Coloring Random Graphs and Maximizing Local Diversity

S. Bounkong, J. van Mourik, and D. Saad
Neural Computing Research Group, Aston University, Birmingham B4 7ET, UK
(Dated: February 2, 2006)

We study a variation of the graph coloring problem on random graphs of finite average connectivity. Given the number of colors, we aim to maximize the number of different colors at neighboring vertices (i.e. one edge distance) of any vertex. Two efficient algorithms, belief propagation and Walksat are adapted to carry out this task. We present experimental results based on two types of random graphs for different system sizes and identify the critical value of the connectivity for the algorithms to find a perfect solution. The problem and the suggested algorithms have practical relevance since various applications, such as distributed storage, can be mapped onto this problem.

PACS numbers: 89.75.-k, 02.60.Pn, 75.10.Nr

I. INTRODUCTION

The graph coloring problem [1, 2] has received a significant level of attention. Much of this interest stems from the fact many real-world optimization problems can be represented as coloring problems. In the original formulation, given q colors, one aims at finding a coloring solution such that any two connected vertices have different colors. In the current work, the aim is to maximize the number of colors at one edge distance of any vertex.

One application can be found in the field of logistics, where each vertex represents a storage unit. The problem is then to find how to distribute the different types of goods such that, at each site, any type can be retrieved either from the given unit or from directly adjacent storage units. The problem that got us interested in this problem is that of distributed data storage where files are divided to a number of segments, which are then distributed over the graph representing the network. Nodes requesting a particular file collect the required number of file segments from neighboring nodes to retrieve the original information. Distributed storage is used in many real world applications such as OceanStore [3].

It should be emphasized that typical properties of the main problem we are interested in should be taken into account when a color assignment algorithm is considered: 1) The problem is characterized by a finite number (of the order of the graph connectivity) of different file segments. 2) An adaptive assignment of colors may be required as the topology continuously changes due to the emergence and disappearance of nodes. 3) The networks considered are of moderate size, 10^2-10^3 nodes.

Although this problem has not yet been shown to be NP-complete, it seems nonetheless intractable for a large system size. Since no research has been carried out on this specific problem, no dedicated tools exist either[4]. However, as we report in this paper, existing optimization algorithms can be adapted quite easily to solve this and similar problems. In particular, we investigate two well established techniques: belief propagation (BP) and a variant Walksat (WSAT) for this purpose.

In this paper, we show how BP and Walksat can be used to solve this particular problem. For a given number of colors q we identify the transition points in terms of the critical connectivity λ_c^q above which the algorithms typically find a perfect coloring. We also calculate the average minimum measure of unsatisfaction $E^q(\lambda)$ as a function of the connectivity λ. The latter is defined as $E^q(\lambda) = \sum_{i=1}^n E_i^q(\lambda)$ where for each vertex i with local connectivity λ_i

$$E_i^q(\lambda) = \min(q, \lambda_i + 1) - q_i \tag{1}$$

is the difference between the number of actually available colors at that node q_i, and the maximal number of available colors (at the vertex and its nearest neighbors $\min(q, \lambda_i + 1)$). In this paper, we only consider graphs with local connectivities $\lambda_i \geq q - 1$, such that $E_i^q(\lambda) = q - q_i$ just counts the number of missing colors. One should note that, contrary to the original graph coloring problem, the problem of finding a coloring for our problem actually becomes easier with increasing connectivity.

The main goal of this paper is to introduce the problem, and investigate the performance of the two algorithms on realistic system sizes. The full analysis of the model in the infinite system size is a separate issue that is currently being investigated.

II. THE ALGORITHMS

Belief propagation: BP, also called the sum-product algorithm, relies on iterative message passing to provide near optimal performance at low computational cost [6, 7]. It is based on conditional probabilistic messages passed from the immediate neighborhood to find the most probable assignment of states to variables given constraints. In our problem, the constraints correspond to the retrieval of $\min(q, \lambda_i + 1)$ colors per vertex, from the vertex itself and its first order neighbors.

These constraints can be represented by clusters of vertices on a graph as in Fig. 1a, where 'A', 'B_1', 'B_2', 'C_{11}' and 'C_{12}' correspond to the vertices, while 'Z_A' and '$\mathrm{Z}_{\mathrm{B}_1}$' are check variables corresponding to the constraints. Checks variables relate to the unsatisfaction of

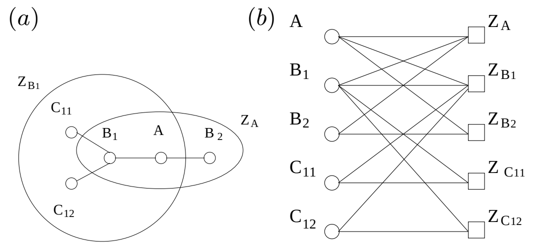

FIG. 1: (a) A graph representing the successive local constraints. (b) A bi-partite graph representation.

given color assignments for the corresponding node and its direct neighbors. For instance, for the node A with q_A available colors ($\beta = 20$ for computational reasons)

$$P(Z_A|A, \{B_i\}) = e^{-\beta(q-q_A)} \; . \tag{2}$$

The graph can be transformed into a bipartite graph, shown in Fig. 1b, which separates the vertices from the checks. The following update rules can be easily obtained by naively adapting the original BP rules [6, 7]. Thus, the messages from a check to a vertex are given by

$$\begin{aligned} P(Z_A|A) &= \sum_{\{B_i\}} P(Z_A|A, \{B_i\}) P(\{B_i\}|\{Z_{B_i}\}, \{Z_{C_{i\cdot}}\}) \; , \\ &\simeq \sum_{\{B_i\}} P(Z_A|A, \{B_i\}) \prod_i P(B_i|Z_{B_i}, \{Z_{C_{i\cdot}}\}) \; , \end{aligned} \tag{3}$$

while the message from a vertex to a check is given by

$$P(A|\{Z_{B_i}\}) = \alpha_A \prod_{\{Z_{B_i}\}} P(Z_{B_i}|A) \; , \tag{4}$$

Finally, the node pseudo-posterior is given by

$$P(A|Z_A, \{Z_{B_i}\}) = \alpha P(Z_A|A) \prod_{\{Z_{B_i}\}} P(Z_{B_i}|A) \, , \tag{5}$$

where α_A and α are normalization coefficients. Note that the factorization in (3) is a relatively crude approximation even in the large system limit, as the $\{B_i\}$ nodes are correlated. To deal with this properly, a more advanced analysis using a cluster expansion [8] is currently been undertaken. Nevertheless, as we will see these approximations work remarkably well.

If convergence of the BP algorithm is reached, the colors of vertices whose (pseudo) posterior is greater than a pre-defined threshold set at 0.9 in our experiments can be fixed. If no such high posterior exists then the vertex with the highest posterior value has its color fixed. Then, the update rules are re-iterated and the decimation process repeated until a global coloring is reached.

A major drawback of the BP algorithm is that convergence is not guaranteed for graphs with loops due to fragmentation of the solution space. Random initialization results in the emergence of competing local solutions and conflicting messages, leading to non convergence.

Time averaging [9] is a way of getting around the problem by carrying out the decimation and color fixing process according to the average posterior (over time i.e. a number of iterations) instead of instantaneous posterior. In the case of non-convergence, this method decimates the vertex with the strongest *average* coloring probability over all competing solutions and thus reduce the fluctuations due to the competition. After several trials, a time window of 30 iterations was chosen for all numerical data presented here. We have opted for BP combined with time averaging due to its improved performance and robustness (e.g., in the distributed storage application nodes may suddenly switch off and on).

Walksat: Walksat is a local search algorithm, originally designed to maximize the number of satisfiable clauses in problems that assume a conjunctive normal form [10]. Although Walksat may seem to be suboptimal at first sight, studies have shown it to be a powerful tool [11]. Many variants of the original algorithm exist [10, 12, 13]. In this study, we have adapted the variant referred to as SKC; it uses the notion of variable *breakcount*, defined as the number of clauses that are currently satisfied, but would become unsatisfied if the variable assignment were to be changed. The SKC variable selection is as follows:

1. If there are variables with breakcount equal to 0, randomly select one such variable.
2. Otherwise
   - with probability p randomly select a variable.
   - with probability $1-p$ randomly select a variable with minimal breakcount.
3. Flip the selected variable.
4. Repeat until all clauses are satisfied or until the max-iterations is reached.

In our problem, the *breakcount* of a variable is given by the number of vertices for which the change of assignment would decrease q_i. Henceforth, the *breakcount* depends on the replacement color. In step (1) of the SKC procedure, the selected replacement color is the one which leads to a *breakcount* equal to 0 (if more than one, choose randomly). In the step (3), a replacement color is selected at random. In our first few attempts, this adaptation of the Walksat algorithm showed mixed results, which were up to 50% worse than those obtained with BP. We therefore adapted another local search algorithm [14] related to Walksat. This algorithm is also iterative and based on a mixture of gradient and "*noisy*" descent. At each iteration, one of these two descents is chosen at random, with some probability. Similarly to the Walksat algorithm, this step is repeated until all checks are satisfied, or the maximal number of iterations is reached.

The gradient descent is operated by the GSAT algorithm [15], which changes at each an iteration the variable assignment that leads to the greatest decrease in the number of unsatisfied clauses. In our problem, changes will correspond to the greatest decrease in unsatisfaction

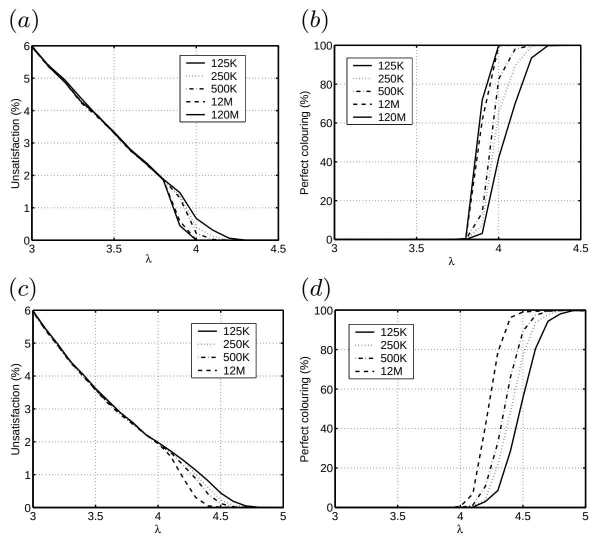

FIG. 2: Walksat performance on linear and Poissonian graphs (n=100) for *nbit* from 125K to 120M iterations and connectivity λ. (a) Unsatisfaction measure - linear. (b) Percentage of perfect coloring - linear.(c) Unsatisfaction measure - Poissonian. (d) Percentage of perfect coloring - Poissonian.

as defined in (1). "*Noisy*" moves in the original algorithm [14] are replaced here by the SKC heuristic. The resulting algorithm is a mixture between SKC and GSAT, which is parameterized by a probability p_m, set to 0.5. Our experiments show that the combined algorithm performs significantly better than SKC with no added cost.

If not all checks are satisfied at the maximal number of iterations (the choice of which is discussed in the next section) the GSAT algorithm is iterated until a local minimum is reached. The combined algorithm, referred to here as Walksat, shows similar results to those obtained using BP both in terms of unsatisfaction $E^q(\lambda)$ and the number of perfect coloring solutions found.

III. SIMULATIONS

Experiments were carried out for $q=4$ colors for two system sizes (n) of 100 and 1000 vertices. The two types of graphs studied have an average connectivity λ:

- *(cut-) Poissonian:* where vertices have local connectivities λ_i given by

$$\lambda_i = \lambda_{\min} + z_{\lambda-\lambda_{\min}} = q-1+z_{\lambda-q+1} \ , \qquad (6)$$

  where $z_{\lambda-q+1}$ is randomly drawn from a Poisson distribution with parameter $\lambda-q+1$.

- *Linear:* where vertices have local connectivities

$$\lambda_i = \lfloor\lambda\rfloor + z_{\lambda-\lfloor\lambda\rfloor} \qquad (7)$$

  where $\lfloor\lambda\rfloor$ is the largest integer smaller or equal to λ, and $z_{\lambda-\lfloor\lambda\rfloor}=1$ with probability $\lambda-\lfloor\lambda\rfloor$ and 0 otherwise.

We study the most interesting range of average connectivities from $\lambda=3$ to $\lambda=5$ with a step of 0.1. For each λ, 1000 graphs of each type were randomly generated and then colored by both the BP and Walksat algorithms.

Graph characteristics: Both graphs and constraints are born from the original problem we have set to solve, namely distributed storage. We point out two observations that may help in getting insight into the characteristics of the problem and solutions found by the algorithms: 1) The number of checks is always equal to the number of vertices as each vertex is associated with a check, that connects it to all vertices at one edge distance. This check is obeyed when the vertex can retrieve all possible colors from vertices at one edge distance. 2) Edges are undirected: if vertex 'B' is connected to the check of 'A', then vertex 'A' is also connected to the check of 'B'. Hence, there are always $\frac{n\lambda}{2}$ short loops, which correspond to the number of edges, in the belief network even in the large system limit. When the connectivity value λ increases, the number of loops increases as well, but it also becomes easier to get a lower value for the average unsatisfaction. Therefore, it is unclear whether the influence of the presence of loops on the performance of the (current) BP algorithm will increase or decrease with λ.

Walksat performance: In the Walksat algorithm, the maximal number of iterations *nbit* is an important parameter. A greater value increases performance, but also computational cost. Unfortunately, the relation between performance and cost is not linear and it is therefore difficult to estimate the optimal number of iterations. In order to understand this relation, we carried out several simulations with different values of *nbit* for the two systems sizes and all connectivity values.

Figure 2 show the results obtained for a system size n=100 and a range of limits on the number of iterations. One notices that improvements in terms of unsatisfaction and perfect coloring are negligible for $\lambda \leq 3.8$ and $\lambda \leq 4.1$ in linear and Poissonian graphs, respectively. In these regions, no perfect solutions are found and the Walksat algorithm stops when the maximum number of iterations is reached and returns a suboptimal solution, even for larger *nbit* values. Perfect coloring solutions exist and are found for $\lambda \gtrsim 3.8$ and $\lambda \gtrsim 4.1$ in linear and Poissonian graphs, respectively. However, a larger number of vertices will also require an exponentially larger number of iterations to achieve the same performance.

To compare the performance of the Walksat and BP algorithms, we take the results achieved by Walksat for roughly the same computational time to the one used by BP. This means $nbit=500K$ and $nbit=12M$ iterations for systems sizes of 100 and 1000 vertices, respectively. We also modify the Walksat algorithm described in Sec. II such that the unsatisfaction returned is the lowest value over all examined color assignments and not the one corresponding to the nearest local minimum.

BP vs. Walksat - a comparison: Figure 3 shows that for small graphs (100 nodes), and far away from the critical connectivity, BP is generally outperformed by the

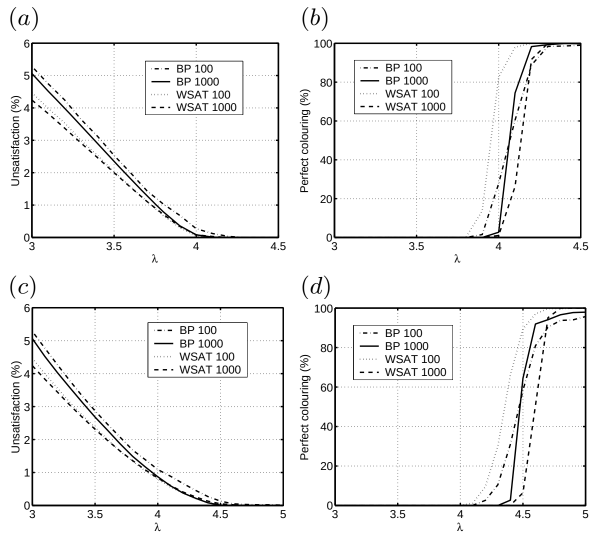

FIG. 3: Comparison of BP and Walksat algorithms for linear and Poissonian graphs. (a) Unsatisfaction measure - linear.(b) Percentage of perfect coloring- linear. (c) Unsatisfaction measure - Poissonian. (d) Percentage of perfect coloring - Poissonian.

Walksat algorithm. We believe this is partially due to the presence of small loops, as discussed earlier, and the use of generalized BP [16] is currently being investigated to improve performance; nevertheless one should note that even the approximative BP algorithm works surprisingly well considering the crude approximation made in (3). In addition, while Walksat clearly outperforms BP for 100 nodes systems, this is definitely not the case for 1000 nodes systems close to the critical connectivity, where results obtained by BP are better both in terms of the percentage of perfectly colored cases and in terms of the minimal average unsatisfaction, both for linear and Poissonian graphs. For increasing system sizes and given computing resources BP is likely to outperform Walksat in the relevant regions.

IV. DISCUSSION

We study a variation of graph coloring on random graphs with finite average connectivity, aimed at maximizing the number of colors accessible by a vertex within one edge distance. The problem is of practical relevance, especially in the area of distributed storage. The BP and Walksat algorithms were adapted to perform the task.

We present experimental results for two types of random graphs and system sizes, and identify critical connectivity values above which the algorithms find a perfect solution. For $q=4$ colors, the critical connectivities found are around 4 and 4.4 for linear and graphs and Poissonian graphs, respectively. In principle, the methods presented are applicable for random graphs of any connectivity profile and number of colors.

We have found that both algorithms give qualitatively very similar results with similar computing costs. The relative efficiency of both algorithms, in terms of the quality of obtained solutions and computing time, does however depend on the combination of parameters (λ, q, n) and graph characteristics. A more detailed analysis of this will be the subject of a separate study, as will be the thermodynamic phase diagram for this model. Further research will focus on improving the message passing approach by using the exact cluster expansion in the large system limit (i.e. focusing on stars and edges as our fundamental clusters instead of stars and nodes), combined with generalised BP, which will also provide us with a phase diagram for the model. It is assumed that this approach will remove the influence of short loops and therefore improve the performance of the algorithm, especially at low connectivity values.

[1] T R Jensen and B Toft, Graph Coloring Problems, Wiley Interscience (1995)

[2] J van Mourik and D Saad, Phys Rev E **66** 056120 (2002)

[3] J Kubiatowicz et al, OceanStore: An Extremely Wide-Area Storage System, UCB//CSD-00-1102 U.C. Berkeley (1999)

[4] It should be mentioned that different systems of similar topology have been investigated in [5]

[5] M Mézard and G Biroli Phys Rev Lett **88** 025501 (2002)

[6] J Pearl, Probabilistic Reasoning in Intelligent Systems Morgan Kauffmann Publishers (1988)

[7] D J C MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press (2003)

[8] G An, Jour of Stat Phys, **52** 727 (1988)

[9] J van Mourik, Time averaged belief propagation, in preperation (2005); B Fabre, Algorithms for coloring random graphs, MSc thesis, Aston University (2003); Y Fortune, Algorithms for solving optimization problem on large random graphs, MSc thesis, Aston University (2004).

[10] B Selman, H Kautz and B Cohen, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, **26** 521 (1996)

[11] E Aurell, U Gordon and S Kirkpatrick, NIPS 2005 **18** (2005)

[12] D McAllester, B Selman and H Kautz Proc. AAAI-97, 321 (1997)

[13] D J Patterson and H Kautz, Electronic Notes in Discrete Mathematics (ENDM)**9**, 1 (2001)

[14] B Selman, H Kautz and B Cohen Proc. AAAI-94, 337 (1994)

[15] B Selman, H J Levesque and D Mitchell, Proc. AAAI-92, 440 (1992)

[16] J S Yedidia, W T Freeman and Y Weiss, NIPS 2000 **13** 689 (2000)