The ground state of the +or-J spin glass from a heuristic matching algorithm

1989 J. Phys. A: Math. Gen. 22 4045

(http://iopscience.iop.org/0305-4470/22/18/036)

J. Phys. A: Math. Gen. 22 (1989) 4045-4059. Printed in the UK

The ground state of the $\pm J$ spin glass from a heuristic matching algorithm

Harald Freund and Peter Grassberger

Physics Department, University of Wuppertal, D-5600 Wuppertal 1, Gauss-Strasse 20, Federal Republic of Germany

Received 22 March 1989

Abstract We present a heuristic matching algorithm for the generation of ground states of the short-range $\pm J$ spin glass in two dimensions. It is much faster than previous heuristic algorithms. It achieves near optimal solutions in time $O(N)$ in contrast to the best known exact algorithm which needs a time of $O(N^{5/2})$. From simulations with lattice sizes of up to 210×210 we confirm a phase transition at $p=0.105$ but we cannot confirm a proposed second transition near $p=0.15$.

1. Introduction

During recent decades, combinatorial optimisation problems have been studied in such diverse fields as applied mathematics, computer chip layout, job scheduling and statistical mechanics [1]. Usually these problems are divided into polynomial and non-polynomial (NP hard) problems [2], depending on whether exact optimisation needs polynomially or exponentially increasing time. But if one is interested in very large systems, this division is no longer useful. Even if an exact algorithm, needing computing time of an order $O(N^{\alpha})$ ($N=$ system size, $\alpha \geqslant 2$) is known, huge amounts of computing time are used for large systems. In this case it would be more convenient to find a heuristic algorithm that reaches near-optimal solutions in a time of $O(N)$, even if they are not strictly optimal.

Many of the above-mentioned problems can be mapped onto spin glasses [1]. Therefore studying, for example, the $\pm J$ spin glass and finding its ground-state energy could be of great help in other fields as well. The $\pm J$ spin glass is described by the Hamiltonian:

$$H=-\sum_{\langle i,i\rangle} J_{ij}s_i s_j \tag{1}$$

where the J_{ij} take on the values $-J$ with probability p and J with probability $1-p$ if i and j are next neighbours, and zero elsewhere.

Among the different methods used to find a ground state, the simplest uses local improvements. The disadvantage of this method is that one is easily trapped in a local, but not global, minimum. To overcome this, one can sample many different starting configurations but this leads to excessive computing time.

The more sophisticated simulated annealing (SA) method is inspired by the slow cooling of melts that is used to get good single crystals. The ground-state energies attained by such Monte Carlo (MC) algorithms are much better than with local

0305-4470/89/184045+15$02.50 © 1989 IOP Publishing Ltd

optimisation methods because at finite T there is always a non-zero probability to jump out of a local minimum. Nevertheless, the true ground state(s) can only be reached when T is reduced in infinitesimal small steps [3] leading again to an infinite amount of CPU time. Therefore one has to define a cooling schedule that reaches $T = 0$ in a finite time. It is clear that the cooling schedule strongly influences the attainable ground-state energy [3]. A simple but quite efficient procedure [4-6] is the following†. One starts with a 'high' temperature T, simulates for a fixed number of MC steps t and then lowers T by a fixed amount ΔT. These steps are repeated until $T = 0$ is reached. The extrapolation to the exact ground-state energy is obtained by comparing runs with different t. In [5] it is argued that the attainable energy per site can be described by

$$E(t) = E_0 + \frac{\text{constant}}{(\ln t)^{\zeta}} \tag{2}$$

with $1 \leqslant \zeta \leqslant 2$. Using the above simple cooling schedule, we verified this logarithmic convergence in [6] and found $\zeta \simeq 1$. For the slowest feasible cooling rate $t = 10\,000$, the results for the $\pm J$ spin glass in two dimensions with a concentration $p = 0.5$ of antiferromagnetic bonds was $E(t) = -1.3893 \pm 0.0004$ [6]. The extrapolated ground-state energy agreed with that found in [8] by means of a transfer matrix formalism. It is $E_0 = -1.4024 \pm 0.0012$, showing that one has to extrapolate over a range $\Delta E_0 \simeq 0.013$.

An advantage of simulated annealing is that it can be coded in a massively parallel way even on standard computers. Working on a vector machine even raises the degree of parallelity. We achieved 29.1 megaflips for a lattice of 512^2 spins on a Cray 2.

In the present paper, we studied (1) for different values of p. For $p = 0$ the simple Ising ferromagnet is obtained which can be treated analytically as was shown by Onsager [9]. For $p = 0.5$ we have a spin glass with a large number of degenerate ground states. From this it follows that a phase transition should occur for a certain value p_c with $0.0 < p_c < 0.5$. In [10] it was conjectured that there are indeed two phase transitions at $p_1 \simeq 0.1$ and $p_2 \simeq 0.15$. Simulated annealing is too slow to test this. Exact minimisation can only be carried out with small lattice sizes [10, 11]. Thus we tried a different heuristic algorithm based on the relation [10-12] with minimal matching problems.

For this, we have first to review a number of well known concepts. Figure 1 shows several elementary plaquettes. Those with an odd number of antiferromagnetic bonds

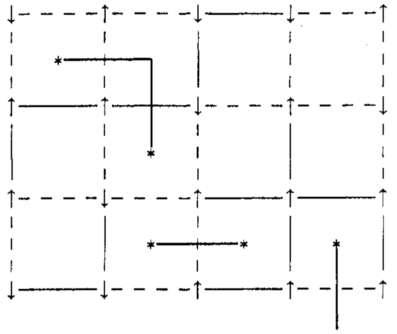

Figure 1. Part of a greater lattice showing strings joining frustrated plaquettes and thus crossing unsaturated bonds. Frustrated plaquettes are denoted by a star. They have an odd number of antiferromagnetic bonds (broken lines).

† For more sophisticated cooling schedules, see [7].

on their perimeter are called *frustrated* [13]. Frustration means there is no spin configuration that minimises all four bond energies $b_{ij} = -J_{ij}s_is_j$. We call bonds 'saturated' if $b_{ij} < 0$ and unsaturated otherwise. So there is at least one unsaturated bond per frustrated plaquette. One can easily verify that in (un)frustrated plaquettes there are an (even) odd number of unsaturated bonds. Drawing lines orthogonal to unsaturated bonds produces strings joining frustrated plaquettes (see figure 1). It is easily seen that the number of frustrated plaquettes is even, if we work on a lattice with periodic boundary conditions.

Obviously, the total energy is minimal if the number of unsaturated bonds is minimal or, equivalently, if the total length of all strings is minimal. Thus, we are able to use matching theory (a part of graph theory) in finding the ground state of the $\pm J$ spin glass.

A graph $G=(V, E)$ consists of a set of vertices V (in our case the frustrated plaquettes) and a set of edges E connecting two vertices (in our case the possible strings between two plaquettes). A perfect matching is a subset M of E such that each vertex is connected to exactly one other vertex†. It is possible to assign a weight to the matching by giving a weight to each edge. If $e \in E$ has the weight $\omega(e)$ then $\omega(\mathrm{M}) = \Sigma_{e\in M}\omega(e)$ is the weight of the matching M. If we assign the length of the strings as their weight (length in x direction plus length in y direction, i.e. 'Manhattan metric') then the ground state of the $\pm J$ spin glass corresponds to a perfect matching of minimal weight. Edmonds [14] found a polynomial algorithm that exactly solves this problem in a time $t \approx N^3$, with $N = L \times L$ the number of vertices (L is the linear extension of the lattice). Recently Vaidya found an algorithm for the two-dimensional matching problem that only needs a time $t \approx N^{5/2}$ [15]. In [16], we gave an algorithm that finds near-optimal solutions in a time $t \approx N$ and that proved successful in the case where the weights assigned to the edges are the Euclidean distances. It is the purpose of the present paper to adapt this method to the present case and to present the results obtained with it.

In the following § 2 we describe our algorithm. Section 3 contains our results. In the last section we discuss our results and compare them to other findings.

2. Algorithm

As seen from § 1, the bond configuration is only needed to compute the frustrated plaquettes. The spins do not have to be considered explicitly at all, if we concentrate on unsaturated bonds [17]. Therefore we consider in this section a square lattice of plaquettes, an even number of which are frustrated, and we want to find a perfect matching of minimal weight.

We begin by constructing some perfect matching where all vertices (i.e. frustrated plaquettes) are linked pairwise. Then we repeat essentially the following elementary step. We select an 'alternating' loop of $2n$ vertices ($n = 2, 3, \ldots$) consisting of n linked edges and n non-linked edges which alternate (see figure 2(*a*)). Then we compute the total length of the linked edges L_{lin} and the total length of the unlinked edges L_{unl}. If $L_{\mathrm{unl}} < L_{\mathrm{lin}}$ we perform a switch (called 'flip' in the following) by replacing linked and unlinked edges (see figure 2(*b*)). If $L_{\mathrm{unl}} > L_{\mathrm{lin}}$ we do not flip and if $L_{\mathrm{unl}} = L_{\mathrm{lin}}$ we flip with probability 50%.

† If one vertex is connected to three other vertices, the matching can still be considered as perfect, with one edge truely connected to the vertex, and the others passing only accidentally through the vertex.

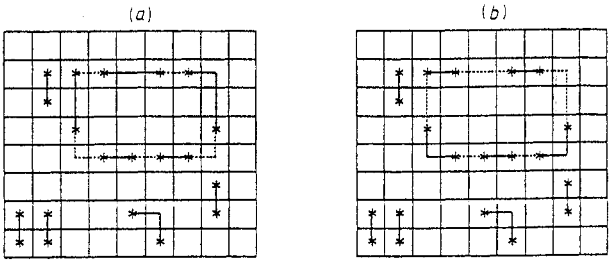

Figure 2. Typical loop with $n = 5$ (full lines are linked edges, broken lines are unlinked edges) before 'flipping' (*a*) and after (*b*).

An easy way to select random loops is the following: we start with a perfect matching, 'break open' one edge and mark one vertex by a flag 'head' (+1) and the other by a flag 'tail' (−1). In the simplest version we randomly chose another vertex in the neighbourhood of the head for matching it to, mark it by a flag −1, break up virtually the link to its previous mate, mark that by a flag +1 and declare it as the new head (see figure 3(*a*)). This is repeated until a loop is formed. Then the above decision whether to flip or not is made. Regardless of this choice, the vertices forming the loop are cleared of their flags. The flags are needed to signal the formation of a loop, and to avoid 'illegal' (i.e. not alternating) loops. The head may only be matched to a vertex without a flag, or to a vertex having the flag −1 as long as it is not its own actual mate (see figure 3(*b*–*d*)).

To speed up the algorithm, lists are stored for the M next neighbours, their distances and their matching probabilities (see point (iv) below) for all frustrated plaquettes. The choices are only from this list (restricting thereby the set of potentially linked edges). The details of the algorithm (or rather class of algorithms) are the following.

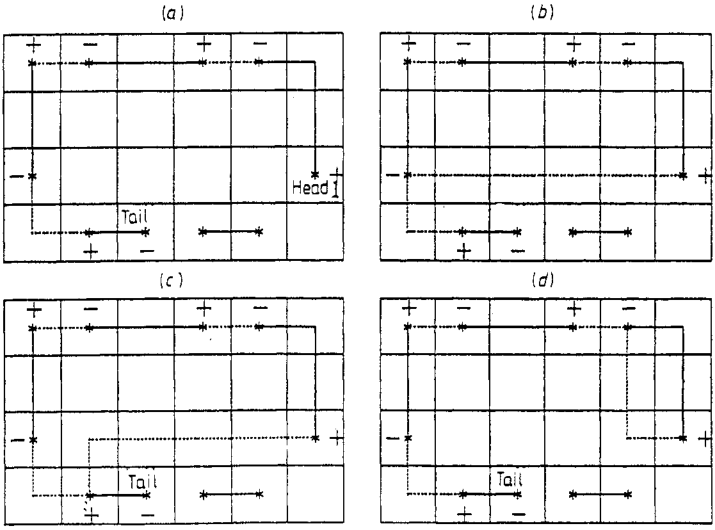

Figure 3. Short walk indicated by full lines between linked edges, and by broken lines between unlinked edges (*a*), legal loop (*b*) and illegal loop (*c*). A loop with only two vertices (*d*) would also be illegal.

(i) There are different possibilities to establish the first matching. Each vertex can be matched to a randomly chosen other vertex. But it is faster if the vertex is matched to one from the above-mentioned list of neighbours. But even when taking only a mate from the list, there are different choices possible. One can go through the list and take the nearest (or the most distant) that is not yet matched, or one chooses randomly from the list. In most simulations we took the nearest possible because it produces a relative short initial matching, but results with the most distant in the list were just as good.

(ii) The first pair that is broken can be taken random or it can be the longest link of the first matching. We tried both but found no effect. In most simulations we made a random choice.

(iii) The number M of next neighbours is mainly limited by available memory. It should be large enough that all edges present in the minimal matching are among the pairs of neighbours. Simulations with $M = 25$ on a lattice of 130 × 130 sites showed that, in the final matching, only links to the 15 nearest neighbours are established (see figure 4). We made runs with $M = 11, 13, 15$ which showed no significant dependence on M. So we feel reasonably safe when using $M = 13$ in most simulations.

(iv) When looking for a possible new mate in the above-mentioned list, we took neighbour k with probability p_k:

$$p_k = A\exp(-\alpha C_p d_k) \tag{3}$$

where d_k is the distance to the kth neighbour, $C_p = 4[p^3(1-p)+p(1-p)^3]$ is the probability that a single plaquette is frustrated, A is a normalisation factor needed to

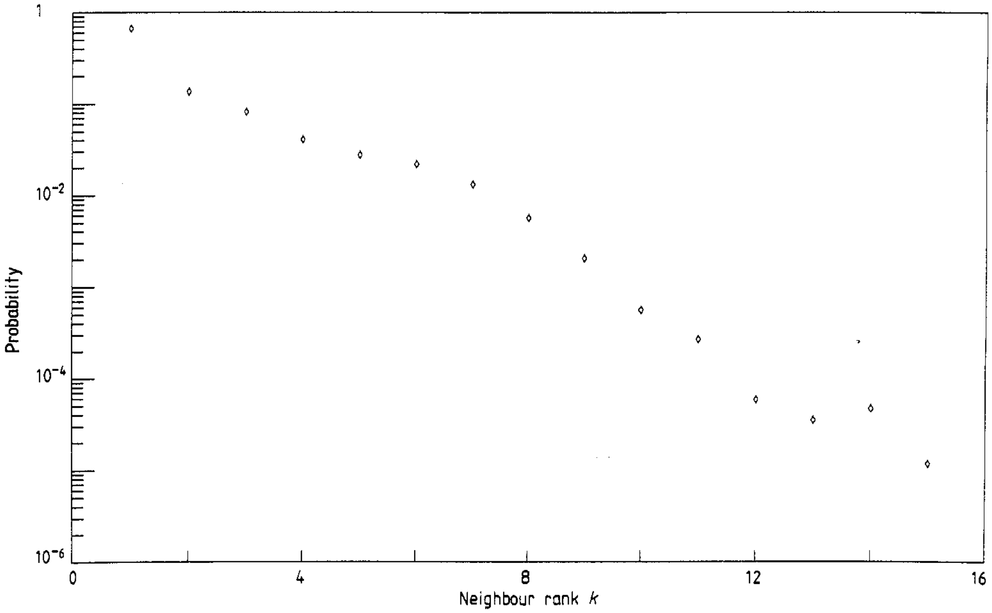

Figure 4. Distribution of neighbour ranks for links in near-optimal configurations (rank is defined such that it increases with distance d, and rank between neighbours with same distance is attributed randomly). These data are from 10 instances with lattice size 130^2, $p = 0.5$ after 5×10^7 walk steps for each instance. Notice that the distribution is cut off at $k = 25$ by the algorithm. Effectively, the distribution is zero for distances $d \geq 4$ and for $k > 15$.

make $\Sigma p_k = 1$, and α is a positive free parameter. This ansatz strongly suppresses long links and softens this suppression for small concentrations of frustrated plaquettes; otherwise, it is completely arbitrary. To speed up the simulation, the list of these probabilities is stored at the beginning. These probabilities are used for the first trial to match a new neighbour. If this trial is unsuccessful (since the chosen neighbour would lead to an illegal loop), we studied two choices for the next trials:

(*a*) we repeatedly choose a neighbour k according to the same probabilities p_k until an acceptable k is found;

(*b*) we choose 'greedily' the nearest acceptable vertex.

Large α corresponds to a greedy choice. This can produce bad results because short loops are repeated very often. In this case iv*a* will fare better than iv*b* because it increases the chance to take a relatively distant neighbour.

Small α allows a generous choice so that flipping usually increases the length which is a bad result, too. But if we combine this with iv*b* we have a kind of generous-greedy choice that will have comparable good results for a wide range of α. We obtained fairly good results with $\alpha \in [7.0 \ldots 9.0]$ (see figure 5). Most simulations were done with $\alpha = 8.2$ and method iv*b*.

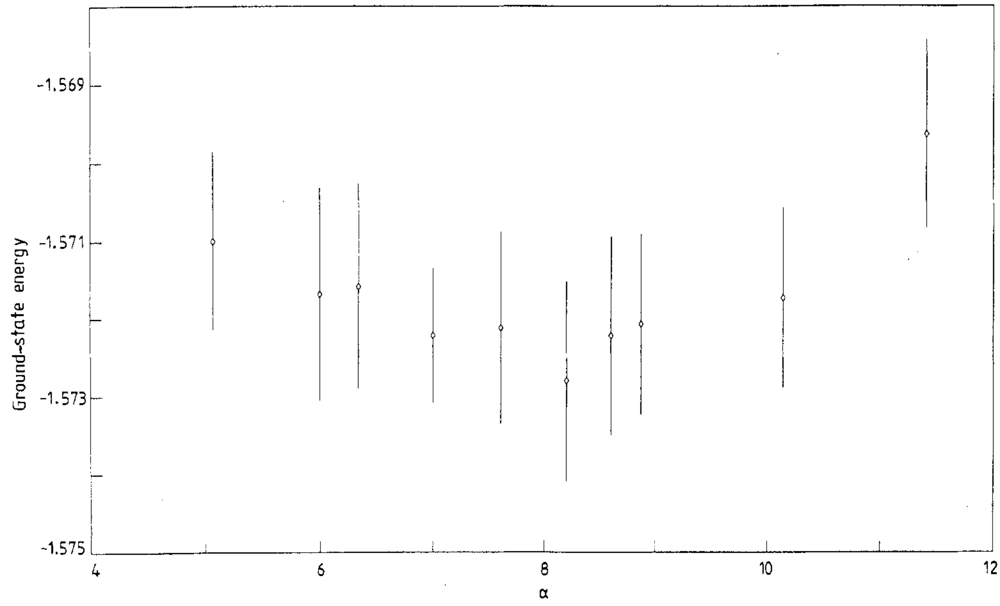

Figure 5. Comparison of the energies reached with different α for lattice size $L = 128$, $p = 0.12$ and total loop length (i.e. total number of steps) $\mathcal{L} = 10^7 C_p$. Method iv*b* was used. Similar runs were made for different p and show the same tendency. The following figures were all with $\alpha = 8.2$.

(v) Usually there are a number of neighbours in the list with the same distance. Let us say, for example, that neighbours $k, k+1, \ldots, k'$ have the same distance d_k. When we now use method iv*b*, neighbour k will be taken in most tries although the other neighbours $k+1, \ldots, k'$ should be taken with the same probability. One way to avoid this problem would be to modify method iv*b* such that we select the next trial randomly among all neighbours with the same distance. Instead of this, we used a faster method where we 'rotate' the neighbours, i.e. we put the chosen neighbour after the k'th, and advance all neighbours between by one position. More precisely, we used the following two alternatives:

(a) we rotate only after a greedy choice;

(b) we rotate always, both after a random and after a greedy choice.

Rotation improved the results. Method vb seems to converge faster although no detailed tests were made.

(vi) When a loop between head and tail is formed (consuming thus the entire walk) there are the same possibilities to go on as in point (ii) above. We always made a random choice.

(vii) There is the possibility to be trapped for a while when repeating the same loop on and on. Going alternatingly in both directions (head and tail) will decrease the chance to be totally trapped. We tried both, two directions showed slighly better results.

(viii) When a loop is formed it is also possible to accept a flip with small (and decreasing) probability if it increases the length, thus turning the algorithm into a version of simulated annealing. But this did not lead to better numerical results, presumably because the way we find loops already enables loops of all lengths.

Figure 6 shows a typical loop statistic. There the number of loops is plotted against the topological length (i.e. the number of linked edges) both for loops not changing the matching (i.e. Manhattan) length and for those lowering it. The simulations stopped when the sum $\mathscr{L}$ of the topological loop lengths exceeded a certain predefined value ($\mathscr{L}$ is the total number of accepted neighbour choices, independent of whether they

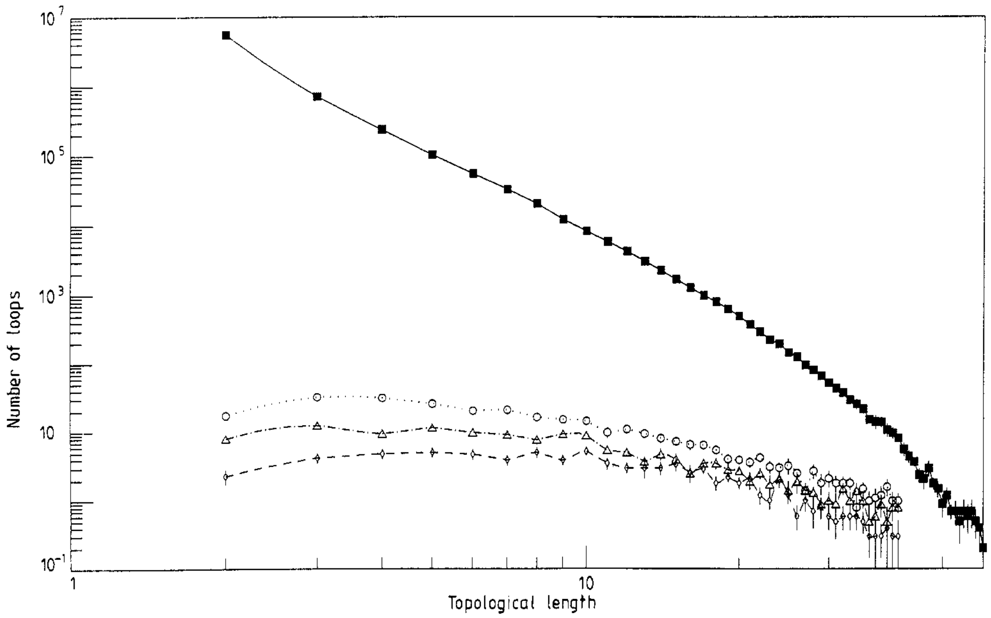

Figure 6. Loop statistic for 10 instances, lattice size 180×180, $p = 0.5$, $\mathscr{L} = 10^8 C_p$ steps. Statistics for $\Delta E = 0$ (full squares) was taken during the last third of the simulation. The circles show the number of length decreasing loops (topological length $n \leqslant 40$) when one thousandth of the simulation was over; the triangles were taken for $0.001\mathscr{L} <$ loop length $< 0.01\mathscr{L}$ and the diamonds for $0.01\mathscr{L} <$ loop length $< \mathscr{L}$.

led finally to loops which were flipped or not). This value secured that each vertex was visited in the average about 1000-6000 times by the random walk. The plot shows that most of the length-decreasing loops were detected before each vertex was visited approximately 50 times. Furthermore we can see that rather long loops are necessary to decrease the matching length. These long loops circumvene clusters of spins with various forms and sizes which are flipped *en bloc* when flipping the loop.

The statistic for length-preserving loops was taken during the last third of the simulation, where we are in states that are only a few excitations above the true ground state. We believe thus that it reflects properties of ground states. These properties will be studied in detail in § 3.

Summarising these results we can see that our algorithm is extremely robust in finding near optimal solutions. What remains to be seen is what we actually achieve in a time of O(N).

3. Results

Our results for the energy reached (per spin) are shown in figure 7. They show two things. On the one hand we can see that (2) can be used to extrapolate to the true ground-state energy, with a best value of $\zeta = 2.0$. This reflects the fact that we employ

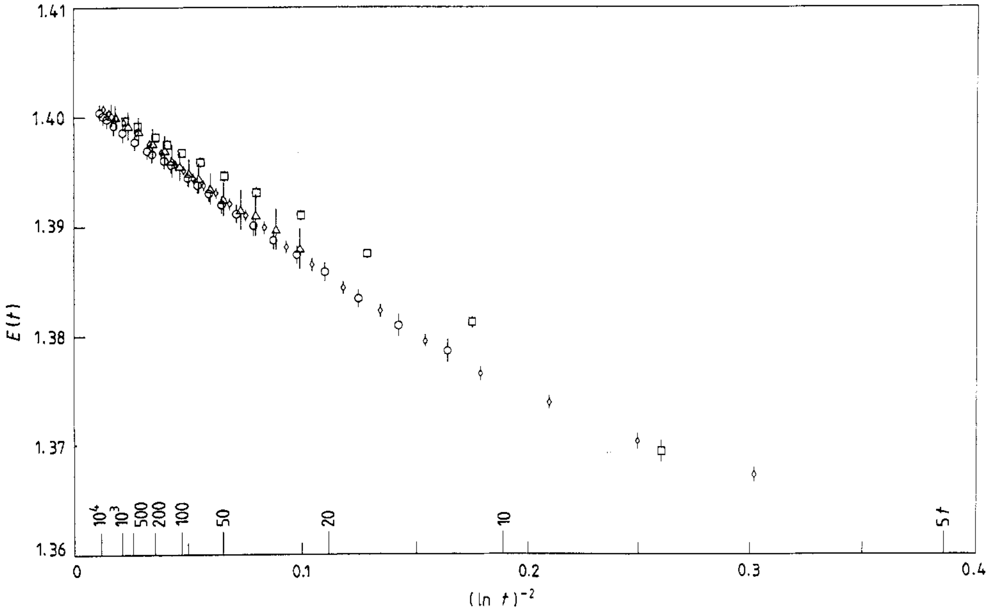

Figure 7. Energy reached for different lattic sizes, versus the iteration time t per frustrated plaquette defined by

$$t = \frac{2\mathscr{L}}{\text{number of frustrated plaquettes}}.$$

Plotted on the abscissa is actually $(\ln t)^{-2}$, in view of (2). The initial matching for lattice sizes $L = 50$ (Δ), 130 (○), 180 (◇) was with the same procedure and method v*a* was used. Whereas method v*b* and a different initial matching procedure were employed for lattice size $L = 210$ (□).

a Monte Carlo algorithm, albeit it is not simulated annealing proper [5]. On the other hand, we can see that the results are nearly independent of details of the algorithm: for lattice size 210^2 the procedure for the first matching was different from the other lattice sizes, method v*b* was used, and the list of neighbours contained 16 instead of 13 entries for each frustrated plaquette. From this plot we extrapolate $E_0 = 1.4023 \pm 0.0015$ for $p = 0.5$ which is in perfect agreement with the best literature estimates using completely different algorithms [6, 8].

As was already mentioned in § 2, we see that the convergence to the true ground state is described by (2) not only for simulated annealing but also for this new algorithm. With the SA algorithm of [6] we need several runs with different timescale t to extrapolate to the true ground-state energy. On a Cray 2 it took approximately 4600 s CPU time to simulate 10 lattices of each 512×512 spins using six different t (500–16 000). Our best result there was $E_0(t = 16\,000) = 1.3899$. In contrast to this, our present algorithm was run on a micro-VAXII. Our largest lattice was 210^2. To reach the same $E(t)$ on 512^2 sites it would take only approximately 16 000 s, although the Cray 2 is at least 300 times as fast as a micro-VAXII. Thus, employing SA on a micro-VAXII would have taken approximately 1.4×10^6 s of CPU time. So we see our algorithm is about two orders of magnitude faster than SA although it can *not* be coded in parallel. The best achieved $E(t)$ with our algorithm was only 0.11% worse than the extrapolated true ground-state energy. Figure 8 shows reached or extrapolated energies E_0 plotted against C_p. The broken line is the function $E_0(C_p) = -1.9147 + 1.0228 C_p$ given in [11]. As expected, we can see no hint to a phase transition in this plot. This does in no way contradict the presence of two phase transitions at $C_p \simeq 0.295$ ($p \simeq 0.1$) and at $C_p \simeq 0.38$ ($p \simeq 0.15$) found by Barahona *et al* [10].

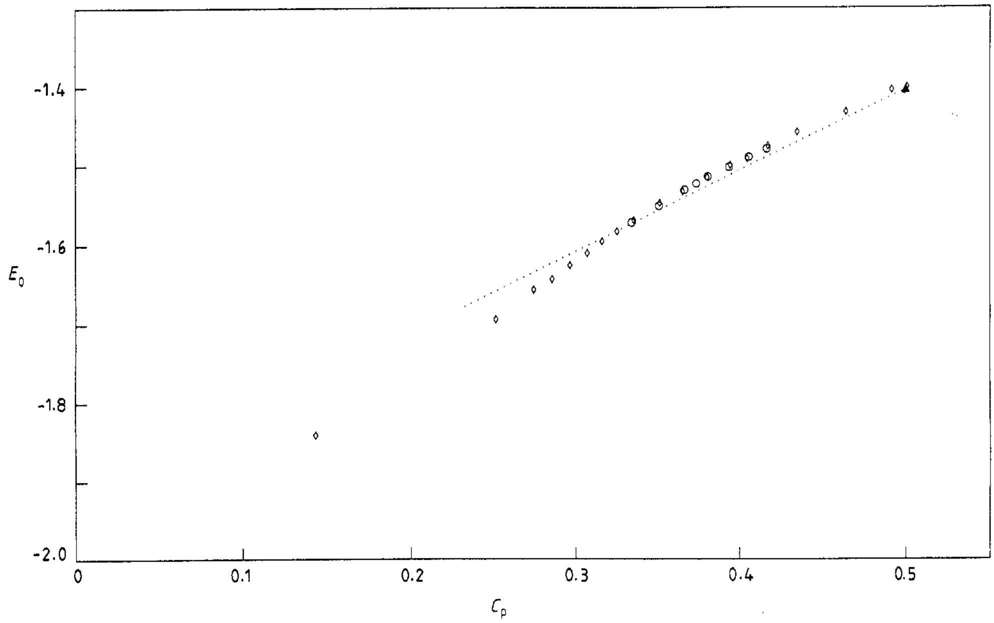

Figure 8. Ground-state energies calculated with different methods plotted against C_p. The dotted line is the function $E_0(C_p) = -1.9147 + 1.0228 C_p$ given in [11]. It gives good estimates for $C_p > 0.315$ ($p > 0.11$). ▲, [6]; ○, SA algorithm of [6] extrapolated to $t = \infty$; ◇, matching algorithm (not extrapolated).

Barahona *et al* studied lattices of up to 20 × 20 spins. They computed true ground-state configurations and analysed their structure. To achieve this they used essentially the algorithm by Edmonds [14]. After finding one ground state they could compute clusters of 'solidary' spins. Solidary spins are spins that have the same orientation in all ground states. They are connected by 'rigid' bonds which are either saturated or unsaturated in all ground states.

By inspecting these clusters they found that below $p_c' \approx 0.15$ the solidary spins form one percolating cluster that implies long-range order. Above p_c' so-called 'fracture lines' appear that break up the percolating cluster into smaller pieces, so that long-range order vanishes. As there are small clusters of solidary spins that can 'flip' easily there is no magnetisation left. They call this phase superparamagnetic, 'super' because not a single spin can flip without changing energy but clusters of solidary spins with different sizes and forms.

In the phase where the percolating cluster exists they find a second phase transition that is characterised by 'magnetic walls'. Usually one employs periodic boundary conditions. But for $p > p_c \approx 0.10$, antiperiodic boundary conditions in one direction lead sometimes to a ground state with a lower energy.

So we see there are three different proposed phases for p in the range 0.0–0.5.

(i) For low concentrations $(0.0 < p < p_c \simeq 0.1)$ we find ferromagnetic ground states. They are characterised by solidary spins forming one percolating cluster and spins pointing mostly in the same direction.

(ii) For moderate concentrations $(0.1 < p < p_c' \simeq 0.15)$ one has a phase which is 'rigid' in the sense that large clusters cannot be flipped, but which is not ferromagnetic either, since the clusters are kept in random magnetisation. This is the random antiphase state.

(iii) For high concentrations $(p > 0.15)$ the percolating cluster is destroyed by fracture lines into small clusters of solidary spins. No long-range correlation can survive. This is the superparamagnetic phase.

Barahona *et al* showed that, when a spin glass is simulated with finite temperature T, there will be a critical temperature T_c only when p is below p_c'. A disadvantage of their method was that they could only investigate small lattices (linear lattice size $L \leqslant 20$). In contrast to this our algorithm can process large lattices $(L \approx 200)$ in a reasonable amount of CPU time. Although we will usually not find a true ground state, we in a way can scan properties of states with a small but finite temperature. So we should find hints that for $p \leqslant p_c'$ there is a percolating cluster of solidary spins. Or to express it differently, loops that can be flipped without changing the total matching length (in the following called floppy loops) can surround only very small clusters of spins that do not belong to the percolating cluster. In figures 6 and 9 the full squares show the statistics of floppy loops when we are only a few excitations above the ground state. For $p = 0.5$ we see that quite large floppy loops are found. In contrast to this we found for small p a distinct cutoff (see figures 9 and 10). For $p = 0.08$ our algorithm detected only loops shorter than $n \approx 10$ for a lattice size $L = 50$, hinting clearly at a phase transition with $p_c > 0.08$. Since each floppy loop encircles a cluster whose magnetisation is not fixed, this is a transition to the superparamagnetic phase. It seems to occur near $p = 0.1$, but a precise location of the transition point from the loop statistics does not seem possible.

So we turned to some other quantity. As we mentioned above, Barahona *et al* used both periodic and antiperiodic boundary conditions. For $0.1 \leqslant p \leqslant 0.15$, antiperiodic boundary conditions did not increase the ground-state energy. In contrast to this, we

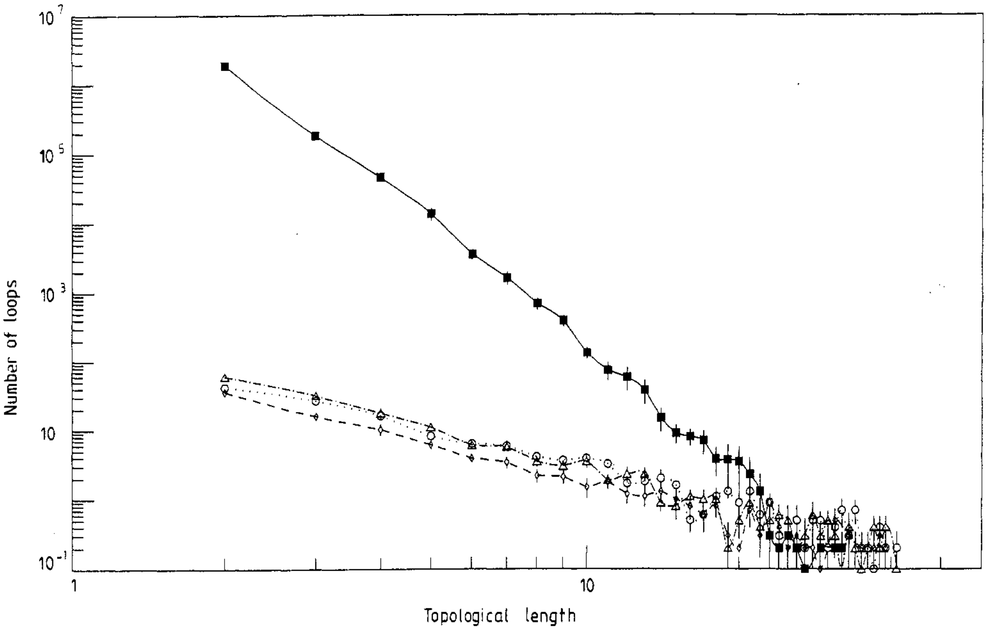

Figure 9. As figure 6, but for $p = 0.08$. No loops with topolgoical length $n \geq 32$ were found.

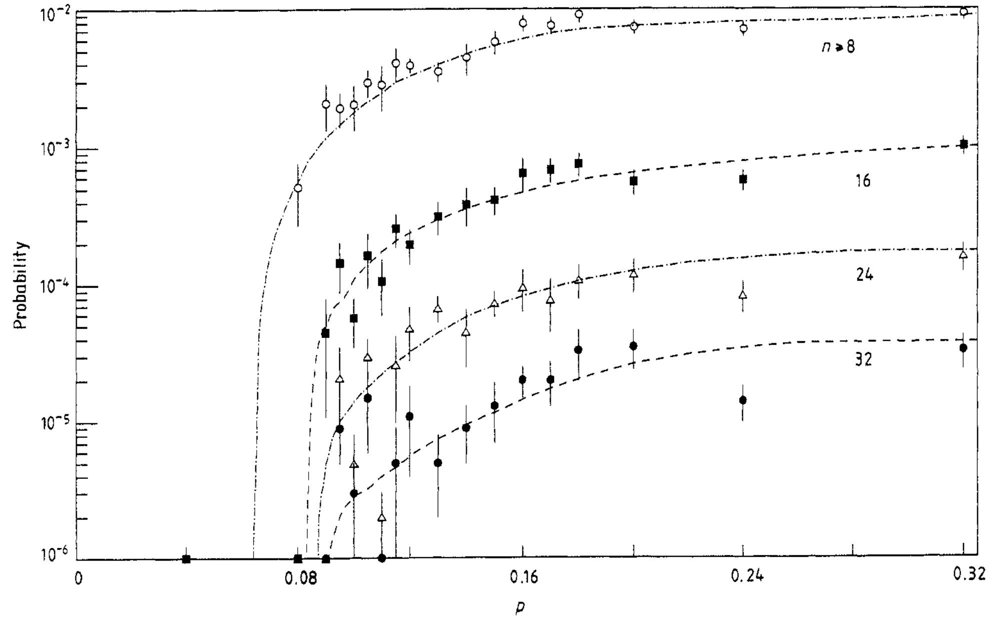

Figure 10. Probability of floppy loops with a topological length n exceeding a certain value. Lines are only given to guide the eyes. Symbols plotted on the horizontal axis indicate zero probability, i.e. absence of any such loop in the sample.

had no way of fixing the boundary conditions as periodic or antiperiodic. They could change during the simulation when a loop with topological index $\neq 0$ was flipped. More precisely, by using periodic or antiperiodic boundary conditions, and with a modified Manhattan metric:

$$\mathrm{d}_{AB} = \min(|x_A - x_B|, L - |y_a - y_B|) + \min(|y_A - y_B|, L - |y_A - y_B|) \tag{4}$$

we actually work on a 2D torus. On such a torus loops can have trivial topology (loop A in figure 11), they can be homotopic to loops B or C which encircle the torus, or they can encircle it several times. Flipping loops of type A corresponds to flipping a cluster of spins. Flipping loop B or C corresponds to flipping the spins on one of its sides but not on the other. This corresponds to exchanging periodic boundary conditions by antiperiodic and vice versa.

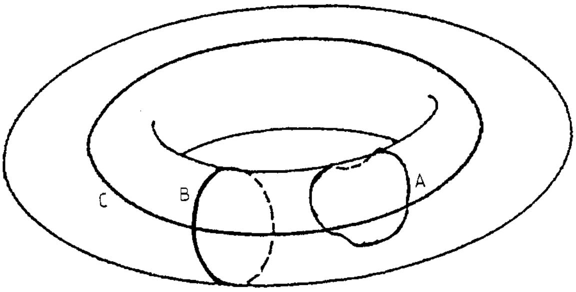

Figure 11. Loop A has trivial topology, whereas B and C encircle the torus.

Our algorithm finds planar loops as well as encircling loops. Thus all kinds of boundary conditions can be in effect during the simulation (periodic in both directions, antiperiodic in both directions or periodic in one and antiperiodic in the other direction). How can we find out what kind of boundary condition is realised at a given instant? To illustrate our method we use a one-dimensional example that can be generalised easily. Think of a one-dimensional $\pm J$ spin glass of length L. There will be m antiferromagnetic couplings and $L - m$ ferromagnetic ones. If we apply (anti) periodic boundary conditions m must be (odd) even if we want to minimise all bond energies. Otherwise at least one bond will not be saturated.

This holds true also in two dimensions. Assume that in a given row there are m_x antiferromagnetic horizontal bonds and b_x unsaturated horizontal bonds. Periodic boundary conditions in the x direction are in effect if and only if $m_x + b_x$ is even. The crucial observation is that, if

$$B_x = \begin{cases} \mathrm{p} & m_x + b_x \text{ even} \\ \mathrm{a} & m_x + b_x \text{ odd} \end{cases} \tag{5}$$

is 'p' for one row, it has to be so for all other rows too. This is easily shown by building up the configuration from a purely ferromagnetic lattice in its ground state by changing bonds and flipping spins. Each such change conserves B_x. The same rule can be formulated for the y direction. So a relatively simple procedure can be written that finds out boundary conditions. A matching is characterised by the values of the two variables B_x, B_y, where B_i can be 'p' for periodic or 'a' for antiperiodic boundary conditions. 'pp' now stands for a matching with periodic boundary conditions in x and y direction and 'pa' for a matching with periodic boundary conditions in x direction

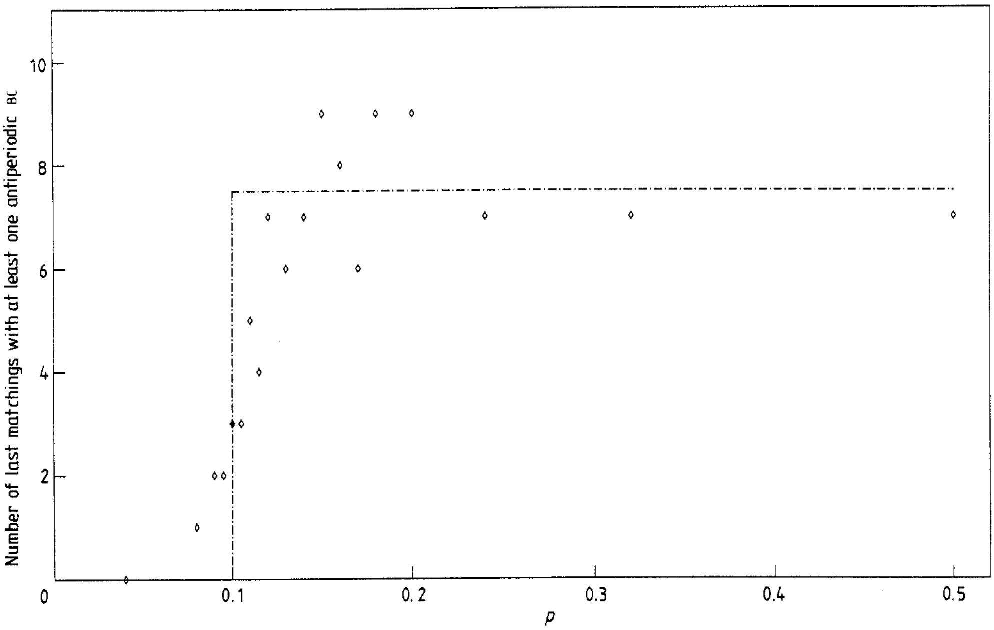

Figure 12. Number of lowest energy configurations with antiperiodic boundary conditions in at least one direction. Lattice size $L=50$, $\mathscr{L}=3\times 10^6 C_p$. There is a distinct change for $p\approx 0.105\pm 0.01$. The chain line would be the asymptotic curve (for $L=\infty$) in the case of a transition at $p_c=0.1$.

and antiperiodic boundary conditions in y direction. For each p we simulated 10 lattices of size $L=50$. In the first matching, all four combinations aa, ap, pa and pp were equally likely, for all values of p. But after the simulation, the last matching showed a clear p dependence. Figure 12 shows that, for low concentrations $(0.0<p\leqslant 0.1)$, we find periodic boundary conditions in both directions in most cases. For high concentration $(p\geqslant 0.12)$ all four boundary conditions are equally likely again. So we locate the phase transition from ferromagnetism to the random antiphase or paramagnetic state at $p_c=0.105\pm 0.01$, in good agreement with the result of Barahona *et al.*

Can the second phase transition at p_c' also be confirmed with our data? This phase transition is characterised by fracture lines that destroy the percolating cluster of solidary spins. Fracture lines correspond to circumvening floppy loops. Unfortunately they are strongly suppressed in our algorithm because, to find such a loop, we have to go in the same direction very often during the random walk. For a lattice with $L=50$ we need a loop that has the topological length $n\approx 25$ to form a circumvening loop. From figure 10, we see that such loops are too rare to give statistically significant results. On the other hand, the absence of any structure in figure 10 near $p=0.15$ suggests that there is no second transition, and thus no random antiphase state.

4. Conclusion

We have presented an algorithm that is much faster than simulated annealing although it cannot be vectorised. It reaches states that are quite near to the true ground states and is thus able to scan properties of low lying states. Essentially it flips clusters of spins with various sizes and forms, which is strongly reminiscent of recent cluster flipping algorithms by Swendsen, Wang and others [18-21]. We could confirm the first phase transition from ferromagnetism towards the spin glass state that was proposed by Barahona *et al* [10], with lattices that in linear dimension are more than twice as large. We were not able to verify a second phase transition found by Barahona *et al* which would imply the èxistence of a random antiphase state. But we must stress that we cannot strictly exclude such a phase either.

Although our algorithm cannot be used in simulating the 2D $\pm J$ spin glass with external magnetic field or the three-dimensional $\pm J$ spin glass, we nevertheless think it could have wide ranging applications. All that is needed is a problem that can be formulated as a minimal or maximal weighted perfect matching problem. This perfect matching does *not* need Manhattan metric or periodic boundary conditions [16] and we are not restricted to two dimensions. It does not have to have any connection to spin glasses, as illustrated by the old interest in minimal matching problems in the mathematical literature [14]. In particular, the relationship between spin glasses and matching problems breaks down in more than two dimensions.

When discussing an Ising-like model, flipping a loop is equivalent to flipping the cluster of spins surrounded by this loop. In our algorithm, the selection of the loops is done via a random walk. It is this latter aspect of our method which could be most useful in generalisations to other spin models in two dimensions. Take for instance a 2D Ising spin glass with Gaussian distribution of the couplings. In this case, the concept of frustration does not apply, and we cannot make walks on the set of frustrated plaquettes, as done in the present paper. But we can select candidates for clusters to be flipped in a Monte Carlo simulation (not necessarily of the ground state) by performing a random walk on the set of all plaquettes, with steps only between next neighbours. Provided that these walks select occasionally large clusters, such an algorithm can be much faster than conventional MC algorithms, and at the same time simpler than other recently proposed cluster flip algorithms [18-21].

Acknowledgment

We very much thank Mørch Pedersen for a critical reading of the manuscript.

References

[1] Kirkpatrick S, Gelatt C D and Vecchi M P 1983 Optimization by simulated annealing *Science* **220** 671

[2] Garey M R and Johnson D S 1979 *Computers and Intractability: A Guide to NP-Completeness* (New York: Freeman)

[3] van Laarhoven P J M and Aarts E H L 1987 *Simulated Annealing: Theory and Application* (Dordrecht: D Reidel)

[4] Grest G S, Soukoulis C M and Levin K 1986 Cooling-rate dependence for the spin-glass ground state energy: implications for optimization by simulated annealing *Phys. Rev. Lett.* **56** 1148

[5] Huse D A and Fisher D S 1986 Residual energies after slow cooling of disordered systems *Phys. Rev. Lett.* **57** 2203

[6] Freund H and Grassberger P 1988 Multispin coding for spin glasses *J. Phys. A: Math. Gen.* **21** L801

[7] Salamon P, Nulton J, Robinson J, Pedersen J M, Ruppeiner G and Liao L 1988 Simulated annealing with constant thermodynamic speed *Comput. Phys. Commun.* **49** 423

[8] Cheung H-F and McMillan W L 1983 Equilibrium properties of the two-dimensional random ($\pm J$) Ising model *J. Phys. C: Solid State Phys.* **16** 7027

[9] Onsager L 1944 Crystal statistics I: a two-dimensional model with an order-disorder transition *Phys. Rev.* **65** 117

[10] Barahona F, Maynard R, Rammal R and Uhry J P 1982 Morpholgy of ground states of two-dimensional frustration models *J. Phys. A: Math. Gen.* **15** 673

[11] Bieche I, Maynard R, Rammal R and Uhry J P 1988 On the ground states of the frustration model of a spin glass by matching method of graph theory *J. Phys. A: Math. Gen.* **13** 2553

[12] Vannimenus J and Toulouse G 1977 Theory of the frustration effect: Ising spins on a square lattice *J. Phys. C: Solid State Phys.* **10** L537

[13] Toulouse G 1977 Theory of the frustration effect in spin glasses: I *Commun. Phys.* **2** 115

[14] Edmonds J 1965 Path, trees, and flowers *Can. J. Math.* **17** 449

[15] Vaidya P M 1988 *Proc. 20th ACM Symp. on the Theory of Computing, Chicago*

[16] Grassberger P and Freund H An efficient heuristic algorithm for minimal matching *Z. Oper. Res.*

[17] Bennett C H, Margolus N and Toffoli T 1988 Bond-energy variables for Ising spin-glass dynamics *Phys. Rev.* B **37** 2254

[18] Swendsen R H and Wang J-S 1986 Replica Monte Carlo simulation of spin-glasses *Phys. Rev. Lett.* **57** 2607

[19] Kandel D, Domany E, Ron D, Brandt A and Loh E Jr 1988 Simulations without cricial slowing down *Phys. Rev. Lett.* **60** 1591

[20] Niedermayer F 1988 General cluster updating method for Monte Carlo simulations *Phys. Rev. Lett.* **61** 2026

[21] Wolff U 1989 Collective Monte Carlo updating for spin systems *Phys. Rev. Lett.* **62** 361