Correspondence

On the Approximation of NP-Complete Problems by Using the Boltzmann Machine Method: The Cases of Some Covering and Packing Problems

V. Zissimopoulos, V. Th. Paschos, and F. Pekergin

***Abstract*—We describe a *Boltzmann Machine (BM) Architecture* in order to solve the problems of *Maximum Independent Set (IS), Set Partitioning (SPP), Clique, Minimum Vertex Cover (VC), Minimum Set Cover (SC),* and *Maximum Set Packing (SP)*. We evaluate the *maximum* and the *average* error of the method where the *error* is defined as the ratio of the cardinality of the obtained solution for an instance with respect to the optimal one. Moreover, we compare our results with the results obtained from the implementation of the heuristic described in [6]. Our model treats the general case of all these problems that is the case when costs are associated with the data (vertices or subsets) describing them, the unweighted case becoming a particular case in our approach. As we show it succeeds to find optimal solutions for a large percentage of the treated instances, for the rest providing a very good performance ratio.**

***Index Terms*— Approximation algorithms, Boltzmann machine, combinatorial optimization, consensus function, feasibility, heuristic, neural network, NP-complete.**

I. Introduction

In this paper, we present a BM architecture [1], [2] for solving SPP, SP, IS, and its complementary VC as well as the Clique problem and we discuss a possible way for solving SC by using the same model. The described model concerns the general case where costs are associated with each variable (each vertex or each subset) and deals with unweighted cases of each of these problems as particular cases. Connections are established to express local constraints, that is, the desirability about incompatible individual states of connected units. Inhibitory connections expressed by negative weights are used to imply antagonistic competition between linked units appearing in a final solution. Globality about constraints satisfaction is considered through the values of the individual weights. These values are chosen such that the global energy of the network reflects the extent of the consensus of the units states. In a general case, approaching an optimization problem by a BM requires local optima of the networks global energy to be feasible solutions of the implemented problem [1]. Moreover, the order induced by the optimization function on any couple of feasible solutions is required to remain the same for the values of the consensus on the corresponding configurations of the BM. Consequently, since we know that a BM with symmetrical strengths always converges to a local optimal (maximal consensus or minimal energy) ([1], [5], [10], [11]), we can obtain, obviously, a feasible solution, and if a mechanism such as simulated annealing [7] is applied to escape from a local optimal to a better local optimal (higher maximal), we are able to obtain a better feasible solution.

Manuscript received February 1, 1991; revised June 12, 1991.
V. Zissimopoulos is with LRI, Université de Paris 11, Centre d'Orsay, 91405 Orsay, France.
V. Th. Paschos is with LRI, Université de Paris 11, Centre d'Orsay, 91405 Orsay, France and CERMSEM, Université de Paris 1-Sorbonne, 75005 Paris, France.
F. Pekergin is with EHEI, Université de Paris 5 (René Descartes), 75006 Paris, France.
IEEE Log Number 9103503.

Thus, if the BM is stabilized over a configuration with the Maximum Consensus, the optimum solution for the optimization problem is obtained. Unfortunately, for an optimization problem it is not evident how to obtain a set of weights leading to a global energy with the previous interesting characteristics. We mention here that in [1], IS is also treated but only in the unweighted case. In this paper, we generalize this construction for the case where costs are associated to each vertex of the graph. We prove that the proposed weights are "well defined," and lead to a suitable global energy function. Moreover, our procedure permits us to "see" the case of cardinality as a particular case.

We treat firstly VC and IS. These two problems are known to be equivalent with respect to their complexity [4], [6]. In fact, there is a very simple relation between the cardinalities of a VC and an IS in a graph. *Given a graph G, an IS is the complement of a VC with respect to the vertex set of G.* Thus, any algorithm that solves the IS gives also the solution for VC by a simple subtraction. Moreover, a method solving these two problems can also solve the Clique Problem because *given a graph G, a set of vertices constituting an independent set on G induces a subgraph that constitutes a clique on $\overline{G}$ where $\overline{G}$ is the complement of G constructed as follows: (u, v) is an edge of $\overline{G}$ iff u and v are not adjacent in G.* Moreover, SP and IS are isomorphic. Thus, the method for IS is immediately applicable to the case of SP. For SPP it is proved that this problem (with or without costs) is equivalent to an IS with costs under the condition that the former one is feasible. Finally we discuss the case of SC. We show that even SC can be approached by the BM model presented in this paper. We test the performance of our model by giving simulation results for SPP, IS, and VC. Of course, the model can be immediately used, as we have already discussed, to reflect also the cases of SP, SC, and Clique.

This paper is organized as follows. In Section II we present the definitions of the examined problems. We discuss also some interesting properties of the representations adopted. In Section III, we discuss all the details of the neural network architecture of our BM model. In the same section also we prove some nontrivial equivalences between the problems we deal with, and we study the properties of the consensus function defined to deal with our model. Finally in Section IV, we present a summary of the obtained results and we make comments about them. We use many performance measures to characterize the effectiveness of the designed BM. These measures give us several kinds of information such as the ratio of the obtained solutions with respect to the optimal ones, the percentage of the optimal solutions obtained by using the proposed method, etc. In the cases of VC and IS we present also information about the results provided by the heuristic of [6].

II. The Problems: Definitions and Adopted Representations

In what follows we give the definitions of the problems as integer linear programs. The definitions for the unweighted cases can be found in [4]. In the sequel, all vectors are column vectors. In what concerns VC and IS we consider a graph $G = (V, E)$, while for SPP, SP, and SC we consider the couple $\mathcal{S}$, Q, and where $\mathcal{S}$ is a family of subsets of the set Q.

Independent Set:

$$\max \sum_{j=1}^{n} c_j x_j \quad \text{subject to} \sum_{j=1}^{n} a_{ij} x_j \leq 1$$

where $i = 1, \cdots, m$, $x_j \in \{0,1\}$, $j = 1, \cdots, n$, $m = |E|$, $n = |V|$, a with $a_{ij} = 0$ or 1 the edge-vertex incidence matrix of G and $c = (c_j)$ the profit vector or cost vector. If $c_j = 1$, $j = 1, \cdots, n$, we look for a subset of nodes which are not adjacent in G (cardinality case).

Vertex Cover:

$$\min \sum_{j=1}^{n} c_j x_j \quad \text{subject to} \sum_{j=1}^{n} a_{ij} x_j \geq 1$$

where $i = 1, \cdots, m$, $x_j \in \{0,1\}$, $j = 1, \cdots, n$, $m = |E|$, $n = |V|$, a with $a_{ij} = 0$ or 1, the edge-vertex incidence matrix of G and $c = (c_j)$ the cost vector (cardinality case when $c_j = 1, j = 1, \cdots, n$). The clique following the discussion of the previous section is defined similarly to IS.

Set Partitioning:

$$\min c^T x \quad \text{subject to } Ax = e$$

where $x_j \in \{0,1\}$, $j = 1, \cdots, n$, $|S| = n$, $|Q| = m$, c^T is the transposed vector of the cost vector, x is the solution vector, e is the m-dimensional unit vector, and $A = (a_{ij})$, $i = 1, \cdots, m$ and $a_{ij} = 1$ if $q_i \in S_j$ and 0 otherwise.

Set Packing:

$$\max c^T x \quad \text{subject to } Ax \leq e$$

where $x_j \in \{0,1\}$, $j = 1, \cdots, n$, $|S| = n$, $|Q| = m$, c^T is the transposed vector of the cost vector, x is the solution vector, e is defined as above, and $A = (a_{ij})$, $i = 1, \cdots, m$, with $a_{ij} = 1$ if $q_i \in S_j$ and 0 otherwise.

Set Cover:

$$\min c^T x \quad \text{subject to } Ax \geq e$$

where $x_j \in \{0,1\}$, $j = 1, \cdots, n$, $|S| = n$, $|Q| = m$, c^T is the transposed vector of the cost vector, x is the solution vector, e is defined as above, and $A = (a_{ij})$, $i = 1, \cdots, m$, with $a_{ij} = 1$ if $q_i \in S_j$ and 0 otherwise.

For the instances of SC, SP, and SPP we construct the following characteristic graph.

Definition 1: The characteristic graph $B = (S, Q, E)$ of an instance of SC (resp., SP, SPP) is a bipartite graph defined as follows:

S: the vertex set of B corresponding to the family of subsets, Q: the vertex set of B corresponding to the set Q and $E = \{(S_i, q_j) : q_j \in S_i \wedge q_j \in Q\}$. □

Moreover, for those problems we can construct another kind of characteristic graph, called the *intersection graph.*

Definition 2: Given a graph B (definition 1) that characterizes an instance of SC (resp., SP, SPP) the intersection graph $G_S = (S, E, l)$ is constructed as follows:

S: The vertex set S of B, $E = \{(s_i, s_j) : \text{there is a path } [S_i, q_k, S_j] \text{ in } B\}$, l: a function from $E(G_S)$ to $Q(B)$ such that if $(s_i, s_j) \in E(G_S)$ and $[S_i, q_k, S_j]$ is a path of length two in B then $l((s_i, s_j)) = q_k$. □

Of course, as there can exist more than one path of length 2 between two given vertices of B, G_S is rather a multigraph. In the case of SP and SPP, as we can see, we do not need to use the labeling function and thus we consider graph G_S as an ordered pair (S, E). For the relation between SP and IS the following proposition holds [8].

Proposition 1: When an instance of SP is represented in terms of G_S, then each solution of SP is exactly an independent set of G_S.

III. Neural Network Architecture

Let $G = (V, E)$ be an undirected graph where $V = \{v_i\}$, $1 \leq i \leq n$ is the set of vertices and E is the set of edges. The neural network is constructed to be ismorphic to the graph G. Each vertex corresponds to a neuron i which is in a state $u_i \in \{1, 0\}$ at each time step $t \in \{0, 1, \cdots\}$. If there is an edge between vertices v_i and v_j then the corresponding neurons i and j are linked with an appropriate strength w_{ij}, otherwise they are not linked or the strength is zero. At each neuron i, there is an external or bias connection expressing the desirability for this unit to be included in the final solution. At any time a configuration of the network, i.e., a global state vector, determines a solution of the implemented combinatorial optimization problem. The connection strengths, imposing local constraints, express the compatibility of two linked units to be both included in the final solution. The values of the connection strengths are chosen in such a way that they ensure mutual exclusion between adjacent units and define the network's energy in a manner that this energy characterizes a suitable overview about global satisfaction of the constraints and global optimality of the optimization function. The network's energy is a function of the states of units and of the strengths on the links between them. It expresses an overall measure of the consensus reached by the states of the units of the BM. For each configuration k of the BM this consensus function C is defined as follows: $C(k) = \sum_{ij} w_{ij} u_i u_j$, where i and j range over all pairs such that there exists a connection between units i and j. The appropriate values of the strengths, which reflect the suitable global overview, should be chosen such that the above consensus function C is feasible, i.e., all local maxima of the consensus function should correspond to feasible solutions (feasibility). Additionally, the consensus function has to be order preserving. This means that, for two configurations of the BM which represent feasible solutions, the optimization function and the consensus function have the same order. The *feasibility* condition of the consensus function implies that when the BM is stuck in a local optimum, a feasible solution is always found. The *order preserving* condition implies that when the BM reaches the maximum consensus (near-maximum), the optimal (near-optimal) solution is found. Futhermore, in order to make the BM escape from locally optimal configurations, a stochastic acceptance criterion is used, allowing the units to adjust their states to those of their neighbors. The generation mechanism of new configurations used here permits only one unit at a time to change its state (sequential BM). Certainly, since the network is not fully connected, a limited parallelism, where nonadjacent units may change their states in parallel, could be applied without introducing erroneously calculated state transitions. This limited parallelism in synchronous or asychronous mode would conserve asymptotical convergence to the set of optimal configurations [1], [3]. The consensus of the BM for each new configuration, by changing the state of a unit i, is completely determined by the states of the neighbors of the unit i and the corresponding strengths, since [according to the definition of $C(k)$] the difference is calculated to be $\Delta C = (1 - 2u_i)\left[\sum_j w_{ij} u_j + w_{ii}\right]$ where $u_i = 1$ if current state of unit i proposed to be changed is "ON" and $u_i = 0$ otherwise, and all j range over the units connected with unit i. Consequently the adjustment of each unit and the evaluation of its state transition can be performed locally, implying a potential for parallel execution. If the consensus for a given configuration of the BM does not increase by a single state transition $\Delta C \leq 0$ for all units, then the BM is lying on a local maximal. The updating rule UR, reflecting the typical sigmoid response of neurons in a (biological) neural network is selected as follows: P (state u_i is changed) $= \frac{1}{1+e^{-\Delta C/T}}$. The term ΔC stands for the difference in consensus between two consecutive configurations. The parameter T is a control parameter. In this work the following law is used: $T_n = \frac{T_{n-1}}{1+\log(t)}$, where $t = f(n)$ and $f(n) = f(n-1)(1+\text{rate})$ with $f(0) = 1.0$, $n = 1, 2, \cdots$ and *rate* is a small suitable parameter to control the decrement of T. The value of this parameter permits us to choose short or long decrements which in relation to the length of the generated Markovian chains will significantly influence the quality of the solutions and the required computation time. The previously discussed model can be used for the examined problem as follows. *We consider the graph G that represents the intersection graph in the cases of SPP, SP, and SC problems. In the cases of IS and VC, we consider directly the initial graph. We construct the BM network to be isomorphic to this graph. Therefore, each unit corresponds to a subset for the SPP, the SP and the SC or to a vertex for the IS and the VC.*

The Independent Set Problem: The external connections are of positive weights, the connections between units which represent linked vertices in G are of negative weights, while the nonlinked vertices in G are not connected (or of zero weights) in the BM model. More precisely the connection matrix is as follows:

$$w_{ij} = c_i \quad \text{if } i = j \quad \text{and}$$
$$w_{ij} = -[\max\{c_i, c_j\} + \epsilon]\delta_{ij} \quad \text{if } i \neq j$$

where $\delta_{ij} = 1$ if $\{i, j\} \in E$ and 0 otherwise. In the above equation, c_i is the cost associated to the vertex i and ϵ is a very small real positive parameter whose importance is discussed in remark 3.

Remark 1: The weights defined above remain valid for the unweighted case. It is sufficient to put $c_i = 1$, $i = 1, 2, \cdots n$ and we will find the same weights proposed in [1]. The solution vector of IS is equal to the vector state of the BM.

Remark 2: It can be proved that the proposed weights are "well defined" and lead to a consensus function having the good properties, discussed in the previous section.

Theorem 1: When dealing with IS the stable states of the network with the above connection matrix correspond to solutions which are *feasible* and *maximal.*

Proof: a) *feasibility:* Let us consider a configuration k of the BM inducing a nonfeasible solution x^k. It can be proved that this configuration is not a local maximum. In fact, since x^k is not feasible there are adjacent units i and j, both on state "ON." By changing the state of one of them, say i, the consensus increases. In fact,

$$\begin{aligned}\Delta C(k) &= C(k : u_i \text{ off}) - C(k : u_i \text{ on}) \\ &= -\left(w_{ii} + \sum_{l=1, l\neq i}^{n} w_{il}\right) \geq -(w_{ii} + w_{ij}) \\ &= -(w_{ii} + [-\max\{w_{ii}, w_{jj}\} - \epsilon]) \geq \epsilon > 0\end{aligned}$$

b) *maximality:* If x^k is not a maximal IS, then there is a nonactivated unit i not adjacent with any activated unit. By changing the state of i the consensus increases since $\Delta C = w_{ii} > 0$. □

Theorem 2: The consensus function derived by the connection matrix IS is order preserving.

Proof: For any configuration of the BM inducing a feasible solution x^k, since nonadjacent units are both "ON," the consensus function is written as

$$\begin{aligned}C(k) &= \sum_{i=1}^{n} w_{ii} u_i^2 + \sum_{i=1, i<j}^{n} \sum_{j=1}^{n} w_{ij} u_i u_j \\ &= \sum_{i=1}^{n} w_{ii} u_i = \sum_{i=1}^{n} c_i u_i\end{aligned}$$

which is identical to the cost function $f(x) = \sum_{i=1}^{n} c_i x_i$. □

Remark 3: The parameter ϵ, which appears in the definition of the connection matrix of IS, validates Theorem 1, while it is also significant for the overall behavior of the network. When small values are attributed to it, a weak level of inhibition between the connected units is determined and so, more freedom is allowed to all units to pass through the firing state according to the sigmoid shape of the UR. Therefore, the capture of high consensus states is more likely to happen. On the contrary, when large values are allowed to the parameter ϵ, then strong inhibitions are imposed, and thus the possibilities for some units to fire, according to UR, are reduced. As a result of this, the network gets quickly stuck to a local maximum by satisfying constraints of problems rather than optimality of the objective function. A tradeoff between the two above cases would be to start with very small values and progressively, as the network has captured information about the objective function, allow for small increments to the values of ϵ.

The Set Packing Problem: As was already mentioned the network is constructed to be identical to the intersection graph, and units to represent the subsets. If c_i is the cost for the subset S_i, for any $i = 1, \cdots, n$ we consider the following connection matrix:

$$w_{ij} = c_i \quad \text{if } i = j$$
$$w_{ij} = -[\max\{c_i, c_j\} + \epsilon]\delta_{ij} \quad \text{if } i \neq j,$$

where $\delta_{i,j} = 1$ if $A_i^T A_j \geq 1$, and 0 otherwise. A_i and A_j are the ith and jth column vectors, respectively, of the $m \times n$ matrix: $A = (a_{ij})$ with $a_{ij} = 1$ if $q_i \in S_j$ and 0 if $q_i \notin S_j$, and ϵ a small positive parameter (remark 3). The solution vector is again equal to the vector state. The choice of such a set of weights is justified through the following considerations: 1) The SP is equivalent to IS through the intersection graph (Definition 2) with the initial costs associated with each vertex, i.e., a solution x is a feasible (optimal) solution for SP if and only if it is a feasible (optimal) solution for IS (Proposition 1), 2) the weights as it has been shown above are *well defined* since the two problems have the same set of optimal solutions.

The Set Partitioning Problem: The network is again isomorphic to the intersection graph, without labels and with units representing the subsets. We consider the following connection matrix:

$$\begin{aligned}w_{ij} &= c^T e|S_i| - c_i \quad \text{if } i = j, \\ w_{ij} &= -\left[\max\left\{c^T e|S_i| - c_i, c^T e|S_j| - c_j\right\} + \epsilon\right]\delta_{ij} \\ &\quad \text{if } i \neq j\end{aligned}$$

where $\delta_{i,j} = 1$ if $A_i^T A_j \geq 1$, and 0 otherwise and where c^T is the transposed cost vector, e is the n-unit column vector and A_i and A_j are the column vectors of the $m \times n$ matrix $A = (a_{ij})$ with $a_{ij} = 1$ if $q_i \in S_j$ and 0 if $q_i \notin S_j$, $|S_i|$ is the cardinality of subset S_i, and ϵ is a small real positive parameter (remark 3), The solution vector again equals the vector state of the BM. The basic motivation in choosing such a connection matrix is due to the equivalence of the (costed) SPP and the *costed* IS. We prove below that the SPP is equivalent to an SP with the same set of optimal solutions, if the first one is feasible. Of course, the resulting SP is equivalent to the IS problem. Let us consider two solution spaces Ω and Σ defined as follows:

$$\Omega = \{x \in \{0,1\}^n \text{ such that } Ax \leq e\},$$
$$\Sigma = \{x \in \{0,1\}^n \text{ such that } Ax = e\}$$

where e is the m-dimensional-unit column vector. Also we define $\Theta = \sum_{i=1}^{n} c_i$.

Lemma 1: If $x \in \Sigma$ and $y \in \Omega - \Sigma$ then $\Theta e^T Ax - c^T x \geq \Theta e^T Ay - c^T y$.

Proof: We have $\Theta\left[e^T Ax - e^T Ay\right] \geq \Theta$ since one component at least of $Ax - Ay$ is 1. Moreover, $c^T x - c^T y \leq c^T x \leq \Theta$. □

Theorem 3: Consider the $m \times n$ binary matrix A, $e^T = (1, \cdots, 1) \in R^m$ and assume that the set Σ is not empty. Then, the following problems are equivalent.

$$\mathrm{SPP} : \min\left\{c^T x\right\}, \quad x \in \sum,$$
$$\mathrm{SP} : \max\left\{\Theta e^T Ax - c^T x\right\}, \quad x \in \Omega,$$
$$\mathrm{SP}' : \min\{c^T x - \Theta e^T A + \Theta m\}, \quad x \in \sum.$$

Proof: From Lemma 1 it is obvious that the solutions of SP are in Σ. Moreover, by the definition of Σ we have obviously: $\Theta e^T Ax = \Theta m$. Thus, SP and SP′ are equivalent in the sense that the vectors of Σ that maximize the function $\Theta m - c^T x$ minimize also the function $c^T x$.

For the equivalence of SPP and SP′ we have simply to observe that because of the equality $\Theta e^T Ax = \Theta m$, SP′ becomes exactly $\min c^T x$, $x \in \Sigma$. □

Now, we proceed as in the case of SP. We construct the associated intersection graph (Definition 2) without labels and we consider the vertices to be costed by the new calculated values $\hat{c}_i = \Theta|S_i| - c_i$ where $|S_i| = e^T A_i$. Thus, if SPP is feasible, then the optimal solution of the IS is also the optimal solution of the SPP. The significance of the above connection matrix is highlighted by the theoretical results that follow. In fact from these results it can be deduced that the network's energy has some "good" properties, that is, the energy is always order preserving even if it is not always feasible. Let us consider two configurations k and l of the BM inducing the *feasible solutions* x^k and x^l for SPP (and obviously for IS) and let us denote by f and $\hat{f}$ the cost functions of the SPP and IS, respectively. Then the following results can be proven.

Lemma 2: If x^k and x^l are feasible solutions of the SPP then: $f(x^k) < f(x^l) \Rightarrow \hat{f}(x^k) > \hat{f}(x^l)$.

Proof: $f(x^k) < f(x^l) \Rightarrow c^T x^k < c^T x^l \Rightarrow \Theta m - c^T x^k > \Theta m - c^T x^l \Rightarrow \Theta e A x^k - c x^k > \Theta e A x^l - c x^l \Rightarrow \hat{c} x^k > \hat{c} x^l \Rightarrow \hat{f}(x^k) > \hat{f}(x^l)$. □

Proposition 2: If one of two feasible solutions of the IS, say x^k, is also feasible for SPP and the other one, x^l, infeasible, then $\hat{f}(x^k) \geq \hat{f}(x^l)$. This means that, the configurations which are feasible solutions of the SPP are of higher maximum independent cost value, with respect to any one which is not feasible.

Proof: Immediately from Lemma 1. □

Proposition 3: If x^k and x^l are feasible solutions for the IS, but infeasible solutions for the SPP and α and β denote the number of covered basic set's elements, by the above solutions, respectively, then: $\left[\left(f(x^k) < f(x^l)\right) \wedge (\alpha > \beta)\right] \Rightarrow \hat{f}(x^k) > \hat{f}(x^l)$.

Proof: $\left[\left(f(x^k) < f(x^l)\right) \wedge (\alpha > \beta)\right] \Rightarrow \left[\left(c^T x^k < c^T x^l\right) \wedge (\alpha > \beta)\right] \Rightarrow \Theta\alpha - c^T x^k > \Theta\beta - c^T x^l \Rightarrow \Theta e^T A x^k - c^T x^k > \Theta e^T A x^l - c^T x^l \Rightarrow \hat{c}^T x^k > \hat{c}^T x^l \Rightarrow \hat{f}(x^k) > \hat{f}(x^l)$. □

Thus, increasing maximum independent cost values, we improve SPP's partial solutions in a large sense. We can obtain either better cost values and fewer covered elements, or worst cost values and more covered elements, or better cost values and more covered elements.

Theorem 4: Let us consider two configurations k and l of the BM, inducing the two *feasible* solutions x^k and x^l of the SPP. Then $f(x^k) < f(x^l) \Rightarrow c(x^k) > c(x^l)$.

Proof: The proof is due to Lemma 2 and Theorem 2. □

Thus, the consensus function is *order preserving* for the SPP. From the above results we can conclude that the higher the local maxima of the consensus function, the higher the obtained maxima of IS and therefore the better the solutions of SPP. However, it is obvious, by Proposition 2, that only configurations with near-optimum consensus will provide feasible solutions of the SPP. Also, by Proposition 3 and Theorem 2, we conclude that the higher the local maxima of the consensus function, even when they do not represent feasible solutions of the SPP, the better the *partial solution* of the SPP in a large sense, that is either on cost improvement or on elements covering. Moreover, *if a local maximal of the consensus function provides a feasible solution for SPP, then any other higher local maximal will also provide a better feasible solution for this problem* (Proposition 2 and Theorem 4). Finally, if the BM is stuck in a (global) maximum, the best feasible solution of the SPP will be found, if this one is feasible. (Theorems 2 and 3).

The Minimum Vertex Cover: The network is constructed to be isomorphic to the initial graph. The connection matrix is the same as that of IS and the solution vector equals to the complement state vector (no firing units). This result is a consequence of the complementation of the two problems. However, as was mentioned in [9], a neural network approach for this problem by a model permitting a cooperation between linked units (for example the competitive activation method) would be more convenient. This happens for the reason that both linked units could be winners and so included in the final solution. The above remark becomes obvious on a triangle form graph where two winners are required. But for problems like the IS where only a winner is permitted among the two competitors, a model like BM with inhibitory connections is more suitable. For the simple example mentioned above one can easily see that the antagonistic competition of the units on a BM will give only one winner, and thus two winners for the VC.

The Set Cover Problem: We consider again the intersection graph but now with labels. We assume that each instance describing a SC does not include elements which belong only to one subset. If this was the case, the subset containing them should be included in the final solution. Thus, we should either exclude them from the instance or associate with them very small costs in order to favorize their appearance in the final solution. The necessary preprocessing is described as follows [8].

1) We construct the intersection multigraph G_S (Definition 2.)
2) For every $k = 1, \cdots, n$ we remove all but one label. If after the deletion there are edges without any label we remove those edges too. Let G'_S be the resulting graph.
3) Construct a BM by considering G'_S as the graph of a VC problem.
4) Assign cost $\hat{c}_i = c_i/|S_i|$ to each vertex.

The vector solution equals the complementary vector state. From the discussion made in Section II *there is a one to one correspondence between the labels of the edges and the edges themselves in G'_S. Thus, by searching for a set of vertices that are adjacent to all the edges of G'_S we find a set of vertices that also "sees" all the labels of G'_S and therefore we have found also a set of subsets that covers all the elements of Q.*

IV. Experiments and Results

Simulations have been performed for SPP on 50 instances, and for IS and VC on 224 instances. The control parameter T was initialized to 2.0 and the decrement was controlled by the parameter *rate* defined in Section III. A very small value of the parameter *rate* leads to a slow decrement of the parameter T and so to better results than a quicker cooling schema. The length of the generated Markovian chains for each temperature is avoided to be large in order to reduce the required computation time and was fixed to be equal to the number of the model units. The parameter ϵ included in the definition of the connection matrix

TABLE I
PERFORMANCE OF BM IN THE CASES OF IS AND VC

Initial Temperature		$T_0 = 2.0$				Heuristic
	Rate	10^{-2}	10^{-5}	$7\,10^{-6}$	10^{-6}	
	% of Opt. Sol. IS and VC	28.3	84.3	86.2	87.1	34.4
IS	Average ϵ (%)	9.9	1.5	1.4	1.3	8.4
	Maximum ϵ (%)	33.3	16.7	16.7	16.7	28.6
VC	Average ϵ (%)	7.6	0.94	0.90	0.73	6.4
	Maximum ϵ (%)	40	10	11.1	10	22.2

was chosen to be equal to 0.1. This leads to a weak inhibition between the connected units avoiding in this way quick convergence of the network to a nonoptimal configuration which satisfies the constraints. The results of these tests are significantly interesting particularly in the case of a slow decrement of the control parameter T. When dealing with IS and VC, feasible solutions have been always produced. The examined instances have been generated as follows. The cardinality of the vertex set V for the small instances were fixed to vary between 15 and 25. We have generated both *regular* (with degrees varying from 2 to 6) and *irregular graphs* (with maximum degree varying from 3 to 6). The edges for the irregular graphs were randomly obtained by uniformly choosing two distinct vertices (no loops) on V. The isolated vertices were removed. The costs associated with the vertices were all equal to one and the optimal solutions of the instances have been found by exhaustive search. The big instances have been produced by using small ones and by adding edges between the components in such a way that the cardinalities of the optimal solutions for the so obtained instances were equal to the sum of the cardinalities of the solutions for the small graphs composing those big instances and that big instances were not disconnected. When regular small graphs were combined to produce a big one the resulting graph was *quasi-regular* because the "articulation points" had their degrees increased by one. For the big graphs the number of vertices varied between 50 and 100 while the number of edges varied between 80 and 180. All the produced instances have been tested with different values of the control parameter *rate*. Feasible solutions have been always produced and 87.1% of the tested instances are always optimally solved when *rate*$= 10^{-6}$. A summary of the performance of our BM model is given in Table I. The overall effectiveness of the model can be characterized by comparing the percentage of the optimum solutions obtained, the average relative error, as well as the maximum error for different values of the control parameter *rate*. It is immediately concluded the crucial importance of this parameter in the whole effectiveness of the model. We mention that for the three smaller values of *rate* depicted in the table (slow cooling schema) the difference between the optimal solution and the one obtained by the model was never greater than 1 in the case of irregular graphs and than 3 in the case of regular ones. Moreover, for these values (of *rate*) the obtained results (by the model) were *always* (on all the treated instances) better than the ones obtained by the heuristic of [6]. In Table I also, by using the same performance measures, the performance of our model in the case of VC is depicted. We note here that the same instances as in the case of IS have been solved. To generate the small instances of SPP we have fixed the cardinality of the set Q as well as the cardinality of the family $\mathcal{S}$. The cardinality of each subset varies from 1 to 5. The elements of each subset are uniformly chosen on the set $[1, \cdots, |Q|]$. The produced instances included at least one feasible solution and their density, fraction of nonzero entries in the matrix describing the instance, is about 6–8%. The optimal solution has been obtained by exhaustive search. The big instances have been obtained similarly as above by using the smaller ones. In order to preserve the connectivity of the (equivalent) characteristic graph B (Definition 1) some subsets have been added that contained elements from more than one set Q of the small instances. The cardinalities of the optimal solutions of the resulting instances were again equal to the sum of the optimal solutions' cardinalities of the small instances composing those big instances. The costs associated with the subsets are taken equal to one. The sizes of the so produced instances ranged between 20 and 50 elements for the set Q and between 30 and 100 for the set (family) $\mathcal{S}$. The performance of the model in the case of SPP is summarized in Tables II, III, and IV. The whole behavior of our BM is examined for different values of *rate* (Table II). The percentage of the feasible solutions as well as the percentage of the solutions not satisfying the covering property is presented. One can remark that low values (slow decrements of the temperature) when attributed to *rate* give a good behavior of the model. For example, when *rate* is taken equal to 7×10^{-6} then 70% of the tested instances are solved with total satisfaction of the constraints, the percentage of the optimal solutions being 28%. Moreover, in only one instance over 50 examined ones, three elements left uncovered (while, even this solution satisfies the packing property). In general we note that small values of *rate* lead to a considerable reduction of the number of "nonfeasible" solutions. This "nonfeasibility" appears with respect to the covering property and even in this case the number of the uncovered elements is very small. We remark here that the "nonfeasible" solutions do not cover only between one to five elements for sets of cardinalities 20–50 (Table II). Furthermore, the percentage of the optimal solutions obtained increases when the temperature decreases slowly. In Table III the maximum and average relative errors are depicted, when 0–4 elements are permitted to be uncovered. In Table IV the average relative error as well as the average number of covered elements are presented. These measures are evaluated for the two smaller values of *rate* used in the experiments. The results presented in the above tables refer to a complete stabilization of the network. As we have already mentioned, for a given temperature the equation UR and the one expressing ΔC are evaluated n times, where n is the size of the problem. When the "slower" cooling schema was applied and for a problem of 100 variables then covergence was attained at about the 3000th value of temperature provided by this schema. As a more "practical" estimation of the convergence time, the problem admitting 100 variables needs about 3 h on a PC with a 386 processor.

V. CONCLUSIONS

The neural network model discussed here is based on a BM architecture, intended to solve approximately hard optimization problems such as the Covering and Packing Problems. For all the five problems

TABLE II
PERFORMANCE OF BM IN THE CASE OF SPP

Initial Temperature		$T_0 = 2.0$							$T_0 = 5.0$
Rate		10^{-6}	7×10^{-6}	9×10^{-6}	10^{-5}	5×10^{-5}	10^{-4}	10^{-3}	10^{-3}
Feasible Solutions (%)		68	70	56	54	40	28	14	10
Valid Partial Solutions	1*	20	20	20	26	24	30	22	24
	2*	6	8	16	14	28	34	34	30
	3*	2	2	4	2	4	6	20	28
	≥4*	4	0	4	4	4	2	10	8
% Optimal Solution		28	28	30	28	22	10	2	2

*Number of uncovered elements

TABLE III
RELATIVE ERROR FOR SPP

Initial Temperature		$T_0 = 2.0$			
Rate		10^{-6}		7×10^{-6}	
		Average Error %	Maximum Error %	Average Error %	Maximum Error %
Elements not Covered	0	8.6	44	9	44
	≤ 1	8.7	44	7.6	44
	≤ 2	8.5	44	8	44
	≤ 3	8.7	44	7.9	44
	≤ 4	9	44	*	*

*No solution was provided with four or more uncovered elements

TABLE IV
FEASIBILITY RATIO AND AVERAGE RELATIVE ERROR FOR SPP

Initial Temperature	$T_0 = 2.0$	
Rate	10^{-6}	7×10^{-6}
Average Error (%)	9.4	7.9
Average Covered Elements (%)	98.4	98.7

examined here the global energy is order preserving, that is when the network is stabilized, the higher the local maximal, the better the obtained solution. Moreover, for all but SPP cases, the energy is feasible, that is a feasible solution is always found, when the network reaches stability. In the case of SPP, this result is not always true. However, if the maximum consensus is reached, then the optimum solution for SPP is obtained. Moreover, the higher the local maxima the higher the probabilities to obtain feasible solutions with reduced cost. Generally, we obtain partial valid solutions (mutual exclusion constraints satisfied) with few elements not covered. Concerning the complexity of the model we point out that the necessary connections are considerably reduced in the cases of SPP, SP, and SC, permitting us to deal with large size problems involved in real optimization applications.

The obtained results appear very promising, and one can deduce that near-optimal solutions are provided by the model, when we deal with instances of moderate sizes.

The obtained approximation ratio (fraction *cardinality of the obtained solution/cardinality of the optimal one* even for the maximation problems) for IS, SP, and Clique is always more than 0.833, (for $rate = 10^{-6}$). For VC the experimentally attained ratio (when $rate = 10^{-6}$) is equal to 1.1. Finally for SPP the error is less than 1.44 (measured on the instances where the obtained solution is feasible).

REFERENCES

[1] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines, A Stochastic Approach to Combinational Optimization and Neural Computing.* New York: Wiley, 1989.

[2] D. Ackley, G. Hinton, and T. Segnowski, "A learning algorithm for Boltzmann machines," *Cognitive Sci.,* vol. 9, pp. 147–169, 1985.

[3] A. Bertoni, P. Campadelli, and F. Grassani, "Full parallelism in Boltzmann machines," in *Proc. Neuro-Nimes,* pp. 361–370, Nimes, Nov. 1989.

[4] R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Comleteness.* San Francisco, CA: Freeman, 1979.

[5] J. J. Hopfield and D. W. Tank, "Neural computation of decision in optimization problems," *Biol. Cybern.,* vol. 52, pp. 141–152, 1985.

[6] D. S. Johnson, "Approximation algorithms for combinatorial problems," *J. Comput. Syst. Sci.,* vol. 9, pp. 256–278, 1974.

[7] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science,* vol. 220, pp. 671–689, 1983.

[8] V. Th. Paschos, "Computational problems of query optimization in database theory," Ph.D. dissertation, Ecole Nationale Polytéchnique d'Athènes, July 1988.

[9] Y. Peng, "A connectionist approach to vertex cover problems," IJCNN, Washington, June 1989.

[10] S. Poljak and M. Sura, "On periodical behaviour in societies with symmetric influences," *Combinatorica,* vol. 3, no. 1, pp. 119–121, 1983.

[11] A. Stafylopatis and V. Zissimopoulos, "On discrete nonlinear associative memories," in *Proc. IV ISCIS,* Çesme, Oct. 1989, pp. 625–634.