Discrete Mathematics 74 (1989) 253–261
North-Holland

A RANDOMISED 3-COLOURING ALGORITHM

A.D. PETFORD
Department of Astrophysics, University of Oxford, U.K.

D.J.A. WELSH
Mathematical Institute, University of Oxford, U.K.

This paper describes a randomised algorithm for the NP-complete problem of 3-colouring the vertices of a graph. The method is based on a model of repulsion in interacting particle systems. Although it seems to work well on most random inputs there is a critical phenomenon apparent reminiscent of critical behaviour in other areas of statistical mechanics.

1. Antivoter models

In the original voter and antivoter models studied by Donnelly and Welsh [2, 3] the vertices of a graph G are either black or white and at random epochs of time a vertex, chosen at random, changes colour according to a specified stochastic mechanism. In the antivoter model each vertex has attached to it a clock which acts independently for each vertex. When the clock at vertex v rings, one of its neighbours, say u, is selected at random and v changes its colour so that it is different from that of u. Thus if u happened to be a different colour to v the colour of v would not change.

The process is a Markov process with state space consisting of all possible 2-colourings of the vertex set V of G.

In the antivoter model there are two possibilities depending on the graph G:

(a) if G is bipartite, the system is, with probability one, absorbed in one of the proper 2-colourings, or

(b) if G is not bipartite then the system continues to evolve without ever reaching an absorbing state.

Prompted by noticing the speed with which the antivoter model with 2 colours seemed to settle down to equilibrium we proceeded to try to extend the model to 3 colours.

The underlying combinatorial problem is now the 3-colouring problem which is known to be NP-complete. Hence the likelihood of being able to find a truly randomised algorithm analogous to the Rabin-Solovay-Strassen algorithm for primality testing is remote. The existence of such an algorithm would imply that random polynomial time (RP) equalled nondeterministic polynomial time (NP). This would be highly surprising unless, of course, NP turns out to equal p (see for example Welsh [10]).

With more than 2 colours the antivoter model has at least two natural

formulations of the way in which a vertex chooses its new colour once it has seen and been repelled by the colour of a randomly chosen neighbour. We proceed to define the general antivoter model with 3 colours as follows.

Consider a fixed instant of time t. For $1 \leqslant i \leqslant 3$ let $S_i(v)$ denote the set of neighbours of vertex v which are coloured i at t and let $s_i(v)$ be the cardinality of $S_i(v)$. When the clock at v rings, instead of v changing colour to that of some randomly chosen neighbour we stipulate that the new colour of v is a random colour X where

$$P(X=j)=p(s_1, s_2, s_3: j), \quad 1 \leqslant j \leqslant 3,$$

and where $s_i = s_i(v)$ and where p is the *transition function* satisfying the conditions

(1) $p(s_1, s_2, s_3: j) \geqslant 0$,

(2) $\sum_{j=1}^{3} p(s_1, s_2, s_3: j) = 1$.

Note. Here and throughout, where necessary, we shall identify the set of colours used with the set of integers $\{1, 2, 3\}$.

By suitably choosing p we get different versions of the antivoter model.

Example 1. Consider now the antivoter model having transition function given by

$$p(s_1, s_2, s_3: i) = \frac{1}{2}\left(1 - \frac{s_i}{(s_1+s_2+s_3)}\right), \quad 1 \leqslant i \leqslant 3.$$

This represents the situation where if the clock at v rings, a neighbour u is chosen at random and vertex v then chooses a colour at random from the two colours which are not the colour of u. In other words it is repelled by u's colour but otherwise chooses independently.

2. The randomised algorithm

The basic idea of our algorithm for deciding if a graph is 3-colourable is as follows:

(1) Colour the vertices arbitrarily with 3-colours

(2) Allow the antivoter mechanism with transition function p to operate on G for a time $t(n)$ where n is the number of vertices and t is the *threshold function*

(3) After time $t(n)$ announce the graph as 3-colourable if a proper 3-colouring has been achieved and as not 3-colourable otherwise.

Clearly the algorithm has the following properties.

(a) If it says G is 3-colourable then it is correct

(b) If it says G is not 3-colourable it may be incorrect.

The probability of an error depends on G, and the choice of threshold function t, and on the choice of transition function p.

The object of the simulation exercise is to

(i) find a 'good' transition function p

(ii) for this good transition function find a threshold function t in which we can have a fair degree of confidence.

Although the physical models studied in [2] and [3] work in continuous time, there is no advantage in sticking to this in using these ideas for developing an algorithm. Accordingly we stipulate that the "clock ringing random mechanism" works as follows:

The Clock Ringing Mechanism: A vertex is *bad* at time t if it has a colour which is the same as any one of its neighbours. B_t denotes the set of bad vertices at time t and as time progresses through $t = 0, 1, 2, \ldots$, the clock chosen to ring at time $t+1$ is a random member of B_t. Initially we start with a random colouring.

As soon as B_t becomes empty the system stops.

We measure the *time* taken by the process to be the number of times a random member is chosen from the set of bad vertices. We denote it by $T(G)$ and it is clearly a random variable whose distribution depends on G and the transition function p.

For any non-trivial p and G it is clear that $T = T(G)$ has a non-zero probability of taking arbitrarily large values and the real quantities of interest are the first moment of T and the size of the tail distribution.

As a test of the proposed algorithm we introduce the concept of a *random benchmark* for the problem, and thus evaluate its performance on 'typical' or 'random' 3-colourable graphs. We construct these graphs in the following way.

Choose integers k_1, k_2, k_3 with $(k_1 + k_2 + k_3 = n)$ and take disjoint sets V_1, V_2, V_3 with $|V_i| = k_i$. Take the vertex set of the benchmark graph to be $V = V_1 \cup V_2 \cup V_3$ and for each pair of vertices x and y with $x \in V_i$ and $y \in V_j (i \neq j)$ join x, y by an edge with probability p, $0 < p < 1$, independently for each such pair of vertices x and y. We denote the class of such graphs by $\mathcal{G}(k_1, k_2, k_3; p)$.

In each of the simulations described below we use $T(n)$ to denote the estimate of the average time to colour an n vertex graph from the proposed class of graphs under consideration.

3. The simulation results

Our pilot runs were on the class of graphs $\mathcal{G}(k, k, k; \frac{1}{2})$ for $k = 4$ to 20. We took the view that any algorithm worth pursuing ought to be tested first on this easily constructed model. Preliminary results were reported in [11] and can be summed up in the following observations:

Observation I. Random algorithms based on antivoter models with transition

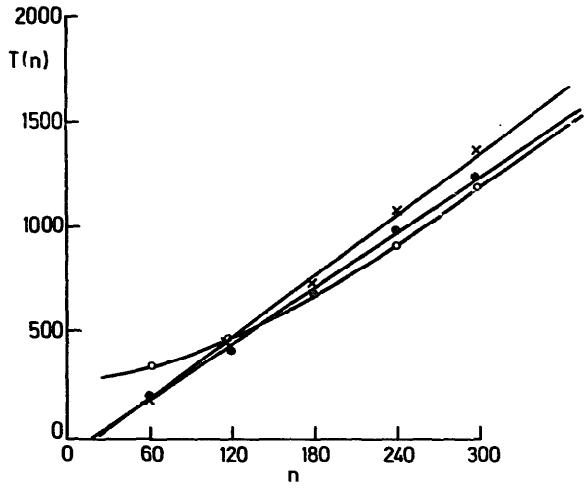

Fig. 1. Data from 100 trials for each value of n.
● $p = 0.5$
× $p = 0.9$
○ $p = 0.3$

functions p of type described in Example 1 above or of the other repulsive type functions

(1) $p_i(s_1, s_2, s_3; i) \sim 1/s_i$

(2) $p_i(s_1, s_2, s_3; i) \sim 1/s_i^2$

do *not* efficiently 3-colour random 3-colourable graphs.

Observation II. A random algorithm based on an antivoter model with transition function p given by

$$p_i(s_1, s_2, s_3; i) \sim 4^{-s_i}$$

appears to achieve a 3-colouring of a random member of the class $\mathcal{G}(n/3, n/3, n/3; \frac{1}{2})$ in time which is linear in n.

Because of the success with this transition function we ran 100 simulations for each n in the range $n = 60$–300, step size 60. The results as shown in Fig. 1 seem to justify Observation II. One slight curiosity in Fig. 1 is the way in which the curve of $p = 0.3$ crosses the two other curves at a relatively low value of n.

We should say that we have no theoretical justification for the choice of $\theta = 4$ in our transition function except that it seems to work at least as well as any other θ used. The same applies to our use of an exponential form for the transition function, though intuitively we believe that the random colouring process we are constructing is somewhat akin to approaching the position of minimum energy in more classical problems of statistical mechanics, as suggested by Kirkpatrick's method of simulated annealing [8] for the travelling salesman problem.

In order to see whether the linear trend exhibited in Fig. 1 holds for larger values of n we proceeded to run the experiment for larger values of n, and for the

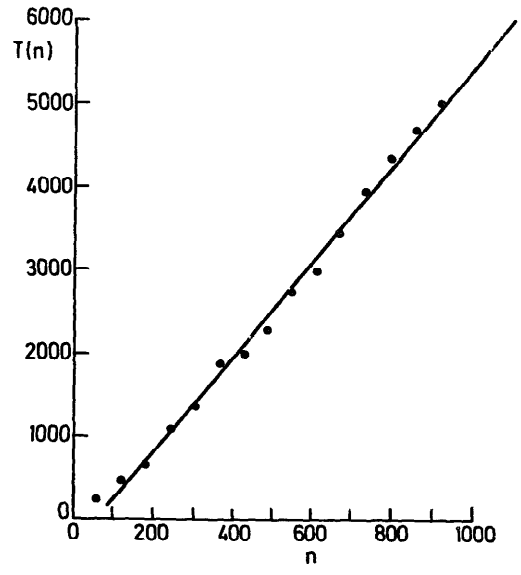

Fig. 2. $p = 0.5$

cases $p = 0.3$, 0.5 and 0.9 found this to be the case. In Fig. 2 we show the results of our simulation for the case $p = 0.5$. The cases $p = 0.3$, $p = 0.9$ are similar.

However for the case $p = 0.1$ as shown in Fig. 3 we did notice a curious phenomenon, namely the existance of a 'hump' at about $n = 80$. This *critical region* was examined in greater detail and the results are shown in Fig. 4.

For future reference note that the critical region ($n \cong 80$) means that the average vertex degree of a vertex in the critical case is $2np/3 \cong 16/3$.

Further simulation for the cases $p = 0.05$ and $p = 0.02$ (one of which is shown in Fig. 5) suggest that the existence of a critical region is not an isolated phenomenon but appears to occur at about a value of n corresponding to the case where the average vertex degree is about 5 or 6. As a further test we ran

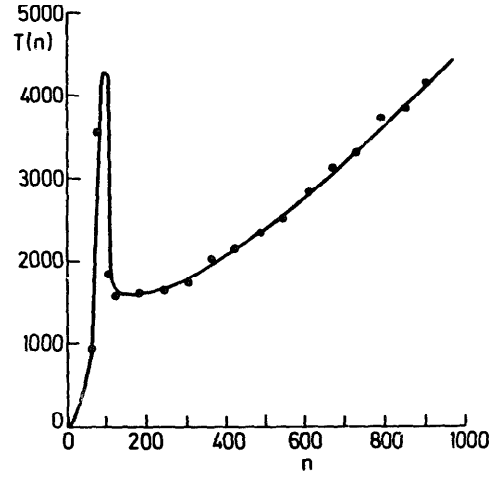

Fig. 3. $p = 0.1$

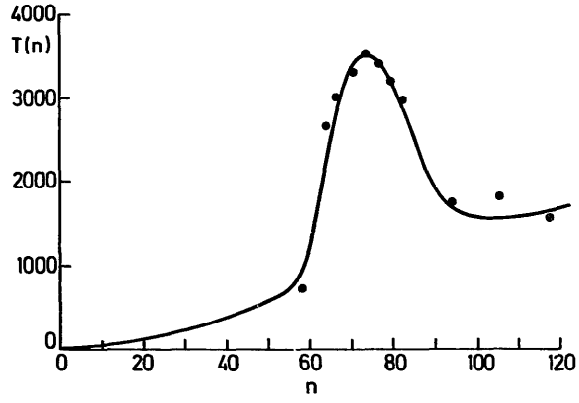

Fig. 4. The critical region in detail for $p = 0.1$

simulations for the case $p = 0.005$ (shown in Fig. 6) which tend to confirm our ideas that for this particular method of colouring the most difficult case among the class of "roughly regular" graphs is the case when the graphs have low vertex degree, say 5 or 6.

We have no theoretical explanation of this curious behaviour. It is not unlike the phenomenon of phase-transition which occurs in the Ising model, Potts model and other models of statistical mechanics and to which this model bears some resemblance (though of course we are using a finite version here).

We close this section by reporting briefly the results of some further experiments.

(I) Changing the value of θ did not seem to affect the position of the critical region in the ranges of n that we were able to work with.

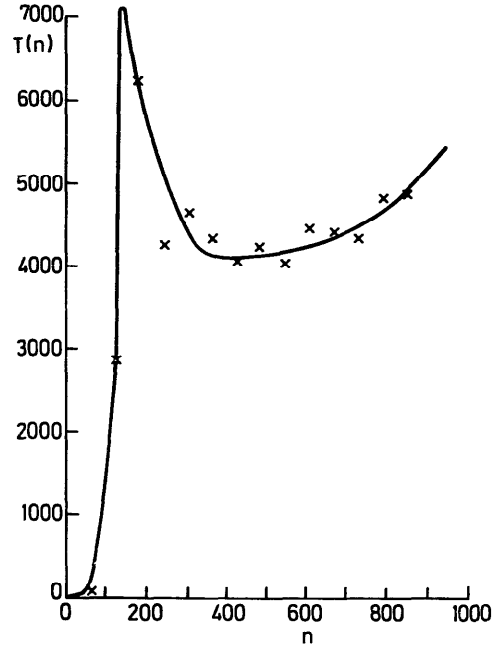

Fig. 5. $p = 0.05$

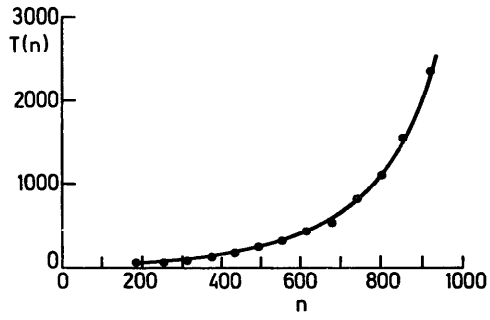

Fig. 6. $p = 0.005$

(II) There seems very little difference in the behaviour of the algorithm on the slightly "more random case" where instead of the benchmark graphs having equal colour sets, the graphs used as test graphs were constructed by the following method used by Kučera [9];

(i) Take a set V of n vertices and arbitrarily colour the members of V with 3 colours so that each vertex has, independently, probability $\frac{1}{3}$ of being assigned any particular colour

(ii) Do not connect a pair of vertices which are assigned the same colour

(iii) For each pair of vertices in different colour sets let them be joined with probability p, independently of the presence of other edges

(iv) Forget the colouring of the vertices.

This procedure will give a graph which by construction is 3-colourable.

4. Conclusion

Various problems are suggested by the above results.

First we should remark that the randomised algorithm proposed does seem to work well in a wide variety of cases. As far as we know the two situations in which it does not appear to 3-colour a 3-colourable graph G efficiently are:

(a) when the only 3-colouring of G decomposes the vertex set V into disjoint sets A, B, C in which the size of A is much larger than that of B or C;

(b) when G is approximately regular with a low vertex degree, say of the order of 5 or 6 for a 1000 vertex graph.

It led us to suggest that possibly Garey Johnson and Stockmeyer [7] did the community a disservice in proving their theorem which states that 3-colouring is NP-complete, (*even*) for graphs of maximum vertex degree 4. It could be that these graphs are among the hardest to colour and it led us to conjecture that if for any fixed $\alpha > 0$ $\mathcal{G}_\alpha(n)$ denotes the collection of graphs with n vertices and minimum vertex degree at least αn then for this class of graphs the colouring problem can be done in polynomial time. This was proved for the case of 3-colouring and $\alpha > \frac{1}{2}$ by Farr [5] and then extended by Edwards [4] who proved

that for $k \geq 3$ the conjecture is true if $\alpha > (k-3)/(k-2)$ and false for $0 < \alpha \leq (k-3)/(k-2)$.

In addition to the "random benchmark" test graphs used in the above simulation we have tried the algorithm out on some "apparently difficult" graphs sent to us as a challenge by R. Irving and K.W. Regan. These examples consisting of two Kneser type graphs of 70 and 130 vertices respectively together with a graph of 341 vertices which was the line graph of a planar graph with no small reducible configuration were correctly coloured in a matter of seconds by the programme (written in the language *C*) on a Perkin Elmer mini-computer.

We have made a preliminary approach to extending the above methods to the k-colouring problem for general $k > 3$. In principle this should not present any greater problem. In practice this does not seem to be the case, preliminary investigations for the case of $k = 10$ suggest that the method is not as good; one reason for this may be that the amount of experimentation needed to find a good transition function in this case takes much more space and time.

Another feature of the method is that preliminary work to start off with a "good colouring" in the sense that the initial bad set B_0 was small did not seem to speed up the algorithm. This can be explained by regarding such a good colouring as approaching a local optimum which is a long way in the metric of exhanges from the true global optimum.

Finally we remark that since the above experiments were first carried out in 1985 Zerovnik [12] has checked the algorithm by independently verifying our results using a different language and machine at the University of Ljubljana.

Acknowledgement

We would like to thank R.W. Irving and K.W. Regan for communicating their "difficult" test graphs, D.E. Blackwell for allowing the use of the computing facilities in the Department of Astrophysics and J. Zerovnik for communicating his results to us.

References

[1] J.A. Bondy and U.S.R. Murty, Graph Theory with Applications (Macmillan, 1976).
[2] P.J. Donnelly and D.J.A. Welsh, Finite particle systems and infection models, Math. Proc. Camb. Phil. Soc. 94 (1983) 167–182.
[3] P.J. Donnelly and D.J.A. Welsh, The antivoter problem: random 2-colourings of graphs, Graph Theory and Combinatorics (ed. B. Bollobas) (Academic Press, 1984) 133–144.
[4] K.J. Edwards, The complexity of colouring problems on dense graphs, Theoretical Computer Science, 43 (1986) 337–343.
[5] G. Farr (private communication, 1985).
[6] M.R. Garey and D.S. Johnson, Computers and Intractability (W.H. Freeman and Co., San Francisco, 1979).

[7] M.R. Garey, D.S. Johnson and L. Stockmeyer, Some simplified NP-complete graph problems, Theor. Comput. Sci. 1 (1976) 237–267.
[8] S. Kirkpatrick, C.D. Gelatt Jr. and M.P. Vecchi, Optimisation by simulated annealing, Science, 220, No. 4598 (1983) 671–680.
[9] L. Kucêra, Expected behaviour of graph colouring algorithms, Lecture Notes in Computer Science, 56 (1977) 477–483.
[10] D.J.A. Welsh, Randomised algorithms, Discrete Applied Maths. 5 (1983) 133–145.
[11] D.J.A. Welsh, Correlated percolation and repulsive particle systems, Proc. Conference Heidelberg (Sept. 1984), Stochastic Spatial Processes (ed. P. Tautu) Springer Lecture Notes 1212, 300–311.
[12] J. Zerovnik (private communication, 1986).