EFFICIENT SOLUTION OF MAX-SAT AND SAT VIA HIGHER ORDER BOLTZMANN MACHINES

C. Hernández, F.X. Albizuri, A.D´Anjou, M. Graña y F.J. Torrealdea
Facultad de Informática UPV/EHU, San Sebastián, Spain

ABSTRACT
High order Boltzmann machines (HOBM) have been proposed some time ago, but few applications have been reported. SAT is the canonical NP-complete problem. MAX-SAT is a generalization of SAT in sense that its solution provides answers to SAT. In this paper we propose a mapping on the MAX-SAT problem, in the propositional calculus setting, into HOBM combinatorial optimization problem. The approximate solution of MAX-SAT can be used as an approximate answer to the SAT question. An extensive experimental study of the behavior of HOBM for this problem has been conducted.

Key words: satisfiability, high order Boltzmann machines, MAX-SAT problem.

RESUMEN
Máquinas de Boltzman de alto orden (MBAO) han sido propuestas hace algún tiempo atrás, pero se han reportado pocas aplicaciones. SAT es el problema canónico NP-duro. MAX-SAT es una generalización de SAT en el sentido de que su solución proporciona respuestas para el SAT. En este trabajo proponemos una aplicación del problema MAX-SAT, en marco del cálculo proposicional, en el problema de optimización combinatoria MBAO. La solución aproximada del MAX-SAT puede usarse como una respuesta aproximativa para la pregunta del SAT. Un extenso estudio experimental del comportamiento del MBAO para este problema ha sido desarrollado.

Palabras clave: máquinas de Boltzman de alto orden, problema MAX-SAT.

1. INTRODUCTION

High order Boltzmann machines (HOBM) were proposed in [16] as a generalization of Boltzmann machines [1] and later formalized in [3]. HOBMs allow connections to be defined involving more than two units (high order connections), giving a high order polynomial as the energy function. Within the combinatorial optimization paradigm, high order connections could model problem constraints quite effectively, giving way to concise formulations of the optimization problems as HOBM´s.

The satisfiability (SAT) of an expression in the propositional calculus, usually given in conjunctive normal form, is the canonical NP-complete problem [8]. For a review of both classical and new approaches to its exact solution see [4, 6, 7, 9, 14, 15]. Selman **et al**. [18, 17] report a local search algorithm for the approximate solution of SAT and an exhaustive experimental study of the SAT problem to characterize a hard area for satisfiability. The problem of finding the maximum subset of clauses that can be satisfied simultaneously (MAX-SAT) can be viewed as a generalization of SAT. For the approximate resolution of MAX-SAT some optimization methods have been proposed [10. 12]. The present work can be framed under these trends to solve MAX-SAT. Approximate solutions of MAX-SAT are used to answer SAT. Solutions to MAX-SAT that equal the whole set of clauses allow positive answers to SAT, the failure to find them provides speculative negative answers to SAT. Of course, negative answers to SAT given on these grounds have some degree of uncertainty due to the non zero probability of suboptimal resolution of MAX-SAT.

In [5] we have proposed a mapping of the MAX-SAT problem into second order Boltzmann machines, and reported several computational experiments. The most appealing result of our work was that the Boltzmann machine approach seemed to allow an easy fit to behave linearly on the number of propositions involved, being quite unaffected by other complexity factors: the number of clauses and their structure (whether they are Horn clauses, their size, etc.). The main drawback of the primitive approach is the need to compute extra penalty functions associated with the clauses during the simulated annealing process. Modelling clauses as high-order connections, provides an inmediate formulation of the MAX-SAT problem as a HOBM. In this paper, we present the high order formulation along with a new set of experimental results more extensive than those in [5]. Moreover, new results have been obtained applying a new neighboring structure in the HOBM.

In section 2 we present the definition of the HOBM. Section 3 provides a formal definition for SAT and formulates MAX-SAT as a combinatorial optimization problem. Section 4 presents the construction of HOBM for solving an instance of SAT, and some comments on the implementations realized. Section 5 summarizes the experimental results. Finally, in section 6 we give some conclusions and further work.

2. HIGH-ORDER BOLTZMANN MACHINES

A HOMB can be defined as a triplet (U, Λ, W), where U is the set of binary units, $U = \{u_1,...,u_n\}$, Λ is the set of connections $\Lambda = \{\lambda \mid \lambda \subset U\}$ and the connection weights are given by the map $w : \Lambda \rightarrow R$. The weight of the connection λ is denoted by w_λ.

A connection is a non-empty subset of units $\lambda = \{u_{i_1},..., u_{i_n}\} \in P(U) - \phi$ where $i_1,...,i_n \in [1,...,n]$ are the indexes of units which form the connection. The set of connections which containts the unit u_j is denoted by $\Lambda_j = \{\lambda \mid u_j \in \lambda\}$. A m-order connection is a connection that has m extremes $|\lambda| = m$. The order of a HOBM is the greatest order of its connections, that is, the order of a HOBM = (U, Λ, W) is $m = \max\{|\lambda| / \lambda \in \Lambda\}$.

A configuration k is given by the global state of the HOBM and is uniquely defined by a sequence of length n, whose jth component $k(u_j) \in \{0,1\}$ denotes the state of unit $u_j \in U$ in the configuration k. The configuration space $\Re$ is given by the set of all possible configurations $\{0,1\}^n$. The activation function of connection $\lambda \in \Lambda$ in the configuration k is:

$$a_\lambda(k) = \prod_{u_i \in \lambda} k(u_i)$$

A connection is active when all its extreme units are in state 1. When a connection is active, it contributes with its weight to the consensus (energy) function. That is, the consensus or energy function associates a real value to every configuration, $C:\Re \rightarrow R$, defined as:

$$C(k) = \sum_{\lambda \in \Lambda} w_\lambda \cdot a_\lambda(k)$$

Note that the consensus function is no longer restricted to a quadratic form in the state of the units.

The behavior of the Boltzmann machine is given by an stochastic process on the set of configurations. This stochastic process is a time-homogeneous Markov chain with transition probabilities:

$$P_{kl}(T) = \frac{1}{1 + \exp\left(-\frac{\Delta C(k,l)}{T}\right)} \qquad (1)$$

where k, l are neighboring configurations according to a topology defined on the set of configurations, $\Delta C(k,l)$ is the increment in the consensus function with regard to the change from configuration k to configuration l, and T is a control parameter usually called temperature due to the similitude of the stochastic process with a physical annealing process. Being a irreducible aperiodic Markov chain, the stochastic process stationary distribution follows the Gibbs or Boltzmann distribution:

$$P_k(T) = \frac{e^{\frac{C(k)}{T}}}{\sum_{k \in \Re} e^{\frac{C(k)}{T}}}$$

As the temperature parameter is gradually lowered to zero, the Markov chain is not time-homogeneous and this distribution becomes:

$$P_k = \begin{cases} 1/n & \text{if} \quad C(k) = \max_l \{C(l)\} \\ 0 & \text{c.c.} \end{cases}$$

where n = $\left| \left\{ k = \max_l \{C(l)\} \right\} \right|$. That is, the dynamics of the Boltzman machine consists of a random search of the consensus function maxima performed through a simulated annealing procedure. This process is simulated via the random generation of neighbor configurations, and the transition to the generated configuration is randomly accepted according to the transition probability (1). In the conventional definition of the Boltzmann Machine two configurations are neighboring configurations if they only differ in the state of one unit. That is, the neighbor configuration l from a given configuration k by the change of the unit u_i is:

$$l(u_j) = \begin{cases} 1 - k(u_j) & u_j = u_i \\ k(u_j) & u_j \neq u_i \end{cases}$$

and the change in the consensus function:

$$\Delta C(k,l) = \left(1 - 2 \cdot k(u_i)\right) \cdot \left[\sum_{\lambda \in \Lambda_i} w_\lambda \cdot a_{\lambda - i}(k) \right]$$

where $a_{\lambda\text{-}i}(k) = \prod_{\substack{u_j \in \lambda \\ u_j \neq u_i}} k(u_j)$ is the activation function of connection λ in the configuration k excluding the i^{th} unit state in the calculus.

In this paper we propose and apply a configuration topology specific to the problem. This new configuration topology generates configurations which correspond to feasible solutions of the problem by changing the unit states that guarantee the correspondence with the problem.

Under this topology, we change the states of more than one unit simultaneously and the neighbor configuration l is:

$$l(u_j) = \begin{cases} 1 - k(u_j) & u_j \in U_{change} \\ k(u_j) & u_j \in U \setminus U_{change} \end{cases}$$

where $U_{change} = \{u_{i_1}, u_{i_2}, \ldots, u_{i_q}\} \subseteq U$ is the set of units subject to change.

The problem specific configuration neighborhoods can be reduced to the conventional case taking into account that given the configurations $k = k_0 \rightarrow k_1 \rightarrow k_2 \rightarrow \ldots \rightarrow k_{q-1} \rightarrow k_q = l$ where q is the cardinal of the set of units to change, the configuration k_p is defined as:

$$k_p(u_j) = \begin{cases} 1 - k_{p-1}(u_j) & u_j = u_{i_p} \\ k_{p-1}(u_j) & u_j \neq u_{i_p} \end{cases}$$

This intermediate configuration corresponds to a configuration in which only the i_p^{th} unit $u_{i_p} \in U_{change}$ changes.

The change in the consensus function can be obtained, in a general form, by sum of consensus increments between the intermediate configurations. That is:

$$\Delta C(k,l) = \sum_{p=1}^{q} \Delta C(k_{p-1}, k_p) = \sum_{p=1}^{q} \left(1 - 2 \cdot k(u_{i_p})\right) \cdot \left[\sum_{\lambda \in \Lambda_{i_p}} w_\lambda \cdot a_{\lambda - i_p}(k) \right]$$

The parallel computation properties are preserved as far as the units involved in the change of configurations are placed in the same processing element of the distributed computer.

3. SAT AND MAX-SAT

Let P denote a set of logical propositions p that can take Boolean values {*true, false*}. Let $p \in P$ denote an arbitrary proposition. A truth assignment of the propositions in P is a mapping $A : P \rightarrow$ {*true, false*}. Let L denote the set of literals that include each proposition and its negation, such that for any $l \in L$ either $l \equiv p$ or $l \equiv \neg p$. We will denote $\neg l = \bar{l}$.

A propositional expression E in conjunctive normal form (CNF) is a conjunction of a set of disjunctive clauses:

$$E = \bigwedge_{j=1}^{Nc} C_j = \bigwedge_{j=1}^{Nc} \left(\bigvee_{i=1}^{N_{C_j}} l_i \right)$$

where Nc is the clause number and N_{C_j} is the literal number in the j^{th} clause. Leu us denote ε_P the space of CNF expressions that can be defined over the set of propositions P. The evaluation of a propositional expression under a truth assignment is a mapping: $\varepsilon_A : \varepsilon_P \rightarrow$ {*true, false*}, which can be defined as follows:

$$1. \varepsilon_A(l) = \begin{cases} A(p) & \text{if } l \equiv p \\ \neg A(p) & \text{if } l \equiv \neg p \end{cases}$$

$$2. \; \varepsilon_A(C) = \bigvee_{i=1}^{N_{C_j}} \varepsilon_A(l_i)$$

$$3. \; \varepsilon_A(E) = \bigwedge_{j=1}^{Nc} = \varepsilon_A(C_j)$$

Definition 1. Given a propositional CNF expression E built up on a set of propositions P, E is satisfiable if there is at least one truth assignment A such that $\varepsilon_A(E)$ = true. The SAT problem is the problem of deciding if E is satisfiable.

Definition 2. Given a propositional CNF expression E built up on a set of propositions P, let $N_A(E)$ be the number of clauses in E that evaluate to true under the truth assignment A. The MAX-SAT problem is defined as the optimization problem

$$N_{max} = \max_A N_A(E) \tag{2}$$

The answer to the SAT problem is positive when $N_{max} = N_c$, and negative in any other case.

The MAX-SAT and SAT problems can be expressed as a 0-1 non-linear programming problem. Given the set of 0-1 variables $\{x_l \mid l \in L\}$ indexed by the set of literals L, and $x \in \{0,1\}^{|L|}$ a 0-1 assignment of the variables. This assignment of variables realices a truth assignment A if and only if

$$x_l = (1 - x_{\bar{l}}) = \begin{cases} 1 & \text{if } \varepsilon_A(l) = \text{true} \\ 0 & \text{if } \varepsilon_A(l) = \text{false} \end{cases} \; ; \; \forall l \in L$$

where $\bar{l}$ denotes the negation of literal l. Let us denote by $\Phi_x(E)$ the numerical evaluation of a propositional expression E given a assignment of the 0-1 variables x that corresponds to a truth assignment A of the set of propositions P, so that

$$\Phi_x(E) = \begin{cases} 1 & \text{if } \varepsilon_A(E) = \text{true} \\ 0 & \text{if } \varepsilon_A(E) = \text{false} \end{cases}$$

The following identities on the evaluation of logical and numerical operators are consistent with this definition:

$\Phi_x(\neg l) = 1 - \Phi_x(l) = 1 - x_1 = x_{\bar{l}}$

$$\Phi_x\left(\bigwedge_{i=1}^{N} l_i\right) = \prod_{i=1}^{N} X_{l_i}$$

The logical disjunctive operator can also be numerically realized through the application of Morgan´s law:

$$\Phi_x\left(\bigvee_{i=1}^{N} l_i\right) = \Phi_x\left(\neg\bigwedge_{i=1}^{N} \neg l_i\right) = 1 - \Phi_x\left(\bigwedge_{i=1}^{N} \neg l_i\right) = 1 - \prod_{i=1}^{N}(1 - x_{l_i})$$

We can, therefore, express the numerical evaluation of a CNF propositional expression as:

$$\varepsilon_A(E) \equiv \Phi_x(E) = \prod_{j=1}^{Nc}(1 - \Phi_x(\neg Cj)) = \prod_{j=1}^{Nc}\left(1 - \prod_{i=1}^{N_{C_j}}(1 - x_{l_i})\right)$$

Definition 3. Given a propositional CNF expression E defined on the set of propositions P, the MAX-SAT problem can be stated as the 0-1 maximization problem

$$N_{max} = \max_x \sum_{j=1}^{Nc}\left(1 - \prod_{i=1}^{N_{C_j}}(1 - x_{l_i})\right) \quad (3)$$

subjetc to $x_l = (1 - x_{\bar{l}})$; $\forall l \in L$ (4)

The constraints force the values of the 0-1 variables to be a truth assignment.

Definition 4. Given a propositional CNF expression E defined on the set of propositions P, the SAT problem can be solved through the solution of MAX-SAT if

$N_{max} = N_c$

which is equivalent to

$\Phi_x(E) = 1$

4. HIGH-ORDER BOLTZMANN MACHINES FOR MAX-SAT

The formulation of MAX-SAT and SAT as a 0-1 programming problem leads to an inmediate mapping of the problems into HOBM. The mapping of a combinatorial optimization problem into the HOBM involves the definition of the unit set, the connections, and the weights, such that the consensus function maxima correspond to solutions of the problem, and besides, the solutions have greater consensus than configurations that do not correspond to problem solutions.

Given a CNF propositional expression E defined on the set of propositions P, an HOBM = (U, Λ, W) for the approximate solution of MAX-SAT is built up as follows.

The set of units corresponds to the set of 0-1 variables so that $U = \{u_l \mid l \in L\}$. We maintain the convenient notation $\neg l = \bar{l}$, so that u_l and $u_{\bar{l}}$ denote a pair of complementary units that represent a proposition and its negation. The state $k(u_l)$ of each unit corresponds to the value of the corresponding 0-1 variable x_l, and a configuration corresponds to a state x_k of the 0-1 variables X. We allow ourselves the freedom of using the literals as unit indexes.

Definition 5. A configuration $k \in \Re$ of the HOBM is consistent, if and only if:

$$k(u_p) = 1 - k(u_{\bar{p}})\ \forall_p \in P \qquad (5)$$

A consistent configuration can be interpreted as a truth assignment to the set of propositions. This definition partitions the configuration space into inconsistent and consistent regions,
$\Re = \Re^i \cup \Re^c$

We propose two ways for defining connections and weights, depending of the neighborhoods assumed for the configurations. In the conventional neighborhood, connections must be included to ensure that configurations that correspond to truth assignments (consistent configurations) have a greater consensus than inconsistent ones. In the problem specific configuration topology no inconsistent configurations are allowed as states of the HOBM.

4.1. The mapping under the general configuration neighborhood

The conventional configuration neighborhood structure of the HOBM, needs to incorporate the truth assignment constraints through some connections and weights. The set of connections is partitioned into two disjoint subsets $\Lambda = \Lambda^C \cup \Lambda^*$. The clauses are modelled by the connections in Λ^C. These connections are constructed, following the formulation of the 0-1 numerical optimization given by (3) and (4), as clause negations:

$$\Lambda^C = \{\lambda_j \mid (u_{\bar{p}} \in \lambda_j \Leftrightarrow p \in C_j) \wedge j = 1,...,Nc\}$$

If $k \in \Re^c$, the activation of the clause connection corresponds to the evaluation of the negated clause:

$$a_{\lambda_j}(k) \equiv \Phi_{x_k} = (\neg C_j)$$

therefore, the number of clauses satisfied by the truth assignment realized by configuration k of the HOBM is

$$N_{SAT}(k) = N_c - \sum_{\lambda \in \Lambda^C} a_\lambda(k)$$

The connections in Λ^* model the truth assignment constraint. In its turn Λ^* is partitioned into bias and inhibitory connections:

$$\Lambda^* = \Lambda^I \cup \Lambda^B$$

$$\Lambda^B = \{\lambda = \{u_p\} \mid \forall u_p \in U\}$$

$$\Lambda^I = \{\lambda = \{u_p, u_{\bar{p}}\} \mid \forall p \in P\}$$

The bias connection weights are positive values $w_\lambda = \alpha;\ \forall \lambda \in \Lambda^B$, the inhibitory connections weights are negative values $w_\lambda = \beta;\ \forall \lambda \in \Lambda^I$ and connections modelling clauses have a negative identical weight $w_{\lambda_j} = -\gamma;\ \forall \lambda_j \in \Lambda^C$.

Definition 6. The consensus function of the HOBM that has been defined to solve the satisfiability problem of a propositional expression, is:

$$C(k) = \sum_{\lambda \in \Lambda^B} \alpha \cdot a_\lambda(k) - \sum_{\lambda \in \Lambda^I} \beta \cdot a_\lambda(k) = \sum_{\lambda \in \Lambda^C} \gamma \cdot a_\lambda(k) = \alpha N_\alpha^{(k)} - \beta N_\beta^{(k)} - \gamma N_\gamma^{(k)}$$

where $N_\alpha^{(k)}, N_\beta^{(k)}$ are the number of active bias and inhibitory connections, respectively, and $c N_\gamma^{(k)}$ is the number of active clause connections in the configuration k, regardless of the consistency of k. Notice $N_{SAT}(k)$ is identical to N_c - $N_\gamma^{(k)}$, but for its restriction to consistent configuration.

The definition of the weight values α, β, γ must ensure that the consensus function possesses two basic properties:

1. All inconsistent configurations have a consensus value lower than any consistent configuration. This guarantees that the HOBM will eventually converge to a consistent configuration.
2. The consensus will monotonically increase with the number of satisfied clauses.

The following propositions prove the conditions on the weight values that ensure them. In [5] we proved similar results for the conventional Boltzmann Machine applied to the SAT problem. The results on the HOBM are much more simple and elegant. We have also found that the HOBM is much more efficient and accurate in practice.

Proposition 7. Given the above definition of the HOBM units, connections and consensus function for a given propositional expression E, all inconsistent configurations have a consensus value lower than any consistent configuration in the following conditions on the weights hold

$\alpha > 0$

$\gamma > 0$

$\beta > \alpha + \gamma \cdot Nc$

Proof. Le us denote $N_i^{(k)}$ the number of inconsistencies $k(u_p) = k(u_{\bar{p}})$ produced by pairs of complementary units with the same state in configuration $k \in \Re$. If $N_i^{(k)} = 0$, the configuration is consistent and its consensus is: $C(k) = \alpha NP - \gamma \cdot N_\gamma^{(k)}$. Let us consider two configurations k^1, $k^2 \in \Re$ for whom $N_i^{(k^1)} = t-1$ and $N_i^{(k^2)} = t$ respectively, $t \geq 1$. The difference in the number of inconsistencies is produced by a pair of complementary units such that $k^1(u_p) \neq k^1(u_{\bar{p}})$ and $k^2 = (u_p) = k^2(u_{\bar{p}})$. The proof of the proposition follows by induction over $N_i^{(k)}$ if the following holds:

$$C(k^1) - C(k^2) > 0 \qquad (6)$$

The proof of this condition depends on the type of the inconsistency:

Case 1. If $k^2(u_p) = k^2(u_{\bar{p}}) = 0$, the consensus difference between these configurations is:

$$C(k^1) - C(k^2) = \alpha + \gamma \left(N_\gamma^{(k^2)} - N_\gamma^{(k^1)} \right)$$

thus, the satisfaction of (6) becomes the following condition upon the bias weights:

$$\alpha > -\gamma\left(N_\gamma^{(k^2)} - N_\gamma^{(k^1)} \right)$$

But $N_\gamma^{(k^2)} \geq N_\gamma^{(k^1)}$, because the zero state can only contribute to the decrement of the number of active clause connections. Therefore, (6) is trivially satisfied if:

$\alpha > 0$

Case 2. If $k^2(u_p) = k^2(u_{\bar{p}}) = 1$, the consensus difference between these configurations is:

$$C(k^1) - C(k^2) = \beta - \alpha - \gamma\left(N_\gamma^{(k^2)} - N_\gamma^{(k^1)} \right)$$

thus, the satisfaction of (6) becomes the following condition upon the inhibitory weights:

$$\beta > \alpha - \gamma \left(N_\gamma^{(k^2)} - N_\gamma^{(k^1)} \right)$$

In the worst case, the inconsistency involves the activation of all the clause connections, therefore, (6) is satisfied in any case when:
$\beta > \alpha + \gamma \cdot Nc$

Proposition 8. Given any positive value of γ, and α, β fulfilling the conditions of the previous proposition, the consensus function grows monotonically with the number of satisfied clauses. Given consistent configurations $k^1, k^2 \in \Re^c$,

$$N_{SAT}(k^1) < N_{SAT}(k^2) \Leftrightarrow C(k^1) < C(k^2)$$

Proof. The conditions on α, β guarantee that we need only to consider consistent configurations. Therefore the consensus function value for both configurations is:

$$C(k^1) = \alpha Np - \gamma N_\gamma^{(k^1)} = \alpha Np - \gamma N_c + \gamma N_{SAT}(k^1)$$

$$C(k^2) = \alpha Np - \gamma N_\gamma^{(k^2)} = \alpha Np - \gamma N_c + \gamma N_{SAT}(k^2)$$

From these expressions the two sides of the implication follow trivially for any $\gamma > 0$:

$$C(k^1) < C(k^2) \Rightarrow N_{SAT}(k^1) < N_{SAT}(k^2)$$

$$N_{SAT}(k^1) < N_{SAT}(k^2) \Rightarrow C(k^1) < C(k^2)$$

Corollary 9. For a given satisfiable propositional expression E, the configurations with maximum consensus of the HOBM built up as described before corresponds to truth assignments that satisfy it:

$$N_{SAT}(k^*) = N_c \Rightarrow C(k^*) = \max_k C(k) = Np \cdot \alpha$$

Therefore, the HOBM will converge to positive solutions of the SAT problem posed on E. If the propositional expression is not satisfiable, the HOBM will converge to a solution of the MAX-SAT problem posed on E. Let N_{max} be the maximum number of satisfiable clauses as defined in (2), then

$$N_{SAT}(k^*) = N_{max} \Rightarrow C(k^*) = \max_k C(k) = Np \cdot \alpha - \gamma(N_c - N_{max})$$

4.2. The mapping under the problem specific configuration neighborhood

The structure of the connections in the previous mapping is conditioned by the neighboring assumed between configurations. The constraint connections Λ^* where imposed by the need to guarantee the convergence to consistent configurations. Assuming that only consistent configurations can be reached the HOBM is greatly simplified and its performance improved, at the cost of loosing some of its parallel computation capabilities. The neighboring configuration is generated in a two steps process:

1. A unit u_p is selected to change its state.

2. The complementary unit state $u_{\bar{p}}$ is also changed.

Definition 10. The consensus function of the HOBM that has been defined to solve the satisfiability problem of a propositional expression, is:

$$C(k) = -\sum_{\lambda \in \Lambda^C} \gamma \cdot a_\lambda(k) = -\lambda N_\gamma^{(k)}$$

where $N_\gamma^{(k)}$ is the number of active clause connections in the configuration k. Notice that in this case $N_{SAT}(k)$ is always identical to $N_c = N_\gamma^{(k)}$, because the configuration space is restricted to the subspace of consistent configurations.

The proof of proposition (6) is not longer needed and that of proposition (7) is trivial under this new mapping.

4. EXPERIMENTAL RESULTS

The results reported in this section were obtained over randomly generated sets of instances. Parameters of this generation were the number of clauses, the number of propositions and the number of literals per clause. We have generated 30 instances per each parameter combination in order to get a mean execution time and an average MAX-SAT. The instances have been generated according to the Random K-SAT model [13]. The number of literals per clause K has been 2, 3 and an uniform distribution between 2 and 7 denoted *du* in the figures. We have performed experiments with up to 1000 clauses and 1000 propositions, very large problems in comparison with the benchmark usually used in experimental studied of SAT. The annealing schedule is the same applied for the experiments reported in [5]. We have reproduced the experiments in [5] applying the HOBM approach.

Figure 1. Mean execution time versus proposition number for 1000 clauses.

We have observed similar results regarding the computational complexity of our approach: the execution time measured by the number of transition trials depends linearly on the number of propositions, and it is quite invariant with respect to the number of clauses. For example, Figure 1 reproduces the linear dependence of the execution time on the number of clauses. We have been able to extend the experiments in [5] to much larger problems (Np = 1000, Nc = 1000). However, a central issue of the usefulness of the HOBM is its accuracy: the confidence that we have in negative responses to SAT. This was the issue for the next experiments.

5.1. Accuracy; the satisfiability regions

Within the scientific community devoted to the approximate solution of SAT, there is a general acceptance of the existence of three broad regions in the space of propositional expressions [5, 18, 17] regarding the solution of SAT.

- A region of EASY satisfiable expressions, for whom a truth assignment that satisfies them is easy to find, and the search is fast.
- A region of HARDLY satisfiable expressions. In this region the probability of finding unsatisfiable expressions is not negligible, and the satisfiable ones have a small set of truth assignments that satisfies them, so the complexity of the search, for the exact solution, becomes exponential.

- A region of EASY unsatisfiability. In this region the probability of finding a satisfiable expression is negligible.

Diverse works have been directed to the identification of these regions [5], [18], [17]. The ratio $\frac{Nc}{Np}$ has been adopted as the parameter describing the complexity of the propositional expression. The above region boundaries depend on the clause structure. For 3-SAT, the above regions can be roughly characterized in terms of this ratio as follows:

Figure 2. Satisfaction rate for random cases of 100 clauses versus the proposition number.

- EASY unsatisfiability $\frac{Nc}{Np} < 3$
- HARDLY satisfiable expressions $3 < \frac{Nc}{Np} < 5$
- EASY satisfiable expressions $\frac{Nc}{Np} > 5$

For the exploration of the response of the HOBM in these regions we have focused in the SAT problem with 100 clauses, a manageable complexity level for systematics exploration. For each sampled value of the radio $\frac{Nc}{Np}$ we have generated 30 instances. Each instance was solved exactly using the Davis-Putnam algorithm, and approximately using the HOBM. Figure 2, shows the evolution of the approximated probability as the ratio decreases, for 2-SAT, 3-SAT and du-SAT. The dependence on the clause structure is evident. We have found that the HOBM result is 100 % consistent with the exact solution in the EASY satisfiable and unsatisfiable regions. In the region of hardly decidable expressions, there is a non negligible probability of error produced by declaring unsatisfiable a satisfiable propositional expression.

A more detailed experiment has been performed in the hard satisfiability region for 3-SAT clauses, generating 60 instances for each value of the ratio and sampling it in smaller increments. Figure 3 shows the

Figure 3. (a) Satisfaction rate for random cases.

(b) Error rate incurred by the HOBM with conventional configuration neighborhoods for the random cases generated in the study of the hard area for satisfiability.

results that allow an evaluation of the error incurred by the HOBM. Figure 3(a) presents the error in the ratio of satisfactions found by the HOBM. Figure 3(b) shows the same result as a relative error. These results show that the error increase around 4, which corresponds to a 0.5 probability of finding an unsatisfiable instance. The results shown in Figure 3 were computed with the conventional mapping. In Figure 4 we reproduce the results of the application of the mapping into the HOBM induced by the use of the consistent neighborhoods. The result is the relative error of the HOBM in the region already explored in Figure 3. The figure shows a dramatic improvement of the simplified HOBM over the conventional HOBM. Our conclusion is that the second mapping is much more accurate.

5.2. The MAX-SAT solutions

The last set of results, reproduced in Figure 5, are devoted to the ability of the HOBM to search for MAX-SAT solutions. Recall that, as stated in section 4, the HOBM performs the search for the maximum set of satisfiable clauses, and the answer to SAT depends on the solution found to MAX-SAT. The figure shows the average size of maximum satisfiable subsets of clauses for expressions of 100 clauses. For all the clause structures, the set of satisfied clauses found by the HOBM is above the 95 % of the number of clauses in the region of hard satisfiability. This result means that the HOBM is very efficient as a MAX-SAT solver. The exact computation of the error incurred by HOBM as MAX-SAT has a prohibitive computational cost. We exphasize the role of HOBM as a MAX-SAT solver because it allows to identify the unsatisfiable clauses, if the expression is not satisfiable, or the hard-to-satisfy clauses, if the problem is satisfiable and the response is not satisfaction. This information can be of some use for the modification of the propositional expression to suit some practical application.

Figure 4. Error rate incurred by the HOBM with conventional and problem specific configuration neighborhoods calculated by the Davis-Putnam algorithm for random generated cases in the study of the hard area for satisfiability.

6. CONCLUSIONS AND FURTHER WORK

We have presented a mapping of SAT and MAX-SAT problem instances into an HOBM that produces approximate responses to these problems. The mapping into the HOBM is based in the formulation of SAT and MAX-SAT as 0-1 non linear optimization problems. We have found that the HOBM retains the good computational features already present in the Boltzmann machine, while increasing its accuracy.

A detailed study of the accuracy of both mapping into HOBM discussed in section 4 has been performed. The results show that the simplified HOBM, based on a neighborhood defined only over consistent configurations, is much more accurate. The main disadvantage of the simplified HOBM is the loss of some parallel computation capabilities: the units representing the proposition and its clause must be in the same processing element in any distributed implementation. This restriction however is not very strong, as the most recent works on the distributed implementation of Boltzmann Machines do consider coarse grain implementations as advantageous in many respects.

Figure 5. MAX-SAT average for random cases of 100 clauses versus the proposition number.

Currently we are working in the mapping of other combinatorial optimization problems, like the Traveling Salesman Problem, or the Stockcutting problem into the HOBM. Preliminary results show that HOBM can be a competitive algorithm againts other more conventional solutions. We are also taking into consideration the study of the learning capacity of the HOBM in terms of SAT and MAX-SAT. Most studies on the capacity of neural networks are based on their function approximation capabilities. The focus in our work will shift to the evaluation of the distance between the desired propositional expression to be learned, and the one effectively apprehended by the HOBM from the data.

ACKNOWLEDGMENTS

This work has been performed under the support of projects UE97/16 and IT97/2 of the Dept. Educación, Univ. e Inv. del Gobierno Vasco. Also we have the support of the Dept. Economía de la Excma. Diputación de Guipúzcoa.

REFERENCES

[1] AARTS, E.H.L. and J.H.M. KORST (1988): **Simulated annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing**, Wiley, N.Y.

[2] ACKLEY, D.H.; G.E. HINTON and T.J. SEJNOWSKI (1985): "A learning algorithm for Boltzmann machines", **Cognitive Science** 9, 147-169.

[3] ALBIZURI, F.X.; A. D´ANJOU; M. GRAÑA, J. TORREALDEA and C. HERNANDEZ (1995): "The high-order Boltzmann machine: learned distribution and topology", **IEEE Transactions on Neural Networks** 6, 767-770.

[4] DAVIS, M. and H. PUTNAM (1960): "A COMPUTING PROCEDURE FOR QUANTIFICATION THEORY", **Journal of ACM**, 7, 201-215.

[5] D´ANJOU, A.; M. GRAÑA, F.J. TORREALDEA and C. HERNANDEZ (1993): "Solving satisfiability via Boltzmann Machines", **IEEE Transaction of Pattern Analysis and Machine Intelligence** 15, 514-521.

[6] DOWLING, W.F. and J.H. GALLIER (1984): "Linear-time algorithms for testing the satisfiability of propositional Horn formulae", **Journal of Logic Programming**, 3, 267-284.

[7] GALLO, G. and G. URBANI (1989): "Algorithms for testing the satisfiability of propositional formulae", **Journal of Logic Programming**, 7, 45-61.

[8] GAREY, M.R. and D.S. JONHSON (1979): **Computers and intractability. A guide to the theory of NP-completeness** (W.H. Freeman, N.Y.).

[9] GELDER, A. van (1988): "A satisfiability tester for non-clausal propositional calculus", **Information and Computation**, 79, 1-21.

[10] HANSEN, P. and B. JAUMARD (1990): "Algorithms for the maximum satisfiability problem", **Computing**, 44, 279-303.

[11] HINTON, G.E. and T.J. SEJNOWSKI (1986): "Learning and Relearning in Boltzmann Machines", in: D.E. Rumelhart **et al**., eds., **Parallel Distributed Processing PDP: Foundations** 1 (MIT Press, Cambridge, MA) 283-317.

[12] LIEBERHERR, K.J. (1982): "Algorithmic extremal problems in combinatorial optimization", **Journal of Algorithms**, 3, 225-244.

[13] MITCHELL, D.G.; B. SELMAN and H.L. LEVESQUE (1992): "Hard and Easy Distributions of SAT Problems", in: **Proceedings of AAAI-92** (San José, CA) 459-465.

[14] MONIEN, B. and E. SPECKENMEYER (1985): "Solving satisfiability in less than 2^n steps", **Discrete Applied Mathematical** 10, 287-295.

[15] PURDOM, P.W. (1984): "Solving satisfiability with less searching", **IEEE Transactions on Pattern Analysis and Machine Intelligence** 6(4), 510-513.

[16] SEJNOWSKI, T.J. (1986): "Higher-order Boltzmann Machines", in: **AIP Conference Proceedings** (Snowbird, UT 151), 398-403.

[17] SELMAN, B. and H.A. KAUTZ (1993): "An empirical study of greedy local search for satisfiability testing", in: **Proceedings of AAAI-93**, (Washington, DC), 46-51.

[18] SELMAN, B.; D.G. MITCHELL and H.J. LEVESQUE (1996): "Generating hard satisfiability problems", **Artificial Intelligence** 81, (1-2), 17-29.