

On the Computational Complexity of Analyzing Hopfield Nets*

Patrik Floréen
Pekka Orponen[†]
Department of Computer Science
University of Helsinki
Finland

Abstract

We prove that the problem of counting the number of stable states in a given Hopfield net is #P-complete, and the problem of computing the size of the attraction domain of a given stable state is NP-hard.

1 Introduction

A binary associative memory network, or “Hopfield net” [6], consists of n fully interconnected threshold logic units, or “neurons”. Associated to each pair of neurons i, j is an interconnection weight w_{ij} , and to each neuron i a threshold value t_i . At any given moment a neuron i can be in one of two states, $x_i = 1$ or $x_i = -1$. Its state at the next moment depends on the current states of the other neurons and the interconnection weights; if $\text{sgn}(\sum_{j=1}^n w_{ij}x_j - t_i) \neq x_i$, the neuron may switch to the opposite state. (Here sgn is the signum function, $\text{sgn}(x) = 1$ for $x \geq 0$, and $\text{sgn}(x) = -1$ for $x < 0$.) Whether the state change actually occurs depends on whether the neuron is selected for updating at this moment. In the synchronous update rule, all neurons are updated at each step; in an asynchronous rule only one neuron at a time is selected for updating.

Let us denote the state vector of a network by $x = (x_1, \dots, x_n)$, its matrix of connection weights by $W = (w_{ij})$, and its vector of thresholds by $t = (t_1, \dots, t_n)$. If W is symmetric with a zero diagonal, and an asynchronous update rule is

*This work has been supported by the Academy of Finland. A preliminary version appears under the title “Counting stable states and sizes of attraction domains in Hopfield nets is hard” in *Proceedings of the International Joint Conference on Neural Networks 1989* (Washington, D.C., June).

[†]Part of this research was carried out while the second author was visiting the Department of Computer Science, University of Toronto.

used, it can be shown [6] that from any initial state, the network eventually converges to a stable state, i.e. to a state x such that $\text{sgn}(Wx - t) = x$. This convergence property makes the use of such networks as associative memories an attractive possibility. Let $u^1, \dots, u^m \in \{-1, 1\}^n$ be a collection of patterns (bit vectors) to be stored. Assume that a connection matrix W and a threshold vector t can be determined so that the state vectors corresponding to u^1, \dots, u^m are stable. Given then as input a pattern which is a distorted version of some u^i , the correct pattern can (hopefully) be recovered by setting the initial states of the neurons corresponding to the input, and letting the network repeat its state-change operation until it stabilizes.

A well-behaved associative memory network should have at least the following properties: the stored patterns should really be stable states of the system; there should be as few spurious stable states as possible; and more generally there should be some reasonably large domain of attraction around each desired stable state, i.e. given an input that differs only slightly from one of the stored patterns, the result of the stabilizing computation should be that pattern.

The fundamental properties and capabilities of Hopfield-type associative memories have been analyzed extensively in the literature (e.g., [1, 2, 5, 6, 7, 8, 11, 12]). Our main interest here is in the situation where some ready-made associative memory network is given, and its quality is to be assessed. Problems of interest in this setting include: given a network, compute its number of stable states, and given a stable state in a network, compute the size of its domain of attraction. As it turns out, both of these tasks are at least *NP*-hard.

We concentrate on the restricted class of networks with symmetric connection matrices with zero diagonals, i.e., the class of networks originally considered by Hopfield in [6]. Asymmetric networks are not considered, because they can simulate finite state automata, and consequently be quite ill-behaved. Moreover, analyzing even the basic properties of general networks is hard. For instance, Porat [9] has shown that the problem of determining whether every computation in an asymmetric network ends up in a stable state is *NP*-hard, and Lipscomb et al. [4] have shown that even the problem of determining whether such a network has *any* stable states is *NP*-hard. The diagonal elements are usually required to be zero in order to guarantee convergence and to exclude trivial connection matrices resembling the identity matrix.

2 Determining the Number of Stable States

Given a network (W, t) , it is easy to check that a set of desired patterns $u^i, i = 1, \dots, m$, are all stable: just compute $\text{sgn}(Wu^i - t)$ for each one. Hence, computing the number of spurious stable states in a network is equivalent to computing how many stable states it has altogether. Unfortunately, this is hard. We consider first the case of symmetric connection matrices with nonzero diagonal elements allowed.

Theorem 1 *The problem “Given a symmetric integer valued matrix W ; does the mapping $x \mapsto \text{sgn}(Wx)$ have fixed points?” is NP-complete.*

Proof. Clearly, the problem is in NP. We prove that it is NP-hard by a reduction from the PARTITION problem [3, p. 223].

Consider an instance of PARTITION: “Given positive integers a_1, \dots, a_k ; does there exist a set of indices $A \subseteq \{1, \dots, k\}$, such that $\sum_{i \in A} a_i = \sum_{i \notin A} a_i$?” To reduce this to an instance of the problem under consideration, denote $c = \sum_{i=1}^k a_i$, and choose

$$W = \begin{pmatrix} -c & c & a_1 & a_2 & \cdots & a_k \\ c & -c & -a_1 & -a_2 & \cdots & -a_k \\ a_1 & -a_1 & 1 & 0 & \cdots & 0 \\ a_2 & -a_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_k & -a_k & 0 & 0 & \cdots & 1 \end{pmatrix}$$

Clearly W can be constructed from a_1, \dots, a_k in polynomial time.

We now prove that there is a solution to the given PARTITION instance if and only if the mapping $x \mapsto \text{sgn}(Wx)$ has a fixed point. For brevity, let us denote $(a_1, \dots, a_k) = a$.

(if) If there is a fixed point of the form $(1, 1, \xi)$, $\xi = (\xi_1, \dots, \xi_k) \in \{-1, 1\}^k$, then by row 1 of matrix W , $a^T \xi \geq 0$, and by row 2 of W , $-a^T \xi \geq 0$. Hence $a^T \xi = 0$, and $\sum_{i \in A} a_i = \sum_{i \notin A} a_i$ for $A = \{i \in \{1, \dots, k\} \mid \xi_i = 1\}$.

Thus, let us prove that the only possible fixed points are of this form:

1. Form $(-1, -1, \xi)$ is not possible, because then rows 1 and 2 of W would imply that $0 > a^T \xi > 0$.
2. Form $(1, -1, \xi)$ is not possible, because then row 1 of W would imply that $-2c + a^T \xi \geq 0$. However, $a^T \xi \leq c$, so $-2c + a^T \xi \leq -c < 0$.
3. Form $(-1, 1, \xi)$ is not possible, because then row 1 of W would imply that $2c + a^T \xi < 0$. However, $a^T \xi \geq -c$, so $2c + a^T \xi \geq c > 0$.

(only if) If A is a solution to the PARTITION instance, then $\sum_{i=1}^k a_i \xi_i = 0$, where $\xi_i = 1$ if $i \in A$, and $\xi_i = -1$ if $i \notin A$. Thus $x = (1, 1, \xi_1, \dots, \xi_k)$ is a fixed point of the mapping $x \mapsto \text{sgn}(Wx)$. \square

We can actually make a slightly stronger statement than in Theorem 1, but for this we need a few additional notions. An integer valued function f belongs to the class #P if there is a nondeterministic polynomial time Turing machine M that on each input x has exactly $f(x)$ accepting computation paths. A function f is #P-complete, if it is in #P, and any other function in #P can be computed by some deterministic polynomial time Turing machine that is allowed to access values of f at unit cost. To each NP decision problem

there corresponds in a natural way a $\#P$ counting problem (i.e., the problem of counting “witnesses” or accepting computation paths), and vice versa. We shall say, somewhat inaccurately, that an NP decision problem A is in $\#P$ (resp. $\#P$ -complete) if the associated counting problem is in $\#P$ (resp. $\#P$ -complete). A polynomial time transformation of one NP problem to another is *parsimonious* if it preserves the number of witnesses to each instance. If the underlying NP problem A of a $\#P$ -complete counting problem can be parsimoniously reduced to another NP problem B , then also the counting problem associated with B is $\#P$ -complete. (For a more extended discussion of these notions, see [3, pp. 168–169].)

Corollary 1 *The problem “Given a symmetric integer valued matrix W ; how many fixed points does the mapping $x \mapsto \text{sgn}(Wx)$ have?” is $\#P$ -complete.*

Proof. The problem is clearly in $\#P$. Since PARTITION is $\#P$ -complete [10], and the reduction in the proof of Theorem 1 is parsimonious, the claim follows. \square

As the following simple proposition illustrates, the two negative diagonal elements in the proof of Theorem 1 are crucial for the (non)existence of stable states.

Proposition 1 *Given a symmetric real-valued matrix W with non-negative diagonal elements, and a real valued threshold vector t , the mapping $x \mapsto \text{sgn}(Wx - t)$ always has a fixed point.*

Proof. As shown by Hopfield in [6], if the neurons in a zero-diagonal network are updated asynchronously, the network always stabilizes to a stable state from any initial state. This is actually true of all networks with non-negative diagonal elements [5]. Since the property of being a stable state is not dependent on the way neurons are updated, a network of this type always has at least one stable state. \square

However, even in this case it is hard to determine the *number* of stable states, as the following result shows.

Theorem 2 *The problem “Given a symmetric integer valued matrix W with zero diagonal elements; does the mapping $x \mapsto \text{sgn}(Wx)$ have at least three fixed points?” is NP -complete.*

Note. This result, except for the zero thresholds, and for two instead of three fixed points, has also been obtained independently by Lipscomb et al. [4].

Proof. Clearly, the problem is in NP . We prove that it is NP -hard by a reduction from a restricted version of PARTITION, where the two sides of the eventual partition are required to contain equally many elements. This problem, which we call EQUIPARTITION, is also NP -complete [3, p. 223].

Thus, consider an instance of EQUIPARTITION: “Given positive integers a_1, \dots, a_k , k even; does there exist a set of indices $A \subseteq \{1, \dots, k\}$, such that $|A| = k/2$ and $\sum_{i \in A} a_i = \sum_{i \notin A} a_i$?”

Denote $c = \sum_{i=1}^k a_i$, and choose

$$W = \begin{pmatrix} 0 & 0 & c + a_1 & c + a_2 & \cdots & c + a_k \\ 0 & 0 & -c - a_1 & -c - a_2 & \cdots & -c - a_k \\ c + a_1 & -c - a_1 & 0 & -1 & \cdots & -1 \\ c + a_2 & -c - a_2 & -1 & 0 & \cdots & -1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c + a_k & -c - a_k & -1 & -1 & \cdots & 0 \end{pmatrix}$$

Clearly W can be constructed from a_1, \dots, a_k in polynomial time. Note also that $c \geq k$, and that the vectors $(1, -1, 1, 1, \dots, 1)$ and $(-1, 1, -1, -1, \dots, -1)$ are always fixed points of the mapping $x \mapsto \text{sgn}(Wx)$. We now prove that the given EQUIPARTITION instance has a solution if and only if this mapping has at least one more fixed point.

(*if*) Assume that the mapping has a fixed point of the form $(1, 1, \xi)$ for some $\xi = (\xi_1, \dots, \xi_k) \in \{-1, 1\}^k$. Let us again denote $(a_1, \dots, a_k) = a$, and for brevity also $(1, 1, \dots, 1) = 1$, so that $a^T \xi = \sum_{i=1}^k a_i \xi_i$ and $1^T \xi = \sum_{i=1}^k \xi_i$. By rows 1 and 2 of matrix W , $c1^T \xi + a^T \xi = 0$. Now if $1^T \xi \neq 0$, then $|1^T \xi| \geq 2$ (recall that k is even), and as $|a^T \xi| \leq c$, $|c1^T \xi + a^T \xi| \geq c > 0$. Thus $1^T \xi = 0$, and so also $a^T \xi = 0$. Hence, if we choose $A = \{i \in \{1, \dots, k\} | \xi_i = 1\}$, we have $|A| = k/2$ and $\sum_{i \in A} a_i = \sum_{i \notin A} a_i$.

Let us then prove that the only possible fixed points besides $(1, -1, 1, \dots, 1)$ and $(-1, 1, -1, \dots, -1)$ are indeed of the form $(1, 1, \xi)$.

1. A fixed point cannot be of the form $(-1, -1, \xi)$, since rows 1 and 2 of W would then require that $0 > c1^T \xi + a^T \xi > 0$.
2. If $(1, -1, \xi)$ is a fixed point, then for every $i = 1, \dots, k$,

$$(Wx)_{i+2} = 2c + 2a_i - (1^T \xi - \xi_i) \geq 2k + 2 - (k - 1) > 0,$$

so $(1, -1, 1, \dots, 1)$ is the only possible fixed point of this form.

3. If $(-1, 1, \xi)$ is a fixed point, then for every $i = 1, \dots, k$,

$$(Wx)_{i+2} = -2c - 2a_i - (1^T \xi - \xi_i) \leq -2k - 2 + (k - 1) < 0,$$

so $(-1, 1, -1, \dots, -1)$ is the only possible fixed point of this form.

(*only if*) If $A \subseteq \{1, \dots, k\}$, $|A| = k/2$, is a solution to the given EQUIPARTITION instance, then defining $\xi_i = 1$ if $i \in A$, and $\xi_i = -1$ if $i \notin A$ yields $\sum_{i=1}^k a_i \xi_i = 0$ and $\sum_{i=1}^k \xi_i = 0$. Setting $x = (1, 1, \xi)$, it is easy to check that

$(Wx)_1 = (Wx)_2 = 0$, and $(Wx)_{i+2} = \xi_i$, for $i = 1, \dots, k$. Thus $x = (1, 1, \xi)$ is a fixed point of the mapping $x \mapsto \text{sgn}(Wx)$. \square

We can also easily strengthen Theorem 2 to a $\#P$ -completeness result after we prove the requisite lemma:

Lemma 1 *The EQUIPARTITION problem is $\#P$ -complete.*

Proof. The problem is clearly in $\#P$. Consider the following transformation from PARTITION, which is $\#P$ -complete: given a PARTITION instance with elements a_1, \dots, a_n , denote $c = \sum_{i=1}^n a_i$, and construct an EQUIPARTITION instance that consists of the elements

$$a_1 + 2c, a_2 + 4c, \dots, a_n + 2^n c, 2c, 4c, \dots, 2^n c.$$

Let us prove that this is in fact a parsimonious reduction, i.e., that to every PARTITION solution there corresponds exactly one EQUIPARTITION solution, and vice versa.

If $A \subseteq \{1, \dots, n\}$ is a solution to the PARTITION instance, so that $\sum_{i \in A} a_i = \sum_{i \notin A} a_i$, then clearly

$$\sum_{i \in A} (a_i + 2^i c) + \sum_{i \notin A} 2^i c = \sum_{i \notin A} (a_i + 2^i c) + \sum_{i \in A} 2^i c,$$

and both sides of the equality have n terms. Moreover, if $B \subseteq \{1, \dots, n\}$ is such that

$$\sum_{i \in A} (a_i + 2^i c) + \sum_{j \in B} 2^j c = \sum_{i \notin A} (a_i + 2^i c) + \sum_{j \notin B} 2^j c,$$

and $|A| + |B| = n$, then both sides of the equality must be equal to $\frac{c}{2} + (2^{n+1} - 2)c$, and so $\sum_{i \in A} 2^i + \sum_{j \in B} 2^j = 2^{n+1} - 2$. But there is only one way to obtain $2^{n+1} - 2$ as a sum of exactly n positive powers of 2; viz., to have $B = \{1, \dots, n\} - A$.

Conversely, let $A, B \subseteq \{1, \dots, n\}$ be a solution to the EQUIPARTITION instance, so that

$$\sum_{i \in A} (a_i + 2^i c) + \sum_{j \in B} 2^j c = \sum_{i \notin A} (a_i + 2^i c) + \sum_{j \notin B} 2^j c,$$

and $|A| + |B| = n$. Since both sides of the equality must equal $\frac{c}{2} + (2^{n+1} - 2)c$, and since $2^k c \geq 2c > \sum_{i \in A} a_i, \sum_{i \notin A} a_i$ for all $k \geq 1$, it must be the case that $\sum_{i \in A} a_i = \sum_{i \notin A} a_i = \frac{c}{2}$. \square

Corollary 2 *The problem “Given a symmetric integer valued matrix W with zero diagonal elements; how many fixed points does the mapping $x \mapsto \text{sgn}(Wx)$ have?” is $\#P$ -complete.*

Proof. The reduction in Theorem 2 provides a parsimonious transformation from EQUIPARTITION to the slightly modified version “Given a symmetric integer valued matrix W with zero diagonal elements; how many fixed points *beyond two* does the mapping $x \mapsto \text{sgn}(Wx)$ have?”. Hence EQUIPARTITION, and any counting problem in $\#P$, can be solved in polynomial time, assuming an ability to solve instances of the original problem in a single step. \square

3 Determining Sizes of Attraction Domains

In this section we consider the problem of computing the size of the attraction domain of a given state, i.e. the number of patterns that converge to this state. We first give a simple result showing that in general it is *NP*-complete to determine whether there are *any* patterns that map to a given state.

Theorem 3 *The problem “Given a symmetric integer valued $n \times n$ -matrix W with zeros as diagonal elements; does there exist a point $x \in \{-1, 1\}^n$ such that $\text{sgn}(Wx) = (1, \dots, 1)$?” is *NP*-complete.*

Proof. The problem is obviously in *NP*. Now, given a PARTITION instance with elements a_1, \dots, a_k , consider

$$W = \begin{pmatrix} 0 & 0 & a_1 & \cdots & a_k \\ 0 & 0 & -a_1 & \cdots & -a_k \\ a_1 & -a_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_k & -a_k & 0 & \cdots & 0 \end{pmatrix}$$

It can be verified that the given instance of PARTITION has a solution if and only if there is an $x \in \{-1, 1\}^{k+2}$ such that $\text{sgn}(Wx) = (1, \dots, 1)$. \square

Of course, if we know that $y \in \{-1, 1\}^n$ is a fixed point, there is trivially at least one point $x \in \{-1, 1\}^n$ such that $\text{sgn}(Wx) = y$, namely y itself. However, our next theorem shows that even in this case asking for another pattern in the attraction domain is *NP*-complete. We first formulate a lemma, of certain interest in itself, showing that a certain very restricted form of 0-1 integer programming is *NP*-complete.

Lemma 2 *The problem “Given a symmetric zero-diagonal $n \times n$ -matrix A with entries from $\{-1, 0, 1\}$; does there exist a point $x \in \{0, 1\}^n$, $x \neq 0$, such that $Ax \geq 0$?” is *NP*-complete.*

Proof. The problem is clearly in *NP*. We prove completeness in two stages: we first reduce the *NP*-complete MINIMUM COVER problem [3, p. 222] to the present problem without requiring that the resulting matrix be symmetric or zero-diagonal, and then we show how to impose these two conditions.

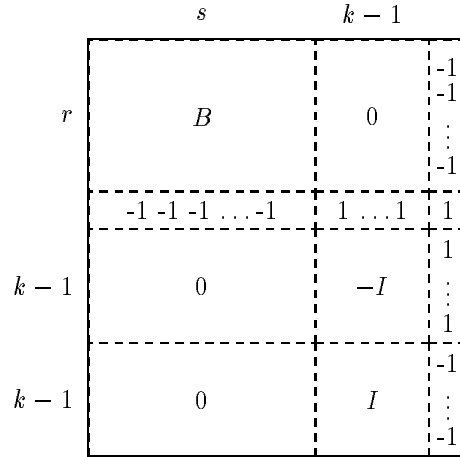


Figure 1: Matrix C .

Consider an instance of MINIMUM COVER: “Given a matrix $B \in \{0, 1\}^{r \times s}$ and a constant k ; is it possible to select from B a set of $q \leq k$ column vectors y_1, \dots, y_q so that $\sum_{i=1}^q y_i \geq 1$?” (Here again we use the notation $1 = (1, \dots, 1)$.)

For the reduction, construct first from B and k an intermediate matrix C as in Figure 1. We claim that B has a set of $q \leq k$ columns y_1, \dots, y_q such that $\sum_{i=1}^q y_i \geq 1$ if and only if C has a set of columns $x_1, \dots, x_p, p \geq 1$, such that $\sum_{i=1}^p x_i \geq 0$. The “only if” direction is clear: given a set of columns from B , choose from C the corresponding columns plus the k rightmost columns. For the “if” direction, note first that whatever columns are chosen from C , the k rightmost columns, in particular the utter rightmost one, must be among these. This implies that for the columns of B obtained as projections of the choices from C , each of the row sums must be at least 1. Also, there cannot be more than k B -columns, because otherwise it would not be possible to get a nonnegative sum on row $r + 1$ of C .

Having constructed the matrix C as above, a symmetric zero-diagonal matrix A can then be constructed as in Figure 2. Matrix A satisfies our requirements in the same way as C does, because in fact none of the columns to the right of column $s + k$ of A can ever be included in a set of columns with nonnegative row sums. \square

Theorem 4 *The problem “Given a symmetric zero-diagonal $n \times n$ -matrix W with entries from $\{-1, 0, 1\}$ such that $\text{sgn}(W1) = 1$; do there exist any points $x \in \{-1, 1\}^n, x \neq 1$, such that $\text{sgn}(Wx) = 1$?” is NP-complete.*

	$s + k$	$r + 2k - 1$		
$s + k$	0	C^T	0	0
			0	0
			⋮	⋮
			0	0
$r + 2k - 1$	C	0	-1	-1
			-1	-1
			-1	-1
			⋮	⋮
			-1	-1
	0 0 ... 0	-1 -1 -1 ... -1	0	-1
	0 0 ... 0	-1 -1 -1 ... -1	-1	0

Figure 2: Final matrix A .

Proof. The problem is clearly in NP . We prove that it is NP -hard by a reduction from the problem in Lemma 2. Consider an instance of this: “Given a symmetric zero-diagonal matrix $A \in \{-1, 0, 1\}^{k \times k}$, does there exist a point $x \in \{0, 1\}^k$, $x \neq 0$ such that $Ax \geq 0$?”

Let K_1 be the symmetric $k \times k$ -matrix whose entry k_{ij} is 1 if $j = i \pm 1 \pmod{k}$, and 0 otherwise, and let K_2 be the symmetric $k \times k$ -matrix whose entry k_{ij} is 1 if $j = i \pm 1$ or $j = i \pm 2 \pmod{k}$, and 0 otherwise. Let I denote the $k \times k$ matrix all of whose entries are 1. Choose $n = 5k$, and let

$$W = \begin{pmatrix} -A & A & 0 & 0 & 0 \\ A^T & 0 & 1 & K_1 & 0 \\ 0 & 1 & -1 + I + K_1 & -K_2 & I \\ 0 & K_1 & -K_2 & K_1 & 0 \\ 0 & 0 & I & 0 & 0 \end{pmatrix}$$

It is easy to verify that $\text{sgn}(W1) = 1$. We now show that there exists a $y \in \{0, 1\}^k$, $y \neq 0$, such that $Ay \geq 0$ in the problem in the Lemma if and only if there exists an $x \in \{-1, 1\}^n$, $x \neq 1$, such that $Wx \geq 0$ in the present problem.

Denote $x = (x^{(1)}, \dots, x^{(5)})$, where $x^{(i)} \in \{-1, 1\}^k$, $i = 1, \dots, 5$.

(if) If $Wx \geq 0$, and $x \neq 1$, then $x^{(3)} = 1$ by rows $4k + 1, \dots, 5k$ of W . Thus $x^{(2)} = x^{(4)} = 1$ by rows $3k + 1, \dots, 4k$, and then $x^{(5)} = 1$ by rows $2k + 1, \dots, 3k$. Hence $x^{(1)} \neq 1$, and by rows $1, \dots, k$, $-Ax^{(1)} + A1 \geq 0$. But this implies that $Ay \geq 0$ for $y = \frac{1}{2}(1 - x^{(1)}) \neq 0$.

(only if) If $Ay \geq 0$, and $y \neq 0$, then take $x = (x^{(1)}, x^{(2)}, \dots, x^{(5)}) = (1 - 2y, 1, \dots, 1)$. Now $Wx \geq 0$, and $x \neq 1$. \square

Corollary 3 *The problem “Given a symmetric zero-diagonal $n \times n$ -matrix W with entries from $\{-1, 0, 1\}$ such that $\text{sgn}(W1) = 1$; how many points $x \in \{-1, 1\}^n$ are there such that $\text{sgn}(Wx) = 1$?” is $\#P$ -complete.*

Proof. The MINIMUM COVER problem is $\#P$ -complete (to verify this, see the sequence of parsimonious reductions in [3, pp. 48–53, 64] that lead from SATISFIABILITY to MINIMUM COVER). Furthermore, the reductions given in Lemma 2 and Theorem 4 are parsimonious. \square

4 Conclusion

We have shown that given a Hopfield net, it is NP -hard to either count the total number of stable states in it, or to count the number of states converging to a given stable state. The latter result holds even when the interconnection weights between neurons are restricted to 0 and ± 1 . More strongly, we have shown that the stable state counting problem, and the problem of counting states converging to a given stable state in a *single parallel update step* are both $\#P$ -complete. A remaining open problem is to show that computing the size of the full attraction domain of a given stable state is also $\#P$ -complete. Also, an important issue we have not addressed at all is computing the attraction *radius* of a given stable state, i.e., the maximal Hamming distance from within which all patterns converge to this state.

References

- [1] A. Dembo, “On the capacity of associative memories with linear threshold functions.” *IEEE Trans. Information Theory* 35 (1989), 709–720.
- [2] F. Fogelman, E. Goles, and G. Weisbuch, “Transient length in sequential iterations of threshold functions.” *Discr. Appl. Math.* 6 (1983), 95–98.
- [3] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman & Co., New York, 1979.
- [4] G. H. Godbeer, J. Lipscomb, and M. Luby, “On the Computational Complexity of Finding Stable State Vectors in Connectionist Models (Hopfield Nets).” Technical Report 208/88, Dept. of Computer Science, Univ. of Toronto, 1988.
- [5] E. Goles, “Lyapunov functions associated to automata networks.” In *Automata Networks in Computer Science — Theory and Applications*, eds.

- F. Fogelman Soulié, Y. Robert, and M. Tchuente. Manchester University Press, Great Britain, 1987. Pp. 58–81.
- [6] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities.” *Proc. Nat. Acad. Sci. USA* 79 (1982), 2554–2558.
 - [7] A. Kuh and B. W. Dickinson, “Information capacity of associative memories.” *IEEE Trans. Information Theory* 35 (1989), 59–68.
 - [8] R. J. McEliece, E. C. Posner, E. R. Rodemich, and S. S. Venkatesh, “The capacity of the Hopfield associative memory.” *IEEE Trans. Information Theory* 33 (1987), 461–482.
 - [9] S. Porat, “Stability and looping in connectionist models with asymmetric weights.” *Biol. Cybern.* 60 (1989), 335–344.
 - [10] J. Simon, “On the difference between the one and the many (preliminary version).” In *Proc. 4th International Colloquium on Automata, Languages and Programming*. Lecture Notes in Computer Science 52, Springer-Verlag, Berlin, 1977. Pp. 480–491.
 - [11] H. J. Sussmann, “On the number of memories that can be perfectly stored in a neural net with Hebb weights.” *IEEE Trans. Information Theory* 35 (1989), 174–178.
 - [12] S. S. Venkatesh, “Epsilon capacity in neural networks.” In *Neural Networks for Computing*, ed. J. Denker. AIP Conference Proceedings Vol. 151 (1986). Pp. 440–445.