This is a repository copy of *Complexity of n-Queens completion (extended abstract)*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/141515/

Version: Accepted Version

**Proceedings Paper:**
Gent, Ian Philip, Jefferson, Christopher Anthony and Nightingale, Peter William (2018)
Complexity of n-Queens completion (extended abstract). In: Lang, Jérôme, (ed.)
Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence. International Joint Conferences on Artificial Intelligence, pp. 5608-5611.

10.24963/ijcai.2018/794

**Reuse**
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

**Takedown**
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.
Complexity of \(n\)-Queens Completion (Extended Abstract)*

Ian P. Gent, Christopher Jefferson, Peter Nightingale
School of Computer Science, University of St Andrews, Fife KY16 9SX, UK
ian.gent@st-andrews.ac.uk, caj21@st-andrews.ac.uk, pwn1@st-andrews.ac.uk

Abstract

The \(n\)-Queens problem is to place \(n\) chess queens on an \(n\) by \(n\) chessboard so that no two queens are on the same row, column or diagonal. The \(n\)-Queens Completion problem is a variant, dating to 1850, in which some queens are already placed and the solver is asked to place the rest, if possible. We show that \(n\)-Queens Completion is both NP-Complete and \#P-Complete. A corollary is that any non-attacking arrangement of queens can be included as a part of a solution to a larger \(n\)-Queens problem. We introduce generators of random instances for \(n\)-Queens Completion and the closely related Blocked \(n\)-Queens and Excluded Diagonals Problem. We describe three solvers for these problems, and empirically analyse the hardness of randomly generated instances. The \(n\)-Queens Completion problem may be one of the simplest NP-Complete problems to explain to people who understand the rules of chess. The problem is “Given an \(n \times n\) chessboard on which some queens are already placed, can you place a queen in every remaining row so that no two queens attack each other?”

We will show that the \(n\)-Queens Completion problem is NP-Complete and \#P-Complete, discuss solvers for the problem, and empirically analyse randomly generated instances. The \(n\)-Queens Completion problem may be one of the simplest NP-Complete problems to explain to people who understand the rules of chess. The problem is “Given an \(n \times n\) chessboard on which some queens are already placed, can you place a queen in every remaining row so that no two queens attack each other?”

1.1 History of \(n\)-Queens

The \(n\)-Queens problem has an extraordinary history for such an apparently unassuming problem, both generally and inside Artificial Intelligence. Formerly, and incorrectly, attributed to Gauss, the problem’s history was clarified by Campbell [1977]. The 8-Queens problem was introduced by Bezzel [1848] and by Nauck [1850] (possibly independently). The latter publication attracted the interest of Gauss, who even made a small mistake in studying the problem.\(^1\) The generalisation to \(n\)-Queens has attracted the interest of many

\(^*\)This paper is an extended abstract of an article in the Journal of Artificial Intelligence Research [Gent et al., 2017].

\(^1\)He reported finding 76 solutions but later realised that four of those were erroneous so had only 72.
other mathematicians, whose results are surveyed by Bell and Stevens [2009]. The first paper to describe backtracking search on computer, presented in 1958, does so for the n-Queens problem [Walker, 1960]. Since then it has been used as an example and/or benchmark problem in many classic AI papers, for example at least six with more than 400 citations on Google Scholar at time of writing [Golomb and Baumert, 1965; Bitner and Reingold, 1975; Mackworth and Freuder, 1985; Minton et al., 1992; Selman et al., 1992; Crawford et al., 1996].

1.2 Complexity of n-Queens

The complexity of the n-Queens problem is often misunderstood. The decision problem is solvable in constant time since there is a solution for all $n > 3$ so is only NP-hard if P=NP. A witnessing solution can be constructed easily [Bell and Stevens, 2009] but note that the witness (a set of n queens) requires $n \log n$ bits to specify but this is not polynomial in the size of the input, which is only $\log n$ bits. The n-Queens problem has often been incorrectly called NP-hard, even in well-cited papers [Mandziuk, 1995; Martinjak and Golub, 2007; Shah-Hosseini, 2009; Nakaguchi et al., 1999, each with at least 29 citations]. The counting version of the problem, i.e. to determine how many solutions to n-Queens there are, is sequence A000170 of the Online Encyclopedia of Integer Sequences [Sloane, 2016]. The sequence is currently known only to $n = 27$, for which the number of solutions is more than $2.34 \times 10^{17}$. No approach better than optimised exhaustive search seems to be known: e.g. the $n = 27$ total was counted using a massively parallel search using FPGAs [Preußer, 2016]. Hsiang et al. [2004] show that solving the n-Queens counting problem is “beyond the #P-class”. Bell and Stevens [2009] states that this means that there is no closed form expression in n for the number of solutions, but Chaiken et al. [2015] claim to give one.²

1.3 Generalising to n-Queens Completion

Cadoli and Schaerf [2006] studied the n-Queens Completion problem without studying its decision or counting complexity. The most closely related work to ours is by Martin [2007], who proves a rather different generalisation of n-Queens to be NP-complete, the key difference being that some squares can be marked as stopping attacks. In particular, this means that solutions to Martin’s problem need not be solutions to the n-Queens problem, unlike n-Queens Completion.

We are contributing to a rich literature on the complexity of puzzles and games. One of the earliest results in the area is that generalised chess is EXPTIME-complete [Fraenkel and Lichtenstein, 1981]. Amongst other games to have been proved NP-complete or harder are the card solitaire games Klondike [Longpré and McKenzie, 2009], Freecell [Helmer, 2003] and Black Hole [Gent et al., 2007], the Sudoku puzzle [Takayuki and Takahiro, 2003], video games like Pac-Man [Viglietta, 2014], and casual games such as Minesweeper.

²It is unclear to us if this is a mathematical dispute or simply a dispute on what it means to be a closed form expression, but in any case Chaiken et al.’s formula has not been used to extend knowledge of the number of solutions beyond $n = 27$.

![Figure 2: The two possible solutions of the 8-Queens Completion instance from Figure 1.](image)

[Kaye, 2000], Candy Crush Saga and Bejeweled [Walsh, 2014; Guala et al., 2014]. Many other games are surveyed by Kendall et al. [2008] and by Demaine and Hearn [2009]. In each case, the complexity-theorist must define a generalised class of instances. Unlike some of the examples above, this step is completely natural for n-Queens Completion.

1.4 Controversy Surrounding n-Queens for Benchmarking

Because of the ease of finding a solution, the n-Queens problem has been subject to repeated controversy in AI over whether it should be used as a benchmark at all. For example a sequence of papers argued the point in the pages of SIGART Bulletin in the early 1990s [Sosic and Gu, 1990; Johnson, 1991; Bernhardsson, 1991; Gu, 1991; Valtorta, 1991], and then in 2014 the issue was raised again in a blog post by Smet [2014]. We resolve this issue in the sense that, as an NP- and #P-Complete problem, n-Queens Completion does provide a valid benchmark problem. Similarly, the Quasigroup Completion Problem (which is to complete a partially filled latin square) is NP-Complete [Colbourn, 1984] and is a challenging and popular benchmark [Gomes and Selman, 1997], whereas constructing a latin square from scratch is trivial. A very closely related problem, “Blocked n-Queens”, has previously been used for benchmarking without complexity guarantees [Namisivayam and Truszczynski, 2009]. Our results show as a corollary that Blocked n-Queens is NP-Complete and #P-Complete. We explore the practical difficulty of these problems, and a new variant, the Excluded
Diagonals Problem. For Blocked $n$-Queens and the Excluded Diagonals Problem, we show the existence of a phase transition associated with hard instances, but we were not able to generate consistently hard instances for the $n$-Queens Completion problem.

1.5 Theoretical Results

In the journal paper [Gent et al., 2017] we give definitions of a sequence of problems, where the first problem is $n$-Queens Completion, and the last is a variant of Boolean Satisfiability (SAT) that is NP- and #P-Complete, with the Excluded Diagonals Problem at an intermediate stage. The proof proceeds by a sequence of polynomial reductions, starting with the last problem and ending with $n$-Queens Completion.

Acknowledgements

We thank Jeff Erickson for raising the question of the complexity of (a slight variant of) $n$-Queens Completion [Erickson, 2014] and Stanislav Živný for bringing his post to our attention by a Facebook post on 1 January 2015. We thank Torsten Schaub, Martin Gebser, Bilal Hussain, and András Salamon. We would like to thank the EPSRC for partially funding this work through grants EP/K015745/1, EP/M003728/1, and EP/P015638/1. In addition, Dr Jefferson is funded by a Royal Society University Research Fellowship.

References


