Exact Ground States of Large Two-Dimensional Planar Ising Spin Glasses

G. Pardella[1] and F. Liers[1]

[1] *Institut für Informatik, Universität zu Köln, Pohligstraße 1, D-50969 Köln, Germany.**

(Dated: October 23, 2008)

Studying spin-glass physics through analyzing their ground-state properties has a long history. Although there exist polynomial-time algorithms for the two-dimensional planar case, where the problem of finding ground states is transformed to a minimum-weight perfect matching problem, the reachable system sizes have been limited both by the needed CPU time and by memory requirements. In this work, we present an algorithm for the calculation of exact ground states for two-dimensional Ising spin glasses with free boundary conditions in at least one direction. The algorithmic foundations of the method date back to the work of Kasteleyn from the 1960s for computing the complete partition function of the Ising model. Using Kasteleyn cities, we calculate exact ground states for huge two-dimensional planar Ising spin-glass lattices (up to 3000^2 spins) within reasonable time. According to our knowledge, these are the largest sizes currently available. Kasteleyn cities were recently also used by Thomas and Middleton in the context of extended ground states on the torus. Moreover, they show that the method can also be used for computing ground states of planar graphs. Furthermore, we point out that the correctness of heuristically computed ground states can easily be verified. Finally, we evaluate the solution quality of heuristic variants of the Bieche et al. approach.

Keywords: Ising spin glass, Kasteleyn cities, perfect matching

I. INTRODUCTION

The determination of spin-glass ground states has raised the interest of both physicists and computer scientists. For an introduction we refer to [1–3]. On the one hand, an analysis of the ground-state properties sheds light on the ruling physics of the system. On the other hand, several different algorithms have been developed and used for the ground-state determination of different models.

For the two-dimensional Edwards-Anderson model [4] (EA) with free boundaries in at least one direction, ground states can be determined exactly with fast algorithms. In fact, the problem is solvable in time bounded by a polynomial in the size of the input. The latter can be achieved by a transformation to a well known graph theoretical problem —the minimum-weight matching problem, for which efficient implementations exist. For general non-planar or three- or higher-dimensional lattices, however, calculating exact ground states is NP-hard [5]. Loosely speaking, this means we cannot expect to be able to design a polynomial-time solution algorithm. In practice, one can use, e.g. branch-and-cut algorithms [6].

In this work, we focus on the polynomially solvable case of two-dimensional lattices with free boundaries in at least one direction. We first review and compare the main known approaches which are those of Bieche et al. [7] and of Barahona [8, 9]. Then we present the approach inspired by Kasteleyn [10]. All method basically follow the same idea: An associated graph is constructed in which a minimum-weight perfect matching is determined that is used to construct an exact ground state. Differences occur in the constructed associated graph. It turns out that the approach inspired by Kasteleyn is the most favorable. In fact, using the latter method, we can determine exact ground states for lattice sizes up to 3000^2, while the possible sizes computed earlier with heuristic variants of the approach of Bieche et al. were considerably smaller. In a forthcoming article [11], we will analyze the physics of the system. Kasteleyn cities were recently also used by Thomas and Middleton [12]. While focussing on extended ground states on the torus, they show that the Kasteleyn-city approach can be successfully used in the planar case, too. Furthermore, they compared their implementation with an implementation of Barahona's method. It turned out that the approach with Kasteleyn cities is less memory consumptive and faster. Apart from this recent work, we are not aware of other computational studies using Barahona's method.

We show how to either prove correctness of heuristically determined ground states or how to correct them using linear programming. Despite the fact that this is fast, it is still advantageous to use the method based on Kasteleyn cities. Finally, we evaluate the quality of the solutions generated with heuristic variants of the Bieche et al. approach.

The outline of the article is as follows. In Section II, we introduce the model. In Section III we introduce definitions necessary for the literature review in Section IV. Finally, we report the results in Section V.

*E-mail: {pardella|liers}@informatik.uni-koeln.de; Visit: http://cophy.informatik.uni-koeln.de/

II. THE MODEL

In the Edwards-Anderson model N spins are placed on a lattice. We focus on quadratic ($N = L^2$) lattices with free boundary conditions in at least one direction. Toric boundary conditions may be applied to at most one lattice axis. The Hamiltonian of the system is

$$\mathcal{H} = -\sum_{<i,j>} J_{ij} S_i S_j, \tag{1}$$

where the sum runs over all nearest-neighbor sites. Each spin S_i is a dynamical variable which has two allowed states, $+1$ and -1. The coupling strengths J_{ij} between spins i and j are independent identically distributed random variables following some probability distribution. The concentration of anti-ferromagnetic ($J_{ij} < 0$) and ferromagnetic bonds ($J_{ij} > 0$) depends on the underlying distribution. The Gaussian and the bimodal $\pm J$ distributions are often used. A spin configuration attaining the global minimum of the energy function $\mathcal{H}$ is called a ground state.

III. PRELIMINARIES

In this section, we briefly summarize some basic definitions from graph theory. For further details, we refer to [13–15] and the references therein. We associate a spin-glass instance with a *graph* $G = (V, E)$ with vertices V (spin sites) and edges E (bonds). The edge set consists of unordered pairs (i, j), with $i, j \in V$. More specifically, for two-dimensional $K \times L$ lattices with free boundaries, the graph is called a *grid graph* and is denoted by $G_{K,L}$. In case periodic boundaries are present in one direction, we call the graph a *half-torus* or a *ring*. The *degree* $deg(v)$ of a vertex v is the number of edges $(v, w_i) \in E$ incident at v. A *path*, $\pi = v_1, v_2, ..., v_k,\ v_i \in V$, is a sequence of vertices such that $(v_1, v_2), (v_2, v_3), ..., (v_{k-1}, v_k)$ are edges of G and the v_i are distinct. A closed path $\pi = v_1, v_2, ..., v_k, v_1$ is a *cycle*.

In many applications a rational cost or a weight $w(e)$ is associated with an edge e. Let $G = (V, E)$ be a weighted graph. For each (possibly empty) subset $Q \subseteq V$, a *cut* $\delta(Q)$ in G is the set of all edges with one vertex in Q and the other in $V \setminus Q$. The weight of a cut is given by $w(\delta(Q)) = \sum_{e=(v,w)\in\delta(Q)} w(e)$. A *minimum cut* (MIN-CUT) asks for a cut $\delta(Q)$ with minimum weight among all vertex sets $Q \subseteq V$. Let K_n denote the *complete graph* with n vertices and edge set $E = V \times V$. A *subgraph* of G is a graph G_H such that every vertex of G_H is a vertex of G, and every edge of G_H is an edge in G also. $G = (V, E)$ is called *Eulerian* if and only if each vertex of G has even degree. A graph G is *planar* if it can be drawn in the plane in such a way that no two edges meet each other except at a vertex. Any such drawing is called a *planar drawing*. Any planar drawing of a graph G divides the plane into regions, called *faces*. One of these faces is unbounded, and called the *outer face* or *unbounded face*. A *geometric dual graph* [16] G_D of a connected planar graph G is a graph G_D with the property that it has a vertex for each face of G and an edge for each edge touching two neighboring faces in G.

A *matching* in a graph $G = (V, E)$ is a set of edges $M \subseteq E$ such that no vertex of G is incident with more than one edge in M. A matching M is *perfect* if every vertex is incident to an edge in the matching. A maximum matching is a matching of maximum weight $w(M) = \sum_{e\in M} w(e)$. Solving the perfect matching problem on general graphs in time bounded by a polynomial in the size of the input remained an elusive goal for a long time until Edmonds [17, 18] gave the first polynomial-time algorithm — the blossom algorithm. More details about matching theory can be found in [19].

IV. REVIEW OF THE KNOWN ALGORITHMIC APPROACHES

Bieche et al. [7] showed that the problem of finding a ground state for two-dimensional planar Ising spin glasses can be transformed to a well known graph theoretical problem — the minimum-weight perfect matching problem (MWPM) on general graphs. The method follows the scheme shown in Algorithm 1 in which an optimum matching is used to construct a spin configuration minimizing the total energy.

Most commonly used exact methods, like the approaches of Bieche et al. [7] and Barahona [8, 9], follow this scheme. In the following, we briefly summarize these two methods. Afterwards, we present a method following the construction introduced by Kasteleyn [10]. More details can be found in the recent tutorial [1] on algorithms for computing ground states in two-dimensional Ising spin glasses.

Review of Exact Methods

1. The Approach of Bieche et al.

Bieche et al. [7] consider the weighted grid graph $G_{K,L} = (V, E)$ where each vertex $i \in V$ is assigned an initial spin value $S_i^0 = \pm 1$. Each edge $e = (i, j)$ receives a weight $w(e) = -J_{ij} S_i^0 S_j^0$, cf. Fig. 1. Often, the trivial configuration $\mathcal{S}^0 = +1\ \forall S_i,\ i \in V$ is used.

An instance can not only be described in terms of spins and bonds, but also by *frustrated plaquettes* and *paths of broken edges*. Plaquettes consist of the 4-cycles in the graph. An edge is said to be satisfied if it attains its minimal weight ($-J_{ij} S_i^0 S_j^0 = -|J_{ij}|$), otherwise it is called unsatisfied. A plaquette is frustrated if there is no spin configuration satisfying all edges. In this case the plaquette has an odd number of negative edges. For the

Algorithm 1 Calculate a ground state of a $K \times L$ spin glass

Input: Planar grid graph $G_{K,L}$
Output: Spin configuration $\mathcal{S}$ minimizing the total energy $\mathcal{H}$
1. Construct an appropriate dual graph $\tilde{G}$
2. Calculate a minimum-weight perfect matching M in $\tilde{G}$
3. Use M to compute a spin configuration $\mathcal{S}$ and the corresponding energy $\mathcal{H}$
4. **return** $\mathcal{S}$ and $\mathcal{H}$

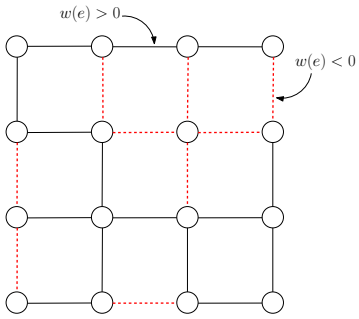

FIG. 1: $G_{4,4}$ grid graph. Dashed lines indicate negative edge weights (Color online).

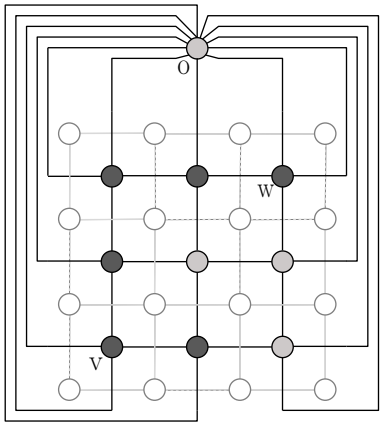

FIG. 2: Geometric dual graph G_D of the grid graph $G_{4,4}$ shown in Figure 1 which is seen translucent. Dark gray vertices represent frustrated plaquettes (assuming the trivial configuration $\mathcal{S}^0 = \{+1 \mid \forall i \in V\}$), and light gray vertices are unfrustrated plaquettes (Color online).

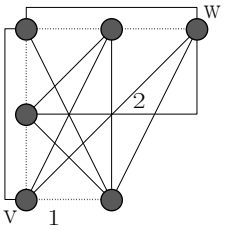

FIG. 3: Graph G_F of the grid graph shown in Fig. 1. Continuous edges indicate distance 2 between vertices, and dotted edges indicate distance 1 (Color online).

remainder let F be the set of frustrated plaquettes in $G_{K,L}$ and P the set of all plaquettes in $G_{K,L}$.

Bieche et al. identify the frustrated plaquettes as vertices of a graph, $G_F = (F, E_F)$ with $F = \{f \mid f$ is a frustrated plaquette in $G\}$ and $E_F = F \times F$. Each edge $e = (f_i, f_j) \in E_F$ is assigned a weight $w(e)$ equal to the sum of the absolute weights of the edges in $G_{K,L}$ crossed by a minimum path connecting f_i with f_j. Figure 3 shows the graph G_F for the grid graph of Figure 1. The underlying dual graph is shown in Figure 2.

It is easy to see that minimizing the sum of the weights of unsatisfied edges connecting frustrated plaquettes yields a spin configuration of minimum energy. The latter is achieved by determining a minimum-weight perfect matching in G_F. Finding a ground state is thus reduced to finding a minimum-weight perfect matching M of the graph G_F, and its energy is given as:

$$\begin{aligned} \mathcal{H} &= -\sum_{<i,j>} J_{ij} S_i S_j \\ &= -\sum_{<i,j>} |J_{ij}| + 2 \sum_{unsatisfied\ edges} |J_{ij}| \\ &= -\sum_{<i,j>} |J_{ij}| + 2w(M) \end{aligned}$$

For a detailed description of this method we refer to [1].

2. *Limits of Bieche's Approach*

The approach of Bieche et al. is simple and intuitive, but comprised two major practical obstacles. First of all, in order to obtain the dual edge weights, one has to calculate shortest paths in G_D between all pairs of vertices. Although this can be done in time and space bounded by a polynomial in the size of the input, the calculations can take long in practice or require a large amount of memory. Equipped with the weights, one can construct the complete graph G_F of frustrated plaquettes. Treatable system sizes are practically limited by the number of edges present in G_F. Assuming 32 bits for representing an edge, one needs nearly 4 GB of memory just for representing E_F for a 300×300 grid graph, assuming 50% of the plaquettes are frustrated. For a 400×400 grid graph, almost 12 GB of memory are necessary, which goes beyond the hardware resources available in ordinary modern computers.

3. Simple Improvement for $\pm J$-distributed Samples

For $\pm J$ distributed instances, one can obtain the length of the shortest paths directly without shortest paths calculations. For this, we project the geometric dual graph G_D on the plane so that each vertex v is provided with definite coordinates (x_v, y_v) preserving the distance function using rectilinear edges. Different vertices are assigned to different coordinates. Then the length of paths between i and j, $\pi = (x_i, y_i), \sigma_1, \sigma_2 \ldots, \sigma_r, (x_j, y_j)$ (with $\sigma_l = (x_l, y_l) = l \in V$) traversing only through the grid graph $G_{K,L}$ without crossing its border, is given as the Manhattan distance $c = |x_i - x_j| + |y_i - y_j|$. This value is to be compared with the length of the path passing through the outer face o, $\pi_o = (x_i, y_i), \sigma_1, \ldots, \sigma_k, o, \sigma_{k+1}, \ldots \sigma_r, (x_j, y_j)$. The weight of this path is given as $c_o = \min\{x_i, y_i, K - x_i, L - y_i\} + \min\{x_j, y_j, K - x_j, L - y_j\}$. The shortest path from i to j is the shorter of the two. In a half-torus graph, analogous calculations can be performed.

4. The Approach of Barahona

Barahona [8, 9] constructs the geometric dual graph that contains a vertex for each plaquette and edges in case the corresponding two plaquettes share an edge. Here the outer face is also interpreted as a plaquette. In formulas, $G_D = (P, E_D)$ of $G_{K,L}$, where $P = \{p \mid p \text{ is a plaquette in } G_{K,L}\} \cup \{o \mid o \text{ is the outer face plaquette}\}$ and $E_D = \{e = (p_i, p_j) \mid \forall p_i, p_j \in P,\ p_i \cap p_j \neq \emptyset\}$. Each dual edge is assigned a weight according to the *absolute* weight of the edge in $G_{K,L}$ crossed by the dual edge. Vertices $p_i \in P$ are called *odd* if they represent a frustrated plaquette, otherwise *even*.

Subsequently, the graph G_D is transformed into a graph G^*. In order to do this, first every vertex $p_i \in P$ with $deg(p_i) > 3$ is expanded to $(deg(p_i) - 2)$ copies of degree 3. Any even vertex remains even, expanding an odd vertex makes one of its copies (arbitrarily) odd and the others even. From now on, one works with vertices of degree 3 only. Next, each vertex is transformed to a K_3 subgraph: Each edge incident to an even vertex is replaced by an intermediate vertex and two edges. At most two new vertices are inserted for each edge connecting two even vertices. Original edges keep their weight, new edges obtain weight zero. For the details, we refer to [8, 9].

On G^* a MWPM is computed. Any even vertex has an even number (including zero) of "outgoing" matching edges, however, any odd vertex has an odd count of those edges. After the matching is calculated, the afore expanded vertices are shrunken, and the remaining matching edges raise shortest paths connecting frustrated plaquettes. As the total length of the induced paths is minimal among all possible paths, the induced set of unsatisfied edges has minimum weight. Following Bieche, this corresponds to a configuration of minimum weight.

5. Evaluation of Barahona's Approach

Barahona's transformation consists of two steps and is a bit more involved than the method of Bieche et al. For a quadratic $L \times L$ grid graph with free boundary conditions, G^* has $|V^*| \approx 12(L-1)[(L-1)+2]-12-b_{odd}$ many vertices, where $b_{odd} = 3 \cdot |\{\text{odd vertices}\}|$ is the number of odd vertices in the graph G_D. The graph G^* is sparse as each vertex in G^* has degree 3, $|E^*| = \frac{3}{2}|V^*|$. Assuming 50% frustrated plaquettes, the number of vertices increases approximately by a factor of 10. Given that for bigger lattices this transformation needs less space than the one by Bieche et al., it is preferable to the former. However, in the next section we describe a method that works with an even smaller graph.

6. The New Approach —Following in Kasteleyn's Footsteps

In this section, we follow an idea first described by Kasteleyn [10, 20] and Fisher [21]. In these works, the goal was to calculate the configurational partition function for dimer coverings on a lattice. The authors exploited that the calculation of the partition function of the Ising model can be reduced to the number of ways in which a given number of edges can be selected to form closed polygons [22], i.e., a polygon configuration such that each lattice vertex has even degree of selected incident polygon edges. The latter can be computed as follows. First, one constructs a so-called cluster lattice graph which is generated by replacing each vertex of the lattice by a Kasteleyn city (a K_4 subgraph). Now the expanded graph is oriented such that the associated skew-symmetric Matrix D shows the property that $|\mathrm{Pf}(D)|$, where $\mathrm{Pf}(D)$ denotes the Pfaffian of the skew-symmetric matrix, gives the number of dimer coverings. As there is a one-to-one correspondence between the number of polygon configurations and the number of dimer coverings, this method yields the generating function for polygon configurations and therefore the generating function for the Ising model.

Thomas and Middleton used Kasteleyn cities for calculating extended ground states on the torus in order to gain relevant information about the physics of spin glasses on toroidal lattices. Furthermore, they point out that the method yields an exact ground-state algorithm on planar lattices.

A closely related approach was used later by Galluccio et al. to design an exact algorithm for the computation of the partition function for the Ising problem that runs in polynomial time for several models of interest [23–25], e.g., for two-dimensional toroidal lattices with $\pm J$ distribution.

Here, we focus on planar grid graphs. The distribution of the edge weights is arbitrary. $G_{K,L}$ is transformed

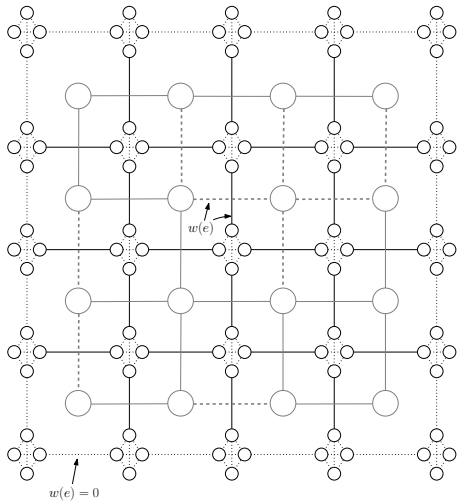

FIG. 4: Expanded dual graph G_{K_4} for the grid graph $G_{4,4}$ (Color online).

into a pseudo-dual graph G_{K_4} as shown in Figure 4. For simplicity, we confine ourselves to quadratic grid graphs in the following, however all results can be easily transformed to general grid graphs.

Formally, first the geometric dual of the grid graph $G_{L,L}$ is constructed, then the outer face vertex is expanded into $2L+1$ (L in the half-torus case) copies, such that the resulting graph is an intermediate grid graph $G_{L+1,L+1}$ ($G_{L,L+1}$). The edge weights of $G_{L+1,L+1}$ are set to the weights of the edges of $G_{L,L}$ that are crossed by the edges of $G_{L+1,L+1}$. Edges that do not cross any other edge obtain weight zero. Next, each vertex of $G_{L+1,L+1}$ is expanded to a K_4 subgraph. Again, newly constructed edges receive weight zero.

The transformation for the half-torus graph is done similarly, but the intermediate graph is a *grid-half-torus graph* $G_{L,L+1}$ which is a grid with $L-1$ additional edges. Edge weights are set as just described. Finally, all vertices are expanded to a K_4 subgraph as before. We denote by G_{inter} the intermediate graph either for the underlying grid or half-torus graph.

On the transformed graph G_{K_4} we calculate a minimum-weight perfect matching M.

The next step is to shrink all the K_4-subgraphs back, resulting in the graph G_{inter}. Also all copies of the outer face vertex are shrunken. Dealing again with the geometric dual graph of $G_{L,L}$, we take the subgraph $G_S = (Q, \delta(Q))$ of the geometric dual graph that consists only of dual edges that were matched, and all dual vertices with degree greater zero restricted to matched edges. This subgraph G_S is an Eulerian graph as each dual vertex is incident to an even number of matching edges. It is well known that there exists a one-to-one correspondence between Eulerian subgraphs in the dual graph and cuts in the original graph. So, Q defines a cut $\delta(Q)$ in the graph $G_{L,L}$, cf. Fig. 5.

In order to show the correctness of the transformation,

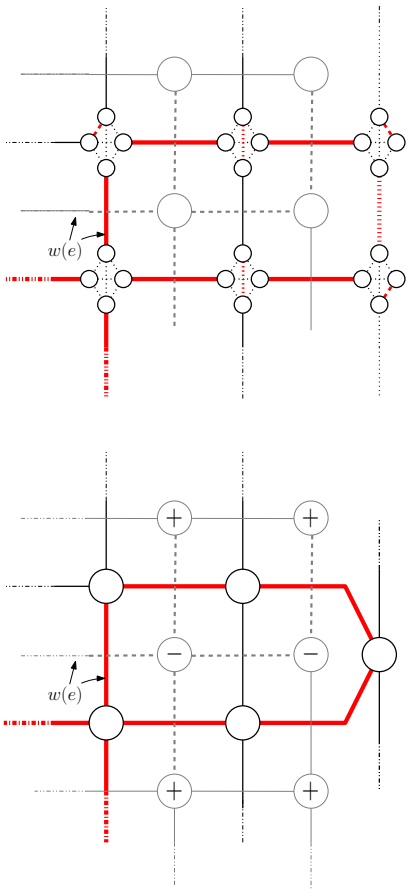

FIG. 5: Backward transformation. A matching (thick lines) in G_{K_4} induces a Eulerian subgraph in G_D and therefore a cut in the grid or half-torus graph. The vertex partition consists of the vertices with spin value $+1$ ($+$) in one partition and those with spin value -1 ($-$) in the other (Color online).

we exploit that each ground state corresponds exactly to a MIN-CUT $\delta(Q)$ in the grid graph $G_{L,L}$. A cut separates the vertex set into two disjoint sets W and $V \setminus W$. Vertices in the same partition get assigned the same spin value. Cut edges are those connecting a pair of vertices with different spin values. The Hamiltonian can be stated as:

$$\begin{aligned} \mathcal{H} &= -\sum_{<i,j>} J_{ij} S_i S_j \\ &= -\sum_{<i,j>} J_{ij} + 2w(\delta(Q)) \end{aligned} \tag{2}$$

We show that the edge set determined with the method described above corresponds to a minimum cut.

$$\begin{aligned} w(M) &= \sum_{\tilde{e} \in E \cap M} w(\tilde{e}) \\ &\underbrace{=}_{w(\tilde{e})=w(e)} \sum_{e \in \delta(Q)} w(e) \\ &= w(\delta(Q)) \end{aligned}$$

As $w(M)$ is the weight of a minimum-weight perfect matching, the weight of the subgraph G_S is minimum, and thus also the weight of the cut $\delta(Q)$.

As Kasteleyn's original method yields the complete partition function, one might ask why the algorithm was modified so that only the ground state is determined. The reason for this is twofold. First of all, minimum-weight perfect matchings can be computed in graphs with millions of vertices, provided they are sparse, which allows us to go to very large system sizes. Furthermore, the partition function does not encode the ground-state configuration itself but only its energy.

7. *Advantages of the New Approach*

The method is intuitive and its implementation is straightforward. We present some computational results in Section V. For a quadratic $L_1 \times L_1$ grid graph G_{L_1,L_1} we construct a graph $G_{K_4}^{L_1,L_1}$ with $|V_{K_4}^{L_1,L_1}| = 4(L_1+1)(L_1+1)$ vertices and $2|V_{K_4}^{L_1,L_1}|-4(L_1+1)$ edges. For an $L_2 \times L_2$ ring graph G_{L_2,L_2} the construction yields a graph $G_{K_4}^{L_2,L_2}$ with $|V_{K_4}^{L_2,L_2}| = 4L_2(L_2+1)$ vertices and $2|V_{K_4}^{L_2,L_2}| - 2L_2 - 4$ edges. In any case, the resulting graphs are very sparse. More specifically, let us compare for an $L \times L$ grid the sizes of the graphs in which a minimum-weight perfect matching is computed. Assuming that around 50% of the original plaquettes are frustrated, the graph based on Kasteleyn cities contains only about one third of the number of vertices and one fourth of the number of edges contained in the graph constructed with Barahona's method.

As the running time of the matching algorithm scales with the number of vertices and edges in the graph, the Kasteleyn construction is preferable to both the Bieche et al. and the Barahona construction.

Computing Domain Walls

For the computation of domain walls, we follow the usual approach [26–28]. A ground state of the system is calculated, having energy E_0. To introduce a domain wall, the system is then usually perturbed by flipping all couplings along a row or a column in the lattice. The ground state for this new system is calculated, having energy E_0^{pert}. The domain-wall energy for a given sample is then given by $\Delta E = |E_0^{pert} - E_0|$. We proceed as described in [29, 30], by first determining a ground state of an EA spin glass with periodic boundary conditions in one direction, say along the x-axis. Then the signs of L edges in one column along the y-axis are flipped. The symmetric difference of these ground states yields a domain wall.

With the help of linear programming [31, 32], one does not need to calculate the second ground state from scratch but can flip the weight of the specific L edges one by one, each followed by a reoptimization step. It is also possible to flip all signs at the same time and do a global reoptimization step. However, in practice for ± 1 distributed weights it often does not pay off to do the reoptimization steps, and so we calculate both ground states from scratch.

Modified Approach of Bieche et al. — Review of a Heuristic Method

As argued above, the original approach of Bieche et al. suffers from an extensive memory usage. In order to overcome these limits, some modifications have been proposed that yield heuristic methods for low-energy states that are however not necessarily exact ground states. In these heuristics, a reduced graph $\tilde{G}_{red}$ is used instead of a complete one. An approach often used is to introduce in $\tilde{G}_{red}$, only those edges with weight less than or equal to a fixed value $c_{\max}$ (often $c_{\max} = cJ_{\max}$ is chosen, with $c = 4, 5, 6$). For continuous spin systems, Weigel [33] recently suggested to introduce for each vertex only the k lightest edges. He used this cutoff-rule successfully for a matching routine embedded in a genetic algorithm.

The reasoning behind this is that 'heavy'-weighted edges are rarely contained in an optimum solution. This can assumed to be true for, e.g., ± 1 distributed couplings and 50% negative weights, for which very often true ground states are reached in practice.

Using the reduced graph $\tilde{G}_{red}$, Hartmann and Young determined high-quality heuristic ground states for $L \times L$ lattices with ± 1 distribution. They could go up to $L \leq 480$ [30]. Also using the heuristic variant based on the approach of Bieche et al., Palmer and Adler report results for $L \leq 1801$ with the choice of $c_{\max} = 6J_{\max}$ [34].

For different, especially smaller, percentage of negative weights, the quality of the heuristic decreases. This can be understood as follows. An edge (u, v) in the transformed graph is assigned a weight that equals the sum of the absolute weights of the edges in $G_{K,L}$ crossed by a minimum path connecting u and v. u and v correspond to a pair of frustrated plaquettes. The latter can be assumed to be spread all over the system. In case of small p, the total number of frustrated plaquettes is small. Therefore, the weight of a minimum path connecting a pair of them can become large which can cause a heuristic with a limited value of $c_{\max}$ to fail.

Furthermore, for a different distribution of the couplings e.g., Gaussian couplings, this heuristic variant has to be used carefully, as good values for $c_{\max}$ are not evident. Certainly, removing heavy-weighted edges will result in reduced graphs, but it is not clear beforehand which weights should be considered heavy. Applying different cut-off rules, e.g. vertex-degree constraints, might be helpful as they were already used for thinning graphs, but suitable cut-off values depend still on the underlying distribution of the couplings.

An experimental evaluation will be given in Section V.

In the next section, we describe how the correctness of heuristic ground states can be verified using linear programming.

8. Checking Whether a Spin Configuration Actually Defines a Ground State

Suppose we have a spin configuration at hand that has been computed by determining an optimum matching on the graph only containing 'light' edges and we want to test whether it is a correct ground state.

First, we compute an optimum matching on the same reduced graph. In so-called pricing steps, we determine whether yet neglected edges exist that need to be taken into account in order to ensure correctness. In case such an edge is reported by pricing, it is introduced in the reduced graph. The process is iterated until no edge is returned any more. Pricing is a general feature in linear programming and combinatorial optimization. For more details, we refer to [31, 32, 35].

Pricing steps can be performed with Cook and Rohe's state-of-the-art Blossom IV code [36] by implementing small modifications. We give some computational results in Section V.

V. COMPUTATIONAL RESULTS

The method proposed in Section IV 6 can be used for any distribution of the couplings. Here, we focus on ± 1 distributed instances. The concentration p of anti-ferromagnetic bonds was set to 0.5. For the computations we used Intel Xeon CPU with 2.3GHz and AMD Opteron Processor 248 with 2.2GHz, each with less than 16GB RAM. The largest instances $L > 1500$ were done on Xeon processors with 16GB RAM. The physics analysis done with our method will be presented in a forthcoming article [11].

Running times for different sizes of grid graphs ($K = L = 100$, 150, 250, 500, 700, 1000, 2000, and 3000) together with the number of computed samples, are shown in Table I. For smaller grids with $L \leq 500$, we computed between $2{*}10^5$ and $5{*}10^5$ instances per size. For medium sized lattices $700 \leq L \leq 1500$, we ran several thousand samples for each size, whereas for the largest systems of 3000^2 spins we generated results for 157 samples. Lattices with $L \leq 500$ can be computed within less than 2 minutes on average, whereas one ground-state determination for the biggest size requires on average 24h CPU time on a single processor. Results of samples for spin glasses with periodic boundary conditions in one direction are presented in Table II. The running times given in Tables I and II scale with $L^{3.50(5)}$ for $L \times L$ spin glasses. In total we invested about 600 CPU days for our experiments.

It turns out that free boundary samples are computationally a bit more demanding than samples with periodic boundaries in one direction because the intermediate graphs G_{inter} are larger. For other concentrations of anti-ferromagnetic bonds the presented data (running times, memory usage, etc.) are comparable, as our method, unlike the method of Bieche et al., does not depend on the concentration of anti-ferromagnetic bonds but only on the grid-graph size KL.

L_{grid}	average time		min-time	max-time	nr. samples
100	0.67	(±0.00)	0.12	3.47	50925
150	2.03	(±0.01)	0.31	16.80	29750
250	9.70	(±0.03)	1.04	121.18	36800
500	109.62	(±0.79)	5.79	1406.24	20867
700	323.19	(±4.54)	18.96	5233.04	5499
1000	1200.33	(±17.60)	59.49	9717.01	3483
1500	5280.29	(±111.79)	288.58	58036.33	2330
2000	14524.34	(±503.69)	701.16	117313.32	942
3000	61166.70	(±4920.19)	3017.97	316581.15	157

TABLE I: Running times and number of computed samples for different spin glass instance sizes with free boundary conditions and ± 1 distributed couplings ($p = 0.5$).

L_{ring}	average time		min-time	max-time	nr. samples
400	36.00	(±1.19)	3.27	712.67	1400
500	88.31	(±2.91)	5.65	1239.81	1900
700	319.38	(±3.53)	17.46	12712.34	15847
1000	1129.03	(±27.20)	49.87	18577.58	3000

TABLE II: Running times and number of computed samples for different spin glass instance sizes with periodic boundary condition in one direction and ± 1 distributed couplings ($p = 0.5$).

Table III reflects the average over the maximal memory usage needed in the ground state calculations. The memory usage roughly scales with $L^{1.9}$ for Ising spin glasses with free boundary conditions and with $L^{1.6}$ for Ising spin glasses with periodic boundaries in one direction. Both from the CPU time and from the necessary memory we conclude that the method is very fast. It needs considerably less memory than the commonly used method of Bieche et al., which in its heuristic variant allows to treat only smaller system sizes than reported here. However, we also note that a good statistics for system sizes beyond 3000^2 would currently be hard to reach.

Heuristic Ground States and Their Correction

In this section, we explore the quality of the heuristic ground-state calculation using the method of Bieche et al. for two-dimensional $\pm J$ Ising spin glasses with free boundary conditions. Within this we use the verification technique explained in Section IV 8.

We consider planar $L \times L$ lattices with $L = 164$ and ± 1 distributed couplings. First we study these lattices with a concentration $p = 0.5$ of anti-ferromagnetic bonds. It turns out that out of 9912 computed samples 9 (0.091%) were wrong when using $c_{\max} = 4$. Thomas and Middleton [12] stated 1.5% inexact solutions on toric sam-

L_{grid}	∅ memory	L_{ring}	∅ memory
100	158.7 MB		
150	158.8 MB		
250	163.4 MB		
		400	245.6 MB
500	332.5 MB	500	321.6 MB
700	572.7 MB	700	544.6 MB
1000	994.6 MB	1000	993.7 MB
1500	2.052 GB		
2000	3.568 GB		
3000	7.832 GB		

TABLE III: Memory usage for different (± 1, $p = 0.5$) sample sizes.

ples with $L \leq 128$ and $c = 8J_{\max}$. We conclude that in practice the heuristic almost always returns true ground states if $c_{\max}$ and p are suitably chosen.

The overall average running time was 45.26 sec., comparing an average of 82.97 sec. when pricing was necessary with 45.23 sec. without pricing. In our tests, one pricing step was always sufficient to correct a wrong ground state. Using the Kasteleyn approach, a ground state computation takes on average only around 1 second for this lattice size (on Xeon processors), as can be seen in Table V.

In order to assess the influence of the cut-off parameter $c_{\max}$ on the number of wrong results and the time to correct them, we varied p and $c_{\max}$ for the ± 1 164×164 lattices (using the AMD Opteron processors). In Table IV we show results, always averaged over 100 instances.

Several conclusions can be drawn from this experiment. Firstly, small values of $c_{\max}$ lead to many wrong results. E.g., for $c_{\max} = 3$ up to 100% of the results were wrong, and the solutions have to be handled with care. Then, the quality of the heuristic increases with increasing $c_{\max}$. For large enough value, the solutions are very often correct and only need a verification step in order to prove their correctness.

Apart from this trend, the results suggest that the quality of the results highly depends on the chosen combination of $c_{\max}$ and p. Clearly, for small percentage p one has to choose a higher cut-off value $c_{\max}$ in order to generate high-quality solutions. E.g. for $c_{\max} = 3$, all 100 results were wrong when the percentage of negative edges was chosen as $p = 0.1$, whereas 39 results were wrong for $p = 0.5$. This can also be seen by looking at different $c_{\max}$ values but fixed p values. For $c_{\max} \leq 6$ and $p = 0.1$ always at least one percent is wrong. This means that especially for smaller values of p, one has to make sure that the cut-off value is chosen big enough. The reason for this behavior is that the weight of a minimum path connecting a pair of frustrated plaquettes can become large for small p. Therefore, the minimum-weighted perfect matching might contain heavy edges. $c_{\max}$ has to be chosen large enough in order to ensure that these heavy edges are contained in the reduced graph. As argued before, the necessity of having to choose high cut-off values can lead to memory problems as the edge density of the generated reduced graphs increases. These difficulties can be avoided by using the approach based on Kasteleyn cities.

% edges w. $w(e) = -1$	average time [sec]		pricing time [sec]	avg. nr. of pricing steps	nr. of wrong results
$c_{\max} = 3$					
10	16.91	($\pm$0.70)	9.10	2.03	100
20	51.59	($\pm$1.13)	18.30	1.11	93
30	56.87	($\pm$1.57)	22.42	1.00	49
40	56.95	($\pm$1.59)	24.01	1.00	36
50	58.72	($\pm$1.69)	25.10	1.00	39
$c_{\max} = 4$					
10	15.60	($\pm$0.35)	4.76	1.09	77
20	37.37	($\pm$0.74)	17.99	1.00	6
30	49.93	($\pm$0.67)	0.00	0.00	0
40	53.74	($\pm$0.72)	0.00	0.00	0
50	54.02	($\pm$0.76)	0.00	0.00	0
$c_{\max} = 5$					
10	13.04	($\pm$0.30)	4.64	1.00	18
20	37.96	($\pm$0.55)	0.00	0.00	0
30	52.41	($\pm$0.93)	0.00	0.00	0
40	56.79	($\pm$1.00)	0.00	0.00	0
50	59.30	($\pm$1.01)	0.00	0.00	0
$c_{\max} = 6$					
10	13.11	($\pm$0.21)	5.04	1.00	1
20	42.52	($\pm$0.71)	0.00	0.00	0
30	59.05	($\pm$1.30)	0.00	0.00	0
40	65.14	($\pm$1.44)	0.00	0.00	0
50	62.26	($\pm$1.16)	0.00	0.00	0
$c_{\max} = 7$					
10	14.06	($\pm$0.23)	0.00	0.00	0
20	46.70	($\pm$0.89)	0.00	0.00	0
30	66.18	($\pm$1.65)	0.00	0.00	0
40	73.18	($\pm$1.95)	0.00	0.00	0
50	69.69	($\pm$1.52)	0.00	0.00	0
$c_{\max} = 8$					
10	15.01	($\pm$0.29)	0.00	0.00	0
20	53.10	($\pm$1.32)	0.00	0.00	0
30	74.54	($\pm$2.15)	0.00	0.00	0
40	83.20	($\pm$2.32)	0.00	0.00	0
50	82.06	($\pm$2.13)	0.00	0.00	0

TABLE IV: Different $c_{\max}$ values used within the heuristic version of the Bieche et al. approach for each 100 ± 1 164×164 grid graphs with various concentration of anti-ferromagnetic bonds.

From Table IV we see that pricing takes only negligible running time. As a conclusion, if the Bieche et al. algorithm is used to determine ground-state properties, it is advantageous to do the pricing steps, too, independent of the size of the reduced graph $\tilde{G}_{red}$. However, despite the fact that pricing is very fast, the heuristic together with the verification is still considerably slower than the method proposed here based on Kasteleyn cities, cf. Table V.

It is also interesting to assess the quality of the heuristic for different distribution of the couplings, e.g. for Gaussian distributed couplings. We study grid graphs $G_{L,L}$ with $L = 50, 100, 150$. The grid graph size L was limited due to the fact that the generation of the graphs G_F takes a long time. This is explainable as one is in the need of all-pair shortest path calculations for the set of frustrated vertices on the dual graph. The latter is a polynomially solvable problem, however the computations can take long. In our tests, computing the short-

$\%\ (w(e)<0)$ / $\lvert V\rvert$	10	20	30	40	50
164^2	0.92 ($\pm$0.005)	1.28 ($\pm$0.005)	1.32 ($\pm$0.007)	1.17 ($\pm$0.007)	1.11 ($\pm$0.009)

TABLE V: Average running times (in sec.) over 10.000 instances, using the method based on Kasteleyn cities on ± 1 164×164 grids with various concentration of anti-ferromagnetic bonds.

est paths usually takes much longer than computing the ground states itself. In Table VI the running times for the matching on the different reduced graphs can be seen.

As it is not clear beforehand how big the cut-off parameter $c_{\max}$ should be, we let ourselves be guided by the corresponding values used in instances with bimodal distribution. More specifically, for some value of $c_{\max}$ (we used $c_{\max} = 5$) we compute the average percentage of 'light' edges from G_F that are contained in the reduced graph $\tilde{G}_{red}$ in instances with bimodal distribution. For $L = 50$, we find that in ± 1 distributed instances with $p = 0.5$ on average the 8% lightest edges are used (this percentage reduces to an average of 2% for $L = 100$ and to 1% for $L = 150$).

For Gaussian distributed instances, we build the reduced graphs with the same percentages of light edges, i.e., 8% (2% or 1%) light edges for $L = 50$ ($L = 100$, $L = 150$ resp.) Results are shown in Table VI. First of all, it turns out that for small grid graphs many results are wrong. For $L = 50$, about 12.5% of the calculated instances do not find correct ground states, and a higher cut-off value has to be used. Considering larger grid graphs the situation looks similar. 47% (52%) instances computed the wrong ground state for $L = 100$ ($L = 150$).

For $L = 50$, we have to go up to a percentage of 16% lightest edges (corresponding to $c_{\max} \approx 8$), for $L = 100$ to 4% (corresponding to $c_{\max} \approx 7$) and for $L = 150$ we have to take into account the 2.5% lightest edges, corresponding to $c_{\max} \approx 8$, in order to ensure correctness of the results for our test data. Again, when comparing the running times for 150^2 lattices, it is by roughly a factor of 25 faster to use the Kasteleyn city approach instead of a heuristic Bieche method.

As the performance of the matching routine scales with the graph sizes, more specifically with the number of vertices and edges, we study the reduced graphs with respect to these two entities. Usually, the graphs for computing the matchings are dense. This is especially true for small grid graphs ($L = 50$), where 16% of the light edges are needed. Increasing the grid graphs leads to a considerable decrease of needed light edges, and decreasing density. (The density of a graph with n vertices is defined as the number of its edges divided by the number of edges of the complete graph K_n.) Nevertheless, in our experiments the grid graphs with $L = 100$ ($L = 150$) contain on average $|V| = 4901 \pm 5$ ($|V| = 11103 \pm 8$) vertices. Taking 4% (2.5%) of all edges means in absolute numbers $4.803(9) * 10^5$ ($1.541(2) * 10^6$, resp.) edges, which are large and dense graphs. This has to be compared with the graphs used within the Kasteleyn approach. Here we have for $L = 100$ ($L = 150$) graphs with $|V| = 40804$ ($|V| = 91204$) and $8.120 * 10^4$ ($1.818 * 10^5$) edges. In fact, the graphs we deal with have more vertices but are considerably sparser. The Blossom IV routine can compute matchings in very large graphs, provided their density is low. This fact is also reflected in the better running times for the Kasteleyn approach presented in this section. Comparing with the situation for instances with bimodal distributed edge weights, we see from Table IV that a value $c_{\max} = 4$ in the case of $p = 0.5$ yields good results. For $L = 100$ ($L = 150$) we have $|V| = 4901 \pm 4$ ($|V| = 11114 \pm 7$) with $|E| = 1.58(7) * 10^5$ ($|E| = 3.66(1) * 10^5$), which are about 1.3% (0.6%) edges of the complete graph. Thus, the graphs with bimodal distribution can be chosen sparser than in the Gaussian case (for $c_{\max} = 4$). However, these graphs are usually denser than the graphs used in the Kasteleyn approach.

VI. CONCLUSIONS

We presented a simple algorithm (Section IV 6) based on Kasteleyn cities. The algorithmic foundations of this method date back to the work of Kasteleyn [10] from the 1960s in which he computed the complete partition function for the Ising model. Using this approach, we can compute exact ground states for two-dimensional planar Ising spin-glass instances. The method is easy to implement, fast and has only limited memory requirements. According to our knowledge, the treatable system sizes are considerably bigger than the ones computed earlier and are always provably exact. Thomas and Middleton [12] used Kasteleyn cities for studying extended ground states. Furthermore, they state that the method can also be used for determining exact ground states of planar graphs.

We evaluated different established exact methods and compared them with respect to running time and memory requirements. It turned out that the approach presented here is both considerably faster and needs less memory than the methods proposed earlier. We showed how heuristically computed ground states can be verified or corrected fast using mathematical optimization. However, the method based on Kasteleyn cities still outperforms this approach. Finally, we evaluated the solution quality of heuristic variants of the Bieche et al. approach.

In the future, we will make our program available in public domain via the Cologne Spin Glass Server that can be found at

L	$c_{\max}$ (± 1 case)	% lightest edges (Gassian case)	% wrong results	matching time [sec.]
50	5	8.0	12.5	0.11 ($\pm$ 0.01)
	6	12.0	2.0	0.18 ($\pm$ 0.01)
	7	15.0	2.0	0.23 ($\pm$ 0.01)
	8	16.0	0.0	0.24 ($\pm$ 0.01)
100	5	2.0	47.0	2.68 ($\pm$ 0.20)
	6	3.0	2.0	7.13 ($\pm$ 0.50)
	7	4.0	0.0	11.27 ($\pm$ 0.80)
	8	5.0	0.0	23.20 ($\pm$ 1.60)
150	5	1.0	52.0	34.60 ($\pm$ 2.21)
	6	1.4	7.0	49.34 ($\pm$ 2.97)
	7	1.9	0.0	66.70 ($\pm$ 4.00)
	8	2.5	0.0	85.18 ($\pm$ 5.14)

TABLE VI: Size of reduced graph $\tilde{G}_{red}$ when using Gaussian distributed couplings compared to the achieved quality by the heuristic variant of the Bieche et al. approach.

http://cophy.informatik.uni-koeln.de/research.html#a248

Acknowledgments

We thank Michael Jünger for fruitful discussions and Vera Schmitz for adapting Blossom IV's pricing method as described above. Thanks to Frank Baumann and Olivier C. Martin for commenting on an earlier version of this article and to Alan Middleton for helpful communications. We are indebted to two anonymous referees for their valuable comments. Last but not least, we thank Oliver Melchert for stimulating discussions and for providing us with some ground-state data. Financial support from the German Science Foundation is acknowledged under contract Li 1675/1-1. Partially supported by the Marie Curie RTN Adonet 504438 funded by the EU.

[1] Hartmann, A., 2008, in *Rugged Free Energy Landscapes*, edited by W. Janke, volume 736 of *Lecture Notes in Physics, Berlin Springer Verlag*, 67–106.

[2] Hartmann, A. K., and H. Rieger, 2002, *Optimization Algorithms in Physics* (Wiley-VCH), ISBN 3-527-40307-8.

[3] Rieger, H., 1998, in *Lecture Notes in Physics, Berlin Springer Verlag*, edited by J. Kertész and I. Kondor, volume 501 of *Lecture Notes in Physics, Berlin Springer Verlag*.

[4] Edwards, S. F., and P. W. Anderson, 1975, "Theory of spin glasses," Journal of Physics F: Metal Physics **5**(5), 965–974.

[5] Papadimitriou, C. H., and K. Steiglitz, 1982, *Combinatorial optimization: algorithms and complexity* (Prentice-Hall, Inc., Upper Saddle River, NJ, USA), ISBN 0-13-152462-3.

[6] Liers, F., M. Jünger, G. Reinelt, and G. Rinaldi, 2004, "Computing Exact Ground States of Hard Ising Spin Glass Problems by Branch-and-Cut," in *New Optimization Algorithms in Physics*, edited by A. K. Hartmann and H. Rieger (Wiley-VCH), 47–68.

[7] Bieche, L., J. P. Uhry, R. Maynard, and R. Rammal, 1980, "On the ground states of the frustration model of a spin glass by a matching method of graph theory,".

[8] Barahona, F., 1982, "On the Computational Complexity of Ising Spin Glass Models," J. Physics A: Mathematical and General 15 **10**, 3241–3253.

[9] Barahona, F., R. Maynard, R. Rammal, and J. P. Uhry, 1982, "Morphology of ground states of two-dimensional frustration model ," Journal of Physics A Mathematical General **15**, 673–699.

[10] Kasteleyn, P. W., 1963, "Dimer Statistics and Phase Transitions," Journal of Mathematical Physics **4**(2), 287–293.

[11] Pardella, G., F. Liers, and F. Krzakala, 2008, manuscript in preparation.

[12] Thomas, C. K., and A. A. Middleton, 2007, "Matching Kasteleyn cities for spin glass ground states," Physical Review B (Condensed Matter and Materials Physics) **76**(22), 220406.

[13] Bollobás, B., 1998, *Modern graph theory*, volume 184 of *Graduate Texts in Mathematics* (Springer-Verlag, New York), ISBN 0-387-98491-7.

[14] Diestel, R., 2006, *Graph Theory*, volume 173 of *Graduate Texts in Mathematics* (Springer-Verlag, Heidelberg), ISBN 3-540-26182-6.

[15] Harary, F., 1969, *Graph Theory* (Addison-Wesley).

[16] Harary, F., 1989, in *Proceedings of the third international conference on Combinatorial mathematics* (New York Academy of Sciences, New York, NY, USA), 216–219.

[17] Edmonds, J., 1965, "Maximum matching and a polyhedron with 0-1 vertices.," Journal of Research at the National Bureau of Standards **69B**, 125–130.

[18] Edmonds, J., 1965, "Paths, trees, and flowers.," Can. J. Math. **17**, 449–467.

[19] Lovász, L., and M. D. Plummer, 1986, *Matching Theory*, volume 29 of *Annals of Discrete Mathematics* (North-

Holland, Amsterdam).
[20] Kasteleyn, P. W., 1961, "The statistics of dimers on a lattice : I. The number of dimer arrangements on a quadratic lattice," Physica **27**, 1209–1225.
[21] Fisher, M. E., 1966, "On the Dimer Solution of Planar Ising Models," Journal of Mathematical Physics **7**(10), 1776–1781.
[22] Domb, C., 1960, "On the theory of cooperative phenomena in crystals," Advances in Physics **9**, 245–361.
[23] Galluccio, A., and M. Loebl, 1999, "On the Theory of Pfaffian Orientations. I. Perfect Matchings and Permanents," Electr. J. Comb. **6**.
[24] Galluccio, A., and M. Loebl, 1999, "On the Theory of Pfaffian Orientations. II. T-joins, k-cuts, and Duality of Enumeration," Electr. J. Comb. **6**.
[25] Galluccio, A., M. Loebl, and J. Vondrák, 2000, "New Algorithm for the Ising Problem: Partition Function for Finite Lattice Graphs," Phys. Rev. Lett. **84**(26), 5924–5927.
[26] Kawashima, N., and H. Rieger, 1997, "Finite-size scaling analysis of exact ground states for ±J spin glass models in two dimensions," EPL (Europhysics Letters) **39**(1), 85–90.
[27] McMillan, W. L., 1984, "Domain-wall renormalization-group study of the three-dimensional random Ising model," Phys. Rev. B **30**(1), 476–477.
[28] Rieger, H., L. Santen, U. Blasum, M. Diehl, M. Jünger, and G. Rinaldi, 1996, "The critical exponents of the two-dimensional Ising spin glass revisited: Exact Ground State Calculations and Monte Carlo Simulations," Journal of Physics A: Mathematical and General **29**, 3939–3950.
[29] Fisch, R., and A. K. Hartmann, 2007, "Ground-State and Domain-Wall Energies in the Spin-Glass Region of the 2D $\pm J$ Random-Bond Ising Model," Physical Review B **75**, 174415.
[30] Hartmann, A. K., and A. P. Young, 2001, "Lower critical dimension of Ising spin glasses," Phys. Rev. B **64**(18), 180404.
[31] Chvátal, V., 1983, *Linear programming*, A Series of Books in the Mathematical Sciences (W. H. Freeman and Company, New York), ISBN 0-7167-1195-8; 0-7167-1587-2.
[32] Nemhauser, G. L., and L. A. Wolsey, 1999, *Integer and Combinatorial Optimization*, In Discrete Mathematics And Optimization (Wiley-Interscience, New York, NY, USA).
[33] Weigel, Martin, 2007, "Genetic embedded matching approach to ground states in continuous-spin sysytems," Phys. Rev. E **76**(6), 066706.
[34] Palmer, R. G., and J. Adler, 1999, "Ground States for Large Samples of Two-Dimensional Ising Spin Glasses," International Journal of Modern Physics C **10**, 667–675.
[35] Cook, W. J., W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver, 1998, *Combinatorial Optimization* (John Wiley & Sons, Inc., New York, NY, USA), ISBN 0-471-55894-X.
[36] Cook, W., and A. Rohe, 1999, "Computing minimum-weight perfect matchings," INFORMS Journal on Computing **11**, 138–148.