Complete and Incomplete Algorithms for the Queen Graph Coloring Problem

Michel Vasquez and Djamal Habet[1]

Abstract. The queen graph coloring problem consists in covering a $n \times n$ chessboard with n^2 queens, so that two queens of the same color cannot attack each other. When the size, n, of the chessboard is a multiple of 2 or 3, it is hard to color the queen graph with only n colors. We have developed an exact algorithm which is able to solve exhaustively this problem for dimensions up to $n = 12$ and find one solution for $n = 14$ in one week of computing time. The 454 solutions of Queens_12^2 show horizontal and vertical symmetries in the color repartition on the chessboard. From this observation, we design a new exact, but incomplete, algorithm which leads us to color Queens_n^2 problems with n colors for $n = 15, 16, 18, 20, 21, 22, 24, 28$ and 32 in less than 24 hours of computing time by the exploitation of symmetries and other geometric properties.

1 INTRODUCTION

Given a $n \times n$ chessboard, a queen graph is a graph with n^2 vertices, each of them corresponding to a square of the board. Two vertices are connected by an edge if the corresponding squares are in the same row, column, or diagonals. This corresponds to the rules for moving the queen in a chess game.

In this paper, we consider the following problem: are n colors sufficient to place n^2 queens on the chessboard so that there is no clash between two queens of same color? If so, n is the chromatic number of Queens_n^2 *–noted* χ_n– since the maximum clique number is n. Indeed, the rows, the columns and the 2 main diagonals constitute the $2n + 2$ maximum cliques of this graph.

A first exact algorithm, working on independent sets, solves the queen graph problems for n up to 14. We have obtained completeness for $n = 10$ (*no solution*) and $n = 12$ (454 *solutions*), but only one solution for Queens_14^2 in 168 hours of computing time. When examining Queens_12^2 solutions, we find symmetries, both vertical and horizontal, between vertices belonging to different independent sets. That means that we could generate vertices by pairs rather than one by one. This observation leads us to consider these symmetries, and some other geometric operations, in order to reduce the size of the search space. This approach is no more complete but we expect to solve larger instances under the hypothesis that such geometric properties exist.

The paper is structured as follows. After a brief overview of the literature, we describe the main characteristics of our search tree algorithm based on independent sets. Then we present the four kinds of geometric operations implemented to improve our results. Finally, we summarize the computing time needed to color Queens_n^2 up to $n = 32$, before concluding the paper.

[1] École des Mines d'Alès–Commissariat à l'Énergie Atomique, site EERIE, Parc Scientifique Georges Besse, 30035 Nîmes cedex 1, France. email: {Michel.Vasquez, Djamal.Habet}@ema.fr

2 BRIEF OVERVIEW

Although the graph coloring problem has been the subject of intense research, applications on the Queens_n^2 problem have received a limited attention: Mehrotra and Trick [8] use a column generation approach to the independent set formulation of the graph coloring problem, devising an efficient algorithm to solve the maximum weighted independent set problem arising in the column generation process, and are able to solve problems up to $n = 9$. Caramia and Dell'Olmo [1] suggest a sophisticated algorithm based on the iterative coloring extension of a maximum clique. Extensive computational results are given where Queens_n^2 problems are solved up to $n = 9$.

On the other hand, heuristic methods are also used to treat the Queens_n^2 problem. For instance, Kochenberger, Glover et al. [6] transform this problem into an unconstrained quadratic binary problem, and solve it by the tabu search method. Other recent works using local search and solution combining are designed to tackle Queens_n^2 problems up to $n = 16$ [3, 5]. However, non-exact methods fail to prove that $\chi_n = n$ and only give an upper bound for the chromatic number.

3 NESTED ENUMERATIONS

Our algorithm to solve the Queens_n^2 problem is based on the enumeration of the independent sets (IS) of the queen graph. An IS is defined as a subset of vertices which are not linked by an edge. Hence, all the vertices of the same IS can have the same color.

The n squares belonging to the first row of the chessboard are definitively colored with n different colors and do not take part in the combinatoric. Indeed, these squares correspond to a maximum clique of the graph. This classical technique has already been exploited in [7] and avoids to explore many symmetric configurations.

Then, the algorithm iteratively constructs independent sets while checking that there are no two queens on the same square. It is based on the following statements.

3.1 IS Enumeration and Assignment

No IS can contain more than n vertices because there are n disjoint cliques of size equal to n (for instance, the n rows of the chessboard). Accordingly, coloring n^2 vertices with n colors leads to find n disjoint independent sets $\mathcal{I}_1 \cdots \mathcal{I}_n$ with exactly n vertices each, among a set IS_n of the IS candidates with n vertices. IS_n is generated by a standard depth-first search.

This enumeration procedure is embedded in a branching algorithm which tries to assign up to n IS subject to the non overlapping constraint: $\forall\, i \neq j$, $\mathcal{I}_i \cap \mathcal{I}_j = \emptyset$ (*only one queen by square*). Since choosing one IS corresponds to coloring n vertices (*or squares*), the depth of the search tree is lower than n.

At each stage of the search we identify the subsets IS_{ji} of IS that cover each free square (j, i) of the chessboard. The uncolored square which can be covered by the smallest number of remaining IS corresponds to the branching node. The process backtracks as soon as there is no more IS to cover an uncolored square.

The IS are not constructed once and for all because of the huge amount of memory required to store them as soon as n is growing up: for $n = 15$, we have $1475300\, IS$ and for $n = 16$, we have $9609410\, IS$ (*taking into account the filtering technique described below*).

3.2 Filtering

Reducing the search space while exploring it is a classical technique for solving Constraint Satisfaction Problem, CSP (*see, for instance, the study of Sabin and Freuder [9]*). Caramia and Dell'Olmo [2] have also applied constraint propagation to the Graph Coloring Problem.

In our approach, the implementation of a filtering process aims at decreasing the huge number of IS to be selected by using the non overlapping constraint that constitutes a dense constraint network. Moreover, we have found another implicit constraint, based on the cliques of the sub-graph of the uncolored vertices. Hence, our filtering procedure is two fold.

At first, every time an IS $\mathcal{I}_i$ is selected, propagation on the non overlapping constraint is carried out. Secondly, after the i^{th} IS assignment, $n - i$ other IS have to be chosen to form a solution. If, at this stage of the search, there is a clique of size $n - i$, $\mathcal{C}_{n-i}$, in the subgraph of the uncolored squares, then all the remaining IS to be chosen must cover one vertex in this clique. This means that if the condition $\mathcal{I}_j \cap \mathcal{C}_{n-i} = \emptyset$ holds for an $\mathcal{I}_j \in IS_n$, then we can remove $\mathcal{I}_j$ from the search space under the current node of the search tree. By construction, the $\mathcal{I}_j$ cannot produce such a condition with squares belonging to the same row or column (*there are no more than n rows and n columns in the chessboard, and each $\mathcal{I}_j$ counts n vertices*). This is not the case for the diagonals. For example, at the root of the search tree, we can delete the independent sets which do not cover one square of each of the 2 main diagonals. At the next node, we can consider these 2 main diagonals plus the 4 with $n - 1$ squares, and so on.

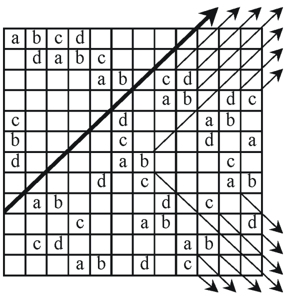

Figure 1. Chessboard configuration at node 4 for Queens_12^2

Figure 1 illustrates how this filtering works on the Queens_12^2 problem. At node 4, 4 colors are assigned and 14 diagonals must be checked before going on. Since the ascendant diagonal (*bold arrow on Figure 1*) defined by the formula $j + i = 8$ has no colored vertex, the process backtracks. Indeed, this clique contains 9 vertices and there are only 8 IS left to select in order to cover completely the chessboard.

Table 1. Comparative statistics

	Without filtering			With filtering		
n	IS	nodes	sec.	IS	nodes	sec.
10	724	2883	0	544	991	0
11	2680	45232	0	1744	15262	0
12	14200	7559966	163	9440	3834450	75
13	73712	1273054614	115146	52008	571824811	45389

Table 1 shows the number of IS generated when considering – or not – the filtering on cliques. It also shows the number of nodes produced at level 3 of the search tree and the elapsed CPU time, in seconds: half of the search space is pruned when filtering on the cliques of the sub-graph constituted by the uncolored vertices.

4 FIRST RESULTS

Up to Queens_11^2 the answer is instantaneous: there is no solution for Queens_10^2. Thus $\chi_{10} \geq 11$ since $\chi_{11} = 11$ we deduce that $\chi_{10} = 11$. Indeed, the first 10 rows and the first 10 columns of the Queens_11^2 solution constitute a 11 colors correct assignment for the 10×10 chessboard. Exploring the search tree for Queens_12^2 requires less than 1 hour CPU. The enumeration finds 454 solutions proving that $\chi_{12} = 12$. Hence, the algorithm achieved completeness for Queens_10^2 and Queens_12^2. However, only one solution is found for Queens_14^2 after a week of computing time. This result is sufficient enough to prove that $\chi_{14} = 14$. But we are not able to solve instances larger than Queens_14^2.

When examining the 454 solutions obtained on Queens_12^2, we observe that 98 of them present only horizontal symmetry, 98 present only vertical symmetry (the vertices of different IS come by pair), and 258 present both vertical and horizontal ones (the vertices of different IS come four at a time).

Those geometric properties and other ones, reflecting the repartition of the colors on the chessboard, are defined and exploited in the next section in order to go ahead with Queens_n^2 problems.

5 USING SYMMETRIES

Before describing each geometric property, let us define the grid maintaining the chessboard of dimension n. The left higher corner square has the coordinates $(0, 0)$ and the central square has the coordinates $(\frac{n-1}{2}, \frac{n-1}{2})$. Note that, if n is even, the value of $\frac{n-1}{2}$ remains fractional.

The general idea is to use some geometric operators to generate more than one independent set per branching node. A first application of this idea when $n = 2p$ consists in generating IS by pairs and consequently in reducing the depth of the search tree by 2. Secondly, depending on the n value, other geometric operations are combined and implemented.

We go on the assumption that there are solutions presenting such properties. Then the resulting enumeration algorithm does not guarantee to find solutions for a given Queens_n^2 instance. Hence, the elaborated algorithm is exact but incomplete.

5.1 Horizontal symmetry for $n = 2p$

The vertex corresponding to the square (j, i) of the chessboard is fixed at the same time with the vertex corresponding to the square (j', i') obtained by the linear transformation:

$$\begin{pmatrix} j' - \frac{n-1}{2} \\ i' - \frac{n-1}{2} \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \times \begin{pmatrix} j - \frac{n-1}{2} \\ i - \frac{n-1}{2} \end{pmatrix}$$

Accordingly, the pseudo code expressing the horizontal symmetry $(j', i') = \mathcal{H}(j, i)$ is:

$$\begin{cases} j' = n - j - 1 \\ i' = i \end{cases}$$

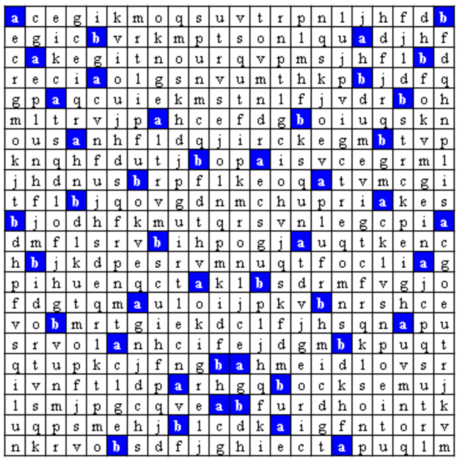

Figure 2. Certificate for $\chi_{22} = 22$

The independent sets $\{a, b\}, \cdots, \{u, v\}$ of Queens_22^2 are enumerated simultaneously: the depth of the search tree is divided by 2. This geometric operator works also for $n = 12, 14, 16$ and $n = 18$. However, the computing time becomes very important (see section 6) when n is growing up and no solution is found for $n = 20$ after 168 hours (*i.e. one week*) of computing time.

5.2 Composition of symmetries for $n = 4p$

In this case, the transformation is a composition of the vertical and horizontal symmetries. Hence, we consider –at the same time– the vertices (j, i), (j', i'), (j'', i'') and (j''', i'''), such as $(j', i') = \mathcal{H}(j, i)$ (horizontal symmetry) and $(j'', i'') = \mathcal{V}(j, i)$ where the operator $\mathcal{V}$ corresponds to the vertical symmetry transformation defined by the linear operation:

$$\begin{pmatrix} j' - \frac{n-1}{2} \\ i' - \frac{n-1}{2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \times \begin{pmatrix} j - \frac{n-1}{2} \\ i - \frac{n-1}{2} \end{pmatrix}$$

and the corresponding pseudo code is:

$$\begin{cases} j' = j \\ i' = n - i - 1 \end{cases}$$

Then we deduce the coordinates of the vertex corresponding to the square (j''', i''') by combining those two operators:

$$(j''', i''') = \mathcal{H} \circ \mathcal{V}\,(j, i).$$

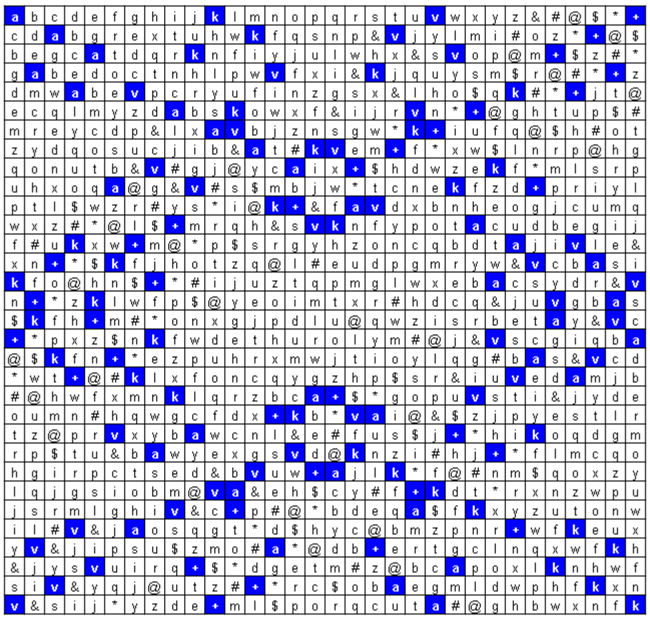

Figure 3. Certificate for $\chi_{32} = 32$

The independent sets $\{a, v, k, +\}, \cdots, \{b, \&, f, *\}$ of Queens_32^2 are enumerated simultaneously: the depth of the search tree is divided by 4. This geometric operator improves the computing times for $n = 12, 16$ and 20. It also enables to solve Queens_n^2 for $n = 24$ and 28.

These first geometric operators are not applicable to Queens_n^2 when n is odd. We consider this case in the sections below.

5.3 Central symmetry for $n = 3p$

The vertices corresponding to the squares (j, i) and (j', i') are assigned simultaneously, by a symmetry to the central square of the chessboard expressed by the linear operation:

$$\begin{pmatrix} j' - \frac{n-1}{2} \\ i' - \frac{n-1}{2} \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \times \begin{pmatrix} j - \frac{n-1}{2} \\ i - \frac{n-1}{2} \end{pmatrix}$$

and the pseudo code generating (i', j') from (i, j) is:

$$\begin{cases} j' = n - j - 1 \\ i' = n - i - 1 \end{cases}$$

This operation is wrong for IS that cover the central square of the chessboard, because this square is an invariant for this symmetry. In fact, it is not necessary to assign this last IS. Indeed, when the $n-1$ first IS are assigned, n squares (*including the central one*) still have to be colored. Each uncolored square belongs to exactly one row, column and diagonal. Let us prove this statement by contradiction: if there are two uncolored squares in a row (*column*) then only $n - 2$ squares are colored on this row (*column*), which is in contradiction with the hypothesis that $n - 1$ IS of size n are fixed. If there are two uncolored squares on a diagonal, then the process backtracks because of the filtering on the cliques constituted by the diagonals.

The independent sets $\{a, b\}, \cdots, \{m, n\}$ of Queens_15^2 are generated simultaneously: the depth of the search tree is divided by 2. The IS $\{o\}$ covering the central square of the chessboard is built implicitly.

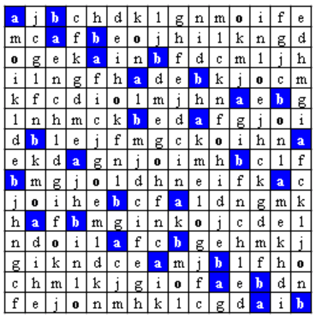

Figure 4. Certificate for $\chi_{15} = 15$

Even if this transformation allowed us to solve the Queens_n^2 for $n = 15$, no solution was found for $n = 21$ after 68 hours of computing time. In the next section, we will deal with this last case.

5.4 Composition of rotations for $n = 4p + 1$

Here the geometric operator is $\pi/2$ rotation around the central square of the chessboard and defined by the linear transformation:

$$\begin{pmatrix} j' - \frac{n-1}{2} \\ i' - \frac{n-1}{2} \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \times \begin{pmatrix} j - \frac{n-1}{2} \\ i - \frac{n-1}{2} \end{pmatrix}$$

and the corresponding pseudo code of the operation $(j', i') = \mathcal{R}(j, i)$ is:

$$\begin{cases} j' = n - i - 1 \\ i' = j \end{cases}$$

Two other vertices, (j'', i'') and (j''', i'''), are also generated simultaneously so that $(j'', i'') = \mathcal{R}(j', i') = \mathcal{R}^2(j, i)$ and $(j''', i''') = \mathcal{R}(j'', i'') = \mathcal{R}^3(j, i)$.

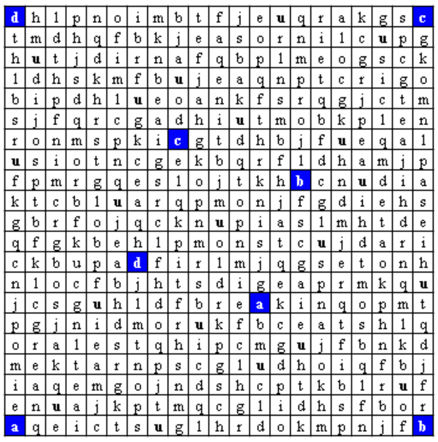

Figure 5. Certificate for $\chi_{21} = 21$

The independent sets $\{a, b, c, d\}, \cdots, \{q, r, s, t\}$ of Queens_21^2 are constructed simultaneously: the depth of the search tree is divided by 4. As for the Queens_15^2 problem, the independent set $\{u\}$, covering the central square, is built implicitly.

6 RESULT SYNTHESIS

The experiments are carried out on a Xeon 3 Ghz machine. The memory required to solve the considered instances is smaller than 1 MB, even if a large number of IS is generated. However, the Queens_22^2 problem was distributed on 20 P4 2.4 Ghz machines. Indeed, a machine is affected to each possible position of the second vertex of the first independent set, which covers the square of left higher corner of the chessboard.

Table 2. Computational time needed to obtain the first solutions.

	Geometric Operations				
n	No Op.	$\mathcal{H}$	$\mathcal{H}\&\mathcal{V}$	$\mathcal{R}^2$	$\mathcal{R}\&\mathcal{R}^2\&\mathcal{R}^3$
12	107	1	1	–	–
14	240952	5	–	–	–
15	–	–	–	4897	–
16	–	243	1	–	–
18	–	2171	–	–	–
20	–	–	1	–	–
21	–	–	–	–	30844
22*	–	233404*	–	–	–
24	–	–	10	–	–
28	–	–	1316	–	–
32	–	–	73790	–	–

In Table 2, for each dimension n, we express the elapsed time in seconds to reach a *first solution* by using (*or not: No Op. column*) the geometric properties described in the previous sections. The column $\mathcal{R}^2$ corresponds to the central symmetry (*$\mathcal{R}$ is the $\pi/2$ rotation described in section 5.4*). The line "–" means that either no solution is found after 168 hours of CPU time, or the geometric operator does not match with the dimension n of the chessboard.

Regarding the results, using the horizontal symmetries (*$\mathcal{H}$ column*) reduces the time needed to reach a solution from 107 sec. to 1 sec. for $n = 12$, and from 240952 sec. to 5 sec. for $n = 14$. The combination of both horizontal and vertical symmetries (*$\mathcal{H}\&\mathcal{V}$ column*) reduces again the required time, from 243 to 1 sec. for the Queens_16^2 problem and permits to solve Queens_n^2 for $n = 20, 24, 28$ and 32. However, in considering the three first columns, the required time to find a solution grows exponentially with the dimension of the Queens_n^2 problem. The resolution of larger problems (dimensions greater than 32) appears to be very difficult.

Finally, readers who are interested in other certificates are invited to consult the web page `www.lgi2p.ema.fr/~vasquez/queen.htm`.

7 CONCLUSION

In this paper, we proposed a branching algorithm to solve the Queens_n^2 coloring problems. It embeds an independent sets enumeration by a depth search procedure, and it is also reinforced by an efficient filtering based on the cliques belonging to the uncolored vertices of the queen graph. This first algorithm achieved the completeness for Queens_10^2 and Queens_12^2 problems and proved that $\chi_{14} = 14$.

The 454 Queens_12^2 solutions show horizontal and vertical symmetries between vertices belonging to pairs of independent sets. This observation gives us the intuition of designing an exact, *but incomplete*, algorithm around these geometric properties. This idea was primordial to improve the previous results. Indeed, the exploitation of the structural properties of the colors repartition on the chessboard (symmetries and rotations) allowed us to deal with larger Queens_n^2 problems. Thus, we improved significantly our first results: we succeeded to color up to 1024 vertices of the Queens_32^2 graph with 32 colors (against 196 vertices of the Queens_14^2 graph).

This work gives eleven counterexamples to the Gardner conjecture [4] stating that "*it is possible to color Queens_n^2 with n colors if and only if n is not divisible by either 2 or 3*".

Our future work consists in investigating two directions in order to deal with large instances: firstly we will attempt to find other geometric operators to generate simultaneously more vertices, and secondly we will try to find other important properties that should lead us to implement a more efficient filtering procedure.

REFERENCES

[1] M. Caramia and P. Dell'Olmo. Iterative Coloring Extension of a Maximum Clique. *Naval Research Logistic*, 48(6):518–550, 2001.

[2] M. Caramia and P. Dell'Olmo. Constraint Propagation in Graph Coloring. *Journal of Heuristic*, 8:83–107, 2002.

[3] M. Chiarandini and T. Stützle. An Application of Iterated Local Search to Graph Coloring Problem. In A. Mehrotra D. S. Johnson and M. Trick, editors, *Proceedings of the Computational Symposium on Graph Coloring and its Generalizations*, pages 112–125, Ithaca, New York, USA, September 2002.

[4] M. Gardner. *Further mathematical diversions : the paradox of the unexpected hanging and others*. Mathematical Association of America, 1995.

[5] J.P. Hamiez. *Coloration de graphes et planification de rencontres sportives: heuristiques, algorithmes et analyses*. PhD thesis, Université d'Angers, décembre 2002.

[6] G. Kochenberger, F. Glover, B. Alidaee, and C. Rego. An Unconstrained Quadratic Binary Programming Approach to the Vertex Coloring Problem. *To appear in Annals of Operations Research*.

[7] A. Marino and R. I. Damper. Breaking the symmetry of the graph colouring problem with genetic algorithms. *In Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2000), Late Breaking Papers*, pages 240–245, Las Vegas, NV, 2000.

[8] A. Mehrotra and M. A. Trick. A Column Generation Approach for Graph Coloring. *INFORMS Journal of Computing*, 8(4):344–354, 1996.

[9] D. Sabin and E. Freuder. Contradicting conventional wisdom in constraint satisfaction. In *ECAI'94*, pages 125–129, Amsterdam, 1994.