Solving the N^2 Queens Problem via the MCMC Metropolis–Hastings Algorithm

Basile Bultez, Jean Ciardo, Tomas Jouven
Department of Computer Science, EPFL, Switzerland

***Abstract*—In this work, we investigate a variant of the classical N-Queens problem, in which N^2 queens must be placed in a N^3 space without any conflicts, using the Markov Chain Monte Carlo method, and more precisely the Metropolis-Hasting algorithm. For this specific formulation, a conflict arises when two queens share any pair of coordinates (generalizing rows and columns), or lie on a common 2D or 3D diagonal.**

I. Theoretical Setup

The core objective of our approach is to employ the Metropolis-Hastings algorithm to sample from the space of valid configurations. The goal is to design a Markov Chain whose stationary distribution approximates the uniform distribution over the solutions to the N^2-Queens problem. To achieve this, we must rigorously define the state space and the probability distributions governing the chain.

A. The Metropolis-Hastings Framework

Let $\mathcal{S}$ denote the state space of possible queen arrangements. We aim to sample from a target distribution $\pi(x)$ defined over $x \in \mathcal{S}$. In the context of constraint satisfaction problems, we typically define an energy function (or cost function) $E : \mathcal{S} \to R_{\geq 0}$, where $E(x)$ represents the number of conflicts in configuration x. A valid solution corresponds to a state x^* such that $E(x^*) = 0$.

We define the target distribution using the Boltzmann distribution:

$$\pi(x) = \frac{1}{Z} e^{-E(x)/T} \tag{1}$$

where Z is a normalizing constant and T is a temperature parameter . As $T \to 0$, the probability mass concentrates on the global minima of $E(x)$, i.e., the solutions.

B. State Space Definition and Reduction

Defining an efficient state space is crucial for the convergence of the MCMC algorithm. We considered a sequence of reductions to minimize the search space dimension.

1) Naive Approach: The 3D Grid: The most intuitive state space consists of all possible ways to place N^2 queens into an $N \times N \times N$ discrete grid. The cardinality of this space is given by the binomial coefficient:

$$|\mathcal{S}_{naive}| = \binom{N^3}{N^2}$$

For any non-trivial N, this magnitude is computationally intractable, making the probability of stumbling upon a valid configuration via random sampling negligible.

2) Matrix Representation (Height Map): To reduce dimensionality, we leverage the problem constraints. Since no two queens can share the same (i, j) coordinates (vertical conflict), we can represent the configuration as an $N \times N$ matrix X, where the entry $X_{i,j} = k \in \{1, \ldots, N\}$ denotes the height (or z-coordinate) of the queen located at position (i, j). This representation implicitly satisfies the vertical non-conflict constraint, as every cell (i, j) contains exactly one queen height. The size of this reduced state space is:

$$|\mathcal{S}_{matrix}| = N^{N^2}$$

While significantly smaller than $\mathcal{S}_{naive}$, this space still grows super-exponentially.

3) Column-Restricted Permutations: To further prune the state space without imposing excessive rigidity, we introduce a constraint along a single dimension. Specifically, we require that for every fixed column j, the vector of heights $(X_{1,j}, \ldots, X_{N,j})$ forms a permutation of $\{1, \ldots, N\}$.

Formally, the state space is reduced to:

$$\mathcal{S}_{col} = \{X \in \{1, \ldots, N\}^{N \times N} \mid \forall j, \bigcup_{i=1}^{N} \{X_{i,j}\} = \{1, \ldots, N\}\}$$

In this formulation, no two queens within the same column share the same height. Consequently, conflicts along the column axes are eliminated by definition. The cardinality of this space is:

$$|\mathcal{S}_{col}| = (N!)^N$$

Since $N! \ll N^N$ for large N, this represents a drastic reduction compared to the unconstrained matrix space $|\mathcal{S}_{matrix}| = N^{N^2}$.

a) Trade-off on Further Constraints: It is possible to restrict the space even further by enforcing uniqueness along the rows as well, which would limit the search to the set of Latin Squares. While this would significantly minimize the state space size, it introduces a major drawback regarding the Markov Chain's dynamics. High-constraint spaces often exhibit poor connectivity, making it difficult to design local moves that allow the chain to explore the space efficiently without getting trapped in local minima. Therefore, in this work, we focus our analysis on the **Matrix** and **Column-Restricted** state spaces, as they offer a balanced trade-off between search space size and the fluidity of the Metropolis-Hastings evolution.

C. Base Chain

The efficiency of the Metropolis-Hastings algorithm relies heavily on the design of the proposal chain $\psi(x)$. This chain dictates how the algorithm explores the state space. We define two distinct base chain corresponding to the two state space formulations discussed previously.

1) Unconstrained Local Update: For the initial matrix state space $\mathcal{S}_{matrix}$, we employ a standard single-site update. This chain is designed to explore the space N^{N^2} by modifying the height of a single queen at a time.

Mechanism: Given a current state X, the proposal is generated as follows:

1) Select a coordinate (i, j) uniformly at random from $\{1, \ldots, N\}^2$.
2) Select a new height $k' \in \{1, \ldots, N\}$ uniformly at random.
3) Construct the candidate state X' by setting $X'_{i,j} = k'$ and keeping all other entries identical to X.

Theoretical Properties: This proposal defines a random walk on the hypergrid of configurations. We can establish the convergence of the algorithm through the following properties:

- **Irreducibility:** Since there is a non-zero probability of transforming any state X into any other state Y through a finite sequence of local updates, the chain is *irreducible*.
- **Positive Recurrence:** As the state space is finite, irreducibility implies that the chain is *positive recurrent*.
- **Aperiodicity:** The existence of self-loops ensures that the base chain is *aperiodic*.

Together, these properties guarantee that the Markov Chain ψ is *ergodic*.

2) Column-Wise Swap Update: In the reduced state space $\mathcal{S}_{col}$, the column constraints must be preserved strictly. A simple single-site update is inadmissible here, as changing a single value $X_{i,j}$ would almost certainly violate the permutation property of column j. Instead, we employ a *swap-based* logic.

Mechanism: To move between states while remaining on the manifold of valid column permutations, we perform a 2-cycle permutation (a swap) within a single column:

1) Select a column index j uniformly at random.
2) Select two distinct row indices i_1, i_2 uniformly at random.
3) Construct the candidate state X' by swapping the values at these positions:

$$X'_{i_1,j} = X_{i_2,j} \quad \text{and} \quad X'_{i_2,j} = X_{i_1,j}$$

Theoretical Properties:

Regarding convergence, we note that the base chain induced by swap moves could theoretically exhibit periodicity. However, strict aperiodicity of the base chain is not a prerequisite for the Metropolis-Hastings algorithm in our context. The Metropolis chain itself becomes *aperiodic* effectively as long as there is a non-zero probability of rejecting a move (i.e., remaining in the current state). Since the acceptance probability $\alpha(x, x') = \min(1, \pi(x')/\pi(x))$ is strictly less than 1 for many proposed transitions in our complex energy landscape, self-transitions occur naturally. These self-loops render the effective Metropolis chain aperiodic, ensuring its convergence to the stationary distribution.

3) Hybrid Proposal Strategy: While the single-site and swap-based updates described above theoretically guarantee ergodicity, they are ”local” in nature. In the high-dimensional energy landscape of the N^2-Queens problem, purely local strategies often suffer from *metastability*: the Markov Chain can become trapped in deep local minima for exponentially long periods, as escaping requires a sequence of energy-increasing moves that are statistically unlikely at low temperatures.

To mitigate this, we implement a **Mixture Base Chain** strategy. This approach augments the standard base chain (either unconstrained or column-wise) with a ”global” perturbation move, selected probabilistically at each step.

Mechanism: Let Q_{base} be the transition of the chosen base chain (e.g., Column-Wise Swap) and $Q_{shuffle}$ be a global shuffling. The effective proposal distribution Q_{hybrid} is defined as a mixture:

$$Q_{hybrid}(x, \cdot) = (1 - \rho)Q_{base}(x, \cdot) + \rho Q_{shuffle}(x, \cdot)$$

where $\rho \in [0, 1]$ is a small mixing parameter.

The **Sub-cube Shuffle** ($Q_{shuffle}$) operates as follows:

1) Select a random center coordinate (c_i, c_j) and a radius r (proportional to N).
2) Identify the sub-grid of cells within this radius.
3) Randomly permute the height values strictly within this sub-grid.

This approach allows the chain to tunnel through high-energy barriers that would be impassable for single-step moves.

Finally, since the hybrid approach includes the base chain transitions with a non-zero probability (specifically $1 - \rho$), the fundamental theoretical properties of irreducibility and positive recurrence established previously remain strictly valid.

D. The Energy Model and Acceptance Probability

The driving force of the Metropolis-Hastings algorithm is the energy function $E : \mathcal{S} \to N$, which quantifies the number of constraint violations in a configuration X. Our objective is to reach a ground state X^* such that $E(X^*) = 0$.

Formally, the energy is defined as the total count of conflicting queen pairs. A conflict arises if two queens share a coordinate along any axis (linear conflict) or lie on the same 2D or 3D diagonal.

Crucially, the transition mechanism of our base chain is **symmetric**. This means that the probability $\Psi_{X,X'}$ of proposing a move from state X to X' is identical to the probability $\Psi_{X',X}$ of the reverse move. This symmetry causes the proposal ratio $\frac{\Psi_{X',X}}{\Psi_{X,X'}}$ to cancel out in the standard Metropolis-Hastings formula. Consequently, the acceptance probability depends solely on the ratio of the target Boltzmann distributions:

$$\alpha = \min\left(1, \frac{\pi(X_{new})}{\pi(X_{old})}\right) = \min\left(1, e^{-\Delta E/T}\right) \qquad (2)$$

where $\Delta E = E(X_{new}) - E(X_{old})$.

To ensure computational efficiency, ΔE is calculated locally. Instead of re-scanning the entire N^3 volume, we only evaluate the conflicts created or removed by the specific queens involved in the transition, reducing the complexity of an update step from $O(N^4)$ to $O(N^2)$.

E. The Final Metropolis Chains

By combining the state space definitions, the symmetric proposal mechanisms Ψ, and the acceptance probability α, we fully characterize the transition matrix P of the Metropolis chain. For any two distinct configurations $X, Y \in \mathcal{S}$, the probability of transitioning from X to Y is given by:

$$P_{X,Y} = \Psi_{X,Y} \cdot \alpha(X, Y) = \Psi_{X,Y} \cdot \min\left(1, e^{-(E(Y)-E(X))/T}\right) \qquad (3)$$

The probability of remaining in the same state (rejecting a move or proposing a null move) is $P_{X,X} = 1 - \sum_{Y \neq X} P_{X,Y}$. We distinguish two specific instantiations of this algorithm based on the chosen base chain:

a) 1. The Unconstrained Chain: Operating on $\mathcal{S}_{matrix}$, this chain utilizes the single-site update proposal. At each step, it attempts to modify one of the N^2 variables. While this chain benefits from a simple implementation and high mobility, it searches within a superexponential space (N^{N^2}) that contains a vast majority of irrelevant configurations.

b) 2. The Constrained Chain (Swap Dynamics): Operating on $\mathcal{S}_{col}$, this chain utilizes the column-wise swap proposal. It evolves strictly on the manifold of column-valid permutations. Although the connectivity is more restricted, the dimension reduction to $(N!)^N$ implies that the chain never wastes iterations on states with i-direction conflicts, potentially accelerating convergence toward a zero-conflict solution ($E(X) = 0$).

II. Temperature Scheduling

In the context of Simulated Annealing applied to the N^2-Queens problem, the temperature parameter T acts as a control variable for the exploration-exploitation trade-off. It directly influences the Metropolis acceptance probability for an energy-increasing move $\Delta E > 0$:

$$P(\text{accept}) = \exp\left(-\frac{\Delta E}{T}\right) \qquad (4)$$

A high temperature allows the Markov Chain to overcome energy barriers (exploration), while a low temperature forces the system to descend into local minima (exploitation). The *cooling schedule* determines how T decreases over time k. We investigate two approaches: Geometric and a custom Adaptive strategy.

A. Initial Temperature Calibration

Before initiating the cooling schedule, it is crucial to determine an appropriate starting temperature T_0. An arbitrarily high T_0 results in a purely random walk, wasting computational resources, while a T_0 that is too low may result in immediate entrapment in a local minimum (quenching).

We employ a calibration method based on the initial acceptance rate. We perform a short random walk phase to collect a sample of energy degradation values $\Delta E > 0$. Let $\langle \Delta E \rangle$ be the average of these positive energy changes. Given a target initial acceptance probability τ_0 (typically 0.8), we derive T_0 by inverting the Boltzmann factor:

$$\tau_0 \approx \exp\left(-\frac{\langle \Delta E \rangle}{T_0}\right) \implies T_0 = -\frac{\langle \Delta E \rangle}{\ln(\tau_0)} \qquad (5)$$

This ensures that at the start of the process, approximately $100 \times \tau_0\%$ of bad moves are accepted, allowing sufficient mobility in the state space.

B. Geometric Schedule

The geometric (or exponential) schedule is the standard approach in Simulated Annealing. The temperature at step $k+1$ is updated using a constant cooling factor α:

$$T_{k+1} = \alpha \cdot T_k \quad \text{with} \quad 0 < \alpha < 1 \qquad (6)$$

This schedule ensures a slow, asymptotic approach to minimum energy, allowing the system to settle into thermal equilibrium at each temperature level if α is sufficiently close to 1.

C. Adaptive Schedule with Reheating

The standard geometric schedule suffers from the risk of ”freezing”: if the temperature drops too quickly while the system is in a deep local minimum, the probability of escaping becomes negligible. To mitigate this, we propose an **Adaptive Schedule** that reacts to the dynamics of the chain.

The adaptive mechanism monitors the *local best energy* (E^*_{local}) observed during the current cooling phase. We define a stagnation counter sc that increments when the current energy E_k fails to improve upon E^*_{local}.

The schedule follows a hybrid logic:

- **Cruising (Exploitation):** If the system is currently finding better states ($E_k < E^*_{local}$), we update the local record and pause the cooling to fully exploit the current basin of attraction.
- **Reheating (Escape):** If the stagnation counter exceeds a predefined limit sc_{limit} (indicating the system is trapped in a local minimum), we trigger a *reheating event*. The temperature is reset to a fraction γ of the initial temperature:

  $$T_{new} = \gamma \cdot T_0 \quad (\text{e.g., } \gamma = 0.3) \qquad (7)$$

  Simultaneously, the stagnation memory is wiped. This injection of ”thermal energy” increases the acceptance probability for $\Delta E > 0$, allowing the chain to climb out of the local minimum.

- **Standard Cooling:** In the absence of improvement or excessive stagnation, the system follows the standard geometric decay $T_{k+1} = \alpha T_k$.

This approach allows the MCMC chain to oscillate between phases of exploration (high T after reheating) and exploitation (low T during cooling), significantly improving robustness for large N.

Collectively, these techniques establish a robust theoretical framework for navigating the complex energy landscape of the N^2-Queens problem, laying the groundwork for the empirical analysis presented in the following section.

III. Numerical Results

A. Parameters

To model the different approaches described above, we had to define several parameters and hyperparameters:

- N: the dimension of the problem.
- max_steps: the maximum number of steps allowed for the MCMC algorithm. The algorithm stops earlier if a zero-energy state is reached.
- T_0: the initial temperature used for simulated annealing.
- α: the annealing coefficient.
- $state$: the choice between `StackState` and `ConstraintStackState`, corresponding respectively to the Matrix and Column-Restricted state spaces mentioned above.
- $mode\_init$: the initialization mode for the state. We implemented the following options:
  - Random: a fully random initialization without any constraint.
  - Random Latin Square: for any given column $(i, *)$, we ensure that all heights k are different, enforcing the constraint that all queens in $(i, *)$ have distinct heights.
  - Noisy Latin Square: this method is motivated by the fact that the previous initialization may be too constrained. We first generate a Random Latin Square initialization, then perturb each position (i, j) with probability p by replacing its height with a random value in $\{1, \ldots, N\}$.
  - Layer-Balanced Random: heights in $\{1, \ldots, N\}$ are assigned to each (i, j) with a bias toward using each height approximately N times overall, without enforcing any row- or column-wise permutation constraints.
- $noisy\_p$: the parameter controlling the perturbation probability p used in the Noisy Latin Square initialization.
- re_heat: a parameter enabling reheating, following the strategy described in Section II.C.
- $n\_simulations$: for running simulations in parallel (if possible).

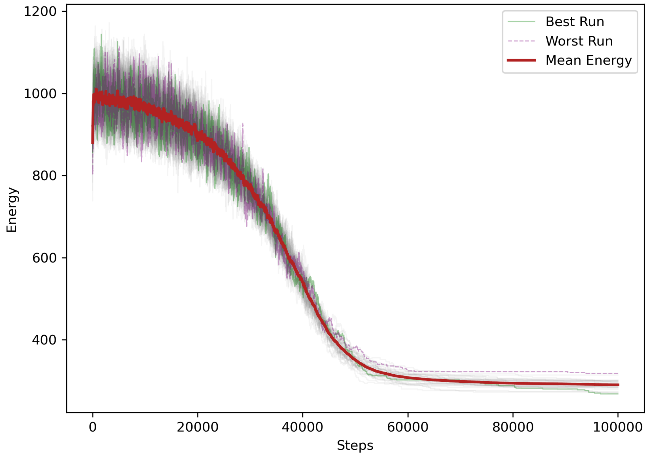

Fig. 1. Mean energy (red) and dispersion (individual runs in grey, best and worst highlighted) as a function of Monte Carlo steps for simulated geometric annealing with $N = 12$ over 40 independent runs from a Noisy Latin Square initialization.

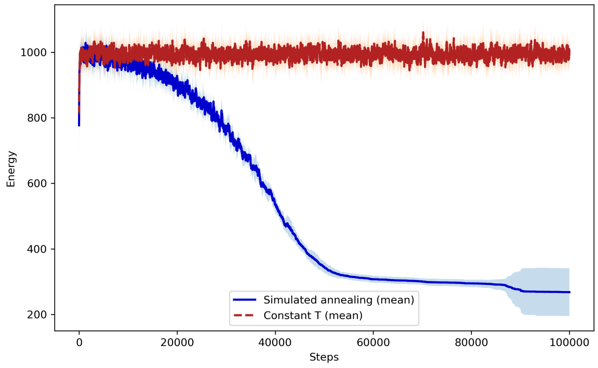

Fig. 2. Mean energy vs. steps with $N = 12$ averaged over 10 runs. Simulated annealing (blue, geometric schedule with $T_0 = 100, \alpha = 0.9999$) converges to much lower energies than constant-temperature MCMC at $T = 100$ (red dashed), showing the benefit of annealing.

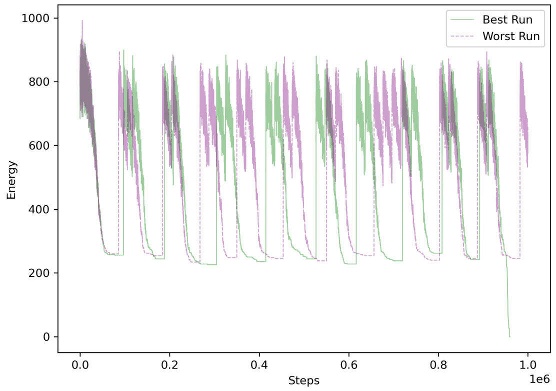

Fig. 3. **Energy landscape exploration for N=11.** Evolution of the system energy (number of conflicts) over MCMC steps, starting a Noisy Latin Square configuration. The plot demonstrates the efficacy of the Adaptive Schedule: sharp vertical spikes correspond to "reheating" events where the temperature is reset to escape local minima, followed by geometric cooling phases (exploitation). The process successfully converges to the ground state $E = 0$.

B. Finding Exact Solutions: The Quest for States of Zero-Energy

1) The Existence of Solutions: Solutions exist only for specific values of N. In fact, the problem admits valid zero-energy configurations only if the square root of the number of queens is coprime to 210 [1], yielding the set $N \in \{11, 13, 17, 19, 23, \ldots\}$. However, theoretical existence does not imply computational feasibility. Since the search space grows super-polynomially with N, even a small increment in problem size pushes the complexity beyond the reach of naive enumeration. Consequently, we focused our search for exact ground states ($E = 0$) on the tractable instances of $N = 11$ and $N = 13$. The case $N = 17$, while theoretically solvable, proved computationally prohibitive for this scope. This is illustrated in figure 4, while we did not succeed in finding an absolute solution for $N = 13$.

2) Experimental Results: To locate the global minimum, we evaluated various combinations of the theoretical components developed in previous sections.

Figure 1 represents a classical simulation result for a value of N for which no perfectly ordered state exists, where the mean energy averaged across multiple runes cleanly decreases. The best and worse runs are in the same order of magnitude as all the others. To illustrate the choices we made during the conception of our different simulations, figure 2 shows the difference between using simulated annealing and a constant temperature.

Figure 3 depicts a representative energy minimization trajectory for the N=11 instance, employing the adaptive reheating schedule coupled with the Mixture proposal. As expected, we were able to find an ordered state where no queens attack each other, relatively easily. Figure 6 shows a similar successful attempt; simpler construction without reheating converging quicker (around 600’000 steps). On the contrary, the illustrated attempt for $N = 13$ in figure 7, shows the difficulty for larger state space to find these elusive states, as the size explodes and the local minima become harder to escape.

One of the solutions to this problem is illustrated in figure 5 for $N = 11$, where we clearly perceive some higher order pattern.

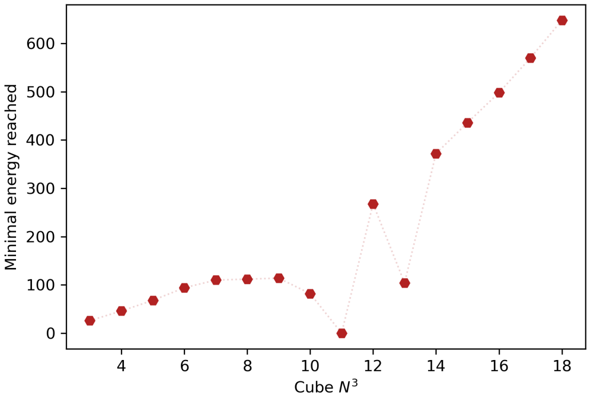

Fig. 4. Lowest energy vs. N, with $max\_steps = 10^6$ We observe that the minimum energy is globally increasing. However, for $N = 11$, we reach 0. We also notice a substantial energy reduction for $N = 13$, we will see that we should have obtain 0

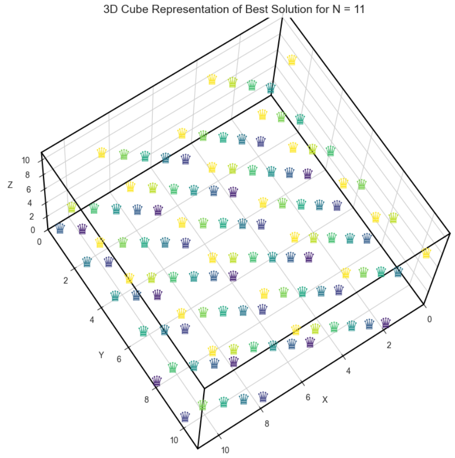

Fig. 5. 3D representation of a solution obtained for $N = 11$. The solution for N=11 is not random but structured: the queen positions follow a pattern that repeats across the successive layers (color-coded for clarity).

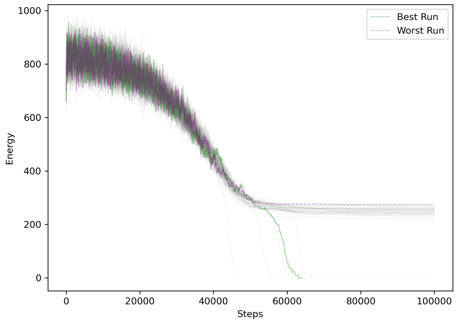

Fig. 6. **Energy landscape exploration for N=11.** Evolution of the system energy (number of conflicts) over MCMC steps, starting a Noisy Latin Square configuration without reheating. The process successfully converges to the ground state $E = 0$.

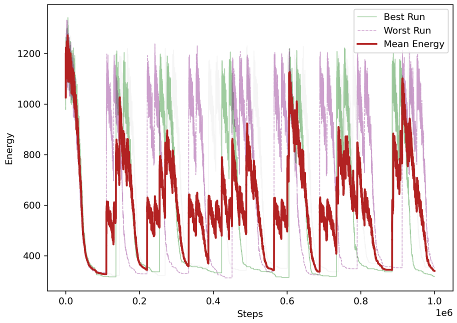

Fig. 7. **Energy landscape exploration for N=13.** Evolution of the system energy (number of conflicts) over MCMC steps, starting a Noisy Latin Square configuration, with reheating. The figure illustrates the complexity of finding absolute solutions for higher values of N.

References

[1] D. Klarner, “The problem of reflecting queens,” *The American Mathematical Monthly*, vol. 74, no. 8, pp. 953–955, 1967.

The implementation is available on our GitHub repository.