

On the approximability of the energy function of Ising spin glasses

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

1994 J. Phys. A: Math. Gen. 27 6719

(<http://iopscience.iop.org/0305-4470/27/20/012>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 171.66.16.68

The article was downloaded on 01/06/2010 at 22:08

Please note that [terms and conditions apply](#).

On the approximability of the energy function of Ising spin glasses*

A Bertoni, P Campadelli and G Molteni

Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano, via Comelico 39/41, I-20135 Milan, Italy

Received 21 February 1994, in final form 24 June 1994

Abstract. We consider polynomial-time algorithms for finding approximate solutions to the ground-state problem for the following three-dimensional case of an Ising spin glass: n spins are arranged on a two-level grid with \sqrt{n} vertical interactions. The main results are:

(i) there is an approximate polynomial-time algorithm with absolute error less than \sqrt{n} , for all n ; and

(ii) there exists a constant $\alpha > 0$ such that every approximate polynomial-time algorithm has absolute error greater than $\alpha \sqrt{n}$ infinitely often, unless $P = NP$.

1. Introduction

Spin glasses represent one of the most challenging problems in solid state and statistical physics. The prototype of a spin glass is a dilute magnetic alloy, such as 1% of Mn or Fe embedded in Cu or Au.

Many models have been proposed to describe the behaviour of these disordered systems, but many important questions still have no satisfactory answers. In particular, the determination of ground-state energy values is a difficult task from a theoretical point of view and experimental investigations give no precise results.

The very long relaxation times are the main difficulty encountered both in experimental settings and in Monte Carlo simulations. The failure of the relaxation methods suggests the use of different algorithms to solve the ground-state problem.

The first attempt to provide exact ground states by a non-relaxation algorithm was performed by Bieche *et al* (1980) for an Ising spin glass on a planar lattice described by the frustration model, where the interactions can have only two symmetrical values, $\pm J$. Edmonds' (1965) algorithm for the minimum perfect matching problem was used to show that, for that particular model of a spin glass, the exact ground states could be generated in a polynomial amount of computing time.

Barahona (1982) proved that, under the widely believed conjecture that $P \neq NP$, for a three-dimensional Ising spin glass with nearest-neighbour interactions chosen randomly from $\{-1, 0, +1\}$, there is no polynomial-time algorithm with which to compute the energy of the ground state and the partition function. This result of NP -completeness makes it necessary to sacrifice optimality and look for approximation algorithms which run in polynomial time.

* This work has been supported by 'Progetto strategico algoritmi e modelli neurali' under CNR grant no 93.05230.ST74 and under MURST 40 'Algoritmi, modelli di calcolo e strutture informative'.

In this paper, we restrict our attention to an Ising spin glass on a two-level grid such that the number of vertical connections is \sqrt{n} , where n is the number of spins, since Barahona's NP -completeness result is related to this model.

Our main result is a lower bound on the absolute error for any approximate polynomial-time algorithm for the ground-state problem restricted to this three-dimensional model.

In fact, we prove:

(i) that there is an approximate polynomial-time algorithm with absolute error less than \sqrt{n} for all n ; and

(ii) that there exists a constant $\alpha > 0$ such that every approximate polynomial-time algorithm has an absolute error greater than $\alpha\sqrt{n}$ infinitely often, unless $P = NP$.

In section 2 we recall some basic elements of complexity theory and in section 3 we introduce some optimization problems and useful facts in the area of combinatorial optimization. In section 4 our main results are stated, section 5 is devoted to the proofs, while section 6 summarizes our conclusions and deals with the unresolved problems.

2. Some basic elements of complexity theory

In this section, we summarize some of the basic notions of complexity theory which we refer to in the present paper.

Intuitively, a *decision problem* Π can be described as a set of instances D_Π and a certain property Φ ; the problem consists in deciding which instances in D_Π satisfy Φ .

Denoting the set of words over a finite alphabet Σ by Σ^* and the length of the word w by $|w|$, the elements of D_Π are codified as words in Σ^* so that the *size* of an instance is given by the length of the encoding word. In this way, the problem can be represented by a language $L \subseteq \Sigma^*$, where L is the set of words codifying the instances that verify the property Φ .

Given this formulation, the solution of a decision problem is equivalent to recognizing a suitable language. Devices that recognize languages, e.g. Turing machines, use computational resources like space and time; the aim of complexity theory is to classify problems with respect to the resources needed to solve them. Two important complexity classes are P and NP .

P is the class of languages which are recognizable in polynomial time by Turing machines, i.e. in a number of computation steps bounded by a function polynomial in the size of the instance. There is general agreement that a problem can be considered 'effectively' computable if it is in class P .

NP is the class of languages L that can be specified as follows (Stockmeyer 1987):

$$w \in L \text{ iff } \exists y \quad \text{such that} \quad |y| \leq p(|w|) \text{ and } R(w, y)$$

for some polynomial p and a binary relation R which is computable in polynomial time by a Turing machine. That is, NP is the class of languages L such that $w \in L$ if and only if there is a short 'proof' y of membership of w in L such that the validity of the proof can be 'easily' verified.

Of course, $P \subseteq NP$. The basic question 'Is $P \neq NP$?' had already been posed by Gödel (1956) in a letter written to Von Neumann (see Hartmanis 1989). Regarding this question, observe that even though a given 'proof' can be verified in polynomial time, there are exponentially many 'potential proofs' to be checked; for this reason, it is intuitively believed that the class P is properly contained in the class NP , although until now there

has been no proof of the validity of this conjecture; indeed, in this paper, we assume $P \neq NP$.

Among the problems in NP , the so-called NP -complete problems deserve particular attention since they are the most difficult from a computational point of view. Roughly speaking, a problem Π is called NP -hard if the existence of a polynomial-time algorithm for Π would imply $P = NP$ and it is called NP -complete if it is NP -hard and belongs to the class NP (for a precise definition, see Garey and Johnson (1979)).

A famous NP -complete problem (and indeed the first which has been proved to be NP -complete (Cook 1971)) is the satisfiability problem SAT, which is the problem of deciding whether a given Boolean formula in conjunctive form admits a satisfying truth assignment. In order to state the problem precisely, let us recall that, given a set of Boolean variables $X = \{x_1, x_2, \dots, x_n\}$, a literal is either a variable x_i or its negation \bar{x}_i , a clause is a disjunction of literals and an assignment is a function that associates 0 or 1 with every variable. A clause is said to be satisfied by an assignment if at least one of its literals has value 1. Formally:

SAT.

Instance. A set $X = \{x_1, x_2, \dots, x_n\}$ of Boolean variables and a collection C of clauses over X .

Question. Is there a satisfying assignment which satisfies all clauses in C ?

The notion of NP -completeness has been used since the early 1970s (Karp 1972) to show the difficulty in finding optimal solutions for a large variety of combinatorial optimization problems. The apparent intractability of these problems motivated the search for approximate solutions (see, for instance, Garey and Johnson (1979)) and the development of an approximability theory for NP -hard optimization problems.

Important results in this field have recently been obtained on the basis of a new characterization of the NP class (Arora *et al* 1992) which we recall for the sake of completeness. The key concept is the notion of interaction between a *verifier* and an *oracle*. An *oracle* is simply a word $y \in \{0, 1\}^*$; a *verifier* is a polynomial-time algorithm M which, during its computation, may 'call' the oracle. The verifier-oracle interaction has the following protocol: if M enters a particular state (*query state*), then it reads an integer k written in its memory at that computation step; at the next step, M obtains the bit y_k in position k in the oracle y and it continues the computation.

An $[r(n), q(n)]$ -*verifier* is a verifier M that, having had a word w input and a random string $r \in \{0, 1\}^*$ of length $O(r(|w|))$, calls an arbitrary oracle y at most $O(q(|w|))$ times. We set $M^y(w, r) = 1$ if M , interacting with y , accepts (w, r) , and $M^y(w, r) = 0$ otherwise.

The class $PCP[r(n), q(n)]$ of languages which are recognized by $[r(n), q(n)]$ -verifiers (Arora and Safra 1992) can now be introduced.

A language $L \subseteq \Sigma^*$ belongs to $PCP[r(n), q(n)]$ if there exists a $[r(n), q(n)]$ -verifier M such that:

- (i) $w \in L$ iff there is an oracle y such that $M^y(w, r) = 1$ for all r ; and
- (ii) $w \notin L$ iff the probability $\{r | M^y(w, r) = 1\} < \frac{1}{2}$ for all y .

Note that, by the previous definitions, the class P coincides with $PCP[0, 0]$ and the class NP with $PCP[0, n^{O(1)}]$. A surprising characterization of NP is given by the following theorem:

Theorem 2.1. (Arora *et al* 1992)

$$NP = PCP[\log n, 1].$$

3. Approximation problems: preliminary definitions and results

For a large class of combinatorial optimization problems, determining an optimal solution is extremely time consuming (Garey and Johnson 1979). For this reason, especially when dealing with large instances of such problems, one has to be satisfied with approximate solutions: trying to classify optimization problems with respect to the complexity of computing approximate solutions is, therefore, extremely relevant.

In this section, some basic definitions and a few recent results concerning the approximability of 'difficult' combinatorial optimization problems are summarized; for notational simplicity, definitions are given only for maximization problems.

Definition 3.1. A maximization problem Π is defined by the triple $\langle \text{In}, \text{Sol}, w \rangle$, where In denotes the set of the instances, Sol is a mapping that, given an instance $I \in \text{In}$, provides the set of feasible solutions, and w is the objective function that associates a non-negative rational number (solution value) with every couple $\langle I, S \rangle$, where $I \in \text{In}$ and $S \in \text{Sol}(I)$.

It is assumed that there is a 'natural' notion of size $|I|$ for every instance I and size $|S|$ for every feasible solution $S \in \text{Sol}(I)$ and it is required that:

- (i) the size of every feasible solution $S \in \text{Sol}(I)$ is polynomially bounded in the size of the instance I , i.e. there is a polynomial p such that $|S| \leq p(|I|)$;
- (ii) the predicate $S \in \text{Sol}(I)$ can be decided in time polynomial in $|I|$ and $|S|$; and
- (iii) the objective function w can be computed in polynomial time.

For any maximization problem, different formulations can be given: a *decision version*; an *evaluation version*; and a *constructive version*. Formally: the *decision version*, given an instance I and a positive rational k , decides whether the couple $\langle I, k \rangle$ belongs to the set $\{\langle I, k \rangle \mid \exists S : w(I, S) \geq k\}$; the *evaluation version* computes the value $w^*(I) = \text{Max}_{S \in \text{Sol}(I)} \{w(I, S)\}$; and the *constructive version* finds an optimal solution S^* that is a feasible solution such that $w(I, S^*) = w^*(I)$.

We now give the constructive version of two maximization problems useful in this context.

MAX 3SAT.

Instance. V , a set of Boolean variables; C , a collection of clauses over V with at most three literals per clause.

Question. Find an assignment on the Boolean variables that satisfies the greatest number of clauses.

MAX CUT-3.

Instance. $G = (V, E)$, a graph of degree $D \leq 3$.

Question. Find a subset $V_1 \subset V$ such that the cardinality of $\text{cut}(V_1)$ is maximum, where $\text{cut}(V_1)$ is the set of edges with one endpoint in V_1 and one endpoint in $V \setminus V_1$.

It is well known (Garey and Johnson 1979) that, under the conjecture $P \neq NP$, if the decision version of a maximization problem is NP -complete, there is no hope of finding 'efficient' (i.e. polynomial-time) algorithms to solve either the constructive or the evaluation version exactly. Thus, one can try and find, in polynomial time, only 'good' approximate solutions; a measure of the quality of an approximate solution is given by the relative error.

Definition 3.2. Given a maximization problem $\Pi = \langle \text{In}, \text{Sol}, w \rangle$, let $S \in \text{Sol}(I)$. We call *relative error* the quantity

$$\text{Err}(I, S) = (w^*(I) - w(I, S))/w^*(I)$$

where $w^*(I) = \text{Max}_{S \in \text{Sol}(I)}\{w(I, S)\}$.

An *approximate algorithm* for a maximization problem $\Pi = \langle \text{In}, \text{Sol}, w \rangle$ is an algorithm A that, having as an input a problem instance I , outputs a solution $A(I) \in \text{Sol}(I)$. We say that A is an approximate algorithm of level $\varepsilon > 0$ (equivalently, an ε -approximate algorithm) if, for every instance I , $\text{Err}(I, A(I)) \leq \varepsilon$. The possibility of finding 'good' approximate solutions for a maximization problem can be formalized by the notion of the *polynomial-time approximation scheme* (PTAS).

Definition 3.3. A maximization problem Π is solved by a PTAS if, for every $\varepsilon > 0$, there exists an algorithm A_ε , with running time bounded by a polynomial in the size of the instance I , that outputs a solution $A_\varepsilon(I)$ such that

$$\text{Err}(I, A_\varepsilon(I)) \leq \varepsilon.$$

Various notions of approximation-preserving reductions have been introduced with the aim of classifying optimization problems from the point of view of their approximability. We recall here the notion of *L-reduction* introduced by Papadimitriou and Yannakakis (1991).

Definition 3.4. Let $\Pi_1 = \langle \text{In}_1, \text{Sol}_1, w_1 \rangle$ and $\Pi_2 = \langle \text{In}_2, \text{Sol}_2, w_2 \rangle$ be two maximization problems. We say that Π_1 *L-reduces* to Π_2 (and write $\Pi_1 \leq_L \Pi_2$) if there are two polynomial-time computable functions f and g and two constants $\alpha, \beta > 0$ such that

(i) with every instance $I \in \text{In}_1$, the function f associates an instance $f(I) \in \text{In}_2$ satisfying $w_2^*(f(I)) \leq \alpha w_1^*(I)$; and

(ii) with every instance $I \in \text{In}_1$ and $S_2 \in \text{Sol}_2(f(I))$, the function g associates a solution $g(I, S_2) \in \text{Sol}_1(I)$ such that $\text{Err}(I, g(I, S_2)) \leq \beta \text{Err}(f(I), S_2)$.

It is not difficult to prove that if $\Pi_1 \leq_L \Pi_2$ and Π_2 admit a PTAS, then Π_1 also admits a PTAS.

Recently, some important results in *NP*-hard optimization-problem approximability theory have been determined by the application to this field of techniques based on interactive protocols (Babai 1985, Goldwasser *et al* 1985). Along these lines, on the basis of the new characterization of the *NP* complexity class given in theorem 2.1, it has been shown that the MAX 3SAT problem does not admit a PTAS unless $P = NP$ (Arora *et al* 1992). From the *L*-reducibility of MAX 3SAT to MAX CUT-3, proved by Papadimitriou and Yannakakis (1991), we can, therefore, conclude:

Theorem 3.1. There does not exist a PTAS for MAX CUT-3, unless $P = NP$.

It can easily be shown that every ε^* -approximate polynomial-time algorithm for MAX CUT-3 restricted to connected graphs can be transformed into an ε^* -approximate polynomial time for arbitrary graphs. We can, therefore, state theorem 3.1 for connected graphs.

Theorem 3.2. If $P \neq NP$, there exists $\varepsilon^* > 0$ such that for every approximate polynomial-time algorithm A for MAX CUT-3 there are infinitely many connected graphs I of degree three for which $\text{Err}(I, A(I)) \geq \varepsilon^*$.

4. Results

Barahona (1982) has shown that, unless $P = NP$, there are no polynomial-time algorithms which find the absolute minima of the energy function of Ising spin glasses on a finite three-dimensional lattice with random nearest-neighbour couplings. In this section, we consider a class of Ising spin glasses studied by Barahona (1982), section 4.2, and estimate upper and lower bounds on the absolute error made by approximate polynomial-time algorithms.

Consider an Ising spin glass on a two-level squared grid (\hat{V}, \hat{E}) such that, if n is the total number of nodes, the total number of vertical edges is at most \sqrt{n} (see figure 1). With each node $i \in \hat{V}$, there is associated a single variable σ_i with values in $\{-1, 1\}$ indicating spin orientations, and with each edge $\{i, j\} \in \hat{E}$ there is associated a weight J_{ij} , randomly chosen in the set $\{-1, 0, 1\}$, indicating the interactions between nearest-neighbour spins: in this way a weighted grid $\hat{G} = (\hat{V}, \hat{E}, J)$, where $J : \hat{E} \rightarrow \{-1, 0, 1\}$, is obtained. The energy of a spin configuration $\sigma = [\sigma_1, \dots, \sigma_n]$ is given by the Hamiltonian $H = -\sum_{\{i,j\} \in \hat{E}} J_{ij} \sigma_i \sigma_j$ and the ground states are those configurations which minimize H .

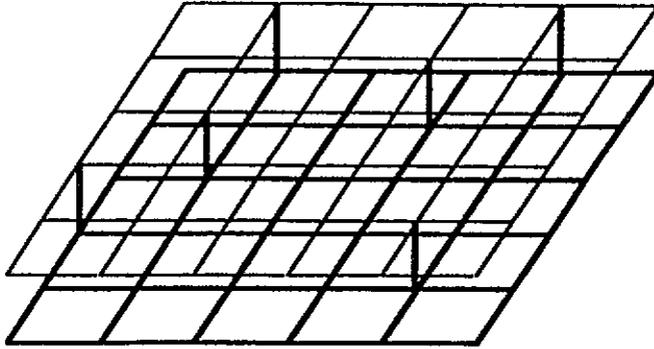


Figure 1. A two-level grid with n spins and, at most, \sqrt{n} vertical interactions.

Let \mathcal{G} be the class of the weighted grids $\hat{G} = (\hat{V}, \hat{E}, J)$ just described, i.e. (\hat{V}, \hat{E}) is a two-level squared grid such that, if n is the total number of nodes, the total number of vertical edges is at most \sqrt{n} and J is a function $J : \hat{E} \rightarrow \{-1, 0, 1\}$. The ground-state (GS) problem we are going to consider is formally defined as follows.

GS.

Instance. A weighted grid $\hat{G} = (\hat{V}, \hat{E}, J) \in \mathcal{G}$.

Question. Determine a spin configuration that minimizes the function $H : \{-1, 1\}^n \rightarrow \mathbb{Z}$ defined as $H = -\sum_{\{i,j\} \in \hat{E}} J_{ij} \sigma_i \sigma_j$.

Barahona's result quoted above can now be stated precisely: the decision version of GS is NP -complete.

Our main result is an $\Omega(\sqrt{|\hat{V}|})$ lower bound on the absolute error made by any approximate polynomial-time algorithm; besides, we show, for the class \mathcal{G} , an approximate polynomial-time algorithm optimal up to a multiplicative constant.

Let $H^*(\hat{G})$ denote the minimum energy value of a spin glass on the weighted grid \hat{G} , i.e. $H^*(\hat{G}) = \min_{\sigma} H(\sigma)$; given an approximate polynomial-time algorithm A for the GS problem, we denote the spin configuration, given by the algorithm A on input \hat{G} , by $A(\hat{G})$ and the corresponding energy value by $H(A(\hat{G}))$.

We can now state our main results.

Fact 4.1. There exists an approximate polynomial-time algorithm A that, for all weighted grids $\hat{G} = \langle \hat{V}, \hat{E}, J \rangle$ of the class \mathcal{G} , finds a solution $A(\hat{G})$ which satisfies the following relation:

$$|H^*(\hat{G}) - H(A(\hat{G}))| \leq \sqrt{|\hat{V}|}.$$

Fact 4.2. If $P \neq NP$, there exists a constant $\alpha > 0$ such that, for every approximate polynomial-time algorithm A , there are infinitely many grids $\hat{G} = \langle \hat{V}, \hat{E}, J \rangle$ of the class \mathcal{G} for which the following relation holds:

$$|H^*(\hat{G}) - H(A(\hat{G}))| \geq \sqrt{|\hat{V}|}.$$

The results about the quality of the solutions given by approximate polynomial-time algorithms for the GS problem are given in terms of absolute error. As far as the relative error is concerned, we note the following considerations: given a two-level squared grid $\langle \hat{V}, \hat{E} \rangle$ of the type described above, let $\hat{G} = \langle \hat{V}, \hat{E}, J \rangle$ be a *weighted random grid* obtained by choosing the weights of the edges randomly and independently according to the uniform probability distribution on the set $\{-1, 0, 1\}$. By a standard argument, it can be shown that $\text{Prob}\{H^*(\hat{G}) \leq -|\hat{V}|/8\} \geq 1 - e^{-c|\hat{V}|}$, where c is a positive constant. This observation, together with fact 4.2, implies the existence of an approximate polynomial-time algorithm that, for ‘asymptotically almost all’ two-level *weighted random grids*, finds solutions with relative error $O(1/\sqrt{|\hat{V}|})$.

5. Proofs

Proof of fact 4.1. Given a two-level weighted grid $\hat{G} \in \mathcal{G}$, it is easy to verify that the associated energy function H can be expressed as

$$H(\underline{x}, \underline{y}) = H_1(\underline{x}) + H_2(\underline{y}) + H_3(\underline{x}, \underline{y})$$

where $\underline{x}, \underline{y}$ are the spin configurations of the upper and lower levels, respectively, $H_1(\underline{x})$ is the energy contribution due to edges between nodes of the upper level, $H_2(\underline{y})$ is the energy contribution due to edges between nodes of the lower level and $H_3(\underline{x}, \underline{y})$ is the energy contribution due to vertical edges.

Consider the following algorithm A for the approximate evaluation of the energy absolute minima.

Algorithm A.

- (i) input $\hat{G} = \langle \hat{V}, \hat{E}, J \rangle$
- (ii) compute $\underline{x}^* = \text{argmin } H_1(\underline{x})$
- (iii) compute $\underline{y}^* = \text{argmin } H_2(\underline{y})$
- (iv) if $H_3(\underline{x}^*, \underline{y}^*) < 0$ then $\underline{z} := (\underline{x}^*, \underline{y}^*)$ else $\underline{z} := (\underline{x}^*, -\underline{y}^*)$
- (v) output \underline{z} .

Since, as shown by Bieche *et al* (1980), the GS problem for planar graphs can be solved in polynomial time by Edmonds’ (1965) algorithm, problems at points (ii) and (iii) can be solved in polynomial time and, therefore, algorithm A works in polynomial time. Moreover, observe that:

(i) $H(\underline{z}) = H_1(\underline{x}^*) + H_2(\underline{y}^*) + \min\{H_3(\underline{x}^*, \underline{y}^*), H_3(\underline{x}^*, -\underline{y}^*)\} \leq H_1(\underline{x}^*) + H_2(\underline{y}^*)$, since $H_3(\underline{x}^*, -\underline{y}^*) = -H_3(\underline{x}^*, \underline{y}^*)$; and

(ii) recalling that $H^*(\hat{G}) = \min_{\underline{x}, \underline{y}} H(\underline{x}, \underline{y})$, we obtain

$$H^*(\hat{G}) = H_1(\underline{x}^*) + H_2(\underline{y}^*) + \min_{\underline{x}, \underline{y}} H_3(\underline{x}, \underline{y}) \geq H_1(\underline{x}^*) + H_2(\underline{y}^*) - \sqrt{|\hat{V}|}$$

since $H_3(\underline{x}, \underline{y}) \geq -\sqrt{|\hat{V}|}$ for all $\underline{x}, \underline{y}$.

From inequalities at points (i) and (ii), it follows that

$$|H^*(\hat{G}) - H(A(\hat{G}))| \leq \sqrt{|\hat{V}|}. \quad \square$$

We need some observations before proving fact 4.2.

Observation 5.1. The GS problem for Ising spin glasses on two-level weighted grids in the class \mathcal{G} is strongly connected to the following problem.

MAX $\{-1, 0, 1\}$ -CUT.

Instance. A weighted grid $\hat{G} = \langle \hat{V}, \hat{E}, J \rangle \in \mathcal{G}$.

Question. Find a subset $\hat{V}_1 \subset \hat{V}$ such that the weight $W(\text{cut}(\hat{V}_1))$ is maximum where $W(\text{cut}(\hat{V}_1)) = \sum_{\{i,j\} \in \text{cut}(\hat{V}_1)} J_{ij}$.

In fact, let H be the energy function of an Ising spin glass on the weighted grid $\hat{G} = \langle \hat{V}, \hat{E}, J \rangle$ and let W be the function that assigns the weight to a cut in the weighted grid $\langle \hat{V}, \hat{E}, -J \rangle$. It is easy to verify that

$$H = - \sum_{\{i,j\} \in \hat{E}} J_{ij} - 2W$$

or, equivalently,

$$W = \frac{1}{2} \left(- \sum_{\{i,j\} \in \hat{E}} J_{ij} - H \right).$$

This relation shows that any approximate polynomial-time algorithm for GS with absolute error a is an approximate polynomial-time algorithm for MAX $\{-1, 0, 1\}$ -CUT with absolute error $a/2$ and *vice versa*.

Observation 5.2. We now consider the problem of embedding a graph of degree $D \leq 3$ into a two-level grid. We say that a graph $\langle V, E \rangle$ is *embeddable* into a two-level grid $\langle \hat{V}, \hat{E} \rangle$ if there exists a couple of functions $\langle h, r \rangle$ such that

(i) $h : V \rightarrow \hat{V}$ is an injective function; and

(ii) r is a function that associates with every edge $\{i, j\} \in E$ a path P_{ij} in $\langle \hat{V}, \hat{E} \rangle$ between nodes $h(i)$ and $h(j)$ in such a way that paths associated with different edges have no internal nodes in common.

In Barahona (1982), section 4.2, a construction is described that allows one to embed graphs of degree $D \leq 3$ into two-level grids in polynomial time. For the sake of simplicity, one pads the grid obtained by the construction presented in Barahona (1982) by a suitable number of disconnected nodes so that it is possible to embed, in polynomial time, a graph $\langle V, E \rangle$ of degree $D \leq 3$ with N nodes into a two-level grid $\langle \hat{V}, \hat{E} \rangle$ such that

- (i) each level in the grid $\langle \hat{V}, \hat{E} \rangle$ has $3N \times 3N$ nodes exactly; and
- (ii) the number of vertical edges is at most $3N$.

In the following lemma a reduction of MAX CUT-3 into MAX $\{-1, 0, 1\}$ -CUT is presented. We will denote the objective functions of MAX CUT-3 and MAX $\{-1, 0, 1\}$ -CUT by w and W , respectively, and recall that, given the instances I_1 and I_2 of MAX CUT-3 and MAX $\{-1, 0, 1\}$ -CUT, $w^*(I_1) = \max_{S \in \text{Sol}(I_1)} \{w(I_1, S)\}$ and $W^*(I_2) = \max_{S \in \text{Sol}(I_2)} \{W(I_2, S)\}$.

Lemma 5.1. There exists a polynomial-time computable function f that associates, with every instance $I = \langle V, E \rangle$ of MAX CUT-3, the instance $f(I) = \langle \hat{V}, \hat{E}, J \rangle$ of MAX $\{-1, 0, 1\}$ -CUT such that

$$w^*(I) = W^*(f(I)).$$

Moreover, there exists a polynomial-time computable function g that associates, with every instance $I = \langle V, E \rangle$ of MAX CUT-3 and with every feasible solution $\hat{S} \in \text{Sol}(f(I))$, a solution $g(\hat{S}) \in \text{Sol}(I)$ such that

$$W(f(I), \hat{S}) \leq w(I, g(\hat{S})).$$

Proof of lemma 5.1. First, we consider a function f that associates with an instance $I = \langle V, E \rangle$ of MAX CUT-3 the instance $f(I) = \langle \hat{V}, \hat{E}, J \rangle$ of MAX $\{-1, 0, 1\}$ -CUT as follows.

(i) $\langle \hat{V}, \hat{E} \rangle$ is the two-level grid in which the graph $\langle V, E \rangle$ can be embedded by the construction defined in observation 5.2. The couple $\langle h, r \rangle$ realizes the embedding.

(ii) Let P_{ij} be the path in $\langle \hat{V}, \hat{E} \rangle$ associated by the map r with the edge $\{i, j\} \in E$, with $i < j$. Define the function $J : \hat{E} \rightarrow \{-1, 0, 1\}$ by assigning, for every $\{i, j\} \in E$, weight $+1$ to the edge in P_{ij} starting from the node $h(i)$ and by assigning weight -1 to all the other edges in P_{ij} ; the remaining edges on each level have weight 0 .

Observe that the weighted grid $\hat{G} = \langle \hat{V}, \hat{E}, J \rangle$ so obtained belongs to the class \mathcal{G} .

Now, consider the function g that associates with every instance $I = \langle V, E \rangle$ of MAX CUT-3, and with every feasible solution $\hat{V}_1 \in \text{Sol}(f(I))$, the solution $V_1 = g(I, \hat{V}_1) \in \text{Sol}(I)$, where node $i \in V_1$ if and only if node $h(i) \in \hat{V}_1$. Observe that the functions f and g are computable in polynomial time.

Let $V_1 \subset V$ be an optimal solution for the instance $\langle V, E \rangle$ of the MAX CUT-3 problem; by definition, $w^*(I) = |\text{cut}(V_1)|$, i.e. $\text{cut}(V_1)$ is a cut of maximum cardinality in the graph $\langle V, E \rangle$. It is easy to see that the set $\hat{E}_1 \subset \hat{E}$ of edges with weight 1 in all the paths P_{ij} with $\{i, j\} \in \text{cut}(V_1)$ is a cut in the grid \hat{G} ; the weight of the cut \hat{E}_1 is exactly $|\text{cut}(V_1)|$. This proves that $w^*(I) \leq W^*(f(I))$.

Now, given an instance I of MAX CUT-3, let \hat{V}_1 be a feasible solution of the instance $f(I)$ of MAX $\{-1, 0, 1\}$ -CUT and let $V_1 = g(I, \hat{V}_1)$ be the corresponding solution of MAX CUT-3. Let $C_{ij} = \{e \in \hat{E} \wedge e \text{ is in the path } P_{ij}\}$ and $T_{ij}(\hat{V}_1) = \sum_{e \in \text{cut}(\hat{V}_1) \cap C_{ij}} J(e)$.

Observe that if $\{i, j\} \in \text{cut}(V_1)$, then $T_{ij}(\hat{V}_1) \leq +1$, else $T_{ij}(\hat{V}_1) \leq 0$. Thus, we can state that, for all feasible solutions $\hat{S} = \hat{V}_1 \in \text{Sol}(f(I))$, the following relations hold:

$$W(f(I), \hat{V}_1) = \sum_{\{i,j\} \in E} T_{ij}(\hat{V}_1) \leq \sum_{\{i,j\} \in \text{cut}(V_1)} T_{ij}(\hat{V}_1) \leq |\text{cut}(V_1)| = w(I, V_1).$$

Finally, since $W(f(I), \hat{V}_1) \leq w(I, V_1)$ for all feasible solutions, one can state $W^*(f(I)) \leq w^*(I)$, completing the proof. \square

We are now ready to prove fact 4.2.

Proof of fact 4.2. Given an approximate polynomial-time algorithm A_{GS} for GS, we can construct an approximate polynomial-time algorithm A_{MC} for MAX CUT-3 as follows.

Algorithm A_{MC} .

- (i) Input an instance I of MAX CUT-3, $I = G = \langle V, E \rangle$
- (ii) Compute $f(I) = \langle \hat{V}, \hat{E}, J \rangle$, where f is the polynomial-time computable function defined in lemma 5.1
- (iii) Apply the algorithm A_{GS} to the weighted grid $\hat{G} = \langle \hat{V}, \hat{E}, -J \rangle$, finding the approximate solution \underline{z}
- (iv) Output $g(\underline{z})$, where g is the polynomial-time computable function defined in lemma 5.1.

From the relation between GS and MAX $\{-1, 0, 1\}$ -CUT, described in observation 5.1, and from lemma 5.1, it follows that

$$\frac{1}{2}|H^*(\hat{G}) - H(A_{GS}(\hat{G}))| \geq |w^*(G) - w(A_{MC}(G))|$$

where $H^*(\hat{G})$ and $w^*(G)$ are the optimal solutions of the GS and MAX CUT-3 problems and $H(A_{GS}(\hat{G}))$ and $w(A_{MC}(G))$ are the approximate solutions given by the algorithms A_{GS} and A_{MC} , respectively.

Since f and g are computable in polynomial time, we can conclude that algorithm A_{MC} runs in polynomial time.

Since A_{MC} is an approximate polynomial-time algorithm, we know that for infinitely many connected graphs $G = \langle V, E \rangle$, theorem 3.2 holds

$$|w^*(G) - w(A_{MC}(G))| \geq \bar{\epsilon}w^*(G).$$

Since for a connected graph $G = \langle V, E \rangle$, the cardinality of the maximum cut is at least $|V| - 1$ and recalling that $|\hat{V}| = 18|V|^2$ (see observation 5.2), we can conclude that for infinitely many spin glasses on two-level grids of the type $\hat{G} = \langle \hat{V}, \hat{E}, -J \rangle$ it holds that:

$$|H^*(\hat{G}) - H(A_{GS}(\hat{G}))| \geq \bar{\epsilon}\sqrt{|\hat{V}|/18}. \quad \square$$

6. Conclusion and unresolved problems

Spin-glass theory is a research area that has many interesting connections with other fields, such as neural networks, combinatorial optimization algorithms and parallel architectures (see Mézard *et al* (1987)).

Following Barahona's approach (1982), in this paper we analyse spin-glass models from the point of view of computational complexity; our attention is devoted to estimating bounds on the error made by approximation algorithms for finding a ground state. In particular, under the conjecture $P \neq NP$, we present a lower bound on the absolute error made by any polynomial-time approximation algorithm for the class of Ising spin glasses on two-level grids where, if n is the total number of spins, \sqrt{n} is the number of vertical connections.

In Barahona (1982), exact polynomial-time algorithms for the same class of models were excluded.

A natural development of this research is to apply the same complexity techniques to analyse more general classes of models, in particular:

(i) two-level grids without restriction on the number of vertical connections; and

(ii) structures which do not consider only nearest-neighbour interactions. This case is related to the symmetric neural networks with sequential updating introduced by Hopfield (1982) and it is relevant in the field of combinatorial optimization.

Finally, we briefly discuss the validity of some of our results in a quantum context. These results are based on the assumption that the Turing machine is an adequate model for all physically realizable devices, whereas current physical theory asserts that the universe is quantum physical. Some years ago, Feynman (1982) indicated that a quantum device might potentially be more powerful than a Turing machine; Deutsch (1985) proposed a precise model of a quantum computer, that is the quantum Turing machine, and the class BQP of languages accepted by polynomial-time quantum Turing machines with error probability at most $\frac{1}{4}$ was introduced (Bernstein and Vazirani 1993). An open question is to see whether our lower bound results hold also in the quantum context, i.e. if they are a consequence of the conjecture $NP \not\subseteq BQP$.

References

- Arora S, Lund C, Motwani R, Sudan M and Szegedy M 1992 *Proof Verification and Hardness of Approximation Problems (Proc. 33rd Ann. IEEE Symp. on Found. Comput. Sci.)* pp 14–23
- Arora S and Safra S 1992 *Probabilistic Checking of Proofs: a New Characterization of NP (Proc. 33rd Ann. IEEE Symp. Found. Comput. Sci.)* pp 2–13
- Babai L 1985 *Trading Group Theory for Randomness (Proc. 17th Ann. ACM Symp. on Theory of Computing)* pp 421–9
- Barahona F 1982 *J. Phys. A: Math. Gen.* **15** 3241–53
- Bernstein E and Vazirani U 1993 *Quantum Complexity Theory (Proc. 25th Ann. ACM Symp. on Theory of Computing)* pp 11–20
- Bieche I, Maynard R, Rammal R and Uhry J P 1980 *J. Phys. A: Math. Gen.* **13** 2553–76
- Cook S 1971 *The Complexity of Theorem-Proving Procedures (Proc. 3rd Ann. ACM Symp. on Theory of Computing)* pp 151–8
- Deutsch D 1985 Quantum theory, the Church–Turing principle and the universal quantum computer *Proc. R. Soc.* 97–117
- Feynman R P 1982 *Int. J. Theor. Phys.* **21** 467–88
- Garey M R and Johnson D S 1979 *Computers and Intractability: a Guide to the Theory of NP-completeness* (San Francisco: Freeman)
- Goldwasser S, Micali S and Rackoff C 1985 *SIAM J. Comput.* **18** 186–208
- Hartmanis J 1989 *Bull. EATCS* **38** 101–7
- Hopfield J J 1982 Neural networks and physical systems with emergent collective computational abilities *Proc. Natl Acad. Sci. USA* **79** 2554–8
- Karp R M 1972 *Reducibility Among Combinatorial Problems (Complexity of Computer Computations)* (New York: Plenum) pp 85–103
- Mézard M, Parisi G and Virasoro M A 1987 *Spin Glass Theory and Beyond* (Singapore: World Scientific)
- Papadimitriou C H and Yannakakis M 1991 *J. Comput. System Sci.* **43** 425–40
- Stockmeyer L 1987 *J. Symb. Logic* **52** 1–43