Explicit Solutions to the N-Queens Problem for all N

Bo Bernhardsson
Department of Automatic Control
Land Institute of Technology
P.O. Box 118
S-22100 Lund SWEDEN
bob@control.lth.se

The n-queens problem is often used as a benchmark problem for AI research and in combinatorial optimization. An example is the recent article [1] in this magazine that presented a polynomial time algorithm for finding a solution. Several CPU-hours were spent finding solutions for some n up to 500,000.

I would like to draw the readers' attention to the less known fact that *explicit* solutions for this problem exist for all $n \geq 4$, see below. This result was published 1969 in [2]. The construction is quite simple and requires no combinatorial search or computer time whatsoever. To find one solution to the n-queens problem is therefore completely trivial. The n-Queens problem is therefore a bad benchmark problem. A lot of CPU-hours can be saved in the future noting this.

A verification of the construction below is given in [2]. Let (i, j) denote the square on row i and column j. Depending on n there are three cases:

(A) n even but not of the form $6k+2$: Place queens on elements

$$(j, 2j)$$
$$(n/2+j, 2j-1), \qquad j = 1, 2, \ldots, n/2$$

(B) n even but not of the form $6k$: Place queens on elements

$$(j, 1+[2(j-1)+n/2-1 \text{ modulo } n])$$
$$(n+1-j, n-[2(j-1)+n/2-1 \text{ modulo } n]),$$
$$j = 1, 2, \ldots n/2$$

(C) n odd: Use case A or case B on $n-1$ and extend with a queen at

$$(n, n)$$

It could finally be noted that the related problem of finding *all* solutions to the n-queens problem is non-trivial. It was introduced by Gauss with $n = 8$.

References

[1] Sosic, R. and Gu, J. (1990): A Polynomial Time Algorithm for the N-Queens Problem, *SIGART Bulletin*, 1, p. 7-11.

[2] Hoffman, E.J., Loessi, J.C. and Moore, R.C. (1969): Constructions for the Solution of the m Queens Problem, *Mathematics Magazine*, p. 66-72.