5B−6

N-クイーン問題の解の1つを線形時間で求めるアルゴリズムについて

秋葉 澄孝
電子技術総合研究所

1 はじめに

N-クイーン問題は19世紀に提案された問題であり、現在ではバックトラックや制約充足アルゴリズムなどの評価に利用されている。しかし、この問題に関して知られている事実は個々のNに関するものが多く、一般のNに対して成り立つ事実はあまり知られていない。

本報告では、Nが奇数の場合にN-クイーン問題の点対称な解を求めるアルゴリズムについて述べる。

2 N-クイーン問題

N-クイーン問題とは、「N行N列の『チェス盤』の上にN個のクイーンの駒を、どのクイーンも互いに他のクイーンの利き筋に入らないように配置する」問題である。N≤3の場合にはそのような配置は不可能なので、以下ではN≥4の場合について議論する。

この問題は数学的な興味から、解の個数、点対称な解の有無などが調べられてきた[5]。その結果、比較的小さい個々のNに関する事実がわかっている。しかし、一般のNに対して成り立つ事実は自明なもの以外ほとんどわかっていない。

工学的な興味からは、この問題をバックトラックや制約充足アルゴリズムの評価のための標準的なベンチマークテストとして用いている[3]。この場合、すべての解を求めるために必要な計算量や時間を用いて、一般的な問題を対象としたアルゴリズムを評価するのが普通である。しかし、すべての解を求められるとは限らないが、解の1つを求めるのに有効な手法を取り入れたN-クイーン問題専用のアルゴリズムも研究されている[4]。

一般のNに対して成り立つN-クイーン問題に関する事実の1つに、Nが偶数の場合に点対称な解の1つを求める方法と、Nが奇数の場合に（点対称ではない）解の1つを求める方法がある[1]。この2つの方法は「パターン」として表現されており、解の1つを$O(N)$の計算量で求めるアルゴリズムを与えているとみなせる。この事実は1950年代前半に発表されたが、あまり知られていない。

最近発表された事実として、N=24m−3の場合に解の1つを求める方法と、N=24m+9の場合に解の1つを求める方法[2]がある。この2つも「パターン」として表現されており、解の1つを$O(N)$の計算量で求めるアルゴリズムを与えているとみなせる。実は、これらの方法で得られる解は点対称な解である。しかし、文献[2]ではこの事実には触れていない。また、これらの方法だけではすべての奇数に対して点対称な解を求めることはできない。

さらに本質的には、文献[2]の2つの方法でそれぞれN=24m±3とN=24m±9の場合に点対象な解の1つを求めることができる。しかし、文献[2]で採用された定式化のままではN=24m−3とN=24m+9の場合にしか適用できない。本報告で述べるN=24m±3とN=24m±9の場合のアルゴリズムは、文献[2]の定式化を改良したものとみなすことができる。

次の章では、奇数Nを3つの場合に分け、それぞれのNに対してN-クイーン問題の点対称な解の1つを$O(N)$の計算量で求めるアルゴリズムついて述べる。

3 Nが奇数の場合のアルゴリズム

本章では、奇数NをN=6m±1 (5,7,11,13,17,19,23,25,...)、N=24m±3 (21,27,...)、N=24m±9 (9,15,33,...)の3つの場合にわけ、それぞれのNに対してN-クイーン問題の解の1つを$O(N)$の計算量で求めるアルゴリズムについて述べる。アルゴリズムはC言語に近い形で記述する。

アルゴリズム中では盤上の位置を表すために、左下の位置を原点とする座標系(x,y)を用いる。つまり、x座標は横方向の位置を、y座標は縦方向の位置を表し、x=0,...,N−1、y=0,...,N−1である。また、(N−1,0)は右下、(N−1,N−1)は右上、(0,N−1)は左上となる。

N=6m±1の場合は図1のアルゴリズムで解を求められる。図2はこのアルゴリズムで求めたN=11の場合の解である。この図からもわかるように、図1のアルゴリズムはクイーンを盤の中央、左上、右下の3つの部分に、それぞれ始点を盤の中心、左辺の中心の1つ上、右辺の中心の1つ下にとり桂馬飛びに置いていくアルゴリズムである。

このアルゴリズムでクイーンを配置した場合、点対称であることと、縦方向、横方向および右あがり方向にクイーンが重ならないことはほとんど自明である。実は、この4つの性質はすべての奇数Nに対して成り立つことが簡単な計算で確かめられる。N=6m±1の場合に右

```
put-queen-on((N−1)/2, (N−1)/2);
for(x=(N−1)/2+1, y=(N−1)/2+2; y<N; x+=1, y+=2)
        put-queen-on(x,y);
for(x=(N−1)/2−1, y=(N−1)/2−2; y≥0; x−=1, y−=2)
        put-queen-on(x,y);
for(x=0, y=(N−1)/2+1; y<N; x+=1, y+=2)
        put-queen-on(x,y);
for(x=N−1, y=(N−1)/2−1; y≥0; x−=1, y−=2)
        put-queen-on(x,y);
```

図1: N=6m±1の場合のアルゴリズム

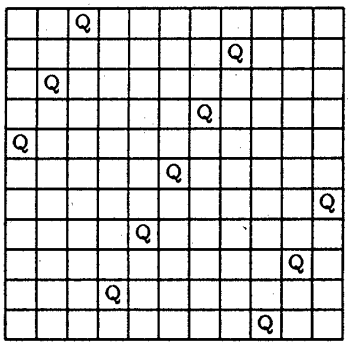

図2: N=11の場合の解

Linear-Time Algorithms for the N-Queens Problem
Sumitaka AKIBA
ELECTROTECHNICAL LABORATORY

さがり方向に重ならないことも簡単な計算で確かめられる。しかも、同じ計算でN=6m+3の場合には重なることが確かめられるので、この場合には図1のアルゴリズムを用いることができない。また、このアルゴリズムの計算量が$O(N)$であることもほとんど明らかである。

N=6m+3の場合には図1のアルゴリズムよりも少し複雑なアルゴリズムで解を求めることができる。このアルゴリズムは本来1つのものとみなすことができる。しかし、単純にするためにN=24m±3とN=24m±9の場合に分けて図3と図4に示した。図5と図6はこのアルゴリズムで求めたN=21およびN=15の場合の解である。

```
put-queen-on((N−1)/2,(N−1)/2);
for(x=(N−1)/2+1, y=(N−1)/2+2; y<N; x+=1, y+=2)
        put-queen-on(x,y);
for(x=(N−1)/2−1, y=(N−1)/2−2; y≥0; x−=1, y−=2)
        put-queen-on(x,y);
for(x=0, y=(N−1)/2−1; y<N; x+=2, y+=4)
        put-queen-on(x,y);
for(x=1, y=(N−1)/2+5; y<N; x+=2; y+=4)
        put-queen-on(x,y);
for(x=N−1, y=(N−1)/2+1; y≥0; x−=2, y−=4)
        put-queen-on(x,y);
for(x=N−2, y=(N−1)/2−5; y≥0; x−=2, y−=4)
        put-queen-on(x,y);
```

図 3: N=24m±3の場合のアルゴリズム

```
put-queen-on((N−1)/2,(N−1)/2);
for(x=(N−1)/2+1, y=(N−1)/2+2; y<N; x+=1, y+=2)
        put-queen-on(x,y);
for(x=(N−1)/2−1, y=(N−1)/2−2; y≥0; x−=1, y−=2)
        put-queen-on(x,y);
for(x=0, y=(N−1)/2+3; y<N; x+=2, y+=4)
        put-queen-on(x,y);
for(x=1, y=(N−1)/2+1; y<N; x+=2, y+=4)
        put-queen-on(x,y);
for(x=N−1, y=(N−1)/2−3; y≥0; x−=2, y−=4)
        put-queen-on(x,y);
for(x=N−2, y=(N−1)/2−1; y≥0; x−=2, y−=4)
        put-queen-on(x,y);
```

図 4: N=24m±9の場合のアルゴリズム

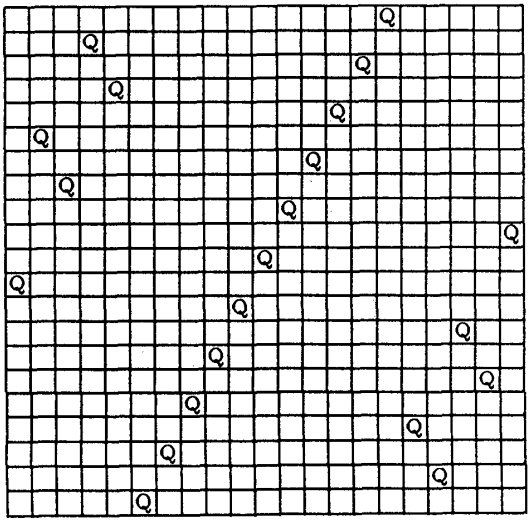

図 5: N=21の場合の解

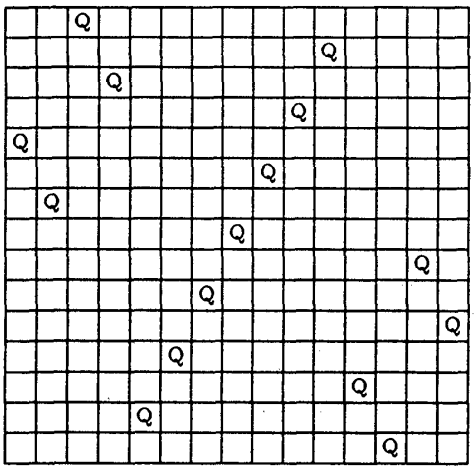

図 6: N=15の場合の解

これらのアルゴリズムで得られるクイーンの配置も3つの部分に分けることができる。中央の部分はN=6m±1の場合と同様に盤の中心から桂馬飛びに置いて作られるが、左上と右下の部分は少し違っている。そこは1つおきの桂馬飛びに2列置いて作られる。

また、図3と図4のアルゴリズムの違いは、これらの列の始点である。つまり、N=24m±3の場合の始点は、左端の行の中心の1つ下、左から2行目の中心の5つ上、右端の行の中心の1つ上、右から2行目の中心の5つ下の4か所だが、N=24m±9の場合は、左端の行の中心の3つ上、左から2行目の中心の1つ上、右端の行の中心の3つ下、右から2行目の中心の1つ下の4か所が始点である。

図1のアルゴリズムと同様に、図3と図4のアルゴリズムが正しいことは簡単な計算で確認できる。また、これらのアルゴリズムの計算量が$O(N)$であることもほとんど明らかである。

4 おわりに

本報告では、Nが奇数の場合にN-クイーン問題の点対称な解の1つを$O(N)$の計算量で求めるアルゴリズムについて述べた。これとNが偶数の場合に関する従来の研究から、一般のN(≥4)に対してN-クイーン問題には点対称な解が存在することがわかった。しかも、その1つをNに比例する時間内に求められるようになった。

参考文献

[1] Abramson, B. and Yung, M.: Divide and Conquer under Global Constraints: A Solution to the N-Queens Problem, *Journal of Parallel and Distributed Computing* 6, pp.649–662, 1989.

[2] Falkowski, B. and Schmitz, L.: A Note on the Queens' Problem, *Infomation Processing Letters* **23**, pp.39–46, 1986.

[3] Minton, S., Johnston, M.D., Philips, A.B. and Laird, P.: Solving Large-Scale Constraint Satisfaction and Scheduling Problems Using a Heuristic Repair Method, *Proc. of AAAI-90*, pp.17–24, 1990.

[4] Sosic, R. and Gu, J.: A Polynomial Time Algorithm for the N-Queens Problem, *SIGART Bulletin*, Vol.1, No.3, pp.7–11, 1990.

[5] 山本 幸一, 8個のクイーンの問題, 数学セミナー増刊「数学100の問題」, 日本評論社, pp.44–46, 1984.