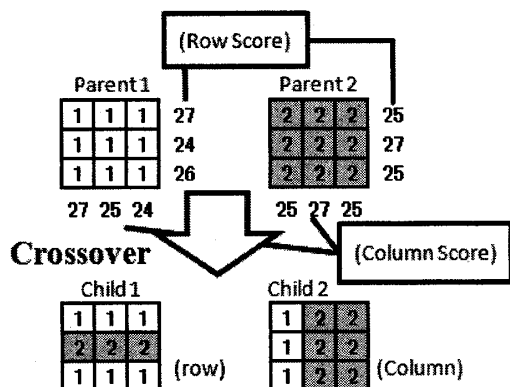


Fig. 2 交叉の説明図

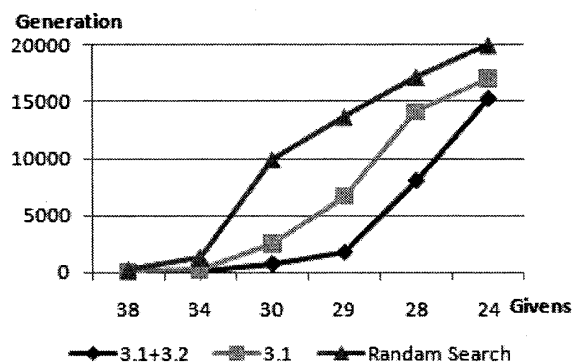


Fig. 3 初期配置数と世代数の関係

### 3. 2. 局所探索能力向上の提案

一般に GA はグローバルサーチに比べ、ローカルサーチの性能が低い。そのため、突然変異で生成する子供の数を、設定した個体数の 2 倍生成し、その中から評価値の高いものを個体数分選択することで、生成される子供のふり幅を大きくし、ローカルサーチの性能を向上させる。

## 4. 実験

### 4. 1. 実験方法

今回の手法では、選択はトーナメント選択 [4] を用いる。

また、評価関数は、「縦・横の各列で数字が重複しない」という条件を元に、各列の要素数を列の評価値とし、全ての列の加算を染色体の評価値とする (1)。各列に入る数字は 1~9 に限る (2) (3)。

$$f(x) = \sum_{i=1}^9 g_i(x) + \sum_{j=1}^9 g_j(x) \quad (1)$$

$$g_i(x) = |x_i| \quad (2)$$

$$g_j(x) = |x_j| \quad (3)$$

問題集 [5] から初級、中級、上級問題を 2 問ずつ、計 6 問に対して、50 回のシミュレーションを行った。

また、比較対象として、3.1 だけ適用したものと、ランダムサーチ (突然変異のみ) での実験も試みた。

#### 4. 1. 1. パラメータ

条件は以下の通りに設定し、実験を行った。

[個体数] 150, [打ち切り世代数] 20000, [交叉率] 0.3, [突然変異率] 0.3, [実行回数] 50

### 4. 2. 実験結果

実験結果のグラフを Fig. 3 に示す。Fig. 3 では、x 軸方向に初期配置数、y 軸方向に世代数を取る。

ランダムサーチと、3.1 を実装した結果を比べると、3.1 を実装した結果の方が早く収束しており、さらに 3.1 とそれにローカルサーチを加えたものとを比べると、ローカルサーチを加えた方が早く収束している。よって、それぞれの提案が効果をあげていることが分かる。

### 4. 3. 考察

初期配置数が同一の問題において、最適解が求まるまでにかかる世代数の平均値に大きな差が出る場合が観測される。その原因は初期配置場所に依存し、また、評価関数の定義が簡素なために配置に左右され易くなっていると考えられ、評価関数の設定方法により、改善できる可能性があると考えられる。

また、同一問題での解が求まるまでの世代数の差は、解の探索範囲が広大で、最適解からは遠いが評価値の高い局所解が多く存在し、局所解に陥りやすく、初期値依存性が高くなっているためと考えられる。そのため、一定の世代毎に突然変異率を高くするなど、局所解からの脱出方法を今後検討する必要がある。

GA を用いた従来手法との比較では、最上級問題であっても、半数が 2 万世代以内に最適解を導き出しており、打ち切り世代が 5 倍の 10 万世代で約 5% の回答率の従来例 [1] と比べ、飛躍的に精度が向上している。

## 5. まとめ

本稿では、GA を用いた数独解法の際に問題となる、ビルディングブロック破壊に対応するための GA のオペレーションを提案し、また、GA にローカルサーチの機能を加えることで、GA 単独での数独解法の精度向上を試みた。その結果、従来例では求解の難しかった問題においても、実行回数の半数で解を導き出すことが出来た。

## 文献

- [1] Timo Mantere and Janne Koljonen "Solving and Rating Sudoku Puzzles with Genetic Algorithms" New Developments in Artificial Intelligence and the Semantic Web Proceedings of the 12th Finnish Artificial Intelligence Conference STeP 2006
- [2] Miguel Nicolau and Conor Ryan "Genetic Operators and Sequencing in the GAuGE System" 2006 IEEE Congress on Evolutionary Computation
- [3] Timo Mantere and Janne Koljonen "Solving and Analyzing Sudokus with Cultural Algorithms" 2008 IEEE Congress on Evolutionary Computation
- [4] 井庭崇, 福原義久, "進化と遺伝的アルゴリズム," 複雑系入門, pp.102-117, (社) NTT 出版, 1998.
- [5] ナンプレベスト 110 選 15, ナンプレプラザ編集部 (編), (社) コスミック出版