# Random Walks course project: Simulated annealing algorithm for graph coloring

The goal of the present project is to code and experiment the Markov Chain Monte Carlo (MCMC) method for the problem of *graph coloring*. One component of the project is to get familiar with the idea of *simulated annealing*, where the temperature parameter in the MCMC is lowered progressively.

This is an open project were you have to experiment in order to improve the performance of the MCMC.

## Formulation of graph coloring problem as an optimization problem

Consider a graph $G = (V, E)$ with vertex set $V = \{1, 2, ..., N\}$ and edge set $E \subset V \times V$, as well as an alphabet of colors $\{1, ..., q\}$ with $q \geq 3$. A *proper $q$-coloring of $G$* is a coloring of the graph such that no two adjacent vertices share the same color. In mathematical terms, this means that a proper $q$-coloring is a coloring $x = (x_v,\, v \in V) \in \{1, \dots, q\}^V$ such that $x_v \neq x_w$, $\forall (v, w) \in E$. Our aim in this project is to find a proper coloring of the graph via the MCMC method.

One way to reformulate this problem is as an optimization problem. Let $x = (x_v, v \in V)$ be any $q$-coloring (not necessarily proper). We define the following cost (or energy, or Hamiltonian) function

$$H(x) = \sum_{(v,w)\in E} \mathbf{1}_{x_v = x_w} \tag{1}$$

This function assigns a unit cost to every edge having two identical colors; it therefore counts the total number of such “bad” edges.

The goal of the MCMC method is to minimize this function and to see if there exist minimizers that yield zero cost. Minimizers with zero cost are proper $q$-colorings. On the other hand, if the global minimum is strictly positive, then there does not exist a proper coloring.

To minimize the function, we propose to use the simulated annealing method. For this reason, we introduce an auxiliary *finite temperature* version of the problem. Consider the *Gibbs-Boltzmann* probability distribution

$$p(x) = \frac{e^{-\beta H(x)}}{Z_\beta} \tag{2}$$

where $\beta = 1/T$ is the *inverse temperature* and $Z_\beta = \sum_{x \in \{1,\dots,q\}^V} e^{-\beta H(x)}$ is the normalizing factor (also called the *partition function*).

The idea described below is to construct a Markov chain that samples correctly from this distribution. As we make the parameter $\beta \to +\infty$ (equivalently $T \to 0$), the obtained sample should converge to a coloring that minimizes the above cost function (1).

## Metropolis chain

For the moment, think of $\beta$ fixed. The simplest possible Metropolis algorithm is:

1. At time $t = 0$, initialize with a coloring $x^0$ taken uniformly at random.
2. At times $t \geq 0$, make a transition $x^t \to x^{t+1}$ according to the following rules:
   - Select a vertex $v \in V$ uniformly at random, with current color $x_v$. Pick a color different from $x_v$ at random and recolor the vertex $v$. Consider the new coloring $x^{\text{new}}$ and compute $\Delta = H(x^{\text{new}}) - H(x^t)$
   - if $\Delta \leq 0$, then accept the new color (with probability one) and set $x^{t+1} = x^{\text{new}}$.
   - if $\Delta > 0$, then accept the new color with probability

     $$\frac{p(H(x^{\text{new}}))}{p(H(x^t))} = \exp(-\beta\Delta)$$

     and set $x^{t+1} = x^{\text{new}}$; reject the move with the complementary probability and set $x^{t+1} = x^t$.
3. Iterate for $n$ time steps.

In the application of the graph defined below, you will have to observe/track (plot) the evolution of the cost with time: $H(x^t)$.

## Simulated annealing

The main idea in simulated annealing is to lower the temperature as time progresses during the Markov chain. Note that the transition probabilities change with time now, so we are not anymore in the framework of time-homogeneous chains.

You will have to experiment to find a suitable schedule for lowering $T$ (resp., increasing $\beta$). If you lower $T$ too slowly, you might not reach the minimizers of (1) in a decent amount of time. If you lower $T$ too quickly, you lose the advantage of the Metropolis step and might end up in a local minimum with strictly positive cost.

One (basic) advice is not to change the temperature at every iteration step, but rather to keep it constant for some steps before lowering the temperature.

## Experiments on random graphs

You will experiment simulated annealing on random Erdös-Rényi graphs. These graphs are generated as follows:

1 Take $N$ independent vertices ($N$ large).

2 For every pair of vertices: put an edge with probability $c/N$ (and otherwise do not put an edge), where $c$ is some constant greater than 1.

Store then the graph as an adjacency $N \times N$ matrix.

The parameter $c$ is called the density of the graph because the number of edges is concentrated on $\frac{c}{2}N$. The degree distributon of the vertices is a random Poisson variable with mean $c$ (so $c$ is also the average degree of a node). There are a few isolated vertices of degree zero (so these can be trivially colored) but one can show that with high probability (with respect to $N$) there is one giant connected component in the graph as soon as $c > 1$.

## Your tasks

1. Form a team of 3 to complete this project, choose a team name and send it to us by email by **Wednesday, May 11.**
2. Write a code that takes as input the adjacency matrix of a graph with $N$ vertices and aims to find a $q$-coloring of the graph with the method described above (your code should manage values of $N$ up to $1'000$ in a reasonable amount of time). *Recommended:* use Matlab for this.
3. Run your algorithm on the Erdös-Renyi random graph described above, and tune the parameters of your algorithm in order to optimize its performance. Plot the curve $H(x^t)$ as a function of time for some parameter values $q$ and $c$.
4. For given values of $q$ and $c$, let $H_{\min}(q, c)$ denote the cost of the coloring found by your algorithm, which is supposedly close to optimal. Plot $H_{\min}$ as a function of $c$ for three different values of $q$ (say $q = 3, 5, 7$). What do you observe?
5. Send us a small report (2-3 pages) as well as your code by **Wednesday, May 25**.
6. Get ready for the competition on **June 1st, 8:15 AM !** (details to come later)