Simulated annealing algorithm for graph coloring

Xavier Fontaine, Thomas Grivaz, Antoine Mougeot

May 24, 2016

Abstract

This report summarizes our methodology and results in the context of experimenting the Markov Chain Monte Carlo method for the problem of graph coloring.

1 Introduction

TODO

2 Metropolis chain

In this section we run experiments with β fixed and we apply the Metropolis-Hastings algorithm as described in the project statement. We analyze the influence of the initial value of β on the minimal value of the Hamiltonian reached by our algorithm. We can think of β as the inverse of a temperature and therefore we note $T = \dfrac{1}{\beta}$.

In these experiments we fix an Erdös-Renyi graph G with parameters $N = 100$ and $c = 20$. We use $d = 5$ colors. We test three different values of temperature: $T = 10$, $T = 1$ and $T = 0.1$ and we run the simulations with 3000 iterations. Here are the curves we obtain:

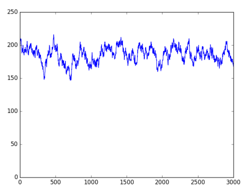

Figure 1: $T = 10.0$ and $\min H = 147$

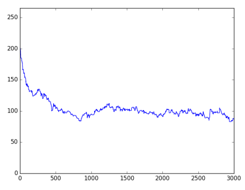

Figure 2: $T = 1.0$ and $\min H = 83$

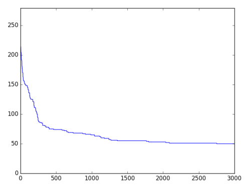

Figure 3: $T = 0.1$ and $\min H = 50$

- When T is high (i.e. β is small), the probability to accept a move that increases the Hamiltonian is high (for $T = 10.0$ it is roughly 80%) and consequently the values of H vary a lot. Therefore we observe a lot of fluctuations and the chain does not converge into a precise state. Moreover, the final Hamiltonian value is not equal to the minimal Hamiltonian value. We observe also that since β is small, π_β is far away from π_∞ and then the minimal Hamiltonian value that we reach is really bigger than the global minimum we could reach. With the graph we used and these settings we obtain an average minimal value of 153 for H.

- With an intermediate value of T ($T = 1.0$ here) the previous phenomenon is less visible. The probability to accept a move increasing the Hamiltonian is indeed smaller (about 10% now) and therefore we observe a global decrease of H. However we can still see some fluctuations but the final Hamiltonian value is near from the minimal Hamiltonian value reached during the 3000 iterations. The distribution π_β is still far from π_∞ and therefore the minimal Hamiltonian value we obtain here (its average is 80) is still much bigger than the global minimum.
- When T is low, the probability to increase H is very low and therefore we obtain a function $t \mapsto H(x^t)$ that is non-increasing. Now the distribution π_β is closer to π_∞ but the probability to get stuck in a local minimum is really high. Therefore the final Hamiltonian value is fast always equal to the minimal Hamiltonian value obtain during the iterations because the chain does not manage to leave this local minimum. However we obtain smaller values for H ($H = 50$ in our case on average).

Therefore we can conclude that when T is fixed we have to choose a low value for T in order to obtain small values for the Hamiltonian. The drawback of this method is that we are likely to get stuck in a local minimum.

In order to obtain better results we will use the method of *Simulated Annealing* which decreases the value of T during the iterations.

3 Annealing Schedule

In this section we will describe how we chose and tuned the parameters of our algorithm. In order to minimize the bias we could have regarding the graph we're trying to color, several graphs were considered for each value of a parameter: one random graph with $N = 100$, $c = 5$ and $q = 3$ (which was the graph provided on the project webpage, that we call G_1), one random graph with $N = 200$, $c = 40$, $q = 7$ (G_2), one random graph with $N = 50$, $c = 10$, $q = 3$ (G_3) and finally a random graph with $N = 100$, $c = 30$, $q = 7$ (G_4). Besides G_1, we chose graphs with high edge probability that result in high final energy after having run the Metropolis algorithm so that's it's easier to notice differences between choices of parameters.

3.1 Initial Temperature

According to Kirkpatrick [?], a suitable temperature T_0 is one that results in an average increase of acceptance probability p_0 of about 0.8. The value of T_0 depends on the scaling of our cost function and hence is problem specific. To estimate this, we conducted an initial search on each graph where all increase are accepted and calculated the average increase over a fixed number of iterations, the initial temperature is given by :

$$T_0 = -\frac{\overline{\triangle_+}}{\ln(p_0)}$$

Where $\triangle_+ = H(x^{new}) - H(x^t)$ is a strictly positive increase. We also tried different base acceptance probabilities (0.5, 0.3) and several fixed values of T_0, but this method gave us the best overall results

3.2 Cooling Function

Several ways of decreasing the temperature were considered. We first tried an exponential schedule, defined as:

$$T(t) = T_0\alpha^t$$

With $\alpha \in [0.7, 0.95]$ with a step of 0.01. A linear schedule was also considered:

$$T(t) = T_0 - \eta t$$

With $\eta \in [0.05, 0.4]$ with a step of 0.05.
For each value of α or η, we ran Metropolis on our 4 graphs and averaged the final energy obtained, we then kept the value for which the final energy was the minimum and ran the whole process again several times to make sure that it was indeed the best overall parameter. Note that even though differences in results were very small (the differences in energy between parameters were in the 5% range), we noticed that the optimal parameter was consistent from one run to another, for example the best value for the exponential schedule was $\alpha = 0.85$, and it was the best in 80% of cases.

With the best value we obtained for each scheme ($\alpha = 0.85$, $\eta = 0.25$), we compared the two by plotting the evolution of $H(x^t)$ for both schemes: We also

include plot of H with expo and linear schedule

tried several epoch lengths (decrease temperature every x iterations)