S3. Queen Domination Problem

(Background + Literature Review)

Taha Rostami

November, 2023

Domination in Chessboards, 2021 [1]

- **Chessboards:** finite, regular tessellations of the plane made up of identical square cells.
- **Different shapes:** n $\times$ n (the most common), rectangular m $\times$ n, triangular, triangular, or diamond-shaped, sawtooth square, circular boards, and three-dimensional boards, etc.
- **Different chess pieces:** queens (the most common), kings, rooks, bishops, knights ~~, and pawn~~.
- **Different placement conditions:** no limitation (the most common), major diagonal, the column nearest the center column, the border, etc.
- **Different Queries (aka Domination Chain of inequalities):** $ir(G) \leq \gamma(G) \leq i(G) \leq \alpha(G) \leq \Gamma(G) \leq IR(G)$

Queen Domination on n $\times$ n chessboard

- **Problem Def.** Minimum number of queens required to attack all squares.
- Q_n denotes the corresponding graph of the problem
- *domination number* $\gamma(Q_n)$ denotes size of the problem's solution
- No standard for numbering the squares [2]

An Important Problem Related to Queen Domination

- **Problem Def.** Minimum number of queens required to attack all squares while no two queens attack each other
- Q_n denotes the corresponding graph of the problem
- *independent domination number* $\mathbf{i}(Q_n)$ denotes size of the problem's solution

Complexity Class

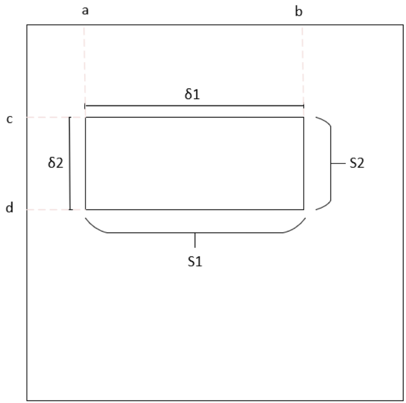
It is unknown whether the decision variants of the two mentioned problems belong to NP-hard or not. Meanwhile, there is no known polynomial time algorithm to solve this problem [2, 3].

Literature Review

- **Theoretical papers**: Theorems and well-established conjectures
  - Lower Bound
  - Upper Bound
- **Computational papers**: using computer to solve the open-cases
  - Search methods

Lower Bound Example: Theorem. $\gamma(Q_n) \geq (n-1)/2$ [4]

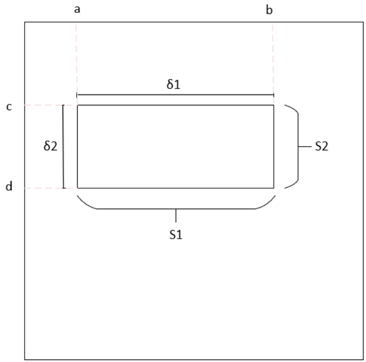
- Assume an arbitrary min. queen-dominating set of size $\gamma(Q_n)$
- Let **a** (**a**) be the left-most (left-most) empty column (the column that there is no queen is placed on it)
- Let **c** (**d**) be the empty row closest to the top (bottom)



S= S1 U S2, without loss of gen. δ1≥δ2

Lower Bound Example: Theorem. $\gamma(Q_n) \geq (n-1)/2$ (Cont.)[4]

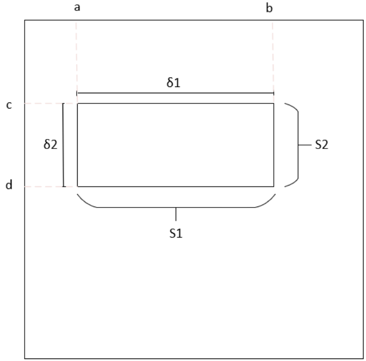
- Since $\delta 1 \geq \delta 2$, no diag. intersects both S1 and S2. Thus, every queen can diagonally attack at most two squares of S (i.e., at most one per diag.)
- queens above row **c** or below row **d** do not attack S by row or column



S= S1 U S2, without loss of gen. δ1≥δ2

Lower Bound Example: Theorem. $\gamma(Q_n) \geq (n-1)/2$ (Cont.)[4]

- By def., there are at least c-1 queens above row c. Also, each row below row d contains at least one queen and $d = c + \delta 2 \leq c + \delta 1$. Thus, all $n - c - \delta 1$ rows below row $c + \delta 1$ have at least one queen. So, there are at least $n - c - \delta 1$ queens below row $c + \delta 1 - 1$.



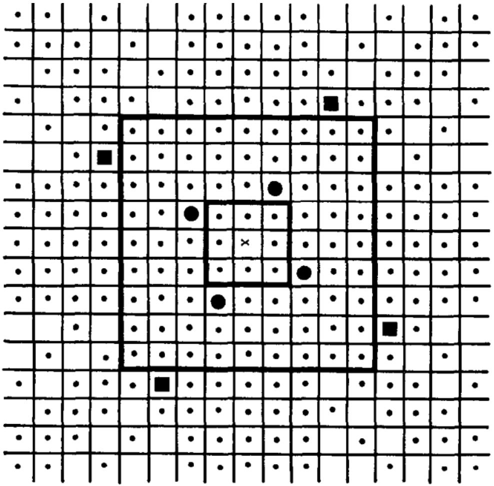
S= S1 U S2, without loss of gen. δ1≥δ2

Lower Bound Example: Theorem. $\gamma(Q_n) \geq (n-1)/2$ (Cont.)[4]

- So, at least $(c-1)+(n-c-\delta 1)=(n-\delta 1-1)$ queens attack at most 2 squares of S.
- The remaining queens are at most $\gamma-(n-\delta 1-1)$, may cover at most 4 squares of S.
- Since all the $2\delta 1$, squares of S must be dominated we have $2(n-\delta 1-1)+4(\gamma-(n-\delta 1-1)) \geq 2\delta 1$.
- Finally, the last one gives us $\gamma(Q_n) \geq (n-1)/2$. $\square$

Upper Bound Example: Theorem. $i(Q_n) \leq 0.705n + 0.895$[4]

- Construct a particular domination set based on some fixed patterns and then show the size of that set



Theoretical papers

- Some worth mentioning theorems [4, 5, 6, 7, 8, 9, 10, 11]:
  - $\gamma(Q_n) \geq (n-1)/2$
  - $n = 4k \Rightarrow \gamma(Q_n) \geq 2k$
  - $n = 4k+1 \Rightarrow \gamma(Q_n) \geq 2k+1$
  - $n = 4k \text{ and } n > 11 \Rightarrow \gamma(Q_n) \geq 2k+2$
  - $\gamma(Q_n) \leq i(Q_n) \leq 0.705n + 0.895$
  - $\gamma(Q_n) \leq i(Q_n) \leq 2n/3 + O(1)$
  - $\gamma(Q_n) \leq 31n/54 + O(1)$

Computational Papers: Gibbons and Webb, 1997 [12]

- backtracking + isomorphism rejection for $i(Q_n)$
- simulated annealing for $\gamma(Q_n)$

Operation	Notation	Square mapping
Identity	I	$(i,j) \to (i,j)$
Rotation anticlockwise by 90	C_1	$(i,j) \to (n\text{-}1\text{-}j,\ i)$
Rotation anticlockwise by 180	C_2	$(i,j) \to (n\text{-}1\text{-}i,\ n\text{-}1\text{-}j)$
Rotation anticlockwise by 270	C_3	$(i,j) \to (j,\ n\text{-}1\text{-}i)$
Reflection about up diagonal	D_1	$(i,j) \to (n\text{-}1\text{-}j,\ n\text{-}1\text{-}i)$
Reflection about down diagonal	D_2	$(i,j) \to (j,i)$
Vertical reflection	R_1	$(i,j) \to (n\text{-}1\text{-}i,j)$
Horizontal reflection	R_1	$(i,j) \to (i,n\text{-}1\text{-}j)$

Computational Papers: Gibbons and Webb, 1997 (Cont.) [12]

- The language C on a DEC ALPHA 3000/600 workstation
- Four new values for $\gamma(Q_n)$ when n=4k + 1 and values $1 \leq i(Q_n) \leq 16$
- The following table is related to $i(Q_n)$

n	k	# of non-equivalent solutions	Time
1	1	1	-
2	1	1	-
3	1	1	-
4	2	0	-
4	3	2	-
5	2	0	-
5	3	2	-
6	3	0	-
6	4	17	-
7	3	0	-
7	4	1	-
8	4	0	-
8	5	91	0.1 sec
9	4	0	0.1 sec
9	5	16	0.4 sec
10	4	0	0.4 sec
10	5	1	1.6 secs

Computational Papers: Gibbons and Webb, 1997 (Cont.) [12]

11	4	0	1.0 secs
11	5	1	5.9 secs
12	6	0	1.4 mins
12	7	105	3.7 mins
13	6	0	5.3 mins
13	7	4	19.6 mins
14	7	0	1.5 hours
14	8	55	4.5 hours
15	8	0	23.5 hours
15	9	1314	58.4 hours
16	8	0	113.7 hours

k	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
n	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65
$\gamma(Q_n)$	5	7	9	11	13	$\leq$ 17	17	19	$\leq$ 23	$\leq$ 25	25	27	29	31	$\leq$ 35

conjecture: $\gamma(Q_{4k+1}) = 2k+1$

Computational Papers: Kearse and Gibbons, 2001 [13]

- M1. adopting backtracking + isomorphism rejection [12] for queen domination problem
- M2. Rather than backtracking on queen placements, an improvement is to systematically identify undominated squares and to try placing pieces in all positions that attack these squares. [Isomorph Rejection is hard for this, so it is applied without such rejection]
- M3. Instead of backtracking based on the positions of the queens, this method backtracks based on the set of queens attacking a given position. [With Isomorph Rejection]
- Cutoff + Dominated queens rejection
- adopting M1+M2+M3, i.e., M1 t1 M2 t2 M3 while enhancing the search by Cutoff + Dominated queens rejection. t1 and t2 are determined experimentally.

Computational Papers: Kearse and Gibbons, 2001 [13]

- The language C on an AlphaServer 2100 Model 4/275 running at 275 Mhz under Digital UNIX V4.0D.

Queens Domination

Board size	Number of Queens	Number of Non-Isomorphic Solutions	Time
3	1	1	-
4	2	3	-
5	3	37	-
6	3	1	-
7	4	13	-
8	5	638	1.1 secs
9	5	21	0.4 secs
10	5	1	0.7 secs
11	5	1	1.2 secs
12	6	1	34 secs
13	7	41	18 mins
14	**8**	**588**	11 hours
15	**8**	**0**	21 hours
15	**9**	**25872**	230 hours
16	**8**	**0**	31 hours
17	8	0	30 hours
18	8	0	43 hours

Independent Queens Domination

Board size	Number of Queens	Number of Non-Isomorphic Solutions	Time
3	1	1	-
4	3	2	-
5	3	2	-
6	4	17	-
7	4	1	-
8	5	91	-
9	5	1	-
10	5	1	-
11	5	1	0.2 secs
12	7	105	34 secs
13	7	4	45 secs
14	8	55	9.4 mins
15	9	1314	110 mins
16	**9**	**16**	5 hours
17	**9**	**2**	11 hours
18	**9**	**0**	28 hours
18	**10**	**28**	120 hours
19	**9**	**0**	62 hours

Computational Papers: W. H. Bird, 2017 [2]

Framework 3.1 Framework for a Backtracking Search

```
procedure FindDominatingSet(G, P, C, B, desired_size)
    if P is a dominating set then
        if |P| < |B| then
            Overwrite B with a copy of P
        end if
        return
    end if
    Compute a lower bound k on the size of a dominating set D
    such that P ⊆ D ⊆ P ∪ C.
    if k ≥ |B| or k > desired_size then return
    T ← ∅
    v ← An undominated vertex of G
    for each vertex u ∈ N[v] ∩ C do
        T ← T ∪ {u}
        FindDominatingSet(G, P ∪ {u}, C − T, B, desired_size)
    end for
end procedure
```

Computational Papers: W. H. Bird, 2017 [2]

- 2h per run, 1 core ; a four-core Intel Core i7-3770 running at 3.4 GHz

Graph	n	m	γ	SageMath Max. Time (s)	Framework 3.1 Variant	Framework 3.1 Max. Time (s)
Queen (10)	100	1470	5	2.74	MDD Bounding (Max. MDD, desc, no FS, RC)	0.014
Queen (11)	121	1980	5	7.488	MDD Bounding (Max. MDD, desc, no FS, RC)	0.01
Queen (12)	144	2596	6	113.8	MDD Bounding (Max. MDD, desc, no FS, RC)	0.166
Queen (13)	169	3328	7	223.4	MDD Bounding (Min. MDD, desc, no FS, RC)	3.412
Queen (14)	196	4186	8	N/A	MDD Bounding (Min. MDD, desc, no FS, RC)	96.59
Queen (15)	225	5180	9	N/A	MDD Bounding (Min. MDD, desc, no FS, RC)	3069

Computational Papers: W. H. Bird, 2017 [2]

- Parallelism. "To solve the open cases of the queen domination problems, the res/mod approach was used to split the computation among a cluster of 64 processors."

Number of Minimum Dominating Sets of Queen Graphs up to Isomorphism

n	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Size	1	2	3	3	4	5	5	5	5	6	7	8	9	9	9	9
#	1	3	37	1	13	638	21	1	1	1	41	588	25872	**43**	**22**	**2**

Number of Minimum Independent Dominating Sets of Queen Graphs up to Isomorphism

n	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Size	1	3	3	4	4	5	5	5	5	7	7	8	9	9	9	10
#	1	2	2	17	1	91	16	1	1	105	4	55	1314	16	2	28

Number of Minimum Border Dominating Sets of Queen Graphs up to Isomorphism

n	3	4	5	6	7	8	9	10	11	12
Size	2	2	3	4	5	6	6	6	9	10
#	4	1	6	19	75	174	1	1	1017	979

n	13	14	15	16	17	18	19	20	21	22
Size	9	12	13	10	14	16	13	18	19	14
#	4	1094	2635	2	32	1457	8	2080	6128	4

n	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
$\gamma(\text{Queen}(n))$	5	5	6	7	8	9	9	9	9	10	**11**	11	**12**	12	**13**
$i(\text{Queen}(n))$	5	5	7	7	8	9	9	9	10	**11**	**11**	11	**12**	**13**	**13**
$\text{bor}(\text{Queen}(n))$	6	9	10	9	**12**	**13**	**10**	**14**	**16**	**13**	**18**	**19**	**14**	**21**	**22**

Computational Papers

- Some worth mentioning search techniques [2, 3, 12, 13]:
  - Backtracking (with different heuristics), Dynamic Programming on Subsets, Treewidth, and Path Decomposition
  - With/Without Isomorphism rejection
  - With/Without Domination rejection
  - With/Without Cutoff
- We mostly covered papers related to queen domination problem. It is worth mentioning that other techniques and formulations such as Integer Linear Programming [14], MaxSAT[15] encoding etc. are adopted for different variants of this problem and also for different graph theory's problems.
- Best known values for $i(Q_n)$ are available here
- According to this link, $i(Q_{26})$ is known, and according to this link a domination set of size 14 for Q_{26} is known. However, the first open-case is still $n = 26$ since $\gamma(Q_{26})$ is unknown.

To Be Read In The Future

- W. D. Weakley, "Queen Domination of Even Square Boards," The Electronic Journal of Combinatorics, pp. 2-50, 2022
- W.D. Weakley, "Domination in the queen's graph," Graph Theory, Combinatorics, and Algorithms, pp. 1223–1232, 1995. [not available online]

Discussion

- 'maplesat' SAT solver, Sinz's method for cardinality enc., standard QDom into SAT enc.

n	$\gamma(Q_{n-1}) - 1$ [non existence]	$\gamma(Q_{n-1})$	$\gamma(Q_{n-1}) + 1$
10	4 (s)	6 (s)	0 (s)
11	5 (s)	15 (s)	10 (s)
12	7*60 (s)	24 (s)	4 (s)
13	90*60 (s)	1 (s)	5 (s)

Discussion (Cont.)

- Queen Redundancy enc using Tseytin; Z3's Tactics, python-nnf [somewhat messing up the code base]; impl. from scratch [risk of introducing bugs]
- Our Encodings;3CNF; Tseytin Enc; equisatisfiable !?
- "The Tseytin transformation, alternatively written Tseitin transformation, takes as input an arbitrary combinatorial logic circuit and produces an equisatisfiable boolean formula in conjunctive normal form (CNF)." Wiki
- "The simplify tactic is equivalence preserving, so it does not change the number of models. On the other hand, no such guarantees are provided for bit-blasting and especially not tseitin-cnf. They were not implemented with model count preservation in mind." Nikolaj Bjorner + Context might be needed

Discussion (Cont.)

- Batch combination/permutation gen.

Discussion (Cont.)

- Cutoff and iso. rej. [have yet to work on]
- pynauty [x windows; + Linux] + one sys for consistency BTW experiments ?!?

Discussion (Cont.)

- Sharing An Idea: "Cameron, C., Hartford, J., Lundy, T., Truong, T., Milligan, A., Chen, R. and Leyton-Brown, K., 2022. Finding the smallest tree in the forest: Monte Carlo Forest Search for UNSAT solving." + "Cameron, C., Hartford, J., Lundy, T., Truong, T., Milligan, A., Chen, R. and Leyton-Brown, K., 2022. Monte Carlo Forest Search: UNSAT Solver Synthesis via Reinforcement learning. arXiv preprint arXiv:2211.12581."

Discussion (Cont.)

- TOEFL

References I

[1] J.T. Hedetniemi, and S.T. Hedetniemi, "Domination in Chessboards," in Structures of Domination in Graphs, pp.341-386, 2021.

[2] W. H. Bird, "Computational methods for domination problems," Doctoral dissertation, 2017.

[3] H. Fernau, "Dynamic programming for queen domination." in CTW, pp. 43-48, 2007.

[4] E. J. Cockayne, "Chessboard domination problems," Discrete Mathematics 86, pp. 13-20, 1990.

[5] V. Raghavan, and S. M. Venkatesan, "On bounds for a board covering problem," Information Processing Letters, pp. 281–284, 1987.

References II

[6] D. Finozhenok, and W. D. Weakley, "An improved lower bound for domination numbers of the queen's graph," Mathematical Sciences Faculty Publications, 2007.

[7] W. D. Weakley, "Domination in the queen's graph," Graph Theory, Combinatorics, and Algorithms, pp. 1223–1232, 1995.

[8] W. D. Weakley, "A lower bound for domination numbers of the queen's graph," The Journal of Combinatorial Mathematics and Combinatorial Computing, pp. 231–254, 2002.

[9] A. Karandikar, and A. Dutta, "Improved lower bounds for Queen's Domination via an exactly-solvable relaxation," arXiv, 2023

[10] A.P. Burger, CM. Mynhardt and E.J. Cockayne, "Domination and Irredundance in the Queen's Graph," Discrete Mathematics, 1997.

References III

[11] C.M. Grinstead, B. Hahne, and D. Van Stone, "On the Queen Domination Problem," In Annals of Discrete Mathematics, pp. 21-26, 1991.

[12] P.B. Gibbons, and J. A. Webb, "Some new results for the queens domination problem," Australasian Journal of Combinatorics, pp.145-160, 1997.

[13] M. D. Kearse, and P. B. Gibbons, "Computational methods and new results for chessboard problems," Centre for Discrete Mathematics and Theoretical Computer Science Research Report Series, 2000.

[14] A. Langlois-Rémillard, C. Müßig, and É. Róldan, "Complexity of Chess Domination," arXiv, 2023.

References IV

[15] Z. Lei, and S. Cai. "Solving Set Cover and Dominating Set via Maximum Satisfiability," In The Thirty-Fourth AAAI Conference on Artificial Intelligence, pp. 1569–1576, 2020.