

ニューラル手法と GA による新解の発見： ”Doubly Attacking Queens”

藤澤 公也[†] 武藤 佳恭^{††}

Recreational Mathematics の分野では Web 上で多くの未解決問題が提示され、活発に議論されている。本研究では Duisenberg's Puzzle において 2000 年 6 月 28 日に提示された未解決問題の一つである “Doubly Attacking Queens” 問題に対してニューラルネットワークをベースとした新しい手法を適用した。本手法はニューラルネットワークの組合せ最適化手法に遺伝的アルゴリズムの突然変異を組み合わせたものであり、この問題における新しい解を発見することに成功した。

A Neural+GA scheme for Doubly Attacking Queens problems

KIMIYA FUJISAWA[†] and YOSHIYASU TAKEFUJI^{††}

In the field of Recreational Mathematics, many unsolved problems have been actively shown on the Internet Web where some solutions have been given. The problems: “Doubly Attacking Queens” is one of the unsolved problems shown on the Web site of Ken Duisenberg's Puzzle on June 28, 2000. In this research, the combined optimizing method using an artificial neural network and a genetic algorithm is proposed and new solutions are discovered.

1. はじめに

Recreational Mathematics の分野では未解決問題が活発に Web 上で提示され、議論し、解決されている。これらの未解決問題に代表されるある種のパズル的な問題は軽視されることもあるが、その解法が工学的に転用される例も少なくない。これらの組合せ最適化問題の解決手法にはさまざまなアルゴリズムが提案されてきており、ニューラルネットワークの最適化手法によるものも数多く発表されている⁵⁾⁶⁾。

本研究では、パズル問題の一つである “Doubly Attacking Queens”¹⁾ を対象とし、より良い解を求めることを試みた。“Doubly Attacking Queens” は Ken's Puzzle of the Week²⁾ の中で 2000 年 6 月 28 日に提示され同年 10 月 28 日に部分解が出され、さらに 2001 年 2 月 26 日に改良解が提示されてきている。

ここでは、ニューラルネットワークを用いた組合せ最適化手法として新しいアルゴリズムを提案する。提案するアルゴリズムにおいては 2 つの工夫を行なっ

ている。すなわち提案するシステムにおいて、ニューラルネットワークの正規化⁴⁾ と遺伝的アルゴリズム⁷⁾ の一部である突然変異を用いている。最初にニューラルネットワークでの組合せ最適化に正規化を導入したのは 4 色問題解決においてである。

本研究では、ニューラルネットワーク手法を用いたときの解法を示しており、この問題におけるサイズ $N = 10, 11$ において改良された新しい解を発見した。

2. “Doubly Attacking Queens” 問題について

“Doubly Attacking Queens” は、 $N \times N$ の盤面にチェスのコマである Queen を配置していくもので、すべての Queen は効き筋に他の 2 つの Queen があるように配置し、この条件を満たしながら配置できる Queen の数の最大値はいくつかというものである。

ただし、このとき「Queen の効き筋」には 2 通りのルールがある。

• タイプ 1:

同じ方向に対しては最も近い Queen だけが効き筋にあるとみなす。横方向に 3 つのみの Queen がならんでいるとき、両端 2 つの Queen は効き筋に 1 つずつの Queen があるとみなし、真中の Queen は 2 つの Queen が効き筋にあるとみなす。

[†] 東京工科大学メディア学部
School of Media Science, Tokyo University of Technology
^{††} 慶応義塾大学 環境情報学部
Faculty of Environmental Information, Keio University

すなわち、Queen からの 8 方向のうち 2 方向だけに他の Queen があるように配置する。

- タイプ 2:

同じ方向に対してはすべての Queen が効き筋にあるとみなす。横方向に 3 つのみの Queen がならんでいるとき、どの Queen も効き筋には 2 つの Queen があるときとみなす。すなわち、Queen からの 8 方向すべての方向の終端までに合計で 2 つの Queen があるように配置する。

二つのタイプの違いは効き筋の範囲である(図 1)。

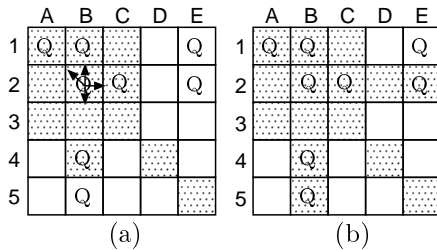


図 1 タイプ 1(図左 a) とタイプ 2(図右 b) の効き筋の違い。網かけ部分は各タイプの効き筋を示す。位置 B2 にある Queen の効き筋上には、タイプ 1 では 8 方向の効き筋のうち 4 方向に Queen があるが、タイプ 2 では 8 方向の領域内に 6 つの Queen がある。

Fig. 1 The difference in the effective area of Type 1(A) and Type 2(B). The mesh shows the effective area of each type. The effective area of the Queen which is on B2 includes 4 Queens at type 1 and 6 Queens at type 2.

3. ニューラルネットワークでの解法

$N \times N$ の盤面の各マスが一つのニューロンを示し、各マスに対して効き筋上にある他のニューロンと接続されたネットワークを用いる。ここで、タイプ 1 の場合には、Queen の配置状況により効き筋の範囲が変化することに注意する必要がある。つまり、タイプ 1 の場合、Queen の配置状況によって、ネットワークのトポロジが変化することを意味する。

ニューロンモデルとして、binary モデルの拡張型である hysteresis binary モデル⁸⁾ を利用する。

3.1 最適化における動作方程式

各ニューロンは、内部値 U_{ij} をもっており、接続されている他のニューロンからの情報を元に U_{ij} の値を変化させる。はじめ、各 U_{ij} は初期値としてランダムな値を所持しており、以下の動作方程式³⁾ にしたがって値を更新する。動作方程式は U_{ij} に関する差分方程式になっており、この差分方程式と 1 次のオイラー

法によって、 U_{ij} を更新する。この内部値が hysteresis binary 関数によって定められる一定値を超えると、ニューロンが発火する。すなわち、そこに Queen が配置されることを意味し、他のニューロンへの出力値 V_{ij} が 1 になる。

$$\frac{dU_{ij}}{dt} = -A \left(\sum d(ij, pq) V_{pq} - 2 \right) \quad (1)$$

ここで、 A は定数である。また、関数 $d(ij, pq)$ はネットワークトポロジを決定する関数であり、ニューロン pq がニューロン ij の効き筋上にある場合は 1 を効き筋上にない場合には 0 となる。関数 $d(ij, pq)$ は 2 種類のルール、タイプ 1 とタイプ 2 では異なる結果を返すことになる。タイプ 2 の場合、ある 2 つのニューロンの関係は一定であり、常に決まった値を返すが、タイプ 1 の場合、タイプ 2 では効き筋の関係にある 2 つのニューロン間であっても、それらのニューロン間に別の発火しているニューロンがある場合には 0 を返す。図 1 において、ニューロン B2 およびニューロン D2 について考えるとタイプ 1 の場合は関数 $d(B2, D2)$ は 0 を返し、タイプ 2 の場合は関数 $d(B2, D2)$ は 1 を返す。

3.2 動作方程式の正規化

ニューロン i の効き筋で発火しているニューロンだけを条件に動作方程式を立てると容易にローカルミニマムに陥る(図 2(A)(B))。ローカルミニマムの原因の一つには局所的に状態が安定し過ぎ、他のニューロンからの発火/抑制すべき情報が正しく反映されないことにある。このような場合、特定の状態に固定化される場合と、複数の状態間で反復的な変化を繰り返す場合がある。このような状態に陥りにくくするために各ニューロンの発火情報をより遠くのニューロンまで効果的に伝えるために正規化を用いる。

式 (1) では、すべてのニューロンがそのニューロンの効き筋上に 2 つの発火したニューロンが存在する状態に収束するように動作する。これを正規化するために式 (1) のように変更する。正規化によって、ニューロン ij の効き筋上にある発火しているニューロンが影響を受けるニューロンの数の平均が 2 に収束するようになっている。

$$\frac{dU_{ij}}{dt} = -A \left(\sum d(ij, pq) V_{pq} - 2 \right) - B \left(\frac{\sum d(ij, pq) \sum d(pq, st) V_{st}}{\sum d(ij, pq) V_{pq}} - 2 \right) \quad (2)$$

オイラー法での U_{ij} の更新は、エネルギー関数

$$E = \sum U_{ij} \quad (3)$$

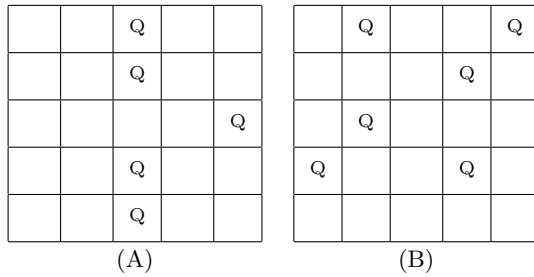


図 2 5 × 5 におけるローカルミニマム解 (1) . (A) はもっとも単純なもの . (B) は正規化を用いた場合でも起こり得るもの
Fig. 2 A local minimum state for N=5 (1). fig.(A) shows a simple local minimum state, and fig.(B) shows another local minimum state which is occurred with normalizing method

が最小の安定状態になったときに終了する . 一般に単純なニューラルネットワークでは

$$\sum \frac{dU_{ij}}{dt} = 0 \quad (4)$$

となったときに E は最小値の安定状態となるが , 動作方程式 (1) を用いた場合 , $\frac{dU_{ij}}{dt} < 0$ は保証されない . 安定状態となるのは ,

$$\sum \left| \frac{dU_{ij}}{dt} \right| = 0 \quad (5)$$

となるときであるので , この状態になった段階で更新を終了する .

3.3 ローカルミニマムからの脱出 (1) : ヒルクライミング項

正規化を行なっている場合 , ローカルミニマムに陥ることがある .

これは式 (2) として , すなわちルールとしての条件は満たしていることを示しているが , 配置している Queen の数が最大でない場合に起こる . 図 2(B) はこの状態を示している . 制約条件的を満たしているが配置している Queen の数は最大ではない状態で収束している .

このローカルミニマム状態から脱出する一つの方法として , ヒルクライミング項³⁾ を利用する方法がある . ヒルクライミング項は , 式 (2) の状態では発火する条件にないニューロンのうち , 発火する条件に近いものを強制的に発火させ , ローカルミニマムの状態を崩しより最適解に近付ける動きをする .

ここではヒルクライミング条件として , ニューロン ij が発火していない状態で , かつ , ニューロン ij の効き筋にある発火中のニューロンの数が 2 個のニューロンが発火しやすくなるように , 動作方程式 (2) にヒルクライミング項

$$+Ch(V_{ij})h\left(\sum d(ij,pq)V_{pq}-2\right) \quad (6)$$

を加える . ここで , C は定数であり , 関数 $h(x)$ はヒルクライミング関数である .

$$h(x) = \begin{cases} 1(x=0) \\ 0(\text{otherwise}) \end{cases} \quad (7)$$

x が 0 のとき 1 を返し , そうでないとき 0 を返す .

3.4 ローカルミニマムからの脱出 (2) : 突然変異の導入

ヒルクライミング項を利用した場合でもエネルギー関数 (3) が収束しローカルミニマムに陥ることがある . 発火していない各ニューロンの効き筋にある発火中のニューロンの数がいずれも 2 よりも大きい時にローカルミニマムに陥る . 図 3 はタイプ 1 の 5 × 5 におけるローカルミニマムを示している . 最適解ではタイプ 1 の 5 × 5 の場合 , 8 個の Queen が配置可能であるが , ここでは 7 個の Queen にとどまっている .

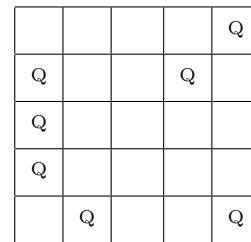


図 3 5 × 5 におけるローカルミニマム解 (2)
Fig. 3 A local minimum state for N=5 (2)

このようなローカルミニマムから抜け出す方法として , 本研究では遺伝的アルゴリズムにある突然変異の導入を提案する . 遺伝的アルゴリズムにおける突然変異は , 優良な解候補に局所的な変更を加えたものを解候補に加えることで解の多様性を高めるために利用されている .

本研究では , ローカルミニマムに陥った状態からの突然変異として , 発火しているニューロンの 1 つをランダムに選択し , 発火を抑制した状態に変化させている . この状態から動作方程式およびオイラー法によるシミュレーションを継続することにより , より発火に適したニューロンが存在すると , そちらが先に発火し新たな解に収束することになる .

4. 結 果

本研究で提案する手法を用いることで , "Doubly Attacking Queens" 問題の解が知られているすべての N

について既知の解と同数以上の Queen を配置することができた。特に、ルール：タイプ 1 の N=10, 11 において改良された 2 つの解を得ることができた(図 4, 図 5)。公開されている部分解および提案した手法による解を表 1 に示す。

N	タイプ 1		タイプ 2	
	(A)	(B)	(A)	(B)
2	-	3	3	3
3	-	4	4	4
4	6	6	6	6
5	8	8	7	7
6	10	10	9	9
7	12	12	11	11
8	14	14	13	13
9	16	16	14	14
10	17	18	-	16
11	19	20	-	18
12	21	21	-	20
13	-	22	-	21

表 1 解の比較：N は盤面のサイズ。各タイプ左側 (A) の数字が既知解での Queen の最大数、右側 (B) の数字が提案する手法での Queen の最大数。“-” は既知解なしを示す。

Table 1 Solutions: N is size of board. For each type, the left hand values (A) is the number of queens of known solutions and right hand values (B) is solutions of new algorithm. “-” means no solution

既存の解では、N=10 においては Queen を 17 個、また N=11 においては Queen を 19 個が最大とされてきたが、本研究で提案する手法を用いることにより N=10 で 18 個、N=11 で 20 個の Queen を配置した新しい解を得ている。

		Q		Q		Q			Q
		Q							
									Q
Q									
									Q
Q									
									Q
Q									
Q									
	Q		Q		Q		Q	Q	Q

図 4 N=10 の新しい解：18 個の Queen を配置
Fig. 4 New solution of N=10: 18 Queens

	Q								
		Q		Q	Q			Q	
									Q
	Q								
									Q
									Q
	Q								
	Q								
									Q
	Q								
Q			Q			Q	Q		Q

図 5 N=11 の新しい解：20 個の Queen を配置
Fig. 5 New solution of N=11: 20 Queens

参考文献

- 1) Ken Duisenberg: “Doubly Attacking Queens”, <http://ken.duisenberg.com/potw/archive/arch00/000628.html>
- 2) Ken Duisenberg: “Ken’s Puzzle of the Week”, <http://ken.duisenberg.com/potw/>
- 3) Yoshiyasu Takefuji: “Neural Network Parallel Computing”, Kluwer Publishers(1992).
- 4) Y. Takefuji, and K. C. Lee: “Artificial neural networks for four-coloring problems and k-colorability problems”, IEEE Trans. on Circuits and Systems, Vol. 38, No. 3, pp. 326–333 (1991).
- 5) Yoshiyasu Takefuji: “Crossbar switch scheduling problems”, Neural Network Parallel Computing, Kluwer Publishers, pp. 27–36 (1992).
- 6) CELSO C. RIBEIRO, MAURICIO G. C. RESENDE: “Algorithm 797: Fortran Subroutines for Approximate Solution of Graph Planarization Problems Using GRASP”, ACM Transactions on Mathematical Software, Vol. 25, No. 3, pp. 341–352(1999)
- 7) D. E. Goldberg, “Genetic Algorithms in Search, Optimization, and Machine Learning”, Addison-Wesley(1989).
- 8) Y. Takefuji, and K. C. Lee: “An artificial hysteresis binary neuron: A model suppressing the oscillatory behaviors of neural dynamics”, Biological Cybernetics, 64, pp. 353–356 (1991).