② 9-14

超立方体的 3-独立集及其在神经联想存储器中的应用*

TP333 O157.5

3-Independent Sets of Hypercubes and Applications in Neural Associative Memories

杨晓帆 何中市
Yang Xiaofan He Zhongshi
(重庆大学计算机研究所，重庆大学系统工程及应用数学系，重庆，630044)
陈廷槐
Chen Tinghuai
(重庆大学计算机研究所，重庆，630044)

摘 要 用 $I_3(n)$ 表示 n-立方体 Q_n 的 3-独立数。提出了构造 Q_n 的 3-独立集的一个算法，证明了 $2^{n-[\log_2 n]-1} \leqslant I_3(n) \leqslant [2^n/(n+1)]$. 这些结果被应用于神经联想存储器的设计.

关键词 图论算法；神经网络；联想存储器 / 超立方体；3-独立集；3-独立数
中国图书资料分类法分类号 O157.5; TP333

ABSTRACT Let $I_3(n)$ denote the 3-independence number of n-cube Q_n. An algorithm is presented for finding a 3-independent set of Q_n, and $2^{n-[\log_2 n]-1} \leqslant I_3(n) \leqslant [2^n/(n+1)]$ is shown. These results are applied to the design of neural associative memories.

KEYWORDS gragh-theoretical algorithm; neural network; associative memory / hypercube; 3-independent set; 3-independence number

0 引 言

A 3-independent set of a graph is a set of vertices such that the length of a shortest path between any two of them is at least 3. A maximum 3-independent set S of a graph is a 3-independent set such that every 3-independent set of the graph has at most $|S|$ vertices. $I_3(G)$ denotes the cardinality of a maximum 3-independent set of a graph G, called the 3-independence number of G.

n-cube, Q_n, is a graph whose vertices can be labeled with all 0-1 sequences of length n so that two vertices are adjacent iff their labels have a Hamming distance 1 (the Hamming distance between

* 收文日期 1994-04-23
This work was supported by the Doctor-Station Foundation of Chinese National Educational Committee.

two 0-1 sequences α and β is the number of bits on which α differs from β, denoted by $d_H(\alpha,\beta)$). For brevity, the vertices of Q_n are identified with their labels. Thus, a set of vertices of Q_n is a 3-independent set iff any two of them have a Hamming distance at least 3. $I_3(Q_n)$ is abbreviated as $I_3(n)$.

It is NP-hard to find $I_3(G)$ for an arbitrary graph G[2]. In this paper, and algorithm is presented for finding a 3-independent set of Q_n, and $2^{n-[\log_2 n]-1} \leqslant I_3(n) \leqslant [2^n/(n+1)]$ is shown ($[x]$ denotes the largest integer no larger than x). These results are applied to the design of neural associative memories.

1 BOUNDS ON $I_3(n)$

Let $F_1(0)=\{0000,1111\}, F_2(0)=\{0011,1100\}, F_3(0)=\{0101,1010\}, F_4(0)=\{0110,1001\}$,

$F_1(1)=\{0001,1110\}, F_2(1)=\{0010,1101\}, F_3(1)=\{0100,1011\}, F_4(1)=\{0111,1000\}$;

$T(1)=\{000,111\}, T(2)=\{001,110\}, T(3)=\{010,101\}, T(4)=\{011,100\}$.

Lemma 1. Let $0\leqslant i\leqslant 1$ and $1\leqslant j\neq k\leqslant 4$. Then the following assertions hold.

(1) If $J_j(i)=\{\alpha,\beta\}$, then $d_H(\alpha,\beta)=4$.

(2) If $T(j)=(\alpha,\beta)$, then $d_H(\alpha,\beta)=3$.

(3) If $\alpha\in F_j(i), \beta\in F_k(i)$, then $d_H(\alpha,\beta)=2$.

(4) If $\alpha\in T(j), \beta\in T(k)$, then $d_H(\alpha,\beta)\geqslant 1$.

Let $[\alpha]_{i-j}$ be the 0-1 sequence consisting of bit i through bit j of a 0-1 sequence α. $[\alpha]_{i-i}$ is abbreviated as $[\alpha]_i$. Let B^n denote the set of all 0-1 sequences of length n. We construct a set $H(\gamma)$ of 0-1 sequences of length $4n+3$ from a 0-1 sequence γ of length n by collecting all α such that:

(A) $[\alpha]_{(4i-3)-4i}\in \bigcup_{j=1}^{4} F_j([\gamma]_i)$ for $1\leqslant i\leqslant n$.

(B) If $[\alpha]_{(4i-3)-4i}\in F_{j_i}([\gamma]_i)$ $(1\leqslant i\leqslant n)$, then $[\alpha]_{(4n+1)-(4n+3)}\in T((\sum_{i=1}^{n} j_i) \bmod 4+1)$.

Lemma 2. Let $\gamma\in B^n$. Then $|H(\gamma)|=2^{3n+1}$.

Proof. $|H(\gamma)|=2\times \prod_{k=1}^{n} |\bigcup_{j=1}^{4} F_j([\gamma]_k)|=8^n\times 2=2^{3n+1}$. *Q. E. D.*

Theorem 3. Let $\gamma\in B^n$. then *$H(\gamma)$ is a 3-independent set of Q_{4n+3}.*

Proof. Let α and β be two distinct elements of $H(\gamma)$, and assume for $1\leqslant i\leqslant n$, $[\alpha]_{(4i-3)-4i}\in F_{j_i}(x_i)$, $[\beta]_{(4j-3)-4i}\in F_{k_i}(x_i)$. Further discussions are divided into three cases:

Case 1. $j_i=k_i(1\leqslant i\leqslant n)$. Then $(\sum_{i=1}^{n} j_i) \bmod 4=(\sum_{i=1}^{n} k_i) \bmod 4$. Since $\alpha\neq\beta$, Then either:

(A) there is $1\leqslant p\leqslant n$ such that $[\alpha]_{(4p+3)-4p}\neq[\beta]_{(4p-3)-4p}$, or

(B) $[\alpha]_{(4n+1)-(4n+3)}\neq[\beta]_{(4p+1)-(4n+3)}$.

By lemma 1, either (A) or (B) would imply $d_H([\alpha]_{(4p-3)-4p},[\beta]_{(4p-3)-4p})\geqslant 4$. Thus, $d_H(\alpha,\beta)\geqslant 3$.

Case 2. $j_i=k_i(1\leqslant i\leqslant n), i\neq p$; but $j_p\neq k_p$. By lemma 2, $d_H([\alpha]_{(4p-3)-4p},[\beta]_{(4p-3)-4p})=2$. Without loss of generality, let us assume $j_p>k_p$, then

$$0 < \sum_{i=1}^{n} j_i - \sum_{i=1}^{n} k_i = j_p - k_p < 4$$

implying $(\sum_{i=1}^{n} j_i) \bmod 4 \neq (\sum_{i=1}^{n} k_i) \bmod 4$.

By lemma 2, $d_H([\alpha]_{(4p+1)-(4n+3)}, [\beta]_{(4p+1)-(4n+3)}) \geq 1$. Thus, $d_H(\alpha, \beta) \geq 3$.

Case 3. There exist $1 \leq p \neq q \leq n$ such that $j_p \neq k_p$ and $j_q \neq k_q$.

By lemma2, $d_H([\alpha]_{(4r-3)-4r}, [\beta]_{(4r-3)-4r}) = 2$ for $r \in \{p, q\}$. Thus, $d_H(\alpha, \beta) \geq 4$.

Combining case 1-case 3, this proposition is proved. *Q. E. D.*

Now we propose a recursive procedure for finding a 3-independent set of Q_n below:

Procedure 3-INDSET

Input: a positive integer n.

Output: a 3-independent set of Q_n.

begin　if $n=1$ then return $(\{0\})$;　else if $n=2$ then return $(\{001\})$;　else if $n=3$ then return $(\{000, 111\})$;　else

begin　$S \leftarrow$ 3-INDSET $([n/4])$;　if $n - 4 \times [n/4] = 3$ then return $(\bigcup_{\alpha \in S} H(\alpha))$;　else if $n - 4 \times [n/4] = 2$ then return $((\{\beta : 0\beta \in \bigcup_{\alpha \in s} H(\alpha)\})$;　else if $n - 4 \times [n/4] = 1$ then return $((\{\beta : 00\beta \in \bigcup_{\alpha \in s} H(\alpha)\})$;　else return $(\{\beta : 000\beta \in \bigcup_{\alpha \in s} H(\alpha)\})$

end

end

Theorem 4. 3-INDSET when run on n returns a 3-independent set of Q_n.

Proof. By induction on n.

Basis step. This proposition is obvious for $1 \leq n \leq 3$.

Inductive step. Assume this proposition is true for $n < p (\geq 4)$. Now consider $n = p$. It results from the induction hypothesis that $d_H(\alpha, \beta) \geq 3$ for any distinct $\alpha, \beta \in$ 3-INDSET$([p/4])$. Without loss of generality, let us assume $[\alpha]_i \neq [\beta]_i$ for $i \in \{a, b, c\}$, then for any $\gamma \in H(\alpha), \theta \in H(\beta)$,

$$[\gamma]_{(4i-3)-4i} \neq [\theta]_{(4i-3)-4i} \text{ for } i \in \{a, b, c\}$$

Fromtheorem 3, $H(\alpha)$ is a 3-independent set of $Q_{4[p/4]+3}$ for every $\alpha \in$ 3-INDSET$([p/4])$. Hence, 3-INDSET (p) is a 3-independent set of Q_p, implying this proposition is true for $n = p$, too. By the induction principle, this proposition is true for every positive integer. *Q. E. D.*

Let $C(n)$ denote the cardinality of the 3-independent set of Q_n obtained by procedure 3-INDSET.

Theorem 5　$C(n) = 2^{n - [\log_2 n] - 1}$.

Proof. Lemma 2 results in the following recursive equations about $C(n)$:

$$C(1) = C(2) = 1, C(3) = 2 \tag{1}$$

$$C(n) = 2^{3[\frac{n}{4}] + n \bmod 4 - 2} C([\frac{n}{4}]) \text{ for } n > 3 \tag{2}$$

Without loss of generality, let us assume $4^k \leq n < 4^{k+1}$. By recursively applying equation (2) and nothcing that $[[n/4]/4] = [n/4^2]$ for every positive integer n, we immediately obtain

$$C(n) = 2^{3\sum_{i=1}^{k}[\frac{n}{4^i}] + \sum_{i=0}^{k-1}\left([\frac{n}{4^i}] \bmod 4\right) - 2k} C([\frac{n}{4^k}]). \tag{3}$$

Let $[c_k c_{k-1} \cdots c_0]_4$ be the representation of n in radix 4 (Notice that $c_k \geqslant 1$). Then

$$C(n) = 2^{3\sum_{i=1}^{k}\sum_{j=i}^{k} c_j 4^{j-i} + \sum_{i=0}^{k-1} c_i - 2k} C[([\frac{n}{4^k}]). \quad (4)$$

Let $S_i = \sum_{j=i}^{k} c_j 4^{j-i}$. Then for $1 \leqslant i \leqslant k-1$, $S_i - S_{i+1} = c_i$ (5)

By adding all equations in (5) together and making some elementary transforms, we obtain

$$3\sum_{i=1}^{k} S_i + \sum_{i=0}^{k-1} c_i = 4S_1 - S_k + c_0 \quad (6)$$

Notice that $S_k = c_k$, $4S_1 = n - c_0$, (6) can be turned into

$$3\sum_{i=1}^{k} S_i + \sum_{i=0}^{k-1} c_i = n - [\frac{n}{4^k}] \quad (7)$$

From (4) and (7). we obtain $C(n) = 2^{n-[\frac{n}{4^k}]-2k} C([\frac{n}{4^k}])$ (8)

Case 1. $[\frac{n}{4^k}] = 1$. Then $2k = [\log_2 n]$. By (1) and (8), $C(n) = 2^{n-[\log_2 n]-1}$.

Case 2. $[\frac{n}{4^k}] = 2$. Then $2k = [\log_2 n] - 1$. By (1) and (8), $C(n) = 2^{n-[\log_2 n]-1}$.

Case 3. $[\frac{n}{4^k}] = 3$. Then $2k = [\log_2 n] - 1$. By (1) and (8), $C(n) = 2^{n-[\log_2 n]-1}$.

Combining case 1-case 3, we obtain $C(n) = 2^{n-[\log_2 n]-1}$. *Q. E. D.*

Theorem 6 $I_3(n) \geqslant C(n) = 2^{n-[\log_2 n]-1}$.

Proof. From theorem 5 and the fact that $I_3(n) \geqslant T(n)$. *Q. E. D.*

Theorem 7 $I_3(n) \leqslant [2^n/(n+1)]$.

Proof. Let S be a 3-independent set of Q_n. For any two elements α, β of S,

$$\{\gamma : d_H(\gamma, \alpha) = 1\} \cap \{\gamma : d_H(\gamma, \beta) = 1\} = \varnothing$$

implying $|S| + \Sigma_{\alpha \in S} |\{\gamma : d_H(\gamma, \alpha) = 1\}| \leqslant 2^n$.

Hence, $|S| \leqslant [2^n/(n+1)]$, implying the proposition. *Q. E. D.*

By theorems 6-7, we obtain

Theorem 8 If $n = 2^k - 1$ $(k \geqslant 2)$, then $I_3(n) = 2^{n-[\log_2 n]-1}$.

2 Application of Neural Associcitive Memories

In a digital computer a desired set of information called a memory is recalled when the correct address of the memory is given. In contrast to this, associative memory (AM) is: afull set of the information of a memory is recalled by a portion of the memory' s information. Some recurrent neural networks are candidates for AM because their dynamical behavior exhibits asymptotically stable equilibria. This time evolution of such a neural network toward one of its equilibria can be interpreted as the evolution of an imperfect pattern twoard the correct (stored) pattem [4~6] Whenimplementing AM by a recurrent netural network, the key problem is how to store each desired pattem as an asymptotically stable equilibrium of the network and how to control the extent of the basin of attraction of each stored patter. Therefore, one should choose a set of vertors whose pairwise Hamming distance is no less than 3 as the set of patterns to be stored in a neural network, and such a set can be obtained by executing the algorithm presented in the preceding section.

An n-order Hopfield network[4] is a network composed of n processing elements (neurons) in which (1) w_{ij} is the weight from neuron j to neuron i, $W(w_{ij})_{n\times n}$ is symmetric and zero-diagonal; θ_i is the threshold of neuron i, $\boldsymbol{\theta}=(\theta_1,\theta_2,\cdots,\theta_n)^T$;

(2) $x_i(t)$ is the state of neuron i at time t, $X(t)=(x_1(t),x_2(t),\cdots,x_n(t))^T$;

(3) the network is assigned an initial state $X(0)$; at each $t>0$, the states of some neurons are updated in this way: if $\sum_{j=1}^{n} w_{ij}x_j(t)\geqslant\theta_i$, then $x_i(t+1)=1$; otherwise $x_i(t+1)=-1$.

A Hopfield network can be described by a triple $(W,\boldsymbol{\theta},R)$ where R describes the order in which the states of the neurons are updated.

Let $(W,\boldsymbol{\theta},R)$ be a Hopfield network, $X^*=(x_1^*,x_2^*,\cdots,x_n^*)^T$ be a vector. If for $1\leqslant i\leqslant n$, $x_i^*=\mathrm{sgn}(\sum_{j=1}^{n} w_{ij}x_j^*-\theta_i)$, where $\mathrm{sgn}(x)$ equals-1 or according as x is negative or nonnegative, then X^* is called an equilibrium of the network. If starting from any initial state, the network will finally reach one of its equilibria, then the network is called globally stable.

An n-order sequential Hopfield network is one such that only a single neuron is updated at each t and the updating order is: $x_1\to x_2\to\cdots\to x_n\to x_1\to x_2\cdots$. Such a network can be described simply by the ordered pair $(W,\boldsymbol{\theta})$. It was shown[4] that a sequential discrete Hopfield network is globally stable, makes them candidates for AM. If the desired vectors are asymptotically stable equilibria of such a network, then when an incomplete (incorrect) pattern is applied, the network can find the complete (correct) pattern

In [6～7], the problem of finding a Hopfield network $(W,\boldsymbol{\theta})$ for storing a set of binary-valued vectors $\{X^{(k)}=\{x_1^{(k)},x_2^{(k)},\cdots,x_n^{(k)}\}^T: 1\leqslant k\leqslant m\}$ has been turned into solving the following linear program:

$$\text{(LP)}\quad \text{minimize } [\sum_{i=1}^{n}\sum_{j=1}^{n}(w_{ij}^1+w_{ij}^2)+\sum_{i=1}^{n}(\theta_i^1+\theta_i^2)]$$

$$\text{subject to } \sum_{j=1}^{n} x_j^{(k)}(w_{ij}^1-w_{ij}^2)+\sum_{i=1}^{n}(\theta_i^1+\theta_i^2)]\geqslant\varepsilon \quad \text{for } x_i^{(k)}=1,$$

$$\sum_{j=1}^{n} x_j^{(k)}(w_{ij}^1-w_{ij}^2)+\sum_{i=1}^{n}(\theta_i^1+\theta_i^2)]\leqslant-\varepsilon \quad \text{for } x_i^{(k)}=-1,$$

$$(w_{ij}^1-w_{ij}^2)-(w_{ji}^1-w_{ji}^2)=0,\ w_{ii}^1-w_{ii}^2=0 \quad \text{for } 1\leqslant i,j\leqslant n,$$

$$(w_{ij}^1\geqslant0,\ w_{ij}^2\geqslant0,\ \theta_i^1\geqslant0,\ \theta_i^2\geqslant0 \quad \text{for } 1\leqslant i,j\leqslant n,$$

where ε is a small positive number. $w_{ij}=w_{ij}^1-w_{ij}^2$, $\theta_i=\theta_i^1-\theta_i^2$ for $1\leqslant i,j\leqslant n$, There are efficient algorithms for linear programming (such as simplex algorithm as well as Karmakar aloorithm).

Example 1. Choose $X^{(1)}=(1,-1,1,-1-1,1,1,-1)^T$, $X^{(2)}=(1,-1,-1,1,1,1,-1,1)^T$, $X^{(3)}=(-1,-1,1,1,1,-1,1,-1)^T$, $X^{(4)}=(1,-1,1,-1,1,-1,1,-1)^T$ from the set obtained by executing 3-INDSET on $n=8$. Solving (LP), we obtain a Hopfield network shown in Fig. 1.

Example 2. Choose $X^{(1)}=(1,-1,1,1,-1,-1,1,1,-1,-1)^T$, $X^{(2)}=(1,-1,1,1,1,1,1,1,-1,-1,-1)^T$, and $X^{(3)}=(-1,-1,-1,-1,1,-1,1,-1,1)^T$ from the set obtained by executing 3-INDSET on $n=10$. Solving (LP), we obtain a Hopfield network shown in Fig. 2.

For investigating the capability of Hopfield network to store a set of vectors provided by the algorithm 3-INDSET, we design a simulation experiment as follows:

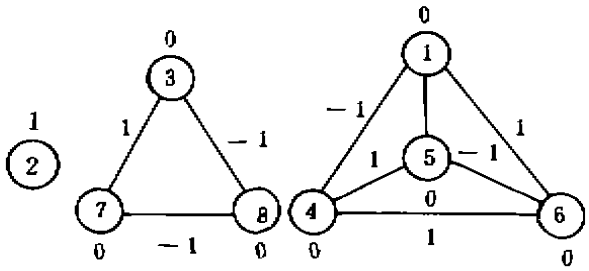

fig. 1 a desired Hopfield network

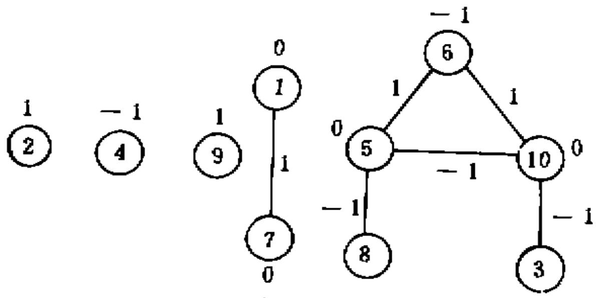

fig. 2. a desired Hopfield network

(1) For each $(p,q)\in\{(6,4)(7,5),(8,6),(9,7),(10,8)\}$, Ten sets of qvectors are randomly selected from the set obtained by running 3-INDSET on p. Thus fifty sets of vectors are produced.

(2) For each of the fifty sets, a Hopfield network is obtained by solving the linear program (LP).

(3) For each of the fifty Hopfield networks, to check whether or not the vectors in the corresponding set are all equilibria of the Hopfield network.

Our experiments are carried out on a 386-microcomputer. The results show that the fifty sets of vectors are all successfully stored ina Hopfield network. Therefore the algorithm 3-INDSET can provide a set of vectors which can be stored with a Hopfield network.

3 SUMMARY

In this paper, we present an algorithm for finding a 3-independent set of n-cube. Computer simulations show that the resulting set can be effectively stored with a Hopfield network. Our further research will be focused on how to store a set of vectors with a Hopfield network so that each of them has an attraction radius of at least 1.

References

1 J A Bondy, U S R Murth. *Graph Theory with Applications*. New York: Elsevier, North-Holland, 1976

2 M R Garey, D S Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA. Freeman, 1979

3 A V Aho, *et al*. *The Design and Analysis of Computer Algorithms*. Reading, MA, Addison-Wesley, 1974

4 J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci. U. S. A.*, 1982, 79: 554～558

5 N Michel, J A Farrell, Associative memories via artificial neural networks. *IEEE Trans. Control Syst. Mag*. (3): 6～17

6 Yang Xiaofan. *Fault Tolerance and Fault Dignosis: Special Topics on Neural Networks and Multiprocessor systems*. Ph. DDissertation, Chongqing University, 1994

7 Yang Xianfan *et al*. Fault Tolerance Analysis and Design of Neural Associative Memories. In *Neurl Network Theory and Applications. 94 Current Advance* (Huang Zailu and Yang Zhongkai ed.), Huazhong Science and Technology University Press, 1994: 71～74