The Onset of Difficulty in A Variant of Graph Coloring and the Jamming Transition

Ronnie Garduño * †1 and Cris Moore‡1,2

[1]Department of Computer Science, University of New Mexico, Albuquerque, NM 87106 USA,
[2]Santa Fe Institute, Santa Fe, NM 87501 USA

August 16, 2011

Abstract

We describe a variant of the well-known graph coloring problem (COL) which we are calling Continuous K-coloring (CONT-COL). This variant has certain properties that make it more amenable to measurement of the computational effort required to find solutions to instances of the problem. We describe the experiments to measure this which we are designing as well. The algorithm used for this purpose is based on the physics algorithm known as the event chain algorithm, which works well with the measures of computational effort we intend to use. These experiments are intended to explore a tentative analogy between the difficulty of various instances of the graph coloring problem and the jamming transition in granular materials from physics, which analogy could add to our understanding of such systems.

*rgarduno@unm.edu
†Funded by NSF REU grant
‡moore@cs.unm.edu

Introduction

The jamming transition is a phenomena which has received a lot of attention of late in the physics communities. It has applications in glass formation, granular material behavior, and many other systems which are well modeled as disorganized arrangements of homogenous particles. [3] Broadly speaking, the jamming transition refers to the tendency of such systems to undergo a phase transition from behaving more like a liquid to behaving more like a solid at a particular critical density. We would like to explore this phenomena more generally and see if it occurs in other theoretical settings. More specifically, we have developed a simple variant of the 3-Coloring problem on graphs (*3COL*) we are calling continuous 3-coloring (*CONT-3COL*). This variant of 3-coloring differs from the original problem by using a different set of labels and criteria for the validity of these labels.

Continuous 3-Coloring

In the *3COL* problem, an assignment of labels ('colors') to the vertices of a graph is a valid one if and only if no two vertices which share an edge also share the same label. Rather than using discrete labels like this, *CONT-3COL* uses real numbers between zero and 2π, which can be thought of as angles on the unit circle modulo 2π. These angles, then, form a valid continuous 3-coloring of the graph if every pair of adjacent vertices have angles which are at least $2\pi/3$ radians away from each other. There is a more general formulation of this problem called continuous q-coloring (*CONT-COL*), but we will leave further explorations of this problem to a later date. The reason for using CONT-3COL for our experiments relates to the algorithm we are working with. This is one of a class of MCMC algorithms called the event chain algorithm [1].

Event Chain Algorithm

The event chain algorithm was designed to simulate motion in a disordered system of hard spheres (or disks). Briefly, this algorithm operates by selecting a sphere at random and displacing it in a direction, also randomly chosen. In the version called the Straight Event Chain, this direction remains constant over a given chain, which I am using as a model for my algorithm. There is also a displacement budget associated with each of the chains, which remains constant over each run of the algorithm and is chosen as a parameter beforehand. The sphere being displaced moves in the chosen direction until it either runs out of displacement by moving far enough, or until it strikes another sphere. In the latter case, the second sphere is chosen to move in the same direction in the same way, and so on until the displacement budget is exhausted. This procedure does not reject any proposed moves, and so it is highly effective, operating in a more efficient manner than the usual Metropolis-Hastings algorithm used for this purpose. What makes this algorithm useful for my purposes is its interaction with the CONT-COL problem previously discussed. A big problem in trying to analyse the amount of computational effort required to solve instances of COL comes from the discrete nature of the labels used in this problem. If we are modifying a 3-colored graph by adding a new edge to the graph which makes the old coloring invalid, we will need to alter the color of at least one of the vertices connected by the new edge. However, the new color chosen for this vertex may conflict with more than one of its neighbors, making for a branching-off in this search space that could lead to inconsistent counts of the number of search nodes needed to find a solution. The same does not hold for continuous coloring due to the continuous nature of the labels used.

Because there are so many possible angles for each vertex, it is highly unlikely that two of a vertex's neighbors will have the same angle. This means that if we are smoothly rotating the angle of that vertex, it is quite likely that it will strike one of its neighbors before any of the others. This should be true for any pair of vertices in CONT-3COL, making an event chain algorithm an efficient and effective way to correct colorings in a consistent amount of work.

Phase Transition in 3COL

Thanks to this property of the CONT-COL problem, it should be much easier to measure the work involved in coloring large numbers of random graphs and get results which compare in a meaningful way with other problems the event chain is run on. This means that we should be able to run comparable experiments with the event chain algorithm on both CONT-3COL instances and on the simulation of systems near the jamming transition. Our conjecture for this approach is that the algorithm will have similar results in both cases, providing an empirical basis for comparing the difficulty of correcting almost-valid CONT-3COL instances and moving spheres in a disordered system at or near the jamming transition. It has been determined through empirical and analytic means (i.e, [4]) that there exists a phase transition in the probability of being able to color random graphs at different edge densities. At a density of approximately 4.69, the probability of random graphs being 3-colorable drops from almost 1 to almost 0. Above this phase transition, random graphs become almost always uncolorable, whereas they are almost certainly colorable below it. A common conjecture in the field, which we adopt for this work, is that there exists a small range of densities below this phase transition which leave random graphs possible to color, but whose colorings are difficult to find. There have been several studies which suggest that this conjecture is true (cite), but none that have proved it yet. We intend to explore this region of state space with an event chain algorithm to see how comparable this increase in difficulty is to the increase in the difficulty of fitting more components into a system near the jamming transition.

Coloring Algorithm

This algorithm operates in the following manner. It first starts with a continuously colored random graph, also built up using this algorithm. It then add a random edge to this graph, which will usually make the old coloring invalid. At this point, one of the two vertices connected by the new edge is selected for modification by the algorithm, along with a random choice of clockwise or counter-clockwise for the new chain's direction of rotation. At this time, a full rotation budget for the chain, chosen as a parameter to the algorithm, is allocated as well. The chosen vertex is rotated until it has used up its rotation budget or until it strikes its nearest neighbor's territory, which is $2\pi/3$ radians away from the neighbor's angle in the CONT-3COL case we are exploring at present. The amount of rotation acheived by this move is then subtracted from the chain's rotation budget. If this budget has been depleted, the chain ends here. If the new angles form a valid continuous coloring on the graph with the new edge, the algorithm has completed its task and halts. Otherwise, a new chain is started in the same manner as previously described. If the previous move has not exhausted the budget of rotations, the algorithm continues by selecting the neighbor which the first vertex collided into and rotates it in the chain's overall direction in the same fashion as before. In this way, our event chain algorithm should be able to correct the coloring of any colored random graph we have perturbed by adding an edge as previously mentioned. Furthermore, as this is performed we can gather various statistics about the coloring of graphs of different densities and sizes, such as the average number of chains needed, the average number of vertices altered by each correction, and so on. These will form the output data from our experiments, which will then be compared with similar statistics from the operation of the event chain algorithm on systems near jamming.

Future Work

Much work still needs to be done in this investigation. The Mathematica code written to implement the coloring algorithm still has a few bugs in it which need fixed. When that is complete, it may turn out that Mathematica is too slow for the experiments to be run in a reasonable amount of time, at which point I would need to port the code to something faster, like Ruby or C. When the experiments have finally been completed, I will need to analyze the resulting data to see whether or not our conjecture about the relationship to jamming is supported. This analysis may prove to be somewhat complex. There have been studies claiming that almost all k-colorable graphs are easy to color [5]. However, some recent work [2] indicates that this may be due to the low number of edges in a random graph which actually affect the colorability of the graph, necessitating a reduction procedure in order to explore the difficulty of coloring these graphs in a meaningful way. Because of this and other considerations, it may also be fruitful to find a formal relationship between COL and CONT-COL, or perhaps between CONT-COL and some other problem.

References

[1] Etienne P. Bernard, Werner Krauth, and David B. Wilson. Event-chain algorithms for hard-sphere systems. *Physical Review E*, 80(5), 2009.

[2] Joseph Culberson and Ian Gent. Frozen development in graph coloring. Technical Report 19, APES Research Group, www.apes.cs.strath.ac.uk, February 2000.

[3] Andrea J. Liu, Sidney R. Nagel, Wim van Saarloos, and Matthieu Wylart. The jamming scenario— an introduction and outlook. *eprint arXiv*, 1006(2365), 2010.

[4] R. Mulet, A. Pagnani, M. Weigt, and R. Zecchina. Coloring random graphs. *Physical Review Letters*, 89(26):4, December 2002.

[5] Jonathan S. Turner. Almost all k-colorable graphs are easy to color. *Journal of Algorithms*, 1988.