SCCS-528 DRAFT

A generalized cluster algorithm for the 3-d fully frustrated Ising model

P.D. Coddington

Northeast Parallel Architectures Center, 111 College Place, Syracuse University, Syracuse, NY 13244, U.S.A.

L. Han

Physics Department, Syracuse University, Syracuse, NY 13244, U.S.A.

February 1, 1994

Monte Carlo cluster update algorithms are very efficient for many simple spin models, however they are ineffective for frustrated spin systems. Recently a generalized cluster algorithm was introduced that works extremely well when applied to the two dimensional fully frustrated Ising model. We have generalized this algorithm to the three dimensional model and found that it works poorly, producing clusters that are too large or too small.

I. INTRODUCTION

Cluster Monte Carlo algorithms [1, 2] can give a great improvement in computational efficiency over the Metropolis algorithm [3, 4] and other standard local update schemes by dramatically reducing critical slowing down [4, 5] in certain ferromagnetic and anti-ferromagnetic spin models. However these algorithms fail for frustrated spin models, the systems for which critical slowing down can be most extreme [6, 7]. This is because at the critical point, these algorithms produce a cluster that encompasses a large fraction (if not all) of the sites in the lattice, so that updating the spins in this cluster is virtually the same as a trivial global spin update.

Kandel, Ben-Av and Domany (KBD) have developed a generalized cluster algorithm [8, 9] that works very well for the two dimensional fully frustrated Ising model [8, 10]. We have recently introduced a practical methodology for constructing a generalized cluster algorithm for any spin model, and applied it to some simple two dimensional frustrated Ising models [11]. Here we apply this method to a more complex and interesting model, the three dimensional fully frustrated Ising model.

In section II we introduce the system to be studied, the fully frustrated Ising model on a simple cubic lattice. In section III we present an overview of the KBD generalized cluster algorithm, and briefly review the method we have developed for formulating a generalized cluster algorithm for a given spin model. We apply this method to the 3-d fully frustrated Ising model in section IV, and attempt to improve on the algorithm in section V.

II. THE 3-D FULLY FRUSTRATED ISING MODEL

The Ising model [4] has an interaction energy given by

$$E = -\sum_{<i,j>} J_{ij}\sigma_i\sigma_j, \tag{1}$$

where the spins σ_i can take the values $+1$ or -1. Here we consider the case where the interaction strength J_{ij} takes on values $+J$ or $-J$, for some constant parameter $J > 0$. A link is defined to be the connection between two neighboring sites i and j on the lattice. A link is said to be satisfied if the energy $-J_{ij}\sigma_i\sigma_j$ of the link is minimized, otherwise it is unsatisfied. In a ferromagnetic spin model, all the links are satisfied in the zero temperature ground state. However for some types of spin models it is not possible for all the links to be satisfied simultaneously. These are known as *frustrated* models [6], with the most well known example being spin glasses [7].

The behavior of spin glasses is complicated by the existence of both frustration and disorder. Villain introduced a simpler model, the fully frustrated Ising model (FFIM), that has frustration but no randomness [12]. In this model each fundamental plaquette is frustrated. For a square lattice, this can be achieved by placing ferromagnetic couplings on all links except for a line of anti-ferromagnetic couplings on every second column of links. This means that every plaquette is frustrated, since it has 3 ferromagnetic links and one anti-ferromagnetic link. This model is simple enough that its critical behavior can be described analytically, and has been shown to have a critical point at zero temperature.

This model can be easily generalized to a simple cubic lattice in three dimensions [14–16] so that every plaquette, or face of a unit cube, is frustrated, having one anti-ferromagnetic and 3 ferromagnetic links, as shown in Fig. 1. The FFIM is a much more interesting model in three dimensions, since is more complex and not analytically soluble. This model has been studied using Monte Carlo simulation [17–19], however standard Monte Carlo methods are very inefficient in this case. In fact the standard Metropolis algorithm does not work at all at zero temperature, since flipping any single spin of a ground state configuration will increase the energy. The model is believed to have a phase transition at a small but non-zero critical temperature T_c, however because of serious critical slowing down due to the ineffectiveness of the Metropolis algorithm, numerical simulations have been restricted to small lattice sizes. Consequently the results are far from conclusive, and even the order of the transition is in doubt [18, 19]. This model also exhibits interesting behavior at low temperature ($T < T_c$), however this is even more difficult to investigate using standard Monte Carlo algorithms. Clearly an effective cluster algorithm for this model would be extremely helpful.

In the Swendsen-Wang (SW) cluster algorithm for the Ising model [1], bonds are introduced between spins on neighboring sites with probability $1-e^{-2K}$ if the link is satisfied, and 0 if it is unsatisfied. Here $K = J/kT$ is a dimensionless coupling constant, where T is the temperature and k is Boltzmann's constant. This procedure creates clusters of bonded sites. The update part of the algorithm consists of changing all the spins in a cluster with probability $\frac{1}{2}$. Since the clusters can be quite large, this clearly produces large, non-local changes in the spin configuration, and thus decorrelates the configurations much faster than local update algorithms.

This algorithm works very well for the ferromagnetic ($J_{ij} = +J$) and anti-ferromagnetic ($J_{ij} = -J$) Ising model, but fails for frustrated systems. The simple reason for this failure is that the region of interest for such models is usually at low temperature (i.e. large K), where the probability of putting a bond on a satisfied link is close to 1. For example, at $T = 0$ the 2-d FFIM has 3 satisfied links per plaquette, so that the SW algorithm will bond every site to 3 of its 4 neighbors on average, which is more than enough to connect all the sites in the lattice into a single cluster [11]. In three dimensions the problem is even worse since the connectivity of the lattice is higher, and for the 3-d FFIM the the SW algorithm bonds each site to 4 of its 6 neighbors on average, again freezing the whole lattice into a single cluster. A more fundamental reason for this problem is that the placement of bonds in the SW algorithm depends only on the information from the links, from which it is not possible to see the frustration in the system. To do that one needs information from the spins over at least a plaquette, which is the basic idea behind the KBD generalized cluster algorithm for the 2-d FFIM.

III. CONSTRUCTING A GENERALIZED CLUSTER ALGORITHM

To construct a generalized cluster algorithm, we begin by expressing the energy of the system as a sum $E = \sum_l V_l$. Here l labels connected groups of sites that can give a tiling of the lattice, that is, they can be replicated over the lattice in such a way that each link is part of one (and only one) such group. We will refer to such a group of sites as a *tile*. For the algorithm presented here for the 3-d FFIM on a simple cubic lattice, we take the tile to be the unit cube. For each tile l we stochastically assign one of n possible operations with a probability $P_i^l(u)$ that depends on the spin configuration u and the operation i. These operations involve placing bonds between certain sites in the tile l. We refer to the placement (or non-placement) of bonds as *freezing* (or *deleting*) the connections between two neighboring sites. The frozen bonds produce clusters of connected sites. The probabilities must of course satisfy

$$\sum_i P_i^l(u) = 1, \qquad 0 \le P_i^l(u) \le 1, \tag{2}$$

for all l and u. KBD show that detailed balance is satisfied if the probabilities also satisfy [9]

$$E(u) - \frac{1}{\beta} \log P_i(u) + C_i = \begin{cases} 0 & \text{for an allowed operation } i \\ \infty & \text{otherwise.} \end{cases} \tag{3}$$

Here $E(u)$ is the energy for the configuration u, C_i is a constant (independent of the configuration), $\beta = 1/kT$ is the inverse temperature, and we have suppressed the subscript l labeling the particular tile. In our implementation we will define an operation as *allowed* if it does not freeze any unsatisfied links of the configuration u.

Since constructing the probabilities depends only on whether the links are satisfied or unsatisfied, we will henceforth take "configuration" to mean "configuration of links". If we define $a_i(u)$ to be 1 if operation i is allowed for configuration u, and 0 otherwise, then we can combine equation 3 and the constraint on the sum of the probabilities in equation 2 into a simple matrix equation [11]

$$\mathbf{A}\,\mathbf{c} = \mathbf{b}, \tag{4}$$

where $c_i = e^{C_i}$ are the elements of the vector $\mathbf{c}$, the Boltzmann factors $b_j = e^{-\beta E(u)}$ are the elements of the vector $\mathbf{b}$, and the $a_i(u)$ make up the elements A_{ij} of the *allowance matrix* $\mathbf{A}$, with j labeling the different possible configurations u.

Constructing a generalized cluster algorithm involves identifying some candidate operations, constructing the allowance matrix $\mathbf{A}$, and then solving the matrix equation 4 for $\mathbf{c}$, from which the probabilities for each operation can be calculated via equation 3. For a given set of operations and configurations, there is no guarantee that a solution to the matrix equation will exist. However it will generally be possible to choose the operations so that the allowance matrix is square (the same number of operations as configurations) and upper triangular. This will guarantee a solution, although it may not satisfy the constraint that the probabilities are all between 0 and 1. There will generally be more than one operation available for a given configuration that will produce an upper triangular allowance matrix.

For frustrated models such as the 3-d FFIM, we want to choose the operation that freezes the least number of bonds, and bonds together the fewest sites into the same cluster, to try to reduce the size of the largest cluster. In some cases it may be advantageous to choose more than one such operation for a given configuration. This gives a system of equations that is under-determined, producing free parameters in the solution. This may allow some optimization in the algorithm by tuning the size of the largest cluster, as can be done for the 2-d FFIM [8, 10, 11].

In problems where there is a lot of symmetry, such as when the tile is a plaquette or a unit cube (as in this case), there will be a number of configurations of the same energy that are just different orientations of the same basic topology of satisfied and unsatisfied links. Similarly, there will be many operations that are just different orientations of the same topology of frozen and deleted bonds. We can greatly simplify the problem of constructing a generalized cluster algorithm by grouping those configurations or operations that are related by symmetry transformations (rotations and parity transformations) into a single representation of the topology. We will refer to such a representative configuration as a *link topology*, and to a representative operation as an *operator*.

Thus to simplify matters, we will initially consider different orientations of the configurations and the freeze/delete operations to be equivalent. In order to use the matrix formalism outlined above, we need to know the number of different link topologies, and the number of allowed orientations n_{ij} of each operator i for any given configuration of link topology j. If we adopt this approach, the elements of the allowance matrix $\mathbf{A}$ will be n_{ij}. Now if we solve equation 4, the probabilities $P_i(j)$ refer to the probability of choosing operator i for a given configuration of link topology j. Each orientation of the operator which is allowed for the particular configuration would then be assigned with equal probability, $1/n_{ij}$.

IV. A GENERALIZED CLUSTER ALGORITHM FOR THE 3-D FFIM

We first checked the performance of the standard SW cluster algorithm for this model. The bond percolation threshold for a cubic lattice is 0.2492 [13]. We measured the bond/link ratio of the SW algorithm for the 3-d Ising ferromagnet at its critical point to be 0.24078(4), with the maximum cluster containing on average 0.2599(1) of the total number of sites of a 16^3 lattice (in three dimensions the SW bonds do not percolate at the critical point, as they do in two dimensions). However at the critical point of the 3-d FFIM ($K_c = 0.738$ [19]), we found that the bond/link ratio was 0.5325(3), with the average maximum cluster size being 0.9942(1) on a 16^3 lattice. So while the SW algorithm produces good sized clusters and works very well for the ferromagnet, it works extremely poorly for the fully frustrated model – much worse than the Metropolis algorithm, in fact. The problem again is that we need to delete more bonds to break up the largest cluster.

We have generalized the KBD algorithm to three dimensions, with the unit cube replacing the plaquette as the tile used. This means that each freeze/delete operator acts on 12 links, rather than the 4 links of a square plaquette. The number of possible operations is therefore very large, and we have a lot of freedom in choosing them. For the plaquette there are only 2 possible energy states, whereas there are 7 for the cube ($E = -6J, -4J, -2J, 0, 2J, 4J, 6J$). A generalized cluster algorithm for the 3-d FFIM Ising model is therefore much more complicated than for the 2-d case.

All possible link topologies for this model are shown in Fig. 2, along with the energy of each. As in two dimensions, the most important operators are those which conserve the energy. These are the only operations allowed at zero temperature. In Fig. 3(G) we show all possible freeze/delete operators for which flipping the bonded spins will conserve the energy ($E = -6J$) of a ground state configuration (link topology 1 in Fig. 2). We have a large amount of freedom in choosing the other (energy non-conserving) freeze/delete operators. This makes it a simple task to find operators that will make the allowance matrix upper triangular, thereby guaranteeing a solution to equation 4.

In Fig. 3 we list some possible operators, grouped as to which link topologies (or equivalently in this case, configuration energies) they are allowed. For instance the operators in Fig. 3(G) are the energy conserving operators that can only act on ground state configurations of (energy $-6J$). Operators in Fig. 3(F) are only allowed for configurations of energy $-6J$ or $-4J$ (link topologies 1 and 2 in Fig. 2), whereas the operator in Fig. 3(A) that deletes all bonds is allowed for any configuration. For the groups E to G, which will make the largest contribution at low temperature, we show all possible operators. At the critical point, we found that approximately 85% of the cubes are acted upon by these operators. For the other groups we chose operators that delete as many bonds as possible, and that reduce to the 2-d plaquette operators on the faces of the cube. We believe that these are among the best possible choices of operators for these groups, and that choosing different operators will not change the results in any significant way.

If we choose one operator from each of the groups A-G, we will have an upper triangular allowance matrix. Choosing the first operator in each group, we get the allowance matrix in Table I. We used Mathematica [20] to solve the matrix equation 4. The numerical values for the probabilities at $K = K_c$ are shown in Table III. A list of the probabilities for arbitrary K is given in the appendix. We therefore have a valid algorithm that satisfies detailed balance, and is guaranteed to be ergodic (for $T > 0$) if we add a Metropolis sweep between each cluster update. Since the Metropolis algorithm is not ergodic at $T = 0$ for this model, we would need to prove that the generalized cluster algorithm is ergodic in order for it to be usable at zero temperature.

A problem with generalized cluster algorithms of this type is that there is no way of knowing *a priori* what kind of cluster size distribution they will give. The algorithm has to be constructed, programmed and the simulations run in order to obtain these results. We can however get an indication of the possible performance of the algorithm by looking at the zero temperature behavior, since the critical point for this model (and for many frustrated models) occurs at low temperature. It is much simpler to implement a zero temperature algorithm rather than the full algorithm, since we need only deal with ground state configurations and energy conserving operators.

We first implemented a simple zero temperature algorithm using only ground state configurations and operator $\mathtt{G}_1$ (i.e. the first operator in group $\mathtt{G}$) in Fig. 3. We found that all the sites in the lattice were bonded into a single cluster. This result might have been expected, since all of the energy conserving operators for this model have a bond/link ratio of 0.5, which is well above the percolation threshold for the simple cubic lattice (although it is lower than for the Swendsen-Wang algorithm). Since each of the energy conserving operations i occur for only one link topology (the ground state), the parameters C_i for these operators are unconstrained, which means that we are free to choose the probabilities of any of the energy conserving operators, as long as they sum to 1 [9, 11]. We can therefore apply the 3 energy conserving operators of Fig. 3($\mathtt{G}$) with probabilities p_1, p_2, and $p_3 = 1-p_1-p_2$ respectively. However, for all choices of these probabilities that we tried, the lattice froze into a single cluster at zero temperature, so this approach does not give a usable zero temperature cluster algorithm for the 3-d FFIM.

However the bond/link ratio, and consequently the size of the largest cluster, will decrease as the temperature is increased. It is unlikely that the bond/link ratio at K_c will decrease by the required factor of two to get near the percolation threshold, but since the two dimensional algorithm works very well with strongly percolating clusters, this may not be necessary for the algorithm to be effective. We therefore implemented the full algorithm described in the appendix, and simulated the model at its critical point, using the measured value $K_c = 0.738$ of Ref. [19], which gives the probabilities in Table III. We measured the bond/link ratio at K_c for a 16^3 lattice to be 0.401(1), which is still well above the percolation threshold. Our results for the average maximum cluster size and average number of clusters at the critical point for this lattice size and for various values of the probabilities p_i are shown in Table II. As can be seen, the maximum cluster sizes are discouragingly large for all of the probabilities tested.

We also tried using the other possible freeze/delete operators in Fig. 3(`E`) and (`F`), as well as combinations of these operators. Adding extra operators produces an under-determined system of equations, giving some free parameters in the choice of probabilities for these operators. However the results we obtained were virtually the same, irrespective of the different parameter values we tried. In all cases the average maximum cluster size is much too large to make effective changes in the configuration at each iteration. We checked the autocorrelation function [4, 5, 21] for the magnetization squared for the Metropolis algorithm and the generalized cluster algorithm with a Metropolis sweep at every iteration. At the critical point the two functions were almost identical, so the cluster update gives virtually no reduction of the autocorrelations. One must also take into account the fact that one iteration of the cluster algorithm takes approximately 3 times longer to perform than a Metropolis iteration. The cluster algorithm therefore performs much worse than the Metropolis algorithm for this model.

To check that the generalized cluster algorithm was implemented correctly, we also ran some simulations at lower values of K using our algorithm and compared the results with runs using the Metropolis algorithm. For these tests we did *not* use a Metropolis sweep between each sweep of the cluster algorithm. We ran at two values of K, a low value ($K = 0.25$) for which the maximum cluster size is only about 21% of the lattice volume, and a higher value ($K = 0.50$) for which the largest cluster takes up about 87% of the volume. The results are shown in Table IV, and agree with high precision for the two different Monte Carlo algorithms.

V. ATTEMPTS TO IMPROVE THE ALGORITHM

Including all possible orientations of operators and link topologies into the matrix formalism outlined in section III will generally give some extra free parameters for the probabilities of choosing the different orientations of the operators. For the generalized cluster algorithm for the 2-d FFIM, we found that we could slightly reduce the size of the largest cluster by using this freedom to bias the choice of operations so that only a particular orientation of a certain operator was used. This tuned algorithm showed substantially reduced critical slowing down, even though the largest cluster size was decreased by only a few percent [11].

We tried the same idea for the 3-d FFIM, hoping to reduce the size of the largest cluster at the critical point. As for the 2-d algorithm, only a subset of all possible orientations is required to construct a valid algorithm. The general algorithm we have described above chooses all orientations (and all subsets) with equal probability. In the 2-d case, we found that we achieved much better results if we chose only a single subset of orientations each time (that is, we assign one such subset with probability 1, and the rest with probability 0). We tried this for the 3-d algorithm for each of the energy conserving operators. In each case the lattice was still frozen into a single cluster at zero temperature, and unlike the 2-d case, this had little or no effect on the cluster size (and consequently the autocorrelations) at the critical point.

VI. CONCLUSIONS

The generalized cluster algorithm of KBD works extremely well for the square lattice FFIM. We have extended this work by using a simple method we have developed that enabled us to construct a generalized cluster algorithm for the cubic lattice FFIM, based on a unit cube rather than a plaquette. Although this algorithm does better than the Swendsen-Wang algorithm for this model, in that the number of frozen bonds is decreased, at zero temperature our algorithm still freezes the lattice into a single cluster. At the critical point it produces a single very large cluster and a number of very small clusters, and gives virtually no improvement over the standard Metropolis algorithm.

We have not tried all possible choices of operations and parameters for this method, so there is still a possibility that this type of method might work. However we have tried a wide enough variety of operations that we feel this is unlikely to be the case. We have obtained similarly discouraging results in constructing generalized cluster algorithms for other frustrated spin models, such as the 2-d Ising spin glass [11]. It therefore appears that some new ideas are needed in order to extend this method to general frustrated spin models.

ACKNOWLEDGMENTS

We are grateful to John Apostolakis, Ofer Biham, Danny Kandel, Enzo Marinari and Alan Sokal for valuable discussions. This work was sponsored in part by Department of Energy grants DE-FG03-85ER25009 and DE-AC03-81ER40050, and by a grant from the IBM corporation.

APPENDIX A: PROBABILITIES FOR THE ALGORITHM

Probabilities $P(\mathtt{O},\mathtt{T})$ for the generalized cluster algorithm for the 3-d fully frustrated Ising model, for an arbitrary coupling K. $\mathtt{O}$ denotes an operator from Fig. 3 and $\mathtt{T}$ denotes a link topology from Fig. 2. Only the non-zero values are listed. The values for $K = K_c$ are given in Table III. Note that the probabilities for each link topology sum to 1, as they must.

$$P(A_1,1) = e^{-12K}$$

$$P(B_1,1) = 6(e^{-10K} - e^{-12K})$$

$$P(C_1,1) = 9(e^{-8K} - 2e^{-10K} + e^{-12K})$$

$$P(D_1,1) = 2(e^{-6K} - 3e^{-8K} + 3e^{-10K} - e^{-12K})$$

$$P(E_1,1) = 3(e^{-4K} - 2e^{-6K} + 2e^{-8K} - 2e^{-10K} + e^{-12K})$$

$$P(F_1,1) = 3(e^{-2K} - 2e^{-4K} + 2e^{-6K} - 4e^{-8K} + 5e^{-10K} - 2e^{-12K})$$

$$P(G_1,1) = 1 - 3e^{-2K} + 3e^{-4K} - 2e^{-6K} + 3e^{-8K} - 3e^{-10K} + e^{-12K}$$

$$P(A_1,2) = e^{-10K}$$

$$P(B_1,2) = 5(e^{-8K} - e^{-10K})$$

$$P(C_1,2) = 6(e^{-6K} - 2e^{-8K} + e^{-10K})$$

$$P(D_1,2) = 2(e^{-4K} - 3e^{-6K} + 3e^{-8K} - e^{-10K})$$

$$P(E_1,2) = 2(e^{-2K} - 2e^{-4K} + 2e^{-6K} - 2e^{-8K} + e^{-10K})$$

$$P(F_1,2) = 1 - 2e^{-2K} + 2e^{-4K} - 4e^{-6K} + 5e^{-8K} - 2e^{-10K}$$

$$P(A_1,3) = e^{-8K}$$

$$P(B_1,3) = 4(e^{-6K} - e^{-8K})$$

$$P(C_1,3)=4(e^{-4K}-2e^{-6K}+e^{-8K})$$

$$P(D_1,3)=2(e^{-2K}-3e^{-4K}+3e^{-6K}-e^{-8K})$$

$$P(E_1,3)=1-2e^{-2K}+2e^{-4K}-2e^{-6K}+e^{-8K}$$

$$P(A_1,4)=e^{-6K}$$

$$P(B_1,4)=3(e^{-4K}-e^{-6K})$$

$$P(C_1,4)=3(e^{-2K}-2e^{-4K}+e^{-6K})$$

$$P(D_1,4)=1-3e^{-2K}+3e^{-4K}-e^{-6K}$$

$$P(A_1,5)=e^{-4K}$$

$$P(B_1,5)=2(e^{-2K}-e^{-4K})$$

$$P(C_1,5)=1-2e^{-2K}+e^{-4K}$$

$$P(A_1,6)=e^{-2K}$$

$$P(B_1,6)=1-e^{-2K}$$

$$P(A_1,7)=1$$

REFERENCES

[1] R.H. Swendsen and J.-S. Wang, Phys. Rev. Lett. **58**, 86 (1987); J.-S. Wang and R.H. Swendsen, Physica A **167**, 565 (1990).

[2] U. Wolff, Phys. Rev. Lett. **62**, 361 (1989).

[3] N. Metropolis *et al.*, J. Chem. Phys. **21**, 1087 (1953).

[4] K. Binder ed., *Monte Carlo Methods in Statistical Physics*, (Springer-Verlag, Berlin, 1986); K. Binder and D.W. Heermann, *Monte Carlo Simulation in Statistical Physics*, (Springer-Verlag, Berlin, 1988); H. Gould and J. Tobochnik, *An Introduction to Computer Simulation Methods, Vol. 2*, (Addison-Wesley, Reading, Mass., 1988).

[5] A.D. Sokal, in *Computer Simulation Studies in Condensed Matter Physics: Recent Developments*, eds. D.P. Landau *et al.* (Springer-Verlag, Berlin-Heidelberg, 1988); A.D. Sokal, in Proc. of the International Conference on Lattice Field Theory, Tallahassee, October 1990, Nucl. Phys. B (Proc. Suppl.) **20**, 55 (1991).

[6] G. Toulouse, Commun. Phys. **2**, 115 (1977).

[7] K. Binder and A.P. Young, Rev. Mod. Phys. **58**, 801 (1986); D. Chowdhury, *Spin Glasses and Other Frustrated Systems*, (Princeton University Press, Princeton, NJ, 1986); M. Mezard and G. Parisi, *Spin Glass Theory and Beyond*, (World Scientific, Singapore, 1987).

[8] D. Kandel, R. Ben-Av and E. Domany, Phys. Rev. Lett. **65**, 941 (1990).

[9] D. Kandel and E. Domany, Phys. Rev. B **43**, 8539 (1991).

[10] D. Kandel, R. Ben-Av and E. Domany, Phys Rev B **45**, 4700 (1992).

[11] P.D. Coddington and L. Han, "On generalized cluster algorithms for frustrated spin models", NPAC technical report SCCS-527, cond-mat/9402030.

[12] J. Villain, J. Phys. C **10**, 1717 (1977).

[13] D. Stauffer, *Introduction to Percolation Theory*, (Taylor and Francis, London, 1985).

[14] B. Derrida *et al.*, J. Physique **40**, 617 (1979).

[15] S.F. Chiu, G. Forgacs, and D.M. Hatch, Phys. Rev. B **25**, 6952 (1982).

[16] D. Blankschtein, M. Ma and A.N. Berker, Phys. Rev. B **30**, 1362 (1984).

[17] S. Kirkpatrick, *Disordered Systems and Localization*, (Springer, Berlin, 1981).

[18] G.S. Grest, J. Phys. C **18**, 6239 (1985).

[19] H.T. Diep, P. Lallemand and O. Nagai, J. Phys. C **18**, 1067 (1985).

[20] S. Wolfram, *Mathematica: a system for doing mathematics by computer*, (Addison-Wesley, Redwood City, Calif., 1990).

[21] C.F. Baillie and P.D. Coddington, Phys. Rev. B **43**, 10617 (1991).

TABLES

TABLE I. The allowance matrix, written as the number of possible orientations n_{ij} of the freeze/delete operators i of Fig. 3 for a given configuration of the link topologies j shown in Fig. 2.

		Operations						
Energy	Config	$\mathtt{A}_1$	$\mathtt{B}_1$	$\mathtt{C}_1$	$\mathtt{D}_1$	$\mathtt{E}_1$	$\mathtt{F}_1$	$\mathtt{G}_1$
$-6J$	1	1	6	9	2	3	6	1
$-4J$	2	1	5	6	2	2	2	0
$-2J$	3	1	4	4	2	1	0	0
$0J$	4	1	3	3	1	0	0	0
$2J$	5	1	2	1	0	0	0	0
$4J$	6	1	1	0	0	0	0	0
$6J$	7	1	0	0	0	0	0	0

TABLE II. The average maximum cluster size for different probabilities p_1, p_2, p_3 of the 3 energy conserving operators at the critical point K_c of the 3-d FFIM, for a 16^3 lattice.

Probabilities of Operators			Maximum cluster size
p_1	p_2	p_3	
1	0	0	0.96208(17)
0	1	0	0.96212(17)
0	0	1	0.95745(7)
$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0.96082(10)
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	0.95981(15)

TABLE III. Probabilities for the freeze/delete operators of Fig. 3 for the cubic lattice FFIM at the critical point K_c.

		Operations						
Energy	Config	$\mathtt{A}_1$	$\mathtt{B}_1$	$\mathtt{C}_1$	$\mathtt{D}_1$	$\mathtt{E}_1$	$\mathtt{F}_1$	$\mathtt{G}_1$
$-6J$	1	0.0001	0.0029	0.0146	0.0110	0.0981	0.4196	0.4536
$-4J$	2	0.0006	0.0105	0.0426	0.0480	0.2862	0.6120	0
$-2J$	3	0.0027	0.0368	0.1243	0.2099	0.6262	0	0
$0J$	4	0.0119	0.1209	0.4081	0.4591	0	0	0
$2J$	5	0.0522	0.3526	0.5952	0	0	0	0
$4J$	6	0.2285	0.7715	0	0	0	0	0
$6J$	7	1	0	0	0	0	0	0

TABLE IV. Comparison of results for the Metropolis algorithm and the generalized cluster algorithm for 10^6 sweeps of an 8^3 lattice at $K = 0.25$ and $K = 0.50$. Quantities measured are the average energy per link E and the average magnetization per site M, and their variances (scaled by the lattice volume) C and χ.

	E	C	M	χ	Max cluster size
$K = 0.25$					
Metropolis	0.61353(7)	1.7065(24)	0.05251(6)	0.8018(18)	
Cluster	0.61365(7)	1.7073(26)	0.05244(5)	0.8021(13)	0.21127(11)
$K = 0.50$					
Metropolis	0.91864(7)	0.9170(15)	0.08765(20)	2.208(10)	
Cluster	0.91866(14)	0.9150(24)	0.08718(36)	2.195(19)	0.86611(2)

FIGURES

FIG. 1. Ferromagnetic (solid lines) and anti-ferromagnetic (dotted lines) links for the simple cubic lattice FFIM.

FIG. 2. All possible link topologies on the unit cube for the 3-d FFIM. Satisfied links are shown as solid lines.

FIG. 3. Some possible freeze/delete operators for the cubic lattice FFIM. Frozen bonds are shown as bold lines. `A` shows the operator that deletes all bonds, the other groups correspond to operators that are not allowed (i.e. $a_{ij} = 0$) for cubes with energies greater than $+4J$ (`B`), $+2J$ (`C`), 0 (`D`), $-2J$ (`E`), $-4J$ (`F`), and $-6J$ (`G`), so `G` shows energy conserving operators. Only `E`-`G` show all possible operators of this kind.

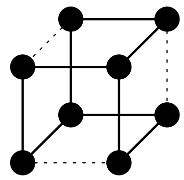

Fig. 1.

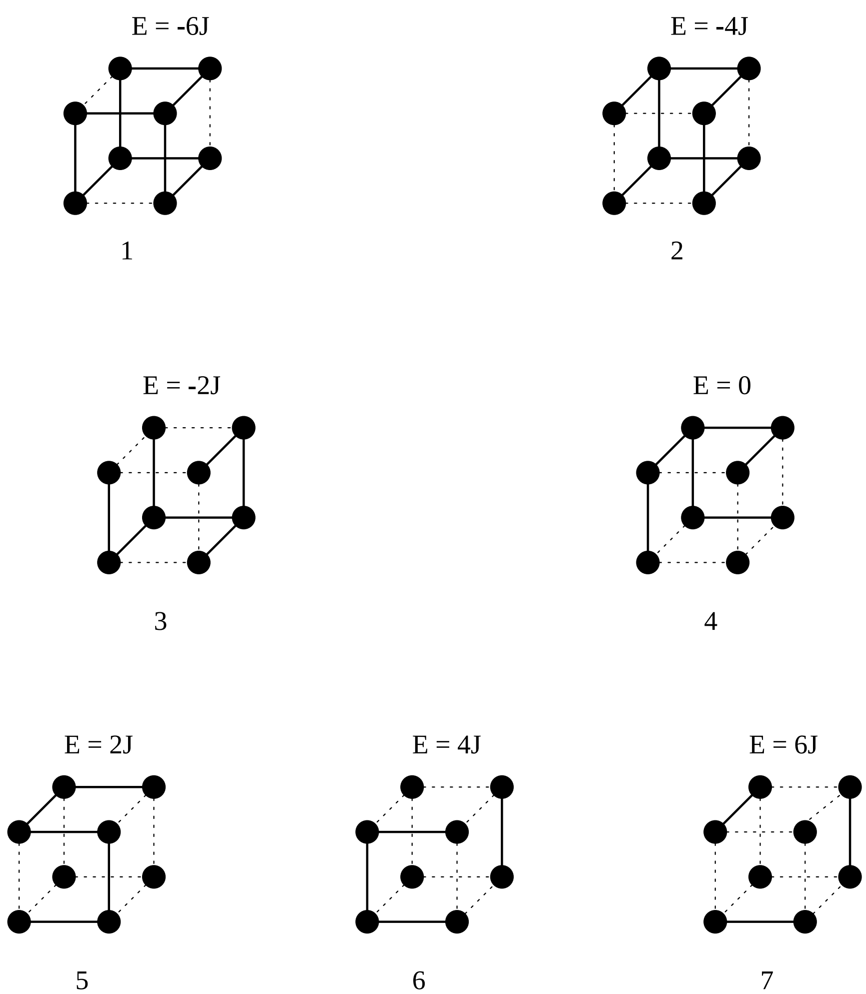

Fig. 2.

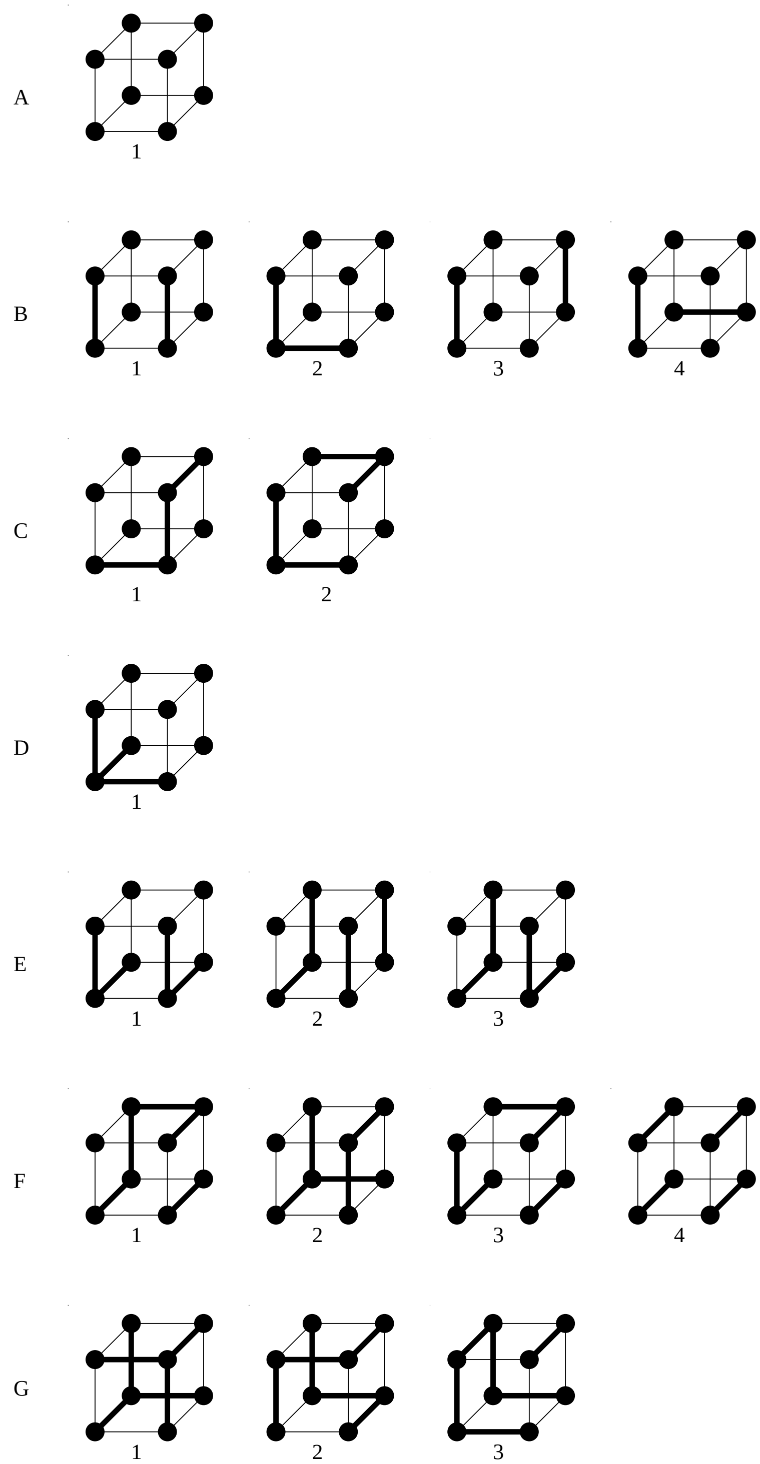

Fig. 3.