Markov Chains Project

Adrien Vauthey Alejandro Hernandezcano Konrad Litwiński

Contents

1 Finding solution

To find a solution, we will define a vector of n components where the index corresponds to the row of the chessboard and the value corresponds to the column. We will put a single queen per row and column. The state space S will be the set of all permutations.

Let us define the energy function $f : S \to \mathbb{R}$ to be $f(x) = \#$collisions between queens.
As we have one queen on each row and each column, collisions only happen in their respective diagonals. Hence for each queen, we check if the absolute difference in their row coordinates is the same as their column coordinates. If it is the case, we increase the number of collision. For the initial case, we devide the result by two, because we count the same collision twice. For each swap, we compute the number of collision from these two queens before and after the swap.

Hence the new number of collisions is: current #collisions $-$#collisions before swap $+$#collisions after swap.

Our base chain is defined as follows:

$$\Psi_{xy} = \begin{cases} \frac{2}{n(n-1)}, & \text{if } x \sim y \\ 0, & \text{otherwise} \end{cases}$$

where we define $x \sim y$ to be a swap of two values in the permutation (e.g. $(02314) \to (02413)$). Hence it is irreducible and symmetric.

the acceptance probability is simplified due to the symmetric property of our base chain. We have

$$a_{xy} = \min(1, \frac{\pi_\beta(y)}{\pi_\beta(x)}) = \min(1, e^{-\beta(f(y)-f(x))})$$

which we can re-write as

$$a_{xy} = \begin{cases} 1, & \text{if } f(y) \leq f(x) \\ e^{-\beta(f(y)-f(x))}, & \text{if } f(y) > f(x) \end{cases}$$

where $\pi_\beta(x) = \frac{e^{-\beta f(x)}}{z_\beta}$ and $z_\beta = \sum_{x \in S} e^{-\beta f(x)}$

We can now compute the probability of our Metropolis chain

$$P_{xy} = \begin{cases} \frac{2}{n(n-1)} a_{xy}, & \text{if } x \sim y \\ \frac{2}{n(n-1)} \sum_{k \neq x} (1 - a_{xk}), & \text{if } x = y \\ 0, & \text{otherwise} \end{cases}$$

Here are some graphs showing the evolution of the number of collisions for $N = 100$ and $N = 1000$. It decreases rather quickly then it takes some time before finding a permutation when no queens collide each other.

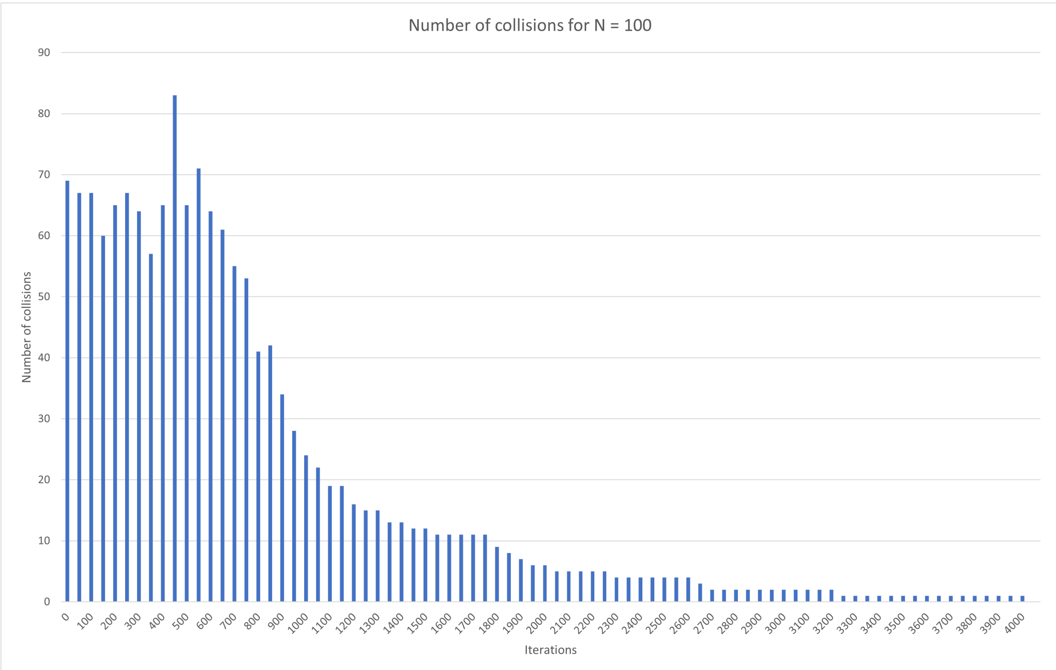

Figure 1: Evolution of number of collisions for $N = 100$

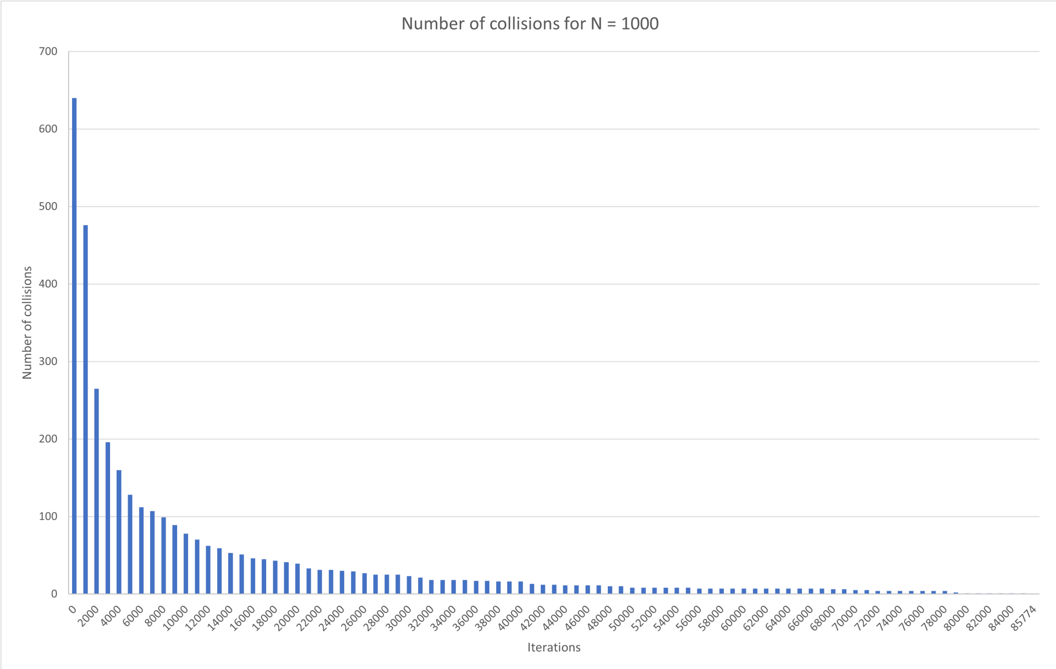

Figure 2: Evolution of number of collisions for $N = 1000$

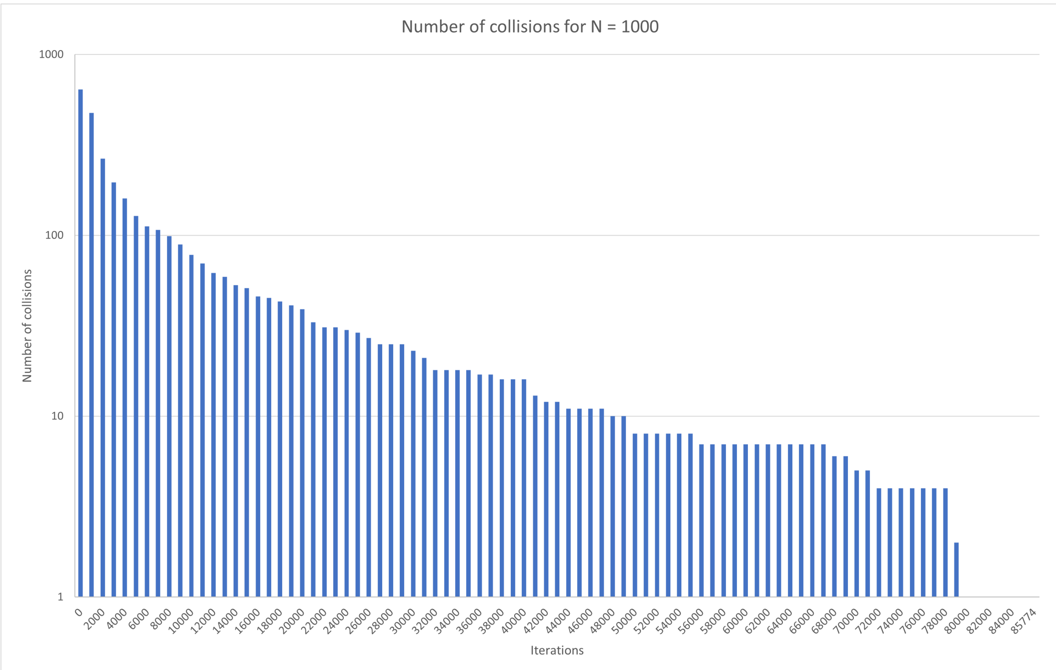

Figure 3: Evolution of number of collisions in logarithmic scale for $N = 1000$

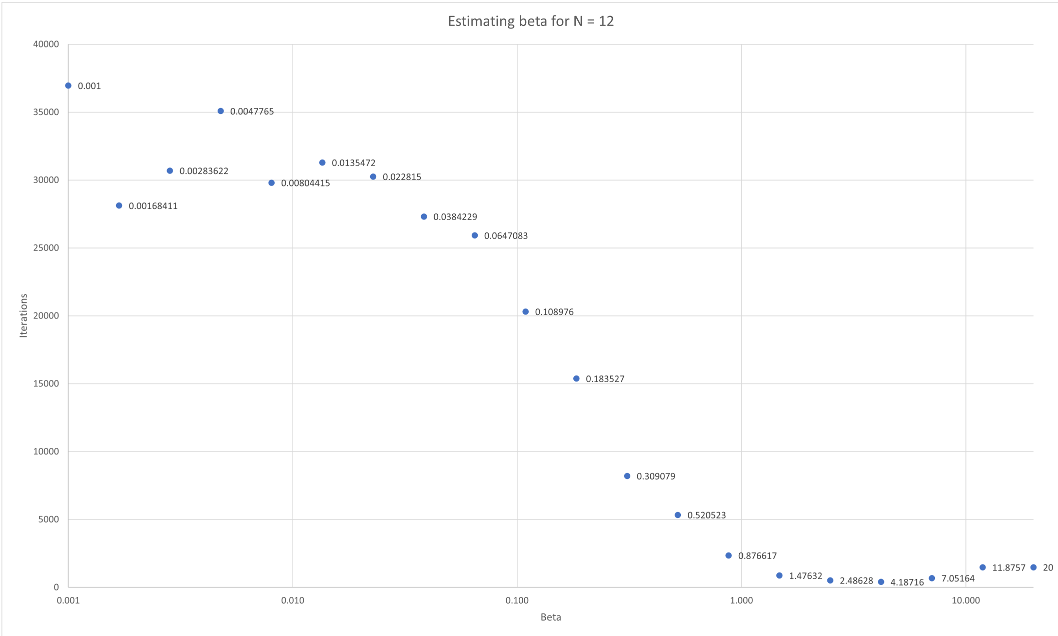

Figure 4: Estimating values of β for $N = 12$ given a geometric progression

2 Estimating number of solutions

To find a good β, we ran our algorithm for a given N with different values β and computed the mean of iterations needed to find a solution.

Values indicated next to the point on the graph of Figure 4 shows the value of β and not the number of iterations. We thought it was harder to estimate β on the axis due to the logarithmic scale. For this case, $\beta \approx 5$ find the solution in a smaller number of iterations compared to others.

Hence we will use $\beta^* = 5$ to estimate the number of solutions. With our algorithm, we found $16,925.50$ which is not too far from the real value (i.e. $14,200$).

The variance of Z_β is correlated to the number of samples m, the number of maximum iterations given to create each sample and the number of queens N. Unfortunately, when trying to find the number of solutions for N greater than 18, it gives a value way off what we expect.

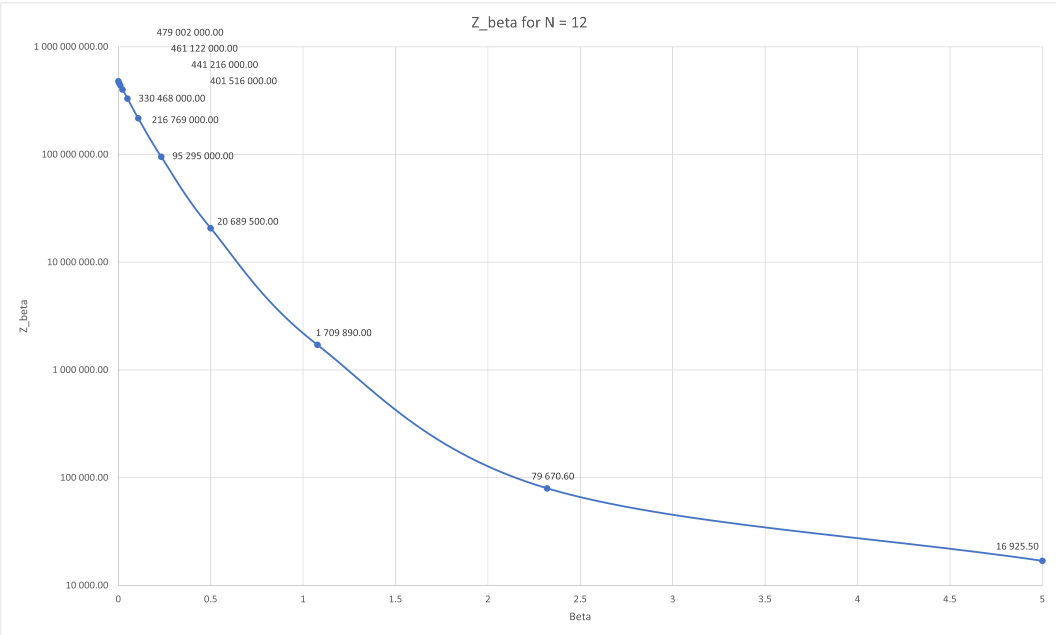

Figure 5: Estimating the number of solutions for $N = 12$